Automatisk närvaroregistrering

Innehållsförteckning

[1 Inledning 3](#_Toc490641865)

[2 Teknologier 3](#_Toc490641866)

[2.1 BLE 3](#_Toc490641867)

[2.2 iBeacon 3](#_Toc490641868)

[2.3 AltBeacon 4](#_Toc490641869)

[2.4 Eddystone 4](#_Toc490641870)

[2.5 UART over BLE 5](#_Toc490641871)

[2.6 Arduino 5](#_Toc490641872)

[2.7 Estimote 5](#_Toc490641873)

[2.8 Fingeravtrycksläsare 5](#_Toc490641874)

[2.9 Programmeringstekniskt 6](#_Toc490641875)

[3 Setups 6](#_Toc490641876)

[3.1 Mobil <–> Mobil 6](#_Toc490641877)

[3.2 Fast hårdvara <–> Mobil 7](#_Toc490641878)

[3.3 Fingeravtrycksläsare 7](#_Toc490641879)

[4 Sammanfattning/rekommendationer 8](#_Toc490641880)

[5 Rekommendationer 8](#_Toc490641881)

Automatisk närvaroregistrering

# Inledning

I det här arbetet har vi undersökt och testat hur det skulle fungera att använda sig av automatiserad närvaroregistrering i skolan (grundskola och gymnasium) med hjälp av beacons eller en fingeravtrycksläsare.Vi har testat att använda en Arduino med en BLE Shield som beacon och även hur det fungerar att ha en mobil som beacon.

En automatiserad närvaroregistrering skulle förhoppningsvis underlätta det administrativa arbetet för lärare och ge möjlighet till att skicka notiser omgående till föräldrar vid ogiltig frånvaro av en elev. I dagsläget måste läraren registrera elevernas närvaro i efterhand. Antingen efter varje lektion eller i slutet av dagen, vilket gör att en notisfunktion blir svårt. Lärarna skulle kunna lägga mindre tid på att närvaro- och frånvaroregistrera elever och istället lägga tiden på andra saker, som till exempel undervisningen.

Syftet med rapporten är att visa vad vi kommit fram till i arbetet (Proof of Concept), som Tieto sedan kan ta del av och få idéer till sin app “Tieto Education”. För att testa beaconfunktionerna skapade vi en simpel testapp med knappar för att stänga av och på advertising mode och scanning av beacons med mobilen.

Rapporten börjar med en introduktion till de teknologier vi har använt oss av eller stött på i arbetet. Efter introduktionen diskuteras det om hur Arduino kan användas som beacon mot en mobil och därefter mobil agerande som beacon mot en annan mobil. Svårigheter och möjligheter med de båda alternativen diskuteras också. Efter det kommer ett avsnitt om fingeravtrycksläsaren och vad vi kom fram till med den. Rapporten avslutas sedan med en sammanfattning över vad vi tycker är det bästa alternativet och övriga praktiska aspekter som är viktiga att ta hänsyn till.

# Teknologier

## BLE

Bluetooth beacons är en typ av sändare som använder Bluetooth Low Energy (BLE, en energisnålare version av Bluetooth) för att sända ut data, oftast en identifierare, till enheter i närheten. BLE kan överföra data på två olika sätt: anslutet eller annonserande. Anslutet läge använder Generic Attribute (GATT) lagret för att skapa en en-till-en anslutning till en enhet för att överföra data. Annonserande läge använder Generic Access Profile (GAP) lagret för att sända ut data till alla lyssnande enheter. Det är följaktligen ett en-till-många kommunikationssätt utan något sätt att bekräfta att datan kommit fram korrekt.

## iBeacon

Det finns många olika typer av implementationer och protokoll när det kommer till beacons. Den första beacon som kom ut på marknaden var Apples iBeacon. iBeacon är ett proprietärt protokoll (som vanligt när det kommer till Apple 😄). Mängden data man kan överföra är ganska begränsad med iBeacon-protokollet; det finns endast 21 bytes att redigera. De första 16 är ett UUID, efter det kommer 2 bytes som kallas major identifier, efter det 2 bytes som kallas minor identifier och till sist den byte med TxPower. Ett 16-bitars UUID är enligt standard bit 17 till bit 32 i ett fullt 128-bitars UUID och används i vanliga fall för att identifiera företaget som använder beaconen. Fältet som kallas major kan ha ett värde mellan 0 och 65535 och används oftast till att identifiera vilken affär eller avdelning inom företaget som har beaconen. På samma sätt används minor till att identifiera en beacon inom affären eller avdelningen. Dessa 20 bytes kan självklart användas på valfritt sätt för att identifiera beaconen och ge någon information till mottagaren. Hur de än används måste mottagaren komma åt en databas eller liknande för att kolla upp vilken mening de olika värdena har. Till sist används TxPower för att ange den signalstyrka enheten har på en meters avstånd, något som måste mätas per enhet, men för många mobiler är värdet kring -70. Värdet används sedan av mottagaren för att bestämma avståndet till sändaren. Eftersom Apple står bakom har iOS inbyggt stöd för att hantera iBeacons, både sända och mottaga. Andra operativsystem har inte inbyggt stöd för iBeacons utan där måste programmerare själv skriva kod för att hantera protokollet med de inbyggda BLE-funktionerna.

## AltBeacon

En annan, öppen, standard är AltBeacon. Det är små skillnader i AltBeacon-protokollet mot för hur iBeacon fungerar. AltBeacon har ett par fler bytes tillgängliga för redigering som tidigare var reserverade av Apple. Dessa bytes kan man till exempel använda för att identifiera företaget och sedan använda UUID:et till annan information. Annars fungerar det på så sätt att man fortfarande måste använda en applikation och/eller databas för att ge mening åt de värden som överförts. Även AltBeacon inkluderar en TxPower byte på samma sätt som iBeacon så det går att räkna ut avståndet mellan sändare och mottagare. Har inget inbyggt stöd i något operativsystem, men det finns bibliotek till många OS för enkel hantering av sändning och mottagning av AltBeacons.

## Eddystone

Eddystone är ett öppet beacon-protokoll från Google som skiljer sig lite från de andra två stora standarderna. Eddystone beacons har inte bara en typ av frames att skicka, en identifierare, de har istället fyra olika typer av frames. De första två heter Eddystone-UID och Eddystone-EID och fungerar som identifieraren som finns i de andra protokollen; den enda skillnaden mellan de två är att en är krypterad. Nästa frame heter Eddystone-TLM och sänder telemetri-information om sändaren. Detta kan till exempel användas för att ta emot information om hur mycket batteri som finns och liknande som kan behövas vid diagnostisering av en sändare. Sista typen av frame heter Eddystone-URL och är det som särskiljer protokollet mest från de andra. Detta är 19 bytes som sänder en URL som mottagaren kan öppna automatiskt i en webbläsare när den tar emot signalen. För att spara på de begränsade bytes som finns att tillgå kodas både prefix (t.ex. http://www.) och suffix (t.ex. .com) som en byte. Detta gör att man bara kan använda vanliga protokoll och toppdomäner, men det är oftast inte ett problem. Anrop till längre URL:er kan enkelt lösas med att använda URL-förkortare. T.ex. <http://api.example.com?param1=true&param2=false> kan förkortas med hjälp av t.ex. goo.gl eller bit.ly och lätt passa in på de 19 bytes som finns att tillgå. En beacon kan sända en eller flera typer av frames samtidigt. Detta gör att det inte finns ett behov för mottagaren att kunna ansluta till en databas eller liknande för att få reda på vad den ska använda identifieraren till. Eddystone är officiellt utvecklat för att enkelt kunna användas på både iOS och Android.

## UART over BLE

UART står för Universal Asynchronous Receiver/Transmitter och namnet förklarar vad det betyder. UART over BLE innebär att data skickas och tas emot över Bluetooth Low Energy. I vårt projekt har vi tittat snabbt på möjligheterna med att använda detta för  tvåvägskommunikation över Bluetooth istället för att bara ha sändare och mottagare. Eftersom denna tvåvägskommunikation är en-till-en dömde vi dock bort den väldigt fort då det inte är rimligt i de scenarion vi undersökte för.

## Arduino

En Arduino är ett litet mikrokontrollerkort (enkretsdator), det vill säga en liten dator med CPU, arbetsminne och programminne integrerat komplett med stödfunktioner. En Arduino ska vara enkel att använda, programmera och har sin egna utvecklingsmiljö som gör det lätt att skriva kod och ladda upp det till kortet. Arduino och Arduinokompatibla kort använder sköldar eller så kallade Shields, vilket är kretskort som monteras ovanpå en Arduino för att få ytterligare funktioner. För att testa använda Arduinon (Arduino Uno) som en beacon kopplades en Bluetooth Low Energy Shield (RedBearLabs low energy shield v2.1) som tillåter att koppla Arduinon till andra BLE enheter. Enheten skickar ständigt ut beaconsignal som man kan bestämmas i Arduinokoden. En annan BLE-enhet kan sedan ta emot signaler från beaconen.

I arbetet började vi först att testa ifall mobilen kunde läsa beaconen, vilket var inga problem. Med hjälp av olika appar kunde vi läsa av beaconen enkelt. Nästa steg blev att testa UART (universal asynchronous receiver/transmitter), det vill säga, skicka och ta emot data. Vi kunde då med mobilen koppla till Arduinon och ta emot ett simpelt meddelande och skicka något tillbaka. Tanken med det är att kunna skicka ett unikt ID (eller något) från mobilen till Arduinon som sedan gör ett API-anrop. Vi lyckades anropa ett API för att visa att det går.

## Estimote

Är det önskat att ha en fast sändare är ett alternativ till Arduino så kallade Estimote beacons. Dessa beacons är utvecklade av företaget [Estimote](http://estimote.com/) för att sättas upp och sända iBeacon- eller Eddystone-paket. De är små och enkla att gömma i något hörn. För att ställa in vad de ska sända för paket finns ett gränssnitt man kan nå med mobilen. Dessa skulle då till exempel kunna sända paket av typen Eddystone-URL som mottagarna öppnar och den URL som sänds skulle kunna uppdateras inför varje lektion. Ett antal olika varianter med olika batteritid och räckvidd finns att köpa.

## Fingeravtrycksläsare

Ett annat alternativ till beacons om vi kollat på är fingeravtrycksläsare. Den modell som vi undersökt heter [Adafruit Fingerprint Sensor](https://www.adafruit.com/product/751) och måste kopplas till en Arduino. (Det finns även [dedikerade fingeravtrycksläsare som går att köpa relativt billigt](https://www.alibaba.com/product-detail/3000-fingerprint-capacity-10000-EM-card_60552369302.html), men vad de har för begränsningar har vi inte kunnat testa.) Fingeravtrycksläsaren vi testat kan läsa, matcha och spara upp till 162 fingeravtryck på fingeravtrycksläsarens inbyggda flash-minne. Fingeravtryck registreras genom att välja ett ID och sätta ett finger mot läsaren två gånger. Fingeravtrycksläsaren extraherar features från fingeravtrycket som den använder för att identifiera det. Sedan sparas det tillsammans med det givna ID:et som sedan returneras vid läsning. Formatet det sparas i och jämförs med finns det varken öppen dokumentation om eller är läsbart av människor. Det gör att matchning måste ske på läsaren om man inte själv vill försöka skriva någon statistisk modell som kan matcha två sparade templates. Det går också att ta bort registrerade fingeravtryck.

Trots att den “bara” är något man kopplar in i en Arduino uppfyller läsaren false positive och false negative kraven för [level 3 security](https://en.wikipedia.org/wiki/FIPS_140-2) (ej en specifikation, bara lite mer läsning). Detta betyder att den har <0,001% false positives och <1% false negatives. Alltså är det inte ett problem att elever inte skulle bli registrerade som närvarande. Vissa skulle även kunna tycka att det är ett problem att skolan lagrar fingeravtryck. Som tur är lagras aldrig någon bild av det faktiska fingeravtrycket, utan bara features i ett format som bara denna läsare förstår.

Det finns ett Adafruit Fingerprint Sensor bibliotek för Arduino som gör det enkelt att använda. I biblioteket finns det exempelkod för att lägga till, läsa och ta bort fingeravtryck.

## Programmeringstekniskt

Det finns många bra plugins och bibliotek för utveckling av beacon-applikationer till mobil eller Arduino. Till Arduino finns tillgängliga bibliotek för BLE i den inbyggda Library Manager. För att utveckla till mobil finns det antingen native-stöd för den typen av beacons, som iBeacon till iOS, eller så är finns det bibliotek att ladda ner för att enklare sätta upp en beacon, som AltBeacon till Android. Om det av någon anledning inte skulle finnas möjlighet att använda bibliotek kan utvecklaren själv skapa de paket som ska sändas över BLE enligt specifikationen för det protokollet. För Cordova som används på Tieto för att utveckla appen Tieto Education finns det ett par plugins för beacons, men de är alla senast uppdaterade för mycket länge sedan och har därför väldigt begränsad funktionalitet. Inget plugin har stöd för att sända från en Android telefon och endast ett plugin har (utdaterat) stöd för att sända som en iBeacon från iOS. För att ta emot finns det däremot plugins för alla de stora beacon-protokollen och att lägga till funktionalitet till dem för att sända är med de resurser som finns idag en rätt enkel uppgift.

# Setups

## Mobil <–> Mobil

En möjlighet för att sätta upp den kommunikation som krävs för att lösa den automatiska närvaroregistreringen vi undersökt är att använda mobiler. En möjlig konfiguration är att läraren har en mobil där hen kan starta en beacon-sändning som sedan eleverna kan ta emot i sina mobiler. Beacon-sändningen skulle kunna vara Eddystone-URI och -URL paket för att identifiera mobilen så att eleverna bara tar emot sändningar från rätt lärare samt får en färdig URL som anmäler närvaron när den öppnas. Att anropa ett API från mobilen är väldigt lätt vilket är en fördel. Att sändningen ska, helst helt automatiskt, innehålla rätt information om vilken lektion som pågår är en utmaning med ett sådant här system. Andra praktiska utmaningar att lösa är att kunna hantera vikarier och lärarbyten, att man egentligen inte kan förutsätta att alla lärare och elever har en mobil, och att eleven som är inloggad faktiskt är den elev som är där. En möjlig lösning på det sistnämnda skulle kunna vara tvåfaktorsautentisering samt att man förlitar sig på att en elev inte skolkar utan sin mobil. Lärare borde även kunna ha möjlighet att manuellt ta bort närvaro från en elev som på något sätt fuskat.

Något vi kom på sista stund är att de flesta elever och lärare får laptops i skolan idag som det finns Bluetooth på. Då vi kom på det i sista sekund har vi inte utforskat begränsningarna lika noggrant som på mobilen, t.ex. om man från en hemsida kan starta Bluetooth-scanning. I detta fall gäller dock ungefär samma förutsättningar som för mobil-till-mobil, men det förlitar sig inte på att alla elever och lärare har mobiler.

## Fast hårdvara <–> Mobil

För att möjligtvis automatisera ännu mer skulle det kunna sättas upp hårdvara i varje klassrum som kan signalera som en beacon till t.ex. elevernas mobiler så att de automatiskt anmäls som närvarande när de kommer in i klassrummet. Detta skulle ta bort kravet på att läraren behöver en mobil och att läraren måste starta sändningen från mobilen. Här finns dock återigen problemet med att hantera att mottagaren bara reagerar på rätt sändare. Att installera och underhålla hårdvara i varje klassrum i en kommun är dock både kostsamt och tidskrävande, även om man bara använder Estimote beacons som kan snabbt klistras fast på väggen. Skulle ett klassrum sluta fungera skulle det kräva mer manuellt jobb av läraren i väntan på reparationer. Att använda en Estimote i varje klassrum skulle även kräva någon till enhet för att uppdatera dess sändning då den själv inte har internetuppkoppling (inget sätt att nå lektioners id). En Arduino skulle kräva en adapter för att kunna ansluta till internet. Dessa skulle även behöva förvaras säkert så inte någon kan förstöra dem, i en olycka eller med vilje.

## Fingeravtrycksläsare

Fingeravtrycksläsare har möjligheten att vara ett mycket bra system för närvaroregistrering, men har problem med bland annat skalbarhet i nuläget. Den fingeravtrycksläsare som vi testat var enkel att registrera fingeravtryck i och bra på att läsa av dem. Eftersom den i princip aldrig gör något fel (<0,001% false positives) eliminerar den problemen som finns hos mobiler med att identifiera rätt person och manuella korrektioner av närvaro från lärare. Systemet skulle då heller inte förlita sig på att elever eller lärare alltid måste ha med sig mobiler eller annan hårdvara. Denna läsare skulle dock kräva en Arduino med internetuppkoppling för att göra närvaroregistreringen vilket skulle, som tidigare nämnt, vara kostsamt att installera och underhålla i alla klassrum. Tyvärr så finns det även ett ännu större problem. Lösningen skalar i princip inte alls utöver en läsare för en klass i ett klassrum. Eftersom matchningen är inbyggd i läsaren och mycket svår att implementera någon annanstans skulle alla elever behöva registrera sig på alla läsare. I denna läsares fall skulle det betyda att det inte kan vara fler än 162 elever som har lektion i någon sal. Om man kringgår det genom att skapa en databas där man lagrar fingeravtryckens templates så att ingen läsare lagrar dessa själv skulle matchningen fortfarande behöva göras på läsaren. Då skulle de behöva (i värsta fall) hämtas från databasen varje fingeravtryck på skolan. Det skalar inte alls bra. Man skulle dock möjligtvis kunna lösa problemet med någon heuristik som gör att varje läsare bara hämtar de templates den oftast matchar mot. Nackdelen då är att det skulle ta mycket längre tid att registrera närvaron när en klass har lektion mer sällan i ett visst klassrum.

Att använda dedikerade fingeravtrycksläsare med nätverkskapacitet och större minne istället för en Arduino skulle också kunna vara en lösning, som också skulle kunna vara billigare, men vad det finns för andra begränsningar där kan vi inte säga då vi inte ha testat dem.

# Sammanfattning/rekommendationer

I arbetet har vi undersökt tre olika sätt att kunna automatisera närvaroregistreringen i skolan:

1. Beacon - Arduino till mobil
2. Beacon - Mobil till mobil
3. Fingeravtrycksläsare

Alla alternativ har olika praktiska fördelar och nackdelar, vilket gör dem mer eller mindre lämpliga att använda. Utifrån arbetet och vad vi har kommit fram till tror vi att alternativ två, (beacon - mobil till mobil) är lämpligast av dessa. Arduino och fingeravtrycksläsaren skulle kräva för mycket vid installation, underhåll och uppdateringar, det är även många fler faktorer som kan skapa problem och krångla.

# Rekommendationer

Vi vet nu att automatiserad närvaroregistrering är möjligt men att det är många praktiska saker som är viktiga att tänka på. Något som kan vara viktigt är att undersöka ifall automatiseringen bidrar med mer nytta än vad kostnaden för att utveckla programvaran för att hantera närvaroregistreringen. Även om själva registreringen är automatisk är det troligt att lärarna fortfarande måste registrera speciella undantag av frånvaro. Därför blir närvaroregistreringen inte helt automatiserad och då kanske det skapar mer jobb än att underlätta.