Project Assignment 1

Team Name: WM Codeworks
Team Members: Wesley Skywalker, Matthew Heseltine
Course Code: CPT_S-421
Date: 9/19/2025

1. Functional Requirements (FRs)

(List at least 8–10. Use FR-01, FR-02 ... numbering)

- **FR-01:** The system must be able to run through data at variable speeds.
- **FR-02:** The system must plot the data into a visual interface.
- **FR-03:** The system must allow the user to interact with the visual interface to narrow down data or manipulate it.
- **FR-04:** The system must be able to calculate and plot the pressure and the derivative of the pressure.
- **FR-05:** The system must be able to convert the voltage reading from the input to microliter values.
- **FR-06:** The system must allow loading of previously record data and display user friendly errors if the user try to load an incompatible file.
- **FR-07:** The system must offload intensive tasks like loading a file to separate threads, so the UI remans responsive even under heavy tasks.
- **FR-08:** The system must be able to automatically resize the UI to properly fit the user desired size.

2. Non-Functional Requirements (NFRs)

(List at least 5–7. Use NFR-01, NFR-02 ... numbering)

- **NFR-01:** The system will read CSV files and use the data stored in them to plot them onto visual media.
- **NFR-02:** The system will produce graphs and other visuals using python libraries dedicated to user interfaces.
- **NFR-03:** The system will use pre-existing code from other similar modules to allow the user to select discrete sections of the data for specific information.
- **NFR-04:** The system will be able to run stably for over 8 hours without any addition input from the user outside of the initial setup.
- **NFR-05:** The system will be able to read data in realtime with less than 2 seconds of latency.

3. User Stories with Acceptance Scenarios

*(Write as many as reasonably possible at this stage. Each story must be peer-reviewed in the team. Add **at least two scenarios per story** — one positive, one negative/edge case.)*

US-01:

As a student/researcher I want the system to plot data so that I can collect results for experiments.

Acceptance Scenarios (Gherkin):

Scenario 1: Positive Flow

Given the system is reading voltage data.

When the data is plotted onto a graph

Then it should be easy to see and understand and be able to be parametrized.

Scenario 2: Negative/Edge Case

Given the system is reading voltage data.

When invalid or corrupt data is read.

Then instead of crashing, the input should be ignored and a message displayed reporting the incident.

US-02:

As a researcher/student who isn't fully familiar with the inner workings of the system, I want calibration to be easy so that I can focus on the experiment instead of troubleshooting.

Acceptance Scenarios (Gherkin):

Scenario 1: Positive Flow

Given the system requires its initial calibration.

When the user enters the calibration points.

Then the system can continue performing.

Scenario 2: Negative/Edge Case

Given the system requires its initial calibration.

When the user incorrectly calibrates the system.

Then the system should recognize the calibration inputs as irregular and propose a solution to the user.

US-03:

As a researcher/student I want to be able to upload CSV files to the system so I can see the species of gasses in the sample.

Acceptance Scenarios (Gherkin):

Scenario 1: Positive Flow

Given the system has gone through calibration.

When the user selects a CSV file as input.

Then the system will plot the data visually for the user.

Scenario 2: Negative/Edge Case

Given the system has gone through calibration.

When the user selects a corrupted or invalid CSV file as input.

Then the system should recognize the file is corrupt and display an error message instead of crashing.

US-04:

As a researcher/student I want to be able easily see what species of gasses are present in the sample.

Acceptance Scenarios (Gherkin):

Scenario 1: Positive Flow

Given the system has processed a file.

When the user wishes to see the composition of the sample.

Then the system will display relevant information in an easy to digest manner.

Scenario 2: Negative/Edge Case

Given the system has processed an empty or corrupted file.

When the user wishes to see the non-existent composition of the sample.

Then the system will display an error message regarding a missing or corrupted file.

US-05:

As a researcher/student I want to be able narrow down the results of the processed file based off time.

Acceptance Scenarios (Gherkin):

Scenario 1: Positive Flow

Given the system has processed a file.

When the user wishes to narrow down the results.

Then the system will allow for the user to easily specify the start and end time of the portion they are interested in.

Scenario 2: Negative/Edge Case

Given the system has processed a missing or corrupted file.

When the user wishes to narrow down the results.

Then the system will display an error message regarding a missing or corrupted file.

(Repeat for all user stories US-03 … US-XX)

---

4. Brainstorming and GenAI Reflection

Step A: Team Brainstorming

(Document initial FRs, NFRs, stories, and scenarios created without AI. Record 3–4 key decision points from your discussion.)

- Decision Point 1: The system should be simple to use. The users will not computer scientist but plant scientist. The system should troubleshoot itself as much as possible.
- Decision Point 2: The users will be scientist; therefore, the visuals produced from the data should be in depth and come equipped with tools to better understand the data.
- Decision Point 3: The users should not have to spend a lot of time learning how to use the system. Functionality for some form of 'handholding' and input validation should be included.

Step B: GenAI-Assisted Brainstorming

(Summarize what GenAI suggested — new FRs, NFRs, user stories, or scenarios.)

- GenAI Suggestion 1: Automatically labelling peaks as specific gas species instead of letting the user label them themselves.
- GenAI Suggestion 2: Create a calibration library which allows the user to input previously made calibrations.
- GenAI Suggestion 3: Allow the system to export a summary of the experiment such as minimums and maximum voltages, number of errors/invalid data points, etc.

Step C: Refined Requirements & Stories

(Update Step A with insights from GenAI. Clearly mark new/modified items — e.g., FR-06 ***[Added after GenAI]****.)*

- **FR-06 [Added after GenAI]:** The system must allow loading of previously record data and display user friendly errors if the user try to load an incompatible file.
- **FR-07 [Added after GenAI]:** The system must offload intensive tasks like loading a file to separate threads, so the UI remans responsive even under heavy tasks.
- **NFR-01: [Modified after GenAI]:** The system will read CSV files and use the data stored in them to plot them onto visual media. Other forms of files should also be support or rejected by the system with accompanying error messages.

Step D: Reflection (200–300 words)

- How did you feel about using GenAI in this exercise (e.g., empowering, surprising, confusing, over-reliant)?

Using generative ai to help come up with user stories feels very useful. My only concern is when I catch myself going to ai before I've even tried of thinking for myself. It's a bad habit and very easy to fall into. That being said, once I've put in the time and effort and have exhausted all ideas, ai becomes an excellent productivity tool. I often find myself in narrow field of thinking, using ai to illuminate other pathways keeps me working past the point where I would have tapped out. A lot of its ideas are pretty surface level which makes sense based from the small amount of information it's been given. Because of this I've found it works great for fleshing out and building from unfished or pre-existing work instead of creating new ideas on its own.

- In what ways did GenAI change or improve your brainstorming compared to your team's initial work? (Consider clarity, creativity, and coverage.)

  GenAI has been very helpful in expanding upon simple and incomplete ideas. It does a very good job fleshing them out and from its output new ideas can be branched from it. Sometimes it oversteps and makes assumptions about things or features which do not exist, but those can be easily ignored. It also does a very good job providing robust coverage. It always seems to be able to 'think' about another condition which I have may have overlooked. It's also great at flat out making things sound better to read. Very useful for making technical sentences more understandable.

- Did GenAI help you uncover new functional, non-functional, or business requirements that you had not considered before? Provide examples.

  GenAI suggested automating the calibration of the system by labelling peaks with the species of gases. This uncovered a functional requirement we hadn't considered. Not only should the system accept calibration data, but it could also proactively assist by matching peaks to a stored calibration library and flagging mismatches. This reduces user error and saves researchers from having to manually double check calibration curves.

  Another example is the suggestion to build a calibration library manager. Initially, we only considered plotting data and allowing users to interact with it. GenAI expanded this by proposing the ability to store, edit, and update calibration standards for commonly used gas species. This requirement adds long-term usability, since experiments often rely on different mixtures and conditions.

Final Submission Checklist

- Functional Requirements (8–10+)
- Non-Functional Requirements (5–7+)
- User Stories (all possible at this stage, with acceptance scenarios)
- Gherkin Scenarios (≥2 per story, reviewed)
- Brainstorming & GenAI Section (Steps A–D complete)
- Reflection (200–300 words)
- PDF format, file named correctly

File name format:
TeamName_Assignment1_Requirements.pdf