

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

1	Project Drivers	1
1.1	The Purpose of the Project	1
1.2	The Stakeholders	1
1.2.1	Developers	1
1.2.2	The Client	1
1.2.3	The Customers	1
1.2.4	Other Stakeholders	2
1.3	Mandated Constraints	2
1.3.1	Solution Constraints	2
1.3.2	Off-the-shelf Software	2
1.3.3	Anticipated Workplace Environment	2
1.3.4	Schedule Constraints	2
1.3.5	Budget Constraints	3
1.4	Naming Conventions and Terminology	3
1.5	Relevant Facts and Assumptions	3
1.5.1	Facts	3
1.5.2	Assumptions	4
1.5.3	User Characteristics	4
2	Functional Requirements	4
2.1	The Scope of the Work and the Product	4
2.1.1	The Context of the Work	5
2.1.2	Work Partitioning	5
2.1.3	Individual Product Use Cases	6
2.2	Functional Requirements	7
3	Non-functional Requirements	7
3.1	Look and Feel Requirements	7
3.2	Usability and Humanity Requirements	7
3.3	Performance Requirements	8
3.4	Operational and Environmental Requirements	9
3.5	Maintainability and Support Requirements	9
3.6	Security Requirements	10
3.7	Cultural Requirements	10
3.8	Legal Requirements	10
3.9	Health and Safety Requirements	10

3.10	Installability Requirements	11
4	Project Issues	11
4.1	Open Issues	11
4.2	Off-the-Shelf Solutions	11
4.2.1	Ready Made Components	11
4.3	New Problems	11
4.3.1	Potential User Problems	12
4.4	Tasks	12
4.5	Migration to the New Product	12
4.6	Risks	12
4.7	Costs	12
4.8	User Documentation and Training	13
4.9	Waiting Room	13
4.10	Ideas for Solutions	13
5	Appendix	14
5.1	Symbolic Parameters	14
5.2	Reflections	14

List of Tables

1	Revision History	iii
2	Work Partitioning	5
3	Work Partitioning - Description	6

List of Figures

1	Basic context diagram (subject to change as project progresses)	5
---	---	---

Table 1: **Revision History**

Date	Developer	Notes/Changes
Sept 25, 2021	Edwin Do	Revision 0 - Initial commit
Oct 5, 2021	Edwin Do	Adopt Volere template + Add content
Oct 5, 2021	Timothy Chen	Added to Non-Functional Requirements
Oct 5, 2021	Abdul Nour Seddiki	Added to Relevant Facts and Assumptions
Oct 5, 2021	Abdul Nour Seddiki	Added to Waiting Room
Oct 5, 2021	Timothy Chen	Added to Reflection
Oct 5, 2021	Joseph Braun	Add functional Requirements

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.

1 Project 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 necessary sampling rate and incorporate the use of Windows GUI.

1.2 The Stakeholders

1.2.1 Developers

The Developers will be responsible for the design, development, and documentation throughout. They will be utilizing the existing lab equipment, 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 TAs of 4G06 who 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 various thermal treatment.

1.2.4 Other Stakeholders

This project has no other stakeholders.

1.3 Mandated Constraints

1.3.1 Solution Constraints

Description: The GUI will run on Windows operating system

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

No off-the-shelf software is required for this project.

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 is 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 the March 20 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

- **JavaScript:** Scripting language used to create and control dynamic content.
- **HTML:** Standard markup language for creating web pages.
- **CSS:** Style sheet language for structuring and styling HTML web page.
- **Windows:** A popular operating system used by many users.
- **Product/Software/Application:** Refers to the final deliverable of this capstone project.
- **User:** The person who will be interacting/ using the application.
- **Electron:** JavaScript framework used to develop Desktop applications.

1.5 Relevant Facts and Assumptions

1.5.1 Facts

The client and control computer is using Windows 7 operating system that is questionably upgradable and it is worth mentioning that Windows 7 is currently unsupported. The equipment for power supply and measurement tools are outdated and require 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 of a 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 resistivity of the materials treated. Effect: If this assumption is false, the product will be rendered unsuitable for the task as the main function of it 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 31st. Effect: These dates are made to observe 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 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

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 Window's based GUI application which can set the data acquisition rate (sample 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

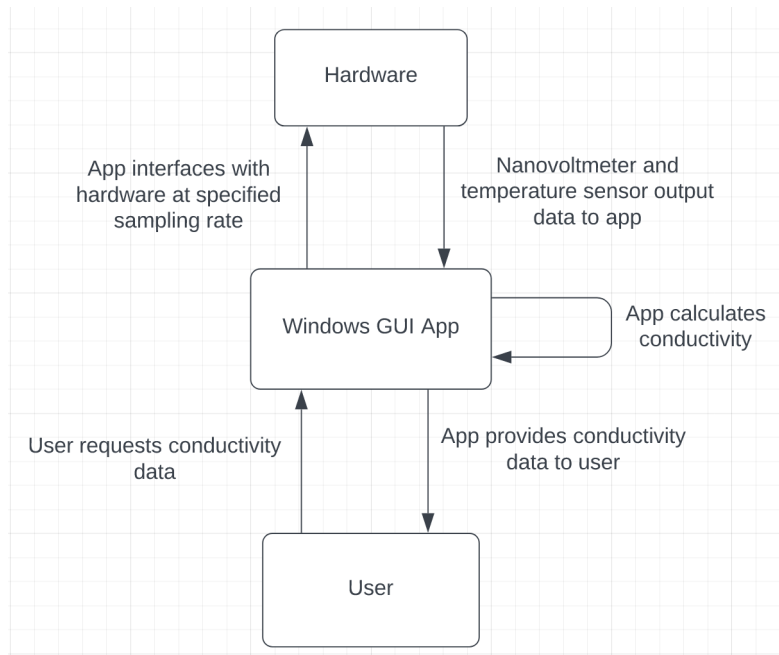


Figure 1: Basic context diagram (subject to change as project progresses)

2.1.2 Work Partitioning

Table 2: Work Partitioning

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

Table 3: Work Partitioning - Description

Event #	Description
1	User opens app and requests conductivity 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 conductivity of sample

2.1.3 Individual Product Use Cases

UC-1: Use the App in the lab to measure conductivity changes in sample material

Related Requirements: FR1, FR2, FR3, FR4

Initiating Actor: User

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

Participating Actors: User, App, Hardware

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

Flow of events for main success:

- 1. User requests conductivity data from App
- 2. App interfaces with hardware
- ← 3. Hardware outputs voltage and temperature data to App
- ← 4. App calculates and outputs conductivity data to User

UC-2: Access the App remotely to monitor conductivity changes in sample material

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

Initiating Actor: User

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

Participating Actors: User, App, Hardware

Pre-conditions: User remotely connected to App; sample material connected to hardware

Flow of events for main success:

- 1. User requests conductivity data
- 2. Request is relayed over network to App

- 3. App interfaces with hardware
- ← 4. Hardware outputs voltage and temperature data to App
- ← 5. App calculates and outputs conductivity data
- ← 6. Conductivity is relayed over network to User

2.2 Functional Requirements

- FR1. The app shall monitor the conductivity of the sample material in real time.
- FR2. The app shall identify critical changes due to phase transition in the sample.
- FR3. The app shall change the data sampling rate as required.
- FR4. The app shall automate the process of identifying slope changes and correlating these to phase changes in the sample.
- FR5. The app shall have remote access and control.

3 Non-functional Requirements

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: 90 percent of new users will be able to complete each task successful within 1 minute.

- NFR-U2. Product shall have a straightforward interface allowing for quick modification to parameters with ease.
Fit Criterion: The time between the user thinking about modifying a parameter to changing it in the interface should be no longer than 5 seconds.
- NFR-U3. Product shall help the user accurately make modification and avoid mistakes.
Fit Criterion: The total rate of mistakes made by the user should be no more than 1 percent over 4 months uses of the product.
- NFR-U4. Users shall not need to remember how to interact with the product.
Fit Criterion: User will be able to use the product accurately within 5 second after 12 hours of not interacting with the product.
- NFR-U5. The product shall conceal detail structures and caluations from the user.
Fit Criterion: The product will not show any caluations 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 8 GB.
- NFR-U7. Product shall not require manually setting up after initial connection to the measurement equipment.
Fit Criterion: The user will have instructions on how to set up the connections the first time, no further set up will be needed after the initial installment.

3.3 Performance Requirements

- NFR-P1. The product shall be able to read 100 samplings/s.
Fit Criterion: The rate of reading will be measured and the rate determined by the measurement shall be 100 or more samplings/s for 95 percent of the time. Readings will be no less than 98 samplings/s.
- NFR-P2. Changes to the parameters will be reflected in the product within 1 second of the user's input.
Fit Criterion: The product will reflect changes given by the user

within 1 second for 95 percent of the time. The changes will take no longer than 2 seconds to show in the product.

NFR-P3. The product shall read measurements accurately to 3 decimal places.
Fit Criterion: The measurements will be accurate to 3 decimal places 100 percent of the time.

NFR-P4. The calculations and data displayed shall be accurate to 2 decimal places.
Fit Criterion: Displayed data provided to the user will be accurate 100 percent of the time.

NFR-P5. When the user is using the product, it shall be up and running 99 percent of the time.
Fit Criterion: The product will be up and running for at least 99 percent of the time during the duration of the tasks set by users.

3.4 Operational and Environmental Requirements

NFR-O1. Product should accept inputs from keyboard and mouse.
Fit Criterion: Keyboard and mouse connected to the computer will be able to interact with the product.

NFR-O2. Product shall be able to be installed with ease by a user with no prior experience with the product.
Fit Criterion: 90 percent of surveys from user with no prior experience should indicate the installation was simple.

NFR-O3. Releases occur at least once every 6 months.
Fit Criterion: The team will make a release with minor bug fixes and new features if needed at least once every 6 months.

3.5 Maintainability and Support Requirements

NFR-M1. Major bugs or issues brought up by the user shall be handled within 72 hours of receiving it.
Fit Criterion: Bugs or issues will be handled by developers within 48 hrs and will be escalated after so it can be resolved by 72 hour mark.

NFR-M2. Product shall work on Windows 7 and Windows 10 operating systems.

Fit Criterion: The product will be installed on both operating systems and have there functions verified.

- NFR-M3. 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 modifcations or injections of measurements.
Fit Criterion: The product will only allow users to read measurements and deny any attempts to change it.
- NFR-S2. Only autherized users are allowed to modify concealed caluations settings and/or parameters.
Fit Criterion: Users who have clearance will have access to modify certain caluations and parameters.

3.7 Cultural Requirements

- NFR-C1. The product must not include any graphics or terms that may be considered offensive or inappropriate to the user.
Fit Criterion: To measure this, a usability survey will be conducted to evaluate the graphics and terms on a scale of 1-10. Above 70% of the surveys returning with a score of 8 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 amy flashing/ flickering.

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

Fit Criterion: Colours of 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: Run the installation file and install the application successfully. Open the application to see if the readings from the lab equipment are reflected correctly.

4 Project Issues

4.1 Open Issues

ADD SOME TEXT HERE

4.2 Off-the-Shelf Solutions

The application will use Electron, a JavaScript framework that allows developers to create cross-platform compatible desktop applications. Since the use cases of this project are more specialized, there are not many existing solutions available on the market.

4.2.1 Ready Made Components

The application will use existing libraries in Electron to further support the 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

ADD STUFF HERE

4.5 Migration to the New Product

N/A

4.6 Risks

A risk to this project is that the current lab computer uses Windows 7 as its operating system. Although Electron has compatibility with Windows 7, there appears to be a few issues in the past on GitHub. In the case that it does not work, the operating system will have to be upgraded to Windows 10 and the compatibility with the lab equipment is uncertain. Additionally, McMaster University had notified the Department of Materials Engineering that Windows 7 is no longer supported but since the lab computer does not require any network connections, it has remained running Windows 7. This poses a future risk of the operating system being forcefully upgraded.

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 and evaluate the project throughout. Additional expense 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 analyse data within processes and making 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 on the physical outputs.

4.10 Ideas for Solutions

ADD MORE STUFF

5 Appendix

N/A

5.1 Symbolic Parameters

- `SAMPLING_RATE_PER_SECOND = 100`

5.2 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?

Response

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 system. In my experience through Co-op and school work, I have yet to interact with Window applications. This knowledge will help us meet the requirements set by the supervisor as well as learning 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.

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 datasheets 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.