# Project Deliverable J - User Manual

**GNG2101**

**Professor Mana Azarm**

**Group name: Node-14**

**Group E14 - SensAct User Interface**

**Anika Prottasha 300119095**
**Fiona Chau 300132210**
**Omar Sherif 300127063**
**Joshua Ekwueme 300117661**
**William Wang 300128012**

**November 28, 2020**

Abstract

A user-interface has been created using node-red and its dashboard, to achieve our goal of creating a simple interface to allow patients to easily communicate with their surroundings during their stay in the hospital. In this user manual, the problem, and the solution will be described in detail, as well as our process in creating the product. This document will be an advanced user guide, that will allow the reader to understand our steps in building the solution, and how it all works.

# Table of Contents


1. **Introduction** ……1
   - the problem ……..……1
   - the solution …….…….1
   - Why Choose us …….……..……..1
2. **Features/Functions/Capabilities** …….……2
   - Figure 1:Patient Systems Main Menu of Interface……..2
   - Figure 2: Instructions to Choose Devices……....3
3. **Concept generation and Final Concept Choice** ……4
   - Table 1: Grouping of generated concepts……...…..……4
   - Figure 3: Final Concept designed on Miro Board ……..5
   - Table 2: BOM of Product ……...……..……..……6
   - Figure 4: Example of Node-Red Code For Basic Menu ……7
   - Figure 5: Sample Image of Basic Screen in UI Prototype ……8
4. **Prototype 1 Testing Summary** …….……...……..9
   - Table 3: Prototype 1 Testing Summary …...……9
   - Figure 6: Node-Red Interface for Second Prototype with Inclusion of Serial Ports ……...10
   - Figure 7: Main Menu of The Interface ……..11
   - Figure 8: The No Touch Device Menu ……11
   - Figure 9: The No Vision Device Menu ……12
   - Figure 10: The No Sound Device Menu ……...12
   - Figure 13: The Extra Settings Menu ……13
   - Figure 14: The Intermediate Functionalities Menu ……..13
   - Figure 15: The Advanced Functionalities Menu ……..14
   - Figure 16: The Final Options Menu ……14
5. **Prototype 2 Testing Results** ……...……..……...……...16
   - Table 4: Prototype 2 Testing Results ……....16
6. **Final Completed Product** ……...……...17
   - Figure 17: Patient Systems Main Menu of Interface ……17
   - Figure18: No-Touch Menu …….……..17
   - Figure 19: No vision menu …….……...……...18
   - Figure 20: No Sound Menu of Interface ……...18
   - Figure 21: Limited Mobility menu ……...18
   - Figure 22: Basic functionalities menu ……..19
   - Figure 23: Extra Settings menu …...……19

- Figure 24: The Intermediate Functionalities Menu ……..19
- Figure 25: The Advanced Functionalities Menu ……..20
- Figure 26: The Final Options Menu ……..20
- How It Works ..……..……21
- How Users navigate our Interface ……..…..21
- Installation/Operation/Maintenance ……..……22
    - Figure 27: Serial Out Node Properties……..…….……22
    - Figure 28: Image File Properties ……...……...23
- Potential Improvements…….……23
- Troubleshooting ……...…….24
- Design Files ……...……...24

7. **Conclusion and recommendations** ……...……...……..26
8. **Bibliography** …….……...…….27
9. **Appendix** ……..……....29

# 1. Introduction

Bocar N'diaye from the Saint-Vincent Hospital currently uses a SensAct device to help patients with disabilities communicate. SensAct is a device that allows the user to connect varying inputs (typically sensors) through a user interface, to easily communicate and use devices such as a computer. The current user-interface requires background knowledge of coding and electronics. For example, the user needs to understand and create loops for each movement that is detected by the SensAct and defines the signal which can be hard to understand. The main goal of this project is to design a neat, highly accessible User Interface for hospital staff to utilize on a SensAct device to improve basic communication with patients.

## 1.1 The Problem

Without access to a simple interface, it will become extremely difficult for a nurse or a healthcare professional to set up the desired configuration for the patient to use. This will result in an inefficient work environment and patient discomfort. Our problem is important because it allows for a solution to be developed that will increase both work efficiency and patient comfort, which is necessary for hospitals and other healthcare facilities.

## 1.2 The Solution

Our User Interface requires little to no knowledge of electronics/coding and it also accommodates different disability needs, customized for each patient. It takes the user an average of 4 buttons from start to finish to get to the designated device to be controlled, and the user interface is very easy to use due to the high level of organization on each screen. Our solution was designed to be intuitive enough for a patient's family member with little to no background in coding or electronics to be able to set-up and configure the Sensact device.

## 1.3 Why Choose us

One of the main priorities of our design was to come up with an interface that is highly accessible as this is what many current solutions lack. We have included images beside the buttons to represent the button functionality along with audio information queues which read out each essential button function. This makes our user interface more

accessible than other UI's in the sense that people who cannot see or cannot hear can still use the User Interface at its full functionality.

## Features/Functions/Capabilities:

Our User Interface was built completely on Node-Red and has several features. To begin, our interface is divided into different phases. Firstly it is divided into 4 sections on the main screen, being different patient systems to fit the specific customized needs of each patient. We have built a no-touch system, no vision system,  no sound system, and limited mobility to accommodate the needs of all patients using our product. On the next screen, there is a choice for the user to select between a basic setting, intermediate, or advanced in which these 3 were built with an incremental increase in customization/difficulty, and the more advanced setting would be used by a technician or hospital staff that may have a bit of training or understanding with the interface. The basic setting on the other hand is very simple, easy to use, and built to be less customizable. It was designed with a patient's family member in mind, meaning there are only "base functions" provided. After selecting a difficulty setting, the user is given the choice of selecting what devices they would want to use, along with the functionality of the device. Figure 2 below shows one of the screens for device choices which contain devices like a touch sensor and joystick which can be controlled by the user. After selecting a device/functionality, the user is prompted to either return to the menu, which takes the user to the first screen (patient system) or to turn off the device which closes the program. It takes 4 buttons/choices to take the user from the first menu to the screen where the user is controlling the selected device, and we designed the product to ensure the UI is as simple and easy to understand as possible.

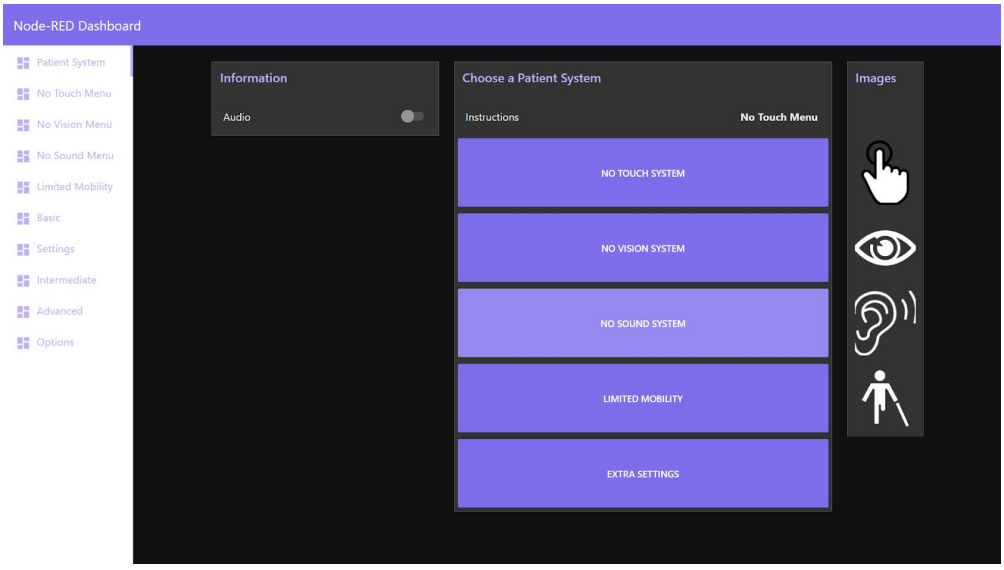


Figure 1:Patient Systems Main Menu of Interface

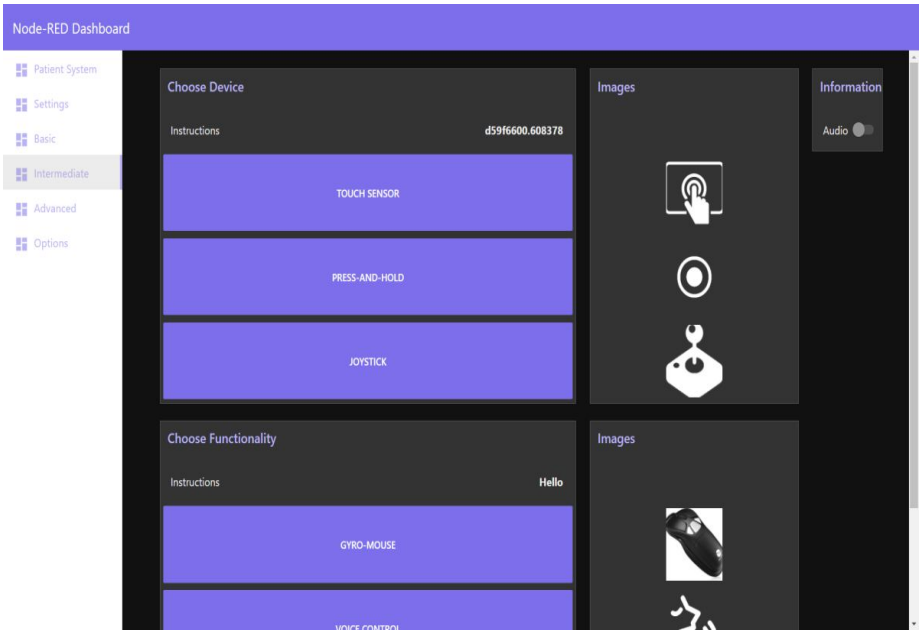


Figure 2: Instructions to Choose Devices

We used 3 main button types to design our interface. These buttons are push buttons which the user clicks to take them to a new corresponding screen, menu buttons which are on the left side panel of each screen and are shortcuts that take the user to different screens quickly without going through several button choices, and lastly slider buttons which are used for playing audio callouts. Each button has texts written over it for a brief explanation of the button function. The right-hand side of each screen contains a column of images which are placed at the side of each push button and aids in providing a visual representation of each button function and was designed to help pictorial learners to easily understand what each button will do when pushed. For example, in our first screen seen in Figure 1, there is a finger to indicate that the system pushed is “no touch system”, and an ear to indicate that is the “no sound system”. We also have included inbuilt- audio callouts for any users that may be visually impaired which are played when the user pushes the audio slider. These audio callouts are on each screen, and briefly explain the functions of each button on each page using a Female Cortana (Windows) voice. We designed the script for the Cortana voice to read out, and this can easily be edited/changed using the Node-Red (Dashboard) audio functions.

# Concept generation and Final Concept Choice:

Table 1: Grouping of generated concepts

| Subsystems |
| --- |
| Concept 1-Focus on Usability |
| Menu system |
| Limited options at once |
| Buttons with pictures |
| Customized based on disabilities |
| Basic and Advanced categories |
| Help button for more information |
| Audio Instructions |
| Concept 2-Only Essentials |
| Pre-programmed common functions (one button) |
| Commands through voice recognition (like Siri) |
| Limit options - Only focus on the most essential functionality |
| Concept 3-Focus on Customization |
| Tutorial videos teaching workers commands |
| ability to change actions that prompt a function (to suit patients) |
| Choose commands - specified options for each command |
| Replace call bell input with the camera to monitor patients |

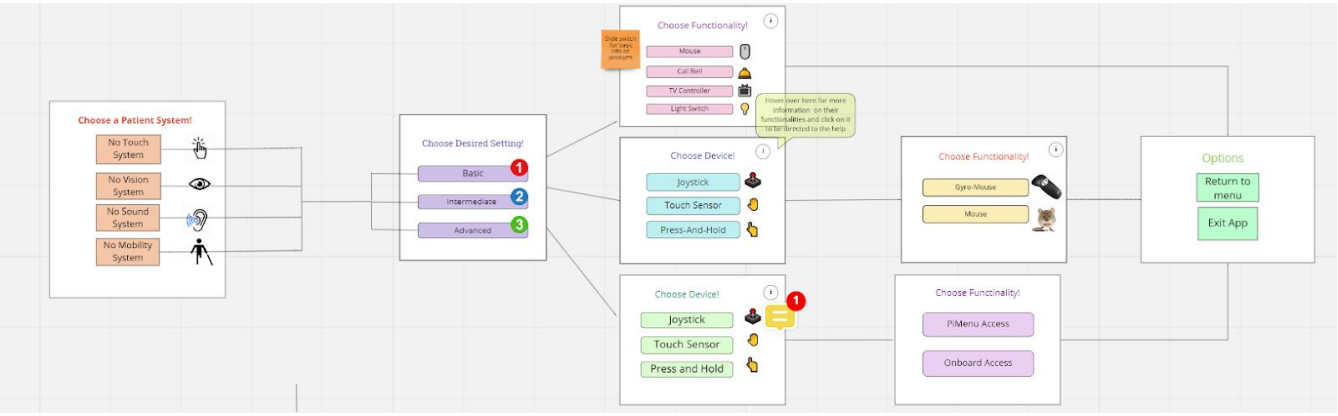


Figure 3: Final Concept designed on Miro Board

After completing a functional decomposition of our product to better understand what it needs to do, during the concept generation phase, our team came up with several design concepts that focused on improving the current SensAct Interface. We came up with solutions like integrating a voice recognition and response system (like Siri) to our User Interface, or even integrate a video-camera to the UI which helped monitor the patients and in turn, could give a much simpler UI since the video camera would carry out most of the intended device functions. We assessed our ideas based on the constraints of the project and the problem we were supposed to solve - creating a simple, customized and accessible UI which can be improved and developed on - and we decided to group similar concepts in the concept generation table and came up with three main concepts; Basic, Usability and Customized. We used a decision matrix to determine that a focus on usability would be the most well-suited concept for our customer requirements. Our final concept involved having different systems to meet the needs of different patients' disabilities (disability systems) along with systems for more knowledgeable users on coding ( intermediate/advanced systems), and the second was to integrate picture (images) and audio callouts alongside each of our buttons for more clarity of each button's functions. After selecting the patient's disability, the user selects how advanced the options would be for device functionality (patients would be using the basic system) and then finally the user chooses the device they want to control and its functionality before being prompted to return to the first menu or turn off the device.

Table 2: BOM of Product

| Item number | Name | Description | Quantity | Cost | Extended Cost |
|---|---|---|---|---|---|
| 1 | Node-Red Software | Our project is done on the Node-Red software, which is a free, flow-based visual programming tool. | 1 | $0 | $0 |
| 2 | SensAct user interface | Our interface will be designed to connect directly to the SensAct device, where it will perform the desired functionalities. This will be provided to us by our client. | 1 | $0 | $0 |
| 3 | Raspberry pi | One of the main expectations of our client was for our device to be functional on a raspberry pi. So we will be using a raspberry pi to test device functionality. A Raspberry pi available on the market ranges from $25-$35, but for this project, we will be able to use one we already have from previous classes. | 1 | $0 | $0 |
| 4 | Stock Images | For our version, we were able to use free images from google, however, if our product were to be recreated in a professional setting, we would require rights to stock images. | 10 | $4.90 | $49 |
| | | | | Total | $49 |

As seen in Table 1, the creation and development of each prototype cost 0 dollars. The platform we developed the Interface on was a free online-based dashboard platform. One can essentially create a whole User Interface for free using Node-Red, but to test the functionality of the User Interface, one would require a SensAct Device or a Raspberry pi( which would most likely be easier to obtain). We got these devices for free, one was given to us by our client for testing, and the other was obtained from a friend, making the creation of each prototype completely cost-free. Replicating this prototype would cost around 30 dollars as one would need to buy the Raspberry pi for testing purposes. A PC is also required to run Node Red's platform on, which can range in prices, averaging usually 500-2000 CAD. When developing our product

professionally in the future we would require additional resources such as Stock Images, whose costs also have to be accounted for, seen in Table 1.

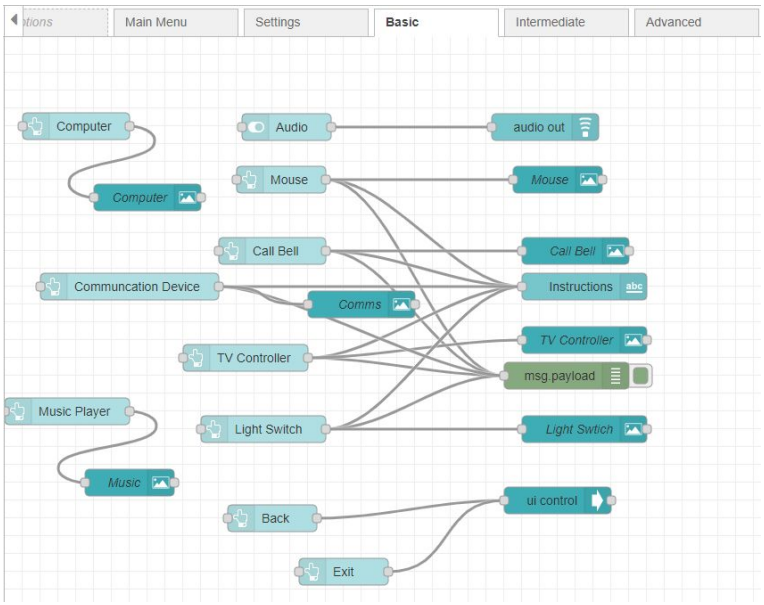


Figure 4: Example of Node-Red Code For Basic Menu

We had 2 main prototyping stages, the first of which was to test how intuitive our design was while the second was to test the functionality of the interface leading up to the final product. After creating a basic design in Miro (as outlined above) the next step was to implement essentially the skeleton into Node-Red. We wanted to ensure that we were able to implement our ideas as our design was created based upon assumptions of what Node-Red was capable of and what we could learn in a short period of time. Thus, our first prototype was a “skeleton” which included all buttons, images and an audio function in order to test how intuitive our design was and the simplicity of it. Images of the prototype and a table of the prototype testing are provided below.

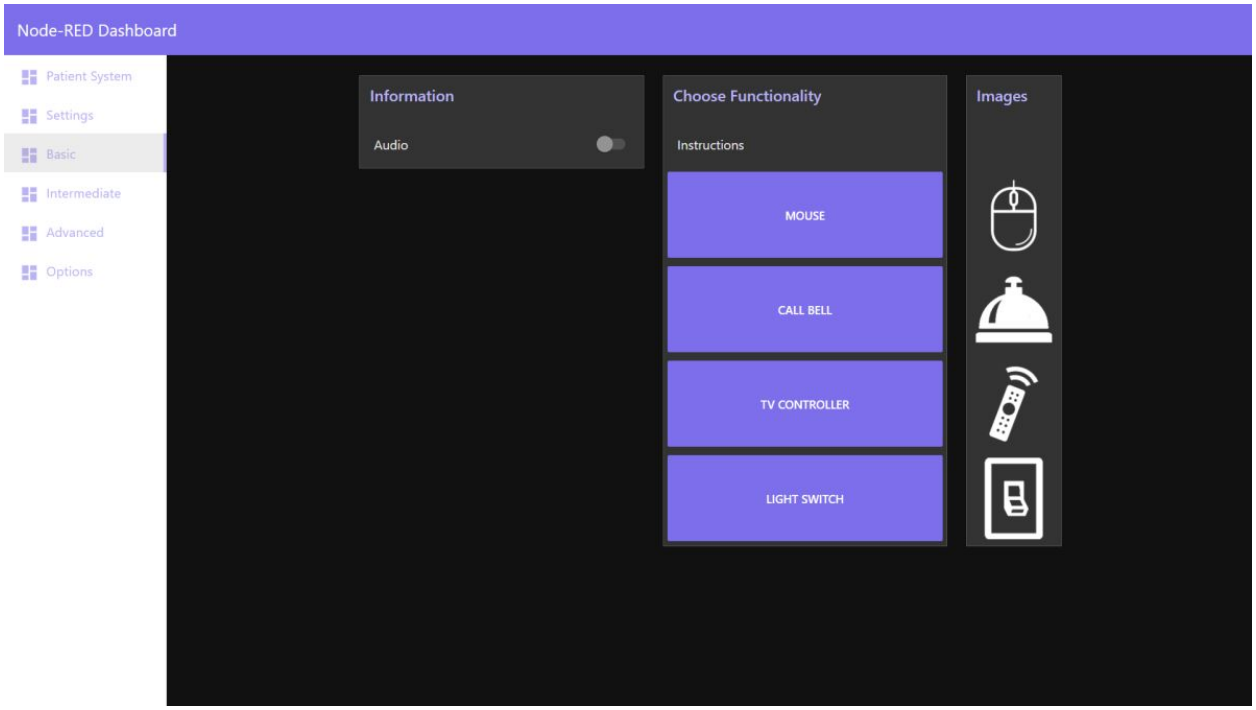


Figure 5: Sample Image of Basic Screen in UI Prototype

# Prototype 1

Table 3: Prototype 1 Testing Summary

| Testing Criteria | Expected Results | Actual Results | Evaluation | Test Units |
|---|---|---|---|---|
| Simplicity | 2-5 | 2-6 | The program is simple, so the screen looks neat and easy to understand despite having more buttons than expected. | # buttons per screen |
| Shortcuts | 6 | 6 | Included 6 shortcuts to each screen on the sidebar of our program. | # shortcuts on sidebar |
| Customizable options | 3 | 3 | The program can run on Mac, Windows, and Raspberry pi. | # devises it works on |
| Easy Navigation | <8 | 7 | 7 clicks to get to the control of a device (not including asking for audio information help). | # clicks to use the desired device |
| Visual Representation | >12 | 18 | Most screens of the interface contain 2-4 images which leads to a total of 18 graphics. | # Of total graphics |
| Low Cost | $0 | $0 | Developing cost $0 as the required software was installed for free. | Money spent on development |
| Intuitivity | 5 | 4 | Feedback: ‘It looks really straightforward to use, some of the images aren’t enough to explain what a device is if the customer doesn’t already know” | Feedback rating from a student with no technical background Scale: 1-5 |

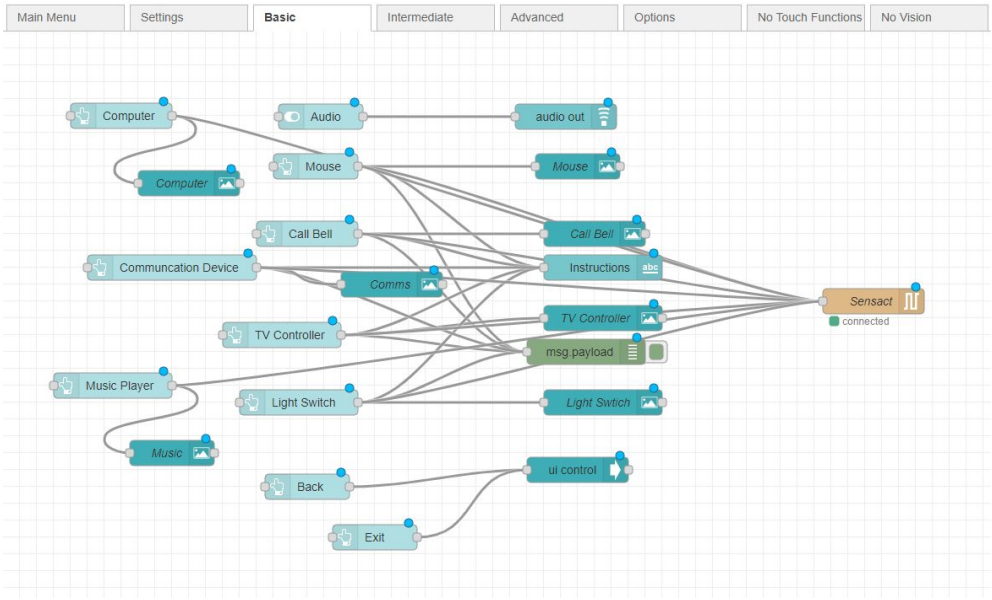


Figure 6: Node-Red Interface for Second Prototype with Inclusion of Serial Ports

Prototype 2 was more heavily focused on functionality. This meant ensuring that the device and our interface were able to communicate with one another and that all the buttons were functioning and brought the user to the correct page. The user will be able to navigate through our interface by clicking buttons. We also added additional buttons and rearranged them to maximize user ease. We did so by grouping specific functionalities that are useful for each patient system. For example, patients with complete dexterity disabilities would not find devices like the Gyro-mouse helpful, so we limited the options to the most helpful devices such as an eye tracker device and voice control commands to ease navigation and suit the users' needs. Our team had some more difficulties with the second prototype as we had difficulty implementing java functionality into Node-Red using the client's java files that he had provided, which is what the SensAct device currently uses. In the second prototype, we added the patient system menus that were not fully implemented in the previous prototype as well as button functionality. We were also able to connect the device with the Node-Red Interface however we're unable to have any communication between the two devices. Images and prototype testing summary are included below:

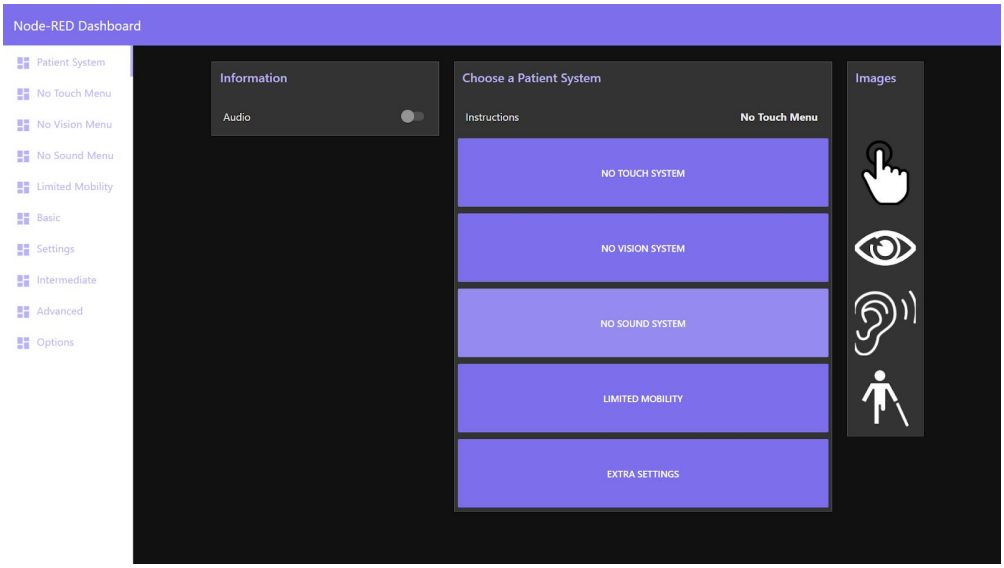


Figure 7: Main Menu of The Interface

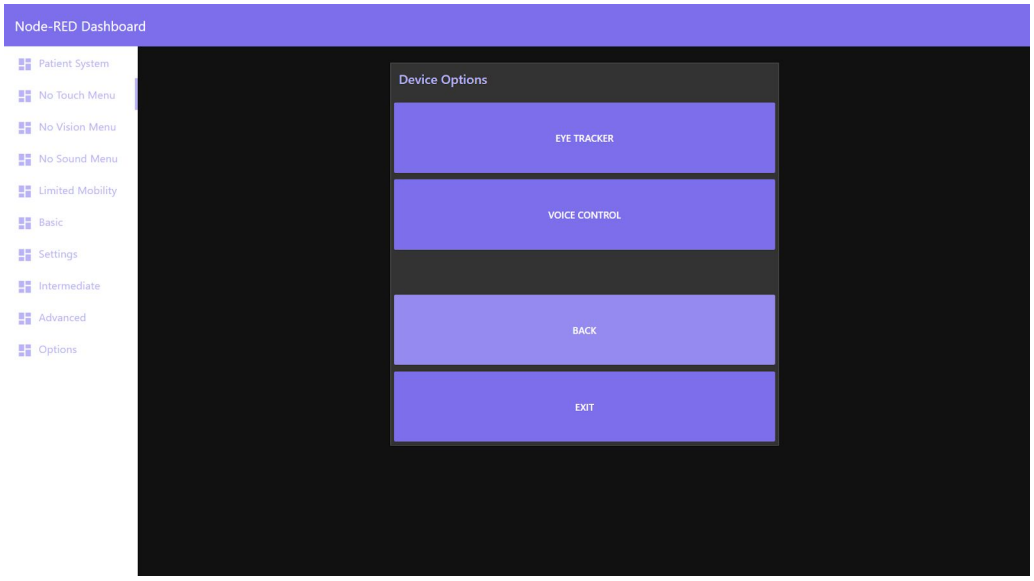


Figure 8: The No Touch Device Menu

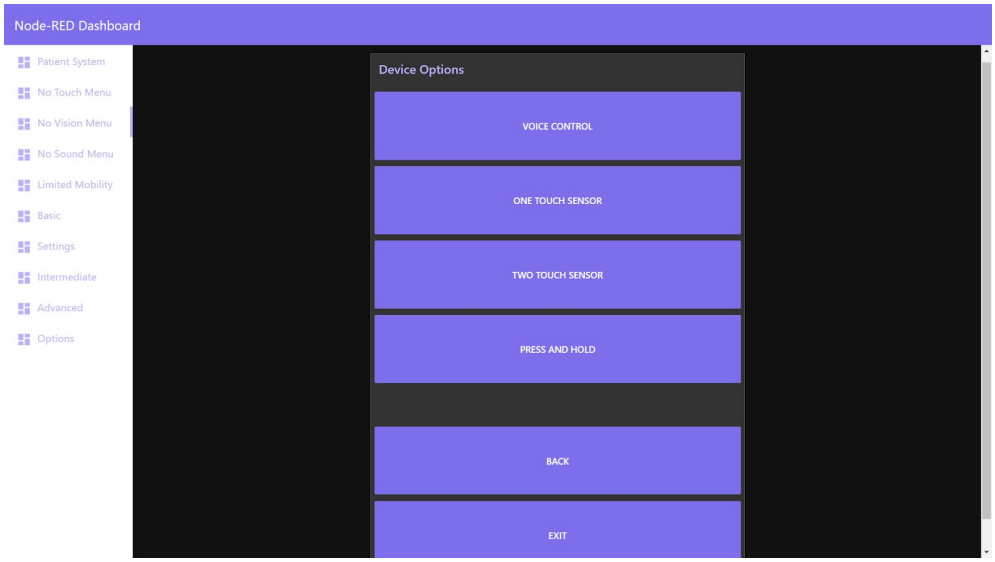


Figure 9: The No Vision Device Menu

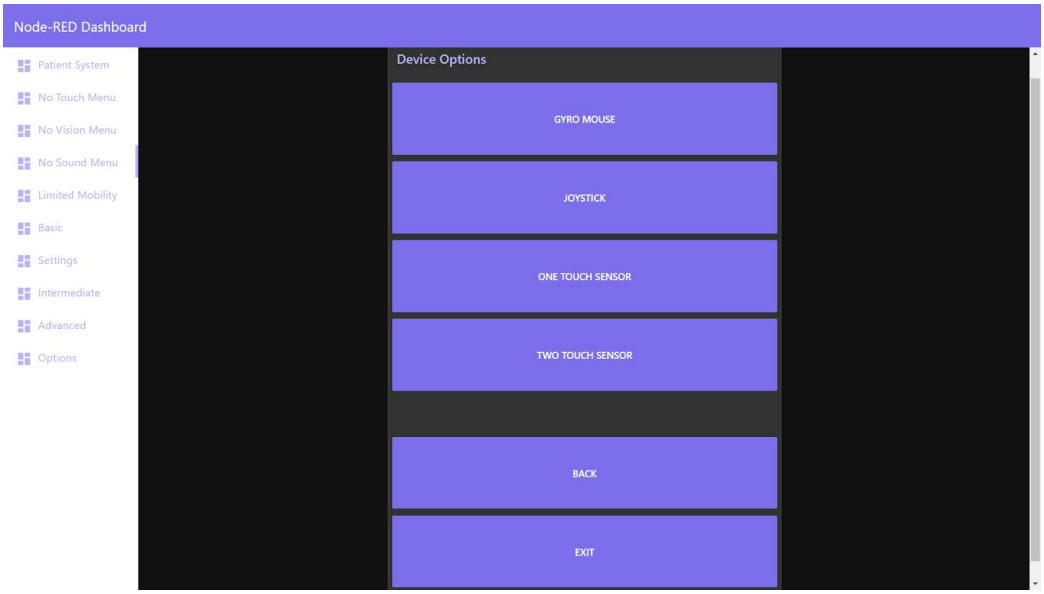


Figure 10: The No Sound Device Menu

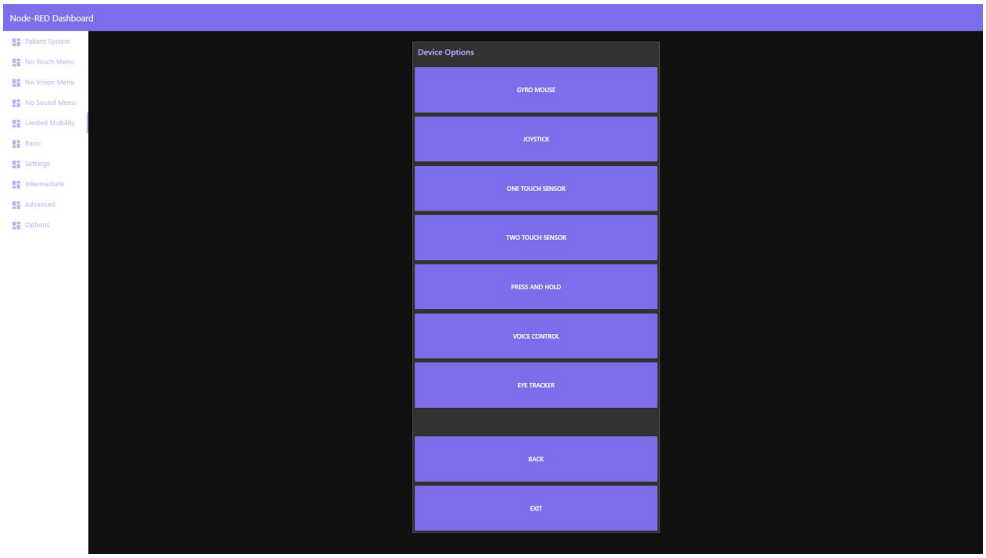


Figure 11: The Limited Mobility Device Menu

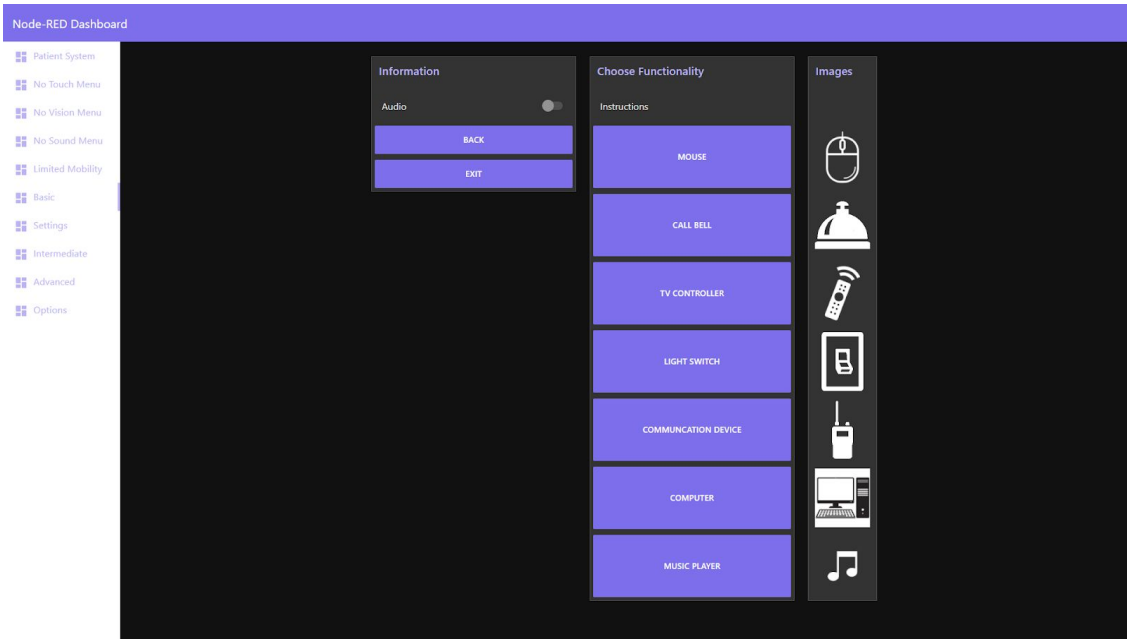


Figure 12: The Basic Functionalities Menu

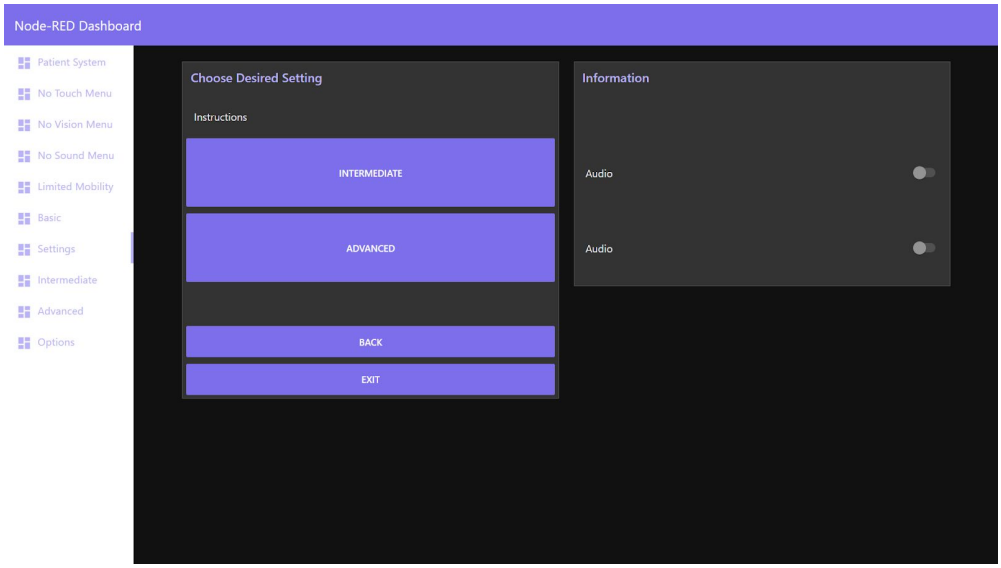


Figure 13: The Extra Settings Menu

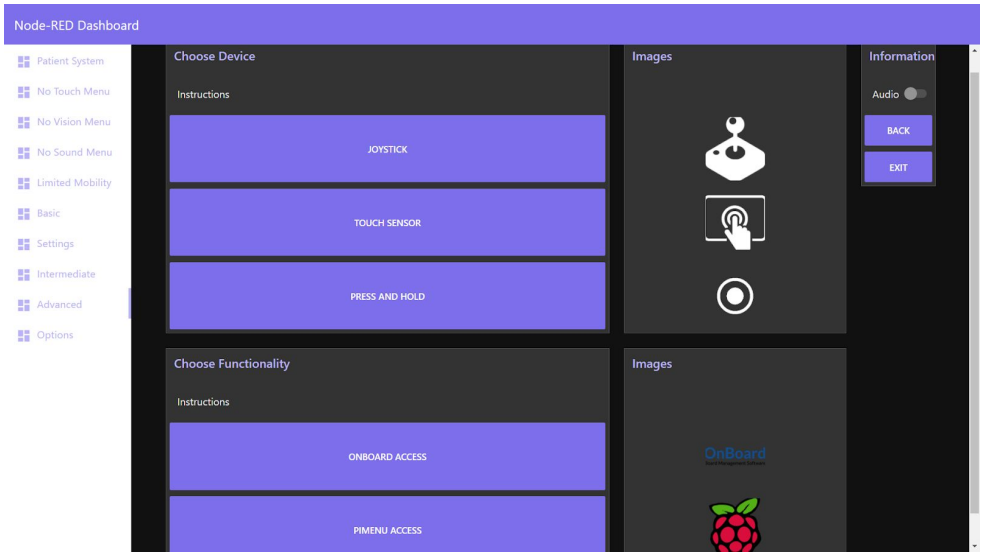


Figure 14: The Intermediate Functionalities Menu

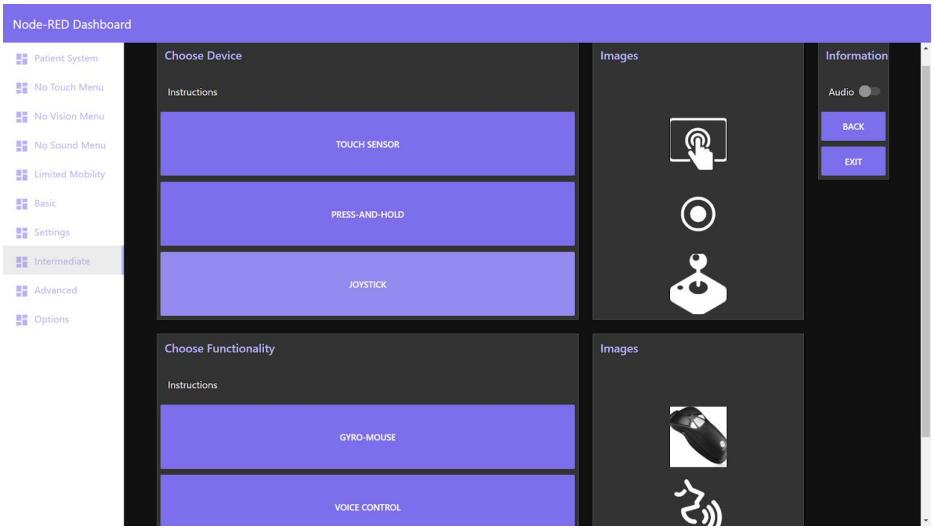


Figure 15: The Advanced Functionalities Menu

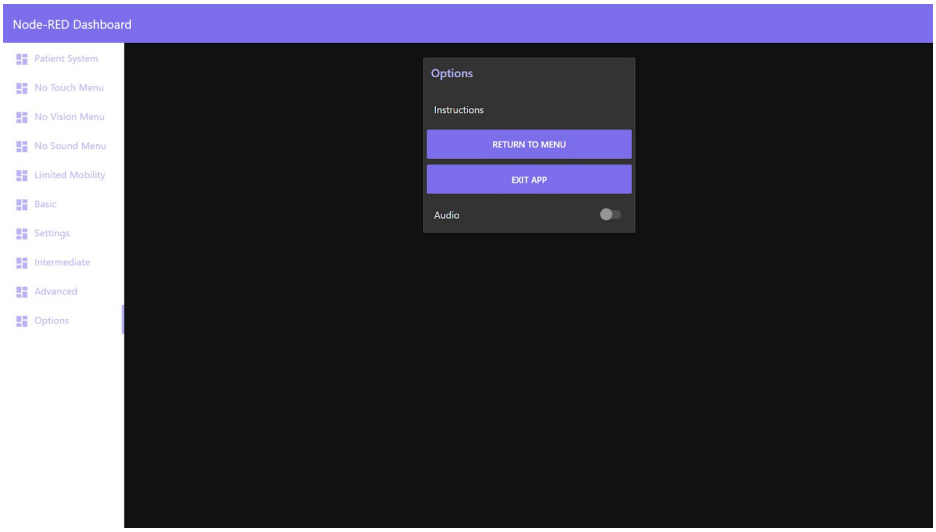


Figure 16: The Final Options Menu

# Prototype 2

Table 4: Prototype 2 Testing Results

| Testing Criteria | Expected Results | Actual Results | Evaluation | Test Units |
|---|---|---|---|---|
| Simplicity | 2-5 | ≥9 | Any screen had at most 9 buttons. May consider removing buttons to make the design more simple | # buttons per screen |
| Shortcuts | 6 | 10 | Value increased greater than expected due to additional patient menus, however, should not affect the usability | # shortcuts on sidebar |
| Easy Navigation [From the main menu to mouse] | <8 | 3 | The result is lower than expected, which means our interface is easy to use | # clicks to use the desired device |
| Visual Representation | >12 | 21 | More graphics for each button/screen due to new screens. | # Of total graphics |
| Intuitivity | 5 | 4 | Feedback:<br>“ Tell the user what to do instead of just providing options instead of ‘eye tracker’, 'Use eye tracker' for example.”<br>“Separated pictures and audio looks kind of confusing”<br>“Menu useful because it lets you see where (what setting) you are currently in”<br>“Audio sliders just say ‘audio’ and seems like it is just one audio option” | Feedback rating from students with no technical backgroun d<br>Scale: 1-5 |

# Final Completed Product

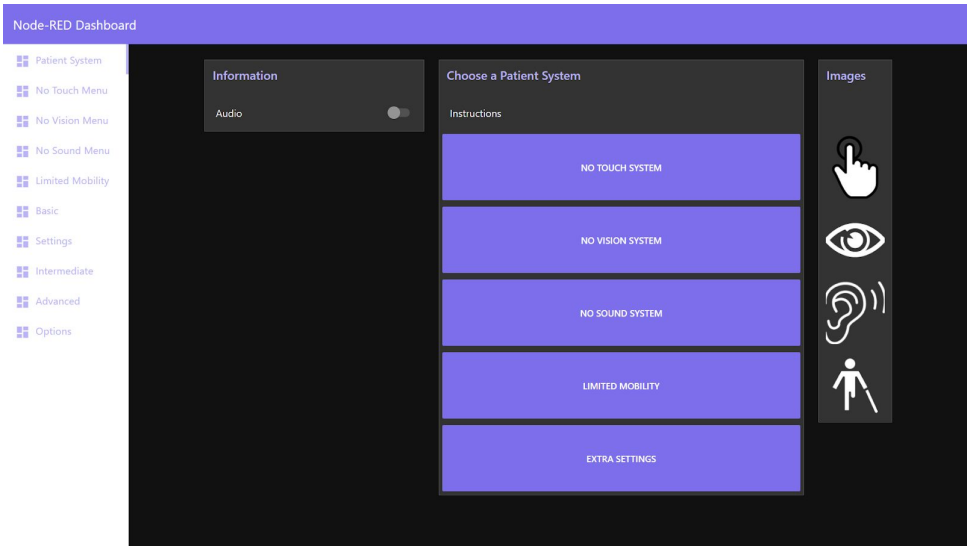


Figure 17: Patient Systems Main Menu of Interface

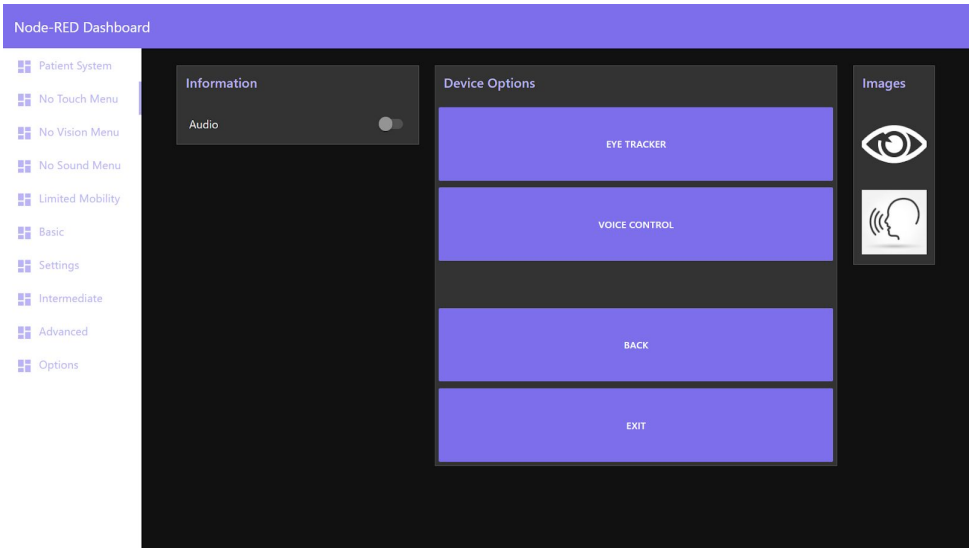


Figure18: No-Touch Menu

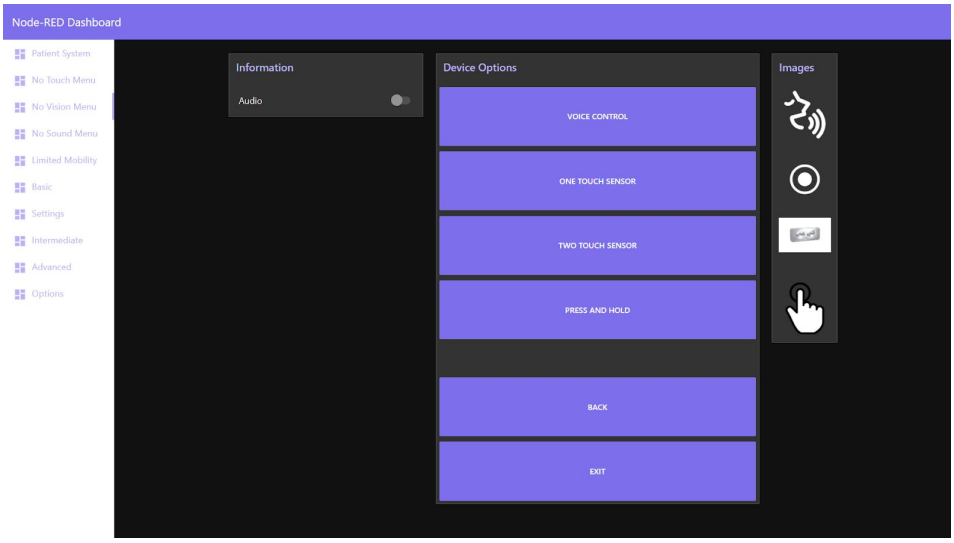


Figure 19: No vision menu

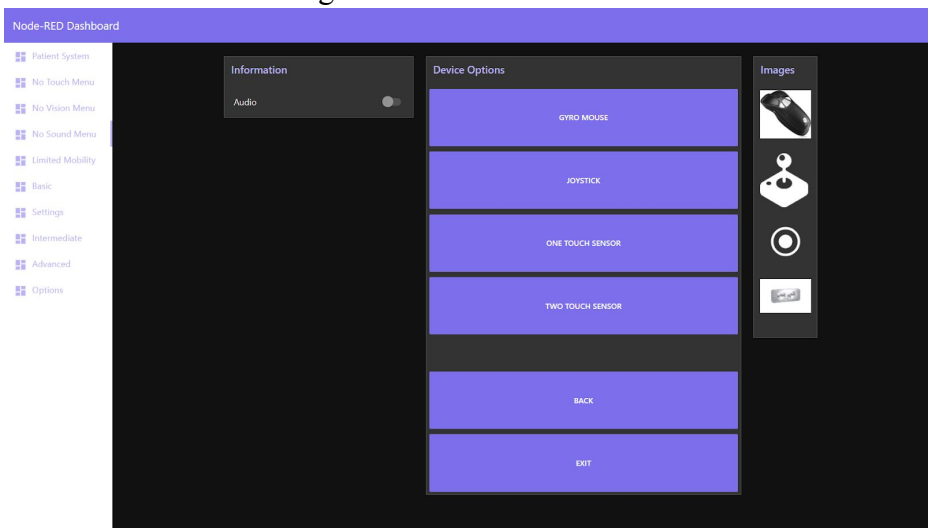


Figure 20: No Sound Menu of Interface

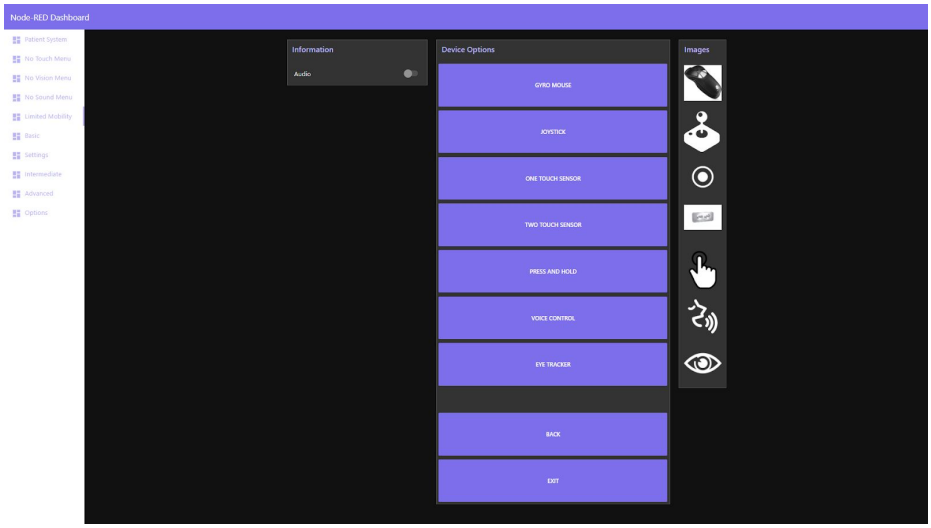

Figure 21: Limited Mobility menu

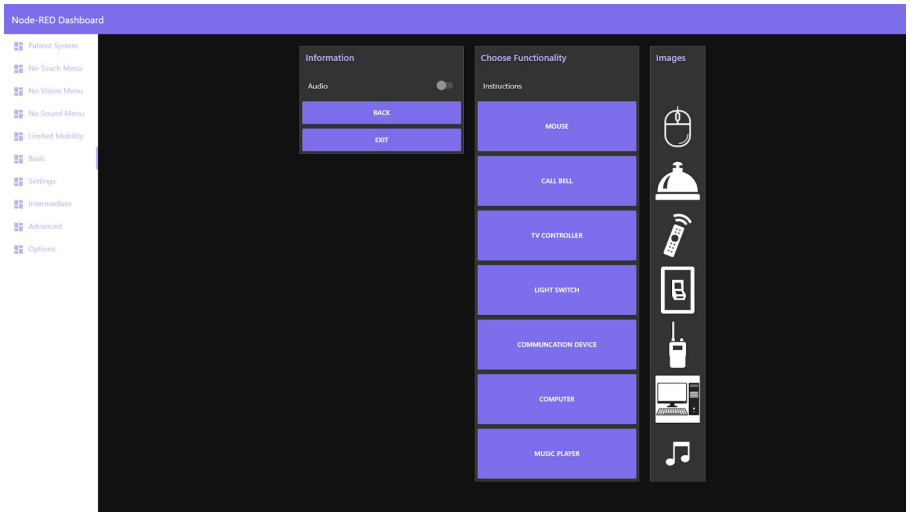


Figure 22: Basic functionalities menu

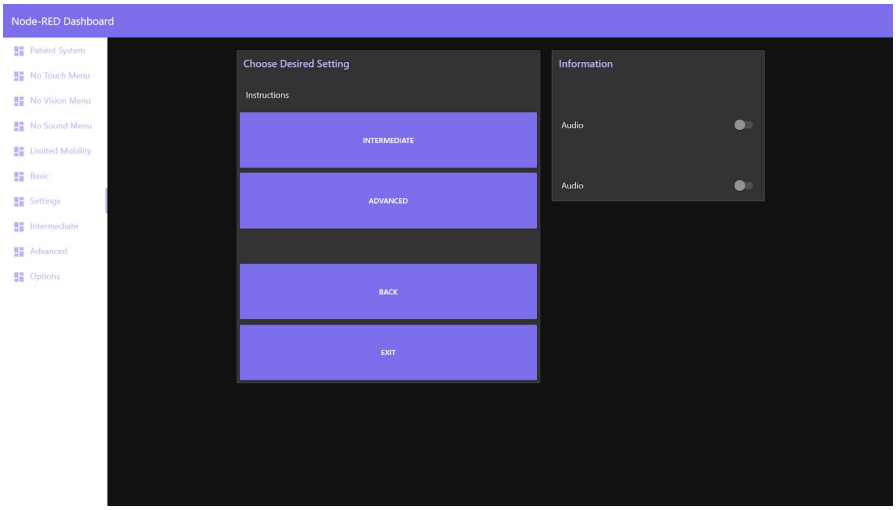


Figure 23: Extra Settings menu

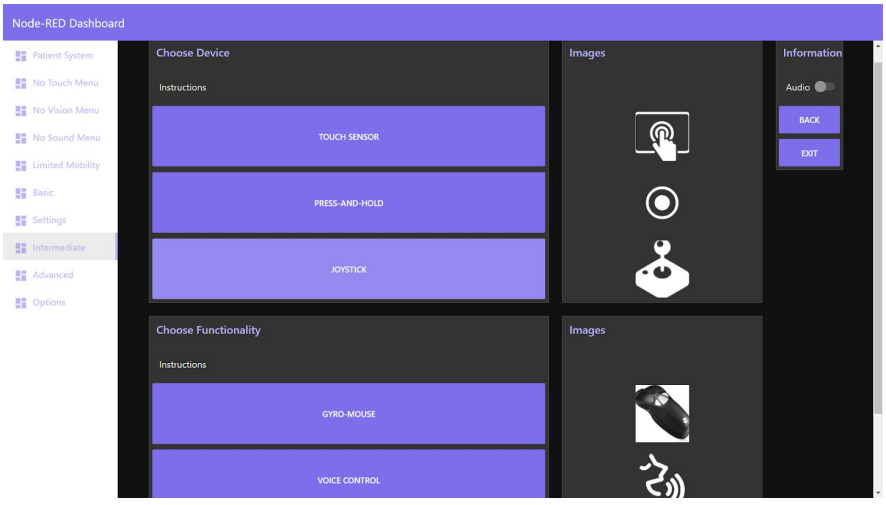


Figure 24: The Intermediate Functionalities Menu

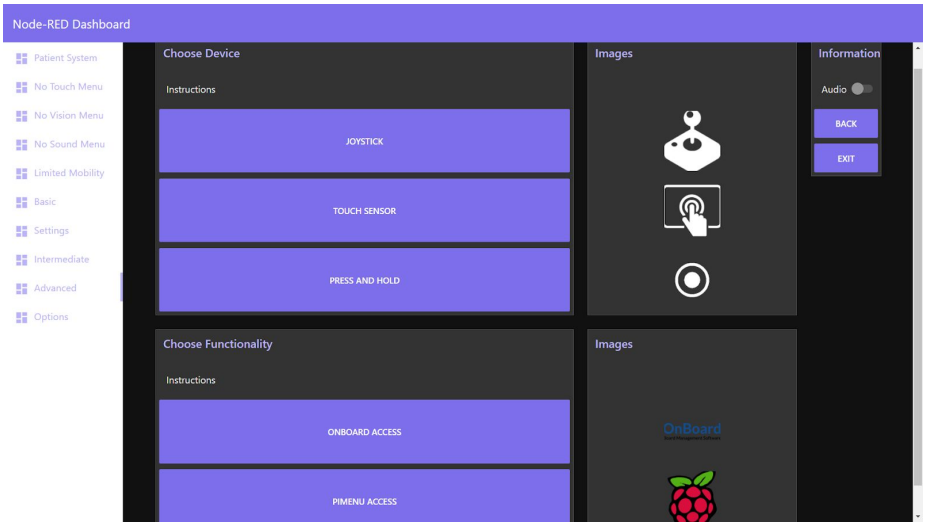


Figure 25: The Advanced Functionalities Menu

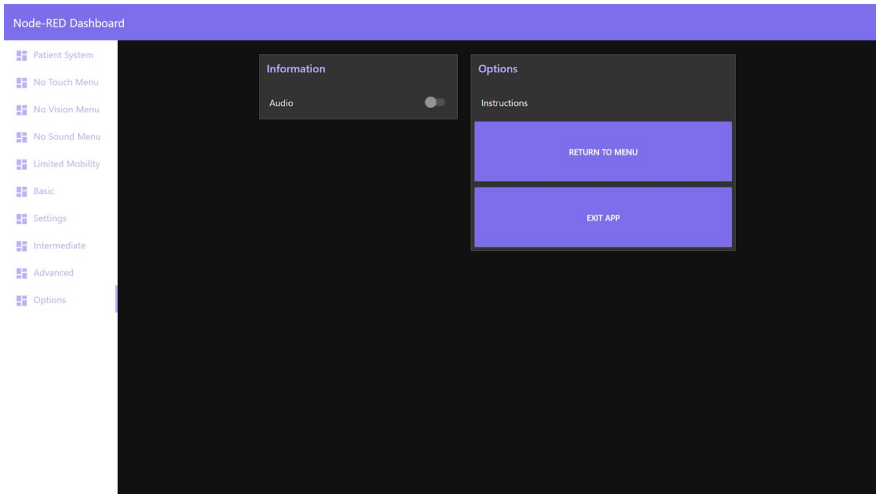


Figure 26: The Final Options Menu

We created our final prototype based on the feedback we received from our previous prototypes. Noticeable changes include pictures and audio added to each of the newly created patient systems and overall adjustments to make sure our product is polished. This prototype ensured we were able to satisfy our client's requirements for our User Interface in order to create a simple

and intuitive User interface that makes it easy for workers to navigate without any technical background. Our interface is able to provide that through helpful features such as a menu, audio and images to guide the workers through our interface, as well as strategic grouping of functionalities with limited options to ensure the workers are able to help their patients fulfill their needs without being overwhelmed. We also added functioning payloads (using the codes our client gave us) to all of our buttons but were still unable to have it function on Java.

## How It Works:

Our interface utilizes NodeRed's capabilities in order to create an intuitive design and experience for our users. In a technical aspect, our interface works by using payloads to communicate with one another. To begin, once the program is deployed in Node-Red the user can use the dashboard to interact with the interface as well as the device. The user interacts with the device through the buttons, each button has a specific payload such that when the button is pressed the payload message is sent to the SensAct device. The Arduino code in the SensAct is looking for specific messages to perform certain actions. For example, all payload messages begin with a "T" so the Arduino board knows it is communicating with it. Thus, when the user clicks on a button, the specific message is sent to the device, and the device will function accordingly.

From a more broad scope, our interface utilizes a serial port to communicate with the SensAct device. The "serial out" node in the interface (as seen in figure _) allows for a connection between the computer port and essentially allows the two to communicate with one another. This serial port connection is what allows the payload message to be delivered as well as any other form of possible communication.

## How Users navigate our Interface:

After following the installation procedure and deploying it to have access to the User Interface, the user will be taken to the main menu of the interface where they will see several patient systems based on patient disabilities. The user will then be able to select the desired system and choose the desired device to complete their task. Each device will take the user to the basic functionalities page where the user is able to choose their desired functionality and exit from there. We have also included additional functionalities that more knowledgeable users can have access to if needed.

**Installation/Operation/Maintenance:**

Initially, our client will be required to download node-red on their personal computer or Raspberry pi and type in 'npm install -g --unsafe-perm node-red' on the command prompt of their PC to open Node-Red. We will export the UI files as JSON files from Node-Red. The client will be sent two folders, one containing the User interface files and the other containing all images. The client will then be required to import the JSON files onto their node-red screen and select the 'Deploy' button to be taken to the User Interface website.

For full functionality, the client must download all required palettes as well as ensure all paths in the Node-Red programs are correct. Firstly, required palettes to run our fully functioning UI include:

node-red
node-red-contrib-image-simple-node
node-red-contrib-ui-media
node-red-dashboard
node-red-node-rbe
node-red-node-serialport
Node-red-node-tail

Each palette has a different purpose and each must be installed into the user's Node-Red or else the file will not function properly. Specifically, node-red-contrib-image-simple-node and node-red-contrib-ui-media are for the images and ensure that the images will show up in the dashboard. node-red-dashboard allows the user to access the final UI in the dashboard which pops up as another tab. node-red-node-serialport is used to allow Node-Red to have a serial connection with the Arduino board in the SensAct and allows for communication.

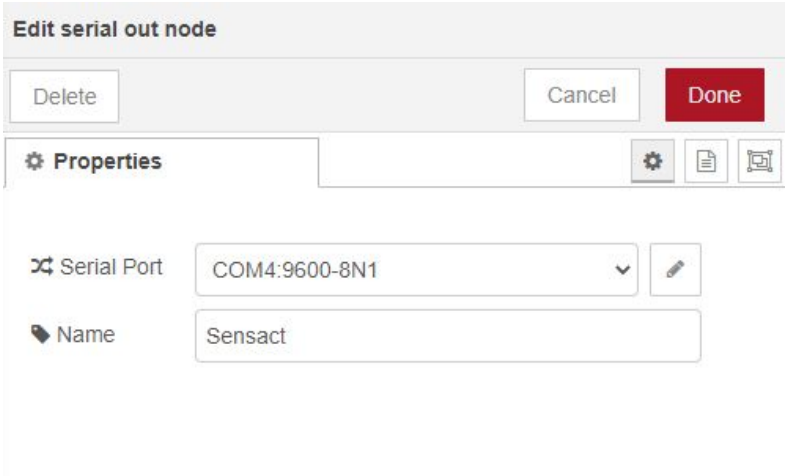


Figure 27: Serial Out Node Properties

It is important to ensure that the serial port matches the port in the computer, upon setup please ensure that the Sensact device is plugged connected to COM4, otherwise switch the port accordingly.

Once all palettes are properly installed, the UI will have all the basics it needs to function on Node-Red. However, if launched in this state the images will still be missing from the interface. All images used in the product are included in a separate file that the user will have to download and ensure that the file is linked correctly so that it shows up in the dashboard, as shown below.

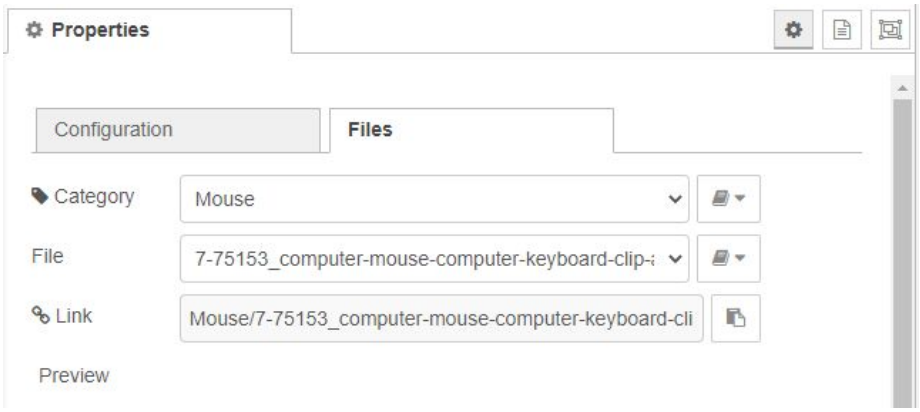


Figure 28: Image File Properties

## Potential Improvements:

We have included Basic, Intermediate and Advanced sections within our interface. Improvements may include implementing java functionality as well as improvements on intermediate and advanced sections. Functionality was implemented mainly into the basic menu in the final product, buttons in the basic menu contain specific payload messages and the user is able to use the basic menu to communicate with the SensAct device. Thus an improvement to our interface includes updating the intermediate and advanced menus to full functionality as well as possibly adding new options.

Another improvement includes implementing java functionality. The current interface only interacts with the Sensact through payload messages, and still requires the java program to allow the device to go into “idle” mode and edit new sensors and create new functions. Thus an improvement that could be added is including full java functionality so the user can edit sensor functions in the interface or put the Sensact into “idle” mode to edit it in the interface instead of having to use the java functionality.

There could also be potential improvements in the visuals and features of our User Interface to make it, even more, user-friendly and customized. For a focus on customization, we previously ideated several features such as voice commands from patients with mobility

disabilities and pre-programmed one-click buttons that complete common functions, which could also be added with more advanced programming skills.

## Troubleshooting:

We recommend troubleshooting our interface manually according to the error message on the screen when trying to deploy onto the dashboard website. Once the files are imported onto node-RED on a device, ensure the required nodes have been installed. Especially 'Dashboard' and 'Serial' nodes.

If the images do not show up when deployed onto the interface, they must be manually implemented by selecting 'Edit media node' → 'Node properties' → 'Files' and after selecting a category, browse to choose the image and deploy onto the dashboard for each individual image.

If the buttons have not been assigned a group or are unused node-red will indicate an error so it is important to make sure all the buttons have been assigned a group, it will also impact how it shows up on the interface.

In general, it is helpful to troubleshoot manually, by implementing a feature onto node-red and going to the dashboard to ensure it shows up the way we want.

[Function]

## Design Files:

Derivable A
https://drive.google.com/drive/u/1/folders/1sqMwRAp2APs1TVJIxP2vB1McsqkZ5pWm
Derivable B
https://drive.google.com/drive/u/1/folders/1LolChRJOYJzo7hYocj5CbcQlhapFbkPQ
Derivable C
https://drive.google.com/drive/u/1/folders/1-ciy24Tcvh-4MIBiuH1VOAoiYDKlbogL
Derivable D
https://drive.google.com/drive/u/1/folders/172roQqZN8RgI65iqyXWuaIWwVI97fcuH
Derivable E
https://drive.google.com/drive/u/1/folders/1mnlJ6GN_yRlaXW5-p-5PRDhQOOzj5Fbl
Derivable F
https://drive.google.com/drive/u/1/folders/1memLEoyaelcTNvX0qIfgbB0FROOQffKj
Derivable G
https://drive.google.com/drive/u/1/folders/1aww7d1R9oa5Wg9-ItB1JSKkE7iOLUEt7
Derivable H
https://drive.google.com/drive/u/1/folders/17Wb3ubeRy4_uk4_KqvZUPWbwz9iI3X0E
Derivable I
https://drive.google.com/drive/u/1/folders/1ahin4LBq2kACLAtk_pXlBJ6NvbJfwyUA
Derivable J
https://drive.google.com/drive/u/1/folders/1MHWomauxy8_W8afsSf3QRpio6icczQg4

Derivable K
https://drive.google.com/drive/u/1/folders/1wUq0QKWq2GvDTRclARr3sEq-OksocEze

## Conclusion and recommendations:

Thus our new user interface provides users with a simple and easy method to use the Sensact device without having to obtain the background knowledge in electronics and programming required for the confusing old interface. Our project is an improved solution for the problem we were presented with and our group overcame many obstacles to reach this point. The current interface allows the user to choose the desired basic patient system, sensor, and functionality which is done by clicking the intended buttons. Some examples of possible functionalities include a mouse, call bell, TV controller, and/or lightswitch.

There were many different skills that our team members developed throughout the whole course, and through this project, we obtained many new skills. The first of which is a background in using Node-RED to "code". Due to the current unforeseen circumstances of COVID-19, it was difficult to learn a new method of coding however through research on youtube and guidance from our TA we were able to understand the basics enough to work towards a functioning UI. We also gained knowledge in business economics and project management. Our client highly valued keeping this project low-cost, thus his request to use Node-RED which is free to use. From those requirements, we were able to keep the costs of our product low however, through an economics evaluation we were able to estimate the cost of creating a company using this product. We were also able to develop our collaboration, problem-solving, and communication skills through this project. There were few conflicts near the beginning of the project with interpreting the client's needs and translating them into a solution and design. We took the time to discuss and seek clarification and from that, we were able to create a better final product. We learned that understanding the project is just as important as finding the solution. Referring to communication skills with client meetings we discovered that to get more important information in a limited time, we should ask more questions and lead the conversation. This is because we could not speak with the client for prolonged times however because he was very passionate about his topic, he would often speak over the time we had to ask other questions.

To stay on track as a group, we updated our project plan frequently, as well as met and discussed our plans often as a group in our group chat. We split and assigned tasks to members and set deadlines ahead of time to ensure we were able to complete our project by design day. The design process was very important to us. For the interface, we decided to make 3 sections at the beginning which were basic functionality, intermediate functionality, and advanced functionality. We separated patient systems according to the requirements and limitations of their disabilities. We also added many visual representations and an audio setting to our interface. Our client was happy with the design, pictures and audio but he suggested we focus more on the functionality due to the limited time and lack of experience in coding. So we decided to focus on the basic functionality of this project.

# Bibliography


"ACCESSIBILITY LEGISLATION IN ONTARIO," *AccessibilityConsultants*. [Online]. Available: https://accessibilityconsultants.ca/resources/legislation. [Accessed: 03-Dec-2020].

"Accountant Salary in Canada," *PayScale*. [Online]. Available: https://www.payscale.com/research/CA/Job=Accountant/Salary. [Accessed: 03-Dec-2020].

"Brochure and Flyer Marketing on a Budget - Resources," *PsPrint*. [Online]. Available: https://www.psprint.com/resources/brochure-flyer-marketing-budget/. [Accessed: 03-Dec-2020].

"Dashboard Hospital Status UI Animation," *Dribbble*. [Online]. Available: https://dribbble.com/shots/12578692-Dashboard-Hospital-Status-UI-Animation. [Accessed: 03-Dec-2020].

"Electricity Prices in Canada (Updated 2020)," *energyhub.org*. [Online]. Available: https://www.energyhub.org/electricity-prices/. [Accessed: 03-Dec-2020].

"Graphic Designer Salary in Canada," *PayScale*. [Online]. Available: https://www.payscale.com/research/CA/Job=Graphic_Designer/Salary. [Accessed: 03-Dec-2020].

"Homestay app vol.5," *Dribbble*. [Online]. Available: https://dribbble.com/shots/6562417-Homestay-app-vol-5. [Accessed: 03-Dec-2020].

"How Much Does YouTube Advertising Cost?," *How Much Does YouTube Advertising Cost? | YouTube Ad Pricing*. [Online]. Available: https://www.webfx.com/internet-marketing/how-much-does-youtube-advertising-cost.html . [Accessed: 03-Dec-2020].

I. Advanced Solutions International, *Ontario Long Term Care Association*. [Online]. Available: https://www.oltca.com/OLTCA/. [Accessed: 03-Dec-2020].

Keith and RobT, "BUS-10103," *Series A Modular Desk 72"W by Bush | OfficeFurniture.com*, 05-Apr-2017. [Online]. Available: https://www.officefurniture.com/Modular-72W-Desk-BUS-10103.aspx. [Accessed: 03-Dec-2020].

"Marketing salary in Canada," *neuvoo*. [Online]. Available: https://neuvoo.ca/salary/?job=Marketing. [Accessed: 03-Dec-2020].

"New Commercial Office Space For Lease - REDUCED by $500 a month.: Commercial & Office Space for Rent: Ottawa," *Kijiji*. [Online]. Available: https://www.kijiji.ca/v-commercial-office-space/ottawa/new-commercial-office-space-for-lease-reduced-by-500-a-month/1497388811. [Accessed: 03-Dec-2020].

"Robohealth: Smart Room Interface for Hospital Patients." [Online]. Available: https://www.idiap.ch/workshop/iros2018/files/06f-brunete_workshop_final.pdf. [Accessed: 03-Dec-2020].

S. Morris, G. Fawcett, L. Brisebois, and J. Hughes, "Statistics Canada," *A demographic, employment and income profile of Canadians with disabilities aged 15 years and over, 2017*, 28-Nov-2018. [Online]. Available: https://www150.statcan.gc.ca/n1/pub/89-654-x/89-654-x2018002-eng.htm. [Accessed: 03-Dec-2020].

"Software Developer Salary in Canada," *PayScale*. [Online]. Available: https://www.payscale.com/research/CA/Job=Software_Developer/Salary. [Accessed: 03-Dec-2020].

"Staples Mesh Task Chair, Black," *Staples.ca: Fast Free Shipping, On All Orders*. [Online]. Available: https://www.staples.ca/products/394403-en-staples-mesh-task-chair-black. [Accessed: 03-Dec-2020].

"Stock Photo, Royalty-Free Image Prices and Plans," *Shutterstock*. [Online]. Available: https://www.shutterstock.com/pricing. [Accessed: 03-Dec-2020].

Tr, "Logitech Keyboard & Mouse Combo MK120," *Best Buy Canada*. [Online]. Available: https://www.bestbuy.ca/en-ca/product/logitech-keyboard-mouse-combo-mk120/10207205. [Accessed: 03-Dec-2020].

"UI Developer Salary in Canada," *PayScale*. [Online]. Available: https://www.payscale.com/research/CA/Job=UI_Developer/Salary. [Accessed: 03-Dec-2020].

"Welcome," *Welcome - Canada Computers & Electronics*. [Online]. Available: https://www.canadacomputers.com/product_info.php?cPath=1517_1518. [Accessed: 03-Dec-2020].

# Appendix

Deliverable A - We summarized our first client meeting.

Deliverable B - We made our customer needs, metrics, and we did benchmarking. Based on benchmarking and metrics, we found our competition product. After that, we made our target specifications

Deliverable C - We brainstormed for our design based on our client meeting summary and our target specification. Then we generated several ideas to one by decision matrix.

Deliverable D - We generate our client feedback and we find out our need statement, design criteria, associated metrics, and benchmarking was wrong so we did research again.

Deliverable E - We made a presentation for our prototype progress

Deliverable F - We decided our business model and we did the feasibility study

Deliverable G - We updated our second prototype

Deliverable H - We did our economic report for three years quarterly, and we found our break-even point is in the third quarter of the second year. And we did a pitch for our product.

Deliverable I - We made our three-minute presentation for design day.

Deliverable J - We made our user manual

Deliverable K - We made our final presentation

Extra - We realized we did something wrong for the previous deliverable after our second client meeting, so we made some changes to needs statements, design criteria, associated metrics, and metrics
https://docs.google.com/document/d/1YF-XjwCBejGqdx5ILEfZk5QfR5CiVb-QNKJvX_lO-I8/edit

Three minutes pitch
https://drive.google.com/file/d/1posi9f9QT1rJFcS-3IQ6orOioD5fPz61/view?usp=sharing