

Project Title: System Verification and Validation Plan for MECHTRON 4TB6

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

November 2, 2022

1 Revision History

Date	Developer	Notes
October 30 Date 2	Stephen 1.1	Added General Information and Plan Notes

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	iv
3	General Information	1
3.1	Summary	1
3.2	Objectives	2
3.3	Relevant Documentation	2
4	Plan	3
4.1	Roadmap	3
4.2	Verification and Validation Team	4
4.3	SRS Verification Plan	5
4.4	Design Verification Plan	5
4.5	Implementation Verification Plan	6
4.6	Automated Testing and Verification Tools	7
4.7	Software Validation Plan	7
5	System Test Description	8
5.1	Tests for Functional Requirements	8
5.1.1	Hardware Device	8
5.1.2	Desktop Application	11
5.1.3	Data Analytics Website	13
5.1.4	Database	14
5.2	Tests for Nonfunctional Requirements	14
5.2.1	Performance	14
5.2.2	Usability	16
5.3	Traceability Between Test Cases and Requirements	18
6	Unit Test Description	19
6.1	Unit Testing Scope	19
6.2	Tests for Functional Requirements	19
6.2.1	Module 1	19
6.2.2	Module 2	19
6.3	Tests for Nonfunctional Requirements	20
6.3.1	Module ?	20
6.3.2	Module ?	20

6.4	Traceability Between Test Cases and Modules	20
7	Appendix	21
7.1	Symbolic Parameters	21
7.2	Usability Survey Questions?	21

List of Tables

List of Figures

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test

This document ...

3 General Information

3.1 Summary

Formulate consists of four subsystems, one hardware subsystem and three software subsystems, which interact to provide the user with a testing device designed to eliminate automatable processes in common testing procedures.

A physical data collection device is the hardware subsystem used as the first point of contact with the measured quantity through a sensor. The sensor obtains physical quantities for the device to buffer, before sending the data to a desktop application for user verification.

The user has the ability to view the data collected by the physical device after a completed test using a desktop application software subsystem. The desktop application enables the user to either accept the test results to then store a collection of data from a test to a database, or reject the test and prevent the test from being stored to a database.

Accepted test data sent from the desktop application aggregates and saves verified test results to a database software subsystem. Users can query the database to obtain common statistics in test data and generate new or obscure relationships by leveraging database language capabilities.

A final dashboard software subsystem then queries the database to visualize key performance indicators on the test data collected and stored in the database.

3.2 Objectives

The objective of the system Verification and Validation (VnV) plan for Formulate is to ensure the intended project qualities are present.

Ease in user understanding is a quality Formulate will achieve to support the system's usability. Specifically, ease in user understanding of each subsystem's function and how subsystems interface will be key qualities of the overall project.

The system will also demonstrate the quality of adequate portability and physical robustness to support system maintainability, portability, and operability.

3.3 Relevant Documentation

Talk about how this document draws from the system requirements gathered during software requirements specification (SRS) and hazard analysis.

This document references a variety of requirements generated during the Software Requirements Specification (SRS) process and the Hazard Analysis (HA) process for the Formulate system.

[Author \(2019\)](#)

4 Plan

4.1 Roadmap

The intention of testing for Formulate is to generate confidence that the project meets qualities relating to usability, maintainability, portability, operability, and safety set out as requirements in SRS and HA documentation. Through sets of unit and system tests that will prove if the system has met the above requirements, Formulate will understand if the project has achieved the desired qualities.

Specifically, requirements that are functional, non-functional, and safety-security related from the SRS and HA documents will be referenced in the Plan, System Test Description, and Unit Test Description sections of this document.

4.2 Verification and Validation Team

Name	Role	Explanation
Stephen	Desktop Application Tester	VnV for software application design and integration with hardware and database
Ahmed	Hardware Device Tester	VnV for embedded program design, chassis design, and integration with desktop application
Muhanad	Database Application Tester	VnV for database design and integration with desktop application and dashboard
Tioluwalayomi	Dashboard Application Tester	VnV for dashboard application design and integration with database
Timofey	Project and Course Teaching Assistant	Detailed low level feedback on planned VnV tests
Dr. Smith	Course Instructor	General high level feedback on planned VnV tests

4.3 SRS Verification Plan

SRS verification will be composed of two approaches to verify that functional and non-functional requirements are met. The first approach is engaging in read through's of the SRS document each month. Individual member progress will be evaluated against the relevant sections(s) of the SRS document to ensure system development is on track to meet the requirements. The second approach is evaluating issues created by classmates on GitHub and incorporating their concerns and suggestions as seen fit.

Stephen and Tioluwalayomi will lead the group wide discussion for SRS verification activities on the first Tuesday of each month.

4.4 Design Verification Plan

Design verification will be composed of two approaches. The first planned approach is completing read through's of each individual's high level design documentation for their respective sub-system. The design documentation covered during these read through's will likely entail mechanical, electrical, and or software schematics or diagrams outlining the architecture of the subsystem. Members will voice concerns during the read through of design decisions made by the individual responsible for the sub-system architecture. The second planned approach is evaluating issues created by classmates on GitHub and incorporating their concerns and suggestions as seen fit.

Ahmed will lead the group wide discussion for design verification activities on the second Tuesday of each month.

4.5 Implementation Verification Plan

Implementation verification will be composed of techniques in both static and dynamic analysis.

Content walkthrough is the primary type of static implementation verification technique the group plans on using. Three similar types of content walkthrough's are planned for use depending on the sub-system under analysis. Software implementation's will receive a code walkthrough, mechanical implementation's will receive a Computer Aided Design (CAD) spin, and electrical implementation's will receive a schematic walkthrough. During the content walkthrough, unit and system tests relevant to the implementation will be considered to critique the quality of the implementation. A meeting will be organized for each content walkthrough once the implementation has reached a notable milestone worthwhile for group analysis.

Live execution of the implementation using a proof of concept style demonstration to the group is the primary type of dynamic implementation technique the group plans on using. During the live demonstration of the implementation, system and unit tests relevant to the implementation will be considered to critique the quality of the implementation. Using the initial state and inputs of the test outlined in the system and unit test sections, the quality of implementation is passed or failed depending on if the actual output of the implementation matches the expected output.

Ahmed and Muhanad will lead the group wide discussion for design verification activities on the second Tuesday of each month.

4.6 Automated Testing and Verification Tools

Tools will not be used to automate testing, profile, or code coverage for software flashed on embedded hardware because of the available tools incur high additional overhead to testing effort and times. The group plans on completing extensive unit testing for embedded software to compensate for the absence of automated testing and coverage tools.

The desktop application will likely use Visual Studio's memory usage tool to profile the program during execution.

PyLint and SQLFluff will be used as the static code analysis tools to support uniformity in the desktop application and database programs respectively.

Code coverage will be completed manually for the desktop, database, and the dashboard programs. This decision will be feasible as the expected size of software for the applications listed above is relatively small. As a result, it is reasonable for to manually check the amount of code coverage achieved through tests.

4.7 Software Validation Plan

There are no current plans to use external data for validation.

5 System Test Description

5.1 Tests for Functional Requirements

5.1.1 Hardware Device

Sensor Validation

1. ST-SV 1

Type: Manual

Initial State: Device is on measuring and sending values to the application, and connection to database has been verified

Input: The device is placed in temperature controlled box at 25°C

Output: The temperature sensor on the device should consistently read 25°C

How test will be performed: The device will be placed in a styrofoam insulated box and a hair dryer will be used to heat the box. Using a digital thermometer we will compare the results of our device to the digital thermometer

2. ST-SV 2

Type: Manual

Initial State: Device is on measuring and sending values to the application, and connection to database has been verified

Input: The device will be placed in an humidity controlled box which will be around 40% humidity

Output: The humidity sensor on the device should consistently read 40% humidity

How test will be performed: The device will be placed in a styrofoam insulated box and a hair dryer will be used to heat the box. Using a digital humidity monitor we will compare the results of our device to the digital sensor

Device Telemetry

1. ST-DT 1

Type: Manual

Initial State: Device is on and is connected to a PC via cable or via Wi-Fi

Input: The start button is pressed

Output: All the readings from all the different sensors should be sent to the PC

How test will be performed: While the device is connected to the PC the start button will be pressed to check if telemetry data is being sent to the PC

2. ST-DT 2

Type: Manual

Initial State: Device is on and is connected to a PC via cable or via Wi-Fi

Input: The start button is pressed multiple times

Output: The device should continue to send data to the PC regardless of how many times the start button is pressed

How test will be performed: While the device is connected to the PC the start button will be pressed and then after a few seconds pressed again 2 more times to check if telemetry data is still being sent to the PC

3. ST-DT 3

Type: Manual

Initial State: Device is on and is connected to a PC via cable or via Wi-Fi

Input: The stop button is pressed

Output: All the readings from all the different sensors should stop being sent to the PC

How test will be performed: While the device is connected to the PC the stop button will be pressed to check if telemetry data is stopped

4. **ST-DT 4**

Type: Manual

Initial State: Device is on and is connected to a PC via cable or via Wi-Fi

Input: The stop button is pressed multiple times

Output: The device should not send data to the PC regardless of how many times the stop button is pressed

How test will be performed: While the device is connected to the PC the stop button will be pressed in an interval of 2 seconds to check if telemetry data is has stopped

Device Hardware

1. **ST-DH 1**

Type: Manual

Initial State: Device is on and is connected to a PC, no sensors are attached

Input: A sensor is connected and start is pressed

Output: The device should start sending data to the PC

How test will be performed: While the device is connected to the PC a sensor is connected to the modular port on the device and the start button is pressed

2. **ST-DH 2**

Type: Manual

Initial State: Device is on but is low on battery

Input: User plugs the battery in with a charger

Output: The battery should begin to charge

How test will be performed: We will continue to use the device until the battery is completely drained, once its low we will start to charge it and check if the battery increases

3. **ST-DH 3**

Type: Manual

Initial State: Device is fully mounted on a test platform

Input: User will apply 49N of force on every side of the device

Output: The device should not move at all

How test will be performed: Once the device is mounted we will place 5kg on top of different sides of the device to check if the mount can withstand it

4. **ST-DH 4**

Type: Manual

Initial State: Device is not mounted on anything

Input: User will begin to mount the device on the Formula SAE car

Output: The device should be fully installed under 5 minutes

How test will be performed: While someone is mounting the device another person is timing them

5.1.2 Desktop Application

1. **ST-DA 1**

Type: Manual

Initial State: Device is on and is connected to a PC, the desktop application is open and connected to the device

Input: User selects the most recent test case

Output: A table with all the raw test data should populate the screen

How test will be performed: After a test is completed on the device, it will be connected to the PC and the user should be able to view the

most recent test after clicking view

2. **ST-DA 2**

Type: Manual

Initial State: Device is on and is connected to a PC, the desktop application is open and test data is open

Input: User selects rows and columns to delete and hits the delete key

Output: The table with the test data should update to reflect the newly deleted rows

How test will be performed: A test will be conducted but the start button will be pressed early, the initial 20 seconds of the test does not provide useful data and the user will select and delete those rows

3. **ST-DA 3**

Type: Manual

Initial State: Device is on and is connected to a PC, the desktop application is open and test data is open

Input: User hits the submit to database button

Output: The test data in the table should be sent to the database

How test will be performed: After a test is conducted and the data will be viewed on the desktop app, the user will preview the data and click submit to database

4. **ST-DA 4**

Type: Manual

Initial State: Device is on and is connected to a PC, the desktop application is open but no tests are done

Input: User hits the submit to database button

Output: The application should alert the user that no data was recorded and not send anything to the database

How test will be performed: No tests will be performed and an empty table will be shown. Submit to database will be clicked

5. **ST-DA 5**

Type: Manual

Initial State: Device is on and is connected to a PC, the desktop application is open

Input: User select live data viewing on the application

Output: The application should start to receive live data from the device

How test will be performed: The hardware device will be plugged into the PC and once we hit the live data button on the application all the sensors will display their output

5.1.3 Data Analytics Website

1. **ST-DAW 1**

Type: Manual

Initial State: The database contains test data and is connected to the analytics platform

Input: User logs into the analytics platform and selects a test

Output: The test data is displayed showing the results over time for that particular test

How test will be performed: We will populate the database with different tests and the user once logged in will select a specific test to view

2. **ST-DAW 2**

Type: Manual

Initial State: The database contains test data and is connected to the analytics platform

Input: Unauthorized user attempts to login

Output: The website alerts the user they are not authorized to view the dashboard

How test will be performed: One team member will not have privileges to view the dashboard and will attempt to login

5.1.4 Database

1. ST-D 1

Type: Manual

Initial State: Database is active

Input: User User submits data multiple times within 5 seconds

Output: Database stops accepting new data and alerts user

How test will be performed: After a test is completed on the device, it will be previewed using the desktop app and the submit to database button will be pressed repeatedly

5.2 Tests for Nonfunctional Requirements

5.2.1 Performance

Operational in physical environment

1. Operational in physical environment

Type: Dynamic, Manual

Initial State: Device is on and mounted to the device, has connected to the application and is waiting to start measuring.

Input/Condition: Vehicle's motor starts and values start to get picked up by device

Output/Result: Device is operational and stays physically intact in all types of weather and at 20% greater than threshold values.

How test will be performed: The device will be tested outdoors under various weather conditions including rain, windy, etc. The device will

also be tested in temperature and vibration conditions that are above threshold values. This will be performed by placing the device in a hot environment and vigoursly shaking it while being on a stationary mount.

2. Viewing live data

Type: Dynamic, Manual

Initial State: Device is on and mounted to the device, has connected to the application and is waiting to start measuring.

Input: Vehicle's motor starts and values start to get picked up by device

Output/Result: Data latency should be less than 30 seconds to simulate viewing live data.

How test will be performed: The amount of time for data to start being viewable on the application will be inspected to be less than 30 seconds. The application will also be inspected to ensure that data is smooth and not lagging while measurements are being performed.

3. Modularity and Maintainability

Type: Dynamic, Manual

Initial State: Device is on measuring and sending values to the application, and connection to database has been verified

Input: Either the device, application, or database is disconnected or turned off

Output: The other two components are still functional even though communication between them is broken.

How test will be performed: While device, application, and database are fully functional and communicating successfully, different combinations of either one or two components will be turned off. The other component(s) will be inspected to ensure that they are operational and indicating that the other component(s) are disconnected.

4. test-id2

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input:

Output:

How test will be performed:

5. test-id2

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input:

Output:

How test will be performed:

5.2.2 Usability

1. Mounting hardware and starting measurements

Type: Dynamic, Manual

Initial State: Device is turned off and nothing is connected, only the application is loaded on to the computer

Input: Users will be asked to setup device and start taking measurements, rate setup process using a survey

Output: Time for setup and data to appear on the application should be less than 5 minutes and

How test will be performed: A test group will be educated on the setup and connection of the device, then they will attempt to do that process. Each person will be timed and compared to the 5 minute threshold. In addition, they will be given a survey to rate the setup process on a scale from 1 to 5 the following categories: ease of use, need for assistance,

2. test-id2

Type: Dynamic, Manual

Initial State: Device is given to McMaster's Formula E team to use

Input: Using a survey, Formula E members will compare their current testing process to the Formulate process

Output: All users need to select Formulate in at least 2 of the 3 categories

How test will be performed: Formula E members will select which process is preferred in the following categories: speed, data collection, ease of use

3. test-id2

Type: Dynamic, Manual

Initial State:

Input:

Output:

How test will be performed:

5.3 Traceability Between Test Cases and Requirements

Requirement	Test
FR 1	ST-SV 1, ST-SV 2
FR 2	ST-DT 1, ST-DT 2, ST-DT 3, ST-DT 4
FR 3	ST-DT 1, ST-DT 2
FR 4	ST-DT 3, ST-DT 4
FR 5	ST-SV 1, ST-SV 2
FR 6	ST-DA 1
FR 7	ST-DA 3 ST-DA 4
FR 8	ST-DAW 1
FR 9	ST-DH 3
FR 10	ST-DH 4
FR 11	ST-DH 2
FR 12	ST-DT 1, ST-DT 2
FR 13	
FR 14	
FR 15	ST-DA 5
FR 16	
FR 17	ST-DA 1
FR 18	ST-DA 2
FR 19	ST-DAW 2
FR 20	ST-DAW 1
FR 21	ST-DAW 1
FR 22	ST-D 1
FR 23	

6 Unit Test Description

6.1 Unit Testing Scope

6.2 Tests for Functional Requirements

6.2.1 Module 1

1. test-id1

Type:

Initial State:

Input:

Output:

Test Case Derivation:

How test will be performed:

2. test-id2

Type:

Initial State:

Input:

Output:

Test Case Derivation:

How test will be performed:

3. ...

6.2.2 Module 2

...

6.3 Tests for Nonfunctional Requirements

6.3.1 Module ?

1. test-id1

Type:

Initial State:

Input/Condition:

Output/Result:

How test will be performed:

2. test-id2

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input:

Output:

How test will be performed:

6.3.2 Module ?

...

6.4 Traceability Between Test Cases and Modules

References

Author Author. System requirements specification. <https://github.com/...>, 2019.

7 Appendix

This is where you can place additional information.

7.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

7.2 Usability Survey Questions?