System Verification and Validation Plan for Stoichiometry Mass-Mass Program

Deemah Alomair

October 28, 2019

1 Revision History

Date	Version	Notes
28/10/2019	1.0	First version of document

Contents

1 Revision History i

2 Symbols, Abbreviations and Acronyms iii

3 General Information 1

3.1 Summary . . . 1

3.2 Objectives . . . 1

3.3 Relevant Documentation . . . 1

4 Plan 1

4.1 Verification and Validation Team . . . 1

4.2 SRS Verification Plan . . . 2

4.3 Design Verification Plan . . . 2

4.4 Implementation Verification Plan . . . 2

4.5 Software Validation Plan . . . 2

5 System Test Description 2

5.1 Tests for Functional Requirements . . . 3

5.1.1 Input Tests . . . 3

5.1.2 Output Tests . . . 4

5.2 Tests for Nonfunctional Requirements . . . 4

5.2.1 Usability . . . 4

5.2.2 Reliabilty . . . 5

5.2.3 Maintainability . . . 5

5.3 Traceability Between Test Cases and Requirements . . . 5

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test
SRS	Software Requirements Specification
SMMP	Stoichiometry Mass-Mass Program
R	Functional requirement
NF	Non-functional requirement

3 General Information

This document is to build verification and validation plan for Stoichiometry Mass-Mass Program and provide description about the testing that will be carried out on the system. The main goal of this document is to check whether SMMP meets the SRS and fulfill its intended purpose. This document will be used as the reference and guidance for testing SMMP.

3.1 Summary

Stoichiometry Mass-Mass Program is a software that convert unbalanced chemical reaction to balanced one. Then do some necessary calculations to get the mass of one of the reactants involved in that chemical reaction.

3.2 Objectives

The objectives of the verification and validation plan is to ensure that SMMP is Reliable. In most cases it provides correct answer and behave as intended. In addition, usability because this software may be used by chemistry student who has less experiences in dealing with computer software. Maintainability is also need to be achieved because this is the first draft of the software and continues changes may be done in next versions.

3.3 Relevant Documentation

Relevant Documents that need to be visited while reading this document include the following:

SRS, which can be found here

4 Plan

4.1 Verification and Validation Team

The test team has one member: Deemah Alomair

4.2 SRS Verification Plan

SRS Verification Plan will include a feedback from the domain expert, secondary reviewer and Dr.Smith.

4.3 Design Verification Plan

Design Verification Plan will include a feedback from the domain expert, secondary reviewer and Dr.Smith.

4.4 Implementation Verification Plan

Implementation Verification Plan will include the followings:

- Dynamic and automatic testing that includes unit testing for each individual function and coverage testing.
- Linters will be considered throughout the implementation of SMMP.
- Parallel testing.
- Code walkthrough for SMMP will be carried out after the implementation by the domain expert and Dr.Smith.

4.5 Software Validation Plan

N.A

5 System Test Description

This section lists the system tests to be performed to verify whether or not the program fulfills the functional requirements and to test how well it meets the non-functional requirements. Note that some of the tests for the functional requirements are unit tests.

5.1 Tests for Functional Requirements

5.1.1 Input Tests

Input constrains tests

1. **T1: Mass value test**

   Control: Functional, dynamic, automatic.

   Initial State: Not Applicable.

   Input: -7.

   Output: Error message indicates that entered mass value should be greater than 0.

   Test Case Derivation: The expected mass value is greater than 0.

   How test will be performed: Automatic unit testing.

2. **T2: Mass format test**

   Control: Functional, dynamic, automatic.

   Initial State: Not Applicable.

   Input: H_2O.

   Output: Error message indicates that entered mass value should be numeric.

   Test Case Derivation: The expected entered mass value is a positive number.

   How test will be performed: Automatic unit testing.

3. **T3: Chemical reaction format test**

   Control: Functional, dynamic, automatic.

   Initial State: Not Applicable.

   Input: $H_2O = 2$.

   Output: Error message indicates that this is not a correct chemical reaction format.

   Test Case Derivation: The expected entered chemical reaction is set of reactants and products like: $NaOH + H_2SO_4 \rightarrow H_2O + NA_2SO_4$

   How test will be performed: Automatic unit testing.

5.1.2 Output Tests

Output constrains tests

1. **T4: Result test**

   Control: Functional, dynamic, automatic.

   Initial State: Not Applicable.

   Input: chemical reaction : $NaOH + H_2SO_4 \rightarrow H_2O + NA_2SO_4$
   mass of known reactant : 3.10 g.
   name of known reactant: H_2SO_4

   Output: mass = 2.53 g.

   Test Case Derivation: The expected value of the mass or error message otherwise.

   How test will be performed: coverage testing , parallel testing .

2. **T5: Chemical reaction balancing test**

   Type: Functional, dynamic, automatic.

   Initial State: Not Applicable.

   Input: $N_2 + 3H_2 \rightarrow NH_3$

   Output: $N_2 + 3H_2 \rightarrow 2NH_3$.

   Test Case Derivation: The expected balance equation or error message otherwise.

   How test will be performed: Parallel testing.

5.2 Tests for Nonfunctional Requirements

5.2.1 Usability

1. **T6: Usability testing**

   Type: Nonfunctional,Dynamic, Manual.

   Initial State: Not Applicable.

Input/Condition: Unbalanced chemical reaction/ mass of one reactant.

Output/Result: Second reactant mass.

How test will be performed: Ask domain expert to use the system.

5.2.2 Reliabilty

1. **T7: Reliability testing**

   Type: Nonfunctional,Dynamic, Manual.

   Initial State: Not Applicable.

   Input/Condition: Unbalanced chemical reaction/ mass of one reactant.

   Output/Result: Second reactant mass.

   How test will be performed: using the system with different inputs and measuring the percentage of correct answers .

5.2.3 Maintainability

1. **T8: Maintainability testing**

   Type: Nonfunctional,Dynamic, Manual.

   Initial State: Not Applicable.

   Input/Condition: Existing SMMP system.

   Output/Result: New version of SMMP.

   How test will be performed: Try to add new feature or fix some constrains and observe system reaction.

5.3 Traceability Between Test Cases and Requirements

A trace between system tests and requirements is provided in Table 1.

	T1	T2	T3	T4	T5	T6	T7	T8
R1	X	X	X					
R2				X	X			
R3					X			
R4	X	X	X	X	X			
NF1				X	X		X	
NF2	X	X	X	X	X	X		
NF3						X		
NF4								X

Table 1: Traceability Matrix Showing the Connections Between Requirements and system tests