Sheffield Hallam University

Faculty of Science, Technology and Arts

Department of Computing

Project (Technical Computing)

[55-604708]

2019/20

Author:	**Kenneth Alegria**
Student ID:	**26035314**
Year Submitted:	**2020**
Supervisor:	**Josh Bamforth**
Second Marker:	
Degree Course:	**BEng (Hons) Software Engineering**
Title of Project:	**Robotic assistance for patient interaction in Minor Injury Unit (MIU) assessment rooms**

Confidentiality Required?

NO ☒

YES ☐

Abstract

A major problem the NHS is facing now is staff shortages as there are not enough nurses to facilitate the number of patients and cases they face on a day to day basis. Staff in the NHS are stressed and overworked which reduces their effectiveness at work and is paramount that they work effectively because of the nature of their work. Staff shortages also means that patients would have to wait longer in the A&E to be assessed resulting in unhelpful and annoyed patients which adds to the stress of staff that needs to deal with them.

This report will explore the idea of a social robot that would be used for patient interactions in the assessment rooms of the A&E in UK hospitals using research on Human-Robot Interactions (HRI) to help with alleviating the workload and stress of staff within the NHS. I will research real life uses of the Nao and Pepper robot to identify features that are desirable for a robot working in the assessment room with a patient, and to see how well a robot can communicate and socialise with humans in an acceptable way. The robot deliverable will aim to be able to do a simple diagnosis and recommend a treatment to patients, similar to the normal procedures of a nurse, by being able to ask and listen to the patient's symptoms and figure out which is the most likely underlying cause. A secondary objective is the ability for the deliverable to be able to read QR codes on a wristband that would replace the normal wristband given to patients to allow the robot to quickly attain information about the patient instantly once the prototype wristband is presented such as name and date of birth, to help streamline the diagnosis procedure.

The first half of the report will cover the research of why there would be a need for a robot in the NHS, identify the current capabilities of Nao and Pepper, and to investigate ways to improve their social capabilities to be more suited in a medical environment. The second half of the paper will focus on applying the findings of the research phase to the development of a robot that is capable of patient interaction, with a final discussion on the challenges and problems encountered in both the research and development phase, as well as to conclude the findings of the paper with focus on the future of robotic use with patient interaction as a priority.

1. Contents

Introduction

Motivation

Right now, there is a major problem with Britain's NHS and its effectiveness and efficiency. Staff shortage is one of the major concerns for the NHS currently and some of the reasons as to why are because of high staff turnover, lack of funding and undergraduate nursing students (Buchan, Charlesworth, Gershlick, & Seccombe, 2019). In an article about nursing staff In December 2018, the Chief Executive of NHS Improvement anticipated that balance between the supply of, and demand for, NHS medical staff will be achieved in the next 5 years, but that there will be a continuations of shortfalls in NHS nursing staff (Ford, 2018). With Brexit happening soon, the NHS situation will get worse before it gets better as over 1000 nurses from the European Economic Area (EEA) has left the register between April and September 2019 while only 968 nurses from EU and EEA countries applied in 2019, showing a small but clear deficit of nurses coming from EU, according to the annual report from the Nursing and Midwifery Council (2019).

On the other hand, the field of robotics has grown with the development of self-driving cars and Sophia – A robot capable of holding a conversation, through the advances in AI, visual recognition,

and visual data processing. Self-driving cars are able to recognise its surroundings using cameras and use its sophisticated AI to make the correct and safest decision in a split second whereas Sophia also uses AI such as facial and emotion recognition coupled with robotic movements generated using deep neural networks.

Aims and final deliverable

This report will look into using technologies used in modern robots – such as the capabilities of Nao and Pepper, and applying desirable functionality to a tailor made robot with the intent of being used by the NHS to help with patient symptom diagnosis to help alleviate the staff shortages of the NHS in the upcoming future. This report will focus on the social interactions between robots and humans and how different mannerisms such as limb movement, looks, speech and conversation capabilities as the final deliverable will be patient focused. The aims of the project are:

- Research the need for robots within the medical field
    - Find out where robots would be most helpful in the medical field
- Research real-life applications of the Nao and Pepper robots that share a similar scope with our aims.
    - Identify and extract the desirable features of the researched robots.
- Research further into what QR codes are and how they work
    - Explore the practicalities of using QR codes in the medical field
- Explore the new technology, environment, and language needed to complete a successful deliverable
    - Gain a deeper knowledge of the Python programming language
    - Gain experience with using the robot IDE (Choregraphe)
    - Use an appropriate software engineering methodology for development
- Successfully implement the desirable functionality from the research section, into our final deliverable
- Implement a QR Code reader functionality as a second deliverable
- To critically evaluate the final deliverable, the development process, and an individual critical evaluation.
    - Critical evaluation of the report
    - Testing of the deliverable
    - Critical analysis of individual performance

Research

To get a better outlook for what we would want for the robot in the project, research will be conducted on previously developed designs, concepts, and additional insight on the need for medical staff in the NHS. By doing this, we can correctly identify the requirements of the NHS in terms of staff shortages and to identify whether features on previously developed designs would be a good fit for the project aims, and if not, to adjust accordingly to the requirements.

NHS and staff shortages

A big concern within the NHS is its shortages of staff throughout the years, primarily because of "high staff turnover, lack of funding and undergraduate nursing students" (Buchan et al, 2019).The paper concluded that "the gap between the demand for and supply of NHS staff is not getting any better. It is, in fact, worsening for some key staff groups and service areas, notably learning disabilities and primary care.", showing that staffing in the NHS is not going to get any better and is therefore in a need of a change in either a push towards recruitment, or to use technology such as robots to help with the lack of manpower in hospitals across the UK. The situation with Brexit also puts the NHS in a precarious situation as it would lower the recruitment pool available for the NHS in the future as it would be harder for nurses outside the UK to apply, as an article (Hervey & McCloskey, 2018) claims that "20,000 NHS England nurses and around 100,000 social care staff" are from the EU, further showing "the sheer scale of reliance on EU migrant workers".

I conducted further research as to why there is a high staff turnover in the NHS and found that there is a high level of emotional exhaustion and job dissatisfaction within the nursing ranks. The results in Table 6 (Sheward et al, 2004) shows that the nurse to patient ratio was "1:9.4" for Scottish hospitals and "1:10.0" in English hospitals that participated in the survey. This further leads to Figure 2 and Figure 3 where it shows that while the mean stress level for both Scottish and English nurses is "20.8%" and "23.3%" respectively, "analysis of the individual nurses' data showed that 27% of Scottish nurses and 34% of English nurses had high scores" (Sheward et al, 2004). and in Sheward's follow up paper (2005), "Over the last few years however increasing concern has also been expressed about a growing nurse shortage in the UK [...] leading to increasing publicity about hospital understaffing." Moreover, Sheward et al that "Nurses […] report that their staffing levels are inadequate to provide safe and effective care and complain of ever increasing workloads […]" (p52).

With the findings, we can conclude that there is a serious need to alleviate workload and stress for nursing staff within the NHS as there is a low nurse to patient ratio, with staff being outnumbered 1:10. We can deduce that one of the reasons as to why there is a long wait time for patients in the

A&E is that there aren't enough staff to look after patients due to being needed in more important matters. By finding the reason why there is a long wait time for patients in A&E, we have identified an environment where a robot can be used in lieu of nurses. This would allow nurses to work on more urgent cases. However, to work fully autonomously, without human guidance, is out of this project's scope. Instead, this project aims to lessen the workload on healthcare professionals.

How can robots be useful in the real world?

There are many uses for robots in the real world, ranging from industry to recreational use, and there are emerging robotic technologies every year which cover different fields. On a blog (Dvorkin & Bharadwaj, 2019) about which industry uses the most robots, the automotive industry employs about 42% of all robots in the world. For example, robots are used in assembly lines to do monotonous and strenuous tasks that would not be fitting for a human to do in the long term.

Robots have been tasked with filling the role of jobs that are monotonous, dangerous, physically stressful or requiring accuracy and precision, and while 42% of all robots are in the automotive industry, robotics branches out to the medical, scientific and gastronomical field to name a few, where at times robots are more useful to have than human hands. For example, specialised doctors would use the Da Vinci surgical system for "Minimally invasive surgery [...] widely used in urology and gynecology procedures, and several other procedures too" as invasive surgery would require incredibly steady hands, so the best course of action would to use the surgical system for assistance in such sensitive scenarios.

Currently, there are robots used in the medical field such as the aforementioned Da Vinci surgical system in the operating theatre of hospitals, to robots such is MEDi (Medicine and Engineering Designing Intelligence), tasked with making doctor visits less terrifying for kids waiting in the waiting room, accompany them on the way to the doctor's office, and to keep them distracted during such sessions.

Pepper and Nao

Due to limitations on what robots can be used to create a suitable deliverable, the robots that will be used will be robots created by Softbank Robotics - Pepper and Nao. Pepper and Nao's purpose is, according to the manufacturers website Softbank Robotics, 2020a), for educational use as there are "Our 17 000+ Pepper and NAO in the global education market are enabling new ways for pedagogy in classrooms.", however there are other designs that use Pepper or Nao to achieve other things such as a Nao variant called MEDi the robot and artificial receptionist variants of Pepper.

Figure 1. Image of Pepper (Left) and Nao (Right). Reprinted from *Paris city hall gives jobs to Pepper and Nao robots* from *Robotics and automation news* by S. Francis, 2017, Retrieved April 20, 2020, from *https://roboticsandautomationnews.com/2017/09/12/paris-authorities-employ-pepper-and-nao-robots-at-city-hall/14052/*

Pepper and Nao are used in similar ways when being used for educational purposes with the main difference is that Nao is used to target a younger audience such as kids in primary school, whereas Pepper would be used for higher education such as Universities and the reason for this is that Nao is a more friendly-looking robot that children can relate to, play and have fun while Pepper, being the bigger sister of the two robots, is used for research and is used for more advanced functionality, as well as being used for special cases such as education for children who are in the autism scale (Kamps, 2016).

Apart from educational and research purposes, Softbank Robotics also acknowledges the practical uses of Nao and Pepper in the healthcare sector (Softbank robotics, 2020b). The company claims that by using their robots, it can "Enhance the efficiency of administrative processes, improve quality and consistency of patient experience", "Improve awareness of prevention care, reduce anxiety on disease & treatment, and enhance stakeholder relationship" and most importantly "Generate real-world evidence to analysis patient health history for a better diagnosis and standard of care", which aligns with the aims of the project - A robot that can diagnose patients. This is great since Pepper and Nao is what will be used due to limitations in what can be used in the development.

Pepper

While Pepper's main purpose, as suggested by their manufacturer, is education, Pepper is also used for different purposes such as commercial and recreational use. There are several different variations that use the Pepper and Nao robot that makes use of the social capabilities which can be modified by using the dedicated IDE Choregraphe to implement such changes.

An example of a bespoke commercial use is Pepper the receptionist (Robotics of London, n.d). This variant of Pepper can welcome visitors once eye contact is made, arrange drinks for visitors and is able to chat freely and autonomously by using a database of over 90,000 different topics amongst other things. Other examples of commercial uses that uses the Pepper robot can be seen in Munich Airport and in parlour cafes in Japan.

Nao

Nao is a smaller version of Pepper and without the screen-like tablet on its chest and is used on a smaller scale than Pepper. Nao is primarily used for primary level education that helps kids learn in a fun and interactive way. It takes advantage of the small stature and cute disposition to connect with children to enhance their learning children, however it is not its only use.

An example of a variant of Nao is MEDi, a robot specialised in interacting with children. The main purpose of MEDi is to ease children's anxieties and stress around medical procedures. MEDi would greet children as they come for a painful procedure, explain the procedure to the child and then MEDi would continue to distract the kid during the procedure through interactive dialogue, allowing MEDi to tell stories, to dance and play games.

MEDi was developed in the University of Calgary (Ockey, 2015), with the purpose of researching how to increase the vaccination rates of children as "One of the reasons children don't get vaccinated is due to fear and anxiety associated with the pain involved in the procedure". The research conducted found that MEDi was "shown to decrease pain associated with vaccinations by 50 per cent [sic] while increasing vaccination rates by 10 per cent[sic].". This shows that even if the robot does not do much in terms of helping a patient out medically, it can help decrease the fear and anxiety of patients when entering a hospital by being more sociable, friendly and interactive. However, the downside to this is that the research only included young patients therefore the effects of such a robot will be unknown when applied to all ranges of patients.

On the other hand, the use of robots seems to have a negative impact on the perception of the older generation (Learner, 2019). Robots such as Nao is planned to be widely used in care homes in the UK and that "It isn't about replacing jobs, it is about complementing existing care" and the CARESSES project was created to "assist older people and adapt to the culture of the individual they are taking care of.". This is because of technophobia - the fear of technology, which is more prevalent in the older generation as they are less experienced and acquainted in using technology compared to younger generations but in Japan it is a very different story ("Robots to transform Japan's social care by 2020", n.d) as "the concept of social care robotics is much more accepted". This is because there is about "A quarter of Japanese people are over 65" and that there is "a shortfall of 370,000 caregivers" by 2025, creating a need for robots to help fill the lack of manpower in the healthcare sector.

Overview

In conclusion, there is a need to help the NHS's shortfall of staff, especially nurses as there is a disproportionate amount of nurses to patients ratio and a way to lower the ratio would be to complement the workforce with robotic help, even if their use is more administrative or low level diagnosis. We can see that there is a certain fear with incorporating robots and technology in the health sector because of the sensitivity and nature of the work but over time and with the help of the CARESSES project, robots such as Nao and Pepper will be more widely introduced in the sector.

We can also look at Japan and how well they accept and integrate with robots in their everyday lives. Japan shows how they have identified their shortfalls in staff within the healthcare sector and used robots to help fill the gap which is effective as the use of robots has leaked from commercial and private use to nationwide healthcare and domestic use.

QR codes

What is the use of QR codes in a medical environment?

QR codes are essentially modern barcodes that can contain data and this project aims to make use of its ability to access patient information. A way to apply QR Codes in a medical environment would be to replace paper wristbands given by hospitals, with QR generated wristbands, once they enter the A&E. The robot's ability to read QR Code will be used once the patient enters the assessment room and presents the robot their QR wristband.

By using QR wristbands, it will enable the robot to read information held normally on paper wristbands such as the patient's name, date of birth and their NHS number. In addition to this, by using QR Codes, normal and red wristbands, Normal wristbands with additional information such as allergies, risk of a fall or if a patient does not want to receive blood ("Why You Need to Wear a Wristband Whilst in Hospital", 2016), can be combined into one QR Coded wristband, therefore reducing waste. Using QR Codes can also speed up the data gathering for both robots and nurses as scanning QR codes is a quick way of extracting data, which is essential for quick responses from the robot/nurse. This adds an additional requirement to the robot, in that, the robot must have a camera available to achieve this functionality.

Planning

The robot

In this project, Nao will be used to help develop a robot to be able to be used in assessment rooms within hospital A&Es. Nao was chosen to be used due to limitation of choice within the university,

however it is a perfect framework to build around the project due to previously developed variants of Nao like MEDi, creating the groundworks of robotic use in the medical sector. Nao can easily be programmed using its dedicated IDE, Choregrahpe. This allows a great degree of freedom with what developers as they can control how Nao speaks, its mannerisms and how it can respond to specially tailored environments.

NAO robot overview

Nao is the smaller version of its big sister Pepper with the same specifications and functionality. The advantages of Nao over Pepper are its portability, due to its lightweight and small frame, and being more user friendly and approachable to children than Pepper.

Nao is capable of:

- Advanced language recognition skills
- Facial recognition and face tracking
- Object recognition
- Voice that can be tailored and express emotion with variable pitch
- HD Cameras and omnidirectional microphone
- Collision-avoidance sensors
- Move limbs

Nao's features:

- Language recognition
    - 4 Directional microphones
    - Engages in conversation
    - Incorporate information form the conversation
- Facial recognition and face tracking
    - Can identify objects and faces
    - Can detect movement and emotion
- Object recognition
    - Recognise and pick objects out of a room
- HD Cameras
    - 2 RGB cameras and 1 3D camera
- Collision-avoidance sensors
    - 2 Ultrasound transmitters and receivers (Range: 3 meters)
    - 6 laser sensors (Range: 3 meters)

- 3 Obstacle detectors in legs (Range: 3 meters)

What features will be used in this project?

The main features used to achieve the aims are:

- Language recognition - Using Nao's microphones will help with communicating with patients in the A&E assessment rooms. Nao can listen to patients and pick up keywords that will help identify the appropriate response, to pick up the symptoms of patients to correctly diagnose the patient. This can be implemented using:
  - The box library - It is a collection of scripts bundled into boxes, a fundamental object used in Choregraphe for manipulating the software in Nao, ranging from speech and listening, to movement and logic and they are written in python.
  - QiChat - QiChat is the advanced functionality for communication. QiChat will make use of a dialogue tree to respond to the varying but similar responses correctly and more appropriately from patient input.
- Limb movements - Nao's built in motors will allow it to move its limbs around. It can be used to do different motions such as waving someone goodbye or to nod its head. These motions can be used in tandem with its speech capabilities to make it more approachable, which is especially useful in the medical environment.
- Cameras - Nao's cameras will enable it to read QR codes from the patient, allowing Nao to obtain patient specific data (Name, DoB etc.). This will allow Nao to have unique interactions with every patient, as Nao will be able to talk to the patient using their names, making interaction with Nao comfortable for the patient.

Choregraphe

Choregraphe is the IDE that will be used to develop the robot. It is the official IDE for developing applications with Nao and Pepper. Choregraphe allows developers to "Create animations, behaviors [sic] and dialogs, Test them on a simulated robot, or directly on a real one, Monitor and control you [sic] robot, Enrich Choregraphe behaviors [sic] with your own Python code" (Aldebaran, n.d.a).

The IDE is a powerful tool, especially for developers who have limited knowledge in programming as Choregraphe allows for the creation of applications that contain Dialogs, services, behaviours such as interactions with people, dance and email sending all without having to write code. This allows a high level of abstraction that is sufficient to develop a Nao variant that can diagnose patients and suggest treatments.

Choregraphe also allows integrated API usage as part of the NaoQi framework (Aldebaran, n.d.b) - The main software that runs on the robot and controls it. NaoQi allows homogenous communication between the different modules/APIs such as motion, audio, and video. The framework also allows for cross-platform (Windows, Mac, Linux) and cross-language compatibility (C++, Python), making NaoQi flexible for any developer. A powerful tool that will be used during the development will be the virtual robot that is built in the IDE, allowing developers to emulate a Nao or Pepper robot without needing to have the physical robot, allowing development outside work hours. The official documentation will be used throughout the development. The documentation contains tutorials on basic and advanced functions for Nao to use.

For the project, Windows 10 will be used as the platform and Python will be used as the language to program the robot. Python will be used instead of C++ as I have experience with writing in the language however by using Python, I am able to expand my limited knowledge of Python further throughout the project.

QR Code

A second deliverable is to generate a QR code that can be read by Nao, which contains the necessary information about the patient such as Name, DoB, blood type etc. For continuity, python will be used generating for the QR code, with Sublime Text 3 being the desired IDE as it is quick in performance and lightweight. Generating the QR code will be handled by using the QR code python library.

Project management

Project management handles the methodologies and tools that will be used during the development of the project, and it is essential to identify them correctly as having the right methodology will help meet the project requirements. This allows a clear plan for the project by ensuring there are deadlines and milestones set, and by creating a clear view of what the final deliverable will be. A journal article (El Emam & Koru, 2008, p87) on a survey of IT software project failures claims that for IT projects, “15.52 percent were cancelled in 2005 and 11.54 percent were cancelled in 2007”. This is an alarmingly big percentage of cancellations, and on the 2007 survey, the article also asked about reasons as to why there are cancellations and found that one common reason “were requirement and scope changes [...] chosen by 33 percent of the respondents” and where followed by “lack of project management skills, [...] chosen by 28 percent of respondents”. This shows us the importance

of project management and how it can prevent cancellations of projects. This section will investigate the risk assessment of the project, and two different software development methodologies suitable for this project, and the accompanying tools that will be used to compliment the chosen methodology for the project.

Risk analysis

Risk analysis is the process of identifying possible risks during the development life cycle, where any identified risk must be planned for and categorised into different risk factors (Low, medium, high). In a paper on risk analysis techniques (Bennet, Bohoris, Aspinwall, Hall, 1996), a quantitative risk analysis is separated into four different stages: “identification of the hazards in the system;”, “estimation of the likelihood of a hazard becoming an accident;”, “discovery of the consequence of the hazard if it were to occur;” and “decision of whether the hazard should be reduced, avoided or eliminated altogether”.

Identification of possible risk during development includes delays occurring due to inexperience and unfamiliarity with Python, Choregraphe and Nao, possible lack of sufficient testing, possible addition of features during/after development and possible changes to development due to lack of resources/materials.

Table 1[1]

Estimation and ranking of risks during the project

Risk description	Likelihood of occurrence	Loss Size (Hours) in the project	Risk Exposure (Hours)	Category ranking[2]
Beginner knowledge of Python, causing delays in code development	90%	10	9	Low
Lack previous knowledge of how to use the Choregraphe IDE, causing for further research and practice	75%	3	2.25	Low
No previous experience in using the Nao robot, causing for further research and practice	75%	12	9	Low
Lack of sufficient testing, leading to research on other types of testing	50%	2	1.0	Low
Unexpected requirement/features to be added during or after development	45%	16	7.2	Medium
Unexpected changes to development methods/lack of	10%	6	0.6	Medium

[1] Estimation of occurrence and loss is calculated using personal opinions with such risk during software development in mind. 10 personal opinions of other software engineering students I have worked with in projects, are used to calculate risk occurrence. To see the guide used on how to estimate objective risk, see Appendix A.

[2] Out of the 10 opinions, I also asked them to individually classify each risk as Low, Medium or High if they did occur, and used an equation to classify each risk overall. For further details, see Appendix B.

equipment				
Total risk exposure			29.05	

Identification of possible risk during development includes delays occurring due to inexperience and unfamiliarity with Python, Choregraphe and Nao, possible lack of sufficient testing, possible addition of features during/after development and possible changes to developments due to lack of resources/materials.

In response to the possible risks occurring, contingency plans must be created to ensure development continues:

- Inexperience in Python, Choregraphe and Nao - These risks can be mitigated by reading documentation and watching tutorials. While this is an easy mitigation of the risk, it is also reflected with its low risk category. However, the risk exposure is high due to the time-consuming nature of the mitigation of such risk
- Lack of sufficient testing - This risk can be mitigated by exploring different ways to test software.
- Risk of adding additional features - This risk can be mitigated depending on the model used for development.
- Risk of unexpected change to development method - This is categorised as medium; however, it is expected that there is a small chance of occurrence.

In conclusion to the risk analysis conducted in Table 1, it can be seen that the project as a whole, with regards to the identified risk, is projected to be a low-medium level of risk with a clear possible solution to the risks that could occur. Since this is a small project, the worst-case scenario that might befall the development would be an increase in overall development time.

Prototyping model

The prototyping methodology is a software development model where a prototype is built, tested then reworked using feedback until there is an acceptable prototype. This is an iterative, trial and error method. In Bischofberger & Pomberger's (2012) book about prototyping, they explain that there are three different types of prototyping: "Explorative", "Experimental" and "Evolutionary" prototyping. This book (Bischofberger et al, 2012) further explains that evolutionary prototyping is "incremental system development, i.e., a successive development strategy with the following approach: A prototype is developed for those user requirements that are obvious from the start."

Advantages of prototyping:

- “Develop high-risk or major functions first” (Alshamrani & Bahattabs, 2015)
- “Risk is spread across smaller increments instead of concentrating in one large development” (Alshamrani et al, 2015)
- “Lessons learned at the end of each incremental delivery can result in positive revisions for the next increment” (Alshamrani et al, 2015)
- “Each release delivers an operational product” ((Alshamrani et al, 2015)
- “Initial product delivery is faster” (Alshamrani et al, 2015)
- “Reduces the risk of failure and changing the requirements” (Alshamrani et al, 2015)

Disadvantages of prototyping:

- “Requires good planning and design” (Alshamrani et al, 2015)
- “Requires early definition of a complete and fully functional system to allow for the definition of increments” (Alshamrani et al, 2015)
- “The model does not allow for iterations within each increment” (Alshamrani et al, 2015)

Spiral model

The spiral model is a more risk-driven software development process compared to the prototyping model. While prototyping does include risk handling, the risks must be identified first before development of an iteration starts, while the spiral model handles the risk after development starts, usually at the end of the phase during a phase review. The spiral model is therefore an improved version of the prototyping model as it allows for a review at the end of each phase within the model.

In Alshamrani et al’s (2015) paper on the comparison of the “Waterfall model, Spiral model, and Incremental/Iterative Model”, they describe the spiral model to be “both design and prototyping in stages for the sake of combining the advantages of top-down and bottom up concepts.”. Further on, they describe the process of the spiral model in the following:

> In this model, the development team starts with a small set of requirements and then goes through each development phase (except Installation and maintenance) for those set of requirements. Therefore, the development team has a chance to learn new lessons from the initial iteration (Via a risk analysis process). (p. 107)

Advantages of the spiral model:

- “High amount of risk analysis” (Alshamrani et al, 2015)

- “Software is produced early in the software life cycle” (Alshamrani et al, 2015)
- “Additional functionality can be added at a later date” (Alshamrani et al, 2015)
- “Project monitoring is very easy and effective” (Alshamrani et al, 2015)
- “Concerned people of a project can early review each phase and each loop as well because of rapid prototyping tools” (Alshamrani et al, 2015)
- “Suitable to develop a highly customized product” (Alshamrani et al, 2015)
- Provides early indication of insurmountable risk” (Alshamrani et al, 2015)

Disadvantages of the spiral model:

- “Cost involved in this model is usually high” (Alshamrani et al, 2015)
- “Risk assessment expertise is required” (Alshamrani et al, 2015)
- “Time spent for evaluating risk for small or low-risk projects may be too large” (Alshamrani et al, 2015)
- “Time spent for planning, resetting objectives, doing risk analysis, and prototyping may be excessive” (Alshamrani et al, 2015)
- “Projects success is highly dependent on risk analysis phase” (Alshamrani et al, 2015)

Comparison and conclusion

When comparing both prototype and spiral models, they are like each other in terms of strengths and weaknesses, as seen in Table 1 of Alshamrani et al’s paper (2015). Instead, looking at how well they apply to the project needs would be a better measure of which model to use for the project.

The model chosen for the project will be the prototype model. In Table one (Alshamrani et al, 2015), the prototype model is best used for “Low to medium-risk projects”, “a project with new technology, allowing the user to adjust to the system in smaller incremental steps rather than leaping to a major new product” and “When it is high risky [sic] to develop the whole system at once.”, which aligns with the project needs and situation. Due to the nature of breaking the project into small working deliverables, it works well with how code/functionality are segregated into diagram boxes within the Choregraphe IDE, enabling development to focus on building a project in increments. This is because of the low coupling between the diagram boxes, allowing for changes/creation of functionality without creating problems with other functions. In addition to this, by using Trello, different functionality can be listed individually to make development of the software easier and organised.

Development

Throughout the development process, the spiral model of software development was used, as researched from the previous section. In addition to the model, the risk analysis conducted will also be taken into consideration during the development to ensure that there is a plan in case a risk does occur. To organise the process, Trello will be used to organise the development to ensure that everything is within the scope.

Throughout the development, three types of boxes will be used when creating scripts for Nao in Choregraphe:

- Script box - Includes a script written in python (i.e. A say box)
- Diagram box - Includes multiple script boxes (i.e. A say and listen script box)
- Timeline box - Includes a Timeline for motion (i.e. A motion box)

All types of boxes will have an input and an output. The input will be triggered by the start of the program or activated by another box's output. To trigger a box's output, a condition must be met within the script of the box to activate. For example, an IF box will have two outputs, where one can be triggered depending on the login in the script. The root diagram box will have the onStart and onStop input and output, signalling the start and end of the behaviour program respectively, whereas a diagram box has similar inputs and outputs, but can be connected to other types of boxes.

Trello

Trello is a Kanban-style list-making application to help keep projects on track. Each task for the project is separated into different functionality that needs to be added to the robot (Joint movements, diagnosis and treatment, QR code etc.). Below, Figure 2.1 shows parts of the Trello board at the beginning of the project, where each functionality is separated into three subsections of itself - Functionality to be added, functionality to be tested and functionality that is finished. Each card is also colour coded according to which functionality they belong to (Green - Joint movements, Yellow - Diagnosis and treatment etc.) to make them easier to see and make sure they do not mix. The way Trello will be used during the development will be integrated with how the spiral model works, where features will be chosen to be developed and tested, then move to the next iteration that will build upon the previous one and its features.

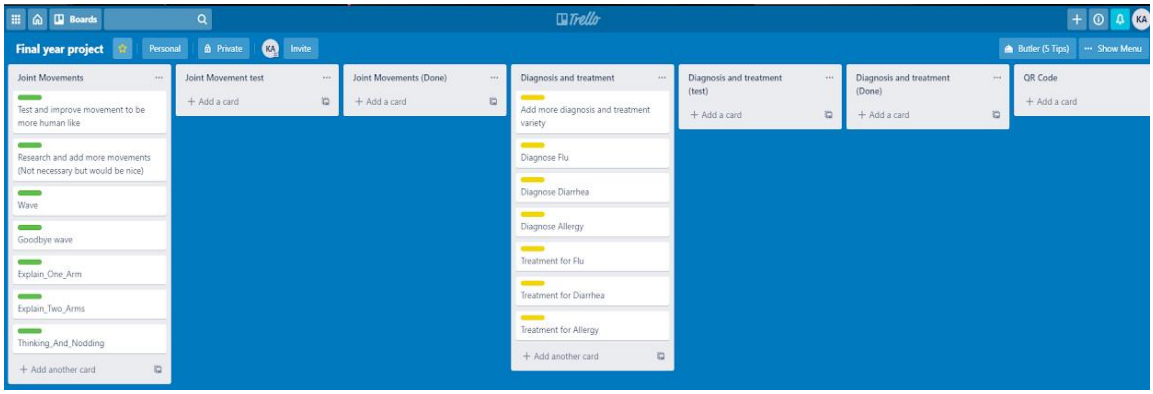


Figure 2.1 - The Trello board at the start of development.

Throughout the project, functionalities will be completed, and the Trello board will reflect the changes throughout as seen on Figure 2.2. In Figure 2.2, the development process is at an advanced iteration of the development and test section, as most features have been created, while some are also tested and fully complete.

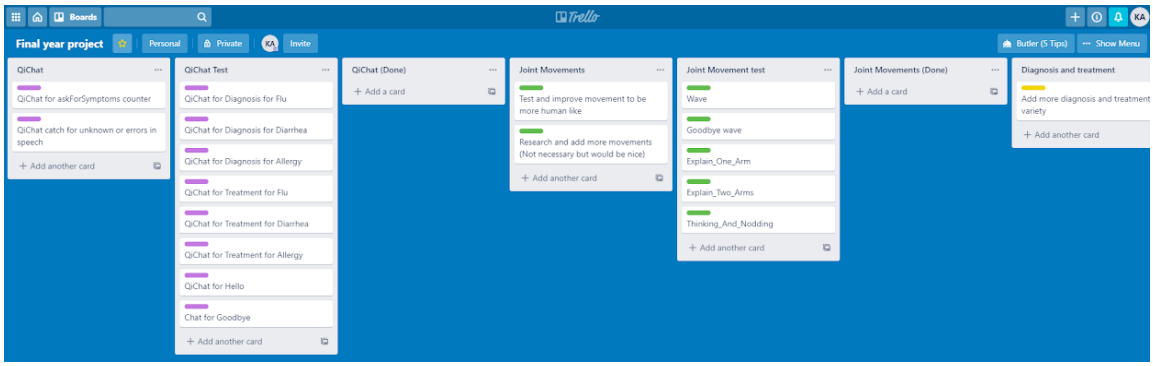


Figure 2.2 – The Trello boarding during the middle of development.

Developing the skeleton layout of the project

The first iteration is to create the flow of conversation between Nao and the patient. This will encompass the beginning of the conversation between Nao and the patient, to suggesting the patient the correct treatment according to Nao's diagnosis, and sending them away. By doing this, the backbone of how a general interaction between Nao and a patient will go can be visualised. This is to take advantage of Choregraphe's box system, where specific functionality can be encapsulated into diagram boxes, allowing for an organised way of implementing different iterations of the skeleton with different functionalities added and modified over time, taking full advantage of the spiral model of software development.

Since Nao is supposed to closely act like a nurse in an assessment centre, a start would be for Nao to greet the patient as they enter the room, followed by a question of what symptoms the patient is experiencing. From here, Nao must find a way to calculate what illness the patient has according to

what symptoms they have then, depending on the illness, correctly suggest a treatment for the patient for their illness, before sending them away.

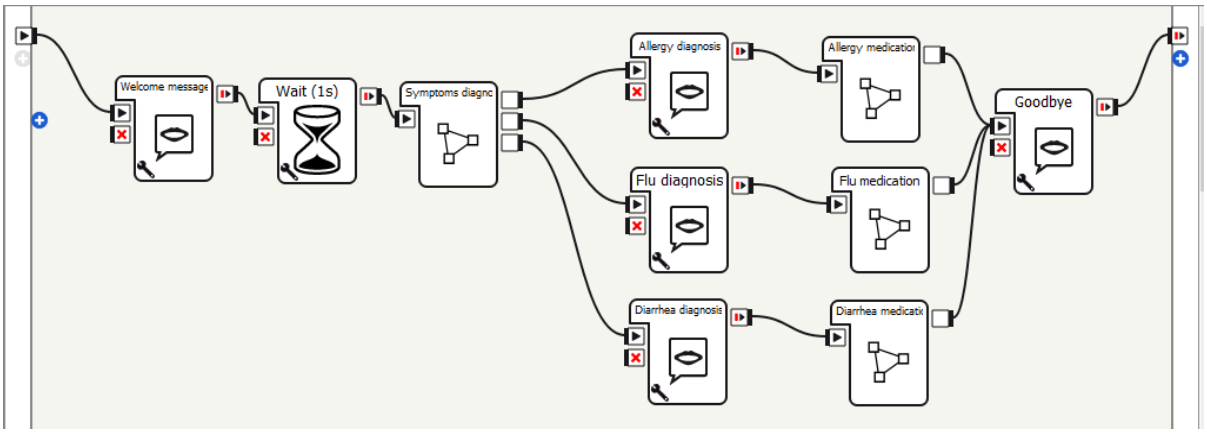


Figure 2.3 – The skeleton of what flow the program will go through from start to finish.

Figure 2.3 illustrates what the overall interaction will be, where the diagram boxes for the symptom diagnosis and the illness medication being functionality will be added and expanded upon, as different iterations of the skeleton layout are implemented. The skeleton is created by dragging and dropping general functionality such as the say box for the welcome and goodbye messages, while for functionality that requires deeper modification such as the symptoms diagnosis box, they can be created by selecting multiple script boxes[3] for the functionality, right clicking and selecting "convert to box". Throughout the development section, these functionalities will be explored and implemented to replace the skeleton layout but retaining the overall flow and structure of interaction between Nao and the patient.

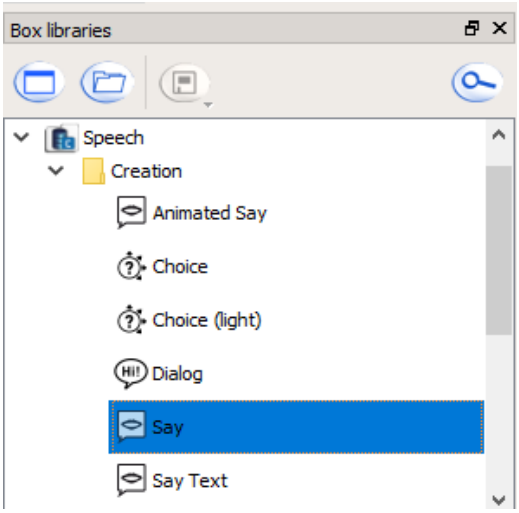


Figure 2.4 – The box library that contains functions for Nao and Pepper.

[3] This can be done by clicking and dragging over the selected script boxes, or by selecting the desired script boxes through shift clicking.

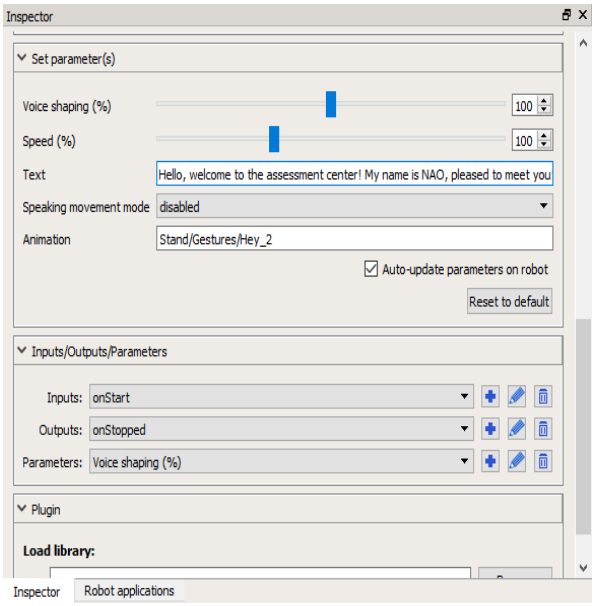


Figure 2.5 – The inspector window, where further tweaks can be done for boxes in Choregraphe.

The welcome and goodbye say box at the two ends of the skeleton, and the illness diagnosis box are implemented using the say boxes from the box library (Figure 2.4) by dragging them into the root area. Once dragged into the area, the inspector window will open (Figure 2.5), detailing the attributes of the box with important attributes being the name, description and the text that Nao will say. Each say script box will be tailor made for their function, for example the welcome say box will greet the patient with a hello.

Creating the diagnosis functionality

Speech and listen script boxes

The say box is the main way to get Nao to output speech from its speakers, whereas the listen box is a way for Nao to get speech input from a human. Implementing a listen box is like implementing a say box. The difference between the two boxes is the underlying script, and the attributes that can be changed. The listen box will have a word list attribute, where the box's output will trigger if Nao hears a word that is in the list.

The next iteration for Nao would be to further develop its speech and listening capabilities. The next step is to implement the symptoms diagnosis and the three illness medication boxes. This will require both say and listen boxes to implement a conversation. A basic layout that implements speaking and listening can be seen on Figure 2.6. This is a loop where Nao can speak and listen, and with an additional say box that loops back if Nao does not understand what has been said. This template is used whenever Nao needs to ask the patient and listen for a response and will be used

on the symptom's recognition diagram box, and the illness (Allergy, Flu and diarrhea) medication diagram boxes.

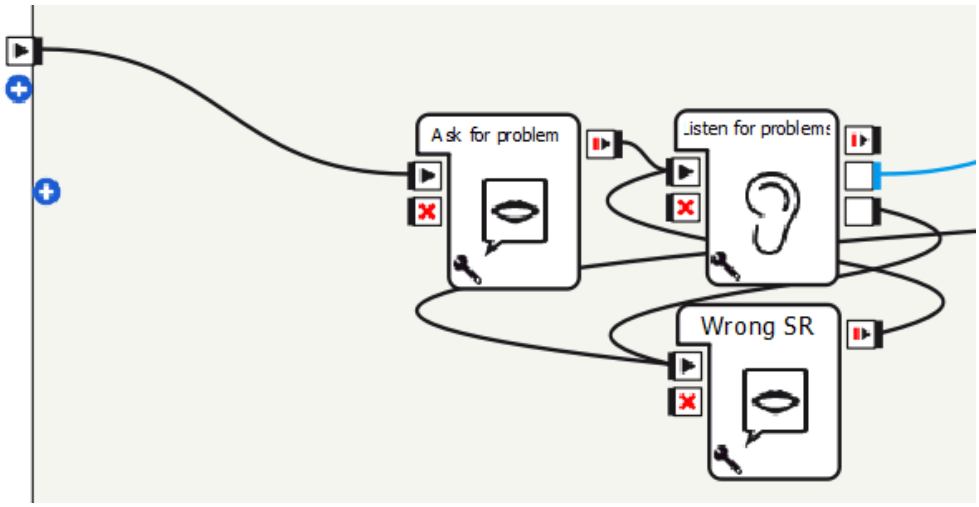


Figure 2.6 – A template used to implement Nao's speaking and listening ability.

Creating and implementing the diagnosis functionality

To diagnose an illness, the robot must listen for keywords of symptoms, and must correlate that to an illness. The problem with diagnosis is that a symptom can mean many illnesses, so there needs to be a way to accurately predict what the illness could be. For example, the flu, cold and COVID-19 symptoms share similar symptoms (See Appendix C) which can be mistaken for each other. A way to distinguish one from the other would be to assign the symptoms to their corresponding illness. Then by counting which labelled symptoms is more prevalent, even if an illness shares a symptom, the underlying illness can be found due to the patient having more symptoms of an illness they have, than an illness they don't have but share some symptoms with.

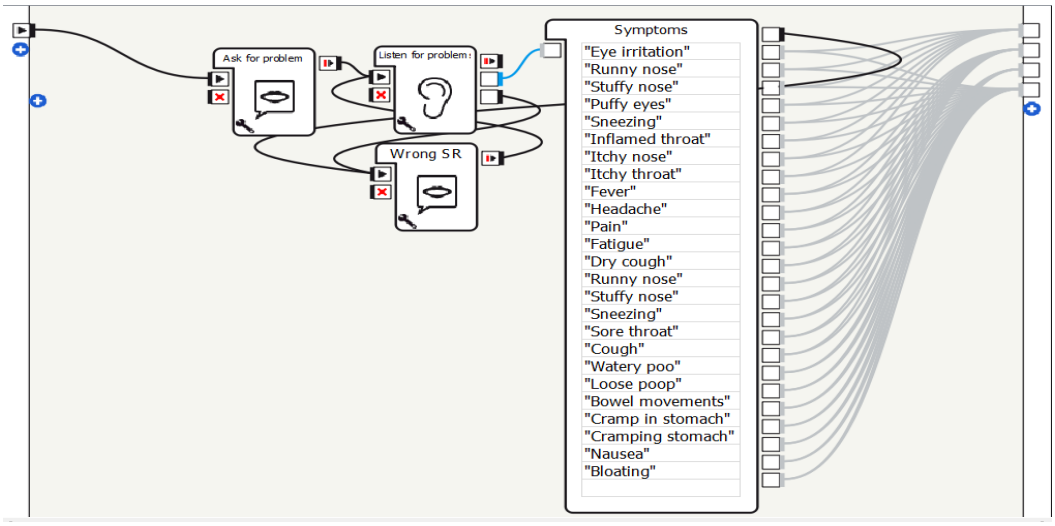


Figure 2.7 – The template for speaking and listening is paired with switch case box to listen for specific symptoms from a list.

In this implementation, the robot will be programmed with diagnosing patients for symptoms of allergies, the flu or diarrhea. They are chosen because they share some similar symptoms with each other, but also diversify the dialogue between Nao and the patient, as Nao will offer different treatments to each illness. The symptoms of each illness are listed in the switch case box called Symptoms, as shown in Figure 2.7[4]. Nao will ask for symptoms that the patient is experiencing, and once Nao hears a word from the list (by activating the blue noodle), it will activate it's two outputs, or, if the patient doesn't list a symptom from the switch case, Nao will ask for the patient to repeat themselves by activating the Wrong SR (Speech recognition) box, looping back to Nao asking the patient to list out a symptom.

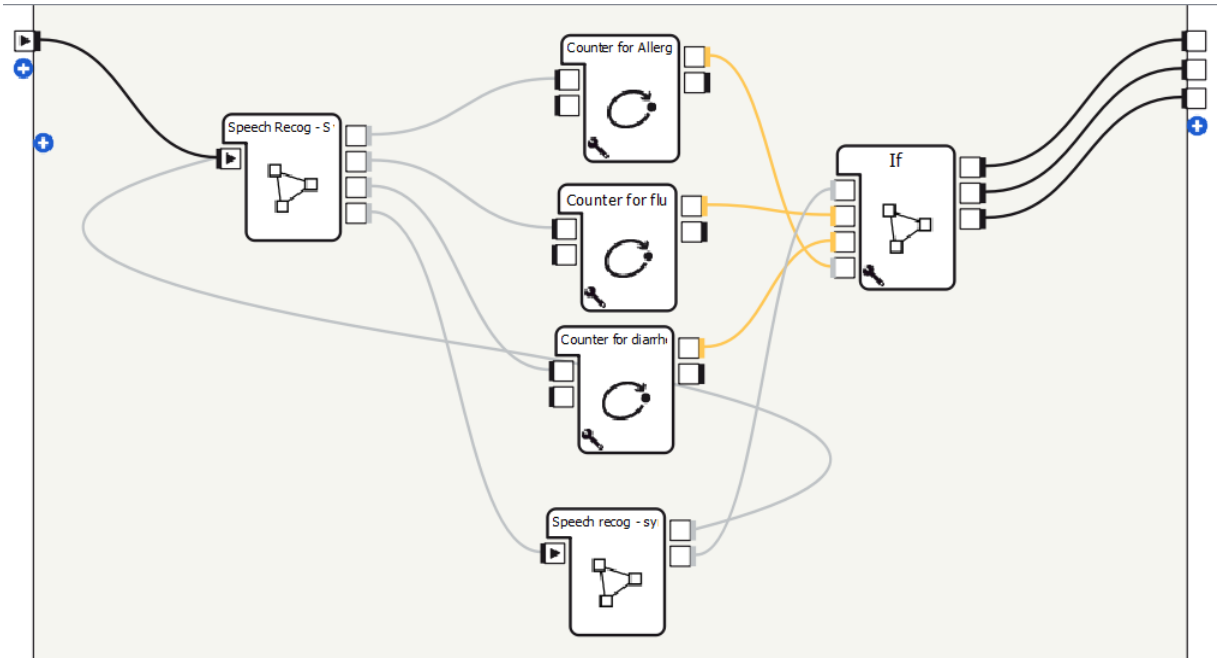


Figure 2.8 – The whole dialogue tree and logic for the symptom's diagnosis. The left-most box is a diagram box that contains the template in figure 2.6 for speaking and listening.

Once the patient says a keyword and Nao hears it, it will activate its two outputs- One (or more, if the symptom is shared by another illness) noodle to increment the correct illness counter, and another to proceed with the dialogue, both seen in figure 2.8. Due to an illness possibly displaying multiple symptoms, and to make the diagnosis more accurate since some illnesses share symptoms, Nao will ask if the patient is experiencing any more symptoms. An affirmative from the patient will loop back to Nao asking and listening for symptoms, while a negative will lead to an IF statement. The IF statement will use the values from all three counters to calculate which illness the patient will

[4] This figure- The speech recognition logic for symptoms, is the box that the onStart input is connected to on figure 2.8

most likely have by comparing all three values and finding which counter has the biggest value, therefore concluding which illness the patient is most likely to have.

```
def onUnload(self):
    #~ puts code for box cleanup here
    pass

def onInput_inputAllergy(self, aCount)
    self.aCount = aCount

#input for flu count
def onInput_inputFlu(self, fCount):
    self.fCount = fCount

#input for diarrhea count
def onInput_inputDiarrhea(self, dCount
    self.dCount = dCount

def onInput_onData(self, p):
    try:
        f = self.fCount
        try:
            f = float(f)
        except:
            f = str(f)
    except:
        f = 0


    try:
        d = self.dCount
        try:
            d = float(d)
        except:
            d = str(d)
    except:
        d = 0

    try:
        p = self.aCount
        try:
            p = float(p)
        except:
            p = str(p)
    except:
        p = 0

    #if allergy counter is bigger
    if ( p > f and p > d):
        self.outputA()
    #if flu counter is bigger
    elif ( f > p and f > d):
        self.outputF()
    #if diarrhea counter is bigger
    elif (d > f and d > p):
        self.outputD()
```

Figure 2.9 – A screenshot of the logic for the IF box to decide which illness the patient has, using their symptom count.

Creating the illness medication functionality

The next interaction between Nao and the robot will be when Nao recommends a medication suitable for the patient's illness. After the IF box calculates and activates the noodle, it will lead to

one of three conversational paths, one for each illness, where Nao will ask the patient if they have any allergies to medication. Nao will then use that information to correctly recommend the patient the correct medication, or an alternative if applicable.

To better plan for a dialogue path, a flow diagram should be created as it is the best way to visualise the different paths the patient can take throughout the conversation. Figure 2.10, Figure 2.11, and Figure 2.12 shows the flow diagrams relating to the three ways Nao can recommend treatment for the flu, allergy and diarrhea illness. The flow diagram can be easily converted into logic in Choregraphe, as each box can be interpreted as a say box that Nao will ask, with each line being the answer given by the patient. In addition to the paths, at the end of each diagnosis, Nao will ask the patient to consult the attending nurse for a prescription for their medication, as well as for them to ask further questions.

Implementing the treatment to allergies

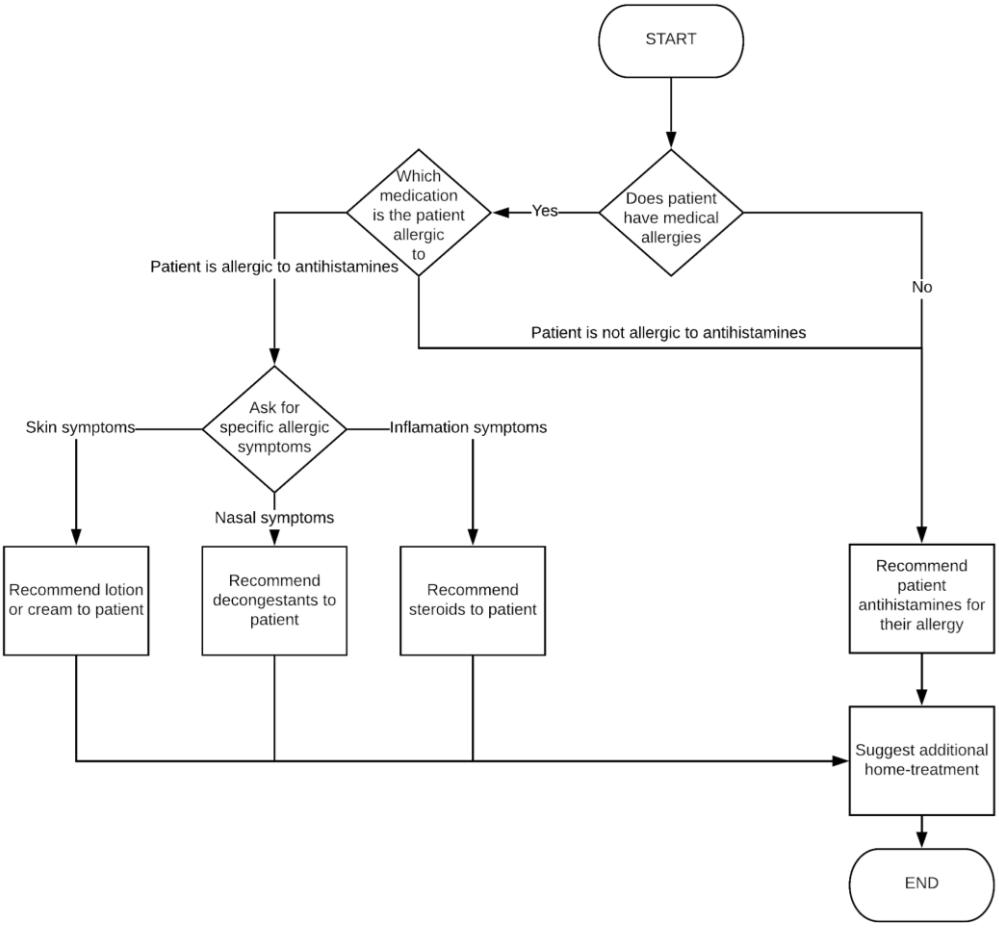


Figure 2.10 – The flow diagram for the allergy treatment dialogue tree.

The first path is to implement the allergy medication dialogue path in accordance with the flow diagram. The first step was to create a way for Nao to ask the patient if they have any allergies to

medication. The template in Figure 2.6 can once again be used and is implemented through an additional diagram box. The next step would be to create the path when the patient does not have any allergies to medication, which will default Nao to suggest antihistamines for the patient's illness, as well as suggesting further at-home treatments that require no medication. Once the path is complete, the path where the patient is allergic to medication must be created. This path will lead to Nao asking which medication the patient is allergic to. Nao will only listen for keywords relating to medication that treats the patient's illness, as Nao needs to know which medication to avoid suggesting. This is implemented by using the same template as Figure 2.6, while changing the switch case to contain medication related to the illness as shown in Figure 6.3. The figure shows that if the patient is allergic to medication that is not antihistamines, Nao will suggest antihistamines as they are used for treating any kinds of allergies, which will then path to home treatments which was implemented before.

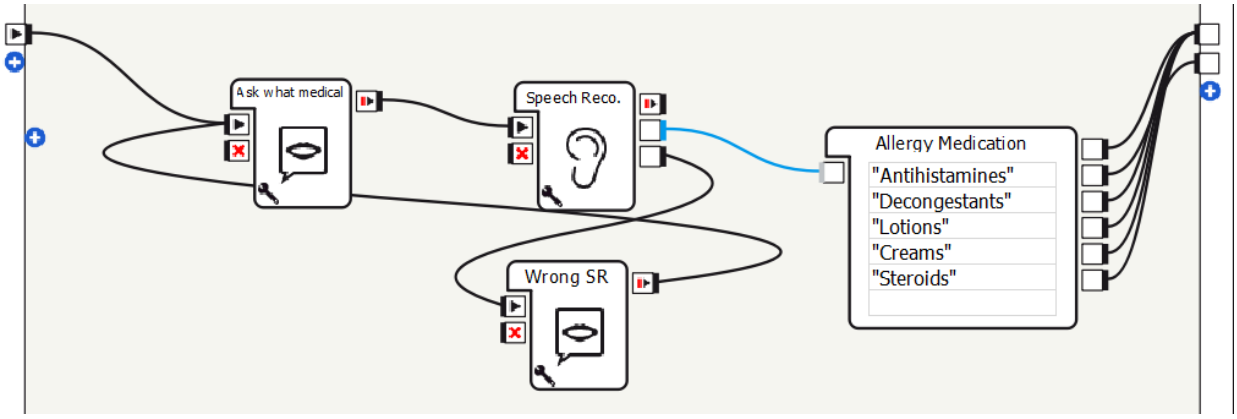


Figure 2.13 – A diagram box that contains the speaking and listening template and a switch box to listen for certain types of medication.

If the patient is allergic to antihistamines, Nao must further ask what symptoms they are experiencing, as allergies can have different symptoms with different medications to use (Figure 2.14). The answer given by the patient will lead to Nao finally having enough information to suggest the correct treatment (Figure 2.15). Once the correct medication has been suggested to the patient, the paths will converge to the say box that suggest further self-treatment at home.

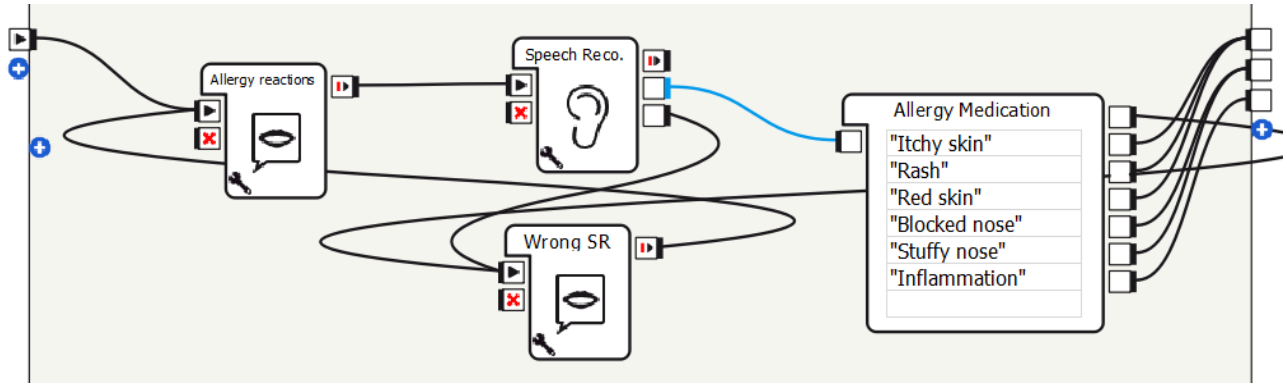


Figure 2.14 – Another diagram box that allows Nao to listen for further allergic symptoms for Nao to correctly suggest the right treatment.

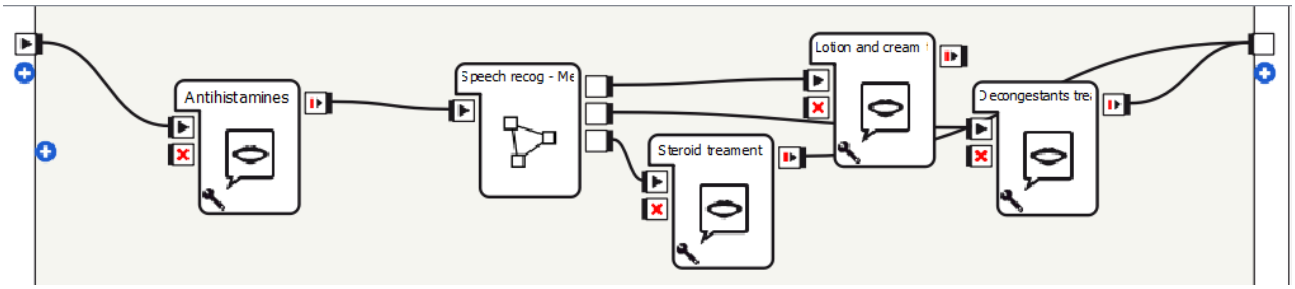


Figure 2.15 – The flow of conversation that determines which medication to suggest if the patient is allergic to antihistamines.

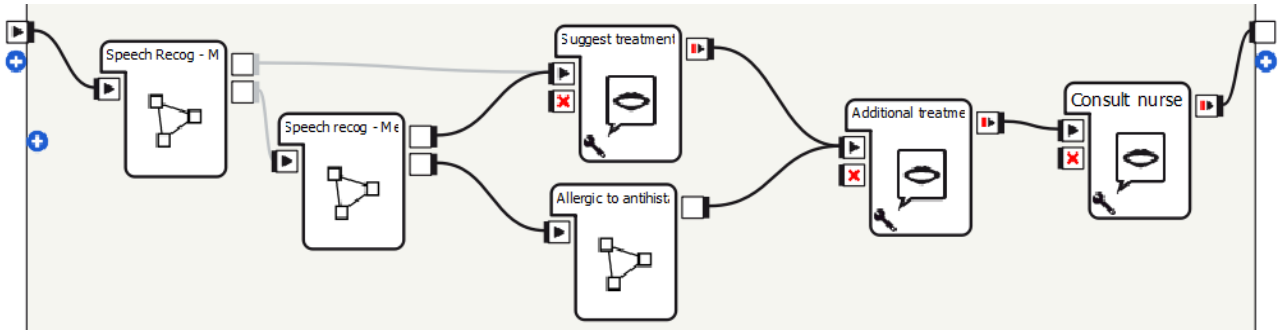


Figure 2.16 - An overview of the allergy diagram box that contains all the speaking and listening features that is tasked with administering the correct treatment for the patient.

Implementing the treatment for flu

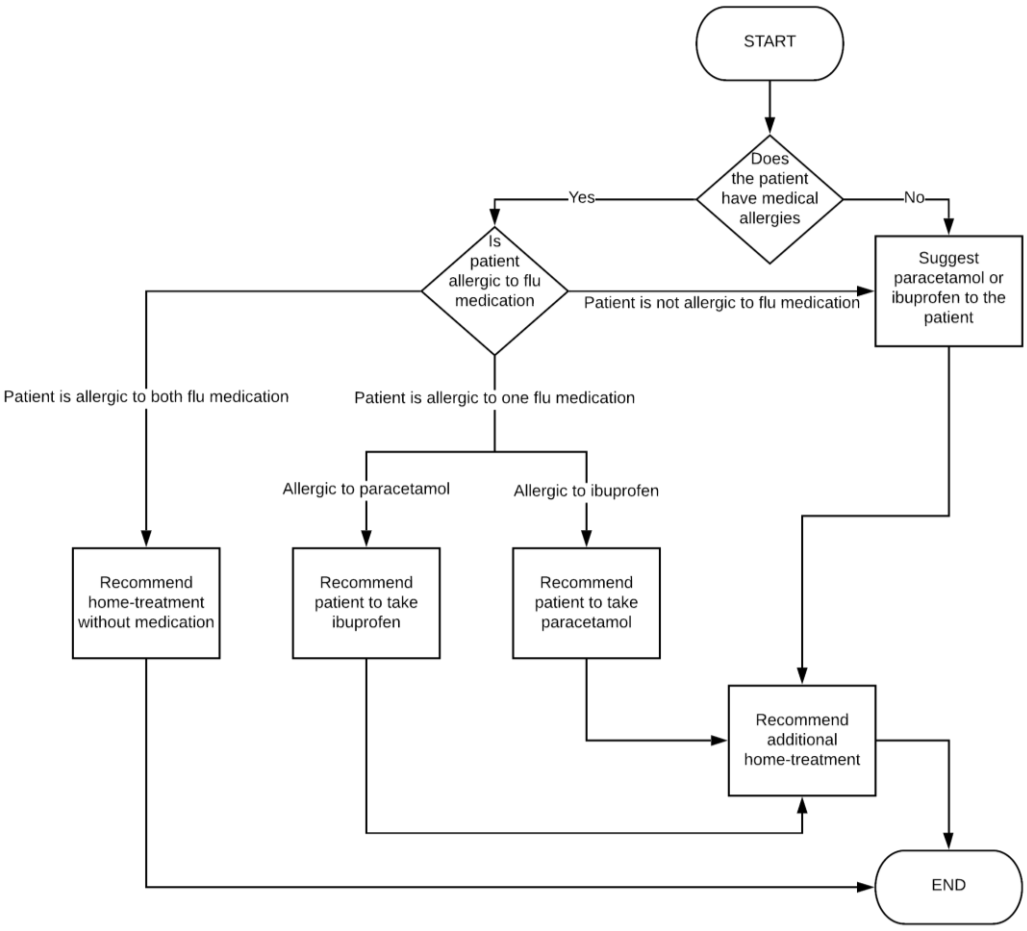


Figure 2.11 – The flow diagram for the flu treatment dialogue tree.

The treatment for the flu is like the treatment for allergies. Nao must ask if the patient is allergic to medication. There, the patient can have no allergies to which Nao will suggest paracetamol or ibuprofen, but suggest more additional at-home treatments, as that is the best way to treat the flu. However, another path would be if the patient is allergic to paracetamol, ibuprofen, both or allergic to medication not related to the treatment of the flu symptoms (Figure 2.17). If the patient is not allergic to either medication, they will be suggested the normal treatments, otherwise Nao will suggest the opposite medication, depending if they are allergic to paracetamol or ibuprofen. The worst-case scenario would be if the patient is allergic to both medications, as Nao must inform them that they cannot assign them medication and to try home remedies instead. All the paths will then converge on the “Consult nurse” say box.

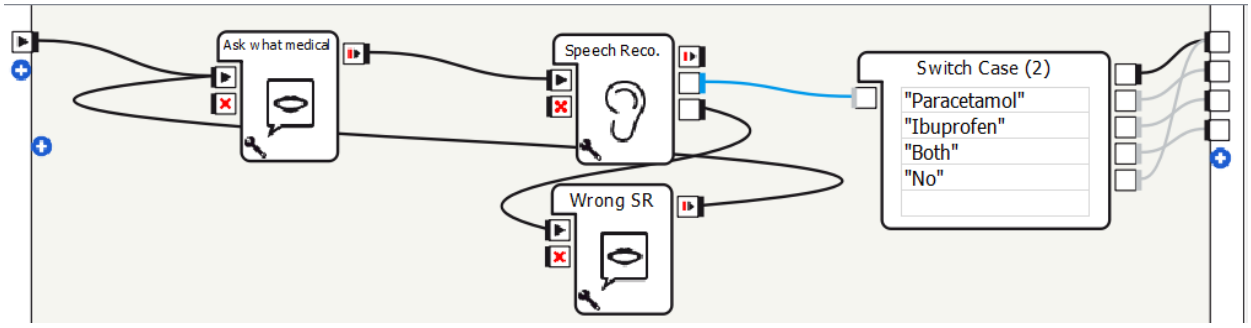


Figure 2.17 – The template being used to listen for any allergic reactions to flu medication.

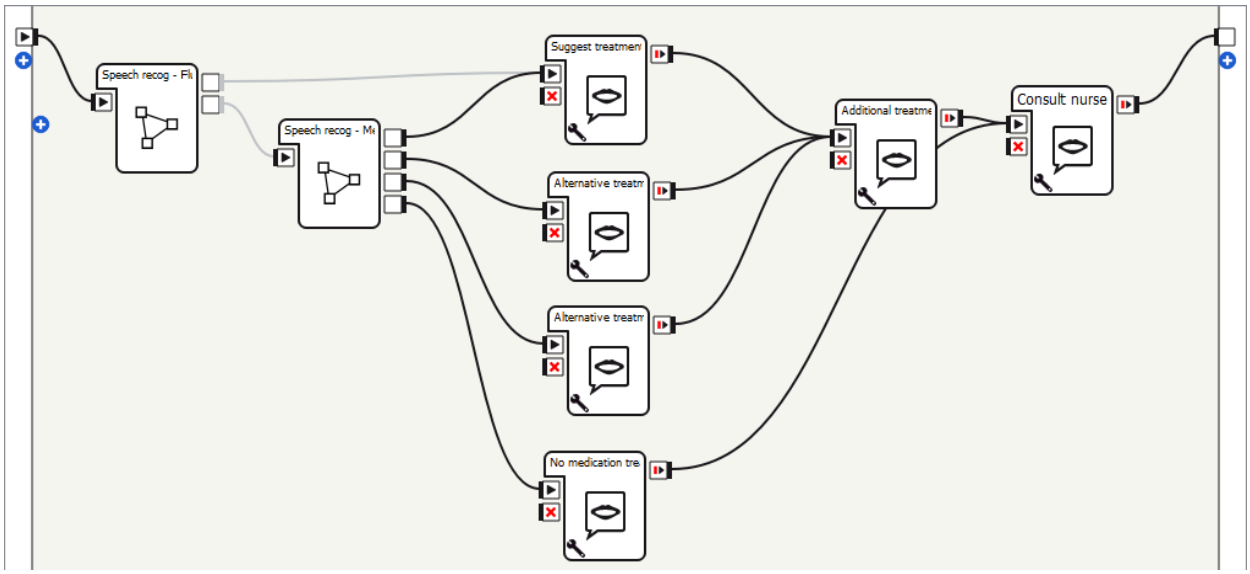


Figure 2.18 – An overview of the whole flu treatment diagram box.

Implementing the diarrhea treatment

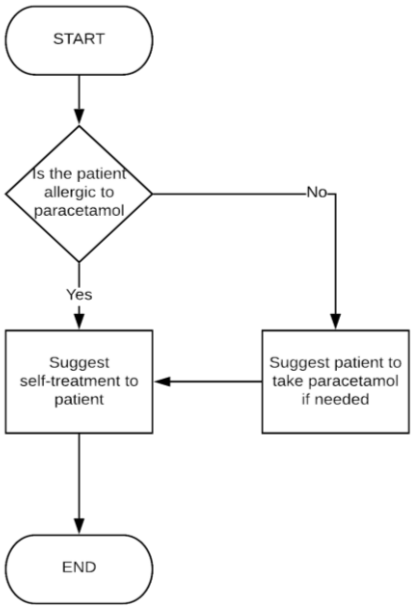


Figure 2.12

Implementing treatment for diarrhea is more straightforward than the other treatments, as the only medication to help treat the symptoms would be paracetamol. Therefore, Nao will only ask if the patient is allergic to medication, where Nao will suggest taking medication to treat their symptoms then the main treatment, or to suggest the main treatment straight away - Self-treatment at home.

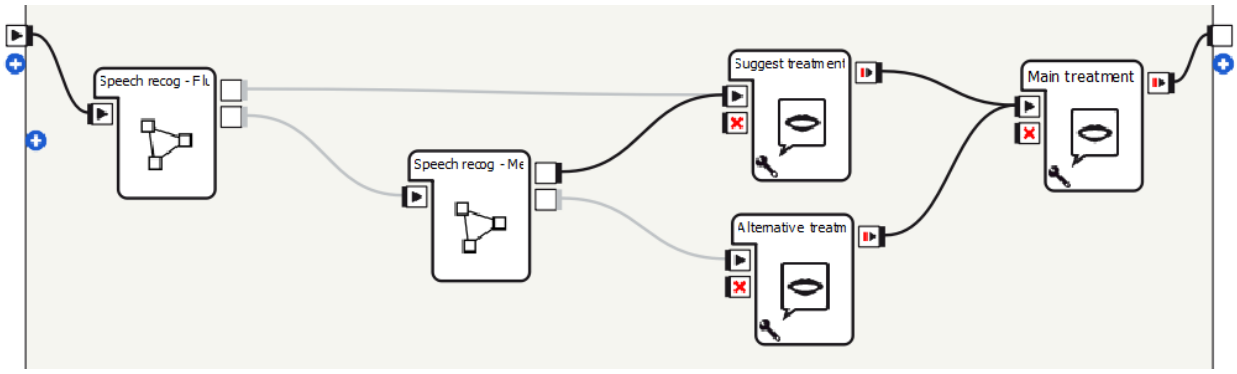


Figure 2.19 – An overview of the diarrhea treatment diagram box.

Implementing QiChat to replace speech and listening boxes

After the review of speech capabilities of Nao, there was a need to use a more efficient and less bulky way of creating dialogue between Nao and the patient. The proposal is to use QiChat. QiChat is the language used to write dialog topic boxes, effectively replacing both say and listen boxes. It is a script driven language that uses a modified version of python with special syntax that focuses on voice chat.

The main concept of QiChat is rules. Rules are what Nao will abide by when it hears an input from the patient, where depending on the input, Nao will respond accordingly. The power of QiChat is the layer of rules (Sub rules), where the scope of each rule can be defined, and of additional syntax such as proposals and topics. Through a combination of rules, proposals and topics, a whole conversation can be made using one box. Another reason as to why QiChat should be implemented is that QiChat allows Nao to listen for different ways a person can relay their intentions using a variation of different words. Also, by implementing QiChat, it will replace the bulky diagram boxes that contain multiple say and listen boxes, while keeping the welcome and goodbye message as they are simple one sentence functions.

Implementing symptoms diagnosis in QiChat

The new QiChat implementation for the symptoms diagnosis functionality replaces the diagram box containing the template that is responsible for listening and reacting to the patient input. The script will fully incorporate proposals and concepts to implement the dialogue tree that Nao will traverse. The script gets activated when the onStart input is activated, where it will start the dialogue tree,

where it will start talking then go to the proposal SYMPTOMS, which signifies the topic of conversation between Nao and the patient. The user's input can be seen with the pattern of "u: (Hello)", where u is the user and text inside the bracket being the input. However, to take in all possible ways the user can respond, concepts are used instead of hard coding user input, where the concepts will list out possible ways the user can respond. The flow of code will follow the proposals - The set ways a conversation can go, by using the ^goto() function. The proposals will then create a dialogue tree, where the number after the user (u1, u2 etc.) will showcase the layer of conversation in the tree. Figure 2.20 shows that there are three branches on the first layer of dialogue, as there are three u1's, and so on.

```
u: (e:onStart) ~askForSymptoms ^goto(SYMPTOMS)

$runnyNose="Runny nose"
$fever=Fever

proposal: %SYMPTOMS
    u1: (_~allergy) ~confirmation . ~question ["$runnyNose==$1^gotoReactivate(SIMILARYSYMPTOMSFLUALLERGY)"
"$allergyCount=1"]
        u2: (~yes) ~thanks . $onStopped=1
        u2: (~no) ~askForMoreSymptoms . ^gotoReactivate(SYMPTOMS)
    u1: (_~flu) ~confirmation . ~question ["$runnyNose==$1 ^gotoReactivate(SIMILARYSYMPTOMSFLUALLERGY)" "$fever==$1
^gotoReactivate(SIMILARYSYMPTOMSFLUDIARRHEA)" "$fluCount=1"]
        u2: (~yes) ~thanks . $onStopped=1
        u2: (~no) ~askForMoreSymptoms . ^gotoReactivate(SYMPTOMS)
    u1: (_~diarrhea) ~confirmation . ~question ["$fever==$1 ^gotoReactivate(SIMILARYSYMPTOMSFLUDIARRHEA)"
"$diarrheaCount=1"]
        u2: (~yes) ~thanks . $onStopped=1
        u2: (~no) ~askForMoreSymptoms . ^gotoReactivate(SYMPTOMS)

proposal: %SIMILARYSYMPTOMSFLUALLERGY $fluCount=1 $allergyCount=1
    u1: (~yes) ~thanks . $onStopped=1
    u1: (~no) ~askForMoreSymptoms . ^gotoReactivate(SYMPTOMS)

proposal: %SIMILARYSYMPTOMSFLUDIARRHEA $fluCount=1 $diarrheaCount=1
    u1: (~yes) ~thanks . $onStopped=1
    u1: (~no) ~askForMoreSymptoms . ^gotoReactivate(SYMPTOMS)
```

Figure 2.20 – The three proposals where the dialogue can branch to.

Concepts are listed above the proposals (Name inside the brackets), where the words or sentence (Wrapped around speech marks) are inside square brackets. This tells Nao what words/sentences it should look out for. This can also go both ways, as Nao can use the same idea and use a concept to vary its response to the patient.

Script editor

AskForSymptoms/AskForSymptoms_enu.top

```
topic: ~AskForSymptoms()
language: enu

# Defining extra concepts out of words or group of words
#concept:(hello) [hello hi hey "good morning" greetings]
concept: (allergy) ["Eye irritation" "Irritated eyes" "Runny nose" "Puffy eyes" Sneezing "Inflamed throat" "Itchy
nose" "Itchy throat"]
concept: (flu) [Fever Headache Pain Fatigue "Dry cough" "Runny nose" "Stuffy nose" Sneezing "Sore throat" Cough]
concept: (diarrhea) ["[Watery loose] [poo shit poop stool]" "[Bowel Stomach Belly Tummy Gut] [movements ache pain
rumble rumbling cramp]" "Cramping [Bowel Stomach Belly Tummy Gut]" Nausea Bloating Fever]
concept: (confirmation) [Ok "uh huh" Yep Alright Right]
concept: (question) ["Is that [all it "all your symptoms" everything]"]
concept: (yes) [Yes yep Yeah "[That's "That is"] [all it everything]" "It is" "I think so"]
concept: (no) [No " Nope "Not yet" "No ["I am" I'm] not [done finished complete]" "I [don't "do not"] think so]
concept: (thanks) ["Ok thank you" "Ok [thanks "thank you" great good]"]
concept: (askForSymptoms) ["Can you please tell me a symptom you are experiencing" "Can you please describe a symptom"
"What is a symptom that you are experiencing"]
concept: (askForMoreSymptoms) ["Can you please tell me another symptom" "Can you please tell me another symptom that
you are experiencing" "Can you please describe another symptom" "What is another symptom that you are experiencing"]
```

Figure 2.21 – The concepts used for the AskForSymptoms QiChat feature.

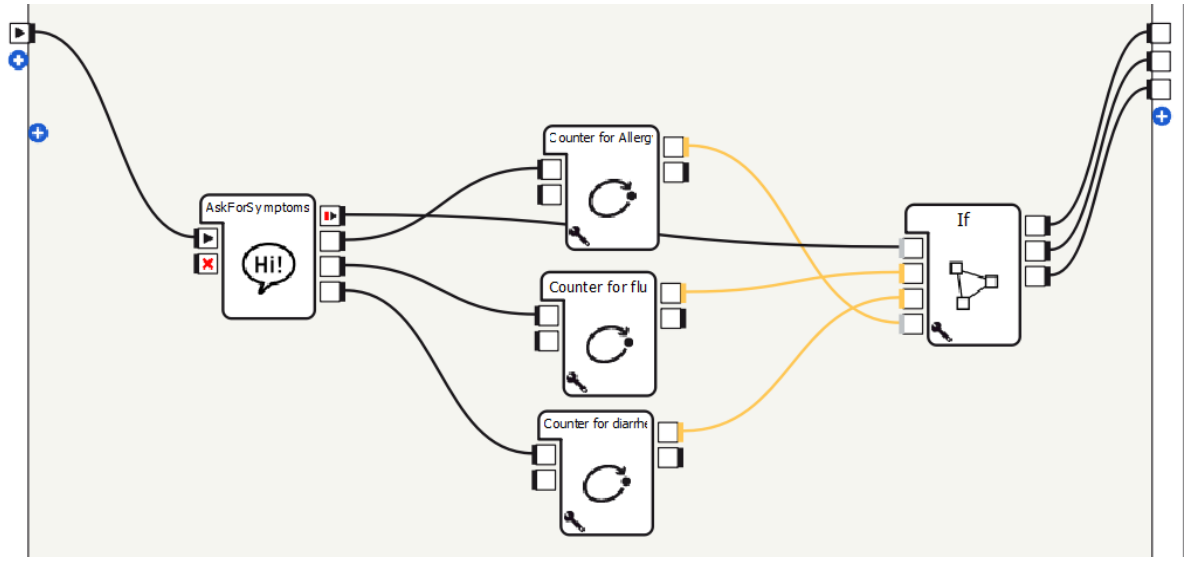


Figure 2.22 – The QiChat feature replacing the previous speech and listen boxes.

Implementing the treatments in QiChat

Implementation for the treatments are like implementing the diagnosis in the previous section. However, by implementing the QiChat feature, it reduces all the say and diagram boxes needed to implement all three treatments to only three dialogue boxes which fully reduces the complexity of the logic, as the dialogue tree is handled inside the script for each individual dialogue box. Each dialogue box is implemented using the same concepts used when implementing the symptoms diagnosis, using proposals and concepts to implement the dialogue tree desired.

```
topic: ~allergyTreatment()
language: enu

#These concepts are:
# - Possible responses by the patient
# - possible responses by the robot
concept: (allergicToMedicine) [Yes "Yes I am" Indeed "That is correct" Yeah Yep]
concept: (whichMeds) ["Which medication are you allergic to?" "Can you please tell me which medication you are
allergic to?" "Please tell me which medication you are allergic to"]
concept: (allergicTo) [Antihistamines Cream Lotion Steroids Decongestants]
concept: (allergicToAntihistamines) "Since you are allergic to antihistamines, I cannot assign you them. Can you
please tell me what allergic reaction you are experiencing."
concept: (notAllergicToAntihistamines) "Since you are not allergic to antihistamines, I will recommend you to use
antihistamines as it is the universal medication for allergies."

concept: (notAllergicToMedicine) [No "I am not" "No I am not" Nope "No I'm not"]

concept: (reactionSkin) ["Itchy skin" "Red skin" "Scaly skin" Rash "Skin irritation"]
concept: (reactionNose) ["Runny nose" "Blocked nose" "Teary eyes" "Trouble breathing with my nose"]
concept: (reactionInflamation) [Inflamation Pain]
concept: (skinTreatment) "Since you are having a skin allergy, I would recommend a cream or lotion."
concept: (noseTreatment) "Since you are having problems with your nose, I would recommend some decongestants to clear
it up."
concept: (inflamationTreatment) "Since you are having inflamation problems, I would recommend a steroidal treatment to
treat your inflamtion"

concept: (additionalTreatment) "In addition to the medication, I would suggest rest, sleep and plenty of water too."

concept: (consultNurse) "Please consult the nurse for a prescription and additional details."

concept: (repeat) ["Can you repeat please" "Can you please repeat what you said" "Come again?" "Sorry, what?" "What
did you said again?" "Sorry I [didn't didnt "did not"] [hear get understand] what you said"]

#The start of the dialogue box conversation
u:(e:onStart) It looks like you are having an allergic reaction! ^goto(ASK_MED_ALLERGIES)

#These proposals are the dialogue tree that the conversation can possible go
proposal: %ASK_MED_ALLERGIES Are you allergic to any medicine?
    u1: (~allergicToMedicine) ~whichMeds
        u2: (_~allergicTo) $allergy=$1 ["$allergy==Antihistamines ~allergicToAntihistamines ^goto(ALLERGIC_REACTION)"
"~notAllergicToAntihistamines ^goto(ADDITIONAL_TREATMENT)"]
    u1: (~notAllergicToMedicine) ~notAllergicToAntihistamines ^goto(ADDITIONAL_TREATMENT)
    u1: (~repeat) Ok I'll repeat. ^sameProposal

proposal: %ALLERGIC_REACTION ~allergicToAntihistamines
    u1: (~reactionSkin) ~skinTreatment ^gotoReactivate(ADDITIONAL_TREATMENT)
    u1: (~reactionNose) ~noseTreatment ^gotoReactivate(ADDITIONAL_TREATMENT)
    u1: (~reactionInflamation) ~inflamationTreatment ^gotoReactivate(ADDITIONAL_TREATMENT)
    u1: (~repeat) Ok I'll repeat. ^sameProposal

proposal: %ADDITIONAL_TREATMENT ~additionalTreatment ~consultNurse $onStopped=1
```

Figure 2.23 – The QiChat script for the allergy treatment.

```
topic: ~fluTreatment()
language: enu

#These concepts are:
# - Possible responses by the patient
# - possible responses by the robot
concept: (allergicToMedicine) [Yes "Yes I am" Indeed "That is correct" Yeah Yep]
concept: (whichMeds) Are you allergic to paracetamol, ibuprofen, both or neither?
concept: (allergicTo) [Paracetamol Ibuprofen Both Neither "I don't know" "I'm not sure"]

concept: (recommendParacetamol) Since you are allergic to ibuprofen, I would recommend you to take some paracetamol to
alleviate the discomfort
concept: (recommendIbuprofen) Since you are allergic to paracetamol, I would recommend you to take some ibuprofen to
alleviate the discomfort
concept: (recommendNeither) Since you are allergic to both medication to alleviate flu like symptoms, I will not
recommend you any medication for your symptoms. Instead I would recommend you to keep warm, rest, sleep and drink
plenty of water.
concept: (recommendEither) Since you are not allergic to either paracetamol or ibuprofen, It is up to you to take
either medication to alleviate your symptoms.

concept: (additionalTreatment) I would also recommend you to keep warm, rest, sleep and drink plenty of water for a
faster recovery.

concept: (notAllergicToMedicine) [No "I am not" "No I am not" Nope "No I'm not"]

concept: (consultNurse) "Please consult the nurse for a prescription and additional details."
concept: (consultNurseNoMeds) "Please consult the nurse for additional details."

concept: (repeat) ["Can you repeat please" "Can you please repeat what you said" "Come again?" "Sorry, what?" "What
did you said again?" "Sorry I [didn't didnt "did not"] [hear get understand] what you said"]

#The start of the dialogue box conversation
u:(e:onStart) It looks like you are displaying flu like symptoms! ^goto(ASK_MED_ALLERGIES)

#These proposals are the dialogue tree that the conversation can possible go
proposal: %ASK_MED_ALLERGIES Are you allergic to any medicine?
    u1: (~allergicToMedicine) ~whichMeds
        u2: (_~allergicTo) $allergy=$1 ["$allergy==Paracetamol ~recommendIbuprofen ^goto(ADDITIONAL_TREATMENT)"
"$allergy==Ibuprofen ~recommendParacetamol ^goto(ADDITIONAL_TREATMENT)" "$allergy==Both ~recommendNeither
^goto(CONSULT_NURSE_NO_MEDS)" "$allergy==Neither ~recommendEither ^goto(ADDITIONAL_TREATMENT)"]
    u1: (~notAllergicToMedicine) ~recommendEither
    u1: (~repeat) Ok I'll repeat. ^sameProposal

proposal: %ADDITIONAL_TREATMENT ~additionalTreatment ^goto(CONSULT_NURSE)

proposal: %CONSULT_NURSE ~consultNurse $onStopped=1

proposal: %CONSULT_NURSE_NO_MEDS ~consultNurseNoMeds $onStopped=1
```

Figure 2.24 – The QiChat script for the flu treatment.

```
topic: ~diarrheaTreatment()
language: enu

#These concepts are:
# - Possible responses by the patient
# - possible responses by the robot
concept: (allergicToMedicine) [Yes "Yes I am" Indeed "That is correct" Yeah Yep]
concept: (notAllergicToMedicine) [No "I am not" "No I am not" Nope "No I'm not"]

concept: (selfTreatment) That's alright, self-treatment is better for treating diarrhea anyway.
concept: (mainTreatment) I would recommend you to get enough rest and sleep, and to drink lots of fluids to keep
hydrated.
concept: (consultNurse) Please go see the nurse for further information on how to treat your condition at home.
concept: (recommendParacetamol) If you are feeling a lot of discomfort, I would recommend you to take paracetamol.
Also,

concept: (repeat) ["Can you repeat please" "Can you please repeat what you said" "Come again?" "Sorry, what?" "What
did you said again?" "Sorry I [didn't didnt "did not"] [hear get understand] what you said"]

#The start of the dialogue box conversation
u:(e:onStart) It looks like you are displaying diarrhea like symptoms! ^goto(ASK_MED_ALLERGIES)

#These proposals are the dialogue tree that the conversation can possible go
proposal: %ASK_MED_ALLERGIES Are you allergic to paracetamol?
    u1: (~allergicToMedicine) ~selfTreatment ^goto(RECOMMEND_SELF_TREATMENT)
    u1: (~notAllergicToMedicine) ~recommendParacetamol ^goto(RECOMMEND_SELF_TREATMENT)
    u1: (~repeat) Ok I'll repeat. ^sameProposal

proposal: %RECOMMEND_SELF_TREATMENT ~mainTreatment ~consultNurse $onStopped=1
```

Figure 2.25 – The QiChat script for the diarrhea treatment.

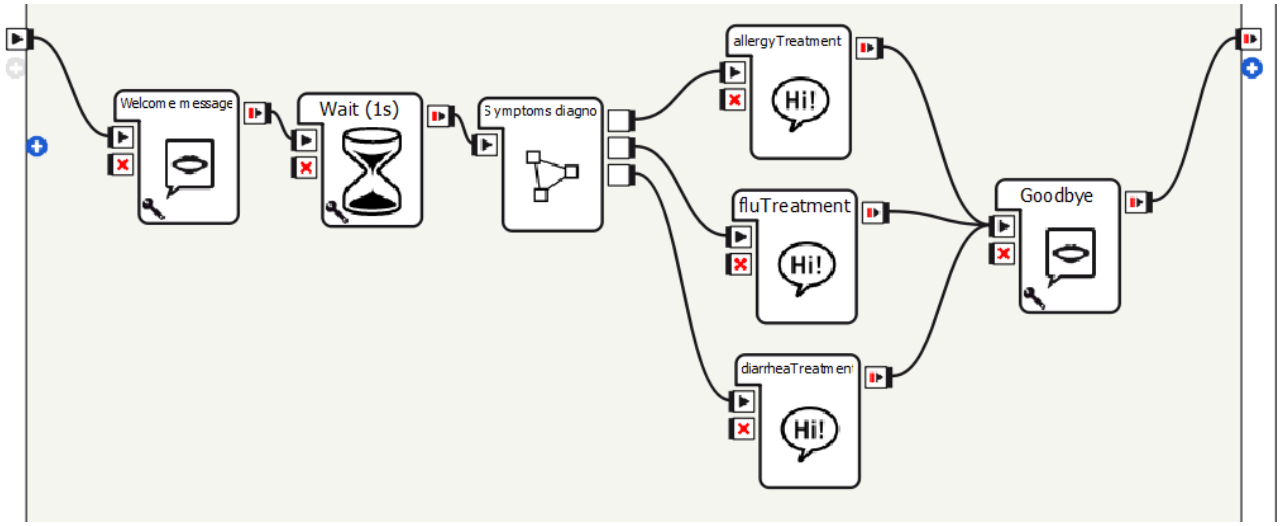


Figure 2.26 – The QiChat functions replacing the previous speech and listen boxes.

Creating limb movements

Introduction

To create/manipulate Nao's movement within Choregraphe, a timeline box must be created. A timeline box allows synchronous manipulation of Nao's limbs in time with other boxes such as speech. This allows Nao to talk and wave its hand to portray a more human-like robot when conversing, for example.

Implementing the wave movement

To implement a wave movement for now, a timeline box where it moves Nao's arm in a waving motion must be created. Right-clicking in the development area will show a dropdown box with an option to create a timeline box.

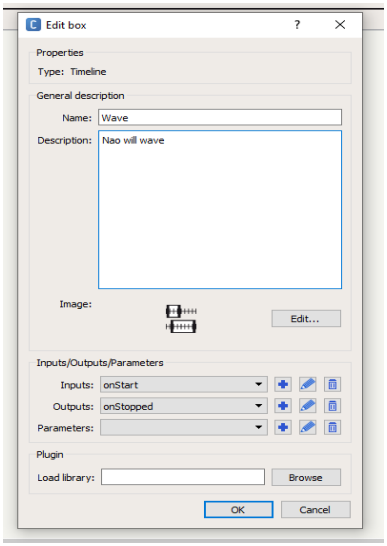


Figure 2.27 – The configuration of a timeline box.

Once created, it will show the timeline view to visually show the timeline of the limbs and joints movement. However, before using the timeline, Nao must be set on a pose (preferably the neutral position - Stand) and that autonomous life is turned off as shown in figure 2.28. This is because Nao may be in a different position, depending on the physical limb location of the Nao robot, and choosing the neutral position for it will ensure that limbs are as well. Turning off autonomous life will make motion development easier as it will stop Nao from swaying and returning its limb to the original pose once joints are being moved around.

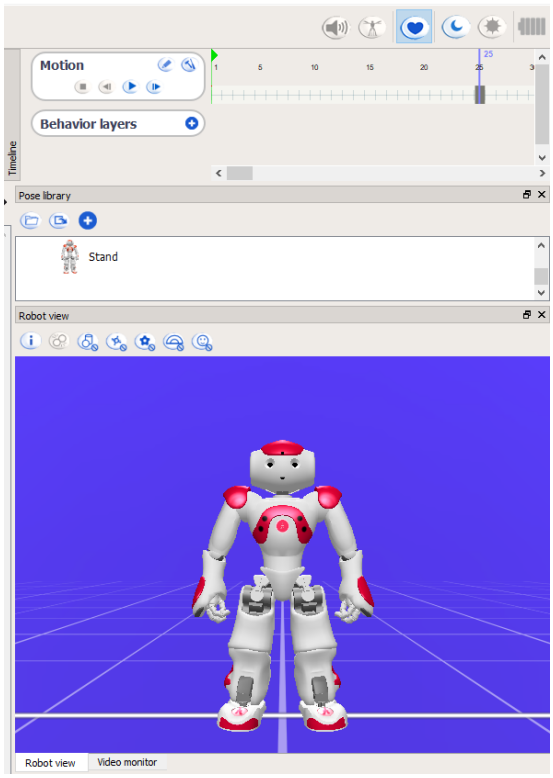


Figure 2.28 – The robot view with autonomous life turned off.

Moving and setting limbs

To move a limb or joint for Nao, joints must be stored on a keyframe within a specific time, in milliseconds, in the timeline. What this does is it tells where the limbs should be at that specific time, to which the IDE will then interpolate where the limbs should be between the time each keyframe to ensure that a smooth transition between keyframes. Due to interpolation between keyframes, each keyframe should not be close to each other with a big difference in limb position, as it will jolt the limb from one position to another in a short period of time which can damage the robot.

There are three ways to manipulate Nao's limbs: By dragging the limbs on the robot view; using the limb inspector and adjusting the position of the limb using degrees on a 3D space; and by moving Nao's physical arms to the desired position. By right-clicking on the timeline and selecting "Store

joints on a keyframe", there will be a choice for which limb will be set, or the whole body. In this instance, the arm must be set to be able to get Nao to wave. Once the arm is stored on a keyframe, the arm must be set on the desired position (See figure 2.29) for the start of the wave by clicking the desired limb to be moved, and dragging the built in 2D gyroscope (Seen in figure 2.29 as the red and blue circles). This can move the limbs in a drastic way to cover large distance, sacrificing accuracy, however this can be fine-tuned by using the inspector that pops up when a limb is clicked, and choosing a more defined position by inputting the desired angle (in degrees) on the corresponding text box, or using the slider.

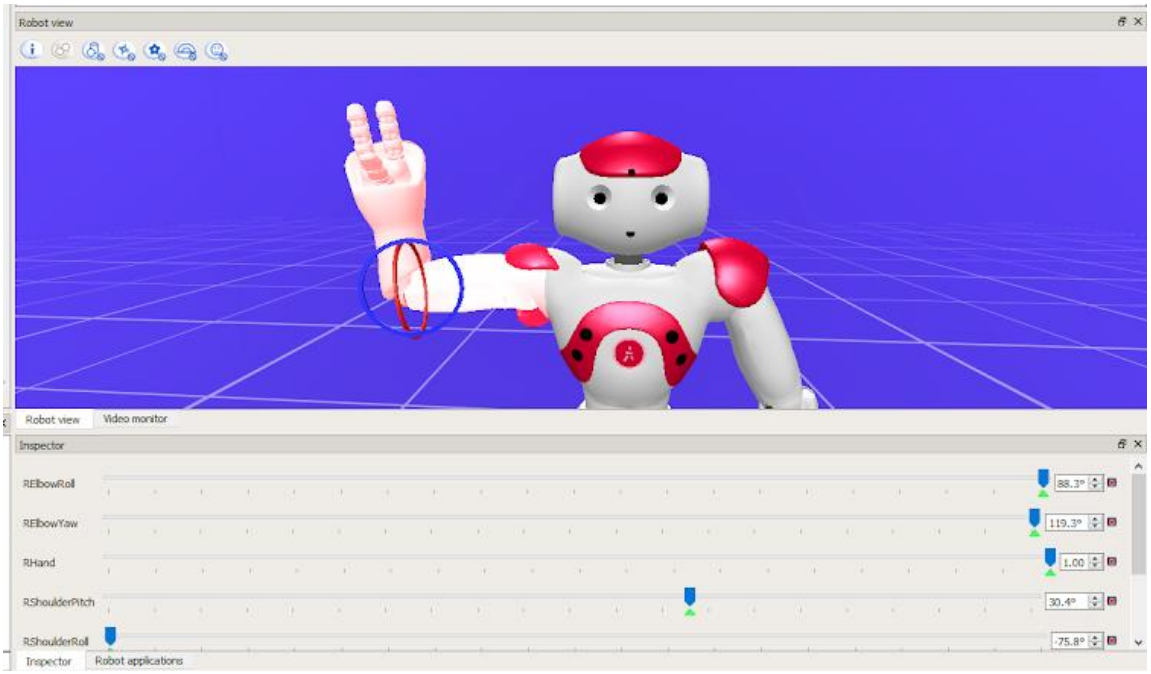


Figure 2.29

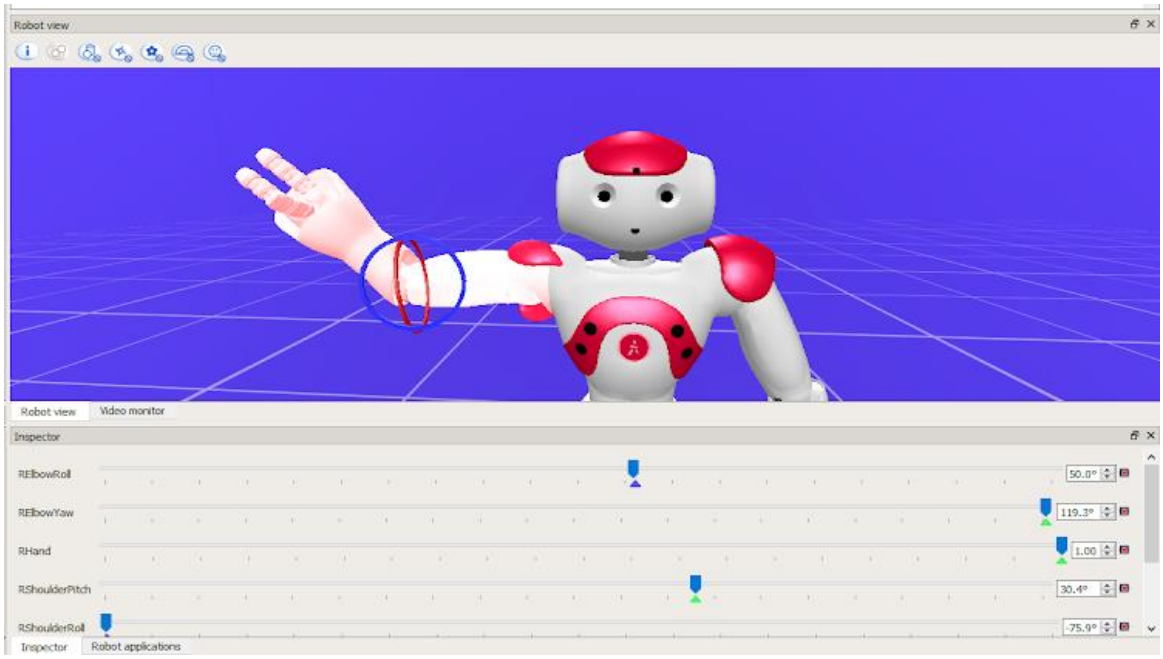


Figure 2.30

Since waving is a simple motion of moving the arm left and right, multiple keyframes of the arm must be set at different intervals of the wave (Starting position (Figure 2.29) and final position (Figure 2.30)), then Choregraphe will interpolate the movement between the positions. By creating multiple keyframes that include both the starting and final position, a waving movement can be simulated over time as shown in figure 2.31. This technique of setting the keyframes and the movement needed to be manipulated on the timeframe, and by manipulating the limbs, can achieve other, more complicated movements. This is used to create other movements such as thinking and nodding whilst listening to a patient, explaining to a patient while moving its arms and head, and waving the patient goodbye.

Figure 2.31

Generating a QR Code

Creating the QR Code

Generating a QR code will be handled using the QR code python library and will be developed on Sublime Text 3. The QR code library being used is the official python library that handles the creation of QR codes. The QRCode class will be used to create the QR object, which helps control the properties of the QR Code.

```
#import qrcode library
import qrcode
#import randomPatient
import RandomPatient

#initialise class
randPat = RandomPatient.patient()

#create qr object
qr = qrcode.QRCode(
    version=1,  #size of the QR Code. 1 - 40 with 1 being the smallest
    box_size=15, #size of the boxes in the qr code
    border=5    #thickness of the border
    )

#This QR code generator combines the two different wristband you can get in A&E into one. eg Known_Allergies would o

#Generate QR code with relevant information
data = 'First_Name: ' + randPat.randFirstName() + ', Last_Name: ' + randPat.randLastName() + ', DoB: ' + str(randPat
#add data to qr code
qr.add_data(data)
#fit the qr code dimensions automatically
qr.make(fit=True)
#save qr code to img
img = qr.make_image(fill='black', back_color='white')
#save image to folder
img.save('Patient_Health_Data.png')

```

Figure 2.32 – The code for creating a QR code on Sublime Text 3.

By using the QRCode class, different properties such as the size of the overall QR Code and boxes within the QR Code can be changed by changing the version and box size value respectively (Figure 2.32, Line 10 - 14). Once created, data can be inserted to the QR code. Since this is supposed to merge both wristbands currently used in the NHS, Name, DoB, Allergies, Risk of falling, Patient willing to receive blood, Blood type and NHS number will be stored in the QR Code. Once stored, the QR Code will be saved on the folder where the source code is created, where it can be viewed and scanned, as seen in figure 2.33.

Figure 2.33 – A generated QR code with randomly generated patient information.

Generating random patient data

Patient data is also randomised for completeness. To implement this, another python file is created where a random patient class is generated. This file will handle the class that randomises the patient data whenever a QR code is generated. This class is initialised on line 7 in figure 2.32.

To generate random information for the patients such as names, allergies and DoB a way to randomly choose from data in an array must be done. To do this, the random function is imported, which allows a choice to be randomly chosen. This is done using the random.choice(patFirstName) in figure 2.34, line 13 for example. This will choose from a selection of data within the array (In this case, patFirstName) and is replicated for the different patient data.

```python
#imports
import random
from random import seed
from random import randint
import datetime

#create patient class
class patient:
    #function randomly chooses from top 5 names from each gender
    def randFirstName(self):
        patFirstName = ['James', 'John', 'Robert', 'Michael', 'William', 'Mary', 'Patricia', 'Jennifer', 'Linda', 'Elizabeth']
        #choose name randomly
        firstName = random.choice(patFirstName)
        return firstName

    #function randomly chooses from top 10 last names from UK
    def randLastName(self):
        patLastName = ['Smith', 'Jones', 'Brown', 'Taylor', 'Wilson', 'Davies', 'Evans', 'Johnson', 'Thomas', 'Roberts']
        #choose last name randomly
        lastName = random.choice(patLastName)
        return lastName
```

Figure 3.34 – Generating random information for patients, to be stored in the QR code.

A difference in randomisation method is seen when randomising a patient's DoB. To randomise a patient's DoB, a minimum and max date is created to set the range of when the patient will be born. With this information, the range between the dates can be converted to days and used to create a random number (within the range) as days. This is then converted back to a date.

```python
    #function randomly creates date between end and start date
    def randDob(self):
        end_date = datetime.date(2002, 1, 1,) #set it to 2002 to make sure patient is voer 18
        start_date = datetime.date(1920, 1, 1,) #set it to 1920 to max age to 100 yrs old
        time_between_dates = end_date - start_date  #get time between dates
        days_between_dates = time_between_dates.days #convert difference in dates to days
        random_number_of_days = random.randrange(days_between_dates) #use days as range (between end and start date) from randomis
        random_date = start_date + datetime.timedelta(days=random_number_of_days) #create random date using the days
        return random_date
```

Figure 3.35 – Generating the random DoB on python.

Testing

Introduction

Unit and user testing will be used. Unit testing will be used because features added to the application can be tested after each integration of an iteration is completed, while user testing involves participants testing the application. The findings for each testing will be used to evaluate the usability and completeness of the application.

Conducting the testing

Unit testing

Unit testing will be conducted manually using a separate Choregraphe environment where each iteration will be tested at the end of the development phase. As this will be manual unit testing, it will be a type of white box testing to evaluate the desired output from a set of inputs. The input will depend on which functionality is tested, while the output will either be successful, by showing the desired output and reaching the onStopped node or fail by ending the application early and not reaching the onStopped Node. An example of manual unit testing can be seen in Figure 3.1, where the QiChat function, counter and IF boxes are tested to test whether the chat can correctly identify the illness when given several symptoms. Unit testing is also done for the QR code functionality, done in Sublime Text 3. Test tables will be used to document the testing and can be seen in Appendix D.

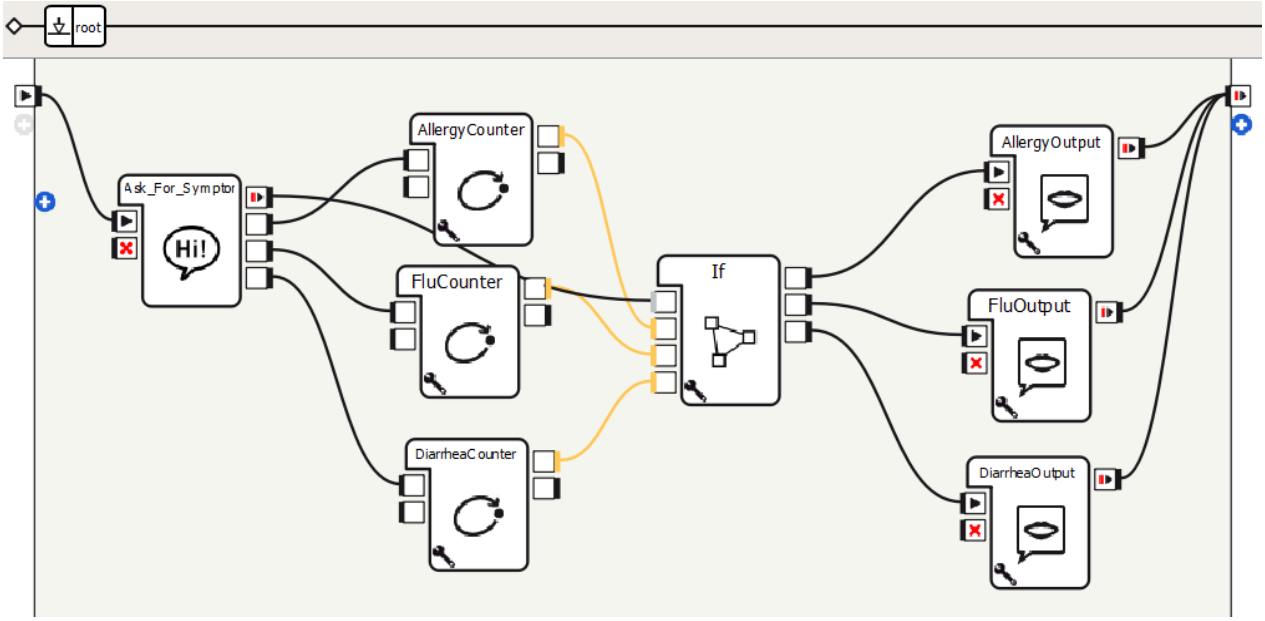


Figure 3.1 – Manual Unit Testing for the diagnosing symptoms.

Conclusion

According to the testing done seen in Appendix D, the tests were successful in getting the desired results, for the most part. However, test 2 failed, as Nao was programmed to listen for

“Antihistamines”, and any other input will not continue down the dialogue tree. This was later fixed to branch out if the user is not allergic to Antihistamines.

User testing

User testing will involve real users to test the final deliverable. User testing will be split into three different parts: Doing the task, Complete the SUS (System Usability Scale) and verbal feedback. The first task involves the user carrying out a list of tasks (Appendix E) on the application. Each user will be assigned an illness at random, where they will be asked to act as a patient with symptoms relating to their assigned illness. They will be asked to say that they are not allergic to any medication, while a second run at the application will ask the user to say that they have an allergy relating to the medication used to treat their illness. During the task, the users were observed, and their actions noted down as they interact with the application. Once this task is completed, the users will then complete the SUS questionnaire (Appendix G). Lastly, the users will give verbal feedback about the application, where their opinions will be noted down and reviewed.

Appendix F shows the consent form given to the participants. This form contains information relating the testing that will be carried out and what will happen when the data is collected. There will be two consent forms that contain the participants signature agreeing to consent, where one will be kept by the participant as a copy.

Conclusion

The findings of the SUS (Appendix H) show that the participants found that the application was received well, according to the classification on Figure 3.2. However, before the participants used the application, they were told beforehand that the intended application was supposed to be on a robot, where speech communication was possible. This can affect how the participants have answered the SUS.

SUS Score	Grade	Adjective Rating
> 80.3	A	Excellent
68 – 80.3	B	Good
68	C	Okay
51 – 68	D	Poor
< 51	F	Awful

Figure 3.2 – SUS score interpretation. Reprinted from *How to Measure Product Usability with System Usability Scale (SUS) Score* from *UX planet* by H. Alathas, 2018, Retrieved April 22, 2020, from https://uxplanet.org/how-to-measure-product-usability-with-the-system-usability-scale-sus-score-69f3875b858f

The information collected from observing the participants are mostly positive, and participants are happy that Nao can correctly diagnose them with the correct illness. Overall, they believe that Nao has good conversational skills relating to diagnosing and treating illness. However, when they tried to say a symptom that is not listed for Nao to listen for, participants were confused as Nao would not continue with the dialogue tree. In this case, participants who experienced this needed outside input to enable them to continue the flow of conversation.

Positive feedback of the application:

- None of the participants found the application too hard to use.
- All the participants, on the second run, like that Nao still responds when they respond differently from before.
- Some participants found that the hand movements and gestures to be a reassuring feature when talking to Nao.
- All the participants found that Nao was clear and concise with its questions.
- Most participants like that Nao can give advice on how to treat their illness.

Negative feedback of the application:

- The participants want a way for Nao to catch words that Nao is not listening for, and to act accordingly.
- Most participants did not know or acknowledge Nao's gestures.
- Some participants did not like that they were talking to a robot and felt uncomfortable.
- Some participants did not like that Nao would only ask for one input from them at a time, where they found it too time consuming and sometimes frustrating.

Critical Evaluation

Introduction

The evaluation will critically analyse the stages of the project to highlight the strengths and weaknesses of the approaches conducted throughout the life cycle of the project. In addition, a personal reflection will be conducted to explore the strength and weaknesses of my individual performance throughout the project.

Critical analysis of the project

Specification and requirements

The specification and requirements of the project was an overall strength. This is because the idea of the project was an exciting and relatable idea to embark on, as well as mentally stimulating. By never having worked with robots or tools that are required for the project, it allowed the development of skills that were not used before such as being able to learn and use a new programming language - Python, and to use hardware such as Nao, which was not possible throughout my studies. This has overall impacted on how to personally develop as a software engineer and as an individual. This idea of learning and using new tools and equipment that was once not available, coupled with creating an application that is physically tangible fuelled the motivation to develop the project which in the end, benefitted the overall development of the final application.

However, while trying new ideas and tools, a problem with ambitiousness and over-engineering occurred throughout the project. Ambition was a problem as there was no previous knowledge on how to implement the idea for the project, which resulted in trying to design a more complicated application than what was done in the project. Ideas such as using AI and machine learning in tandem with Nao to fully replace nurses in the assessment room was too ambitious and time consuming, in addition to the complexity of the project if such features were implemented. The design was also in danger of over-engineering which can be evident when implementing the QiChat system, as new ideas were always considered to improve the application, which in hindsight was

troublesome when using the iterative spiral model of development. This often led to missing deadlines and increasing the development time of the project. If the project was done again, Trello must be used as early into the project life-cycle as possible to ensure that the project scope doesn't get too large and to stick with the original plans to further avoid the time problems that occurred at the end of development.

While weaknesses in ambitiousness, over-engineering and time management was apparent, the use of tools such as Trello was a great way to keep the project within scope. By using Trello, keeping deadlines, and ensuring that there were not too many features that kept getting added to the project helped with the completion of the project. In addition, the use of the spiral model helped more than it did hinder, as it helped with ensuring that features were added throughout development, albeit at smaller capacity compared to other development methods.

Research and planning

The research of the project was very interesting and thought provoking, as a research aim was to look if Nao is capable of being used to develop many applications with the use of Choregraphe. Throughout the research, it was shown that there are Nao-like robots used in the medical field, but at a lower capacity. The research also gave a perspective on how robots are used in the real world, which helps create an idea of where the project fits in and helped push the idea of the need to use robots within the medical field. In the end, the research was a massive strength as it helped put into perspective what can and can't be done in the medical field with the use of robots, which further tightens the scope of the possible capabilities of what Nao can do for the project.

Furthermore, the choice of using the spiral model for software development is considered a strength, as it features a high amount of risk analysis, which when paired with the risk analysis done beforehand, enabled to mitigate the possible loss of time when a problem occurred during development. The spiral model also enabled a strong way to control documentation, where documentation of the development can be written in parallel to the development of features to ensure a more complete and accurate documentation, compared to when being done after development. The spiral model also allowed the integration of the QiChat at a later stage of development, which was essential as it allowed Nao to use a more sophisticated way of speaking and speech recognition. By using the spiral model with Trello, Trello was able to clearly show what features need to be implemented, tested, or completed which compliments well into the use of the spiral models use of iteration when implementing new features. In the end, this allowed a more organised process during development which kept focus to within the project scope.

However, a weakness within the research was the failure to research advanced functionality for Nao. This meant that implementing QiChat was done without much research done. This further slowed down the development progress as time was used into researching the knowledge needed to implement QiChat into the application. In order to improve on this in the future, more time must be spent on analysing the capabilities of the hardware, and build upon the strengths of Nao while mitigating what can't be done, rather than trying to research robots and apply their features on Nao without knowledge of Nao's capabilities.

Development

A massive strength during development was being able to implement features on a feature-by-feature basis due to the low coupling allowed on the Choregraphe IDE. This feature with the spiral model allowed for an easy transition between design and development, as designing features for the application can be done after each phase during each iteration, and vice versa. This allowed a level of flexibility that helped with plugging the shortcomings of research, where for example, QiChat was implemented after the first iteration of speech was already developed, where the use of other development models such as the waterfall wouldn't be capable of accommodating such change, using the right model during development helped with implementing new features.

Another strength during development was the integration of the diagnosis. Participant feedback was very happy that Nao was able to correctly identify what the participant's illness was, and to correctly give them the correct medication/treatment. For the diagnosis and treatment feature to be a massive success with positive participant feedback, the success would achieve one of the aims of the project. This was in part, due to the integration of the QiChat features, as it allowed Nao to catch different ways that users would speak to it, enabling a flexibility of inputs that Nao can take. The implementation also allowed the development to be easier while lowering the complexity as it reduces the number of boxes needed to implement speech and listening into one, helping with the debugging and maintenance on the long term.

The QR code is also a strength in the development. The QR code was a functional outcome that was able to replicate the data of what normal medical wristbands would contain in the NHS. However, the practically of the QR code was not tried or tested due to not being able to have Nao at my disposal due to the Covid-19 Outbreak, therefore I was not able to test if Nao is able to successfully read them. However, reading the QR code using a QR code reader app does replicate what Nao should be able to do.

A weakness, however, concerns the spiral model. Using the spiral model can also be time consuming and can increase the project scope. This is because another feature that was supposed to be implemented for Nao was the behavioural layers, which was supposed to fluidly integrate the limb movements of Nao with its speech capabilities. While not necessary, it was supposed to bring the movement and speech together in the same way that QiChat was supposed to bring the speaking and listening capabilities of Nao into one. This was not possible due to the time constraints of the project. With this in mind, if the project was to be done again, research must be improved to further acknowledge the capabilities of Nao, while maybe a change in model should be considered during development as a weakness of the spiral model is that it is not suitable for smaller projects, which can result in time costs.

One major weakness was found in the QiChat feature, which was found during the testing of the application. The testers found out that Nao will not respond to the user if Nao was not scripted to listen to. This led to confusion with the user as they believed that Nao was not working properly. This is a major problem as there was no way for Nao to cover all of the ways that a user can speak to it, therefore I found it hard to fully implement a way to catch all possible was a user can answer Nao. If the project was to be done again, I would ensure that there is a way to have a feature that catches all possible ways that Nao should listen for, to be able to respond to all types of user input.

Another feature during the development that is deemed a weakness would be the limb movements. While most participants did not notice the movements and gestures from Nao, it should be improved to be more interactive. Arm movements and head nods may be too uninteresting for people to notice, and therefore is not that important of a feature, while fully omitting movement would be too robotic and unnatural for humans to talk to. A more interactive way should have been developed for Nao to interact with users. In the future, if the project was done again the movement will be changed to include physical interaction with the users with the use of the handshakes and eye tracking to further enhance Nao's interaction capabilities.

Testing

The testing conducted during the project was a massive success. The testing allowed an outside perspective that is different from a developer with the use of the SUS. The SUS was able to give a metric on how usable the application was from a user's perspective, but to also highlight the shortcoming of the application which would not have been as clear to see from a developer's side of view. The use of unit testing was also complimented with the use of the spiral model as it allowed for parallel testing of individual features during development. This further streamlined the development of the overall project.

However, a massive weakness to the user testing was that most of the participants were with a medical background, with little to no knowledge of using robots or software systems. This led to a more biased feedback, where they were impressed with the diagnosis and treatment of the illnesses, but the feedback regarding the system was not as detailed or varied as hoped. Another weakness to this is that there is not a variety of backgrounds within the ranks of the participants, which skewed the results once again. In hindsight, a more varied population of participants should have been used to test the usability of the application rather than to use participants to test whether certain features work best.

Lastly, the inclusion of state transition testing should have been used for the QiChat feature. This is essential for testing out the QiChat system due to its dialogue tree. Since this was not tested, there can be branches that may lead to erroneous or incomplete branches on all the dialogue boxes used in the application.

Individual evaluation

Throughout the development of the application, significant improvements on personal development and growth was made. This is because the project used new tools and hardware that I was unfamiliar with before the project. These tools and skills were learnt and developed throughout the project by using research and through practical application. This was further enhanced with the idea of teaching yourself the skills and tools, which led to learning on my own pace, ensuring that I have a good understanding of the tools and skills needed to implement them, which contrasts the educational journey in University where teachers would give you guidance and work to go through throughout the years.

A big improvement in ability was project management skills, which was enhanced with the use of project management tools such as Trello. I have never used Trello before the project, but after using the tool it majorly improved my ability to manage the project, keep within the range of the scope, and to focus on the features laid out ahead of me instead of adding features on the go. Time management skills were also heavily developed as the project showcased an assignment that was present throughout the whole year, which disciplined me to work on the project in between assignments. For example, at the start of the year I was able to fit some project work during free time but in the second semester, where there was a multitude of assignments due, the project enabled me to practice my time management skills as I had to develop a high level of discipline to be able to not drown in work, even when some deadlines have been passed.

Ethics was also questioned and developed throughout the project, as the project was involving health. The main problem was the question of wanting to have a robot examine and give out medical advice, compared to a human one. This is a concerning thought as most people would not want to have a robot do what a human can, who has years of experience. This was evident with the research done on robots, where it can be interpolated that the robots used within industries was for automation, and only replaced jobs where it was a detriment to human health, such as car manufacturing, therefore robots wouldn't be as well received in the health sector. However, further research however did counter that there is a need for robots, albeit in more restrained manner such as nursing homes or for use with children.

Technical skills were also developed. Python was unfamiliar to me, as it was not taught on the syllabus, therefore having the project focus on the language that I was unfamiliar with was daunting, but exciting even though Choregraphe can be programmed using C++, a language I was very comfortable with. Hardware was also another shortcoming when starting the project, as the only hardware that I had experience with was during the first year when we were tasked with developing a basic robot made out of Legos to do an obstacle course. I chose to sit in on the "Programming things" optional module lectures, where I was able to further develop my knowledge and confidence with using robots with the software. However, I did have, albeit limited, experience with using microcomputers such as the raspberry Pi and Arduino due to my placement.

References

Alathas, H. (2018, November 19). How to Measure Product Usability with the System Usability Scale (SUS) Score. *UX Planet*. https://uxplanet.org/how-to-measure-product-usability-with-the-system-usability-scale-sus-score-69f3875b858f

Aldebaran. (n.d.a). What is Choregraphe. Retrieved April 12, 2020, from http://doc.aldebaran.com/2-1/software/choregraphe/choregraphe_overview.html

Aldebaran. (n.d.b). NAOqi Framework. Retrieved April 12, 202, from http://doc.aldebaran.com/1-14/dev/naoqi/index.html#naoqi-framework-overview

Alshamrani, A., & Bahattab, A. (2015). *A comparison between three SDLC models Waterfall model, Spiral Model, and Incremental/Iterative Model*. International Journal of Computer Science Issues. (pp. 106 - 111).

Bennet, J.C., Bohoris, G.A., Aspinwall, E.M., Hall, R.C. (1996). *Risk analysis techniques and their application to software development*. Elsevier B.V.

Bischofberger, W.R., & Pomberger, G., (2012). *Prototyping-Oriented Software Development: Concepts and Tools*. Springer-Verlag.

Buchan, J., Charlesworth, A., Gershlick, B., & Seccombe, I. (2019*). A critical moment: NHS staffing trends, retention and attrition*. The Health Foundation.

Dvorkin, M., & Bharadwaj, A., (2019. November 7). Which Countries and Industries Use the Most Robots?. *Federal Reserve Bank of St. Louis*. https://www.stlouisfed.org/on-the-economy/2019/november/robots-affecting-local-labor-markets

El Emam, K., & Koru, A.G., (2008). *A replicated survey of IT software project failures*. IEEE Computer Society. (pp. 84-90).

Ford, S. (2018, December 20). NHS will still be short of nurses in five years, warns regulator chief. *Nursing Times*. https://www.nursingtimes.net/news/workforce/nhs-will-still-be-short-of-nurses-in-five-years-warns-regulator-chief-20-12-2018/

Francis, S. (2017, September 12). Paris city hall gives jobs to Pepper and Nao robots. *Robotics and Automation.* https://roboticsandautomationnews.com/2017/09/12/paris-authorities-employ-pepper-and-nao-robots-at-city-hall/14052/

Hervey, T., McCloskey, S. (2018). Brexit and the NHS. *The UK in a Changing Europe.* https://ukandeu.ac.uk/wp-content/uploads/2018/03/Brexit-and-the-NHS-.pdf. (pp. 8 – 10)

Kamps, H.J. (2016, September 11). The Autism Solutions bot helps autistic kids. *Tech Crunch*. https://techcrunch.com/2016/09/11/autism-solutions/

Learner, S. (2019, April 25). Dear is stifling potential use of robots in UK care homes, says AI researcher. *Home care.* https://www.homecare.co.uk/news/article.cfm/id/1609000/fear-stifling-potential-robots-uk-care-homes-AI

Nursing and Midwifery Council. (2019). Annual Report and Accounts 2018-2019 and Strategic plan 2019-2020. *Nursing and Midwifery Council.*

Ockey, M. (2015). *MEDi the Robot*. Business in Calgary.

ROBOTICS OF LODON. (n.d) Pepper the receptionist. Retrieved April 12, 2020, from https://www.robotsoflondon.co.uk/pepper-the-receptionist

Robots to transform Japan's social care by 2020. (n.d). Retrieved from https://internetofbusiness.com/robots-japan-social-care/

Sheward, L., Hunt, J., Hagen, S., Macleod, M., & Ball, J. (2004). *The relationship between UK hospital nurse staffing and emotional exhaustion and job dissatisfaction*. Journal of Nursing Management.

Why You Need to Wear a Wristband Whilst in Hospital. (2016). Retrieved from https://www.nlg.nhs.uk/content/uploads/2014/04/IFP-0617-1.pdf

Appendices

A. Retrieved from dummies.com:https://www.dummies.com/careers/project-management/assessing-the-likelihood-of-a-risk-in-your-project/ - In the website, the page says that when determining the likelihood of different objective risks, such as past project reports:

"- Consider previous experience with similar projects.

- Consider as many similar situations as possible.

- Keep in mind that the more similar situations you consider, the more confidence you can have in your conclusion"

And when counting on personal opinions:

"- Define the category name as clearly as possible.

- Consider the opinions of as many people as possible.

- Be sure the projects your respondents have worked on are truly similar to yours.

- Don't allow people to discuss their estimates with each other before they share them with you. After they've submitted their initial estimates to you, consider having the people discuss their reasons for their estimates with one another and then asking them whether they want to revise their estimates."

B. Retrieved from dummies.com:https://www.dummies.com/careers/project-management/assessing-the-likelihood-of-a-risk-in-your-project/ - To calculate the risk, I have used the follow equation from the web page:

"ask them to rate the likelihood of a specific risk as high, medium, or low. Suppose six people choose high, two choose medium, and two choose low. You may then develop your estimate of the likelihood by assigning values of 3, 2, and 1, to high, medium, and low, respectively, and determining the weighted average of the responses as follows:

$(6 \times 3) + (2 \times 2) + (2 \times 1) = (18 + 4 + 2) / 10 = 2.4$

This formula suggests the risk has medium to high likelihood of occurring (because 2.4 is between 2 and 3)"

With this information, the equation to calculate risk is:

Weighted response to risk = $[(P_H \times W_H) + (P_M \times W_m) + (P_L \times W_L)]/P_T$

Where:

Weight of high risk = $W_h = 3$

Weight of Medium risk = $M_M = 2$

Weight of Low risk = $W_L = 1$

Total people asked = $P_T = 10$

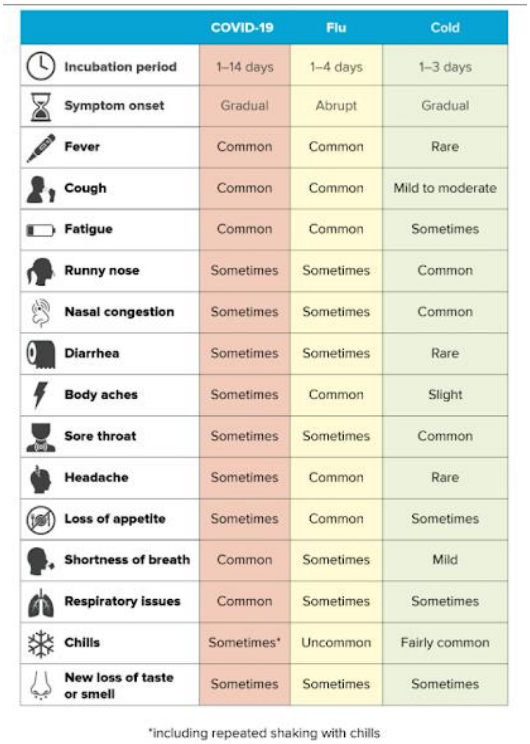
People with opinion of risk being high = O_H

People with opinion of risk being Medium = O_M

People with opinion of risk being Low = O_L

C. Image taken from https://www.medicalnewstoday.com/articles/coronavirus-vs-flu#symptoms.

A comparison of symptoms between the flu, cold and covid-19 disease.



	COVID-19	Flu	Cold
Incubation period	1–14 days	1–4 days	1–3 days
Symptom onset	Gradual	Abrupt	Gradual
Fever	Common	Common	Rare
Cough	Common	Common	Mild to moderate
Fatigue	Common	Common	Sometimes
Runny nose	Sometimes	Sometimes	Common
Nasal congestion	Sometimes	Sometimes	Common
Diarrhea	Sometimes	Sometimes	Rare
Body aches	Sometimes	Common	Slight
Sore throat	Sometimes	Sometimes	Common
Headache	Sometimes	Common	Rare
Loss of appetite	Sometimes	Common	Sometimes
Shortness of breath	Common	Sometimes	Mild
Respiratory issues	Common	Sometimes	Sometimes
Chills	Sometimes*	Uncommon	Fairly common
New loss of taste or smell	Sometimes	Sometimes	Sometimes

*including repeated shaking with chills

D. Test table for unit testing

Test number	Test name	Description	Test data	Expected result	Actual result	Passed
1	Testing symptoms diagnosis functionality	To test how well picked up keywords are assigned to their desired counters, resulting in the right diagnosis	Keywords such as: “Puffy eyes”, “Sneezing”, “Stomach cramps”, “Loose stool”, “Headache ”	Nao will correctly diagnose the illness, according to the input given.	Nao is able to correctly count how many symptoms are given, by calculating which illness has the most symptoms.	Yes
2	Testing allergy treatment functionality	To test whether Nao can correctly pick up keywords	Keywords such as: “Antihistamines”, “Yes”,	Nao will react correctly, according to the input	Nao was unable to continue the diagnosis after asking	No

		and is able to continue to the end of the conversation.	“No”, “I don't know”, “Skin rash”, “Blocked nose”	given.	the patient if they are allergic to medication.	
3	Testing flu treatment functionality	To test whether Nao can correctly pick up keywords and is able to continue to the end of the conversation.	Keywords such as: “Ibuprofen”, “Yes”, “No”, “I don't know”	Nao will react correctly, according to the input given.	Nao is able to correctly react and respond according to the input given.	Yes
4	Testing diarrhea treatment functionality	To test whether Nao can correctly pick up keywords and is able to continue to the end of the conversation.	Keywords such as: “Yes”, “No”, “I don't know”	Nao will react correctly, according to the input given.	Nao is able to correctly react and respond according to the input given.	Yes
5	Testing the creation of a QR code	To test whether the code creates a QR Code	N/A	The program should generate a QR Code	The QR code is created	Yes
6	Testing the information inside the QR code	Test to see if the created QR code contains the correct information	The QR Code	The QR code should contain the correct information and layout	The QR Code did contain the correct information and layout of data	Yes

E. The list of task for the users to conduct on the application:

Task number	Task
1	Start the application.

2	When Nao asks for your symptoms, give Nao 5 symptoms relating to your illness.
3	Give Nao 2 symptoms that are related to the two other illnesses that you are not assigned to.
4.1	When Nao asks if you have an allergy to medication, reply with a negative. If this is the second run, go to task 4.2.
4.2	When Nao asks if you have an allergy to medication, reply with a positive.

F. The consent form participants will read:

Consent form for use of final year project data in

Robotic assistance for patient interaction in Minor Injury Unit (MIU) assessment rooms

The aim of this stage of the research is to evaluate the projects usability for robotic assistance in an assessment room environment. You have been asked to participate in the research randomly. What will be required of you is to carry out set of tasks using the final year project that has been developed, this will be observed by Kenneth Alegria.

After the tasks have been complete there will be a questionnaire and time to share about your experiences that you encountered during the evaluation. This evaluation should take between 15 to 30 minutes but can be exceeded. This evaluation will be taking place in private spaces such as home space or a meeting room. The benefits for being involved in the evaluation will be possible insight into how software processes can be improved and specialized for patients, allowing robots to be more understanding and accessible to use. If during the evaluation, for any reason, you would wish to stop the evaluation, this is your right. All data that was collected during the evaluation will then be destroyed and will no longer be part of the findings of the research project. If after the evaluation, for any reason, you wish your data not to be used in the research, this is your right. At the end of the evaluation you can ask questions about the evaluation and research or offer any feedback about the research. Kenneth Alegria is responsible of the data 12 months after the research has been conducted and will be deleted after this time limit.

In each of these publications and presentations we will illustrate the project findings using descriptions of the activities that took place and quotations. These quotations may be presented anonymously through using pseudonyms (e.g. "Jane Smith, an artist said…", "Bob, a teacher said….").

The University undertakes research as part of its function for the community under its legal status. Data protection allows us to use personal data for research with appropriate safeguards in place under the legal basis of public tasks that are in the public interest. However, all University research is reviewed to ensure that participants are treated appropriately, and their rights respected. Further information at: https://www.shu.ac.uk/research/ethics-integrity-and-practice

The findings of this research will be shared with researchers at Sheffield Hallam University and the findings may also be published. This may be in the form of academic papers, reports, presentations, and talks. If you would like to receive a copy of the report by email. If you have any concerns, or if you want to ask more questions about the research please contact:

Kenneth Alegria,

26 Leadmill Rd, Leadmill Point, Flat 52 Sheffield, South Yorkshire, S1 4SD

Email: b6035314@my.shu.ac.uk

Participants signature: Researcher's signature:

G. The SUS form:

The answers are weighted from 1 to 5, starting from "strongly disagree" at 1, to "strongly agree" at 5.

System Usability Scale (SUS)

This is a standard questionnaire that measures the overall usability of a system. Please select the answer that best expresses how you feel about each statement after using the system today.

	Strongly Disagree	Somewhat Disagree	Neutral	Somewhat Agree	Strongly Agree
1. I think I would like to use this application frequently.					
2. I found the application unnecessarily complex.					
3. I thought the application was easy to use.					
4. I think that I would need the support of a technical person to be able to use this application.					
5. I found the various functions in this tool were well integrated.					
6. I thought the app was too inconsistent.					
7. I would imagine that most people would learn to use this application very quickly.					
8. I found the application very cumbersome to use.					
9. I felt very confident using the application.					
10. I needed to learn a lot of things before I could get going with this application.					

H. SUS results from the 10 participants:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	**SUS Calculation**															
2																
3	**Participant**	**q1**	**q2**	**q3**	**q4**	**q5**	**q6**	**q7**	**q8**	**q9**	**q10**	**SUS Score**				
4	1	4	2	4	2	3	2	3	3	4	2	**67.5**			Average	69.3
5	2	4	2	4	2	4	1	4	2	4	2	**77.5**				
6	3	4	2	3	3	4	2	4	2	3	3	**65.0**				
7	4	3	2	4	2	3	2	3	2	4	2	**67.5**				
8	5	4	3	4	2	4	2	4	2	4	2	**72.5**				
9	6	3	2	4	3	4	2	3	3	3	3	**60.0**				
10	7	4	2	4	2	4	2	4	2	4	2	**75.0**				
11	8	4	3	4	2	4	2	3	2	4	2	**70.0**				
12	9	4	2	4	2	4	2	4	2	3	3	**70.0**				
13	10	3	2	2	2	4	2	4	2	3	1	**67.5**				
14																

The way that SUS is calculated is by using the instructions provided by:

https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/

The calculation is done using Microsoft Excel

Here, SUS is calculated by:

- For each of the odd numbered questions, subtract 1 from the score.
- For each of the even numbered questions, subtract their value from 5.
- Take these new values which you have found, and add up the total score. Then multiply this by 2.5.

PERSONAS

I. Persona - Vick Bollard

Persona	
Name	Vicky Bollard
Job title	Unemployed - Retired
User Type	Patient

Demographics	• 67 Years Old
	• Married
	• Lives in a semi-detached house with a retired 72-year-old husband on the outskirts of Leeds.
	• Has two children:
	  • Son, 29 Years old
	  • Daughter, 21 Years old
	• Has 2 A levels and 1 BTEC
	• Was a teacher before retiring

Biography	Vicky is a woman living her retired life with her husband in the suburbs of Leeds. After graduating from college, she decided to pursue a career in education, more specifically secondary school education. As the years have gone by, she has increased back and joint problems due to her profession asking her to stand up for a long period of time paired with her above average weight. At her age, she is not well acquainted with technology, especially emerging technology such as robots and AI as she did not get a lot of exposure throughout her life with technology until the emergence of smart phones. However, she does use a computer both at home and at work and knows basic computer skills to be able to competently work in her past job. If she is having any problems with technology, she asks for help from her two children. She believes that she doesn't need technology with her day to day life and keeps herself to a simple life during her retirement. Her day to day life involves cleaning the house, cooking meals for herself and her husband and occasionally for her children when they visit, garden and tend to her vegetables and at times go out to shop or go out to the pub to relax after a long day. Her hobbies consist of baking, knitting and gardening; and occasionally enter local competitions for her hobbies.
Environment	Due to her lack of experience with technology, she does not know about the latest technology and is slightly afraid of what it will do to her way of life. She is willing to use smartphones, albeit ineffectively, to communicate with friends and family but tends to keep away from such things. With her age and her chronic back pains and joint swell, she is diagnosed with osteoarthritis[3] just before her retirement and is what pushed her to join her husband in retirement.

	During her time in retirement, she has increased difficulty with walking and keeping her posture free of pain without medication.
Quote	“I love my job so much but having the onset of osteoarthritis has sadly pushed me to go into retirement.”

J. Persona - Alan McKay

Persona	
Name	Alan McKay
Job title	Head Technician
User Type	Maintenance and service
Demographics	• 34 Years old • Lives in a flat in Leeds with his fiancé, a 4-year-old and dog. • Has 3 A levels – One in ICT • Technology enthusiast • Has worked with the hospital in Leeds for 10 years and is well versed with basic medical procedure and paperwork. His fiancé is a nurse in the hospital he works in.

Biography	Alan is a middle-aged man with experience in ICT and the medical environment. Due to his years of exposure in the medical field, he knows the ins and outs of the technical needs of the hospital. He is one of many younger staff within the hospital who is trying to push for more use of technology within the medical world. A big reason as to why he wants to push technology into the medical world is because of his fiancé. Every day he sees his fiancé stressed out after work due to her workload and the lack of staff, especially nurses, in the NHS. He finds that his fiancé is more moody and tired after work and that their 4-year-old child does not help with keeping stress at home low. This has put a strain on their relationship but also his fiancés work performance as she gets less sleep due to her being called into work due to lack of staff. His idea for the hospital to help with nurses and her fiancé is to introduce a robot into the workplace to help reduce the workload and stress for staff. He intends to target the minor injury assessment rooms as he knows how long waiting in the A&E can last. He knows his way around technology and frequently plays games a lot. One night one of his friends talked about how new technology in robotics has increased in past years and is now being used as receptionist in offices, therefore he has found it a great idea to see if they can be used to help nurses within the A&E in interacting with patients with diagnosis and advice that nurses would normally be tasked with within the assessment rooms.

Environment	Due to Alan having an A level in ICT but no further education past A levels, he is competent with technology with intermediate experience with the python language although he doesn't have experience with working with robots.
Quote	"The NHS needs to adopt a way to combat the shortage of staff, and technology is the way forward."

K. Persona - Stacy Porter

Persona	
Name	Stacy Porter
Job title	Nurse
User Type	Assistant to Robot
Demographics	• 28 Years old • Lives in a house with 3 other young adults • Has a degree in nursing from University of Leeds • Single

Biography	Stacy has recently graduated from university of Leeds with a degree in nursing. She has taken a gap year after graduation and has recently started working in the hospital in Leeds. She is living in a house with her past flatmates from university to help with her rent, as well as ensuring that she would have a social life outside of work knowing that she will be working hard during her work hours due to NHS understaffing. She lives a frugal life, which she learned from university, as she comes from a poor family and is the only one in her family that has achieved a degree. Because of this, she is the leading support for her family in terms of finance. In turn, she requests more overtime and spends a lot of time, including the weekends, at work to get more hours. Due to overworking, low staff and the general stress she gets at work, when she gets home she is not in the right mood to be going out with her friends on the weekends and she feels like she is getting left out and finds that she is not enjoying her young adult life and because of her commitment to her family and their support during her university days, she will not compromise her hours for free time. At work, she and her fellow nurses are working around the clock and are instructed to keep their phone on them at all times just in case they get called in, which is often for young nurses such as herself. Stacy feels like they don't get enough breaks at work which she feels is warranted due to the nature of their job but fears that fatigue will lead to mistakes which is critical to how well they do their job and wants a way to ensure that nurses in her hospital are at their best when working.

Environment	Stacy has good communication skills built up through her time in the hospital, especially when communicating with patients. She can use technology and understands the need for them in the future and welcomes a technological change for her field of work.
Quote	“I can’t believe the NHS is so short staffed. If we want to stay afloat, we need all the help we can get.”

Scenarios

L. Scenario - Vicky Bollard

Vicky gets up early in the morning to cook breakfast for herself and her husband. Afterwards she would tend to her plants in her garden and harvest any vegetables ready. She would then use those vegetables for lunch and dinner later in the day.

Afterwards she would maintain the house to a high standard by cleaning and tidying the house before lunchtime. This activity requires her to be active throughout her day to day life which affects her physical well-being as continuous activity with osteoarthritis is detrimental to her joints and back. Because of this, Vicky must take medication for her pain and stiffness twice a day, one before lunch and another before dinner. She would take her first medication in the afternoon to help Vicky cope with her pain during the day.

In the afternoon, she would continue her hobby by knitting jumpers for her children and a few woolly hats for charity and her knitting group where she would compete with other retired citizens in her local village. However, throughout the years this has been a more challenging task as she has lost her dexterity on her fingers due to her type of arthritis. When she has more free time, she would go to her local GP to get more prescriptions for her medication to help combat her daily pain, however this is not often the case due to the long waiting times for her local GP. At times when her pain flares up more than normal, she becomes paranoid and goes to the A&E minor injuries waiting room to get a more serious check-up however the problem still persists for her in that she has to wait a long period of time to get a check-up and prescription. At this time, her pain is usually unbearable and when finally meeting a nurse for an assessment, she would have a temper due to wanting to ease her pain and pushes through the assessment to get her medication as fast as possible, in the detriment to the nurses morale.

During the evening, she would take her last medication of the day before preparing dinner for herself and her husband. This helps her sleep at night as she finds it hard to sleep when in constant pain without her medication. She would take her nights easy by lying in bed and reading a book or watching TV programs with her husband until they both go to bed.

M. Scenario - Alan McKay

Alan wakes up early on the weekends as he needs to be at work by 9am. He works a 9-5 job in the hospital with a chance to do overtime if there is a problem in the hospital as well as being on call in case the ICT system goes down at any time.

His typical morning includes getting a shower, coffee and a light breakfast before tending to the child and making sure that the nanny is around to take care of the child while Alan and his fiancé is at work. If his fiancé is working at the same time as him, they would go to work together by car.

Once he gets to work, he would review the backlog to which he organises his day around. He would gather his team and brief them of the day's agenda and assign them the work. Due to him being in a managerial position within the hospital, he would be tasked with planning for the future of the hospital and implementing new technology to help with the overall function of the hospital such as installation of new routers, computers etc. and updates of new software, as well as programming new features on the IT system.

Before lunchtime, he would be in a meeting with other managerial staff in the hospital for their weekly updates on the needs of the hospital. During the meeting, Alan would have the chance to propose a new idea/technology that the members would discuss and review. Once the group votes on the proposal, if the proposal gets pushed through, he would be in charge of planning to implement the idea to the hospital coupled with a budget and deadline. During the meeting, there's the possibility that other members would bring up issues relating to IT in the workplace that Alan would be tasked with adding to the backlog and fixing.

The biggest feedback he gets from staff, especially the head of nurses is a way of helping reduce their workload using technology. The issue is that there isn't enough staff to cover the needs of the hospital, with the most urgent medical cases receiving the highest priority resulting in overworked staff, so they ask Alan what he can do to help with the lack of numbers and find a way for stress relief.

After the meeting, Alan would go back to working on what the agenda for the day would be until the end of the day, where he would go home with his fiancé depending if she is not needed in the hospital. If she is not available, he would either put in overtime or go home without her and tend to their household needs until she comes home.

At home, Alan is the one responsible for cooking and taking care of their child, as his fiancé usually arrives home tired and stressed. This has put a massive strain on their relationship ever since the birth of their child four years ago. He would think of ways to help reduce the workload for the staff in the hospital, which in turn would help with creating a better environment for his fiancé which would help ease their lives at home.

During his free time at home and on the weekends, he would either play video games with his friends online or go to the park with his family. From time to time, Alan and his fiancé would continue planning their wedding.

N. Scenario - Stacy Porter

On a normal day, Stacy would wake up early to make herself a coffee and a balanced breakfast. She likes to keep a healthy diet and be active as she can to ensure that she has maximum performance at work. She travels to work by walking and usually gets there before her shift starts to prep and sterilize herself.

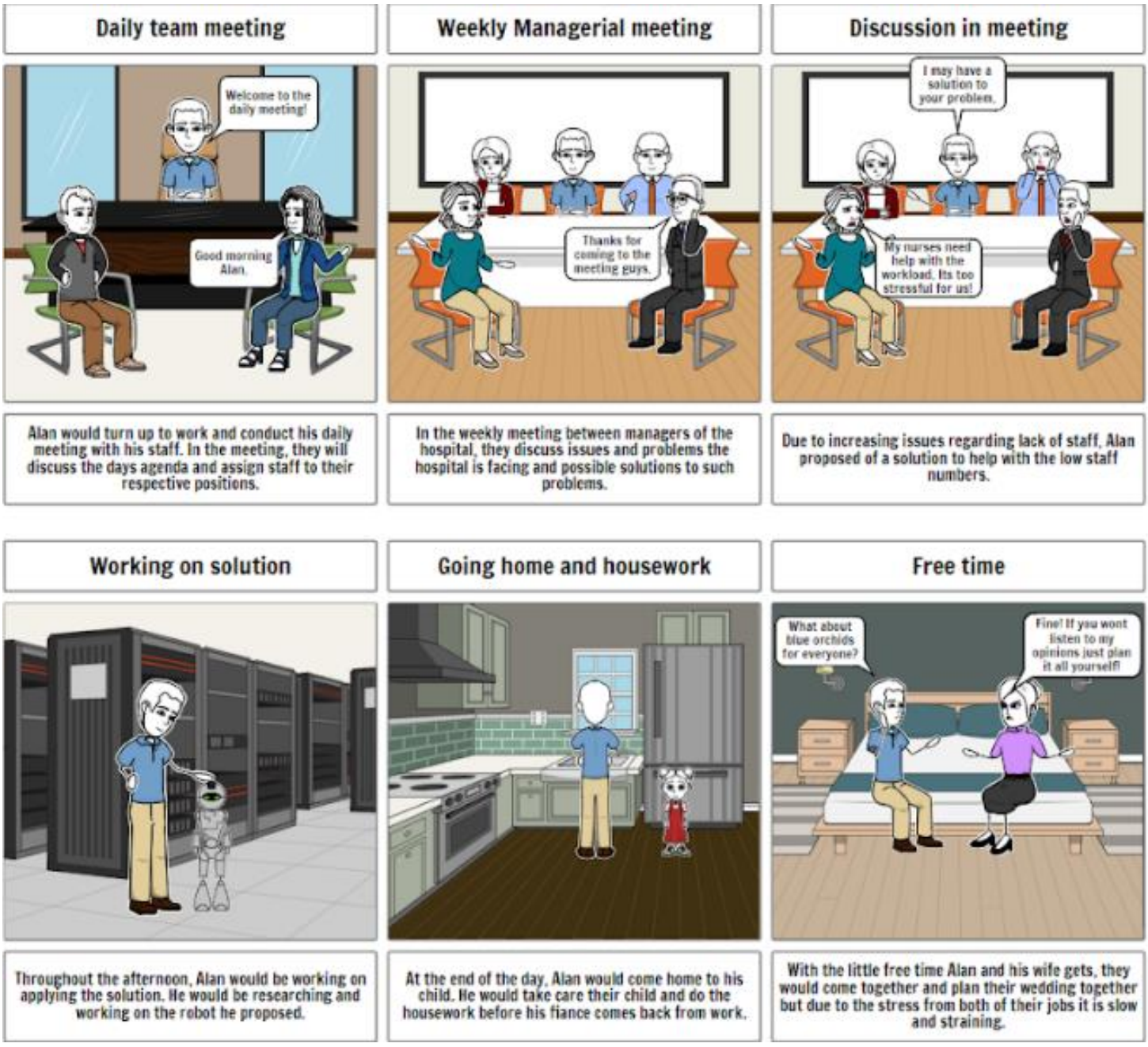
At work, she is usually busy with little time to rest apart from lunch breaks. Since she is a junior nurse, she is stationed in the A&E minor injuries assessment rooms. Her main job is to assess any injuries patients may have and to give them treatment or medication if needed. However, due to the lack of staff, Stacy and other nurses in the assessment rooms are occasionally called up to more urgent tasks such as being a theatre room nurse. In this situation, the assessment rooms are left with a skeleton crew that would increase the waiting time for the patients due to lack of staff. This happens often which tires the staff, to which they would rotate with staff in the assessment rooms for brief respite. Although this takes Stacy away from a more stressful environment, it lowers her ability to function at 100% when assessing patients in the waiting rooms.

Because of such a stressful and high-risk job, this creates friction between staff as they need their co-workers to work at maximum efficiency knowing that a mistake would be costly in their line of job. This causes a low morale within the ranks of nurses in Stacy's workplace which is carried back to life after work. After work, she would be tired and wouldn't be herself which results in a poor relationship with her friends and family. Back home after work, she would usually end up back in the evening with only time to cook dinner and a quick shower. She doesn't get much free time which alienates her from her friends, especially on the weekends where her friends would be free from where whereas she would be occasionally called in for work where 9 – 8 hours would not be uncommon.

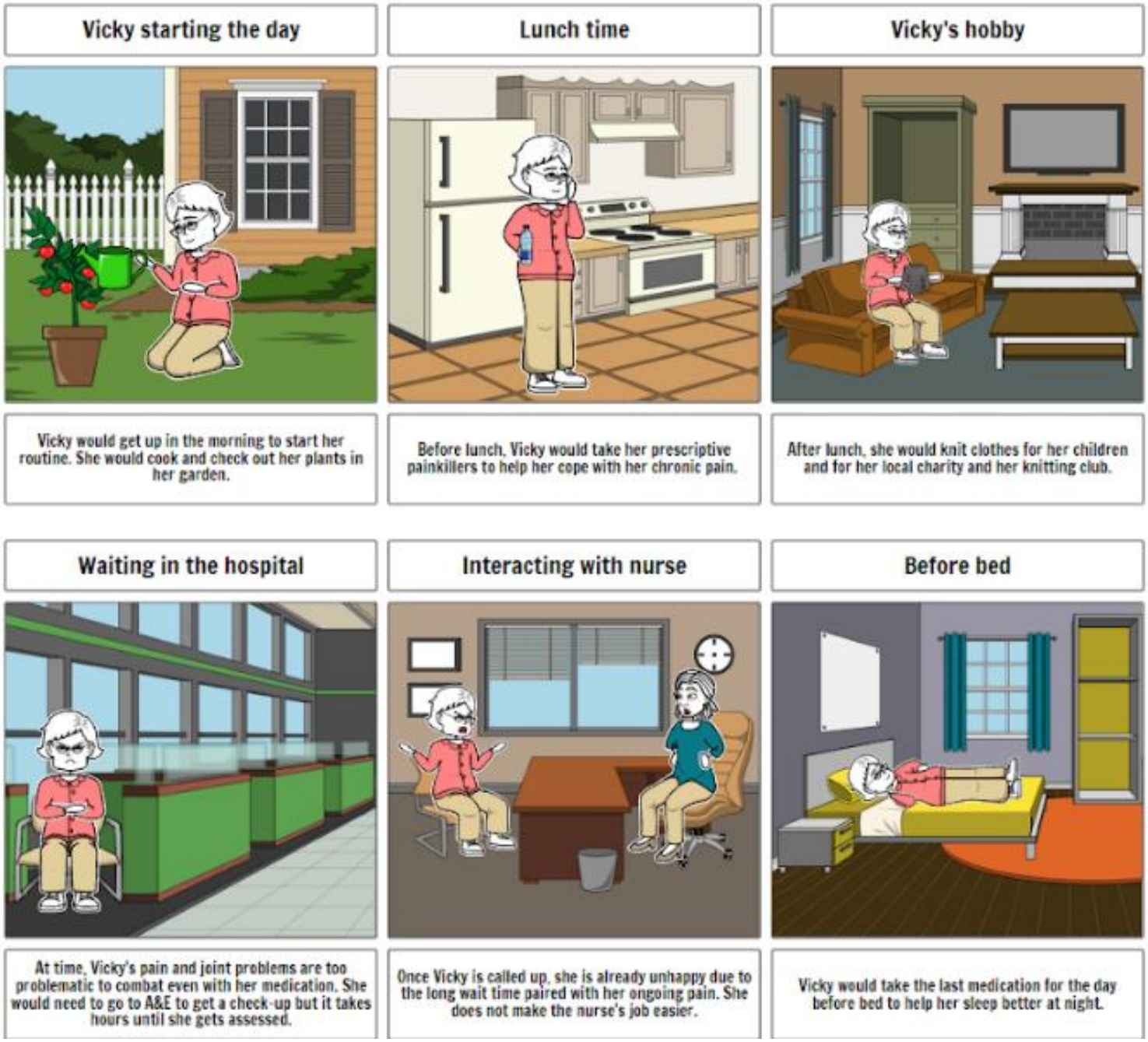
In other situations, her work hours would rotate to night shifts, where she would need to start work at 8pm and finish at 8 in the morning. This shift is more relaxed compared to the normal shifts she is used to as there are less patients but there are cases where she would need to deal with worse patients than what Stacy would normally get during the daytime. This also affects her social life as her friends would be conducting parties and nights out without her due to being at work.

Storyboards

O. Storyboard - Alan McKay



P. Storyboard - Vicky Bollard



Q. Storyboard - Stacy Porter