

# Software Requirements Specification for Software Engineering: Alkalytics

Team 21, Alkalytics

Sumanya Gulati

Kate Min

Jennifer Ye

Jason Tran

October 1, 2024

# Contents

<b>1</b>	<b>Mandated Constraints</b>	<b>vi</b>
1.1	Solution Constraints . . . . .	vi
1.2	Implementation Environment of the Current System . . . . .	vi
1.3	Off-the-Shelf Software . . . . .	vi
1.4	Anticipated Workplace Environment . . . . .	vi
1.5	Partner or Collaborative Applications . . . . .	vii
1.6	Schedule Constraints . . . . .	vii
1.7	Budget Constraints . . . . .	vii
1.8	Enterprise Constraints . . . . .	vii
<b>2</b>	<b>Naming Conventions and Terminology</b>	<b>viii</b>
2.1	Glossary of All Terms, Including Acronyms, Used by Stakeholders Involved in the Project . . . . .	viii
<b>3</b>	<b>Relevant Facts And Assumptions</b>	<b>ix</b>
3.1	Relevant Facts . . . . .	ix
3.2	Business Rules . . . . .	ix
3.3	Assumptions . . . . .	x
<b>4</b>	<b>The Scope of the Work</b>	<b>x</b>
4.1	The Current Situation . . . . .	x
4.2	The Context of the Work . . . . .	x
4.3	Work Partitioning . . . . .	x
4.4	Specifying a Business Use Case (BUC) . . . . .	x
<b>5</b>	<b>Business Data Model and Data Dictionary</b>	<b>x</b>
5.1	Business Data Model . . . . .	x
5.2	Data Dictionary . . . . .	x
<b>6</b>	<b>The Scope of the Product</b>	<b>x</b>
6.1	Product Boundary . . . . .	x
6.2	Product Use Case Table . . . . .	xi
6.3	Individual Product Use Cases (PUC's) . . . . .	xi
<b>7</b>	<b>Functional Requirements</b>	<b>xi</b>
7.1	Data Input Requirements . . . . .	xi
7.2	Data Migration and Organization Requirements . . . . .	xi

7.3	Data Search and Query Requirements . . . . .	xii
7.4	Data Visualization Requirements . . . . .	xii
7.5	Data Analysis Requirements . . . . .	xiii
7.6	Error Tracking Requirements . . . . .	xiii
7.7	User Access Management Requirements . . . . .	xiv
7.8	Data Export Requirements . . . . .	xiv
<b>8</b>	<b>Look and Feel Requirements</b>	<b>xiv</b>
8.1	Appearance Requirements . . . . .	xv
8.2	Style Requirements . . . . .	xv
<b>9</b>	<b>Usability and Humanity Requirements</b>	<b>xvi</b>
9.1	Ease of Use Requirements . . . . .	xvi
9.2	Personalization and Internationalization Requirements . . . . .	xvi
9.3	Learning Requirements . . . . .	xvii
9.4	Understandability and Politeness Requirements . . . . .	xvii
9.5	Accessibility Requirements . . . . .	xvii
<b>10</b>	<b>Performance Requirements</b>	<b>xvii</b>
10.1	Speed and Latency Requirements . . . . .	xvii
10.2	Safety-Critical Requirements . . . . .	xviii
10.3	Precision or Accuracy Requirements . . . . .	xviii
10.4	Robustness or Fault-Tolerance Requirements . . . . .	xviii
10.5	Capacity Requirements . . . . .	xviii
10.6	Scalability or Extensibility Requirements . . . . .	xix
10.7	Longevity Requirements . . . . .	xix
<b>11</b>	<b>Operational and Environmental Requirements</b>	<b>xix</b>
11.1	Expected Physical Environment . . . . .	xix
11.2	Wider Environment Requirements . . . . .	xix
11.3	Requirements for Interfacing with Adjacent Systems . . . . .	xx
11.4	Productization Requirements . . . . .	xx
11.5	Release Requirements . . . . .	xx
<b>12</b>	<b>Maintainability and Support Requirements</b>	<b>xx</b>
12.1	Maintenance Requirements . . . . .	xx
12.2	Supportability Requirements . . . . .	xxi
12.3	Adaptability Requirements . . . . .	xxi

<b>13 Security Requirements</b>	<b>xxii</b>
13.1 Access Requirements . . . . .	xxii
13.2 Integrity Requirements . . . . .	xxii
13.3 Privacy Requirements . . . . .	xxiii
13.4 Audit Requirements . . . . .	xxiii
13.5 Immunity Requirements . . . . .	xxiv
<b>14 Cultural Requirements</b>	<b>xxiv</b>
14.1 Cultural Requirements . . . . .	xxiv
<b>15 Compliance Requirements</b>	<b>xxiv</b>
15.1 Legal Requirements . . . . .	xxiv
15.2 Standards Compliance Requirements . . . . .	xxiv
<b>16 Open Issues</b>	<b>xxv</b>
<b>17 Off-the-Shelf Solutions</b>	<b>xxv</b>
17.1 Ready-Made Products . . . . .	xxv
17.2 Reusable Components . . . . .	xxv
17.3 Products That Can Be Copied . . . . .	xxv
<b>18 New Problems</b>	<b>xxv</b>
18.1 Effects on the Current Environment . . . . .	xxv
18.2 Effects on the Installed Systems . . . . .	xxv
18.3 Potential User Problems . . . . .	xxv
18.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product . . . . .	xxv
18.5 Follow-Up Problems . . . . .	xxvi
<b>19 Tasks</b>	<b>xxvi</b>
19.1 Project Planning . . . . .	xxvi
19.2 Planning of the Milestones . . . . .	xxvi
19.3 Planning of the Development Phases . . . . .	xxvii
<b>20 Migration to the New Product</b>	<b>xxviii</b>
20.1 Requirements for Migration to the New Product . . . . .	xxviii
20.2 Data That Has to be Modified or Translated for the New System	xxix
<b>21 Costs</b>	<b>xxx</b>

<b>22 Look and Feel Requirements</b>	<b>xxx</b>
22.1 Appearance Requirements . . . . .	xxx
22.2 Style Requirements . . . . .	xxxii
<b>23 Waiting Room</b>	<b>xxxii</b>
23.1 Automatically Download Data from Lab Apparatus . . . . .	xxxii
23.2 Dynamic Dashboard to Generate Comparison Reports . . . . .	xxxii
23.3 Machine Learning Analysis and Projections . . . . .	xxxii
23.4 Mobile Development and Accessibility . . . . .	xxxii
<b>24 Ideas for Solution</b>	<b>xxxii</b>

## Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

git pull --rebase origin documentation/SRS

# 1 Mandated Constraints

## 1.1 Solution Constraints

**Description:** 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.

## 1.2 Implementation Environment of the Current System

**Description:** 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 machine and analyse the produced results.

**Fit Criterion:** The product shall be approved as Windows compliant by testers and developers.

## 1.3 Off-the-Shelf Software

**Description:** [MongoDB](#) - a document-oriented, NoSQL database product shall be used to store the datapoints.

**Rationale:** Using an existing, versatile and scalable solution like MongoDB that does not use SQL and is thus, non-relational, will allow greater flexibility in storing datapoints.

## 1.4 Anticipated Workplace Environment

**Description:** The product shall be used in the Chemical Engineering Lab run by Dr. Charles de Lannoy and Bassel Abdelkader.

## 1.5 Partner or Collaborative Applications

**Description:** The product shall be used in collaboration with the *name of lab software*.

**Rationale:** The *name of lab software* is used to retrieve data from the lab apparatus. The retrieved data shall be used as input for the product.

## 1.6 Schedule Constraints

**Description:** 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

## 1.7 Budget Constraints

**Description:** 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.

## 1.8 Enterprise Constraints

*N/A*



## 2 Naming Conventions and Terminology

### 2.1 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<sub>2</sub>.
- **CO<sub>2</sub>:** 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.
- **GitHub:** A platform for version control and collaboration on code.
- **Kanban:** A task management method used in GitHub Projects 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.

## 3 Relevant Facts And Assumptions

### 3.1 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. These values are manually inputted by the user and remain constant throughout each experiment.

### 3.2 Business Rules

*N/A*

### **3.3 Assumptions**

*N/A*

## **4 The Scope of the Work**

### **4.1 The Current Situation**

*Insert your content here.*

### **4.2 The Context of the Work**

*Insert your content here.*

### **4.3 Work Partitioning**

*Insert your content here.*

### **4.4 Specifying a Business Use Case (BUC)**

*Insert your content here.*

## **5 Business Data Model and Data Dictionary**

### **5.1 Business Data Model**

*Insert your content here.*

### **5.2 Data Dictionary**

*Insert your content here.*

## **6 The Scope of the Product**

### **6.1 Product Boundary**

*Insert your content here.*

## 6.2 Product Use Case Table

*Insert your content here.*

## 6.3 Individual Product Use Cases (PUC's)

*Insert your content here.*

# 7 Functional Requirements

## 7.1 Data Input Requirements

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

- *Rationale:* The system 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 with 0 errors.

**FR-2.** The system 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 system database parameters and values shall match the original experiment data parameters and values.

## 7.2 Data Migration and Organization Requirements

**FR-3.** The system 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 system for continuity and analysis.
- *Fit Criterion:* The system shall read and import the data files with 0 errors.

**FR-4.** The system 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.

### 7.3 Data Search and Query Requirements

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

- *Rationale:* Allows for quick look-ups of certain experiments and their results.
- *Fit Criterion:* The system shall retrieve the correct experiments based on the matching parameters.

**FR-6.** The system 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 analysis.
- *Fit Criterion:* The system shall retrieve the correct parameters and/or experiments based on the query inputs.

**FR-7.** The system 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.

### 7.4 Data Visualization Requirements

**FR-8.** The system 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.

**FR-9.** The system 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.

## 7.5 Data Analysis Requirements

**FR-10.** The system 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 system shall generate a result of the analysis to display to the user.

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

- *Rationale:* Allows for future predictions of data and efficiency in running future experiments.
- *Fit Criterion:* The system shall generate a report of value predictions or interpolate a graph and provide the interpolated data points.

## 7.6 Error Tracking Requirements

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

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

- *Rationale:* Helps users identify irrelevant or missing parameters.
- *Fit Criterion:* Missing values from input data should be flagged.

**FR-13.** The system shall remove data logged as errors.

- *Rationale:* Ensures data is organized and produce accurate results in analysis.
- *Fit Criterion:* Flagged data should be removed from the database after user confirmation.

## 7.7 User Access Management Requirements

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

**FR-14.** The system 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.

## 7.8 Data Export Requirements

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

**FR-15.** The system 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.

# 8 Look and Feel Requirements

All subitems in this sections are the fit criterion for its respective requirements.

## 8.1 Appearance Requirements

AR1. The website should have a simple and organized layout, with clearly defined sections where all major functions should be easily accessible and viewable

AR1.1 A users should be able to identify all the major functions of the website within five minuets of use.

AR2. The website shall be responsive on all computer and laptop screens aside from mobile screens

AR2.1 The usability of website should be the same as the default view on big and smaller computer, laptop and monitor screens

AR3. The website's functions and buttons shall be properly labeled where no button is ambiguous to users

AR3.1 A user should be able to tell what all buttons inherently does without the needing to ask questions

AR4.1 The produced plot from the data shall be properly labeled

AR4.1 The plots should not be ambiguous. Users should be able to understand what is the plot about within five minuets of viewing it.

## 8.2 Style Requirements

SR1. All icons on the website must be in the artistic style

SR1.1 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.

SR2. All colours must match the theme of the website

SR2.1 After a user's first encounter with the product, 80% of users should agree that there is a common theme throughout the website.

SR3. All fonts are to be consistent throughout the website



SR3.1 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.

## 9 Usability and Humanity Requirements

### 9.1 Ease of Use Requirements

**Description:** The product must be easy to navigate and use for 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 managing files and folders.

**Fit Criterion:** An individual with basic computer literacy must be able to launch the application and upload an input file without any assistance from the administrator.

### 9.2 Personalization and Internationalization Requirements

**Description:** 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.

**Description:** 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.

## 9.3 Learning Requirements

**Description:** 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 navigate 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.

## 9.4 Understandability and Politeness Requirements

N/A

## 9.5 Accessibility Requirements

N/A

# 10 Performance Requirements

## 10.1 Speed and Latency Requirements

1. The system shall store new data or parameters within 60 seconds of input.
  2. The system shall retrieve data from the database within 50ms for typical search and queries.
  3. The interaction between the interface and the user shall have a maximum response time of 2 seconds.
  4. The system shall have a maximum latency of 2 seconds for typical search and queries.
  5. The system shall generate a visualization of the data within 5 seconds.
- *Rationale:* Quick response times ensure efficiency and smooth user experience without disrupting the flow of the user's thought processes.

- *Fit Criterion:* The system shall satisfy the requirements above.

## 10.2 Safety-Critical Requirements

The product does not have safety-critical requirements to consider.

## 10.3 Precision or Accuracy Requirements

1. All parameter values shall be accurate to four decimal places.
  2. All timestamps of experiment data shall be accurate to milliseconds.
  3. Values on visual data plots shall be accurate to four decimal places.
- *Rationale:* Accuracy of the data is critical for data analysis, prediction, and interpolation.
  - *Fit Criterion:* The system shall satisfy the requirements above.

## 10.4 Robustness or Fault-Tolerance Requirements

1. The application shall not terminate but display an error message if it loses connection to the backend server.
  2. The application shall provide basic functionality if it loses connection to the internet.
- *Rationale:* The system should not fail or crash when experiencing unexpected circumstances.

## 10.5 Capacity Requirements

1. The application shall allow for up to three simultaneous users.
  2. The system shall store up to x amount of data.
- *Rationale:* The system must be capable of storing and processing large amounts of data.
  - *Fit Criterion:* The system shall satisfy the requirements above.

## 10.6 Scalability or Extensibility Requirements

1. The system shall be able to process and store the existing data. The amount of data going into the system is expected to grow until the experiment study comes to an end.
  2. The system shall be able to add additional parameters that did not previously exist in the database at the discretion of the user.
- *Rationale:* The system must be able to expand to keep up with future experiments.

## 10.7 Longevity Requirements

1. The system shall operate for the duration of the experiment study.

# 11 Operational and Environmental Requirements

## 11.1 Expected Physical Environment

1. The application shall operate in a typical office environment with reliable internet connectivity.
  2. The application shall be compatible with a desktop or laptop environment.
- *Rationale:* Ensures functionality in environments where end-users are most likely to use the application, accomodating several screen sizes and operating systems.
  - *Fit Criterion:* Testing will be conducted on the two most common operating systems, Windows and macOS.

## 11.2 Wider Environment Requirements

*Insert your content here.*

### 11.3 Requirements for Interfacing with Adjacent Systems

1. The application shall operate on the most recent versions of Google Chrome and Apple Safari.
  - *Rationale:* The application must be able to operate on these two most common web browsers, as these will be the primary platforms where it is hosted and accessed by users.
  - *Fit Criterion:* Performance testing shall be done to ensure the application functions correctly.

### 11.4 Productization Requirements

1. The system shall be distributed as a web application.
2. The system shall have an easy onboarding process with user documentation.
  - *Rationale:* Ensures that users can use the application without needing frequent support.
  - *Fit Criterion:* Usability testing shall be done to ensure users are able to onboard easily.

### 11.5 Release Requirements

1. The first version of the system shall be released after project completion.

## 12 Maintainability and Support Requirements

### 12.1 Maintenance Requirements

- The application's maintenance must be the responsibility of the development team with no involvement from the users.
  - *Rationale:* This ensures that skilled personnel handle maintenance.

- Documentation must be provided to be referenced for future maintenance and to enable seamless knowledge transfer to new developers.
  - *Rationale*: This ensures smooth onboarding and continuity in development.
  - *Fit Criterion*: The documentation must be updated with every major release.
- The application must be designed to accommodate future development, including the addition of new experimental parameters or features without backwards progression.
  - *Rationale*: This ensures the application can scale and evolve without compromising existing features.

## 12.2 Supportability Requirements

- The application must be self-supporting, featuring an intuitive user interface that minimizes the need for external assistance.
  - *Rationale*: This ensures a user-friendly experience that reduces the need for help desk support.
  - *Fit Criterion*: At least 90% of users should complete tasks without needing support, as measured by usability testing.
- The application must have automated guidance, such as error messages, to assist users in troubleshooting common issues.
  - *Rationale*: This 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.

## 12.3 Adaptability Requirements

- The application must be compatible with modern web browsers to ensure widespread accessibility.
  - *Rationale*: This 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.

## 13 Security Requirements

### 13.1 Access Requirements

- Access to the application must be restricted to authorized personnel, with an authentication mechanism.
  - *Rationale*: This ensures that only authorized users can interact with the application.
  - *Fit Criterion*: Only users with valid credentials should access the application.
- 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*: This ensures users can only perform actions that align with their roles.
  - *Fit Criterion*: Role-based access control (RBAC) must restrict 60% of actions to users' defined permissions.

### 13.2 Integrity Requirements

- The application must validate data inputs to ensure they conform to expected formats and values before they are processed.
  - *Rationale*: This ensures only valid data is processed, reducing errors.
  - *Fit Criterion*: 100% of inputs must pass validation checks before processing.
- The application must not modify the data unnecessarily through its transfer process.
  - *Rationale*: This ensures the original data remains accurate and unaltered.

- *Fit Criterion*: Data should remain unchanged unless explicitly modified, with logs confirming its integrity.
- The application must ensure that any data processed or transferred is free from duplication or inconsistencies.
  - *Rationale*: This ensures data consistency and prevents corruption.
  - *Fit Criterion*: The application must detect and prevent 100% of duplicated records.
- The application must have safeguards in place to maintain the accuracy of the transferred data.
  - *Rationale*: This ensures reliable data transfer without loss or error.
  - *Fit Criterion*: Transfer operations should maintain 100% data accuracy, verified by validation tests.

### 13.3 Privacy Requirements

- 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*: This ensures user privacy and legal compliance.
- 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*: This ensures transparency and control over data sharing.

### 13.4 Audit Requirements

- The application must maintain a comprehensive audit trail, logging all access and modification events, including timestamps and identities of users performing actions.
  - *Rationale*: This ensures accountability and traceability of actions.



- *Fit Criterion*: 100% of data access and modification events must be logged and retrievable.
- Audit logs must be securely stored and accessible only by authorized personnel.
  - *Rationale*: This ensures the security and integrity of audit data.
  - *Fit Criterion*: Logs must be encrypted and accessible only to users with administrative privileges.

## 13.5 Immunity Requirements

- 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*: This 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.

## 14 Cultural Requirements

### 14.1 Cultural Requirements

*Insert your content here.*

## 15 Compliance Requirements

### 15.1 Legal Requirements

*Insert your content here.*

### 15.2 Standards Compliance Requirements

*Insert your content here.*

## **16 Open Issues**

*Insert your content here.*

## **17 Off-the-Shelf Solutions**

### **17.1 Ready-Made Products**

*Insert your content here.*

### **17.2 Reusable Components**

*Insert your content here.*

### **17.3 Products That Can Be Copied**

*Insert your content here.*

## **18 New Problems**

### **18.1 Effects on the Current Environment**

*Insert your content here.*

### **18.2 Effects on the Installed Systems**

*Insert your content here.*

### **18.3 Potential User Problems**

*Insert your content here.*

### **18.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product**

*Insert your content here.*

## 18.5 Follow-Up Problems

*Insert your content here.*

## 19 Tasks

### 19.1 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.

### 19.2 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.

Stage	Milestone	Deadline
Stage 1	Problem Statement, POC Plan, Development Plan Requirements Document Revision 0	Sept 24 Oct 9
Stage 2	Hazard Analysis V&V Plan Revision 0	Oct 23 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) EXPO Demonstration Final Documentation (Revision 1)	Mar 24 - 30 Apr TBD Apr 2

Table 1: Project Decomposition and Deadlines

### 19.3 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
Phase 1	POC	November 11
Phase 2	Revision 0 Demonstration	February 3
Phase 3	Revision 1 Final Demonstration	March 24

Table 2: 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** is the Revision 1 Final Demonstration, where the final version of the project is presented. 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.

## 20 Migration to the New Product

### 20.1 Requirements for Migration to the New Product

**Q:** Will you use a phased implementation to install the new system? if so, describe which requirements will be implemented by each of the major phases?

**A:** No, the system will be installed in a single go after being tested and approved.

*Note:* Include cross-references between development tasks, project phases and the Product Use Cases and atomic requirements.

**Q:** What kind of data conversion is necessary? Must special programs be written to transport data from an existing system to a new one? If so, describe the requirements for these programs here.

**A:** Currently, the files are stored in the CSV format. In order to make them compatible for migration to the new database (MongoDB), the files must be converted into JavaScript Object Notation (JSON) format. Yes, a special program must be written to transport the data as the new system will only accept JSON files. This means that when the file is inputted to the new system (in CSV format), a special program must first convert it to JSON so that the data can be entered in the database.

**Q:** What kind of manual backup is needed while the new system is installed?

**A:** The installation of the new program should not be a long process and can be accomplished on a day when there are no experiments being run in the lab. This means, no manual backup would be necessary while the new system is being installed.

**Q:** When are each of the major components to be put in place? When are the phases of the implementation to be released?

**A:** No phases of implementation to be released. Major components will be put in place based on this timeline -

- Database with migrated data and functionality to compare parameters: Proof of Concept Demonstration
- Interface that allows inputting data and adding new parameters: End of December (before Christmas break)
- Dashboard that allows analysis of data and visualizing graphs: End of January

**Q:** Is there a need to run the new product in parallel with the existing product?

**A:** No, there is no need to run the new product in parallel with the existing product. Data can be migrated at regular intervals once the database is up and running.

**Q:** Is there any special effort needed to decommission the old product?

**A:** The old product is essentially a collection of spreadsheets so no special effort is needed to decommission them. A visual inspection and testing might be required after data is migrated to the new product to ensure the accuracy and integrity of the data.

## **20.2 Data That Has to be Modified or Translated for the New System**

**Q:** Description of current technology that holds data.

**A:** Currently the data is retrieved in the form of CSV files and copied onto existing Excel templates that are used to sort and analyse the data. Although the templates are sophisticated and well-designed, 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 parametric 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.

**Q:** Description of data translation tasks.

**A:** A script will be written to automate the translation of files from CSV to JSON format.

**Q:** Foreseeable Problems.

**A:** N/A

## 21 Costs

*Insert your content here.*

## 22 Look and Feel Requirements

All subitems in this sections are the fit criterion for its respective requirements.

### 22.1 Appearance Requirements

AR1. The website should have a simple and organized layout, with clearly defined sections where all major functions should be easily accessible and viewable

AR1.1 A users should be able to identify all the major functions of the website within five minuets of use.

AR2. The website shall be responsive on all computer and laptop screens aside from mobile screens

AR2.1 The usability of website should be the same as the default view on big and smaller computer, laptop and monitor screens

AR3. The website's functions and buttons shall be properly labeled where no button is ambiguous to users

AR3.1 A user should be able to tell what all buttons inherently does without the needing to ask questions

AR4.1 The produced plot from the data shall be properly labeled

AR4.1 The plots should not be ambiguous. Users should be able to understand what is the plot about within five minuets of viewing it.

## 22.2 Style Requirements

SR1. All icons on the website must be in the artistic style

SR1.1 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.

SR2. All colours must match the theme of the website

SR2.1 After a user's first encounter with the product, 80% of users should agree that there is a common theme throughout the website.

SR3. All fonts are to be consistent throughout the website

SR3.1 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.

## 23 Waiting Room

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

### 23.1 Automatically Download Data from Lab Apparatus

**Description:** 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

## 23.2 Dynamic Dashboard to Generate Comparison Reports

**Description:** 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

## 23.3 Machine Learning Analysis and Projections

**Description:** 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 thought of. The aim is to explore analyses that might not have been considered by a user.

**Expected Version:** Version 3

## 23.4 Mobile Development and Accessibility

**Description:** The product shall comply with relevant accessibility standards both for the web version and the 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

## 24 Ideas for Solution

*Insert your content here.*

## Appendix — 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 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.
2. 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?