Software Requirements Specification

for

Scientific Calculator

Prepared by Joe Corona

Central Washington University

November 9, 2021

Table of Contents

Contents

1. Introduction

1.1 Purpose

This will help you better understand the purpose of the scientific calculator if you haven't used one in high school. Scientific calculators are built and programmed to collect the input of the user and then providing an output no matter the complexity of that mathematical problem. This Scientific Calculator was created with the Advanced Software Engineering class in mind; class number 480. The goal of that project was to prove to our peers and especially to ourselves that we really do have the skills to solve complex problems, and send them out before the due date. Parameters and goals were achieved so now to look at who the targeted audience is, the users have to first open the SciCalc.py file out of the Source_Code_Corona_Joe.zip file. In that same file a HowTo_Joe_Corona file is also included to read and get you prepared to use the calculator.

1.2 Intended Audience and Reading Suggestions

This project is a document for the Scientific Calculator that was turned in as Lab 3 to show the user that it is an easy-to-use software application for calculation purposes. This was under the guidance of college professors.

1.3 Project Scope

A software that is easy-to-use still has many purposes but the main one is that it hopes to reach the users satisfaction goals. Above all, the hopes are that the users will feel comfortable enough to keep coming back in order to get their mathematics calculations in order.

1.4 References

<The references are:

- ✓ *Math library Python*
- ✓ *Python Tkinter Tutorial - GeeksforGeeks*

2. System Features

Description and priority of the scientific calculator are that of having the functions organized properly in order to have the users feel almost as if it is in their second nature. Of course, this project is at the highest of priorities because it is difficult to make all of the calculations by hand in an efficient manner.

2.1 Functional Requirements

2.1.1 Functional Requirements

The purpose of the requirements part is that it discusses the functionalities that are needed in the system. For instance, we are required to make software user friendly and make sure that the learning process isn't arduous. If the user manages to get used to the software then the job has been done.

2.2 Non-functional Requirements

2.2.1 Non-functional Requirements

The purpose of the Non-functional requirements and or necessities are to accommodate the software which won't be expressed as functions. The maintainability of the software, and usability of the software.

1. *Accuracy of results*
   Well if there is a large equation that the user will want to include into the calculator then that is an issue that might arise. Most of the answers should output the precise outcome since the math library was used.
2. *Human-computer interface issues*
   The issues that could potentially be at hand are if the user is opening the software and running it is that they might not have a machine that is capable of running

3. Analysis and Checks

3.1 Feasibility Analysis

The way in which programmers/software developers test their product is by making a software feasibility test while executing the program. This will save time and money the two most important things when it comes to software. When development began we started doing these feasibility *analysis as we were producing the software in order to not make the mistakes worse down the line, therefore saving time and money.*

Some people have some things to say about the feasibility analysis being that it is a government regulation and so forth. Yes, the reason behind this analysis is something that can be too arduous to read but it is one of the most important pieces of an SRS document.

3.2 Consistency Check

The consistency check that was done for the scientific calculator was fairly easy going as it didn't cause many issues while executing the software because the functionalities are stable. All of the buttons work the way that they are supposed to in order to create the calculator. Ultimately, everything is consistent and works well.

3.3 Validity Check

When running the validity check we were worried that they might not run on all of the machines but figured that once the user opens it the program will execute and they will access the scientific calculator. The data while creating this project was all correct because it was thanks to the math library that python provides for us.