Table 1: Revision History

Date	**Developer(s)**	**Change**
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...	...	...

Development Plan
Mechatronics

Team #20, Team Name
Robert Zhu zhul49
Zifan Meng mengz17
Jiahui Chen chenj194
Kelvin Huynh huynhk12
Runze Zhu zhur25
Mirza Nafi Hasan hasanm21

[Put your introductory blurb here. —SS]

1 Team Meeting Plan

The team is planned to have a team meeting after every 4TB6 lecture and during the lecture time if there is no lecture on that day at the library or an alternative workspace. The team meeting is organized three times a week and each meeting is two to three hours depending on the project. Meeting times are not completely fixed and more may be scheduled to accommodate deadlines and the progress of the project. There will be a meeting agenda for every meeting, and the agenda will include the chair for that meeting, the planned tasks, final decision and take-home deliverables for each team member. The chair for the meeting will be dependent on the content of the meeting. For example, if the meeting is about hardware selection, the team member who is responsible for hardware will be the chair for the meeting.

2 Team Communication Plan

Team communication outside of meetings is to be conducted primarily on Facebook Messenger for administrative related matters and project specifics. Communication relevant to the project source code is encouraged to be conveyed through the use of Github Issues and/or Pull Requests but may also be done through Facebook. In addition, communication to the relevant supervisor, stakeholder, TA and/or course instructor will be conducted through the use of email.

3 Team Member Roles

As of now, every team member will have the same responsibilities consisting of coding, identifying issues, testing, reviewing, and commenting on code. Further responsibilities will include hardware-software integration and testing. More responsibilities will be added as more specific tasks are discovered during the development process.

Table 2: Roles and Responsibilities

Name	**Responsibilities**
Robert Zhu Zifan Meng Fred Zhu Mirza Nafi Hasan	Coding, issue identification, testing, reviewing, and commenting on code. Additionally hardware-software integration.
Jiahui Chen Kelvin Huynh	Research, hardware

4 Workflow Plan

- How will you be using git, including branches, pull request, etc.?
- How will you be managing issues, including template issues, issue classificaiton, etc.?

5 Proof of Concept Demonstration Plan

Risks that can affect the success of the project include poor hand joint tracking (which would have an effect on accuracy) and potentially a faulty machine learning model that produces the incorrect results. Another risk would be that none of us have prior knowledge of signing so that would affect the number of signable words and phrases that we can use to further test validity and reliability of the machine learning model. During the proof of concept demo, we hope to provide a mixture of basic signs and some complicated signs such that we can see that we're on the right track and that some part of the prototype works. This will enable us to further refine our machine learning model and provide a baseline that we can revert back to if it fails.

6 Technology

The coding for the project will be done in Python3 utilizing Flake8 as the linter to ensure error-free and idiomatic code. Unit testing for the Python code will be done through the use of the Pytest framework where various tests can be defined based on the intended code functionality. The same framework of which can and will be used to generate a measure of code coverage through the pytest-cov plugin. Continuous integration (CI) is planned to be used to ensure that coding errors and bugs are detected within a reasonable amount of time, however the specifics are to be determined as the group is unfamiliar with implementing the concept at this time. Libraries that are currently planned for use include OpenCV, Tensorflow, and Pyserial for serial communication with an Arduino board. Performance measuring tools will be used appropriately as the need arises, but some examples would include OpenCV's getTickCount() and getTickFrequency() functions as well as Python's time.perf_counter() function to track execution time of code. Additional tools may be declared as the need arises.

7 Coding Standard

Code will loosely follow the PEP8 Python coding standard. The Flake8 linter will also ensure that written code will follow this standard.

8 Project Scheduling

[How will the project be scheduled? —SS]