NTNU

Master Thesis

Exploring Practical Implementation of Touchless Access Control Using iBeacons in Norwegian Hospitals

Sebastian Brage Hansen

Student 684950

Oslo, 1st June 2016

IMT4904 Master Thesis Project

30 ECTS

Master of Interaction Design

Department of Computer Science and Media Technology

NTNU

Department of Computer Science and Media Technology
NTNU
Box 191
N-2802 Gjøvik
NORWAY

A. Abstract

One of the leading causes of infections and cross contaminations in hospitals is surface contact. Much is done in form of routine and design to limit pathogen transmission through surfaces, however laps in routine and new technology make room for improvement. By removing some of the key touchpoints hospital employees interact with, one might be able to decrease cross contamination. This thesis explore the practical implementation of touchless access control in a hospital environment using beacon technology. The basis for the research is an ethnographic study adapted for HCI resulting in three use cases. A prototype consisting of three Estimote Bluetooth low energy beacons/transmitters; one Android smartphone receiver; a cloud based API and one Raspberry Pi 3 mini computer is used to triangulate the smartphones position in a virtual grid to assess if a door should be opened by the Raspberry. Measurements show the selected beacons and their Software Developer Kit (SDK) to vary too much in the radio frequency/RSSI signal strength to accurately report distance making triangulation of position inaccurate. Although inaccurate, the signal strength is shown to be consistent and significantly different within 5 sets of 100 measurements at different distances, enabling further projects to improve on the design by making a classification model. The thesis concludes that although RSSI signal strength and the provided SDK alone is not sufficiently accurate for positioning the suggested classification model should be able to give an accurate positioning in 2D space. However regardless how the beacons are used the design must follow universal design principles like affordance, boundaries and cuing and respond within 500ms to be perceived as instantaneous and functional for an end user.

B. Acknowledgements

Before listing all the people, know that if you helped me in any way, big or small, I am grateful. This thesis is a work of many ideas and many contributions, from inspirational ones to factual. They all count. They all make a unity. With this in mind, and a wish for all to be able to see and contribute to my work, I wrote this thesis using only open source and free to use tool. Tools that afford collaboration and asynchronous work, over different time zones and borders. It was also important for me to have the ability to work from anywhere, at any given time. For me Google Doc and Google Sheets made this possible. It has been interesting writing an entire Master Thesis using these tools: On the phone while on the bus. In a park with my laptop. On the plane with an iPad. I would recommend it. Work where you feel creative. Work free.

I would like to thank Tone Nordbø, for sending me on the path to becoming an Interaction designer; Tone Ruud Engen, for believing in me and pushing me; Miriam Eileen Nes Begnum, for helping me collect my thoughts and refocus my efforts; Alexander Brage Hansen and Aud Helen Singstad, for helping select my topic and being a constant source of inspiration and joy; Anders Løvlie, for pushing for academic excellence; Frode Volden, for organizing the master program during my last two semesters; Simon McCallum, for offering insightful help and philosophical perspectives; my dad, Ola Brage Hansen, for being a strong man and beacon in a chaotic period – you have helped me through a lot more than you know; my mom, Kjersti-Lill Lund, for helping me push my academic boundaries and just being a mom; all my classmates, for sharing my experience and venting with me in times of despair; Rikke Julie Foss-Pedersen, Eirk Ghile and Lisa Canneaux, for excellent colloquium sessions in Oslo and carpooling when visiting Gjøvik; Jonathan Bourne, for being a wiz at statistical analysis; Dan Audne, for being a ninja programmer, teaching me the fundamentals of coding for Android; Sashidharan Komandur, for supervising my thesis; and last but not least, Inge Thorud and Hildegunn Amanda Soldal at Dagbladet, for giving me the chance to complete this education alongside work, proving my worth and allowing me to spread my knowledge to Aller Media.

Thank you all.
Next round is on me!

C. Figures

D .Tables

E. Index

1.0 Introduction

Every year 6% of patients committed to Norwegian hospitals are contaminated by disease not related to their initial affliction (Aftenposten 2012). For Denmark the figures are about 10% (Burcharth et al. 2014) and models created for estimating the general average show it to be 5-10% (D'Agata et al. 2012).

Hand sanitation is by healthcare professionals, researchers and specialists alike believed to be the single most important contributor to reduce spread of pathogens (D'Agata et al. 2012. Personal communication, Senior Consultant Bjørn Iversen, November 26th 2015. Personal communication, Specialist Bio Engineer at Diakonhjemmet Center for Psychopharmacology, November 6th 2015). Still, research show that even trained professionals forgo personal hand hygiene, forgetting to wash their hands after visiting bathrooms (Burcharth et al. 2014). Much has been done to improve sanitation and prevent spread via surface contacts, but there is still much to desire from the design of things healthcare workers interact with on daily basis.

Arguing that the fewer items healthcare workers touch when performing their everyday job, the less indirect spread of pathogens via surface area will occur, this thesis explores the possibility for deploying a touchless interface for doors in restricted areas using low frequency Bluetooth technology. Building on previous work on automated door control using iBeacons (Andersson 2014), an affordable, easy to use, "intelligent" access control system is prototyped and tested in an environment replicating those found at Diakonhjemmet Center for Psychopharmacology, a laboratory where pathogen spread via surface contact is omnipresent. Although the beacon technology is found not to be suitable for the suggested prototype at its current state, many of the constraints and design challenges with similar systems are discovered and discussed, generating further knowledge for designing similar solutions.

The current extent of indirect infections caused by pathogens, the technology and basic principle for how the prototype work is covered in chapter 2; methodology for gathering information, creating prototypes and testing is described in chapter 3; results are detailed in chapter 4; interpretation of results and thoughts on what they might mean going forwards is described in chapter 5, 6 and 8; and the thesis' limitations are covered in chapter 7.

1.1 Problem Statement

The leading cause for spread of pathogens within hospitals is surface contact. Much has been done in forms of sanitation routines and touchless interfaces, like automation for sanitation dispenser and doors, but there is still room for improvement. During an unstructured interview (appendix 8.1) Senior Consultant at Folkehelsa, Bjørn Iversen, expressed it simply: “Our hands and bodies are what makes us contagious. We carry pathogens with us on our hands from patients or deceased to someone who is healthy [...] we touch our keychains, pull on strings [levers to open mechanical doors (editor’s note)], punch in our key codes leaving pathogens behind [...] designing hospitals to minimise areas for surface contact is what we need.” (personal communication, November 26th 2015). Access control is one of the areas where medical staff are in constant contact with items like keys, key cards or keychains when locking or unlocking doors. Through the use of modern, affordable technology these touch points can be re-envisioned and adapted for a touchless interface.

When Douglas Adams pictured the door of the future in his pop culture hit *Hitchhikers Guide To the Galaxy*, he portrayed a self aware door with a personality: It knew if it should open based on who was coming, it had emotion and it absolutely loved its job: “As the door closed behind them it became apparent that it did indeed have a satisfied sigh-like quality to it. ‘Hummmmmmmyummmmmmm ah!’ it said...” Picture the same for a hospital: A door that knows if it should open for whomever is coming without use of key cards or pin codes, understands a doctor is running towards it and opens faster than normal, or that closes before fully opening because the doctor forgot something and turned around before running though. Such a door would work wonders in high contamination area, where the risk of germs spreading from sector to sector via surface contact is omnipresent.

How can iBeacons help create the touchless, invisible, interfaces needed to help reduce this indirect spread of pathogens? One way is by removing the need for surface contact, pin code pads and card swipes altogether. Beacons using the iBeacon protocol can be acknowledged/discovered by smartphone apps. When a beacon is discovered, the app can perform predefined actions (Gast 2014). Pairing this with a cloud based service that registers inputs and in turn triggers actions for other devices, a proximity based, automated system can be created: A 21st century invisible UI for opening doors, powered by The Internet of Things (figure 1.1).

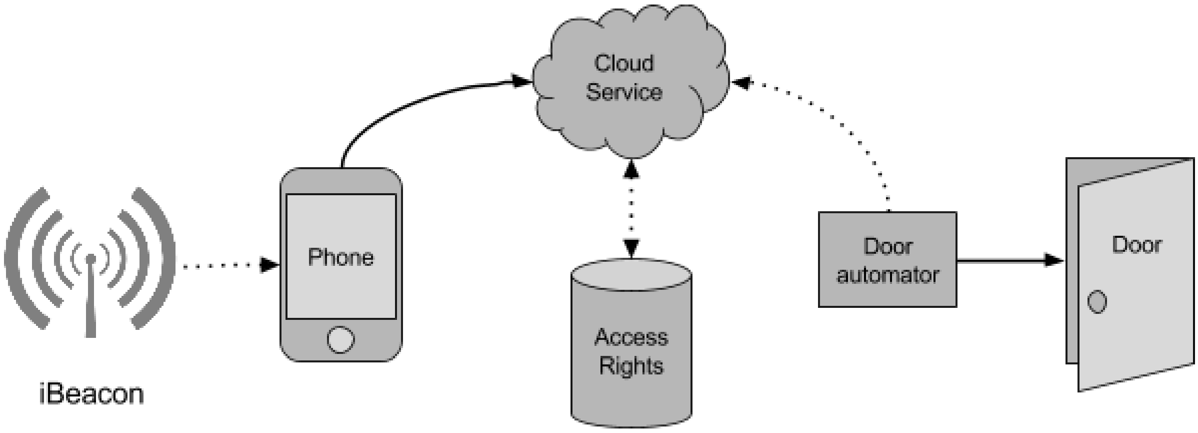


Figure 1.1: The phone discovers a beacon and sends a request for the door to be opened. The cloud service validates the request and generates a command for the door to be opened. The door automator continually listens to the cloud service for a command for what to do with its door and acts accordingly

Although a seemingly straightforward design, there exists a challenge beyond the technical: What happens when a door truly becomes autonomous? How will people know how to interact with it? In an episode of the popular animated series *The Simpsons,* one of the main characters, *Bart*, sells his soul to his best friend to prove a point. At first everything seems normal. Then when he tries to enter the local convenient store through an automated door, things have changed: The door will not open. Bart becomes frustrated and irritated when to discover that the door only opens for people with souls. This cartoon – although a joke – illustrates a central topic in design: How can users know how to interact with something, if they do not know the internal system or its constraints? Designer Don Norman highlights this when talking about door design: “Affordances provide strong clues to the operations of things. [...] When simple things need pictures, labels, or instructions, the design has failed.” (Norman 2002, p. 9).

Building on Tim Andersson (2014) work for the University of Linköping, where iBeacons were found reliable for automated door locks, this thesis explores some of the change in user behavior and usability that might be encountered. Small issues, like knowing what side of the door a user is on to allow for exits but not entry, was discovered to be challenging in Andersson’s project and highlighted from a technical standpoint. It illustrates the need for more understanding of the user's interactions. What happens in practise when removing the affordability a key code input console offers? How will users know if they are in restricted zones and need permission to enter new zones?

1.2 Research Questions

This project explores interaction obstacles between user and technology, when using a touchless access control system based on Bluetooth low energy beacons. To give the project a relevant frame, a secure biotechnical lab is chosen for basis and observations. The thesis consists of three main parts: 1. Establishing knowledge of common usability issues with access control in a hospital environment. 2. prototyping a solution that affords the movement through access controlled doors without need for surface contact. 3. discussing what positive effects removing surface contact has to reducing spread of pathogens. To align these goals, three main research questions is the focus of the thesis:

RQ1. How are access controls encountered, used and manipulated in hospitals?
RQ2. What are the typical obstacles for an invisible UI for access control?
RQ3. What benefits and limitations do iBeacons offer over pre-existing, conventional access controls in hospital environments?

To further investigate the ability to collect data for increasing knowledge in the field of medical research concerning pathogen distribution modeling, a fourth research question will to some extent be examined.

RQ4. What are the practical uses of 2D tracking of healthcare personnel working in high contagious areas?

1.3 Keywords

Healthcare, Pathogen Reduction, Bluetooth Low Energy (BLE), iBeacon (Beacon), Radio Frequency Identification (RFID), Near Field Communication (NFC), Received Signal Strength Indicator (RSSI), Intelligent Automatic Doors, Access Control, Security, Internet of Things (IoT), Automatic Identification and Data Capture (AIDC).

1.4 Abbreviations and Terms

3G – Third generation wireless network communication protocol (HSPA).

4G – Fourth generation wireless network communication protocol (LTE).

ANOVA – Analysis of variance. Statistical comparison of groups to determine difference.

API – Application Programming Interface. A way for a system to interact with a web service.

App – Application, typically used for small programs running on portable devices.

Beacon – Bluetooth Low Energy device discoverable by devices like smartphones.

BLE – Bluetooth Low Energy. Bluetooth device that consumes less battery.

Cloud – Cloud service, cloud software, etc. Internet based computing or software.

EDGE – Enhanced Data rates for GSM Evolution. Standard for wireless communication.

iBeacon – Apple Inc’s protocol for exchange data wirelessly between beacon and device.

GPIO – General Purpose Input Output. Units for sending/receiving from a microcontroller.

HCI – Human Computer Interaction. How we uses and manipulates systems and machines.

HSPA – Evolved High Speed Packet Access, high-speed wireless communication standard.

IoT – Internet of Things. An interconnection of devices through wired or wireless technology.

LTE – Long-Term Evolution, high-speed wireless communication standard.

MySQL – Database engine and query language for storing information.

Protocol – Set of rules governing how a system communicates.

Pathogen – A biological infectious agent that causes disease or illness in its host

PHP – Server-side scripting language designed for web development.

REST – API standard using verbs as input to controller action and output.

Raspberry Pi – Small pocket sized computer capable of manipulating electronic devices.

RSSI – Received Signal Strength Indicator.

SDK – Software Development Kit. Bundling of features and code-developing software.

Smartphone – A mobile phone with advanced operating system enabling the use of apps.

UUID – Universally Unique Identifier. Randomised ID with low possibility for duplication.

WIFI – A wireless network for exchanging information and accessing the internet.

1.5 Expected Findings & Planned Contributions

The project replicates and re-imagines a system for a non-touch, invisible UI access control in theory and partly in practice. Prototypes are created to understand how iBeacons might be used to facilitate an invisible UI and take advantage of the Internet of Things to grant access for the right users at the right time (figure 1.1).

The limitations and possibilities of using iBeacon for security measures is expected to help reimagine how hospitals facilitate security while preventing the spread of pathogens. Quantitative or qualitative measurements of actual decrease in the spread of pathogens when using the prototypes will not be part of the project. Security issues and ethical questions that arises when using the system – concerning amongst others the systems architecture or the privacy matters issues with tracking individual users habits in an “offline environment” as beacons allow for (Datatilsynet 2015) – are noted and discussed, but not analyzed or investigated.

Data gathered is available for public, academic and corporate use under the DBAD 1.0 license granting unlimited use of code, data and patterns except where otherwise stated.

- **DBAD 1.0 -** http://www.dbad-license.org/

2.0 Theory, Background & Existing Literature

2.1 Spread of Pathogens, Sanitation & Precautions

Every year 6% of patients committed to Norwegian hospitals are contaminated by disease not related to their initial affliction (Aftenposten 2012). A study performed by Odense University Hospital found the similar trend in Denmark where 10% of their patients were subject to post commitment infections (Burcharth et al. 2014), a randomized ethnographic study found the numbers for USA to be 4% (Magill et al. 2014) and the exposure model established in a general study on multi resistant pathogens correlates with these, estimating the affected amount to be 5-10% (D'Agata et al. 2012). Much of the contamination was attributed to pathogens spreading from medical personnel or patients through surface contact due to improper cleanliness attributed to haphazard, improper training and personal neglect (Burcharth et al. 2014, Yokoe et al. 2014).

Multiple studies have been done to categorise, evaluate and limit the spread of these pathogens (Yokoe et al., 2014). In general, the research highlights four main strategies for prevention:

1. Early detection of infections
2. Interpersonal contact precautions
3. Hand hygiene
4. Separation/isolation of infected patients

Ever since the 1800's it has been evident that hand sanitation has a direct effect on transmission of diseases (Best & Neuhauser 2004). Research describing the spread of multiresistant pathogens concretes this further: "Hand hygiene has the most beneficial effect, as it limits transmission from all colonized patients, including those that are known and unknown to be colonized." (D'Agata et al. 2012, p. 6). Senior Consultant at Folkehelsa, Bjørn Iversen, also expresses that hand sanitation and surface contact are amongst the most important areas of focus when preventing spread of pathogens, highlighting keys, keychains and keypads as a problem area: When touching these items pathogens are transferred from a person's hand to the surface of the object and left there for someone else to "carry on" when they touch the same item (personal communication, November 26th 2015). Still with all this knowledge on the risks of improper hand sanitation, an ethnographic study performed in 2012 at the American College of Surgeons (ACS) Clinical Congress 2012 and the American Medical Writers Association (AMWA) Congress 2012 found 1 out of 5 doctors not to wash their hands after visiting the bathroom (Burcharth et al. 2014). This neglect amongst even leading authorities highlight the benefits of looking into other methods for minimizing spread of pathogens – post exposure – alongside current routines.

Much of the focus within Norwegian health care has been on touchless sanitation devices, such as faucets and dispensers (personal communication, Senior Consultant at Folkehelsa, Bjørn Iversen, November 26th 2015). Iversen explains that a lot of studies have been done to make sure these items are located in optimal positions/areas – since the elapsed time from dealing with infectious material to sanitation is as important as what areas and body parts are sanitised. At labs dealing with items that can be at varying degrees of contamination the precaution, as told by Camilla Hoff, Specialist Biomedical Laboratory Scientist at The Center for Psychopharmacology at Diakonhjemmet Hospital, is simple: “We treat all our contact and items as contaminated, regardless of their actual state. This gives us one single routine that is strict, but effective for all situations regardless of severity” (personal communication, November 6th 2015). The focus and routines have in both cases shifted: Sanitising stations formerly located around wash stations have been moved closer to where the patients is, or where else they are needed. There has also been an increasing demand for touchless interfaces for both sanitation systems (e.g. faucets or dispensers), tool collections (e.g. cupboards or drawers) and access control (e.g. levers, doors and keypads) and other items healthcare workers interact with throughout the day.

In general, both Iversen and Hoff express the need for better – simpler – design for these touchless interface, explaining there is much to be done both in terms of interaction-, production- and service design related to spread of pathogens via surface contact.

2.2 Access Control & Positioning

In its simplest form, access control works by two components: a lock, and a key. The lock is encoded with a pattern, and will only open if the correct pattern is applied. The key holds the correct pattern for the lock. Modern access control follows the same basic principle: A keycard acts as the key, holding a unique combination. A card swipe slot acts as the lock. When the slot reads the key, the key is checked against a database holding permissions for this key. If permissions for the card are correct, access is granted. Since the security aspect of access control falls outside the scope of this project proposal, this is a simplified explanation, the basic principles however remain the same.

This simplest form of access control, where only a single piece of information needs to be correct (the key), is called a *one component security* or *one factor identification*. The introduction of a second secure component is called a *two-factor identification*. In the old days this would have been a secondary key, with modern key cards the common practice is to have a personal identification number (PIN) working like a secret code only the owner of the card knows. The procedure follows the same principles: The card is swiped in a reader, a PIN is supplied, a database is queried to see if the PIN matches the card and if the card has proper access rights (note: card, not user). Having this two-factor identification, greatly improves security because an attacker needs to have both a physical component, the keycard, and a secret component, the PIN (Yang et al. 2008). (Notice how the card is in fact the one with the access rights, not the user.) Wikipedia gives a great everyday example of Two-factor authentication (Wikipedia 2011c):

> *Two-factor authentication [...] is a technology patented in 1984 that provides identification of users by means of the combination of two different components. These components may be something that the user knows, something that the user possesses or something that is inseparable from the user. A good example from everyday life is the withdrawing of money from a cash machine. Only the correct combination of a bank card (something that the user possesses) and a PIN (personal identification number, i.e. something that the user knows) allows the transaction to be carried out.*

Wikipedia mentions a *third type of identifier* inseparable from the user, this is a *biometric component* - i.e. a part of your body. Currently the most typical biometric component is a fingerprint (Yang et al. 2008) but studies into movement patterns, or *gait* – i.e. the way you walk – show promising applications towards use as a biometric component in two-factor security, since your movement pattern would be unique for you and you alone (Yam, Nixon & Carter, 2004. Derawi et al., 2010).

The prototype design in this thesis, needs to open the door only when the right person is in front of it (see use cases in chapter 3.2 Ethnography). GIven the notion that the user's smartphone will act as their biometric component – the unique and secret part they will always carry with them – it can be used to trigger doors to open based on its location. To achieve this, the person's position relative to the door must be known. The common way to position someone in a 2D space is triangulation between three fixed points on a constant X,Y axis (figure 2.1). To illustrate one could picture a person in the woods: Looking around he would probably see three distinct features: a tree, a rock and some sort of weird, bushy-thing. The rock would be to the left, the tree to the right side and the bush in front of him. If one measured the distance from the person to each of these points, one would be able to, using the *Law of Cosines* and *Pythagoras Theorem*, calculate your position relative to them, giving a fixed position in a *relative space* – i.e. knowing where the person is in the space between the three points. The trick for finding a fixed position in a *fixed space* – knowing where one actually is on earth – is to know the actual/exact position of the tree points used for reference (the tree, the rock and that bushy-thing) giving a fixed position between three fixed points. This process of figuring out where you are based on three fixed items is called *triangulation*.

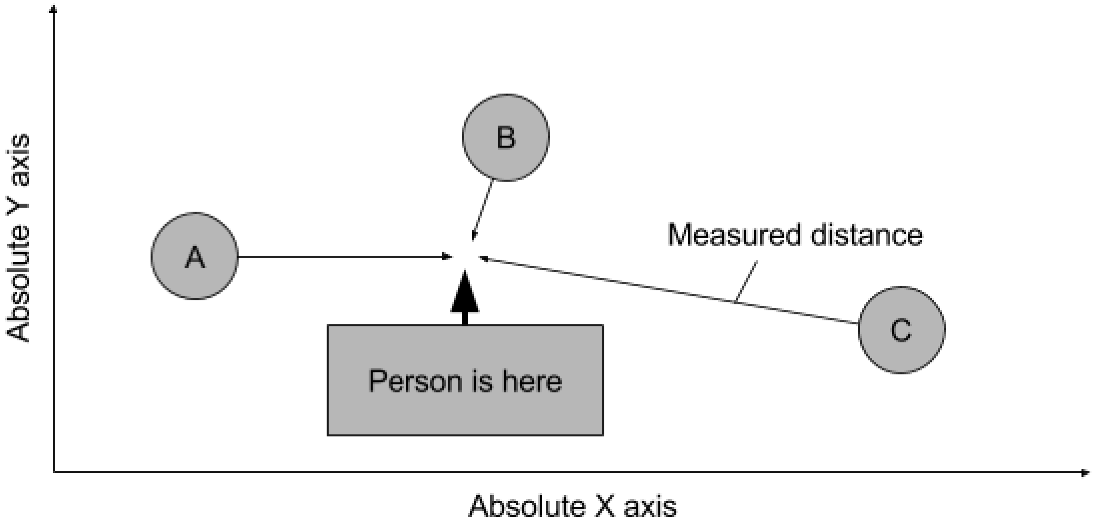


Figure 2.1: Triangulation between points A, B and C. If a person knows the distance to each point he is able to calculate his position between them. And if the exact (and fixed) position of the three points.is known, he now knows his exact position in a fixed space.

The fixed points can, as mentioned, be replicated using digital devices acting like beacons (representing the rock, tree and bushy-thing mentioned earlier) for a device, like a smartphone. Using the beacon as a reference point, the smartphone can exchange data with the beacon to get proximity and distance to it. This distance from multiple beacons, can in turn be used to triangulate its position relative to them. Beacons like the Estimote iBeacons used by Perez Barragan (2014) for indoor positioning in hospitals, enables this kind of ranging. These beacons use the Apple iBeacon protocol to allow developer to easily integrate with the beacons via code in their apps. From Wikipedia (2013):

> *iBeacon is a protocol standardized by Apple and introduced at the Apple Worldwide Developers Conference in 2013. Various vendors have since made iBeacon-compatible hardware transmitters - typically called beacons- a class of Bluetooth low energy (LE) devices that broadcast their identifier to nearby portable electronic devices. The technology enables smartphones, tablets and other devices to perform actions when in close proximity to an iBeacon.*

When calculating the distance to a beacon, the smartphone receives information from the beacon enabling it – and in extension it's user – to "talk" to the beacon. This ability to, in layman's terms, "to exchange information between a person and an electronic device" is nothing new. Technology like Radio Frequency Identifier (RFID), commonly used for access control like key cards, was first patented in 1983 and the base technology later announced as Near Field Communication (NFC) in 2006 was patented already in 1997 (Computerworld 2014, PCT/GB1996/002975). The 1997 version of NFC was even used with Star Wars toys from Hasbro, enabling them to respond to each other when brought together (editor's note).

NFC, RFID and iBeacons all work in similar ways – RFID is in fact the predecessor of NFC. RFID and iBeacons are similar in the way they transmit data without the transmitter being able to store, and NFC is similar to RFID in it being a passive technology requiring a radio field to activate. From Wikipedia (2011b):

> *NFC is a set of short-range wireless technologies, typically requiring a distance of 10 cm or less. [...] NFC always involves an initiator and a target; the initiator actively generates a RF field that can power a passive target. This enables NFC targets to take very simple form factors such as tags, stickers, key fobs, or cards that do not require batteries. NFC peer-to-peer communication is possible, provided both devices are powered.*

This is the same principles as for RFID: A radio frequency is used to broadcast data. But where NFC is fine tuned to work at 5–10cm, RFID works at longer distances. The increased distance is apparent in systems like the European automated toll booth *Autopass*. When using Autopass a car will have a RFID transmitter fixed to the front windshield that is registered when driving through a reader in the form of a toll booth.

Another area where NFC differ is the device and tags can operate. Where RFID and iBeacons allows for two modes, sending or receiving data, NFC can operate in three different modes: 1. read data, 2. save data, 3. transmit data. Usually one device transmits information to another, and the receiving device performs an action based on the received data. A illustration for this data exchange with NFC, is storing contact information on a NFC enabled business card, enabling it to be read by a smartphone, which in turn can store the detail as a virtual business card (Ok et al. 2010).

One of the main differences between RFID, NFC and beacons is therefore at what distance and with what complexity they can operate. All three technology uses radio frequency when transmitting. The wavelength of NFC require close proximity and motion to activate, RFID and iBeacons can operate at a greater distance. NFC and RFID is a passive technology required an active radio field broadcasted by a reader to transmit data. iBeacons run on a power source and is therefore “always on” and transmitting regardless of readers. The decisive factor for choosing iBeacons for this project is however that the iBeacon protocol accounts for the physical distance from the beacon to the reader, thereby making motion detection possible. The distance is calculated by measuring the signal strength from the source, as expressed via a Received Signal Strength Indicator (RSSI). As radio waves decrease over distance following a known formula, the strength of the signal when received will directly correlate with the distance the signal has traveled (Anderson 2014). In addition, where NFC/RFID readers can only communicate with one item at a time, beacons can simultaneously be accessed. By having three beacons transmitting to the same reader, the reader's position can therefore be triangulated (figure 2.2) and by continuously triangulating positions over time, a vector of travel can be established (Andersen 2014).

The purpose for choosing beacons over NFC and RFID in this project is therefore twofold: Firstly, it’s a new technology compared to NFC and RFID; and secondly, the increased operational distance and flexibility of the iBeacon iOS protocol can be better suited for creating touchless access control, where no card, unit or touchpoint need to physically interact – i.e. an invisible UI powered by the Internet of Things.

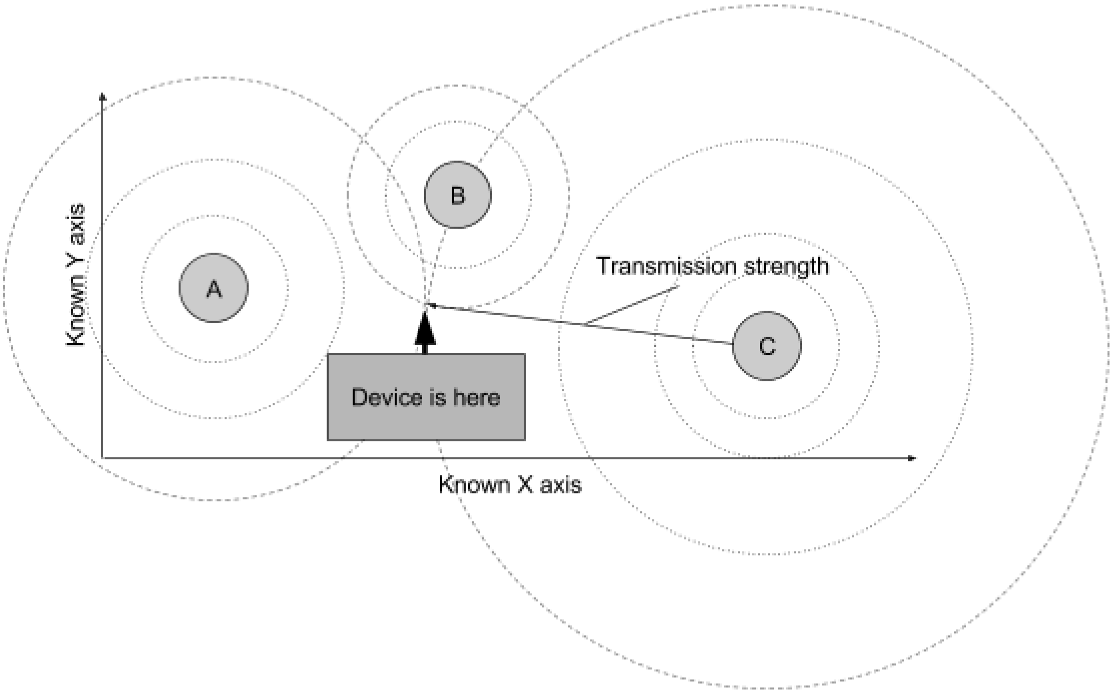


Figure 2.2: Triangulation between three beacons A, B and C. The device knows its distance from each beacon based on signal strength, enabling it to calculate its position between them. As for triangulation in figure 2.1, if the exact (and fixed) position of the three beacons are known, the exact position in a fixed space can be calculated.

2.3 iBeacons in Hospital Environments

In 2011 Daniel Berkvam Hatlevoll explored the general use of Bluetooth Low Energy (BLE) in hospital environments in his thesis *Using Bluetooth Low Energy in Sensor Devices: Possible Applications in Welfare Technology* (Hatlevoll 2011). He found BLE to be suitable and unobtrusive. Sara Perez Barragan built further on this study with her thesis *iBeacon Technology in Welfare: A study of Bluetooth Low Energy for Indoor Positioning.* (Perez Barragan, 2014) to see how indoor tracking and triangulation could increase the effectiveness of healthcare workers. Both studies explored some of the technical implementations and found no hazards towards implementing large scale solutions using BLE, since the frequency does not interfere with technical equipment at hospitals, like EKG[1] and other wireless systems.

When discussing the topic of interference with hospital equipment, with Senior Consultant at Folkehelsa, Bjørn Iversen, he expresses no concern of the matter. “We used to have cellphone free zones like in aircrafts, but we no longer enforce this. There is no known risk of

[1] Electrocardiography: The process of recording the electrical activity of the heart over a period of time using electrodes placed on the skin.

interference from equipment and radio waves." (personal communications, November 26th 2016) There are some restrictions for using cellphones, but these are in regards to distractions and noise control, not technical limitations. And some areas, like where MRI[2] or CT[3] scan machines are located, are considered hazardous to user electronics since they omit huge amount of electromagnetism that can short circuit electronics.

A concrete challenge for the use of BLE beacons, is however the iBeacon protocol and form factors of the beacons themselves. Since the iBeacon protocol is relatively new, no centralized system for security exists (Estimote 2014a). In his book on creating iBeacon application, Matthew S. Gast expresses: "With a relatively simple protocol, iBeacons have straightforward privacy and security implications. Essentially, the protocol provides very little in the way of management tools, and there are huge opportunities to add on to the basic framework." (Gast 2014, p. 38). Estimote, a supplier of beacons and SDK's for use with iBeacons, sums up the security risks into four main types (Estimote 2014a):

1. **Spoofing –** Where an attacker reads your beacon's UUID, configures a secondary beacon with the same UUID and makes your app communicate with the attackers beacon. I.e. steals your UUID and uses it for their own purpose.
2. **Piggybacking –** Where an attacker sniffs your beacon's UUID, implements it into a secondary app and makes your beacon communicate with the attackers app.
3. **Alteration –** Where an attacker changes the UUID of your beacon, making it unknown to your app and renders it useless (or makes it communicate with the attackers app).
4. **Theft –** Where an attacker physically steals your beacon.

To avoid physical theft it is recommended that the beacons are stored at a secure location when not deployed in user testing or during implementation of prototypes. Challenges concerning spoofing, piggybacking and alteration are however not relevant to this project, both due to project scope and since the application will be a prototype only accessible for a short period of time at a controlled, closed location.

[2] Magnetic resonance imaging. A powerful magnet is used to generate images of bone and living tissue.
[3] X-ray computed tomography. Radiation is used to generate images of bone and living tissue.

2.4 Interaction, The Internet of Things & Invisible UI

If the iBeacons and access control are the building blocks of this project, The Internet of Things – an interconnection of devices through wired or wireless technology – is the glue holding it all together. The term was first coined by british technology pioneer Kevin Ashton in 1999, explaining the idea that each device can report key metrics about its use and surroundings, enabling other devices to interpret the data and make calculated choices based on the data available from third party outside sources (Wood 2015). From Wikipedia (2011a):

> *The Internet of Things (IoT) is the network of physical objects or "things" embedded with electronics, software, sensors, and network connectivity, which enables these objects to collect and exchange data. The Internet of Things allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration between the physical world and computer-based systems, and resulting in improved efficiency, accuracy and economic benefit. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure.*

In essence the IoT means your fridge should be able to sense if it is out of milk, relay it to your phone and the phone should remind you to pick up milk when you walk past the convenience store. Employing the IoT towards access control means that an app on your smartphone could tell a physical door to open based on the phone's proximity to a beacon (figure 1.1).

The invisible UI is closely linked with IoT. At its core the invisible UI makes sure a user is not over stimulated by audiovisual or other feedback when performing tasks. It does so by removing the feedback and need for choice from the user and leveraging smart software and hardware to do choices and calculations for the user – i.e. the user can make one choice and the system figures out the rest. The invisible UI does not need to be literally invisible, but rather as frictionless as possible offering the best degree of affordance. In *Don't Make Me Think* Steve Krug (2005) expresses: "[...] every question mark adds to our cognitive workload, distracting our attention from the task at hand. The distractions might be slight but they add up, and sometimes it doesn't take much to throw us." (p. 15) Krug is talking about web interfaces, but the same principles apply for access controlled doors: Shifting concentration from walking to a destination to gaining access, distracts from the initial reason/motive for walking through the door. French writer and pioneering aviator Antoine de Saint-Exupery said it best “A designer knows he has achieved perfection not when there is nothing left to add, but when there is nothing left to take away.” These are the same principles explained by *any* elementary design book: Affordance, safety and control. The UI should offer just enough information to enable the best choice, not distract from the end goal. From this one might conclude that traditional access control is, in its current form – with keycards, passcodes, pressure triggers and door handles – a distraction.

There are however some pitfalls to making a UI invisible. Users normally need to be in control of their situations: If someone loses control of a situation they were suppose to be on top, it might make them lose confidence. It might make them feel unsafe. In 2007 Apple Inc. opened an Apple Store in Tokyo illustrating what happens when you take away control from the user: In the store there is an elevator with no buttons, it simply travels between floors on its own schedule (figure 2.3). Not a first of it’s kind, but it made headlines due to it disruptive nature. Apple removed users’ affordance and ability to control their situations. Over time the elevator was accepted because of Apple’s unique status and history for disruption, but from a usability perspective it remains bad design (Gizmodo 2008). Giles Colborne (2011) explains: “Without the sense of control (calling and directing the elevator) or the sense that a visible person is in control (the guy in front who just pushed a button for your floor) and the feedback that it’s working (the button that illuminates when you push it), all you can do is hand yourself over to the machine and hope.” (p. 104).

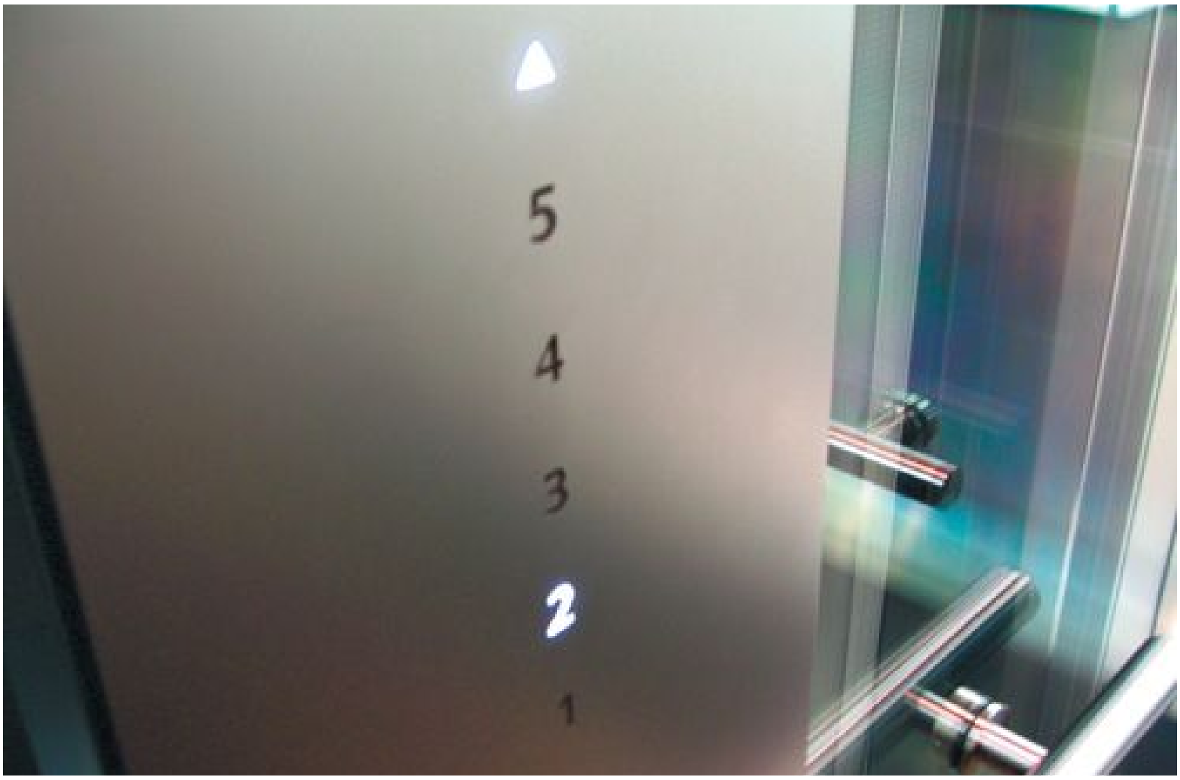


Figure 2.3: Photography of Apple’s button-less elevator in Tokyo. (Colborne 2011, p. 104).

Cognitive and Design Researcher Donald Norman gives some insight into why affordance and control is important to achieve good design: “Affordances provide strong clues to the operations of things. Plates are for pushing. Knobs are for turning. Slots are for inserting things into. [...] When affordances are taken advantage of, the user knows what to do just by looking: no picture, label, or instruction needed. Complex things may require explanation, but simple things should not. When simple things need pictures, labels, or instructions, the design has failed.” (Norman 2002, p. 9). By removing the affordance of an object you potentially remove the user's understanding of how the item operates, rendering it useless.

Another example illustrating the potential danger of altering a pre existing UI is *automated crossings with timers*. These pedestrian crossings have a counter showing the amount of seconds until the next change in light from red to green and vice versa (figure 2.3). In 2011 Statens Vegvesen published a report summarizing field tests of different types automated crossings. The study concluded that although timers decreases the amount of jaywalking by subjectively making the time passed while waiting for a green light seem less; the pedestrians’ focus was shifted from paying attention to actual traffic to the countdown and they would start walking “automatically” the moment the timer reaches zero regardless of traffic. This introduction of visual feedback shift the user’s focus and resulted in dangerous situations (Statens Vegvesen 2011, p. 87).

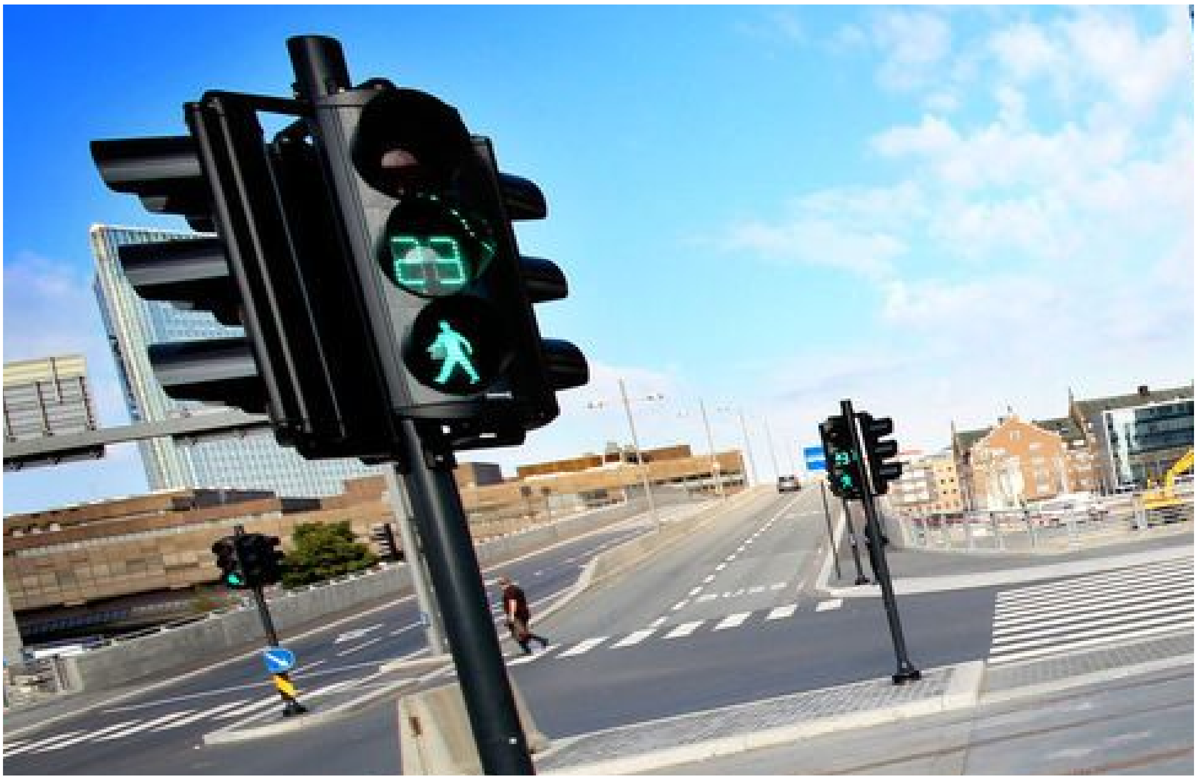


Figure 2.4: Automated crossing with countdown timer. Timer shows green light signaling the street is open for pedestrian crossing. (Source: http://www.dinside.no/930507/tester-gangfelt-med-nedtelling Image by: Ole Petter Baugerød Stokke. Image is subject to copyright and used per agreement).

2.5 Door Design, Automation & Interaction

Things we interact with in everyday life should be intuitive. They should just work. If one walk up to a door it should be obvious how to use it. Herein lies some of the challenges for autonomous doors. How are they obvious? How do they signal if you have access or not? How does one know if they are broken? In his quintessential book on design, *Design of Everyday Things,* Donald Norman explains some of the challenges when dealing with door design: “[...], when the design of something as simple as a door has to come with an instruction manual-even a one word-manual – then it's a failure, poorly designed [...] The violation of simple constraints on doors can have serious consequences” (Norman 2012, p. 87,88) Doors not following this philosophy of design have been nicknamed “Norman Doors”. They operate differently than expected. They need instructions to be operated. For a door – that frankly only has one feature; to open or close – this lack of affordance can only be explained by faulty design. Jonty Sharples, UX Design Director at Albion London, passionately argued at South by Southwest 2013 that this is *arrogant design*: A designer's action that emphasis how they aesthetically want to perceive the world without accounting for how the object actually will be used. By designing this way “[...] someone, somewhere is getting shafted.” (Sharples 2013, 3m 38s), be it the end user or the designers integrity.

Reasons for bad design are as plentiful as there are designers. It might be a result of limited knowledge, time, cost or other external or internal factors. Whatever the reason, you can effectively measure bad design: Donald Norman outlines six key principles to look for when designing a feature for everyday use, as described by Preece, Sharp and Roger (2011):

1. **Visibility** – The more visible functions are, the more likely users will be able to know what to do next. In contrast, when functions are "out of sight," it makes them more difficult to find and know how to use.
2. **Feedback** – Feedback is about sending back information about what action has been done and what has been accomplished, allowing the person to continue with the activity. Various kinds of feedback are available for interaction design-audio, tactile, verbal, and combinations of these.
3. **Constraints** – The design concept of constraining refers to determining ways of restricting the kind of user interaction that can take place at a given moment. There are various ways this can be achieved.
4. **Mapping** – This refers to the relationship between controls and their effects in the world. Nearly all artifacts need some kind of mapping between controls and effects, whether it is a flashlight, car, power plant, or cockpit. An example of a good mapping between control and effect is the up and down arrows used to represent the up and down movement of the cursor, respectively, on a computer keyboard.
5. **Consistency** – This refers to designing interfaces to have similar operations and use similar elements for achieving similar tasks. In particular, a consistent interface is one that follows rules, such as using the same operation to select all objects. For example, a consistent operation is using the same input action to highlight any graphical object at the interface, such as always clicking the left mouse button. Inconsistent interfaces, on the other hand, allow exceptions to a rule.
6. **Affordance** – is a term used to refer to an attribute of an object that allows people to know how to use it. For example, a mouse button invites pushing (in so doing acting clicking) by the way it is physically constrained in its plastic shell. At a very simple level, to afford means "to give a clue" (Norman 1988). When the affordances of a physical object are perceptually obvious it is easy to know how to interact with it.

Further, for an automated door to feel natural it needs to react as fast as we would ourselves. The number of milliseconds elapsed before something stops being instantaneous can be quantified to 100ms (Nielsen 1993). If the time elapsed exceeds 100ms the user’s might loose their “train of thoughts”, undermining the desired effect and yield a bad user experience. As expressed in Nielsen (1993):

> *1.0 second is about the limit for the user's flow of thought to stay uninterrupted, even though the user will notice the delay. Normally, no special feedback is necessary during delays of more than 0.1 but less than 1.0 second, but the user does lose the feeling of operating directly on the data.*

3.0 Methods

The project's focus is on creating further knowledge for practical implementation of invisible UI's when dealing with access control. To frame the study, and give it some real life application, it is introduced to a Biopharmaceutical Lab. The lab environment affords exploring the invisible UI's real life application and how it might help reduce contact pathogen transmission through contact surfaces. The methods used to investigate are:

1. **Literature review**
2. **Ethnographic study** (adapted for HCI)
3. **Prototyping**
4. **User testing**
5. **Data analysis**

The explorative nature of the thesis will stress a good understanding of how healthcare workers encounter access control throughout their day, what their current routines for limiting surface contact is and how they deal with pathogens. To get some broad field knowledge a literature review is conducted focusing on two main aspect: Current use of beacons, NFC and RFID for access control, and the current state of pathogen prevention dealing with surface contamination (see chapter 2.0 Theory, Background & Existing Literature).

Following the literature review, an ethnological study gives better understanding of how users interact with access controls in their native environment, what limitations the current systems have and what possibilities for improvement beacon technology might offer. As described by Preece, Sharp and Rogers (2011): "Ethnography has become popular within interaction design because it allows designers to obtained detailed and nuanced understanding of people's behavior and the use of technology that cannot be obtained by other methods of data collection." (p. 252). The ethnological research forms the basis for several use cases needed for prototyping a solution. The ethnographic study follows "fast and dirty" guidelines adapted to suite Human Computer Interaction (HCI) explained in The Ux Book (Hartson & Pyla 2012): "[...] it takes place in the natural setting of the people being studied; it involves observation of user activity, listening to what user says, asking questions, and discussing the work with the people who do it. [...] In contrast to long-term 'pure' ethnography, [...], the 'quick and dirty' version of ethnography has been adapted for HCI.

Although involving significant shorter time with subjects and correspondingly less of analysis, this version still requires observation of subjects in their own environment and still requires attending to sociality of subjects in their work context" (p. 126).

Rubin and Chisnell (2008) expresses how the value of a prototype can be increased by testing it in its native environment: "Explorative test usually dictates extensive interactions between the participant and test moderator to establish the efficacy of the preliminary design concept. One way to answer very fundamental questions [...] is to develop preliminary versions of the product's interface and/or its support materials for evaluation by representative users" (p. 30). Since the way the targeted user group will interact with the door throughout a day will wary highly based on task, it is deemed best for this study to be performed on site, dealing with real tasks in a real environment. However to account for changes in Diakonhjemmet Center for Psychopharmacology's schedule, and ability to facilitate implementations of the prototypes, the prototypes is tested in an off site semi controlled environment. Creating the touchless interface at a separate location, controlled by the researcher, allows for greater flexibility in terms of access and design, but lower details in actual behavior pattern. This secondary approach lets users from outside the healthcare sector, conducting tests based on predefined use cases.

Use of rapidly produced prototypes is at the project's core. The prototypes make it possible to understand if the use cases discovered in the ethnological study, are covered effectively. The layout found at Diakonhjemmet Center for Psychopharmacology, is replicated and prototypes of increasing fidelity is used to test selected scenarios. However, since "when developing a prototype, one need not represent the entire functionality. Rather, one need only show enough functionality to address the particular test objective" (Rubin & Chisnell 2008, p. 31), the prototype is limited to not include the entire system chain in all tests. This way some tests can focus on the intent to open, and some on the feedback loop when opening, rather than the complete system and functionality as a whole. See chapter 7.0 Limitations for further details.

3.1 Literature Review

The literature studied is covered in sections 1.0 Introduction; 2.0 Theory, Background & Existing Literature; 5.0 Discussion and to some extent in section 3.0 Methods and 4.0 Results.

3.2 Ethnography

The first initial meeting was conducted with Camilla Hoff at The Center for Psychopharmacology at Diakonhjemmet Hospital on February 19th 2016. Hoff gave a guided tour of the premicis and explained in a broad sense how the automatic doors are used. Some of the staff were casually observed and interacted with following loosely adapted versions of questions from the interview guide (appendix 1).

The Pharmaceutical Lab is located in a side building on the premises accessible through three entrances. The lab itself is located on the third floor behind two sets of keycard controlled doors. Entrance to the restricted areas are controlled through personal RFID key cards with matching Personal Identification Number (PIN). The PIN is required if entering restricted zones after working hours or during weekends. Some doors are fully automated, opening after validating a keycard and some are manually operated after the electronic locks opened through validation of key card. The doors locks are RFID controlled through a keypad. In addition there are some doors inside the lab with traditional turn key locks. Of the common entrance doors only the stairwell is fitted with an automatic actuator (figure 3.1).

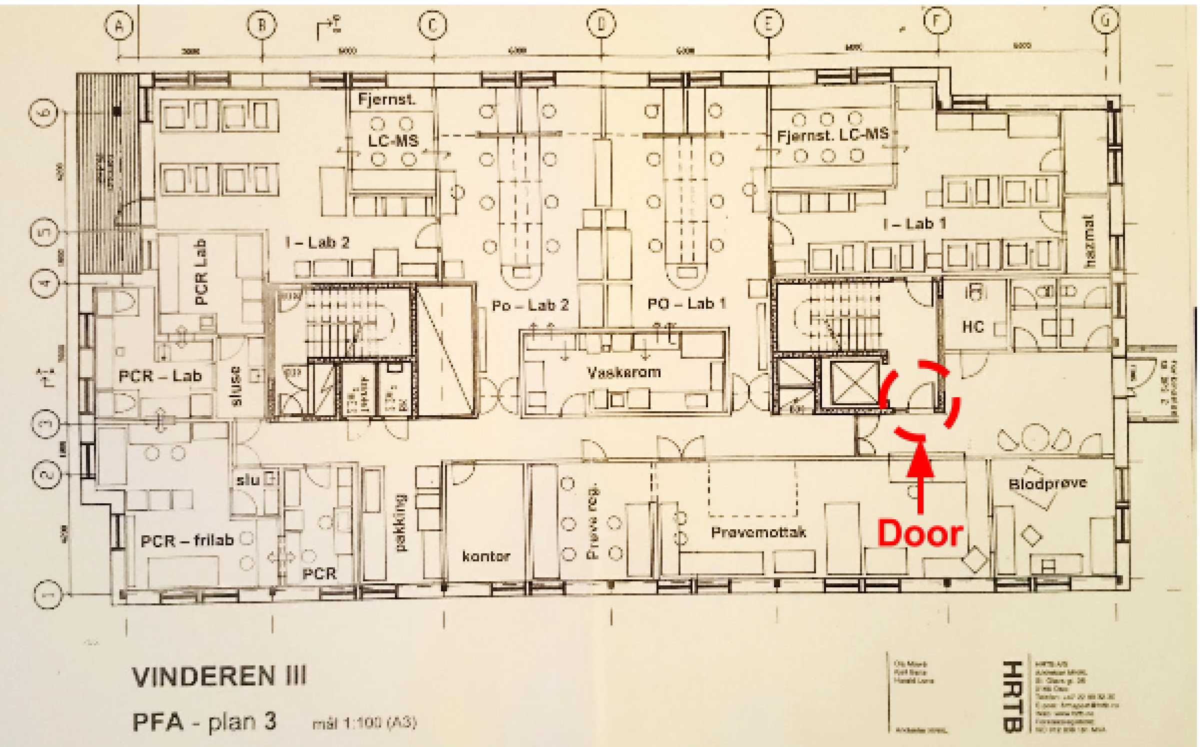


Figure 3.1: Floorplan of the Lab. Stairwells in sector F3 (marked by red, dashed circle) is found to be the most suitable location for testing. Image is subject to copyright and used per agreement.

Through observing the staffs' routines and casual interviews while on location three use cases are of interest when evaluating the use of iBeacons in this hospital setting:

Case 1. Passing through a door.
Case 2. Passing alongside a door without it opening.
Case 3. Passing through a door while carrying equipment.

Case 1 is the default, or "control" use case that is imperative to solve for this thesis. Passing alongside a door has to do with security as a door should not be triggered by a false positive and open without the user's intent. This was part of the staffs concern as it could grant access unauthorised or outside personnel access to the lab, a scenario that would not be an improvement to the current keycard system. Finally for case 3 the staff current need to put equipment and other items they carry on the ground, balance them on one leg or pin them against the wall if they need to free a hand to operate the door controls. As the ground is seen as a "dirty zone" – especially outside of the lab where access control to the restricted zone is located – placing equipment on it increases the risk of cross contamination by surface contact. In addition the chances of human error resulting in damaged or contaminated equipment as a result of "juggling" expensive lab gear and/or test results while trying to free an extra hand to operate the door controls is ominous even with skilled researchers.

3.3 Prototyping

To open doors based on proximity, the prototype uses three Estimote iBeacons from estimote.com, a Raspberry Pi 3 microcomputer, a cloud based PHP Application Programing Interface (API) and a smartphone (figure 1.1). In essence the smartphone uses the beacons to validate whether it should tell the Raspberry to open the door. In more details the components are:

1. **A transmitter –** The beacons.
2. **A receiver –** The smartphone running an application.
3. **A logic circuit –** A PHP cloud service or "endpoint".
4. **An "activator" –** A Raspberry Pi 3.

Each beacon is positioned using double sided tape on the inside walls of the automated doors. To avoid physical theft of the beacons they are stored at a secure location when not deployed in user testing or during implementation of prototypes. Since the prototype is only accessible for a short period of time in a closed location, spoofing, piggybacking and alteration is of no concern the the project. The beacons' SSIDs are therefore used in a “raw format”, without encryption or other security protocols

The receiver is fitted with an app that allows it to locate the iBeacons over Bluetooth (BLE) for triangulation. Ones all three beacons are in range, the app calculates the user's position and vector of travel relative to the beacons. In essence the placement of the three beacons donate a “doorway” the receiver can detect and react to accordingly. When the app calculates that a user's vector will intersect the imaginary doorway denoted by the beacons, it calls an "endpoint" on the cloud service wirelessly over the internet using HTTP protocol through the phone's WIFI, LTE (4G), HSPA/EDGE (3G) or equivalent.

To open the door, the easiest route is – following a rapid prototyping approach – to “hijack” one of the buttons used for opening a door located *inside* the secure zone. As the inside door switches are simple connection switches, that when pressed let's current flow through a wire to the native door control/data bus (a collection of circuitry and wires that perform different operations depending on input) all current through this wire, be it by button press or “hijacking”, will result in the door control thinking it should opening the door as if the user was on the inside – albeit making the door swing “out” towards the user and not away from if the door swings both ways. By having the activator “listens” to instructions from the device outside the secure zone, relayed by the cloud service, it bypasses the button if needed, achieving the desired effect (see detailed explanation for prototype in chapter 3.3.2 and 3.3.3). The layout of the lab, the placement of the automated doors, the presence of an inside door switch and the fact that the door only swings one way, makes the experiment best suited to be carried out at one of the automated stairwell doors in sector F3, marked in red, dashed circle in figure 3.1.

3.3.1 The Cloud Service Logic Circuit/API

The cloud service/endpoint is a PHP (version 5.4) program running on a public server (hosted by one.com). The service is configured using REST principles, where an Application Programing Interface (API) can be used to control the service using “verbs”. For this particular implementation the verbs GET, POST, PUT and DELETE are the ones used, more verbs exist but they are not of interest to this particular implementation. When in operation, the API responds to three main verbs: GET, allowing a service to read or “get” the state of a given gateway; PUT, allowing the state of any given door to be changed to “open” (or to “put” the new state); and DELETE, allowing the state of any given doorway to be changed to “close” (or “delete” its permissions). In addition POST and DELETE can be used when “provisioning” the database – where one builds or removes the database needed for the API to store the states of doors – allowing for an easy way of replication the API to any PHP/MySQL enabled web service. MySQL is a type of database chosen for the experiment since it is readily available. It runs on the same webserver as the PHP application.

Without going further into detail about REST-architecture, it is sufficient to say this design allows both the receiver/activator and the transmitter (mobile app) to seamlessly exchange data with the cloud-app over the internet, using a verbose language for accessible and rapid development. This verbosity also allows for a more clean and human readable code base – a curious choice when making something others might improve on.

Whenever a doorways state is changed using the PUT or DELETE method, four things happen simultaneously as illustrated in figure 3.2:

1. The action is logged in a table named "action_log".
2. The doors new state is updated in the table called "door".
3. The doors "last_update" timestamp is updated to the current time (a feature within the MySQL database itself)
4. Every door that has a "last_update" timestamp older than 30 seconds is set to "close".

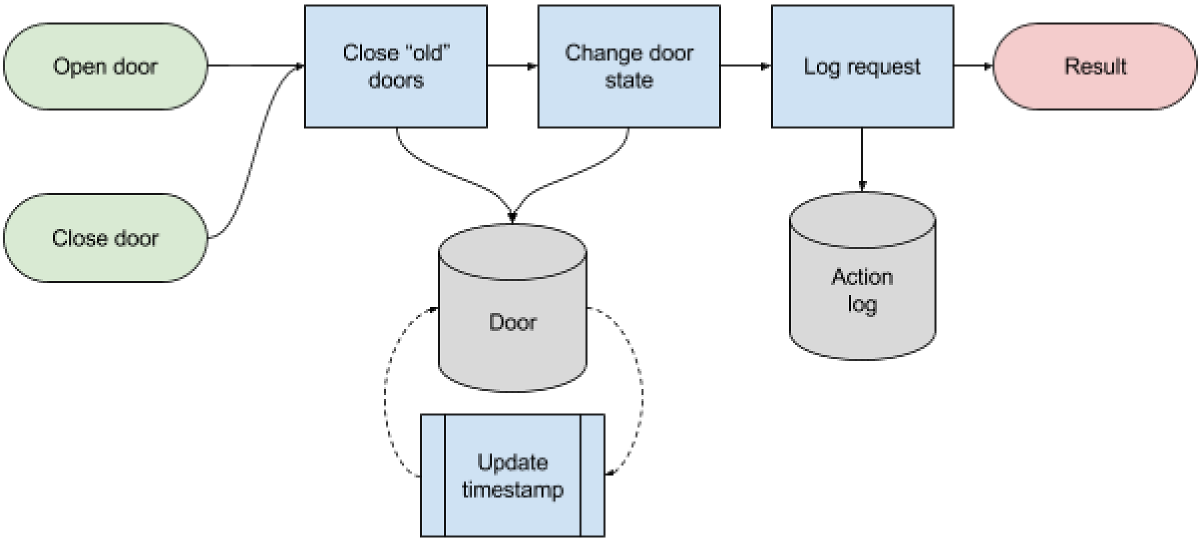


Figure 3.2: Flowchart describing the process that happens when a doors state is changed to “open” or “close” with the API. The flow goes from left to right: All doors with a timestamp older than 30 seconds is updated, the selected door is updated, the request is logged and the result returned.

The automatic "closing" of all open doors is a feature ensuring no doors are left in an open state over longer periods of time during the experiment. This safety feature also circumvents one of the “inherent flaws” of how the program on the Raspberry Pi is running (more on this under descriptions of the Raspberry Pi implementation in chapter 3.3.2). It is however important to note that the state any door is set to in the database, may not be reflected in the physical world. A door marked as "close" might be open and vice versa. The state is only so the receiver can check the state for a given door and act accordingly.

3.3.2 Raspberry Pi 3 Activator

The activator (Raspberry Pi) (figure 3.3) is a small, pocket sized computer. It has several input and output units than can be programmatically operated using Python, a simple yet powerful scripting language. The Raspberry’s operating system is an open source variant of the operative system Linux, called Raspbian accessible from http://raspberrypi.org/. The unit can run either on power supplied directly from the mains electricity using an alternator or from a battery. In either case it needs to be powered at a constant 5V, if the current dips below 5V the Raspberry Pi can short circuit and “fry”, destroying it forever – the technical detail of why this is so falls a bit outside the scope of this thesis, suffice to say the statement should not be challenged. During the experiment the device runs off a 5V 2600mAh lithium ion battery pack. The lithium ion battery ensures the voltage is consistent even while the battery is being drained. When looking at a discharge graph for a similar type of lithium ion battery (figure 3.4) it is apparent that the battery should be checked frequently to ensure it does not dip below a 50-30% charge to ensure optimal voltage. A battery with a visual feedback for charge is therefore optimal (figure 3.5).

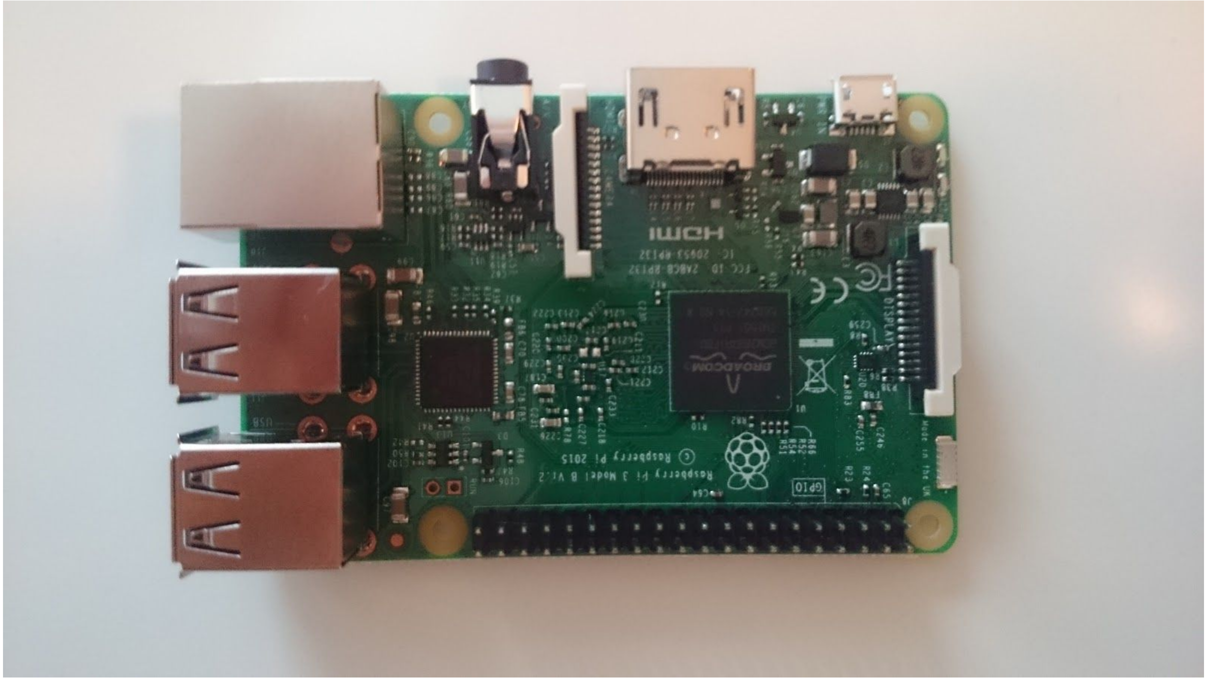


Figure 3.3: The Raspberry Pi 3 is a pocket sized computer running Raspbian, a version of Linux. The unit has multiple General Purpose Input Output (GPIO) pins that allows it to control a multitude of hardware like sensors and relays by running either high or low current through the pins.

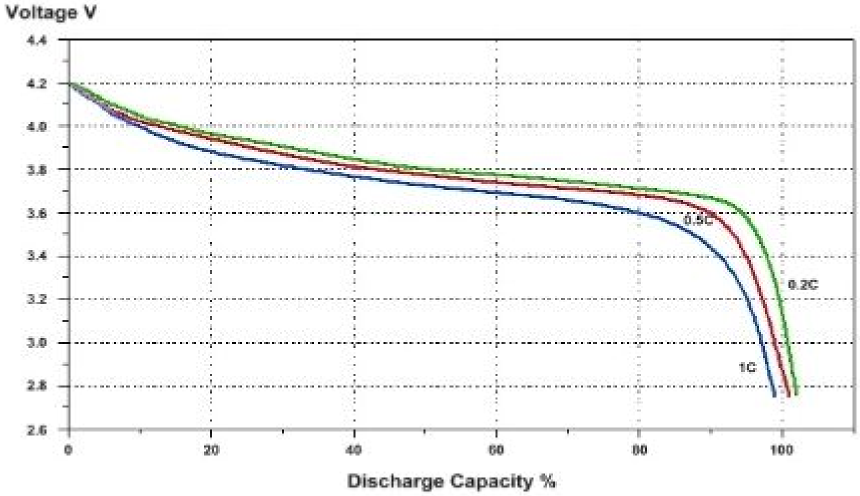


Figure 3.4: Typical discharge rate of lithium ion batteries. The graph shows the results from a 1800mAh 4.2V lithium ion battery discharge test. Note how the voltage drops considerably at ~90%. (Source: http://www.ibt-power.com/Battery_packs/Li_Ion/Lithium_ion_tech.html)

Figure 3.5: 2600mAh lithium ion battery pack with four blue lights indication remaining charge when button is pressed.

The Raspberry is located by the door chosen for the experiment. The door is an automated door operated either with a keypad from the outside or a simple button from the inside. To allow for unintrusive operation of the automated door during the experiment, without hijacking or tapping into Diakonhjemmets door security system, the activator is attached to the button located on the inside using wires attached in a non-intrusive manner. The wires are also connected to a circuit breaker that can be activated by the Raspberry to simulate the same outcome achieved by manually pressing the button – bypassing the normal operation of the button and triggering a “false manual press”. In addition to being the least obtrusive way to install the prototype, having it on the inside of the locked door – inside the secure zone – removed some of the security issues dealing with tampering and theft that might occur, as explained for Estimotes beacons in background literature, chapter 2.3.

The Raspberry Pi runs a small program that routinely polls a door’s state from the API using GET. The program is written in Python allowing it to manipulate the input and output units located on the Raspberry. If the API returns the “open” state for a door, the script tells the Raspberry to send a 3.3V power burst to one of its General Purpose Input Output-pins (GPIO). The pin is connected to a relay board that requires 3.3V to switch on. A relay board is a circuit component that lets a small voltage (in this case 5V) trigger an electronic switch engaging or disengaging a larger, separate circuit (in this case 220V mains). The relay switch on the relay board is in turn connected to the 220V wire bypassing the doors inside opening button, allowing the native system that operates doors at diakonhjemmet to open the door (figure 3.7). The Raspberry Pi polls the door's state every 10ms to be perceived as "instantaneous" by the user (see chapter 2.5. Door Design, Automation & Interaction). However the SRD-05VDC-SL-C relay board connected to the Raspberry Pi is not able to power down when sending the corresponding signal from the GPIO, to bypass this in a "rapid prototyping"-fashion the entire GPIO unit is reset when one second has elapsed, effectively powering down the relay. This method is not recommended as it will reset every input or output device connected to the Raspberry Pi – in addition it might result in some problems if a secondary user approaches the door at the exact same time the circuit is powering down – however for this short experiment it is deemed the most feasible approach since no extra circuitry needs to be created. *Take special note: Never manipulate 220V without proper education or supervision.*

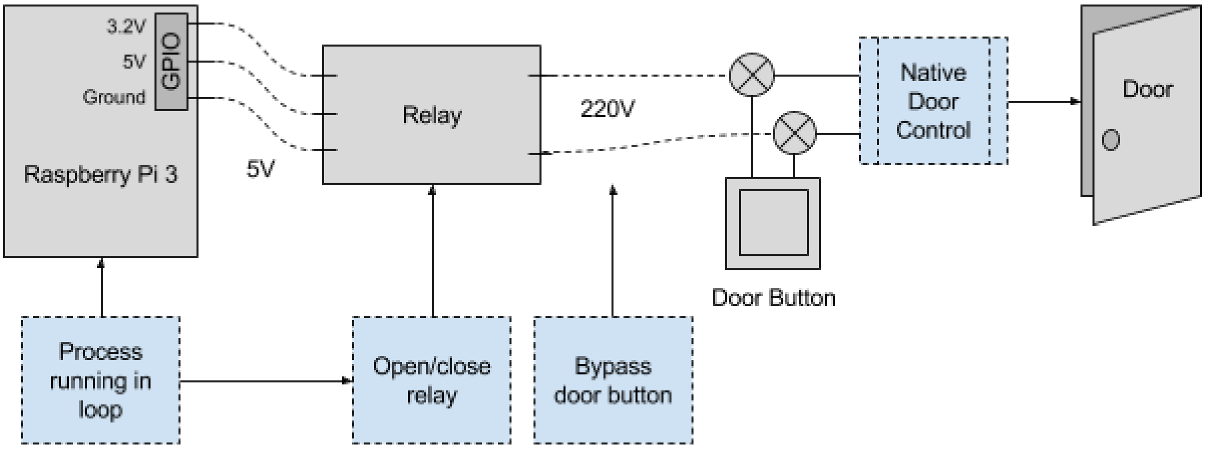
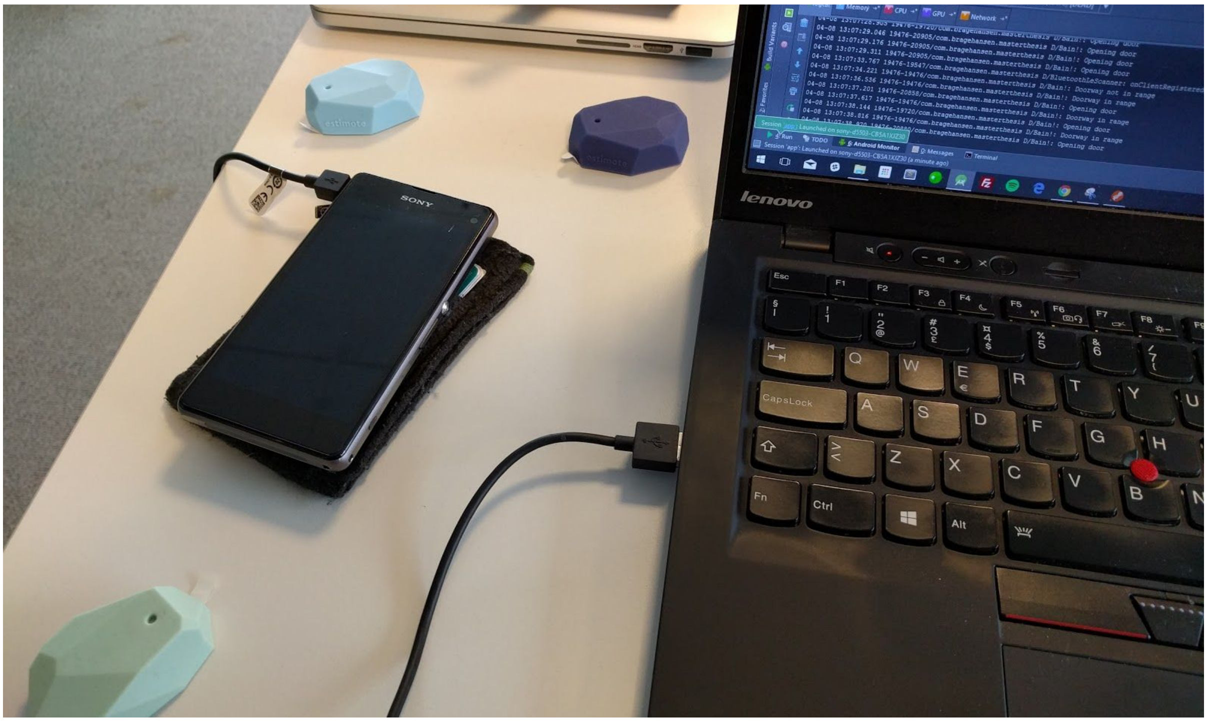


Figure 3.6:A rudimentary sketch of the electrical/logic of the Raspberry Pi 3 bypassing the 220V circuit needed to trigger the native door controller to open the door..

3.3.3 The Smartphone Receiver

The receiver runs a small program (app) that routinely checks for the presence of beacons. When a beacon is located, its distance is measured using a SDK created by Estimote (who also created the beacons used). The SDK allows for easily monitoring of a beacon and easily ranging by translating the broadcasting strength of any given beacon by interpreting the Received Signal Strength Indicator (RSSI) of the transmission. The received signal strength will be influenced by battery strength and obstructions between the receiver and transmitter, so a set of fresh batteries are recommended for each beacon. For the prototype a set of three new beacon, shipped with fresh batteries from http://estimote.com are used.

A process runs continuously in the background thread of the app. It scans for nearby beacons every 100ms. For each scan it checks if the three beacons associated with the doorway is among the discovered. If they are, the ranges are checked, the smartphone's position is triangulated and the vector of travel calculated. If the smartphone is approaching the door a PUT request to the API endpoint toggling the doors state to “open”. For this to work, the beacons need to be stationary at a predefined location (figure 2.2) and the signal strength of each beacon must not wary or flux. The app was developed using Android Studio 1.5.1 running on a Windows 10 portable computer. During the development, the phone was tethered to the computer using an USB 2.0 cable to install and monitor the app (figure 3.7).



```
04-08 13:07:29.046 19476-20905/com.bragehansen.masterthesis D/Bain!: Opening door
04-08 13:07:29.176 19476-20905/com.bragehansen.masterthesis D/Bain!: Opening door
04-08 13:07:29.311 19476-20905/com.bragehansen.masterthesis D/Bain!: Opening door
04-08 13:07:33.767 19476-19547/com.bragehansen.masterthesis D/Bain!: Opening door
04-08 13:07:34.221 19476-19476/com.bragehansen.masterthesis D/BluetoothLeScanner: onClientRegistered
04-08 13:07:36.536 19476-19476/com.bragehansen.masterthesis D/Bain!: Doorway not in range
04-08 13:07:37.201 19476-20858/com.bragehansen.masterthesis D/Bain!: Doorway in range
04-08 13:07:37.617 19476-19476/com.bragehansen.masterthesis D/Bain!: Opening door
04-08 13:07:38.144 19476-19720/com.bragehansen.masterthesis D/Bain!: Doorway in range
04-08 13:07:38.816 19476-19476/com.bragehansen.masterthesis D/Bain!: Opening door
```

Session 'app': Launched on sony-d5503-CB5A1XJZ30 (a minute ago)

Figure 3.7: A Windows 10 laptop running Android Studio 1.5.1 while connected to a Xperia Z1 Compact Three estimote beacons are being ranged. The computer is showing debug data displaying the beacons reported distances to the phone using the Estimote SDK for ranging based on RSSI signal strength.

To simplify the design, two separate functions are invoked; `getGridLocation( distance1, distance2, distance3 ),` which takes the distance of the three beacons as parameters; and `isOnCollisionCourseWithDoor(),` which returns a boolean (true or false). Each function works by defining a virtual grid. Each beacon has a position in this grid. The first function uses algebra, employing the Law of Cosine and Pythagoras Theorem, to return a grid position for the smartphone based on the predefined positions of the beacons, the distance between them and the distance to the smartphone from each beacon. The second function uses vectors. This approach means all beacons must be placed in exact measurements from each other. The choice of using a grid for spatial mapping, is to make debugging the code (analysing output logs and debug messages with Android Studio) easier and the system more visual when analysing user positions and beacon locations.

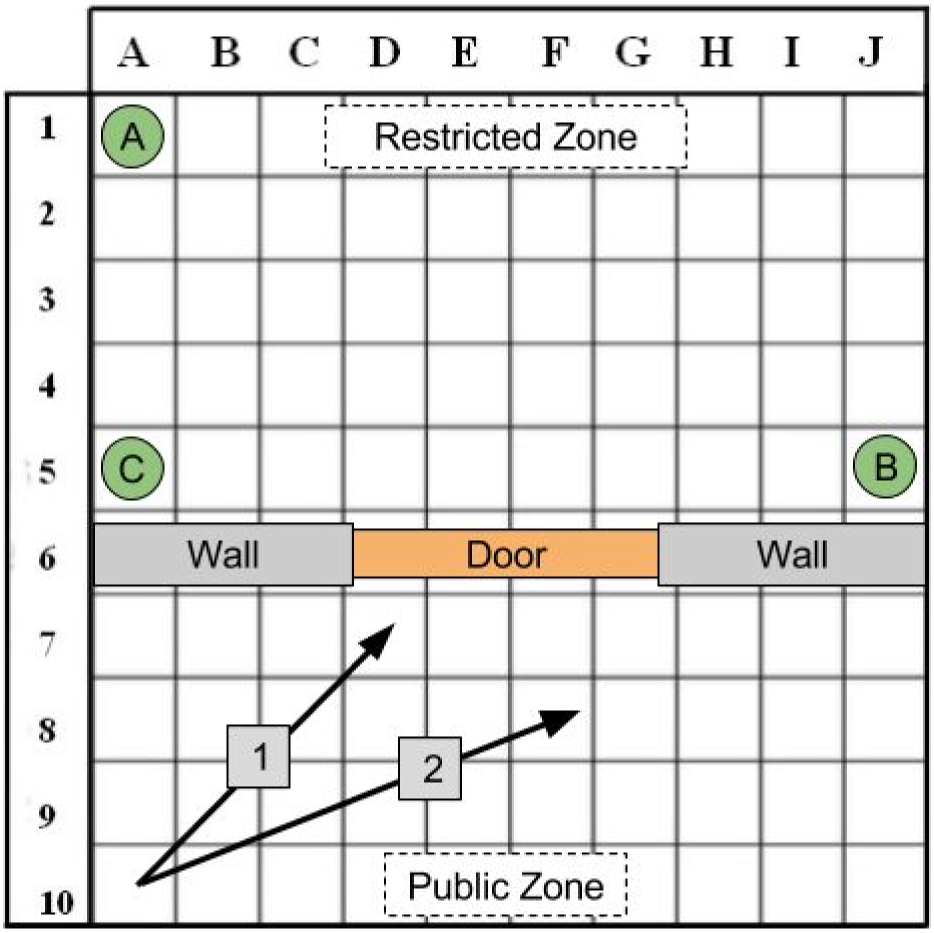


Figure 3.8:The theoretical grid used by the smartphone application, to position the user on a path towards the door. Beacons (A, B and C) are positioned at prefixed distances to the door on the "inside" of the secure zone. The smartphones path (1 and 2) is revealed by sampling its location to points in time. The resulting vector is used to determine if the smartphone will intersect the doorway (1) or "miss" the door (2).

Using path 1 on figure 3.8 as an example: The user starts at position A10 which is calculated by `getGridLocation()` as point 1, then moved to position D7 which is calculated as position 2. `getGridLocation()` uses the Law of Cosine and Pythagoras Theorem in junction to get the position on the X and Y axis. Since the distance from all beacons to the smartphone is known, and the distance between the beacons are known, we can solve for travel along the Y axis by first finding the angle of A, then solving for the height of the triangle drawn by two beacons and the smartphones position (figure 3.9): $Cos(A) = \frac{b^2 + c^2 - a^2}{2bc}$. When Cos(A) is known, the angle of A is found by $A = Cos^{-1}(Cos(A))$, further the height is revealed by $height = b \times Sin(A)$ in addition the travel of X (d in figure 3.9) can be solved by $b^2 = height^2 + d^2 \Rightarrow d = \sqrt{b^2 - height^2}$

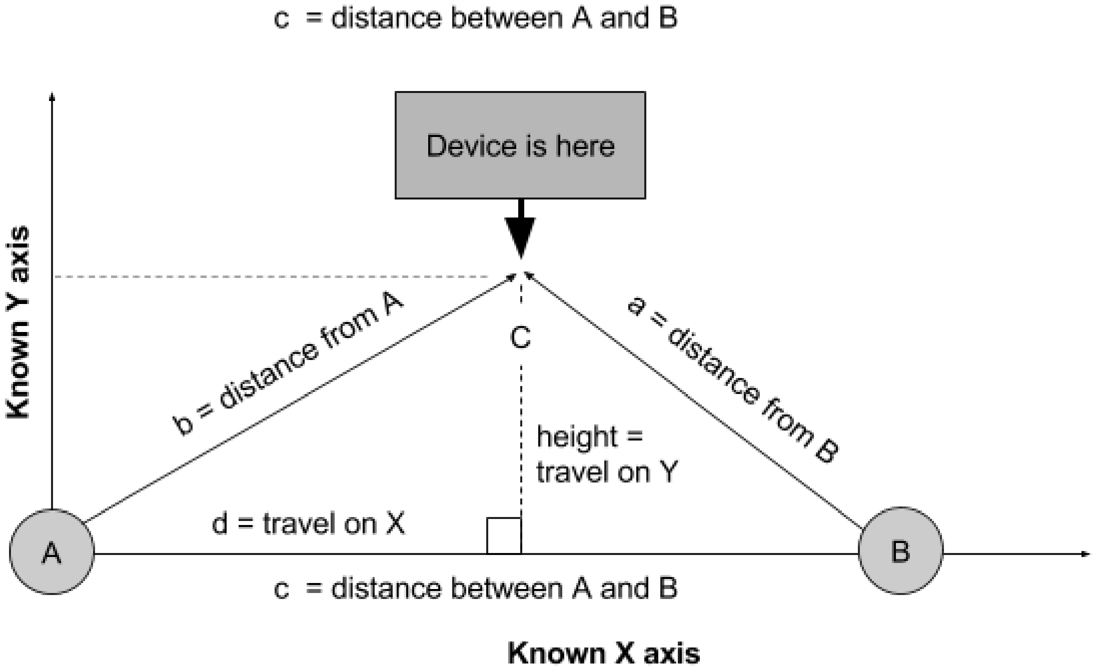


Figure 3.9: Pythagoras and Law of Cosine is used to calculate X and Y travel to give a grid position. A beacon is placed at point A and B, the distance from A to the device is b, the distance from B to the device is a. Using the theorems, the app solves for height (position on Y axis) and d (position on X axis).

The same calculations are done for pairs of each of the three beacons (A, B, C) in different constellations, to calculate the smartphones position in the 10x10 grid when on the move. For each calculation the current grid-position is stored in the app, enabling a second function `userOnPathToDoor()` to use the first and last position measured (A10 and D7) to calculate whether or not they create an approach vector to the door area (D6-G6). This approach is similar to the design suggested by Anderssons (2014) where two beacons are used, one in the restricted zone and one in the public, to determine which zone the smartphone is in. This approach assumes the strongest signal always will come from the beacon located at the same zone as the smartphone, since the closest beacon in theory should yield the strongest signal when the beacons are positioned at equal distance from the door. See appendix 10.3 for further details on code.

3.4 Data Analysis

Data gathered from the prototypes are analyzed and plotted using Google Sheets – a free to use, cloud based “Microsoft Excel equivalent” for processing data. A preliminary classification model is created using *The R Project for Statistical Computing* (R) – an open source scripting language and software for statistical calculations to investigate if the accuracy of the beacons could be increased with current hardware. Standardised techniques are used to calculate min, max, quartile 1, quartile 2, mean and median. An Analysis of variance (ANOVA) test is performed to determine how distinct different groups of data are. All tools are chosen following the ideal of openness and shareability as set forth by chapter A. Acknowledgements.

4.0 Results

4.1 Prototype 1

Prototype 1 consisted of the smartphone app, the Raspberry Pi polling every 10ms, a light bulb connected to a 220V circuit, the cloud API and a cluster of three beacon transmitting at 950ms. When the phone was in range of all beacons a signal was sent to the API telling the Raspberry to turn on the light.

When first prototyped, the iBeacons were found to have a less than optimum response rate for perceived instantaneous interaction. Where the Raspberry Pi polls a doors state every 10 ms, the iBeacons relative position to the phone are only updated every 950 ms. Using Estimotes developer app on the smartphone, the beacon's transmission interval was decreased to the allowed minimum of 100 ms. Although the lowest setting available it is not low enough to meet the planned 10 ms response time.

4.2 Prototype 2

Prototype 2 was the same as prototype 1 but with beacon's transmission speed adjusted to the minimum available 100ms.

The second iteration adjusted for transmission broadcasting cycles that granted a faster response. Still, some of the latency persisted. When deploying the initial prototype for a proof of concept run, a Sony Xperia Z1 Compact (Xperia) smartphone was used. The phone was found not to be powerful enough to continuously monitor all beacons at optimal efficiency. For every “loop” of the monitored beacons – where the program iterates over every single beacon in range one by one and checks if the current beacon is one of the three used for triangulation – the Xperia would run significantly slower than a more powerful Nexus 6. When test users were given the Nexus 6 they would describe the system as “more responsive” and “faster” than for the Xperia. Whether or not this subjective feeling of “slowness” was due to hardware or software issues was not investigated. Later when looking over timestamps of when beacons were read while scanning, on both phones separately, the Xperia was shown to read beacons at a slower rate (field notes, timestamps of measurements from the two phones in debug log).

4.3 Prototype 3

The third prototype used the grid design (figure 3.8) for positioning the user and calculating valid approach vectors.

When calculating grid positions, a new issue was discovered: The beacons vary in transmission strength even when stationary (figure 4.1) throwing the triangulation off and making it impossible to calculate an exact vector or path in high resolution/small areas. Based on 5 sets of 100 distance measurements at increasing distances (table 10.1) the app is shown to report an average inaccuracy of ~±30% ($inaccuracy = standard\ deviation / mean$) (table 4.1).

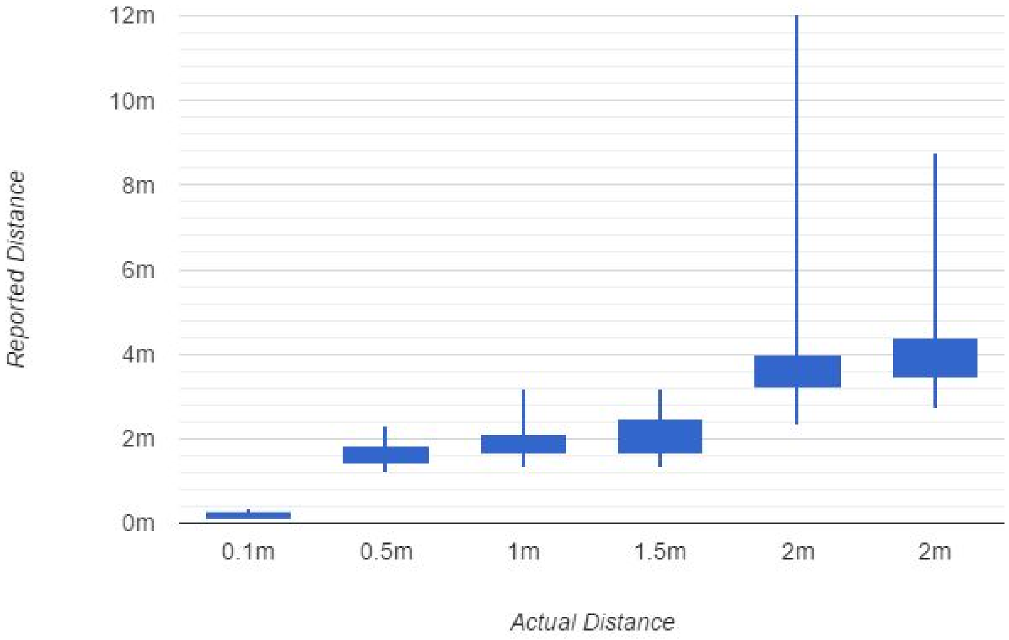


Figure 4.1: Box graph visualizing deviance in distance measurements based on RSSI signal strength as reported by Estimote SDK. The fifth measurement at 2m marked with an asterisk is with a person acting as organic obstruction between the ibeacon and receiver.

Table 4.1:Analyzed results of table 6.1 showing min, max, quartiles, mean and standard deviations for 5 series of 100 range measurements as reported by Estimote SDK using RSSI signal strength of a single beacon. Series marked with an asterisk are measurements done with a human obstruction between phone and beacon.

	Min	**Q1**	**Median**	**Mean**	**Q3**	**Max**	**Standard deviation**
0.1cm	0.1733m	0.2317m	0.2723m	0.2649m	0.2723m	0.3635m	0.0517m
0.5m	1.2082m	1.3856m	1.4145m	1.5810m	1.8456m	2.2982m	0.2837m
1m	1.2989m	1.6244m	1.8456m	1.9854m	2.1163m	3.1719m	1.2989m
1.5m	1.2989m	1.6244m	2.4606m	2.1677m	2.4606m	3.1719m	1.2989m
2m	2.2982m	3.1719m	3.4313m	4.1060m	4.0059m	12.3438m	3.1719m
2m*	2.7041m	3.4313m	3.9136m	4.1565m	4.3835m	8.7609m	3.4313m

However when looking closer at the 500 measurements (table 10.1), there is a statistical significance between some of the datasets at different intervals. This becomes even more apparent when running a *one way analysis of variance test* (ANOVA) comparing data from different ranges to find difference in sets (table 4.2): All clusters of ranges are significantly different, except 1m and 1.5m which ends up with $P \sim = 0.7$ alongside 2m and 2m obstructed with $P \sim = 1.0$.

Table 4.2: Result of one way ANOVA test comparing the five groups between each other.

Pairwise Comparisons		**Tukey-Kramer**	
Factor 1	Factor 2	q	P
0.1m	0.5m	14.2295	0.0001
0.1m	1m	18.6021	0.0001
0.1m	1.5m	20.5730	0.0001
0.1m	2m	41.5300	0.0001
0.1m	2m*	42.0755	0.0001
0.5m	1m	4.3726	0.0292
0.5m	1.5m	6.3436	0.0003
0.5m	2m	27.3006	0.0001
0.5m	2m*	27.8461	0.0001
1m	1.5m	1.9710	0.7308

1m	2m	22.9280	0.0001
1m	2m*	23.4735	0.0001
1.5m	2m	20.9570	0.0001
1.5m	2m*	21.5025	0.0001
2m	2m*	0.5455	0.9989

When each cluster is plotted on a histogram graph, the overlapping of group 1m and 1.5m becomes apparent (figure 4.2). It is also interesting to note that only 29 unique distances are reported within the 500 measurements. See chapter 7.0 Limitations further details on data gathering.

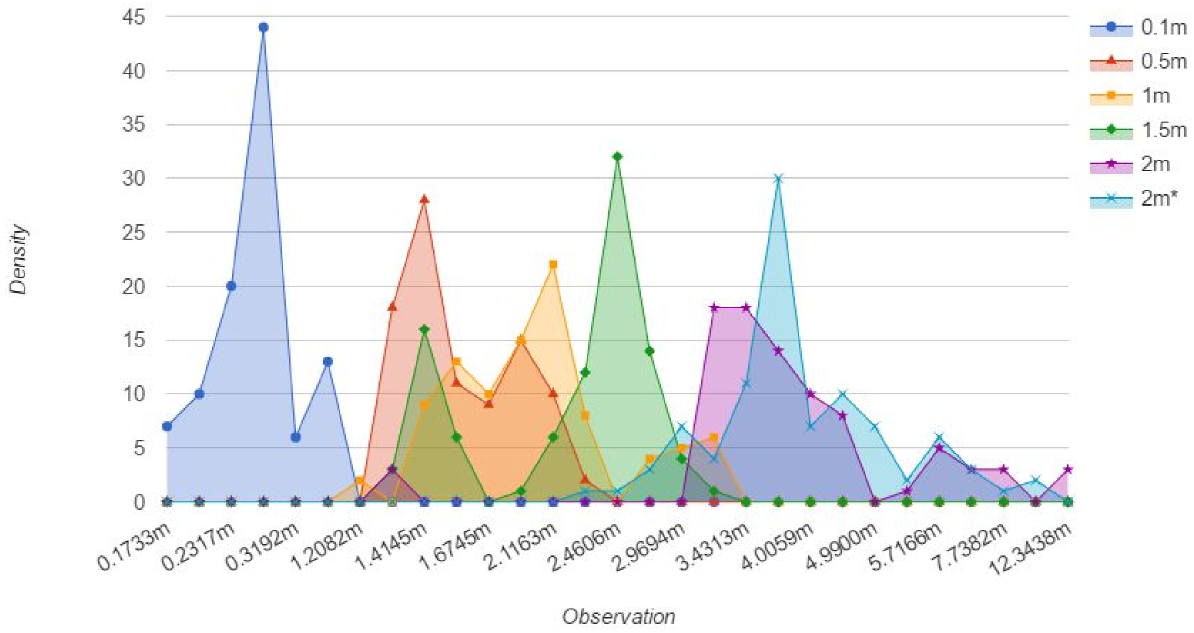


Figure 4.2: Area graph showing 500 data points from 4 different distances form significant different clusters of observations. Each unique observed distance is plotted on the horizontal axis ([0.1733m, …, 12.348m] for a total of 29 unique observations) and the count/density on the vertical axis.

4.4 Prototype 3.5

To adjust for oscillation in signal strength an average of two measured points was used as the reported point by the app. This is the same approach as used by GPS and other systems where multiple triangulations are used to give a more accurate positioning.

Prototype 3.5 granted the same results as for prototype 3. The position was not accurate enough for vector calculation and did not reduce the uncertainty of positioning.

4.5 Prototype 4

The fourth and final prototype scaled back the positioning to only use one estimote beacon broadcasting at 100ms. If the phone was within 0.1m of the door, the door's state was updated using the API regardless of vector or grid position. A light bulb connected to a 220V circuit illustrates the state of the door by turning on when the door state is set to "open".

Scaling down the prototypes to a bare minimum, using a single beacon and a light illustrating when the door should open, still did not grant a response time that made users feel the door was opening as expected. Even with the stronger Nexus 6 phone, respondents reported the door to feel "broken" and did not feel confident using it over traditional methods. The door would open at different distances than expected; sometimes opening at a great distance; sometimes only when on top of the beacon; and sometimes not opening at all resulting in the participants shaking their phone rigorously at, over and around the beacon to "provoke" a response. Figure 4.3 illustrates at what close proximity participants would try to force the interaction between beacon and phone, to happen.

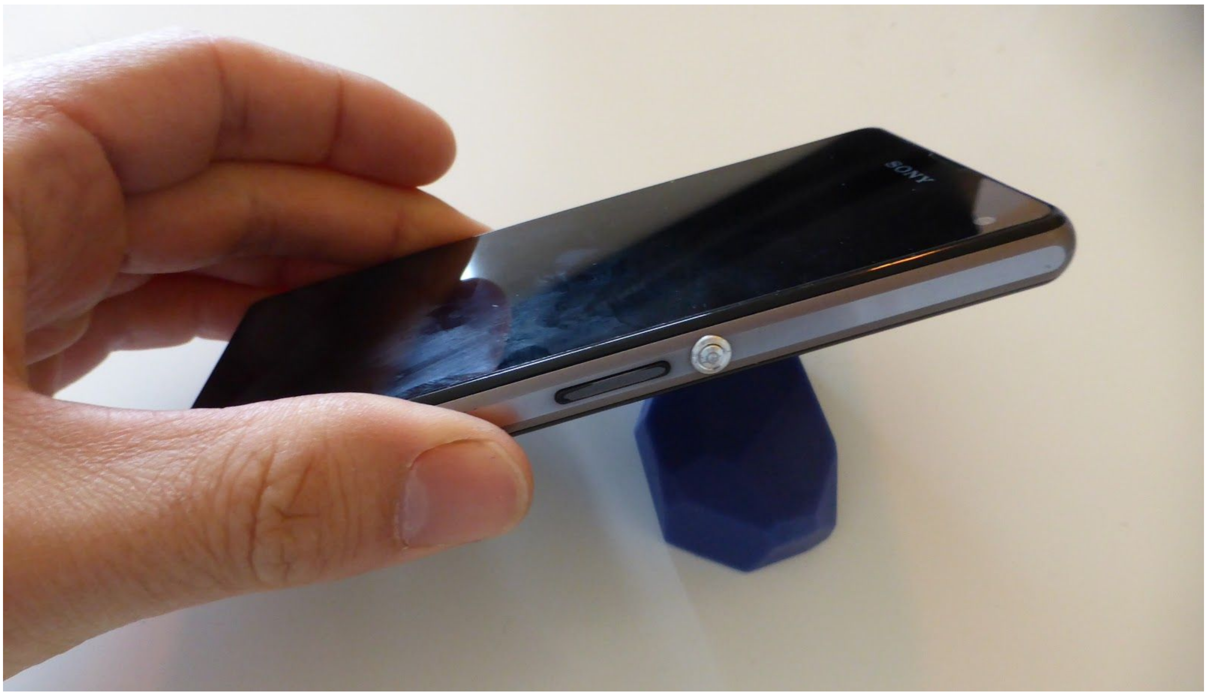


Figure 4.3: Illustration showing the distance at which participants would try to provoke the beacon to register. Used to illustrate is the Xperia Z1 Compact and an Estimote beacon.

5.0 Discussion

The main concern for the users is the system not feeling responsive. To the best of knowledge, this stems from the vector calculations not being precise enough to position the user in a 2D space relative to the door. For the vector calculation to work optimally, you need a high interval (to ensure a lot of coordinates) and an accurate measurement for each coordinate (to ensure rigorous data). When trying to adjust for accuracy using an average position based on multiple points the ~±30% accuracy still means you need more than two points of origin. In their indoor SDK Estimote explains they have improved this by using four beacons for triangulation, although the accuracy is at approximately 1.5m (Estimote 2014b).

When running the app with the smartphone sitting perfectly still on a surface, but with sequentially increasing the distance to the receiver after each measurement (see chapter 7.0 Limitations), the results still show a concerning amount of deviance. As the signal strength – or more accurate the Estimote SDK interpretation of the RSSI – will wary with how long the smartphone is allowed to monitor a beacon before calculating its range, the reported range is ultimately depending on how long each beacon is monitored. Since the app needs a low refresh rate to act instantaneous, this directly counteracts the apps accuracy of measured distance. Tim Baker at Think&Go NFC replied with similatr insight on a relateed issue on the question driven community Quora.com (Baker 2014):

> *From my experience so far it is very difficult to get any reasonable accuracy in a reasonable time as the signal strength (RSSI) is extremely unstable over time and changing environment.*
>
> *The only way to get some decent evaluation of distance is to sample the signal strength over a long period (several seconds) but this means that the reaction time of any calculation is very slow. (several seconds). Things are better when you are very close to the iBeacon (less variation in signal sketch) but just don't believe the claims of "pin point accuracy" or "within a few centimeters" [SIC]*

This statement resonates well both with this thesis measurements into RSSI distance translation (chapter 7.0 Limitations) and other studies all finding it to be highly acceptable to outside interference. However Wo et al. did not find longer sampling times to significantly reduce the error of ranging by measuring RSSI signal strength (Wu 2008). Andersson (2014) confirms this fluctuation when monitoring a single beacon, finding the mean latency in five sets of measurements of the Estimote iBeacons, to be 324 milliseconds. This might mean that although we sample the beacon's signal every 100ms, it will still be slower at responding with proper signal strength, in turn influencing position calculation. Although Andersson shows no indication of what response frequency he uses with his beacons, he did however conclude that "Despite the large difference in the expected latency and the actual latency, the connection latency was concluded to be sufficiently small to not make the technology unusable." (Anderson 2014, p.9) This statement is in hard contrast to the quantitative measurements found in this study, showing the beacons to have a ~±30% inaccuracy in their reported distance using the Estimote SDK. The addition that the reported distance from smartphone to beacon by the SDK did not correlate with the real world distance – showing an average over reporting of distance by 2.1 times the distance at different intervals (table 4.1) – contradicting this statement further. Anderson might be right about the *latency* of the beacon itself not conflicting with Jacob Nielsen's findings of a system responding before 1 second has passed to feel like a natural process (Nielsen 1993). The latency aside, the severity of ranging errors does interfere with the use of the beacons for precision based automation. In any case the systems need for calibration is an absolute necessity when dealing with at least Estimote SDK. Between Anderson and this thesis the definition of "unusable" is however still very much subjective and up for debate.

However, the statistical significance between the measurements at different distances, as found when measuring distances during prototype phase 3 (table 4.2), could be used to circumvent this shortcoming. Given that major distributions are unique within each dataset, as confirmed by the ANOVA test, a classification model should be possible. By taking extensive distance measurements for each position in the grid design (figure 3.8), one should be able able to accurately position a user in the grid based on a subset of measurements. If the user's distance is measured ten times when approaching, the distribution of these measurements can be compared to the pre-gathered distributions data by using the standard deviations of the set of ten, and the pre-calculated standard deviations for each grid position. See further details on the classification system in chapter 8.0 Future Work.

Going back to the "Norman doors" mentioned in chapter 3, where a simple item like a door is rendered useless or hard to understand by faulty design, some of those concerns seem to arise with the automated doors. Allowing the doors to open automatically at distance leverages an inside knowledge of how the doors work, however outsiders will not understand this logic. Where a keypad offers a strong cue of the doors constraints, relying on the automation removes this affordance. Norman expresses (2002, p.87)

> *When we approach a door, we have to find both the side that opens and the part to be manipulated: [...] we need to figure out what to do and where to do it. We expect to find some visible signal for the correct operation: a plate, an extension, a hollow, an indentation-something that allows the hand to touch, grasp, turn or fit into. This tells us where to act. [...] Doors come in amazing variety. Some open only if a button is pushed, and some don't open at all, have neither buttons, nor hardware, nor any other sign of their operation. The door might be operated with a foot pedal. Or maybe it is voice operate, and we must speak a magic phrase ("Open Simsim!") [...] The violation of simple use of constraints on a door can have serious implications.*

The uncertainty that Norman describes when approaching a new and unknown door is the same as observed with the autonomous door, especially when the door did not react in a timely fashion or in other ways than expected; like not opening at all or opening when it should stay closed. To communicate the how and why of these situations, a set of additional feedback should be present, like a light indicating that the door is open, closed or receiving input, or some other form of indication clearly signaling the door has different states that are operated outside of the user's control.

At the design conference South by Southwest in Austin, Texas, associate professor Nick Bowman spoke about constraints in design in the same matter. Where any interaction between two parties relies on cues from one party to the other. The actor will put forth a set of cues for the receiver to interpret and respond to. This interchange of information is continuous and rapid in fluent interaction, but when it is slowed down or halted, the interaction seems broken. The receiver is left in a void with no cues to decipher and a feeling of frustration and sometimes anger can arise (McEwan 2016). If you have a significant other you most likely have experienced this misinterpretation or lack of cues at least ones in your relationship - like when your mother in law suddenly arrives "out of nowhere".

6.0 Conclusion

This thesis has explored four research questions; three concerning door automation in a hospital environment using Bluetooth Low Energy beacons, and a fourth axillary dealing with the added benefits of indoor positioning in the healthcare sector. Use cases was described following an ethnographic study adapted for HCI following the guidelines of Hartson & Pyla (2012). Although technical and practical limitations affected the thesis goal of a high fidelity prototype implemented on-site in a hospital environment, the thesis research questions have still been explored to show satisfactory results. Enough data is gathered from preliminary studies, literature and low fidelity prototypes to conclude on R1–4 as follows:

RQ1: How are access controls encountered, used and manipulated in hospitals?

Access control will to some extent vary from hospital to hospital. For Diakonhjemmet the operation is both technological using key cards and automated doors, and manual using old fashioned keys and pull doors. This mimics the way doors are used in general and seems to depend on the design choices and budget when building or adapting a hospital, rather than design requirements. Some doors are changed on request by lab technicians, as western style saloon door going into the lab allowing for hands free movement both ways through the door.

RQ2: What are the typical obstacles for an invisible UI for access control?

This study found the most challenging obstacle to be technical limitations affecting the speed of operations. Whenever a door did not open as intended the users would be frustrated and proceed to open the door manually or by keycard. As expressed by McEwan et. al (2016), cues is a fundamental part of any interaction. When they are missing or hard to interpret the experience will feel broken or unpleasant. For a door to be well designed, good cues must be present, a thought also entertained by notable designer Don Norman (2002).

To assure the proper delivery of cues to the user, the designed could be altered with a feedback loop following best practice guidelines as Norman Nielsen's 10 heuristics for system design to offer affordance as described by Donald Norman (2002). This feedback loop could be in form of a light signal describing the doors different states, a mechanical flag or something similar clarifying when the door is active and operational, occupied performing an action or in another state. However Norman has a valid point in stating: “The violation of the simple use of constraints on doors can have serious implications.” (Norman 2002. P. 88) so the chosen cues should be investigated, prototyped and user tested.

RQ3: What benefits and limitations do iBeacons offer over pre-existing, conventional access controls in hospital environments?

The beacons afford the ability to track a single user in 2D space inside the hospital. This tracking is however not accurate enough to give further advantages to fully automated door in its current from. To achieve the required accuracy needed for positioning using BLE, a statistical approaches could be used. Alternatively the devices could be used in a simpler form, as checkpoints or broad term locators for personnel, but not for the pin point precision needed to fully replace existing door sensory – like infrared sensors, cameras or in deed push buttons.

RQ4: What are the practical uses of 2D tracking of healthcare personnel working in high contagious areas?

As shown in Hatlevoll (2011), 2D tracking can help locating personnel throughout keypoints and greater areas. The ranging of the beacons afford for a close, medium, far classification of user positions. Gathering of which rooms healthcare workers have been in and thereby which doors they have passed could therefore strengthen the general knowledge of pathogen spread, however the fine details associated with pinpoint positioning explored in this thesis is not possible using current generation BLE, due to interference and varying RSSI signal strength affecting the triangulation using the iBeacon protocol.

* * *

Dr. Stephen Hawking once said “How we connect with the digital world is key to the progress we will make in the future. In the smartest cities, the smartest homes will be equipped with devices that are so intuitive they will be almost effortless to interact with.” (Hawking 2013). Autonomous doors are a part of this vision, but they need to be more responsive than what the current generation of Bluetooth low energy beacons afford using the Estimote SDK “as supplied”. BLE devices using the iBeacon protocol have too much lag, interference and inaccuracy making instantaneous experiences that “just work” more difficult than by using other technologies currently available. This shortcoming might be adjusted using classification models to more accurately position a user at a certain distance by sampling distance often and comparing the distribution of the samples to bigger datasets of existing measurements. This approach will however rely on near instantaneous sampling to neutralise the user’s movement while sampling, which might not be possible using the current version of the iBeacon protocol due to latency when transmitting/reporting.

When autonomous doors however do arrive, as fast scary Star Trek versions or the kind, friendly ones from Hitchhiker's Guide to the Galaxy, it is important they all adhere to tried and true design principles: Feedback, control, cues, affordance and boundaries. These principles are part of how we humans interpret and communicate with each other and the world around us. Diverging from these principles creates uncertainties, unpleasantry and in some cases straight up dangerous situations. In the words of Colborne (2011): “People need to feel in control. They prefer to be pilots rather than passengers. When they’re at the mercy of chance or hidden forces, they become so anxious that they invent superstitious behaviors that help them regain a sense of being in charge, like avoiding the cracks in the pavement or wearing a ‘lucky shirt.’” (p. 104).

Then again, loss of control might just be the path to the future Dr. Hawkins is dreaming about: “Any sufficiently advanced technology is indistinguishable from magic.” – Arthur C. Clarke.

7.0 Limitations

7.1 Beacon Ranging

During the real life implementation of the prototype, fluctuations in the beacon's signal interfered with the prototypes operations. Although replicated and adjusted for during development of the prototype, the oscillation was considerable enough to render the prototype unresponsive at times. To better understand this oscillations, a single beacon was scanned 100 times at 10cm, 50cm, 100cm, 150cm and 200cm for a total of 500 scans (appendix 10.4 table 10.1). The distances reported by the Estimote SDK for each distance was logged, statistically analysed and visualized using Google Sheets. Looking at the data it shows an increasing inaccuracy at range (figure 4.1). In addition the reported range is shown to be much greater than the physical range of the beacon. To check for interference of the signal a fifth measurement was done at 200cm using the researcher’s body as an organic obstruction between the mobile receiver and the beacon. Results for this fifth series show an increased range in distance, matching a drop in signal strength as estimated, as well as a decrease fluctuation in signal strength.

7.2 Implementation of Prototype at Location

Due to security concerns, hardwiring into the trigger switch as illustrated in figure 3.7 was not possible at Diakonhjemmet. The layout of Diakonhjemmet (figure 3.1) was therefore replicated at an alternate location as shown in figure 7.1 and 7.2, and different test subjects were chosen to perform the experiment.

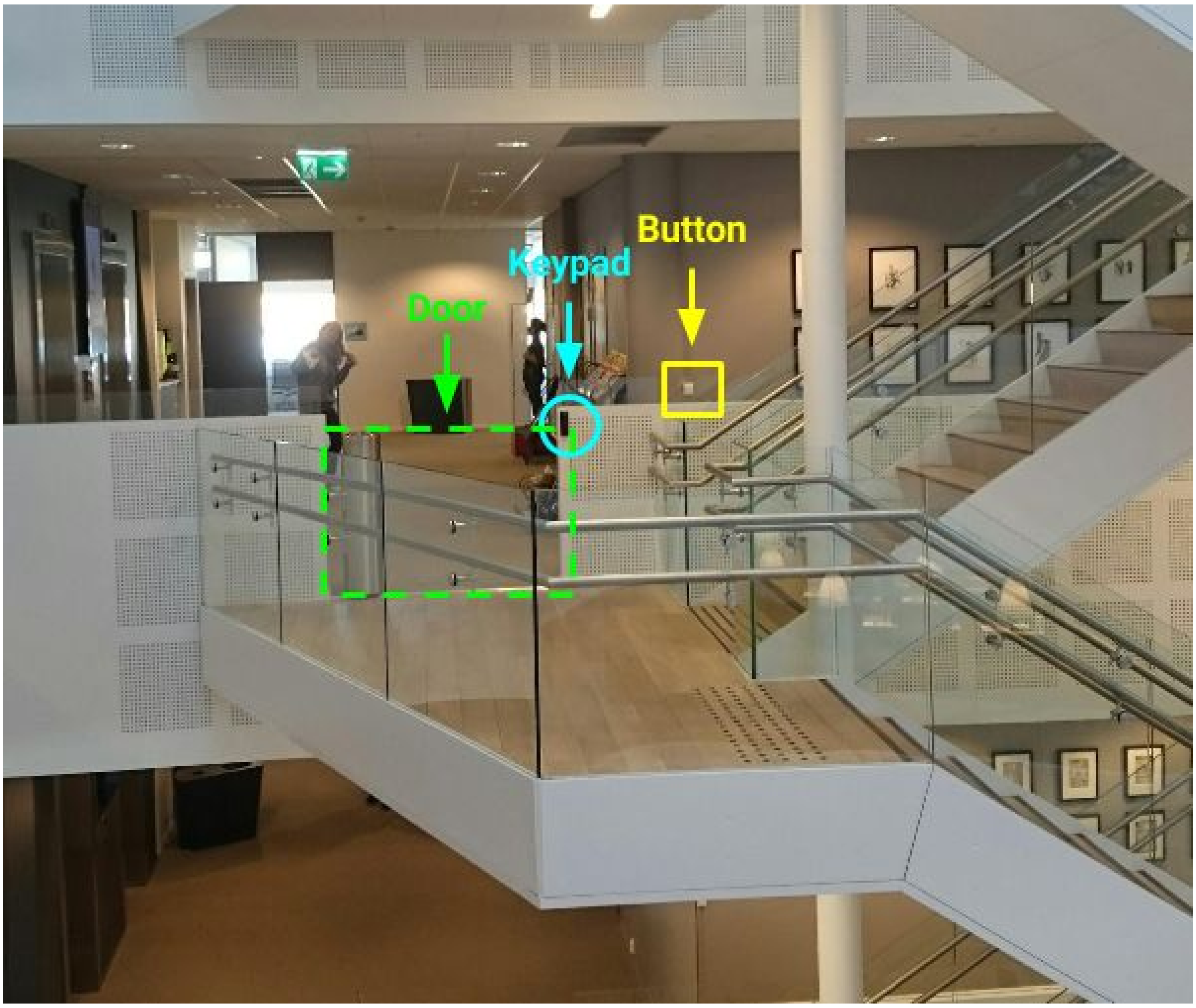


Figure 7.1: Location chosen to approximate diakonhjemmet. The door (marked with dashed square), a waist height automated glass swivel door, is separating a restricted zone (the floor space) from the unrestricted stairwell. A keypad, marked with red full circle, is used as access point and the inside door opening button, marked by yellow full square, as bypass point.

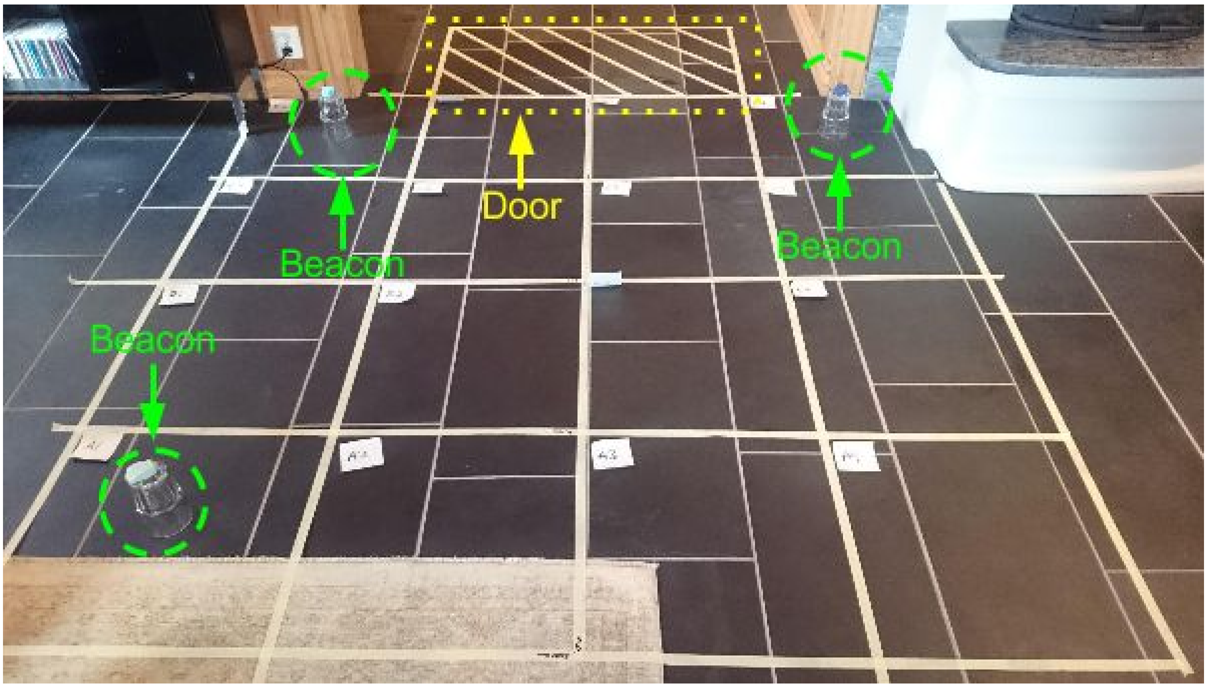


Figure 7.2: Testing location used for prototype version 3 and 3.5. The virtual grid is replicated using painters tape on the floor. Each grid square is given a name (A1-D4) and three beacons are placed inside the grid at one corner each (A1, A4, D1). The "door" is marked by a zagging stripe pattern. The beacons were raised 20cm off the ground by positioning them on "milk-style" glasses.

7.3 Controlling 220V Relay Using Raspberry Pi 3

The selected relay board powering the 220V bypass circuit was found not to be able to be triggered to an *off state* by the Raspberry's 3.3V GPIO pin. Uncertainties remain of this was due to the Raspberry's low signal not being 0V or if the on signal only 3.2V, regardless the issue was easily "hacked" (in a fast and effective manner) by resetting the entire GPIO board to zero volt when the relay needed to be closed (Appendix 10.2). Not doubt not a solution worthy of an electrical engineer, then again this is leveraging the fast and effective approach described in so many books on interaction design and lean/agile prototyping so you can still sleep comfortably if you replicate in this manner.

7.4 Triangulation using iBeacon and RSSI signal strength

Triangulation using Law of Cosine and Pythagoras Theorem based on RSSI signal did not grant an accurate enough position to be useful for indoor position. The indoor position granted by this method was off by as much as 2 meters, putting the user anywhere from outside the grid to behind the door. The added complexity of 3D space meant this could make the prototype trigger a door opening even if the user was on a floor below the door in question.

During investigation into why the RSSI signal strength fluctuates, a series of 500 measurements were taken at 4 different intervals (table 10.1). The measurements confirmed the suspicion of the fluctuating signal, showing it to vary by ~±30%. Further investigation into the series as a whole, show the sampled groups to not have much overlap when distributed and compared on a scale (figure 4.2). Most groups were also shown to be unique in its distribution (table 4.2), allowing for the use of *standard deviations* and *means* to more accurately positioning a user by constructing a classification model based on performing multiple samples.

7.5 Listening to the Cloud Service

The Raspberry Pi listened to the Cloud API by "polling" – or actively asking – the API for the doors status over HTTP. This results in the need to poll every 10ms for a fast response rate. Polling every 10ms can result in an overload of the server and it shutting down in "Denial of Service" (DOS). A better solution would be to leverage "service workers" or "socket connections" allowing the Raspberry Pi to receive status from the API rather than asking for status. This way there are fewer HTTP requests and no DOS.

8.0 Future Work

When investigating RSSI signal fluctuation it was found that the SDK seemingly reports distance based on a predefined set of fixed distances. Of 500 measurements only 29 unique distances were discovered (figure 4.2). These finding could form the basis for a classification mode. Developing a preliminary model using R, show positive preliminary results (appendix 10.5). When clustering data into four groups, [0.1m], [0.5m] [1.0m–1.5m] and [2m], and trained with 80% of the existing data, the model was able to predict the range of the remaining 20% of data with an accuracy of 80%. These results are according to Jonathan Bourne, UCL MSc “The preliminary results are pretty good and indicate that promise that a functioning system could be made, especially if using the combined results of two sensors [...] could be pretty awesome [SIC]” (Personal communication, May 25th 2016).

9.0 References

Aftenposten (2011). *Smittefare ved norske sykehus* [online] URL: http://www.aftenposten.no/meninger/leder/Smittefare-ved-norske-sykehus-6597907.html 6 Nov. 2015.

Andersson, T. (2014). *Bluetooth Low Energy and Smartphones for Proximity-Based Automatic Door Locks.* Linköping University.

Baker, T. (2014) *What is the positioning accuracy of iBeacon (and other BLE) technology?* [online] Quora. URL: https://www.quora.com/What-is-the-positioning-accuracy-of-iBeacon-and-other-BLE-technology 10 Mar. 2016

Best, M., & Neuhauser, D. (2004). Ignaz Semmelweis and the birth of infection control. *Quality and Safety in Health Care*, *13*(3), 233-234.

Burcharth, J., Pommergaard, H., Alamili, M., Danielsen, A. K., & Rosenberg, J. (2014). Hver femte kirurg vasker ikke hænder efter toiletbesøg–et etnografisk feltstudie. *Ugeskrift for læger*. 2014 (25). p. 2364-2367

Colborne, G. (2010). S*imple and usable web, mobile, and interaction design*. New Riders.

Computerworld (2014). *A short history of NFC* [online] URL: http://www.computerworld.com/article/2493888/mobile-payments/a-short-history-of-nfc.html 11 Dec. 2015

Datatilsynet (2015) *Det store datakappløpet - Rapport om hvordan kommersiell bruk av personopplysninger utfordrer personvernet, november 2015* [online] URL: https://www.datatilsynet.no/Global/04_planer_rapporter/kommersialiseringsrapport.pdf 9 Nov. 2015

Derawi, M. O., Nickel, C., Bours, P., & Busch, C. (2010). Unobtrusive user-authentication on mobile phones using biometric gait recognition. *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on*. IEEE.

D'Agata, E., Horn, M. A., Ruan, S., Webb, G. F., & Wares, J. R. (2012). Efficacy of infection control interventions in reducing the spread of multidrug-resistant organisms in the hospital setting. *PLoS One*, *7*(2), e30170.

Estimote (2014a) *iBeacon security: understanding the risks* [online] URL: http://blog.estimote.com/post/104765561910/ibeacon-security-understanding-the-risks 7 Nov. 2015

Estimote (2014b) *Super simple and accurate indoor positioning with the updated Estimote Indoor Location SDK* [online] URL: http://blog.estimote.com/post/118294444205/super-simple-and-accurate-indoor-positioning-with 13 May. 2016

Gast, M. S. (2014, September 26). *Building Applications with IBeacon: Proximity and Location Services with Bluetooth Low Energy*. " O'Reilly Media, Inc."

Gizmodo (2008). *Ginza's Apple Store and the Buttonless Elevator* [online] URL: http://gizmodo.com/313252/ginzas-apple-store-and-the-buttonless-elevator. 5 Des 2015.

Hatlevoll, D. B. (2011). *Using Bluetooth Low Energy in Sensor Devices: Possible Applications in Welfare Technology*. NTNU.

Hartson, R., & Pardha S. P. (2012) *The UX Book: Process and guidelines for ensuring a quality user experience*. Elsevier.

Hawkin, S. (2013) *Stephen Hawking's Science of the Future - Season 1 Episode 4 - Perfect City.* December 6, 2013

Krug, S. (2005). *Don't make me think: A common sense approach to web usability*. Pearson Education India.

Magill, S. S., Edwards, J. R., Bamberg, W., Beldavs, Z. G., Dumyati, G., Kainer, M. A., et al. (2014). Multistate point-prevalence survey of health care–associated infections. *New England Journal of Medicine*, *370*(13), 1198-1208.

McEwan, B., Westerman, D., Sumner, E., Bowman, N. (2016) *Let's Interface: Connecting Social Research to UXD.* Panel Debate at South by Southwest Interactive 2016. Austin, Texas. 12 March 2016.

Norman, D. A. (2002). *The design of everyday things*. Basic books.

Ok, K., Coskun, V., Aydin, M. N., & Ozdenizci, B. (2010). Current benefits and future directions of NFC services. *Education and Management Technology (ICEMT), 2010 International Conference on*. IEEE.

Nielsen, J. (1993) *Usability Engineering (Interactive Technologies)*. Academic Press Ltd..

Perez Barragan, S. (2014). *iBeacon Technology in Welfare: A study of Bluetooth Low Energy for Indoor Positioning.* NTNU.

Preece, J., Sharp, H., & Rogers, Y. (2011). *Interaction Design-beyond human-computer interaction 3rd Edition*. John Wiley & Sons.

Rubin, J., & Chisnell D. (2008). *Handbook of Usability Testing Second Edition: How to Plan, Design and Conduct Eff*ective Tests. Wiley Publishing Inc.

Sharples, J. (2013) *Arrogance & Confidence in Design*. South by Southwest Interactive 10 March 2013, Austin, TX, USA. Last viewd 5 March 2016 https://soundcloud.com/officialsxsw/arrogance-confidence-in-design

Statens Vegvesen (2011). *Gatekryss i bysentrum* [online] URL: http://www.vegvesen.no/Fag/Publikasjoner/Publikasjoner/VD+rapport/_attachment/257209?_ts=13200cebf70. 5 Des 2015

Wikipedia (2011a) *Internet of Things* [online] URL: https://en.wikipedia.org/wiki/Internet_of_Things 8 Nov. 2015

Wikipedia (2011b) *Near field communication* [online] URL: https://en.wikipedia.org/wiki/Near_field_communication 07 Nov. 2015

Wikipedia (2011c) *Two-factor authentication* [online] URL https://en.wikipedia.org/wiki/Two-factor_authentication 29 Nov 2015

Wikipedia (2013) *iBeacon* [online] URL: https://en.wikipedia.org/wiki/IBeacon 07 Nov. 2015

Wood, Alex (2015) *The internet of things is revolutionising our lives, but standards are a must* [online] London: The Guardian. URL: http://www.theguardian.com/media-network/2015/mar/31/the-internet-of-things-is-revolutionising-our-lives-but-standards-are-a-must 8 Nov. 2015

Wu, R. et al. (2008) Study of characteristics of RSSI signal. *Industrial Technology, 2008. ICIT 2008. IEEE International Conference on* 21 Apr. 2008: 1-3.

Yam, C., Nixon, M. S., & Carter, J. N. (2004). Automated person recognition by walking and running via model-based approaches. *Pattern Recognition*, *37*(5), 1057-1072.

Yang, G., Wong, D. S., Wang, H., & Deng, X. (2008). Two-factor mutual authentication based on smart cards and passwords. *Journal of Computer and System Sciences*, *74*(7), 1160-1172.

Yokoe, D. S., Anderson, D. J., Berenholtz, S. M., Calfee, D. P., Dubberke, E. R., Ellingson, K. D., et al. (2014). A compendium of strategies to prevent healthcare-associated infections in acute care hospitals: 2014 updates. *American journal of infection control*, *42*(8), 820-828.

10.0 Appendix

10.1 Interview guide, unstructured interview

The interviews are unstructured conversations based around open ended questions. The topic for the conversation is spread of pathogens, sanitation in the area where the participant works and revolves around the following questions:

1. From your perspective, what is the greatest challenge when trying to reduce the spread of pathogens?
2. What current measures are you aware of that help prevent spread of pathogens at hospitals?
3. What role does surface contact have in the spreading of pathogens, and how much does surface area contribute to the spread?
4. How are the HSE routines related to reducing spread of pathogens on surfaces?
5. Are there different approaches in Norway and abroad? How do they differ? Why do they differ?
6. Are you aware of any projects that directly relates to security touch points, like key cards, keys or key code input, and their role in spread of pathogens?

10.2 Raspberry loop code

Assert of source code for the Raspberry controller unit. `openDoor()` will configure the GPIO each time it is called rather than just switching current to the GPIO. `closeDoor()` will reset the entire GPIO board. Full code available at https://github.com/sbhansen/master-thesis

```
# Trigger the relay to switch on current
# by configuring GPIO and applying current to pin.
#
# @param (int) doorId
#
# @return void
#
def openDoor( doorId ):
    print( "Opening Door %s" % doorId )
    pin = doorPinMapping[ doorId ]
    GPIO.setmode( GPIO.BCM )
    GPIO.setup( pin, GPIO.OUT )
    GPIO.output( pin, GPIO.HIGH )

# Trigger the relay to switch off current
# by resetting GPIO setup.
# set a given door's state to "open" in the API.
#
# @param (int) doorId
#
# @return void
#
def closeDoor( doorId ):
    print( "Closing Door %s" % doorId )
    query = { "id": doorId }
    respons = requests.delete( doorEndpointUrl, params=query )
    ## hack workaround to
    ## GPIO.output( pin, GPIO.LOW )
    ## not working as expected
    GPIO.cleanup()
```

10.3 Android Application

Assert of android Application code running on the smartphone. The code runs at a loop and acts whenever a beacon's position is updated. When all three beacons are in range the app triangulates and calculates the grid position using sets of two beacons before checking if the user is on an approach vector to the door. Since beacons were shown to not grant accurate positions `userOnPathToDoor()` was never fully developed and a placeholders were used.

```
// A listener that checks if all beacons are in range
// If all beacons are in range the users position is triangulated and logged.
// For each new position the last vector of approach is calculated to see if the user is
// intersecting with the door.
beaconManager = new BeaconManager(this);
beaconManager.setForegroundScanPeriod( scanForMilliseconds, delayForMilliseconds );
beaconManager.setRangingListener(new BeaconManager.RangingListener() {
    @Override
    public void onBeaconsDiscovered(Region region, List<Beacon> list) {
        Map <String, Double> distances = new HashMap<>();
        for (Beacon beacon : list) {
            Integer minor = beacon.getMinor();
            Integer major = beacon.getMajor();

            if (beacons.get( major + ":" + minor ) != null) {
                Double distance;
                distance = Utils.computeAccuracy( beacon );
                String name = beacons.get(major + ":" + minor);
                distances.put( name, distance );
            }
        }

        // If all three beacons are in range, plot the phones position.
        if ( distances.size() == 3 ) {

            // Height of triangle between 1:1, 1:4, and position gives us the x position.
            // We use to sets of triangulations to get the average reported position
            // based on the three measurements.
            List<String> positionList = new ArrayList<>();
            positionList.add(getDistanceFromPlane( 2.0, distances.get("ICE"),
distances.get("MINT"), false ) );
```

```
            positionList.add(getDistanceFromPlane( 2.0, distances.get("MINT"),
distances.get("BLUEBERRY"), true ) );
            coordinates.add( getAveragePosition( positionList ) );

            if( userOnPathToDoor() ){
                DoorOpener doorOpener = new DoorOpener();
                doorOpener.execute( doorId );
            }
        }
        else {
            Log.d(TAG, "To few beacons " + distances.size() );
        }

    }

    // Returns boolean (true/false) depending on if coordinates gathered from beacons are
    // on approach vector with door and last position is inside the "openDoorAtDistance"
distance.
    // If we are not concerned with use case This can also be solved by checking if the
user is in in one of the grid positions
    // adjasent to the door on either side.
    private Boolean userOnPathToDoor(){
        String firstCoordinate;
        String secondCoordinate;
        return true;
    }

    // Return the average grid position based on all positions in list
    private String getAveragePosition( List positions ){
        Integer xPos = 0;
        Integer yPos = 0;
        for (int index = 0; index < positions.size(); index++ ) {
            String[] pos;
            pos = positions.get( index ).toString().split(":");
            xPos += Integer.parseInt( pos[0] );
            yPos += Integer.parseInt( pos[1] );
        }
        return ( Math.round( xPos / positions.size() ) ) + ":" + ( Math.round( yPos /
positions.size() ) ) ;
    }

    // Calculates and returns the distance from the base of the triangle using
    // origo (A) and the outermost X or Y (B) as the base of the triangle (c).
    // Distance a and b refers to the length of the two other legs of the triangle
    // Making h the height from the point where a and b meets (C) the distance from b,b to
    // origo (A) and 0,Y (B) in the triangle. See report figure additional information.
    //
    //                       C = unknown point
    //                      /|\
    //                     / | \
    //    legBlength = b  / h|  \ a = legAlength
    //                   /   |   \
    //       beacon 1 = A ---|--- B = beacon 2
    //                     d, c = entire base
    //
    private String getDistanceFromPlane( double legLengthA, double legLengthB, double
baseLength, boolean rotate90degrees ){
        Double a = legLengthA;                           // distance to point A
        Double b = legLengthB;                           // distance to point B
        Double c = ( baseLength  / gridSpacingMeeters ); // distance between point A and B
        Double A;                                        // Angle A.
        Double C;                                        // Angle C.
        Double height;                                   // Distance from origo in Y
direction
        Double length;                                   // Distance from origo in X
direction (d)
        Double cosA;

        // Using Law of Cosine to calculate Cos( A )
        // Cos( A ) = ( b^2 + c^2 - a^2 ) / 2bc
        cosA = ( Math.pow( b, 2 ) + Math.pow( c, 2 ) - Math.pow( a, 2 ) ) / ( 2 * b * c );

        // Using law of cosine to find A in radians
        // A = cos^-1( cos( A ) )
        A = Math.acos(cosA);

        // Using Law of cosine to calculate h form A
        // a = b * Sin( A )
        height = b * Math.sin( A );

        // Using pythagoras theorem to find d
        // leg^2 + leg^2 = hypotenuse^2 =>
        // d^2 + height^2 = b^2
        // d = sqrt( b^2 - h^2  )
        length = Math.sqrt( Math.pow( b, 2 ) - Math.pow( height, 2 ) );

        // Dependent on if we are measuring the baseline or the "edge" of our grid.
        // If we are using the edge (position 4:4 and 1:4) the distances must be adjusted
        // relative to origo (1:1)
        Integer posX;
        Integer posY;
        // Map X/Y and adjust for grid size
        if( rotate90degrees ){
            posX = (int) Math.round( ( baseLength - height )  / gridSpacingMeeters );
            posY = (int) Math.round( length / gridSpacingMeeters );
        }
        else {
            posY = (int) Math.round( height  / gridSpacingMeeters );
            posX = (int) Math.round( length / gridSpacingMeeters );
        }

        return posY.toString() + ":" + posX.toString();
    }
});
```

10.4 Beacon Ranging data

Table 10.1:100 measurements at increasing distances as reported by the Estimote SDK, ordered by distance.

0.1m	0.5m	1m	1.5m	2m	2m obstructed
0.1733m	1.2082m	1.2989m	1.2989m	2.2982m	2.7041m
0.1733m	1.2082m	1.2989m	1.2989m	2.4606m	2.7041m
0.1733m	1.2082m	1.2989m	1.2989m	2.7041m	2.9694m
0.1733m	1.2082m	1.2989m	1.2989m	2.7041m	2.9694m
0.1733m	1.2989m	1.2989m	1.2989m	2.7041m	2.9694m
0.1733m	1.2989m	1.2989m	1.2989m	2.9694m	2.9694m
0.1733m	1.2989m	1.4145m	1.4145m	2.9694m	2.9694m
0.2020m	1.2989m	1.4145m	1.4145m	2.9694m	3.1719m
0.2020m	1.2989m	1.4145m	1.4145m	2.9694m	3.1719m
0.2020m	1.2989m	1.4145m	1.4145m	2.9694m	3.1719m
0.2020m	1.2989m	1.4145m	1.4145m	2.9694m	3.1719m
0.2020m	1.2989m	1.4145m	1.4145m	2.9694m	3.1719m
0.2020m	1.2989m	1.4145m	1.4145m	2.9694m	3.1719m
0.2020m	1.2989m	1.4145m	1.4145m	2.9694m	3.1719m
0.2020m	1.2989m	1.4145m	1.4145m	2.9694m	3.1719m
0.2020m	1.2989m	1.4145m	1.4145m	3.1719m	3.1719m
0.2020m	1.2989m	1.4145m	1.4145m	3.1719m	3.1719m
0.2317m	1.2989m	1.6244m	1.6244m	3.1719m	3.1719m
0.2317m	1.2989m	1.6244m	1.6244m	3.1719m	3.1719m
0.2317m	1.2989m	1.6244m	1.6244m	3.1719m	3.1719m
0.2317m	1.2989m	1.6244m	1.6244m	3.1719m	3.1719m
0.2317m	1.2989m	1.6244m	1.6244m	3.1719m	3.4313m
0.2317m	1.2989m	1.6244m	1.6244m	3.1719m	3.4313m
0.2317m	1.2989m	1.6244m	1.6244m	3.1719m	3.4313m
0.2317m	1.2989m	1.6244m	1.6244m	3.1719m	3.4313m
0.2317m	1.4145m	1.6244m	1.6244m	3.1719m	3.4313m
0.2317m	1.4145m	1.6244m	1.6244m	3.1719m	3.4313m

0.2317m	1.4145m	1.6244m	1.6244m	3.1719m	3.4313m
0.2317m	1.4145m	1.6244m	1.6244m	3.1719m	3.4313m
0.2317m	1.4145m	1.6244m	1.6244m	3.1719m	3.4313m
0.2317m	1.4145m	1.6745m	1.6745m	3.1719m	3.4313m
0.2317m	1.4145m	1.6745m	1.6745m	3.1719m	3.4313m
0.2317m	1.4145m	1.6745m	1.6745m	3.1719m	3.9136m
0.2317m	1.4145m	1.6745m	1.6745m	3.1719m	3.9136m
0.2317m	1.4145m	1.6745m	1.6745m	3.1719m	3.9136m
0.2317m	1.4145m	1.6745m	1.6745m	3.4313m	3.9136m
0.2317m	1.4145m	1.6745m	1.6745m	3.4313m	3.9136m
0.2723m	1.4145m	1.6745m	1.6745m	3.4313m	3.9136m
0.2723m	1.4145m	1.6745m	1.6745m	3.4313m	3.9136m
0.2723m	1.4145m	1.6745m	1.6745m	3.4313m	3.9136m
0.2723m	1.4145m	1.8456m	1.8456m	3.4313m	3.9136m
0.2723m	1.4145m	1.8456m	1.8456m	3.4313m	3.9136m
0.2723m	1.4145m	1.8456m	1.8456m	3.4313m	3.9136m
0.2723m	1.4145m	1.8456m	1.8456m	3.4313m	3.9136m
0.2723m	1.4145m	1.8456m	1.8456m	3.4313m	3.9136m
0.2723m	1.4145m	1.8456m	1.8456m	3.4313m	3.9136m
0.2723m	1.4145m	1.8456m	1.8456m	3.4313m	3.9136m
0.2723m	1.4145m	1.8456m	1.8456m	3.4313m	3.9136m
0.2723m	1.4145m	1.8456m	1.8456m	3.4313m	3.9136m
0.2723m	1.4145m	1.8456m	1.8456m	3.4313m	3.9136m
0.2723m	1.4145m	1.8456m	1.8456m	3.4313m	3.9136m
0.2723m	1.4145m	1.8456m	1.8456m	3.4313m	3.9136m
0.2723m	1.4145m	1.8456m	1.8456m	3.4313m	3.9136m
0.2723m	1.6244m	1.8456m	1.8456m	3.9136m	3.9136m
0.2723m	1.6244m	1.8456m	1.8456m	3.9136m	3.9136m
0.2723m	1.6244m	2.1163m	2.1163m	3.9136m	3.9136m
0.2723m	1.6244m	2.1163m	2.1163m	3.9136m	3.9136m
0.2723m	1.6244m	2.1163m	2.1163m	3.9136m	3.9136m
0.2723m	1.6244m	2.1163m	2.1163m	3.9136m	3.9136m

0.2723m	1.6244m	2.1163m	2.1163m	3.9136m	3.9136m
0.2723m	1.6244m	2.1163m	2.1163m	3.9136m	3.9136m
0.2723m	1.6244m	2.1163m	2.1163m	3.9136m	3.9136m
0.2723m	1.6244m	2.1163m	2.1163m	3.9136m	4.0059m
0.2723m	1.6244m	2.1163m	2.1163m	3.9136m	4.0059m
0.2723m	1.6745m	2.1163m	2.1163m	3.9136m	4.0059m
0.2723m	1.6745m	2.1163m	2.1163m	3.9136m	4.0059m
0.2723m	1.6745m	2.1163m	2.1163m	3.9136m	4.0059m
0.2723m	1.6745m	2.1163m	2.1163m	4.0059m	4.0059m
0.2723m	1.6745m	2.1163m	2.1163m	4.0059m	4.0059m
0.2723m	1.6745m	2.1163m	2.1163m	4.0059m	4.3835m
0.2723m	1.6745m	2.1163m	2.1163m	4.0059m	4.3835m
0.2723m	1.6745m	2.1163m	2.1163m	4.0059m	4.3835m
0.2723m	1.6745m	2.1163m	2.1163m	4.0059m	4.3835m
0.2723m	1.8456m	2.1163m	2.1163m	4.0059m	4.3835m
0.2723m	1.8456m	2.1163m	2.1163m	4.0059m	4.3835m
0.2723m	1.8456m	2.1163m	2.1163m	4.0059m	4.3835m
0.2723m	1.8456m	2.1163m	2.1163m	4.0059m	4.3835m
0.2723m	1.8456m	2.2982m	2.2982m	4.3835m	4.3835m
0.2723m	1.8456m	2.2982m	2.2982m	4.3835m	4.3835m
0.2723m	1.8456m	2.2982m	2.2982m	4.3835m	4.9900m
0.2723m	1.8456m	2.2982m	2.2982m	4.3835m	4.9900m
0.3192m	1.8456m	2.2982m	2.2982m	4.3835m	4.9900m
0.3192m	1.8456m	2.2982m	2.2982m	4.3835m	4.9900m
0.3192m	1.8456m	2.2982m	2.2982m	4.3835m	4.9900m
0.3192m	1.8456m	2.2982m	2.2982m	4.3835m	4.9900m
0.3192m	1.8456m	2.7041m	2.7041m	5.3790m	4.9900m
0.3192m	1.8456m	2.7041m	2.7041m	5.7166m	5.3790m
0.3635m	1.8456m	2.7041m	2.7041m	5.7166m	5.3790m
0.3635m	2.1163m	2.7041m	2.7041m	5.7166m	5.7166m
0.3635m	2.1163m	2.9694m	2.9694m	5.7166m	5.7166m
0.3635m	2.1163m	2.9694m	2.9694m	5.7166m	5.7166m

0.3635m	2.1163m	2.9694m	2.9694m	6.2357m	5.7166m
0.3635m	2.1163m	2.9694m	2.9694m	6.2357m	5.7166m
0.3635m	2.1163m	2.9694m	2.9694m	6.2357m	5.7166m
0.3635m	2.1163m	3.1719m	3.1719m	7.7382m	6.2357m
0.3635m	2.1163m	3.1719m	3.1719m	7.7382m	6.2357m
0.3635m	2.1163m	3.1719m	3.1719m	7.7382m	6.2357m
0.3635m	2.1163m	3.1719m	3.1719m	12.3438m	7.7382m
0.3635m	2.2982m	3.1719m	3.1719m	12.3438m	8.7609m
0.3635m	2.2982m	3.1719m	3.1719m	12.3438m	8.7609m

10.5 Preliminary Classification Model

A preliminary classification model was coded by Jonathan Bourne, UCL MSc, using the statistical tool and program language R. When the dataset of table 10.4 is grouped in four groups: [1.0m], [0.5m], [1.0m–1.5m] and [2.0m] the findings show the model to have an accuracy of 78% when predicting distances when trained with 80% of the data and tested with the remaining 20% (see code output 10.5.2). As Bourne explains: “The random forest model classifies the different groups. The data was broken out into 80% training data and 20% test data. The model achieved an accuracy of approx 80% on the test set. With the confusion matrix showing that most of the errors come from the the 0.5 and 1–1.5 groups. 0.1m and 2m being very well predicted. [SIC]” (Personal communication, May 25th 2016).

10.5.1 Code

~~~
---
title: "Sebastians Master Thesis Classification Model"
author: "Jonathan Bourne"
date: "25 May 2016"
output: html_document
---
```{r}
packages <- c("dplyr", "magrittr", "tidyr", "ggplot2", "caret")
lapply(packages, library, character.only = TRUE)
setwd("~/R/Seb thesis")(
test <-read.csv( "Range Measurement Data - Clean data.csv", skip = 2)
test %<>% select(contains("X")) %>% gather(key = distance, value = observation) %>%
  mutate(distance = gsub("X", "", distance))
ggplot(test %>% filter(distance != "2m.obstruction"), aes(x= observation, fill = distance,
colour = distance)) +
  geom_density(alpha = 0.3) +
   ggtitle("Density plot, of Observed distance distributions\n data is truncated ylimit is
set to 2.5 xlimit is set to 7") +
  ylim(0,2.5) +xlim(0,7)
top_n(test, 10, observation)
test %>% group_by(distance) %>% summarise_each(funs(mean, median))
```
```{r}
data2 <- test %>% mutate(distance= ifelse(distance == "1.5m","1m", distance )) %>%
  filter(distance != "2m.obstruction")
trainindex<- createDataPartition(data2$distance, p=0.8, list = FALSE)
~~~

```
tr <- trainControl(method = "cv", classProbs = TRUE, number = 5)
classmod <- train(make.names(distance) ~ observation, data = data2[trainindex,], method =
"rf",trControl =tr)
Classes <- predict(classmod, newdata = data2[-trainindex,],)
confusionMatrix(Classes, make.names(data2[-trainindex, 1]))
```

10.5.2 Output

```
Confusion Matrix and Statistics

Reference
Prediction X0.1m X0.5m X1m X2m
X0.1m    20     0   0   0
X0.5m     0     4   4   0
X1m       0    16  36   2
X2m       0     0   0  18

Overall Statistics

Accuracy : 0.78
95% CI : (0.6861, 0.8567)
No Information Rate : 0.4
P-Value [Acc > NIR] : 1.079e-14

Kappa : 0.6821
Mcnemar's Test P-Value : NA

Statistics by Class:

Class: X0.1m Class: X0.5m Class: X1m Class: X2m
Sensitivity                  1.0       0.2000     0.9000     0.9000
Specificity                  1.0       0.9500     0.7000     1.0000
Pos Pred Value               1.0       0.5000     0.6667     1.0000
Neg Pred Value               1.0       0.8261     0.9130     0.9756
Prevalence                   0.2       0.2000     0.4000     0.2000
Detection Rate               0.2       0.0400     0.3600     0.1800
Detection Prevalence         0.2       0.0800     0.5400     0.1800
Balanced Accuracy            1.0       0.5750     0.8000     0.9500
```

10.6 Revision History

Version	Date	Addition/change
v2.0.0	06-nov-2015	New theme: Beacons. Added intro, problem statement, scope and background info on Beacons.
v2.0.1	07-nov-2015	Expanded problem statement with narrative. Added background on NFC. Commented on security issues using beacons.
Com.	08-nov-2015	Personal communication with Camilla Hoff, Specialist Biomedical Laboratory Scientist at The Center for Psychopharmacology at Diakonhjemmet Hospital (an access controlled pharmaceutical labs) about current state of security and pathological contamination at Diakonhjemmet.
v2.1.0	08-nov-2015	Merged Scope and Expected Findings/Planned Contributions.
v2.1.1	09-nov-2015	Added privacy issues with use of beacons from Datatilsynet’s report on user privacy within media and rudimentary system model for problem statement.
v2.2.0	10-nov-2015	Added project plan and budget
v2.2.1	12-nov-2015	Focused introduction on surface contact. Added feasibility section with notes on feasibility and resources.
v2.2.2	13-nov-2015	Added some data for current strategies for limit infections. Rephrased introduction to include new data.
v2.2.3	20-nov-2015	Added sources to ethnographic method description.
Com.	26-nov-2015	Spoke with Bjørn Iversen from Folkehelsa about current state and issues with pathogen prevention.
v2.3.0	27-nov-2015	Added background on pathogen prevention in hospitals. Added practical concerns regarding access to research locations. Added notes from conversation with Bjørn Iversen.
v2.4.0	28-nov-2015	Added info from introduction to background.
v2.4.1	29-nov-2015	Rewrote into summarizing the project proposal. Restructured background literature.
v2.4.2	30-nov-2015	Removed duplicated sections. Added “the glue” aka, internet of things.
Com.	30-nov-2015	Spoke to former student at HiG pointing in the direction of Simon for automated doors and gait recognition. Spoke to Simon about gait recognition. Simon says to talk to Patrik Bours and Fuazi has done work on this for HiG.
v2.5.0	01-des-2015	Introduces Invisible UI to heading. Started writing on invisible UI. Fine tuned some language and removed some paragraphs.
Counsel	02-des-2015	It’s apparent that I should shift focus from pathogens to the interaction: What happens when you introduce new tech into an existing interaction?
v2.6.0	04-des-2015	New title focusing on access control in restricted zones.

v2.6.1	05-des-2015	Introduced some problem areas when removing user control. Referenced study on automated door control using iBeacons. Made the worst title ever… Needs work.
v2.6.2	06-des-2015	Rewrote title. Cleaned texts. Added anecdote for how loss of control influences affordance and interaction using Apple's buttonless elevators as example.
Counsel	10-des-2015	Title needs adjustment. It's too long and focuses on issues not covered by the thesis. Methods need to account for alternate approach described in risk section.
v2.6.3	10-des-2015	Adjusted title.
v2.6.4	11-des-2015	Added paragraph on backup research approach for prototyping and ethnography. Proof read. Had co-student proof read. Delivered.
v2.6.5	15-des-2015	Checked citations. Removed unused references.
v2.6.6	17-des-2015	Amended citations as per Camilla Hoff's feedback.
Com.	19-feb-2016	Ethnography at Diakonhjemmet w/ Camilla Hoff
v3.0.0	26-feb-2016	Master Thesis. Added preliminary ethnographic.
v3.1.0	19-mar-2016	Method Description: Prototyping
v3.2.0	24-mar-2016	Developed API
v3.3.0	25-mar-2016	Proof of concept using Raspberry (need different relay or using modulator) Documented API and prototype
v3.3.1	26-mar-2016	Added illustrations for experiment logic. Added illustration and details for where to conduct experiment. Added logging to API.
Coding	27-mar-2016	Studying Android SDK. Made "Hello world" app
Coding	28-mar-2016	Studying Android SDK.
Counsel	30-mar-2016	Velocity ok. Use Method to describe findings from Ethnography that govern prototype design. Describe design's limitations in separate chapter.
Coding	30-mar-2016	Connected Raspberry Pi to API using Python. I can now toggle my lights using the internet. Great success.
v3.3.2	30-mar-2016	Added illustrations for battery. Added details from first round of ethnography.
Coding	31-mar-2016	Uploaded stuff to github repo.
Coding	01-apr-2016	Got fooled.
Coding	02-apr-2016	Investigated use of PUT and DELETE through android HTTP controllers.
Coding & testing	08-apr-2016	Pair Coded proof of concept Android App with leet hacker Dan Audne Turns out Estimote beacons are not accurate enough. Turned them up to 100ms response still sluggish. Changed my phone to a more powerful from the Sony Xperia Z1 Compact to a Nexus 6. Response improved but is still not accurate enough for real time updates.
v3.3.3	11-apr-2016	Compared findings of fluctuation in beacon strength to literature.

Counsel	12-apr-2016	Agreed plan to measure fluctuation at different scan intervals and distances to get deviation documented is good approach.
Coding & testing	15-apr-2016	Coded app to position user in grid. Tested ranging within grid. Coded app to log calculated distance from one beacon at different ranges. Gathered 500 data points for further study.
v3.3.4	16-apr-2016	Added references and quotes from Don Norman on door simplicity for discussion.
v3.3.5	26-apr-2016	Proofreading introduction + theory and background.
v3.4.0	29-apr-2016	Analyzed ranging data and plotted Box Graph of results. Updated results with data. Renamed all figures. Added acknowledgements.
v3.4.1	30-apr-2016	Updated acknowledgements.
v3.5.0	13-may-2016	Moved sampling tables to appendix. Added code asserts in appendix. Adjusted results and limitations.
v3.6.0	16-may-2016	Updated figure list and corrected all figure references. Updated acknowledgements.
v3.6.1	20-mai-2016	Expanded findings with significance of standard deviation in datasets, how they distribute identically and what this might mean.
v3.6.2	21-may-2016	Elaborated on how to make a classification model based on dataset.
v3.7.0	24-may.2016	Added abstract. Changed wrong bell curve diagram. Data vas not distributed following bell curve. Added histogram of data point distribution.
v3.7.1	25-may-2016	Updated graphs to better portray key points. Adjusted abstract following colloquium comments.
Counsel	26-may-2016	Change some of the structure. Add a paragraph on future work showing the classification model. Be less absolute on claims.
v3.8.0	27-may-2016	Performed ANOVA test to prove difference/similarities between data sets. Added chapter on future work. Restructured chapter order to Discussion, Conclusion, Limitations, Future Work.
v3.8.1	28-may-2016	Proof reading.
v3.8.2	30-may-2016	Proof reading. Prototype is beacon powered, not bacon powered. Although I wish this was the case. Mmm. Bacon.
v3.9.0	31-may-2016	Final proof reading. Updated numbering. Thesis finalised. So long, and thanks for all the fish!