Phillip Daniel

Room Scanner Project

Computer Science Non-Examined Assessment

Table of Contents

NEA Proposal

For my NEA I would like to develop a product which can accurately depict a 3D space using a 2D representation. To do this, I plan to use a proximity sensor, such as a LiDAR or ultrasonic sensor to calculate the distances from the device to the room's walls. In addition to this, I plan to use a program which will allow you to add objects to the completed scan of the house. This is important as just mapping the walls of a room is not immensely useful, as it does not give you an idea of what the room is used for, whereas if you were able to see that there was a bed in the room, you would immediately be able to identify that it was a bedroom. I also plan to make a user interface such as an app or website, which will allow users to use my product without an expansive knowledge of how it works, opening the market to a lot more people who may want to scan their room. Throughout doing this, I would like to make sure the components I use are cost-effective and easy to source, allowing for similar devices to be created by others with minimum difficulty.

I became interested in room scanners when attempting to find the dimensions of my bedroom to see whether I could fit different sizes of desks into it. I found that it was incredibly expensive to purchase or even rent an industrial grade room scanner, and there aren’t many cheaper alternatives which can accurately give you the dimensions of a room.

Analysis

Introduction:

Recreating an object or environment accurately has been an important skill for decades, as with exact measurements, people are able to effectively redesign a room or help design a missing part for a system.

With the recent Covid-19 pandemic, many people have been in lockdown, and therefore have been stuck at home, with little to do. This has massively increased the amount of people who have decided to spend time doing do-it-yourself home renovation, as they are spending more time at home, and are worried about hiring a contractor because of fears of the pandemic. Because of this, demand in a low-cost scanner product has increased by substantial amounts recently, however the market is currently extremely small, with almost no low-cost solutions for creating an accurate representation of a room, causing some to look at alternatives such as slightly higher end scanners. However high-end scanners are extremely expensive, meaning that for home enthusiasts and amateur designers, they are inaccessible without some type of industrial backing, meaning that they are left to settle with pen and paper, which is incredibly time consuming, and may have to be repeated multiple times to be accurate, should a small mistake be made.

My overall idea is to create a 3D Space Mapper. Therefore, for my project I have decided to create a device which is much cheaper compared to professional products, but also has multiple functions which higher end products do not have, such as the ability to add multiple different named objected to the completed scan. This will allow for a 2D map of a room to be created, which will allow for easy measurement calculations of the length and widths of a room. To do this, I will need to use type of scanner, such as lidar or ultrasonic.

Description of Current Systems

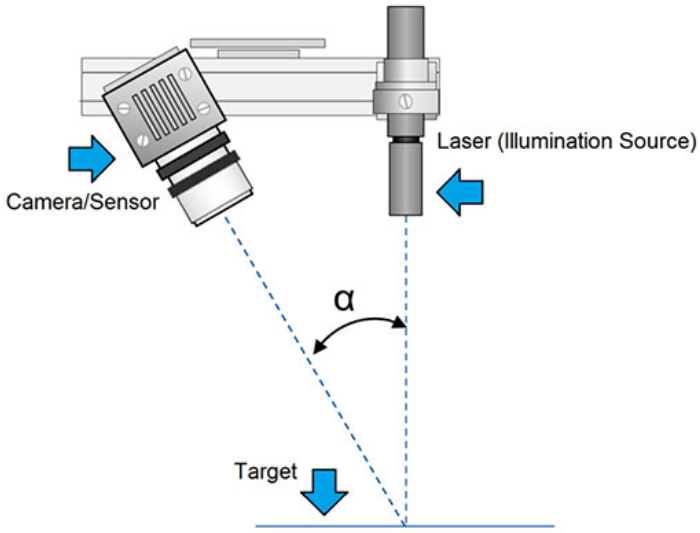
3D Scanning devices can generally be divided into two categories, contact and non-contact scanners. Contact scanners probe objects with mechanical sensors which measure the shape and distances of smaller objects. Non-contact scanners can be divided into active and passive scanners. Active Scanners emit light whereas passive scanners do not emit anything but instead use the already existing ambient light.

Due to the slow speed and reliance on the contact of the object needing scanning, it is inappropriate to use a contact scanner for room scanning, and they are instead more suitable for the scanning of smaller objects.

Active Scanning Methods

Laser-based 3D scanners

Laser-based 3D scanners project a beam of laser onto an object and then capture its reflection with a sensor. The position of the point of the object reflecting the beam is then calculated using trigonometric triangulation. This is a type of active scanner and is one of the most popular methods of 3D scanning, as they are not as sensitive to changing light conditions.

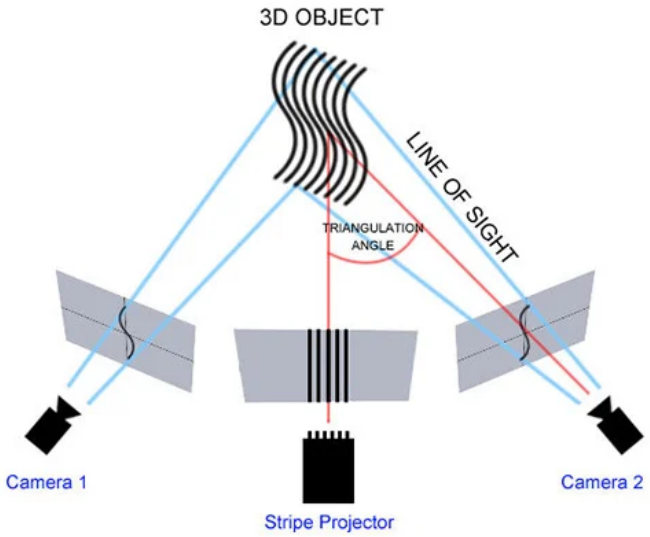


A diagram showing how the distance between an object and the scanner can be calculated through calculating the angle at which the beam is reflected.

An example of a Laser-based scanner is the Artec Ray (pictured below) is a tripod mounted example of a 3D Laser-based scanner. It is priced at approximately £48,000, which is extremely expensive for amateur home renovation use.

Projected Light 3D Scanners

Projected Light 3D scanners project a light pattern onto the object being scanned. Sensors then look at the edges of the pattern and using trigonometric triangulation they can determine the objects 3D shape and the distance the object is from the scanner. Project Light 3D Scanners are popular because of their incredibly fast scan times. However, this type of scanner is very sensitive to different lighting conditions, meaning that it may not work well in extremely bright conditions.

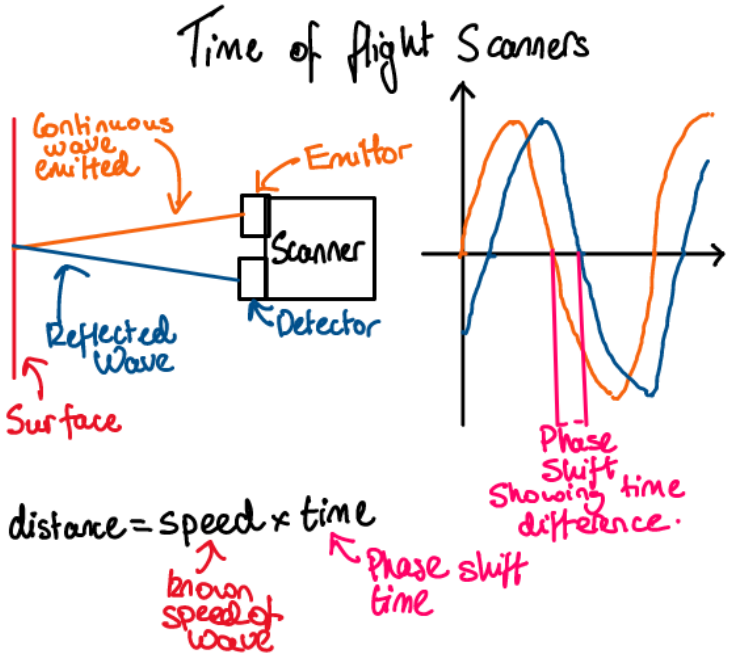


A diagram showing how Projected Light scanners work, by projecting a pattern onto an object and then using cameras/sensors to look at the pattern distortion.

An example of a projected light scanner is the Shining 3D Einscan-Pro 2X Plus (pictured below) is an example of a handheld Projected Light Scanner, priced at approximately £7000, which is cheaper than the laser-based scanner, but still extremely expensive.

Time of Flight Scanners

Time of Flight Scanners, also known as Laser pulse-based scanners, are based on knowing an incredibly accurate value of the speed of light. The length of time a laser takes to reach and object and reflect to the sensor is measured, so the distance from the sensor to the object can be calculated. By rotating, the laser can scan a full 360 degrees. Laser pulse-based scanners are suited for scanning objects up to 300m, they can also scan up to a million points in a second, meaning each 3D scan has millions of points.



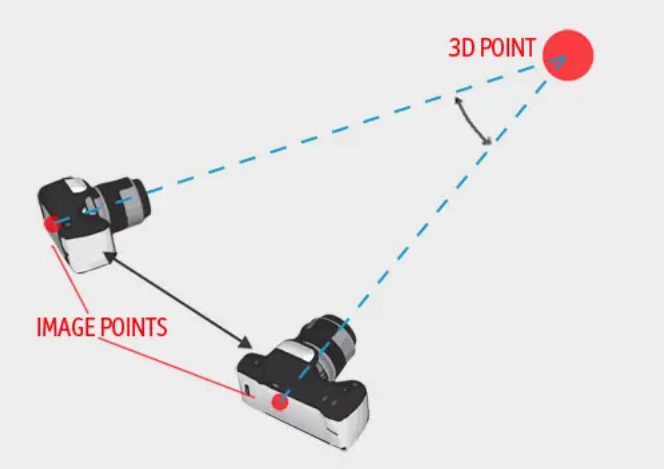
A diagram showing how typical time of flight scanners work.

The Helios2 TOF Camera (pictured below) is an example of a time-of-flight camera, priced at $1495 USD on release. This product is much cheaper than the others, although is still extremely expensive for any amateur to purchase, however, is only a camera, and not the entire device, meaning other components are needed to use the product.

Passive Scanning Methods

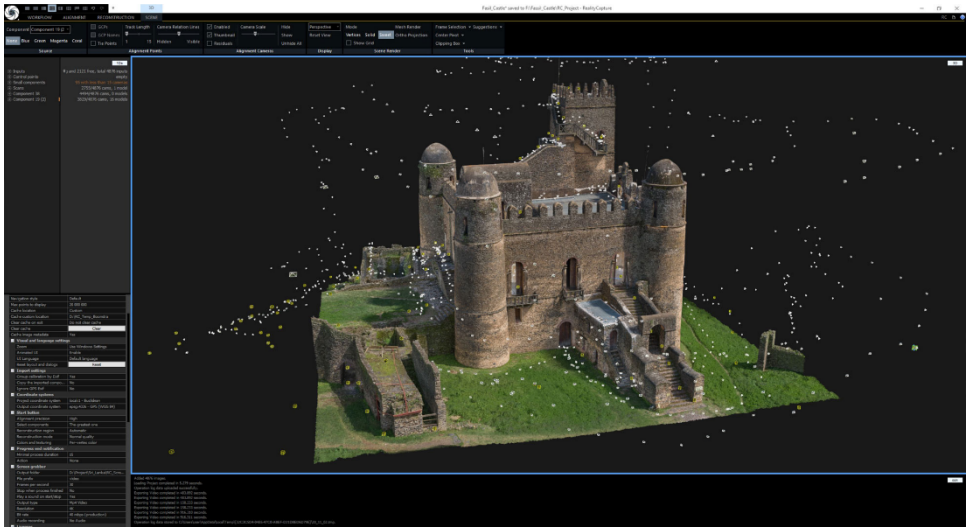
Photogrammetry

Photogrammetry is quite simple, photographs of an object from multiple different angles are stitched together. Software then finds the pixels which correspond to the same physical point and brings the pictures together. The photos can be taken using any camera, with specific settings, however the merging of the photos needs to be done by special software, such a reality capture, blender etc. Photogrammetry can acquire data of an object quite quickly, however the result relies completely on the quality and resolution of the images taken.



A diagram showing how photogrammetry works, with multiple different photos of the same item taken at different angles, and then merged to create a single 3D model.

Reality Capture (pictured below) is a piece of software which can be used for photogrammetry, costing 10 USD per certain amount of images scanner, however a device is also needed to take the photo. Above is a screenshot of Reality Capture being used to bring a UNESCO world heritage site into 3D.



Identification of End-Users

I have decided to aim my product towards home enthusiasts and amateur designers, as they will be most likely to often be creating designs with pen and paper, as due to the incredibly expensive price of most Computer Aided Design software, will only have a basic knowledge of how CAD works. For this reason, I believe that it is important to create a simple and easy to use user interface.

I decided to create a questionnaire to post in home design forums to find out what other functions would be useful, and the uses of the product.

Survey Reponses from Potential Users

Question: Have you ever used a 3D Roomer scanner before? (If not, why?)

Responses:

No, too expensive.

Yes, occasionally as I let out a room.

Yes, I have tried it out.

No, I am not very technological.

No, too expensive for me to use for just my house.

Yes, my company often uses them.

No, they seem quite tedious and complicated to use.

Yes, I have a mobile app.

No, I have thought about buying one but having seen the price, it doesn't seem worth it.

I asked this question as I wanted to know the reasons why people didn't buy 3D room scanners, the results were very interesting, mainly the price and having to understand how the device works. This showed that my device would have to be moderately cheap and simple to use. Asking this question gave me a good understanding of some of the requirements which would attract potential users of the product.

Question: What do you think the most useful functions of a 3D room scanner are?

Responses:

Never used them.

Being able to get a map of the room quickly.

Being able to measure stuff without being in the room.

Never looked into them but seeing the room online would be useful for online listings etc.

Quickly getting the measurements of a room.

The ability to quickly edit areas of the room with CAD.

Being able to show other people the room would be useful due to covid.

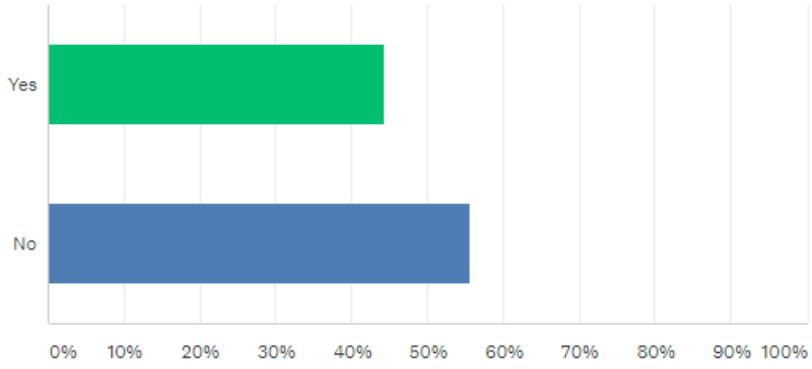
Allows me to add information to CAD software and create designs to 3D print.

Being able to quickly see the dimensions of a room without having to manually measure it all.

I asked this question as I wanted to know what features would be important for my scanner to have, the results showed that my device needs a way to quick and easily access the dimensions of a room.

Question: Do you have any experience with Computer Aided Design (CAD)?

Responses:

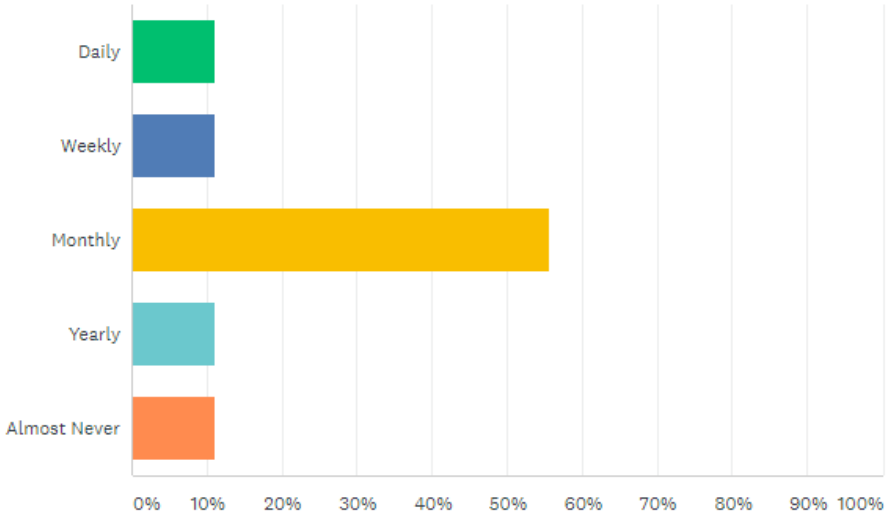


ANSWER CHOICES	RESPONSES	
Yes	44.44%	4
No	55.56%	5

I knew that most people using 3D scanners professionally would have to have experience in Computer Aided Design, however I was unsure about amateur home renovators, so I decided to ask this question as it would allow me to see whether people would be able to use complicated and hard to learn Computer Aided Design software.

Question: How often would you find it useful to have a 3D room scanner accessible?

Responses:

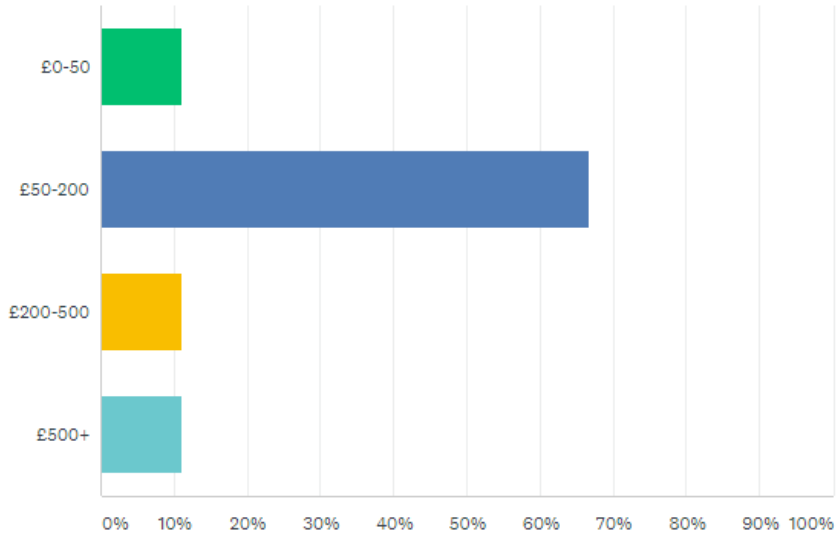


ANSWER CHOICES	RESPONSES	
Daily	11.11%	1
Weekly	11.11%	1
Monthly	55.56%	5
Yearly	11.11%	1
Almost Never	11.11%	1

I asked this question to get more of an insight into how often a product would be used. This helped as it would help me to see how much demand there would be for a product like this and see how much a product would be used and therefore see how expensive a product like this should be, as the more you use something, the more you would be willing to spend on it.

Question: What price range would you buy a 3D Room scanner in?

Reponses:

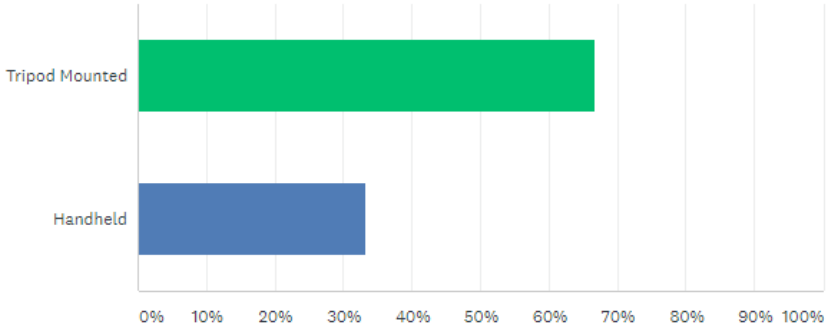


ANSWER CHOICES	RESPONSES	
£0-50	11.11%	1
£50-200	66.67%	6
£200-500	11.11%	1
£500+	11.11%	1

Expanding on the earlier question, I asked how much someone would spend a product like this. The most common answer was £50-£200, which seems like a sensible amount, as it is the amount which components needed for a device like this would cost.

Question: Would you prefer a hand-held scanner or a tripod mounted scanner?

Responses:



ANSWER CHOICES	RESPONSES	
Tripod Mounted	66.67%	6
Handheld	33.33%	3
TOTAL		9

I asked this question to decide how I wanted my product to scan the area, as tripod mounted, and hand-held scanners are the most common way of using 3D room scanners. The most common

answer was tripod mounted, I would assume this was the answer because it is easier to use, as no human input is needed to take the scan, only positioning is needed.

Question: What is the size of the average room that you would want to scan?

Responses:

> Not too large, medium sized rooms.
>
> Max of 6 metres in both directions.
>
> Small Bedrooms.
>
> Small to medium size.
>
> A few metres in both directions.
>
> Usually, a minimum of 4 metres in both directions.
>
> Couple of metres length and width
>
> 3-6 metres by 3-6 metres
>
> 5-10 metres both sides.

I asked this question to get information about the size of rooms that I would have to scan, to know the range of scanners I would have to buy. The maximum size was 10 metres by 10 metres, with most responses ranging between 4 and 6 metres on both sides. Therefore, I need to buy components which can scan this sort of range.

Selection and Interview of End User

I decided to choose an end user so that I could ask what they wanted for the product as well as allow them to test my product so that I could find ways to improve my product. In the end, I chose my end user to be Richard Daniel, who during the Covid 19 lockdown period, decided to renovate one of rooms in his house, because of this first-hand experience, I was able to get information on what processes many had to settle with when they were not able to afford high end, industrial equipment, this allowed me to see the bare minimums of the product that would be needed, and additionally to see what kind of features would benefit a user of the product.

Interview with Primary User Richard Daniel, an avid home enthusiast.

Question: Have you ever investigated using a 3D scanner?

> Response: At one point I was looking into renting out on for getting a scan of a bedroom which I was planning on renovating, however they were incredibly pricey, and with it being a one-time use, getting to understand the software etc just wasn't work it. So, ended up just getting the dimensions by hand. Should the price of a device like it reduce, would love to have one as it would make everything so much quicker and simpler when checking if furniture will fit.

Question: Are there any devices that you would like to use in conjunction with the 3D scanner?

Response: I would like to be able to easily print out a map of the area I have scanned, e.g., length, width, and items in the way. I'd also like to be able to get dimensions easily so that I can quickly make things using my CNC router which I can then add to the room.

Question: How much would you be willing to spend on a 3D scanner?

Response: I think the maximum that I would be willing to spend is about £150, as I do not see myself using it often enough to spend more than that, however depending on the features of the device, I would spend less.

Question: What features do you think would be useful for a 3D scanner?

Response: I would like to be able to quickly print out a plan of the room that I have just scanned, as it allows me to be able to quickly use the measurements to create things. I'd also like to be able to just leave the scanner in the room and it be able to just do its job, and only need to come back to get the data. It would also be particularly good if it was simple to use.

Question: How often would you find yourself using a product like this?

Response: I think that I would use something like this quite often in the next couple of weeks, as I am currently trying to refit one of the rooms in my house. A product like this would be incredibly useful for that. As I could quickly see the area of the room etc for when I am buying furniture.

User Wants and Needs

Once I had completed my interview with my primary user, I also asked him to make a list of the features that they thought would be important or useful for my project to be able to do, and to rank them in order of importance.

	User Need	Importance
1	A 2D map of the area scanned should be created and it should be in a format which can be used in conjunction with other applications (like word), e.g., jpg, png, pdf.	Need
2	The 2D map should be easily measurable, with a scale or other to compare measurements.	Need
3	The product should be mountable on a tripod and should only require human input to begin the scan, e.g., doesn't need someone to spin it around, can just be left to complete the job once it has been started.	Want
4	The components needed to create the product should cost less than £150, to allow for it to be suitable for the key users of the product.	Want
5	The product should have scanning range long enough to scan moderately large rooms, e.g., 10 metres by 10 metres rooms.	Want
6	The product should be small and easily portable so that it can be used in many different situations.	Preferred

Set of Objectives

Once I had gained information from my primary user, I created a list of objectives, using my User Wants and Needs in combination with information from my survey responses, which I could then review throughout the creation of my product, and once the product had been completed to order to make sure my product followed these objectives, so I could decide whether my product was successful or whether there were areas in which I could improve.

	Objective	Reason for implementing this Feature
1	The program must be able to take an accurate measurement of the distance from product to the wall it is facing, using a Lidar distance sensor.	I think that this feature is incredibly important, as if this is going to replace a normal tape measure, it needs to be just as accurate as it, or there is no point in the product.
2	The program for the product should be written in a way which will allow for easy adaptations of the product, allowing for functions to be added to edited.	This is an incredibly important feature, as it would mean that some users could decide that they wanted to change sections of the program, and would be able to add new modules easily, which would allow for the product to do other things, e.g., instead of just emailing a user a finished scan, it could send a text message with the scan attached to it.
3	Once the product has taken measurements of the entire room, it should be able to create an image showing an accurate 2-dimensional representation of the room, this image should be easy to understand and in a commonly used format to allow to be opened and shared by everyone. This image should be shared in an easily accessible way.	Using a commonly used format for the final scan file allows for the user to gain information about their scan and increases the capability of the product produces an easily accessible and easy to use piece of information upon the competition of the scan.
4	The product should have a simple and easy to use interface to allow people who aren't as confident at technology to use it with easy, allowing for more potential users.	This feature allows more flexibility on potential end users of the product, as it allows for many different people to be able to use the product, this is important as it increases the number of potential users for this product.
5	The product should allow for objects to be added to the scan to allow users to add more detail to their completed scan should they want to.	This feature is important, as I don't believe a full room scan has been completed without the addition of objects within the room, as without them, you have just shown the structure of the walls within the room, and you cannot tell what the room is from the scan. Having objects in the room will allow for someone to identify the purpose of the room, which may help when trying to redesign the room.
6	The product must not require constant human altercations, e.g., should be left to complete its task without having to be constantly adjusted.	I think that this is another very important feature, as since this is a product which is meant to replace existing product, such as a tape measure, I need to be able to show the benefits of using this, instead of the previous product, one of which is not having to take all the measurements yourself.

Acceptable Limitations

In addition to the creation of a set of objectives, I also discussed potential acceptable limitations for my product with my primary user, which were then agreed to.

	Features I would like to include	Reason for not including
1	The product should be small and easily carry so that it can be used in many different situations.	Because this project is computer science based, I will not be focusing on the design/aesthetics of the product.
2	The product should not need to be plugged in to power or ethernet connection. This will allow the product to be more portable.	As this is a first demonstration of the product and this is not a primary focus of the product, I would prefer for the system to work effectively then work without connections.
3	The product should be designed in a way which will allow multiple different people to use multiple different devices simultaneously. Which would make each product completely independent of another.	Although this would be a goal in the future, I believe as this is just a demonstration of how the product could work, this isn't needed, and I would rather focus on making sure that the product works exactly how my end user would like it to work, as this feature may be time consuming and may require expensive resources for running servers which I do not have.
4	The product should be able to scan a room very quickly, allowing for multiple rooms to be scanned in a very short amount of time.	Because the bulk of my code will be using Python, an interpreted language, the application will not be able to run extremely fast. In addition, the focus of my product is the accuracy of the scans, meaning that the time taken of a scan is not a massive factor.
5	The product shouldn't require a Raspberry Pi or similar computer, and instead should just be pluggable into a laptop or computer and should be able to run through that.	Although I think this would be quite good for my product, I think that it is unneeded, and because there are multiple different operating systems, which my product would have to work with, I think that this may be incredibly time consuming, and would not align with my main goals for this project, which I would rather focus on
6	The program should run as efficiently as possible and have as small of a file size as possible in order for the files to be easily distributed.	Although in the future I believe this is important, I think it would be time consuming and this project is more based on demonstrating that this kind of product can be created at a low price. For that reason, I don't think that this objective is necessary at the current point.

Further Investigation:

Further Investigation into Types of Distance Sensors

I am going to need a component which can measure the distance between itself and a point to get the measurements of a room. Therefore, I decided to investigate distance sensors so that I could find which would be the most appropriate for what I would need it to do.

Ambient Light Sensors

Ambient light sensors use photodiodes to measure the intensity of light. They do this by measuring the photo voltage, the size of this value depends on the sensitivity of the photodiode

and the integration time (the time taken for measurements). With ambient Light sensors, there are formulas to calculate an absolute value in lux from the bare reading. However, there are some issues with ambient light sensors:

Common light sources have an infrared light content, the photodiodes are sensitive to this IR light, resulting in high readings, as well as this, the photodiodes don't have a constant sensitivity over the entire light spectrum and because different light sources have different spectra, there are different measured values for the same luminous intensity. Some sensors have a separate infrared photodiode which corrects the measured value of visible light; however, the IR spectra of different sources are also difference, resulting in the same issue.

The SparkFun RGB and Gesture Sensor – APDS-9960 is an example of an Ambient Light sensor and is one of the best gesture sensors on the market, even being used for the Samsung Galaxy S5, and is priced at $14.95**.**

Ultrasonic Sensor

Ultrasonic sensors emit a high-frequency sound wave, when an object is present with the sensor's detection range, it will reflect a sound wave back to its source, distance is then measured by how long it takes the soundwave to return. Ultrasonic sensors are a cost-effective solution and can be used in environments with high levels of dust or humidity, however the sensors accuracy is affected by soft materials, which absorb the sound waves, which makes it hard for the sensor to see the target, they also have a moderately low detection range.

The HC-SR04 Ultrasonic Sensor is an example of an Ultrasonic sensor and is relatively cheap compared to other methods of scanning, at £6.91, however has a small range, meaning to may not be suitable for my uses.

LiDAR Sensors

LiDAR sensors are a type of laser distance sensor which measures the distance from the surface, they work by emitting pulses in all directions and measuring how long it takes for them to bounce back from the object. LiDAR differ from ultrasonic sensors as they use a different frequency, and LiDAR uses a laser beam instead of sound waves to measure the distance. Because LiDAR sensors use laser beams, they aren't affected by light. They also have a large measurement range and high accuracy, however, are usually more expensive and may miss objects like glass.

The Benewake TFmini S Micro Lidar is an example of a LiDAR sensors, priced at £29.75. It has a 12-metre detection range and a field of view of 2.3 degrees.

Distance Sensors Conclusion

From look at different types of distance sensors, I have concluded that a LiDAR sensor would be most appropriate, as it has the largest range, and the highest accuracy out of the three. In addition, because they are unaffected by light, it will mean that measurements will not differ depending on the time of day, or whether you have your windows open or not, allowing for an overall more consistent result.

Further Investigation into Motors

There are many different types of motors which I could use for my product, however I was unsure of which would suit my product the best, so I decided to investigate the most common types in order to help me decide which would be best for me.

DC Motors

The first and most common type of motor used in electronics is the DC motor. They are incredibly easy to use, however are very unprecise. They are typically used to power conveyors, turntables, or other items where adjustable speed and constant or low speed torque is required. However, there are some downsides to DC motors, mainly the lack of feedback about the motors position, as it is an open loop operation (this means that it has no self-regulation or control action over the output value, meaning each input setting determines a fixed operating position for the controller), however this can be solved using a rotary encoder, which is attached to the motor shaft and will measure how much the motor has moved.

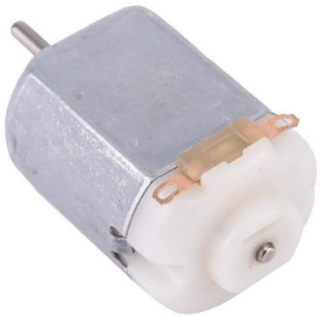


Image of a simple DC Motor.

DC motors are typically cheap to buy, depending on the specifications, with some which would be suitable for rotating a small sensor being less than £3.00.

Stepper Motors

Stepper motors are slightly more complicated than DC motors and have more control. Instead of having 2 connections, like a DC motor, stepper motors typically have up to 6, these connections energize parts of the motor winding is a specific order, which causes the shaft to spin. Each time a connection is energized, it causes the motor to move by a very precise amount. However, a specialized motor driver is required for sending the pulses to different parts of the motor. Stepper motors are typically used for printers, and other devices which require precise movement. Although stepper motors are more precisely controlled then DC motors, they also do not provide feedback about their position, this means that if the stepper motor were to slip, it could miss steps and end in an incorrect position, without the user knowing.

Picture of a commonly used stepper motor and stepper motor driver.

Servo Motors

Servo motors combine a DC motor with built in feedback, however, have a limited range of motion, typically they rotate about 180 degrees, and are limited at each end of travel. They are typically used for steering RC cars and planes and are popular for hobbyists, meaning that large, powerful servo motors are not widely available. Like stepper motors, servo motors are controlled by pulses. Because servo motors do not spin freely, this means that the speed of the pulses does not control the speed of the motor.

Image of a micro servo motor, which are commonly used in electronic projects.

Motor Investigation Conclusion

From looking into these different types of motors, I think that a stepper motor would be most suitable for my use, as it gives me control over the number of rotations done and doesn't have a limit on the degrees of rotation like a servo motor, which would mean I could only scan half a room.

Further Investigation into Image Manipulation

For my project to work well, I knew I had to find a way to manipulate or create images of the scanned room. So, I decided it would be important for me to investigate how this could be done.

When looking, the most useful solution seemed to be using the NumPy and PIL modules, which would allow me to create a NumPy two-dimensional array, with each value in the array representing a single pixel, this image array could then be converted into an image file using the PIL module. This would allow me to easily manipulate a simple image and create a two-dimensional representation of the room. However, it would make it slightly difficult to add letters, as there was no function within either of the modules which allowed for adding text to the array, so I would have to work out a way to manipulate the array before the image creation, or change the image after it had been created, so that I could add objects around the room.

Another alternative solution that I found was to use the OpenCV library, which allowed for very simple manipulation of images, I found that it would be especially easy to add text onto a created image, as there was a built in function called putText() which would allow for text of multiple different fonts to be added. However, OpenCV didn't seem to have an efficient way of manipulating individual pixels of an image, which would mean I would find it difficult to create the representation of the room.

Proposed Solution

From the research that I have gathered during my analysis and investigation into the problem and potential solutions, my proposed solution to this problem will be:

I will be creating a custom program to interface with a website, to allow for control over a Raspberry Pi, which will control a stepper motor and lidar sensor module, which will be placed in the centre of the room, and once given settings will rotate 360 degrees, whilst taking distance measurements. Once the product has rotated 360 degrees, it will then represent the room as an image. I will be creating a png image, as it allows me to keep the file universal for other programs, I may wish to use it with in the future, and it is also a widely used file format. For users to interface with the product I plan to use PHP for website development, and to control

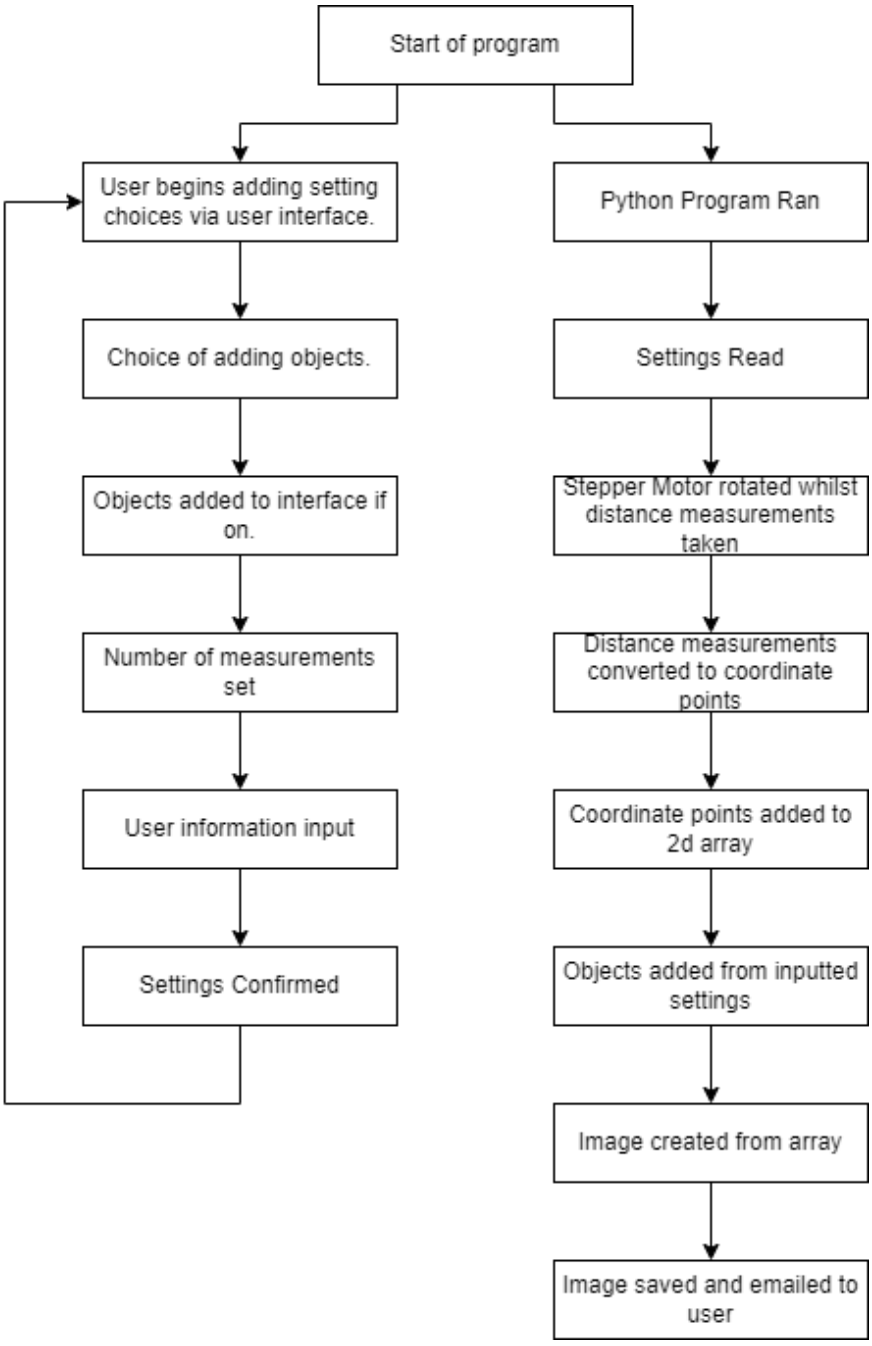
the Raspberry Pi, I plan to create a python program. The website will allow for the user to input different settings for their scan, such as how many distance measurements they want to take in the scan, and their email address for the scan to be sent to. In addition, I plan to use the website to allow for the user to add objects to the scan, which will make the scan more detailed, this will be done by the user inputting the name of the object and the distance and angle in comparison to the scanner. These objects will then be plotted on the scanned png file before it is saved and emailed to the user, allowing for the completed scan to be easily accessible for the user.

Documented Design

Overall System Design

Top-Down Diagram

I created a Top-Down diagram, which shows the overall functions and classes of the program, showing the main aspects of the program and shows how the program should flow.



Critical Path Analysis

I have split the project into multiple different segments, each with a different goal, once all these segments were completed, they could then be joined together into the finalised project. This made the project much easier, as once I had completed one segment and then tested it to make sure that

it worked fully, I could then move on to the next segment, knowing that the previously created programs for other sections would work well. Each segment would need to be completed for the next section to be completed, giving an order in which, the program had to be created in.

Segment One – Measurement Control

On the top-down diagram, this will cover the "Stepper Motor rotated whilst distance measurements taken" section. This segment should accumulate in a python program consisting of a function, which when called will output a distance measurement.

This covers Objective 1.

Segment Two – Rotation Control

This segment consists of controlling a stepper motor, specifically rotating 360 degrees, with different steps amounts (e.g., stops for a measurement every certain number of steps). It should accumulate with a python program which consists of a function, which when inputted with the number of steps in one 360° rotation, should rotate the stepper motor. On the top-down diagram, this will cover the "Stepper Motor rotated whilst distance measurements taken" section.

This covers Objective 6.

Segment Three – Output data

In this segment, once a python program has been completed, an image file of a completed scan should be created. This section should accumulate with a python program consisting of parts of segment one and two, which should take distances measurements as the stepper motor rotates. Once these measurements are taken, they should be plotted onto an image, which should be saved on the raspberry pi. On the top-down diagram, this will cover the "Image created from array" section.

This covers Objective 1 and partially Objective 3.

Segment Four – Addition of Objects

This segment should accumulate with a python program, which will take an input of an object name, distance, and angle, and plot the objects name onto an image saved onto the raspberry pi. On the top-down diagram, this will cover the "Objects added from input settings" section.

This covers Objective 5.

Segment Five – Sharing of Files

This segment should accumulate with a python program, which will take an input of an email address and a file name and should then email the file which has been inputted as an attachment to the inputted email address., this will cover the "Image saved and emailed to user" section.

This covers Objective 3.

Segment Six – Website Design

This segment should accumulate with a simple to use website which allows users to input information about the scan, and objects that they want to add to their scan, which can then be transferred to the raspberry pi in the next segment. This will cover the entirety of the user interface section.

This covers Objective 4.

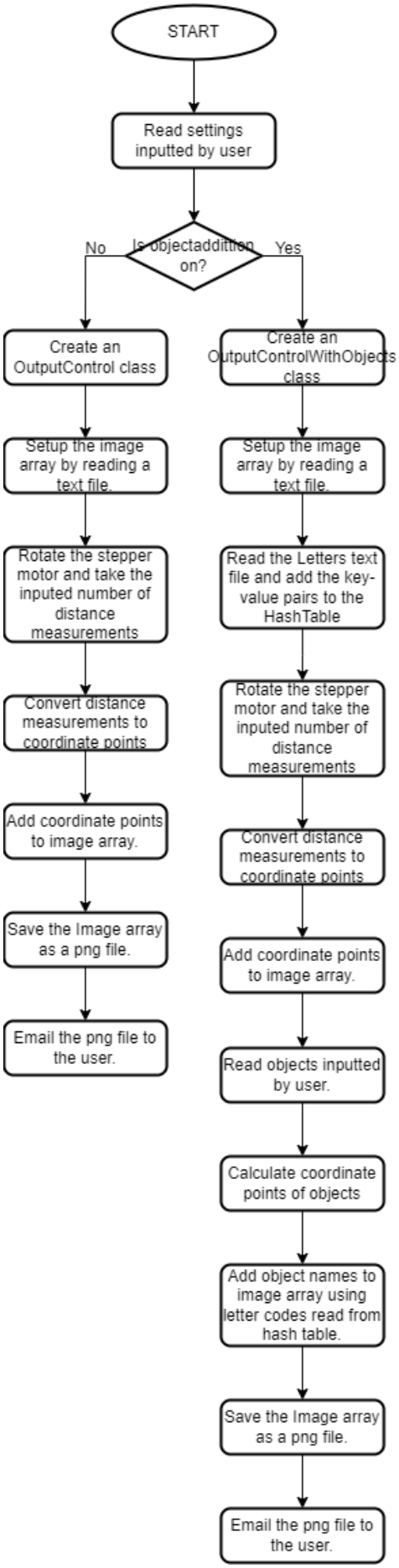
Segment Seven – Communication between Raspberry Pi and Website

This segment should accumulate with adaptations to the website, and a python function which will allow for the settings that the user has input into the website to be read by a python program. This will cover the “settings read” section on the top-down diagram.

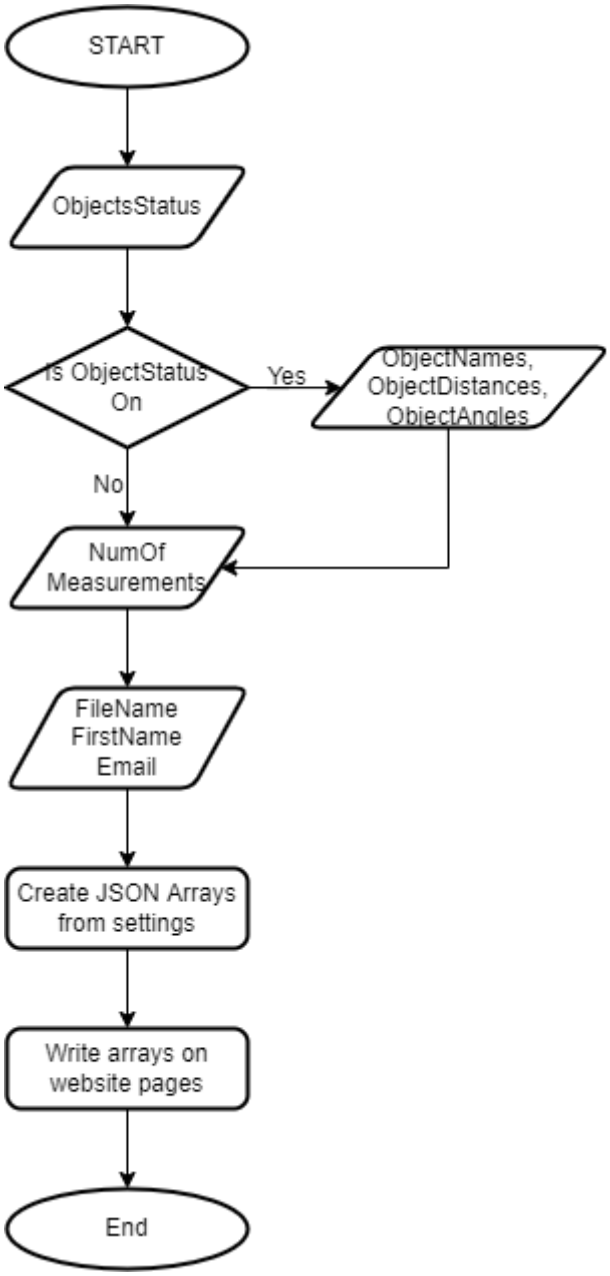
This covers Objective 2.

System Flowchart

I created a flowchart to provide an in-depth view into the operation and those processes and logic of my products code.


START
Read settings inputted by user
Is objectaddition on?
No
Yes
Create an OutputControl class
Setup the image array by reading a text file.
Rotate the stepper motor and take the inputed number of distance measurements
Convert distance measurements to coordinate points
Add coordinate points to image array.
Save the Image array as a png file.
Email the png file to the user.
Create an OutputControlWithObjects class
Setup the image array by reading a text file.
Read the Letters text file and add the key-value pairs to the HashTable
Rotate the stepper motor and take the inputed number of distance measurements
Convert distance measurements to coordinate points
Add coordinate points to image array.
Read objects inputted by user.
Calculate coordinate points of objects
Add object names to image array using letter codes read from hash table.
Save the Image array as a png file.
Email the png file to the user.

In addition, I have also created a flow chart which shows how the flow of input data will be made by the user on the user interface.



IPSO chart

I created an IPSO (Input, Process, Storage, Output) chart. It shows the flow of data in a system, showing how an input is processed to get the desired outcome

Input: These are the human or external sources from which data is inputted.

Process: This is the algorithms which process the data within the program

Storage: This shows where external (databases, files) and internal (data structures, such as arrays) data storage.

Output: This is what is outputted to the user, such as information being printed.

IPSO	Program Section	Item
Input	Inputting name, email, and file name.	Changing settings using the website interface.
	Turning Object Addition on or off.	
	Setting number of measurements in a rotation.	
	Adding number of objects, object names, object positions and object angles compared to scanner.	Adding objects using the website interface
	Reading from a text file	Reading initial pixel locations for setting up the image array.
		Reading letter codes to be added to the letter hash table.
	Controlling the Lidar sensor.	Getting a distance measurement.
Processing	Reading a json array from a website page.	Reading the settings inputted by the user.
	Controlling the stepper motor.	Rotating the lidar sensor a full circle.
	Control of a Hash Table	Hashing a key value
		Adding a value to the hash table
		Retrieving a value from the hash table
	Calculating coordinate points	Calculating a point to be added to the image array from a distance and an angle.
	Changing the value of a position in an array.	Changing the colour of a point within the image array.
	Converting from an array to an image.	Converting the image array into a png image file.
Storage	Saving an image	Saving the completed image array as a png file.
Output	Sharing the image	Emailing the image file to the users email address

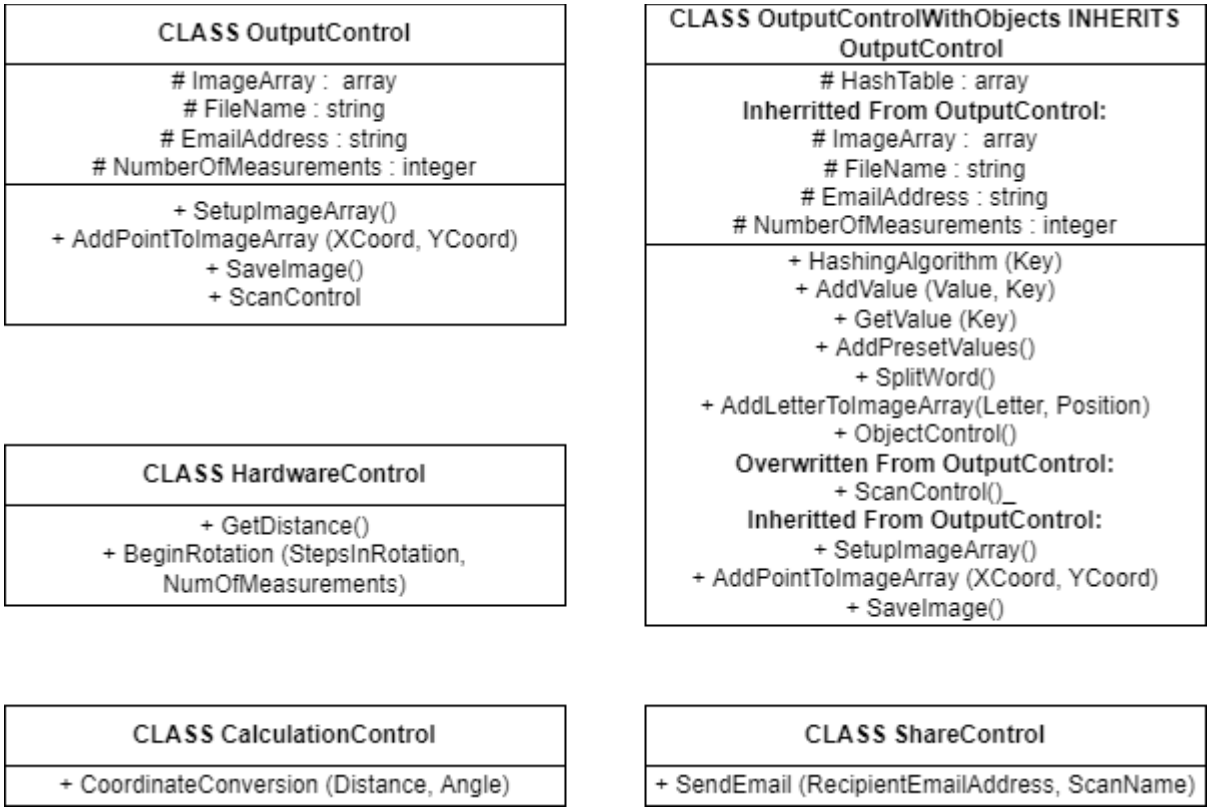
Object Orientated Programming Classes

I decided that my program structure would be best made in an object-oriented paradigm; so, I have created a class diagram below:

This diagram shows the class diagram for my project, shows that OutputControlWithObjects will inherit from the OutputControl Class. I have designed the classes so that they all have a different role to play in the project, with the CalculationControl class being the class which does the mathematical based parts of the projects, like converting from polar coordinates to X-Y coordinates, HardwareControl controlling the LiDAR sensor and stepper motor, ShareScan controlling the distribution of completed scans, and the OutputControlWithObjects and OutputControl classes controlling the overall operation of the product.

In addition, I have also created a class definition diagram for all my classes. This helps me plan my program, as I know which functions, I should be using for what, and where these functions should be used. I have created two diagrams, both showing the classes and their functions.

I have made the ShareControl class with only one function, as in the future I may add other methods of sharing a completed scan, such as through social media, such as discord or Instagram. Therefore, I have created a separate class for sharing so that it will be moderately simple to add different sharing methods to the program. I have also done the same for the CalculationControl class. This allows for one of my objectives, which was "The program for the product should be written in a way which will allow for easy adaptations of the product, allowing for functions to be added to edited", I believe this will help with this objective.



The OutputControl Class

Attribute	Description

FileName	This is a string variable, holding the name of the scanned image which is to be saved as. This value is taken from the input of the user on the user interface.
EmailAddress	This is another string variable, holding the email address of the user which the completed scan should be sent to. This value is taken from the input of the user on the user interface.
NumberOfMeasurements	This an integer variable, holding the number of distance measurements which should be taken in one rotation. This value is taken from the input of the user on the user interface.
ImageArray	This is a two-dimensional array, of 500 by 500. With each position used to hold the pixel colour information of a single pixel within an image.

Method	**Description**
SetupImageArray()	Once all the settings have been confirmed by the user, and the user has started the scan, the SetupImageArray function is used to alter the 2-dimension image array so that it contains information such as the scale for the image, which shows the length of a 1 metre section, and shows the centre point, where the room scanner is located within the room.
AddPointToImageArray()	This function will take the input of the calculated XCoordinatePoint and YCoordinatePoint and change the value of that specific point to 0, which will represents the colour black.
SaveImage()	The purpose of this function is to convert the ImageArray contents into a png image file, with the name stored in the variable self.FileName, and then save the image file in the CompletedScans folder.
ScanControl()	This function will control the room scan process, this means that is will call all the other functions in the correct order, so that a scan can be created. Firstly it will call the SetupImageArray() function, then the HardwareControl.BeginRotation() function, once this has been completed the returned value from this function will be used as the parameters for calling the CalculationControl.CoordinateConversion() function. The returned value will be used in the AddPointToImageArray() function, finally the SaveImage() function will be called, and the image will be shared using the ShareControl.SendEmail() function.

The OutputControlWithObjects Class

Attribute	Description
FileName	Inherited from its parent class, OutputControl.
EmailAddress	
NumberOfMeasurements	
ImageArray	
HashTable	This will be an array; it will hold the key-value pairs which have been added to the hash table

Method	Description
SetupImageArray()	Inherited from its parent class, OutputControl.
AddPointToImageArray()	
SaveImage()	
ScanControl()	Function inherited from parent class, OutputControl is overwritten. This method will be almost identical to the function that it overwrites, however it will also call the ObjectControl() method, so that objects can be added to the image array.
HashingAlgorithm()	This method is used to get a hashed key from an inputted value. This will be to add and retrieve values from the hash table. The returned value will be an integer value.
GetValue()	This method will be used to retrieve a value from the HashTable variable. It will be inputted with a key, which will be hashed using the HashingAlgorithm() function. The hashed key will represent the position of the key-value pair within in HashTable array. If there is no value in that position, an error will be raised. Otherwise if there is a key-value in this position, if the inputted key is the same as the key in the array position, then the corresponding value is returned.
AddValue()	This method will be used to add a key-value pair to the HashTable variable. It will do this by firstly hashing the key by using the HashingAlgorithm function. Then it will append the key-value pair to the position of the hashed key in the HashTable array.
AddPresetWords()	This method will be for adding preset information to the HashTable. The information will be read from a text file, which will allow for additional information to be added or removed with ease. In the file will be sets of letter-code pairs, with each code representing how the letter is displayed in an image. These values are read and added to the hash table, with the

	letter acting as the key, and the code acting as the value.
SplitWord()	The SplitWord() method will take a word and the position of the word within the room as input parameters. For each letter within the word, the letter code will be retrieved from the hash table using the GetValue() method. Then each letter will be plotted on the ImageArray in order, with the centre of the word being the position of the word within the room which was inputted, this will be done by calling the AddLetterToImageArray() method.
AddLetterToImageArray()	This method will take two inputs as parameters, the letter code, and the position of the letter. It will then plot the letter using the letter code and the position of the letter, this can be done by calling the AddPointToImageArray() method where the pixel colour needs to be changed black.
ObjectControl()	The ObjectControl() method will be responsible for controlling the addition of objects onto the image array. It calls the functions required to add the objects to the array in the correct order.

The CalculationControl Class

Method	**Description**
CoordinateConversion()	This function will take an input of the distance measured from the LIDAR and how many points it has rotated. It will then convert the RotationPoint into degrees. Depending on whether the angle is less than 90, 180, ect, the variables will be assigned an XCoordinateSign, and YCoordinateSign, this will decide whether the point will be plotted in the X,Y/X,-Y/-X,Y,-X,-Y quadrant of the image. The function then converts the RotationPoint into radians. Then using trigonometry, the length of the point along the X and Y axis are calculated. Coordinates are then changed using XCoordinateSign/YCoordinateSign, to create coordinates in different quadrants. To convert these values to distances in a 500x500 array, they are multiplied by 25, and rounded to the nearest integer. Then the X/YCoordinatePoints are converted to points in the array by adding 250 to the value, as the midpoint of the array is 250,250. The XCoordinatePoint and YCoordinatePoint will then be returned.

The HardwareControl Class

Method	Description
GetDistance()	This method is used to control the LiDAR distance sensor, it will calculate the between itself and the object that it is facing and return the distance value in centimetres.
RotationControl()	This method will be used to control the stepper motor and also call the GetDistance() function at the appropriate times throughout the rotation so that the correct amount of measurements are take. The value returned from this function should be a list of length the same as the number of measurements taken, with a measurement taken in each position within the array.

The ShareControl Class

Method	Description
SendEmail()	This method will be used to share the completed scan image via an email. It will take input parameters of the email address the scan should be sent to, and the name of the scan image being sent. It will then email the completed scan image to that specific address.

Beginning the Program

This function will start the process of scanning the room. Firstly, it will retrieve the settings data inputted by the user on the user interface. It would then read these settings. If the corresponding value of the key “objectidentificationstatus” was on, the OutputControlWithObjectMarkings class would be initalised with the name Scan, and the corresponding values of the key's “filename”, “emailaddress” and “measurements” and inputted as the parameters for the initialisation, and the ScanControl function is called. Otherwise, the OutputControl class is initialised with the same name and same parameters, and the ScanControl function is called. This would allow for the program to start either a scan with or without objects added.

Record Structure

I will be saving the scans of the rooms as an image file. For this I will be converting a 500 by 500 array into a Png image file. This will create a monochrome 500 by 500 pixels image. As it is a black and white image, the colour depth is only 1 bit. This makes the resolution of the image 250000 (pixel width * pixel height).

This makes the size of the image file 250000 bytes (resolution * bit depth), which is 0.25 megabytes.

Descriptions of Data Structures

Hash Table

The “HashTable” class represented a hash table, which is stored in the “array” property.

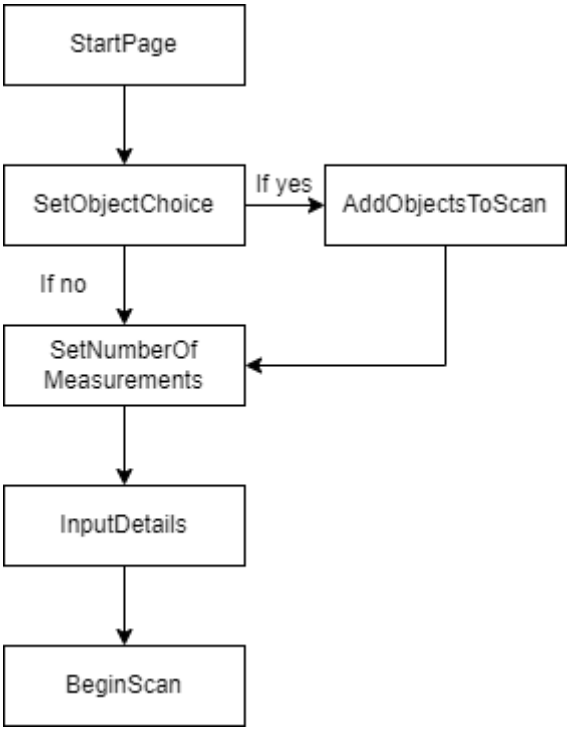
The pseudocode below represents the hash table functions should work. The entire of this pseudocode would be within the OutputControlWithObjects class. The key things to note about this pseudecode is firstly, the key-value pairs are stored in an array of length 50. The position of the key-value pair is calculated by hashing the key value, which will return a value between 0 and 49. This

value is the position in which the key-value pair is stored within the array. In the hashing algorithm, each value within the inputted key is converted into its integer ascii value, and each value is added and then modded by 50 to get a number between 0 and 49. I have also created pseudocode which will allow me to both add and retrieve data from the hash table, which are important functions for a hash table.

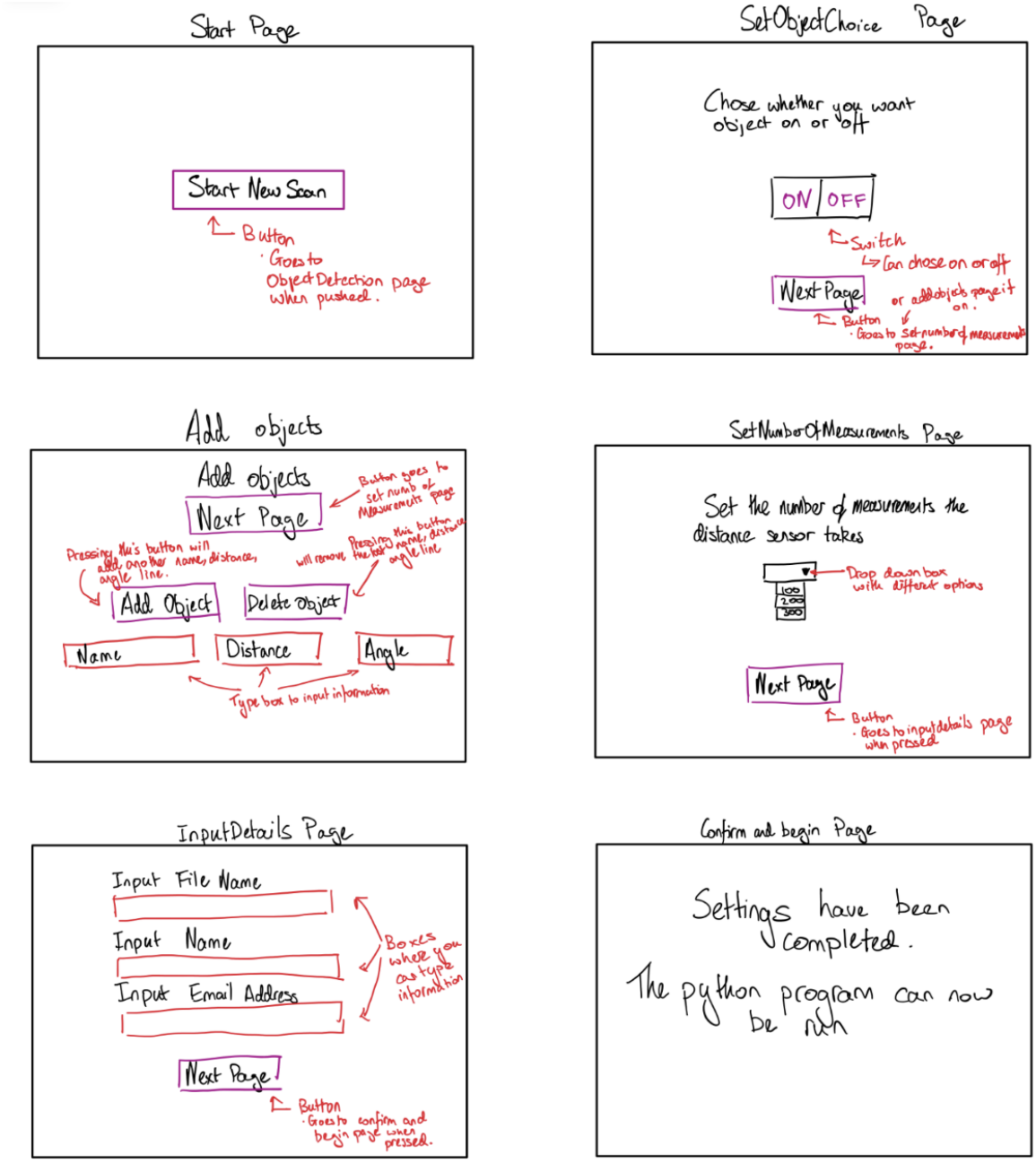
```
CLASS HashTable:
      # Initialiser
      FUNCTION constructor():
      self.array = [None] * 50

      # Methods
      FUNCTION HashingAlgorithm(key)
      VAR HashKey = 0
      VAR Key = list(Key)
      VAR KeyLength = len(Key)
      FOR counter=0 TO KeyLength
            VAR CalculationVariable = ord(Key[counter])
            VAR HashKey += CalculationVariable
      RETURN (HashKey%50)

      FUNCTION AddValue(Key, Value):
      VAR Position = self.HashingAlgorithm(Key)
      IF self.array[Position] IS NOT None
            FOR Count IN self.array[Position]
            if Count[0] == Key
                  Count[1] = Value
                  BREAK
            ELSE
                  self.array[Position].append([Key, Value)
      ELSE
            self.array[Position] = []
            self.array[Position].append([Key, Value])

      FUNCTION GetValue(Key):
            VAR Index = self.HashingAlgorithm(Key)
            IF self.array[Index] IS None
                  RAISE KeyError()
            ELSE
                  FOR Count IN self.array[Index]
                        IF Count[0] == Key
                              RETURN Count[1]
            RAISE KeyError()
```

Design of User Interface

The user interface of my product is particularly important, as it is how users of the product will interact with it. This means that it needs to be easy to understand and simple to use. I will be creating a website for my users to interface with, and therefore will be creating the interface using PHP.



This diagram shows how the flow of the different pages within the user interface will work. I have also created a diagram showing how all the different pages within the user interface should look. This is useful as it allows for me to easily design my interface off what I have sketched.



To allow for the data which has been inputted by my user to be read, I will have to create a way for my user interface and python program running on the raspberry pi to interface with each other.

I have created some simple pseudocode to show how this will be done:

In the user interface:

```
VAR settings_json = json_encode(array(
VAR firstname => SESSION_VAR[firstname],
VAR filename => SESSION_VAR[filename],
VAR emailaddress => SESSION_VAR[emailaddress],
VAR objectidentificationstatus => SESSION_VAR[objectidentification],
VAR measurements => SESSION_VAR[measurements]
));
```

```
VAR fp = fopen('data.php', 'a');
fwrite(VAR fp, VAR settings_json);
fclose(VAR fp);
```

In this pseudocode, I firstly create a json array out of all the data which has been inputted by a user. Then, I create a new website page called /data, where the json array will be written into. This will allow for a python program to read the specific website page, and therefore read the json array within it. I will do the same thing with the objects which have been inputted by the user.

For the python program, to read the information, I can do the following:

```
VAR settings =
urllib.request.urlopen('https://roomscanner.ml/scan/data').read()
VAR decodedsettings = jsonload(settings.decode("utf-8"))
```

This pseudocode will read the page, and then load the json, so that the json array can be read. Then I can do statements such as:

```
if VAR decodedsettings["objectidentificationstatus"] == "on":
```

This will allow me to use the settings inputted by my user in my python program, allowing for the program to work with the inputted settings.

Hardware Design and Selection

Because my project is not fully software based, and requires some robotics, I need to select specific components and hardware.

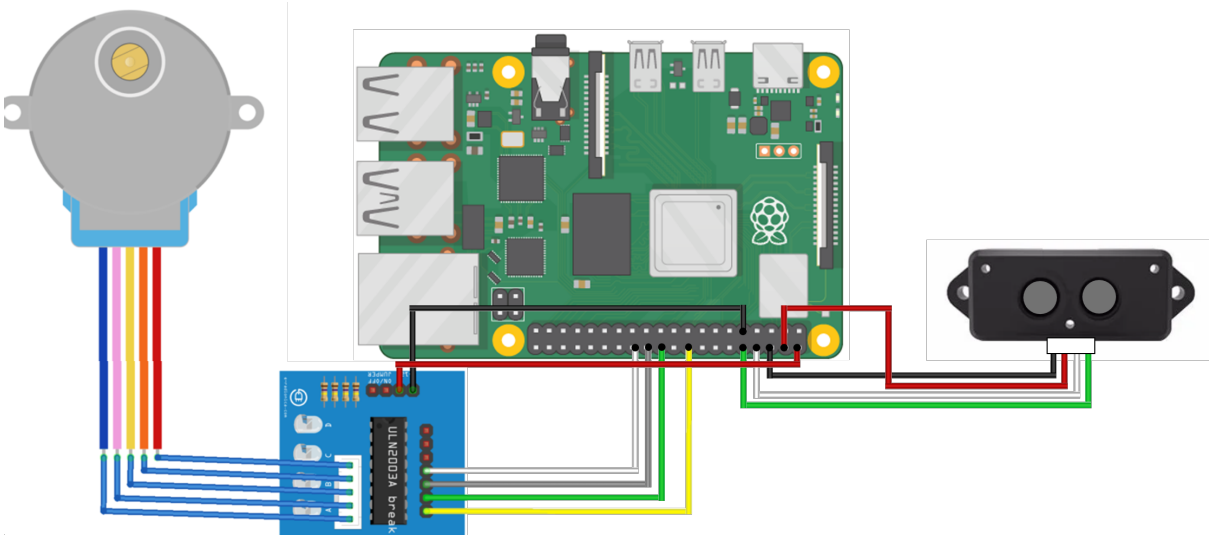
Components List

Component Name	Description of Component	Price of Component
Raspberry Pi 4 Model B 4GB	The Raspberry Pi is a small, single-board computer used often to learn programming skills, build hardware projects or do home automation, among other things. I will be using the Raspberry Pi to control the other components, using the python programming language.	£34
28BYJ-48 Stepper Motor & Driver Module Board	A stepper motor is a brushless DC electric motor, which divides a full rotation into several steps. The position of the motor can be controlled, and therefore can be made to move and hold at a specific step. I will be using the stepper motor to hold the LIDAR sensor and raspberry pi, and rotate around to get distance measurements.	Approx. £4
Benewake TFMini	The Benewake TFMini is a LIDAR sensor, previously mentioned in Further Investigation into types of Distance Sensors, I will be using it to measure the distance from my product to the wall it is facing.	£29.75
Jumper Wires	These wires will be used to connect all the components up to each other, which will allow them to get power and communicate with each other.	£2.00

The total price of the components required for my product is approximately £92, although some additional components such as jumper wires and cables are also required. This is a lot lower of a price than existing products.

Circuit Diagram

Using an application called Fritzing, I created a circuit diagram of my product, this is useful as it depicts the physical arrangement of the wires and the components which they are connected to. This allows the components to be assembled easily by those who do not understand how the circuit works, as they are able to follow the picture.



Raspberry Pi Pin	**Connection to:**
Pin 2 (5V Power)	Stepper Motor Driver
Pin 4 (5V Power)	LiDAR sensor
Pin 6 (Ground)	LiDAR sensor
Pin 8 (GPIO 14)	LiDAR sensor
Pin 9 (Ground)	Stepper Motor Driver
Pin 10 (GPIO 15)	LiDAR sensor
Pin 18 (GPIO 24)	Stepper Motor Driver
Pin 22 (GPIO 25)	Stepper Motor Driver
Pin 24 (GPIO 8)	Stepper Motor Driver
Pin 26 (GPIO 7)	Stepper Motor Driver

System Security and Integrity of Data

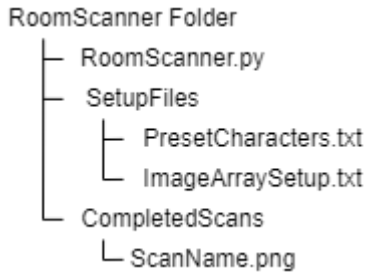
As the program will store no private or confidential data, encryption is therefore unnecessary.

When the user inputs settings, the data will undergo a commonly used type-validation scheme. This will ensure that the data inputted is valid. Most of the data input will be done via drop down boxes, or switches, such as number of measurements take, and whether objects should be added to the scan or not. Because I will be using forms in html to get my information, I am also able to specify some of the information which needs to be submitted, this can be done by using <email> etc, which means that the user must input an email address.
In addition as I will be loading information from word documents, I need to ensure that the data files follow a valid schema, meaning that it can be loaded correctly without an errors.

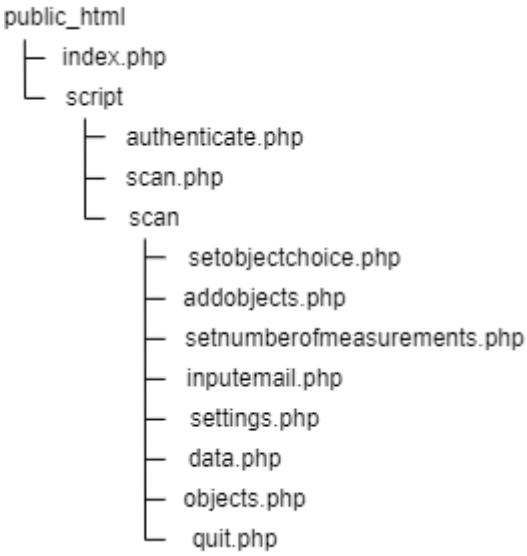
Technical Solution

File Organisation and Structure

For ease of use, I have structured the files as followed, to allow for a user to be able to find scans easily should they have lost them. In addition to this, they have been arranged in this way so that files are arranged for direct access by the program, with the files that the program will need to compete a scan in the SetupFiles folder, and all completed scans being saved in the CompletedScans folder, which also allows for direct access to the files when the program is emailing the completed scan to the user.



In addition to the python program, which runs on a Raspberry Pi, I have also created the user interface, which runs on a website, www.roomscanner.ml. The files containing the code is arranged as followed so that it is simple to understand how it works and easier to search for a specific peace of code.



The Environment

```
import numpy as np
import math
from PIL import Image
import serial
import time
import RPi.GPIO as GPIO
import urllib.request
import json
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.image import MIMEimage
```

These lines import the relevant modules and libraries of code which are used to allow the program to run.

Name of Library	Description
NumPy	NumPy (Numerical Python) is a Python library used with arrays. It also used when working with linear algebra, Fourier transform and matrices. It is used as typically in python, lists are used to serve the role of arrays, however, are slow to process. Because NumPy stores arrays in one continuous place in memory, processes can access and manipulate them very efficiently. I will be using NumPy to create and manipulate an array holding the pixel data of an image.
Math	The math module is a built-in module used for mathematical tasks, it has a set of methods (such as finding the log of numbers or converting a number into radians) as well as a set of constants (which are often used number such as pi or e). I will be using the math module for its trigonometry methods, allowing me to find the angles and lengths of sides of triangles.
PIL	PIL (also known as the Python Image Library) is an image processing library. It is often used as it has extensive file format support, an efficient internal representation and has powerful image processing capabilities. It can be used to add filters to images, blurring images, cropping, etc. I will be using PIL to convert the NumPy array into an image file.
Serial	The Serial library gives control of the serial port, allowing serial data to be read. I will be using this library to control the LIDAR sensor and to read the distances it outputs.
Time	The time library supplies multiple time-related functions, such as getting an exact time. I will be using the time library to wait whilst distance measurements are taken using the LIDAR sensor, before the stepper motor rotates.
RPI.GPIO	This library allows control of the GPIO on a Raspberry Pi. I will be using it to control the stepper motor, allowing it to rotate around.
Urllib.requests	Urllib is a package that contains several modules which help when working with different URLs, I am using the urllib.requests library, which is used for both opening and reading URLs, which will allow me to see what scan settings users have chosen.
Json	Json is a package which is built into python, it is used for encoding and decoding JSON data (JavaScript Object Notation), I will be using the json package to decode JSON data arrays, which contain information about the scan settings which users have chosen.
smtplib	The smtplib module defines an SMTP client session object, this can then be used to send emails to any internet machine with an SMTP or ESMTP listener daemon. I will be using this module to help send emails containing the scanned room.
Email.mime.multipart and email.mime.image	This is a library which allows to me read, write and send email messages. I will be using this module to send MIME (Multipurpose Internet Mail Extensions) messages, containing an image (png) file to product users.

The OutputControl Class

Intialisation

```
    def __init__(self, FileName, EmailAddress, NumberOfMeasurements):
        self.ImageArray = np.full((500,500), 255, np.uint8)
        self.FileName = FileName
        self.EmailAddress = EmailAddress
        self.NumberOfMeasurements = int(NumberOfMeasurements)
```

There are 3 parameters which are passed into this method when the class is initialised, FileName which is what the user wants to call the final output of the scan, and is a string, EmailAddress, the email address the user wants the scan data to be sent to once the scan has been completed, it is also a string, and the NumberOfMeasurements, which is the number of distance measurements the user wants the device to take in once scan, this is an integer.

In addition, the attribute self.ImageArray is a 2-dimensional array, which can be thought of as a grid of 500 by 500 boxes, with each box holding the data on the colour a specific pixel of an image. Here I am initialising, the array, and filling every position in the array with 255, which represents the colour white when the array is converted into an image file. Here I am using np.full as it allows for the array to be generated very quickly and efficiently.

Changing the colour of a point in the Image Array

```
    def AddPointToImageArray(self, XCoordinate, YCoordinate):
        self.ImageArray[XCoordinate, YCoordinate] = np.uint(0)
```

This function takes 2 parameters, the X and Y coordinates of a point in the array. It then changes the value of this position within the ImageArray variable to the unsigned integer 0. When the ImageArray is converted to an image file, these points will be black.

Setting up the Image Array for a Scan

```
    def SetupImageArray(self):
        with open('SetupSettings/SetupImage.txt') as LetterFile:
            Content = LetterFile.readlines()
            NumberOfLines = (len(Content))
            for Count in range(NumberOfLines):
                Line = list(Content[Count])
                Line.pop()
                CommaFound = False
                LineLength = len(Line)
                while CommaFound == False:
                    for i in range(LineLength):
                        if Line[i] == ",":
                            CommaPosition = i
                            CommaFound = True
                XCoordinate = [0] * (CommaPosition)
                for i in range (CommaPosition):
                    XCoordinate[i] = Line[i]
                XCoordinate = int("".join(XCoordinate))
                YCoordinate = [0] * (LineLength - CommaPosition-1)
                for i in range (CommaPosition):
                    YCoordinate[i] = Line[i+CommaPosition+1]
                YCoordinate = int("".join(YCoordinate))
                self.AddPointToImageArray(XCoordinate, YCoordinate)
```

The purpose of this function is to add basic elements to the image array, such as a scale and a centre point, showing where the scanner is in relation to the scan. Firstly the SetupImage.txt file is opened, and the contents is copied into the Content variable. The text file is of the format of:

```
247 247
246 247
247 253
246 253
248 248
247 248
248 252
247 252
249 249
248 249
249 251
248 251
250 250
249 250
250 250
249 250
251 251
250 251
251 249
250 249
252 252
251 252
252 248
251 248
253 253
252 253
253 247
252 247
10 22
10 23
22 10
23 10
```

Each line contains the X coordinate and Y coordinate of a point in the array, in the format XCoordinate,YCoordinate. For every line in the text file, it is converted into an array and the last element is popped, this removed the hidden character "/n" which starts a new line. The while statement is then used to find the position of the comma within the array, which allows for the integers before and after the comma to be split, this creates two new arrays, XCoordinate and YCoordinate. These arrays are then converted into an integer, by first changing them into a list using the built in .join() method. The AddPointToArray function of the OutputControl class is then called, with the XCoordinate and YCoordinate as the two parameters. The for loop means that this occurs for every line of text within the SetupImage text file.

Saving a Completed Scan Image

```
    def SaveImage(self):
        im = Image.fromarray(self.ImageArray)
        ImageLocation = "CompletedScans/"+self.FileName+".png"
        im.save(ImageLocation)
```

This function converts ImageArray into a png image file, using the Pillow Image library. The location in which the image should be saved in is then defined, with it being saved in the CompletedScans folder, under the name created by the user. The image is then saved under this name using the Pillow Image Library.

Sharing a Scan

```
    def ShareImage(self):
        ShareScan.sendmail(self, self.EmailAddress, self.FileName)
```

This function calls the function sendmail, which is in the ShareScan class, with the parameters self.EmailAddress and self.FileName. In the future, should I decide to add different methods for

sharing the image, I could simply add an if statement here, saying if ScanMethod = sendmail, then ShareScan.sendmail(), which would allow me to implement other sharing methods incredibly easily.

Scan Control

```
    def ScanControl(self):
        self.SetupImageArray()
        DistanceArray = HardwareControl.BeginRotation(2048, self.NumberOfMeasurements)
        for i in range(self.NumberOfMeasurements):
            Angle = (360/self.NumberOfMeasurements)*i
            self.CoordinatePoint = CalculationControl.CoordinateConversion(DistanceArray[i], Angle)
            self.AddPointToImageArray(self.CoordinatePoint[0], self.CoordinatePoint[1])
        self.SaveImage()
        self.ShareImage()
```

The ScanControl function is responsible for calling other functions in order, so that a scan is created. It firstly calls the SetupImageArray() function, which is in the same class as it, this sets up the template for the scale, centre point, etc for the final image. Then the function begins the scanning process by calling the BeginRotation() function, which is in the HardwareControl class, with the parameters of 2048 and NumberOfMeasurements. 2048 is the number of rotations that the stepper motor does in 1 full rotation, and it is inputted so that the product knows it is doing a full spin, instead of half a spin. The returned array is stored in the DistanceArray attribute. A for loop is then used to calculate the coordinates of the position and add it to the ImageArray for every measurement in the DistanceArray variable. The SaveImage() function is then called, which saves the image file, then the ShareImage function is then called, which will email the scan image file to the user.

The OutputControlWithObjectMarkings Class

Intilisation

```
    def __init__(self, FileName, EmailAddress, NumberOfMeasurements):
        self.array = [None] * 50
        OutputControl.__init__(self, FileName, EmailAddress, NumberOfMeasurements)
```

In the initialisation of the OutputControlWithObjectMarkings, the same three parameters as the OutputControl are called for the initialisation, and the attributes of the OutputControl are initialised under this class as it inherits functions and attributes from the OutputControl class. In addition to this, the self.array attribute is created, this is a 1-dimensional array of length 50, with each value in it being None. This will hold the data for the hash table functions used in the class.

Hashing Algorithm

```
    def HashingAlgorithm(self, Key):
        HashKey = 0
        Key = str(Key)
        Key = list(Key)
        KeyLength = len(Key)
        for i in range(KeyLength):
            CalculationVariable = ord(Key[i])
            HashKey = HashKey + CalculationVariable
        return (HashKey) % 50
```

This function is the hashing algorithm of the hash table. It takes a single input, of the key which is being inputted. The key is then converted into an array, and then every element in the array is converted into its corresponding Unicode value, and all each elements Unicode value is added together. This is done using a for loop, with the range of the length of the key array. The hash key value is then calculated by taking modulus 50 of it, meaning that the hash key will be a number between 0 and 49, which will correspond to a position in the hash table array. This value is then returned. The HashingAlgorithm, AddValue and GetValue functions for the basis of the main hash table algorithm for this program.

Add a Value to the Hash Table

```
def AddValue(self, Key, Value):
    Position = self.HashingAlgorithm(Key)
    if self.array[Position] is not None:
        for Count in self.array[Position]:
            if Count[0] == Key:
                Count[1] = Value
                break
        else:
            self.array[Position].append([Key, Value])
    else:
        self.array[Position] = []
        self.array[Position].append([Key, Value])
```

This function is used to add a key-value pair to the hash table. First, the hashing algorithm is completed on the key, and the value is stored in the Position variable. In this hash table, the value of the hashed key corresponded to the position of the key-value pair within the array storing the data. Using an if statement, I check whether the position defined by the hashed key is empty, if it is empty, the key-value pair is appended into that position. If it is not empty, the key-value pair currently in the position is replaced by the new key-value pair.

Retrieving a Value from the Hash Table

```
def GetValue(self, Key):
    index = self.HashingAlgorithm(Key)
    if self.array[index] is None:
        raise KeyError()
    else:
        for Count in self.array[index]:
            if Count[0] == Key:
                return Count[1]
        raise KeyError()
```

The GetValue function is used to get the value of a specific key-value pair. The function takes a single input, of the key. The HashingAlgorithm function is then called, with the parameter of the key. This allows for the function to get the hashed value of the key. Since the hashing algorithm stores the key-value pair in the position of the hashed key value, I then use an if statement to check whether that position within the array is empty, if it is, I raised a KeyError, this is part of the error handling for this function. If the value is not empty, the function then gets the key-value pair in the hashed key values position. The function then checks whether the stored key is the same as the key which has been passed into the function, if it is, then the corresponding value is returned, otherwise a KeyError is returned.

Adding Pre-set Values to the Hash Table

```
def AddPresetCharacters(self):
    with open('SetupSettings/Letters.txt') as LetterFile:
        Content = LetterFile.readlines()
        NumberOfLines = (len(Content))
        if NumberOfLines % 2 != 0:
            print("Error, the letter:letter code ratio is not correct")
        else:
            LineNumber = 0
            for Count in range(int(NumberOfLines / 2)):
                Letter = list(Content[LineNumber])
                Content2 = list(Content[LineNumber+1])
                Content2.pop()
                self.AddValue(Letter[0], ("".join(Content2)))
                LineNumber = LineNumber + 2
```

In this function, firstly the text file Letters.txt is opened and read. The file is in the following format:

```
a
00100010100111001010010 10
b
01100010100110001010011 00
c
01110010000100001000011 10
d
01100010100101001010011 00
e
01110010000111001000011 10
f
01110010000111001000010 00
g
01110010000101001010011 10
h
01010010100111001010010 10
i
01110001000010000100011 10
j
01110001000010000100011 00
k
01010010100110001010010 10
l
01000010000100001000011 10
m
11111101011010110101101 01
n
01001011010111101011010 01
o
00100010100101001010001 00
p
01110010100111001000010 00
q
01110010100111000011000 10
r
01100010100110001010010 10
```

With the letter, and the corresponding code for the letter below it. The number of lines within the text file is then checked, this is part of the error handling for my program. If there is not an even number of lines, it means that there must be a problem with the text file, as it means there is a mismatch between the number of letters are number of digit codes for the numbers. If this occurs, an error is printed. Otherwise, each line of the text file is turned into an array, this is needed as every line in the text file has a hidden character of "/n" at the end, which tells the file to go to the next array. To get the letter, the function gets the first character in the even numbered lines of the text file. To get the image code, the last element of every odd numbered line is removed using the pop function. The AddValue function is then called, with the letter and the letter code as parameters. I have used a text file, as it allows me to easily add other characters and their corresponding code incredibly easily.

Splitting a Word into Individual Characters

```
    def SplitWord(self, Word, WordMidpoint):
        WordLength = len(Word)
        Word = list(Word)
        Coordinate = [WordMidpoint[0], (WordMidpoint[1]-(round(WordLength*2.5)))]
        for Count in range(WordLength):
            LetterCode = self.GetValue(Word[Count])
            self.AddLetter(LetterCode, Coordinate)
            Coordinate[0] = int(Coordinate[0]) + 5
```

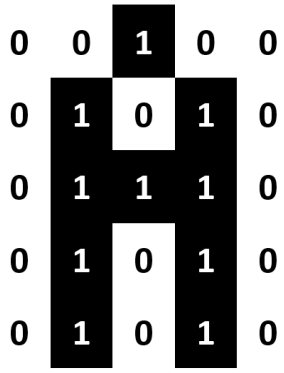
The SplitWord function takes 2 input parameters, the word being split, Word, and the position of the centre of the word, the WordMidpoint. Word is stored as a string to begin with, and the WordMidpoint is an array of the form [YCoordinate, XCoordinate]. The length of the word is calculated using the len() function, and the word is converted in an array using the list() function.

The position of the first letter in the word is then calculated, as it has the same YCoordinate and an XCoordinate of the length of the word * length of the image representation of a letter * 0.5. Using a for loop, the letters are then added to the image. This is done by calling the GetValue function and storing the returned value as LetterCode. The AddLetter function is then called with the parameters LetterCode and Coordinate.

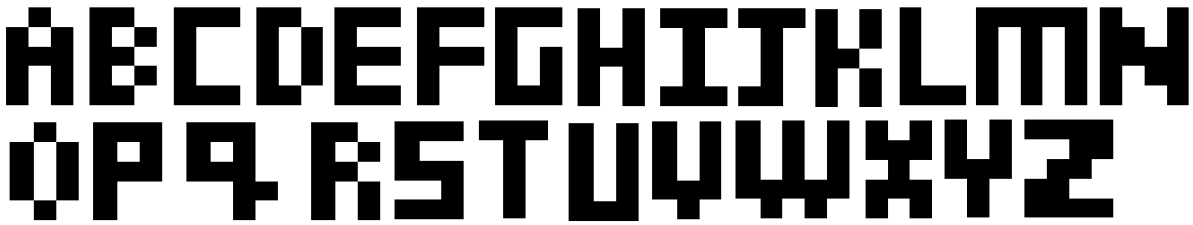
Add a Letter to the Image

```
    def AddLetter(self, LetterCode, StartingPosition):
        LetterCode = str(LetterCode)
        LetterCode = list(LetterCode)
        RowCount = 0
        for Count in range(25):
            if LetterCode[Count] == "1":
                LetterPosition = [(Count % 5), RowCount]
                LetterPosition = [(LetterPosition[1] + StartingPosition[1]),(LetterPosition[0] +
StartingPosition[0])]
                self.AddPointToImageArray(LetterPosition[0], LetterPosition[1])
            if Count % 5 == 0:
                RowCount += 1
```

The AddLetter function takes two parameters, the LetterCode, an integer, and the StartingPosition, an array. The LetterCode is then converted into a list. The LetterCode is a 25 digit number, of either 1 and 0, such as 0010001010011100101001010, it represents a 5x5 array, with the first 5 digits being the 5 numbers in the first row, and the second 5 digits being the 5 numbers in the second row, etc, of the colours black and white. If the value is 1, then the position should be coloured black. The letter code mentioned before represents the letter A and looks as followed.



With all the letters looking like the following:



This allows for all the letters in the alphabet to be represented in image form. Using a for loop, with a range of the length of the array (25), each value is checked to see if it is 1, if it is 1, it is added the Image Array at calculated position, depending on its position within the array, using the

AddPointToImageArray() function. The position can be calculated by modulus the position of the current value being checked in the array by 5 and putting it in that position, and increasing the row count by one if it is divisible by 5, so that the value after will be on the next line.

Object Control

```
    def ObjectControl(self, Word, Angle, Distance):
        MidPoint = CalculationControl.CoordinateConversion(Distance, Angle)
        self.SplitWord(Word, MidPoint)
```

In this function, two parameters three parameters are taken as input, the word to be added to the ImageArray, and the angle and distance from the LiDAR sensor at which the word should be plotted. The point at which the centre of the word should be is then calculated by calling the CoordinateConversion function in the CalculationControl class, giving the parameters of the Distance and the Angle. The returned value is an array, stored under the variable Midpoint. The SplitWord function is then called, with the parameters of the word being added, and the midpoint of the the word.

Scan Control

```
    def ScanControl(self):
        self.SetupImageArray()
        self.AddPresetCharacters()
        objects = urllib.request.urlopen('https://roomscanner.ml/scan/objects').read()
        obj_decode = json.loads(objects.decode("utf-8"))
        for object, distance_from_scanner, angle_from_scanner in
zip(obj_decode["list_of_object_names"],obj_decode["list_of_distances_from_scanner"],obj_decode["list_of_a
ngles_from_scanner"]):
            angle = int(f"{angle_from_scanner}")
            object = (f"{object}")
            distance = int(f"{distance_from_scanner}")
            self.ObjectControl(object.lower(), angle, distance*100)
        DistanceArray = HardwareControl.BeginRotation(2048, self.NumberOfMeasurements)
        for i in range(self.NumberOfMeasurements):
            Angle = (360/self.NumberOfMeasurements)*i
            self.CoordinatePoint = CalculationControl.CoordinateConversion(DistanceArray[i], Angle)
            self.AddPointToImageArray(self.CoordinatePoint[0], self.CoordinatePoint[1])
        self.SaveImage()
        self.ShareImage()
```

This function is a function inherited from its parent class, OutputControl, but is overwritten. It begins the same, by calling the SetupImageArray() function, which is inherited from the parent class. Then it calls the AddPresentCharacters() function. It then reads the "https://roomscanner.ml/scan/objects" page of the website, which stores all of objects listed by the user when they are completing the settings. This is done use the urllib.request module. The contents of this page is a json array, which looks like the following:

{"list_of_object_names":["Table","Chair","Mat"],"list_of_distances_from_scanner":["5","3","9"],"list_of_angles_from_scanner":["20","350","354"]}

Using the json.load() function, it then takes the contents of the page and returns it as a json object, meaning that the data is in the form of a key/value pair. For every set of object, distance and angle, the ObjectControl function is called, with the lowercase of the object name, the angle, and the distance in centimetres inputted as parameters. This will add the objects to the ImageArray. Then the function begins the scanning process by calling the BeginRotation() function, which is in the HardwareControl class, with the parameters of 2048 and NumberOfMeasurements. 2048 is the number of rotations that the stepper motor does in 1 full rotation, and it is inputted so that the product knows it is doing a full spin, instead of half a spin. The returned array is stored in the DistanceArray attribute. A for loop is then used to calculate the coordinates of the position and add it to the ImageArray for every measurement in the DistanceArray variable. The SaveImage() function

is then called, which saves the image file, then the ShareImage function is then called, which will email the scan image file to the user.

The HardwareControl Class

Getting a Distance Measurement

```
    def GetDistance():
        ser = serial.Serial("/dev/serial0", 115200) # Lidar serial
        if ser.is_open == False:
            ser.open()
        count = ser.in_waiting
        if count > 8:
            recv = ser.read(9)
            ser.reset_input_buffer()
            if recv[0] == 0x59 and recv[1] == 0x59:  # 0x59 is 'Y'
                distance = recv[2] + recv[3] * 256
                strength = recv[4] + recv[5] * 256
                distance = distance + 10
                if distance == None:
                    distance = HardwareControl.GetDistance()
                return distance
```

Firstly, the serial port being used for the lidar sensor is defined using the serial module. If the serial port is closed, it is then opened using the same module, this is done using an if statement. The attribute count is then turned into the number of bytes in the input buffer, this is an integer value. If the number of bytes in the input buffer is above 8, then it reads 9 bytes from the serial port and puts the values of this into the recv variable. Then the contents of the input buffer is flushed, discarding all of its contents. This is done using the serial module. The distance is then taken, and the strength is also taken from the recv variable array. I have added the ability to take the strength of the measurement so that in the future I can add additional details to the completed scan information should the user want it. I then add 10 to the distance measurement as the value of the distance is 0cm at 10cm, due to the minimum distance for the LiDAR sensor being 10 cm. As error handling for this distance measurement, I then use an if statement to check whether the measured distance is “None” which occasionally happens when the there is a problem, this is mentioned in my testing phase. If the statement is true, it recursively calls the function to get a new distance value. The distance value is then returned once it isn’t “None”.

Rotation Control

```
    def BeginRotation(StepsPerRevolution, MeasurementCount):
        GPIO.setmode(GPIO.BCM)
        Pins = [24, 25, 8, 7]
        for PinCount in Pins:
            GPIO.setup(PinCount, GPIO.OUT)
            GPIO.output(PinCount, False)
        StepCount = 4
        StepperSequence = []
        StepperSequence = [i for i in range(0, StepCount)]
        StepperSequence[0] = [1, 0, 0, 0]
        StepperSequence[1] = [0, 1, 0, 0]
        StepperSequence[2] = [0, 0, 1, 0]
        StepperSequence[3] = [0, 0, 0, 1]
        StepCounter = 0
        NumOfSteps = 0
        MeasurementCount = int(MeasurementCount)
        DistanceMeasurements = [0] * MeasurementCount
        DistanceMeasurementsCount = 0
        if StepsPerRevolution < 0:
            sign = -1
        else:
            sign = 1
        StepsPerRevolution = sign * StepsPerRevolution
        ModulusCalculation = StepsPerRevolution / MeasurementCount
        for i in range(StepsPerRevolution):
            for pin in range(4):
                xpin = Pins[pin]
                if StepperSequence[StepCounter][pin] != 0:
                    GPIO.output(xpin, True)
                else:
                    GPIO.output(xpin, False)
            StepCounter += sign
            if NumOfSteps % ModulusCalculation == 0:
                time.sleep(0.01)
                DistanceMeasurements[DistanceMeasurementsCount] = HardwareControl.GetDistance()
                DistanceMeasurementsCount += 1
            NumOfSteps += 1
        return DistanceMeasurements
```

In this function, the pins used by the stepper motor and stepper motor driver are set up as output pins, using the GPIO.setup() function, I also set the output of all of the pins to false, as they are not in use and therefore should not be accidentally rotating. I then create a simple sequence for the stepper motor to follow, using an array. To create the array, I have used list comprehension using the StepCount parameter to ensure that the array is of the correct dimensions. I have used list comprehension as it allows for the array to be generated incredibly quickly and efficiently. I also create a second array named DistanceMeasurements, which a one-dimensional array, with a length equal to the number of distance measurements which the user wants the device to take in a single scan. I have added the ability for more complicated stepper motor sequences to be added, in addition to the stepper motor being able to rotate in both directions by changing the sign of the number of StepsPerRevolution to a positive to minus, this allows for additional settings to be added in the future very quickly and easily should a user want to double back whilst making a scan to ensure that the distance measurements are correct. To find the steps on the stepper motor where I need to take a distance measurement, I use the ModulusCalculation attribute, which is calculated by StepsPerRevolution/MeasurementCount, this gives an integer value. This value allows me to check whether I need to take a distance measurement, by using an if statement, as if the NumOfSteps which the stepper has done so far in one rotation % ModulusCalculation is 0, this means that a measurement should be taken, and the GetDistance function is called. The rotation of the stepper motor is controlled by the StepperSequence array, and an if statement, within a for loop, where if the value of StepperSequence in the position [StepperCount, pin] is 1, the GPIO.output will be set to true, causing the stepper motor to rotate. The DistanceMeasurements array which contains all the distance measurements taken during the scan is then returned.

The CalculationControl Class

Calculating the Coordinate Points

```
def CoordinateConversion(LidarDistance, Angle):
    LidarDistance = int(LidarDistance)
    LidarDistance = LidarDistance / 100
    if 0 < Angle < 90:
        XCoordinateSign = 0
        YCoordinateSign = 1
    elif 90 < Angle < 180:
        Angle = Angle - 90
        Angle = 90 - Angle
        XCoordinateSign = 0
        YCoordinateSign = 0
    elif 180 < Angle < 270:
        Angle = Angle - 180
        XCoordinateSign = 1
        YCoordinateSign = 0
    elif 270 < Angle < 360:
        Angle = Angle - 270
        Angle = 90 - Angle
        XCoordinateSign = 1
        YCoordinateSign = 1

    RotationRadian = ((Angle) * math.pi) / 180
    XCoordinatePoint = LidarDistance * math.sin(RotationRadian)
    YCoordinatePoint = LidarDistance * math.cos(RotationRadian)

    if Angle == 0:
        XCoordinatePoint = 0
        YCoordinatePoint = LidarDistance
        XCoordinateSign = 1
        YCoordinateSign = 1
    if Angle == 90:
        XCoordinatePoint = LidarDistance
        YCoordinatePoint = 0
        XCoordinateSign = 0
        YCoordinateSign = 1
    if Angle == 180:
        XCoordinatePoint = 0
        YCoordinatePoint = LidarDistance
        XCoordinateSign = 1
        YCoordinateSign = 0
    if Angle == 270:
        XCoordinatePoint = LidarDistance
        YCoordinatePoint = 0
        XCoordinateSign = 1
        YCoordinateSign = 1

    if XCoordinateSign == 0:
        XCoordinatePoint = XCoordinatePoint * -1
    if YCoordinateSign == 0:
        YCoordinatePoint = YCoordinatePoint * -1

    XCoordinatePoint = XCoordinatePoint * 25
    XCoordinatePoint = round(XCoordinatePoint)
    YCoordinatePoint = YCoordinatePoint * 25
    YCoordinatePoint = round(YCoordinatePoint)

    XCoordinatePoint = XCoordinatePoint + 250
    YCoordinatePoint = YCoordinatePoint + 250
    return [XCoordinatePoint, YCoordinatePoint]
```

The CoordinateCalculation function is used to calculate coordinate points which can be plotted on an array from a distance and an angle, it uses basic trigonometry for this. It takes two parameters as input, which are both integers, the LiDAR distance, which is the measured distance from the LiDAR sensor to the wall in centimetres, and the angle with which the LiDAR sensor is facing currently compared to its initial position. The distance is then converted into metres by dividing it by 100. The values are then converted from a polar coordinate graph, where a direction and magnitude is used to describe a position, into an X-Y graph, where two coordinate points used to describe a position, by firstly finding where in an X-Y graph a point would be, if the angle was between 0 and 90, it would

be in the positive X, positive Y area, 90 to 180, would be in the negative X, positive Y area, etc. The distance value is assigned an XCoordinateSign and YCoordinateSign, which is 1 if they are in positive X/Y area, and negative if in the negative X/Y area. Pythagoras is then used to find the X component and Y component of the distance. The signs are distance components are then multiplied by their sign to make them positive or negative, and multiplied by 25, so that they are using the same scale as the ArrayImage. They are then rounded to the nearest integer. Since the centre point of ImageArray is 250,250, 250 is added to each coordinate point, and the values are returned in the form of an array of [XCoordinate, YCoordinate]

The ShareScan class

The ShareScan class contains the functions which can share the completed image scan with the person who made the scan. I have created a separate class for the function so that in the future I can add other methods of communication, such as sending a text message with the scan image or using a social media site such as Snapchat or Instagram, which will allow for more choice for the user on how they want their scan to work.

Sending an Email

```
    def sendmail(self,recipient, image):
        SMTPSERVER = 'smtp.gmail.com'
        SMTPPORT = 587
        SenderEmailAddress = 'roomscanner.ml@gmail.com'
        SenderEmailPassword = 'password'

        emailData = MIMEMultipart()
        emailData['Subject'] = "Your Room Scan has been successfully completed!"
        emailData['To'] = recipient
        emailData['From'] = SenderEmailAddress

        imageData = MIMEImage(open(("CompletedScans/"+image+".png"), 'rb').read(), 'jpg')
        imageData.add_header('Content-Disposition', 'attachment; filename="ScannedRoom.jpg"')
        emailData.attach(imageData)

        session = smtplib.SMTP(SMTPSERVER, SMTPPORT)
        session.ehlo()
        session.starttls()
        session.ehlo()

        session.login(SenderEmailAddress, SenderEmailPassword)

        session.sendmail(SenderEmailAddress, recipient, emailData.as_string())
        session.quit
```

The sendmail function takes an input of two parameters, the recipient email address, a string stored under the recipient variable, and the name of the image file, also a string, stored under the image variable. The server, port, and SenderEmailAddress and password are then defined, this will allow us to send an email from the SenderEmailAddress. Using the MIMEMultipart module, the contents of the email is then defined, and is stored as emailData, and the imageData is then defined and attached to the emailData. Then, using the smtplib module, I create a session, and log into the email using the SenderEmailAddress and SenderEmailPassword, and send the use the sendmail() function with SenderEmailAddress, recipient and emailData.as_string() as parameters, before quitting the session.

The BeginProcess Function

```
def BeginProcess():
    settings = urllib.request.urlopen('https://roomscanner.ml/scan/data').read()
    settings_decode = json.loads(settings.decode("utf-8"))
    if settings_decode["objectidentificationstatus"] == "on":
        Scan = OutputControlWithObjectMarkings(settings_decode["filename"],
settings_decode["emailaddress"], settings_decode["measurements"])
        Scan.ScanControl()
    else:
        Scan = OutputControl(settings_decode["filename"], settings_decode["emailaddress"],
settings_decode["measurements"])
        Scan.ScanControl()
```

This function starts the process of scanning the room. Firstly, it retrieves the settings data which has been submitted by the user on the website interface. This is done by reading the opening and reading the "https://roomscanner.ml/scan/data" page, which contains a json array of the chosen settings. This is in the following form:

{"firstname":"TestName","filename":"TestFile","emailaddress":"TestEmail@gmail.com","objectidentificationstatus":"on","measurements":"2048"}

Using the json.load() function, it then takes the contents of the page and returns it as a json object, meaning that the data is in the form of a key/value pair.

An if statement checks if the corresponding value of the key "objectidentificationstatus" is either "on" or "off", should it be on, the OutputControlWithObjectMarkings class is initalised with the name Scan, and the corresponding values of the key's "filename", "emailaddress" and "measurements" and inputted as the parameters for the initialisation, and the ScanControl function is called. Otherwise, the OutputControl class is initialised with the same name and same parameters, and the ScanControl function is called.

The User Interface

Index.php

```php
<?php
session_name("WEBCONSOLE");
session_start();
$_SESSION['route'] = "setobjectchoice";
header("Location: /scan/setobjectchoice");
?>
```

This php file is used to handle the session, this makes the data inputted by the user available across multiple different pages of the website. Firstly, should there be a previous session, I retrieve the session, if there isn't a previous session I name the new session "WEBCONSOLE", this is done using the session_name() function. I then use the session_start() function to begin or resume the session. A session variable, named route is then defined, with it being a string. The value "setobjectchoice" is then defined as the variable, this will describe the current setting which the user is doing, so that they are able to restart the session at their previous position. The header() function then redirects the user to the start page.

Authenticate.php

```php
<?php
$application_password = "1234";  // Password to use for web application
session_name("WEBCONSOLE");
session_start();
if($_SESSION['authenticated'] == "true"){
  header("Location: /");
}else{
  if($_POST['password']){
    if($_POST['password']  === $application_password){
      $_SESSION['authenticated'] = "true";
      header("Location: /");
    }else{
      echo "<p>PASSWORD INCORRECT</p>";
    }
  }
}
?>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<p>You must be authenticated</p>
<p>Please enter the application password</p>
<p>Your IP is <?php echo $_SERVER['REMOTE_ADDR'];?></p>
<form action="/authenticate" method="post">
  <input type="password" name="password">
  <input type="submit">
</form>
```

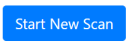
This website page is used to check that only users who are meant to be using the application are using it. It looks as follows:

The page states your IP address and asks for a password to be submitted. It is submitted through a simple form with two parts, one part for typing a password, the input type is a password, meaning it doesn't reveal what you have typed, the second part is a submit button. Once the submit button has been pressed, the password is compared to the application_password using an if statement. If it is the same, the session variable 'authenticated' is changed to true, and the user is forwarded to the main page of the scan. Meaning that the user does not need to log in again in their current session. If the password is incorrect, "PASSWORD INCORRECT" is printed, using the echo() function at the top of the page, and the user can resubmit the password. The password is not in the html part of the program, and therefore is not publicly available should the user try and inspect element to find the password.

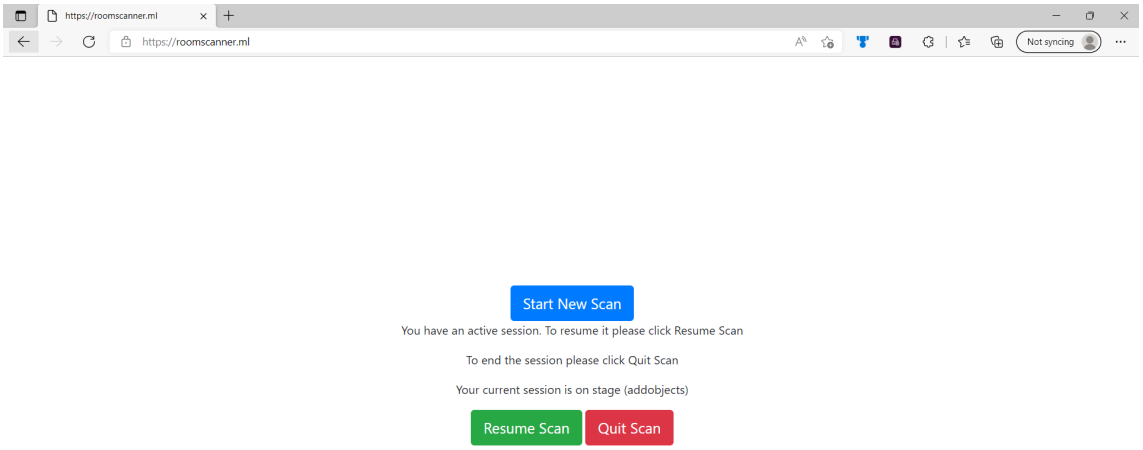
Scan.php

```php
<?php
$title = "Room Scanner";
require("../includes/core.php");
?>
<script type="text/javascript">
function newScan(){
    window.location.href = "https://roomscanner.ml/scan";
}
function resumeScan(){
    window.location.href = "/scan/<?php echo $_SESSION['route'];?>";
}
function quitScan(){
    window.location.href = "/scan/quit";
}
</script>
<div id="progressbar"></div>
<center>
  <div class="h-100 w-70 row align-items-center">
      <div class="col" style="background:white">
    <div class="btn-group btn-group-lg">
        <button type="button" class="btn btn-primary" onclick="newScan()">Start New Scan</button>
    </div>
    <?php
    if($_SESSION['route']){
      echo '
      <p>You have an active session. To resume it please click Resume Scan</p>
      <p>To end the session please click Quit Scan</p>
      <p>Your current session is on stage ('.$_SESSION['route'].')
      <div class="btn-group btn-group-lg">
            <button type="button" class="btn btn-success" onclick="resumeScan()">Resume Scan</button>
        </div>
      <div class="btn-group btn-group-lg">
        <button type="button" class="btn btn-danger" onclick="quitScan()">Quit Scan</button>
      </div>
      ';
    }
     ?>
  </div>
      </div>
</center>
```

This page can look different depending on whether you are starting a new session or are resuming an old session. If you are starting a new session, the page will look as follows:



However, if you are resuming a previous scan the page will look as follows:



In the program, firstly the title of the page is set as RoomScanner, and the core config file is imported using the require() function. I then create 3 functions, newScan, resumeScan and quitScan. If the function newScan() is called, the user will be redirected to the beginning of the scan process. If the function resumeScan() is called, the user will be redirected to the previous position in the settings process, this is done by directing them to the session variable route, which is assigned a different value depending on the stage of which process the user is on. Since it is a session variable, it is kept for the entire session, this means even if the tab is closed, the user can still return to their previous position. The final function created is the quitScan() function, which redirects the user to the /scan/quit page of the website.

On the website page, I create a button, called Start New Scan, using the onclick event, I run the NewScan function, should the button be clicked. In addition, I use an if statement, to check if there anything in the session route variable, if the variable isn't empty, I create two more buttons, the resume scan button, and the quit scan button, which use the same technique as the NewScan button to call their corresponding functions.

Quit.php

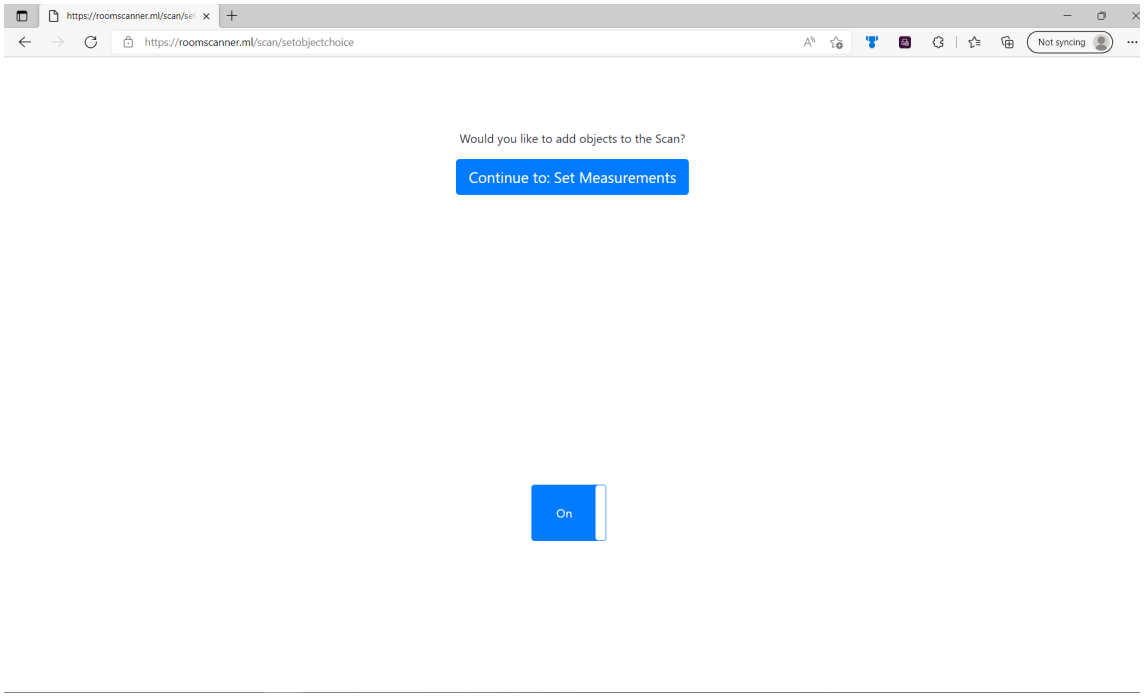
```php
<?php
include("../../../includes/core.php");
unset ($_SESSION["route"]);
header("Location: /");
```

The quit page firstly uses the include() function to check that the core config file has been imported. The unset() function is then used to destroy the route variable. This clears all the information within the variable, so that the session route no longer exists. This allows for the user to begin a new scan. The user is then redirected to the beginning page.

Set Object Choice

```php
<?php
$title = "Set Object Identification";
include("../../../includes/core.php");
if($_SESSION['route'] != "setobjectchoice"){
  if($_SESSION['route']){
    header("Location: {$_SESSION['route']}");
  }else{
    header("Location: /");
  }
}

if($_POST['confirmation'] == "true"){
  if($_POST['objectidentification'] == "on"){
    $_SESSION['objectidentification'] = "on";
    $_SESSION['route'] = "addobjects";
    header("Location: /scan/addobjects");
    die();
  }else{
    $_SESSION['objectidentification'] = "off";
  }
  $_SESSION['route'] = "setnumberofmeasurements";
  header("Location: /scan/setnumberofmeasurements");
}
?>
<link href="https://cdn.jsdelivr.net/gh/gitbrent/bootstrap4-toggle@3.6.1/css/bootstrap4-toggle.min.css"
rel="stylesheet">
<script src="https://cdn.jsdelivr.net/gh/gitbrent/bootstrap4-toggle@3.6.1/js/bootstrap4-toggle.min.js">
</script>
<br>
<br>
<br>
<br>
<form name="NEXT_PAGE" method="post">
<input type="text" name="confirmation" value="true" hidden="true">
<center>
     <p>Would you like to add objects to the Scan?</p>
     <div class="col" style="background:white">
     <div class="btn-group btn-group-lg">
     <button type="submit" class="btn btn-primary" value="Submit">Continue to: Set Measurements</button>
   </div>
    </div>
</center>
<div class="h-100 w-100 row align-items-center">
    <div class="col d-flex justify-content-center" style="background:white">
      <div class="form-check form-switch">
        <input name="objectidentification" type="checkbox" checked data-toggle="toggle" data-width="100"
data-height="75">
      </div>
    </div>
</div>
</form>
```

This page looks as followed:



At the beginning of the program, I use the title() function to name the page, and then use the include() to check that the core config file is imported. Then, using an if statement, I then check whether the user has come from the correct page, which should be “setobjectchoice”, if not, then using another if statement, it redirects the user to either their last page before they quit, or the start page. This means that the user is forced to input all the information and is unable to skip parts of the process, which could cause errors when trying to read the information, this is done on every page, with a different ‘route’, to make sure this happens.

I once again use a form for the inputs, one button and one switch. The button is used to to go to the next page. The switch is a simple on/off switch, created using the checkbox type. To get to the next page, I used an if statement, to check whether the button has been checked, then, because I wanted to redirect the user to different pages depending on their choice on the switch, I use an additional if statement, if the switch is set to on by the user when the button it pressed, a session variable named ‘objectidentification’ is created and given the value “on”, and the value of session variable ‘route’ is changed to “addobjects”. The user is then directed to the /scan/addobjects page, I then use the die() function, so that it does not run the rest of the code, which would cause it to do the next part I have described as well. However, if the switch is set to off by the user when the button is pressed, the value given to the ‘objectidentification’ session variable is instead “off”, the value of the ‘route’ session variable is changed to “setnumberofmeasurements”, and the user is redirected to the /scan/setnumberofmeasurements page.

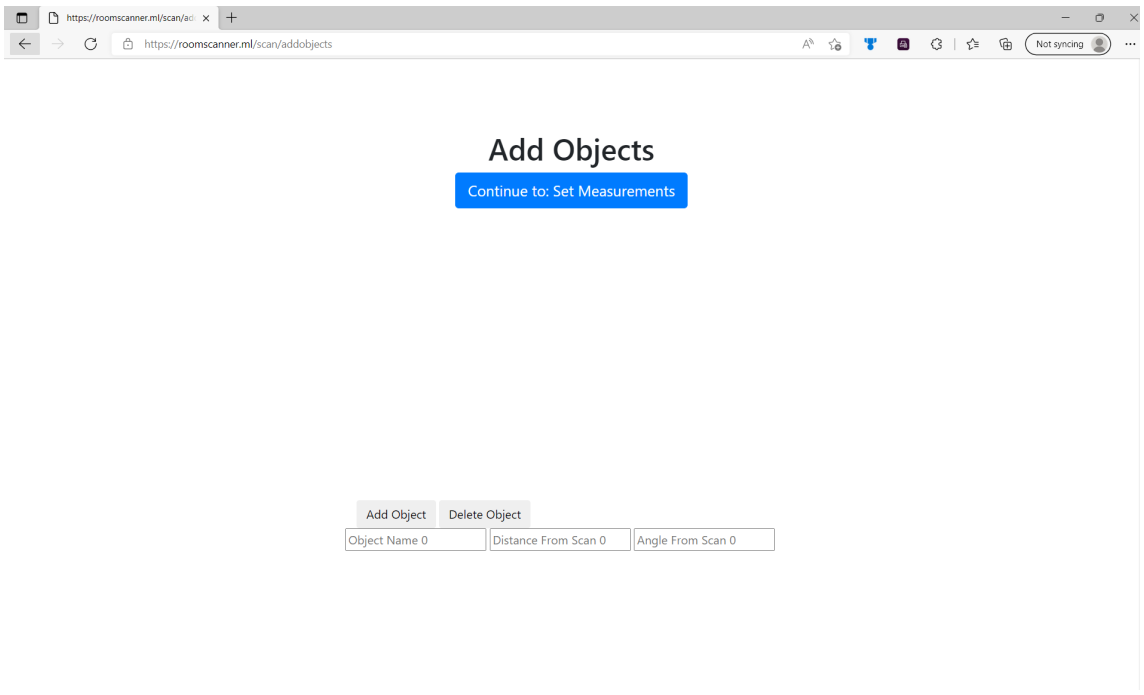
Add objects

```php
<?php
$title = "Add Objects";
include("../../../includes/core.php");
if($_SESSION['route'] != "addobjects"){
  if($_SESSION['route']){
    header("Location: {$_SESSION['route']}");
  }else{
    header("Location: /");
  }
}
if($_POST['confirmation'] == "true"){
  $list_of_object_names = [];
  if(is_array($_POST['object_name'])) foreach($_POST['object_name'] as $id => $value){
    array_push($list_of_object_names, $value);
  }
  $list_of_distances_from_scanner = [];
  if(is_array($_POST['distance_from_scanner'])) foreach($_POST['distance_from_scanner'] as $id => $value)
{
    array_push($list_of_distances_from_scanner, $value);
  }
  $list_of_angles_from_scanner = [];
  if(is_array($_POST['angle_from_scan'])) foreach($_POST['angle_from_scan'] as $id => $value){
    array_push($list_of_angles_from_scanner, $value);
  }
  $_SESSION["list_of_object_names"] = $list_of_object_names;
  $_SESSION["list_of_distances_from_scanner"] = $list_of_distances_from_scanner;
  $_SESSION["list_of_angles_from_scanner"] = $list_of_angles_from_scanner;

  $_SESSION['route'] = "setnumberofmeasurements";
  header("Location: /scan/setnumberofmeasurements");
}
?>
```

```
<link href="https://cdn.jsdelivr.net/gh/gitbrent/bootstrap4-toggle@3.6.1/css/bootstrap4-toggle.min.css"
rel="stylesheet">
<script src="https://cdn.jsdelivr.net/gh/gitbrent/bootstrap4-toggle@3.6.1/js/bootstrap4-toggle.min.js">
</script>
<br>
<br>
<br>
<br>
<form name="NEXT_PAGE" method="post">
<input type="text" name="confirmation" value="true" hidden="true">
<center>
     <h1>Add Objects</h1>
     <div class="col" style="background:white">
     <div class="btn-group btn-group-lg">
     <button type="submit" class="btn btn-primary" value="Submit">Continue to: Set Measurements</button>
   </div>
    </div>
</center>
<div class="h-100 w-100 row align-items-center">
    <div class="col d-flex justify-content-center" style="background:white">
        <div>
           <div class="container ">
        <div>
                <button class="button addMember btn">
                    Add Object
                </button>
                <button class="button delMember btn">
                    Delete Object
                </button>
      </div>
</div>
<section class="addinputs">
</section>
</div>
    </div>
</div>
</form>
<script>
document.querySelector(".addMember").addEventListener('click', getAdd);
document.querySelector(".delMember").addEventListener('click', getDel);

var objects=0;

function getAdd(event) {
    event.preventDefault();
    if(objects <= 10) {
        document.querySelector('.addinputs').innerHTML+=`
          <div class="addedMember">
            <input required class="object_name" type="text" placeholder="Object Name ${objects}"
name="object_name[${objects}]">
            <input required class="distance_from_scanner" type="number" placeholder="Distance From Scan
${objects}" name="distance_from_scanner[${objects}]">
            <input required class="angle_from_scan" type="number" placeholder="Angle From Scan
${objects}" name="angle_from_scan[${objects}]" id="angle_from_scanner[${objects}]">
          </div>
        `;
         objects++;
    }
}

function getDel(event) {
    event.preventDefault();
    if(objects > 0) {
        var added = document.getElementsByClassName("addedMember");
        var lastadded = added[added.length - 1];
        lastadded.parentNode.removeChild(lastadded);
        objects--;
    }

}
</script>
```

In this code, firstly the title is set to “AddObjects”, and the core imports are checked with the include() function. Using an if statement, I then check whether the user has come from the correct page, which is “AddObjects”, if not, then using another if statement, it redirects the user to either

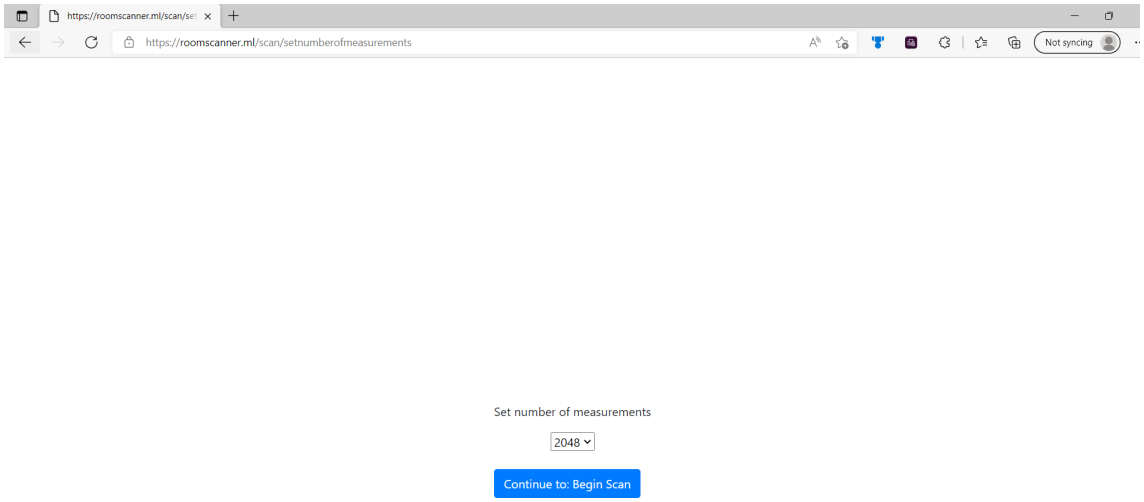
their last page before they quit, or the start page. To create the actual user interface, I have created two functions, getAdd(), and getDel(), and a variable called objects, which counts the number of objects that the user has created. In the getAdd() function, I firstly check that the number of objects added by the user is less than or equal to 10, this sets a limit on the amount of objects that the user is able to add, when the getAdd() function is called, it will add another row of boxes for the user to input information into, and then increase the object count by 1. In the getDel() function, if the number of objects is more then 0, then the most recently added object row will be removed, and the object count will reduce by 1. In the actual form, I create 3 buttons, a “Continue to: Set Measurements” button, an “Add Object” button and a “Delete Object” button. Using an if statement, when the “Add Object” button is pressed, the getAdd() function will be called, I do the same with the “Delete Object” button and getDel() function. If the “Continue to: Set Number of Measurements” button is pressed, the global variable, confirmation is changed to true, and an if statement is used to find when it becomes true. I then create 3 arrays, the first of which is list_of_object_names, I then use an if statement to push the values of the inputted object_names onto the array. I do the same for the distances and angles. I then create three session variables to store each of the three arrays. Then, the session variable ‘route’ is changed to “setnumberofmeasurements” and the user is redirected to /scan/setnumberofmeasurements, using the header() function.



Set number of Measurements

```php
<?php
$title = "Set Number of Measurements";
include("../../../includes/core.php");
if($_SESSION['route'] != "setnumberofmeasurements"){
  if($_SESSION['route']){
    header("Location: {$_SESSION['route']}");
  }else{
    header("Location: /");
  }
}

if($_POST['confirmation'] == "true"){
  $_SESSION['measurements'] = $_POST['measurements'];
  $_SESSION['route'] = "inputemail";
  header("Location: /scan/inputemail");
}
?>
<link href="https://cdn.jsdelivr.net/gh/gitbrent/bootstrap4-toggle@3.6.1/css/bootstrap4-toggle.min.css"
rel="stylesheet">
<script src="https://cdn.jsdelivr.net/gh/gitbrent/bootstrap4-toggle@3.6.1/js/bootstrap4-toggle.min.js">
</script>
<br>
<br>
<br>
<br>
<form id="NEXT_PAGE" method="POST" >
<input type="text" name="confirmation" value="true" hidden="true">

<div class="h-100 w-100 row align-items-center">
    <div class="col d-flex justify-content-center" style="background:white">
      <div class="form-check form-switch">
                  <p>Set number of measurements</p>
        <center>
            <select class="form-select" name="measurements" form="NEXT_PAGE">
              <option value="2048">2048</option>
              <option value="1024">1024</option>
              <option value="512">512</option>
              <option value="256">256</option>
          </select>
          </center>
          <br>
          <button type="submit" class="btn btn-primary" value="Submit">Continue to: Begin Scan</button>
      </div>
    </div>
</div>
</form>
```

In this page, firstly the title is set to “Set Number of Measurements”, and the core imports are checked with the include() function. Using an if statement, I then check whether the user has come from the correct page, which is “setnumberofmeasurements”, if not, then using another if statement, it redirects the user to either their last page before they quit, or the start page. This means that the user is forced to input all the information and is unable to skip parts of the process, which could cause errors when trying to read the information. I then use an if statement to check whether the “Continue to: Begin Scan” button within the form is pressed, if it is pressed, the value of measurements is saved as a session variable, and the value of the session variable ‘route’ is changed to “inputemail”, as the next page the user will be going to is the inputemail page. The header() function then directs the user to the inputemail page. To make the website interface, I use a form, with a form, with 4 different values, within a dropdown menu, which allows the user to chose between 4 of the values. I also add a button, which is called “Continue to: Begin Scan”. I use to make sure that the form is in the middle of the page, so that it looks slightly nicer, especially on mobile devices. The website page looks like the below image:

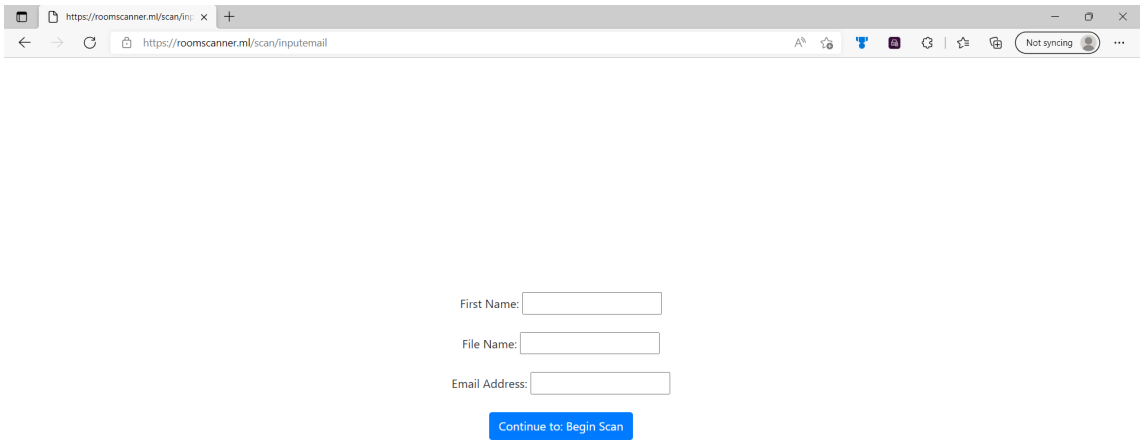


InputDetails

```php
<?php
$title = "Input Email Address";
include("../../../includes/core.php");
if($_SESSION['route'] != "inputemail"){
  if($_SESSION['route']){
    header("Location: {$_SESSION['route']}");
  }else{
    header("Location: /");
  }
}

?>
<html>
<div class="h-100 w-100 row align-items-center">
    <div class="col d-flex justify-content-center" style="background:white">
    <form action="settings" method="post" action="post">
    <center>
    First Name: <input required type="text" name="FirstName" /><br /><br>
    File Name: <input required type="text" name="FileName" /><br /><br>
    Email Address: <input required type="email" name="EmailAddress" /><br /><br>
    <button type="submit" class="btn btn-primary" value="Submit">Continue to: Begin Scan</button>
    </center>
    </form>
  </div>
  </div>
</html>
```

In this page input 3 different pieces of information, your first name, file name and email address, once you have inputted the data, you then can press the button, which will cause you to be forwarded to the next page. In each variable, I have put <input required type=”text” …>, this means that there must be something in the text box created in the form or, the continue to: Begin to scan button will not work. In addition, I have changed the input type for the Email Address to email, this means that an email address must be submitted for it to go to the next page. The inputs are done using a html form, and the values are saved as variables, because the next page of the website controls the data input, I do not need to make these session variables. In the <?php ?> section of the program, the title of the webpage is set to “Input Email Address”, then the include() function checks that the core config file is imported. I then use an if statement, to check what page the user should

be on, by checking the contents of the session variable root, if the user is on the incorrect page, they are then redirected to the correct page. The page looks as follows:



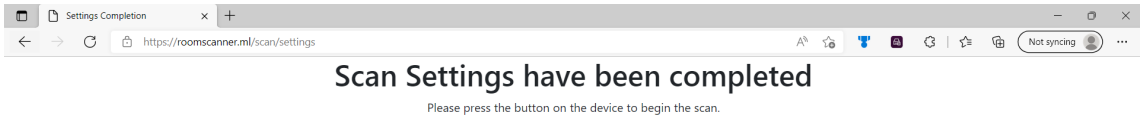
Settings.php

```php
<?php
$title = "Input Email Address";
include("../../../includes/core.php");

$firstname = $_POST["FirstName"];
$filename = $_POST["FileName"];
$emailaddress = $_POST["EmailAddress"];

$settings_json = json_encode(array(
"firstname" => $firstname,
"filename" => $filename,
"emailaddress" => $emailaddress,
"objectidentificationstatus" => $_SESSION["objectidentification"],
"measurements" => $_SESSION["measurements"]
));

$data = $settings_json;
unlink("data.php");
$fp = fopen('data.php', 'a');
fwrite($fp, $data);
fclose($fp);

$items_json = array(
"list_of_object_names" => $_SESSION["list_of_object_names"],
"list_of_distances_from_scanner" => $_SESSION["list_of_distances_from_scanner"],
"list_of_angles_from_scanner" => $_SESSION["list_of_angles_from_scanner"]);

unlink("objects.php");
$fp = fopen('objects.php', 'a');
fwrite($fp, json_encode($items_json));
fclose($fp);
?>
<html>
  <head>
<title>Settings Completion</title>
</head>
<body>
<center>
<h1>Scan Settings have been completed</h1>
<p>Please press the button on the device to begin the scan.</p>
</center>
</body>
</html>
```

This website page encodes the settings data which has been submitted as a json array. Firstly, it takes the session data about the users first name, file name, email address, object choice and number of measurements choice, calling it settings_json, and encodes it as an array. It then deletes the data.php file, using the unlink() function, and creates a new data.php, then it saves the setting_json array into that page. It then does a similar thing with the objects, distances and angle, creating another array called items_json, and gets the session information about these choice. It then deletes the objects.php, the same as data.php and creates a new one, and writes the items_json onto the file. This means that both the data and objects can be easily viewable by a python program which can read json files, by looking at their corresponding website pages, https://www.roomscanner.ml/scan/data and https://www.roomscanner.ml/scan/objects.

The reasons for using the unlink function to delete the previous settings in the data.php and object.php files are mentioned in my pre-competition testing.

In addition to this, it also prints a simple html message saying that the settings have been completed and they can begin the scan. This is to make it straightforward for the user to use. The page looks as follows:



The two pages containing the settings information, data.php and objects.php look as following:

A video demonstrating the user interface:

https://www.youtube.com/watch?v=PtwaN2sv1q4

Timestamps:

0:00 – Start Page

0:03 – Set Object Choice Page and demonstration of on and off switch

0:08 – Add objects page, demonstration of add objects, delete objects and input validation, which makes sure all the information has been inputting before going to the next page.

0:24 – Set number of measurements page, demonstration of drop-down choice. Note, due to the screen recorder, the drop-down box isn't visible in the recording.

0:31 – Input detail page, demonstration of input validation, ensuring a correct email address has been inputted.

0:54 – Completed Settings Page

0:58 – Demonstration of ability to quit scan, or resume scan from previous stage.

1:09 – Demonstration of the fact you cannot access pages which you are not on the correct step for.

QR code link to the video.

Testing

Testing Strategy

There are multiple different methods with which I can use to test whether my product follows the set of objectives. Common ones are:

Method of Testing	Description of Test
End-User testing	This is typically called beta testing. This is when an end user of the product uses the product, using real life data and then notes any errors.
White box testing	This is also known as structural testing. This typically uses trace tables to test every possible logical path through individual modules of codes.
Black box testing	In black box testing, interactions with the interface are tested, to check that inputs are producing the correct outputs.
Module testing	In module testing, you test individual components of a system, such as procedures or functions to check they work.

During the pre-completion phase of testing, I will be using the module testing method, this will allow me to test whether individual functions which I create work as planned. Once my program has been completed, and module testing has also been completed, I will be doing a mix of black box testing and end-user testing. End-user testing will allow me to get feedback from my end-user on how it can be improved, making sure that it is functions how a user would want it to work, and not just how a programmer would want it to work.

Pre-completion Testing

During the development period of the product, I made sure to complete multiple different tests as I went along, this ensured that the code that I had created would work as I intended for it to work, and make sure that different modules worked together as intended. This allowed for any issues to be resolved quickly and efficiently. A couple of examples of issues that I had to solve were:

Issue
If RotationPoint was 0, 180, 270 or 360 an error would occur, and the program would not fully run. In addition, values of the XCoordinatePoint and YCoordinatePoint were not integer numbers, meaning that should they be added as coordinate points in an array there would be an error.
Previous Code

```
def CoordinateConversion(self, LidarDistance, RotationPoint):
    # Working out what side of a midpoint the Coordinates would be - GOING ANTICLOCKWISE
    # If in -X,Y Quadrant
    if 0 < RotationPoint < 90:
        XCoordinateSign = 0  # Negative
        YCoordinateSign = 1  # Positive
    # If in -X,-Y Quadrant
    elif 90 < RotationPoint < 180:
        RotationPoint = RotationPoint - 90
        RotationPoint = 90 - RotationPoint
        XCoordinateSign = 0  # Negative
        YCoordinateSign = 0  # Negative
    # If in X,-Y Quadrant
    elif 180 < RotationPoint < 270:
        RotationPoint = RotationPoint - 180
        XCoordinateSign = 1  # Positive
        YCoordinateSign = 0  # Negative
    # If in X,Y Quadrant
    elif 270 < RotationPoint < 360:
        RotationPoint = RotationPoint - 270
        RotationPoint = 90 - RotationPoint
        XCoordinateSign = 1  # Positive
        YCoordinateSign = 1  # Negative

    # Convert Point of Rotation into a Angle as a radian.
    RotationRadian = ((RotationPoint) * math.pi) / 180
    # Calculate X component of Distance
    XCoordinatePoint = LidarDistance * math.sin(RotationRadian)
    # Calculates Y component of Distance
    YCoordinatePoint = LidarDistance * math.cos(RotationRadian)

    #Changing Coordinate Signs
    if XCoordinateSign == 0:  # Change X Coordinate Signs
        XCoordinatePoint = XCoordinatePoint * -1
    if YCoordinateSign == 0:  # Change Y Coordinate Signs
        YCoordinatePoint = YCoordinatePoint * -1

    # Converting to Pixel Distance
    XCoordinatePoint = XCoordinatePoint * 25
    XCoordinatePoint = round(XCoordinatePoint)
    YCoordinatePoint = YCoordinatePoint * 25
    YCoordinatePoint = round(YCoordinatePoint)

    # Converting to position in array
    XCoordinatePoint = XCoordinatePoint + 250
    YCoordinatePoint = YCoordinatePoint + 250

    return (XCoordinatePoint, YCoordinatePoint)
```

New Code

```
def CoordinateConversion(self, LidarDistance, RotationPoint):
    # Working out what side of a midpoint the Coordinates would be - GOING ANTICLOCKWISE
    # If in -X,Y Quadrant
    if 0 < RotationPoint < 90:
        XCoordinateSign = 0  # Negative
        YCoordinateSign = 1  # Positive
    # If in -X,-Y Quadrant
    elif 90 < RotationPoint < 180:
        RotationPoint = RotationPoint - 90
        RotationPoint = 90 - RotationPoint
        XCoordinateSign = 0  # Negative
        YCoordinateSign = 0  # Negative
    # If in X,-Y Quadrant
    elif 180 < RotationPoint < 270:
        RotationPoint = RotationPoint - 180
        XCoordinateSign = 1  # Positive
        YCoordinateSign = 0  # Negative
    # If in X,Y Quadrant
    elif 270 < RotationPoint < 360:
        RotationPoint = RotationPoint - 270
        RotationPoint = 90 - RotationPoint
        XCoordinateSign = 1  # Positive
        YCoordinateSign = 1  # Negative

    # Convert Point of Rotation into a Angle as a radian.
    RotationRadian = ((RotationPoint) * math.pi) / 180
    # Calculate X component of Distance
    XCoordinatePoint = LidarDistance * math.sin(RotationRadian)
    # Calculates Y component of Distance
    YCoordinatePoint = LidarDistance * math.cos(RotationRadian)

    # Adding Distances and signs for if exactly 0, 90, 180 or 270 degrees
    if RotationPoint == 0:
        XCoordinatePoint = 0
        YCoordinatePoint = LidarDistance
        XCoordinateSign = 1
        YCoordinateSign = 1
    if RotationPoint == 90:
        XCoordinatePoint = LidarDistance
        YCoordinatePoint = 0
        XCoordinateSign = 0
        YCoordinateSign = 1
    if RotationPoint == 180:
        XCoordinatePoint = 0
        YCoordinatePoint = LidarDistance
        XCoordinateSign = 1
        YCoordinateSign = 0
    if RotationPoint == 270:
        XCoordinatePoint = LidarDistance
        YCoordinatePoint = 0
        XCoordinateSign = 1
        YCoordinateSign = 1

    #Changing Coordinate Signs
    if XCoordinateSign == 0:  # Change X Coordinate Signs
        XCoordinatePoint = XCoordinatePoint * -1
    if YCoordinateSign == 0:  # Change Y Coordinate Signs
        YCoordinatePoint = YCoordinatePoint * -1

    # Converting to Pixel Distance
    XCoordinatePoint = XCoordinatePoint * 25
    XCoordinatePoint = round(XCoordinatePoint)
    YCoordinatePoint = YCoordinatePoint * 25
    YCoordinatePoint = round(YCoordinatePoint)

    # Converting to position in array
    XCoordinatePoint = XCoordinatePoint + 250
    YCoordinatePoint = YCoordinatePoint + 250

    return (XCoordinatePoint, YCoordinatePoint)
```

What was changed

To fix the issues when RotationPoint was 0, 90, 180 or 270 I created additional if statements for if RotationPoint was these specific values, and changed the earlier statements to ‘<’,

rather than '<=’. To solve the issue with XCoordinatePoint and YCoordinatePoint not being integers, I used the round function to round them to 0 decimal places.

Issue

When running the distance measurement code, an error message would be outputted as the serial port was closed, this was usually because the Raspberry Pi had been used for other projects where the serial port had been closed.

Previous Code

```
def GetDistance(self):
    count = ser.in_waiting
    if count > 8:
        recv = ser.read(9)
        ser.reset_input_buffer()
        if recv[0] == 0x59 and recv[1] == 0x59:  # 0x59 is 'Y'
            distance = recv[2] + recv[3] * 256
            strength = recv[4] + recv[5] * 256
            return distance + 10
```

New Code

```
#Opening the serial port if it is closed, this is placed at the beginning of the program.
if ser.is_open == False
    ser.open()

def GetDistance(self):
    count = ser.in_waiting
    if count > 8:
        recv = ser.read(9)
        ser.reset_input_buffer()
        if recv[0] == 0x59 and recv[1] == 0x59:  # 0x59 is 'Y'
            distance = recv[2] + recv[3] * 256
            strength = recv[4] + recv[5] * 256
            return distance + 10
```

What was changed

To solve this issue, at the beginning of the program I added an if statement, which would open the serial port should it be closed. This meant that the serial port would always be open when I wanted to run the program.

Issue

When running the distance measurement code, values of “None” for the distance between the wall and the LIDAR would be outputted, this would cause an error to occur, as when working out the coordinate positions, an integer had to be inputted, and “None” wasn’t an integer.

Previous Code

```
def GetDistance(self):
    count = ser.in_waiting
    if count > 8:
        recv = ser.read(9)
        ser.reset_input_buffer()
        if recv[0] == 0x59 and recv[1] == 0x59:  # 0x59 is 'Y'
            distance = recv[2] + recv[3] * 256
            strength = recv[4] + recv[5] * 256
            return distance + 10
```

New Code

```
def GetDistance(self):
    count = ser.in_waiting
    if count > 8:
        recv = ser.read(9)
        ser.reset_input_buffer()
        if recv[0] == 0x59 and recv[1] == 0x59:  # 0x59 is 'Y'
            distance = recv[2] + recv[3] * 256
            strength = recv[4] + recv[5] * 256
            if distance == None:
                distance = self.GetDistance()
            return distance + 10
```

What was changed

To solve this issue, once a distance had been calculated, I added an if statement, which would check whether the value of distance was “None”, if it was, it would recursively call the GetDistance function until a value of distance was calculated which wasn’t “None”. This solved the issue, and also formed part of the error handling within my code, which helped fix potential errors, meaning that the program could successfully finish without errors.

Issue

When the data array was added to the data.php page, instead of replacing the previous settings, the data array was just added to the page, on the line underneath the page, which would cause errors when reading the information on the page on the python program.

Previous Code

```
<?php
$title = "Input Email Address";
include("../../../includes/core.php");

$firstname = $_POST["FirstName"];
$filename = $_POST["FileName"];
$emailaddress = $_POST["EmailAddress"];

$settings_json = json_encode(array(
"firstname" => $firstname,
"filename" => $filename,
"emailaddress" => $emailaddress,
"objectidentificationstatus" => $_SESSION["objectidentification"],
"measurements" => $_SESSION["measurements"]
));

$data = $settings_json;
$fp = fopen('data.php', 'a');
fwrite($fp, $data);
fclose($fp);

$items_json = array(
"list_of_object_names" => $_SESSION["list_of_object_names"],
"list_of_distances_from_scanner" => $_SESSION["list_of_distances_from_scanner"],
"list_of_angles_from_scanner" => $_SESSION["list_of_angles_from_scanner"]);

$fp = fopen('objects.php', 'a');
fwrite($fp, json_encode($items_json));
fclose($fp);
?>
```

New Code

```php
<?php
$title = "Input Email Address";
include("../../../includes/core.php");

$firstname = $_POST["FirstName"];
$filename = $_POST["FileName"];
$emailaddress = $_POST["EmailAddress"];

$settings_json = json_encode(array(
"firstname" => $firstname,
"filename" => $filename,
"emailaddress" => $emailaddress,
"objectidentificationstatus" => $_SESSION["objectidentification"],
"measurements" => $_SESSION["measurements"]
));

unlink("data.php");
$data = $settings_json;
$fp = fopen('data.php', 'a');
fwrite($fp, $data);
fclose($fp);

$items_json = array(
"list_of_object_names" => $_SESSION["list_of_object_names"],
"list_of_distances_from_scanner" => $_SESSION["list_of_distances_from_scanner"],
"list_of_angles_from_scanner" => $_SESSION["list_of_angles_from_scanner"]);

unlink("objects.php");
$fp = fopen('objects.php', 'a');
fwrite($fp, json_encode($items_json));
fclose($fp);
?>
```

What was changed

Previously, when adding new information to the website, it would end up as followed, with a new line created for the new information:

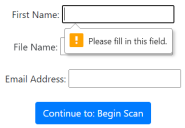
However, I added another line of code, and used the unlink() function, which deleted the contents of the page. Then when the data was added, the page would be empty, which solved the problem.

Now, when adding data, having updated the programming by unlinking the page before, the result ended up as followed:

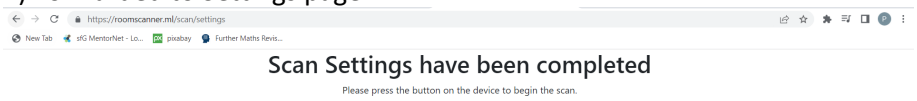
Testing Plan

My testing plan is found below. In this testing, I have first created appropriate tests for each objective (which were testable). I have indicated whether these objectives have been met as well. In addition, I have also tested against other items which could be relevant, such as error handling and input validation.

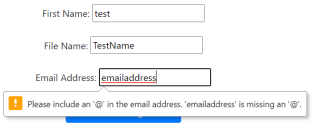
Test ID	1
Objective Number	N/A – Input Validation Test

Purpose of Test	This test is to check to make sure that the user cannot input incorrect information into any of the website setup pages. This test will show that my user interface has the correct input validation.
Description of Test	I will be using multiple different pieces of test data, each will be testing the addobjects and inputemail pages, which both require a written input from the user.
Test Data	For inputemail: 1) FirstName: *Nothing entered* FileName: *Nothing entered* EmailAddress: *Nothing entered* 2) FirstName: Test FileName: TestName EmailAddress: emailaddress@gmail.com 3) FirstName: Test FileName: TestName EmailAddress: emailaddress For addobjects: 1) 1 object: Object 1: Information only entered for distance. 2) 2 objects: Object 1: *Nothing entered* Object 2: Table, 5, 100
Expected Result	For inputemail: 1) Error saying that you need to fill the FirstName field in. 2) Forwards to the settings page 3) Error saying that you need to add an @ to make it a valid email address For addobjects: 1) Error saying that you need to fill in the Object Name field 2) Forwards to the set number of measurements page
Actual Result	**PASS** – The user is only forwarded to the next page if they have entered all the information correctly. For inputemail: 1) 

2) Forwarded to Settings page:

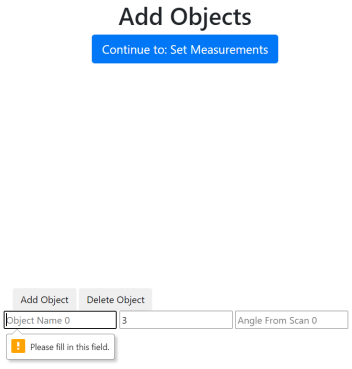


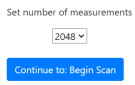
3)

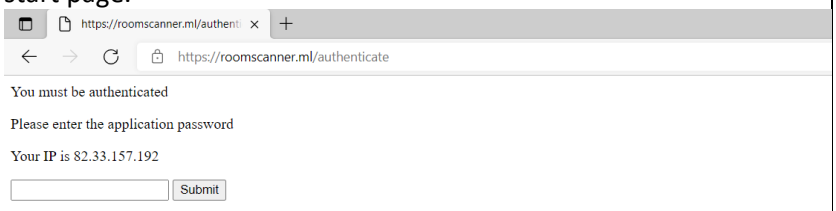


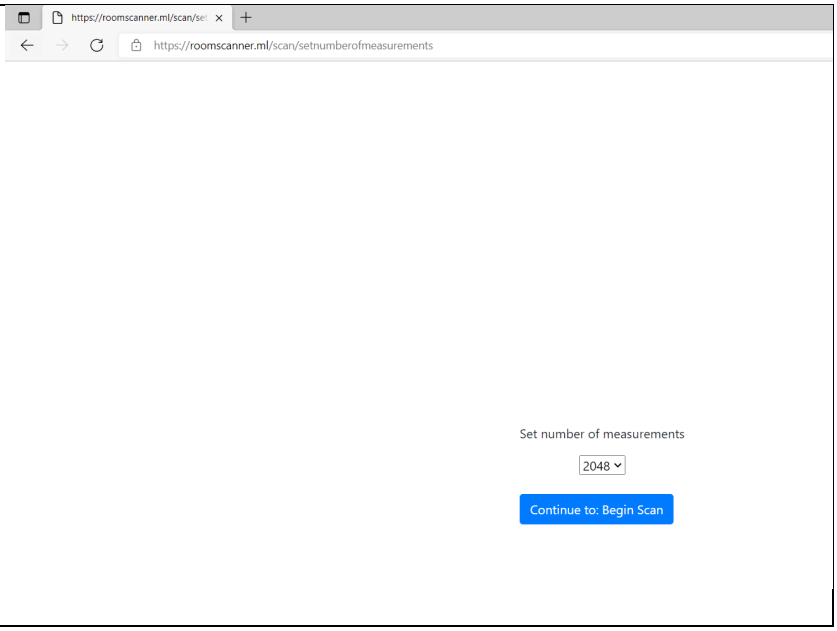
For addobjects:

1)



	2) Was forwarded to this page: 

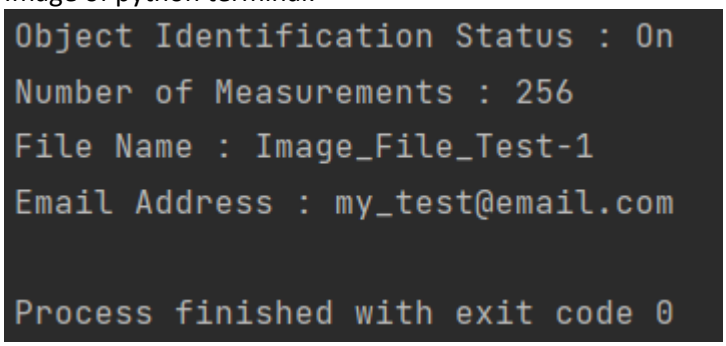
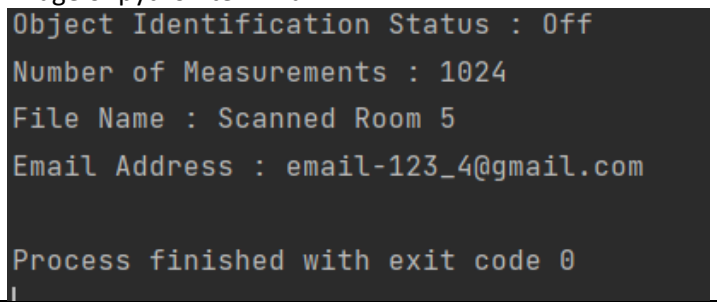
Test ID	2
Objective Number	4
Purpose of Test	This test is to check to make sure that the user cannot access pages which they are not meant to be accessing, meaning they cannot access stages of the settings when they have not completed the previous settings. If my interface demonstrates it can do this, and forwards the user back to their previous page, it shows that the interface is made to be simple to use.
Description of Test	From the start page of the website, I will clear my cookies and then try and navigate to a page of the website through the search bar when I have not inputted any information for the page beforehand.
Test Data	1) From the start page, without completing any settings, the user should try and access the: https://roomscanner.ml/scan/addobjects page. 2) From the start page, begin the settings, choose no for adding objects, and then try and access the: https://roomscanner.ml/scan/inputemail page.
Expected Result	1) When pressing the button, the user should be redirected back to the start page. 2) When pressing the button, the user should be redirected back to the page they were previously on, which should be the number of measurements page.
Actual Result	**PASS** – This test demonstrated that the user was only able to access sections of the website at the points in the settings process where they were meant to be accessing them. 1) Was redirected back to the page I was previously on, which was the start page.  2) Was redirected back to the number of measurements page.

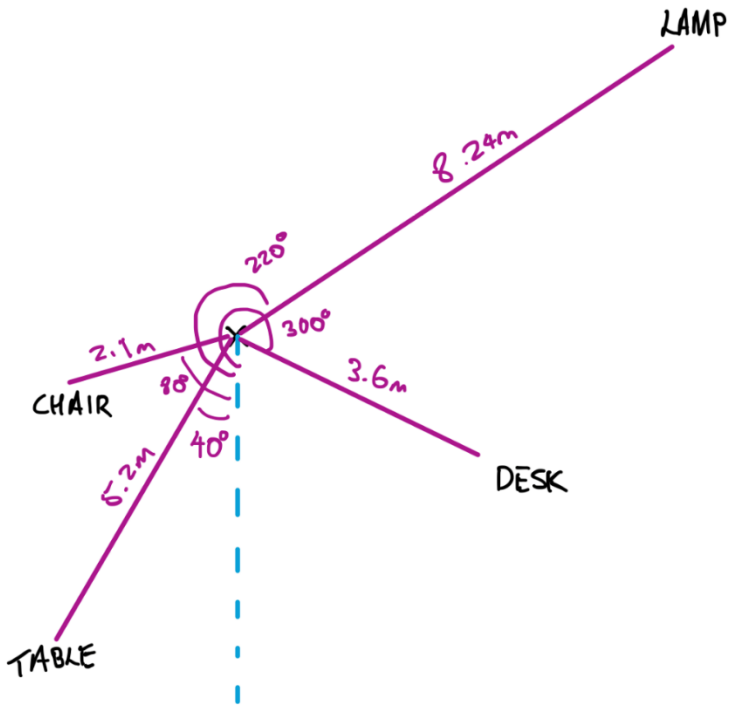
Test ID	3
Objective Number	4
Purpose of Test	This test is used to demonstrate the working of the user interface, in particular the reading of settings submitted by the user by the python program.

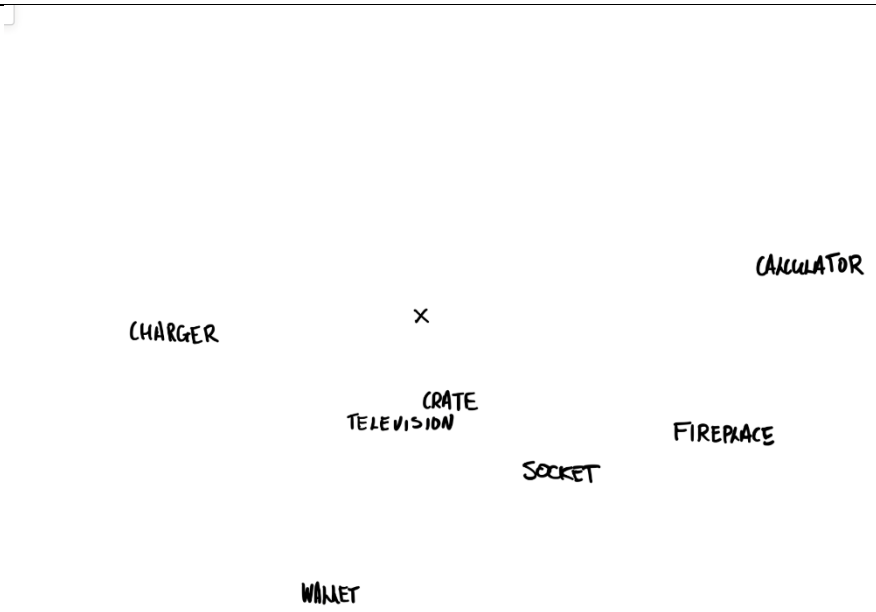
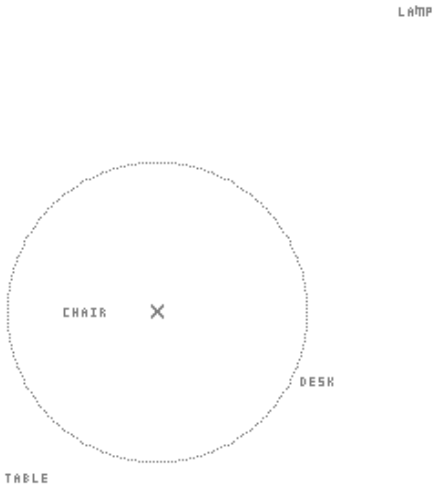
Description of Test	This test will consist of inputting different settings, and then assessing whether the python program is able to correctly read it. I will be using a modified version of my program, which only reads the website pages, and prints out the relevant details.
	The python program used is as found below:
	`import json`
	`import urllib.request`
	`settings = urllib.request.urlopen('https://roomscanner.ml/scan/data').read()`
	`settings_decode = json.loads(settings.decode("utf-8"))`
	`print ("Object Identification Status : ", settings_decode["objectidentificationstatus"])`
	`print ("File Name : ", settings_decode["filename"])`
	`print ("Email Address : ", settings_decode["emailaddress"])`
	`print ("Number of Measurements : ", settings_decode["measurements"])`

Test Data	The relevant information being input can be found below (the information being entered into the objects page is not being tested here, and therefore the content is irrelevant). I will be doing two tests, each with different information to check whether characters such as spaces, hyphens, etc can be used without causing problems to the user interface or python program. In each of the tests, the following information should be inputted into the user interface, which can be accessed at https://www.roomscanner.ml 1) Add Objects: On Number of Measurements: 256 File Name: Image_File_Test-1 Email Address: my_test@email.com 2) Add Objects: Off Number of Measurements: 1024 File Name: Scanned Room 5 Email Address: email-123_4@gmail.com
Expected Result	For each test, the expected result would be the following content being printed in the python terminal after the information had been inputted into the user interface and the python program ran.

	1) Object Identification Status : On Number of Measurements : 256 File Name : Image_File_Test-1 Email Address : my_test@email.com 2) Object Identification Status : Off Number of Measurements : 1024 File Name : Scanned Room 5 Email Address : email-123_4@gmail.com
Actual Result	**Pass** – This test showed that the information inputted by the user in the user interface on the website could successfully be read by a python program, running on a completely different device, and also demonstrated that the user only had to input the information on the interface and not do anything else for the program to work. 1) Image of python terminal:  2) Image of python terminal: 

Test ID	4
Objective Number	3&5
Purpose of Test	The purpose of this test is to demonstrate the ability to add different objects to the room scanner.
Description of Test	This test will consist of taking a scan with multiple different objects added to the scan, showing the ability of the room scanner to have objects within it, I will be using a constant distance for this scan, by removing the lidar from the stepper motor, as I am not currently testing the distance function.
Test Data	For each of the scans, I will be using 256 measurements per rotation. 1) 4 objects: Table, 5.2m, 40 degrees Chair, 2.1m, 80 degrees Lamp, 8.24m, 220 degrees Desk, 3.6m, 300 degrees 2) 7 objects: Wallet, 7.5m, 30 degrees Charger, 8.1m, 90 degrees

	Television, 2.5m, 24 degrees Fireplace, 4.1m, 300 degrees Crate, 1.5m, 2 degrees Calculator, 5.6m, 250 degrees Socket, 2.5m, 330 degrees
Expected Result	Note, for the first expected result, I have displayed the measurements showing how I have approximated the result, however I have not done this for the second one, and instead I am just showing what should be outputted by the completed scan. In addition, I am not adding any distance measurements of the wall, which would result in a large circle around the measurements, although this will be see in the actual result, as I am not testing the distance measurements, but instead the 1)  2)

	
Actual Result	**PASS** – The expected results match the actual results, showing that my system is accurate in taking the inputted details and plotting them in an image. 1)   2)

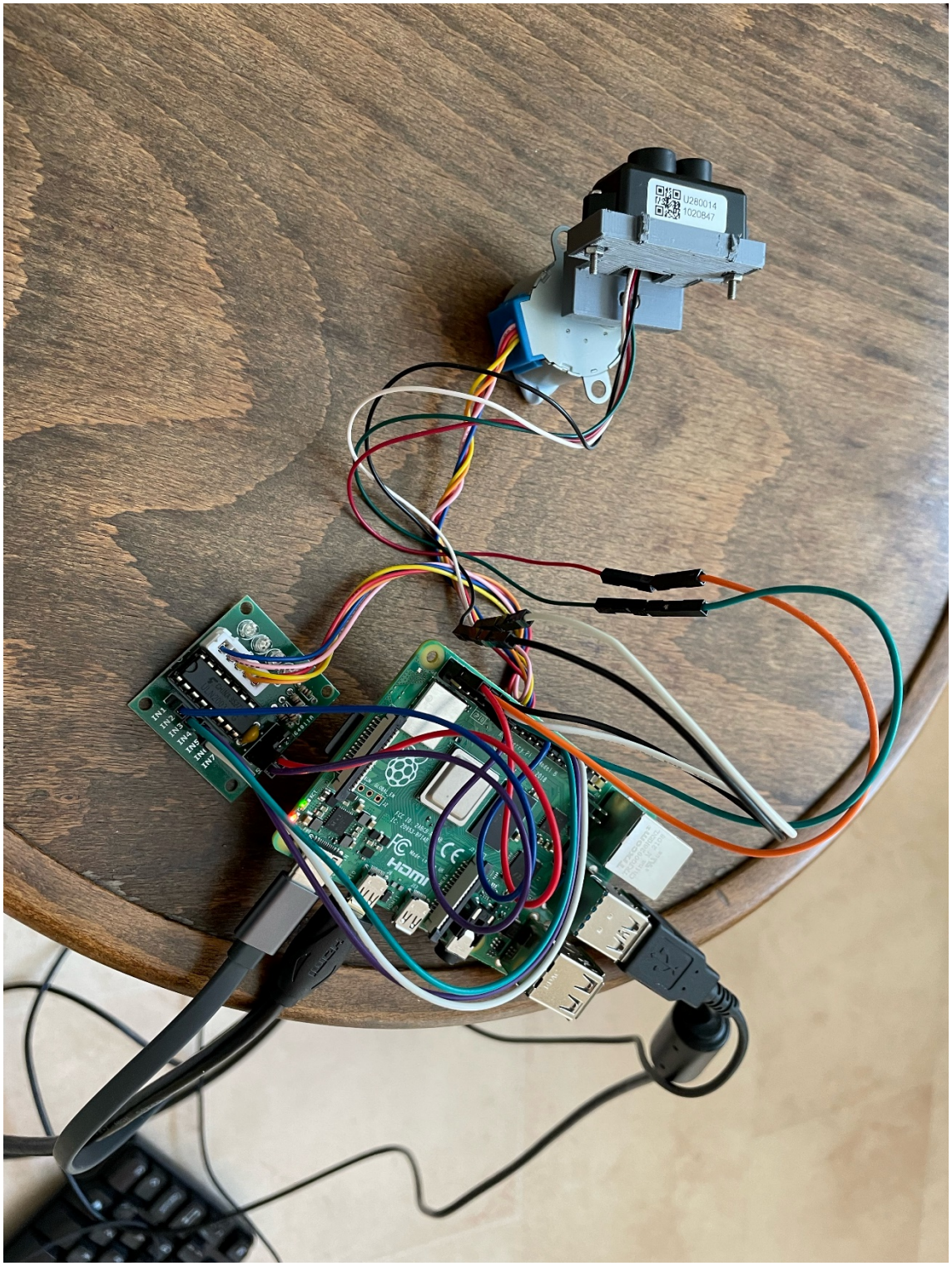
Full System Testing

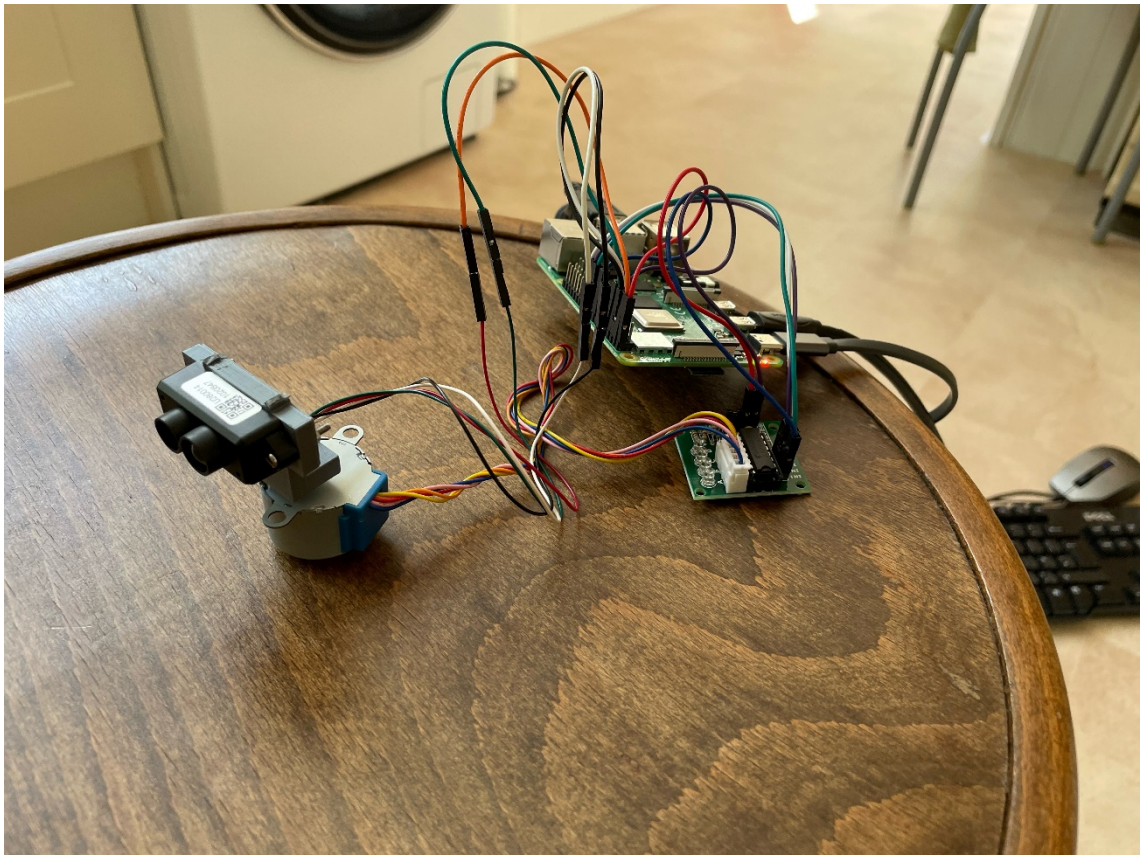
In addition to these tests, which were used to check whether specific functions of the room scanner worked, I decided it would be important test the entire system, as this would be how my users would be using it. I did these tests with my primary user present, so that I could gain feedback on my product, which can be found in the User Feedback section of the document. These tests will demonstrate the majority of the rest of the user objectives which can be tested including, 1,2,4,5 and 6.

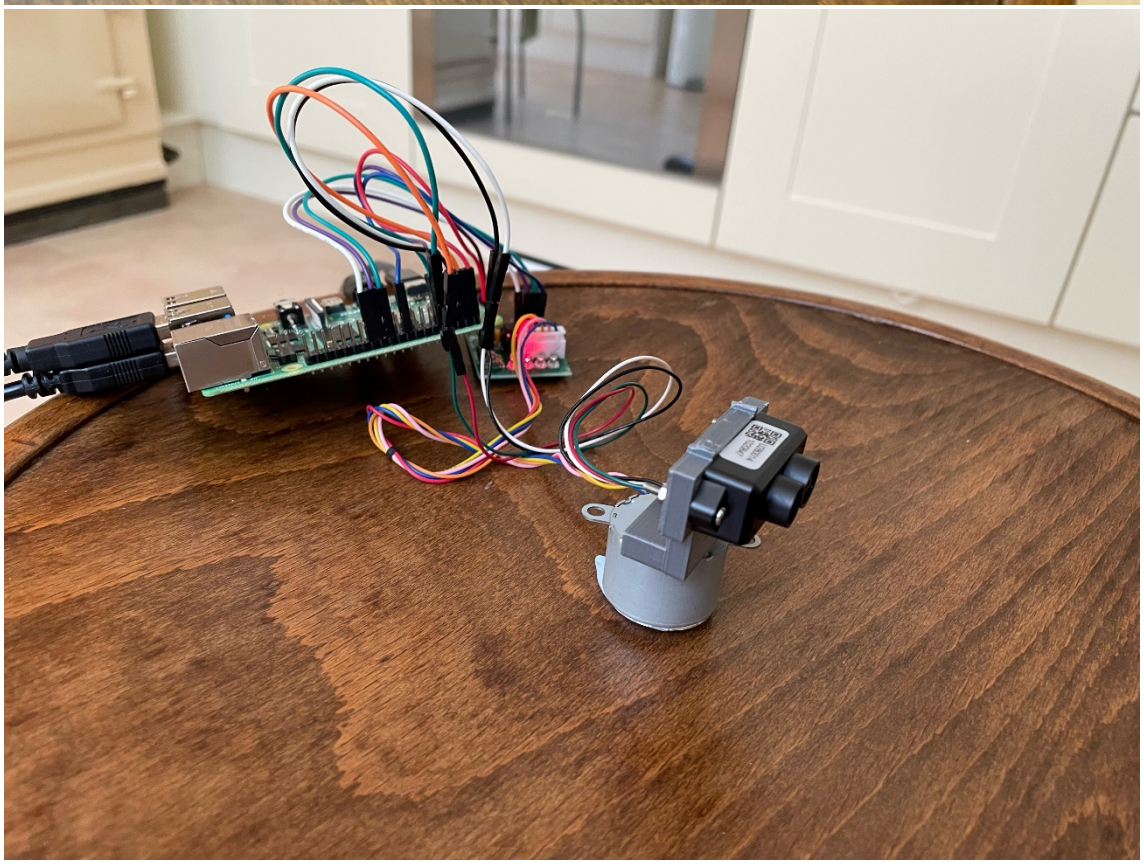
Low Number of Measurements in a small room

The first test that I did was with a low number of measurements, in a small room. I have created a map of the room which I have created by hand which can be compared to the scan created by the room scanner. In addition, I completed this test in normal light conditions, in daytime under sunlight. This would allow me to compare with other scans to see if there was a difference in measurements depending on the light quality, this can be seen in my third test.

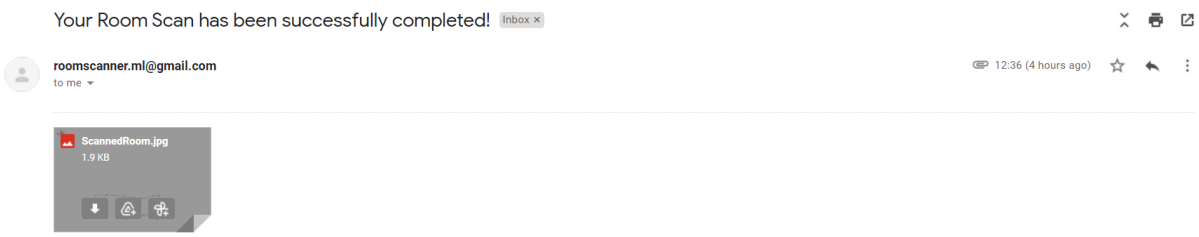
The settings inputted by the user were as followed:
ObjectStatus: Off
NumberOfMeasurements: 512
File Name: LowMeasurementScan
Email Address: DANI5041@rgs.newcastle.sch.uk
Images showing the scan process can be seen below:


U280014
1020847



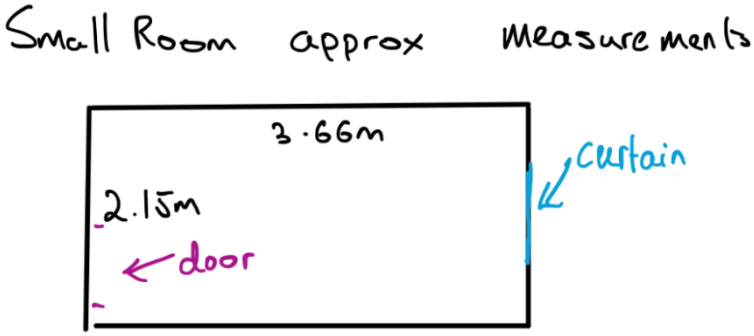


The email received by the user once the scan had been completed can be seen below:



The scanned room image can be found below:

The room map done by hand can be found below:



The scan was shown to be quite accurate, even within a small range. However, there were areas which caused issues, as shown in the left, which was caused by closed curtains, other than this the other parts were practically perfect.

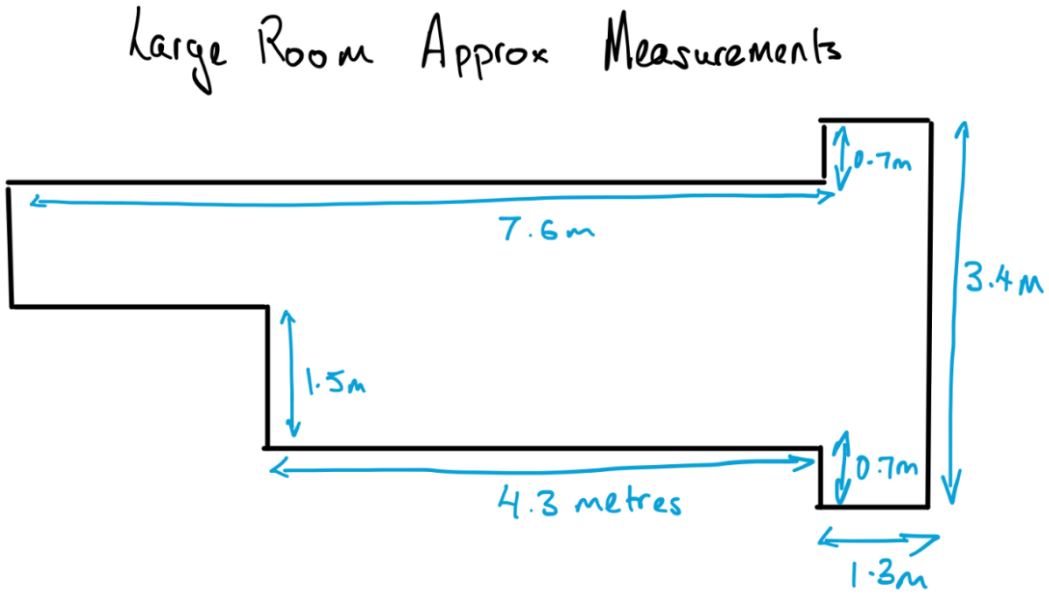
High Number of Measurements in a large room

The second test that I did was with a high number of measurements, in a larger room, this would allow me to demonstrate the detail of which could be displayed by the room scanner.

The settings inputted by the user were as followed:
ObjectStatus: Off
NumberOfMeasurements: 2048
File Name: HighMeasurementScan
Email Address: DANI5041@rgs.newcastle.sch.uk

The scanned room image can be found below:

The room map done by hand can be found below:



This test demonstrated that although the entire scan system worked, since I had created such a large range for the scans, even the largest room in my house looked relatively small compared to the entire image, due to the max scanning distance being 20 metres by 20 metres. A solution to this can be found in my evaluation. The only other issue was that some areas were not being scanned, as they were blocked by other parts of the room, for example in the top left, a hallway was there but wasn't visible on the scan.

Low Number of Measurements with Multiple Objects

The third and final test that I did was using the small room as the first room, with the same number of measurements. However, this time, I did it with little light, by partially closing the blinds in the room. In addition, because I knew that the light would have no effect on the objects added to the scan, I added 4 objects to the scan, to test the object process to confirm that it worked as wanted.

The settings inputted were as followed:

ObjectStatus: On
Object 1:
    Name: Table
    Distance: 0.5
    Angle: 25
Object 2:
    Name: Chair
    Distance: 0.25
    Angle: 140
Object 3:
    Name: Stool
    Distance: 0.75
    Angle: 276

Object 4:

Name: Stool

Distance: 0.75

Angle: 358

NumberOfMeasurements: 512

File Name: LowMeasurementScanWObjects

Email Address: DANI5041@rgs.newcastle.sch.uk

The scanned room image can be found below:

I decided to add the objects again, more spread out to check whether it also worked within a much larger room, which I did not have access to.

Here I used the same settings, however I changed the object distances.

Object 1:

Name: Table

Distance: 1.3

Angle: 25

Object 2:

Name: Chair

Distance:6.2

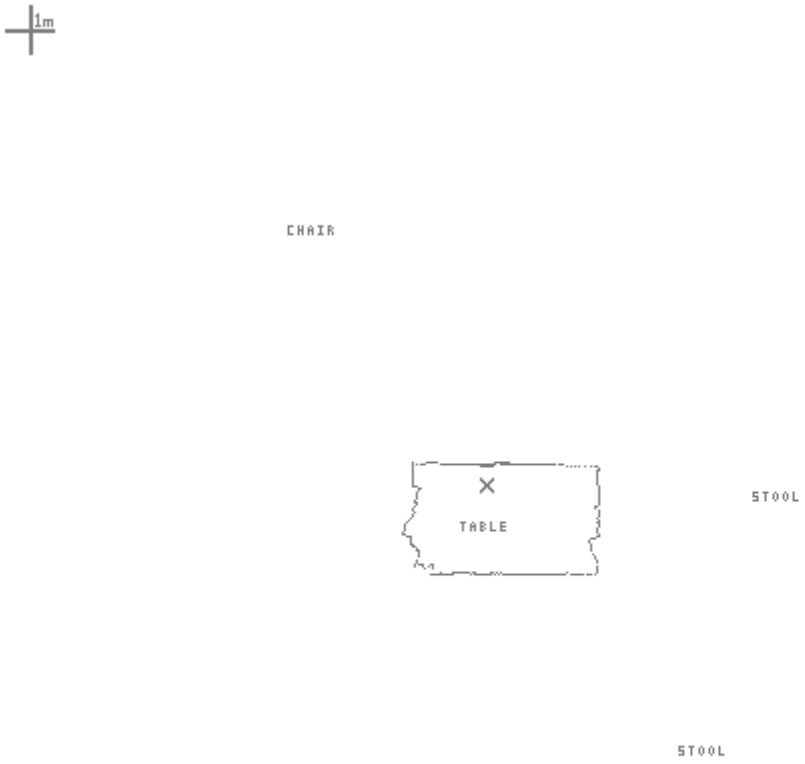
Angle: 140

Object 3:

Name: Stool
Distance: 5.3
Angle: 276

Object 4:

Name: Stool
Distance: 6.8
Angle: 326



These tests all demonstrated to me that the product worked as intended, however there were some areas which could be improved, which are mentioned in the evaluation section of my document.

Evaluation

User Feedback

Once I had completed my project, I then showed it to my end user, Richard. This allowed me to retrieve feedback from him. I asked him multiple questions, mainly to see whether the product worked how he wanted it to work.

Was it easy to complete the settings for the Room Scanner?

> Yes, it was very easy to set up as, the options were very clear, and it was easy to understand the process. I think that it would be useful to have a final page on the interface which listed all the settings choices so that you could check that they were all correct before beginning the scan. I liked how easy the language was to understand, as it made everything very straight forward. It was also very useful that you were able to have the choice on whether to start the settings from the beginning or to start again from where you were last time.

Would you use this system over the alternatives which are currently available?

> Yes, I would use this system over the alternatives which are currently available, as this system is so much cheaper than other alternatives, meaning that I am much more willing to use a product like this, instead of worrying about damaging an incredibly expensive piece of kit. It would be nice in the future if this was to become a product, for a strong and sturdy case for the device to be stored in, however I understand that this is unrelated to Computer Science.

What did you think about the sharing of scans functionality of the system?

> I thought that the way that scans were shared was very quick and efficient, as they were sent automatically once the scan had been completed via an email. This meant that no additional input had to be done, and I could access the completed scan from anywhere, so long as I had internet connect. To improve on the current system, it would be nice to see different sharing methods, such as a WhatsApp or Discord message, which would allow you to see your scans even if you were not logged into your email account. In addition, it would be nice to have a way of viewing all the completed scans. Although this can be done through looking at the CompletedScans folder within the Raspberry Pi, this may be difficult for someone with little technical expertise to do.

Would there be any other settings that you think would be important to add?

> I think that the current settings are very good, especially the fact that you get a choice on the number of measurements that you want to take, which can allow you to take a very short scan or take an incredibly long detailed scan. However, I think that to improve it, you could add an option which will allow you to change the scale of the image, this would allow you to scan smaller rooms at higher detail, and the image size would depend on the scale setting.

Potential User Feedback

In addition to getting user feedback from my end user, I decided it would also be useful to get feedback from the forum websites that I used to gain information in the forms of a survey in my analysis of the problem, so I posted a brief description of the problem I was solved and then numerous images and a description of the end product and asked users to give me their feedback on

what they liked about the product and what they thought could be improved. The feedback I received was as follows:

> "I think this is a very good idea, although it would be nice if it was slightly simpler to create it yourself as a DIY product, as it currently seems quite complicated to make. However, I think that the features look superb, and it could be useful for getting measurements of a room, as I find it incredibly boring taking measurements of a room"

> "I quite like the idea of the product, however I think it would be much better if you could add dimensions to the objects that you are adding to the scan, because although you are adding a table, you won't be able to know whether it is a large dining table, or just a very small desk. Other than this, I think it is a good idea and has been executed well."

> "This looks amazing! It looks incredibly suitable to use, even though I don't frequently use technology like this, I think that I could really benefit from something like this. If the measurements of the device are pretty accurate, I would definitely think about getting something like this!"

Objective Evaluation

No	Object	Evaluation
1	The program must be able to take an accurate measurement of the distance from product to the wall it is facing, using a Lidar distance sensor.	**Completely Achieved** This objective has been met well, when testing the value of the distance calculated with the scanner was within 10% of the value of distance measured with a tape measure between 5-10 metres, and within 5% between 0-5 metres, this is a very accurate value. This outcome could be improved even more by using a more accurate LiDAR sensor, however it is debatable whether the percentage decrease in accuracy will be worth the price increase, as this product is meant to demonstrate that a room scanner can be made at a relatively cheap price.
2	The program for the product should be written in a way which will allow for easy adaptations of the product, allowing for functions to be added to edited.	**Completely Achieved** Through the usage of object orientated program, I have separated the process into multiple classes. This has made it very easy to add new functions into the program, as all you need to do is add the function to its corresponding class, e.g. the CalcualationControl class, and then edit the OutputControl class so that the newly added function is called in the correct place during the scan.
3	Once the product has taken measurements of the entire room, it should be able to create an image showing an accurate 2-dimensional representation of the room, this image should be easy to understand and in a commonly used format to allow to be opened and shared by everyone.	**Completely Achieved** This has also been completely achieved, I think with the operation of the LiDAR and stepper motor working perfectly together. The saving of the scan as a png image is also very effective as it is used by practically every image processing application. I also think that the email sharing works well, however, to improve, after speaking to my end user, I think it would be good to add other ways of sharing the completed scan, such as sending via social media. After researching it, I think it would be quite simple to use applications such as discord for sharing as well.

4	The product should have a simple and easy to use interface to allow people who aren't as confident at technology to use it with easy, allowing for more potential users.	**Completely Achieved** From talking to my end user, I believe that this objective was also very successful, with my end user being able to setup and complete a scan without any help from myself, showing that even without much technological knowledge, it is easy to set up and complete the scan.
5	The product should allow for objects to be added to the scan to allow users to add more detail to their completed scan should they want to.	**Completely Achieved** This objective has been successfully achieved. I think that the usage of the files to hold the file data for letters, which is then read and added to a hash table could potentially be made slightly more efficient in the future, however this process works very well with the letters getting added to the images perfectly, which allows for more detail to be added to a completed room scan. In addition, as another improvement for this, I think another potential change would to be add boxes around the words which would give a better demonstration of the size of the object. However, I think that the current features work well how they are as well.
6	The product must not require constant human altercations, e.g., should be left to complete its task without having to be constantly adjusted.	**Completely Achieved** From talking to my end user, this object has been completely achieved. Once the settings have been completed and the python program has been run, the user can leave and let the scan complete by itself, and then get the completed scan image through emails, without any adjustments throughout the process.

Analysis of User Feedback with Further Suggestions and Improvement

To improve my product, I think that I should create a better way for the scanned images to be shared. Currently, they are emailed to the user who completed the scan and are also available to access through the Raspberry Pi, however this may be difficult to do for someone who doesn't understand how to use a Pi, meaning if the email containing the scan is lost, it will be difficult for people to recover the scan, in the future I would like to add an additional section to the website, called "View Previous Scans", where you would be able to see all the previous scans which had been completed on the device, and also share them with others. This would allow for the scan images to be accessed much easier, as it wouldn't require direct access to the Raspberry Pi should the email attachment be lost. This was highlighted as something that my end user thought would be an important thing to add. In addition, it was highlighted that I would be able to add different methods for sharing the scan, such as through different social media applications, such as discord or snapchat. This could be easily done by adding a new function within the ShareScan class, another page in the user interface which allowed the user to select how they wanted their data scanned etc.

In addition, I think that another change which could improve my product would be to add outlines around the objects which the user had added, demonstrating the actual size of the object compared to the room. This could be done by adding more information for the user to input for each object, which could be the width and depth of the object, allowing for boxes to be made around the

position of the inputted object. This was mentioned by my potential user as addition which may have been useful.

Bibliography

- “Research on Distance and Proximity Sensors” - https://www.barkhauseninstitut.org/research/lab-1/our-blog/distance-sensor-research
- Mehmet G, Umit I. and Melih B. “A review of recent research in indoor modelling & mapping” - https://www.researchgate.net/publication/305768116_A_REVIEW_OF_RECENT_RESEARCH_IN_INDOOR_MODELLING_MAPPING
- Ninad M. and Srushti N. “Review of Lidar Technology” - https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3604309
- Johan M. “3d scanner, Accuracy performance and challenges with a low cost 3d scanning platform” - https://www.diva-portal.org/smash/get/diva2:1200549/FULLTEXT01.pdf
- Andrew R., Oliver W. “Crowd-sourcing world models with OpenRoomMap” - https://www.cl.cam.ac.uk/research/dtg/www/files/publications/public/acr31/rice-openroommap.pdf