Test Report: Stoichiometry Mass-Mass Program

Deemah Alomair

December 26, 2019

1 Revision History

Date	**Version**	**Notes**
26/12/2019	1.0	First version of the document

2 Symbols, Abbreviations and Acronyms

symbol	**description**
T	Test
SMMP	Stoichiometry Mass-Mass Program

Contents

List of Tables

List of Figures

This document report result from system test cases found in System Verification and Validation Plan document (1).

3 Functional Requirements Evaluation

Functional requirements are evaluated using system test cases from id1 to id10. All the details regarding functional requirements evaluation can be found in section 6. The traceability between system test cases and functional requirements found in table 2 of section 9.

4 Nonfunctional Requirements Evaluation

Nonfunctional Requirements evaluated in system test cases id11 - id12. All the details regarding Nonfunctional requirements evaluation can be found in this section. The traceability between system test cases and Nonfunctional requirements found in table 3 of section 9.

4.1 Usability

system test id11 measures the Usability of SMMP system. The survey measured the satisfaction level of potential user after using SMMP. How easy and understandable the system is?. It needs to be filled by any user and if satisfactory level is low then enhancement need to be taken into account. survey can be found in Appendix of (1).

4.2 Reliability

system test id12 measures the Reliability of SMMP system. 25 different unbalanced chemical reactions were tested and compared to answer of online balancer (2) as parallel testing. the aim was to get 100% of correct answers that includes right balance reaction and correct mass value. Below table shows the final result compared to the online balancer and the final percentage of correctness. The goal of Reliability had been accomplished. all tested reactions were balanced correctly as compared to the online balancer.

Unbalanced Chemical Reaction	Online Balancer Result	SMMP Result	correctness
$CH_4 + O_2 \rightarrow CO_2 + H_2O$	$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O$	$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O$	correct
$Fe_2O_3 + C \rightarrow Fe + CO_2$	$2Fe_2O_3 + 3C \rightarrow 4Fe + 3CO_2$	$2Fe_2O_3 + 3C \rightarrow 4Fe + 3CO_2$	correct
$N_2 + O_2 \rightarrow N_2O_5$	$2N_2 + 5O_2 \rightarrow 2N_2O_5$	$2N_2 + 5O_2 \rightarrow 2N_2O_5$	correct
$CH_4 + Cl_2 \rightarrow CCl_4 + HCl$	$CH_4 + 4Cl_2 \rightarrow CCl_4 + 4HCl$	$4CH_4 + Cl_2 \rightarrow CCl_4 + 4HCl$	correct
$N_2 + H_2 \rightarrow NH_3$	$N_2 + 3H_2 \rightarrow 2NH_3$	$N_2 + 3H_2 \rightarrow 2NH_3$	correct
$Fe + H_2O \rightarrow Fe_3O_4 + H_2$	$3Fe + 4H_2O \rightarrow Fe_3O_4 + 4H_2$	$3Fe + 4H_2O \rightarrow Fe_3O_4 + 4H_2$	correct
$Xe + F_2 \rightarrow XeF_6$	$Xe + 3F_2 \rightarrow XeF_6$	$Xe + 3F_2 \rightarrow XeF_6$	correct
$Hg + O_2 \rightarrow HgO$	$2Hg + O_2 \rightarrow 2HgO$	$2Hg + O_2 \rightarrow 2HgO$	correct
$CaO + C \rightarrow CaC_2 + CO$	$CaO + 3C \rightarrow CaC_2 + CO$	$CaO + 3C \rightarrow CaC_2 + CO$	correct
$S_8 + F_2 \rightarrow SF_6$	$S_8 + 24F_2 \rightarrow 8SF_6$	$S_8 + 24F_2 \rightarrow 8SF_6$	correct
$Mg + N_2 \rightarrow Mg_3N_2$	$3Mg + N_2 \rightarrow Mg_3N_2$	$3Mg + N_2 \rightarrow Mg_3N_2$	correct
$BeF_2 + Mg \rightarrow MgF_2 + Be$	balance	balance	correct
$Zn + HCl \rightarrow ZnCl_2 + H_2$	$Zn + 2HCl \rightarrow ZnCl_2 + H_2$	$Zn + 2HCl \rightarrow ZnCl_2 + H_2$	correct
$SiC + Cl_2 \rightarrow SiCl_4 + C$	$SiC + 2Cl_2 \rightarrow SiCl_4 + C$	$SiC + 2Cl_2 \rightarrow SiCl_4 + C$	correct
$MnS + HCl \rightarrow H_2S + MnCl_2$	$MnS + 2HCl \rightarrow H_2S + MnCl_2$	$MnS + 2HCl \rightarrow H_2S + MnCl_2$	correct
$UF_4 + Mg \rightarrow MgF_2 + U$	$UF_4 + 2Mg \rightarrow 2MgF_2 + U$	$UF_4 + 2Mg \rightarrow 2MgF_2 + U$	correct
$S + N_2O \rightarrow SO_2 + N_2$	$S + 2N_2O \rightarrow SO_2 + 2N_2$	$S + 2N_2O \rightarrow SO_2 + 2N_2$	correct
$SiH_4 + O_2 \rightarrow SiO_2 + H_2O$	$SiH_4 + 2O_2 \rightarrow SiO_2 + 2H_2O$	$SiH_4 + 2O_2 \rightarrow SiO_2 + 2H_2O$	correct
$TiCl_4 + Mg \rightarrow MgCl_2 + Ti$	$TiCl_4 + 2Mg \rightarrow 2MgCl_2 + Ti$	$TiCl_4 + 2Mg \rightarrow 2MgCl_2 + Ti$	correct
$Si + S_8 \rightarrow Si_2S_4$	$4Si + S_8 \rightarrow 2Si_2S_4$	$4Si + S_8 \rightarrow 2Si_2S_4$	correct
$SiO_2 + HF \rightarrow SiF_4 + H_2O$	$SiO_2 + 4HF \rightarrow SiF_4 + 2H_2O$	$SiO_2 + 4HF \rightarrow SiF_4 + 2H_2O$	correct
$P_4 + O_2 \rightarrow P_2O_5$	$P_4 + 5O_2 \rightarrow 2P_2O_5$	$P_4 + 5O_2 \rightarrow 2P_2O_5$	correct
$Sb + O_2 \rightarrow Sb_4O_6$	$4Sb + 3O_2 \rightarrow Sb_4O_6$	$4Sb + 3O_2 \rightarrow Sb_4O_6$	correct
$UO_2 + HF \rightarrow UF_4 + H_2O$	$UO_2 + 4HF \rightarrow UF_4 + 2H_2O$	$UO_2 + 4HF \rightarrow UF_4 + 2H_2O$	correct
$Al + O_2 \rightarrow Al_2O_3$	$4Al + 3O_2 \rightarrow 2Al_2O_3$	$4Al + 3O_2 \rightarrow 2Al_2O_3$	correct

Table 1: Reliability testing of SMMP as comparison to online balancer.

5 Comparison to Existing Implementation

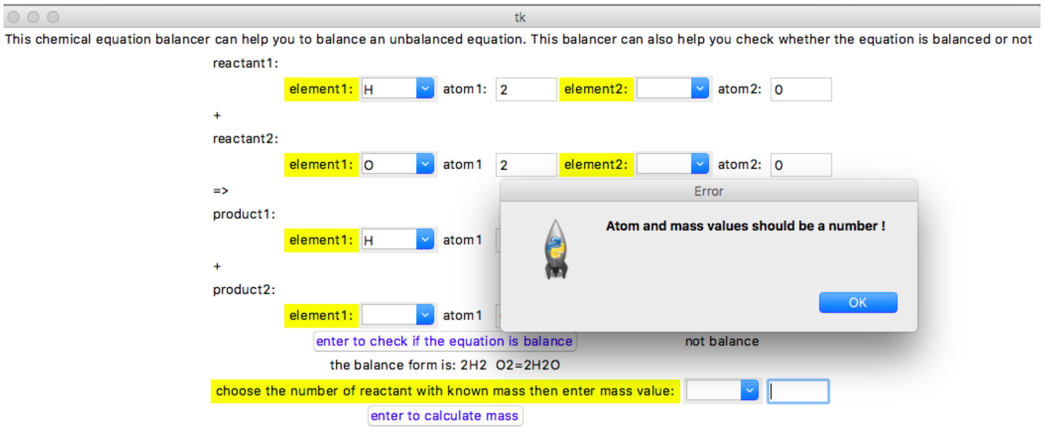
This is stand alone system. If there is any existing systems with same functionalities and goals the developer is not aware about them and no comparison had been made.

6 Unit Testing

6.1 Mass Input Test-id1

The goal of this test is to make sure user enter a positive number in mass value widget. the default mass value is 1. pictures below illustrate all possible entries by user that cause error massage which matches cases of id1 ,Table1 in

(1). The tests had been passed correctly and system responded as intended.



This chemical equation balancer can help you to balance an unbalanced equation. This balancer can also help you check whether the equation is balanced or not

reactant1:

element1: H atom1: 2 element2: atom2: 0

+

reactant2:

element1: O atom1 2 element2: atom2: 0

=>

product1:

element1: H atom1

+

product2:

element1: atom1

Error

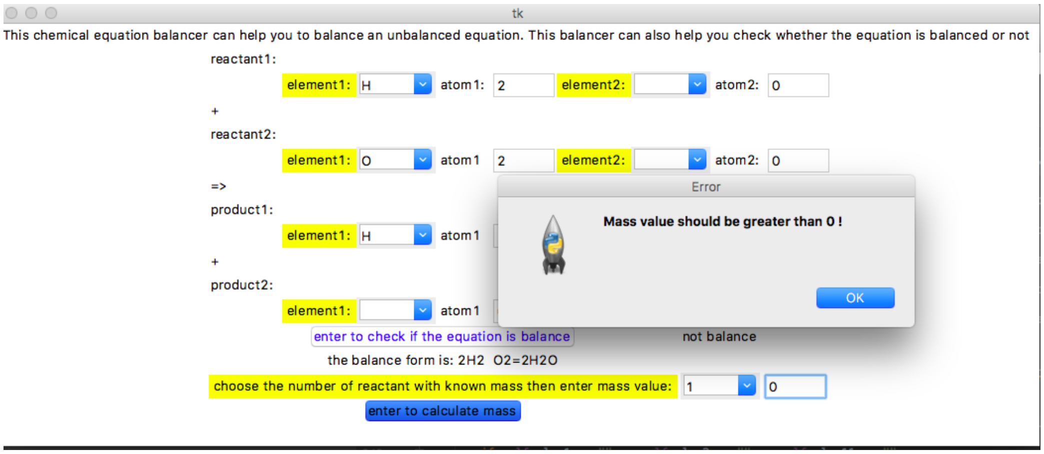
Atom and mass values should be a number !

not balance

the balance form is: 2H2 O2=2H2O

choose the number of reactant with known mass then enter mass value:

Figure 1: error massage if user enter non number mass value



This chemical equation balancer can help you to balance an unbalanced equation. This balancer can also help you check whether the equation is balanced or not

reactant1:

element1: H atom1: 2 element2: atom2: 0

+

reactant2:

element1: O atom1 2 element2: atom2: 0

=>

product1:

element1: H atom1

+

product2:

element1: atom1

Error

Mass value should be greater than 0 !

not balance

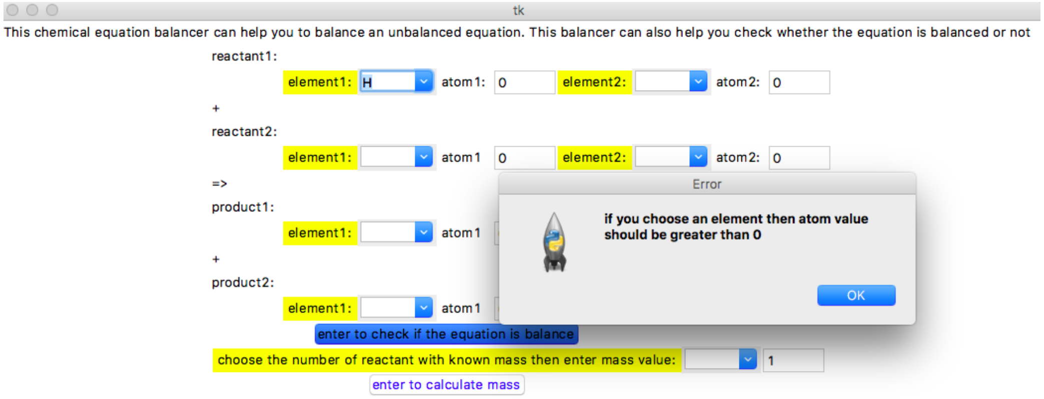
the balance form is: 2H2 O2=2H2O

choose the number of reactant with known mass then enter mass value: 1 0

Figure 2: error massage if user enter mass less than 1

6.2 Reaction Input Test-id2

The goal of this test is to ensure that user had entered a correct chemical reaction format that satisfies the system requirements and enter a positive number for atom value. More details of accepted reaction format can be found in Unit Verification and Validation Plan (3) and Unit Verification and Validation Test Report (4). Test case id2 testing entered atom value. pictures bellow show all possible entires of atom value and system responds to each case. The test passed with right system behavior.



This chemical equation balancer can help you to balance an unbalanced equation. This balancer can also help you check whether the equation is balanced or not

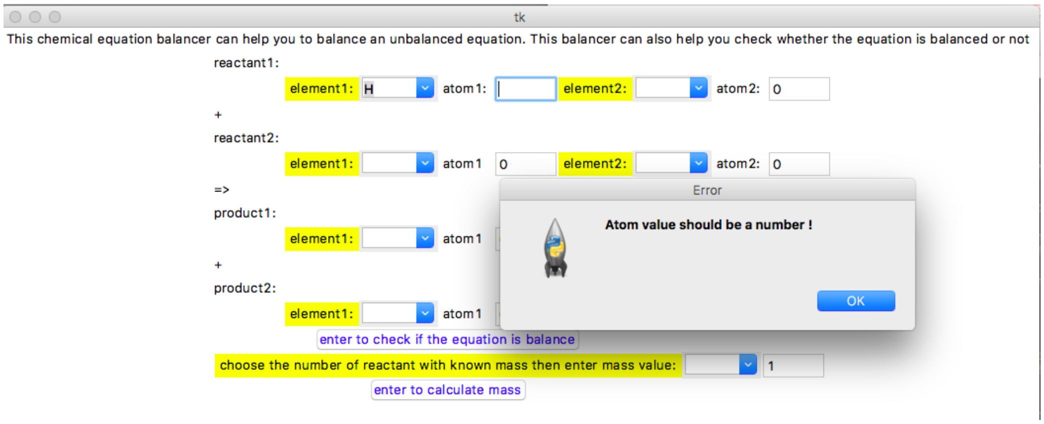
reactant1:
element1: H atom1: 0 element2: atom2: 0
+
reactant2:
element1: atom1 0 element2: atom2: 0
=>
product1:
element1: atom1
+
product2:
element1: atom1

Error

if you choose an element then atom value should be greater than 0

choose the number of reactant with known mass then enter mass value: 1

Figure 3: error massage if user enter atom less than 1 after selecting an element



This chemical equation balancer can help you to balance an unbalanced equation. This balancer can also help you check whether the equation is balanced or not

reactant1:
element1: H atom1: element2: atom2: 0
+
reactant2:
element1: atom1 0 element2: atom2: 0
=>
product1:
element1: atom1
+
product2:
element1: atom1

Error

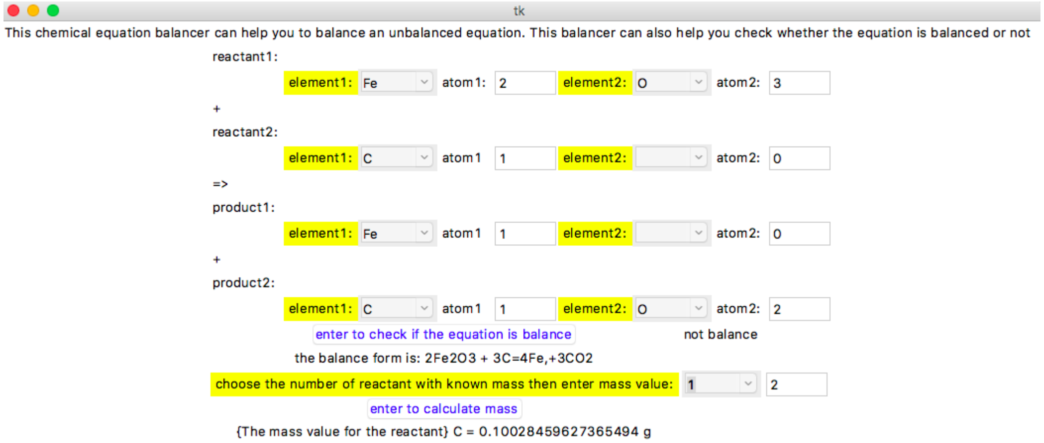
Atom value should be a number !

choose the number of reactant with known mass then enter mass value: 1

Figure 4: error massage if user enter non number atom value

6.3 Reaction Output Test-id3 , Mass Output Test-id4

The goal of this test is to output the final result including the mass and balance reaction to GUI. This is the main goal of the system. Picture below shows how system will print out the final result to end user. This test is for test cases id3, id4 in (1). Test passed correctly and system responded as intended.



This chemical equation balancer can help you to balance an unbalanced equation. This balancer can also help you check whether the equation is balanced or not

reactant1:

element1: Fe atom1: 2 element2: O atom2: 3

+

reactant2:

element1: C atom1 1 element2: atom2: 0

=>

product1:

element1: Fe atom1 1 element2: atom2: 0

+

product2:

element1: C atom1 1 element2: O atom2: 2

not balance

the balance form is: 2Fe2O3 + 3C=4Fe,+3CO2

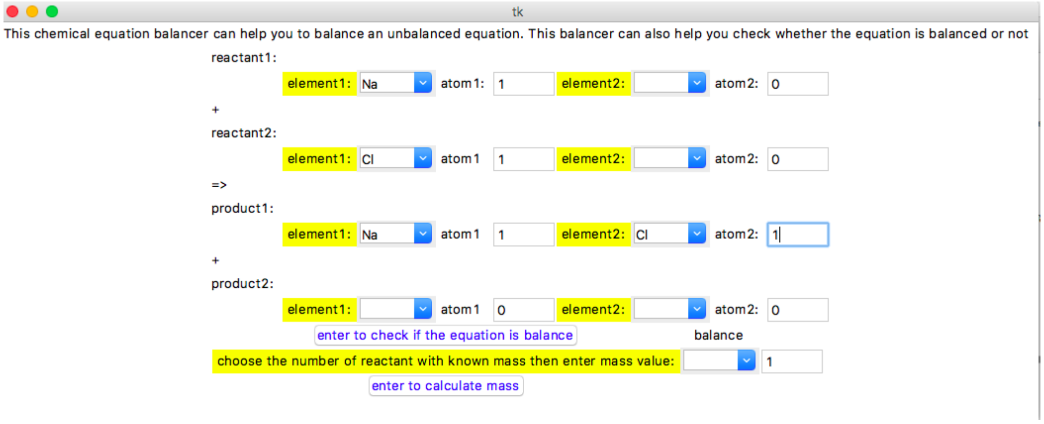
choose the number of reactant with known mass then enter mass value: 1 2

{The mass value for the reactant} C = 0.10028459627365494 g

Figure 5: final system output

6.4 Check Balance Test-id5

The goal of this test is to check if the entered reaction is balance or not. Display ”balance” if yes and ”not balance” with new balanced form if not. Pictures below show how system respond to balance and non balance reactions. The test passed correctly and system responded as intended.



This chemical equation balancer can help you to balance an unbalanced equation. This balancer can also help you check whether the equation is balanced or not

reactant1:

element1: Na atom1: 1 element2: atom2: 0

+

reactant2:

element1: Cl atom1 1 element2: atom2: 0

=>

product1:

element1: Na atom1 1 element2: Cl atom2: 1

+

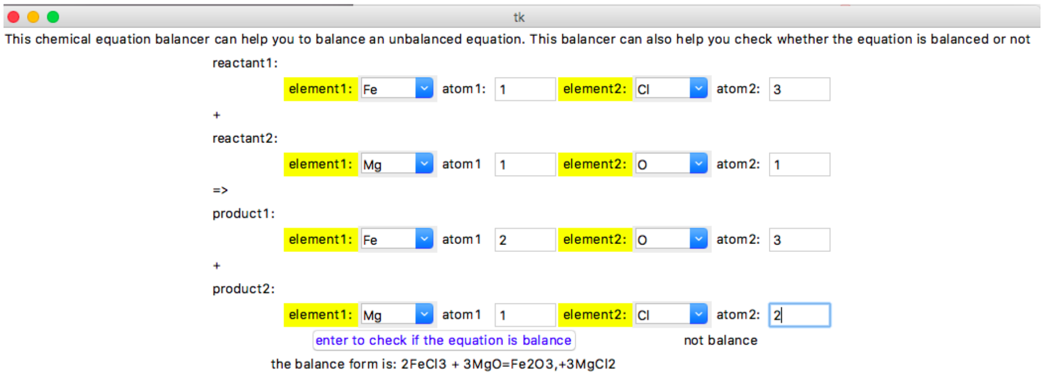
product2:

element1: atom1 0 element2: atom2: 0

balance

choose the number of reactant with known mass then enter mass value: 1

Figure 6: balance reaction



This chemical equation balancer can help you to balance an unbalanced equation. This balancer can also help you check whether the equation is balanced or not

reactant1:

element1: Fe atom1: 1 element2: Cl atom2: 3

+

reactant2:

element1: Mg atom1 1 element2: O atom2: 1

=>

product1:

element1: Fe atom1 2 element2: O atom2: 3

+

product2:

element1: Mg atom1 1 element2: Cl atom2: 2

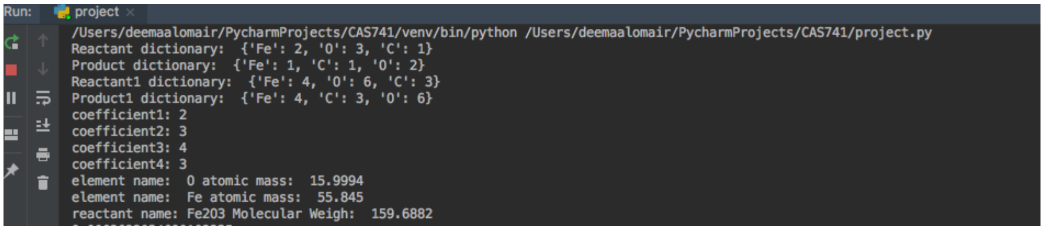
not balance

the balance form is: 2FeCl3 + 3MgO=Fe2O3,+3MgCl2

Figure 7: non balance reaction

6.5 Molecular Weight Calculation Test-id6

The goal of this test is to get molecular weight for a reactant. Picture below shows molecular weight is printed for reactant " Fe_2O_3" as the test case id6 in (1) and the result is identical. Test passed correctly and system responded as intended.

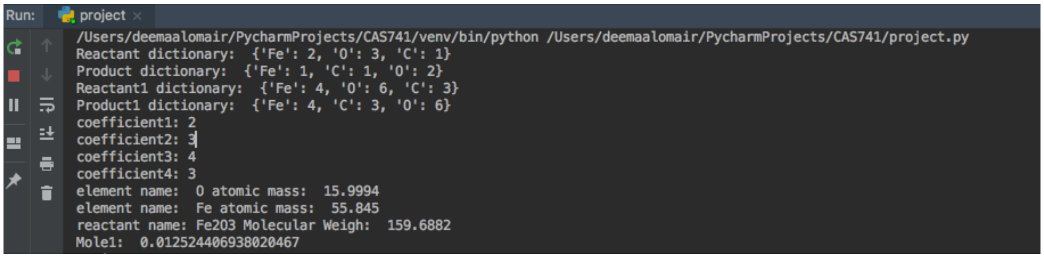


```
/Users/deemaalomair/PycharmProjects/CAS741/venv/bin/python /Users/deemaalomair/PycharmProjects/CAS741/project.py
Reactant dictionary:  {'Fe': 2, 'O': 3, 'C': 1}
Product dictionary:  {'Fe': 1, 'C': 1, 'O': 2}
Reactant1 dictionary:  {'Fe': 4, 'O': 6, 'C': 3}
Product1 dictionary:  {'Fe': 4, 'C': 3, 'O': 6}
coefficient1: 2
coefficient2: 3
coefficient3: 4
coefficient4: 3
element name:  O atomic mass:  15.9994
element name:  Fe atomic mass:  55.845
reactant name: Fe2O3 Molecular Weigh:  159.6882
```

Figure 8: molecular weight value

6.6 Mole1 Calculation Test-id7

The goal of this test is to get mole for reactant with known mass. we name this mole "Mole1". Picture below shows Mole1 value printed for reactant " Fe_2O_3" as the test case id7 in (1) and the result is identical. Test passed correctly and system responded as intended.

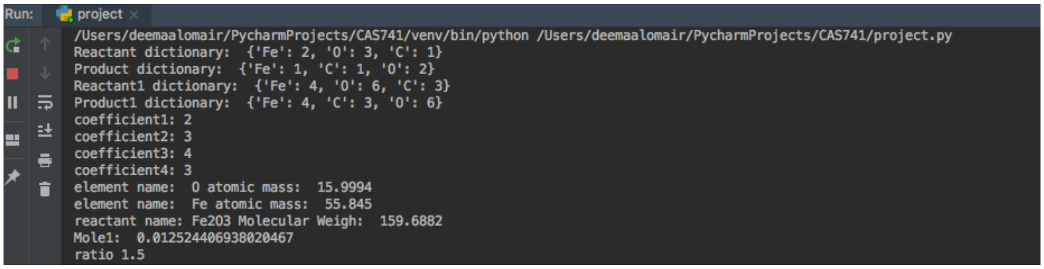


```
/Users/deemaalomair/PycharmProjects/CAS741/venv/bin/python /Users/deemaalomair/PycharmProjects/CAS741/project.py
Reactant dictionary:  {'Fe': 2, 'O': 3, 'C': 1}
Product dictionary:  {'Fe': 1, 'C': 1, 'O': 2}
Reactant1 dictionary:  {'Fe': 4, 'O': 6, 'C': 3}
Product1 dictionary:  {'Fe': 4, 'C': 3, 'O': 6}
coefficient1: 2
coefficient2: 3
coefficient3: 4
coefficient4: 3
element name:  O atomic mass:  15.9994
element name:  Fe atomic mass:  55.845
reactant name: Fe2O3 Molecular Weigh:  159.6882
Mole1:  0.012524406938020467
```

Figure 9: Mole1 value

6.7 Mole Ratio Calculation Test-id8

The goal of this test is to get mole ratio between given reactants. Picture below shows Mole Ratio value for coefficient of reactant2/coefficient of reactant1 printed as the test case id8 in (1) and the result is identical. Test passed correctly and system responded as intended.

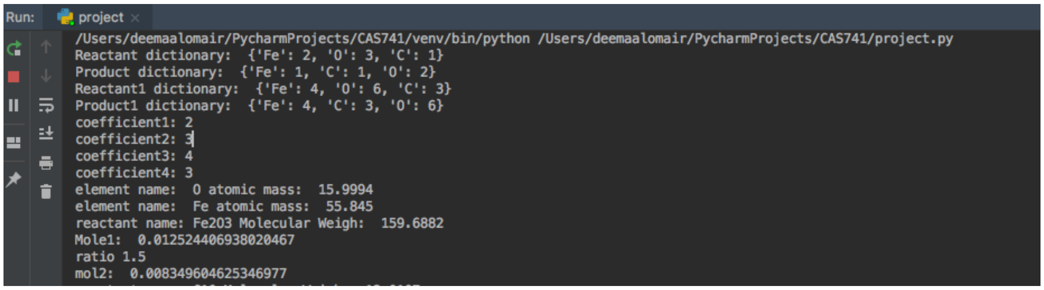


```
/Users/deemaalomair/PycharmProjects/CAS741/venv/bin/python /Users/deemaalomair/PycharmProjects/CAS741/project.py
Reactant dictionary:  {'Fe': 2, 'O': 3, 'C': 1}
Product dictionary:  {'Fe': 1, 'C': 1, 'O': 2}
Reactant1 dictionary:  {'Fe': 4, 'O': 6, 'C': 3}
Product1 dictionary:  {'Fe': 4, 'C': 3, 'O': 6}
coefficient1: 2
coefficient2: 3
coefficient3: 4
coefficient4: 3
element name:  O atomic mass:  15.9994
element name:  Fe atomic mass:  55.845
reactant name: Fe2O3 Molecular Weigh:  159.6882
Mole1:  0.012524406938020467
ratio 1.5
```

Figure 10: Mole Ratio value

6.8 Mole2 Calculation Test-id9

The goal of this test is to get mole for reactant with unknown mass. we name this mole ”Mole2”. Picture below shows Mole2 value printed as the test case id9 in (1) and the result is identical. Test passed correctly and system responded as intended.

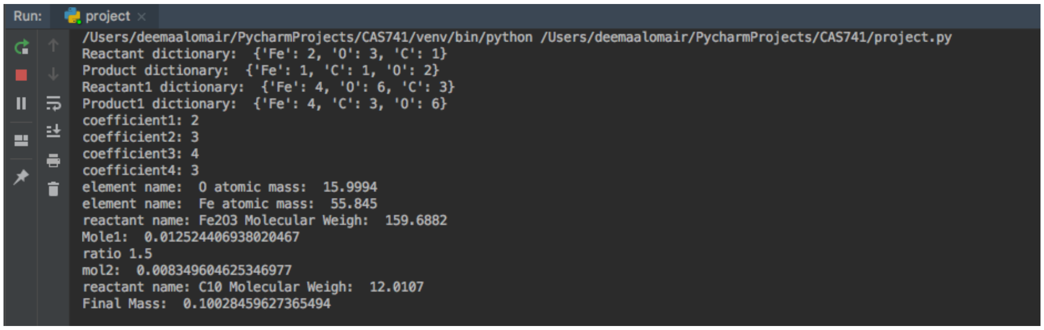


```
/Users/deemaalomair/PycharmProjects/CAS741/venv/bin/python /Users/deemaalomair/PycharmProjects/CAS741/project.py
Reactant dictionary:  {'Fe': 2, 'O': 3, 'C': 1}
Product dictionary:  {'Fe': 1, 'C': 1, 'O': 2}
Reactant1 dictionary:  {'Fe': 4, 'O': 6, 'C': 3}
Product1 dictionary:  {'Fe': 4, 'C': 3, 'O': 6}
coefficient1: 2
coefficient2: 3
coefficient3: 4
coefficient4: 3
element name:  O atomic mass:  15.9994
element name:  Fe atomic mass:  55.845
reactant name: Fe2O3 Molecular Weigh:  159.6882
Mole1:  0.012524406938020467
ratio 1.5
mol2:  0.008349604625346977
```

Figure 11: Mole2 value

6.9 Mass Calculation Test-id10

The goal of this test is to get the final mass result for reactant with unknown mass. Picture below shows mass value printed as the test case id10 in (1) and the result is identical. Test passed correctly and system responded as intended.



```
/Users/deemaalomair/PycharmProjects/CAS741/venv/bin/python /Users/deemaalomair/PycharmProjects/CAS741/project.py
Reactant dictionary:  {'Fe': 2, 'O': 3, 'C': 1}
Product dictionary:  {'Fe': 1, 'C': 1, 'O': 2}
Reactant1 dictionary:  {'Fe': 4, 'O': 6, 'C': 3}
Product1 dictionary:  {'Fe': 4, 'C': 3, 'O': 6}
coefficient1: 2
coefficient2: 3
coefficient3: 4
coefficient4: 3
element name:  O atomic mass:  15.9994
element name:  Fe atomic mass:  55.845
reactant name: Fe2O3 Molecular Weigh:  159.6882
Mole1:  0.012524406938020467
ratio 1.5
mol2:  0.008349604625346977
reactant name: C10 Molecular Weigh:  12.0107
Final Mass:  0.10028459627365494
```

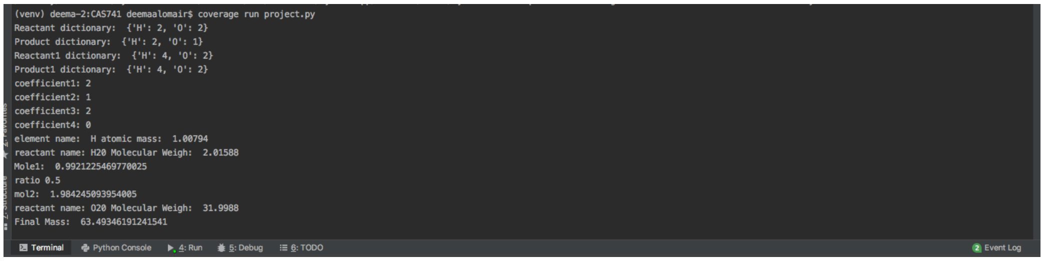
Figure 12: final mass value

7 Changes Due to Testing

No changes are necessary to the first stage of implementation due to these test results.

8 Automated Testing

- unit testing preformed for id1, id2 are all automated testing. they are done using pre-install ””unittest” package used by python. if assertion is not met then error massage will arise directly.
- coverage testing was preformed using coverage package without any error. blew picture shows coverage testing.



```
(venv) deema-2:CAS741 deemaalomair$ coverage run project.py
Reactant dictionary:  {'H': 2, 'O': 2}
Product dictionary:  {'H': 2, 'O': 1}
Reactant1 dictionary:  {'H': 4, 'O': 2}
Product1 dictionary:  {'H': 4, 'O': 2}
coefficient1: 2
coefficient2: 1
coefficient3: 2
coefficient4: 0
element name:  H atomic mass:  1.00794
reactant name: H20 Molecular Weigh:  2.01588
Mole1:  0.9921225469770025
ratio 0.5
mol2:  1.984245093954005
reactant name: O20 Molecular Weigh:  31.9988
Final Mass:  63.49346191241541
```

Figure 13: coverage testing

9 Trace to Requirements

	id1	id2	id3	id4	id5	id6	id7	id8	id9	id10
R1(input)	X	X								
R2(output)			X	X						
R3(calculation)						X	X	X	X	X
R4(VerifyInputOutput)	X	X	X	X						

Table 2: Traceability Matrix Showing the Connections Between unit test cases and functional requirements

	id11	id12
NF1		X
NF2	X	

Table 3: Traceability Matrix Showing the Connections Between unit test cases and Nonfunctional requirements

10 Trace to Modules

A complete description of modules is found in the MG (5).

	id1	id2	id3	id4	id5	id6	id7	id8	id9	id10
Input Module	X	X								
Atomic Mass Module						X				
Balancing Chemical Reaction Module			X		X					
Mass Calculation Module						X	X	X	X	X
GUI Module			X	X						

Table 4: Traceability Matrix Showing the Connections Between Modules and Test Cases

11 Code Coverage Metrics

coverage test was done for the whole system and covers all functional requirements. It covers test cases from id1 to id10.

References

[1] https://github.com/deemaalomair1/CAS741project/blob/master/docs/VnVPlan/SystVnVPlan/SystVnVPlan.pdf

[2] http://www.endmemo.com/chem/balancer.php

[3] https://github.com/deemaalomair1/CAS741project/blob/master/docs/VnVPlan/UnitVnVPlan/UnitVnVPlan.pdf

[4] https://github.com/deemaalomair1/CAS741project/tree/master/docs/VnVReport

[5] https://github.com/deemaalomair1/CAS741project/tree/master/docs/Design