

Software Requirements Specification (SRS)

1. Introduction

1.1 Purpose

The purpose of this SRS document is to provide a detailed description of the requirements for the Advanced Tic Tac Toe Game. It will outline the functional and non-functional requirements, system behavior, and performance criteria to ensure that the game meets the desired specifications.

1. Scope

The Advanced Tic Tac Toe Game will be a software application that allows users to play Tic Tac Toe against another player or an AI opponent. The application will include user authentication, personalized game history tracking, and an intelligent AI opponent. The game will be developed using C++ and will feature a graphical user interface created with Qt.

2. Overall Description

2.1 Product Perspective

The Advanced Tic Tac Toe Game is an independent software product that will be used for entertainment and educational purposes. It will be designed to run on desktop computers with support for different operating systems using the Qt framework.

2.2 Product Functions

- User registration and login.
- Playing Tic Tac Toe in player-vs-player and player-vs-AI modes
- Intelligent AI opponent using the minimax algorithm with alpha-beta pruning.

2.3 User Classes and Characteristics

- **Players:** Individuals who will play the game. They may be either registered users or guests.

2.4 Operating Environment

- Desktop application running on Windows
- Developed using C++ and Qt.
- Data storage using SQLite database.

2.5 Design and Implementation Constraints

- Must follow the Google C++ Style Guide.
- Must implement secure password hashing and session management.
- Must integrate CI/CD using GitHub Actions..

2.6 Assumptions and Dependencies

- Users have basic knowledge of using desktop applications.
- Qt libraries are installed on the user's system.
- Internet connection required for user authentication and updates.

3. Specific Requirements

3.1 Functional Requirements

3.1.1 User Registration and Authentication

- **FR1.1:** The system shall allow users to create an account with a unique username and password.
- **FR1.2:** The system shall use secure hashing algorithms to store user passwords. using SHA256 including the Qcryptographic library
- **FR1.3:** The system shall allow users to log in using their username and password.
- **FR1.4:** The system shall manage user sessions and keep users logged in until they manually log out.

3.1.2 Game Mechanics

- **FR2.1:** The system shall allow two players to play Tic Tac Toe on a 3x3 grid.
- **FR2.2:** The system shall allow a player to play against an AI opponent.
- **FR2.3:** The system shall check for a win, loss, or tie after each move.
- **FR2.4:** The system shall display the score of sequence games.

3.1.3 AI Opponent

- **FR3.1:** The AI opponent shall use the minimax algorithm with alpha-beta pruning.
- **FR3.2:** The AI opponent shall make moves based on the current game state to maximize its chances of winning.

3.1.4 Game History

- **FR4.1:** The system shall save the game history for each registered user.
- **FR4.2:** The system shall allow users to view their past games.

3.2 Non-Functional Requirements

3.2.1 Performance

- **NFR1.1:** The system shall wait for the user to enter the data.
- **NFR1.2:** The AI opponent's move calculation shall not exceed 1 second.

3.2.2 Usability

- **NFR2.1:** The GUI shall be intuitive and easy to navigate.
- **NFR2.2:** The system shall provide feedback for user actions (e.g., move confirmation, error messages).

3.2.3 Maintainability

- **NFR5.1:** The codebase shall follow the Google C++ Style Guide.
- **NFR5.2:** The system shall include comprehensive unit and integration tests.

3.3 External Interface Requirements

3.3.1 User Interfaces

- **UI1:** The main game screen displaying the Tic Tac Toe board.
- **UI2:** Login and registration forms.

3.3.2 Hardware Interfaces

- **HI1:** The system shall interact with standard desktop hardware (keyboard, mouse).

3.3.3 Software Interfaces

- **SI1:** The system shall use SQLite for data storage.
- **SI2:** The system shall integrate with GitHub Actions for CI/CD.

3.4 System Features

3.4.1 Player-vs-Player Mode

- **SF1:** The system shall support two players taking turns.

3.4.2 Player-vs-AI Mode

- **SF2:** The system shall support a player competing against the AI
- **SF3:** We have a competition: beat the AI if you can, this is a message to users.

4. Appendices

4.1 Appendix A: Glossary

- **Minimax Algorithm:** A recursive algorithm for decision-making and game theory.
- **Alpha-Beta Pruning:** An optimization technique for the minimax algorithm.

4.2 Appendix B: References

- Google C++ Style Guide
- Qt Documentation