Software Requirements Specification (SRS)

For the project titled

Number Cracker

Prepared by: Dipsana Roy
Version 1.0

1. Introduction

The Software Requirements Specification (SRS) document provides a detailed description of the NumberQuest game, a terminal-based number guessing game developed in C. This document follows the IEEE standard for SRS documentation and serves as a reference for developers, testers, and maintainers of the project.

1.1 Purpose

The purpose of this document is to describe the software requirements for the NumberQuest application. The document defines the functional and non-functional requirements necessary for the development, maintenance, and enhancement of the game.

1.2 Scope

NumberQuest is a simple yet engaging number guessing game designed for console execution. The player is prompted to guess a randomly generated number between 1 and 100. The game provides hints after each attempt (higher/lower) and tracks the total number of attempts taken. It validates input and allows continuous play with user-controlled exit options.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS** – Software Requirements Specification
- **C** – Programming Language used for implementation
- **Console/Terminal** – The command-line interface where the program runs

2. Overall Description

2.1 Product Perspective

NumberQuest is a standalone console-based application that requires no external libraries beyond the standard C library. It operates in a text-based environment and does not rely on any network or database connection.

2.2 Product Functions

- Generate a random number between 1 and 100
- Accept player input for guessing
- Validate user input for correctness
- Provide hints ("Higher" or "Lower")
- Display the number of attempts after successful guess
- Offer replay or exit options

2.3 User Characteristics

The game is intended for casual players of all ages with basic familiarity with command-line usage. No programming experience is required to play the game.

2.4 Constraints

- The application must run on a terminal or command-line interface.
- Input must be numeric and within the range 1–100.
- Game logic must rely solely on the standard C library.

3. Specific Requirements

3.1 Functional Requirements

- **FR1:** The system shall generate a random number between 1 and 100.
- **FR2:** The system shall accept user input and validate it.
- **FR3:** The system shall display hints ("Higher" or "Lower") based on the guess.
- **FR4:** The system shall count and display the total number of attempts upon success.
- **FR5:** The system shall ask the user whether to play again or exit.

3.2 Non-Functional Requirements

- **NFR1:** The system shall respond to each user input within one second.
- **NFR2:** The system shall require less than 200 KB of memory.
- **NFR3:** The system shall be portable across Windows, Linux, and macOS.
- **NFR4:** The system shall be developed in ISO C99 standard.

4. System Features

- Simple command-line interface for easy interaction
- Smart input validation for invalid or out-of-range entries
- Randomized number generation for replay ability
- Replay option to continue gameplay without restarting the program

5. Future Enhancements

- Add difficulty modes (Easy, Medium, Hard) with variable ranges

- Introduce a scoring system based on attempts
- Add color-coded feedback for terminal outputs
- Implement a time-based challenge mode

6. Appendix

This document conforms to the IEEE 830-1998 standard for Software Requirements Specifications.