Boston University
Electrical & Computer Engineering
EC463 Senior Design Project

First Semester Report

MOSS Chair: Modular Open-Source Smart Wheelchair

Submitted to

Professor Osama Alshayk
Boston University
5 St. Mary's St.
Boston MA, 02215

by

Team 11
MOSS Wheelchair

Team Members

Sebastian Gangemi sgangemi@bu.edu
Aya Kassem ayak@bu.edu
Norbert Malik norbert@bu.edu
Maha Noor mahakdn@bu.edu

Submitted: 10 December 2023

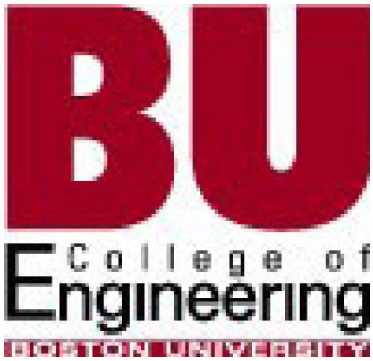


Table of Contents

Executive Summary (Maha and Aya) - Around 1% of the world's population needs wheelchairs, but a basic wheelchair cannot meet everyone's needs and expectations. Therefore, we are proposing to build a Modular Open Source Smart (MOSS) Wheelchair in which engineers will be able to create modules for a user's wheelchair as needed. For our project, we will implement three modules while also creating an open-source website for easy access to our modules. The first module introduces a 3D-printed robotic arm implemented with servo motors for precision in movement and manipulation. The second module features navigation, where we employ the YOLOv8 algorithm for computer vision. This algorithm facilitates advanced real-time object detection and recognition, allowing the wheelchair to navigate its environment intelligently. The third module tackles obstacle detection and avoidance, utilizing the outputs of the YOLOv8 algorithm to intricately control the motors and electronics responsible for wheel movement and robotic arm functionality. Using computer vision will allow our wheelchair system to gain an enhanced ability to interpret its surroundings, offering users a more sophisticated and adaptive mobility solution. Through this comprehensive approach, we aim to redefine wheelchair capabilities and foster inclusivity while providing users with customizable, state-of-the-art mobility solutions.

I. Introduction (Aya)

Customer's Problem:
A limited fraction of the global population requires smart wheelchairs, each with unique functional needs. Current products are often tailored to specific disabilities, making it challenging for companies to efficiently address the diverse demands within this niche market.

Purpose:
The purpose of our project is to create an open-source platform for smart wheelchairs. This platform will allow developers to implement modular designs, offering a continuously expanding range of features to cater to the varying needs of wheelchair users.

General Approach:
Our team's approach involves creating a modular framework where developers can implement and customize features based on individual user requirements. By providing a foundation of essential features, we encourage a community of developers to utilize our project, fostering innovation and adaptability.

Solution to User's Problem/Needs:
Our approach solves the user's problem by offering an open-source platform with modular designs. The wheelchair will initially feature assistive navigational functions, motorization for autonomous capabilities, computer vision algorithms like YOLOv8 for reliable indoor navigation, and compatibility with a programmable assistive robotic arm. This modular design allows developers to tailor features according to specific user needs, providing a customizable and adaptive solution.

Highlights of the Modular Open-Source Smart Wheelchair:

- **Modular Design:** Enables developers to build and customize features within the framework.
- **Assistive Navigational Functions:** Motorization for autonomous capabilities and reliable indoor navigation using YOLOv8.

- **Programmable Robotic Arm:** Compatibility with an assistive robotic arm, offering a wide range of functionalities.
- **Community Collaboration:** Encourages a community of developers to contribute and innovate, expanding the range of available features over time.

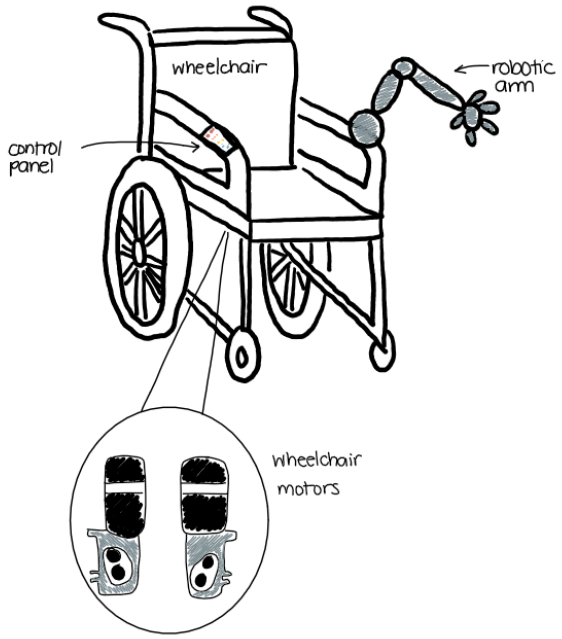


Figure 1: This drawing shows the overall anatomy of the smart wheelchair. It includes two motors for each wheelchair, which will be connected to an engine. This engine will be controlled by the control panel which will be where the user enters their commands. The control panel will also control the robotic arm, shown on the right armrest of the wheelchair. This robotic arm will be responsible for basic motor commands such as pressing elevator buttons, opening doors, etc.

II. CONCEPT DEVELOPMENT (AYA)

Addressing the Customer's Problem:

The customer confronts a pressing challenge rooted in the constrained adaptability of contemporary smart wheelchairs, which are predominantly tailored for specific disabilities. This limitation curtails the flexibility required to cater to users with diverse mobility needs effectively. Our engineering analysis discerns the need for a solution that not only rectifies the immediate user requirements but also nurtures a culture of community-driven innovation. At the core of this engineering understanding lies the crucial requirement for open-source solutions, empowering users and developers to collaboratively shape the capabilities of the wheelchair.

Engineering Analysis:

Examining the engineering aspects of the customer's problem reveals a bottleneck in the adaptability of contemporary smart wheelchairs. These devices, often designed with a one-size-fits-all mindset, fall short in addressing the varied needs of users with diverse mobility impairments. The inflexible design of current wheelchairs hampers adaptability, creating a significant gap in meeting the unique requirements of a broader user base. Users demand a solution that not only offers essential features but also caters to the dynamic and evolving nature of mobility challenges.

The engineering challenge, therefore, is to develop a smart wheelchair that not only serves as a mobility aid but also evolves to meet the diverse and evolving needs of users.

Conceptual Approach:

Our strategic plan entails creating an open-source, modular framework for smart wheelchairs. This approach prioritizes modularity, intending to offer not only a foundational set of features, such as assistive navigation and robotic arm compatibility but also to create an environment where users and developers can actively contribute to and customize functionalities. This conceptual approach is rooted in the belief that openness is not just a feature for adaptability but a fundamental principle for encouraging the continuous integration of diverse assistive technologies. The goal is to create a smart wheelchair that serves as a dynamic platform, capable of addressing immediate needs while remaining adaptable to the future and ever-changing requirements of a broad and diverse user base.

Why the Chosen Concept:

The decision to prioritize modularity and open-source principles is crucial in effectively addressing the various aspects of the customer's problem. Modularity ensures flexibility, allowing the wheelchair to adapt to varying user needs, while open-source principles encourage collaboration and innovation within the community. The chosen conceptual approach stems from the recognition that a rigid, closed-system approach would fall short in addressing the dynamic and diverse landscape of mobility challenges. By adopting a modular, open-source framework, we lay the groundwork for a solution that not only meets present needs but also invites ongoing collaboration, adaptation, and innovation to cater to emerging user requirements.

Alternative Solutions Considered:

In exploring potential solutions, several alternatives were considered. One option involved constructing a smart wheelchair entirely from scratch, offering a customized design. However, this alternative was dismissed due to the challenges associated with extended development timelines, increased costs, and potential difficulties in engaging the community effectively. Another alternative considered an integrated specialized design tailored for specific disabilities. Still, this avenue was abandoned as it lacked the adaptability required for a broad user base and failed to account for the evolving nature of mobility needs.

The chosen conceptual approach of a modular, open-source framework emerged as the most practical and sustainable solution, striking a delicate balance between customization and a robust foundational design.

In conclusion, our detailed engineering analysis sheds light on the crucial need for flexibility and adaptability in smart wheelchairs. The chosen conceptual approach of a modular, open-source framework is grounded in a nuanced understanding of user needs and the dynamic nature of mobility challenges. This chosen concept not only addresses immediate user needs but also sets the stage for a collaborative community actively contributing to the ongoing evolution of the

wheelchair's capabilities. Striking a harmonious balance between user customization and a robust foundational design, our solution ensures a dynamic and continually aligned response to the diverse and evolving needs of users with a wide spectrum of mobility requirements.

III. System Description

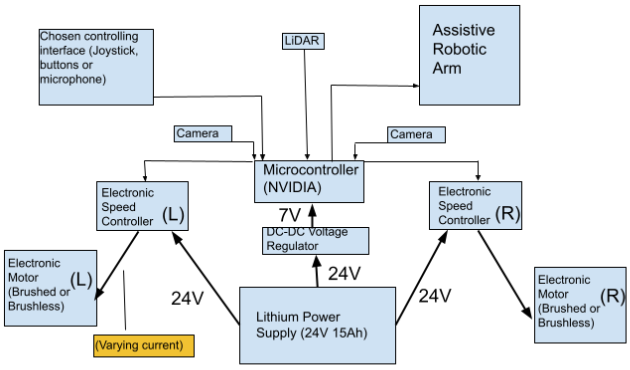


Figure 2: Block Diagram of the Autonomous Wheelchair

Control System (Maha):

Our system will be centrally controlled by the Nvidia Jetson Nano, a small computer that allows us to run multiple neural networks in parallel, especially useful for the obstacle detection and computer vision aspects of our project. The Nvidia Jetson Nano will receive inputs from the camera and LiDAR sensor to process the outside world for the wheelchair user and move accordingly. Additionally, the Nvidia Jetson Nano will be capable of taking in user inputs through either a joystick, buttons, a microphone, or a screen. The microcontroller will then send output signals to the motor-controlling system which correspond to the desired speeds of the wheel. The microcontroller will also output commands to the robotic arm, which will be controlled by several servo motors, which will move to execute the desired command. This whole system will be battery-powered and regulated by a DCDC voltage regulator.

Motors (Sebastian):
For the motor controlling system of our design, we will have 2 24V 250W brushed DC motors each controlled by an electronic speed controller. The electronic speed controllers will be receiving power from the 24V 15Ah lithium battery, and the Jetson Nano will receive PWM (0-5V) signals corresponding to the desired speed and direction for each motor. The electronic speed controller will then regulate the voltage from the power supply accordingly so that the rpm of the motor changes proportionally to the PWM speed input. We will set the scale of the speed controller by having a 5V PWM input correspond to the max desired speed of the wheelchair. Finally, each motor will be fixed onto its respective wheel using a gearbox to transfer the high rpm low torque capabilities of the motors to the low rpm high torque demands of moving a wheelchair.

Robotic Arm (Norbert):

The robotic arm is a 3-joint system that attaches to the side of the wheelchair. Fully 3D printed, the arm will be made up of several smaller pieces to end in 3 main parts: bicep, forearm, and end effector. Motors to drive the movement of the arm will be found inside it or in a base casing at the proximal end of the arm. The motors will drive strings that run through the arm to make the end effector grip things and bend the arm. In order to overcome torque requirements, we may use gearboxes or drive screws as part of the system to obtain a mechanical advantage.

Software (Aya):

The software powering our smart wheelchair project encompasses a robust system centered on object and obstacle detection. Leveraging the YOLOv8 algorithm, our software enables real-time identification and classification of objects in the wheelchair's environment. This computer vision system processes incoming frames swiftly, ensuring timely responses to potential obstacles. The integration of an intuitive user interface will allow the user to control the wheelchair through a touch screen or voice control, facilitating a user-friendly experience.

IV. First Semester Progress

Overall (Maha and Sebastian):
This semester, our team defined our project and started prototype testing. To define our project, we devised the 3 modules and solidified our idea of using an open-source platform. We then started to research how to create a prototype for the three modules. For the robotic arm, we were referred to a website called inmoov.fr, where we were able to 3D print an arm and use 6 servo motors to control it. This arm can be used as a base for our final robotic arm that we wish to implement on our wheelchair. We also made a prototype for the wheelchair motors themselves which will be used to move the wheelchair. We used an Arduino, a speed controller, and a brushless motor to show forward and reverse motion along with smooth acceleration and deceleration.

Motor (Sebastian):
Initial results for the forward direction motion are shown in the figures below:

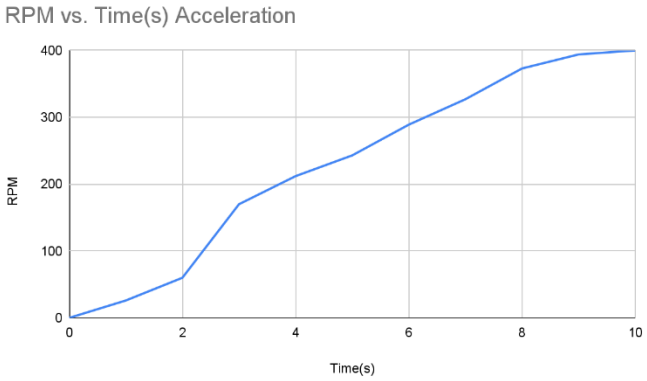


Figure 3a: This graph shows the motor RPM in the forward direction. From the line, we can see a relatively smooth acceleration.

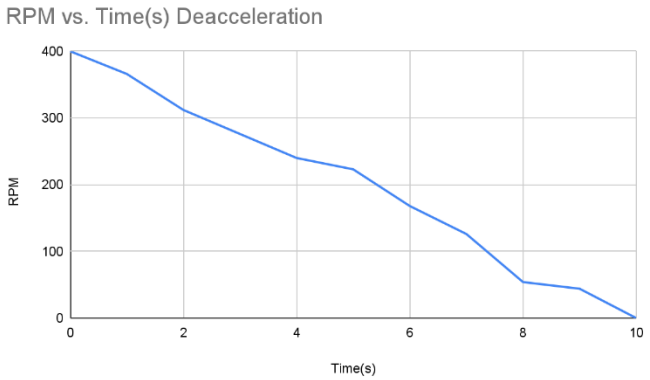


Figure 3b: This graph shows the motor RPM in the forward direction. From the line, we can see a relatively smooth deceleration.

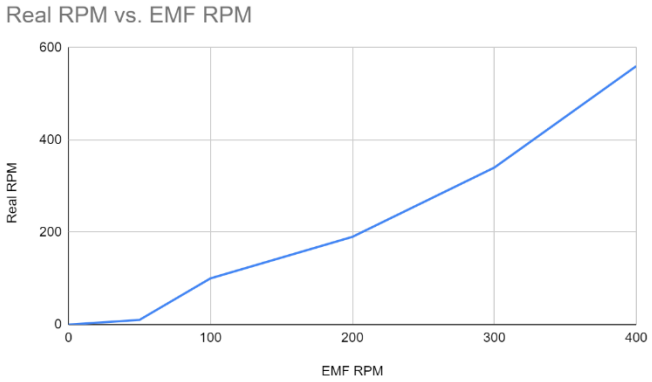


Figure 4: This graph shows the RPM that the speed controller is trying to achieve vs the actual RPM of the motor. Note that in this case the speed controller is calibrated to be accurate at lower RPMs(Up to 300) due to the low power used for the prototype, while in the final product the range in which the EMF RPM ≈ Real RPM may include RPMs up to 2000 RPM

Object Detection (Aya):

In terms of the computer vision aspect of our project, we also managed to successfully detect objects using computer vision object detection algorithms, more precisely Ultralytics' YOLOv8 algorithm. We first began by simply importing a pre-trained YOLOv8 model. This model did pretty well at recognizing common objects from everyday life. However, we wanted to adapt our wheelchair to detect not only common objects but also specific objects that might cause accessibility problems to people with severe disabilities. For example, we want the MOSS Chair's robotic arm to be able to complete everyday tasks such as opening doors. This requires it to detect a door handle, which is not an object present in the original COCO dataset that the YOLOv8 model is pre-trained on. Similarly, we want the robotic arm to be able to press elevator buttons, which requires the algorithm to detect elevator buttons specifically (simply detecting an elevator does not suffice).

In order to overcome this obstacle, we found some Roboflow datasets specific to these objects and combined them into one dataset that also contains instances of the COCO dataset. We proceeded to import the pre-trained model but then apply transfer learning to let it learn new data and therefore detect the new specific objects we want it to detect. The reason behind including instances of the COCO dataset in the new dataset is that despite us importing a pre-trained model, this model forgets the data it was originally trained on when trained on all new data, which means that the model would perform well on the new specific objects we teach the model to recognize, but it would perform poorly on the objects of the COCO dataset it was originally trained to detect.

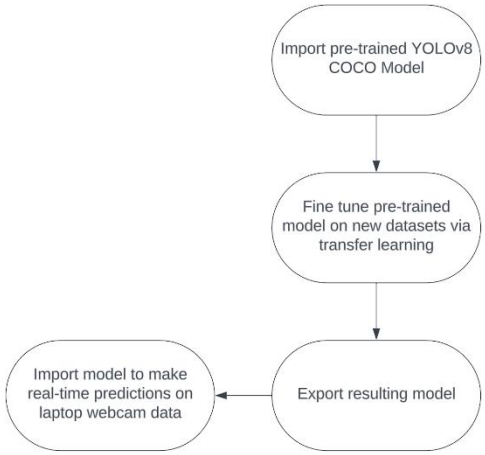


Figure 5: Workflow diagram showing the steps taken to prepare and export the project's YOLOv8 object detection model

The fine-tuned model was trained on our custom dataset (which is a combination of existing Roboflow datasets that contain objects we want our wheelchair cameras to detect) for 100 epochs. The following results were generated:

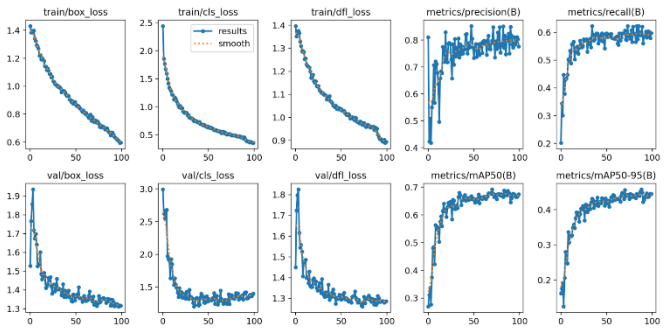


Figure 6: Training loss, validation loss, and precision results after fine-tuning the Ultralytics' YOLOv8 model for 100 epochs on a custom dataset

According to the results, the training loss seems like it could potentially go lower, so a future step is to train this model for more epochs (or we can train the model and check training loss values after each epoch and only stop when they don't seem to change for multiple epochs in a row - in other words, train until convergence).

Here are a couple of examples of the resulting model's inference using a laptop's webcam:

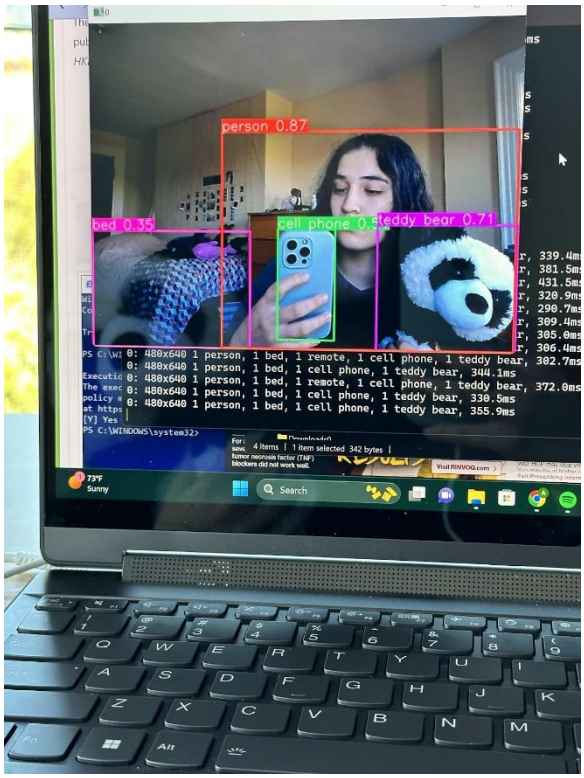


Figure 7: Results of the pre-trained YOLOv8 model. This model is pre-trained on the COCO dataset and is trained to recognize common objects such as the ones in the picture.

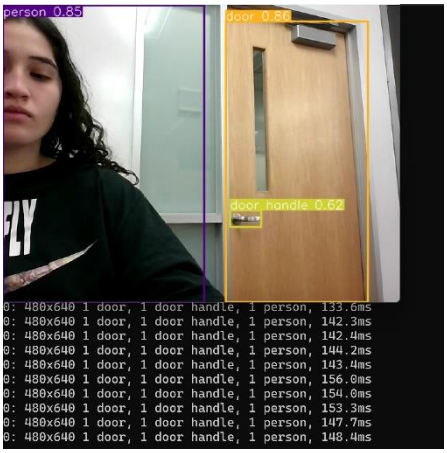


Figure 8: The resulting fine-tuned YOLOv8 model. Unlike the pre-trained model, the fine-tuned model can recognize instances from the COCO dataset such as people, but it can also recognize more specific objects from the custom dataset such as doors and door handles.

The resulting model was able to detect objects in real-time (<500 ms), which aligned with our goal of detecting objects and obstacles within 1 second.

Robotic Arm (Norbert):
The robotic arm prototype was sourced from INmoov, which is an open-source arm that can be downloaded and printed for free online. It has realistic dimensions and the scale of a normal human arm. Strings running through the fingers and hand result in a means of articulating the joints of the arm and hand. Printing the parts was done over the course of a week on 5 separate printers, but required constant presence in the lab attending the printers, restarting prints when they failed, and continuously switching out print jobs. There are approximately 45 part groups, with close to 80 individual pieces in the print, as depicted in Figure 9. Assembling this arm proved very difficult, especially in delicate pieces, such as the fingers. The friction between the pieces led to very rigid motion that made the joints difficult to fully close and open using the strings. Smoothing out one finger in post-processing took almost 2 hours and still resulted in jittery movement, but the finger was fully articulated.

Figure 9: All of the printed parts needed to assemble the arm.

Figure 10: The articulated hand, attached finger with strings for articulation visible.

Figure 11: The assembled forearm and hand.

The arm is an art piece as much as anything practical, and as such it will serve as a baseline for future models. We will need to reinforce the design and simplify it so it is easier and faster to replicate. Currently, the design also takes close to 100 hours to finish printing each piece. This is also on low-quality print settings, with minimal infill. High infill for structural integrity and high-quality print settings so pieces fit well together and smoothly would require several hundred hours of printing or post-processing. Hence I hope to simplify the design and focus solely on practicality and not design appeal.

V. Technical plan

Control System (Maha):

Task 1: Flash and Start Nvidia Jetson Nano
The Nvidia Jetson Nano requires to be flashed before it can be used. This is to ensure that the SD cards that we use in the future can be properly read and written to by the Jetson Nano. Additionally, we will use this opportunity to set up the Jetson Nano to fit our application with the wheelchair. Lead: Maha

Task 2: Integrate Jetson Nano with Object Detection and Camera Sensors
The Object Detection code will be integrated into the Jetson Nano. Along with the code, the camera sensor will also be hardwired into the board. We will connect the code to interface with the camera and test that the object detection is working as expected to ensure obstacle avoidance with the wheelchair. Lead Maha; Assisting: Aya

Task 3: Integrate Jetson Nano with Motors
The motors that are used for moving the wheelchair and robotic arm will have to be hardwired with the Jetson Nano and the code for controlling them from the user interface. We will write code to inform the Jetson Nano how to move the motors according to different user inputs and then wire the motors to the Jetson Nano to act accordingly. Lead: Maha; Assisting: Sebastian, Norbert

Task 4: Create a harness for any wires connected to the Jetson Nano
To make the wheelchair look neat, along with decreasing the chance of any wires being snagged or pulled out, a cable harness will be assembled. This harness will also ensure that the wheelchair is suitable for outside use. The harness will be made of a woven material that encloses the wires and keeps them safe from substances such as water and dirt. Lead: Maha

Task 5: Battery Power Supply
A 24 V, 15 Ah battery will be acquired and tested on a dummy resistance that will be roughly equivalent to our system. It should be rechargeable from an external connector and meet specifications for weight, battery life, and heat dissipation to ensure user safety. Lead: Maha, Assisting: Aya, Sebastian

Motors and Sensors (Sebastian):

Task 1: Tune new motors
As we are using stronger motors than what we prototyped with, capable of higher RPM and more wattage, we will have to reprogram the speed controller to handle these higher RPMs corresponding to higher amperage. This will include changing the scale PWM input V/ output RPM, as well as doing testing to calculate the back EMF constant(V/RPM) to calibrate the controller's feedback, ensuring the speed increase will be proportional to the input signal as desired. Lead: Sebastian

Task 2: Mount the system onto the wheelchair
As we now have all of our components delivered, we will have to design a mounting solution for every component we wish to incorporate. The main components(i.e. not sensing) will be kept beneath the wheelchair, using plywood or a 3D-printed surface to hold the battery, microcontroller, speed controllers, and motors. The sensors will each have their solution for mounting onto the chair according to their functionality and dimensions. For example, the Joystick will rest on the armrest of the chair while LiDAR will be mounted on the front side of the chair. Lead: Sebastian Assisting: Norbert, Maha

Task 3: Fix gearboxes onto wheels
The motors are to be attached to the wheels via a gearbox with a chain and sprocket system. We will design a solution so that the motorization can be modularly attached and detached. Due to the precise positioning needed so that the chain from the gearbox sprocket to the axle sprocket stays flat, we will also need to create a mounting solution that ensures the gearbox and motor always get mounted in the same position relative to the wheel. Lead: Sebastian Assisting: Norbert

Robotic Arm (Norbert):

Task 1: Redesign arm
Since the weaknesses of the arm prototype are now understood and that I have obtained a familiarity with the arm and what would be hard to do and what is needed, I will be able to redesign the arm in a way to maximize its efficacy while simplifying the design to limit print time and post processing. Lead: Norbert

Task 2: Redesign end effector

I will give special attention to designing an end effector that works well and performs above what we hope as a part of the arm that will not need to be modified in future iterations of this project. Lead: Norbert

Software (Aya):

Task 1: Finalize Custom Dataset
The current custom dataset used for the object detection portion of the project contained sample data mostly developed for testing purposes. It contains data for common objects, doors, door handles, and elevator buttons. The dataset will be expanded to contain additional objects the wheelchair should detect. Lead: Aya

Task 2: Fine-tune YOLOv8 Object Detection Model on Final Dataset
Since we already have a working method for fine-tuning the YOLOv8 object detection model, the same method will be used to train the model on the finalized dataset mentioned above in task 1. The fine-tuning parameters, however, will depend on the structure of our finalized dataset, so part of this task includes finding the optimal fine-tuning parameters that will lead to optimal accuracies. Lead: Aya

Task 3. Test Object Detection on Nvidia Jetson Nano and Raspberry Pi V2 Camera
The fine-tuned model that will be trained on the dataset mentioned above in task 1 will be tested on the Nvidia Jetson Nano. The Nvidia Jetson Nano is a powerful board computationally capable of running YOLOv8, but the main goal here is to get similar detection times as the results we got with a laptop, aka <500 ms detection time to ensure objects are detected in real-time. Lead: Aya; Assisting: Maha

Task 4: Integrate Object Detection with Motors and Sensors:
After getting our object detection model running on the Nvidia Jetson Nano, the final step will be to have it integrated with the rest of the project. This consists of having the results of the object detection model be used to control the navigation and robotic arm motors of the wheelchair. The output of the object detection script and the wheelchair's sensors will be used to generate instructions that will be passed as input to the motors. Lead: Aya; Assisting: Maha, Sebastian, Norbert

VI. BUDGET ESTIMATE

Item	**Description**	Cost($)
1	Nvidia Jetson Nano	149
2	Cameras(2X)	49
3	Lidar	99
4	Microphone	15
5*	Brushed Motors with Gearboxes (x2)	
5*	Joystick	259
6	Lithium Battery	170
7**	Electronic Speed Controllers (2x)	200

Grand Total: $741**

*Item 5 came in a bundle
**Item 7 was donated, including 7 total comes to $941

Our main costs for the project are the Nvidia Jetson Nano ($149), the motors ($259) and the lithium ion battery ($170). This makes sense as these components are the largest and most integral to our project. The Jetson Nano is quite expensive due to its advanced computing abilities. The motors are quite expensive because they have a large torque and can spin under large weight constraints and the battery is expensive due to its substantial voltage.

To stay within our budget, we have been using many parts that we already have such as Arduinos, old motors for prototype testing, wires, and other electrical components from past projects. We have been sending links to the components we need purchased to Professor Osama who has been buying these items for us. Other small costs, such as SD cards, 3D printing filament, and cable adapters have been bought by us out of pocket.

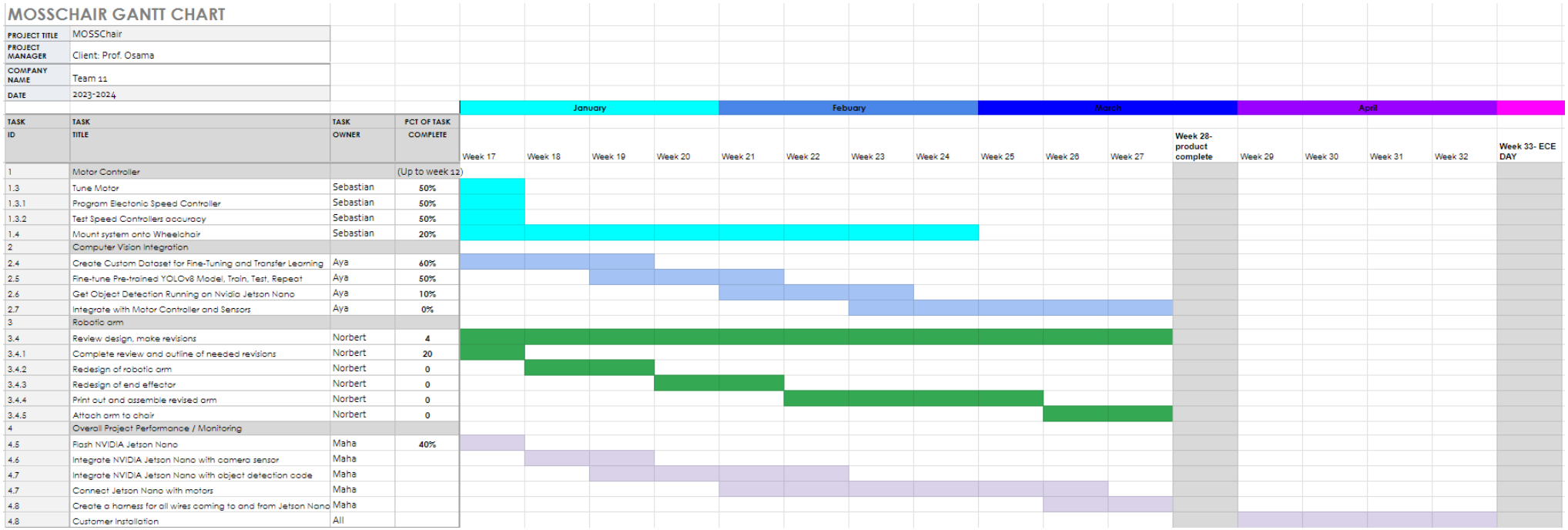
VII. ATTACHMENTS

A. Appendix 1- Engineering Requirements

Team #11
Project Name: MOSS Wheelchair

Requirement	**Value, range, tolerance, units**
High Object Detection Accuracy	>= 90% Accuracy
Real-Time Processing	<= 500 ms processing time
Night Vision Object Detection	No change in detection accuracy under different brightness conditions
Max Speed	4.5-8 mph
Battery Life	≈8 Hours, must be adjustable(modular battery)
Arm Weight	12 lbs
Weight that arm can lift	5 lbs
Arm reach length	3 ft

B. Gantt Chart



MOSSCHAIR GANTT CHART

PROJECT TITLE	MOSSChair
PROJECT MANAGER	Client: Prof. Osama
COMPANY NAME	Team 11
DATE	2023-2024

TASK ID	TASK TITLE	TASK OWNER	PCT OF TASK COMPLETE
1	Motor Controller		(Up to week 12)
1.3	Tune Motor	Sebastian	50%
1.3.1	Program Electonic Speed Controller	Sebastian	50%
1.3.2	Test Speed Controllers accuracy	Sebastian	50%
1.4	Mount system onto Wheelchair	Sebastian	20%
2	Computer Vision Integration		
2.4	Create Custom Dataset for Fine-Tuning and Transfer Learning	Aya	60%
2.5	Fine-tune Pre-trained YOLOv8 Model, Train, Test, Repeat	Aya	50%
2.6	Get Object Detection Running on Nvidia Jetson Nano	Aya	10%
2.7	Integrate with Motor Controller and Sensors	Aya	0%
3	Robotic arm		
3.4	Review design, make revisions	Norbert	4
3.4.1	Complete review and outline of needed revisions	Norbert	20
3.4.2	Redesign of robotic arm	Norbert	0
3.4.3	Redesign of end effector	Norbert	0
3.4.4	Print out and assemble revised arm	Norbert	0
3.4.5	Attach arm to chair	Norbert	0
4	Overall Project Performance / Monitoring		
4.5	Flash NVIDIA Jetson Nano	Maha	40%
4.6	Integrate NVIDIA Jetson Nano with camera sensor	Maha	
4.7	Integrate NVIDIA Jetson Nano with object detection code	Maha	
4.7	Connect Jetson Nano with motors	Maha	
4.8	Create a harness for all wires coming to and from Jetson Nano	Maha	
4.8	Customer Installation	All	