Andreea Calculator Project

Software Requirements Specification

Version 1.1

15/03/2023

Revision History

Date	Description	Author	Comments
10/03/2023	Version 1	Pantaru Andreea	Initial Draft
15/03/2023	Version 1.1	Pantaru Andreea	Added section

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Printed Name	Title	Date
Panatru Codrut	Lead Software Engg.	09/03/2023
Pantaru Hailey	Lead QA Engineer	09/03/2023
Trufanda Anamaria	Project Sponsor	09/03/2023

1. Introduction

The Andreea Calculator project aims to provide a simple calculator for processing operations.

1.1 Purpose

The Purpose of this document is to outline the requirements for the Calculator project.

This document will be used by all stakeholders including developers and testers.

1.2 Scope

The scope of this project is limited to the testing of the features described in the succeeding sections of this document.
Non-functional testing like stress,performance is beyond scope of this project.
Automation testing is beyond scope.
Functional testing & external interfaces are in scope and need to be tested.
The calculator site will be only compatible with Chrome version 27 and above.

2. Specific Requirements

Andreea Calculator Site has following modules

- Input Module
- Numbers Module
- Operands Module
- Results Module

Description of the modules

BR#	Module Name	Description
B1	Input Module	The client can introduce the input.
B2	Numbers Module	The client can write numbers from 0-9
B3	Operands Module	The client can use mathematical signs.
B4	Results Module	The client can do mathematical operations and have displayed the results in input.

2.1 External Interface Requirements

2.1.1 User Interfaces

None

2.1.2 Hardware Interfaces

None

2.1.3 Software Interfaces

None

2.1.4 Communications Interfaces

None

3.1 Front End Details

This section describes the Front end of the Calculator.

Following is a list of module wise fields.

Operators Buttons:

- Clear
- Divide
- Multiply
- Add
- Substract
- Decimal
- Equal

Number Buttons:

- Zero
- One
- Two
- Three
- Four
- Five
- Six
- Seven
- Eight
- Nine
- Ten

3.2 Technical Requirements

T1 My calculator should contain a clickable element containing an = (equal sign) with a corresponding id="equals".

T2 My calculator should contain 10 clickable elements containing one number each from 0-9, with the following corresponding IDs: id="zero", id="one", id="two", id="three", id="four", id="five", id="six", id="seven", id="eight", and id="nine".

T3 My calculator should contain 4 clickable elements each containing one of the 4 primary mathematical operators with the following corresponding IDs: id="add", id="subtract", id="multiply", id="divide".

T4 My calculator should contain a clickable element containing a . (decimal point) symbol with a corresponding id="decimal".

T5 My calculator should contain a clickable element with an id="clear".

T6 My calculator should contain an element to display values with a corresponding id="display".

T7 At any time, pressing the clear button clears the input and output values, and returns the calculator to its initialized state; 0 should be shown in the element with the id of display.

T8 As I input numbers, I should be able to see my input in the element with the id of display.

T9 In any order, I should be able to add, subtract, multiply and divide a chain of numbers of any length, and when I hit =, the correct result should be shown in the element with the id of display.

T10 When inputting numbers, my calculator should not allow a number to begin with multiple zeros.

T11 When the decimal element is clicked, a . should append to the currently displayed value; two . in one number should not be accepted.

T12 I should be able to perform any operation (+, -, *, /) on numbers containing decimal points

T13 If 2 or more operators are entered consecutively, the operation performed should be the last operator entered (excluding the negative (-) sign). For example, if 5 + * 7 = is entered, the result should be 35 (i.e. 5 * 7); if 5 * - 5 = is entered, the result should be -25 (i.e. 5 * (-5)).

T14 Pressing an operator immediately following = should start a new calculation that operates on the result of the previous evaluation.

T15 My calculator should have several decimal places of precision when it comes to rounding (note that there is no exact standard, but you should be able to handle calculations like 2 / 7 with reasonable precision to at least 4 decimal places).

3.3 Functional validations

1.The results must respect the mathematical rules.

3.4 Classes / Objects

3.5.1.1 Attributes
3.5.1.2 Functions

3.5 Non-Functional Requirements

1.The operation should be displayed in less than 1 second.

3.6 Inverse Requirements

Nil.

3.7 Design Constraints

Many of the Calculator users may not have adequate computer knowledge to use the site.
Hence, System must be intuitive and easy to understand.

3.8 Logical Database Requirements

None

3.9 Design Requirements

DR 1 - The site's background color is #808080.
DR 2 - The container has a height of 100vh, flex displayed, aligned to center.
DR 3 - The class calculator has a height of 445px, max-width of 460px, a border radius of 4 px.
DR 4 - The class calculator has background-color of turquoise, and a border of 5px solid #20b2aa.
DR 5 - The class output has a height of 80px.
DR 6 - The class keys has a gap of 5 px, a padding of 10px, grid displayed with 4 1fr columns.
DR 7 - The buttons have a font-size of 28px and a border-radius of 40px.
DR 8 - The pseudoselector "keys button:nth-child(15)" has a black color, a grid-row of 4/6, a grid-column of 4/4 and a background of linear-gradient(turquoise, #20b2aa, pink, #20b2aa, turquoise);
DR 9 - The pseudoselector "keys button:nth-child(16)" has a grid-column-start of 1 ad a grid-column-end of 3.
DR 10 - The classes "input" and "result" have a black color, height of 40px, are block displayed, with a front-size of 40px, right text-align and a 10px padding-right.

4. Analysis Models

None

5. Change Management Process

Changes to the SRS either from the development, testing team or the client side will be communicated to the project sponsor.

Any change made to the SRS will require a sign off from the Development lead , QA lead and the client.