1. **Process**

How will the project be developed?

The project will be developed using the TM4C123. There will be in-mouth switches, and an IMU to provide different functionality of the device. The system will be built on a PCB and will have several testing components as discussed in the prototype of this document. There will be the following subcomponents within the system: A IMU to get inertial measurements from, USB connection between the microcontroller (the device), and a host PC where the system acts as an HID-compliant USB mouse, In mouth buttons, and a speaker system which passes in audio from a audio jack. The process will be to design and test each of these individually before combining them into one working system. This will be done by programming individual modules/subcomonents separately, and building different components in separate files to facilitate development and testing. Proper documentation will make it easier to jointly develop the code and make it easier for all team members to understand the software.

1. **Roles and Responsibilities**

Who will do what? Who are the clients?

Aeybel - Software

Daniel - Software

Harsh - Hardware  
Brandon - Hardware: PCB design

Clients - TA/Mc.Dermott, patients paralyzed from the neck down, flight-sim enthusiasts.

1. **Scope**

List the phases and what will be delivered in each phase.

The project will be delivered in three different phases. The first is a preparation phase where the bill of materials and basic requirements document will be discussed with the client. The second phase is checkout where the working system will be demoed to the client. The third phase is a report where a completely working system will be presented with different analysis points of the system. Between phases 1 and 2 there will be multiple meetings within the group to discuss changes in the requirements document and testing of project aspects.

1. **Prototypes**

How will intermediate progress be demonstrated?

* Individual circuits for each subcomponent will be implemented on breadboard or perfboard
* Separately test components and debug as necessary.
* Test IMU standalone, then incorporate it into the embedded system.
  + Be able to communicate with the IMU via a protocol such as I2C or SPI
  + Poll measurements from the accelerometer and gyroscope
  + Accelerometer and gyroscope measurements for different gestures are recorded and processed within a library such as SciPy to produce a decision tree displaying different thresholds for the various sensor axis.
    - The decision tree for each gesture is then either turned into IMU FSM instructions if using the IMU to handle gestures or into C code if the TM4C will do gesture recognition
* Test speaker: Ports, speakers, and protection.
* Test USB: Port connections, communication between PC and embedded system

1. **Performance**

Define the measures and describe how they will be determined.

* *Dysfunctional*: Catastrophe; everything has gone wrong and the device does not work.
  + Cannot communicate with IMU
  + Cannot communicate with a host device
  + The circuitry does not work
* *Basic functionality*: The device can operate somewhat at a very basic level.
  + Can communicate with the IMU and get sensor measurements
  + Can send mouse updates to a host device
  + Circuitry works with little to no issue
* *Good functionality*: Operates reliably with small errors allowed
  + Can communicate with the IMU and get sensor measurements
    - IMU can be put into a low power mode, with no issue in communication such as incorrect I2C messages or SPI message
  + Can perform all mouse functionality on a host device, a user should feel as if the cursor moves to the same effect as a normal mouse if not better
    - Subjective measurement: a user shouldn’t feel any noticeable difference in how the cursor moves other than the fact its done with head movement
  + Circuitry works with no issues
    - Optimal power consumption, the TM4C can be put into a low-power state
    - Minimal noise on any data signals such as SPI or USB, measurable with an oscilloscope to see the noise and logic analyzer to see the logic levels
* *Great functionality*: Operates smoothly and without errors consistently.
  + Can communicate with the IMU, IMU handles gesture processing
    - IMU can be put into a low power mode, with no issue in communication such as incorrect I2C messages or SPI message
    - The IMU will perform the gesture processing and trigger an interrupt if it detects a programmed gesture
      * The programmed gestures should work with little to no error. (gestures detected correctly by IMU)
      * The trained gestures should work for a multitude of users
  + Can perform all mouse functionality on a host device, a user should feel as if the cursor moves to the same effect as a normal mouse if not better
    - Subjective measurement: a user shouldn’t feel any noticeable difference in how the cursor moves other than the fact it's done with head movement
    - Change in position vertically/horizontally is proportional to how far the head is pitched/yawed
  + Circuitry and general hardware works with no issues
    - For optimal power consumption, the TM4C can be put into a low-power state and wake on interrupts
    - Minimal noise on any data signals such as SPI or USB, measurable with an oscilloscope to see the noise and logic analyzer to see the logic levels
    - Small PCB footprint, comfortable buttons and head-mounted enclosure

Pick 2 metrics:

We will measure the latency in communication from gesture recognition on the IMU to the moment it is sent over USB.

Less than 5% false positives for gesture recognition