1. Hoofdstuk 1: Inleiding

1.1 Context   
*Open Webslides* is een project in onderzoeksfase aan de universiteit Gent. De *webslides* 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 *webslides* krijgen ook zelf de keuze over de beschikbaarheid van deze info.

Op dit moment zijn de *webslides* 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 *webslide*-formaat. Zo kunnen alle cursussen met gemak toegevoegd worden in het vervolg. In de toekomst zouden nog enkele *features* toegevoegd worden, zoals het “cherrypicken” en het slimme afdrukken. Via *cherrypicking* kunnen er annotaties bijgehouden worden, gelinkt aan een welbepaald profiel. Zo kunnen studenten deze gemakkelijk nakijken of afdrukken tussen hun cursus, ingevoegd op de juiste plek.

Naast die mogelijke toekomstige *features*, is het basisidee al volledig uitgedokterd. 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 *webslides* 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 cursus-materiaal 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.

1. Hoofdstuk 2: Gebruikersaspecten

2.1 High-level requirements   
Het *Open Webslides* *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*, categorie en visuele aanbreng 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.

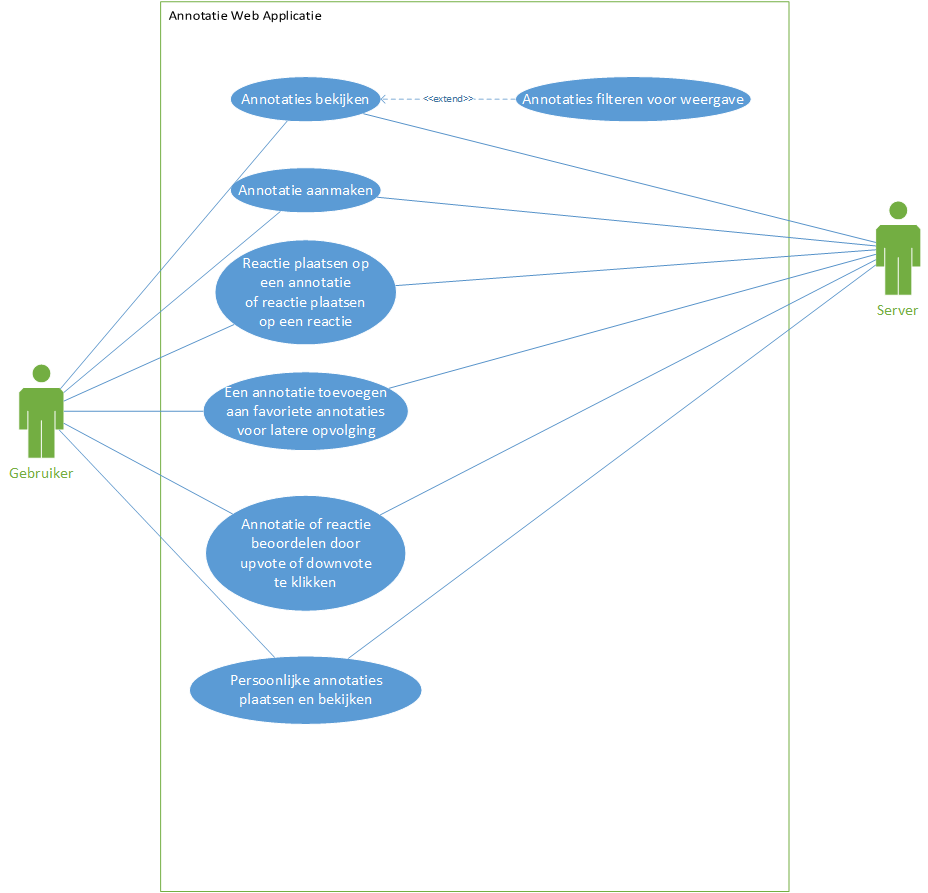
Use case 1  
Naam: annotatie bekijken en zoeken  
Doelstelling: De gebruiker kan de annotaties van één *webslide* bekijken op het *default*-tabblad of de annotaties van de hele presentatie bekijken op het *overview*-tabblad. Daarnaast kunnen de annotaties van een presentatie gefilterd worden op *commentary*, *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 zijn zodat de annotaties opgevraagd kunnen worden. Postcondities: /  
Successcenario: 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: /  
Successcenario: 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: /  
Successcenario: 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: /  
successcenario: 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 of er gebeurt niks omdat er geen commentaar werd ingevuld. Die foutmeldingen kunnen steeds weggeklikt worden door de gebruiker. De gebruiker kan de aanpassing ook annuleren door terug te keren naar het *default*-tabblad of door het sluiten van de *sidebar*.

Use case 5  
Naam: annotatie/reactie beoordelen   
Doelstelling: De gebruiker kan een annotatie/reactie beoordelen door klikken op bijhorende duimen. Hierdoor verandert het weergegeven aantal van *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: /  
Successcenario: 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 *sidebar* openen en verslepen | Sprint1 | Hoog |  |
| 2 | De gebruiker kan gegenereerde annotaties bekijken in de *sidebar* | Sprint1 | Hoog |  |
| 3 | De gebruiker kan annotaties cherrypicken, filteren, *upvoten* en *downvoten* | Sprint1 | Hoog |  |
| 4 | De gebruiker kan zelf annotaties aanmaken. | Sprint1 | Hoog |  |
| 5 | Annotaties worden bijgehouden in de databank en kunnen aangepast worden (*put, post, delete, get*). | Sprint2 | Hoog |  |
| 6 | Gebruikers kunnen interactieve gesprekken hebben. | Sprint2 | Hoog |  |
| 7 | De *sidebar* werkt op verschillende *browsers*. | Sprint2 | Laag |  |
| 8 | Er is een overzichtspagina met zoek- en filterfuncties. | Sprint2 | Hoog |  |
| 9 | Er gebeurde integratie met bestaande *webslides* | Sprint2 | Hoog |  |

Tabel 1: *Backlog*  
  
2.4 Werking

2.4.1Dit moet vervangen worden door de handleiding  
Wanneer de cursus geopend wordt, wordt de overzichtspagina weergegeven met rechtsboven de knop voor het uit- en inklappen van de sidebar. Die balk kan door de gebruiker verder uitgetrokken worden. Er zit wel een limiet op de minimale grootte en de maximale grootte is fullscreen-modus. In de balk zijn er drie tabbladen aanwezig om het geheel minder druk te maken: new, view en search.

In het new-tabblad zit een formulier voor het aanmaken van een annotatie. Dit bevat tekstvelden en keuzelijsten voor categorieën en tags, zichtbaarheidsniveau, commentaar en een zelfgeplaatste link naar een element uit de cursus via omkadering, inkleuring of onderlijning. Dat formulier wordt uitgeschakeld op de overzichtspagina omdat er geen bijhorende slide werd uitgekozen.

Het view-tabblad geeft alle annotaties en reacties weer voor die slide of een zelfgekozen aantal *top comments* bij de overzichtspagina. De gebruiker kan alles *up-* of *downvoten* en eigen toevoegingen aanpassen of verwijderen.

De zoek- en filteropdrachten worden weergegeven in het search-tabblad en kunnen over eigen slide of alle slides zoeken bij gebruik vanuit de overzichtspagina.

Figuur 2: screenshot Figuur 3: screenshot Figuur 4: screenshot  
*default*-tabblad *add*-tabblad *overview*-tabblad

**2.4.2 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.

**2.4.2.1 Docker installeren**

Hieronder wordt weergegeven hoe *Docker* geïnstalleerd wordt met *Windows* als *OS*, versie 10. Indien u een andere OS of versie gebruikt, kan u de volledige uitleg terugvinden op de link *www.docs.docker.com.*

Allereerst wordt *Docker* gedownload op de *website* [*www.docker.com/get-docker*](http://www.docker.com/get-docker). Indien de installatie compleet is, ziet u drie nieuwe iconen op uw bureaublad. Na het openen van het programma *Docker Quickstart Terminal*, wordt het volgende commando meegegeven.<<docker run hello-world  
Pas wanneer deze een boodschap teruggeeft, is de installatie geslaagd. Vervolgens zijn er nog een aantal aanpassingen nodig.  
<<cd C:/Temp/webslides-01/docker/webslides-01-glassfish  
<<dos2unix start.sh  
<<cd C:/Temp/webslides-01   
<<dos2unix buildandrun.sh  
Hierna opent u de *docker*-map en voert u *buildandrun.sh* uit door dubbelklikken. Wanneer dit compleet is, kan de databank geraadpleegd worden na paragraaf 2.4.2.2 op *192.168.99.100:9003.*  
 **2.4.2.2 MongoDB installeren**De *software* voor de gebruikte databank kan gedownload worden op [*www.mongodb.com*](http://www.mongodb.com). Er wordt verondersteld dat Windows, minimum 2008, gebruikt wordt als OS. Uw versie kan gecontroleerd worden door het volgende commando in de opdrachtprompt in te voeren:  
<<wmic os get caption  
Bij het downloaden van de databank moet gekozen worden voor de *Enterprise*-optie. Wanneer de download compleet is, moeten er nog enkele commando’s meegegeven worden via de opdrachtprompt. Spring naar de schijf waarop het programma werd gedownload en maak een map aan om data uit de databank in op te slaan.  
<<cd D:/ indien de installatie in de C-schijf gebeurde, is deze stap niet nodig.  
<<md \data\db  
Nu is alles geregeld en kan de databank geopend worden. Indien u de melding *“waiting for connections”* ontvangt, verloopt alles zoals verwacht. Bij het zien van een foutmelding, moet *“private networks, such as my home or work network”* aangevinkt en doorgestuurd worden via *“allow access”.*

**2.4.2.3 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.

**2.4.2.4 includeren in bestaand project**  
Op dit moment wordt er gewerkt met de *webslides* *Web Fundamentals* uit de verkregen link [*https://github.com/RubenVerborgh/WebFundamentals*](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*. Waar de *sidebar* eindig met commentaar “<!--einde sidebar code-->”, zal de inhoud van de *webslides* beginnen. 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 werd verondersteld dat deze door de klant vervangen zouden worden met een eigen authenticatiemethode. Het voorzien van een *editor* voor de tekstvelden werd niet als noodzakelijk beschouwd, dus werd enkel het minieme geregeld. *FixHtml* zorgt namelijk dat geen html toegelaten kan worden in de *text*- en *commentary*-velden van de *Annotation*- en *Reaction*-klasse.

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*.

1. Hoofdstuk 3: Systeemarchitectuur

**3.1** **High-level systeemmodel**   
Dit wordt weergegeven op Figuur 5 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 zal communiceren met de databank, die ook in een *Docker*-container geplaatst wordt.

Figuur 5: 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 | *Id* is een unieke identificatie dat gebruikt wordt voor de *AJAX-calls* (*get, post, put, delete*) met de databank. Bij aanmaak is de *id* nog null, want het wordt pas ingevuld na interactie met de databank. | | |
| PresentationId | *PresentationId* stelt de unieke naam van de presentatie van *webslides* voor. | | |
| SlideNumber | *SlideNumber* verwijst naar een specifieke *webslide* in de presentatie. Dit komt overeen met een id in de html van een *webslide* en kan dus zowel getallen als tekst bevatten. | | |
| Title | *Title* is de titel die door de gebruiker aan de annotatie wordtgegeven. | | |
| Element | *Element* is een html-element waarop de annotatie van toepassing is. Dit kan optioneel geselecteerd worden. | | |
| Op | *Op* (*original poster*) is de gebruiker die de annotatie opstelde. | | |
| Content | *Content* werd als anoniem object opgesteld omdat een eigen klasse onnodig is. De content omvat de inhoud van een annotatie | | |
|  | Commentary | *Commentary* is de commentaar die door de *op* werd ingegeven | |
|  | MarkType | *MarkType* staat *by default* ingesteld op fluo, maar zou als uitbreiding zelf gekozen kunnen worden Andere voorgestelde mogelijkheden zijn onderlijning en omkadering. | |
|  | Category | *Category* is de soort annotatie die gemaakt werd. Er is momenteel enkel keuze uit *Extra*, *Question* en *Remark*. | |
| ContentTags | *ContentTags* zijn de *tags* die meegegeven worden door de *op* bij creatie. Voorbeelden hiervan zijn “belangrijk”, “opgelost” en “info”. Deze kunnen vrij worden gekozen. | | |
| View | *View* is de zichtbaarheid en kan op *private* en *public* ingesteld worden. | | |
| Cherrypicking | *Cherrypicking* stelt een dynamische favorietenlijst voor. Elke gebruiker die een annotatie wilt bijhouden, maakt ze tot favoriet. Hierdoor bevat *cherrypicking* alle gebruikers die in die specifieke annotatie geïnteresseerd zijn. | | |
| Rating | *Rating* is een anoniem object, stelt het beoordelingssysteem voor van een annotatie en wordt bij aanmaak ingesteld op een lijst met nul *upvotes* en *downvotes*. | | |
|  | ThumbsUp/  ThumbsDown | | *ThumbsUp/ThumbsDown* bevat alle gebruikers die een *upvote/downvote* gaven. |
| Date | *Date* is de datum van plaatsing en wordt automatisch ingevuld. | | |
| Reactions | *Reactions* bevat de reacties van andere gebruikers of de *op* zelf en is oorspronkelijk leeg. Reactions behoren tot de Reaction-klasse. | | |
| Upvote/Downvote | Via *upvote/downvote* worden *upvote*/*downvote* uit *Voter* opgeroepen. | | |
| DeleteReaction | DeleteReaction zorgt voor het verwijderen van reacties uit *reactions* van een annotatie. | | |
| Cherrypick | Via c*herrypick* worden gebruikers in en uit de favorietenlijst *cherrypicking* gezet. | | |

Tabel 2: Leden *Annotation*-klasse

|  |  |  |
| --- | --- | --- |
| Leden van Reaction | Beschrijving | |
| Person | *Person* is de gebruiker die de reactie opstelde. | |
| Date | *Date* is de datum van plaatsing en wordt automatisch ingevuld. | |
| Rating | *Rating* is een anoniem object, stelt het beoordelingssysteem voor van een reactie en wordt bij aanmaak ingesteld op een lijst met nul *upvotes* en *downvotes.* | |
|  | ThumbsUp/  ThumbsDown | *ThumbsUp/ThumbsDown* bevat alle gebruikers die een *upvote/downvote* gaven. |
| Text | *Text* is de reactie die door *person* werd ingegeven. | |
| Upvote/Downvote | Via *upvote/downvote* worden *upvote*/*downvote* uit *Voter* 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 gelet of de gebruiker zijn eigen annotatie of reactie niet probeert te beoordelen.

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

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 ververst. 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 | *AnnotationList* is een lokale lijst met alle annotaties na een laatste *refresh* uit de databank. |
| GetAnnotations | Via *getAnnotations* wordt de *annotationList* opgehaald. |
| AddAnnotation | *AddAnnotation* voegt een nieuwe annotatie toe aan de lokale *annotationList*. De nieuwe annotatie wordt niet in de databank gestoken via deze methode. |
| ClearData | *AnnotationList* kan geleegd worden via *clearData*. |
| FindAnnotationByHtmlId/ FindReactionByHtmlId | Via *findAnnotationByHtmlId*/*findReactionByHtmlId* 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*, *contentTags* en *rating* weergeven. 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. Wanneer de annotatie gemaximaliseerd is, zijn ook de *commentary* en *reactions* zichtbaar. Op elke reactie en annotatie die van de gebruiker zelf afkomstig is, is er een *edit*- en *delete*-knop aanwezig. Ook kan er een reactie bijgeplaatst worden door de gebruiker. Het geminimaliseerde formaat wordt standaard uitgeprint bij weergave van de annotaties.

|  |  |
| --- | --- |
| TagColors | Dit is een lijst die bepaalde kleuren kan *mappen* op een *tag*. |
| DrawAnnotations/ DrawAnnotationsMax/ DrawAnnotationsMin | Alle annotaties die meegegeven worden aan *drawAnnotations* worden uitgetekend in minimale versie *by default*. Wanneer je erna de annotatie wilt maximaliseren/minimaliseren, worden Draw*AnnotationsMax*/ *DrawAnnotationsMin* gebruikt om de reeds uitgetekende html-code aan te passen. Deze methoden roepen *MakeContent*/*MakeContentMax* en *FillTags* op. De maximale versie van een annotatie heeft ook nog *FillReactions* en *MakeNewReaction* 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 *makeNewReaction.* |
| DrawReaction | Nadat een nieuwe reactie opgesteld is en doorgestuurd wordt met een *AJAX-call*, wordt ze uitgetekend in *html* via *drawReaction* en roept ze daarbij *MakeReaction* op. DrawReaction verwacht het annotation-object waarvan de reactie afkomstig is, het reaction-object dat moet worden gevisualiseerd en een html object als target waar de nieuwe html reactie aan moet worden toegevoegd. |
| FillReactions | *fillReactions* splitst de lijst *reactions* op in aparte reaction-objecten en tekent ze uit in *html*-code in de annotatie via *makeReaction*. |
| MakeReaction | *MakeReaction* krijgt een reaction-object mee en zal aan de hand hiervan een html-container teruggeven die dan aan de html pagina kan worden toegevoegd. De tweede parameter is een nummer, dat duid op de hoeveelste reactie dit is op de annotatie en wordt gebruikt als id voor de html-container. |
| MakeContent/ MakeContentMax | Bij een geminimaliseerd formaat (dus ook *default*-versie) wordt de inhoud van de annotatie via *makeContent* opgebouwd, anders wordt *makeContentMax* gebruikt. Deze methode zorgt voor de html opbouw van de binnenste delen van de html-container die een annotatie visualiseerd. |
| FillTags | *FillTags* splitst *contentTags* op in aparte woorden en tekent ze uit in *html*-code in de html annotatie-container, die als *target* wordt meegegeven. |
| SetAnnotationButtons-Style | De gebruiker kan zien welke annotaties of reacties hijzelf al *geupvote* of *gedownvote* heeft of als favoriet beschouwt door inkleuring van de symbolen, duim omhoog, omlaag en een ster, via *setAnnotationButtonsStyle*. Door deze methode worden de laatste wijzigingen onthouden na navigatie. Deze methode wordt opgeroepen als de een annotatie door *maximize* vergroot wordt en hierdoor die symbolen zichtbaar worden. |
| RefreshRatingButtons | Dit is een hulpmethode die gebruikt wordt om bij de duim-omhoog en omlaag symbolen, die in een html-container met css-klasse *rating* geplaatst zijn, het bijhorend getal aan te passen naargelang het aantal gebruikers in de overeenkomstige *rating* lijst van de meegegeven annotatie of reactie. |
| Favorite | Via *favorite* wordt het ster-icoon van kleur veranderd zodat de gebruiker ziet of hij de annotatie aan zijn favorieten toegevoegd heeft en wordt de methode *cherrypick* van de annotatie opgeroepen. |
| AnnotationUpvote/ AnnotationDownvote/ ReactionUpvote/ ReactionDownvote | Deze methodes worden opgeroepen bij het klikken op een duim-icoontje. Via deze methodes worden de kleuren van de duim-icoontjes aangepast en de bijhorende methodes van het annotatie of reactieobject opgeroepen om een annotatie of reactie te beoordelen. Deze methodes verschillen met de *SetAnnotationByttonsStyle* methode, waarbij deze methodes enkel het aangeklikte icoontje zullen aanpassen en de laatstgenoemde methode alle icoontjes in de volledige html 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 *maximize*/*minimize* opgeroepen die de draw-methodes (*drawAnnotationsMax*/*Min*) oproepen om de reeds uitgetekende *html*-code aan te passen. |
| AnnotationEdit/ ReactionEdit | De gebruiker kan steeds zijn eigen annotaties/reacties aanpassen door op de *edit*-knop te klikken in de gemaximaliseerde versie van de annotatie. Deze knop roept *annotationEdit*/*ReactionEdit* op. |
| ReactionDelete/ AnnotationDelete | De gebruiker kan steeds zijn eigen annotaties/reacties verwijderen door op de *delete*-knop te klikken in de gemaximaliseerde versie van de annotatie. Deze knop roept *annotationDelete*/*ReactionDelete* op. |
| ShowOnDia | Wanneer men een annotatie bekijkt in gemaximaliseerd formaat bevat die ook een *show*-knop om het bijhorende element in fluo weer te geven op de webslide of zelfs eerst naar die webslide te springen voor het aangeduid kan worden. Die knop roept *showOnDia* op. |
| DrawError/ DrawReactionError | *DrawError* 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. *DrawReactionError* tekent een icoon met uitroepteken zodat de gebruiker ziet dat er iets verkeerds gebeurde. 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 *drawEmptyAnnotations*. |
| SetToDefault | Wanneer de gebruiker de *sidebar* sluit en terug opent, het formulier afsluit, vanuit een annotatie naar de juiste slide springt of terugkeert naar de *default* *view*, worden via *SetToDefault* 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.

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

|  |  |
| --- | --- |
| Size\_sidebar\_absolute | Er wordt steeds gerekend met de absolute grootte van de *sidebar*. |
| Open\_sidebar | De *sidebar* wordt geopend door op het tekstvaksymbool langst de rechterkant te klikken, dit roept open\_sidebar op. |
| Close\_sidebar | De *sidebar* wordt gesloten door op het kruisje rechtsboven te klikken, dit roept *close\_sidebar* op. |
| Show | *Show* 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 *sidebar* 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 *default view* 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 *Callerklasse* om data te kunnen doorgeven aan de databank.

|  |  |
| --- | --- |
| VoteColor/  FavoriteColor/  IconColor | Voor de *upvotes*, *downvotes*, favorieten en andere iconen in de navigatiebalk werden standaardkleuren voorzien. |
| PresentationId/ GetPresentationId | Na het opvragen van de huidige presentatie via *getPresentationId*, wordt het bijgehouden in *presentationId*. |
| SlideNumber/ GetSlideNumber | Na het opvragen van de huidige *webslide* van de presentatie via *getSlideNumber* wordt het bijgehouden in *slideNumber*. |
| User/GetUser | De ingelogde gebruiker wordt opgevraagd via *getUser* en bijgehouden in *user*. |
| LocalData | Er wordt een instantie van de *LocalData*-klasse bijgehouden waarin een *annotationList* zit met lokale annotaties. |
| Caller | Er wordt een instantie van de *Caller*-klasse bijgehouden om AJAX-calls te doen voor dat *localData*-object. |
| MutationObserver | De *mutationObserver* is een *observer* die wijzigingen in de webpagina en opmaak kan detecteren. Deze variabele verwijst naar de *mutationObserver* van de *window*. |
| MutationObserverConfig | Dit is de configuratie van de *MutationObserver*. Deze zal enkel naar de html-elementen, inclusief hun attributen, kijken waaraan hij is toegevoegd en niet naar hun kindelementen. |
| Observer | In *observer* wordt een nieuwe instantie van de MutationObserver aangemaakt. Deze krijgt de methode *detectActiveSlideChange* 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 *sidebar* opnieuw wordt opgevuld, nu met enkel de annotaties voor de momenteel actieve *slide*. |
| AddSlideChangeMutation- Observer | Deze methode voegt de *MutationObserver* toe aan alle elementen met de css-klasse *slide*. Deze moet bij opstarten van de *sidebar* worden opgeroepen. |
| Main | In de *main* wordt *refreshTable* en *AddSlideChangeMutationObserver* opgeroepen. Deze methode wordt opgeroepen als alle bestanden zijn geladen. |
| RefreshTable | Dit roept steeds *loadAnnotationsFromServer* op om alle annotaties van de server op te halen en zal dan de *sidebar* opnieuw laten tekenen. |
| FixHtml | De gebruiker mag bij het schrijven van een annotatie of reactie geen *html*-code gebruiken, anders zouden er problemen kunnen optreden bij opslag of uitschrijven. De ingevoerde code wordt door *fixHtml* *html*-loos gemaakt. |
| GetAnnotationsById- Overview | Uit een gegeven lijst worden via *getAnnotationsByIdOverview* alle annotaties gehaald die behoren tot een specifieke presentatie. |
| GetAnnotationsById | Uit een gegeven lijst worden via *getAnnotationsById* alle annotaties gehaald die behoren tot een specifieke presentatie en *webslide*. |
| OpenHelpContainer | Dit opent de *helpcontainer* met een beschrijving van elk icoon. |
| CloseHelpContainer | Dit sluit de *helpcontainer* 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*.

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

|  |  |
| --- | --- |
| Select | Dit is de 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 *hooveren*. |
| Selected | Er is al minstens één keer geklikt geweest waardoor *hooveren* 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 de tweede knop in de navigatiebalk. Deze knop roept *openForm* op. |
| Showtag | Er wordt een *button* met *tag* toegevoegd via *showtag*. |
| Deletetag | Er wordt een *button* met *tag* verwijderd via *deleteTag*. |
| Init | *Init* zorgt voor het kleuren van *webslide*-elementen in fluo, het bevestigen van het uiteindelijke element en het hernemen van die keuze. |
| EditStyle | *EditStyle* verandert de stijl van de elementen waarover *gehoovered* wordt naar fluo. Dit zou uitgebreid kunnen worden met onderlijning en omkadering. |
| SubmitAnnotation | Alle velden van het formulier worden ingelezen door *submitAnnotation* 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 toch alle info behalve *title, contentTags*, *category*, *commentary*, *view* en *element* behouden. S*ubmitAnnotationAgain* zorgt voor die bewerking. |
| CancelAnnotation | De gebruiker kan het formulier verlaten via *cancelAnnotation*. Dit zorgt ervoor dat selecteren van elementen wordt afgebroken. |
| PostReaction | Wanneer een reactie gepost wordt in het tekstvak onder de gemaximaliseerde annotatie, wordt *postReaction* opgeroepen bij drukken op de knop. Dit roept *sendReaction* op op *caller* en tekent die reactie uit via *drawReaction*. |
| PostReactionAgain | Als een reactie aangepast wordt, moet enkel de *text* veranderd worden en een *AJAX-call* gestuurd worden met de annotatie om die te overschrijven via *postReactionAgain*. Dit roept *sendReaction* op van de *caller.* |

Tabel 11: Leden *AnnotationForm*-module

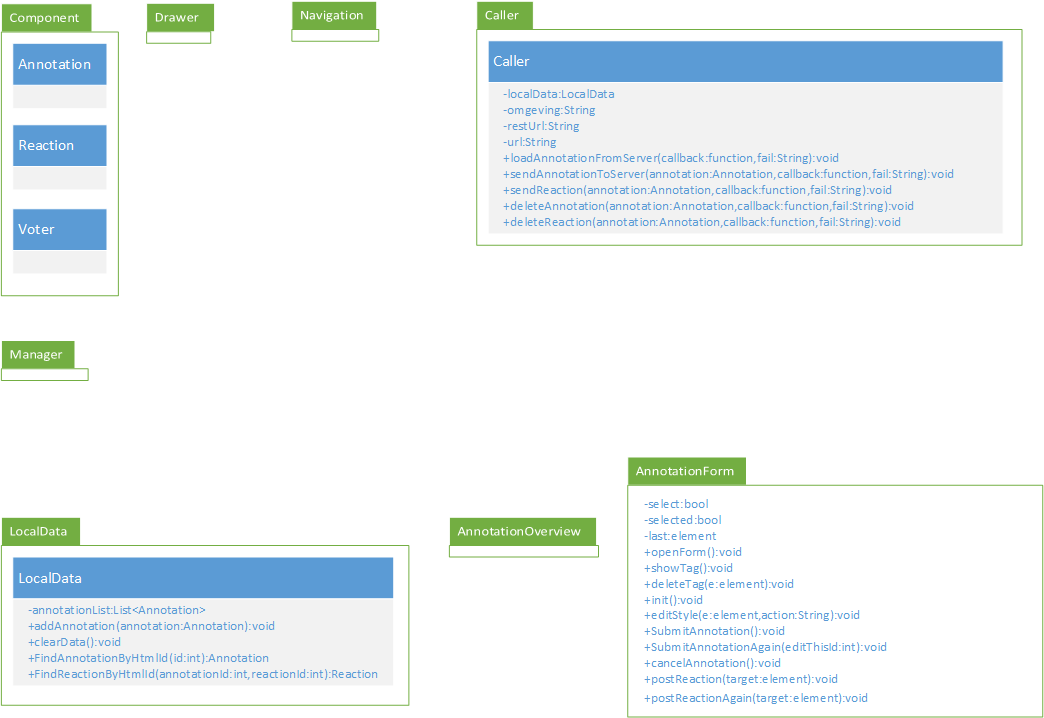
Tussen deze modules en klassen komen verschillende relaties voor.

|  |  |
| --- | --- |
| Aggregation (bezitter en bezit): | |
| Annotation en Reaction | Een annotatie kan leven zonder reacties, maar een reactie hoort altijd bij een annotatie. |
| LocalData en Caller | Een *Caller*-object moet steeds een *LocalData-*instantie meekrijgen in de constructor. |
| Association: Using (gebruikte en gebruiker): | |
| Voter en Annotation, Reaction | Annotaties en reacties gebruiken beiden de statische methode uit *Voter* voor hun beoordelingssysteem. |
| Annotation en AnnotationOverview | In de *overview* worden alle zoekfuncties op annotaties toegepast. |
| Annotation, Reaction, LocalData, Caller en AnnotationForm | Bij het versturen van annotaties en reacties wordt een *LocalData*-object gebruikt om lokale (reeds opgehaalde) data aan te passen en een *caller*-instantie gebruikt voor de *AJAX-calls*. |
| Caller en Manager | Wanneer annotaties en hun reacties ververst worden, wordt de *Caller*-instantie gebruikt voor de *AJAX-call*. |
| Nav en AnnotationForm, Drawer, AnnotationOverview | Bij veranderen van tabblad in de navigatiebalk, moeten het formulier en overzicht over presentatie of specifieke webslide gemaakt kunnen worden. |
| Annotation, Reaction, Caller, LocalData, Manager en Drawer | Als annotaties en reacties gemaakt worden, moeten er *AJAX-calls* gebeuren. Voor het opvragen en uittekenen, wordt de lokale data opgevraagd. Om het kleur van de iconen te achterhalen, wordt de *Manager* geraadpleegd. |

Tabel 12: Relaties tussen klassen en modules

Hieronder, op Figuren 6,7 en 8 werden drie diagrammen bijgevoegd waarin de relaties tussen de verschillende modules en klassen duidelijk gemaakt worden.

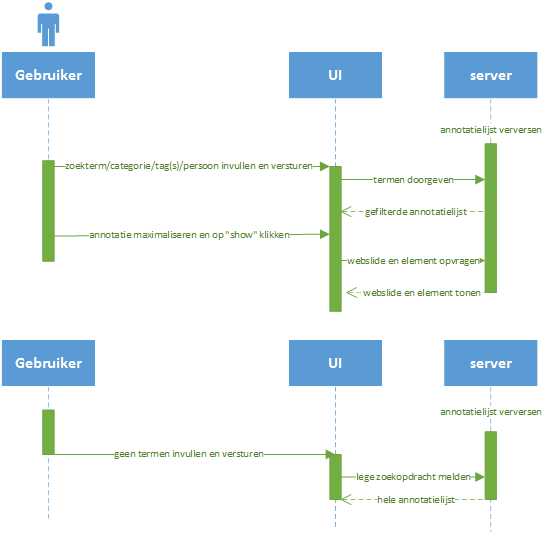
Niet vergeten de aantallen en pijlen te noteren/bij te tekenen op de diagrammen na afdrukken.  
De vraagtekens moeten nog vervangen worden

  
Figuur 6: Klassendiagram met relaties en inhoud van *LocalData*-, *Caller*- en *AnnotationForm*-modules

  
Figuur 7: Klassendiagram met inhoud van *Component*-module

  
Figuur 8: Klassendiagram met inhoud van *Manager*-, *AnnotationOverview*-, *Drawer*- en *Navigation*-modules

3.3 Sequentiediagrammen  
Op Figuur 9 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.

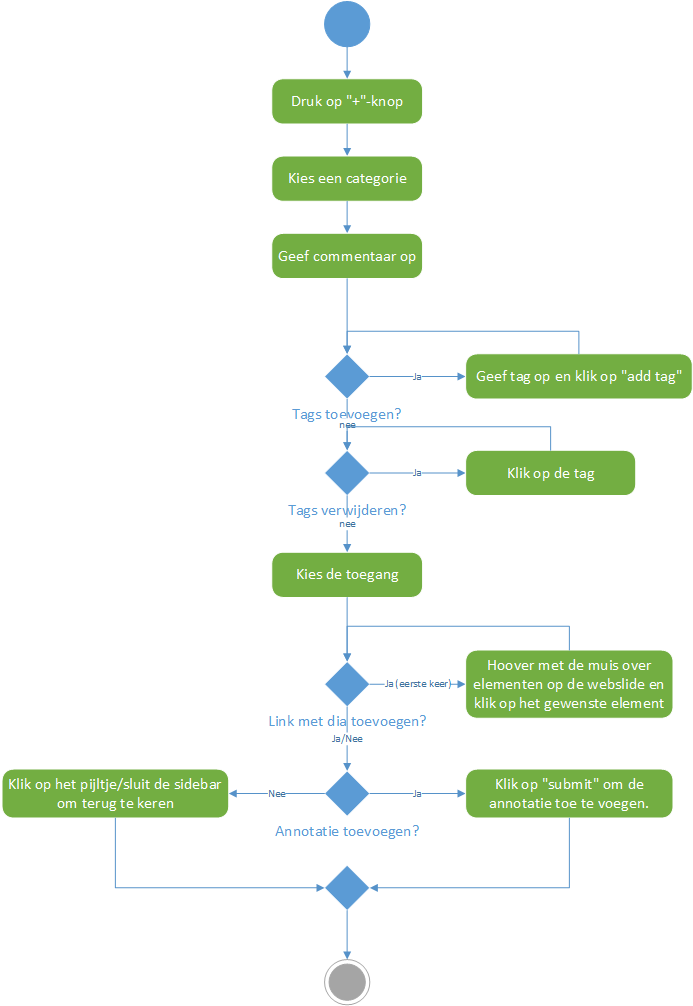
  
Figuur 9: Sequentiediagram *AnnotationOverview*-module

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. Het klassendiagram van de *servlet* bevindt zich op Figuur 10.

Figuur 10: Databank

-annotatieService afgeleid van annotation?  
-servlet op diagram?  
-wie gebruikt annotationService?

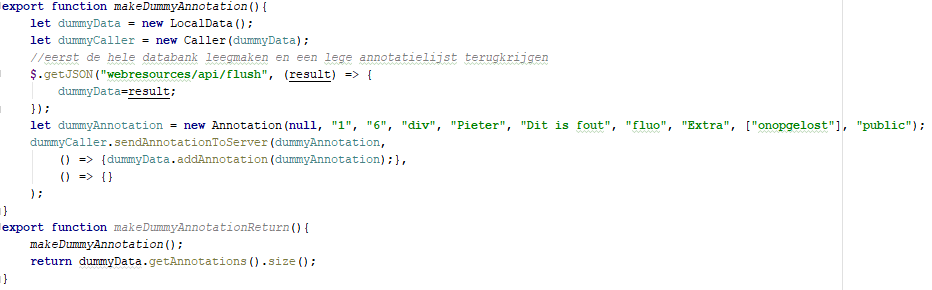
3.5 Activiteitendiagram  
Wanneer de gebruiker een nieuwe annotatie wilt plaatsen, kan dat op het *add*-tabblad (met het “+”-icoon). De enige verplichting is het invullen van de commentaar en titel en indien dit niet gedaan wordt voor het versturen van de annotatie, komt er een foutmelding onder het 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* *hoovert*, 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 11 staat het activiteitendiagram bij het invullen van het formulier.

  
Figuur 11: Activiteitendiagram bij toevoegen van een annotatie

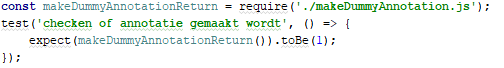
4 Testplan  
Door rekening te houden met de *use cases* werden volgende testen voorzien: toevoegen, verwijderen en aanpassen van een annotatie en een reactie. Naast deze automatische tests, werden er natuurlijk ook manuele voorzien. Hierbij hielden we rekening met de *pop-ups* die verschijnen wanneer de verbinding opeens zou wegvallen, de complexiteit van het eindproduct en de ondersteuning op andere *browsers*. In het testplan in Tabel 13 staan de uitgevoerde testen en hun scenario’s. Na overleg werden *unit* testen niet nodig beschouwd in het project, deze werden vervangen door *usability* en *integration* testen.

4.1 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 2.4 werden de nodige installaties voor het uitvoeren van *jest*-testen overlopen. Nu alles geïnstalleerd is, kan het formaat is 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.

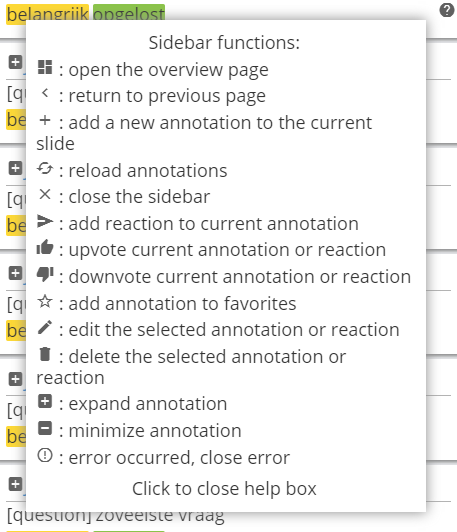
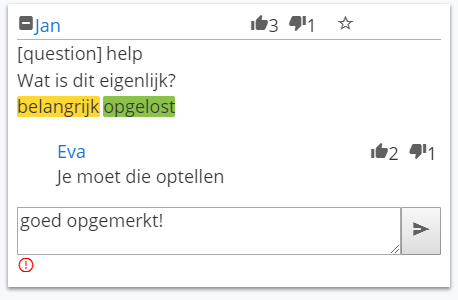
Als voorbeeld werden de bestanden *makeDummyAnnotation.js* en *makeDummyAnnotation-.test.js* gekozen. Het doel van deze test is om een dummy-annotatie aan te maken, te verzenden naar de databank en te checken of deze wel werd toegevoegd. Onderstaande code is afkomstig uit *makeDummyAnnotation.js.*   
Figuur 12: inhoud van makeDummyAnnotation.js

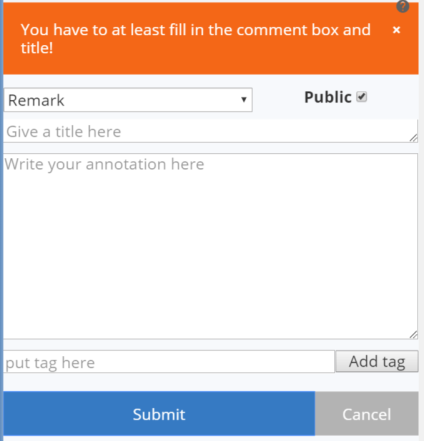
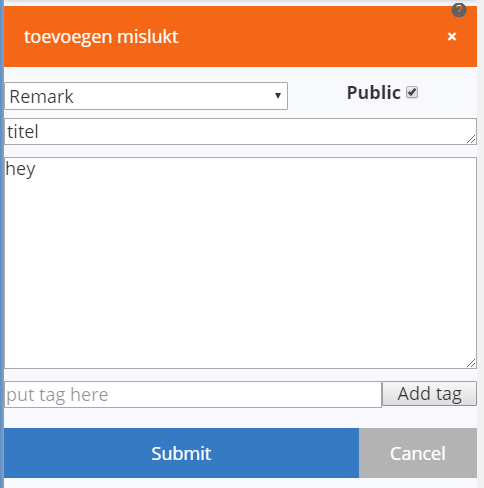
In het *js*-bestand wordt telkens een exporteerbare functie gemaakt met een return-waarde, in dit geval *makeDummyAnnotationResult*. Vaak veronderstellen tests dat bepaalde onderdelen al gecheckt werden in vorige tests zodat er verder op gebouwd kan worden. Indien je de inhoud van een huidige test later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

  
Figuur 13: inhoud van makeDummyAnnotation.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 werd in het project al toegevoegd in “..<..<..<”. Het uitvoeren van de testen kan via onderstaand commando   
<<npm test

**4.2 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 leiden 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, hebben we 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 is er een Help-knop aanwezig die elk icoon een korte beschrijving geeft.

Wanneer de internetverbinding van de gebruiker opeens wegvalt, kan geen data meer doorgegeven of opgehaald worden van de databank. Bij het aanmaken/editeren/verwijderen van een annotatie of reactie zal een wegklikbare melding verschijnen. Bij het *upvoten*, *downvoten* en verkiezen zullen de gemaakte wijzigingen niet doorgaan.

    
Figuur 14: help-container Figuur 15: foutmelding

    
Figuren 16, 17 en 18: foutmeldingen

**4.3 Platform testen**  
Het project werd op verschillende browsers uitgeprobeerd en werkt op de laatste versies van Chrome en Firefox.

|  |  |  |  |
| --- | --- | --- | --- |
| wat | hoe | frequentie | wie |
| 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 |  |  |
| Ondersteuning van browsers | Platform test |  |  |
|  |  |  |  |

Tabel 13: testplan