

# **SRS DOCUMENT FOR “SCIENTIFIC CALCULATOR”**

PROJECT NAME:- Console Based Scientific Calculator
Date: 19/12/2023
Version: 1.1.1

# INDEX

## **1. Introduction**

1.1 Product Scope

1.2 Product Value

1.3 Intended Audience

1.4 Intended Use

1.5 General Description

## **2. Project Strategy**

2.1 Define Project Strategy

2.2 Milestones

## **3. System Architecture**

3.1- Hardware and Software Requirement

3.2- Environment and Interface Requirements

## **4. Flowchart of Application**

## **5. Functional Requirements**

5.1 Mathematical Operations

- 5.1.1 Basic Arithmetic Operations
- 5.1.2 Trigonometric Functions
- 5.1.3 Logarithmic and Exponential Functions
- 5.1.4 Factorial and Power Functions
- 5.1.5 Square Root Function
- 5.1.6 Cube Root Function
- 5.1.7 Factorial Function
- 5.1.8 Modulus Function
- 5.1.9 Absolute Function
- 5.1.10 Ceiling Function
- 5.1.11 Floor Function
- 5.1.12 Degrees-Minutes-Seconds (DMS) Function
- 5.1.13 Exponential Function
- 5.2 Memory Functions
  - 5.2.1 Memory Storage and Recall
- 5.3 Error Handling
  - 5.3.1 Input Validation
  - 5.3.2 Precision Handling
  - 5.3.3 Output
- 5.4 User Interface
  - 5.4.1 Console-Based Interface
- 5.5 Mode Selection

#### 5.5.1 Degrees and Radians Mode

### 5.6 Key Mapping

### 5.7 Non-functional Requirements

#### 5.7.1 Usability

#### 5.7.2 Performance

#### 5.7.3 Compatibility

#### 5.7.4 Reliability

#### 5.7.5 Scalability

#### 5.7.6 Maintainability

## 6. Key Mapping

- A detailed list of key functions and their corresponding keyboard shortcuts.

## 7. Non-functional Requirements

### 7.1 Usability

### 7.2 Performance

### 7.3 Compatibility

### 7.4 Reliability

### 7.5 Scalability

### 7.6 Maintainability

# **1. Introduction-**

## **1.1 Product Scope-**

The Scientific Calculator is a piece of software that provides refined mathematical functions for applications in science and engineering. This application aims to offer a complete range of mathematical operations, functions, and capabilities beyond what is typically found in calculators. The project's scope includes developing a powerful and intuitive software tool that simplifies complex calculations and data analysis.

## **1.2 Product Value-**

The goal of the Scientific Calculator is to provide users in the scientific, engineering, professional, and academic sectors with important advantages. The software helps clients become more productive, accurate, and convenient when performing a variety of mathematical and analytical tasks by providing a wide range of mathematical functions, accurate calculations, and an intuitive user interface.

## **1.3 Intended Audience-**

The Scientific Calculator's main target audience consists of scientists, engineers, educators, students, and professionals working in fields where complex mathematical calculations are essential. The program serves users looking for a dependable, feature-rich tool to effectively solve challenging mathematical problems.

## **1.4 Intended Use-**

The Scientific Calculator is designed to give users a flexible and easily navigable platform for carrying out complex mathematical operations. The program can be used for advanced mathematical operations like trigonometry, statistical analysis, and algebraic computations. Personal computers, laptops, and other devices that are compatible with it are meant to be used with the software.

## **1.5 General Description-**

Developed to be a stand-alone software program, the Scientific Calculator will have a Console Based interface that is simple to use. Numerous mathematical operations, including but not limited to arithmetic operations, scientific notation, logarithmic functions, statistical computations, and trigonometric functions, will be supported by the application. It will be built with precision, responsiveness, and accuracy in mind, providing exact results for a range of mathematical situations.

## **2. Project Strategy-**

### **2.1. Define Project Strategy-**

Implement a scientific calculator with a modular structure that focuses on individual mathematical operations. Conduct rigorous testing of each module and fix issues early in the development cycle to ensure accuracy and reliability.

### **2.2. Milestones-**

The project strategy for the C# console-based environment uses an agile and iterative development approach to enable continuous improvement based on user feedback. Development involves building and improving individual math skills, such as basic math, simple math including degree, radian and degree methods, and advanced math including square roots and variables. Other features include improvements to the user interface to improve clarity and interaction, including the introduction of interactive functions such as F-E (factorial) and E-F(exponential).

Rigorous testing, error handling, and integration validation are critical to ensure accuracy and reliability. The development process includes user acceptance testing, final revisions, and proof of concept for the final version of the proven calculator. Post-release support is provided for bug fixes, user queries, and future improvements based on user feedback. This strategy ensures a systematic and flexible approach that focuses on technical aspects and user experience throughout the project life cycle.

### 3. System Architecture-

The application will have a modular design with separate components for input parsing, calculation processing, output formatting, and error handling. This modularity will facilitate future enhancements and maintenance.

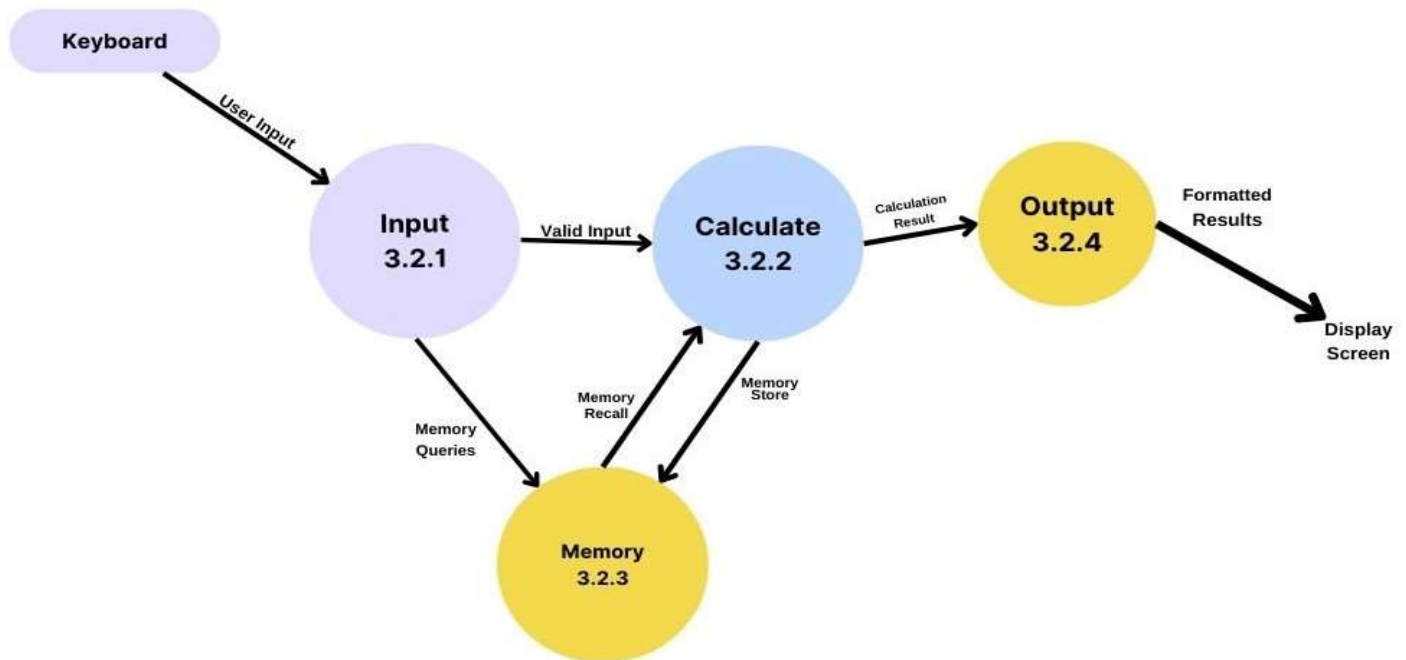
#### 3.1- Hardware and Software Requirement-

Hardware and Software Requirement	
Programming Language	C#
Integrations	Console Application
OS	Windows/Linux /macOS
Frameworks	.NET

#### 3.2- Environment and Interface Requirements-

- The program shall be written in standard C#. The program shall use only standard C# library functions. The program shall be usable on any system which supports the compiler, and shall not require any particular hardware or software.
- External Data Bases and File Interfaces- There are no existing external data bases or les that will be needed by this program.
- Human Interfaces-The program shall operate as much as possible in the same way as a regular handheld calculator, such that anyone familiar with the operation of such a device should have little trouble using the program. The user can then type a sequence of numbers and operators, similar to a regular calculator. The program will display the calculated answer to the entered problem on the line following the input line.

## 4. Flowchart of Application



## 5. Functional Requirements: -

### 5.1. Mathematical Operations:

#### 5.1.1 Basic Arithmetic Operations:

Requirement: To perform Addition, Subtraction, Multiplication & Division.

Acceptance Criteria: -

- There must be at least 2 numbers to perform this operation.
- Addition: The calculator should accurately add two numbers.
- Subtraction: It should accurately subtract one number from another.
- Multiplication: Accurate multiplication of two numbers.
- Division: Perform correct division operations.



### 5.1.2 Trigonometric Functions:

Requirement: To implement:

- a.) Sine ( $\sin$ ), Cosine ( $\cos$ ), Tangent ( $\tan$ ), Cosecant ( $\operatorname{cosec}$ ), Secant ( $\sec$ ), Cotangent ( $\cot$ )
- b.) Arcsine ( $\sin^{-1}$ ), Arccosine ( $\cos^{-1}$ ), Arctangent ( $\tan^{-1}$ ), Arccosecant ( $\operatorname{cosec}^{-1}$ ), Arcsecant ( $\sec^{-1}$ ), Arccotangent ( $\cot^{-1}$ )
- c.) Hyperbolic sine, Hyperbolic cosine & Hyperbolic tangent

Acceptance Criteria:

- There must be the functions should correctly calculate trigonometric ratios for given angles (in degree, radian or gradian).

### 5.1.3 Logarithmic and Exponential Functions:

Requirement: Support logarithm, exponentiation, and square root.

Acceptance Criteria:

- Logarithm: Accurately compute logarithm to a base.
- Exponentiation: Compute exponential values.
- Square Root: Accurately calculate the square root of a number.

### 5.1.4 Factorial and Power Functions:

- Requirement: Compute factorial and power of a number.
- Factorial: Calculate the factorial of an integer.
- Power: Compute the power of a number.

### 5.1.5 Square Root Function

- Requirement: Calculate the square root of a number.
- Acceptance Criteria: Provide accurate square root values for non-negative numbers.

### 5.1.6 Cube Root Function

- Requirement: Compute the cube root of a number.
- Acceptance Criteria: Provide accurate cube root values for both positive and negative numbers.

### 5.1.7 Factorial Function

- Requirement: Compute the factorial of a non-negative integer.
- Acceptance Criteria: Calculate  $n!$  accurately for input values of  $n$  (0 or greater).

### 5.1.8 Modulus Function

- Requirement: Calculate the modulus of two integers.
- Acceptance Criteria: Provide the absolute value of the remainder when dividing one integer by another.

### 5.1.9 Absolute Function

- Requirement: Compute the absolute value of a number.
- Acceptance Criteria: Return the positive magnitude of the provided number, disregarding its sign.

### 5.1.10 Ceiling Function

- Requirement: Determine the smallest integer greater than or equal to a given number.
- Acceptance Criteria: Round up the input value to the nearest integer, if necessary.

### 5.1.11 Floor Function

- Requirement: Determine the largest integer less than or equal to a given number.

- Acceptance Criteria: Round down the input value to the nearest integer, if necessary.

#### 5.1.12 Degrees-Minutes-Seconds (DMS) Function

- Requirement: Convert decimal degrees to degrees, minutes, and seconds.
- Acceptance Criteria: Accurately convert decimal degrees to its equivalent in degrees, minutes, and seconds format.

#### 5.1.13 Exponential Function

Requirement: Compute the exponential value of a number.

Acceptance Criteria: Accurately calculate  $e^x$  for a given input  $x$ .

### 5.2. Memory Functions:

#### 5.2.1 Memory Storage and Recall:

Requirement: Allow users to store, delete and recall values.

Acceptance Criteria:

Add(M+) : Add the result of the last computation to the present contents of memory

Subtract(M-) : Subtract the result of the last computation to the present contents of memory

Store(MS) : Save a value in memory for later use.

Recall(MR) : Retrieve stored values for subsequent calculations.

Clear(MC) : Pressing the "Memory Clear" button (MC) resets the memory register to zero. Any previously stored value in the memory register is erased.

## **5.3 Error Handling:**

### **5.3.1 Input Validation:**

Requirement: Validate user inputs to prevent calculation errors.

Acceptance Criteria:

Handle invalid inputs such as non-numeric characters or division by zero.

Display appropriate error messages for invalid inputs.

### **5.3.2 Precision Handling:**

Requirement: Ensure precision in calculations.

Acceptance Criteria:

Maintain accuracy in calculations, especially for floating-point operations.

### **5.3.3 Output:**

Results should be displayed under the following conditions:

1. Whenever 'MR' is input (the contents of memory is displayed).
2. whenever the 'ENTER' or '=' key is pressed.

## **5.4 User Interface:**

### **5.4.1 Console-Based Interface:**

Requirement: Provide a user-friendly console interface.

Acceptance Criteria:-

Display a clear and intuitive interface for user interactions.

Guide users through input and output effectively.

## **5.5 Mode Selection:**

### **5.5.1 Degrees,Radian and Gradians Mode:**

Requirement: Allow users to switch between degrees, gradians and radians.

Acceptance Criteria:-

Enable users to toggle between degree , gradian and radian modes for trigonometric calculations.

### **5.5.2 Standard Decimal & Scientific Notation**

Requirement: Allow users to switch between 'F-E' & 'E-F' mode.

Acceptance Criteria:- Enable users to toggle between Scientific Notation and Standard Decimal notation modes and get output accordingly in the console.

## **6.Non-functional Requirements:-**

### **6.1 Usability:**

- The user interface should be intuitive and easy to navigate using keyboard inputs.
- The calculator should support a help command to provide users with information on available functions and usage.

### **6.2 Performance:**

- The calculator should respond to user inputs within 1 second for standard operations.
- Complex calculations, such as trigonometric or logarithmic functions, should be executed efficiently.

### **6.3 Compatibility:**

- The calculator should be compatible with a wide range of terminal emulators and console environments.

- It should adhere to standard input and output conventions of the chosen platform.

#### 6.4 Reliability:

- The calculator should handle erroneous inputs gracefully, providing clear error messages to the user.
- The calculator should be resistant to crashing or freezing, even when faced with unexpected inputs.

#### 6.5 Scalability:

- The calculator should be able to handle a large number of calculations without a significant decrease in performance.

#### 6.6 Maintainability:

- Code should be well-documented to facilitate future maintenance and updates.
- The software should be modular, allowing for easy addition or modification of functions.

### 7. Key Mapping

KEY FUNCTION	KEY COMBINATION
Radian Mode	ctrl+P
Degree Mode	ctrl+Q
Gradian Mode	ctrl+R
Memory Addition (M+)	ctrl+Y
Memory Recall (MR)	ctrl+B
Memory Clear (MC)	ctrl+N
Memory Store (MS)	ctrl+M

Memory Subtraction (M-)	ctrl+W
Exit	ctrl+A
Restart	ctrl+Z
F-E Mode	ctrl+E
E-F Mode	ctrl+F
Negate	y
Sin Function	s
Arcsin Function	S
Sinh Function	l
Cosec Function	g
Arccosec Function	G
Cosech Function	u
Cos Function	c
Arccos Function	C
Cosh Function	m
Sec Function	h
Arcsec Function	H
Sech Function	v
Tan Function	t
Arctan Function	T
Tanh Function	n
Cot Function	k
Arccot Function	K
Coth Function	w
Pi	p

Random Number	r
e (Exponential)	e
Log Function	L
Natural Log (ln)	O
YlogX Function	W
X^Y Function	U
MOD Function	V
$x\sqrt{Y}$ Function	X
Square Function	Q
Cube Function	#
Square Root Function	q
Cube Root Function	Z
$10^x$ Function	z
$2^x$ Function	b
Factorial Function	f
1/x Function	i
Absolute Function	A
$e^x$ Function	E
Floor Function	F
Ceil Function	B
DMS Function	d