**SOFTWARE REQUIREMENT SPECIFICATION**

**DOCUMENT**

**CALCULATOR TOOL**

**Version:** Version 2.0

**ABSTRACT**

This document is intended to be the SRS for developing the **CALCULATOR TOOL**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Title** | **Calculator Tool** | | |
| **Lead Institution** | **THE INTERNATIONAL SCHOOL - DUY TAN UNIVERSITY** | | |
| **Project Mentor** | **Mr. Nguyen Dang Quang Huy** | | |
| **Team Name** | **Team 3** | | |
| **Team Members** | **Nguyen Nam Hung** | | |
| **Nguyen Trong Khoi** | | |
| **Doan Thi Ngoc Han** | | |
| **Nguyen Tran Duc Khai** | | |
| **Do Van Hieu** | | |
| **Start Date** | Mar 01, 2025 | **End Date** | Mar 08, 2025 |

**ROPRIETARY INFORMATION**: The information contained in this document is the property of **TEAM 3**. Except as specifically authorized in writing by **TEAM 3**, the holder of this document shall keep all information contained herein confidential and shall protect the same in whole or in part from disclosure and dissemination to all third parties

**Simple Calculator**

[**1. Introduction 2**](#_heading=h.gjdgxs)

[**1.1.Purpose: 2**](#_heading=h.30j0zll)

[**1.2. Scope: 2**](#_heading=h.1fob9te)

**1.3 Intended** [**Users: 2**](#_heading=h.3znysh7)

[**2. Functional Requirements 2**](#_heading=h.2et92p0)

[**2.1 Basic Arithmetic Operations 2**](#_heading=h.tyjcwt)

[**2.2 Advanced Arithmetic Operations 3**](#_heading=h.3dy6vkm)

[**2.3 Control Functions 4**](#_heading=h.1t3h5sf)

[**2.4 Calculation History 4**](#_heading=h.4d34og8)

[**2.5 Error Handling 4**](#_heading=h.2s8eyo1)

[**3. Non-functional requirement 5**](#_heading=h.17dp8vu)

[**NFR-01: Performance 5**](#_heading=h.3rdcrjn)

[**NFR-02: Usability 5**](#_heading=h.26in1rg)

[**NFR-03: Maintainability 5**](#_heading=h.lnxbz9)

[**NFR-04: Reliability 5**](#_heading=h.35nkun2)

[**NFR-05: Compatibility 5**](#_heading=h.1ksv4uv)

[**4. Project Analyst: 5**](#_heading=h.44sinio)

[**4.1. Business Function Diagram ( BFD) 5**](#_heading=h.2jxsxqh)

[**4.2. Context Diagram 6**](#_heading=h.z337ya)

[**4.3. Use case diagram 8**](#_heading=h.3j2qqm3)

[**4.6. Use case diagram 9**](#_heading=h.1y810tw)

[**4.7. Activity Diagrams: 17**](#_heading=h.4i7ojhp)

Application Name**:** **Calculator Tool (Java-Based)**

## 1. Introduction

**1.1 Purpose:** This document defines the functional requirements for a simple calculator application that allows users to perform basic and advanced arithmetic operations with additional control functionalities.

### 1.2. Scope:

* The application supports basic arithmetic operations (addition, subtraction, multiplication, division).
* Advanced operations such as exponentiation, square root, and percentage calculation.
* Additional features include clearing inputs, backspace functionality, and maintaining a temporary calculation history.
* The application will be developed in Java.

### 1.3.Intended Users:

* General users who require a simple and efficient calculator.
* Students and professionals for quick mathematical calculations.

## 2. Functional Requirements

### 2.1 Basic Arithmetic Operations

#### FR-01: Addition (+)

* **Description:** The system shall allow users to add two numbers.
* **Input:** Two real numbers.
* **Processing:** Sum calculation result = x + y.
* **Output:** Display the sum.

#### FR-02: Subtraction (-)

* **Description:** The system shall allow users to subtract one number from another.
* **Input:** Two real numbers.
* **Processing:** Difference calculation result = x - y.
* **Output:** Display the difference.

#### FR-03: Multiplication (\*)

* **Description:** The system shall allow users to multiply two numbers.
* **Input:** Two real numbers.
* **Processing:** Product calculation result = x \* y.
* **Output:** Display the product.

#### FR-04: Division (/)

* **Description:** The system shall allow users to divide one number by another.
* **Input:** Two real numbers, where the divisor is not zero.
* **Processing:** Quotient calculation result = x / y.
* **Output:** Display the quotient or an error message if y = 0.

### 2.2 Advanced Arithmetic Operations

#### FR-05: Exponentiation (xʸ)

* **Description:** The system shall allow users to compute the power of a number.
* **Input:** Base x and exponent y.
* **Processing:** Power calculation result = xʸ.
* **Output:** Display the computed result.
* **Constraints:** If x = 0 and y ≤ 0, display an error message.

**FR-06: Square Root (√x)**

* **Description:** The system shall allow users to compute the square root of a number.
* **Input:** A real number x.
* **Processing:** Root calculation result = √x.
* **Output:** Display the computed result.
* **Constraints:** If x < 0, display an error message.

#### FR-07: Percentage (%)

* **Description:** The system shall allow users to compute the percentage of a number.
* **Input:** A real number x.
* **Processing:** Percentage calculation result = x / 100.
* **Output:** Display the computed percentage.

### 2.3 Control Functions

#### FR-08: Backspace

* **Description:** The system shall allow users to delete the last entered digit or character.
* **Input:** Pressing the "Backspace" button.
* **Processing:** Remove the last digit from the input.
* **Output:** Display the updated input.

#### FR-09: Clear (C)

* **Description:** The system shall allow users to clear the entire input.
* **Input:** Pressing the "C" button.
* **Processing:** Remove all entered characters.
* **Output:** Display an empty input field.

### 2.4 Calculation History

#### FR-10: Temporary History Storage

* **Description:** The system shall allow users to view the history of calculations within the session.
* **Input:** Perform calculations.
* **Processing:** Store calculation results temporarily.
* **Output:** Display past calculations in a text box or list box.
* **Constraints:** History is only stored during the session and is not saved after application closure.

### 2.5 Error Handling

#### FR-11: Error Messages

* **Description:** The system shall display appropriate error messages for invalid operations.
* **Conditions:**
  + Division by zero.
  + Square root of a negative number.
  + Invalid exponentiation cases (0^0, x^-y for x=0).
* **Output:** Error message displayed to the user.

**2.6 Scientific Calculations**

**FR-12: Calculate Factorial**

* **Description:** The system shall calculate the factorial of a non-negative integer input.
* **Conditions:**  
  ○ User inputs a non-negative integer n.  
  ○ User selects the factorial (!) function.
* **Output:** The result n! is displayed. If input is invalid (negative or non-integer), an error is shown.

**FR-13: Calculate Logarithm (log₁₀, ln)**

* **Description:** The system shall compute base-10 (log) and natural (ln) logarithms.
* **Conditions:**  
  ○ Input value must be greater than 0.  
  ○ User selects log or ln function.
* **Output:** The logarithmic value is displayed. For invalid input (≤ 0), an error is shown.

**FR-14: Trigonometric Functions ( sin, cos , tan , cot )**

* **Description:** The system shall evaluate trigonometric functions including sin, cos, tan, and cot.
* **Conditions:**  
  ○ Input is a valid number.  
  ○ Unit mode (degrees or radians) is selected.
* **Output:** Function result is shown. For undefined values (e.g., tan(90°)), an error is displayed.

**FR-15: Convert degrees radians**

* **Description:** The system shall convert angles between degrees and radians.
* **Conditions:**  
  ○ Input is a valid numerical angle.  
  ○ User selects "deg↔rad" toggle.
* **Output:** Converted angle is displayed in the other unit.

**FR-16: Evaluate Complex Expression**

* **Description:** The system shall evaluate complex expressions involving functions, operators, and parentheses.
* **Conditions:**  
  ○ Input follows valid mathematical syntax.
* **Output:** Expression result is displayed. If syntax is invalid, an error message is shown.

**2.7 Calculation History**

**FR-17: Save History to File**

* **Description:** The system shall allow users to export calculation history.
* **Conditions:**  
  ○ History exists.  
  ○ User selects export format (.txt, .csv, .json).
* **Output:** History file is saved to specified location.

**FR-18: Search History**

* **Description:** The system shall support keyword-based search in history.
* **Conditions:**  
  ○ History contains entries.  
  ○ User provides a keyword.
* **Output:** Matching history entries are shown.

**FR-19: Delete History Entry**

* **Description:** The system shall allow deletion of specific history entries.
* **Conditions:**  
  ○ User selects a valid history entry.
* **Output:** Entry is removed from history. If not found, an error is displayed.

**2.8 UI Customization**

### ****FR-20: Dark/Light Mode****

* **Description:** The system shall support switching between light and dark UI themes.
* **Conditions:**  
  ○ User toggles the display mode.
* **Output:** Interface updates to selected theme.

### ****FR-21: Change Font****

* **Description:** The system shall allow users to change the font used in the interface.
* **Conditions:**  
  ○ User selects a font from the provided list.
* **Output:** UI updates with selected font.

### ****FR-22: Change UI Colors****

* **Description:** The system shall allow customization of UI color themes.
* **Conditions:**  
  ○ User selects primary, background, or text color.
* **Output:** Color scheme is updated accordingly.

### ****FR-23: Support Keyboard Shortcuts****

* **Description:** The system shall support keyboard shortcuts for key operations.
* **Conditions:**  
  ○ Shortcuts include:
  + Ctrl+C to copy result
  + Ctrl+V to paste
  + Ctrl+Enter to calculate
* **Output:** Shortcut actions are executed as expected.

**2.9 Advance Controls & Result Interaction**

### ****FR-24: Delete One Character****

* **Description:** The system shall delete the character at the current cursor position.
* **Conditions:**  
  ○ Cursor is placed within input.  
  ○ User presses Delete.
* **Output:** Character after cursor is removed.

### ****FR-25: Delete Character to the Left of Cursor****

* **Description:** The system shall delete the character to the left of the cursor.
* **Conditions:**  
  ○ Cursor is placed within input.  
  ○ User presses Backspace.
* **Output:** Character before cursor is removed.

### ****FR-26: Delete Character to the Right of Cursor****

* **Description:** The system shall delete the character to the right of the cursor (if any).
* **Conditions:**  
  ○ Cursor is placed before a character.  
  ○ User presses Delete.
* **Output:** Character is removed from input field.

### ****FR-27: Clear Entry****

* **Description:** The system shall clear the current input without affecting history.
* **Conditions:**  
  ○ User presses CE (Clear Entry).
* **Output:** Input field is reset to empty.

### ****FR-28: Copy Result****

* **Description:** The system shall copy the last result to the clipboard.
* **Conditions:**  
  ○ User clicks "Copy" or presses Ctrl+C.
* **Output:** Result is saved to clipboard.

### ****FR-29: Paste Result****

* **Description:** The system shall paste the last result into the input field.
* **Conditions:**  
  ○ Clipboard contains a valid result.  
  ○ User clicks "Paste" or presses Ctrl+V.
* **Output:** Result is inserted into current expression.

## 3. Non-Functional Requirement

### NFR-01: Performance

* The system shall respond to user inputs within 500 milliseconds.
* The system shall be able to handle at least 100 calculations per minute without performance degradation.

### NFR-02: Usability

* The user interface shall be intuitive, with clearly labeled buttons for all operations.
* The interface shall support both mouse and keyboard inputs.

### NFR-03: Maintainability

* The system shall be modular, allowing easy extension for additional operations.
* The code shall be well-documented for future maintenance and upgrades.

### NFR-04: Reliability

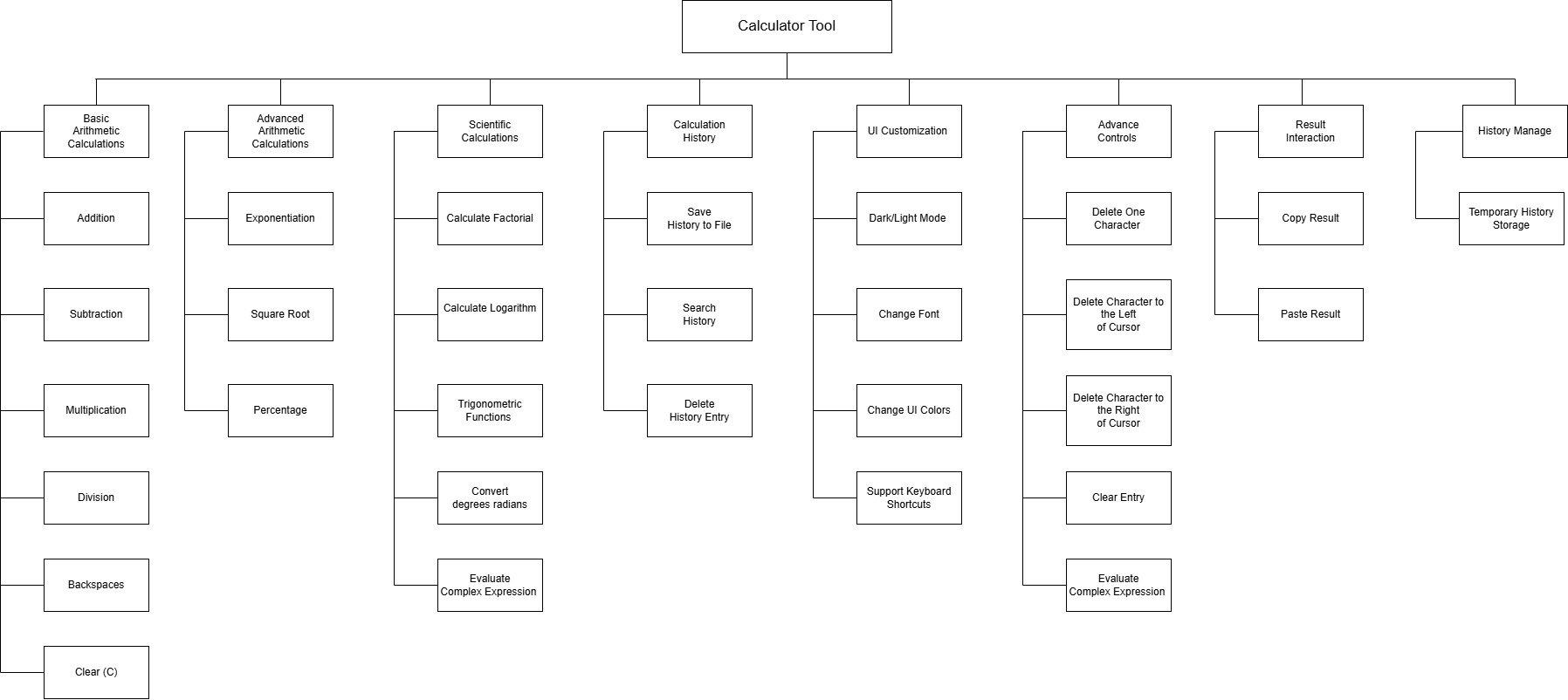
* The system shall handle invalid inputs gracefully without crashing.
* The system shall recover smoothly from unexpected errors.

### NFR-05: Compatibility

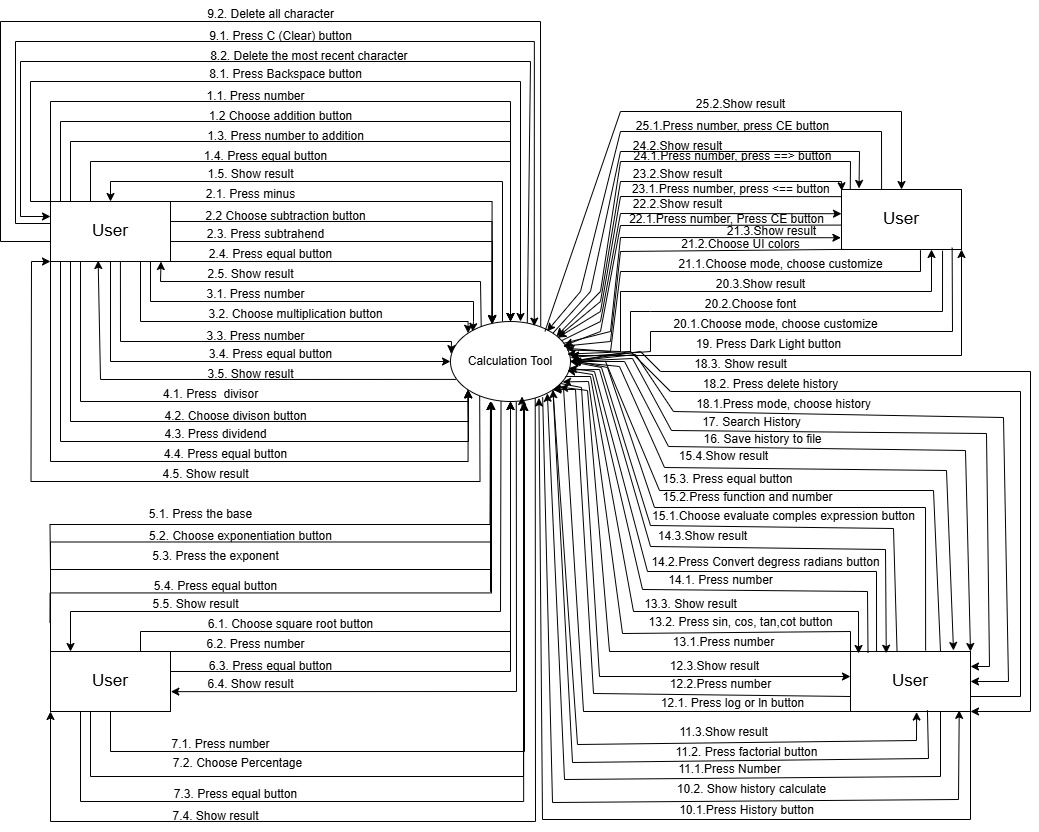
* The system shall run on Windows, macOS, and Linux environments with Java installed.
* The system shall work with different screen resolutions without distortion.

## 4. Project Analyst :

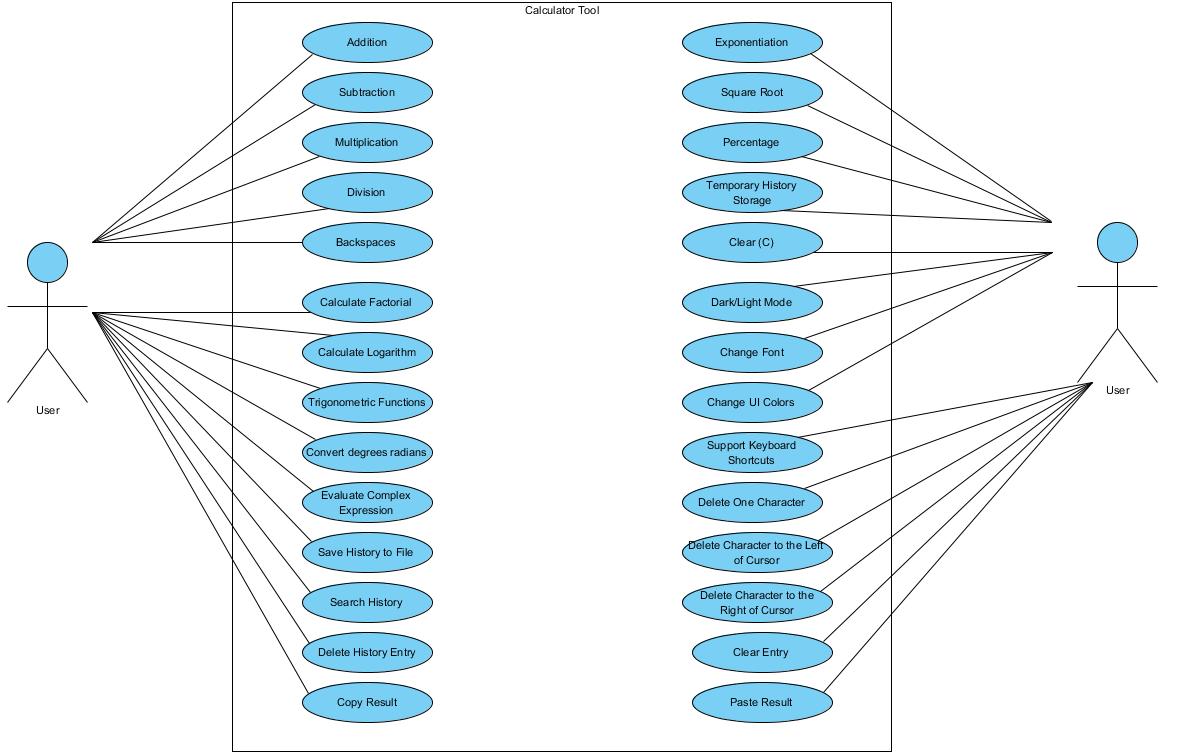
### 4.1. Business Function Diagram ( BFD ):



### 4.2. Context Diagram

****

### 4.3. Use case diagram

****

### 4.6. Use case diagram

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.01 | | | | |
| Use case name | Addition | | | | |
| Create by | Ngoc Han | | **Last updated by** | | Ngoc Han |
| Date created | March 4, 2025 | | **Date last updated** | | March 4, 2025 |
| Actor | User | | | | |
| Description | This use case describes how a user performs an addition operation in the calculator. | | | | |
| Trigger | User inputs two numbers and selects the addition operation (+). | | | | |
| Pre-condition | The calculator application is running. | | | | |
| Post-condition | The result of the addition is displayed on the screen. | | | | |
| Main Success Scenario: | **Step** | **Actor Action** | | **System Response** | |
| 1 | User enters the first number | | The number appears on the display | |
| 2 | User presses the "+" button | | The system registers the addition operation | |
| 3 | User enters the second number | | The number appears on the display | |
| 4 | User presses the "=" button | | The system calculates and displays the result | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 3.1 | User presses "=" without entering a second number | | System displays an error message "Math ERROR" | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User does not input any number and presses "=" | | System displays "Math ERROR " | |
| Priority | High | | | | |
| Business rule | N/A | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.02 | | | | |
| Use case name | Subtraction | | | | |
| Create by | Ngoc Han | | **Last updated by** | | Ngoc Han |
| Date created | March 4, 2025 | | **Date last updated** | | March 4, 2025 |
| Actor | User | | | | |
| Description | This use case describes how a user performs a subtraction operation in the calculator. | | | | |
| Trigger | User inputs two numbers and selects the subtraction operation (-). | | | | |
| Pre-condition | The calculator application is running. | | | | |
| Post-condition | The result of the subtraction is displayed on the screen. | | | | |
| Main Success Scenario: | **Step** | **Actor Action** | | **System Response** | |
| 1 | User enters the first number | | The number appears on the display | |
| 2 | User presses the "-" button | | The system registers the subtraction operation | |
| 3 | User enters the second number | | The number appears on the display | |
| 4 | User presses the "=" button | | The system calculates and displays the result | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 3.1 | User presses "=" without entering a second number | | System displays an error message " Math ERROR " | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User does not input any number and presses "=" | | System displays " Math ERROR" | |
| Priority | High | | | | |
| Business rule | N/A | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.03 | | | | |
| Use case name | Multiplication | | | | |
| Create by | Ngoc Han | | **Last updated by** | | Ngoc Han |
| Date created | March 4, 2025 | | **Date last updated** | | March 4, 2025 |
| Actor | User | | | | |
| Description | This use case describes how a user performs a multiplication operation in the calculator. | | | | |
| Trigger | User inputs two numbers and selects multiplication (x). | | | | |
| Pre-condition | The calculator application is running. | | | | |
| Post-condition | The result of the multiplication is displayed on the screen. | | | | |
| Main Success Scenario: | **Step** | **Actor Action** | | **System Response** | |
| 1 | User enters the first number | | The number appears on the display | |
| 2 | User presses the "x" button | | The system registers the multiplication operation | |
| 3 | User enters the second number | | The number appears on the display | |
| 4 | User presses "=" button | | The system calculates and displays the result | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 3.1 | User presses "=" without entering a second number | | System displays an error message " Math ERROR " | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User does not input any number and presses "=" | | System displays " Math ERROR" | |
| Priority | High | | | | |
| Business rule | N/A | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.04 | | | | |
| Use case name | Division | | | | |
| Create by | Ngoc Han | | **Last updated by** | | Ngoc Han |
| Date created | March 4, 2025 | | **Date last updated** | | March 4, 2025 |
| Actor | User | | | | |
| Description | This use case describes how a user performs a division operation in the calculator. | | | | |
| Trigger | User inputs two numbers and selects division (÷). | | | | |
| Pre-condition | The calculator application is running. | | | | |
| Post-condition | The result of the division is displayed on the screen. | | | | |
| Main Success Scenario: | **Step** | **Actor Action** | | **System Response** | |
| 1 | User enters the first number | | The number appears on the display | |
| 2 | User presses the "÷" button | | The system registers the division operation | |
| 3 | User enters the second number | | The number appears on the display | |
| 4 | User presses "=" button | | The system calculates and displays the result | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 3.1 | User presses "=" without entering a second number | | System displays an error message " Math ERROR " | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User does not input any number and presses "=" | | System displays " Math ERROR" | |
| 3.2 | User inputs zero as the second number | | System displays " Math ERROR " | |
| Priority | High | | | | |
| Business rule | N/A | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.05 | | | | |
| Use case name | Exponentiation | | | | |
| Create by | Ngoc Han | | **Last updated by** | | Ngoc Han |
| Date created | March 4, 2025 | | **Date last updated** | | March 4, 2025 |
| Actor | User | | | | |
| Description | This use case describes how a user performs exponentiation on two numbers using the calculator. | | | | |
| Trigger | User inputs a base number, selects the "^/√" button and then chooses the exponentiation (^) button, then inputs the exponent number. | | | | |
| Pre-condition | The user must enter valid numerical values. | | | | |
| Post-condition | The calculator displays the correct exponentiation result. | | | | |
| Main Success Scenario: | **Step** | **Actor Action** | | **System Response** | |
| 1 | User enters the base number | | Number appears on the display | |
| 2 | User presses the "^/√" button | | The system displays options for exponentiation (^) and square root (√) | |
| 3 | User presses the exponentiation (^) key | | The exponentiation symbol appears | |
| 4 | User enters the exponent number | | The full expression appears | |
| 5 | User presses "=" key | | System calculates and displays the result | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 4.1 | User does not enter a number | | System displays " Math ERROR" | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User does not input any number and presses "=" | | System displays " Math ERROR" | |
| Priority | High | | | | |
| Business rule | N/A | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.06 | | | | |
| Use case name | Square Root | | | | |
| Create by | Ngoc Han | | **Last updated by** | | Ngoc Han |
| Date created | March 4, 2025 | | **Date last updated** | | March 4, 2025 |
| Actor | User | | | | |
| Description | User selects the "^/√" button and then chooses the square root (√) function before entering a number. | | | | |
| Trigger | User inputs a number and presses the square root (√) button. | | | | |
| Pre-condition | The user must enter valid numerical values. | | | | |
| Post-condition | The calculator displays the correct square root result. | | | | |
| Main Success Scenario: | **Step** | **Actor Action** | | **System Response** | |
| 1 | User presses the "^/√" button | | The system displays options for exponentiation (^) and square root (√) | |
| 2 | User selects the square root (√) function | | The system prepares for square root input | |
| 3 | User enters a number | | Number appears on the display | |
| 4 | User presses "=" key | | System calculates and displays the result | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 3.1 | User does not enter a number | | System displays " Math ERROR" | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User does not input any number and presses "=" | | System displays " Math ERROR" | |
| 3.2 | User attempts square root of negative number | | System displays "Error: Invalid Input" | |
| Priority | High | | | | |
| Business rule | N/A | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.07 | | | | |
| Use case name | Percentage Calculation | | | | |
| Create by | Ngoc Han | | **Last updated by** | | Ngoc Han |
| Date created | March 4, 2025 | | **Date last updated** | | March 4, 2025 |
| Actor | User | | | | |
| Description | This use case describes how a user performs a percentage calculation in the calculator. | | | | |
| Trigger | User inputs a number and selects the percentage operation (%). | | | | |
| Pre-condition | The calculator application is running. | | | | |
| Post-condition | The result of the percentage calculation is displayed on the screen. | | | | |
| Main Success Scenario: | **Step** | **Actor Action** | | **System Response** | |
| 1 | User enters a number | | The number appears on the display | |
| 2 | User presses the "%" button | | The system registers the percentage operation | |
| 3 | User presses the "=" button | | The system calculates and displays the percentage result | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 2.1 | User enter a negative number | | The system still calculates a/100 and displays the negative result | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User does not input any number and presses "=" | | System displays " Math ERROR" | |
| Priority | High | | | | |
| Business rule | N/A | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.08 | | | | |
| Use case name | Backspace | | | | |
| Create by | Ngoc Han | | **Last updated by** | | Ngoc Han |
| Date created | March 4, 2025 | | **Date last updated** | | March 4, 2025 |
| Actor | User | | | | |
| Description | This use case describes how a user deletes one character from the current input. | | | | |
| Trigger | User presses the "←" button. | | | | |
| Pre-condition | The calculator must have at least one character entered. | | | | |
| Post-condition | One character is deleted from the current input. | | | | |
| Main Success Scenario: | **Step** | **Actor Action** | | **System Response** | |
| 1 | User presses "←" | | The last character is removed from input | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User does not enter a number | | System waits for valid input, does not calculate | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1 | System encounters an error | | System displays "Error" message | |
| Priority | High | | | | |
| Business rule | N/A | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.09 | | | | |
| Use case name | Clear(C) | | | | |
| Create by | Ngoc Han | | **Last updated by** | | Ngoc Han |
| Date created | March 4, 2025 | | **Date last updated** | | March 4, 2025 |
| Actor | User | | | | |
| Description | This use case describes how a user clears all input from the calculator. | | | | |
| Trigger | User presses the "C" button. | | | | |
| Pre-condition | The calculator must have some input displayed. | | | | |
| Post-condition | All input is cleared, and the display resets to zero. | | | | |
| Main Success Scenario: | **Step** | **Actor Action** | | **System Response** | |
| 1 | User presses "C" | | The system clears all input and resets to "0" | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User presses "C" with no input | | System remains at "0" | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1 | System encounters an error | | System displays "Error" message | |
| Priority | High | | | | |
| Business rule | N/A | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.10 | | | | |
| Use case name | Temporary History Storage | | | | |
| Create by | Ngoc Han | | **Last updated by** | | Ngoc Han |
| Date created | March 4, 2025 | | **Date last updated** | | March 4, 2025 |
| Actor | User | | | | |
| Description | This use case describes how a user can view the history of previous calculations performed on the calculator. | | | | |
| Trigger | User presses the history button. | | | | |
| Pre-condition | The calculator must have previously executed calculations stored in memory. | | | | |
| Post-condition | The calculator displays the history of calculations. | | | | |
| Main Success Scenario: | **Step** | **Actor Action** | | **System Response** | |
| 1 | User presses the history button | | The system checks if there is any stored history | |
| 2 |  | | The system displays the stored history | |
| 3 | User can view the history | |  | |
| 4 | User selects "Clear History" | | System deletes all stored history and updates the display to an empty history screen | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 2.1 | No history is available | | The system displays an empty history screen | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1 | System encounters an error | | System displays "Error" message | |
| Priority | High | | | | |
| Business rule | N/A | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.11 | | | | |
| Use case name | Factorial | | | | |
| Create by | Ngoc Han | | **Last updated by** | | Ngoc Han |
| Date created | April 18, 2025 | | **Date last updated** | | April 18, 2025 |
| Actor | User | | | | |
| Description | This use case describes how a user calculates the factorial of a number using the calculator. | | | | |
| Trigger | User inputs a number and selects the factorial operation (!) | | | | |
| Pre-condition | The calculator application is running. | | | | |
| Post-condition | The result of the factorial is displayed on the screen. | | | | |
| Main Success Scenario: | **Step** | **Actor Action** | | **System Response** | |
| 1 | |  | | --- | | User enters a non-negative integer | | | |  | | --- | | User enters a non-negative integer | | |
| 2 | |  | | --- | | User presses the "!" button | | | |  | | --- | | User presses the "!" button | | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User enters a negative number and presses "!" | | System displays an error message "Math ERROR" | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1.2 | User does not input any number and presses "!" | | System displays "Math ERROR " | |
| Priority | High | | | | |
| Business rule | N/A | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.12 | | | | |
| Use case name | Logarithms | | | | |
| Create by | Ngoc Han | | **Last updated by** | | Ngoc Han |
| Date created | April 18, 2025 | | **Date last updated** | | April 18, 2025 |
| Actor | User | | | | |
| Description | This use case describes how a user calculates log₁₀ or natural log (ln) of a number using the calculator. | | | | |
| Trigger | User inputs a number and selects log₁₀ or natural log (ln) operation | | | | |
| Pre-condition | The calculator application is running. | | | | |
| Post-condition | The result of the logarithms is displayed on the screen. | | | | |
| Main Success Scenario: | **Step** | **Actor Action** | | **System Response** | |
| 1 | |  | | --- | | User enters a positive number | | | |  | | --- | | User enters a positive number | | |
| 2 | |  | | --- | | User presses "log" or "ln" button | | | |  | | --- | | User presses "log" or "ln" button | | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User inputs zero or negative number and presses "log" or "ln" | | System displays an error message "Math ERROR" | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1.2 | User presses "log" or "ln" without entering a number | | System displays "Math ERROR " | |
| Priority | High | | | | |
| Business rule | N/A | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.13 | | | | |
| Use case name | Calculate Trigonometric | | | | |
| Create by | Ngoc Han | | **Last updated by** | | Ngoc Han |
| Date created | April 18, 2025 | | **Date last updated** | | April 18, 2025 |
| Actor | User | | | | |
| Description | This use case describes how a user calculates trigonometric values (sin, cos, tan, cot) using the calculator. | | | | |
| Trigger | User enters a number and selects a trigonometric function. | | | | |
| Pre-condition | The calculator application is running. | | | | |
| Post-condition | The result of the trigonometric function is displayed on the screen. | | | | |
| Main Success Scenario: | 1. sin(x) | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User enters a number (angle) | | The number appears on the display | |
| 1.2 | |  | | --- | | User presses the "sin" button | | | |  | | --- | | The system prepares the sin(x) operation | | |
| 1.3 | User presses "=" | | The system calculates and displays sin(x) | |
| 2. cos(x) | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 2.1 | User enters a number (angle) | | The number appears on the display | |
| 2.2 | |  | | --- | | User presses the "cos" button | | | |  | | --- | | The system prepares the cos(x) operation | | |
| 2.3 | User presses "=" | | The system calculates and displays cos(x) | |
| 3. tan(x) | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 3.1 | User enters a number (angle) | | The number appears on the display | |
| 3.2 | |  | | --- | | User presses the "tan" button | | | |  | | --- | | The system prepares the tan(x) operation | | |
| 3.3 | User presses "=" | | The system calculates and displays tan(x) | |
| 4. cot(x) | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 4.1 | User enters a number (angle) | | The number appears on the display | |
| 4.2 | |  | | --- | | User presses the "cot" button | | | |  | | --- | | The system prepares the cot(x) operation | | |
|  | 4.3 | User presses "=" | | The system calculates and displays cot(x) | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User presses "=" without selecting a trig function | | System display the entered number as-is | |
| 1.2 | User selects a trig function but doesn't press "=" | | System displays function name with no result | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User enters value leading to undefined result | | System displays "Math ERROR " | |
| Priority | High | | | | |
| Business rule | N/A | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.14 | | | | |
| Use case name | Convert Degrees ↔ Radians | | | | |
| Create by | Ngoc Han | | Create by | | Ngoc Han |
| Date created | April 18, 2025 | | Date created | | April 18, 2025 |
| Actor | User | | | | |
| Description | This use case describes how a user converts between degrees and radians. | | | | |
| Trigger | User inputs a number and selects conversion operation | | | | |
| Pre-condition | The calculator application is running. | | | | |
| Post-condition | The result is displayed on the screen. | | | | |
| Main Success Scenario: | Step | Actor Action | | System Response | |
| 1 | |  | | --- | | User enters an angle | | | |  | | --- | | User enters an angle | | |
| 2 | |  | | --- | | User selects "Convert to Radians" or "Convert to Degrees" | | | |  | | --- | | The system performs conversion and displays the result | | |
| Exception | Step | Actor Action | | System Response | |
| 1.1 | |  | | --- | | User presses conversion without input |  |  | | --- | |  | | | System displays "Math ERROR " | |
| Priority | High | | | | |
| Business rule | The calculator must follow standard operator precedence (BODMAS/PEMDAS). | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.15 | | | | |
| Use case name | |  | | --- | | Evaluate Complex Expressions |  |  | | --- | |  | | | | | |
| Create by | Ngoc Han | | **Create by** | | Ngoc Han |
| Date created | April 18, 2025 | | **Date created** | | April 18, 2025 |
| Actor | User | | | | |
| Description | This use case describes how a user inputs a mathematical expression and the calculator evaluates the result. | | | | |
| Trigger | User inputs a valid expression and presses "=" | | | | |
| Pre-condition | The calculator application is running. | | | | |
| Post-condition | The result is displayed on the screen. | | | | |
| Main Success Scenario: | **Step** | **Actor Action** | | **System Response** | |
| 1 | User inputs a valid expression | | Expression appears on the display | |
| 2 | User presses “=” button | | |  | | --- | | The system evaluates and displays the result | | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User inputs an invalid expression | | System displays an error message "Math ERROR" | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1.2 | |  | | --- | | User presses "=" without entering any expression |  |  | | --- | |  | | | System displays "Math ERROR " | |
| Priority | High | | | | |
| Business rule | The calculator must follow standard operator precedence (BODMAS/PEMDAS). | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.16 | | | | |
| Use case name | |  |  |  | | --- | --- | --- | | |  | | --- | | Calculation History Management |  |  | | --- | |  | |  |  | | --- | |  | | | | | |
| Create by | Ngoc Han | | **Create by** | | Ngoc Han |
| Date created | April 18, 2025 | | **Date created** | | April 18, 2025 |
| Actor | User | | | | |
| Description | This use case describes how the user manages their calculation history, including saving to file, searching, and deleting entries. | | | | |
| Trigger | User performs a calculation or interacts with the history panel. | | | | |
| Pre-condition | The calculator application is running and at least one calculation has been made. | | | | |
| Post-condition | History is updated accordingly based on user actions. | | | | |
| Main Success Scenario: | 1. Save History to File | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User selects "Save history" option | | |  | | --- | | System opens file save dialog |  |  | | --- | |  | | |
| 1.2 | User choose file format | | System saves history to chosen file type | |
| 1.3 | |  | | --- | | User confirms file path |  |  | | --- | |  | | | System saves successfully and shows confirmation | |
| 2. Search History | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 2.1 | User types keyword in search box | | |  |  |  | | --- | --- | --- | | |  | | --- | | System filters and displays matching entries |  |  | | --- | |  | |  |  | | --- | |  | | |
| 2.2 | |  | | --- | | User clicks on a result |  |  | | --- | |  | | | System highlights and shows the full calculation | |
| 3. Delete History Entry | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 3.1 | User selects a history entry | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | Entry is highlighted |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |
| 3.2 | |  |  |  | | --- | --- | --- | | |  | | --- | | User clicks "Delete" or presses  key |  |  | | --- | |  | |  |  | | --- | |  | | | System prompts for confirmation | |
| 3.3 | System prompts for confirmation | | System removes entry and updates the view | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User cancels during save or delete operation | | |  | | --- | | System closes dialog, no  changes made |  |  | | --- | |  | | |
| 2.1 | |  | | --- | | No matches found during search |  |  | | --- | |  | | | System shows message: "No results found" | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | |  | | --- | | User tries to save when no history exists |  |  | | --- | |  | | | |  | | --- | | System displays error: "No  history to save" |  |  | | --- | |  | | |
| 2.1 | |  | | --- | | File save fails due to permission  or disk error |  |  | | --- | |  | | | |  | | --- | | System shows: "Unable to  save file" |  |  | | --- | |  | | |
| 3.1 | |  | | --- | | User tries to delete a non-existent  entry |  |  | | --- | |  | | | System refreshes list and shows error | |
| Priority | High | | | | |
| Business rule | History should persist between sessions if user enables saving.  Only selected entries can be deleted.  JSON file should contain structured data for easy reuse. | | | | |

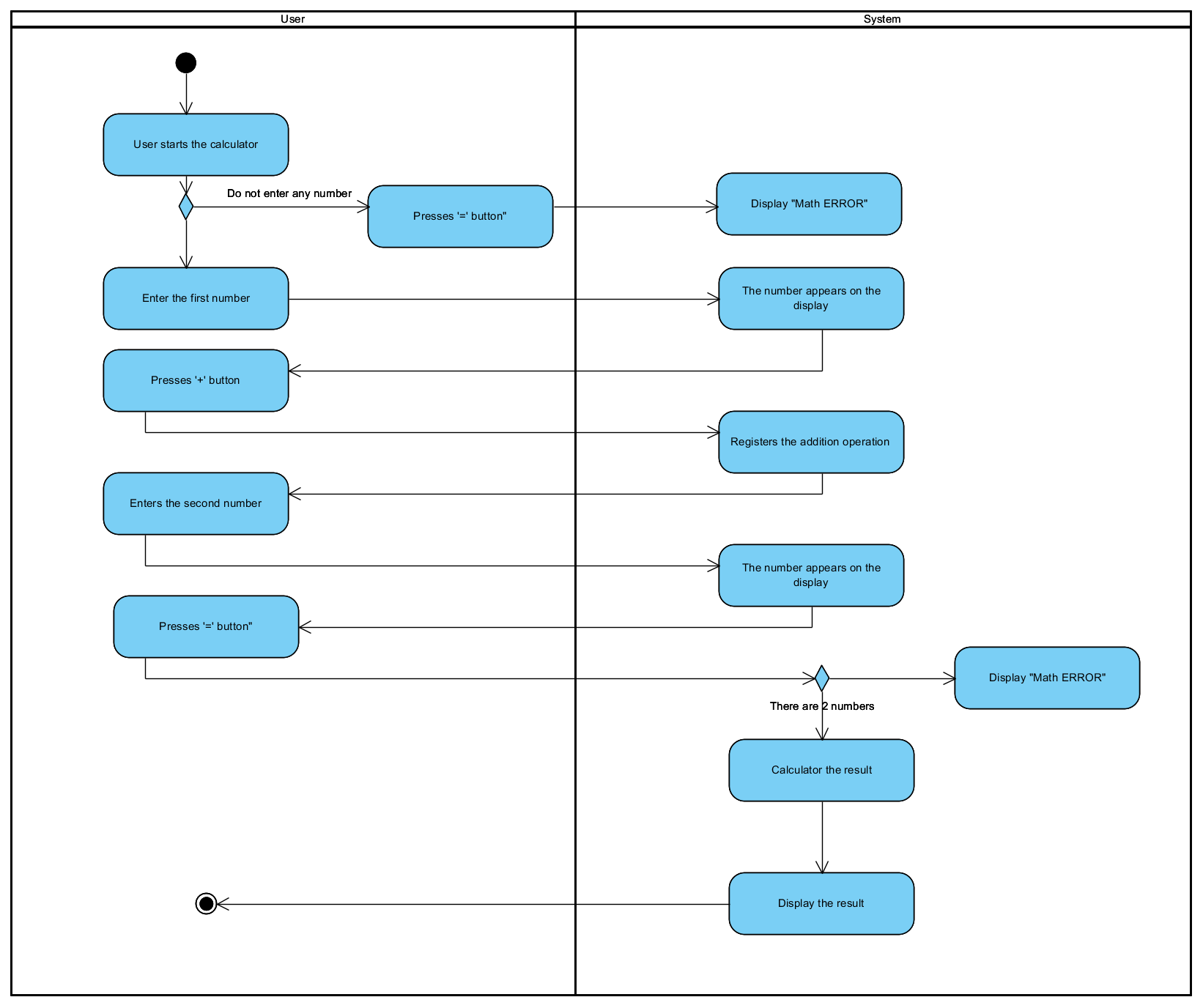
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.17 | | | | |
| Use case name | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | UI Customization |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | | | | |
| Create by | Ngoc Han | | **Create by** | | Ngoc Han |
| Date created | April 18, 2025 | | **Date created** | | April 18, 2025 |
| Actor | User | | | | |
| Description | This use case describes how a user customizes the calculator interface, including theme, font, colors, and keyboard shortcuts. | | | | |
| Trigger | |  | | --- | | User opens the Settings or UI customization menu. |  |  | | --- | |  | | | | | |
| Pre-condition | |  | | --- | | The calculator application is running. |  |  | | --- | |  | | | | | |
| Post-condition | The interface updates based on user-selected preferences. | | | | |
| Main Success Scenario: | 1. Dark/Light Mode | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User opens UI settings | | |  |  |  | | --- | --- | --- | | |  | | --- | | System displays customization options |  |  | | --- | |  | |  |  | | --- | |  | | |
| 1.2 | User selects "Dark Mode" or "Light Mode" | | System applies the selected theme instantly | |
| **2. Change Font Style** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 2.1 | User opens Font settings | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | System shows a list of available fonts |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |
| 2.2 | |  |  |  | | --- | --- | --- | | |  | | --- | | User selects a font |  |  | | --- | |  | |  |  | | --- | |  | | | System updates all text using the new font style | |
| 3. Change UI Colors | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 3.1 | |  | | --- | | User selects UI color customization |  |  | | --- | |  | | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | System displays color pickers for UI elements |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |
| 3.2 | |  |  |  | | --- | --- | --- | | |  | | --- | | User adjusts primary or accent colors |  |  | | --- | |  | |  |  | | --- | |  | | | System previews and applies changes immediately | |
| 4. Enable/Use Keyboard Shortcuts | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 4.1 | User opens keyboard shortcut settings | | |  | | --- | | System displays a list of available shortcuts |  |  | | --- | |  | | |
| 4.2 | |  | | --- | | User enables or modifies a shortcut key |  |  | | --- | |  | | | |  | | --- | | System updates and listens for new shortcut |  |  | | --- | |  | | |
| 4.3 | |  | | --- | | User uses a shortcut during calculation |  |  | | --- | |  | | | System executes the corresponding function | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User cancels customization without saving | | |  |  |  | | --- | --- | --- | | |  | | --- | | System reverts to previous settings |  |  | | --- | |  | |  |  | | --- | |  | | |
| 2.1 | |  |  |  | | --- | --- | --- | | |  | | --- | | User selects unsupported font |  |  | | --- | |  | |  |  | | --- | |  | | | System shows fallback font with a warning | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | |  |  |  | | --- | --- | --- | | |  | | --- | | User applies invalid color code |  |  | | --- | |  | |  |  | | --- | |  | | | |  |  |  | | --- | --- | --- | | |  | | --- | | System displays error message: "Invalid color" |  |  | | --- | |  | |  |  | | --- | |  | | |
| 2.1 | |  |  |  | | --- | --- | --- | | |  | | --- | | System fails to save preferences due to error |  |  | | --- | |  | |  |  | | --- | |  | | | |  |  |  | | --- | --- | --- | | |  | | --- | | System shows: "Settings not saved. Try again." |  |  | | --- | |  | |  |  | | --- | |  | | |
| 3.1 | |  |  |  | | --- | --- | --- | | |  | | --- | | Keyboard shortcut conflicts with existing shortcut |  |  | | --- | |  | |  |  | | --- | |  | | | System alerts user of conflict and prevents save | |
| Priority | High | | | | |
| Business rule | UI changes should be reversible immediately.  Changes must persist between sessions if auto-save is enabled.  Keyboard shortcuts must not override OS/system-level combinations | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.18 | | | | |
| Use case name | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | Advanced Controls |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | | | | |
| Create by | Ngoc Han | | **Create by** | | Ngoc Han |
| Date created | April 18, 2025 | | **Date created** | | April 18, 2025 |
| Actor | User | | | | |
| Description | |  | | --- | | This use case describes advanced input control functions including deleting characters and clearing input entries. |  |  | | --- | |  | | | | | |
| Trigger | |  |  |  | | --- | --- | --- | | |  | | --- | | User interacts with Backspace, cursor keys, or Clear Entry button. |  |  | | --- | |  | |  |  | | --- | |  | | | | | |
| Pre-condition | |  |  |  | | --- | --- | --- | | |  | | --- | | Calculator app is active with an on going input. |  |  | | --- | |  | |  |  | | --- | |  | | | | | |
| Post-condition | Input field is updated based on user action. | | | | |
| Main Success Scenario: | 1. **Delete One Character (Backspace)** | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 1.1 | |  | | --- | | User places cursor inside input field |  |  | | --- | |  | | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | Cursor appears at selected location |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |
| 1.2 | |  | | --- | | User presses Backspace key |  |  | | --- | |  | | | System deletes character **before** cursor | |
| **2**.**** Delete Character to the Left of Cursor (← Key) | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 2.1 | User presses ← key | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | Cursor moves one position to the left |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |
| 2.2 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | User presses Delete/Backspace |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | | System deletes character **left** of cursor | |
| 3. Delete Character to the Right of Cursor (→ Key + Del) | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 3.1 | |  |  |  | | --- | --- | --- | | |  | | --- | | User presses → key |  |  | | --- | |  | |  |  | | --- | |  | | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | Cursor moves one position  to the right |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |
| 3.2 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | User presses Delete key |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | | System deletes character **right** of cursor | |
| 4. Clear Entry (CE) | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 4.1 | User presses CE button | | |  |  |  | | --- | --- | --- | | |  | | --- | | System removes only the  current number (not the full  expression) |  |  | | --- | |  | |  |  | | --- | |  | | |
| 4.2 | |  | | --- | | User resumes input |  |  | | --- | |  | | | |  | | --- | | System allows new number to be entered |  |  | | --- | |  | | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | Cursor is at start of input, user presses Backspace | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | No change; system ignores  action |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | |
| 2.1 | |  |  |  | | --- | --- | --- | | |  | | --- | | Input is already cleared, user presses CE |  |  | | --- | |  | |  |  | | --- | |  | | | System remains unchanged | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
|  | 1.1 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | System cannot access input  field (e.g., frozen UI) |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | | |  |  |  | | --- | --- | --- | | |  | | --- | | System shows error: "Input not  available" |  |  | | --- | |  | |  |  | | --- | |  | | |
|  | 2.1 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | CE used in empty input |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | | |  |  |  | | --- | --- | --- | | |  | | --- | | System shows message:  "Nothing to clear" |  |  | | --- | |  | |  |  | | --- | |  | | |
| Priority | High | | | | |
| Business rule | Backspace only deletes 1 character at a time.  CE removes only the last complete number/entry, not operators or full expressions.  Cursor navigation must be smooth and reflect changes in real time. | | | | |

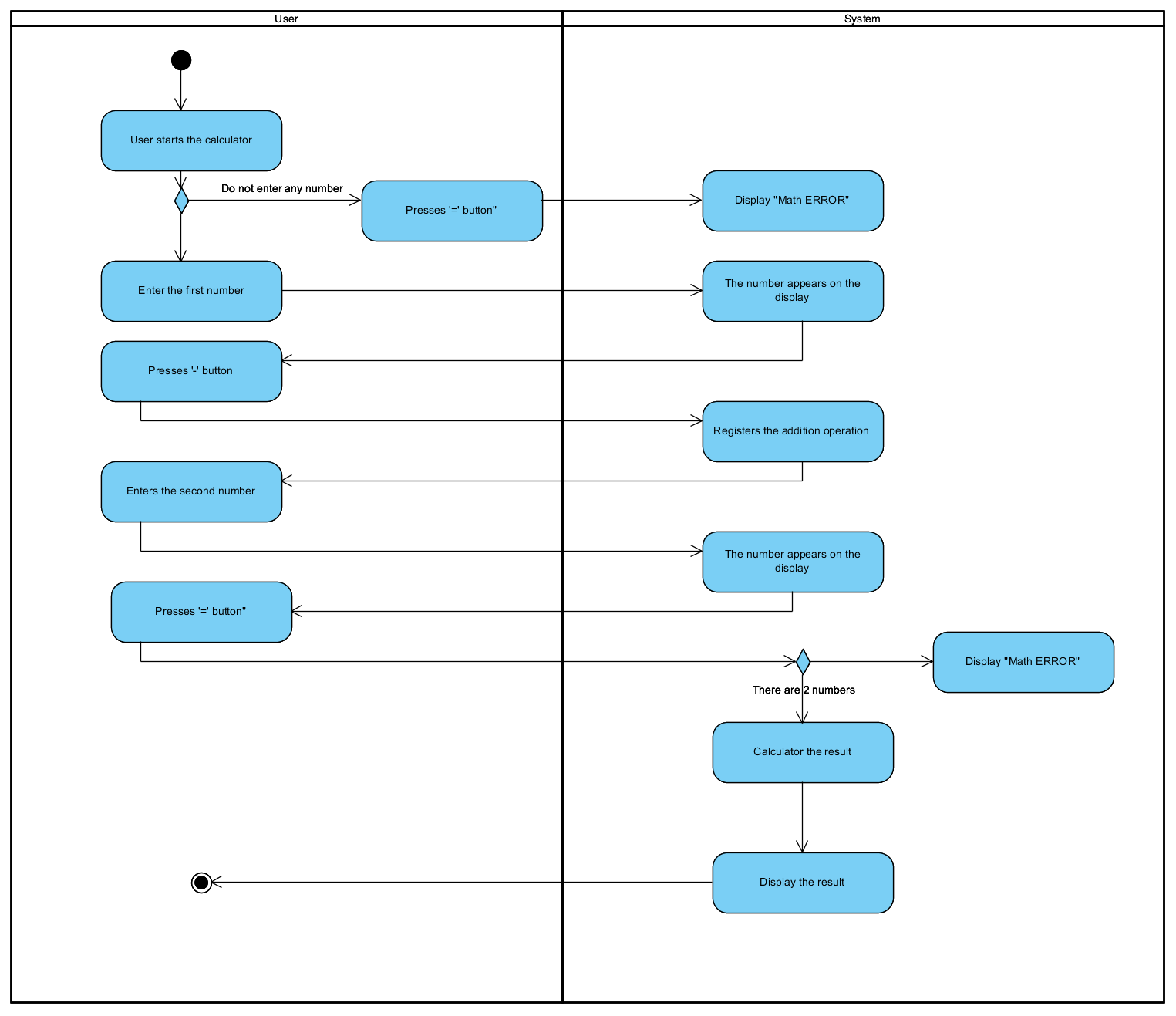
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use case ID | UC.19 | | | | |
| Use case name | |  |  |  | | --- | --- | --- | | |  | | --- | | Result Interaction |  |  | | --- | |  | |  |  | | --- | |  | | | | | |
| Create by | Ngoc Han | | **Create by** | | Ngoc Han |
| Date created | April 18, 2025 | | **Date created** | | April 18, 2025 |
| Actor | User | | | | |
| Description | This use case describes how the user can copy and paste the result from the calculator for reuse in other contexts or inputs | | | | |
| Trigger | User interacts with result field using shortcut key or button | | | | |
| Pre-condition | A calculator result is available in the result field | | | | |
| Post-condition | The result is copied to clipboard or pasted into input field | | | | |
| Main Success Scenario: | 1. Copy Result | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User clicks the "Copy" button or presses Ctrl+C | | System copies result value to clipboard | |
| 1.2 | User opens another app or stays in calculator | | Result remains in clipboard for use | |
| 2. Paste Result | | | | |
| **Step** | **Actor Action** | | **System Response** | |
| 2.1 | User clicks "Paste" or presses Ctrl+V | | |  |  |  | | --- | --- | --- | | |  | | --- | | System inserts clipboard content into input field |  |  | | --- | |  | |  |  | | --- | |  | | |
| 2.2 | |  | | --- | | User continues calculation |  |  | | --- | |  | | | Calculator processes the pasted number/expression | |
| Alternative Scenario | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | User pastes when field already has input | | |  | | --- | | System appends clipboard content to cursor position |  |  | | --- | |  | | |
| 2.1 | |  | | --- | | No matches found during search |  |  | | --- | |  | | | System shows message: "No results found" | |
| Exception | **Step** | **Actor Action** | | **System Response** | |
| 1.1 | |  | | --- | | User tries to copy with empty result |  |  | | --- | |  | | | |  | | --- | | System shows: "Nothing to  copy" |  |  | | --- | |  | | |
| 2.1 | |  | | --- | | Clipboard is empty on paste |  |  | | --- | |  | | | |  | | --- | | System shows: "Clipboard is  empty" |  |  | | --- | |  | | |
| 3.1 | |  | | --- | | Clipboard content is not a number  or valid expression |  |  | | --- | |  | | | System shows: "Invalid input" | |
| Priority | High | | | | |
| Business rule | Copy/Paste only handles valid numerical data or expressions.  Clipboard interaction must comply with OS-level permissions.  System should support both mouse and keyboard interactions. | | | | |

### 4.7. Activity Diagrams:

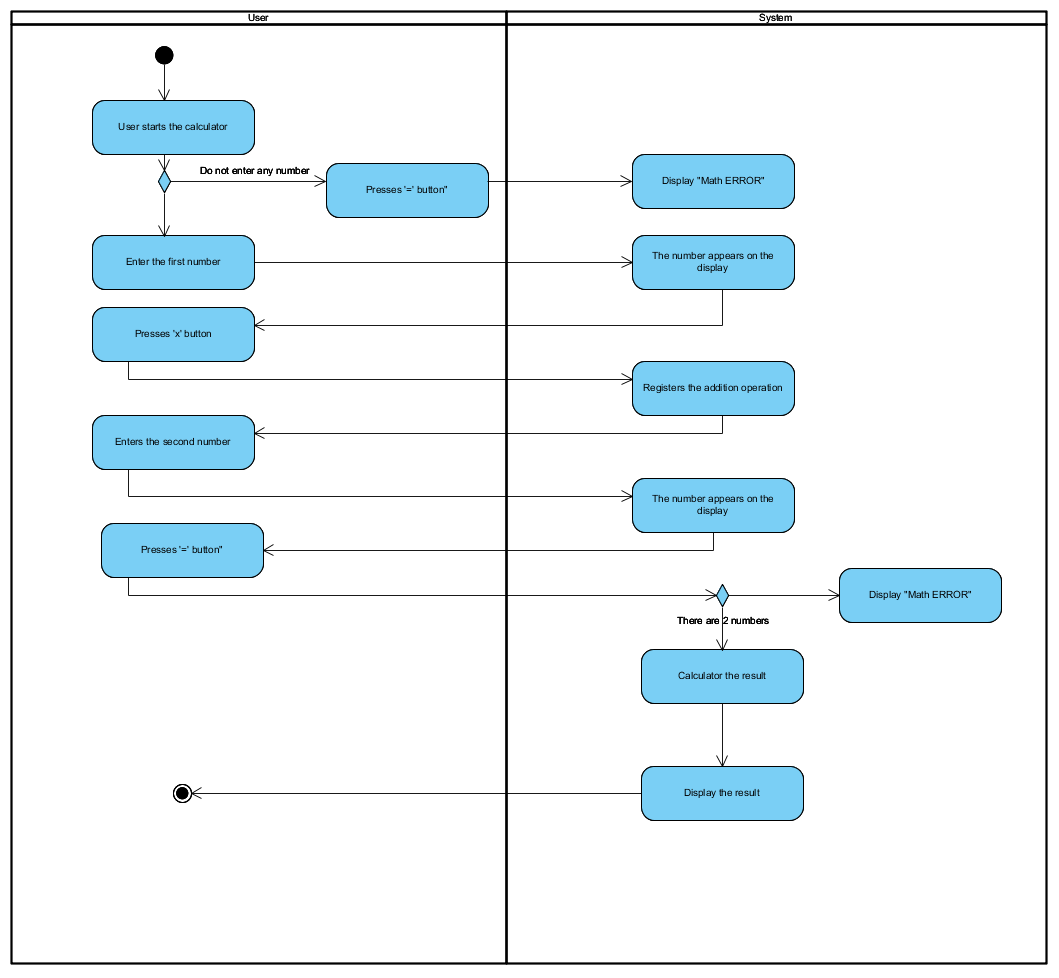
#### 1. Addition



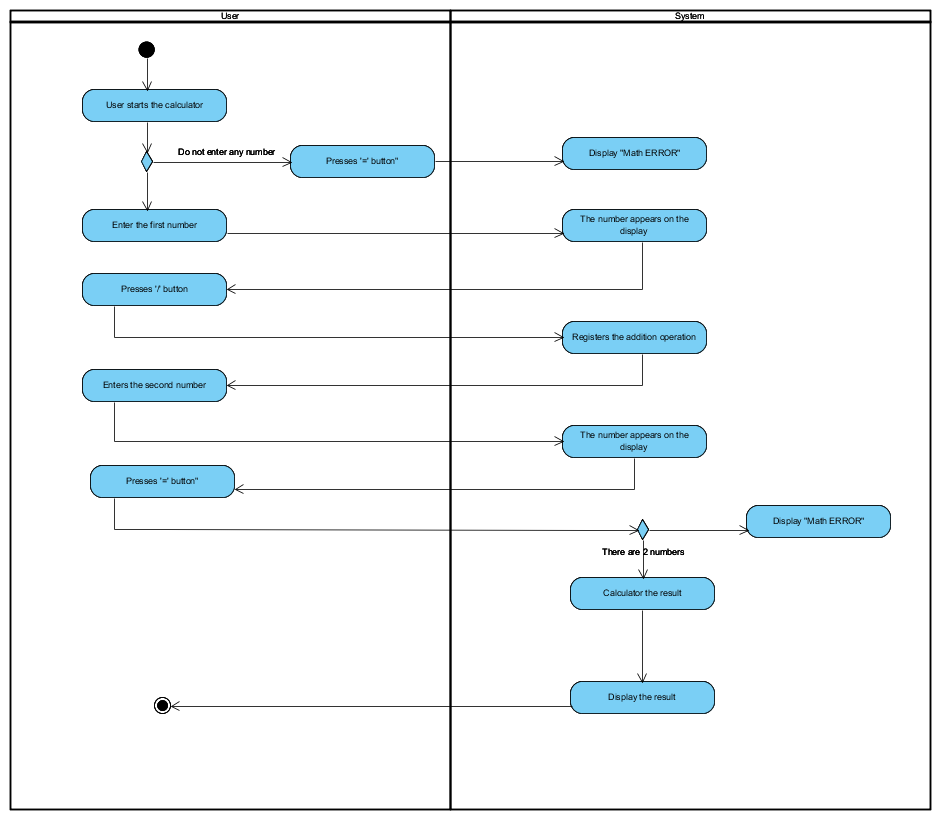
#### 2. Subtraction:



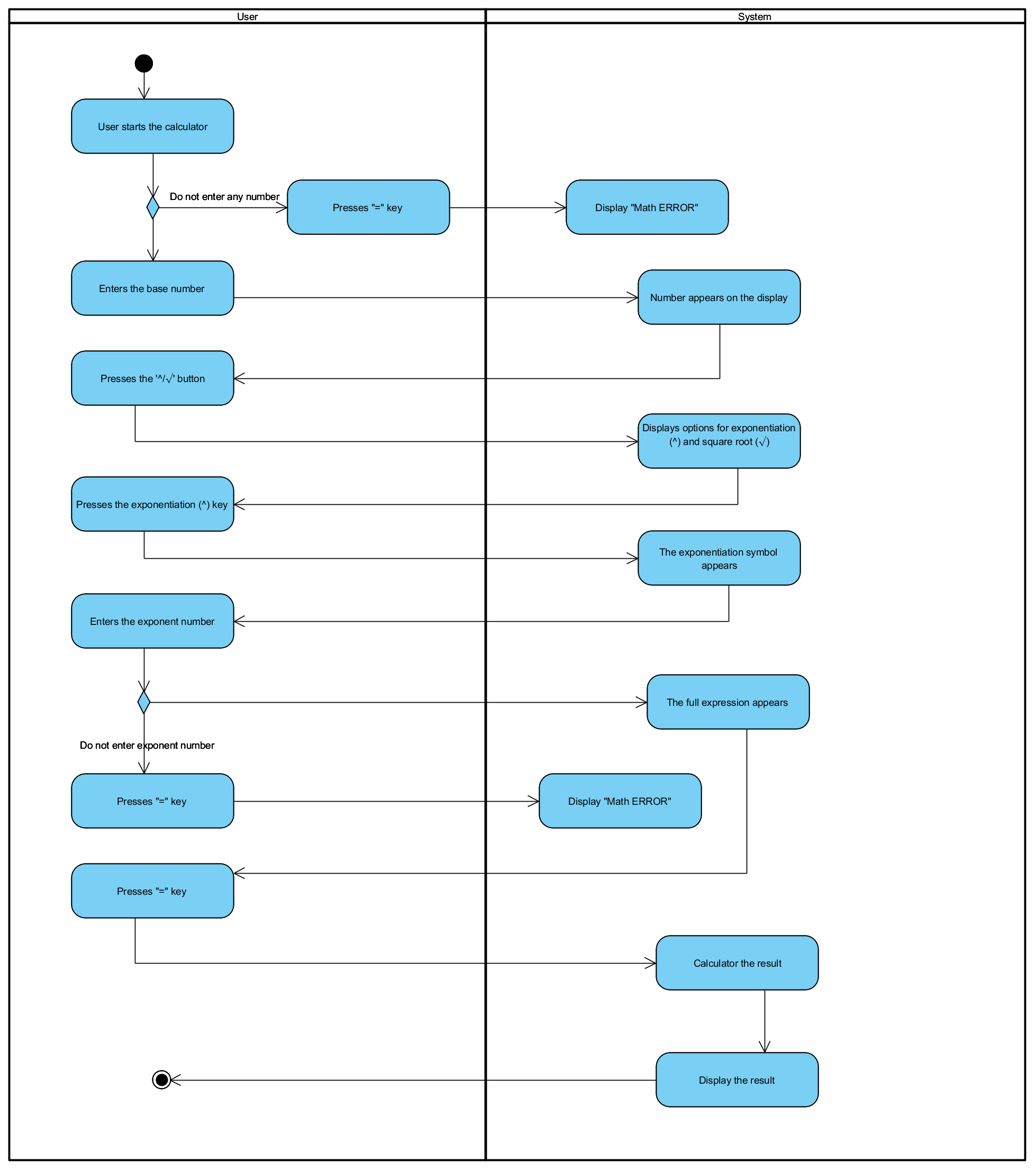
#### 3. Multiplication:



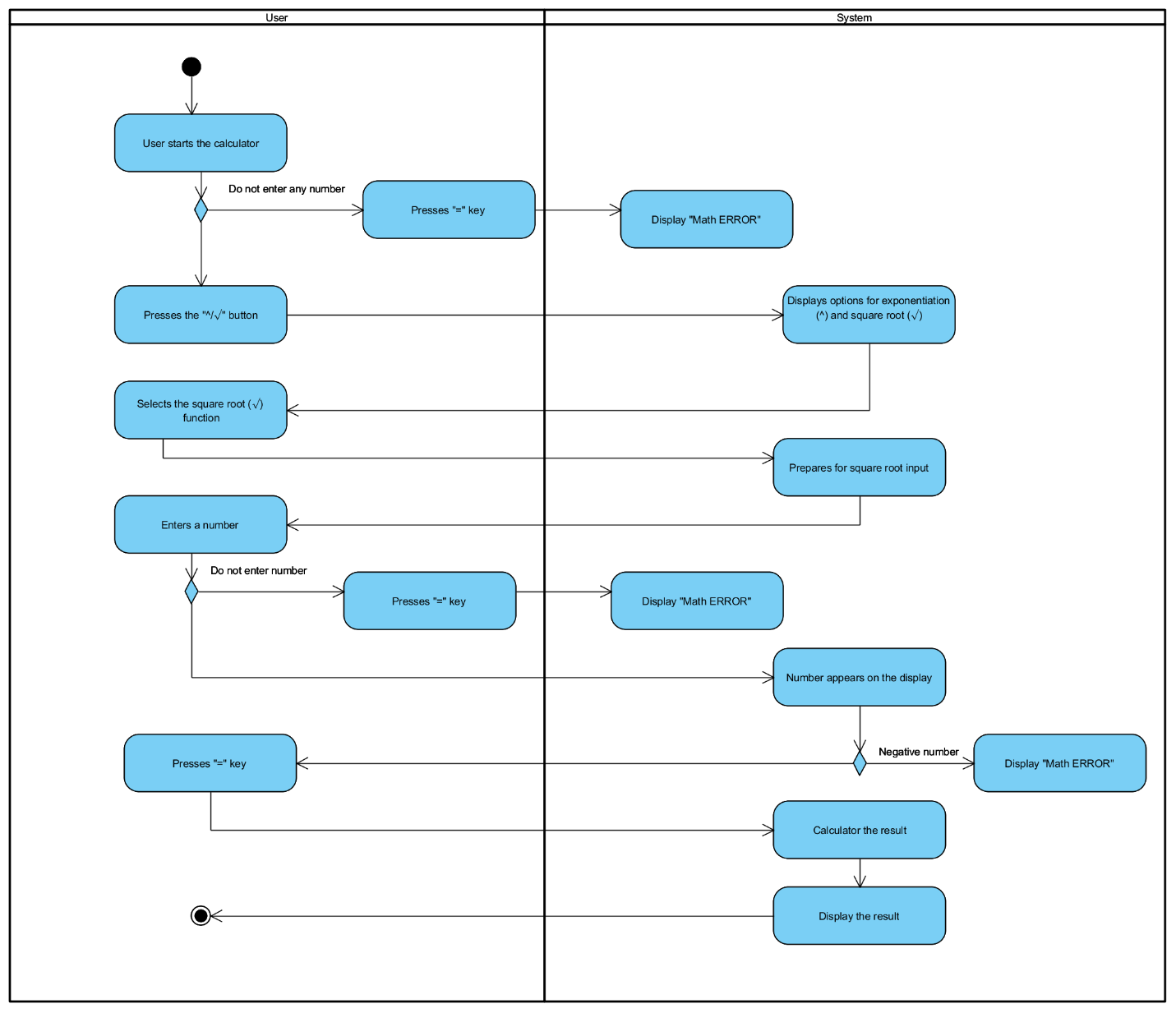
#### 4. Division



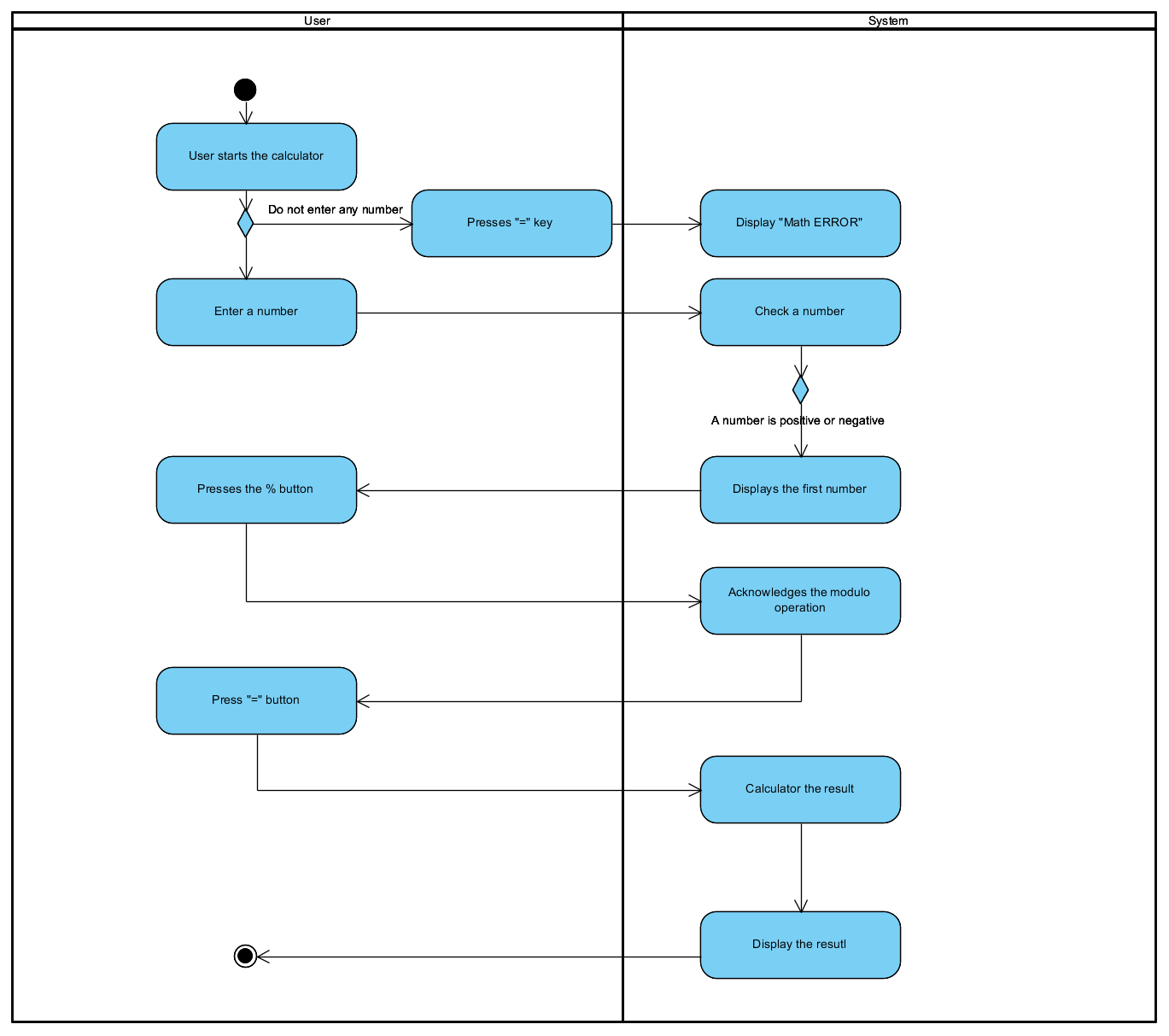
#### 5. Exponentiation



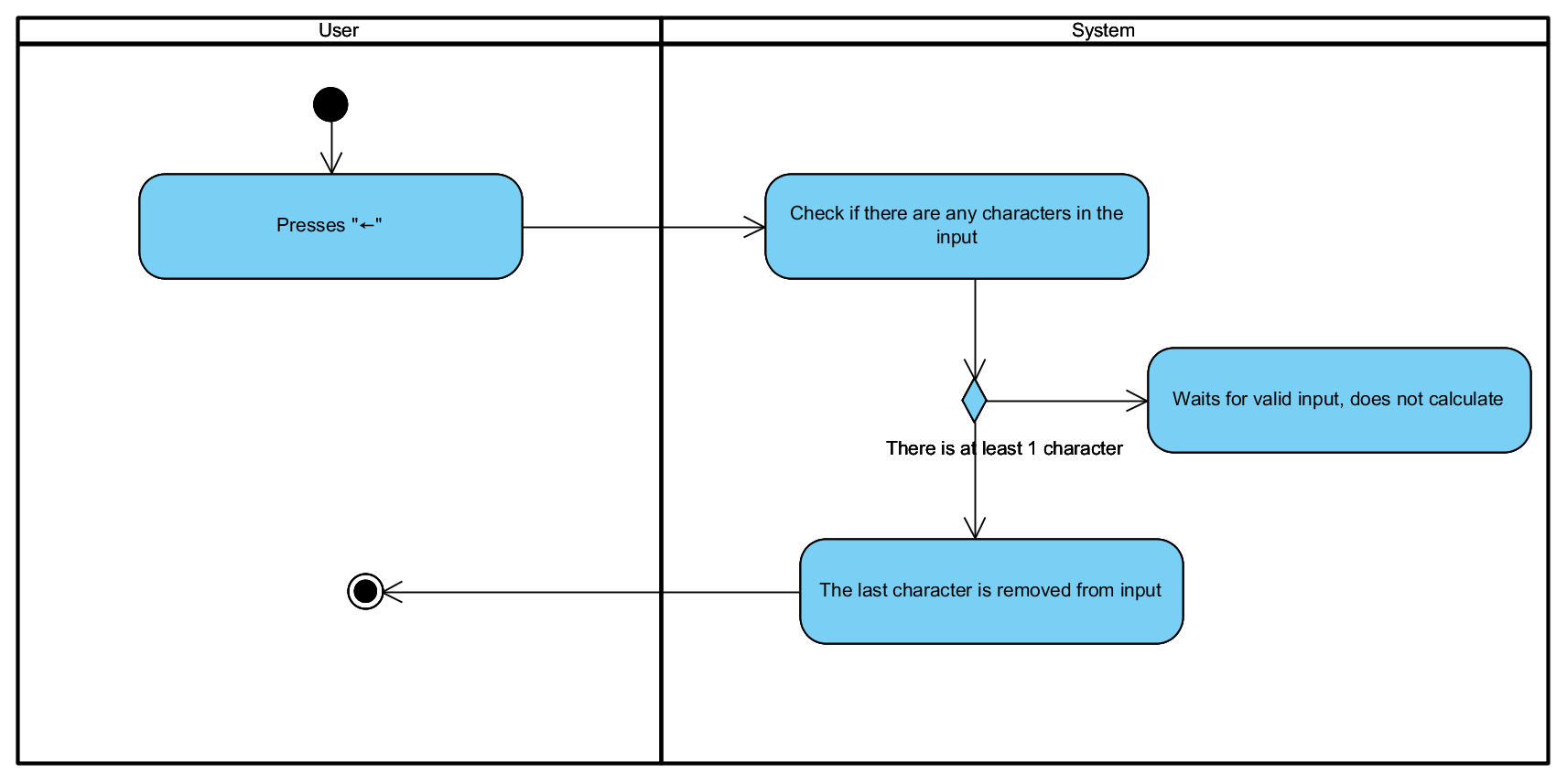
#### 6. Square root:



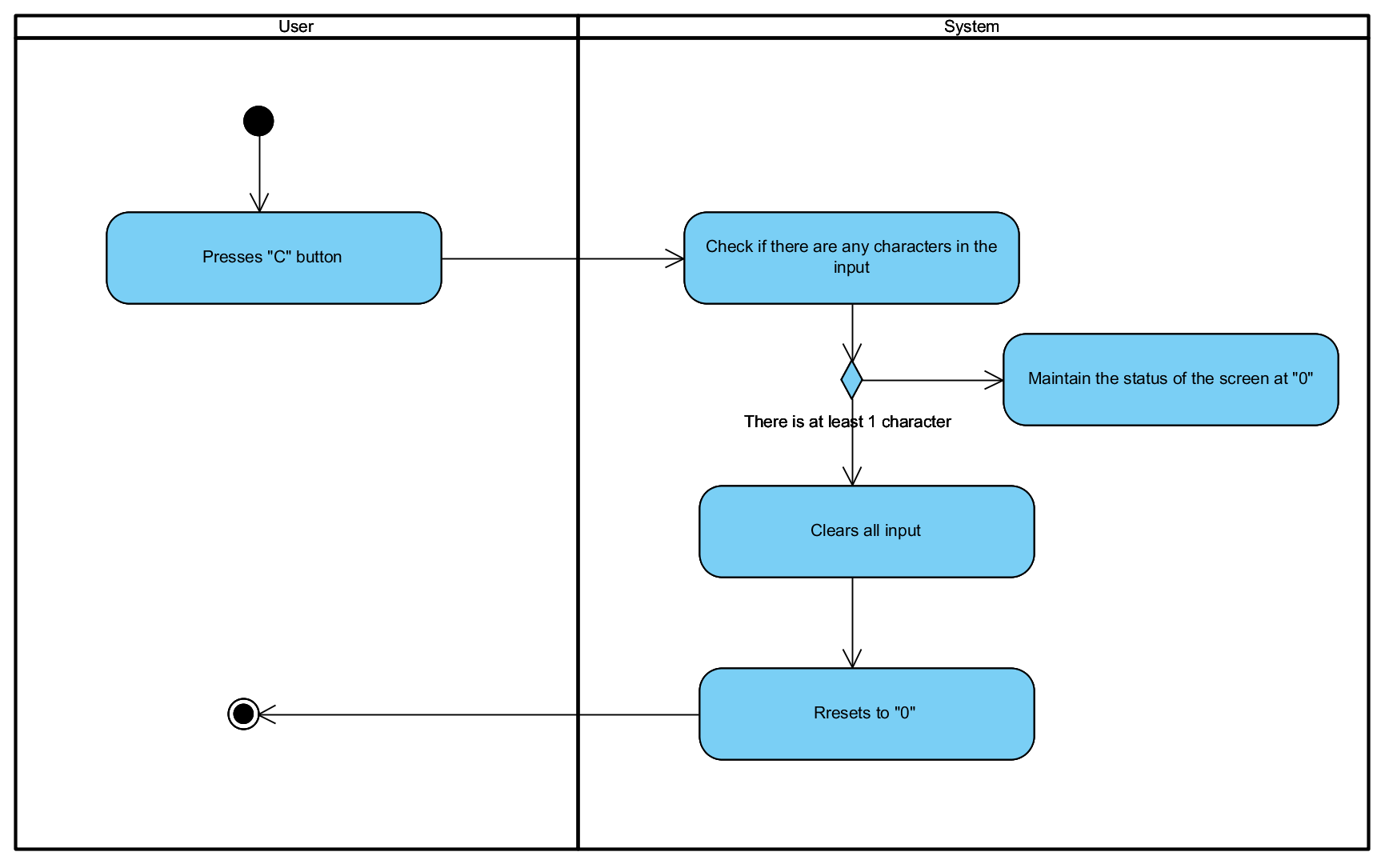
#### 7. Percentage Calculation:



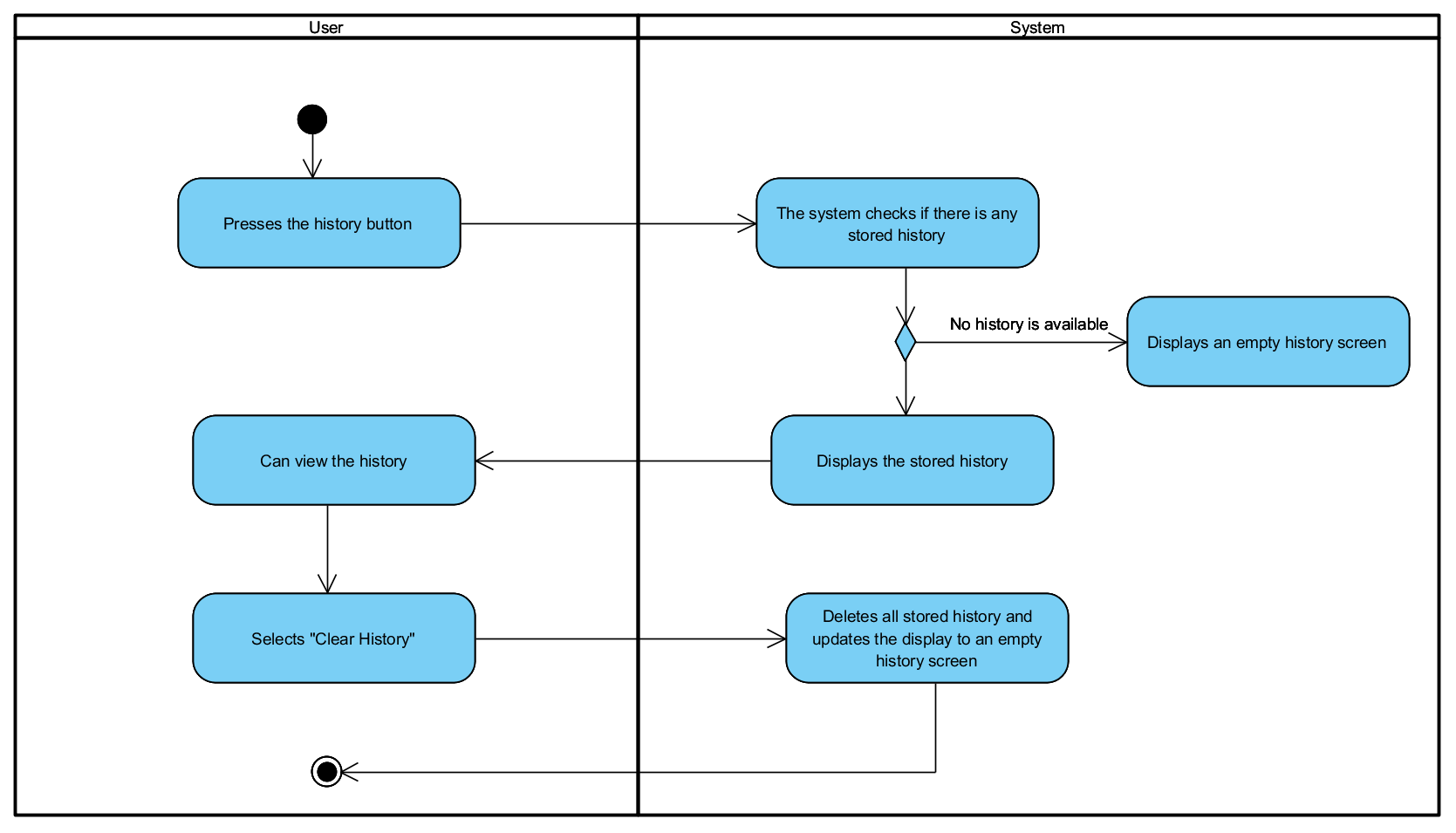
#### 8. Backspace:



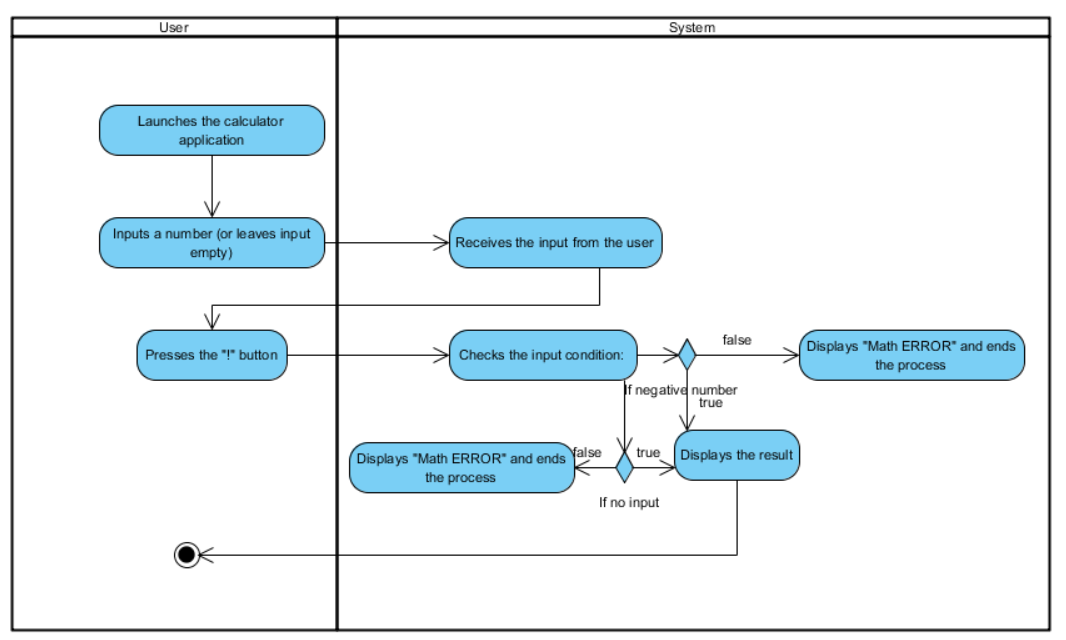
#### 9. Clear:



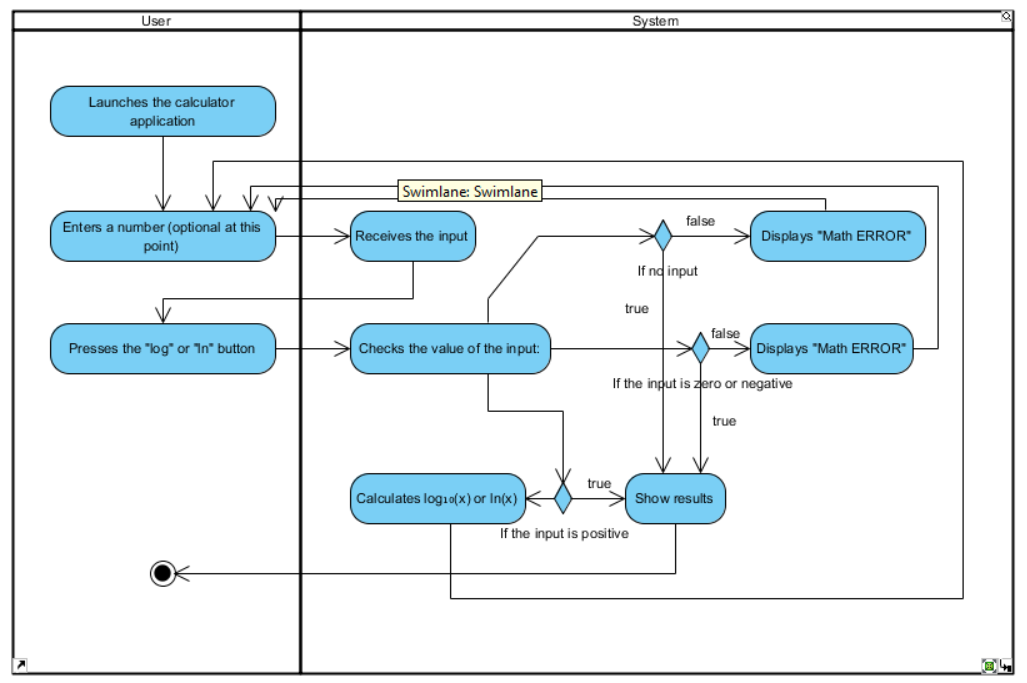
#### 10. Temporary History Storage



**11. Factorial**

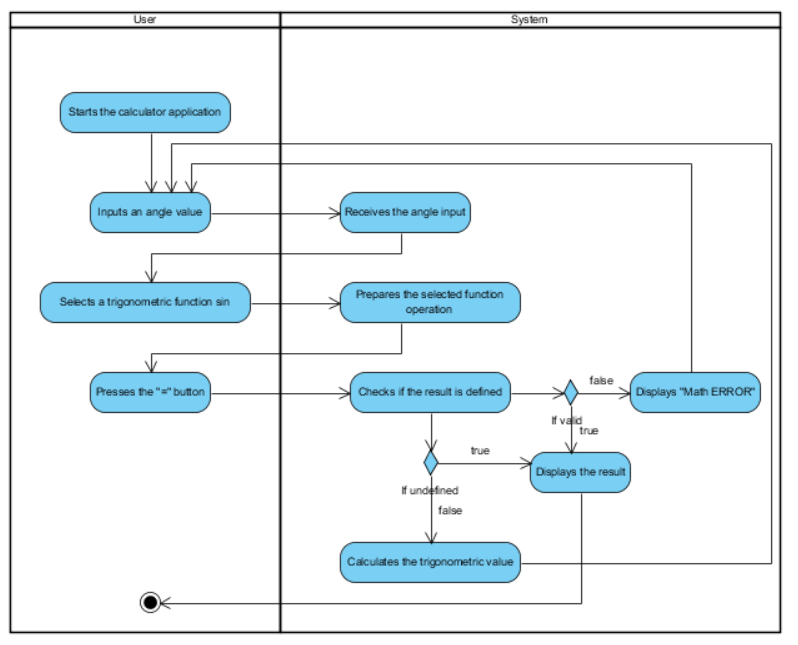


**12.** **Logarithms**

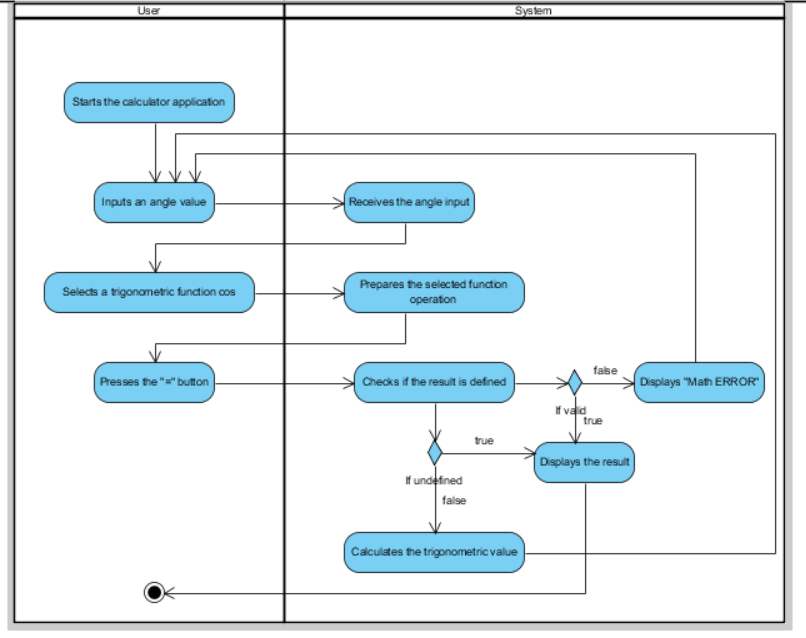


**13. Calculate Trigonometric**

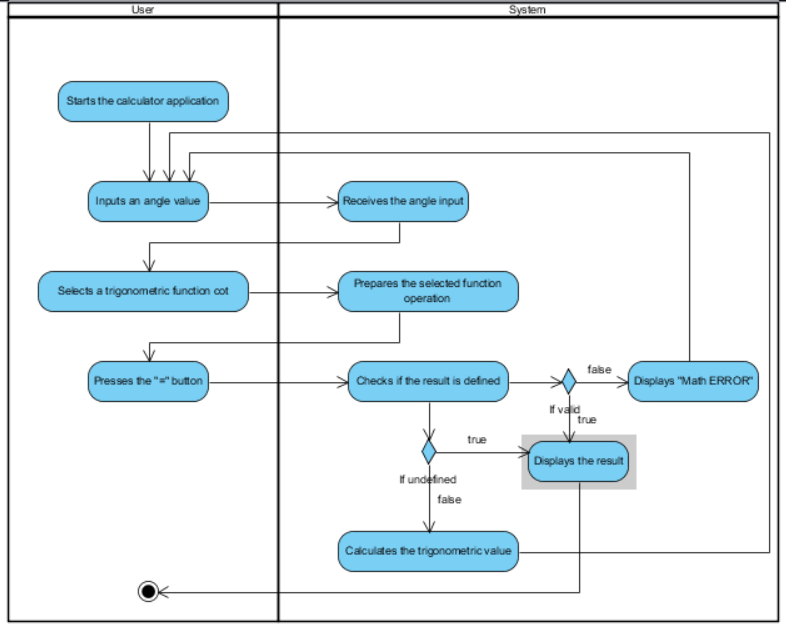
**13.1 sin(x)**



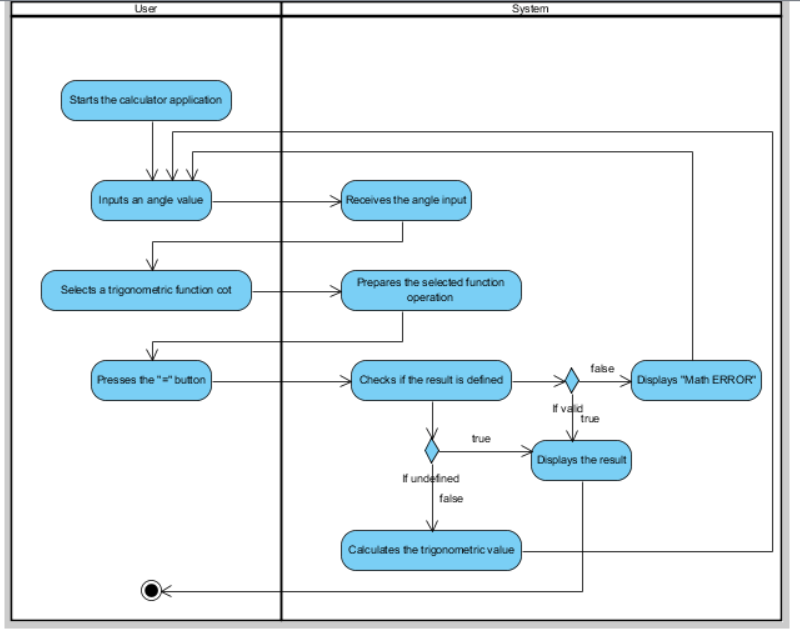
**13.2 cos(x)**



**13.3 tan(x)**



**13.4 cot(x)**



### 14. Convert Degrees ↔ Radians

### 

### 15. Evaluate Complex Expressions

### 

### 16. Calculator History Management

### 16.1 Save History to File

### 

### 16.2 Search History

### 

### 16.3 Delete History Entry

### 

### 17. UI Customization

### 17.1 Dark/Light Mode

### 

### 17.2 Change font Style

### 

### 17.3 Change UI Color

### 

### 17.4 Enable/Use Keyboard Shortcuts

### 

### 18. Advanced Controls

### 18.1 Backspace

### 

### 18.2 Delete Character to the Left of Cursor

### 

### 18.3 Delete Character to the Right of Cursor

### 

### 18.4 Clear Entry

### 

### 19. Result Interaction

### 