**Department of Electronic and Telecommunication Engineering**

**University of Moratuwa**


## EN 3023 – Electronic Design Realization

**Report on the Ideation, Design, and Product development**

# Elderbot

*A detection and warning device for elderly patients against sudden falls*

**Group Number 5:**

| | |
|---|---|
| Y.S Ginige | - 180195A |
| W.A.S.N. Jayawardena | - 180285B |
| P.A.S. Kaushalya | - 180320F |
| D.S.P.A. Niwarthana | - 180433E |
| W.A.V. Ravihansa | - 180544U |
| P.H.D.S. Silva | - 180604F |
| P.M.P.H Somarathna | - 180616T |
| A. Thieshanthan | - 180641N |
| L.T.N. Wickremasinghe | - 180701B |

# Table of Contents

**1 Feasibility and Concept** **3**
1.1 Identify the Problem . . . 3
1.1.1 Market Research . . . 3
1.2 Identify the Criteria and Constraints . . . 3
1.2.1 Toolkit for Inclusive Design . . . 4
1.3 Brainstorm Possible Solutions . . . 4
1.4 Select a Design . . . 5
1.5 Build a Model or Prototype . . . 6
1.5.1 PCB Design . . . 6
1.5.2 Enclosure Design . . . 8
1.5.3 Web Interface and IoT Component . . . 10

**2 Testing and Evaluations** **12**

**A Appendix** **13**

# 1 Feasibility and Concept

## 1.1 Identify the Problem

In recent years, the risk of falling, especially among the aging population, has been propelled by the rising prevalence of various chronic diseases such as Alzheimer's and Dementia along with the increasing incidence of chronic disorders such as diabetes, chronic obstructive pulmonary disease (COPD), cardiovascular diseases, etc.

Implementation of fall detection systems is vital because falls are risky especially for individuals who live alone in their homes. According to the World Health Organization1, about 28% - 35% of individuals over the age of 65 are projected to go through one fall at a minimum each year, and this danger rises to 42% for individuals over the age of 70. Such a situation impels the necessity for fall detection systems.

Fall detection systems can mitigate some of the adverse consequences of a fall. One major outcome of a fall is a bone fracture, for instance, a fracture of the hip, femur, forearm, leg, or ankle bone. Immediate medical intervention could effectively prevent serious deterioration of bodily functions. Through this project, we develop ElderBOT, a wearable fall detection device with an alert system that utilizes 3 modes of notifications in case of an emergency. Our product is a reliable, effective, and cost-efficient device designed to cater to user requirements. Our product is a reliable, effective, and cost-efficient device designed to cater to user requirements.

### 1.1.1 Market Research

As part of the primary research, an online qualitative survey was conducted to discover the requirements and concerns of potential users. The form can be accessed here: https://forms/elder-bot. The questionnaire consists of 10/11 questions, depending on whether the responder has previously used a similar device or not. The results of the survey are attached in the Appendix.

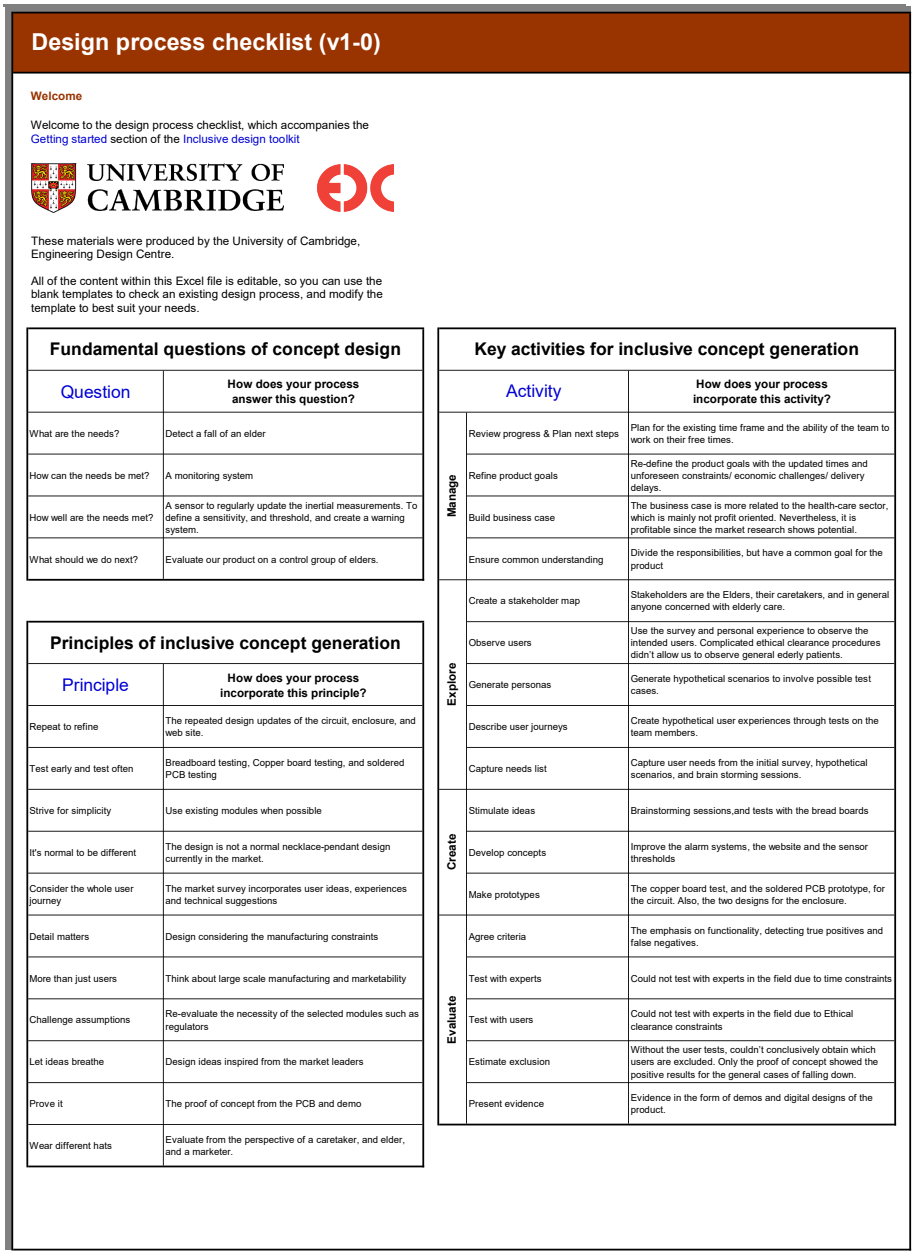
## 1.2 Identify the Criteria and Constraints

The project idea was chosen considering the capacity of the team within the given time frame, with limited access to lab facilities, and space constraints / specifications. Selected design specifications to be implemented are:

- Required metrics:
  - Since it is related to Human health, need to aim for 100% True Positives, and 0% False Negatives, where True Positive is accurate detection of a fall.
  - A system to override an alarm so that the False positive alarms could be manually switched off.
- Time window for informing the caretaker of a fall- within 2 seconds of falling
- Time window for informing distant relatives/ guardians- within 30 seconds of falling
- Dimensions of the device: 10 cm x 6 cm x 6 cm
- Expected overall weight: 200g - 300g
- Rechargeable DC battery-powered device
- Wearable around the waste of an elderly person- ideally attached to a regular belt.
- Simple and user-friendly operation with a button that is accessible to an elderly adult.
- Enclosure suitable for a wearable device on the body
- Initially planned cap on the product cost: Rs.5000

The idea and user cases were validated from the market survey, the design team, and consulting an instructor for perspectives and opinions. These specifications were further refined during the testing phases of the project.

### 1.2.1 Toolkit for Inclusive Design



**Design process checklist (v1-0)**

**Welcome**

Welcome to the design process checklist, which accompanies the Getting started section of the Inclusive design toolkit

UNIVERSITY OF CAMBRIDGE

These materials were produced by the University of Cambridge, Engineering Design Centre.

All of the content within this Excel file is editable, so you can use the blank templates to check an existing design process, and modify the template to best suit your needs.

**Fundamental questions of concept design**

| Question | How does your process answer this question? |
|---|---|
| What are the needs? | Detect a fall of an elder |
| How can the needs be met? | A monitoring system |
| How well are the needs met? | A sensor to regularly update the inertial measurements. To define a sensitivity, and threshold, and create a warning system. |
| What should we do next? | Evaluate our product on a control group of elders. |

**Principles of inclusive concept generation**

| Principle | How does your process incorporate this principle? |
|---|---|
| Repeat to refine | The repeated design updates of the circuit, enclosure, and web site. |
| Test early and test often | Breadboard testing, Copper board testing, and soldered PCB testing |
| Strive for simplicity | Use existing modules when possible |
| It's normal to be different | The design is not a normal necklace-pendant design currently in the market. |
| Consider the whole user journey | The market survey incorporates user ideas, experiences and technical suggestions |
| Detail matters | Design considering the manufacturing constraints |
| More than just users | Think about large scale manufacturing and marketability |
| Challenge assumptions | Re-evaluate the necessity of the selected modules such as regulators |
| Let ideas breathe | Design ideas inspired from the market leaders |
| Prove it | The proof of concept from the PCB and demo |
| Wear different hats | Evaluate from the perspective of a caretaker, and elder, and a marketer. |

**Key activities for inclusive concept generation**

| | Activity | How does your process incorporate this activity? |
|---|---|---|
| Manage | Review progress & Plan next steps | Plan for the existing time frame and the ability of the team to work on their free times. |
| | Refine product goals | Re-define the product goals with the updated times and unforeseen constraints/ economic challenges/ delivery delays. |
| | Build business case | The business case is more related to the health-care sector, which is mainly not profit oriented. Nevertheless, it is profitable since the market research shows potential. |
| | Ensure common understanding | Divide the responsibilities, but have a common goal for the product |
| Explore | Create a stakeholder map | Stakeholders are the Elders, their caretakers, and in general anyone concerned with elderly care. |
| | Observe users | Use the survey and personal experience to observe the intended users. Complicated ethical clearance procedures didn't allow us to observe general ederly patients. |
| | Generate personas | Generate hypothetical scenarios to involve possible test cases. |
| | Describe user journeys | Create hypothetical user experiences through tests on the team members. |
| | Capture needs list | Capture user needs from the initial survey, hypothetical scenarios, and brain storming sessions. |
| Create | Stimulate ideas | Brainstorming sessions,and tests with the bread boards |
| | Develop concepts | Improve the alarm systems, the website and the sensor thresholds |
| | Make prototypes | The copper board test, and the soldered PCB prototype, for the circuit. Also, the two designs for the enclosure. |
| Evaluate | Agree criteria | The emphasis on functionality, detecting true positives and false negatives. |
| | Test with experts | Could not test with experts in the field due to time constraints |
| | Test with users | Could not test with experts in the field due to Ethical clearance constraints |
| | Estimate exclusion | Without the user tests, couldn't conclusively obtain which users are excluded. Only the proof of concept showed the positive results for the general cases of falling down. |
| | Present evidence | Evidence in the form of demos and digital designs of the product. |

**Fig. 1:** Design Process checklist of Elderbot

To follow the Inclusive design approach, involving Human centred design (HCD) and the Activity centred design (ACD), we follow the recommended toolkit to track the current status, and immediate next steps for Elderbot. It is shown in figure 1. The Impact factor analysis with the relevant stakeholders, and the initial assessment was a guide in deciding which aspects of the design to focus on. This is included in the Appendix.

## 1.3 Brainstorm Possible Solutions

Following the in-class activity conducted regarding this on our semester 2 projects, we carried out a brainstorming session to gather ideas. We decided to engage in a different project title, we re-did this step, with more inputs from the market survey and other stakeholders. The analysis is summarized in

figure 2, and the table in figure 3

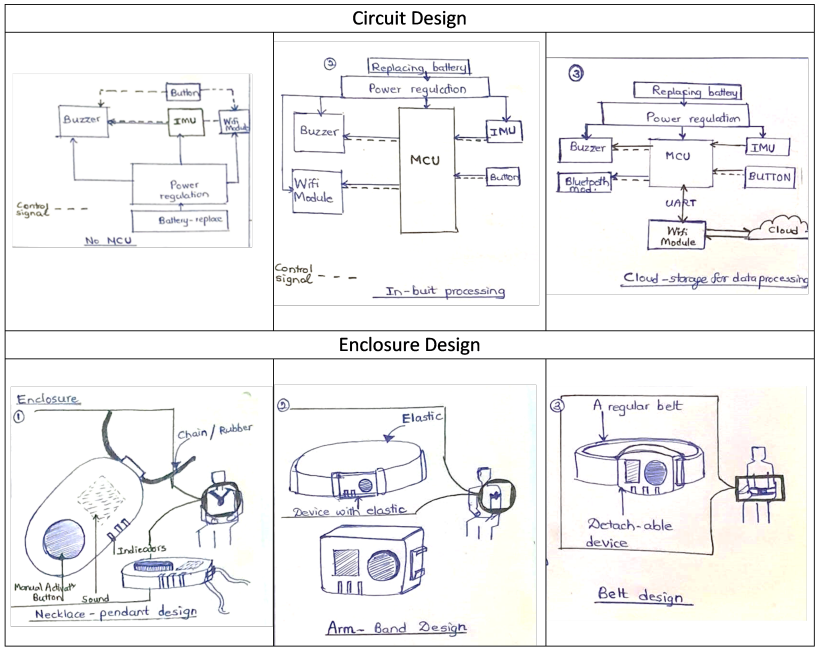



**Fig. 2:** Designs of the circuit and enclosure

| Design | | Compatibility Marks | Marks for the circuit | Marks for the enclosure | Total Marks | Rank |
|---|---|---|---|---|---|---|
| Circuit | Enclosure | | | | | |
| 1 | 1 | 30 | 30 | 50 | 36.6 | 9 |
| | 2 | 45 | 30 | 70 | 48.3 | 8 |
| | 3 | 45 | 30 | 90 | 51.6 | 7 |
| 2 | 1 | 60 | 90 | 50 | 66.6 | 4 |
| | 2 | 65 | 90 | 70 | 68.3 | 3 |
| | 3 | 60 | 90 | 90 | 80.0 | 1 |
| 3 | 1 | 55 | 70 | 50 | 58.3 | 6 |
| | 2 | 60 | 70 | 70 | 66.6 | 4 |
| | 3 | 60 | 70 | 90 | 73.3 | 2 |

**Fig. 3:** Evaluations of circuit and enclosure designs

## 1.4 Select a Design

The selected circuit plan number 2, with the selected enclosure plan number 3, is the model for Elderbot. It is a cuboidal shaped device which can be attached to a commonly found belt, and is to be worn around the waist of the elder. The high-level functionality of the device is explained in the figure 4.

We identify 3 use cases (modes) of our device, which will trigger different notifications. When an elder falls, the sensors in the device are expected to sense the change in motion and pass that information to the circuit which will be used to detect whether a fall occurred according to pre-calculated thresholds. After confirmation ("Fall" mode), a message is to be sent to the server informing the guardian of the user. If the change in motion is not confirmed to be a fall either by the algorithm, or by the user by manually turning the alarm off ("Normal" mode), no message will be sent. A user can also trigger the device manually to alert their guardians ("Emergency" mode).

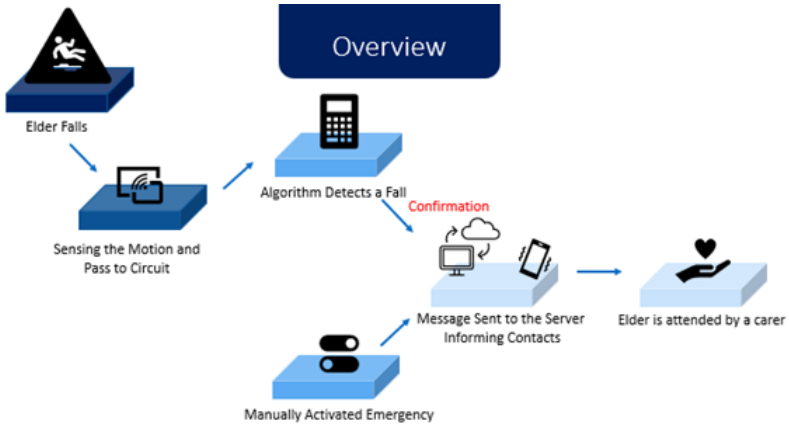



**Fig. 4:** High level functionality of Elderbot

## 1.5 Build a Model or Prototype

### 1.5.1 PCB Design

The design flow of the PCB design is as follows,

- **Library Creation** - Creating the Schematic and PCB libraries of the components looking at their datasheets. (Ex: MPU6050 the 6-axis IMU sensor, SX1308 the boost converter, TP4056 the lithium-ion battery charging module, Push Button, and the Switch; downloaded the integrated library of the NodeMCU.)
- **Schematic Capturing** - Drawing the schematic diagram, eliminating all the Altium warnings.
- **PCB Layout** - Our PCB is a two-layer PCB. When drawing the PCB, we stick to the design rules.
- **Generating Output Packages** (ex: Fabrication outputs - Gerber files, NC drill Files, BOM) The BOM is shown in figure 5
- **Manufacturing the PCB** The PCB was manufactured by https://jlcpcb.com

**Schematic Diagram of the Circuit:** The following describes image 6. The main components of the circuit are NodeMCU, IMU sensor, the power supply unit to the NodeMCU, battery voltage indicator unit, buzzer, the push button, and the array of LEDs. NodeMCU is powered using a 3.7V lithium-ion battery. The boost converter is used to boost the 3.7 V to 5V. The 5V boosted voltage is connected to the switch and the switch is connected to the 5V pin of NodeMCU. The battery terminal is connected to the TP4056 battery charger module. Thus, the battery can be charged using the 5V MicroUSB DataCable. While charging the battery, it is recommended to turn off the switch. We are using a Li-ion battery with a capacity of 4800 mAh. The approximate current flow of the circuit is 75 mA. Thus the discharge duration would be nearly 64 hrs which makes our system feasible to operate consecutively for upto approximate 2.5 days. For our task, only accelerometer and gyroscope measurements are enough thus when selecting an IMU sensor we selected MPU6050 module, the commonly used 6-axis imu sensor. The LED array

| Line # | Name | Description | Designator | Quantity | Supplier 1 | Supplier Unit Price 1 | Supplier Subtotal 1 |
|---|---|---|---|---|---|---|---|
| 1 | NodeMCU | An open-source firmware and development kit that helps you to Prototype your IOT product within a few Lua script lines | A | 1 | https://lk-tronics.com/product/nodemcu-esp8266-ch340g-development-board/ | LKR 1700.00 | LKR 1700.00 |
| 2 | LED3 | Typical BLUE SiC LED | D1, D2, D3, D4 | 4 | https://scionelectronics.com/product/1206-blue-smd-led/ | LKR 3.00 | LKR 12.00 |
| 3 | Buzzer | Magnetic Transducer Buzzer | LS1 | 1 | https://www.digikey.com.au/en/products/detail/cui-devices/CMI-1295-0585T/10326184 | LKR 625.00 | LKR 625.00 |
| 4 | Resistor 1k +/-1% 1206 250 mW | Chip Resistor, 1 KOhm, +/- 1%, 0.25 W, -55 to 155 degC, 1206 (3216 Metric) | R1, R2, R3, R4, R5, R6, R7, R8 | 8 | https://www.mouser.com/ProductDetail/Vishay-Dale/RCC08051K00FKEA?qs=sGAEpiMZZMtlubZbdhIBINZyO39%252BOXIfQnKiza%252BxGEw%3D | LKR 80.00 | LKR 640.00 |
| 5 | Switch | | S1 | 1 | https://www.digikey.com.au/en/products/detail/e-switch/EG1247/1804449?utm_adgroup=General&utm_source=google&utm_medium=cpc&utm_campaign=Smart%20Shopping_Product_Zombie%20SKUS&utm_term=&productid=1804449&utm_id=go_cmp-12195090931_adg-112248982970_ad-494933794766_pla-891529596407_dev-c_ext-_prd-1804449_sig-CjwKCAjws8yUBhA1EiwAi_tpERC_oFJlJowykDFGgxVjahLr7T3bfCEBuG_2e3xjvoKBM-ullfyfiRoC94oQAvD_BwE&gclid=CjwKCAjws8yUBhA1EiwAi_tpERC_oFJlJowykDFGgxVjahLr7T3bfCEBuG_2e3xjvoKBM-ullfyfiRoC94oQAvD_BwE | LKR 450.00 | LKR 450.00 |
| 6 | PushButton | | S2 | 1 | https://www.amazon.com/QTEATAK-Momentary-Tactile-Button-Assortment/dp/B07VQF8P2Y/ref=sr_1_5?keywords=4+Pin+Momentary+Push+Button+Switch&qid=1655025695&sr=8-5 | LKR 30.00 | LKR 30.00 |
| 7 | MPU6050 | | U1 | 1 | https://nilambaraelectronics.com/product/gy-521-mpu6050-3-axis-analog-gyroscope/ | LKR 430.00 | LKR 430.00 |
| 8 | TP4056 | | U2 | 1 | http://www.senith.lk/shop/item/1238/tp4056-lithium-ion-18650-battery-charger-module | LKR 180.00 | LKR 180.00 |
| 9 | SX1308 | | U3 | 1 | https://www.aliexpress.com/item/1005001621801605.html?spm=a2g0o.search0304.0.0.29de660fszmtVU&algo_pvid=02cc3c29-862b-4cd9-8984-69da757b7727&aem_p4p_detail=2022061201402213525580923064740026800419&algo_exp_id=02cc3c29-862b-4cd9-8984-69da757b7727-0&pdp_ext_f=%7B%22sku_id%22%3A%221200001684648185 5%22%7D&pdp_npi=2%40dis%21LKR%21%21136.59%21%21%21136.59%21%21%402103399116550232228044435efc0d%211200001684648185 5%21sea | LKR 172.53 | LKR 172.53 |
| Total | | | | | | | LKR 4239.53 |

**Fig. 5:** PCB Bill of materials

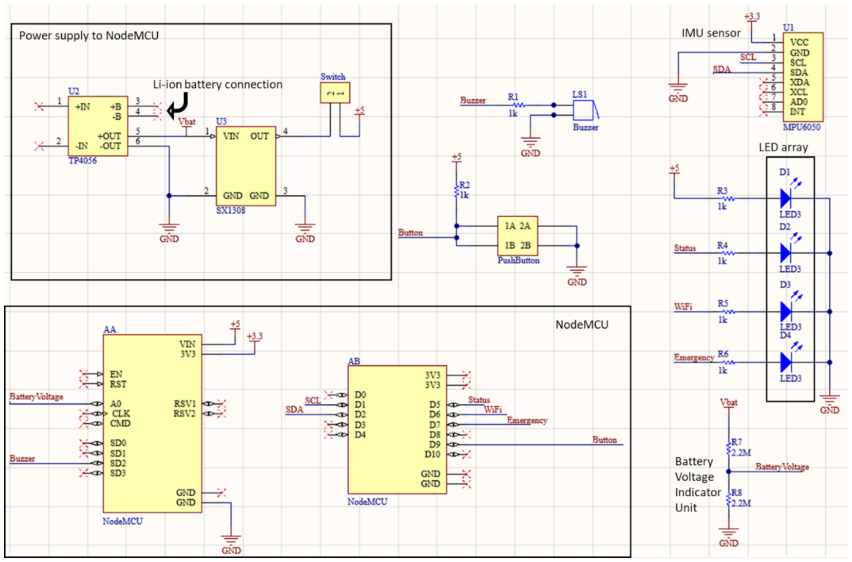



**Fig. 6:** Schematic diagram of the circuit

indicates information on the system's power and notify of a fall detection. We verified the functionality of the circuit several times by implementing it on a dot board.

The PCB layout after routing is shown in figure 8 and some images of the output files (Gerber and NC drill files) are shown in figure 9. The decisions made with this are:

- Selecting an IMU sensor - When selecting an IMU sensor, we had to choose between 6-axis and 9-axis IMU sensors. As mentioned earlier as the accelerometer and gyroscope readings are suffcient for the purpose we decided to go with the most commonly used MPU6050 IMU sensor.
- Adhering to the concept of modularity - Our initial decision was to build the circuit from scratch expecting that it would cost us lesser space rather than using the modules. But it increased the complexity of the circuit. Thus we decided to go with existing modules.
- Component Placement - When placing the components on the PCB layout (figure 7), special attention was paid to the convenience of the enclosure design. Placing the edge which contains the MicroUSB data jack of the battery charging module parallel to an edge of the PCB is an example

consideration. Special attention was paid when placing the LEDs and buttons and for ease in battery replacement.

- Size of the PCB - The volume and the weight are critical characteristics of our system. Our goal was making the PCB as small as possible. Going for a double layered PCB is a result of that.

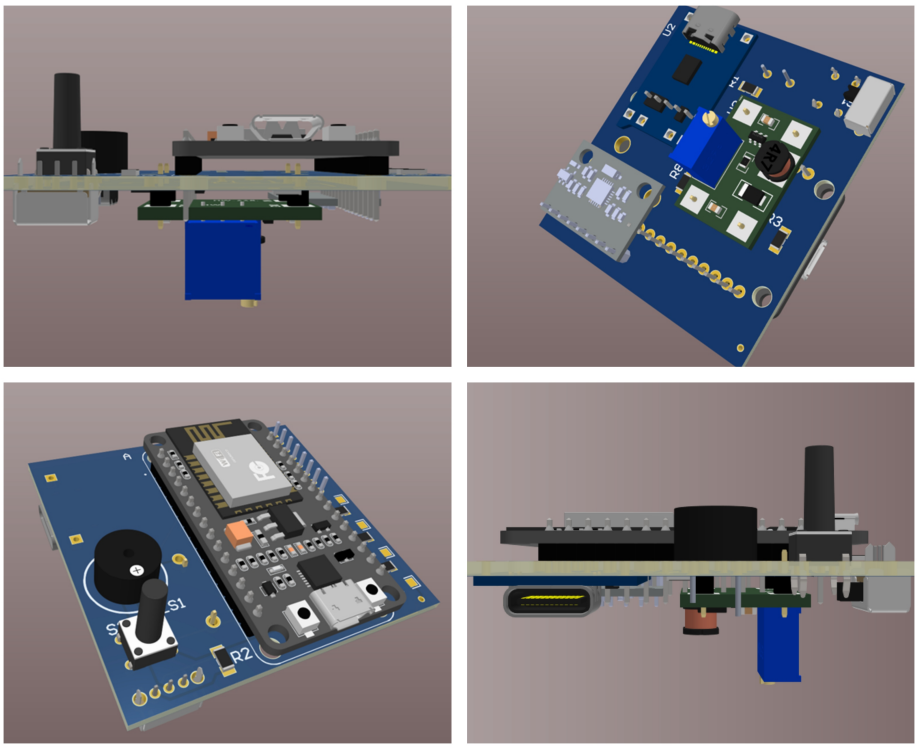



**Fig. 7:** PCB- 3D model

The challenges faced are,

- Soldering the SMD components - Due to the smaller size of the SMD LEDs and resistors, it was challenging in soldering without a hot airgun.
- Footprint non-overlaps - For the buzzer and the switch we couldn't find the exact component we drew the footprint for. It caused the distance between the pins to not match with the footprint.

### 1.5.2 Enclosure Design

The online qualitative survey was conducted to discover the requirements and concerns of potential users. From the survey, we observed that 56% of the users prefer having a device around the waist. This is shown in figure 10

The enclosure design has two parts,

- Main enclosure to enclose the PCB and components.
- Outer cover with loops to wear with a belt on the waist.

The combination is shown in figure 11(d,e). The design for the main box was done considering the PCB placement, component placement, sizes, and ports. The design consists of two parts, the upper lid (figure11(a,b)) and the bottom part (figure11(c)) which holds the PCB. The upper lid consists of the push button, cuts for the LEDs, buzzer and the support which connects the lid to the bottom part. The bottom part holds the PCB, battery, ports for USB and switch. The assembly of the two parts were tested for any interferences using interference detection test available in solidworks.

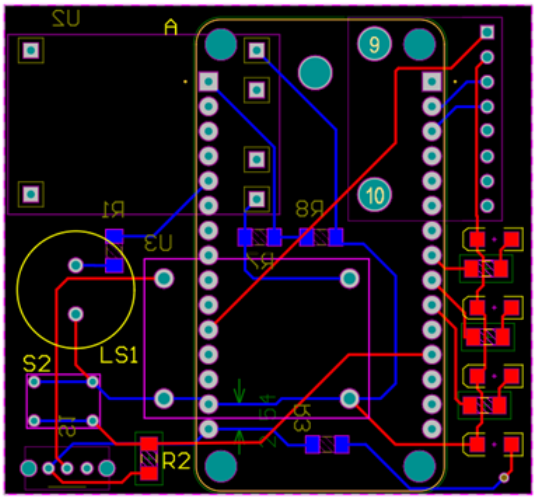



**Fig. 8:** PCB layout

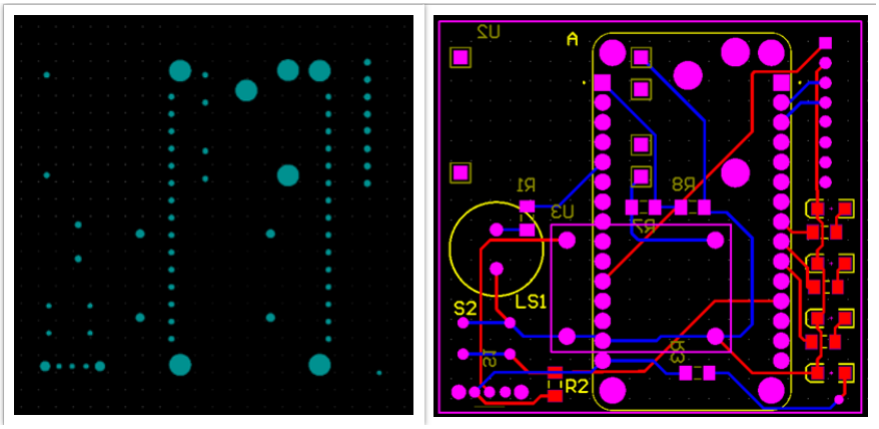



**Fig. 9:** PCB design output files: A Gerber file(left) and NC drill files(right)


**Fig. 10:** Survey results of the preferred position

**The outer cover:** This provides extra protection to the main enclosure along with loops to wear with a belt on the waist. The design of the outer cover is shown in figure 12(left). The design covers the main box and openings are provided so as to not interfere with alert mechanisms (LEDs, Buzzer).

In this first design (figure 12 (right)), we observed that the edges are unable to handle high stresses. Therefore, in the second design, we decided to add horizontal bars to provide extra support and increased the wall thickness as well.

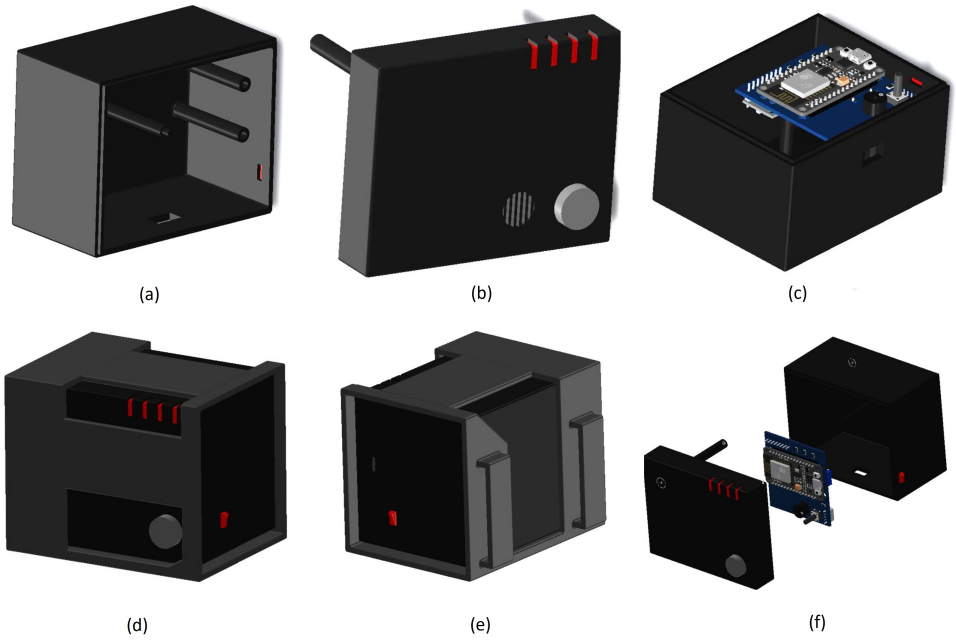



**Fig. 11:** (a) Lower lid of the main enclosure (b) Lower lid with the PCB attached (c) Exploded view of

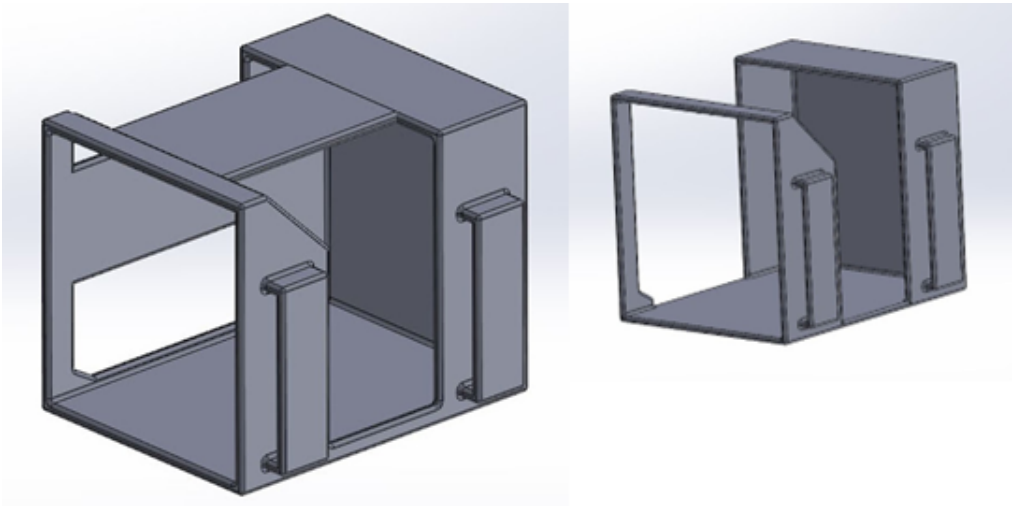


**Fig. 12:** Outer cover: final design(left) and initial design(right)

### 1.5.3 Web Interface and IoT Component

The details of the IoT components of Elderbot, which mainly deals with the communications related to the device are detailed below:

- An AWS cloud server is used to store and handle user information and other management activities like sending SMS data to the SMS hub, running the BACK-END service of the web interface, etc. After buying an Elderbot device, the user must go through a registering process in order to connect with our database.
- When a new device is given to a user, he/she should be able to connect the device to a WIFI network. To facilitate that we provide a windows application to the user, (The UI in windows is shown in figure 13) and he/she has to fill in relevant information like the user's WIFI SSID,

Password, and etc. This information is sent to the Elderbot device (ESP8266-NodeMCU module) using serial communication and after the connection is established, user details will be sent to the AWS server for user registration.

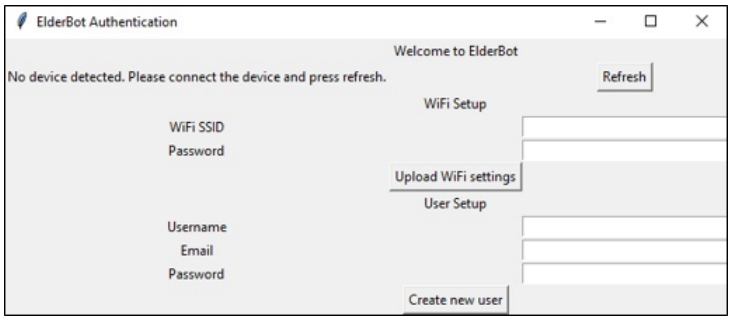



**Fig. 13:** UI on windows software

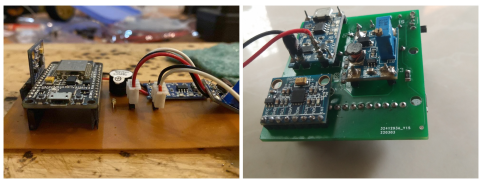
- The Elderbot device is connected to the internet network based on the given credentials and user details are sent to the AWS Server which is used to create accounts. After successful completion of this process, an acknowledgment is sent back to the device.


**Fig. 14:** Communication between devices

- Communication between the device and the AWS server is built via the MQTT protocol stack since it is the most suitable protocol to handle the situation considering the scalability of users.
- Once all devices and servers are successfully connected, whenever a falling or emergency status occurs NodeMCU is triggered and after the triggering, the module sends data frames which consist of User ID, status, and a small message to the AWS Server.
- All devices have subscribed to a single topic for the uplink. The server sends a message as an acknowledgment and the device starts to ring the buzzer. If the action mistakenly occurred, the user can withdraw the trigger by just pressing the emergency button again within 10 seconds.
- We have provided a medication alarm service where a user can set medication alarms by logging into his profile in the web application. A timely reminder message will be sent to the Elderbot device, and an alarm will ring until it is forcibly stopped by pressing the button. In this downlink, all the devices are subscribed to a single topic and the User ID in the payload is used to identify relevant messages.
- We have developed our own SMS hub as a cost-cutting step for the SMS sending stage. There we have used a NodeMCU to subscribe to the MQTT client and a GSM9001A module to send SMS.
- When a fall or an emergency happens, the device is triggered, and the data frame is sent to the server. After that, the server publishes a data packet containing the phone number and a message to be sent. SMS hub collects these packets and sends a relevant message to the particular user. After this is completed, an acknowledgment is sent back to the server.

There are several challenges for the above mentioned tasks.

**(a)** Code tests



**(b)** Circuit

**Fig. 15:** Testing the circuit

- At the beginning we didn't have an idea about the complexity of the fallen detection algorithm and its performance requirements. So, it was difficult to select a microcontroller or a single board computer. Since one of the main targets was to minimize the cost, we decided to send all the raw data to a server and do all the processing steps there and send back the result. But fortunately, at the end we could simplify the algorithm which can be run on the NodeMCU itself.
- First step of using elderbot is registering a device with user information. The main difficulty we faced here was sending internet connectivity information to the device. We have created a new windows application that collects the connectivity credentials and sends them to the device through a serial communication, using a USB cable. This step should be done in the most user friendly way otherwise users find it hard to establish the connectivity. So it took some time to figure out this step.
- We tried to find a free or less costly SMS hub on the internet. They were pricey since they are using international SMS and it took comparatively a long period to do the research. Finally, we came to the conclusion of creating our own SMShub for sending SMS with the required details. GSM modules of SMShub had an issue in firmware and we had to find a solution for it.
- Choosing the protocol stack was a bit difficult because the Emergency and the Fallen status should be communicated with high reliability and less time. On the other hand our services should be scalable for a large number of users. So we decided to use the MQTT protocol stack and developed a manual acknowledgement to enrich the reliability achieving both requirements.
- Maintaining the connectivity between server and the device was one of the major concerns. We could make the connection back in 4 seconds in a connection loss for any reason, except the loss of WiFi connectivity, by optimizing the code.

## 2 Testing and Evaluations

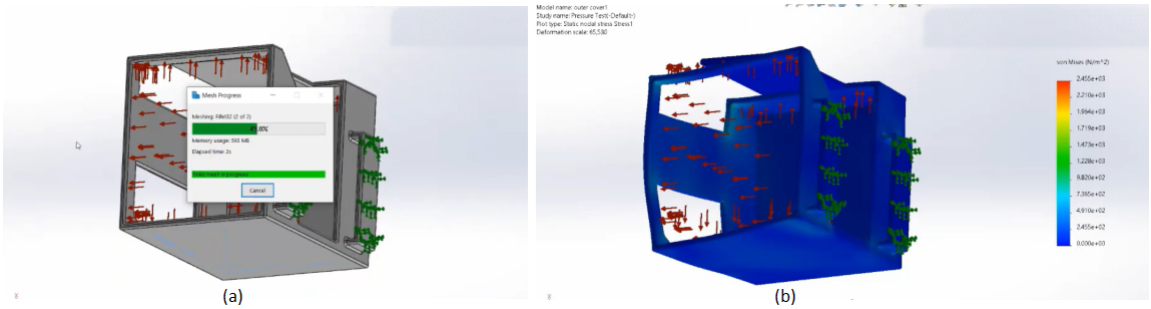



**Fig. 16:** Pressure test simulation

The model is tested with members of the team. We couldn't organize a trial with elderly people. Therefore the evaluations were limited to various hypothetical scenarios - and our team re-enacting them (Figure 15a).

The circuit was tested in a bread board, followed by a copper board. Finally the circuit was tested in with the soldered PCB. (Figure 15b)

The PCB design was error checked and produced after checking for any design rule violations. The enclosure was tested physically, as well as a simulation on Solidworks to test of it withstood pressure from a sudden fall (Figure 16).

**The cost evaluation** This is done by considering the component cost, the enclosure cost, and the manufacturing overheads. To decide the price point of the product at this level of the design, a 5% profit margin is kept. (Figure 17)

| | Prototype (per unit) LKR | Mass Production (per unit) LKR |
|---|---|---|
| Components | 4240 | 3800 |
| PCB Manufacturing | 300 | 300 |
| Enclosure Manufacturing | 1000 | 500 |
| Labor | 200 | 150 |
| Other costs | 1000 | 500 |
| **Total Production Cost** | 5740 | 4950 |
| Distribution + warehousing | N/A | 100 |
| **Total Cost** | 5740 | 4950 |
| **Profit Margin (5%)** | 287 | 247 |
| **Price** | **7027** | **5597** |

**Fig. 17:** Cost and price estimate per unit

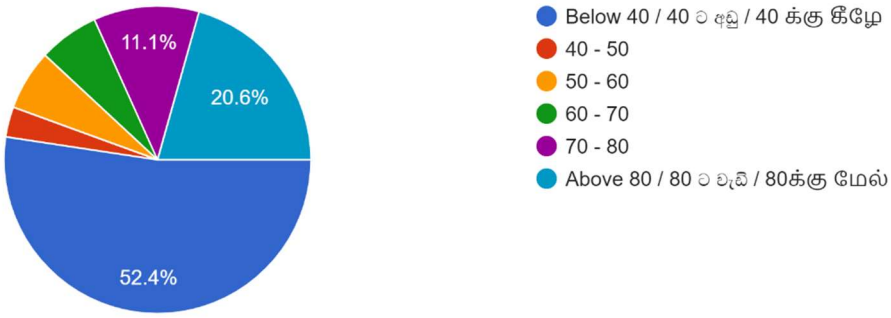
# A Appendix

The appendix contains the following pdfs

1. The market survey on the user attitudes
2. The impact factor analysis for ElderBOT
3. The initial assessment on stakeholders and impact factors
4. Schematic Design
5. PCB Layout and Silk Screen
6. Engineering drawing upperlid
7. Engineering drawing bottom part
8. Web interface code
9. MCU code

## Appendix I: Market Survey

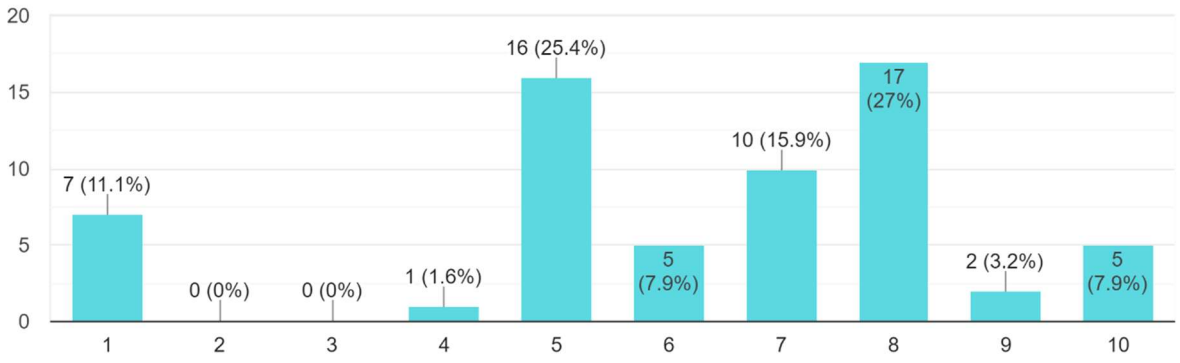
1. **What is your age?**




2. **Are falls a particular concern for you?**

   **32 Yes 31 No**

3. **If our new product was available today, how likely would you be using it instead of products currently available.**

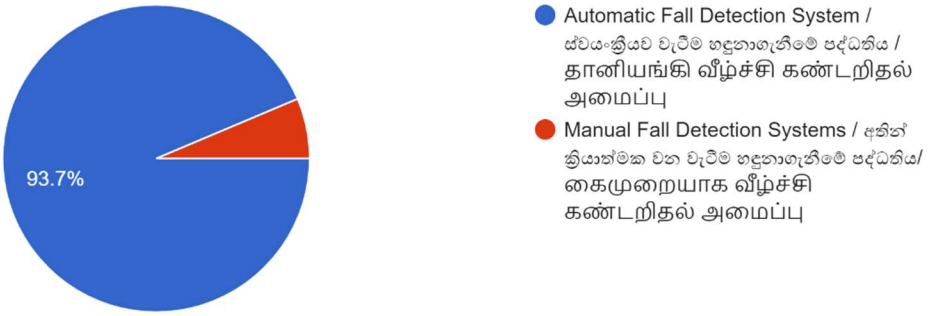



4. **Do you currently use any product with this capability?**

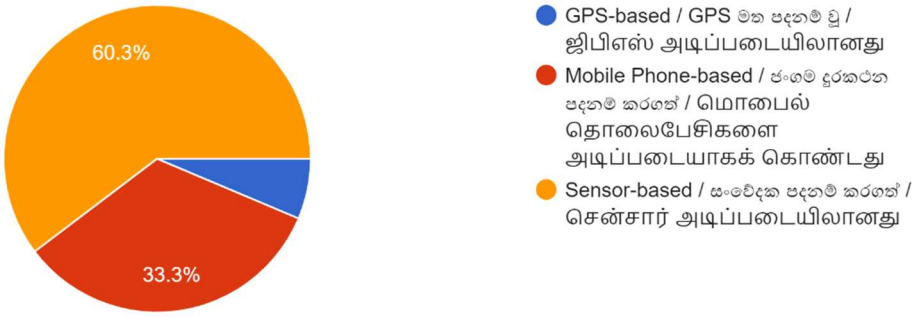
   **0 Yes 63 No**

5. **Where would you prefer to wear such a product?**


6. **I like the product to have a _____**



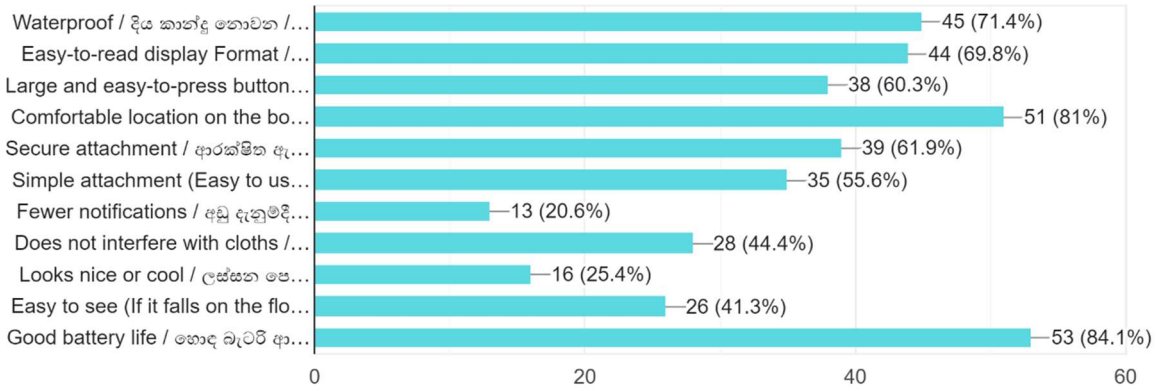

7. **The product technology should be _____**




8. **Compared to our competitors, how do you feel this product should be priced?**

**Mean 10,000 LKR**

9. **What are the features (apart from accuracy) you would consider important when you buy this product?**




10. **Any comments or suggestions?**

- **Thank you for this effort**
- **What if it is something wearable like a watch? Can you attach a watch to this?**
- **good luck!**
- **Maybe a notification system for the guardian to notify when the elderly person fell down, instead of all your suggestions where to wear it may be on the wrist or ankle would be great.**
- **An emergency call added on top of this**
- **Assumed that you are planning a fall detection system. Wrist (like a wristwatch) would be a better place than the given options if a commercial product is intended.**
- **Very useful product.**
- **Good luck with your timely, innovative idea. Hope it becomes available in the market soon.**
- **A good product will be very useful. From a public health perspective, must be effective and the price must be within reach. Or need to get the government to get it for the elderly. This can happen if the product proves to be very useful.**
- **Shouldn't break in case it hits the ground**
- **if you can give a robotic solution through your device itself that is better**
- ශරීරයට අනවශ්‍ය බරක් හා අපඅපහසුතාවයක් ඇති නොකරන භාවිතයට සරල ක්‍රමවේද සහිත නිෂ්පාදනයක් විය යුතුයි.

Appendix II

## Possible impact factors

**Possible costs and benefits associated with an inclusive design project. Each section examines the factors associated with one stakeholder.**

| Is this significant for my project? | Impact Factor | Type of Cost / Benefit | How might inclusive design impact it? | Case studies (ref numbers) | General figures (ref numbers) |
|---|---|---|---|---|---|
| | | | **Stakeholder: Strategy** | | |
| ■ | Long-term development costs | Decreased cost | Inclusive design can reduce long-term development costs by reducing the need for architectural redesign to make future versions of the product easier to use. | | |
| □ | Revenue for other parts of the business | Increased revenue | Some designs can encourage sales of related products or the use of related services. | 25 | |
| ■ | Meeting business objectives | Other benefit | Inclusive design can help to meet Corporate Social Responsibility (CSR) goals and other aspects of the business objectives. | 5,11 | |
| □ | Other | | | | |
| | | | **Stakeholder: Development** | | |
| ■ | Cost of change (making changes to the design) | Decreased cost | Inclusive design can help to detect required changes early on (especially when these are due to usability and inclusivity problems). Changes are much easier and cheaper to make early in the development process. | 1,33 | 1,2 |
| ■ | Cost of project slippage (delayed time to market) | Decreased cost | Inclusive design often reduces the risk of project slippage and reduces development timelines. This is because there is often less need for rework, as rework is often required due to incorrect design and miscommunication. In addition, the validity of designs can be confirmed at early stages of the project. | 31 | 4 |
| □ | Documentation cost | Decreased cost | Inclusive products are often easier to explain and thus to document. | 26 | |
| ■ | Development cost | Decreased or Increased cost | Inclusive design can reduce overall development costs by helping developers to identify and focus on the relevant functionality, by detecting required changes early and by reducing the need for rework. There are also development costs associated with conducting inclusive design (e.g. staff costs, cost of running usability studies) but these are often outweighed by the overall cost reduction. Both of these should be considered in the cost-benefit calculation. | 14,15 | |
| ■ | Appropriate functionality | Other benefit | Inclusive design enables you to prioritise the functionality that's important for customers rather than spending time developing undesired and unused features | 31 | 4,5 |
| □ | Risk management | Other benefit | Inclusive design can help to improve risk management by testing the validity of designs early in project development. | | 3 |
| ■ | Development effectiveness | Other benefit | Inclusive design can help to encourage good development practices, e.g. accessible websites need to be programmed in a clearer and more structured way. | | |
| □ | Other | | | | |
| | | | **Stakeholder: Manufacturing** | | |
| ■ | BOM (Bill of Materials) | Decreased or Increased cost | Some designs require different materials to be used in manufacturing the product. If this is the case, then the change in the cost of the materials should be considered. | | |
| ■ | Tooling costs | Decreased or Increased cost | Different designs require different tooling set-ups which may cost different amounts. In particular, making a change to an existing product may incur retooling costs. | | |
| ■ | Production costs | Decreased or Increased cost | The cost of producing each item (in addition to materials and tooling) may also be affected. | | |
| □ | Other | | | | |
| | | | **Stakeholder: Sales** | | |
| ■ | Advertising costs | Decreased cost | Inclusive design can improve marketability and thus reduce advertising costs. A positive press response and user response to an inclusive product can also reduce the need for advertising. | | 6 |
| ■ | Retail costs | Decreased cost | Some projects may affect retail costs, e.g. making the stock management systems easier to use may improve staff efficiency and reduce retail costs. | | |
| ■ | Sales | Increased revenue | Inclusive design can result in increased sales for a variety of reasons. It increases the customer base and the likelihood of repeat customers. In addition, designs that are easy to use and meet the users' needs often "demo better and sell better". In the particular sector of e-commerce, websites that are designed to be usable and inclusive enjoy an increased visitor count and conversion rate. | 5,6,7,9,11, 12,15,18, 21,22,25, 27,37,38, 39,40,41 | 7,8,9 |
| ■ | Customer base | Other benefit | Inclusive design increases the customer base by enabling more people to use the product or service. This includes both people with reduced abilities and those who are limited by their environment (e.g. by noise, cold or limited bandwidth). | 5,11,21 | |
| □ | Repeat customers | Other benefit | Usable and inclusive designs are more likely to result in satisfied customers who will return and use the service again or buy another product from the company. | 22 | 10,11,14 |
| ■ | Brand | Other benefit | Inclusive design can improve the brand image and market perceptions of a company. | 21 | |
| ■ | Differentiation | Other benefit | Inclusive design can help a product to stand out, particularly in a saturated market. | 8,21 | |
| ■ | Longevity | Other benefit | Inclusive designs may last longer in the market (reducing long-term development costs and increasing the revenue from each version of the product) | | |
| ■ | Word-of-mouth recommendation/ condemnation | Other benefit | Inclusive design often leads to more customer satisfaction and thus to more word-of-mouth recommendation. It also avoids the word-of-mouth condemnation that often arises from poor usability. | 16 | |
| ■ | Press response | Other benefit | Usable and inclusive designs can generate a positive press response (e.g. positive reviews), and avoid the negative publicity that can come from usability problems. | 2,8,11 | |
| ■ | Ease of finding the desired products (specific to e-commerce sites) | Other benefit | An inclusively designed e-commerce website is easier to navigate and thus it is easier to find the desired products. This leads to an increase in sales. | 20,36 | 12,13 |
| □ | Other | | | | |
| | | | **Stakeholder: Operations** | | |
| □ | Back office costs | Decreased cost | Some projects may affect back office costs, e.g. making distribution systems easier to use may improve staff efficiency. | | |
| ■ | Cost of returns | Decreased cost | Inclusive design can reduce the rate of returns, as some products are returned because the customers find them too difficult to use. | 6 | |
| ■ | Support costs | Decreased cost | Inclusive products cause less usability problems and hence generate less support calls. However, note that introducing any change to a system can initially increase the number of support calls before they settle down to the new daily rate. | 10,12,15, 17,18,19, 35 | 11,12,13 |
| □ | Training costs (internal) | Decreased cost | Fewer training materials and less training time are needed for more usable systems. | 4,15,28,29 | |
| ■ | Maintenance costs | Decreased cost | Inclusive systems are often better structured and thus easier to maintain. | 16 | 18 |
| ■ | System costs (servers, databases, etc to keep the product running) | Decreased cost | The system costs vary a lot depending on the nature of the product but need to be taken into account. An inclusive design example is that an inclusive website with clear and consistent navigation reduces unwanted page downloading and thus reduces bandwidth and server load. | 16 | |
| ■ | Legal costs | Decreased cost | Inclusive products reduce the risk of legal action (and negative publicity) associated with inaccessibility. | 23,24 | |
| ■ | User/employee satisfaction and productivity | Other benefit | More inclusive and usable systems can be used more efficiently by staff. In addition, less time is wasted by experienced staff providing assistance when new users encounter difficulties. Inclusive systems are also often more pleasant to use, increasing user and staff satisfaction. | 3,13,14,16, 30,32,34, 55,56,57 | |
| □ | Other | | | | |

Appendix III

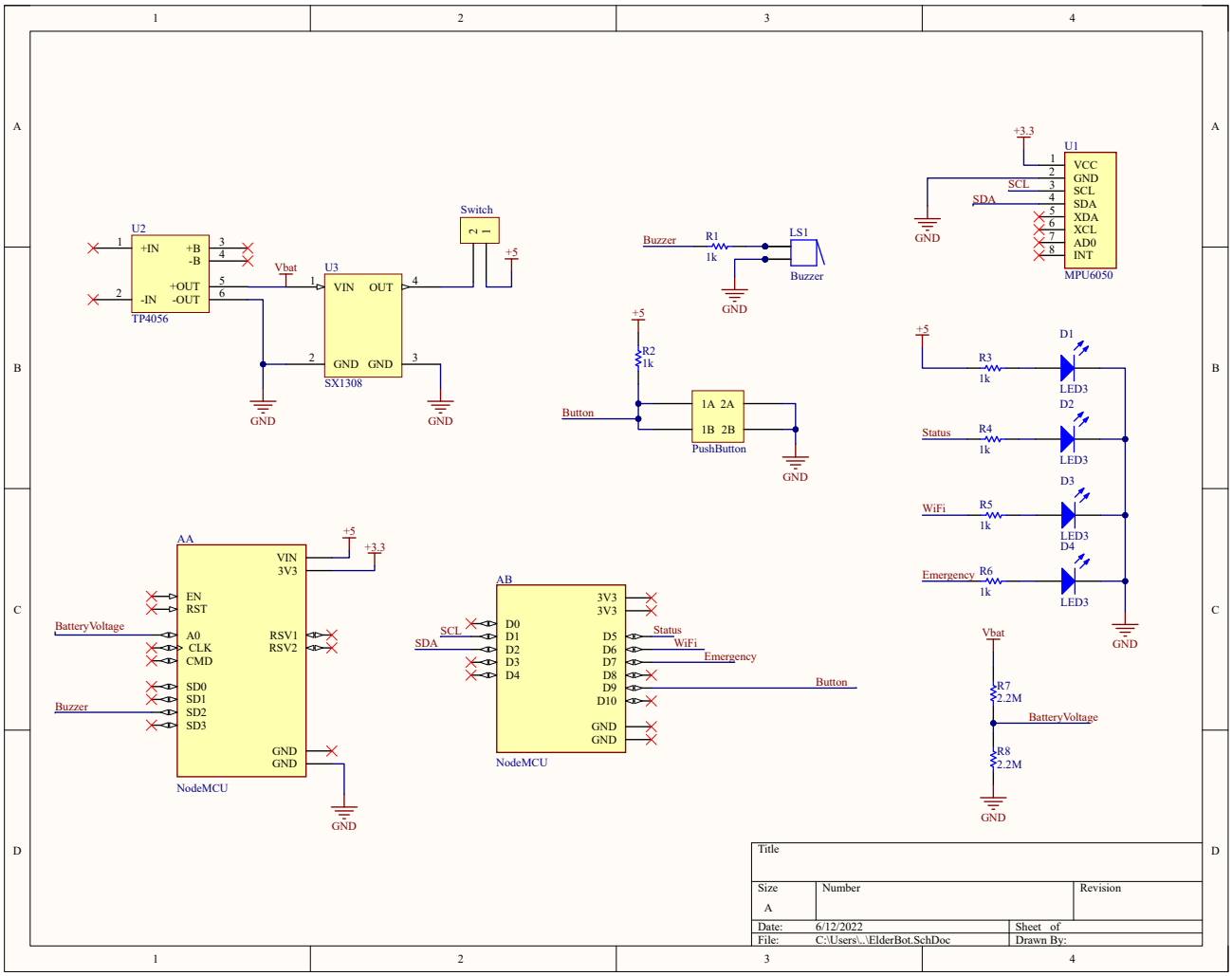
**Initial assessment**

# Initial impact assessment

**Project Name: Elderbot** **Date:** 08/04/2022

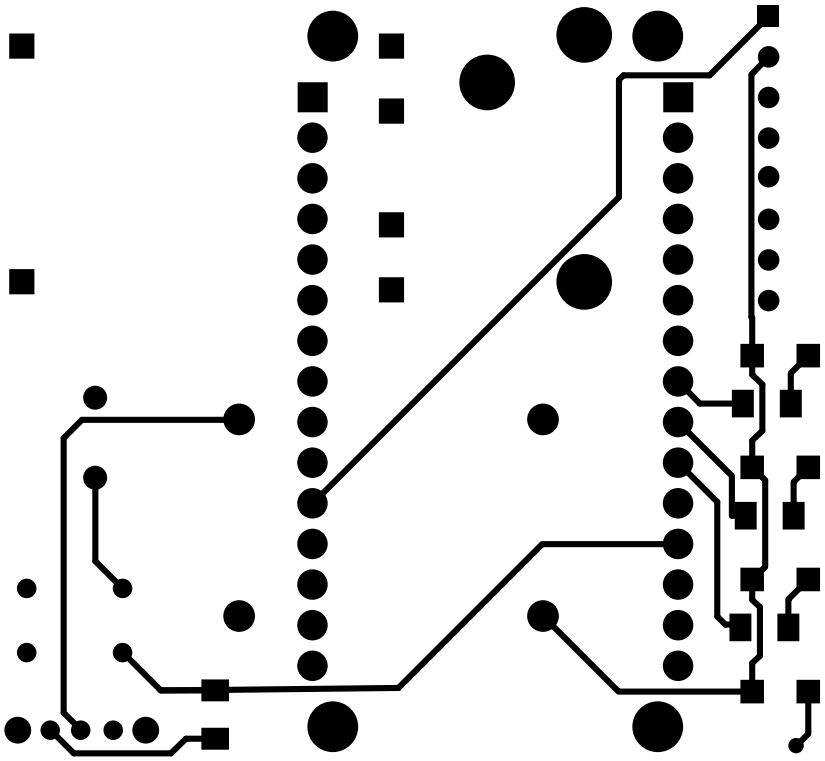
| Impact Factor Descriptions | | | | Size of Impact (choose from drop-down list) |
|---|---|---|---|---|
| **Stakeholder** | **Impact Factor** | **Type of Cost / Benefit** (choose from drop-down list) | **Rationale** | |
| Example<br>Operations | Support | Decreased cost | Fewer calls to support helpline as many calls are about usability problems | High |
| Strategy<br>< Possible factors: Long-term development costs, Revenue for other parts of the business, Business objectives, Other > | Long term development costs | Decreased cost | HCD, ACD and Inclusive design princip | Medium |
| Development<br>< Possible factors: Cost of change, Time to market, Documentation, Development costs, Other > | Development costs | Increased cost | Higher development costs lead to increased cost per unit of Elderbot. | High |
| Manufacturing<br>< Possible factors: Bill of Materials, Tooling, Production, Other > | BOM | Increased cost | Higher total in the BOM leads to higher cost per unit of Elderbot | High |
| Sales<br><Possible factors: Advertising, Retail, Sales, Brand, Press response, Longevity, Other > | Brand | Increased revenue | More brand recognition leads to higher sales and revenue | Medium |
| Operations<br>< Possible factors: Back office, Returns, Support, Training, Maintenance, System costs, Legal issues, User/Employee productivity, Other > | Returns | Decreased revenue | More returns from customers lead to low revenue, and damages the product brand | High |
| | | | | |

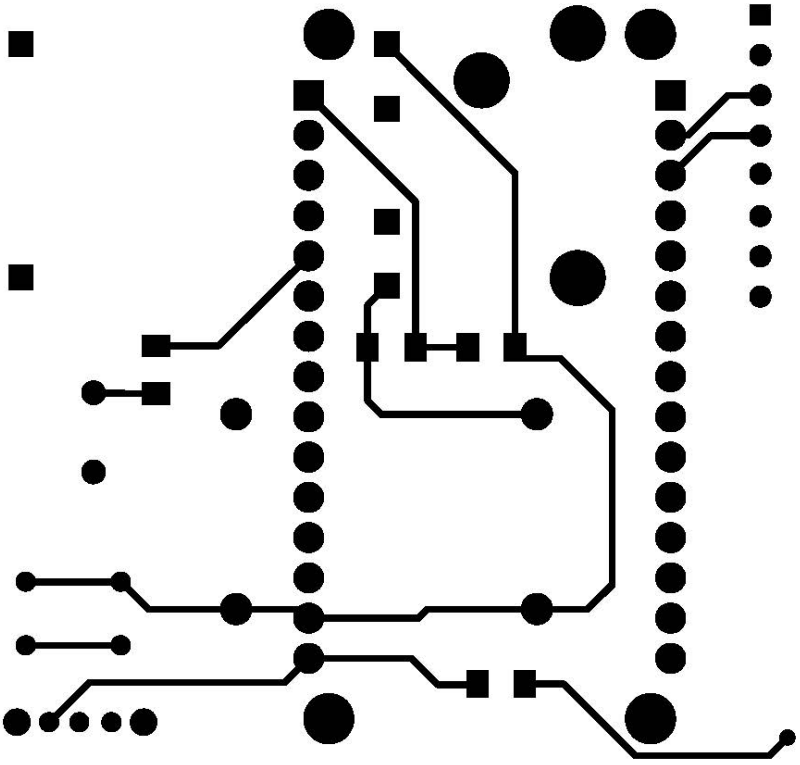
Appendix IV : Schematic Design



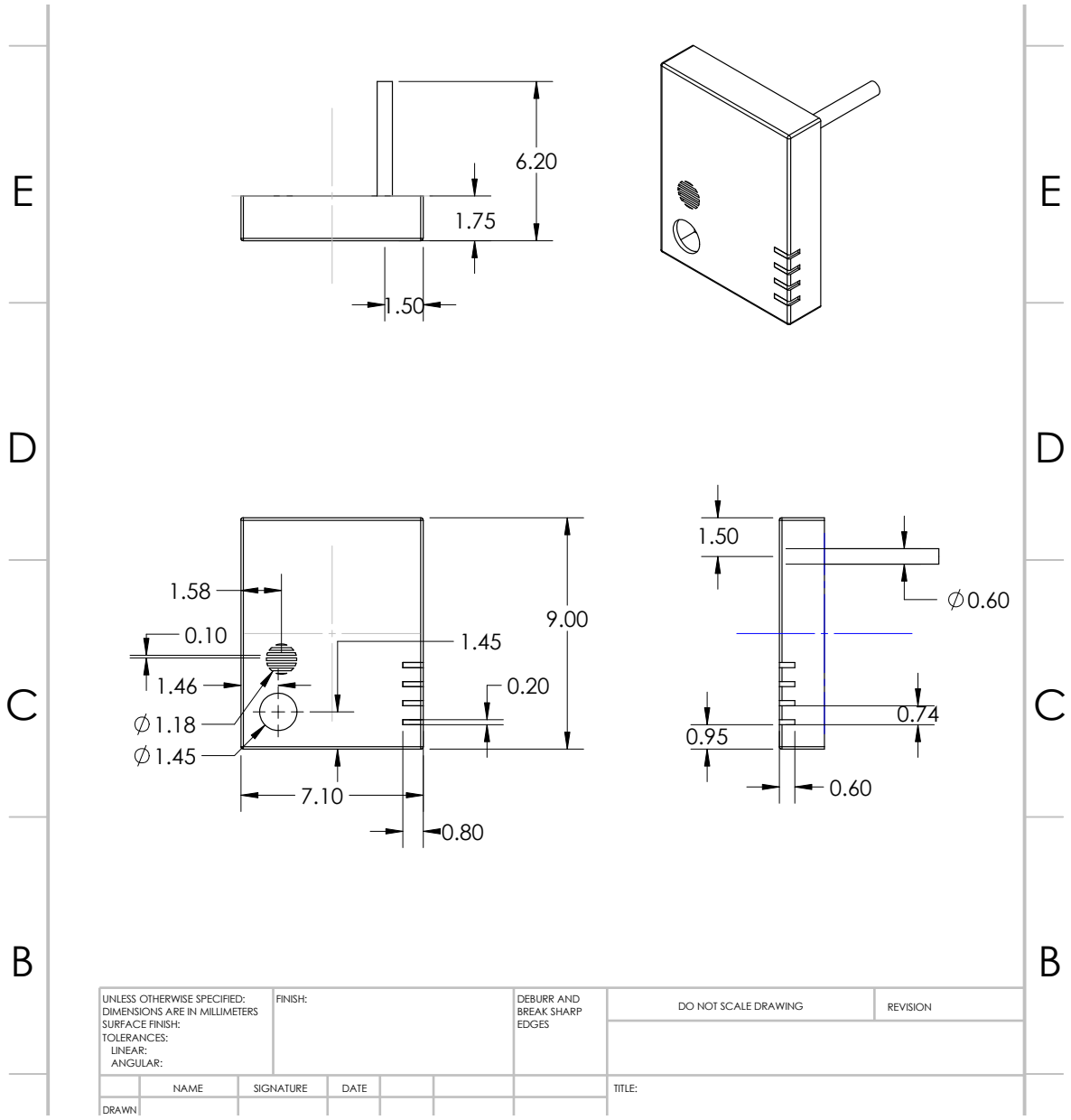
CAMtasticDXP (TM):

Appendix V: PCB layout files

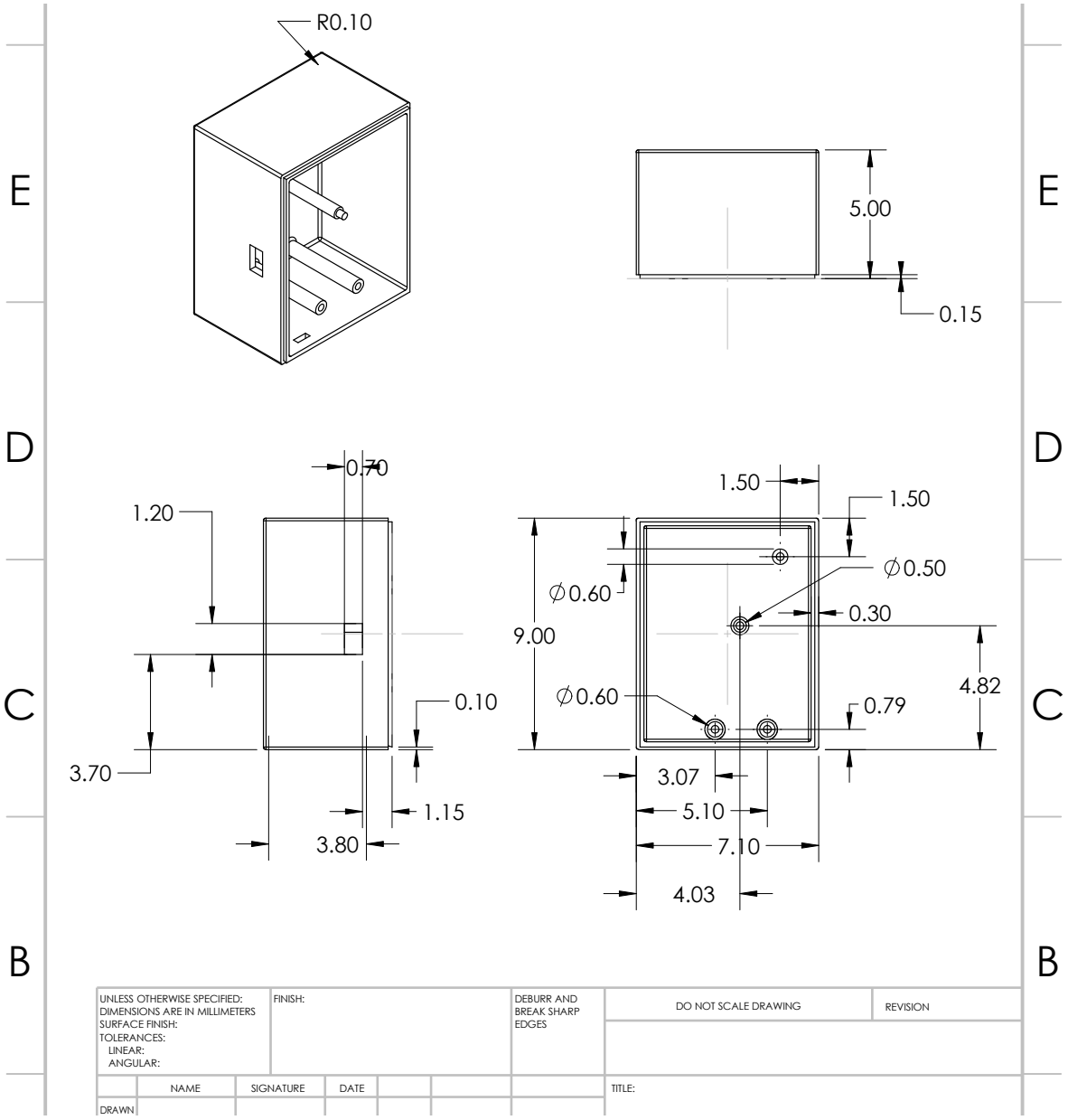




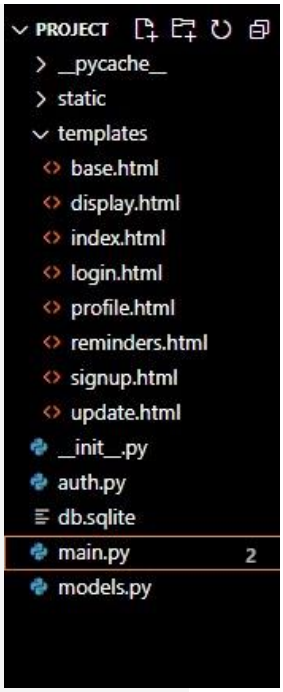
Appendix VI: Engineering drawing Upper lid



Appendix VII: Engineering drawing bottom part



**Appendix VIII Web Interface Main.py**




```
from flask import Blueprint, render_template, redirect, url_for,
request, flash
from flask_login import login_required, current_user
from .models import User
from . import db

main = Blueprint('main', __name__)

@main.route('/')
def index():
    return render_template('index.html')

@main.route('/profile')
@login_required
def profile():
    user = User.query.filter_by(email=current_user.email).first()
    tele = user.tel_num
    if tele == '0000000000':
        msg = "You haven't entered your phone number, Please enter it!!!"
    else:
        msg = "You have already entered your phone number, If you want to
update it please enter a new number!!"
    return render_template('profile.html', name=current_user.name, messg =
msg)

@main.route('/profile', methods=['POST'])
def profile_post():
    telephoneNumber = request.form.get('Telephone number')
    user = User.query.filter_by(email=current_user.email).first()

    user.tel_num = telephoneNumber
    db.session.commit()

    return redirect(url_for('main.profile'))

@main.route('/reminders')

@login_required
def reminders():
    return render_template('reminders.html')

@main.route('/reminders', methods=['POST'])
def signup_post():
    # code to validate and add user to database goes here
    description = request.form.get('description')
    time = request.form.get('time')
```

```
    user = User.query.filter_by(email=current_user.email).first()
    # create a new user with the form data. Hash the password so the plaintext
version isn't saved.    #current_user = User(alarm1=alarm1, time1=time1)
    count = user.medication_count
    if count == 0:
        user.medication_0_description = description
        user.medication_0_name = time
        user.medication_count = user.medication_count + 1
        db.session.commit()
        return redirect(url_for('main.display'))
    elif count == 1:
        user.medication_1_description = description
        user.medication_1_name = time
        user.medication_count = user.medication_count + 1
        db.session.commit()
        return redirect(url_for('main.display'))
    elif count == 2:
        user.medication_2_description = description
        user.medication_2_name = time
        user.medication_count = user.medication_count + 1
        db.session.commit()
        return redirect(url_for('main.display'))
    elif count == 3:
        user.medication_3_description = description
        user.medication_3_name = time
        user.medication_count = user.medication_count + 1
        db.session.commit()
        return redirect(url_for('main.display'))
    elif count == 4:
        user.medication_4_description = description
        user.medication_4_name = time
        user.medication_count = user.medication_count + 1
        db.session.commit()
        return redirect(url_for('main.display'))
    elif count == 5:
        user.medication_5_description = description
        user.medication_5_name = time
        user.medication_count = user.medication_count + 1
        db.session.commit()
        return redirect(url_for('main.display'))
    elif count == 6:
        user.medication_6_description = description
        user.medication_6_name = time
        user.medication_count = user.medication_count + 1
        db.session.commit()
        return redirect(url_for('main.display'))
    elif count == 7:
        user.medication_7_description = description
```

```
        user.medication_7_name = time
        user.medication_count = user.medication_count + 1
        db.session.commit()
        return redirect(url_for('main.display'))
    elif count == 8:
        user.medication_8_description = description
        user.medication_8_name = time
        user.medication_count = user.medication_count + 1
        db.session.commit()
        return redirect(url_for('main.display'))
    elif count == 9:
        user.medication_9_description = description
        user.medication_9_name = time
        user.medication_count = user.medication_count + 1
        db.session.commit()
        return redirect(url_for('main.display'))
    else:
        flash('You have reached the maximum number of reminders, Please update
unwanted reminders in the display TAB')
        return redirect(url_for('main.reminders'))

@main.route('/display')
@login_required
def display():
    return render_template('display.html', user=current_user)

@main.route('/update')
@login_required
def update():
    return render_template('update.html')

@main.route('/update/<int:id>', methods=['POST','GET'])
def update_post(id):
    #id = request.args.get('id')
    # code to validate and add user to database goes here
    if request.method == 'POST':
        description = request.form.get('description')
        time = request.form.get('time')
        user = User.query.filter_by(email=current_user.email).first()

        # create a new user with the form data. Hash the password so the
plaintext version isn't saved.
        #current_user = User(alarm1=alarm1, time1=time1
        if id == 0:
            user.medication_0_description = description
            user.medication_0_name = time
            db.session.commit()
            return redirect(url_for('main.display'))
```

```
        elif id == 1:
            user.medication_1_description = description
            user.medication_1_name = time
            db.session.commit()
            return redirect(url_for('main.display'))
        elif id == 2:
            user.medication_2_description = description
            user.medication_2_name = time
            db.session.commit()
            return redirect(url_for('main.display'))
        elif id == 3:
            user.medication_3_description = description
            user.medication_3_name = time
            db.session.commit()
            return redirect(url_for('main.display'))
        elif id == 4:
            user.medication_4_description = description
            user.medication_4_name = time
            db.session.commit()
            return redirect(url_for('main.display'))
        elif id == 5:
            user.medication_5_description = description
            user.medication_5_name = time
            db.session.commit()
            return redirect(url_for('main.display'))
        elif id == 6:
            user.medication_6_description = description
            user.medication_6_name = time
            db.session.commit()
            return redirect(url_for('main.display'))
        elif id == 7:
            user.medication_7_description = description
            user.medication_7_name = time
            db.session.commit()
            return redirect(url_for('main.display'))
        elif id == 8:
            user.medication_8_description = description
            user.medication_8_name = time
            db.session.commit()
            return redirect(url_for('main.display'))
        elif id == 9:
            user.medication_9_description = description
            user.medication_9_name = time
            db.session.commit()
            return redirect(url_for('main.display'))
    else:
        return render_template('update.html', id = id)
```

**Appendix IX MCU code**

```
#include "FS.h"
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <PubSubClient.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <stdio.h>
#include <string.h>
#include <cstring>
#include <SoftwareSerial.h>

SoftwareSerial mySerial(D7, D8);

const char* ssid = "SLT_FIBER_ruhunage";
const char* password = "Sandeepa@44";
//char* UserInfo ="";
bool needWiFi = true;
char str[32] = { 0 };
String Data;
const char* AWS_endpoint = "a3diuj9r5yc78a-ats.iot.us-east-2.amazonaws.com";
long lastMsg = 0;
char msg[50];
char* Sens_Read="0";
char* Status="Normal";             // States: Normal / Fallen / Emergency
int value=0;
const char* D2S = "D2S";
const char* S2D = "S2D";
String Device_ID_Per="Yasod_ESP";
String Device_ID="0";
String Phone_Num="None";
String MSG="None";
bool New_Message=false;
int Buzzer=6;
uint8_t Emergency = D3;


void callback(char* topic, byte* payload, unsigned int Length)
{                     // Take the incomming MQTT Messages and update NodeMCU
global variables
  String Data_in = "";
  for (int i = 1; i < Length-1; i++) {
    Data_in += (char)payload[i];
  }
    Phone_Num = getValue(Data_in, ',', 0);
    MSG = getValue(Data_in, ',', 1);
    char msgr[30]="MQTT receievd";
    Serial.println(msgr);
```

```
    Serial.println(Phone_Num);
    Serial.println(MSG);
    New_Message=true;
}
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");
WiFiClientSecure espClient;
PubSubClient client(AWS_endpoint, 8883, callback, espClient);

void setupWiFi(){
  delay(10);
  espClient.setBufferSizes(512, 512);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.print(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED){delay(500); Serial.print(".");}
  Serial.println("");
  Serial.println("WiFi connected");
  timeClient.begin();
  while (!timeClient.update()) {timeClient.forceUpdate();}
  espClient.setX509Time(timeClient.getEpochTime());
}

void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    if (client.connect("Yasod_ESP")) {
      Serial.println("connected");
      //client.publish("D2S", "MQTT Connected");
      client.subscribe("SMSCenter");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");

      char buf[256];
      espClient.getLastSSLError(buf, 256);
      Serial.print("WiFiClientSecure SSL error: ");
      Serial.println(buf);
      delay(5000);
    }
  }
}

bool initWifi()
{                                                                              /
/ Reinitializing WiFi connectivity after awaking
```

```
  WiFi.begin(ssid, password);
  delay(500);
  int count=0;
  while((count<50) &&
(WiFi.status()!=WL_CONNECTED)){digitalWrite(16,HIGH);delay(30);digitalWrite(16
, LOW);delay(30);count++;}
  return WiFi.status()==WL_CONNECTED ? true : false;
 }
String getValue(String data, char separator, int index)
{
    int found = 0;
    int strIndex[] = { 1, -1 };
    int maxIndex = data.length() - 1;
    for (int i = 0; i <= maxIndex && found <= index; i++) {
        if (data.charAt(i) == separator || i == maxIndex) {
            found++;
            strIndex[0] = strIndex[1] + 1;
            strIndex[1] = (i == maxIndex) ? i+1 : i;
        }
    }
    return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}

String SMS_Command="";
int SendSMS(String Number,String Message)
{
  Serial.println("Sending SMS...");               //Show this message on
serial monitor
  mySerial.println("AT+CMGF=1");
  delay(500);
  //snprintf(SMS_Command,32, "AT+CMGS=\"%s\"\r", Number);
  //mySerial.println(SMS_Command);
  mySerial.println("AT+CMGS=\"+94703716801\"\r");
  delay(500);
  mySerial.print(Message);
  delay(500);
  mySerial.write((char)26);
  mySerial.println("AT+CMGF=1");
  delay(500);
  mySerial.println("AT+CMGS=\"+94773604959\"\r");
  delay(500);
  mySerial.print(Message);
  delay(500);
  mySerial.write((char)26);
  Serial.println("Text Sent.");
  delay(500);
}
```

```
void setup() {

  setupWiFi();
  mySerial.begin(9600);
  Serial.begin(9600);
  delay(500);
  if (!SPIFFS.begin()) {Serial.println("Failed to mount file system");
return;}
  Serial.print("Heap: "); Serial.println(ESP.getFreeHeap());

  File cert = SPIFFS.open("/cert.der", "r");
  if (!cert) Serial.println("Failed to open cert file");
  else Serial.println("Successfully opened cert file");
  delay(1000);
  if (espClient.loadCertificate(cert)) Serial.println("cert loaded");
  else Serial.println("cert load failed");

  File private_key = SPIFFS.open("/private.der", "r");
  if (!private_key) Serial.println("Failed to open private_key file");
  else Serial.println("Successfully opened private_key file");
  delay(1000);
  if (espClient.loadPrivateKey(private_key)) Serial.println("private_key
loaded");
  else Serial.println("private_key load failed");
  File ca = SPIFFS.open("/ca.der", "r");
  if (!ca) Serial.println("Failed to open ca file");
  else Serial.println("Successfully opened ca file");
  delay(1000);
  if (espClient.loadCACert(ca)) Serial.println("ca loaded");
  else Serial.println("ca load failed");
  Serial.print("Heap: "); Serial.println(ESP.getFreeHeap());
  delay(1000);
  reconnect();
  client.setCallback(callback);
}

void loop() {
  if (!client.connected()) {
      reconnect();
    }
  client.loop();
  delay(100);
  if(New_Message){
    SendSMS(Phone_Num,MSG);
    New_Message=false;
  }
}
```