Software Requirements Specification (SRS) for Sudoku Game

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the Sudoku game software. It will outline the functional and non-functional requirements, user interface requirements, and other necessary specifications to guide the development process.

1.2 Scope

The Sudoku game is a console-based application that allows users to play Sudoku puzzles of varying difficulty levels (Easy, Medium, Hard). The application will generate a valid Sudoku puzzle, allow users to input their moves, and validate those moves against the rules of Sudoku.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS**: Software Requirements Specification
- **Sudoku**: A logic-based combinatorial number-placement puzzle.
- **User **: Any individual who interacts with the Sudoku game.

1.4 References

- Sudoku Rules
- C++ Programming Language

2. Overall Description

2.1 Product Perspective

The Sudoku game will be a standalone application that runs in a console environment. It will not depend on any external systems or services.

2.2 Product Functions

- Generate a valid Sudoku puzzle based on the selected difficulty level.
- Display the current state of the Sudoku grid.
- Allow users to input their moves (row, column, number).
- Validate user inputs and moves.
- Check if the Sudoku puzzle is complete.

2.3 User Classes and Characteristics

- **Casual Players**: Users who play Sudoku for fun and may not be familiar with all the rules.

- **Experienced Players**: Users who are familiar with Sudoku and may seek more challenging puzzles.

2.4 Operating Environment

- The application will run on Windows, Linux, and macOS operating systems.
- It will require a terminal or console for user interaction.

2.5 Design and Implementation Constraints

- The application must be implemented in C++.
- The user interface will be text-based and will not include graphical elements.

3. Specific Requirements

3.1 Functional Requirements

3.1.1 Puzzle Generation

- **Description**: The system shall generate a valid Sudoku puzzle based on the selected difficulty level.
- **Inputs**: Difficulty level (1 for Easy, 2 for Medium, 3 for Hard).
- **Outputs**: A partially filled Sudoku grid.

3.1.2 Display Grid

- **Description**: The system shall display the current state of the Sudoku grid.
- **Outputs**: A visual representation of the grid in the console.

3.1.3 User Input

- **Description**: The system shall allow users to input their moves.
- **Inputs**: Row (1-9), Column (1-9), Number (1-9).
- **Validation**: The system shall validate the input to ensure it is within the allowed range.

3.1.4 Move Validation

- **Description**: The system shall check if the user's move is valid according to Sudoku rules.
- **Outputs**: Confirmation of a correct move or an error message for an invalid move.

3.1.5 Completion Check

- **Description**: The system shall check if the Sudoku puzzle is complete.
- **Outputs**: A message indicating whether the user has completed the puzzle.

3.2 Non-Functional Requirements

3.2.1 Performance

- The system shall generate a Sudoku puzzle within 2 seconds.

3.2.2 Usability

- The user interface shall be intuitive and easy to navigate for both casual and experienced players.

3.2.3 Reliability

- The system shall handle invalid inputs gracefully without crashing.

3.2.4 Portability

- The application shall run on multiple operating systems (Windows, Linux, macOS) without modification.

4. User Interface Requirements

4.1 Console Interface

- The application shall use a text-based console interface.
- The grid shall be displayed in a clear format, with rows and columns labeled.
- User prompts shall be clear and informative.

5. Other Requirements

5.1 Security

- The application shall not store any user data or personal information.

5.2 Maintenance

- The code shall be well-documented to facilitate future maintenance and updates.

This SRS document serves as a foundational guide for the development of the Sudoku game. You can expand on each section as needed, depending on the complexity and specific requirements of your project.

6.1 Glossary

- **Grid**: The 9x9 matrix used in Sudoku, consisting of 81 cells.
- **Cell**: An individual square in the Sudoku grid where a number can be placed.
- **Valid Move**: A move that adheres to Sudoku rules, meaning no duplicate numbers in the same row, column, or 3x3 subgrid.

6.2 Use Case Diagram

- A visual representation of the interactions between users and the system can be included here to illustrate the main functionalities.

6.3 User Stories

- As a casual player, I want to play Sudoku without needing to understand complex rules, so that I can enjoy the game.
- As an experienced player, I want to select a difficulty level, so that I can challenge myself.

7. Future Enhancements

- Implement a timer feature to track how long it takes to complete a puzzle.
- Add a hint system that provides users with suggestions for valid moves.
- Create a leaderboard to track high scores and completion times.

8. Conclusion

This SRS document outlines the essential requirements for the Sudoku game. It serves as a comprehensive guide for developers and stakeholders to ensure that the final product meets user expectations and functions as intended. Further iterations of this document may be necessary as the project evolves and additional features are considered.