

Walking Aid Reminder Device for Dementia Patients

Bangor Health Clinic Group

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Masters of Engineering



Swansea University
Prifysgol Abertawe

Faculty of Science and Engineering
Swansea University

December 12, 2021

Chapter 1

Topic

Within this chapter we will detail the problem our project aims to solve along with the limitations that could effect our project and an analysis of the current solutions our project would have to compete with. We will provide some background research on dementia patients and their issues with forgetting their walking aids when moving and how wearable devices can have a psychological impact on them. Finally, we will detail our current progress consisting of two initial meetings with our client, and our takeaways from these meetings.

1.1 Background

1.2 The Problem

Upon completion of our initial meeting with the client, we clarified the motivation behind this project and the problem that we are working together to solve. That problem is to develop a solution that detects when a dementia patient is moving without their walking aid, and reminds the patient (with a recorded message by a friend or a relative) to take their walking aid with them. Initial discussions between ourselves and the client identified current issues with dementia patients feeling uncomfortable in being forced to wear foreign objects, meaning we would need to take this into account when developing our solution. We also clarified that dementia patients get easily alarmed and frightened by generic alarms, often associating them with danger notifying alarms such as fire alarms, with the client suggesting that we facilitate a recording feature within our solution that would allow recognisable voices to the dementia patient to remind them to use their walking aid.

1.2.1 Similar Solutions

Current solutions include the use of locator systems that allow dementia patients to easily track down valuables such as keys or a wallet. Such systems include the Tile ecosystem which allows a dementia patient to attach a Tile device to their valuables and then use a smartphone app to fire an alarm from the Tile device that notifies the patient of the location of their valuable. As previously mentioned, dementia patients can get frightened and disorientated by the sounds of alarms often associating them with danger rendering these forms of solutions unsuitable for our problem. This is without considering how difficult a dementia patient mind find navigating through a smartphone device to open an application and request their Tile device to ring an alarm to help them identify the location of their valuable. Other more old fashioned systems that carers may use to notify themselves that their patient is moving include hanging items from door frames that clatter together when the patient walks through the door, or adding pressure pads under door mats that sound an alarm when the patient steps on the door mat. But what we are trying to expand upon with our solution is the protection of dementia patients that are alone and wanting to move around their home or ward. Meaning that door mat pressure pad solutions and methods for alarming a carer would be insufficient.

1.2.2 Limitations

Our main limitation for our project is that we need to develop a discrete device that will not make the dementia patient feel uncomfortable in any way. Early plans for the device lean towards a watch style device that the patients can wear on their wrist to track their movement. But if we are to create a wrist wearable device then we would need to ensure that the footprint of device is small enough to be worn on a wrist. This limits the hardware that we can feasibly use for our project. We also need to consider the number of devices being used and how they can all be fit within a wrist device. The head of the Embedded Systems module within our department has kindly offered to supply us with ESP32 based TinyPICO devices which would be suitable for this project due to its small form factor. We would need to consider how an accelerometer could be attached to the TinyPICO to allow both devices to fit within a watch casing. Other limitations for the wrist device part of the system is that it should not contain any LEDs, vibration motors or alarms to avoid startling the dementia patient. Avoiding the use of LEDs would actually be of benefit to us here as it would allow the watch device to save battery during operation. On the topic of saving battery, another limitation to the watch device is that it would need to be power efficient in order to avoid the dementia patient needing to

frequently charge the device to create a more user friendly experience for them. The ESP32 chips included on the TinyPICO boards utilise a system called 'deep sleep', which effectively powers down certain modules connected to the board. We could theoretically create a system here that fires an interrupt when the accelerometer detects movement, then forcing the ESP32 to wake up and handle the interrupt. Thus meaning that when the patient is static, the device can be in a 'deep sleep' state to save battery.

Limitations for the device being fitted to the walking aid are far less. With this device, we do not need to consider a small form factor as it is not being worn, but will be using a TinyPICO for this device too for consistency. Our limitations with this device is to also disble the use of LEDs to avoid startling th patients, and to include a speaker and microphone in order to allow a relative or carer to record a voicenote which will be played to remind the patient to take their walking aid with them when moving. The TinyPICO boards include a very minimal amount of storage space and so we may need to include an sd card to store the voice notes on. We will also be limited to the budget of £150 that we have been assigned, and must ensure that all the devices needed to build the system can be purchased within our budget.

1.3 Current Work

As stated earlier in this chapter, we have held initial meetings with the client where we have clarified the problem they aim to solve with this project and outlining the project scope. On the 18th of Novemeber we held an introductory meeting with the client where we gained an understanding of what the problem is and what kind of system the client was expecting to be produced. We clarified that we would need to gain our supply of hardware ourselves and that a budget of £150 would be allocated to us to aid with the procurement of the necessary hardware devices. However, within this initial meeting we failed to identify the final direction the client wanted the project to head down and instead came away with the option of either developing a wearable device that would detect when the dementia patient was moving, or to use a non-wearable device such as a pressure blanket that would detect when the patient had got up from where they were static. We agreed with the client that we would schedule a second meeting for the 25th of Novemeber and within that time analyse the advantages and disadvantages to each method of developing the solution. We then agreed that we would return with a solution that we thought would best suit the design brief and that would best suit the development talent available to us within our team.

Within our own intra-team meeting we decided upon building and developing a wearable

device solution over a non-wearable solution due to the extra features that could be included into a wearable device such as a fall detection system, a system recommended to be included by the client. We felt that despite a non-wearable device being a plausible route to take the project down, that factors such as a dementia patient moving off a pressure pad without actually standing up and walking would diminish the effectiveness of our solution.

On the 25th of November we hosted our second meeting with the client and established the team's preferred route for the development of this project. The client was content with this and agreed that the solution should be developed as a wearable device. We finalised the £150 budget with the client and agreed that our next steps would be to complete our milestone 1 document, including user requirements, and compiling a list of necessary hardware to develop the project. Our next meeting with the client is scheduled for December 16th where we will finalise the user requirements and compile a list of hardware to be purchased with the budget made available to us.

1.4 Aims and Goals

In order for this project to be successful, there are a few broad targets and ambitions we are looking to fulfill with our final system. First of all, the system needs to be able to effectively remind its user that they have forgotten their walking aid, and actively encourage them to use it. The goal is to prevent accidental falls, which according to our client, are a common problem with dementia patients or senior citizens, who often forget or don't want to use their walking aid. This must also easily integrate with the patient's existing lifestyle, as often they have a large amount of inertia and would prefer to avoid change. This is why we are pursuing a small wearable device, like similar solutions in the market they may have experienced before, and incorporating a familiar voice to help calm and ease them into using our system.

Chapter 2

Requirements

In this section, we will detail the project’s functional and non-functional requirements, which will be broken down to higher-level user requirements as well as lower-level specifications that will detail the process our team will go through to ensure the user requirements are met.

2.1 Functional Requirements

Table 2.1: A table of functional requirements split into user requirements and their relevant specifications needed to meet those user requirements.

Code	User Requirement	Specification
FREQ1	The wearable device should detect when a patient has walked more than 1 metre before communicating with the walking aid.	We can use a tri axial accelerometer to detect changes in acceleration that are indicative of the user moving or being mobile. Once movement is confirmed, we will then communicate with the walking aid device to ensure the user has successively reached and engaged with it, prior to alerting them to use it.

Continued on next page

2. Requirements

Code	User Requirement	Specification
FREQ2	Patients should be alerted with the voice of a friend, carer or relative to avoid startling them.	The device to be attached to the walking aid should include a microphone and speaker that will allow the user to record a voice note and store it on the device. We may need to include an SD card within this device that will store the voice note if need be.
FREQ3	The wearable device should include a solution for deaf people that still reminds them to take their walking aid with them without the need for an audio alarm.	The wearable device could use a vibration motor here that vibrates to remind the user to use their walking aid. We can also utilise the LEDs on board the TinyPICOs to flash to remind the user also. There are issues here with potentially startling the patient with the use of vibration and LEDs, however we feel this is most feasible method for meeting this user requirement.
FREQ4	If development time allows, the system should include fall detection as a stretch goal feature.	Using the tri axial accelerometer mentioned in the specification of FREQ1, we could detect acceleration and movement along the negative side of the y-axis in attempt to detect when the patient has fallen. An alert system can be used in accordance to alert a nearby carer or relative.
FREQ5	The wearable device should communicate to the walking aid device to let it know when it's started moving.	To meet this requirement we investigate the use of 433MHz Rx/Tx modules for low power and low level communication between the 2 devices in the system, this technology should allow for the basic level of communication required, with minimal power use and minimal complexity.

2.2 Non-Functional Requirements

Table 2.2: A table of non-functional requirements split into user requirements and their relevant specifications needed to meet those user requirements.

Code	User Requirement	Specification
NONFREQ1	The watch should be a small enough form factor to fit on the wrist of the patient.	Deciding to use TinyPICO devices as the main board of the device will allow us to keep the device to a small form factor given the TinyPICO is 18mm x 32mm. We will also take into account the form factor when deciding upon extra hardware to add to the devices.
NONFREQ2	The devices shall be power efficient to avoid the patient needing to charge them often.	The TinyPICO devices we will use as the main boards for the devices include an ESP32 chip capable of using deep sleep cycles. These cycles allow the ESP32 to power down non critical components in order to save power. We can create an interrupt within the code here that powers the devices on when an alarm needs to be fired due to the patient moving. This means that the devices will only need to be fully powered on when movement is detected.
NONFREQ3	The devices shall avoid startling the patients with the use of LEDs and vibrations unless they are deaf.	In this case we would power down the LEDs and Vibration motor at all times to avoid startling the patient. Powering down these devices will also allow us to save battery.
Continued on next page		

2. Requirements

Code	User Requirement	Specification
NONFREQ4	The wearable device should be discrete enough that it does not make patients uncomfortable wearing it.	We intend to design the device to make it as close the design of a watch as possible, keeping it small and sleek so that it looks like a fashion accessory rather than a medical device. Using small hardware devices we can keep a small form factor so that the device is not overly noticeable on the patient's wrist.
NONFREQ5	Security of devices should prohibit outside devices from communicating with the network.	A possibility here is using an agreed upon 'sync word' between our 2 devices that only reads communications from devices using the same 'sync word'. This would stop other devices being able to communicate with the network unless they knew the sync word being used.

Chapter 3

Specification

In this chapter, we'll go through the project in further depth. More specifically, we will outline a proposed solution as well as describing the product.

3.1 Description

The challenge of this project is to develop an effective system that interacts with users, who are dementia patients. This project will support the development of a wearable device including an application. The dementia patient or user will be encouraged to use the walking assistance in this application. A recorded message from a family member or relative will act as the notification. The device will detect if the user is sleeping and if is not, the device will only then be activated, since we do not want the device to begin notifying the user while he is sleeping, as the user might be startled by the notice.

The user does not need to log in or register with the system since the device will obtain the necessary data through an SD card, such as the familiar voice of a relative or family member. Although dementia patients may be deaf, the wearable device will support a solution for deaf individuals, such as reminding them to take their walking aid with them without utilising any audio, such as a slight vibration.

3.2 Software behaviors

The wearable device must communicate with the walking aid device in order for all of this to be happen. This communication, can be happened by using a transmitter. As soon as movement is detected, the transmitter will send message to the wearable device, letting it know that the user started walking. In this way the wearable device will be able to tell when the patient has used the walking aid and is actually walking. Otherwise, as mentioned before a message will be played to remind the patient to take their walking aid with them. The device will cease warning the user once the system detects that the user has used the walking aid and begun walking.

3. *Specification*

Furthermore, dementia all types of people at varying ages. That means that the wearable device must be able to fit on anyone's wrist. Because we do not want patients to feel uncomfortable while using the wearable device, it will be designed to look like a watch. That means that the user, who in this case is a dementia patient, will not interact with the watch, by any means of having to select an option on the device or even to change a setting. Apart from the device's appearance and dimensions, the device should be power efficient, as the patients most of the time will forget to charge it.

The device will also include a fall detection functionality. If the development time allows, we will integrate this capability into the device. A tri axial accelerometer will be used to determine the user's hand height, and more particularly the device's height. If the user's hand is extremely close to the ground, it is likely that the user has fallen. If the device detects that the user has been on the ground for an extended amount of time, it will begin informing a relative or a nearby caregiver.

Chapter 4

Methodology

Due to the nature of our team and project. We thought it best to implement an agile based development methodology. As a small team, it is easy for us to communicate with each other about the project, and easier for us to be aware of what the other members of our team are doing, key factors for choosing an agile methodology. The question then is, which agile based methodology to use? There are a variety of methodologies out there each with their own pros and cons, which we will address some of which shortly. But based on these we decided to use a scrum-based development methodology. Scrum is a combination of iterative and incremental development. Allowing for a best of both worlds approach of having early builds working, but also being agile and able to add requirements during the development process. [1]

The scrum methodology is rather simple. To effectively work, it requires collaboration between an appointed 'Scrum Master' and the rest of the team, as well as a 'Product Owner'. These members will work in close collaboration over multiple, continuous iterations of the software build to create a finished product. The role of the scrum master is to eliminate impediments [1], or in other terms, to create the conditions that allow the development team to work in their most effective manner, they take a leading role in choosing what sprints the team work on.

A typical scrum workflow consists of iterative scrum cycles, typically lasting one to three weeks each. Initially a requirements backlog is created. This is done in collaboration with the client and is the documentation that describes all the requirements that the software must meet. It is essentially a description of the product that the team are aiming to build. Once an initial backlog is created, the scrum cycles can start. As before mentioned, each scrum cycle lasts typically up to three weeks. The cycle kicks off with the team, headed by the scrum master deciding which requirements to prioritise and implement in the upcoming cycle. The aim is to have a potentially shippable product at the end of each sprint, although this may not be possible immediately. Once the team have allocated the suitable requirements to the sprint, the sprint begins, the team works on the product, meeting each day in what are known as 'scrums' to review progress made on each day. At the end of the sprint, the team should

4. Methodology

have made a shippable product. The team will also meet and hold a sprint review, along with the product owner. That demonstrates the product and how it has been developed since the last sprint. The team will also review the previous sprint, to see if any changes need to be made in anticipation of the next one. [2]

In our case, we will be slightly modifying the implementation of scrum. The above-mentioned workflow is designed as a framework for full time software developers, which we are not. As much as we may perhaps wish too, we will not be able to meet daily to discuss the project. As students it is an unrealistic expectation that we will be able to hold daily scrums, but we can still follow the methodology. We will aim to hold scrums at frequent intervals, in periods perhaps of every week, to keep tabs on the current sprints running. We can also compensate for this by being realistic in what we can achieve, and what we set as the sprint's goals.

Ultimately, we chose to use scrum for a variety of reasons. Primarily of all, we realised that as a small team, frequent communication would be easy to achieve, so we would be able to make use of this framework in the most effective manner, compared to other methodologies. We would be able to meet frequently and use that to ensure progress is made on the project. Scrum works well with this. Using scrum would enable us to work together to frequently get working products out and allow us to put ourselves in a position to implement more features. We also feel that using scrum would allow us to have a higher quality product, as the frequent iterations during sprints would allow us to streamline our implementation and constantly improve it. It also allows us to minimise risk. Take the waterfall methodology for example, where implementation begins a lot later in the lifecycle. Using scrum, where we start programming earlier, gives us more time to deal with any difficulties we encounter. It allows us to spot problems earlier, have longer to address them, or to have time to plan alternative implementations that avoid such problems.

We chose scrum over other agile methodologies for a few reasons. But mainly because we thought it was the best fit for our team and situation. Other methodologies, like extreme programming, could have worked well for us, but didn't suit our situation as students as well. For example, extreme programming requires pair programming, something that would be hard for us to do and would drastically tank our productivity. Overall we made our decision based on what would work best for us as team.

Chapter 5

Risk Analysis

Within this section, we will perform a risk analysis upon the proposed project by providing a list of possible risks that could negatively effect it. We will provide a list of these risks along with mitigation strategies that we will use to avoid the occurrence of said risks. Upon completion of the risk analysis, we will include a matrix demonstrating the likelihood of a risk occurring and the impact that risk could have on the project should it occur.

5.1 Risk Identification and Mitigation Strategies

Table 5.1: A table of risks along with strategies to mitigate those risks.

Code	Risk	Mitigation
RSK1	Our hardware devices may fail and will limit development and testing.	To mitigate this risk we will choose to use low cost but still effective hardware, which will allow for extra funds within our £150 budget should we need it to replace hardware during development. We also have the opportunity to attain TinyPICO devices from the University for this project, allowing us to minimise the effects on our budget.
Continued on next page		

5. Risk Analysis

Code	Risk	Mitigation
RSK2	The Rx/Tx modules could fail disabling communication between the wearable and walking aid devices.	The replacement of these modules should not be much of an issue due to their low cost. The real issue would arise when an Rx/Tx module fails whilst in operation for a patient. We would need to form a protocol here that detects when communication is unable to occur between the 2 devices, and can alert the patient's carer of this.
RSK3	Uploading erroneous code to our TinyPICO devices could brick the TinyPICO devices.	Should this occur, we could attempt to reflash previously working code onto the TinyPICO. If this fails, we would be left with the occurrence of risk RSK1 where we would need to replace the TinyPICO devices that have been bricked. The low cost of the TinyPICO devices should enable us to purchase some replacements if need be.
RSK4	GitHub experience a malicious attack or a server failure which could cause our repository to be lost.	Mitigating this risk is difficult. It's unlikely this will happen and that we would lose our repository as GitHub likely uses a vast backup storage solution. But, should it happen it would be catastrophic and so we should mitigate against this risk. To do this, each developer within the team will store a clone of the repository on their personal system and the team will be able to piece the code back together should this risk arise.

Continued on next page

5.1. Risk Identification and Mitigation Strategies

Code	Risk	Mitigation
RSK5	The user requirements we accept could be too large to implement within the given timeframe for the project, potentially leaving the client disappointed at project hand-over.	To mitigate against this risk, we have discussed our user requirements with the clients and feel that we have decided upon a set of features that we feel we can confidently implement within the time frame given to us. We have also some marked some features as stretch goals to ensure that the most important features are added first.
RSK6	University commitments could impact the development and testing of the product.	We have attempted to account for this within our schedule by allowing for slippage. Slippage time could allow for the development of unfinished features. We could also spread unfinished work between developers as extra work in an attempt to complete the development of features on time.
RSK7	The hardware we choose to use may lack the libraries and compatibility with other hardware for quality feature development.	To avoid this, we will select hardware that is compatible with the Arduino ecosystem to ensure that they are compatible with each other and that libraries are available for code development.
RSK8	Natural disasters could bring about the loss of hardware and software being used for the development of our product.	The use of low cost hardware in this system will allow us to replace any compromised hardware if need be. We have decided to store the team's code in a GitHub repository which will allow our code to be protected in an off site facility. The loss of developer computer systems is by far the biggest risk here, as our budget would not be able to cover the replacement of such computer systems.
Continued on next page		

5. Risk Analysis

Code	Risk	Mitigation
RSK9	Especially in the current coronavirus climate, our developers may be unwell for a period of time that has a negative impact on the development of the project.	Within our schedule we have included time for slippage that should allow for any time needed by the team to be taken off due to illness. Should a developer need to self isolate and should they not be experiencing symptoms, they could continue to work on the product from a remote location using the GitHub repository.
RSK10	An Inadequate testing strategy could allow unidentified bugs to be released in the product when handing it over to the client. This could lead to a disappointed client.	To mitigate against this risk we have devised a testing strategy that ensures thorough testing is carried out throughout the development of our product. We are confident that the proposed testing strategy will allow the team to identify and correct errors in the system before the product is released to the client.
RSK11	Poorly developed code could mean that despite substantial testing, many bugs could still be included in the product at the conclusion of the project lifecycle, leaving our product to be ineffective for the uses that the client requires.	To mitigate this risk, we will be following our testing strategy outlined in this document to allow for integration testing. This means that testing will take place every time a new feature is added to the system, allowing us to detect bugs quickly as features are implemented.
RSK12	The wearable device we create may cause patients to feel uncomfortable limiting their use of the device.	We are slightly out of control with this risk as it mainly depends on how the patient reacts to the wearable. Having said this, we aim to make the wearable as discrete and as watch like as possible in an attempt to avoid the patient feeling uncomfortable when wearing it.

Continued on next page

Code	Risk	Mitigation
RSK13	Our device could startle or scare the patient putting them in danger.	We have taken on board the advice of the client for this risk and will be ensuring the watch does not use vibrations or LEDs unless the patient is deaf. We will ensure that generic alarms are not used also in an attempt to avoid startling the patient.
RSK14	One of our developers could leave the institution, lowering the number of developers we have available to work on the project.	Should this occur, we would definitely need to use the slippage time we have allowed for in our schedule. We would not be able to bring another developer into the team and would therefore need to share the workload of the developer that is leaving to the other developers on the team.

5.2 Risk Likelihood and Impact Matrix

Table 5.2: A matrix detailing the likelihood of a risk occurring along with the relative impact caused by that risk occurring.

Likelihood \ Impact	Low	Medium	High
Low		RSK2, RSK4, RSK7	RSK8, RSK13, RSK14
Medium	RSK12	RSK3	RSK1, RSK5, RSK10, RSK11
High		RSK9	RSK6

Bibliography

- [1] A. Srivastava, S. Bhardwaj, and S. Saraswat, “Scrum model for agile methodology,” *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 05 2017.
- [2] E. d. Nuevo, M. Piattini, and F. J. Pino, “Scrum-based methodology for distributed software development,” *2011 IEEE Sixth International Conference on Global Software Engineering*, 08 2011.