

# WFFLIX

Een website creëren waar gebruikers coderings video's kunnen bekijken.

Team C3

Aman Maksoedan, Chevelly Somopawiro, Edwin Mulder, Hermijn Verbaan,  
Jamal Arnhem,, Amy Agterberg, Annemieke Kieft en Daphne Zwuup.

## Management samenvatting

Als studenten van ADSD Windesheim in Almere zijn wij benaderd om een platform te creëren waarin men videomateriaal kan bekijken omtrent programmeertalen en software, zodat programmeurs en geïnteresseerden op een gemakkelijke manier informatie kunnen opdoen met behulp van een video. Er wordt een platform tot stand gebracht met videomateriaal voor programmeer tutorials, tips etc. De klant wilt niet dat er alleen over C#, PHP en Javascript gesproken wordt, maar ook over andere talen zoals Unix, Python, Ruby, C++ etc. Dit zal inzicht, gemak en de mogelijkheid bieden om verschillende programmeertalen op de markt bekend(er) te maken.

# Inhoud

|  |           |
|--|-----------|
| <b>Aanleiding</b>                          | <b>4</b>  |
| <b>Visie</b>                               | <b>5</b>  |
| <b>Projectgrenzen en randvoorwaarden.</b>  | <b>6</b>  |
| <b>Vraagstelling</b>                       | <b>7</b>  |
| <b>Probleemstelling</b>                    | <b>8</b>  |
| <b>Doelstelling</b>                        | <b>8</b>  |
| <b>Usecase</b>                             | <b>9</b>  |
| Diagram gebruiker login:                   | 9         |
| Diagram werknemer bewerken en verwijderen: | 10        |
| <b>Activiteit diagram</b>                  | <b>14</b> |
| Actor logt in                              | 14        |
| <b>(Test) scenario</b>                     | <b>17</b> |
| Actor logt in                              | 17        |
| Deelpaden                                  | 17        |
| Testacties                                 | 17        |
| Actor registreert                          | 18        |
| Deelpaden                                  | 18        |
| Testacties                                 | 18        |
| <b>Actor verwijdert video</b>              | <b>19</b> |
| Deelpaden                                  | 19        |
| Testacties                                 | 19        |
| <b>Storyboards</b>                         | <b>20</b> |
| <b>Pagina lijst</b>                        | <b>22</b> |
| <b>Database Lijst</b>                      | <b>22</b> |
| <b>ERD</b>                                 | <b>23</b> |
| <b>Sequentiediagrammen</b>                 | <b>24</b> |
| Inloggen                                   | 24        |
| Aanmelden                                  | 25        |
| <b>Klassendiagram</b>                      | <b>26</b> |
| <b>Producten</b>                           | <b>27</b> |
| <b>Kwaliteitscriteria</b>                  | <b>28</b> |
| <b>Acceptatiecriteria</b>                  | <b>28</b> |



# Aanleiding

Achtergrond: De school heeft aan de eerstejaars studenten van Windesheim in Almere, van de opleiding AD Software Development, gevraagd om een platform te maken. Er moet een platform gecreëerd worden waarin videomateriaal wordt aangeboden, zodat gebruikers op een snelle en efficiënte manier informatie kunnen verwerven over verschillende programmeertalen.

Doelgroep en/of markt: Het platform legt vooral focus op het gehele spectrum aan programmeurs van beginners tot experts. Ook legt het platform focus op ieder die geïnteresseerd is in het verkrijgen van informatie over programmeertalen en software.

Stakeholders: De stakeholders van WFFLIX zijn het projectteam en de product eigenaren (Rudy and Matthijs), in naam van de opdrachtgever Windesheim. Het projectteam wordt betaald in EC's door de product eigenaren.

# Visie

De klant wilt een platform aanbieden waarin men videomateriaal kan bekijken omtrent programmeertalen en software, zodat programmeurs en geïnteresseerden op een makkelijke manier informatie kunnen opdoen met behulp van video's.

De onderwerpen kunnen informatie over: HTML, CSS, C#, PHP, Javascript, Unix, Python, Ruby, C++, XML, Ajax en Java bevatten. De visie van de klant zou geheel tot zijn recht komen als alle gebruikers weten hoe er in Unix een directory op de command line gemaakt kan worden.

# Projectgrenzen en randvoorwaarden.

Als programmeurs zijn de studenten niet verantwoordelijk voor:

- Het opleveren van content
- Nieuwe content toevoegen
- Onderhoud van de website
- Werven van nieuwe gebruikers
- Marktonderzoek en success

De minimale project eisen die de studenten moeten regelen zijn:

- Webserver die php ondersteund
- Een geldige https certificaat
- Een online database met een PhpMyAdmin login

# Vraagstelling

De minimale functies die de klant heeft aangegeven waar het product aan moet voldoen:

- In opdracht van Windesheim een platform bouwen om codeer tutorials beschikbaar te maken.
- Gebruikersregistratie
- PHP website
- Video's

# Probleemstelling

Een gebrek aan een platform waarin men gemakkelijk en doelgericht videomateriaal kan bekijken omtrent programmeertalen en software.

# Doelstelling

De gebruikers kunnen gemakkelijk informatie opdoen door middel van het platform, hierdoor kan de tijd van de doelgroep efficiënter gebruikt worden.



## Usecase

Diagram gebruiker login:

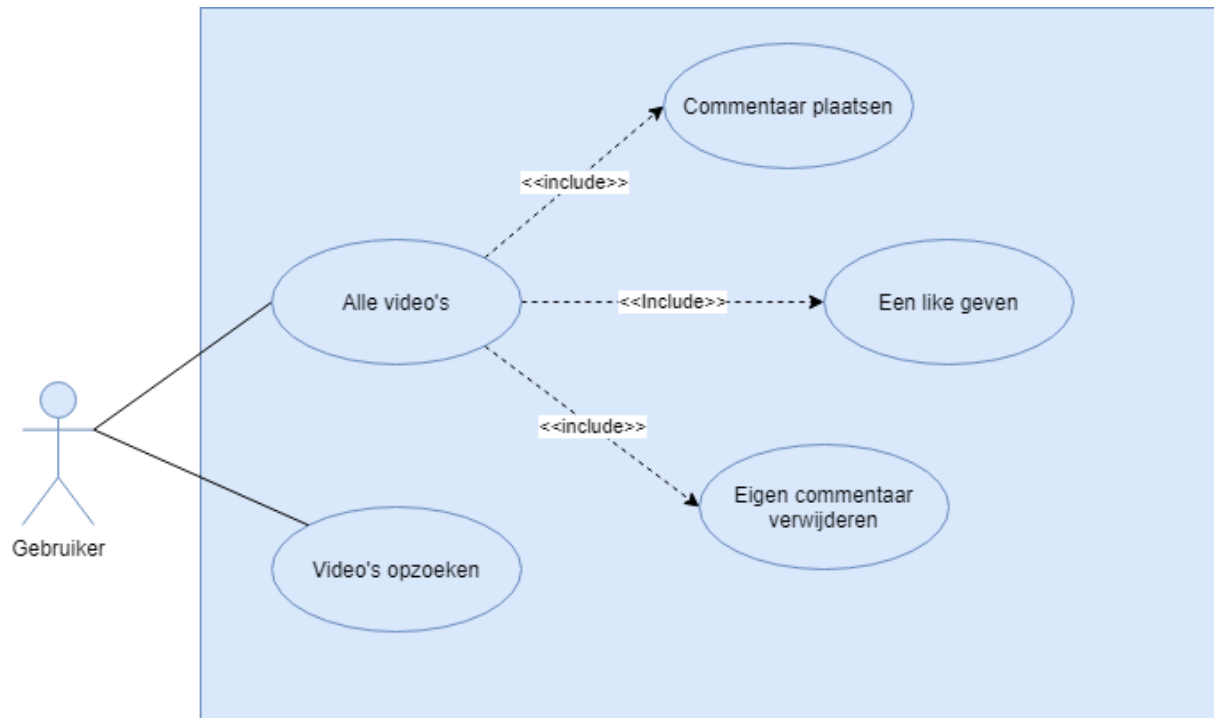
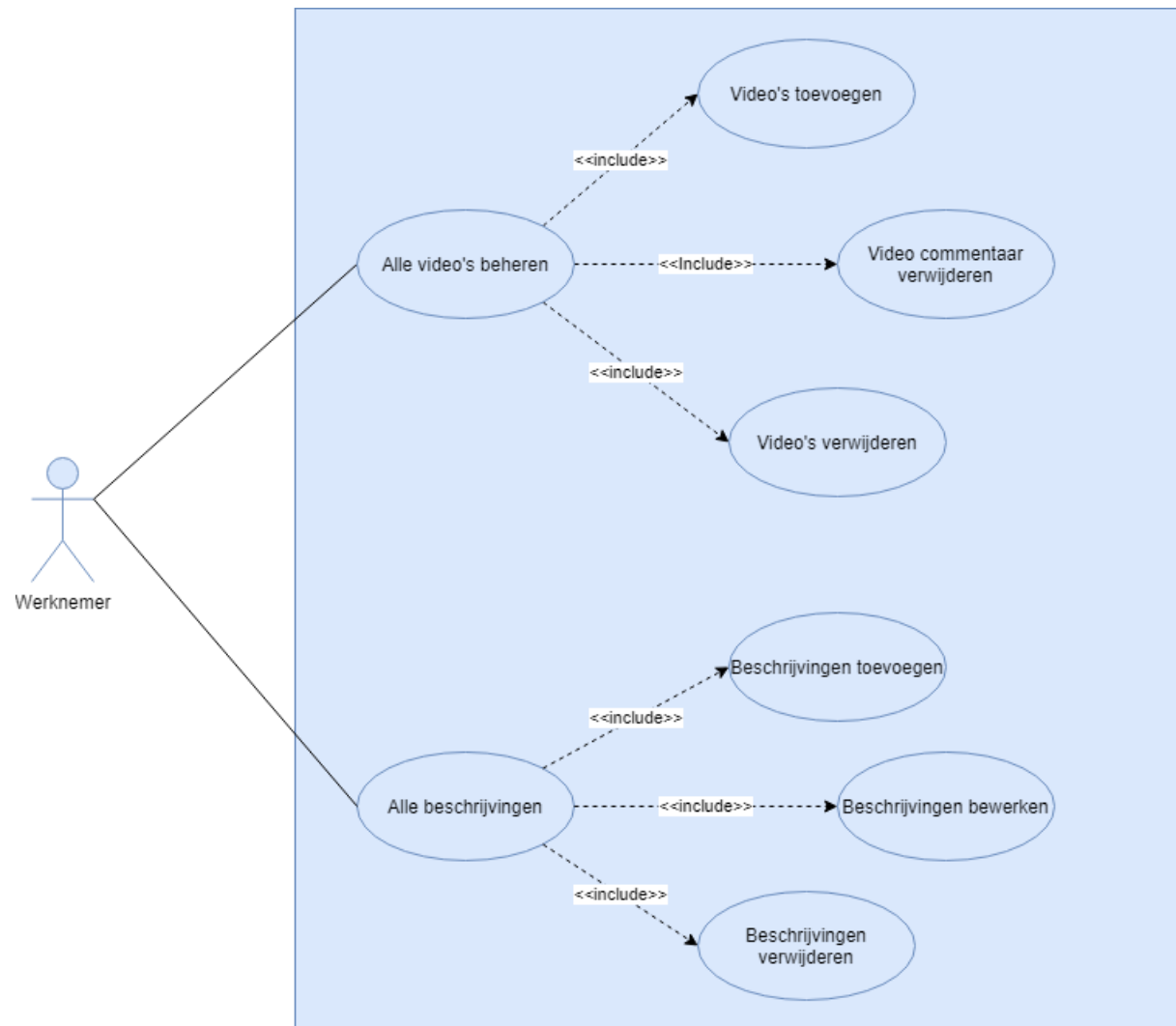


Diagram werknemer bewerken en verwijderen:



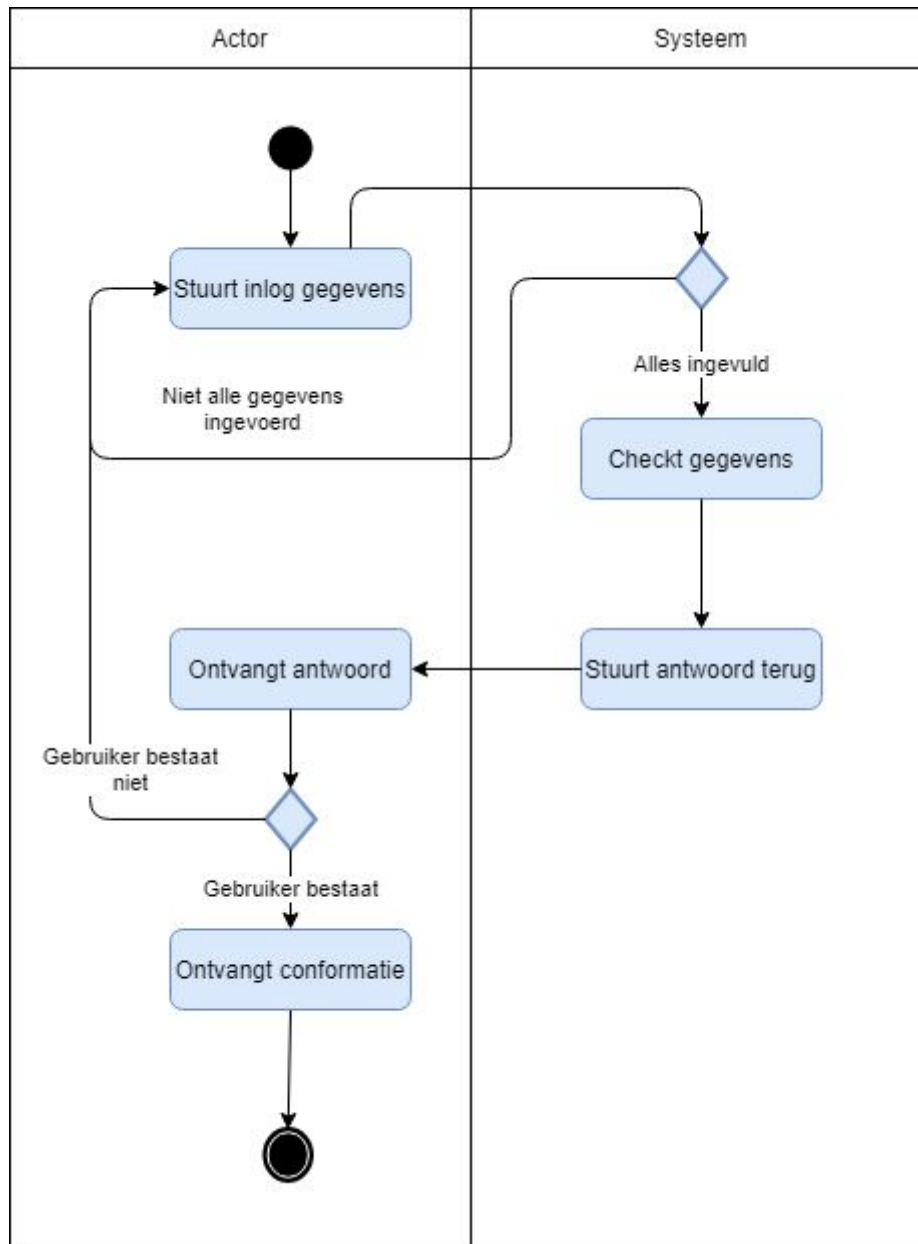
|                         |   |
|-------------------------|---|
| Nr.                     | U1.1  |
| Versie nummer           | 1.0   |
| Auteur                  | Aman Maksoedan, Chevelly Somopawiro, Edwin Mulder, Hermijn Verbaan, Jamal Arnhem.   |
| Use case naam           | Actor logt in.  |
| Actoren                 | Gebruikers.   |
| Doel                    | Het inloggen van de actor om toegang te krijgen tot video's.  |
| Triggers                | Een actor klikt op de inlog knop of drukt op enter.   |
| Pre-conditie            | Actor is geregistreerd in de database.  |
| Post-conditie           | <ol style="list-style-type: none"> <li>1. De actor is ingelogd.</li> <li>2. Actor arriveert op de home pagina.</li> <li>3. Actor heeft toegang tot de video's.</li> </ol>   |
| Hoofdscenario           | <ol style="list-style-type: none"> <li>1. Actor voert inloggegevens in.</li> <li>2. Systeem controleert of alle gegevens zijn ingevoerd.</li> <li>3. Systeem controleert of de inloggegevens bestaan in de database.</li> <li>4. Systeem geeft een bevestiging dat de actor is ingelogd, door de actor naar de homepage te sturen.</li> </ol> |
| Alternatieve scenario's | <ol style="list-style-type: none"> <li>1-1 Actor logt meer dan 2 keer verkeerd in.</li> <li>1-2 Systeem vraagt actor of zij hun wachtwoord willen veranderen.</li> </ol>  |
| Excepties               | <ol style="list-style-type: none"> <li>1. De actor heeft geen account om in te loggen.</li> </ol>   |
| kwaliteiten             | Wanneer een exceptie voorkomt keert het systeem terug naar de laatst bezochte pagina.   |

|                         |  |
|-------------------------|--|
| Nr.                     | U1.2   |
| Versie nummer           | 1.0  |
| Auteur                  | Aman Maksoedan, Chevelly Somopawiro, Edwin Mulder, Hermijn Verbaan, Jamal Arnhem.  |
| Use case naam           | Actor meld zich aan.   |
| Actoren                 | Gebruikers.  |
| Doel                    | Het aanmelden van de actor om toegang te krijgen tot de site.  |
| Triggers                | Een Actor klikt op de aanmeld knop of drukt op enter.  |
| Pre-conditie            | Actor is geregistreerd.  |
| Post-conditie           | Actor is aangemeld.<br>Gegevens zijn in de database opgeslagen.  |
| Hoofdscenario           | <ol style="list-style-type: none"> <li>1. Actor geeft gegevens op (Voornaam, achternaam, e-mailadres, adres, woonplaats).</li> <li>2. Systeem controleert of alle gegevens zijn ingevoerd.</li> <li>3. Systeem controleert of actor al bestaat in de database.</li> <li>4. Systeem verwijst naar de inlog pagina.</li> </ol> |
| Alternatieve scenario's | <ol style="list-style-type: none"> <li>1-1 Actor bestaat al.</li> <li>1-2 Systeem vraagt actor of zij hun wachtwoord willen veranderen.</li> </ol>   |
| Excepties               | Service wordt niet aangeboden in uw regio.   |
| kwaliteiten             | Wanneer een exceptie voorkomt keert het systeem terug naar de laatst bezochte pagina.  |

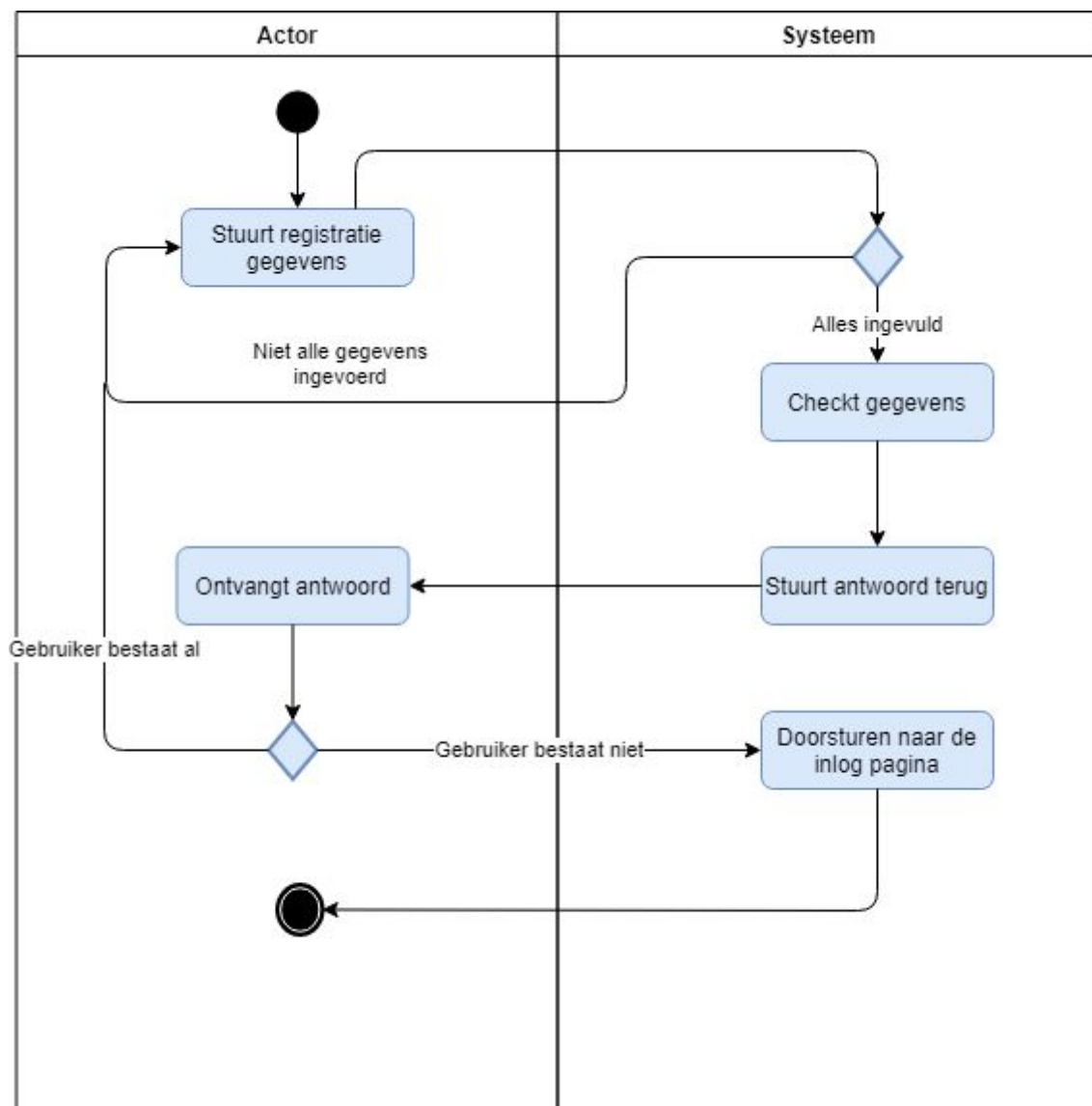
|                         |   |
|-------------------------|---|
| Nr.                     | U1.3  |
| Versie nummer           | 1.0   |
| Auteur                  | Aman Maksoedan, Chevelly Somopawiro, Edwin Mulder, Hermijn Verbaan, Jamal Arnhem.   |
| Use case naam           | Actor bewerkt content.  |
| Actoren                 | Werknemers.   |
| Doel                    | Het verwijderen van videos.   |
| Triggers                | Een actor klikt op de verwijder knop.   |
| Pre-conditie            | Actor moet aangemeld zijn.  |
| Post-conditie           | Een stuk content is verwijderd.   |
| Hoofdscenario           | <ol style="list-style-type: none"> <li>1. Actor heeft ingelogd.</li> <li>2. Actor selecteert content.</li> <li>3. Systeem checkt of de content bestaat.</li> <li>4. Actor verwijderd de content.</li> <li>5. Systeem voert de wijzigingen door naar de database.</li> <li>6. Systeem geeft bevestiging aan de actor.</li> </ol> |
| Alternatieve scenario's | <p>1-1 Geselecteerde content wordt door 2 actoren tegelijk bewerkt.</p> <p>1-2 Systeem geeft aan dat de geselecteerde content is bewerkt terwijl de actor bezig is.</p>   |
| Excepties               | Geselecteerde content bestaat niet meer   |
| kwaliteiten             | Wanneer een exceptie voorkomt keert het systeem terug naar de laatst bezochte pagina.   |

# Activiteit diagram

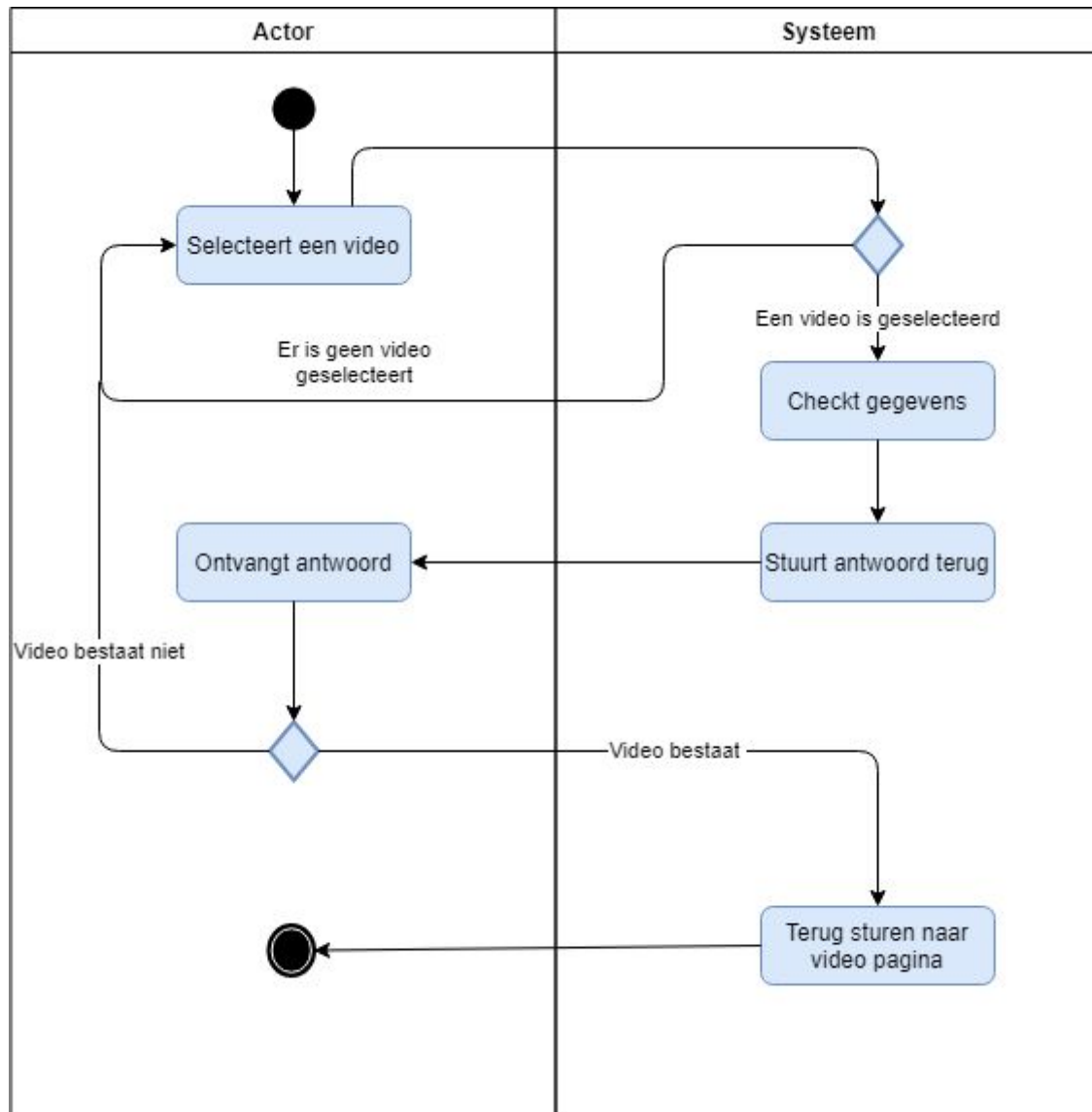
## Actor logt in



## Actor registreert zich



## Werknemer verwijdert video





# (Test) scenario

## Actor logt in

Welke deelpaden

- 1                   =>     1
- 2 & 3 & 4 & 5   =>     2
- 6                   =>     3

## Deelpaden

1. 1
2. 1, 2, 3, 4, 5, 6
3. 1, 2, 3, 4, 5

1. Actor geeft gegevens door.
2. Systeem ontvangt gegevens van actor en vraagt database of gegevens bestaan.
3. Database checkt of gegevens al bestaan.
4. Database stuurt antwoord terug naar systeem.
5. Systeem ontvangt het antwoord van de database.
6. Actor wordt verwezen naar de homepage.

## Testacties

1. Actor geeft gegevens door.
1. Actor geeft gegevens door.
2. Systeem ontvangt gegevens van actor en vraagt database of gegevens bestaan.
3. Database checkt of gegevens al bestaan.
4. Database stuurt antwoord terug naar systeem.
5. Systeem ontvangt het antwoord van de database.
6. Actor wordt verwezen naar de homepage.
1. Actor geeft gegevens door.
2. Systeem ontvangt gegevens van actor en vraagt database of gegevens bestaan.
3. Database checkt of gegevens al bestaan.
4. Database stuurt antwoord terug naar systeem.
5. Systeem ontvangt het antwoord van de database.

## Actor registreert

Welke deelpaden

- 1                   => 1
- 2 & 3 & 4 & 5   => 2
- 6 & 7             => 3

## Deelpaden

1. 1
2. 1, 2, 3, 4, 5, 6, 7
3. 1, 2, 3, 4, 5

1. Actor stuurt registratiegegevens door.
2. Systeem ontvangt gegevens van actor en verstuurd de gegevens naar de database.
3. Database checkt of gegevens al bestaan.
4. Database stuurt antwoord terug naar systeem.
5. Systeem ontvangt het antwoord van de database.
6. Database slaat actor op als nieuwe gebruiker.
7. Actor wordt verwezen naar de inlogpagina.

## Testacties

1. Actor geeft gegevens door.

1. Actor geeft gegevens door.
2. Systeem ontvangt gegevens van actor en verstuurd de gegevens naar de database.
3. Database checkt of gegevens al bestaan.
4. Database stuurt antwoord terug naar systeem.
5. Systeem ontvangt het antwoord van de database.
6. Database slaat actor op als nieuwe gebruiker.
7. Actor wordt verwezen naar de homepagina.

1. Actor geeft gegevens door.
2. Systeem ontvangt gegevens van actor en vraagt database of gegevens bestaan.
3. Database checkt of gegevens al bestaan.
4. Database stuurt antwoord terug naar systeem.
5. Systeem ontvangt het antwoord van de database.

## Actor verwijdert video

Welke deelpaden

- 1                   => 1
- 2 & 3 & 4 & 5   => 2
- 6 & 7            => 3

## Deelpaden

1. 1
2. 1, 2, 3, 4, 5, 6, 7
3. 1, 2, 3, 4, 5

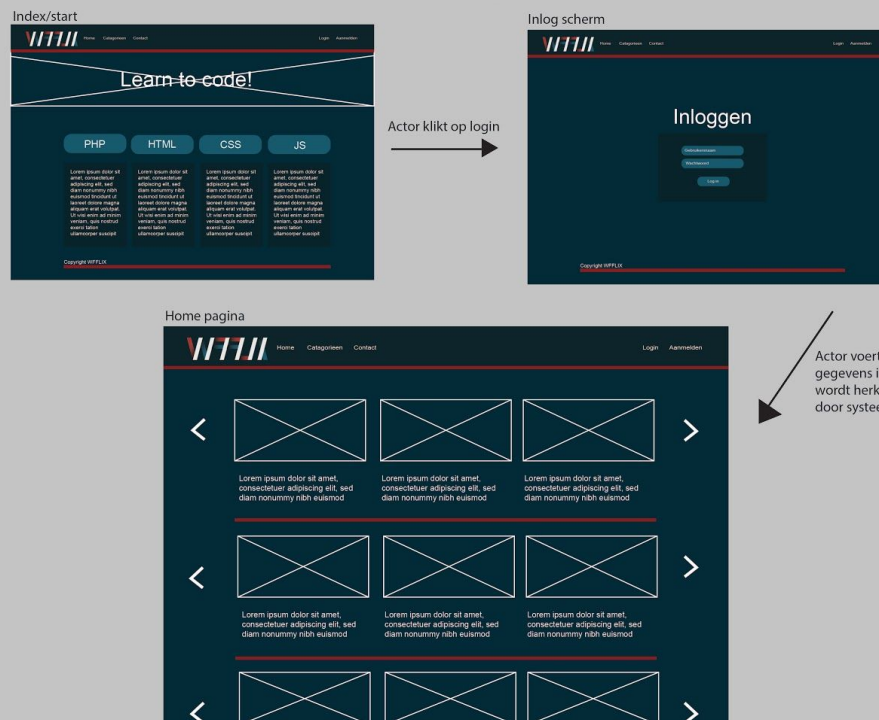
1. Actor selecteert een video.
2. Systeem ontvangt en verstuurd de video gegevens naar de database.
3. Database checkt of de video gegevens al bestaan.
4. Database stuurt antwoord terug naar systeem.
5. Systeem ontvangt het antwoord van de database.
6. Database update de video met behulp van een boolean.
7. Actor wordt verwezen naar de video pagina.

## Testacties

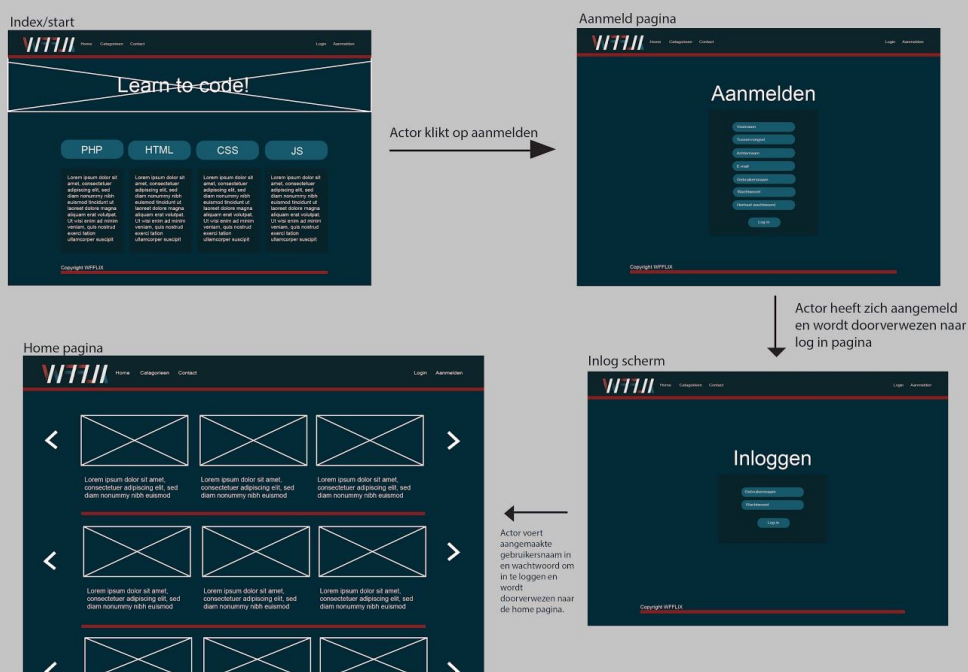
1. Actor selecteert een video.
- 
1. Actor selecteert een video.
  2. Systeem ontvangt en verstuurd de video gegevens naar de database.
  3. Database checkt of de video gegevens al bestaan.
  4. Database stuurt antwoord terug naar systeem.
  5. Systeem ontvangt het antwoord van de database.
  6. Database update de video met behulp van een boolean.
  7. Actor wordt verwezen naar de video pagina.
- 
1. Actor selecteert een video.
  2. Systeem ontvangt en verstuurd de video gegevens naar de database.
  3. Database checkt of de video gegevens al bestaan.
  4. Database stuurt antwoord terug naar systeem.
  5. Systeem ontvangt het antwoord van de database.

# Storyboards

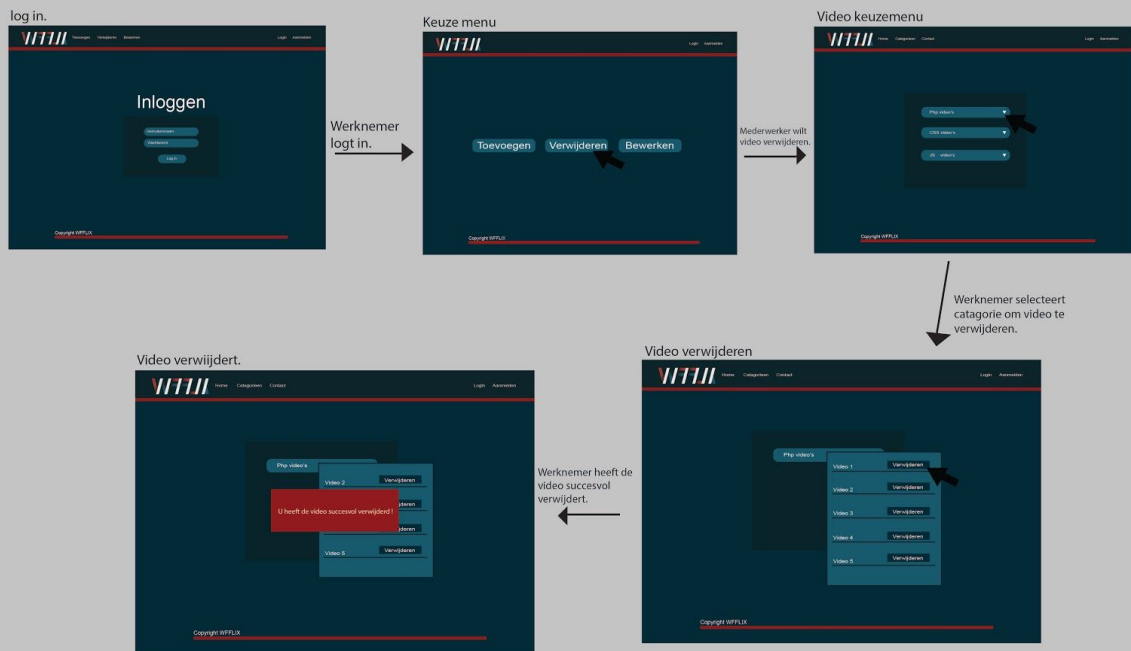
## Actor logt in.



## Actor registreert.



# Actor verwijderd video.



# Pagina lijst

- Index pagina (nog niet ingelogd)
- About pagina
- Aanmeld pagina
- Home pagina (ingelogd)
- Profiel pagina
- Video pagina's
- Medewerker home pagina
- Medewerker video bewerk
- Medewerker video toevoegen
- Medewerker video verwijderen
- Medewerker categorie bewerk
- Medewerker categorie toevoegen
- Medewerker categorie verwijderen

# Database Lijst

## Klant

- **klant\_id (primary key)**
- voornaam
- tussenvoegsel
- achternaam
- e-mail
- gebruikersnaam
- wachtwoord

## Categorie

- **categorie\_id (primary key)**
- naam
- beschrijving

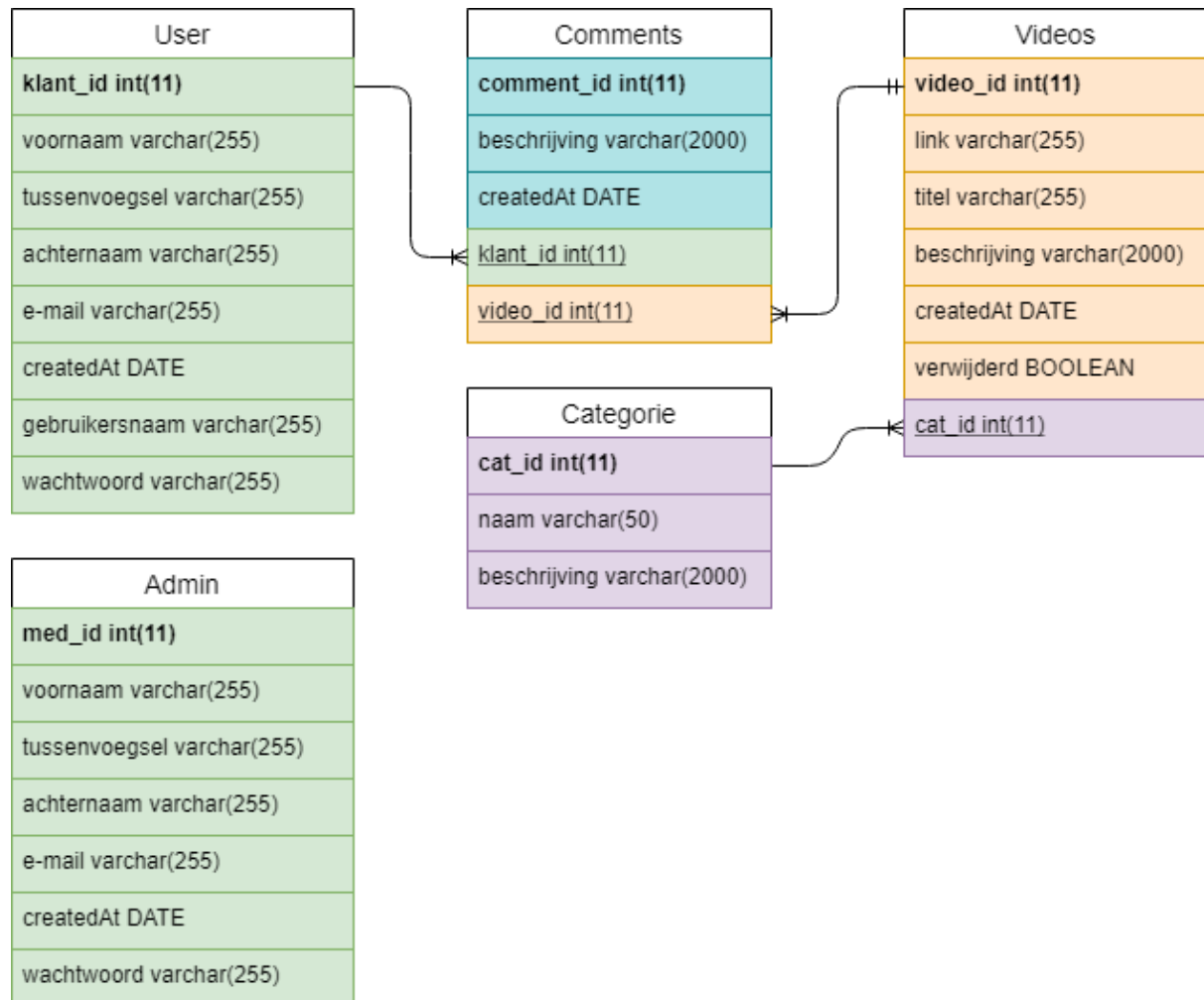
## Video's

- **video\_id (primary key)**
- link
- titel
- beschrijving
- like
- categorie\_id (foreign key)

## Comments

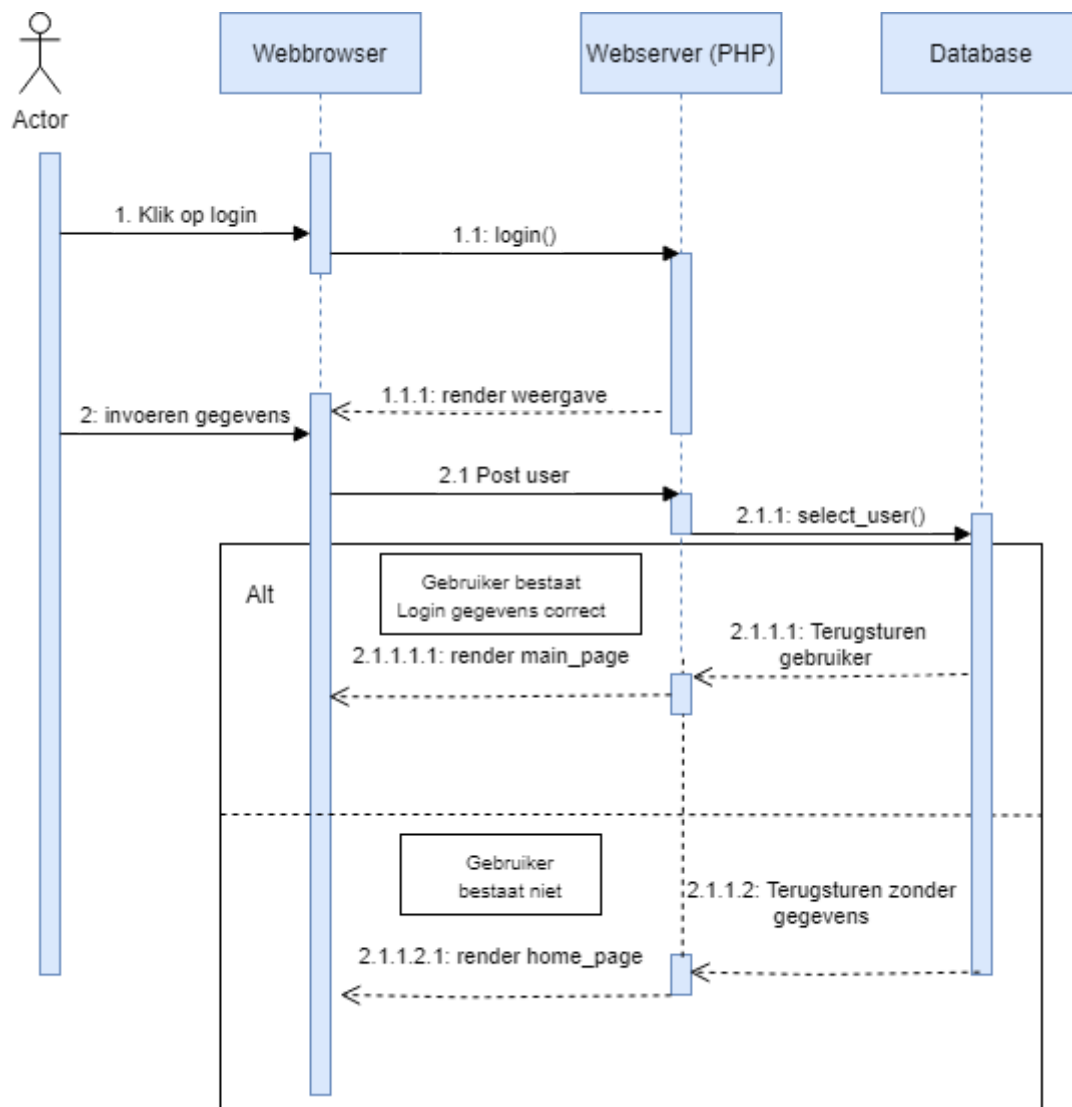
- **comment\_id (primary key)**
- beschrijving
- klant\_id (foreign key)
- video\_id (foreign key)

# ERD



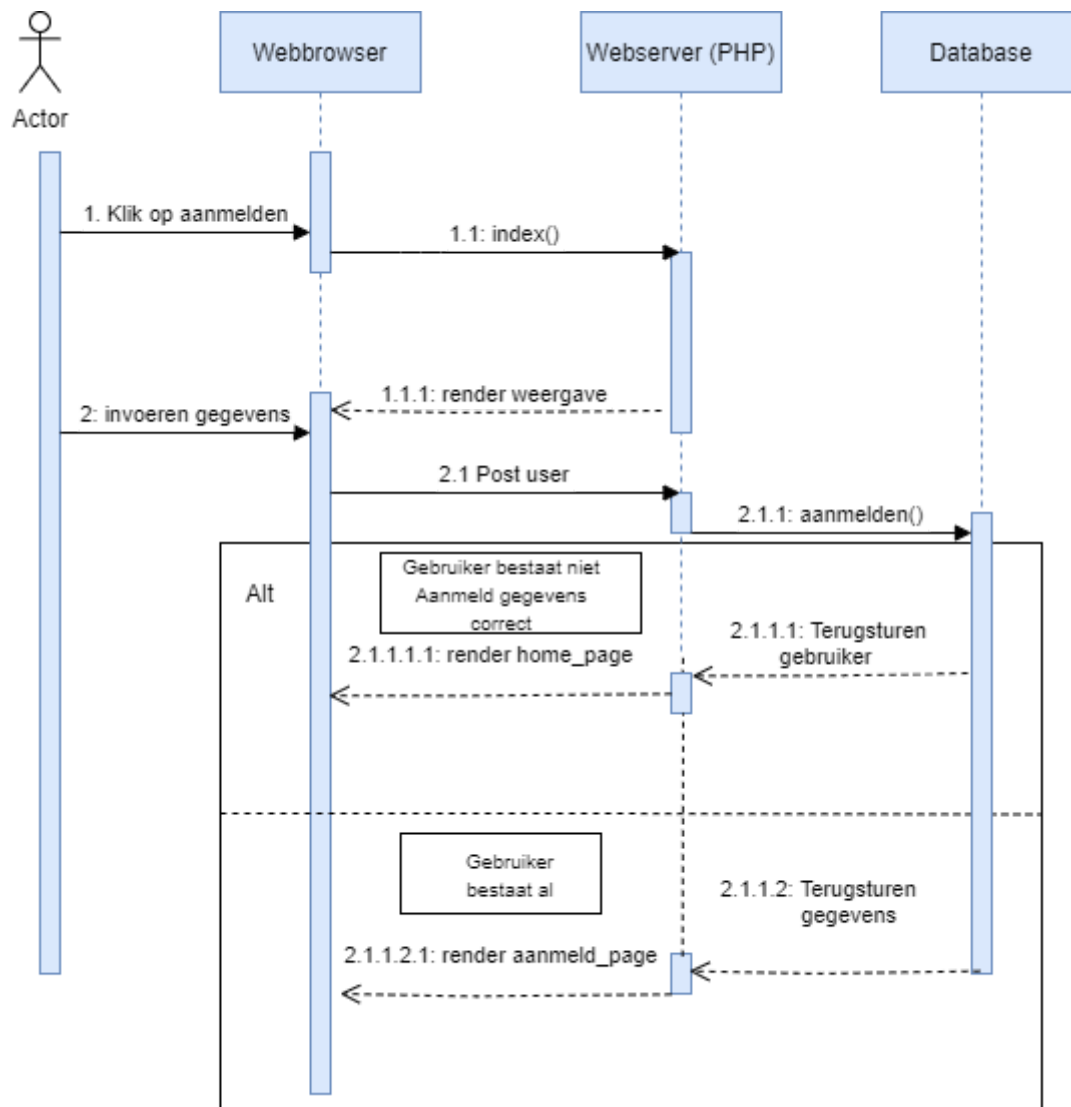
# Sequentiediagrammen

## Inloggen

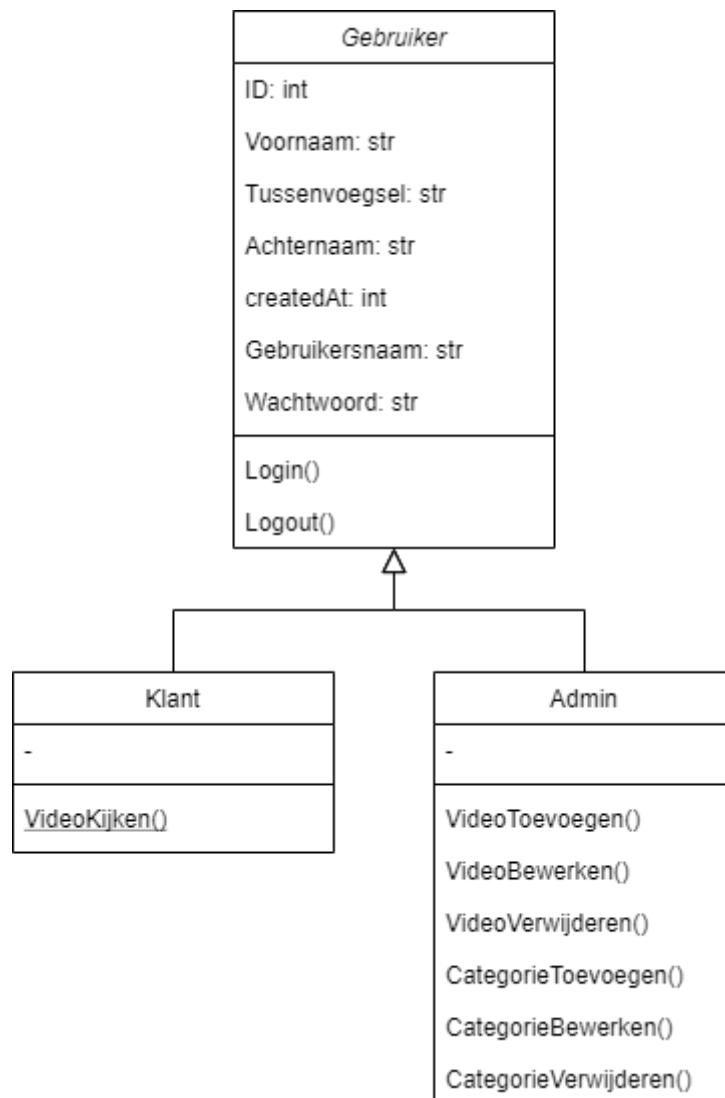




# Aanmelden



# Klassendiagram



# Producten

- Front-end
- Login product
- uitlog product
- Registratie product
- Video view product
- Comment product
- Profielpagina product
- Contact pagina product
- Recent videos product
- Video toevoeg product
- Video bewerk product
- Video verwijder product
- Categorie toevoeg product
- Categorie bewerk product
- Categorie verwijder product

# Kwaliteitscriteria

1. Er wordt gebruikt gemaakt van een gedetailleerde wekelijkse planning per teamlid die bij het opleverdocument (aangevulde projectdefinitie) meegeleverd wordt.
2. Er wordt gebruik gemaakt van een gedeelde Gitlab die bereikbaar is voor de docenten van de opleiding.
3. Het project wordt uitgevoerd met het hele team, dit betekent dat het hele team kennis heeft van de gemaakte producten zoals de source code en het ontwerp.

# Acceptatiecriteria

Het product zal geaccepteerd worden wanneer er is voldaan aan alle kwaliteitscriteria voor de inleverdatum.