Location, Location, Communication

Invenio Solutions

Elias Andersson
Bastian Fredriksson
Tobias Hindersson
Tom Johansson
Joel Kallin Kilman
Joakim Lindberg
Elias Lousseief
Edvin Lundberg
Marcus Löf
Martin Steier

Version 2.3 (second draft)

2014-04-24

Abstract

I detta dokument går vi i detalj igenom vilka de tänkta användarna av vår applikation är, och analyserar utifrån detta vilka krav som ställs på vår applikation. Vi tar även upp hur den nuvarande miljön ser ut för användarna och vilka externa system vår applikation behöver interagera med. Utifrån alla krav och beroenden beskriver vi vilka funktioner vår applikation behöver innehålla för att tillfredsställa användarnas behov och fungera i den nuvarande miljön.

Table of contents

1 Introduction 1
1.1 Purpose 1
1.2 Scope of the software 1
1.3 Definitions, acronyms and abbreviations 1
1.4 References 3
1.5 Overview of the document 4
2 General Description 5
2.1 Product Perspective 5
2.2 General Capabilities 5
2.3 General constraints 7
2.4 User characteristics 9
2.5 Assumptions and dependencies 17
2.6 Operational environment 21
3 Specific Requirements 26
3.1 Capability requirements 27
3.2 3.2 Constraint requirements 36
Appendix A – Data Constraints Diagram 42
Appendix B – Architecture Diagram 43
Appendix C – Requirements Diagram 44

Document Change Record

Datum	Version	Vad ändrades?
2014-02-22	0.1	Lade till Kapitel 2.3.
2014-02-23	0.2	Lade till Kapitel 2.5.
2014-02-24	0.3	Lade till Kapitel 3.1 och Kapitel 3.2.
2014-02-25	0.4	Lade till Kapitel 2.1 och Kapitel 2.2.
2014-02-27	0.5	Lade till Appendix A.
2014-02-28	0.6	Lade till Kapitel 2.6.
2014-02-28	0.7	Lade till Kapitel 1.1, Kapitel 1.2 och Kapitel 2.4.
2014-02-28	0.8	Lade till Appendix B och Appendix C.
2014-03-01	1.0	• Skrev om och korrigerade i stort sett alla kapitel. • Ändrade storleken på bilderna.
2014-03-01	1.1	Korrigerade några mindre språkfel.
2014-03-04	1.2	• Utökade Kapitel 1.3, Kapitel 2.2, Kapitel 2.4 och Kapitel 3. • Gjorde prototypen i Appendix C mer detaljerad. • Lade till Appendix D. • Lade till flera nya diagram. • Förbättrade bildkvaliteten på vissa bilder.
2014-03-05	1.3	Ändrade formateringen i Kapitel 3.
2014-04-20	2.0	Förändrade Kapitel 2 så att det stämmer överens med dokumentet ADD-1.
2014-04-22	2.1	Skrev om och korrigerade i stort sett alla kapitel.
2014-04-23	2.2	Lade in bilder och ändrade formateringen i Kapitel 2.4.8.
2014-04-24	2.3	• Skrev om och lade till vissa krav i Kapitel 3, och uppdaterade därefter Appendix C. • Uppdaterade bilden i Kapitel 2.6.9.

1 Introduction

1.1 Purpose

Syftet med detta dokument är att fastställa de krav som finns hos slutanvändaren, samt att fastställa vilka av dessa krav som måste tillfredsställas och vilka av dessa krav som skall tillfredsställas i mån av tid. Dessutom måste vi undersöka vilka krav som är tekniskt möjliga att tillfredsställa och om några krav omöjliggör andra krav.

Dokumentet ska ställa upp kraven i form av funktioner som beståndsdelar av kraven, och funktionerna skall tilldelas en av tre kategorier för att avgöra hur viktiga de är. En kategori (Core) är grundkravet på slutprodukten, d.v.s. den minsta mängd funktioner som vi behöver uppfylla för att anses ha slutfört projektet. De andra två kategorierna (Standard och Deluxe) är funktionalitet som bygger på Core-funktionaliteten för att skapa en mer komplett produkt. Funktioner tillhörande Standard och Deluxe är värdelösa om inte grundkriterierna har uppfyllts och färdigställs därför i mån av tid.

Även om vi inte har skrivit på ett formellt avtal gällande vad som skall avklaras så ser vi våra grundfunktioner som informellt bindande. Därför väljer vi att konsultera vår projektgivare innan vi bestämmer oss varför och vilka funktioner som är informellt bindande. Målet är att detta dokument tydliggör vad vi måste göra och vad vi kan göra i mån av tid. De delar av dokumentet som bör ses som bindande av projektmedlemmar samt projektgivare är framförallt de delar som gäller funktioner.

Eftersom detta är andra iterationen av detta dokument har vi möjlighet att mer definitivt avgöra vilka krav som kommer kunna tillfredsställas. Vi har dessutom genom demonstration av en alpha-version av applikationen samt en design workshop fått möjlighet till återkoppling från CIF på de implementationsbeslut vi hittills tagit, och således fått en vägledning gällande vilken riktning vi i framtiden bör välja i utvecklingen av applikationen.

1.2 Scope of the software

Produkten är en navigationsapplikation för användning inom sjukhus för vårdpersonal. Applikationen ska kunna hitta specifika människor och förmedla deras position. Applikationen ska kommunicera via en server samt databas och helst inte lagra några positioner.

1.3 Definitions, acronyms and abbreviations

AuC – Authentication Centre. Den del av servern som hanterar inloggning/utloggning i nätverket. Kommunicerar med HLR för att ta reda på om användarens lösenfras (PIN-kod) är korrekt. Utfärdar tokens.

Beacon – En enhet som sänder ut BLE-signaler.

BID – Beacon ID. Fyra bytes (två bytes minor och två bytes major) som används för att identifiera en beacon.

BLE – Bluetooth low energy. En version av Bluetooth som är designad för att dra mindre ström än vanlig Bluetooth.

Bluetooth - Kan referera till en protokollstack och en typ av hårdvaruchip som är vanlig för att på korta avstånd trådlöst överföra data mellan tekniska apparater. Den senaste standarden är Bluetooth 4.0.

Brute force-attack - Ett försök att komma åt ett lösenord genom att låta en dator systematiskt testa alla möjliga kombinationer av lösenord i en viss serie, som t.ex. alla bokstäver i alfabetet.

CIF – Clinical Innovation Fellowship. Projektets arbetsgivare. CIF är en del av CTMH (Center for Technology in Medicine and Health), som är ett samarbetsprojekt mellan Karolinska Institutet, Kungliga Tekniska högskolan och Stockholms Läns Landsting.

ForeignUID – Användarnamn för den person som blir förfrågad.

GCM – Google Cloud Messaging. En tjänst som Google tillhandahåller för att låta en tredjepartsserver skicka push-meddelanden till en Android-telefon som har registrerat sig på GCM. Google Cloud Messaging är en del av Google Play Services. När en Android-telefon registrerar sig på GCM erhåller den ett registreringsid som skickas till vår tredjepartsserver. Denna server använder detta registreringsid och en API-nyckel till GCM för att skicka push-meddelanden till klienten via GCM över POST. Ett pushmeddelande är ett litet datapaket med information, t ex begäran om att skicka beacon-id. När ett pushmeddelande kommer till GCM-servern slår Google upp rätt IP-adress med hjälp av registreringsid och skickar meddelandet till telefonen. Android-operativsystemet läser meddelandet och startar den applikation som meddelandet hör till. Att använda GCM sparar batteritid eftersom telefonen inte behöver hålla en kommunikationskanal öppen till vår server, dessutom håller GCM koll på klientens IP-adress.

HLR – Home Location Register. En databas som innehåller alla användare som har tillgång till en viss tjänst, samt vad användarna skall ha tillgång till (inte relevant i vårt fall då det inte skall finnas några användare med utökade rättigheter).

HTTP - Hypertext Transfer Protocol. Ett kommunikationsprotokoll som används för att överföra webbsidor på World Wide Web.

MSC – Mobile Switching Center. Den primära delen av en GSM-server som används för att dirigera trafik från en klient till en annan. I vårt fall är det den del av servern som är ansvarig för besvara och skicka ut positionsförfrågningar.

MSD – Manufacturer Specific Data. Serienumret för de beacons vi skall kommunicera med.

Opcode – Operation code. Koden för den operation som ska utföras.

PhoneUID – Användarnamn för den person som skickar förfrågan.

RSSI – Received signal strength indication. Beskriver hur stor effekt den mottagna radiosignalen har.

Token – Genereras av servern när en användare loggar in. Används för att autentisera användaren när förfrågningar skickas till servern.

TxPower – Den kalibrerade styrkan hos en radiosignal.

UUID - Universally Unique Identifier. Ett serienummer som används för att identifiera ägaren till en beacon. Serienumret består av 16 bytes på formatet XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX.

VLR – Visiting Location Register. En databas som innehåller alla användare som är inloggade på en viss tjänst.

1.4 References

[1] Galaxy S III is the most popular Samsung smartphone in the US, hämtad 2014-02-28, http://www.sammobile.com/2014/01/16/galaxy-s-iii-is-the-most-popular-samsung-smartphone-in-the-us/
Innehåll: Visar att Galaxy S3 är en populär mobil, så att batteritiden för en typisk mobil kan tas upp.

[2] Samsung, Galaxy S3, Mobil, Mobiltelefon, Smartphone, Galaxy - SPECIFIKATIONER, hämtad 2014-02-28, http://www.samsung.com/se/consumer/mobil/mobil/smartphones/GT-I9300RWDNEE-spec
Innehåll: Redogör batteritiden för S3 och är intressant för ämnet (begränsningar/batteri).

[3] Android Developer, Optimizing Downloads for Efficient Network Access, hämtad 2014-02-22, http://developer.android.com/training/efficient-downloads/efficient-network-access.html
Innehåll: Innehåller information om hur en typisk enhets radioantenn påverkar batteriförbrukningen, och hur man skall hantera nätverksanslutningar på ett så energieffektivt sätt som möjligt.

[4] Wi-Fi and Bluetooth: Potential sources of wireless interference, hämtad 2014-02-27, http://support.apple.com/kb/HT1365?viewlocale=en_US

Innehåll: En lista som innehåller störningspotential av olika material. Användbar eftersom många av dessa material kommer att finnas på plats på Nya Karolinska Solna.

[6] GCM Advanced Topics, hämtad 2014-02-28,
http://developer.android.com/google/gcm/adv.html
Innehåll: Teknisk och detaljerad information om hur GCM fungerar. Redogör för storleken på meddelanden som skickas.

1.5 Overview of the document

Kapitel 2.1 beskriver hur den nuvarande miljön ser ut för vår användargrupp, hur vår produkt passar in i den miljön och vilka externa system som vår produkt behöver interagera med. Kapitel 2.2 och 2.3 går igenom kraven på respektive begränsningarna för vår produkt. Kapitel 2.4 beskriver användargruppen i detalj, och utifrån detta diskuteras hur produkten bör implementeras för att tillfredsställa användarnas behov. Dessutom beskrivs en exekverbar prototyp som vi implementerat i form av en Androidapplikation. Kapitel 2.5 går igenom vilka beroenden vi jobbar mot och vilka antaganden vi gjort. Kapitel 2.6 är ett relativt tekniskt kapitel som beskriver hur produkten använder sig av externa system. Kapitel 3.1 och 3.2 listar de funktionella respektive icke-funktionella kraven på produkten.

2 General Description

2.1 Product Perspective

De tänkta användarna, d.v.s. personalen på Nya Karolinska Solna, jobbar redan idag i en miljö med många IT-system som är mer eller mindre användarvänliga. Meningen med applikationen är att den ska vara så självständig som möjligt. Anledningen till detta är att applikationen inte ska bli ytterligare en börda för sjukhusets IT-avdelning, samtidigt som det ska bli lätt att återanvända den i andra sammanhang, exempelvis på en annan arbetsplats. Dessa krav medför att det inte finns några nuvarande externa system på sjukhuset som applikationen behöver kommunicera med. Däremot har vi efter att vi fick kraven kommit fram till en designlösning som kommer att ha ett simpelt användargränssnitt som kommunicerar med en enkel databas.

Applikationen utvecklar vi för Android, och den kommer att använda sig av beacons för att kunna lokalisera personer. Vi kommer även använda oss av GCM för att skicka positionsförfrågningar till användarnas Androidenheter. Dessa system blir då de enda externa system som vi kommer vara beroende av för vår applikation. Dessa beroenden bör inte ge upphov till några problem, då alla tre har väldefinierade användargränssnitt som inte lär förändras på ett sådant sätt att det får negativa konsekvenser för projektet.

Sjukhusmiljön är stressig och tid är en stor bristvara. Man behöver i branschen ofta kommunicera med sina kollegor, och idag spenderar personalen relativt mycket tid med att bara leta efter varandra. Förutom annan personal kan det också vara svårt att hitta utrustning eftersom den ofta flyttas runt. Denna situation kommer bara att förvärras i och med det nya sjukhuset, som inte bara är betydligt mycket större utan även enbart har enkelrum för patienter. Personalen använder just nu, utöver sökare som sjukhuset tillhandahåller, sina privata mobiltelefoner för att lokalisera varandra, antingen genom att ringa den sökta personen eller genom olika chattapplikationer som exempelvis *WhatsApp.* Här ser klienten en affärsmöjlighet att ersätta dessa lösningar med ett alternativ som är specifikt designat för just detta problem. Om den är mer effektiv och användarvänlig än tidigare lösningar borde det inte vara svårt för en applikation att bli använd eftersom applikationer redan är ett accepterat och använt alternativ.

2.2 General Capabilities

De viktigaste kraven ur ett användarperspektiv är de som fokuserar på funktionaliteten och den personliga integriteten. Då applikationen är tänkt att ersätta de system som redan används på sjukhuset är det viktigt att den klarar av allt som dessa system redan gör och dessutom gör det på ett bättre och snabbare sätt.

Det mest grundläggande kravet är därför att applikationen klarar av att lokalisera personal på

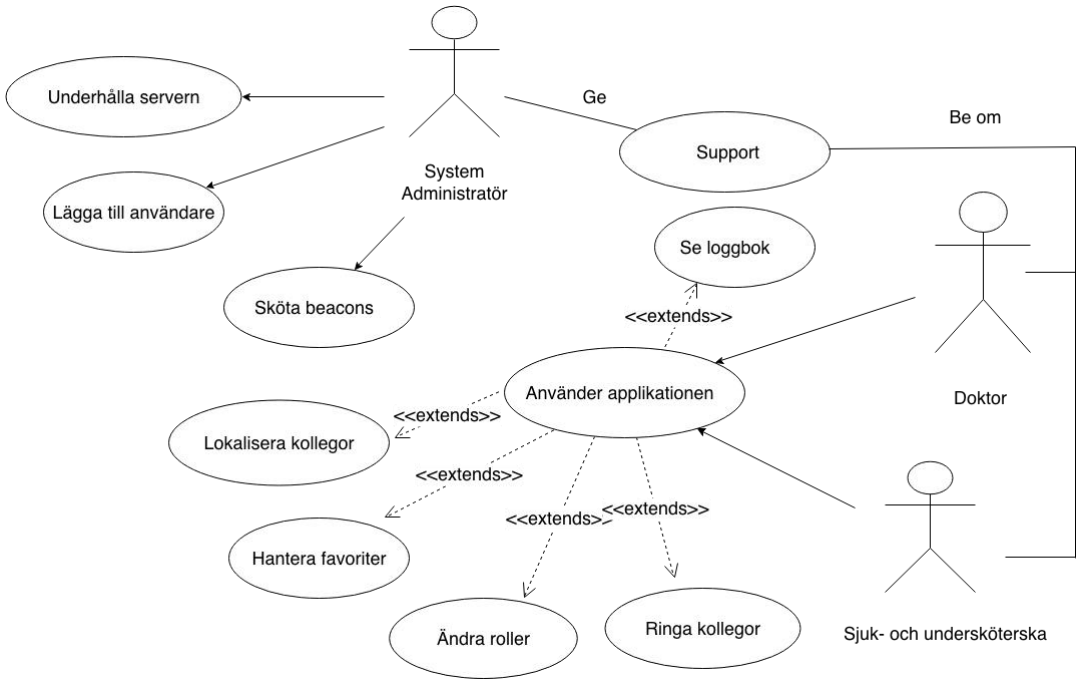
sjukhuset. Detta underlättar då för användaren som inte behöver fråga och vänta på svar från den tillfrågade, utan får istället svaret direkt från applikationen. Dock kan detta göra att användaren känner sig övervakad, och det finns därför finns en del följdkrav på applikationen för att användarna ska känna sig bekväma med att använda den.

För att uppnå detta så ska enbart den nuvarande positionen vara tillgänglig vid en förfrågan så att det inte går att få ut historik ur applikationen om var användaren har varit under dagen. Dessutom ska alla positionsförfrågningar sparas i en logg så det kan upptäckas om någon skickar ut en stor mängd positionsförfrågningar i onödan.

Det finns dessutom några funktionalitetskrav på applikationen för att användarna ska vilja använda den. Först och främst måste den vara snabb nog för att användarna inte ska tycka att applikationen är så långsam att de drar sig för att använda den och istället återgår till tidigare lösningar. Dessutom får den inte dra så mycket batteri att det blir ett problem för användarna att mobiltelefonen laddas ur under dagen, vilket redan är ett stort problem eftersom batterikapaciteten för smartphones är väldigt begränsad.

Ett annat viktigt krav är att applikationen inte stör användaren under det dagliga arbetet. Detta är speciellt viktigt på ett sjukhus där personalen redan är i en stressig miljö och absolut inte vill ha ännu en distraktion. Om användarna tycker att applikationen är störande kommer de inte heller vilja använda den. Den ska vara ett hjälpmedel som underlättar arbetet och minska tidsslöseriet.

Nedan illustreras i ett *use case diagram* hur olika användare interagerar med systemet.



2.3 General constraints

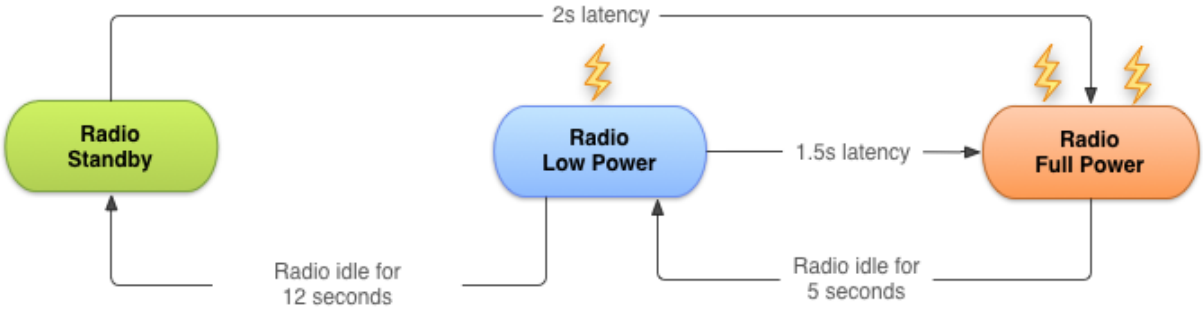
Restriktionerna på vår produkt kan sammanfattas i följande fem punkter:

- Mobiltelefonens begränsningar
- Krav på integritet
- Krav på behörighet
- Brist på tid och kunskap
- Platsens begränsningar

2.3.1 Mobiltelefonens begränsningar

2.3.1.1 Batteri

Telefonen drivs med hjälp av ett batteri som typiskt inte räcker speciellt länge. Många smarta telefoner har redan problem med batteritid, och därför kan vi behöva begränsa funktionaliteten i vissa delar av programmet för att minska på batteriförbrukningen. Exempel på en sådan begränsning är att lagra position lokalt i stället för att ladda upp den direkt på servern. En populär Androidmobil som stödjer BLE är Samsung Galaxy S3 [1], vars batteri har en kapacitet på 2100 mAh, vilket motsvarar 770 timmar standby, eller 11,6 timmar samtalstid [2].



Figuren ovan illustrerar hur radiokretsen i en typisk mobiltelefon fungerar. [3] Varje gång nätverkstrafik skickas eller tas emot går radiokretsen upp i *Radio Full Power*, det läge i vilket kretsen förbrukar som mest energi. Efter fem sekunder går radiokretsen ner i energisparläge, *Radio Low Power*. Om ingen nätverksanslutning är öppen under tolv sekunder går kretsen ned i *Radio Standby*. För att spara batteri försöker man passa på att skicka så mycket data som möjligt (en s.k. *burst*) när radiokretsen är i Radio Full Power. Att upprepade gånger skicka små paket data gör att radiokretsen aldrig går i vila, vilket gör att telefonens batteri tar slut fort. Att hitta en smart modell för att skicka BID:s så sällan som möjligt är väsentligt för att optimera batteriförbrukningen. Samtidigt är det viktigt att datat som tas emot av andra klienter inte är missvisande.

2.3.1.2 Bandbredd

Även om applikationen är tänkt att användas över en Wi-Fi-anslutning, kan det finnas många användare som använder sig av 3G/4G på jobbet. Tänkbara anledningar kan vara för att slippa slå på och logga in på Wi-Fi för att täckningen är bättre över mobilnätet, eller för att man får snabbare datauppkoppling över mobilnätet.

Många användare har begränsat med datatrafik. Detta gör att vi måste tänka på att inte skicka data mellan server och klient i onödan (vilket också har med ovanstående punkt om batteri att göra).

Restriktionen med datatrafik visar sig då större mängder data skall hämtas från servern, exempelvis när man vill ha avatarer för en lista med användare. Om vi ska använda oss av *dynamisk sökning*, krävs det att man hämtar nya data från servern vid varje nytt inmatat tecken. Detta drar också ganska mycket data, men eftersom det bara är text som skickas så är vår bedömning att detta inte kommer vara något stort problem. Den typ av dynbamiska sökning vi kommer implementera liknar Omnibox i t ex Google Chrome. Användaren skall i ett och samma sökfält kunna söka efter både för- och efternamn samt roller. Sökresultat skall dyka upp i realtid, dvs samtidigt som använder söker utan att användaren behöver trycka på en knapp Sök eller dylikt. Fördelen med dynamisk sökning är att den är mer intutiv och går snabbare att genomföra.

2.3.1.3 Teknikkrav

Endast de nyaste telefonerna på marknaden stödjer den nya antennkretsen BLE, vilket gör att användare med äldre telefoner inte kommer att kunna använda applikationen.

2.3.1.4 Integritet

Vi får inte spara vilken information som helst. Om vi gör det riskerar vi att få problem med Datainspektionen, vilket i värsta fall kan leda till att hela projektet läggs ned. Den information vi sparar måste dessutom säkras (se nästa punkt).

2.3.1.5 Behörighet

Vem som helst ska inte kunna komma åt informationen vi sparar. Denna begränsning finns till för att skydda användarnas personliga uppgifter som exempelvis namn, telefonnummer och position. En konsekvens av begränsningen är att det blir svårare för utomstående att ta del av tjänsten. Begränsningen manifesterar sig på sådant sätt att man måste ha inloggningsuppgifter för att använda tjänsten.

2.3.1.6 Tid och kunskap

Alla projektmedlemmar läser andra kurser parallellt, vilket gör att vi måste avsätta mycket tid till andra saker. Ingen av oss är professionella Androidutvecklare, något som kan påverka

kvaliteten på produkten. Bristande kunskaper kan också begränsa oss på andra sätt, exempelvis vid val av programspråk och applikationens funktionalitet

2.3.2 Platsens begränsningar

2.3.2.1 Bluetooth-störningar

Störningar av Bluetooth-signaler kan orsakas av ett antal olika faktorer som påverkar räckvidden av beacons [4]. I sjukhusmiljön som applikationen kommer att användas förmodas väggar och dörrar bestå av material som kan ha negativ påverkan. Vissa material som glas och trä leder till minimala störningar, medan betong och metall har en stor påverkan. Även vatten kan leda till störningar, och eftersom människor huvudsakligen består av vatten, skulle en radiosignal potentiellt kunna störas av en stor folkgrupps närvaro. Andra Bluetooth-sändare och Wi-Fi signaler kan också potentiellt leda till störningar eftersom Bluetooth och Wi-Fi använder sig av samma radiofrekvenser.

2.3.2.2 Anslutningsförmåga

För att lokaliseringsfunktionaliteten ska fungera måste användare kunna ansluta sig till en server som hanterar förfrågningar. För att komma åt servern krävs en internetuppkoppling, till exempel genom Wi-Fi på plats.

2.3.3 Begränsningar på data

Begränsningar på data i form av ett UML-diagram återfinnes i Appendix A.

2.4 User characteristics

2.4.1 User requirements capture

När vi hade bildat oss en grundläggande uppfattning om vad vi ville att applikationen skulle innehålla, använde vi oss av brainstorming för att komma på olika funktioner som skulle kunna implementeras för att lösa problemet. Vi höll det material vi hade fått från CIF i åtanke för att inte gå för långt från deras mål. Samtidigt försökte vi även reflektera fritt över frågeställningen från ett fräscht perspektiv för att se om vi kunde komma med några nya insikter. Detta gav oss en stor lista med alla möjliga funktioner vi ville ha med i vår applikation om vi hade haft oändligt med tid på oss.

Eftersom vi inte har oändligt med tid, insåg vi att de flesta funktioner inte skulle vara möjliga att implementera. Exempelvis var ett förslag som kom upp att implementera en trianguleringsmetod för att bättre kunna lokalisera användare. Detta hade säkerligen fått en positiv effekt på vår applikation, men tidsmässigt var det inte realistiskt att genomföra. Dessa omöjliga krav tog vi bort, och kvar blev då en uppsättning grundfunktioner som vi ville att vår applikation skulle innehålla.

En annan metod vi använde för att insamla användarbehov var intervjuer. Vår primära

användargrupp är vårdpersonal, så vi valde ut två personer ur denna grupp för intervjuer. Vår första intervju var med läkaren Anders Lindström. Anders är utbildad ortoped och har erfarenhet av att jobba inom sjukvården. Vår andra intervju var med läkarstuderande Zacharyah Abdulkarim, som studerar sitt sista år på läkarlinjen.

Under intervjuerna ställde vi frågor baserade på de krav som vi framtagit med hjälp av brainstormingen. Frågor som diskuterades var bland annat vilka användargrupper som hade störst behov av vår applikation, hur ett sjukhus fungerar på en abstrakt nivå, behov av funktioner och utformning av användargränssnittet.

Med hjälp av intervjuerna kunde vi få en verklighetsförankring istället för att bara utgå från våra egna tankar och brainstormingen. Vi fick insikt i hur en läkares dag kunde se ut och vilka funktioner som var viktiga för läkare. Ett exempel är att det är viktigt att inte bli störd i onödan, vilket i nuläget ofta händer under en vanlig arbetsdag, eftersom schemat redan är väldigt tajt. Ett annat exempel är att det är svårt att få kontakt med vissa kollegor av olika anledningar. Om andra användare då kan se att läkaren är i ett rum i vilket han inte vill bli störd, skulle det kunna minska antalet onödiga störningar.

När vi sedan fick mer konkreta krav från vår uppdragsgivare jämförde vi deras krav med de som vi hade tagit fram. Vi fick då ta bort några idéer som inte passade ihop med deras integritetskrav men även lägga till några krav som vi inte hade tänkt på. Ett exempel på en viktig funktion som vi lade till är att applikationen måste ha en låg energiförbrukning. Denna sammanfogade lista blev sedan grunden för våra krav på applikationen.

Därefter körde vi en ny brainstorming där vi fokuserade mer på hur funktionerna skulle kunna implementeras istället för vad som skulle implementeras. Då fick vi fram flera olika idéer för bland annat hur lokaliseringen av andra användare skulle kunna gå till, och för- och nackdelar med olika implementeringar. Vi tänkte hela tiden på om vi hade tillräckliga kunskaper för att implementera de föreslagna idéerna, och dessutom på hur den personliga integriteten påverkades eftersom vi med vissa idéer skulle behöva spara känslig information i en databas. Slutligen delade vi upp de idéer vi fått fram i två kategorier: Core och Deluxe. Dessa har senare utvecklats till de nuvarande kategorierna Core, Standard och Deluxe (se Kapitel 3).

2.4.2 Who will use the software?

Användaren är personer som jobbar på sjukhus och har behov av att finna kollegor. Eftersom det finns många jobbroller på ett sjukhus – bland annat vårdpersonal, administrativ personal, vaktmästare och matberedningspersonal – kan man nästan likna sjukhus vid små städer. Vi måste således begränsa våra användargrupper till vårdpersonal och administrativ personal, och antar att de andra yrkesrollerna inte kommer att använda applikationen. Vårt huvudfokus vad gäller användare är vårdpersonal, och det är utefter dessa personers behov vi måste studera användarbehovet.

2.4.3 Expected background, previous training and level of skill

De olika typerna av vårdpersonal, exempelvis läkare och sjuksköterskor, jobbar inom samma område men kan ha vitt skilda bakgrunder. Framförallt så är utbildningarna närbesläktade men inte identiska. Från undersökningar har vi fått veta att de flesta inom vården får utbildning inom vissa datorsystem, exempelvis journalhanteringsprogram. Vi kan alltså anta att de som jobbar med administrativa uppgifter har bättre teknisk kunskap än gemene personal, men en sådan teknisk kunskap ska inte behövas för att använda vår applikation. Därför antar vi att vår användare inte har några speciella tekniska kunskaper förutom kunskaperna som de har fått genom att äga en smartphone.

De systemansvariga på sjukhuset som tar hand om databasen kommer att behöva specialkompetens. Eftersom användarna måste finnas i databasen och det fortfarande skall vara enkelt att logga in så kommer de systemansvariga behöva lägga till användare i databasen och ge dem en engångskod för verifiering. Detta innebär alltså två roller som kan kallas användare samt moderator/systemansvarig.

2.4.4 Job roles and contexts of use

Läkare är en väldigt bred term som beskriver många jobb. Läkare kan exempelvis jobba på mottagningen, akuten, operationen eller kliniken. De kan även vara avdelningsläkare, gå på jour eller bakjour. Det finns alltså väldigt många roller inom yrket.

Vår applikation ska hjälpa användare att hitta en andra användare. De scenarion där detta kan vara aktuellt är exempelvis när en sjuksköterska behöver hitta jourhavande läkare eller när en läkare behöver hitta en sjuksköterska. I båda dessa fall kan det vara brådskande att hitta personen som söks, exempelvis om det handlar om att få hjälp med en skadad person. Dessutom är det viktigt att inte störa den som söks om denna person exempelvis råkar befinna sig på en operation. Därför är det ett krav att sökningen kan göras snabbt och dessutom utan att i onödan störa eller kräva aktiv respons från den sökta personen.

Ett annat tänkbart fall är att man har ett möte på sjukhuset och att en person saknas. Då kan man med hjälp av applikationen lokalisera denna person snabbt. Är personen i närheten är det förmodligen bara en kort försening och ingen vidare åtgärd krävs.

En person som går på jour eller bakjour kan behöva kontakta eller hitta vissa människor eller roller. Två scenarier är att en sådan person behöver hitta en specialist inom ett visst område, eller hitta närmaste sjuksköterska. Just inom jour och bakjour kan det bli mycket onödigt springande i lokalerna för att hitta personal, och därför är det extra viktigt att det finns en möjlighet att lokalisera personal snabbt. Det är inte bara läkarna som behöver kontakta personal, utan även annan vårdpersonal som sjuksköterskor och undersköterskor har samma behov.

För dem som jobbar på operationen är det viktigt att inte bli störd, men samtidigt ska annan personal kunna se var personerna är (om personerna tillåter det). Läkare på mottagningen jobbar relativt självständigt men ändock kan det sporadiska behovet uppstå.

2.4.5 User interface requirements

Användargränssnittet måste vara intuitivt för en helt ny användare så att endast en minimal inlärningsprocess är nödvändig. På ett sjukhus har personalen varken tid eller lust att sätta sig in i hur en applikation används. Funktionerna bör alla upplevas som relevanta och användbara.

Vi har genom undersökningar märkt att ett genomgående krav är att applikationen har överskådlig design, att den är snabb och att den är lättanvänd. Detta implementeras genom att göra applikationen så avskalad som möjligt med endast de mest nödvändiga funktionerna. Dessa ska presenteras på ett tilltalande sätt; färre knappar och menyer är bättre än ett kluster av knappar och menyer.

Ett mantra som vi har följt är att ”det ska vara snabbare att använda applikationen än att manuellt titta i intilliggande rum”. Detta är något vi tagit fasta på när vi designat applikationen. En sökruta ska hela tiden finnas tillgänglig så att snabba sökningar möjliggörs. Sökrutan kan komplettera sökordet automatiskt; om en användare exempelvis skriver ”Abc” i sökrutan så kommer en lista upp med de personer vars namn börjar på ”Abc”. Nedanför sökrutan ska det finnas favoriter och kontakter sorterade efter användarens önskan. Sorteringen kan ske i alfabetisk ordning eller utifrån sökningsfrekvens. Dock ska knappar som möjliggör snabba sökningar efter viktiga personer alltid finnas synliga, som exempelvis ”Närmaste sjuksköterska”.

Ytterligare en aspekt är integritet. Eftersom applikationen tillåter en användare att lokalisera andra användare, vilket kan vara känsligt, måste vi se till att användaren inte känner sig övervakad. Exempelvis kan applikationen visa användarens egen position, så att användaren vet exakt vad som rapporteras till andra användare.

2.4.6 Online/offline user support and project documentation

Det svåraste med vår produkt är det moment som möter användaren när denne använder applikationen för första gången. Eftersom vi kräver ett konto för att använda applikationen och detta är något som endast systemansvarig kan tillhandahålla, måste vi guida användaren första gången denne startar applikationen. Detta sker lämpligast genom en välkomstskärm som upplyser användaren om hur denne kontaktar systemansvarige. Tyvärr kan det väldigt lätt ske att användaren skippar denna skärm; många användare tycker sig inte behöva hjälp förrän det är försent. Därför kan användaren erbjudas att läsa hjälpen om denne misslyckats logga in exempelvis tre gånger i rad. Denna support kräver inte att mobilen är uppkopplad. Dock behöver applikationen vara uppkopplad för att användaren ska kunna logga in, precis som resten av applikationens tjänster kräver uppkoppling.

Produktdokumentation är något som lämpar sig bättre på en dator, bland annat eftersom en mobiltelefons skärm inte är av en storlek som lämpar sig för dokumentation. Dessutom har vi konstant strävat efter enkelhet och överskådlighet. En möjlighet att tillhandahålla support och dokumentation är i form av små pop-up-rutor i applikationen, med alternativ dokumentation på sjukhusets intranät. Dessa rutor får absolut inte vara störande utan endast innehålla kort och koncis hjälp som verkligen är användbar.

2.4.7 Detailed usability analysis

Då användarna har väldigt varierande teknisk kunskap är det viktigt att användaren kan få hjälp med applikationens olika funktioner. Tyvärr kommer vi inte ha tid för någon avancerad hjälpfunktion utan tänker implementera detta med hjälp av pop-up-rutorna som nämnts ovan. För vissa specifika problem kan vi dock implementera lite mer aktiv hjälp, som exempelvis vid inloggning. Om användaren misslyckas med att logga in ett visst antal gånger kan en ruta automatiskt visas som hänvisar användaren att kontakta IT-avdelningen för support.

Att tillhandahålla sorteringsmöjligheter för kontakter och/eller favoriter är förvisso inte särskilt svårt men kan ses som integritetskränkande eller i värsta fall orsaka problem med Datainspektionen. Vi måste därför se till att denna information bara lagrar sökningar och inga positioner. Detta måste lagras på användarsidan.

Ett komplicerat problem är att säkerställa att applikationen är snabb. Vi måste tydligt visa när förfrågningar till servern sker så inte användaren misstar en sökning för att applikationen har kraschat. Dessutom måste våra frågor till databasen vara snabba och hitta rätt användare och position. Exakt hur stort detta problem är återstår att se, men databastestningen bör visa detta innan vi egentligen börjar programmera applikationen. Om lösningen är för långsam behöver vi alltså endast göra om databasdesignen och inte själva applikationen, vilket minimerar tidsförlusten.

Något som vi har lagt ned mycket tid på är designen av databasen för att tillfredsställa integritetskravet. För ögonblicket sker konstanta förfrågningar vilket gör att ingen platsinformation loggas, men som tidigare förklarat kan detta innebära reducerad hastighet. I värsta fall kan vi behöva göra ett val mellan hastighet och integritet, men förhoppningsvis kan vi kombinera båda aspekterna.

En tanke är att använda telefonböckerna i smartphones som metafor för användargränssnittet. Skillnaden är att istället för att ringa eller skicka ett meddelande till en kontakt, får man deras position. Detta skulle göra att användaren lättare känner igen sig och inte behöver lära sig ett helt nytt användargränssnitt.

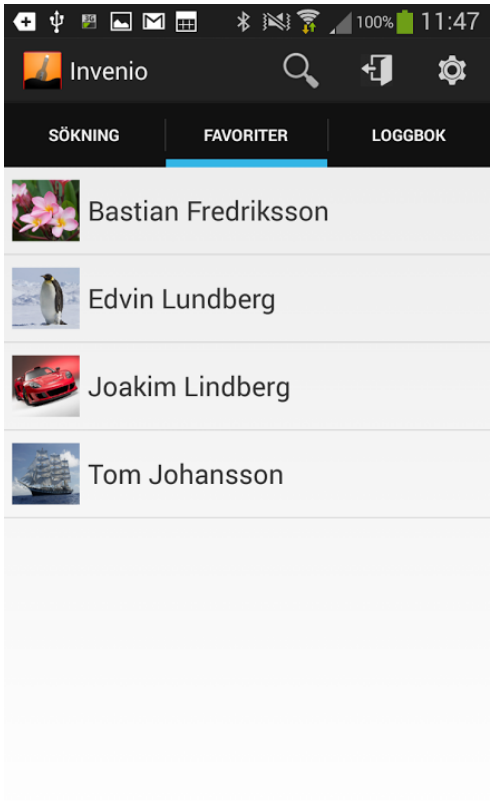
Följande användbarhetsbehov har CIF efterfrågat och ska alltså implementeras i vår applikation:

- Kunna eftersöka en annan persons position.
- Lättillgänglig sökning efter personer som ofta eftersöks.
- Söka person utifrån namn.
- Söka person utifrån roll.
- Ringfunktion för kommunikation mellan sjukhuspersonalen.
- Möjlighet att ange sina egna roller.
- Kunna få information om vem som sökt en person och när sökningen skedde.
- Inga distraktionsmoment.
- Snabb och lättnavigerad.

Utifrån dessa behov har vi skapat en exekverbar prototyp i form av en Androidapplikation, vilken nu kommer att beskrivas.

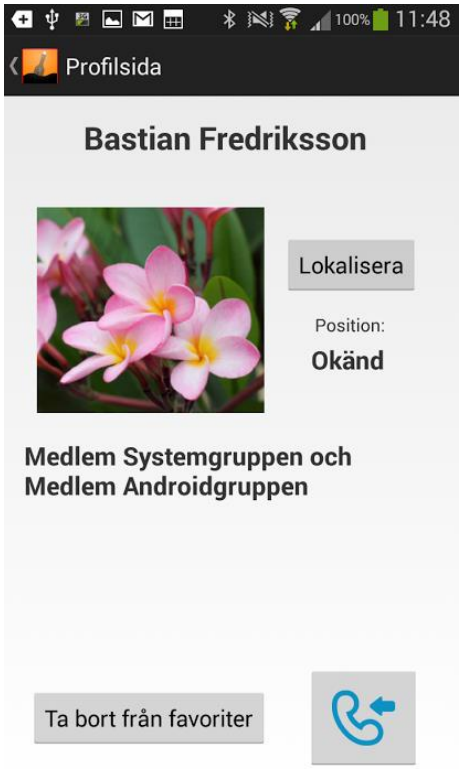
2.4.8 Executable Prototype GUI

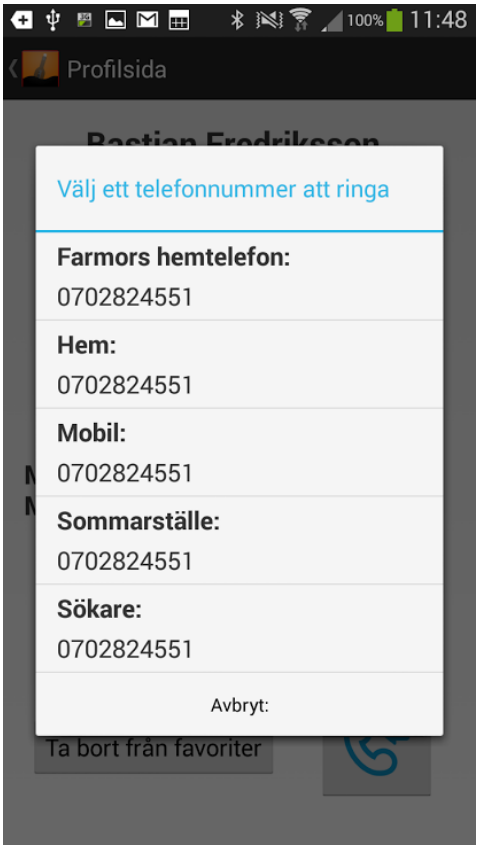
CIF:s främsta behov med applikationen är att det ska gå att söka efter en annan persons position och få positionen presenterad i applikationen. För att kunna tillfredsställa detta behov måste det gå att identifiera enskilda användare, och detta har vi löst genom att implementera inloggning. Inloggning är dock inte det enda som krävs; det måste givetvis också vara möjligt att leta upp den person som söks i applikationen.



CIF ville att det skulle vara enkelt att söka personer som en användare ville söka ofta, och därför har vi valt att implementera en favoritfunktion. Denna presenteras i form av en flik som kallas “Favoriter”, vilken illustreras i figuren till vänster. Denna favoritflik är tänkt att vara det första som syns efter att användaren loggat in, vilket gör det enkelt att snabbt hitta de personer som är ens favoriter. I favoritfliken finns en lista med den inloggade användarens favoriter. När användaren klickar på personen som eftersöks, dyker då personens profilsida upp med diverse information om denne. Det går sedan att lokalisera användaren genom att trycka på knappen “Lokalisera”.

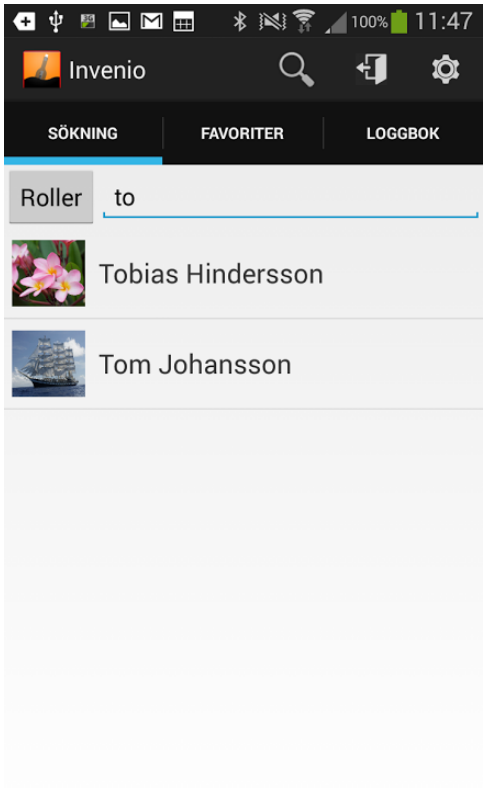
För att kunna hålla sin favoritlista uppdaterad krävs det att användaren har möjlighet till att lägga till och ta bort favoriter. Detta görs på personers profilsidor, och ett exempel på en sådan profilsida visas på bilden till höger. CIF ville att det skulle vara möjligt att se personers roller, och vi har därför valt att visa rollerna i profilsidorna.

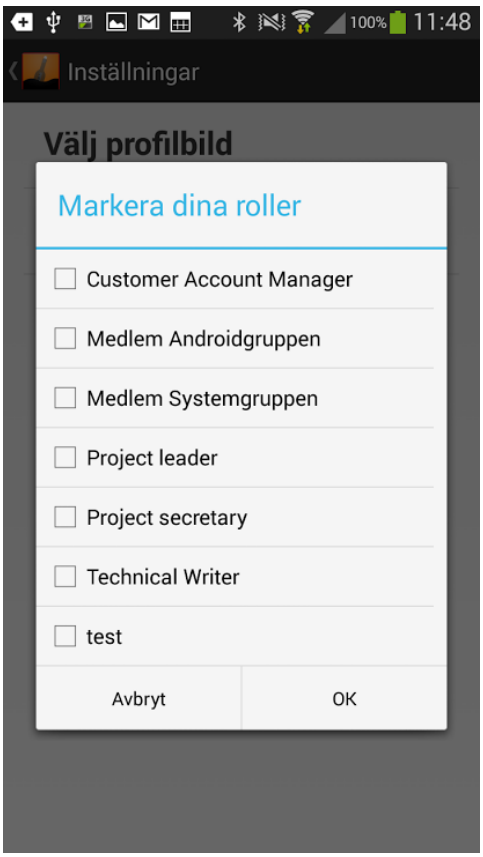




Dessutom har CIF efterfrågat möjligheten till att kunna ringa genom applikationen. På sjukhuset krävs det ofta att personalen behöver kommunicera med varandra, och en ringfunktion skulle därför underlätta deras arbete. När användaren trycker på telefonknappen i en persons profilsida, så kommer det upp en lista med alla nummer som är kopplade till personen. Dessa nummer går sedan att ringa genom att klicka på dem. Ringfunktionen är tänkt att fungera som en användarprofil i en telefonbok, där det går att klicka på exempelvis “Hem” eller “Mobil”. En illustration av detta visas i figuren till vänster.

Det är dock inte alltid samma personer som eftersöks, och det måste därför också vara möjligt att söka efter andra människor än sina favoriter. På CIF:s kravspecifikation stod det att de ville ha möjlighet att söka efter personer utifrån namn och roller. För detta tänker vi implementera en sökflik i vilken användaren kan ange namn eller roller för att sedan få upp en lista på alla matchande alternativ. I bilden till höger illustreras att en användare skriver in “to”, vilket leder till att användarna som stämmer in på sökningen dyker upp i listan. I figuren ligger sökfältet precis över resultatlistan, men vi har även funderat på att integrera sökfältet i området till höger om den orangea loggan i figuren.

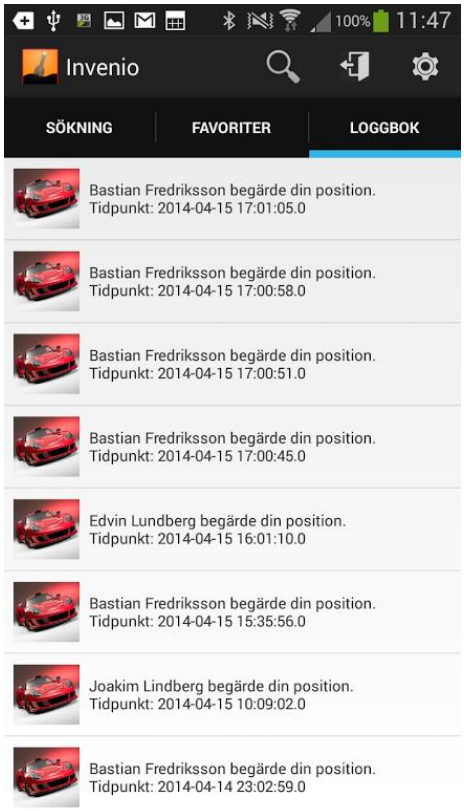




Rollerna för en person ska enligt önskan från CIF kunna sättas av personen själv direkt i en inställningsmeny i applikationen. Detta kommer vi att implementera i form av en lista som illustreras i bilden till vänster.

Dessutom kommer det vara möjligt att i sökfliken (bilden ovan) filtrera sökningen utifrån roller genom att användaren kryssar i vilka roller som ska filtreras fram. Detta kommer att implementeras ungefär som bilden till vänster. Exempelvis skulle användaren kunna kryssa i “Läkare” och “Jourläkare” för att göra en sökning på båda. Från den resulterande söklistan kan sedan personens profilsida genom att klicka på personen.

I och med sin förmåga att leta upp specifika personers position är applikationen ett väldigt kraftfullt verktyg som skulle kunna användas till att förfölja en person. Detta är givetvis inte syftet, och CIF vill därför att en användare ska kunna se vilka personer som sökt denne, och således kunna identifiera eventuella försök till förföljning. För att tillgodose CIF:s behov har vi valt att implementera en loggboksflik, vilken illustreras i bilden till höger. I fliken syns vem som har gjort en positionsförfrågan på den inloggade användaren och när förfrågan gjordes. Detta förhindrar att någon kan förfölja en användare utan dennes vetskap.



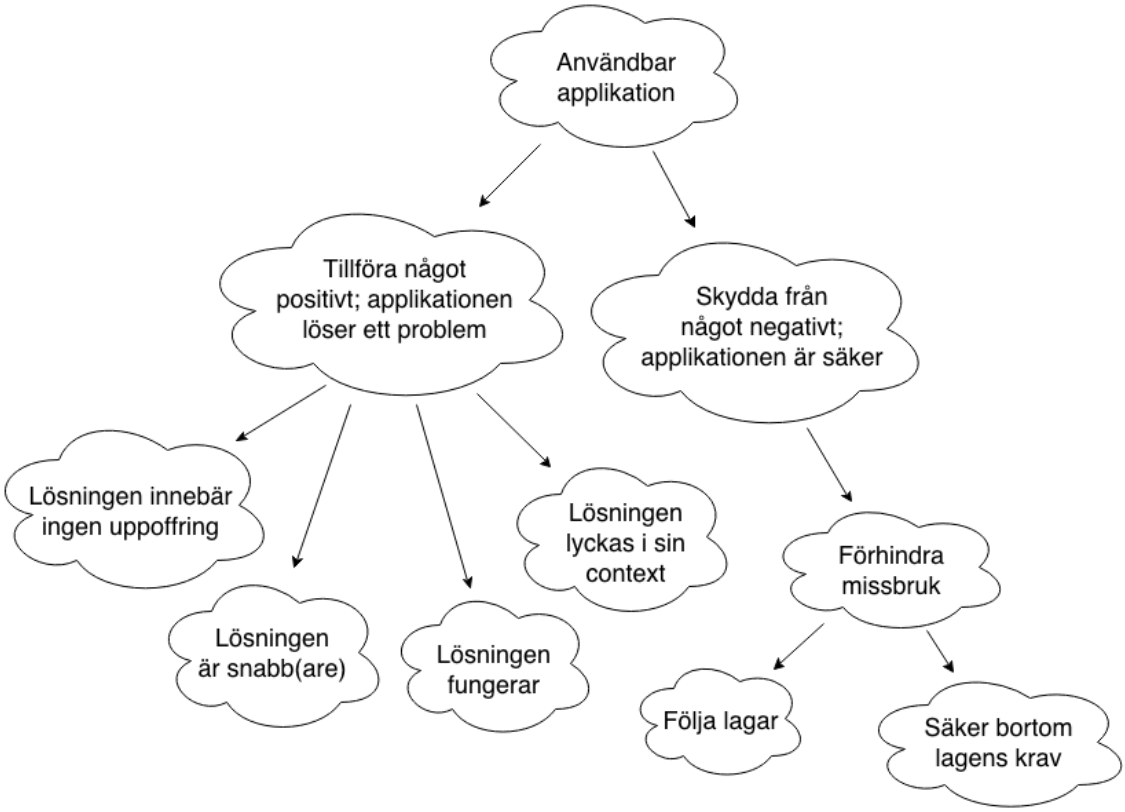
Applikationen är tänkt att vara ett användbart arbetsverktyg, och får därför enligt CIF inte vara ett distraktionsmoment. Inga notiser kommer därför att dyka upp när en användare blir sökt, och som samtliga bilder i detta kapitel illustrerar finns ingen överflödig och distraherande information. Applikationen är enkel och avskalad, precis som CIF ville ha den.

I och med att syftet med hela applikationen är att den ska effektivisera sjukhuspersonalens arbetstid, är det av yttersta vikt att den är snabb och lättnavigerad. CIF:s behov med applikationen är att det ska vara snabbare att söka efter en person med applikationen än att gå till grannrummet och leta efter denne. Därför tillåter applikationen användaren att nå samtliga vyer med maximalt tre klick från huvudvyn.

2.5 Assumptions and dependencies

En rimlig utgångspunkt för att diskutera såväl beroenden som antaganden är att börja med de behov och krav som produkten förväntas leva upp till för att vara konkurrenskraftig. När dessa etablerats kan vi utifrån vår planering avgöra vilka faktorer dessa kommer att bero på. Eftersom vår projektmodell i skrivande stund är aktuell och framtagits inte minst med målet att den ska vara genomförbar, och verka för att projektet realiseras på ett så bra sätt som möjligt, följer därav att dessa faktorer också bygger på aktuella antaganden.

Till att börja med utgår vi från att vår produkt kommer att vara användbar på något vis och utvecklar sedan denna tanke i riktning mot abstrakta beskrivningar av vad vår produkt måste kunna göra, alltså det som framgången hos vår produkt beror på. Figuren nedan beskriver de viktigaste funktionerna och beroendena som vår produkt måste tillgodose för att vara framgångsrik.



Vi går nu in på dessa abstrakta beroenden för att mer konkret redogöra för vad dessa innebär och vilka förutsättningar dessa vilar på.

2.5.1 Applikationen ska tillföra något positivt

2.5.1.1 Lösningen innebär ingen uppoffring

En fungerande applikation tillför något men ej på alltför stor bekostnad av något annat. Vår applikation ska fungera på de avsedda användarnas mobiltelefoner och ej i alltför hög grad försämra den tidigare upplevelsen av dessa. Konkret innebär detta två beroenden:

- Beroende: Applikationen får ej konsumera för mycket batterikapacitet.
    - Antagande: Konstant sökning efter beacons via Bluetooth kommer ej att konsumera nämnvärt mycket av batterikapacitetet. Skulle detta visa sig vara ett felaktigt antagande kan vi finna alternativa lösningar, som exempelvis att endast låta en klient söka efter beacons då en positionsförfrågan av denne har gjorts.

    - Antagande: Genom att endast ansluta till servern, och därmed aktivera radiochippet, vid begäran om position (förutom på användarens egen begäran), ska applikationen förbruka såpass lite batterikapacitet att batteriets livslängd efter en laddning inte ska försämras nämnvärt vid normal användning.
- Beroende: Applikationen får ej konsumera för mycket bandbredd.
    - Antagande: De funktioner som applikationsservern kommer att begära av klienten och vice versa kommer att kräva relativt lite datatrafik, tack vare ett enkelt kommunikationsprotokoll och en kärnfunktionalitet som inte fordrar storskalig dataöverföring.

2.5.1.2 Lösningen är snabb (snabbare än befintlig metod för att lösa samma problem)

En kritisk punkt för vår applikation är att dess främsta funktion inte möjliggör något som tidigare varit omöjligt, utan snarare ska effektivisera något som redan görs. Detta medför konkurrens och därmed en rad beroenden som går att relatera till såväl det tekniska som det interaktiva området. Dessa beroenden är givetvis förekommande inom applikationens alla användningsområden även om de är som mest kritiska då man använder applikationens huvudfunktion, nämligen lokalisering av användare.

- Beroende: Applikationen måste ur teknisk synvinkel prestera snabbt.
    - Antagande: Applikationen kommer att kunna lagra närmaste beacon redan innan en förfrågan görs varför den är redo att skicka denna information med mycket kort fördröjning. Detta antagande beror även av antaganden gjorda i föregående punkter.
    - Antagande: GCM presterar snabbt.
    - Antagande: Kommunikationen vid en positionsförfrågan kommer, bortsett från den trafik som går via GCM, innebära överföring av endast små mängder data varför denna kommer att ha förutsättning att vara snabb.
- Beroende: Applikationen måste prestera snabbt i interaktionen med användaren.
    - Antagande: Vi kan göra gränssnitt för Android som är smidiga och intuitiva.
    - Antagande: Vi kan kategorisera data som användaren kan behöva söka bland på ett sådant sätt att datan aldrig blir ohanterlig för användaren.
    - Antagande: Information ska i applikationen hämtas och lagras på sådant sätt att väntetider uteblir i så hög grad det går.

2.5.1.3 Lösningen fungerar

En grundförutsättning för applikationens framgång är att den fungerar. Detta kommer givetvis att bero på en mängd faktorer, men några av dem som är mest kritiska berör det faktum att vi har med relativt ny teknik att göra.

- Beroende: Den nya teknik med beacons som applikationen använder måste fungera i kritiska aspekter.

- Antagande: Beacons går att placera ut på ett sådant vis att den beacon som telefonen uppfattar som närmast faktiskt också är den beacon som i verkligheten förknippas med den plats där användaren befinner sig. Detta förutsätter bland annat att beacons i vissa fall går att ställa ner i signalstyrka samt att beacons, i mer eller mindre utsträckning, begränsas signalmässigt av vissa typer av väggar.
- Antagande: Ett småskaligt system kommer att fungera lika väl vid skalning. Detta bygger dels på att våra resurser räcker till, dels på att det inte finns något hinder för detta i den redan existerande Bluetooth-lösning som vi kommer att använda.
- Antagande: GCM är i stort sett alltid stabilt och i drift.

2.5.1.4 Lösningen lyckas i sin kontext

Vår produkt ska passa in i en arbetsvardag där folk har beprövade metoder och vanor som de använt under många år. Vid undersökningen som föregick initiativtagandet till detta projekt framkom att våra målanvändare lägger ned mycket tid på att leta efter kollegor på jobbet, vilket vi hoppas kunna lösa med resultatet av detta projekt. Men finns det faktorer som inte förutsetts? Har vårdpersonalen behov av att gå runt och leta eftersom detta innebär välbehövda pauser från arbetsuppgifterna? Innebär letandet att folk från olika avdelningar snabbt bekantar sig med varandra och att sökandet därmed fyller en högre social funktion? Är läkar- och sjuksköterskekåren konservativa vad gäller förändringar och föredrar att göra som de alltid gjort? Resultatet av applikationen kan tyckas bero helt och hållet på föregående frågor men det finns trots allt en renodlat social aspekt av det hela som enbart till viss del beror av tidigare nämnda frågor. För att vår produkt ska lyckas fullt ut måste vi ha våra slutanvändare med oss och användarundersökningarna får inte ha missat avgörande faktorer som i slutändan kan påverka och göra att personalen fortsätter att använda de beprövade metoder de är vana vid.

- Beroende: Det finns en vilja att använda vår applikation samt ett socialt utrymme och förutsättningar för att introducera applikationen på arbetsplatser.
  - Antagande: Användarundersökningarna har tagit hänsyn till att ett sjukhus är en dynamisk och intensiv arbetsplats med speciella arbetsförhållanden och inte missat någon viktig faktor som kan tala för att problemet som vi ämnar lösa kan existera av anledningar som fyller en viktig funktion.
  - Antagande: Arbetsplatser där denna applikation introduceras har ett socialt klimat där man välkomnar förändringar och är villig att pröva nymodigheter.
  - Antagande: Projektgivarens uppskattning att det är genomförbart att skapa en produkt som kan konkurera ut vanor som grundlagts och existerat i åratal är riktig.

2.5.2 Applikationen ska inte tillföra något negativt

2.5.2.1 Förhindra missbruk genom att följa lagar

Vi är i Sverige måna om att värna den personliga integriteten vilket är en fråga som kommit att bli mer och mer aktuell i takt med vår tekniska utveckling. Det är lätt att tänka sig att en applikation som kan ta reda på var dess användare befinner sig ligger i riskzonen för att ses som integritetskränkande, varför vi i projektet måste se till att hålla oss inom lagens ramar.

- Beroende: Applikationen får ej bryta mot den personliga integriteten och de lagar som skyddar denna.
    - Antagande: Eftersom vi endast begär information om en användares position i stunden och inte lagrar denna begår vi inget fel. Informationen kan således inte komma i orätta händer genom intrång i databas eller dylikt då sådan information inte lagras.
    - Antagande: Ett register över användare är godkänt att lagra i en databas om dessa kan ses som medlemmar i någon form av organisation. De ramar inom vilka denna applikation är tänkta att användas kan ses som en organisation.
    - Antagande: Vi kan skydda de användaruppgifter och dylikt som finns i databasen på erforderligt vis.
    - Antagande: Vi kan logga positionsförfrågningar på sådant vis att man vid behov kan kontrollera vem som sökt vem vid vilken tid.

2.5.2.2 Förhindra missbruk genom att tillhandahålla säkerhet bortom vad lagen kräver

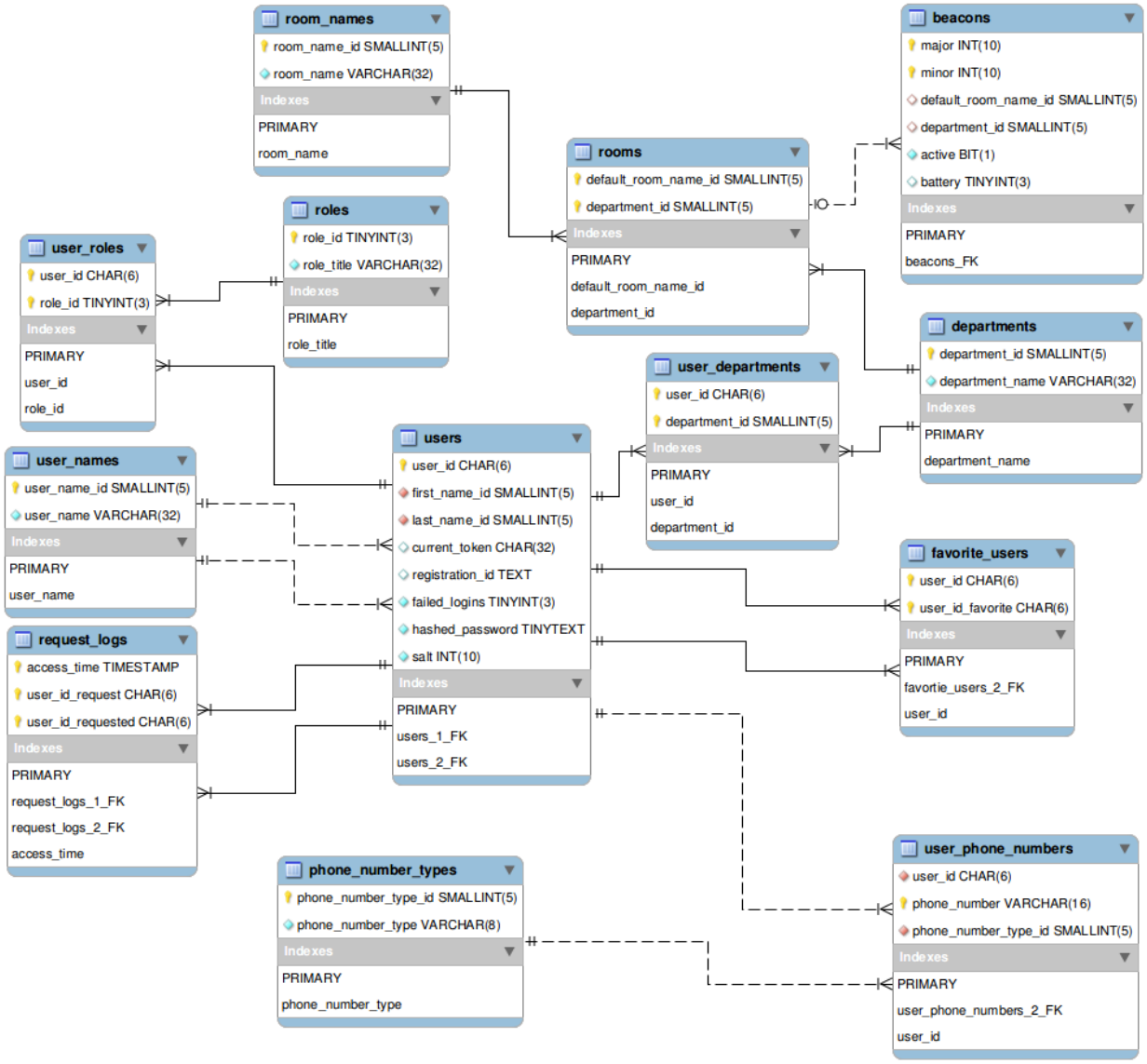
Förutom att skydda medlemsregister bör också en tjänst innebära någon form av säkerhet för användaren. Användaren bör kunna vara säker på att den information som erhålls från servern är tillförlitlig. Det bör alltså inte vara möjligt att förfalska någon annans information och därigenom ge sken av att denne befinner sig någonstans där den inte är.

- Beroende: Applikationen måste skydda information som skickas för att garantera tillförlitlighet och sekretess.
    - Antagande: Lösenord förvaras datasäkerhetsmässigt tillfredställande i databasen.
    - Antagande: En skyddsmekanism med autentiseringstoken går att implementera för att fastställa legitimiteten hos uppladdad data.

2.6 Operational environment

2.6.1 Arkitektur

Den tänkta lösningen innefattar fyra huvudkomponenter: beacons, handhållna Androidenheter, GCM och ett centralt nav i form av en server med tillhörande databas. Arkitekturen illustreras övergripande i Appendix B. Ett ER-diagram för databasen illustreras nedan.



2.6.2 Positionering

För positionering kommer beacons placeras ut på fasta platser, exempelvis i rum, som registreras i databasen. De handhållna enheterna kommer lagra vilken beacon de befinner sig närmast för tillfället genom att mäta signalstyrkan och lagra BID tillhörande den beacon med starkast signalstyrka. BID kopplas i databasen till en fysisk plats, exempelvis ett rumsnummer.

2.6.3 Registrering

Alla användare kommer att registreras i databasen. Informationen som registreras i databasen är användarnamn, lösenord och eventuellt kontaktuppgifter. Registrering är ett krav för att användaren ska kunna logga in och använda applikationen. Till en början är det tänkt att en systemadministratör kommer att skapa användarkonton till användarna. En alternativ lösning

skulle vara att låta användarna registrera sig själva, antingen direkt i telefonen via applikationen eller genom ett separat registreringsformulär via exempelvis en hemsida.

2.6.4 Inloggning

Vid inloggning ska phoneUID och lösenord matas in av användaren i applikationen. Inloggningsuppgifterna kommer att skickas tillsammans med opcode och deviceID till servern för verifikation. Vid korrekt angivna uppgifter genererar servern en unik token som skickas ut och detta sparas av applikationen. Endast en token är giltig åt gången för varje användare, och vid varje inloggning genereras en ny.

2.6.5 Token

Den unikt genererade token som skickats från servern till klienten efter lyckad inloggning kommer att sparas av klienten. Efter inloggningen kommer klienten att skicka med token i samtliga meddelanden och förfrågningar till servern. Detta syftar till att identifiera användaren utan att behöva skicka användarens inloggningsuppgifter upprepade gånger. Det finns två huvudsyften till detta förfarande, nämligen att

- hindra användaren från att vara inloggad på flera enheter samtidigt och att
- minska risken för att illvilliga tillskansar sig användarens inloggningsuppgifter via en brute force-attack.

Skickar applikationen en ogiltig eller gammal token kommer servern att svara med en begäran om att applikationen ska logga ut och att användaren tvingas logga in på nytt för att tilldelas en ny token. Tanken är även att användarens konto ska blockeras efter ett flertal misslyckade inloggningsförsök för att ytterligare öka säkerheten.

2.6.6 Integritet

Användarnas integritet är av yttersta vikt och ska genomsyra hela systemets uppbyggnad. För att förhindra att användarna missbrukar systemet ska samtliga positionsförfrågningar loggas i servern. Anledningen till detta är att det ska vara möjligt att i efterhand ta reda på om någon utnyttjat systemet på ett otillbörligt sätt. Användarna ska själva ha möjligheten att granska när och av vem deras position blivit begärd. Det ska däremot inte vara möjligt att logga användarnas positioner och därför kommer inte användarnas positioner att lagras i databasen. Dessa kommer istället att skickas direkt från den förfrågade användarens telefon via servern till den användare som efterfrågat positionen. Användarnas telefoner kommer endast att lagra senast kända BID och dessa kommer endast att kopplas till en position genom översättning via databasen som finns lagrad på servern.

2.6.7 Google Cloud Messaging (GCM)

För att genomföra push-förfrågningar till användarnas mobiltelefoner kommer vi använda oss av GCM. Anledningen varför vi valt att använda GCM är att minska applikationens batteriförbrukning och därmed minimera risken för att användarnas batterier laddar ur.

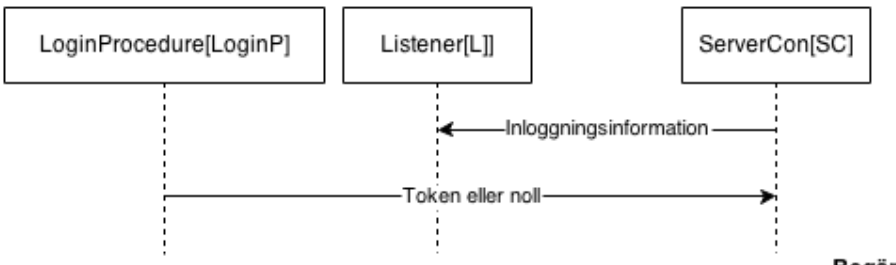
För att tillgodose användarupplevelsen är det viktigt att användaren omgående får svar vid en positionsförfrågan. För att uppnå detta kommer vi använda oss av HTTP-förfrågningar som gör det möjligt att skicka meddelande innehållande information på upp till 4 kB. [6] Detta för att i meddelandet kunna skicka begäran innehållande opcode och token. Svar från klienten kommer att skickas direkt till servern utan att gå via GCM för att ytterligare minska svarstiden, eftersom användning av GCM medför en viss fördröjning.

2.6.8 Positionsförfrågan

När en användare genomför en positionsförfrågan skickas opcode, phoneUID, foreignUID och token till servern. Servern kontrollerar att den skickade token stämmer överens med den som servern senast genererat. Vid felaktig token svarar servern med begäran om att applikationen ska logga ut. Vid korrekt token loggar servern positionsförfrågan och skickar deviceID, opcode till GCM som pushar opcode till den berörda enheten. Den mottagande klientapplikationen skickar BID, tillhörande den beacon som den senast registrerat starkast signalstyrka ifrån, direkt till servern utan att gå via GCM. Servern söker med hjälp av BID fram aktuellt rumsnummer i databasen och skickar det till klienten vars användare genomfört positionsförfrågan.

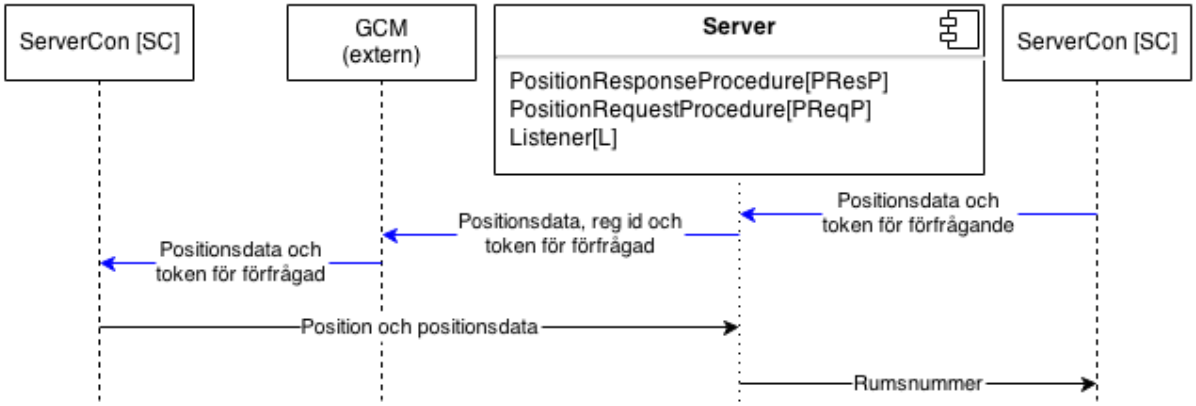
2.6.9 Sekvensdiagram

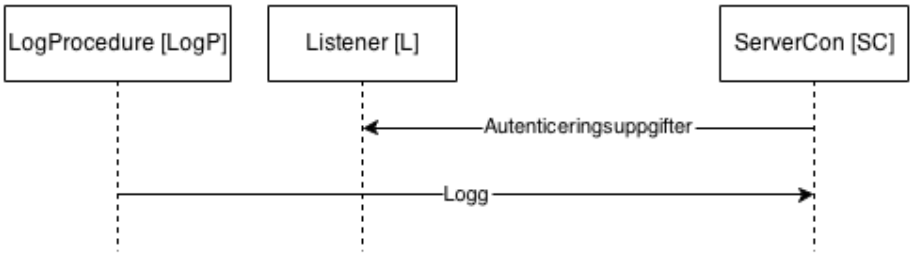
Nedan illustreras ett sekvensdiagram som beskriver olika typer av interaktion mellan servern och klienten. ServerCon är en Javaklass som finns i klienten, och alla andra Javaklasser i figuren finns i servern. GCM är en extern komponent och finns hos Googles servrar.



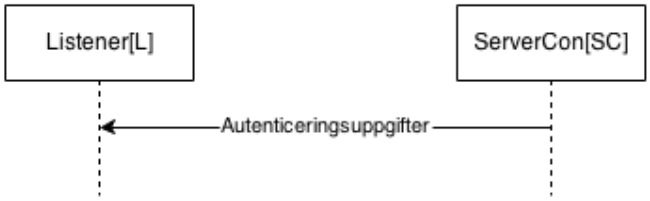
Inloggning

Begäran av position (blå)
och svar på denna begäran (svart)





Begäran av logg



Utloggning

3 Specific Requirements

I detta kapitel finns tabulerade krav för applikationen. I Appendix C återfinnes ett diagram som beskriver hur de olika kraven är relaterade till varandra.

Fältet *Need* i tabellerna har följande tolkningar:

- **Core** – Minimala funktioner som krävs för att kunna demonstrera produktidén.
- **Standard** – Funktioner som förväntas av produkten för verkliga situationer.
- **Deluxe** – Experimentella funktioner som inte behövs men som kan förbättra produkten.

Fältet *Priority* i tabellerna har följande tolkningar:

- **Hög** – Grundstenar i systemet och måste således implementeras tidigt.
- **Medel** – Dessa funktioner behöver implementeras efter funktionerna med hög prioritet.
- **Låg** – Dessa funktioner behöver implementeras efter funktionerna med medelprioritet.

Fältet *Stability* i tabellerna har följande tolkningar:

- **Stabil** – Funktioner och/eller krav som kommer att stå kvar.
- **Mellan** – Funktioner och/eller krav som inte är skrivna i sten, men kvarstår sannolikt.
- **Instabil** – Funktioner och/eller krav som kan komma att tas bort eller ändras.

3.1 Capability requirements

3.1.1 Hitta närmaste beacon

Identifier	Hitta närmaste beacon.
Requirement Description	Applikationen ska kunna identifiera den beacon som är närmast den handhållna enheten. Detta baseras på txPower och RSSI.
Justification	Utan detta krav är det en omöjlighet för någon annan användare att söka efter personen, och således skulle hela applikationen vara värdelös.
Need	Core.
Priority	Hög.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Användartester ska göras där beacons kommer sättas upp i olika rum. Två beacons sätts upp i olika rum, en användare (X) går mellan rummen med sin enhet och en annan användare (Y) efterfrågar positionen på X under tiden. Rätt position för X ska visas för Y. Om detta lyckas 9 av 10 gånger med intervall på 20 sekunder mellan testerna, byte av rum mellan två tester minst 5 gånger, så har funktionaliteten verifierats.

3.1.2 Lokalisering på rumsnivå

Identifier	Lokalisering på rumsnivå.
Requirement Description	Kravet hänger ihop med *Hitta närmaste beacon*. Det är dock viktigt att precisionen för detta ska vara på rumsnivå.
Justification	Det krävs att precisionen är på rumsnivå för att kunna hitta personal på sjukhuset.
Need	Core.
Priority	Hög.
Stability	Mellan.
Source	Kravet kommer från CIF.
Verifiability	Testning kommer ske på liknande sätt som för *Hitta närmaste beacon*. Beacons kommer sättas upp i olika rum. Målet är sedan att när användaren går in i ett rum med en beacon, ska en annan användare kunna lokalisera denne på rumsnivå med hjälp av applikationen.

3.1.3 GUI (delvis textbaserat)

Identifier	GUI (delvis textbaserat)
Requirement Description	Kravet innebär att applikationen baserar sig på ett enkelt GUI, där information om andra användares positioner fås som text.
Justification	Utan GUI blir det givetvis svårt att kunna få någon som helst information, och det är något som krävs för alla applikationer.
Need	Core.
Priority	Hög.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Användartester kommer att göras för att se till att GUI:t är funktionellt och gör vad det ska utifrån aktioner. Om användaren exempelvis efterfrågar en annan användares position, så ska positionen synas i användarens enhet i form av text på den sökta användarens profilsida.

3.1.4 Söka efter användare med namn

Identifier	Söka efter användare med namn.
Requirement Description	För att kunna få andra användares position, måste det gå att välja vilken person som ska efterfrågas.
Justification	Kravet behövs för att kunna utföra applikationens mål, d.v.s. att kunna söka efter andra personer och få deras position.
Need	Core.
Priority	Medel.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Användartester ska utföras där en användare (X) loggar in på applikationen. En annan användare (Y) ska också logga in på applikationen, söka efter namnet på X och få X som träff på sökningen.

3.1.5 Broadcastförfrågan

Identifier	Broadcastförfrågan.
Requirement Description	Då en användare söker en annan användare ska sökaren få den söktas position just nu. Positionen hämtas endast från den söktas telefon genom servern.
Justification	Kravet gör det möjligt att undvika lagring av användares senaste positioner i databasen. Informationen utkrävs istället från användarens telefon då den behövs.
Need	Standard.
Priority	Medel.
Stability	Stabil.
Source	Kravet har tagits fram av CIF tillsammans med projektgruppen.
Verifiability	Samma som *Lokalisering på rumsnivå*.

3.1.6 Logga förfrågningar

Identifier	Logga förfrågningar.
Requirement Description	Alla förfrågningar som görs om andras positioner ska loggas och sparas i databasen. Det som ska sparas är den som sökte, personen som söktes och tidpunkten för sökningen.
Justification	Detta är ett sätt att kunna se till att inte någon missbrukar systemet genom att övervaka andra användare. Det är alltså lite av ett integritetsskydd.
Need	Core.
Priority	Medel.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Användartester ska göras där beacons sätts upp och användare söker efter andra användare. Informationen om dessa sökningar ska sedan finnas lagrad i databasen. Detta verifieras på samma sätt som *Få information om användaren blivit lokaliserad*.

3.1.7 Få information om användaren blivit lokaliserad

Identifier	Få information om användaren blivit lokaliserad.
Requirement Description	En sökt användare ska kunna se att denne blivit sökt genom att hämta en förfrågningslogg från databasen, där det står när och av vem användaren har blivit sökt.
Justification	Det ska vara möjligt att se om någon användare använder systemet i form av att övervaka andra användare.
Need	Deluxe.
Priority	Låg.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Användartester ska göras där beacons sätts upp och användare söker efter andra användare. Den sökta användaren ska då kunna ta del av informationen om dessa sökningar genom applikationen. Det ska finnas information om vem sökningen gjorts av och vid vilken tidpunkt den skedde. Användare X söker användare Y och Y ska då kunna se att logginlägget tillkommit med rätt uppgifter.

3.1.8 Automatisk uppdatering vid ny påträffad beacon

Identifier	Automatisk uppdatering vid ny påträffad beacon.
Requirement Description	Applikationen ska automatiskt uppdatera vilken beacon som är närmast då enheten flyttas och befinner sig närmare en annan beacon. Detta baseras på txPower och RSSI.
Justification	Kravet behövs för att kunna lokalisera andra användare. Applikationen behöver ha kännedom om närmaste beacon för att den ska kunna kopplas till ett rum vid lokalisering.
Need	Core.
Priority	Hög.
Stability	Mellan.
Source	Kravet kommer från CIF.
Verifiability	Samma som *Hitta närmaste beacon*.

3.1.9 Söka efter roll

Identifier	Söka efter roll.
Requirement Description	Det ska gå att söka specifika roller som exempelvis läkare.
Justification	Detta kan vara användbart om det behövs en specifik typ av personal till ett rum, utan något krav på vem det ska vara. Dessutom har personalen olika roller vid olika tidpunkter, som exempelvis jourläkare.
Need	Deluxe.
Priority	Låg.
Stability	Medel.
Source	Kravet kommer från CIF.
Verifiability	Genom att ge användare X rollen “läkare” och sedan söka efter en läkare från en annan användares enhet, så ska resultatet bli de användare som är läkare, och det inkluderar alltså användare X.

3.1.10 Välja roller

Identifier	Välja roller.
Requirement Description	Det ska gå för användaren att själv välja de roller som användaren innehar, exempelvis läkare eller kirurg.
Justification	Gör det enklare för personal att kunna söka efter roller, om rollerna stämmer överens med de faktiska rollerna personen innehar.
Need	Standard.
Priority	Medel.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Detta testas genom att en inloggad användare kryssar i sina roller, exempelvis läkare. Sedan söker en annan användare efter läkare och denne ska då finnas i listan. Det kan också valideras genom att besöka användarens profilsida. Det ska då stå att personen har rollen läkare. Sedan kryssas läkare ut, och samma procedur sker igen. Då ska varken personen hittas vid sökning efter läkare, eller läkare finnas som roll för användaren.

3.1.11 Dynamisk filtrering

Identifier	Dynamisk filtrering.
Requirement Description	När en användare använder sig av sökfunktionen ska sökalternativen filtreras i realtid.
Justification	Funktionen är användbar och smidig eftersom det är enkelt att hitta det som söks utan att alltid behöva skriva in hela namnet.
Need	Deluxe.
Priority	Låg.
Stability	Stabil.
Source	Kravet kommer från projektgruppen.
Verifiability	Genom att försöka söka efter personer ska det tydligt kunna ses att applikationen filtrerar korrekt. Exempelvis om det i sökrutan står “Abc” ska personer som heter “Abcde” i förnamn kunna dyka upp men sådana som heter något som inte börjar på “Abc” ska inte finnas med.

3.1.12 Visuellt, scrolla

Identifier	Visuellt, scrolla.
Requirement Description	För att kunna välja person i den lista som fås efter sökning, ska det gå att scrolla igenom listan. Innan sökningen gjorts är listan tom.
Justification	Gör det enklare och smidigare att hitta personen som söks.
Need	Deluxe.
Priority	Låg.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Det ska testas om det fungerar att scrolla i listan med sökresultat.

3.1.13 Logga in

Identifier	Logga in.
Requirement Description	För att kunna avgöra vem som använder applikationen, och vem de personliga inställningarna ska ställas in efter, kan användaren logga in på sitt användarkonto. Det finns också en “kom ihåg mig”-kryssruta som tillåter användaren att slippa skriva in uppgifter vid varje inloggning.
Justification	Krävs för att kunna avgöra vem som är inloggad i systemet, och är en grundsten för att applikationen ska fungera. Det handlar dels om personliga inställningar, dels att det exempelvis bara ska gå att söka på de användare som faktiskt är inloggade.
Need	Core.
Priority	Mellan.
Stability	Stabil.
Source	Kravet kommer från CIF i kombination med projektgruppen.
Verifiability	Funktionen testas genom att med korrekt angivet användarnamn och lösenord lyckas logga in. Dessutom ska felaktiga uppgifter testas, så att det inte ska kunna gå att logga in med inkorrekt information.

3.1.14 Ringfunktion

Identifier	Ringfunktion
Requirement Description	Det ska gå att ringa alla inloggade användare genom applikationen.
Justification	Kommunikation på avdelningarna behövs, och således ska det vara enkelt att kunna ringa de personer man vill.
Need	Deluxe.
Priority	Låg.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Ringfunktionen testas genom att gå in på en inloggad användares profilsida och försöka ringa denne. Förutsatt att rätt telefonnummer finns inmatat i databasen för denna användare ska den sökta användaren få ett inkommande samtal.

3.1.15 Favoriter

Identifier	Favoriter.
Requirement Description	En användare ska kunna ha en lista med favoriter, det vill säga personer som användaren ofta eftersöker.
Justification	Kravet finns för att underlätta sökningar som görs ofta. Om en användare alltid söker en och samma person är det omständligt och onödigt att varje gång söka upp personen med namn.
Need	Standard.
Priority	Mellan.
Stability	Stabil.
Source	Kravet har tagits fram av CIF tillsammans med projektgruppen.
Verifiability	Det ska gå att från favoritlistan nå en specifik användares profilsida genom att klicka på användaren.

3.1.16 Välja favoriter

Identifier	Välja favoriter.
Requirement Description	Det ska vara möjligt att lägga till och ta bort en favorit ur sin favoritlista.
Justification	Användaren vet själv vilka personer användaren vill ha lättillgängliga, och därför bör det vara möjligt för användaren att modifiera favoritlistan.
Need	Standard.
Priority	Mellan.
Stability	Stabil.
Source	Kravet har tagits fram av CIF tillsammans med projektgruppen.
Verifiability	Till en början ska inga favoriter finnas i en användares favoritflik. Därefter söker användaren upp en inloggad användare och lägger till användaren som favorit. Då ska den nytillagda användaren finnas i favoritfliken. Därefter tar användaren bort denna favorit, och då ska favoritfliken återigen vara tom.

3.1.17 Profilbild

Identifier	Profilbild
Requirement Description	En användare ska kunna ha en profilbild.
Justification	Det är enklare och tydligare att hitta den person man söker om det utöver namnet på personen även visas en profilbild.
Need	Deluxe.
Priority	Låg.
Stability	Mellan.
Source	Kravet har tagits fram av CIF tillsammans med projektgruppen.
Verifiability	Kravet ska testas genom att en användare X tilldelas en profilbild vid skapandet av användarkontot. Därefter ska en annan användare leta upp X:s profilsida och verifiera att korrekt profilbild visas.

3.2 Constraint requirements

3.2.1 Endast lagring av närmaste beacon

Identifier	Endast lagring av närmaste beacon.
Requirement Description	För att kunna söka efter användares position krävs endast den senaste beacon som varit närmast användaren.
Justification	Eftersom integritet är en viktig fråga för CIF, ska systemet inte spara användarnas positioner. Närmaste beacon ska finnas lokalt på användarnas enheter och kunna kopplas till ett specifikt rum genom servern.
Need	Core.
Priority	Hög.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Användartester ska göras där beacons sätts upp och en användare befinner sig i ett rum. Det ska då gå att se att information angående närmaste beacon finns i enheten. Avståndsbedömningen får baseras på txPower och RSSI.

3.2.2 Ingen lagring av positioner i databasen

Identifier	Ingen lagring av positioner i databasen.
Requirement Description	Ingen lagring av användarnas position görs i databasen.
Justification	För att bevara den personliga integriteten skall ingen positionsinformation lagras i databasen.
Need	Core.
Priority	Hög.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Varje inlägg i databasen dokumenteras med loggutskrifter. När beacon-information skickas till servern ska denna inte loggas i någon tabell i databasen. Ingen information relaterad till något rum ska heller loggas. Varken beacon-informationen eller rumsinformationen ska alltså dyka upp i loggutskrifterna då en positionsförfrågan görs.

3.2.3 Applikationens språk

Identifier	Applikationens språk.
Requirement Description	Applikationen ska vara på svenska, vilket innebär att all text är på svenska.
Justification	Det måste givetvis finnas ett språk för informationen som visas i applikationen.
Need	Standard.
Priority	Hög.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Användartester görs genom att använda sig av applikationen och se att språket är korrekt.

3.2.4 Kodkommentarer på engelska

Identifier	Kodkommentarer på engelska.
Requirement Description	All kod som skrivs ska kommenteras på engelska.
Justification	Kravet behövs eftersom koden kan komma att läsas av programmerare från olika länder.
Need	Standard.
Priority	Hög.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Läs koden och se att den är kommenterad på rätt språk.

3.2.5 Ej vara för batterikrävande

Identifier	Ej vara för batterikrävande.
Requirement Description	Applikationen ska ej vara för batterikrävande.
Justification	Detta behövs för att inte riskera att applikationen laddar ur användarnas batteri innan deras arbetsdag är över.
Need	Deluxe.
Priority	Låg.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Testet kommer att utföras på en Nexus 5. Från att ha varit fulladdad ska applikationen startas på enheten. Den ska vara igång i 9 timmar. Någon annans position ska sökas 4 gånger i timmen. Den inloggade användaren ska bli sökt 8 gånger i timmen. Loggen kommer användas för att visa att detta har skett. Information om hur länge applikationen varit igång och hur länge enheten varit utan laddning går att se i genom Android. Kravet uppfylls om mobilen har batteri kvar och fortfarande är på.

3.2.6 Ej vara för minneskrävande

Identifier	Ej vara för minneskrävande.
Requirement Description	Applikationen får inte vara för minneskrävande för användarnas telefoner.
Justification	Applikationen ska inte vara för minneskrävande så att den tynger ner telefonen.
Need	Standard.
Priority	Medel.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Beroende på vilken virtuell maskin som används får applikationen kräva olika mängder minne. Vid normal körning ska applikationen kräva högst 50 MB minne om Dalvik VM används och högst 100 MB om ART Runtime används. Denna information går att se genom den i Android inbyggda informationen om minnesanvändning.

3.2.7 Snabb

Identifier	Snabb.
Requirement Description	Det ska gå snabbare att använda applikationen för lokalisering än att gå till grannrummet och leta.
Justification	Applikationen ska vara ett verktyg som effektiviserar personalens arbete, och då måste den vara snabb.
Need	Standard.
Priority	Låg.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Från det att användaren öppnar applikationen förutsatt att denne redan är inloggad, ska det vara möjligt att lokalisera en inloggad användare på mindre än 30 sekunder.

3.2.8 Inga distraktionsmoment

Identifier	Inga distraktionsmoment.
Requirement Description	Applikationen ska vara stilren. Den ska inte vara distraherande eller stressande.
Justification	Applikationen ska vara ett verktyg som effektiviserar personalens arbete, och då får den inte vara ett distraktionsmoment.
Need	Standard.
Priority	Medel.
Stability	Stabil.
Source	Kravet kommer från CIF.
Verifiability	Applikationens standardinställningar ska från början vara: • Inga vibrationer • Inga ljud Applikationen ska inte ha några onödiga animationer. Exempelvis räknas en animation som visar att applikationen laddar information eller arbetar *inte* som onödig, medan en animerad figur som studsar runt utan något syfte definitivt räknas som onödig. Övergångsanimationer mellan fönster är okej. Testning går ut på att använda applikationen och se att ovan gäller.

3.2.9 BLE-stöd

Identifier	BLE-stöd.
Requirement Description	Enheten måste ha stöd för BLE; annars tillåts inte en användare vara inloggad.
Justification	BLE behövs för att enheterna ska kunna kommunicera med beacons.
Need	Core.
Priority	Hög.
Stability	Stabil.
Source	Kravet kommer från att beacons kräver BLE-kommunikation.
Verifiability	Se till att enheten som används vid testning har stöd för BLE.

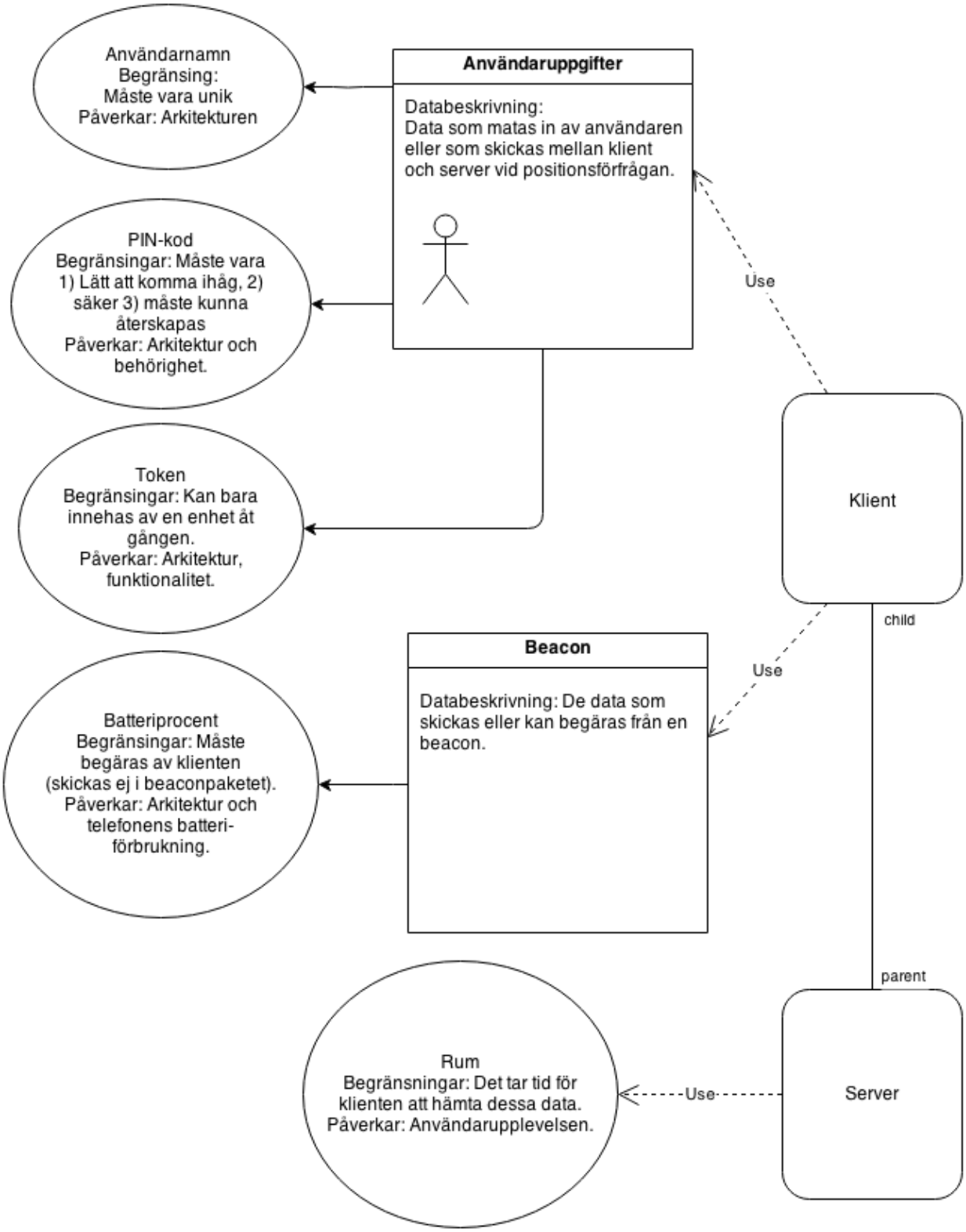
3.2.10 Android 4.3

Identifier	Android 4.3.
Requirement Description	För enheten krävs Android version 4.3 (SDK version 18) eller senare.
Justification	Externa system (BLE) kräver Android 4.3 eller senare för att fungera.
Need	Core.
Priority	Hög.
Stability	Stabil.
Source	Kravet kommer från externa system (BLE) vilka kräver Android 4.3 eller senare.
Verifiability	Se till att enheten som används vid testning kör Android 4.3 eller senare.

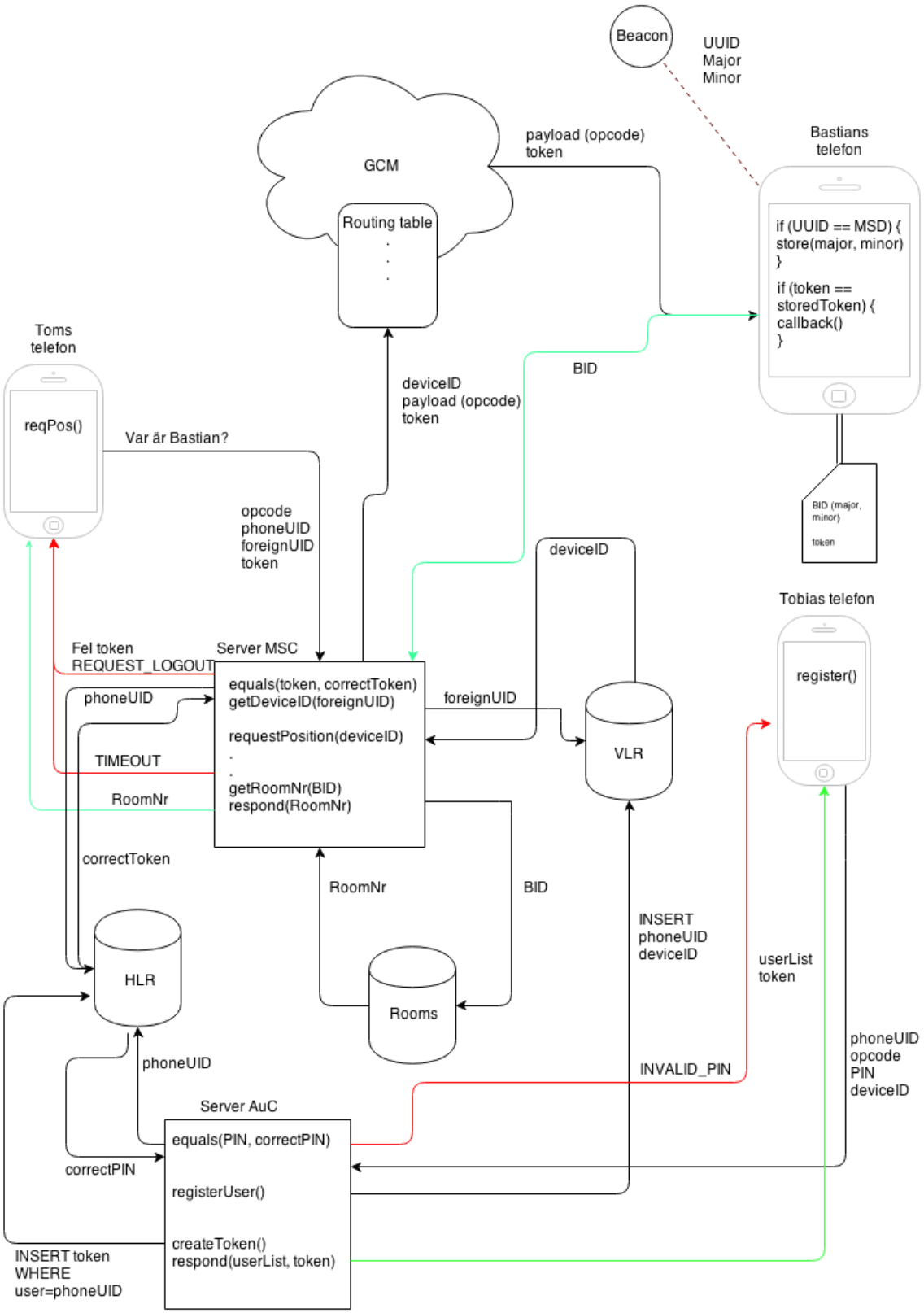
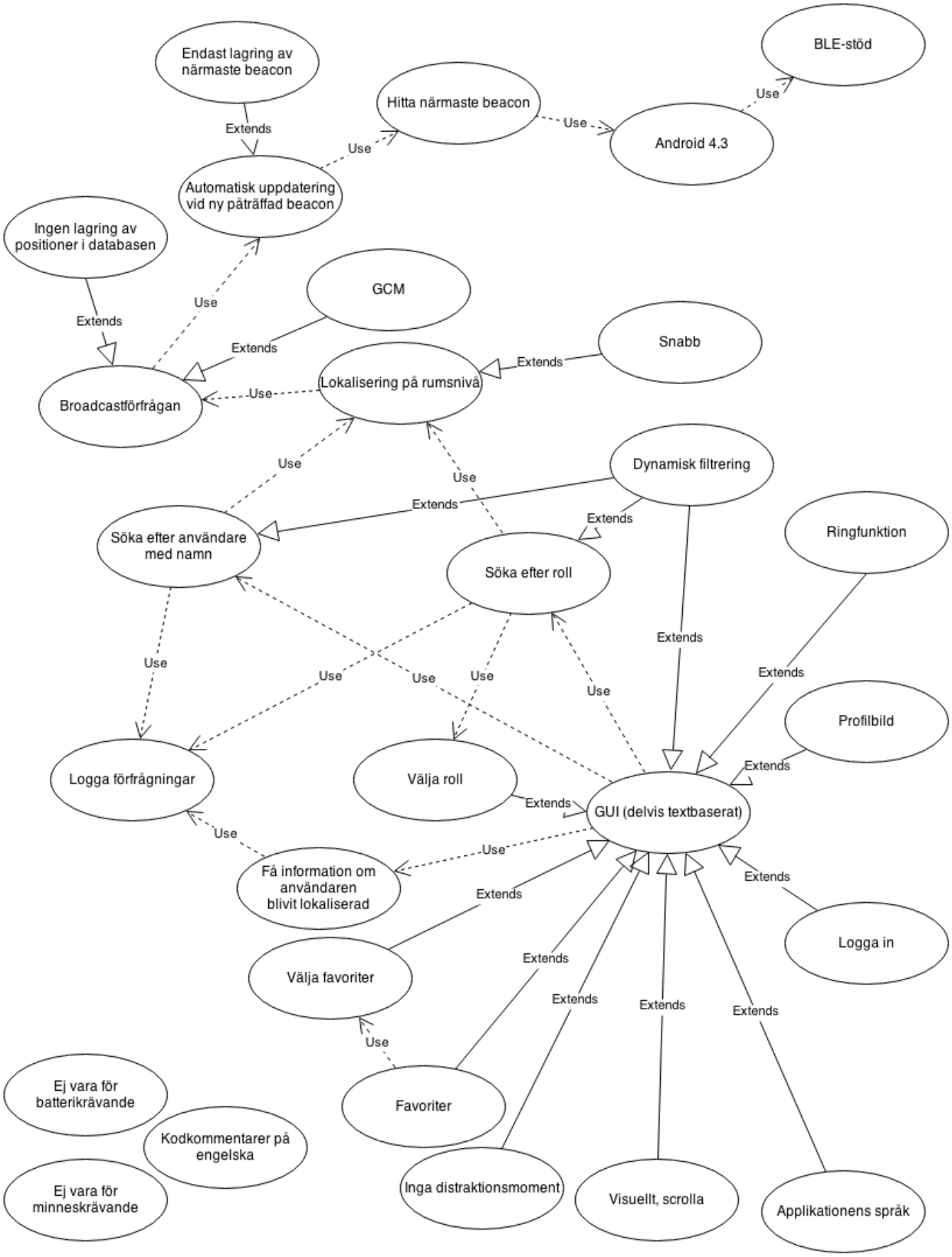
3.2.11 GCM

Identifier	GCM.
Requirement Description	GCM (Google Cloud Messaging) är en tjänst som applikationen kommer att använda.
Justification	För att kunna göra en positionsförfrågan behöver man nå den söktas telefon. Detta görs genom att skicka en push-notifikation via GCM för att väcka den söktas telefon ur viloläge så att dennes telefon kan vidta avsedd åtgärd.
Need	Standard.
Priority	Medel.
Stability	Stabil.
Source	Kravet kommer från projektgruppen.
Verifiability	Kravet testas genom att försöka söka en användares position. Om en position återfås har kravet verifierats, eftersom det då har visat sig möjligt att väcka den söktas telefon och återfå en position.

Appendix A – Data Constraints Diagram



Appendix B – Architecture Diagram



Appendix C – Requirements Diagram