HO
GENT

Faculteit Bedrijf en Organisatie

Snow- en Surfboard komen boven water:
Een onderzoek en proof of concept naar een budgetvriendelijke GPS-tracker

Indy Van Canegem

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Jens Buysse
Co-promotor:
Thomas Pollet en Kevin DeRudder

Instelling: —

Academiejaar: 2019-2020

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Snow- en Surfboard komen boven water:
Een onderzoek en proof of concept naar een budgetvriendelijke GPS-tracker

Indy Van Canegem

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Jens Buysse
Co-promotor:
Thomas Pollet en Kevin DeRudder

Instelling: —

Academiejaar: 2019-2020

Tweede examenperiode

Woord vooraf

Eerst en vooral wil ik mijn familie bedanken voor alle steun tijdens mijn hele opleiding. De studie was zeker en vast niet makkelijk, maar ze stonden altijd paraat wanneer ik ze nodig had. Hiervoor, een welgemeende dankjewel.

Ook wil ik mijn promotor, Jens Buysse, bedanken voor het aanreiken van het interessante onderwerp. Het onderwerp is zeer 'Internet of Things' gerelateerd, wat mooi aansluit bij mijn interesses en afstudeerrichting. Ikzelf ben ook een snowboarder en surfer, waardoor ik zelf ook het belang inzie van een economisch rendabele GPS-tracker. Ik hoop dat mijn werk gebruikt kan worden in de winter- en watersport.

Graag bedank ik ook mijn co-promotors Kevin DeRudder en Thomas Pollet voor de constructieve feedback.

Tijdens deze bachelorproef kon ik proeven van verschillende vakgebieden zoals het ontwerpen van een backend in Mongoose, het ontwikkelen van een webapplicatie in Vue.JS en een cross-platform applicatie. Ik heb mij deze vaardigheden tijdens de proef eigen gemaakt.

Verder hoop ik dat deze bacherlorproef een verrijking is voor de lezer en hem eventueel kan inspireren tot het ontwikkelen van loT devices.

Samenvatting

Water- en wintersporters hebben nood aan een gebruiksvriendelijke, waterdichte en betrouwbare GPS-tracker. Deze toestellen kunnen voor allerlei situaties gebruikt worden zoals het terugvinden van uitrusting en als lawinepieper. Vooralsnog zijn de bestaande toestellen vrij duur, waardoor de barrière voor water- en wintersporters hoger is om een dergelijk toestel aan te schaffen. Er dringt zich dus een onderzoek op naar de economische rendabiliteit van een GPS-toestel dat niet bezwijkt onder ongunstige omstandigheden. In dit onderzoek wordt onderzocht welke locatiebepalingstechnologieën er gebruikt kunnen worden en aan welke eisen de hardware moet voldoen. Hieruit blijkt dat de MKR GSM module in combinatie met de MKR GPS SHIELD de beste resultaten oplevert. Geschikte locatiebepalingstechnologieën zijn het Standard Positioning System (SPS) en General Packet Radio Service (GPRS). De proof of concept die ontwikkeld werd, is in staat om diens locatie te delen aan de hand van een zelf ontwikkelde webapplicatie. De proof of concept voldoet aan alle eisen die werden uiteengezet in de probleemstelling. De corona-pandemie heeft de initiële reikwijdte van het onderzoek enigszins beperkt. Het was door het verbod op niet-essentiële verplaatsingen niet mogelijk om de invloed van sneeuw en zout water te achterhalen. De andere onderzoeksvragen konden wel beantwoord worden aan de hand van een experimenteel design. Er kan geconcludeerd worden dat het mogelijk is om een economisch rendabel toestel te ontwikkelen dat niet bezwijkt onder ongunstige omstandigheden en voldoet aan alle kwaliteitsvereisten.

Inhoudsopgave

Lijst van figuren

Lijst van tabellen

Acroniemenlijst

GPS Global Positioning System
PoC Proof of concept
GNSS Global Navigation Satellite System
BDS BeiDou Navigation Satellite System
GLONASS Globalnaya navigatsionnaya sputnikovaya sistema
IRNSS Indian Regional Navigation Satellite System
QZSS Quasi-Zenith Satellite System
USAF United States Air Force
EU Europese Unie
SPS Standard Positioning System
PPS Precision Positioning System
DGPS Differential Global Positioning System
FLEPOS Flemish Positioning Service
P Precision code
WAAS Wide Area Augmentation System
TTFF Time To First Fix
AGPS/A-GPS/aGPS Assisted Global Positioning System
AML Advanced Mobile Location
gsm global system for mobile communications
sms Short Message Service
SSID Service Set Identifiers
MAC Media Access Control
GPRS General Packet Radio Service
RPI Raspberry Pi
MEVN Mongoose-Express-Vue-NodeJS

1. Inleiding

Veel onderzoek stelt dat de huidige tijd gekenmerkt wordt door een verstrekkende technologische revolutie. Deze revolutie heeft veel systemische veranderingen teweeg gebracht in de manier waarop mensen hun leefwereld structureren en betekenis geven. Eén van die nieuwe toepassingen is het Global Posititon System (GPS) dat mensen in staat stelt om zich waar dan ook op deze planeet te navigeren. Het wordt beschouwd als één van de meest verspreide ICT-toepassingen ter wereld. De opkomst van de GPS verloopt quasi simultaan met de verspreiding van de smartphone. Geschat wordt dat er inmiddels 3,5 miljard mensen zijn met een smartphone, hetgeen het alledaagse gebruik van de GPS in verregaande mate gefaciliteerd heeft (Holst, 2019)

Een specifieke toepassing van het GPS-systeem dat in deze paper aan bod komt, is de zogenaamde lawinepieper of avalanche transceiver. Deze avalanche transceiver worden gebruikt om mensen terug te vinden die bedolven worden door een lawine bij het wintersporten. Uniek aan deze systemen is dat ze opereren op een eigen frequentie (457kHz).

In deze paper zal onderzocht worden of er - naar analogie met de lawinepieper - een soortgelijke toepassing mogelijk is voor het opsporen van vermiste personen onder water. Dit is een uitdaging via de huidige beschikbare technologieën omdat ze weinig efficiënt zijn bij het opsporen van materiaal dat zich onder water bevindt (Gunnar Taraldsen, 2011). Satellieten ondervinden problemen bij het opvangen van elektromagnetische signalen onder het wateroppervlakte. Toestellen die hier wel in slagen, kosten ook veel geld. Vandaag de dag kost een goede lawinepieper minstens 195,63 euro (Ian Nicholson, 2018). Dit zorgt ervoor dat winter- en watersporters vaker bereid zijn om veiligheid in te ruilen voor een goedkopere uitoefening van hun hobby. Tot slot moet de casing van dergelijke gps-toestellen ook water- en weerbestendig zijn. Winter- en watersporten worden vaak uitgeoefend in uitdagende omstandigheden.

In het kader van dit onderzoek werd een proof of concept (PoC) gerealiseerd dat getest werd in ongunsitge omstandigheden. Tot slot werd ook een applicatie ontwikkeld om de tracker op te volgen.

1.1 Probleemstelling

Water- en wintersporters opteren vaak niet voor een GPS-tracker doordat deze toestellen duur, niet gebruiksvriendelijk en moeilijk te configureren zijn. GPS-trackers kunnen, als watersporter, gebruikt worden om verloren uitrusting terug te vinden. Een andere optie is dat de GPS-tracker gebruikt wordt als lawinepieper. GPS-trackers kunnen, afhankelijk van de situatie, gebruikt worden om levens te redden en verloren goederen te localiseren. Dit onderzoek focust zich op winter- en watersporters die nood hebben aan een betrouwbaar, goedkoop, waterdicht en gebruiksvriendelijke GPS-toestel.

1.2 Onderzoeksvraag

1.2.1 Hoofdonderzoeksvraag

Zoals reeds aangegeven zal dit onderzoek zich vooral focussen op het ontwerpen van een goedkoop GPS-toestel dat goed presteert in ongunstige omstandigheden. Hieruit volgt dan de volgende hoofdonderzoeksvraag:

> *Is het mogelijk om een economisch rendabel GPS-toestel te ontwikkelen dat niet bezwijkt onder ongunstige omstandigheden?*

1.2.2 Deelonderzoeksvragen

Als resultaat van dit onderzoek moet er een proof of concept (PoC) ontworpen worden dat voldoet aan de volgende vereisten:

- Is de proof of concept in staat om zijn locatie te delen?
- Is de proof of concept goedkoper dan een smartphone (200 euro)?
- Is de proof of concept zoutbestendig?
- Is de proof of concept waterbestendig?
- Kan de proof of concept functioneel blijven werken onder water en sneeuw?
- Kan de proof of concept minstens even accuraat werken als een ingebouwde GPS in een gsm?
- Kan de proof of concept getracked worden aan de hand van een webapplicatie?
- Is de webapplicatie gebruiksvriendelijk?

1.3 Onderzoeksdoelstelling

Het hoofddoel is een proof of concept ontwerpen dat toegankelijk is voor water- en wintersporters. Deze doelgroep heeft nood aan een GPS-systeem dat goed blijft functioneren in alle omstandigheden. Het onderzoek is geslaagd wanneer aan alle vereisten voldaan worden.

1.4 Opzet van deze bachelorproef

Het vervolg van deze bachelorproef is opgebouwd als volgt:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.

In Hoofdstuk 3 wordt een vergelijking gemaakt tussen verschillende toestellen/opties. De beste kandidaat is de proof of concept.

In Hoofdstuk 4 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 5 en Hoofdstuk 6 vindt u het volledig bekomen resultaat (proof of concept met bijhorende applicatie).

In Hoofdstuk 7, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen.

2. Stand van zaken

2.1 Global Positioning System (GPS)

Het Global Positioning System is een wereldwijd gekend systeem, maar toch zijn er heel wat verschillen. Als er gesproken wordt over GPS, wordt er impliciet NAVSTAR bedoeld.(Grimes, 2008a) Dit eerste operationele GPS-systeem had oorspronkelijk enkel militaire toepassingen. Het is ontwikkeld door de 'United States Air Force'. (National Coordination Office for Space-Based Positioning & Timing, 2020) Later is NAVSTAR toegankelijk gemaakt voor de gewone gebruiker. In deze sectie wordt toegelicht welke de verschillende systemen zijn en welke gebruikt kunnen worden.

2.1.1 De werking van GPS

Het doel van een GPS is om te navigeren. Daarvoor moet de locatie heel nauwkeurig bepaald worden. Hiervoor zijn drie zaken nodig:

- Grondstation, deze zijn ontworpen voor buitenplanetaire[1] draadloze communicatie. Ze communiceren door het ontvangen en verzenden van radiogolven met zeer hoge frequentie. (Zie figuur: 2.1)
- Een satelliet, of ook vaak kunstmaan genoemd, is een object dat zich in een baan om een hemellichaam bevindt. (Christiaens, g.d.) (Zie figuur: 2.2)
- GPS-ontvanger, dit ontvangt de elektromagnetische signalen van de satellieten.

De grondstations worden gebruikt om de locatie bij te houden van de GPS-satellieten. Deze

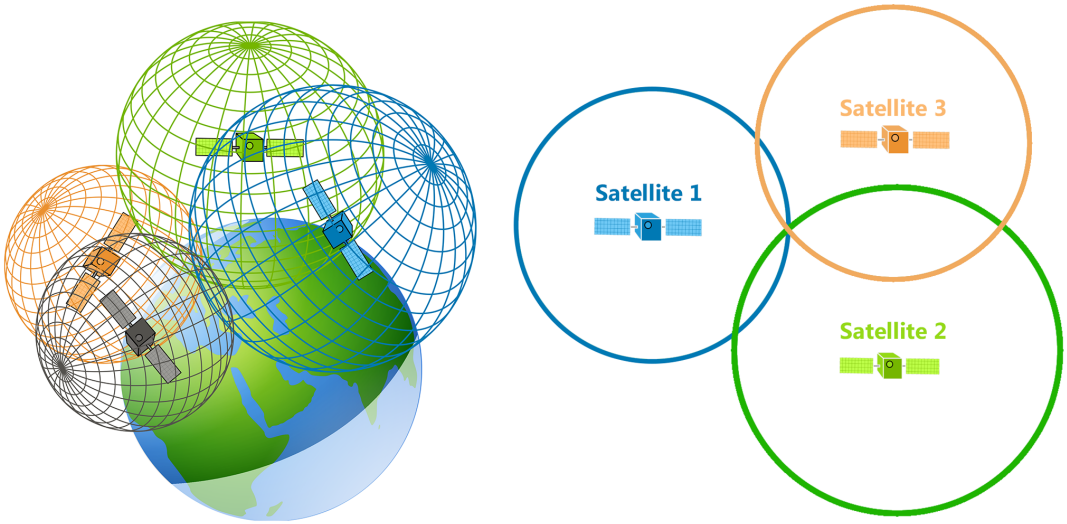
[1] De ruimte buiten onze planeet

Figuur 2.1: Voorbeeld grondstation (ESA, g.d.)

Figuur 2.2: Voorbeeld GPS-satelliet (SpaceNews, 2014)

stations worden niet gebruikt voor het bepalen van de huidige locatie van een gebruiker.

De GPS-satellieten zenden een signaal uit die de afstand en tijd bevat. Deze tijd wordt berekend met een klok aan boord. Een kwartskristalklok kan na 6 weken een volledige milliseconde afwijken van de werkelijke tijd door temperatuurschommelingen. Dit is gevaarlijk omdat het een grote impact heeft op het opmeten van posities bij snel bewegend ruimtevaartuig. Hierdoor maken satellieten gebruik van een ander type klok, de atoomklok. Atoomklokken zijn een 'verbetering' ten opzichte van de kwartskristal klok. Na 10 jaar wijkt een atoomklok slechts een microseconde af, wat stabieler is in vergelijking met de kwartskristal klok. Alles moet stabiel blijven voor ruimtevaartmisies, waardoor de afwijking van de atoomklok dagelijks gecorrigeerd wordt. (Samuelson, 2020)
Als de GPS-ontvanger de tijd en afstand uit het signaal heeft kunnen afleiden, kan het aan de hand van trilateratie zijn locatie bepalen. (Zie figuur: 2.3) Per satelliet wordt een radius bepaald. Hiervoor zijn minstens drie satellieten nodig. De overlapping van de radiussen bepalen één punt, de locatie van de GPS-ontvanger. Soms kan dit niet accuraat gebeuren, waarbij de radius van een vierde satelliet kan helpen. De laatste radius sluit dan foute mogelijkheden uit waardoor de locatie accurater is. Er is met andere woorden een positieve correlatie tussen het aantal satellieten dat gebruikt wordt en de nauwkeurigheid van de locatiebepaling. (Grimes, 2008a)



Figuur 2.3: Visualisatie hoe 4 GPS-satellieten een locatie vastleggen aan de hand van trilateratie. Het snijpunt van de 4 radiussen is de berekende locatie (gisgeography, 2016)

2.1.2 Global Navigation Satellite Systems (GNSS)

Als er gesproken wordt over het bepalen van een locatie, gebruiken we systematisch de term 'GPS'. (National Coordination Office for Space-Based Positioning & Timing, 2020) Dit is echter niet correct, want er bestaan veel meer technologieën en systemen die hierbij helpen. Deze technologieën worden besproken in dit hoofdstuk.
Het 'Global Positioning System' is een 'ruimte gebaseerd' navigatiesysteem dat eigendom is van de Verenigde Staten en bestuurd wordt door de 'United States Air Force' (USAF). Officieel heet dit systeem 'Navigation Satellite Time And Ranging' (NAVSTAR). De allereerste versie werd gelanceerd in 1978 en in 1995 werd het operationeel verklaard. GPS is in staat om een ongelimiteerd gebruikers met een GPS-ontvanger te voorzien van hun locatie op ieder moment van de dag, onafhankelijk van het weer en over de hele wereld. (Grimes, 2008a)
Naast NAVSTAR bestaat er ook:

- Galileo, beheerd door de Europese Unie (EU).
- Globalnaya navigatsionnaya sputnikovaya sistema (GLONASS), beheerd door Rusland.
- BeiDou Navigation Satellite System (BDS), beheerd door China, ook Compass genoemd.
- Indian Regional Navigation Satellite System (IRNSS/NavIC), beheerd door India.
- Quasi-Zenith Satellite System (QZSS), beheerd door Japan

Er zijn verschillen tussen tussen deze Global Navigation Satellite Systems, namelijk de accuraatheid en het aantal satellieten. QZSS en IRNSS zijn niet ontworpen als globale

Systeem	Aantal satellieten (actief)	Nauwkeurigheid
BDS	35	3.6 meter
NAVSTAR	31	3-5 meter
GLONASS	24+	4-10 meter
Galileo	24+	20-50 centimeter
IRNSS	7	<10 meter over India en >20 meter over de Indische Oceaan regio
QZSS	7	1 - 0,01 meter

Tabel 2.1: Overzicht Global National Satellite Systems

GPS-systemen. (Zie tabel:2.1.2)
De reden voor het bestaan van deze verschillende systemen is van politieke aard. De EU en Rusland wilden niet afhankelijk zijn van de Verenigde Staten (NAVSTAR).

2.2 Verschillende Locatiebepaling-technologieën

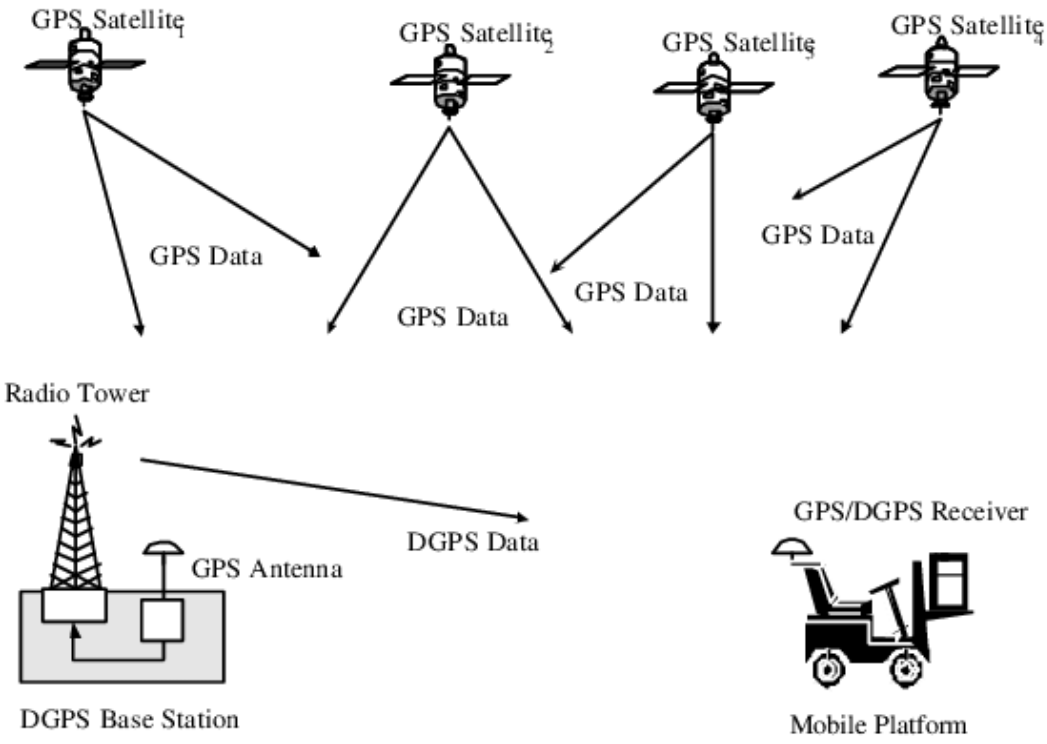
Het bepalen van een positie kan op veel manieren gebeuren. De systemen die besproken werden in de vorige sectie vallen allemaal terug op de traditionele methode, trilateratie. De werking van deze methode wordt in deze sectie verder toegelicht. Naast de traditionele methode is het belangrijk om te weten welke andere opties er nog bestaan. Er is onderzoek gedaan naar alle locatiebepaling-technologieën om de meest accurate technologie te bepalen die gebruikt kan worden door de proof of concept. Naast de accuraatheid moet ook de locatiebepaling op zoveel mogelijk plaatsen succesvol werken zodat de proof of concept gebruikt kan worden als lawinepieper in de bergen of als GPS-tracker aan een kust.

2.2.1 Standard Positioning Service (SPS)

De 'Standard Positioning Service' (SPS) is een positionerings- en timingdienst die signalen broadcast op de GPS L1-frequentie. Deze frequentie, die gebruikt wordt door alle NAVSTAR GPS-satellieten, zorgt ervoor dat GPS vrij te gebruiken is voor iedereen. (Grimes, 2008a)
Deze vorm van GPS minder accuraat in vergelijking met het 'Precision Positioning Service'.

2.2.2 Precision Positioning Service (PPS)

De 'Precision Positioning Service' (PPS) is een positionerings- en timingdienst die signalen broadcast op de GPS L1 en L2 frequenties. Het functioneert hetzelfde als SPS, maar hierbij wordt er ook gebruik gemaakt van een precisiecode (P) die alleen beschikbaar is voor geautoriseerde gebruikers. Hiermee wordt voornamelijk het Amerikaans leger en zijn allianties bedoeld. (Grimes, 2007) Voor september 2007 werd er gebruik gemaakt van 'Selective Availability' (SA). Zo kon het leger van de Verenigde Staten de accuraatheid

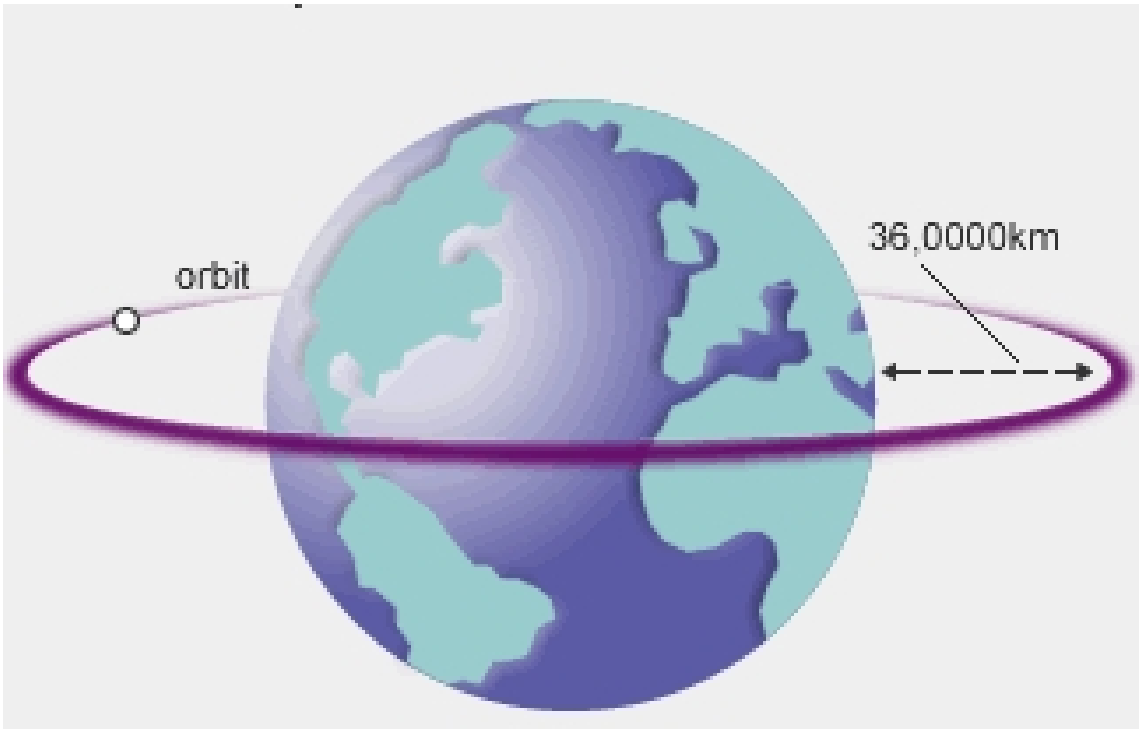


Figuur 2.4: Na het bepalen van de locatie via satellieten, ontvangt de GPS-ontvanger correcties van een DGPS station (Restivo, 2004)

bewust degraderen voor veiligheidsredenen. De nieuwere GPS III-satellieten zouden niet meer uitgerust zijn met de SA-functie.
Deze service wordt voor de proof of concept niet gebruikt omdat het ontoegankelijk is voor gewone burgers.

2.2.3 Differential Global Positioning System (DGPS)

Differential Global Positioning System (DGPS) verbetert de accuraatheid van GPS. Dit systeem maakt gebruik van correctietechnieken. Deze technieken kunnen real-time gebeuren of ook na het bepalen van de locatie. DGPS vereist een GPS-ontvanger (referentiestation/basisstation) waarvan de locatie gekend is (fixed). De referentie gaat zijn locatie herberekenen en berekent het verschil tussen de gekende locatie en de berekende locatie. Het verschil wordt als volgt toegepast op de GPS-ontvanger zijn berekende locatie. Deze techniek resulteert in een betere locatiebepaling voor de tweede GPS-ontvanger. (Chivers, g.d.) (Zie figuur: 2.4)
De 'Flemish Positioning Service' (FLEPOS) maakt eveneens gebruik van deze techniek. Het FLEPOS 3.0-netwerk bestaat momenteel uit 45 GNSS-referentiestations, waarvan 33 referentiestations in eigen beheer. Het gebruik hiervan is gratis indien de aanvraag voor een abonnement is goedgekeurd. Deze referentiestations bevinden zich in België, Nederland, Frankrijk, Duitsland en Engeland.
Door het beperkt aantal referentiestations kan deze methode niet geïmplementeerd worden



Figuur 2.5: Satellieten moeten zich op een hoogte van 36.000 km bevinden om op de geostationaire baan mee te draaien (Aidan-Phaswana, g.d.)

binnen de proof of concept, omdat het bereik niet globaal genoeg is.

2.2.4 Wide Area Augmentation System (WAAS)

'Wide Area Augmentation' (WAAS) kan, naast basisstation op aarde (DGPS), ook GEO-satellieten gebruiken als vaste referentiepunten. Deze satellieten bevinden zich op 36.000 km hoogte boven de evenaar en draaien rond op dezelfde snelheid als de aarde. Hierdoor blijft de positie van de geosatelliet stationair ten opzichte van de aarde, waardoor het gebruikt kan worden als referentiestation.(Zie figuur: 2.5)

2.2.5 Assisted Global Positioning System (A-GPS)

Een GPS-toestel kan er enkele minuten over doen om een locatie te bepalen. Deze verloren tijd heet 'Time To First Fix' (TTFF). De duur van deze TTFF hangt af van de exacte locatie en storing. Een open veld zal een kleinere TTFF hebben dan een bebouwde oppervlakte. Assisted Global Positioning (AGPS/A-GPS/aGPS) verkleint de TTFF.
Gsm-masten hebben GPS-ontvangers die voortdurend informatie van GPS-satellieten verzamelen. Op deze informatie worden allerlei bewerkingen uitgevoerd en vervolgens opgeslagen in een database. Dit heeft enkele voordelen voor de locatiebepaling van een gsm:

- Snellere locatie bepaling;

- Minder computerkracht nodig;
- Minder batterijverbruik;
- Mogelijkheid om de locatie indoor te bepalen.

Opmerkelijk is dat de accuraatheid minder goed is in vergelijking met het zelf bepalen van de locatie (Zie: 2.1.2). Er zullen field tests en metingen worden uitgevoerd om deze stellingen te controleren. (Rubino, 2009)

2.2.6 Advanced Mobile Location (AML)

Tot nu toe was er nog geen enkele techniek die een exacte locatie kon bepalen. 'Advanced Mobile Location' (AML) is de nieuwste (open source) techniek die hiervoor een oplossing biedt. Het maakt gebruik van:

- Wifi
- Sms
- GNSS

Door gebruik te maken van deze positioneringstechnieken is het de meest accurate en tegelijkertijd de traagste methode van alle besproken methodes.
Deze methode wordt gebruikt bij telefonische noodoproepen. Bij het begin van het bellen wordt er eerst gekeken of de wifi en GPS aanstaat. Als de 'battery check' positief is, worden deze services automatisch geactiveerd.
AML gaat parallel zijn locatie proberen bepalen aan de hand van wifi, sms en GNSS, zodat het zo min mogelijk tijd verliest om de beller te localiseren. De locatiebepaling gebeurt aan de hand van een geprioritiseerde rij. Indien GNSS als eerste de locatie kan bepalen wordt deze als eerste verzonden naar de hulpdiensten. Als tweede optie wordt de locatie bepaald met behulp van wifi. Hierbij wordt de locatie gebaseerd op wifi 'Service Set Identifiers' (SSIDs) of MAC-adressen van access points waarmee de gsm verbonden is. Als laatste optie wordt de cel ID verstuurd van de mast waarmee de beller verbonden is. Indien men geen locatie kan bepalen, wordt er een sms verstuurd dat alle positioneringstechnieken (wifi, GNSS, cel ID) gefaald hebben.
In 2020 zijn er wereldwijd slechts 19 landen die AML implementeren. AML kan alleen gebruikt worden bij het bellen van het noodnummer. De proof of concept werkt alleen met een eigen nummer, waardoor AML geen optie is. (EENA, 2016) Het nodeloos opbellen van een noodnummer is illegaal.

2.2.7 General Packet Radio Service (GPRS)

General Packet Radio Service (GPRS) is een totaal andere manier van locatiebepaling. Deze technologie werkt via roaming en vereist geen GPS. Deze verbinding haalt de locatie op van gsm-masten, waardoor via trilateratie de locatie bepaalt kan worden (Zie figuur: 2.6). Deze techniek werkt hetzelfde als GPS.
Het gebruik van deze techniek is echter veel minder accuraat zoals weergeven op figuur 2.6. Volgens CellTrack, g.d. kan dit variëren van 20 tot 50 meter. Het voordeel aan GPRS

Figuur 2.6: De locatie (blauwe ster) wordt bepaald aan de hand van trilateratie met gsm-masten. De berekende schatting kan zich overal bevinden in het blauwe vlak

is dat het in staat is om via roaming of sms zijn locatie te delen. Het gebruik van sms zorgt ervoor dat de proof of concept op een zeer globale manier gebruikt kan worden. Deze optie is budgetvriendelijker, want sms-en zijn voor providers zo goed als gratis in tegenstelling tot mobiele data. De combinatie van een GPS-ontvanger (ondersteuning voor NAVSTAR, Galileo en GLONASS) en de mogelijkheid tot het versturen van een sms zou zorgen voor een ideale proof of concept.

2.3 Ongunstige omstandigheden

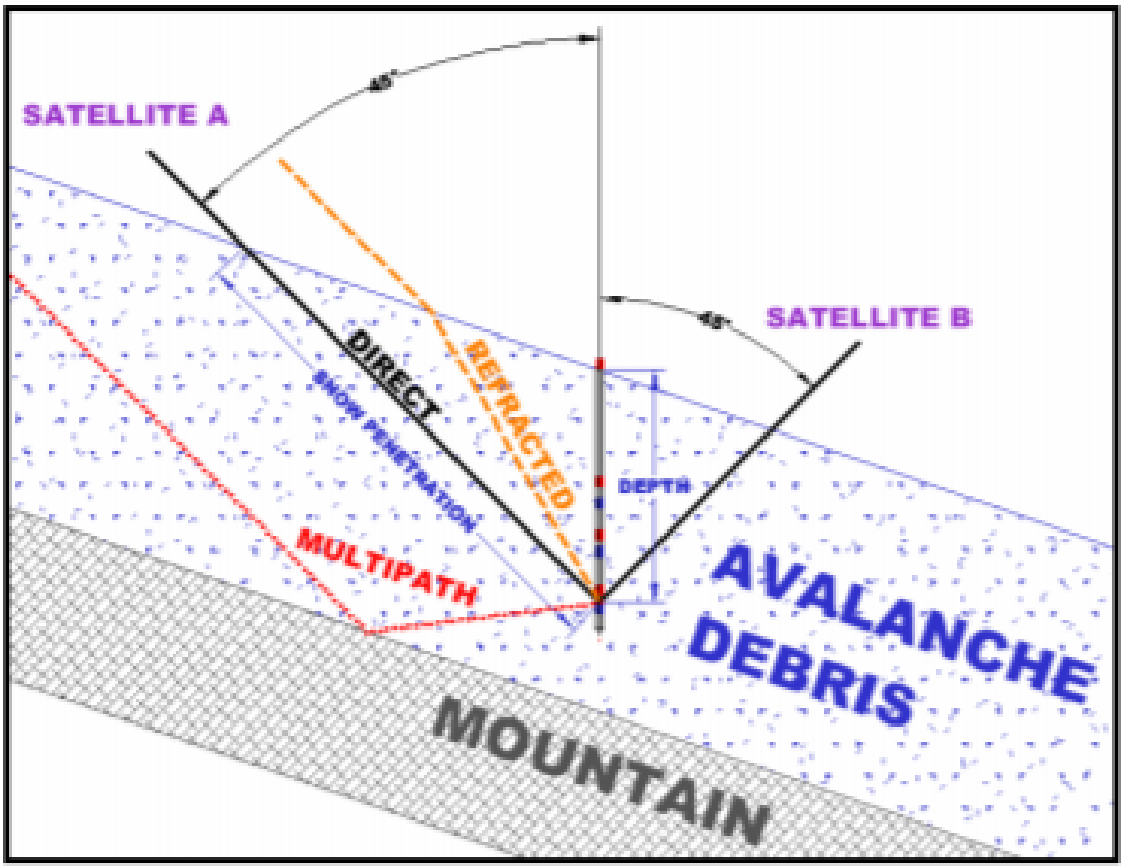
De proof of concept moet blijven functioneren in ongunstige omstandigheden, zoals sneeuw en water. Het moet in staat zijn om te functioneren als GPS-ontvanger op een surfboard en als lawinepieper. In deze sectie wordt onderzocht aan welke voorwaarden de proof of concept moet voldoen.

2.3.1 Water

Het grootste probleem voor GPS-ontvangers is dat ze niet kunnen functioneren onder water. Dit komt doordat de sterkte van het signaal te snel afneemt onder water. Hetzelfde geldt voor het mobiele netwerk, want beiden zijn elektromagnetische golven. (Taraldsen, Reinen & Berg, 2011) Er is ook een verschil tussen zout en zoetwater. Het zout in zeewater versterkt het geleidend effect waardoor GPS-signalen nog minder presteren in zeewater dan in zoetwater. Voor de proof of concept is dit geen struikelblok, indien het aan de zijkant van een surfboard wordt gemonteerd. Surfboards zijn zodanig gemaakt dat ze altijd blijven drijven, waardoor de proof of concept zal blijven functioneren naar behoren. Indien de plank gedraaid is, blijft de proof of concept werken. Dit wordt besproken in hoofdstuk 6.

2.3.2 Sneeuw

In tegenstelling tot water, blijven GPS-signalen wel functioneren onder sneeuw tot een zekere diepte. Een GPS-signaal dat passeert door lawine puin zal onderworpen worden aan verschillende effecten zoals verzwakking, reflectie, verstrooiing, refractie, multipad en vervagen. (Zie figuur: 2.7) Sneeuw is net als water een geleidend materiaal, waardoor het (L1 band) GPS-signaal verzwakt. De verzwakking hangt af van de hoeveelheid water in de sneeuw. Een GPS-signaal kan bijvoorbeeld tot vierhonderd meter diep gaan bij sneeuw met een massadichtheid van 0.3 g/cm^3. Bij sneeuw waarbij het volumepercentage meer dan 1% bedraagt kan het GPS-signaal slechts 3 meter diep doordringen.(Schleppe & Lachapelle, 2006)



Figuur 2.7: Negatieve invloed van refractie en weerkaatsing op GPS-signalen tijdens het penetreren van sneeuw (Schleppe & Lachapelle, 2006)

3. Toestellen

Voor de proof of concept werd er goed nagedacht welke locatiebepaling-technologieën gebruikt kunnen worden. Het moet niet enkel functioneren in ideale omstandigheden, maar ook in ongunstige. Er moet ook voldaan worden aan de volgende vereisten:

- Mogelijkheid tot het bepalen van de locatie;
- Mogelijkheid tot het delen van de locatie via sms;
- Mogelijkheheid tot het delen van de locatie via roaming;
- Mogelijkheid om te werken op batterij;
- Mogelijkheid tot waterdichtheid;
- Zo goedkoop mogelijk blijven.

Naast deze eisen moet de batterijduur zo lang mogelijk meegaan en de locatiebepaling moet zo accuraat mogelijk zijn.
Er is bewust gekozen om geen smartphones te gebruiken. Eerst en vooral hebben smartphones een hoog batterijverbruik omdat er de hele tijd een besturingssyteem actief is. Hiernaast zijn GPS chips die gebruikt worden in smartphones vaak inaccuraat.

3.1 Raspberry Pi

Er bestaan verschillende veries van de Raspberry Pi. De versies die in aanmerking komen kunnen teruggevonden worden in tabel 3.1
Naast de Raspberry Pi zijn er nog componenten nodig (zie tabel:3.2).

Model	Prijs (euro)
Raspberry Pi 4 Model B	39.95
Raspberry Pi 3 B+	39.95
Raspberry Pi 3 B	37.95
Raspberry Pi 3 A+	26.80

Tabel 3.1: Prijsvergelijking modellen Raspberry Pi. De prijzen zijn geraadpleegd op https://www.raspberrypi.org/products

Component	Prijs (euro)	Doel
Raspberry Pi 3A+	26.80	Het hoofd toestel, draait een besturingssysteem en voert alles aan.
SD-kaart (min. 8GB)	9.99	Hierop wordt het besturingssysteem en het script opgeslagen.
GSM HAT	46.61	Dit zorgt voor online communicatie om de data door te sturen.
5V Battery	54.95	Hierdoor kan de proof of concept draadloos werken.
GPS HAT	39.95	Een meer accurate locatiebepaling in vergelijking met AGPS.
Totale kost:	**178.30**	

Tabel 3.2: Prijs opstelling Raspberry Pi

3.2 Arduino

Er kan eveneens gebruik gemaakt worden van een Arduino. Arduino is de marktleider in open source hardware en bezit over hun eigen software eco-systeem. Zo beschikken ze over hun eigen programmeertaal en code text editor. Ze brengen verschillende hardware uit op de markt die gebruikt kan worden voor 'Internet of Things'. Ze bieden een module aan die geprogrammeerd kan worden die voldoet aan alle eisen, namelijk de MKR 1400 GSM module. (Zie figuur: 3.1) Er zijn uiteraard nog componenten nodig (zie tabel: 3.3).

Component	Prijs (euro)	Doel
MKR GSM	80.35	Dit component runt het script en stuurt de andere componenten aan.
MKR GPS Shield	41.75	Een meer accurate locatiebepaling in vergelijking met GPRS.
LiPo 3.7V 2500mAh	25.90	Dit zorgt ervoor dat het MKR board draadloos kan werken.
Totale kost:	**148.00**	

Tabel 3.3: Prijs opstelling Arduino

Figuur 3.1: Arduino MKR 1400 GSM module (Arduino, g.d.). Linksboven = Arduino MKR GSM, onderaan = GSM antenne, rechts = simkaart

4. Methodologie

Het onderzoek is begonnen met een literatuurstudie over het Global Positioning System en alle andere locatiebepalingstechnieken. De literatuurstudie is terug te vinden in hoofdstuk 2.

Na de literatuurstudie volgt het ontwikkelen van de proof of concept, de webapplicatie, de backend en de mobiele applicatie. In sectie 5.1 wordt er besproken hoe de proof of concept ontworpen is. Voor het script is er gebruik gemaakt van de code-editor van Arduino.

De data van de proof of concept wordt verwerkt in een backend die online geplaatst is op Heroku. Deze backend wordt besproken in sectie 5.2.

Vervolgens wordt de data die ontvangen is weergegeven in een webapplicatie (sectie 5.3). Dit zorgt ervoor dat de proof of concept gebruiksvriendelijk is.

Na het ontwikkelen van de hele applicatie en de proof of concept werden verschillende experimenten en testen uitgevoerd. Deze resultaten zijn terug te vinden in hoofdstuk 6.

4.1 Requirementanalyse

Voor dit onderzoek is er een requirementanalyse uitgevoerd om te weten te komen of de proof of concept een meerwaarde kan bieden voor ervaren sneeuw- en/of watersporters. De requirementanalyse is uitgevoerd aan de hand van een enquête (zie bijlage B).
De enquête onderzocht de vereisten en verwachtingen van de proof of concept. De

resultaten van de requirementanalyse worden besproken in Hoofdstuk 6.

5. Corpus

5.1 Proof of concept

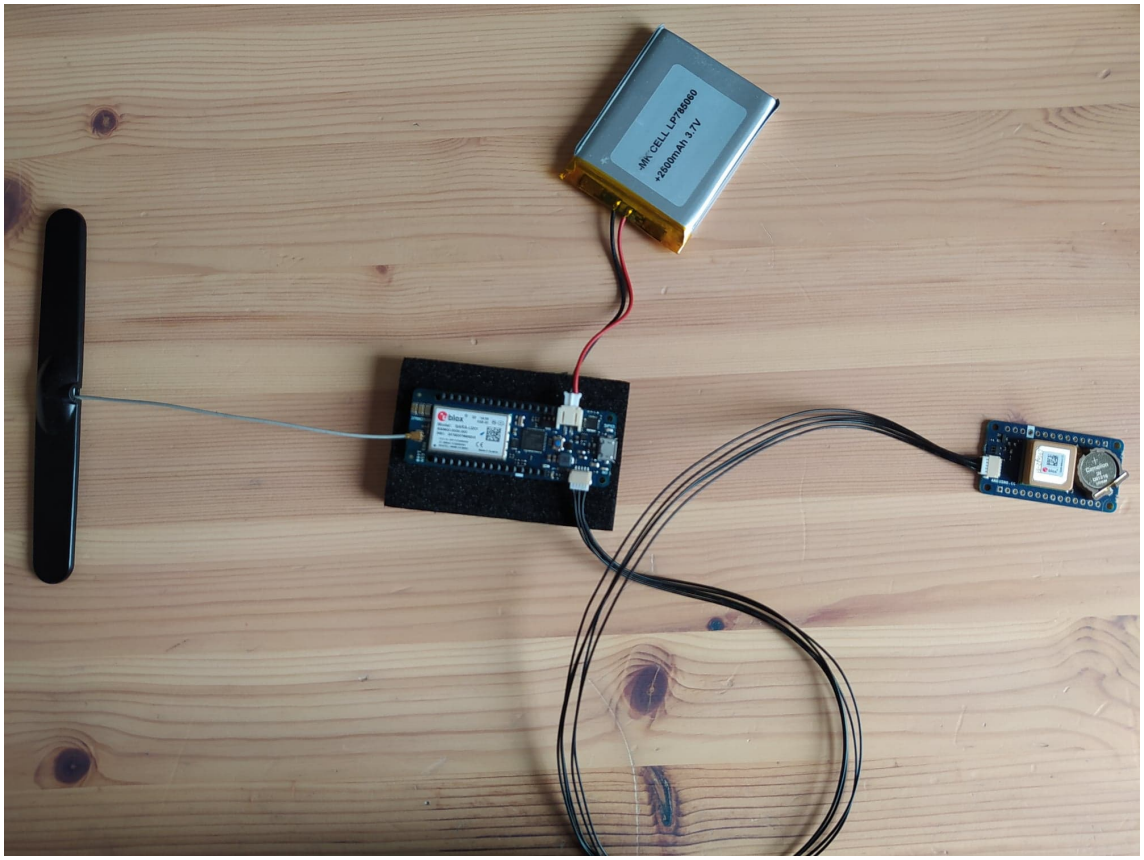
Arduino is een bekend begrip binnen IoT. Het is hardware waar veel wordt op teruggevallen voor diverse projecten. Hierdoor kan er gebruik gemaakt worden van veel 'libraries' die ontwikkelt zijn door de grote gemeenschap die achter Arduino staat.
Voor de proof of concept is de Arduino gekozen omdat dit een microcontroller is. Microcontrollers zijn zuiniger qua stroomgebruik ten opzichte van een 'computer' (Raspberry Pi). Hierdoor heeft de Arduino een langere batterijduur. Naast het verbruik is het eveneens goedkoper (zie hoofdstuk 2).
Naast de mogelijkheden op vlak van programmatie en performantie, is de grootte ook belangrijk. Hoe kleiner hoe beter, zodat de winter-/watersporter minder verhinderd wordt door de GPS-tracker tijdens het sporten.

5.1.1 Opstelling proof of concept

De **uitgewerkte proof of concept** (zie figuur: 5.1) bestaat uit:

- Arduino MKR GSM 1400 Cellular Kit (80.35 euro);
- Arduino MKR GPS Shield (41.75 euro);
- Lithium Ion Polymeer Accu - 3.7V 2500mAh (25.90 euro).

De totale kost van de proof of concept komt neer op **148.00 euro**. Naast de lage kost heeft de proof of concept het voordeel dat hij volledig zelf geconfigureerd kan worden. In functie van het onderzoek werd een eigen script ontwikkeld. De configuratie is gebeurd in



Figuur 5.1: opstelling proof of concept

de editor van Arduino zelf. Het script is in staat om de volgende functies uit te voeren:

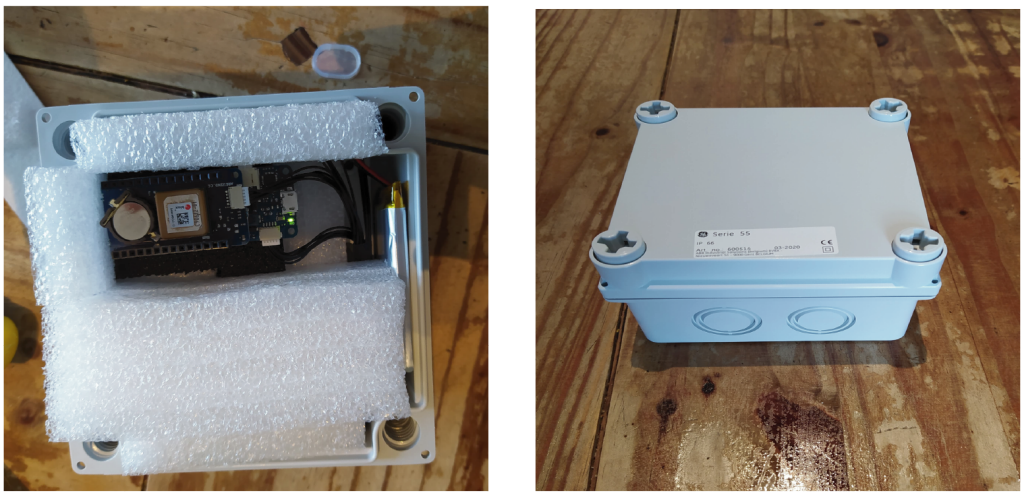
- De locatie bepalen aan de hand van GPS;
- De locatie bepalen aan de hand van GPRS;
- De locatie doorsturen naar een REST API aan de hand van een POST-request (JSON).

Het script is zelfontwikkeld en staat open source op een github repository. In het script werd gebruik gemaakt van verschillende libraries om het traceren mogelijk te maken. Er werd gebruik gemaakt van:

- ArduinoHttpClient: voor het versturen van data aan de hand van POST requests;
- MKRGSM: voor het gebruik van mobiele data;
- ArduinoJson: het mogelijk maken om makkelijk een JSON body op te vullen;
- ArduinoMKRGPS: voor het bepalen van de locatie via GPS.

De proof of concept maakt gebruik van een simkaart (zie figuur: 5.2) om online data te kunnen versturen.

Figuur 5.2: Achterkant van de MKR 1400 gsm module waar de simkaart wordt geplaatst



Figuur 5.3: Waterdichte behuizing

5.1.2 Waterdichte casing

In eerste instantie werd er niet geopteerd voor een waterdichte case te kopen, omdat dit de kost van de proof of concept onnodig verhoogt. Hierdoor werd er gekozen om zelf een waterdichte behuizing te ontwikkelen.
Het maken van een waterdichte behuizing is niet eenvoudig. In de ontwikkelingsfase werden twee modellen uitgeprint en getest. Daaruit bleek dat beide modellen niet waterdicht waren. Dit komt doordat de gebruikte 3D-printer (Anycubic Predator) niet nauwkeurig genoeg is om waterdichtheid te garanderen. Bovendien beschikt het gebruikte materiaal (PLA) niet over de juiste eigenschappen. Door de coronapandemie is het praktisch niet mogelijk om een nieuwe behuizing te ontwikkelen die wel waterdicht is. Daarom is er noodgedwongen gebruik gemaakt van een aftakdoos. Deze doos is volledig waterdicht en beschikt over een grote binnenruimte. Dit wilt zeggen dat er schokdemping toegevoegd kan worden zodat de proof of concept niet beschadigd wordt. De prijs is 18,14 euro.
De eerste case die geprint werd had een vrij eenvoudig ontwerp en functioneerde via een potdeksel-systeem. Na een test bleek de case niet waterdicht. De printer functioneerde niet nauwkeurig genoeg, waardoor de case onvoldoende afsloot. Daarna werd een 2de model uitgewerkt (zie figuur: 5.4), deze werd ontworpen met een schroefsysteem. Er waren een paar eisen waaraan de case moest voldoen. Doelstelling was om een optimale reikwijdte van de signalen te bekomen en dit door een mogelijkheid te creëren om de antenne buiten de behuizing aan te sluiten. Bij 2 opeenvolgende testen bleek dat deze case niet waterdicht , noch gebruiksvriendelijk was. Conclusie: de printer werkt niet nauwkeurig genoeg om een waterdichte behuizing te creëren, het gebruikte materiaal is niet optimaal om een waterdicht systeem te bekomen. Na twee pogingen en 36 uur printen, werd er gezocht naar een 3de mogelijkheid. De huidige behuizing is IP66 gecertificeerd. Dit certificaat garandeert dat de case waterdicht en stofvrij is.

Figuur 5.4: Tweede print

5.2 Backend

5.2.1 Ontwikkeling

Voor backend is er gebruik gemaakt van Mongoose. De gehele infrastructuur is gebaseerd op de MEVN-stack. Deze stack staat voor:

- Mongoose
- Express
- Vue
- Node.JS

Mongoose maakt gebruik van MongoDB Atlas. De backend, geschreven in Javascript, is open source en staat online op een github repository.
De functie van de backend is om locaties van de proof of concept te ontvangen. Deze locaties worden opgeslagen in een NO-SQL database. De database slaagt per locatie de volgende gegevens op:

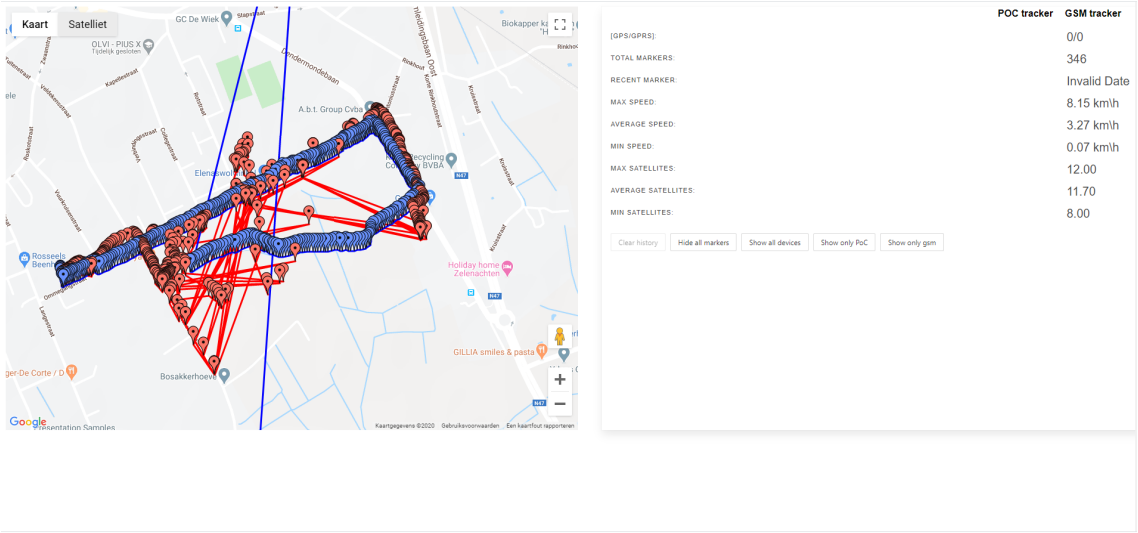
- Longitude
- Latitude
- Altitude
- Snelheid (in km/h)
- Aantal satellieten
- Method (locatie bepaalt via GPS of GPRS)

De methodiek hangt af van de beschikbaarheid van het signaal. Indien er geen GPS-signaal gevonden kan worden, schakelt de proof of concept automatisch over op GPRS. Maar GPS blijft wel de eerste keuze omdat dit accurater is dan GPRS. Indien de locatie bepaald is via

GPRS, heeft men geen informatie over de snelheid en het aantal satellieten. GPRS-data worden immers bepaald op basis van gsm-masten (zie hoofdstuk 2).

5.2.2 Deployment

Naast de code staat ook het programma online. De backend kan hier bekeken worden. Voor de deployment is er gebruik gemaakt van Heroku. Doordat het programma online draait, kan de webapplicatie deze data ophalen en weergeven.
De backend maakt gebruik van een gratis hosting plan, dit kan nadelig zijn. Het programma kan na 30 minuten inactiviteit in slaapstand gaan. Dat wil zeggen dat de webapplicatie twee keer moet opgeladen worden. De eerste keer dient om de backend terug te activeren, de tweede keer om effectief te gebruiken. Het probleem wordt opgelost door de webapplicatie te refreshen.



Figuur 5.5: Webapplicatie die de gebruiksvriendelijkheid van de GPS-tracker verhoogt

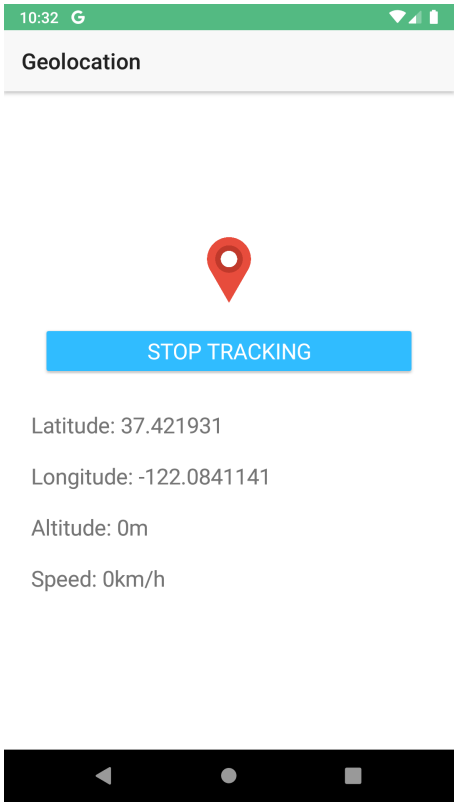
5.3 Frontend

5.3.1 Ontwikkeling

Zoals reeds verteld is er gebruik gemaakt van de MEVN-stack. Dit wil zeggen dat de webapplicatie ontwikkeld is in Vue, een nieuwer javascript-framework. De code van deze webapplicatie is open source en staat online op een github repository.
Voor de webapplicatie (zie figuur: 5.5) is er gebruik gemaakt van 'Google Maps'. Google Maps is een populaire online kaartendienst waarmee geografische locaties kunnen opgezocht worden waardoor dezelfde layout een vorm van betrouwbaarheid en herkenbaarheid geeft voor de aankoper van het toestel. Dit bevordert de gebruiksvriendelijkheid van de webapplicatie. Naast het tracken zelf, geeft het ook extra informatie weer, zoals welke methode gebruikt wordt, hoe snel de GPS-tracker zich voortbeweegt en hoe betrouwbaar de coördinaten zijn (aantal satellieten). De gebruiker kan de geschiedenis van de locaties wissen via een knop.
Voor de implementatie van Google Maps is er gebruik gemaakt van een open source en gratis te gebruiken library, namelijk vue2-google-maps.

5.3.2 Deployment

Om de website online te zetten is er gebruik gemaakt van Netlify. Netlify is een hosting bedrijf dat een gratis plan aanbiedt. De webapplicatie staat hier online.



Figuur 5.6: Zelf ontwikkelde mobiele applicatie

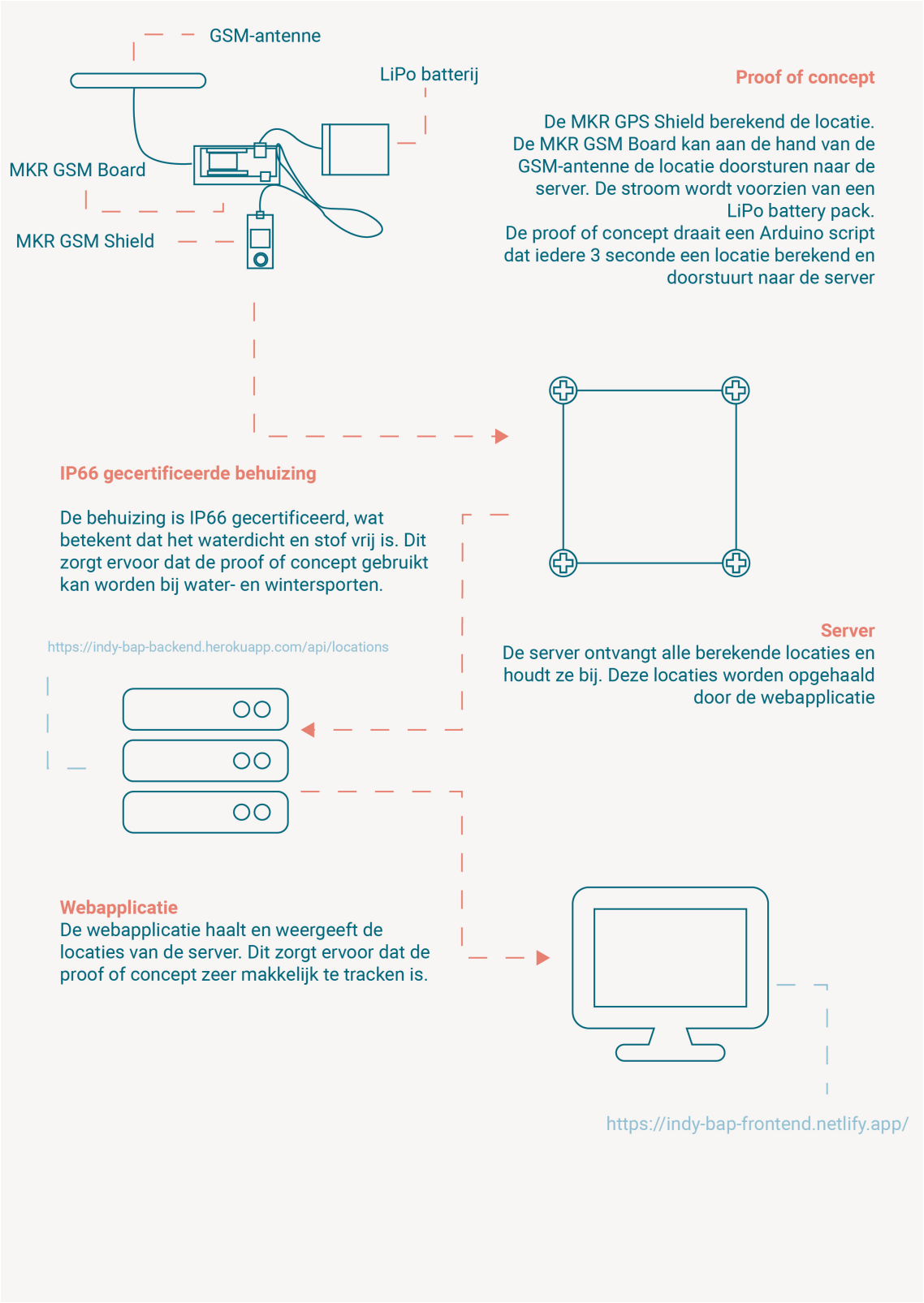
5.4 Mobiele applicatie

Om de vergelijking te kunnen maken tussen de GPS in een gsm en de proof of concept, is er een mobiele applicatie ontwikkeld (zie figuur: 5.6). Deze mobiele applicatie herberekent de locatie met een interval van drie seconden, waarna ze dan wordt doorgestuurd naar de backend. (sectie 5.2). De mobiele applicatie is ontwikkeld in NativeScript. NativeScript is een open source framework dat gebruikt kan worden om native mobiele applicaties te maken voor Android en iOS. Om NativeScript te gebruiken kan er gekozen worden uit vier opties om de app te ontwikkelen:

- Angular
- Vue
- Javascript
- Typescript

Voor de mobiele applicatie is er gebruik gemaakt van Vue. De code van de mobiele applicatie is open source en staat online op een github repository.

5.5 High-level architectuur

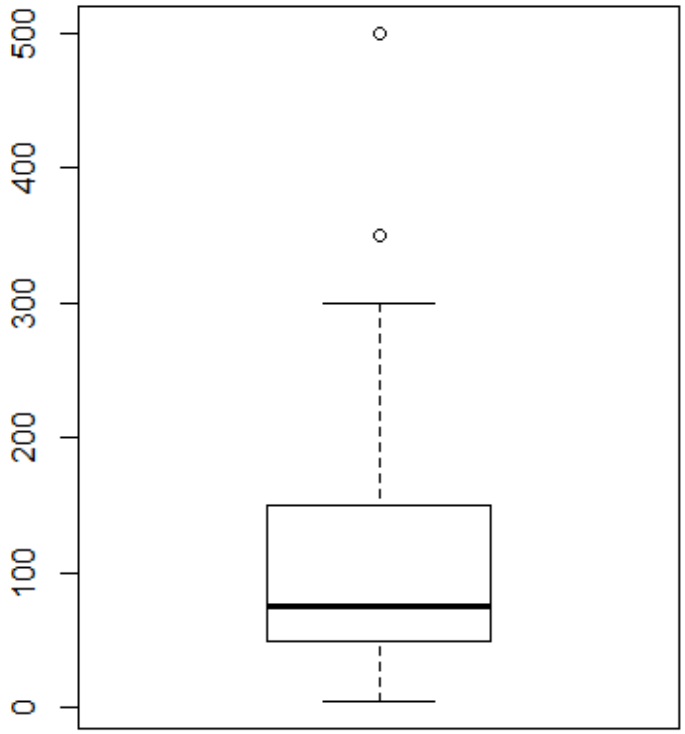


Figuur 5.7: Schematisch overzicht van de high-level architectuur van de proof of concept

6. Resultaten

6.1 Bespreking requirementanalyse

De requirementanalyse heeft een beperkte steekproefomvang (n=70). Dit is echter niet problematisch, aangezien de opzet van de analyse niet is om te generaliseren naar de algemene populatie. Alleen water- en wintersporters werden bevraagd, dit is geen eenvoudige doelgroep om te bereiken. Bovendien werd de dataverzameling bemoeilijkt door de coronapandemie. Het was bijvoorbeeld niet mogelijk om respondenten te rekruteren op andere manieren dan via een online survey. De resultaten beperken zich tot een univariate bespreking. Meer complexe analyses behoren immers niet tot de reikwijdte van het onderzoek. Bovendien zouden de resultaten wellicht vertekend zijn door de beperkte steekproefomvang. Uit de analyse blijkt dat de overgrote meerderheid van de respondenten (95,7%) reeds van de GPS-tracker gehoord heeft. Ondanks de bekendheid met het concept, is het opvallend dat slechts een minderheid van de respondenten (28,6%) reeds overwogen heeft om een dergelijke tracker aan te kopen. Wellicht vormt de hoge kostprijs een barrière voor de aanschaf van een dergelijk toestel. Slechts een beperkte proportie van de respondenten (42,9%) is reeds op de hoogte van het bestaan van GPS-trackers op basis van een webapplicatie. Uit de enquête (zie figuur: 6.1) blijkt dat de respondenten 166,14 euro doorgaans te veel vinden voor een GPS-tracker. Gemiddeld is men bereid om maximaal 99,98 euro te betalen voor een dergelijke tracker (mediaan: 75 euro).



Figuur 6.1: Boxplot over de maximumprijs die een respondent wil betalen voor een GPS-tracker

Code in R voor het plotten van de boxplot:

```
library("readxl")
library(ggplot2)

> prices<-read_excel("C:/Users/IndyV/Desktop/maxPrijs.xlsx")
> prices<-as.numeric(unlist(prices))
> mean<-mean(prices)
> mean
[1] 99.97969
> median<-median(prices)
> median
[1] 75
> prices=data.frame(prices)
```

```
> plot <- boxplot(prices)
```

Om toeval uit te sluiten, is er een T-test uitgevoerd. De T-test wijst uit dat de gemiddelden op waterbestendigheid (X= 9,61429), accuraatheid (X=8,45714) en gebruiksvriendelijkheid (X=8,05714) significant verschillend zijn, ondanks de relatief kleine steekproef. Hieruit kan geconcludeerd worden dat waterbestendigheid de belangrijkste vereiste is voor respondenten. De proof of concept voldoet aan deze eis.

Code in R voor het berekenen van de significantie:

```
library("readxl")

> dataset<-read_excel("C:/Users/IndyV/Desktop/dataset.xlsx",col_names=TRUE)
> df<-data.frame(dataset)
> mean_accuraatheid<-mean(df$accuraatheid)
> mean_waterbestendigheid<-mean(df$waterbestendigheid)
> mean_gebruiksvriendelijkheid<-mean(df$ gebruiksvriendelijkheid)
>
> smallest_mean<-min(mean_accuraatheid,mean_waterbestendigheid,mean_gebruiksvriende
>
> t.test(df$accuraatheid,mu = smallest_mean)

One Sample t-test

data:  df$accuraatheid
t = 2.0642, df = 69, p-value = 0.04276
alternative hypothesis: true mean is not equal to 8.057143
95 percent confidence interval:
8.070561 8.843725
sample estimates:
mean of x
8.457143

> t.test(df$waterbestendigheid,mu = smallest_mean)

One Sample t-test

data:  df$waterbestendigheid
t = 17.422, df = 69, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 8.057143
95 percent confidence interval:
9.435978 9.792594
sample estimates:
mean of x
9.614286
```

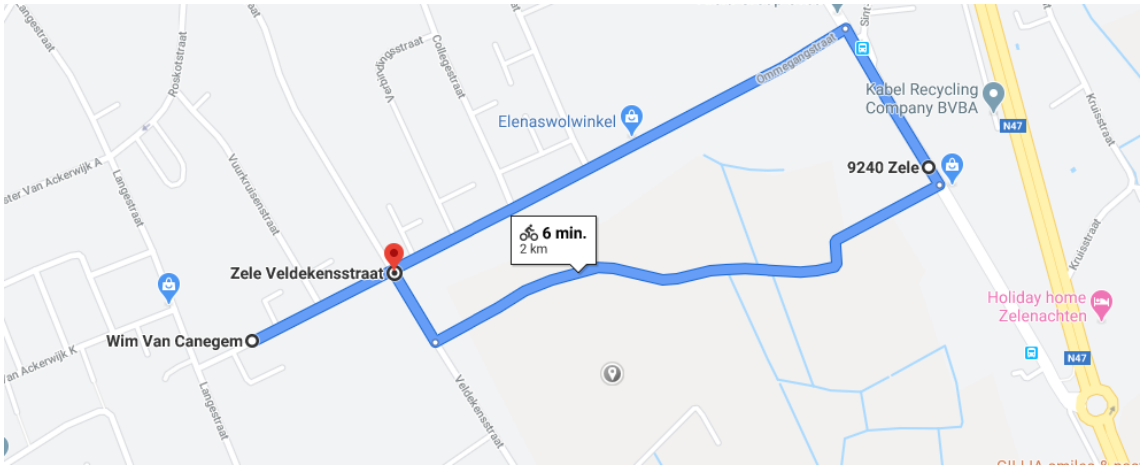
```
> t.test(df$gebruiksvriendelijkheid,mu = smallest_mean)

One Sample t-test

data:  df$gebruiksvriendelijkheid
t = 0, df = 69, p-value = 1
alternative hypothesis: true mean is not equal to 8.057143
95 percent confidence interval:
7.692046 8.422240
sample estimates:
mean of x
8.057143
```

De batterijduur van de proof of concept voldoet ook aan de eisen van de respondenten, namelijk 24 uur. De batterijduur kan ook verlengd worden indien er gebruik gemaakt wordt van een powerbank.
Indien de proof of concept op de markt komt, toont 28,6% van de respondenten zich bereid om deze aanschaffen, ondanks de vrij hoge kostprijs. Een groot deel van de respondenten (54.3%) twijfelt nog.
De proof of concept zou gebruikt worden voor:

- watersport = 38,6&
- sneeuwsport = 22,9%
- water- en sneeuwsport = 38,6%



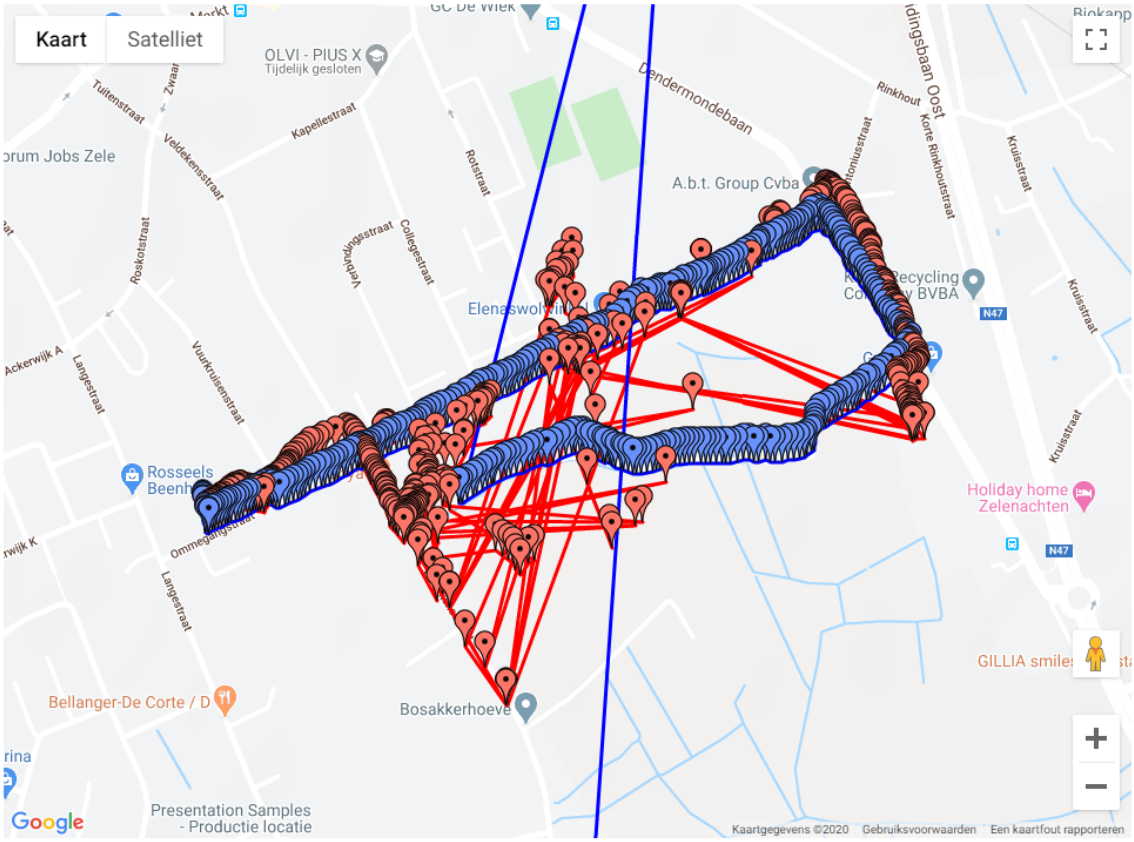
Figuur 6.2: De uitgestippelde route voor de eerste field test

Figuur 6.3: Het uitvoeren van de eerste field test

6.2 Vergelijking tussen GPS van een gsm en proof of concept

Voor de eerste test is er een mobiele applicatie ontwikkeld zodat er gegarandeerd wordt dat de gsm geen gebruik kan maken van extra informatie en/of algoritmen van Google Maps. Tijdens de eerste test werden er 2 toestellen gebruikt: de proof of concept en een gsm. De gsm is een Xiaomi Mi 9T (kostprijs 332 euro). Voor de eerste field test is er op voorhand een route uitgestippeld (zie figuur: 6.2). Tijdens het wandelen van deze route stuurden de proof of concept (5.1) en de mobiele applicatie (5.4) tegelijkertijd hun locaties door met een tijdsinterval van telkens drie seconden. (Zie figuur: 6.3)

Het resultaat is zeer opmerkelijk. In figuur 6.4 is het duidelijk dat de proof of concept (blauwe markers) beter scoort dan dan de GPS van een gsm (rode markers). Het is moeilijk om de exacte oorzaak van deze discrepantie in resultaten te achterhalen, omdat de data van de GPS-chip alleen verkrijgbaar zijn voor de fabrikant. De enige data die verkregen kunnen worden van de GPS-chip is de berekende locatie.

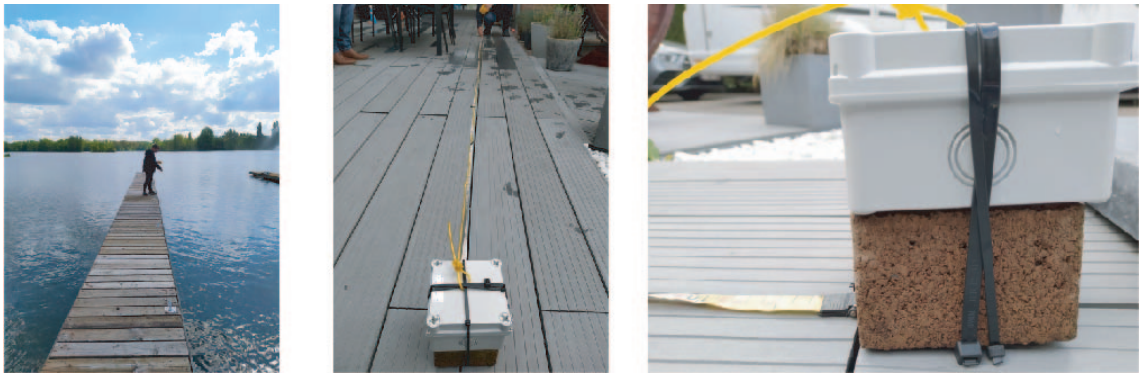


Figuur 6.4: Resultaat van de eerste field test

Het gebruik van Google Maps op een gsm kan een meer accurate locatiebepaling doen in vergelijking met het resultaat van de field test. Dit komt omdat Android en iOS extra informatie gebruiken zoals gsm-masten en wifi signalen (SSIDs). Google Maps maakt ook gebruik van verschillende algoritmen om de locatie te schatten op een straat zodat deze accurater lijken.
Om de probleemstelling op te lossen, moet een GPS-tracker werken aan de kust en in berggebieden. Ook is de route niet op voorhand geweten, waardoor een gsm zeer slecht zou scoren als GPS-tracker. Google Maps, hetgeen gebruikt wordt door gsm's, bepaalt de locatie immers op basis van straten. In gebieden waar geen of weinig straten zijn, is een exacte locatiebepaling derhalve onmogelijk. De proof of concept lijkt wel een goede oplossing te zijn voor de probleemstelling, omdat het volledig onafhankelijk is van extra informatie en/of algoritmen tijdens het gebruik van Google Maps.
Uit de eerste field test kan er dus geconcludeerd worden dat de proof of concept slaagt in zijn opzet.

6.3 Prestaties onder water

Zoals vermeld in hoofdstuk 2 kunnen elektromagnetische golven water amper of niet penetreren. Hierdoor leek dit een interessante test.



Figuur 6.5: Het uitvoeren van de tweede field test

De tweede test (zie figuur: 6.5) onderzocht de waterdichtheid van de proof of concept en tot hoe diep de proof of concept kan blijven functioneren. Allereerst blijft de proof of concept functioneren onder het wateroppervlak. Na het openmaken van de behuizing werd er geen water teruggevonden. Om de test uit te voeren is er gebruik gemaakt van een waslijn met markeringen iedere 0,25 meter.
Eerst en vooral is er getest geweest tot hoe diep de proof of concept functioneert. De proof of concept kan tot 0,25 meter blijven functioneren. Dit wil zeggen dat, indien het bevestigd wordt aan een surfboard, het in staat is om het surfboard te localiseren. Een surfboard blijft drijven aan het wateroppervlak. Naast de waterbestendigheid moest ook de traceerbaarheid bekeken worden. Om uit te sluiten dat het verzonden GPS-signaal berekend werd boven het wateroppervlak, is de proof of concept op een diepte van 0,25 meter verplaatst rond de hele pier. De verplaatsing van de proof of concept was zichtbaar, waardoor er vastgesteld kan worden dat de proof of concept op een diepte van 0,25 meter perfect functioneert.
Er viel op dat er op een diepte van 0,25 meter niks van data ontvingen werd door de back-end, waardoor er geconcludeerd kan worden dat de internet verbinding de 'bottleneck' is. Dit is mogelijks te wijten aan de gebruikte frequenties. Hoe kleiner de frequentie, hoe dieper een signaal water kan penetreren. Dit werd niet verder onderzocht, omdat mobiele data gebruik kan maken van verschillende frequenties.
De resultaten zijn positief, want naast GPS-tracker kan de proof of concept ook als lawinepieper functioneren. De massadichtheid van sneeuw is lager dan die van water, waardoor de proof of concept dieper dan 0,25 meter blijft functioneren. Hoe diep juist, kan niet onderzocht worden wegens de coronapandemie.

7. Conclusie

Dit onderzoek geeft een antwoord op de onderzoeksvraag 'is het mogelijk om een economisch rendabel GPS-toestel te ontwikkelen dat niet bezwijkt onder ongunstige omstandigheden?'.

Voor het onderzoek is er een literatuurstudie uitgevoerd om de juiste locatiebepalingstechnieken te bepalen. Ook zijn de vereisten hier naar boven gekomen waaraan de proof of concept moest voldoen.

De proof of concept slaagt slechts gedeeltelijk, want uit de requirementanalyse blijkt dat de prijs hoger ligt dan wat respondenten er maximum aan zouden willen uitgeven. Een mogelijke verklaring hiervoor is dat de ondervraagden niet volledig weten wat de mogelijkheden van een GPS-tracker op basis van een webapplicatie zijn, aangezien 42,9% van de respondenten niet bekend is met dit concept. Toch is de prijs veel lager dan de concurrentie en de vooropgestelde maximumprijs van 200 euro. In het onderzoek kon een waterdichte, nauwkeurige GPS-tracker gecreëerd worden voor **166,14**. Uit de testen bleek dat de proof of concept resistent is tegen ongunstige omstandigheden.

Indien er gekeken wordt naar de deelonderzoeksvragen, los van de requirementanalyse, slaagt de proof of concept.

- De proof of concept is in staat zijn locatie te delen.
- De proof of concept is goedkoper dan een smartphone (200 euro).
- De proof of concept is waterbestendig.
- De proof of concept blijft functioneel werken onder water tot een diepte van 0,25 meter.
- De proof of concept werkt accurater dan een ingebouwde GPS van een gsm.

- De proof of concept kan getracked worden aan de hand van een webapplicatie en is gebruiksvriendelijk (subjectief).

De deelonderzoeksvraag 'is de proof of concept zoutbestendig?' kon niet onderzocht worden wegens de coronapandemie. Ook de functionaliteit van de proof of concept onder sneeuw kon niet onderzocht worden. Er kan vermoed worden dat de proof of concept ook blijft functioneren onder een sneeuwlaag van meer dan 0,25 meter omdat de massadichtheid van water hoger ligt dan die van sneeuw. Hoe de GPS-tracker functioneert in zout water, is vooralsnog onbekend.
De proof of concept slaagt voor alle (onderzochte) onderzoeksvragen. Het is derhalve bewezen dat het mogelijk is om een gebruiksvriendelijk, economisch rendabel GPS-toestel te ontwikkelen dat niet bezwijkt onder ongunstige omstandigheden.

A. Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

A.1 Introductie

Een van de meest verspreide ICT-toepassingen als gevolg van de recente technologische revolutie is het Global Positioning System (GPS). Met GPS wordt er impliciet verwezen naar de Standard Positioning Service (SPS). De SPS is een positionerings- en timingdienst die wordt aangeboden door middel van variërende signalen die worden uitgezonden op de GPS L1 frequentie (Grimes, 2008b). Deze technologische toepassing heeft verregaande gevolgen gehad inzake de manier waarop onze samenleving georganiseerd wordt. Mensen gebruiken de GPS om files te ontwijken op weg naar hun werk en om niet te verdwalen op reis, maar GPS systemen kunnen ook gebruikt worden voor water- en wintersporten. Praktische voorbeelden hiervan zijn lawinepiepers. Een lawinepieper is een radiozender en -ontvanger die wordt gebruikt om mensen op te sporen die onder een lawine zijn geraakt (Jürg Schweizer, 2003).

Lawinepiepers zijn vaak niet goedkoop en er bestaan weinig goedkope alternatieven die dezelfde kwaliteit behouden. Lawinepiepers zijn te verkrijgen vanaf ongeveer 200 euro. Tussen de verschillende merken van deze lawinepiepers heerst er een groot verschil aan kwaliteit en prijs. In deze bachelorproef wordt er nagegaan of het mogelijk is om een product te ontwikkelen dat even kwalitatief werkt als professionele toestellen die momenteel gebruikt kunnen worden in ongunstige omstandigheden zoals bij het uitoefenen van water- en wintersporten, maar dan goedkoper.

A.2 State-of-the-art

Volgens Jürg Schweizer (2003) is het niet vanzelfsprekend om iemand te vinden met behulp van lawinepiepers. Het vinden van de beacon hangt sterk af van welk merk er gebruikt wordt. Uit het onderzoek van Jürg Schweizer (2003) blijkt dat er een significant verschil is in de kwaliteit van de toestellen en in de tijd waarin het duurt om iemand terug te vinden. Het tijdsbestek waarin fictieve lawineslachtoffers teruggevonden werden, varieert tussen 109,5 seconden en 215 seconden. In beide gevallen werd de mediaan gebruikt voor het bepalen van het tijdsbestek per model. De kwaliteit van deze toestellen kan afhangen van welke GPS-service ze gebruik maken. Zo is er de Standard Positioning Service (SPS), die als minder nauwkeurig beschouwd wordt, en de Precise Positioning Service (PPS). De SPS-service is vrij te gebruiken door particulieren en bedrijven, terwijl de PPS-service een autorisatie vereist die in de praktijk alleen te gebruiken valt voor militaire toepassingen. (GMV, 2011) Lawinepiepers zijn niet geautoriseerd om PPS te kunnen gebruiken, waardoor we deze GPS-service level uitsluiten in dit onderzoek. Naast de twee GPS-service levels (SPS en PPS) bestaan er ook alternatieve manieren om locaties te bepalen. We hebben het traditionele Global Positioning System (GPS), dat gebruikt maakt van satellieten, maar ook Assisted Global Positioning Systems (A-GPS), die daarnaast ook gebruik maakt van masten voor mobiele telefoons. (Nikhilesh, Poonam, Kate & Pooja, 2019) Een andere mogelijkheid is het gebruik van De Flemish Positioning Service. In deze bachelorproef wordt er een proof of concept (PoC) opgeleverd die gebruik zal maken van de besproken technologieën. Deze PoC zal bestaan uit een Rasberry Pi 3 (RP3), aangezien deze voldoet aan de volgende vereisten: low-cost, mogelijkheid tot gebruik van GPS en mogelijkheid tot waterdichte casing. Voor waterdichtheid en bescherming van de RP3 zal er gebruik gemaakt worden van een plastic omhulsel, aangezien deze meer resistent zijn in ongunstige omstandigheden dan metaal (aipprecision, 2019). Bij dit onderzoek kan de volgende hypothese opgesteld worden en proberen we de nulhypothese te verwerpen: H1: Het is mogelijk om een low-cost GPS-systeem te ontwikkelen die bijna even kwalitatief is als een bestaand professioneel toestel die overleeft in ongunstige omstandigheden (water/sneeuw/zout). H0: Het is niet mogelijk om een low-cost GPS-systeem te ontwikkelen die een gelijkaardige kwaliteit bevat als een bestaand professioneel toestel dat overleeft in ongunstige omstandigheden (water/sneeuw/zout).

A.3 Methodologie

Er zullen experimenten/simulaties uitgevoerd worden waarbij de PoC gemonitord en vergeleken wordt met reeds bestaande toestellen. Hiervoor zullen er field tests uitgevoerd worden. Na de field tests wordt er bekeken hoe de PoC verbeterd kan worden om de kwaliteit van de reeds bestaande toestellen te evenaren en tegelijkertijd een low-cost device te blijven. Zo willen we de maximale diepte bepalen waarop de PoC een GPS-signaal blijft sturen, en dit zowel in water als in sneeuw. Ook de mate waarin de locatiebepaling accuraat is, wordt onder de loep genomen. Volgens (Meyer, 2016) zouden GPS-signalen zich niet goed kunnen verplaatsen in water. Vreemd genoeg claimt het merk divenetgps een diepte te bereiken van 300m. Voor de wetenschappelijke relevantie wordt getest hoe diep een

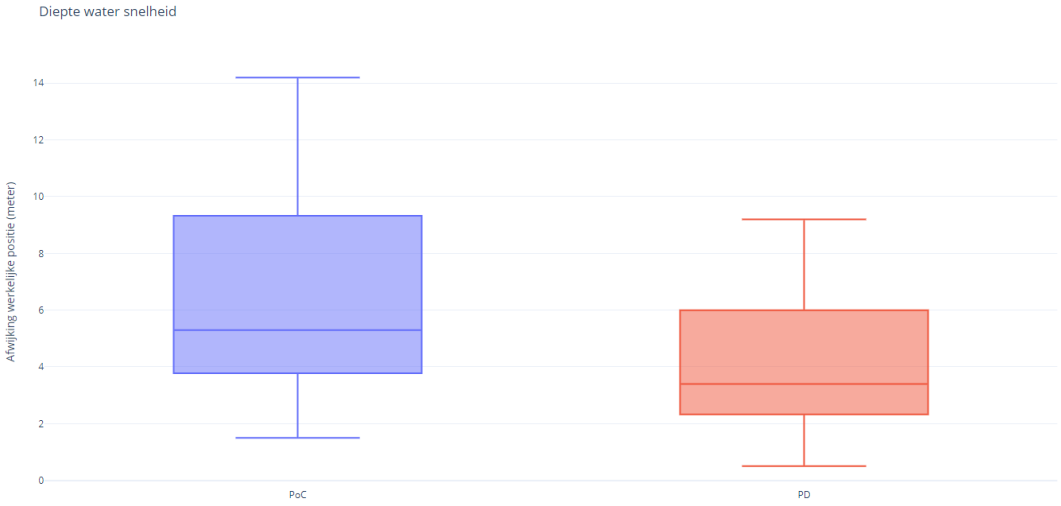
GPS-systeem blijft functioneren naar behoren onder water. Indien dit mogelijk is, kunnen er toepassingen uitgewerkt worden binnen watersporten, zoals het terugvinden van een surfboard. Om uit te schijnen tussen bestaande toestellen zal de gebruiksvriendelijkheid ook verbetert worden door middel van een applicatie die het 'tracken' van iets of iemand zeer eenvoudig maakt.

A.4 Verwachte resultaten

Voor de volgende grafieken geldt: (Deze grafieken zijn fictief, ze zijn slechts een verwachting)
PD = Professional Device, bestaand product op de markt
PoC = zelf ontwikkelde Proof of Concept

Verwachte resultaten van field tests in ongunstige omstandigheden (sneeuw):

Verwachte resultaten van field tests in ongunstige omstandigheden (water):



A.5 Verwachte conclusies

Uit dit onderzoek moet duidelijk blijken of het mogelijk is een low-cost GPS-systeem te ontwikkelen dat niet onderdoet voor de bestaande producten. De PoC zal relatief goede resultaten behalen ten opzichte van de producten die al op de markt zijn. Hopelijk vloeit uit dit onderzoek een PoC voort die klaar is voor de consument die opzoek is naar een GPS-systeem van vergelijkbare kwaliteit die gebruikt kan worden voor doeleinden zoals het vinden van een verloren surfboard tot het terugvinden van lawineslachtoffers.

B. Enquête

1. Weet u wat een GPS-tracker is?
   - Ja, ik weet wat een GPS-tracker is
   - Ja, ik heb er reeds van gehoord maar kan het begrip nog niet 100 percent duiden
   - Nee, ik ken het begrip niet
2. Heeft u reeds overwogen om een GPS-tracker aan te schaffen?
   - Ja
   - Nee
3. Heeft u ooit al gehoord van een GPS-tracker die gebruik maakt van een applicatie? (bijvoorbeeld Runkeeper, maar dan in de webbrowser)
   - Ja, ik heb er ervaring mee
   - Ja, ik heb er al van gehoord maar nog nooit één gebruikt
   - Nee, ik heb er geen ervaring mee
4. Indien u reeds een GPS-tracker heeft aangekocht, hoeveel heeft deze gekost?
   - 0 - 50 euro
   - 51 - 100 euro
   - 101 - 150 euro
   - 151 - 200 euro
   - Meer dan 200 euro
5. Hoeveel zou u maximum besteden aan een GPS-tracker?
6. Hoe belangrijk vindt u de accuraatheid van de locatiebepaling? Van weinig belangrijk (1) tot heel belangrijk (10).
7. Hoe belangrijk vindt u, als watersporter/sneeuwsporter, de waterbestendigheid van een GPS-tracker die u zou gebruiken tijdens het sporten? Van weinig belangrijk (1) tot heel belangrijk (10).
8. Hoe belangrijk vindt u de gebruiksvriendelijkheid van het tracken? Van weinig belangrijk (1) tot heel belangrijk (10).

9. Hoelang vindt u dat de GPS-tracker minstens moet blijven werken? Uitdrukken in aantal uren
    - 0-6
    - 6-12
    - 12-18
    - 18-24
10. Zou u de webapplicatie, die u toegang biedt tot het tracken, alleen gebruiken in nood en/of om activiteiten te delen op sociale media? (bijvoorbeeld Runkeeper, ...)
    - Alleen in noodgevallen
    - Alleen om activiteiten te delen op sociale media
    - Beide antwoordopties
11. Zou u een budgetvriendelijke GPS-tracker (max. 150 euro) aankopen indien deze op de markt komt?
    - Ja
    - Misschien
    - Nee
12. Indien u een budgetvriendelijke GPS-tracker aankoopt, voor welke sporten zou u deze gebruiken?
    - Watersport
    - Sneeuwsport
    - Allebei

C. Broncode Arduino

```
#include <ArduinoHttpClient.h>
#include <MKRGSM.h>
#include <ArduinoJson.h>
#include <Arduino_MKRGPS.h>

// PIN Number
const char PINNUMBER[]     = ""; //blank if no pin
// APN data: check settings on mobile for sim provider or contact carrier for network settings
const char GPRS_APN[]      = "telenetwap.be"; //This is for Telenet
const char GPRS_LOGIN[]    = "";
const char GPRS_PASSWORD[] = "";


GSMClient client;
GPRS gprs;
GSM gsmAccess;
GSMLocation location;

char server[] = "indy-bap-backend.herokuapp.com";
char path[] = "/api/locations";
int port = 80;

StaticJsonDocument<200> jsonBuffer;
HttpClient httpClient = HttpClient(client, server, port);
JsonObject root = jsonBuffer.to<JsonObject>();
String response;
int statusCode = 0;
String dataStr;

void setup() {
  // initialize serial communications and wait for port to open:
```

```
32    Serial.begin(9600);
33    while (!Serial) {
34      ; // wait for serial port to connect. Needed for native USB
            port only
35    }
36
37    Serial.println("Starting GSM connection to server!");
38    // connection state
39    boolean connected = false;
40
41    // After starting the modem with GSM.begin()
42    // attach the shield to the GPRS network with the APN, login and
        password
43    while (!connected) {
44      if ((gsmAccess.begin(PINNUMBER) == GSM_READY) &&
45      (gprs.attachGPRS(GPRS_APN, GPRS_LOGIN, GPRS_PASSWORD) ==
            GPRS_READY)) {
46        connected = true;
47        location.begin();
48      } else {
49        Serial.println("Not connected");
50        delay(1000);
51      }
52    }
53    Serial.println("connecting...");
54    // if you get a connection, report back via serial:
55    if (client.connect(server, port)) {
56      Serial.println("connected to server");
57    } else {
58      // if you didn't get a connection to the server:
59      Serial.println("connection to server failed");
60    }
61    if(!GPS.begin()){
62      Serial.println("Failed to initialize GPS!");
63      while(1);
64    }
65  }
66
67  void loop() {
68    unsigned long wait = millis();
69    bool gps = false;
70    while (millis() - wait < 2000 && !gps) {
71      if(GPS.available()){
72        Serial.println("GPS method");
73        root["longitude"] = GPS.longitude();
74        root["latitude"] = GPS.latitude();
75        root["altitude"] = GPS.altitude();
76        root["speed"] = GPS.speed();
77        root["satellites"] = GPS.satellites();
78        root["method"] = "GPS";
79        root["device"] = "poc";
80        gps=true;
81      }
82    }
83    if(!gps){
84      Serial.println("GPRS method");
```

```
85      unsigned long timeout = millis();
86      while (millis() - timeout < 2000) {
87        if (location.available() && location.accuracy() < 300 &&
              location.accuracy() != 0) {
88          root["longitude"] = String(location.longitude(),20);
89          root["latitude"] = String(location.latitude(),20);
90          root["altitude"] = String(location.altitude(),20);
91          root["method"] =  "GPRS";
92          root["device"] = "poc";
93        }
94      }
95    }
96    //if you get a connection, report back via serial:
97    if (client.connect(server, port)) {
98      postToServer(root);
99    } else {
100     // if you didn't get a connection to the server:
101     Serial.println("connection failed");
102   }
103   delay(3000); // Wait for 3 seconds to post again
104   // read the status code and body of the response
105   statusCode = httpClient.responseStatusCode();
106   response = httpClient.responseBody();
107
108   Serial.print("Status code: ");
109   Serial.println(statusCode);
110   Serial.print("Response: ");
111   Serial.println(response);
112
113 }
114
115 void postToServer(JsonObject data) {
116   dataStr = "";
117   serializeJson(data, dataStr);
118   Serial.println(dataStr);
119   httpClient.beginRequest();
120   httpClient.post(path);
121   httpClient.sendHeader("Content-Type", "application/json");
122   httpClient.sendHeader("Content-Length", dataStr.length());
123   httpClient.beginBody();
124   httpClient.print(dataStr);
125   httpClient.endRequest();
126 }
```

D. Broncode frontend

App.vue

```
<template>
  <div id="app" class="tile is-ancestor">
    <Maps class="tile is-6" />
    <Controls class="tile is-6" />
  </div>
</template>

<script>
import Maps from "./components/Maps";
import Controls from "./components/Controls";

export default {
  name: "App",
  components: {
    Maps,
    Controls
    }
  };
</script>

<style>
html,
body {
  padding: 0;
  margin: 0;
}
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
```

```
31    text-align: center;
32    height: 100vh;
33    width: 100vw;
34  }
35  </style>
```

main.js

```
1   import Vue from "vue";
2   import App from "./App.vue";
3   import store from "./store/index";
4   import * as VueGoogleMaps from "vue2-google-maps";
5
6   require("./styles/main.scss");
7
8   Vue.use(VueGoogleMaps, {
9     load: {
10    key: "AIzaSyAbvjSIFZ5jJqopl330QpfFtIDyWYTxhIg",
11    libraries: "places" // This is required if you use the
          Autocomplete plugin
12    // OR: libraries: 'places,drawing'
13    // OR: libraries: 'places,drawing,visualization'
14    // (as you require)
15
16    //// If you want to set the version, you can do so:
17    // v: '3.26',
18  }
19
20    //// If you intend to programmatically custom event listener code
21    //// (e.g. `this.$refs.gmap.$on('zoom_changed', someFunc)`)
22    //// instead of going through Vue templates (e.g. `<GmapMap
          @zoom_changed="someFunc">`)
23    //// you might need to turn this on.
24    // autobindAllEvents: false,
25
26    //// If you want to manually install components, e.g.
27    //// import {GmapMarker} from 'vue2-google-maps/src/components/
          marker'
28    //// Vue.component('GmapMarker', GmapMarker)
29    //// then disable the following:
30    // installComponents: true,
31  });
32  Vue.config.productionTip = false;
33
34  new Vue({
35    store,
36    render: h => h(App)
37  }).$mount("#app");
```

Maps.vue

```
1   <template>
2     <section class="container">
3       <GmapMap
4       v-if="getLocations.length>0"
5       ref="gmap"
```

```
6       :center="center"
7       :zoom="zoom"
8       map-type-id="terrain"
9       style="width: 80vw; height: 80vh;"
10      >
11        <GmapInfoWindow
12        :options="infoOptions"
13        :position="infoPosition"
14        :opened="infoOpened"
15        @closeclick="infoOpened=false"
16        >
17          <span v-html="infoContent" class="has-text-left"></span>
18        </GmapInfoWindow>
19        <div v-if="getShow==0 || getShow==2">
20          <div v-if="getShowMarkers">
21            <GmapMarker
22            :key="i"
23            v-for="(m, i) in getGSM"
24            :position="m"
25            :clickable="true"
26            @click="toggleInfo(m, i)"
27            icon="http://maps.google.com/mapfiles/ms/icons/red-dot.
                png"
28            />
29            </div>
30            <GmapPolyline :path="getGSM" v-bind:options="{
                strokeColor:'#FF0000'}"></GmapPolyline>
31          </div>
32          <div v-if="getShow==0 || getShow==1">
33            <div v-if="getShowMarkers">
34              <GmapMarker
35              :key="-i-1"
36              v-for="(m, i) in getPOC"
37              :position="m"
38              :clickable="true"
39              @click="toggleInfo(m, i)"
40              icon="http://maps.google.com/mapfiles/ms/icons/blue-dot
                  .png"
41              />
42            </div>
43          <GmapPolyline :path="getPOC" v-bind:options="{ strokeColor
              :'#0000FF'}"></GmapPolyline>
44        </div>
45      </GmapMap>
46      <div v-else>
47        <p>No locations yet!</p>
48      </div>
49    </section>
50  </template>
51
52  <script>
53  import { mapGetters, mapActions } from "vuex";
54  import { gmapApi } from "vue2-google-maps";
55  //import GmapCustomMarker from 'vue2-gmap-custom-marker';
56  export default {
57    mounted() {
```

```
58      this.createMap();
59    },
60    components: {},
61      data() {
62        return {
63          markers: [],
64          center: { lat: 10, lng: 10 },
65          zoom: 7,
66          infoPosition: null,
67          infoContent: null,
68          infoOpened: false,
69          infoCurrentKey: null,
70          infoOptions: {
71          pixelOffset: {
72          width: 0,
73          height: -35
74        }
75      },
76      flag: false
77      };
78    },
79    methods: {
80      ...mapActions(["$fetchLocations"]),
81      /**CALCULATE CENTER */
82      rad2degr(rad) {
83        return (rad * 180) / Math.PI;
84      },
85      /**CALCULATE CENTER */
86      degr2rad(degr) {
87        return (degr * Math.PI) / 180;
88      },
89      /**CALCULATE CENTER */
90      /**
91      * @param latLngInDeg array of arrays with latitude and
          longtitude
92      *   pairs in degrees. e.g. [[latitude1, longtitude1], [
          latitude2
93      *   [longtitude2] ...]
94      *
95      * @return array with the center latitude longtitude pairs in
96      *   degrees.
97      */
98      getLatLngCenter(latLngInDegr) {
99        let LATIDX = 0;
100       let LNGIDX = 1;
101       let sumX = 0;
102       let sumY = 0;
103       let sumZ = 0;
104       for (let i = 0; i < latLngInDegr.length; i++) {
105         let lat = this.degr2rad(latLngInDegr[i][LATIDX]);
106         let lng = this.degr2rad(latLngInDegr[i][LNGIDX]);
107         // sum of cartesian coordinates
108         sumX += Math.cos(lat) * Math.cos(lng);
109         sumY += Math.cos(lat) * Math.sin(lng);
110         sumZ += Math.sin(lat);
111       }
```

```
112       let avgX = sumX / latLngInDegr.length;
113       let avgY = sumY / latLngInDegr.length;
114       let avgZ = sumZ / latLngInDegr.length;
115       // convert average x, y, z coordinate to latitude and
              longtitude
116       let lng = Math.atan2(avgY, avgX);
117       let hyp = Math.sqrt(avgX * avgX + avgY * avgY);
118       let lat = Math.atan2(avgZ, hyp);
119       return { lat: this.rad2degr(lat), lng: this.rad2degr(lng) };
120     },
121     /**MARKER CLICK */
122     toggleInfo(m, i) {
123       this.infoPosition = m;
124       const info = `<p>Date: ${m.date.toLocaleString() ?? "unknown"
              } <br/>
125       Speed: ${m.speed ?? "unknown"} <br/>
126       Satellites: ${m.satellites ?? "unknown"} </p>`;
127       this.infoContent = info;
128       if (this.infoCurrentKey == i) {
129         this.infoOpened = !this.infoOpened;
130       } else {
131         this.infoOpened = true;
132         this.infoCurrentKey = i;
133       }
134     },
135     /** CALCULATE ZOOM */
136     calculateBounds(markers) {
137       let bounds = new this.google.maps.LatLngBounds();
138       markers.forEach(m => {
139         bounds.extend(m);
140       });
141       this.flag = true;
142       return bounds;
143     },
144     /** CALCULATE ZOOM */
145     getBoundsZoomLevel(bounds, mapDim) {
146     function latRad(lat) {
147       var sin = Math.sin((lat * Math.PI) / 180);
148       var radX2 = Math.log((1 + sin) / (1 - sin)) / 2;
149       return Math.max(Math.min(radX2, Math.PI), -Math.PI) / 2;
150     }
151     function zoom(mapPx, worldPx, fraction) {
152       return Math.floor(Math.log(mapPx / worldPx / fraction) / Math
              .LN2);
153     }
154     var WORLD_DIM = { height: 256, width: 256 };
155     var ZOOM_MAX = 21;
156     var ne = bounds.getNorthEast();
157     var sw = bounds.getSouthWest();
158     var latFraction = (latRad(ne.lat()) - latRad(sw.lat())) / Math.
          PI;
159     var lngDiff = ne.lng() - sw.lng();
160     var lngFraction = (lngDiff < 0 ? lngDiff + 360 : lngDiff) /
          360;
161     var latZoom = zoom(mapDim.height, WORLD_DIM.height, latFraction
          );
```

```
162     var lngZoom = zoom(mapDim.width, WORLD_DIM.width, lngFraction);
163     return Math.min(latZoom, lngZoom, ZOOM_MAX);
164     },
165     createMap() {
166     this.markers = this.getLocations.map(loc => {
167       return {
168         lat: Number(loc.latitude),
169         lng: Number(loc.longitude),
170         date: new Date(loc.createdAt),
171         speed: loc.speed,
172         satellites: loc.satellites,
173         device: loc.device
174       };
175     });
176     this.center = this.getLatLngCenter(this.markers.map(m => [m.lat
           , m.lng]));
177     this.zoom = this.getBoundsZoomLevel(this.calculateBounds(this.
           markers), {
178     height: 600,
179     width: 1000
180     });
181     }
182   },
183   computed: {
184     ...mapGetters(["getLocations", "getGSM", "getPOC","getShow","
           getShowMarkers"]),
185     google: gmapApi
186   },
187   watch: {
188     google() {
189     this.createMap();
190   },
191   getLocations() {
192     this.createMap();
193   }
194   },
195   created() {
196     this.$fetchLocations();
197     // window.setInterval(() => {
198     //   this.$fetchLocations();
199     // }, 5000);
200   }
201   };
202 </script>
203 <style scoped>
204 .container {
205   box-sizing: border-box;
206   margin: 20px;
207 }
208 </style>
```

Controls.vue

```
1 <template>
2   <section class="card">
3     <div class="tab">
```

```
4         <a class="poc-tab" @click="activeTab=0">POC tracker</a>
5         <a class="gsm-tab" @click="activeTab=1">GSM tracker</a>
6       </div>
7
8       <div class="padding-15" v-if="getPOC.length>0 && activeTab==0">
9         <div>
10          <div class="columns has-text-left">
11            <p class="heading column">[GPS/GPRS]:</p>
12            <p class="subtitle column">{{gpsMethod}}/{{gprsMethod}}</
                p>
13          </div>
14        </div>
15        <div>
16          <div class="columns has-text-left">
17            <p class="heading column">Total markers:</p>
18            <p class="subtitle column">{{totalMarkers}}</p>
19          </div>
20        </div>
21        <div>
22          <div class="columns has-text-left">
23            <p class="heading column">Recent marker:</p>
24            <p class="subtitle column">{{recentMarker}}</p>
25          </div>
26        </div>
27        <div>
28          <div class="columns has-text-left">
29            <p class="heading column">Max speed:</p>
30            <p class="subtitle column">{{maxSpeed.toFixed(2)}} km\h</
                p>
31          </div>
32        </div>
33        <div>
34          <div class="columns has-text-left">
35            <p class="heading column">Average speed:</p>
36            <p class="subtitle column">{{averageSpeed.toFixed(2)}} km
                \h</p>
37          </div>
38        </div>
39        <div>
40          <div class="columns has-text-left">
41            <p class="heading column">Min speed:</p>
42            <p class="subtitle column">{{minSpeed.toFixed(2)}} km\h</
                p>
43          </div>
44        </div>
45        <div>
46          <div class="columns has-text-left">
47            <p class="heading column">Max satellites:</p>
48            <p class="subtitle column">{{maxSatellites.toFixed(2)}}</
                p>
49          </div>
50        </div>
51        <div>
52          <div class="columns has-text-left">
53            <p class="heading column">Average satellites:</p>
54            <p class="subtitle column">{{averageSatellites.toFixed(2)
```

```
            }}</p>
55        </div>
56      </div>
57      <div>
58        <div class="columns has-text-left">
59          <p class="heading column">Min satellites:</p>
60          <p class="subtitle column">{{minSatellites.toFixed(2)}}</
              p>
61        </div>
62      </div>
63    <div class="buttons">
64    <button class="button is-small" @click="deleteLocations()">
        Clear history</button>
65    <button class="button is-small" @click="$toggleMarkers()">Hide
        all markers</button>
66    <button class="button is-small" @click="$setShow(0)">Show all
        devices</button>
67    <button class="button is-small" @click="$setShow(1)">Show only
        PoC</button>
68    <button class="button is-small" @click="$setShow(2)">Show only
        gsm</button>
69    </div>
70  </div>
71  <div v-else-if="activeTab==0">
72    <p>No data from the proof of concept. You may check gsm-tracker
        .</p>
73  </div>
74
75  <div class="padding-15" v-if="getGSM.length>0 && activeTab==1">
76    <div>
77      <div class="columns has-text-left">
78        <p class="heading column">Total markers:</p>
79        <p class="subtitle column">{{totalMarkersGSM}}</p>
80      </div>
81    </div>
82    <div>
83      <div class="columns has-text-left">
84        <p class="heading column">Recent marker:</p>
85        <p class="subtitle column">{{recentMarkerGSM}}</p>
86      </div>
87    </div>
88    <div>
89      <div class="columns has-text-left">
90        <p class="heading column">Max speed:</p>
91        <p class="subtitle column">{{maxSpeedGSM.toFixed(2)}} km\h
            </p>
92      </div>
93    </div>
94    <div>
95      <div class="columns has-text-left">
96        <p class="heading column">Average speed:</p>
97        <p class="subtitle column">{{averageSpeedGSM.toFixed(2)}}
            km\h</p>
98      </div>
99    </div>
100   <div>
```

```
101         <div class="columns has-text-left">
102           <p class="heading column">Min speed:</p>
103           <p class="subtitle column">{{minSpeedGSM.toFixed(2)}} km\h
                </p>
104         </div>
105       </div>
106       <div class="buttons">
107         <button class="button is-small" @click="deleteLocations()">
              Clear history</button>
108         <button class="button is-small" @click="$toggleMarkers()">
              Hide all markers</button>
109         <button class="button is-small" @click="$setShow(0)">Show all
               devices</button>
110         <button class="button is-small" @click="$setShow(1)">Show
              only PoC</button>
111         <button class="button is-small" @click="$setShow(2)">Show
              only gsm</button>
112       </div>
113     </div>
114     <div v-else-if="activeTab==1">
115       <p>No data from the gsm-tracker. You may check the proof of
            concept.</p>
116     </div>
117     </section>
118 </template>
119 
120 <script>
121 import { mapGetters, mapActions } from "vuex";
122 export default {
123   name: "Controls",
124   data() {
125     return {
126     activeTab: 0
127     };
128   },
129   methods: {
130     ...mapActions(["$deleteLocations", "$toggleMarkers", "$setShow"
            ]),
131     deleteLocations() {
132       let result = confirm("Are you sure you want to delete all
              locations?");
133       if (result) {
134         this.$deleteLocations();
135       }
136     }
137   },
138   computed: {
139     ...mapGetters(["getPOC", "getGSM"]),
140     gpsMethod() {
141       return this.getPOC.filter(l => l.method == "GPS").length;
142     },
143     gprsMethod() {
144       return this.getPOC.filter(l => l.method == "GPRS").length;
145     },
146     totalMarkers() {
147       return this.getPOC.length;
```

```
148     },
149     recentMarker() {
150       return new Date(
151         this.getPOC[this.getPOC.length - 1].date
152       ).toLocaleString();
153     },
154     maxSpeed() {
155       const values = [
156       ...this.getPOC
157       .filter(l => l.speed != undefined && l.speed != null && l.
              speed >= 0)
158       .map(l => Number(l.speed))
159       ];
160       if (values.length > 0) return Math.max(values) ?? 0;
161       else return 0;
162     },
163     averageSpeed() {
164       let values = [
165       ...this.getPOC
166       .filter(l => l.speed != undefined && l.speed != null && l.
              speed >= 0)
167       .map(l => Number(l.speed))
168       ];
169       if (values.length > 0) {
170         let sum = values.reduce((previous, current) => (current +=
                previous));
171         let avg = sum / values.length;
172         return avg ?? 0;
173       }
174       return 0;
175     },
176     minSpeed() {
177       const values = [
178       ...this.getPOC
179       .filter(l => l.speed != undefined && l.speed != null && l.
              speed >= 0)
180       .map(l => Number(l.speed))
181       ];
182       if (values.length > 0) return Math.min(values) ?? 0;
183       else return 0;
184     },
185     maxSatellites() {
186       const values = [
187       ...this.getPOC
188       .filter(
189       l =>
190       l.satellites != undefined &&
191       l.satellites != null &&
192       l.satellites >= 0
193       )
194       .map(l => Number(l.satellites))
195       ];
196       if (values.length > 0) return Math.max(values) ?? 0;
197       else return 0;
198     },
199     averageSatellites() {
```

```
200       const values = [
201       ...this.getPOC
202       .filter(
203       l =>
204       l.satellites != undefined &&
205       l.satellites != null &&
206       l.satellites >= 0
207       )
208       .map(l => Number(l.satellites))
209       ];
210       if (values.length > 0) {
211       let sum = values.reduce((previous, current) => (current +=
              previous));
212       let avg = sum / values.length;
213       return avg ?? 0;
214       }
215       return 0;
216     },
217     minSatellites() {
218       const values = [
219       ...this.getPOC
220       .filter(
221       l =>
222       l.satellites != undefined &&
223       l.satellites != null &&
224       l.satellites >= 0
225       )
226       .map(l => Number(l.satellites))
227       ];
228       if (values.length > 0) return Math.min(values) ?? 0;
229       else return 0;
230     },
231     totalMarkersGSM() {
232       return this.getGSM.length;
233     },
234     recentMarkerGSM() {
235       return new Date(
236       this.getGSM[this.getGSM.length - 1].date
237       ).toLocaleString();
238     },
239     maxSpeedGSM() {
240       const values = [
241       ...this.getGSM
242       .filter(l => l.speed != undefined && l.speed != null && l.
              speed >= 0)
243       .map(l => Number(l.speed))
244       ];
245       if (values.length > 0) return Math.max(values) ?? 0;
246       else return 0;
247     },
248     averageSpeedGSM() {
249       let values = [
250       ...this.getGSM
251       .filter(l => l.speed != undefined && l.speed != null && l.
              speed >= 0)
252       .map(l => Number(l.speed))
```

```
253         ];
254         let sum = values.reduce((previous, current) => (current +=
                previous));
255         let avg = sum / values.length;
256         return avg ?? 0;
257       },
258       minSpeedGSM() {
259         const values = [
260         ...this.getGSM
261         .filter(l => l.speed != undefined && l.speed != null && l.
                speed >= 0)
262         .map(l => Number(l.speed))
263         ];
264         if (values.length > 0) return Math.min(values) ?? 0;
265         else return 0;
266       }
267     }
268 };
269 </script>
270 <style scoped>
271 .card {
272 width: 80vw;
273 height: 80vh;
274 margin: 20px;
275 display: flex;
276 flex-direction: column;
277 }
278 .tab {
279 width: 80%;
280 margin: 0 auto;
281 }
282 .gsm-tab,
283 .poc-tab {
284 padding: 10px;
285 color: black;
286 font-weight: bold;
287 }
288 .gsm-tab:hover {
289 color: red;
290 }
291 .poc-tab:hover {
292 color: blue;
293 }
294 .padding-15 {
295 padding: 15px;
296 }
297 .columns {
298 display: flex;
299 flex-direction: row;
300 align-items: center;
301 }
302 .button {
303 margin-top: 20px;
304 }
305 .controls {
306 display: flex;
```

```
flex-direction: row;
flex-wrap: wrap;
}
</style>
```

Vuex index.js

```
import Vue from "vue";
import Vuex from "vuex";
//modules
import location from "./modules/location";

Vue.use(Vuex);

export default new Vuex.Store({
  modules: {
    location
  }
});
```

Vuex location.js

```
import axios from "axios";

const state = {
locations: [],
gsm: [],
poc: [],
show: 0, //0 = all, 1 = poc, 2 = gsm
showMarkers: true,
};

const getters = {
  getLocations: (state) => state.locations,
  getGSM: (state) => state.gsm,
  getPOC: (state) => state.poc,
  getShow: (state) => state.show,
  getShowMarkers: (state) => state.showMarkers,
};

const actions = {
  async $fetchLocations({ commit }) {
    const response = await axios.get(
    "https://indy-bap-backend.herokuapp.com/api/locations"
    );
    commit("SET_LOCATIONS", response.data.data);
  },

  async $deleteLocations({ commit }) {
    await axios.delete("https://indy-bap-backend.herokuapp.com/api/
        locations");
    commit("SET_LOCATIONS", []);
    },
    $setShow({ commit }, payload) {
    commit("SET_SHOW", payload);
    },
```

```
34      $toggleMarkers({ commit, state }) {
35      const bool = state.showMarkers;
36      commit("SET_SHOW_MARKERS", !bool);
37    },
38  };
39
40  const mutations = {
41  SET_LOCATIONS: (state, locs) => {
42    state.locations = locs;
43    state.gsm = locs
44    .filter((l) => l.device == "gsm")
45    .map((loc) => {
46    return {
47      lat: Number(loc.latitude),
48      lng: Number(loc.longitude),
49      date: new Date(loc.createdAt),
50      speed: loc.speed,
51      satellites: loc.satellites,
52      device: loc.device,
53    };
54    });
55    state.poc = locs
56    .filter((l) => l.device == "poc")
57    .map((loc) => {
58    return {
59      lat: Number(loc.latitude),
60      lng: Number(loc.longitude),
61      date: new Date(loc.createdAt),
62      speed: loc.speed,
63      satellites: loc.satellites,
64      device: loc.device,
65      method: loc.method,
66    };
67    });
68  },
69  SET_SHOW: (state, payload) => (state.show = payload),
70  SET_SHOW_MARKERS: (state, payload) => (state.showMarkers = payload)
      ,
71  };
72
73  export default {
74    state,
75    getters,
76    actions,
77    mutations,
78  };
```

E. Broncode backend

server.js

```
const express = require("express");
const dotenv = require("dotenv");
const colors = require("colors");
const morgan = require("morgan");
const connectDB = require("./config/db");
const locations = require("./routes/locations");
const path = require("path");

// PORT
const PORT = process.env.PORT || 5000;

// CONFIGURE DOTENV, WHICH FILE TO USE
dotenv.config({ path: "./config/config.env" });
// START CONNECTION WITH MONGODB ATLAS
connectDB();

const app = express();
// USE JSON TO PARSE BODIES
app.use(express.json());

if (process.env.NODE_ENV === "development") {
app.use(morgan("dev"));
}

// ADDING ROUTES
app.use("/api/locations", locations);

app.listen(
PORT,
console.log(
```

```
31 `Server running in ${process.env.NODE_ENV} mode on port ${PORT}`.
     yellow.bold
32 )
33 );
```

/controllers/locations.js

```
1  const Location = require("../models/Location");
2  // @desc    Get all locations
3  // @route   GET /api/locations
4  // @access  Public
5  exports.getLocations = async (req, res, next) => {
6  try {
7    const locations = await Location.find();
8    return res.status(200).json({
9      success: true,
10     count: locations.length,
11     data: locations
12   });
13 } catch (error) {
14   return res.status(500).json({
15     success: false,
16     error: "Server error"
17   });
18 }
19 };
20
21 // @desc    Add location
22 // @route   POST /api/locations
23 // @access  Public
24 exports.addLocation = async (req, res, next) => {
25   try {
26     const location = await Location.create(req.body);
27     return res.status(201).json({
28       success: true,
29       data: location
30     });
31   } catch (err) {
32     if (err.name == "ValidationError") {
33       const messages = Object.values(err.errors).map(val => val.
           message);
34       return res.status(400).json({
35         success: false,
36         error: messages
37       });
38     } else {
39       return res.status(500).json({
40         success: false,
41         error: "Server error"
42         });
43     }
44   }
45 };
46
47 // @desc    Delete all locations
48 // @route   DELETE /api/locations
```

```
// @access  Public
exports.deleteLocations = async (req, res, next) => {
  try {
    await Location.deleteMany({});
      return res.status(200).json({
      success: true,
      data: {}
    });
  } catch (error) {
    return res.status(500).json({
      success: false,
      error: "Server error"
    });
  }
};
```

/models/Location.js

```
const mongoose = require("mongoose");

const LocationSchema = new mongoose.Schema({
  longitude: {
    type: String,
    trim: true
  },
  latitude: {
    type: String,
    trim: true
  },
  altitude: {
    type: String,
    trim: true
  },
  createdAt: {
    type: Date,
    default: Date.now
  }
});

module.exports = mongoose.model("Location", LocationSchema);
```

/routes/locations.js

```
const express = require("express");
const router = express.Router();
const {
getLocations,
addLocation,
deleteLocations
} = require("../controllers/locations");

router
.route("/")
.get(getLocations)
.post(addLocation)
.delete(deleteLocations);
```

```
14
15 module.exports = router;
```

F. Broncode mobiele applicatie

app.js

```
1  import Vue from 'nativescript-vue';
2
3  import HelloWorld from './components/HelloWorld';
4
5  // Uncommment the following to see NativeScript-Vue output logs
6  // Vue.config.silent = false;
7
8  new Vue({
9
10 render: h => h('frame', [h(HelloWorld)])
11
12 }).$start();
```

app.scss

```
1  /*
2  In NativeScript, the app.css file is where you place CSS rules that
3  you would like to apply to your entire application. Check out
4  http://docs.nativescript.org/ui/styling for a full list of the CSS
5  selectors and properties you can use to style UI components.
6  /*
7  In many cases you may want to use the NativeScript core theme
       instead
8  of writing your own CSS rules. For a full list of class names in
       the theme
9  refer to http://docs.nativescript.org/ui/theme.
10 The imported CSS rules must precede all other types of rules.
11 */
12 @import '~nativescript-theme-core/css/core.light.css';
13
```

```
.home-panel {
  font-size: 20;
  margin: 15;
}

.description-label {
  margin-bottom: 15;
}

.lbl {
  margin: 10;
}

.btn {
  font-size: 20;
  margin: 20;
}
```

```
<template>
  <Page class="page">
    <ActionBar title="Geolocation" class="action-bar" />
    <ScrollView>
      <StackLayout class="home-panel">
        <Image src="~/images/map-marker-icon.png" height="60" />
        <Button
        :text="active==true?'Stop tracking':'Start tracking'"
        @tap="start"
        class="btn btn-primary"
        />

        <Label :text="'Latitude: ' + latitude" class="lbl" />
        <Label :text="'Longitude: ' + longitude" class="lbl" />
        <Label :text="'Altitude: '+altitude + 'm'" class="lbl" />
        <Label :text="'Speed: ' + speed + 'km/h'" class="lbl" />
      </StackLayout>
    </ScrollView>
  </Page>
</template>

<script>
const geolocation = require("nativescript-geolocation");
const { Accuracy } = require("tns-core-modules/ui/enums");
import axios from "axios/dist/axios";
export default {
  data() {
    return {
      latitude: "",
      longitude: "",
      speed: "",
      altitude: "",
      device: "gsm",
      active: false
    };
  },
  methods: {
  start() {
```

```
39      this.active = !this.active;
40      this.track();
41    },
42    track() {
43      if (this.active) {
44        const sec = 3;
45        this.getLocation().then(res => {
46          this.latitude = res.latitude;
47          this.longitude = res.longitude;
48          this.speed = res.speed * 3.6;
49          this.altitude = res.altitude;
50          const payload = {
51            latitude: this.latitude,
52            longitude: this.longitude,
53            speed: this.speed,
54            altitude: this.altitude,
55            device: this.device
56          };
57          axios
58          .post(
59          "https://indy-bap-backend.herokuapp.com/api/locations",
60          payload
61          )
62          .then(() => {
63            console.log("succes");
64          });
65        });
66        setTimeout(this.track, sec * 1000);
67      } else {
68        (this.latitude = ""),
69        (this.longitude = ""),
70        (this.speed = ""),
71        (this.altitude = "");
72      }
73    },
74    getLocation() {
75      return geolocation.getCurrentLocation({
76        desiredAccuracy: Accuracy.High,
77        timeout: 3000,
78        updateTime: 2000,
79        minimumUpdateTime: 3000
80        });
81      }
82    },
83    mounted() {
84      geolocation.enableLocationRequest();
85    }
86  };
87  </script>
88
89  <style scoped>
90  .home-panel {
91  vertical-align: center;
92  font-size: 20;
93  margin: 15;
94  }
```

```
95 .description-label {
96 margin-bottom: 15;
97 }
98 </style>
```

Bibliografie

Aidan-Phaswana. (g.d.). De kosmische ruimte: de ruimtevaart en haar toepassingen. Verkregen van https://quizlet.com/393158527/de-kosmische-ruimte-de-ruimtevaart-en-haar-toepassingen-flash-cards/

aipprecision. (2019). Advantages of Polymer Components over Metallic Materials for Machined Parts. *Advantages of Polymer Components over Metallic Materials for Machined Parts*.

Arduino. (g.d.). ARDUINO MKR GSM 1400 CELLULAR KIT. Verkregen van https://store.arduino.cc/arduino-sim-mkr-gsm-1400-cellular-kit-1417?fbclid=IwAR0kJk6t6PVON-YakV_EiSOnb5y2RgBRQW0c6pVmpRw-hJlPRHO99qDdjSA

CellTrack. (g.d.). Plaatsbepaling via GSM. Verkregen van https://www.celltrack.eu/telecom-info/gsm-locatie-tracking

Chivers, M. (g.d.). Differential GPS Explained. *esri*.

Christiaens, K. (g.d.). Wat is een satelliet? Verkregen van https://www.spacepage.be/artikelen/ruimtevaart/algemene-info/wat-is-een-satelliet

EENA. (2016, maart 2). Advanced Mobile Location (AML) Specifications & Requirements.

ESA. (g.d.). Cebreros ground station. Verkregen van https://www.esa.int/About_Us/ESAC/Cebreros_ground_station

gisgeography. (2016, november 12). Trilateration vs Triangulation – How GPS Receivers Work. Verkregen van https://gisgeography.com/trilateration-triangulation-gps/

GMV. (2011). GPS Performances. *gssc.esa.int*.

Grimes, J. G. (2007). *GLOBAL POSITIONING SYSTEM PRECISE POSITIONING SERVICE PERFORMANCE STANDARD* (onderzoeksrap., Department Of Defense).

Grimes, J. G. (2008a). *GLOBAL POSITIONING SYSTEM STANDARD POSITIONING SERVICE PERFORMANCE STANDARD* (onderzoeksrap., Department Of Defense).

Grimes, J. G. (2008b, september 1). *GLOBAL POSITIONING SYSTEM STANDARD POSITIONING SERVICE PERFORMANCE STANDARD*. 6000 defense Pentagon.

Gunnar Taraldsen, T. B., Tor Arne Reinen. (2011). The underwater GPS problem. *The underwater GPS problem*.

Holst, A. (2019). Number of smartphone users worldwide 2016-2020. *statista.com*.

Ian Nicholson, C. M. (2018, maart 30). The Best Avalanche Beacons. Verkregen van https://www.outdoorgearlab.com/topics/snow-sports/best-avalanche-beacon

Jürg Schweizer, G. K. (2003). Testing the performance of avalanche transceivers. *Elsevier*.

Meyer, R. (2016). GPS Doesn't Work Underwater. *theatlantic.com*.

National Coordination Office for Space-Based Positioning, N. & Timing. (2020, april 22). Other Global Navigation Satellite Systems (GNSS). Verkregen van https://www.gps.gov/systems/gnss/

Nikhilesh, J., Poonam, S., Kate, T. & Pooja, S. (2019). A-GPS vs GPS. *diffen.com*.

Restivo, F. (2004, januari 1). Components of a DGPS System. Verkregen van https://www.researchgate.net/figure/Components-of-a-DGPS-System_fig1_252064818

Rubino, D. (2009, januari 3). GPS vs. aGPS: A Quick Tutorial. Verkregen van https://www.windowscentral.com/gps-vs-agps-quick-tutorial

Samuelson, A. (2020, februari 20). What is an Atomic Clock? Verkregen van https://www.nasa.gov/feature/jpl/what-is-an-atomic-clock

Schleppe, J. B. & Lachapelle, G. (2006). *GPS Tracking Performance under Avalanche Deposited Snow* (proefschrift, University of Calgary).

SpaceNews. (2014, mei 12). GPS 2F-6 Navigation Satellite Slated To Launch on May 15. Verkregen van https://spacenews.com/40530gps-2f-6-navigation-satellite-slated-to-launch-on-may-15/

Taraldsen, G., Reinen, T. A. & Berg, T. (2011, augustus 30). *The underwater GPS problem* (tech. rap.).