<Project Name>

Version <1.0>

[Note: The following template is provided for use with the Unified Process for EDUcation. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 3/14/2024 | 2.1 | Part 1 | Ibrahim |
| 3/22/2024 | 2.2 | Updated section 1 | Katie |
|  |  |  |  |
|  |  |  |  |

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 Software Interfaces 5

2.1.5 Communication Interfaces 5

2.1.6 Memory Constraints 5

2.1.7 Operations 5

2.2 Product functions 5

2.3 User characteristics 5

2.4 Constraints 5

2.5 Assumptions and dependencies 5

2.6 Requirements subsets 5

3. Specific Requirements 5

3.1 Functionality 5

3.1.1 <Functional Requirement One> 6

3.2 Use-Case Specifications 6

3.3 Supplementary Requirements 6

4. Classification of Functional Requirements 6

5. Appendices 6

# Introduction

## Purpose

The goal of this is to act as a document that lays out the needs for the Boolean Logic Calculator project. Its purpose is to offer an explicit explanation of how the application should function. This involves defining the tasks for the Boolean Logic Calculator and any restrictions or boundaries that need to be followed throughout the development process.

## Scope

The scope of this document is to document the software side of requirements for this project. This program is designed to interpret and assess Boolean logic input, by users. It has operators like AND, OR, NOT, XOR and more which are described in section 3: Specific Requirements. The program will return true or false using the supported logic operators based on the user’s prompt.

## Definitions, Acronyms, and Abbreviations

AND – The AND operation takes two Boolean inputs and produces an output that is true only when both inputs are true.

OR – The OR operation takes two Boolean inputs and produces an output that is true when one input is true, or both are true.

NOT - This operation takes a Boolean value and returns the opposite Boolean value.

NAND - It combines AND and NOT. This returns the inverted value of AND.

NOR- It combines OR and NOT. This returns the inverted value of OR.

XOR – The XOR operation takes two Boolean inputs and produces an output that is true if exactly one of the inputs is true.

Boolean – Boolean expressions take one of two possible values, true or false. Under positive logic, 1 is used to denote true and 0 is used to denote false.

## References

EECS 348 Spring 2024 Project Description Document obtained from Professor Hossein Saiedian.

## Overview

[This subsection describes what the rest of the **SRS** contains and explains how the document is organized.]

The overview section describes the contents of the SRS and explains how the document is organized. It provides a brief preview of subsequent sections, including requirements, design constraints, and supporting information for the Boolean Expression Evaluator software project.

# Overall Description

[This section of the **SRS** describes the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3, and makes them easier to understand. Include such items as:

## Product perspective

### System Interfaces

### User Interfaces

### Hardware Interfaces

### Software Interfaces

### Communication Interfaces

### Memory Constraints

### Operations

## Product functions

## User characteristics

## Constraints

## Assumptions and dependencies

## Requirements subsets

# Specific Requirements

[This section of the **SRS** contains all software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. When using use-case modeling, these requirements are captured in the Use Cases and the applicable supplementary specifications. If use-case modeling is not used, the outline for supplementary specifications may be inserted directly into this section, as shown below.]

## Functionality

[This section describes the functional requirements of the system for those requirements that are expressed in the natural language style. For many applications, this may constitute the bulk of the **SRS** package and thought should be given to the organization of this section. This section is typically organized by feature, but alternative organization methods may also be appropriate; for example, organization by user or organization by subsystem. Functional requirements may include feature sets, capabilities, and security.

Where application development tools, such as requirements tools, modeling tools, and the like, are employed to capture the functionality, this section of the document would refer to the availability of that data, indicating the location and name of the tool used to capture the data.]

### <Functional Requirement One>

[The requirement description.]

## Use-Case Specifications

[In use-case modeling, the use cases often define the majority of the functional requirements of the system, along with some non-functional requirements.]

## Supplementary Requirements

[Supplementary Specifications capture other requirements, e.g., non-functional requirements and development constraints, that are not included in the use cases and non-functional requirements.]

# Classification of Functional Requirements

[List, usually in a table, all functional requirements and order them by Type (Essential, Desirable, and Optional) or by order of appearance in the document.]

|  |  |
| --- | --- |
| **Functionality** | **Type** |
| ... |  |
| ... |  |

# Appendices

[When appendices are included, the **SRS** should explicitly state whether or not the appendices are to be considered part of the requirements]