
SQUISHY CATS<Company Name>

LogiCalc
Software Requirements Specifications
Version 1.0

LogiCalc	Version: 1.0
Software Requirements Specifications	Date: 03/24/2024
B1	

Revision History

Date	Version	Description	Author
03/22/2024	0.1	Document Initiation	Olivia, Mariam
03/24/2024	1.0	First version completed	Olivia, Mariam

LogiCalc	Version: 1.0
Software Requirements Specifications	Date: 03/24/2024
B1	

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	User Interfaces	5
2.1.2	Software Interfaces	5
2.1.3	Memory Constraints	5
2.2	Product Functions	5
2.3	User Characteristics	5
2.4	Constraints	5
2.5	Assumptions and Dependencies	5
3.	Specific Requirements	5
3.1	Functionality	5
3.1.1	Operator Support	5
3.1.2	Expression Parsing	6
3.1.3	Truth Value Input	6
3.1.4	Evaluation and Output	6
3.1.5	Error Handling	6
3.1.6	Parenthesis Handling	6
3.2	Use-Case Specifications	6
3.2.1	Evaluate Boolean Expression	6
3.2.2	Define Truth Values	6
3.2.3	Handle Errors in Expressions	7
3.2.4	Respect Operator Precedence	7
3.2.5	Support Complex Expressions	7
3.2.6	Provide Clear Output	7
3.3	Supplementary Requirements	8
3.3.1	Input Specifications	8
3.3.2	Errors	8
3.3.3	Reliability	8
4.	Classification of Functional Requirements	8
5.	Appendices	8

LogiCalc	Version: 1.0
Software Requirements Specifications	Date: 03/24/2024
B1	

Software Requirements Specifications

1. Introduction

This document, Software Requirements Specifications (SRS), details the requirements to build a CLI Boolean Algebra Calculator. The program, which allows users to input different Boolean Algebra expressions, is created to fulfill the requirements of the EECS 348 course project. This document reports the requirements based on the official meetings of the members of Squishy Cats of EECS 348, with the product owner, Mariam Oraby.

1.1 Purpose

This document provides the user characteristics and functions of the program and specifies the details of all functional and quality requirements of the program. This document is intended for Professor Hossein Saiedian, our team members, and teaching assistants reviewing the project.

1.2 Scope

The command-line interface program, named LogiCalc, is unified for the use of anyone calculating Boolean algebra expressions. The program prompts the user for an expression (in specific order), using different Boolean expression operators. The program then takes in the user input, runs the functions necessary for the correct output, and displays the output for the user. The program will not solve/accept expressions outside of the desired order, which is specified in the CLI. After the program runs, the user may either quit the program or run the program again to input another expression.

1.3 Definitions, Acronyms, and Abbreviations

CLI: command line interface

SRS: software requirements specifications

1.4 References

For the Software Requirements Specifications, the list of referenced and supporting artifacts includes:

- Team roles
- Software Development Plan
- Iteration Plans
- Software Architecture
- Test Cases
- User Manual

1.5 Overview

This Software Requirements Specifications document contains the following information:

Overall Description — Describes the general factors that affect the product and its requirements.

Specific Requirements — Contains use-case specifications and software requirements in detail

Classification of Functional Requirements — Functional requirements and their type

Appendices — States whether the appendices are considered in SRS

LogiCalc	Version: 1.0
Software Requirements Specifications	Date: 03/24/2024
B1	

2. Overall Description

2.1 Product perspective

2.1.1 User Interfaces

The UI should be easy to manipulate without additional training. The user should be able to interact with the program using the CLI to input properly formatted expressions. The program is fully accessible via the CLI.

2.1.2 Software Interfaces

The program is used via a Command-Line Interface (CLI), where the user can input necessary data to use the program.

2.1.3 Memory Constraints TBD

2.2 Product functions

- 2.2.1 Operator Support
- 2.2.2 Expression Parsing
- 2.2.3 Truth Value Input
- 2.2.4 Evaluation and output
- 2.2.5 Error Handling
- 2.2.6 Parenthesis Handling

2.3 User characteristics

The users of this product should have basic knowledge of Boolean Logic.

2.4 Constraints

The software must adhere to C++ programming language standards and have a user-friendly interface for interacting with the evaluator.

2.5 Assumptions and dependencies

The software assumes that users have a basic understanding of Boolean logic.

3. Specific Requirements

This section of the SRS contains all software requirements to a level of detail sufficient to enable the team to design a system to satisfy those requirements, and to test that the system satisfies those requirements.

3.1 Functionality

The program allows users to input complex Boolean expressions using Boolean operators such as AND, OR, NOT, NAND, XOR, along with parentheses to specify precedence, handles parsing and evaluation, and provides informative error messages for any issues encountered during the process, ensuring a reliable and user-friendly experience for working with Boolean logic.

3.1.1 Operator support

Implement logical operations for the following:

- AND (&): Returns True if both operands are True.
- OR (|): Returns True if at least one operand is True.
- NOT (!): Inverts the truth value of its operand.

LogiCalc	Version: 1.0
Software Requirements Specifications	Date: 03/24/2024
B1	

- NAND (@): Returns True only if both operands are False (opposite of AND).
- XOR (\$): Returns True if exactly one operand is True.

3.1.2 *Expression Parsing*

The program parses user-provided Boolean expressions in infix notation, respecting operator precedence and parentheses. Infix notation means the operators are placed between operands, as opposed to prefix or postfix notations where operators come before or after operands, respectively.

3.1.3 *Truth Value Input*

Users can define truth values (True or False) for each variable represented by 'T' and 'F' in the expressions.

3.1.4 *Evaluation and Output*

After parsing the expression and assigning truth values to variables, the program calculates the final truth value of the entire expression and presents it clearly as either True or False.

3.1.5 *Error Handling*

The program implements robust error handling for various issues such as invalid expressions, missing parentheses, unknown operators, circular logic, empty expressions, double operators, missing truth values for variables, inconsistent characters, and invalid characters.

3.1.6 *Parenthesis Handling*

The program ensures that expressions enclosed within parentheses are evaluated first to determine the order of evaluation, respecting the precedence rules.

3.2 **Use-Case Specifications**

3.2.1 *Evaluate Boolean Expression*

Actor: User

Description: The user inputs a Boolean expression using the supported logical operators and parentheses. The program parses, evaluates, and returns the final truth value of the expression.

Functional Requirements:

- Accept user input for Boolean expressions.
- Parse the input expression to identify variables, operators, and parentheses.
- Evaluate the expression according to operator precedence and truth values of variables.
- Display the result (True or False) to the user.

3.2.2 *Define Truth Values*

Actor: User

Description: The user defines truth values (True or False) for each variable represented by 'T' and 'F' in the Boolean expression.

Functional Requirements:

- Allow users to assign truth values to variables before evaluating expressions.
- Validate user input to ensure truth values are correctly assigned.
- Use assigned truth values when evaluating expressions.

LogiCalc	Version: 1.0
Software Requirements Specifications	Date: 03/24/2024
B1	

3.2.3 *Handle Errors in Expressions*

Actor: System

Description: The program detects and handles errors in user-provided Boolean expressions to ensure accurate evaluation.

Functional Requirements:

- Check for missing operands, unknown operators, mismatched parentheses, circular logic, empty expressions, and invalid characters.
- Provide informative error messages to users indicating the nature of the error.
- Prevent the evaluation of expressions with errors and prompt users to correct them.

3.2.4 *Respect Operator Precedence*

Actor: System

Description: The program respects operator precedence rules while evaluating Boolean expressions to ensure correct logical computations.

Functional Requirements:

- Implement precedence rules for logical operators (NOT > AND > OR > NAND > XOR).
- Use parentheses to override default precedence and specify evaluation order.
- Evaluate expressions according to the defined precedence rules.

3.2.5 *Support Complex Expressions*

Actor: System

Description: The program supports complex Boolean expressions with multiple operators, parentheses, and variables.

Functional Requirements:

- Handle expressions with nested parentheses to determine the order of evaluation.
- Evaluate expressions containing multiple operators (AND, OR, NOT, NAND, XOR) correctly.
- Handle large expressions without performance issues.

3.2.6 *Provide Clear Output*

Actor: System

Description: The program provides clear and understandable output to users after evaluating Boolean expressions.

Functional Requirements:

- Display the final truth value of the expression (True or False) in a clear format.
- Include any intermediate steps or evaluations if necessary for transparency.
- Ensure the output is easy to interpret and use for further analysis or decision-making.

LogiCalc	Version: 1.0
Software Requirements Specifications	Date: 03/24/2024
B1	

3.3 Supplementary Requirements

3.3.1 Input Specifications

The program will run efficiently and reliably if the user-input complies with the necessary operations and specifications needed for a Boolean expression.

3.3.2 Errors

The program will output a meaningful error message depending on the error to occur in the program. The error message will specify whether the error occurred during input or output.

3.3.3 Reliability

The program should work reliably at any point. After the user's input has been entered into the program and the program runs, the output for the Boolean expression is clear and concise. The output will represent the final truth value of the expression (True or False)

4. Classification of Functional Requirements

Functionality	Type
Operator support	Essential
Expression Parsing	Essential
Truth Value Input	Essential
Evaluation and Output	Essential
Error Handling	Essential
Parenthesis Handling	Essential

5. Appendices

This section of the Software Requirements Specifications (SRS) is not to be included as part of the project's requirements. This section does not include any additional information, as there is no cost to be estimated for this project.