

Tracker

Automatisch werkuren tracken met locatie

door Elias Meire, Haroen Viaene en Arnaud Weyts  
Bachelor in de Elektronica-ICT, afstudeerrichting ICT 2015-2016  
Mentor: Bram Gosseye

# Inhoudsopgave

[Inleiding 2](#_Toc445477000)

[1. Mogelijke oplossingen en gekozen oplossing 3](#_Toc445477001)

[1.1 Front-end 3](#_Toc445477002)

[1.1.1 Native applicatie 3](#_Toc445477003)

[1.1.2 Hybride applicatie 4](#_Toc445477004)

[1.1.3 Responsive website 4](#_Toc445477005)

[1.1.4 Gekozen front-end 5](#_Toc445477006)

[1.2 Back-end 5](#_Toc445477007)

[1.2.1 Eigen oplossing 5](#_Toc445477008)

[1.2.2 MBaaS 5](#_Toc445477009)

[1.2.3 Gekozen back-end 6](#_Toc445477010)

[2. Technische detailuitwerking van gekozen oplossing (met controle en verificatie) 7](#_Toc445477011)

[3. Risicoanalyse 8](#_Toc445477012)

[4. Kostenraming en levensduur 9](#_Toc445477013)

[Conclusie 10](#_Toc445477014)

[Literatuurlijst 11](#_Toc445477015)

[Bijlage 1: Uiteindelijke werkplanning 12](#_Toc445477016)

# Inleiding

Momenteel is het moeilijk om als werkgever bij te houden hoeveel uur je werknemers op verplaatsing werken. Vooral bij kleine zelfstandigen worden de werkuren veelal nog handmatig ingebracht door de werknemers. Dat legt de verantwoordelijkheid voor de juistheid van deze gegevens bij de werknemers, waardoor deze foutief kunnen zijn. Voor de werknemers kan het ook storend zijn om steeds te moeten onthouden waar ze geweest zijn en hoe lang dat duurde.

Een oplossing hiervoor is een applicatie die alle werkuren en verplaatsingen van de medewerkers bijhoudt op een centrale plaats. Zo kan de werkgever deze op een overzichtelijke manier beheren en controleren. Tevens zou de werkgever hieruit gemakkelijk de loonkosten kunnen berekenen. Deze applicatie kan men onderverdelen in 3 deelaspecten: een interface voor de werkgever, een interface voor de werknemers en een back-end waarin de gegevens van de werknemers worden opgeslagen.

Het doel is om als werknemer de app te kunnen gebruiken om de werkuren in te vullen. Dit kan ook geautomatiseerd worden met behulp van een locatieservice. De werkgever kan de gewerkte uren van zijn werknemers bekijken via de webinterface. Dit kan later uitgebreid worden zodat ook de werkgever toegang heeft tot een webinterface en omgekeerd ook de werkgever. Omwille van de tijdsbeperking van twaalf weken voor dit project wordt er beslist een app voor Android bruikbaar voor werknemers, en een webinterface bruikbaar voor werkgevers te maken.

Een belangrijk vraagstuk in elke applicatie die gebruikersdata verzamelt, is beveiliging. Er wordt met aandacht nagedacht over implicaties van beslissingen in verband met privacy en op elk moment wordt gecontroleerd of het mogelijk is aan bepaalde data te komen, zonder correct geauthentificeerd te zijn. Indien dat het geval is, wordt de beveiliging verbeterd.

Bij het maken van de applicatie wordt er gebruik gemaakt van verschillende internetbronnen, documentatie van de gebruikte technologieën, raad van professionals enzovoort. Er is tevens besloten om de app “open source” te ontwikkelen, dit wil zeggen dat alle code zichtbaar zal zijn voor iedereen die er iets uit wil leren.

Dit rapport is op de volgende manier gestructureerd. Allereerst worden de mogelijke oplossingen in detail bestudeerd. Daaropvolgend wordt de gekozen oplossing in technisch detail uitgelegd. Voor de gekozen oplossing wordt vervolgens een gedetailleerde risicoanalyse opgesteld. Ten slotte wordt er een kostenraming gemaakt voor de realisatie en een bespreking opgesteld over de toekomst van het project.

# Mogelijke oplossingen en gekozen oplossing

## Front-end

Het maken van een front-end applicatie kan op veel verschillende methoden. De belangrijkste keuze die gemaakt moet worden is die tussen hybride en native applicatieontwikkeling.

### Native applicatie

Het grootste voordeel voor native ontwikkeling is performantie want in tegenstelling tot hybride applicaties zijn er geen tussenlagen zoals webviews of interpreters. De code van de applicatie kan dus direct gecompileerd en uitgevoerd worden op het apparaat. Bovendien zijn alle API’s van het gekozen platform beschikbaar en goed gedocumenteerd. Bijgevolg heeft een ontwikkelaar veel betere controle over de werking van de applicatie. Als laatste voordeel heeft elk platform zijn eigen IDE die krachtige tools bevat om efficiënt te kunnen ontwikkelen zoals debuggers en optimalisatietools. Dat is tevens de manier die Android, iOS en andere mobiele besturingssystemen aanraden om apps te ontwikkelen [1].

Native applicatie brengt echter wel enkele nadelen met zich mee, zo zal de code van de applicatie volledig specifiek zijn aan het platform. In het geval dat je dus een native app voor meerdere platformen zou willen maken, zouden deze geen code gemeenschappelijk hebben [2]. Er zouden dus eigenlijk twee aparte applicaties moeten geschreven worden.

Het grootste nadeel van native apps is dat ze platformspecifiek zijn en dus zou er in het kader van dit project een platform moeten worden gekozen. Bijgevolg wordt er een aanzienlijk aantal potentiele gebruikers verloren. In figuur 1 zijn de wereldwijde marktaandelen van smartphones per besturingssysteem te zien. Hieruit blijkt dat Android het grootste platform is, en dus de beste kandidaat lijkt. Als bijkomend voordeel wordt er op Android geprogrammeerd in Java, een programmeertaal waar veel programmeurs mee vertrouwd zijn. Java wordt bovendien in het eerste jaar van de opleiding “bachelor in de elektronica-ICT” behandeld [3].

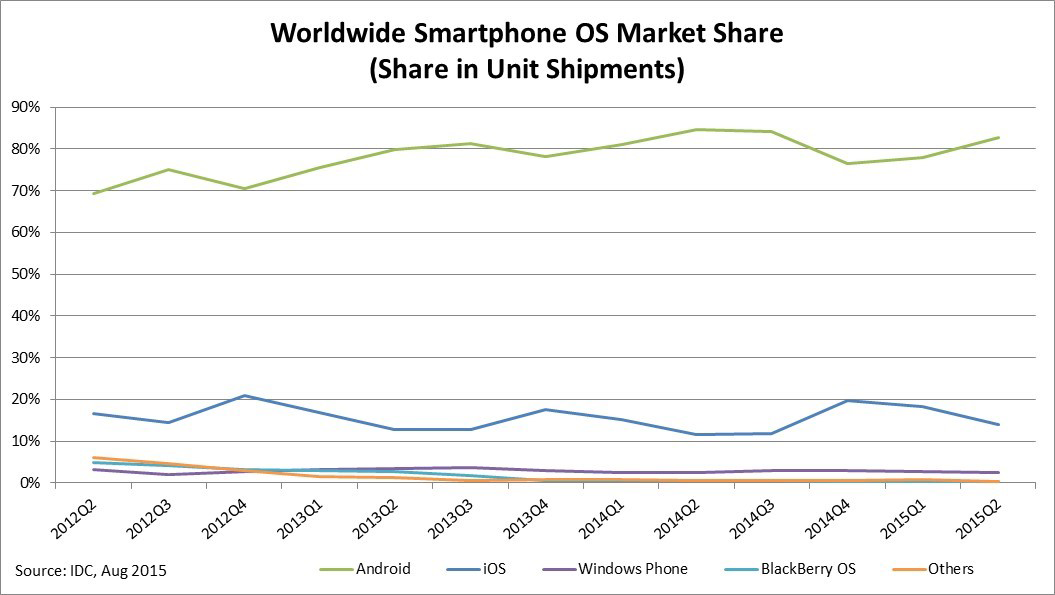


Figure 1: Marktaandeel smartphone besturingssystemen [4]

### Hybride applicatie

Een alternatief voor een native app is een hybride app, waarin alle code geschreven wordt in één taal of framework. Hierbij wordt de app meestal volledig in een webview gemaakt. Dit wil zeggen dat alles in html, CSS en Javascript geschreven is. De populairste frameworks die deze strategie gebruiken zijn Ionic, Phonegap en Cordova [4]. Het grote voordeel aan deze werkwijze is uiteraard dat je maar één codebase hebt om apps te maken voor alle platformen. Hierdoor kan er efficiënt in team gewerkt worden en snel een applicatie voor meerdere platformen worden gerealiseerd.

Het grootste nadeel bij een hybride app is dat een hybride app steeds weinig controle zal hebben over hoe de code uiteindelijk zal uitgevoerd worden op de hardware. Dit heeft ook als gevolg dat een hybride app nooit dezelfde performantie zal behalen als een native app [5].

Een nieuwkomer in deze categorie is React Native. Deze maakt gebruik van React als framework en gebruikt Javascript als programmeertaal voor elk aspect van de app. Het grote voordeel bij React Native is dat je slechts één codebase hebt voor beide platformen, maar dat er wel gebruik kan gemaakt worden van native componenten van de populairste platformen. Het nadeel hiervan is dat er een grote expertise van ontwikkeling in React vereist is. Bovendien staat React Native nog in zijn kinderschoenen [6].

### Responsive website

Een website heeft als voordeel dat het volledig onafhankelijk van platform is. Elk besturingssysteem met een browser kan een website weergeven. Door gebruik te maken van responsive design kan men een website gebruiksvriendelijk houden op elke schermgrootte. Een responsive website heeft als groot nadeel dat men geen API’s kan aanspreken van het platform, de website is namelijk gesandboxed in de browser. Zeker het gebruik van de applicatie terwijl die niet specifiek open is, is zeer moeilijk [7].

### Gekozen front-end

Voor de werknemer zal er als front-end geopteerd worden voor een mobiele app. Meer specifiek wordt er gekozen voor een native Android app. Deze optie is gekozen omdat het voor werknemers belangrijk is om toegang te hebben tot de Android API’s om zo efficiënt om te gaan met bijvoorbeeld het bijhouden van locaties. Er wordt gekozen voor Android als platform omwille van het marktaandeel en het gebruik van Java.

Voor werkgevers wordt er gekozen voor een responsive website. Bij de werkgever moeten vooral gegevens uit de back-end getoond worden, hier is het niet belangrijk om toegang te hebben tot alle API’s. Door te kiezen voor een responsive website behouden we maximale compatibiliteit met zowel desktop computers als mobiele apparaten.

## Back-end

Bij het opzetten van een back-end zijn er een aantal opties beschikbaar. Enerzijds is het een mogelijkheid om zelf een back-end te hosten, waarbinnen er ook nog gekozen kan worden tussen verschillende types databanken zoals MySQL, MongoDB, MariaDB ... en verschillende programmeertalen om deze in te schrijven. Anderzijds kan er ook gebruik gemaakt worden van een MBaaS [8].

### Eigen oplossing

Een eigen back-end ontwikkelen brengt een aantal voordelen met zich mee. Je hebt namelijk als ontwikkelaar volledige controle over de back-end. Als gevolg hiervan kan de back-end veel specifieker gemaakt worden en dus sneller zijn. Naarmate de applicatie op grotere schaal gebruikt wordt, zal deze optie ook goedkoper zijn dan een MBaaS. Een groot nadeel van deze oplossing is dat er veel tijd nodig is om een goede back-end te ontwikkelen. Je bent tevens zelf verantwoordelijk voor aspecten zoals beveiliging en hosting.

### MBaaS

Bij een MBaaS wordt er gebruik gemaakt van een soort geprefabriceerde back-end. Alle hosting gebeurt op een platform, zoals Firebase. Een voordeel hiervan is dat er als ontwikkelaar niet moet worden nagedacht over aspecten zoals beveiliging en hosting.

Veel verschillende types MBaaS zorgen ook zelf voor gemakkelijke integratie op populaire platformen zoals Android, iOS, web … waardoor er minder inspanning nodig is om verschillende clients te verbinden met je data.

De beste optie is momenteel Firebase. Deze is vorig jaar aangekocht door Google en biedt naast het verbinden met een database ook een gestroomlijnde manier om met gebruikers te werken. Zo is het mogelijk om via standaardmethoden gebruikers te laten inloggen met OAuth [9].

Het nadeel van het gebruiken van een MBaaS is dat gratis formules meestal slechts een beperkt aantal gebruikers toelaten. In deze toepassing is dit echter niet zo relevant, aangezien er initieel geen groot gebruikersaantal verwacht wordt.

### Gekozen back-end

In dit project is er gekozen om met de MBaaS Firebase te werken als back-end omdat deze een goede integratie met OAuth biedt en zo logins en gebruikersbeheer vereenvoudigt. Bijgevolg kan er gefocust worden op het ontwikkelen van de applicatie zelf. Bovendien heeft Firebase een gratis plan dat geen strenge beperkingen stelt naar gebruikersaantal [10].

# Technische detailuitwerking van gekozen oplossing (met controle en verificatie)

# Risicoanalyse

# Kostenraming en levensduur

# Conclusie

# Literatuurlijst

|  |  |
| --- | --- |
| [1] | Wijmo, „Mobile App Development: How to Decide on Hybrid vs. Native,” 10 december 2015. [Online]. Available: http://wijmo.com/mobile-app-development-how-to-decide-on-hybrid-vs-native/. [Geopend 9 maart 2016]. |
| [2] | A. Ziflaj, „Native vs Hybrid App Development,” Sitepoint, 15 augustus 2014. [Online]. Available: http://www.sitepoint.com/native-vs-hybrid-app-development/. [Geopend 8 maart 2016]. |
| [3] | „Programmeren 2,” Odisee, [Online]. Available: http://onderwijsaanbod.odisee.be/2015/syllabi/n/JPW274N.htm#activetab=doelstellingen\_idp1159136. [Geopend 11 maart 2016]. |
| [4] | Ramon Llamas, „Smartphone OS Market Share, 2015 Q2,” IDC, augustus 2015. [Online]. Available: http://www.idc.com/prodserv/smartphone-os-market-share.jsp. [Geopend 1 maart 2016]. |
| [5] | "What are the pros and cons of using Apache Cordova (phone-gap) for building your iOS app?," Quora, 24 juli 2012. [Online]. Available: https://www.quora.com/What-are-the-pros-and-cons-of-using-Apache-Cordova-phone-gap-for-building-your-iOS-app. [Accessed 5 maart 2016]. |
| [6] | R. Kraus, "From Native To Hybrid App Development And Back," TechCrunch, 19 november 2015. [Online]. Available: http://techcrunch.com/2015/11/19/lessons-in-switching-from-native-to-hybrid-app-development-and-back/. [Accessed 3 maart 2016]. |
| [7] | S. Almog, "Java is superior to React Native in practically every way," Codename One, 2 november 2015. [Online]. Available: http://www.codenameone.com/blog/java-is-superior-to-react-native-in-practically-every-way.html. [Accessed 6 maart 2016]. |
| [8] | „It is possible to watch the location in the background on Mobile (iOS / Android)?,” Stackoverflow, 15 november 2015. [Online]. Available: http://stackoverflow.com/questions/33926217/it-is-possible-to-watch-the-location-in-the-background-on-mobile-ios-android. [Geopend 10 maart 2016]. |
| [9] | "Should I use a MBaaS platform (like Parse) or create my own backend?," Quora, [Online]. Available: https://www.quora.com/Should-I-use-a-MBaaS-platform-like-Parse-or-create-my-own-backend. [Accessed 10 maart 2016]. |
| [10] | Firebase, "Firebase Authentication," Firebase, [Online]. Available: https://www.firebase.com/features.html#features-authentication. [Accessed 11 maart 2016]. |
| [11] | Firebase, "Pricing & Plans," Firebase, [Online]. Available: https://www.firebase.com/pricing.html. [Accessed 11 maart 2016]. |

# 

# Bijlage 1: Uiteindelijke werkplanning