Table 1: Revision History

Date	**Developer(s)**	**Change**
09/25/22	Muhanad Sada	Workflow Plan, POC Demo Plan, Technology
09/25/22	Ahmed Nazir	Technology

Development Plan
MECHTRON 4TB6

Team 25, Formulate
Ahmed Nazir, nazira1
Stephen Oh, ohs9
Muhanad Sada, sadam
Tioluwalayomi Babayeju, babayejt

1 Team Meeting Plan

The team plans on having in person meetings weekly on Mondays from 2:30PM - 4:30 PM. These meetings will be used to catch up on work done in the last week, next steps and any important updates. If in person meetings are not possible, we will conduct them through Microsoft Teams. This weekly meeting is mandatory but we may also have other meetings throughout the week depending on specific project needs. All meetings will have an agenda and each team lead will give updates on the next steps.

2 Team Communication Plan

Our team will use Microsoft Teams as our main method of communication via chat and GitHub's issue tracker to delegate specific deliverable tasks.

3 Team Member Roles

- Stephen Oh - Electrical and Mechanical Hardware Lead
- Ahmed Nazir - Mechanical Hardware Lead, Firmware Lead
- Muhanad Sada - Software User Interface Lead, Database Lead
- Tioluwalayomi Babayeju - Software Dashboard Platform lead

4 Workflow Plan

Our team will use the GitHub repository dedicated for the capstone project. The feature branch workflow will be used whenever there are any code changes

except for simple fixes such as syntax errors, comments, variable renaming, etc. Branches will be utilized for significant documentation changes such as section additions or modifications and diagram insertions. Pull requests will be used in conjunction with branches to review code and document changes. Branches will follow the following the naming structure of scope/description. For example: Feat/Adding new function.

4.1 Issues

The issues feature in GitHub is used to track all project tasks. Once the team or individual members identify a task, an issue will be created. In order to keep issues updated with the corresponding commit, the following commit message structure will be followed: (scope): [#issue number] description. When creating an issue, a team member will select one of the issue templates based on the scope of the task. There are a total of five templates:

- Bug report - any tasks used to report a bug and fix it
- Feature – any tasks that involve requesting and implementing a feature
- Enhancement - any tasks that require updating code for enhancement purposes
- Documentation – tasks that involve adding or editing documentation
- Miscellaneous – any tasks that are not covered under the scope of the other templates

Labels will be added to issues to identify the affected product subsystem. The following labels will be utilized: database, desktop application, GUI, hardware, and mechanical design.

4.2 Project Board

The project board will be used to organize and identify the status of each task. The project contains five columns each describing the current status of the issue:

- To-do - When tasks are first created, they are placed in this list
- In-progress – The issue has been assigned to a team member and is currently being worked on
- In-Review – The work has been completed and now to needs to be reviewed
- Done – Once team member(s) review and approve the changes, the issue will be moved to this stage
- Outdated/Ignored – issues that were created but later determined to be unnecessary

5 Proof of Concept Demonstration Plan

The proof of concept demonstration will prove three essential product functionalities. The first is the sensor's ability to measure and output the desired data and to the computing hardware. The second capability will be to receive and send data between three different subsystems, which includes the electrical hardware, desktop application, and database. The POC will show that hardware can receive information from a sensor and send the collected sensor data to a simple desktop application. The application should then be able to receive and display the sensor data on the GUI. At this point, the application gives the user the ability to send the sensor data information to a database, populating relevant tables accordingly.

6 Technology

6.1 Programming Language

The programming language of choice is Python. Python provides the ideal balance between data analysis, data manipulation capabilities, libraries available for data analysis, and documentation widely available to support development.

6.1.1 Libraries

- Pandas: Data manipulation library
- PyQT: Python GUI library

6.2 Software Tools

To assist with developing our product we will be using the following software tools.

6.3 Hardware

- Arduino Board: Open-source microcontroller
- Wifi module: Gives Arduino Wifi access
- Sensors: Measuring desired values
- 3D Printer: Printing CAD designed electronics enclosure and supporting mechanical components

Tool Name	Explanation
Autodesk Inventor	Autodesk's Inventor will be used to complete CAD component and assembly design work to create our hardware
KiCad	Kicad's schematic editor will be used to complete the electrical design of the electrical hardware component's power and signal connections. Kicad will also be used to generate the PCB layouts and trace connections
VSCode	Our team will be using VSCode as our primary code editor because of large number of extensions available. VSCode also integrates well with Git and GitHub
GitHub Desktop	GitHub desktop is an easy GUI to use and interacts with our GitHub repo to make code and file editing more efficient
Arduino IDE	Software used to write and upload firmware code onto the Arduino board
Amazon RDS	Also known as the Amazon Relational Database Service that allows you to create, connect to, and manage database instances

7 Coding Standard

Coding Standard	Explanation
Coding readability	• Capitalize SQL key words to differentiate them from columns and table names • Avoiding deep nested loops to help make it easier to follow and read • Avoid creating long function that do multiple tasks, instead make small functions which do a single task. • Writing comments consistently explaining what is happening in each section of the code • Using meaningful variables to help make code more understandable • Using appropriate naming conventions
Module headers	• Creating module names • Creation date • History of changes to modules • Summary of what each module does • Functions and variable name changed in each modules • Tracking various issues in each module
Proper indentation	• Proper space between two function arguments after a comma • Proper indentation and spaces for nested blocks in code • All braces should start from a new line and the end of the braces also start on a new line

8 Project Scheduling

Weekly meetings to identify product development will be used to manage our project's development progress and meet key development deadlines. We will go over each person's delegated tasks and see if they were met during the meeting. If the target was not met, we analyze how the setback affects the development schedule and allocate additional resources accordingly to ensure completion. We will have major deadlines added to a shared calendar on teams to maintain high deadline visibility as we work on our project. At the beginning of each month during our Monday meeting, we will discuss potential roadblocks and what must be completed for the month. Delegating tasks will be made during weekly meetings and are based on the schedules of each individual and their expertise.