Welke platforms zijn geschikt voor het project?

Door:

* Sander Gelderman(1443143)
* Silas Vaessens(1820435)

Datum:

* 09-03-2020

Module:

* BM07 Cross Platform App Development

Inhoudsopgave

[Inleiding 2](#_Toc34647066)

[Native applicaties 2](#_Toc34647067)

[Mobiele web applicaties 2](#_Toc34647068)

[Hybrid applicaties 3](#_Toc34647069)

[Conclusie native, web en hybrid 3](#_Toc34647070)

[Keuze criteria 3](#_Toc34647071)

[Keuze 4](#_Toc34647072)

[Bibliografie 5](#_Toc34647073)

# Inleiding

Uit het gesprek met Ramon Jongen is gebleken dat er een mobiele applicatie nodig is die zowel op IOS als op Android apparaten beschikbaar is. Op de applicatie moet routes voor slechtziende en blinde mensen worden toegevoegd. Daarbij moeten er oefeningen op bepaalde punten gegeven kunnen worden. In het vervolg van dit bestand wordt er naar de applicatie gerefereerd met de naam: route-applicatie. Om de route-applicatie te ontwikkelen zijn er drie standaard mogelijkheden. Er kan voor gekozen worden voor het bouwen van twee native applicaties, een mobiele web applicatie of een hybrid applicatie. Allereerst wordt er onderzocht wat de voor -en nadelen zijn van de drie opties. Vervolgens wordt er een keuze gemaakt welke van de drie opties het best past bij de route-applicatie. Daarna zullen de verschillende platformen voor die optie afgewogen worden en wordt er vervolgens een platform gekozen die het beste past voor de applicatie.

# Native applicaties

Een Native applicatie is een applicatie die ontwikkeld is voor het gebruik op een specifieke platform of apparaat. Als een applicatie ontwikkeld dient te worden voor meerdere platforms dan zullen er ook meerdere native applicaties gebouwd moeten worden. De voordelen hiervan is dat de gebruikers ervaring hiermee erg positief is, omdat de applicaties voor de specifieke apparaten gebouwd zijn (Milkers, 2016). Er wordt vaak beter gebruik gemaakt van de mogelijkheden die de verschillende devices bieden. Echter is het voor de route-applicatie niet de meest ideale optie. De opdrachtgever heeft aangegeven dat de applicatie niet wordt onderhouden en het dus afhankelijk is van nieuwe studenten. Als twee native applicaties onderhouden moeten worden, kost dit extra werk en tijd. Daarnaast zal het ook extra tijd gaan kosten voor het ontwikkelen van de applicatie. Aangezien de groep die de applicatie gaat uitwerken beperkt de tijd heeft (10 weken), kan dit problemen veroorzaken.

# Mobiele web applicaties

De mobiele web applicatie is een applicatie die specifiek gemaakt is voor smartphones en tablets. Daarnaast is de applicatie ook te bereiken door middel van een browser op een mobiel apparaat. Het grote voordeel van een mobiele web applicatie (vooral ten opzichte van een native applicatie) is dat de applicatie voor meerdere apparaten gemaakt kan worden (cross-platform). Voor de route-applicatie is dit gunstig, omdat de route-applicatie voor zowel IOS als Android apparaten gebruikt moet kunnen worden. Daarnaast is de web app makkelijk en goedkoop te maken. HTML/CSS/Javascript zijn de standaard talen die vaak gebruikt worden. Dit zijn eenvoudige talen om te leren. Dit kan veel tijd schelen bij het uitwerken van de opdracht. Een belangrijk voordeel is ook dat de applicatie makkelijk te onderhouden is. Vooral ook omdat de opdrachtgever aangegeven heeft geen extra budget vrij te maken om de applicatie te onderhouden. Er zijn echter ook een aantal nadelen van de mobiele web applicatie. In het onderzoek van William Jobe (Jobe, 2013) wordt geconcludeerd dat de GPS locatie vaak niet correct was. Dit is een groot probleem voor de route-applicatie, aangezien blinden en slechtzienden dan niet meer weten waar ze zijn. De mobiele web applicaties kunnen ook niet in de app store gedeeld worden. Dit was een van de requirements van de opdrachtgever.

# Hybrid applicaties

De derde optie om een mobiele applicatie te ontwikkelen is de Hybrid applicatie. Een Hybrid app combineert een Web-app met mogelijkheden van een Native besturingssysteem. Bij Hybrid apps wordt een interface vaak weergegeven in een web browser, maar kan er wel door middel van native code gebruik gemaakt worden van specifieke device functionaliteiten. Hybrid applicaties zorgen voor een besparing van ontwikkeltijd. Een Hybrid applicatie kan in tegenstelling tot de mobiele web applicatie wel gedownload worden van de app store, er zijn echter wel meer eisen voor Hybrid apps ten opzichte van native apps. Er hoeft echter niet zoals bij de native apps een geheel nieuwe app ontwikkeld worden. Een belangrijk voordeel voor de route-applicatie is dat de onderhoudskosten normaal gesproken laag blijven. Een nadeel ten opzichte van native applicaties is echter wel dat geavanceerde functionaliteiten niet altijd beschikbaar kunnen zijn voor HTML/CSS/javascript.

# Conclusie native, web en hybrid

Na het analyseren van de voor en nadelen van de drie verschillende opties is de conclusie dat de Hybrid applicatie het meest geschikt is om te gebruiken bij de route-applicatie. De voordelen van native en web worden goed gecombineerd bij deze optie. De belangrijkste overwegingen waren dat de onderhoudskosten beperkt zijn, de applicatie kan in de app store gezet worden en device specifieke functionaliteiten kunnen vaak nog steeds gebruikt worden. Nu er gekozen is voor een Hybrid applicatie, moet er bepaald worden welk platform het meest geschikt is voor de route-applicatie. Aangezien de onderhoud eenvoudig moet zijn, wordt er in dit onderzoek aandacht besteed aan de meest gebruikte cross-platforms. De platforms die onderzocht gaan worden zijn: **Xamarin, Flutter, React Native, Ionic en NativeScript**. Deze platforms worden onderzocht omdat ze bij de meest gebruikte platforms horen volgens de artikelen *Top Cross Platform App Development Frameworks in 2020* (Dev Technosys, 2020) en *Where Do Cross-Platform App Frameworks Stand in 2020?* (Manchanda, 2019).

# Keuze criteria

Wat voor soort platform er gebruikt gaat worden hangt af van een aantal requirements. Meeste van deze requirements zijn te vinden in het document *PvE 1a bewegingsapp voor visuele beperking* (Gelderman, Van Winssen, & Vaessens, 2020). Omdat niet alle requirements invloed hebben, worden hier alleen de requirements neergezet die wel invloed hebben op de keuze van het platform. Daarnaast staan hieronder ook een aantal eisen die niet in *PvE 1a bewegingsapp voor visuele beperking* (Gelderman et al., 2020) staan.

1. Moet Android en iOS ondersteunen
2. Moet gebruik kunnen maken van GPS
3. Ondersteuning Maps en basisfunctionaliteiten Maps (zowel voor Android en IOS)
4. Licenties waar ze gebruik van maken
5. Mate van gemak van onderhoud

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Ondersteunt android | Ondersteunt iOS | Ondersteunt GPS | Ondersteund Maps | licentie |
| Xamarin | x | x | x | x | MIT |
| React Native | x | x | x | x | MIT |
| Flutter | x | x | x | x | New BSD |
| Ionic | x | x | x | x | MIT |
| NativeScript | x | x | x | x | Apache 2.0 |

# Keuze

In de tabel is te zien dat de vijf platforms voldoen aan de basisvoorwaarden. Er zijn echter naast de basisvoorwaarden ook andere redenen om voor een specifiek platform te kiezen. Wij hebben gekozen om het platform Xamarin te gebruiken. Xamarin is een ouder programma en er is veel documentatie over te vinden. Daarnaast geeft Xamarin de mogelijkheid om veel native te werken, waardoor de kwaliteit van de app beter kan worden door geavanceerde functies van iOS en Android. Daarnaast wordt er in Xamarin geprogrammeerd door C#. De deelnemers willen daarnaast meer ervaring op doen met C#, omdat het een programmeertaal is met veel mogelijkheden.

# Bibliografie

Dev Technosys. (2020, Februari 21). *Top Cross Platform App Development Frameworks in 2020*. Opgehaald van Mobile App Daily: https://www.mobileappdaily.com/top-cross-platform-app-development-frameworks

Drifty Co. (2016, Januari 08). *ionic / LICENSE*. Opgehaald van Github: https://github.com/ionic-team/ionic/blob/master/LICENSE

Facebook. (2018, September 12). *react-native / LICENSE*. Opgehaald van Github: https://github.com/facebook/react-native/blob/master/LICENSE

Gelderman, S., van Winssen, T., & Vaessens, S. (2020, Maart 03). PvE 1a bewegingsapp voor visuele beperking. Heerlen, Limburg, Nederland.

Google. (2019, November 28). *flutter / LICENSE*. Opgehaald van Github: https://github.com/flutter/flutter/blob/master/LICENSE

Jobe, W. (2013). Native Apps vs. Mobile Web Apps. *International Journal of Interactive Mobile Technologies*, 32.

Laurent, A. M. (2004). *Understanding Open Source and Free Software Licensing.* O'Reilly Media.

Manchanda, A. (2019, December 06). *Where Do Cross-Platform App Frameworks Stand in 2020?* Opgehaald van net solutions: https://www.netsolutions.com/insights/cross-platform-app-frameworks-in-2019/

Microsoft. (2019, April 29). *Xamerin.Forms / LICENSE*. Opgehaald van Github: https://github.com/xamarin/Xamarin.Forms/blob/master/LICENSE

Milkers. (2016, Augustus 15). *Mobile Web Apps vs. Native Apps: make the right choice*. Opgehaald van slideshare: https://www.slideshare.net/milkers/mobile-web-apps-vs-mobile-native-apps-how-to-make-the-right-choice

Progress Software. (2019, Januari 07). *NativeScript / LICENSE*. Opgehaald van Github: https://github.com/NativeScript/NativeScript/blob/master/LICENSE

Licenties:

Flutter: New BSD License

BSD: modify code, in schrijven dat je iemand anders shsit gebruikt. Niet iemand anders naam gebruiken voor promotie. Include everything.

ReactNative: MIT License

NativeScript: Apache 2.0 software License

Apache: don’t have to include everything, single reference. Commercial: mensen kunnen contributen makkelijker,

Ionic: MIT License

MIT: kan alles veranderen, alleen include license.

Xamarin: MIT License

<https://play.google.com/intl/nl/about/developer-content-policy-print/>

<https://developer.apple.com/app-store/review/guidelines/>