Software Requirements Specification (SRS)

1. Introduction

1.1 Purpose

The purpose of this document is to specify the software requirements for a multiplayer & AI Tic-Tac-Toe application with a graphical user interface (GUI) developed using Qt. This application allows two players to compete in a game of Tic-Tac-Toe, allows a player to compete with an AI, records game results, and maintains player statistics.

1.2 Scope

This application provides the following features:

- User login and registration
- Single-player and multiplayer gameplay
- Game result recording and statistics tracking
- GUI for interaction

2. Overall Description

2.1 Product Perspective

The Tic-Tac-Toe application is a standalone desktop application. It provides a GUI for user interaction and uses an SQLite database to store user data and game statistics.

2.2 Product Functions

- User authentication (login and sign-up)
- Play Tic-Tac-Toe in single-player and multiplayer modes
- Display game results
- Maintain player statistics (wins, losses, draws)
- View match history and replay it.

2.3 User Classes and Characteristics

- **Regular Users**: Users who log in or sign up to play the game, view their profile.
- **Guest Users**: Users who play the game without logging in. Their statistics are not saved and they have no profile.

2.4 Design and Implementation Constraints

- The application was developed using the Qt framework.
- SQLite is used for database management.

2.5 Assumptions and Dependencies

- Users have a valid username and password for logging in.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- **Login Window**: Allows users to log in, sign up, or play as a guest.
- **Main Window**: Displays the main menu and game options.
- **Multiplayer Window**: Displays the game board and allows two players to play Tic-Tac-Toe.
- **AI Window**: Displays the game board and allows the user to play against AI.
- **Profile window**: Displays match history and player statistics.

3.1.2 Hardware Interfaces

- Standard keyboard and mouse for user interaction.

3.1.3 Software Interfaces

- SQLite database for storing user data and game statistics.
- Qt framework for the GUI.

3.2 Functional Requirements

3.2.1 User Authentication

- The system allows users to log in using a username and password, create new account, or play as guest.

3.2.2 Gameplay

- The system allows users to start a multiplayer game.
- The system allows users to start a single-player game versus AI.
- The system displays the game board with 9 buttons representing the grid.
- The system allows players to take turns marking the grid.
- The system checks for a win or draw after each move.
- The system displays the game result (win/lose/draw).

3.2.3 Game Results and Statistics

- The system records the game result in the database.
- The system updates player statistics (wins, losses, draws).
- The system displays player statistics on request (profile).

3.3 Non-Functional Requirements

3.3.1 Performance Requirements

- Hard move function first execution takes around 30 milliseconds then the execution time reduces a lot with every execution in one game.
- board check takes around 43 milliseconds at the first execution and then it takes around 37 milliseconds
- the login process takes 17 milliseconds
- the replay process takes around 16 milliseconds
- anything else needs access to database takes around 17 milliseconds
- all other functions are less than 1 millisecond
- transition between windows take around 23 milliseconds

- Very small memory size required.

3.3.2 Security Requirements

- The system provides hashing algorithm on user passwords before storing them in database.
- The system uses prepared statements to prevent SQL injection.

3.3.3 Usability Requirements

- The system provides a user-friendly interface with clear instructions.
- The system is accessible to users with basic computer skills.

4. System Features

4.1 Login Window

4.1.1 Description and Priority

The login window is the entry point for users. It allows users to log in, sign up, or play as a guest.

4.1.2 Functional Requirements

- The system displays fields for username and password.
- The system validates user credentials against the database.
- The system displays error messages for invalid login attempts (wrong username or password).

- The system displays error messages for invalid sign-up attempts (username or email already exist).
- The system provides a link to the sign-up window.
- The system allows guest users to proceed without logging in.

4.2 Main Window

4.2.1 Description and Priority

The main window displays the main menu and game options.

4.2.2 Functional Requirements

- The system displays buttons for starting a single-player or multiplayer game.
- The system displays player profile and sign-out button.

4.3 Multiplayer Window

4.3.1 Description and Priority

The multiplayer window allows two players to compete in a game of Tic-Tac-Toe.

4.3.2 Functional Requirements

- The system displays the game board with 9 buttons.
- The system allows players to take turns marking the grid.
- The system checks for a win or draw after each move.
- The system displays the game result.
- The system records the game result in the database.
- The system updates player statistics.

4.4 AI Window

4.4.1 Description and Priority

The AI window allows user to compete against AI with 3 different difficulties in a game of Tic-Tac-Toe.

4.4.2 Functional Requirements

- The system displays the game board with 9 buttons.
- The system allows player and AI to take turns marking the grid.
- One of the difficulties is implemented using minimax algorithm.
- The current player's turn is displayed.
- The player starts the game with symbol “X”.
- The system checks for a win or draw after each move.

- The system displays the game result.
- The system records the game result in the database.
- The system updates player statistics.

4.4.3 Game Over

- The game declares the result as one of the following: “You win!”, “AI win!”, or “It’s a Tie!”.
- The game statistics are added in the SQLite database, including player username, winner (either username or AI), moves made, and game ID.
- User’s profile is updated with the last game in match history and number of wins, loses, or draws.

4.5 Game Statistics

4.5.1 Description and Priority

The system maintains and displays match history and ability to replay previous games.

4.5.2 Functional Requirements

- The system stores player statistics (wins, losses, draws).
- The system displays player profile with match history and statistics.
- The system replays old games from match history on request.

5. Other Requirements

5.1 Database Requirements

- The database stores user credentials securely.
- The database stores game results and player statistics.
- The database supports concurrent access.

5.2 Security Requirements

- The system uses secure hashing algorithms for password storage.
- The system validates all inputs.

5.3 Usability Requirements

- The system provides clear instructions and feedback to the user.
- The system has an intuitive and user-friendly interface.
- Game theme is colors are simple and comfortable on the eyes.