Advanced Tic Tac Toe Game

Software Requirements Specification (SRS)

Supervisor: Dr.Omar Nasr

Team: Chat GPA

Name	Student-ID	BN

CONTENTS

1. Introduction

1.1 Purpose

The purpose of this document is to detail all the functional and non-functional requirements for the development of a Tic-Tac-Toe game application. This document will serve as a guide for developers, testers, and stakeholders to understand the system's behavior, performance requirements, and game rules.

1.2 Scope

The Tic-Tac-Toe game allows human players to sign up, sign in, play against other human players or AI opponents of varying difficulty levels, and view game histories. The game records and displays the history of moves and game results.

1.3 Definitions, Acronyms, and Abbreviations

- **AI**: Artificial Intelligence
- **SRS**: Software Requirements Specification
- **User**: A human player who interacts with the game
- **Game Board**: A 3x3 grid where players place their symbols ('X' or 'O')

2 Overall Description

2.1 Product Perspective

The Tic-Tac-Toe game is a standalone application designed for entertainment and educational purposes. It can be used by individuals who want to play a quick game of Tic-Tac-Toe against other humans or AI opponents.

2.2 Product Functions

- User Sign-Up and Sign-In
- Play Game (Human vs. Human, Human vs. AI)
- View Game History
- Save Game History
- Display Game Board and Moves

2.3 User Classes and Characteristics

Regular User: A human player who wants to play Tic-Tac-Toe. Regular users can sign up, sign in, play games, and view their game history.

2.4 Operating Environment

- The application will run on any standard computer with a modern C++ compiler.
- The application will use file-based storage for user data and game history.

2.5 Design and Implementation Constraints

- The application will be implemented using C++.
- User data and game history will be stored in plain text files.

2.6 Assumptions and Dependencies

- Users will have access to a computer with a modern C++ compiler.
- Users will be familiar with basic command-line operations.

3 Specific Requirements

3.1 Functional Requirements

3.1.1 User Authentication

- **Sign-Up:**
  - Users must provide a unique username and a password to sign up.
  - The system must check for duplicate usernames and prompt for a different username if necessary.
- **Sign-In:**
  - Users must provide their username and password to sign in.
  - The system must authenticate the user by checking the provided credentials against stored data.

3.1.2 Gameplay

- **Start Game:**
  - Users can start a new game after signing in.
  - Users can choose to play against another human or an AI opponent.
- **Make Move:**
  - Human players must be able to input their move.
  - The system must validate the move (i.e., check if the move is within bounds and the chosen cell is empty).
- **AI Move:**
  - The system must provide AI players with different difficulty levels (random, medium, hard).
  - The AI must make a valid move based on its difficulty algorithm.
- **End Game:**
  - The game ends when a player wins, or the board is full.
  - The system must announce the winner or a draw.

3.1.3 Game History

- **Save Game History:**
  - The system must save the game history after each game.
  - The history must include player names, winner, and moves.
- **View Game History:**
  - Users must be able to view their game history.
  - The system must display a list of past games and allow users to select a game to view detailed moves.

3.2 Non-Functional Requirements

3.2.1 Performance

- The game must respond to user inputs within 1 second.
- The AI must make its move within 2 seconds.

3.2.2 Usability

- The game must have a simple and intuitive command-line interface.
- Instructions and prompts must be clear and concise.

3.2.3 Reliability

- The system must handle incorrect inputs gracefully.
- User data and game history must be stored persistently.

3.2.4 Maintainability

- The code must be well-documented to facilitate future maintenance.
- The design must follow standard coding conventions.

3.3 Game Rules

- The game is played on a 3x3 grid.
- Players take turns placing their symbol (X or O) in an empty cell.
- The first player to place three of their symbols in a row (horizontally, vertically, or diagonally) wins.
- If all cells are filled and no player has three in a row, the game is a draw.

4 Specific Requirements

4.1 Sample Input/Output

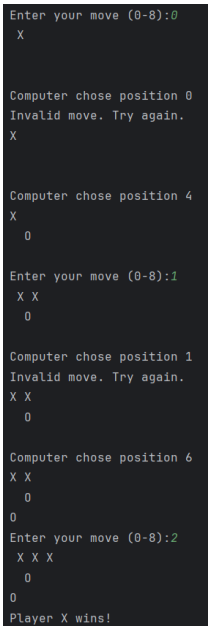
- Sample sign-up:

```
Welcome to X-O Game. :)
Select an option:
1. Sign Up
2. Sign In
3. Exit
Enter your choice (1-3):1
 Enter username:Ahmed_Amr
 Enter password:123456
 Sign up successful!
```

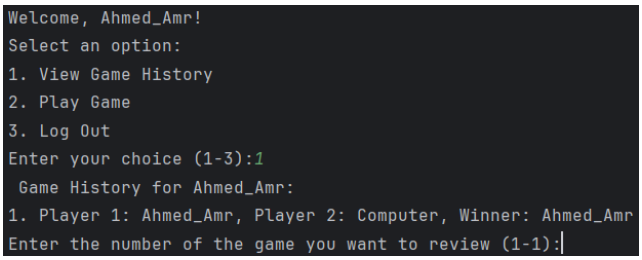
- Sample sign-in:

```
Welcome to X-O Game. :)
Select an option:
1. Sign Up
2. Sign In
3. Exit
Enter your choice (1-3):2
 Enter username:Ahmed_Amr
 Enter password:123456
 Sign in successful!
Welcome, Ahmed_Amr!
Select an option:
1. View Game History
2. Play Game
3. Log Out
Enter your choice (1-3):
```

- Sample Gameplay:



- Sample Game History:



- Sample Game Review: