

SE 4G06: Software Requirements Specification
*Measuring Microstructure Changes During
Thermal Treatment*

Team #30, ReSprint
Edwin Do
Joseph Braun
Timothy Chen
Abdul Nour Seddiki
Tyler Magarelli

Contents

List of Tables

List of Figures

Table 1: **Revision History**

| Date | Developer | Notes/Changes |
|---------------|--------------------|---|
| Sept 25, 2022 | Edwin Do | Revision 0 - Initial commit |
| Oct 5, 2022 | Edwin Do | Adopt Volere template + Add content |
| Oct 5, 2022 | Timothy Chen | Added to Non-Functional Requirements |
| Oct 5, 2022 | Abdul Nour Seddiki | Added to Relevant Facts and Assumptions |
| Oct 5, 2022 | Abdul Nour Seddiki | Added to Waiting Room |
| Oct 5, 2022 | Timothy Chen | Added to Reflection |
| Oct 5, 2022 | Joseph Braun | Add functional Requirements |
| Oct 5, 2022 | Abdul Nour Seddiki | Added to Reflections |
| Oct 5, 2022 | Edwin Do | Add reflection + small fixes |
| Oct 6, 2022 | Edwin Do | Add missing tasks, ideas for solutions, and open issues |
| Oct 31, 2022 | Edwin Do | Update with the use of Visual Studio |
| Mar 8, 2023 | Abdul Nour Seddiki | Added two functional requirements |
| Apr 1, 2023 | Edwin Do | Added definitions for resistivity and conductivity |
| Apr 1, 2023 | Tyler Magarelli | Outlined variation from the template document |
| Apr 1, 2023 | Abdul Nour Seddiki | Added dangling section descriptions and fix spelling errors |
| Apr 1, 2023 | Joseph Braun | Added to Facts and updated Context Diagram |
| Apr 1, 2023 | Tyler Magarelli | Added Traceability Matrices |
| Apr 1, 2023 | Edwin Do | Revised requirements and use cases |

This document describes the software requirements for the capstone project of measuring microstructure changes of samples during thermal treatment. The template for the Software Requirements Specification (SRS) is a subset of the Volere template. The section regarding likely changes was renamed to 'Waiting Room' and the team removed the section for unlikely changes. The 'Project Drivers' section encompasses the Introduction, General and Specific System description sections from the template.

1 Project Drivers

This section will introduce the purpose of the project and the stakeholders, along with the requirement constraints and facts and terminology in order to set the scene about the project's main drivers.

1.1 The Purpose of the Project

The purpose of this project is to assist the Department of Materials Engineering in measuring the changes to a material's microstructure during thermal treatment. By doing so, the resistivity of the sample can be measured at different thermal levels. The goal is to be able to collect the data at the necessary sampling rate and incorporate the use of Windows GUI.

1.2 The Stakeholders

The stakeholders will be introduced in this section, they include the developers, clients, and customers.

1.2.1 Developers

The Developers will be responsible for the design, development, and documentation throughout. They will be utilizing the existing lab equipment and computer for the duration of this project. Developers will also use the feedback from the client to deliver the final product.

1.2.2 The Client

The Client for this project is the Department of Materials Engineering and the Computing and Software Department at McMaster University. More

specifically, Dr. Zurob, Dr. Smith, and the TAs of 4G06 will be the ones to evaluate, review and provide feedback on the project throughout the development process.

1.2.3 The Customers

The Customers for this project are anyone who will be conducting research or require data that measures the microstructural changes of materials under thermal treatment.

1.2.4 Other Stakeholders

This project has no other stakeholders.

1.3 Mandated Constraints

This section outlines the general constraints of this project and categorizes them on a solution, software, environment, schedule, and budget constraints bases.

1.3.1 Solution Constraints

Description: The GUI will run on Windows operating systems

Rationale: The application is a Desktop application. The lab computer that has the capability to connect to other required lab equipment currently runs on Windows.

Fit Criterion: Users can successfully install and open the application on a supported Windows operating system.

Description: The sampling rate of the equipment to the GUI will be at least 100 times per second

Rationale: According to Dr. Zurob, this is the minimum sampling rate needed to see any meaningful data

Fit Criterion: The equipment samples the data at 100 times per second and the GUI accurately reflects the measurements

1.3.2 Off-the-shelf Software

The application will use Visual Studio to create a WPF (Windows Presentation Foundation) application that will ensure compatibility with the lab Windows computer.

1.3.3 Anticipated Workplace Environment

The software and equipment will be designed for the expected environment of a lab. The reason is that the lab equipment and computer are needed for the software to run successfully and should not be easily accessible outside of campus.

1.3.4 Schedule Constraints

The deadline for the final product is March 27, 2023. There will be other milestones during the development process that must be accomplished throughout. This will be outlined in our GitHub milestones.

1.3.5 Budget Constraints

At this point, there is an estimated budget of \$1000. This may change as the team determines what additional equipment is needed to work with the lab equipment.

1.4 Naming Conventions and Terminology

- **C#**: Scripting language used to create and control dynamic content.
- **Visual Studio**: Development IDE used to create a variety of applications
- **WPF**: Windows Presentation Foundation - Library used to create Windows Applications.
- **Windows**: A popular operating system used by many users.
- **GUI**: Graphical User Interface - A digital interface which allows the user to interact with electronic devices through graphical icons.

- **Product/Software/Application:** Refers to the final deliverable of this capstone project.
- **User:** The person who will be interacting/ using the application.
- **Resistivity:** A measure of the resisting power of specified material to the flow of an electric current.
- **Conductivity:** The ability to move heat or electricity from one place to another.
- **Data Acquisition Rate:** The internal clock timing of the instruments used in calculating measurements, the range is between MIN_SAMPLE_RATE and MAX_SAMPLE_RATE.

1.5 Relevant Facts and Assumptions

Facts, assumptions, and user characteristics that are relevant to the scope of this project are shown in this section.

1.5.1 Facts

The lab computer is using the Windows XP operating system which is questionably upgradeable and it is worth mentioning that Windows XP is currently unsupported. The equipment for power supply and measurement tools is outdated and requires special connection media, both physical connections and drivers, to communicate with the computer. There exists an application with a similar purpose and function to our project's on the computer, which does not fulfill the goals of the user anymore, and the source code for that application is not provided.

Resistivity is the measure of the electrical resistance of a conductor of unit cross-sectional area and unit length. It is a characteristic property of each material and is useful for comparing materials on their ability to conduct electric currents. Conductivity is the measure of the ease at which an electric current can pass through a material. Conductivity is the inverse of resistivity.

Resistivity of sample material is calculated by the following equation. Where ρ is the resistivity, R is the resistance, A is the cross-sectional area, and L is the length of the sample: $\rho = \frac{RA}{L}$

1.5.2 Assumptions

Statement: It is assumed that materials will have microstructural changes when they undergo heat treatment. Effect: In case this assumption is false, our product will display no output.

Statement: Another assumption is that these microstructural changes will result in changes in the resistivity of the materials treated. Effect: If this assumption is false, the product will be rendered unsuitable for the task as its main function is to measure changes in resistivity.

Statement: According to the project timeline, it is assumed that a proof of concept will be ready by November 14th, a first functional design will be ready by February 6th, and a final revision will be demoed by March 27th. Effect: These dates are made to observe the progress of the project. In case the final demo is not met, the project will have been incomplete, since that will be the end-of-term for the developers.

1.5.3 User Characteristics

An assumption made about this project is that the users will have the necessary knowledge to safely operate the necessary lab equipment. This is required to collect the data and display it on the GUI. The user is also assumed to have a general working knowledge of how to install and open a Windows application, as well as the use of a mouse and keyboard input. Another assumption is that the user is literate in English.

2 Functional Requirements

The functional requirements section puts the functional needs of the stakeholders in context, shows possible use cases, and finally lists the functional requirements themselves.

2.1 The Scope of the Work and the Product

The hardware for this project is provided by the Department of Materials Engineering through the project supervisor, Dr. Zurob. A Windows computer, current source, and nanovoltmeter have already been included. A

fourth hardware device, for measuring the temperature of the sample material, will also be provided or else a new device shall be purchased for this purpose. The product will be a Windows-based GUI application that can set the data acquisition rate, connect to the hardware devices, acquire the output data, and calculate the conductivity of the sample material in real time. The application should also be able to be accessed remotely to check on the progress of ongoing experiments.

2.1.1 The Context of the Work

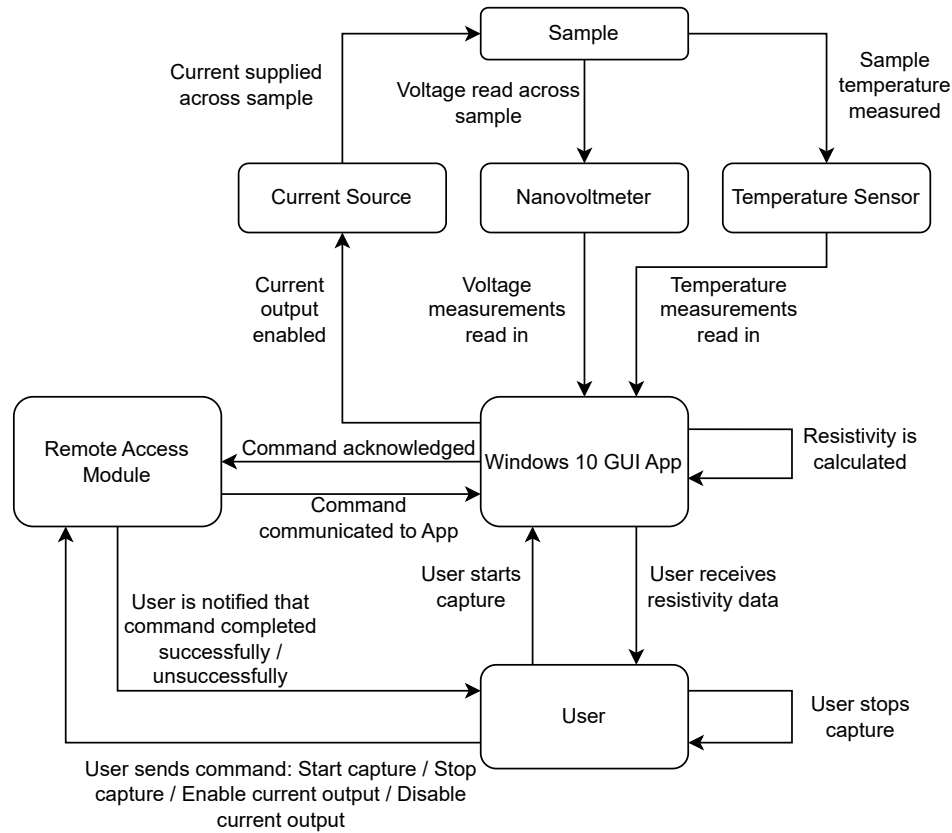


Figure 1: Basic context diagram (subject to change as the project progresses)

2.1.2 Work Partitioning

Table 2: Work Partitioning

| Event # | Event Name | Input | Output |
|---------|--------------------------------|------------------------------|------------------------------|
| 1 | User requests resistivity data | Data request | Resistivity data |
| 2 | App connects to hardware | Data sampling rate | Voltage and temperature data |
| 3 | App calculates resistivity | Voltage and temperature data | Sample resistivity |

Table 3: Work Partitioning - Description

| Event # | Description |
|---------|---|
| 1 | User opens app and requests resistivity data for the current sample connected to hardware |
| 2 | App interfaces with hardware, sets sampling rate, and records data from hardware |
| 3 | App uses data received from hardware to calculate resistivity of sample |

2.1.3 Individual Product Use Cases

UC-1: Install and use the application in the lab to measure resistivity changes in sample material

Related Requirements: ??, ??, ??, ??, ??

Initiating Actor: The User

Actor's Goal: Record resistivity changes in sample material during thermal treatment

Participating Actors: User, Application, Hardware

Pre-conditions: User in lab; sample material connected to hardware

Flow of events for main success:

- 1. The user installs and opens the application
- 2. The user starts the experiment
- 3. The user requests resistivity data from the application
- 4. The application interfaces with hardware

- ← 5. Hardware outputs voltage and temperature data to the application
- ← 6. The application calculates and outputs resistivity data to the user

UC-2: Control the lab devices remotely while an experiment that monitors resistivity changes is running

Related Requirements: FR1, FR2, FR3, FR4, FR5

Initiating Actor: The User

Actor's Goal: Monitor resistivity changes in sample material remotely during thermal treatment

Participating Actors: User, Application, Hardware

Pre-conditions: The user remotely connected to the application; sample material connected to hardware

Flow of events for main success:

- 1. The user requests to turn on the current output
- ← 2. The request is relayed over a network to the application
- ← 3. The application interfaces with hardware
- 4. The current output is turned on
- ← 5. The application checks the status of the current output
- ← 6. The application sends confirmation back to the user remotely

UC-3: Use the application to view a visual representation of the data and export it to a file that can be saved in other file locations

Related Requirements: FR6, FR7

Initiating Actor: The User

Actor's Goal: See a visual output of captured data and save it

Participating Actors: User, Application

Pre-conditions: User is successfully running an experiment that is capturing data

- Flow of events for main success:**
- 1. The user stops capture
 - 2. The user uses the graphical display and its tools to analyze data visually
 - 3. The user requests the application to output the graphical data
 - ← 4. A file containing an image of the graphical display is outputted to the computer
 - ← 5. The user requests the application to output the captured data
 - ← 6. A file containing the data of the captured data is outputted to

the computer

2.2 Functional Requirements

- FR1. The app shall monitor the resistivity of the sample material in real time.
- FR2. The app shall change the data sampling rate as required.
- FR3. The app shall have remote access and control.
- FR4. The app shall display measurements and calculations in a graph.
- FR5. The app shall have a file output system.
- FR6. The app shall be able to be installed on a Windows computer.

3 Non-functional Requirements

This section puts the requirements that are not necessarily essential to the function of the product into perspective. Requirements such as the look, usability, performance, security, and legal requirements are specified below.

3.1 Look and Feel Requirements

- NFR-L1. The product shall feel simple to use.
Fit Criterion: Survey should reflect 90 percent of users should feel like the product is uncomplicated to use.
- NFR-L2. The product shall be in English only.
Fit Criterion: The language used throughout the product will be in English.

3.2 Usability and Humanity Requirements

- NFR-U1. Users with no prior experience with the product should be able to use it.
Fit Criterion: New user will be able to complete each task successfully within 1 minute.

- NFR-U2. Product shall have a straightforward interface allowing for quick modifications to parameters with ease.
Fit Criterion: The time between the user interacting with the application and modifying a parameter to change it in the interface should be no longer than INTERACT_TIME.
- NFR-U3. Product shall help the user accurately make modifications and avoid mistakes.
Fit Criterion: The total number of mistakes made by the user should be no more than MAX_MISTAKE when completing a set of tasks.
- NFR-U4. Users should be able to learn the use of the product over time.
Fit Criterion: User will be able to complete a set of tasks faster the next day for the second time compared to the first time.
- NFR-U5. The product shall conceal detailed structures and calculations from the user.
Fit Criterion: The product will not show any calculations used for producing output based on the user's parameters.
- NFR-U6. The capacity of the product shall not be large.
Fit Criterion: The product will be no more than MAX_SIZE.

3.3 Performance Requirements

- NFR-P1. The product shall be able to read at MIN_SAMPLE_RATE
Fit Criterion: The rate of reading will be measured and the rate determined by the measurement shall be no less than MIN_SAMPLE_RATE.
- NFR-P2. Changes to the parameters will be reflected in the product within TIME_ACCEPTED of the user's input.
Fit Criterion: The product will reflect changes given by the user within TIME_ACCEPTED.
- NFR-P3. The product shall read measurements and calculations shall be accurate to ACCEPTED_SIGFIG.
Fit Criterion: The measurements and calculations will be verified to ACCEPTED_SIGFIG.
- NFR-P4. When the user is using the product, it shall be up and running for at least MIN_UPTIME.

Fit Criterion: The product will be up and running during the interactions with the user and for at least MIN_UPTIME after.

3.4 Operational and Environmental Requirements

NFR-O1. Product should accept inputs from keyboard and mouse.

Fit Criterion: The keyboard and mouse connected to the computer will be able to interact with the product.

NFR-O2. Product shall be able to be installed by a user with no prior experience with the product on a Windows Computer.

Fit Criterion: User with no prior experience of installing the application should be able to install the application with the optional assistance of an user guide.

3.5 Maintainability and Support Requirements

NFR-M1. Product shall work on Windows 10/11 operating systems.

Fit Criterion: The product will be installed on the operating system and have its functions verified.

NFR-M2. Product shall be expected to work on the computers in the lab.

Fit Criterion: The product will be installed and used as expected on the lab's computer.

3.6 Security Requirements

NFR-S1. The product shall prevent modifications or injections of measurements.

Fit Criterion: The product will only allow users to read measurements and deny any attempts to change them.

NFR-S2. Only authorized users are allowed to modify concealed calculations, settings, and/or parameters.

Fit Criterion: Users who have clearance will have access to modify certain calculations and parameters.

3.7 Cultural Requirements

NFR-C1. The product must not include any graphics or wording that may be considered offensive or inappropriate to the user.

Fit Criterion: To measure this, a usability survey will be conducted. User indicating they agree or strongly agree graphics and wording are not offensive or inappropriate will be considered successful.

3.8 Legal Requirements

N/A

3.9 Health and Safety Requirements

NFR-H1. Colours and graphics used in the application should take into account users who may be prone to seizures.

Fit Criterion: There should be no animations that simulate flashing/ flickering (i.e change of brightness or colour at a rapid rate). There should also be no static optical illusions that may simulate any flashing/ flickering.

NFR-H2. Colours should not be too bright, causing potential harm to users' eyes.

Fit Criterion: Colours of the GUI should be checked to ensure it does not simulate extra light. Example: colours that include the words 'bright,' 'flashy' or 'neon'.

3.10 Installability Requirements

NFR-I1. Product requires a Windows computer with the necessary ports to connect to the lab equipment.

Fit Criterion: Copy over the zipped file with the executable within to the new computer. Unzip the folder and open the application to see if the readings from the lab equipment are reflected correctly.

4 Project Issues

This section will go into detail on the lists of next items to address, off-the-shelf solutions, potential problems, risks, costs, ideas, and waiting room items.

4.1 Open Issues

- Determine maintainability and stability of the Windows XP operating system
- Identify the best-suited method for communication between lab equipment and the application
- Identify the appropriate hardware and software development to ensure the best sampling rates
- Investigate the necessary drivers, if any, on the lab computer for it to work with the lab equipment

4.2 Off-the-Shelf Solutions

The application will use Visual Studio to create a WPF (Windows Presentation Foundation) application that will ensure compatibility with the lab Windows computer.

4.2.1 Ready Made Components

The application will use many of the existing IDE extensions in Visual Studio to further support communication with any equipment in the lab.

4.3 New Problems

A potential problem from our product that may arise is the user's ability to learn the software.

4.3.1 Potential User Problems

This product introduces a new learning curve for the user to use the application. To minimize this problem, the product will be implemented with a quick start guide and developers will design a user-friendly interface.

4.4 Tasks

- Find and create a method of communication between the lab equipment and the application (i.e. serial ports, named pipes, inter-process communication)
- Accurately read the data output in real-time from the application
- Create a GUI that can be installed and successfully run on a Windows 10 operating system

4.5 Migration to the New Product

N/A

4.6 Risks

A risk to this project is that the current lab computer has special ports to communicate with the necessary hardware equipment. Although creating a WPF application should help with compatibility on Windows computers, there is a risk to whether the necessary drivers are available and if the communication will still work after upgrading to Windows 10.

Another risk is that the lab equipment does not offer the necessary sampling or is not compatible with the lab computer.

4.7 Costs

The largest estimated cost of this project is time. It will require both the developers and the client's time to work on and evaluate the project throughout. Additional expenses may be added if additional or new lab equipment is required.

4.8 User Documentation and Training

A main README file will be created and documented for information such as installation, system requirements, and available features. An additional safety document will also be created for users, before using any of the lab equipment.

4.9 Waiting Room

- Requiring the highest possible rate of output.
- Requiring the software to work on the latest (supported) version of Windows.
- The option of using the application remotely.
- The option of using the application on different platforms, operating systems, devices, and with different input and measurement tools.
- Ability of the system to be a real-time system where it is used concurrently with the heat treatment process of materials, connected to the heat treatment systems and controlling them, measuring changes in real-time, and making decisions on the fly.
- Incorporation of machine learning and artificial intelligence in order to analyze data within processes and make adjustments to the treatment for customized optimal results.
- Addition of other methods of measurement of microstructural change; like adding high-speed electronic microscopes with slow-motion capture for manual or automated analysis of the physical outputs.

4.10 Ideas for Solutions

- Use Electron to create Desktop Application
- Use Visual Studio to create UWP/WPF (Windows Presentation Foundation) Application
- Write the data from the lab equipment to a serial port for communication with the application

- Utilize electron libraries to support communication on a serial port
- Utilize inter-process communication for communication between the lab equipment and application

5 Traceability Matrix

In this section, an overview of the relationships between use cases and functional requirements, and other relationships between functional and non-functional requirements are provided. This allows for easy bi-directional navigation and ensures all requirements and use cases are accounted for.

| | ?? | ?? | ?? | ?? | ?? | ?? | ?? |
|----|----|----|----|----|----|----|----|
| ?? | X | X | X | | | | X |
| ?? | X | X | X | X | | | |
| ?? | | | | | X | X | |

Table 4: Traceability Matrix Showing the Connections Between Use Cases and Functional Requirements

| | ?? | ?? | ?? | ?? | ?? | ?? |
|----|----|----|----|----|----|----|
| ?? | | | | X | X | |
| ?? | | | | | | |
| ?? | | | | | | |
| ?? | | | X | | | |
| ?? | | X | | X | X | |
| ?? | | | | | | |
| ?? | | | | | | |
| ?? | | | | | | |
| ?? | | X | | | | |
| ?? | X | | X | X | | |
| ?? | X | | | X | X | |
| ?? | X | | | | | |
| ?? | | | | | | |
| ?? | | | | | | |
| ?? | | | | | | |
| ?? | | | | | | |
| ?? | X | | | | | |
| ?? | | | | | | |
| ?? | | | | X | X | |
| ?? | | | | X | X | |
| ?? | | | | X | X | |
| ?? | | | | | | X |

Table 5: Traceability Matrix Showing the Connections Between Functional and Non-Functional Requirements

6 Appendix

This section contains a table making reference to symbolic parameters used throughout the document.

6.1 Symbolic Parameters

| symbol | description |
|-----------------|------------------------|
| TARGET_TIME | 60 seconds |
| INTERACT_TIME | 5 seconds |
| MAX_MISTAKE | 2 |
| MAX_SIZE | 8GB |
| MIN_UPTIME | 30 minutes |
| MIN_SAMPLE_RATE | 60 samples per second |
| MAX_SAMPLE_RATE | 600 samples per second |
| TIME_ACCEPTED | 1 second |
| ACCEPTED_SIGFIG | 3 decimals |

7 Phase In Plan / Development Plan

| Priority | Functional Requirement | Planned Completion Date |
|----------|------------------------|-------------------------|
| 1 | ?? | November 10, 2022 |
| 2 | ?? | November 15, 2022 |
| 3 | ?? | November 24, 2022 |
| 5 | ?? | December 12, 2022 |
| 6 | ?? | January 17, 2023 |
| 7 | ?? | February 3, 2023 |

Table 6: Development Plan chart that highlights priorities of development and planned completion dates.

8 Reflections

Q1: What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project?

Q2: 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? From the identified approaches, which will each team member pursue, and why did they make this choice?

Responses

Timothy - We need to acquire the knowledge and skills on developing a Window application using the Electron as the framework. One of the main requirements from the supervisor is that it has to work on Windows operating systems. In my experience through Co-op and school work, I have yet to interact with Windows applications. This knowledge will hope us meet the requirements set by the supervisor as well as learn a new technical skill.

There are a few approaches to acquiring this knowledge. The first way would be to look for blogs from other developers and learn from their examples. The second way would be to watch and read tutorials online. The third way would be to read the documentation and conduct research. Lastly, we could also learn by trial and error. The approach I will be taking will be watching and reading tutorials online as they will be able to explain concepts in a simpler form with visual aids. This will help me gain knowledge in this concept and make it a skill I can gain from this project.

Abdul - The team needs to learn about programming in relation to signal inputs and outputs. Potentially signal processing knowledge would be involved. Another aspect we will need to learn about is simple circuit design specifically for testing metallic specimens, which involves dealing with current sources and nanovoltmeters.

Online resources are a very good candidate for these skills. Mostly happening through continuous research along with trial and error. Some material science-specific questions could be raised to Dr. Zurob.

Joseph - One of the main components of the project will be to interface the Windows application with the hardware. Since none of the team members has worked with this particular hardware setup before, this will be new knowledge

we are required to learn.

There are two main ways I can think of to acquire this knowledge. The first way would be to use the old Windows application which is currently installed on the lab computer. This application could be reverse-engineered to determine how the software interfaced with the hardware, and then this method could be copied for our own application. A second way would be to study the data sheets and/or user manuals of the hardware in order to learn how the hardware is designed to interface with software. The approach I will be taking is the latter. I may find out through this approach that there is more than one possible method, in which case we could choose the one most suitable for our needs rather than be limited to one method.

Edwin - The group needs to collectively learn Electron, a JavaScript framework that the team is planning on utilizing to develop the desktop application. In addition, the group will also need to collectively learn different methods of how to enable communication between the lab equipment and the application. An example of this may be inter-process communication.

Two approaches to learning Electron as a new framework would be to create a prototype and follow online tutorials. For myself, I will approach learning Electron by creating quick prototypes that accomplish a task we need. This may be supplemented by reading available documentation.

Learning the different methods that allow hardware and software to communicate will require research and tutorials. By researching we can learn the strengths and weaknesses to determine which method is best suited for our project. Tutorials will help provide quick hands-on experience on how to get communication to work.

Tyler - The group needs to learn how to use a JavaScript framework to receive inputs from external hardware.

This can be done through extensive research into the framework at hand and its built-in libraries and functionality. Starting with small partitioned tasks and increasing in size and difficulty slowly will help us to fully understand how to use it all.