

Functioneel en technisch designdocument “StoryScore”-tool

Bas Bergervoet (onderwijstechnoloog) en Thomas Peeters (programmeur)

LIMEL - KU Leuven / OOF-project Seeing Sound

September 2014

StoryScore-tool

Concept en doel

De StoryScore-tool stelt animatie- en compositiestudenten van de KHLIM en KU Leuven in staat om samen aan een audiovisueel concept van een animatiefilm te werken. De tool combineert de principes van een storyboard en een partituur in een gebruikersvriendelijke, collaboratieve webapplicatie. Het concept van de StoryScore-tool wordt in dit document beschreven in het kader van het OOF-project ‘Seeing Sound’. In de bijlage vindt u een interactieve mock-up van de tool.

Functioneel ontwerp

Requirementsanalyse

Op basis van voorbeelden van studenten uit het OOF-project en verschillende bijeenkomsten met de projectpartners en experts zijn de volgende functionele requirements vastgelegd:

- De StoryScore-tool is webgebaseerd (cross-platform)
- Er zijn drie gebruikersrollen:
 - Admin (toegang tot alle data, settings en user management)
 - Teacher (heeft toegang tot verschillende versies StoryScore. Kan enkel items bekijken, de StoryScore afspelen en commentaren toevoegen - deze rol wordt pas op het niveau van de StoryScore bepaald)
 - User (maker van de StoryScore, meestal de student)
- Een StoryScore kan gemaakt worden met meerdere users en becommentarieerd worden door meerdere teachers.
- De basis van de StoryScore is een canvas met een tijdslijn.
 - Gebruikers kunnen een achtergrondaafbeelding uploaden om een geheel te suggereren.
 - Resources kunnen vrij op het canvas geplaatst worden, maar er wordt gebruik gemaakt van horizontale tracks (default = 4) om hiërarchieën en emoties te kunnen suggereren

- Resources kunnen ook aan punten in de tijdslijn of andere resources worden 'gesnapt' (uitgelijnd op een raster).
- De tijdslijn is afspeelbaar: een tijdslijn speelt horizontaal de verschillende losse elementen op de tijdslijn af (soms ook tegelijkertijd)
- Gebruikers moeten media kunnen toevoegen aan de StoryScore, zowel door deze zelf te uploaden als via verschillende cloud services.
 - Afbeeldingen
 - Video
 - Audio
- Gebruikers moeten annotaties kunnen maken op de canvas (vrij van de tijdslijn en zonder 'snapping'):
 - Tekst
 - Vormen
 - Vrij tekenen
 - De zichtbaarheid van annotaties moeten aan- en uitgeschakeld kunnen worden
- Gebruikers moeten commentaar kunnen geven op zowel de hele canvas als per asset op de canvas. Hierbij moet duidelijk zichtbaar zijn welke commentaren van de docent komen. De zichtbaarheid van commentaren op het canvas moeten kunnen worden aan- en uitgezet.
- Zowel resources als de hele StoryScore hebben een versiesysteem om het leerproces vast te kunnen leggen
 - Versioning per resource gebeurt door binnen de resource steeds een nieuwe versie te uploaden
 - Versioning per StoryScore gebeurt doordat de gebruiker actief de huidige StoryScore als 'versie X' opslaat (de laatste versie wordt sowieso automatisch opgeslagen)

Use case scenario's

De belangrijkste schermen en functies worden hieronder in verschillende use case scenario's behandeld. Een uitgebreidere interactieve pdf-versie van de functionele design-mock-up vindt u in de bijlage.

1. User management
2. Nieuwe StoryScore aanmaken
3. Co-auteurschap aanvaarden
4. Resources uploaden en toevoegen aan de storyscore
5. Het trimmen van een resource
6. Afspelen van de tijdslijn van het canvas
7. Commentaar toevoegen aan een resource
8. Commentaar toevoegen op het canvas
9. Annotatie toevoegen aan het canvas
10. Annotaties en/of commentaar toggelen
11. Nieuwe versie aan een resource toevoegen
12. Nieuwe versie van de StoryScore opslaan

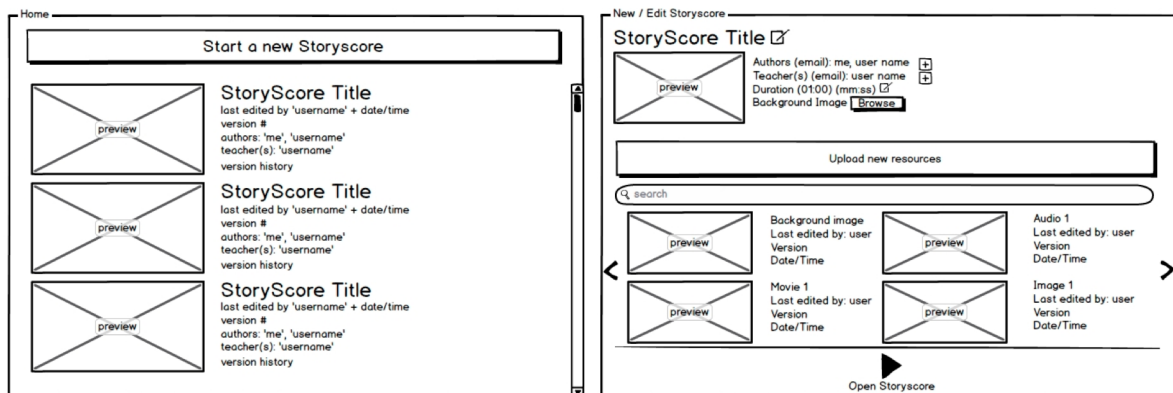
Use case scenario 1 (UCS1)

Use-case naam	User management
Actor(en)	Admin
Beschrijving	het aanmaken, aanpassen en verwijderen van user-accounts en het raadplegen van bijbehorende StoryScores en resources
Aanleiding	Beheer van de applicatie
Voorwaarden	admin heeft toegang
Workflow	Admin gaat naar adminpagina Raadpleegt lijst met users Raadpleegt per user lijst met storyscores en resources verbonden aan user door op username te klikken
Alternatieve workflow(s)	Vanaf: raadpleegt lijst met users • Maakt nieuwe users aan • Verwijdert user(s) • Past user-rol aan (admin of niet)

Van deze use case is geen screen mock-up gemaakt

Use case scenario 2 (UCS2)

Use-case naam	Nieuwe StoryScore aanmaken
Actor(en)	User
Beschrijving	Het aanmaken van een nieuwe StoryScore
Aanleiding	Opdracht om StoryScore te maken of vrijwillig
Voorwaarden	User heeft een account en is ingelogd
Workflow	User klikt op 'add new StoryScore' User landt op StoryScore infopagina User creert StoryScore titel User nodigt co-auteurs uit User nodigt teachers uit User voegt background image toe User upload resources (zie UCS4) User klikt op 'open StoryScore'
Alternatieve workflow(s)	User upload nog geen resources User nodigt nog geen co-auteurs of teachers uit



Home screen (links) en StoryScore infoscreen (rechts)

Use case scenario 3 (UCS3)

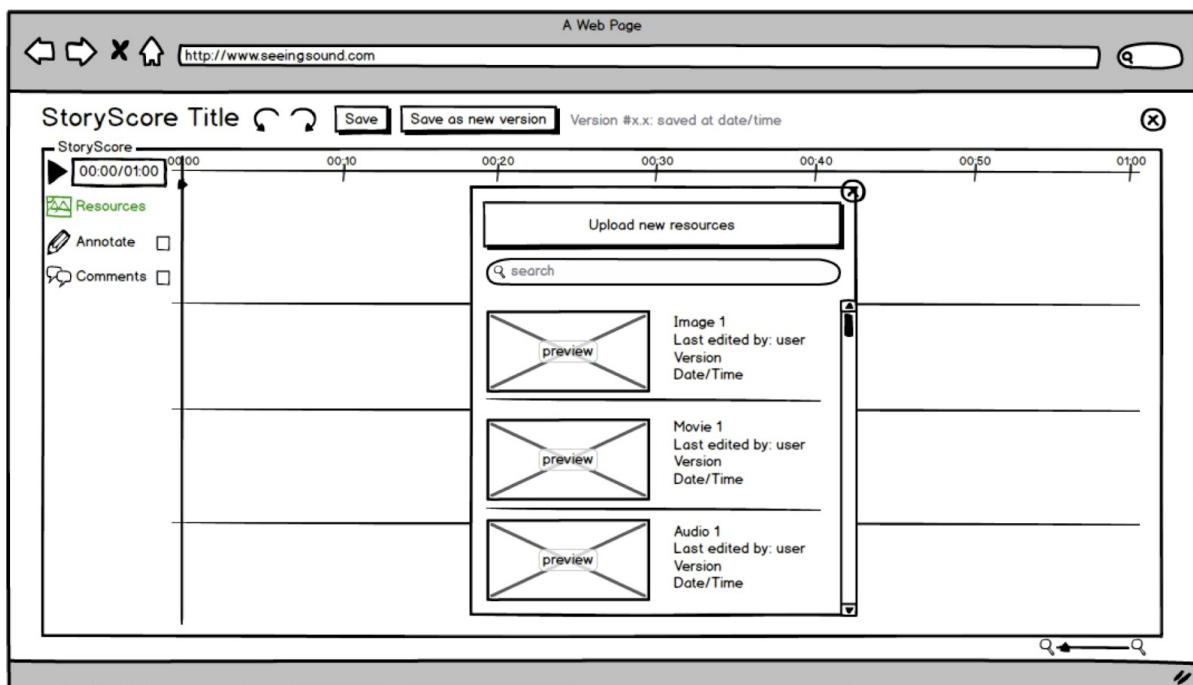
Use-case naam	Co-auteurschap aanvaarden
Actor(en)	User
Beschrijving	StoryScores kunnen met meerdere auteurs worden gemaakt. Een gebruiker kan worden uitgenodigd om mee te werken aan een nieuwe of bestaande StoryScore.
Aanleiding	Uitnodiging om mee te werken aan een StoryScore
Voorwaarden	User heeft een e-mailadres (optioneel: user heeft een StoryScore-account) User werd door andere user uitgenodigd (UC2)
Workflow	User ontvangt e-mail met uitnodiging User logt in of maakt account aan User kan StoryScore waarvoor hij/zij is uitgenodigd aanklikken, bekijken en bewerken als volwaardige auteur
Alternatieve workflow(s)	Admin voegt user toe aan StoryScore Teacher-rol wordt aanvaard (door bestaande of nieuwe user)

Van deze use case is geen screen mock-up gemaakt

Use case scenario 4 (UCS4)

Use-case naam	Resources uploaden en toevoegen aan StoryScore
Actor(en)	User
Beschrijving	Het uploaden van media naar de StoryScore
Aanleiding	Opdracht om StoryScore te maken of vrijwillig
Voorwaarden	User heeft een account en is ingelogd User heeft een StoryScore aangemaakt (USC2)
Workflow	User klikt op 'Resources'

	User klikt op 'Upload new resource' User selecteert 1 of meerdere bestanden Systeem geeft informatie wanneer uploads compleet zijn Uploads verschijnen in Resources-lijst User kan Resources gebruiken in StoryScore door een item in de lijst aan te klikken of op het canvas te slepen
Alternatieve workflow(s)	User drag&dropt resources op canvas Resources uploaden in StoryScore infopagina (zie USC2)

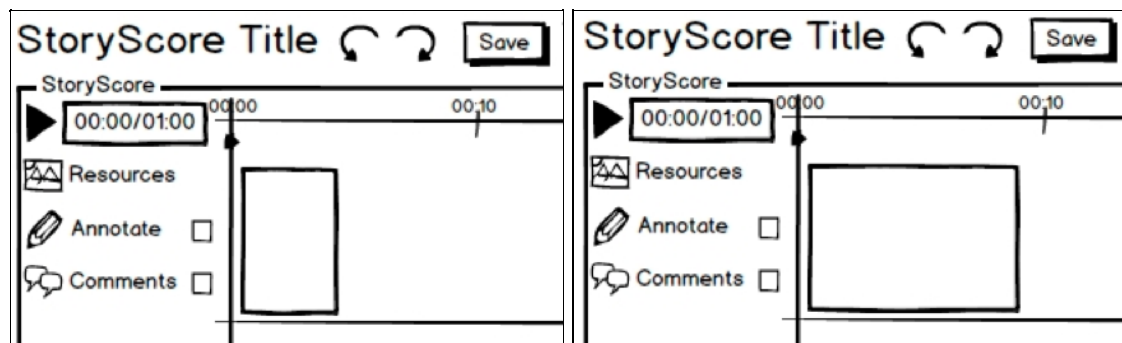


StoryScore met resourcelijst en uploadknop

Use case scenario 5(UCS5)

Use-case naam	Het trimmen van een resource
Actor(en)	User
Beschrijving	De lengte van een resource kan direct op de tijdslijn worden aangepast.
Aanleiding	Een resource moet langer of korter verschijnen/klinken op de tijdslijn (artistieke keuze)
Voorwaarden	User heeft een account en is ingelogd User heeft toegang tot StoryScore User heeft een resource geupload User heeft een resource op het canvas geplaatst
Workflow	User hooft over de zijkant van een resource tot het muisicoon verandert User trekt het uiteinde van de resource naar de gewenste locatie

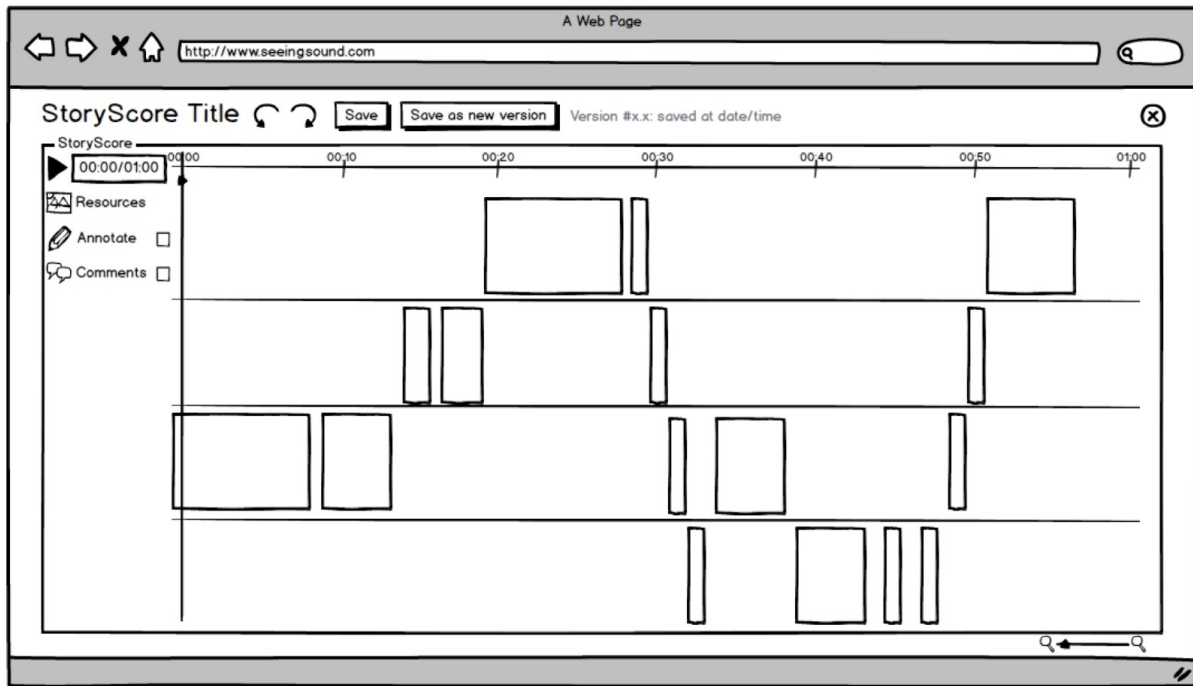
Alternatieve workflow	Bij audio en video: User klikt op resource en komt automatisch in preview tabblad User klikt op 'trim resource' User zet in en outpoint User klikt op OK Lengte van de resource past zich automatisch aan op het canvas (omliggende resource schuiven op in de tijd)
-----------------------	--



Resource op canvas StoryScore (links) en getrimde resource (rechts)

Use case scenario 6 (UCS6)

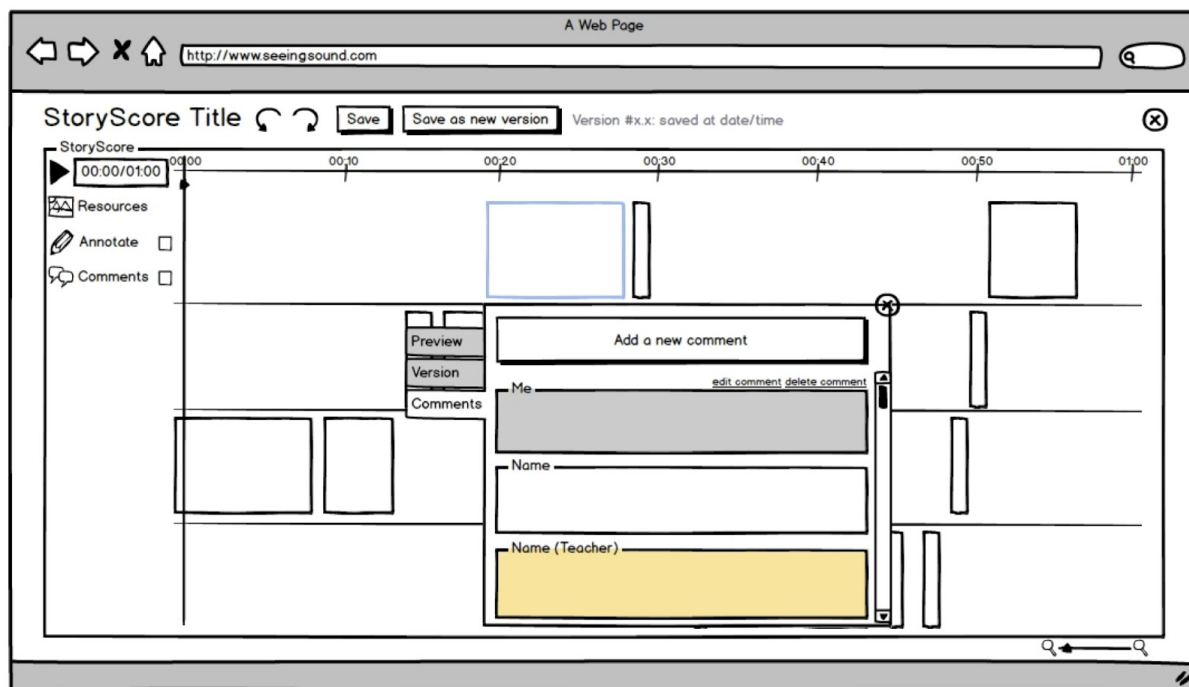
Use-case naam	Afspelen van de tijdslijn op het canvas
Actor(en)	User, Teacher
Beschrijving	De resources op het canvas kunnen worden afgespeeld in een tijdslijn.
Aanleiding	Resultaat van huidige status/versie StoryScore bekijken
Voorwaarden	User heeft een account en is ingelogd Teacher heeft een account en is ingelogd User heeft toegang tot StoryScore Teacher heeft toegang tot StoryScore
Workflow	User/Teacher klikt op play-knop User/Teacher klikt op pauzeknop
Alternatieve workflow	User/Teacher klikt tijdens afspelen op punt in de tijdslijn



Gevulde StoryScore met afspeelknop (linksboven)

Use case scenario 7 (UCS7)

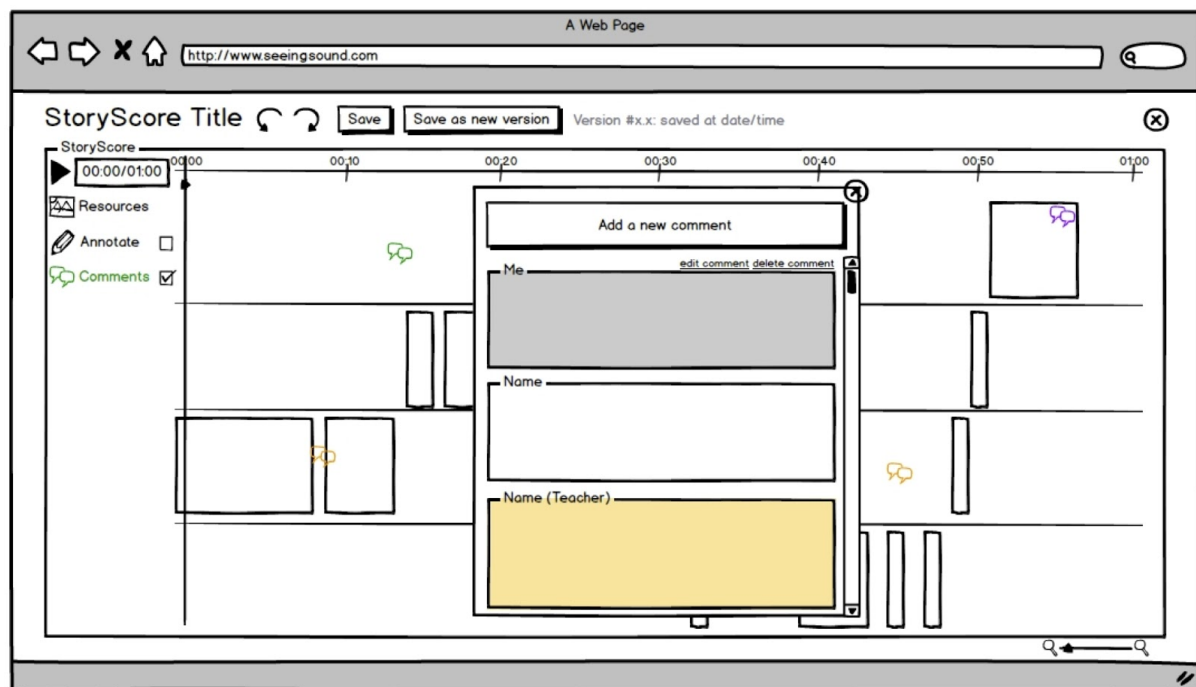
Use-case naam	Commentaar toevoegen aan een resource
Actor(en)	User, Teacher
Beschrijving	Aan elke resource kunnen commentaren worden toegevoegd. Commentaren van teachers zijn duidelijk gekenmerkt.
Aanleiding	Feedback geven op een resource
Voorwaarden	User heeft een account en is ingelogd Teacher heeft een account en is ingelogd User heeft toegang tot StoryScore Teacher heeft toegang tot StoryScore
Workflow	User/Teacher klikt op resource op canvas User/Teacher klikt op comments-tab in de resource-preview lightbox User/Teacher klikt op 'add new comment' User/Teacher typt commentaar in User/Teacher klikt op 'submit'
Alternatieve workflow	User/Teacher klikt direct op commentaar-icoon binnen het clipkader van de resource op het canvas



Geselecteerde resource op canvas met open comments tabblad

Use case scenario 8 (UCS8)

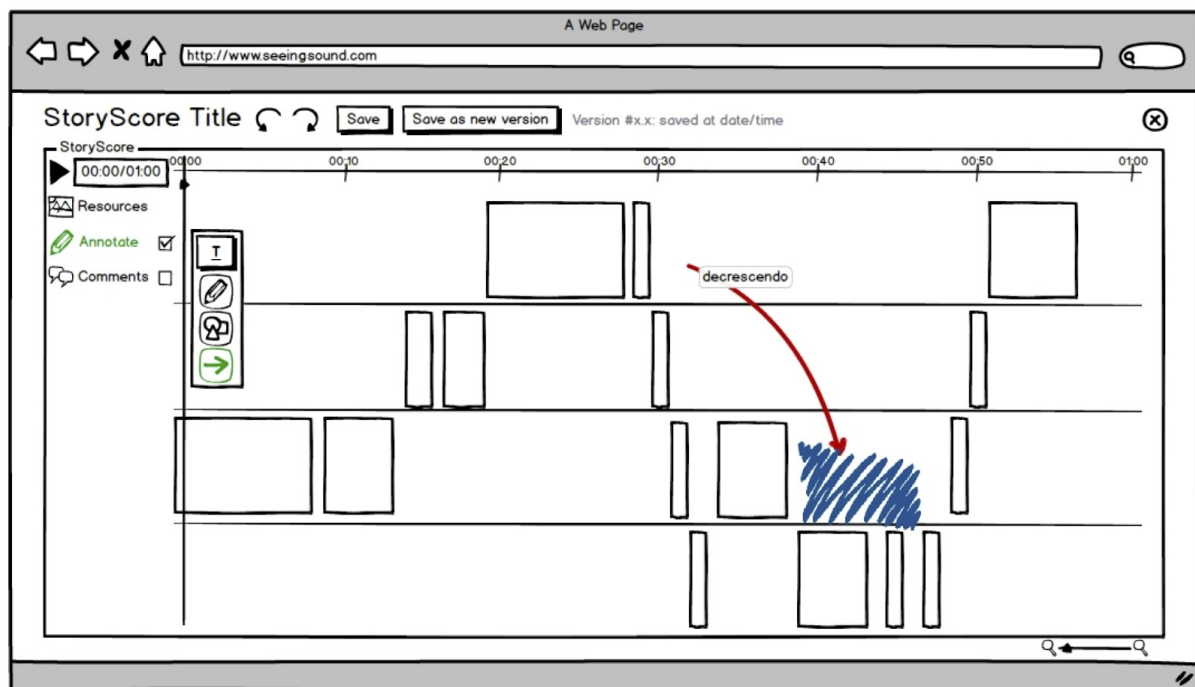
Use-case naam	Commentaar toevoegen aan het canvas
Actor(en)	User, Teacher
Beschrijving	Commentaren kunnen op elk gewenst punt op het canvas worden toegevoegd. Commentaren van teachers zijn duidelijk gekenmerkt.
Aanleiding	Feedback geven op het geheel of de samenhang van de StoryScore
Voorwaarden	User heeft een account en is ingelogd Teacher heeft een account en is ingelogd User heeft toegang tot StoryScore Teacher heeft toegang tot StoryScore
Workflow	User/Teacher klikt op 'comments' in navigatie User/Teacher dubbelklikt op canvas User/Teacher klikt op 'add new comment' in comment-lightbox User/Teacher typt commentaar in User/Teacher klikt op 'submit'
Alternatieve workflow	User/Teacher klikt direct op commentaar-icoon op het canvas



Verschillende comments op het canvas met comment thread overlay

Use case scenario 9 (UCS9)

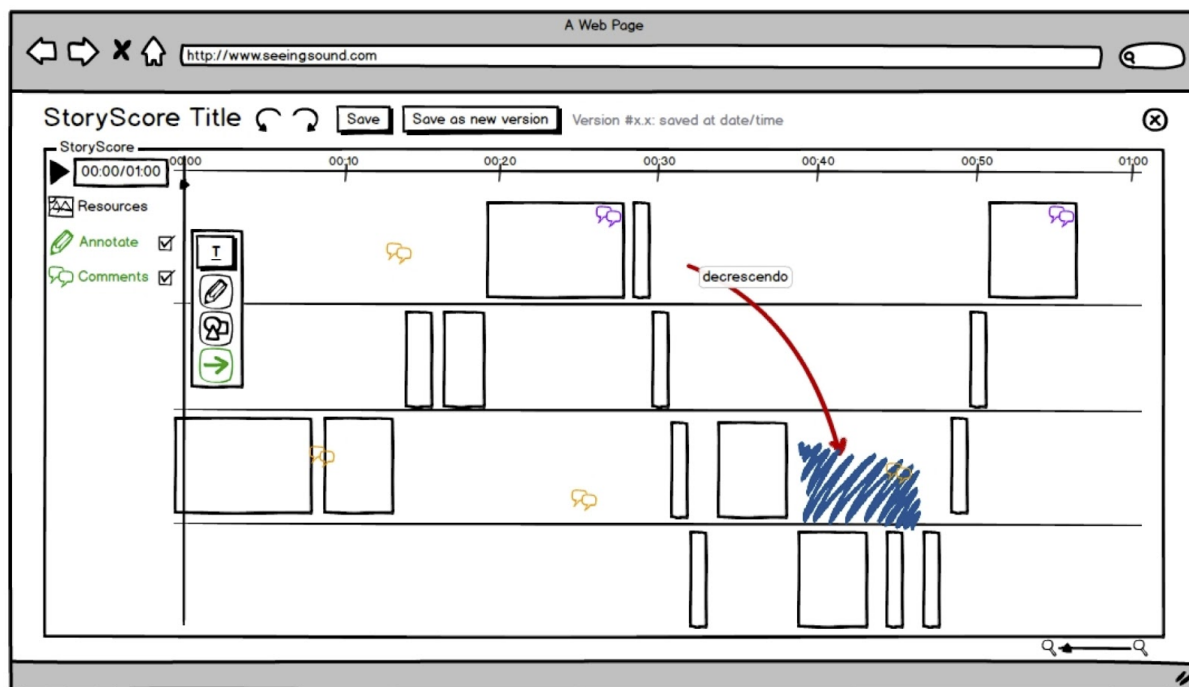
Use-case naam	Annotatie toevoegen aan het canvas
Actor(en)	User
Beschrijving	Annotaties kunnen op elk gewenst punt op het canvas worden toegevoegd.
Aanleiding	Aantekeningen maken, samenhang weergeven, emoties/sfeer uitdrukken, etc.
Voorwaarden	User heeft een account en is ingelogd User heeft toegang tot StoryScore
Workflow	User klikt op 'annotate' in navigatie User kiest annotatietype (tekst, vorm, free draw) User voegt annotatie toe op punt of gedeelte van het canvas



StoryScore met annotaties

Use case scenario 10 (UCS10)

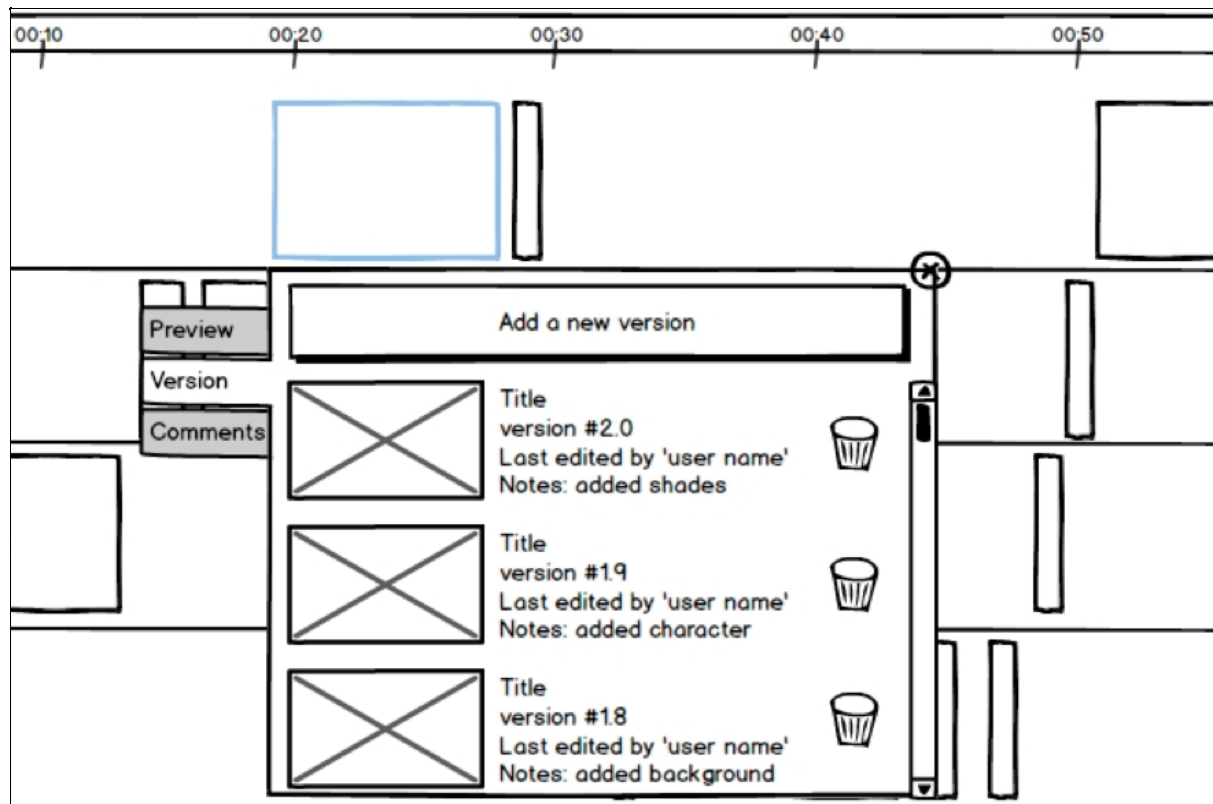
Use-case naam	Annotaties en/of commentaar toggelen
Actor(en)	User, Teacher
Beschrijving	Annotaties en commentaren kunnen worden aan- en uitgezet
Aanleiding	StoryScore met of zonder commentaar en/of annotaties willen bekijken
Voorwaarden	User heeft een account en is ingelogd Teacher heeft een account en is ingelogd User heeft toegang tot StoryScore Teacher heeft toegang tot StoryScore
Workflow	User/Teacher vinkt comments of annotate aan of uit in navigatie



StoryScore met annotaties en comments 'aan' (zie vinkjes in navigatie links)

Use case scenario 11 (UCS11)

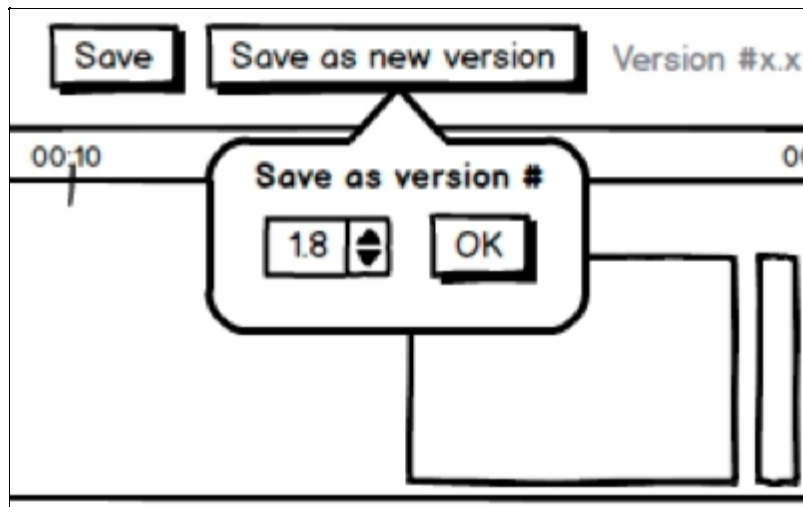
Use-case naam	Nieuwe versie aan een resource toevoegen
Actor(en)	User
Beschrijving	Van elke resource op het canvas kunnen meerdere versies worden geupload. De versies worden bijgehouden en 1 van de versies kan op het canvas verschijnen. Door de versies kan de vooruitgang van de StoryScore en het leerproces worden bijgehouden.
Aanleiding	Er is een nieuwe versie van een resource gemaakt die in de StoryScore moet verschijnen
Voorwaarden	User heeft een account en is ingelogd User heeft toegang tot StoryScore User heeft een resource geupload User heeft een resource op het canvas geplaatst
Workflow	User selecteert resource User klikt op tabblad 'version' User klikt op 'add new version' User selecteert en upload bestand vanaf computer



Geselecteerde resource op canvas met 'version' tabblad open en 'add new version'-knop

Use case scenario 12 (UCS12)

Use-case naam	Nieuwe versie van de StoryScore opslaan
Actor(en)	User
Beschrijving	Naast de autosave (laatste versie die automatisch toegankelijk is) kunnen users actief beslissen om een bepaalde versie van de StoryScore als zodanig op te slaan. Door het bijhouden van verschillende versies kan de vooruitgang van de StoryScore en het leerproces worden geëxpliciteerd.
Aanleiding	Een nieuwe mijlpaal in het maken van de StoryScore is bereikt, of een deadline om een versie af te hebben nadert.
Voorwaarden	User heeft een account en is ingelogd User heeft toegang tot StoryScore
Workflow	User klikt op knop 'sla op als versie' User voert versienummer in User klikt op submit en krijgt systeemfeedback (Versies verschijnen op de infopagina van de StoryScore)



Aangeklikte 'save as new version'-knop met dialoog

Software-architectuur

Servicegeoriënteerde architectuur

Wat?

De fundamentele processen waarvan de applicatie gebruik maakt worden hier opgedeeld in losstaande modules, genaamd services. De communicatie tussen de services en de webapplicatie gebeurt via 1 centrale module, de Enterprise Service Bus (ESB).

De modules zijn opgebouwd in Node.js en de services gebruiken web sockets voor hun communicatie met de ESB. Dit creëert een snelle, flexibele, event-driven architectuur. Aangezien Node.js gebruik maakt van Javascript, gebeurt alle communicatie via JSON.

De meeste services gebruiken een simpele, niet-relatieve MongoDB database om hun eigen documenten in op te slaan.

Er zijn 5 services:

1. **Database Service**
Doet de communicatie met de databank en voert alle CRUD (Create Read Update Delete) operaties uit. Dit voor zowel de MySQL als de MongoDB databases.
2. **Security Service**
Genereert en valideert tokens, zoals de UploadTokens en ApiKeys. Een UploadToken geeft toegang tot de Upload Service File Transport en een ApiKey autoriseert gebruik van de web API.
3. **User Provider Service**
Beheert generieke gebruikersgegevens en legt de brug met een eventuele Identity Provider (zoals bijvoorbeeld Facebook of Shibboleth). De gegevens worden als een User object opgeslagen en/of verzonden.

4. Upload Service

Opent een directe web socket connectie met de client, de File Transport, die geautoriseerd wordt aan de hand van een UploadToken. De service dient daarom ook een client-side Javascript te hebben om deze connectie leggen en de upload van de bestanden in goede banen te leiden.

Voor elke upload wordt een FileUpload-document aangemaakt dat de status en gegevens over het bestand bijhoudt.

5. File Manager Service

Voert taken uit zoals het verwijderen en verplaatsen van bestanden en beheert de bestanden in MongoDB File-documenten.

Een zesde “Video Compression Service” is mogelijk, indien geüploade media naar een streamingformaat moet omgezet worden. Hiervoor is het het gemakkelijkst om een cloud service te hanteren.

De ESB staat via een web socket in contact met de webapplicatie zodat deze requests kan maken en JSON responses kan opvangen.

Waarom?

We hebben gekozen voor een servicegeoriënteerde architectuur (SOA) met het oog op een eventuele toekomstige integratie met andere software of webapplicaties. Een SOA geeft meer flexibiliteit aan de applicatie, vooral op het vlak van herbruikbaarheid en compatibiliteit met andere systemen. De verschillende services zijn abstract en niet alleen inzetbaar in meerdere projecten tegelijk, maar klikken ook makkelijk in op systemen van derde partijen. Daarnaast is schaalbaarheid en uitbreidbaarheid ook makkelijker wanneer elke component apart in een eigen, gerichte omgeving geplaatst kan worden.

Hoe?

Op SOA diagram 1 is te zien hoe een mogelijke opstelling gemaakt kan worden. De servers waarop de services draaien zijn vrij te kiezen, vanwege de flexibele architectuur die de web sockets mogelijk maken. Het diagram is een voorbeeld van een praktische groepering, rekening houdende met de nodige poorten per service en gangbare firewall-instellingen.

Ter illustratie volgt hieronder meer uitleg in de vorm van een voorbeeldscenario van de samenwerking tussen verschillende services. De delen die door het systeem worden uitgevoerd staan in het vet.

Voorbeeldscenario

Probleem	User wilt een nieuwe resource toevoegen aan een StoryScore
Oplossing	Open de webpagina om de StoryScore te bewerken en uploadt een nieuwe resource. Opmerking: het gaat hier om een HTML5, chunked uploadsysteem.
Workflow	1. Navigeer naar de webpagina om de StoryScore te bewerken 2. Klik links in het menu op “Resources”. Een dialoogvenster opent waarin de user een nieuwe resource kan uploaden.

	a. De webapplicatie vraagt een nieuwe UploadToken aan op basis van de ingeloge gebruikersgegevens b. Upload client maakt verbinding met Upload Service File Transport en stuurt de UploadToken mee c. De Upload Service valideert de UploadToken en autoriseert de verbinding d. De upload client krijgt het groene licht en voert een door de webpagina gedefinieerde functie uit e. De webpagina activeert de knop "Upload new resource" 3. Klik in het nieuwe dialoogvenster op de knop "Upload new resource" 4. Kies het bestand om te uploaden a. Een referentie naar gekozen bestand wordt door de upload client in het geheugen bijgehouden. b. De bestandsinformatie wordt langs de web socket in JSON-formaat naar de Upload Service gestuurd. c. De Upload Service maakt hier een FileUpload document van en stuurt een JSON request via de ESB naar de Database Service om het document op te slaan in de collectie fileuploads in de MongoDB database. Om het antwoord op te vangen genereert de service een unieke code - de promiseID - en koppelt hij deze aan een callback-functie. Deze unieke code zal doorheen de workflow steeds meegenomen worden. Dit systeem laat toe om asynchroon te werken. d. De ESB valideert het formaat van de request volgens de JSON Schema standaard. e. Na validatie leest de ESB de request uit en stuurt deze door naar de Database Service. Maar voor hij dat doet, houdt hij de promiseID en afzender bij. f. De Database Service ontvangt de request, leest de instructies uit en slaat het document op. g. Omdat er een promiseID aan de request plakt, stuurt de Database Service het resultaat van de operatie terug naar de ESB. h. De ESB ontvangt deze response, valideert die tegen een JSON Schema en leest de promiseID uit. i. Als de promiseID gevonden wordt in het geheugen, wordt de response naar de originele afzender gestuurd, in dit geval de Upload Service. j. De Upload Service ontvangt de response en voert de callback-functie uit die aan de promiseID gekoppeld is, met de response-data als argument. k. De Upload Service voegt de FileUpload toe aan zijn wachtrij en stuurt een "file queued" event door naar de upload client. l. De upload client ontvangt de boodschap en voegt het bestand toe aan zijn overzicht
--	--

	m. De upload client stuurt een request om de wachtrij te beginnen uploaden n. De Upload Service vraagt de eerste chunk van het bestand o. De upload client leest de gevraagde chunk uit en stuurt deze door als een Base64 encoded string (of andere cross-browser encoding) p. De Upload Service vraagt de volgende chunk. q. De upload client updatet de interface en stuurt de volgende chunk door. r. Zo gaat het voort tot het einde van het bestand. s. Na het succesvol uploaden van het bestand wordt de nieuwe resource in de interface getoond. 5. De gebruiker kan nu zijn nieuwe resource gebruiken.
--	---

Webapplicatie

Back-end

Dit is het fundament van de webapplicatie en bepaalt de gebruikte programmeertaal en software-architecturale principes.

Symfony PHP framework

Wat?

Het Symfony PHP framework biedt een open-source fundament voor ontwikkelaars om op verder te bouwen en maakt het makkelijk om eenvormige code te schrijven binnen eenzelfde methodologie. Het maakt gebruik van het Model-View-Controller (MVC) software-architecturaal principe.

Waarom?

Er zijn verschillende componenten beschikbaar om bv. templating, routing, security en gebruikersbeheer te doen, maar Symfony behoudt ook de flexibiliteit om eigen componenten te maken, bestaande componenten aan te passen en/of uit te breiden en om open-source componenten te implementeren.

Deze componenten zijn – mits basiskennis van Symfony - makkelijk te gebruiken en te onderhouden (bv. via de Composer PHP dependency manager). Symfony biedt zelf ook uitgebreide richtlijnen voor het structureren en onderhouden van deze componenten.

Enkele bijgeleverde componenten en functionaliteiten zijn o.a.:

- o Routing: voor het manipuleren van http requests en het proper opbouwen van URL's. Het geeft de applicatie structuur zowel intern als extern.
Het routing systeem van symfony is ook ideaal voor het opbouwen van de web API.
- o Twig: voor het aanmaken van intelligente templates. Zo kan je pagina's opdelen in contextgevoelige, herbruikbare modules.

- Security: neemt de infrastructuur voor beveiliging van de webapplicatie en autorisatie van de gebruikers voor zich.

Symfony heeft ook een groot draagvlak vanwege de achterliggende open-source community en het gebruik van bepaalde kerncomponenten van Symfony in enkele grootschalige projecten, waaronder Drupal en phpBB. Dit verzekert de duurzaamheid van de technologie.

MySQL-database

Wat?

Een open-source relationele database manager.

Waarom?

Het is een van de standaardtechnologieën voor relationele databases in webontwikkeling.

Hoe?

Database diagram 1 geeft meer inzicht in de opbouw van de relationele database.

Wat uitleg bij de belangrijkste klassen:

- StoryScore: een StoryScore met unieke id
 - StoryScoreVersion: een versie van een StoryScore, met een foreign key naar de StoryScore. In de StoryScoreVersion wordt de meeste informatie over de StoryScore bijgehouden. dit laat toe om terug te gaan naar een vorige versie.
 - StoryScoreAuthor: de gebruikers die toegang hebben tot een StoryScoreVersion met hun rol in de StoryScoreVersion (zie Role klasse).
 - StoryScoreComment: opmerkingen op een StoryScoreVersion.
 - StoryScoreAnnotation: annotaties die bij de StoryScoreVersion horen. Dit is een abstracte klasse die steeds geïntanceerd moet worden als een van de subklassen (zie diagram).
 - StoryScoreClip: bevat een referentie naar een ResourceVersion en waar deze zich moet bevinden binnen de StoryScore
- User: houdt applicatie-specifieke gebruikersdata bij, naast een referentie naar een User-object (User Provider Service). Wordt vooral gebruikt als foreign key in andere klassen.
- Resource: een resource die bruikbaar is in een StoryScore. Het achterliggende bestand is een referentie naar een file document van de File Manager Service.
 - ResourceVersion: hetzelfde systeem als een StoryScoreVersion. Laat toe om verschillende versies op te slaan van een Resource.
 - ResourceComment: opmerkingen op een ResourceVersion.
- ApplicationSettings: simpele klasse om globale instellingen bij te houden in een key-value paar. Enkele voorbeelden staan vermeld op het diagram.

Front-end

De front-end technologieën zijn vooral libraries en tools om interface-ontwikkeling te stroomlijnen.

Angular.js

Deze library laat het toe om zeer makkelijk dynamische interfaces te creëren. Het minimaliseert de hoeveelheid code die geschreven moet worden en maakt een propere scheiding tussen layout en functionaliteit mogelijk zonder te moeten inboeten in mogelijkheden.

JQuery & JQuery UI

JQuery UI bevat verschillende interface-componenten en -functionaliteit, waaronder functionaliteit die toelaat om elementen automatisch uit te lijnen op een raster.

Bootstrap 3

Bootstrap is een verzameling interface elementen - zoals controls, menu's, formulieren, etc. - en design best practices die een uniforme huisstijl volgen. De basishuisstijl is makkelijk te overschrijven met een eigen stijl en werkt als een handige richtlijn.

Naast het volledige Bootstrap pakket is het mogelijk enkel de Bootstrap CSS te gebruiken, of enkel de Javascript. Bij het gebruik van de Javascript library is ook het gebruik van JQuery verplicht.

Font Awesome

Bootstrap 3 bevat een collectie icoontjes (in de vorm van een web font) die vrij te gebruiken zijn, maar is eerder beperkt voor gespecialiseerde projecten. Font Awesome kan als vervanging en uitbreiding dienen moest de standaardset van Bootstrap 3 tekortkomen.

Video.js

Video.js is een gratis open source mediaspeler voor de browser. Deze kan automatisch wisselen tussen HTML5 en Flash, al naargelang de beschikbare technologie op de client.

Browser, apparaten en resolutie

De applicatie is gemaakt met gebruik van een desktop- of laptopcomputer in het achterhoofd. De resolutie zou moeten worden vastgelegd op wat op dit moment gangbaar is in browsers (bijvoorbeeld 1280x800). We bevelen aan om de applicatie in Google Chrome te gebruiken.