

Deze project-eisen zijn tot stand gekomen in een [telefoongesprek](#) met de opdrachtgever. In deze map bevindt zich een pdf met daarin een transcriptie van dit gesprek. Ik heb hier voornamelijk de requirements van de opdrachtgever gedocumenteerd. ([4. Requirements v1.1 10-10-19 \[Herleid uit telefoongesprek1\].pdf](#)) (Verder bestaat deze vastgestelde requirements ook uit de requirements van [de opdracht zelf](#).)

Dat deze gewenste requirements door de opdrachtgever zijn opgesteld betekende niet dat er aan alles voldaan kon worden. De haalbaarheidsstudie bood hierbij inzicht of het behalen van deze projecteisen (binnen de gegeven tijd) wel mogelijk waren. De requirements zijn getest op: meetbaarheid, acceptatie, realisme en tijd. Ik heb de requirements vervolgens helemaal SMART uitgewerkt omdat dit als verbeterpunt werd gevraagd voor deze beoordeling.

Na de haalbaarheidsstudie kon ik een aantal conclusies trekken en ben ik weer in contact gekomen met de opdrachtgever met een aantal requirements-wijzigingen.

Uiteindelijk heeft de opdrachtgever akkoord gegeven voor deze lijst met requirements. En dat is deze lijst hieronder.

*Het kan zijn dat tijdens productie minuscule logica (ongenoemde requirements) zijn verwerkt. Deze zal ik zoveel mogelijk benoemen in het verantwoordingsdocument.

De requirements zijn in 5 onderdelen verdeeld zodat de meest gerelateerde requirements bij elkaar zitten. Zo is er geen mengelmoeis van requirements en behoudt de lezer het overzicht. Ik heb per requirement de MoSCoW methode toegepast om de belangrijkheid van de functionaliteit te categoriseren.

Alle requirements in het zwart vallen onder “Must have’s” en “should have’s”. Dit vermeld ik er soms ook expliciet bij. De requirements in het oranje zijn de “nice to have’s” (ookwel “could have’s” genoemd).

Om het document kort te houden, hier een artikel over de [MoSCoW Methode](#).

Conclusie: Dit is hoe deze lijst met requirements en de onderverdeling daarvan tot stand is gekomen.

En dan hieronder de definitieve lijst met afgesproken requirements.

Algemene requirements/Niet-Functionele Requirements

ID	Requirement
01	Het Novi beoordelingsteam moet in staat zijn de eindopdracht zelf te kunnen draaien: De applicatie wordt gedraaid op hun LocalHost.
02	De applicatie moet in de modernere browsers: Chrome en FireFox kunnen draaien
03	De webapplicatie moet voor de userside responsive zijn. Voor de backoffice en admin zal dat er niet zijn. De frontend is alleen optimaal voor desktopgebruik.
04	De taal van de applicatie is in het Engels
05	De applicatie bevat Unit tests (Volgens Eindopdracht gegeven requirement) <ul style="list-style-type: none">- In geval van dit project bevat de applicatie unit tests voor de relevante logica's
06	Hoge snelheid van de applicatie is een: "Nice to have" maar daar wordt niet op getoetst.

De functionele requirements volgen hier beneden. (De Requirements in oranje zijn de nice to haves voor de applicatie)

User-side Requirements

ID	Requirement
07	Gebruikers kunnen een account aanmaken
08	Benodigde informatie van de gebruiker zijn: <i>Voor naam, Achternaam, Artiestennaam, Bio, Email, Wachtwoord. (Optioneel: profielfoto)</i> <ul style="list-style-type: none">- <i>(Het emailadres wordt gebruikt om in te loggen)</i>
09	Bij het aanmaken van een account moet er akkoord worden gegaan met Algemene Voorwaarden
10	Bij het aanmaken van een account wordt een welkomsmail verstuurd met daarin ook een link om het profiel te activeren <i>(Want dit voorkomt spoofing)</i>
11	

	Gebruikers krijgen een profiel
12	Elk profiel (dus account) krijgt één standaard interface (user interface, vergelijkbaar met facebook, gmail .ect).
13	Op de site wordt vermeld dat een bestand een maximale grootte van 15 mb en 128kbps mag zijn.
	Een account bevat <u>niet</u> meerdere profielen, Mocht een gebruiker het willen, kan hij door opnieuw te registreren een andere account maken onder een andere email
14	De gebruiker kan alleen een .mp3 tot maximaal 15 mb en 128 kbps uploaden. - (Uiteraard kan bij de upload géén andere bestandstype als demo geupload worden zoals: word-bestand, pdf, txt. Dus uitsluitend .mp3 bestanden.)
15	Het systeem controleert en vraagt de gebruiker het opnieuw te proberen als er een ongeldig bestand wordt geupload.
16	De gebruiker kan bij de demo een omschrijving meesturen. - (De maximale karakters bij een omschrijving is gebaseerd op de user experience en UI van zowel de “user-side” als de “bo-side”: max 350 karakters: (7-11 zinnen.)) - (De gebruiker kan na het opsturen géén omschrijving meesturen of wijzigen)
17	De opgestuurde demo + de informatie over de gebruiker komen bij de backoffice-side en Admin terecht.
18	De gebruiker mag maximaal 10 onbehandelde demo's op zijn profiel hebben
19	De gebruiker is in staat zijn demo terug te trekken zolang een backofficemedewerker zijn demo nog niet in behandeling heeft genomen
20	Na het in behandeling nemen van de demo blijft het bestand eigendom van het DemoDrop-systeem
21	De gebruiker moet al zijn gegevens inclusief het wachtwoord kunnen wijzigen. (m.u.v. het e-mailadres)
22	

	<p>De gebruiker moet zijn account kunnen sluiten.</p> <ul style="list-style-type: none"> - Het is noodzakelijk om het wachtwoord in te voeren voordat je het account kan verwijderen - De volgende informatie wordt uit het database gewist: Alles behalve de geüploade demo's
--	--

Backoffice-side Requirements

ID	Requirement
23	Het systeem geeft de demo's op chronologische volgorde voor de medewerker weer.
24	<p>Een medewerker kan aangeven dat hij een demo "In behandeling neemt". Hiervoor in de plaats: Een medewerker kan het beoordelingsproces "starten" en krijgt door het systeem de demo's automatisch toegewezen.</p> <p>Na goedkeuring of afkeuring op de pagina, kan de medewerker op "next demo" klikken en krijgt vervolgens een de volgende demo toegewezen. Of "exit review" om "uit" het beoordelingssysteem te stappen.</p>
25	Herhaling: Op het moment dat een demo in behandeling wordt genomen vervalt de mogelijkheid tot verwijdering bij de verzender.
26	Binnen het beoordelingsproces wordt naast de demo óók de informatie van de verzender getoond: Naam, Artiestennaam, Bio, Email Adres (Nice to have: foto)
27	Het systeem kan de demo afspelen.
28	De medewerker kan de demo ook downloaden
29	<p>De medewerker kan een demo goed of afkeuren</p> <ul style="list-style-type: none"> - Bij goedkeuring wordt direct de ingestelde standaardtekst meegestuurd: Nice to have: Er wordt direct een emailbericht hierover verstuurd. - Bij afkeuring wordt direct de ingestelde standaardtekst meegestuurd. Nice to have: Er wordt direct een emailbericht hierover verstuurd.
30	

	Goedgekeurde én afgekeurde demos worden ergens zichtbaar voor de medewerker (Want dit gaat repudiation op dat gebied tegen)
31	Bij goedkeuring zorgt het backoffice-team zélf dat die demo bij Don Diablo terecht komt. Dit kan door middel van de downloadknop.

Admin-side Requirements

ID	Requirement
32	De admin kan accounts voor het backofficeteam aanmaken/verwijderen <ul style="list-style-type: none"> - Het systeem stuurt bij runtime een activatie-link, én een gegenereerd wachtwoord op naar het mailadres. Het account van de gebruiker blijft onbereikbaar, totdat de activatielink is gebruikt. De gebruiker kan vervolgens het bijgeleverde wachtwoord gebruiken en na inloggen het wachtwoord wijzigen.
33	De admin-gebruiker kan 2 standaardteksten voor het backofficeteam instellen.
34	De admin kan handmatig (nep)accounts verwijderen.

Security Requirements **(Hier maak ik meteen inzichtelijk waarom de tegenmaatregel precies gekozen is. Ook lijkt de tegenmaatregel mij gezien de tijd en kennis de geschikte oplossing).*

ID	Requirement
35	Bij Spoofing: Aanval op authenticatie. Om te voorkomen dat gebruikers zich voor een ander kunnen voordoen wordt er een welkomsmail met een bevestigingslink gestuurd om het account te activeren. Ook kunnen er géén andere accounts op dat mailadres worden aangemaakt.
36	Bij Tampering: Aanval op Integriteit

	Om te voorkomen dat gebruikers (opgeslagen) data kunnen wijzigen zijn er 3 duidelijke rollen met elk hun restricties: gebruiker, backofficemedewerker en Admin.
37	<p>Bij Repudiation: Aanval op *geschiedenis</p> <p>Om voorkomen dat gebruikers kunnen ontkennen dat ze iets (wel of niet) hebben gedaan wordt alleen bij de backoffice duidelijk zichtbaar gemaakt welke gebruiker een demo heeft beoordeeld.</p> <p>Acties van de gebruiker en admin opslaan is een: "Should have" maar tijdsgezien niet haalbaar. De applicatie kan zonder deze functionaliteit alsnog als mvp fungeren.</p>
38	<p>Bij Information Disclosure: Aanval op Vertrouwelijkheid (<i>lees verder</i>)</p> <p>Om te voorkomen dat gebruikers/ontwikkelaars informatie zien die ze niet mogen zien wordt er met Spring Security de juiste restricties per gebruikersrol ingesteld.</p> <p>Wachtwoorden worden ge-encrypt zodat de wachtwoorden niet openlijk zichtbaar zijn in de DB. (Verder wordt er complexiteitsvalidatie (minimumlengte) op de wachtwoorden uitgevoerd om brute forcing meer tegen te gaan.)</p> <p>De applicatie wordt er beschermt tegen sql injection, zodat een gebruiker niet zomaar informatie kan opvragen.</p>
39	<p>Bij Denial of Service: Aanval op de beschikbaarheid van de server</p> <p>Hiertegen wordt geen maatregel genomen gezien het doel: het product gaat niet de lucht in.</p>
40	<p>Bij Elevation of Privilege: Aanval op Autorisatie</p> <p>Om te voorkomen dat gebruikers handelingen kunnen verrichten waarvoor zij niet geautoriseerd zijn moeten we elke manier van toegang daartoe (zoals sql injections) tegengaan. Hiervoor zal ik csfr gebruiken en de verschillende tools binnen het Spring security framework. (Meer maatregelen zijn een nice to have)</p>

Afgesproken Weggelaten requirements omdat deze uit onderzoek niet haalbaar waren:

Requirement
<p>Een backofficemedewerker kan de ingestelde tekst ook aanpassen.</p> <ul style="list-style-type: none"> - Ik kan mij niet zo goed bedenken hoe ik dit voor elkaar krijg. Dit zou ook betekenen dat een gebruiker een reactie terug kan geven aan de backofficemedewerker. Maar

ook dat de backofficemedewerker daar een melding voor moet krijgen. Dit lijkt mij gezien de tijd en kennis niet nodig
Ik heb daarom gekozen (en stel hierbij voor) om hiervan een eenrichtingsverkeer te maken: De backofficemedewerker kan aan gebruikers laten weten of hun demo wél of niet is doorgestuurd naar Don Diablo.
Een oplossing hiervoor (als een backofficemedewerker meer contact met een verzender wil) is het mailadres van de verzender te gebruiken. Deze is bij "Handled Demo's" zichtbaar.

Een demo omzetten naar mp3 15 mb 128kbps.

- Om de code compact te houden en de user de snelste service te bieden vragen we de producer zelf het bestand in 15mb 128kbps op te sturen.
Het bestand kan zo ook meteen de database ingezet worden zonder dat dat eerst bij de server moet worden omgezet