



STAGEVERSLAG

Push notificaties in een low-code platform

Marten Krings – 412293 – Saxion
412293@student.saxion.nl

VOORWOORD

Dit is het afstudeerverslag “Push notificaties in een low-code platform”. Het onderzoek naar en implementatie van deze push notificaties is uitgevoerd bij het bedrijf Thinkwise. Dit verslag is geschreven in verband met mijn afstuderen aan de opleiding HBO-ICT Software Engineering aan de Hogeschool Saxion. Deze afstudeerstage duurde vanaf februari 2019 tot en met juni 2019.

De opdracht is bedacht in het kader van een openstaand project binnen Thinkwise en de vernieuwing van de gebruikersinterface die hier wordt ontwikkeld.

Graag wil ik Nick Matlung bedanken voor zijn paraatheid om mijn vragen te beantwoorden en Jasper Kloost voor zijn begeleiding tijdens mijn stage.

Marten Krings

Apeldoorn, 17 juni 2019

INTRODUCTIE

Push notificaties worden op mobiele telefoons al langere tijd gebruikt om gebruikers te bereiken en deze tijdig te informeren over belangrijke informatie en om gebruikers opnieuw te betrekken bij een applicatie. Er zijn inmiddels verschillende services die push notificaties verzorgen voor verschillende platformen, met verscheidene doelen in gedachten.

Thinkwise wil deze functionaliteit ook aanbieden, bijvoorbeeld om Twee-factor-authenticatie aan te bieden. Ten behoeve van dit doel moet er worden bepaald hoe deze implementatie tot stand zal komen. Er zijn verschillende technologieën en frameworks die mogelijk gebruikt kunnen worden om deze functionaliteit te realiseren.

Er moet worden bepaald welke technologieën in overweging genomen zullen worden, welke gebruikt gaat worden, en er moeten prototypes gebouwd worden met deze technologie.

Vanwege de complexiteit van het Thinkwise Platform ligt de eventuele integratie van push notificaties buiten de scope van deze opdracht.

LEESWIJZER

Deze Leeswijzer is bedoeld om de lezer aan te geven wat het doel is van elk hoofdstuk en welke informatie het bevat.

Ten eerste zal het bedrijf waar de stage plaats heeft gevonden worden geïntroduceerd. Vervolgens zullen de aanleiding voor deze opdracht en de belanghebbende partijen worden benoemd. Ook de eisen aan het eindproduct zullen worden uitgelegd.

Daarna zullen de onderzoeksvragen die zijn beantwoord ten behoeve van de uitvoering van de opdracht worden verklaard. Vervolgens zal er worden ingegaan op het ontwerp van de prototypes die zijn gebouwd als resultaat van deze opdracht.

Ten slotte zullen de conclusies en aanbevelingen uiteen worden gezet om het verslag af te ronden.

INHOUD

Voorwoord	1
Introductie	2
Leeswijzer	2
Begrippenlijst	5
Samenvatting	6
Probleemstelling	6
Onderzoek	6
Resultaat	6
Achtergrond	7
Thinkwise	7
Probleemstelling	9
Aanleiding	9
Probleem	9
Belanghebbende partijen	10
Requirements	11
Functionele Requirements	11
Non-functionele Requirements	12
Methodiek	13
Tijdslijn	13
Scrum	14
Onderzoek	15



Wat is Model Gedreven Ontwikkeling?	15
Wat zijn push notificaties?	16
Wat zijn WebSockets?	17
Wat is het Thinkwise Platform?	18
<i>Thinkwise</i>	18
Wat is Cordova?	21
Wat zijn Progressive Web Apps?	22
Analyse	23
Prijsvergelijking.....	28
Conclusie & Aanbevelingen.....	30
Ontwerp	32
Back-end	32
Front-End	34
Ontwerp	34
Conclusie	36
Aanbevelingen	37
Literatuurlijst	39
Versiebericht.....	44
Bijlage 1	45
Backlog.....	45
Bijlage 2	46
Functioneel Ontwerp	46



BEGRIPPENLIJST

De begrippenlijst bevat een lijst termen en hun definities en dient als referentie.

Begrip	Definitie
ERP	Enterprise Resource Planning, software die gebruikt wordt ter ondersteuning van bedrijfsprocessen, bijvoorbeeld administratie of het bijhouden van voorraden.
Push Notificatie	Een notificatie die naar een applicatie gestuurd wordt en deze kan aanspreken zonder dat deze actief is.
Progressive Web App	Een website die geïnstalleerd kan worden op een telefoon of desktop als standaard applicatie, en offline benaderd kan worden.
Cordova	Een framework dat programmeurs in staat stelt in één keer voor verschillende mobiele platformen te ontwikkelen zonder meerdere applicaties te hoeven ontwikkelen.
API	Application Programming Interface, een koppeling tussen meerdere applicaties waardoor ze met elkaar kunnen communiceren.
HTTPS	Hypertext Transfer Protocol Secure, de versleutelde versie van het Hypertext Transfer Protocol dat gebruikt wordt om webverkeer te versturen.
Run-time	De tijd dat een applicatie aan staat.
Multitierarchitectuur	Multitierarchitectuur is een architectuur voor het bouwen van client server applicaties waarbij een applicatie wordt opgedeeld in een aantal onafhankelijke lagen, zodat deze flexibeler te deployen zijn. (Microsoft, 2014)

SAMENVATTING

Thinkwise bouwt een low-code, of model gedreven ontwikkelplatform om ERP-software te realiseren. Dit platform (de Thinkwise Software Factory) stelt de ontwikkelaar in staat om ERP-software te bouwen, waarbij bijna geen, of zelfs geen programmeerwerk nodig is.

Probleemstelling

Thinkwise heeft al een aantal jaren de vraag gekregen van klanten om push notificaties te implementeren. Dit begon bij het inzetten van hun eerste mobiele applicatie. Er was toen vraag voor de mogelijkheid een notificatie te sturen wanneer de agenda van een medewerker was aangepast. Deze vraag kwam later bij andere klanten terug, toen zij bijvoorbeeld notificaties wilden versturen wanneer een offerte goedgekeurd moest worden.

Graag zou Thinkwise deze functionaliteit aanbieden om sturing te geven binnen applicaties die gebouwd zijn met de Software Factory. Deze implementatie moet ontwikkelaars die het Thinkwise Platform gebruiken in staat stellen eenvoudig zonder programmeerwerk push notificaties toe te voegen aan een applicatie.

Onderzoek

Ten eerste moest er bepaald worden welke technologie en/of framework gebruikt moest worden om push notificaties te implementeren. Daarom is er exploratief onderzoek uitgevoerd om verschillende mogelijkheden in kaart te brengen. Hieruit is een lijst opties gekomen die dit mogelijk konden verzorgen..

Vervolgens is er, door een combinatie van deskresearch en laboratoriumonderzoek, bepaald welke opties aan alle eisen konden voldoen, en is er na een prijsvergelijking een besluit genomen voor één framework.

Resultaat

Uiteindelijk is er gekozen voor Firebase. Dit framework is gebruikt om aan alle gestelde eisen aan push notificaties te voldoen. Het is geïmplementeerd in de vorm van een Progressive Web App, een Cordova Applicatie, en een API.

ACHTERGROND

Dit hoofdstuk dient om het stagebedrijf aan de lezer te introduceren.

THINKWISE

Thinkwise is een Nederlands softwarebedrijf dat de Thinkwise Software Factory bouwt. De Thinkwise Software Factory is een ontwikkelplatform waarmee enterprise software snel en simpel gerealiseerd kan worden.

Thinkwise is opgericht in 2002 door Robert van der Linden en Victor Klaren. Zij vonden destijds een gat in de markt toen zij zich realiseerden dat professionele enterprise software in gebruik vaak inflexibel en verouderd was, ondanks het feit dat deze software cruciaal was voor het succes van het bedrijf dat het in gebruik had.

Het vernieuwen en bouwen van deze software is een grote kostenpost, kost veel tijd, en brengt risico met zich mee in de vorm van bijvoorbeeld bugs. Robert en Victor besloten hun ervaring in het realiseren van Enterprise software toe te passen in een nieuw product. Dit product werd uiteindelijk de Thinkwise Software Factory.

De Thinkwise Software Factory is een ontwikkelplatform om model gedreven software te realiseren. Dit houdt in dat de applicatie gemodelleerd wordt in de Software Factory, vervolgens wordt dit model op runtime geïnterpreteerd en omgezet tot een functionele applicatie. Dit model is technologie onafhankelijk. Daardoor kan Thinkwise de onderliggende technologie gemakkelijk vervangen en vernieuwen.

Het modelleren van een applicatie kost minder tijd dan het bouwen van een applicatie en vereist geen software engineers, waardoor applicaties sneller en goedkoper kunnen worden gerealiseerd. Daarnaast zorgt het hergebruik van bewezen componenten die het model interpreteren ervoor dat het risico op bugs in de software afneemt.

De Software Factory is het core product van Thinkwise. Thinkwise neemt opdrachten aan om software te realiseren, maar verkoopt inmiddels ook licenties aan andere

softwareleveranciers. Thinkwise is op dit moment hard aan het groeien en is recentelijk verhuisd naar een nieuw pand om deze groei te kunnen accommoderen.

Thinkwise bestaat momenteel uit 155 werknemers. De stage afdeling, Product Innovation, bestaat momenteel uit meer dan 20 werknemers.

PROBLEEMSTELLING

Dit hoofdstuk gaat in op het probleem dat opgelost moet worden en beschrijft de uiteindelijke producten.

AANLEIDING

Toen Thinkwise voor het eerst een mobiele applicatie inzette, en sindsdien meer, kwam er van hun klanten de vraag om push notificaties te gebruiken. Dit staat al een aantal jaren gepland, maar Thinkwise heeft dit tot nu toe nooit opgepakt.

Destijds was er de vraag om de gebruiker van een applicatie een notificatie te sturen wanneer de agenda van de gebruiker werd aangepast. Daarnaast wil Thinkwise graag push notificaties inzetten om gebruikers sturing te geven binnen gebouwde applicaties, bijvoorbeeld door herinneringen te sturen voor taken die een deadline hebben, zoals het boeken van uren. De push notificatie zou de gebruiker dan direct naar de juiste locatie in de applicatie moeten kunnen navigeren.

PROBLEEM

Het Thinkwise Platform is een model gedreven (low-code) ontwikkelplatform voor grote enterprise systemen, zoals enterprise resource planning. Met deze opdracht wil Thinkwise op een generieke manier push notificatie functionaliteit aanbieden die in te zetten is in alle applicaties die ontwikkeld worden met het Thinkwise Platform en op alle ondersteunde devices: Desktop, Web en Mobile.

Vanwege het feit dat het Thinkwise Platform een low-code ontwikkelplatform is, moet hier rekening mee gehouden worden in het selecteren van een framework, Het moet programmatisch te configureren zijn, zodat deze complexiteit weggenomen kan worden voor ontwikkelaars die het Thinkwise Platform gebruiken.

Daarnaast moet de onderliggende technologie van de implementatie gemakkelijk te vervangen en te updaten zijn, om deze toekomstbestendig te maken.

BELANGHEBBENDE PARTIJEN

De partijen die baat hebben bij deze opdracht zijn als volgt:

Thinkwise – De eerste belanghebbende partij is Thinkwise zelf. Thinkwise gebruikt de software factory ook voor interne software, het bedrijf heeft er dus baat bij om push notificaties te kunnen gebruiken. Daarnaast is het in het voordeel van Thinkwise deze functionaliteit aan te kunnen bieden aan hun klanten en licentiehouders.

Klanten Thinkwise – De klanten van Thinkwise hebben baat wanneer updates en nieuwe functionaliteit ingezet kunnen worden. Push notificaties waren een vraag vanuit een klant, er zal aan deze vraag kunnen worden voldaan.

Licentiehouders Software Factory – De licentiehouders van de Software Factory zullen deze nieuwe functionaliteit in kunnen zetten bij de software die zij voor derde partijen ontwikkelen.

REQUIREMENTS

Dit hoofdstuk beschrijft de eisen aan de oplossing die gerealiseerd moet worden aan het einde van de stage.

FUNCTIONELE REQUIREMENTS

Must

Real-time push notificaties

De notificaties moeten in real-time ontvangen worden door de gebruiker.

Push notificaties naar specifieke gebruikers en groepen

Push notificaties moeten verstuurd kunnen worden naar specifieke gebruikers en naar segmenten van de gebruikers van de applicatie. Deze segmenten zijn vooraf gedefinieerd en opgeslagen in de onderliggende database.

Push notificaties naar offline gebruikers

Push notificaties moeten ook ontvangen worden als de gebruiker geen internet verbinding heeft op het moment dat de notificatie verstuurd wordt. Wanneer de gebruiker weer online komt moet de notificatie worden ontvangen.

Should

Push notificaties met standaard acties

Het aanklikken van een notificatie moet applicatie code kunnen laten uitvoeren. Bijvoorbeeld: het automatisch uitloggen van gebruikers of het verversen van het applicatie-model.

Push notificaties met deeplinks

Het klikken van een push notificatie moet een gebruiker naar de relevante plaats in de applicatie kunnen sturen en deze schermen openen als ze dat nog niet zijn.

Push notificaties met een vervaltermijn

Push notificaties zijn op een gegeven moment niet meer relevant voor de eindgebruiker, in dat geval moet er een verval termijn bestaan voor de notificatie. Wanneer dit termijn voorbij is moet de notificatie niet meer verstuurd worden naar deze gebruiker. Dit is relevant wanneer de gebruiker langere tijd de applicatie niet heeft geopend.

Could

Interactieve push notificaties

Push notificaties moeten input toestaan, bijvoorbeeld de mogelijkheid om een inlogpoging goed te keuren voor twee factor authenticatie.

NON-FUNCTIONELE REQUIREMENTS

Must

Geschikt voor alle gebruikersinterfaces

De push notificaties moeten werken voor alle gebruikersinterfaces waar op dit moment voor ontwikkeld wordt bij Thinkwise. Dit zijn:

- (Windows) Desktop applicatie – via een Progressive Web App
- Web applicatie – via een Progressive Web App
- Android App – via Cordova
- iOS App – via Cordova

Secure push notificaties

De push notificaties moeten veilig verstuurd worden. Dit houdt in dat de push berichten versleuteld verstuurd worden.

Won't

Op dit moment worden ook een .NET Windows en een ASP.NET Web applicatie gebruikt. Ondersteuning voor deze platformen heeft echter geen prioriteit om dat deze in de toekomst uit gefaseerd zullen worden voor de PWA.

METHODIEK

Dit hoofdstuk beschrijft hoe er te werk is gegaan en hoe aan de requirements zal worden voldaan.

TIJDLIJN

Ten eerste zijn er vanuit Thinkwise trainingen gevolgd over het gebruik van het Thinkwise Platform. Dit diende om geïntroduceerd te worden aan het product en een goed beeld te vormen over de activiteiten van Thinkwise.

Vervolgens is er een plan van aanpak opgesteld om de activiteiten van de stage in kaart te brengen. Hierbij zijn de verwachtingen van alle betrokken partijen afgestemd door een functioneel ontwerp te maken.

Nadat alle partijen akkoord waren, zijn er een aantal onderzoeken uitgevoerd. Als eerste is er onderzoek uitgevoerd om een aantal kernbegrippen in kaart te brengen. Dit waren “Model Gedreven Ontwikkeling”, “Push Notificaties”, “WebSockets”, Cordova, en Progressive Web Apps. Deze begrippen zijn onderzocht om te bepalen of er complicaties zouden ontstaan bij de uiteindelijke implementatie. Er is vervolgens een onderzoek gedaan om ook het product van Thinkwise in kaart te brengen om dezelfde reden. Ten slotte is onderzocht welke technologie het meest geschikt was voor de implementatie van push notificaties, daarbij is ook gekeken naar de prijs van elke optie.

Ten slotte is er gewerkt om deze technologie te gebruiken om prototypes te bouwen die aan alle voorgenoemde requirements voldoen.



SCRUM

Scrum is een framework voor het samenwerken in teams aan complexe producten. Scrum is een half jaar geleden opgepakt bij Thinkwise. Vandaar is er tijdens het onderzoek en het bouwen van prototypes gewerkt met onderdelen van Scrum. Omdat de opdracht niet in een team is uitgevoerd is Scrum niet in het geheel gebruikt.

Er is een backlog bijgehouden van taken waaraan gewerkt is, en er is elke twee weken een review geweest om te bepalen wat er gedaan moest worden en om de opdrachtgever op de hoogte te houden van het werk dat verzet was.

ONDERZOEK

Dit hoofdstuk beschrijft het onderzoek en de daaruit volgende resultaten, maar geeft eerst uitleg over een aantal kernbegrippen.

WAT IS MODEL GEDREVEN ONTWIKKELING?

Model gedreven softwareontwikkeling gebruikt voorgebouwde componenten en een grafische interface om een model om te zetten in een applicatie, om software ontwikkeling makkelijker en sneller te maken. Zowel in het updaten, uitbreiden en nieuw bouwen van software. Het werken met model gedreven ontwikkelen zorgt er dus voor dat kleinere teams sneller grote applicaties kunnen realiseren.

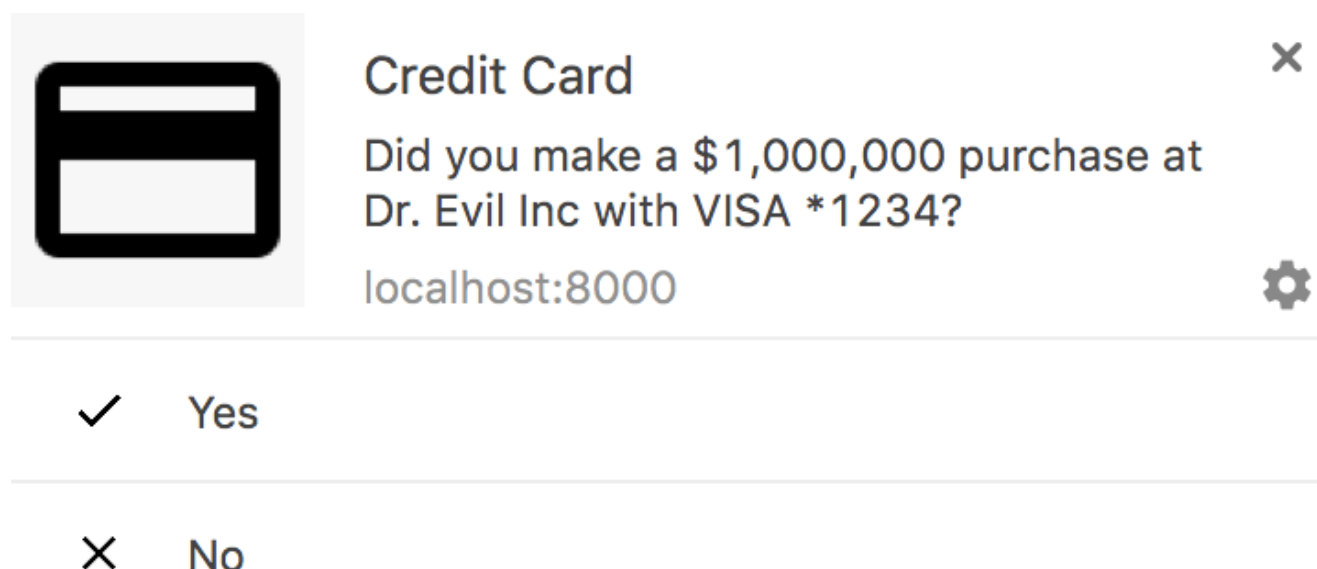
Ten eerste wordt er een applicatie gemodelleerd. Dit model bevat alle informatie over een applicatie. Het model wordt vervolgens geconverteerd naar een werkende applicatie door het model te interpreteren. Deze transformatie gebeurt wanneer de applicatie gebruikt wordt op run-time. Op deze manier is er geen behoefte code te genereren of te schrijven.

Model gedreven ontwikkelmethodes zijn vooral geschikt in situaties waarin gelijksoortige functionaliteit vereist is in meerdere applicaties.

(Klaren, z.d.)

WAT ZIJN PUSH NOTIFICATIES?

Notificaties zijn berichten in een applicatie die een pop-up maken op het device van een gebruiker. Wanneer een notificatie niet vanuit een applicatie zelf komt maar “gepusht” wordt naar een applicatie is het een push notificatie.



Figuur 1 Voorbeeld push notificatie (Medley, 2019)

Push berichten stellen een ontwikkelaar in staat gebruikers aan te spreken zonder dat een applicatie actief is. Normaal gesproken is het niet mogelijk om een applicatie aan te spreken wanneer deze niet op de computer gestart is. Elk platform en browser heeft een eigen push service die notificaties voor dat systeem afhandelt. De app abonneert zich op de push service en geeft berichten door aan de juiste applicatie. Als een applicatie niet actief is, kan deze service de applicatie opstarten om de notificatie af te handelen.

Door een push bericht te sturen naar een applicatie, kan de applicatie de inhoud van dit bericht gebruiken om een notificatie aan te maken en weer te geven.

Om een push notificatie te sturen naar een applicatie die op dat moment niet actief is, is het nodig om een service te gebruiken die buiten de applicatie aan staat.

Voor Android applicaties was dit voorheen Google Cloud Messaging, nu is dat Firebase Cloud Messaging.

Bij iOS en macOS applicaties moet gebruik gemaakt worden van de Apple Push Notification Service (APNS). De APNS wordt ook gebruikt om push notificaties te sturen naar macOS desktop applicaties.

Windows heeft ook een eigen push notificatie service voor het leveren van push notificaties naar Universal Windows Platform applicaties.

Elke browser die de Push API ondersteunt heeft ook een eigen service. (Google, 2019b)

Om het proces van het versturen van notificaties te versimpelen kan het wenselijk zijn een framework of service te gebruiken van een derde partij, die deze services tegelijkertijd aan kan spreken.

WAT ZIJN WEBSOCKETS?

WebSockets is een communicatieprotocol dat een client applicatie in staat stelt om een interactieve verbinding te maken met een server zonder dat de server gepolld moet worden voor een antwoord. Deze communicatie gebeurt via events. Een socket connectie wordt aan event listeners gekoppeld die de verbinding afwachten. Wanneer een bericht wordt ontvangen wordt de bijbehorende event listener aangeroepen om dat bericht af te handelen.

WebSockets kunnen gebruikt worden wanneer real-time communicatie met een server van belang is, bijvoorbeeld in het geval van een applicatie die de prijs van aandelen bijhoudt, een multiplayer web game, of een chat app. ("What are WebSockets? | Pusher", z.d.)

WAT IS HET THINKWISE PLATFORM?

Het Thinkwise Platform is het core product van Thinkwise. Het wordt gebruikt in elke fase van de ontwikkeling en deployment van een applicatie. Het Thinkwise Platform is low-code, dat betekent dat er weinig of geen programmeerwerk nodig is bij het bouwen van een applicatie. Het Thinkwise Platform bestaat uit vier belangrijke onderdelen met elk een eigen verantwoordelijkheid. Deze onderdelen zijn als volgt:

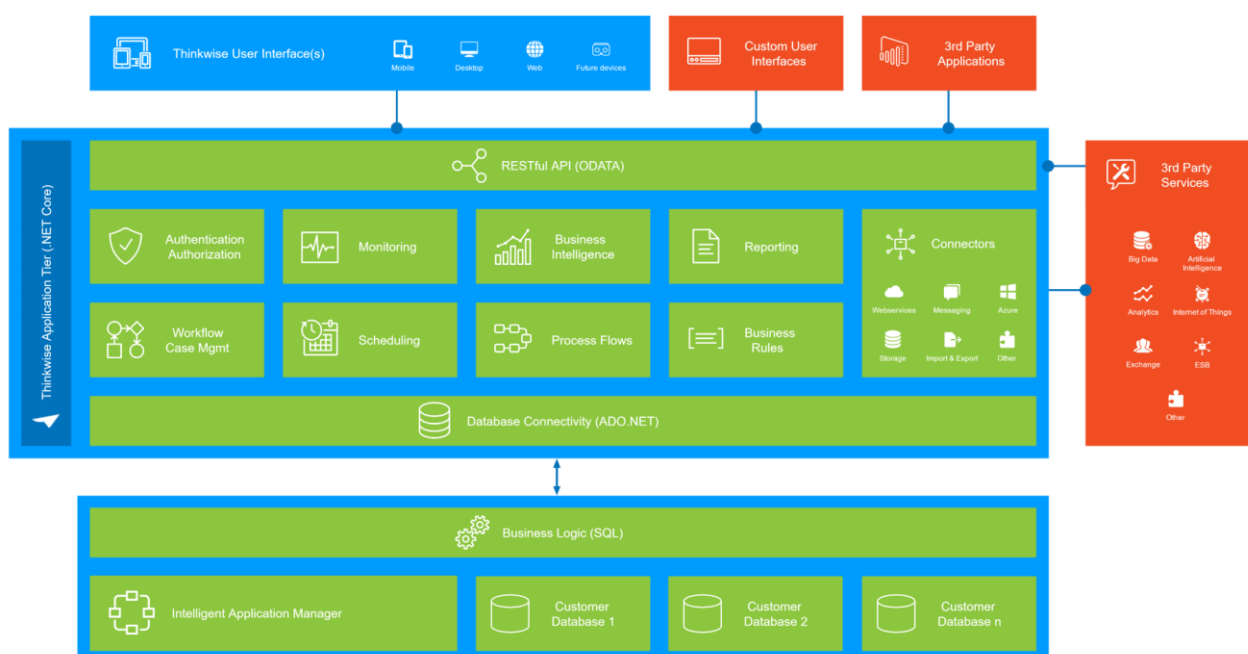
Thinkwise Software Factory

De Thinkwise Software Factory wordt gebruikt voor het gehele ontwikkelproces van een applicatie. De ontwikkeling in de Software Factory is model gedreven, dit betekent dat er een model ontwikkeld wordt dat ten grondslag ligt aan de applicatie. Dit model beschrijft alle informatie, functionaliteit en business logica van een applicatie. De Software Factory genereert uiteindelijk een script dat het model aan kan maken op de productie-server.

De Software Factory beheert de volgende onderdelen van het ontwikkelproces:

- Projectbeheer/Versiebeheer
- Design Specificaties
- Datamodel Ontwerp
- User Interface Ontwerp
- Proces Logica
- Business Logica
- Kwaliteitscontrole & Testen
- Deployment

Intelligent Application Manager



Figuur 2. Thinkwise Platform Runtime Architectuur

De Intelligent Application Manager (IAM) wordt in productie gebruikt om de applicaties te managen. Een enkele instantie van IAM kan meerdere applicaties beheren. De IAM is onder andere verantwoordelijk voor het bewaren van gebruikersgegevens, rollen in de applicaties en de rechten die daarbij horen.

De IAM bevat op productie ook het model dat door de Software Factory is gegenereerd. IAM geeft een subset van dit model op basis van de rechten die een gebruiker heeft.

Indicium Application Tier

Indicium is een generieke application-tier in een multitierarchitectuur. Dat betekent dat de structuur van de applicatie bestaat uit een aantal aparte lagen die ieder op een apart systeem kunnen draaien.

Indicium interpreteert het model dat in de Software Factory is gemaakt. Het is mogelijk om Indicium voor meerdere applicaties tegelijk te gebruiken, onder andere de IAM.

Indicium interpreteert het model dat het ontvangt uit de IAM en zet een API op om de data in het eindproduct te kunnen manipuleren

Indicium is verantwoordelijk voor het authentifieren en autoriseren van gebruikers. Indicium kan ook koppelen met services van derde partijen, bijvoorbeeld:

- Artificial Intelligence services (Bots, Virtual Assistants, Machine Learning, Image recognition)
- Big Data solutions (Search, Analytics, Internet of Things)
- Office integration (Exchange, Office 365, SharePoint)
- Third party applications (custom user interfaces, financial applications)

Universal Graphical User Interface

De userinterface interpreteert het model van de IAM dat geleverd wordt door Indicium en levert de bijbehorende graphical userinterface. Er is een userinterface beschikbaar voor Android, iOS en desktop voor Windows, Mac en Linux, daarnaast is er ook een web interface beschikbaar.

Mobiele applicaties worden gebouwd met Cordova, een mobiel development framework dat ontwikkelaars in staat stelt om hybride applicaties te bouwen voor Android en iOS. Cordova maakt ook gebruik van HTML, CSS en Javascript om applicaties te bouwen. Een Cordova applicatie gebruikt de Cordova Wrapper voor het platform om functionaliteiten aan te roepen van de telefoon. Dit zorgt ervoor dat er maar één codebase nodig is voor zowel een Android als een iOS app. Ten slotte wil Thinkwise in de toekomst een Progressive Web App in gaan zetten.

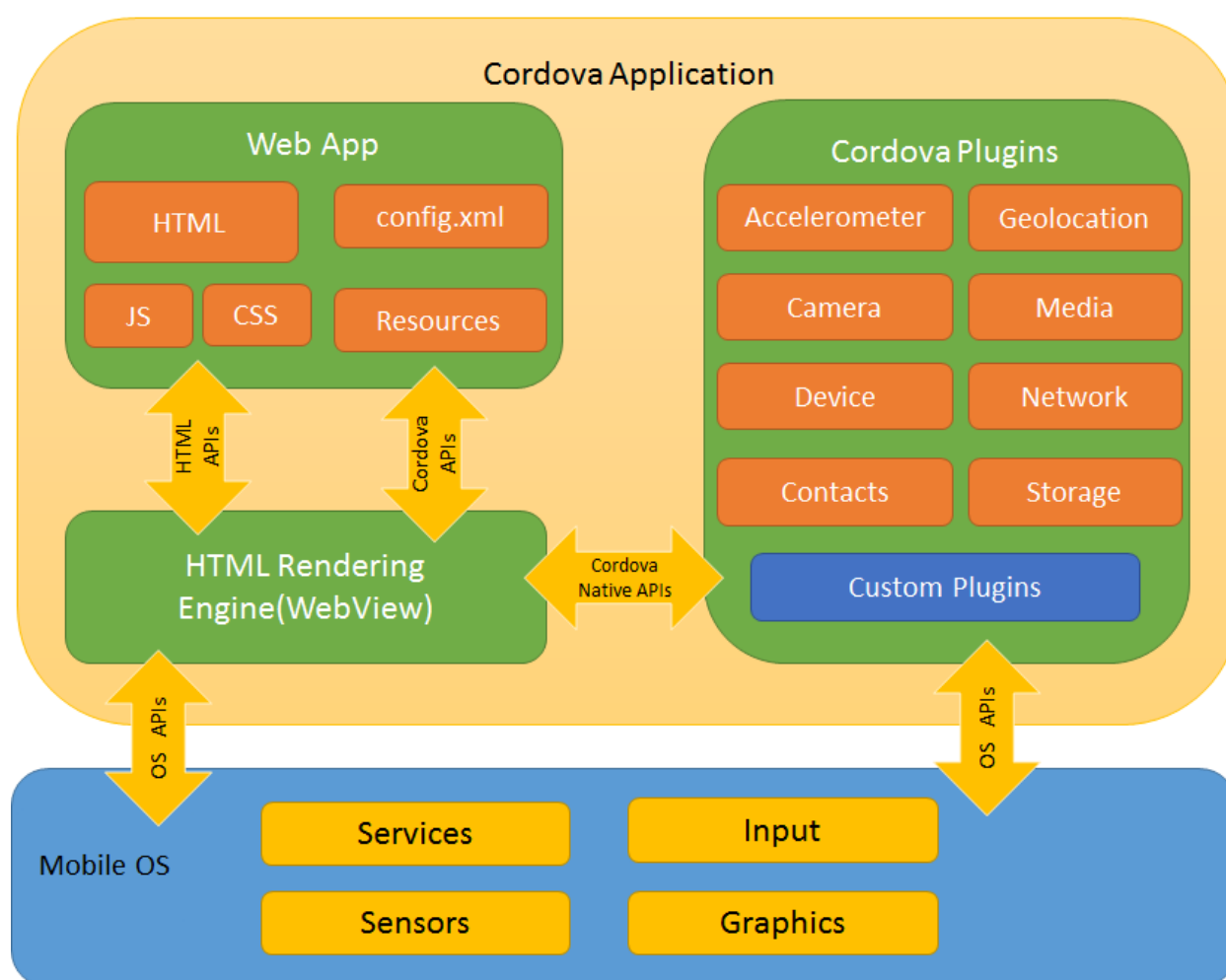
Conclusie

Voor deze opdracht zijn vooral Indicium en de User Interface van belang. Het versturen van Push Notificaties moet gebeuren vanuit Indicium, door een service buiten Indicium aan te roepen. Het ontvangen en weergeven van push notificaties moet gebeuren vanuit de user interface. In een uiteindelijke integratie moet ook de Software Factory in beschouwing genomen worden, zodat notificaties meegenomen kunnen worden in de Proces- en Business Logica.

Wat is Cordova?

Apache Cordova is een open-source mobile development framework. Cordova wordt op mobiele devices gedraaid in een wrapper voor dat platform. Om gebruik te maken van Native functionaliteit worden Cordova plug-ins gebruikt. Voor Cordova wordt met HTML, CSS en Javascript geprogrammeerd.

Cordova Apps worden weergegeven in een webview met toegang tot de Cordova Api's.



Figuur 3 Cordova Architectuur

("React – A JavaScript library for building user interfaces", z.d.)

Wat zijn Progressive Web Apps?

Progressive Web Apps zijn websites die geïnstalleerd kunnen worden op pc's of op het home scherm van een mobiele telefoon, PWA's hebben echter geen tabbladen of adresbalken. Daardoor zijn ze in gebruik hetzelfde als native applicaties. PWA's werken zowel online als offline. Met behulp van service workers wordt de inhoud van webpagina's lokaal opgeslagen en worden verzoeken naar de server bewaart tot er een internet verbinding beschikbaar is. (Google, 2019)

ANALYSE

De vraag is nu: **Welke technologie of service voldoet het beste aan de gestelde requirements?**

Uit exploratief onderzoek zijn de volgende kandidaten voor push notificatie frameworks naar voren gekomen.

- SignalR (WebSockets)
- Google Firebase
- Microsoft App Center Push
- Amazon Simple Notification Service
- OneSignal
- Urban Airship.

Van ieder van deze frameworks is onderzocht of deze aan de voorgenoemde requirements kunnen voldoen. Ook is er een prijsvergelijking gemaakt om te bepalen wat voor kosten verbonden zijn aan het gebruik van deze services.

Uiteindelijk is Firebase gekozen, voor de volledigheid zijn de resultaten van elk platform opgenomen in dit verslag.

SignalR

SignalR is een library in het .NET Core framework dat real-time communicatie kan verzorgen tussen client en server (Microsoft, 2018). Het werkt op basis van het Websockets communicatieprotocol. Aangezien Indicium gebouwd is met het .NET Core Framework lijkt dit in eerste instantie een perfecte kandidaat voor push notificaties.

Het is echter niet gebouwd met notificaties in gedachte, dat betekent dat er geen ingebouwde functionaliteit is om communicatie te bewaren tot een gebruiker online komt. Ook is er geen mogelijkheid te communiceren met een applicatie die niet actief is.

Deze functionaliteit zou gebouwd moeten worden bovenop het SignalR framework, Dat betekent dat er geen gebruik gemaakt zou worden van de bestaande services die push berichten bezorgen.

Uiteindelijk is SignalR om deze redenen geen geschikte kandidaat om push notificaties te implementeren.

Google Firebase

Google Firebase is een cloud service platform van Google. Het biedt een groot aantal functionaliteiten aan, waaronder de mogelijkheid tot push notificaties (Firebase, z.d.-a). Firebase Cloud Messaging is onderdeel van dit platform.

Firebase Cloud Messaging (FCM) biedt de mogelijkheid push notificaties te versturen naar Android devices (Firebase, 2019f). Ook integreert Firebase de APNS in zijn eigen services, dit betekent dat Firebase ook voor iOS devices gebruikt kan worden (Firebase, 2019c). FCM kan ook gebruikt worden om push notificaties te versturen naar browsers (Firebase, 2019b).

Firebase biedt standaard geen ondersteuning voor Cordova. Maar er zijn modules beschikbaar van derde partijen (Chemerisuk, 2019). Firebase biedt standaard geen mogelijkheid aan om Windows notificaties te sturen.

Firebase stelt de ontwikkelaar in staat device groepen te maken die door een server beheerd kunnen worden. Het is dus mogelijk om notificaties te sturen naar verschillende gebruikersgroepen van een applicatie. Ook is het mogelijk gebruikers aan toesturen te koppelen om notificaties te versturen naar specifieke gebruikers door deze device id's op te slaan in een database op een eigen server (Firebase, 2019d) (Firebase, 2019e) .

Firebase Cloud Messaging maakt gebruik van een time-to-live property voor berichten. Dit betekent dat een bericht niet meer afgeleverd zal worden wanneer het bericht verlopen is. Wanneer deze time-to-live op nul gezet wordt, wordt het bericht alleen naar alle online gebruikers gestuurd. Wanneer de time-to-live groter is dan nul wordt het bericht bewaard tot het afgeleverd kan worden. (Firebase, 2019a).

FCM heeft ook de mogelijkheid om Topics aan te maken. Dit zijn endpoints op de Firebase servers waar clients zich aan kunnen abonneren. Wanneer een bericht verstuurd wordt, ontvangen alle abonnees een notificatie. Het abonneren op een Topic kan vanuit een client applicatie en vereist geen speciale rechten, daarom is het alleen geschikt om publieke informatie te versturen (Firebase, 2019h). Abonnementen kunnen ook aangemaakt worden via de API van Firebase (Firebase, 2019g).

Firebase is op dit moment een API aan het beta testen om het managen van projecten programmatisch te kunnen verzorgen. Firebase projecten zijn onder de kap Google Cloud Projecten. Deze kunnen aangemaakt worden via de GCP Resource Manager API. Vervolgens kan de Firebase Management API gebruikt worden om Firebase toe te voegen aan dit project. Vervolgens kan de API gebruikt worden om apps toe te voegen (Firebase, 2019i).

In conclusie, Firebase heeft één tekortkoming omdat het geen Universal Windows Platform Notificaties ondersteunt. Electron heeft ook geen standaard ondersteuning voor FCM. In een Progressive Web App zou deze tekortkoming verdwijnen, omdat er dan een service worker gebruikt kan worden om in de achtergrond te luisteren naar push notificaties.

Microsoft App Center Push

App Center Push is een service die Firebase Cloud Messaging, Apple Notification Service en Windows Notification Service (WNS) omvat (Microsoft, 2019b). Het is onderdeel van het Microsoft App Center platform, een suite services voor het bouwen, het testen, de distributie, en het monitoren van een applicatie (Microsoft, z.d.).

Om een notificatie te versturen met WNS moet de applicatie geregistreerd zijn in de Windows Store. Deze notificaties werken echter alleen voor Windows 10 en 8.1. Ook werken ze alleen voor Universal Windows Platform applicaties die geregistreerd zijn in de Windows Store. Dit is een limiet dat uit de Windows Notification Service komt, het is alleen mogelijk Windows desktop notifications te versturen naar applicaties die aan de voorgenoemde eisen voldoen (Microsoft, 2017).

App Center Push biedt de mogelijkheid om gebruikers te verdelen in segmenten genaamd Audiences. Een applicatie kan maximaal 200 Audiences hebben. Daarnaast is het mogelijk om notificaties te versturen naar maximaal 20 devices aan de hand van device Id's (Microsoft, 2019b).

Een Audience kan maximaal 1,000 devices bevatten. Als een Audience meer devices omvangt zullen de berichten naar de eerste 1,000 devices worden gestuurd die met het Audience matchen, de rest worden genegeerd (Microsoft, 2019c).

Door device Id's op te slaan in een eigen database en te associëren met een gebruiker is het mogelijk om berichten te sturen naar specifieke gebruikers.

Microsoft App Center Push biedt geen web integratie aan. Dit zou apart geïmplementeerd moeten worden als het ondersteund moet worden, bijvoorbeeld met Firebase.

Ten slotte is het mogelijk om custom data te versturen in de vorm van key/value pairs (Microsoft, 2019d).

Amazon Simple Notification Service

Amazon SNS is een service voor het versturen van push notificaties. Het kan verschillende push frameworks gebruiken waaronder FCM, APNS en WNS (Amazon, z.d.-a).

Amazon SNS is om Topics heen gebouwd. Gebruikers kiezen ervoor om zich aan een Topic te abonneren, vervolgens zullen ze de notificaties die bij dat Topic horen ontvangen. Als eigenaar van een applicatie in Amazon SNS kun je de toegang tot een Topic afschermen.

Wanneer een bericht gepubliceerd wordt naar een Topic wordt deze verstuurd naar alle abonnees (Amazon, z.d.-b).

OneSignal

OneSignal is een service voor customer engagement, het biedt push notificaties, e-mail, en in-app messages aan (OneSignal, 2019).

Het ondersteunt web push notificaties en heeft een Cordova plug-in (OneSignal, z.d.-a), (OneSignal, z.d.-b).

Onder de kap moet OneSignal APNS en Firebase gebruiken om notificaties te versturen naar Android en iOS devices. OneSignal mist echter de functionaliteit om notificaties te sturen naar macOS. (OneSignal, z.d.-c).

OneSignal ondersteunt het verdelen van gebruikers in segmenten aan de hand van custom user tags. Deze tags kunnen als voorwaarde gebruikt worden voor deelname aan een segment (OneSignal, z.d.-d).

OneSignal ondersteunt ook het verzenden van custom data payloads (OneSignal, z.d.-e).

OneSignal mist alleen een API call om programmatisch segmenten aan te maken in een applicatie (OneSignal, z.d.-e).

Urban Airship

Urban Airship is een service voor customer engagement, het biedt onder andere push notificaties aan, dit is ook het hoofdproduct (Urban Airship, z.d.-a).

Urban Airship ondersteunt zowel web push notificaties als Cordova integratie (Urban Airship, z.d.-b), (Urban Airship, z.d.-c).

Urban Airship biedt de onderliggende time-to-live en expiry attributen van Firebase en APNS aan in de API (Urban Airship, z.d.-d). Dit betekent dat Urban Airship van deze functionaliteit gebruik kan maken om berichten alleen naar actieve gebruikers te sturen, of naar gebruikers die op dit moment niet online zijn.

Ten slotte kan Urban Airship zowel specifieke gebruikers push notificaties sturen als lijsten devices (Urban Airship, z.d.-e), (Urban Airship, z.d.-f).

Prijsvergelijking

Omdat push notificaties geleverd zullen worden als integraal onderdeel van het Thinkwise Platform, gaat de voorkeur uit naar een oplossing waarbij de klanten van Thinkwise niet extra hoeven te betalen voor deze functionaliteit.

APNS

De APNS levert geen extra kosten op voor een ontwikkelaar. Deze is gratis te gebruiken met een Apple developer account (Apple, z.d.).

Amazon SNS

Amazon SNS is onderdeel van Amazon Web Services (AWS). De eerste miljoen requests zijn gratis, daarna kosten elke miljoen requests \$0,50 (Amazon, z.d.-c).

Firebase Cloud Messaging

Cloud Messaging is een gratis onderdeel van Firebase. Als er echter geen betaalde versie van Firebase gebruikt wordt zal Google Analytics verzamelen van je service. Firebase heeft een betaald plan dat het “Blaze” plan heet, hierbij wordt alleen betaald wanneer het gebruikslimiet van het gratis “Spark” plan wordt overschreden (Firebase, z.d.-b).

In het “Blaze” plan is Cloud Messaging nog steeds gratis. Het gebruik van Firebase zou dus geen extra kosten opleveren.

App Center Push

App center Push is onderdeel van App Center, een ontwikkel suite voor mobiele applicaties. Het biedt gratis 5 audience segmenten aan, meer segmenten zal \$10,-- per maand kosten per 100,000 actieve devices. Het absolute maximum aantal Audiences is 200 (Microsoft, 2019c).

OneSignal

OneSignal biedt een gratis plan aan dat een oneindig aantal push notificaties bevat en 30.000 subscribers (OneSignal, z.d.-f). Bij het gratis plan wordt data echter wel gedeeld met adverteerders, en onderzoeksbedrijven (OneSignal, z.d.-g).

Het plan dat waarschijnlijk interessant is voor Thinkwise kost \$500,- dollar per maand (OneSignal, z.d.-f).

SignalR

Er zijn geen extra kosten voor het gebruik van SignalR.

Urban Airship

Urban Airship rekent per gebruiker een prijs. De prijzen zijn als volgt (Urban Airship, z.d.-g):

Addressable Users	Basic	API
0-10,000 gebruikers	\$99	\$49
10,001-25,000 gebruikers	\$249	\$149
25,001-50,000 gebruikers	\$399	\$249

Er zijn grotere plannen beschikbaar, maar daarvoor moet contact opgenomen worden met Urban Airship.

Conclusie & Aanbevelingen

Requirements:

- Functioneel:
 1. Real-time push notificaties
 2. Push notificaties naar specifieke gebruikers en groepen
 3. Push notificaties naar enkel actieve gebruikers (of andere filters)
 4. Push notificaties naar offline gebruikers
 5. Push notificaties met standaard acties
 6. Push notificaties met deeplinks
 7. Push notificaties met een vervaltermijn
 8. Interactieve push notificaties
- Non Functioneel:
 1. Geschikt voor alle devices
 2. Secure push notificaties
 3. Prijs

	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	NFR1	NFR2
SignalR	✓	*	*	X	*	*	*	*	✓	✓
Firebase	✓	✓	✓	✓	*	✓	✓	*	✓	✓
App Center	✓	✓	✓	✓	*	*	✓	*	X	✓
Amazon SNS	✓	✓	✓	✓	*	*	✓	*	✓	✓
OneSignal	✓	✓	✓	✓	*	*	✓	*	X	✓
Urban Airship	✓	✓	✓	✓	*	*	✓	*	✓	✓

*: Er is hier geen functionaliteit voor ingebouwd, maar het framework ondersteunt deze uitbreiding.

Advies

Als advies wordt meegegeven om Firebase te kiezen. Firebase heeft alle benodigde onderdelen. Daarnaast heeft het de grootste community. Firebase wordt ook gebruikt door zowel App center als Amazon SNS.

App Center Push heeft een limiet op Audiences dat duur is om uit te breiden. Daarnaast kan het limiet van 1000 devices in een audience een zeer groot probleem vormen wanneer dat limiet bereikt wordt, aangezien er geen mogelijkheid is dit te vergroten.

Zowel Amazon SNS, OneSignal, en Urban Airship bieden geen extra relevante functionaliteit en kosten meer dan Firebase.

Firebase is nodig bij alle alternatieven om push notificaties te sturen naar Android telefoons. Daarom is het advies om Firebase direct te gebruiken.

Conclusie

In eerste instantie was er een voorkeur voor Microsoft App Center Push, omdat Thinkwise meer onderdelen van deze service gebruikt. Dit was echter niet mogelijk omdat App Center nog geen web apps ondersteunt.

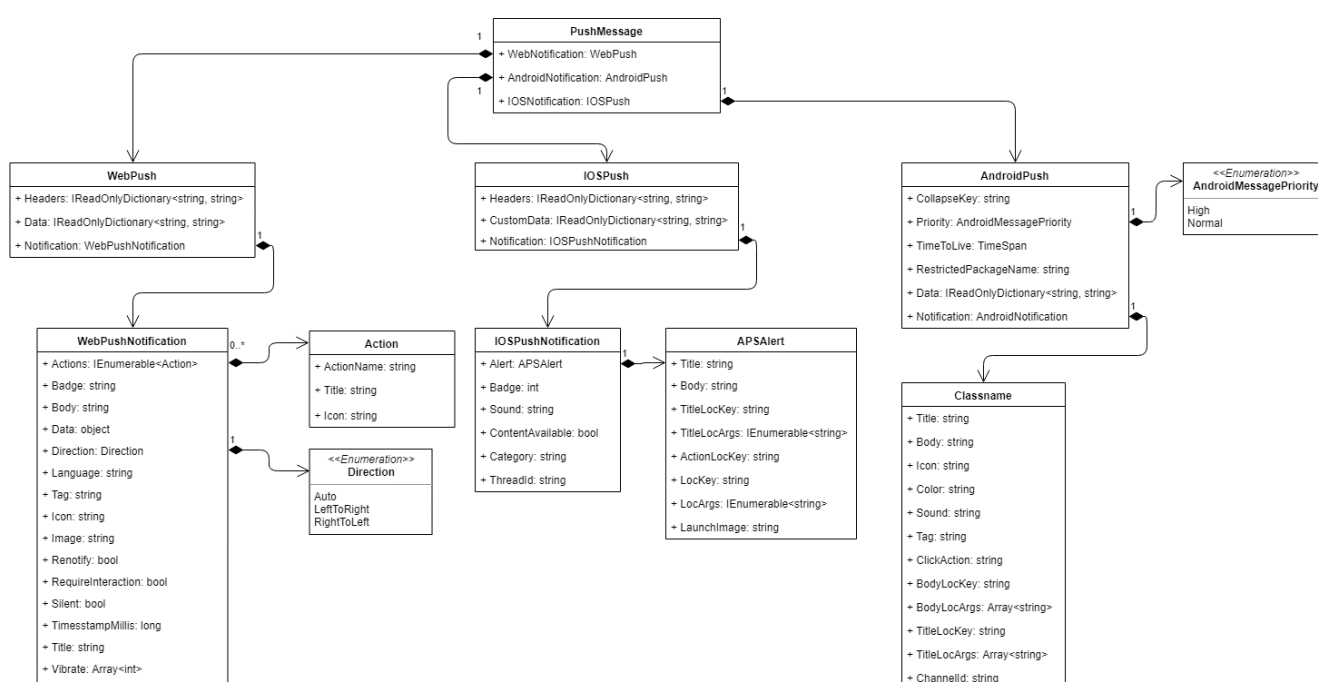
Uiteindelijk is het advies opgevolgd en is er voor Firebase gekozen.



ONTWERP

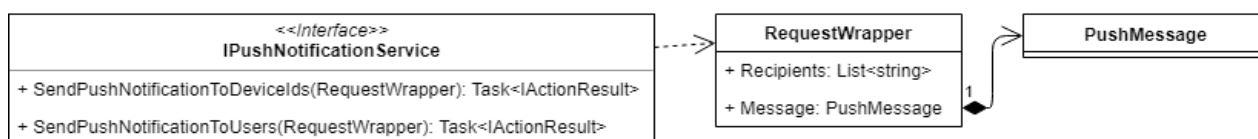
BACK-END

De back-end voor notificaties die bij Indicium geïmplementeerd kan worden moet een interface bieden dat, onafhankelijk van de onderliggende technologie, push berichten kan ontvangen. Dit generieke bericht zal vervolgens omgezet worden in een vorm die door het onderliggende framework begrepen kan worden. Deze implementatie is gebouwd in C#.



Figuur 4 Anatomie van een push bericht

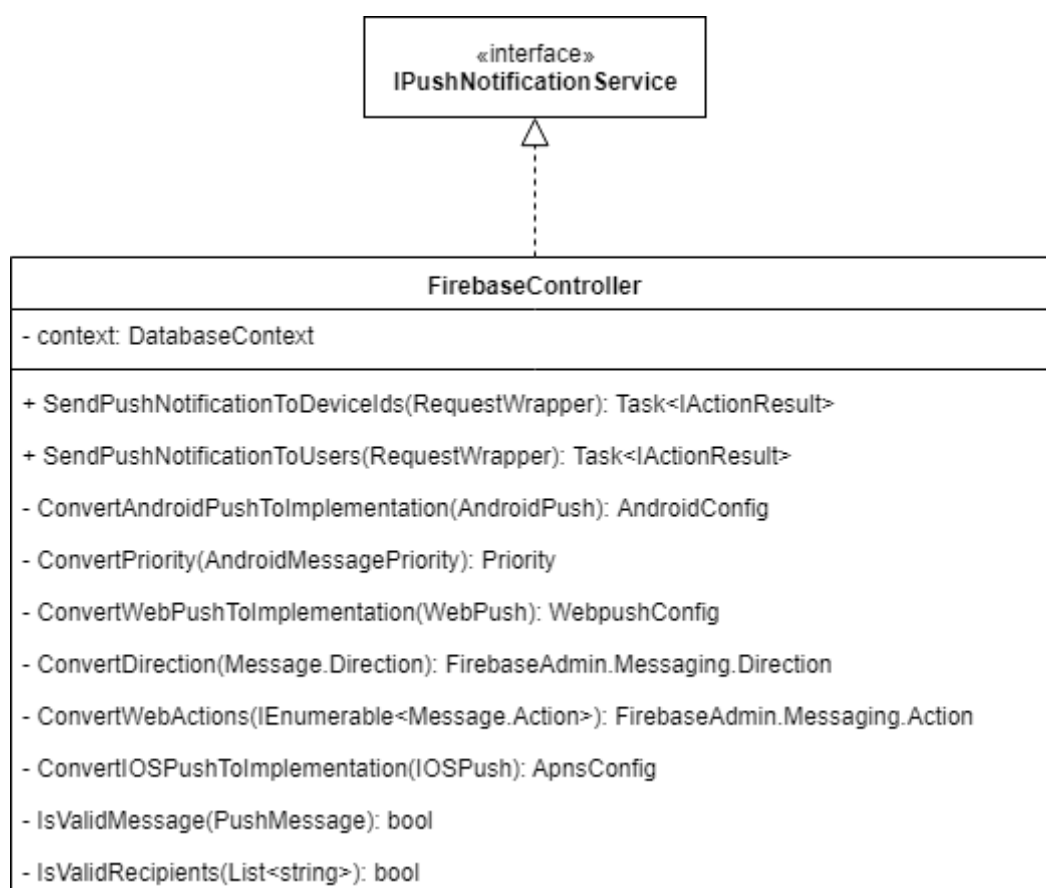
De vorm van dit model is gekozen om zoveel mogelijk te lijken op de notificaties van elk platform (Web, Apple, en Android). Dat betekent dat dit model gebruikt kan worden om notificaties te representeren in elke mogelijk implementatie, zonder dat het functionaliteit verliest, omdat alle mogelijke opties van deze notificaties aanwezig zijn.



Figuur 5 Push service architectuur

Hierboven wordt de generieke interface beschreven. Het bestaat uit een interface dat twee methodes bevat die gebonden kunnen worden aan een http request. De “Task<ActionResult>” is de standaard voor .NET Core methodes die een status code terug geven. Deze twee methodes worden in de implementatie gebonden aan een url.

Het is in .NET Core alleen mogelijk om één argument uit de request-body te halen. Daarom wordt de RequestWrapper gebruikt om zowel de ontvangers van, en het notificatie bericht zelf te kunnen ontvangen.



Figuur 6 Controller architectuur

Hierboven wordt de implementatie van de push notificatie service beschreven. Deze biedt een consistente interface aan met behulp van het voorgenoemde model. Het binnengekomen bericht wordt vertaald met behulp van de “convert” methodes, en wordt vervolgens doorgegeven aan de Firebase SDK. Alle responses worden gebundeld in één en worden vervolgens teruggestuurd naar de gebruiker. Dit is nodig, omdat Firebase slechts honderd ontvangers per request accepteert.

Zo bevat deze implementatie alle functionaliteit die aangegeven is door de requirements.

Deze API kan worden geïntegreerd in het Thinkwise Platform of uitgebreid worden om als losstaande service te functioneren en op deze manier te verbinden met Indicum.

FRONT-END

Het Thinkwise platform gebruikt verscheidene technologieën om gebouwde applicaties beschikbaar te maken op verschillende platformen. Voor het maken van de user interface wordt React.js gebruikt, om de functies van een telefoon te benaderen wordt het Cordova framework gebruikt. Voor desktop applicaties worden de Windows- en webapplicatie gebruikt.

Thinkwise wil graag overstappen naar PWA's. Vandaar dat er geen implementatie worden gebouwd van push notificaties voor de native Windows applicatie.

Ontwerp

Cordova

Cordova kan native functionaliteit aanspreken met behulp van plug-ins. In dit geval zijn er twee plug-ins nodig om interactieve push notificaties tot stand te kunnen brengen.

Cordova-plugin-fcm

De push service die berichten ontvangt verschilt tussen Android en iOS, daarom is er plug-in nodig die de juiste service gebruikt afhankelijk van het platform.

Deze plug-in is nodig om notificatie berichten te kunnen ontvangen en deze door te geven aan de Cordova Applicatie. Ook geeft deze plug-in aan of de notificatie aangeklikt is, dit onderdeel wordt echter niet gebruikt, want standaard notificaties bevatten geen mogelijkheid om interactie toe te voegen in de vorm van buttons of invoervelden.

Cordova-plugin-local-notification

Android push notificaties bevatten geen logica om interactiviteit aan te geven. Dit betekent dat push notificaties geen buttons of input kunnen bevatten. Het is echter wel mogelijk om data berichten te sturen in de achtergrond van een applicatie. Dit is waar de local notification plugin belangrijk is. Door een data bericht te sturen i.p.v. een notificatie bericht kan de applicatie een lokale notificatie bouwen met de data uit dit bericht. Deze lokale notificaties kunnen wel interactief zijn, in tegenstelling tot de standaard push notificatie.

Door deze twee plug-ins te combineren zijn interactieve push notificaties voor Cordova mogelijk gemaakt.

Progressive Web App

Een PWA moet op twee verschillende plaatsen in de applicatie push notificaties kunnen ontvangen. Wanneer de applicatie in de voorgrond actief is zal de notificatie direct bij het proces van de applicatie aankomen, er moet in dit proces dus een methode gemaakt worden die het push bericht omzet in een notificatie.

Daarnaast moet Dezelfde functionaliteit worden geïmplementeerd in de service worker die de website registreert. Deze functionaliteit wordt aangeroepen wanneer de applicatie zich niet in de voorgrond bevindt.

CONCLUSIE

Er zijn uiteindelijk drie software producten opgeleverd, een Web API, een Cordova Applicatie en een Progressive Web App. Het resultaat is een set applicaties die web notificaties kunnen ontvangen en versturen, en die klaar staan om interactieve notificaties aan te bieden.

Push notificaties op de mobiele telefoon zijn al langer in gebruik en als resultaat zijn de oplossingen voor deze implementaties robuuster en beter ondersteund. Dit verschil was vooral te merken bij de implementatie van de Progressive Web App. Niet alle browsers ondersteunen namelijk alle functionaliteit van notificaties. Opera en Chrome ondersteunen bijvoorbeeld een maximum van twee knoppen in een notificatie, Firefox ondersteunt dit totaal niet. Service workers en web notificaties zijn nog nieuwe technologie, en dat viel zeer te merken.

AANBEVELINGEN

Voor de toekomst van push notificaties bij Thinkwise zijn er een aantal aanbevelingen.

Micro services

Tijdens het project is er een aantal keer naar voren gekomen dat er een zorg is over de performance van Indicium in het geval van het inplannen van notificaties. Indicium bespaart namelijk resources door alleen te activeren wanneer deze aangeroepen wordt.

Hierbij zou er gekeken kunnen worden naar het opsplitsen van Indicium in micro-services. Door Indicium op te splitsen zou er slechts een deel hoeven te draaien om geplande notificaties te versturen.

Het zou zelfs mogelijk zijn om de performance te verbeteren doordat Indicium niet meer in zijn geheel opgestart hoeft te worden en verdeeld zou kunnen worden over verschillende machines.

Als deze verandering te intensief zou zijn is het ook mogelijk de push notificatie in een aparte service te laten draaien.

Progressive Web App

Tijdens de stage is er gesproken over het overstappen naar Progressive Web Apps en daarbij ook Cordova applicaties uit te faseren. Dit is op dit moment nog geen goed idee wat de functionaliteit van notificaties betreft.

Web notificaties hebben zeer beperkte mogelijkheden, die niet door alle browsers ondersteund wordt. Deze technologie is echter constant in ontwikkeling, dus is het belangrijk de voortgang goed bij te houden. In het geval deze functionaliteit ooit gelijk wordt aan de mogelijkheden op mobiele devices, is het tijd om volledig over te stappen.

Tot die tijd moet er zorgvuldig overwogen worden of de extra mogelijkheden van Cordova de extra ontwikkeltijd verantwoorden.

Tooling

Terwijl er gewerkt werd aan de progressive web app was één van de meest frustrerende onderdelen van de ontwikkeling het werken met SSL certificaten. Daarom wordt er aangeraden om een service te gebruiken zoals Github Pages wanneer er een web applicaite ontwikkeld wordt. Pages heeft al een geldig certificaat, dat zal veel frustratie schelen.

Daarnaast is er gebruik gemaakt van de Chrome extensie Lighthouse. Lighthouse is een programma dat een rapport genereert over de huidige site waarin gekeken wordt of de website voldoet aan de eisen van een progressive web app.

De extensie rapporteert waar enige problemen zijn en is een zeer waardevolle debug-tool.

LITERATUURLIJST

Amazon. (z.d.-a). Amazon Simple Notification Service (SNS) | AWS. Geraadpleegd op 29 maart 2019, van <https://aws.amazon.com/sns/>

Amazon. (z.d.-b). What is Amazon Simple Notification Service? - Amazon Simple Notification Service. Geraadpleegd op 29 maart 2019, van <https://docs.aws.amazon.com/sns/latest/dg/welcome.html>

Amazon. (z.d.-c). Pricing. Geraadpleegd op 12 april 2019, van <https://aws.amazon.com/sns/pricing/>

Apple. (z.d.). UserNotifications | Apple Developer Documentation. Geraadpleegd op 10 april 2019, van <https://developer.apple.com/documentation/usernotifications>

Chemerisuk, M. (2019, 22 maart). npm: cordova-plugin-firebase-messaging. Geraadpleegd op 28 maart 2019, van <https://www.npmjs.com/package/cordova-plugin-firebase-messaging>

Firebase. (z.d.-a). Firebase. Geraadpleegd op 28 maart 2019, van <https://firebase.google.com>

Firebase. (z.d.-b). Firebase Pricing. Geraadpleegd op 29 maart 2019, van <https://firebase.google.com/pricing/>

Firebase. (2019a, 28 maart). About FCM messages | Firebase. Geraadpleegd op 29 maart 2019, van <https://firebase.google.com/docs/cloud-messaging/concept-options>

Firebase. (2019b, 19 februari). Firebase Cloud Messaging | Firebase. Geraadpleegd op 28 maart 2019, van <https://firebase.google.com/docs/cloud-messaging/>

Firebase. (2019c, 14 maart). Set up a Firebase Cloud Messaging client app on iOS | Firebase. Geraadpleegd op 28 maart 2019, van <https://firebase.google.com/docs/cloud-messaging/ios/client>

Firebase. (2019d, 25 maart). Set User Properties | Firebase. Geraadpleegd op 28 maart 2019, van <https://firebase.google.com/docs/analytics/android/properties>

Firebase. (2019e, 28 maart). Send messages to device groups on Android | Firebase. Geraadpleegd op 28 maart 2019, van <https://firebase.google.com/docs/cloud-messaging/android/device-group>

Firebase. (2019f, 6 april). Add Firebase to your Android project | Firebase. Geraadpleegd op 8 april 2019, van <https://firebase.google.com/docs/android/setup>

Firebase. (2019g, 14 maart). Firebase Management API | Firebase. Geraadpleegd op 8 april 2019, van <https://firebase.google.com/docs/projects/api/reference/rest/>

Firebase. (2019h, 8 april). Topic messaging on Android | Firebase. Geraadpleegd op 9 april 2019, van <https://firebase.google.com/docs/cloud-messaging/android/topic-messaging>

Firebase. (2019i, 14 maart). Firebase Management API | Firebase. Geraadpleegd op 10 april 2019, van <https://firebase.google.com/docs/projects/api/reference/rest/>

Google. (2019a, 29 mei). Desktop Progressive Web Apps | Web | Google Developers. Geraadpleegd op 4 juni 2019, van <https://developers.google.com/web/progressive-web-apps/desktop>

Google. (2019b, 29 mei). Introduction to Push Notifications | Web | Google Developers. Geraadpleegd op 14 juni 2019, van <https://developers.google.com/web/ilt/pwa/introduction-to-push-notifications>

Katzer, S. (z.d.). katzer/cordova-plugin-local-notifications. Geraadpleegd op 4 juni 2019, van <https://github.com/katzer/cordova-plugin-local-notifications>

Klaren, V. (z.d.). Drie low-code softwarevarianten uitgelegd. Geraadpleegd op 14 juni 2019, van <https://blog.thinkwisesoftware.com/blog/drie-low-code-softwarevarianten-uitgelegd>

Lemoine, M. (2018, 19 juni). My journey to bring Web Push support to Node and Electron. Geraadpleegd op 28 maart 2019, van <https://medium.com/@MatthieuLemoine/my-journey-to-bring-web-push-support-to-node-and-electron-ce70eea1c0b0>

Medley, J. (2019, 29 mei). Web Push Notifications: Timely, Relevant, and Precise | Web Fundamentals | Google Developers. Geraadpleegd op 14 juni 2019, van <https://developers.google.com/web/fundamentals/push-notifications/>

Microsoft. (z.d.). Overview - Visual Studio App Center. Geraadpleegd op 10 april 2019, van <https://docs.microsoft.com/en-us/appcenter/>

Microsoft. (2014, 17 maart). Deployment Patterns. Geraadpleegd op 17 maart 2019, van [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff646997\(v=pandp.10\)](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff646997(v=pandp.10))

Microsoft. (2017, 19 mei). Windows Push Notification Services (WNS) overview - Windows UWP applications. Geraadpleegd op 29 maart 2019, van <https://docs.microsoft.com/en-us/windows/uwp/design/shell/tiles-and-notifications/windows-push-notification-services--wns--overview>

Microsoft. (2018, 25 april). Introduction to ASP.NET Core SignalR. Geraadpleegd op 28 maart 2019, van <https://docs.microsoft.com/en-us/aspnet/core/signalr/introduction?view=aspnetcore-2.2>

Microsoft. (2019a, 22 maart). App Center Push for Android - Visual Studio App Center. Geraadpleegd op 29 maart 2019, van <https://docs.microsoft.com/en-us/appcenter/sdk/push/android>

Microsoft. (2019b, 11 februari). App Center Push - Visual Studio App Center. Geraadpleegd op 29 maart 2019, van <https://docs.microsoft.com/en-us/appcenter/push/>

Microsoft. (2019c, 15 februari). Sending Push Notifications - Visual Studio App Center. Geraadpleegd op 29 maart 2019, van <https://docs.microsoft.com/en-us/appcenter/push/send-notification>

Microsoft. (2019d, 11 februari). Using the Push API - Visual Studio App Center. Geraadpleegd op 10 april 2019, van <https://docs.microsoft.com/en-us/appcenter/push/rest-api>

OneSignal. (z.d.-a). Web Push Quickstart. Geraadpleegd op 12 april 2019, van <https://documentation.onesignal.com/docs/web-push-quickstart>

OneSignal. (z.d.-b). OneSignal/OneSignal-Cordova-SDK. Geraadpleegd op 12 april 2019, van <https://github.com/OneSignal/OneSignal-Cordova-SDK>

OneSignal. (z.d.-c). Setup. Geraadpleegd op 12 april 2019, van <https://documentation.onesignal.com/docs/setup>

OneSignal. (z.d.-d). Add Data Tags. Geraadpleegd op 12 april 2019, van <https://documentation.onesignal.com/docs/add-user-data-tags>

OneSignal. (z.d.-e). OneSignal Push Notification Service Documentation. Geraadpleegd op 12 april 2019, van <https://documentation.onesignal.com/reference>

OneSignal. (z.d.-f). Pricing - OneSignal. Geraadpleegd op 12 april 2019, van <https://docs.urbanairship.com/tutorials/audience/target-specific-users/>

OneSignal. (z.d.-g). Product Overview. Geraadpleegd op 12 april 2019, van <https://documentation.onesignal.com/docs>

OneSignal. (2019, 19 maart). High Volume Mobile and Web Push Notifications - OneSignal. Geraadpleegd op 12 april 2019, van <https://www.onesignal.com>

React – A JavaScript library for building user interfaces. (z.d.). Geraadpleegd op 31 mei 2019, van <https://reactjs.org>

Urban Airship. (z.d.-a). What Is Urban Airship? ·Urban Airship Docs. Geraadpleegd op 12 april 2019, van <https://docs.urbanairship.com/tutorials/getting-started/overview/about-ua/>

Urban Airship. (z.d.-b). urbanairship/urbanairship-cordova. Geraadpleegd op 12 april 2019, van <https://github.com/urbanairship/urbanairship-cordova>

Urban Airship. (z.d.-c). Web Push Notification Tutorial ·Urban Airship Docs. Geraadpleegd op 12 april 2019, van <https://docs.urbanairship.com/tutorials/messages/web-push-notification/>

Urban Airship. (z.d.-d). Urban Airship API Reference ·Urban Airship Docs. Geraadpleegd op 12 april 2019, van <https://docs.urbanairship.com/api/ua/>

Urban Airship. (z.d.-e). Uploaded List Tutorial ·Urban Airship Docs. Geraadpleegd op 12 april 2019, van <https://docs.urbanairship.com/tutorials/audience/uploaded-list/>

Urban Airship. (z.d.-f). Target Specific Users ·Urban Airship Docs. Geraadpleegd op 12 april 2019, van <https://docs.urbanairship.com/tutorials/audience/target-specific-users/>

Urban Airship. (z.d.-g). Tiered Pricing | Legal | Urban Airship. Geraadpleegd op 15 april 2019, van <https://www.urbanairship.com/legal/tiered-pricing>



**think
wise.**

What are WebSockets? | Pusher. (z.d.). Geraadpleegd op 14 juni 2019, van <https://pusher.com/websockets>

VERSIEBERICHT

Versie nr.	Beschrijving	Datum
V0.1	Eerste Concept versie	15-03-2019
V1.0	Conceptversie ingelever bij Saxion, De feedback van de eerste versie is verwerkt.	11-06-2019
V1.1	Methodiek hoofdstuk toegevoeg, spelling/grammatica verbeterd, feedback op concept versie verwerkt.	17-06-2019

BIJLAGE 1

Backlog

1. Schrijf het plan van aanpak
2. Bouw interface voor push notificaties
3. Voeg time-to-live attribuut toe aan berichten
4. Bouw functie voor het sturen van een notificatie naar een lijst devices
5. Bouw functie voor het sturen van een notificatie naar één device
6. Schrijf de eerste versie van het stage verslag
7. Voeg functionaliteit voor custom data toe
8. Bouw een React PWA die notificaties kan ontvangen
9. Bouw een Cordova applicatie die notificaties kan ontvangen
10. Schrijf een opsomming van de mogelijkheden voor interactie
11. Bouw interactie in de Cordova applicatie
12. Bouw interactie in de PWA
13. Herbouw PWA zonder React
14. Schrijf het stage verslag.
15. Test de Cordova App op iOS
16. Breidt de PWA uit voor iOS

BIJLAGE 2

Functioneel Ontwerp



STAGEVERSLAG

Push notificaties in een low-code platform

[Abstract](#)
Stageverslag

Marten Krings – 412293 – Saxion
412293@student.saxion.nl

INHOUD

Inleiding	49
Begrippenlijst	50
Requirements	51
Must	51
Could.....	51
Should	52
Scenario's	53
Real-time Push Notificaties	53
Push notificaties naar specifieke gebruikers en groepen	53
Push notificaties naar enkel actieve gebruikers (of andere filters)	53
Push notificaties met standaard acties	54
Push notificaties met deeplinks	54
Push notificaties met een vervaltermijn	54
Interactieve push notificaties	54
Versiebericht.....	44

INLEIDING

In dit document wordt beschreven over welke functionaliteit de push notificaties moeten hebben in de Software Factory. Het doel van dit document is het vormen van een duidelijk beeld van de functionele eisen aan push notificaties voor het Thinkwise Platform.



BEGRIPPENLIJST

Begrip	Definitie
applicatie-model	Het model dat de informatie bevat die het Thinkwise Platform nodig heeft om een applicatie te genereren.
deeplinks	Een link die verwijst naar een locatie in een applicatie i.p.v. een website.

REQUIREMENTS

MUST

Real-time push notificaties

De notificaties moeten in real-time ontvangen worden door de gebruiker.

Push notificaties naar specifieke gebruikers en groepen

Push notificaties moeten verstuurd kunnen worden naar specifieke gebruikers en naar segmenten van de gebruikers van de applicatie. Deze segmenten zijn vooraf gedefinieerd en opgeslagen in de onderliggende database.

Push notificaties naar enkel actieve gebruikers (of andere filters)

Push notificaties moeten naar secties van de gebruikers gestuurd worden. Deze groepen moeten op run time bijgehouden worden.

Push notificaties naar offline gebruikers

Push notificaties moeten ook ontvangen worden als de gebruiker geen internet verbinding heeft op het moment dat de notificatie verstuurd wordt. Wanneer de gebruiker weer online komt moet de notificatie worden ontvangen.

COULD

Push notificaties met standaard acties

Het aanklikken van een notificatie moet applicatie code kunnen laten uitvoeren. Bijvoorbeeld: het automatisch uitloggen van gebruikers of het verversen van het applicatie-model.

Push notificaties met deeplinks

Het klikken van een push notificatie moet een gebruiker naar de relevante plaats in de applicatie kunnen sturen en deze schermen openen als ze dat nog niet zijn.

Push notificaties met een vervaltijd

Push notificaties zijn op een gegeven moment niet meer relevant voor de eindgebruiker, in dat geval moet er een verval termijn bestaan voor de notificatie. Wanneer dit termijn voorbij is moet de notificatie niet meer verstuurd worden naar deze gebruiker. Dit is relevant wanneer de gebruiker langere tijd de applicatie niet heeft geopend.

SHOULD

Interactieve push notificaties

Push notificaties moeten input toestaan, bijvoorbeeld de mogelijkheid om een inlogpoging goed te keuren voor twee factor authenticatie.

SCENARIO'S

Thinkwise heeft twee soorten klanten, klanten die een systeem afnemen en ook eindgebruiker zijn van de applicatie, en klanten die een licentie kopen om te ontwikkelen met het Thinkwise Platform.

REAL-TIME PUSH NOTIFICATIES

- Als gebruiker wil ik zo snel mogelijk op de hoogte gesteld te worden van belangrijke informatie.
- Als ontwikkelaar wil ik dat notificaties direct en zonder vertraging aankomen bij gebruikers.

PUSH NOTIFICATIES NAAR SPECIFIEKE GEBRUIKERS EN GROEPEN

- Als gebruiker wil ik alleen notificaties ontvangen die voor mij relevant zijn.
- Als ontwikkelaar wil ik notificaties kunnen sturen naar specifieke gebruikers of naar groepen van gebruikers.

PUSH NOTIFICATIES NAAR ENKEL ACTIEVE GEBRUIKERS (OF ANDERE FILTERS)

- Als gebruiker wil ik geen berichten ontvangen wanneer deze niet langer relevant zijn.
- Als ontwikkelaar wil ik notificaties kunnen sturen naar gebruikers die op dat moment online zijn, omdat mijn bericht alleen op dit moment relevant is.

PUSH NOTIFICATIES NAAR OFFLINE GEBRUIKERS

- Als gebruiker wil ik geen belangrijke informatie missen vanwege een instabiele internetverbinding

- Als ontwikkelaar wil ik notificaties kunnen sturen naar gebruikers die op dat moment de applicatie niet gestart of actief hebben.

PUSH NOTIFICATIES MET STANDAARD ACTIES

- Als gebruiker wil ik de applicatie zo veel mogelijk werk laten doen om sneller te kunnen werken.
- Als ontwikkelaar wil ik middels notificaties een gebruiker kunnen laten uitloggen door de applicatie, de applicatie het model opnieuw kunnen laten inladen en de applicatie kunnen laten herstarten.

PUSH NOTIFICATIES MET DEEPLINKS

- Als gebruiker wil ik niet hoeven zoeken in de applicatie om de pagina te vinden die relevant is voor de notificatie die ik heb ontvangen.
- Als ontwikkelaar wil ik notificaties kunnen sturen met deeplinks waarbij de gebruiker na klikken op de deeplink naar een specifieke locatie binnen de applicatie worden geleid.

PUSH NOTIFICATIES MET EEN VERVALTERMIJN

- Als gebruiker wil ik geen verouderde notificaties ontvangen aangezien ze niet langer relevant zijn.
- Als ontwikkelaar wil ik notificaties sturen met een vervalttermijn zodat deze notificaties niet meer getoond worden na het verstrijken van de vervalttermijn, aangezien ze niet langer relevant zijn.

INTERACTIEVE PUSH NOTIFICATIES

- Als gebruiker wil ik de applicatie niet te hoeven openen als dat niet nodig is, dan kan ik sneller werken.
- Als ontwikkelaar wil ik notificaties sturen waarop de gebruiker kan reageren middels knoppen of tekst invoer zonder hiervoor de applicatie te hoeven openen