# 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

October 31, 2022

# 1 Revision History

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

# Contents

# 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 operationality.

## 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, operationality, 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 and Visualization 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 relevenat 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 will lead the group wide discussion for SRS verification activities on the first Tuesday of each month.

## 4.4 Design Verification Plan

## 4.5 Implementation Verification Plan

## 4.6 Automated Testing and Verification Tools

## 4.7 Software Validation Plan

# 5 System Test Description

## 5.1 Tests for Functional Requirements

### 5.1.1 Area of Testing1

**Title for Test**

1. test-id1

   Control: Manual versus Automatic

   Initial State:

   Input:

   Output:

   Test Case Derivation:

   How test will be performed:

2. test-id2

   Control: Manual versus Automatic
   Initial State:
   Input:
   Output:
   Test Case Derivation:
   How test will be performed:

### 5.1.2 Area of Testing2

...

## 5.2 Tests for Nonfunctional Requirements

### 5.2.1 Area of Testing1

**Title for Test**

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:

#### 5.2.2 Area of Testing2

...

### 5.3 Traceability Between Test Cases and Requirements

## 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?