MySimpleCalculator:

Software Requirements Specifications

Samantha Hipple

CS 504: Software Engineering

MSCS - City University of Seattle

September 3, 2021

hipplesamantha@cityuniversity.edu

**Abstract**

Software Requirements Specifications document for MySimpleCalculator, a Windows form program designed and developed by Samantha Hipple as a sample project for a software engineering course. This document provides an overview of the overall requirements and expectations of the MySimpleCalculator application while also examining the purpose of the different subsections throughout a Software Requirements Specifications document. MySimpleCalculator is advertised to be a basic add-on application that can perform the most basic of math operations that eLearning developers can offer as a student tool during computer-based examinations. It can perform one basic operation (addition, subtraction, multiplication or division) between two user input values before requiring a reset or clear to ensure proper function.

**Keywords:** Software Requirements Specifications (SRS), Project Scope, System Requirements (functional & non-functional), User Interface, Assumptions and Dependencies, Intended Audience & Development Roles

**1.** **INTRODUCTION**

This document provides an in-depth review of the Software Requirements Specifications (SRS) of MySimpleCalculator.

**Purpose**

The goal of MySimpleCalculator is to be an easy and efficient add-on application designed to assist students during a computer-based examination with the simplest of calculations.

The objectives that MySimpleCalculator aims to accomplish are: (1) improve student time & accuracy during computer-based examinations compared to exams taken that do not allow any calculator use and (2) decrease instances of cheating via user programmable / internet accessible calculators in exams that want to allow calculator usage by providing a built-in alternative to student provided calculators.

**Document Conventions**

Within this document, one will find five main sections covering the requirement specifications of MySimpleCalculator, with a number of subsections labeled by the left-aligned, boldfaced subheadings.

Additionally, in order to allow this paper to exist as a future learning tool as well as a requirements document, it has been designed to include information concerning the different subsections throughout to better explain their purpose within such a document – especially those that did not apply to this particular sample project.

**Intended Audience & Development Roles**

This document is intended for the developers, project managers, testers, and consumers of MySimpleCalculator.

This document is also an example document, created to fulfill a university assignment requirement, and therefore peers and instructors are also a part of the intended audience.

Samantha Hipple fulfills all roles for this project, from project manager to tester.

**Project Scope**

*Software Goals:*

* Window’s application designed to add, subtract, multiply or divide between two user-input values.

*Business Goals:*

* Prevent cheating on student examinations.
* Provide a calculator to assist with basic calculations during examinations.

MySimpleCalculator would allow users to divide, add, subtract or multiply between two values at a time - providing students with assistance in completing or ch4ecking their solutions during examinations, without allowing the calculator to do all of the work. Additionally, providing this tool as the standard for computer-based examinations, would help prevent cheating compared to student-owned calculator usage.

**References**

*A list of other documents that the SRS document refers to including sources such as websites or written literature.*

*Where was this format/list of SRS outline found?*

**2. GENERAL DESCRIPTION**

An application that can be used as an add-on to computer-based examinations that performs the most basic of calculations.

**Product Perspective**

This product was created at a whim and its purpose decided at a later date. The true purpose of this product is simply to fulfill a sample role within an example project created for a university course assignment. The feigned purpose is to be an add-on tool that eLearning developers can use to improve their exams’ integrity.

**Product Features**

MySimpleCalculator will be able to perform one operation before needing to be reset, between two user-input values.

The operations available are:

* Addition
* Subtraction
* Multiplication
* Division

Values can be integers or decimals, positive or negative. Users can choose between backspacing [<] during input, clearing their current entry [CE], or clearing all (resetting) [C] as they enter their calculations.

A label above the input screen records the initial user operation once the user has chosen the operator for their calculation. This is reset via Clear All [C].

**User Class and Characteristics**

*eLearning developers* would use this application as an add-on to the virtual examinations they produce.

*Teachers/Proctors* would need to understand how the application functions in order to aid student use during examinations.

*Students* would need to find this tool intuitive and simple to utilize during an exam.

**Operating Environment**

MySimpleCalculator is a Window’s Form Application and therefore requires the user to be using a Window’s OS. Specifically, this application was designed and tested on a Windows 10 Home Edition, 2004 Version operating system.

MySimpleCalculator was designed and tested on a device with the following specifications:

Text

Description automatically generated



No attempts have been made to discover the application’s minimum device specification requirements.

**Design & Implementation Constraints**

MySimpleCalculator is the product of a brand-new computer science student using YouTube and their past experience from one entry-level C# university course. Therefore, there are a number of design flaws and preferences for how this application functions that have not yet been included in the initial release due to a lack of programming knowledge/experience.

**Assumptions and Dependencies**

*Assumptions. . .*

The two framework dependencies listed in MySimpleCalculator are:

1. Microsoft.NETCore.App.Ref\3.1.0
2. Microsoft.WindowsDesktop.App.Ref\3.1.0

**3. SYSTEM REQUIREMENTS**

This section provides a detailed look at the different functions of MySimpleCalculator that are available for users and how they are expected to work.

Users must interact directly with the application using mouse clicks/cursor selects on the buttons of the GUI to make the calculator function, it does not currently recognize keyboard input.

**Numeric Buttons [0] – [9] & Decimal [.]**

* The text on each button reflects user input and will appear above the buttons in the textbox used to create the input screen once a button has been selected.
* The decimal point can only be applied once per entry/solution.
* The default zero that appears in the input screen upon loading/resetting the application, will be replaced upon user input.
* Once the first half of an operation has been entered by the user (value + operator), the text within the input screen will move up to the label, recording the first half of every equation, and reset for the second value input.

**Basic Operators [+] [-] [\*] [/]**

* The operator buttons are labeled to reflect their input values upon selection.
* Upon selection, current input screen value and the operator chosen will appear in the label above the input screen.
* Choosing another operator prior to inputting the next value will change the operator in the label and calculation.
* Initializing the second value of the operation will erase the primary value from the input screen (now recorded in the label) and allow the input screen to record the second input value for the operation.

**Execute [=]**

* The operation performed will equal the string displayed in the label above the input screen combined with the current input screen value.
* The results of the operation will be displayed in the input screen’s textbox.

**Backspace [<]**

* Deletes the most recent input value from the input window textbox, allowing users to edit their entries.
* Removes the chosen operator to allow the user to continue editing their initial value entry when used after the operator has been chosen, but before the secondary value has been inputted.

Future versions should focus on removing this ability from calculated result strings that are displayed in the input window textbox.

**Clear Entry [CE]**

* Clears only the input/logic that is currently displayed in the input window textbox.
* When clearing the secondary value input, the initial value should remain recorded in the label and as the first portion of the calculation.

**Clear All [C]**

* Resets application to default.
* At this time, users must manually reset the application using the [C] button after each calculation

Future versions should aim to automatically clear the application between completed calculations if not designed to be able to use the results as the new initial value of the next desired operation.

**4. EXTERNAL INTERFACE REQUIREMENTS**

**User Interfaces**

A picture containing text, electronics

Description automatically generated

**Hardware Interfaces**

**Communications Interfaces**

**Software Interfaces**

**5. NON-FUNCTIONAL REQUIREMENTS**

**Performance Requirements**

**Safety Requirements**

**Security Requirements**

**Software Quality Attributes**

**Other Requirements**

**6. CONCLUSIONS**

WHY SRS IS NECESSARY/USEFUL

**7.** **REFERENCES**

http://users.encs.concordia.ca/~eshihab/teaching/slides/srs\_template\_sep14.pdf