

Användning av BLE-beacons för inomhuspositionering

MATTIAS BERLIN

ALBIN CEDERBLAD

BACHELOR'S THESIS

DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY

FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY



Användning av BLE-beacons för inomhuspositionering

- En undersökning av Bluetooth Low Energy beacons för positionering i en byggnad



LUNDS UNIVERSITET
Campus Helsingborg

LTH Ingenjörshögskolan vid Campus Helsingborg
Institutionen för datavetenskap

Examensarbete:

Mattias Berlin
Albin Cederblad

© Copyright Mattias Berlin och Albin Cederblad

LTH Ingenjörshögskolan vid Campus Helsingborg
Lunds universitet
Box 882
251 08 Helsingborg

LTH School of Engineering
Lund University
Box 882
SE-251 08 Helsingborg
Sweden

Tryckt i Sverige
Lunds universitet
Lund 2017

Sammanfattning

I detta examensarbete undersöks hur Bluetooth Low Energy (BLE) beacons kan användas för att implementera ett inomhuspositioneringssystem. För att göra detta utvecklades en prototyp som består av en Android-applikation och en DynamoDB-databas. Att använda sig av trilateration för positionering har redan testats och utvärderats ett flertal gånger, därför fokuserar detta arbete istället på en alternativ metod. Genom att placera ut ett beacon i var rum eller plats som man vill ha med i systemet (som man vill ska visas i applikationen). Det beacon vars signaler har störst signalstyrka antas vara det som är närmast. Denna plats markeras på en karta i applikationen. I prototypen finns en karta över ett kontor där prototypen testades. En extern webbaserad statistiksida implementerades även som visar hur många besökare det har varit i var tillagt rum under en given dag.

Så länge varje beacon ställs in på en lämplig signalstyrka med tanke på rummets storlek samt väggarnas tjocklek och material fungerade denna metod bra vid utvärderingar. Denna slags system tros ha potential, delvis med tanke på ett ökande antal företag som utvecklar inomhuspositioneringssystem. Även andra användningar av BLE-beacons kan växa, särskilt om beaconen blir ännu mindre och energisnålare skulle det vara sannolikt att dessa används ännu mer i bland annat butiker och liknande.

Nyckelord: Bluetooth Low Energy, BLE, Eddystone, Smartphone, Indoor Location, Indoor Positioning System, Android, Java

Abstract

This thesis aims to examine how Bluetooth Low Energy (BLE) beacons can be used to implement an indoor positioning system. To do this, a prototype was developed consisting of an Android application and a DynamoDB database. Using trilateration for positioning has already been tested and evaluated several times. This work instead focuses on an alternative method. This method works by placing a beacon in any room or place you want to be included in the system. The beacon whose signals are perceived as the strongest are assumed to be the closest. This location is marked on a map in the application. The map used in the prototype is a floor plan of the office where the system was tested. An external web application was also implemented to show statistics for the number of visitors of each room on a given day.

If the beacons are configured with an appropriate signal strength relative to the size of the room and the thickness of the walls, this method is evaluated to work well. This type of system is believed to have potential, partly in view of an increasing number of companies that develop indoor positioning systems. Other uses of BLE beacons can also grow, especially if the development leads to even smaller and more energy efficient beacons, it is likely that these would see more use cases in stores and other commercial environments.

Keywords: Bluetooth Low Energy, BLE, Eddystone, Smartphone, Indoor Location, Indoor Positioning System, Android, Java

Förord

Vi skulle vilja tacka Per Bränner på Knowit som varit vår handledare och gett oss möjligheten att genomföra detta examensarbete. Vi vill även tacka Johan Lohmander på Knowit för hans hjälp och kunskap inom detta område som hjälpt oss genomföra examensarbetet. Vi har fått mycket kunskap om bland annat olika ramverk och bibliotek under arbetets gång. Även erfarenhet om hur det är att vara i en professionell miljö vilket vi ser som väldigt värdefull.

Tack till Christin Lindholm och Christian Nyberg som varit vår handledare respektive examinator i detta examensarbete.

Innehållsförteckning

1	Inledning	1
1.1	Bakgrund	1
1.2	Syfte och målsättning	1
1.3	Problemformulering	2
1.4	Avgränsningar	2
2	Teknisk bakgrund	3
2.1	Olika tekniker för inomhuspositionering	3
2.1.1	Global navigation satellite system (GNSS)	3
2.1.2	Wi-Fi	3
2.1.3	Bluetooth Low Energy (BLE)	4
2.2	BLE-beacons	4
2.2.1	BLE-protokoll	5
2.2.1.1	Eddystone	5
2.2.1.2	iBeacon	9
2.2.1.3	AltBeacon	9
2.3	Android Studio	9
2.4	Amazon Web Services	9
2.4.1	Amazon DynamoDB	10
2.4.2	Amazon Cognito	10
2.4.3	AWS Lambda	10
2.5	iBKS Config Tool	10
2.6	Leaflet	10
2.7	WebView	10
2.8	Canvas	11
2.9	OpenGL ES	11
3	Arbetsmetod	13
3.1	Tidsplan	13
3.2	Arbetsprocess	13
4	Analys	15
4.1	Implementeringssätt för inomhuspositionering	15
4.2	Kommunikation mellan front- och backend	16
4.3	Navigationshjälp	17

4.4	Källkritik	17
4.4.1	Primära källor	17
4.4.2	Sekundära källor	18
5	Resultat	19
5.1	Hårdvara	19
5.2	Mobilapplikationen	19
5.2.1	Skanning efter beacons	20
5.2.2	Leaflet	20
5.2.3	Backend	21
5.2.4	Gränssnitt	22
5.2.5	Statistik	23
5.3	Statistiksida	23
6	Slutsats och diskussion	25
6.1	Problemformulering	25
6.2	Framtida utvecklingsmöjligheter och förbättringar	26
6.3	Etiska aspekter	26
7	Referenser	27
8	Terminologi	31

1 Inledning

1.1 Bakgrund

Uppkoppling av föremål mot Internet blir allt vanligare. Detta har öppnat upp en ny marknad med nya möjligheter. Användning av små sändare — så kallade beacons — som med jämna intervall skickar ut information om sig själva har börjat öka tillsammans med mobilapplikationer inom en rad olika områden för platsbaserade tjänster. Beacons kan till exempel användas för platsbaserade notifikationer (till exempel om man går till en viss hylla i en butik så får man erbjudanden relaterade till produkterna på den), platsbaserade händelser (en viss ljudfil spelar upp när man går nära en utställning) i museer eller för automatisk tidsrapportering på arbetsplatser.

Knowit AB är ett konsultbolag som arbetar inom design, kommunikation, managementkonsulting samt IT. Företaget startades i Sverige men finns idag även i Norge, Finland, Danmark, Tyskland och Estland. Knowit är uppdelat i flera olika affärsområden, exempelvis tillhandahåller Knowit en digital bank åt en av världens största bankkoncerner: Banco Santander.

Knowit tror att användningen av beacons kommer att växa. Företag inom bland annat handeln och nöjesbranchen kan ha nytta av lösningar som omfattar beacons. Knowit vill därför undersöka hur beacons fungerar genom att i detta examensarbete ta fram en prototyp till ett positioneringssystem som använder sig av just sådana sändare. Denna prototyp gör det möjligt att se i vilket rum (de rum där beacon är placerade) man befinner sig i på en karta i sin smartphone. På så sätt blir det lättare att navigera till andra rum, speciellt i de fall då man inte känner till byggnaden. Prototypen kan vidareutvecklas till att exempelvis även fungera som ett bokningssystem för att boka mötesrum, och visa vilka rum som är lediga samt bokade i ett kontor.

Det finns flera företag som har utvecklat liknande system som använder sig av beacons såsom Estimote, Locatify och Aruba vilket visar att det finns intresse för det.

1.2 Syfte och målsättning

Syftet med examensarbetet är att utveckla en prototyp till detta positioneringssystem samt utforska potentialen för Bluetooth Low Energy beacons i den här typen av system. Prototypen omfattar en mobilapplikation och en lösning för backend för att bestämma mobilens position i

den fördefinierade byggnaden. Olika tjänster för backend-lösningen kommer därför att utvärderas.

För att utveckla denna prototyp utforskas olika metoder för att hitta sin position med BLE-beacons.

Målet med examensarbetet är att utveckla en mobilapplikation där användaren kan se i vilket rum man befinner sig i. Denna applikation ska använda sig av en databas för att lagra och hämta information om beacons. Någon form av navigeringshjälp (från rum till rum) ska vara implementerad, exempelvis med en pil som visar riktningen till rummet, eller pathfinding.

Mobilapplikationen ska kunna ta emot paketen som BLE-beacons skickar ut och skicka vidare den data som krävs för positionering till backend. Backend ska göra det möjligt för applikationen att få information relaterad till beaconen, såsom namnet på rummet och positionen på kartan.

Statistik om hur många gånger var rum besöks per dag ska sparas i backend och kunna nås via ett gränssnitt.

1.3 Problemformulering

I examensarbetet kommer följande frågor att besvaras:

- Hur många Bluetooth Low Energy beacons krävs för att med hög precision kunna hitta sin position och möjliggöra navigering mellan olika rum på Knowits kontor?
- På vilka sätt kan man med hjälp av BLE-beacons få reda på var i en byggnad mobilen befinner sig? Hur precisa är de olika metoderna?
- Vad finns det för möjligheter med anseende på implementering av navigation till mål inom byggnaden?
- Vilken inställning på signalstyrkan är lämplig för de olika positioneringsmetoderna?

1.4 Avgränsningar

Följande är avgränsningar för examensarbetet:

- Mobilapplikationen blev endast utvecklat för operativsystemet Android.
- Mobilapplikationen kan endast ta emot och detektera signaler från beacons som följer Eddystone-protokollet (Googles Bluetooth Low Energy beacon-plattform).

2 Teknisk bakgrund

Detta kapitel beskriver olika tekniker som kan användas för inomhuspositionering, hur BLE fungerar samt hur de mest använda BLE-protokollen ser ut. Kapitlet ger även en teknisk bakgrund av hårdvaran och de verktyg som använts i examensarbetet.

2.1 Olika tekniker för inomhuspositionering

Det finns ett flertal olika metoder för att kunna implementera funktionaliteten som möjliggör positionering inomhus. Några av dessa är GPS, Wi-Fi och Bluetooth Low Energy (BLE) beacons. I de följande avsnitten beskrivs det hur man kan använda dessa och hur lämpliga de skulle vara.

2.1.1 Global navigation satellite system (GNSS)

Global navigation satellite system (GNSS) såsom GPS, GLONASS och Galileo är system som gör det möjligt för elektronisk utrustning att ta reda på sin position. [1] En stor fördel med dessa system är att de har global täckning med hjälp av satelliter i omlopp runt jorden. Dessa skickar ständigt ut tidssignaler som mottagaren jämför med tiden som dem togs emot. Med hjälp av denna tidsskillnad kan man beräkna avståndet mellan satelliten och mottagaren då man vet signalens hastighet; vilket är ljusets hastighet.

Om man endast hade använt sig av en sådan satellit hade man fått en sfär av möjliga positioner för mottagaren då alla punkter på den sfären har samma avstånd mellan sändare och mottagare. Om man istället använder två bildas det en cirkel med möjliga positioner där de två sfärerna skär varandra. Det krävs alltså tre satelliter för att bestämma vilken latitud och longitud som mottagaren befinner sig i. Dock blir det två möjliga altituder i detta fall så man måste ha signaler från fyra eller fler satelliter för att bestämma även altituden.

Denna teknologi utvecklades ursprungligen för militära ändamål, exempelvis för guidade vapen och för att göra det möjligt för trupper att lokalisera sig själva. Nu har den dock mycket användning även inom det civila, framförallt i transport (både flyg- och motortrafik).

Med denna teknik kan man få reda på sin position med en precision på några meter utomhus, men inomhus har mikrovågor en tendens att försvagas och studsas på tak, väggar och andra objekt vilket försämrar precisionen. Med tanke på detta kan inte GNSS ses som särskilt lämpligt för denna slags tillämpningar.

2.1.2 Wi-Fi

Wi-Fi [2] kan användas genom att man utnyttjar Wi-Fi accesspunkter (AP) och använder trilateration, vilket betyder att man beräknar ens position med hjälp av tre olika signaler, i detta fall genom att utnyttja deras signalstyrkor på liknande sätt som GNSS fungerar. En signifikant fördel är att de flesta utrymmen som man skulle vilja ha ett positioneringssystem inomhus för redan har accesspunkter installerade. Med andra ord skulle man inte behöva köpa och installera extra hårdvara för denna funktionalitet. De är dock inte utformade för att användas på detta sätt och det kan även finnas problem med att det inte finns tillräckligt med

AP:er eller att de är inställda på att inte vilja användas på det sättet.

2.1.3 Bluetooth Low Energy (BLE)

Bluetooth är en standard för trådlös kommunikation som började utvecklas under 90-talet. [3] Den har sedan dess gått genom flera generationer av förbättringar och tillägg. Bluetooth Special Interest Group (SIG) släppte 2010 specifikationen Bluetooth 4.0 där Bluetooth Low Energy-protokoll (BLE) är inkluderade. BLE togs huvudsakligen fram för att sända och ta emot data utan att behöva skapa en anslutning. Man ville även reducera energiförbrukningen, chipstorleken och kostnaden men behålla likheten med Bluetooth-standarderna. Anledningen till detta var för att nå marknader inom bland annat smarta hem, hälsa, fitness och försäljning.

Både denna teknologi och dess användning är i skrivande stund i ett tidigt stadium och håller fortfarande på att utvecklas. Bluetooth SIG har gjort förbättringar för BLE i specifikationerna 4.2 och 5.0. [4] Dock tar det ett tag innan dessa kommer i allmänt bruk.

I nuläget används BLE i huvudsak av beacons, till exempel för att ge användare i närområdet information om erbjudanden i butiker på sin smartphone (se kapitel 2.2).

2.2 BLE-beacons

BLE-beacons är små sändare som med ett inställbart intervall skickar ut små datapaket och generellt kräver väldigt lite energi. [5] I den vanligaste typen av paket skickar man bara ett unikt ID som appen eller servern sedan får tolka och eventuellt utföra något som till exempel att visa ett meddelande för användaren.

Med tanke på de låga energikraven kan man göra enheterna som skickar ut signalerna väldigt små. En vanlig form är små och cirkulära som iBSK 105 [6] som används i detta examensarbete (se figur 1). Det finns även modeller som är små USB-sändare (ofta kallade nanosändare) som man kan koppla in i exempelvis en dator.

Dessa beacons batteritid beror på hur de är konfigurerade. Om man ökar signalstyrkan och signalfrekvensen minskar den förväntade livslängden på batteriet. Även längden på paketen som skickas är en faktor. Detta bör man ta hänsyn till om enheten använder batterier vilket de flesta beacons gör. I tillverkarens produktspecifikation brukar det finnas en tabell som visar hur batteriets livslängd förhåller sig till signalstyrkan vid en viss signalfrekvens.



Figur 1. iBKS 105 som användes i examensarbetet

2.2.1 BLE-protokoll

Det finns ett flertal protokoll som är implementerade enligt BLE:s specifikation, till exempel Eddystone [7] (Google), iBeacon [8] (Apple) och AltBeacon [9] (Radius Networks). I detta projekt användes Eddystone så det är detta protokoll som kommer att stå i fokus i rapporten och därmed beskrivs i större detalj.

2.2.1.1 Eddystone

Eddystone släpptes i juli 2015 av Google och fungerar både till Android och iOS. Eddystone-protokollet är öppet för alla tillverkare att implementera för alla ändamål och specifikationen finns tillgänglig på GitHub. [10] Både signalfrekvensen och signalstyrkan är inställbara.

Protokollet består av fyra typer av ramar: UID (Unique Identifier), URL (Uniform Resource Locator), TLM (Telemetry) och EID (Encrypted Ephemeral Identifier). [10] Dessa ramar har samma början, förutom längden på tjänstdatan, vilket är ett UUID (Universally Unique Identifier) samt ramen, som är olika för var ram. Detta format kan ses i tabell 1.

Tabell 1. Format för Eddystone

Fält	Antal bytes	Beskrivning
Längd	1	Längd på flaggsektionen (0x02)
Flaggor datatyp	1	
Flaggor	1	
Längd	1	Inkluderad för att möjliggöra bakgrundsskanning i iOS
Tjänst datatyp	1	Inkluderad för att möjliggöra bakgrundsskanning i iOS
Eddystone UUID	2	Inkluderad för att möjliggöra bakgrundsskanning i iOS
Längd	1	Längd på tjänstdatan (inklusive ramen)
Tjänstdata datatyp	1	
Eddystone UUID	2	
Eddystone-ram	Max 20	En av Eddystones ramar (exempelvis Eddystone-UID)

Eddystone-UID (Unique Identifier) sänder en unik ID för var beacon. Detta ID består av 16 bytes, varav 10 bytes används för namespace (sv: *namnrymd*) ID och 6 bytes för instance (sv: *instans*) ID. [11] Namespace ID kan användas för att gruppera beacons och instance ID för att identifiera individuella beacons i gruppen. Tabell 2 visar formatet för denna ram.

Tabell 2. Ramformat för Eddystone-UID

Fält	Ramtyp	Avståndsdata	Namespace ID	Instance ID	Reserverad
Antal bytes	1	1	10	6	2
Beskrivning	0x00	Kalibrerad sändningsstyrka vid 0 meter	Kan användas för att gruppera beacons	Unikt id för att identifiera individuella beacons i gruppen	Reserverad för framtida användning, måste vara 0x00

Eddystone-URL sänder en komprimerad URL som kan öppnas av klienter med Internetanslutning. [12] Den är baserad på ett projekt kallat UriBeacon som hade som mål att undersöka hur man kan använda BLE-teknologi för att dela URL:er. [13] URL:en

komprimeras för att göra det möjligt att få plats med längre länkar i det begränsade annonseringsutrymmet. Detta görs genom att matcha URL:ens prefix i ett kodat prefixschema enligt tabell 3 nedan. Det finns även kodning för vanliga toppdomän som kan användas i sista fältet (kodad URL) i ramen (se tabell 4).

Tabell 3. URL prefixschema för Eddystone-URL

Decimal	Hex	Expanderat
0	0x00	http://www.
1	0x01	https://www.
2	0x02	http://
3	0x03	https://

Tabell 4. Ramformat för Eddystone-URL

Fält	Ramtyp	Avståndsdata	URL-schema	Kodad URL
Antal bytes	1	1	1	Max 17
Beskrivning	0x10	Kalibrerad sändningsstyrka vid 0 meter	Kodat prefixschema (se tabell 2)	URL med möjlighet till kodning för de vanligaste toppdomänen

Eddystone-TLM står för telemetry (sv: *telemetri*) och används för att skicka data om ett beacons tillstånd. [14][15] Man kan få reda på batterispänningen, temperaturen, antal skickade paket sedan omstart samt tiden sedan den senaste omstarten. Denna ram finns i både krypterad samt okrypterad variant. De har både samma ramtyp (0x20) men versionsfältet identifierar om den är krypterad eller inte.

Då TLM-paket inte innehåller beacon ID måste dessa skickas tillsammans med någon annan ram (till exempel UID), så att man kan identifiera vilken beacon som telemetrin tillhör. Se tabell 5 och 6 för formatet för okrypterade respektive krypterade TLM-ramar.

Tabell 5. Ramformat för okrypterad Eddystone-TLM

Fält	Ramtyp	Versio n	Batterispänning	Temperatur	Annonserings- räknare	Tidräknare
Antal bytes	1	1	2	2	4	4
Beskrivning	0x20	0x00	1 millivolt per bit	Celsius	Antalet skickade ramar sedan omstart	Tid sedan omstart, 0.1 sekund upplösning

Tabell 6. Ramformat för krypterad Eddystone-TLM

Fält	Ramtyp	Versio n	Krypterad TLM- data	Salt	Integritetskoll
Antal bytes	1	1	12	2	2
Beskrivning	0x20	0x01		Används för att försvåra dekryptering	Används för att kolla så att ramens innehåll inte har ändrats

Eddystone-EID ramen används för att skicka en kortlivad identifierare som periodiskt ändras. [16] Dess information kan beräknas av användaren som registrerade den eftersom de delar en krypteringsnyckel, men uppfattas som slumpvald för alla andra användare. Denna ram är avsedd för att ge utvecklaren kontroll över vilka klienter som kan använda deras beaconsignaler. Ramformatet visas i tabell 7.

Tabell 7. Ramformat för Eddystone-EID

Fält	Ramtyp	Avståndsdata	EID
Antal bytes	1	1	8
Beskrivning	0x30	Kalibrerad sändningsstyrka vid 0 meter	Identifierare som ändras periodiskt.

Google har släppt två API:er som är relevanta för beacons: Nearby API och Proximity Beacon API. I den förstnämnda finns det en funktion kallad Nearby Notifications som nyligen är tillagd och är till för att med hjälp av beacons kunna skicka meddelanden och länkar utan att den måste ha en specifik app installerad för detta. [17] Den andra API:n, Proximity Beacon API, är en molntjänst som tillåter en att se all relevant data om sina registrerade beacons via ett gränssnitt. [18] Med hjälp av TLM-ramarna beskrivna ovan kan man se all tillståndsdata

som skickas i dessa, såsom batterinivån. Man kan även associera data med sina beacons och skicka dessa som en bilaga via Nearby Messages API.

2.2.1.2 iBeacon

iBeacon släpptes 2013 av Apple. [19] Några nämnvärda skillnader mellan iBeacon och det ovan beskriva Eddystone är:

- iBeacon endast kan skicka en typ av signal som består av ett UUID samt två inställbara värden (kallade major och minor), som kan användas för att dela in och segmentera ett område. Till exempel kan alla beacons i en butik ha samma major men olika minor-värden för att kunna identifiera individuella beacons.
- Apple tillhandahåller inga API:er relaterade till beacons.
- iBeacon är proprietär programvara, till skillnad från Eddystone som har öppen källkod och är licensierad under Apache v2.0.
- iBeacon har endast officiell support av iOS medan Eddystone har support av både iOS och Android. Det går även att skanna efter dessa på Android men filtreringen och tolkningen av BLE-paket måste i så fall göras med egen kod eller med ett externt bibliotek som till exempel Radius Networks “Android Beacon Library”.

2.2.1.3 AltBeacon

AltBeacon släpptes 2014 av Radius Networks och är en öppen beaconspecifikation. [20]

AltBeacon skapades för att erbjuda kompatibilitet med både Android och iOS.

Några nämnvärda skillnader från Eddystone och iBeacon är:

- Deras protokoll är öppet för alla tillverkare att implementera och kräver inga ytterligare avgifter, till skillnad från iBeacon där tillverkaren måste godkänna ett avtal och få en licens. [21]
- AltBeacon-paketen kan innehålla ett ID som är specifikt för tillverkaren samt information om tillverkaren i slutet av paketet. [20]

2.3 Android Studio

Android Studio [22] är en integrerad utvecklingsmiljö (*en*: Integrated Development Environment, IDE) som är specifikt designad för att utveckla Android-applikationer. Android Studio är baserad på IntelliJ IDEA (en IDE som används för utveckling av Java-applikationer). Programmeringsspråket som används för att utveckla applikationer till Android är Java.

2.4 Amazon Web Services

Amazon Web Services (AWS) [23] är en samling tjänster som tillhandahålls av ett dotterföretag till Amazon. Det tillåter prenumaranter att använda deras serverfarmar för diverse molntjänster såsom databaser. Det finns en gratis version kallas “AWS Free Tier”. [24] I denna finns det många tjänster som är tillgängliga i 12 månader men även en del som varar oändligt (dock har dem andra begränsningar såsom max lagringsutrymme).

2.4.1 Amazon DynamoDB

Amazon DynamoDB [25] är en NoSQL-databas och är en del av Amazon Web Services (AWS). Ursprungligen refererade namnet NoSQL till att databasen inte är en relationsdatabas men det betyder inte nödvändigtvis att den inte stödjer sökningar på SQL-format. I DynamoDB har man inte förbestämda attribut (förutom en primärnyckel, och eventuellt en så kallad sorteringsnyckel) i var tabell såsom relationsdatabaser har. Exempelvis kan en rad (kallad post eller tupel) i en tabell med primärnyckeln "Namn" ha värdet 37 i som "Ålder"-attribut medan en annan post inte har attributet "Ålder" men har istället ett attribut "Födelseort" satt som "Lund".

I AWS Free Tier ingår det 25 GB data och tillräckligt med läs- och skrivkapacitet för upp till 200 miljoner anrop per månad. [24] Denna tjänst används i examensarbetet för att lagra information om beaconen och statistik relaterade till dessa (antal besök i var mötesrum per dag).

2.4.2 Amazon Cognito

Amazon Cognito [26] används för att möjliggöra användarauthentiserad åtkomst till databasen via bland annat mobilapplikationer. Med detta kan man skapa användarpooler (*en*: user pools) och specificera vilka rättigheter dessa har.

2.4.3 AWS Lambda

AWS Lambda [27] är en tjänst som tillåter en att ladda upp kod som exekveras på Amazons servrar. I dess marknadsföring läggs stor fokus på att man slipper administrera och hantera servrar. Man behöver inte heller tänka på hur mycket prestanda som behövs då varje anrop exekveras parallellt vilket betyder att hanteringen automatiskt skalar upp baserat på hur många anrop man gör. De första en miljonerna anrop var månad är gratis, därefter kostar anrop i kvantiteter av en miljon.

2.5 iBKS Config Tool

iBKS Config Tool [28] är en mobilapplikation utvecklad av Accent Systems som används för att skanna efter och konfigurera inställningar på iBKS-beacons.

2.6 Leaflet

Leaflet [29] är ett javascript-bibliotek som används för att skapa mobilvänliga och interaktiva kartor. Biblioteket har en öppen källkod vilket har möjliggjort utvecklingen av nya funktioner, så kallade plugins, av tredje parter. Leaflet har fokuserat på kärnfunktionalitet och har ett system som gör det lätt att utvidga funktionaliteten med hjälp av dessa plugins.

2.7 WebView

WebView [30] är en androidklass som skapar ett gränssnitt för att visa innehåll i en webbläsare. WebView använder en motor som heter WebKit för att rendera hemsidans innehåll. [30]

2.8 Canvas

Canvas [31] är en androidklass som används för att rita objekt eller figurer till en bitmap.

2.9 OpenGL ES

OpenGL ES [32] är en grafik-API för inbyggda system som utvecklats av Khronos Group och används för att rendera 2D och 3D-grafik, oftast med hjälp av en grafikprocessor.

3 Arbetsmetod

Detta kapitel beskriver de metoder som använts samt ger en motivering till de val av metoder och verktyg som använts genom att analysera relevant information i examensarbetet.

3.1 Tidsplan

Den initiala tidsplanen kan ses i figur 2.

Det tog betydligt kortare tid att implementera backend än först tänkt då det istället användes en tjänst som inte kräver serverkod för att köras. Implementationen av applikationen tog dock längre tid än förväntat, på grund av att det senare bestämdes att en statistiksida skulle utvecklas. Med detta i åtanke flyttades presentationen fram.

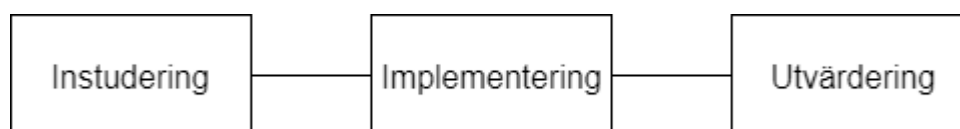
	Vecka	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1 Förstudie, insamling av information						Tentavecka											Presentation
2 Tentaplugg						Tentavecka											Presentation
3 Implementera backend						Tentavecka											Presentation
4 Implementera android-app						Tentavecka											Presentation
5 Skriva examensrapport						Tentavecka											Presentation
6 Utvärdering						Tentavecka											Presentation
7 Förbereda inför presentation, poster						Tentavecka											Presentation

Figur 2. Initiell tidsplan.

3.2 Arbetsprocess

Arbetet skedde till stor del på Knowits kontor i Malmö där examensarbetarna fick tillgång till kontorsplats och egna datorer. Genom att arbeta tillsammans på kontoret blev det lätt att få feedback från varandra och från handledarna med kontinuerlig testning av applikationen. Vid utveckling av applikationen användes versionshanteringsystemet Git via en webbaserad lagringstjänst som heter BitBucket. Detta användes för att kunna utveckla applikationen parallellt utan konflikter och för att ha koll på historiken.

Exjobbet delades upp i tre faser: instudering, implementering och utvärdering (se figur 3).



Figur 3. Faserna som examensarbetet utfördes i.

Instuderingsfasen började med möten med Per Bränner och Johan Lohmander på Knowit för att diskutera och få kunskap om arbetet och möjliga tekniska lösningar. Därefter skedde en insamling och informationssökning på internet efter metoder och tekniker som skulle kunna användas för att implementera idén för prototypen.

Följande är de huvudområden som undersöktes:

- Möjliga positioneringstekniker
- Hur BLE-beacons fungerar
- Hur man scannar efter beacons i Android
- Implementering av karta i Android
- Möjliga lösningar för databasen

Denna information analyserades, testades, diskuterades och mynnade ut i en grov plan för att kunna börja implementera.

Under implementeringsfasen användes agila utvecklingsprinciper där testningen utfördes parallellt med implementeringen. Till att börja med implementerades funktionalitet för att skanna efter beacons och alla beacons konfigurerades. Därefter implementerades kartan med hjälp av Leaflet-biblioteket. En stor del av utvecklingen gick även till att finputsas kartan och interaktionen med skanningen av beacons. Därefter implementerades databasen för att lagra beaconinformation och back-end funktionalitet för att hantera kommunikationen med databasen. Till sist implementerades en statistiksida. Resultatet av implementeringsfasen beskrivs i kapitel 5.

Under utvärderingsfasen utvärderades prototypen med avseende på användbarhet, potential till förbättring och utvecklingsmöjligheter samt etiska aspekter.

4 Analys

I detta kapitel diskuteras de val som har gjorts under examenarbetets gång.

4.1 Implementeringssätt för inomhuspositionering

Det finns flera sätt att implementera ett positioneringssystem med BLE-beacons. Ett sätt att bestämma en användares position är att använda sig av trilateration med hjälp av BLE-paketens signalstyrkor för att beräkna enhetens position. Detta har testats innan i en analys publicerad av R. Faragher och R. Harle. [33] och det har visat sig ha en felmarginal på några centimeter om avståndet till sändaren är under 1 meter. När avståndet från sändaren är 10 meter ligger felmarginalen på ± 5 meter, vilket gör metoden olämplig för trilateration. Detta till stor del på grund av att BLE använder en liten bandbredd för annonsering på samma 2.4 GHz band som redan används av WiFi. Signalstyrkan påverkas även märkbart av fysiska objekt — samt personer — som befinner sig mellan sändare och mottagare, detta innebär att det är svårt att på ett konsekvent sätt konvertera signalstyrka till avstånd.

Ett annat sätt att bestämma användarens position är att placera ut en beacon i varje rum och anta att användaren befinner sig i det rum vars BLE-signal är starkast. Varje beacon-id är då kopplad till ett rum i databasen och applikationen hittar sin position genom att skicka en förfrågan med ett beacon-id.

Med dessa olika sätt i åtanke togs beslutet att applikationen ska fungera så att användarens position antas vara i det rum vars BLE-signal från beaconet är starkast. Med denna lösning uppstår inga problem om mobilen upptäcker flera beacons, men detta bör ändå undvikas för att minimera sannolikheten att fel beacon tycks vara närmast mobilen på grund av signalstyrkans felmarginal. Det kan därför vara lämpligt att ställa in beaconens signalstyrka på en lagom nivå så att signalen inte läcker så mycket att det påverkar intilliggande rum, men samtidigt täcker hela rummet. Detta skiljer sig då från rum till rum då rummets storlek inte är uniform. Hur mycket signalen läcker varierar även beroende på väggarnas egenskaper, såsom tjocklek och material. I dokumentationen för beaconet kan man se hur långt signalen sträcker sig i förhållande till signalstyrkan (se figur 4). Även om den som installerar ett sådant system har dimensionerna för varje rum som beacons ska installeras i så kan det vara en bra idé att testa olika signalstyrkor i varje rum för att se vilket värde som är passande enligt ovan.

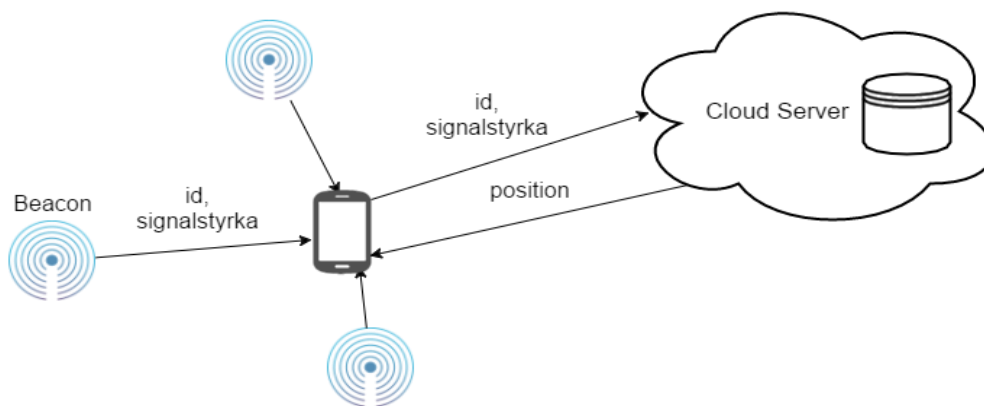
Distance (m)	TX Power (dBm)							
	-30	-20	-16	-12	-8	-4	0	+4
0	-63	-47	-42	-40	-40	-34	-27	-23
1	-92	-79	-71	-68	-66	-58	-59	-57
3	-99	-84	-82	-74	-73	-70	-67	-62
5	-101	-90	-88	-84	-82	-74	-71	-67
10		-95	-95	-89	-87	-81	-79	-75
15		-98	-97	-95	-92	-90	-84	-82
20		-102	-102	-97	-94	-92	-87	-85
30				-102	-96	-93	-92	-88
40					-99	-99	-94	-91
50					-102	-102	-100	-99

Figur 4. Förhållande mellan TX Power (dBm) och distans (m). [34]

Var i rummet beaconen ska vara placerade är också något som är bra att analysera. Det kan vara fördelaktigt att ha dem så långt ifrån beacons i andra rum som möjligt, för att göra det så tydligt som möjligt för applikationen vilket rum som mobilen befinner sig i.

4.2 Kommunikation mellan front- och backend

I instuderingsfasen var tanken att systemet kunde ha varit designat enligt figur 5. Då var tanken att backenden skulle hantera all logik och applikationen endast skulle skicka information och ta emot ett svar.



Figur 5. Tidigt kontextdiagram för koncept.

Eftersom det inte är så mycket beräkningar som sker, bestämdes det att det skulle vara praktiskt lättare att göra beräkningen i applikationen. Beräkningen är relativt enkel, och omfattar att lagra vilket rum som är närmast, alla beacons signalstyrka och ta bort de beacons som inte längre upptäcks från minnet.

4.3 Navigationshjälp

Ett flertal olika metoder för navigationshjälp har undersökts. Att använda sig av pathfinding (ibland kallat *vägplanering* på svenska) har varit i åtanke under examenarbetets gång om det hade funnits tid till det. För att implementera det kan A*-sökalgoritmen användas. Denna är baserad på Dijkstras algoritm och precis som den hittar den kortaste vägen mellan två punkter. Den används i bland annat spel och GPS-navigering. Dock med tanke på tiden det skulle ta att implementera denna funktionalitet och då det inte ansågs ha särskilt hög prioritet bestämdes det att en enklare metod var lämpligare.

Vad som istället bestämdes vara tillräckligt var att markera den plats som väljs på kartan. Detta för att kartan själv bör vara hjälp nog för att hitta målet, särskilt då områdena ett sånt här system kan tänkas täcka förmodligen är ganska litet (som till exempel Knowits kontor som systemet testades i). En relativt enkel funktion att implementera skulle vara en pil som pekar i riktningen från den nuvarande positionen till målets position men även detta ansågs något överflödigt.

4.4 Källkritik

För att analysera källorna som använts i examensarbetet var framförallt följande två frågor i åtanke:

- Vem står bakom källan?
- Varför är källan skapad?

Dessa frågor kommer från Kristina Alexandersons bok “Källkritik på Internet” [35]. De typer av källor som använts i examensarbetet är hemsidor, manualer, artiklar och publikationer.

4.4.1 Primära källor

European Global Navigation Satellite Systems Agency anses vara en pålitlig källa då det är den officiella hemsidan till den europeiska byrån för Global Navigation Satellite Systems (GNSS). Deras uppgift är att bevaka allmänintresset i samband med program för GNSS. [1]

Bluetooth SIG anses vara en pålitlig källa eftersom det är bluetooth-organisationens officiella hemsida. Informationen på hemsidan bör vara korrekt eftersom det är en mängd olika företag som ingår i gruppen som förlitar sig på att hemsidan är pålitlig. [3] [4]

Accent Systems är ett företag som fokuserar på proximitetsteknologi. Sedan BLE-beacons släpptes har de varit en stor tillverkare av dessa. Det är dessa som skapade beaconen som användes i samband med detta examensarbete och eftersom det är deras egen produkt kan man betrakta deras information som pålitlig. [9] [20]

Google specialiserar sig på internetbaserade produkter och tjänster. De är skaparna av Eddystone vilket är BLE-protokollet som användes i detta examensarbete. Därför ansågs det pålitligt att hämta information om detta från Googles officiella hemsidor. [10–18] [22] [30–

32] [38]

Apple designar, utvecklar och säljer konsumentprodukter, mjukvara och onlinetjänster. De är skaparna av iBeacon och det ansågs därför vara pålitligt att hämta information från Apples officiella hemsida för att beskriva detta. [8] [19] [21]

Amazon fokuserar sig på onlinehandel samt molnlösningar. Ett exempel på det sistnämnda är Amazon Web Services (AWS) som är en av tjänsterna som de tillhandahåller. Information om bland annat Lambda, DynamoDB och Cognito hämtades av denna anledning därifrån. Ett blogginlägg från Werner Vogels angående DynamoDB användes även. Han är Chief Technology Officer (CTO) och vicepresident (VP) för Amazon och anses därför ha insikt i AWS. [23–27] [37]

Leafletjs.com är den officiella sidan för Leaflet, vilket är det bibliotek som användes för kartan i detta examensarbete. [29] [36]

4.4.2 Sekundära källor

Infsoft är ett företag som erbjuder lösningar för bland annat inomhusnavigering och platsbaserade tjänster. Företaget har flera kunder, t.ex. Frankfurt flygplats och Vodafone. Deras officiella hemsida anses därför vara pålitlig. [2]

Pointr Labs är ett företag som specialiserar sig på inomhuspositioneringssystem som använder sig av bland annat BLE-beacons och anses därför vara pålitligt för att beskriva dessa. [5]

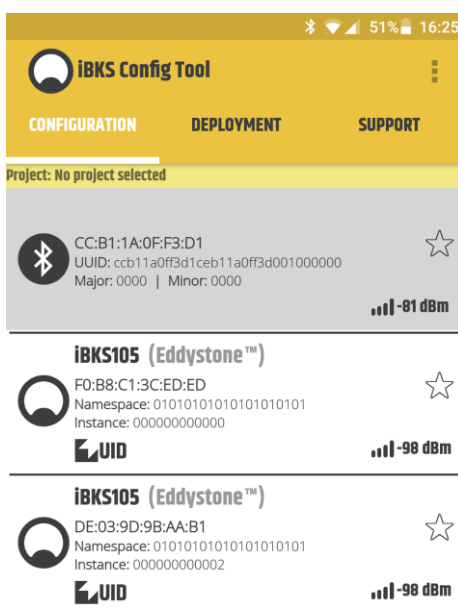
Publikationen “An Analysis of the Accuracy of Bluetooth Low Energy for Indoor Positioning Applications” ansågs vara en pålitlig källa då det är en artikel som är publicerad under University of Cambridge, med pålitliga referenser och med ett syfte att utbilda. [33]

5 Resultat

I detta kapitel visas resultatet av den utvecklade prototypen och lösningen för dess backend.

5.1 Hårdvara

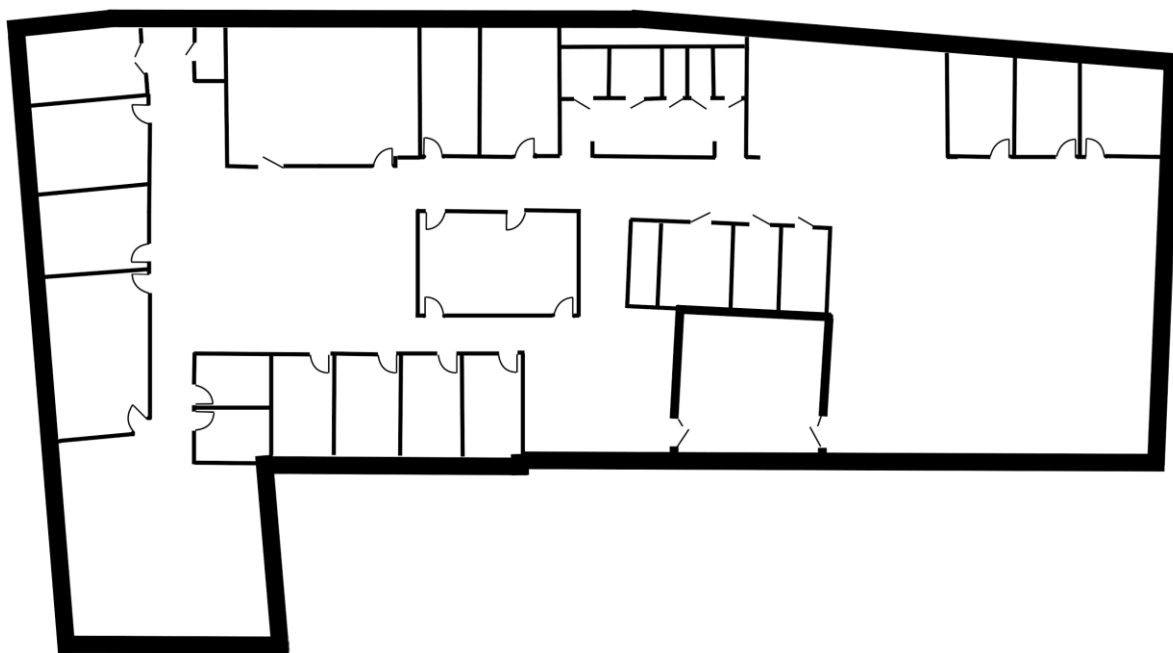
För prototypen användes iBKS 105 som BLE-sändare. För att konfigurera dessa användes iBKS Config Tool vars gränssnitt kan ses i figur 6. För att testa applikationen via Android Studio användes två Androidtelefoner med Android version 7.1.1 samt version 6.0.1.



Figur 6. iBKS Config Tool.

5.2 Mobilapplikationen

Applikationen innehåller en karta över Knowits kontor (se figur 7) som kan användas för att navigera sig på kontoret. Kartan är en avskalad version av kontorets planlösning.



Figur 7. Redigerad planlösning använd som karta i applikationen.

5.2.1 Skanning efter beacons

För att skanna efter beacons i applikationen används Androids egna bibliotek (Android BLE API) för att i detta fall skanna efter Eddystone-UID-paket. De beacons som användes i examensarbetet konfigurerades så att de använde samma namespace ID men ett unikt instance ID. Applikationen filtrerar då efter detta namespace ID:et och ignorerar alla ID som inte passar detta.

5.2.2 Leaflet

Ett flertal lösningar för att implementera kartan övervägdes för att implementera kartan. De som såg lämpligast ut var Leaflet eller en egen implementation med hjälp av Androids grafikkomponent Canvas och OpenGL. Fördelarna med Canvas och OpenGL ES är bättre prestanda och att båda implementeras med Java. Nackdelen med dessa är att det tar tid att implementera funktionalitet för att exempelvis zooma och rotera.

På grund av detta bestämde examensarbetarna att Leaflet var passande för examenarbetets ändamål eftersom denna funktionalitet redan är implementerad och lätt att använda. Leaflet är ett JavaScript-bibliotek för interaktiva kartor med öppen källkod som stödjer mobila plattformar. Det är även möjligt att rita en overlay på kartan vilket passar bra för att markera vilket rum användaren befinner sig i samt för att markera destinationen. Av dessa anledningar ansågs det passa väl för examenarbetet.

I examensarbetet användes en version av Leaflet som heter Leaflet Rotate vilket gör det möjligt att rotera kartan både manuellt (med fingrarna) och automatiskt med hjälp av mobilens kompass. [36]

Kartan visas i applikationen med hjälp av en WebView och kommunicerar med Java-koden

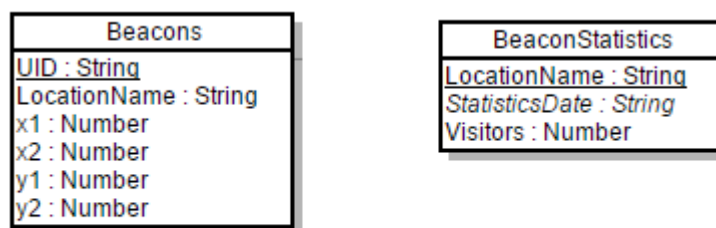
genom en klass som tillåter JavaScript-koden att anropa dess metoder. Det är på detta sätt som det grafiska gränssnittet uppdateras vid behov.

5.2.3 Backend

En backend används för att inte behöva släppa en uppdatering till applikationen var gång informationen om någon beacon ändras eller en ny beacon läggs till. En av de första sakerna som diskuterades och undersöktes i början av examensarbetet var vilken slags tjänst som skulle användas som backend. Tidigt bestämdes det att Amazon Web Services (AWS) skulle passa bra med tanke på dess gratisversion “Free Tier” (se kapitel 2.4). Först togs AWS Lambda i beaktande då denna kan användas tillsammans med AWS andra tjänster såsom deras databaser. I slutändan bestämdes det att detta mellansteg inte behövdes för prototypen utan det skulle gå bra att direkt interagera med databasen via applikationen.

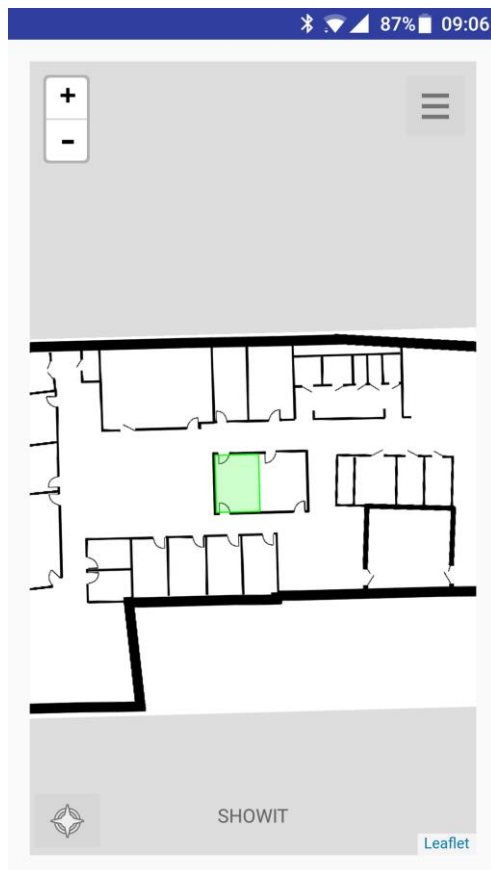
Som databas bestämdes det att DynamoDB tillsammans med Cognito skulle vara passande. De främsta anledningarna är att denna tjänst inte kräver mycket tid för installation och den är även gjord för att vara skalbar, vilket skulle vara till nytta om prototypen i framtiden skulle vidareutvecklas. Datan är även lagrad på Solid State Drives (SSD) vilket gör att hämtningen och skrivningen av datan är snabb. [37]

Det beslutades att två tabeller skulle vara lämpligt: *Beacons* för att koppla relevant information till beaconen och *BeaconStatistics* för att lagra statistik om var plats. I den förstämnda behövs ett namn på platsen som beaconen ska vara i, samt koordinater som platsen befinner sig på i kartan. I den andra tabellen kopplas platsernas namn ihop med antalet besökare vid ett visst datum. Eftersom *StatisticsDate* är en sorteringsnyckel är det möjligt att ha flera rader med samma värde (namn) på platsen. Då DynamoDB inte har stöd för datumformat som attribut var det nödvändigt att istället använda en textsträng i ett förbestämt format (YYYY-MM-DD). Se figur 8 för ett UML-diagram över databasen.



Figur 8. UML-diagram över databasen som användes i examensarbetet. Understrukna rader markerar primärnycklar och kursiv markerar sorteringsnycklar.

Varje gång mobilapplikationen startar laddas samtliga beacons in i applikationens minne. När ett beacon sedan upptäcks jämförs det med de inladdade beaconen och den relevanta informationen används för att visa platsens namn samt markera området på kartan, se figur 9.

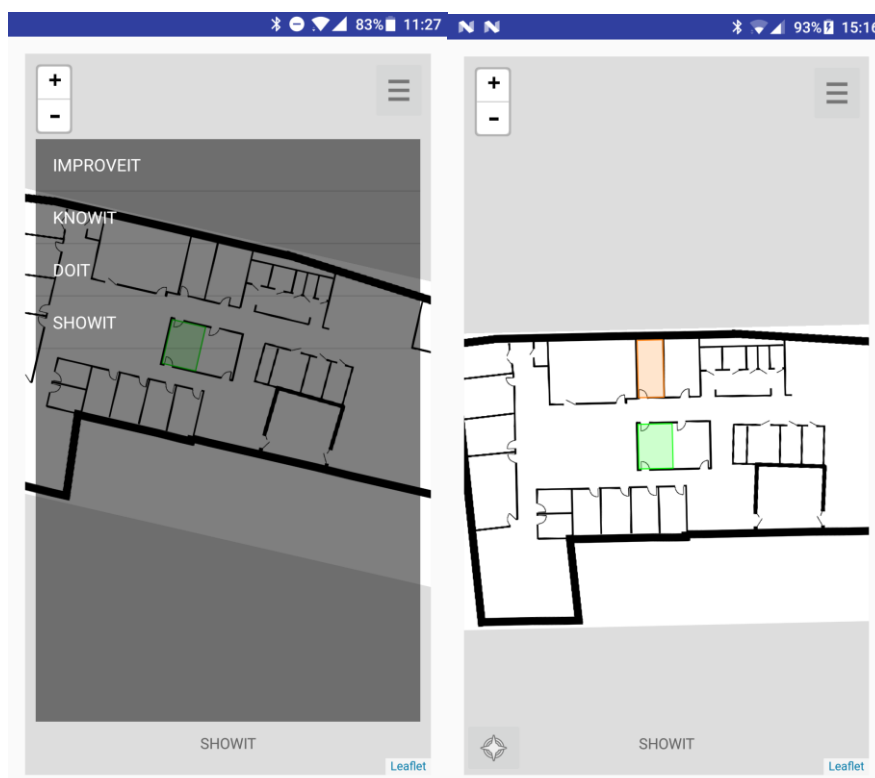


Figur 9. Område där användaren befinner sig markerat i grönt samt platsens namn utskrivet längst ner (SHOWIT).

5.2.4 Gränssnitt

Kartan kan zoomas in och ut genom att antingen dra två fingrar från respektive mot varandra på mobilskärmen, eller genom att använda knapparna längst upp till vänster.

När appen först startas roterar kartan baserat på mobilens kompass. Det går att slå på och av denna funktion genom att klicka på kompassikonen nere till vänster. Det går även att rotera kartan genom att vrida två fingrar med- eller motsols när kompassrotationen är avstängd. Den gröna rutan visar det mötesrum användaren befinner sig i och under kartan visas rummets namn.



Figur 10. Lista med mötesrum som visas när man klickar på menyknappen uppe till höger.

Figur 11. Prototypen med en användare som befinner sig i SHOWIT och med ett rum markerat med orange som mål.

Användaren kan klicka på menyknappen som visas uppe i höger hörn och får då upp en lista på alla mötesrum på kontoret, se figur 10. När användaren klickar på ett rum i listan visas rummet med en orange färg som i figur 11 om användaren inte redan befinner sig i det valda rummet. När användaren förflyttar sig till det valda rummet ändras rutan från orange till grön.

5.2.5 Statistik

Prototypen samlar även grundläggande statistik i samband med att applikationen körs. Den statistik som lagras är antalet besök i var rum per dag, vilket lagras i en separat tabell i databasen. Affärer och liknande skulle kunna använda sådan statistik för att se vilka delar av lokalen som besöks mycket respektive lite.

5.3 Statistiksida

Ett webbgränssnitt har utvecklats för att visa informationen som lagras i statistiktabellen, se figur 12. Statistiksida implementerades som en server med Java och Spring Framework. När användaren först kommer in på sidan visas statistik från en vecka bakåt till dagens datum. Användaren kan skriva in ett eget datumintervall för att visa statistik för detta intervall istället. En tabell genereras som visar datum längst upp med rumnamnen till vänster. En graf visar även samma data visuellt med hjälp av Google Charts [38].

Huvudanledningen till att statistiksidan skapades var för att bevisa att det går att hämta information från databasen externt från applikationen, vilket bevisligen fungerade som förväntat.

Beaconit Statistics

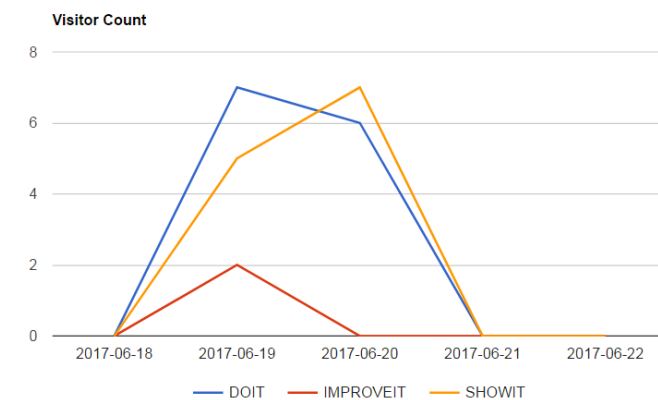
From:

2017-06-18

To:

2017-06-22

☐ Rounded graph



	2017-06-18	2017-06-19	2017-06-20	2017-06-21	2017-06-22	Sum
DOIT	0	7	6	0	0	13
IMPROVEIT	0	2	0	0	0	2
SHOWIT	0	5	7	0	0	12

Figur 12. Gränssnittet för att visa statistiken lagrad i databasen.

6 Slutsats och diskussion

Detta kapitel besvarar de problemformuleringar som ställdes i kapitel 1.3. I kapitlet diskuteras även potentialen för framtida utvecklingsmöjligheter av applikationen och BLE-beacons i allmänhet, samt de etiska aspekter som applikationen kan föra med sig.

6.1 Problemformulering

Följande är de problemformuleringar som ställdes i examensarbetet.

- *Hur många Bluetooth Low Energy beacons krävs för att med hög precision kunna hitta sin position och möjliggöra navigering mellan olika rum på Knowits kontor?*

På sättet det implementerades i examensarbetet krävs det i de flesta fall ett beacon per rum som man vill ha med i systemet. Ifall rummet är för stort eller har en form som kräver mer täckning skulle det kunna användas flera beacons som ställs in med samma Eddystone-UID och därmed kopplas till samma rum i databasen.

- *På vilka sätt kan man med hjälp av BLE-beacons få reda på var i en byggnad mobilen befinner sig? Hur precisa är de olika metoderna?*

Ett sätt är att använda sig av trilateration med hjälp av minst tre beacons signaler. Detta har dock testats innan, vilket har visats kan ha flera meter felmarginal, [28] och därför valdes det att implementeras på ett alternativt sätt i examensarbetet. Man placerar ett beacon i varje rum som man vill ha med i systemet. Mobilen anses sedan vara i det rum vars beaconsignal den tar emot starkast. Denna metod kan alltså inte sägas ha en felmarginal på samma sätt, utan om det blir fel estimerar den att man befinner sig i ett rum som man egentligen inte befinner sig i. Detta har dock inte visat sig vara ett problem om beaconens signalstyrka ställs in på en lämplig nivå som beskrivs i kapitel 4.1 och i sista frågeställningen.

- *Vad finns det för möjligheter med anseende på implementering av navigation till mål inom byggnaden?*

Ett sätt hade varit att implementera navigation till mål med hjälp av t.ex. A*-sökalgoritmen. På grund av brist på kunskap inom detta område och tidsbrist ansågs det olämpligt för examensarbetet.

Ett annat sätt hade varit att visa riktningen till målet med t.ex. en pil. Detta ansågs vara överflödigt och inte tillräckligt användbart för att implementeras.

Istället för dessa metoder ansågs det vara tillräckligt att markera det nuvarande rummet och rummet användaren ska navigera sig till på kartan.

- *Vilken inställning på signalstyrkan är lämplig för de olika positioneringsmetoderna?*

Med den implementationsmetod som valdes är en lämplig signalstyrka stark nog för att täcka det rum som beaconet befinner sig i men inte läcker ut och påverkar intilliggande rum. Inställningen kan alltså variera från rum till rum.

Om man använder trilateration kan det vara lämpligt att ha en så stark signalstyrka som möjligt för att säkerställa att mobilen får signal från så många beacons som möjligt och att en så stor yta som möjligt täcks av systemet.

6.2 Framtida utvecklingsmöjligheter och förbättringar

Den utvecklade prototypen är möjlig att vidareutvecklas för olika ändamål eller hämta inspiration och kunskap från. Möjliga vidareutvecklingar skulle kunna vara ett bokningssystem, en integration till ett tidrapporteringssystem för automatisk tidrapportering eller att samla statistik om vilka avdelningar som har flest besökare i en affär. En annan vidareutveckling kunde varit att använda pathfinding för att visa närmaste vägen till ett annat rum.

Prototypen skulle även kunna förbättras på olika sätt, exempelvis genom att göra kartkomponenten helt och hållet i Java med en grafikkomponent, till exempel Canvas eller OpenGL ES. Då skulle man slippa JavaScript-delen och dess gränssnitt vilket komplicerar ändringen av kartan. Detta hade även gett applikationen bättre prestanda och gjort det lättare att implementera pathfinding för navigering.

6.3 Etiska aspekter

Prototypen som utvecklades i samband med detta examensarbete samlar endast anonym statistik som inte kan kopplas till användare. Endast information om hur många som besöker vart mötesrum var dag lagras. Men detta system eller ett liknande skulle kunna samla information om var varje användare befinner sig. Detta skulle kunna ses som att det begränsar användarnas integritet och bör isåfall ha tydligt specificerats i användarvillkor.

Samtidigt kan statistik vara väldigt användbar för ägaren till lokalen (exempelvis ge en uppskattning av hur många mötesrum som behövs) och därför kan det ses som att det handlar om en avvägning mellan detta och användarens integritet. En annan positiv aspekt är att prototypen skulle kunna vidareutvecklas och till exempel göra det lättare att hantera bokning av mötesrum.

7 Referenser

- [1] European Global Navigation Satellite Systems Agency. What is GNSS?. 2016-11-25.
<https://www.gsa.europa.eu/european-gnss/what-gnss> (Hämtad 2017-06-22)
- [2] infsoft. Indoor Navigation Using Wi-Fi as a Positioning Technology. 2015-09-29
<https://www.infsoft.com/blog-en/articleid/40/indoor-navigation-using-wifi-as-a-positioning-technology> (Hämtad 2017-08-03)
- [3] Bluetooth SIG. Our History.
<https://www.bluetooth.com/about-us/our-history> (Hämtad 2017-05-26)
- [4] Bluetooth SIG. Bluetooth® 5 Quadruples Range, Doubles Speed, Increases Data Broadcasting Capacity by 800%. 2016-06-16.
<https://www.bluetooth.com/news/pressreleases/2016/06/16/-bluetooth5-quadruples-rangedoubles-speedincreases-data-broadcasting-capacity-by-800> (Hämtad 2017-05-26)
- [5] Olcay. Beacons: Everything you need to know.
<http://www.pointrlabs.com/posts/beacons-everything-you-need-to-know/> (Hämtad 2017-08-03)
- [6] Accent Systems. iBKS 105.
<https://accent-systems.com/product/ibks-105/> (Hämtad 2017-08-03)
- [7] Google. Eddystone format. 2017-07-05.
<https://developers.google.com/beacons/edystone> (Hämtad 2017-08-11)
- [8] Apple. iBeacon.
<https://developer.apple.com/ibeacon/> (Hämtad 2017-08-11)
- [9] Radius Networks. AltBeacon.
<http://altbeacon.org/> (Hämtad 2017-08-11)
- [10] Google. Eddystone Protocol Specification. 2016-04-14.
<https://github.com/google/edystone/blob/master/protocol-specification.md> (Hämtad 2017-05-26)
- [11] Google. Eddystone-UID. 2016-08-11.
<https://github.com/google/edystone/tree/master/edystone-uid> (Hämtad 2017-05-26)
- [12] Google. Eddystone-URL. 2016-09-28.
<https://github.com/google/edystone/tree/master/edystone-url> (Hämtad 2017-05-26)
- [13] Google. UriBeacon.
<http://google.github.io/uribeacon/> (Hämtad 2017-05-26)

- [14] Google. Unencrypted TLM Frame Specification. 2016-04-14.
<https://github.com/google/eddystone/blob/master/eddystone-tlm/tlm-plain.md> (Hämtad 2017-05-26)
- [15] Google. Encrypted TLM Frame Specification. 2016-04-14.
<https://github.com/google/eddystone/blob/master/eddystone-tlm/tlm-encrypted.md> (Hämtad 2017-05-26)
- [16] Google. Eddystone-EID. 2016-04-20.
<https://github.com/google/eddystone/tree/master/eddystone-eid> (Hämtad 2017-05-26)
- [17] Google. Nearby.
<https://developers.google.com/nearby/> (Hämtad 2017-05-28)
- [18] Google. Overview. 2017-02-09.
<https://developers.google.com/beacons/proximity/guides> (Hämtad 2017-05-28)
- [19] Apple. Core Bluetooth. 2013.
<https://developer.apple.com/videos/play/wwdc2013/703/> (Hämtad 2017-08-11)
- [20] Radius Networks. AltBeacon Technical Specification.
<https://github.com/AltBeacon/spec> (Hämtad 2017-06-22)
- [21] Apple. Getting Started With iBeacon. 2014-06-02.
<https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf> (Hämtad 2017-06-22)
- [22] Google. Android Studio.
<https://developer.android.com/studio/index.html> (Hämtad 2017-08-10)
- [23] Amazon.
<https://aws.amazon.com/> (Hämtad 2017-08-10)
- [24] Amazon. AWS Free Tier.
<https://aws.amazon.com/free/#legal> (Hämtad 2017-06-22)
- [25] Amazon. Amazon DynamoDB.
<https://aws.amazon.com/dynamodb/> (Hämtad 2017-08-11)
- [26] Amazon. Amazon Cognito.
<https://aws.amazon.com/cognito/> (Hämtad 2017-08-10)
- [27] Amazon. AWS Lambda.
<https://aws.amazon.com/lambda/> (Hämtad 2017-08-11)
- [28] Accent Systems. iBKS Config Tool.
https://play.google.com/store/apps/details?id=com.accent_systems.ibks_config_tool (Hämtad

2017-08-10)

[29] Agafonkin, Vladimir. Leaflet.
<http://leafletjs.com/> (Hämtad 2017-08-11)

[30] Google. WebView.
<https://developer.android.com/reference/android/webkit/WebView.html> (Hämtad 2017-08-11)

[31] Google. Canvas.
<https://developer.android.com/reference/android/graphics/Canvas.html> (Hämtad 2017-08-27)

[32] Google. OpenGL ES.
<https://developer.android.com/reference/android/opengl/package-summary.html> (Hämtad 2017-08-27)

[33] Faragher, R., Harle, R. (2014). An Analysis of the Accuracy of Bluetooth Low Energy for Indoor Positioning Applications.
<http://www.cl.cam.ac.uk/~rmf25/papers/BLE.pdf> (Hämtad 2017-06-13)

[34] Accent Systems. iBKS105 Datasheet Revision 3. Januari 2017.
https://accent-systems.com/wp-content/uploads/iBKS105_datasheet_rev3.pdf (Hämtad 2017-06-19)

[35] Alexandersson, Kristina. *Källkritik på Internet*. Ödeshög: DanagårdsLiTHO, 2012.
<https://www.iis.se/fakta/kallkritik-pa-internet/> (Hämtad 2018-08-03)

[36] Nicollet, Fabien. Leaflet rotate. 2017-06-02.
<https://github.com/fnicollet/Leaflet/tree/rotate-master> (Hämtad 2017-06-28)

[37] Vogels, Werner. Amazon DynamoDB – a Fast and Scalable NoSQL Database Service Designed for Internet Scale Applications. 2012-01-18.
<http://www.allthingsdistributed.com/2012/01/amazon-dynamodb.html> (Hämtad 2017-08-04)

[38] Google. Display live data on your site.
<https://developers.google.com/chart/> (Hämtad 2017-08-11)

8 Terminologi

AltBeacon	Ett BLE-protokoll släppt i juli 2014 av Radius Networks.
API	Application Programming Interface - Ett antal publika metoder eller variabler som fungerar som gränssnitt till exempelvis ett bibliotek med kod eller kommunikationen mellan program.
AWS	Amazon Web Services. Diverse molnlösningar såsom databaser tillhandahållet av Amazon.
Back-end	Severn (databasen) samt den del av mjukvaran som hanterar kommunikationen mellan databasen och klienten.
Beacon	I teknologiska sammanhang menas ofta små sändare med huvudsyfte att skicka ut information.
BLE	Bluetooth Low Energy - En mindre energikrävande version av Bluetooth.
BLE-beacon	Små sändare som skickar ut signaler enligt Bluetooth SIG:s specifikation för BLE.
DynamoDB	En del av Amazon Web Services (AWS). Databaslösning i molnet.
Eddystone	Ett BLE-protokoll släppt i juli 2015 av Google.
Front-end	Presentationslagret i en mjukvara. I detta fall refererar det till klienten i klient-servermodellen.
iBeacon	Ett BLE-protokoll släppt i december 2013 av Apple.

Trilateration

En metod för att beräkna skärningspunkten mellan tre sfäriska ytor. Kan användas till exempel för att beräkna positionen för en enhet som får signal från minst tre olika källor.



LUND
UNIVERSITY

Series of Bachelor's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2017-599
<http://www.eit.lth.se>