Group 12

Calculator
Software Requirements Specifications

Version 1.7

Revision History

Date	Version	Description	Author
30/Sept/24	1.0	Transferred Basic Information	Maxfield Freeman
7/Oct/24	1.1	Assigned Roles to members	Maxfield Freeman
16/Oct/24	1.2	Transferred information from previous versions	Caleb Neil
16/Oct/24	1.3	Section 2 Completed	Noah Haskins
16/Oct/24	1.4	Section 3.1, 3.2 Completed	Carson Schraad
18/Oct/24	1.5	Updating information, completed functional requirements	Maxfield Freeman
19/Oct/24	1.6	Wrote Introduction & revised other sections. Added to specific requirements.	Maren Proplesch
20/Oct/24	1.7	Added UML Use Case Diagram	Carson Schraad

Table of Contents

1. Introduction 4
   - 1.1 Purpose 4
   - 1.2 Scope 4
   - 1.3 Definitions, Acronyms, and Abbreviations 4
   - 1.4 References 4
   - 1.5 Overview 5

2. Overall Description 5
   - 2.1 Product perspective 5
     - 2.1.1 User Interfaces 5
     - 2.1.2 Software Interfaces 5
     - 2.1.3 Memory Constraints 5
   - 2.2 Product functions 5
   - 2.3 User characteristics 5
   - 2.4 Constraints 5
   - 2.5 Assumptions and dependencies 5

3. Specific Requirements 5
   - 3.1 Functionality 5
     - 3.1.1 Expression Parsing 6
     - 3.1.2 Operator Support 6
     - 3.1.3 Parenthesis Handling 6
     - 3.1.4 Numeric Constant Handling 6
     - 3.1.5 Error Handling 6
     - 3.1.6 User-Friendly User Interface 6
   - 3.2 Use-Case Specifications 6
   - 3.3 Supplementary Requirements 6
   - 3.1.6 User-Friendly User Interface 6

4. Classification of Functional Requirements 6

5. Appendices 6

Software Requirements Specifications

1. Introduction

This document describes what the Calculator App will do and what's needed to build it. The goal is to create a simple, yet powerful tool that handles basic math like addition and subtraction, as well as more advanced stuff like parenthesis and order of operations. It's meant to guide everyone involved in the project, from developers to testers, to ensure everything works smoothly.

1.1 Purpose

The purpose of the *Software Requirement Specifications* is to fully describe the external behavior of the Calculator application. It describes factors such as nonfunctional requirements and design constraints that are necessary to provide an accurate and well rounded description of the requirements for the software.

1.2 Scope

This document applies to a calculator software application that takes in input expression, and outputs the answer. The scope will be limited to a CLI based application.

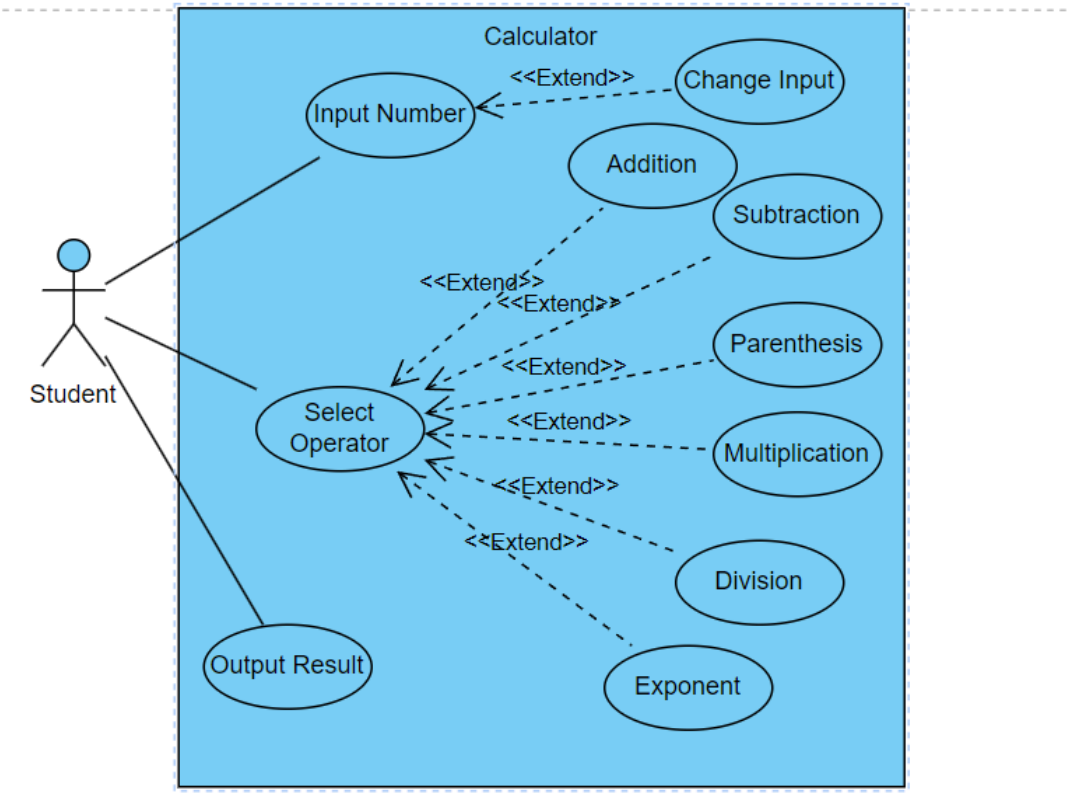
1.3 Definitions, Acronyms, and Abbreviations

See the Project Glossary

1.4 References

*For the **Software Requirements Specification**, the list of referenced artifacts includes:*

- *Glossary Available at Link*
- *UML Use-Case Diagram:*



1.5 Overview

This document describes the requirements for creating our software product.

It contains:

Overall Description	—	A description of the product that we will create including the product perspective, functions, characteristics, constraints, and dependencies.
Specification Requirements	—	This section contains all the requirements in extreme detail.
Functional Requirements	—	A list of all functional requirements for the application.
Appendices	—	The UML Use-Case Diagram.

2. Overall Description

2.1 Product perspective

2.1.1 User Interfaces

The user will input a mathematical expression and the program will output the solution to said equation.

2.1.2 Software Interfaces

String variable holding equation followed by functions splitting and solving the equations to produce an output variable.

2.1.3 Memory Constraints

Simple error checking will be required in order to prevent large overflow errors when doing more advanced arithmetic.

2.2 Product functions

Calculate basic mathematical functions.

2.3 User characteristics

Math Students are the most common users.

2.4 Constraints

Anything that would constrain the performance of the calculator, overflow error, and using less than 50% of available RAM. Ideal usage is under 100 megabytes.

2.5 Assumptions and dependencies

Dependences: math.h

3. Specific Requirements

The application should receive input from the user via CLI. The application should be able to handle multiple requests as error handling to prevent any crashes. The program should be capable of evaluating mathematical expressions of any length such as 1 + 2 * 3 / 4. It should also support parenthesis, as well as unary operations such as trigonometric operations. The output should be a decimal. It should obey the rules of parenthesis and order of operations. A syntax error should be displayed if the expression is an invalid expression, as opposed to exiting.

3.1 Functionality

The functionality of our program is designated by its functional requirements, which are classified later in this document. The functionality of our program is based on 2 major things, understanding what the user is asking the program to do and outputting the correct mathematical solutions. If a user inputs five plus three then our program will understand that as a mathematical equation and output eight.

3.1.1 Operator Support

Our program has support for the following operators: '+', '-', *' ', '/', '%', '**', '(', ')'*

3.1.2 Parenthesis Handling

Our program has support for parenthesis so expressions like (5+3)/2 will be evaluated correctly.

3.1.3 Numeric Constant Handling

Our program has support for some numeric constants which have a fixed value

3.2 Use-Case Specifications

Use cases include: all arithmetic functions such as addition, subtraction, multiplication, division and exponentiation. Another use-case could be clearing the input that the user has submitted.

3.3 Supplementary Requirements

There are 3 main non-functional requirements this program has; expression parsing, error handling and a user-friendly interface. Our program will parse the user's input in a readable way for the program to understand it as a mathematical equation. This program will also identify any inputs that are not mathematical equations and will handle those errors. The program will also have an interface that has a command line prompt with which users can enter their equations.

4. Classification of Functional Requirements

Functionality	Type
3.1.1 Operator Support	Essential
3.1.2 Parenthesis Handling	Essential
3.1.3 Numeric Constant Handling	Essential

5. Appendices