Timeframe: 03/20 ~ 03/26

- **Completed tasks:**
  - Reviewed the project brief
  - Reviewed project deliverables
- **Notes:**
  - Was not able to meet expectations due to the university semester just starting and we did not really engage with the project.

Timeframe: 03/25 ~ 04/03

- **Completed tasks:**
  - Identified all deliverables for the project
  - Received hardware (Raspberry Pi 5) for the project
  - Defined project scope, objectives
  - Defined basic work in progress research questions
  - Defined a project management timeline
  - Conducted research on:
    - The idea of sign language recognition
    - Various machine learning frameworks and toolsets
    - Raspberry Pi as a platform for ML applications
    - Limitations and research gaps
  - Draft up our own literature review report
  - Received feedback and performed modifications on literature review.
- **Notes:**
  - We completed our tasks from our previous timeframe in this timeframe effectively and completed all of our expected tasks.

Timeframe: 04/04 ~ 04/15

- **Completed tasks:**
  - We did not do anything
- **Notes:**
  - We failed to meet the expected timeline, completely

Timeframe: 04/16 ~ 04/30

- **Completed tasks:**
  - Setup initial Github repository
  - Created markdown files showing possible datasets to use
  - Created markdown files showing possible methodologies to use
  - Attempt to setup machine learning environment on Raspberry Pi 5 (failed)
- **Notes:**
  - We did not successfully set up the machine learning environment on the Raspberry Pi 5, as we do not have too much experience with Linux systems.
  - We also did not perform any train/test/validation split on our datasets.

Timeframe: 05/01 ~ 05/30

- **Completed tasks:**
  - Deeper look into model architectures from research
  - Set current focus to static sign language gestures on American Sign Language
  - Setup machine learning environment on Raspberry Pi 5
  - Performed train/test/validation split on American Sign Language dataset
  - Created basic CNN model for sign language recognition (poor accuracy)
- **Notes:**
  - We sort of met our expectations by completing our expected tasks, but the task of training a CNN model was a bit under our expectations because it did not yield good results.

Timeframe: 06/02 ~ 07/04

- **Completed tasks:**
  - Continued to develop CNN models, narrowed down to MobileNetV2.
  - Performed iterative experiments on MobileNetV2 models.
- **Notes:**
  - At this stage we have not been able to discover bottlenecks, as training and testing are all done on a desktop environment with strong computational resources.
  - We also did not do documentation at this stage.

Timeframe: 06/13 ~ 07/17

- **Completed tasks:**
  - With "good enough" CNN models, we deployed on to the Raspberry Pi 5 for some testing without quantization.
  - We added YOLO object detection model to our system, which drastically improved our accuracy by leveraging its ability to detect a hand and run our MobileNetV2 classifier on the hand region.
- **Notes:**
  - At this stage we only cared about accuracy, but we did notice that running the raw model on the Raspberry Pi 5 without quantization yielded lower accuracy compared to a desktop environment.

Timeframe: 07/28 ~ 08/29

- **Completed tasks:**
  - We split into two, Tony to perform optimization for our YOLO + MobileNetV2 model on the Raspberry Pi 5 to increase accuracy and speed performance. Jerry to look into dynamic hand gesture recognition.
  - Custom images were gathered to prevent overfitting on static model.
  - Jerry walked down the path of utilizing LSTM model architecture.
  - For dynamic we looked into using the WLASL dataset.
- **Notes:**
  - Jerry's attempts at LSTM failed miserably.

Extra-Timeframe: 08/29 ~ 09/15

- **Completed tasks:**
  - Tony continued on optimizing the static system.
  - Jerry continued on the LSTM dynamic system.
- **Notes:**
  - Not much positive results were outputted during this stage.

Extra-Timeframe: 09/16 ~ 10/12

- **Completed tasks:**
  - Tony continued on optimizing the static system.
  - Jerry implemented a transformers architecture on the LSA64 dataset for the dynamic system.
  - Tony implemented brand new dual language static system using similar architecture but with new BSL (British Sign Language) dataset.
- **Notes:**
  - ASL + BSL system yielded great results
  - Dynamic LSA64 system yielded great results on validation split of dataset