MOBIEL EN ITERATIEF BEHEER VAN IOT SENSOREN IN DE TUINBOUW

Analyse- en onderzoeksrapport

Viersen, Jimmy van

Quantified B.V. | Langegracht 70, 2313 NV Leiden
Warner Venstra | wjv@quantifiedair.nl | +31(0)6 51 77 59 25

Hogeschool Leiden | Zernikedreef 11, 2333 CK Leiden
Ami Tolba | tolba.a@hsleiden.nl | +31(0)6 48 13 39 89

Inhoudsopgave

Doelgroep Analyse

Doelstelling

Het doel van deze doelgroep analyse is om vast te leggen wie de eindgebruikers zijn van de mobiele applicatie die er ontwikkeld wordt tijdens mijn afstudeertraject. Omdat de gebruikers verschillende use-cases hebben wil ik hun eisen, wensen en verwachtingen zo duidelijk mogelijk vastleggen om zo een beter beeld te krijgen van de functionaliteiten die aan de applicatie zullen worden meegegeven.

Categorisatie van gegevens

Om een beter beeld te schetsen van de doelgroep heb ik de volgende informatie nodig:

- Demografische kenmerken:
  - Organisatie;
  - Naam (voor eigen administratie);
  - Email adres (voor eigen administratie);
  - Leeftijd;
  - Geslacht;
  - Welke talen spreekt men;

- Geografische kenmerken:
  - Werklocatie;
  - Taal gebruik op werklocatie;

- Sociaaleconomische kenmerken:
  - Functie binnen de organisatie;
  - Model mobiele telefoon persoonlijk;
  - Model mobiele telefoon werk;

- Psychografische kenmerken:
  - Domeinspecifiek:
    - De mate van betrokkenheid in gebruik van de app;
    - Aard van het productgebruik;

  - Merkspecifiek:
    - Gegeven ten opzichte van de naamsbekendheid van Quantified;
    - Gegevens ten opzichte van de naamsbekendheid van de Concurrent;
    - De houding tegenover het soort product;

  - Bereidheid om mee te werken aan de ontwikkeling van de applicatie (door bereikbaar te zijn voor verdere vragen en deel te nemen aan Usability tests);

Informatieverzameling

De informatieverzameling zal voornamelijk van kwalitatieve aard zijn met kwantitatieve - vragen, het gaat hier om creëren van een interpretatie van de doelgroep. De opzet van het afstudeerproject is begeleid door Paul Kengen. Een deel hiervan was het verkrijgen van toestemming voor het afnemen van interviews met een aantal eindgebruikers. Paul heeft hiervoor een selectie gemaakt van bedrijven die belangrijke input kunnen geven. Deze interviews zullen voornamelijk de bron van informatie zijn ter invulling van de in hoofdstuk 2 benoemde gegevens.

Om de interviews af te kunnen nemen zal ik contact moeten leggen met de verkregen personen en afspraken in moeten plannen. Om alles zo toegankelijk mogelijk te houden heb ik gekozen om de interviews via Microsoft Teams af te nemen. Ik heb een overzicht van mogelijk dagen en momenten opgesteld en laat de personen hier zelf een gewenst moment voor prikken.

Voorafgaand aan de interviews heb ik geïnventariseerd binnen Quantified welke informatie er mogelijk al beschikbaar was. Zo heb ik op de werkplaats een aantal product prototypes voorbij zien komen waarmee de geselecteerde bedrijven mee zullen gaan werken. Hier heb ik dan ook de gelegenheid genomen om collega's die eraan werkte te raadplegen over het concept en de manier waarop het gebruikt verwacht te worden.

Dit helpt bij het vormen van een beeld en gevoel voor de mogelijke use-cases die in de interviews naar voren zullen komen. Daarbij heeft Paul ook geholpen om een beeld te schetsen van wat Quantified's verwachtingen zijn van de applicatie, de use-cases en de prototypes die er in ontwikkeling zijn.

Daarnaast heb ik de website van de bedrijven zelf bezocht om een indruk te krijgen van het bedrijf, de medewerkers, de branche en de werkzaamheden.

De interviews zullen worden opgesteld in GoogleForms. Dit is een platform waarin er snel en gemakkelijk vragenlijsten kunnen worden opgesteld, opgeslagen en uitgewerkt worden. Deze vragenlijst zal niet worden uitgegeven aan de deelnemende personen, maar door de afnemer van de interviews ingevuld worden.

In 2013 beschreef Bijker dat er verschillend vaardigheden zijn om gesprekken mee te beïnvloeden om het zo in een gewenste richting te sturen. Mensen zijn sneller geneigd om zich verder open te stellen als men actief deelneemt aan een gesprek. Een van de beschreven vaardigheden is de L:SD-methode, wat staat voor Luisteren, Samenvatten en Doorvragen. Om te zorgen dat er tijdens de interviews zo duidelijk mogelijke antwoorden worden gegeven, is het van belang dat ik deze methode bewust toepas. Hierbij moet er wel worden opgelet dat er niet wordt weggestuurd van de gespreksstof en interviewvragen.

Deelvragen

Persoonlijke informatie

Persoonlijke informatie om contact mee te kunnen houden als de geïnterviewde instemt om verder mee te werken aan de ontwikkeling van de applicatie d.m.v. inzicht en feedback te geven en deel te nemen aan Usability Tests:

- Wat is de naam van de organisatie?
- Wat is de naam van de persoon?
- Via welk emailadres is de persoon benaderbaar?
- Via welk telefoonnummer is de persoon benaderbaar?

Doelgroep

Vragen specifiek gericht op de doelgroep waaruit informatie gewonnen zal worden voor het opstellen van een Persona de User Journey:

- Demografische kenmerken:
  - Wat is de leeftijd van de doelgroep?
  - Wat is het geslacht van de doelgroep?
  - Welke talen spreekt de doelgroep?

- Geografische kenmerken:
  - Op welke locaties (binnen- en buiten Nederland) is de doelgroep werkzaam?
  - Welke taal wordt er gesproken op de werklocaties?

- Sociaaleconomische kenmerken:
  - Wat is de rol/functie van de doelgroep binnen de organisatie?
  - Wat voor model persoonlijke mobiele telefoon heeft de doelgroep?
  - Wat voor model werk mobiele telefoon heeft de doelgroep?
  - In hoeverre denkt de doelgroep bekwaam te zijn met dit toestel (gesteld voor beide varianten)?

- Psychografische kenmerken:
  - Domein specifiek:
    - Wat is de verwachte mate van gebruik van de mobiele applicatie van de doelgroep?
    - Wat is de verwachte aard (use-case) van het gebruik van de mobiele applicatie?
    - Wat wil de doelgroep meten, of wordt er gemeten?
    - Wat wil de doelgroep zien van de mobiele applicatie?
    - Welke eisen stelt de doelgroep aan de mobiele applicatie?
    - Welke verwachtingen heeft de doelgroep van de mobiele applicatie?
    - Welke wensen heeft de doelgroep van de mobiele applicatie?
    - Is de doelgroep bereid om verder mee te werken aan de ontwikkeling d.m.v. interviews, feedbackmomenten en tests?

- Merk specifiek:
  - Is de doelgroep bekend met de oplossing van Quantified?
  - Wat zijn de sterke/zwakke punten van de oplossing?
  - Is de doelgroep bekend met een soortgelijke oplossing van een concurrent?
  - Wie is de concurrent?
  - Wat zijn de sterke/zwakke punten van de concurrent?
  - Wat is de houding van de doelgroep tegenover het soort product?

Dataverwerking

Ordening

De eerste stap voor het verwerken van de verkregen data is het ordenen. Daarvoor heb ik de data heb onderverdeeld in de volgende analyse-eenheden:

- Woorden;
- Zinnen;
- Fragmenten;
- Thema;

Woorden

Er is een selectie gemaakt van opvallende zelfstandige naamwoorden in de reacties, antwoorden en quotes van de deelnemers van de interviews.

Zinnen

Door het stellen van open vragen tijdens de interviews werden de deelnemers gedwongen om uitgebreide antwoorden te geven. Hier zitten zowel direct antwoorden als quotes bij die er zijn vastgelegd. Deze zinnen kunnen dienen als antwoord op een aantal specifieke vragen.

Fragmenten

Fragmenten bestaan uit delen van zinnen, alinea's of quotes, afgerond zodat het een deel vormt wat geïnterpreteerd kan worden. Het zal een duidelijke indruk moeten geven waar het over gaat.

Thema's

Alles wat geen specifieke plaats kan krijgen zal hieronder globaal geïnterpreteerd worden.

Labelen

De volgende stap is het categoriseren of labelen van de onderverdeelde data. Dit kadert de onderwerpen af waar elk stuk data onder zal vallen. Voor het gemak heb ik de leidraad van de deelvragen uit hoofdstuk 4 aangehouden, daar zijn de volgende labels uit naar voren gekomen:

- Demografisch;
- Geografisch;
- Sociaaleconomisch;
- Betrokkenheid;
- Requirements;
- Concurrentie;
- Globaal.

Verbanden leggen

Door de data te labelen of categoriseren is het makkelijker geworden om te filteren op mogelijke antwoorden op de eerder gestelde deelvragen in hoofdstuk 4. Vervolgens kunnen er vanuit de antwoorden nieuwe inzichten worden opgedaan en worden uitgewerkt in een aantal begeleidende deliverables. Dit zijn de Persona en User Journey Map.

Resultaten

Informatie

Voor het opstellen een passend persona heb ik de antwoorden op de volgende deelvragen nodig, dit is verkregen en ingevuld aan de hand van de interviewresultaten:

- *Wie zijn de doelgroep?*
  Uit de interviews kwam naar voren dat de doelgroep klanten van Quantified zijn die gebruik maken van de FireFly sensormodules, waarvoor de mobiele applicatie wordt ontwikkeld waarmee meetgegevens inzichtelijker gemaakt gaan worden.

- *Wat is de motivatie van de doelgroep?*
  Uit de interviews kwam naar voren dat de doelgroep inzicht wilt op verschillende landbouwinstallaties en toepassingen die hun gebruiken voor verschillende doeleinden. De grootste drijfveer voor de doelgroep is het verkrijgen van meetgegevens om zo proactief te kunnen handelen op situaties, om de kwaliteit van hun producten te kunnen waarborgen.

- *Wat zijn de verschillende use-cases van de doelgroep?*
  Uit de interviews kwam naar voren dat de doelgroep de FireFly sensormodules inzet voor verschillende doeleinden, met name gericht op data verzameling vanuit de land- en tuinbouw.

- *Welke Eisen, wensen en verwachtingen heeft de doelgroep van de applicatie?*
  Uit de interviews kwam naar voren dat doelgroep:
  - Een mobiele applciatie wilt;
  - Wat meetgegevens kan inladen en weergeven;
  - In grafieken, geomaps en heatmaps;
  - Waarbij meetpunten afzonderlijk te onderscheiden zijn;
  - Met overlap van meetdata uit meerdere sensormodules;
  - Met instelbare onder- en overschrijdings waardes;
  - Waarbij bij onder- of overschrijding een alarm in de vorm van een pushnotificatie gegeven wordt;
  - Waarmee er bij voorkeur op meerdere apparaten gewerkt kan worden;
  - Waarbij ook meetdata van derden partijen kan worden ingeladen om zo een totaalbeeld te creëren.

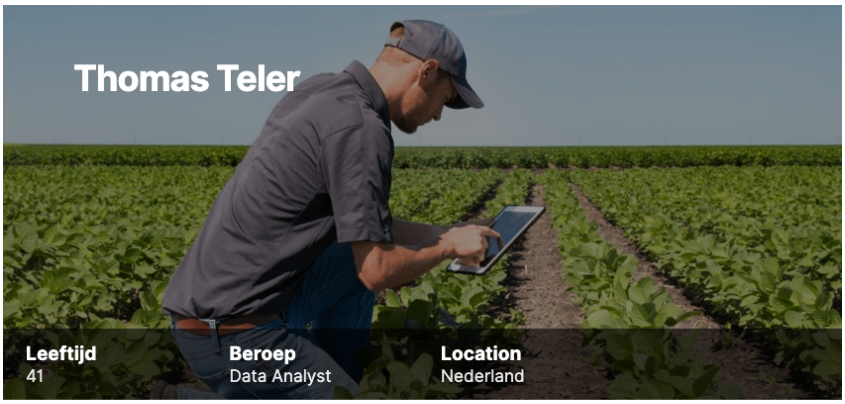
- *Welke mobiele telefoons gebruikt de doelgroep?*
  Uit de interviews kwam naar voren dat de doelgroep voornamelijk werkt met Android toestellen, waarvan de Samsung Galaxy het meest voorkwam.

Persona

Aan de hand van de van de verkregen informatie vanuit de interviews en de antwoorden op de deelvragen heb ik het volgende persona opgesteld.

Dit persona dient als representatie van de gemiddelde gebruiker binnen de doelgroep voor voor dit project.

Tijdens de ontwerpfase zal er regelmatig naar het persona gekeken worden met het oog op het bouwen van een User Experience wat zo nauw mogelijk aansluit deze gebruiker.



Quote

> Ik vertrouw het wel, maar check het toch nog een keer voor de zekerheid.

Bio

Thomas werkt voor een internationale producten en leverancier van groenten en fruit. Om de kwaliteit van de productie te waarborgen moet Thomas inzicht hebben op de sensor data van verschillende meetopstellingen die er in gebruik zijn. Deze staan op verschillende locatie binnen en buiten Nederland, en verzorgen metingen op afstand.

Met deze data kan hij inzien hoe de teelt er bijstaat, of er moet worden ingegrepen en zelfs voorspellen wat er kan gaan gebeuren. Hij kan niet altijd achter de computer zitten en zal dan ook het veld in moeten waar hij dan rondloopt met zijn mobiele telefoon en/of tablet.

Goals

- Visueel, duidelijk overzicht op meerdere projecten
- Voorspellingen doen op weer en klimaat
- Vertrouwen in de productkwaliteit ondersteunen

Frustraties

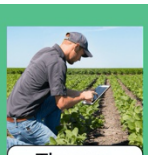
- Onduidelijkheid
- Onnodige handelingen herhalen
- Technologie wat tegenwerkt

Tools

- Desktop
- Samsung Galaxy Smartphone / Tablet
- Quantified Mobile App

User Journey

Ter ondersteuning van het persona is er ook een korte User Journey uitgewerkt, gericht op het werken met de applicatie. Hiermee wordt er een beeld geschetst over hoe de gebruiker van plan is om te gaan werken met de applicatie in de praktijk, gericht op de happy flow van het scenario. Hierin staan ook een aantal mogelijke knelpunten gebaseerd op het persona en een aantal “Kansen” voor verbetering in het proces.



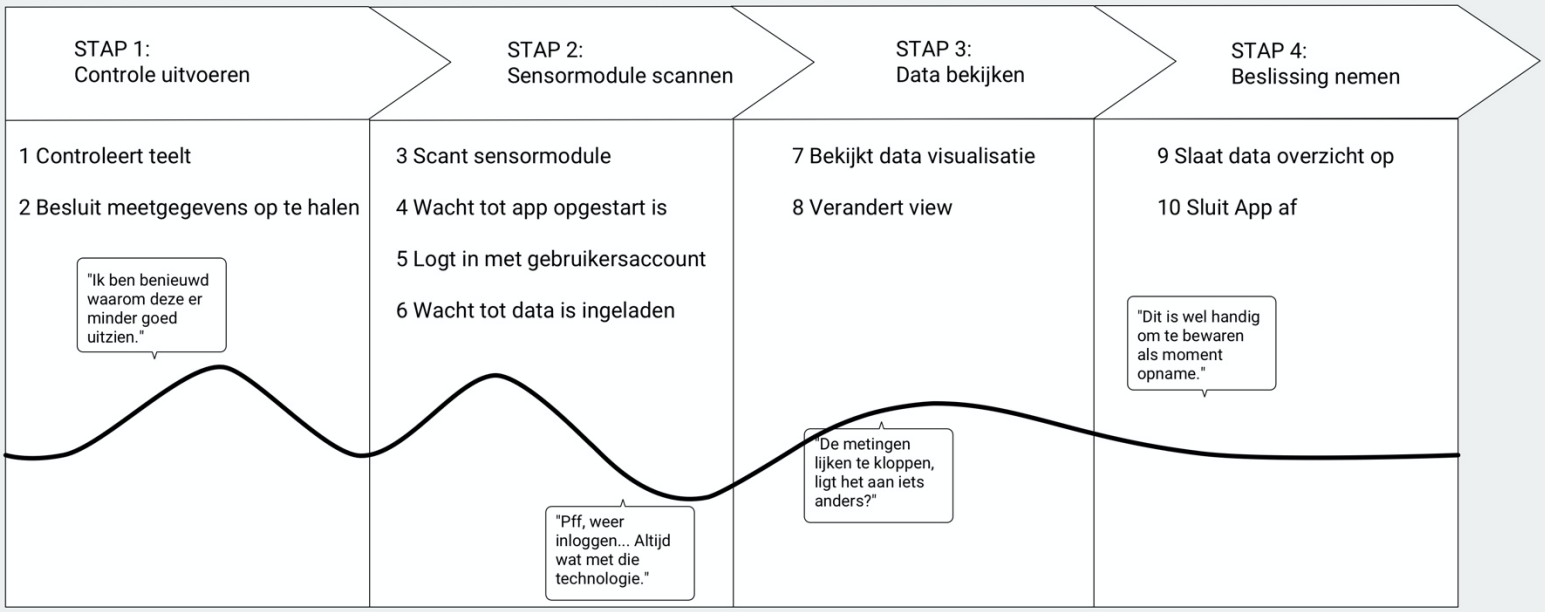
Thomas

Scenario

Thomas loopt in het veld om de teelt te controleren. Als er delen zijn waarbij de kwaliteit minder lijkt te zijn wilt hij de sensormodule met zijn mobiele telefoon scannen om zo de sensordata te kunnen inzien.

Doel en verwachtingen

- Gevisualiseerd en real-time overzicht van de sensordata
- Mogelijkheid om de visualisatie categorie te veranderen
- Snelle en simpele bediening met weinig handelingen



Kansen voor verbetering

- Vergelijken van overzichten van verschillende sensormodules
- Inzicht in verschillende meetpunten
- Downloaden van een overzicht
- Snelle login als de app recent gebruikt is

Wordcloud

Vanuit de selectie aan opvallende zelfstandige naamwoorden heb ik een wordcloud gegenereerd. Een wordcloud is een handige manier om grote hoeveelheden tekstuele data overzichtelijk te maken.

In de wordcloud springen er direct een aantal woorden uit vergeleken met de rest. Dit zijn onderwerpen waar de deelnemers van de interviews veel waarde aan hechten. Er zit een overlap aan functionele eisen, wensen en verwachtingen in, verschillende meetpunten waarmee gewerkt wordt of gewerkt gaat worden, mobiele telefoon modellen en operating systemen, etc.

Programma van Eisen

Quantified wil de gebruikers van een aantal nieuwe sensortoepassingen bedienen door het ontwikkelen van een mobiele applicatie. Deze applicatie dient de volgende doelen:

1 Een mobiel platform voor het werken met de FireFly sensormodules;
2 De mogelijkheid voor realtime inzicht op data van een specifieke sensormodule (op een gewenst moment);
3 Het visueel inzichtelijk maken van data verkregen uit sensormetingen;

De doelgroep van de toekomstige applicatie omvat organisaties die werken met sensormetingen binnen de land- en tuinbouw branche.

Quantified wil een applicatie ontwikkelen doe de volgende taken uitvoert:

1 Scannen van een QR-code op het productlabel op de sensormodule;
2 Ophalen van sensordata gekoppeld aan de gescande sensormodule;
3 Naar aanleiding van de wensen van de gebruiker de data visualiseren;

Gebruikers moeten de optie krijgen om zelf te kunnen schakelen tussen verschillende sensormodules die gekoppeld zijn aan hun gebruikersaccount, zoals nu geregistreerd staat bij het Quantified web portaal: insight.

Om Quantifed te helpen bij de ontwikkeling van deze applicatie is er een doelgroep analyse uitgevoerd. Hierin is er duidelijk gemaakt hoe de gebruikers het toekomstige product graag in gebruik willen nemen en wat ervan verwacht wordt.

Op het gebied van bediening van de gebruikersverwachtingen is er afgesproken om samen te werken met de doelgroep, medewerkers van “Van OERS United” en “Universal Greenfields”. Beide partijen zullen bereikbaar zijn voor informatie, terugkoppeling, overeenstemming en deelname aan Usability Tests, waarbij er de gelegenheid is voor het uitspreken van hun verwachtingen, wensen en eisen.

Dit document beschrijft de eisen en wensen die er vast zijn gesteld voor deze mobiele applicatie.

Totstandkoming

Met vertegenwoordiging van de doelgroep (medewerkers vanuit Van OERS United en Universal Greenfields) zijn er interviews gehouden. Tijdens deze interviews is er gekeken naar de huidige werkwijze en de toekomstige werkwijze wanneer de mobiele applicatie operationeel is. Met deze interviews zijn ankerpunten en verschilpunten inzichtelijk geworden. Voor het vastleggen van informatie uit de interviews is in GoogleForms een vragenlijst opgesteld waarin de gegeven antwoorden zijn ingevoerd en vastgelegd zijn in een overzicht.

Aan de hand van de verkregen informatie zijn er een User Persona, Empathy Map en User Journey uitgewerkt en toegevoegd aan de Doelgroep Analyse. Deze zullen tijdens de verdere ontwikkelingen regelmatig geraadpleegd worden om zo de belangen van de doelgroep niet uit het oog te verliezen.

De gezamenlijke ambitie en gewilligheid om mee te werken aan nieuwe ontwikkelingen voor het werkveld spelen een grote rol in de vorming van dit Programma van Eisen.

MoSCoW lijst

De MoSCoW methode is een efficiënte manier om de vereisten de prioriteren en zo een duidelijk scheidingslijn te creëren voor de minimale eisen waaraan de toekomstige applicatie aan moet voldoen.

Het woord MoSCoW is onder te verdelen in 4 categorieen: Must-haves, Should-haves, Could-haves en Will Not- Would Like of Wish (afhankelijk van de situatie).

Must haves

De mobiele applicatie moet de volgende functionaliteiten bezitten om minimaal inzetbaar te zijn:

Qr scanner
De applicatie moet toegang hebben tot de camera van de mobiele telefoon. Hiermee kan de QR-code op het productlabel van de sensormodule worden gescand. Uit de QR-code zal een unieke indentifier worden opgehaald waarmee de applicatie de specifieke sensormodule wordt herkend.

Back-end koppeling
De applicatie moet d.m.v. een internet- of dataverbinding toegang hebben tot de back-end van het Quantified web portaal. Daar zal de applicatie aan de hand van de eerder beschreven indentifier de sensordata van de bijhorende sensormodule ophalen en weergeven.

In de huidige situatie hebben gebruikers een account op insight waaraan de sensormodules gekoppeld staan. Het is essentieel dat de applicatie gebruik maakt van deze accounts om privacy en data integriteit te kunnen waarborgen.

Data visualisatie
Na het ophalen van de sensordata moet de applicatie dit op een overzichtelijke manier visualiseren, in de vorm van Graphs, Heatmaps en Geomaps.

Should haves
Dit zijn de functionele toevoegingen die niet cruciaal zijn, maar wel een waardevolle bijdrage leveren:

Alarmen instellen
De gebruiker moet onder- en overschrijdingsalarmen kunnen instellen in de app.

Push notificaties
Bij het afwijken van de verwachte ingestelde alarmen moet de app een notificatie geven zodat de gebruiker aangespoord wordt om actie te ondernemen.

Visueel afzonderlijke meetpunten
Het visueel mogelijk maken van het bekijken van waardes van afzonderlijke meetpunten in een grafiek.

Overlapping van data in een grafiek
Bij het meten van meerdere datavarianten kan de mogelijkheid om dit gezamenlijk in 1 grafiek te weergeven een handige toepassing zijn.

Could haves
Dit zijn de functionele toevoegingen die prettig zijn om te hebben, maar geen grote impact hebben:

Bruikbaarheid op andere apparaten
De mobiele applicatie wordt ontwikkeld met het principe: “Mobile first”, wat inhoudt dat de focus ligt op het gebruik met een mobiele telefoon of tablet. De mogelijkheid om de applicatie ook op een desktop of laptop te kunnen gebruiken kan een handige toepassing zijn.

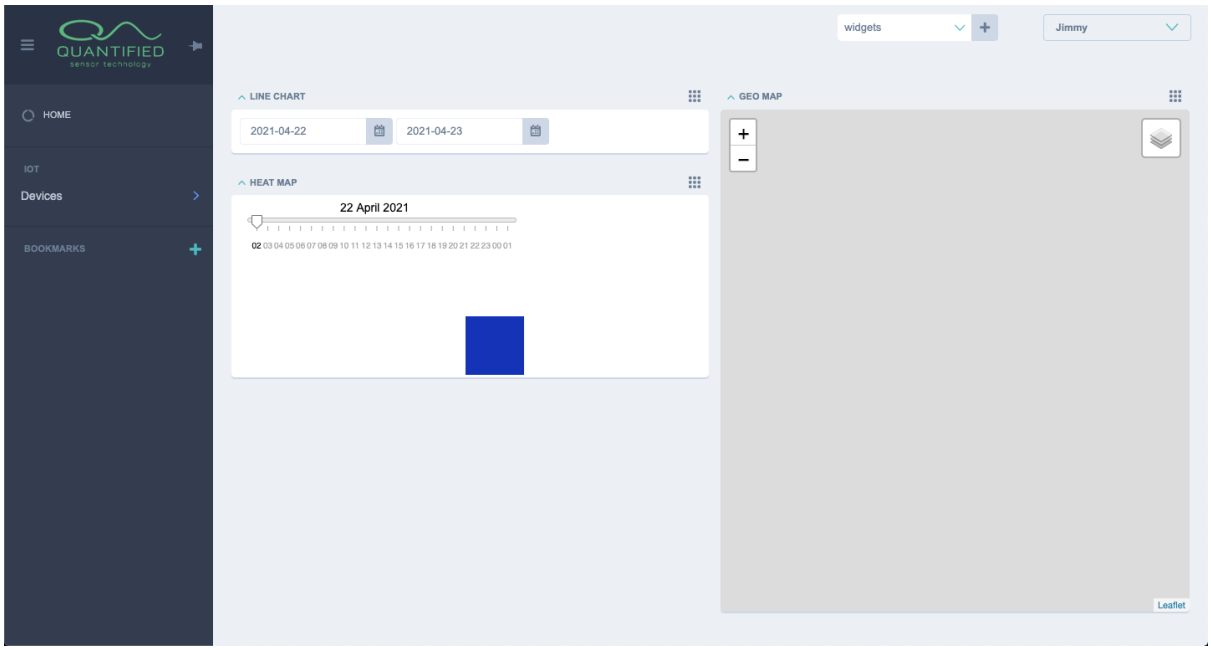
Will Not
Dit zijn de functionele toevoegingen waar geen prioriteit aan wordt gegeven binnen dit project:

Integratie met derden partijen
De mogelijkheid om meetgegevens van derden in te laden en weer te geven op eenzelfde manier als er met de Quantified verkregen data gedaan wordt. Dit geldt voor zowel het koppelen van externe sensoren als het ophalen van data via een externe API.

Prototyping

Web-portaal

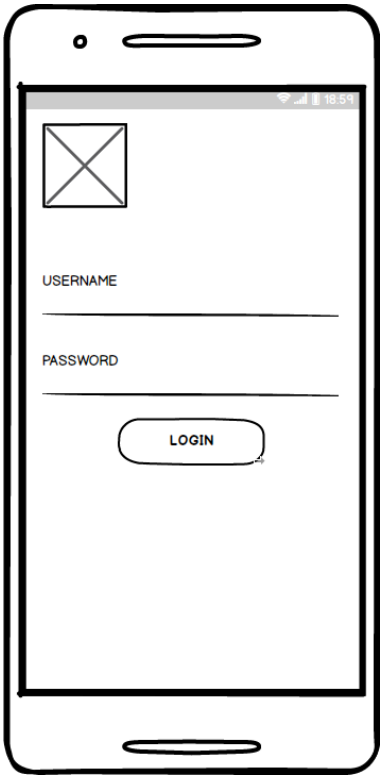
Quantified heeft een web-portaal genaamd Insight, waartoe gebruikers toegang krijgen zodra zij sensormodules van de organisatie in gebruik nemen. Insight biedt de gebruikers de mogelijkheid om zelf een dashboard te bouwen en zo inzicht te krijgen tot data vanuit metingen. Hier dient de gebruiker zelf een dashboard te bouwen aan de hand van beschikbare widgets, er is een handleiding beschikbaar hoe dit verder in zijn werk gaat.



Het web-portaal haalt sensordata op vanuit de back-end, mits een gebruiker daar toegang tot heeft, en laad dit vervolgens in het dashboard wat er gebouwd is. Het web-portaal wil ik verder niet te diep op ingaan omdat het niet moet afleiden van de afstudeeropdracht. Het portaal bestaat en gebruikers kunnen er gebruik van maken.

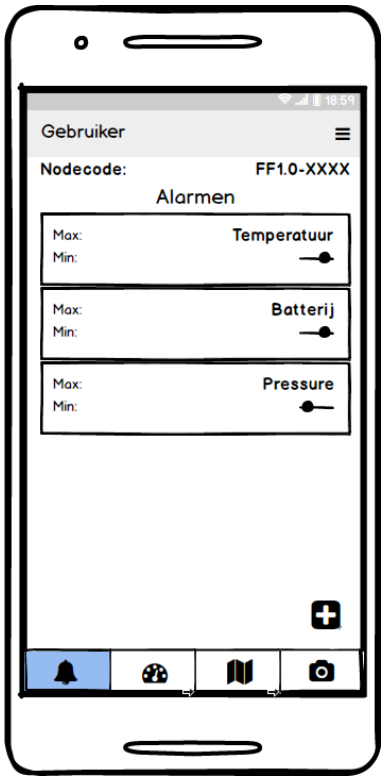
Lo-Fi Wireframes

Login screen



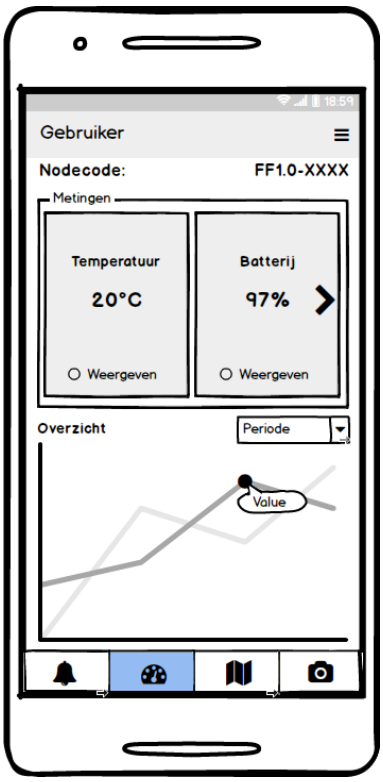
Het loginscherm is erg standaard, niet te veel ingewikkelde opties of verwachtingen. Dit is het eerste scherm wat men ziet bij het opstarten van de app en moet niet meteen de gebruiker afschrikken.

Alarmscreen



Het alarmscherm wordt gebruikt om alarmen op in te stellen. Als er een bepaalde meetwaarde onder- of overschreden wordt moet de gebruiker hier notificatie van krijgen.

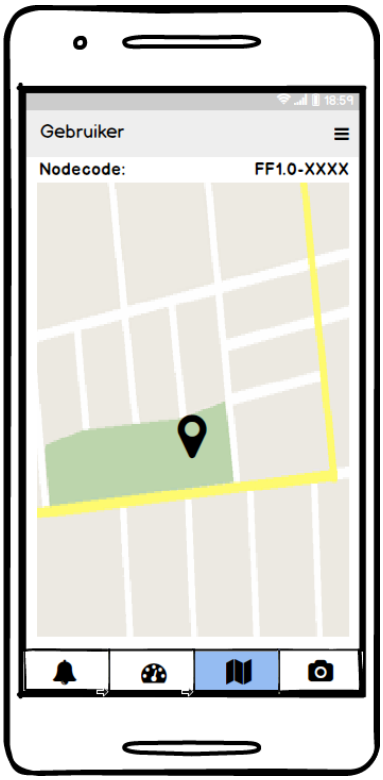
Homescreen (of Dashboard)



Dit is het scherm waarop de gebruiker data binnen krijgt en in een oogopslag kan zien wat de status van specifieke sensoren is.

De onderste helft van het scherm wordt beslagen door een grafiek, waarbij er gekeken kan worden naar een periode meeting (dag, week, maand, jaar) en kunnen meetpunten aangetikt worden om zo hun waardes te zien.

Mapscreen



De mapscreen geeft de locatie van een sensormodule weer, door het gebruik van de gps-module in de sensor of d.m.v. ingevoerde coördinaten in het geval waar een module geen gps bevat.

Testen

Voor het testen van de Lo-Fi Wireframes heb ik gekozen voor een techniek genaamd *Clickable Wireframes*, en is niets meer dan het koppelen van specifieke punten van de verschillende wireframes zodat er doorheen genavigeert kan worden “als een werkende applicatie”. Hiervoor heb ik gebruikt gemaakt van het programma Balsemic Mockups 3, wat niet alleen een handige tools is voor het ontwerpen van Lo-Fi Wireframes, maar ook nog eens de gemaakt opzet mee exporteert vanuit het programma. Op deze manier heb ik een snelle en lichte oplossing die ik makkelijk kan delen en testen met de doelgroep.

Voor het testen is er gebruik gemaakt van Microsoft Teams, waarbij de sessies niet langer duurde dan maximaal 30 minuten. Ik gaf eerst een korte introductie naar wat er verwacht kan worden en hoe dit in zijn werking gaat, vervolgens gaf ik de gebruiker toegang tot de testlocatie. Eerst gaf ik de gebruiker een paar minuten om zelfstandig door het concept heen te lopen, waarbij ik lette op handelingen en reacties (in hoeverre dit mogelijk was op afstand). Vervolgens gaf ik de gebruiker een aantal taken uitvoeren, en na afloop besproken wij de ervaring en feedback die er ontstond.

De feedback heb ik verwerkt in een GoogleForm, dit is een makkelijke en snelle manier om gegevens te verzamelen en te verwerken. En ben hier de overlappende knelpunten gaan onderscheiden en uitschrijven.

Knelpunten

Design

No.	Scherm	Knelpunt	Impact
1	Login	-	-
2	Home	Icoon is lijkt op een snelheidsmeter, geen home of dashboard	3
3	Alarm	De “Add” knop is niet consistent, is de ene keer vierkant en de andere keer rond.	3
4	Map	De kaar neemt het hele scherm in beslag, misschien kan je dat beter verkleinen en de ruimte voor iets anders gebruiken.	8

Usability

No.	Scherm	Knelpunt	Impact
1	Login	De optie om login te onthouden	8
2		De optie voor wachtwoord reset	6
3	Home	De mogelijkheid om dashboard aan te passen naar voorkeur	7
4		De optie om grafiek-type aan te passen	8
5	Alarm	De optie om een alarm tijdens het toevoegen al aan of uit te zetten	5
6	Map	Geen optie om te schakelen tussen verschillende sensormodules	8

Onderzoek Mobile application Frameworks en App Types

Introductie

De mobiele telefoon is vrijwel niet meer weg te denken uit het beeld van het dagelijks leven. Blair (2021) schrijft dat er in 2021 wereldwijd zo'n 3.2 biljoen smartphone-gebruikers zijn die hun apparaat voor allerlei verschillende doeleinden gebruiken. Daarnaast zijn er in 2021 wereldwijd zo'n 1.14 biljoen tablet-gebruikers.

De gemiddelde smartphone gebruiker besteed per dag gemiddeld zo'n 3 uur en 10 minuten aan hun apparaat, waarvan zo'n 2 uur en 51 minuten aan apps besteed wordt. Dit vertaalt naar 90% van de totale tijd wat er aan verschillende apps besteed wordt. Blair (2021) verwacht dan ook dat deze trends zichzelf in een stijgende lijn zullen voort blijven zetten.

Deze berichten geven veel bedrijven dan ook een stimulerende duw in de richting van het ontwikkelen van eigen Apps voor de mobiele markt. Er is momenteel een brede selectie van frameworks beschikbaar waarmee er Apps ontwikkeld kunnen worden, wat het tegelijkertijd ook lastig maakt om een passend framework te kiezen.

Het doel van dit onderzoek is om de verschillende app types en frameworks in perspectief te stellen, waarbij er gekeken wordt naar:

- App types:
  - Overzicht van bestaande app types;
  - Wat zijn momenteel de meest gebruikte frameworks;
  - Overeenkomsten en verschillen tussen frameworks;

- Frameworks:
  - Overzicht van bestaande frameworks;
  - Wat zijn momenteel de meest gebruikte frameworks;
  - Overeenkomsten en verschillen tussen frameworks;

Scope

De scope van dit onderzoek is het beschrijven en vergelijken van de verschillend app types en frameworks, dit zal in een overzicht worden uitgewerkt. Het doel van dit onderzoek is het in-/overzichtelijk maken van de beschikbare opties, de meest gebruikte opties en de toepassingsmogelijkheden op het moment van schrijven.

Dit overzicht zal Quantified ondersteunen bij het maken van de beslissingen die nodig zijn voor de verdere uitvoer van mijn afstudeeropdracht en zal ook een aantal deelvragen zoals beschreven in mijn afstudeerplan beantwoorden.

Methodiek

Dit onderzoek is van vergelijkende aard, wat betekent dat er twee of meer objecten ter vergelijking naast elkaar worden gehouden. Voor elk object zullen de voor- en nadelen tegenover elkaar worden gezet, dit creëert twee overzichten voor beide onderwerpen waarbij het in een oogopslag duidelijk zal zijn hoe de verschillende opties tegen elkaar afwegen.

Overzicht

App types

Bij het ontwikkelen van een mobiele applicatie zal er gekeken moeten worden naar wat voor type mobiele applicatie het wordt. Het type is bepalend voor de manier waarop de applicatie of service beschikbaar is voor gebruikers.

Native Apps

Native apps zijn gebouwd voor gebruik met een specifiek operating system (OS). Hierdoor is het niet mogelijk om een Android of iOS specifieke App op een ander platform te draaien.

Vanwege de focus op een enkel platform hebben native apps het voordeel sneller en betrouwbarder te presteren dan de andere opties. Er wordt efficiënter omgegaan met de device resources zoals geheugen en opslag, zijn geoptimaliseerd voor het User Interface van het apparaat en hebben een betere integratie met device features zoals contacten, camera, Bluetooth etc.

Het nadeel ligt in het ontwikkelen voor verschillende platformen omdat de geschreven code niet hergebruikt kan worden. Daarnaast moeten updates constant gedownload en geïnstalleerd worden door de gebruiker, wat kostbare opslag opvult.

Hieronder worden de meest bekende mobiele operating systems aangekaart.

Windows Phone

Dit is het operating system ontwikkeld door Microsoft en werkt exclusief op hun eigen Windows Phones (WP). Apps werden geschreven in C en C++. Hoewel de telefoons nog wel in omloop zijn is dit OS sinds 2017 niet meer in gebruik.

IOS

Dit is het operating system ontwikkeld door Apple en werkt exclusief op hun eigen mobiele producten zoals de iPhone en iPad. Apps worden geschreven in Objective-C of Swift.

Android

Dit is het operating system ontwikkeld door Google en de Open Handset Alliance en is terug te vinden op vrijwel elke Android smartphone op de markt. Android is gratis en open-source en zodoende makkelijk benaderbaar. Android apps kunnen in verschillende talen worden geschreven, maar voornamelijk in Java, C, C++.

Web-based Apps

Web based apps zijn gebouwd om te werken in verschillende internetbrowsers. De apps worden apps worden geschreven in HTML/CSS/JavaScript waardoor het mogelijk is om op verschillende platformen (ook los van de mobiele telefoon) te draaien zolang er een internetbrowser en verbinding aanwezig zijn om de achterliggende server te bereiken.

Het voordeel zit hier vooral in de onafhankelijkheid van specifieke operating systems. Daarnaast hoeft er niets gedownload te worden wat device resources scheelt, kan onderhoudt ten alle tijden worden uitgevoerd en updates live over het web gepusht worden.

Het nadeel zit hem in de afhankelijkheid van een internetbrowser en een stabiele internetverbinding. De gebruikerservaring kan variëren per internetbrowser omdat niet alle functionaliteiten overal beschikbaar zijn.

Hybrid Apps

Hybrid apps (of Cross-Platform apps) zijn gebouwd met de focus op het gebruik met meerdere platformen en gedragen zich vrijwel gelijk aan Native Apps. Hybrid apps zijn websiteapplicaties omwikkeld met een wrapper, wat ervoor zorgt dat de applicatie op meerdere platformen kan functioneren. Hybrid-apps functioneren volgens het *“write-once-run-everywhere”* principe.

Het bouwen van Hybrid-apps is snel en zuinig proces. Geschreven code kan makkelijk hergebruikt op andere operating systems, de app hebben lage laadtijden en functioneren ook prima met een langzamere internetverbinding.

Het nadeel van Hybrid-apps is dat ze minder krachtig en snel zijn, wat voor native apps toch wel kenmerkend is.

Hier zijn twee mogelijk varianten voor.

Web-Hybrid App Development

Alle operating systems hebben een web view, de mogelijkheid om web content weer te geven. Web-Hybrid apps worden geschreven in HTML/CSS/JavaScript wat de app bruikbaar maakt met elk mobiel operating system. Deze apps hebben een native wrapper waarmee ze meer toegangsmogelijkheden hebben tot de hardware van het apparaat dan de web-based app.

Native-Hybrid App Development

Native-Hybrid apps worden geschreven in meer voorkomende programmeertalen waarna het wordt geconverteerd naar een uitvoerbaar formaat bruikbaar met elk mobiel operating system. Dit soort app zal de gebruiker door de *“look and feel”* de indruk wekken dat er met een native app gewerkt wordt.

Opsomming

Dit overzicht bevat een opsomming van de app types in combinatie met een aantal essentiële parameters.

Parameters	Native	Web-based	Hybrid
TECHNOLOGY USED	Java, Kotlin, Python, Swift, Objective-C, C++, React.	HTML, CSS, JavaScript.	Ionic, Objective-C, Swift, HTML en anderen.
DEVELOPMENT COSTS	**Hoog** vergeleken met de rest, als er voor meerdere platformen wordt ontwikkeld.	**Laagste** kosten door beperkt code gebruik.	**Lage** kosten, vereist meer skills voor Hybrid tools.
DEVELOPMENT TIME	**Hoog.** Elk platform vereist aparte code. Brede beschikbaarheid van libraries voor de meeste functionaliteiten maar veel logica zal vanaf scratch geschreven moeten worden.	**Laag.** Draait met minimale aanpassingen in verschillende browsers.	**Gemiddeld.** Het merendeel van de code werk op meerdere platformen. De kwaliteit is erg goed door betere compile-time checks, debugging en profiling tools.
PERFORMANCE	**Optimaal.** Heeft een geoptimaliseerde integratie met de functionaliteit van het apparaat, de inhoud, structuur en visuele elementen worden lokaal opgeslagen.	**Minimaal.** Performance is afhankelijk van de browser en netwerkverbinding. Kost meer geheugen en CPU.	**Minimaal/Goed.** Vergelijkbaar met Native. De app bevat slechts de wrapper voor het apparaat, de meeste data ingeladen wordt vanaf een server.
BUILD SIZE	Veelal de meest gestroomlijnde broncode.	Is afhankelijk van de grote en complexiteit van de applicatie.	Onoverzichtelijker vergeleken met de native app, dit varieert per framework.
DEVICE FEATURES	**Optimaal.** Native platform code heeft een brede toegang tot device API's.	**Minimaal.** Web code Is beperkt in toegang tot device API's	**Minimaal/Goed.** Veel API's werken vergelijkbaar als Native, sommige zijn beperkt zoals bij Web-based.
USER INTERFACE	Apps worden voorzien van zeer herkenbare en originele UI ontwerpen voor het platform.	Zal geen volledige native ervaring bieden door andere design guidelines, maar kan wel gericht ontworpen worden.	Zal geen volledige native ervaring bieden door andere design guidelines, maar kan wel gericht ontworpen worden.

MAINTENANCE	**Hoog.** Hoger dan bij de andere, vooral afhankelijk van het aantal platformen waarvoor er ontwikkeld wordt.	**Laag.** Zolang er met 1 codebase is die onderhouden wordt, zijn de acties veel eenvoudiger en sneller	**Laag.** Zolang er met 1 codebase is die onderhouden wordt, zijn de acties veel eenvoudiger en sneller
UPDATES	**Hoge impact.** Elke update moet gedownload worden	**Lage impact.** Updates kunnen ten alle tijden over he netwerk gepusht worden.	**Lage impact.** Updates kunnen ten alle tijden over he netwerk gepusht worden.
CODE PORTABILITY	Code kan lastig hergebruikt worden voor andere platformen.	Draait met minimale aanpassingen in verschillende browsers.	De meeste Hybrid codebase tools kunnen overgebracht worden naar de grote platformen.
DISTRIBUTION	App stores brengen een aantal voordelen zoals ranglijsten, feature indeling.	Geen app store restricties, maar ook geen voordelen.	App stores brengen een aantal voordelen zoals ranglijsten, feature indeling.

Frameworks

Een framework is een platform waarop software ontwikkeld kan worden. Momenteel is er een brede selectie aan platformen beschikbaar waarmee verschillende applicaties gebouwd kunnen worden, en hierdoor lastig om te weten wat voor een specifiek project de beste optie is. Omdat dit onderzoek zich richt op de ontwikkeling voor mobiele applicaties, wordt er ook specifiek gekeken naar frameworks die hierop aansluiten.

Om te voorkomen dat hier een oneindige lijst aan frameworks beschreven gaat worden is er gekozen recente trends in mobile app development aan te houden als filter. Dit geldt met name voor de Hybrid frameworks.

Ter analyse gebruik ik per framework een Pros en Cons lijst, waarbij ik waardes op een schaal van 0 tot 10 toeken aan de factoren in de lijst. Deze waardes geven aan hoe belangrijk elke factor is en hoe zwaar deze meeweegt bij de beslissing vorming.

Native frameworks

iOS - Swift

Swift is het door Apple ontworpen framework voor iOS, iPad, macOS, watchOS, tvOS en Linux applicaties.

PROS	37	17	CONS
List	Value (0-10)		List
Tijd-efficiënt, snel programmeerbaar en makkelijk te lezen	8	6	**Populariteit,** niet de populairste keuze onder developers.
Ontwikkeling, apps zijn makkelijk uit te breiden (met nieuwe features) en overdraagbaar	8	5	**Slechte ondersteuning**, voor oudere iOS varianten;
Onderhoud, door de manier waarop men programmeert in Swift is het traceren van bugs en obstakels een snel en vrijwel pijnloos proces.	8	6	**Slechte interoperabiliteit** met derde partijen en IDE's.
Performance, geoptimaliseerd voor eigen platform	6		
Sterke beveiligingsmaatregelen	7		

Android – Java en Kotlin

Java

Java is een veelgebruikte programmeertaal. Java is class-based, object-georiënteerd en ontworpen om zo min mogelijk implementatieafhankelijkheden te hebben.

PROS	37	42	CONS
List	Value (0-10)		List
Multiplatform, werk op elk apparaat, server of operating system.	8	7	**Redelijk complexe syntax** vergeleken met bijvoorbeeld Python of C++.
Sterke beveiligingsmaatregelen, voorkomt geheugen corruptie.	7	7	**Niet alle inhoud is toegankelijk voor de gebruiker** omdat het mogelijk niet geschikt is voor het apparaat.
Modulaire applicaties zijn makkelijk te maken door de herbruikbaarheid van programmacode	8	6	**Toegankelijkheid** van vorderingen op mobile development is lastig.
Werkt goed samen met derde partijen	7	8	Heeft **bekende issues** met API-design in Android.
Is vrij makkelijk te hanteren	7	8	**Ontwikkeling gaat langzaam** doordat er vaak getest en debugged moet worden.
		6	**Presteert langzamer** dan andere talen en kost flink wat geheugen.

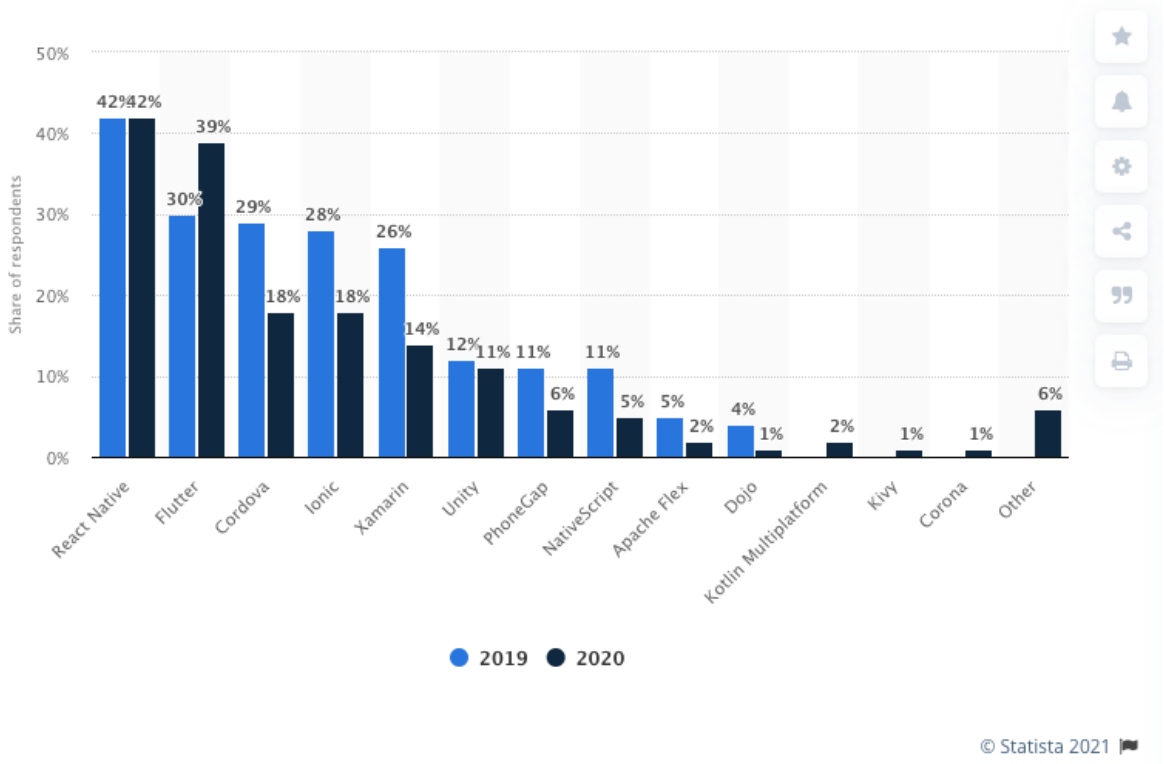
Kotlin

Kotlin is sinds 2019 het voorkeusplatform voor het bouwen van Android-applicaties. Kotlin is ontworpen met volledige integratie van Java, de JVM-versie van Kotlin's standaard libraries en is afhankelijk van de Java Class Library.

PROS	42	13	CONS
List	Value (0-10)		List
Tijd-efficiënt, snel programmeerbaar en makkelijk te lezen.	8	5	**Jonge taal,** daardoor een kleinere user-base, support- en developer community.
Beperkte applicatie grote, bespaart lokale opslag.	6	8	**Steile learning curve** door strikte syntax.
Compactere code, minder bugs en snellere debugging.	7		
Kan JVM libraries uitvoeren, wat Java framworks en libraries bruikbaar maakt binnen Kotlin.	8		
Geoptimaliseerd voor ontwikkeling in Android Studio.	6		
Goede **integratie met JavaScript**.	7		

Web-based / Hybrid frameworks

De basis voor web-based applicatieontwikkeling is HTML, CSS en JavaScript, waarbij een developer zich vervolgens kan gaan verdiepen (of specialiseren) in een brede selectie aan overlappende frameworks. Deze frameworks worden voor zowel web-based als Hybrid applicaties toegepast, zodoende wordt er alleen ingegaan op de meest gebruikte frameworks (op het moment van schrijven).



Cross-platform ontwikkelen is een populaire trend onder software developers. Zo blijkt uit het onderzoek: *"Cross-platform mobile frameworks used by developers worldwide 2019 and 2020"* dat de 5 meest gebruikte frameworks in juni 2020 React Native, Flutter, Cordova, Ionic en Xamarin waren. Daarnaast is het bekend dat binnen Quantified de meeste programmeurs bekend zijn met Python, waardoor dit ook een optie zou kunnen zijn als uiteindelijk framework om een applicatie mee te bouwen. Hieronder worden deze 6 frameworks benoemd en vergeleken.

React

React Native is een open-source framework ontworpen door Facebook, Instagram en een grote community van developers. Het maakt gebruik van JavaScript en JSX (JavaScript-XML) om "native" apps voor iOS en Android te maken. Daarnaast is er ook de ReactJS variant waarmee er web-based app's gebouwd kunnen worden.

PROS	**38**	**26**	**CONS**
List	Value (0-10)		List
Ontwikkelsnelheid en kosten, met ervaring werkt het makkelijk en kan het makkelijk overgedragen worden	8	6	**Performance** scoort lager dan volledig native apps
Write-Once, Deploy-anywhere, werkt op iOS, Android, MacOS en Windows	8	7	**Design-wise** niet het meest efficiënt.
User Experience, makkelijk om high-performance apps met geweldige UX te bouwen	8	6	**Custom modules,** ondanks de vele libraries kan het voorkomen dat voor een specifieke toepassing er zelf componenten geschreven moeten worden

Hot-Reloading, aanpassingen en updates worden direct doorgevoerd zonder de app te hoeven herstarten of te herbouwen	6	7	**Updating Issues,** het is lastig om de app constant up-to-date te houden met de laatste React Native versie.
Onderhoud, snel en eenvoudig want er wordt met 1 codebase gewerkt.	8		

Flutter

De Flutter website introduceert zichzelf als volgt: *"Flutter is Google's UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase."*

PROS	47	22	CONS
List	Value (0-10)		List
Ontwikkelsnelheid en kosten, met ervaring werkt het makkelijk en kan het makkelijk overgedragen worden	8	8	**Native Look and Feel**, lastig om af te stemmen op Google's Material Design principes voor Android en Apple's Design Systeem voor iOS
Write-Once, Deploy-anywhere, werkt op iOS, Android, MacOS en Windows	8	6	**Jonge technologie**, Flutter heeft een goede ondersteuning van Google maar kan daardoor ook in een ongewenste richting geduwd worden;
Adapt to Screens, Flutter gaat gemakkelijk om met verschillende schermafmetingen;	6	8	**Updates voor iOS- en Android-features**, developers zijn afhankelijk van geïntroduceerde toepassingen. De kans is groot dat men een feature zelf ontwikkeld die vervolgens later in een update wordt toegevoegd.
Hot-Reloading, aanpassingen en updates worden direct doorgevoerd zonder de app te hoeven herstarten of te herbouwen	7		
Onderhoud, snel en eenvoudig want er wordt met 1 codebase gewerkt.	8		
Leverage Device Hardware, Flutter is in staat om apparaat functies volledig te benutten zonder enige vertraging.	8		

Cordova

Voorheen bekent als PhoneGap, is Cordova een gratis en open-source framework voor het ontwikkelen van mobiele applicaties. Het creëert en cross-platform applicatie door gebruik te maken van HTML, CSS en JavaScript.

PROS	24	21	CONS
List	Value (0-10)		List
Ontwikkelsnelheid en kosten, met ervaring werkt het makkelijk en kan het makkelijk overgedragen worden.	8	6	**Performance**, door cross-platform mogelijkheden werkt de Hybrid app langzamer.
Write-Once, Deploy-anywhere, werkt op iOS, Android, MacOS en Windows.	8	7	**Compatibiliteit issues**, met verschillende plug-ins verspreid over verschillende platformen en apparaten.
Onderhoud (en Updates), snel en eenvoudig want er wordt met 1 codebase gewerkt.	8	8	**Plug-ins** functioneren mogelijk niet zoals verwacht en zullen daardoor gemodificeerd moeten worden.

Ionic

Een open source mobiele UI-toolkit voor het bouwen van hoogwaardige, platformonafhankelijke native en webapp-ervaringen.

PROS	49	32	CONS
List	Value (0-10)		List
Write-Once, Deploy-anywhere, werkt op iOS, Android, MacOS en Windows.	8	6	**Performance**, door cross-platform mogelijkheden werkt de Hybrid app langzamer.
Populaire technologie en makkelijk te leren.	8	7	**Plugin afhankelijk**, Ionic is niet in staat om zowel web-based als native simultaan uit te voeren zonder native plug-ins om te vormen in JavaScript.
Sterke community.	6	5	**GEEN hot reloading**,is een populaire en zeer behulpzame feature.
Brede integratie van mogelijkheden en plug-ins.	7	7	**Security Issues**, mogelijke valkuil waarbij developers zelf de code lastiger toegankelijk moeten maken voor buitenstaaners.
Ruimte selectie aan UI elementen en mogelijkheid voor **snelle prototypes**.	7	7	**Applicatie grote**, door het samenvoegen van HTML, CSS, JavaScript, libraries, plug-ins, dependencies en andere benodigdheden nemen applicaties snel toe in grote.
Integratie met frontend frameworks zoals Angular, React, Vue en (vanilla) JavaScript.	6		

Onderhoud (en updates), snel en eenvoudig want er wordt met 1 codebase gewerkt.	8		

Xamarin

Xamarin is een open-source platform voor het bouwen van moderne en goed presterende applicaties voor iOS, Android en Windows met .NET. Xamarin gebruikt enkel C# als programmeertaal.

PROS	44	32	CONS
List	Value (0-10)		List
Write-Once, Deploy-anywhere, werkt op iOS, Android, MacOS en Windows.	8	6	**Vertraagde support** voor platform updates.
Performance, dichterbij Native dan andere cross-platform oplossingen.	6	7	**Gelimiteerd toegang** tot Open Source Libraries.
Native UX, door ondersteuning van libraries en frameworks.	8	8	Professional en Enterprise gebruik is prijzig.
Full Hardware Support, volledige toegang tot apparaat funcies.	8	6	**Compiling en Deployment** naar iOS vereist momenteel een apparaat voorzien van MacOS.
Onderhoud en updates, snel en eenvoudig want er wordt met 1 codebase gewerkt.	8	5	**Applicatie grote**, afhankelijk van het type en de complexiteit.
Hot-Reloading, aanpassingen en updates worden direct doorgevoerd zonder de app te hoeven herstarten of te herbouwen.	6		

Python

Bij het bouwen van een mobiele applicatie is Python niet bepaald het eerste framework wat meteen ter sprake komt, maar dat betekent niet dat het onmogelijk is om te gebruiken. Met behulp van de juiste libraries (Kivy, PyQt of BeeWare) is het mogelijk om een cross-platform mobiele applicatie te bouwen.

PROS	28	23	CONS
List	Value (0-10)		List
Simpele en duidelijke syntax, geen verwarrende brackets etc.	8	8	**Ontwikkelsnelheid en kosten,** het ontwikkeltraject met Python Mobile loopt heel snel op qua tijd.
Automatische geheugen toewijzing.	6	8	Wordt **langzaam en onpraktisch** bij grotere en complexere applicaties.
Actieve community.	6	7	Voor bepaalde taken kan **impliciete geheugentoewijzing** nadelig zijn.
Veel ondersteunende libraries.	8		

AWS (Amazon Web Services)

Quantified maakt gebruik van het AWS-platform. Volgens de website is AWS Amplify de *"Fastest, easiest way to build mobile and web-based apps that scale".* Dit platform ondersteunt een overlap van populaire web-based frameworks en mobile platforms waarvan een groot deel beschreven is in dit onderzoek.

Web-based frameworks:

- JavaScript;
- React
- Angular
- Vue
- Next.js

Mobile Platforms:

- Android;
- iOS;
- React Native;
- Ionic;
- Flutter;

Bevindingen

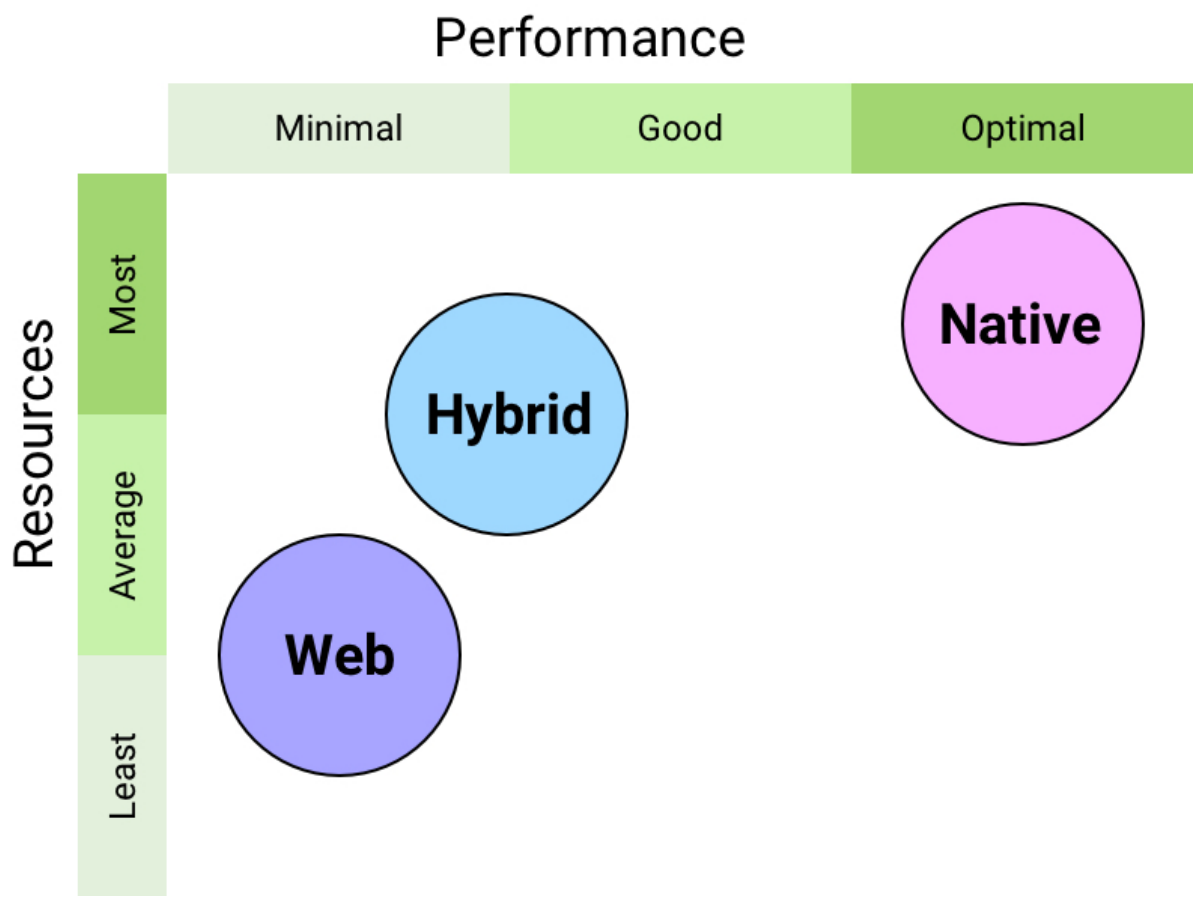
App types

Bij het ontwikkelen van een applicatie zal er bewust gekeken moeten worden naar wat er specifiek gemaakt gaat worden. Er zijn een aantal factoren die voor de ene soort app ideaal zijn maar niet voor een ander.

De belangrijkste beslissende factoren (gepaard met het afstudeerproject) zijn:

- Tijd;
- Resources;
- Performance.

Ter visualisatie heb ik de app types en hun Performance/Resource scores (benoemd in hoofdstuk 4, sub-hoofdstuk 4.1.4) uitgewerkt in de onderstaande afbeelding.



Zo wordt het al snel duidelijk dat:

- Web-based apps weinig resources kosten maar minimaal presteren;
- Hybrid apps meer resources kosten, maar ten aanzien van de web-based toepassing beter presteren;

Native apps de meeste resources kosten, maar ook de beste prestatie leveren.

5.1.1 Tijd

Voor een vlotte ontwikkeling en snelle publicatie wordt de Web-based app geadviseerd. De doorlooptijd van het ontwikkelen ligt laag en er worden snel resultaten behaald.

5.1.2 Resources

Bij een gebrek aan resources zoals tijd, geld of mensen worden de Web-based en Hybrid-app geadviseerd.

5.1.3 Performance

Als performance de grootste rol speelt wordt de Native app geadviseerd. Dit geeft de app de beste opties voor snelheid, stabiliteit en aanpassingsmogelijkheden.

Advies
De gemaakte keuze zal altijd moeten aansluiten op de eisen en wensen van de gebruiker, het is dan ook belangrijk dat deze vooraf, duidelijk in kaart worden gebracht. Als de applicatie op meerdere soorten devices gebruikt zal gaan worden is voor een breed bereik is de hybrid applicatie, met de focus op mobile development first, het meest praktisch.

Frameworks
Bij het selecteren van een framework zal er goed gekeken moeten worden naar wat voor soort applicatie er ontwikkeld gaat worden en wat de applicatie specifiek zal moeten gaan doen.

De belangrijkste beslissende factoren (gepaard met het afstudeerproject) zijn:
- **Beveiliging,** met name van data;
- **Ontwikkeltijd**, hoe snel is er een werkend product;
- **Onderhoud,** in hoeverre is dit makkelijk en snel te doen;
- **Ondersteuning van libraries**, voorkomt dat het wiel opnieuw moet worden uitgevonden;
- **User Experience**, de "*look and feel*" van de native app gepaard met de manier waarop het interacteert met de gebruiker;

Beveiliging
De applicatie zal werken met sensor data wat, wat gekoppeld is aan een gebruikers account bij de back-end van Quantified. De sensordata moet binnen de app alleen beschikbaar moeten zijn voor de gebruiker. Een framework waarbij er al rekening wordt gehouden met veiligheid bij het ontwikkelen is daarvoor ideaal.

Ontwikkeltijd
Voor het afstudeertraject is het van belang om binnen 20 weken een werken product op te leveren, wat wenselijk direct ingezet kan worden in de praktijk. Hierdoor speelt de gemiddelde ontwikkeltijd van applicatie op een specifiek platform een grote rol.

Onderhoud
De applicatie zal na uitrol onderhouden en geüpdatet moeten worden vanuit Quantified. Hierbij is het wenselijk het zo simpel mogelijk te houden. Een framework wat met meerdere codebases werkt is veel lastiger te onderhouden door de complexere opzet.

Ondersteuning van libraries
Libraries bestaan om het leven van developers over het algemeen makkelijker te maken. Libraries bevatten verschillende functionaliteiten die ingezet kunnen worden in de applicatie als het framework dit ondersteund. Dit komt de ontwikkeltijd weer ten goede.

User Experience

Het is niet alleen van belang dat de applicatie voelt alsof het gebouwd is voor het apparaat waarmee het gebruikt gaat worden, maar ook efficiënt interacteert met de gebruiker. Dit zorgt voor een prettige manier van werken met-, en duidelijkheid m.b.t. gebruik van de applicatie. Het doel is om de gebruiker zo min mogelijk te frustreren.

Advies

Aan de hand van de informatie uit hoofdstuk 5, sub-hoofdstuk 5.2, moet het duidelijk worden welke opties qua frameworks er voor dit project beschikbaar zijn. Er is duidelijk gemaakt dat Quantified de voorkeur geeft aan het ontwikkelen met Python, maar zoals duidelijk wordt komt niet zonder risico's en gebreken.

Het is ook verstandig te kijken naar de populariteit van een framework omdat dit de kans vergroot dat er meer aanbod van specifieke developers is. De groei, verhouding en populariteit van React ten aanzien van de andere opties mag dan ook zeker niet genegeerd worden.

Conclusie

In dit onderzoek is gezocht naar antwoorden voor de volgende deelvragen:

- Wat voor mobiele applicatie types zijn er
- Welk applicatie type past het beste binnen de opdracht
- Welke frameworks zijn er
- Welk framework past het beste binnen de opdracht

Hiervoor is een literatuuronderzoek uitgevoerd naar de huidige mobiele applicatie types, de verschillende frameworks en hun toepassingsmogelijkheden.

Uit de resultaten naar de mobiele applicatie types is gebleken dat er drie opties zijn, native, web-based en hybrid. De web-based applicatie type lijkt geen passende optie voor deze opdracht omdat het beperkt is in het gebruik van de verschillende hardware elementen van mobiele apparaten. Dit maakt de applicatie bruikbaar op elk apparaat met een webbrowser en een actieve netwerkverbinding, maar zal geen User Experience creëren die vergelijkbaar is met een native applicatie.

Het Native applicatie type staat sterk door de mate waarin het toegang heeft tot de hardware elementen van mobiele apparaten, maar ook op de manier waarop het een unieke en herkenbare User Experience zal geven voor een specifiek platform. Dit wordt ondersteund door de opgestelde design guidelines voor deze platformen. Het nadeel ligt echter in de compatibiliteit met meerdere platformen wat ervoor zorgt dat als Quantified in de toekomst de applicatie op andere platformen wilt aanbieden, hier een andere applicatie voor zal moeten worden ontwikkeld.

Het Hybrid Applicatie type staat sterk door de inzetbaarheid op meerdere platformen, maar is meer beperkt in toegankelijkheid met hardware elementen. Dit type werkt volgens het "*Write-Once, deploy-everywhere*" principe, waardoor cross-platform ondersteuning zeer gangbaar is en heeft de kortste doorlooptijd m.b.t. ontwikkeling.

Uit de voorafgaande doelgroep analyse is het duidelijk geworden dat het voorkeursplatform voor de eindgebruikers Google's Android betreft en met het oog op trends in de groeiende mobiele applicatie markt mag de populariteit van zowel het React framework, als van het Flutter framework zeker niet genegeerd worden.

Uit de resultaten blijkt dat beide frameworks voldoende mogelijkheden en ondersteuning bieden voor het creëren van een User Experience die vrijwel identiek is aan de Native frameworks. Daarbij komt het gemak van het werken met een enkele codebase en cross-platform compatibiliteit, waardoor het makkelijk is om ook voor andere mobiele platformen te ontwikkelen. En tot slot zal de applicatie hiermee ook het makkelijkst te onderhouden en te updaten zijn.

Literatuurlijst

Blankenvoort, E. (2015, 2 maart). *Doelgroepanalyse maken in 10 stappen!* Salland Communicatie. https://www.sallandcommunicatie.nl/blog/marketingcommunicatie/23-doelgroepanalyse-maken-in-10-stappen

Van Oers United. (z.d.). Van OERS United. https://www.vanoersunited.nl/

Universal Greenfields. (2020, 14 juli). *Specialist in groene daken en gevels*. https://www.universalgreenfields.nl/

Bijker, J. (2013). *Tien beinvloedingsvaardigheden* (3de editie). Thema.

Blair, I. (2021, 19 februari). *Mobile App Download and Usage Statistics (2021)*. BuildFire. https://buildfire.com/app-statistics/#

Valdellon, L. (2020, 2 november). *What Are the Different Types of Mobile Apps? And How Do You Choose?* CleverTap. https://clevertap.com/blog/types-of-mobile-apps/

Wittwer, J. (z.d.). *Pros and Cons List Template*. Vertex42.Com. https://www.vertex42.com/ExcelTemplates/pros-and-cons-list.html

Swift - Apple Developer. (z.d.). Apple Developer. https://developer.apple.com/swift/

Java Resources for Students, Hobbyists and More | go.Java | Oracle. (z.d.). Java Powers Our Digital Wold. https://go.java/?intcmp=gojava-banner-java-com

Kotlin and Android |. (z.d.). Android Developers. https://developer.android.com/kotlin

Liu, S. (2020, 2 juli). *Cross-platform mobile frameworks used by developers worldwide 2019 and 2020*. Statista. https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/

React Native · A framework for building native apps using React · React Native. (z.d.). React Native. https://reactnative.dev/

Marcak, M. (2021, 23 maart). *React Native Pros and Cons in 2021*. Pagepro Blog. https://pagepro.co/blog/react-native-pros-and-cons/

Flutter - Beautiful native apps in record time. (z.d.). Flutter. https://flutter.dev/?gclid=CjwKCAjwxuuCBhATEiwAIIIz0eFxLALjc-Gbo9Xyl_a8lHTdepYT3kPHXy-9vmp36FmWAqLavvqmMxoCTWkQAvD_BwE&gclsrc=aw.ds

Apache Cordova. (z.d.). CORDOVA. https://cordova.apache.org/

Ionic - Cross-Platform Mobile App Development. (z.d.). Ionic Framework. https://ionicframework.com/

Microsoft. (z.d.). *Xamarin | Open-source mobile app platform for .NET*. https://dotnet.microsoft.com/apps/xamarin

Python Mobile SIG. (z.d.). Python.Org. https://www.python.org/community/sigs/current/mobile-sig/