Arithmetic Expression Evaluator

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 10/02/2023 | <1.0> | Update Introduction | Elizabeth Channel |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 4

2. Overall Description 5

2.1 Product perspective 5

2.1.1 System Interfaces 5

2.1.2 User Interfaces 5

2.1.3 Hardware Interfaces 5

2.1.4 Software Interfaces 5

2.1.5 Communication Interfaces 5

2.1.6 Memory Constraints 5

2.1.7 Operations 5

2.2 Product functions 5

2.3 User characteristics 5

2.4 Constraints 5

2.5 Assumptions and dependencies 5

2.6 Requirements subsets 5

3. Specific Requirements 5

3.1 Functionality 5

3.1.1 <Binary Operators> 5

3.1.2 <Unary Operators> 6

3.1.3 <Parentheses> 6

3.1.4 <Operator Precedence> 6

3.1.5 <Errors> 6

3.1.6 <User Input> 6

3.1.7 <System Output> 6

3.1.8 <User Interface> 6

3.2 Use-Case Specifications 7

3.3 Supplementary Requirements 7

4. Classification of Functional Requirements 7

5. Appendices 7

# Introduction

The following document will capture the complete software requirements for the system. It will be referred to throughout the *Arithmetic Expression Evaluator* project.

## Purpose

The purpose of the Software Requirements Specifications (SRS) document is to detail the external needs of the application. It describes all nonfunctional requirements along with interfaces, constraints, and functionality. It will be referred to throughout the project to ensure organization.

## Scope

This SRS applies to the *Arithmetic Expression Evaluator* application. It is associated with the Use-Case model for the entire system and its functions. The organization of software requirements and functionality revolve around this document.

## Definitions, Acronyms, and Abbreviations

See *Glossary* in Appendices

See *Project Glossary* in Annexes of *Software Development Plan*

## References

* Software Development Plan – 9/22/2023, CAVJAC Group

## Overview

The rest document will capture the complete software requirements for the system. It contains many aspects of the project such as interfaces and functionality. It is organized by specification, starting general is an overview of the project and ending with specific requirements.

# Overall Description

[This section of the **SRS** describes the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3, and makes them easier to understand. Include such items as:

## Product perspective

### System Interfaces

### User Interfaces

### Hardware Interfaces

### Software Interfaces

### Communication Interfaces

### Memory Constraints

### Operations

## Product functions

## User characteristics

## Constraints

## Assumptions and dependencies

## Requirements subsets

# Specific Requirements

[This section of the **SRS** contains all software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. When using use-case modeling, these requirements are captured in the Use Cases and the applicable supplementary specifications. If use-case modeling is not used, the outline for supplementary specifications may be inserted directly into this section, as shown below.]

## Functionality

[This section describes the functional requirements of the system for those requirements that are expressed in the natural language style. For many applications, this may constitute the bulk of the **SRS** package and thought should be given to the organization of this section. This section is typically organized by feature, but alternative organization methods may also be appropriate; for example, organization by user or organization by subsystem. Functional requirements may include feature sets, capabilities, and security.

Where application development tools, such as requirements tools, modeling tools, and the like, are employed to capture the functionality, this section of the document would refer to the availability of that data, indicating the location and name of the tool used to capture the data.]

### Binary Operators

Binary operators are operators that require two operands. The operator is preceded by its first operand and proceeded by its second. Their operands shall be either constant numbers or expressions which produce numerical values when evaluated. If either of their operands lacks a numerical value, or the operator has fewer than two operands, an error exists, as specified in section 3.1.5. All binary operators shall evaluate to a numerical value.

#### Addition

The calculator shall be able to perform addition with two operands. Addition shall be represented with the “+” symbol. It shall conform to the requirements for binary operators as defined in section 3.1.1. Its value shall equal the sum of its operands.

#### Subtraction

The calculator shall be able to perform subtraction with two operands. Subtraction shall be represented with the “-” symbol. It shall conform to the requirements for binary operators as defined in section 3.1.1. Its value shall equal the difference between its first operand and its second.

#### Multiplication

The calculator shall be able to perform multiplication with two operands. Multiplication shall be represented with the “\*” symbol. It shall conform to the requirements for binary operators as defined in section 3.1.1. Its value shall equal the product of its operands.

#### Division

The calculator shall be able to perform integer division with two operands. Division shall be represented with the “/” symbol. It shall conform to the requirements for binary operators as defined in section 3.1.1. Its value shall equal the quotient of its first and second operands. Its value shall be an integer, truncating any fractional values.

#### Modulo

The calculator shall be able to perform modulo with two operands. Modulo shall be represented with the “%” symbol. It shall conform to the requirements for binary operators as defined in section 3.1.1. Its value shall equal the value of its first operand mod its second, or the remainder of its first operand divided by its second.

#### Exponentiation

The calculator shall be able to perform exponentiation with two operands. Exponentiation shall be represented with the “\*\*” symbol. It shall conform to the requirements for binary operators as defined in section 3.1.1. Its value shall equal its first operand to the power of its second operand, or its first operand times itself a number of times equal to the second operand. The second operand may not be negative; if it is, an error exists, as specified in section 3.1.5.

### Unary Operators

Unary operators are operators that require one operand. The operator precedes its operand. Its operand shall be either a constant number or an expression which produces a numerical value when evaluated. If its operand lacks a numerical value, or the operator has no operands, an error exists, as specified in section 3.1.5.

#### Negation

#### Absolute Value

### Parentheses

### Operator Precedence

In equations with more than one operator, operations shall be performed with the standard operator precedence, as follows.

Level 5: Parentheses

Level 4: Exponentiation

Level 3: Multiplication, division, and modulo

Level 2: Negation and absolute value

Level 1: Addition and Subtraction

Operations shall be executed from highest precedence level to lowest.

### Errors

#### Error type 1

### User Input

### System Output

The system shall have two categories of outputs: the numerical value of the user-provided equation, or an error message. These will be communicated to the user as described in section 3.1.8.

The system shall be able to output up to twelve characters of numerical output. The numerical output shall be an integer in base ten. The system shall output only characters necessary to communicate the numerical value; there shall be no preceding zeroes. Numerical output shall consist only of the numbers 0-9 and ‘-’.

The system shall output error messages in the following format: “Error: [message]”, where the message shall be a message relevant to the error, as defined in section 3.1.5. The total output shall be no longer than 32 characters.

### User Interface

The user shall be able to interact with the calculator through a textual interface. The user shall be able to type an equation consisting of the symbols specified in section 3.1.6. The user may input symbols outside this set of symbols; however, this will result in an error, as described in section 3.1.5.

The user shall be able to evaluate their equation by pressing the Enter key. At this point, the calculator shall display to the user the numerical result of their equation, or the relevant error if an error has occurred, as specified in section 3.1.7.

The user shall be able to edit their equation after evaluation. This shall not change the displayed output until the equation is evaluated again.

The user shall be able to exit the calculator by pressing the window closing button standard to the current operating system.

## Use-Case Specifications

[In use-case modeling, the use cases often define the majority of the functional requirements of the system, along with some non-functional requirements.]

## Supplementary Requirements

[Supplementary Specifications capture other requirements, e.g., non-functional requirements and development constraints, that are not included in the use cases and non-functional requirements.]

# Classification of Functional Requirements

[List, usually in a table, all functional requirements and order them by Type (Essential, Desirable, and Optional) or by order of appearance in the document.]

|  |  |
| --- | --- |
| **Functionality** | **Type** |
| ... |  |
| ... |  |

# Appendices

[When appendices are included, the **SRS** should explicitly state whether or not the appendices are to be considered part of the requirements]