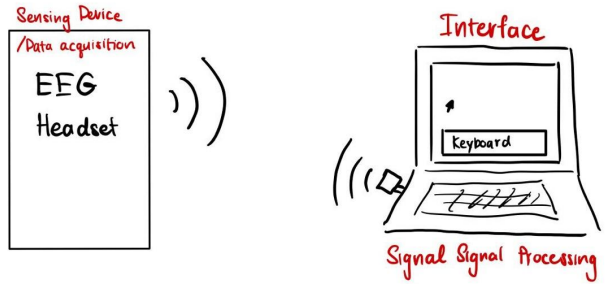
# NeuroController

Team A0: Jean, Jonathan, Wendy

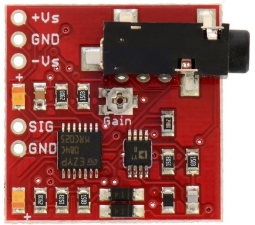
# Application Area

**Goal**: Increase accessibility of computer interfaces for amputees and individuals with difficulty controlling muscle functions.

**Our solution**: Create an application that serves as the mouse/keyboard interface that is controlled through EEG and EMG signals.

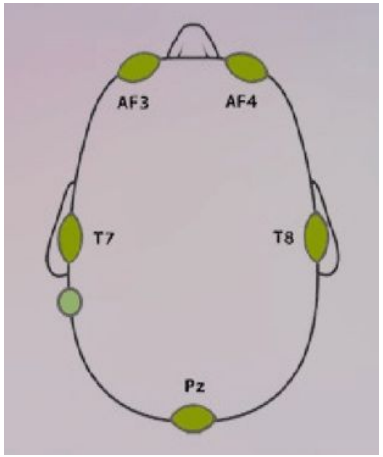


# Hardware




**Emotiv Insight (5 channels electrodes)**

- Acquire EEG data (cortical electrical activity from scalp) from Frontal lobe (AF3, AF4), Temporal lobe (T7, T8), PZ (parietal lobe).




**EMG sensor**

- Measure muscular electrical activity (usually 30mV - 50mV)

**Arduino - UNO**

- Convert EMG analog signals to digital and forward via bluetooth

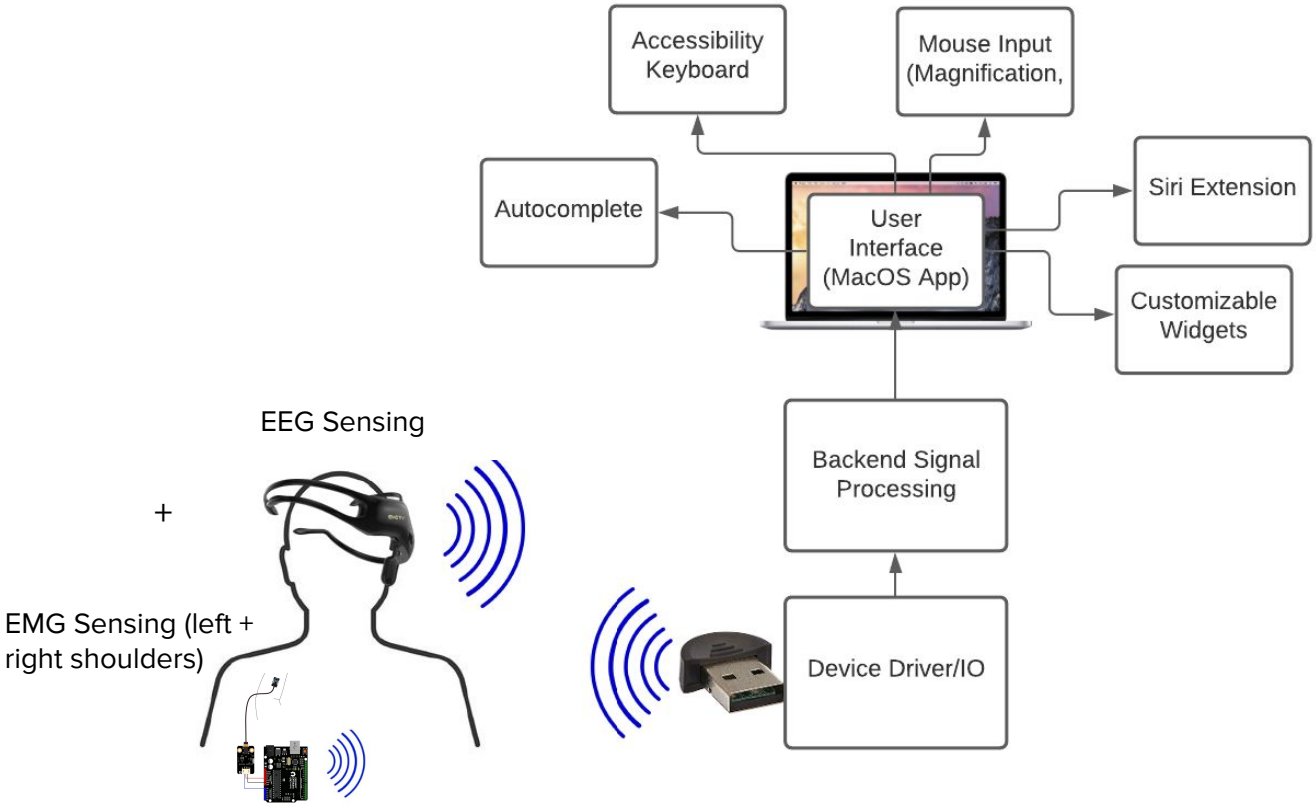
**Bluetooth module - HM10 DSD tech**

- Allows wireless communication between the EMG + Arduino platform with computer.

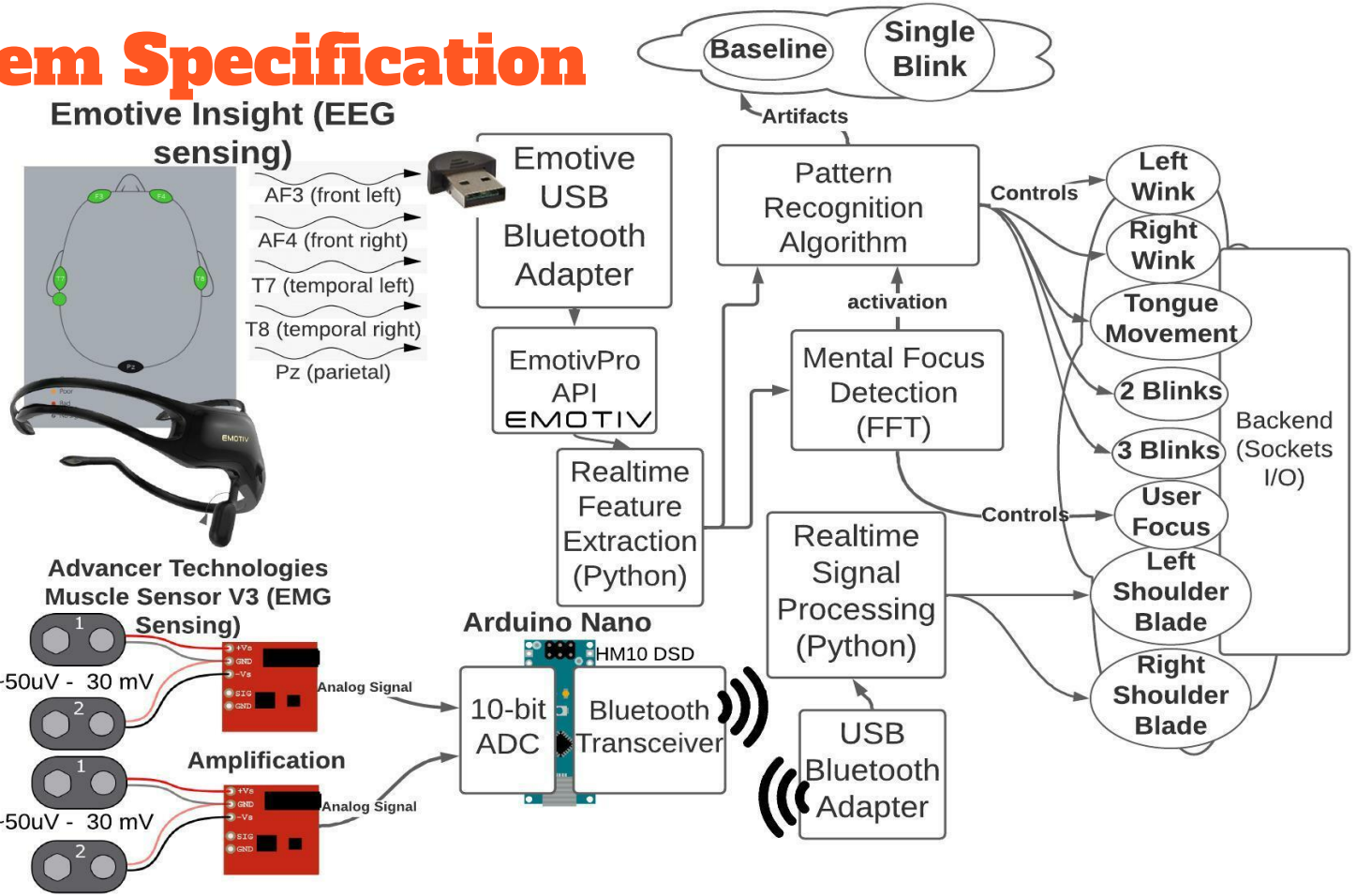
# Software

| Emotiv Pro (Data Acquisition) | Python (Backend) | Flutter (Frontend) |
| --- | --- | --- |
| Signal Acquisition | Processing | User Interface |
| - Acquire EEG signals and make available through real-time to API | - Receive data from EMOTIV<br>- Signal processing and fitting<br>- ML classification | - Mouse<br>- Keyboard<br>- Tab controllability<br>- Autocomplete |

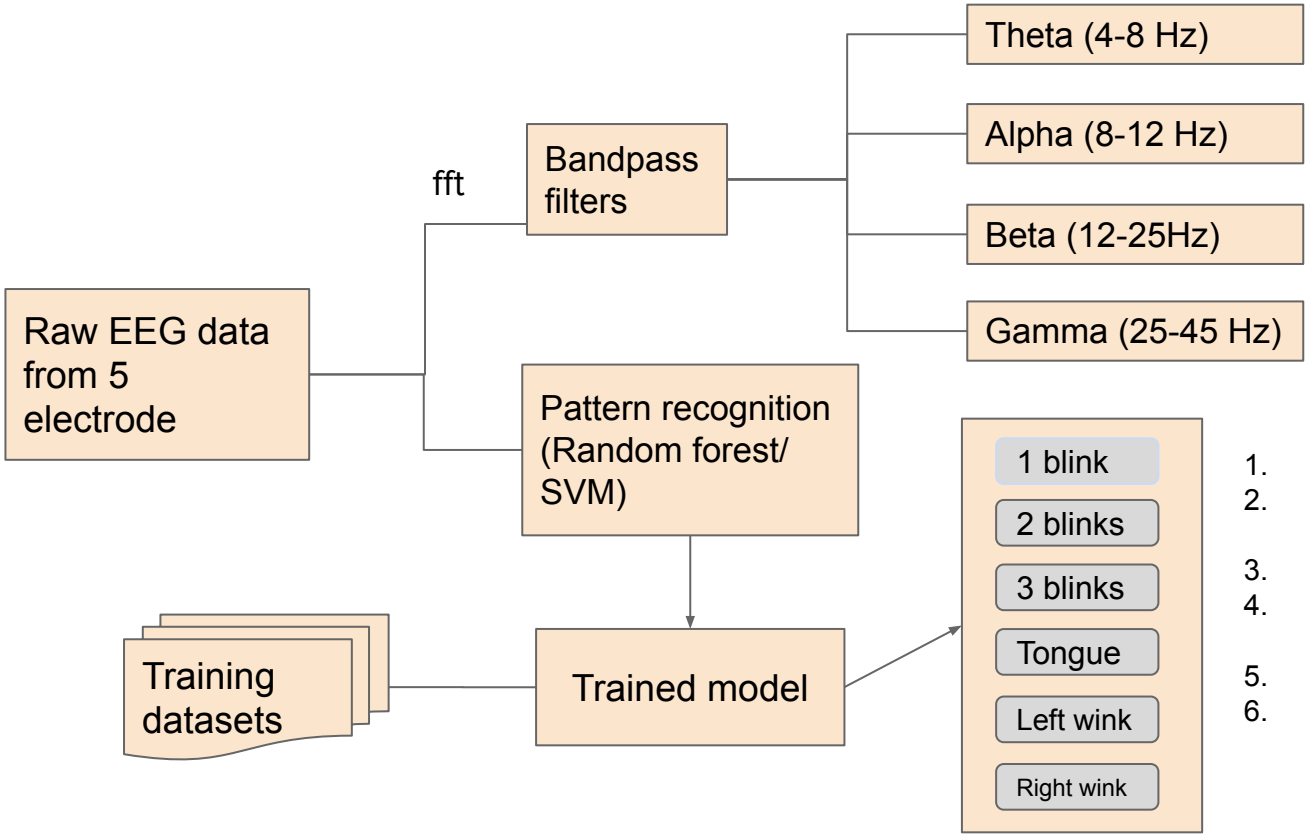
# Solution Approach



# System Specification



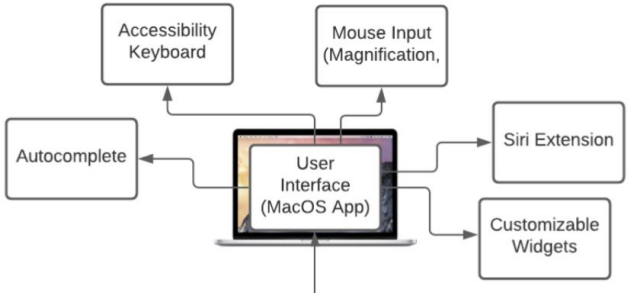
# Signal Processing Algorithm




Sample rate: 250 Hz

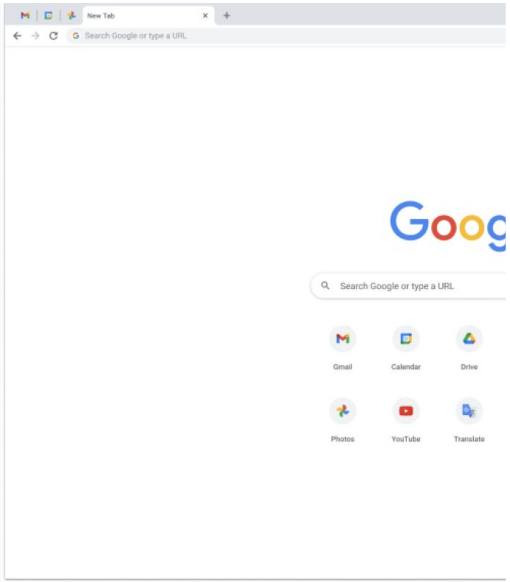
1. Use datasets to train/create a model
2. I/P testing data to ensure the correctness of data detection (97%)
3. Try a mixed datasets
4. Ensure that every types of single detection is accurate
5. Test on sequential data
6. Test on varying sequential signal

# User Interface

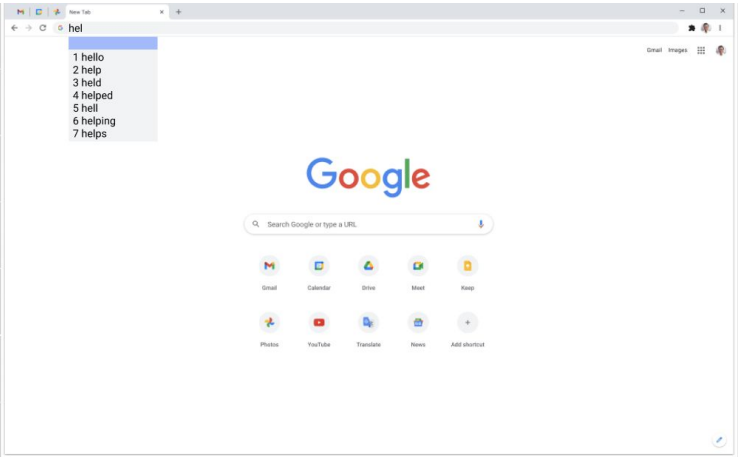



Mouse speed:
100-400 CPI
(count per inch)

| |
|---|
| Page up |
| Page down |
| Go to top of page |
| Go to bottom of page |
| Go to top of page |
| Keyboard/ Mouse |
| Settings |



| | | | | |
|---|---|---|---|---|
| a | b | c | d | e |
| f | g | h | i | j |
| k | l | m | n | o |
| p | q | r | s | t |
| u | v | w | x | y |
| z | Space | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 | 8 |
| 9 | 0 | . | Ret | Bs |
| ? | ! | ; | ‘ | , |
| Shift | Caps | Tab | Help | Back |



# Implementation Plan

## Copying/Downloading

- Bluetooth module code
- API template code
- Emotiv API

## Buying

- EMG sensor
- EEG Headset (EMOTIV Insight)
- Bluetooth module

## Assembling

- EMG wireless sensor platform

## Designing

- Pattern recognition algorithm
- Data fitting training
- UI
- Desktop application

# Metrics and Validation

| Requirement | Testing Strategy | Quantitative Metric |
|---|---|---|
| User Accuracy & speed | - Point and click test | User can click 5 randomly spaced targets within 30 seconds |
| EEG Signal Processing Accuracy | - Signal pattern recognition test | 90% accuracy on identifying labeled test data |
| User Latency | - Human benchmark test | Time user reaction to a stimulus and display reaction within 500 ms (eg. clicking) |
| User Experience | - Ask users who have tried this product if they would recommend this to someone in need | Over 70% recommendation out of 15 people |

# Risk Factors and Mitigation Plan

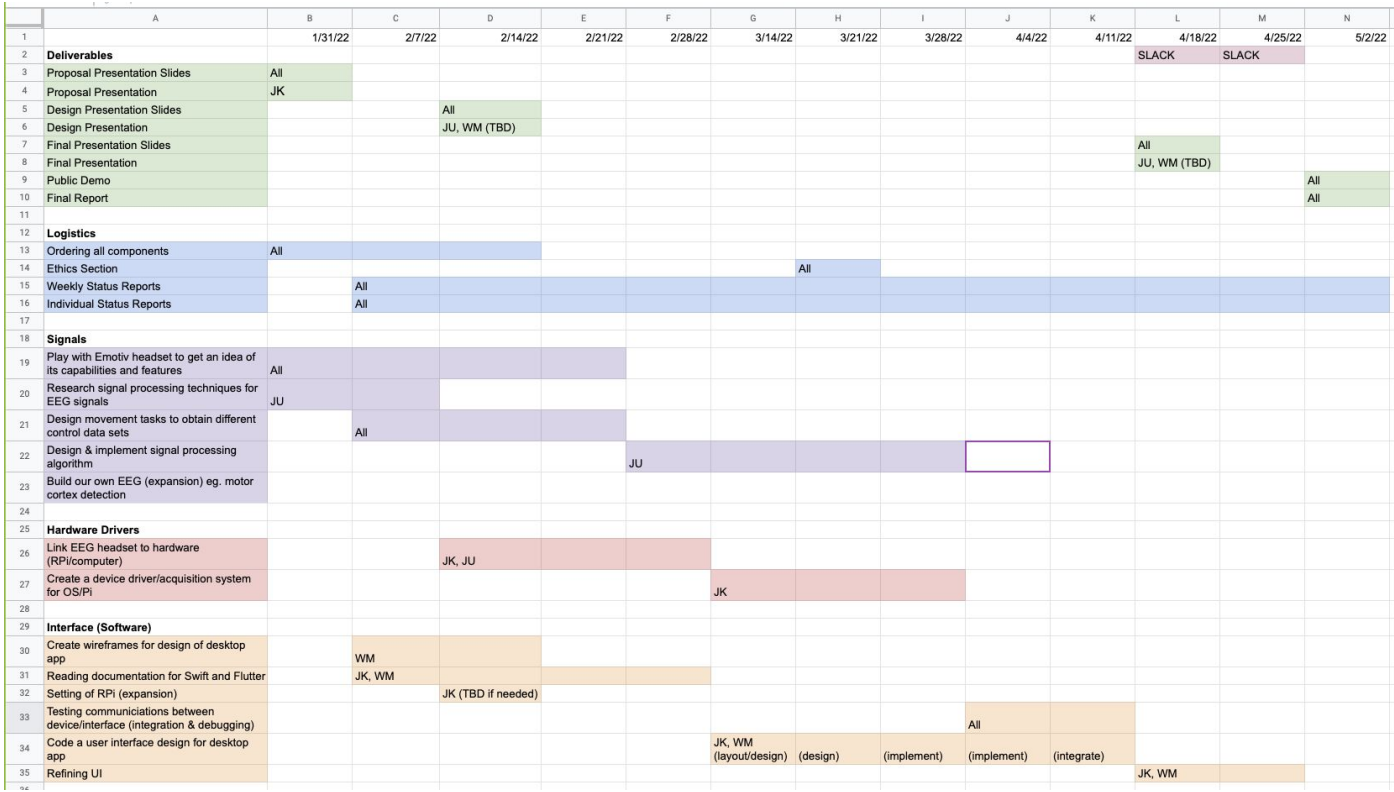
- High latency - Python signal processing, hardware on EMG
- Choosing or generating a pattern recognition algorithm to discern the different signals
- Discern and process a common signal that may vary across multiple subjects

If we want to stick to non-physical transmission of control signals, we will pivot to using CV detection through eye-tracking.

# Project Management

Work Distribution:

- Jean- Signals
  - Signal processing algorithm/research
- Jonathan- Hardware/Software Interfacing
  - Create acquisition and processing backend
- Wendy- Software
  - Create desktop application and interface



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 1/31/22 | 2/7/22 | 2/14/22 | 2/21/22 | 2/28/22 | 3/14/22 | 3/21/22 | 3/28/22 | 4/4/22 | 4/11/22 | 4/18/22 | 4/25/22 | 5/2/22 |
| 2 | **Deliverables** | | | | | | | | | | | SLACK | SLACK | |
| 3 | Proposal Presentation Slides | All | | | | | | | | | | | | |
| 4 | Proposal Presentation | JK | | | | | | | | | | | | |
| 5 | Design Presentation Slides | | | All | | | | | | | | | | |
| 6 | Design Presentation | | | JU, WM (TBD) | | | | | | | | | | |
| 7 | Final Presentation Slides | | | | | | | | | | | All | | |
| 8 | Final Presentation | | | | | | | | | | | JU, WM (TBD) | | |
| 9 | Public Demo | | | | | | | | | | | | | All |
| 10 | Final Report | | | | | | | | | | | | | All |
| 11 | | | | | | | | | | | | | | |
| 12 | **Logistics** | | | | | | | | | | | | | |
| 13 | Ordering all components | All | | | | | | | | | | | | |
| 14 | Ethics Section | | | | | | | All | | | | | | |
| 15 | Weekly Status Reports | | All | | | | | | | | | | | |
| 16 | Individual Status Reports | | All | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | |
| 18 | **Signals** | | | | | | | | | | | | | |
| 19 | Play with Emotiv headset to get an idea of its capabilities and features | All | | | | | | | | | | | | |
| 20 | Research signal processing techniques for EEG signals | JU | | | | | | | | | | | | |
| 21 | Design movement tasks to obtain different control data sets | | All | | | | | | | | | | | |
| 22 | Design & implement signal processing algorithm | | | | | JU | | | | | | | | |
| 23 | Build our own EEG (expansion) eg. motor cortex detection | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | | |
| 25 | **Hardware Drivers** | | | | | | | | | | | | | |
| 26 | Link EEG headset to hardware (RPi/computer) | | | JK, JU | | | | | | | | | | |
| 27 | Create a device driver/acquisition system for OS/Pi | | | | | | JK | | | | | | | |
| 28 | | | | | | | | | | | | | | |
| 29 | **Interface (Software)** | | | | | | | | | | | | | |
| 30 | Create wireframes for design of desktop app | | WM | | | | | | | | | | | |
| 31 | Reading documentation for Swift and Flutter | | JK, WM | | | | | | | | | | | |
| 32 | Setting of RPi (expansion) | | | JK (TBD if needed) | | | | | | | | | | |
| 33 | Testing communiciations between device/interface (integration & debugging) | | | | | | | | | All | | | | |
| 34 | Code a user interface design for desktop app | | | | | | JK, WM (layout/design) | (design) | (implement) | (implement) | (integrate) | | | |
| 35 | Refining UI | | | | | | | | | | | JK, WM | | |

# Conclusion

- Testing:
  - License was approved
- Developing:
- So far on schedule!