NORTHERN ARIZONA UNIVERSITY | School of Informatics, Computing, and Cyber Systems

To: Lauren May, Taylor Yee, Khaled Khaled

From: Daniel Beckett

CC: N/A

Date: 12/4/19

Re: Individual Sub-system planning

This document contains proposed models of our Work Breakdown Structure (WBS) and Critical Cain Management (CCM). The focus of both models being the mobility sub-system of the Augmented Power Mobility device.

Augmented Mobility Platform Education (AmpEd)

Daniel Beckett, Khaled Khaled, Lauren May, Taylor Yee

Project:

Augmented Power Mobility

Emails of Team-members:

Kaa325@nau.edu – Khaled Khaled

Dbb72@nau.edu – Daniel Beckett

Lm933@nau.edu – Lauren May

Tey24@nau.edu – Taylor Yee

Client & GTA emails:

Kyle.Winfree@nau.edu – Dr. Winfree

si229@nau.edu – GTA Shafkat

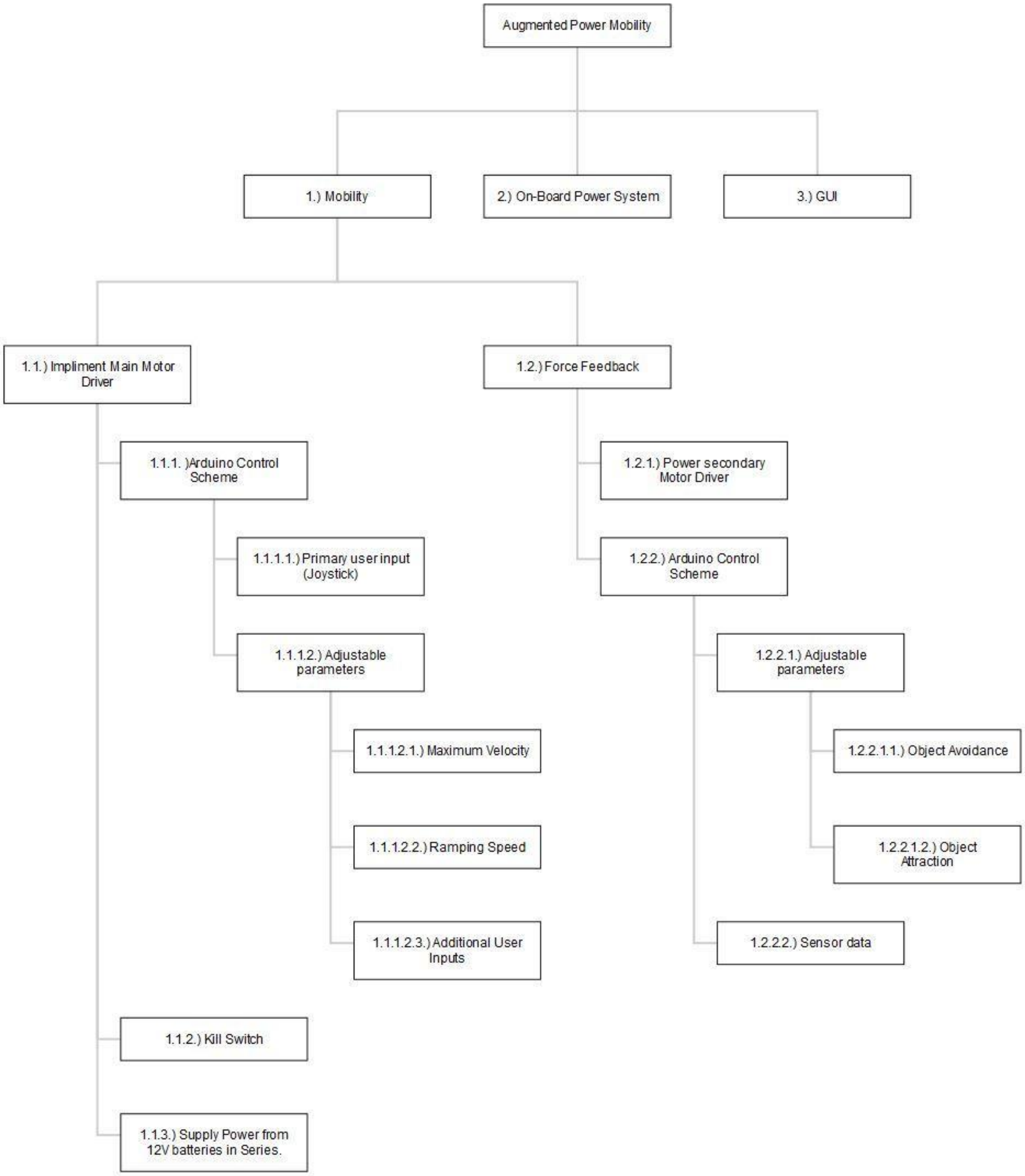
Faculty Advisors:

Client: Dr. Kyle Winfree

Sponsor: Dr. Kyle Winfree

Faculty Mentor: Shafkat Islam

Work Breakdown Structure:



1.1.) **Implement Main Motor Driver**

Have the Sabertooth 2x32 Motor driver incorporated into the system. It is responsible for powering and operating both primary motors as well as the Arduino. It receives serial packet commands from a physical link to the Arduino. The proper motor driver dip switch configuration is: 1=OFF, 2=OFF, 3=OFF, 4=ON, 5=ON, 6=ON. Using Library functions for the Sabertooth 2x32, the Arduino has several commands that send 8-bit serial packets to change how the motors operate. This serial exchange has a BAUD rate of 9600 and the input on the motor driver is labeled “S1.Using analog input values from the joystick into the Arduino, write code to specify which command to run according to joystick positioning. Implement options to adjust motor driver parameters, this is to help our client shape the behavior of our device to best suit each individual kid. Reduce the maximum PWM value when mapping the joystick’s analog input values. Lowering the duty cycle will limit maximum achievable velocity. Implement a feature that systematically breaks down changes to PWM and uses delays to reduce the rate at which top speed can be achieved. Alternative user inputs should be implemented in order to broaden our audience by making our device more accessible. No such method has been finalized, possible ideas are EEG devices, gyroscopes, and piezoelectric strips. A kill switch controls whether the Sabertooth 2x32 receives power from the two 12V batteries. It does this by using a relay to cut off the positive terminal of the 24V series battery when the kill switch is pressed.

1.1.1.) **Arduino Control Scheme**

Using Library functions for the Sabertooth 2x32, the Arduino has several commands that send 8-bit serial packets to change how the motors operate. This serial exchange has a BAUD rate of 9600 and the input on the motor driver is labeled “S1”

1.1.1.1.) **Primary User Input (Joystick)**

Using analog input values from the joystick into the Arduino, write code to specify which command to run according to joystick positioning.

1.1.1.2.) **Adjustable Parameters**

Implement options to adjust motor driver parameters, this is to help our client shape the behavior of our device to best suit each individual kid.

1.1.1.2.1.) **Maximum Velocity**

Reduce the maximum PWM value when mapping the joystick’s analog input values. Lowering the duty cycle will limit maximum achievable velocity.

1.1.1.2.2.) **Ramping Speed**

Implement a feature that systematically breaks down changes to PWM and uses delays to reduce the rate at which top speed can be achieved.

1.1.1.2.3.) **Additional User Inputs**

Alternative user inputs should be implemented in order to broaden our audience by making our device more accessible. No such method has been finalized, possible ideas are EEG devices, gyroscopes, and piezoelectric strips.

1.1.2.) **Kill Switch**

A kill switch controls whether the Sabertooth 2x32 receives power from the two 12V batteries. It does this by using a relay to cut off the positive terminal of the 24V series battery when the kill switch is pressed.

1.1.3.) **Power**

Have both 12V car batteries in series to supply the necessary 24V needed to power the motors.

1.2.) **Force Feedback**

While force feedback is limited to those who can grasp a joystick, it should still be implemented because of its potential to improve teaching through either added guidance or challenge. Force feedback is achieved by two DC motors that can adjust the X and Y axis position of the joystick. In order to control both of these motors, a smaller secondary motor driver is needed. This secondary motor driver will be located within the base of the controller. Code will need to be written to ensure the force feedback motors act in accordance to the distances of detected objects around the device. Exactly how the force feedback motors respond should be able to be tweaked by our client to ensure the device teaches most effectively. Changing the motors to pull the joystick towards any detected objects within a certain range. This will challenge the user to either react to an unexpected stimulus or increase spatial awareness. The Arduino will be controlling the force feedback motor driver based on analog readings from distance-based sensors. Our team is still debating on which exact method to use for detecting surrounding objects.

1.2.1.) **Power Secondary Motor Driver**

Force feedback is achieved by two DC motors that can adjust the X and Y axis position of the joystick. In order to control both of these motors, a smaller secondary motor driver is needed. This secondary motor driver will be located within the base of the controller.

1.2.2.) **Arduino Control Scheme**

Code will need to be written to ensure the force feedback motors act in accordance to the distances of detected objects around the device.

1.2.2.1.) **Adjustable Parameters**

Exactly how the force feedback motors respond should be able to be tweaked by our client to ensure the device teaches most effectively.

1.2.2.1.1.) **Object Avoidance**

For less practiced drivers, force feedback can help pull their hands away from incoming collisions. The hope is that this will reinforce the drivers procedural learning.

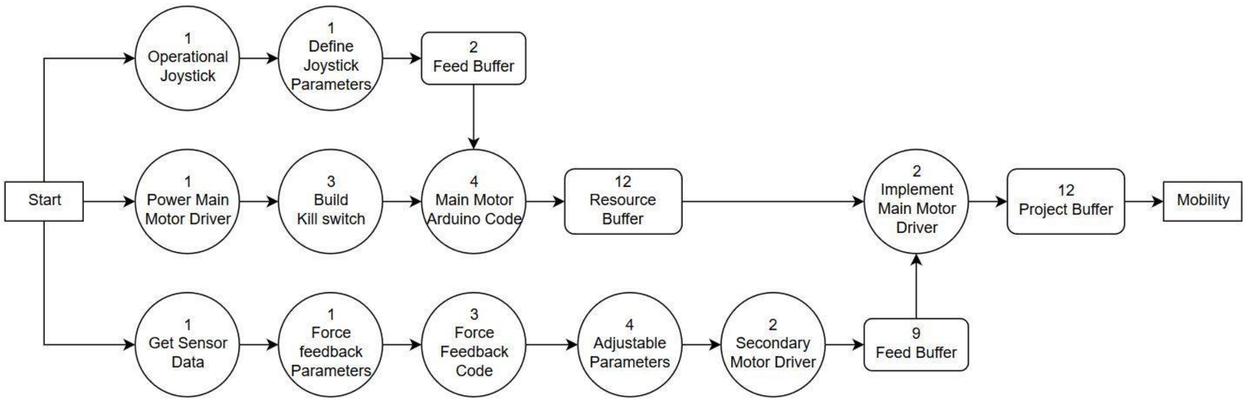
1.2.2.1.2.) **Object Attraction**

Changing the motors to pull the joystick towards any detected objects within a certain range. This will challenge the user to either react to an unexpected stimulus or increase spatial awareness.

1.2.2.2.) **Sensor Data**

The Arduino will be controlling the force feedback motor driver based on analog readings from distance-based sensors. Our team is still debating on which exact method to use for detecting surrounding objects.

Critical Chain Management:



TO : **Professor Kyle Winfree**

FROM : **Khaled Khaled**

DATE : **2nd December, 2019**

SUBJECT : **Baby GUI**

The graphical user interface is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicator such as primary notation, instead of text-based user interfaces, typed command labels or text navigation. In developing this, the following are required: Input Controls which includes checkboxes, radio buttons, dropdown lists, list boxes, buttons, toggles, text fields, date field; Navigational Components which includes breadcrumb, slider, search field, pagination, slider, tags, icons; Informational Components which includes tooltips, icons, progress bar, notifications, message boxes, modal window and lastly Containers that include accordion.

Input Controls are an integral part to any form. They allow a user to perform a variety of functions, e.g. type in text, select items from a list, upload a file. Among the challenges that may arise from this is programming errors but when fixed, everything will work perfectly for the function of the prototype.

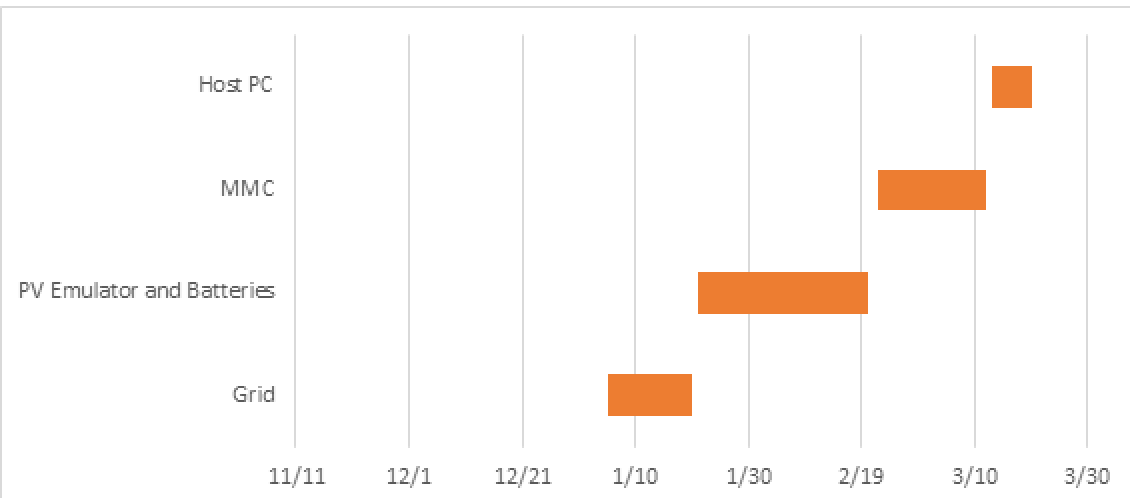
The navigational components enable moving from one tab to another or from one function to another. This allows the user to navigate perfectly. Some of the challenges includes putting the table in the wrong position or responding to wrong function.

Lastly, Informational Components transmits information are required by the user. This allows to responding of commands.

<table>
<tr><th>Level 1</th><th></th><th>Level 2</th><th>Level 3</th></tr>
<tr><td rowspan="2">1.0 Augmented Powered Mobility</td><td></td><td>1.1 Baby GUI</td><td rowspan="2">1.1.1 Input controls
Checkboxes, radio buttons, dropdown lists, list boxes, buttons, toggles, text fields, date field
1.1.2 Navigational Components
Dread crumb, slider, search field,</td></tr>
<tr><td></td><td>1.2 Motor with Joystick</td></tr>
</table>

		1.3 Data Processing and Transfer from a single sensor	pagination, slider, tags, icons **1.1.3 Informational Components** Tooltips, icons, progress bar, notifications, message boxes, modal windows **1.1.4 Containers** Accordion

Gantt Chart



NORTHERN ARIZONA UNIVERSITY | School of Informatics, Computing, and Cyber Systems

To : Dr. Kyle Winfree
From : Lauren May
Date : 4th December 2019
RE : Individual Feasibility and Planning

Dear Dr. Winfree,

Attached below is a work breakdown structure (WBS) as well as a Gantt Chart detailing the feasibility and how I plan to lead the implementation of the driving controls. The WBS describes various deliverables that are integral to the completion of several driving controls that would be able to be used by those with disabilities. The Gantt Chart shows a visual representation of a realistic timeline to complete those deliverables.

Thank you,

Lauren May

Augmented Mobility Platform Education (AmpEd)
Daniel Beckett, Khaled Khaled, Lauren May, Taylor Yee

Project:

Augmented Power Mobility

Individual Feasibility and Planning Assignment

Emails of Team-members:

Kaa325@nau.edu –	Khaled Khaled
Dbb72@nau.edu –	Daniel Beckett
Lm933@nau.edu –	Lauren May
Tey24@nau.edu –	Taylor Yee

Client & GTA emails:

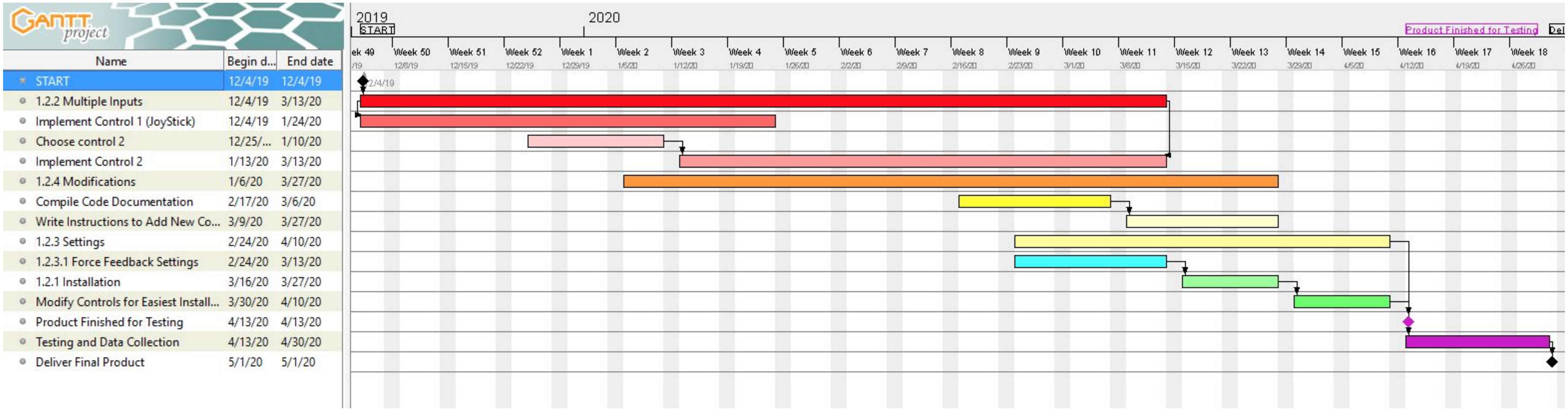
Kyle.Winfree@nau.edu –	Dr. Winfree
si229@nau.edu –	GTA Shafkat

Faculty Advisors:
Client: Dr. Kyle Winfree
Sponsor: Dr. Kyle Winfree
Faculty Mentor: Shafkat Islam

WBS for Driving Controls

- **1. Augmented Power Mobility (AmpEd)**
  A platform that helps disabled children gain experience driving a powered wheelchair which accommodates many disabilities
  - **1.1 Motors and Power Bussing**
    - **1.1.1 Self Sufficient:** The final product should only rely off its own power supply and the main batteries need to have the ability to be recharged when depleted.
  - **1.2 Controls**
    - **1.2.1 Easy Installation:** Any controls need to be optimized for easy attachment to different user's wheelchairs. For each component there should be at least two different methods of attachment to the wheelchair.
    - **1.2.2 Multiple Inputs:** Since there are a wide range of possible disabilities, the design should include at least one other input other than a joystick.
    - **1.2.3 Settings:** The controls should accommodate a variety of disabilities by being adaptable. The physical therapist can use the same control but use them in different ways by changing various settings (i.e. head switches that control direction can be changed to control speed). This is to be implemented through code, no physical components needed.
      - **1.2.3.1 Force Feedback Setting:** The joystick should have 3 settings: 1) No force being applied, 2) Force steering away from objects, 3) Force steering into objects.
    - **1.2.4 Future Modifications:** The final product needs to be designed with the idea of future improvements. There should be proper documentation, including a summary and instructions, on how both controls are implemented in the code so that control modules can be added. Furthermore, any code used to implement the control functions should be in distinctly separate functions from other code.
  - **1.3 Microcontroller**
    - **1.3.1 Handle Multiple Input/Outputs:** The microcontroller must have the capability to handle all inputs (ex. steering using the joystick) while simultaneously sending data collected by the sensors to the GUI.
    - **1.3.3 Log Data Files:** Although the data will be shown on screen in real time, there should be a separate location to just store data from the sensors. Any calculations should be made before storage so if the back up data stored on the remote disk is opened, the data would be able to be interpreted by a physical therapist.
  - **1.4 Sensor I/O**
    - **1.4.1 Wireless:** All data sent from the sensors should be sent wirelessly to the GUI and should not need any wired connection to download except in the case of accessing data stored in the remote disk.
  - **1.5 Graphical User Interface (GUI)**
    - **1.5.1 Display Data and Inputs:** This is the interface that the physical therapist will interact with and should there for be intuitive to use with little to no training. Using the data collected from the sensors, there should be graphs that update in real time that show the velocity and the proximity to any obstacles. There should also be a running totals of the total distance made, how many times the driver had hit an obstacle, and the duration of the ride.
    - **1.5.2 Multiple Settings:** The physical therapist should have control over multiple settings including the force feedback of the driving controls, the maximum velocity, the file that the data on the computer is stored under, what driving controls are being used and an emergency stop.

Gantt Chart for Driving Controls



Name	Begin d...	End date
START	12/4/19	12/4/19
1.2.2 Multiple Inputs	12/4/19	3/13/20
Implement Control 1 (JoyStick)	12/4/19	1/24/20
Choose control 2	12/25/...	1/10/20
Implement Control 2	1/13/20	3/13/20
1.2.4 Modifications	1/6/20	3/27/20
Compile Code Documentation	2/17/20	3/6/20
Write Instructions to Add New Co...	3/9/20	3/27/20
1.2.3 Settings	2/24/20	4/10/20
1.2.3.1 Force Feedback Settings	2/24/20	3/13/20
1.2.1 Installation	3/16/20	3/27/20
Modify Controls for Easiest Install...	3/30/20	4/10/20
Product Finished for Testing	4/13/20	4/13/20
Testing and Data Collection	4/13/20	4/30/20
Deliver Final Product	5/1/20	5/1/20

NORTHERN ARIZONA UNIVERSITY | School of Informatics, Computing, and Cyber Systems

To : AmpEd
From : Taylor Yee
Date : 2nd December 2019
RE : Feasibility Planning

Dear AmpEd,

Attached below is my "Feasibility Planning" document. In this assignment, I tentatively break down our capstone project in two forms: a Work Breakdown Structure (WBS), and a Gantt Chart. Specific emphasis is placed on the graphical user interface (GUI), which I plan on taking the lead in developing. The WBS details several of the deliverables that I believe to be key to the success of the development of the GUI, while the Gantt Chart provides a visual depiction of a timeline for the development.

Very respectfully,

Taylor Yee

Augmented Mobility Platform Education (AmpEd)

Daniel Beckett, Khaled Khaled, Lauren May, Taylor Yee

Project:

Augmented Power Mobility

Emails of Team-members:

Kaa325@nau.edu – Khaled Khaled

Dbb72@nau.edu – Daniel Beckett

Lm933@nau.edu – Lauren May

Tey24@nau.edu – Taylor Yee

Client & GTA emails:

Kyle.Winfree@nau.edu – Dr. Winfree

si229@nau.edu – GTA Shafkat

Faculty Advisors:
Client: Dr. Kyle Winfree
Sponsor: Dr. Kyle Winfree
Faculty Mentor: Shafkat Islam

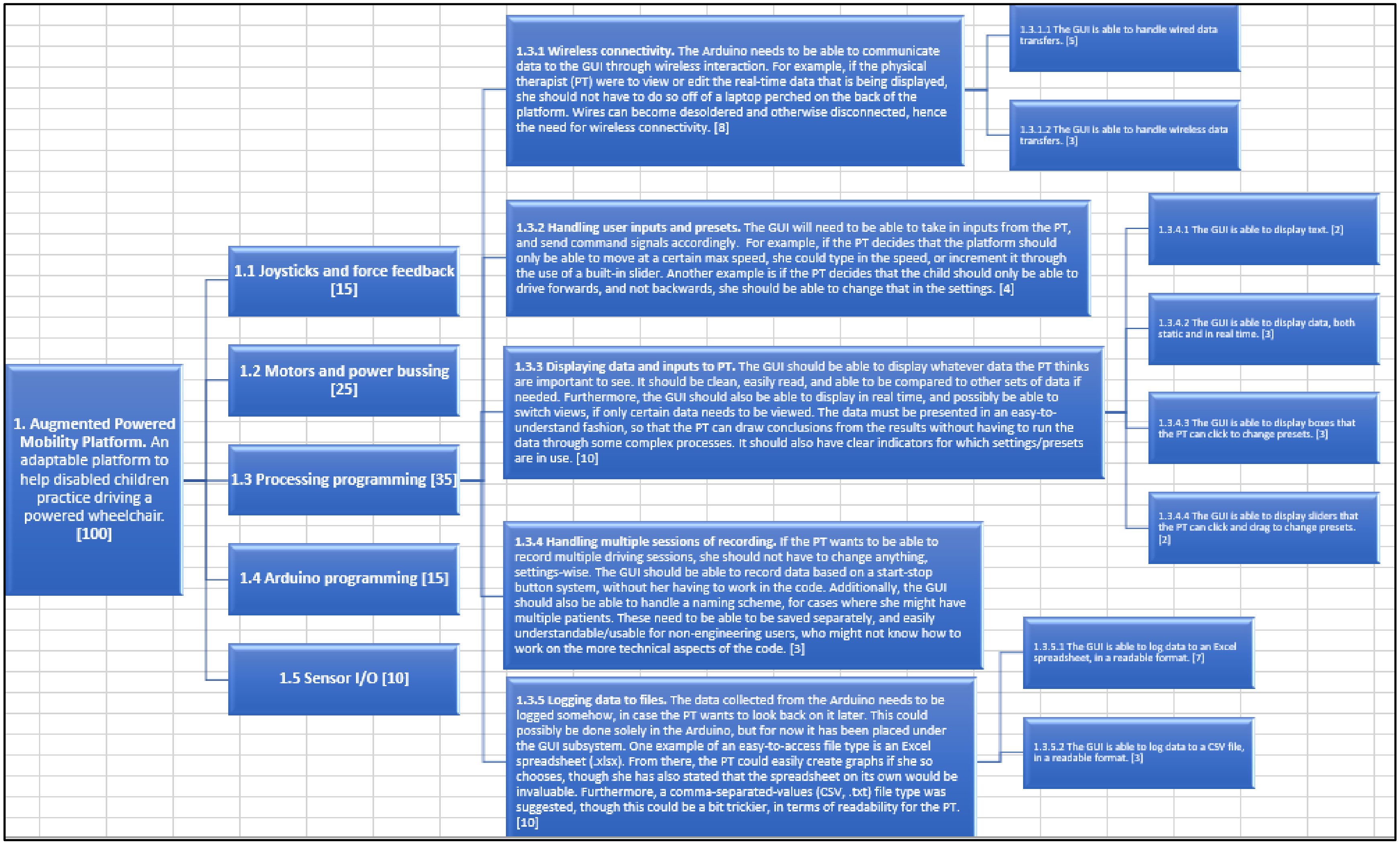


Figure 1: WBS of GUI

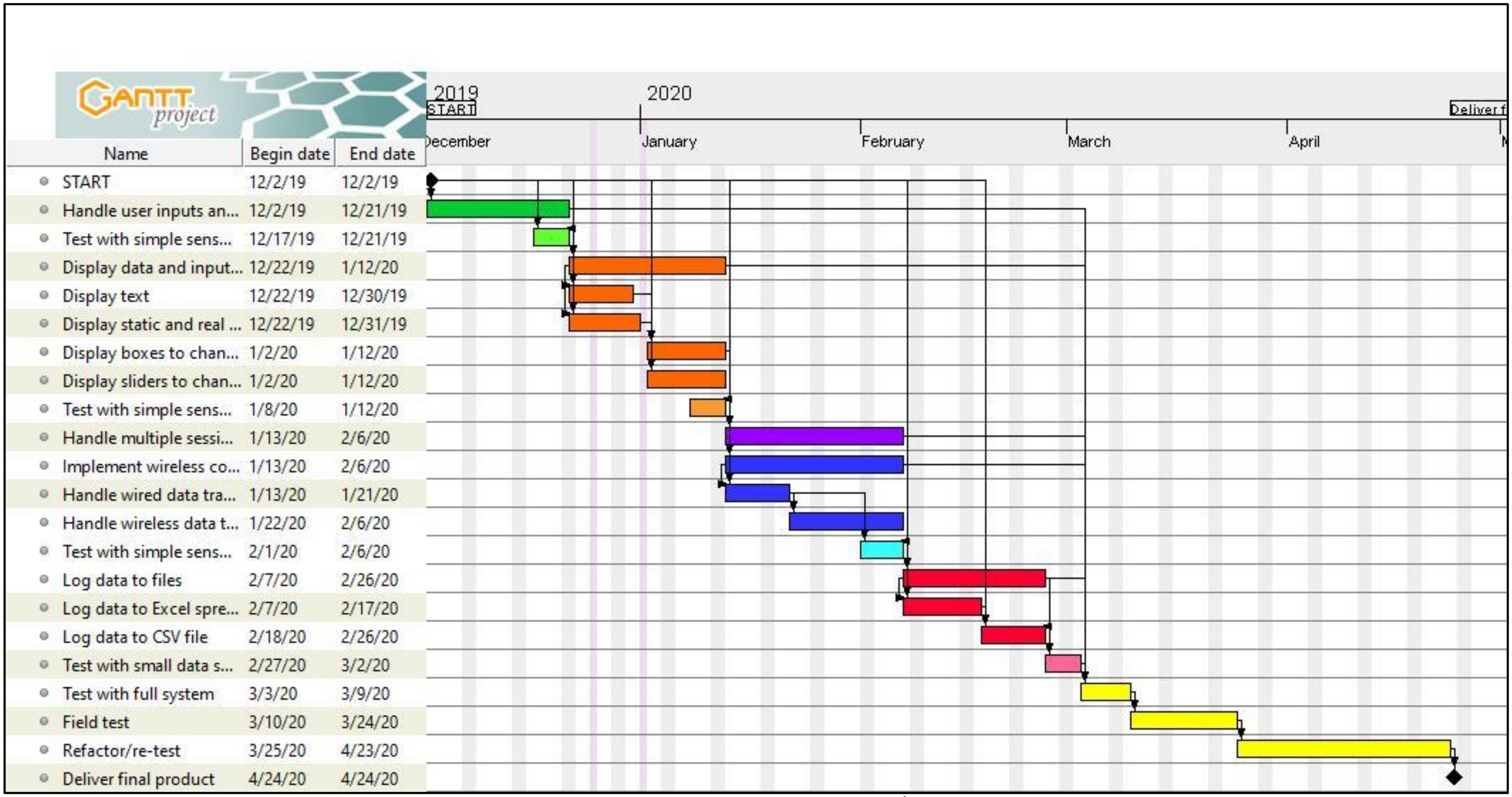


Figure 2: Gantt Chart of GUI[1]

[1] Extra float room was provided for deliverables that fell under holiday breaks (Christmas and spring breaks). Extra padding was provided for the field test and refactoring deliverables.