Software Requirements Specification
MECHTRON 4TB6: Formulate

Team 25, Formulate
Ahmed Nazir, nazira1
Stephen Oh, ohs9
Muhanad Sada, sadam
Tioluwalayomi Babayeju, babayejt

October 5, 2022

Contents

Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

1 Introduction

1.1 Document Purpose

This document provides the set of Software Requirements Specifications (SRS) used to describe the system developed to assist testing efforts in technical teams. Both hardware and software system requirements were included to fully specify all system requirements.

The user can expect to understand the system behavior under expected use cases, the functional and non-functional requirements the system must adhere to, and a phase in plan.

1.2 Project Description

Effective test data collection and storage is a common challenge extra-curricular teams face in the technical domain. In teams who do not invest in streamlining data collection and storage, teams cannot fully utilize test data to validate designs. As a result, teams encounter difficulty proving design validity during competition, experience reduced competitiveness when presenting an under-validated system, and fail to generate trends on aggregated test data to efficiently find areas of improvement in design.

Project "Formulate" enables Formula teams to streamline data collection and storage, resulting in testing overhead reduction and increased control of raw test data gathered by automating aspects of the testing procedure.

1.3 Project Scope

Project Formulate aims to provide the McMaster Formula Electric team with a well-documented and complete system. To accomplish the project goals within an 8 month timeline, the following scope of requirements were developed to set clear boundaries on deliverables.

In of Scope Items:

1. Documentation for device integration into testing workflows for common tests
2. Hardware capable of collecting data from test equipment
3. User interface to interact with raw data and submit the data to a database
4. Record of organized, historical data
5. Visualization of test data stored in a database with auto-generated KPI metrics

Out of Scope Items:

1. Custom website to visualize test data results stored in a database
2. Security through data encryption
3. Predictive intelligence to estimate if rate of test data collected is on track to produce a fully validated product

1.4 Table of Symbols

Symbol	**Unit**	**Description**
A_C	m^2	coil surface area

1.5 Abbreviations and Acronyms

Symbol	**Description**
SAE	Society of Automotive Engineers
LC	Likely Change
ULC	Unlikely to Change
SRS	Software Requirements Specification
DBTL	Design Build Test Learning
KPI	Key Performance Indicators
FR	Functional Requirements
NFR	Non-functional Requirements
PC	Personal Computer
CAD	Computer Aided Design

2 User Characteristics

2.1 Stakeholders

2.2 Use Cases

2.3 User Consideration

2.4 Impact

3 System Description

3.1 Assumptions

3.2 Context Diagram

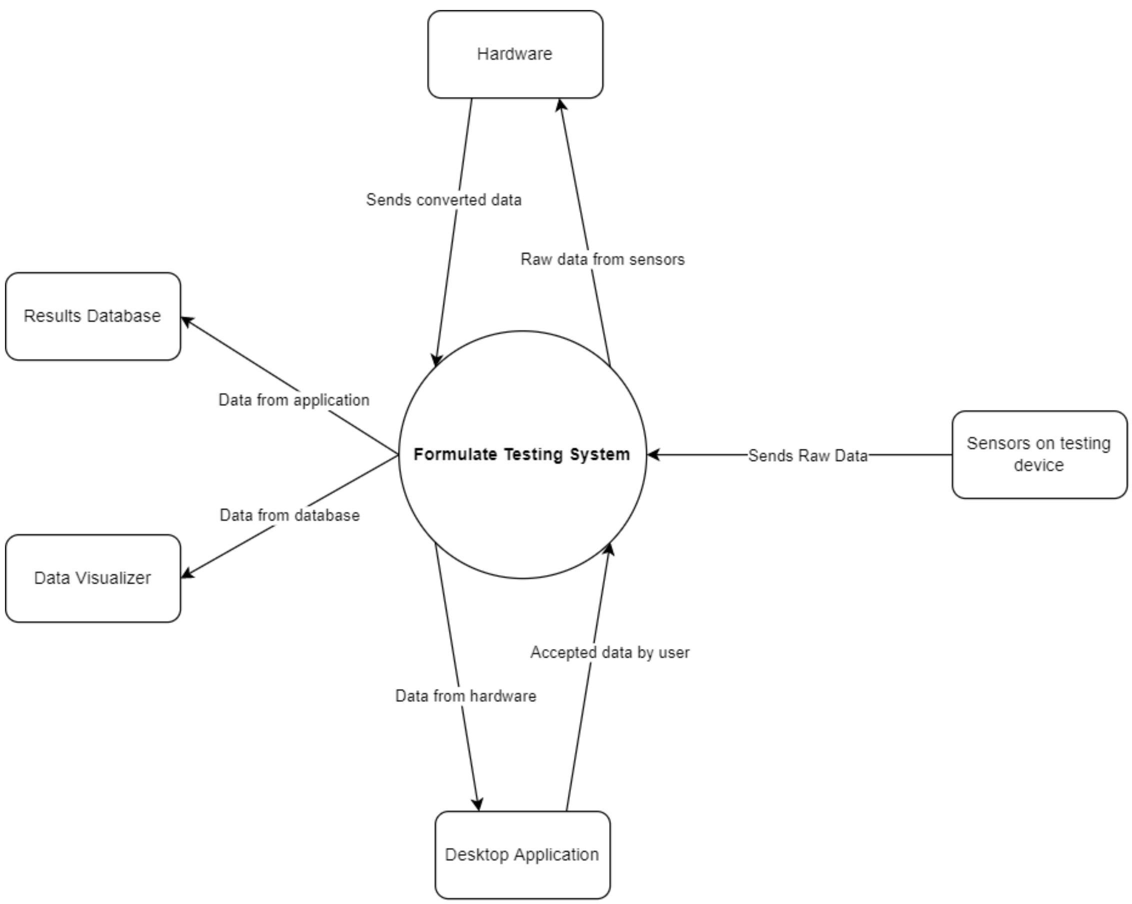

Figure 1: System Context Diagram

3.3 State Transition Diagram

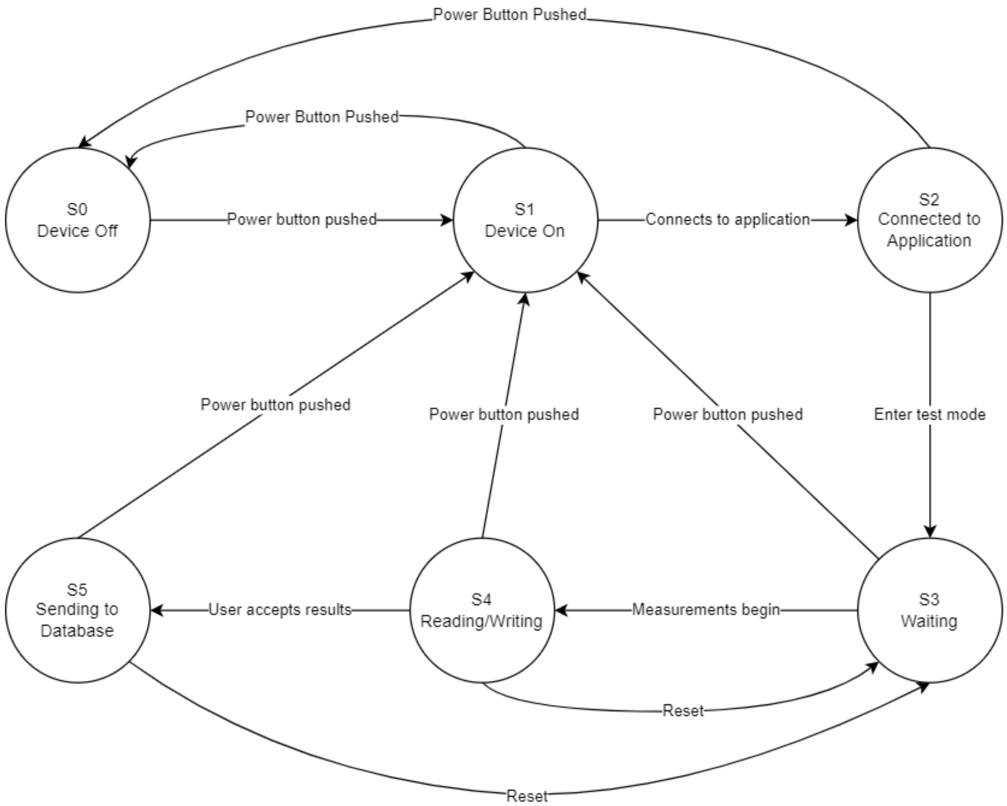

Figure 2: State Machine Diagram

State	Description
Device Off	The testing device is not powered on
Device On	The testing device is powered on
Connected to Application	The testing device has connected to the application
Waiting	Waiting for sensor to be inserted so that measurements can begin
Reading/Writing	Testing device is reading raw data from sensors, converting them, and sending them to the desktop application
Sending to Database	User has accepted the results and they are now sent to the database

Table 1.0: State descriptions

3.4 Monitored and Controlled Variables

Monitored Variable	Type	Units	Description
m_vibration	Analog	V	A signal monitoring the vibration resistance of the motor
m_humidity	Analog	V	A signal monitoring the humidity of the motor's environment
m_temperature	Analog	V	A signal monitoring the temperature of the motor's environment
m_shock	Analog	V	A signal monitoring the shock resistance of the motor
m_conv_vibration	Digital	g	Converted vibration values that are in useful units
m_conv_humidity	Digital	%	Converted humidity values that are in useful units
m_conv_temperature	Digital	°C	Converted temperature values that are in useful units
m_conv_shock	Digital	g	Converted shock values that are in useful units
m_data_accepted	Digital	T/F	Determines if user has accepted the results and wants to send it to the database

Controlled Variable	Type	Units	Description
c_green_light	Digital	1/0	Green LED light on testing device that indicates passed measurements
c_red_light	Digital	1/0	Red LED light on testing device that indicates failed measurements
c_sent_to_database	Digital	T/F	Determines if results displayed on the application are sent to the database

Constant	Units	Value	Description
k_temperature_range	°C	5-40	Acceptable ambient temperature values for a Formula Electric motor
k_humidity_range	%	5-85	Acceptable relative humidity values for a Formula electric motor
k_max_shock	g	100	Maximum shock resistance for a Formula Electric motor
k_max_vibration	g	20	Maximum vibration resistance for a Formula Electric motor

3.5 Functional Decomposition Diagram

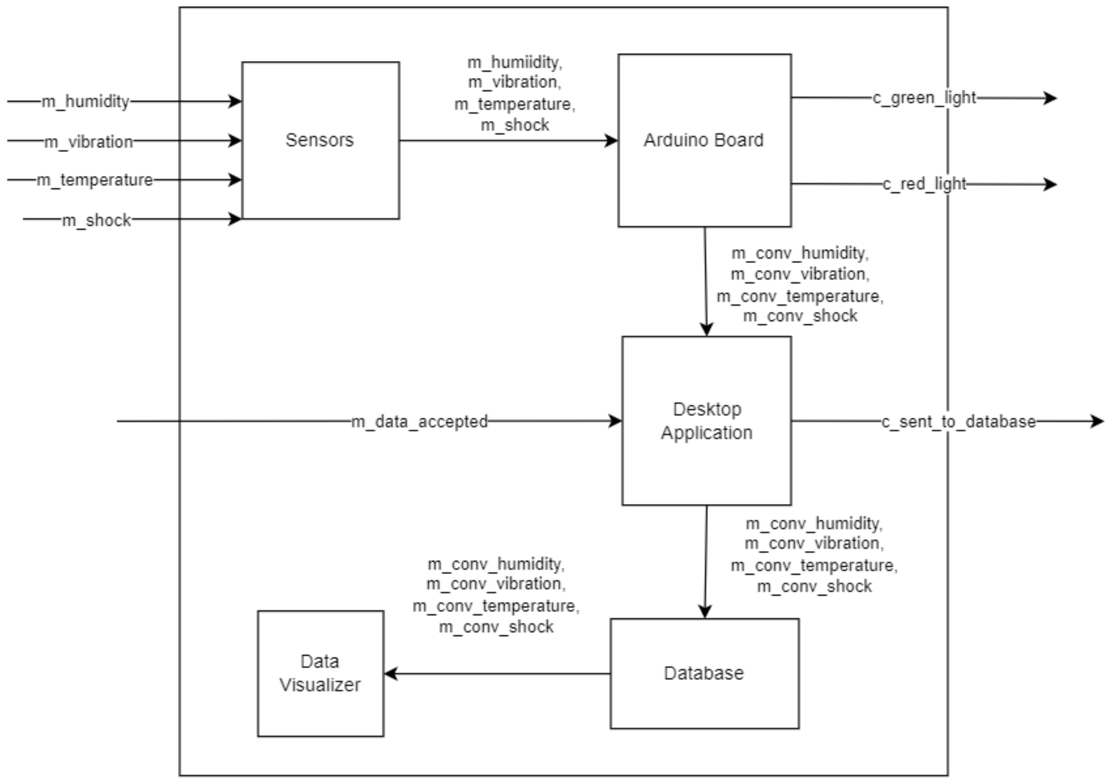

Figure 3: Functional Decomposition Diagram

4 Requirements

4.1 Functional Requirements

Formulate consists of 3 main components, each with its own functional requirements. The device addresses the sensors and physical device which interacts directly with the user. The desktop application is the means for the user to select modes and submit data, and the data visualizer (website) is for the user to view old test case data to check if KPI's are met.

4.1.1 Priority 1

FR 1: The device should be able to measure vibration, temperature, humidity, and shock

FR 2: The device should connect to a PC wired to transmit data

FR 3: The device should have a start button which activates the telemetry to start reading values between the PC and device

FR 4: The device should have a stop button which stops the telemetry stops reading values between the PC and device

FR 5: The application should allow users to preview the data after a test

FR 6: The application should allow the user to send the data to the database

FR 7: The website should be able to read the data from the database

4.1.2 Priority 2

FR 8: The device should easily mount to the Formula SAE car

FR 9: The device should contain a rechargeable battery

Rationale: The device needs its own independent power source which will allow for it to be placed in areas without a power socket

FR 10: The device should connect to a PC wirelessly to transmit data

FR 11: The device should have a screen to display the current status to the user

FR 12: The modular sensors should have a snap on mounting mechanism to connect to the base

Rationale Modular sensors need to have a rigid connection with the board with minimal movement to get the most accurate values from the sensor

FR 13: The application should show live raw data from the sensors

FR 14: The application should allow the user to configure the device's settings

Rationale: The device will need to have the wifi setting configured which will be done in the application

4.1.3 Priority 3

FR 15: The device should have 4 connection ports to add module sensors to it

Rationale Each connection port will make the device more modular and allow for users to add more sensors in the future for other tests

FR 16: The application should allow the user to trim the data before sending it to the database

FR 17: The website should only allow users who have access to view the data

FR 18: The website should have the option to filter out the data by test conducted

FR 19: The website should show whether the tests passed according to threshold values

FR 20: Any data pushed to the database should not be editable by the user

FR 21: The device should alert the user if any tests exceed the operating condition of the car

Rationale If at any point during the test it exceed operating conditions, the devices should make it obvious to the user

4.2 Nonfunctional Requirements

4.2.1 Usability

NFR1: **Ease of Learning**
The user will be able to learn the device's operation quickly to integrate into their testing workflow efficiently

NFR2: **Ease of Use**
The system will be fast at processing data such that additional overhead through the use of the device is less than if all components of the testing workflow were completed individually

4.2.2 Performance

NFR3: **Speed**
The system bandwidth will be high enough to support testing equipment with high data collection frequencies

NFR4: **Reliability and Availability**
The system will be fail-safe to withstand single point of failures in components with high probability of operational failure

4.2.3 Operational

NFR5: **Expected Technological Environment**
The device will be able to facilitate a variety of tests using a range of equipment, as long as the equipment is compatible with the data measuring hardware

NFR6: **Expected Physical Environment**
The system will be operational under a wide range of temperatures and operational vibrations

4.2.4 Maintainability and Portability

NFR7: **Maintainability**
The system will be modular and have low cohesion such that users can adapt elements of the device's hardware and software infrastructure to current needs without breaking other elements

NFR8: **Portability**
The user's ability to conduct tests will not be affected by the physical constraints from the device

4.2.5 Security

NFR9: **Software Integrity**
The system will be secure against malicious spam aimed at reducing validity of aggregate test data stored in the database

4.3 Likely Changes

LC1: Method to start and stop the test on the device

Rationale: The user should have the ability to perform a hardware start/stop or a remote start/stop from their PC

LC2: The initial data we are collecting (Vibration, Shock, Temperature, Humidity)

Rationale: Since the device is set to be modular, we may change the initial measured values we are testing with other ones

LC3: The number of modular ports on the device

Rationale: Depending on the number of input ports available on the board used, the number of input signals available may vary

4.4 Unlikely Changes

ULC1: The sensors will remain modular to adapt to different tests that need to be conducted

Rationale: The product should be expandable in the future to be able to test different values

ULC2: The communication methods between the data measurement hardware and the PC will be wired or wireless

5 Phase in Plan

The phase in plan is categorized into multiple sections, where each section represented a significant phase in the progress of project execution. A section is given a number in the hundreds (X00) to denote a significant phase in the project. Each section is subdivided further into segments given by numbers specified in the tens (XX0) to denote smaller steps within each phase. The expected order of segment completion follows the order of increasing number count; the lowest number segment should be completed first and the highest number segment should be completed last.

Each segment has an overall goal that can include the coordination of multiple team members. Upon completion of each segment, the team members relevant to the segment will review and buyoff the readyness of the segment. Upon completion of buying off each segment within a section, the overall phase is considered to be bought off and completed with confidence. The relevant stakeholders must aim to buyoff each segment in a phase before the phase deadline.

ADD SECTION RELATING PRIORITIES FROM FUNCTIONAL REQUIREMENTS TO X00 BUYOFFS
Phase 0: Preperation (000 series)
Phase 0 Deadline: October 28, 2022

000 Buyoffs	**Explanation**	**Stakeholder(s)**
010	Purchase sensor equipment, data measurement hardware, 3D print material.	Stephen
020	Obtain licenses for 3D CAD software use and database access	Stephen
030	Document material costs and licensing constraints	Stephen
040	Distribute materials and licensing to relevant project area Stakeholder	Stephen
050	Completion of device chassis mechanical design and modelling	Stephen
060	Completion of electrical connection hardware circuit design and schematic	Stephen
090	Device chassis manufactured	Stephen

Phase 1: Proof of Concept (100 series)
Phase 1 Deadline: November 11, 2022

100 Buyoffs	**Explanation**	**Stakeholder(s)**
110	Desktop application program developed with basic user interface	Stephen
120	Desktop application program can recieve data from data measurement device using a wired connection	Stephen
130	Desktop application program can interface with database to send data	Stephen
140	Desktop application program can edit data from data measurement device before sending it to the database	Stephen
150	Visualization application can pull data and generate KPI metrics from the database	Stephen
160	Integration between data measurement device and desktop application	Stephen
170	Integration between desktop application and visualization application	Stephen
190	Integration between data measurement device, desktop application, and data measurement device	Stephen

Phase 2: Revision 0 Presentation (200 series)
Phase 2 Deadline: February 3, 2023

200 Buyoffs	**Explanation**	**Stakeholder(s)**
210	Mechanical design and modelling completion of physical user interface components on device chassis and connection modules	Stephen
220	Completion of wireless communication between data measurement device and desktop application	Stephen
230	Completion of database security against tests that break utility of database	Stephen
290	Completion of extended KPI features for visualization application	Stephen

Phase 3: Final Demonstrations (300 series)
Phase 3 Deadline: March 17, 2023

6 Appendix

6.1 Knowledge Requirement

1. **Sensors/Embedded Systems:** To successfully complete this project our group needs to collectively understand how sensors and microcontrollers work. The main premise of our project is to automate testing of key components in a Formula Electric vehicle, this can only be done using a variety of sensors to monitor those components. To create almost any hardware device sensors are required as it is equivalent to being the eyes of the device, it allows for hardware to understand and map it's environment. These measurements that the sensor collects can be used by a microcontroller to make decisions on what to do according to the requirements. In our project specifically our hardware device needs to collect measurements from its environment and react accordingly with it, (i.e transmit it to a PC, alert the user if thresholds are exceeded, etc.)
2. **Project Management:** STEPHEN
3. **Databases:** MUHANAD
4. **Presentations:** TIO

6.2 Knowledge Acquirement

1. **Sensors/Embedded Systems:** There are many resources to help in mastering the skill, McMaster provides access to a platform called LinkedIn Learning, it provides useful relevant courses to programming an Arduino and how to use sensors. Other online resources such as YouTube also provides similar knowledge in these topics. The best way to learn something in my opinion is by experimenting with the technology, this is called the discovery phase. Getting familiar with the technology by messing around with it drastically improves how comfortable you are with it and how to use it. Ahmed will pursue this because he has some relevant coop experience in sensors and embedded systems and he would like to get more hands on experience working on these technologies. These skills are relevant to the type of work he wants to do in the future.
2. **Project Management:** STEPHEN
3. **Databases:** MUHANAD
4. **Presentations:** TIO

References

[The following is not part of the template, just some things to consider when filing in the template. —TPLT]
[Grammar, flow and LaTeXadvice:

- For Mac users `*.DS_Store` should be in `.gitignore`
- LaTeX and formatting rules
  - Variables are italic, everything else not, includes subscripts (link to document)
    - Conventions
    - Watch out for implied multiplication
  - Use BibTeX
  - Use cross-referencing
- Grammar and writing rules
  - Acronyms expanded on first usage (not just in table of acronyms)
  - "In order to" should be "to"

—TPLT]
[Advice on using the template:

- Difference between physical and software constraints
- Properties of a correct solution means *additional* properties, not a restating of the requirements (may be "not applicable" for your problem). If you have a table of output constraints, then these are properties of a correct solution.
- Assumptions have to be invoked somewhere
- "Referenced by" implies that there is an explicit reference
- Think of traceability matrix, list of assumption invocations and list of reference by fields as automatically generatable
- If you say the format of the output (plot, table etc), then your requirement could be more abstract

—TPLT]