Software Requirements Specification for Software Engineering: Alkalytics

Team 21, Alkalytics
Sumanya Gulati
Kate Min
Jennifer Ye
Jason Tran

October 27, 2024

Contents

List of Figures

List of Tables

Volere Template Changes

Section Name	Action
6.3 Work Partioning	Removed
6.4 Specifying a Business Use Case (BUC)	Removed
17.1 and 17.2	Merged sub-sections
21.3 Planning of the Development Phases	Added

Revision History

Date	Version	Notes
1 October 2024	1.0	Partially completed version shared with supervisors and TA for feedback.
7 October 2024	1.1	Incorporated feedback and presented during informal presentation with TA.
11 October 2024	1.2	Finished document for Revision 0 submission.
25 October 2024	1.3	Updated safety requirements to align with HA documentation.

1 Purpose of the Project

1.1 User Business

This project aims to aid in the data management and analysis of an ocean alkalinity enhancement experiment process. The research is working towards a scalable process to capture CO_2 using a combination of electric fields and membranes.

The experiment's efficient generation process creates a very dilute base. As earth's temperature rises, so does the ocean's, affects its pH balance. The experimental study aims to provide a solution, to decrease the ocean's pH levels to be able to absorb more CO_2 which will in turn help reduce global temperatures.

As the production process to generate this dilute base is still on a small scale, it is currently being perfected which means many more experiments must be done to be able to bring it to a global scale. However, this requires a massive production operation to do so. Optimization of the experimental data is critical to improve process efficiency. This is a big data software problem requiring the ability to find and fine-tune specific parameters.

1.2 Goals of the Project

This application will be able to consolidate and organize the data from the experiments with proper labeling across all given datasets allowing for a centralize method of data storage that is both scalable and maintainable.

On the application users can request to see certain data points from the inputted data sets given any specified order. Once the data is returned to the user the application can show inter-parameter comparability to better aid data analysis.

This comparability acts as a starting point to analysis and is by no means show a final analysis of the data. This will all be presented in a web interface where all the user functions will be displayed and can be shared among those involved in the experiment.

2 Stakeholders

2.1 Client

Dr. Charles de Lannoy serves as the main client for this project as he is the lead supervisor of the research study. This solution directly affects his work and is intended to be a custom solution for the problem. Bassel Abdelkader is another client of this project as he is the person that works directly with the research data. One of his responsibilities is to record the experimental data and upload them to their current data storage system, Microsoft Excel.

2.2 Customer

Although this project is a tailored solution to one research study, its application can be extended to any other situation where large sets of data is involved. This could be shared among other researchers to aid in their data management and analysis. Depending on the research team's structure, there could also be different levels of permissions that can be obtained.

2.3 Other Stakeholders

Current student assistants and members of the lab working on the study is can also be considered stakeholders for the same reasons as the clients. However, since they will only be working with the study for a short amount of time without daily or consistent interaction, they do not serve as a main stakeholder. The founder of the study, who is currently funding the research project is another stakeholder. However, since they do not work directly with the processes of the study rather oversee the process, they may not have strong interest in the details of the solution.

2.4 Hands-On Users of the Project

The following is a special type of stakeholder as after this capstone project term, the project will be passed back to the research team to maintain.

- User name/category: Research project team for maintaining this project post capstone

- User Role: Maintain the codebase, adding new features, fixing potential bugs project post capstone
- Subject matter experience: Master knowledge on the goals of the research project. Master knowledge on how the experiment processes work.
- Technical knowledge: Novice knowledge on the technology stack being used in this project, Python, MongoDB, JavaScript, other front-end frameworks

2.5 Personas

- John Doe is an 23 year old McMaster undergraduate student who has a research position on the ocean alkalinity research project. They have been tasked to aid the experiment data collection process. After being told that the data is being stored in a master Excel file; they find that is it hard to use. Being an engineering student without much experience with Excel, they struggle to find the data they want. Inputting data is still a manageable process but they find themselves to be spending a lot of time looking at Excel documentation which they find frustrating as that time could be allocated to being more productive during the school term. Although, they want to a better way to manage the data, they know that it is not up to their decision on what tools are being used but suggested that there could be another better solution to use.
- Dr. Carly Kelvon is a 60 years old professor at a university and is working on her own research project for over five years. She has gathered lots of data and thankfully she has always been great at Excel. However, other the last two years she has found that Excel is becoming less sustainable. The queries are a lot slower and sifting through pages and pages of data is wasting a lot of her time. She sees this more evidently through those that work along side her as they are also facing the same struggles with even less Excel experience as her. She wants to find a more scalable solution but she fears that her lack of digital knowledge will do her more harm than good, as a result she fears that if she introduces a new application to serve her needs better that she will find it hard and confusing to use.

- Dr. Alex Stark is a 30 year old associate professor who has recently gotten funding for his innovative research idea and has been dedicating all his time on perfecting its methodology. It has only been one year since his research started but had recently found a great application of his ideas to reach far more people and be more impactful that he had originally thought. But with his current data management set up, he quickly realises that it is not sustainable. He finds that there are many other solutions on the market but they do not exactly meet his needs and cost a lot more than what he can spend on a tool. He decided that the best way is to create his own tool but lacks the software knowledge to create something stable and reliable.

2.6 Priorities Assigned to Users

2.6.1 Primary Users

- Dr. Charles de Lannoy
- Bassel Abdelkader
- Student Assistant working on the experiment

2.6.2 Secondary Users

- Researcher with their own research studies
- The founder of the study

2.7 User Participation

Since this project is a personalized solution for a research team, there is no other user participants other than the research time and others teams with a similar need.

2.8 Maintenance Users and Service Technicians

Not a relevant section

3 Mandated Constraints

The following constraints highlight the restrictions and limitations imposed by the stakeholders, impacting the implementation of the product.

3.1 Solution Constraints

MC-1. The product must accept Comma-Separated Value (CSV) files as input.

- *Rationale*: The lab apparatus generates and stores results as CSV files.
- *Fit Criterion*: The product's input process (the processing and acceptance of input data) into the database shall be approved by testers and developers.

3.2 Implementation Environment of the Current System

MC-2. The product must be able to run on a Windows machine.

- *Rationale*: Currently, the lab has a Windows machine that is used to operate the apparatus and analyse the produced results.
- *Fit Criterion*: The product shall be approved as Windows compliant by testers and developers.

3.3 Off-the-Shelf Software

N/A

3.4 Anticipated Workplace Environment

MC-3. The product shall be used in the Chemical Engineering Lab run by Dr. Charles de Lannoy, Bassel Abdelkader and their team of researchers.

3.5 Partner or Collaborative Applications

N/A

3.6 Schedule Constraints

MC-4. The project must be finished within the course of the current academic year.

- *Rationale*: The finished product, as outlined in the project requirements, must be submitted by the end of the academic year. A few relevant deadlines include:
  - Proof of Concept Demonstration: November 11 to 22, 2024
  - Revision 0 Demonstration: February 3 to 14, 2025
  - Final Demonstration (Revision 1): March 24 to 30, 2025

3.7 Budget Constraints

MC-5. The total cost of the project must not exceed $750.

- *Rationale*: The product must be economically feasible and all teams must have an equal budget to ensure conformity and equality in terms of access of resources.

3.8 Enterprise Constraints

N/A

4 Naming Conventions and Terminology

The following are standardized terms used throughout the project and its documentation to ensure clarity and consistency in communication.

4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders Involved in the Project

- **Alkalinity Enhancement**: A process in ocean engineering to increase the ocean's ability to absorb CO_2.
- **CO_2**: Carbon Dioxide, a greenhouse gas that contributes to global warming.

- **Ion Exchange**: The process of exchanging ions between the dilute base and seawater to increase alkalinity.
- **pH Level**: A measure of acidity or alkalinity, critical in assessing the effectiveness of the alkalinity enhancement process.
- **Buffering Capacity**: The ability of seawater to resist changes in pH, essential for maintaining stable conditions during experiments.
- **Electrodialysis**: A process that uses electric fields to drive ion movement through membranes, facilitating the generation of the dilute base.
- **Oceans' Carbon Cycle**: The natural process by which carbon is exchanged between the ocean, atmosphere, and land, impacting global climate.
- **POC (Proof of Concept)**: A demonstration used to verify that a concept is feasible.
- **V&V (Verification and Validation)**: Ensures that the software meets the required standards and performs as expected.
- **CSV (Comma-Separated Values)**: A file format used for storing tabular data, such as those from experiments.
- **Data Migration**: The process of transferring data between storage types or formats.
- **Backend**: The server side responsible for logic, data management, and API services.
- **Frontend**: The user interface built using React, interacting with the backend.
- **Database**: A structured storage system (MongoDB) used to manage project data.
- **API (Application Programming Interface)**: The means by which the frontend communicates with the backend, using GraphQL.
- **CI/CD (Continuous Integration/Continuous Deployment)**: Automating testing and deployment to ensure reliable updates.

- **Git**: Distributed version control for tracking code changes and enabling collaborative development.
- **Kanban**: A task management method used in to track progress.
- **Branch**: A separate version of the codebase where changes are developed.
- **Commit**: A recorded change to the codebase.
- **Pull Request**: A request to merge changes into the main codebase after review.
- **Deployment**: Releasing the application to a live environment.

5 Relevant Facts And Assumptions

External factors that may have an effect on the product have been recorded as relevant facts and assumptions in this section.

5.1 Relevant Facts

Currently, two sources of data input are used -

- The CSV files that contain datapoints generated by the apparatus.
- The initial parameter values for each experiment such as power voltage, age of membrane, density module and more that are inputted into template excel files. These values are manually inputted by the user and remain constant throughout each experiment.

5.2 Business Rules

N/A

5.3 Assumptions

N/A

6 The Scope of the Work

This section determines the boundaries of the project to be studied and outlines how it fits into its environment.

6.1 The Current Situation

Currently, a system of excel template files is used to store and manage the data. CSV files are generated by the lab apparatus which are then downloaded and copied onto an 'experiment' excel template. This step involves getting rid of redundant data when a single CSV file is split into multiple excel files and removing inaccurate data (such as rows of zeroes).

This data in the experiment template is then analysed and the results from the analysis of each experiment is then added to another consolidated excel template, referred to here as 'observations' that reords all experiment observations. This observations template also contains additional parameters for each experiment such as membrane, configuration and more wich are entered manually by the user.

6.2 The Context of the Work

The path highlighted with blue arrows indicates the current process and how the existing data will be migrated.
The path highlighted with green arrows indicates how the product is expected to work after installation.

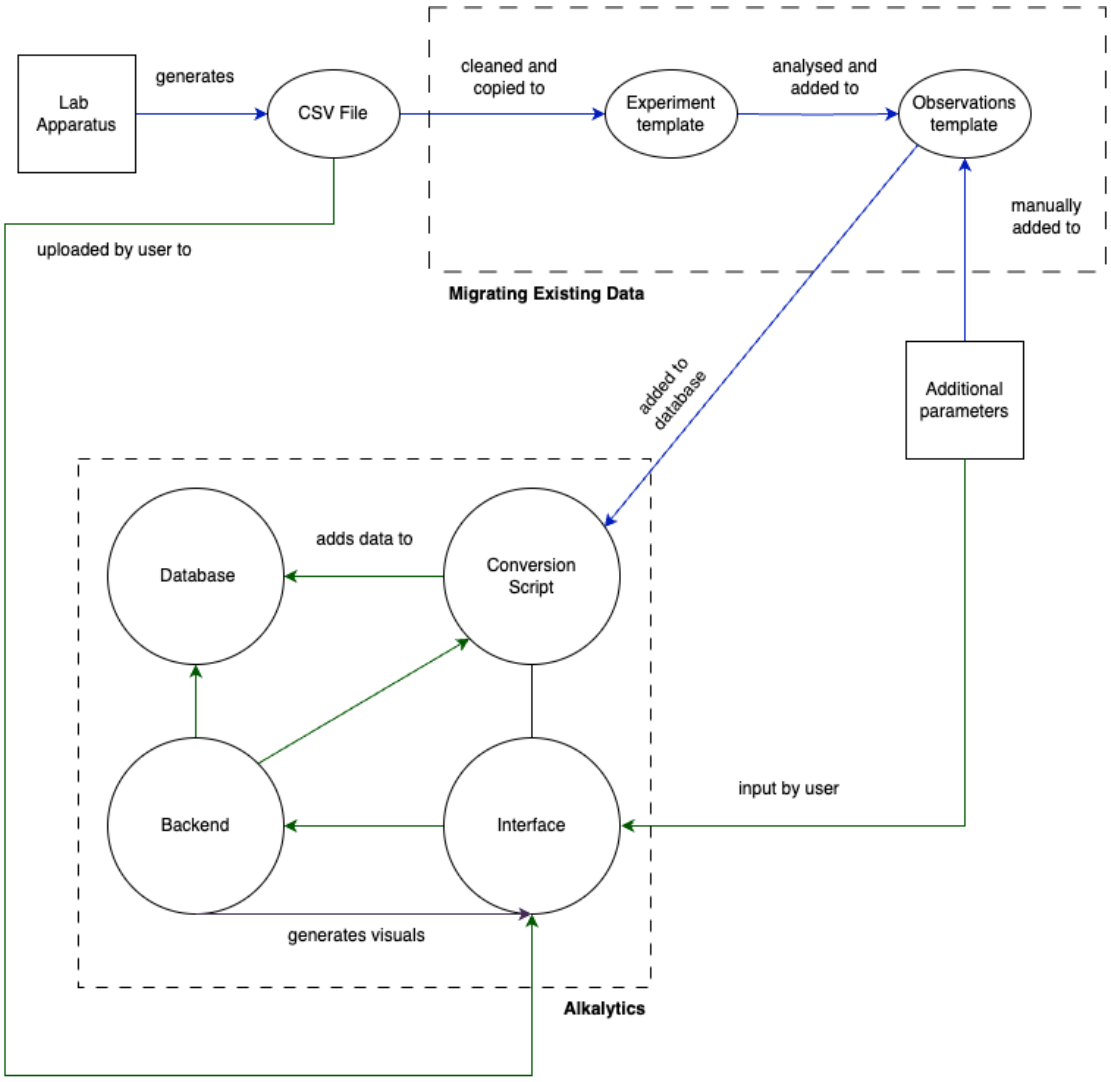

Figure 1: Work Context Model

7 Business Data Model and Data Dictionary

7.1 Business Data Model

The diagram represents a business data model that encapsulates all the data that is created, referenced, updated and deleted by processes in the platform.

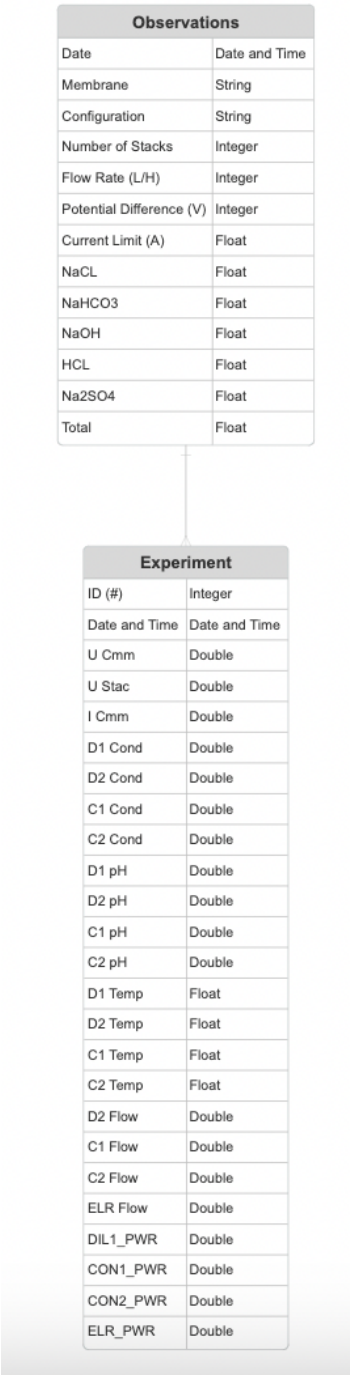

Figure 2: Business Data Model

7.2 Data Dictionary

N/A

8 The Scope of the Product

The following section will highlight use cases of the product and how users and stakeholders will interact with the product.

8.1 Product Boundary

Below is a use case diagram involving the two clients and major customers. The diagram includes the back end database as an actor as it interacts with the interface system. Level 1 to 3 labeled actors are part of the research time customer mentioned in section 2.2 where depending how the team is structured, there are differnt use cases for different level actors. These are added for clarity of the system.

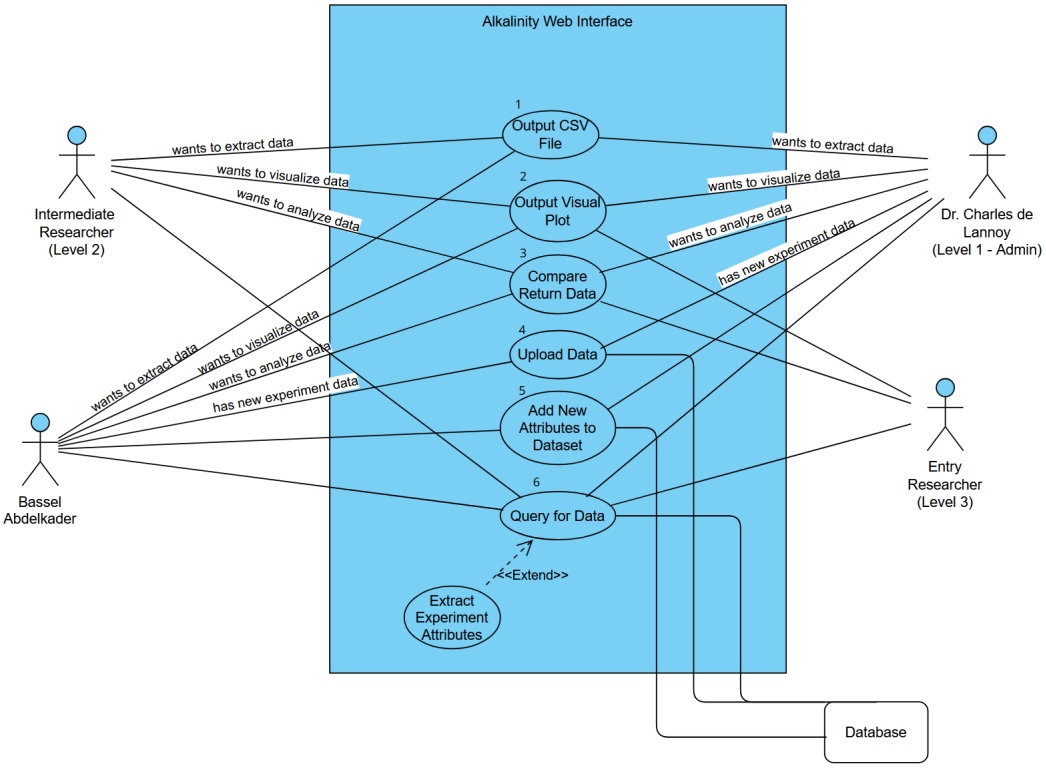

Figure 3: Use Case Diagram

8.2 Product Use Case Table

The product use case table (PUC) is an extention of the use case diagram in the pervious section. The table aims to provide more details.

PUC No.	PUC Name	Actor(s)	Input & Output
1	Output CSV	Dr. Charles de Lannoy Bassel Abdelkader Intermediate Researcher	Downloadable CSV file with appropriate data (out) Success or failure message (out)
2	Output Visual Plot	Dr. Charles de Lannoy Bassel Abdelkader Intermediate Researcher Entry Researcher	downloadable portable network graphic (PNG) or portable document format (PDF) file (out) Online viewable plot graphic (out)
3	Compare Return Data	Dr. Charles de Lannoy Bassel Abdelkader Intermediate Researcher Entry Researcher	Online viewable table of result (out) Data analysis results of involved data(out)
4	Upload Data	Dr. Charles de Lannoy Bassel Abdelkader Database	CSV file in expected format and attributes (in) Success or failure user message (out) Update database with inputted data (out)
5	Add New Attributes to Dataset	Dr. Charles de Lannoy Bassel Abdelkader Database	Update database with the new attribute (out) Success or failure user message (out) Attribute name of string type (in)
6	Query for Data	Dr. Charles de Lannoy Bassel Abdelkader Intermediate Researcher Entry Researcher Database	Visual online table with the queried data (out) Selected attribute(s), dates, restrictions (in)

Table 1: Product Use Cases

8.3 Individual Product Use Cases (PUC's)

8.3.1 Output CSV

- Description: The user can export/output the data that they have received from any function on the website system
- Pre-condition: There must be some data in the database. The information the user wants to output must also be able to be stored in a CSV file
- Post-condition: The user will get to download the desired CSV file containing data from the database. If the download file has been sent to the system, the user will receive a success or failure message.
- Basic path: After querying the data from the database is successful, the user can click on the download CSV button to export the file onto their system.

8.3.2 Output Visual Plot

- Description: The user is able to generate a graphical plot based on data from the database. The contents of the graph can be customized to include any sort of data that the user decides from the data base. The graph type can be determined by the user.
- Pre-condition: There must be some selected data to be used to plot a graph.
- Post-condition: A online view of the plot will be generated and displayed on the website. There will also be an option for downloadable version.
- Basic path: After querying the database for desired data, the user can pick the option to plot the returned data. The website will then generate a graphical plot based on the returned data.

8.3.3 Compare Return Data

- Description: This use case will allow the user to compare two sets of data such as differences or similarities among other criteria.

- Pre-condition: There must be two sets of data and what attribute(s) are being compared.
- Post-condition: The analysis of the data will be presented in a visual manner. Possible outcome may include colour coded elements or a small statement.
- Basic path: After querying the database for desired data, the user can choose to enable the compare function. The user will then need to pick what attributes they would like to compare and then continue with the website prompts which will eventually give the data analysis result.

8.3.4 Upload Data

- Description: When the user has more experimental data and would like to update the database, this function of the website will be used to take in the data and update/add to the existing database.
- Pre-condition: The file with the new data will be a CSV file and will need to be uploaded through the website interface.
- Post-condition: A success or failure message will be displayed to the user once the file is done uploading and have been added to the database.
- Basic path: The user will need to click on the upload data button and find the file they wish to upload on their system and upload it through the website's drop box.

8.3.5 Add New Attributes to Dataset

- Description: As the experiment and data become larger or changes have been made, there will be times when the dataset will need to be changed by adding new attributes.
- Pre-condition: The user must specify which dataset that they want to edit. The user must then specify what is the attribute name is in the form of a string type input.
- Post-condition: A success or failure message will be displayed to the user once the new attribute has been added to the database.

- Basic path: The user will need to click on the edit dataset button. The user must then fill in the input fields with the correct information

8.3.6 Query for Data

- Description: The user will want to get data from their past experiments. To do so, they will need to decide what they want. This is one of the main use case for any user/actor.
- Pre-condition: The database must have the some data. The user's query request must be filled in through the input fields. The fields can range from dates, to strings, to constraints.
- Post-condition: The returned data will be shown to the user through a visual table representation and will be presented with a range of other functions that the user can do with the returned data.
- Basic path: The user will need to click on the query function button. They will then need to fill in the input fields, go through the user prompts to then get the output.

9 Functional Requirements

This section outlines the key functionalities the application must satisfy with justifications and fit criteria, as well as proposed phase-in dates.

9.1 Data Input Requirements

FR-1. The application shall allow the user to input new experiment data or parameters.

- *Rationale*: The application needs to be kept up-to-date with ongoing experiments, which may include new parameters that did not exist previously.
- *Fit Criterion*: The user should be able to input new data and parameters without any errors.
- *Priority/Phase-in date*: Medium/Dec 6; This is dependent on having FR-2 implemented and can be developed later on.

- *Traceability*: FR-2, PR-1, PR-14, SR-3.

FR-2. The application shall store experiment data in the database with all associated parameters and values correctly labelled.

- *Rationale*: Ensures that data retrieval and analysis will be correct and accurate.
- *Fit Criterion*: The application's database parameters and values shall match the original experiment data parameters and values.
- *Priority/Phase-in date*: High/Nov 10; This is part of the functionalities that must be implemented for the proof of concept demonstration.
- *Traceability*: UHR-3, FR-1, FR-4, FR-5, FR-6, FR-8, FR-10, FR-11, PR-1, PR-6, PR-7, PR-12, PR-13, SR-3.

9.2 Data Migration and Organization Requirements

FR-3. The application shall read existing experiment data stored in .CSV files.

- *Rationale*: Existing experiment data is stored in Excel spreadsheets and must be integrated into the new application for continuity and analysis.
- *Fit Criterion*: The application shall read and import the data files without any errors.
- *Priority/Phase-in date*: High/Nov 10; This is part of the functionalities that must be implemented for the proof of concept demonstration.
- *Traceability*: PR-12, SR-4, SR-5, SR-6.

FR-4. The application shall organize experiment data by timestamps and experiment ID for unique identification.

- *Rationale*: Each experiment needs to be separately identified for quick retrieval of data and efficiency in search or query actions.
- *Fit Criterion*: Each ID and timestamp shall be traceable to one experiment.

- *Priority/Phase-in date*: High/Nov 10; This is part of the functionalities that must be implemented for the proof of concept demonstration.
- *Traceability*: FR-2, PR-7, FR-5, FR-6, FR-8, MSR-3, SR-3, SR-4, SR-5, SR-6.

9.3 Data Search and Query Requirements

FR-5. The application shall allow the user to search for specific datasets based on different parameters.

- *Rationale*: The user must be able to do quick look-ups of certain experiments and their results.
- *Fit Criterion*: The application shall retrieve the correct experiments based on the matching parameters.
- *Priority/Phase-in date*: High/Nov 10; This is part of the functionalities that must be implemented for the proof of concept demonstration.
- *Traceability*: FR-2, FR-4, PR-6, FR-7, FR-10, PR-2.

FR-6. The application shall allow the user to query two or more parameters or datasets for comparison and analysis.

- *Rationale*: Allows for direct comparisons between different experiment parameters and/or results, which is necessary for data analysis.
- *Fit Criterion*: The application shall retrieve the correct parameters and/or experiments based on the query inputs.
- *Priority/Phase-in date*: High/Nov 10; This is one of the first functionalities that must be implemented for the proof of concept demonstration.
- *Traceability*: FR-2, FR-4, PR-6, FR-7, FR-10, FR-11, FR-15, PR-2.

FR-7. The application shall display the results of a user's selected search or query in a format that is readable to the user.

- *Rationale*: The user needs to see the results in a format that they can interpret.

- *Fit Criterion*: The results shall be displayed in a table with all labels correct and legible.
- *Priority/Phase-in date*: Medium/Jan 17; Its implementation depends on the successful implementation of several other requirements, thus will be properly implemented at a later time.
- *Traceability*: FR-5, FR-6, PR-3, PR-10.

9.4 Data Visualization Requirements

FR-8. The application shall generate visual graphs based on selected parameters and datasets.

- *Rationale*: Visual representation of the data allows for easy interpretation and graphical analysis.
- *Fit Criterion*: The result should display a graphical plot with a title, axes, labels, and a legend.
- *Priority/Phase-in date*: High/Jan 17; Although this is a core functionality for the application, its implementation depends on the successful implementation of several other requirements, thus will be properly implemented at a later time.
- *Traceability*: FR-2, FR-4, FR-9, LFR-4, PR-5, PR-8, PR-10.

FR-9. The application shall allow the user to customize the data visualization by adjusting axes, data ranges, labels, etc.

- *Rationale*: Allows the user to adjust the graphical representation to their needs for their analysis.
- *Fit Criterion*: Modifications to axes, data ranges, labels should be reflected in the generated graph in real-time.
- *Priority/Phase-in date*: Low/March 21; This depends on the successful implementation of FR-8 and thus will be implemented at a later time.
- *Traceability*: FR-8.

9.5 Data Analysis Requirements

FR-10. The application shall analyze patterns and trends in the experiment data based on the user's selected parameters.

- *Rationale*: Trend analysis is critical for the user to discover important findings pertaining to the experiment.
- *Fit Criterion*: The application shall generate a result of the analysis to display to the user.
- *Priority/Phase-in date*: Medium/Jan 17; Although this is a core functionality for the application, its implementation depends on the successful implementation of many other requirements and thus will be implemented at a later time.
- *Traceability*: FR-2, FR-5, FR-6, PR-6, FR-11.

FR-11. The application shall use machine learning algorithms to predict and interpolate the data.

- *Rationale*: Allows for future predictions of data and efficiency when designing future experiments.
- *Fit Criterion*: The application shall generate a report of value predictions or interpolate a graph and provide the interpolated data points.
- *Priority/Phase-in date*: Medium/Jan 17; Although this is a core functionality for the application, its implementation depends on the successful implementation of many other requirements and thus will be implemented at a later time.
- *Traceability*: FR-2, FR-6, FR-10, PR-6.

9.6 Error Tracking Requirements

This section outlines functional requirements for one of the project's stretch goals.

FR-12. The application shall track and log errors in the experiment data.

- *Rationale*: Users need to identify irrelevant or missing parameters.

- *Fit Criterion*: Missing values in the input data should be flagged.
- *Priority/Phase-in date*: Low/March 21; As this is part of the stretch goals for the application, its implementation will not be considered until all higher priority requirements are satisfied.
- *Traceability*: FR-13, MSR-3, MSR-5, SR-9.

FR-13. The application shall remove data logged as errors.

- *Rationale*: Ensures data is organized and produces accurate results in analysis.
- *Fit Criterion*: Flagged data should be removed from the database by the algorithm.
- *Priority/Phase-in date*: Low/March 21; As this is part of the stretch goals for the application, its implementation will not be considered until all higher priority requirements are satisfied.
- *Traceability*: FR-12, SR-9.

9.7 User Access Management Requirements

This section outlines functional requirements for one of the project's stretch goals.

FR-14. The application shall allow the user to sign in with valid credentials.

- *Rationale*: Ensures the data can only be accessed and modified by authorized users.
- *Fit Criterion*: The user shall be able to log in with a username and password.
- *Priority/Phase-in date*: Low/March 21; As this is part of the stretch goals for the application, its implementation will not be considered until all higher priority requirements are satisfied.
- *Traceability*: PR-11, SR-1, SR-2, SR-8.

9.8 Data Export Requirements

This section outlines functional requirements for one of the project's stretch goals.

FR-15. The application shall generate a report of queries in a session for the user to save or download.

- *Rationale*: Allows user to keep a record of their findings for future use or reference.
- *Fit Criterion*: The report should be exported in CSV or PDF format.
- *Priority/Phase-in date*: Low/March 21; As this is part of the stretch goals for the application, its implementation will not be considered until the higher priority requirements are satisfied.
- *Traceability*: FR-6.

10 Look and Feel Requirements

This section will highlight the look and feel of the web interface for the project involving the appearance and the style of the user interface and experience.

10.1 Appearance Requirements

LFR-1. The website should have a simple and organized layout, with clearly defined sections where all major functions should be easily accessible and viewable.

- *Rationale:* Having a simple organized layout will ensure that the navigation of the website is quick and intuitive for accessing features and functions, which will enhance the user experience.
- *Fit Criterion:* A user should be able to identify all the major functions of the website within five minutes of use.

LFR-2. The website shall be responsive on all computer and laptop screens aside from mobile screens.

- *Rationale:* Having a responsive website will ensure that the application accommodates the majority, if not all, of the user base in having a proper user experience.
- *Fit Criterion:* The usability of the website should be the same as the default view on larger and smaller computer, laptop, and monitor screens.

LFR-3. The website's functions and buttons shall be properly labeled so that no button is ambiguous to users.

- *Rationale:* Limiting ambiguity will ensure that users understand and recognize the functions to minimize confusion and improve efficiency.
- *Fit Criterion:* A user should be able to tell what all buttons inherently do without needing to ask questions.

LFR-4. The produced plot from the data shall be properly labeled.

- *Rationale:* Properly labeling plots will help users accurately interpret the data to make important analytical understandings.
- *Fit Criterion:* The plots should not be ambiguous; users should be able to understand what the plot is about within five minutes of viewing it.

10.2 Style Requirements

LFR-5. All icons on the website must be in the design standard.

- *Rationale:* To enforce an identity and unity for the website.
- *Fit Criterion:* After a user's first encounter with the product, 90% of users should see that there is unity among all the icons on the website.

LFR-6. All colors must match the theme of the website.

- *Rationale:* Applying a theme will ensure users have an engaging visual experience.

- *Fit Criterion:* After a user's first encounter with the product, 80% of users should agree that there is a common theme throughout the website.

LFR-7. All fonts are to be consistent throughout the website.

- *Rationale:* Consistent fonts will increase readability, ensuring users can focus on the functionalities that matter on the page rather than inconsistencies.
- *Fit Criterion:* After a user's first encounter with the product, there should be no user who feels that any fonts do not belong on the website.

11 Usability and Humanity Requirements

This section outlines the requirements that are intended to make the product usable and ergonomically acceptable to its users.

11.1 Ease of Use Requirements

UHR-1. The product must be easy to navigate and use for 90% of individuals with basic computer literacy.

- *Rationale*: The product must be user-friendly. In the context of this project, basic computer literacy is defined to encompass five computer skills - using a keyboard to type, using a mouse to navigate, understanding basic software applications such as word processing and spreadsheets, browsing the internet, and managaing files and folders.
- *Fit Criterion*: An individual with basic computer literacy, such as John Doe as described in Personas must be able to launch the application and upload an input file without any assistance from the administrator.

11.2 Personalization and Internationalization Requirements

UHR-2. The current version of the product will only be available in English (EN-US) and more languages can be added in the later versions.

- *Rationale*: Currently, the product is only expected to be used by McMaster faculty and staff who are fluent in English.

UHR-3. The product must recognize commonly used scientific and mathematical symbols.

- *Rationale*: The product shall be used to store scientific parameters as datapoints so the product must be able to recognize commonly used symbols used to specify scientific properties.
- *Fit Criterion*: The product must be able to recognize the uppercase and lowercase Greek Alphabet.

11.3 Learning Requirements

UHR-4. 85% of users must be able to use the product without any formal training and with minimal guidance.

- *Rationale*: The product shall be intuitive to use. Users must be able to freely naviagte and experiment with the product after a simple product walkthrough.
- *Fit Criterion*: A new user with basic computer literacy skills should be able to upload an input file, enter initial experiment parameters, select fields to be compared and view their graph after a simple product walkthrough by the administrator.

11.4 Understandability and Politeness Requirements

UHR-5. The product shall maintain consistent use of terminology across the platform to avoid confusion.

- *Rationale*: As an example, if the term "parameters" has been used in one part of the interface, it will not be used interchangeably with "variables" in a different part of the interface.
- *Fit Criterion*: A thorough visual inspection conducted by testers should not reveal any discrepancies.

11.5 Accessibility Requirements

UHR-6. The web interface of the product must be fully compliant with the Web Content Accessibility Guidelines (WCAG) 2.1 standards at Level AA.

- *Rationale*: WCAG 2.1, Level AA compliance is an internationally accepted, common standard for making web content more accessible.
- *Fit Criterion*: Compliance with the standard will be tested by testers and measured using the Level AA compliance checklist.

12 Performance Requirements

Performance requirements define the measurable standards the application must meet regarding speed, accuracy, robustness, and resource usage to ensure optimal performance of the application's functions.

12.1 Speed and Latency Requirements

PR-1. The application shall store new data or parameters within 60 seconds of input.

- *Rationale*: The speed at which the task can be completed depends on the size of the new data to be inputted or how the addition of a new parameter will affect the existing database structure. The quantified timeframe allows for adequate processing time that aligns with user expectations.
- *Traceability*: FR-1, FR-2.

PR-2. The application shall take a maximum of 3 seconds to retrieve data from the database for search and query requests.

- *Rationale*: The speed at which the task can be completed varies based on internet connectivity and the complexity of the query being executed.
- *Traceability*: FR-5, FR-6.

PR-3. The interaction between the interface and the user shall have an average response time of 1 second under stable internet connection.

- *Rationale*: Ensures a fluid interaction between the user and the application.
- *Traceability*: FR-7.

PR-4. The application shall have a maximum latency of 2 seconds for search and query actions.

- *Rationale*: Ensures that the delay between a user's request and the application's response is minimized to provide the user with quick feedback and a smoother user experience.

PR-5. The application shall generate visualizations of data in an average time of 10 seconds, depending on the size of the dataset being processed.

- *Rationale*: The time required to render a graph will be affected by the number of datapoints included in the plot. Larger datasets will take longer to process and display.
- *Traceability*: FR-8.
- *Fit Criteria for PR-1 to PR-5*: The application shall satisfy the requirements above.

12.2 Safety-Critical Requirements

N/A; the application does not have safety-critical requirements to consider.

12.3 Precision or Accuracy Requirements

PR-6. All parameter values shall be accurate to four decimal places.

- *Traceability*: FR-2, FR-5, FR-6, FR-10, FR-11.

PR-7. All timestamps of experiment data shall be accurate to milliseconds.

- *Traceability*: FR-2, FR-4.

PR-8. Values on visual data plots shall be accurate to four decimal places.

- *Traceability*: FR-8.
- *Rationales for PR-6 to PR-8*: Accuracy of the data is critical for data analysis, prediction, and interpolation.
- *Fit Criteria for PR-6 to PR-8*: The application shall satisfy the requirements above.

12.4 Robustness or Fault-Tolerance Requirements

PR-9. The application shall not crash but display an error message if it loses connection to the backend server.

PR-10. The application shall not terminate but allow the user to view previously loaded queries and generated plots if it loses connection to the internet.

- *Rationales for PR-9 and PR-10*: The application should not fail or crash when experiencing unexpected circumstances.
- *Traceability*: FR-7, FR-8.

12.5 Capacity Requirements

PR-11. The application shall allow for up to three simultaneous users.

- *Rationale*: The number of anticipated end-users is currently three.
- *Traceability*: FR-14.

PR-12. The application shall be capable of storing data for up to 300 experiments.

- *Rationale*: The application must be capable of storing and processing large amounts of data.

- *Traceability*: FR-2, FR-3, PR-13.

- *Fit Criteria for PR-11 and PR-12*: The application shall satisfy the requirements above.

12.6 Scalability or Extensibility Requirements

PR-13. The application shall be able to process and store the existing data. The amount of data going into the application is expected to grow until the experiment study comes to an end.

- *Traceability*: FR-2, PR-12, PR-15.

PR-14. The application shall be able to add additional parameters that did not previously exist in the database at the discretion of the user.

- *Traceability*: FR-1, PR-15.

- *Rationales for PR-13 and PR-14*: The application must be able to expand to keep up with future experiments.

12.7 Longevity Requirements

PR-15. The application shall operate for the duration of the experiment study.

- *Traceability*: PR-13, PR-14.

13 Operational and Environmental Requirements

Operational requirements outline the conditions in which the application is expected to operate, which includes system compatibility and distribution requirements.

13.1 Expected Physical Environment

OER-1. The application shall operate in a typical lab environment with reliable internet connectivity.

- *Rationale*: Ensures functionality in environments where end-users are most likely to use the application.
- *Traceability*: MC-1

13.2 Wider Environment Requirements

OER-2. The application shall be compatible with desktop environments running Windows.

- *Rationale*: Ensures functionality in the technological environment the product is expected to run on.
- *Fit Criterion*: Testing will be conducted on the Windows operating system.
- *Traceability*: MC-2

13.3 Requirements for Interfacing with Adjacent Systems

OER-3. The application shall operate on the most recent versions of Chromium-based web browsers.

- *Rationale*: The application should be compatible with Chromium-based web browsers to ensure broad availability and smooth operation on widely-used platforms by the end-users.
- *Fit Criterion*: Performance testing shall be done to ensure the application functions correctly.

13.4 Productization Requirements

OER-4. The application shall be distributed as a web application.

- *Rationale*: Ensures easy access and usability for the end-users who do not have enough experience in database management.

OER-5. The application shall have an easy onboarding process with user documentation.

- *Rationale*: Ensures that users can use the application without needing frequent external support.
- *Fit Criterion*: Usability testing shall be done to ensure users are able to onboard easily.

13.5 Release Requirements

OER-6. The first version of the application shall be released after project completion.

- *Traceability*: MC-4.

14 Maintainability and Support Requirements

Maintenance requirements encompass the strategies and processes needed to ensure a system remains functional, efficient, and up-to-date throughout its lifecycle.

14.1 Maintenance Requirements

MSR-1. The application's maintenance must be the responsibility of the development team with no involvement from the users.

- *Rationale:* Ensures that skilled personnel handle maintenance.

MSR-2. Documentation must be provided to be referenced for future maintenance and to enable seamless knowledge transfer to a new team.

- *Rationale:* Ensures smooth onboarding and continuity in development.
- *Fit Criterion:* The documentation must be updated with every major release.

- *Traceability:* MSR-3.

MSR-3. The application must be designed to accommodate future development, including the addition of new experimental parameters or features without backwards progression.

- *Rationale:* Ensures the application can scale and evolve without compromising existing features.
- *Traceability:* FR-4, FR-12, MSR-2.

14.2 Supportability Requirements

MSR-4. The application must have an intuitive user interface that allows users to operate it independently without requiring external assistance.

- *Rationale:* Ensures a user-friendly experience that reduces the need for help desk support.
- *Fit Criterion:* At least 90% of users should be able to complete tasks without needing support.
- *Traceability:* LF-1, LF-2, LF-3, UHR-1, UHR-4.

MSR-5. The application must have automated guidance, such as error messages, to assist users in troubleshooting common issues.

- *Rationale:* Ensures users can resolve issues on their own, reducing the volume of support requests.
- *Fit Criterion:* The documentation must be updated with every major release and reviewed quarterly to ensure accuracy.
- *Traceability:* FR-12.

14.3 Adaptability Requirements

MSR-6. The application must be compatible with modern web browsers to ensure widespread accessibility.

- *Rationale:* Ensures the application is accessible to a broad range of users and devices.
- *Fit Criterion:* The application should at least be able to run on the latest version of Chromium-based web browsers.
- *Traceability:* LF-1, LF-2.

15 Security Requirements

Security requirements focus on protecting data, controlling access, ensuring integrity, and auditing user actions within the application.

15.1 Access Requirements

SR-1. Access to the application must be restricted to authorized personnel, with an authentication mechanism.

- *Rationale:* Ensures that only authorized users can interact with the application.
- *Fit Criterion:* Only users with valid credentials should access the application.
- *Traceability:* FR-14, SR-2, SR-4, SR-8.

SR-2. Only authenticated users should have the ability to query or modify the data, and each user's access must be limited to their capabilities within the application.

- *Rationale:* Ensures users can only perform actions that align with their roles.
- *Fit Criterion:* The application should restrict 100% of actions that are not permitted to the user's level of access.

- *Traceability:* FR-14, SR-1, SR-3, SR-8.

SR-3. The system must restrict sensitive operations (e.g., data export) to authorized personnel only.

- *Rationale:* Prevents unauthorized users from exporting or sharing sensitive data, protecting data integrity.
- *Fit Criterion:* Only authorized users must be able to perform sensitive operations like data export.
- *Traceability:* FR-15, SR-2.

SR-4. The system must enforce session timeout and automatic logouts after a period of inactivity.

- *Rationale:* Protects the system from unauthorized access if users leave their session unattended.
- *Fit Criterion:* Sessions must time out and log users off automatically after a specified inactivity period.
- *Traceability:* FR-14, SR-1, SR-4.

15.2 Integrity Requirements

SR-5. The application must validate data inputs to ensure they conform to expected formats and values before they are processed.

- *Rationale:* Ensures only valid data is processed, reducing errors.
- *Fit Criterion:* 100% of inputs must pass validation checks before processing.
- *Traceability:* FR-1, FR-2, FR-4.

SR-6. The application must not modify the data unnecessarily through its transfer process.

- *Rationale:* Ensures the original data remains accurate and unaltered.

- *Fit Criterion:* Data should remain unchanged unless explicitly modified, with logs confirming its integrity.
- *Traceability:* FR-3, FR-4, SR-7, SR-8.

SR-7. The application must ensure that any data processed or transferred is free from duplication or inconsistencies.

- *Rationale:* Ensures data consistency and prevents corruption.
- *Fit Criterion:* The application must detect and prevent 100% of duplicated records.
- *Traceability:* FR-3, FR-4, SR-6, SR-8.

SR-8. The application must have safeguards in place to maintain the accuracy of the transferred data.

- *Rationale:* Ensures reliable data transfer without loss or error.
- *Fit Criterion:* Transfer operations should maintain 100% data accuracy, verified by validation tests.
- *Traceability:* FR-3, FR-4, SR-6, SR-7.

SR-9. The system must validate CSV data thoroughly before upload to prevent corrupted or incomplete data entries.

- *Rationale:* Ensures that only valid, complete, and accurate data enters the system to prevent faulty analysis.
- *Fit Criterion:* The application shall reject any data that does not meet the validation criteria.
- *Traceability:* FR-3, FR-4.

15.3 Privacy Requirements

SR-10. All personal information related to experimental participants or stakeholders, if applicable, must be anonymized and handled in accordance with relevant privacy laws and regulations.

- *Rationale:* Ensures user privacy and legal compliance.
- *Traceability:* SR-11.

SR-11. The application must restrict data sharing with external parties unless expressly authorized by stakeholders, and users must be fully informed about the privacy policies.

- *Rationale:* Ensures transparency and control over data sharing.
- *Traceability:* FR-14, SR-1, SR-2, SR-10.

SR-12. The system must monitor database storage capacity and alert administrators when thresholds are reached to prevent system crashes.

- *Rationale:* Ensures the system continues operating smoothly by addressing storage limits proactively.
- *Fit Criterion:* The system shall send alerts when storage capacity exceeds 80% usage.
- *Traceability:* PR-12, PR-13.

15.4 Audit Requirements

SR-13. The application must maintain a comprehensive audit trail, logging all access and modification events, including timestamps and identities of users performing actions.

- *Rationale:* Ensures accountability and traceability of actions.
- *Fit Criterion:* 100% of data access and modification events must be logged and retrievable.
- *Traceability:* FR-12, FR-13.

SR-14. Audit logs must be securely stored and accessible only by authorized personnel.

- *Rationale:* Ensures the security and integrity of audit data.
- *Fit Criterion:* Logs must be encrypted and accessible only to users with administrative privileges.
- *Traceability:* FR-14, SR-1.

SR-15. The system must provide real-time error logging and display error messages to users to enhance troubleshooting.

- *Rationale:* Ensures users are informed about system issues and can take corrective action promptly.
- *Fit Criterion:* All errors must be logged and displayed clearly to users in real-time.
- *Traceability:* FR-12, MSR-5.

15.5 Immunity Requirements

SR-16. The application must have proactive measures to detect and mitigate suspicious activities, such as repeated unauthorized access attempts, ensuring the application remains secure at all times.

- *Rationale:* Ensures early detection and prevention of security breaches.
- *Fit Criterion:* The application must detect and block unauthorized attempts after three failed login attempts, with automated alerts sent to administrators.
- *Traceability:* FR-14.

SR-17. Real-time monitoring and optimization of system resources must be implemented to avoid crashes due to resource overload.

- *Rationale:* Prevents system downtime by ensuring efficient use of CPU and memory.
- *Fit Criterion:* The system must manage memory and CPU usage dynamically to avoid overloads.
- *Traceability:* PR-4, PR-9.

16 Cultural Requirements

CR-1. The product must not be biased towards certain geographic regions or countries and their societal norms.

- *Rationale*: In scientific fields, different countries use distinct systems of units with the two most common systems being the metric system and the imperial system. The product must support multiple systems of units and allow users to switch between them easily.

17 Compliance Requirements

N/A as no relevant standard compliance and legal requirements could be identified.

18 Open Issues

N/A

19 Off-the-Shelf Solutions

Off-the-shelf solutions are evaluated to explore whether existing products can meet the project's needs or if custom development is required. By comparing available options, the team can determine which aspects of these products satisfy the project's requirements while also identifying their limitations.

19.1 Ready-Made Products

Products that could meet most or all of the project's requirements without significant customization:

- **Microsoft Power BI**: A business analytics tool capable of handling large datasets, offering data import from CSV files and advanced visualizations.
  - **Key Features:**
    - Supports CSV imports and data querying.

    - Scalable data handling.
    - Generates dynamic visualizations.
  - **Limitations:**
    - High-cost licensing for continuous use.
    - Real-time CSV data updates might be challenging to implement.
    - Lacks flexibility for scientific applications, especially niche inter-parameter comparability.

- **Tableau**: A data visualization tool known for creating interactive dashboards from large datasets.
  - **Key Features:**
    - Powerful querying.
    - Generates dynamic visualizations.
    - Interactive dashboards with advanced analytics.
  - **Limitations:**
    - High-cost licensing for continuous use.
    - Lacks support for scientific-specific customization.
    - Not designed for handling algorithmic data comparisons or extending to new experimental parameters.

19.2 Reusable Components

Products that could be used by the project.

- **D3.js**: A JavaScript library for creating dynamic, interactive visualizations.
  - **Key Features:**
    - Highly customizable for complex data visualizations.
    - Integrates with web technologies (e.g., React).

- **Plotly.js**: A JavaScript library for building interactive plots and graphs in web applications.

- **Key Features:**
  - Customizable graphs and charts for web interfaces.
  - Supports real-time updates.

19.3 Products That Can Be Copied

N/A; no product exists that fits all the requirements out-of-the-box.

20 New Problems

This section outlines potential problems the new application may introduce into the environment in which it will be deployed.

20.1 Effects on the Current Environment

The application is expected to be compatible with the current implementation environment. There should be no adverse effects that come from the new application.

20.2 Effects on the Installed Systems

N/A; There should be no conflicts with existing systems.

20.3 Potential User Problems

N/A; There are no anticipated user problems.

20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

The reliance on stable internet connection may pose limitations on product access and functionality. Additionally, the server must have sufficient resources allocated to handle the anticipated volume of data and data processing demands.

20.5 Follow-Up Problems

Scalability may become a concern if data volume grows larger than anticipated, which will require monitoring and adjustments to ensure optimal performance.

21 Tasks

The team will follow an effective task management plan that emphasizes structure, iterative progress, milestone tracking, and accountability for successful project outcomes.

21.1 Project Planning

The team will adopt an agile lifecycle approach, focusing on iterative progress and adaptability. Work will be organized into stages, milestones, and phases, with regular reviews and adjustments to ensure alignment with goals. Issues are managed via GitHub and all communications are documented to ensure accountability. Stakeholder feedback will be integrated throughout the process to ensure the solution meets evolving requirements.

In addition, project planning will include weekly team meetings, biweekly supervisor meetings. Deliverables are categorized into stages where roles are rotated among team members to share responsibility.

21.2 Planning of the Milestones

The milestones provide a structured approach to the project's documentation, planning, and demonstration activities. By the deadlines, the team is expected to complete and review the documents to ensure accuracy, compliance with project requirements, and readiness for subsequent stages.

21.3 Planning of the Development Phases

The development phases outline the progression of the project's coding and implementation efforts. Each phase is focused on achieving significant technical progress that works alongside the milestones to allow for incremental progress and continuous refinement.

Stage	Milestone	Deadline
Stage 1	Problem Statement, POC Plan, Development Plan	Sept 24
	Requirements Document Revision 0	Oct 9
Stage 2	Hazard Analysis	Oct 23
	V&V Plan Revision 0	Nov 1
Stage 3	POC Demonstration	Nov 11 - 22
Stage 4	Design Document Revision 0	Jan 15
Stage 5	Revision 0 Demonstration	Feb 3 - 14
Stage 6	V&V Report Revision 0	Mar 7
Stage 7	Final Demonstration (Revision 1)	Mar 24 - 30
	EXPO Demonstration	Apr TBD
	Final Documentation (Revision 1)	Apr 2

Table 2: Project Decomposition and Deadlines

Stage	Milestone	Deadline
Phase 1	Backend Database Development	Nov 1
	Data Migration Algorithm	Nov 8
	POC Functional Testing	Nov 10
	POC Demonstration	Nov 11
Phase 2	Backend Refinement & Bug Fixes	Dec 6
	Basic Frontend Development	Jan 3
	Frontend-Backend Integration	Jan 17
	Preliminary Testing & Bug Resolution	Jan 24
	Revision 0 Demonstration	Feb 3
Phase 3	Final Frontend Features Development	Feb 14
	Backend Optimization	Feb 21
	Full System Testing & Debugging	March 21
	Revision 1 Final Demonstration	March 24

Table 3: Project Development Phases and Deadlines

Phase 1 focuses on the Proof of Concept (POC), where the team is expected to develop and present a functional prototype that demonstrates the core features and feasibility of the project. This phase focuses on the backend, including building a database capable of querying and sorting data, and implementing an algorithm to transfer data from a CSV file into the database

format.

Phase 2 involves the Revision 0 Demonstration, where the backend is expected to be fully completed according to the project requirements. In addition, a basic frontend will be developed and integrated into a website to allow interaction with the backend functionality. It is expected that some bugs or issues will be displayed.

Phase 3 Phase 3 is the Revision 1 Final Demonstration, where the final version of the project is presented. During this phase, final frontend features will be developed to improve the user interface. The backend will be optimized for performance. Full system testing and debugging will address any outstanding issues. This phase represents the culmination of all development efforts, with a fully functioning product that meets all requirements. Ultimately, it highlights the project's readiness for production.

22 Migration to the New Product

This section includes a list of coversion activities that are necessary that are necessary for the project planning process and outlines a rough timeline for their implementation.

22.1 Requirements for Migration to the New Product

The following conversion tasks have been identified as a part of the project planning process.

22.1.1 Use of Phased Implementation

The system will not be implemented in phases and would be installed as a part of a single process after testing and approval. This would be done following Stage 7 described in table 2 and Phase 3.

22.1.2 Data Conversion

Currently, the files are stored in CSV format. In order to make them compatiable for migration to the new database, the files must be converted into the appropriate format, likely, JavaScript Object Notaion (JSON) format. This

means that a special program must be written to transport the data as the new system will only accept JSON files. The goal is, when a file is inputted to the new system (in CSV format), the program will convert it to JSON so that the data can be entered into the database.

22.1.3 Major Components

The product has been decomposed into the following major components that will be put in place based on this timeline -

- Database with migrated data and funtionality to compare parameters by the Proof of Concept Demonstration
- Interface that allows inputting data and adding new parametersby the end of December 2024
- Dashboard that allows analysis of data and visualizing graphs by the end of January 2025

22.1.4 Decomissioning the Old Product

Based on conversations with primary stakeholders, it has been determined that there is no need to run the new product in parallel with the existing product. After the database is set-up by Stage 3 as outlined in table 2, all existing data up until that point will be migrated to the new database. The old product will be used to record data from continued experiments which will be migrated to the new database at the end of each month post Stage 3 and up until Stage 7.

Since the old product is just a collection of spreadsheets, no special effort is needed to decomission them. A visual inspection and testing will be performed by testers and stakeholders each time the data is migrated to the new product to ensure its accuracy and integrity.

22.2 Data That Has to be Modified or Translated for the New System

22.2.1 Current Technology

Currently the data is retreived in the form of CSV files and copied onto existing Excel templates that are used to clean, sort and analyse the data. Although the templates are sophisticated and well-deisgned, a lot of manual work is involved to flag incorrect data (rows of zeroes for all parameters), remove redundant data (the data file is sometimes split into multiple CSV files that might contain some overlapping data) and more.

Some of the collected paramteric data is also inaccurate (such as the power voltage and density module) and so those columns of data are ignored. To compare different experiments, the user is required to switch between experiment files to view their respective graphs. This is cumbersome and as the number of experiments increase, unsustainable in the long-term.

22.2.2 Data Translation

A scirpt will be written to automate the translation of files from CSV to JSON format. This has been described in detail in the Data Conversion section.

23 Costs

N/A

24 User Documentation Requirements

User documentation will cover the more complicated features and functionalities of the website.

This documentation will include how to add new attributes to the dataset, how to request data plots and any major functionality features.

This document is intended to be a help/tutorial for the end users who will

be using the site for its functionalities. Its target is for those who have limited software knowledge. Although most functions aims to be as intuitive as possible to the user, since this is an analysis application there should be documentation explaining for those who have no idea about how research data works and how to interact with it using the website.

24.1 Training Requirements

No training is required to use the end product.

25 Waiting Room

The following requirements are beyond the sophistication of, or time allowed for, initial release of the product.

25.1 Automatically Download Data from Lab Apparatus

WR-1.: The product shall automatically download the CSV data files from the lab apparatus and after conversion, upload them to the database.

- *Rationale*: Currently, the data has to be manually exported from the machine at the end of every week to ensure that the data does not get lost or overwritten.
- *Expected Version*: Version 2

25.2 Dynamic Dashboard to Generate Data Reports

WR-2.: The product shall include a dashboard that hosts multiple visual plots that are dynamically updated based on changes in the data.

- *Rationale*: For the initial release, only basic plots that are already being used by the user will be available for viewing. Whenever data is modified, the graph must be regenerated to view the updated plot.
- *Expected Version*: Version 2

25.3 Machine Learning Analysis and Projections

WR-3.: The product shall use machine learning algorithms to automatically compare parameters and generate visual graphs based on simple text prompts.

- *Rationale*: The idea is to use artificial intelligence to suggest parametric comparisons or visual graphs that the user might not have though of. The aim is to explore analyses that might not have been considered by a user.
- *Expected Version*: Version 3

25.4 Mobile Development

WR-4.: The product shall have a mobile version.

- *Rationale*: The project can be extended to include a mobile application version of the product to increase accessibility and reachability.
- *Expected Version*: Version 4

26 Ideas for Solution

- Use a NoSQL database such as MongoDB instead of a SQL database.
- Use JavaScript to implement the interface.

Appendix A — Traceability

This section was added to the Volere template to provide a traceability table for the functional and non-functional requirements.

Requirement ID	Dependent on	Traced from
FR-1	FR-2	PR-1, PR-14, SR-3
FR-2	UHR-3	FR-1, FR-4, FR-5, FR-6, FR-8, FR-10, FR-11, PR-1, PR-6, PR-7, PR-12, PR-13, SR-3
FR-3	N/A	PR-12, SR-4, SR-5, SR-6
FR-4	FR-2, PR-7	FR-5, FR-6, FR-8, MSR-3, SR-3, SR-4, SR-5, SR-6
FR-5	FR-2, FR-4, PR-6	FR-7, FR-10, PR-2
FR-6	FR-2, FR-4, PR-6	FR-7, FR-10, FR-11, FR-15, PR-2
FR-7	FR-5, FR-6	PR-3, PR-10
FR-8	FR-2, FR-4	FR-9, LFR-4, PR-5, PR-8, PR-10
FR-9	FR-8	N/A
FR-10	FR-2, FR-5, FR-6, PR-6	FR-11
FR-11	FR-2, FR-6, FR-10, PR-6	N/A
FR-12	N/A	FR-13, MSR-3, MSR-5, SR-9
FR-13	FR-12	SR-9
FR-14	N/A	PR-11, SR-1, SR-2, SR-8, SR-10, SR-11
FR-15	FR-6	N/A
LFR-1	N/A	UHR-1, MSR-4, MSR-6
LFR-2	N/A	MSR-4, MSR-6
LFR-3	N/A	UHR-1, MSR-4
LFR-4	FR-8	N/A
LFR-5	N/A	N/A
LFR-6	N/A	N/A
LFR-7	N/A	N/A
UHR-1	LFR-1, LFR-3	MSR-4

Requirement ID	Dependent on	Traced from
UHR-2	N/A	N/A
UHR-3	N/A	FR-2
UHR-4	N/A	MSR-4
PR-1	FR-1, FR-2	N/A
PR-2	FR-5, FR-6	N/A
PR-3	FR-7	N/A
PR-4	N/A	N/A
PR-5	FR-8	N/A
PR-6	FR-2	FR-5, FR-6, FR-10, FR-11
PR-7	FR-2	FR-4
PR-8	FR-8	N/A
PR-9	N/A	N/A
PR-10	FR-7, FR-8	N/A
PR-11	FR-14	N/A
PR-12	FR-2, FR-3	PR-13
PR-13	FR-2, PR-12	PR-15
PR-14	FR-1	PR-15
PR-15	PR-13, PR-14	N/A
OER-1	N/A	MC-1
OER-2	N/A	MC-2
OER-3	N/A	N/A
OER-4	N/A	N/A
OER-5	N/A	N/A
OER-6	N/A	MC-4
MSR-1	N/A	N/A
MSR-2	N/A	MSR-3
MSR-3	FR-4, FR-12, MSR-2	N/A
MSR-4	LFR-1, LFR-2, LFR-3, UHR-1, UHR-4	N/A
MSR-5	FR-12	N/A
MSR-6	LFR-1, LFR-2	N/A
SR-1	FR-14	SR-2, SR-8, SR-10
SR-2	FR-14, SR-1	SR-8
SR-3	FR-1, FR-2, FR-4	N/A
SR-4	FR-3, FR-4	SR-5, SR-6

Requirement ID	**Dependent on**	**Traced from**
SR-5	FR-3, FR-4, SR-4, SR-6	N/A
SR-6	FR-3, FR-4, SR-4	SR-5
SR-7	SR-8	N/A
SR-8	FR-14, SR-1, SR-2	SR-7
SR-9	FR-12, FR-13	N/A
SR-10	FR-14, SR-1	N/A
SR-11	FR-14	N/A

Table 4: Traceability table of the requirements.

Appendix B — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. What went well while writing this deliverable?

   Writing this deliverable gave us greater clarity regarding the project and made us think of questions we hadn't considered before such as what our data migration plan would look like, if we need RBAC (Role Based Access Control) and what each role and security level should include and more.

   Additionally, we had already acquired significant knowledge about our supervisors' needs/expectations for the project before we began the writing process, which gave us a clearer guide on modelling our requirements and allowed us to start earlier rather than later. As a result, we were able to develop an initial draft, which they reviewed and provided feedback on, enabling us to refine our document in an iterative manner.

2. What pain points did you experience during this deliverable, and how did you resolve them?

   The major pain point for this deliverable was the length and the amount of content that had to be condensed into a single document in such a short span of time. With deadline for other courses, coordinating with supervisors and giving them enough time to review our SRS for feedback - most of the steps felt rushed. Since the Volere template is so detailed, we also ran into multiple issues where we either did not know what a section or sub-section was asking for or figuring out what to write for a non-relevant section that is included in the rubric.

   However, we were able to address our questions and concerns through a meeting that we had scheduled with our TA, on top of the informal review session. This helped us gain a better understanding of the expectations for each section we faced difficulties with.

3. How many of your requirements were inspired by speaking to you client(s) or proxies (e.g. your peers, stakeholders, potential users)?

   Most of our FRs were inspired by our client, primarily our supervisors. Since they do not have a concrete plan for how they want this project implemented and instead have an abstract idea of what they would want to see in the product, it gave us a lot more room to be creative and consider multiple scenarios for our requirements, ultimately choosing the ones that suit our project the best. Most of the NFRs had to be brainstormed by our team and then validated by the stakeholders since they hadn't given much thought to these aspects of the product.

4. Which of the courses you have taken, or are currently taking, will help your team be successful with your capstone project?

   3RA3 or Software Engineering Requirements was very helpful in writing up the SRS because we weren't completely clueless and did not have to spend a lot of time researching what an SRS is or what requirements are. This meant we could dive straight into work. Another course that has been helpful is 3DB3 or Databases mostly because after taking that course, we've had experience dealing with relational databases and were able to determine very early on how a relational database would not be the most suitable for our project and thus, pivot towards NoSQL databases.

5. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

   This question was answered individually by all the team members and their responses are as follows:

   (a) Jason Tran

To successfully complete our project, I will need to learn more about backend and data migration. As I have more professional experience with the frontend or how to connect the front end to the backend, I will now need to learn the final step of creating the backend itself. This includes how to code with Python and how to create algorithms myself.

(b) Jennifer Ye

Although I have good skills in writing and team management, the one item that I will need to acquire in this capstone is my back-end knowledge. I know how back-end applications works with aspects like APIs and databases as a general concept, but I have no real experience with implementation. With this capstone project being an end-to-end project, with a focus on back end, I believe it will be critical for me to grain that knowledge to be an impactful team member to the project.

More specifically, I will need to acquire more advanced knowledge of databases and API creation. My strength is in mobile and web development, but since that portion of the project is in the later half of the project, I do not want to wait for then to be impactful. I would like to learn and practice the knowledge throughout the term.

(c) Kate Min

I will direct my focus towards machine learning algorithms and NoSQL databases. While I have experience with PostgreSQL, I'm not as familiar with NoSQL/MongoDB to make significant contributions at this time.

Developing a strong foundation in this area would allow me to contribute meaningfully to the backend development. I don't have much experience in machine learning, but this aspect will be essential to our application for data analysis. Successfully acquiring the knowledge needed in both areas will strengthen my ability to work on different parts of the project.

(d) Sumanya Gulati
Using version control conventions and such for a large-scale project. In the past, my use of git has been limited to just storing code in a centralized repository that my team members could access. We did not do any proper documentation or reviews.

This is the first project where our team has come up with a merging strategy, naming conventions and more to keep the git history clean and to ensure conformity and proper documentation. In terms of technology, I have never worked with NoSQL databases before so I am looking forward to exploring that!

6. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?

This question was answered individually by all the team members and their responses are as follows:

(a) Jason Tran
I will utilize two approaches on my learning experience, which is to explore resources online and jump straight into the deep end. This will apply what I have learned online in a real application which, in my experience, allows me to connect the dots a lot faster. Ultimately, jumping into the deep end will help progress the project along and create an iterative learning style.

(b) Jennifer Ye
One approach I will use to learn more about API and database creation is to follow tutorial crash courses. With this method, I can have a clearer fundamental understanding on the concepts. This might include parts like how they operate, simple manipulation and customization.

Another approach which can also serve as a follow up approach will be to use them in practice and to learn as I go. This can be

in the form of a quick side project. I will be pursuing both as I deem it necessary to be able to succeed in gaining the knowledge in a valuable way. I think learning all the content/knowledge of the back-end from tutorials and cash courses might even be all I need to gain the knowledge and contribute to the project on that front.

However, with my learning style, I would do better if I also applied the things I learned. This can be done straight from the capstone project, or even having a small side project to just apply what I learned before going into the project is another possibility. In conclusion, I will be using the first method to gain the knowledge I need but will also do the second method due to my learning style.

(c) Kate Min
I don't expect the learning curve from Postgres to MongoDB to be too steep, so either an online course or doing a guided project would be ideal. I would go with the latter approach to gain the necessary knowledge/experience quickly, since we'll want to implement the database early on in the development process.

Both an online course or a small-scale project would give me a strong foundation in ML models. I'd prefer to take an online course as it would allow me to gain a deeper understanding and more experience over time. Since we'll likely focus on implementing the ML at a later time, it will be easier for me to grasp the core concepts and thus make significant contributions to its implementation.

(d) Sumanya Gulati
The two approaches to mastering git and NoSQL databases would be following a tutorial to learn how git works and basic commands; and, just practicing it for all deliverables and learning along the way.

I will be leveraging my existing knowledge of git commands and workflows and use it to learn how it can be adapted to large-scale,

long-term projects that are maintainable and scalable. As for the NoSQL database, I would choose the former strategy and follow a guided tutorial first to familiarize myself with the topic so I can actually contribute to the project.