**L&T Technology Services Limited**

20

**Learning Report**

**Applied SDLC – Project: Smart Tools**

**Pradeep V (PS: 99002647)**

Table of Contents

[1 Introduction 5](#_Toc53040273)

[1.1 Purpose 5](#_Toc53040274)

[1.2 Intended Audience and Reading Suggestions 5](#_Toc53040275)

[1.3 Product Scope 5](#_Toc53040276)

[1.4 References 5](#_Toc53040277)

[2 Overall Description 6](#_Toc53040278)

[2.1 Product Perspective 6](#_Toc53040279)

[2.2 Product Functions 6](#_Toc53040280)

[2.3 Operating Environment 6](#_Toc53040281)

[3 High Level System Requirements 7](#_Toc53040282)

[3.1 Menu Driven UI 7](#_Toc53040283)

[3.2 Main Modules 7](#_Toc53040284)

[4 Low Level System Requirement 8](#_Toc53040285)

[4.1 Swot Analysis 8](#_Toc53040286)

[4.2 Aging and costing 8](#_Toc53040287)

[4.3 System Feature 8](#_Toc53040288)

[4.3.1 Menu 8](#_Toc53040289)

[4.3.2 Calculator 8](#_Toc53040290)

[4.3.3 Unit Convertor 8](#_Toc53040291)

[4.4 Smart Tools: Component Diagram 9](#_Toc53040292)

[4.5 Smart Tools: Class Diagram 9](#_Toc53040293)

[4.6 Smart Tools: Use Case Diagram 10](#_Toc53040294)

[5 Test Cases 12](#_Toc53040295)

[6 Non-Functional Requirements 13](#_Toc53040296)

[6.1 Performance Requirements 13](#_Toc53040297)

[6.2 Safety requirements 13](#_Toc53040298)

[7 Build 13](#_Toc53040299)

[8 Introduction 14](#_Toc53040300)

[8.1 Purpose 14](#_Toc53040301)

[8.2 Intended Audience and Reading Suggestions 14](#_Toc53040302)

[8.3 Product Scope 14](#_Toc53040303)

[8.4 References 14](#_Toc53040304)

[9 Overall Description 15](#_Toc53040305)

[9.1 Product Perspective 15](#_Toc53040306)

[9.2 Product Functions 15](#_Toc53040307)

[9.3 Operating Environment 15](#_Toc53040308)

[10 High Level System Requirements 16](#_Toc53040309)

[10.1 Menu Driven UI 16](#_Toc53040310)

[10.2 Main Modules 16](#_Toc53040311)

[11 Low Level System Requirement 17](#_Toc53040312)

[11.1 System Feature 17](#_Toc53040313)

[11.1.1 Menu 17](#_Toc53040314)

[11.1.2 Calculator 17](#_Toc53040315)

[11.2 HiperCalculator: Component Diagram 18](#_Toc53040316)

[11.3 HiperCalculator: Class Diagram 19](#_Toc53040317)

[11.4 HiperCalculator: Use Case Diagram 20](#_Toc53040318)

[11.5 HiperCalculator: Sequence Diagram 21](#_Toc53040319)

[12 Test Cases 22](#_Toc53040320)

[13 Non-Functional Requirements 23](#_Toc53040321)

[13.1 Performance Requirements 23](#_Toc53040322)

[13.2 Safety requirements 23](#_Toc53040323)

[14 CPP Check 23](#_Toc53040324)

**Table of Figures**

[Figure 1: Component Diagram 7](#_Toc52966985)

[Figure 2: Class Diagram 8](#_Toc52966986)

[Figure 3: Use Case Diagram 9](#_Toc52966987)

[Figure 4: Sequence Diagram 10](#_Toc52966988)

[Figure 5: Building the project 12](#_Toc52966989)

Day1: Activity 1

# Introduction

## Purpose

The purpose of this project entitled “Smart Tools” is to design and develop a software system that consists of multiple day to day tools like Calculator, Unit Convertor and etc.

## Intended Audience and Reading Suggestions

The document is intended for peer reviewers and subject matter experts of L&T Technology Services Limited. This project is strictly for educational purpose and shall not be used for any commercial use. The author is not responsible for any damages happened by implementing this project.

## Product Scope

The purpose of this project is to learn different stages of SDLC, unit testing and CI/CD frame work.

## References

www.stackoverflow.com

www.draw.io

# Overall Description

## Product Perspective

Smart tool is a collection of day to day tools like calculator and unit convertor under a single package. The system is expected to run smoothly and efficiently across windows and Linux platforms.

## Product Functions

The major functions of this product are a Calculator and a Unit convertor that can convert at least temperature, distance and volume. The product should have menu driven UI.

## Operating Environment

The product only requires minimal hardware support so that it should run on any basic system that runs on Windows or Linux operating system.

# High Level System Requirements

## Menu Driven UI

The system should have menu driven UI. Each time when an user performing some functionality, after finishing the function it should return back to the menu until the user opted for exit

## Main Modules

The system should contain at least two tools Calculator and Unit Convertor.

# Low Level System Requirement

## Swot Analysis

|  |  |
| --- | --- |
| Strength  Fast and Reliable | Weakness  Can’t handle big numbers in Calculator |
| Opportunities  More functions in the future Currency convertor | **Threat**  Memory overflow |

## Aging and costing

|  |  |  |
| --- | --- | --- |
| Version | Aging | Costing |
| 2000 - 2005 | Basic 4 Operations | 250 |
| 2005 - 2010 | Basic 4 Operations + Unit Convertor | 300 |
| 2010 - present | Basic 4 Operation + Square + Modulus + Unit convertor | 500 |

## System Feature

### Menu

Menu should prompt available tools (Calculator and Unit Convertor). Priority: HIGH

REQ-01: Each time when user is completed the task menu should prompt again until the user press exit.

### Calculator

The Calculator module is to do the basic mathematical operations. Priority: HIGH

REQ-02: It should contain at least 4 functions (Add, Subtract, Multiply and Divide)

### Unit Convertor

Unit convertor converts different units such as temperature, distance and etc. Priority: HIGH

REQ-03: It should contain at least 3 options Temperature, Distance and Volume.

## Smart Tools: Component Diagram

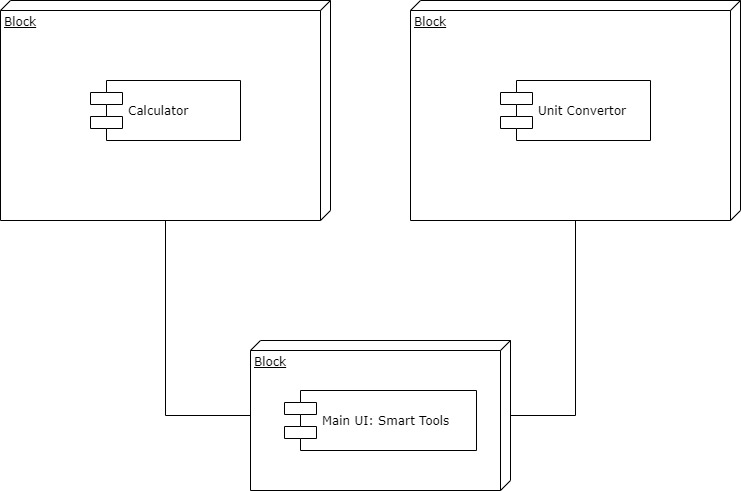


Figure 1: Component Diagram

## Smart Tools: Class Diagram

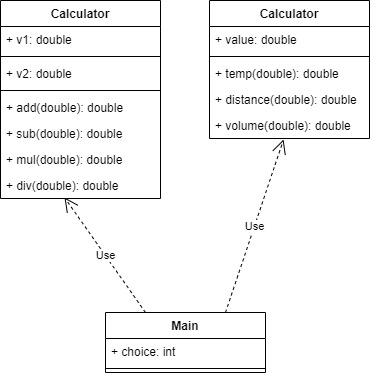


Figure 2: Class Diagram

## Smart Tools: Use Case Diagram

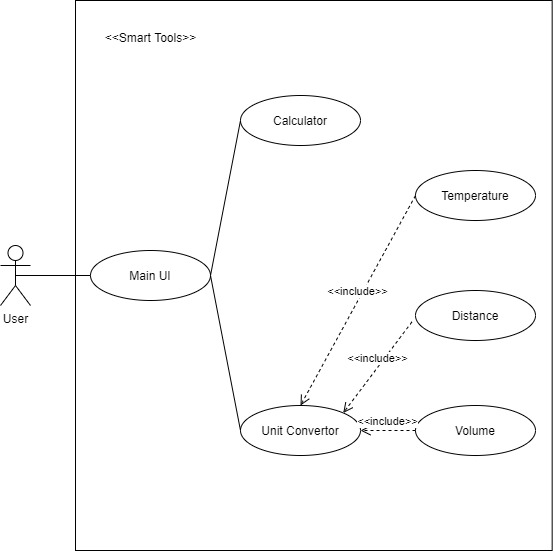


Figure 3: Use Case Diagram

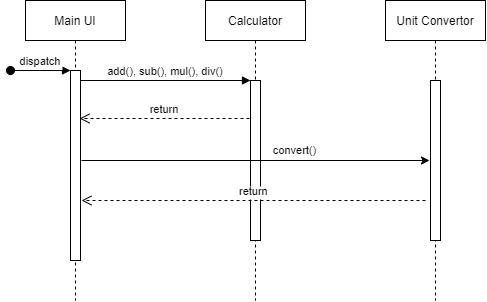


Figure 4: Sequence Diagram

# Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| TC01 | Test Add Functionality of the system | 25, 65 | 90 |  |  |
| TC02 | Test Add Functionality of the system | 455, 911 | 1366 |  |  |
| TC03 | Test Subtract Functionality of the system | 13589, 789 | 12800 |  |  |
| TC04 | Test Multiply Functionality of the system | 144, 23 | 3312 |  |  |
| TC05 | Test Divide Functionality of the system | 45, 5 | 9 |  |  |
| TC06 | Test Divide Functionality of the system | 66, 0 | Print “Can’t Divide by 0” |  |  |

# Non-Functional Requirements

## Performance Requirements

The system should work with minimal hardware requirement. I should work with any basic machine that runs on Windows or Linux operating system.

## Safety requirements

The system should safely run on any machine without producing any damages to it. There should be mechanism to ensure that it doesn’t overflow the memory or processing power while operating it.

# Build

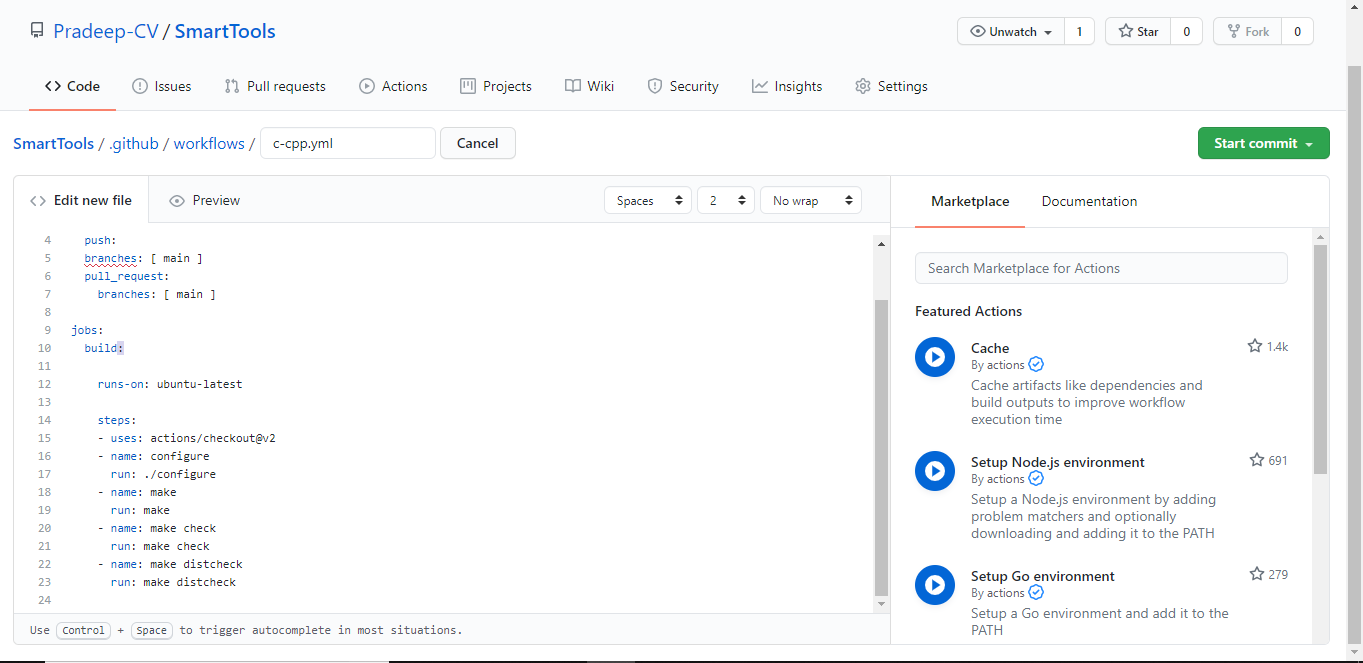


Figure 5: Building the project

Day 2: Activity 1: Calculator

# Introduction

## Purpose

The purpose of this project entitled “HiperCalculator” is to design and develop a software system that consists of basic mathematical operations in addition to two special functions.

## Intended Audience and Reading Suggestions

The document is intended for peer reviewers and subject matter experts of L&T Technology Services Limited. This project is strictly for educational purpose and shall not be used for any commercial use. The author is not responsible for any damages happened by implementing this project.

## Product Scope

The purpose of this project is to learn different stages of SDLC, unit testing and CI/CD frame work.

## References

www.stackoverflow.com

www.draw.io

# Overall Description

## Product Perspective

HiperCalculator is software that consists of 4 basic mathematical functions along with square and modulus operations under a single package. The system is expected to run smoothly and efficiently across windows and Linux platforms.

## Product Functions

The major functions are 4 basic operations Addition, Subtraction, Multiplication and division along with 2 special functions Square and Modulus. The product should have menu driven UI.

## Operating Environment

The product only requires minimal hardware support so that it should run on any basic system that runs on Windows or Linux operating system.

# High Level System Requirements

## Menu Driven UI

The system should have menu driven UI. Each time when an user performing some functionality, after finishing the function it should return back to the menu until the user opted for exit

## Main Modules

The system should contain at least 6 functions, 4 basic operations along with 2 more operations Square and Modulus.

# Low Level System Requirement

## System Feature

### Menu

Menu should prompt available operations (Add, Subtract, Multiply Divide, Square and Modulus). Priority: HIGH

REQ-01: Each time when user is completed the task menu should prompt again until the user press exit.

### Calculator

The Calculator module is to do the basic mathematical operations. Priority: HIGH

REQ-02: It should contain at least 4 functions (Add, Subtract, Multiply and Divide), in addition to that it should contain Square and Modulus.

## HiperCalculator: Component Diagram

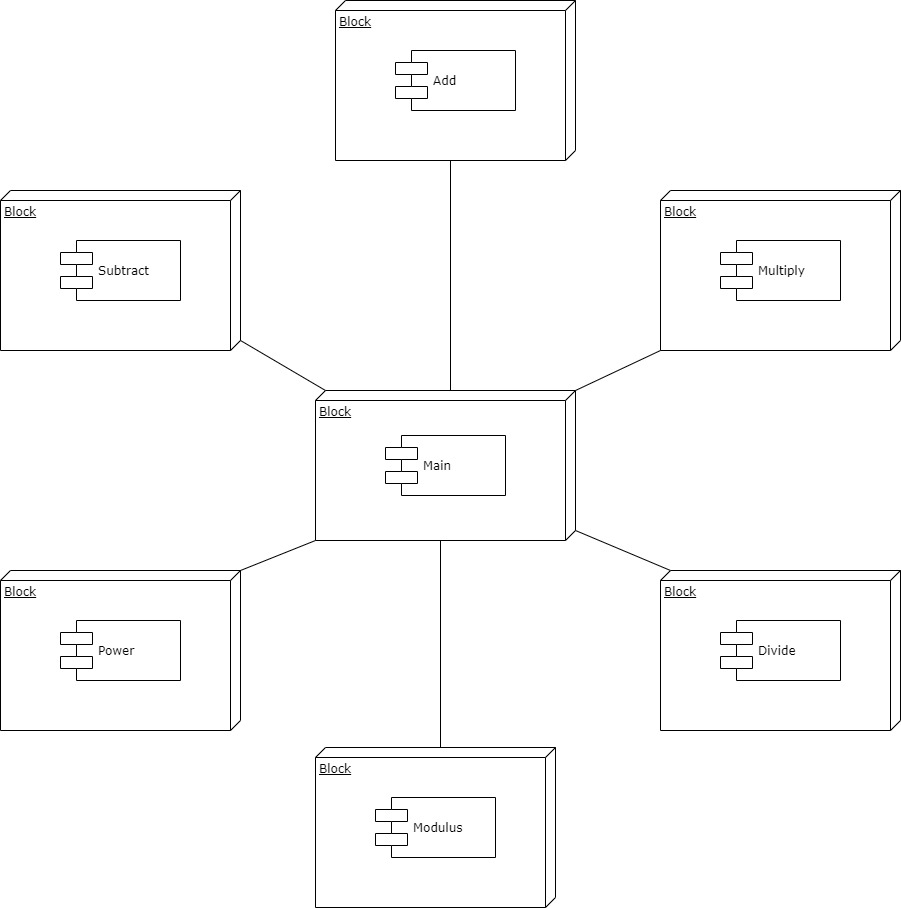


Figure 6: Component Diagram

## HiperCalculator: Class Diagram

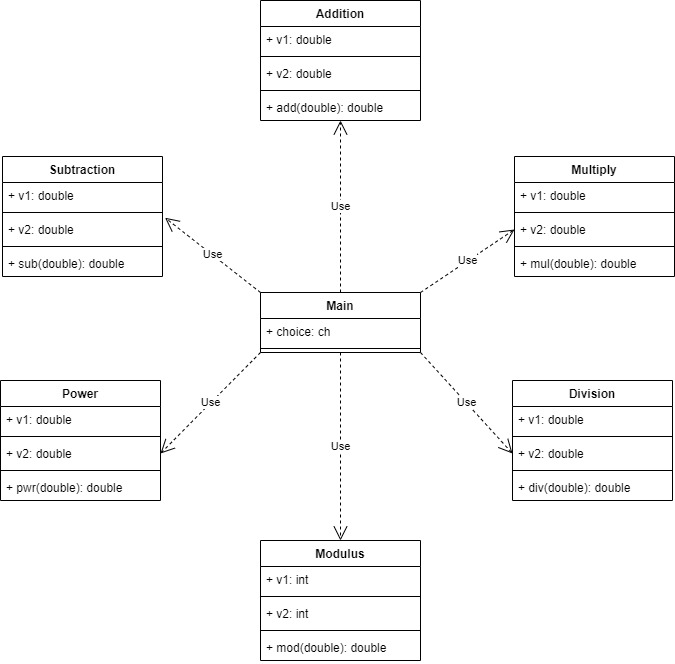


Figure 7: Class Diagram

## HiperCalculator: Use Case Diagram

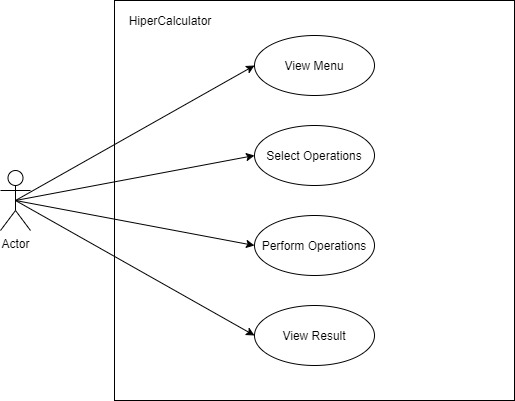


Figure 8: Use Case Diagram

## HiperCalculator: Sequence Diagram

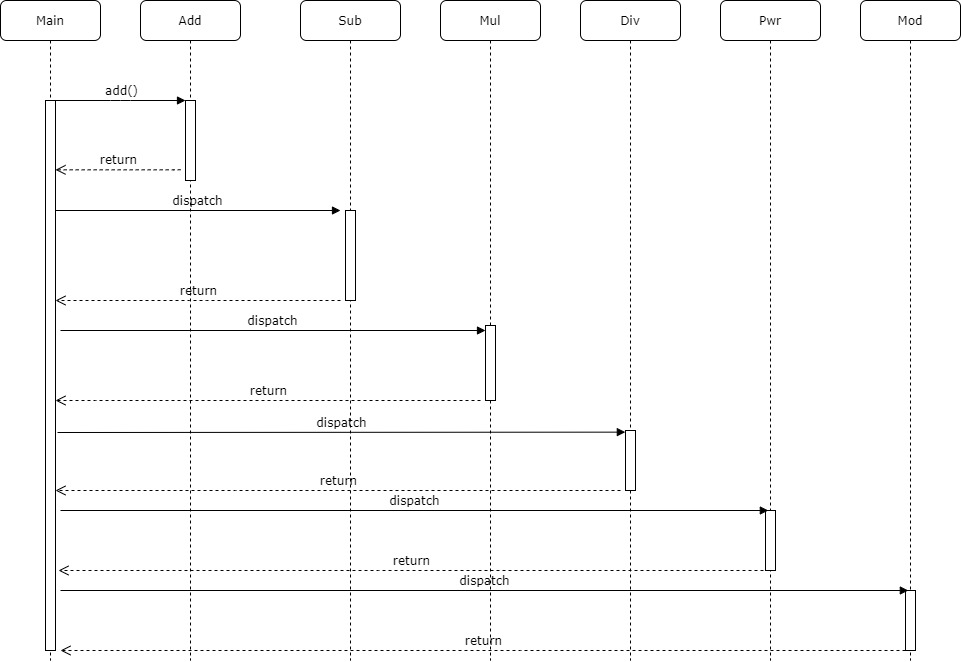


Figure 9: Sequence Diagram

# Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Data | Expected Result | Actual Result | Pass/Fail |
| TC01 | Test Add Functionality of the system | 25, 65 | 90 |  |  |
| TC02 | Test Add Functionality of the system | 455, 911 | 1366 |  |  |
| TC03 | Test Subtract Functionality of the system | 13589, 789 | 12800 |  |  |
| TC04 | Test Multiply Functionality of the system | 144, 23 | 3312 |  |  |
| TC05 | Test Divide Functionality of the system | 45, 5 | 9 |  |  |
| TC06 | Test Divide Functionality of the system | 66, 0 | Print “Can’t Divide by 0” |  |  |
| Boundary Test | | | | | |
| TC07 | Test Addition of large number | 65535, 1 | Print “Overflow” |  |  |
| TC08 | Test divide by 0 | 455, 0 | Should handle the exception |  |  |

# Non-Functional Requirements

## Performance Requirements

The system should work with minimal hardware requirement. I should work with any basic machine that runs on Windows or Linux operating system.

## Safety requirements

The system should safely run on any machine without producing any damages to it. There should be mechanism to ensure that it doesn’t overflow the memory or processing power while operating it.

# CPP Check

## Link to GitHub: [GitHub](https://github.com/99002647/HiperCalculator)

