

HOGESCHOOL VOOR DE KUNSTEN UTRECHT

MUZIEK EN TECHNOLOGIE

SUPPORTIVE NARRATIVE

Creatieve systemen

Het ontwikkelen van een online streamingdienst die de creativiteit van
de artiest bevordert

Author

Jochem VAN ITERSON

Student nr:

3026764

Supervisor

Marc GROENEWEGEN

7 juni 2021



Inhoudsopgave

1	Introductie	2
2	Context en Doel	3
2.1	NichePlayer	3
3	Critical Review	5
3.1	CMS-systemen	5
3.2	Media servers	7
3.3	Streaming diensten	8
3.4	Tussen conclusie	9
4	Methode	10
4.1	Community vorming	10
4.2	Gebruikerservaring	10
4.3	Business model	11
4.4	Ontwerp en ontwikkel proces	13
4.5	Technology Readiness Levels	14
4.6	MoSCoW	15
4.7	Proces	16
5	Uitvoering en resultaten	17
5.1	Backend	17
5.2	Frontend	18
5.3	Systeemdiagram	19
5.4	Resultaten	20
6	Conclusie en reflectie	22
	Referenties	24
	Appendices	25
A	Een self-hostable stramingsdienst, waarom is het nodig?	26
B	Database diagram NichePlayer	37
C	Screenshots NichePlayer	38

1 Introductie

Dit Supportive Narrative is een ondersteunend onderzoek bij het ontwikkelen van de NichePlayer, een open-source, self-hostable streamingdienst. In de critical review van dit onderzoek worden bestaande diensten en domeinen met elkaar vergeleken. Daarnaast worden meerdere hiervoor relevante business modellen uitgedacht. Met de informatie uit dit Supportive Narrative wordt een proof of concept van de NichePlayer ontwikkeld waarmee eerste gebruikerstests kunnen worden uitgevoerd. Dit onderzoek bouwt verder op op een onderzoek wat is uitgevoerd tijdens het schrijven van mijn paper in jaar drie (zie appendix A) [Ite20].

Het eindresultaat van het onderzoek is een prototype. Deze kan vervolgens, samen met de onderbouwende uitkomst van het onderzoek, gebruikt worden om andere ontwikkelaars bij het project te betrekken en artiesten, labels en producers uit te nodigen tot het gebruik van het systeem.

Het is interessant om een systeem te ontwerpen wat open genoeg moet kunnen zijn voor de creatieve gebruikers die de doelgroep zijn van het project. Bij het ontwerpen van het systeem moet genoeg ruimte vrijgelaten worden om de artiest zijn eigen gang te kunnen laten gaan. Tegelijk moet de Nicheplayer een stabiele basis kunnen aanbieden om op te werken.

2 Context en Doel

Al sinds de uitvinding van de grammofoonplaat is de manier en het medium waarop muziek geluisterd wordt in beweging. Van live muziek op echte instrumenten naar een analoog opslagmedia als de lp, de digitalisering naar cd, en uiteindelijk de opkomst van de computer en het internet. Waar het vroeger nodig was fysiek naar een uitvoering van een stuk te gaan is nu alle muziek toegankelijk op je telefoon. Deze verandering heeft niet alleen invloed gehad in de manier van luisteren, ook de muziek zelf heeft zich aangepast. Met name popmuziek heeft zich aangepast aan de mogelijkheden van het nieuwe medium. Bij de uitvinding van de single werden stukken ingekort of opgerekt zodat ze precies binnen de vijf minuten van de plaat pasten. Door de uitvinding van stereo geluid maakte bands als The Beatles sterk gebruik van panning. Na iedere uitvinding wordt het vorige medium snel minder gebruikt.

De meest recente uitvinding van muziek streaming heeft interessante ontwikkelingen opgeleverd. Zo heeft het luisteren van muziek via een streamingdienst er voor gezorgd dat mensen niet meer naar volledige albums luisteren maar meer naar de losse nummers. Ook hebben playlists een grotere rol gekregen. Om in deze playlists de juiste muziek aan te bieden hebben diensten als Spotify algoritmes ontwikkeld die het luistergedrag analyseren. Op basis daarvan kan vervolgens nieuwe muziek worden aanbevolen aan de luisteraars.

Een andere verandering is de manier waarop er voor muziek wordt betaald. Waar artiesten en labels vroeger betaald kregen per album of single krijgen zij via het streaming platform betaald per afgespeeld nummer. Het afspelen telt als een nummer langer dan 30 seconden is afgespeeld. De opbrengst per stream is erg laag, voor Spotify is dit op het moment ongeveer \$0,003 per stream. Van dit bedrag gaat vervolgens ook een deel van naar het label of de producer. Om als artiest toch genoeg geld te verdienen aan muziek worden nummers nu steeds korter. Daarnaast is de drop of climax van nummers nu in nagenoeg alle gevallen na 30 seconden. Hiermee kan het nummer vaker worden afgespeeld, en blijven mensen langer in het nummer plakken.

Voor kleine en beginnende artiesten is deze nieuwe manier van luistergedrag niet rendabel. Hun muziek is niet bekend genoeg en krijgt hierdoor te weinig streams om de oude omzet met de verkoop van cd's en platen te evenaren. Voor de opkomst van streaming diensten verdienden kleine artiesten hun geld door merchandise als cd's en shirts aan fans te verkopen bij optredens en festivals. Omdat tegenwoordig niemand meer een cd speler heeft, en de optredens en festivals vanwege de pandemie niet meer worden gehouden, hebben deze artiesten een inkomsten tekort.

2.1 NichePlayer

Er is een steeds groter wordende groep artiesten en producers dat niet meer rond kan komen van hun werk. Kleine en beginnende artiesten draaien lang verlies met hun werk, en zijn genooddaakt bijbanen te nemen. De drempel tot het professionele niveau is steeds hoger, wat een probleem is voor de diversiteit van de muziekcultuur op streaming platforms.

Een oplossing van dit probleem is het introduceren van een platform of medium dat speciaal is ontworpen voor deze groep artiesten. Het ontwikkelen van deze technologie is onderdeel van dit practice based onderzoek. De werktitel van dit project is NichePlayer.

NichePlayer is een open-source streamingdienst. Met dit systeem is het mogelijk voor artiesten hun muziek en content op hun eigen manier uit te brengen. Hiermee krijgen zij naast de vorm waarop gedeeld wordt ook controle over hoeveel zij er aan verdienen. De beoogde doelgroep is kleine artiesten en labels. Voor hen is muziek aanbieden via bestaande diensten als Spotify niet rendabel (\$0.003 per stream), zeker in vergelijking met de verkoop van cd's van vijftien jaar geleden. Met dit product krijgt de artiest weer grip op wat hij doet met zijn muziek. Daarnaast wordt de NichePlayer ook een platform waar ontwikkelaars nieuwe technologieën op kunnen uitproberen. Zoals de titel aangeeft, *“een creatief platform dat de creativiteit van de artiest bevordert”*.

3 Critical Review

Veel bestaande systemen en services bieden delen aan van de functionaliteit van de NichePlayer. Deze systemen zijn op te splitsen in drie domeinen: CMS¹, media servers en streaming diensten. Door deze systemen onderling met elkaar te vergelijken kunnen de sterke en zwakke punten worden gevonden in relatie tot het beoogde NichePlayer systeem. Een groot deel van deze diensten zijn open-source en self-hostable. Dit kader is gesteld aangezien de NichePlayer zelf ook op deze manier aangeboden gaat worden. De te onderzoeken diensten worden vergeleken aan de hand van de volgende thema's: business model van de ontwikkelaar, community om het systeem heen, mate van configuratie en onderhoud, en hoe het systeem creativiteit mogelijk maakt.

3.1 CMS-systemen

Een CMS is een online systeem wat digitale content beheerd. Hiermee kunnen mensen zonder technische kennis op een eenvoudige manier websites en andere toepassingen maken. Zij hoeven daarbij niet te programmeren en kunnen op een visuele manier hun pagina's maken. Bekende voorbeelden van CMS systemen zijn Wordpress, Joomla en Drupal.

Wordpress

Wordpress is een blog systeem gebaseerd op PHP. Naast blogs kunnen ook statische pagina's worden gemaakt. Het bedrijf achter Wordpress bestaat uit twee delen: .org en .com. Op de .org site [Wor21b] kan de broncode gratis worden gedownload. Deze kan vervolgens worden geüpload naar de server. Daarnaast is de .org site een plek waar ontwikkelaars plugins en thema's kunnen uploaden. Wordpress.com [Wor21a] levert op basis van het systeem van .org een webhost. Dit doet het in de vorm van een abonnementsdienst.



Figuur 1: Wordpress

De layout en functionaliteit van Wordpress is aan te passen door middel van themes en plugins. Deze kunnen door het open-source karakter van het systeem door iedereen worden gemaakt. Hiermee is Wordpress naast een framework om snel blogs en websites op te kunnen zetten ook een platform waarop ontwikkelaars nieuwe technologieën kunnen maken, uitproberen en verkopen.

Wordpress heeft voor het project meerdere interessante aspecten. Zo is het de ontwikkelaars van Wordpress gelukt een open-source product aan te bieden, waar toch een business model omheen is gebouwd. Wordpress is vrij te downloaden en aan te passen met plugins en thema's. Hiermee kan iedereen zijn eigen web blog of pagina aanmaken en via het dashboard beheren. Voor mensen die geen of weinig kennis hebben van het beheren van een eigen server of webhost heeft Wordpress een business model opgezet. Voor deze mensen biedt het namelijk een eigen hosting server. Deze server wordt door Wordpress beheerd, en is met een zeer versimpelde weergave te configureren door de

¹Content Management System

gebruiker. De functionaliteit van het systeem blijft hetzelfde, alleen wordt het voor de gebruiker makkelijker gemaakt en kan deze zich focussen op de content die op de site moet komen te staan.

Verder is de manier waarop de gebruiker pagina's en blogs kan maken bruikbaar voor de NichePlayer. Pagina's op Wordpress worden geschreven op basis van blocks, losse widgets die zelf content kunnen laten zien met een eigen speciale weergave. Hiermee kan de structuur van de pagina makkelijk worden aangepast. Daarnaast is het door plugins makkelijk nieuwe functies en blocks toe te voegen.

Wordpress is geen audio player, en kan dat vanwege zijn ontwerp ook niet makkelijk worden. Het is ontwikkeld als een systeem waarbinnen eenvoudig blogs en statische pagina's te ontwikkelen zijn.

PicoCMS

PicoCMS [Pic21] is een flat file CMS. Dit houdt in dat bestanden als ruwe tekst opgeslagen staan op de server. Het voordeel hiervan is dat er geen database nodig is om het systeem te laten draaien. De files die worden gebruikt zijn geschreven in Markdown.

De functionaliteit die uit PicoCMS gehaald kan worden is de manier waarop de pagina's worden gebouwd en opgeslagen. Door te schrijven in Markdown is voor de auteur direct duidelijk hoe de pagina er uit gaan zien, het is als het schrijven van een normaal document. De opslag van de pagina is ook eenvoudig aangezien het in platte tekst is.

PicoCMS heeft geen (online) configuratie paneel, alle instellingen dienen te worden veranderd door middel van het editen van de files op de server. Ook de grote hoeveelheid open-source plugins en thema's moeten op deze manier worden geïnstalleerd. Veel gebruikers kiezen daarom voor eenvoudigere systemen als WordPress.

Shopify

Er zijn veel e-commerce applicaties beschikbaar. Via een webpagina als Shopify kan een ondernemer snel een eigen webshop opzetten en hierop zijn product verkopen. Net als andere CMS systemen is het bij Shopify mogelijk de opmaak en inhoud van de site aan te passen. Er ligt daarbij wel een sterke focus op de verkoop van producten. Binnen het dashboard heeft informatie over inkomsten een prominente rol. In Shopify zit automatisch een betaalsysteem verwerkt.



Het business model van Shopify is een abonnementsdienst. Daarnaast moet de verkoper een extra bedrag betalen over zijn inkomsten als deze via de betaalmiddelen van Shopify lopen. Shopify biedt drie verschillende abonnementen aan, die van elkaar verschillen door support en het ontbreken van enkele functionaliteiten.

Figuur 2: Shopify

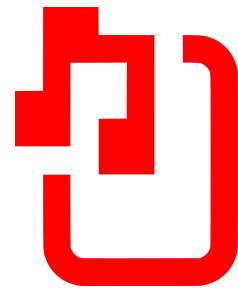
MusiCards

In de winter van 2019 heb ik stage gelopen bij de ontwikkelaar van MusiCards [Rul21]. Het bedrijf achter MusiCards verkoopt fysieke kaarten van verschillende formaten met hierin een (zelf ontworpen) NFC² chip. Na het scannen van de kaart krijgt de gebruiker toegang tot een online portal. Dit portal is door de beheerder van de kaart te bewerken via een door MusiCards ontwikkeld CMS.

MusiCards verkoopt met het kaartje niet alleen een fysiek product, maar ook (beveiligde) toegang tot het systeem. Dit principe is bruikbaar voor de NichePlayer door een NFC chip te verstoppen in een lp of 3D-geprint object. Het fysieke product geeft extra waarde aan het digitale platform.

MusiCards heeft in zijn systeem een zeer uitgebreide audio player verwerkt. Met deze player kan worden gecast naar externe players. Daarnaast kan de kwaliteit worden aangepast tijdens het afspelen.

Wat MusiCards niet heeft is een open-source variant. Alleen de ontwikkelaar kan nieuwe 'kaarten' aanmaken waardoor deze een niet te omzeilen commerciële positie heeft.



Figuur 3: MusiCards

3.2 Media servers

Naast het kunnen bewerken van de content is het delen en organiseren van media ook een belangrijk aspect van de NichePlayer. Met de onderstaande diensten kan een zelf opgebouwde media bibliotheek worden gedeeld, beluisterd en bekeken.

Ampache

Ampache is een op php gebaseerd systeem wat de muziek bibliotheek van de gebruiker deelt. De server indexeert folders en laadt hieruit alle metadata van de muziek bestanden. De player is gebaseerd op HTML5 audio. Hierdoor werkt Ampache, via de browser, op alle computers. Omdat de beheerder zelf eigenaar is van de bestanden kan hij deze ook (zolang er is ingelogd) downloaden.

Een van de nadelen van Ampache is dat de muziek niet binnen het systeem te uploaden is naar de server. De beheerder moet altijd toegang hebben tot de bestanden van de server om nieuwe muziek aan de library toe te voegen. Daarnaast is de player niet geoptimaliseerd voor mobiele apparaten, zowel wat betreft layout als de manier van afspelen. De layout is niet volgens de responsive richtlijnen (layout verandert afhankelijk van de grootte van het scherm) waardoor mobiele apparaten nog steeds met de (grote) desktop versie werken. De player werkt niet offline en ondersteunt niet het cachen van bestanden.



Figuur 4: Ampache

²Near Field Communication

Plex

Plex lijkt qua vorm erg op Netflix. Net als Ampache kan met Plex een eigen (film)bibliotheek worden gedeeld. Door het gebruik van apps is de ervaring altijd geoptimaliseerd voor het apparaat waarop wordt gekeken. Dit is ook mogelijk doordat de server de media realtime kan transcoderen, waarmee altijd de best haalbare kwaliteit wordt gebruikt voor het apparaat en moment van afspelen.

Na het indexeren van de media bibliotheek gaat Plex op verschillende internetbronnen zoeken naar de correcte metadata van het bestand. Dit blijft niet alleen bij een poster afbeelding, Plex gaat ook automatisch op zoek naar ondertiteling, informatie over de cast en beschrijvingen van de film of serie.

De apps van Plex vereisen een account bij de hoofddienst van Plex. Hierdoor zal Plex altijd weten welke apparaten zijn ingelogd en worden gebruikt door een account. Om op mobiele apparaten te kunnen kijken moet de gebruiker een abonnement afsluiten. Daarnaast biedt Plex, ook weer achter een abonnement, toegang tot films en series.



Figuur 5: Plex

3.3 Streaming diensten

Spotify

Spotify is de bekendste streaming diensten van het afgelopen decennium. Na de opkomst van digitale media (denk aan het downloaden van mp3 files) [Wit15] heeft Spotify samen met andere diensten gezorgd voor een nieuwe manier van luisteren naar muziek.

Een van de krachtige punten van Spotify die relevant is voor de NichePlayer is het offline afspelen van muziek. Hiermee kan, ook wanneer er geen of een slechte internetverbinding is, worden geluisterd naar muziek. De ervaring blijft nagenoeg hetzelfde, met als verschil dat er niet naar nieuwe muziek gezocht kan worden. Het businessmodel van Spotify is een abonnementsdienst. Voor €9,99 heeft de gebruiker toegang tot miljoenen nummers. Als de gebruiker geen abonnement heeft bij Spotify dan worden er tussen nummers door reclames afgespeeld.

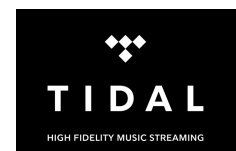


Figuur 6: Spotify

Tidal

TIDAL is bekend geworden doordat een aantal grote artiesten zich verzetten tegen het grote Spotify en met hun eigen dienst kwamen. TIDAL is een voorbeeld van een steeds vaker voorkomende ontwikkeling van alternatieve concurrerende diensten. De NichePlayer beweegt mee met deze trend van decentralisering.

TIDAL is in tegenstelling tot andere diensten vanaf het begin bezig met het aanbieden van hoge kwaliteit audio (in het formaat FLAC³). De kwaliteit en psycho-akoestische luisterervaring van dit format is vele malen



Figuur 7: Tidal

³Free Lossless Audio Codec

beter dan dat van mp3 of andere codecs van andere diensten.

Youtube (Music)

TODO: schrijven cr youtube

- Vrij uploaden van content (User generated content)



Figuur 8: YouTube

3.4 Tussen conclusie

De NichePlayer is een creatief platform dat de creativiteit van de artiest bevordert. Met dit in het achterhoofd, als de voorbeelden van de critical review op een rij worden gezet dan gaat er voor de NichePlayer gebruik worden gemaakt van de volgende elementen.

Het businessmodel, admin panel en de pluginstructuur van Wordpress zijn punten die voor de NichePlayer nuttig zijn. **TODO: Verder schrijven conclusie Meenemen**

- Business model Wordpress
- Admin panel Wordpress
- Pluginstructuur Wordpress
- Flat page storage PicoCMS
- Geïntegreerd betalingsmethode Shopify
- NFC MusiCards
- **TODO: Aanvullen**

Verbeteren

- Algoritme van Spotify NIET overnemen
- **TODO: Aanvullen**

4 Methode

Mijn werk in het algemeen en dus ook dit onderzoek rust op vier kern onderwerpen: Community vorming, Gebruikerservaring, Businessmodel en Ontwerp en ontwikkel proces. Deze onderwerpen staan aan de basis van mijn projecten en bezigheden en zijn de drijvende factor van mijn werk. Door de NichePlayer vanuit deze vier onderwerpen te benaderen kom ik tot concepten en features van het systeem.

4.1 Community vorming

Community vorming is het verbinden van verschillende mensen of groepen. Door het wegvallen van de fysieke verkoop van muziek en het gebrek aan optredens door de pandemie is het moeilijker contact te leggen tussen fans en artiest.

Met de NichePlayer wordt een platform ontwikkeld waarbinnen een community kan worden opgebouwd. Binnen het systeem zal het mogelijk zijn contact te leggen tussen de artiest en de luisteraar, wat momenteel binnen de bestaande platformen zoals Spotify niet mogelijk is. Ook moeten de fans zich onderling met elkaar verbonden voelen door middel van het systeem.

Naast de community van de fans moeten de artiesten ook onderling met elkaar verbonden worden. Door een hub te ontwikkelen waar de afzonderlijke NichePlayers op gebundeld worden is het mogelijk alles aan elkaar te linken, en kan worden ingehaakt op elkaars fanbase. Daarnaast kan de player met een plugin- en theme-store worden aangepast naar de voorkeur van de artiest.

Het systeem zal ook een plek worden waar ontwikkelaars nieuwe technologieën kunnen maken en uitproberen. Het is hierom belangrijk dat er een goede API en referentie materiaal wordt ontwikkeld om plugins op te kunnen schrijven. De plugins moeten makkelijk te installeren zijn zodat iedereen er mee aan de slag kan.

4.2 Gebruikerservaring

In de wereld van software ontwikkeling wordt het steeds belangrijker dat een systeem er modern uitziet. Gebruikers zijn sneller geneigd de applicatie niet te gebruiken als deze visueel verouderd lijkt. Daarnaast willen gebruikers dat het instapniveau van de applicatie makkelijker wordt. Interfaces moeten overzichtelijker en meer uniform.

Er zijn hiervoor meerdere design guides beschikbaar, waarmee wordt gezorgd dat applicaties een onderbouwd en uniform uiterlijk hebben. Hierbij wordt gestreefd naar minimalisme en simpliciteit. Een voorbeeld van een design guide is Material Design [Goo21]. Material Design is de norm die wordt gebruikt bij het ontwikkelen van Android apps, maar wordt ook gebruikt in verschillende web frameworks als Quasar.

De NichePlayer heeft naast de visuele gebruikerservaring ook een functionele gebruikerservaring. Deze is opgesplitst in artiest en luisteraar.

Artiest

Een doel van dit project is om een dienst te ontwikkelen waarmee artiesten snel hun eigen streamingdienst kunnen opzetten. Een aanname hierbij is dat de artiest, producer of het label geen kennis of interesse heeft in het hosten en beheren van een (LAMP⁴) webserver. Installeren van de NichePlayer doet de artiest door een zip bestand of een folder te uploaden naar zijn server en de installer van het systeem te draaien. Dit is vergelijkbaar met het installatie proces van Wordpress. Dit moet het enige zijn om het systeem te configureren, na het installeren moet het niet meer nodig zijn om in te loggen op de ftp/ssh van de server en de broncode aan te passen. Hiervoor is er een uitgebreid, maar overzichtelijk, admin panel nodig waarin alles van het systeem aangepast en beheerd kan worden.

Voor het dagelijkse gebruik zijn er in dit systeem twee UI⁵ ingangen: het admin panel en de player. Het admin panel is browser-first ontwikkeld, de player mobile-first. Dit betekent niet dat de player niet werkt op een (desktop) computer. Het admin panel is bedoeld om (groot) onderhoud te plegen aan het systeem, en zal daarom ook niet veel op een telefoon worden gebruikt. De player zal daarentegen wel op veel verschillende apparaten gebruikt worden, van mobiel tot desktop. Door in de player een responsive layout norm te gebruiken kan deze op zowel mobiel als desktop computers goed werken.

Luisteraar

De luisteraar zal door het gebruik van de NichePlayer een extra streamingdienst app op zijn apparaat moeten installeren. Dit kan voor sommige gebruikers een drempel zijn. Het is daarom belangrijk dat de NichePlayer genoeg waarde toevoegt. Voor het ontwikkelen van de NichePlayer is de aanname gedaan dat luisteraars het geen probleem vinden een extra dienst of app te moeten aanschaffen, als ze er een betere of unieke ervaring voor terug krijgen. De unieke ervaring komt door extra functionaliteiten die in andere diensten niet zitten, waaronder de mogelijkheid tot direct contact met de artiest.

4.3 Business model

Een business model of bedrijfsmodel is een model wat verschillende bedrijfsaspecten beschrijft waaronder de inkomsten, uitgaven en de waarde van het producten die worden aangeboden. Er zijn in dit systeem twee niveaus waar een business model naar voren komt, dat van de artiest en van de ontwikkelaar.

Artiest

De artiest, producer of het label wil, door middel van de NichePlayer, meer inkomsten wil genereren. Reden hiervoor is dat andere diensten een te hoge drempel hebben om genoeg geld te verdienen. NichePlayer springt hierop in door de big corporates (Spotify,

⁴Linux Apache MySQL PHP

⁵User Interface

Apple Music, Tidal, ...) tussen de gebruiker en de artiest uit te halen. Het contact gaat bij de NichePlayer direct tussen de artiest en de fan. Hierdoor kan de artiest zelf bepalen hoeveel hij wil verdienen aan een album of het algemene gebruik van zijn dienst. In mijn paper uit jaar drie (zie appendix A) [Ite20] heb ik verschillende business modellen onderzocht. Daarnaast ben ik in deze paper bij een aantal artiesten nagegaan wat zij vinden van het concept. **TODO: Kort beschrijven wat dan precies uit het onderzoek is gekomen.**

De waarde van het systeem komt in het geval van de NichePlayer uit de exclusiviteit. Content wordt alleen op de NichePlayer uitgegeven. Het direct contact tussen artiest en fan gaat niet alleen over de betaling van de muziek. Ook kunnen er via de NichePlayer functionaliteiten worden toegevoegd die de fan dichterbij de artiest laat staan. Denk hierbij bijvoorbeeld aan het binnen het systeem verkopen van merchandise en bijzondere functies als een webchat of een manier om op eigen instrument mee te spelen met de muziek.

Ontwikkelaar

Op een ander niveau zit de ontwikkelaar van de NichePlayer. Omdat de NichePlayer een creatief platform moet zijn, is het een vereiste dat het open-source wordt ontwikkeld. Het voordeel van open-source is dat een grotere groep ontwikkelaars bezig kan zijn met het systeem, zich betrokken voelen bij de ontwikkeling ervan, en in staat zijn technische ondersteuning te bieden waar nodig.

Open-source software heeft een ingewikkelde relatie met het genereren van inkomsten. Doordat de soft- of hardware gratis toegankelijk moet zijn op het internet is het ingewikkeld hier een businessmodel aan te hangen. **TODO: Verder uitwerken**

Er zijn verschillende business modellen mogelijk, die elkaar niet uitsluiten. Door het aanbieden van betaalde functies kan de waarde van het product hoger worden gemaakt. De artiest kan werken zonder deze functies, maar heeft hierbij wel het nadeel dat het product minder waard is. Een voorbeeld van een betaalde functionaliteit is het aanbieden van NFC⁶ kaarten en objecten. NFC is een chip die onder andere wordt gebruikt in betaalpassen, de OV chipkaart en toegangspassen in hotels. De technologie is gebaseerd op een chip met een antenne. Deze antenne kan vervolgens in veel vormen worden gedrukt. Door het online album te vervangen door een fysiek object met een NFC chip krijgt dit album een fysieke factor terug. Daarbij krijgt het een fysieke waarde vergelijkbaar aan de cd of lp. Mensen zijn hierdoor sneller geneigd meer geld uit te geven omdat zij iets fysieks bezitten.

Een ander business model voor de ontwikkelaar is het aanbieden van technisch support en webhosting. Hiermee kan de artiest zich focussen op het schrijven van de muziek en hoeft hij zich niet bezig te houden met het onderhouden van het websysteem. Dit business model is vergelijkbaar met dat van Wordpress.

De NichePlayer moet een systeem worden waarop nieuwe technologieën kunnen worden uitgetest en ontwikkeld. Voor betaalde plugins en thema's kan worden gekozen

⁶Near Field Communication

een percentage in te houden. Ook kan ontwikkelaars in het algemeen worden gevraagd een licentie af te nemen als zij plugins willen ontwikkelen. Dit breekt echter de open-source mindset.

4.4 Ontwerp en ontwikkel proces

Is het mogelijk om een systeem te ontwikkelen wat stabiel en volledig is en tegelijk toch genoeg mogelijkheden en openingen biedt voor andere ontwikkelaars en creatievelingen? Hierin zit een tweestrijd. Om een systeem 100% stabiel te maken moet alles volledig afgewerkt en dichtgetimmerd zijn. Alle bouwstenen in het systeem hebben een duidelijke eigen functie en hebben geen losse eindjes.

Hier tegenover staat een systeem wat volledig open is, veel aanknopingspunten biedt voor ontwikkelaars om op in te haken en de mogelijkheid biedt nieuwe functionaliteiten toe te voegen. Het is moeilijk dit te ontwikkelen met een systeem wat streeft naar 100% stabiliteit. Er zijn veel losse eindjes nodig om op te kunnen bouwen, en het is voor het hoofdsysteem onbekend of de verschillende losse onderdelen met elkaar kunnen samenwerken. Hierdoor kan het systeem chaotisch overkomen.

Deze tweestrijd speelt niet alleen bij het systeem, ook de ontwikkelaar en artiest hebben hier mee te maken. De ontwikkelaar kan niet uitgaan van een stabiel en volledig eisenpakket, het creatieve karakter van het platform zal altijd nieuwe en vooral onverwachte functionaliteiten vragen. De artiest wil zich creatief kunnen onderscheiden, zowel muzikaal als visueel. Dit vraagt om openheid van het systeem.

Een oplossingsrichting is om een systeem te ontwikkelen op basis van bestaande (stabile) bouwstenen, en het ontwikkelen op deze manier modulair aan te pakken. De losse onderdelen zijn verzekerd van 100% stabiliteit, en met een duidelijke API⁷ en overzichtelijk systeemdiagram zijn de modules aan elkaar te knopen waarbij de stabiliteit gewaarborgd wordt.

Om dit alles werkend te krijgen moet er een goede backend zijn. Hierin zijn verschillende keuzes te maken. Zo kan de backend volledig vanaf scratch worden geschreven. Zo is honderd procent zeker dat het systeem werkt voor deze specifieke toepassing. Daarnaast kan de backend op deze manier zo minimalistisch mogelijk gehouden worden.

Er kan ook voor worden gekozen een bestaande backend library te gebruiken en deze aan te passen en in te vullen met de gewenste extensies. Een voordeel hierbij is dat niet het wiel opnieuw hoeft worden uitgevonden. Er zijn veel goed werkende systemen die slechts aangepast hoeven te worden.

Binnen dit Supportive Narrative wordt onderzocht wat de beste manier is om de backend op te zetten. Hierbij wordt bekeken wat de voor- en nadelen zijn van verschillende libraries en het zelf ontwikkelen van een backend.

Een ander aspect wat onderzocht moet worden is wat het beste distributietype is. De installatie van de NichePlayer moet zo soepel en makkelijk mogelijk zijn. Een mogelijk

⁷Application Programming Interface

distributietype is het aanbieden als zip file. Deze moet de artiest uitpakken en uploaden naar zijn server. Hiervoor moet de artiest (eenmalig) toegang hebben tot bestanden op de webserver.

Een andere manier is het aanbieden van een Docker image. Binnen docker draaien systemen in eigen gesandboxte containers waardoor het erg veilig is. Ook de installatie van de NichePlayer is eenvoudig, er hoeft slechts een commando in een terminal uitgevoerd te worden. Wel is het instapniveau om een terminal te gebruiken voor sommige gebruikers te hoog. Daarnaast bieden niet alle hosting providers de toegang tot een terminal aan.

4.5 Technology Readiness Levels



Figuur 9: Technology Readiness Levels, omschreven door de EU

Het Technology Readiness Level (TRL) is een ontwerpstrategie ontwikkeld door NASA in de jaren zeventig. [Inn21] Het wordt gebruikt om de voortgang van een systeem of project aan te duiden. Dit doet het door de voortgang te toetsen aan negen verschillende levels, te zien in de afbeelding hierboven (Figuur 9).

Dit project zal worden ontwikkeld tot TRL 4. Het eindresultaat is een werkend prototype. Het systeem moet intern kunnen draaien en kunnen fungeren als ondersteunend product bij gesprekken en acquisities.

Voor de ontwikkeling van de NichePlayer wordt de volgende interpretatie gebruikt van het TRL. In TRL 0 (Idea) wordt het concept volledig uitgeschreven en gedacht. Hierbij worden aannames gedaan die later worden getest. In TRL 1 (Basic Research) worden

kleinschalige tests en onderzoek uitgevoerd waarmee wordt uitgezocht welke technologieën gebruikt gaan worden. In TRL 2 (Technology formation) zijn deze technologieën vastgesteld en wordt een eerste basic versie ontwikkeld van het systeem. Aan het einde van dit stadium moet de *Must* van de MoSCoW (hieronder uitgelegd) af zijn. In TRL 3 (Applied research) wordt gewerkt naar een proof of concept. Intern wordt getest of het systeem naar behoren functioneert. Uit de tests van TRL 3 zijn nieuwe inzichten gekomen. Deze ga worden in TRL 4 (Small scale prototype) verwerkt. Daarnaast moeten hier de *Should* van de MoSCoW uitgevoerd zijn. Hierna volgt nog een kleinschalige test met een enkele testcase. Dit moet de uiteindelijke versie worden waar dit het Supportive Narrative op is gebaseerd. De NichePlayer moet alles van de MoSCoW exclusief de *Would* hebben geïmplementeerd. Dit kan ook in de vorm van een Wizard of Oz test waarbij de functie op de achtergrond wordt uitgevoerd door echt persoon, maar het voor de gebruiker lijkt alsof het systeem het doet.

4.6 MoSCoW

Om dit project gestructureerd uit te gaan voeren wordt de ontwerpstrategie MoSCoW gevolgd. Deze ontwerpstrategie verdeelt taken en bezigheden aan de hand van prioriteit door deze onder te verdelen in vier groepen: Must have, Could have, Should have en Would have. De taken die in *Must have* staan zijn een vereiste voor het slagen van het project. Alle taken van de Must have moeten zijn voldaan voordat het systeem of project af is. *Should have* zijn taken die belangrijk zijn voor het project, maar niet verplicht zijn voor tijdlijn tot aan de volgende release. *Could have* zijn punten die de ervaring van het product kunnen verbeteren. *Would have* is de groep waar nieuwe ideeën en toekomstplannen in komen te staan. Deze punten hoeven pas aangepakt te worden als al het vorige af is. Voor deze groep wordt ook wel eens *Won't have* gebruikt omdat de ontwikkeling vaak niet tot dit punt komt. De MoSCoW van dit project ziet er als volgt uit:

Must	• Afspelen van muziek • Admin users • Admin panel • Inzicht in afgespeelde streams • Self-hostable first (makkelijk te installeren en te onderhouden)
Should	• Uploaden en beheren muziek • Mockup implementatie betaling methodes • Aanpasbare layout en thema • Normale users • Toegang via fysiek object
Could	• Blog functionaliteit • Plugins backend – Content (gravicon...) – Source (webdav...) • Plugins frontend – Functionaliteit (lyrics...) • Implementatie betaling methodes (PayPal)
Would	• Dedicated apps • Hub • Offline downloaden van muziek • Meerdere kwaliteiten van de muziek – Upload in flac, server convert naar juiste formats • Streamen naar externe players (UPnP, DLNA, Airplay, Chromecast)

4.7 Proces

Ik ga het ontwikkelen van de NichePlayer benaderen als een creatief project. Door te werk te gaan als rapid prototyper kan ik snel veel concepten uitwerken waarna er uiteindelijk een kan worden afgemaakt. Door deze manier van werken kunnen ideeën uitgetest, gefinetuned, en gereleased worden binnen een korte tijd. Nieuwe functionaliteiten worden als MVP ⁸ toegevoegd met als doel om snel feedback te kunnen krijgen. Rapid prototyping is een strategie die creativiteit bevordert. Dit is van belang bij de NichePlayer zowel in het gebruik als in het ontwikkelen van het platform. Als een systeem open en creatief moet zijn, dan moet dit in het proces dat ook zijn.

⁸Minimum Viable Product

5 Uitvoering en resultaten

5.1 Backend

Om dit systeem goed te laten werken en voornamelijk toekomstbestendig te maken moet er een stevige backend zijn. Deze kan volledig zelf worden ontwikkeld, maar er kan gebruik gemaakt worden van bestaande frameworks. Frameworks staan aan de basis van veel software pakketten of applicaties. Een framework biedt functionaliteiten die de programmeertaal zelf niet aanbiedt, denk aan visuele error logging of bouwstenen voor het schrijven van een UI.

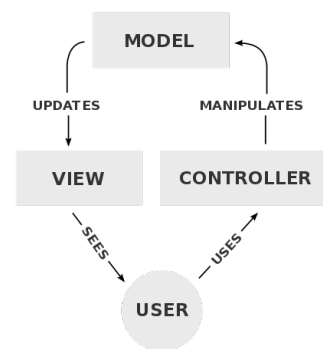
De backend kan ook volledig zelf worden ontwikkeld. Hiermee is zeker dat de backend precies doet wat nodig is voor de NichePlayer. Hiermee blijft de backend clean en doet het geen onnodige berekeningen. Het ontwikkelen van een eigen backend is echter zeer complex. Daarnaast is onderhoud zwaarder.

Er zijn verschillende frameworks die worden uitgetoetst. Al deze frameworks zijn op basis van de PHP programmeertaal om te kunnen werken op webserver. Veel van deze backend frameworks zijn gebaseerd op het MVC ontwerp: Model, View Controller (Figuur 10). Bij een MVC wordt het opslaan, visualiseren en aansturen van data opgesplitst in verschillende files of sub-onderdelen van het systeem. Hiermee is het duidelijker wat een bepaald deel van het systeem doet.

Er is voor gekozen API-platform te gebruiken als backend library. Om bij dit framework uit te komen zijn tests uitgevoerd met verschillende frameworks. Door iteratief te werk te gaan en in alle verschillende frameworks snel het MVP (Minimum Viable Product) te implementeren kon uiteindelijk het meest geschikte framework worden gevonden. Deze manier van aanvoelen of een framework bruikbaar is door met systeem te experimenteren, is typerend voor een rapid prototyper. Dit staat tegenover het via de specificaties op functionaliteiten filteren en uitzoeken van een systeem. Andere frameworks waren te gesloten of moeilijk te configureren, waardoor zij ongeschikt waren om creatief in te kunnen ontwikkelen. Hierdoor waren zij niet inzetbaar als fundering van de NichePlayer.

Als eerste werd Laravel onderzocht, een framework wat vanaf 2011 in ontwikkeling is. Binnen Laravel bleek de *View* van MVC te sterk gedefinieerd. Het was niet mogelijk de scheiding te maken tussen backend en frontend (door middel van een API), wat nodig is om modulair te werk te kunnen gaan. Laravel heeft veel functies geïmplementeerd die voor het ontwikkelen van een API backend niet nodig zijn. Daarnaast was het in Laravel moeilijk hiernaast extra functionaliteiten toe te voegen. Laravel is te veel dichtgetimmerd om te kunnen werken als creatief platform.

Lumen is een framework gebaseerd op Laravel. Waar Laravel op veel gebieden te complex en breed was, is Lumen juist te eenvoudig. Veel functies die Laravel te ingewik-



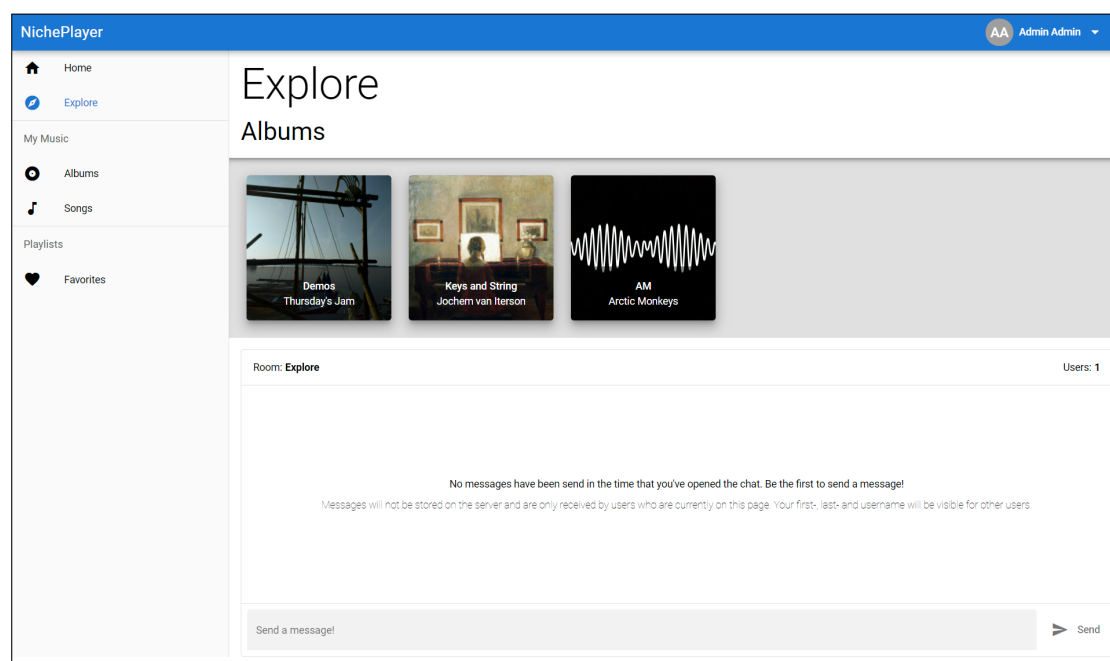
Figuur 10: Model View Controller

keld maken zitten niet verwerkt in Lumen, waardoor het makkelijker is om te gebruiken. Echter zitten de minpunten van Laravel ook in Lumen. Binnen Lumen is het namelijk net als bij Laravel ingewikkeld om extensies te installeren. Omdat Lumen een nog uitgekledere versie is van Laravel is het hierom moeilijk om te gebruiken als basis voor de NichePlayer.

Een alternatief van Laravel is Symfony. Symfony is een backend library die aan de basis staat van veel bekende platformen waaronder NextCloud. Symfony heeft een groot aantal extensies en plugins, en ondersteunt hiermee de vraag voor een uitbreidbare basis. De basis van Symfony is echter erg kaal, het is de bedoeling dat de ontwikkelaar zelf alle plugins aan elkaar knoopt. Dit vergt veel kennis en inzicht, wat niet past bij het werken als rapid prototyper. Hierdoor is ook Symfony ongeschikt, er moet te veel worden geconfigureerd om een basissysteem aan de praat te krijgen.

Uiteindelijk is er gekozen voor API-Platform. Onder de motorkap is API-Platform gebaseerd op Symfony Framework. Het heeft een zeer grote en stabiele basis en is door zijn Symfony roots makkelijk uit te breiden en aan te passen. Daarbij is API-Platform ontwikkeld

5.2 Frontend



Figuur 11: NichePlayer explore pagina

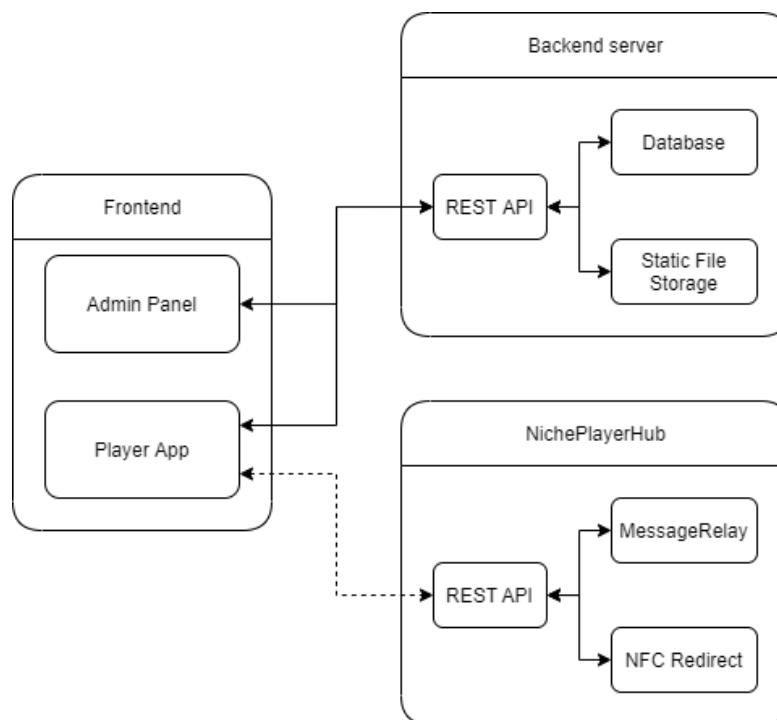
Door positieve ervaringen uit vorige projecten is voor de frontend van de player gekozen voor Quasar. Dit javascript framework gebaseerd op VueJS implementeert de Material Design Guide, waardoor het bouwen van een UI eenvoudig is. Quasar zelf

is geschreven op het VueJS Framework. Binnen VueJS wordt de layout opgebouwd uit components, losse blokken code voor bijvoorbeeld een button of een header bar, wat goed aansluit bij de wensen van een creatief platform. Verder is VueJS reactive, wat betekend dat variabelen als een boomstructuur worden doorgegeven aan de verschillende components. Hierdoor blijft de code overzichtelijk.

Door het gebruik van API-Platform is er een kant en klare admin panel aanwezig op basis van ReactAdmin. ReactAdmin kan vanuit de documentatie van de backend een admin panel genereren. Dit scheelt veel werk. Voor aangepaste layouts is het mogelijk custom pagina's te schrijven.

Om de openheid van de technische implementatie (back- en frontend) van de NichePlayer te onderzoeken is in een periode van een week geprobeerd drie nieuwe functies te implementeren. Als deze functies makkelijk en stabiel konden worden ontwikkeld geeft dit aan dat de basis van het systeem stabiel en uitbreidbaar is. De drie nieuwe functies die hiervoor ontwikkeld zijn, zijn het hebben van een chat in de NichePlayer, het verwerken van NFC kaarten in het systeem, en het mogelijk maken web-audio effecten toe te voegen aan de player.

5.3 Systeemdiagram



Figuur 12: Systeem diagram

Frontend en backend maken een complex geheel. Door middel van een systeemdiagram wordt geprobeerd dit overzichtelijk te maken. **TODO: Aanvullen**

5.4 Resultaten

Het prototype van de NichePlayer is geïmplementeerd volgens de blauwdruk van het systeemdiagram. Met dit prototype ben ik in gesprek gegaan met een viertal medestudenten en enkele docenten. Hierin werd steeds uitgetoetst of het concept van het systeem duidelijk was en aansloot op de ideeën van de ander. Uit deze gesprekken kwamen nieuwe inzichten. Zo bleken enkele features waaronder de fullscreen view van de player niet intuïtief genoeg te vinden te zijn. Ook het idee van de chat functionaliteit is hier geopperd. Het meest opvallende was dat uit veel van deze gesprekken EDM als geschikt doel-genre kwam. Ik ben met de uitkomsten van deze gesprekken gaan doorontwikkelen op basis van de vier hoofdonderwerpen zoals beschreven in hoofdstuk 4 (Methode).

Community

Chat functie

In eerste instantie was er het idee om betalingen en communicatie volledig uit handen te geven aan de artiest. Deze moest met zijn eigen mail contact opnemen met fans over betalingen en andere vragen. Later is bedacht een message functionaliteit gratis mee te leveren met de NichePlayer. In deze chat kan niet alleen worden gepraat met de artiest, ook kunnen de fans onderling met elkaar chatten over albums, festivals of andere gebeurtenissen van de artiest. De gebruikers blijven hierbij in het systeem waardoor het meer een geheel wordt. Een aspect dat de community vorming versterkt is dat gebruikers die tegelijk naar hetzelfde album luisteren automatisch met elkaar worden verbonden via de chat. Deze berichten verdwijnen zodra gebruikers de pagina verlaten.

Gebruikerservaring

ReactAdmin

Omdat tijdens het ontwikkelen van het admin paneel steeds vaker custom pagina's werden geschreven is er voor gekozen toch een admin panel te schrijven in Quasar. Dit heeft het voordeel dat er slechts een ui framework hoeft worden onderhouden. Daarnaast zorgt het samenvoegen voor een gelijke interface en ervaring binnen het systeem. Er hoeft daarbij ook minder dubbel worden ontwikkeld.

Fullscreen view

Na gesprekken met gebruikers bleek dat de fullscreen view van de player niet te vinden was. Deze zou worden geopend als er op de player balk wordt gedrukt. Omdat er geen dedicated knop voor was, en het klikken op de player balk niet een veel voorkomende handeling is, wisten gebruikers niet van het bestaan af. Ik had de gebruikerservaring en interactie niet goed ingeschat.

Businessmodel

In de NichePlayer zijn verschillende businessmodellen verwerkt. De artiest kan bepalen hoeveel hij krijgt en op welke manier er betaald wordt voor de muziek. De betalingen

lopen vervolgens via het systeem van de NichePlayer waardoor het makkelijker is om alles op te zetten.

Als ontwikkelaar biedt ik extra diensten die de NichePlayer van de artiest meer waard kunnen maken. NFC tags zorgen voor een extra fysieke waarde, vergelijkbaar met de cd. Artiesten willen deze extra diensten gebruiken zodat zij meer geld kunnen vragen voor hun muziek. Om te voorkomen dat iedereen zijn account koppelt met één NFC kaart is bedacht dat de kaarten een roulerende toegang bieden. Met roulerende toegang kan worden gezorgd dat bijvoorbeeld maar drie accounts kunnen koppelen met de kaart. De vierde persoon die scant gooit vervolgens de eerste van de lijst er weer uit.

Ontwerp en ontwikkel proces

Schaalbaarheid

Bij het ontwikkelen van het concept is niet nagedacht over de schaalbaarheid van het systeem. In gesprek met enkele developers bleek dat het opschalen of klaarmaken voor een hoger aantal gebruikers veel werk vereist, wat kan worden voorkomen door dit al in de conceptfase mee te nemen. Door de keuze van een enkele php server wordt de webserver zwaar belast en heeft deze al moeite bij simultaan gebruik van drie users. Voor het huidige einddoel is dit geen ramp, er wordt binnen een TRL 4 maar een handvol gebruikers verwacht, maar voor de toekomst is een grote herindeling van het systeem nodig.

6 Conclusie en reflectie

Beoogde doelen technologisch

De beoogde doelen op het technologisch vlak zijn behaald. Er is met success gewerkt tot een Technology Readiness Level 4. Hierbij zijn de Must have en Should have van de MoSCoW allebei uitgevoerd. Resultaat is een stabiel geheel waarmee reeds al gesprekken zijn gevoerd met medestudenten en andere geïnteresseerden. Het systeem is stabiel en wereld toegankelijk op een eigen webserver.

Beoogd Doel creatief platform

In het onderzoek ben ik bezig geweest een creatief platform te ontwikkelen. Deze mate van creativiteit is terug te vinden op verschillende plekken.

Bij het kiezen en ontwikkelen van de technologie is constant in de gaten gehouden of het een open structuur heeft. Dit is gewenst voor een creatief platform. Door verschillende libraries en frameworks iteratief uit te proberen kon ik aanvoelen of het aansloot bij deze wens. Ik ben als maker en rapid prototyper creatief te werk gegaan.

De openheid van het systeem is aangetoond door aan het einde van het ontwikkelen in korte tijd drie nieuwe functionaliteiten toe te voegen. Hiervoor moest het de basis van het systeem worden aangepast. Dit aanpassen ging soepel, wat de openheid van het systeem aantoont.

In het concept van de NichePlayer is veel aandacht besteed aan dat de gebruikers en artiesten creatief met het platform kunnen werken. Voor artiesten is gezorgd dat de NichePlayer zo eenvoudig mogelijk te onderhouden is. Hierdoor kan de ariest zich focussen op wat hij het liefst doen: muziek maken.

Aanbevelingen voor de toekomst

Gebrek aan gebruikerstests

Tijdens het ontwikkelen is vaak van de planning afgeweken. Dit kwam naast persoonlijke omstandigheden ook door meer werk om de basis van het systeem op te zetten. Hierdoor is een belangrijk onderdeel van de ontwikkeling niet gebeurd: gebruikerstests. Pas op het allerlaatste moment zijn een aantal gesprekken gevoerd met luisteraars en artiesten. Hieruit kwamen inzichten die veel eerder in het proces naar voren hadden moeten komen (chat in plaats van mail en EDM als doel-genre). Bij het maken van een systeem voor gebruikers is het belangrijk dat deze gebruikers ook invloed hebben in de ontwikkeling van het systeem. In de toekomst van de NichePlayer zullen er kwalitatieve en kwantitatieve gebruikerstests worden uitgevoerd.

Documentatie en vooronderzoek

Bij het ontwerpen van een complex systeem is het belangrijk om uitgebreid te onderzoeken en te noteren wat allemaal nodig is. In vorige projecten heb ik dit niet vaak

gedaan waardoor deze dan ook vaak zijn vastgelopen. Door dit project op te nemen in dit Supportive Narrative word ik er toe gezet eerst te onderzoeken, en daarna pas te gaan ontwikkelen.

Referenties

- [Goo21] Google. *Material Design Guide*. <https://material.io>. 2021.
- [Inn21] Innovencio. *Technology Readiness Levels (TRL) en subsidieregelingen*. <https://innovencio.nl/blog/readiness-levels/>. 2021.
- [Ite20] Jochem van Iterson. „Een self-hostable streamingdienst, waarom is het nodig?” Appendix A. 2020.
- [Pic21] PicoCMS. *PicoCMS*. <http://picocms.org/>. 2021.
- [Rul21] Peter Rullmann. *MusiCards*. <https://musicards.eu/en/>. 2021.
- [Wit15] Stephen Witt. *How Music got Free*. Vintage Publishing, 2015. ISBN: 9780099590071.
- [Wor21a] Wordpress.com. *Wordpress.com*. <https://wordpress.com/>. 2021.
- [Wor21b] Wordpress.org. *Wordpress.org*. <https://wordpress.org/>. 2021.

Appendices

Een self-hostable streamingdienst, waarom is het nodig?

PAPER M&T JAAR 3

JOCHEM VAN ITERSON

Inhoud

Inhoud	1
Introductie	2
Critical Review	3
Wat maakt dat streamingdiensten werken t.o.v. verkoop van cd's?	3
Geen speciale hardware	3
Meer muziek	3
Nieuwe manier van luisteren	3
Self-hosted en open-source software	4
Waarom heeft een artiest dit systeem nodig?	5
Personaliseren en customizen	5
Verdienmodel voor artiest en ontwikkelaar	5
Fans	6
Te ingewikkeld	6
Wat vinden gebruikers van het idee om nog een streamingdienst te hebben?	7
Meer versnippering	7
Nieuwe manier van ontdekken	7
Ondersteunen artiest	7
Nieuwe functionaliteiten	8
Conclusie	9
Bibliografie	10

Introductie

De afgelopen jaren is de muziekindustrie sterk veranderd. Tot enkele jaren geleden verdienden artiesten hun geld met het verkopen van cd's. Door de opkomst van streaming diensten als Spotify en Apple Music zijn de inkomsten aan cd's echter sterk gedaald. Waar voorheen per album van €15,99 ongeveer €1,60 (Meindertsma 2008) werd verdiend is dit nu via streaming €0,0040 per nummer (Hoe 2013). Voor grote artiesten is dit nog rendabel, maar voor artiesten die meer in een niche zitten is dit niet genoeg om van te leven (Routley 2019). Zij moeten dan gaan verdienen van het doen van optredens en festivals of het verkopen van merchandise. Hierdoor lopen zij de inkomsten van het verkopen van cd's, zowel digitaal als analoog, steeds meer mis.

Zeker in de tijd van Corona hebben deze artiesten het moeilijk (Music Maker 2020). Omdat evenementen massaal worden afgelast en kroegen en andere podia op slot zitten hebben artiesten moeite rond te komen. Waar zij vroeger de verkoop van albums hadden om op terug te vallen moeten zij nu meer creatieve oplossingen vinden om toch geld te verdienen.

Een mogelijke oplossing hiervoor is het ontwikkelen van een alternatieve streamingdienst waar de artiest de controle weer terug krijgt over wat hij wil bereiken en wat hij er aan verdient. Dit is echter lastig omdat er al veel streamingdiensten zijn die ook vaak al erg goed draaien. Er moet dus iets significant nieuws of anders zijn dan wat er al bestaat. Mijn concept is om een self-hostable, opensource streamingdienst te maken. Als inspiratie gebruik ik Wordpress (WordPress sd), een open source pakket wat je zelf op je server kan zetten waarna je je eigen blog hebt. Deze kun je vervolgens uitbreiden en personaliseren met plugins en themes.

De vraag die ik in dit paper wil gaan beantwoorden is de volgende: Is het relevant om een self-hostable streamingdienst te ontwikkelen.

Deze vraag ga ik beantwoorden aan de hand van de volgende deelvragen:

- Wat maakt dat streamingdiensten werken t.o.v. verkoop van cd's?
- Waarom heeft een artiest of producer dit systeem nodig?
- Wat voor effect heeft het hebben van nóg een streamingdienst op de consument?

Om deze vragen te beantwoorden heb ik een vragenlijst gedeeld met artiesten en gebruikers. In deze enquête heb ik het naast bestaande diensten ook over het opzetten van een nieuwe streaming service. In een week tijd heb ik hier meer dan honderd reacties op ontvangen. Deze reacties heb ik vervolgens geanalyseerd, open vragen gelabeld, en conclusies uit getrokken. Resultaten van dit onderzoek staan in de bijlage.

Critical Review

Wat maakt dat streamingdiensten werken t.o.v. verkoop van cd's?

Het is goed om te analyseren waardoor streamingdiensten in de eerste plaats zo groot zijn geworden. Daarbij is het interessant om de opkomst van online muziek te bekijken. Het boek *How Music Got Free* van Stephen Witt (Witt, *How Music Got Free* 2015) geeft een goede beschrijving van de tijd van fysieke distributie van muziek tot de opkomst van de streamingdiensten. In de jaren voorafgaand aan streamingdiensten werd er veel muziek gedownload. Dit gebeurde veelal illegaal via diensten als Napster en IRC kanalen. (Witt, *IRC channels* 2015) Er werd in reactie daarop via een meer legale weg muziek als download aangeboden. Een voorbeeld van zo'n dienst is iTunes van Apple waar je een album koopt en download op een apparaat. Deze bestanden kun je dan wel alleen via iTunes afspelen.

Geen speciale hardware

Internet heeft een grote invloed gehad in hoe we naar muziek luisteren. Zo is het bezitten van muziek veel toegankelijker geworden. Waar vroeger speciale hardware nodig was om een album te kunnen luisteren kan dat nu vanaf vrijwel elk apparaat op ieder moment. Een nadeel is wel dat er erg veel opslag nodig was om al je muziek op te kunnen slaan. Streamingdiensten hebben dit probleem opgelost door gebruikers de muziek slechts tijdelijk te laten opslaan en het na afspelen weer te kunnen verwijderen (cachen). Het enige wat nu nog nodig is is een betaalbare, stabiele internetverbinding. Als men toch zonder internet wil luisteren kan dat ook, maar voor een bepaalde tijd.

Meer muziek

Daarmee krijgt de luisteraar met een abonnement op een streamingdienst toegang tot miljoenen nummers van duizenden artiesten. Dit abonnement is ook vele malen goedkoper dan als al de muziek afzonderlijk zou moeten worden gekocht of gehuurd. Toegang tot miljoenen nummers heeft ook weer een keerzijde. Waar moet men naar luisteren? Door in de dienst een algoritme te verwerken wat het luistergedrag van een gebruiker analyseert kan er vergelijkbare muziek worden voorgesteld. Hierdoor ontdekt de gebruiker nieuwe muziek.

Nieuwe manier van luisteren

Een opvallend effect is dat er minder wordt geluisterd naar albums maar meer naar de losse nummers. Vroeger als men een album kocht luisterde hij deze van begin tot eind, geen nummer werd overgeslagen. Nu wordt er vooral geluisterd naar (zelf samengestelde) playlists met muziek van meerdere artiesten door elkaar. Dit heeft er daarnaast ook voor gezorgd dat nummers steeds korter zijn geworden (Raatgever 2019). Dit komt waarschijnlijk vooral omdat een artiest betaald krijgt per keer dat het wordt afgespeeld. Door het nummer korter te maken kan hij vaker worden gedraaid.

Self-hosted en open-source software

Er is veel software geschreven. De laatste decennia is open-source software steeds prominenter geworden (Open source versus closed source 2015). Open-source houdt in dat de code publiekelijk toegankelijk is waardoor er met meerdere (enthousiaste) programmeurs aan kan worden gewerkt. Dit heeft meerdere voordelen. Omdat de code voor iedereen toegankelijk is kan er makkelijker een bug worden opgelost. Ook is het bij open-source projecten vaak het geval dat het uit te breiden is door plugins.

Voor het uitleggen van self-hosted systemen neem ik WordPress als uitgangspunt.

Wordpress is een pakket waarmee snel een eigen blog kan worden opgezet. Het installeren is zo simpel als bestanden op een server zetten en enkele instellingen aanpassen. Door pakketten op deze manier aan te bieden krijgt de gebruiker meer controle over hoe het systeem gaat gedragen en wat de content is die er op komt te staan. Een tweede voordeel is dat het minder bandbreedte nodig heeft ten opzichte van reguliere hosters. In plaats van een enkele server zijn er nu honderden losse waarop blogs kunnen worden gehost.

Waarom heeft een artiest dit systeem nodig?

Hoewel het lijkt dat streaming en downloaden een uitkomst is voor de muziekindustrie, is dit in de praktijk meer genuanceerd: voor de industrie en de gebruiker is het fijn, voor de maker (artiest en producer) over het algemeen niet zo gunstig. Doordat muziek nu via een (commercieel) tussenmedium wordt gedistribueerd verdienen artiesten en producers minder per nummer. Waar voorheen aan een album €1,60 werd verdiend is dit nu €0,0040 per stream. Dit betekent dat een nummer vaker gedraaid moet worden om dezelfde hoeveelheid geld op te brengen.

Het systeem wat ik voor wil stellen is een self-hostable, opensource streamingdienst die de artiest zelf op een server kan zetten. Door weg te gaan van het gecentraliseerde model van huidige diensten als Spotify en iTunes is het mogelijk een meer divers en gevarieerde online muziek cultuur te krijgen. Een probleem met bandbreedte is er bijna niet. Er hoeft door het gedecentraliseerde model maar weinig data te worden opgehaald.

Een voorbeeld van een pakket wat bestaande systemen heeft gedecentraliseerd is WordPress. Na het uploaden en installeren op de server kan er gelijk worden begonnen aan het vullen en personaliseren van je eigen (WordPress) blog. Naast dat WordPress kan worden geïnstalleerd op de eigen server kan er ook voor worden gekozen te hosten bij wordpress zelf. Dit principe wil ik ook gaan implementeren voor artiesten met minder kennis of interesse van servers en websites.

Personaliseren en customizen

Dit pakket is in tegenstelling tot andere streamingdiensten uit te breiden met plugins en thema's. Hiermee kan het systeem meer worden gepersonaliseerd en worden uitgebreid met extra functionaliteiten.

Een voorbeeld van een plugin is een interactieve player waarin het volume van verschillende partijen kan worden veranderd. Hierdoor kan een gebruiker meespelen met de muziek door het instrument wat hij speelt uit te zetten in de multitrack. Ook is het mogelijk om een plugin te maken waarmee in de app een remix kan worden gemaakt met de nummers die worden aangeboden.

Ook kan er een lyrics functie worden ingebouwd waarmee je live songteksten kan meelezen met de muziek. Deze functie heeft een tijd in Spotify gezeten, maar is na een recente update verwijderd.

Door het systeem open source te maken kunnen ontwikkelaars ook hun eigen plugins schrijven en uitgeven. Hiermee is het naast een simpele streaming service ook een plek waar technologie kan worden uitgetoetst.

Verdienmodel voor artiest en ontwikkelaar

Door zelf het distributie medium aan te bieden kan ook worden bepaald op welke manier er geld mee wordt verdiend door de artiest. Iedere fanbase is anders en wil of kan op een andere manier de artiest steunen. Door verschillende businessmodellen aan te bieden kan hier zo goed mogelijk op worden ingespeeld. Zo kan er gekozen worden om (net als het businessmodel van iTunes) te laten betalen per album of nummer, waarna je toegang krijgt deze te mogen streamen. Ook kan er worden gekozen voor een abonnementsdienst zoals

bij spotify en deezer, een maandelijkse subscription geeft toegang tot alle content die beschikbaar is via het systeem. Verder is er nog een model mogelijk waarbij eenmalig wordt betaald voor toegang tot het systeem. Dit zou dan vergelijkbaar zijn met het kopen van een album, maar dan met meer digitale functionaliteiten als toegang tot extra content. Tot slot kan er betaald worden per stream, je betaald voor hoe lang je hebt geluisterd. Dit laatste zou vergelijkbaar zijn met hoe (internet) radio op het moment betaald voor muziek. Voor mij als ontwikkelaar zijn ook meerdere verdienmodellen mogelijk. Ik kan naast het aanbieden van de broncode ook een service aanbieden waarop ik als hoster fungeer. Zo hoeft de artiest niet zelf een server te beheren en kan zich meer focussen op het vullen er van. Een ander model is het aanbieden van betaalde plugins en extensies die het systeem beter maken. Tot slot kan er geld worden gevraagd voor het delen op een hub. Hiermee krijgt de artiest meer publiciteit en kan hij makkelijker worden gevonden.

Fans

Een bijkomend voordeel van zelf hosten is de mogelijkheid tot contact met fans. Door in het systeem een chatdienst te verwerken kunnen fans met elkaar of met de artiest praten over de muziek. Ook kan via het systeem speciale content worden verdeeld aan fans. Denk hierbij aan demo's of backstage video's, maar ook kaartjes en andere fysieke merchandise kan via het systeem worden verkocht. De content kan ook worden gekoppeld aan bepaalde condities. Zo kan de content bijvoorbeeld ook worden gekoppeld aan een fysieke locatie (de gebruiker moet aanwezig zijn (geweest) op een festival om later de extra content te kunnen bekijken), hoe vaak de gebruiker de app opent, of op aanmelding in een bepaald tijdsframe.

Te ingewikkeld

Een risico voor een dergelijk systeem zou zijn dat het te ingewikkeld is om te installeren en te beheren. Artiesten zouden zich beter kunnen richten op het maken van de muziek en het distribueren aan de mensen overlaten aan de mensen die daar weer verstand van hebben. De uitdaging voor dit systeem is dus om het zó eenvoudig te maken dat het deze drempel niet wordt overschreden. Zo kan er naast het zelf moeten installeren ook een service worden aangeboden die alles voor de artiest opzet. Een ander probleem wat zich voordoet bij zelfhosten is dat er beperkt opslag beschikbaar is. Dit kan worden opgelost door te kunnen koppelen andere diensten als Dropbox en andere cloudstorage.

Wat vinden gebruikers van het idee om nog een streamingdienst te hebben?

Uit mijn onderzoek blijkt dat niet iedereen zit te wachten op nog een streamingdienst. Iets minder dan de helft van de mensen zegt niets te zien in een nieuwe dienst. De voornaamste reden hiervoor is dat ze vinden dat de streaming industrie al oververzadigd is. Meer dan de helft is wel enthousiast. Zij zien een nieuwe manier van muziek ontdekken en ondersteunen van de artiest.

Meer versnippering

Er is wel een groot nadeel als artiesten hun eigen diensten gaan aanbieden. De markt is op het moment al redelijk versnipperd (sommige artiesten zitten op Spotify, anderen alleen maar op Apple Music, etc). Als artiesten hier hun eigen systemen aan toevoegen wordt dit alleen maar erger. Daarnaast moet de consument meer betalen en krijgt hij naar zijn gevoel hetzelfde voor terug.

Voor dit probleem zijn meerdere oplossingen. Zo kan er simpelweg voor worden gekozen dat het systeem niet iedereen zal gaan aanspreken. Het zal voornamelijk een luxeproduct gaan worden voor fans die de artiest willen ondersteunen.

Om een groot deel van de versnippering aan te pakken kan er een hub komen waar alle artiesten die het systeem gebruiken in worden meegenomen. De muziek is dan nog wel zelf gehost maar kan makkelijker worden gevonden. Ook hoeft de gebruiker niet steeds te wisselen van app als hij een andere artiest wil gaan luisteren. In de hub kan ook een koppeling worden gemaakt met andere diensten. Hierdoor is het niet alleen een verzamelplek voor het systeem maar kan het ook fungeren als een plek om nieuwe muziek te vinden.

Nieuwe manier van ontdekken

Huidige streamingdiensten zijn steeds meer gericht op de populaire muziek. Op de voorpagina staan releases van grote artiesten, voorgestelde afspeellijsten bevatten voornamelijk de populaire muziek en de afspelen radio (functie in Spotify als de gekozen lijst is afgelopen) gaat door met meer van hetzelfde. Hierdoor komen kleine artiesten die iets eigens maken minder naar voren. Door een eigen dienst aan te bieden krijgen zij hierin een grotere positie. Zij kunnen hun eigen pr doen en fans voorstellen doen van wat zij denken dat goed bij hun muziek past. Hierbij kan een hub ook er voor zorgen dat er een letterlijke koppeling is tussen de artiesten.

Ondersteunen artiest

Veel gebruikers realiseren zich dat artiesten weinig verdienen aan streamingdiensten. Om een artiest te ondersteunen kopen zij LPs, CDs en andere merchandise, of gaan ze naar een festival of ander live optreden. Door een systeem aan te bieden wat speciaal ontworpen is voor de artiest komt de fan veel dichterbij te staan. Zij hebben meer het gevoel de artiest zelf te ondersteunen dan dat zij meebetalen aan de inkomsten van de populaire artiesten.

Door de merch in de app zelf aan te bieden krijgen fans ook meer het gevoel dat ze de spullen kopen bij de artiest zelf. Naast het financieel steunen van de artiest kan er ook makkelijker contact worden gezocht. Dit is mogelijk door een chatfunctie in te bouwen waarop fans en artiesten met elkaar kunnen praten en bijvoorbeeld remixes kunnen delen.

Nieuwe functionaliteiten

Hoewel gebruikers weinig behoefte hadden aan een extra streamingdienst, kwamen ze wel met veel verbeterpunten. Een veel voorkomende was de vraag naar een songtekst functie. Deze is recent uit Spotify weggehaald, maar wordt in veel gevallen toch wel gemist. Waarschijnlijk is dit gebeurd omdat het opvragen van kwalitatieve lyrics te veel moeite bleek te zijn. Omdat de artiest hier zelf de muziek deelt kan hij ook makkelijker passende songteksten meegeven.

Een ander idee is om het voorspellend algoritme meer aanpasbaar te maken. Zo kan er naast het luistergedrag ook een score worden gegeven aan een nummer. Ook kan een gebruiker zelf een nummer labelen. Een derde vraag die uit de lijst kwam was het kunnen streamen van zelf geüploade muziek. Spotify kan wel lokaal nummers afspelen maar biedt deze vervolgens niet aan op andere apparaten met hetzelfde account. Tot slot wilde gebruikers binnen een gegenereerde afspeellijst bepaalde nummers altijd kunnen skippen. Op dit moment moet de gebruiker ofwel handmatig skippen of eerst de hele afspeellijst kopiëren om vervolgens die nummers er uit te halen.

Omdat het systeem open source en self hostable is, is het mogelijk om plugins te ontwikkelen. In het vorige hoofdstuk geef ik hier enkele voorbeelden van.

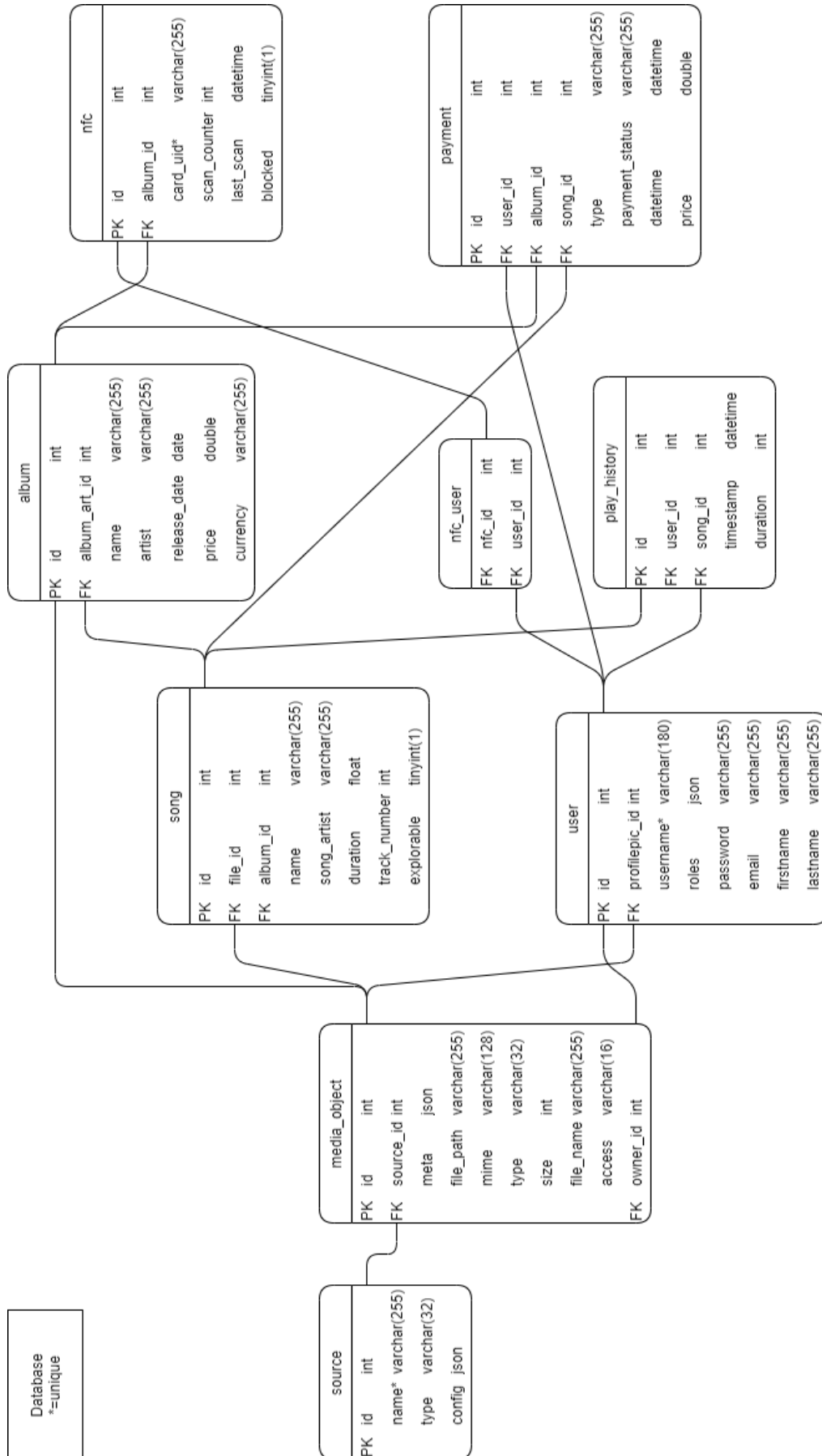
Conclusie

Het maken van een open source streamingdienst zal niet voor iedereen werken. Het is moeilijk te concurreren met bestaande grote diensten. Daarnaast willen artiesten zich vaker meer kunnen richten op het maken van de muziek, een eigen distributiekanaal moeten opzetten en beheren lijkt hen te veel werk. Door een tussenvorm aan te bieden kunnen zij het systeem nog steeds gebruiken zonder zich druk te hoeven maken om serveronderhoud. Toch zijn er ook veel voordelen. Artiesten krijgen door het pakket zelf aan te bieden meer controle over hoe het eruitziet en hoe veel zij eraan kunnen verdienen, de 'middle man' wordt verwijderd. Hiermee komen zij ook nog veel dichterbij de fan te staan. Dit is voornamelijk voor de niche artiesten interessant gezien zij meestal een wat kleinere fanbase hebben. Daarnaast kan systeem ook gaan werken als een plek om nieuwe technologie voor distributie van muziek uit te proberen. Een functie die bestaande diensten niet hebben is een algemene hub die alle installaties van het systeem bundeld. Op de vragenlijst kwam als kritiek dat het moeilijk zou zijn als alle artiesten apart gaan concurreren met de grote diensten. Door te bundelen door middel van een hub is dit meer centraal en krijgt het hele systeem een eigen karakter.

Bibliografie

2020. 15 03. Geopend Mei 11, 2020. <https://www.musicmaker.nl/nieuws/artikel/2-23706/corona-en-het-drama-voor-de-professionele-muzikant>.
- Hoe, Colin van. 2013. *Spotify betaalt rechthebbenden...* 03 December. Geopend Mei 11, 2020. <https://www.nu.nl/tech/3644197/spotify-betaalt-044-062-eurocent-per-stream.html>.
- Meindertsma, Peter. 2008. *Hoeveel verdient een artiest per CD?* 22 04. Geopend Mei 11, 2020. <https://www.petermeindertsma.nl/blog/hoeveel-verdient-een-artiest-per-cd/>.
2015. *Open source versus closed source*. 09 Oktober. Geopend Mei 11, 2020. <https://www.brightanswers.eu/nl/academy/open-source-versus-closed-source/index.xml>.
- Raatgever, Stefan. 2019. *Artiesten zetten schaar in muziek door Spotify*. 24 Maart. Geopend Mei 11, 2020. <https://www.parool.nl/amsterdam/artiesten-zetten-schaar-in-muziek-door-spotify~bb970a00/>.
- Routley, Nick. 2019. *How Many Music Streams Does it Take to Earn a Dollar?* 12 September. Geopend Mei 11, 2020. <https://www.visualcapitalist.com/how-many-music-streams-to-earn-a-dollar/>.
- Witt, Stephen. 2015. In *How Music Got Free*, door Stephen Witt. Vintage.
- Witt, Stephen. 2015. „IRC channels.” In *How Music Got Free*, 105. Vintage.
- sd. *WordPress*. Geopend Mei 11, 2020. <https://nl.wordpress.org>.

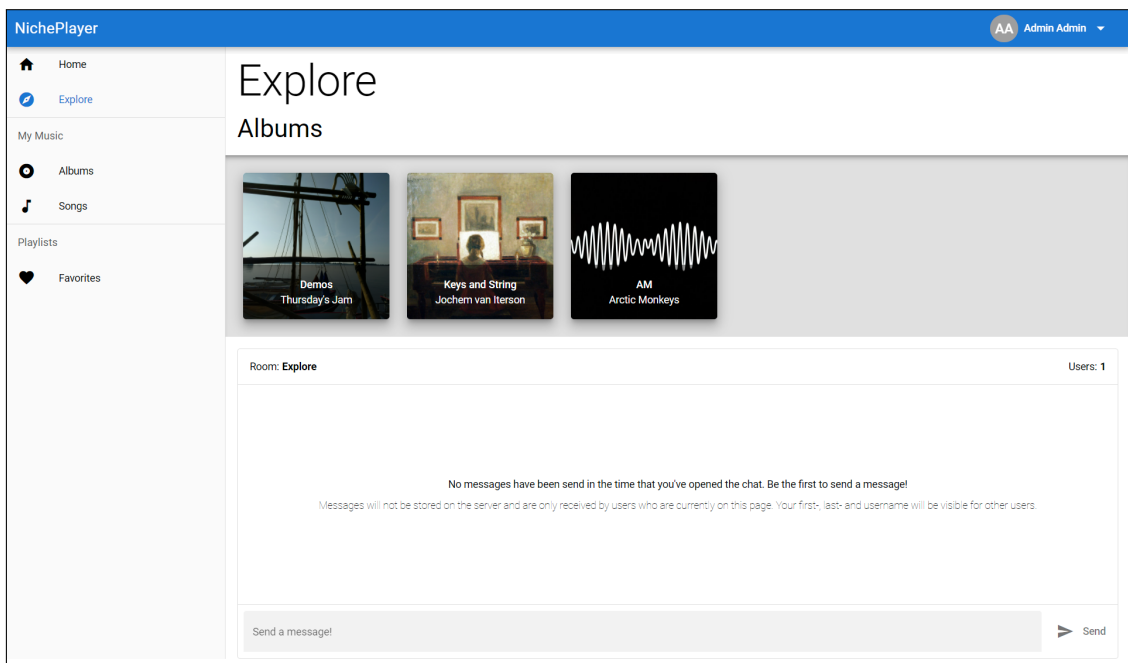
B DATABASE DIAGRAM NICHEPLAYER



C Screenshots NichePlayer



NichePlayer Login screen



NichePlayer Explore page

The screenshot shows the NichePlayer web interface. The top navigation bar is blue with the 'NichePlayer' logo on the left and a user profile 'AA Admin Admin' on the right. A left sidebar contains navigation links: Home, Explore, My Music (with sub-links for Albums and Songs), Playlists, and Favorites. The main content area features an album preview for 'Demos' by 'Thursday's Jam' with a background image of sailboats. Below the preview, a 'Purchase' section shows a price of '€10' and a 'Buy Album' button. A 'Tracks' section below that contains a table with three tracks. A 'Preview album' button is located at the top right of the tracks section.

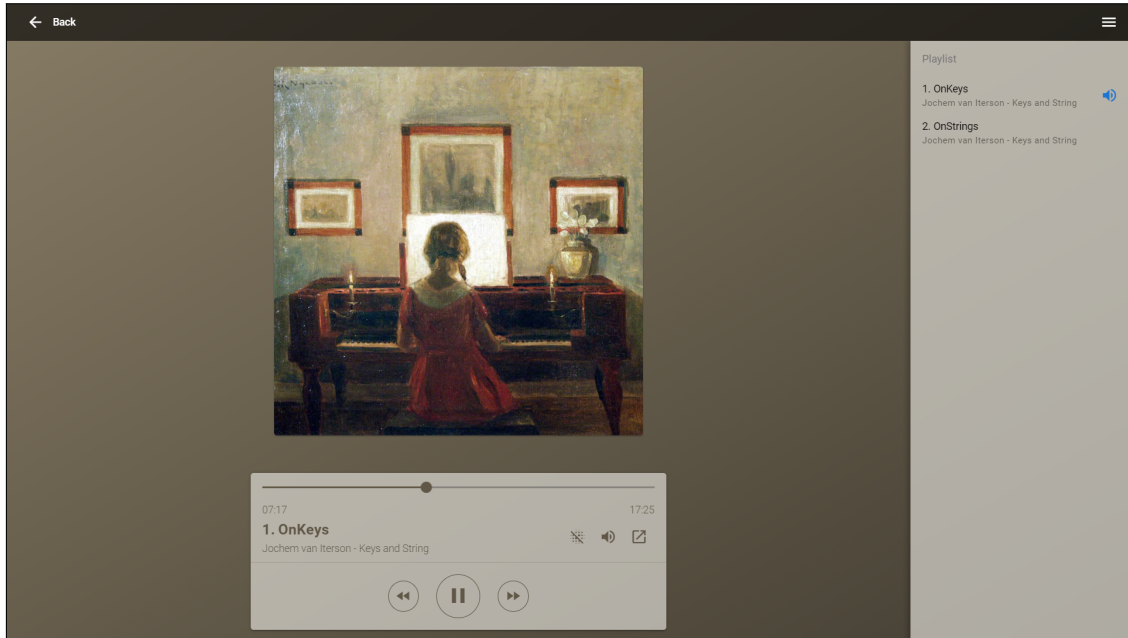
Track	Title	Artist	Duration
1	I Wish	Thursday's Jam	03:18
2	Sunday Morning	Thursday's Jam	04:12
3	Something's got a hold on me	Thursday's Jam	03:30

NichePlayer Unpurchased album

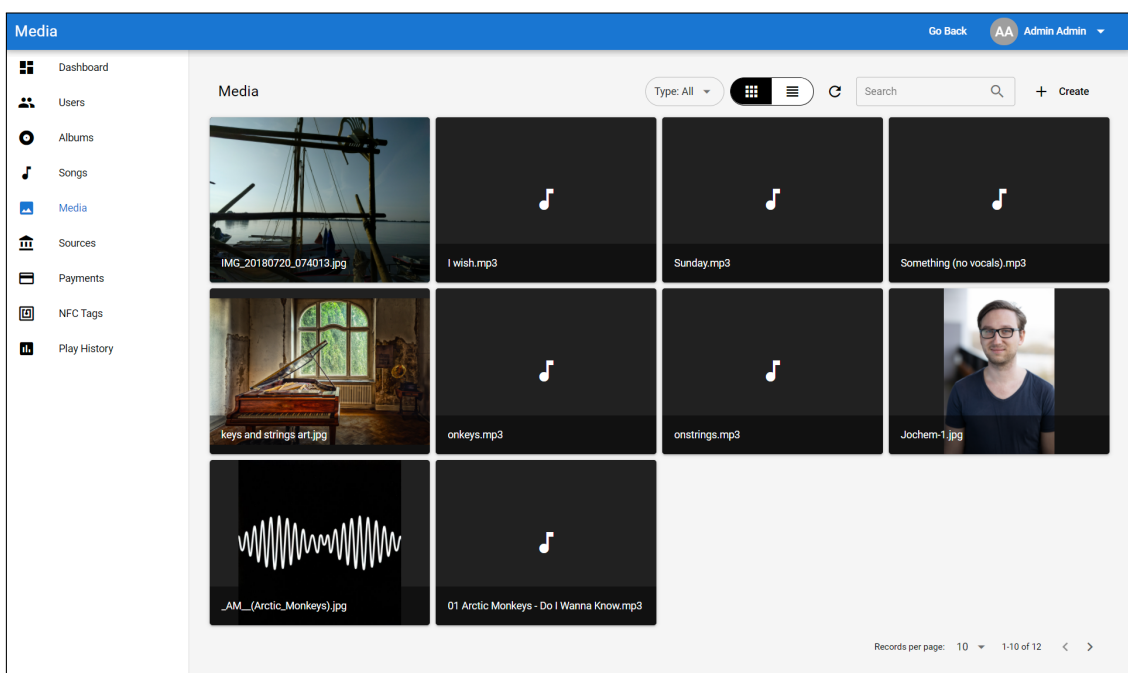
This screenshot shows the NichePlayer interface for a purchased album, 'Keys and String' by 'Jochem van Iterson'. The layout is similar to the previous one, but the 'Purchase' section is replaced by a 'Play album' button. Below the tracks table, a chat room interface is visible, titled 'Room: Jochem van Iterson - Keys and String' with 'Users: 1'. The chat area contains a message prompt and a text input field with a 'Send' button. At the bottom, a music player bar shows the current track '1. OnKeys' by 'Jochem van Iterson' and includes standard playback controls.

Track	Title	Artist	Duration
1	OnKeys	Jochem van Iterson	17:25
2	OnStrings	Jochem van Iterson	09:52

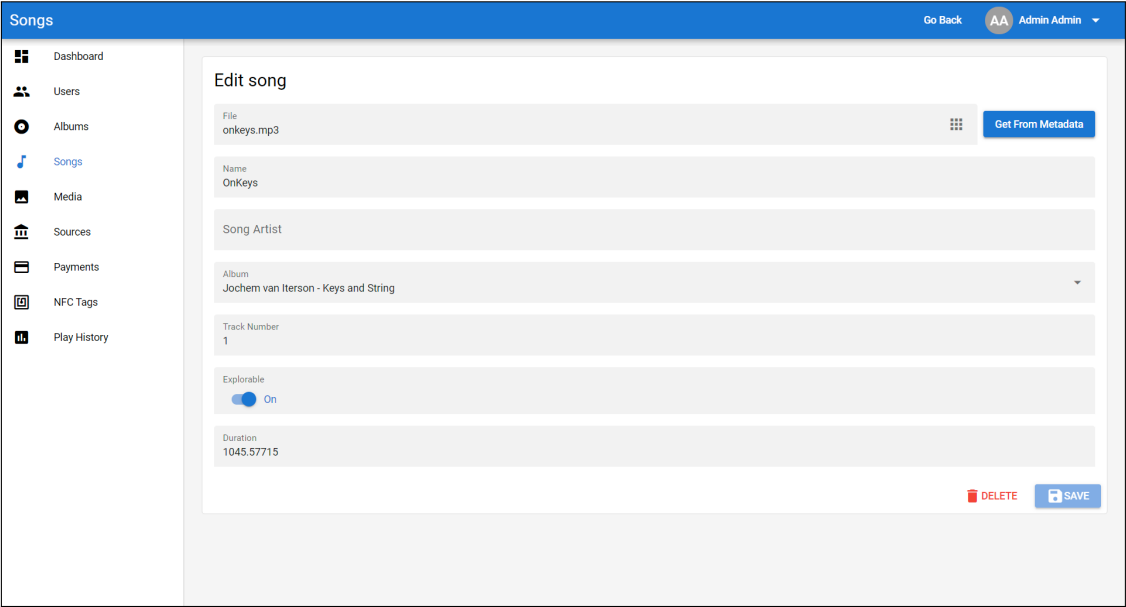
NichePlayer Purchased album



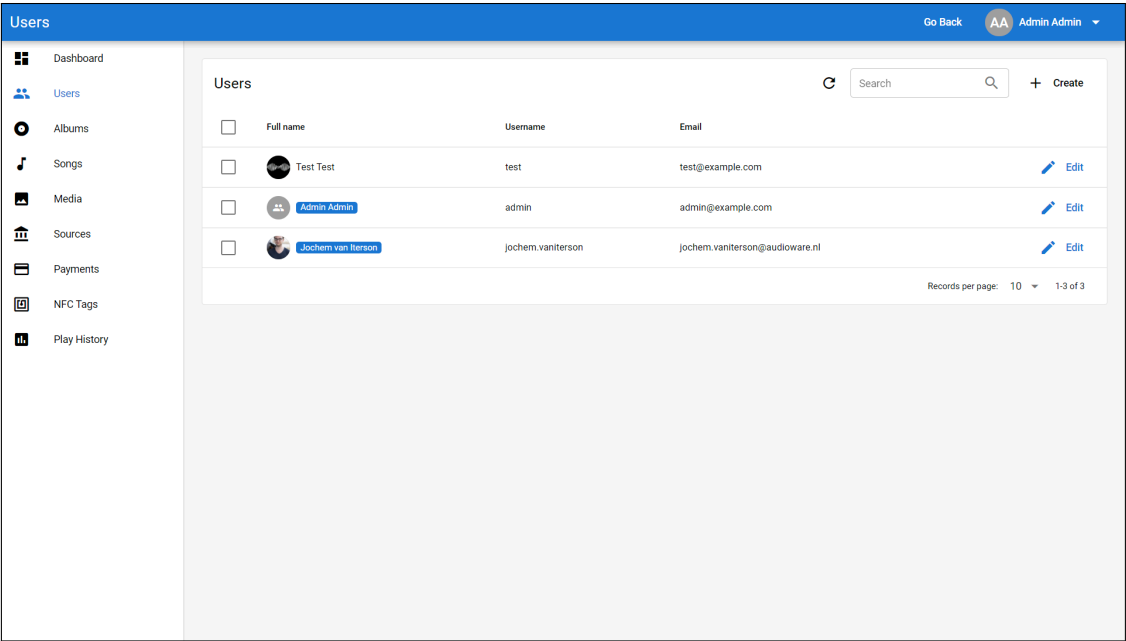
NichePlayer Fullscreen Player



NichePlayer Adminpanel Media



NichePlayer Adminpanel Song editor



NichePlayer Adminpanel User list