

Bachelorproef: Sprint 2  
Groep: Webslides\_01  
webslides-01.project.tiwi.be  
24/04/2018

Bavo Deprez  
Kurt D'haene  
Phaedra Neirinck  
Simeon Bruyland

The image shows a presentation slide titled "Web Fundamentals Introduction" by Ruben Verborgh, Ghent University - imec. The slide is part of a course called "Web Fundamentals" and is the first slide of a series. It lists five topics: 1. A Bird's-Eye View of the Web, 2. Web Architecture & Technologies, 3. Web APIs, 4. The Semantic Web & Linked Data, and 5. Linked Data Publishing. The slide also mentions "Next topic: A Bird's-Eye View of the Web" and "I occasionally teach courses like this one, and I do this for 3 main reasons." The reasons are: 1. I like teaching. 2. I want to spread the vision of the Web. 3. I aim to attract talented people. The slide also includes a note about opinions being marked with a triangle symbol and a note about the Web being the best invention of the past 30 years.

The side panel shows a list of annotations. The first annotation is by "agent name" and is a question about the first slide. The second annotation is by "jos" and is a question about the second slide. The third annotation is by "jan" and is a question about the third slide. The fourth annotation is by "jan" and is a question about the fourth slide. The fifth annotation is by "jan" and is a question about the fifth slide. The sixth annotation is by "jan" and is a question about the sixth slide. The seventh annotation is by "jan" and is a question about the seventh slide. The eighth annotation is by "jan" and is a question about the eighth slide. The ninth annotation is by "jan" and is a question about the ninth slide. The tenth annotation is by "jan" and is a question about the tenth slide.

Het project rond Open Weblides is in onderzoeksfase aan de universiteit Gent. Er wordt gezocht naar een gebruiksvriendelijk, gratis platform voor het delen van cursusmateriaal. Dit wordt gerealiseerd via een mini-forum rond weblides waarbij interactie tussen studenten en professoren mogelijk is. Met de annotatietool kunnen opmerkingen verbonden worden aan cursuselementen. Vervolgens kan er gemakkelijk genavigeerd worden tussen die opmerkingen via verschillende functies. De weblides zijn een betere manier om informatie te bundelen en kunnen hierbij andere platformen integreren.

## Inhoudstafel

1 Inleiding-----	
1.1 Context-----	p.1
1.2 Probleemstelling-----	p.1
1.3 Doelstelling-----	p.2
1.4 Structuur van het verslag-----	p.2
2 Gebruikersaspecten-----	
2.1 High-level requirements-----	p.3
2.2 Use case diagrammen-----	p.3
2.3 Volledige feature list-----	p.6
3 Systeemarchitectuur -----	
3.1 High-level systeemmodel-----	p.7
3.2 Klassendiagrammen-----	p.8
3.3 Sequentiediagrammen-----	p.24
3.4 Databank-----	p.27
3.5 Activiteitendiagrammen-----	p.27
4 Testplan-----	
4.1 Unit testen-----	p.29
4.2 Integration testen-----	p.29
4.3 Usability testen-----	p.30
4.4 Platform testen-----	p.31
5 Intallatiehandleiding en gebruikershandleiding-----	
5.1 Installatiehandleiding-----	p.32
5.1.1 Applicatie includeren in bestaand project-----	p.33
5.1.2 Wijziging doorvoeren-----	p.33
5.1.3 Applicatie starten-----	p.33
5.1.4 Applicatie stoppen-----	p.34
5.1.5 Configuratie-----	p.34
5.1.6 Jest installeren-----	p.34
5.2 Gebruikershandleiding-----	p.34
5.2.1 Openen van de annotatietool-----	p.35
5.2.2 Navigatieknoppen-----	p.35
5.2.3 Annotaties filteren en bekijken-----	p.35
5.2.4 Annotaties plaatsen-----	p.36
5.2.5 Annotaties aanpassen en verwijderen-----	p.37
5.2.6 Reactie plaatsen op een annotatie-----	p.37
5.2.7 Annotaties en reacties beoordelen-----	p.37

## **Overzicht van de toegevoegde figuren**

Figuur 1: use case diagram

Figuur 2: Deployment diagram

Figuur 3: Klassendiagram met inhoud van Component-module

Figuur 4: Klassendiagram met relaties en inhoud van LocalData-, Caller- en AnnotationForm-modules

Figuur 5: Klassendiagram met inhoud van Manager-, AnnotationOverview-, Drawer- en Navigation-modules

Figuur 6: Klassendiagram back-end

Figuur 7: Sequentiediagram AnnotationOverview-module

Figuur 8: Sequentiediagram AnnotationForm-module

Figuur 9: Activiteitendiagram bij toevoegen van een annotatie in add-tabblad

Figuur 10: inhoud van makeDummyAnnotation.js

Figuur 11: inhoud van makeDummyAnnotation.test.js

Figuur 12: help-container

Figuur 13: foutmelding

Figuur 14: foutmelding

Figuur 15: foutmelding

Figuur 16: foutmelding

Figuur 16: navigatieknoppen

Figuur 17: Default-tabblad

Figuur 18: Overview-tabblad

Figuur 19: Add-tabblad

Figuur 20: annotatie met reacties

## **Overzicht van de toegevoegde tabellen**

Tabel 1: backlog

Tabel 2: Leden Annotation-klasse

Tabel 3: Leden Reaction-klasse

Tabel 4: Leden Voter-klasse

Tabel 5: Leden LocalData-klasse

Tabel 6: Leden Drawer-module

Tabel 7: Leden Caller-module

Tabel 8: Leden Nav-module

Tabel 9: Leden Manager-module

Tabel 10: Leden AnnotationOverview-module

Tabel 11: Leden AnnotationForm-module

Tabel 12: Relaties tussen klassen en modules

Tabel 13: Leden AnnotationsService-klasse

Tabel 14: leden AnnotationsCRUD-klasse

Tabel 15: leden AnnotationsAdapter-klasse

Tabel 16: testplan

# 1 Hoofdstuk 1: Inleiding

## 1.1 Context

*Open Weblides* is een project in onderzoeksfase aan de universiteit Gent. De *weblides* kunnen visueel vergeleken worden met een online PowerPointpresentatie, vergezeld van extra functionaliteit. Deze slides gebruiken namelijk enkel *JavaScript*, *HTML* en *CSS* waardoor ze steeds aanpasbaar zijn, net zoals gewone websites. Er moeten geen speciale programma's gedownload worden, want ze zijn zowel mobiel als op vaste toestellen beschikbaar. Het sleutelwoord "open" verwijst naar het gratis platform en de keuze tussen verschillende browsers. Diegenen die hun informatie online zetten op *weblides* krijgen ook zelf de keuze over de beschikbaarheid van deze info.

Op dit moment zijn de *weblides* al in gebruik door de makers bij het lesgeven, wel zonder bepaalde *features*. Vorig jaar werd er een bachelorproef ingediend rond het converteren van documenten naar *weblide*-formaat. Zo kunnen alle cursussen met gemak toegevoegd worden. In de toekomst zouden nog enkele *features* bijgevoegd worden, zoals het annoteren van slides, *cherrypicking* en het slimme afdrukken. Via een annotatietool kunnen er annotaties aangemaakt worden en achteraf, door middel van *cherrypicking*, verkozen worden tot favoriet door gebruikers. Studenten kunnen deze gemakkelijk nakijken of afdrukken tussen hun cursus, ingevoegd op de juiste plek.

Naast die mogelijke toekomstige *features*, is het basisidee al volledig uitgewerkt. De studenten zijn via hun UGent-account verbonden en kunnen discussies starten over verschillende *topics* rond hun cursusmateriaal. Door feedback van de professoren en medestudenten wordt dit uitgebouwd tot een heus forum, wat een positief effect heeft op toekomstige studenten.

## 1.2 Probleemstelling

Wanneer studenten vragen hebben in verband met de leerstof, raadplegen ze wiki's, sociale media en fora of sturen ze mails naar de professor. De benodigde informatie staat op verscheidene bronnen waardoor het een hele opgave is die te bundelen en hun referentie naar het juiste deel in de cursus bij te houden.

Gesprekken tussen professoren en studenten blijven privé en gaan uiteindelijk verloren. Ook de info die gevonden wordt op internet is vaak fout of niet meer geldig, wat niet steeds duidelijk is voor de studenten. Er zou een manier moeten komen waarop de prof zicht heeft op die bronnen en kan bijsturen waar nodig. Doorheen de jaren hebben studenten zelf oplossingen gezocht om elkaar te helpen, maar zonder inbreng van de professoren. Dit leidt tot wanordelijke documenten met vragen en discussies, maar zonder een uiteindelijk antwoord. Ook worden er statussen over verschillende *topics* gestart op sociale media, dit leidt tot "vakgroepen" die bijgehouden worden doorheen de jaren. Ze bevatten vaak vragen en antwoorden over een verouderde cursus en na veel zoekwerk blijkt er meestal niet veel nuttigs terug te vinden.

Op fora, sociale media en documenten is het niet gemakkelijk om te verwijzen naar delen uit de cursus of extra info mee te delen via bekende platformen als *YouTube*. Het zou handig zijn indien die verwijzingen integreerbaar zouden zijn, want dan hoeven die linken, pagina- en slidenummers niet meer vermeld te worden.

Vaak worden er vragen gesteld in de les die niet hoorbaar zijn voor het hele auditorium of enkel privé worden beantwoord in de pauze. Deze vragen zijn nuttig voor alle aanwezige studenten, dus het zou een verbetering zijn indien die bijgehouden kunnen worden op een online prikbord. De prof kan die gezamenlijk beantwoorden en op die manier op zijn eigen vragen ook snel reactie krijgen.

### **1.3 Doelstelling**

Via *websites* zou er een oplossing bedacht worden om zowel voor professoren als studenten op één plek alle informatie te verzamelen. De openbare cursussen kunnen door het publiek geraadpleegd worden, door professoren inhoudelijk aangepast worden en opmerkingen krijgen door studenten. Andere platformen zijn volledig integreerbaar dus die annotaties kunnen bestaan uit tekst, afbeeldingen of video's en quotes uit sociale media. De vraagsteller heeft daarom verschillende categorieën om uit te kiezen en kan *tags* toevoegen zodat er gemakkelijker gezocht en gefilterd kan worden. De zichtbaarheid van zo'n annotatie kan ingesteld worden om opmerkingen voor zichzelf te houden en er kunnen favoriete annotaties vastgezet worden. Op deze manier wordt er een mini-forum gecreëerd tussen medestudenten met betrouwbare informatie door inbreng van de professor. Elke annotatie en reactie daarop heeft namelijk een beoordelingssysteem en kan door aangeduide moderator weggehaald worden.

De applicatie kan gebruikt worden door professoren tijdens hun spreekuren, waarbij de annotatietools tijdelijk verborgen kunnen worden. Naast het forum zijn er nog andere gebruiksmogelijkheden, zo is er real-time interactie mogelijk waarbij studenten vragen kunnen beantwoorden via annotaties gedurende de les. Er is daardoor snelle feedback beschikbaar en engagement wordt gestimuleerd tijdens de lessen.

De *Open Webslides* zijn een gebruiksvriendelijk, gratis platform voor het delen van cursusmateriaal en helpen de toekomstige studenten en professoren.

### **1.4 Structuur van het verslag**

In vorige paragrafen werd het idee achter *Open Webslides* aangehaald en welke doelen er voorop gesteld werden. In hoofdstuk 2 worden de gebruikersaspecten overlopen en in hoofdstuk 3 komt de systeemarchitectuur aan bod. Het testplan wordt besproken in hoofdstuk 4.

De oorspronkelijke vraag van de klant wordt eerst weergegeven in paragraaf 2.1. De features werden uitgetekend in *use case* diagrammen, deze bevinden zich met verdere uitleg in paragraaf 2.2. Vanuit de *backlog* met noodzakelijke *features* en nieuwe ideeën, werden drie sprints opgesteld. Het complete overzicht en de gekozen opdeling worden vermeld in 2.3. De gebruiker en klant kunnen respectievelijk een gebruiks- en installatiehandleiding terugvinden in paragraaf 2.4.

Doorheen hoofdstuk 3 wordt de werking van de webapplicatie verduidelijkt met verschillende figuren, startend met het high-level systeem model (3.1) en vervolgens klasse- (3.2), sequentie- (3.3), databank- (3.4) en activiteitendiagrammen (3.5).

Welke tests en hoe ze uitgevoerd werden, worden besproken in het testplan in paragraaf 4.

## 2 Hoofdstuk 2: Gebruikersaspecten

### 2.1 High-level requirements

Het *Open Weblides team* had al een welbepaald idee achter het project en vele *features* daarvan werden al toegevoegd. De annotatietool is een nieuw idee dat de basis vormt voor dit project. Die annotaties zouden door de gebruiker volledig naar eigen zin gevormd kunnen worden, met gekozen zichtbaarheidsniveau, *tags*, titel, commentaar, categorie en visuele aanbrenge op de *slides* zelf. Via zoek- en filterfuncties zijn annotaties gemakkelijk op te sporen en via het “*cherrypicken*” kunnen favoriete annotaties bijgehouden worden door elke gebruiker. Op de overzichtspagina van de cursus worden alle annotaties van een presentatie weergegeven en kan de gebruiker door één klik navigeren naar een annotatie van keuze. De userinterface moet bovenal gebruiksvriendelijk zijn zodat iedereen ermee overweg kan en de annotatietools kunnen verborgen worden.

### 2.2 Use case diagrammen

Het *use case* diagram toont de verschillende *features* voor de gebruikers, bevindt zich op Figuur 1 en wordt hieronder verduidelijkt.

#### Use case 1

**Naam:** annotatie bekijken en zoeken

**Doelstelling:** De gebruiker kan de annotaties van één *webslide* bekijken op het *default*-tabblad of de annotaties van de hele presentatie bekijken op het *overview*-tabblad. Daarnaast kunnen de annotaties van een presentatie gefilterd worden op *commentary*, *contentTags*, *category* en *cherrypicking*. Dit wil zeggen dat de gebruiker achtereenvolgens kan filteren op de inhoud van de annotatie, de *tags* die meegegeven werden, de gekozen categorie en bij *cherrypicking* wordt er gecheckt of de annotatie wel een favoriet is van de opgegeven persoon.

**Actoren:** primair: gebruiker

**Precondities:** Er moet verbinding met internet en de databank zijn zodat de annotaties opgevraagd kunnen worden.

**Postcondities:** /

**Successscenario:** De gebruiker krijgt een gefilterde lijst met annotaties.

**Alternatieve scenario's:** De gebruiker krijgt geen lijst terug omdat geen annotaties matchen met de vraag.

#### Use case 2

**Naam:** annotatie plaatsen

**Doelstelling:** De gebruiker kan een annotatie opstellen door kiezen van een categorie (*remark*,

*extra, question*), titel, zichtbaarheidsniveau (*private, public*), *tags*, commentaar en het aanduiden van een bijpassend element in de *webslide*. Enkel het invullen van commentaar en titel is verplicht.

**Actoren:** primair: gebruiker

**Precondities:** De gebruiker klikt op het *add*-tabblad in de navigatiebalk. Er moet ook verbinding zijn zodat de annotatie verstuurd kan worden.

**Postcondities:** /

**Successscenario:** De gebruiker plaatst een annotatie en vindt deze terug op het *default*- of *overview*-tabblad.

**Alternatieve scenario's:** De gebruiker krijgt een foutmelding wanneer hij geen commentaar en/of titel meegegeven heeft of wanneer er geen verbinding is waardoor de annotatie niet geplaatst kan worden. Die foutmeldingen kunnen steeds weggeklikt worden door de gebruiker. De gebruiker kan de plaatsing ook annuleren door terug te keren naar het *default*-tabblad via de pijl of door het sluiten van de *sidebar*.

### Use case 3

**Naam:** reactie plaatsen

**Doelstelling:** De gebruiker kan een reactie opstellen door het invullen van het tekstvak onder een annotatie.

**Actoren:** primair: gebruiker

**Precondities:** De gebruiker maximaliseert een annotatie om het tekstvak voor het plaatsen van reacties te kunnen raadplegen. Er moet ook verbinding zijn zodat annotaties opgevraagd kunnen worden voor het weergeven van de maximale annotatie en de reactie verzonden kan worden.

**Postcondities:** /

**Successscenario:** De gebruiker plaatst een reactie en vindt deze terug op het *default*- of *overview*-tabblad na maximalisatie van de annotatie.

**Alternatieve scenario's:** De gebruiker krijgt een foutmelding wanneer er geen verbinding is waardoor de reactie niet geplaatst kan worden of er gebeurt niks omdat er geen tekst werd ingevuld in het tekstvak. Die foutmeldingen kunnen steeds weggeklikt worden door de gebruiker. De gebruiker kan de plaatsing ook annuleren door niet op de pijl te klikken, het *default*-tabblad te vernieuwen, naar het *overview*- of *add*-tabblad te navigeren of door het sluiten van de *sidebar*.

### Use case 4

**Naam:** annotatie/reactie aanpassen of verwijderen

**Doelstelling:** De gebruiker kan een annotatie/reactie aanpassen of verwijderen door klikken op de *edit*- of *delete*-knop. Wanneer er op de *edit*-knop geklikt wordt, wordt er genavigeerd naar een formulier met de huidige waarden van de annotatie of wordt het tekstvak waarin de reactie staat aanpasbaar.

**Actoren:** primair: gebruiker

**Precondities:** De gebruiker maximaliseert een zelfgeschreven annotatie of andere annotatie met eigen reacties om de *edit*- en *delete*-knop te kunnen raadplegen. Er moet ook verbinding zijn zodat annotaties opgevraagd kunnen worden en de aanpassing of verwijdering verzonden kan worden.



**Postcondities:** /

**Successscenario:** De gebruiker vindt zijn annotatie/reactie aangepast of niet terug bij verwijdering, op het *default*- of *overview*-tabblad.

**Alternatieve scenario's:** De gebruiker krijgt een foutmelding wanneer er geen verbinding is waardoor de annotatie/reactie niet aangepast of verwijderd kan worden. Die foutmeldingen kunnen steeds weggeklikt worden door de gebruiker. Net zoals bij het aanmaken van reacties/annotaties kan de aanpassing steeds stopgezet worden. De gebruiker kan de verandering annuleren door terug te keren naar het *default*-tabblad via de pijl of door het sluiten van de *sidebar*. De aanpassing van een reactie kan stopgezet worden door niet op de pijl te klikken, het *default*-tabblad te vernieuwen, naar het *overview*- of *add*-tabblad te navigeren of door het sluiten van de *sidebar*.

#### Use case 5

**Naam:** annotatie/reactie beoordelen

**Doelstelling:** De gebruiker kan een annotatie/reactie beoordelen door te klikken op de bijhorende duimen. Hierdoor verandert het weergegeven aantal *upvotes* en/of *downvotes* en deze aanpassing zal zichtbaar zijn telkens de gebruiker de annotatie/reactie herbekijkt. Door de ster aan te klikken, wordt de annotatie als favoriet gekozen en kan het teruggevonden worden wanneer iemand zoekt op de naam van de gebruiker in de *cherrypicking*-zoekfunctie.

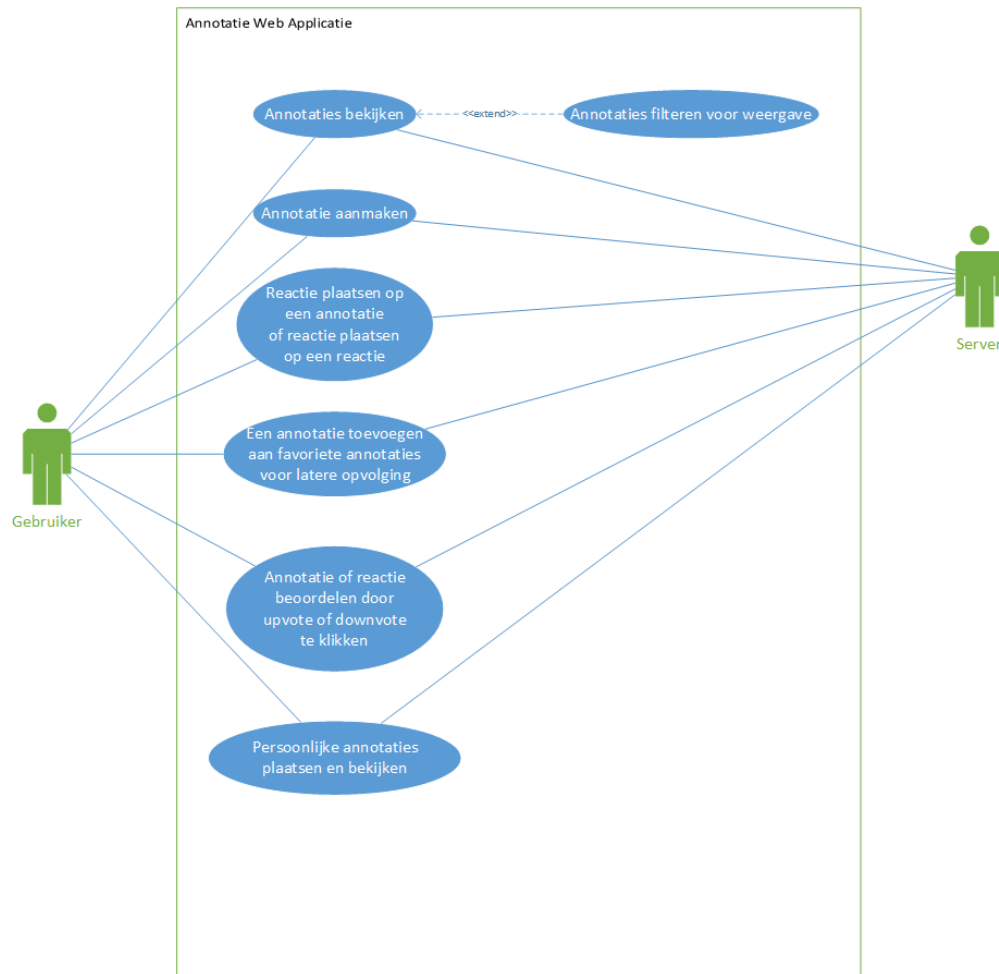
**Actoren:** primair: gebruiker

**Precondities:** De gebruiker maximaliseert een annotatie/reactie om de *upvotes* en *downvotes* te kunnen raadplegen en te checken of die annotatie al behoort tot zijn favorieten. Er moet ook verbinding zijn zodat annotaties opgevraagd kunnen worden en de aanpassing verzonden kan worden.

**Postcondities:** /

**Successscenario:** De gebruiker vindt de *upvotes* en/of *downvotes* en *cherrypicking*-ster van de annotatie/reactie aangepast terug op het *default*- of *overview*-tabblad.

**alternatieve scenario's:** Wanneer er geen verbinding is, gebeuren de veranderingen enkel lokaal. De gebruiker zal bij een volgende sessie opmerken dat die kwijt zijn.



Figuur 1: *use case* diagram

## 2.3 Volledige feature list

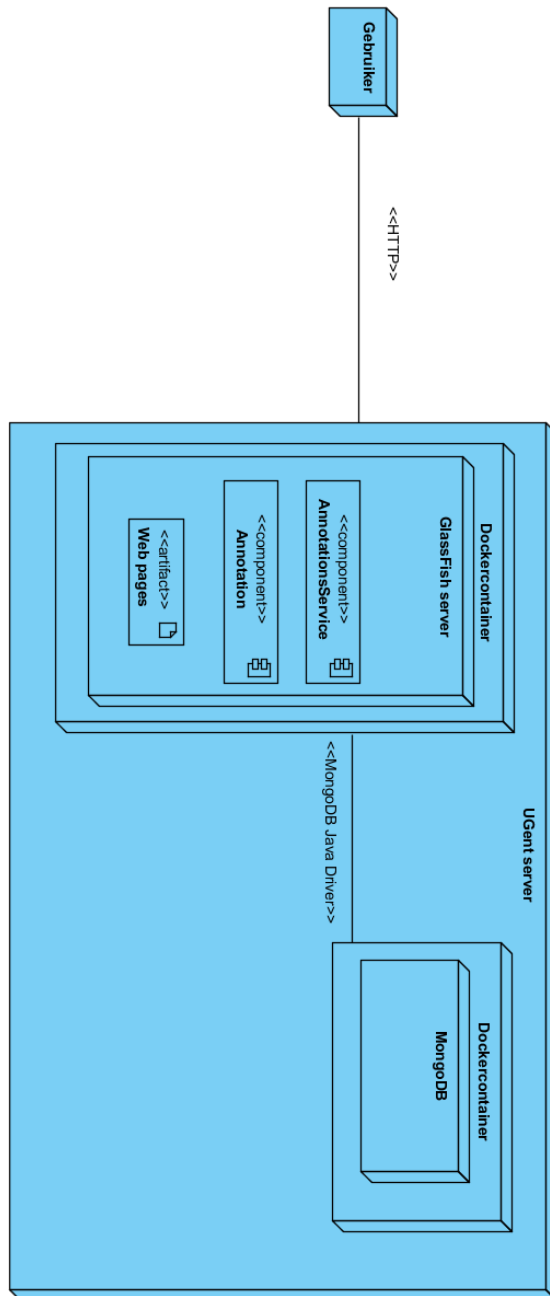
nr	Feature	sprint	prioriteit	complexiteit
1	De gebruiker kan de <i>sidebar</i> openen en verslepen	Sprint1	Hoog	
2	De gebruiker kan gegenereerde annotaties bekijken in de <i>sidebar</i>	Sprint1	Hoog	
3	De gebruiker kan annotaties cherrypicken, <i>upvoten</i> en <i>downvoten</i>	Sprint1	Hoog	
4	De gebruiker kan zelf annotaties aanmaken.	Sprint1	Hoog	
5	Annotaties worden bijgehouden in de databank en kunnen aangepast worden ( <i>put</i> , <i>post</i> , <i>delete</i> , <i>get</i> ).	Sprint2	Hoog	
6	Gebruikers kunnen interactieve gesprekken hebben.	Sprint2	Hoog	
7	De <i>sidebar</i> werkt op verschillende <i>browsers</i> .	Sprint2	Laag	
8	Er is een overzichtspagina met zoek- en filterfuncties.	Sprint2	Hoog	
9	Integratie met bestaande <i>webslides</i>	Sprint2	Hoog	

Tabel 1: *Backlog*

### 3 Hoofdstuk 3: Systeemarchitectuur

#### 3.1 High-level systeemmodel

Dit wordt weergegeven op Figuur 2 onderaan deze alinea. Er is gekozen om *Docker*-containers te gebruiken als productie-omgeving. Deze lichtgewicht virtuele machine draait een versie van een *GlassFish*-server, waarin de webpagina's zich bevinden die naar de *browser* van de gebruiker gestuurd worden. Deze *server* zal ook reageren op oproepen om annotaties te laden en al communiceren met de databank, die ook in een *Docker*-container geplaatst wordt.



Figuur 2: Deployment diagram

## 3.2 Klassendiagrammen

Het klassendiagram bevat de klassen *Annotation*, *Reaction* en *Voter*, deze zitten allen in de *Component*-module.

Leden van Component	Beschrijving	
Id	<i>Id</i> is een unieke identificatie dat gebruikt wordt voor de <i>AJAX-calls</i> ( <i>get</i> , <i>post</i> , <i>put</i> , <i>delete</i> ) met de databank. Bij aanmaak is de <i>id</i> nog null, want het wordt pas ingevuld na interactie met de databank.	
PresentationId	<i>PresentationId</i> stelt de unieke naam van de presentatie van <i>webslides</i> voor.	
SlideNumber	<i>SlideNumber</i> verwijst naar een specifieke <i>webslide</i> in de presentatie. Dit komt overeen met een id in de html van een <i>webslide</i> en kan dus zowel getallen als tekst bevatten.	
Title	<i>Title</i> is de titel die door de gebruiker aan de annotatie wordt gegeven.	
Element	<i>Element</i> is een html-element waarop de annotatie van toepassing is. Dit kan optioneel geselecteerd worden.	
Op	<i>Op</i> ( <i>original poster</i> ) is de gebruiker die de annotatie opstelde.	
Content	<i>Content</i> werd als anoniem object opgesteld omdat een eigen klasse onnodig is. De <i>content</i> omvat de inhoud van een annotatie	
	Commentary	<i>Commentary</i> is de commentaar die door de <i>op</i> werd ingegeven
	MarkType	<i>MarkType</i> staat <i>by default</i> ingesteld op fluo, maar zou als uitbreiding zelf gekozen kunnen worden Andere voorgestelde mogelijkheden zijn onderlijning en omkadering.
	Category	<i>Category</i> is de soort annotatie die gemaakt werd. Er is momenteel enkel keuze uit <i>Extra</i> , <i>Question</i> en <i>Remark</i> .
ContentTags	<i>ContentTags</i> zijn de <i>tags</i> die meegegeven worden door de <i>op</i> bij creatie. Voorbeelden hiervan zijn “belangrijk”, “opgelost” en “info”. Deze kunnen vrij worden gekozen.	
View	<i>View</i> is de zichtbaarheid en kan op <i>private</i> en <i>public</i> ingesteld worden.	
Cherrypicking	<i>Cherrypicking</i> stelt een dynamische favorietenlijst voor. Elke gebruiker die een annotatie wilt bijhouden, maakt ze tot favoriet. Hierdoor bevat <i>cherrypicking</i> alle gebruikers die in die specifieke annotatie geïnteresseerd zijn.	
Rating	<i>Rating</i> is een anoniem object, stelt het beoordelingssysteem voor van een annotatie en wordt bij aanmaak ingesteld op een lijst met nul <i>upvotes</i> en <i>downvotes</i> .	
	ThumbsUp/ThumbsDown	<i>ThumbsUp/ThumbsDown</i> bevat alle gebruikers die een <i>upvote/downvote</i> gaven.
Date	<i>Date</i> is de datum van plaatsing en wordt automatisch ingevuld.	
Reactions	<i>Reactions</i> bevat de reacties van andere gebruikers of de <i>op</i> zelf en is oorspronkelijk leeg. Deze behoren tot de <i>Reaction</i> -klasse.	
Upvote/Downvote	Via <i>upvote/downvote</i> worden <i>upvote/downvote</i> uit <i>Voter</i> opgeroepen.	

DeleteReaction	<i>DeleteReaction</i> zorgt voor het verwijderen van reacties uit <i>reactions</i> van een annotatie.
Cherrypick	Via <i>cherrypick</i> worden gebruikers in en uit de favorietenlijst <i>cherry-picking</i> gezet.

Tabel 2: Leden *Annotation*-klasse

Leden van Reaction	Beschrijving	
Person	<i>Person</i> is de gebruiker die de reactie opstelde.	
Date	<i>Date</i> is de datum van plaatsing en wordt automatisch ingevuld.	
Rating	<i>Rating</i> is een anoniem object, stelt het beoordelingssysteem voor van een reactie en wordt bij aanmaak ingesteld op een lijst met nul <i>upvotes</i> en <i>downvotes</i> .	
	ThumbsUp/ ThumbsDown	<i>ThumbsUp/ThumbsDown</i> bevat alle gebruikers die een <i>upvote/downvote</i> gaven.
Text	<i>Text</i> is de reactie die door <i>person</i> werd ingegeven.	
Upvote/Downvote	Via <i>upvote/downvote</i> worden <i>upvote/downvote</i> uit <i>Voter</i> opgeroepen.	

Tabel 3: Leden *Reaction*-klasse

*Voter* bevat statische methodes waar zowel *Annotation* als *Reaction* gebruik van maken om duplicatie te vermijden. Op vlak van werking is er geen verschil tussen *upvotes* en *downvotes* bij annotaties en reacties waardoor ze voor het grootste deel dezelfde hulpmethoden kunnen gebruiken. Hierbij wordt er gecontroleerd of de gebruiker zijn eigen annotatie of reactie niet probeert te beoordelen.

Leden van Voter	Beschrijving
Upvote/Downvote (statisch)	Via <i>upvote/downvote</i> worden namen toegevoegd/verwijderd uit de <i>thumbsUp</i> - en <i>thumbsDown</i> -lijsten in <i>rating</i> .

Tabel 4: Leden *Voter*-klasse

De *LocalData*-module bevat de gelijknamige klasse. Hierin zitten verscheidene methodes om lokaal de annotaties te kunnen bijhouden en bewerken. Op deze manier hoeft men niet steeds alles op te halen van de databank, toch wordt er op de nodige momenten data ververs. Uit de *annotationList* van *LocalData* kunnen specifieke annotaties opgevraagd worden door het *id* mee te geven dat werd opgegeven in de databank na creatie. De reacties krijgen geen *id* in de databank, maar hun plaats in de lijst *reactions* van een annotatie blijft steeds gelijk en wordt daarom als *id* beschouwd. Ook de *LocalData*-klasse kan teruggevonden worden in het klassendiagram.

Leden van LocalData	Beschrijving
AnnotationList	<i>AnnotationList</i> is een lokale lijst met alle annotaties na een laatste <i>refresh</i> uit de databank.
GetAnnotations	Via <i>getAnnotations</i> wordt de <i>annotationList</i> opgehaald.
AddAnnotation	<i>AddAnnotation</i> voegt een nieuwe annotatie toe aan de lokale <i>annotationList</i> . De nieuwe annotatie wordt niet in de databank gestoken via deze methode.
ClearData	<i>AnnotationList</i> kan geleegd worden via <i>clearData</i> .

FindAnnotationByHtmlId/ FindReactionByHtmlId	Via <i>findAnnotationByHtmlId</i> / <i>findReactionByHtmlId</i> kan een annotatie/reactie opgevraagd worden uit de lokale data.
---	---

Tabel 5: Leden *LocalData*-klasse

In de *Drawer*-module zitten methodes die reacties en annotaties maken en uittekenen. Er bestaan twee formaten van de annotaties: ze kunnen geminimaliseerd zijn waarbij ze enkel hun *op*, *title*, *category* en *contentTags* weergeven. Wanneer de annotatie gemaximaliseerd is, zijn ook de *commentary*, *reactions* (waaronder *person* en *text*) en *rating* zichtbaar. Daarnaast kan er aan de symbolen ook gezien worden of de ingelogde gebruiker zelf een *upvote* of *downvote* gaf of die annotatie als favoriet beschouwt. Op elke reactie en annotatie die van de gebruiker zelf afkomstig is, is er een *edit*- en *delete*-knop aanwezig. Enkel in dit formaat kunnen extra reacties toegevoegd worden. Het geminimaliseerde formaat wordt standaard uitgeprint bij weergave van de annotaties.

Leden van Drawer	Beschrijving
TagColors	Dit is een lijst die bepaalde kleuren kan <i>mappen</i> op een <i>tag</i> .
DrawAnnotations/ DrawAnnotationsMax/ DrawAnnotationsMin	Alle annotaties die meegegeven worden aan <i>drawAnnotations</i> worden uitgetekend in minimale versie <i>by default</i> . Wanneer je erna de annotatie wilt maximaliseren/minimaliseren, worden <i>DrawAnnotationsMax</i> / <i>DrawAnnotationsMin</i> gebruikt om de reeds uitgetekende html-code aan te passen. Deze methoden roepen <i>MakeContent</i> / <i>MakeContentMax</i> en <i>FillTags</i> op. De maximale versie van een annotatie heeft ook nog <i>FillReactions</i> en <i>MakeNewReaction</i> nodig.
MakeNewReaction	Wanneer de annotatie in maximaal formaat getoond wordt, staat er onderaan een extra tekstvak om reacties toe te voegen. Dat tekstvak wordt getekend door <i>makeNewReaction</i> .
DrawReaction	Nadat een nieuwe reactie opgesteld is en doorgestuurd wordt met een <i>AJAX-call</i> , wordt ze uitgetekend in <i>html</i> via <i>drawReaction</i> en roept ze daarbij <i>MakeReaction</i> op. <i>DrawReaction</i> verwacht het <i>annotation</i> -object waarvan de reactie afkomstig is, het <i>reaction</i> -object dat moet worden gevisualiseerd en een html-object als target waar de nieuwe html-reactie aan moet worden toegevoegd.
FillReactions	<i>fillReactions</i> splitst de lijst <i>reactions</i> van een annotatie op in aparte <i>reaction</i> -objecten en tekent ze elk uit in <i>html</i> in de <i>html</i> -annotatie via <i>makeReaction</i> .
MakeReaction	<i>MakeReaction</i> krijgt een <i>reaction</i> -object mee en zal aan de hand hiervan een <i>html</i> -container teruggeven die dan aan de <i>html</i> -pagina kan worden toegevoegd. De tweede parameter is een nummer, dat duidt op de hoeveelste reactie dit is op de annotatie en wordt gebruikt als id.
MakeContent/ MakeContentMax	Bij een geminimaliseerd formaat (dus ook <i>default</i> -versie) wordt de inhoud van de annotatie via <i>makeContent</i> opgebouwd, anders wordt <i>makeContentMax</i> gebruikt. Deze methode zorgt voor de <i>html</i> -opbouw van de binnenste delen van de <i>html</i> -annotatie.
FillTags	<i>FillTags</i> splitst <i>contentTags</i> op in aparte woorden en tekent ze uit in <i>html</i> -code in de <i>html</i> -annotatie, die als <i>target</i> wordt meegegeven.

SetAnnotationButtons-Style	De gebruiker kan zien welke annotaties of reacties hijzelf al <i>geupvote</i> of <i>gedownvote</i> heeft of als favoriet beschouwt door inkleuring van de symbolen (duim omhoog, duim omlaag en een ster) via <i>setAnnotationButtonsStyle</i> . Door deze methode worden de laatste wijzigingen onthouden na navigatie. Dit wordt opgeroepen wanneer een annotatie door <i>maximize</i> vergroot wordt waardoor die symbolen zichtbaar worden.
RefreshRatingButtons	<i>RefreshRatingButtons</i> is een hulpmethode die gebruikt wordt om naast de duimen het bijhorende aantal gebruikers te plaatsen uit de <i>rating</i> -lijst van de meegegeven annotatie of reactie.
Favorite	Via <i>favorite</i> wordt het ster-icoon van kleur veranderd zodat de gebruiker ziet of hij de annotatie aan zijn favorieten toegevoegd heeft en wordt de methode <i>cherrypick</i> van de annotatie opgeroepen.
AnnotationUpvote/ AnnotationDownvote/ ReactionUpvote/ ReactionDownvote	Deze methodes worden opgeroepen bij het klikken op een duim-icoontje. Ze zorgen voor het aanpassen van de kleuren van de duimen en roepen de bijhorende methodes van het annotatie- of reactieobject op om die annotatie of reactie te beoordelen. Deze methodes verschillen met de <i>SetAnnotationByttonsStyle</i> methode, waarbij deze methodes enkel het aangeklikte icoontje zullen aanpassen en de laatstgenoemde methode alle icoontjes in de volledige <i>html</i> -pagina bij het opstarten of maximaliseren van een annotatie zal controleren op de juiste kleur.
Maximize/Minimize	Wanneer op de knoppen linksbovenaan de annotaties, met een “+” of “-”, geklikt wordt, wordt de methode <i>maximize/minimize</i> opgeroepen die de draw-methodes ( <i>drawAnnotationsMax/Min</i> ) oproepen om de reeds uitgetekende <i>html</i> -code aan te passen.
AnnotationEdit/ ReactionEdit	De gebruiker kan steeds zijn eigen annotaties/reacties aanpassen door op de <i>edit</i> -knop te klikken in de gemaximaliseerde versie van de annotatie. Deze knop roept <i>annotationEdit/ReactionEdit</i> op.
ReactionDelete/ AnnotationDelete	De gebruiker kan steeds zijn eigen annotaties/reacties verwijderen door op de <i>delete</i> -knop te klikken in de gemaximaliseerde versie van de annotatie. Deze knop roept <i>annotationDelete/ReactionDelete</i> op.
ShowOnDia	Wanneer men een annotatie bekijkt in gemaximaliseerd formaat bevat die ook een <i>show</i> -knop om het bijhorende element in fluo weer te geven op de <i>webslide</i> of zelfs eerst naar die <i>webslide</i> te springen voor het aangeduid kan worden. Die knop roept <i>showOnDia</i> op.
DrawError/ DrawReactionError	<i>DrawError</i> is een algemene methode waarmee fouten kunnen weergegeven worden aan de gebruikers. Er komt een gekleurde balk tevoorschijn met extra uitleg die door de gebruiker gesloten kan worden. Deze methode wordt gebruikt in het formulier wanneer een annotatie gemaakt of aangepast moet worden. <i>DrawReactionError</i> tekent een icoon met uitroepteken zodat de gebruiker ziet dat er iets verkeerd is gebeurd. Dit wordt gebruikt bij het maken of aanpassen van een reactie.

DrawEmptyAnnotations	Wanneer er door slechte connectie geen annotaties opgehaald kunnen worden van de databank, of er geen annotaties zijn om weer te geven, wordt een melding in de plaats gegeven door <i>drawEmptyAnnotations</i> .
SetToDefault	Wanneer de gebruiker de <i>sidebar</i> sluit en terug opent, het formulier afsluit, vanuit een annotatie naar de juiste slide springt of terugkeert naar de <i>default view</i> , worden via <i>SetToDefault</i> zowel het formulier geledigd als de oude annotaties weggehaald om alles opnieuw te tekenen zoals in begintoestand.

Tabel 6: Leden *Drawer*-module

De *Caller*-module bevat de *Caller*-klasse en maakt interactie met de databank via *AJAX-calls* (*get*, *post*, *put*, *delete*). Dit heeft steeds een *LocalData*-object nodig.

Leden van Caller	Beschrijving
LocalData	Dit is een object van de <i>LocalData</i> -klasse waarin de <i>Caller</i> de opgehaalde annotaties in het lokale geheugen zal opslaan. Een <i>localData</i> -object is nodig om alle gegevens lokaal bij te kunnen houden voor interactie met de databank.
Omgeving	<i>Omgeving</i> bevat de <i>url</i> waarop de server zich bevindt. Tijdens het coderen worden soms dummy-waarden gebruikt via de <i>MyJson</i> -website, voor productie kan dit veranderd worden naar <a href="http://webslides-01.project.tiwi.be">http://webslides-01.project.tiwi.be</a> en voor testen met <i>Docker</i> kan dit op <i>Linux</i> vervangen worden door <a href="http://localhost:9003">http://localhost:9003</a> of op <i>Windows</i> door <a href="http://192.168.99.100:9003">http://192.168.99.100:9003</a> .
RestURL	<i>RestURL</i> is het gedeeltelijke <i>REST service</i> -pad waar de server naar luistert. Indien <i>/webresources/api/DUMMY</i> wordt gebruikt, worden dummy-waarden gegenereerd. Een lege werkende databank is beschikbaar via <i>/webresources/api</i> .
Url	<i>Url</i> bevat de volledige <i>url</i> waarnaartoe de <i>AJAX-calls</i> verlopen en bestaat uit <i>omgeving</i> + <i>RestURL</i> .
LoadAnnotationsFromServer	<i>LoadAnnotationsFromServer</i> haalt alle annotaties op via een <i>get-call</i> in <i>JSON</i> -formaat, splitst ze op en steekt ze in annotatie-objecten. Bij succes wordt de <i>callback</i> opgeroepen, bij falen wordt de <i>fail</i> opgeroepen.
SendAnnotationToServer	<i>SendAnnotationToServer</i> stuurt een <i>post-call</i> om een annotatie toe te voegen aan de databank. Bij succes wordt de <i>callback</i> opgeroepen, bij falen wordt de <i>fail</i> opgeroepen.
SendReaction	<i>SendReaction</i> stuurt een <i>post-call</i> om een reactie toe te voegen aan de databank en doet daarbij hetzelfde als <i>sendAnnotationToServer</i> . Bij succes wordt de <i>callback</i> opgeroepen, bij falen wordt de <i>fail</i> opgeroepen.
DeleteAnnotation	<i>DeleteAnnotation</i> stuurt een <i>delete-call</i> om die annotatie uit de databank te verwijderen. Bij succes wordt de <i>callback</i> opgeroepen, bij falen wordt de <i>fail</i> opgeroepen.
DeleteReaction	<i>DeleteReaction</i> roept <i>sendAnnotationToServer</i> op en verwijdert een reactie door de annotatie te overschrijven. Bij



	succes wordt de <i>callback</i> opgeroepen, bij falen wordt de <i>fail</i> opgeroepen.
--	--

Tabel 7: Leden *Caller*-module

De *sidebar* met de navigatiebalk wordt geregeld door de *Nav*-module. De *sidebar* heeft een minimale grootte (*default*-grootte) en kan door de gebruiker uitgetrokken worden tot gewenste grootte. De grootte wordt sowieso naar *full screen-mode* aangepast wanneer de gebruiker naar de *overview* navigeert.

Leden van Nav	Beschrijving
Size_sidebar_absolute	Er wordt steeds gerekend met de absolute grootte van de <i>sidebar</i> in <i>Size_sidebar_absolute</i> .
Open_sidebar	De <i>sidebar</i> wordt geopend door op het tekstvaksymbool langs de rechterkant te klikken, dit roept <i>open_sidebar</i> op.
Close_sidebar	De <i>sidebar</i> wordt gesloten door op het kruisje rechtsboven te klikken, dit roept <i>close_sidebar</i> op.
Show	<i>Show</i> wordt opgeroepen telkens er op een knop op de navigatiebalk wordt geklikt en zorgt daarbij voor het (on)zichtbaar maken van bepaalde elementen.
SetToDefaultButtons	Wanneer de gebruiker de <i>sidebar</i> sluit en terug opent, het formulier afsluit, vanuit een annotatie naar de juiste slide springt of terugkeert naar de default view, wordt er gesprongen naar de <i>default view</i> en wordt de navigatiebalk hersteld naar de juiste knoppen.

Tabel 8: Leden *Nav*-module

De *Manager*-module regelt de basiszaken. Het regelt de kleuren van de iconen, checkt welke gebruiker ingelogd is en waar de gebruiker zich bevindt: welke presentatie en specifieke *webslide*. Het zorgt voor instanties van de *LocalData*- en *Caller*klasse om data te kunnen doorgeven aan de databank.

Leden van Manager	Beschrijving
VoteColor/ FavoriteColor/ IconColor	Voor de <i>upvotes</i> , <i>downvotes</i> , favorieten en andere iconen in de navigatiebalk werden standaardkleuren voorzien.
PresentationId/ GetPresentationId	Na het opvragen van de huidige presentatie via <i>getPresentationId</i> , wordt het bijgehouden in <i>presentationId</i> .
SlideNumber/ GetSlideNumber	Na het opvragen van de huidige <i>webslide</i> van de presentatie via <i>getSlideNumber</i> , wordt het bijgehouden in <i>slideNumber</i> .
User/GetUser	De ingelogde gebruiker wordt opgevraagd via <i>getUser</i> en bijgehouden in <i>user</i> .
LocalData	Er wordt een instantie van de <i>LocalData</i> -klasse bijgehouden waarin een <i>annotationList</i> zit met lokale annotaties.

Caller	Er wordt een instantie van de <i>Caller</i> -klasse bijgehouden om <i>AJAX-calls</i> te doen voor dat <i>localData</i> -object.
MutationObserver	De <i>mutationObserver</i> is een <i>observer</i> die wijzigingen in de webpagina en opmaak kan detecteren. Deze variabele verwijst naar de <i>mutationObserver</i> van de <i>window</i> .
MutationObserverConfig	Dit is de configuratie van de <i>MutationObserver</i> . Deze zal enkel naar de <i>html</i> -elementen, inclusief hun attributen, kijken waaraan hij is toegevoegd en niet naar hun kindelementen.
Observer	In <i>observer</i> wordt een nieuwe instantie van de <i>MutationObserver</i> aangemaakt. Deze krijgt de methode <i>detectActiveSlideChange</i> mee als parameter en zal deze uitvoeren bij het detecteren van een wijziging aan de elementen waarnaar hij luistert.
DetectActiveSlideChange	Deze methode zorgt ervoor dat de <i>sidebar</i> opnieuw wordt opgevuld, nu met enkel de annotaties voor de momenteel actieve <i>slide</i> .
AddSlideChangeMutation-Observer	Deze methode voegt de <i>MutationObserver</i> toe aan alle elementen met de <i>CSS</i> -klasse <i>slide</i> . Deze moet bij opstarten van de <i>sidebar</i> worden opgeroepen.
Main	In de <i>main</i> wordt <i>refreshTable</i> en <i>AddSlideChangeMutation-Observer</i> opgeroepen. Deze methode wordt opgeroepen als alle bestanden zijn geladen.
RefreshTable	Dit roept steeds <i>loadAnnotationsFromServer</i> op om alle annotaties van de server op te halen en zal dan de <i>sidebar</i> opnieuw laten tekenen.
FixHtml	De gebruiker mag bij het schrijven van een annotatie of reactie geen <i>html</i> -code gebruiken, anders zouden er problemen kunnen optreden bij opslag of uitschrijven. De ingevoerde code wordt door <i>fixHtml</i> <i>html</i> -loos gemaakt.
GetAnnotationsById-Overview	Uit een gegeven lijst worden via <i>getAnnotationsByIdOverview</i> alle annotaties gehaald die behoren tot een specifieke presentatie.
GetAnnotationsById	Uit een gegeven lijst worden via <i>getAnnotationsById</i> alle annotaties gehaald die behoren tot een specifieke presentatie en <i>webslide</i> .
OpenHelpContainer	Dit opent de <i>helpcontainer</i> met een beschrijving van elk icoon.
CloseHelpContainer	Dit sluit de <i>helpcontainer</i> met een beschrijving van elk icoon.

Tabel 9: Leden *Manager*-module

De *AnnotationOverview*-module zorgt voornamelijk voor de zoekfuncties in het *overview*-tabblad waarmee er gezocht kan worden op *commentary*, *contentTags*, *category* en *cherrypicking*.

Leden van Annotation-Overview	Beschrijving
-------------------------------	--------------

FilterTopComments	Er kan via <i>FilterTopComments</i> een top N van de annotaties in een presentatie opgevraagd worden, afhankelijk van het beoordelingssysteem.
ViewAnnotation	In de <i>overview</i> kan via <i>ViewAnnotation</i> elke zoekfunctie gebruikt worden ( <i>commentary</i> , <i>contentTags</i> , <i>category</i> en <i>cherrypicking</i> ): eerst wordt alles opgehaald via <i>SearchFilter</i> en daarna wordt elke helpfilter erop toegepast. Vervolgens wordt alles uitgeschreven via <i>putResults</i> .
SearchFilter	Eerst worden alle annotaties voor een bepaalde presentatie opgehaald via <i>GetAnnotationsByIdOverview</i> en daarna wordt <i>HelpSearchFilter</i> daarop opgeroepen.
HelpSearchFilter	Uit een gegeven annotatielijst worden via <i>HelpSearchFilter</i> alle annotaties gehaald die een specifiek woord bevatten in hun <i>commentary</i> .
HelpTagFilter	Uit een gegeven annotatielijst worden via <i>HelpTagFilter</i> alle annotaties gehaald die een specifieke <i>contentTag</i> hebben.
HelpCategoryFilter	Uit een gegeven annotatielijst worden via <i>HelpTagFilter</i> alle annotaties gehaald die tot een specifieke <i>category</i> behoren.
HelpCherryPicking	Uit een gegeven annotatielijst worden via <i>HelpCherryPicking</i> alle annotaties gehaald die favoriet zijn voor een persoon uit de meegegeven personenlijst.
PutResults	Via <i>putResults</i> worden alle gevonden annotaties na zoeken verdeeld over vier kolommen en uitgetekend, <i>by default</i> in minimaal formaat.

Tabel 10: Leden *AnnotationOverview*-module

In de *AnnotationForm*-module zitten methodes die helpen bij de opbouw van het formulier. Wanneer tags worden toegevoegd door de gebruiker, krijgt elke tag een eigen knop. De gebruiker kan die tags verwijderen door zo'n *button* aan te klikken. Wanneer de gebruiker over de elementen in de *webslide* hovert, gaan ze één voor één gekleurd worden in fluo. Door te klikken, bevestigt de gebruiker welk element bij de annotatie hoort. Hierna kan de gebruiker zijn keuze veranderen door te herklikken. Naast het formulier voor de annotaties bevat dit ook de functies die gebruikt worden voor het “mini-formulier” van reacties.

Leden van <i>AnnotationForm</i>	Beschrijving
Select	<i>Select</i> is een status die weergeeft of de gebruiker op dit moment een element wil uitkiezen. Indien dit aan staat, zullen de elementen in fluo aangeduid worden bij het <i>hooveren</i> .
Selected	Wanneer <i>selected</i> aan staat, is al minstens één keer geklikt geweest waardoor <i>hooveren</i> geen invloed meer heeft. Indien er wel geklikt wordt op een ander element, verandert de selectie met fluo nog.
Last	Dit houdt het laatst geselecteerde element bij en is nodig om de fluo weg te krijgen.
OpenForm	Wanneer de gebruiker een nieuwe annotatie wilt maken, moet hij een nieuw formulier invullen dat geopend wordt vanuit het

	<i>add</i> -tabblad in de navigatiebalk (“+”-icoon). Deze knop roept <i>openForm</i> op.
Showtag	Er wordt een <i>button</i> met <i>tag</i> toegevoegd via <i>showtag</i> .
Deletetag	Er wordt een <i>button</i> met <i>tag</i> verwijderd via <i>deleteTag</i> .
Init	<i>Init</i> zorgt voor het kleuren van <i>webslide</i> -elementen in fluo, het bevestigen van het uiteindelijke element en het hernemen van die keuze.
EditStyle	<i>EditStyle</i> verandert de stijl van de elementen waarover <i>gehoovered</i> wordt naar fluo. Dit zou uitgebreid kunnen worden met onderlijning en omkadring.
SubmitAnnotation	Alle velden van het formulier worden ingelezen door <i>submitAnnotation</i> zodat er een annotatie-object opgesteld kan worden en doorgegeven kan worden aan de databank.
SubmitAnnotationAgain	Wanneer de gebruiker terugkeert naar een eerder gemaakte annotatie om die aan te passen, zal het alle info behalve <i>title</i> , <i>contentTags</i> , <i>category</i> , <i>commentary</i> , <i>view</i> en <i>element</i> behouden. Het formulier wordt ingevuld met de oude waarden. Het aanpassen van de annotatie gebeurt via <i>SubmitAnnotationAgain</i>
CancelAnnotation	De gebruiker kan het formulier verlaten via <i>cancelAnnotation</i> . Dit zorgt ervoor dat selecteren van elementen wordt afgebroken.
PostReaction	Wanneer een reactie gepost wordt in het tekstvak onder de gemaximaliseerde annotatie, wordt <i>postReaction</i> opgeroepen bij drukken op de knop. Dit roept <i>sendReaction</i> op op <i>caller</i> en tekent die reactie uit via <i>drawReaction</i> .
PostReactionAgain	Als een reactie aangepast wordt, moet enkel de <i>text</i> veranderd worden en een <i>AJAX-call</i> gestuurd worden met de annotatie om die te overschrijven via <i>postReactionAgain</i> . Dit roept <i>sendReaction</i> op van de <i>caller</i> .

Tabel 11: Leden *AnnotationForm*-module

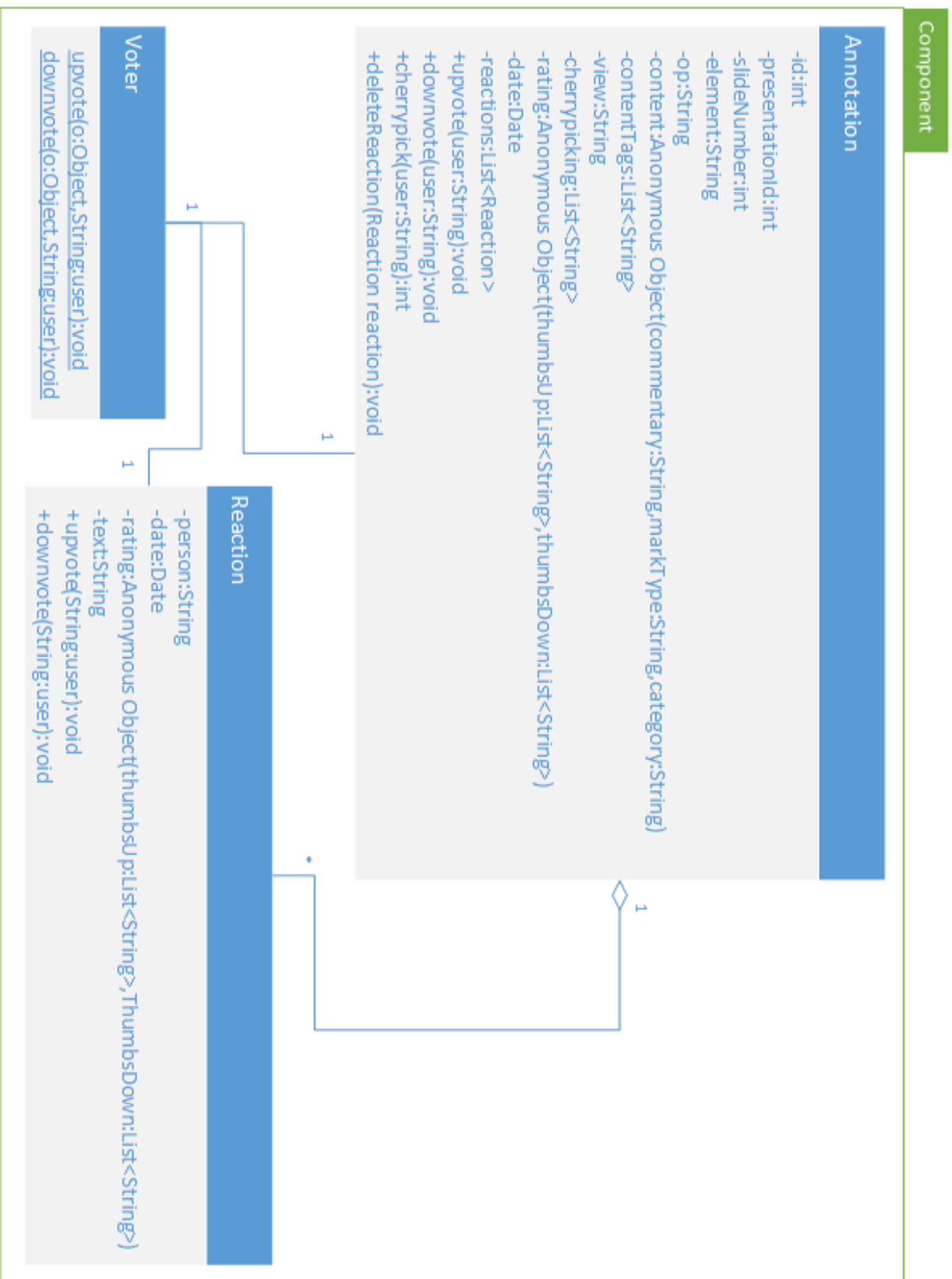
Tussen deze modules en klassen komen verschillende relaties voor.

Aggregation (bezitter en bezit):	
Annotation en Reaction	Een annotatie kan leven zonder reacties, maar een reactie hoort altijd bij een annotatie.
Caller en LocalData	Een <i>Caller</i> -object moet steeds een <i>LocalData</i> -instantie meekrijgen in de constructor.
Association (gebruikte en gebruiker)	
Voter en Annotation, Reaction	Annotaties en reacties gebruiken beiden de statische methode uit <i>Voter</i> voor hun beoordelingssysteem.
Annotation en AnnotationOverview	In de <i>overview</i> worden alle zoekfuncties op annotaties toegepast.
Annotation, Reaction, LocalData, Caller en AnnotationForm	Bij het versturen/aanpassen van annotaties en reacties in <i>AnnotationForm</i> wordt een

Caller en Manager	<i>LocalData</i> -object gebruikt om lokale (reeds opgehaalde) data aan te passen en een <i>caller</i> -instantie gebruikt voor de <i>AJAX-calls</i> . Wanneer annotaties en hun reacties ververst worden in de <i>Manager</i> , wordt de <i>Caller</i> -instantie gebruikt voor de <i>AJAX-call</i> .
Nav en AnnotationForm, Drawer, AnnotationOverview	Bij veranderen van tabblad in de navigatiebalk, worden elementen (on)zichtbaar gemaakt in <i>Nav</i> .
Annotation, Reaction, Caller, LocalData, Manager en Drawer	Voor het opvragen en uittekenen van annotaties/reacties in <i>Drawer</i> , wordt de lokale data opgevraagd via <i>AJAX-calls</i> . Om het kleur van de iconen te achterhalen, wordt de <i>Manager</i> geraadpleegd.

Tabel 12: Relaties tussen klassen en modules

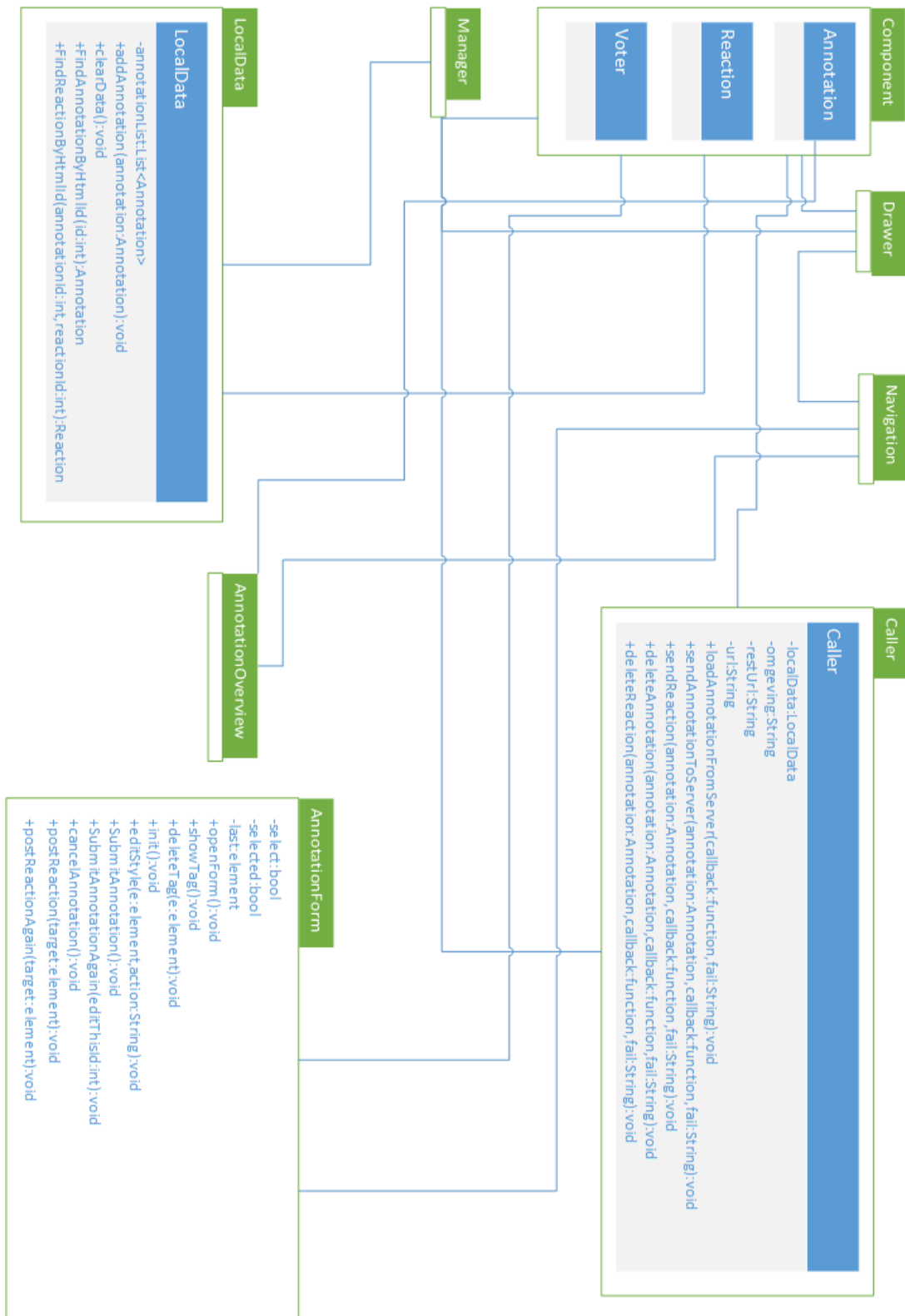
Hieronder, op Figuren 3,4 en 5 werden drie diagrammen bijgevoegd waarin de relaties tussen de verschillende modules en klassen duidelijk gemaakt worden.



Figuur 3: Klassendiagram met inhoud van *Component*-module



Figuur 4: Klassendiagram met relaties en inhoud van *LocalData*-, *Caller*- en *AnnotationForm*-modules



Figuur 5: Klassendiagram met inhoud van *Manager*-, *AnnotationOverview*-, *Drawer*- en *Navigation*-modules



De *back-end* heeft ook meerdere klassen om het gebruik van de annotatietool te vergemakkelijken, deze zijn in Figuur 6 als een klassendiagram voorgesteld. De klasse *AnnotationsService* regelt alle *HTTP-requests*. Hierbij zijn drie methoden voorzien:

Leden van AnnotationsService	Beschrijving
GetJson	<i>GetJson</i> haalt elke annotatie in de databank op en verstuurt die naar de aanvrager. Dit reageert op een <i>GET</i> -aanvraag.
PutJson	<i>PutJson</i> verwacht een annotatie in <i>JSON</i> -formaat en stuurt deze door om in de databank op te slaan. Dit reageert op een <i>POST</i> -aanvraag.
DeleteItem	<i>DeleteItem</i> verwacht een <i>id</i> van een annotatie en verwijdert die uit de databank. Dit reageert op een <i>DELETE</i> -aanvraag.

Tabel 13: Leden *AnnotationsService*-klasse

Deze klasse moet dus kunnen communiceren met de databank, hiervoor is een aparte klasse *AnnotationsCRUD* voorzien. Deze kan verbinding maken met de databank, om bv. annotaties op te halen of te verwijderen. Aangezien het belangrijk is dat deze verbindingen gesloten worden indien ze niet meer gebruikt worden, is gekozen voor een singletonpatroon.

Leden van AnnotationsCRUD	Beschrijving
Instance	Op elk gegeven moment zou <i>instance</i> de enigste instantie van deze klasse mogen zijn.
ConnString	<i>ConnString</i> is de connectiestring die gebruikt wordt om met de databank te connecteren.
DbNaam	<i>DbNaam</i> is de naam die men aan de databank wil geven (staat default op BachelorproefDB).
CollNaam	<i>CollNaam</i> is de naam van de collectie (te vergelijken met een <i>table</i> in <i>SQL</i> ) (staat <i>by default</i> op Annotaties).
GetInstance	<i>GetInstance</i> verkrijgt de instantie van de klasse.
DeleteAnnotationById	<i>DeleteAnnotationById</i> zoekt in de databank naar een annotatie met hetzelfde <i>id</i> dat meegegeven werd, en indien dit bestaat, wordt deze verwijderd.
GetItems	<i>GetItems</i> levert een lijst van elke annotatie uit de databank op.
PutJson	<i>PutJson</i> slaat de meegegeven annotatie op in de databank.

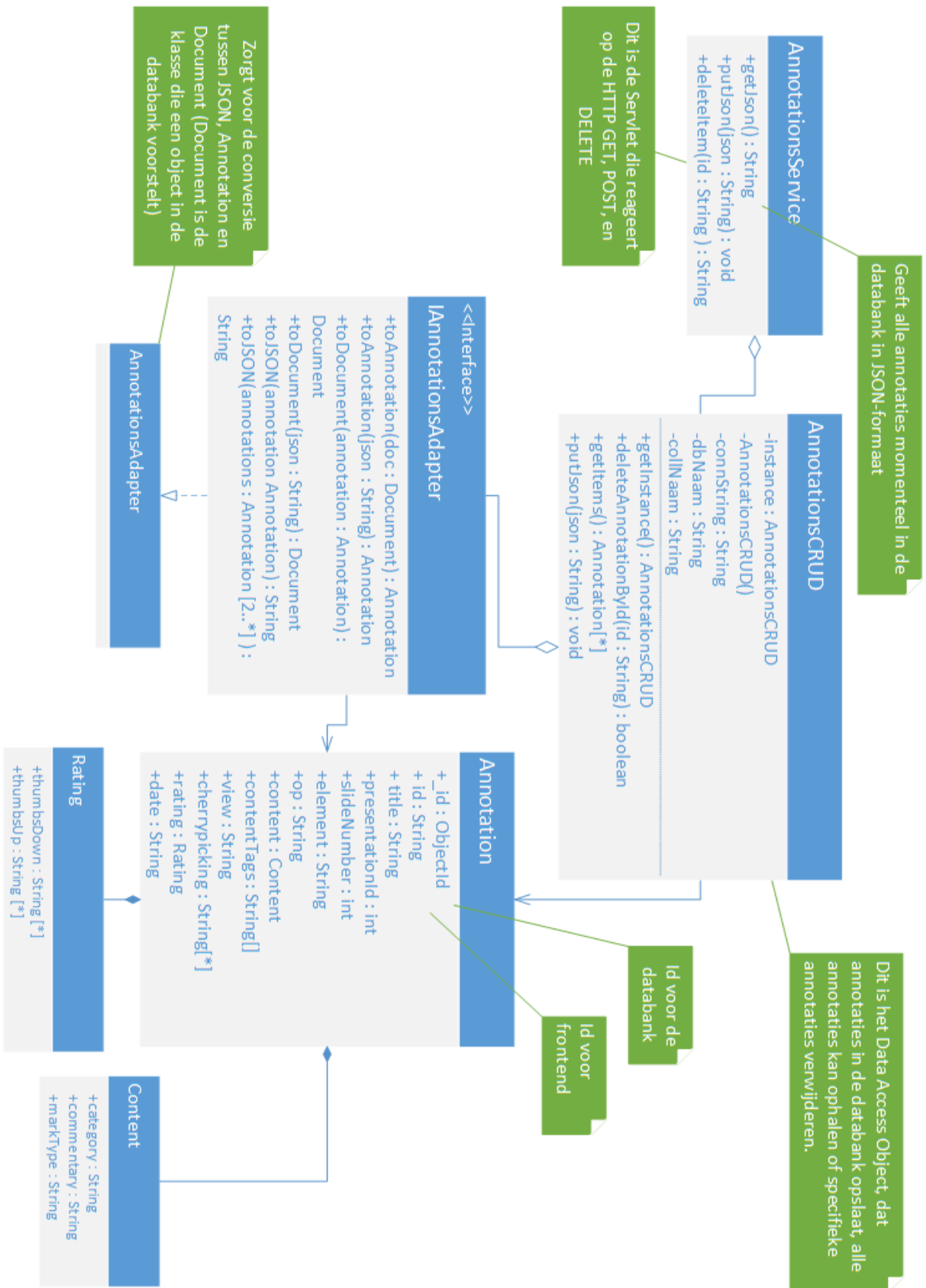
Tabel 14: leden *AnnotationsCRUD*-klasse

Voordat de annotaties opgeslaan worden, stelt men ze eerst voor als object. Hiervoor is de *Annotation*-klasse voorzien. Deze klasse dient als voorstelling van de eerdergenoemde *Annotation* van de *front-end* en is een opsomming van zijn leden identiek aan Tabel 2.

De *AnnotationsCRUD* moet vlot kunnen converteren tussen *JSON*, *Annotation* en *Document* (de klasse die een object in de databank voorstelt). Hiervoor is een aparte klasse voorzien: de *AnnotationsAdapter*.

Leden van AnnotationsAdapter	Beschrijving
toAnnotation	<i>ToAnnotation</i> kan ofwel een <i>Document</i> -instantie, ofwel een <i>JSON</i> -bestand omzetten naar <i>Annotation</i> .
toDocument	<i>ToDocument</i> zet ofwel een annotatie, ofwel een <i>JSON</i> -bestand om naar een <i>Document</i> -instantie.
toJSON	<i>ToJSON</i> kan één of meerdere annotaties omzetten naar <i>JSON</i> .

Tabel 15: leden *AnnotationsAdapter*-klasse

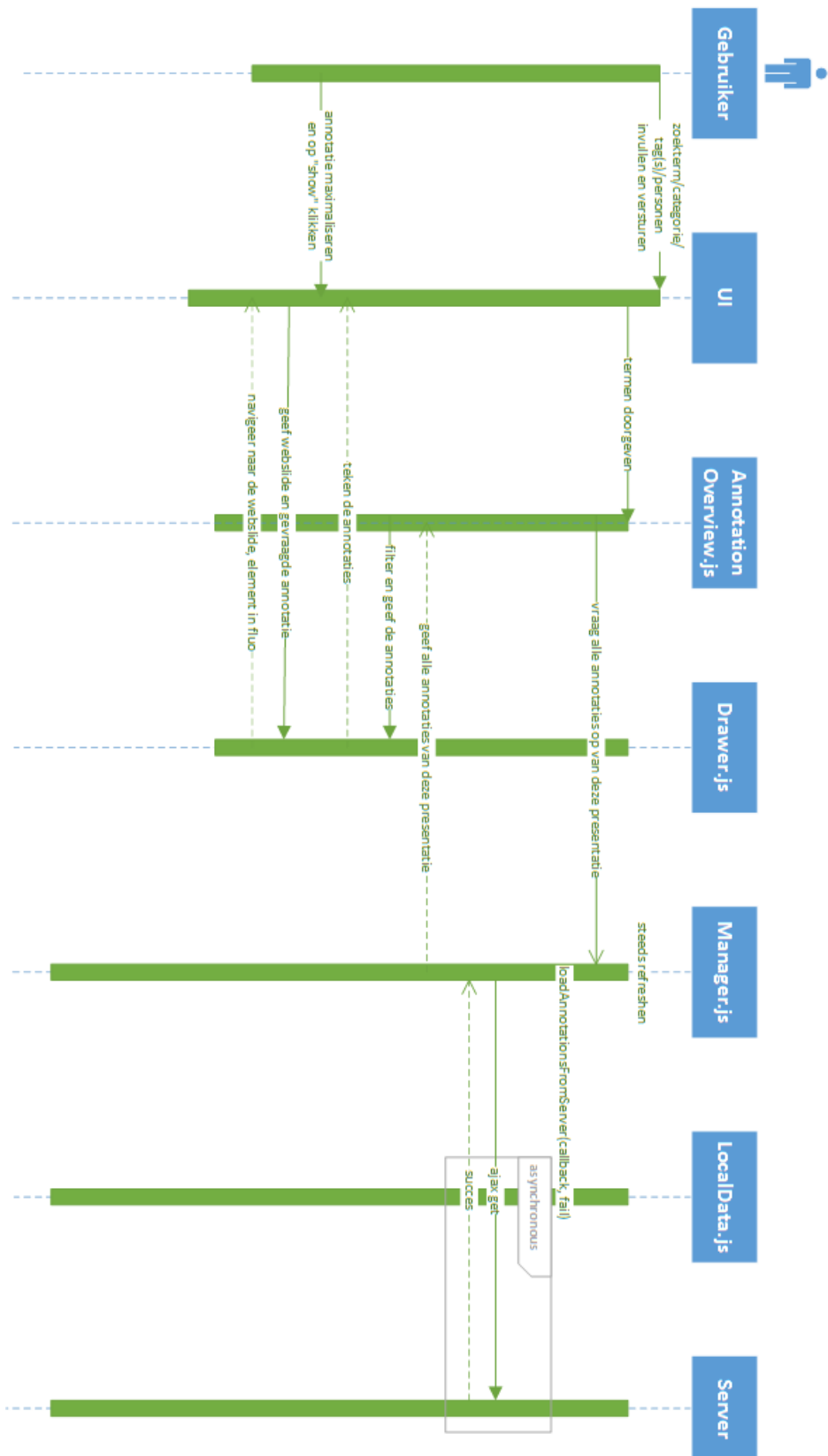


Figuur 6: Klassendiagram back-end

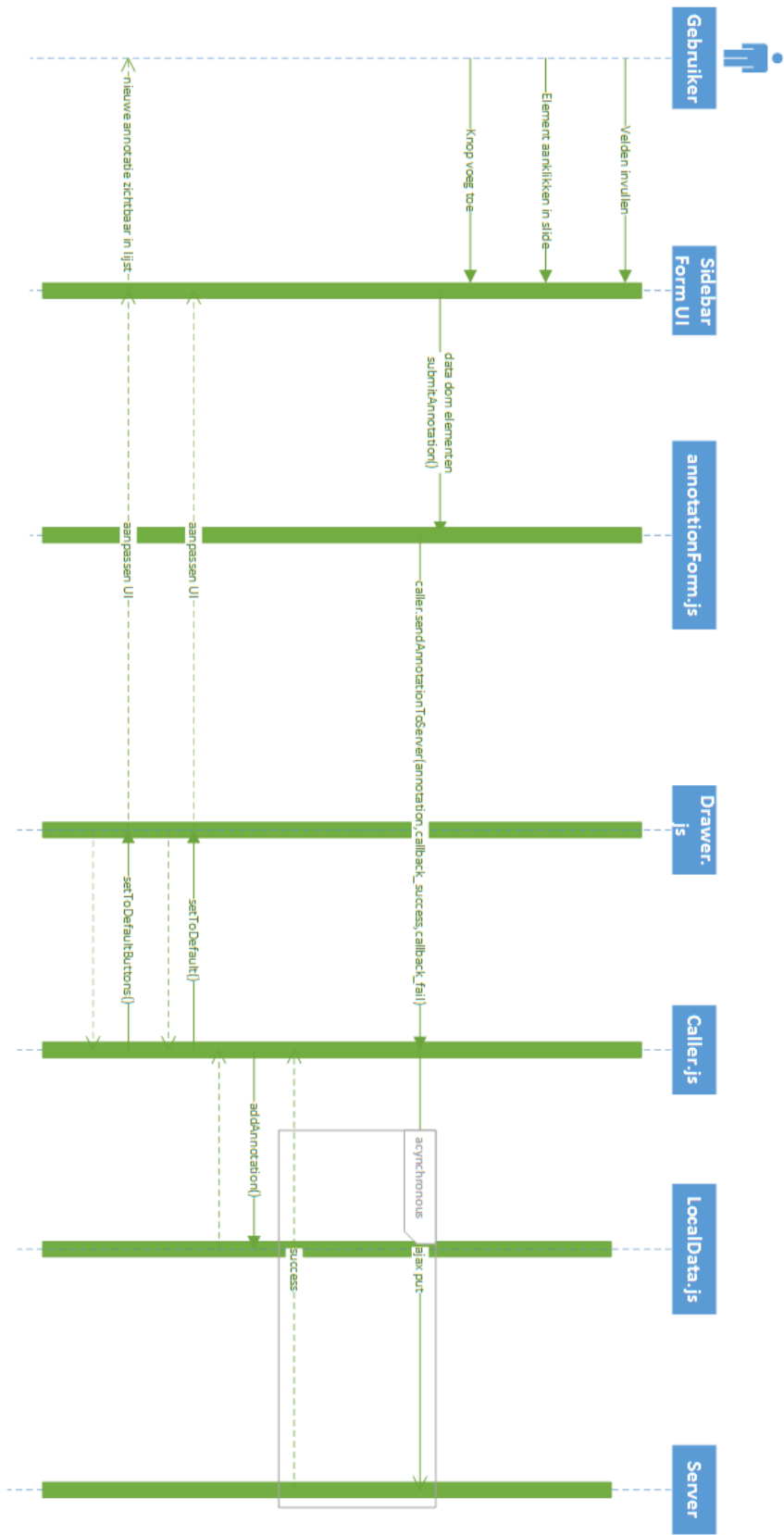
### 3.3 Sequentiediagrammen

Op Figuur 7 staat het sequentiediagram bij het zoeken van specifieke annotaties. In het *overview*-tabblad worden eerst alle annotaties uit de presentatie opgevraagd zodat er daarna achtereenvolgens gezocht kan worden op *commentary*, *contentTags*, *category* en *cherrypicking*. Wanneer er op geen enkele term gefilterd wordt, worden alle annotaties teruggegeven en wanneer er niks gevonden wordt na filtering, zal er ook niks weergegeven worden. Indien de gebruiker het bijhorend element bij die annotaties wilt zien, klikt hij op de *show*-knop. Er wordt vervolgens automatisch genavigeerd naar de juiste *webslide* in de presentatie met het bijhorende element in fluo.

Op Figuur 8 staat het sequentiediagram bij het invullen van het formulier. Er wordt samengewerkt tussen de *annotationForm*-, *Drawer*-, *Caller*- en *LocalData*-modules zodat de gemaakte annotatie naar de databank gestuurd kan worden en achteraf te zien is op de lijsten in het *default*- en *overview*-tabblad. De meeste interacties in de databank verlopen op een soortgelijke manier, waarbij de asynchrone *AJAX-calls* bij succes resulteren in het aanpassen van de weergaven in *html*. Hierbij staan de call in de *Caller*-klasse en het aanpassen van de weergave gebeurt in de *Drawer*-module.



Figuur 7: Sequentiedigram *AnnotationOverview*-module



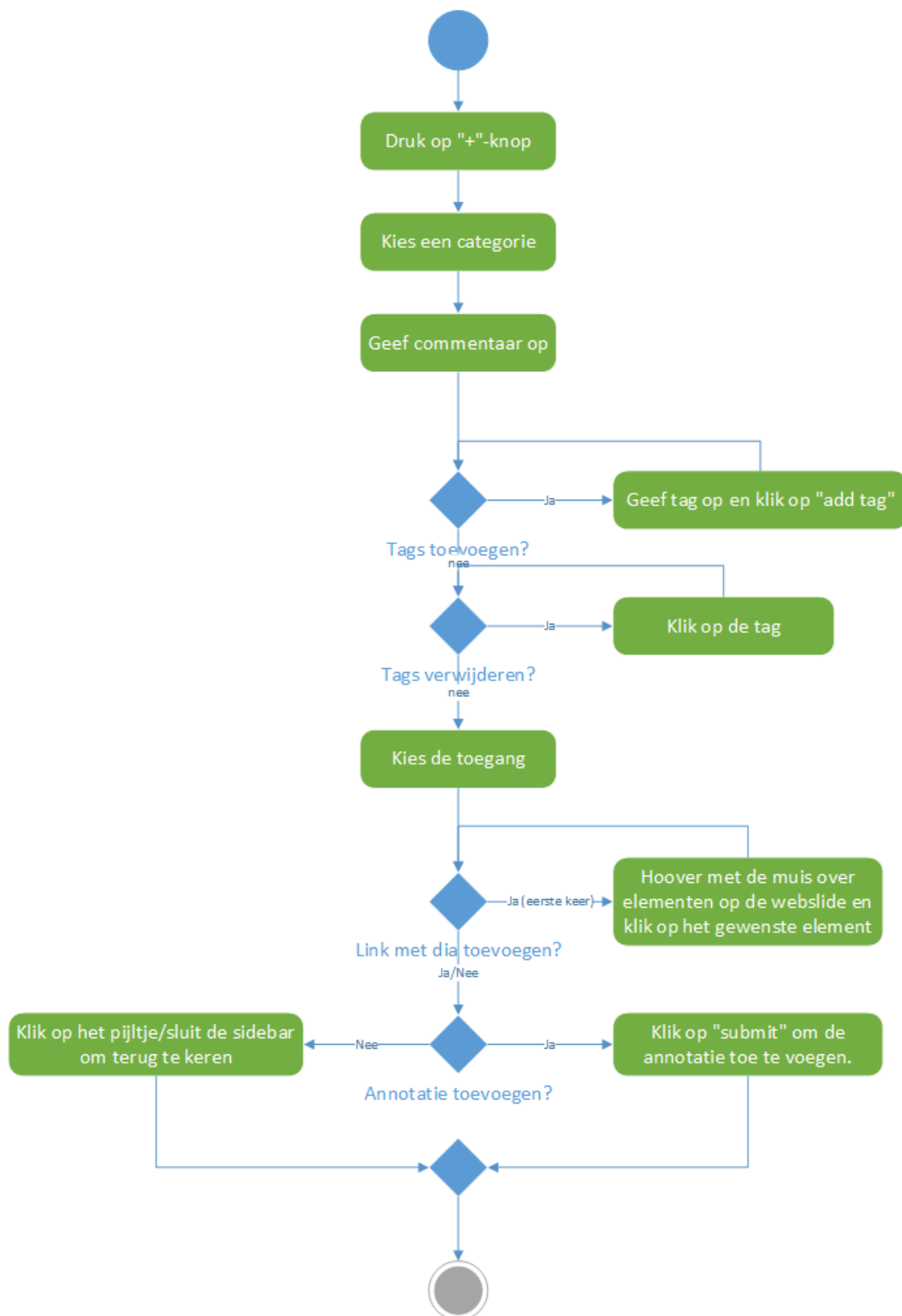
Figuur 8: Sequentiediagram *AnnotationForm*-module

### 3.4 Databank

Er werd gekozen voor de *NO-SQL* databank *mongoDB* (*opensource*) omdat het data bijhoudt in *JSON*-formaat, dit werd gebruikt voor *AJAX-calls* naar de *server* (*get*, *put*, *post*, *delete*). Deze databank is zeer flexibel en zorgt ervoor dat de velden kunnen variëren tussen verschillende documenten. Distributie gebeurt met gemak tussen verschillende computers.

### 3.5 Activiteitendiagram

Wanneer de gebruiker een nieuwe annotatie wilt plaatsen, kan dat op het *add*-tabblad (met het “+”-icoon). De enige verplichting is het invullen van de commentaar en titel en indien dit niet gedaan wordt voor het versturen van de annotatie, komt er een foutmelding onder het formulier. Indien de gebruiker het formulier volledig wil invullen, zal hij een categorie (*remark*, *question*, *extra*), *tags* en zichtbaarheidsniveau (*public*, *private*) moeten kiezen. Wanneer de muis over de bijhorende *webslide* *hovert*, zullen de *DOM*-elementen één voor één oplichten. Hieruit kan er een passend element gekozen worden door te klikken. Wanneer de annotatie gepost is, kan diezelfde gebruiker deze aanpassen of verwijderen via de *edit*- en *delete*-knop. Op Figuur 9 staat het activiteitendiagram bij het invullen van het formulier.



Figuur 9: Activiteitendiagram bij toevoegen van een annotatie in *add*-tabblad



## 4 Testplan

Door rekening te houden met de *use cases* werden volgende testen voorzien: toevoegen, verwijderen en aanpassen van een annotatie en een reactie. Naast deze automatische tests, werden er natuurlijk ook manuele voorzien. Hierbij hielden we rekening met de *pop-ups* die verschijnen wanneer de verbinding opeens zou wegvallen, de complexiteit van het eindproduct en de ondersteuning op andere *browsers*. In het testplan in Tabel 16 staan de te uitvoeren testen en hun scenario's.

### 4.1 Unit testen

De *back-end server* is geschreven in *Java*. De functionaliteiten van de server worden getest aan de hand van *unit testen*. Hierbij worden vooral de methodes voor toevoegen en ophalen uit de databank getest.

### 4.2 Integration testen

Er werd gebruik gemaakt van *Jest* om de webapplicatie uit te testen. Dit platform werd ontwikkeld door *Facebook* en wordt door veel sociale media geïntegreerd. Hiermee werden zowel de aparte modules als hun samenwerking beoordeeld.

In paragraaf 6.1.6 worden de nodige installaties voor het uitvoeren van *jest*-testen overlopen. Nu alles geïnstalleerd is, kan het formaat van een huidige test bekeken worden zodat er in de toekomst met gemak nieuwe testen gevormd kunnen worden. De testen bevinden zich in *sidebar/scripts/tests* en bestaan elk uit twee bestanden, namelijk een *js*- en *test.js*-bestand.

Als voorbeeld werden de bestanden *makeDummyAnnotation.js* en *makeDummyAnnotation.test.js* gekozen. Het doel van deze test is om een dummy-annotatie aan te maken, te verzenden naar de databank en te checken of deze wel werd toegevoegd. Onderstaande code op Figuur 10 is afkomstig uit *makeDummyAnnotation.js*.

```
export function makeDummyAnnotation(){
  let dummyData = new LocalData();
  let dummyCaller = new Caller(dummyData);
  //eerst de hele databank leegmaken en een lege annotatielijst terugkrijgen
  $.getJSON("webresources/api/flush", (result) => {
    dummyData=result;
  });
  let dummyAnnotation = new Annotation(null, "1", "6", "div", "Pieter", "Dit is fout", "fluo", "Extra", ["onopgelost"], "public");
  dummyCaller.sendAnnotationToServer(dummyAnnotation,
    () => {dummyData.addAnnotation(dummyAnnotation);},
    () => {}
  );
}
export function makeDummyAnnotationReturn(){
  makeDummyAnnotation();
  return dummyData.getAnnotations().size();
}
```

Figuur 10: inhoud van *makeDummyAnnotation.js*

In het *js*-bestand wordt telkens een exporteerbare functie gemaakt met een return-waarde, in dit geval *makeDummyAnnotationResult*. Vaak veronderstellen tests dat bepaalde onderdelen al gecheckt werden in vorige tests zodat er verder op gebouwd kan worden. Indien je de inhoud van een huidige test later wilt gebruiken in een volgende test kan die ook exporteerbaar gemaakt worden zoals in *makeDummyAnnotation* op Figuur 11.

```
const makeDummyAnnotationReturn = require('./makeDummyAnnotation.js');
test('checken of annotatie gemaakt wordt', () => {
  expect(makeDummyAnnotationReturn()).toBe(1);
});
```

Figuur 11: inhoud van makeDummyAnnotation.test.js

In het *test.js*-bestand wordt de functie geïmporteerd uit het *js*-bestand en constant beschouwd. Via een testfunctie wordt er vervolgens een boodschap met het doel van de test en de verwachte uitvoer weergegeven.

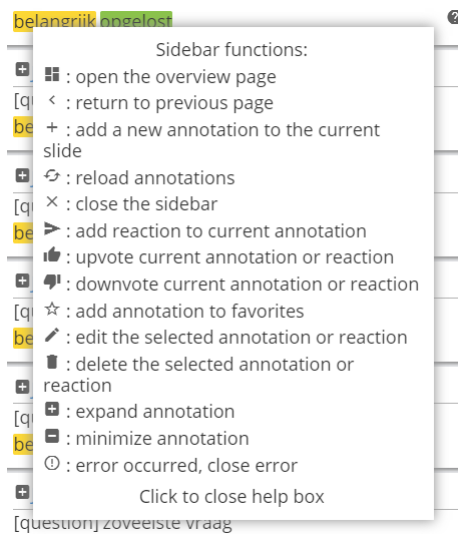
Nadat alle testen opgesteld zijn, moet *Jest* ook vermeld worden in de *package.json*, dit is al gebeurd in het project. Het uitvoeren van de testen kan via onderstaand commando:

```
<<npm test
```

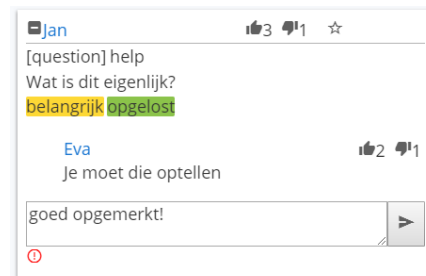
### 4.3 Usability testen

Het project werd doorheen de drie sprints aan verscheidene proefpersonen voorgelegd om te controleren of die courante versie van de annotatietool niet te complex was. Deze personen werden gevraagd enkele simpele opdrachten uit te voeren zoals het plaatsen van een annotatie, het beoordelen van een reactie, het opzoeken van een specifieke annotatie en het navigeren naar die *webslide*. Door de moeilijkheden die ze ondervonden en hun opmerkingen werden aanpassingen gedaan die leidden naar de laatste versie. Er moest een evenwicht gezocht worden tussen de simpele lay-out zonder beschrijvingen, gevraagd door de klant en een duidelijke lay-out met uitleg, gevraagd door de gebruiker. Om de website zo simpel mogelijk te laten ogen, werden beschrijvingen weggelaten en vervangen door iconen, zoals een duim voor de beoordeling en een ster als favoriet. Deze iconen komen vaak voor op sociale media, maar waren niet door iedereen gekend. Als oplossing hiervoor is er een Help-knop aanwezig die bij elk icoon een korte beschrijving geeft, zie Figuur 12.

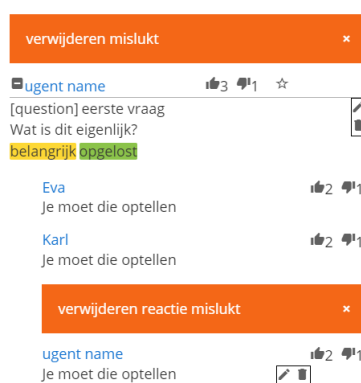
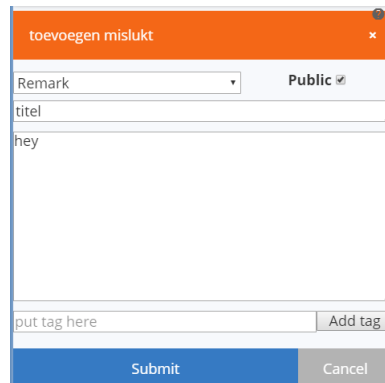
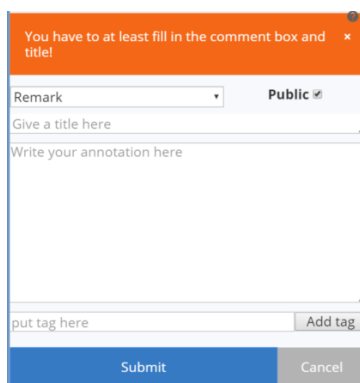
Wanneer de internetverbinding van de gebruiker opeens wegvalt, kan geen data meer doorgegeven of opgehaald worden van de databank. Bij het aanmaken/aanpassen/verwijderen van een annotatie of reactie zal een wegklikbare melding verschijnen, zie Figuren 13, 14, 15 en 16. Bij het *upvoten*, *downvoten* en toevoegen aan favorieten zullen de gemaakte wijzigingen niet doorgaan.



Figuur 12: help-container



Figuur 13: foutmelding



Figuren 14, 15 en 16: foutmeldingen

## 4.4 Platform testen

Het project werd op verschillende *browsers* uitgetest en werkt op de laatste versies van *Chrome* en *Firefox*.

wat	hoe	frequentie	wie
Front-end			
Aanmaken van annotatie	Integration test		
Aanmaken van reactie	Integration test		
Aanpassen van annotatie	Integration test		
Aanpassen van reactie	Integration test		
Verwijderen van annotatie	Integration test		
Verwijderen van reactie	Integration test		

Beoordelen van annotatie	Integration test		
Beoordelen van annotatie	Integration test		
Zoeken van annotatie	Usability test		
Foutmeldingen	Usability test		
Html-code invoeren	Usability test		
Aanmaken van annotatie	Usability test		
Plaatsten van reactie	Usability test		
Aanpassen van annotatie	Usability test		
Aanpassen van reactie	Usability test		
Verwijderen van annotatie	Usability test		
Verwijderen van reactie	Usability test		
Beoordelen van annotatie	Usability test		
Beoordelen van reactie	Usability test		
Ondersteuning van browsers	Platform test		
Back-end			
Annotatie verwijderen met specifieke <i>id</i>	Integration test		
Alle annotaties verwijderen	Integration test		
Annotatie toevoegen	Integration test		
Alle annotaties ophalen	Integration test		
Annotatie ophalen met specifieke <i>id</i>	Integration test		

Tabel 16: testplan

## 5 Installatiehandleiding en gebruikershandleiding

### 5.1 Installatiehandleiding

Hier vindt de klant een stappenplan voor het installeren van de nodige programma's zodat er in de toekomst gemakkelijk aanpassingen gedaan kunnen worden aan de code of dit geïntegreerd kan worden in een reeds bestaand project. Er wordt meegegeven hoe de *back-end* gerealiseerd werd (*Docker* en *MongoDB*) en hoe de testen (*Jest*) uitgevoerd kunnen worden.

De webapplicatie dient om annotaties aan te maken op de *webslides* en deze op een later ogenblik weer opnieuw te kunnen opvragen. De gegevens dienen dan ook persistent opgeslagen te worden, hiervoor werd gekozen voor de *NoSQL*-databank *MongoDB*.

Anderzijds worden de annotaties over het Internet naar de databank gestuurd, via een *GlassFish Server*. Deze reageert op inkomende oproepen om bijvoorbeeld annotaties op te halen of op te slaan.

Men kan via *Docker* deze software in een container starten, hierdoor dient men deze software niet lokaal zelf te installeren. Deze containers doen dienst als een soort lichtgewicht virtuele machine, waardoor de omgeving altijd dezelfde is. Verder wordt er verondersteld dat *Docker* geïnstalleerd is; dit kan op

*Windows* ([https://docs.docker.com/toolbox/toolbox\\_install\\_windows/](https://docs.docker.com/toolbox/toolbox_install_windows/)),

*Linux* (<https://docs.docker.com/install/linux/docker-ce/ubuntu/>) of

*MacOS* (<https://docs.docker.com/docker-for-mac/install/>).

### 5.1.1 Applicatie includeren in bestaand project

Op dit moment wordt er gewerkt met de *webslides Web Fundamentals* uit de verkregen link <https://github.com/RubenVerborgh/WebFundamentals>. Het is mogelijk de *sidebar* in andere projecten te includeren door de mappen *docker*, *sidebar*, *servlets* en *src* te kopiëren naar een eigen project. De *index.html* zal info bevatten van zowel de *sidebar* als de eigen *webslides*. Hier dient men alle code tussen `<!--sidebar code-->` en `<!--einde sidebar code-->` te kopiëren, en direct onder `<body class="shower list">` in het eigen *html*-bestand te plaatsen. Vervolgens moeten alle *imports* uit *index.html* vanaf `<!--w3css-->` t.e.m. `<!-- Google icons -->` toegevoegd worden in het eigen *html*-bestand. Deze *imports* moeten naar de juiste bestanden verwijzen in de *Sidebar*-map.

Het is belangrijk dat geen enkel *id* of *class* uit de CSS overeenkomt met het bijgevoegde project. Dit werd gerealiseerd door overal “*AnnotationTool\_*” voor te plaatsen zodat u hier geen rekening mee hoeft te houden.

In *Sidebar/scripts/data/manager* werden de methodes *getUser* (gebruikersnaam, wachtwoord) en *getPresentationId* voorzien die steeds werden ingevuld met dummy-waarden. Er wordt verondersteld dat deze door de klant vervangen worden met een eigen authenticatiemethode, indien nodig.

Ook kunnen er extensies toegevoegd worden, zoals de aanduiding van een gekozen element waarvoor momenteel enkel fluo beschikbaar is. Hiervoor werd er in de klasse *Annotation* al een lid voorzien met de naam *markType*.

### 5.1.2 Wijziging doorvoeren

Als ergens in de presentatie een wijziging gebeurt, kan men dit doorvoeren door de inhoud van die presentatie (met sidebar) te kopiëren naar *Servlets/AnnotationToolServer/web/WebApplicationApp* en vervolgens van de webapplicatie een nieuw distributiebestand te maken (<https://www.javatpoint.com/war-file>). Dit distributiebestand moet in *docker/webslides-01-glassfish* het vorig distributiebestand vervangen. Hierna kunnen de *Docker*-containers opnieuw opgestart worden.

### 5.1.3 Applicatie starten

Wanneer de *sidebar* correct is toegevoegd, is alles klaar om de containers op te starten. Open de *Docker Quickstart Terminal*, en navigeer daarna naar de map van de *repository* (via het commando *cd*). Hierin staat een script *buildandrun.sh*. Dit kan men uitvoeren als volgt:

```
$ ./buildandrun.sh
```

Dit script start beide *Docker*-containers op. Hierbij worden de *GlassFish Server* en *MongoDB*-databank in aparte containers opgestart. Vervolgens kan men naar <http://192.168.99.100:9003> surfen om de applicatie te gebruiken.

### 5.1.4 Applicatie stoppen

Start de *Docker Quickstart Terminal* op via volgend commando:

```
$ docker ps
```

U krijgt een overzicht van alle actieve containers, inclusief een hexadecimaal id. Hiertussen zouden de *GlassFish Server*- en *MongoDB*-container moeten staan. Indien men onderstaand commando met de eigen ID's in de plaats van ID1 en ID2 uitvoert, stopt men de applicatie.

```
$ docker stop ID1 ID2
```

### 5.1.5 Configuratie

De *GlassFish Server* wordt op poort 8080 gestart, in zijn container. Dit kan aangepast worden in *docker/webslides-01-glassfish/Dockerfile*, maar blijft het beste hierop staan. De uiteindelijke poort die opengezet wordt, vindt men terug in *docker/webslides-01.yml*. Default staat deze op 9003.

De URL van de server die de annotaties teruggeeft, staat hardgecodeerd in *WebFundamentals-master/Sidebar/scripts/data/Caller.js*, als de variabele *omgeving*. Indien een eigen server gebruikt wordt, wordt deze best aangepast.

### 5.1.6 Jest installeren

Alvorens de testen via *Jest* uitgevoerd kunnen worden, moeten enkele installaties gedaan worden. Ten eerste moet *Node.js* reeds aanwezig zijn op de computer waarin de *Node Package Manager (npm)* geïncludeerd is. Om dit te checken, worden volgende commando's uitgevoerd na het openen van de opdrachtprompt.

```
<<node -v
```

```
<<npm -v
```

Indien deze een foutmelding geven, worden ze beiden geïnstalleerd via [www.nodejs.org/en](http://www.nodejs.org/en).

Indien deze een specifieke versie weergeven, is alles al geregeld en kan *Jest* gedownload worden. Dit kan vervolgens via het commando:

```
<<npm install --save-dev jest
```

Nu alles geïnstalleerd is, kunnen de bestaande *tests* uitgevoerd worden.

```
<<npm test
```

In paragraaf 4 wordt het testplan weergegeven met *unit*, *integration* en *usability tests*, hun scenario en frequentie. Hierin wordt het formaat van een test besproken met een voorbeeld zodat nieuwe testen gemakkelijk opgesteld kunnen worden.

## 5.2 Gebruikershandleiding

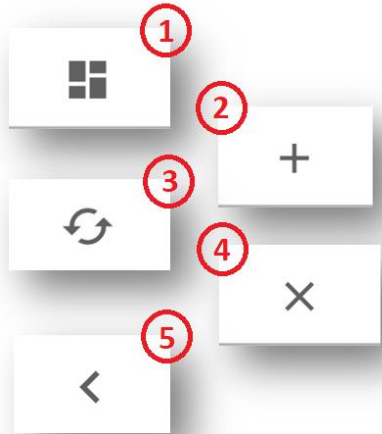
Functionaliteit:

- 1: Openen van de annotatietool
- 2: Navigatieknoppen
- 3: Annotaties filteren en bekijken
- 4: Annotaties plaatsen

- 5: Reactie plaatsen op een annotatie
- 6: Annotaties aanpassen en verwijderen
- 7: Annotaties en reacties beoordelen

### 5.2.1 Openen van de annotatietool

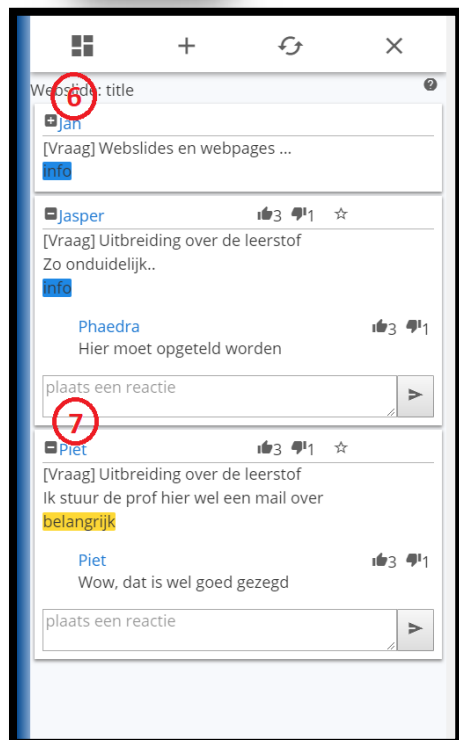
Om de annotatietool te openen, klikt u op het blauwe icoontje aan de rechterkant van het scherm. Door de blauwgekleurde balk aan de linkerkant van de annotatietool te verslepen kan u de grootte ervan aanpassen.



### 5.2.2 Navigatieknoppen

Via het *overview*-icoon (1) wordt het *overview*-tabblad geopend waarin u kan filteren op annotaties van de gehele cursus. Via het plus-icoon (2) wordt het *add*-tabblad getoond en kan u zelf een annotatie schrijven en toevoegen. Het *refreshen* van de annotaties kan door klikken op het *refresh*-icoon (3) te klikken. Met behulp van het kruis (4) sluit u de annotatietool. Om terug te gaan naar de *default*-tabblad, klikt u op de pijl. (5)

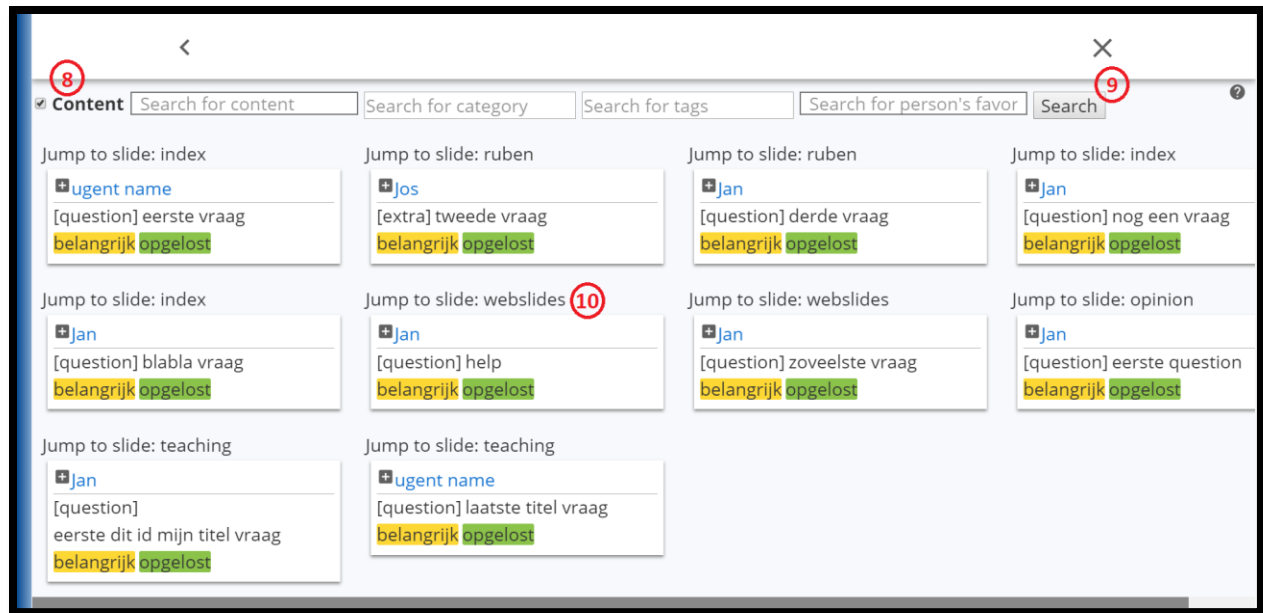
Figuur 17: navigatieknoppen



### 5.2.3 Annotaties filteren en bekijken

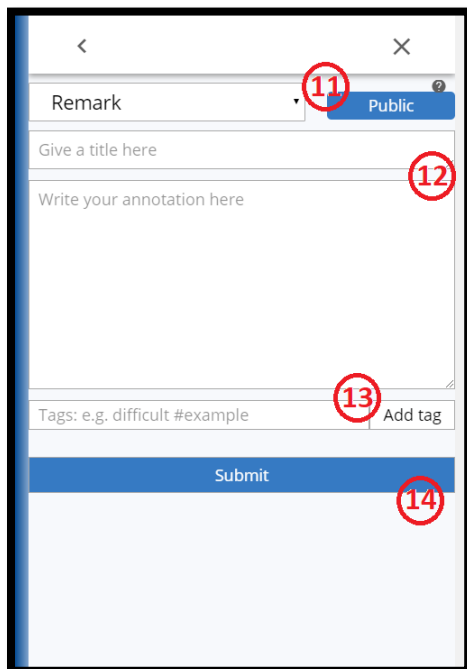
Wanneer de annotatietool is geopend, worden *by default* alle annotaties die bij de huidige *webslide* horen onder elkaar geminimaliseerd weergegeven. In deze geminimaliseerde versie zijn enkel de auteur, titel, categorie en tags te zien van de annotatie. Om de inhoud van de annotatie met zijn reacties en beoordelingssysteem te zien, klikt u op het plus-icoon (6) naast de auteur. Om de annotatie opnieuw te minimaliseren, klikt u op het min-icoon. (7)

Figuur 18: Default-tabblad



Figuur 19: Overview-tabblad

In het *overview*-tabblad kan men de annotaties van de gehele cursus bekijken en filteren (8). Hierbij zijn er verschillende mogelijkheden, er kan namelijk op annotatie-inhoud, op een categorie, op tag(s) en favorietenlijst gezocht worden. Dit zijn de annotaties die door een bepaalde persoon als favoriet zijn aangeduid. Als men na het ingeven van de filtertermen op de *search*-knop (9) drukt, worden de gefilterde annotaties in het kader eronder weergegeven. Klik op 'Jump to slide' (10) boven een annotatie in de lijst om naar de bijhorende slide te navigeren.



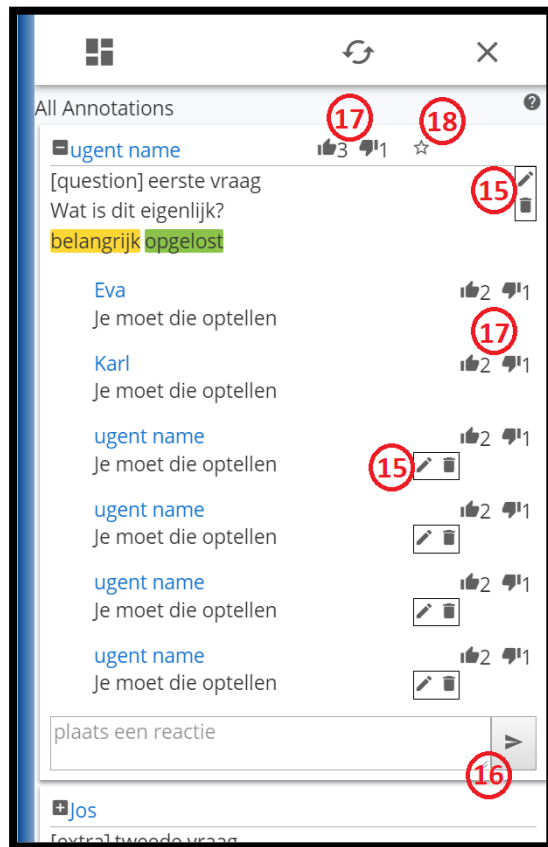
#### 5.2.4 Annotaties plaatsen

Om een annotatie te plaatsen bij een bepaalde slide, gaat u naar het *add*-tabblad. Er moet weliswaar eerst een *webslide* geselecteerd zijn. Men kan een categorie kiezen via de *drop down list* (11) en aanduiden of de annotatie publiek of privaat moet worden geplaatst.

In de invoegkaders (12) eronder geeft u een titel aan de annotatie en schrijft u de inhoud ervan. Er kunnen eventueel tags worden toegevoegd door een tag op te geven en hierna op de 'add tag'-knop (13) te klikken. Deze opgegeven tag verschijnt boven het invoegbalkje. Verwijder een tag door erop te klikken. Normaliter wordt een annotatie gekoppeld aan de gehele *slide*. Om een annotatie aan een specifiek deel van die *webslide* te koppelen, klikt u op het gewenste element na *hooveren* over uw keuzes in de *webslide*. Om de annotatie toe te voegen, klikt u op de *submit*-knop (14). Klik op de terug-knop links bovenaan om de annotatie te annuleren en terug te keren naar het default-tabblad.

Figuur 20: Add-tabblad





Figuur 21: annotatie met reacties

### 5.2.5 Annotaties aanpassen en verwijderen

Wanneer men in het *overview*-tabblad of *default*-tabblad een annotatie maximaliseert, zijn er twee nieuwe iconen te zien aan de rechterkant: Een potlood-icoon en een vuilnisemmer-icoon. (15) Na het klikken op het potlood-icoon kan men een annotatie bewerken. Het *add*-tabblad wordt geopend met daarin reeds ingevulde velden. Om de wijziging ook effectief door te voeren, klikt u op de *submit*-knop. Om een annotatie te verwijderen, klikt u op het vuilnisemmer-icoon.

### 5.2.6 Reactie plaatsen op een annotatie

Wanneer een annotatie gemaximaliseerd is, kan men er een reactie op plaatsen door commentaar in te geven in het invoegkader en daarna op de pijl (16) te klikken. De reactie verschijnt nu onderaan de lijst van reacties op deze annotatie. Net zoals bij de annotatie zelf kan men een reactie bewerken en verwijderen met behulp van het potlood- en vuilnisemmer-icoon (15).

### 5.2.7 Annotaties en reacties beoordelen

Wanneer een annotatie gemaximaliseerd is, kan u uw mening geven over een annotatie of reactie. Om dit te upvoten of downvoten, klikt u op de duim (17) naar omhoog of omlaag. Om een annotatie te verkiezen tot uw favorieten, klikt u op de ster (18). Andere gebruikers kunnen filteren op de annotaties die u als favoriet hebt aangeduid in het *overview*-tabblad.