

**FLETNIX**

**See what's next**

WATCH ANYWHERE. CANCEL ANYTIME.

JOIN FREE FOR A MONTH

**FLETNIX**

## Functioneel & Technisch ontwerp

Nicky Romeijn  
Student nr. 535507

Docent  
Mark Giessen

Opdracht  
Design and build a scaleable webapplication

# Inloggen op productieomgeving

Zie onderstaande gegevens om in te kunnen loggen op de productieomgeving. Om in te loggen ga naar <https://fletnix.azurewebsites.net> en klik op de signin button rechtsboven in het scherm.

Daarnaast heb ik (per ongeluk) beide mijn apps in de verenigde staten gedeployed. Dit kan ervoor zorgen dat de requests en responses en wat langere tijd op zich laten wachten dan normaal.

Fletnix: <https://fletnix.azurewebsites.net>

Username: [nicky.romeijn@gmail.com](mailto:nicky.romeijn@gmail.com)

Password: *Welkom123!*

Let op dat je https gebruikt. Het forceren naar https is nog niet gelukt op de productieomgeving dus momenteel worden beide http en https ondersteund. Inloggen vanaf de http variant werkt niet i.v.m. client permissions op identityserver.

# 1 Niet functionele requirements

- Ontwikkeld in Microsoft ASP .Net 5.0 Core
- Wordt gehost op het Microsoft Azure Platform
- Maakt gebruik van ASP .Net MVC
- De applicatie moet veilig zijn. Gebruik de OWASP top 10 om de meest voorkomende onveiligheden op te sporen en af te dichten.
- De applicatie is aantoonbaar highly-scalable
- Repository pattern
- Provider pattern
- Entity framework
- Identity server 4

## 1.1 Aandachtspunten m.b.t. Bottlenecks

- Maximale user throughput (connection pool)
- Te grote hoeveelheden data naar de front-end sturen
- Zoek functionaliteiten, hoe kan er gezocht worden naar films en hoe wordt dit efficiënt?
- Verticaal of horizontaal schalen?
- Indexing op database kolommen
- Entiteiten updaten los; i.p.v. gezamenlijk met model? (bad of good practice?)

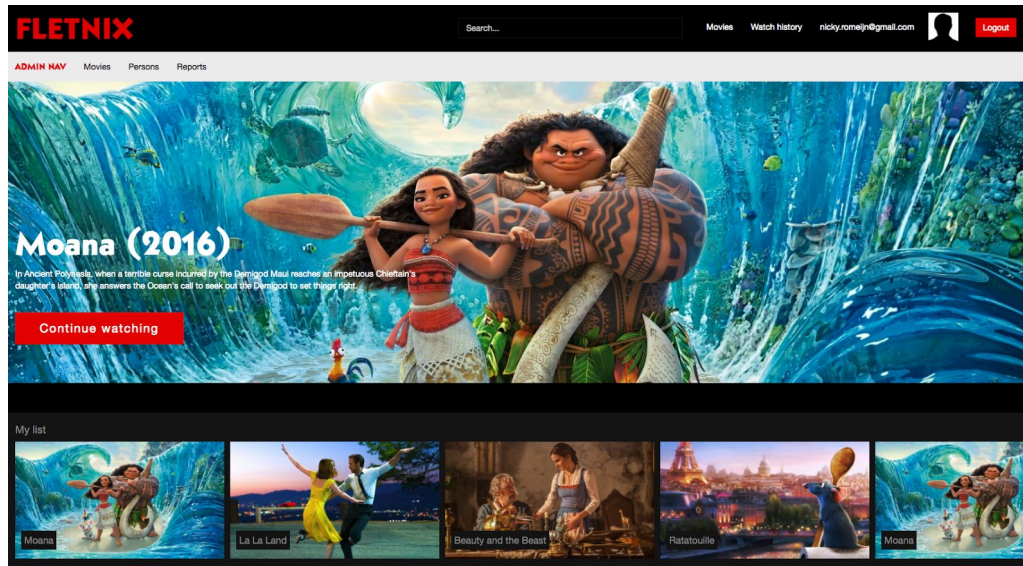
## 2 Use cases / Functionele requirements

### 2.1 Use case 1: Beheren film

<i>Use Case: Beheren film</i>	
<i>Purpose:</i> Onderhouden van filmgegevens, toevoegen nieuwe items en aanpassen van bestaande items	
<i>Description of use case:</i> Gebruiker kiest een item wijzigt de gewenste gegevens	
<i>Primary actor:</i> Applicatiebeheerder <i>Secondary actor:</i> -	
<i>Stakeholders and interests:</i> Klant	
<i>Preconditions:</i> -	
<i>Postconditions (Success Guarantee):</i> Gewenste wijzingen doorgevoerd	
<i>Basic Flow (Main Success Scenario)</i>	
<i>Actor action</i>	<i>System responsibility</i>
1. Applicatiebeheerder constateert dat er gegevens gewijzigd moeten worden.	2. Systeem toont een lijst van films
3. Applicatiebeheerder kiest film	
4. Applicatiebeheerder wijzigt gegevens van de gekozen film (inclusief cast, genre en directors van de betreffende film)	5. Systeem slaat gegevens op
<i>Alternative flows</i>	
A3. Applicatiebeheerder kiest om een nieuwe film in te voeren	
6. Applicatiebeheerder voert gegevens in (inclusief cast, genre en directors)	7. Systeem slaat gegevens op
A4. Applicatiebeheerder geeft aan geselecteerde film te willen verwijderen	8. Systeem verwijdert film

## Wireframes

Om te experimenteren met dot net MVC heb ik voor deze schermen geen wireframes gemaakt en heb ik deze on-the-fly in elkaar gezet; eerst middels scaffolding en vervolgens ( omdat naar mijn idee alle entities los beheren een ramp is) een gezamenlijke crud pagina gemaakt voor movies inclusief bijbehorende entiteiten (awards/nominaties,acteurs en directors).



Afbeelding 2.1.1 Homepage als een gebruiker ingelogd is)

Homepage (mits gebruiker ingelogd is en admin is). Voor een admin is er een extra navigatiebalk bovenin het scherm. Via deze navigatiebalk is het mogelijk bij het beheer gedeelte te komen van films.

	Title	Duration	Description	Publication year	Price	Uri	Previous part			
	Alien: Covenant	122	The crew of a colony ship, bound for a remote planet, discover an uncharted paradise with a threat beyond their imagination, and must attempt a harrowing escape.	2017	2.50	http://www.imdb.com/title/tt2316204/?ref_=rv_sr_1	Aliens	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Beauty and the Beast	129	An adaptation of the fairy tale about a monstrous-looking prince and a young woman who fall in love.	2017	3.00	http://www.imdb.com/title/tt271200/?ref_=rv_sr_1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Guardians of the Galaxy Vol. 2	136	The Guardians must fight to keep their newfound family together as they unravel the mystery of Peter Quill's true parentage.	2017	2.50	http://www.imdb.com/title/tt3896198/?ref_=rv_sr_1	Guardians of the Galaxy	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	La La Land	133	A jazz pianist falls for an aspiring actress in Los Angeles.	2016	3.00	http://www.imdb.com/title/tt376958/?ref_=rv_sr_1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Moana	107	In Ancient Polynesia, when a terrible curse incurred by the Demigod Maui reaches an impetuous Chieftain's daughter's island, she answers the Ocean's call to seek out the Demigod to set things right.	2016	2.50	http://www.imdb.com/title/tt3521164/?ref_=rv_sr_1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Guardians of the Galaxy	121	A group of intergalactic criminals are forced to work together to stop a fanatical warrior from taking control of the universe.	2014	2.50	http://www.imdb.com/title/tt2015381/?ref_=rv_sr_2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Birdman	119	Illustrated upon the progress of his latest Broadway play, a former popular actor's struggle to cope with his current life as a wasted actor is shown.	2014	3.00	http://www.imdb.com/title/tt2562332/?ref_=rv_sr_1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Harry Potter and the	106	Description of Harry Potter and the Half-Blood Prince	2008	2.50			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Afbeelding 2.1.2 [ADMIN] Overzicht films (paginated)



Als de admin klikt op 'movies' krijgt deze een overzicht van alle films te zien in tabelstructuur. (maximaal 15 films per pagina). Deze films zijn sorteerbaar op publication\_year en title, en zijn zoekbaar op title,description en movie\_id

**FLETNIX**

[Movies](#)
[Watch history](#)
[nickky.romain@gmail.com](#)
[Logout](#)

**EDITING: GUARDIANS OF THE GALAXY VOL. 2 (2017)**
[Movies](#)
[Persons](#)
[Reports](#)

Editing Movie

Back to List

MovieId

Title

Duration

Description

PublicationYear

PreviousPart

Price

Uri

Genre

Save changes

**Movie cast**

Show 10 entries

Search:

Gender	Name	Role			
	<a href="#">Chris Pratt as...</a>	<a href="#">Peter Quill</a>	<a href="#">Edit</a>	<a href="#">Delete</a>	

Afbeelding 2.1.3 [ADMIN] Edit pagina van een film

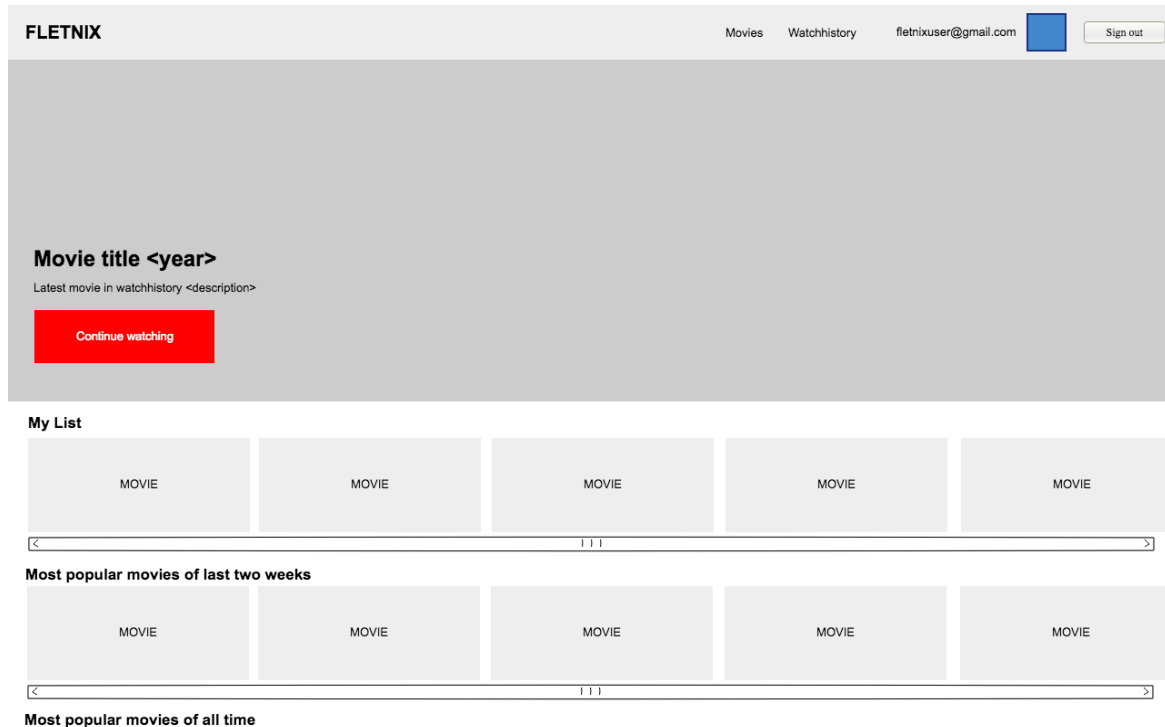
Gebruiker heeft geklikt op het 'edit' icoon in het voorgaande scherm. Vervolgens krijgt hij een pagina te zien met meerdere formulieren waarop de film en alle bijhorende entiteiten aan te passen zijn.

## 2.2 Use case 2: Kijken film

<i>Use Case: Kijken film</i>	
<i>Purpose:</i> Selecteren en bekijken film	
<i>Description of use case:</i> Gebruiker selecteert op basis van criteria de gewenste film en bekijkt deze, systeem bewaart gegevens	
<i>Primary actor:</i> Klant <i>Secondary actor:</i> Movieplayer	
<i>Stakeholders and interests:</i> applicatiebeheerder, financial manager	
<i>Preconditions:</i> Gebruiker is geregistreerd	
<i>Postconditions (Success Guarantee):</i> Gemaakte keuzes zijn vastgelegd	
<i>Basic Flow (Main Success Scenario)</i>	
<i>Actor action</i>	<i>System responsibility</i>
0. Klant logt in op de applicatie	
1. Klant kiest om films te bladeren	2. Systeem toont een lijst van films ingedeeld in de volgende categorieën: - meest populaire films ooit (gesorteerd op populariteit)  - meest populaire films van de laatste twee weken (gesorteerd op populariteit)
3. Klant kiest een film	4. Systeem toont de gekozen film gegevens (inclusief cast, genre en directors van de betreffende film). Deze gegevens zijn niet aan te passen.
5. Klant kiest om film te kijken	6. Movieplayer speelt film af op basis van de URL.
	7. Systeem slaat gekozen film en prijs op in de watchhistory
<i>Alternative flows</i>	

A5. Klant geeft aan meer films uit deze reeks te willen zien	8. Systeem toont een lijst van andere films uit dezelfde reeks.
Ga verder bij stap 3 van de Basic Flow	
A1. Klant geeft aan een film te willen zoeken	10. Systeem toont een lijst van criteria waarop gezocht kan worden
11. Klant vult één of meerdere criteria in	12. Systeem toont alle films die aan criteria voldoen
13. Ga verder bij stap 3 uit de basic flow	

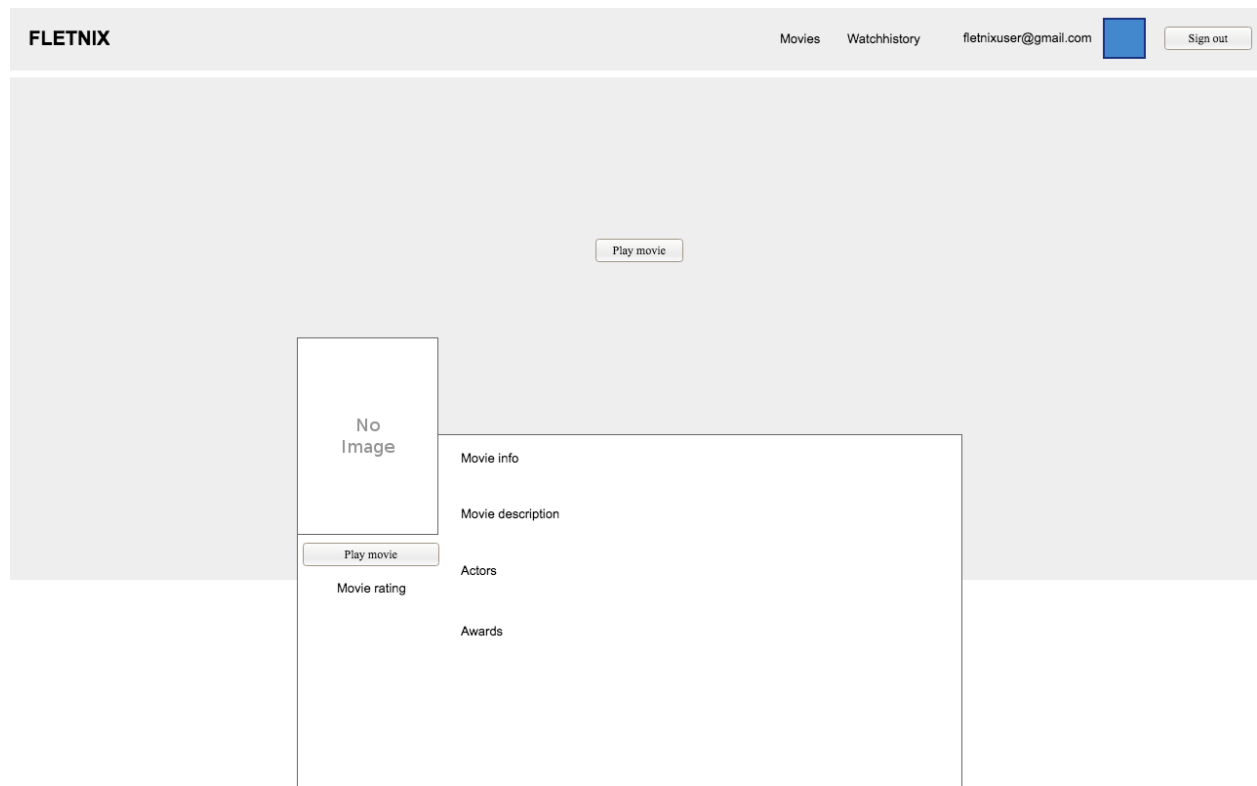
## Wireframes



*Afbeelding 2.2.1 Homepage als een gebruiker is ingelogd*

*Gebruiker (mits ingelogd) krijgt een dashboard te zien met zijn watchhistory (list), Meest populaire films van afgelopen twee weken en een lijst van meest populaire films aller tijden.*





*Afbeelding 2.2.2 Movie detail scherm voor een gebruiker die ingelogd is*

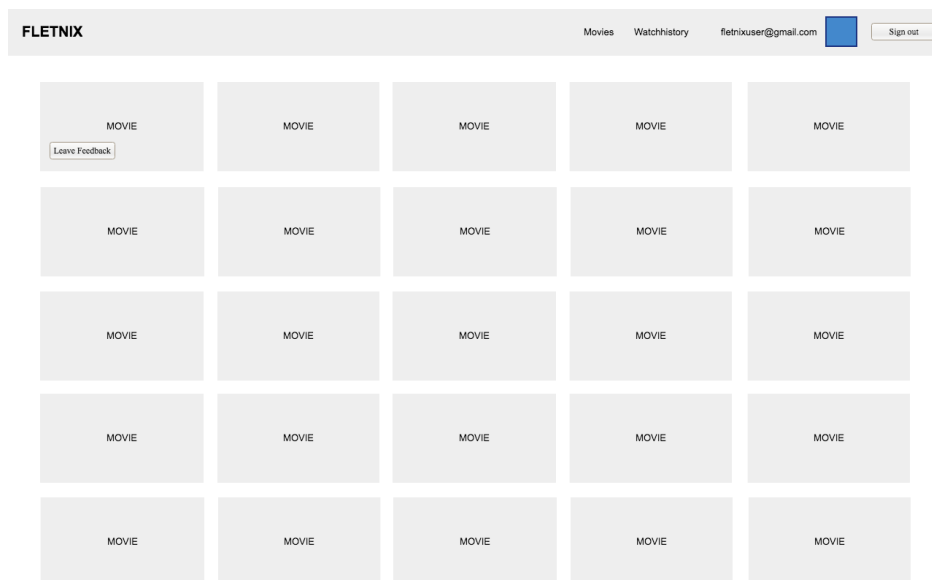
Gebruiker (mits ingelogd) heeft een film geselecteerd en krijgt hiervan het detail overzicht te zien. Op deze pagina is te zien welke acteurs meespelen, welke prijzen ze hebben gewonnen en een algemene beschrijving van de film. Gebruiker kan hier ook de film afspelen.

## 2.3 Use case 3 : Beoordelen van een film

<i>Use Case: Beoordelen van film</i>	
<i>Purpose:</i> Beoordelen van een film door het geven van feedback en een waardering.	
<i>Description of use case:</i> Gebruiker vult beoordeling in na het bekijken van een film.	
<i>Primary actor:</i> Klant <i>Secondary actor:</i> -	
<i>Stakeholders and interests:</i> financial manager	
<i>Preconditions:</i> Klant heeft film bekeken. Klant is ingelogd.	
<i>Postconditions (Success Guarantee):</i> Beoordeling van bekeken film is ingevuld.	
<i>Basic Flow (Main Success Scenario)</i>	
<i>Actor action</i>	<i>System responsibility</i>
1. Klant vraagt een overzicht van bekeken films op	2. Systeem toont een lijst van films die door de klant bekeken zijn.
3. Klant kiest een door hem/haar reeds bekeken film.	4. Systeem toont de gekozen film gegevens (inclusief cast, genre en directors van de betreffende film). Deze gegevens zijn niet aan te passen.
5. Klant kiest om beoordeling in te vullen	6. Systeem toont mogelijkheid om beoordeling (comments) en waardering (rating) in te vullen. Systeem vult daarbij zelf de datum (feedback date) waarop de beoordeling is gegeven in.
7. Klant voert beoordeling in	8. Systeem controleert ingevoerde gegevens en slaat de gegevens op indien aan voorwaarde voldaan is. Voorwaarde voor geldige invoer zijn; comments moet minimaal 10 karakters bevatten en rating moet tussen 1 en 10 liggen.
<i>Alternative flows</i>	
A5. Klant kiest ervoor om bestaande beoordeling aan te passen	Ga verder bij stap 6 van de Basic Flow

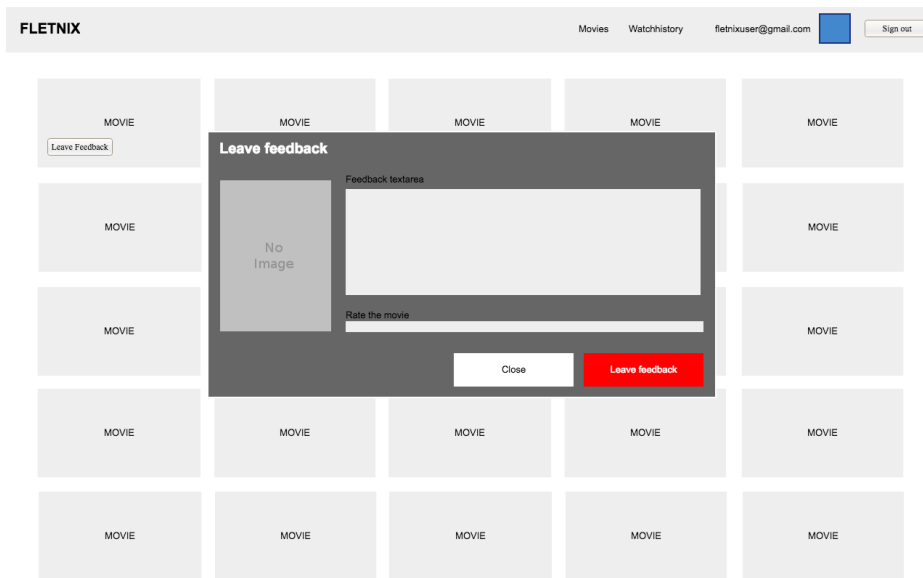
A5. Klant kiest ervoor om bestaande beoordeling te verwijderen	9. Systeem verwijdert de beoordeling van de klant.
	A8. Systeem geeft foutmelding indien niet voldaan wordt aan de voorwaarden.
9. Klant past gegevens aan.	Ga verder bij stap 8 van de Basic Flow

### 2.3.1 Wireframes



*Afbeelding 2.3.1 Overzicht watchhistory van de gebruiker*

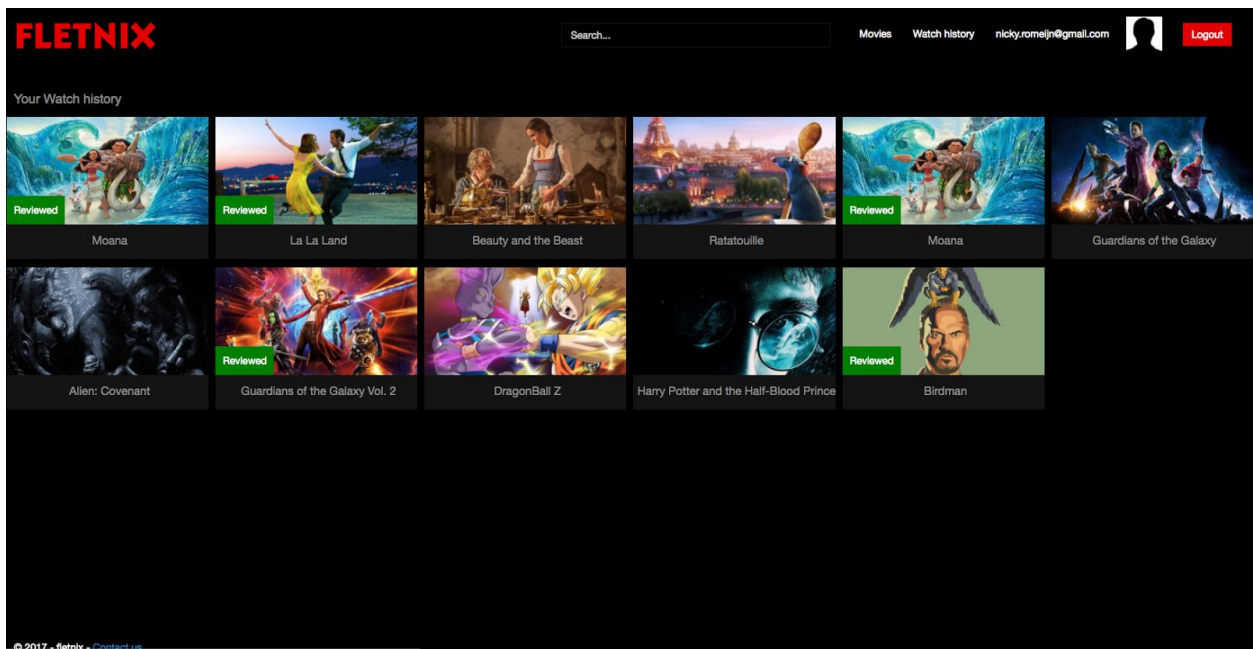
Gebruiker (mits ingelogd) kiest voor het menu item 'watchhistory' en ziet vervolgens een lijst met alle bekeken films. Door op een film te klikken kan de gebruiker naar de detailpagina van de betreffende film. Als een gebruiker 'hoovert' over een film komt er een feedback button tevoorschijn. Als een gebruiker hierop klikt verschijnt er een modal waarin de gebruiker feedback kan achterlaten voor de betreffende film. Zie volgende wireframe:



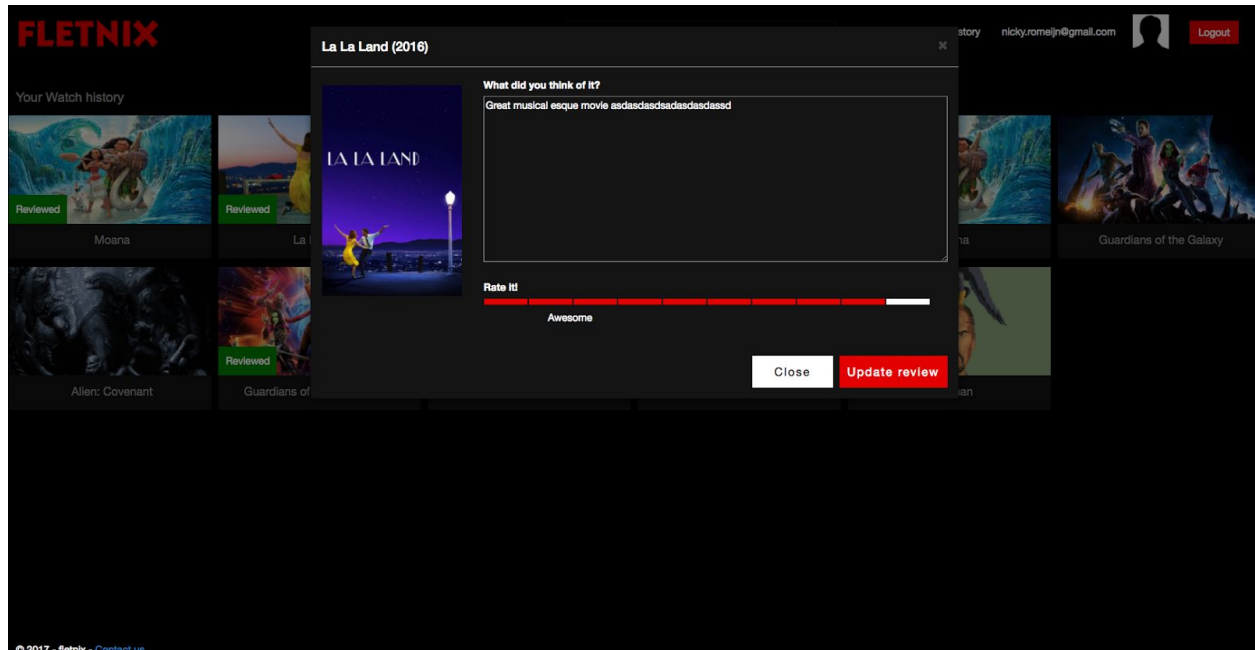
Afbeelding 2.3.2 Overzicht watchhistory van de gebruiker terwijl deze een review achterlaat bij een film.

Gebruiker ziet een popup met daarin de poster van de film, een textarea voor geschreven feedback en een rating systeem waarin de gebruiker 1-10 als score kan geven.

## Resultaat:



Afbeelding 2.3.3 Overzicht watch history van de gebruiker. In dit overzicht kan de gebruiker ervoor kiezen films te reviewen.



Afbeelding 2.3.4 Movie review box. Hierin kan de gebruiker een review aanpassen of aanmaken.

## 2.4 Use case 4 : Film nominaties

<i>Use Case: Film nominaties</i>	
<i>Purpose:</i> Het bijwerken van de nominaties van een film	
<i>Description of use case:</i> Applicatiebeheerder kiest een film waarvoor de film nominaties moeten worden bijgewerkt.	
<i>Primary actor:</i> Applicatiebeheerder <i>Secondary actor:</i>	
<i>Stakeholders and interests:</i> Klant	
<i>Preconditions:</i> -	
<i>Postconditions (Success Guarantee):</i> Gewenste wijzingen doorgevoerd	
<i>Basic Flow (Main Success Scenario)</i>	
<i>Actor action</i>	<i>System responsibility</i>
1. Applicatiebeheerder constateert dat er gegevens gewijzigd moeten worden.	2. Systeem toont een lijst van films
3. Applicatiebeheerder kiest film	4. Systeem toont de gekozen film gegevens met daarbij de eventueel aanwezig zijnde nominaties.
5. Applicatiebeheerder kiest er voor om nominaties vast te leggen.	6. Systeem toont de mogelijkheid voor het vastleggen van de nominaties.
7. Applicatiebeheerder voert één of meerdere nominaties in. Daarbij worden de volgende gegevens per nominatie vastgelegd: award, result (keuze uit Won, Nominated) en aantal gewonnen of genomineerde awards.	8. Systeem controleert de ingevoerde gegevens en slaat de wijzigingen (bij juiste invoer) op.
<i>Alternative flows</i>	

A5 Applicatiebeheerder kiest ervoor om bestaande nominatie aan te passen	Ga verder bij stap 5 van de Basic Flow
A5. Applicatiebeheerder kiest ervoor om bestaande nominatie te verwijderen	9. Systeem verwijdert de nominatie van de film.

## 2.4.1 Wireframes

Om te experimenteren met dot net MVC heb ik voor deze schermen geen wireframes gemaakt en heb ik deze on-the-fly in elkaar gezet; eerst middels scaffolding en vervolgens ( omdat naar mijn idee alle entities los beheren een ramp is) een gezamenlijke crud pagina gemaakt voor movies inclusief bijbehorende entiteiten (awards/nominaties, acteurs en directors).

[Movies](#)
[Watch history](#)
[nicky.romeijn@gmail.com](#)
[Logout](#)

[SHOWING ALL MOVIES](#)
[Movies](#)
[Persons](#)
[Reports](#)

[Create New](#)

[Search](#)
[Back to Full List](#)

	Title	Duration	Description	Publication year	Price	Uri	Previous part			
	Alien: Covenant	122	The crew of a colony ship, bound for a remote planet, discover an uncharted paradise with a threat beyond their imagination, and must attempt a harrowing escape.	2017	2.50	<a href="http://www.imdb.com/title/tt2316204/?ref_=nv_sr_1">http://www.imdb.com/title/tt2316204/?ref_=nv_sr_1</a>	Aliens			
	Beauty and the Beast	129	An adaptation of the fairy tale about a monstrous-looking prince and a young woman who fall in love.	2017	3.00	<a href="http://www.imdb.com/title/tt271200/?ref_=nv_sr_1">http://www.imdb.com/title/tt271200/?ref_=nv_sr_1</a>				
	Guardians of the Galaxy Vol. 2	136	The Guardians must fight to keep their newfound family together as they unravel the mystery of Peter Quill's true parentage.	2017	2.50	<a href="http://www.imdb.com/title/tt3896198/?ref_=nv_sr_1">http://www.imdb.com/title/tt3896198/?ref_=nv_sr_1</a>	Guardians of the Galaxy			
	La La Land	133	A jazz pianist falls for an aspiring actress in Los Angeles.	2016	3.00	<a href="http://www.imdb.com/title/tt3783958/?ref_=nv_sr_1">http://www.imdb.com/title/tt3783958/?ref_=nv_sr_1</a>				
	Moana	107	In Ancient Polynesia, when a terrible curse incurred by the Demigod Maui reaches an impetuous Chieftain's daughter's island, she answers the Ocean's call to seek out the Demigod to set things right.	2016	2.50	<a href="http://www.imdb.com/title/tt3521164/?ref_=fn_al_tt_1">http://www.imdb.com/title/tt3521164/?ref_=fn_al_tt_1</a>				
	Guardians of the Galaxy	121	A group of intergalactic criminals are forced to work together to stop a fanatical warrior from taking control of the universe.	2014	2.50	<a href="http://www.imdb.com/title/tt2015381/?ref_=nv_sr_2">http://www.imdb.com/title/tt2015381/?ref_=nv_sr_2</a>				
	Birdman	119	Illustrated upon the progress of his latest Broadway play, a former popular actor's struggle to cope with his current life as a wasted actor is shown.	2014	3.00	<a href="http://www.imdb.com/title/tt2562232/?ref_=fn_al_tt_1">http://www.imdb.com/title/tt2562232/?ref_=fn_al_tt_1</a>				
	Harry Potter and the Half-Blood Prince	106	Description of Harry Potter and the Half-Blood Prince	2008	2.50					

*Afbeelding 2.4.1 Overzicht alle films (paginated in sets van 15 results). Films zijn sorteerbaar op publication\_year en title en zoekbaar op movie\_id, title en description.*



*Afbeelding 2.4.2 CRUD overzicht van een film. Voor detail uitleg zie use case 1.*

*Afbeelding 2.4.3 Modal waarmee movie awards toegevoegd kunnen worden*

Het toevoegen of aanpassen van een movie-award. Binnen een film (edit) is het mogelijk nominaties of awards toe te voegen. Onderaan het scherm is een overzicht te vinden met alle

nominaties en awards. Door op de knop 'Add new award' te klikken is verschijnt er een modal met alle mogelijke opties voor het toevoegen van nominaties.

## 2.5 Use case 5: Overzicht awards

<i>Use Case: Overzicht awards</i>	
<i>Purpose:</i> Samenstellen van de film awards	
<i>Description of use case:</i> Weergeven van een overzicht af met film awards.	
<i>Primary actor:</i> Financial Manager <i>Secondary actor:</i> -	
<i>Stakeholders and interests:</i> Financial Manager	
<i>Preconditions:</i> -	
<i>Postconditions (Success Guarantee):</i> gewenste overzicht is afgedrukt	
<i>Basic Flow (Main Success Scenario)</i>	
<i>Actor action</i>	<i>System responsibility</i>
1. Financial Manager kiest om overzicht met film awards af te drukken.	2. Systeem toont een formulier met daarop de mogelijkheid tot het kunnen opgegeven van een bepaalde periode (jaartal vanaf en jaartal tot en met) waarop het overzicht moet zijn gebaseerd. Het jaartal betreft het jaartal (publication year) waarop de films zijn uitgebracht.
3. Financial Manager voert een periode in	4. Systeem toont een rapport met daarop van de opgegeven periode (vanaf – tot en met) gegeroepeerd per jaar: alle uitgebrachte films met daarbij per film een lijst met de betreffende award en het aantal gewonnen en/of genomineerde awards. Per film wordt tevens een totaal (grand total) aantal gewonnen en genomineerde awards vermeld.
<i>Alternative flows</i>	

A3. Financial Manager kiest alleen een vanaf jaartal (en laat dus tot en met jaartal leeg).	<p>5.Systeem toont een rapport met daarop van de opgegeven periode (vanaf opgegeven jaartal) gegroepeerd per jaar: alle uitgebrachte films met daarbij per film een lijst met de betreffende award en het aantal gewonnen en/of genomineerde awards.</p> <p>Per film wordt tevens een totaal (grand total) aantal gewonnen en genomineerde awards vermeld.</p>
A3. Financial Manager kiest geen jaartal (en laat dus vanaf en tot en met jaartal leeg).	<p>6.Systeem toont een rapport met daarop gegroepeerd per jaar: alle uitgebrachte films met daarbij per film een lijst met de betreffende award en het aantal gewonnen en/of genomineerde awards.</p> <p>Per film wordt tevens een totaal (grand total) aantal gewonnen en genomineerde awards vermeld.</p>

## Schermen:

FLETNIX			
Search...		Movies Watch history nicky.romijn@gmail.com	Logout
ADMIN NAV Movies Persons Reports			
Viewing all movies with awards from 2007 till 2018		From 2007 To 2018	Generate
		Previous	Next
2017	Awards	number of nominations	number of won awards
(1790809) Pirates of the Caribbean: Dead Men Tell No Tales	(2017) BAFTA Awards Best Cinematography	1	0
(2316204) Alien: Covenant	(2017) BAFTA Awards Best Cinematography	1	0
(2771200) Beauty and the Beast		0	0
(3896188) Guardians of the Galaxy Vol. 2		0	0
2016	Awards	number of nominations	number of won awards
(989112) Moana	(2017) Academy Awards Best Film Editing (2017) BAFTA Awards Best Editing	1	1
(3783958) La La Land		0	0
2014	Awards	number of nominations	number of won awards
(2015381) Guardians of the Galaxy		0	0
(2662232) Birdman		0	0
(2911666) John Wick		0	0
2008	Awards	number of nominations	number of won awards
Contact us			

Afbeelding 2.5.1 Rapport movie awards 2007-2018


## Use case 6: Afdrukken rating rapport

<i>Use Case: Afdrukken Rating rapport</i>	
<i>Purpose:</i> Samenstellen en afdrukken rapport met film ratings	
<i>Description of use case:</i> Afdrukken van overzicht over film ratings	
<i>Primary actor:</i> CEO (Chief Executive Officer) <i>Secondary actor:</i> -	
<i>Stakeholders and interests:</i> -	
<i>Preconditions:</i> -	
<i>Postconditions (Success Guarantee):</i> gewenste overzichten zijn afgedrukt	
<i>Basic Flow (Main Success Scenario)</i>	
<i>Actor action</i>	<i>System responsibility</i>
1. CEO kiest ervoor om rapporten af te drukken	2. Systeem toont een lijst van de beschikbare rapporten. Dit zijn: films met laagst gemiddelde rating (top 10), films met hoogst gemiddelde rating (top 10), films met rating-price index.
3. CEO kiest "laagste gemiddelde rating	4. Systeem toont een rapport met daarop de 10 films met laagst gemiddelde rating. Er worden alleen films getoond die een beoordeelde rating hebben gekregen.
<i>Alternative flows</i>	
A3. CEO kiest "films met hoogst gemiddelde rating	6. Systeem toont een rapport met daarop de 10 films met hoogst gemiddelde rating. Er worden alleen films getoond die een beoordeelde rating hebben gekregen

## 2.6.1 Schermen

**FLETNIX**

Search...

MoviesWatch historynicky.romeljn@gmail.comLogout

ADMIN NAVMoviesPersonsReports

### Average rating report

Highest average rating  
Lowest average rating  
Highest average price index rating  
lowest average price index rating

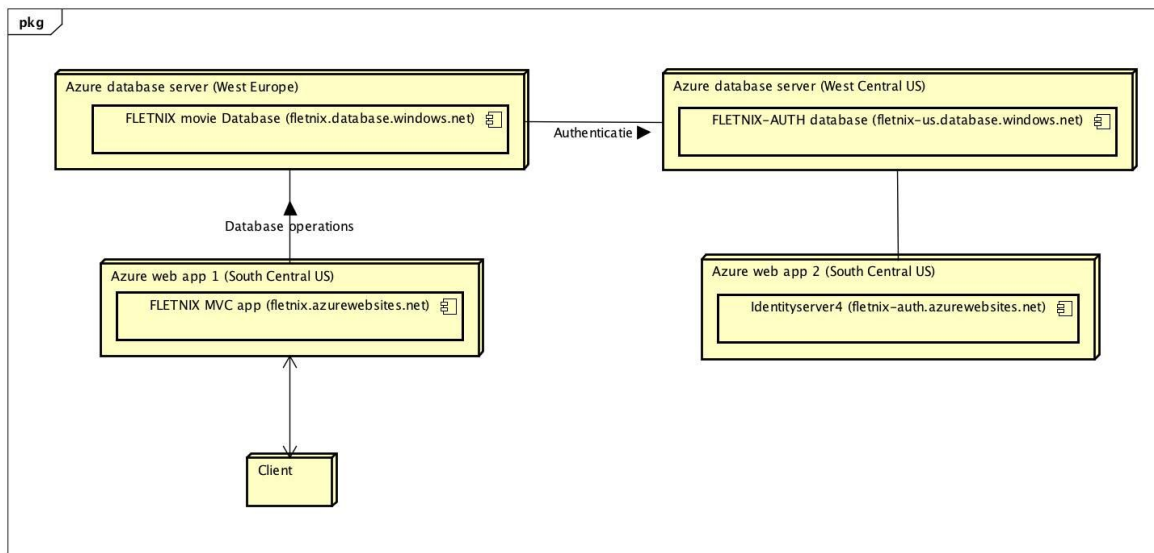
Movie id	Title	PriceIndexRating	Average rating
3783958	La La Land	3	9
999112	Moana	2.6	6.5
3896198	Guardians of the Galaxy Vol. 2	2.4	6
2562232	Birdman	1.6666666666666667	5
139653	Harry Potter and the Half-Blood Prince	1.2	3
0	#28	0	0

Afbeelding 2.6.1 Average rating rapport

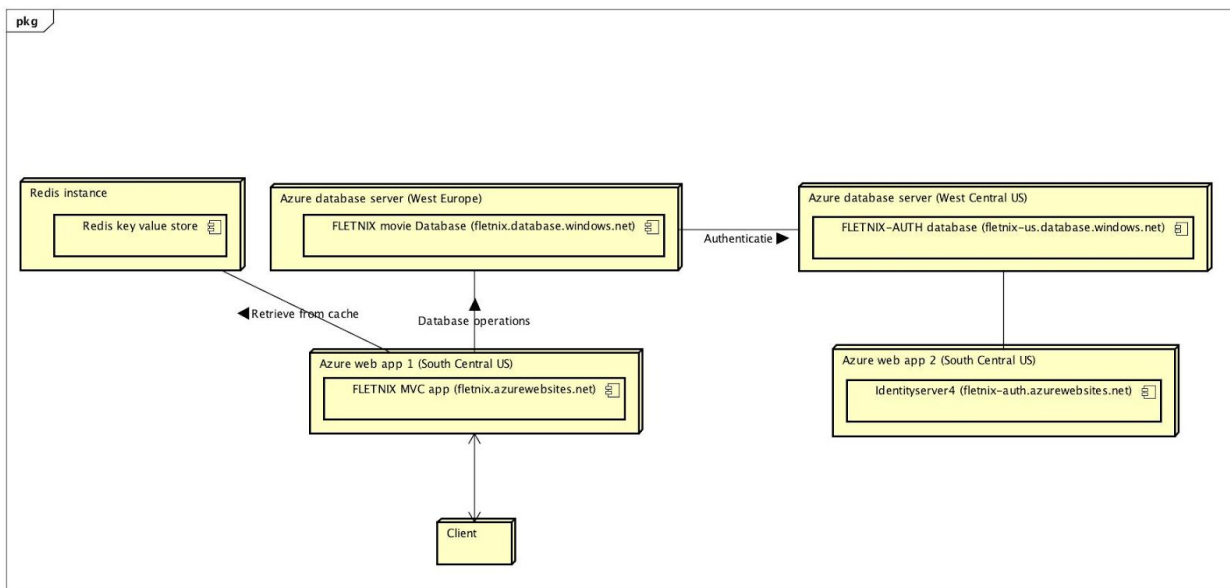
## 3 Technisch ontwerp

### 3.1 Ontwerp

#### 3.1.1 Applicatie structuur



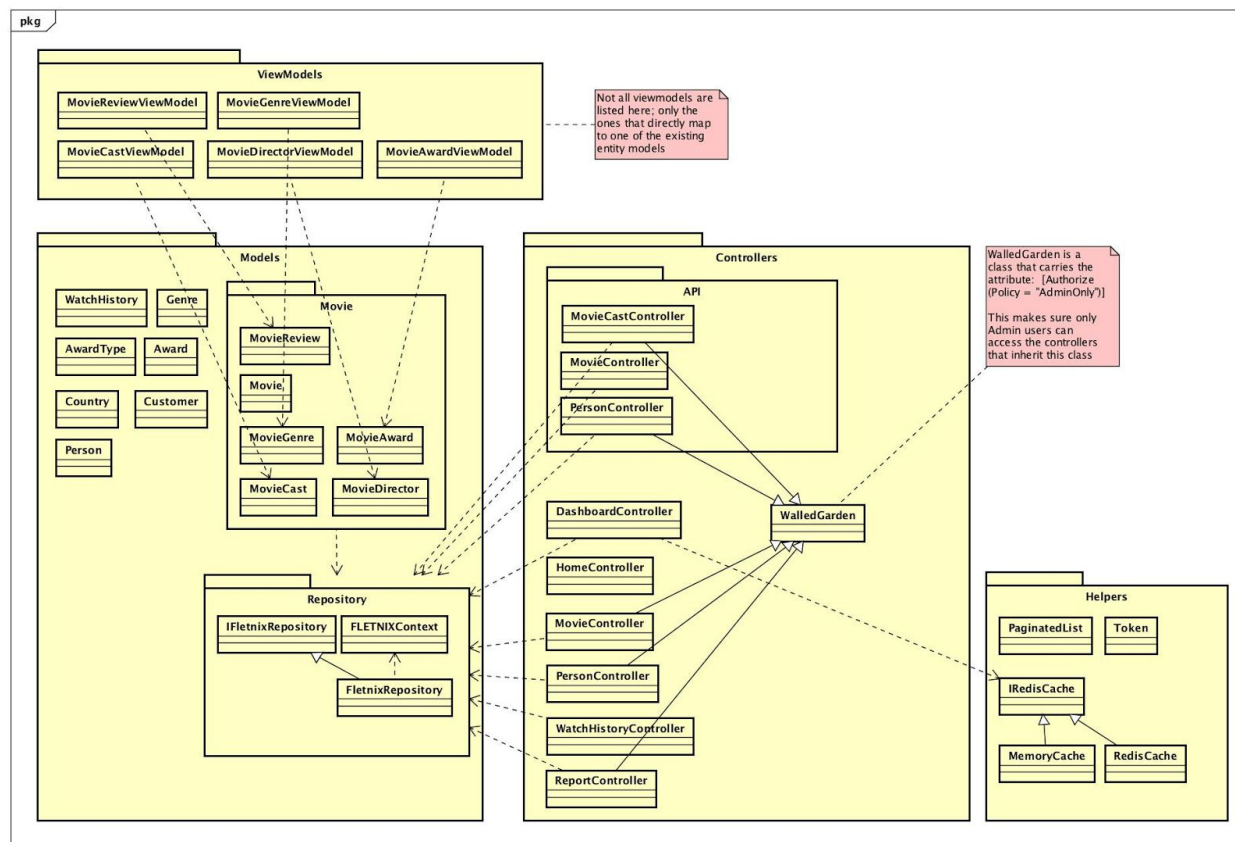
Afbeelding 3.1 Deployment diagram (prod) zonder redis



Afbeelding 3.2 Deployment diagram (local) redis

### 3.1.1.1 Verschillen tussen development en productie

Binnen de development omgeving wordt er gebruik gemaakt van caching m.b.v. redis. Een aantal grote queries waarvan het resultaat niet telkens verandert wordt opgeslagen in de redis store. Omdat azure (productieomgeving) geen support biedt voor redis met het gebruikte abonnement is er gekozen voor een andere implementatie. I.p.v. redis te gebruiken wordt er op productie binnen de MVC app een dictionary bijgehouden met dezelfde data. (Dit simuleert een vergelijkbare situatie.)



Afbeelding 3.3 Component diagram



## 3.2 Performance

### [Movies Overzicht]

#### **Compleet overzicht laten zien van alle films in de gescaffolde views / controllers.**

Dit is iets wat ik zelf niet zag als bottleneck maar als standaard practice bij het bouwen van dergelijke applicaties. Om dit op te lossen heb ik een generic paginated view class gebruikt die ervoor zorgt dat data opgedeeld wordt in sets van x records. Op deze manier blijft de hoeveelheid data die naar de client gestuurd wordt klein en kan de gebruiker bladeren door meerdere paginas van x records.

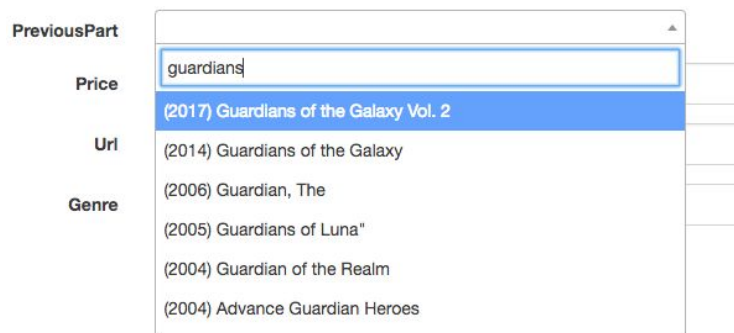
Referentie:

<https://docs.microsoft.com/en-us/aspnet/core/data/ef-mvc/sort-filter-page>

### [Movies Edit]

#### **Het laten zien van “previous\_part” bij een film.**

Initieel werd hier een select box getoond met alle films uit de database als mogelijke optie. Het resultaat hiervan was dat de browser crashte door de grote hoeveelheid data. (2mil+ records). Om dit op te lossen heb ik een searchbox geïmplementeerd die op de achtergrond een AJAX request uitvoert met daarin een zoekstring. Het respons op dit request is een lijst met films die overeenkomt met de zoekstring; dit resultaat wordt vervolgens weergegeven in de selectbox als mogelijke opties.



The screenshot shows a web form titled "PreviousPart". It contains a search box with the text "guardians" entered. Below the search box is a dropdown menu displaying a list of search results. The first result, "(2017) Guardians of the Galaxy Vol. 2", is highlighted in blue. The other visible results are "(2014) Guardians of the Galaxy", "(2006) Guardian, The", "(2005) Guardians of Luna", "(2004) Guardian of the Realm", and "(2004) Advance Guardian Heroes". To the left of the dropdown, the labels "Price", "Url", and "Genre" are visible, corresponding to the form fields.

*Afbeelding 3.1 Ajax select2 box*

#### **Het laten zien van “persons” bij het toevoegen van movie cast of directors.**

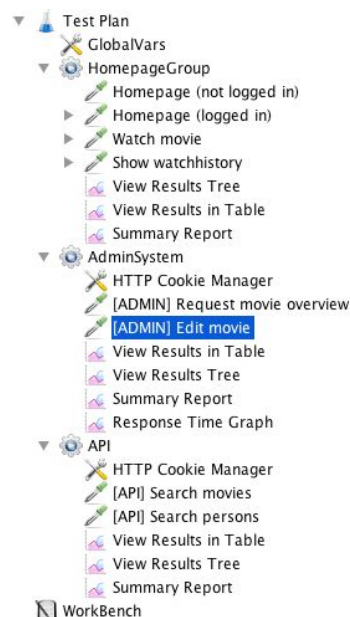
Het selecteren van persons bracht exact hetzelfde probleem met zich mee als bovenstaand punt. Door de grote hoeveelheid persons in de database werd de select box te groot en crashte de browser. Om dit op te lossen is hetzelfde gedaan als wat hierboven beschreven staat. Een select2 box met een ajax request die een lijst met personen teruggeeft op basis van een zoekstring.

### 3.2.1 Testplan

De hele casus is gerealiseerd onder OSX; mede als bewijs dat het dotnet core platform tezamen met testmogelijkheden tegenwoordig prima uitvoerbaar zijn onder dit operating system.

Als performance test software is Jmeter gebruikt. Binnen Jmeter zijn er een aantal testplannen aangemaakt die de werking van fletnix testen door een grote hoeveelheid gebruikers te simuleren en deze requests te laten doen naar performance 'zware' endpoints. Denk hierbij aan zoeken en het ophalen van grote datasets.

*Onderstaande tests zijn uitgevoerd op mijn lokale development omgeving.*



#### Test 1

Homepage | Gebruiker is ingelogd | Meest populaire films en meest populaire films van de afgelopen twee weken zijn gecached | 4 Query's worden async uitgevoerd.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Homepage (logged in)	50	11893	2778	21462	7863,76	0,000%	1,91454	131,90	14,54	70547,3
TOTAL	50	11893	2778	21462	7863,76	0,000%	1,91454	131,90	14,54	70547,3

#### Test 2

Homepage | Gebruiker is ingelogd | Meest populaire films en meest populaire films van de afgelopen twee weken zijn NIET gecached | 4 Query's worden sequentieel uitgevoerd.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Homepage (logged in)	20	248960	103840	393751	144690,24	50,000%	,04011	1,45	0,23	36950,8
TOTAL	20	248960	103840	393751	144690,24	50,000%	,04011	1,45	0,23	36950,8

Als bovenstaande resultaten vergeleken worden zie je meteen een groot verschil. In het scenario waar (data die niet vaak veranderd) gecached wordt en de queries async worden uitgevoerd zie je dat de responstijd gemiddeld 11893 ms is. (in de browser met een enkele user is dit ~30ms)

Op het moment dat de 4 query's voor het dashboard sequentieel worden uitgevoerd zonder het resultaat hiervan te cachen word het voor mijn lokale development omgeving al teveel bij 20 gebruikers. Het ophalen van de meest populaire films van de afgelopen twee weken duurt dusdanig lang dat er timeouts ontstaan binnen de applicatie. (Date comparison is langzaam?)

In de beginselen van mijn applicatie had ik i.p.v. vier query's op de homepage er maar twee. Het sequentieel uitvoeren van deze queries liep uit op een responsetijd van tussen de 33 en 40 ms. Vervolgens heb ik na het onderzoek voor de presentatie over async de query's omgezet naar Tasks om ze asynchroon uit te kunnen voeren. Hierdoor veranderde de responstijd van de pagina van 33-40ms naar 20-24ms; een snelheidswinst van ~33%.

### Tests op Azure platform; microsoft imagine tier

Na de tests lokaal uitgevoerd te hebben heb ik een poging gedaan dezelfde tests uit te voeren op azure met vergelijkbare resultaten. De 'imagine' tier database kan niet genoeg connecties tegelijkertijd aan om met een groot aantal users te testen. 35 users 4 database connecties per request = 140 connecties in totaal.

#### Test 1

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Homepage (logged in)	70	16896	1150	36721	13860,59	84,286%	1,65782	79,39	10,33	49038,6
TOTAL	70	16896	1150	36721	13860,59	84,286%	1,65782	79,39	10,33	49038,6

#### Request errors:

*<h2 class="stackerror">Win32Exception: An attempt was made to access a socket in a way forbidden by its access permissions</h2>*

*<h2 class="stackerror">SqlException: A network-related or instance-specific error occurred while establishing a connection to SQL Server. The server was not found or was not accessible. Verify that the instance name is correct and that SQL Server is configured to allow remote connections. (provider: TCP Provider, error: 0 - An attempt was made to access a socket in a way forbidden by its access permissions.)</h2>*

### OPGELOST

Nadat ik bovenstaande resultaten bekeek vond ik de responstijden toch wel erg lang duren. Ik ben vervolgens gaan kijken naar de LINQ queries en heb ze 1 voor 1 getest. Wat bleek:

```
var latestmovies = (from m in context.Movie select new PopularMoviesViewModel
{Movie = m}).OrderByDescending(m=>m.Movie.PublicationYear)
.AsNoTracking().Take(50).ToList();
```

Doordat `.Take(50)` achter `.AsNoTracking()` stond werd eerst de volledige movie tabel opgehaald, gesorteerd en van de gehele resultset (alle films) 50 entries gepakt. (Natuurlijk erg langzaam)

Nieuwe query:

```
var latestmovies = (from m in context.Movie select new PopularMoviesViewModel
{Movie = m}).OrderByDescending(m=>m.Movie.PublicationYear).Take(50)
```

```
.AsNoTracking().ToList();
```

.Take(50) wordt nu meegenomen in de vertaalde SQL query i.p.v. op applicatie niveau afgehandeld. Performance winst van **1000%**

Ik heb nogmaals exact dezelfde test uitgevoerd op de productieomgeving (azure) alleen nu met 200+ gebruikers. Zie resultaat:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Homepage (logged in)	200	1339	1242	2442	118,40	0,000%	31,41690	2080,99	207,62	67827,6
TOTAL	200	1339	1242	2442	118,40	0,000%	31,41690	2080,99	207,62	67827,6

### 3.2.2 [DATABASE INDEXES]

```
CREATE NONCLUSTERED INDEX IX_MOVIE_PUBYEAR ON Movie (publication_year)
```

```
CREATE NONCLUSTERED INDEX IX_CUSTOMER_EMAIL ON Customer  
(customer_mail_address)
```

```
CREATE NONCLUSTERED INDEX IX_WATCHHISTORY_WATCHDATE ON Watchhistory  
(watch_date)
```

Om het opvragen van records sneller te laten verlopen heb ik op een aantal kolommen die worden gebruikt voor sorteren en zoeken indexes aangemaakt.

Test	Samples	Avg	Min	Max	error
Sort pub_year zonder INDEX	50	14682ms	10631ms	21234ms	0%
Sort pub_year met INDEX	50	82ms	73ms	121ms	0%

Tabel 3.1 Verschil in snelheid met NONCLUSTERED INDEX op pub\_year

### 3.2.3 Performance verbeteringen en suggesties

#### **[Fault tolerance] Database connection pool errors**

Uit de performance test van JMeter is gebleken dat het draaien van een mssql server binnen een docker container met 1gb ram een grote bottleneck is. Bij ongeveer 60 gebruikers kwamen de eerste excepties met de melding dat er niet genoeg connecties beschikbaar waren in de connection pool. Initieel was deze error te wijden aan het asynchroon maken van 3 grote queries. Om deze query's asynchroon uitvoerbaar te maken moest er voor elke operatie een nieuwe DbContext aangemaakt worden; oftewel een nieuwe database connectie. Dit is opgelost door een using block om de initialisatie van de nieuwe connectie te zetten. Volgens de documentatie van microsoft word op het einde van dit 'block' dispose aangeroepen en wordt de connectie teruggegeven aan de connection pool.

#### **Onopgelost:**

Volgens de documentatie van entity zouden DbContext instanties direct na het uitvoeren van query's de connectie terug moeten geven aan de pool. Dit lijkt echter niet te gebeuren.

- Te weinig resources m.b.t hardware (het is geen leak?)

#### **[Caching / Redis] Date comparison query**

Erg langzaam; lijkt wel tot 3-4 seconden te duren?

Verwacht result: Een lijst met films selecteren die de afgelopen twee weken bekeken zijn.

Normale query zonder date comparison duurt een aantal milliseconden. Op het moment dat de vergelijking toegevoegd wordt neemt de query tijd toe tot 3-4sec.

#### **Oplossing**

De dataset die vergeleken moeten worden op basis van datum (meest populaire film van afgelopen twee weken) hoeft natuurlijk niet elke keer dat een gebruiker het dashboard opent opnieuw opgehaald te worden uit de database. Om dit op te lossen heb ik ervoor gekozen een redis cache toe te voegen aan het project. De lijst met films word eenmalig opgehaald (per dag) en vervolgens met een sliding window in redis gestopt. Alle opvolgende requests krijgen vervolgens te zien wat er in de redis cache zit. Dit verandert de responstijd van een aantal seconden naar enkele milliseconden.

Omdat ik gebruik maak van een 'free' microsoft imagine tier op azure is het niet mogelijk Redis te hosten binnen dit platform. Als 'noodoplossing' voor de productieomgeving heb ik de functionaliteiten van redis geabstraheerd met een interface. De productieversie van de applicatie gebruikt dan ook geen redis maar een Dictionary om de data in op te slaan. (De oplossing simuleert het serialiseren van de objecten naar JSON en weer terug.)

Testresultaten dashboard. Op dit dashboard worden vier queries uitgevoerd waaronder:

- Nieuwste films: Lijst van 50 films gesorteerd op publication\_year
- Meest populaire films van de afgelopen twee weken
- Meest populaire films allertijden
- Watchhistory van de gebruiker

In onderstaande tabel worden vier verschillende scenario's behandeld m.b.t. het resultaat en de uitvoer van deze queries.

Test	Samples	Avg	Min	Max	Err
No cache & async	50	19ms	15ms	47ms	0%
No cache & sequential	50	25ms	19ms	103ms	0%
Cached & async	50	14ms	12ms	33ms	0%
Cached & sequential	50	28ms	23ms	89ms	0%

*Tabel 3.2 Verschil tussen caching & geen caching & async & sync*

### **[Searching] Movie search query**

Binnen het admin systeem is het mogelijk te zoeken naar films op basis van titel, description en movie\_id. Deze zoek query duurt vrij lang maar is acceptabel (voor admins) gezien de grootte van de movie tabel. Om efficiënt zoeken voor de eindgebruiker mogelijk te maken is het slimmer om een dedicated text search engine te gebruiken. Bijv: Elasticsearch. Een zoekopdracht naar deze engine kan vervolgens data retourneren van een film waarmee het volledige record opgehaald kan worden uit de database.

Als alternatief op Elasticsearch heb ik gekeken naar de volgende functionaliteiten van MSSQL.

#### *Referenties:*

<https://docs.microsoft.com/en-us/sql/relational-databases/search/full-text-search>

<https://docs.microsoft.com/en-us/sql/relational-databases/search/create-and-manage-full-text-in-dexes>

Om fulltext search te kunnen gebruik moet er een plugin toegevoegd worden aan mssql server. Helaas is dit mij niet gelukt onder osx met mssql in docker.

Het wachten op zoekresultaten kan bij een aantal van 75 gebruikers tot wel 20 seconden duren. De query doet contains comparison op de titel van de film, de omschrijving en het id. Bij het endpoint voor persons wordt er gezocht binnen de kolommen firstname en lastname.

### Resultaten zonder full-text index

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
[API] Search movies	15	3686	947	5288	1575,55	0,000%	,82713	0,93	3,18	1157,0
[API] Search persons	15	16705	7757	21786	3878,53	0,000%	,51083	0,33	1,95	667,0
TOTAL	30	10196	947	21786	7151,09	0,000%	,98974	0,88	3,79	912,0

### [Asynchronicity / Queueing / Locking]

Een onderdeel waar bovenstaande concepten bij kunnen komen kijken is het genereren van rapporten. Ik ben nog niet toegekomen aan het implementeren van de laatste twee use cases. Maar heb al wel een aantal ideeën om dit zo performant mogelijk te maken. Een van de rapporten heeft als requirement dat de hoogst en laagst reviewde films in een overzicht te zien zijn. Op het moment dat er veel feedback in de database zit kan het genereren van een dergelijk rapport aardig wat tijd kosten.

Om te voorkomen dat de complete feedback tabel gelockt wordt tijdens het genereren van het rapport is het verstandig de tabel uit te lezen met een READ UNCOMMITTED isolation level. Op deze manier kunnen gebruikers feedback blijven achterlaten zelfs tijdens het genereren van het rapport.

Hiernaast wil je als gebruiker natuurlijk niet constant blijven wachten op een page-load en een timeout voorkomen tijdens het genereren van een dergelijk rapport. Om dit netjes asynchroon af te vangen en systematisch op te lossen zou een queue gebruikt kunnen worden. Op het moment dat een gebruiker kiest voor het genereren van een rapport zou er een queue message naar een derde service gestuurd kunnen worden met daarin een trigger dat er een nieuw rapport gegenereerd moet worden.

De MVC applicatie of een vierde service kan vervolgens de queue in de gaten houden en op basis van het bericht het rapport genereren (op de achtergrond). Als het rapport gegenereerd is kan deze gezien laten worden aan de gebruiker.

Afhankelijk van hoe vaak een dergelijk rapport opgevraagd moet worden is het wellicht verstandiger dit ook nog te cachen. Op deze manier verzacht je de load op de queue service en de rapport generatie service en wordt het gehele systeem meer performant.

```
using (IDbContextTransaction transaction = context.Database.BeginTransaction(
    IsolationLevel.ReadUncommitted))
```

(Het zetten van isolationlevel binnen een entity query)



<i>Test</i>	<i>Samples</i>	<i>Avg</i>	<i>Min</i>	<i>Max</i>	<i>Error</i>
<i>READ COMMITTED</i>	<i>50</i>	<i>134</i>	<i>106</i>	<i>229</i>	<i>0%</i>
<i>READ UNCOMMITTED</i>	<i>50</i>	<i>156</i>	<i>106</i>	<i>314</i>	<i>0%</i>

*Tabel 3.3 Verschil tussen read committed & read uncommitted tijdens het opvragen van alle films + awards van 2005 - 2018. De movie tabel heeft hier een index op publication\_year.*

### **[ISOLATION]**

Isolation binnen fletnix is het best aantoonbaar via de losse authenticatie server (identityserver4). Door deze los te trekken van de MVC applicatie is het mogelijk meerdere programma's te laten authenticeren via een losse centrale service. Een voorbeeld hiervan is; de webapplicatie en een Android of iPhone app. Deze kunnen door gebruik van een losse service beide authenticeren via hetzelfde systeem.

### **[Asynchronicity]**

Op mijn dashboard (de homepage van gebruikers als ze ingelogd zijn) voer ik vier queries uit die als ze synchroon uitgevoerd worden aardig wat tijd kosten ~1-2sec. Om dit op te lossen heb ik ervoor gezorgd dat deze vier queries async worden uitgevoerd. Het verschil tussen de responstijden is gemeten met JMeter. De originele responstijd van de pagina was ~33-40ms en is nu teruggelopen naar 20-24ms een snelheidswinst van ongeveer 33%.

### **[Redundancy]**

Altijd handig ongeacht welke webapp dan ook; loadbalancer met twee applicaties erachter. Tevens ook slim voor een deploymentstraat. Op deze manier blijft tijdens het uitrollen van updates altijd een oude versie draaien totdat de uitrol klaar is.

### 3.2.4 Resultaten testcases

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
[API] Search movies	15	3686	947	5288	1575,55	0,000%	,82713	0,93	3,18	1157,0
[API] Search persons	15	16705	7757	21786	3878,53	0,000%	,51083	0,33	1,95	667,0
TOTAL	30	10196	947	21786	7151,09	0,000%	,98974	0,88	3,79	912,0

*Test 1: Testresultaten: 15 users zoeken van persons / movies op title, firstname, lastname zonder full-text index (full text index niet meer kunnen testen vanwege plugin problemen + docker)*

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
[ADMIN] Show award report	50	12313	7	20134	5433,61	0,000%	2,35394	18,30	9,05	7962,0
[ADMIN] Show rating report	50	19033	13024	27421	3826,80	0,000%	1,33174	32,75	5,11	25183,0
TOTAL	100	15673	7	27421	5776,96	0,000%	2,66283	43,10	10,23	16572,5

*Test 2: Testresultaten: 50 users die een movie awards rapport opvragen van een grote range jaartallen. (Met een NONCLUSTERED INDEX op publication year)*

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
[ADMIN] Show award report	50	746	6	2331	691,45	0,000%	9,51294	73,97	36,56	7962,0
[ADMIN] Show rating report	50	935	10	4460	1390,58	0,000%	5,44544	115,98	20,90	21810,0
TOTAL	100	840	6	4460	1102,19	0,000%	10,85423	157,79	41,69	14886,0

*Test 3: Testresultaten: 50 users die een movie awards rapport opvragen van een kleine range jaartallen. (Met een NONCLUSTERED INDEX op publication year)*

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
[API] Search movies	100	29734	11140	54273	13230,17	44,000%	,99450	1,46	2,14	1500,2
[API] Search persons	100	19300	2	40182	15106,58	100,000%	1,24998	2,12	0,00	1739,2
TOTAL	200	24517	2	54273	15127,49	72,000%	1,98878	3,15	2,14	1619,7

*Test 4: Testresultaten: 50 users zoeken van persons / movies op title, firstname, lastname zonder full-text index (full text index niet meer kunnen testen vanwege plugin problemen + docker). Hoge error rate bij zoeken met veel personen tegelijk. Database connecties blijven te lang openstaan en zorgen voor timeouts.*

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
[ADMIN] Request movie overview	400	11094	130	44133	11687,35	0,500%	6,93445	190,82	26,38	28178,1
[ADMIN] Edit movie	400	14961	45	43385	17141,64	0,250%	6,93217	237,52	26,42	35086,3
TOTAL	800	13027	45	44133	14797,09	0,375%	13,83006	427,22	52,66	31632,2

*Test 5: Testresultaten: 200 users die het admin systeem tegelijkertijd gebruiken. Het opvragen van het paginated overzicht van movies en de edit pagina van een movie.*