

---

**EECS 348 Group 10**

---

**Web Calculator for University of Kansas Students  
Software Requirements Specifications**

**Version 2.0**

Web Calculator for University of Kansas Students	Version: 2.0
Software Requirements Specifications	Date: 12/03/2023

## Revision History

Date	Version	Description	Author
11/10/23	1.0	Initial version	Harlan Williams
12/03/23	2.0	Updated version	Nora Manolescu

Web Calculator for University of Kansas Students	Version: 2.0
Software Requirements Specifications	Date: 12/03/2023

## Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms, and Abbreviations	4
1.4	References	4
1.5	Overview	4
2.	Overall Description	5
2.1	Product perspective	5
2.1.1	System Interfaces	5
2.1.2	User Interfaces	5
2.1.3	Hardware Interfaces	5
2.1.4	Communication Interfaces	5
2.1.5	Memory Constraints	5
2.1.6	Operations	5
2.2	Product functions	5
2.3	User characteristics	5
2.4	Constraints	5
2.5	Assumptions and dependencies	5
3.	Specific Requirements	6
3.1	Functionality	6
3.1.1	Allow users to input expressions.	6
3.1.2	Evaluate inputted user expressions.	6
3.1.3	Display results of evaluation.	6
3.2	Use-Case Specifications	7
3.3	Supplementary Requirements	7
3.3.1	Handle Errors and invalid expressions	7
3.3.2	Project finished by 12/5/2023	7
3.3.3	Project to be implemented in C++	7
3.3.4	Project includes a README	7
3.3.5	Project must be open-source	7
3.3.6	Project must be able to be hosted as a droplet on DigitalOcean	7
4.	Classification of Functional Requirements	7

Web Calculator for University of Kansas Students	Version: 2.0
Software Requirements Specifications	Date: 12/03/2023

# Software Requirements Specifications

## 1. Introduction

### 1.1 Purpose

This document describes the functional and non-functional requirements of the calculator project, as well as the constraints under which the project resides. This document intends to describe all functionality provided by the project that will be visible to the end-user, and will be referenced during the design stage of the project in order to guide the development team's decisions.

### 1.2 Scope

This document describes requirements both for the calculator subsystem and the web application. It contains a description of: functional requirements important to the core functionality of the application, non-functional requirements which improve the usability of the application for the end-user, and constraints which are set forward both by the project requirements of EECS 348, as well as the particular need for the project to run on Linux and be hosted as a web application on DigitalOcean.

### 1.3 Definitions, Acronyms, and Abbreviations

This document uses the same definitions, acronyms, and abbreviations described in the Project Plan.

### 1.4 References

This document references the Project Plan for definitions, acronyms, and abbreviations, as well as the use-case diagram at use-cases.pdf in the project documentation folder.

### 1.5 Overview

The document first presents a high-level description of all external factors that affect the project from a software and hardware view, the general requirements of users and the type of user that is expected to use the project, and what factors act as constraints on the project's design. Then a list of specific requirements is presented, along with a brief description of the requirement. These are both functional and non-functional requirements. The constraints are then also enumerated along with a brief description of where the constraint arises from.

Finally, the functional requirements are classified as either *Necessary*, *Desirable*, or *Optional*. Necessary functional requirements are non-negotiable and must be fulfilled for the project to function at all as a calculator. Desirable requirements are those which are secondary to the project's primary goal to implement a calculator. Optional requirements extend the basic functionality of the project, but can be removed from the project as well without affecting core functionality.

Web Calculator for University of Kansas Students	Version: 2.0
Software Requirements Specifications	Date: 12/03/2023

## **2. Overall Description**

### **2.1 Product perspective**

#### *2.1.1 System Interfaces*

The project must run in a Linux virtual machine environment, and as such must use POSIX system calls whenever accessing OS operations.

#### *2.1.2 User Interfaces*

The primary user interface will be through a web app that will provide a web page for evaluating expressions.

#### *2.1.3 Hardware Interfaces*

The project does not itself manage hardware interfaces such as sockets for communication.

#### *2.1.4 Communication Interfaces*

The web application will use the cpp-httplib library to handle outbound and inbound web connections. At a low-level, this will involve the use of OS sockets, and also will necessitate port forwarding to be hosted on the web.

#### *2.1.5 Memory Constraints*

The web application is constrained to use less than 512 MiB of memory to be able to be hosted on a basic-tier DigitalOcean droplet.

#### *2.1.6 Operations*

The application will take user requests through an HTTP interface and respond in a similar way. On the back-end, the project will process user input, evaluating an expression with the calculator that the project will also develop.

### **2.2 Product functions**

The application will provide an environment in which users can evaluate arithmetic expressions. To do this, it provides a calculator interface.

### **2.3 User characteristics**

The application is being developed with college students in mind, but will carry out functionalities useful towards anyone who has use for a basic calculator.

### **2.4 Constraints**

The project must be implemented in C++, and must be able to compile and run on a DigitalOcean Linux virtual machine environment. To achieve this, the project must use POSIX system calls whenever interacting with the operating system. The project must also be released under an open-source license to permit graders access to the source code, as well as to permit the project to be extended in the future.

The project is being developed during the Fall 2023 semester of EECS348 Software Engineering I, and thus must be in a completed and gradable state by the end of the semester: December 5th, 2023. Because the project is being developed for a class, it must contain a README and various other documentation about the development process to ease grading as well as to allow graders, and potential future users, to easily learn how to use the application or to contribute to the project under open-source.

### **2.5 Assumptions and dependencies**

The project is dependent upon the following open-source libraries: Nlohmann JSON for Modern C++, cpp-httplib, Inja, and sqlite3. The project also will depend on DigitalOcean for hosting the web app, and assumes that DigitalOcean will be able to meet that need.

Web Calculator for University of Kansas Students	Version: 2.0
Software Requirements Specifications	Date: 12/03/2023

### 3. Specific Requirements

#### 3.1 Functionality

##### 3.1.1 *Allow users to input expressions.*

The user will have access to a web interface where they can input in text an arithmetic expression, and then submit the expression to the application. Users will have access to the following operations:

##### **Addition**

The addition operation calculates the sum between integers. This operation should be implemented to follow the arithmetic rules of addition and should be able to correctly navigate negated numbers in a given sum.

##### **Subtraction**

The subtraction operation calculates the difference between integers. This operation should be implemented to follow the arithmetic rules of subtraction, outputting negative results when the proceeding term in the operation is greater than the previous term.

##### **Multiplication**

The multiplication operation calculates the product of integers. This operation should be implemented to follow the arithmetic rules of multiplication and the sign of the product must correspond correctly to the number of negations involved in the inputted expression.

##### **Division**

The division operation calculates the quotient of integers. This operation should be implemented to follow the arithmetic rules of division and the sign of the quotient must correspond correctly to the number of negations involved in the inputted expression. Further, because this is an integer calculator, the integer representation must be outputted for decimal and fractional possibilities that could happen with having this operation.

##### **Modulus**

The modulus operation calculates the modulus of integers. This operation should be implemented to follow the arithmetic rules of modulus, correctly navigating negations on either term involved in a modulus.

##### **Exponentiation**

The exponentiation operation calculates the exponent of integers. This operation should be implemented to follow the arithmetic rules of exponentiation and the sign of the output must correctly correspond with consideration of both the sign of the base and whether it is being taken to an even or odd power. Further, because this is an integer calculator, the integer representation must be outputted for decimal and fractional possibilities that could happen with inputs such as negative powers.

##### 3.1.2 *Evaluate inputted user expressions.*

The user will be able to submit their inputted expression to the application for it to be solved. Commands to evaluate, clear, or delete input on the calculator must thus be implemented for users to move forward after inputting their expressions. The application will take the user input and tokenize it to perform the operations in the correct order of precedence indicated by the user through the use of parentheses and PEMDAS.

##### 3.1.3 *Display results of evaluation.*

After calculating the result, the application will return it to the user. Otherwise, the calculator will also indicate if an error occurred, either due to the user inputting incorrectly formatted arithmetic, or a server-side error.

Web Calculator for University of Kansas Students	Version: 2.0
Software Requirements Specifications	Date: 12/03/2023

### 3.2 Use-Case Specifications

See the use-case diagram 'use-cases.pdf' in the project documentation folder.

### 3.3 Supplementary Requirements

#### 3.3.1 *Handle Errors and invalid expressions*

The calculator should be able to inform users when they enter an invalid expression and possibly help the user correct it.

#### 3.3.2 *Project finished by 12/5/2023*

Requirement set by EECS348.

#### 3.3.3 *Project to be implemented in C++*

Requirement set by EECS348.

#### 3.3.4 *Project includes a README*

Allows users to quickly learn how the project is set up and how to use it.

#### 3.3.5 *Project must be open-source*

Allows other contributors to extend the project after it is originally completed for EECS348

#### 3.3.6 *Project must be able to be hosted as a droplet on DigitalOcean*

Required if the application is to be available online.

## 4. Classification of Functional Requirements

Functionality	Type
Perform: Addition, Subtraction, Multiplication, Division, Exponentiation, and Modulo	Essential
Perform operations in PEMDAS order of precedence	Essential
Prompt user for input and return the evaluated expression, and reject invalid expressions	Essential
Project must be able to be hosted on a DigitalOcean droplet as a web application	Desirable
Handle errors/invalid expressions by returning the error to the user	Optional
Project must be open-source	Optional

## 5.