**Software Engineering**

**Software Requirements Specifications**

**IP02: MySimpleCalculator**

**August 08, 2021**

**Version 1**

**Samantha Hipple**

**City University of Seattle**

**MSCS Program**

|  |
| --- |
| **Revisions** |

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1 | Samantha Hipple | Initial SRS for MySimpleCalculator | 08/25/2021 |

|  |
| --- |
| **Review & Approval** |

Requirements Document Approval History

| Approving Party | Version Approved | Signature | Date |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Requirements Document Review History

| Reviewer | Version Reviewed | Signature | Date |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| **Table of Contents** |

[1. Introduction 3](#_Toc244519333)

1.1 Purpose 3

1.2 Document Conventions…………………………………………………………………….3

1.3 Intended Audience…………………………………………………………………………3

1.4 Scope……………………………………………………………………………………….3

1.5 References………………………………………………………………………………….3

[2. General Description 3](#_Toc244519334)

2.1 Product Perspective………………………………………………………………………...3

2.2 Product Features 3

2.3 User Class Characteristics 3

2.4 Operating Environment 3

2.5 Constraints 3

2.6 Assumptions and Dependencies 3

[3. System Requirements 3](#_Toc244519335)

[4. External Interface Requirements 4](#_Toc244519336)

[4.1 User Interfaces 4](#_Toc244519337)

[4.2 Hardware Interfaces 4](#_Toc244519338)

[4.3 Communications Interfaces 4](#_Toc244519339)

[4.4 Software Interfaces 4](#_Toc244519340)

[5. Non Functional Requirements 4](#_Toc244519341)

1. Introduction

* 1. **Purpose:** The goal of your project and the objectives it aims to accomplish

*MySimpleCalculator is a Window’s application designed to be used during examinations for single-step calculation assistance in order to prevent the use of user-programmable/Wi-Fi-capable calculators that exist today.*

* 1. **Document conventions:** The typographical methodologies followed within the document. For e.g. any abbreviations, typographical stylization of content or change of fonts and its significance.

*Author input is typed in italics below the various section descriptions of this document.*

* 1. **Intended audience:** Describe which part of the SRS document is intended for which reader. Include a list of all stakeholders of the project, developers, project managers, and testers for better clarity.

*Developers: swiseWHAT*

*Project Managers: Hipples*

*Testers: Hipples & swiseWHAT*

* 1. **Scope:** Specify how the software goals align with the overall business goals and outline the benefits of the project to business.

*Software Goals:*

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

*Business Goals:*

*x. Prevent cheating on student examinations.*

*x. Provide a calculator to assist with simple math problems during examinations.*

*MySimpleCalculator would allow users to divide, add, subtract or multiply between two values at a time. This provides students assistance in completing/checking their math solutions during examinations, without allowing the calculator to do all of the work. Additionally, providing this tool as the standard for your examinations, can help prevent cheating via the use of the user-programmable and internet-capable calculators that exist today.*

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

## 2. General Description

**2.1 Product perspective:** Describe the context and origin of the product

*MySimpleCalculator is an extremely basic Window’s application designed to compute simple calculations between two user input values.*

**2.2 Product features:** A high level summary of the functions the software would perform and the features to be included.

* *Add, subtract, multiply, divide 2 user input values*
* *Decimals enabled*
* *Label to record first portion (value + operator) of the calculation*
* *Backspace*
* *Clear Entry*
* *Clear All*
  1. **User class and characteristics:** A categorization and profiling of the users the software is intended for and their classification into different user classes

*eLearning Developers, Teachers, Proctors & Students*

* 1. **Operating environment:** Specification of the environment the software is being designed to operate in.

*Microsoft Window’s OS*

* 1. **Constraints:** Any limiting factors that would pose challenge to the development of the software. These include both design as well as implementation constraints.

*Project manager is a beginner level programmer & designer with no experience in implementation or distribution. I mean, do you need Microsoft studio to be able to run my program? I have no idea how that works.*

* 1. **Assumptions and dependencies:** A list of all assumptions that you have made regarding the software product and the environment along with any external dependencies which may affect the project.

*Creating a “Window’s Form” application in Microsoft Visual Studio should be all one needs to do for the application to work on any Window’s OS.*

*Honestly, that probably isn’t true, but that’s the assumption at this moment in time.*

*If not, then whoever uses this application will need to download Microsoft Studio.*

*Dependencies:*

* *C:\Program Files\dotnet\packs\Microsoft.NETCore.App.Ref\3.1.0\*
* *C:\Program Files\dotnet\packs\Microsoft.WindowsDesktop.App.Ref\3.1.0\*

## 3. System Requirements

**3.1 Functional requirements:** All the requirements within the system or sub-system in order to determine the output that the software is expected to give in relation to the given input. These consist of the design requirements, graphics requirements, operating system requirements and constraints if any.

## 4.External Interface Requirements

4.1 User Interfaces

The logic behind the interactions between the users and the software. This includes the sample screen layout, buttons and functions that would appear on every screen, messages to be displayed on each screen and the style guides to be used.

4.2 Hardware Interfaces

All the hardware-software interactions with the list of supported devices on which the software is intended to run on, the network requirements along with the list of communication protocols to be used.

4.3 Communications Interfaces

Determination of all the communication standards to be utilized by the software as a part of the project

4.4 Software Interfaces

The interaction of the software to be developed with other software components such as frontend and the backend framework to the used, the database management system and libraries describing the need and the purpose behind each of them.

## 5. Non-Functional Requirements

**5.1 Performance requirements**

The performance requirements need to be specified for every functional requirement. The rationale behind it also needs to be elaborated upon.

**5.2 Safety requirements**

List out any safeguards that need to be incorporated as a measure against any possible harm the use of the software application may cause.

**5.3 Security requirements**

Privacy and data protection regulations that need to be adhered to while designing of the product

**5.4 Software quality attributes**

Detailing on the additional qualities that need to be incorporated within the software like maintainability, adaptability, flexibility, usability, reliability, portability etc.

**5.5 Other requirements**

These may include the legal requirements, resource utilizations, future updates etc.