

INTELLIGENT LIVING ENVIRONMENT

Nanne APJ Kuperus
Ties Wertheim Salomonson
Guan-Ying Chen
Joona Kirves

ABSTRACT

Our project aimed to develop an intelligent living system that could alleviate the staff of nursing homes of the current work pressure and stress by providing psychogeriatric (PG) clients with a method of allowing them to increase their autonomy and sense of freedom whilst retaining their safety. At the moment these clients are not allowed out of the closed environment they reside in on their own and as such they have to be accompanied by members of staff or authorized family / friends if they do want to go outside.

Our system aims to provide a safe and sensible way to allow these clients to outside on their own, allow them to perform their own actions and roam free by monitoring their position and health vital parameters. The health parameters that can be monitored are various and equally important vitals such as heartbeat, blood pressure, sugar glucose levels, etcetera and serve as a way of detecting oncoming illnesses or sudden accidents as quickly as possible. This allows us to react to most situations where the client befalls an emergency, or in the cases where this is not possible (sudden falls) allows the system to react as fast as possible and alert the staff to the situation, who can then enact upon it. To demonstrate his functionality we have implemented a fall detection sensor that can detect when the client has fallen down, send an alert to the server of the nursing home to inform them this has happened and the client needs attention immediately.

To find the client in both the indoor and outdoor environments he or she may be present we have also included a location detection system: for outdoor environments the system makes use of GPS to find the client and LoRaWAN to send the location data and the vital health sensor data back to the nursing home servers. For the indoor environment we have created and employed a setup in which the position of the client can be detected by the triangulation of the Wi-Fi signal strength from any nearby signal emitters: it registers the present Wi-Fi signal strength from all emitters and compares this against profiles for every room in the building to find the room the client is currently in.

This system allows the staff of the nursing home to find the clients in both indoor and outdoor environments, monitor their health remotely and act accordingly if these parameters do not fall within the desired bounds: this typically means that the client is in need of (medical) attention and alerting the staff to this need is what this system is designed to do.

Contents

ABSTRACT.....	1
Chapter 1 – Rationale	4
Chapter 2 – Situational & Theoretical analysis	5
Tracking technology.....	5
Global Positioning System (GPS).....	5
Camera vision based tracking	6
Near field communication (NFC)	7
Inertial Navigation System (INS)	7
Ultrasound navigation system.....	8
Triangulation by wavelength (radio) signal	9
Tracking technology conclusion	10
Optical blood flow sensor	10
Accelerometer / gyroscope	11
Temperature sensor	11
Blood pressure sensor	11
Weight sensor	11
Measuring focus / attention.....	11
Measuring brain activity	12
Health parameters sensors conclusion	12
Emergency contact mode.....	12
Voice activation.....	12
Emergency button.....	13
Touch screen	13
Emergency contact mode conclusion	13
Data Transmission	13
Data transmission conclusion.....	14
Chapter 3 - Research design.....	15
Environment analysis.....	15
User analysis	15
User interaction.....	17
Failure Modes and Effects Analysis.....	18
Function schedule.....	20
Quality Function Deployment (QFD)	21
Chapter 4 - Research Results	22
Fall Detection.....	22
Location system	24

<i>Outdoor component</i>	24
<i>Data communication</i>	25
<i>Indoor component</i>	26
Chapter 5 Discussion	30
Chapter 6 - Conclusions and Recommendations.....	31
Appendices.....	33
Appendix A - Implemented Arduino code for fall detection.....	33
Appendix B - Installation instructions for Find cleint by Scholltz[6]	36

Chapter 1 – Rationale

Home automation, known as a part of intelligent living environment and a sensor-based system, is being implemented into more and more homes of older adults and people with disabilities in order to maintain their independence and safety. The application of certain technology in the nursing home can be anticipated. For this project, we were tasked with creating a system for the nursing home Hoprank in Peize, for the company Interzorg that manages this and many other nursing homes in the northern part of the Netherlands. In this research, we are going to find suitable options for sensor-based system which can limit the access to certain areas whilst providing a safe and mobile environment for the certain types of users. The patients of this nursing home are older (50+) individuals, generally diagnosed with some disability (suggested user scenario presents them with Alzheimer's). Suitable system means the elderly won't feel awkward about the limitations and the system should be easily operated and maintained by the caretaker.

The following chapters of this proposal concern our approach to the research question from different aspects. As the Intelligent Environments Conference (2007) points out: "Types of Intelligent Environments range from private to public and from fixed to mobile; The realization of Intelligent Environments requires the convergence of different disciplines: Information and Computer Science, Architecture, Material Engineering, Artificial Intelligence, Sociology and Design."

The system we would develop has the goal to prevent vulnerable patients at this facility from getting hurt, by not allowing them access to areas where they may hurt themselves and preventing them from drifting away from the facility itself. The current system in place relies on classic lock and keys technology, but a more flexible system for staff and visitors is desired to easier facilitate these groups entering the otherwise closed area(s). This proposed system does so by applying alternative sensors (including sensors from smartphones and tablets) into a flexible layer system, which is incorporated with the aforementioned classic system into a dual layer system.

We find this to be an intriguing project that not only has merit because of the proposed technological 'dual layer system' solution, but even more so because it fulfills a societal need from the nursing home and the clients. The development of suchlike systems current plays and will play a pivotal role in the upcoming years as it allows significant improvement of life for clients in nursing home and other medical facilities, due to increase in perceived freedom and ability to cope independently. For example, this technology would offer a large boost in independence to clients, such as being able to partake in daily activities whilst their safety is maintained. In cooperation with the client we hope to be able to produce a system or proof of concept which adheres closely to their stated functionality requirements, which is able to produce the level of flexibility they are looking for whilst maintaining the safety of the clients as the number one priority.

Chapter 2 – Situational & Theoretical analysis

Tracking technology

To keep watch of where the clients are located a tracking system of sorts would help to locate the clients in case of an emergency, such as when the client becomes lost or the presence of an (environmental) danger that necessitates the client being found (such as a snow storm). However, the risk of including such a system is that it can intrude upon the privacy of the clients at the nursing home or that it may add an increase in workload to the already quite significant workload the staff currently experiences.

To avoid infringing upon the privacy of the client it is important to design the system as such that the data is only accessed (or perhaps only sent) when there is a need to, e.g. when the client is in danger or is lost. Alternative, ensuring that the data is secured during transmission (such as via encryption of the data) can also prevent infringement of privacy.

For the tracking functionality itself we have the following applicable methods:

- 1) Global Positioning System (GPS)
- 2) Camera vision based tracking
 - a. Camera for normal light
 - b. Infrared camera
- 3) Near-Field Communication
- 4) Inertial Navigation System
- 5) Ultrasound positioning system
- 6) Triangulation by wavelength (radio) signal

The following chapters will provide a detailed description of all tracking methods listed above.

Global Positioning System (GPS)

The GPS system is a global (worldwide) system that allows you to find your position, or the position of any capable device via the use of satellite signals. There are a number of GPS satellites in orbit around the earth and although the GPS system itself can be divided into subsystems operated by country or countries that operate them (see table) it suffices to tell that the system works by emitting radio signals.

Name	Owner	Amount of satellites	Coverage
BeiDou	China	35	Regional (Global by 2020)
Galileo	Europe	30 (currently 15)	Global
Glionass	Russia	28	Global
GPS (NavStar)	United States of America	31	Global
Navic	India	5	Regional
QZSS	Japan	7 (currently 4)	Regional

Table 1: Global location services

These radio signals are picked up by the GPS receiver on your device (phone, satnav, car) and allow your device to calculate its location on the globe via the use of signal itself: each satellite transmits its location in orbit and the time it transmitted the signal (satellites have an synchronized atomic clock and is maintained to allot for time drift). The signal transmitted travels to your receiver over a certain amount of time (time between the transmission of data versus the time on your device) and thus your GPS receiver knows where that satellite is located. These amounts of time are minutely small: the time it takes the satellite to send its signal containing the position and time of transmission to your GPS receiver is approximately $6.73132344 \times 10^{-5}$ seconds.

Each GPS receiver is in constant view with at least 4 satellites, though typically you will be in view with 8 or 9 satellites depending on your

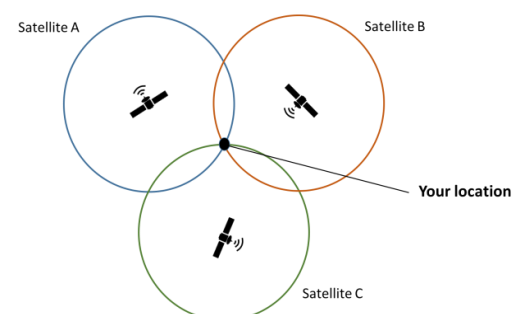


Figure 1: Triangulation by GPS

position on the globe. By receiving the signals continuously broadcast by all those satellites (with the information where the satellites were at the time of transmission) it allows the GPS receiver / device to calculate its position on the globe. It does so by taking the distances to the satellites and the positions from the satellite data to calculate by process of trilateration the position of your device. This works by plotting one circle with the distance between you and satellite 1, then plotting a circle between you and satellite 2, etcetera: you can then find your position by intersecting these circles as shown in the image. Note that there is thus no data uplink / upload to the satellites: the satellites only broadcast data to the GPS receiver on your device, they do not receive it.

For our project we could use this system by applying it for our outdoor system to find the clients location. There are quite a number of applicable small-sized GPS receivers that could be applicable to our project, whilst providing a reasonable degree of accuracy. However, the level of accuracy provided (~ 10 meter accuracy) with the error being of up to potentially 15 meters. Additionally, this way that GPS functions means that it does not do well in large built-up areas or specifically inside buildings due to the radio signal being distorted or even completely interrupted. As such, it does well outside of buildings but if the goal is to track inside as well it does not hold up.

Global Positioning System	
Advantages	Disadvantages
Free service	Does not work inside buildings
Relatively inexpensive receivers (~ €30,-)	Does not work (well) in built-up areas
Relatively good accuracy for outside use (~ 10 meters)	High error (~ 15 meters)
Simple to use via dedicated receiver circuits and software	Accuracy not suitable for indoor use
Easy data format to apply to further systems / apps	

Table 2: Global Positioning System

Camera vision based tracking

A camera system can be used to track movement when coupled with camera vision software; such software allows you to determine what to track (such as certain objects, colors, or text) and thus allows for tracking those items through the camera feed. Any sort of camera can be used with this type of software and method, as long as it produces a camera feed / footage that can be opened with a computer; typical distinctions between common types of cameras are visible light cameras ('normal' cameras), infrared cameras that (may) also record infrared lights (in addition to visible light) and thermal imaging cameras. Although the cameras differ in what they record, the software type that allows them to be used in this scenario remains the same and thus this method will be evaluated based on the software.

Camera vision software allows definition of objects via programming, which in turn allows for all manner of operations to be applied to the camera feed. For example, the image shows the use of camera vision software for finding the people present in the room. It then filters out the rest of room and shows only the outlines of the people present (hence why the outline of the individuals is white and the rest of the room is black). Any sort of object can in theory be tracked with the use of camera vision software, but the downside to this method is that it requires precise coupling between cameras to allow for the tracking of an object as soon as it leaves one cameras view and appears in another seconds later. This is made worse by the fact that for individuals with a similar appearance the camera may not be able to distinguish between them and thus becomes confused about which person is which. But by far the worst of all is that camera vision in itself is a highly resource intensive method for tracking objects, let alone multiple objects in multiple feeds at the same time.

As such camera vision may not be a feasible method for our situation as it will require multiple tracking objects over multiple different cameras and the resource requirements would be enormous, besides the fact that the tracking in between cameras (from camera 1 to camera 2) will not work smoothly. The last point of concern would be the quality of the cameras, which in turn again ties in to the resource intensive processing issue: the quality of the camera needs be quite high (1080p or higher) in order to be able to get useable camera footage at even close ranges of 5 ~ 10 meters. Any further distance warrants a higher quality camera (1440p or even 4K) but this again eats into the resource intensiveness.

Camera vision system	
Advantages	Disadvantages
Broad definable objects (can track nearly everything)	Resource intensive
Current (camera) infrastructure can be included	Problem with similar people being mixed up
Software freely available (OpenCV)	Needs good video footage quality
	Expensive due to hardware (camera + capable processing board) (~ €200,- + €50,-)

Table 3: Camera vision system

Near field communication (NFC)

Near field communication or more commonly known by the abbreviation NFC is a communication method that allows for rapid communication at close distance without need to power the (passive) transmitter as this is done by the receiver. The transmitter in the side of the NFC that holds the data, which it does by holding the data on an integrated circuits chip without need for power to hold this data; the power is transmitted to this IC Chip via the attached coil that receives power in turn from the receiver side. The receiver side is the side that does need to be powered in order to power the induction coils to be able to power the transmitter; this then creates a response in the transmitted power signal in accordance with the password stored on the chip.

The password stored on the NFC IC chip is typically not a straight forward text password but is instead encrypted or via a so-called hash. Hashing is a function that encodes the password submitted and turning it into a fixed-length (same length for every entry) string to store on a database: for example, it could turn the password 'password1' into the string '2cf24dba5fb0a30e'. There is no un-hashing, that is to say, the original password 'password1' cannot be retrieved in any way (from the database in which it is stored). Instead, any attempt to input a matching password will be checked against the hashed string to see if it matches (after all, the hash function remains the same). If the entered attempt is for example 'password2' this will become '2cf24dba5fbc0e817'; note that the last 5 characters are different due to the wrong password, thus the hashes do not match.

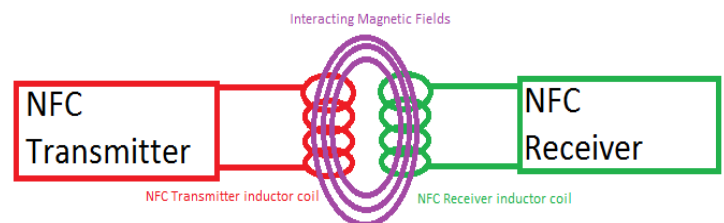


Figure 2: Near Field Communication

However, despite the excellent safety features featured in NFC the lack of range does mean that in our project the feasibility of using this method is pretty slim: although there currently is a similar system in place at the Rolde location of the Interzorg nursing homes, this system is not ideal due to the need to tap a beacon at every door that needs to be secured with this method. Additionally, the lack of range means that the system really only works in a 'closed, unless...' situation whereas the requirement is to create an open environment.

Near Field Communication	
Advantages	Disadvantages
Encryption options available (privacy)	Lack of / Limited range; only useable in close proximity
Only relies on the receiver requiring power	
Simple to implement	
Relatively inexpensive (~ €50)	

Table 4: Near Field Communication

Inertial Navigation System (INS)

An inertial navigation system is a combination of both an accelerometer and gyroscope (occasionally also includes a magnetometer) to be able to calculate (via dead reckoning) positioning, orientation and velocity. It does so without needing external influences and is widely used on a variety of vehicles such as ships, aircraft and guided weaponry. The use of both a gyroscope and accelerometers means that the system can benefit from both sensors' strengths.

The gyroscope works by having vibrating arms in the sensor that are oscillated by a crystal oscillator (piezoelectric transducer) at a fixed frequency; this happens when the sensor is (made) active and the orientation at which the object begins to vibrate is our setpoint, orientation 0 degrees. Now, when the orientation of the sensor changes, the vibration will want to retain its original vibration angle of 0 degrees; there is a sensing arm connect to the vibrating arms (via pivoting points) and this sensing arm is made out of a conductive material. Whenever the vibrating arms want to bend back to their original set point (0 degrees) these sensing arms will be bent as well, which in turn creates a potential difference (voltage difference) being either positive or negative depending on which way the sensing arms bend. This voltage potential is then transferred into a change in bearing / orientation.

The accelerometer works via having an extremely small weight etched into the silicon surface of the sensor chip, with the weight being supported by support beams that also function as the sensing arms described previously: the piezoelectric conductive material changes voltage potential depending on which way the weight bends the support arms, with these support arms being bent due to their being a force acting upon the weight. Depending on the type of sensor (which have differences in the amount and orientation of the support beams) this voltage potential can then be translated into acceleration of either 1, 2 or 3 axes, with the most common being accelerometers that can measure acceleration (forces) in all 3 dimensions (up/down, left/right, forward/backward).

Combining these sensors makes it quite a robust system, as it cannot be influenced by external influences that may throw off its aim / accuracy. However, this unfortunately also proves to be its biggest boon, as there is no external influence to tell the INS where it actually is on the map during its run: instead all navigation goes via dead reckoning, essentially the system knowing the begin point and then mapping where it is via changes in speed (acceleration) and direction (orientation). This makes this sensor severely prone to error (the so called integration drift) and with no way to correct for this, it does yield a difficult to implement system as it really needs another navigation system such as GPS to complement it to make up for this error.

We therefore do not see it as a viable system to implement in our product due to the large risk of error and the fact it needs another system to complement it, whilst we could just use such a system that is more capable on its own rather than add the INS to it. Although the system does provide quite accurate speed and direction data (based on the sensor chosen) this may be of little use to us as the speed and direction changes of the client are so low / few between that the accuracy is rather overkill.

Inertial Navigation System	
Advantages	Disadvantages
Not influenced by external factors	Large risk of error (drift)
Self-contained: does not rely on external factors after start	No position correction possible without additional navigation system
Accurate speed + direction information	
Relatively inexpensive (~ €25,-)	

Table 5: Inertial Navigation System

Ultrasound navigation system

An ultrasound system works by sending out high frequency sound and measuring the time it takes for the signal to return to the transceiver after it has bounced off of the first obstacle it encountered in front of the transceiver. The time elapsed from the signal being transmitted to the signal coming back to the transceiver is twice the distance that the sound has lapsed from the transceiver to the object: it has gone to the object and returned back to the transceiver. Thus, if we half the time it has taken for the sound to travel to the object and multiply the speed of the sound (which is a constant) by the amount of time it took the sound wave to come back, we obtain the distance that the sound has traveled to the object.

Although the ultrasound sensor thus offers a simplistic yet quite effective way to measure the distance to an object, the main problem is that the very essence of the sensor itself makes it so that it will not be quite as useful to the project. This is mainly due to the fact that the ultrasound sensor needs to be pointed at the object and that it has a rather limited field of view or cone in which it can see things (for most ultrasound sensors this would be around 30 degrees). Thus, the problem becomes that the ultrasound sensor cannot see the objects unless the product is pointing at the obstacle.

Although it certainly is possible to mount the sensor on a rotating disc or mount multiple of them in a 360-degree radius (or spherical), to try and cover around the client, the problem then becomes that the returns of the ultrasound sent out are picked up by more than only the sensor who sent it out. You are also likely to get incorrect returns from multiple signals distorting one another in this fashion. All in all, the use of ultrasound as a navigation system in this scenario would not be most feasible of ideas due to the many false and/or incorrect returns you are likely to incur. Similar issues occur when trying to use this system by lining the walls with ultrasound sensors: other problems such as cost and effective data management will then also pop up.

Ultrasound navigation system	
Advantages	Disadvantages
Encryption options available (privacy)	Lack of / Limited range; only useable in close proximity
Only relies on the receiver requiring power	
Simple to implement	
Relatively inexpensive (~ €10)	

Table 6: Ultrasound navigation system

Triangulation by wavelength (radio) signal

Another method available to track the position of the clients is the use of triangulation via a radio signal. This system works by taking a known waveform with fixed frequency / wavelength and sending this signal to a transceiver, which in turn sends the signal back to the original transceiver. If we then again take half of the time elapsed between signal sent and signal received, we can calculate the distance to the radio beacon.

If we then acquire more beacons and repeat this process, we can track down where the location of the device is: a similar process is applied for finding a phone by only using the mobile network itself and not relying on the GPS receiver in the phone itself. As is shown in the example image, by obtaining the distance to three cell phone towers (by sending a signal, waiting for the response, and calculating the distance by multiplying half the time with the speed of the signal) we can generate 3 circles, each with the radius 'distance from the cellphone to the tower'. We can then find the location of the phone by taking the spot where these 3 circles intersect: this is can be done in a matter of milliseconds as the speed of the mobile network signal is high (high wavelength) and the calculations can be done quickly.

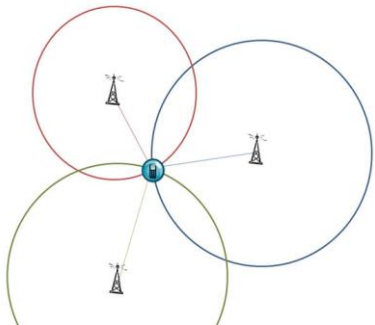


Figure 3: Triangulation by wavelength

Depending on the radio signal frequency used this method can be reasonably accurate: for the mobile network the device can generally be narrowed down to approximately 8 meters, whilst using WiFi + Bluetooth can yield an accuracy of 0.5 ~ 1 meter. It is possible to use even higher wavebands, such as UWB (Ultra-WideBand) to increase the level of accuracy even further, but this happens at the expense of range and signal penetration: higher frequencies are more susceptible to being absorbed or distorted by obstacles such as buildings (walls), trees or background radio signals such as microwave radiation.

The example described previously takes place in a 2-dimensional environment (flat plane), where the elevation is not present, but the system can be modified to include elevation data as well. For this to work more beacons need to be added, as the elevation requires at least 4 beacons to range distance to. The setup as shown here with 8 beacons allows for both finding objects inside of the area denoted by the beacons (the red dots at the outer edges) but also allows finding objects that are outside of this area, provided that they are within range of the beacons. Adding more than the required 4 beacons (adding more beacons in general) allows for an increase in positioning accuracy but does come at the cost of (slightly) increased processing time. This method would make an excellent choice for our indoor tracking system, as the accuracy and the nature of the beacon / node system would work well with an indoor environment. Coupled with the fact that the system is simple in nature and only need a receiver coupled with an accurate clock to determine the position of the device means that we feel it would be feasible and conform the technical requirements to implement this system.

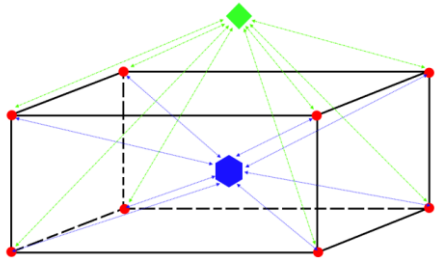


Figure 4: Triangulation in 3D space

Triangulation by wavelength	
Advantages	Disadvantages
Compatible with WiFi	Absorption by walls / objects
Can work with WiFi and BLE (Bluetooth Low Energy)	Background radio interference
Relatively accurate (0.5 ~ 1 meter)	Ultra-WideBand system expensive (€600,-)
Relatively inexpensive WiFi / BLE beacons (€120,-)	
WiFi beacons experience in Peize / Rolde	

Table 7: Triangulation by wavelength

Tracking technology conclusion

Overall, we think it best to go with a combination of the GPS system for outside use and the Triangulation by wavelength for indoor use: that way we should be assured of an accurate system for both user case scenarios and we might be able to obtain existing knowledge regarding the beacon system current in place at Peize / Rolde locations of Interzorg.Healthcare sensors

In accordance with the technical requirement of reducing workload on the staff we have given thought to the inclusion of healthcare sensors that will be able to take part of the workload off of the staff member's hands. It will ideally do so by providing the staff with health parameter information so that they will not have to go and collect this information themselves: it will already be present and ready to be used for diagnosing the client health.

As we do not want to overextend our scope, nor do we want to spend too much time on the health parameter sensors in general we have decided to keep our options open and thus not make a concrete decision about the healthcare sensors themselves. Instead, we opted to give a list of sensors to healthcare students for them to evaluate which sensors may or may not be feasible to have information for. They have told us that pretty much none of these parameters are measured daily in the normal situation, with the exception being clients that have some disease that warrants measuring a parameter. An example of this would be the client with a heart condition that has his blood pressure, blood flow and heart rate measured daily.

Additionally, the health sensors that we do implement can be used to find whether the watch is still being worn by the client it was assigned to, as well as being able to remotely assess if the client is doing alright. For example, if we detect a skin temperature (that is different than the ambient temperature) but there is no heartbeat present, we can directly alarm the emergency services to go to the client's location for resuscitation.

For the healthcare aspect sensors, we have evaluated the following parameters:

1. Optical blood flow sensor
2. Accelerometer / gyroscope
3. Temperature sensor
4. Blood pressure sensor
5. Weight sensor
6. Measuring focus / attention
7. Brain activity

Optical blood flow sensor

An optical blood flow sensor can be used to measure the heart rhythm / the blood flow in a body; it does this by sending laser light into a blood vessel and measure the frequency of light with a photodiode. Depending on the blood-flow velocity, the frequency of the light varies (so-called Doppler shift) and this shift of frequency is measured by the photodiode. The light reflects off of red blood cells in blood, which means that a greater volume of blood cells gives a stronger reflected light return. By reading the pulses in the blood (differences in blood flow over time) we can also acquire the heart rate of the client hooked up to the sensor.

We could apply this sensor to find the blood flow and heart rate information to check whether this falls into healthy values, but this information would also allow use to create an alarm system where staff or emergency services are contacted if the heart rate and/or blood flow falls too low.

Accelerometer / gyroscope

A combination of an accelerometer and gyroscope can be implemented in the product to fulfill a number of functions via the data obtained with the combined sensors. We have thought of the following possible applications for this sensor combination:

- Fall detection, can be achieved by measuring for large spikes in accelerometer data followed by no other movement data. Could be coupled to the movement system to verify that the client is no longer moving
- Bathroom detection, currently the Peize location of Interzorg is evaluating a system that allows them to monitor how often a client goes to the bathroom at night. However, the average amount of bathroom visits is different for every client and the staff has to manually compare the values for each night against the average threshold value for each patient. Implementing this into with the accelerometer and gyroscope with automated thresholding would thus reduce staff workload.
- The accelerometer might also be sensitive enough to be able to measure sleep patterns and thus present more information to the staff automatically that may be able to help them diagnose the client(s) and keep them healthy.

Temperature sensor

Including a temperature sensor would allow for the system to be able to measure the client's temperature continuously and warn staff when this temperature deviates from the average. Although the sensor measures skin temperature and can thus be easily influenced by environmental conditions, it may be able to keep track of exposure to extremely cold / hot temperature to avoid respectively undercooling and heatstroke. Additionally, the temperature sensor might be able to validate whether the client is still wearing the watch or not; if the temperature suddenly drops and the client has stopped moving for an extended period of time it would at least warrant sending a notice to the staff for them to ensure that the client is still OK, with potentially the client having lost / removed the watch and thus presenting the need to instigate a search for the client.

Blood pressure sensor

With a blood pressure sensor (either in combination with the optical blood flow sensor or standalone) we would also be able to present data to the staff that may prevent serious injuries for the clients. A prolonged period of time with high blood pressure indicates that the client is currently not well health-wise and should thus be further examined to avoid illnesses on long term.

Weight sensor

Although a consideration was made to add a weight sensor to be able to measure body weight, we were told by the experts that this practice really is not common as the body weight does not tell any relevant information about the health of the client; the only times that the weight is measured is when the client is malnourished or when the client is taking part in an experiment / trial where weight is relevant to be measured.

Measuring focus / attention

By creating a system that displays graphics and asks timed input from the clients it would be possible to measure the focus / attention span of the client, which when done with any sort of regularity (such as measuring weekly) can give you information regarding the state of the client's mind. As this would be particularly helpful for measuring the mental state of PG clients we have given this prospect some serious thought, but due to the extensive setup and research this system would require we have chosen not to pursue it lest it take up too much time and the additional fact it is well and truly out of our scope for this project.

Measuring brain activity

A similar idea to measure the brain activity was expressed, with again the intent to be able to measure the mental activity of the (PG) clients to ensure mental longevity and wellness. However, as with the measuring focus / attention the research alone would be quite severe for making such a system that involves electrodes and measuring brainwaves that we have decided not to pursue this component either.

Health parameters sensors conclusion

We will be evaluating the following sensors in close coordination with the healthcare students also tied into the project:

1. Optical blood flow sensor
2. Accelerometer / gyroscope
3. Temperature sensor
4. Blood pressure sensor

Should it become apparent that it is indeed viable to implement any or all of these sensors, we will then subsequently give thought and effort into doing so, for the added benefits to the system would be worth it from multiple perspectives, both healthcare and to add more layers to our tracking system.

Emergency contact mode

To allow our clients to contact the emergency services or staff in case of an emergency we considered 3 different methods to be implemented that allow our clients to make this contact:

1. Voice activation
2. Button
3. Touch screen

Voice activation

Voice activation is becoming more and more prevalent in all manner of systems, because it is a command method that nearly everyone can use due to the simplistic nature of telling the system what you want it to do. However, current systems are not quite there yet as they require a certain talking cadence and not all speech accents are recognized by such systems. There is also the matter of the system misunderstanding you, which leads to wrong commands being entered with potentially crucial errors.

There is the additional matter of selecting a speech recognitions service that allows for modification / implementation, as we could not find any services that are open source or available to be used locally: there is a large number of online voice assistants (Alexa, Bixby, Google Voice) but these require always on functionality via the internet which is a concern both redundancy wise and safety wise.

The largest issue we foresee with using this method of calling for help is that the client will not be able to remember how to or be able to use it in case of emergency, which is a big issue. If the client cannot learn or remember how to (these clients are fairly set in their way) use it at the time they need it most the system might as well not be included. However, some clients might instinctively yell for help if they need it, so then the system would have merit.

Voice activation of help	
Advantages	Disadvantages
Easy to use	Clients might not be able to learn how to use it
Might be used instinctively	Clients might forget how to use it at moment of need
	No local speech recognition software
	Privacy sensitive

Table 8: Voice activation of help

Emergency button

A solution that may be more up the client's alley is the implementation of an emergency button, quite similar to what they have right now. Current nursing homes and hospitals nearly all make use of the Ascom emergency button, which is a rather larger red button worn around their neck to be able to alert the staff via a button press.

The inclusion of a similar button in our product would mean that it is a. easy to find and press in case of an emergency and b. due to this system being used so widespread we can state that nearly everyone should be able to use it even. Therefore, the use of a button would have our preference, as well as due to the fact it should be a straightforward implementation provided that the communication works.

Help button	
Advantages	Disadvantages
Easy to use	Accidental presses
Used instinctively	
Simple to implement	
Inexpensive (~ €10,-)	

Table 9: Help button

Touch screen

The last item we considered to be implemented for the emergency contact mode was a touchscreen, with the idea that it could show the contact information on the touchscreen as well as the position to the client(s). That way if they simply became lost they could use the map (or directions) to find their way back to the nursing home. However, the touchscreen would be finicky in use for those not accustomed to them (which our clients are in all likelihood) and if the screen is damaged it also does not function anymore.

Help touch screen	
Advantages	Disadvantages
Can show map / directions	Tricky to use
	Relatively expensive (~ €70,-)

Table 10: Help touch screen

Emergency contact mode conclusion

We think it most suitable to use a button for the use within the project, as the clients have been accustomed to this method and know how to operate it (instinctively). It also has a good price point and it should be fairly straightforward to implement.

Data Transmission

Our last point of the concept list revolves around the various data transmission methods we could use to get the data from the product back to the main server at the Interzorg location. We have identified the following methods:

1. Bluetooth (BLE)
2. Zigbee
3. Z-Wave
4. Wifi
5. Cellular
6. NFC
7. Lora(WAN)

Rather than evaluate the complete method we have chosen to instead create a table with all the various specifications (range, frequency, data rate) listed so we are able to do a quick comparison. Going into detail on the actual method is hardly necessary for comparison as they main system is the same for all; they differ in specifications, but they all need a pair of transceivers to transmit the data between them.

Technique	Range	Frequency	Data Rate
Bluetooth	50 – 150 m (BLE)	2.4 Ghz	1 Mbps
Zigbee	10 – 100 m	2.4 Ghz	250 kbps
Z-wave	30 m	900 Mhz	9.6 / 40 / 100 kbits
Wifi	50 m	2.4 / 5 Ghz	600 Mbps max, 150 – 200 Mbps nominal
Cellular	35 – 200 km	900/1800/1900/2100Mhz	120 kbps – 10 Mbps
NFC	10 cm	13.56Mhz	100 – 420 kbps
LoRa(WAN)	2 – 5, 15 km	Various	0.3 – 50 kbps

Table 11: Data transmission

Data transmission conclusion

With these specifications in mind we have chosen to opt for the LoRa(WAN) system, as the long range allows us to cover the entirety of the village of Peize without needing a large amount of repeater transceivers. The LoRa(WAN) network will not yield the fastest communication out of all the methods listed here but it certainly adequate for our goal of transmitting location and perhaps health parameter information to the main server.

Chapter 3 - Research design

The Research Design chapter contains everything that you have done in order to answer your research question. Write and state things in such a way that somebody can repeat your research and arrive at the exact same results. You can use the chapter in your graduation definition as a start, but generally, research designs end up differently from what you intended to do in the first place. A special remark: it should be written in the PAST TENSE, since you now have already done it.

Our research was followed by the Ullman design method and we started from the product discovery phase to the product support phase. In the problem definition phase and the conceptual design phase We developed most of our research of design by using several models of different research design method to define the needs of client and create the specification for our product.

Environment analysis

In this environment analysis, we took the good relationship between the Peize village and nursing and issue of privacy into consideration, which helped us to figure out what service of system we should use.

SWOT Analysis

Strengths	Weaknesses
• Very involved employees • Good connections with family and other customer relations • Great community	• Hard to retrain employees on technical aspects • Very diverse group of customers
Opportunities	Threats
• New building which leaves room for new technologies • New technologies available	• Privacy of inhabitants

User analysis

In the user analysis, we came up with all the scenarios about the reaction when users wear the device and how the system works with all kinds of users. And with this analysis, we came up with which design was more suitable and which function do we need.

User scenario

The user scenario gives a compiled list of all the actions that the consumer or user can perform on / with the product alongside with all the handling characteristics, details and criteria that are relevant for such operations. For example, when the user can use a remote device to turn the lights on and off, handling refers to the ability of the user to hold the device which in turn details the grip size, grip comfort level and material and criteria such that the user must be able to hold the device comfortably: no one would want to hold an awkwardly shaped square remote with a texture akin to sandpaper.

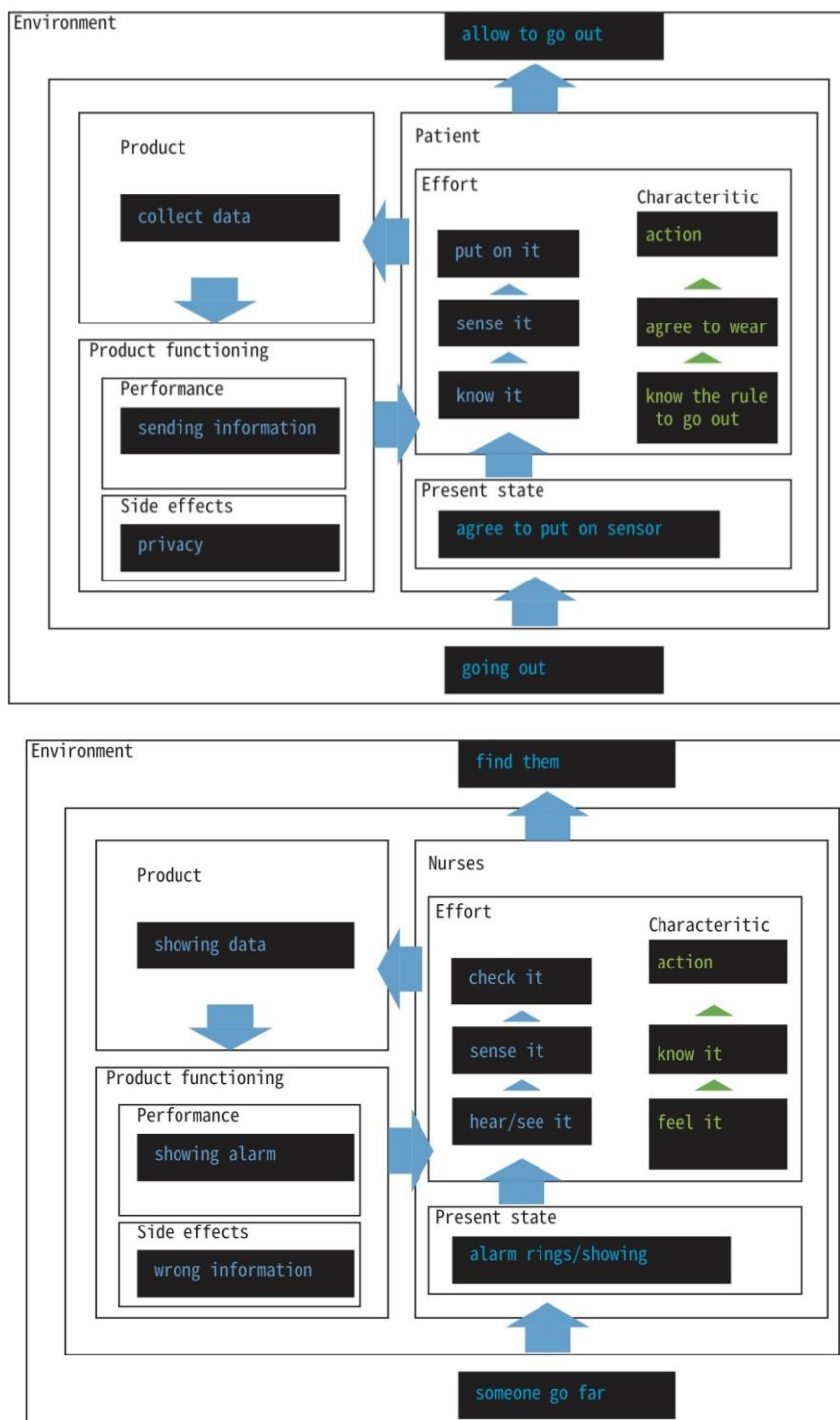
These defining details and characteristics are important for the design team to not only obtain the necessary information for the design (process) of the device, but also permits / forces them to think like the eventual end user (in scenarios) of the product and incorporate the necessary requirements.

Phase in use	Handling	Details	Further Details	Possible criteria
Install	set up system	indoor/outdoor/application	distance, phone system computer system	1. fit in certain amount of area indoor 2. certain platform
	put on chips	small, physical comfortable	always keeps on the clothes/body	1.hard to remove
	watch-setting	easy-operating	big text/big button/indicator	1.setting step less than 3 2.button bigger than 50mm 3.with voice
Daily use	sending information	automatically	accuracy	measurement error<1m
	wear watch	non-allergic material non sharp shape waterproof dustproof	agreement not to take off	1.wearing agreement 2.round shape 3.IP57/67
	read watch	clear /big text/indicator		1.text>18pt 2.with indicated light /voice
	receive information	easy to understand	immediately	1. place show on a map or 3d model 2.receiving time < 30 (sec) 3.with patient's name or code
	charge battery	easy to charge	reusable	1.usb 2.short charging time
	emergency recall	easy to put	a big button bright color	1. button size>30mm 2. red
	fix system	limit the damage	short fix time	1.backup sensor
Repair	fix watch	limit the damage	short fix time	1.backup watch
	sensor part broken	replace the broken unit		1.plug in unit
	unintended push the emergency button	push the button accidentally when moving		1.push twice to start it
Discard	keep the unbroken part			1.assembled watch

User interaction

The user interaction provides a flowchart of sorts of the way in which the user interacts with the product and more specifically in which order. Rather than the user scenario, which lists all the interactions with the product as separate actions, the user interaction diagram views these actions as interconnected and treats them as such to discover the flow that is the most convenient / efficient for the end user.

User interaction is heavily prevalent in domains such as hardware controls / controllers as well as user interfaces; these typically require a combination of inputs to produce the result(ing action) that the end user is looking for. Determining which flow works best is therefore critical to offer a good product experience.



Failure Modes and Effects Analysis

The failure modes and effects analysis (FMEA) allows the development group to identify issues and their causes early on (even in an initial stage as the development cycle) and remedy the identified issues by recommending actions, performing these actions and describing the resulting effect whether desired or not. These steps are then put into the FMEA template matrix and allow not only the pinpointing of actions that have been taking to improve the individual components as well as the whole product, but also serve as a future guide for further improvements and refinement to the product.

Function affected	Potential Failure Modes	Potential Failure Effects	Potential Causes of Failure	Recommended Actions	Taken Actions
Networking (Smart Watch)	No cell tower available / No WiFi	Loss of data	Wandering too far away / somewhere with no reception	<i>Search the area where the client is last seen</i>	<i>A search team is set up who are going to search the area</i>
Networking (Smart Watch)	No cell tower available / No WiFi	Loss of data	Wandering too far away / somewhere with no reception or a lot of room insulation	Ensure the local area has reception	Local cell tower coverage has been inspected and deemed acceptable
Networking (Smart Watch)	No cell tower available / No WiFi	Loss of data	Room insulation	Ensure the rooms in the building do not block WiFi signals	Recommend a type of insulation and inspect the WiFi signal once implemented
Networking (Smart Watch)	No cell tower available / No WiFi	Loss of data	Room insulation	Ensure that the data is stored on the Smart Watch for a while	The Smart Watch stores a certain amount of data when there is no connection to the internet
Smart Watch	Loss of power	Low / no Power	Battery was not charged	Ensure that the batteries are charged before use	The batteries charges are measured before use
Smart Watch	Wrist band breaks	Loss of some sensory data like heart rate	Long or rough usage	Ensure that the wrist bands are strong enough to hold out for a while	The wrist bands are stress tested
Smart Watch Tracking System	Networking device broken	Physical damage	No networking capabilities	Ensure that the networking devices are not broken during shipping or manufacturing	The devices are checked if they function correctly before use
Smart Watch Alarm	Button broke	Alarm cannot be activated	Component use	Ensure buttons are long lasting	The buttons are bought with a specific duration in mind
Smart Watch Alarm	No cell tower available / No WiFi	Alarm cannot be activated	Wandering too far away / somewhere with no reception or a lot of room insulation	<i>Unknown</i>	<i>Unknown</i>
GPS Tracking System	No satellites found (GPS)	Loss of data	Room insulation in buildings	<i>Unknown</i>	<i>Unknown</i>

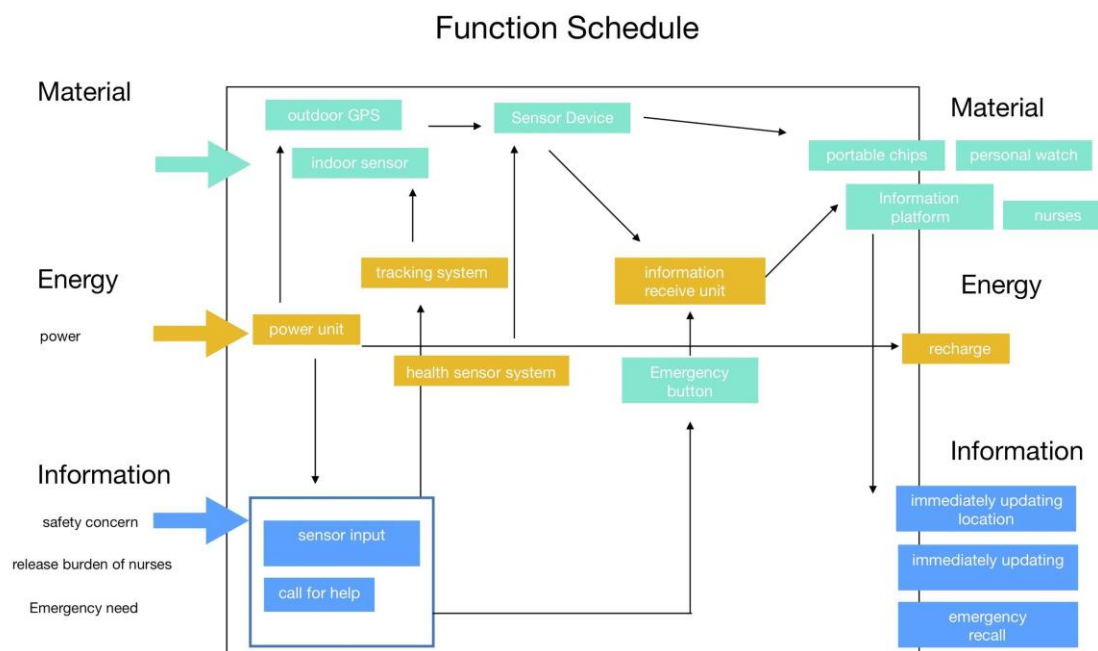
Indoor Tracking System	Loss of power	No access to data and client locations	Power outage	<i>Have a backup generator</i>	<i>The backup generator is started</i>
Indoor Tracking System / Server / Smart Watch	Software crashes	Loss of sensor data	Incorrect data / Programming errors	Create software tests before use	The software tests are run before deployment
Indoor Tracking System / Server / Smart Watch	Software crashes	Loss of sensor data	Incorrect data / Programming errors	Test if the program works correctly by testing with a prototype	A prototype is tested before it is deployed
Indoor Tracking System	Loss of network connection	Loss of sensor data	Network loses power / Network gets reset	Store data temporarily on the Indoor Tracking System	Programmatically store temporary data on the Indoor Tracking System until a network connection is restored
Server	Loss of power	Corruption of sensor data	Power outage	Make regular backups	Automated backup system

Product specifications

In this product analysis, we looked into certain devices, and establish the flowchart for every function part. Also build the specification for the product by comparing to the products in the market.

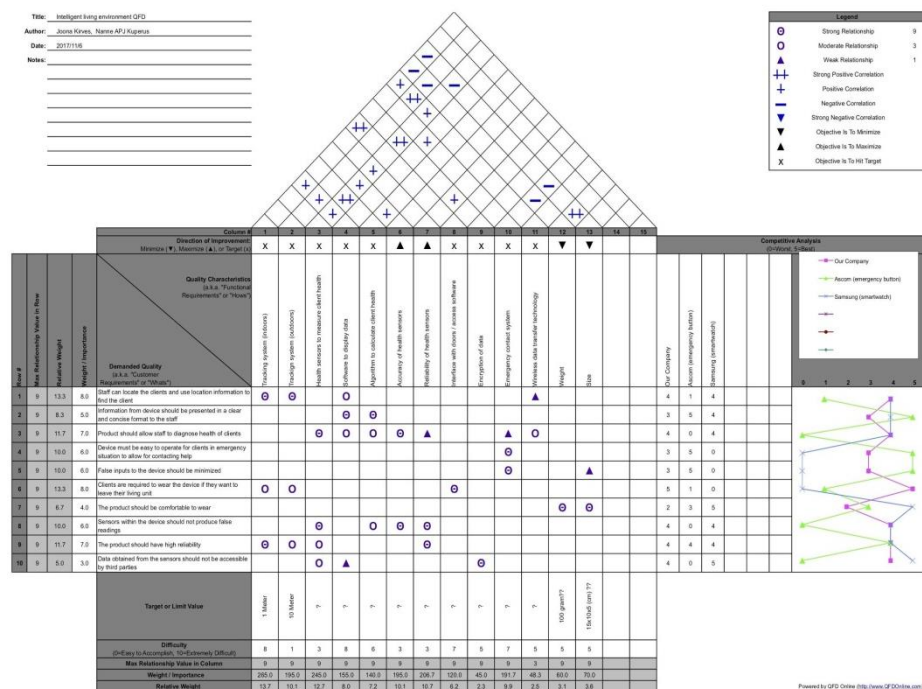
Function schedule

The function schedule details the inputs and outputs in material, energy and information for the product (manufacturing) cycle. This is incredibly relevant for trying to minimize the carbon footprint (carbon emissions required for developing and manufacturing) of the product you are developing: listing the MEI (Material Energy Information) for both in- and output allows you to examine them more closely and create alternative methods for obtaining these. Such alternatives can be items like biodegradable materials used or clean energy being generated by a solar panel attached to the product.



Quality Function Deployment (QFD)

Quality function deployment (QFD) was used to priorities the given requirements. Customers' requirements were compared to technical requirements to see which technical solutions would be the ones that we want to proceed with. Comparison was also made to already available solutions on the market to understand better what would be needed to priorities in order to have as good system as possible.



Chapter 4 - Research Results

Fall Detection

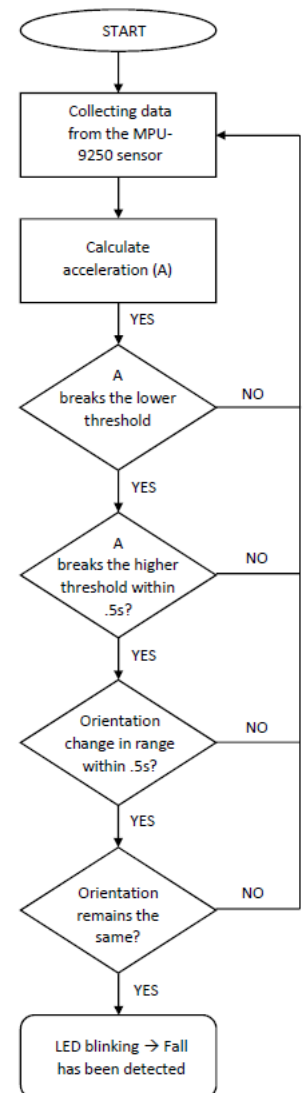
Results of the research showed us that it is not feasible to tackle the fall detection only by using accelerometer.[1] A combination of sensors with an Arduino[2] board was needed, and we had couple of options.

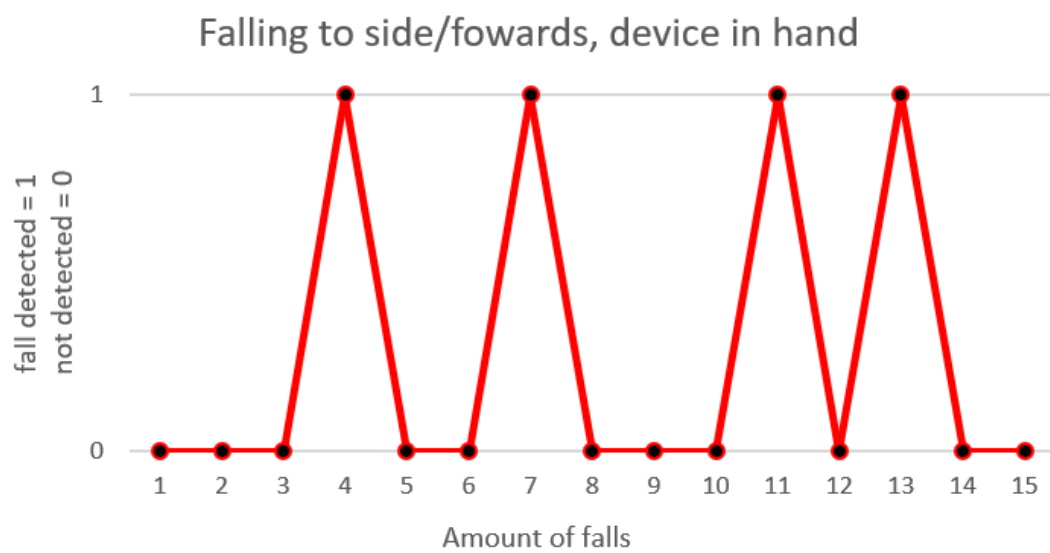
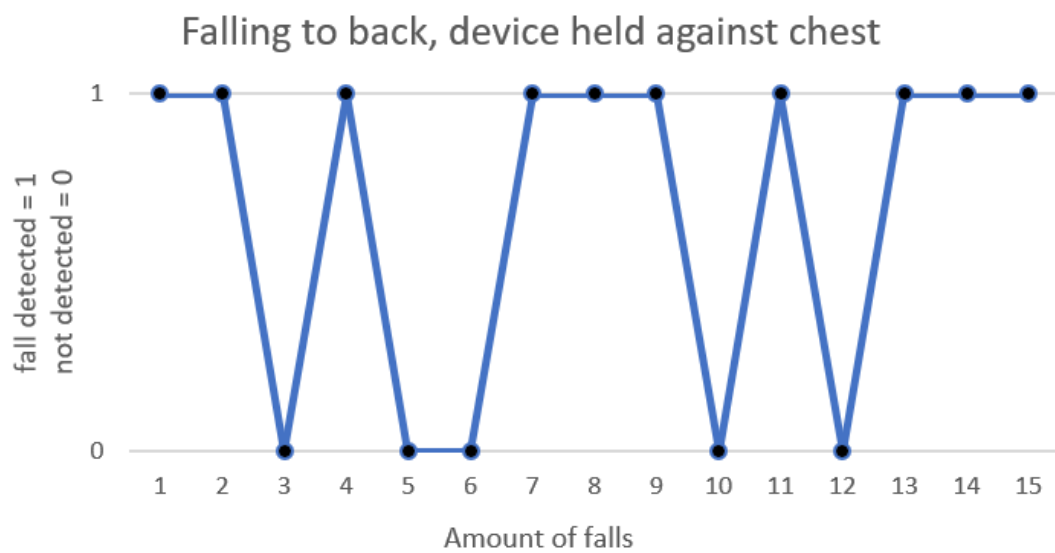
First, we had a look on sensor called GY-80 which is a five-sensor breakout board with accelerometer, gyroscope, altitude sensor, magnetometer and a temperature sensor. For some unknown reason the sensor was overheating, and we were not able to use that particular breakout board. We also had the MPU-6050 at school. That is a breakout board with two sensors in it, accelerometer and gyroscope. First tests were made with this sensor and we used it for testing the calibration code that we had. Everything worked fine but we knew that to improve the accuracy of the detection, we would want to have a third sensor on the board. Luckily, we got a breakout board called MPU-9250, which has exactly the same accelerometer and gyroscope in it with an extra magnetometer too. This sensor is currently in use in our system.

We were able to use the same code for the calibration since the two sensors were the same. We proceeded to calibrate the magnetometer which turned out to be not so simple as the accelerometer and gyroscope. A lot of time and effort went into trying to figure out the examples on internet.[3] We decided not to implement the magnetometer into the system because we were simply not experienced enough to do it and wanted not to make Nanne to do everything for us since he already had his hands full with the positioning system.

In the Arduino code we calculated the vector amplitude for the accelerometer from all the 3 axes.[4][5] This amplitude value is used to see the difference in the acceleration. When falling down, people will first experience short period of free fall, followed by a spike in the acceleration and an orientation change.[5] Two trigger values were set, higher and lower. Lower is currently set to 0,5G's and higher to 3G's. Both values need to be crossed within 0,5 seconds to keep the system detecting the fall. Orientation change is needed after crossing these two threshold values. If the orientation stays the same, the system will detect the fall and starts blinking the LED.

Testing was started by trying to replicate a real-life fall. This turned out to be harder than expected which can be seen in the results. The results were also depended on where the device was being held. We were able to trigger the detection with varying results. Best results were obtained when falling to the persons back while holding the device on chest height pressed against the body. 60% of the times the fall was detected. Holding it in the hand while falling forwards or to the persons side did not show very good results. Fall was triggered with only 26% of the times. This was due to the fact that the gyroscope did not receive big enough change in the orientation.





Location system

For the implementation of a tracking system we considered and evaluated the various different techniques we found during our research design phase; such as the various methods of global positioning systems, inertial sensing systems and direct field sensing. However, it became quickly apparent to us that neither of the systems we evaluated was capable of providing tracking with a reasonably accuracy in both indoor and outdoor environments. Although the GPS systems (global positioning systems in general, not to be confused with the currently widely employed 'NavStar' GPS system owned by the United States Government) such as Glonass, Galileo and (formerly) 'NavStar' GPS can provide some measure of tracking when presented with an indoor environment, the requirements of the project and the intended use of the system necessitated a higher level of accuracy for indoors.

Similarly, an indoor system would not be able to provide good tracking data and thus client positioning when presented with an outdoor situation; although it can typically measure more than just the indoors, the small amount of outdoor habitat being measured would not constitute enough range to be able to measure the entire village. We have therefore opted to separate system into the indoor and outdoor components to achieve the best accuracy and precision for the tracking system.

Outdoor component

The outdoor system component we have chosen to implement uses (Navstar) global positioning to determine the client location and send this position off to the database in order to monitor it / track the client. Although we considered a variety of different methods in the concept phase, none of the other methods envisioned in this phase offer the same functionality at the same level of cost-effectiveness as the (Navstar) GPS.

Although there are other Global Positioning Services available (Galileo, Glonass, Beidou) in addition to the already established and well-known (widely used) Navstar GPS service created and maintained by the United States Air Force, these services are not sufficiently advanced / available in their function yet to be considered viable for this project. Although both Galileo's (European) principles function near similarly to Navstar GPS so that the existing hardware is interoperable (and as a result need no additional hardware, but doing so may yield a higher accuracy), it is currently still in development and are slated to become fully operational in the next few years (Galileo in 2019). Beidou (Chinese) is also not available yet and whilst Glonass is already available, both of these global positioning services sadly function in just a slightly different manner than current GPS hardware and as such a different kind of hardware is required.

In this comparison we have not considered the regional positioning services (Navic, QZSS) viable for our product as these services will only permit localization within the envelope of the geostationary satellite. These satellites are in a permanent orbit above a singular point, in these cases all countries, which prohibits their use outside of these areas; with none of them being even remotely close to our operating environment they should thus not be considered in the evaluation between services.

For all of the different satellite services besides GPS and Galileo (as they are interoperable) a new type of hardware would be required for the development of products including these services. These so-called GNSS (Global Navigation Satellite Systems) receivers would be able to interface with any of the positioning services listed above (and in the concepts) and whilst integrated chips are already available with these kind of receivers, there are as of the moment of writing none available for the use with (Raspberry Pi / Arduino) development boards.

We have thus chosen to implement the already proven and widely available (in terms of hardware) Navstar GPS system as the outdoor component for our location system (as part of our product). However, just obtaining this data in the field on a local device will not yield us the desired result: we want to relay this information back to the staff of the nursing home. To that end we have evaluated a number of technologies (again, listed in the concept phase chapter) to find the best suitable method to transmit this data back from the outside environment.

Data communication

We mainly considered the use of already existing and proven technologies, mainly the use of LoRa, 3G / 4G GPRS and Bluetooth / Wifi. Either of the suggested Bluetooth / Wifi methods (grouped due to their commonality of the same radio frequency spectrum) does not have the sufficient range for our product if implemented with only 1 (centralized) base station, but due to the open nature of these technologies we could easily foresee this being made into a MESH network node-based approach, which would remove the flaw of having limited range (compared to the other options). However, the main problem with the MESH network approach is that a multitude of nodes would be required; whilst this would be fine if the number of nodes required would be less than 10, the maximum range of Bluetooth BLE (Bluetooth version made for range and low power consumption) yields approximately 150 meters (in ideal conditions).

When comparing this with the maximum range of GPRS or LoRa it becomes quickly evident that these 2 options would be (vastly) superior to the Wifi / Bluetooth options. Although the price per module unit is slightly more expensive than the Wifi / Bluetooth modules, the requirement of creating a MESH network with individual modules for the nodes dissuades us from choosing this method of approach.

This leads us to pit the GPRS and LoRa against one another to evaluate which method would be best suited for our product. General Packet Radio Service is the technology that mobile phones and other devices make use of to send text messages and transmit other information to and from another, operating on the GSM (Global System for Mobile Communications) network; aside from working with mobile phones, GPRS can also work on multiple additional devices (laptops, servers via additional PCMCIA cards) and can even support modems to create their own (sub)networks. The benefit of using GPRS would be that the infrastructure is already in place and there is no need to create our own transmit / receive station on the end of the staff / nursing home: although new infrastructure would need to be installed and routed into the server, this is proven technology and can be directly acquired from certified companies.

However, the downside to us using GPRS is that it would require every device / product to be fitted with an individual SIM card (module), as GPRS does not support networking over the infrastructure; instead it is expected to interface with your own internal network. As such, requiring the use of individual SIM cards presents with it an additional difficulty level oriented on logistics, as each individual SIM card address (phone number) must be carefully allocated to one client and that specific client only. Additionally, per our specifications from the client we have the requirement to avoid GSM as much as possible, given that the direct area around the nursing home Hoprank in Peize suffers from frequent drops in GSM signal and from a low signal reception in general.

Therefore we have opted to implement LoRa in our product, which works on a lower frequency than the other options mentioned here and in the concept list. Although this means that the signal transmitted will travel slower and thus heavily suggests abbreviating messages (transmission of clear and concise messages) it also yields the benefit of having a relatively long transmission range as advertised at 20+ kilometers. Whilst this is obviously an ideal maximum range, we expect that out of all the evaluated technologies this will offer us the most range per set of transceivers.

An additional benefit of implementing the LoRa network is that it already implements MESH networking from the ground / base layer up; although you define where the message is headed (defined as it leaves the transmitter node) the message can be automatically routed through a number of nodes or via the internet (if you have correctly setup the LoRaWAN network access for your nodes). This means that any LoRa nodes 'along the way' will get used to boost the signal and thus the range of the LoRa.

However the main components of this (sub) system still are the transmitter antenna on the client device side and the receiver antenna on the server / nursing home side, which together ferry the relevant data to the nursing home staff so that they can use the data (implemented via software that was out of our scope) to fulfill the requirements of the project (autonomy and safety).

The implementation of LoRa as mentioned before is there to transmit any data from the outside environment to the server / staff at the nursing home. This data incorporates both the GPS position for the outdoor location service as well as any health parameter sensors added to the product; please reference the appropriate part of this chapter for the developments on the fall detection sensor as an example for this.

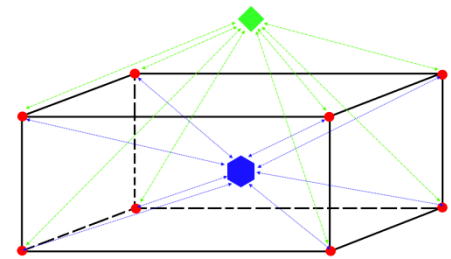
Indoor component

The last part of this system would be the indoor location system, for which we also considered a number of techniques. The requirement for which this system would be created is the need for our customer to detect their clients (in both the indoor and outdoor environment). This stems from the need to be able to provide immediate care to their clients in case of an emergency; whilst the client themselves can currently inform the nursing home staff that they are experiencing an emergency and are thus in (immediate) need of care, the system will forward this care towards their listed apartment rather than the current position of the client (as the system does not possess nor forward this information at the moment).

For this reason we have created an indoor tracking / location system that can find the position of each of the products / devices, which will be representative of the client's position. This of course requires the client to keep the device on them at all times, but if the eventual product is worked into a bracelet or belt buckle this can be easily achieved.

The major technologies we considered for the indoor location system were triangulation and signal strength, each of them evaluated with different signal based techniques such as Bluetooth / Wifi, LoRa, and NFC (other examples listed in the concept phase chapter). In our research endeavors we found that only the Wifi signal strength and triangulation methods really bear feasibility, as the other systems simply are not suited towards this sort of approach. Both the transmission time and the signal strength based approaches make use of a triangulation equation to transfer the measured quantities into a location.

Both systems work on a node basis, where the nodes transmit some sort of information that the client device can register and translate into a position. The difference between both types of system is the measured information, but the fact that applies for both systems is that a larger amount of nodes will increase the accuracy of the system. Furthermore, both systems also benefit from reduction in noise and stationary nodes to bolster accuracy.



The general intent for the transmission time based approach is to find the time difference of the signal, this being the time (in seconds) between sending the message from a transmitter and receiving the message on a receiver (in our case the client device). This type of system is based on the same principles that GPS satellites use to relay information to the navigation system in your car: roughly speaking (there is a bit more to this) the transmitted information contains the timestamp at which the signal was transmitted and the location from which the signal was transmitted. Although satellites typically transmit its location with a so called ephemeris, a precise satellite orbit dataset pertaining to the relative current position in orbit (called the 'navigation message') this is largely a stipulation due to the movement of the satellites themselves. As our system in the building will be a 'simple' 3D environment where the nodes used for localization will be stationary we are instead able to substitute this with a more simplistic XYZ Euclidian format, with X, Y and Z axes representing the distance in meters.

The time difference obtained between the node and the client device will then yield the distance between these two points by multiplying the time elapsed (in seconds) times the speed of the signal (which depends on which signal type method is used). This will then yield the $\text{time in seconds} * \text{speed in } \frac{\text{meters}}{\text{seconds}} = \text{distance in meters}$ between the two nodes. This distance is the quantifiable unit that will be used in the triangulation equation to determine the position; more on that after the explanation of the signal strength based approach (as that too applies the triangulation equation to get the position).

The signal strength approach works on a similar basis as the transmission time based approach, but rather than get a quantifiable distance to use in an equation it detects the signal strength of neighboring signals (typically in dBm). The signal strength ratio can then yield an approximation for the distance that the signal has traveled, but this typically does not yield consistent results due to the variability introduced into the measured signal strength quantities by the means of dampening obstacles, such as (reinforced) walls, (microwave) radiation and dense crowds. As such, there is a preferable method to use when applying signal strength to a localization protocol, which will be explained later.

A triangulation equation can then be applied to either method with the respective quantities to find the location of the client device based on the distance information from multiple nodes; although an increase in the amount of nodes yields a higher accuracy (as more nodes sent their information that can be translated into a more accurate position) it also means that the calculation time on the user device / product goes up. However, as this time is relatively low to begin with (~ 200 milliseconds depending on the equation itself and the amount of signals) this does not bode an immediate problem, and an easy permanent solution is to apply a cutoff limit to avoid sampling all available signals that can be used for the localization measurement.

The triangulation equation itself can be any trigonometric equation that can be successfully applied to a 3D environment, with the supplied information being the locations of the nodes in Euclidian XYZ coordinates and the correlating distance between those respective nodes and the client node. Equations as simple as the Pythagorean Theorem can be used within the 3D space (though this requires a straight corner / right triangle), but more complex equations such as Ptolemy's Theorem may also yield satisfactory results. However, as we have not yet had time to compare the effects and effectiveness of the various applicable theorems, we will instead apply a sphere intersection: each node serves as the origin for a sphere with radius of the quantifiable measurement (signal strength, signal transmission time) with the client device being located at the intersection of these spheres (as long as the amount of spheres exceeds 3 to avoid multiple valid answers).

However, both of these approaches did not yield satisfactory results when implemented as described above, as though we had tried to implement both approaches on separate occasions, they did not provide the results in an accurate and precise manner (or at least not accurate / precise enough per our standards). The signal transmission time based approach in particular was something we developed and implemented to a relatively advanced stage of completion, before we hit a major snag and changed the system type.

The main issue with the implementation of the transmission time is that the timing of all the interconnected devices (the nodes and the client devices) is that the time has to be consistent across devices down to the millisecond; if this is not so then the transmitted time stamp (on the node) and the received time stamp (on the client device) will vary more than they should and thus the returned distance between these two devices will also not match according to their real world specifics.

To try and implement the strict timing requirement we have opted to use the Raspberry Pi, which also lets us do more advanced functionality regarding the use of a network / signaling in general. The nodes used for the localization would consist of a Raspberry Pi with an additional Wi-Fi (USB) antenna installed to allow them to act as a repeater for the local area network at school, but with additional control and fine-tuning capabilities. Similarly, the client device would also consist of a Raspberry Pi in tandem with an Arduino Uno for the fall detection itself; although the fall detection could be integrated into the Raspberry Pi itself, we decided to keep them separate for the duration of the development so that we could independently work on both devices. We would then bridge them at the end of the development cycle so the Arduino Uno would send the status of the fall detection (has the client fallen?) to the Raspberry Pi who could then transfer it to the server / nursing home staff side.

However, the Raspberry Pi did not prove to be fast enough to handle the precise timing required for this approach to be successfully implemented; the test results we obtained whilst verifying the implementation and feasibility of this system led us to cancel this approach due to lack of accuracy and precision whilst using this method. Although we did consider the use of a Real Time Clock as an additional part to this system to ensure precise timing, conferring with other lecturers / experts and consulting the documentation online led us to realize even this would not net us the desired accuracy in the system and would thus not yield us the desired system (results).

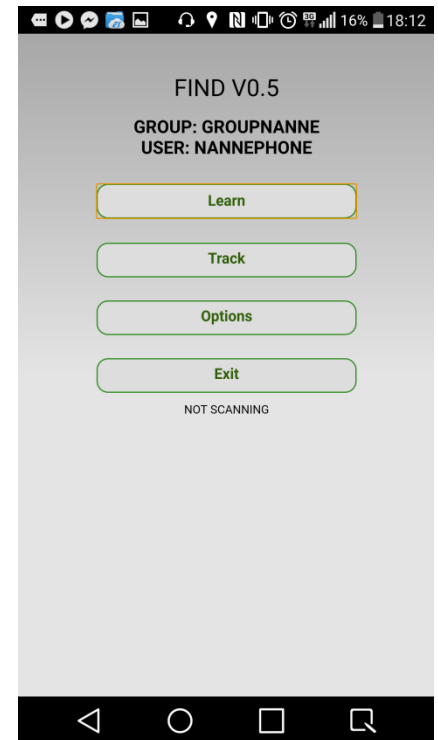
Instead we opted to use the signal strength based approach; as explained before this system measures the signal strength from various emission sources (typically routers) and converts this to a quantifiable measurement that can be used in the triangulation (equation). However, this system assumes that all emitter sources provide an equal emission of signal strength; whilst this would prove true for an ideal system with no dampening of the signal strength the nature of the system being designed for an indoor system compromises this.

The system would have to deal with the dampening force of (reinforced) walls / absorbent materials which would degrade the signal strength (though by a constant amount, as the walls and the signal emitters do not move relative to each other) and the occasional signal disruption by microwave radiation. However, as it can be expected that such a radio signal interference would only occur intermittently and that it would not affect all emitter nodes present (only the ones closest to the source of the disruption radiation) we did not consider this as a deal breaker, though definitely worth listing on the FMEA and other relevant procedure(s) (documents).

Instead, we were able to acquire the use of an open-source library that would allow us to perform these measurements and the subsequent tracking near-automatically: the software called "Find Positioning" made by Zack Schollz [6] is an open-source program that allows the measurement of signal strengths on various devices. Rather than actually measuring the signal strength itself and using this in a triangulation equation like described previously, this software approaches the location conundrum a bit differently: the software maps the signal strengths for any given room and compares the client device's current available signal strengths to this.

This requires the software to first map the room to find all the available Wi-Fi signal emitters[7] to make a map out of the available strengths, so it is recommend to walk through the room (particularly skimming the edges) to obtain the minimum and maximum signal strength values for each of the rooms you wish to be able to localize your clients in. This is done by utilizing the software to 'learn' the room; as mentioned before it maps the available signal strength and allots a range of the signal strength to the mac address of each measured signal emitter when the recording (client) device is moved through the room. This learning process can be applied fairly ingeniously; it is possible to divide (larger) rooms into smaller sub rooms by only learning the aspired space for that room (by only walking in that area during the learning process) and repeating it as desired to create the sub rooms.

The signal readings themselves are obtained multiple times per second, via effectively pinging the router / beacon on the signal it is emitting. Although this works with most routers out of the box, enterprise encryptions do not allow this functionality due to safety regulations so an alternate solution is required: more on this later in this chapter. The data storage is done on a webserver: a group can be created (and password protected) to store both the measured (calibration) data for each of the rooms and the (past and current) client positions of deices within that group. The location data for each room is stored as a set of MAC addresses of the available signal emitters for that room, coupled to the signal strength range for each MAC address. An example is shown here:



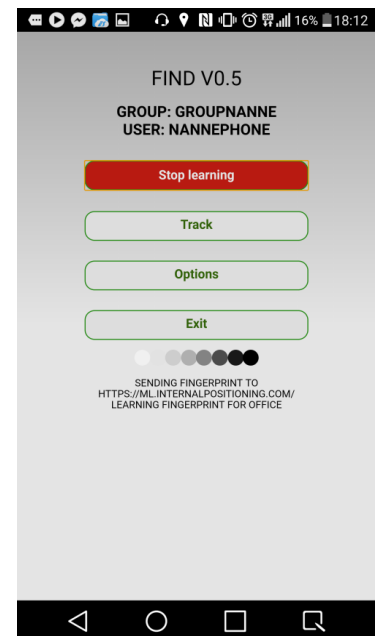
Room	Emitter number	Signal emitter MAC address	Signal strength Min (dB)	Signal strength Min (dB)
Big Lab	1	07-ED-CF-2B-34-FF	2.2	6.7
	2	A5-CA-3F-B4-11-52	2.3	4.5
	3	44-97-01-BD-75-4B	4.5	8.7
	4	DB-24-28-CD-15-19	1.1	1.9
Sun Lab	1	07-ED-CF-2B-34-FF	6.1	7.4
	2	A5-CA-3F-B4-11-52	8.9	9.0
	3	44-97-01-BD-75-4B	8.2	9.4
	n/A	Not Available	-	-
Discovery	1	07-ED-CF-2B-34-FF	5.3	8.1
	2	A5-CA-3F-B4-11-52	4.4	7.7
	3	44-97-01-BD-75-4B	7.0	9.2
	5	C0-6F-45-2D-E8-75	3.6	4.2

This data can be recorded by any type of device running the Find software[6] and thus any device can be used to setup the rooms (initially), although devices with a finer affinity for signal quality (better antenna, less noise interference) will create better results for the mapping of the signal emitters tied to that room and as a result will subsequently provide better location tracking for the clients. However, if during the employment of the system the consensus is reached that the system is performing sub optimally, additional room measurements ('learning process') can be initialized at any time and with any device to further define and refine the room with the respective signal emitters and measurements.

Although the best results for localization with any given device will be obtained by letting the device itself create a map of the emitter signals in the room, devices with similar antenna and hardware characteristics will yield results that are similar enough to provide accurate results. That being said, each room will typically have multiple mappings / readings associated with it (as this is strongly recommend to do during the setup, and does not even cost significantly extra in terms of time investment) so it is uncommon for the system to not be able to find the client at all.

It can however be expected that in some niche cases (e.g. client is in close proximity to the wall and thus rooms may conflict) the client cannot be accurately positioned in only 1 room: however in such a case the system will return all the rooms that the client is in, with percentages matching these rooms and the likelihood the client is in that respective room.

There are currently multiple example / demonstration methods in place that yield the location of the client independently by taking some algorithm and apply this to the variables in order to determine the client location in terms of a percentage that attributes the client's location fit to that room (with 0% being no fit at all and 100% being absolute fit to that particular room). Variables that are or could be applied are the signal strength of the emitters, the duration the client has spent within boundaries of those emitters (the longer in range the more likely he is in that room), proximity to other rooms, or client's affinity for that room (people are creatures of habit so they frequent the same rooms and there is also the case of clients being allocated private rooms). Depending on the variables entered and the algorithm used, the client's position may differ, but if implemented correctly (as such is the case in with current demonstration algorithms) the majority of them should yield the correct current room the client is in. The final result could then be obtained by taking these percentages of the rooms from the different methods and finding the most common room between them; this to obtain the correct room that the client is actually located in.



Several apps / clients for this software exist that allow tracking on Android, Mac, Windows and ARM devices such as Raspberry Pi (sadly no IOS clients are available). However, it is recommend to develop and host your own server solution to allow for more flexibility whilst tightly controlling access to the data for security goals; although there are example server files available (for a corrected installation tutorial please consult the appendices) they offer limited control overall. As we were not able to successfully setup our own server in such a way that we could operate it in our test environment, due to our inability to successfully parse the data from the clients to the server, we had opted to instead employ the example server software.

The benefits of this type of approach for the system compared to the signal transmission delay approach are that the room can be precisely defined through the software itself, to immediately return the position of the client to the software itself but also to any subsequent programs / frameworks wishing to make use of this data. This also allows for the creation of multiple sub rooms in rooms that typically might be only regarded as being a singular room, such as the dining area or the hallways: walls are not the limiting factor for the deployment of the rooms, as this is simply only dependent on the availability of the wireless signal emitters (Wi-Fi in our case).

The signal emitters themselves, as explained before, do not require to be connected to the internet (although it is recommend as it allows for a more easily available connection to these nodes for setup or additional refinement of the signal emitters). As the software works on signal strength basis only, nodes can be setup quickly and easily as connecting to the internet is not required for the software to work. Although it can detect most existing Wi-Fi signals, most routers in school, office and corporate environments are encrypted with WPA-PSK2 (Enterprise) that prevents the use of this software as it is regarded as some form of DDOS / proxy attack due to the repeated measurements it tries to take. As a result on such networks the system is shut out or can only take so many samples per unit of time, lowering system accuracy or even shutting it out completely.

Our type of approach also allows us to program in our own equations for determining which room the person is in (as mentioned before this is taking into account various parameters such as proximity to nodes, considering that signal strength themselves are not uniform in emission strength, time spent on / in one location and favoring certain locations) which can give us additional accuracy in the results. However, such implementation was out of the scope of this project. It also permits the export / import of the measured room / signal strength data, which permits the upgrade of the system itself (clean installing, upgrading hardware) whilst halting the need for re-establishing all the rooms in the building, as these can be simply imported again from the export data.

The downside of this system is that everyone room that requires localization in it, ergo we want to find the client(s) in that room, requires calibration beforehand to allow the system to recognize this room or the client will not be fixed to an actual current position if in that room. The stipulation is that the rooms do not have to be re-mapped for system upgrades: the current database can be exported in case of maintenance or other modifications to the current infrastructure and can be imported again once complete. However, if excessive hardware changes to the emitters (hardware) are made it should be strongly considered to remap the rooms due to the otherwise incurred penalty in accuracy due to the different characteristics of the emitters.

Chapter 5 Discussion

Overall we are quite pleased with the progress we have been able to make and the results we have been able to deliver in the end, but as with any project, there were several items and procedures in which we felt things could have been done better if we had approached them differently.

Our main topic that we felt we could have improved upon was the installation of the Find software server which although we were able to install successfully and create a local address for we could not access via the various clients we tried to access it with. The issue here was most likely that the environment in which we setup the test system / prototype was not compatible with routing the necessary data / information as desired, which led us to obtain various errors whilst trying to send the data to our own server.

Although we attempted multiple iterations of re-installing the software as well as debugging the server programming with the errors given, we were not able to successfully parse data from one client to the server or data from the server back to the client (such as the location information or the server health status). As time became an apparent issue whilst trying to resolve these errors, we instead opted to implement the less capable but working example server software provided from the creator of the software.

We would have rather developed our own version of the server software but this was not as aforementioned not a successful path: this did however mean we had to contend with a less capable version, but at the least we had a working demonstration of the software to show off during the final presentation(s) for our fellow students and of course our client.

There are also some problems with the need for every device to take their own readings to be effective: it does require a set of more than 10 readings to attain reasonable accuracy for all devices. With more specialized (more effective) signal emitters this will remove the need for every device to take signal emitter readings to be effective, but for now every device should take at least 2 or 3 readings for the device itself due to different antennas in each device. This means that currently devices do not work right out of the gate with the software installed, as the error in which room they are in can yield them a completely different room if multiple rooms are in close proximity of each other and as such have relatively similar emitter signal results.

This could potentially be solved by investing in more specialized signal emitters that allow for more accurate readings regardless of the antenna trying to take the reading of the signal emitters: alternately this could be solved by only employing a limited number of different devices, such as implementing the Raspberry Pi clients for the client devices as previously proposed.

We additionally had some issues with the development of our fall detection system, specifically with the prospects of the integration of the various sensors that together would make up the fall detection. As mentioned before, we have implemented the use of an accelerometer and a gyroscope to try and detect the user falling down, which is measured by both a rotation shift and a peak in the accelerometer data. However, we had also intended to implement a magnetometer (magnetic compass), which we currently do not use.

As explained, the reason for implementing a magnetometer is to avoid the drift of the gyroscope that becomes apparent due to the accumulating error values over time. Due to the fact that no 2 rotations are going to yield exactly the same results (for example, rotation 1 may be 89.3 degrees and rotation 2 may be -90.5) and thus the origin or 0 degree indication is going to drift over time due to the accumulative error of the gyroscope. An easy method to avoid such drift is to incorporate the usage of a magnetometer, which measures rotation independently based on the magnetic field of the earth (such as the direction of North).

This way you have two sets of rotation data which can be cross referenced and more importantly the strongest reading (save for any magnetic objects in the vicinity with a larger magnetic field) should be the magnetic North, which can be used as a reference. This way you can use this as an origin value (0 degrees) and adjust the gyroscope accordingly: it also nicely combines the flaw of the magnetometer with the gyroscope as the magnetometer is thus very accurate (can detect origin point quite well) but it is not very precise (does not measure small changes that well). We chose not to pursue this course of action as the implementation of such sensor fusion as this requires the use of a Kalman filter to blend these sensor outputs together. As not all of our team members were familiar or comfortable with this technique we decided not to implement this filter, and instead focus on accuracy through the 2 sensors we could implement relatively easily: the gyroscope and the accelerometer.

Chapter 6 - Conclusions and Recommendations

Fall Detection

Test results show that the best solution for our system would be to keep the device stationary to remove the false negatives from arm movement and allow the gyroscope to get the needed orientation change. Best solution for such would be a necklace. There is currently an emergency button system in place in the Hoprank so the fall detection device could be implemented in to the same system. Other possible solution could be a belt case.

Benefit of our system is the price. It is a cost-efficient device with an estimated price of 40€. Most of the price comes from the Arduino board itself with the sensor adding a price of 15€ to the total price.

Biggest problem currently is the accuracy of the device. A lot more testing should be done in the future to even consider this system to be in a real-life use. It should also be kept in mind that this is just a proof of concept and it was not the purpose of this project to make a market-ready product. More testing would allow to further refine the threshold values for acceleration and the orientation change need to trigger the fall. Only by implementing the use of magnetometer in to the code would make the system to rule out much more of the falls positives and false negative. Use of a 5DOF breakout board such as GY-80 would give even more accurate result if the altitude sensor was to be used.

One of the downsides of the device is also the size. Most of the size comes from the box that all the components are currently in place. It could be highly reduced by using a smaller Arduino board such as Arduino Mini or Arduino Micro. Size could also be reduced by plugging the breakout board into the Raspberry Pi that is already in use for the positioning system. In that case, a new code to detect falling should be implemented for Raspberry Pi.

Positioning service

For the positioning service we recommend that the current prototype be further developed with the integration of a privately owned server such as describe before, to allow for more advanced application of the software and the data itself. This also brings with it a tremendous security update, as although the data is currently encrypted with a password, it is accessible via the internet: as there is no need for this data to be 'publically' available as such, we recommend that the service is constricted to local operation on the network of the Hoprank (or to Interzorg locations) only. The software is more than capable of operation in such a manner and it improves the security aspect by removing the internet access to this data.

We would also recommend that the signal emitter hardware is further scrutinized before they are put into deployment, as our approach towards the selection of these emitters was focused on the ability to program them rather than cost-effectiveness. Therefore a closer look at such signal emitters may be warranted to avoid additional costs: alternatively the existing hardware may be compatible or could be repurposed to save on such expenses.

Additionally we recommend the implementation of user hardware as per our Raspberry Pi client rather than the use of the mobile phones with the respective apps as we have demonstrated and implemented at the current. The Raspberry Pi client offers the same functionality as the mobile app does, but allows for more flexibility and also negates the different results for different antennas effect we have described previously.

Overall we are quite content with the developments we have been able to make in both the fall detection and particularly the positioning service, and we hope that the developments made are of significant interest to the client and perhaps other developers to continue our research and development on this front.

References cited

- [1] Particle, "How to detect a persons fall with accelerometer?," 2014. .
- [2] M. Banz, D. Cuartielles, T. Igoe, G. Martino, and D. Mellis, "Arduino - Software." .
- [3] K. Weiner, "Simple and Effective Magnetometer Calibration," 2017.
- [4] taifur, "Emergency Fall Notifier," 2016. .
- [5] J. Ning, "Detecting Human Falls with a 3-Axis Digital Accelerometer | Analog Devices," 2009. .
- [6] Zack Schollz, "FIND software," 2017. [Online]. Available: <https://www.internalpositioning.com/>. [Accessed: 17-Feb-2018].
- [7] Phil Martin, "Using your new Raspberry Pi 3 as a WiFi access point with hostapd," *Frillip.com*, 2016. [Online]. Available: <https://frillip.com/using-your-raspberry-pi-3-as-a-wifi-access-point-with-hostapd/>. [Accessed: 08-Feb-2018].

Appendices

Appendix A - Implemented Arduino code for fall detection

```
#include<Wire.h>

const int MPU_addr=0x68; // I2C address
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
float ax=0, ay=0, az=0, gx=0, gy=0, gz=0;

boolean fall = false;
boolean trig1=false;
boolean trig2=false;
boolean trig3=false;

byte trig1count=0;
byte trig2count=0;
byte trig3count=0;
int angleChange=0;

void setup(){
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Serial.begin(9600);

  pinMode(13, OUTPUT);
  digitalWrite(13, LOW);
}
void loop(){

  mpu_read();

  //Accelerometer calibration values
  ax = (AcX+80)/16384.00;
  ay = (AcY-170)/16384.00;
  az = (AcZ+8200)/16384.00;

  //Gyroscope calibration values
  gx = (GyX+10)/131.07;
  gy = (GyY+10)/131.07;
  gz = (GyZ-18)/131.07;

  // calculating Amplitude vector for 3 axis
  float Raw_AM = pow(pow(ax,2)+pow(ay,2)+pow(az,2),0.5);
  int AM = Raw_AM * 10; // as values are within 0 to 1, I multiplied
    // it by for using if else conditions

  Serial.println(AM);
```

```

if (trig3==true){
  trig3count++;
  if (trig3count>=10){
    angleChange = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5);
    Serial.print("Angle change: ");
    Serial.println(angleChange);
    if ((angleChange>=0) && (angleChange<=10)){ //if orientation changes remains between 0-10 degrees
      fall=true; trig3=false; trig3count=0;
      Serial.print("Angle change: ");
      Serial.println(angleChange);
    }
    else{ //normal orientation
      trig3=false; trig3count=0;
      Serial.println("TRIGGER 3 DEACTIVATED");
    }
  }
}
}
if (fall==true){ //fall happens
  Serial.println("FALL DETECTED");
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  fall=false;
}
if (trig2count>=6){ //allow 0.5s for orientation change
  trig2=false; trig2count=0;
  Serial.println("TRIGGER 2 DEACTIVATED");
}
}

```

```

if (trig1count>=6){ //allow 0.5s for AM to break upper threshold
  trig1=false; trig1count=0;
  Serial.println("TRIGGER 1 DEACTIVATED");
}
if (trig2==true){
  trig2count++;
  angleChange = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5);
  Serial.print("Angle change: ");
  Serial.println(angleChange);
  if (angleChange>=30 && angleChange<=400){ //if orientation changes by between 80-100 degrees
    trig3=true; trig2=false; trig2count=0;
    Serial.print("Angle change: ");
    Serial.println(angleChange);
    Serial.println("TRIGGER 3 ACTIVATED");
  }
}
if (trig1==true){
  trig1count++;
  if (AM>=12){ //if AM breaks upper threshold
    trig2=true;
    Serial.println("TRIGGER 2 ACTIVATED");
    trig1=false; trig1count=0;
  }
}
if (AM<=3 && trig2==false){ //if AM breaks lower threshold
  trig1=true;
  Serial.println("TRIGGER 1 ACTIVATED");
}
delay(100);
}

void mpu_read(){
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr,14,true); // 14 registers
  AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
  Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
  GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
  GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
  GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
}

```

Appendix B - Installation instructions for Find client by Schollitz[6]

The installers for the clients can all be obtained from the Android App store or from the github / site of Find by Scholz, <https://www.internalpositioning.com/quickstart/>

You will then need to create a group on the server or the client and set it up accordingly with server allocation and group name + password (please also consult the server installation instructions).

Appendix C - Installation instruction for Find server by Schollitz

The installation process for the server is a bit more complicated as it requires port forwarding on most networks to work efficiently, not to consider the myriad of additional services that need to be installed for the server to operate.

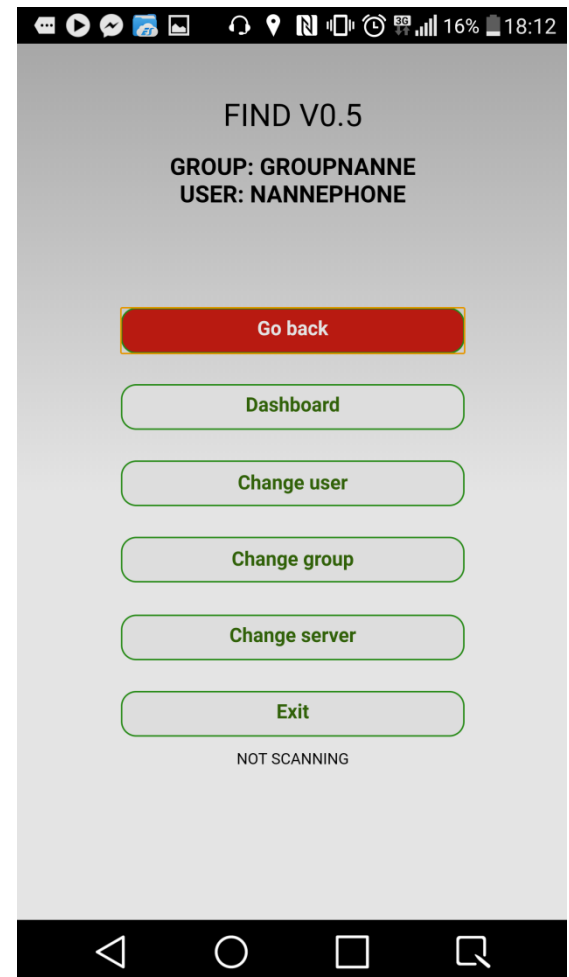
The first requirement is to install the Golang library for the Raspberry Pi, which will be the target platform of this tutorial. The Golang library can be obtained by either unpacking and installing the obtained .tar compressed file from <http://golang.org/> or by simply running the command in the terminal to install it on the Raspberry Pi: `"sudo apt-get install golang"`. Please be aware that the Raspberry Pi install command installs an older version of the Golang library (version 1.7) than the one that is available online (version 1.10), but this should not affect operability or efficiency as the server requires Golang 1.5+ to be installed and which both versions meet the requirement of.

We can then install the FIND by Scholtz server platform to get the Find server software up and running, which we do by entering the following commands into the Raspberry Pi terminal. The first command `"sudo git clone https://github.com/schollz/find.git"` obtains a clone from the latest available server version of the software on github.com and downloads it onto our Raspberry Pi. We can then use the command `"cd find"` to move to this directory and compile the software for installing by typing `"sudo go get ./..."`: the `go get ./...` command tells the golang library to prepare the find software (location `./...` specifies the bin folder where the executable files are) for installing. We can then execute the command `"sudo go build"` to build the executable and get the software ready: we can start the software when the installation is complete by running the command `"sudo ./find"`, but please note that this command has to be run in the find directory to be effective. As such, on subsequent restarts of the server platform we have to re-navigate to the appropriate folder `"cd find"` and re-run the command `"sudo ./find"` to start the server, or simply run it as a singular command as follows `"sudo find/find"` to execute.

If all is correctly installed, we should then see some nice ASCII artwork as depicted below to inform us that the software is up and running and we will be informed of the server address + software version that it is running:

```
$ ./find
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
  _ _ _ _ _
 /  /  /  /  /  \  _ _ _ _ _
 /  /  /  /  /  /  ( < - )  /  /  /  /  /  \
 /  /  /  /  /  /  /  /  /  /  /  /  /  /  /

(version 2.X) is up and running on http://192.168.1.2:8003
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```



However, please note that it will then be running on the default server provided (accessible by internet) and it is up to the client to ensure group name and password is correct, whilst the client will also not be able to access the learned rooms / areas from our system. Any learned areas are committed to the open-source group and whilst the client could thus operate their own group and localize themselves, we will not be able to do so from the server.

In order to find them from our server and localize our clients, we will need to install and create our own MQTT (mosquito) configuration for our installed server. First of all, we will need to install mosquito, which is a message broker that allows us to transmit and receive messages using a lightweight format in an IOT environment. Therefore we first install mosquito by using the command "`sudo wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key`" to obtain the install key, which works like an online repository to tell the Linux operating system and Raspberry Pi where to download the program from. We then add the key to the Linux OS by command "`sudo apt-key add mosquitto-repo.gpg.key`", navigate to the desired install location for our mosquito program "`cd /etc/apt/sources.list.d/`" and download the required files to be able to communicate with the Raspberry Pi OS and wheezy specifically: "`sudo wget http://repo.mosquitto.org/debian/mosquitto-wheezy.list`". Before we install the required files however, we ensure that we are using the correct installer for the version of the OS we are installing: both raspbian versions that are currently widely employed (Jessie and Stretch) require different versions. We then run the update command to ensure that all other frameworks and libraries are up to date ("`sudo apt-get update`") before we install mosquito and the required clients: "`sudo apt-get install mosquitto-clients mosquitto`".

That means that we can now alter the server parameters to reflect our required private settings. We do this by navigating to the find directory and creating a folder for the mosquito configuration: "`mkdir /find/mosquitto`" and then creating and editing this configuration file (if such a file does not exist editing it will create an empty file for it): "`sudo touch /find/mosquitto/conf`". We now want to start mosquito before we can alter our server settings "`mosquitto -c /path/to/find/mosquitto/conf -d`" and then we can alter the server settings by starting FIND with the desired server parameters:

```
"/find -mqtt ADDRESS:1883 -mqttadmin ADMIN -mqttadminpass ADMIN_PASS -mosquitto `pgrep mosquitto` -p :PORT ADDRESS:PORT"
```

Where ADDRESS:1883 is our IP address for the server with port 1883 for mosquito (default port)
ADMIN is our admin login name (to be generated)
ADMIN_PASS is our admin login password (to be generated)
PORT is our port for the FIND server (8003 by default unless otherwise specified)
'pgrep mosquitto' is the mosquito PID
ADDRESS:PORT is our IP address and the FIND server port (8003 by default)

This means we can now register any group made to the MQTT server to store the results and display them. We can make a group by registering: `curl -X PUT "https://ml.internalpositioning.com/mqtt?group=YOURGROUP" "` where YOURGROUP is the desired name for the group. This will then return a prompt with the message whether the attempt was successful and with a password for the group.

We can now subscribe to this group with the server so that we can see the group itself by using the command: "`mosquitto_sub -h ml.internalpositioning.com -u YOURGROUP -P YOURPASSWORD -t "YOURGROUP/location/#" "`

Where YOURGROUP is the chosen group name
YOURPASSWORD is the chosen password for the group name
YOURGROUP/location/# lets us see all the locations in this group with the clients in them sorted