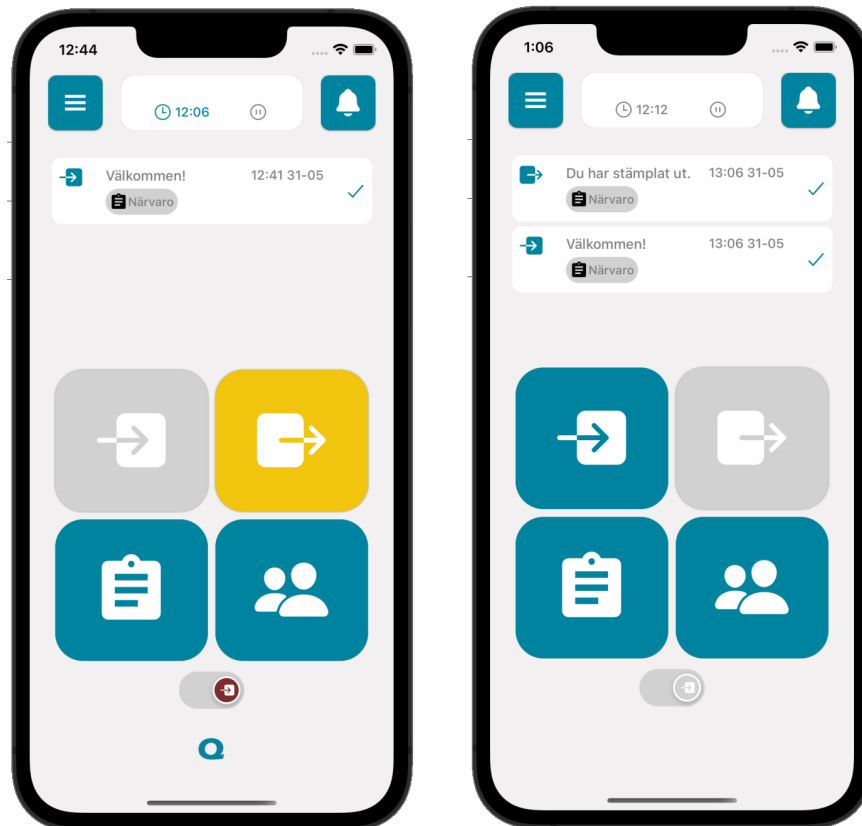




CHALMERS



Lokaliseringsteknologier för en automatiserad tidrapporteringsapplikation.

Tillämpning av haversine formel, RSSI Distance Model och lämplighet för automation.

Examensarbete inom Data- och Informationsteknik

Erik Kieu

Dang Tuan Phong Nguyen

Institutionen för Data- och Informationsteknik

CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg, Sverige 2022
www.chalmers.se

EXAMENSARBETE 2022

Lokaliseringsteknologier för en automatiserad tidrapporterings applikation.

Tillämpning av Haversine formel, RSSI Distance Model och lämplighet för automation.

Erik Kieu
Dang Tuan Phong Nguyen



CHALMERS

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg, Sverige 2022

Lokaliseringsteknologier för en automatiserad tidrapporterings applikation.
Tillämpning av Haversine formel, RSSI Distance Model och lämplighet för automation.
Erik Kieu
Dang Tuan Phong Nguyen

© Erik Kieu, 2022. © Dang Tuan Phong Nguyen, 2022.

Handledare: Jonas Almström Duregård, Institutionen för data- och information- teknik
Examinator: Lars Svensson, Institutionen för fysik
Företagshandledare: Dan Rozen, QBIS Business Systems

Examensarbete 2022
Institutionen för Data- och Informationsteknik
Chalmers Tekniska Högskola / Göteborgs Universitet
412 96 Göteborg
Telefon: 031-772 1000

Omslag: *Skärmdump på tillstånden in- och utstämplad i applikationen*

Institutionen för Data- och Informationsteknik
Göteborg 2022

Sammanfattning

Många företag hanterar idag tidrapporteringen manuellt vilket kan anses vara omodernt. QBIS är ett tidrapporteringsföretag som vill hitta en innovativ lösning för att automatisera detta. Målet med projektet är att tillsammans med QBIS skapa en mobilapplikation som hanterar olika uppgifter automatiskt med bland annat BLE, Wi-Fi och GPS. Syftet med rapporten är att avgöra vilken av dessa tre teknologier är bäst och lämpligheten av automationsnivå för olika uppgifter.

Utifrån resultatet ansågs BLE fungera bäst vid korta distanser och GPS vid längre distanser. Automationen för tidrapporteringen anses vara översiktligt villkorligt automatiskt vilket också är lämpligt för mobilapplikationen.

Keywords: React-Native, BLE, Wi-Fi, GPS, Automation, Haversine, RSSI

Abstract

Many companies today handle time reporting manually which can be considered outdated. The company QBIS which specializes in time reporting wants to find an inovative solution to automate this chore. The goal of this project is to work together with QBIS inorder to create a mobileapplication which can automate tasks related to time reporting with the technologies BLE, Wi-Fi, and GPS. The purpose with this report is to determine which of these three technologies is the best and a suitable level of automation to handle different tasks.

The results show that BLE is better at shorter distances and GPS is better at longer distances. In general the automated tasks are considered to be conditionally automatic which is also suitable for this application.

Förord

Vi vill tacka till vår handledare Jonas Duregård för hans hjälp och rådgivning under projektets gång. Samt våra företagshandledare Dan Rozen som projektledare och Martin Augustsson för hans expertis.

Erik Kieu & Dang Tuan Phong Nguyen, Göteborg, June 13, 2022

Ordlista

BLE	Bluetooth Low Energy
GPS	Global Positioning System
npm	Node Package Manager
RSSI	Received Signal Strength Indicator
dBm	Decibel-milliwatts
MVP	Minimum Viable Product

Innehållsförteckning

1	Inledning	1
1.1	Bakgrund	1
1.2	Syfte	1
1.3	Frågeställning	1
1.4	Mål	1
1.5	Avgränsningar	1
2	Teknisk bakgrund	2
2.1	Geografiska lokaliseringsteknologier	2
2.2	Automation	3
2.3	Mobilapplikation	4
3	Metod	6
3.1	Testupplägg	6
3.2	Automation	6
4	Systemkonstruktion	8
4.1	Lokaliseringsprotokoll	8
4.2	Automationslogik	11
5	Resultat	13
5.1	Uppmätningar på Router, BLE och GPS	13
5.2	Jämförelse mellan BLE, GPS, Wi-Fi	15
5.3	Automatiserade handlingar	16
6	Diskussion	17
6.1	Metodkritik	17
6.2	Pålitligheten av positionering med BLE, Wi-Fi, GPS	17
6.3	Automation för handlingarna	17
6.4	Etik	18
7	Slutsats	19
	Källor	20
A	Källkod - Protokoll logik	I
B	Felmarginal för individuella resultat	III

1 Inledning

I denna sektion redogörs bakgrunden och syfte för rapporten. Här presenteras även undersökningsområdet och målet med projektet.

1.1 Bakgrund

Något som är problematiskt för många företag idag är hur tidrapportering hanteras där manuell inmatning är en gemensam trend vilket kan vara tidskrävande och krångligt. QBIS är ett av Sveriges ledande företag inom tidrapportering och vill fortsätta förbättra hur rapporteringen hanteras i olika företag genom att utveckla en mobilapplikation. Tanken med mobilapplikationen är att underlätta tidrapportering genom att introducera automation för bland annat instämpling. Målgruppen som QBIS riktar sig in på är framförallt arbetsplatser med inomhusmiljöer.

1.2 Syfte

Syftet med projektet är att utreda noggrannheten av automatisk tidrapportering med hjälp av RSSI och geografiska koordinater, samt undersöka lämpligheten av automationen.

1.3 Frågeställning

Frågor och undersökningar som kommer att behandlas är följande:

- Hur kommer rangordningen se ut för en optimal sökning av användarens position utifrån teknologierna BLE, Wi-Fi och GPS?
- Vad är en lämplig grad av automation för mobilapplikationen utifrån uppgifterna:
 - Inloggning
 - Instämpling av arbetsplats
 - Gå på rast
 - Återuppta arbetspass
 - Hantering av bytet mellan BLE, Wi-Fi och GPS
 - Utstämpling av arbetsplats

1.4 Mål

Målet med projektet är att utveckla en MVP i form av en mobilapplikation för att besvara frågeställningarna. Applikationen ska ersätta manuell tidrapportering och istället använda sig av en automatiserad lösning med hjälp av dem tekniska verktygen BLE, Wi-Fi och GPS. Mobilapplikationen ska vara kompatibel med operativsystemen Android och iOS.

1.5 Avgränsningar

- Undersökningarna kommer inte utföras med mobiler av operativsystemet iOS.
- GPS testas inte till full utsträckning gentemot BLE och Wi-Fi.

2 Teknisk bakgrund

I denna sektion beskrivs den tekniska bakgrunden i tre delar som behandlar de geografiska lokaliseringsteknologierna, automationen och mobilapplikationen.

2.1 Geografiska lokaliseringsteknologier

Denna sektion beskriver de olika geografiska lokaliseringsteknologierna och ekvationer som använts för att utreda frågeställningen ”Hur kommer rangordningen se ut för en optimal sökning av användarens position utifrån teknologierna BLE, Wi-Fi och GPS?”

BLE

BLE är en teknologi som kompletterar Classic Bluetooth radio som tillhandahåller trådlös anslutning mellan olika elektroniska enheter[1]. Till skillnad från Classic Bluetooth som är ensidig i anslutningsmodellen eftersom data endast skickas till en användare i taget. Så kan BLE utsända information till flera användare samtidigt. Dessutom är energikonsumtionen lägre för BLE-enheter jämfört med classic bluetooth. En stor anledning är överföringshastigheten som har sänkts för att optimera energiförbrukningen[2].

Bluetooth beacon

BLE teknologin har utvidgas till större användningsområden inom positionering. Bluetooth beacon är en enhet som kan annonsera data till andra enheter som har bluetooth aktiverade och befinner sig inom räckvidd[3]. En bluetooth beacon har i genomsnitt en räckvidd upp till 100 meter och har en pålitlig dataöverföring upp till 30 meter. Bluetooth beacon har som de flesta elektroniska hårdvaror en MAC-adress för identifiering[4].

Wi-Fi

Wi-Fi teknologin är ett nätverks protokoll som ansluter trådlöst kompatibla elektroniska-enheter till ett nätverk. Wi-Fi fungerar i varierande frekvenser och har olika räckvidd beroende på Wi-Fi enhetens specifikationer. Wi-Fi är själva radiosignalen som skickas till en enhet vilket tolkar datan. För anslutning skickas en radiosignal tillbaka med konfirmation om anslutningen[5].

Trådlös router

Trådlös router är hårdvara som är relativt vanlig i ett hushåll. Användningsområdet för trådlösa routrar är oftast för att koppla upp sig till sitt hushållsnätverk. En genomsnittlig räckvidd för en router är 45 till 90 meter i inomhusmiljö[6][7].

GPS

GPS är ett satellitnavigeringssystem som används för positionering. Systemet använder tekniken trilateration som med hjälp av flera satelliter kan bestämma en noggrann positionering.[8][9].

Avstånds uppskattning

Haversine-formeln är en ekvation som beräknar avståndet mellan två koordinater bestående av latitud och longitud på ett klot[10].

$$d = 2r \sin^{-1} \left(\sqrt{\sin^2 \left(\frac{a_1 - a_2}{2} \right) + \cos(a_2) \cos(a_1) \sin^2 \left(\frac{o_1 - o_2}{2} \right)} \right) \quad (1)$$

där:

$a_{1,2}$ är latitud koordinater för från och till

$o_{1,2}$ är longitud koordinater för från och till

r är klotets radie (km)

d är avståndet mellan koordinaterna (km)

Genom att tillämpa formeln och med hjälp av grafteori har flera projekt även kunnat uppskatta den kortaste vägen på en kart-vy[10]. Grafteori används vanligen för att modellera relationer mellan noder och kanter, där noderna och kanterna värderas beroende på uppmätningen[11].

RSSI Distance Model är en metod för enheter som har egenskapen RSSI. Metoden används framförallt för att uppskatta avståndet mellan två enheter, där signal styrkan relateras beroende på avståndet[12]. I en tidigare forskning[13] nämner skribenterna att formeln brister som indikator för att uppskatta avstånd. En viktig anledning till detta är just den instabila signalstyrkan.

$$RSSI = -10n \log(d) + A \quad (2)$$

$$d = 10^{\left(\frac{A-RSSI}{10n}\right)} \quad (3)$$

där:

A är den uppmätta signalstyrkan en meter mellan två enheter (dBm)

$RSSI$ är den uppmätta signal styrkan mellan två enheter (dBm)

n är en konstant som varierar beroende på hinder (fri yta uppskattas som 2)

d är avståndet mellan enheterna (m)

2.2 Automation

Många system är idag automatiserade och för att redogöra automationen kan en sammanställning av automationsnivåer nyttjas som ett alternativ för en överblick av systemet. Exempelvis när man talar om automation i självkörande bilar delas automationen in i olika nivåer från 0-5 som definierats enligt The Society of Automotive Engineers[14]. Dessa nivåer sträcker sig från ingen automation (nivå 0) till full automation (nivå 5). I tabell 2.1 presenteras en summerad beskrivning på de olika automationsnivåerna.

Automationsnivå	Förklaring	Subjekt för körande
Nivå 0 (Ingen automation)	Föraren hanterar alla operationer.	Förare
Nivå 1 (Förar assistans)	Under särskilda omständigheter kan systemet delvis hantera handlingarna bromsa, styra eller acceleration.	Förare
Nivå 2 (Delvis Assistans)	Under särskilda omständigheter hanterar systemet handlingarna bromsa, styra och acceleration	Förare
Nivå 3 (Villkorlig automation)	Under särskilda omständigheter hanterar systemet alla operationer och föraren behöver endast ingripa när system slutar fungera	Förare + System
Nivå 4 (Hög automation)	Under särskilda omständigheter hanterar systemet alla operationer.	System
Nivå 5 (Full automation)	Systemet hanterar alla operationer villkorslöst.	System

Tabell 2.1: Automationsnivåerna för bilar

2.3 Mobilapplikation

Denna sektion beskriver de olika resurserna som använts för att utveckla en MVP i form av en mobilapplikationen.

React Native

React Native är ett ramverk som används till att utveckla operativspecifika mobilapplikationer med användargränssnitt för Android och iOS. Den är baserad på Facebooks Javascript ramverk React.[15].

Node JS

Node JS är ett JavaScript-verktyg som används till att utveckla skalbara nätverks applikationer genom att utgå från en event-driven och icke-blockerande modell[16].

npm

npm är en online-plattform där utvecklare kan dela och hämta ned paket bestående av kod som är färdig för att användas i applikationer. Plattformen består av huvudsakligen två delar: en webbsida för att hitta paket och ett kommandoradsverktyg som hanterar installationen av paketet[17].

Relevanta paket

Relevanta paket som används i mobilapplikationen är följande:

react-native-ble-plx: Detta paket tillåter bland annat att applikationen får tillgång till BLE-enheter som upphittats av mobilen[18].

react-native-wifi-reborn: Detta paket tillåter att applikationen får tillgång till mobilens Wi-fi information[19].

react-native-geolocation-service: Detta paket tillåter att applikationen får tillgång till mobilens GPS koordinater[20].

react-native-background-actions: Detta paket tillåter att applikationen körs i bakgrunden[21].

react-native-async-storage: Detta paket möjliggör lokal lagring av information[22].

3 Metod

Denna sektion beskriver om vilka hårdvaror som nyttjades för projektet och hur testerna utfördes för att besvara frågeställningarna.

3.1 Testupplägg

Vid utföranden av testerna användes RadBeacon Dot som är en bluetooth beacon, Belkin Wireless N300 Router som är en router, Samsung Galaxy S10 för att använda mobilapplikationen och One Plus 5 för att jämföra uppmätningar av RadBeacon Dot. För observering och notering av undersökningarna användes konsolen i programmeringsmiljön Visual Studio Code.

Max-distans utan hinder

Uppmätningar av RSSI undersöktes i en utomhusmiljö från en utgångspunkt där bluetooth-enheten och routern placerades. Då ingen signal mellan enheterna och mobil-enheten kunde upptäckas mättes avståndet fysiskt från utgångspunkten och positionen på mobil-enheten.

Max-distans med hinder

Uppmätningar av RSSI undersöktes i en inomhusmiljö från en utgångspunkt där bluetooth-enheten och routern placerades. För att representera hinder nyttjades ett flertal rum med gipsväggar. Varje hinder var dubbla gipsväggar som estimerades till 1cm i bredd och 13cm i mellanrum. Avståndet från utgångspunkten mättes fysiskt då ingen signal mellan enheterna och mobil-enheten kunde upptäckas.

Träffsäkerhet

Under testerna mättes fem olika stäckor från en utgångspunkt där bluetooth-enheten och routern placerades. Mobil-enheten placerades på de fem avstånden för att fånga upp RSSI-signalerna för respektive enhet och koordinaterna. Uppmätningarna observerades i flera iterationer för att få genomsnittliga värden som gynnade avståndsuppskattningen när *Haversine* och *RSSI Distance Model* tillämpades.

3.2 Automation

För att undersöka automationen av applikationen anpassades teorin om automationsnivåerna enligt The Society of Automotive Engineer till en mobilapplikation. Automationen av flera handlingar observerades enligt den modifierade tabellen 3.1. Detta skedde i en kontrollerad miljö där mobilapplikationen hade tillgång till platstjänster och kunde bearbetas i förgrunden såväl som i bakgrunden.

Automations-nivå	Förklaring	Subjekt för applikations användning
Nivå 0 (Ingen automation)	Användaren hanterar alla handlingar.	Användare
Nivå 1 (Användar assistans)	Under särskilda omständigheter kan systemet delvis hantera handlingarna med hjälp av användarens inmatning.	Användare
Nivå 2 (Villkorlig automation)	Under särskilda omständigheter hanterar systemet alla handlingar och användaren behöver endast ingripa när system slutar fungera	Användare + System
Nivå 3 (Full automation)	Systemet hanterar alla handlingar villkorslöst.	System

Tabell 3.1: Automationsnivåerna för mobilapplikationen.

4 Systemkonstruktion

Denna sektion beskriver hur logiken för teknologierna BLE, Wi-Fi och GPS implementerades. För en detaljerad källkod av teknologierna hänvisas till appendix A. Sektionen behandlar även implementationen för de situationer som kan uppstå för användaren i mobilapplikationen.

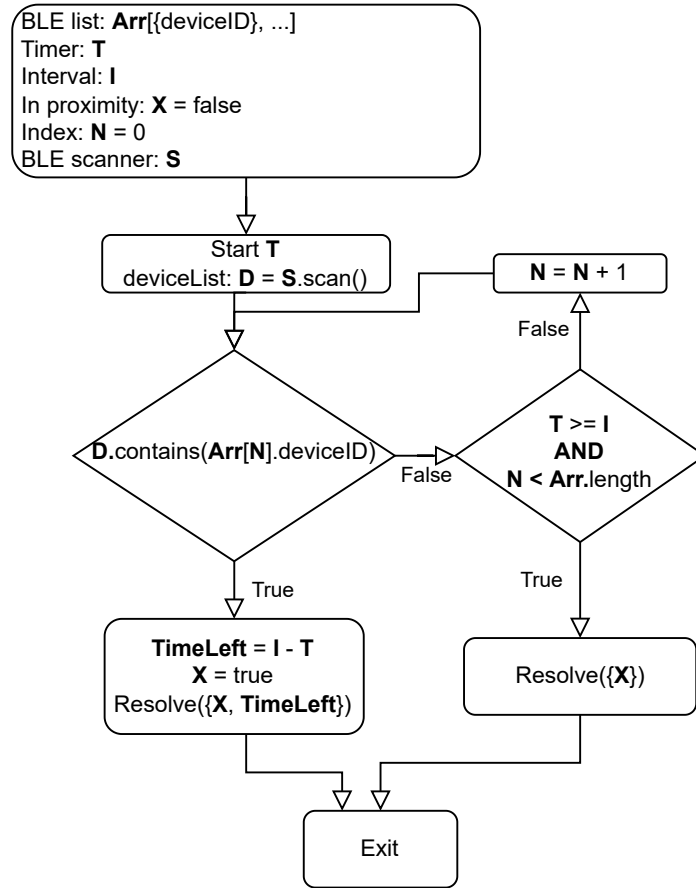
4.1 Lokaliseringsprotokoll

Implementeringen på de olika protokollen är konstruerad så att varje protokoll itererar så länge mobilapplikationen är inom räckhåll för respektive protokoll. Vid fallen då mobilapplikationen inte är inom räckhåll avbryts iterationen för protokollet.

BLE-protokoll

För logiken bakom BLE-protokollet används BleManager från paketet react-native-ble-plx. BleManager har som uppgift att söka BLE-enheter i närheten av mobil-enheten som är i bruk. De uppgifter från BLE-enheten som nyttjas är RSSI-signalen och MAC-adressen för att upptäcka matchande BLE-enhet.

I figur 4.1 används T som en tidsgräns på protokollet, Arr är en lista bestående av giltiga BLE-enheter för användaren och D är en lista bestående av sökta BLE-enheter. Ifall listorna har åtminstone en matchande enhet (contains) innan T sätts X till true, vilket indikerar att BLE-enheten är i närheten (resolve).

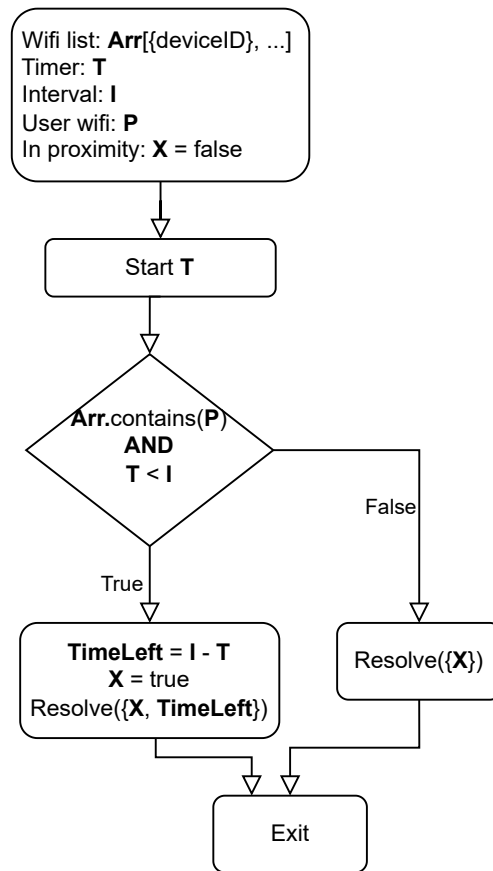


Figur 4.1: Flödesschema för BLE-protokoll.

Wi-Fi protokoll

Wi-Fi protokollet använder WifiManager från paketet react-native-wifi-reborn. WifiManager hanterar mobilens Wi-Fi konfigurationer och förser mobilapplikationen med namnet samt RSSI för Wi-Fi enheten som mobil-enheten är ansluten till.

Figur 4.2 illustrerar flödesschemat på Wi-Fi protokollet där P är den anslutna Wi-Fi enheten och Arr en lista på giltiga Wi-Fi enheterna för användaren. Ifall Arr innehåller namnet på Wi-Fi enheten (contains) P innan tidsgränsen T har utgått så tilldelas X till true, vilket indikerar att Wi-Fi enheten är i närheten (resolve).

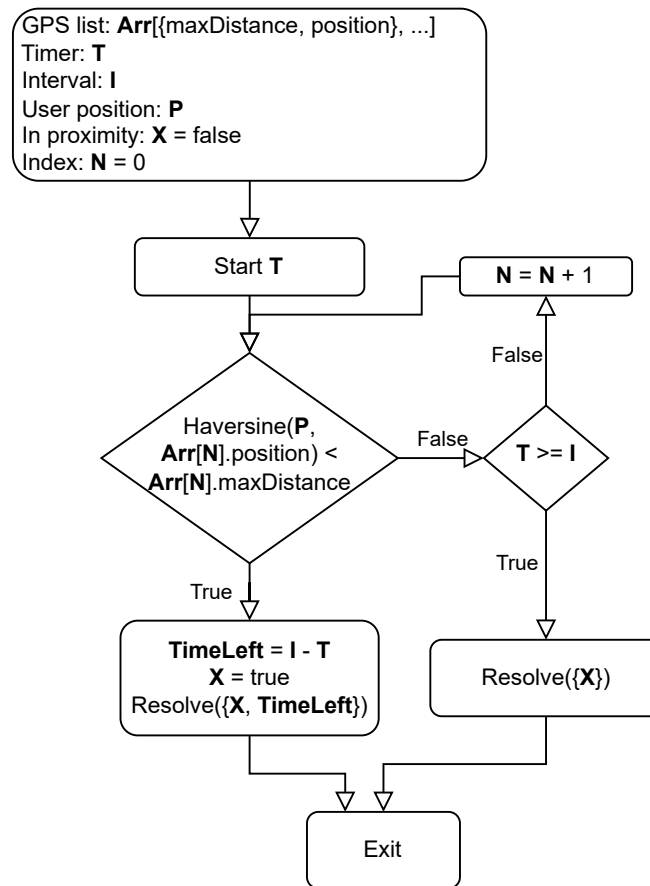


Figur 4.2: Flödesschema för Wi-Fi protokoll.

GPS-protokoll

Geolocation från paketet react-native-geolocation-service används för att uppskatta koordinaterna för mobil-enheten som är i bruk.

I Figur 4.3 används *Haversine* med användarens koordinater P och koordinater för varje GPS-objekt $Arr[N]$ i GPS-listan Arr . Varje GPS-objekt har även ett max-avstånd som användaren får vara ifrån. Ifall avståndet mellan användaren och åtminstone ett utav GPS-koordinaterna i $Arr[N]$ är mindre än max-avståndet samt tidsgränsen T inte har utgått, då kommer X tilldelas till true, vilket indikerar att mobilapplikationen är i närheten (resolve).



Figur 4.3: Flödesschema för GPS protokoll.

4.2 Automationslogik

Inloggningssystemet fungerar genom att manuellt mata in användarnamnet samt lösenordet för en användare. Användaren har även möjlighet att hålla sig inloggad och spara användarinformationen med *session*, som sparas lokalt i användarens mobil med paketet *react-native-async-storage*.

När användaren navigerar i mobilapplikationen till huvudsidan aktiveras instämplingsprocessen. Ifall något av protokollen lyckas lokalisera åtminstone en av enheterna som är giltiga för användaren initieras en bakgrundsuppgift för instämpling. Bakgrundsuppgiften implementerades med paketet *react-native-background-actions*. Användaren har även möjlighet att stämpla in manuellt ifall protokollen misslyckas med lokaliseringen.

Antalet protokoll som ska köras i bakgrundsuppgiften bestäms ifall det finns giltiga enheter eller koordinater för användaren och tilldelas som första, andra och tredje protokoll. Varje protokoll körs inom ett intervall och repeteras ifall användaren fortfarande befinner sig inom räckhåll. När första protokollet misslyckas övergår mobilapplikationen till nästa protokoll.

När det sista protokollet misslyckas stämplas användaren slutligen ut oberoende av ifall mobilapplikationen är i förgrunden eller bakgrunden. Under hela bakgrundsuppgiften har användaren även möjlighet att avbryta instämplingen med en utstämplingsknapp.

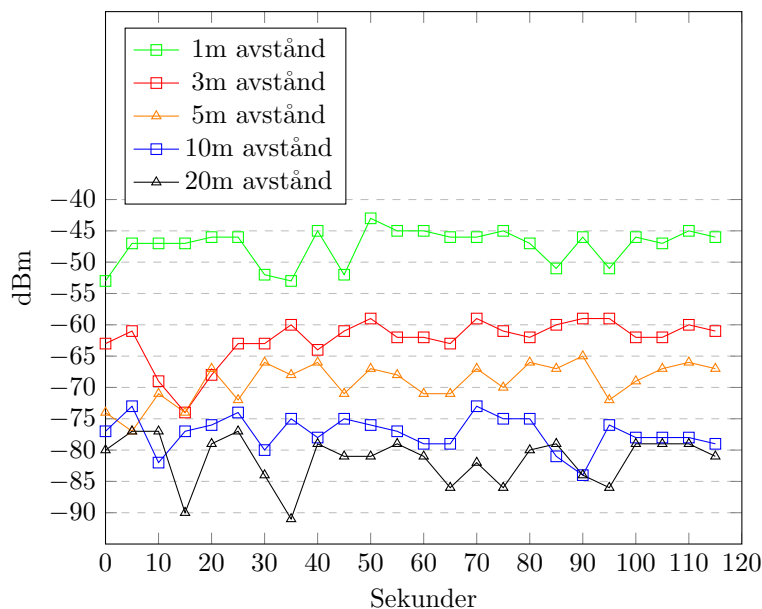
Under instämplingen kan användaren påbörja sin rast genom att använda rastknappen och ställa in en tidsgräns för rasten. Efter att tidsgränsen har utgått återupptar användaren sitt arbetspass.

5 Resultat

I denna del introduceras resultat på testerna och dess genomsnittliga värden på de olika protokollen som har tillämpats samt automationsnivåerna för varje handling i mobilapplikationen.

5.1 Uppmätningar på Router, BLE och GPS

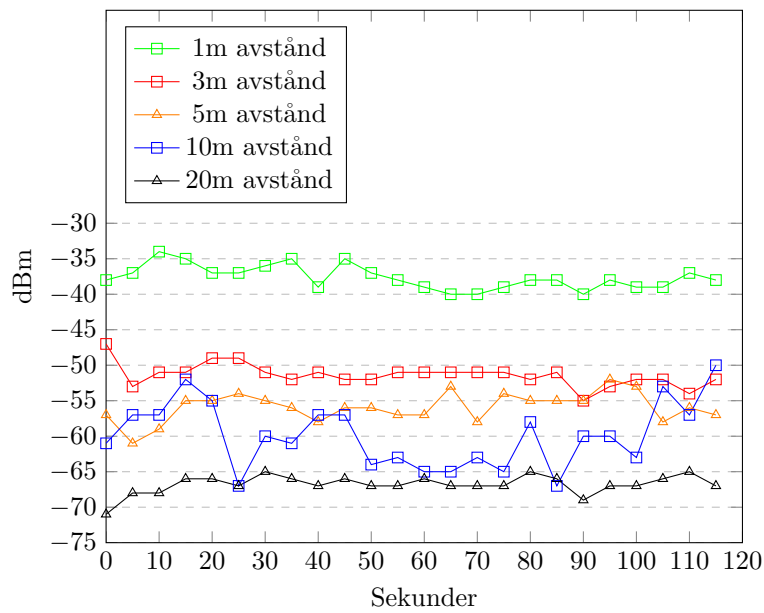
De uppmätta RSSI för BLE och Wi-Fi protokollen presenteras i figur 5.1, och 5.2. I Tabell 5.1, 5.2 och 5.3 presenteras den genomsnittliga RSSI eller GPS koordinater över 24 samplingar för respektive protokoll.



Figur 5.1: RSSI-uppmätningar för BLE-beacon.

1m från utgångspunkt	3m från utgångspunkt	5m från utgångspunkt	10m från utgångspunkt	20m från utgångspunkt
-53.625	-62.375	-69.125	-77.292	-81.542

Tabell 5.1: Genomsnittlig RSSI för BLE-beacon.



Figur 5.2: RSSI-uppmätningar för Wi-Fi router.

1m från utgångspunkt	3m från utgångspunkt	5m från utgångspunkt	10m från utgångspunkt	20m från utgångspunkt
-37.625	-49.417	-55.917	-59.875	-66.792

Tabell 5.2: Genomsnittlig RSSI för Wi-Fi router.

	Latitud	Longitud
0m från utgångspunkt	57.7061534751	11.9386697511
3m från utgångspunkt	57.7061553667	11.9386145833
5m från utgångspunkt	57.7061629167	11.9386056125
10m från utgångspunkt	57.7062042083	11.9385361962
20m från utgångspunkt	57.7063625917	11.9385756625

Tabell 5.3: Genomsnittliga GPS koordinater.

5.2 Jämförelse mellan BLE, GPS, Wi-Fi

Rangordningen för varje undersökning är baserad på färgkodningen där orange är första plats, röd är andra plats och grå på tredje plats. I Tabell 5.4 presenteras de maximala räckvidden med och utan hinder för Wi-Fi, BLE och GPS. I Tabell 5.5 presenteras resultaten efter *Haversine* och *RSSI Distance Model* tillämpningarna. Även fel marginalerna i procent för de olika avstånden och protokollen, se appendix B över felmarginal för individuella resultat.

	Wi-Fi router	BLE-beacon	GPS
Max-avstånd utan hinder (utomhus)	49.9m	42.3m	Ingen gräns ¹
Max-avstånd med gipshinder (6 dubbel gipsväggar)	30.5m	26m	Ingen gräns ¹

Tabell 5.4: Max-avstånd med och utan hinder.

	Wi-Fi router	BLE-beacon	GPS
Uppskattad avstånd (3m)	3.89m (+29.7%)	2.74m (-8.7%)	3.28m (+9.3%)
Uppskattad avstånd (5m)	8.22m (+64.4%)	5.96m (+19.2%)	3.95m (-21.0%)
Uppskattad avstånd (10m)	12.96m (+29.6%)	15.25m (+52.5%)	9.74m (-2.6%)
Uppskattad avstånd (20m)	28.72m (+43.6%)	24.87m (+24.4%)	23.92m (+19.6%)

Tabell 5.5: Genomsnittliga avståndsberäkningar

¹Ingen räckviddsgräns vid bra GPS-signaler från GPS-satelliter.

5.3 Automatiserade handlingar

I denna sektion presenteras det uppnådda resultatet av automationen i form av text och tabell.

Inloggning

Användaren har möjlighet till att hålla sig inloggad efter första interaktionen.

Instämpling av arbetsplats

Åtminstone ett av protokollen behöver vara tillgänglig samtidigt som användaren är inom området för automatisk instämpling. Annars krävs manuell inmatning från användaren.

Gå på rast

Användaren har möjlighet att gå på rast vid manuell inmatning.

Återuppta arbetspass

Användaren återupptar sitt arbetspass från automations-uppgiften "Gå på rast" efter en tidsgräns. Användaren har även möjlighet att återuppta arbetet med en manuell inmatning.

Hantering av bytet mellan BLE, Wi-Fi och GPS

Bytet av teknologierna sker automatiskt enligt prioriteringen då ett protokoll misslyckas.

Utstämpling

Användaren stämplas ut automatiskt ifall det sista prioriterade protokollet misslyckas med att fånga upp användarens position. Användaren har även möjlighet till att stämpla ut med en manuell inmatning.

Handling	Automationsnivå
Inloggning	2
Instämpling av arbetsplats	2
Gå på rast	0
Återuppta arbetspass	3
Hantering av bytet mellan BLE, Wi-Fi och GPS	3
Utstämpling av arbetsplats	3

Tabell 5.6: En samling av handlingar som testas utifrån automationsnivå.

6 Diskussion

I denna sektion diskuteras frågeställningarna med hjälp av resultaten och komplikationer som uppstod i detta projekt.

6.1 Metodkritik

Undersökningarna utreddes framförallt med fysiska Android-enheter och det vore lämpligt att även undersöka med iOS-enheter för att jämföra resultatet.

För en optimal undersökning på max-avståndet med och utan hinder vore att det lämpligt att utförandet av testerna sker i samma miljö. Anledning är att denna utredning utfördes inomhus för max-avstånd med hinder och max-avstånd utan hinder utomhus på grund av lokalbrist.

Överlag används React Native paketen från npm för att underlätta utveckling, för att inte behöva återskapa innehållet av paketen. Däremot blir mobilapplikationen beroende av paketen och det kan förekomma komplikationer om de inte underhålls för framtida mobil-enheter.

I tidigare forskning nämns det att RSSI är ett dåligt sätt att mäta avstånd då reflektion, spridning och fysiska egenskaper har en stor påverkan. Det innebär att pålitligheten av avståndsberäkningarna är bristande.

6.2 Pålitligheten av positionering med BLE, Wi-Fi, GPS

För positionering av GPS ansågs endast träffsäkerheten vid olika avstånd vara av intresse. Anledningen till det är att GPS positionering kan ske globalt och avståndet har inga restriktioner för våra undersökningar. Därmed fastställdes resultatet för GPS i tabell 5.4 till "Ingen gräns".

En stor faktor som påverkar utredningen på max-avståndet av BLE-protokollet är intervallet mellan sökningar. För att hitta en BLE-enhet har BLE-protokollet en tidsgräns innan det går över till ett annat protokoll. Detta kan ge komplikationer vid korta intervaller eftersom sökningen av en enhet kräver längre tid ju större avståndet är mellan BLE- och mobil-enheten. Därmed påverkas även avståndsmätningen i resultatet.

Observationerna från tabell 5.4 visar att signalerna blir lägre med hinder. Därav vid användning av *RSSI Distance Model* måste konstanten n från formlerna (3) och (2) justeras för att överensstämmas med det faktiska avståndet och RSSI-signalen.

Utifrån resultatet i tabell 5.5 fluktuerar felmarginalen drastiskt över de olika protokoll avstånden, framförallt Wi-Fi routern och BLE-beacon. Detta kan bero på att RSSI värdena som returneras är strikt heltal. Ifall beräkningen från resultatet i tabell 5.1 hade beräknats med tre decimaler där A är -53.625 och RSSI är -62.375. Så hade det genomsnittliga avståndet varit 2.738419634 istället för 4. Alltså hade ett RSSI värde i decimaltal resulterat i ett mer exakt värde. Utöver detta kan signalstyrkan även bero på hårdvarans kvalitet.

6.3 Automation för handlingarna

Mobilapplikationen är ett fleranvändarsystem och kräver identifiering av användaren för att särskilja sig ifrån andra användare, alltså krävs åtminstone en första interaktion för inloggningen. Följaktligen anses automationsnivån 2 vara lämplig.

För instämpling bestämdes och ansågs automationsnivån 2 vara lämplig. I scenariot då

mobilapplikationen är i bakgrunden måste användaren starta igång mobilapplikationen för att initiera instämplingsprocessen. Om handlingen hade varit helt automatiserat, skulle det ge komplikationer för scenariot om användaren är inom området och inte är schemalagd att vara instämplad under den tiden.

För uppgiften ”Gå på rast” uppnåddes automationsnivån 0, men det vore lämpligt att få den uppmot automationsnivå 3. Anledningen till att lämpligheten är uppmot 3 är för att reducera interaktionen med mobilapplikationen så fort användaren ska gå på rast. En lösning som diskuterades var att kunna hantera schemalagda tider för användaren och inkludera rasttiden så mobilapplikationen hamnar i rastläge.

För båda uppgifterna ”Återuppta arbetspass” och ”Hantering av bytet mellan BLE, Wi-Fi och GPS” uppnåddes automationsnivån 3, vilket anses vara lämpligt. Eftersom det minimerar interaktionen med mobilapplikationen och ytterligare hantering anses inte vara nödvändigt.

Automationsnivån för uppgiften ”Utstämpling av arbetsplats” uppnåddes till automationsnivå 3 vilket också ansågs vara lämpligast. Ett scenario som kan ge komplikationer med tidrapporteringen är ifall användaren fortfarande befinner sig inom området trots att arbetspasset är över. Även i detta fall diskuterades en lösning där mobilapplikationen hanterar schemalagda tider för användaren som inkluderar utstämpling efter en viss tid.

6.4 Etik

Enligt dataskyddsförordningen[23] måste hanteringen av persondata upphållas. Vid inloggning hämtar mobilapplikationen persondata såsom förnamn och efternamn hos användaren från QBIS webbtjänster. Persondatan används bara när användaren är inloggad i mobilapplikationen och raderas sedan vid utloggning.

En annan viktig aspekt att ha i åtanke är övervakningsaspekten av mobilapplikationen där användarens platsdata uppenbaras. Detta bör också ha dataskyddsförordningen[24] i åtanke hur platsdatan används. Platsdatan för applikationen används i kombination med protokollen och bestämmer ifall användaren är inom området för att vara instämplad. Själva platsdatan sparas inte utan bara ankomst- och avgångstid. För användning av mobilapplikationen uppmanas användaren att tillåta platstjänster.

7 Slutsats

Enligt resultatet är bluetooth protokollet bäst vid distanser upp till 5m för träffsäkerheten. Däremot vid distanser över 5m konstaterades GPS protokollet som mest lämplig. I jämförelse mellan Wi-Fi och bluetooth anses Wi-Fi som ett bättre alternativ när det gäller max-avstånd däremot försämras träffsäkerheten. Sammanfattningsvis bör protokollen följa prioriteringen GPS, bluetooth och sedan Wi-Fi. Däremot begränsas pålitligheten till resultatet på grund av att RSSI är ett dåligt sätt att uppskatta avstånd. Därmed bör framtida undersökningar med liknande mål ha uppsikt över denna metod.

Utav automationsuppgifterna som mättes ansågs fem av sex vara lämpliga enligt resultatet där uppgiften "Gå på rast" bör vara på automationsnivå 3 istället för 0. Handlingarna i mobilapplikationen anses därför vara översiktligt villkorlig automatisk (automationsnivå 2).

Källor

- [1] Bluetooth. “Bluetooth technology overview.” (2022), [Online]. Available: <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/> (visited on 02/14/2022).
- [2] R. Heydon, *Bluetooth Low Energy, The Developer’s Handbook*. New Jersey NJ, USA: Pearson Education, 2013.
- [3] J. Lindh, “Bluetooth low energy beacons,” Texas TX, USA, Tech. Rep., 2016.
- [4] Mapsted. “What you must know about bluetooth beacons before purchasing.” (2022), [Online]. Available: <https://mapsted.com/blog/what-you-must-know-about-bluetooth-beacons-before-purchasing> (visited on 02/05/2022).
- [5] Cisco. “What is wi-fi.” (2022), [Online]. Available: <https://www.cisco.com/c/en/us/products/wireless/what-is-wifi.html> (visited on 02/14/2022).
- [6] Verizon. “What is wi-fi.” (2022), [Online]. Available: <https://www.verizon.com/info/definitions/wifi/> (visited on 02/14/2022).
- [7] S. Dhawan, “Analogy of promising wireless technologies on different frequencies: Bluetooth, wifi, and wimax,” in *The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007)*, 2007, pp. 14–14.
- [8] Geotab. “What is gps?” (2020), [Online]. Available: <https://www.geotab.com/blog/what-is-gps/> (visited on 03/07/2022).
- [9] B. T. Fang, “Trilateration and extension to global positioning system navigation,” *J. Guidance*, vol. 9, no. 6, p. 715, 2012.
- [10] N. R. Chopde and M. K. Nichat, “Trilateration and extension to global positioning system navigation,” *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 2, pp. 298–302, 2013.
- [11] K. Eriksson and H. Gavel, “Grafer,” in *Diskret matematik och diskreta modeller*, 2:6, Lund, SE: Studentlitteratur AB, 2018.
- [12] G. Li, E. Geng, Z. Ye, Y. Xu, J. Lin, and Y. Pang, “Indoor positioning algorithm based on the improved rssi distance model,” *Sensors*, vol. 18, no. 2820, pp. 1–6, 2018.
- [13] K. Benkic, M. Malajner, P. Planinsic, and Z. Cucej, “Using rssi value for distance estimation in wireless sensor networks based on zigbee,” in *2008 15th International Conference on Systems, Signals and Image Processing*, 2008, pp. 303–306.
- [14] S. Yoo and S. Managi, “To fully automate or not? investigating demands and willingness to pay for autonomous vehicles based on automation levels,” *IATSS Research*, vol. 45, no. 4, pp. 459–468, 2021.
- [15] Geotab. “React native learn once write anywhere.” (2020), [Online]. Available: <https://reactnative.dev/> (visited on 03/08/2022).
- [16] OpenJS. “About node.js.” (2022), [Online]. Available: <https://nodejs.org/en/about/> (visited on 02/14/2022).
- [17] npm. “About npm.” (2022), [Online]. Available: <https://www.npmjs.com/about> (visited on 02/14/2022).
- [18] P. Lenart and et al. “React native ble plx.” (2022), [Online]. Available: <https://dotintent.github.io/react-native-ble-plx/> (visited on 02/14/2022).
- [19] J. S. D. Cubides and et al. “React-native-wifi-reborn.” (2022), [Online]. Available: <https://github.com/JuanSeBestia/react-native-wifi-reborn> (visited on 02/14/2022).
- [20] I. Rifat and et al. “React-native-geolocation-service.” (2022), [Online]. Available: <https://github.com/Agontuk/react-native-geolocation-service> (visited on 02/14/2022).

- [21] R. martin and et al. “React native background service.” (2022), [Online]. Available: <https://github.com/Rapsssito/react-native-background-actions> (visited on 02/14/2022).
- [22] K. Borowy and et al. “React native async storage.” (2022), [Online]. Available: <https://github.com/react-native-async-storage/async-storage> (visited on 02/14/2022).
- [23] Integritetsskyddsmyndigheten. “Personuppgifter.” (2021), [Online]. Available: <https://mapsted.com/blog/what-you-must-know-about-bluetooth-beacons-before-purchasing> (visited on 05/10/2022).
- [24] —, “Ställ in hur din mobil använder platsdata.” (2021), [Online]. Available: <https://www.imy.se/privatperson/dataskydd/informationssakerhet/stall-in-hur-din-mobil-anvander-platsdata/> (visited on 05/10/2022).

A Källkod - Protokoll logik

```
export const isConnectedToCompanyBLE = async (BLEList) => {
  return new Promise((resolve, reject) => {
    var start = moment().utc(true)
    protocolIntervalTimer = setTimeout(async () => {
      bleManager.stopDeviceScan();
      resolve({ isConnected: false })
    }, actionInterval)
    bleManager.startDeviceScan(null, null, (error, device) => {
      bleManager.state().then((res) => {
        if (res == 'PoweredOn') {
          if (error) {
            bleManager.stopDeviceScan();
            console.log(error)
            resolve({ isConnected: false })
          }
          var isFound = BLEList.find((res) => res.id == device.id)
          if (isFound) {
            var end = moment().utc(true)
            var timeLeft = actionInterval - end.diff(start)
            console.log(`Found company device:${device.id}`);
            isConnected = true;
            bleManager.stopDeviceScan();
            resolve({
              isConnected: true,
              location: isFound.location,
              timeLeft: timeLeft})
          }
        }
        else {
          resolve({ isConnected: false })
        }
      })
    })
  });
}

export const isConnectedToCompanyWifi = async (WifiList) => {
  return new Promise((resolve, reject) => {
    var start = moment().utc(true)
    protocolIntervalTimer = setTimeout(() => {
      resolve({ isConnected: false });
    }, actionInterval)
    WifiManager.getCurrentWifiSSID().then(async (ssid) => {
      var isFound = WifiList.find((res) => res.id === ssid);
      if (isFound) {
        var end = moment().utc(true)
        var timeLeft = actionInterval - end.diff(start)
        console.log(`Connected to company Wifi${ssid}`)
        resolve({
          isConnected: true,
```

```

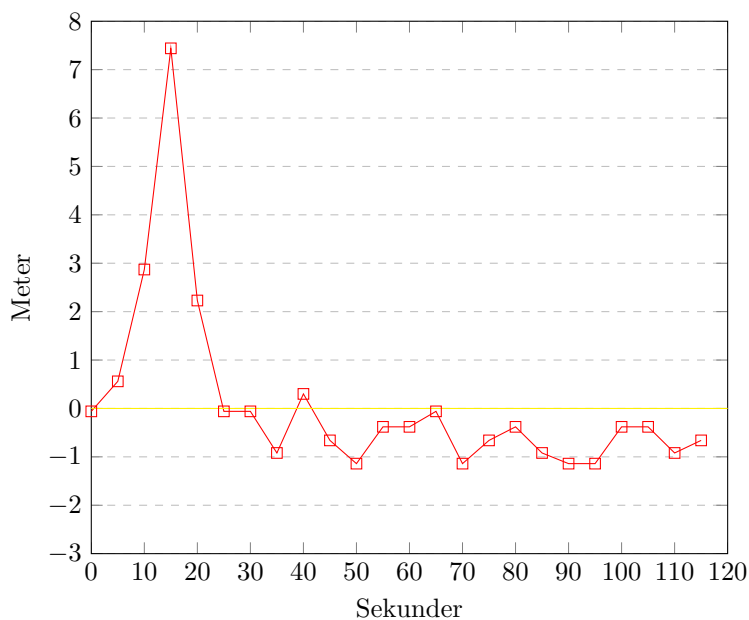
        location: isFound.location,
        timeLeft: timeLeft });
    }
    resolve({ isConnected: false });
  },
  (error) => {
    resolve({ isConnected: false });
  }
);
})
}

export const inProximityToCompanyGPS = async (GPSList) => {
  return new Promise((resolve, reject) => {
    var start = moment().utc(true)
    protocolIntervalTimer = setTimeout(() => {
      resolve({ isConnected: false });
    }, actionInterval)
    Geolocation.getCurrentPosition(
      (position) => {
        var currentLatPosition = position.coords.latitude;
        var currentLongPosition = position.coords.longitude;
        console.log(`Lat: ${currentLatPosition} Long: ${currentLongPosition}`)
        GPSList.forEach((companyLocation) => {
          var companyLatPosition = companyLocation.Lat;
          var compantLongPosition = companyLocation.Long;
          var maxDistance = companyLocation.distance;
          var distance = haversineToMeter(currentLatPosition,
                                           currentLongPosition,
                                           companyLatPosition,
                                           compantLongPosition);
          console.log(`distance:${distance} maxDistance:${maxDistance}`);
          if (distance < maxDistance) {
            var end = moment().utc(true)
            var timeLeft = actionInterval - end.diff(start)
            resolve({
              isConnected: true,
              location: companyLocation.location,
              timeLeft: timeLeft});
          }
        })
      })
    resolve({ isConnected: false });
  }, (error) => {
    reject(false);
  }, { enableHighAccuracy: true, timeout: 15000, maximumAge: 10000 }
);
})
}

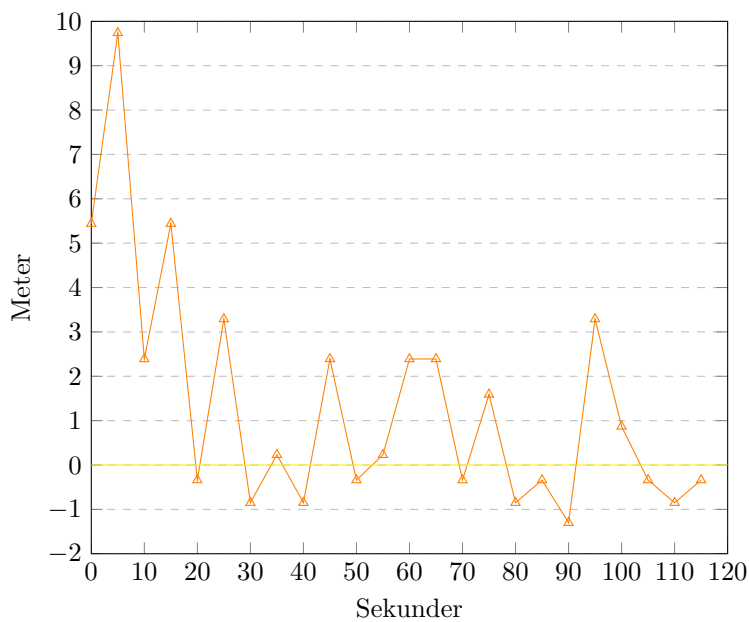
```

B Felmarginal för individuella resultat

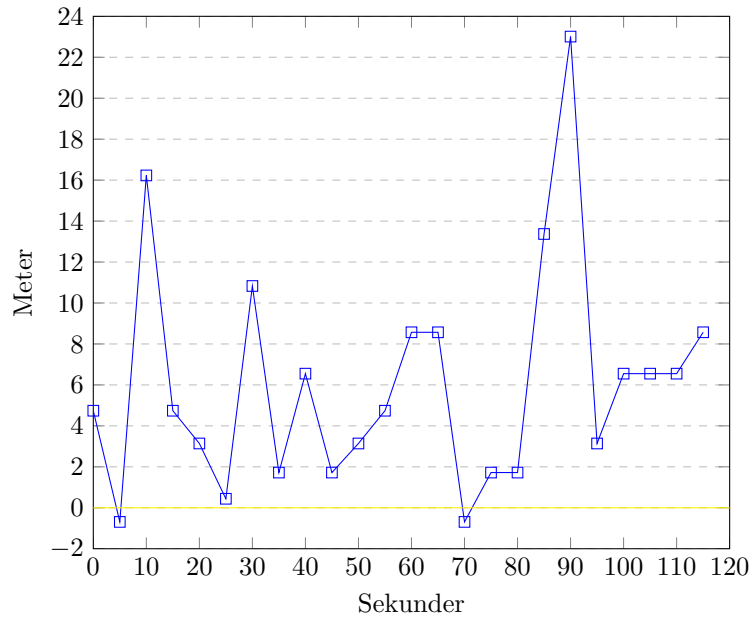
Figurerna visar felmarginaler där noll är utgångspunkten från de olika avstånden. En punkt ovanför utgångspunkten innebär en överskattning av avståndet medan en punkt under utgångspunkten innebär en underskattning av avståndet.



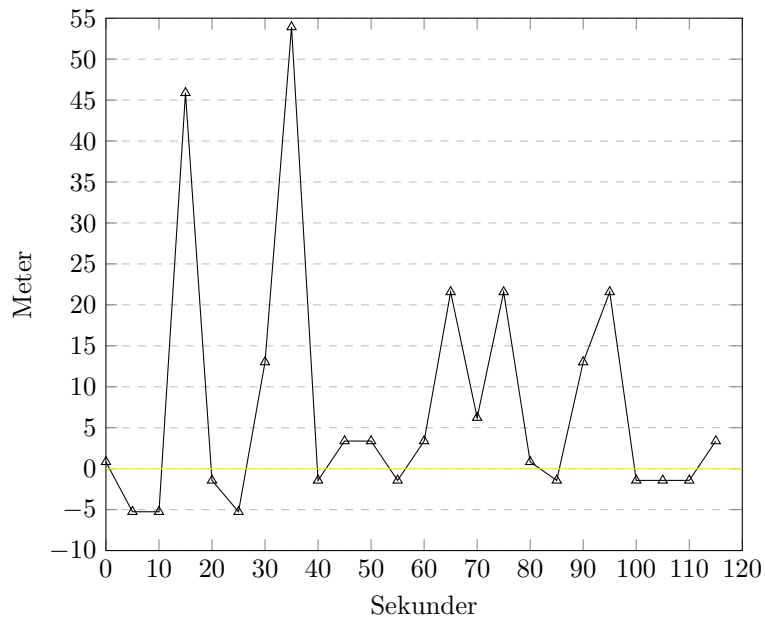
Figur B.1: Felmarginal efter avståndsberäkning för BLE-beacon från 3 meters avstånd.



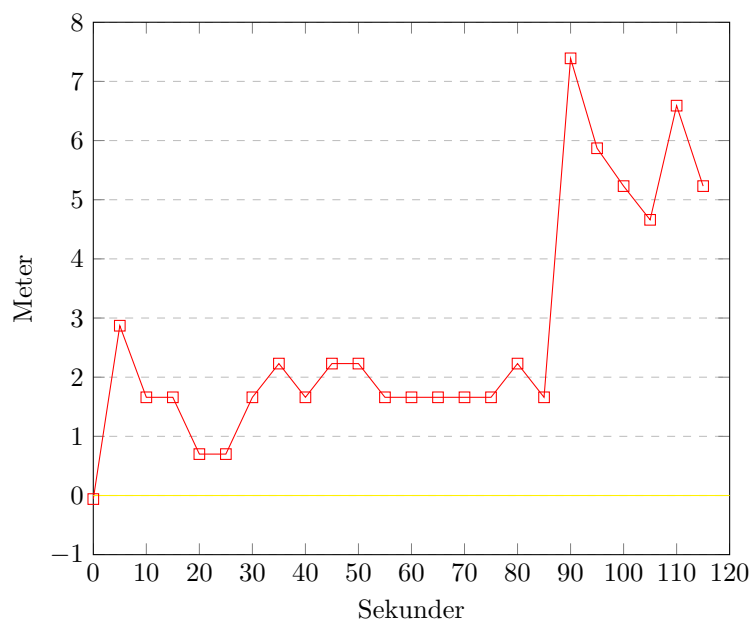
Figur B.2: Felmarginal efter avståndsberäkning för BLE-beacon från 5 meters avstånd.



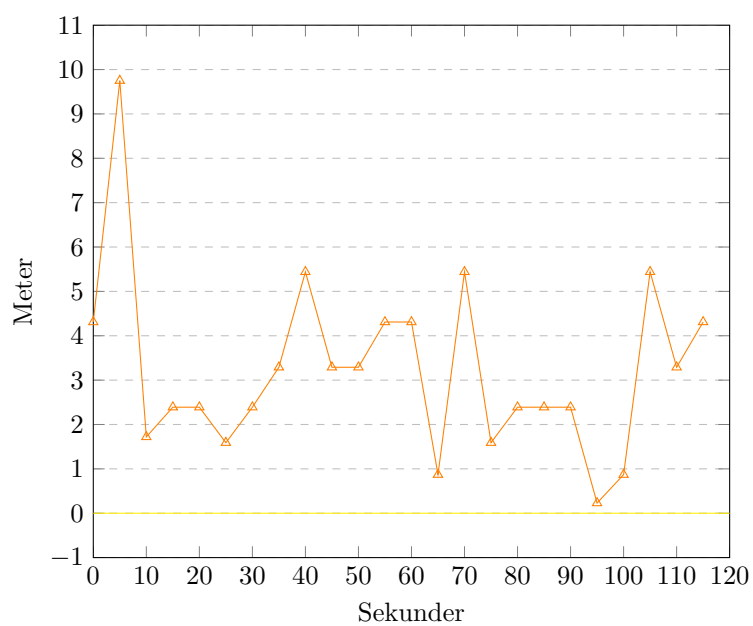
Figur B.3: Felmarginal efter avståndsberäkning för BLE-beacon från 10 meters avstånd.



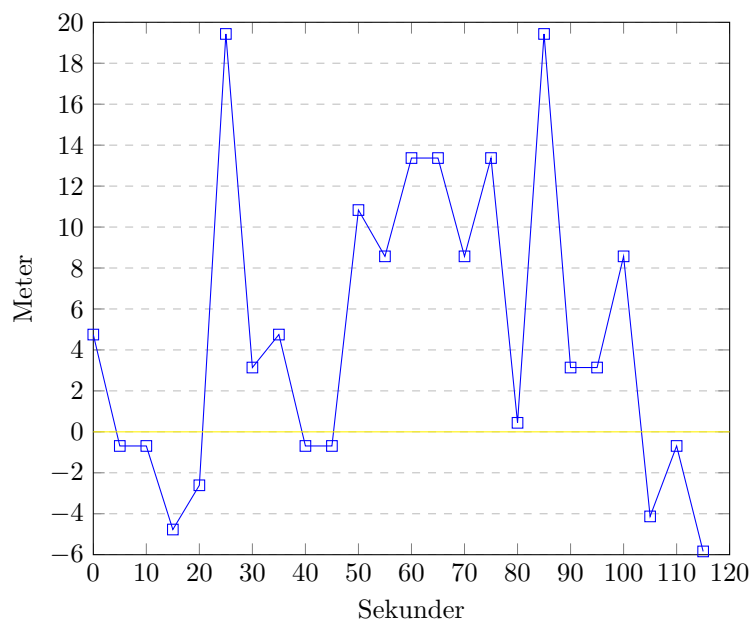
Figur B.4: Felmarginal efter avståndsberäkning för BLE-beacon från 20 meters avstånd.



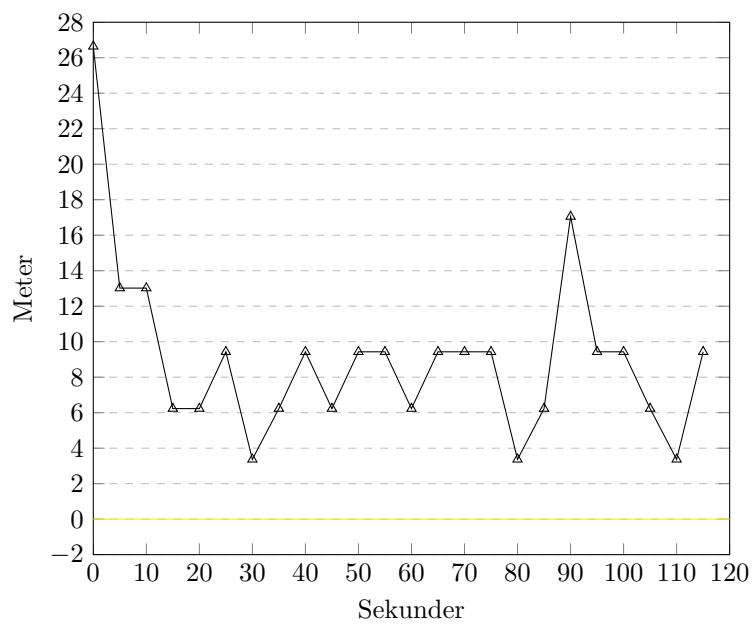
Figur B.5: Felmarginal efter avståndsberäkning för Wi-Fi router från 3 meters avstånd.



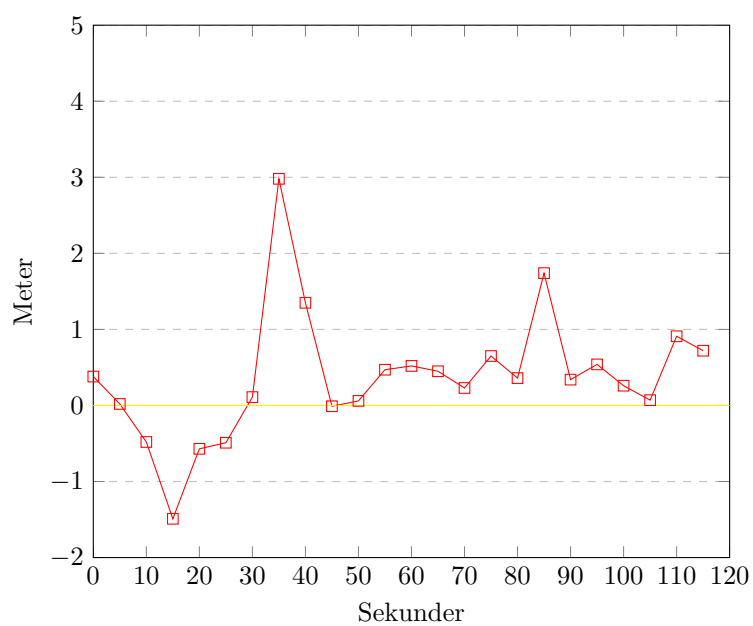
Figur B.6: Felmarginal efter avståndsberäkning för Wi-Fi router från 5 meters avstånd.



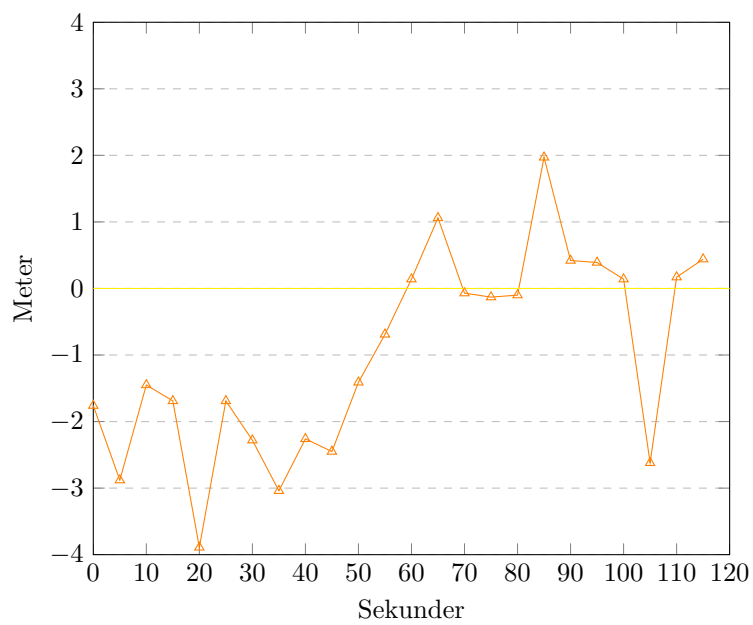
Figur B.7: Felmarginal efter avståndsberäkning för Wi-Fi router från 10 meters avstånd.



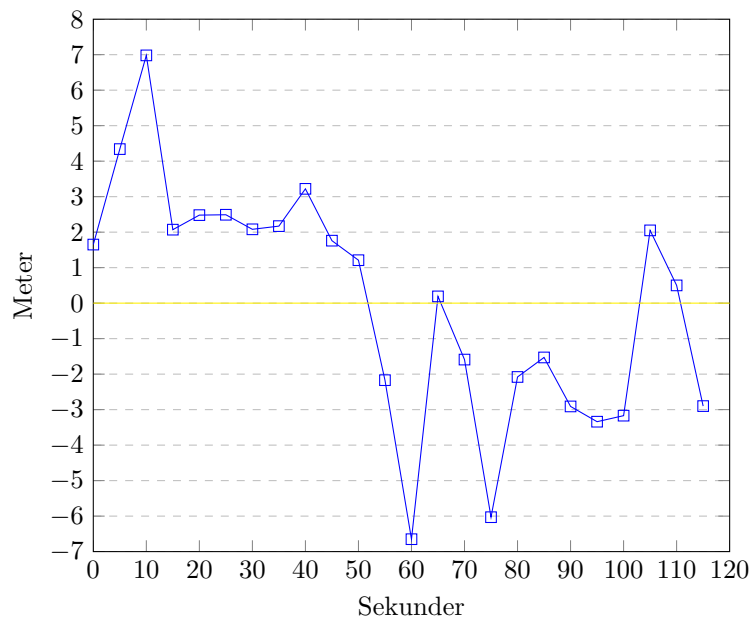
Figur B.8: Felmarginal efter avståndsberäkning för Wi-Fi router från 20 meters avstånd.



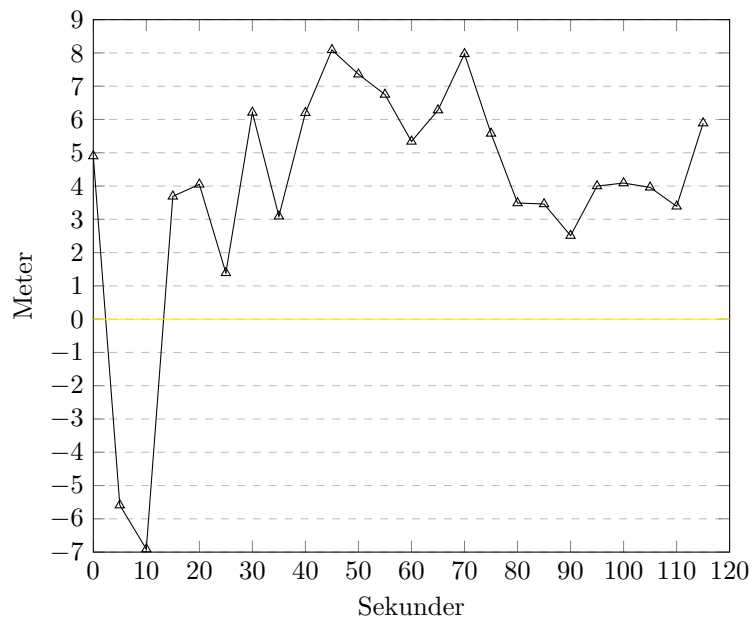
Figur B.9: Felmarginal efter avståndsberäkning för GPS från 3 meters avstånd.



Figur B.10: Felmarginal efter avståndsberäkning för GPS från 5 meters avstånd.



Figur B.11: Felmarginal efter avståndsberäkning för GPS från 10 meters avstånd.



Figur B.12: Felmarginal efter avståndsberäkning för GPS från 20 meters avstånd.



CHALMERS