Macrosoft

Boolean Logic Calculator
Software Requirements Specifications

Version <0.1>

Revision History

Date	Version	Description	Author
23/03/2024	0.1	Software Specifications filled out	Ahmad Awan

Table of Contents

Software Requirements Specifications

1.Introduction

The objective of this Software Requirements Specification (SRS) is to delineate the requirements and additional details pertaining to the C++ Arithmetic Evaluator (referred to as the project herein). Detailed project specifications, timeline, and deadlines are documented in the project's Software Development Plan (SDP), accessible through Section 1.3 Definitions, Acronyms, and Abbreviations, and/or Section 1.4 References.

This SRS comprehensively outlines all systems and subsystems within the project, providing insight into the project's context and specific product perspectives. Additionally, it elucidates constraints, assumptions, as well as both core and non-core requirements derived from these considerations.

1.1.Purpose

The purpose of the SRS is to describe the project. The project should be a C++ program can parse and evaluate arithmetic expressions containing the following operators:

- AND(&)
- OR(|)
- NOT(!)
- NAND(@)
- XOR($)

The program should also be able to handle expressions with parentheses to define precedence and grouping.

1.2.Scope

The SRS encompasses all systems and subsystems of the project, providing a comprehensive overview of both functional and non-functional requirements expected throughout the entirety of the project.

1.3.Definitions, Acronyms, and Abbreviations

The project/this project/the program: “C++ Boolean Logic Calculator”, described by the Software Development Plan (SDS). The repository for all code and documentation lives on GitHub, at https://github.com/sawan201/EECS-348-Project/tree/main

SRS: Software Requirements Specification. See Section 1.4 References - SRS.

SDP: Software Development Plan. See Section 1.4 References – SDP.

Functional Requirement: Program functionality that requires code modification to meet the requirement.
Supplemental Requirement: Program documentation or otherwise that does not require code modification but can improve useability and usefulness of program.

1.4.References

SRS: Available at https://github.com/sawan201/EECS-348-Project/tree/main.
SDP: Available at https://github.com/sawan201/EECS-348-Project/tree/main.

1.5.Overview

The remaining sections of the document cover an overarching description, product perspectives—including assumptions, user characteristics, and interfaces—as well as functional and non-functional requirements.

2.Overall Description

The Boolean Logic Calculator is envisioned as a command-line application designed to accept a Boolean logic expression as input and generate the corresponding evaluation result. Implemented in C++, the program will employ diverse programming techniques, potentially incorporating recursion, stacks, linked lists, and other methodologies.

2.1.Product perspective

2.1.1.User Interfaces

The Boolean logic calculator will have a simple command-line user interface. The user will be able to enter a Boolean logic calculator as input and the program will output the result of evaluating the expression.

2.1.2.Software Interfaces

The Boolean logic calculator will not have any software interfaces.

2.1.3.Memory Constraints

The Boolean logic calculator parser will have minimal memory constraints. The program will only need to store the Boolean logic expression being parsed and the result of evaluating the expression.

2.1.4.Operations

The arithmetic expression parser will be able to perform the following operations:

- Parse Boolean logic expressions
- Evaluate Boolean logic expressions

2.2.Product functions

The program will be able to parse and evaluate Boolean logic expressions containing the following operators &, |, !, @, and $. The program should be able to handle expressions with parentheses to define precedence and grouping.

2.3.User characteristics

The arithmetic expression parser is designed to cater to students, programmers, and other users requiring the parsing and evaluation of arithmetic expressions.

2.4.Constraints

The implementation of the Boolean logic calculator must utilize the C++ programming language. Additionally, it should be compatible with various operating systems such as Linux, macOS, and Windows.

2.5.Assumptions and dependencies

The following assumptions are made about the Boolean logic calculator:

- The user can enter any valid or invalid Boolean logic expression.
- The user may enter invalid characters

3.Specific Requirements

3.1.Functionality

This section and its subsections describe the **Functional Requirements.**

3.1.1.Parse Boolean Logic Expression

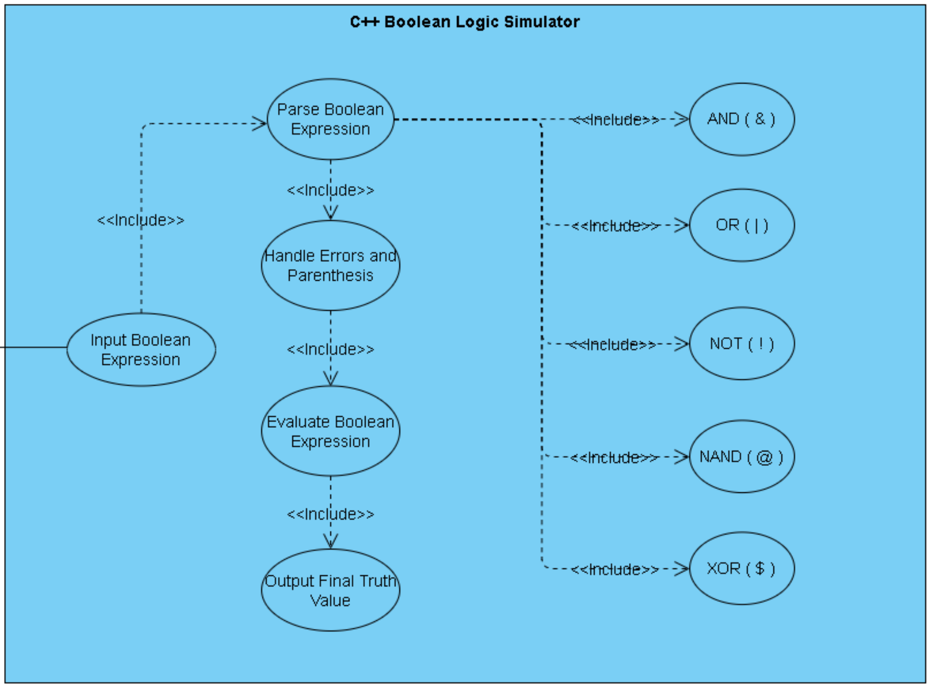
The software should possess the capability to parse and assess Boolean logic expressions incorporating operators such as &, |, !, @, and $. Furthermore, additional requirements outline particular interactions among different boolean operators and operands or delineate subsystems extending beyond generic Boolean expressions.

3.1.2.Boolean Logic Operators

This program supports Boolean logic operands. Example: 1 & 1. Or, 0 | 1.

3.2.Use-Case Specifications

Below is a model of the use case diagram of the project.



3.3.Supplementary Requirements

This section and associated subsections describe the **Supplemental Requirements.**

4.Classification of Functional Requirements

Functionality	Type
Parse Boolean Logic Expression	Essential
Boolean Logic Operators	Essential

5.Appendices

Appendices will either be specified as part of the SRS. As of current revision, there are no appendices.