UNIVERSITY OF PLYMOUTH

COMP3000 – Project Portfolio

Computing Project

2022/2023

Audio Authentication via Voice Recognition on QTrobot (AAVR)

Gregory Kua Wee Leng

10687413

BSc (Hons) Computer Science (Cyber Security)

Acknowledgement

I would like to thank my supervisor Dr Hai-Van Dang for all the support and help that has been given to me over the course of this project. The time that she has spent with me to help me improve upon the work I have done is invaluable.

I would also like to thank my friends and family for being supportive of me during this time. They have been really helpful during my lows and I appreciate them.

1 Abstract

AAVR is an application that allows the user to authenticate themselves with a robot. The robot in question is a QTrobot which is a social robot designed by LuxAI. The authentication process is done with the use of 2FA (two factor authentication). The program is able to send a 6 digit code to the user's email address when stated and is able to a communicate with the user naturally. The program can allow the robot to store user's name so that it increases the human touch while talking with a user. This program aims to reduce the amount of interaction that is needed within a work place environment such as a place with a receptionist. In this case, the environment that was targeted was in a dentistry environment. This is to reduce the workload that is needed for tasks that can be done by the user themselves such as retrieving their medical record.

Word Count: 10522

Repository: https://github.com/greatyarn/COMP3000_Computing_Project

Contents

List of Tables

List of Figures

2 Links

Document Reference	Online Links
Gantt Chart	High Res Image (Imgur), (Mirror)
Trello Board	Trello
Version Control	GitHub
Meeting Minutes	OneDrive
Linux GUI Issue	High Res Image (Imgur) (Mirror)
Flowchart Maker and Diagram Software	Draw.io
Gantt Chart Software	GanttProject
Operating System for QTrobot	Ubuntu
Poster & Thumbnail	Poster (Imgur) (Mirror), Thumbnail (Mirror)
UML Diagrams: First Iteration	High Res Image (Imgur) (Mirror)
UML Diagrams: Second Iteration	High Res Image (Imgur) (Mirror)
Project Initiation Document	OneDrive
Video Demonstration	Youtube

Table 1 Links

3 Introduction

Speech recognition has come a long way since the invention of the telephone by Alexander Graham Bell (Juang & Rabiner, 2006). The first major breakthrough was not actually a speech recognition system rather a speech synthesizer called the voder (Voice Operating Demonstrator). Speech recognition was later stemmed out from this. It is stated by Juang & Rabiner (2006) that speech recognition was not the first thing that humans were interested in rather the ability to be able to mimic a human's speech. In regard to speech recognition specifically, Cole (2021) states that the Audrey system were the first listening computers. These computers are only able recognise numbers in specific environments. According to Shaughnessy (2004), this is partially due to the cepstrum being sensitive to changes within the environment. Early modals of how speech recognition works is by comparing them with known images of how a word would sound.

One of the main problems in this day and age is not being able to be authenticated with a device without needing any human interaction. Braz & Robert (2006) shows that most of the current method of authentication includes having people to interact with the system at hand such as Passwords, Pins, Cards, challenge response and much more. This report will give a solution to the problem at hand with the help of a social robot called a QTrobot.

3.1 Project Vision

The QTrobot is a toddler-like humanoid robot built by LuxAI. This project, Audio Authentication via Voice Recognition on QTrobot, or AAVR for short, aims at providing two-factor authentication for the QTrobot which is a social robot that is originally used as an autism robot tutor for improving a child's learning outcome at home (Qtrobot, n.d.) This authentication process with the use of a human voice and SMS allows for patients to retrieve their medical records, which might contain their age, height, and weight. This method of authenticating by voice is more convenient as it reduces the amount of human interaction that is needed.

3.2 Background / Scene Setting

This project is done in conjunction with PRIDE (Privacy-preserving Robotics in DEntistry). It is an initiative between dentistry, robotics and cyber security (*Privacy-preserving Robotics in DEntistry (PRiDE), n.d.*). This project is part of a wider program that stems and involves with students and lecturers across the world. It is meant to be used in the world of dentistry. Patients are allowed to communicate with the Qtrobot directly instead of going to a counter. This allows easier access for people who have problem with socialising with others but who still needs to get healthcare access.

Henschel et al., (2021) states that there is no one definition of a social robot. This makes it harder to define if a robot is social or not. Within the field of Human-robot interaction, as long as the robots interact with humans as peers or companions, they can be considered as a social robot. These robots will have the expectation on trying to provide solutions to people's social need. However, there needs to be thoughts on how the security of the social robots are (Miller et al., 2018). This is because the robot can be compromised in multitude of ways if the attacker has some sort of access to the robot. This can consist of exploitable ports within the robot from the outside and internal attacks from the local area network. Yaacoub et al., (2021) has a figure that shows a wide arrange of causes and consequences for a security robot viewpoint.

In this case, the QTrobot can be considered a social robot in this sense as it is able to provide some social help by being able to authenticate a user without the need of another human interaction. This project allows a user or a patient to identify to the Qtrobot to allow to authenticate themselves. It uses an OTP (One time password) authentication method with an email address to allow for the user to proof that they are in the system. The OTP will be sent to the patient's email address, and they will be able to speak the code out. The code will then be checked against the code that was sent out. If the code is correct, the user will be able to hear the next part of what they can do within the health centre.

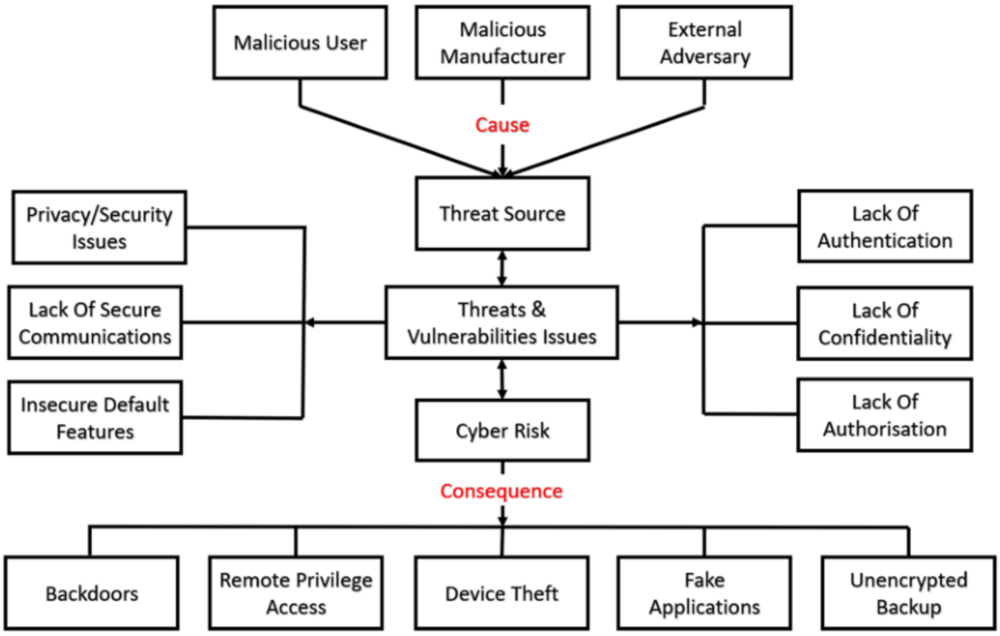


Figure 1 Security of robot (Yaacoub et al., 2021)

3.2.1 Regarding project changes

This section is just to explain that within the course of this project's timeline, there have been changes to the direction of this project. Mainly, when this project first started out, it was meant to have voice recognition in built to recognise different profiles when a user speaks to it but due to some circumstances that will be explained in the Project Methodology and Implementation section this project steered towards speech recognition instead while making use of user's personal detail to authenticate themselves other means.

3.2.2 Infrastructure of the QTrobot

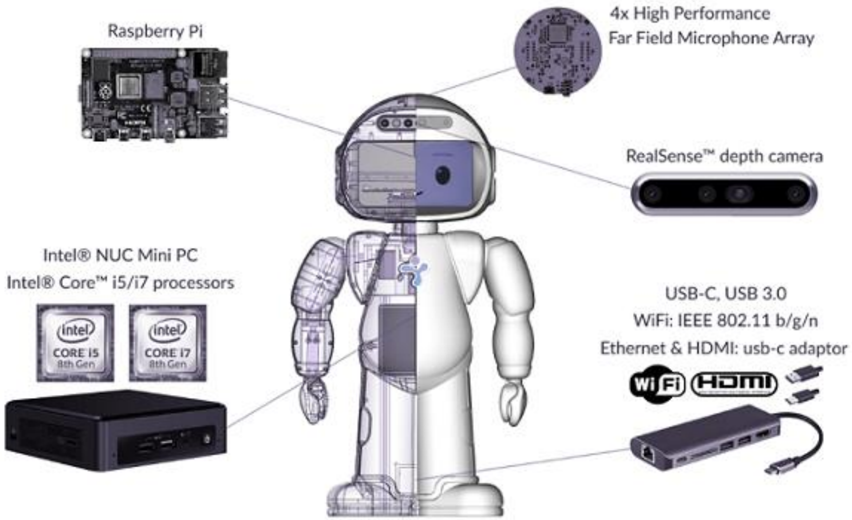


Figure 2 Graphical Infrastructure of the QTrobot

(QTrobot documentation, n.d.)

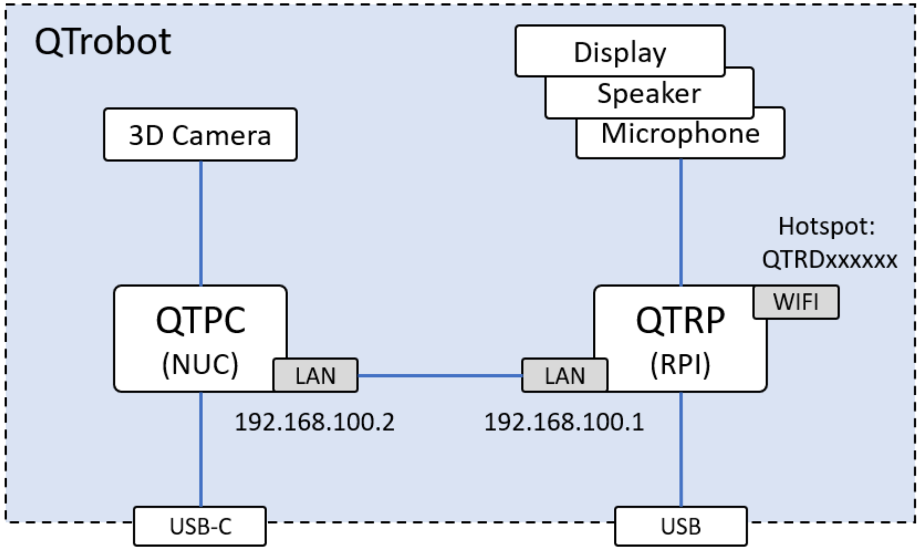


Figure 4 QTrobot Networking (QTrobot documentation, n.d.)

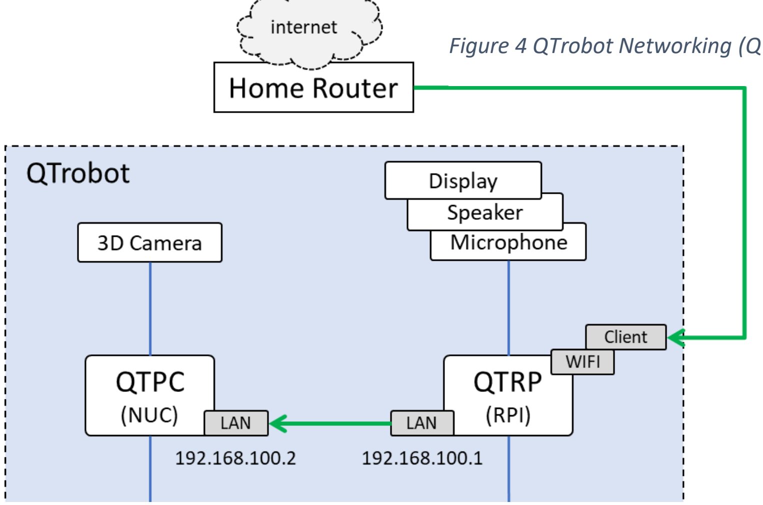


Figure 3 QTrobot Networking #2 (QTrobot documentation, n.d.)

The QTrobot is an autism robot tutor for improving a child's learning outcome at home (QTrobot, 2022). Figure 1 shows that the robot is made of a raspberry pi, 4x microphones, a depth camera powered by Intel's RealSense and a Nuc (*QTrobot documentation, n.d.*). The camera that is in the robot is connected to the Intel Nuc. The USB C adapter that is provided is also connected to the Nuc. The raspberry pi connects the rest of the components together. The Raspberry Pi links to the Nuc via an internal LAN (Local Area Network) cable. Each device, the nuc and the raspberry pi has been setup with different IP addresses. This component consists of the display that is on the face opf the robot, the speakers and microphone. There is also another usb at the back of the robot to plug in to the Raspberry Pi directly. This can be seen in Figure 2 and Figure 3 above.

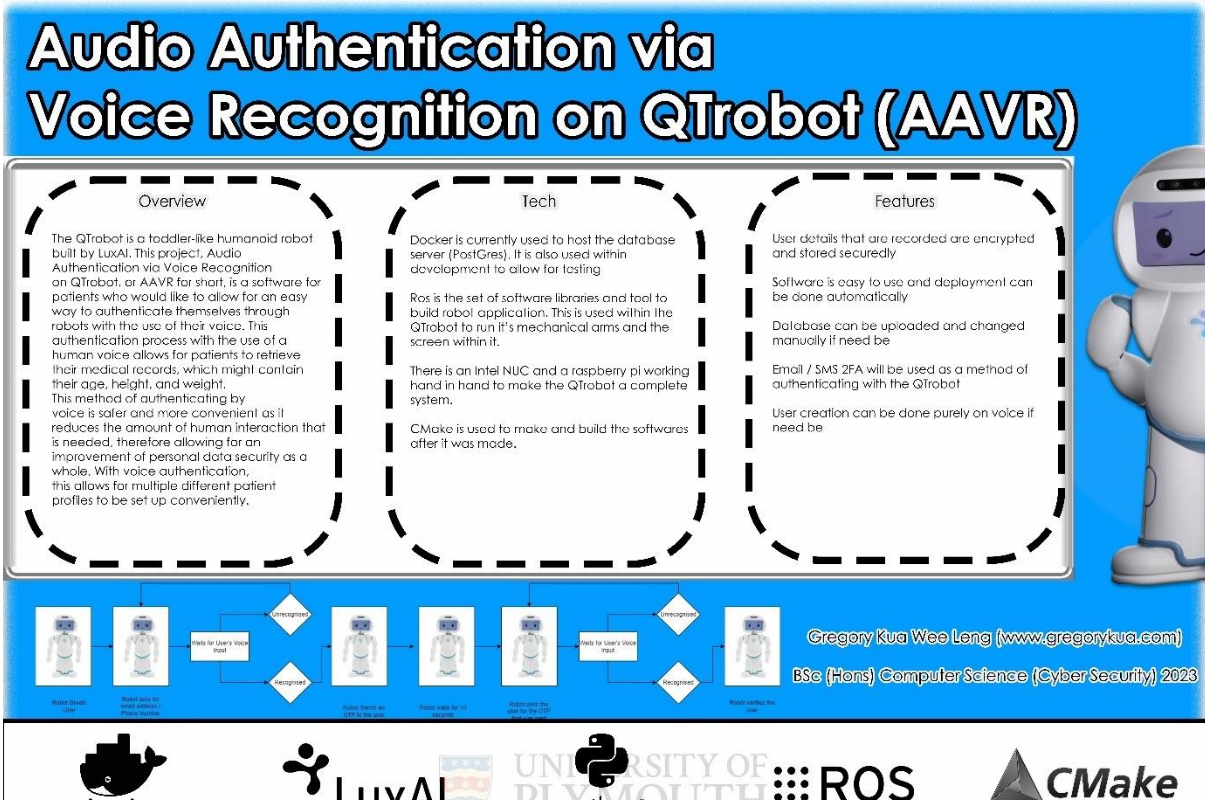
The QTRobot in the current configuration for this report is done via wifi through the Raspberry Pi. The Raspberry Pi passes the internet connection that is received through the LAN cable which is in the robot internally. This is how both the Raspberry Pi and the NUC can get an internet connection through one network. Currently, to work on the robot itself it is necessary for the developer to be within the robot's vicinity.

3.2.3 Poster

There are two images that are shown below. The two images are the thumbnail and the poster itself. Both images were made within Photoshop. Assets that were used are publicly available on their respective websites. This poster was made for the showcase that will happen after the submission of this report. The thumbnail and poster will also be uploaded on the university's website if need be. Both of the high-resolution images can be found here.

Audio Authentication via Voice Recognition on QTrobot (AAVR)
LuxAI
ROS
UNIVERSITY OF PLYMOUTH

Figure 6 Thumbnail


Audio Authentication via Voice Recognition on QTrobot (AAVR)

Overview

The QTrobot is a toddler-like humanoid robot built by LuxAI. This project, Audio Authentication via Voice Recognition on QTrobot, or AAVR for short, is a software for patients who would like to allow for an easy way to authenticate themselves through robots with the use of their voice. This authentication process with the use of a human voice allows for patients to retrieve their medical records, which might contain their age, height, and weight.
This method of authenticating by voice is safer and more convenient as it reduces the amount of human interaction that is needed, therefore allowing for an improvement of personal data security as a whole. With voice authentication, this allows for multiple different patient profiles to be set up conveniently.

Tech

Docker is currently used to host the database server (PostGres). It is also used within development to allow for testing

Ros is the set of software libraries and tool to build robot application. This is used within the QTrobot to run it's mechanical arms and the screen within it.

There is an Intel NUC and a raspberry pi working hand in hand to make the QTrobot a complete system.

CMake is used to make and build the softwares after it was made.

Features

User details that are recorded are encrypted and stored securedly

Software is easy to use and deployment can be done automatically

Database can be uploaded and changed manually if need be

Email / SMS 2FA will be used as a method of authenticating with the QTrobot

User creation can be done purely on voice if need be

Robot Greets User
Robot asks for email address / Phone Number
Waits for User's Voice Input
Unrecognised
Recognised
Robot Sends an OTP to the user
Robot waits for 10 seconds
Robot asks the user for the OTP that was sent
Waits for User's Voice Input
Unrecognised
Recognised
Robot verifies the user
LuxAI
UNIVERSITY OF PLYMOUTH
ROS
CMake

Gregory Kua Wee Leng (www.gregorykua.com)

BSc (Hons) Computer Science (Cyber Security) 2023

Figure 5 Poster

3.3 Market Research

Market research is done to see if there is any relevant services or providers that have done similar projects to what this project is doing. The following below are the market research that have been done early in the project's planning phase.

3.3.1 HSBC Banking

HSBC UK developed a new voice-driven technology back in 2020. When one calls the support help line on their mobile phone. The user would be instructed to repeat a sentence which is "My Voice is my password". This allows HSBC to do a check whether if the user calling is the person that owns the bank account (HSBC, n.d.).

3.3.2 Google Assistant

Google Assistant has a feature called Voice Match. Voice Match allows users of Google Devices to let their device knows that they are the owner of a Google Device. This is used on one of their Nest product line-up called the Google Nest Hub, where it is able to show personal results using your personal search history and mail history if it is able to authenticate you when you say the key words, "Hey Google" or "Okay Google". This also works on other Nest Enabled Devices and the Google Assistant Software on Android Phones. iOS and iPadOS devices can connect their Google Devices such as their Nest Mini or Nest Hub via their device but are not able to use Voice Match directly on their Apple Device (Google, n.d.).

3.3.3 Windows 10 / 11

Windows is an operating system (OS) created by Microsoft. In version 10 and 11 of Windows, Users are able to enable Speech Recognition on their windows compatible devices if it meets the standard requirements. It is able to learn your voice and users are able to train it so that the device understands the user better. It is used for simple and complex tasks to allow the user to use their devices completely hands free if they want to (Microsoft, n.d.)

3.4 Requirements

The requirements section explains all the requirements that is needed for the project to be a success. This section is broken down into multiple sections; namely, Objectives, User Stories, Functional and non-functional requirements and Use Cases. Each of the sections will explain what the section entails for within the project.

3.4.1 Objectives

Objectives are set to allow for goals to be achieved within this project. The evaluation section here will explain if the objectives that have been set are achieved.

The objectives that are set are as follows:

- AAVR is able to authenticate users when requested.
- AAVR can recognise the user's voice and authenticate it based off that.

3.4.2 User Stories

User stories is a tool in Agile development where it is meant to aid in the development of the project (What are user stories?, n.d.). They are from the point of view of what a user can be expected to do with the program.

The user stories of this project consist of:

- AAVR should allow the user to quickly log in to the robot so that they are able to retrieve their personalised data when I ask questions.
- AAVR should allow the user to talk to the robot naturally so that they are able to use it comfortably
- AAVR should allow the user to log in to their personalised data without hassle so that they do not feel that it is an annoyance.

3.4.3 Functional Requirements

According to Puzhevich (2021), functional requirements are requirements that shows how the system is meant to work. It includes functions of an application that was created by a software developer. It also defines the scope of the system as a whole (Whitney, 2021).

The functional requirements for this project are as follows:

- The program must be able to understand and respond to the user correctly.
- The program must be able to send out an OTP to the user's email address when requested.
- The program must be able to confirm if the OTP stated is correct or not.
- The program must be able to check if the user has an account within the database to avoid duplicates.

3.4.4 Non-functional Requirements

Non-functional requirements are requirements that includes features that are not necessary for the product to function. It includes features such as system usability, security and maintainability (Puzhevich, 2021).

The non-functional requirements for this project are as follows:

- The program must be easy to use.
- The program must be simple to set up.
- The program needs to be scalable.

3.4.5 Use Cases

Use cases are basically defined as actions (Rolland & Achour, 1998). It can be defined as a singular action or a flow of action. A singular action is defined when a single task is being done at one point in time while a flow of action starts from a single task and then evolves from that. The use cases in this project stems from a combination of both

The use cases requirements for this project are as follows:

- The user is able to speak their name to the robot.
- The user is able to speak their email to the robot.
- The user is able to tell the robot to append any email provider.
- The user is able to authenticate themselves with the robot.

3.5 Legal, Social, Ethical & Professional Issues

This section will go through the considerations of the LSEP issues that might happen during and after the project's development. The scenarios that will be listed below are possibilities that could happen and is not a direct result that it will happen.

3.5.1 Legal Issue

4.5.1.1 Creative Commons Licensing

The creative commons license that has been used is a CC 4.0 license. This license allows for any individual to share and adapt the project at will as the developer is not allowed to revoke the license so long as the license's terms have not been broken. The primary reason for the selection of this license is to allow for any individual who wishes to work on the project in the future to do so in case the developer has decided to no longer continue development of the program. Under this license, any new developers will have been granted the opportunity to freely fork the project under its licensing terms and continue to develop it in their own time with minimal regards to any potential licensing issues that may or may not arise in the future.

4.5.1.2 Protection of Personal Voice Data

At this point of time, voice recordings are stored locally on the device itself and are not uploaded anywhere on the internet. The voice data are processed locally and the transcript from the voice is then only uploaded to the database of choice which in this case is PostgreSQL.

Data deletion from the database is not currently directly available from talking with the QTrobot and needs to be done externally. Since this is not part of the main project's focus, this was not implemented.

4.5.1.3 UK General Data Protection Regulation (UK GDPR)

The UK General Data Protection Regulation or commonly known as the UK GDPR is the UK's implementation of the General Data Protection Regulation (GDPR) which is an EU law (*Guide to the UK General Data Protection Regulation (UK GDPR), n.d.*).

The principles (n.d.) stated by the ICO states that there are seven key principles to the UK GDPR which includes.

- **Lawfulness, fairness and transparency**: Data that is used must be processed lawfully. Data that is used needs to be processed fairly and organisations needs to be transparent on how the data is being used.
- **Purpose limitation:** Data that is collected needs to be for specific reasons and for a legitimate purpose. All the data on how it is being used needs to be written out within a document.
- **Data minimisation:** Data should be limited to only what is necessary for a service to be useful to its purpose.
- **Accuracy:** Data that is processed needs to be accurate and be kept up to date if necessary. Organisations needs to ensure that the data that is being collected for legitimate uses is accurate.
- **Storage limitation:** Data should not be kept for any longer that is necessary for the service e or product to function.
- **Integrity and confidentiality (security):** Security of the personal data should be top priority and it needs to be processed in that way. Examples would be protection against unauthorized processing of data and accidental data deletion.
- **Accountability:** UK GDPR states that organisations should take appropriate measures to ensure compliance. They must be able to demonstrate that measures have been taken place to ensure that they are able to be held accountable if something happens to the data that they were processing.

With this in mind, the application has been taken into consideration of the UK GDPR act where only necessary data that is recorded to ensure nothing else is being recorded at any point of time. The user who is using the QTrobot will at all times know when they are being recorded and all the information used can be retrieved if need be.

4.5.1.4 Licensing for external libraries

A number of external libraries have been used to aid the project's progress. When developing the application, no proprietary code was used in this case and the libraries that were used have licenses that allow for reuse. The full list of external libraries that was used is listed in the Appendix here.

3.5.2 Social Issue

One of the main social issues that can arise from the project is the fact that the QTrobot could be taking jobs away from receptionists within the dental industry. If the project expanded further and not only within the authentication phase that this project relies on. A lot of what a single receptionist can do, can be replicated with a robot.

Lewicki et al. (2019) states that the current generations of robots are unintelligent employees. They are following a set of rules. However, a machine learning modal of some sort could have the potential of eliminating the need of human work if the workload can be passed off. This in turn allows for organisations and services to have more profit overall.

Another issue that can arise is the transcription that is used. Specifically, complicated names and foreign English names. This is because at the end of the day there is a microphone that is within the QTrobot that is being used, if the microphone is not able to pick up the name or email address correctly, the program is not able to be used.

3.5.3 Ethical Issue

Survey data that is collected are used in an ethically matter. Data that is collected from the public are told specifically what is being used for. The data that is being collected are all related to towards the robot's response and how would they use it. These data is collected to improve further on the project and to understand what is the current people's perspective on authentication with their voice using the robot.

3.5.4 Professional Issue

ACM Code of Ethics and Professional Conduct (n.d.) and BCS Code of conduct (n.d.) have ethics and conducts that have been abided by during the development of this project. This project at this point of time is hosted on GitHub so if anyone would like to fork the project in the future is able to do so to improve on the project. Developers of the project have taken into account of the privacy and health of others. No grounds of discriminations have been made and within the report things that have went wrong will be stated.

3.6 Meeting Minutes

There are two types of meetings that have happened during the duration of this project from the start. These two are the biweekly meetings with the supervisor and the biweekly meetings with the group that was allocated to the students.

These meetings exist to allow to update people to know what has happened during the sprints. It is also used as a peer meeting session to elaborate on what has done. The meeting minutes have been recorded and can be seen in the appendix section here.

Other meetings that were not recorded but had happened during the development of this project includes meetings with one of the developers in LuxAI directly to help with some of the issues that has happened during the course of this project.

4 Project Management

Every project will need to have some planning done in preparation before the implementation phase starts. This allows for the project to have a direction on where the end goal needs to be. It also allows for the developer to manage their time properly. This section includes the method of approach, gantt charts, kanban board, version control, risk assessment and sprint reviews. Each of the section will be explained what its for.

4.1 Method of Approach

This section is split into two parts. Although they work hand in hand. The reason why it was split is to accommodate and explain its history on how it works. This will result in a better understanding on how it integrates with this project.

4.1.1 Agile

Dyba et al. (2014) states that for managing software projects, Agile shows a new approach of doing it. It allows for less strict control and more informal collaboration and planning. Compared to the traditional approach for product management, Agile allows for management of complex projects. This includes understanding that between planning and execution there should be a shorter time frame between those two. Other than that, Agile explains that planning for an action does not provide all the details of its implementation. Finally, Agile explains that to make sense of an environment during software development learning and creativity are needed. To summarise, Agile project management consists of short cycles of incremental delivery of project features. This can be seen by the chart below by Kintone (2019)

Within this project, Agile development can be seen in the form of user stories, scrum, and Kanban Boards.

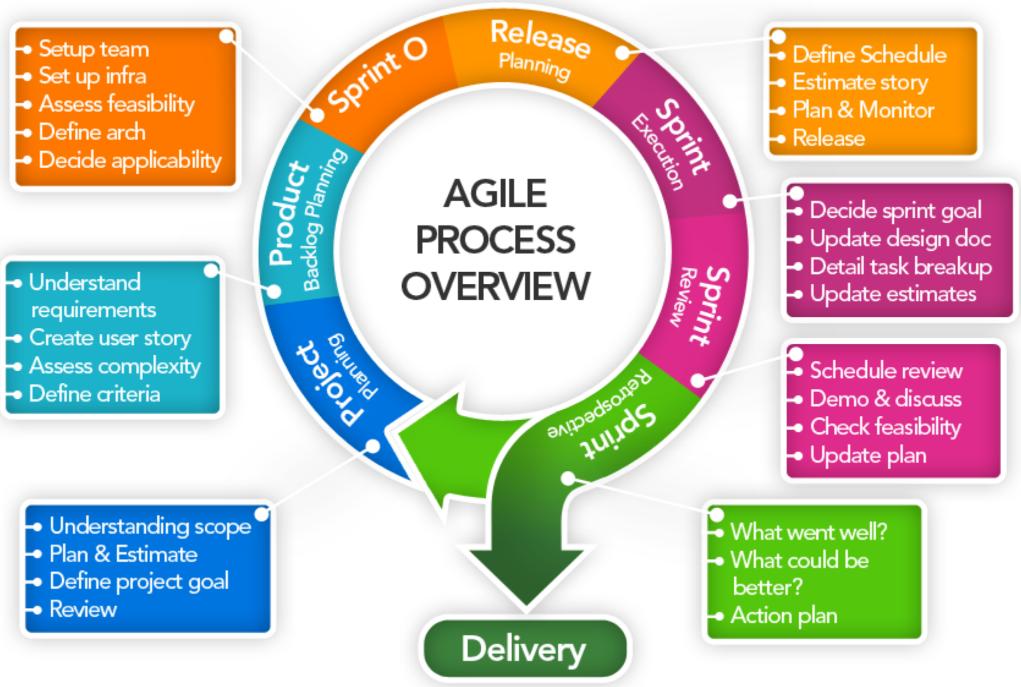


Figure 7 Agile Process Overview (Kintone, 2019)

4.1.2 Scrum

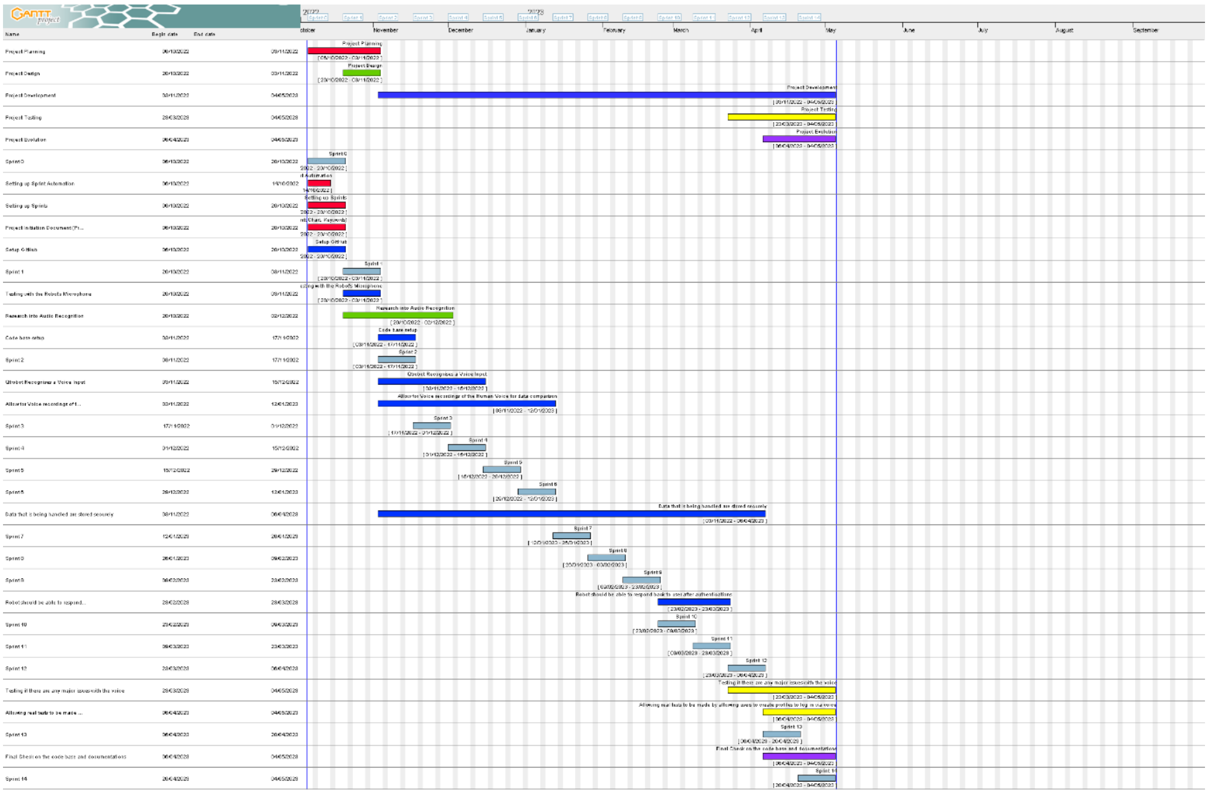
Sustaining complex product and maintaining a framework for developing is where scrum comes in (*What is Scrum?, n.d.*). Scrum was developed in the early 1990s to help people generate value for complex situations with adaptive solutions. Scrum is a part of the Agile methodology which can be seen within the kanban boards. Sprints are the main part of scrum. They are fix lengths of events and normally consists of a month or less. This is to ensure consistency between each sprints. Although it's not applicable within this project since it is a one developer team. A scrum team normally consists of at least, the product owner, scrum master and developers. The product owner is responsible for prioritising the product backlog that was made with the help of the developers. They will have to ensure that the most important features are being developed first before minor features. The scrum master ensures that their team is following the scrum protocol and by making sure that the product backlog is managed effectively. Finally, the developers have the role of making sure that each itemised item that is being added to the product backlog is something that is a useful implementation. This also includes making sure that iterative features within the product are being implemented properly during the sprints that was allocated to.

For this project, the sprints consist of two-week intervals starting from the 6th of October 2022. The sprints do not include vacation days or public holidays which will be stated within the sprint reviews.

4.2 Gantt Chart / Sprint Plannings

Gantt charts are used to show an estimated project flow from the start of the project's lifespan to the end. This flow will be split based on the sprints that have been set. The sprints will be set at every two weeks. The blue line at the start and at the end marks the start of the first sprint and the final day of the last sprint. The image attached below shows the first revision of the gantt chart that has been made. The gantt chart below is made using GanttProject which is an open source Gantt Chart Maker tool that have been found (GanttProject, n.d.).

The gantt chart exists to allow for the project to be split between sections. It also gives a top down overview of an estimation on how the project will be run.



Name	Begin date	End date
Project Planning	06/10/2022	03/11/2022
Project Design	20/10/2022	03/11/2022
Project Development	03/11/2022	04/05/2023
Project Testing	23/03/2023	04/05/2023
Project Evolution	06/04/2023	04/05/2023
Sprint 0	06/10/2022	20/10/2022
Setting up Sprint Automation	06/10/2022	14/10/2022
Setting up Sprints	06/10/2022	20/10/2022
Project Initiation Document (Pr...	06/10/2022	20/10/2022
Setup GitHub	06/10/2022	20/10/2022
Sprint 1	20/10/2022	03/11/2022
Testing with the Robots Microphone	20/10/2022	03/11/2022
Research into Audio Recognition	20/10/2022	02/12/2022
Code base setup	03/11/2022	17/11/2022
Sprint 2	03/11/2022	17/11/2022
Qtrobot Recognises a Voice Input	03/11/2022	15/12/2022
Allocate Voice recordings of t...	03/11/2022	12/01/2023
Sprint 3	17/11/2022	01/12/2022
Sprint 4	01/12/2022	15/12/2022
Sprint 5	15/12/2022	29/12/2022
Sprint 6	29/12/2022	12/01/2023
Data that is being handled are stored securely	03/11/2022	06/04/2023
Sprint 7	12/01/2023	26/01/2023
Sprint 8	26/01/2023	09/02/2023
Sprint 9	09/02/2023	23/02/2023
Robot should be able to respond...	23/02/2023	23/03/2023
Sprint 10	23/02/2023	09/03/2023
Sprint 11	09/03/2023	23/03/2023
Sprint 12	23/03/2023	06/04/2023
Testing if there are any major issues with the voice	23/03/2023	04/05/2023
Allowing real tests to be made ...	06/04/2023	04/05/2023
Sprint 13	06/04/2023	20/04/2023
Final Check on the code base and documentations	06/04/2023	04/05/2023
Sprint 14	20/04/2023	04/05/2023

Figure 8 Gantt Chart (High Res Here)

4.3 Kanban Board

The Kanban board that is being used is being made with the help of Trello. A kanban board is a project management tool that is used to visualise the work that is being and / or currently been done (Rehkopf, n.d.). It uses cards and columns to help the developer to manage their workflow and help them to be on the right track. Compared to other project management board on the market, Trello is one of the more popular ones that are used by big companies such as Visa and Zoom (*Trello for product management teams, n.d.*). Automations and tagging are also some of the features that have been used to keep this project in check.

The kanban board in this case has been set up with a Backlog, Sprint 0 – 11, Room for Improvement and a resources column. The backlog column contains backlog tasks that have not been set in a specific sprint that needs to be done. The Sprints column contains tasks that are in progress or have been completed during the sprint. The Room for improvement column includes cards that allows for improvement within the project, but it is not deemed necessary for the project at hand. Finally, the resources tabs contain links and more information about the project for quick assess to the other files. This board is always up to date with what has happened. Automations have been set up at the start of the project to easily identify which tasks is in progress, its sprint and which has been done. This includes automatic adding of tags based on its column, buttons for automatic completion and much more.

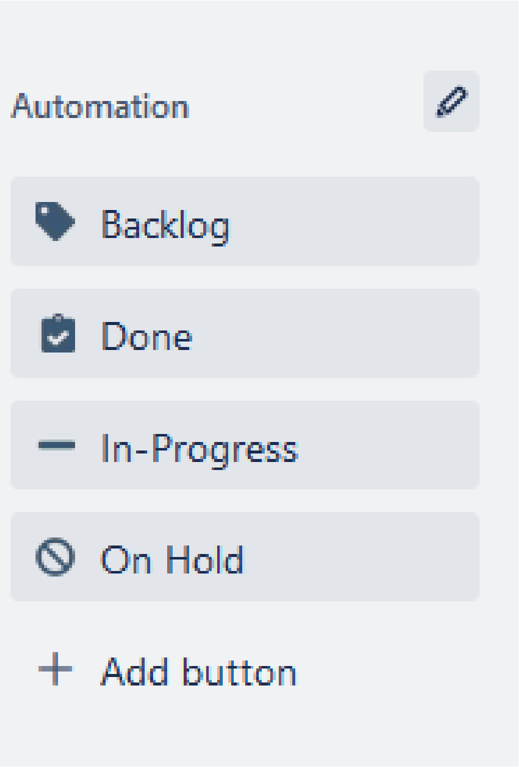
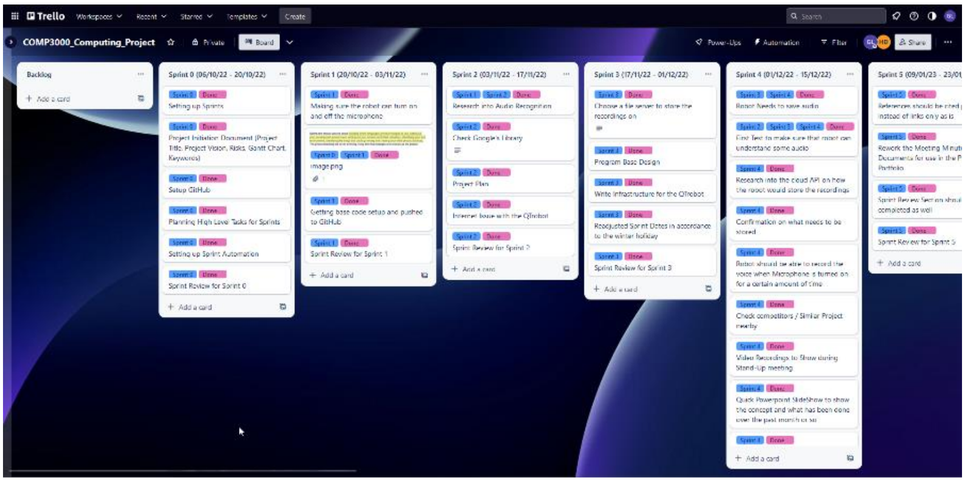


Figure 9 Trello Automations



COMP3000_Computing_Project

Backlog

Sprint 0 (06/10/22 - 20/10/22)

Setting up Sprints

Project Initiation Document (Project Title, Project Vision, Risks, Gantt Chart, Keywords)

Setup GitHub

Planning High Level Tasks for Sprints

Setting up Sprint Automation

Sprint Review for Sprint 0

Sprint 1 (20/10/22 - 03/11/22)

Making sure the robot can turn on and off the microphone

image.png

Getting base code setup and pushed to GitHub

Sprint Review for Sprint 1

Sprint 2 (03/11/22 - 17/11/22)

Research into Audio Recognition

Check Google's Library

Project Plan

Internet Issue with the QTrobot

Sprint Review for Sprint 2

Sprint 3 (17/11/22 - 01/12/22)

Choose a file server to store the recordings on

Program Base Design

Write Infrastructure for the QTrobot

Readjusted Sprint Dates in accordance to the winter holiday

Sprint Review for Sprint 3

Sprint 4 (01/12/22 - 15/12/22)

Robot Needs to save audio

First Test to make sure that robot can understand some audio

Research into the cloud API on how the robot would store the recordings

Confirmation on what needs to be stored

Robot should be able to record the voice when Microphone is turned on for a certain amount of time

Check competitors / Similar Project nearby

Video Recordings to Show during Stand-Up meeting

Quick Powerpoint SlideShow to show the concept and what has been done over the past month or so

Sprint 5 (09/01/23 - 23/01

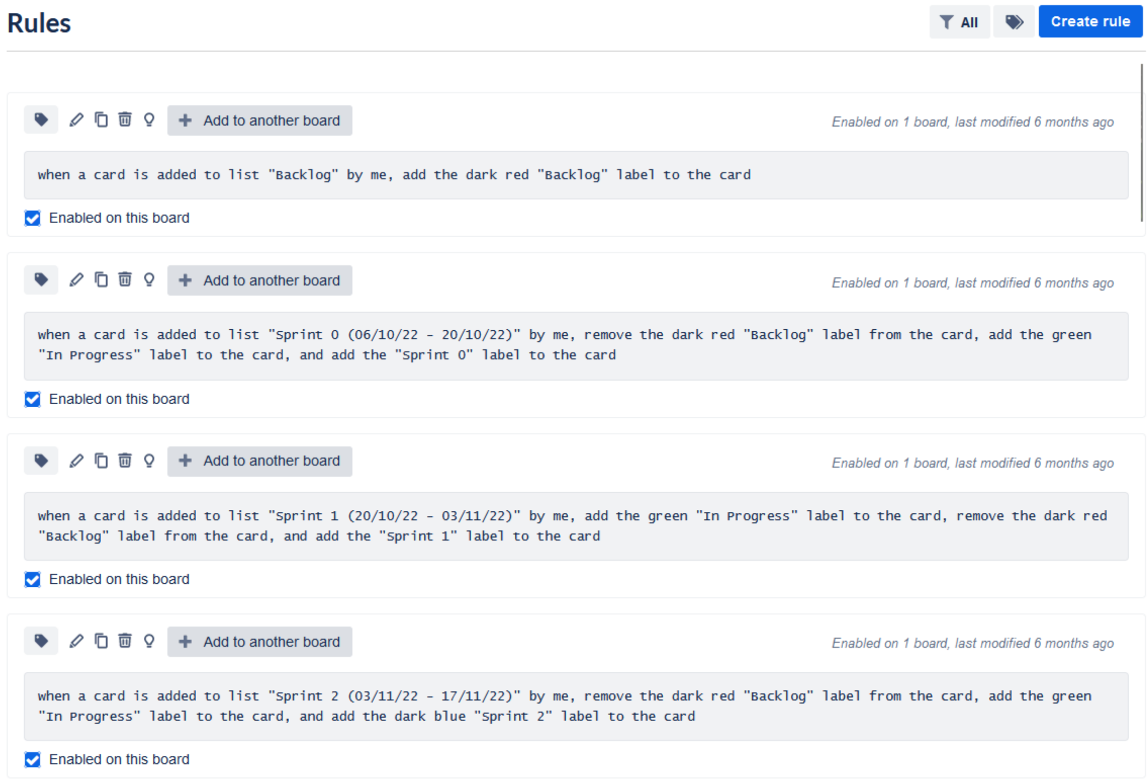
References should be cited instead of links only as is

Rework the Meeting Minute Documents for use in the P Portfolio

Sprint Review Section shoul completed as well

Sprint Review for Sprint 5

Figure 10 Trello Board (Kanban Board)



Rules

Enabled on 1 board, last modified 6 months ago

```
when a card is added to list "Backlog" by me, add the dark red "Backlog" label to the card
```

☑ Enabled on this board

Enabled on 1 board, last modified 6 months ago

```
when a card is added to list "Sprint 0 (06/10/22 - 20/10/22)" by me, remove the dark red "Backlog" label from the card, add the green
"In Progress" label to the card, and add the "Sprint 0" label to the card
```

☑ Enabled on this board

Enabled on 1 board, last modified 6 months ago

```
when a card is added to list "Sprint 1 (20/10/22 - 03/11/22)" by me, add the green "In Progress" label to the card, remove the dark red
"Backlog" label from the card, and add the "Sprint 1" label to the card
```

☑ Enabled on this board

Enabled on 1 board, last modified 6 months ago

```
when a card is added to list "Sprint 2 (03/11/22 - 17/11/22)" by me, remove the dark red "Backlog" label from the card, add the green
"In Progress" label to the card, and add the dark blue "Sprint 2" label to the card
```

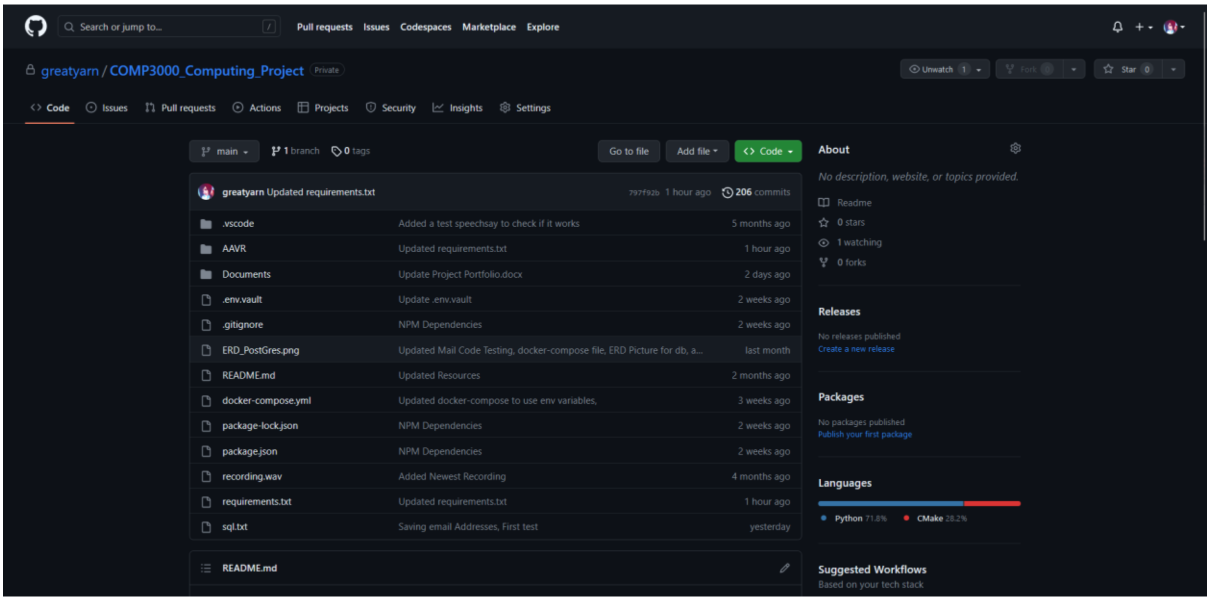
☑ Enabled on this board

Figure 11 Trello Automation Rules

4.4 Version Control

A version control is used within this project to keep track of changes within the code. This allows for any accidental deletion of codes if the previous versions of it has been committed and pushed. This also serves as a memory bank to look back at lines of codes or files that have been deleted or changed. With this, the developer is able to look back at which section of the file that could have broken the newer version of the program and revert back to it if needed.

The version control that is being used in this project is Git. GitHub is used as it is free to use and it allows for the use of git command line. It is also a big part of this project's testing phase which will be explained in more detail in the Project Testing section of this report.



greatyarn / COMP3000_Computing_Project Private

1 branch 0 tags

greatyarn Updated requirements.txt 797f92b 1 hour ago 206 commits

.vscode	Added a test speechsay to check if it works	5 months ago
AAVR	Updated requirements.txt	1 hour ago
Documents	Update Project Portfolio.docx	2 days ago
.env.vault	Update .env.vault	2 weeks ago
.gitignore	NPM Dependencies	2 weeks ago
ERD_PostGres.png	Updated Mail Code Testing, docker-compose file, ERD Picture for db, a...	last month
README.md	Updated Resources	2 months ago
docker-compose.yml	Updated docker-compose to use env variables,	3 weeks ago
package-lock.json	NPM Dependencies	2 weeks ago
package.json	NPM Dependencies	2 weeks ago
recording.wav	Added Newest Recording	4 months ago
requirements.txt	Updated requirements.txt	1 hour ago
sql.txt	Saving email Addresses, First test	yesterday

README.md

About

No description, website, or topics provided.

Readme
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Python 71.8% CMake 28.2%

Suggested Workflows
Based on your tech stack

Figure 13 GitHub Repository Website

Figure 14 Git (Git, n.d.)

Figure 12 GitHub (GitHub, n.d.)

4.5 Risk Assessment

Risk assessments are done at the start of the sprint. This is to ensure that any risks that is probable will be taken into account during the lifespan of the project. A table is used to list the risks out to ensure easy readability. There are three columns, which lists the risks, the probability of the risk happening and how would the developer address the issue if the issue were to happen.

Risks	**Probability (Low, Moderate, Significant, High)**	**Addressing the Issue**
Sprint does not get completed within the allocated time	Low	The sprint that was not completed on time will be moved to the next sprint and be updated on the Kanban Board accordingly
QTrobot does not work as expected due to unexpected issues	Moderate	Fixes and patches will be applied accordingly before attempting to move further on with the project
Computer that is used stops working due to human error or accidents	Low	Quickly find a replacement device to ensure sprints are done to minimise delays
Testing procedures might take longer than usual	High	Do the testing thoroughly
Computer might corrupt local disc which causes loss of data that impacts the project	Low	Having local backups and a git version control set in place
Getting burnt out on the project due to overwork and stress	Moderate	Having breaks in between and having a good work-study balance

Table 2 Risk Assessment

4.6 Sprint Reviews

This project uses the scrum methodology thus needing sprint reviews. Sprint reviews are used to include what has been completed during the set time, and what the backlog will look like within the next sprint (Wrike, n.d.). Sprints within this project is in two weeks intervals and does not include public holidays, winter vacation and spring vacation. Comparing the dates from the Gantt Chart and the Sprints. There have been some deviations due to some delays that have happened within the project. In total, there are 12 sprints which its dates have been stated below in the review itself. The screenshots of the kanban boards can be seen within the appendix of this report here.

5 Project Methodology and Implementation

This section explains the methodology and implementation of this project, it will include designs that were made originally, the development phase that has happened, deviations, testing solutions, alternative solutions and issues that has come up during the lifecycle of this project.

5.1 Project Design

The project design section includes multiple sections including diagrams, design principles, technologies and skills. This section is done before the implementation phase to allow for a better understanding on what will be done later. It also allows for an explanation on changes that had happen during the project's lifespan.

5.1.1 UML Diagrams

These Unified Modelling Language Diagrams or UML for short is used to design software is a visual language to show how a system would be defined (Koç et al., 2021). UML diagrams allows for a complex system within an environment to be condensed into a visual representation. This allows for a better understanding on how a system will work when trying to manage the workload during the implementation phase of a project.

5.1.1.1 First Iteration

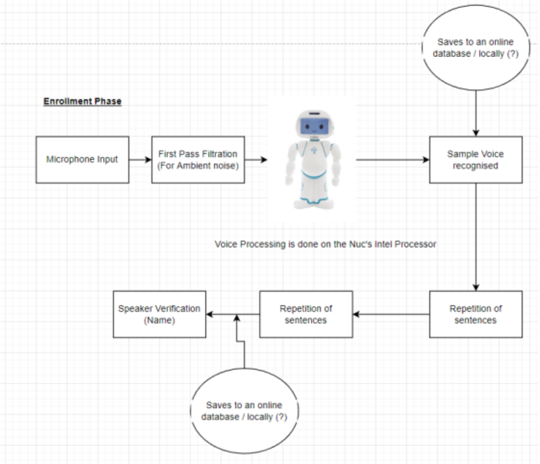

Saves to an online database / locally (?)
Enrollment Phase
Microphone Input
First Pass Filtration (For Ambient noise)
Sample Voice recognised
Voice Processing is done on the Nuc's Intel Processor
Speaker Verification (Name)
Repetition of sentences
Repetition of sentences
Saves to an online database / locally (?)

Figure 15 First iteration: Enrollment Phase

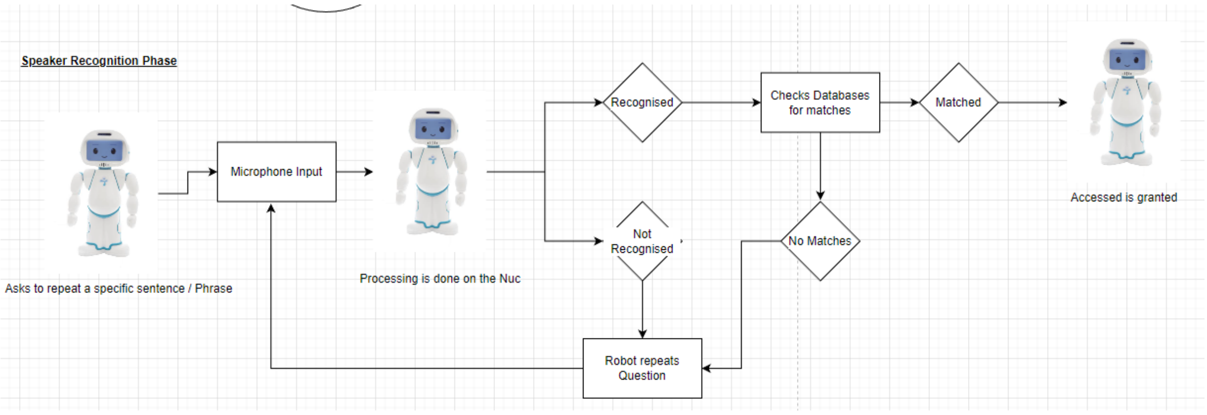

Speaker Recognition Phase
Microphone Input
Recognised
Checks Databases for matches
Matched
Accessed is granted
Not Recognised
No Matches
Asks to repeat a specific sentence / Phrase
Processing is done on the Nuc
Robot repeats Question

Figure 16 First Iteration: Speaker Recognition Phase

These diagrams were designed early on during the project's lifespan. It consists of two phases. The first phase is the Enrollment Phase which is the phase where a patient would like to enroll themselves to the clinic. This includes recording the patient's voice with multiple sentences to allow for the robot to remember the way that the patient speaks.

The second phase is the speaker recognition phase, this phase includes checking the database where all the voice recording samples are being saved. These samples are then matched to the current voice recording to check if that patient have already registered with them before. If they have been registered, access will be granted if they are not, the process cycle will repeat.

5.1.1.2 Second Iteration

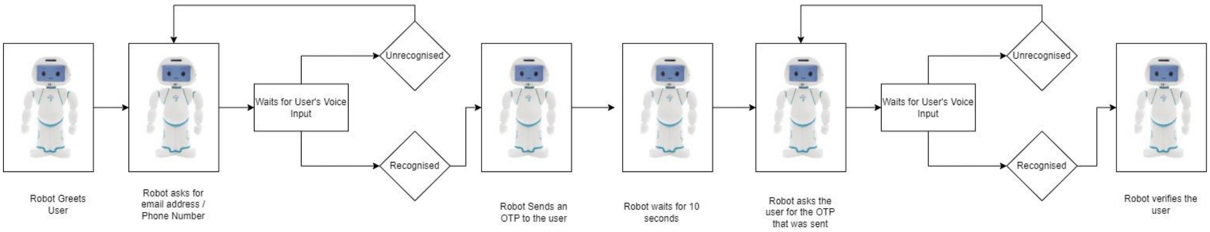

Robot Greets User
Robot asks for email address / Phone Number
Waits for User's Voice Input
Unrecognised
Recognised
Robot Sends an OTP to the user
Robot waits for 10 seconds
Robot asks the user for the OTP that was sent
Waits for User's Voice Input
Unrecognised
Recognised
Robot verifies the user

Figure 17 Second Iteration: OTP Recognition

This second iteration is the current one that has been implemented. The robot starts by greeting the user by getting their name. After the name is confirmed, the user asks for their email address. During this section, there are checks in place to see if the user has been registered before or not. This is to make sure that duplicates will not happen. This does not include checks if the user has multiple email addresses. After confirming the email address, the robot will send the user an OTP via the main email address that has been preset. The robot will then wait for a bit to allow for the OTP to be sent. The robot will then ask for the OTP that was sent to the user's email address that was stated. The user will then state the OTP and if the robot recognises the OTP, access will be granted to the user if not the robot will reset the process.

5.1.2 Design Principles

5.1.2.1 YAGNI

Plosch et al. (2016) states that YAGNI (You aren't gonna need it) is a principle that is similar to KISS (Keep It Simple and Sweet) where solutions are focused to not be overloaded by unnecessary functionality. This is to ensure that additional code should only be added to a software if the feature is necessary at it's time.

Since it stems from KISS in a way, solutions should not be complex, and the built software should be simple and easy to understand. This allows for cost savings and allow for the software that is built to be efficient.

5.1.3 Technologies

Every program that is built is based on some sort of technology. Every GUI, CMD line-based tools, games, media application and much more are normally built from scratch. This section explains the technologies that are being used within this project that helps make the project comes to life. This section also explains some technology that is used to make developing much easier on the developer that saves time and money as well.

5.1.3.1 Python

Figure 18 Python Logo (Python, n.d.)

Python is one of the fastest growing programming language out there (Srinath, 2017). This is because it is a suitable language for newcomers who wants to learn a programming language and for experts in the field. With python being very user-friendly and object oriented, it is very popular amongst beginners.

Python was one of the three choices that was available to use with ROS. The other two choices were C++ and JavaScript. Python was chosen at the end of the day due to its flexibility and its availability of using offline recognition which is the basis of this project.

5.1.3.2 ROS

Figure 19 ROS Logo (ROS, n.d.)

ROS (Robot Operating System) allows for developer to build robot applications. ROS is a set of software libraries and tools. ROS is open source and has an active community. It is also open source which makes it great for developers to work on it if they wish to. ROS is also modular and extensible which allows for feature reduction or addition to suit a developer's need.

A client library for ROS is used called rospy. Rospy is also one of the main reasons on why python was chosen with this project. Rospy is a library for python for ROS (ROS, n.d.). This library allows to interface with ROS topics, subscribers and services. This is the main functionality that allows for communication with the QTrobot in this project.

5.1.3.3 Docker

Figure 20 Docker (Docker, n.d.)

Docker is a platform for developers where they are able to develop, ship and run applications in a containerised environment (Docker, n.d.). Docker allows for portability, reproducibility, security and efficiency. This can be seen as docker images are portable which allows them to be deployed in various environment. Other than that, if the same image is deployed, they are reproducible. This is because it always create the same environment as stated. Since each image have its own filesystem and network, they are relatively secure as they cannot interact with each other. Lastly, docker is lightweight, which means multiple applications can be deployed quickly if needed to.

In this project, a docker-compose file was made specifically for testing. The docker-compose file can be seen within the appendix here. The file consists of two programs which are the database and a GUI for the database. More information regarding the database used in development can be found after this section. The docker-compose file also creates a volume which allows for both programs to interact with each other and gives persistence. This is to ensure that when the file is relaunched, the data within the database is not deleted on each run.

5.1.3.4 Database

Figure 22 PostgreSQL Logo (PostgreSQL, n.d.)

Figure 21 PgAdmin Logo (PgAdmin, n.d.)

Databases are a collection of data that is stored in a computer system electronically. It is usually controlled by a DBMS (Database management system). A database system is often shortened as a database which includes the data, DBMS and the application that is associated with them. To simplify, a database is used to store all kinds of data (Oracle, n.d.).

PostgreSQL is the DBMS that is used within this project during development. There are multiple choices to choose from but some of the main reasons why this was chosen was due to it being open source and its stability (PostgreSQL, n.d.). It is also feature-rich and supports a wide range of features. It also uses SQL (Structured Query Language) as its query language which is easy to use and understand.

Alongside PostgreSQL, pgadmin4 was used as well as the GUI for it. PgAdmin is used to manage PostgreSQL databases and it is a powerful tool (PgAdmin, n.d.). It is not necessary for this tool for PostgreSQL to function, but it is able to make development easier and faster while querying the databases within Postgres.

Although this project was tested and done in PostgreSQL, it is not a requirement to use it specifically and any other DBMS should work with the program as the database connection file is separate and can be plugged in and modified to work with other DBMS.

5.2 Overview Development

At the end of this project's lifecycle, this project will have added another method of authentication with the social robot. This is with the proposal of a two-factor authentication protocol using speech recognition. It is stated that most of the systems out there that involves social robots interacts with anyone but on a high level (Yaacoub et al., 2021) (Miller et al., 2018). The user only needs to speak out a 6 digit code to be able to authenticate themselves which makes it convenient as seen in the user testing later in this report.

The proposed protocol will be implemented within the development section below and a video demonstration can be found here as well. The implementation is then checked with various test methods from user testing to developer testing.

5.3 Deviation in Original Planning

This section is to explain the deviation that has happened compared to the original gantt chart that was proposed. The gantt chart that was proposed early on did not take into account of university vacation day. Comparing the dates however, the deviation starts from sprint 6 in the old chart that the robot is able to authenticate the user but in reality, this only happened in sprint 10 in the current iteration. Portfolio has a lot of deviation and delays which can be seen in the kanban board in the appendix here.

5.4 Development

This section discusses the development process over the course of the project's lifespan. Since this project was done based on the Agile Methodology, the development section is split into sprints. The development process took place over 12 sprints not including public holidays. There is a stark change within the middle of the project as stated above as the project's development steered into speech recognition. Each of the sprints has a main title to associate what was the big update during said sprint.

There will be a code overview section at the end with a video sample on how the project works at the end of sprint 11. This is because due to the nature of the project, there is not many screenshots that can be shown during the development process.

5.4.1 Sprint 0 – Early Project Planning

Sprint 0 runs from the 6th October till 20th October. The project initiation document (PID) was made that includes the title of the project, project vision, risks, Gantt chart. This initial document will be the basis for the whole project which shows where the project would head in the future and acts as a stepping stone. The PID document can be seen here. The GitHub Repository has also been made during this sprint. A Trello board has also been set up to allow for management of tasks and within Trello automations has been set up for automatic labelling.

5.4.2 Sprint 1 – Testing Robot's Capabilities

Sprint 1 runs from 20th October till 3rd November 2022. During this sprint the QTrobot was checked to make sure that it can turn on and off its microphone via the ROS service. The base code was pushed with a template. It was here where there was a Linux GUI issue which did not allow me to run the terminal or any system apps. (Images can be found in the Links section of this document) It was then fix within a week.

5.4.3 Sprint 2 – More in-depth Researching

Sprint 2 runs from the 3rd of November 2022 until the 17th November 2022. This sprint went into more research into Audio Recognition. Google's speech recognition library and Amazon's Rekognition was also checked accordingly to see which is more suitable for this project. Project Planning for the base designs for the enrolment phase and the recognition phase has been made using draw.io. The internet issue that has been persistent has been fixed during this time. Details about the internet issue can be found here.

5.4.4 Sprint 3 – Learning Speech Recognition

Sprint 3 runs from 17th November 2022 until the 1st December 2022. This sprint consisted of a meeting with Dr Hooman from the University of Hertfordshire on the 18th November. This meeting consisted of information on Visual and Spoken interfaces, Dealing with Voice Biometric. There were talks on Fourier Transformation, Frequency domain and time domains. This meeting helped with the base understanding for Speech Recognition. S3 by Amazon was chosen as the file server to store the audio recordings. Base first designs were made for the infrastructure section of the report and was written out. Sprint Dates were also readjusted to follow the winter holidays.

5.4.5 Sprint 4 – Minimum Viable Product (MVP)

Sprint 4 runs from the 1st of December till the 15th of December 2022. At the end of this sprint, there will be a showcase to the supervisor that shows the progress of what has been done over the past 4 sprints. One of the main developments of the project that can be stated is that the robot is able to save the audio that is recorded. The data is stored locally. Other than that, considerations on what data that needs to be stored has been taken into account. Within the same spectrum, the user would be able to know when they are being recorded. Other competitors and / or similar projects has been checked out and written within this report here. A quick PowerPoint presentation was made up to show the concept that has been done as well. The PowerPoint presentation can be found over here.

5.4.6 Sprint 5 – Documentation Work

Sprint 5 runs from 9th January to 23rd January. There is a time jump between these two sprints due to the fact that it was the winter vacations. Documentation has been updated during this sprint. Meeting Minutes has been reworked to be easier to read and to prepare for insertion within the documentation. References has been reworked to follow Harvard referencing instead of short links. There has been no major development to this project due to other modules taking over most of the time. Main heading for Sprint Review section has been written out here as well.

5.4.7 Sprint 6 – Testing Preparation + Documentation

Sprint 6 runs from 23rd January to 6th February. This sprint, Ethical approval forms have been checked. Document has been updated with more hyperlinks and have surpassed 2000 words within this sprint. Testing methodology draft was set up at this point to prepare for testing.

5.4.8 Sprint 7 – Voice Recognition -> Speech Recognition

Sprint 7 runs from 6th February to 20th February. During this sprint, plans for downscaling the project was made here. This is due to the fact that there were some unknown issues with boto3's API. This caused some inconsistency on when it will run. Timeline from here on out has been affected slightly.

5.4.9 Sprint 8 – OTP Authentication + Poster Designs for showcase

Sprint 8 runs from 20th February till 6th March. During this sprint, the poster designs were started to be made. This includes with checking what topics and text would need to be on the poster. This is for the March submission. Next, program designs were made as well. Email uploading functionality have been started. OTP (One Time Passcode) functionality has also been implemented. During this sprint, the developer decided to learn docker in full to make development easier.

5.4.10 Sprint 9 – Database Authentication

Sprint 9 runs from the 6th of March to 20th March. With this sprint, the database server (PostgreSQL) is made and tested. The email server was then put and set up but due to some issues the self-hosted email server was than not used within the project after this. Project Vision was then updated with the new changes to the project that was done. Finally, for the next submission there was a poster deadline and the template for it was started.

5.4.11 Sprint 10 – Poster Completion

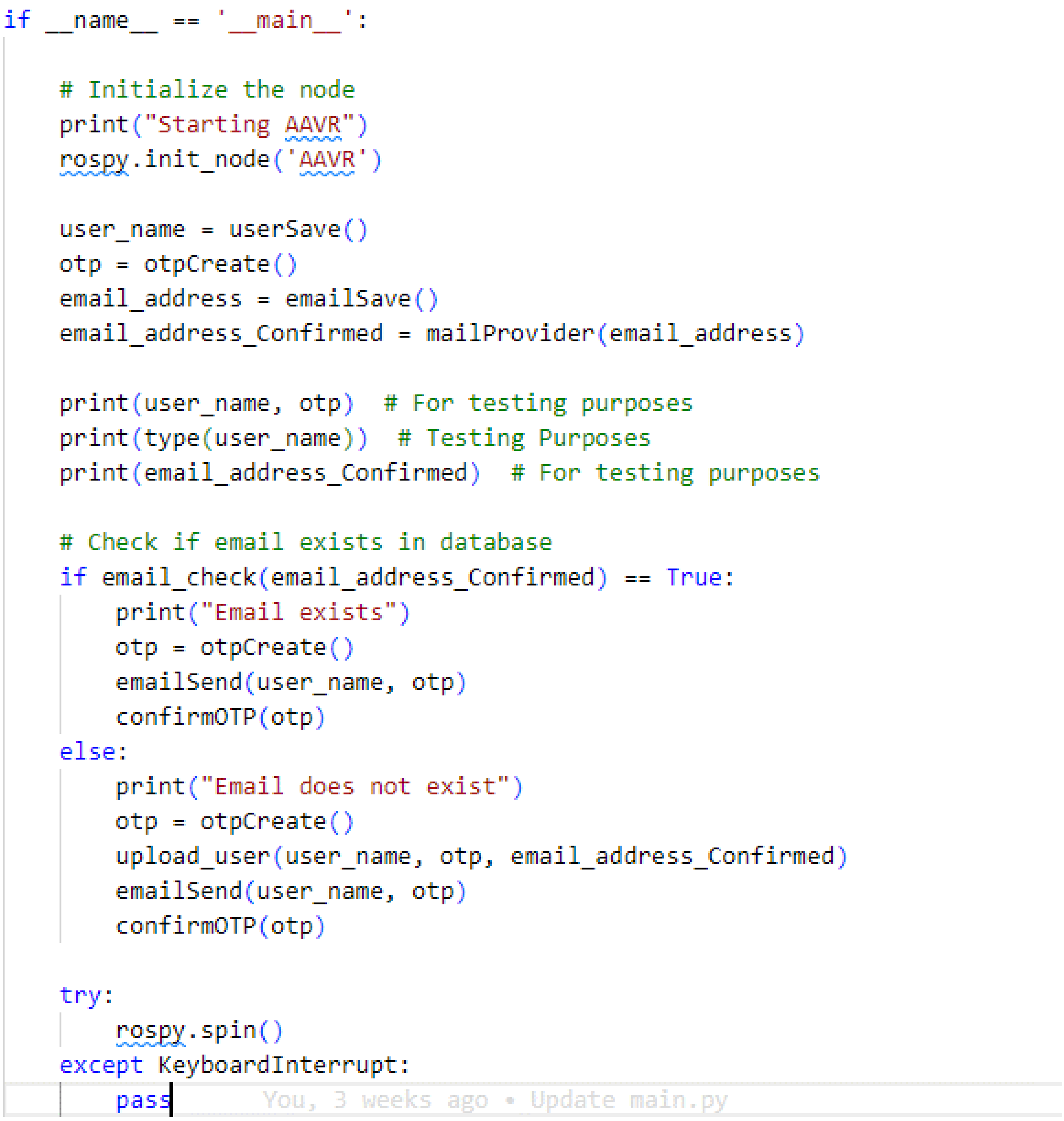
Sprint 10 runs from the 20th March till 3rd April. This was a short sprint due to the deadlines for the other modules that the author was taking. The poster was completed and submitted on time. The description for the website and the poster was also written down and uploaded to the DLE on time. This report was then sent in for checks during this sprint.

5.4.12 Sprint 11 – User Testing + Video + Project's finalisation

Sprint 11 runs from 24th April till 8th May. This sprint starts after the Spring Break. This is why there is a jump between the last sprint and this sprint. This sprint will also be the final spring in this project's development. The report is completed during this sprint in preparation for submission. The video is also recorded during this sprint. The video includes the background to the project and the program running on the robot in real time. User testing is also completed during this time to have a better understanding from an end user's perspective. The database is now connected with the program and is able to store and check for user emails. Audio that is recorded within the robot can be transcribed and is being processed correctly.

5.4.13 Code Overview

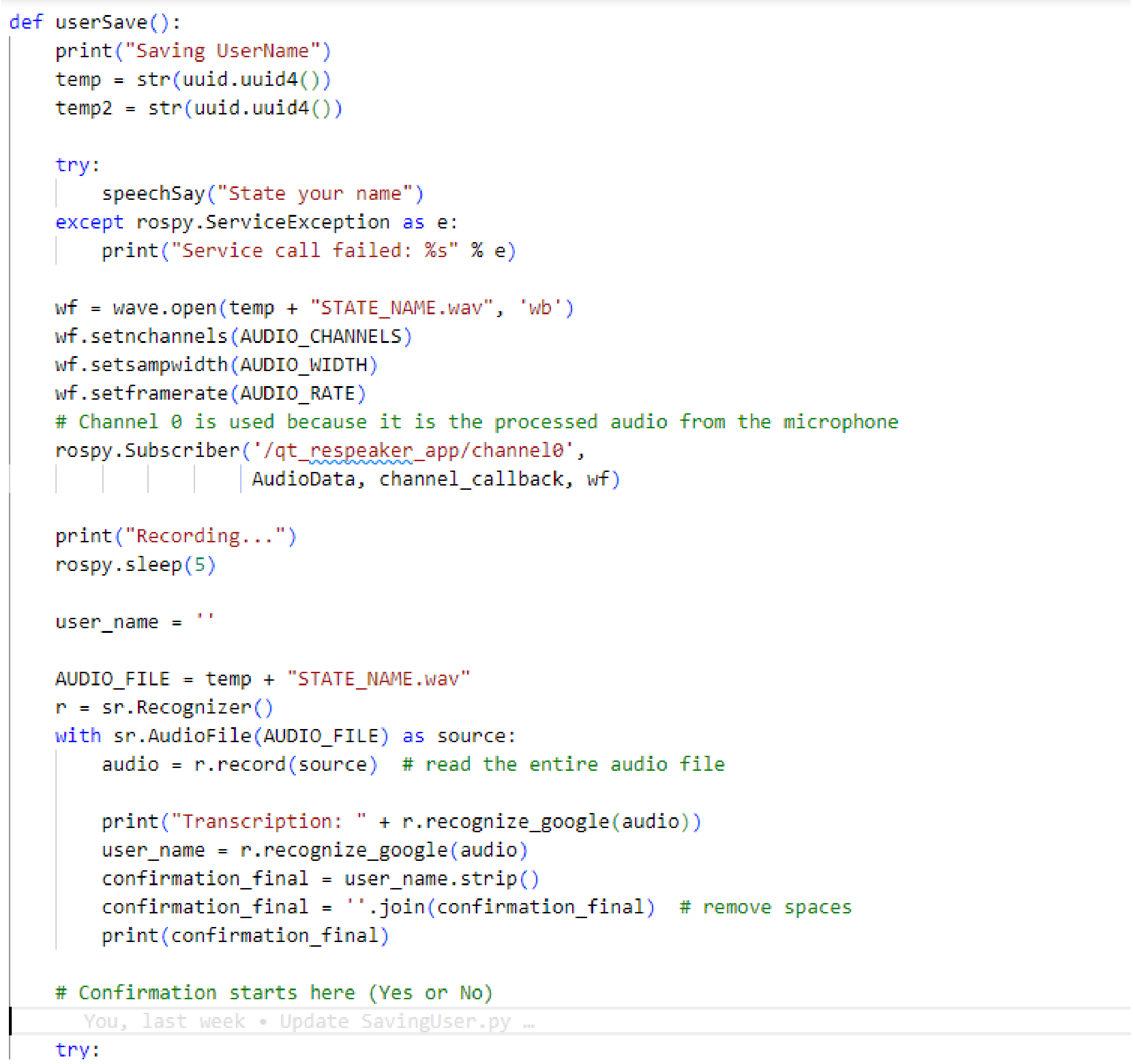
To run the project, there is a number of requirements that needed to be done before hand. The full user guide can be found in the appendix here. The code is split up into multiple python files to allow for scalability and improvement. TT



```
if __name__ == '__main__':

    # Initialize the node
    print("Starting AAVR")
    rospy.init_node('AAVR')

    user_name = userSave()
    otp = otpCreate()
    email_address = emailSave()
    email_address_Confirmed = mailProvider(email_address)

    print(user_name, otp)  # For testing purposes
    print(type(user_name))  # Testing Purposes
    print(email_address_Confirmed)  # For testing purposes

    # Check if email exists in database
    if email_check(email_address_Confirmed) == True:
        print("Email exists")
        otp = otpCreate()
        emailSend(user_name, otp)
        confirmOTP(otp)
    else:
        print("Email does not exist")
        otp = otpCreate()
        upload_user(user_name, otp, email_address_Confirmed)
        emailSend(user_name, otp)
        confirmOTP(otp)

    try:
        rospy.spin()
    except KeyboardInterrupt:
        pass        You, 3 weeks ago • Update main.py
```

Figure 23 main.py File

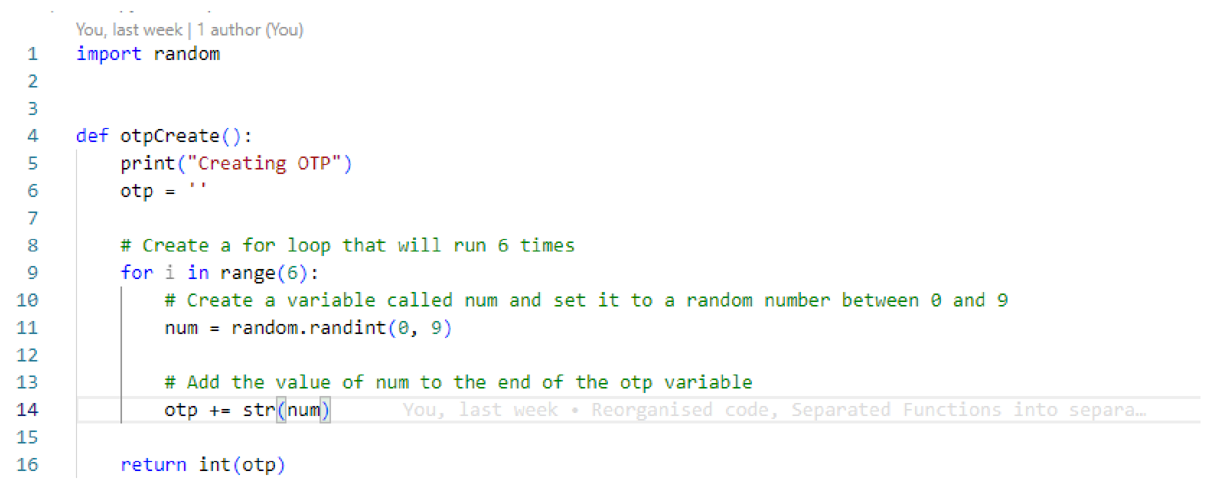
The project starts out with the main function with initialising the nodes. It starts out by going to the userSave function to ask the user about their name. This is the function where the user is allowed to state their name to the robot. It then generates a one-time password (OTP) that consists of 6 digit number with the otpCreate function. The user is then requested to give their email address without the provider. This is to allow for the program itself to append the mail provider later with the mailProvider function. It is then checked within the database (in this case on PostGres) if the email exists or not. This is a simple check to ensure no duplicates are allowed in the database. If the email exists, the OTP will be sent using the emailSend function and the user is allowed to confirm the otp with the confirmOTP function. If in the case that the user's email address is not on the database, the upload_user function will upload the user's name and email address.



```
def userSave():
    print("Saving UserName")
    temp = str(uuid.uuid4())
    temp2 = str(uuid.uuid4())

    try:
        speechSay("State your name")
    except rospy.ServiceException as e:
        print("Service call failed: %s" % e)

    wf = wave.open(temp + "STATE_NAME.wav", 'wb')
    wf.setnchannels(AUDIO_CHANNELS)
    wf.setsampwidth(AUDIO_WIDTH)
    wf.setframerate(AUDIO_RATE)
    # Channel 0 is used because it is the processed audio from the microphone
    rospy.Subscriber('/qt_respeaker_app/channel0',
                     AudioData, channel_callback, wf)

    print("Recording...")
    rospy.sleep(5)

    user_name = ''

    AUDIO_FILE = temp + "STATE_NAME.wav"
    r = sr.Recognizer()
    with sr.AudioFile(AUDIO_FILE) as source:
        audio = r.record(source)  # read the entire audio file

        print("Transcription: " + r.recognize_google(audio))
        user_name = r.recognize_google(audio)
        confirmation_final = user_name.strip()
        confirmation_final = ''.join(confirmation_final)  # remove spaces
        print(confirmation_final)

    # Confirmation starts here (Yes or No)
      You, last week • Update SavingUser.py …
    try:
```

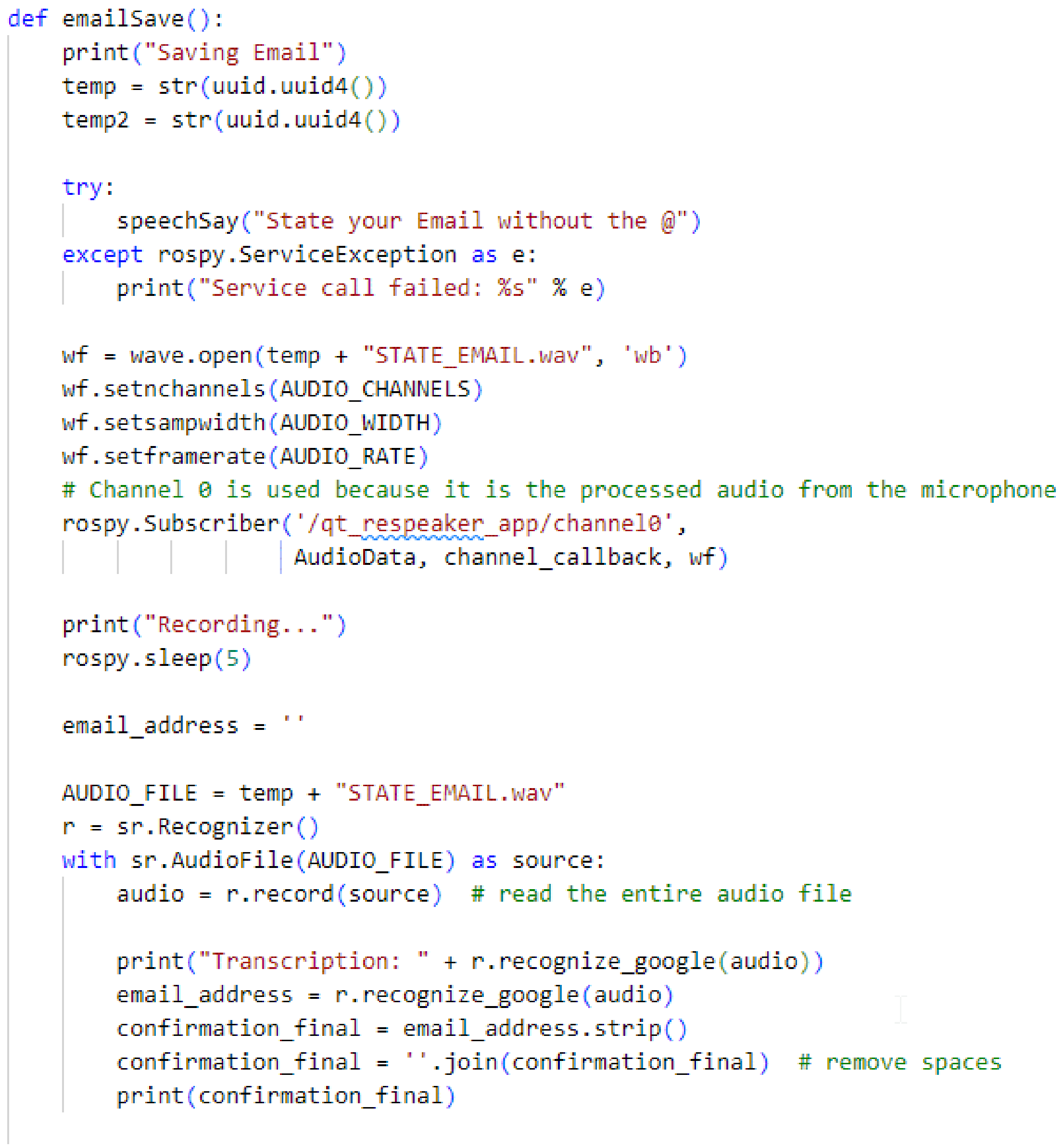
Figure 24 SavingUser.py (userSave() partial)

The temporary variables that will show up in the code is used to store the recording names. This will allow for Google's speech recognition to function. All recognised speech will combine the words together as a check as during testing, the transcript leads to having gaps within the words. The confirmation will check if the user is happy with the name that is being said by the robot.



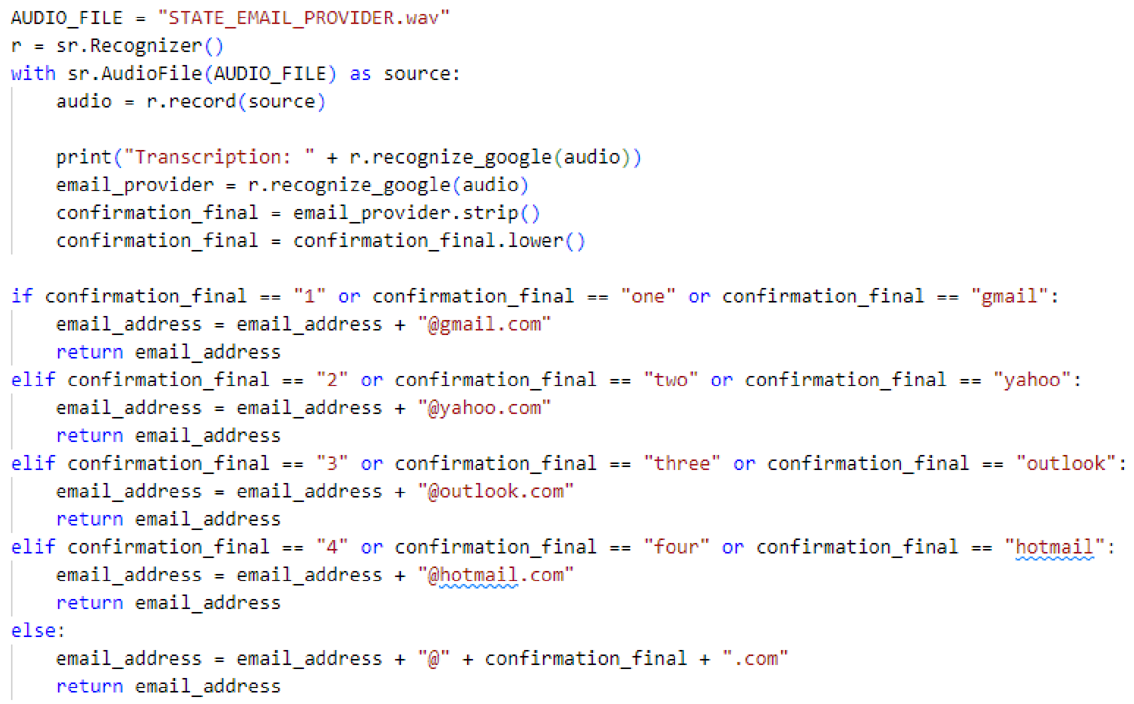
```
You, last week | 1 author (You)
import random


def otpCreate():
    print("Creating OTP")
    otp = ''

    # Create a for loop that will run 6 times
    for i in range(6):
        # Create a variable called num and set it to a random number between 0 and 9
        num = random.randint(0, 9)

        # Add the value of num to the end of the otp variable
        otp += str(num)        You, last week • Reorganised code, Separated Functions into separa…

    return int(otp)
```

Figure 25 otpCreate.py (otpCreate())



```
def emailSave():
    print("Saving Email")
    temp = str(uuid.uuid4())
    temp2 = str(uuid.uuid4())

    try:
        speechSay("State your Email without the @")
    except rospy.ServiceException as e:
        print("Service call failed: %s" % e)

    wf = wave.open(temp + "STATE_EMAIL.wav", 'wb')
    wf.setnchannels(AUDIO_CHANNELS)
    wf.setsampwidth(AUDIO_WIDTH)
    wf.setframerate(AUDIO_RATE)
    # Channel 0 is used because it is the processed audio from the microphone
    rospy.Subscriber('/qt_respeaker_app/channel0',
                     AudioData, channel_callback, wf)

    print("Recording...")
    rospy.sleep(5)

    email_address = ''

    AUDIO_FILE = temp + "STATE_EMAIL.wav"
    r = sr.Recognizer()
    with sr.AudioFile(AUDIO_FILE) as source:
        audio = r.record(source)  # read the entire audio file

        print("Transcription: " + r.recognize_google(audio))
        email_address = r.recognize_google(audio)
        confirmation_final = email_address.strip()
        confirmation_final = ''.join(confirmation_final)  # remove spaces
        print(confirmation_final)
```

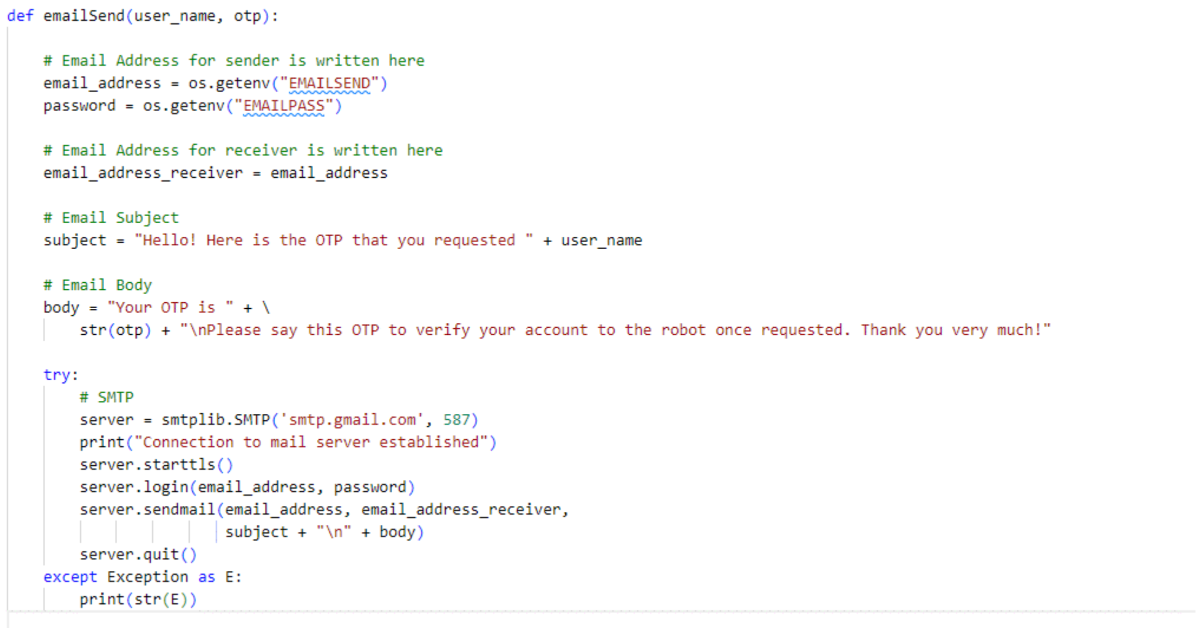
Figure 26 SavingEmail.py (emailSave())



```
AUDIO_FILE = "STATE_EMAIL_PROVIDER.wav"
r = sr.Recognizer()
with sr.AudioFile(AUDIO_FILE) as source:
    audio = r.record(source)

    print("Transcription: " + r.recognize_google(audio))
    email_provider = r.recognize_google(audio)
    confirmation_final = email_provider.strip()
    confirmation_final = confirmation_final.lower()

if confirmation_final == "1" or confirmation_final == "one" or confirmation_final == "gmail":
    email_address = email_address + "@gmail.com"
    return email_address
elif confirmation_final == "2" or confirmation_final == "two" or confirmation_final == "yahoo":
    email_address = email_address + "@yahoo.com"
    return email_address
elif confirmation_final == "3" or confirmation_final == "three" or confirmation_final == "outlook":
    email_address = email_address + "@outlook.com"
    return email_address
elif confirmation_final == "4" or confirmation_final == "four" or confirmation_final == "hotmail":
    email_address = email_address + "@hotmail.com"
    return email_address
else:
    email_address = email_address + "@" + confirmation_final + ".com"
    return email_address
```

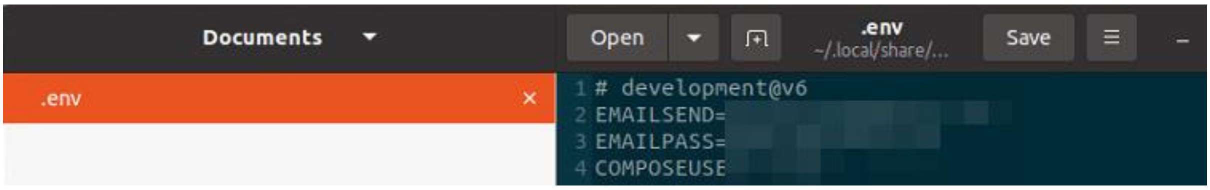
Figure 27 MailProvider.py (mailProvider() partial)

This file was to ensure that the user is able append the right email address with just one word. Some of the most popular email addresses will be recognised immediately. If the user has a custom email domain, the robot has a tendency to not understand it properly. This has been tested with the Plymouth University's student email which can not be understood every time.



```
def emailSend(user_name, otp):

    # Email Address for sender is written here
    email_address = os.getenv("EMAILSEND")
    password = os.getenv("EMAILPASS")

    # Email Address for receiver is written here
    email_address_receiver = email_address

    # Email Subject
    subject = "Hello! Here is the OTP that you requested " + user_name

    # Email Body
    body = "Your OTP is " + \
        str(otp) + "\nPlease say this OTP to verify your account to the robot once requested. Thank you very much!"

    try:
        # SMTP
        server = smtplib.SMTP('smtp.gmail.com', 587)
        print("Connection to mail server established")
        server.starttls()
        server.login(email_address, password)
        server.sendmail(email_address, email_address_receiver,
                        subject + "\n" + body)
        server.quit()
    except Exception as E:
        print(str(E))
```

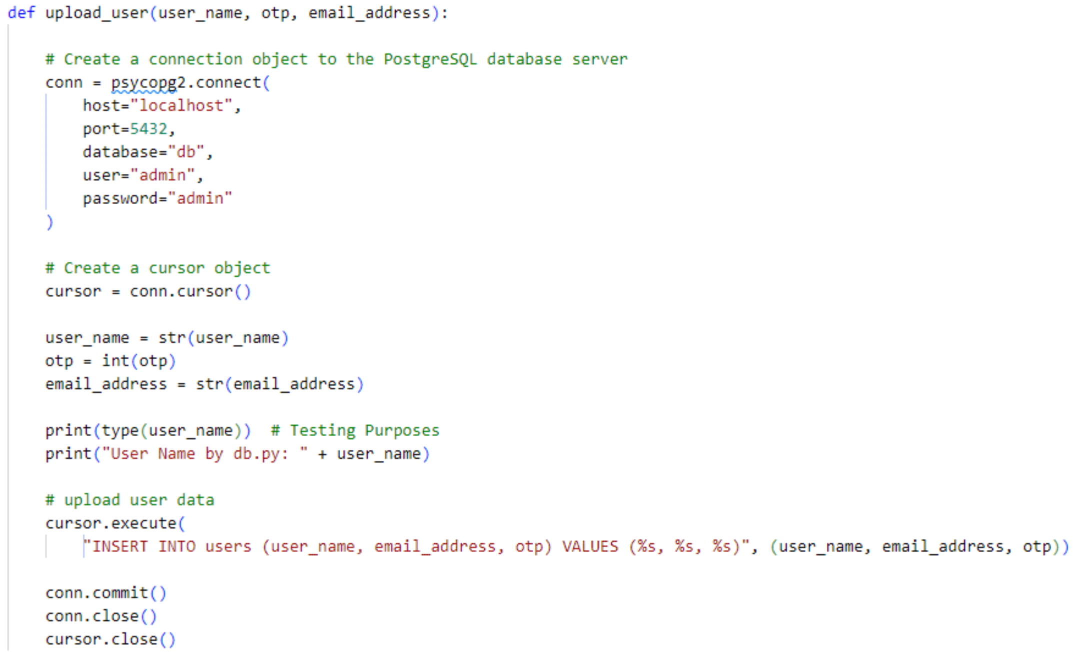
Figure 28 email_send.py (emailSend(user_name, otp))



```
# development@v6
EMAILSEND=[redacted]
EMAILPASS=[redacted]
COMPOSEUSE
```

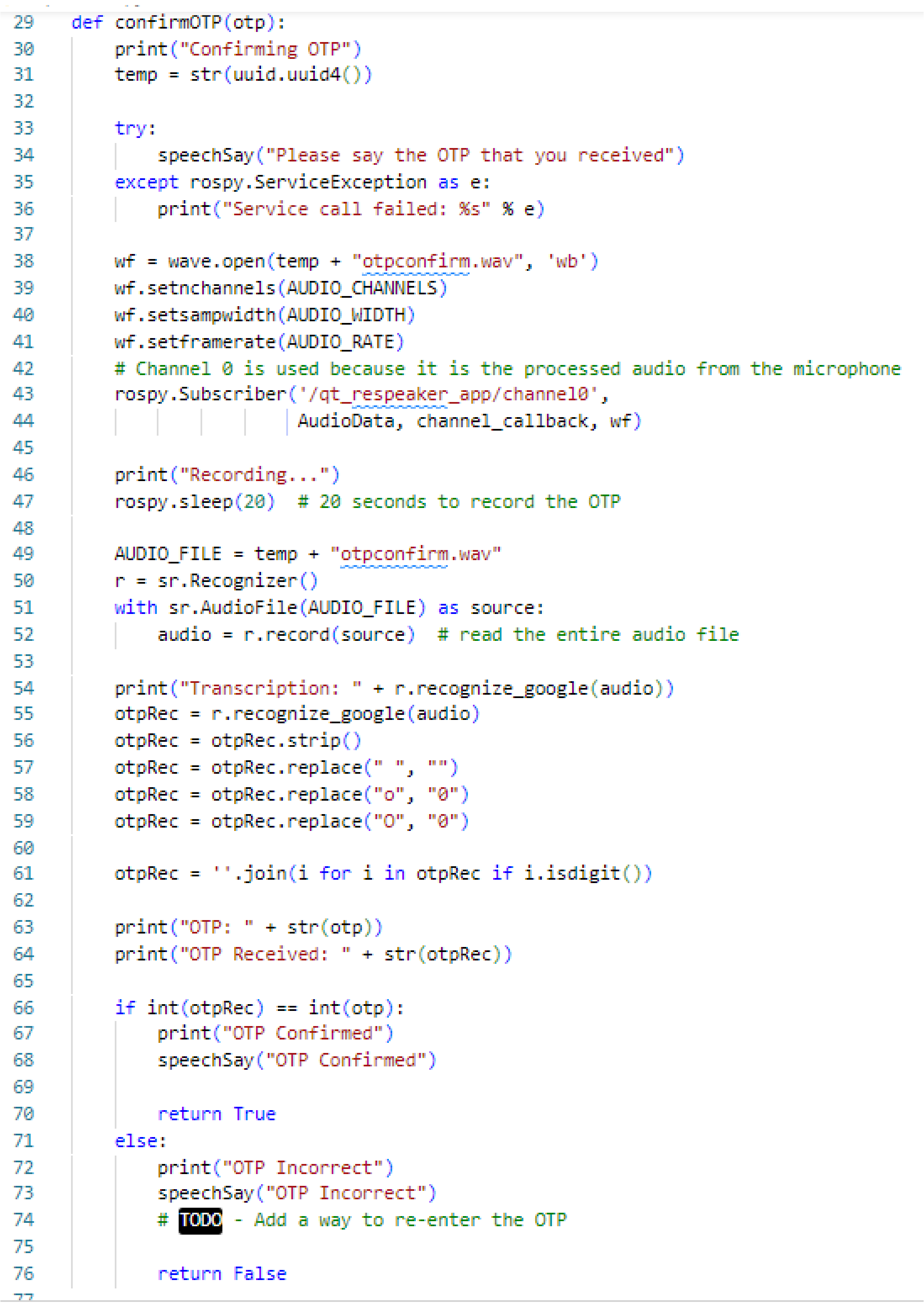
Figure 29 .env File

Environment Variables are used within this file to ensure anonymity. The .env file currently is not shared because the developer is currently using their personal email address to send the email out when requested. This file can be easily edited and the template of the file will be given out if needed. Dotenv is used to allow for environment variables to be synced (Dotenv, n.d.).



```
def upload_user(user_name, otp, email_address):

    # Create a connection object to the PostgreSQL database server
    conn = psycopg2.connect(
        host="localhost",
        port=5432,
        database="db",
        user="admin",
        password="admin"
    )

    # Create a cursor object
    cursor = conn.cursor()

    user_name = str(user_name)
    otp = int(otp)
    email_address = str(email_address)

    print(type(user_name))  # Testing Purposes
    print("User Name by db.py: " + user_name)

    # upload user data
    cursor.execute(
        "INSERT INTO users (user_name, email_address, otp) VALUES (%s, %s, %s)", (user_name, email_address, otp))

    conn.commit()
    conn.close()
    cursor.close()
```

Figure 30 db.py (upload_user(user_name, otp, email_address))



```python
def confirmOTP(otp):
    print("Confirming OTP")
    temp = str(uuid.uuid4())

    try:
        speechSay("Please say the OTP that you received")
    except rospy.ServiceException as e:
        print("Service call failed: %s" % e)

    wf = wave.open(temp + "otpconfirm.wav", 'wb')
    wf.setnchannels(AUDIO_CHANNELS)
    wf.setsampwidth(AUDIO_WIDTH)
    wf.setframerate(AUDIO_RATE)
    # Channel 0 is used because it is the processed audio from the microphone
    rospy.Subscriber('/qt_respeaker_app/channel0',
                     AudioData, channel_callback, wf)

    print("Recording...")
    rospy.sleep(20)  # 20 seconds to record the OTP

    AUDIO_FILE = temp + "otpconfirm.wav"
    r = sr.Recognizer()
    with sr.AudioFile(AUDIO_FILE) as source:
        audio = r.record(source)  # read the entire audio file

    print("Transcription: " + r.recognize_google(audio))
    otpRec = r.recognize_google(audio)
    otpRec = otpRec.strip()
    otpRec = otpRec.replace(" ", "")
    otpRec = otpRec.replace("o", "0")
    otpRec = otpRec.replace("O", "0")

    otpRec = ''.join(i for i in otpRec if i.isdigit())

    print("OTP: " + str(otp))
    print("OTP Received: " + str(otpRec))

    if int(otpRec) == int(otp):
        print("OTP Confirmed")
        speechSay("OTP Confirmed")

        return True
    else:
        print("OTP Incorrect")
        speechSay("OTP Incorrect")
        # TODO - Add a way to re-enter the OTP

        return False
```

Figure 31 otpConfirm.py (confirmOTP(otp))

The video demonstration of the finished outcome can be seen here. Not all the code is shown within this documentation. This can be seen at the GitHub Repository here.

5.5 Project Testing

Project Testing is done during the development of this project. Testing allows for any changes that needs to be made based on user's suggestions. It also provides the opportunity for the developer to learn what the user's interests are.

5.5.1 User Testing

User testing is done through a google form. This allows for the developer to easily understand and interpret the results that is provided. The users are provided with a video of how the interaction of the robot would work. The set of questions are asked as follows

- Would you rather talk to a receptionist directly or to a robot?
- Would you trust the robot with your data? (Eg, Name & Email Address)
- What is the convenience factor with using the robot in your opinion?
- How easy it is to authenticate yourself to the robot in your opinion?
- How natural in your opinion do you think authenticating yourself to the robot is?

The questions that were asked is to further understand people's opinion on having a robot as part of their lives. The questions were also tailored to figure if the user stories have been achieved or not. The following charts shows what are the responses to the questions above.

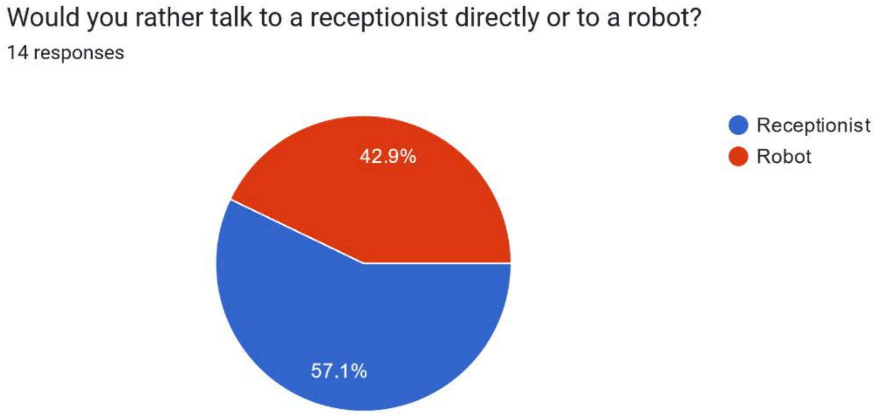
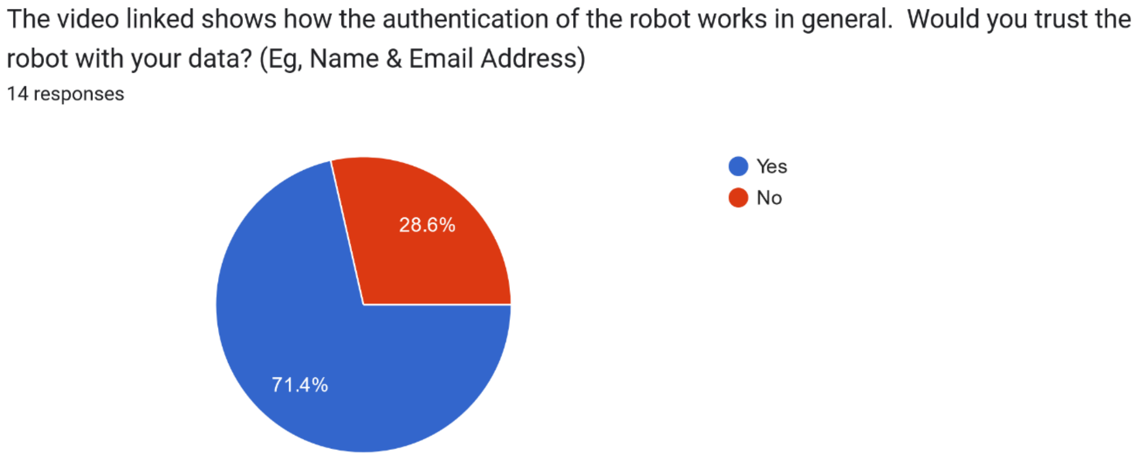


Figure 32 Question 1 Chart Spread

This question shows that the majority of people would rather talk to a receptionist than to a robot. A follow up question was asked as an open ended question on why they picked the answer that they did. Almost all of them who answered receptionist was due to it being faster and more accurate. On the other hand, the people who answered robot was cause they want less human interaction and to some it is less stressful.



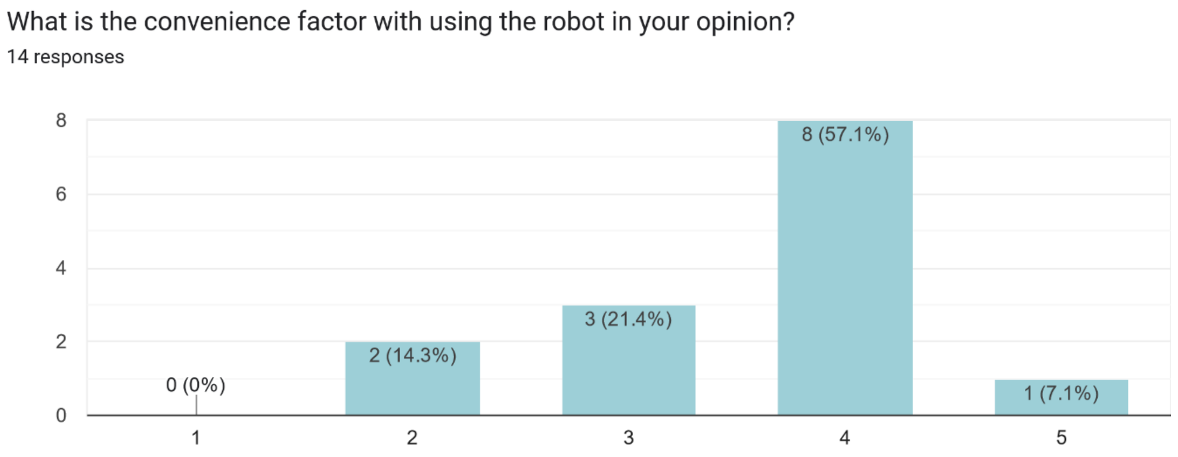
The video linked shows how the authentication of the robot works in general. Would you trust the robot with your data? (Eg, Name & Email Address)

14 responses

Figure 33 Question 2 Chart Spread

The question is asked to check what is the general reception of the public if the user were to talk with a robot to authenticate themselves with the robot. The youtube video that was shown to the user is linked here.

It can be evaluated that although the amount of people that will rather talk to the receptionist. It can be seen that they do trust the robot with their data. In some of the additional comments, regarding this question specifically, it is dependent on if the robot is connected to the internet or not and how the data is being used.

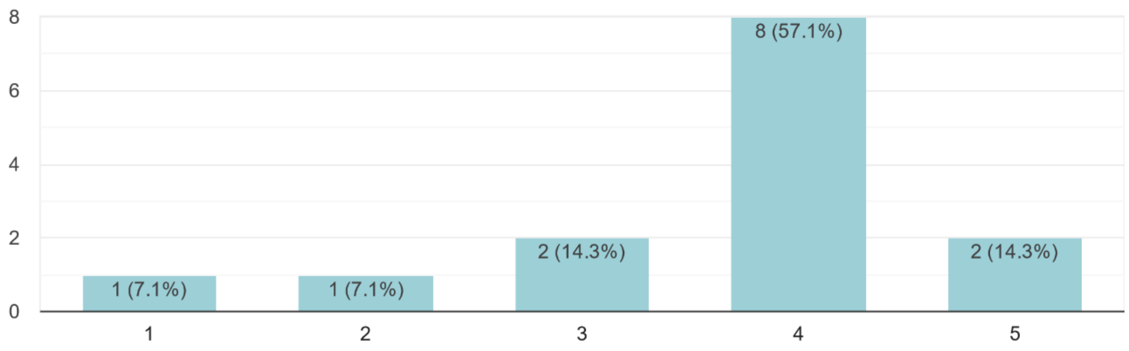


What is the convenience factor with using the robot in your opinion?

14 responses

Figure 34 Question 3 Chart Spread

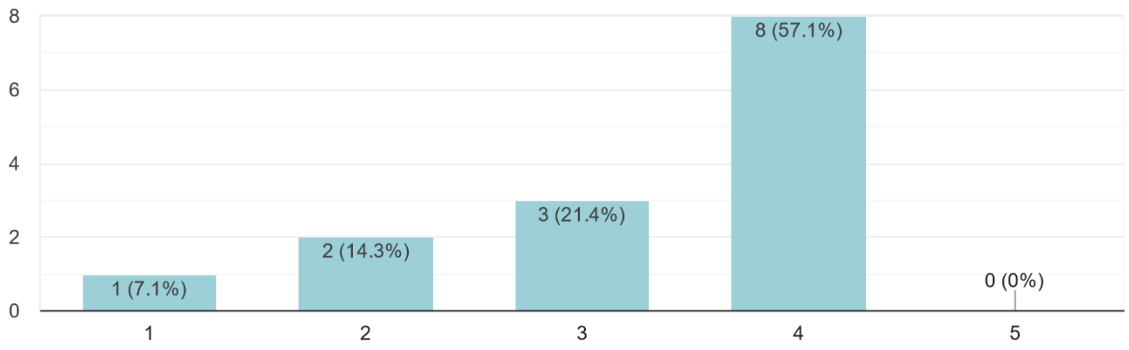
This question was asked to answer one of the non-functional requirements. This is to check if the program that was designed to ensure if it is convenient to talk to the robot or not. It can be seen that majority of the participants do enjoy the convenience of talking to the robot if time was not of a factor. This can be seen in the additional comments that the robot takes a while to respond to the user.



How easy it is to authenticate yourself to the robot in your opinion?

14 responses

This question was asked to answer the non-functional requirement that was set out. It can be seen that similar to the question above, majority of the participants agrees that the authentication process to the robot was relatively easy. It has to be kept in mind that the participants were only seen a video of the process demonstrating how the authentication process works and they are not able to enter the room where the robot was stationed at.



How natural in your opinion do you think authenticating yourself to the robot is?

14 responses

This question was asked to answer two of the user stories that was set in the project. It can be seen that the user stories were achieved but only by a margin. Due to how the robot's microphone works. The filter that was set to eliminate background noise makes it so that the user needs to be clear in when they are talking to the robot. This can be seen in the lower rated responses where the participants thinks its not natural to be talking like that compared to talking to a human receptionist.

The full test results can be found within the appendix here.

5.5.2 Developer Testing

These tests that is within this section are the tests that the developer has made for the purposes of ensuring that the software that is developed is working. The tests here are a number of tests that a general user might do while interacting with the robot.

5.5.2.1 Functional Testing

Functional testing is done to ensure that the functional requirements that have been met are working as intended.

The following table shows the tests that were done to test the specific requirements.

Requirements	Checks	Expected Result	Actual Result
Robot is able to respond to user	Dev says their name when requested. Dev confirms their name to the robot. Dev declines confirmation their name to the robot	Robot listens and replies. Robot listens and replies. Robot listens and replies.	As expected, As expected, Errors out and stops the program
Robot is able to send out an OTP to the user's email	Email received on dev's email when stated	Email gets sent with OTP	As expected,
Robot is able to confirm OTP that is stated	OTP stated by dev can be received Wrong OTP stated by dev can be received	OTP Confirmed OTP Denied, asks for reconfirmation	As expected, User don't get access, but reconfirmation does not go through
Robot is able to check if user has an account	If user has an account, don't reupload their details to the database If user does not have an account, upload their details	Robot doesn't upload the duplicates Robot uploads the details to the database	As expected, As expected,

Table 3 Functional Testing

It can be deduced that most of the functional requirements have been met.

5.5.2.2 Non-Functional Testing

The testing procedure for non-functional testing are similar to 5.5.2.2, where tests are made to ensure that the non-functional requirements have been met.

The following table shows the tests that were done to test the specific requirements.

Requirements	Checks	Expected Result	Actual Result
Program needs to be easy to use	Survey Data with users	Overall scores > 4	As expected,
Program must be simple to set up	Setting up the program from scratch	Easy to set up	Has a few requirements that needs to be met before

Table 4 Non-Functional Requirements

Not all the non-functional requirements are working as intended at this point of time. Based on the user surveys that is done although on the user's point of view

5.6 Alternative Solutions

Solutions that are listed here are alternative solutions that can be implemented within this project but is not implemented. The solutions here will explain on why these solutions might be useful.

5.6.1 OTP via SMS

People who uses the system might not have an email address. Although this is unlikely as most social media in this day and age needs an email address to create an account. It is good to cover all basis. The method of authenticating via SMS might be more useful to some to those who does not have an email address. This allows for anyone with a mobile phone number to be able to authenticate themselves. However, it is needed to keep in mind that SMS is not usually the safest. According to Mulliner et al. (2013), SMS OTP relies on the service provider. Attacks against 3G networks have shown that SMS messages does not have confidentiality. Other than that, there are specialised trojans on mobile phones that have been created by criminals.

5.6.2 Using AI to recognise voices

Rekognition by Amazon was one of the main ways of using AI to recognise human voices (AWS, 2011). This method allows the AI to recognise a ton of voices and be able to understand their intonation. This was one of the early methods during this project to allow for the idea to come to fruition.

5.7 Major Issues

This section explains some of the major hurdles that have come up during the duration of the project. Major issues are defined as big changes that have delayed development of the project.

5.7.1 Linux GUI Issues

As seen in the pictures here, this issue stemmed from a botched update that could not be reverted properly. LuxAI had to be contacted to ask for advice on how to fix the issue. It was then resolved within a week or so. This happened early on within the project but due to the nature of the issue and the waiting time, not much was done during this time period.

5.7.2 Boto3 API for file uploading

This was the main reason why the project was switched from voice recognition to speech recognition. There is currently an unknown issue that could not be resolved where the API calls for uploading the recorded voice is not working as intended. With this, the rekognition API by Amazon can not be used.

5.8 Minor Issues

5.8.1 Internet Connectivity Issue with the QTrobot

The QTrobot had trouble connecting to the university's internet in the early stages of the project. At that time, having a hotspot is necessary to allow for the QTrobot to connect to the internet. This was then fix in Sprint 2 by not using the ethernet cable that was plugged in and using Wi-Fi only. I am not able to go into more detail due to the fact that there are some privacy concerns regarding this setup.

6 Discussion

6.1 Critical Evaluation

The project's goal was partially achieved. This is because the project is meant to be able to authenticate user conveniently with a social robot which it can do but due to some of the bugs that is currently within the project, in its current state it is not deployable. The objectives within this project have been partially met. AAVR is able to authenticate the user via email when requested but it is not able to recognise the user's voice and authenticate it off that.

The user testing shows that people are willing to trust the robot with their data if needed to but would prefer to talk to a human. This is due to the reliability and speed concerns of the robot. Majority of the participants thinks that its relatively natural to talk to the robot. Finally, from the user testing most participants thinks that it is easy enough to authenticate themselves to the robot. These shows that the user stories have been achieved.

The functional requirements are partially achieved as the program currently is not able to confidently decline statements stated by the user, this can be seen in the testing above. Other than that, if the OTP gets declined, there is currently no way for the user to re-input the OTP to authenticate. The non-functional requirements are partially achieved as well. The program currently have a number of requirements that needs to be set up before it is able to run. All of the use cases that have been set out have been achieved in this case.

During this project, new skills that were developed includes learning docker fully and integrating it within this project. Other than that, is with learning how ROS works with the robot. Before this, the developer has not heard of ROS before and was unaware of its existence.

6.2 Sprints Evaluation

Based on the sprints that have been documented, the sprints could have gone better with a better redesign after the changing of project. This can be seen where after Sprint 6, the number of tasks that needed to be done was pushed to the end quite a bit. More information regarding how the sprints went can be seen in the appendix here.

6.3 Post Evaluation

If the developer were given another go at doing this project, one of the main objectives that would have been done differently was not to use AWS within the integration of the project early on. This would have saved more time in implementation. Another one is to allow user to input multiple email addresses. This is because some users might have multiple emails and since currently the robot points a user to one name it is not able to currently associate different emails to a singular user.

7 Conclusion

To summarise this report, the project's goals that was set after the change was mostly achieved. The solution that currently works allows the user to authenticate themselves with the social robot with the use of their personal device. The authentication is done via 2FA with the code being sent to the user's email address. The program than compares and check if the 2FA is correct. Only then, this allows for the user to be able to receive their personal information with ease. This solution would work best in an environment where noise is not much of an issue as it is heavily based upon the internal microphone's of the robot.

8 References

ACM Code of Ethics and Professional Conduct (n.d.) *Code of Ethics*. ACM. Available at: https://www.acm.org/code-of-ethics.

AWS (2011) *Amazon Rekognition*, *Amazon*. Available at: https://aws.amazon.com/rekognition/.

BCS Code of conduct (n.d.) *BCS*. BCS. Available at: https://www.bcs.org/membership-and-registrations/become-a-member/bcs-code-of-conduct/.

Braz, C. and Robert, J.-M. (2006) "Security and Usability: The Case of the User Authentication Methods," *Proceedings of the 18th international conference on Association Francophone d'Interaction Homme-Machine - IHM '06* [Preprint]. Available at: https://doi.org/10.1145/1132736.1132768.

Cole, G.J. (2021) *The evolution of Speech Recognition Technology*, *TechRadar*. TechRadar pro. Available at: https://www.techradar.com/news/the-evolution-of-speech-recognition-technology.

Docker (n.d.) *Docker docs: How to build, share, and run applications*, *Docker Documentation*. Available at: https://docs.docker.com/.

Dotenv (n.d.) *Simplify your secrets*, *Dotenv*. Available at: https://www.dotenv.org/.

Dybå, T., Dingsøyr, T. and Moe, N.B., 2014. Agile project management. *Software project management in a changing world*, pp.277-300.

GanttProject (no date) *Free Project Management Tool for Windows, macos and linux*, *GanttProject*. Available at: https://www.ganttproject.biz/.

Google (n.d.) *Teach google assistant to recognize your voice with voice match - android*, *Google Assistant Help*. Google. Available at: https://support.google.com/assistant/answer/9071681#vm_pr&zippy=%2Cvoice-match-personal-results.

Git (n.d.) *Git*. Available at: https://git-scm.com/.

GitHub (n.d.) *GitHub*. Available at: https://github.com/.

Guide to the UK General Data Protection Regulation (UK GDPR) (n.d.) *ICO*. Available at: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/.

Henschel, A., Laban, G. and Cross, E.S. (2021) "What makes a robot social? A review of social robots from science fiction to a home or hospital near you," *Current Robotics Reports*, 2(1), pp. 9–19. Available at: https://doi.org/10.1007/s43154-020-00035-0.

HSBC (n.d.) *Telephone banking: Give our friendly team A call - HSBC UK, Telephone Banking | Give Our Friendly Team a Call - HSBC UK*. HSBC. Available at: https://www.hsbc.co.uk/ways-to-bank/phone-banking/.

Juang, B.-H. and Rabiner, L.R. (2006) "Speech recognition, automatic: History," *Encyclopedia of Language & Linguistics*, pp. 806–819. Available at: https://doi.org/10.1016/b0-08-044854-2/00906-8.

Kintone (2019) What is Agile?. Kintone. Available at: https://blog.kintone.com/business-with-heart/what-is-agile-development

Koç, H. *et al.* (2021) "UML diagrams in Software Engineering Research: A systematic literature review," *The 7th International Management Information Systems Conference* [Preprint]. Available at: https://doi.org/10.3390/proceedings2021074013.

Lewicki, P., Tochowicz, J. and van Genuchten, J. (2019) "Are robots taking our jobs? A RoboPlatform at a bank," *IEEE Software*, 36(3), pp. 101–104. Available at: https://doi.org/10.1109/ms.2019.2897337.

Microsoft (n.d.) *Use voice recognition in Windows*, *Microsoft Support*. Microsoft. Available at: https://support.microsoft.com/en-gb/windows/use-voice-recognition-in-windows-83ff75bd-63eb-0b6c-18d4-6fae94050571.

Microsoft (n.d.) *Windows Speech Recognition commands*, *Microsoft Support*. Microsoft. Available at: https://support.microsoft.com/en-gb/windows/windows-speech-recognition-commands-9d25ef36-994d-f367-a81a-a326160128c7#WindowsVersion=Windows_11.

Miller, J., Williams, A.B. and Perouli, D. (2018) "A case study on the Cybersecurity of Social Robots," *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction* [Preprint]. Available at: https://doi.org/10.1145/3173386.3177078.

Mulliner, C. *et al.* (2013) "SMS-based one-time passwords: Attacks and Defense," *Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 150–159. Available at: https://doi.org/10.1007/978-3-642-39235-1_9.

Oracle (n.d.) *What is a database?*, *Oracle*. Available at: https://www.oracle.com/database/what-is-database/.

QTrobot, an engaging educational robot for children with autism and special needs education (2022) *LuxAI S.A.* LuxAI. Available at: https://luxai.com/robot-for-teaching-children-with-autism-at-home/.

QTrobot documentation (n.d.) *QTrobot Documentation*. Available at: https://docs.luxai.com/.

PgAdmin (n.d.) *PgAdmin*, *pgAdmin*. Available at: https://www.pgadmin.org/.

Plösch, R. *et al.* (2016) 'Measuring, assessing and improving softwarequality based on object-oriented designprinciples', *Open Computer Science*, 6(1), pp. 187–207. doi:10.1515/comp-2016-0016.

PostgreSQL (n.d.) *PostgreSQL: The World's Most Advanced Open Source Relational Database*, *PostgreSQL*. Available at: https://www.postgresql.org/.

Privacy-preserving Robotics in DEntistry (PRiDE) (n.d.) *Privacy-preserving Robotics in DEntistry*. University of Plymouth. Available at: https://www.plymouth.ac.uk/research/privacy-preserving-robotics-in-dentistry.

Puzhevich, V. (2021) *Functional vs Non-Functional Requirements: Key Differences*, *SCAND*. Available at: https://scand.com/company/blog/functional-vs-non-functional-requirements/.

Python (n.d.) *Welcome to Python.org*, *Python.org*. Available at: https://www.python.org/.

Rehkopf, M. (n.d.) *What is a Kanban Board?*, *Atlassian*. Available at: https://www.atlassian.com/agile/kanban/boards.

Robot operating system (n.d.) *ROS*. Available at: https://www.ros.org/.

Rolland, C. and Achour, C.B. (1998) "Guiding the construction of textual use case specifications," *Data & Knowledge Engineering*, 25(1-2), pp. 125–160. Available at: https://doi.org/10.1016/s0169-023x(97)86223-4.

Shaughnessy, D. (2004) "Improving speech analysis methods for robust automatic recognition," *Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No.04CH37513)* [Preprint]. Available at: https://doi.org/10.1109/ccece.2004.1344981.

Srinath, K.R., 2017. Python–the fastest growing programming language. *International Research Journal of Engineering and Technology*, *4*(12), pp.354-357.

The principles (n.d.) *ICO*. Available at: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/principles/.

Trello for product management teams (n.d.) *Trello*. Available at: https://trello.com/teams/product.

What are user stories? (n.d.) *Agile Alliance |*. Available at: https://www.agilealliance.org/glossary/user-stories/.

What is Scrum? (n.d.) *Home | Scrum Guides*. Available at: https://scrumguides.org/.

Whitney, E. (2021) *Introduction to gathering requirements and creating use cases*, *Gathering Requirements and Creating Use Cases, An Introduction*. Available at:

https://www.codemag.com/Article/0102061/Introduction-to-Gathering-Requirements-and-Creating-Use-Cases.

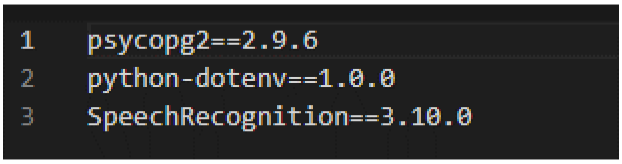
Yaacoub, J.-P.A. *et al.* (2021) "Robotics cyber security: Vulnerabilities, attacks, countermeasures, and recommendations," *International Journal of Information Security*, 21(1), pp. 115–158. Available at: https://doi.org/10.1007/s10207-021-00545-8.

9 Appendix

9.1 User Guide

Hardware needed: QTrobot

Software Requirements:



```
psycopg2==2.9.6
python-dotenv==1.0.0
SpeechRecognition==3.10.0
```

Figure 35 requirements.txt

Creating the Package and installing dependencies

These commands allows you to create the package

```
cd ~/catkin_ws/src
catkin_create_pkg AAVR rospy roscpp -D "AAVR"
```

Within AAVR, copy all the files from the repository to the newly created folder.

Within the repository folder in catkin_ws, run the requirements command with

```
pip install -r requirements.txt
```

Creating .env file

Command to create a blank .env file

```
touch .env
```

Then, edit the file with this command

```
nano .env
```

Template below for .env

```
EMAILSEND="Write which email you would like to send from here"
EMAILPASS="Write the email pass here, needs to be app password"
COMPOSEUSER=root
```

Building and running project

Run this to build the project

```
catkin_make
```

Then, from the AAVR folder run

```
chmod +x /src/main.py
```

This is to give write access to the program.

```
rosrun AAVR main.py
```

This command then allows the program to run.

9.2 Licensing for external libraries

Python-dotenv = https://github.com/theskumar/python-dotenv

Rospy = http://wiki.ros.org/rospy

speech_recognition = https://github.com/Uberi/speech_recognition

9.3 Meeting Minutes

Date & Time: 17/10/22 | 4.10PM – 4.40PM

Location: A327 PSQ Building
Participants: Gregory Kua, Xinyao Huang, Noah Keedle-Isack, Alvaro Resende, Hai-Van Dang

Introduction

- Name & Course names, Quick introduction
- The students have the call to do their own projects. The supervisor is there to advice if there are any risks / problems that might arise.
- You need to think about your careers now. Check out the careers / opportunity available.

Main

- Shared folders for the documents, Version Control shared, Planners,
- Quick rundown on the assessment brief
- Writing a main outline for the final report
- Break-down the main tasks into subtasks
- Define a clear output, User Journey Maps implementation (Show clear steps how to use your project)
- MVP needs to be done by Christmas (If not there is a chance you might fail)
- Reminder on the deadlines
- Objective, Output, MVP, Features + Plans for Sprint 1 is written on a paper

---

Date & Time: 28/10/22 | 11.15AM – 12.59PM

Location: Room 205 Library
Participants: Gregory Kua, Nathan Everett, Xinyao Huang, Alvaro Resende, Michael McDonald

- Noah could not attend the meeting due to an illness

Project Discussions

- Quick introduction on what each student's topic are for their Final Year Project
- Hosting Ideas for the project
- Sprint Planning Ideas
- Discussed frameworks and languages
- Affordable Hosting Sites
- Shared Discord contacts

Date & Time: 4/11/22 | 11.15AM – 12.00PM

Location: Smeaton 101
Participants: Gregory Kua, Nathan Everett, Xinyao Huang, Noah Keedle-Isack, Alvaro Resende

Intro

- Discussion by supervisor
- Rounds on what each student have done for their sprints, Suggestions and advice have been given to each student accordingly

Supervisor Notes

- Write the documentation about the Microphone in the Project Portfolio
- LuxAI structure on how the robot works can be taken but needs to be documented in the references.
- Ask Thomas about the ethical approval for User testing

Date & Time: 11/11/22 | 11.15AM – 12.00PM

Location: Room 112 Library
Participants: Gregory Kua, Noah Keedle-Isack, Alvaro Resende, Michael McDonald, Xinyao Huang

- Nathan could not attend the meeting due to an illness.

- Quick update on our final year project
    - Chatting about frameworks
        - Aurelia
    - Sprint Review
        - **Sam**
        - **Michael**
        - **Gregory**
            - Followed up on feedback since last week,
            - Added more headings to final report
            - Updated Project Plan with numbered
        - **Noah -**
            - Added missing headings to project report template.
            - Updated project plan and project initiation based on feedback from DLE, and from previous meeting.
            - Created Gantt chart and added it to project initiation.
        - **Xinyao -**
            - Modify the project template content.
            - Join the school's software engineering discord.

Date & Time: 18/11/22 | 11.15AM – 12.00PM

Location: A327 PSQ Building
Participants: Gregory Kua, Nathan Everett, Noah Keedle-Isack, Alvaro Resende, Michael McDonald, Xinyao Huang

Sprint Planning

- Checking Sprint Planning for every students

Main

- Kind reminder to write everything in the portfolio

Greg

What was done in sprint 2?

1. Resolve issue with robot

What to do in sprint 3?

1. The robot needs to save the audio
2. Write infrastructure of the robot
3. Design of the protocol/modules
4. Planning: Meet me on Monday,2-3pm for planning, data research

Michael

What was done in sprint 2?

1. Backend development: basic functionalities (show the names of gamers, actual data) using public API from games, show the data in web)

What to do in sprint 3?

1. Frontend development
2. Whole picture: API prototypes, front end design, list of features
3. Project vision and market research, list of features

Meet me on Monday,2-3pm for planning

Nathan

What was done in sprint 2?

1. Gateway, implementing the services
2. Uploading, downloading data to gcloud storage

What to do in sprint 3?

1. Draw the interaction picture into the report
2. Write about the flexity to switch to a cloud into the report
3. Combining compression with uploading/downloading
4. Paper to compress using machine learning: select the best one and integrate it into the system

Noah

What was done in sprint 2?

1. Web socket server set up
2. Test
3. Half way to authentication
4. Finalise the plan

What to do in sprint 3?

1. Write the report: infastructure, put the project initiation into the report0
2. Write about technology
3. Write about authentication system with microsoft identity into the report
4. Design the front end

Sam

What was done in sprint 2?

1. API for log in, listing plants, adding plants

What to do in sprint 3?

1. UI implementation
2. Set up landing page
3. Add more details of plants
4. Feature to add more plants of own user
5. Report: 1 section about security consideration/ sign up
6. Write section about the used devices/sensors

Xinyao

What was done in sprint 2?

1. Demo for log in is not working

What to do in sprint 3?

1. Organise the backlog into sprints
2. Make log in page running
3. More details about the plan
4. Design of the websites
5. Design the database schema
6. Implement the page to show the exam to the students

Date & Time: 25/11/22 | 11.15AM – 12.00PM

Location: Room 216 Library Building
Participants: Gregory Kua, Nathan Everett, Xinyao Huang, Alvaro Resende,

Gregory -

- Showed Project designs via phases, UML diagrams
- Explained how the enrolment and validation phases work

Xinyao –

- Showed database ER diagram

Date & Time: 02/12/22 | 11.15AM – 12.00PM

Location: A327 PSQ Building
Participants: Gregory Kua, Nathan Everett, Xinyao Huang, Noah Keedle-Isack, Alvaro Resende,

Gregory

- What was done in sprint 3?
  - Writing the infrastructure for the robot
  - Readjusted Sprint Dates
  - Program base Designs
  - Filled in Trello
  - Meeting with Dr Hooman, Read through notes that was given
  - Chose where to store the audio recordings
    - AWS
      - 2000 Puts
      - 20000 Gets
      - 5GB Storage
- What to do in sprint 3?
  - Showing MVP by then,

Sam:

What was done in sprint 3?

- Landing page implementation refined (probably complete)

What to do in sprint 4?

- UI Implementation for plant details
- Signup page (endpoint completed)
- Deleting/Updating plants

  - Endpoint for collecting metrics (POSTable by any device)

Issues:

Nathan
What was done in sprint 3?

- fixing bugs

What to do in sprint 4?

- Add security to gateway (authorisation, access control)
- Frontend to upload/download

- Draw the interaction picture into the report
- Write about the flexity to switch to a cloud into the report
- Combining compression with uploading/downloading
- Paper to compress using machine learning: select the best one and integrate it into the system

Issues: no

Noah:
What was done in spritn 3?

- Design for mobile ui in report
- Technology section in report
- Found dataset for AI

Issue: may need to use matlab online for training online
Next sprint?

- Finish mobile front end
- Detection of text messages

Melissa
What has been done

- AI algo to predict possible trains using data from gwr

Next sprint

- Implement UI
- Integrate AI backend with frontend in mobile

Issue: no

Xinyao
What has been done

- Finished the database and drew an ER diagram
- Basically done with the login and registration screen

Next sprint

- Finding server Issues

Issue:

- Servers sometimes fail to start

Date & Time: 09/12/22 | 11.15AM – 12.00PM

Location: Room 205 Library Building
Participants: Gregory Kua, Nathan Everett, Xinyao Huang, Alvaro Resende, Michael McDonald

- Gregory
  - Shows audio recordings from the robot
  - Explaining the debugging problems that I've been having
- Xinyao
  - Readjusted Sprint content
- Sam
  - Added locations, refactored app to use locations
  - Lots of UI work, particularly the signup page is now functional
  - Need ability to update/remove plants and locations
  - Need to create the routes for POSTing data to locations and plants

---

Date & Time: 16/12/22 | 11.15AM – 12.00PM

Location: A327 PSQ Building
Participants: Gregory Kua, Nathan Everett, Noah Keedle-Isack, Alvaro Resende, Xinyao Huang, Melissa, Jasper

Gregory
What has been done in Sprint 4

- Robot is finally able to save recordings of voice (From Sprint 2)
- Robot is able to understand Human Voices and ignores background noise (From Sprint 2)
  - Processing in the NUC automatically
- Cloud API Research on how to use Boto3 to implement S3 storage uploading
- Stored Data file name will be randomised
  - To avoid the same file name be uploaded to the cloud that will cause conflictions
- Robot is now able to record the voice within a certain time, (10 seconds at the moment)
  - This is to allow the user to know when they'll be recorded and allow for time for user to "prepare"
- Project Portfolio
  - Reference List updated
  - Market Research has been done
    - HSBC Bank UK
    - Google Assistant
    - Windows 10 / 11
- ~~Audio Data is Storing Correctly and Safely~~
- ~~File Uploading on AWS S3~~

What to do in Sprint 5 (Starting 9th January)

- Research into streaming services on how they process their audio

Issues: Code for Uploader is correct but currently there are authentication issue with the credentials.

Noah
What has been done in Sprint 4

- Got mobile project compiling on Android (iOS remains to be looked at)
- Messages can now be sent and received via the app
- Formatted abusive speech dataset for use in training
- Trained first version of Neural Network and exported network as DLL. This will then be loaded into the API next sprint.

What to do in Sprint 5

- Refine Mobile UI
- Add authentication to mobile app
- Pass messages through Neural Network to ensure they are safe to send

Issues:

- Way too many other assignments were due during this sprint... ☹

Sam

Completed in Sprint 4:

- Angular frontend containerised (Docker) on samoboolean/ng-plantmo
- Node TS Express backend containerised (Docker) on samoboolean/plantmo-server
- Lots of code quality improvements, moved common code to npm package (plantmo-common)
- Refactored code base to allow for locations (locations store collections of plants)
- Metrics collecting is available for locations and plants. However not viewing/displaying it yet
- Wired up the raspberry pi to the sensors with the analogue to digital converter and got data collecting (soil moisture) in a percentage

To do in sprint 5:

- Pick charting library
- Write python on the Pi to collect and send metrics to the server

Issues:

- Containerisation + constant build issues, but it is all stable now. Sometimes the docker build takes like 20 minutes on my x64 PC

Demonstration:

Greg demonstrated with recording of robot: robot can record but not upload to cloud yet
Noah showed the code of model training using neural network, and messaging application with simple GUI
Xinyao showed webpages for user registration, login, exam with multiple choice questions (single answer only)
Melissa showed the mobile app with GUI for choosing train route. It would show train, time, destination.
Melissa also presented the code for training model to learn about the available seats on the train.
Luke demonstrated a mobile app which can show the list of manga with basic information.
Jasper demonstrated a remote server sending and running a script on a windows machine.

Date & Time: 20/01/23 | 11.15AM – 12.00PM

Location: A327 PSQ Building
Participants: Gregory Kua, Nathan Everett, Xinyao Huang, Noah Keedle-Isack, Alvaro Resende, Michael McDonald

Gregory:

Completed in Sprint 5

- Due to assignments Not much was done due to other assignments that needs to be done

To do in Sprint 6

- Sprint 5's Tasks and some of Sprint 6's Task
- Includes working on reports
- Project Testing and ethical approval

Sam:

Sprint 5: completed suggestions based on last meeting. Charts implemented but not fetching from live data. More recently, worked on the report. Did not complete the requirements of the sprint because of other deadlines
Sprint 6: Continue working on report, crud implementations and security considerations implemented (sql injection protection etc). Start to plan the user experiments

Date & Time: 27/01/23 | 11.00AM – 1.00PM

Location: Library Room 108
Participants: Gregory Kua, Noah Keedle-Isack, Alvaro Resende

Sam

- Investigating and experimenting with charts for showing location temperature history. Data is being collected for locations (from Pi Pico)
- Working on updating and deleting locations and plants
- Dark mode interface
- Need to expand more sections in the report, and discuss how the Pi's have been wired up

Noah

- Continued filling out report for work done for MVP.
- Investigating ways to improve AI detector, and looking at switching to a pattern recognition implementation.
- Next Week: UI Overhaul + Store previous messages in Database.

Gregory

Documentation Updates

- Reference List is now done properly and has been cited
- Meeting Minutes document for the Project portfolio has been rewritten to accommodate the portfolio
- Sprints has been rewritten as well

By Next Meeting

- Testing Methodologies will be done
- Ethical Forms have been checked

Date & Time: 03/02/23 | 1.15PM – 2.00PM

Location: A327 PSQ Building
Participants: Gregory Kua, Nathan Everett, Xinyao Huang, Alvaro Resende

Xinyao:

Completed admin managemnent
Next: fix the bug, start working on documentation, continue with implementation

Gregory:

Completed in Sprint 6

- Project Testing Draft is done on how it'll work
- Ethical Approval Forms have been checked and can be used

To do in Sprint 7

- Uploaded File should be processed via aws rekognition
- Poster template for the submission + Description will be started

Sam:

Completed in Sprint 6:

- SQL query sanitization changes (made API injection proof)
- Location CRUD changes (updating now works, still no DELETE as it required extra contrainst on the tables and model/schema changes
- Fully implemented dark theme throughout

Next sprint:

- Viewing plants properly (and their moisture info) + Updating/Deleting
- Investigation into the email (smtp) settings for sending notifications
- Implement toasts into the app
- Have a platform for integration tests

Date & Time: 10/02/23 | 11.15AM – 12.00PM
Location: Library Room 216
Participants: Gregory Kua, Xinyao Huang, Alvaro Resende

- **Xinyao -**
  - Improve the document
- **Gregory -**
  - Documentation is 20% Completed (2K Words)
  - Updates on an alternation token based authentication for AWS Is in the works
    - Contacted Nathan for help
  - Gantt Chart Section has been updated as well to take into account of changes
  - Sprint 6 needs to be rewritten due to some health issue that has happen
- **Sam -**
  - Added constraints to some tables to allow for cascading when entities are deleted (Locations)
  - Locations can be deleted, carries across to temp + humidity data
  - Currently halfway through this sprint, I still need to expand the same functionality into the plants (editing, deleting and cascading)

Date & Time: 17/02/23 | 11.15AM – 12.00PM
Location: A327 PSQ Building
Participants: Gregory Kua, Xinyao Huang, Noah Keedle-Isack, Alvaro Resende

- **Gregory:**
  - Completed in Sprint 7
    - Documentation
      - Project Development Section has been updated
        - Early Stages still
        - Tools and software's updated in Dev Section
      - Other sections has been revised to reduce word count
        - Eg, Using hyperlinks
    - Development
      - Have been busy working with AWS Auth, Rewrote some code
        - Still not working properly (Might need to look at alternative)
      - Tested AWS Rekognition with their demo file
        - Works well for video and image recognition
          - Audio recognition needs some tweaking but it is possible
        - Does not work directly with the audio files from the QTrobot
          - (Which is concerning)
  - To do in Sprint 8

- Documentation
  - Poster + Desc has not been started, will be started ASAP
- Development
  - Continue to try to fix the aws issue
- Continue to work on the voice recognition with recognition

- Challenges:
  - There is no existing implementation for recognize which person based on voice

- **Noah:**
  - Completed in Sprint 7
    - Report
      - Introduction and Background Sections
      - Found references for Background section
    - Development
      - Added authentication to SignalR (WebSockets)
      - Still not working 100% of the time.
      - Currently working on adding authentication to mobile app – looking at secure ways to store credentials on devices
      - Created functionality for group chats. Changed how the database was modelled.
  - To do in Sprint 8
    - Report
      - Add UML and UI Designs
      - Write more on technologies used and reasoning
    - Development
      - CI/CD Setup with Microsoft Azure
      - App UI Overhaul
      - Export Trained Neural Network from MATLAB as a DLL and link to project – add to Chat pipeline

- **Sam:**
  - Sprint 7: plants showing moisture information (finally have a viewer for this). Location temperature shown on plants in dashboard
  - Sprint 8: need to setup the raspberry pi to post data so I have live data for testing. Still need to add editing/deleting functionality for plants. Also need to fix a few inconsistencies with adding plants (related to the choose from template functionality) The testing environment also needs to be set up so that I know when something becomes a breaking change (for the API)

- **Xinyao:**
  - Completed in Sprint 7
    - Documentation
    - Teacher's functionalities
  - To do in Sprint 8
    - Plan for testing: unit test/ functionality test/ usability test
    - Work on the documentation

Date & Time: 24/02/23 | 11.00AM – 1.00PM

Location: Library Room 111

Participants: Gregory Kua, Nathan Everett, Alvaro Resende, Xinyao Huang,

- **Xinyao:**
  - Improve the document

- **Gregory:**
  - Things that has been done this week
    - Research into documentation
    - Redid the trello board
    - AWS Auth does work sometimes after checking, still has not found what was fixed
  - Things to be done next week
    - Research more into facial recognition and it's possibilities

- **Sam:**
  - Both of the raspberry pi's are communicating with the server and sending their data. The pico is sending location temperature to the server, the pi 4 is collecting and sending the soil moisture to the server.
  - The web app is showing the soil moisture and location temperature in a graph with a timeline (to allow for data zooming)
  - Still haven't added the ability to edit/delete plants yet

---

Date & Time: 03/03/23 | 11.00AM – 1.00PM

Location: A327 PSQ Building
Participants: Gregory Kua, Xinyao Huang, Noah Keedle-Isack, Alvaro Resende

Gregory:
Completed in Sprint 8

- Poster confirmation on what needs to be written on it
- New Program Design has been made
- OTP Functionality
- Email Upload functionality started

To do in Sprint 9

- Documentation Update, Proj Vision, Proj Designs,
- Sprint 7 & 8's updates
- Poster Template started
- Authentication phase for the user after OTP

Sam:
Completed in Sprint 8:

- Hosting the stack on the internet now
- Pis have collected enough data for testing, got a few thousand rows in the database.

To do in sprint 8:

- Still need to fix inconsistencies with the plant add/edit interface, and start on testing functionality
- Migrate some plant data to the hosted version
- Look into an email container to use as a service for sending emails

Noah:
Completed:

- Finished “chat” UI and started on Friends List UI
- Tried adding authentication to SignalR (WebSockets). Having issues with setting up CORS for requests coming from Android Emulator. To negate this have been testing on a physical device but this is eating up a lot of time.
- Not much was done this sprint as focussing on other assessments.

To Do in Sprint 9

- Solve CORS issue and start on CI/CD.
- Finally build MATLAB Neural Network DLL and integrate with API (focused)
- Polish up UI
- Continue with the report: dataset and evaluation criteria

Xinyao:

Completed in Sprint 8

- Documentation: survey and results
- Testing: functionality test (incomplete)

To do in Sprint 9

- Try to solve the problem
- Work on the documentation

Date & Time: 10/03/23 | 11.00AM – 1.00PM

Location: Library Floor 1

Participants: Gregory Kua, Alvaro Resende

Sam

- Planning the user studies to begin asking questions about the interface. Need to narrrow it down to a specific group of people (technical users) to get feedback relevant to the project
- Hosted site works well and is stable. Plant has been posting data to the public site just fine
- Still haven’t added plant editing/deleting (coming soon)
- Need to add a section to my report about this user study and my plan of action for it. How it will play through and the expected feedback (goals) of the study

Gregory

- Things that have been done this week
    - Docker Crash course
    - Database Server has been completed
    - Mail Server is up, but has not yet been used
- Things to be done next week
    - OTP via mail?
    - Robot is uploading correct information to the db server
    - Leftover documentation, target 3K

Date & Time: 17/03/23 | 1.15PM – 2.00PM
Location: A327 PSQ Building
Participants: Gregory Kua, Nathan Everett, Xinyao Huang, Noah Keedle-Isack, Alvaro Resende, Michael McDonald

Gregory:
Completed in Sprint 9

- Database Tables for storing User Details
- Learnt Docker, Using it in test environment
- Sprint 7 & 8's Documentation, Project Designs,

To do in Sprint 10

- Documentation Sprint over Easter

Sam:
Sprint 9:

- User studies planning/investigation

To do in Sprint 10:

- Still need to look into email container + look into testing the frontend

Xinyao:
Completed in Sprint 9

- Documentation: Requirements

To do in Sprint 10

- Work on the documentation

Noah:
Completed in Sprint 9:

- Compiled Neural Network to DLL and linked with API Project
- Messages are now checked for offensive language, and blocked from being viewed by the other connected client.
- Continued with poster, due at the end of the month.

To Do in Sprint 10:

- Further report work
- Documentation
- Tidy up UI and bug test

Nathan:

- Front end: re-design, hosting on cloud
- Log in, getting data
- Using cloud provider for data storage
- Features implemented: upload, download, view files, compress files

---

Date & Time: 24/03/23 | 11.00AM – 12.00PM
Location: Library Room 107
Participants: Gregory Kua, Alvaro Resende

Gregory

- Things that have been done this week
    - Rewrote meeting minutes in the portfolio
    - Updated Documentations

(Not much was done within this week on the side of development)

- Things to be done next week
    - Have a second demonstration
    - Video demonstrations and Slides
    - Poster and Description

Sam

- Previous Week
    - Redesigned some parts of the dashboard page
    - Added weather tracking functionality (based on users location)
    - Added plant searching from external API, and local caching of results for performance
- Next Week
    - Work on the poster

Date & Time: 31/03/23 | 1.00PM – 2.00PM
Location: A327 PSQ Building
Participants: Gregory Kua, Xinyao Huang, Alvaro Resende,

Sam:

Completed in Sprint 10:

- Redesigned some UI parts to make them more clear
- Added emailing. A self hosted user can provide SMTP details and the server will use it to send emails from their own email address.
- Added an option to verify new users (send registration codes)
- Added a weather forecasting section to the dashboard, users can share their location to the server and provided they have an API key, it will fetch weather information for their location. Users with no API key can just skip this
- Added plant searching – also with an API key, users can enable extra functionality to search for their plant species for information to use on soil moisture and care notes.

To do in Sprint 11:

- I still haven't added plant editing!
- Start adding some tests to the backend to test functionality upon every change

Gregory:

Completed in Sprint 10

- Email sending in python using SMTPLIB (Will be using dotenv for secrets)
- Have not implemented checks for the OTP
- Posters and Thumbnails

To do Sprint 11

- Be done with Software
- Be done with Portfolio
- Be done with Video

Xinyao:

Completed in Sprint 10

- Documentation: Development Stage & System Design
- Poster: first edition

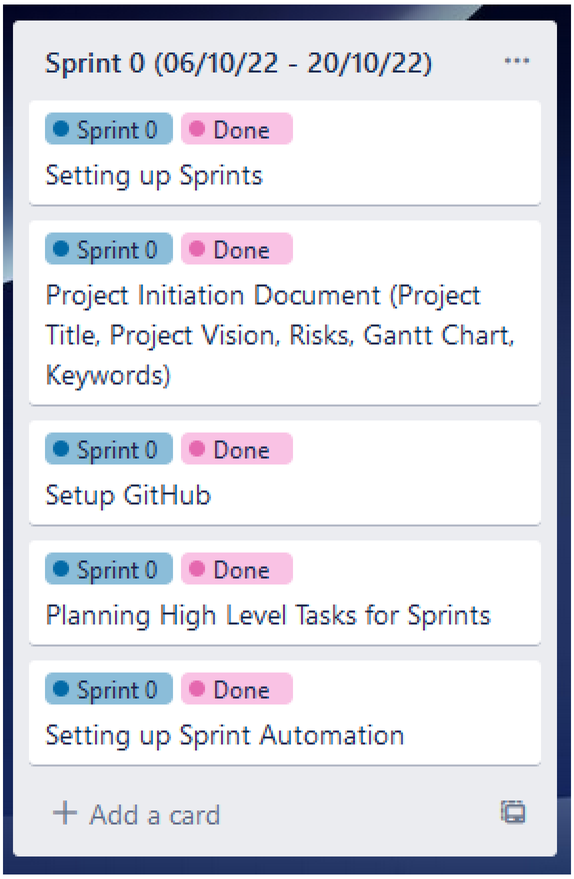
To do in Sprint 11

- Work on the documentation

- Try to optimise the UI for generating questions
- Redesigned poster

9.4 Sprint Reviews

9.4.1 Sprint 0



Sprint 0 (06/10/22 - 20/10/22)

Sprint 0 | Done
Setting up Sprints

Sprint 0 | Done
Project Initiation Document (Project Title, Project Vision, Risks, Gantt Chart, Keywords)

Sprint 0 | Done
Setup GitHub

Sprint 0 | Done
Planning High Level Tasks for Sprints

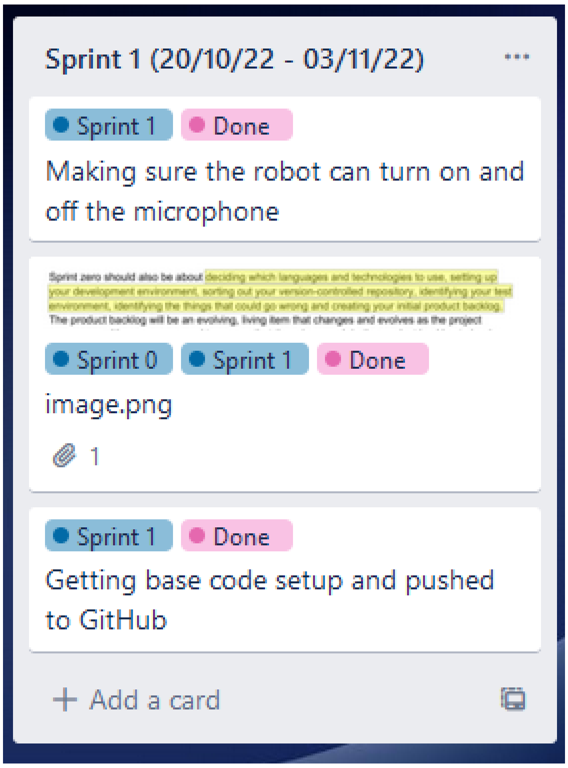
Sprint 0 | Done
Setting up Sprint Automation

Figure 36 Sprint 0 Board

What went well:	What did not go well
Early setup to start the project. Everything went relatively smoothly	n/a

Table 5 Sprint 0 Table

9.4.2 Sprint 1



Sprint 1 (20/10/22 - 03/11/22)

Sprint 1 | Done

Making sure the robot can turn on and off the microphone

Sprint zero should also be about deciding which languages and technologies to use, setting up your development environment, sorting out your version-controlled repository, identifying your test environment, identifying the things that could go wrong and creating your initial product backlog. The product backlog will be an evolving, living item that changes and evolves as the project

Sprint 0 | Sprint 1 | Done

image.png

1

Sprint 1 | Done

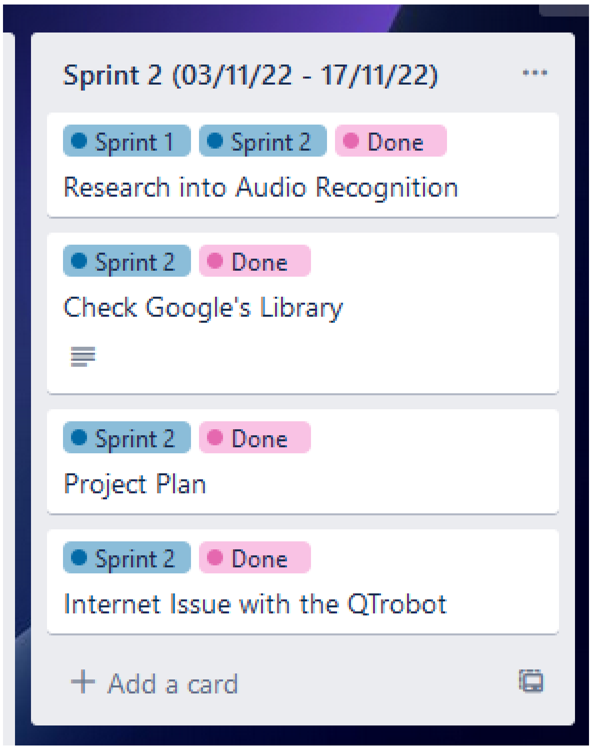
Getting base code setup and pushed to GitHub

Figure 37 Sprint 1 Board

What went well:	**What did not go well**
Base code has started. Microphone was working within the QTrobot	Internet issues plagued the QTrobot at this point of time. Having to use hotspot to work on the robot

Table 6 Sprint 1 Table

9.4.3 Sprint 2



Sprint 2 (03/11/22 - 17/11/22)

Sprint 1 Sprint 2 Done
Research into Audio Recognition

Sprint 2 Done
Check Google's Library

Sprint 2 Done
Project Plan

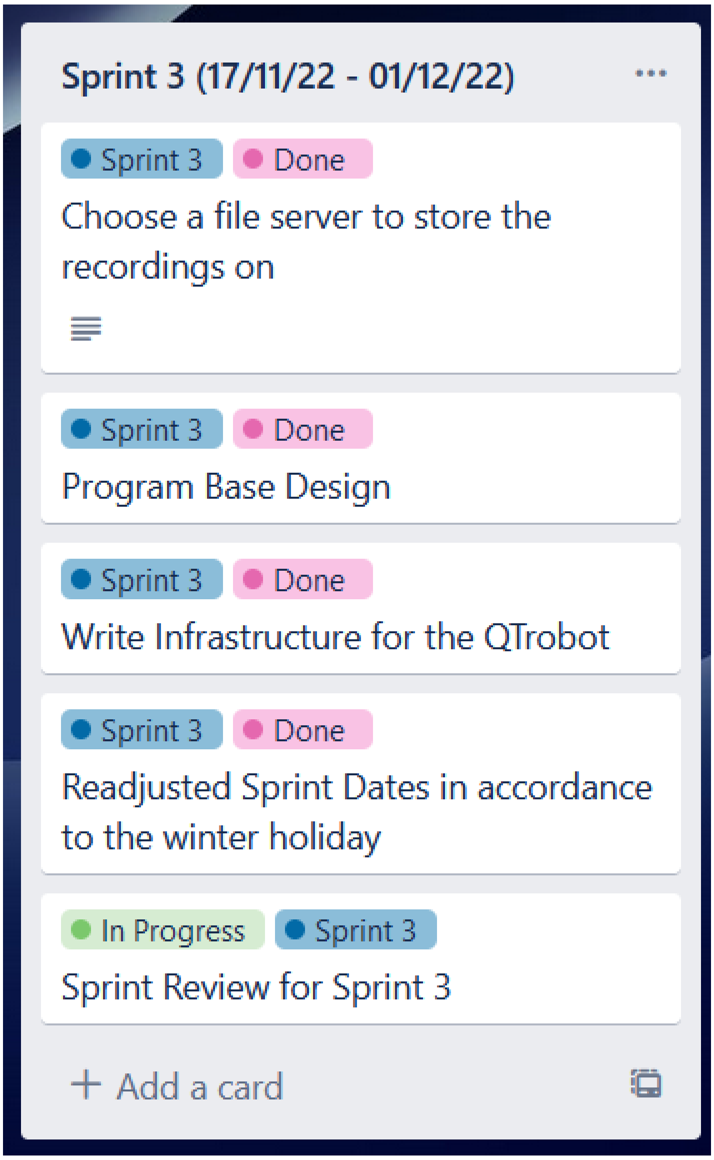
Sprint 2 Done
Internet Issue with the QTrobot

Figure 38 Sprint 2 Board

What went well:	**What did not go well**
Internet issue was fixed finally. Research week	n/a

Table 7 Sprint 2 Table

9.4.4 Sprint 3



Sprint 3 (17/11/22 - 01/12/22)

Sprint 3 Done
Choose a file server to store the recordings on

Sprint 3 Done
Program Base Design

Sprint 3 Done
Write Infrastructure for the QTrobot

Sprint 3 Done
Readjusted Sprint Dates in accordance to the winter holiday

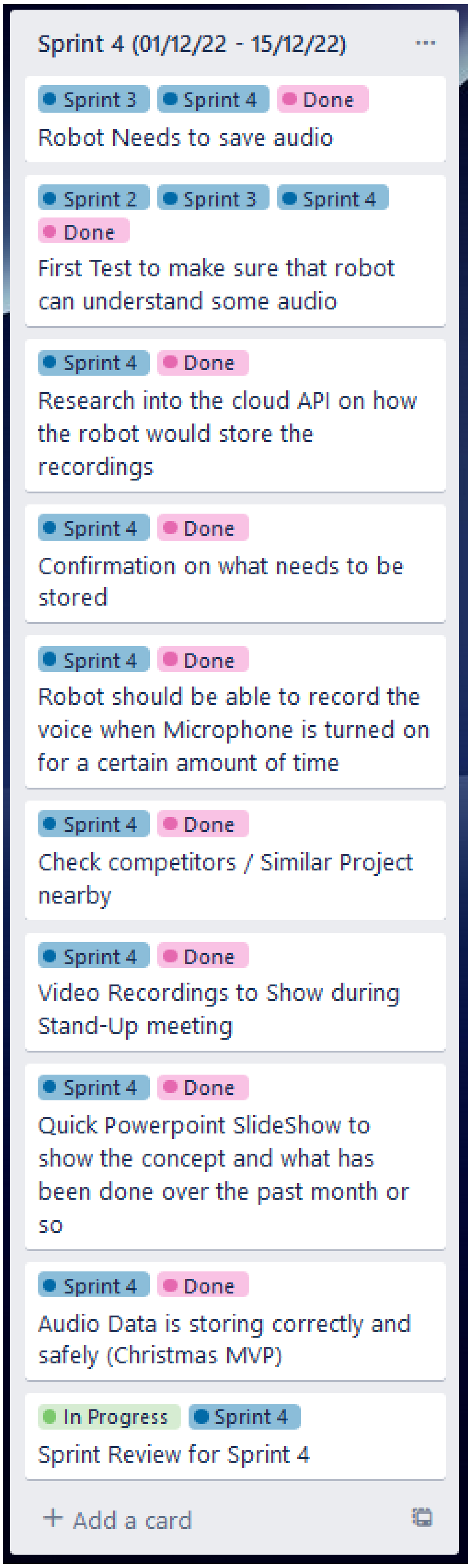
In Progress Sprint 3
Sprint Review for Sprint 3

Figure 39 Sprint 3 Board

What went well:	**What did not go well**
Base designs were done, Documentation was going well	File server took a while to choose. Trello board had to be readjusted cause dates did not take into account vacation days.

Table 8 Sprint 3 Table

9.4.5 Sprint 4



Sprint 4 (01/12/22 - 15/12/22)

Sprint 3 Sprint 4 Done
Robot Needs to save audio

Sprint 2 Sprint 3 Sprint 4 Done
First Test to make sure that robot can understand some audio

Sprint 4 Done
Research into the cloud API on how the robot would store the recordings

Sprint 4 Done
Confirmation on what needs to be stored

Sprint 4 Done
Robot should be able to record the voice when Microphone is turned on for a certain amount of time

Sprint 4 Done
Check competitors / Similar Project nearby

Sprint 4 Done
Video Recordings to Show during Stand-Up meeting

Sprint 4 Done
Quick Powerpoint SlideShow to show the concept and what has been done over the past month or so

Sprint 4 Done
Audio Data is storing correctly and safely (Christmas MVP)

In Progress Sprint 4
Sprint Review for Sprint 4

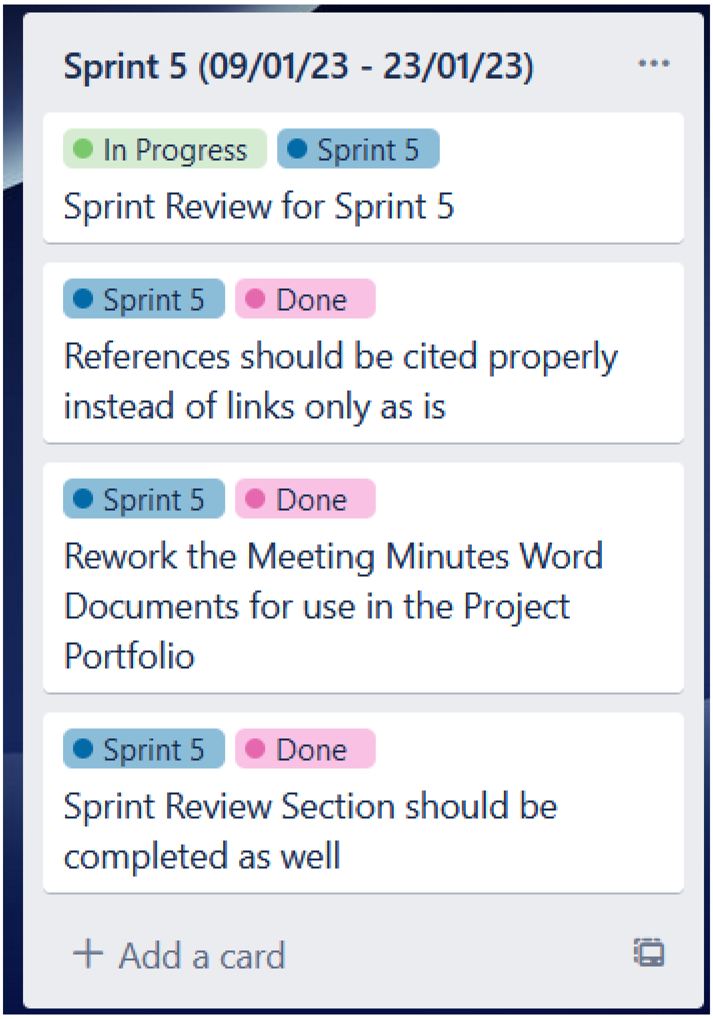
Figure 40 Sprint 4 Board

What went well:	What did not go well

Robot at this point of time is able to save audio. MVP was met.	Linux GUI on the QTrobot started to bug out, Had to contact LuxAI to get it fix. Took a week. The GUI made it unusable to do anything with the robot.

Table 9 Sprint 4 Table

9.4.6 Sprint 5



Sprint 5 (09/01/23 - 23/01/23)

In Progress, Sprint 5
Sprint Review for Sprint 5

Sprint 5, Done
References should be cited properly instead of links only as is

Sprint 5, Done
Rework the Meeting Minutes Word Documents for use in the Project Portfolio

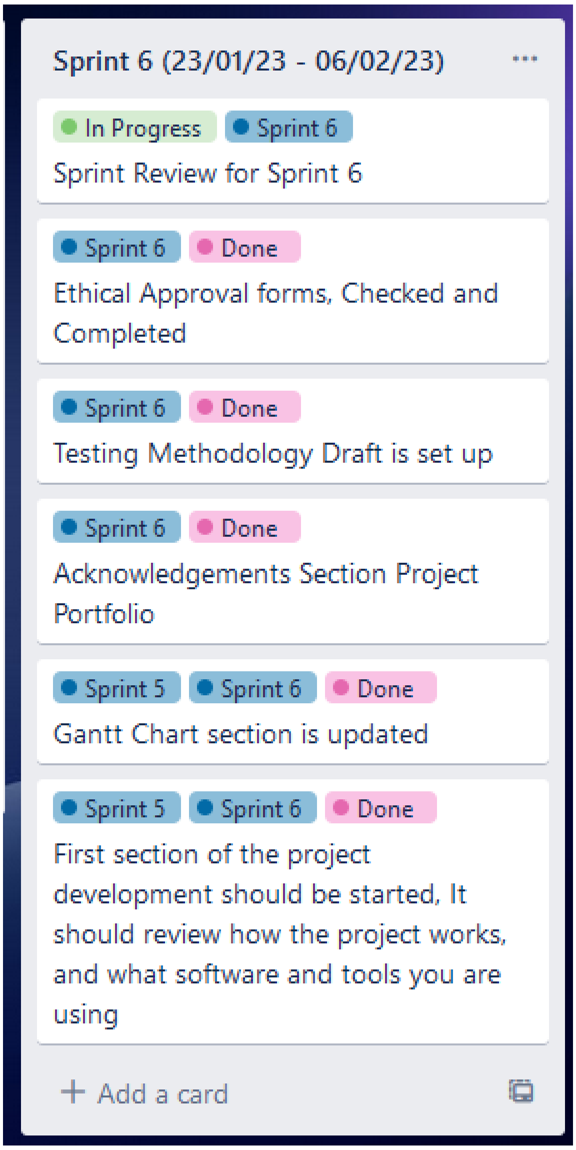
Sprint 5, Done
Sprint Review Section should be completed as well

Figure 41 Sprint 5 Board

What went well:	**What did not go well**
More documentation update.	Time management with other assignments, this sprint has taken a toll.

Table 10 Sprint 5 Table

9.4.7 Sprint 6



Sprint 6 (23/01/23 - 06/02/23)

In Progress Sprint 6
Sprint Review for Sprint 6

Sprint 6 Done
Ethical Approval forms, Checked and Completed

Sprint 6 Done
Testing Methodology Draft is set up

Sprint 6 Done
Acknowledgements Section Project Portfolio

Sprint 5 Sprint 6 Done
Gantt Chart section is updated

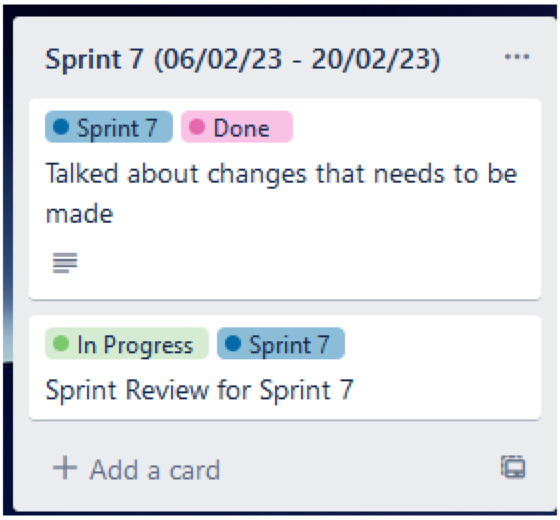
Sprint 5 Sprint 6 Done
First section of the project development should be started, It should review how the project works, and what software and tools you are using

Figure 42 Sprint 6 Board

What went well:	What did not go well
Testing draft was set up. Ethical approval forms were double checked to make sure nothing's missed	Bato3 api was still not working at this point since sprint 4.

Table 11 Sprint 6 Table

9.4.8 Sprint 7



Sprint 7 (06/02/23 - 20/02/23)

Sprint 7 Done
Talked about changes that needs to be made

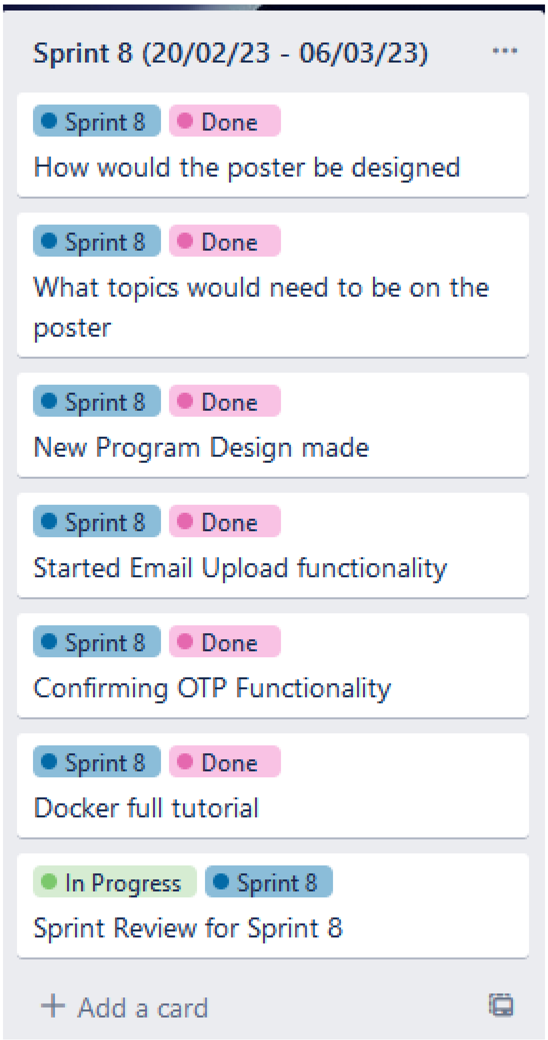
In Progress Sprint 7
Sprint Review for Sprint 7

Figure 43 Sprint 7 Board

What went well:	What did not go well
Supervisor chat has happened during this time, project vision changed.	Old projects focus were changed. A lot of changes needed to be made

Table 12 Sprint 7 Table

9.4.9 Sprint 8



Sprint 8 (20/02/23 - 06/03/23) ...

● Sprint 8 ● Done
How would the poster be designed

● Sprint 8 ● Done
What topics would need to be on the poster

● Sprint 8 ● Done
New Program Design made

● Sprint 8 ● Done
Started Email Upload functionality

● Sprint 8 ● Done
Confirming OTP Functionality

● Sprint 8 ● Done
Docker full tutorial

● In Progress ● Sprint 8
Sprint Review for Sprint 8

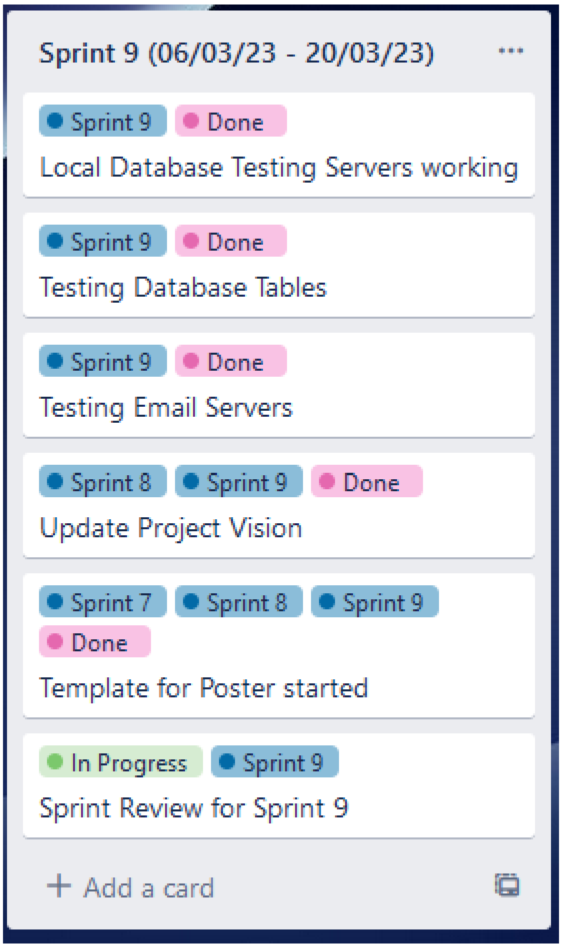
+ Add a card

Figure 44 Sprint 8 Board

What went well:	What did not go well
Poster thought were started to accommodate the showcase. Partial code working at this point in time	Local email servers were not working still at this time.

Table 13 Sprint 8 Table

9.4.10 Sprint 9



Sprint 9 (06/03/23 - 20/03/23)

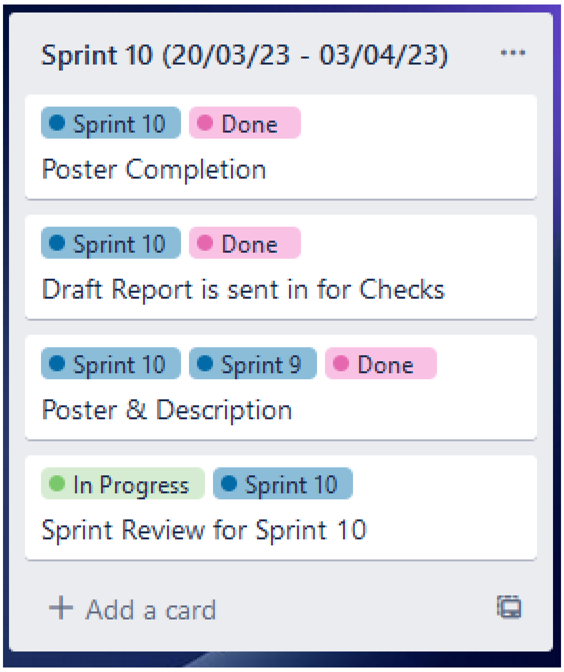
- Sprint 9 · Done
  Local Database Testing Servers working
- Sprint 9 · Done
  Testing Database Tables
- Sprint 9 · Done
  Testing Email Servers
- Sprint 8 · Sprint 9 · Done
  Update Project Vision
- Sprint 7 · Sprint 8 · Sprint 9 · Done
  Template for Poster started
- In Progress · Sprint 9
  Sprint Review for Sprint 9

Figure 45 Sprint 9 Board

What went well:	**What did not go well**
Database is working at this point. Poster continuation.	Local email server was scrapped. Used personal emails in the end.

Table 14 Sprint 9 Table

9.4.11 Sprint 10



Sprint 10 (20/03/23 - 03/04/23)

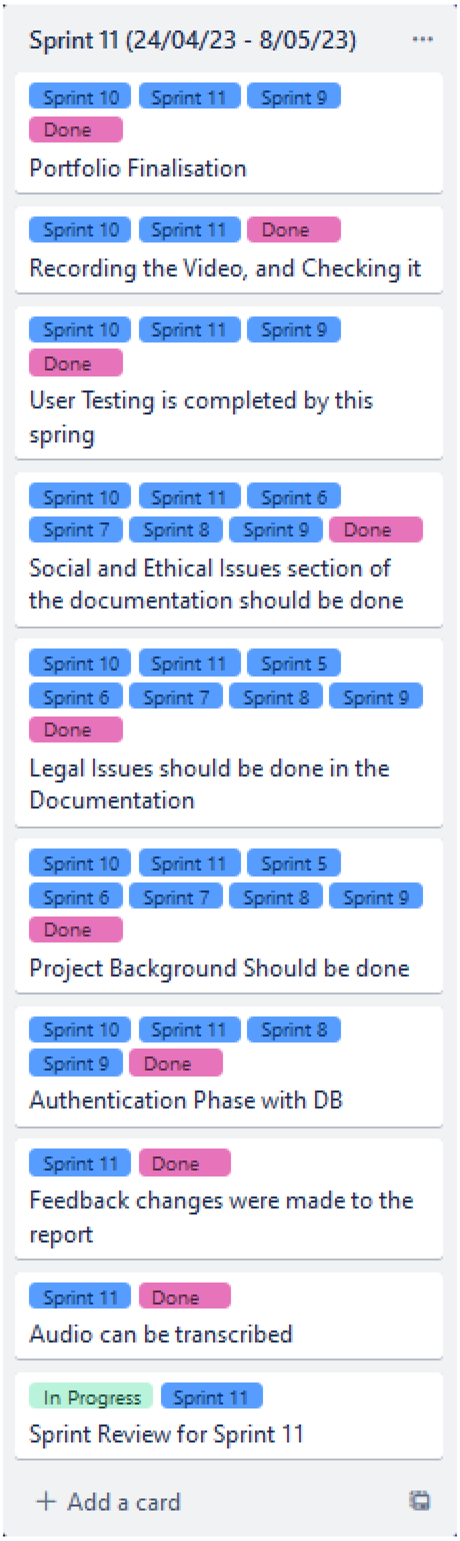
- Sprint 10 · Done — Poster Completion
- Sprint 10 · Done — Draft Report is sent in for Checks
- Sprint 10 · Sprint 9 · Done — Poster & Description
- In Progress · Sprint 10 — Sprint Review for Sprint 10

Figure 46 Sprint 10 Board

What went well:	What did not go well
Poster is completed, Report draft was sent in. Email functionality is working at this point in time.	n/a

Table 15 Sprint 10 Table

9.4.12 Sprint 11



Sprint 11 (24/04/23 - 8/05/23)

Sprint 10 Sprint 11 Sprint 9 Done
Portfolio Finalisation

Sprint 10 Sprint 11 Done
Recording the Video, and Checking it

Sprint 10 Sprint 11 Sprint 9 Done
User Testing is completed by this spring

Sprint 10 Sprint 11 Sprint 6 Sprint 7 Sprint 8 Sprint 9 Done
Social and Ethical Issues section of the documentation should be done

Sprint 10 Sprint 11 Sprint 5 Sprint 6 Sprint 7 Sprint 8 Sprint 9 Done
Legal Issues should be done in the Documentation

Sprint 10 Sprint 11 Sprint 5 Sprint 6 Sprint 7 Sprint 8 Sprint 9 Done
Project Background Should be done

Sprint 10 Sprint 11 Sprint 8 Sprint 9 Done
Authentication Phase with DB

Sprint 11 Done
Feedback changes were made to the report

Sprint 11 Done
Audio can be transcribed

In Progress Sprint 11
Sprint Review for Sprint 11

Figure 47 Sprint 11 Board

What went well:	**What did not go well**
Report is done by the end of this sprint. Video was recorded as well. Program is semi functional	Program does not work fully and it is not bug free

Table 16 Sprint 11 Table

9.5 Docker Compose File

```
services:
  postgres:
    user: ${COMPOSEUSER}
    image: postgres:latest
    ports:
      - "5432:5432"
    environment:
      POSTGRES_PASSWORD: ${POSTGRESPASS}
      POSTGRES_USER: ${POSTGRESUSER}
      POSTGRES_DB: db
    volumes:
      - comp3000:/var/lib/postgresql/data

  pgadmin4:
    user: ${COMPOSEUSER}
    image: dpage/pgadmin4:latest
    ports:
      - "5050:80"
    environment:
      PGADMIN_DEFAULT_EMAIL: ${PGADMINUSER}
      PGADMIN_DEFAULT_PASSWORD: ${PGADMINPASS}
      PGADMIN_LISTEN_ADDRESS: 0.0.0.0
    volumes:
      - comp3000:/var/lib/postgresql/data
      - comp3000:/var/lib/pgadmin

volumes:
  comp3000:
```

9.6 Testing Results

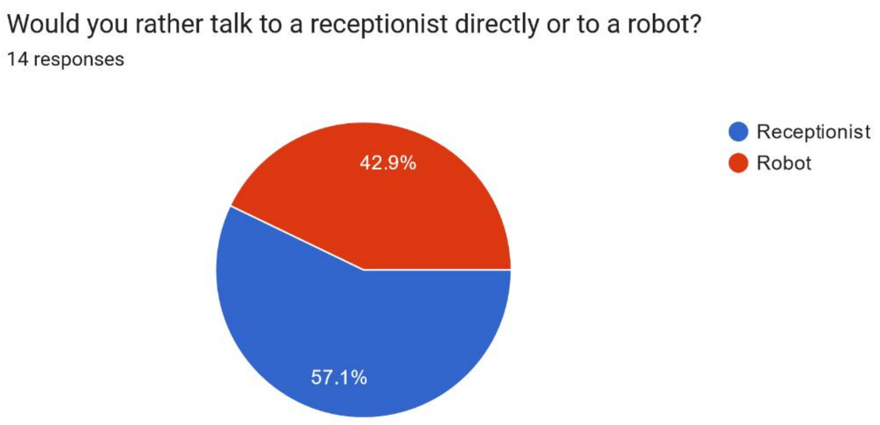
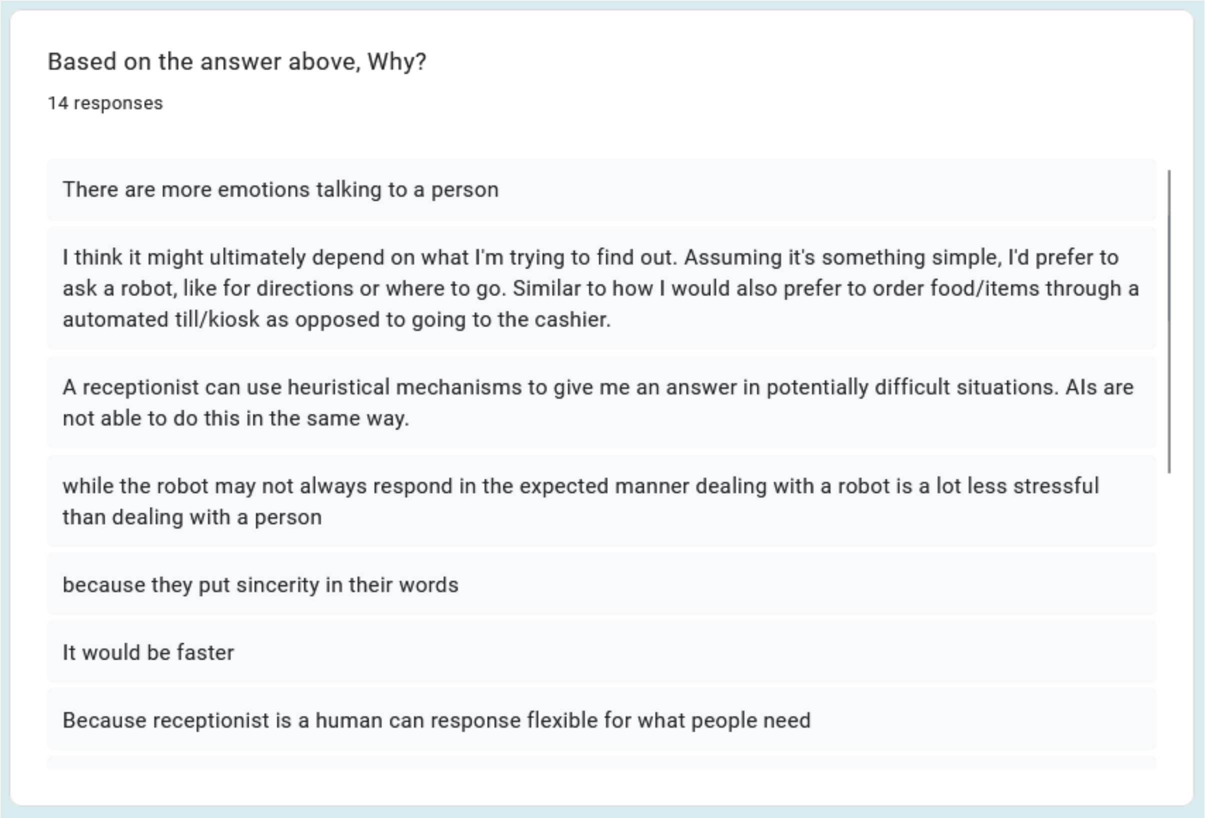


Figure 48 Testing Results Figure #1



Based on the answer above, Why?

14 responses

There are more emotions talking to a person

I think it might ultimately depend on what I'm trying to find out. Assuming it's something simple, I'd prefer to ask a robot, like for directions or where to go. Similar to how I would also prefer to order food/items through a automated till/kiosk as opposed to going to the cashier.

A receptionist can use heuristical mechanisms to give me an answer in potentially difficult situations. AIs are not able to do this in the same way.

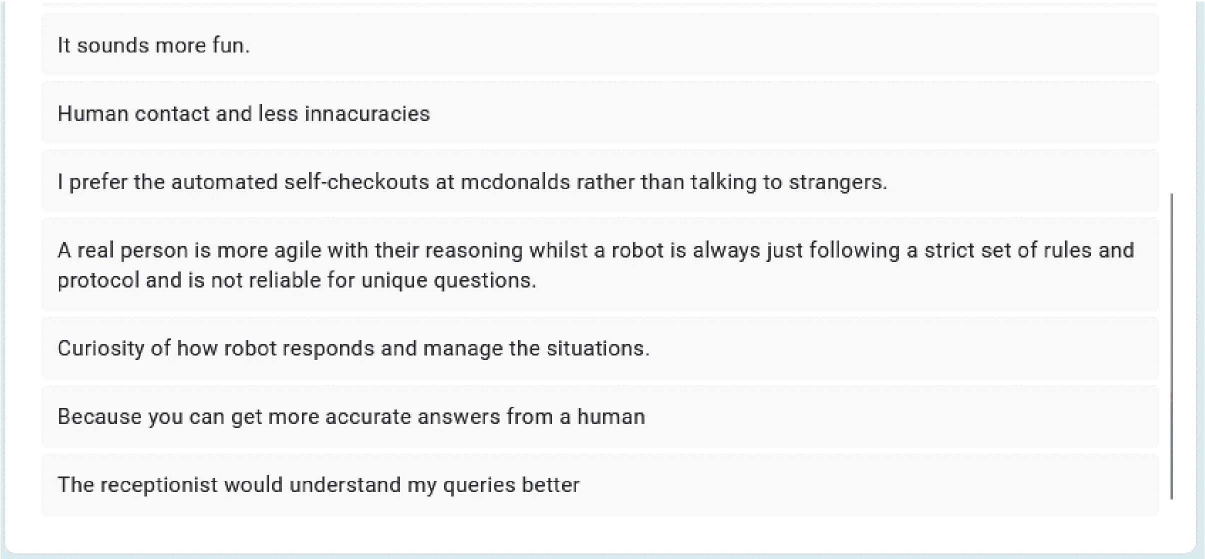
while the robot may not always respond in the expected manner dealing with a robot is a lot less stressful than dealing with a person

because they put sincerity in their words

It would be faster

Because receptionist is a human can response flexible for what people need

Figure 49 Testing Results Figure #2



It sounds more fun.

Human contact and less innacuracies

I prefer the automated self-checkouts at mcdonalds rather than talking to strangers.

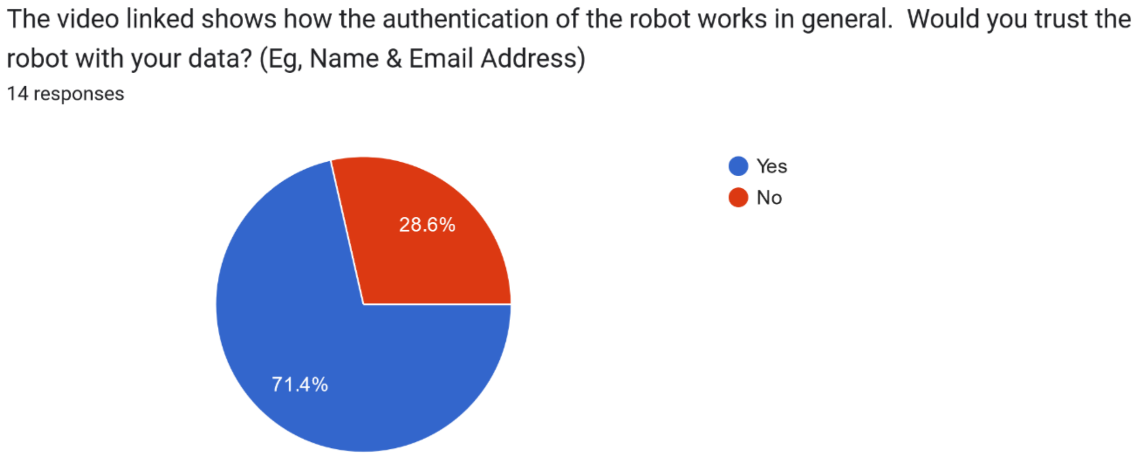
A real person is more agile with their reasoning whilst a robot is always just following a strict set of rules and protocol and is not reliable for unique questions.

Curiosity of how robot responds and manage the situations.

Because you can get more accurate answers from a human

The receptionist would understand my queries better

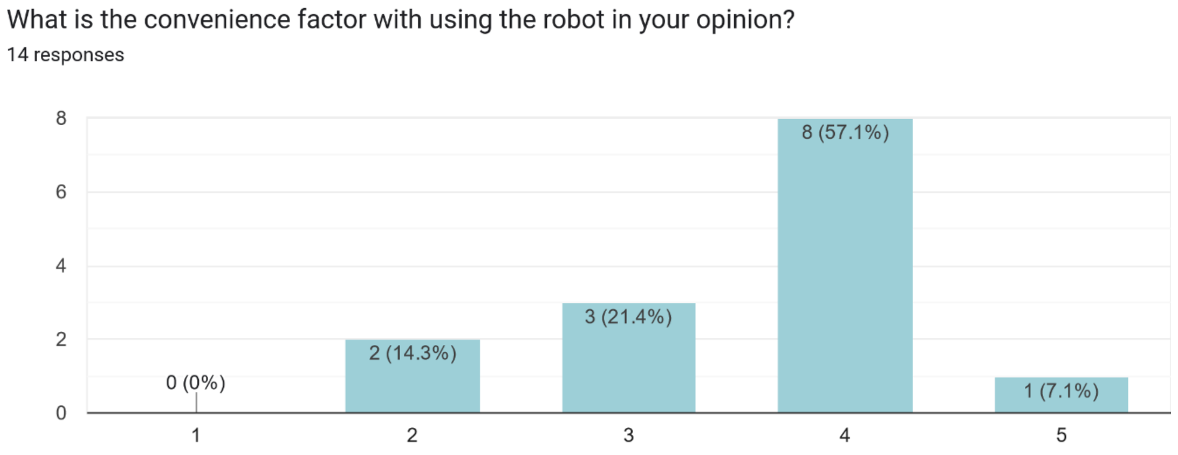
Figure 50 Testing Results Figure #3



The video linked shows how the authentication of the robot works in general. Would you trust the robot with your data? (Eg, Name & Email Address)

14 responses

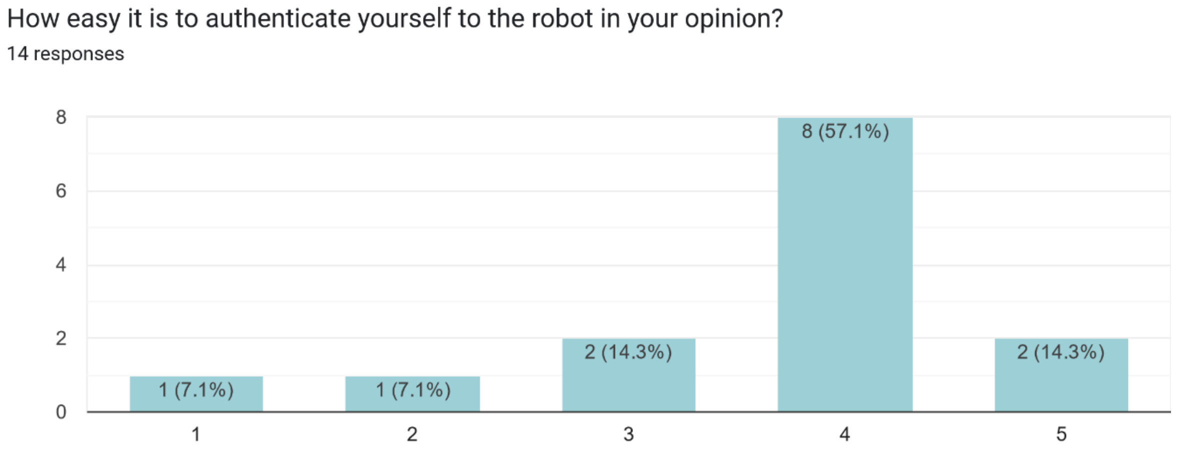
Figure 51 Testing Results Figure #4



What is the convenience factor with using the robot in your opinion?

14 responses

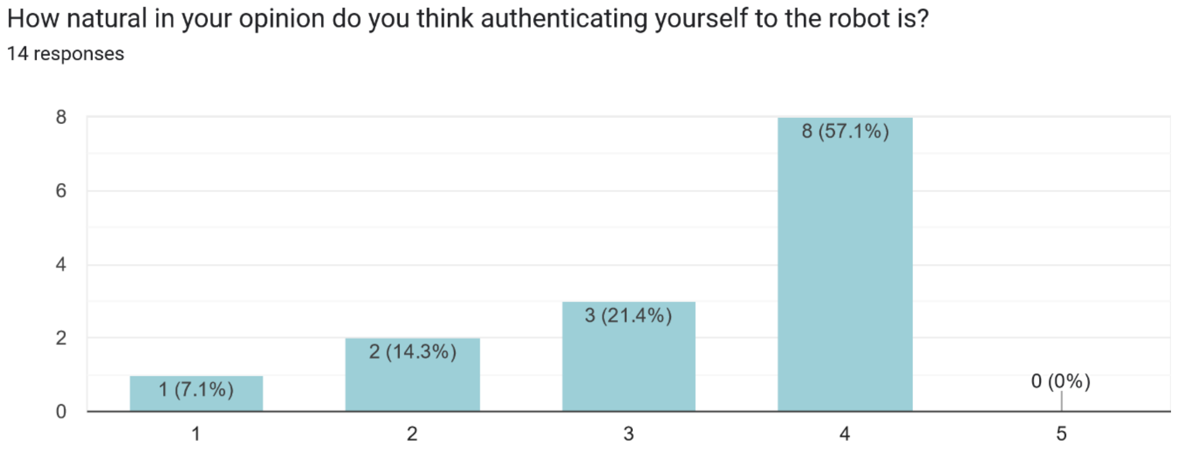
Figure 52 Testing Results Figure #5



How easy it is to authenticate yourself to the robot in your opinion?

14 responses

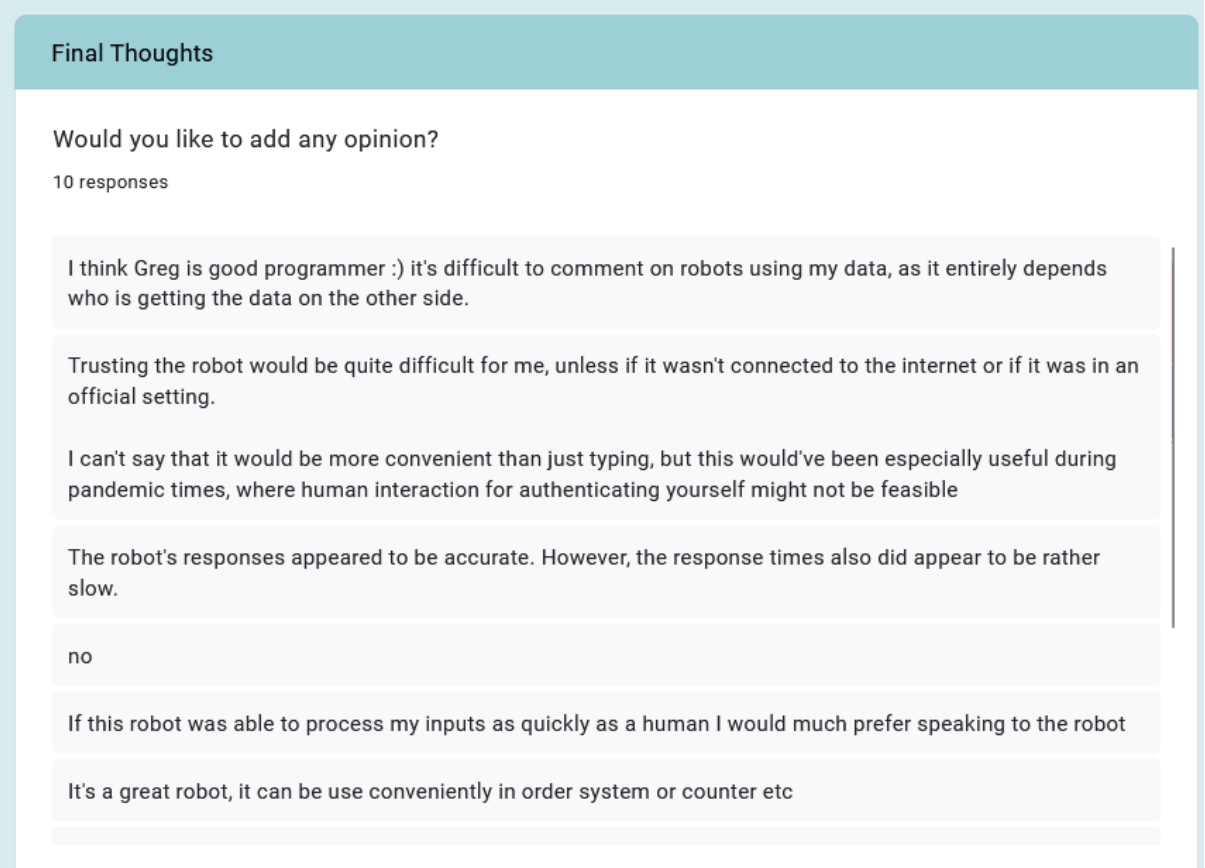
Figure 53 Testing Results Figure #6



How natural in your opinion do you think authenticating yourself to the robot is?

14 responses

Figure 54 Testing Results Figure #7



Final Thoughts

Would you like to add any opinion?

10 responses

I think Greg is good programmer :) it's difficult to comment on robots using my data, as it entirely depends who is getting the data on the other side.

Trusting the robot would be quite difficult for me, unless if it wasn't connected to the internet or if it was in an official setting.

I can't say that it would be more convenient than just typing, but this would've been especially useful during pandemic times, where human interaction for authenticating yourself might not be feasible

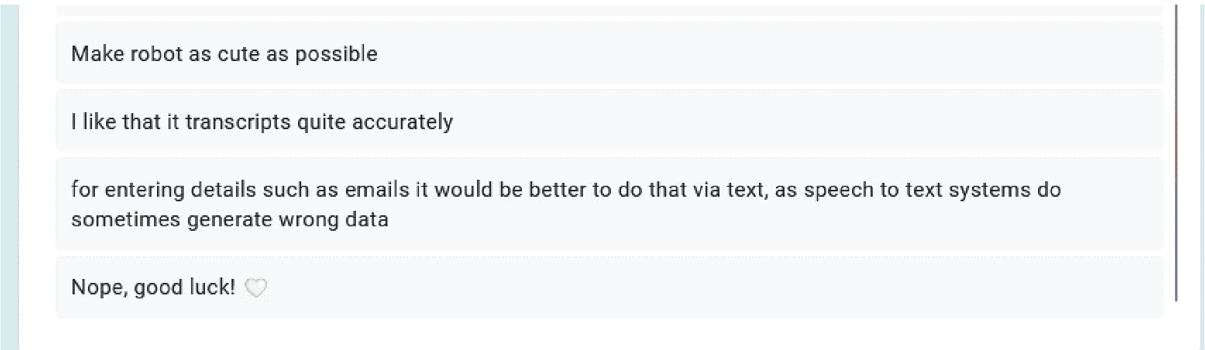
The robot's responses appeared to be accurate. However, the response times also did appear to be rather slow.

no

If this robot was able to process my inputs as quickly as a human I would much prefer speaking to the robot

It's a great robot, it can be use conveniently in order system or counter etc

Figure 55 Testing Results Figure #8



Make robot as cute as possible

I like that it transcripts quite accurately

for entering details such as emails it would be better to do that via text, as speech to text systems do sometimes generate wrong data

Nope, good luck! ♡

Figure 56 Testing Results Figure #9