System Requirements Document

Prepared by:

Aeris Bolen

Rebecca Kenny

Robert Strickland

Preface

i. This document is intended for the development team, stakeholders such as the professor, and potential users. This requirements document is the first version. It was created to be reviewed by stakeholders as well as before development begins.

Introduction

i. This software is intended to provide quick mathematical calculations to assist students. The software will receive numerical inputs and perform various operations. It is not meant to replace more complex calculators but can provide quick examples to help a student check their work and ensure their competency and understanding of math concepts.

Glossary

i. In the system requirements document, various terms will be used to address the program intentions, requirements, system models, and specifications.
    a. *Average:* the mean, which is calculated by dividing the sum of the values in the set by their number.
    b. *Sum:* The total of the numbers when added together.
    c. *Product:* The result of multiplying 2 or more numbers.
    d. *Input:* What is entered by a user.
    e. *Output:* The information produced by a system based off of the input.

User Requirements Definition

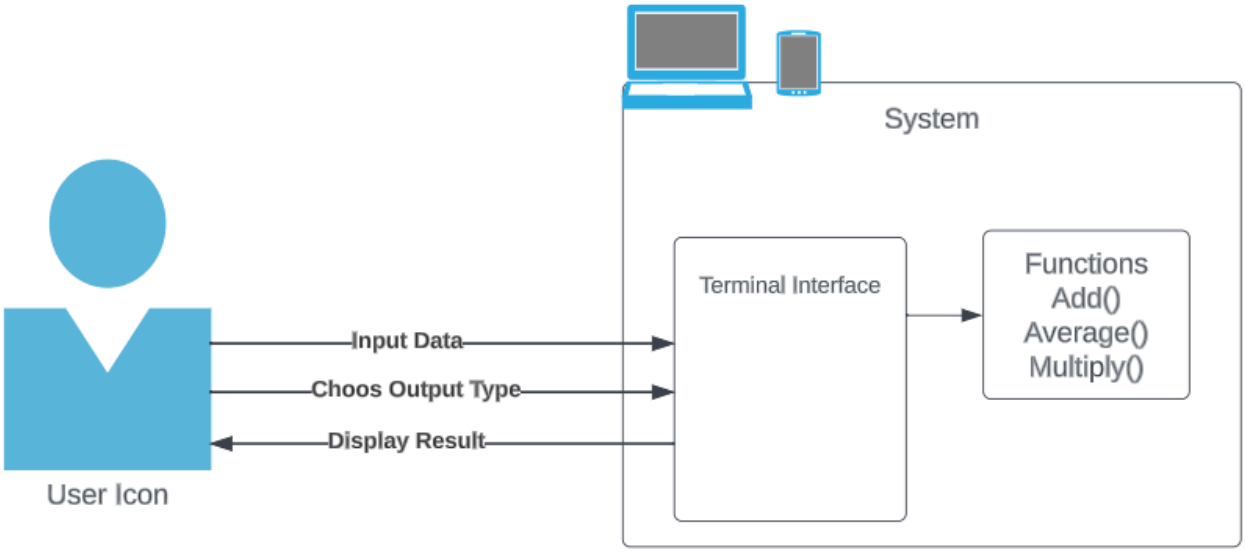
i. The requirements for this project would be to make a program that can receive three individual values and output three individual values.
ii. User Requirement Statement:
    a. The [user] shall [use the software] in order to [perform calculations]. iii.

The user requirements of the project are:

a. The user's machine must possess Python v3 or later
b. Users will need to input three numbers; 1-10, 20-30, 70-100.
c. Inputs must conform to the prompt's requests.
d. The user shall be allowed to select which output to produce with the provided numbers.

System Architecture

i.



System Requirements Specification

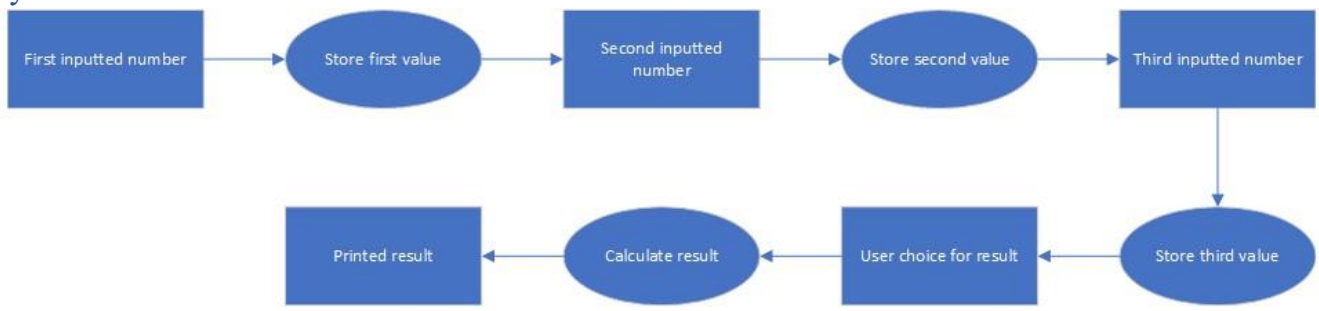
I. The system requirements specification involves the functional aspects of the program.
II. The software requirements of the program are:
    a. The requirements will be to:
        i. Input a number from 1-10.
        ii. Input a number from 20-30.
        iii. Input a number from 70-100.
    b. The project is required to produce three outputs:
        i. The average (mean) of the numbers.
        ii. The sum of the numbers.
        iii. The product of the numbers.
    c. The program will be implemented in Python.
    d. The nonfunctional requirements for the program are based on the attributes of good software:
        i. Maintainability
            1. The program needs to reliably calculate the provided numbers.
            2. It shall also be subject to change to better fit the needs of users for the program.
                a. This can be accomplished by expanding the numbers it can calculate and input.
        ii. Efficiency
            1. The program will be efficient by using low resources while providing a timely output.
            2. It shall be within a reasonable size.
        iii. Acceptability

1. The program must be easy to understand
2. The interface must be intuitive
3. The calculator must provide the correct results
4. The program must be compatible with all systems as long as they run the version of Python preveiously stated.

iv. Dependability and Security

1. Due to the program simply focusing on calculating numbers, if it malfunctions, it shall not cause damage to the user. It must remain accessible.

System Models



System Evolution

i. Potential future changes may include adding additional functionality to the program. This could include changing how many numbers a user will input and what results they can choose from. There may also be changes made to the format and range of numbers to be inputted. These modifications would arrive from users based on their experience interaction with the program. Any alteration would not require hardware changes and would only be implemented within the code base.

Index

i. Preface – Context for the creation of the software
ii. Introduction – The goal of the software
iii. Glossary – various terms and vocabulary used in document
iv. User Requirements Definition – Hardware and software required by the user for the program to run. General reasoning for why the user would want to use the software.
v. System Architecture – Visual diagram of system
vi. System Requirements Specification – The requirements that are to be delivered by the development team.
vii. System Evolution – Possible future modifications to the software viii. Appendices – Additional diagrams and information
ix. Index – A series of explanations for the various portions of the Systems Requirements Document