Leaving Certificate

Technology

2021 | 2022

Higher Level Thematic Brief

Exam Number 111392

Contents

Thematic Brief

The term disability can refer to any condition of the body or mind (impairment) that makes it more difficult for the person with the condition to do certain activities (activity limitation) or interact with the world around them (participation restrictions). It is a complex phenomenon, reflecting the interaction between features of a person's body and features of the society in which he or she lives.

There are many types of disabilities, including, but not limited to, those that affect a person's:

- Vision
- Movement
- Learning
- Communicating
- Hearing
- Social relationships

In this context and with a focus on modern materials and processes, design and manufacture a working model of a device, system or technological aid that could enhance or improve the quality of life of a person with a disability.

What is assistive technology?

What is assistive technology?

Technology has the power to strengthen opportunities for everyone, but it must be intuitive to have an impact.

Assistive technology is technology used by individuals with disabilities in order to perform functions that might otherwise be difficult or impossible. Assistive technology can include mobility devices such as walkers and wheelchairs, as well as hardware, software, and peripherals that assist people with disabilities in accessing computers or other information technologies.

For example, people with limited hand function may use a keyboard with large keys or a special mouse to operate a computer, people who are blind may use software that reads text on the screen in a computer-generated voice, people with low vision may use software that enlarges screen content, people who are deaf may use a TTY (text telephone), or people with speech impairments may use a device that speaks out loud as they enter text via a keyboard.

Ultimately, AT is any device, software, or equipment that helps people with learning disabilities learn, communicate, or function better.

Examples of Assistive Tech

Vision

Large screens and brighter screens, and narrators to read text are great technologies that can help people who are blind, color blind, or have low vision.

Hearing

For those who are hard of hearing, have hearing loss, or have deafness, there are specialized features that can provide solutions including closed captioning, mono sound, and live call transcription.

Neurodiversity

Innovative tools such as dictation and Windows Hello sign-in can make the digital world more accessible for those who live with dyslexia, seizures, autism, or other cognitive differences.

Learning

Computer applications for people living with learning disabilities can help increase focus, concentration, and understanding—and include tools to improve reading and writing skills.

Mobility

Many products help people living with arthritis, quadriplegia, spinal cord injuries, and other mobility issues to navigate the digital world in non-traditional ways.

Mental health

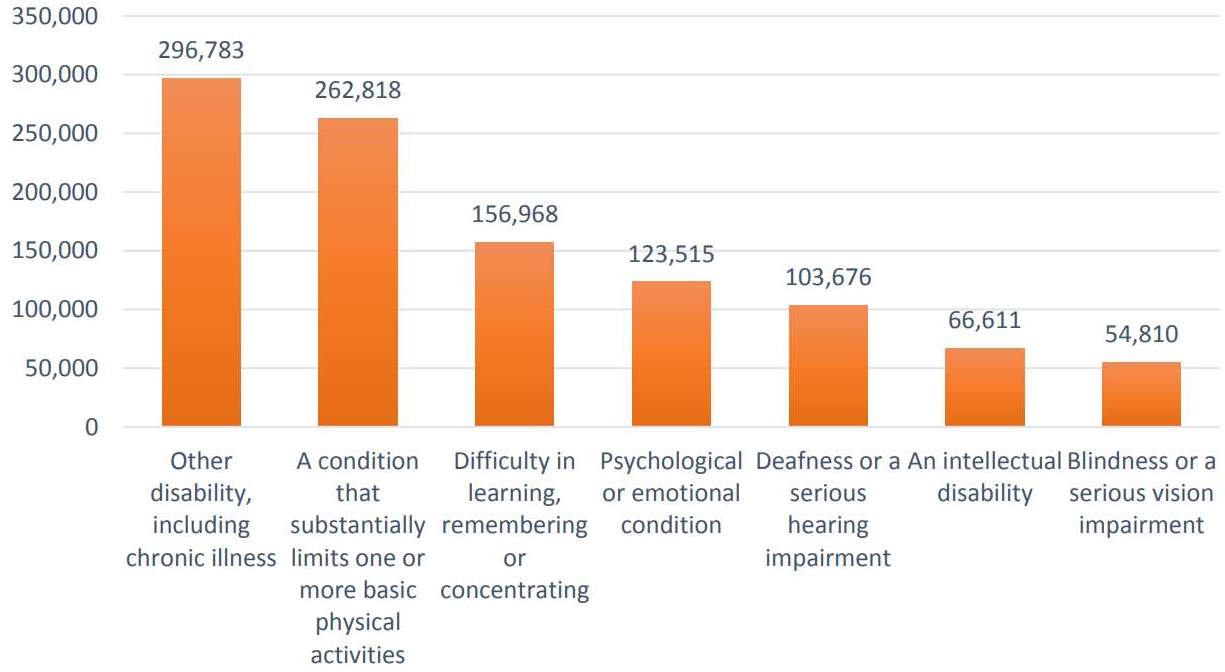
There are assistive technologies for people living with issues such as bipolar disorder, anxiety, PTSD, depression, or ADHD. Products can help with distraction, reading, and concentration.

What is a disability?

A **disability** is classified as a physical or mental problem that makes it difficult or impossible for a person to walk, see, hear, speak, learn, or do other important things. Some disabilities are permanent, or last forever. A disability can be something a person was born with. Or it can be the result of an illness or an accident.

The most common type of disability is 'other disability, including chronic illness'. This might include, for example, asthma, diabetes or heart disease. The next most common is 'a condition that substantially limits one or more basic physical activities', like arthritis or a spinal injury.

Figure 1: Common types of disability



DISABILITY FEDERATION OF IRELAND

DISABILITY PROFILE: IRELAND[1]

13.5% of the population of Ireland, or **643,131** people have at least one disability.

AGE

Their age profile is:

0-14	59,086
15-24	53,465
25-44	123,638
45-64	182,554
65+	224,388

Type of Disability

People in Ireland reported having the following types of disabilities and conditions:

- **296,783** Difficulty with pain, breathing, or other chronic illness/condition
- **262,818** Difficulties with basic physical activities
- **156,968** Learning difficulty
- **123,515** Mental health
- **103,676** Deafness or serious hearing impairment
- **66,611** Intellectual disability
- **54,810** Blindness or a serious vision impairment

EDUCATION[2]

In 2016, only 13% of people who have a disability were in education compared to 19% of the general population in Ireland.[3] For those whose education had finished:[4]

- **31%** did not progress beyond primary education compared to 13% of the general population.
- **16%** had completed upper secondary compared to 20% of the general population.
- **13%** had completed further education compared to 16% of the general population.
- **20%** had completed higher level education compared to 36% of the general population.

EMPLOYMENT

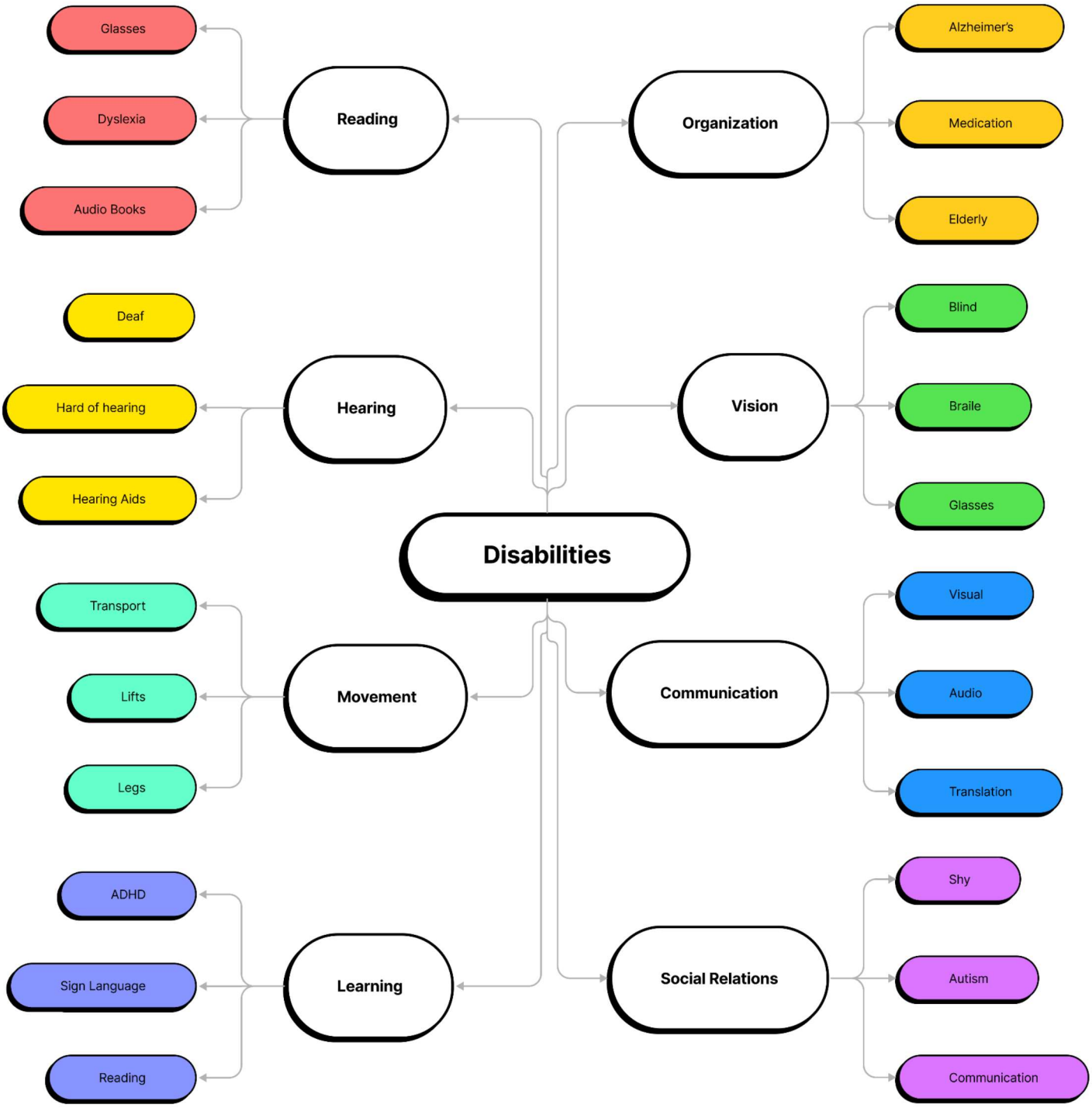
22% of people who have a disability were at work compared to 53% of the general population.

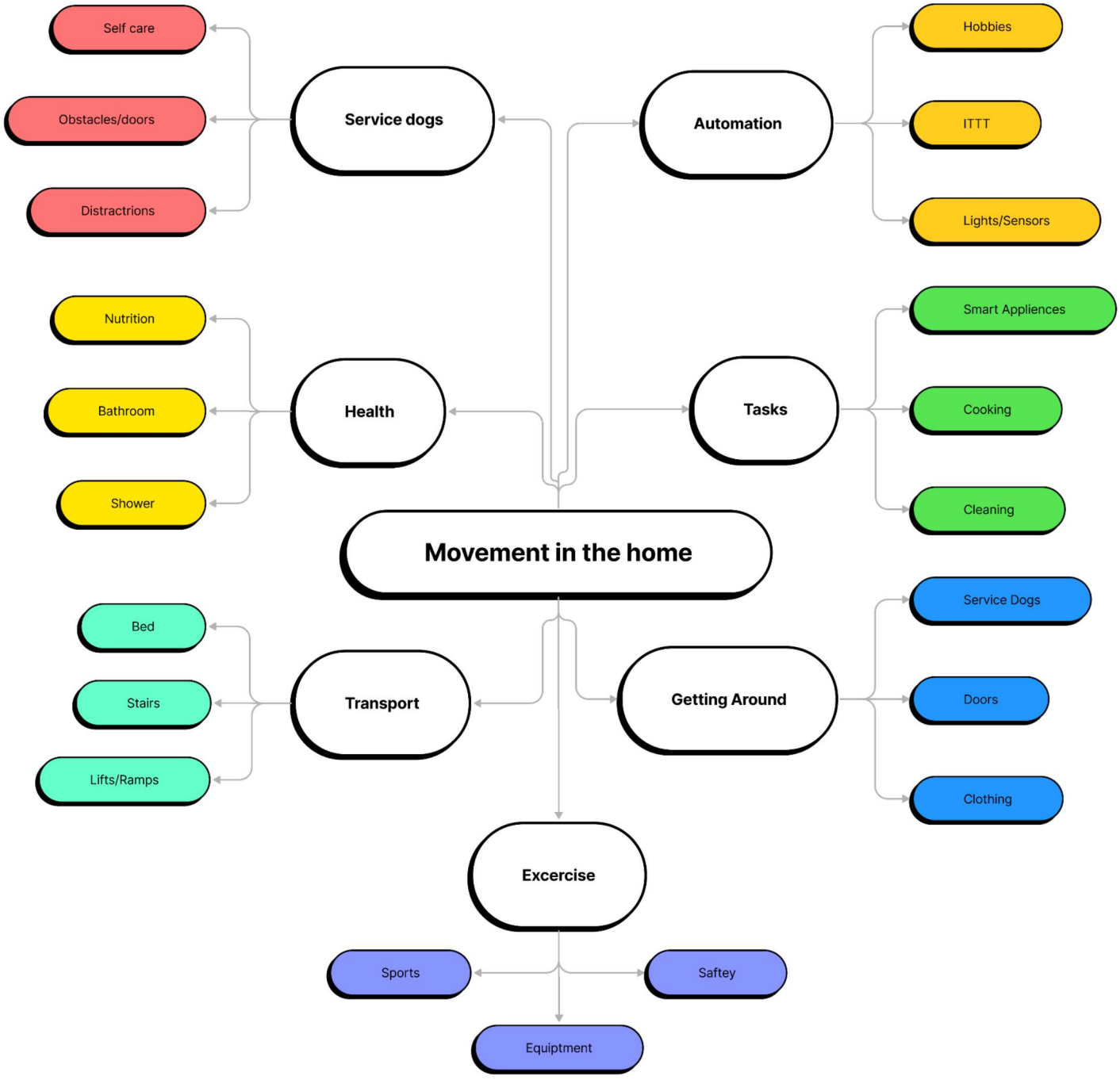
TRANSPORT

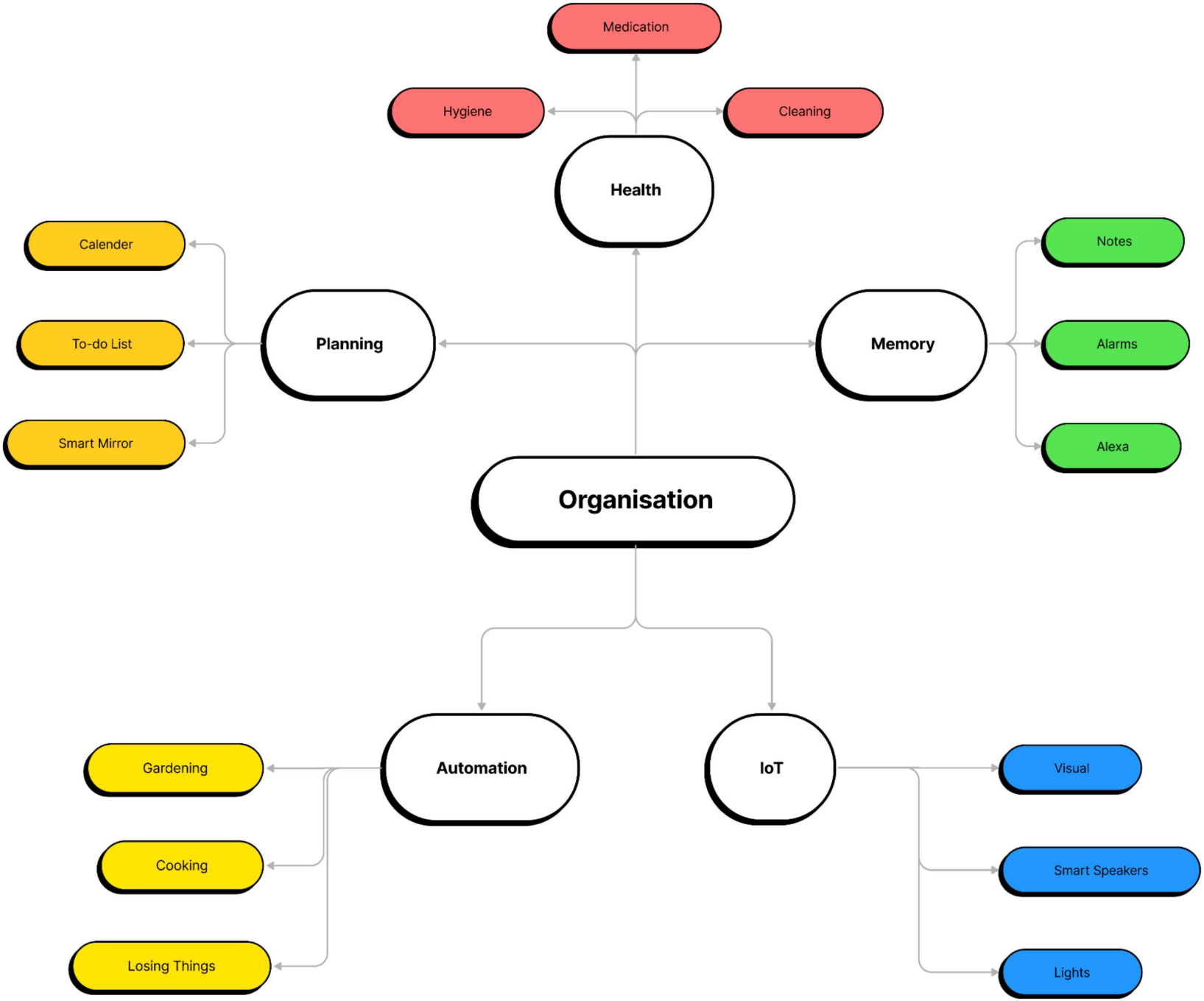
24% of people who have a disability did not have access to a car or van compared to 12% of the general population.

HOUSING

15% of people who have a disability lived in social housing compared to 9% of the general population.[5]


Disabilities
Reading
Glasses
Dyslexia
Audio Books
Hearing
Deaf
Hard of hearing
Hearing Aids
Movement
Transport
Lifts
Legs
Learning
ADHD
Sign Language
Reading
Organization
Alzheimer's
Medication
Elderly
Vision
Blind
Braile
Glasses
Communication
Visual
Audio
Translation
Social Relations
Shy
Autism
Communication


Movement in the home
Service dogs
Self care
Obstacles/doors
Distractrions
Automation
Hobbies
ITTT
Lights/Sensors
Health
Nutrition
Bathroom
Shower
Tasks
Smart Appliences
Cooking
Cleaning
Transport
Bed
Stairs
Lifts/Ramps
Getting Around
Service Dogs
Doors
Clothing
Excercise
Sports
Saftey
Equiptment


Medication
Hygiene
Cleaning
Health
Calender
To-do List
Smart Mirror
Planning
Memory
Notes
Alarms
Alexa
Organisation
Automation
IoT
Gardening
Cooking
Losing Things
Visual
Smart Speakers
Lights

What am I asked to do?

Having carried out my research I believe assistive technology can be:

1. Hardware, software or equipment.
2. Help people with any kind of disability.
3. Help people day to day or in special instances.

I am asked to design an work model of a device or system that:

1. Helps a person with a disability in everyday life.
2. Disabilities may include: vision, movement, learning, communication, hearing, and social relations.
3. Is focused on modern technology.
4. Has an electro-mechanical element; such as a motor, servo or pump.
5. Has dimensions less than or equal to a 500mm3 area.

Other specification that are import in my project include:

1. The system must function well and be fit for purpose
2. My portfolio should be fully complete and comprehensive of my project
3. The design and realization of my project must be environmentally friendly
4. The artifact must be made out of safe materials
5. The artifact must not cause any harm to the user or people around it
6. All electronics and circuitry should be housed to a good standard and not have any risk of causing electric shock, shorts, sparks or fire.
7. I should leave as little waste in my project as possible.

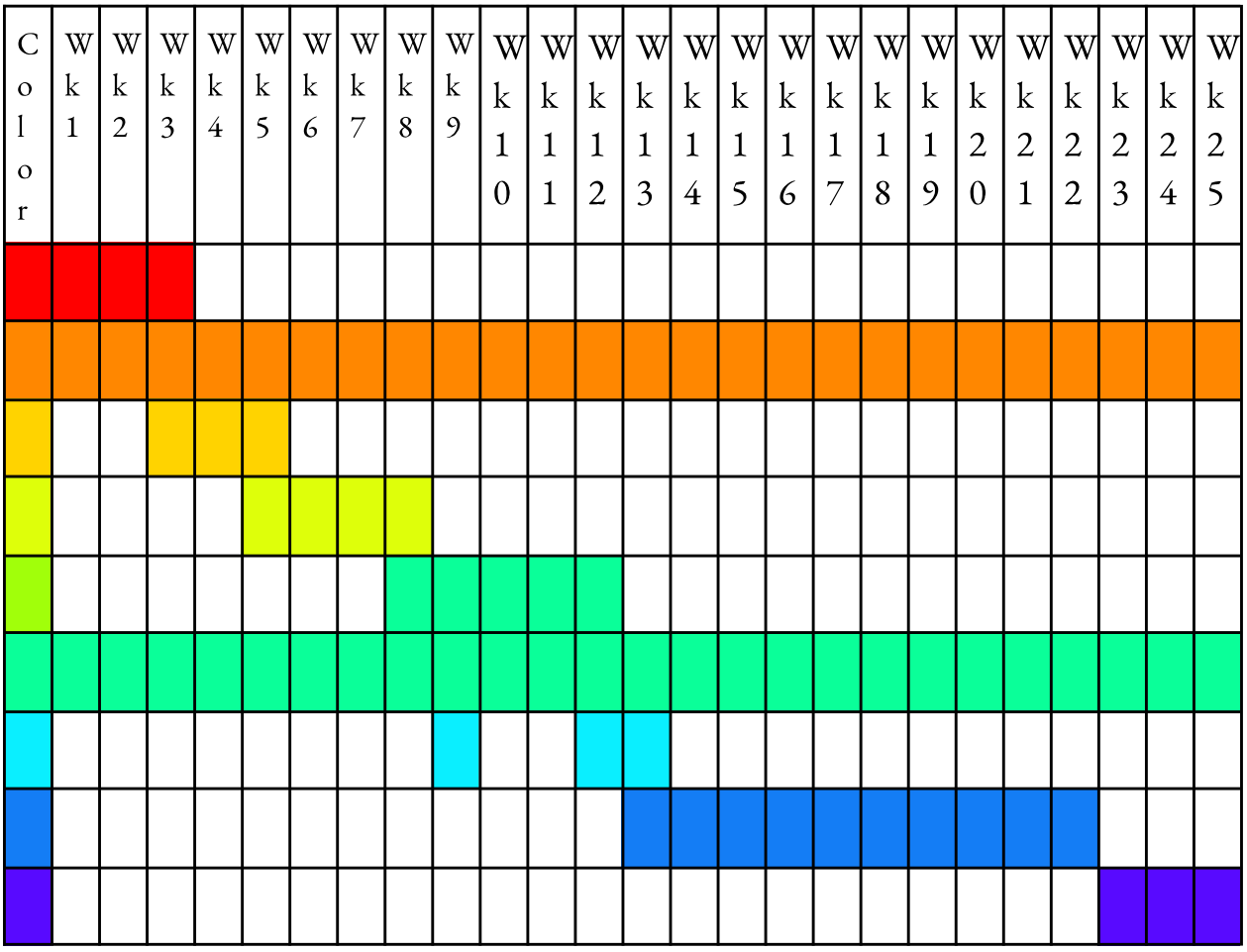
The most import things to keep in mind is the project:

1. Be made of safe and reusable materials.
2. Be safe for the user to operate.
3. Be designed and manufactured based on the skills gained in secondary school to date.

Overall Management

I have 2 double classes and one single class of technology per week. The project work will be designed and manufactured during the two double classes over a period of 25 weeks. In order not to waste time I have set out a plan of how I will manage the project below.

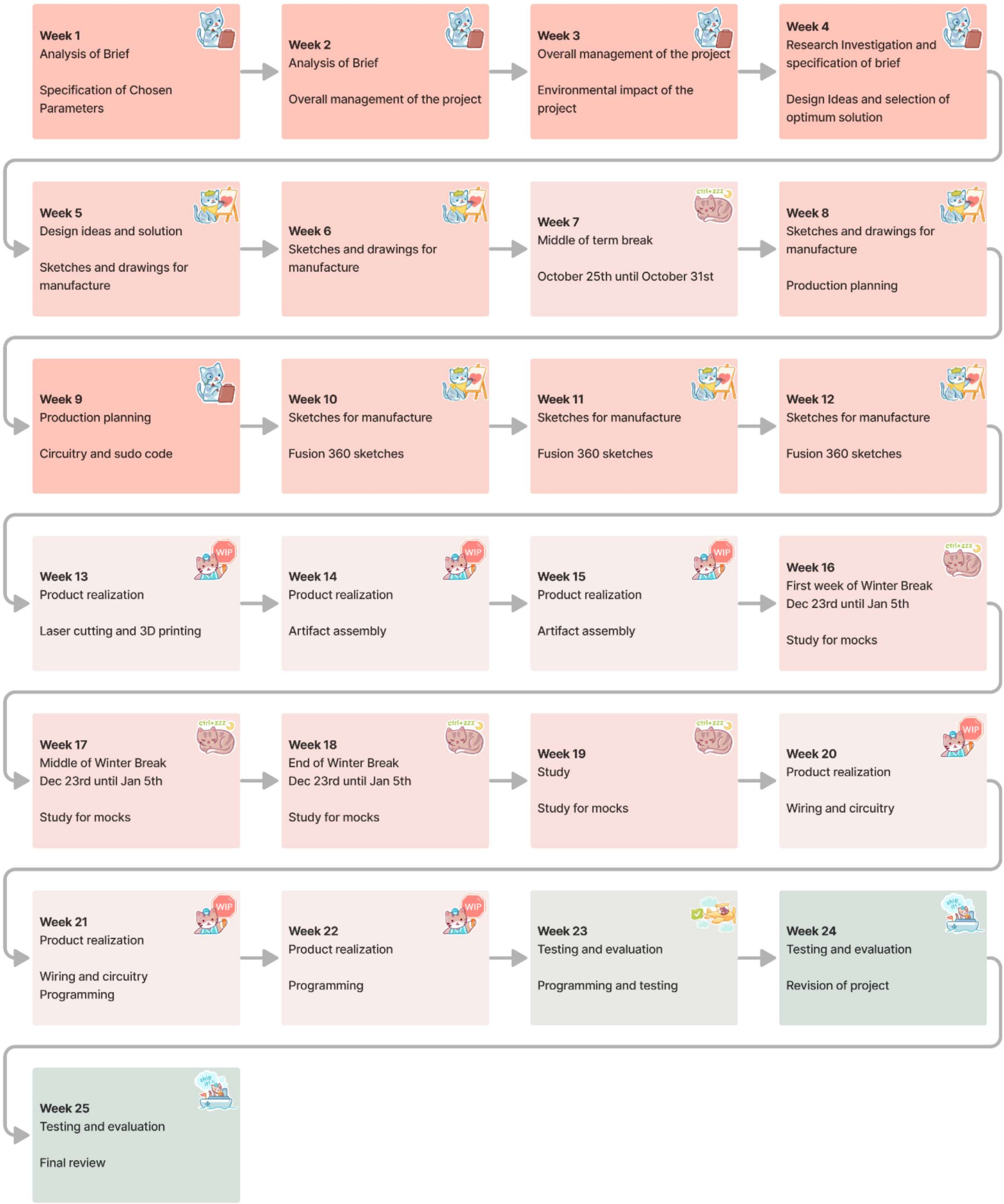
Gantt Chart:

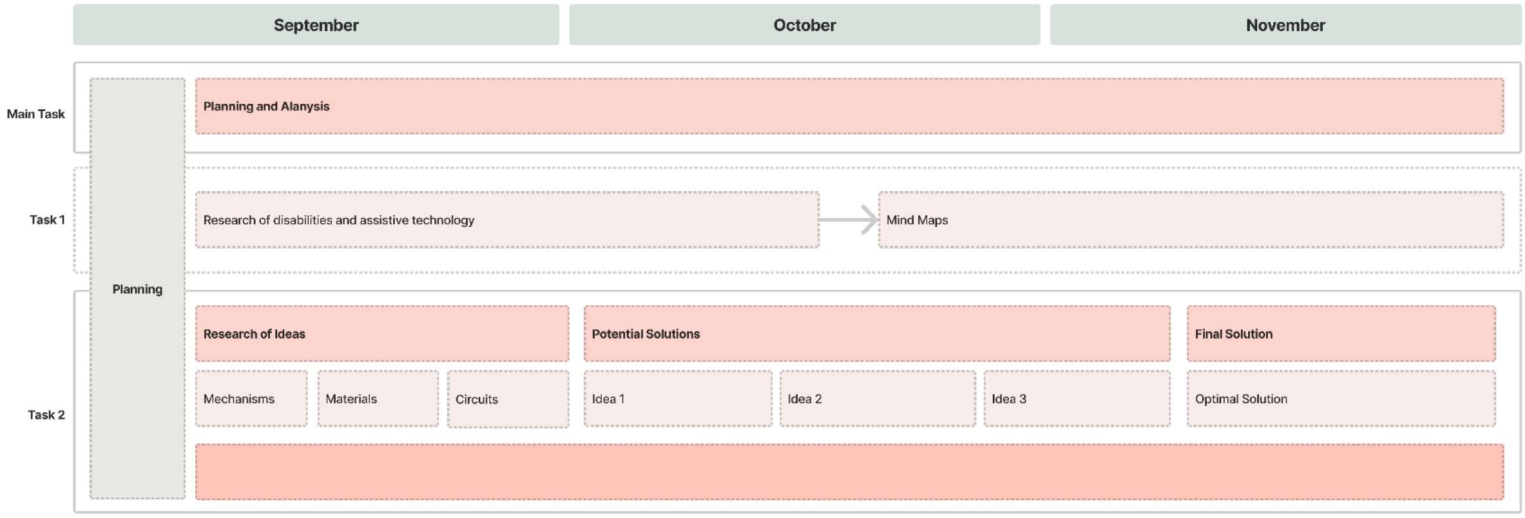


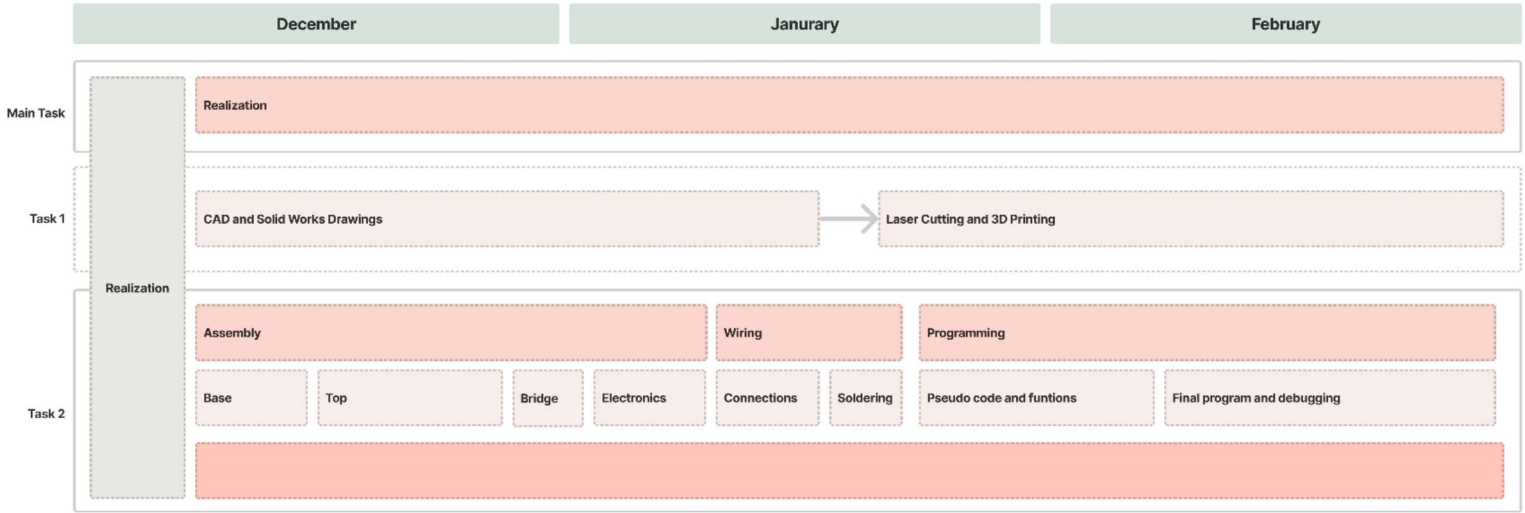
Legend:

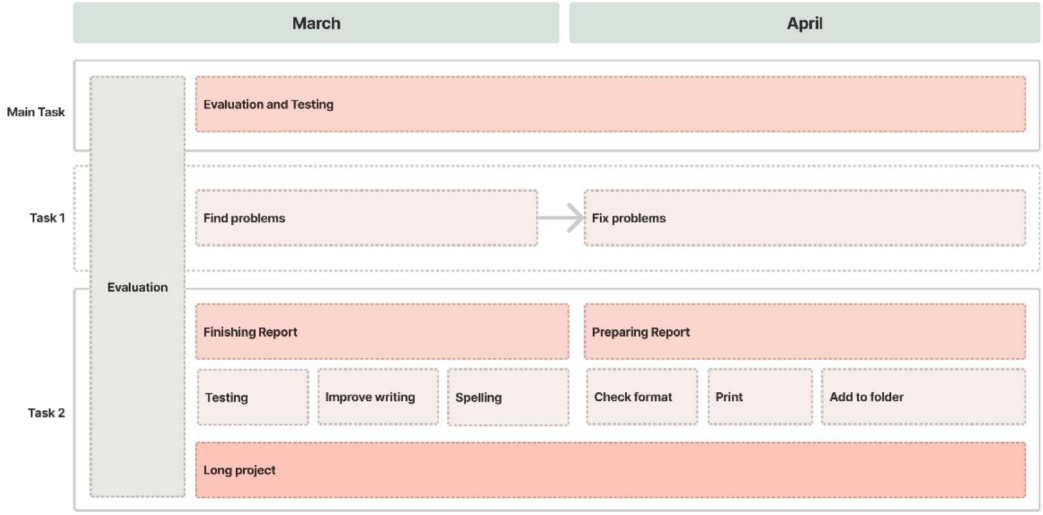
Color	Meaning
	Analysis of Thematic Brief
	Analysis of Thematic Brief
	Research, Investigation & Specifications
	Design Ideas and Selection of Solution
	Sketches and Drawings for Manufacture
	Environmental Impact
	Production Planning
	Product Realization
	Evaluation & Critical Reflection

Resources: I have access to a fully equipped LC Technology room. Machine tools available to me include: drills, laser cutter, sanding belt and spray paint. I also have access to a wide range of plastics, metals and woods. I have to pay for some of the materials and components that I use and would like to keep costs low or reuse components that I already have. I also have to consider my own skill level. I must be able to make my final solution in the Technology classroom. If I make the design too difficult it will be hard to complete within the timeframe. The project folder must be completed within the same timeframe so I will need to allocate some of my class time to documenting my work on a PC but I may do some of the write up on my iPad in free classes or at home.


Week 1
Analysis of Brief
Specification of Chosen Parameters
Week 2
Analysis of Brief
Overall management of the project
Week 3
Overall management of the project
Environmental impact of the project
Week 4
Research Investigation and specification of brief
Design Ideas and selection of optimum solution
Week 5
Design ideas and solution
Sketches and drawings for manufacture
Week 6
Sketches and drawings for manufacture
Week 7
Middle of term break
October 25th until October 31st
Week 8
Sketches and drawings for manufacture
Production planning
Week 9
Production planning
Circuitry and sudo code
Week 10
Sketches for manufacture
Fusion 360 sketches
Week 11
Sketches for manufacture
Fusion 360 sketches
Week 12
Sketches for manufacture
Fusion 360 sketches
Week 13
Product realization
Laser cutting and 3D printing
Week 14
Product realization
Artifact assembly
Week 15
Product realization
Artifact assembly
Week 16
First week of Winter Break
Dec 23rd until Jan 5th
Study for mocks
Week 17
Middle of Winter Break
Dec 23rd until Jan 5th
Study for mocks
Week 18
End of Winter Break
Dec 23rd until Jan 5th
Study for mocks
Week 19
Study
Study for mocks
Week 20
Product realization
Wiring and circuitry
Week 21
Product realization
Wiring and circuitry
Programming
Week 22
Product realization
Programming
Week 23
Testing and evaluation
Programming and testing
Week 24
Testing and evaluation
Revision of project
Week 25
Testing and evaluation
Final review


September
October
November
Main Task
Planning and Alanysis
Task 1
Research of disabilities and assistive technology
Mind Maps
Planning
Research of Ideas
Potential Solutions
Final Solution
Task 2
Mechanisms
Materials
Circuits
Idea 1
Idea 2
Idea 3
Optimal Solution


December
Janurary
February
Main Task
Realization
Task 1
CAD and Solid Works Drawings
Laser Cutting and 3D Printing
Realization
Assembly
Wiring
Programming
Task 2
Base
Top
Bridge
Electronics
Connections
Soldering
Pseudo code and funtions
Final program and debugging


March
April
Main Task
Evaluation and Testing
Task 1
Find problems
Fix problems
Evaluation
Finishing Report
Preparing Report
Task 2
Testing
Improve writing
Spelling
Check format
Print
Add to folder
Long project

Circuits

Micro-controllers

Board	Description	Language	Advantages	Disadvantages	Cost
RPI Pico	A tiny, fast, and versatile board built using RP2040.	MicroPython	Easy to program, low cost, fast, bread-board friendly.	Not a lot of resources.	€4.02
Micro: Bit	A pocket-sized computer that introduces micro-controllers.	Blocks	Lots of guides, easy code.	Slow, expensive, unreliable.	€19.32
Micro: Bit v2	A computer similar to the M: B with improved features.	Blocks	Built in sensors and speakers. Easy code.	Expensive, unreliable with IO.	€29.32
Arduino	A micro-controller based on easy-to-use hardware and software.	C++	Fastest, easy IO.	Big, expensive, hard to code.	€24.97
Arduino Nano	A small board based on the ATmega328. Essentially a smaller Arduino.	C++	Very small, fast.	Hard to code, can be unreliable.	€11.80

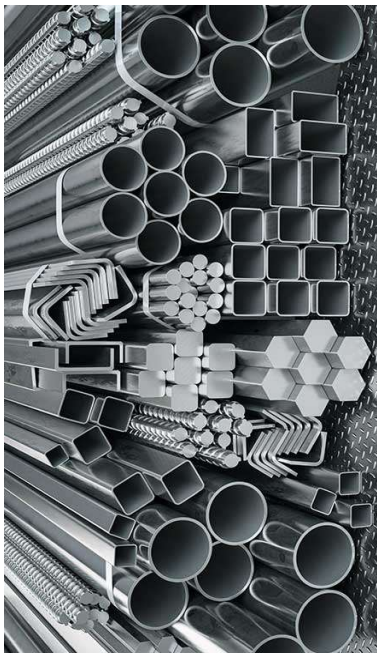
Components

Inputs	Description	Outputs	Description
Button	Push on or off input.	LED	Light up in a certain color when powered.
Switch	Clicked on or off, open or close a circuit.	DC Motor	Rotates when powered.
Encoder	Outputs direction and clicks on or off.	LCD Display	Displays characters programmed on controller.
LDR	Outputs light level by resistance.	Buzzer	Buzzed when powered.
PIR	Proximity sensor.	Servo Motor	Rotate up to 180 to specified angle.
Thermistor	Temperature sensor.	Stepper Motor	Rotate 360 a specified number of steps.

Materials

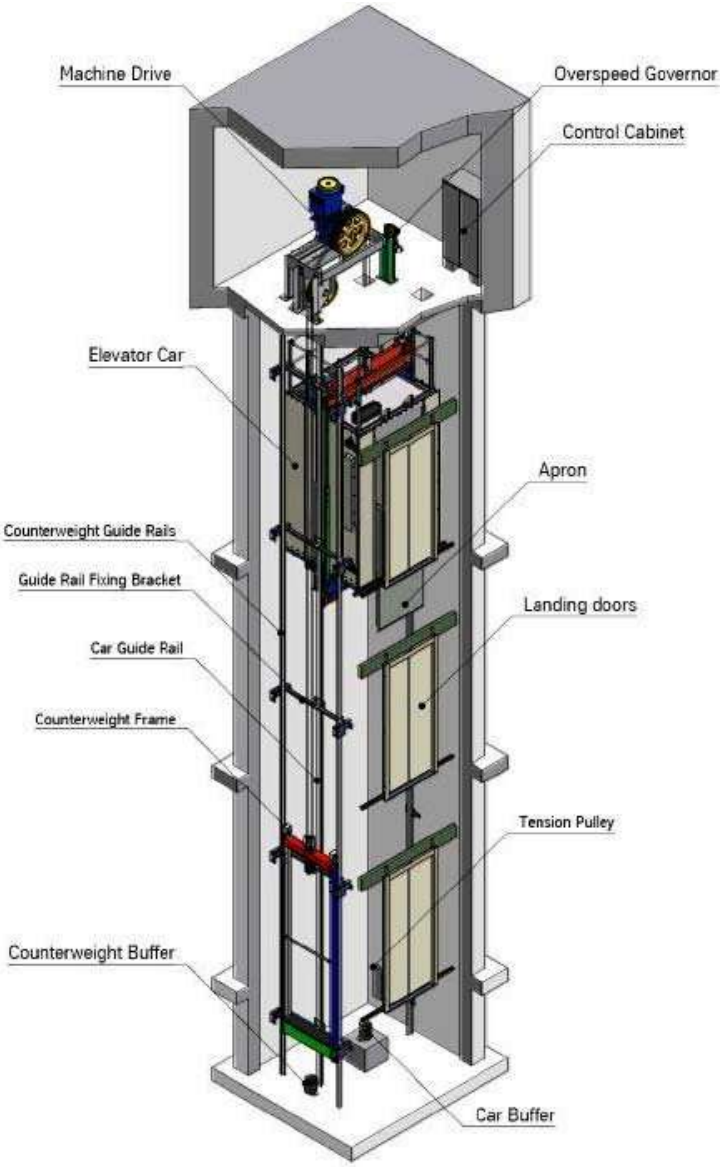
Materials

Material	Properties	Uses
Acrylic	Stiff, hard, scratches easily, good insulator, aesthetically pleasing.	Signs, covers, boxes.
ABS	Impact resistant, tough, heat resistant.	Lego bricks, car breaks, electronic casings.
PET	Good moisture barrier, light, strong, impact resistant.	Synthetic fibers, bottles, sails.
Wood	Strong, aesthetically pleasing, good workability, rigid or flexible.	Boxes, tables, stools, etc.
Manufactured boards	Cheap, strong, rigid, non brittle, environmentally friendly, good workability.	Crates, fences, construction, etc.
Steel	Cheap, strong, shiny, dense, hard, conductive.	Screws, nuts, bolts, tools.
Aluminum	Cheap, strong, flexible, ductile, malleable, lightweight.	Aircrafts, wires, cans.



Mechanisms

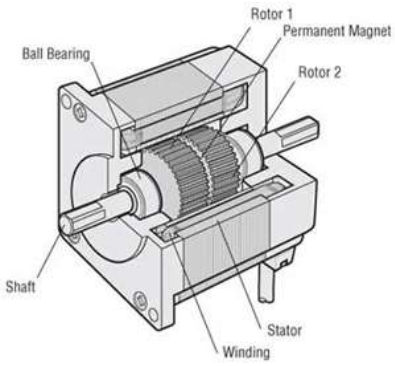
Linear Motion



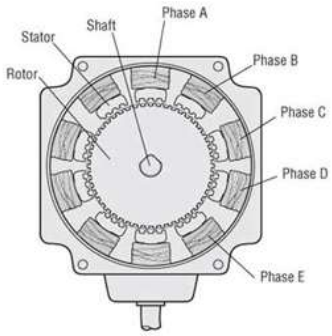
Linear motion can be created using: a rack and pinion, a pulley system, and oscillating arms. In context to my brief I found that the most prominent use of this mechanism was in access lifts. These lifts include, stair lifts to assist use of the stairs, floor lifts for wheelchairs and even pool lifts for swimming pools.

Mechanisms

Rotary Motion



Motor Structural Diagram: Cross-Section Parallel to Shaft



Motor Structural Diagram: Cross-Section Perpendicular to Shaft

Rotary motion can be easily achieved by use of a DC motor or stepper motors. This motion can be modified and translated using pulleys or gears.

In context to my brief I found that the most prominent use of this mechanism was in wheelchairs and transport technologies.
These included mobility scooters, electric wheelchairs and motorized ramps that swing outwards when needed.

Design

Idea 1

Smart Room

Idea 1

Smart Room

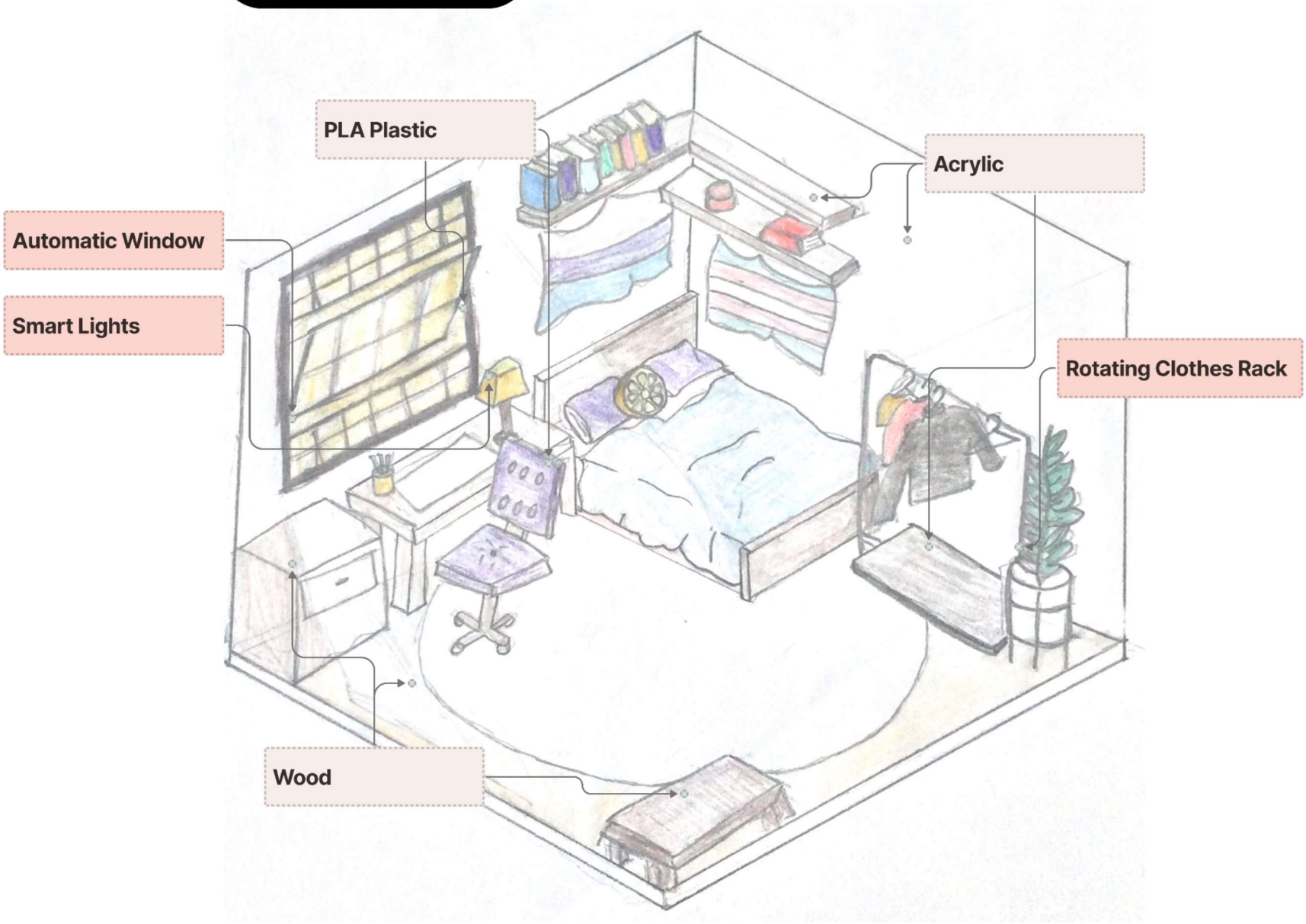
My first idea is a smart room for someone that has difficulties with mobility.
I would include a number of small features, these include:

Feature	**Description**	**Justification**
Automatic Window	A window connected to a motor that will automatically open or close depending on the air quality of the room.	Windows are often in hard to reach places, with high handles.
Rotating Clothes Rack	A clothes rack that can be rotated by the user, this will bring the item they want to wear to the front, making it easily accessible and low down.	Clothes in wardrobes can be hard to grab and people may not get the wanted item.
Smart Lights	Lights controlled by a smart switch that can be activated by a voice command or by a motion sensor.	It may require extra effort for someone with mobility difficulty to turn on the light.

Advantages:
I can include lots of features.
Shows manual and automatic inputs and outputs.

Disadvantages:
Small scale.
Lots of small moving parts.
No everyday use for models so not environmentally friendly.

Smart Room


PLA Plastic
Acrylic
Automatic Window
Smart Lights
Rotating Clothes Rack
Wood

Design

Idea 2

Pill Dispenser

Idea 2

Pill Dispenser

My second idea is an automatic pill dispenser for people who may have difficulty remembering medication or taking it out of the packet.

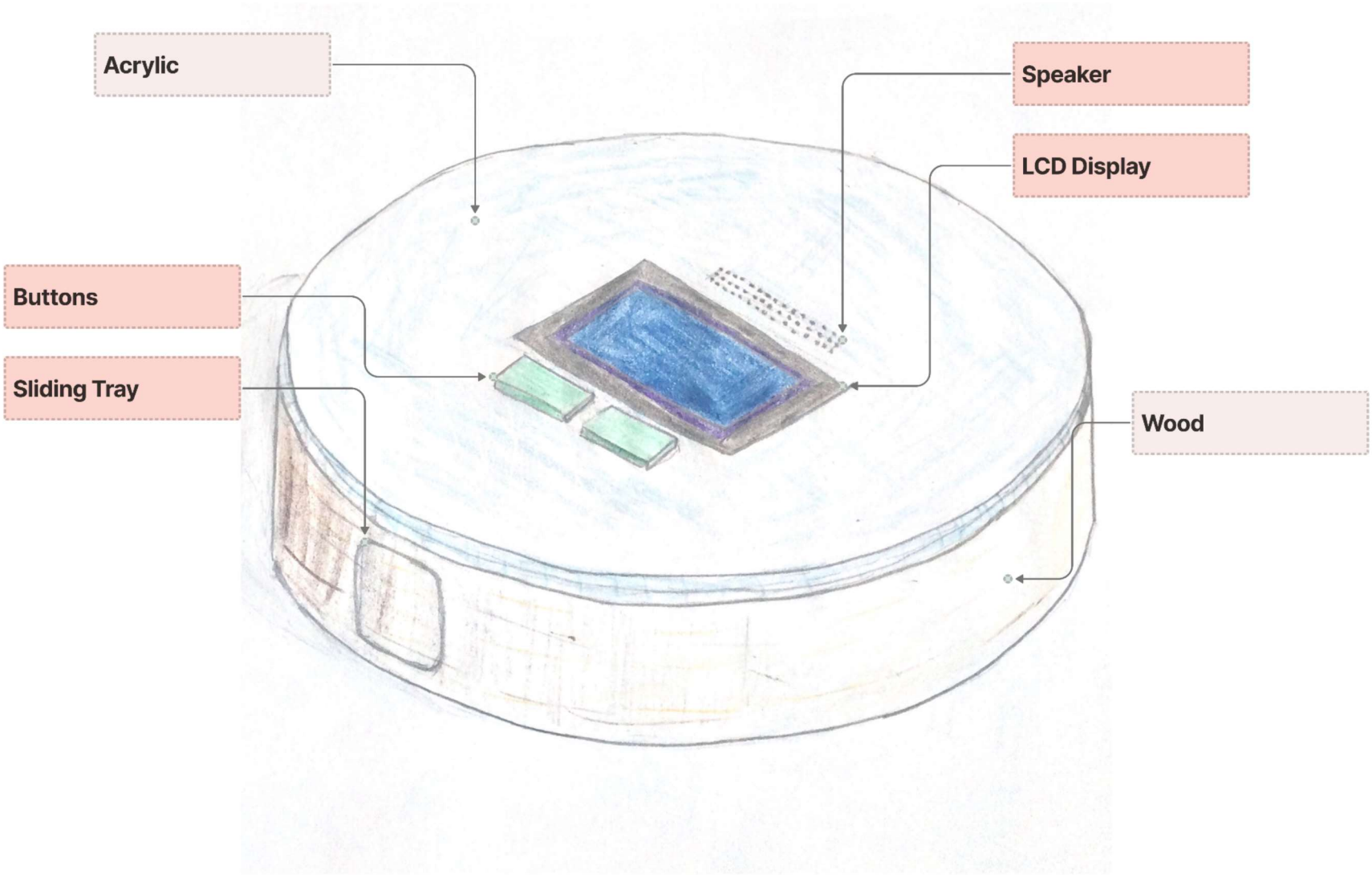
I would include a number of different features, these include:

Feature	**Description**	**Justification**
Buttons	A way for the user to release the medication or snooze the timer.	So the system can turn off the buzzer and know to go to the next timer.
LCD display	A screen to let the user get information about what is happening. Time and what time is the next medication.	So the user can see when they have to take the next medication.
Sliding tray	A tray that each pill will drop into so the user can slide it out.	To stop the user from taking the medication at the wrong time and the pills from falling out.

Advantages:
Everyday use.
Coding and mechanical elements.
Primary research from pharmacy.

Disadvantages:
May be difficult to operate.
Time keeping and clocks may lose time over large periods with Pico.

Pill Dispenser



Design

Idea 3

Smart Planter

Idea 3

Smart Planter

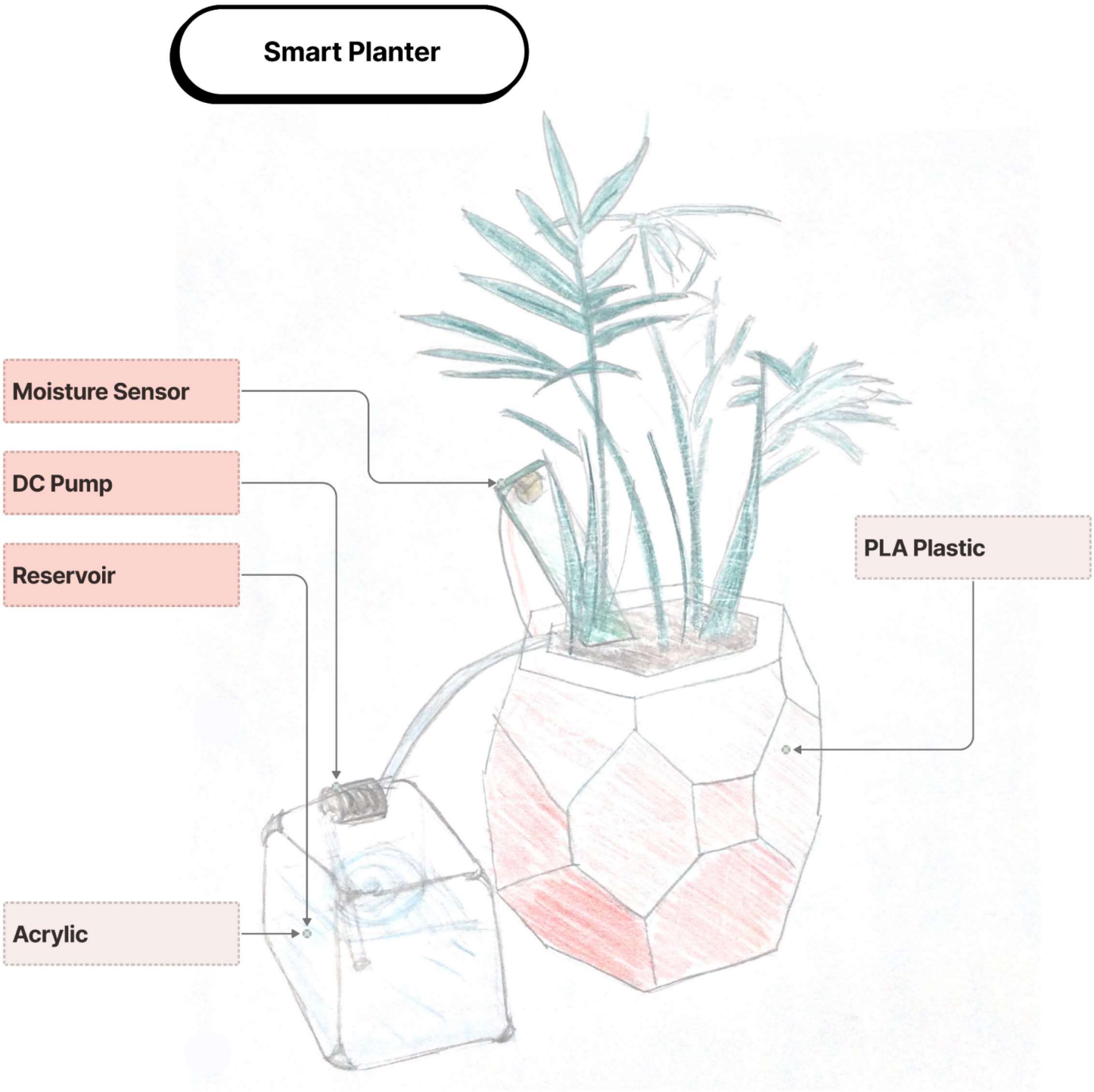
My third idea is a smart planter for those who have trouble watering their plants. This would help people that might have trouble reaching their plants, pouring the water or remembering to water them.

I would include a number of different features, these include:

Feature	**Description**	**Justification**
Soil Moisture	A soil moisture probe, placed in the soil, tells the program when water is needed.	This is important to tell the system the humidity. It is better than a timer.
DC Pump	A DC pump placed in the reservoir, connected around the plant pot to evenly deliver water.	This method is the easiest and most reliable method of water delivery.
Large Reservoir	A large water reservoir that the user fills up.	There should be a large tank of water as somebody can help do it every once and a while.

Advantages:
Plants are proven to improve mental health.
Easy to make.

Disadvantages:
May not fit the project brief very well.
Limited features.
Water and electronics may not be safe.


Smart Planter
Moisture Sensor
DC Pump
Reservoir
Acrylic
PLA Plastic

Optimum Solution

Pill Dispenser

My final solution is an automatic pill dispenser for people who may have difficulty remembering medication or taking it out of the packet.
From my experience working in a local pharmacy I have noticed that a lot of patients have problems both opening their medication packets and remembering what to take and when.
I feel like this project has everyday use and solves a really big problem for people.

I believe that this project relates to more than one area of disability that I have researched.
My project can help the area of organization, helping those with dementia or Alzheimer's , but also the area of movement, helping people that struggle with fine motor movement or Parkinson's.

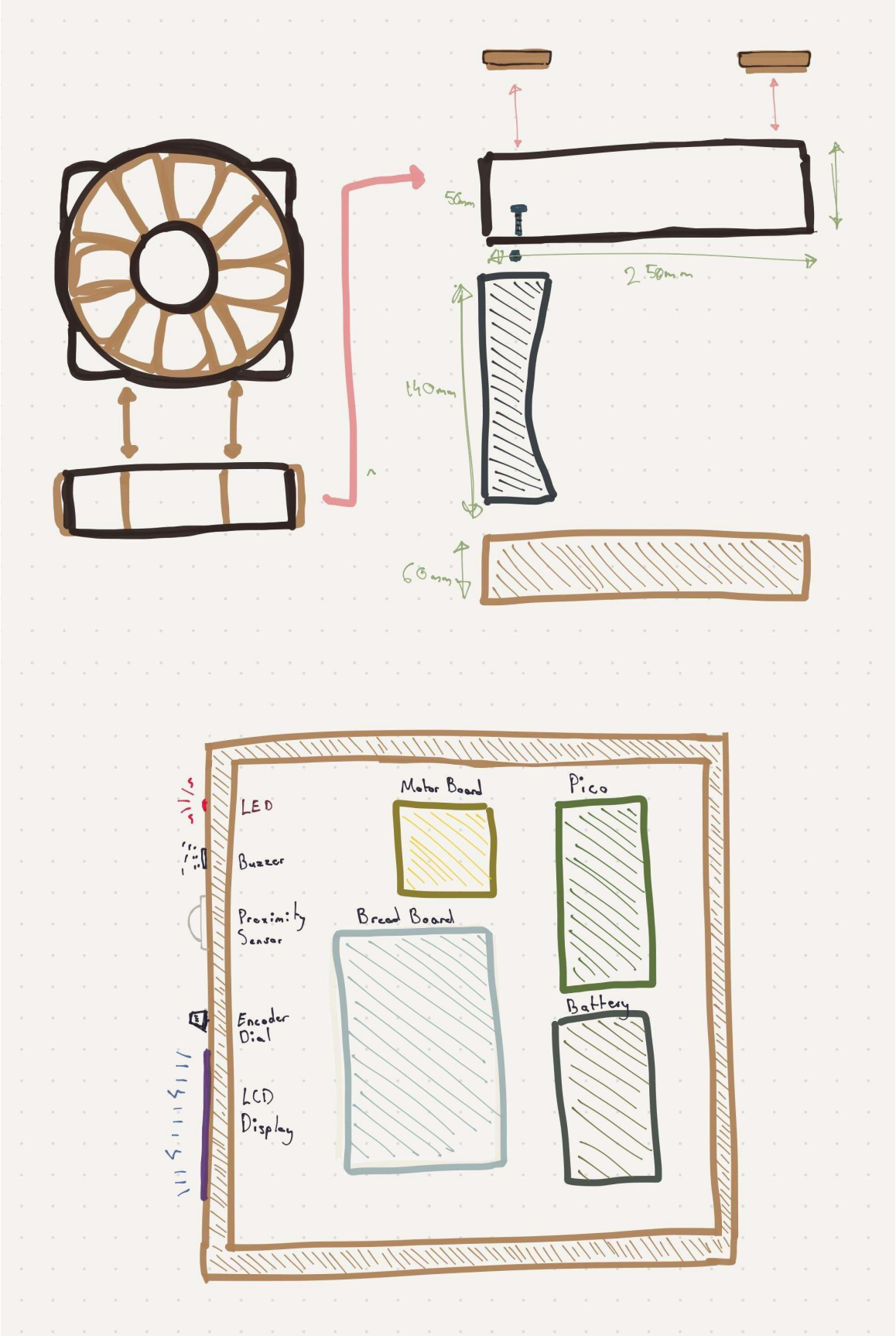
Another reason I chose to make this project is because of the environmental impact. I will build a working full-scale usable project so that after the examination the project can be used.

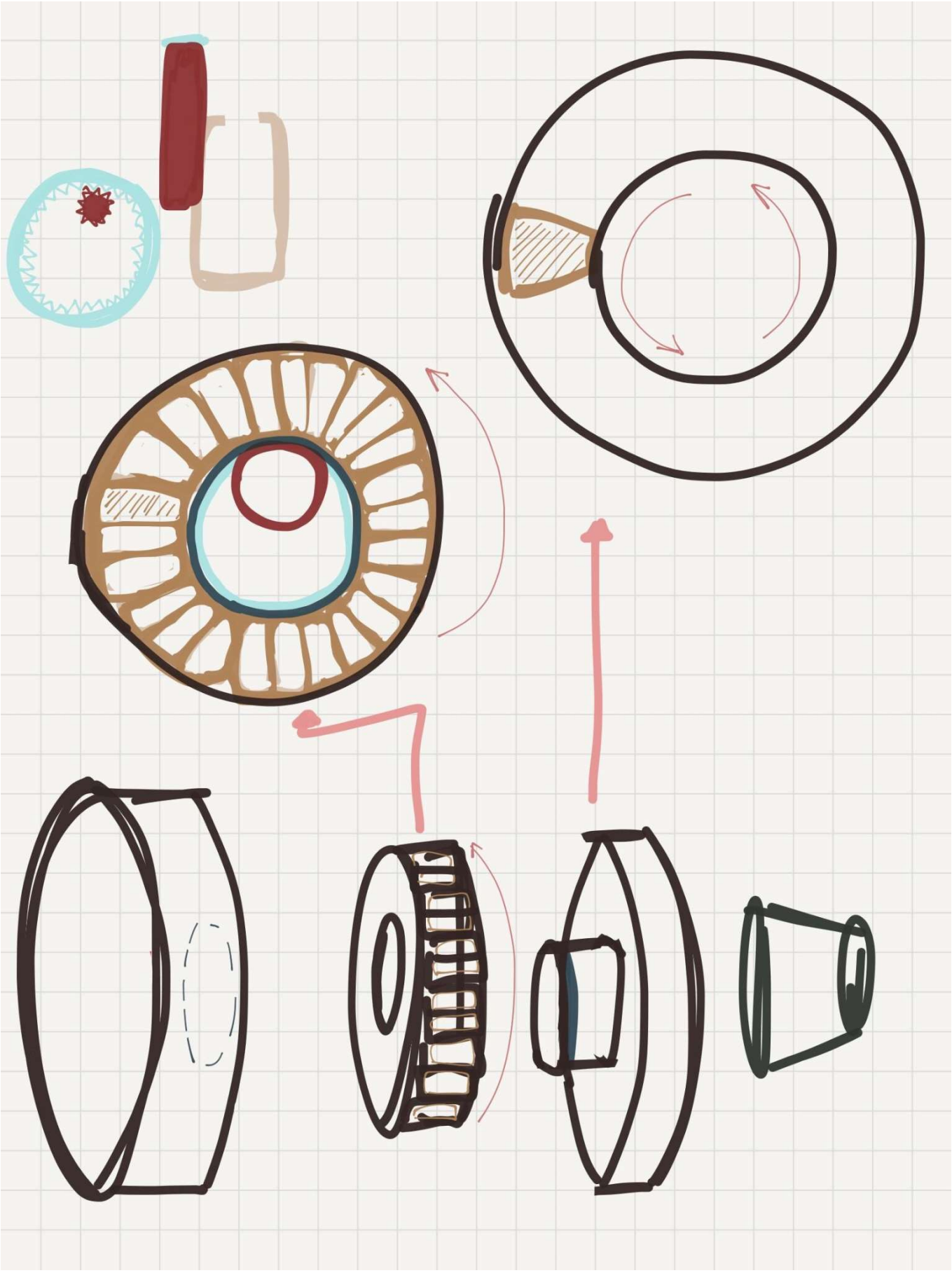
Overall, I feel that this solution clearly demonstrates what is asked in the brief about disability, is environmentally friendly and demonstrates my design, programming and electronics, and manufacturing skills that I have developed during my time in technology classes in school.

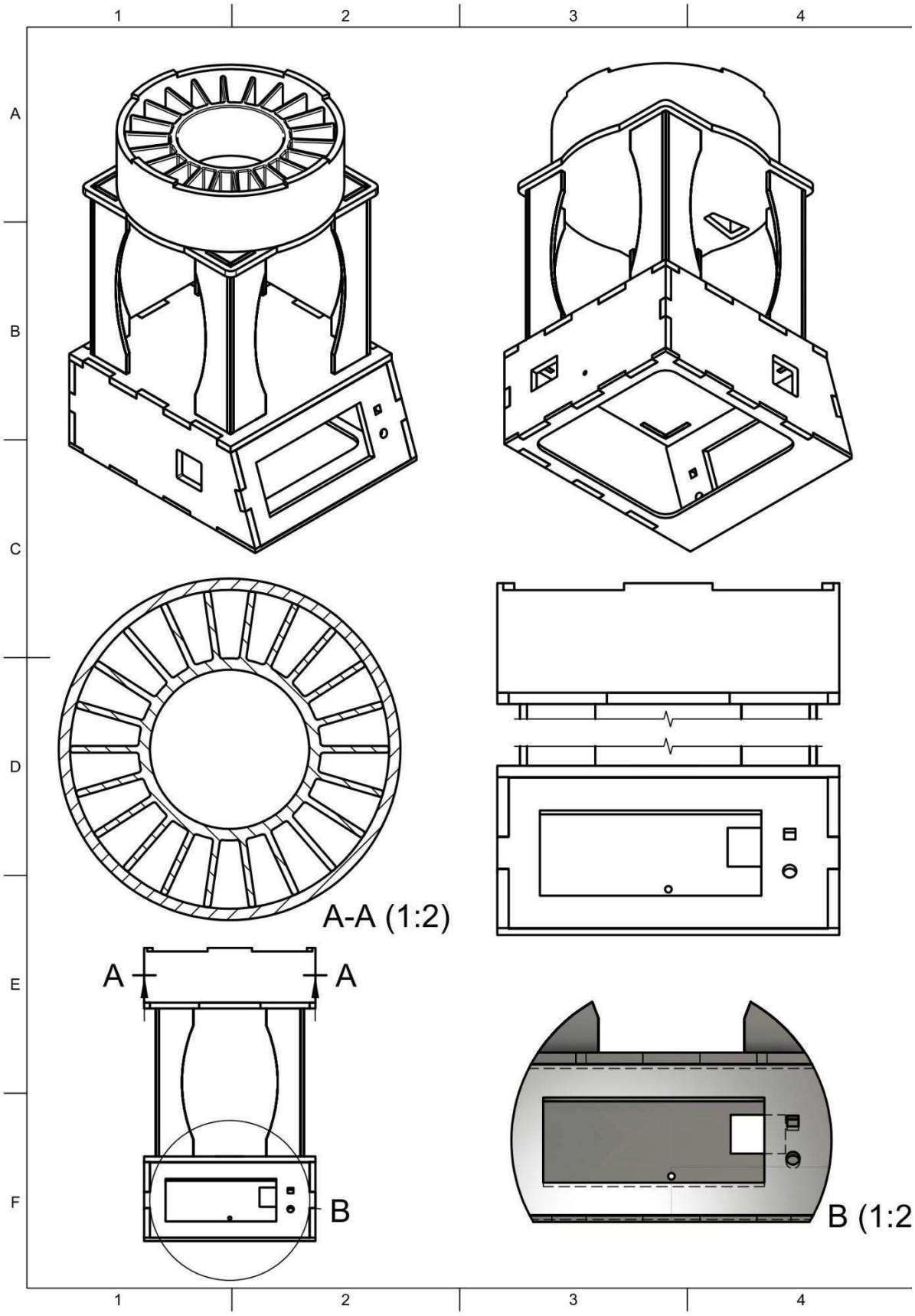
Features that I will include:

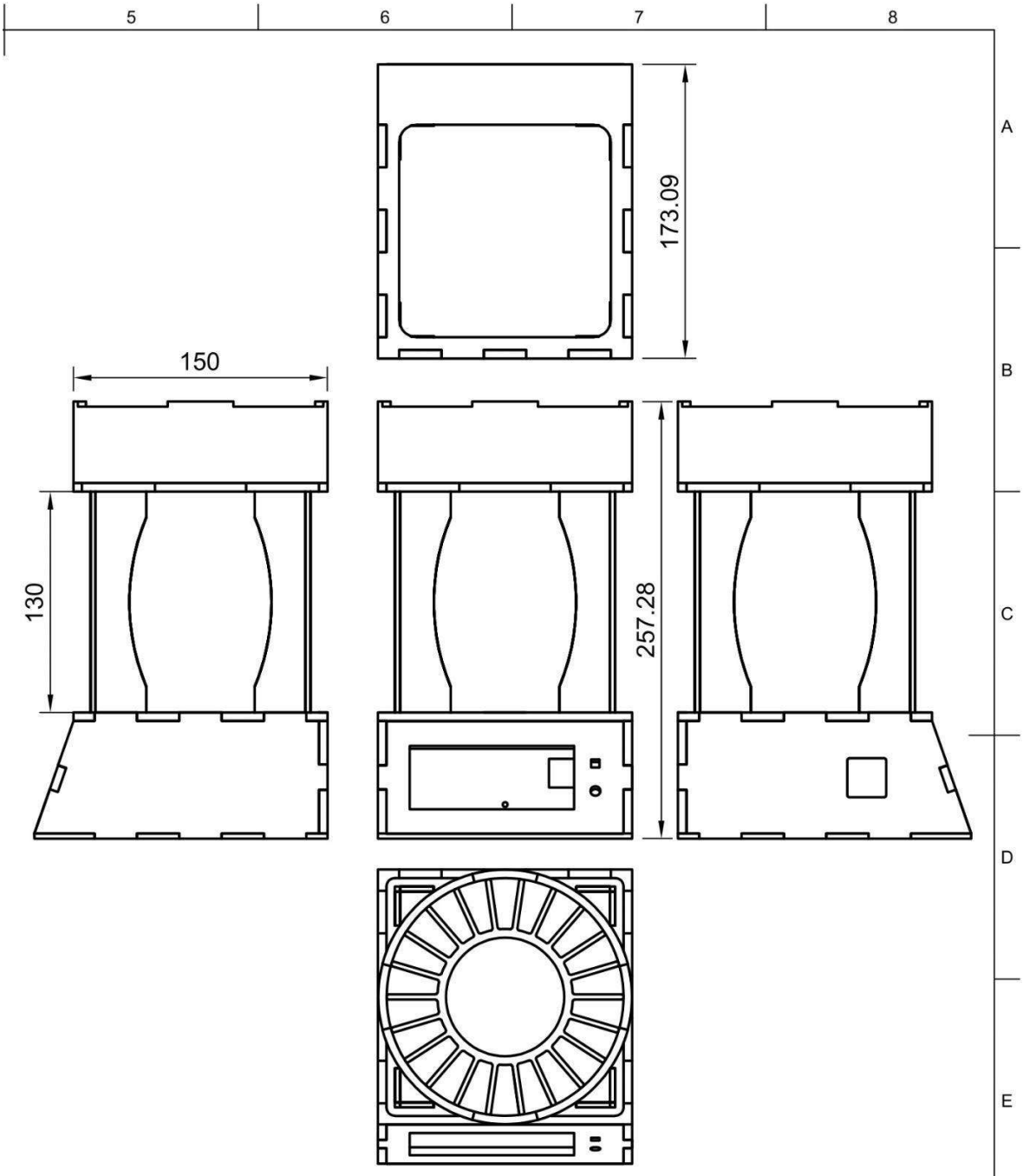
Feature	Description	Justification
21 Spaces	Having 21 compartments means that the user can add a week's worth of medication, to be dispensed morning, day and night.	This allows the user or the nurse to fill up a week from a 'blister pack'.
Buzzer	A buzzer makes a sound when the medication is dispensed in order to notify the user.	This is important if the user does not see the flashing light.
LCD Display	A screen to let the user control everything about the device and get information about what is happening.	This lets the user change settings without needing a computer.
Dial	A rotary encoder to control the menu of the LCD.	The device needs an input.
PIR Sensors	A proximity sensor to stop beeping when it detects movement in the room and turn the screen off when there is no movement.	Adds functionality to the system. Environmentally friendly.
Breadboard	A breadboard to do all the wiring. Allows parts to be easily changed in the future if they break.	For easy repair and to make it environmentally friendly.
LED	A light built into the front of the device to flash when the medication is dispensed in order to notify the user.	This is useful to alert hard of hearing users.
Software settings	Having settings programmed into the software to allow the user to modify all the variables of the device, including: time, dispense times, audio levels, light levels, stepper steps, etc.	This is important because the user must be able to change information to reflect what they need.

Sketches


55mm
250mm
140mm
60mm
LED
Buzzer
Proximity Sensor
Encoder Dial
LCD Display
Motor Board
Pico
Bread Board
Battery




1
2
3
4
A
B
C
D
E
F
A-A (1:2)
A
A
B
B (1:2



Dept.	Technical reference	Created by		Approved by	
	Technology	111392	10/01/2021	111392	12/01/2021
		Document type		Document status	
				Complete	
		Title		DWG No.	
		Assemby		111392	
				Rev. / Date of issue	Sheet
					1/1

Environmental Impacts

During the designing and development of phases of my project I knew it was very important to look out for the environmental impacts and make sure that it was manufactured in a sustainable way.

Some of the steps I will take during my project include:

Using CAD tools such as Fusion 360 and Solid works.
I will use CAM such as laser cutting and 3D printing to increase accuracy and reduce any waste that I might create if I were to make by hand or make mistakes.

Using digital design software such as GIMP and Gravity Sketch to quickly create and modify initial designs to save paper.

I will use a breadboard instead of permanently soldering components so they can easily be **replaced** and **reused**.

I will finish parts by using hand tools to **reduce energy consumption** caused by power tools.

Another important step to making sure that this project is sustainable is **recycling**. I will make sure to find materials to **reuse** in my artifact to reduce material waste.
This includes reusing components such as switches and buttons along with larger materials such as wood and acrylic.

During laser cutting I will work with my friends to see where we can work together and use less material overall. We will fit our CAD parts together where possible to squeeze as many parts as possible out of the acrylic sheets for laser cutting.

During 3D printing I will use Octoprint, an open-source project that runs on a Raspberry Pi, to monitor my prints to detect possible failures that would waste plastic for printing.
I would like to use PET plastic for printing as it is made from recycled plastic bottles and is a safe plastic.

I plan to print my portfolio on recycled paper to make my document environmentally friendly.

Parts List

Laser Cutting List

Name	Material	Thickness	Size	Count
Base Bottom	Smokey Acrylic	3mm	173.094mm x 150.0mm	1
Base Back	Black Acrylic	5mm	50.0mm x 150.0mm	1
Base Front	Black Acrylic	5mm	46.188mm x 150.0mm	1
Base Left Side	Black Acrylic	5mm	50.0mm x 173.094mm	1
Base Right Side	Black Acrylic	5mm	50.0mm x 173.094mm	1
Base Top	Black Acrylic	5mm	150.0mm x 150.0mm	1
Panel Short	Wood	3mm	140.0mm x 30.0mm	4
Panel Long	Wood	3mm	140.0mm x 35.0mm	4
Top Cover	Smokey Acrylic	3mm	150.0mm x 150.0mm	1
Top Base	Black Acrylic	5mm	150.0mm x 150.0mm	1
Total	Acrylic		125150.0 $125150.0mm^2$	
Total	Wood		36400.0 $36400.0mm^2$	

3D Printing List

Name	Material	Thickness	Size	Price	Count
Top Inside	White PLA	40mm	140.0mm x 140.0mm	€1.06	1
Top Side Panel	White PLA	48mm	150.0mm x 150.0mm	€1.44	1

Total				€2.50	

Components List

Name	**Price**	**Count**
LCD Display	€13.11	1
Rotary Encoder	€2.15	1
Raspberry Pico	€4.02	1
Breadboard	€3.57	1
LED	€0.02	1
Stepper Motor	€4.42	1
Buzzer	€1.21	1
PIR Sensor	€2.19	1
DH11 Sensor	€2.54	1
Jumper Wires	€0.84	12
Total	€34.07	

Almost all components on the list will be re-used from my own previous projects so I will not have to buy any new components.

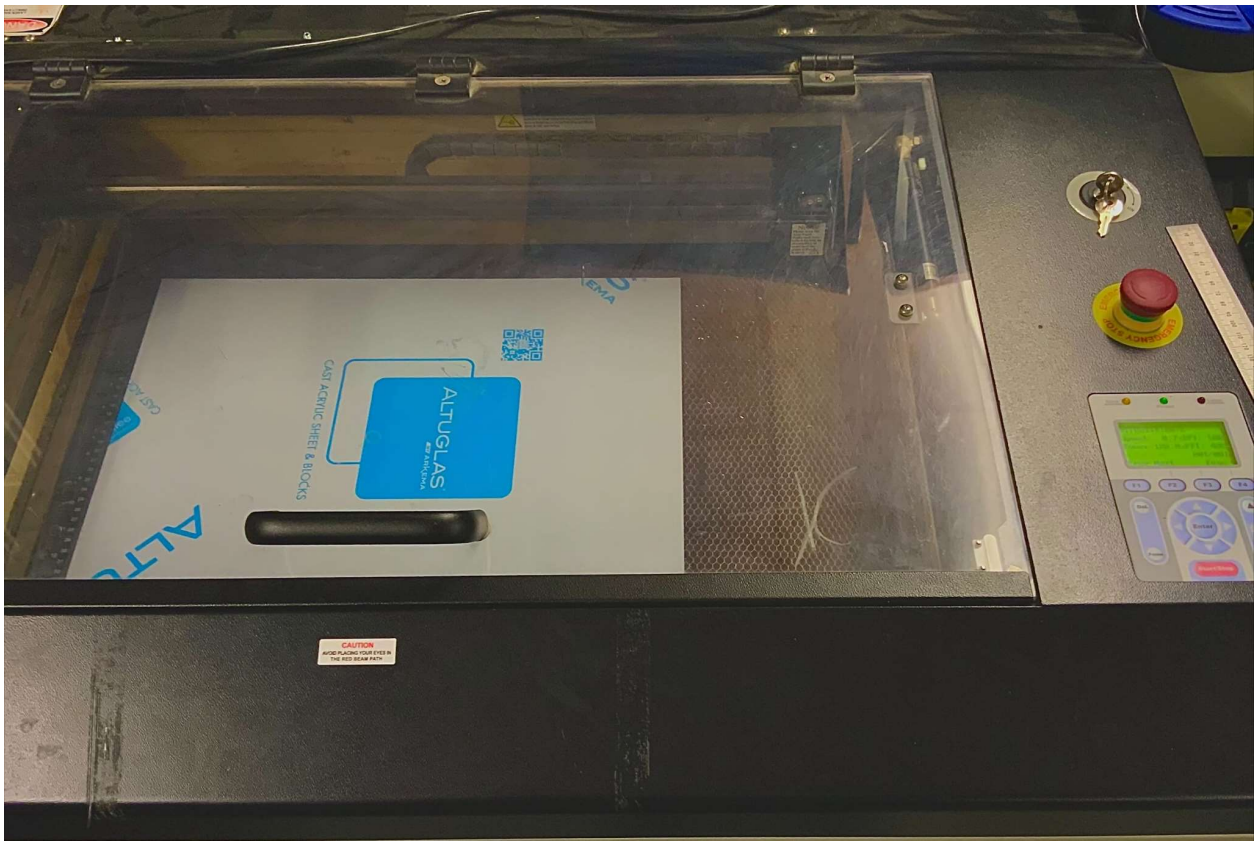
Realization

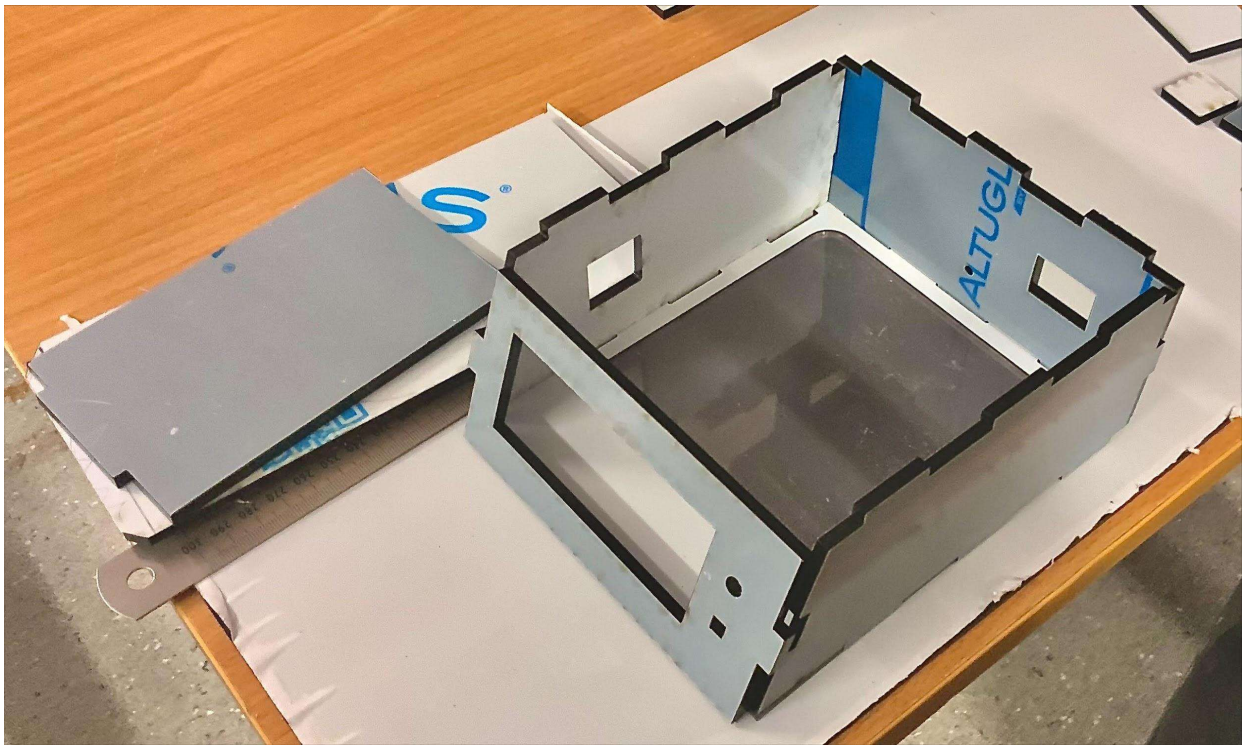
Laser Cutting

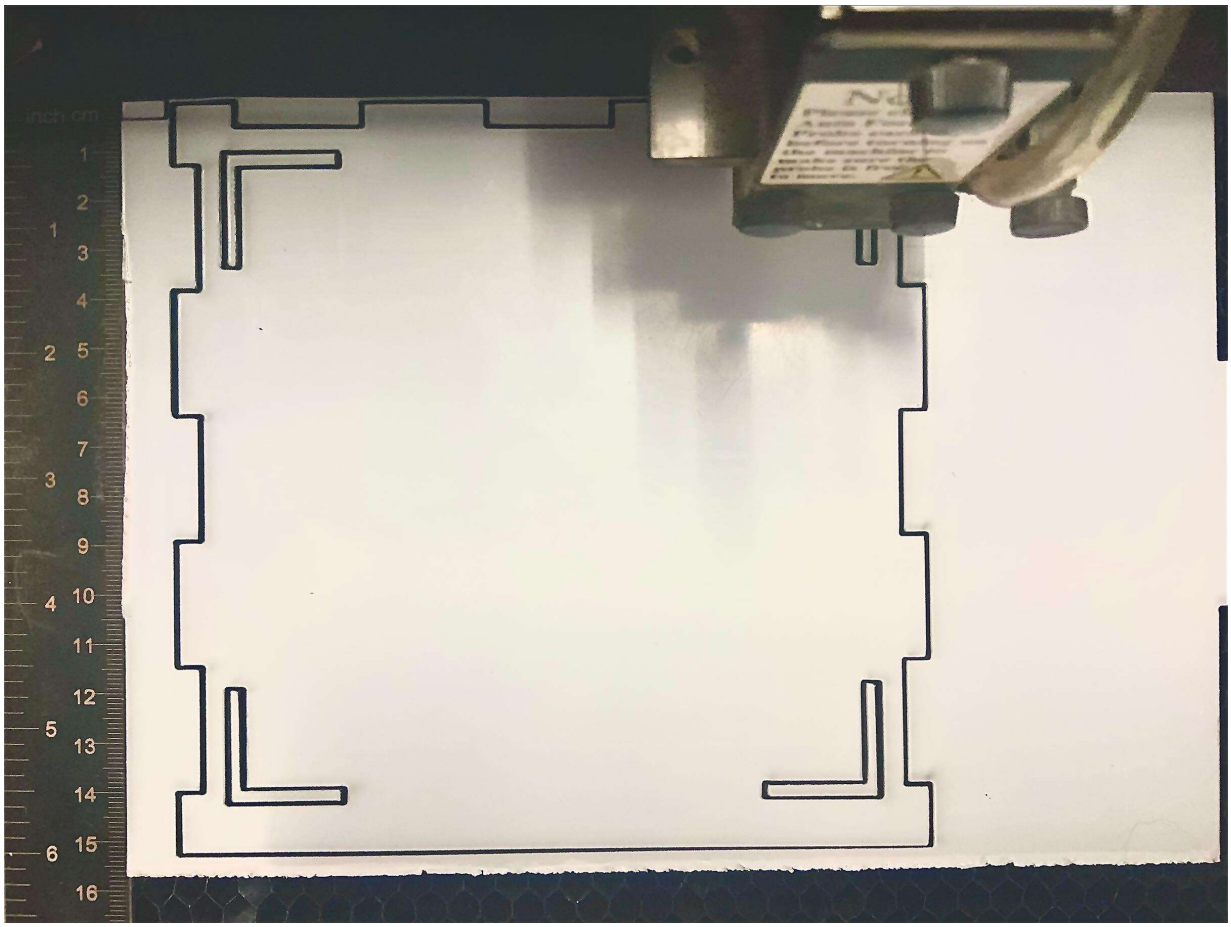
I used the laser cutter in this project for the majority of my pieces.
I chose to use the laser cutter because I wanted to make sure that this project was environmentally friendly, quick and waste free.

To make the parts I exported the sketches from Fusion 360 as DXF files. I then emailed them to my teachers account so I could open them on the school's laptop for laser cutting.
I downloaded the files and opened them in Coreldraw and set up the sizes in the canvas, making sure to reference my size sheet.

I made sure to double check the measurements, placement and power in order to avoid mistakes that could waste materials.


ALTUGLAS
CAST ACRYLIC SHEET & BLOCKS
CAUTION


ALTUGL



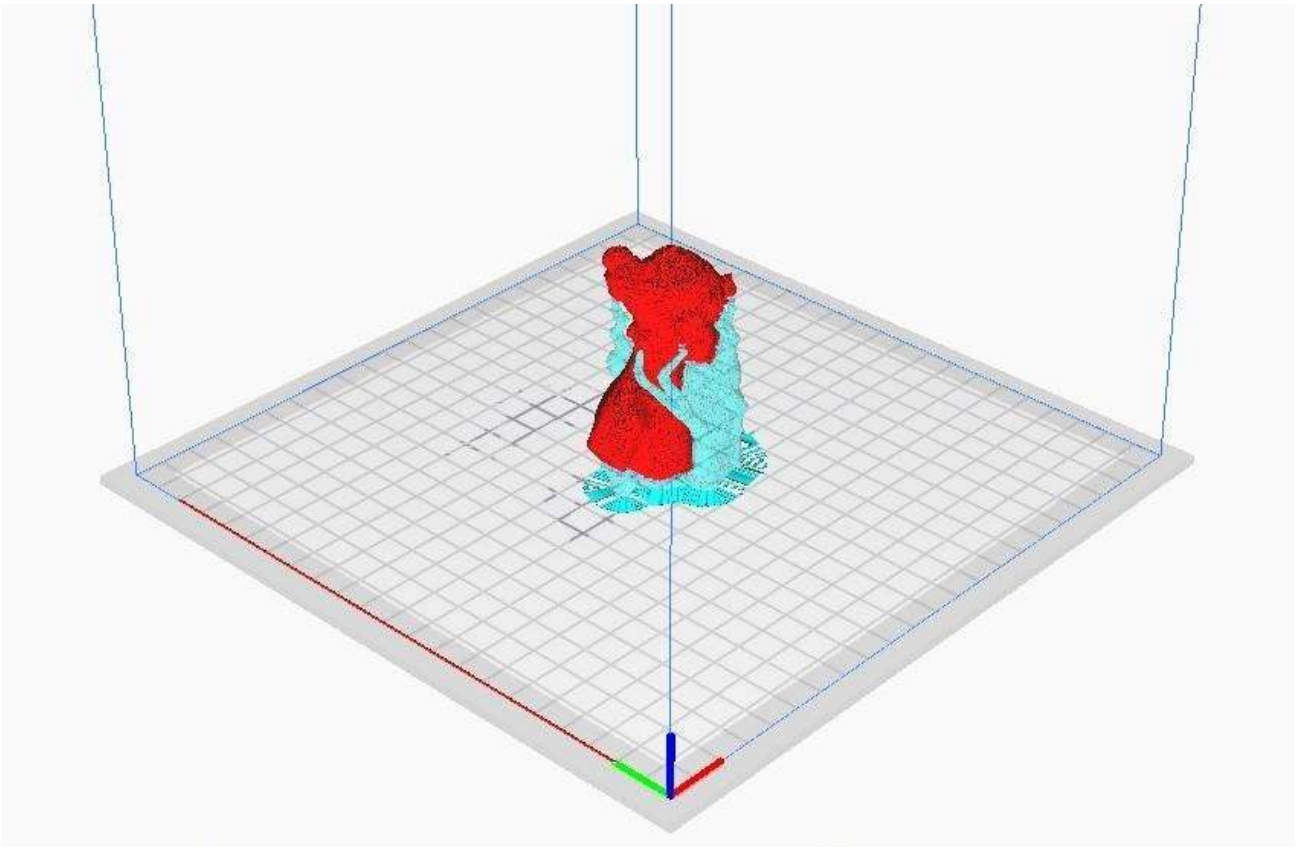
Realization

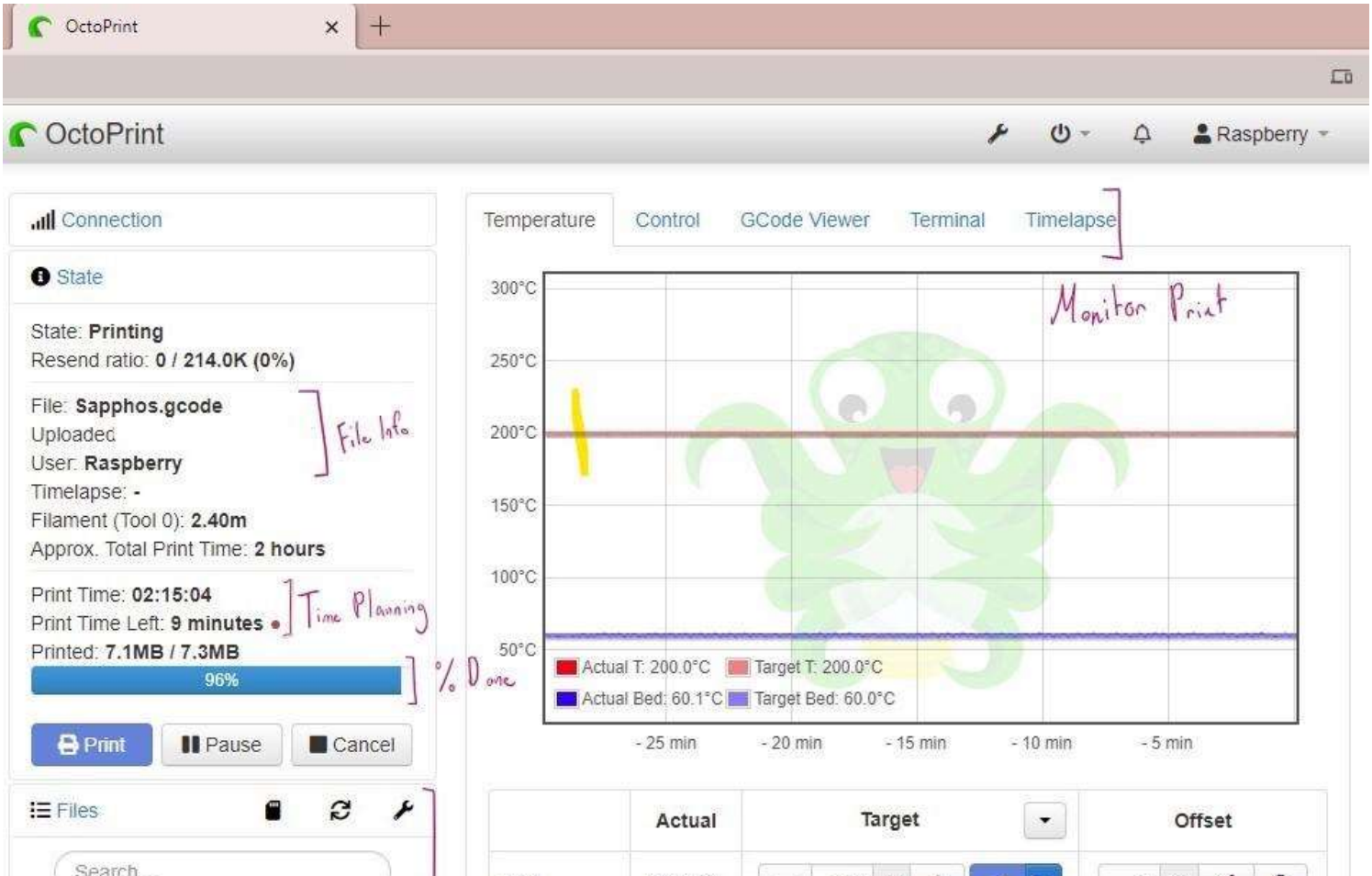
3D Printing

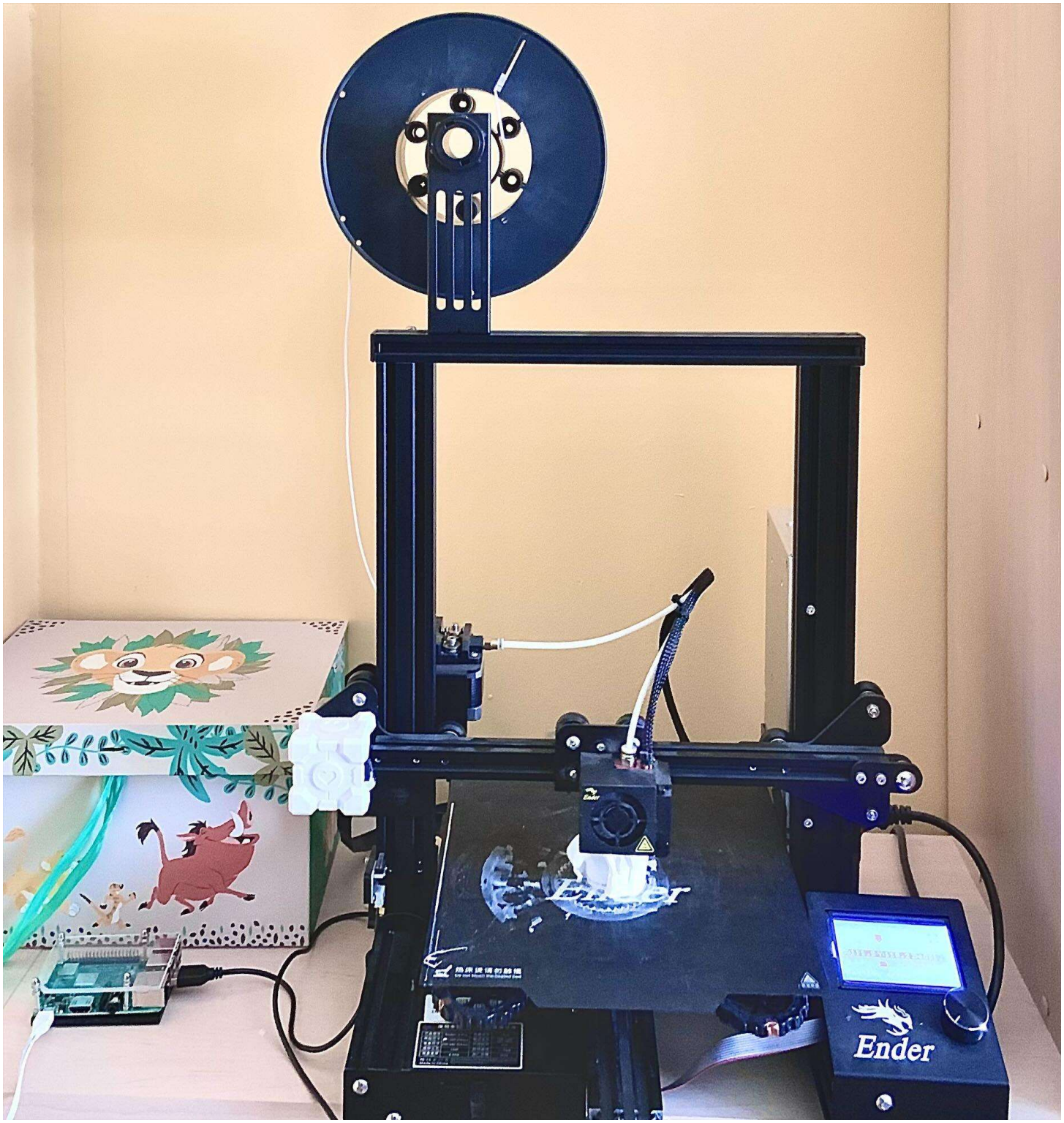
Another technology thatI used in this project was 3D printing. This process allowed me to easily create complicated or rounded shapes of any dimension that I need.
I used my 3D printer at home to print the parts, then I brought them into school to paint.

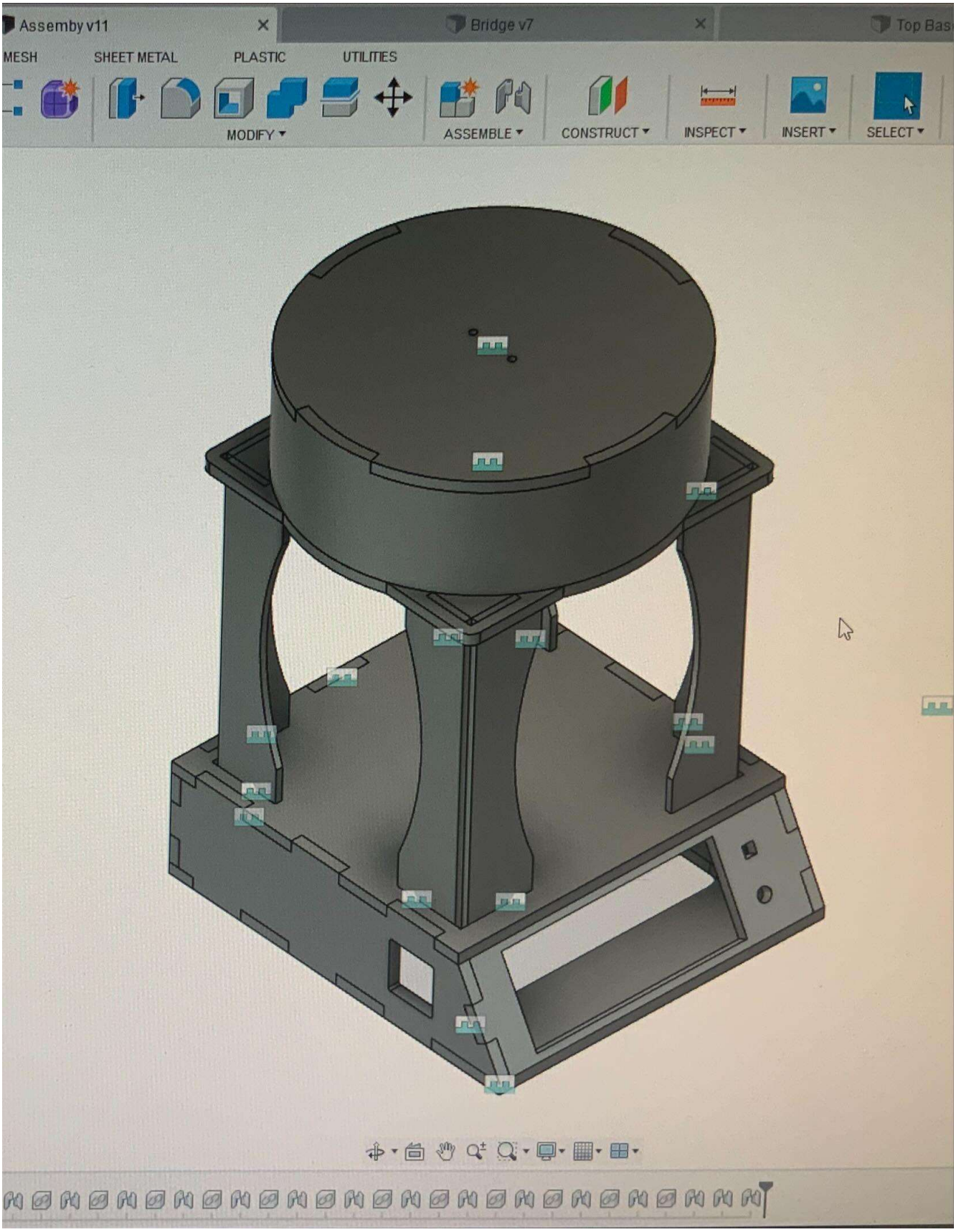
I exported the files from Fusion 360 and set them up for printing in Cura.

In order to make it safe and environmentally friendly I used a software called Octoprint, set up on a Raspberry Pi to monitor and control the prints as needed.



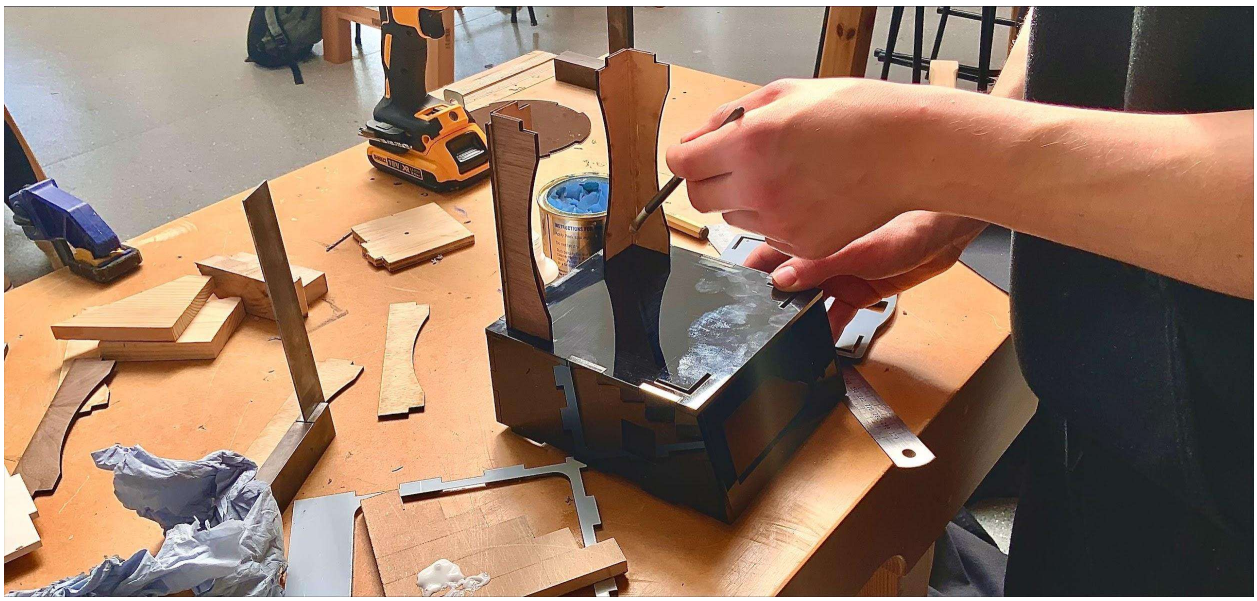

OctoPrint
OctoPrint
Raspberry
Connection
State
State: Printing
Resend ratio: 0 / 214.0K (0%)
File: Sapphos.gcode
Uploaded
User: Raspberry
Timelapse: -
Filament (Tool 0): 2.40m
Approx. Total Print Time: 2 hours
File Info
Print Time: 02:15:04
Print Time Left: 9 minutes
Printed: 7.1MB / 7.3MB
Time Planning
96%
% Done
Print
Pause
Cancel
Files
Temperature
Control
GCode Viewer
Terminal
Timelapse
Monitor Print
300°C
250°C
200°C
150°C
100°C
50°C
Actual T: 200.0°C
Target T: 200.0°C
Actual Bed: 60.1°C
Target Bed: 60.0°C
- 25 min
- 20 min
- 15 min
- 10 min
- 5 min
Actual
Target
Offset


Ender


Assemby v11
Bridge v7
Top Bas
MESH
SHEET METAL
PLASTIC
UTILITIES
MODIFY
ASSEMBLE
CONSTRUCT
INSPECT
INSERT
SELECT

Realization

Assembly



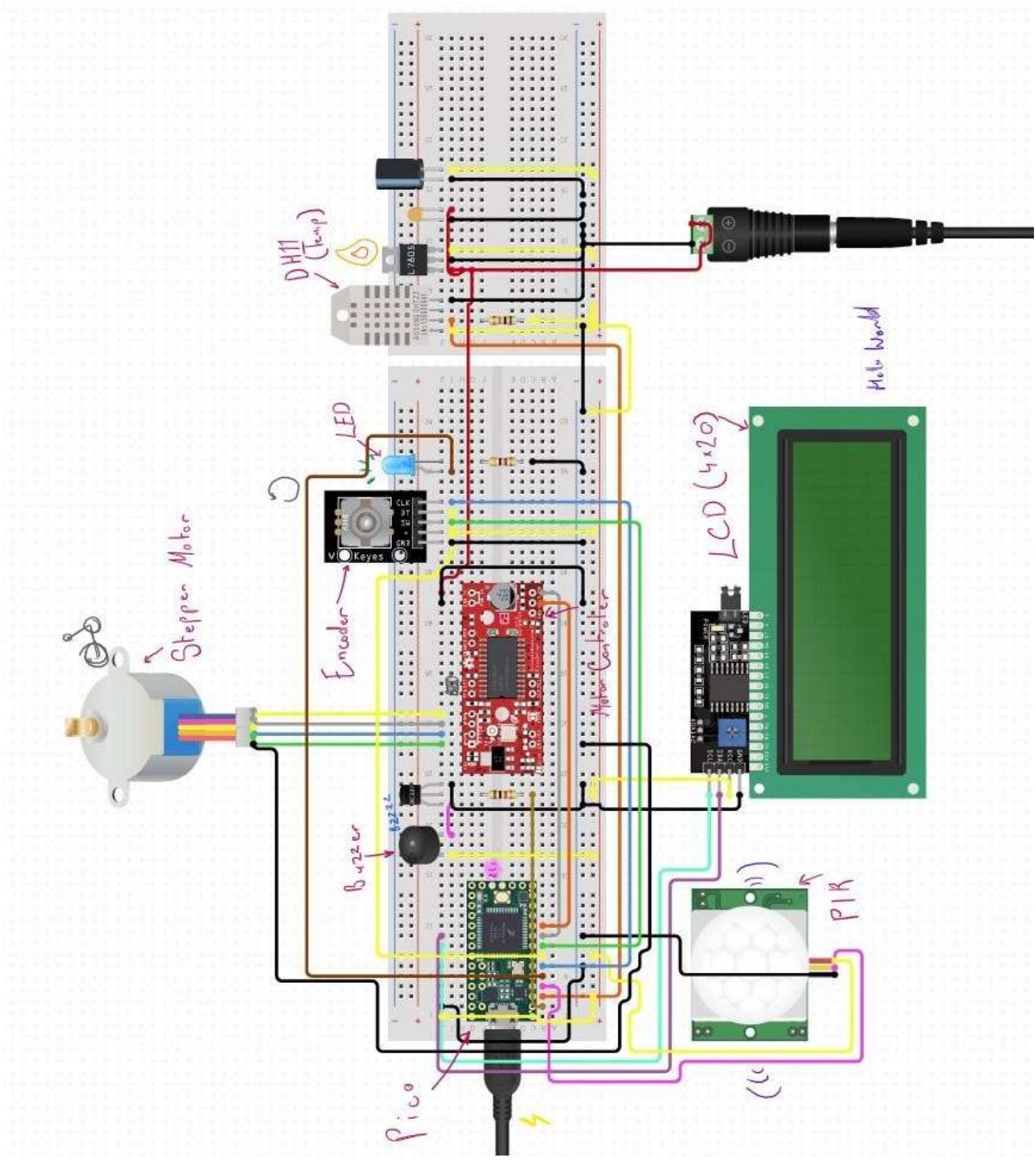
Once I had all my pieces I had to assemble them and fix them together.

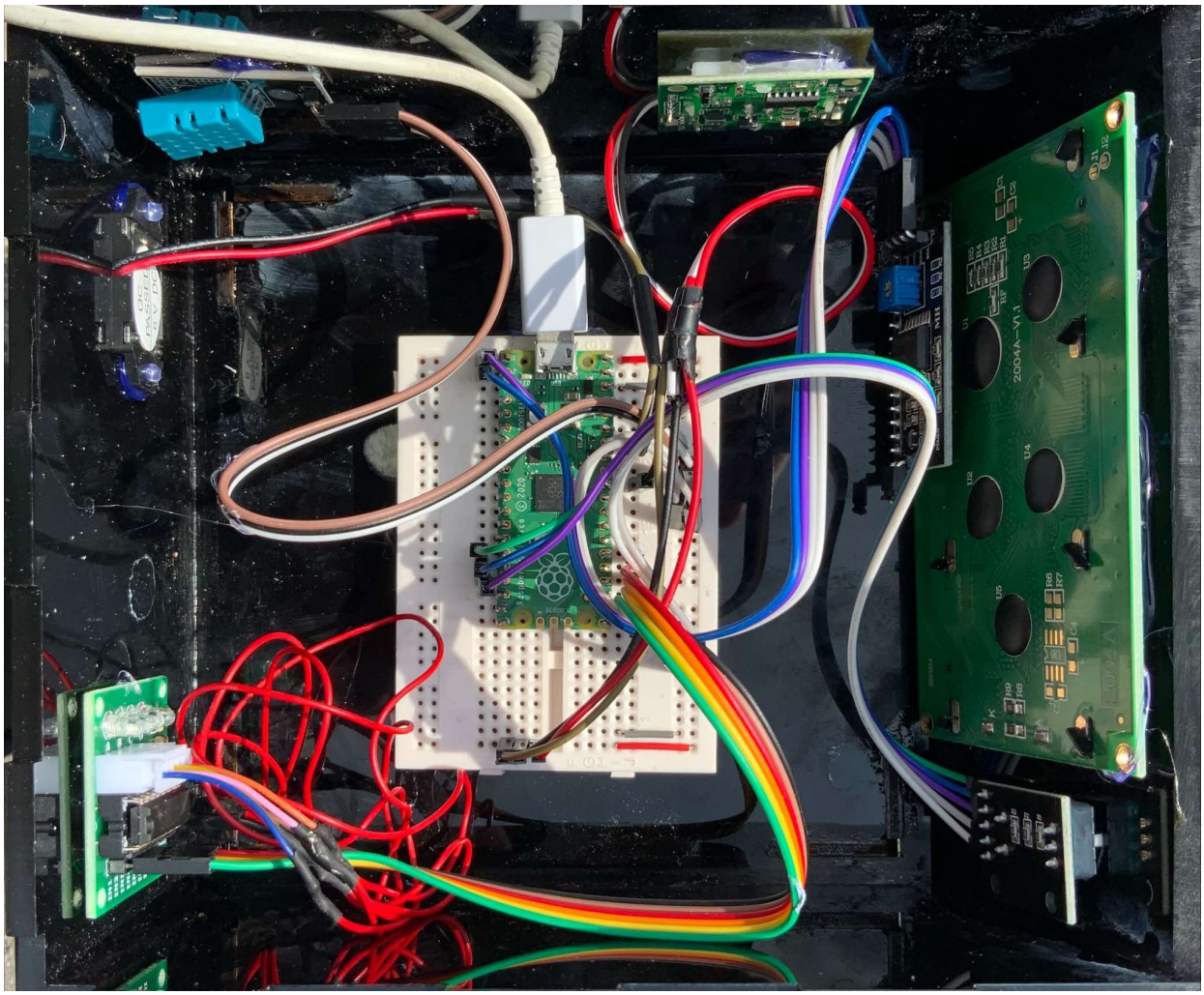
When I knew that the frame all fit together right I was happy to glue them together as unlike the electrical components the structure was unlikely not to work from technical error.

I used solvent and some super glue to assemble the plastic pieces and then used wood glue to assemble the wooden legs, as shown in the photograph above.

I used a tri-square to make sure all my corners were exactly 90° when they were supposed to be as this was important to make sure the top element would fit on top.


Conforms to BS 939
for Squareness


DHT11 (Temp)
LED
Encoder
Keyes
CLK
DT
SW
GND
Stepper Motor
Motor Controller
LCD (4x20)
Hello World!
Buzzer
PIR
Pico


2004A-V1.1

Realization

Coding

To make the project work I had to write a lot of code to get each part working.

I wrote code in Thonny, although it is not my preferred IDE it is the easiest IDE to write code for the Raspberry Pico. It comes pre loaded with all the interfacing and modules.

I broke my code into functions and called them as they were needed.
Below I have written notes on how the code I wrote works:

main.py > getTime

```python
from machine import I2C, Pin, PWM
from pico_i2c_lcd import I2cLcd
from machine import Pin, I2C
from dht import DHT11, InvalidChecksum

import time as t
import utime

pins = [
    Pin(18,Pin.OUT),
    Pin(20,Pin.OUT),
    Pin(16,Pin.OUT),
    Pin(17,Pin.OUT),
    ]

full_step_sequence = [
    [1,0,0,0],
    [0,1,0,0],
    [0,0,1,0],
    [0,0,0,1]
    ]

button_pin = Pin(13, Pin.IN, Pin.PULL_UP)
direction_pin = Pin(12, Pin.IN, Pin.PULL_UP)
step_pin  = Pin(11, Pin.IN, Pin.PULL_UP)

previous_value = True
button_down = False
n = ("dir", direction_pin.value(), "step", step_pin.value(), "button", button_pin.value())

def Step(steps):
    while steps >= 0:
        for step in full_step_sequence:
            for i in range(len(pins)):
                pins[i].value(step[i])
                utime.sleep(0.001)
        steps -= 1

def FullStep():
    Step(22)
```

Impent moduals

Define Stepper Steps

Define encoder

Define turn function

Coding

```python
def getTime(c,h,m):
    #Fix
    hourc = h
    minc = m
    millis = int(t.ticks_ms())
    minutes=(millis/(1000*60))%60
    minutes = int(minutes) + minc
    hours=(millis/(1000*60*60))%24
    hours = int(hours) + hourc



    if len(str(minutes)) < 2:
        minutes = "0"+ str(minutes)

    if len(str(hours)) < 2:
        if len(str(hours)) < 1:
            hours = "00"+ str(hours)
        hours = "0"+ str(hours)

    return (f"{hours}:{minutes}")

def replaceInfoTime():
    lcd.move_to(2,1)
    lcd.putstr(getTime(False,0,0))
```

Get CPU cycles → Time

Coding


```
i2c = I2C(0, sda=Pin(0), scl=Pin(1), freq=400000)

I2C_ADDR = i2c.scan()[0]
lcd = I2cLcd(i2c, I2C_ADDR, 4, 20)

buzzer = PWM(Pin(28))

#Commands
# lcd.blink_cursor_on()
# lcd.blink_cursor_off()
#
# lcd.backlight_on()
# lcd.backlight_off()
#
# lcd.putstr()
# lcd.clear()
#
# lcd.move_to(x,y)

#Menus max 20 per item
time = '00:00'
date = '03/04/2022'

led_level = "100"
buzzer_level = "100"
stepper_steps = "22"

Drop1 = "09:00"
Drop2 = "18:00"
Drop3 = "21:00"

Temp, Humidity = getEnv()

Info_Menu = ["Main Menu",
             f"T:      D:{date}",
             f"La:{Drop1} Ne:{Drop2}",
             f"Temp:{Temp} Hmd:{Humidity}"]

Main_Menu = ["Information","Settings","Config","About","Dispense Next"]

Settings_Menu = ["Main Menu",
                 f"Time :     ",
                 f"Date :{date}",
                 f"Drop 1 :{Drop1}",
                 f"Drop 2 :{Drop2}",
                 f"Drop 3 :{Drop3}"]

Config_Menu = ["Main Menu",
               f"LED :{led_level}",
               f"Loudness :{buzzer_level}",
               f"Steps :{stepper_steps}"]

About_Menu = ["Main Menu",
              "Titus Medical",
              "Designed by [Blank]",
              "LC Technology 2022"]
```

Coding

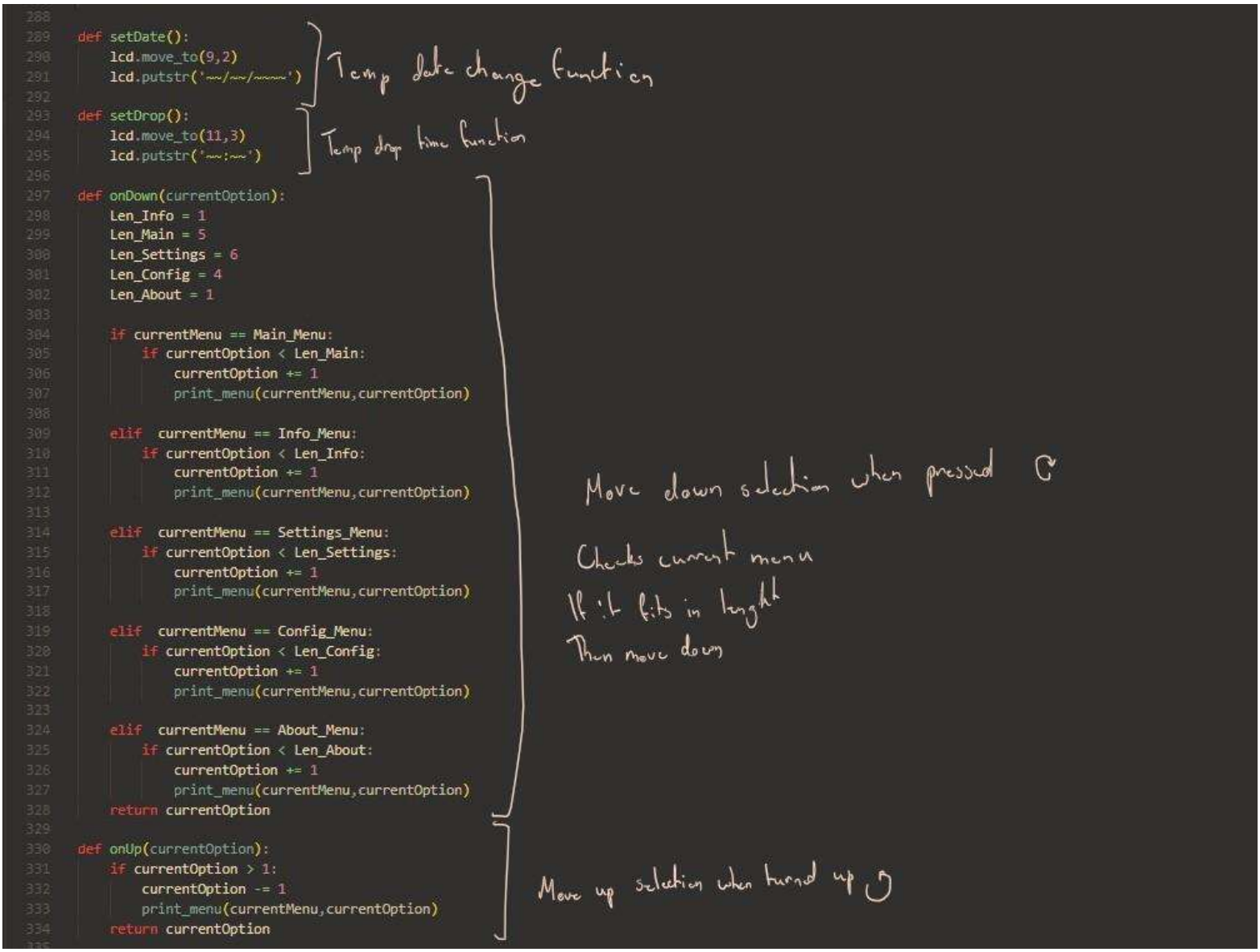

```
def print_menu(options,selection):
    display_height = 4
    current_menu = options
    s = selection -1

    if selection - display_height < 0:
        down_vote = 0
    elif selection - display_height > len(current_menu) - display_height:
        down_vote = 0
    else:
        down_vote = selection - display_height

    lcd.clear()
    #height check
    if len(current_menu) >= display_height:
        for i in range(display_height):
            lcd.move_to(0,i)
            if s > len(current_menu) -1:
                s = len(current_menu) -1
            elif s < 0:
                s = 0

            if i +down_vote  == s:
                string = str(">> " + current_menu[i +down_vote])
                lcd.putstr(string)
            else:
                string = str(current_menu[i +down_vote])
                lcd.putstr(string)
    else:
        display_height = len(current_menu)
        for i in range(display_height):
            lcd.move_to(0,i)
            if s > len(current_menu) -1:
                s = len(current_menu) -1
            elif s < 0:
                s = 0

            if i == s:
                string = str(">> " + current_menu[i])
                lcd.putstr(string)
            else:
                string = str(current_menu[i])
                lcd.putstr(string)

currentMenu = Info_Menu
currentOption = 1
print_menu(currentMenu,currentOption)
replaceInfoTime()

def buzz(s):
    for i in range(s):
        buzzer.freq(1000)
        buzzer.duty_u16(50000)
        t.sleep(1)
        buzzer.duty_u16(0)
        t.sleep(1)
```

Coding

```
def setTime():
    button_pin = Pin(13, Pin.IN, Pin.PULL_UP)
    direction_pin = Pin(12, Pin.IN, Pin.PULL_UP)
    step_pin  = Pin(11, Pin.IN, Pin.PULL_UP)

    previous_value = True
    button_down = False
    n = ("dir", direction_pin.value(), "step", step_pin.value(), "button", button_pin.value())

    minshift = 0
    hourshift = 0

    lcd.move_to(9,1)
    lcd.putstr('      ')

    close = False
    t.sleep(0.5)
    while close == False:
        if n != ("dir", direction_pin.value(), "step", step_pin.value(), "button", button_pin.value()):
            n = ("dir", direction_pin.value(), "step", step_pin.value(), "button", button_pin.value())
            if previous_value != step_pin.value():
                if step_pin.value() == False:
                    if direction_pin.value() == False:
                        print("turned left")
                        hourshift -= 1
                    else:
                        print("turned right")
                        hourshift += 1
                previous_value = step_pin.value()
        if button_pin.value() == False:
            print("exit change")
            close = True
        if hourshift >= 0:
            msg = "+"+str(hourshift)+ " hrs  "
        else:
            msg = str(hourshift)+ " hrs  "
        lcd.move_to(10,1)
        lcd.putstr(msg)


    close = False
    t.sleep(0.5)
    while close == False:
        if n != ("dir", direction_pin.value(), "step", step_pin.value(), "button", button_pin.value()):
            n = ("dir", direction_pin.value(), "step", step_pin.value(), "button", button_pin.value())
            if previous_value != step_pin.value():
                if step_pin.value() == False:
                    if direction_pin.value() == False:
                        print("turned left")
                        if minshift == -59:
                            minshift = 59
                        else:
                            minshift -= 1
                    else:
                        print("turned right")
                        if minshift == 59:
                            minshift = -59
                        else:
                            minshift += 1
                previous_value = step_pin.value()
        if button_pin.value() == False:
            print("exit change")
            close = True
        lcd.move_to(10,1)
        if minshift >= 0:
            msg = "+"+str(minshift)+ " mins  "
        else:
            msg = str(minshift)+ " mins  "
        lcd.putstr(msg)
    lcd.move_to(9,1)
    lcd.putstr('       ')
    print("Adding",str(hourshift)+":"+str(minshift))
    getTime(True,hourshift,minshift)
    replaceSettingsTime()
```

Allows the user to change the time or add offset

Coding




```
def setDate():
    lcd.move_to(9,2)
    lcd.putstr('~~/~~/~~~~')

def setDrop():
    lcd.move_to(11,3)
    lcd.putstr('~~:~~')

def onDown(currentOption):
    Len_Info = 1
    Len_Main = 5
    Len_Settings = 6
    Len_Config = 4
    Len_About = 1

    if currentMenu == Main_Menu:
        if currentOption < Len_Main:
            currentOption += 1
            print_menu(currentMenu,currentOption)

    elif  currentMenu == Info_Menu:
        if currentOption < Len_Info:
            currentOption += 1
            print_menu(currentMenu,currentOption)

    elif  currentMenu == Settings_Menu:
        if currentOption < Len_Settings:
            currentOption += 1
            print_menu(currentMenu,currentOption)

    elif  currentMenu == Config_Menu:
        if currentOption < Len_Config:
            currentOption += 1
            print_menu(currentMenu,currentOption)

    elif  currentMenu == About_Menu:
        if currentOption < Len_About:
            currentOption += 1
            print_menu(currentMenu,currentOption)
    return currentOption

def onUp(currentOption):
    if currentOption > 1:
        currentOption -= 1
        print_menu(currentMenu,currentOption)
    return currentOption
```

Coding

```
def onSelect(currentMenu,currentOption):
    #Main
    if currentMenu == Main_Menu:
        if currentOption == 1:
            currentMenu = Info_Menu
            currentOption = 1
            print_menu(Info_Menu,currentOption)
            replaceInfoTime()
        elif currentOption == 2:
            currentMenu = Settings_Menu
            currentOption = 1
            print_menu(currentMenu,currentOption)
        elif currentOption == 3:
            currentMenu = Config_Menu
            currentOption = 1
            print_menu(currentMenu,currentOption)
        elif currentOption == 4:
            currentMenu = About_Menu
            currentOption = 1
            print_menu(currentMenu,currentOption)
        elif currentOption == 5:
            dispence()

    #Info
    elif  currentMenu == Info_Menu:
        if currentOption == 1:
            currentMenu = Main_Menu
            currentOption = 1
            print_menu(currentMenu,currentOption)

    #Settings
    elif  currentMenu == Settings_Menu:
        if currentOption == 1:
            currentMenu = Main_Menu
            currentOption = 1
            print_menu(currentMenu,currentOption)
        elif currentOption == 2:
            setTime()
        elif currentOption == 3:
            setDate()
        elif currentOption == 4:
            setDrop()
        elif currentOption == 5:
            setDrop()
        elif currentOption == 6:
            setDrop()

    #Config
    elif  currentMenu == Config_Menu:
        if currentOption == 1:
            currentMenu = Main_Menu
            currentOption = 1
            print_menu(currentMenu,currentOption)

    #About
    elif  currentMenu == About_Menu:
        if currentOption == 1:
            currentMenu = Main_Menu
            currentOption = 1
            print_menu(currentMenu,currentOption)

    return currentMenu,currentOption

last_pressed = t.ticks_ms()
screen = False
```

When encoden clicked
Call function for current selected

Menu → Info
→ Settings
→ Config
→ About
→ Demo

Info → Main menu
→ null
↳ For info only

Settings → Main menu
→ Change time
→ Change date
→ Change dispensing

Config → Main menu
→ Modify small things
→ Adjust steps

About → Main menu
→ null
↳ For info only

Then reset screen time out

Coding


while True:
if t.ticks_ms() - last_pressed > 20000:
screen = False
lcd.backlight_off()
Timeout after 20s → Screen off While powered on → Forever
showTime = getTime(False,0,0)
if showTime != getTime(False,0,0) and currentMenu == Info_Menu:
showTime = getTime(False,0,0)
replaceInfoTime()
if currentMenu == Settings_Menu:
replaceSettingsTime()
if showTime in [Drop1, Drop2, Drop3]:
dispense()
Get the time
Update displayed time
If time = drop time → dispense
if screen == False:
if n != ("dir", direction_pin.value(), "step", step_pin.value(), "button", button_pin.value()):
n = ("dir", direction_pin.value(), "step", step_pin.value(), "button", button_pin.value())
if previous_value != step_pin.value():
if step_pin.value() == False:
if direction_pin.value() == False:
screen = True
last_pressed = t.ticks_ms()
lcd.backlight_on()
else:
screen = True
last_pressed = t.ticks_ms()
lcd.backlight_on()
previous_value = step_pin.value()
if button_pin.value() == False and not button_down:
screen = True
last_pressed = t.ticks_ms()
lcd.backlight_on()
button_down = True
if button_pin.value() == True and button_down:
button_down = False
If the screen is off
Then when a button is
press → turn on
and → reset timer
else:
if n != ("dir", direction_pin.value(), "step", step_pin.value(), "button", button_pin.value()):
n = ("dir", direction_pin.value(), "step", step_pin.value(), "button", button_pin.value())
if previous_value != step_pin.value():
if step_pin.value() == False:
if direction_pin.value() == False:
print("turned left")
currentOption = onUp(currentOption)
last_pressed = t.ticks_ms()
else:
print("turned right")
currentOption = onDown(currentOption)
last_pressed = t.ticks_ms()
previous_value = step_pin.value()
if button_pin.value() == False and not button_down:
print("button pushed")
last_pressed = t.ticks_ms()
button_down = True
if button_pin.value() == True and button_down:
button_down = False
if button_pin.value() == False:
print("button pressed")
currentMenu,currentOption = onSelect(currentMenu,currentOption)
last_pressed = t.ticks_ms()
If screen is on
when clicked call pressed function
when turned → find direction
if up → call up
if down → call down
and reset timer

Evaluation and Critical Reflection

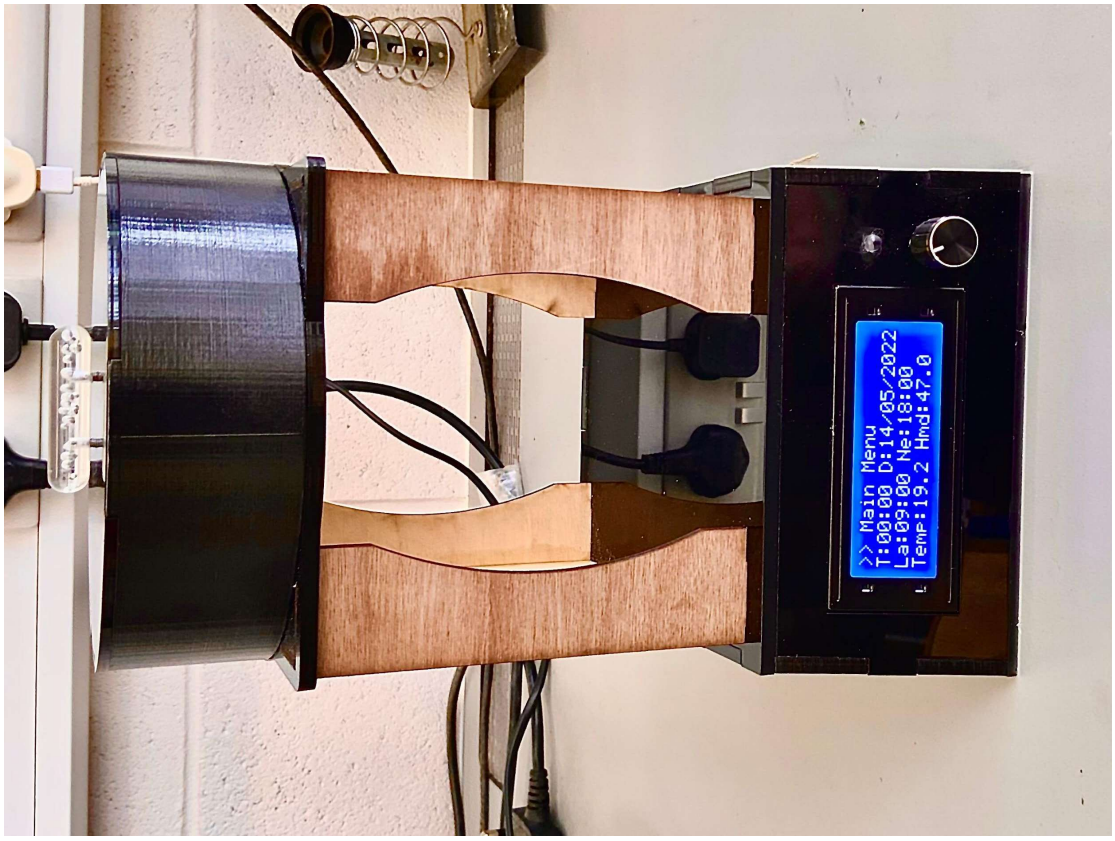
Ultimately I am happy with this project. I feel that my final product fits the brief well and over all the artifacts functions as it's supposed to and is finished well. I think that the design is visually appealing and would fit well in a bedroom. I am also happy with the functionality of this project, I believe that it solves my original problem and has the original features.

Unfortunately there were some elements that did not work out as planned.

Firstly, my original design did not work after I tried it, this was because the motor was much weaker than expected. This was because even with the mechanical advantage I got from a small to large gear ratio, any friction at all would stop the motor from turning.
To solve this I firstly used the belt sander to remove material on the edges of the inside of the dispenser, and then smoothed the outer ring in order to reduce friction as much as possible. This did not work. I decided to use a direct drive from the stepper motor itself, this was a little better but it would still get stuck every so often and was not reliable. This was due to the friction between the lubricated floor and the bottom of the part that rotates.
My final solution was to slightly elevate this part so it was only touching the stepper in the center. This worked well.

Secondly, I had problems with my code and the size of memory on my Pico. Although the Raspberry Pico has a large memory, it was still too small to store all the code I needed for all the original features I planned. While the code worked running off a computer, the files were too large to flash onto the Pico, this meant I had to remove some functionality in the electronic controls.
To solve this I had to remove the large functions where the user is allowed to modify the clock and drop times without a computer. Instead the user has set times for medication and will have to open the file to change the times via USB on a laptop or desktop.

These were the main issues with my project but I feel like the solutions that I provided were able to fix these problems to a satisfactory standard.


>> Main Menu
T:00:00 D:14/05/2022
La:09:00 Ne:18:00
Temp:19.2 Hmd:47.0