

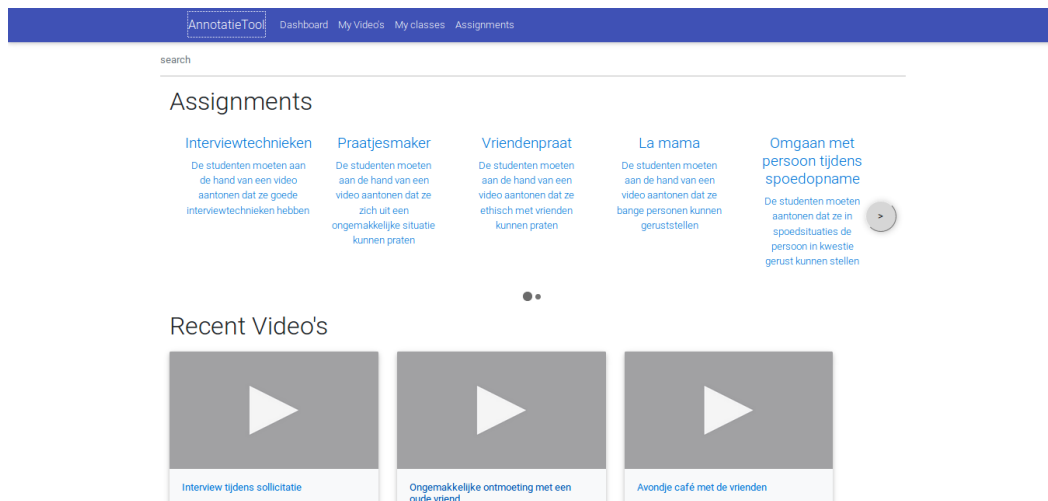
Vakoverschrijdend Eindproject

Annotatietool

Team 1

Eindverslag

Arnout Lenaerts
Bert De Saffel
Emiel Vandenbussche
Glenn Goossens
Nick van Hurck
Ruben Janssens
Timothy Thiecke



Website: <http://annotatietool.tiwi.ugent.be/>

21 mei 2018

Inhoudsopgave

1	Inleiding	3
1.1	Context	3
1.2	Probleemstelling	3
1.3	Doelstelling	3
1.4	Overzicht van dit document	3
2	Gebruikersaspecten	4
2.1	High-Level requirements	4
2.1.1	Functionele requirements	4
2.1.2	Niet-functionele requirements	5
2.1.3	Identificatie	5
2.2	Use case diagram	5
2.3	Use cases	7
2.3.1	Upload video	7
2.3.2	Add video annotation	8
2.3.3	Create class	9
2.3.4	Create assignment	10
2.3.5	Download video data	11
2.4	Backlog	12
3	Systeemarchitectuur	14
3.1	High-level systeem model	14
3.1.1	Webapplicatie	14
3.1.2	MP4 Converter	14
3.2	Klassendiagram	15
3.3	Deploymentdiagram	16
3.4	Databank	16
3.5	Sequentiediagrammen	16
4	Testplan	19
4.1	Unit testen	19
4.1.1	Testsuite Webapplicatie	19
4.1.2	Testsuite MP4 converter	20
4.2	Usability testen	20
4.2.1	Reflectieverslag Usability test sprint 2	20

5	Installatiehandleiding	22
5.1	Hardware en Software	22
5.2	Downloaden repository	23
5.2.1	Structuur repository	24
5.3	Runnen applicatie	25
5.4	Het uitvoeren van testen	25
5.4.1	Testen van de webapplicatie	25
5.4.2	Testen van de MP4 Converter	26

Hoofdstuk 1

Inleiding

1.1 Context

Binnen de UGent loopt een onderwijsinnovatie project waarin video annotatie wordt ingezet om communicatievaardigheden, die centraal staan in diverse opleidingsonderdelen in verschillende opleidingen, te trainen en op een authentieke manier te toetsen. Hiervoor wordt ingezet op peer-assessment, waarbij studenten elkaar feedback geven op gefilmde situaties waarin de student communicatievaardigheden toepast.

1.2 Probleemstelling

Er bestaat al commerciële video-annotatiesoftware, maar deze hebben echter een duur prijskaartje en zijn bovendien complex om te gebruiken. Ondertussen heeft dit project wel een gratis alternatief gevonden, CommentBubble ¹. CommentBubble maakt gebruik van YouTube om video's te hosten. Door het feit dat het videomateriaal enerzijds vertrouwelijke informatie bevat, en anderzijds om voldoende privacy te garanderen, gaat de voorkeur uit naar een annotatietool waarbij alle data intern opgeslagen word binnen de UGent.

1.3 Doelstelling

De video-annotatietool die ontworpen moet worden zal gebruikt worden door zowel docenten als studenten. Om installaties van de software te vermijden wordt er gekozen om deze tool te implementeren als een webapplicatie.

1.4 Overzicht van dit document

In hoofdstuk 1 werd de opdracht al toegelicht. In hoofdstuk 2 worden de noodzakelijke features besproken. In hoofdstuk 3 wordt de onderliggende structuur van het project besproken. In hoofdstuk 4 wordt besproken hoe het project uitgetest zal worden. In hoofdstuk 5 wordt een overzicht gegeven hoe de software te installeren op een toestel.

¹<https://commentbubble.com/>

Hoofdstuk 2

Gebruikersaspecten

2.1 High-Level requirements

De applicatie kan gebruikt worden door twee type actoren. Een normale gebruiker (student) en een docent. De docent is een speciaal geval van een normale gebruiker met extra rechten, maar heeft ook dezelfde rechten als een normale gebruiker.

2.1.1 Functionele requirements

Login

- De gebruiker kan zich inloggen via het UGent CAS-systeem of via een dummy systeem dat dit nabootst.

Dashboard

- De gebruiker kan een video selecteren op het dashboard.
- De gebruiker kan een videobestand uploaden in de applicatie
- De docent kan gebruikers koppelen aan een video.
- De docent kan een deadline instellen van een video.
- De docent kan een label toevoegen aan een video.
- De docent kan groepjes van gebruikers aanmaken.
- De docent kan de video's filteren per thema of categorie.
- De docent kan quick comments beheren.
- De docent kan quick comments koppelen aan een video.
- De docent kan het niveau van toegankelijkheid voor de annotaties van een bepaalde video instellen.
- De docent kan voor een video of meerdere video's de annotaties exporteren.
- De docent kan een .csv file uploaden.

Video interactief

- De gebruiker kan op een specifiek moment een annotatie plaatsen.
- De gebruiker kan op een annotatie klikken om naar dat punt in de video te springen waarop de annotatie geplaatst is.
- De gebruiker kan een quick comment toevoegen op het huidige tijdstip van de video.
- De docent kan een video afspelen of pauzeren.
- De docent kan zoeken tussen alle annotaties van een bepaalde video.

2.1.2 Niet-functionele requirements

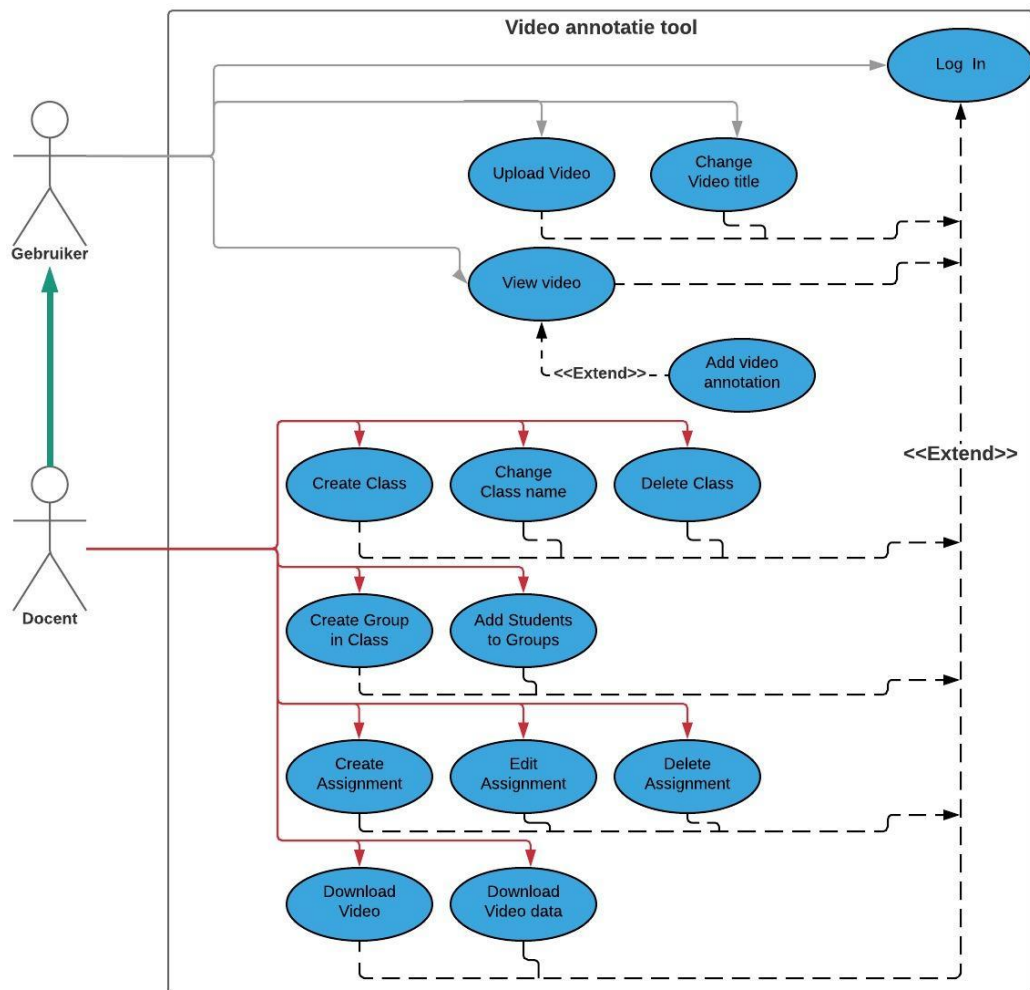
- De website moet gebruiksvriendelijk zijn. De website moet eenvoudig te begrijpen zijn voor onervaren computergebruikers.
- Aangezien dat de video's vertrouwelijke inhoud kunnen bevatten, moet hiervoor de nodige beveiliging voorzien worden.
- Tijdens het typen van een annotatie zou de informatie voortdurend automatisch opgeslagen moeten worden zodat er geen werk verloren gaat.

2.1.3 Identificatie

Gebruikers worden herkend aan de hand van hun UGent credentials. De applicatie kan alleen maar gebruikt worden door personen die verbonden zijn aan de UGent.

2.2 Use case diagram

Figuur 2.1 toont het use case diagram. Het systeem kent slechts 2 actoren: een normale gebruiker en een docent. Aangezien er een poging werd gedaan om het UGent CAS loginsysteem te gebruiken, bestaat er geen actie 'registreren'.



Figuur 2.1: Het use case diagram

2.3 Use cases

2.3.1 Upload video

Use Case 1 Upload video	
<i>Actor:</i>	Gebruikers en docenten
<i>Trigger:</i>	De gebruiker selecteert de optie om een video te uploaden
<i>Precondities:</i> <ul style="list-style-type: none">- De gebruiker is ingelogd (Log in use case)- Er is een opdracht aangemaakt door de docent (Create assignment use case)	
<i>Postcondities:</i> <ul style="list-style-type: none">- De video is geüpload en zichtbaar voor de gebruiker zelf en voor de gebruikers met wie de video gedeeld is	
<i>Basis flow:</i> <ol style="list-style-type: none">1. Het systeem toont het scherm waarop een video kan geüpload worden2. De gebruiker selecteert lokaal een video file3. Het systeem controleert of het om een video file gaat4. De gebruiker geeft de video een titel5. De gebruiker selecteert de bijhorende opdracht6. De gebruiker bevestigt de upload7. Het systeem slaat de video en bijhorende info op8. Het systeem toont de geüploade video	
<i>Alternatieve flow:</i> <ol style="list-style-type: none">3a. De gebruiker selecteerde een file dat geen video format is<ol style="list-style-type: none">3a.1 Het systeem toont een foutboodschap en vraagt om de juiste file te selecteren3a.2 Terug naar stap 2	

2.3.2 Add video annotation

Use Case 2 Add video annotation	
<i>Actor:</i>	Gebruikers en docenten
<i>Trigger:</i>	De gebruiker selecteert een video
<i>Preconditions:</i> <ul style="list-style-type: none">- De gebruiker is ingelogd (Log in use case)- De gebruiker is een video aan het bekijken (View video use case)	
<i>Postconditions:</i> <ul style="list-style-type: none">- Een annotatie is opgeslagen voor een video	
<i>Basis flow:</i> <ol style="list-style-type: none">1. De gebruiker pauzeert de video stream2. De gebruiker geeft de tekst voor de annotatie in3. De gebruiker bevestigt de annotatie4. Het systeem slaat de annotatie met bijhorende timestamp op en geeft deze weer	
<i>Alternatieve flow:</i> <ol style="list-style-type: none">1a. De video wordt niet gepauzeerd<ol style="list-style-type: none">1a.1 De gebruiker selecteert een quick comment annotatie1a.2. Ga verder met stap 42a. Het gaat om een Drop down menu annotatie<ol style="list-style-type: none">2a.1 De gebruiker selecteert de juist optie in de dropdown2a.2 De gebruiker geeft bijhorende tekst in voor de annotatie2a.3 Ga verder met stap 3	

2.3.3 Create class

Use Case 3 Create class	
<i>Actor:</i>	Docenten
<i>Trigger:</i>	De gebruiker selecteert de optie om een klas aan te maken
<i>Precondities:</i> <ul style="list-style-type: none">- De gebruiker is ingelogd (Log in use case)	
<i>Postcondities:</i> <ul style="list-style-type: none">- Een nieuwe klas is aangemaakt met bijhorende gebruikers	
<i>Basis flow:</i> <ol style="list-style-type: none">1. Het systeem toont een pop-up waarin de gebruiker een klas kan aanmaken2. De gebruiker selecteert lokaal een .csv file met gebruikers3. Het systeem controleert of het een .csv file is en of deze correcte formattering gebruikt4. De gebruiker geeft de klas een naam5. De gebruiker bevestigt het aanmaken6. Het systeem slaat de klas en bijhorende gebruikers op	
<i>Alternatieve flow:</i> <ol style="list-style-type: none">3a. Het .csv bestand heeft geen geldig formaat<ol style="list-style-type: none">3a.1 Het systeem toont een gepaste melding3a.2 Keer terug naar stap 2	

2.3.4 Create assignment

Use Case 6 Create assignment	
<i>Actor:</i>	Docenten
<i>Trigger:</i>	De gebruiker selecteert de optie om een opdracht aan te maken
<i>Preconditions:</i> <ul style="list-style-type: none">- De gebruiker is ingelogd (Log in use case)	
<i>Postconditions:</i> <ul style="list-style-type: none">- Een nieuwe opdracht is aangemaakt	
<i>Basis flow:</i> <ol style="list-style-type: none">1. Het systeem toont een pop-up waarin de gebruiker een opdracht kan aanmaken2. De gebruiker geeft de opdracht een naam3. De gebruiker activeert de gewenste annotaties en vult de juiste bijhorende waarden in4. De gebruiker geeft aan of de annotaties privé zijn of niet5. De gebruiker selecteert voor welke klas of groep de opdracht is6. De gebruiker geeft aan wie de uploads mag doen (docent of gebruiker)7. De gebruiker geeft mogelijk een deadline mee8. Het systeem controleert de data9. Het systeem maakt de opdracht aan en slaat de data op10. Het systeem toont een overzicht van alle opdrachten	
<i>Alternatieve flow:</i> <ol style="list-style-type: none">8a. De data is niet volledig<ol style="list-style-type: none">8a.1 Het systeem toont een foutboodschap8a.2 De gebruiker overloopt opnieuw vanaf stap 2	

2.3.5 Download video data

Use Case 8	Download video data
Actor:	Docenten
Trigger:	De gebruiker selecteert de optie om de video data te downloaden
<i>Preconditions:</i> <ul style="list-style-type: none">- De gebruiker is ingelogd (Log in use case)- De gebruiker is een video aan het bekijken (View video use case)	
<i>Postconditions:</i> <ul style="list-style-type: none">- De gebruiker heeft een lokale versie van de verschillende annotaties van de video	
<i>Basis flow:</i> <ol style="list-style-type: none">1. Het systeem pauzeert de video2. Het systeem vraagt of de gebruiker zeker is3. De gebruiker bevestigt4. Het systeem genereert de data file5. Het systeem start een download van die file naar de computer van de gebruiker	
<i>Alternatieve flow:</i> /	

2.4 Backlog

De volgende lijst stelt de product backlog voor, inclusief volgende attributen: extra uitleg, de prioriteit, de complexiteit, in welk sprint deze feature afgewerkt is en eventueel extra opmerkingen.

- Als een gebruiker wil ik een video kunnen uploaden (**Sprint 1**)
Uitleg Het uploaden van een video is één van de basisfunctionaliteiten.
Prioriteit: 5/5 **Complexiteit:** 3/5
- Als een gebruiker wil ik een video kunnen bekijken (**Sprint 1**)
Uitleg: Het bekijken van een video is één van de basisfunctionaliteiten.
Prioriteit: 5/5 **Complexiteit:** 2/5
- Als een gebruiker wil ik annotaties kunnen plaatsen op een video (**Sprint 2**)
Uitleg: Het toevoegen van annotaties is één van de basisfunctionaliteiten.
Prioriteit: 5/5 **Complexiteit:** 3/5
- Als een gebruiker wil ik quick annotations toevoegen aan een video (**Sprint 2**)
Uitleg: Het toevoegen van quick annotations is een uitbreiding op de normale annotaties. **Prioriteit:** 4/5 **Complexiteit:** 1/5
- Als een gebruiker wil ik naar een punt in de video springen waar een annotatie staat zodat ik de annotatie kan bekijken (**Sprint 2**)
Uitleg: Een gebruiker wil een eenvoudige manier om het filmpje af te spelen op de plaats waar een annotatie geplaatst is
Prioriteit: 3/5 **Complexiteit:** 1/5
- Als een gebruiker wil ik een dashboard zodat ik een overzicht heb van mijn taken en video's (**Sprint 2**)
Uitleg: Het dashboard is de eerste pagina dat getoond wordt wanneer een gebruiker inlogt. Deze pagina moet direct een overzicht bieden van de nog te maken taken, en video's die kunnen bekeken worden
Prioriteit: 4/5 **Complexiteit:** 3/5
- Als een docent wil ik klasgroepen importeren uit een .csv bestand zodat ik een overzicht heb van mijn studenten (**Sprint 2**)

Uitleg: Aangezien het UGent CAS systeem uiteindelijk niet op de applicatie is aangesloten, kunnen de klasgroepen niet rechtstreeks vanuit dit systeem gehaald worden. De applicatie laat toe een .csv bestand up te loaden die een .csv export van Minerva nabootst zodat hieruit klasgroepen kunnen gemaakt worden.

Prioriteit: 3/5 **Complexiteit:** 4/5

- Als een gebruiker wil ik taken bekijken zodat ik daar een video voor kan uploaden (**Sprint 3**)

Uitleg: De gebruiker moet een overzicht hebben van zijn taken om te weten waar hij een video kan uploaden

Prioriteit: 3/5 **Complexiteit:** 3/5

- Als een gebruiker wil ik kunnen inloggen zodat ik gebruik kan maken van de applicatie (**Sprint 3**)

Uitleg: Een geldig UGent email-adres hebben is verplicht om gebruik te kunnen maken van de applicatie. Indien de gebruiker een geldig UGent email-adres heeft kan de gebruiker inloggen op de applicatie.

Prioriteit: 5/5 **Complexiteit:** 4/5

Opmerking: Deze feature werd ook al eens in sprint 1 opgenomen, maar er was onvoldoende kennis over het UGent Cas systeem om de implementatie te starten

Hoofdstuk 3

Systeemarchitectuur

3.1 High-level systeem model

Het project bestaat uit twee delen. Het eerste deel is de webapplicatie die gebruikt wordt door de actor. Het tweede deel is de MP4 Converter. Dit is een aparte module dat video's van allerlei formaten omvormt tot het standaardformaat dat gebruikt wordt in de applicatie, namelijk .mp4.

3.1.1 Webapplicatie

De webapplicatie wordt geprogrammeerd met gebruik van de MEAN ¹ stack. Dit is een verzameling van technologieën die samen gebruikt worden om webapplicaties te ontwikkelen. De MEAN stack bevat volgende onderdelen:

- **MongoDb:** Dit is een open source NoSQL databank dat gebruik maakt van JSON documenten om gegevens te bewaren.
- **Express:** Dit is een implementatie van Node dat geschikt is voor het opstellen van webserver.
- **Angular:** Dit is een client-side javascript framework dat geschikt is voor het creëren van dynamische webapplicaties. De interactie tussen het DOM en javascript wordt vereenvoudigt met Angular.
- **Node.js:** Dit is een javascript runtime omgeving met als onderliggende basis de javascript engine van Chrome. Een gevolg van Node is het node packaging systeem (npm) dat toelaat om modules toe te voegen aan een applicatie met slechts één lijn code.

Hoe deze componenten met elkaar samenwerken kan bekeken worden op figuur 3.2.

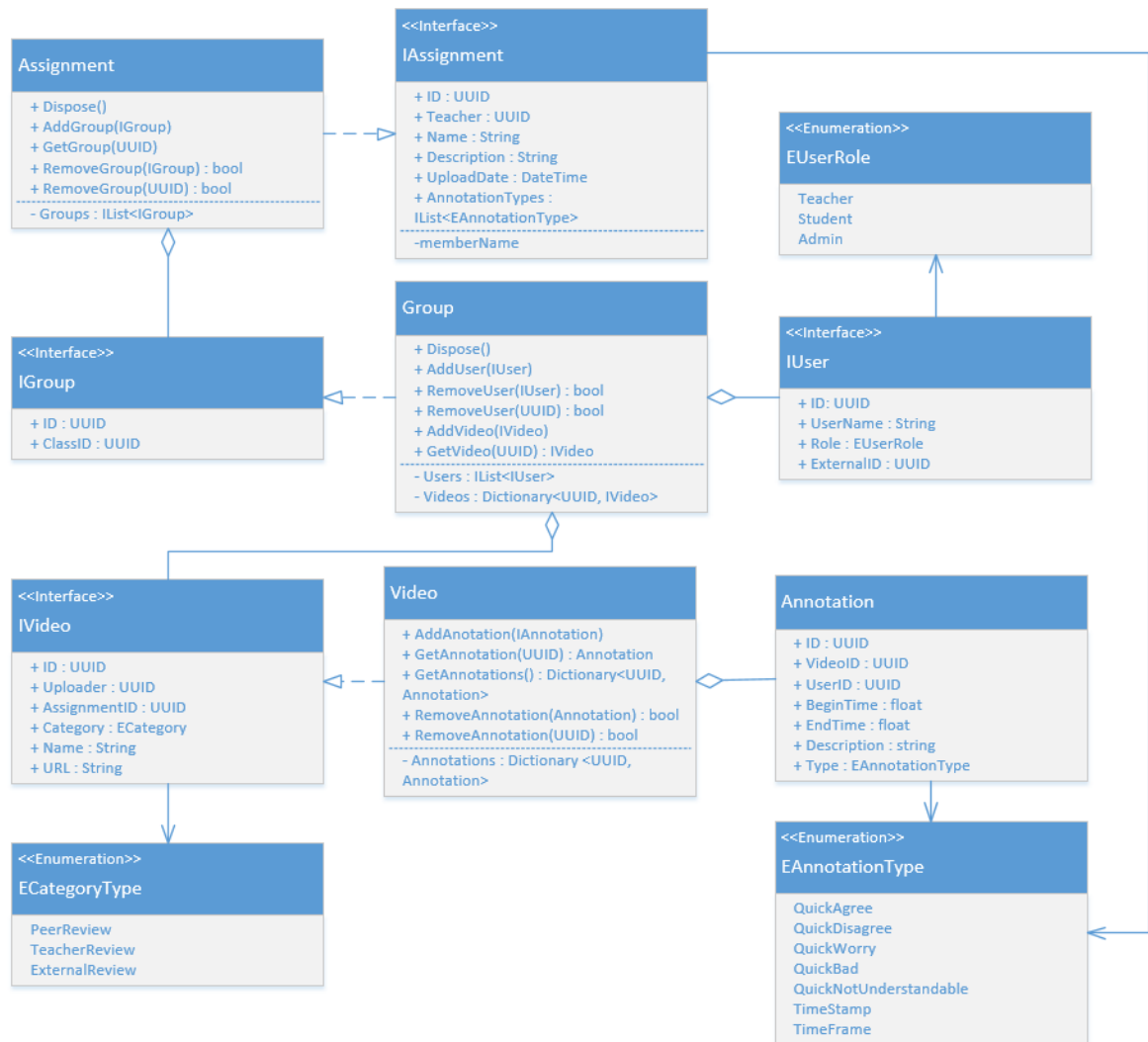
3.1.2 MP4 Converter

De MP4 Converter is geprogrammeerd in Java en wordt gebruikt door de webapplicatie om videobestanden van verschillende formaten om te zetten naar het mp4 formaat.

¹<http://mean.io/>

Als onderliggende basis wordt gebruik gemaakt van `ffmpeg`². Dit is een handige tool dat toelaat om video's of geluid op te nemen en te converteren. Deze applicatie maakt enkel gebruik van het converteren van een video. De MP4 Converter is met andere woorden een Java wrapper voor `ffmpeg` met slechts beperkte functionaliteit.

3.2 Klassendiagram

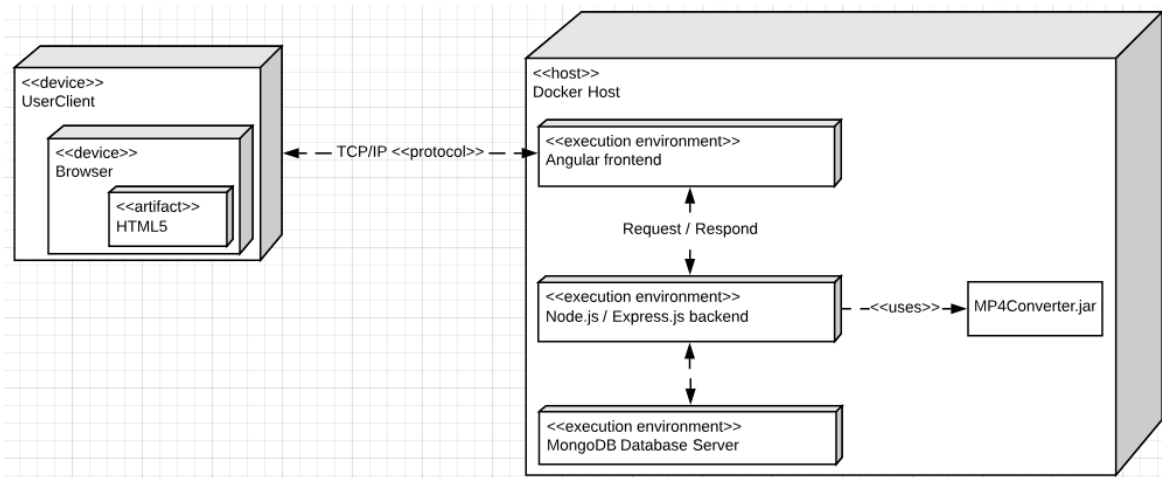


Figuur 3.1: Het klassendiagram

In figuur 3.1 is een overzicht te vinden van het klassendiagram. Dit diagram komt overeen met de implementaties van de verschillende database modelklassen en de manier hoe ze in relatie tot elkaar staan in de backend.

²<https://ffmpeg.org/>

3.3 Deploymentdiagram



Figuur 3.2: Het deploymentdiagram

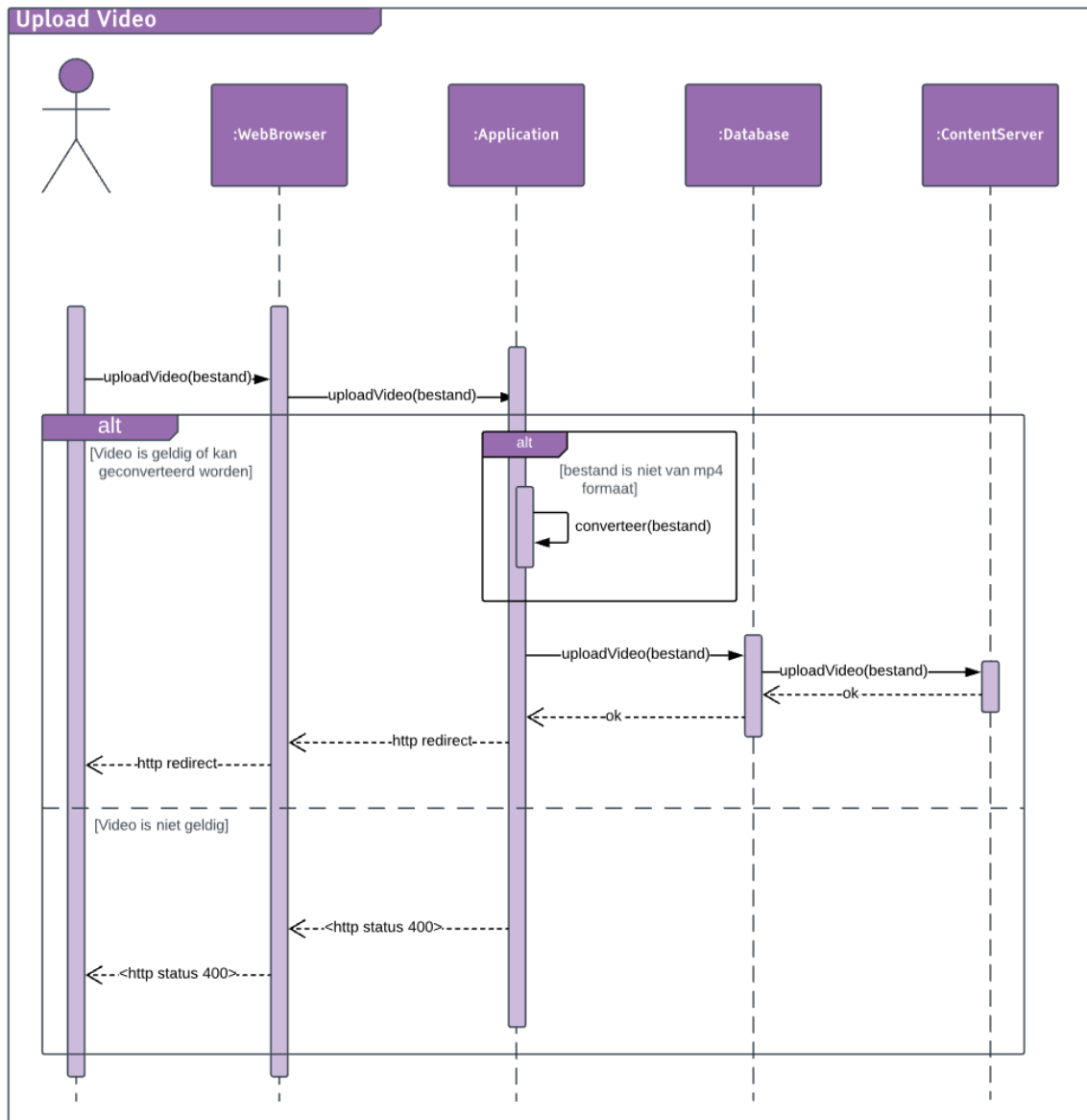
In figuur 3.2 is een overzicht te vinden van het deploymentdiagram. De applicatie draait in drie aparte docker containers op de server: de Node.js container (backend), de Angular frontend container en de MongoDB database container.

3.4 Databank

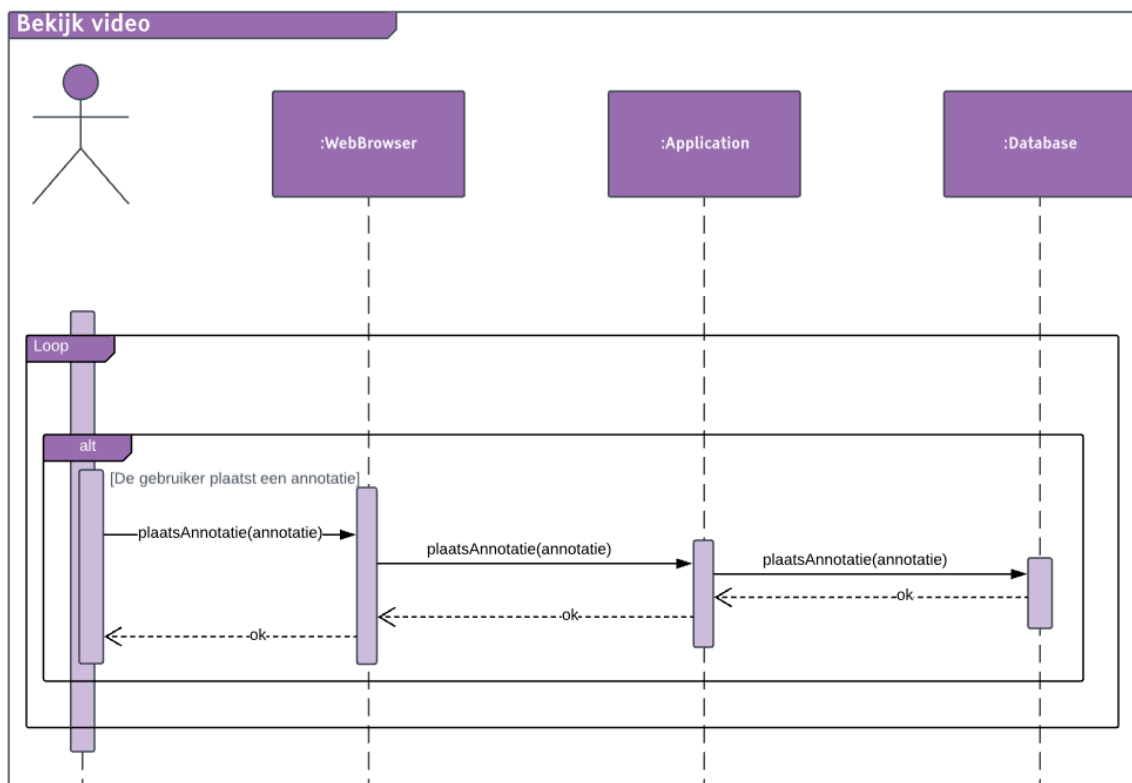
De databank is opgesteld in MongoDB. Dit is een NoSQL omgeving waardoor het moeilijk is om een databankdiagram op te stellen. Om een overzicht te hebben van de relaties tussen de verschillende klassen zoals ze zijn geïmplementeerd in de code kan gekeken worden naar het klassendiagram op figuur 3.1

3.5 Sequentiediagrammen

Figuren 3.3 en 3.4 zijn sequentiediagrammen van twee belangrijke events die kunnen voorkomen in de applicatie. De verschillende events maken gebruik van synchrone gebruikersinput die asynchroon wordt behandeld.



Figuur 3.3: Sequentiediagram van upload video



Figuur 3.4: Sequentiediagram van show video

Hoofdstuk 4

Testplan

De applicatie wordt op twee verschillende manieren getest om fouten of onduidelijkheden te voorkomen. De volgende onderdelen bespreken de verschillende soorten testen en wanneer deze uitgevoerd worden.

4.1 Unit testen

Unit testen dienen om specifieke functionaliteiten van modules te testen zonder afhankelijk te zijn van andere implementaties door gebruik te maken van mock-objecten. Zulke objecten kunnen bijvoorbeeld de database of langdurige operaties vervangen.

Dit project heeft twee soorten unit testen. Er bestaan unit testen voor zowel de webapplicatie als het Java-stuk.

De unit-testen zijn geschreven met Jasmine. Meer specifiek de node versie hiervan, `jasmine-node`. Voor volgende functionaliteiten zijn er testen geschreven:

- Inloggen op de applicatie
- Plaatsen van annotaties
- Aanmaken van groepen op basis van een .csv bestand

Aangezien een videobestand meerdere formaten kan aannemen, zijn er ook unit testen die de conversie van zo een bestand test. In de applicatie zijn enkel .mp4 bestanden toegelaten.

4.1.1 Testsuite Webapplicatie

De volgende scenario's worden getest op de webapplicatie.

Met betrekking tot het plaatsen van annotaties

- Er wordt een annotatie aan een video toegevoegd indien de annotatie geldig is.
- Er wordt een fout gegeven wanneer een annotatie ongeldig is. Een annotatie is ongeldig wanneer:

1. de begintijd een negatieve waarde bevat.
2. wanneer de eindtijd eerder dan de begintijd komt.

Met betrekking tot het inloggen in de applicatie

- De gebruiker kan inloggen indien een correct UGent email-adres en paswoord gebruikt wordt.
- Er wordt een fout gegeven indien een paswoord of een gebruikersnaam incorrect is.

Met betrekking tot het aanmaken van groepen

- Er wordt een klasgroep aangemaakt indien een correct .csv bestand meegegeven wordt. Een correct .csv bestand heeft het volgende formaat:
`Officiële code;Voornaam;Familiennaam;UGent-email;Rol;Subgroep`

4.1.2 Testsuite MP4 converter

De volgende scenario's worden getest op de MP4 converter.

- Er wordt een correcte conversie van een toegelaten videobestand naar MP4 formaat uitgevoerd.
- Er wordt een fout weergegeven indien een bestand geen extensie heeft.
- Er wordt een fout gegeven indien een bestand een niet toegelaten extensie heeft.
- Er wordt een fout gegeven indien het bestand niet bestaat

Omdat de broncode nodig is om de testen uit te voeren en aangezien dit pas besproken wordt in 5.2 hoe dit moet gebeuren, wordt pas in sectie 5.4 besproken hoe de testen kunnen uitgevoerd worden.

4.2 Usability testen

Uiteindelijk moet de applicatie bruikbaar zijn voor personen die het zullen gebruiken. Er wordt aan het einde van een sprint aan een onafhankelijke persoon gevraagd om de functionaliteiten die geïmplementeerd zijn te testen. Wanneer er onduidelijkheden zijn wordt er samen met deze persoon naar een alternatief gezocht. Er wordt een kort verslag gemaakt met commentaar van de gebruiker.

4.2.1 Reflectieverslag Usability test sprint 2

De webapplicatie die online staat na sprint 2 heeft volgende beperkte functionaliteiten:

- Overzicht van video's en assignments
- Video uploaden

- Video bekijken
- Annotatie plaatsen
- Groepen bekijken
- Assignments bekijken

Testpersonen, Yana De Brouwer, Herman Goossens & Marijke Van Wayenberg, hebben deze functionaliteiten uitgetest zonder veel extra uitleg en op basis van de test-sessie enkele bevindingen aangebracht.

- Het is niet duidelijk bij het dashboard of de assignments en video's voor jou bedoelt zijn.
- Een video afspelen werkt zoals het zou moeten en is makkelijk te bereiken vanuit het dashboard.
- Er is geen melding of een video uploaden al dan niet gelukt of bezig is. Je wordt gewoon herleid naar de hoofdpagina.
- Het overzicht van de assignments is bombastisch en dus niet overzichtelijk, idem groups.
- De styling van de applicatie kan beter.

Deze bevindingen worden meegenomen naar de laatste sprint en worden gebruikt om verbeteringen aan te brengen in de applicatie.

Hoofdstuk 5

Installatiehandleiding

Dit hoofdstuk bespreekt hoe het software-pakket lokaal op een toestel kan draaien. De volgende logingegevens kunnen gebruikt worden om te authenticeren op de applicatie:

username	passwoord	rol
docent	test	docent
student	test	student

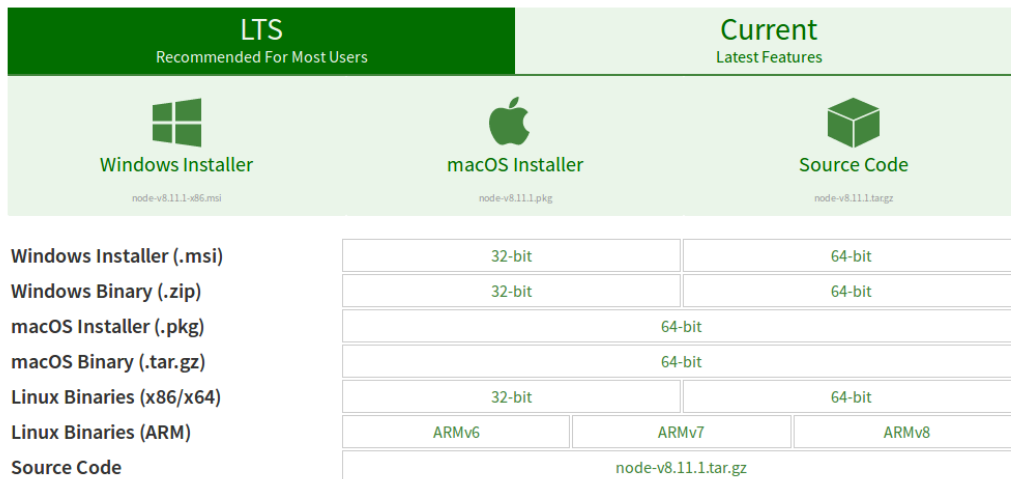
5.1 Hardware en Software

De applicatie kan op volgende besturingssystemen draaien:

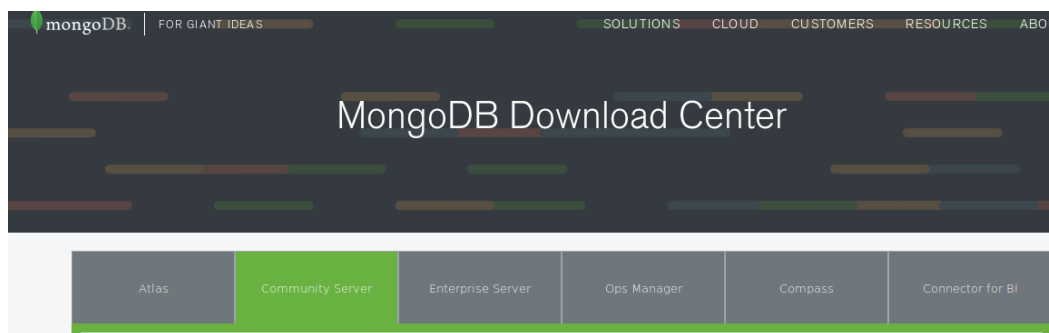
- Linux (32 en 64 bit)
- macOS (64 bit)
- Windows 7 of hoger (32 en 64 bit)

Verder moeten ook nog volgende softwarepakketten geïnstalleerd worden.

- Node.js (minimum v8.9.4): <https://nodejs.org/en/download/>. Opteer voor het *.msi* bestand voor Windows en het *.pkg* bestand voor macOS zoals te zien op figuur 5.1 om de installatieprocedure zo eenvoudig mogelijk uit te voeren. Controleer of dit juist geïnstalleerd is door het commando `node -v` uit te voeren in een terminal of console.
- MongoDB (minimum v2.6.10): <https://www.mongodb.com/download-center> en klik op **Community Server** zoals aangegeven in figuur 5.2. Klik vervolgens op het gewenste besturingssysteem en klik dan op download zoals te zien op figuur 5.3. Controleer of dit juist geïnstalleerd is door het commando `mongo --version` uit te voeren in een terminal of console.
- Java 8 (minimum *JDK 8u171*): <https://www.java.com/en/download/win10.jsp> : Klik op de download knop en volg de installatieprocedure. Controleer of dit juist geïnstalleerd is door het commando `java --version` uit te voeren in een terminal of console.



Figuur 5.1: Node pagina - download



Figuur 5.2: Mongo pagina - Community Server

- **ffmpeg** : Dit bevat enkel builds voor Windows en macOS. Een .bz2 bestand kan teruggevonden worden op <https://ffmpeg.org/download.html>. Om te controleren of dit programma correct is geïnstalleerd kan het commando **ffmpeg** uitgevoerd worden in de console of terminal.

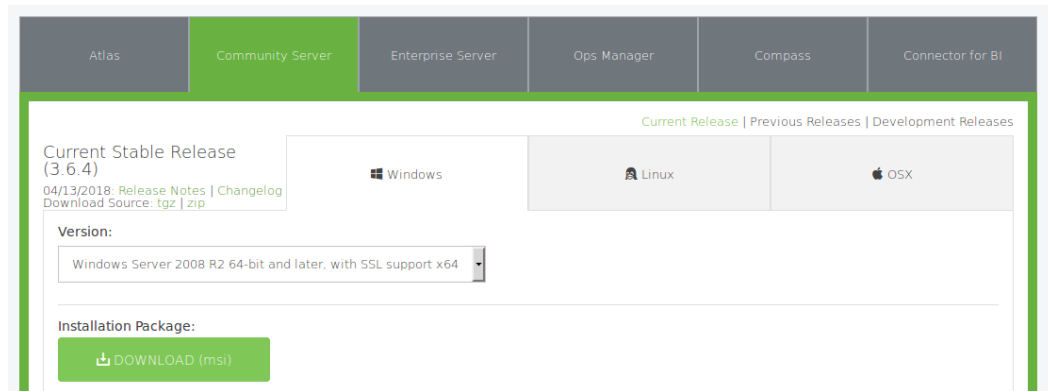
5.2 Downloaden repository

Nadat al de benodigde software geïnstalleerd is moet de broncode op het toestel geplaatst worden. Dit kan op twee manieren:

1. Indien *git* geïnstalleerd is op het toestel kan het commando

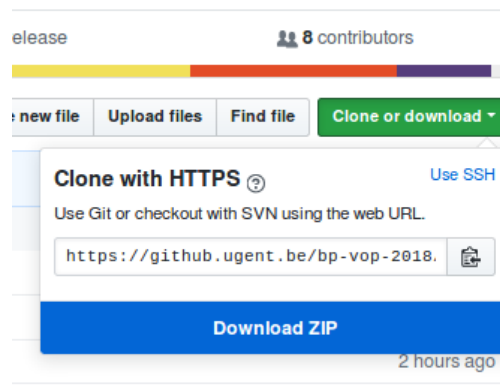
```
git clone https://github.ugent.be/bp-vop-2018/annotatietool-01.git
```

gebruikt worden. Dit zal een folder aanmaken met de naam **annotatietool-01** waarin de broncode terug te vinden is.
2. Indien het toestel geen *git* heeft, kan de repository nog altijd manueel gedownload worden via <https://github.ugent.be/bp-vop-2018/annotatietool-01>. Klik op **Clone or download** en dan **Download ZIP** zoals te zien op figuur 5.4. Dit zal een ZIP bestand downloaden waarin de broncode terug te vinden is.



Figuur 5.3: Mongo pagina - download

Plaats de inhoud van het ZIP bestand (folder genaamd **annotatietool-01**) op het toestel.



Figuur 5.4: Github pagina : download

5.2.1 Structuur repository

Al de broncode van het project bevindt zich in **annotatietool-01**. De belangrijkste items in deze folder zijn:

- **app.js** : Dit bestand is het startpunt van de applicatie.
- **src** : Deze folder bevat alle code in verband met de frontend.
- **backend** : Deze folder bevat all code met betrekking tot de backend.
- **spec** : Deze folder bevat integratietesten voor de webapplicatie (zie 5.4.1 om de testen uit te voeren).
- **java** : Deze folder bevat het java-gedeelte van het project (de MP4 converter). De unit testen zitten ook in deze folder (zie 5.4.2 om de testen uit te voeren).

5.3 Runnen applicatie

Er bestaan twee mogelijkheden om de applicatie te draaien op een lokaal toestel.

Docker

Hiervoor moet Docker geïnstalleerd worden op het toestel. Navigeer naar <https://store.docker.com/search?type=edition&offering=community> om naar een lijst te gaan van de community edition van Docker per besturingssysteem. Kies de versie dat past bij het besturingssysteem van het toestel en voer de installatieprocedure uit. Indien Docker geïnstalleerd is op het toestel kan het commando `./buildandrun.sh` uitgevoerd worden in de root directory van het project (annotatietool-01). Als de docker frontend container in productie gedeployed wordt, dan moet de `--disable-host-check` flag verwijderd worden uit de Dockerfile, alsook moet de host gespecificeerd worden voor de productieomgeving. De webapplicatie kan vanaf nu bediend worden op het lokaal adres `localhost:9000` via een webbrowser.

Zonder docker

Indien geen gebruik gemaakt wordt van docker moet de technologiestack zelfstandig geïnstalleerd worden. Voer volgende stappen uit.

1. Navigeer naar de `project` folder (`annotatie-tool-01/project`).
2. Open een terminal of de console (afhankelijk van het besturingssysteem) en voer het commando `npm install` uit. Dit zal alle nodige afhankelijkheden van het project laden.
3. Wanneer alle afhankelijkheden geladen zijn, kan het commando `npm start` uitgevoerd worden. Dit zal de backend opstarten met als onderliggende commando `ng build && nodemon www.js`. Het defaultadres van de backend is `localhost:3000`. Figuur 5.5 toont voorbeelduitvoer van dit commando.
4. Open een nieuwe terminal of console en voer het commando `ng serve` uit. Dit zal de webapplicatie lokaal starten. Dit commando zal zeggen op welk adres de applicatie draait. Het defaultadres is `localhost:4200`. Figuur 5.6 toont voorbeelduitvoer van dit commando.
5. Navigeer in een webbrowser naar `localhost:4200` om de applicatie te bedienen.

5.4 Het uitvoeren van testen

Er zijn twee luiken van testen. Het eerste luik test de operaties die worden gebruikt door de webapplicatie. Het tweede luik zijn de JUnit testen voor de MP4 Converter.

5.4.1 Testen van de webapplicatie

De benodigde software is voor dit luik al geïnstalleerd indien de vorige secties correct uitgevoerd zijn. Start de applicatie zoals beschreven in 5.3 en navigeer naar de folder

```

69% building modules 568/569 modules 1 active ...gular-bootstrap-md/scss/mdb-fr
Date: 2018-05-04T16:01:05.352Z
Hash: 00180b81f4312f965311
Time: 17144ms
chunk {inline} inline.bundle.js, inline.bundle.js.map (inline) 3.89 kB [entry] [
rendered]
chunk {main} main.bundle.js, main.bundle.js.map (main) 131 kB [initial] [rende
d]
chunk {polyfills} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 202 k
B [initial] [rendered]
chunk {scripts} scripts.bundle.js, scripts.bundle.js.map (scripts) 425 kB [initi
al] [rendered]
chunk {styles} styles.bundle.js, styles.bundle.js.map (styles) 550 kB [initial]
[rendered]
chunk {vendor} vendor.bundle.js, vendor.bundle.js.map (vendor) 6.73 MB [initial]
[rendered]
[nodemon] 1.17.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node ./backend/www.js`
body-parser deprecated bodyParser: use individual json/urlencoded middlewares ba
ckend/app.js:59:9
body-parser deprecated undefined extended: provide extended option node_modules/
body-parser/index.js:105:29

```

Figuur 5.5: Uitvoer van `npm start` op een linux toestel

`annotatietool-01/project` en voer het commando `npm test` uit. Indien de testen correct zijn uitgevoerd wordt de output van figuur 5.7 weergegeven.

5.4.2 Testen van de MP4 Converter

Aangezien de MP4 Converter geschreven is in Java is het logisch dat de testsuites geschreven zijn met behulp van JUnit. Indien sectie 5.1 correct uitgevoerd is, kunnen de JUnit testen uitgevoerd worden. Navigeer vanuit de root folder naar `java/MP4Converter/` en voer het volgende commando uit: `java -jar RunTests.jar`. Om te weten dat alle tests correct uitgevoerd worden zou de output van figuur 5.8 tevoorschijn moeten komen.

```

69% building modules 585/587 modules 2 active ...trap-md/scss/bootstrap/bootstr
69% building modules 585/588 modules 3 active ..._modules/core-js/modules/_is-a
69% building modules 585/589 modules 4 active ...core-js/modules/_to-absolute-i
69% building modules 585/590 modules 5 active ...modules/rxjs/_esm5/scheduler/q
69% building modules 586/590 modules 4 active ...modules/rxjs/_esm5/scheduler/q
69% building modules 587/590 modules 3 active ...modules/rxjs/_esm5/scheduler/q
69% building modules 588/590 modules 2 active ...trap-md/scss/bootstrap/bootstr
69% building modules 588/591 modules 3 active ...s/rxjs/_esm5/scheduler/QueueAc
69% building modules 588/592 modules 4 active ...xjs/_esm5/scheduler/QueueSched
69% building modules 589/592 modules 3 active ...xjs/_esm5/scheduler/QueueSched
69% building modules 590/592 modules 2 active ...trap-md/scss/bootstrap/bootstr
69% building modules 591/592 modules 1 active ...gular-bootstrap-md/scss/mdb-fr
Date: 2018-05-04T15:48:34.051Z
Hash: c7fe41aa517078cafc0e
Time: 17481ms
chunk {inline} inline.bundle.js (inline) 3.85 kB [entry] [rendered]
chunk {main} main.bundle.js (main) 306 kB [initial] [rendered]
chunk {polyfills} polyfills.bundle.js (polyfills) 549 kB [initial] [rendered]
chunk {scripts} scripts.bundle.js (scripts) 425 kB [initial] [rendered]
chunk {styles} styles.bundle.js (styles) 1.51 MB [initial] [rendered]
chunk {vendor} vendor.bundle.js (vendor) 18.3 MB [initial] [rendered]
webpack: Compiled successfully.

```

Figuur 5.6: Uitvoer van `ng serve` op een linux toestel

```

PS E:\annotatietool-01\project> npm test
> annotatietool@1.0.0 test E:\annotatietool-01\project
> jasmine

Randomized with seed 95453
Started
.....

8 specs, 0 failures
Finished in 3.064 seconds
Randomized with seed 95453 (jasmine --random=true --seed=95453)

```

Figuur 5.7: De uitvoer van correct uitgevoerde integratietesten

```

PS E:\annotatietool-01\java\MP4Converter> java -jar .\RunTests.jar
Testsuite klaar in 299 milliseconden
Aantal uitgevoerde testen : 6
Aantal genegeerde testen : 0
Aantal gefaalde testen : 0
PS E:\annotatietool-01\java\MP4Converter>

```

Figuur 5.8: De uitvoer van correct uitgevoerde JUnit testen