Advanced Tic Tac Toe Game

Software Requirements Specifications (SRS)

By

The X Factor

Table of Contents

1. Introduction

1.1. Purpose ……..1

1.2. Intended Audience ……...……...……..…1

1.3. Intended Use……...……2

1.4. Product Scope ……...……2

1.5. Definitions and Acronyms ……..….4

2. Overall Description

2.1. User Needs……5

2.2. Assumptions and Dependencies ……..……..6

3. System Features and Requirements

3.1. Functional Requirements ……...……7

3.1.1. User Authentication ……7

3.1.2. Game Mechanics ……...……7

3.1.3. AI Opponent ……..……8

3.1.4. Game History ……8

3.1.5. Graphical User Interface (GUI) ……8

3.1.6. Testing and Quality Assurance ……9

3.1.7. CI/CD Integration ……9

3.2. External Interface Requirements ……..…9

3.2.1. User Interfaces ……9

3.2.2. Hardware Interfaces ……..10

3.2.3. Software Interfaces ……10

3.2.4. Communication Interfaces ……..…10

3.3. System Features ……...…11

3.3.1. Interactive Tic Tac Toe Gameplay ……11

3.3.2. User Authentication and Profile Management ……..…11

3.3.3. Game History and Replay ……..……12

3.3.4. Intelligent AI Opponent ……..…12

3.3.5. Graphical User Interface (GUI) ……..…12

3.3.6. Testing and Quality Assurance ……12

3.3.7. Performance Optimization ……13

3.4. Non-Functional Requirements ……13

3.4.1. Performance Requirements ……13

3.4.2. Reliability Requirements ……..…13

3.4.3. Security Requirements ……..…13

3.4.4. Usability Requirements ……..…14

3.4.5. Maintainability Requirements ……14

3.4.6. Portability Requirements ……..…14

3.4.7. Scalability Requirements ……..…14

3.4.8. Legal and Regulatory Requirements ……..…15

1. Introduction

1.1 Purpose

The purpose of this project is to develop an advanced version of the classic Tic Tac Toe game, incorporating modern features to enhance user engagement and experience. This advanced version will include user authentication to manage individual player profiles and game histories, as well as an intelligent AI opponent to provide a challenging gameplay experience. The game will support both player-vs-player and player-vs-AI modes, allowing users to log in, track their game history, and analyze past games. By employing best practices in software engineering, including secure user management, rigorous testing, and professional version control workflows, this project aims to deliver a high-quality, reliable, and enjoyable Tic Tac Toe game. The project also seeks to provide an educational experience in software development, promoting the use of strategic algorithms, comprehensive testing, and continuous integration and deployment processes.

1.2 Intended Audience

This Software Requirements Specification (SRS) document is intended for the following stakeholders involved in the development and utilization of the advanced Tic Tac Toe game:

1. **Project Team Members**:
    - **Developers**: Responsible for implementing the game, including its core mechanics, AI opponent, and user management features.
    - **Designers**: Tasked with creating the graphical user interface (GUI) and ensuring a user-friendly experience.
    - **Testers**: Charged with developing and executing unit, integration, and system tests to ensure the software's quality and reliability.
    - **Project Managers**: Overseeing the project's progress, ensuring that milestones are met, and maintaining communication among team members.
2. **End Users**:
    - **Players**: Individuals who will interact with the game, including casual players and those interested in a more strategic challenge against the AI opponent. This group includes both new users and returning players who wish to track their game history and progress.
3. **Academic and Educational Institutions**:
    - **Students and Educators**: Utilizing the project as a learning tool to understand software development practices, including user authentication, AI development, testing methodologies, and continuous integration/deployment workflows.
4. **Reviewers and Evaluators**:
    - **Academic Reviewers**: Professors and academic staff responsible for evaluating the project's implementation and adherence to software engineering principles.
    - **Peers and Mentors**: Providing feedback and guidance throughout the development process to ensure the project's success.

By catering to these diverse groups, this SRS aims to provide a clear and comprehensive guide to the requirements and expectations for the development of the advanced Tic Tac Toe game.

1.3 Intended Use

The advanced Tic Tac Toe game is intended to serve multiple purposes across different user groups:

1. **Entertainment**:
    - **Players**: To provide a fun and engaging game experience for users of all ages. The game offers both player-vs-player and player-vs-AI modes, catering to those seeking a casual game with friends or a challenging solo experience against an intelligent opponent.
2. **Educational Tool**:
    - **Students and Educators**: To act as a practical example of software development principles and best practices. The project demonstrates the implementation of user authentication, AI algorithms, GUI design, and comprehensive testing, making it a valuable resource for teaching and learning purposes.
3. **Skill Development**:
    - **Developers**: To enhance the technical skills of developers working on the project by providing hands-on experience with C++, object-oriented programming, and modern software engineering techniques such as CI/CD and secure coding practices.
4. **Project Management**:
    - **Project Managers and Teams**: To serve as a case study for effective project management, including task delegation, version control, and continuous integration/deployment. It offers insights into collaborative development processes and the importance of maintaining high code quality.
5. **Performance Analysis**:
    - **Researchers and Enthusiasts**: To analyze and optimize the performance of a real-world application. The project includes monitoring and optimizing metrics such as response time and system resource utilization, which are critical for delivering a smooth user experience.

By addressing these intended uses, the project aims to deliver value not only as a game but also as a comprehensive educational and developmental tool.

1.4 Product Scope

The scope of the advanced Tic Tac Toe game project encompasses the following key features and functionalities:

1. **Game Mechanics**:
    - **Core Gameplay**: Implement the classic Tic Tac Toe rules on a 3x3 grid, allowing two players or one player against an AI to take turns marking the grid.
    - **Win/Tie Detection**: Automatically check for win conditions or ties after each move and display the results accordingly.
2. **Artificial Intelligence (AI) Opponent**:
    - **AI Algorithms**: Develop a challenging AI opponent using the minimax algorithm with alpha-beta pruning or other strategic techniques to provide a competitive gameplay experience.
    - **Difficulty Levels**: Offer multiple difficulty levels for the AI to cater to players of varying skill levels.

3. **User Authentication and Management**:
    - **Registration and Login**: Implement secure user authentication, allowing players to create accounts, log in, and manage their profiles.
    - **Password Security**: Use secure hashing algorithms to store passwords safely.
    - **Session Management**: Maintain user sessions to keep players logged in during their gameplay.
4. **Personalized Game History**:
    - **Game Tracking**: Save detailed records of each game session, including moves made, outcomes, and timestamps.
    - **History Viewing**: Allow players to view and analyze their past games through an intuitive interface.
    - **Replay Feature**: Enable users to replay previous games to study strategies and improve their skills.
5. **Graphical User Interface (GUI)**:
    - **User-Friendly Design**: Create an interactive and visually appealing GUI using frameworks like Qt to ensure an engaging user experience.
    - **Interactive Elements**: Include elements such as clickable game board, login and registration forms, and game history views.
6. **Testing and Quality Assurance**:
    - **Unit Testing**: Develop comprehensive unit tests for individual components using Google Test to ensure correctness.
    - **Integration Testing**: Conduct integration tests to verify that different parts of the system work together seamlessly.
    - **Automated Testing**: Integrate continuous testing processes using GitHub Actions to maintain high code quality.
7. **Continuous Integration and Deployment (CI/CD)**:
    - **Automated Workflows**: Set up CI/CD pipelines using GitHub Actions to automate the building, testing, and deployment processes.
    - **Version Control**: Utilize Git and GitHub for effective version control and collaboration.
8. **Performance Optimization**:
    - **Monitoring**: Continuously monitor the game's performance, focusing on metrics such as response time and resource utilization.
    - **Optimization**: Implement strategies to optimize performance and ensure a smooth and responsive user experience.

By defining these comprehensive features and functionalities, the product scope outlines the development goals and boundaries for the advanced Tic Tac Toe game project, ensuring that all critical aspects are addressed to deliver a high-quality, feature-rich game.

1.5 Definitions and Acronyms

This section provides definitions and acronyms used throughout the Software Requirements Specification (SRS) document to ensure clarity and a common understanding of terms.

Definitions:

- **Game Mechanics**: The rules and systems that govern the gameplay of Tic Tac Toe, including how players take turns, win, or tie.
- **Artificial Intelligence (AI)**: The simulation of human intelligence in the game, allowing the system to make decisions and play against human opponents.
- **Minimax Algorithm**: A recursive algorithm used in decision-making and game theory to minimize the possible loss for a worst-case scenario.
- **Alpha-Beta Pruning**: An optimization technique for the minimax algorithm that reduces the number of nodes evaluated by the algorithm.
- **Graphical User Interface (GUI)**: The visual component of the software that allows users to interact with the game through graphical elements like buttons and text fields.
- **User Authentication**: The process of verifying the identity of a user attempting to access the game system.
- **Session Management**: Techniques used to manage user sessions, keeping users logged in and maintaining their interactions with the system.
- **Continuous Integration (CI)**: A development practice where developers frequently integrate code into a shared repository, triggering automated tests.
- **Continuous Deployment (CD)**: A software release process that uses automated testing to validate changes, ensuring that code changes are automatically deployed to production.
- **Unit Testing**: Testing individual components of the software to ensure they function correctly in isolation.
- **Integration Testing**: Testing combined parts of an application to ensure they work together as expected.

Acronyms:

- **SRS**: Software Requirements Specification
- **AI**: Artificial Intelligence
- **GUI**: Graphical User Interface
- **CI**: Continuous Integration
- **CD**: Continuous Deployment
- **OOP**: Object-Oriented Programming
- **Qt**: A cross-platform GUI framework for C++ applications
- **SQL**: Structured Query Language, used for managing databases
- **Git**: A version control system for tracking changes in source code
- **SDS**: Software Design Specification

By providing these definitions and acronyms, this section ensures that all stakeholders have a clear and consistent understanding of the terminology used in this SRS document.

2. Overall Description

2.1 User Needs

The advanced Tic Tac Toe game project aims to meet the following user needs:

1. **Engaging Gameplay**:
    - **Interactive Experience**: Users need an engaging and interactive game experience with smooth, intuitive controls and responsive gameplay.
    - **Game Modes**: Users require the flexibility to play in both player-vs-player and player-vs-AI modes to enjoy different types of challenges.
2. **Challenging AI Opponent**:
    - **Intelligent AI**: Users expect a challenging AI opponent that can provide a competitive game, utilizing strategic decision-making algorithms to simulate a human-like playing experience.
    - **Adjustable Difficulty Levels**: Users need the ability to select different difficulty levels for the AI to match their skill level and preferences.
3. **User Authentication and Personalization**:
    - **Secure Login and Registration**: Users require a secure system for creating accounts and logging in to ensure that their personal information and game data are protected.
    - **Profile Management**: Users need the ability to manage their profiles, including updating personal information and viewing game history.
4. **Game History and Analysis**:
    - **Game Tracking**: Users want to track their game history, including the details of each match played, to analyze their performance and strategies.
    - **Replay Feature**: Users need a feature to replay past games, allowing them to review moves and improve their skills.
5. **User-Friendly Interface**:
    - **Intuitive GUI**: Users expect a user-friendly graphical interface that is easy to navigate, visually appealing, and enhances the overall gaming experience.
    - **Accessibility**: Users need the interface to be accessible, accommodating various devices and screen sizes.
6. **Performance and Reliability**:
    - **Smooth Performance**: Users require the game to perform smoothly without lag or crashes, providing a seamless gaming experience.
    - **Reliable Functionality**: Users need the game to function reliably, with all features working as intended and minimal downtime.
7. **Learning and Improvement**:
    - **Educational Value**: Users, especially students and educators, seek educational value from the project, using it as a tool to learn about software development, AI, and game design.
    - **Feedback and Updates**: Users need regular feedback mechanisms and updates to ensure the game remains relevant, functional, and enjoyable.

By addressing these user needs, the advanced Tic Tac Toe game project aims to provide a comprehensive, enjoyable, and educational gaming experience that caters to a wide range of users.

2.2 Assumptions and Dependencies

The development and successful implementation of the advanced Tic Tac Toe game project are based on the following assumptions and dependencies:

1. **Technology Stack**:
    - **Assumption**: The project will use C++ as the primary programming language, leveraging object-oriented programming techniques.
    - **Dependency**: Availability and compatibility of necessary development tools and libraries, such as Qt for the graphical user interface and Google Test for testing.
2. **Development Environment**:
    - **Assumption**: All team members will have access to a consistent and reliable development environment, including necessary hardware and software tools.
    - **Dependency**: Access to integrated development environments (IDEs), version control systems (Git), and continuous integration tools (GitHub Actions).
3. **User Base**:
    - **Assumption**: Users have basic computer literacy and familiarity with playing digital games, including an understanding of Tic Tac Toe rules.
    - **Dependency**: User devices meet the minimum system requirements to run the game smoothly.
4. **Team Collaboration**:
    - **Assumption**: Effective communication and collaboration among team members will be maintained throughout the project.
    - **Dependency**: Use of collaboration tools such as Slack, GitHub, and project management software to facilitate coordination and track progress.
5. **Security and Privacy**:
    - **Assumption**: Users expect their personal information and game data to be stored securely and handled with confidentiality.
    - **Dependency**: Implementation of secure authentication mechanisms and adherence to best practices in data encryption and privacy protection.
6. **AI Implementation**:
    - **Assumption**: The AI opponent will be implemented using algorithms like minimax with alpha-beta pruning to provide a challenging experience.
    - **Dependency**: Availability of resources and documentation for AI algorithm implementation and integration.
7. **Performance and Reliability**:
    - **Assumption**: The game will perform efficiently and reliably on the target platforms without significant bugs or performance issues.
    - **Dependency**: Rigorous testing and optimization to ensure smooth gameplay and quick response times.
8. **Educational Objectives**:
    - **Assumption**: The project will serve as a valuable learning tool for students and educators.
    - **Dependency**: Availability of documentation, tutorials, and resources to support the educational objectives and facilitate learning.

9. **Regulatory Compliance**:

    - **Assumption**: The project will comply with relevant regulations and standards for software development and data protection.
    - **Dependency**: Adherence to industry standards, such as GDPR for data protection and relevant software development guidelines.

By clearly identifying these assumptions and dependencies, the project team can better anticipate potential challenges and ensure that all necessary conditions are met for the successful development and deployment of the advanced Tic Tac Toe game.

3. System Features and Requirements

3.1 Functional Requirements

2. User Authentication

1.1 **User Registration**

- The system shall allow users to register an account by providing a username and password.
- The system shall validate the uniqueness of the username and ensure the password meets security requirements (e.g., minimum length, complexity).

1.2 **User Login**

- The system shall allow registered users to log in by providing their username and password.
- The system shall validate the credentials against stored records and grant access upon successful authentication.

1.3 **Password Security**

- The system shall store user passwords securely using hashing algorithms.
- The system shall not store passwords in plain text.

1.4 **Session Management**

- The system shall maintain user sessions securely, ensuring that users remain logged in across multiple interactions until they choose to log out.

2. Game Mechanics

2.1 **Basic Gameplay**

- The system shall allow two players to take turns marking a 3x3 grid with their respective symbols (X and O).
- The system shall alternate turns between the two players until the game ends.

2.2 **Win Condition**

- The system shall check for a win condition after each move, which occurs when a player has three of their symbols in a row, column, or diagonal.
- The system shall declare the winning player and end the game if a win condition is met.

2.3 **Tie Condition**

- The system shall check for a tie condition when all grid cells are filled and no player has met the win condition.
- The system shall declare the game a tie and end the game if a tie condition is met.

3. AI Opponent

3.1 **AI Gameplay**

- The system shall provide an AI opponent option for single-player mode.
- The AI opponent shall take turns against the player, marking the grid with its symbol.

3.2 **Minimax Algorithm**

- The AI opponent shall use the minimax algorithm with alpha-beta pruning to determine its moves.
- The AI opponent shall evaluate the game state and make strategic moves to challenge the player.

4. Game History

4.1 **Save Game History**

- The system shall save the details of each game session, including the players, moves, and outcome (win, loss, tie).
- The system shall securely store game histories in a database or a customized file format.

4.2 **View Game History**

- The system shall allow users to view their past game histories.
- The system shall display game histories in an organized manner, showing details such as the date of the game, opponent, and result.

4.3 **Replay Past Games**

- The system shall enable users to replay past games from their history.
- The system shall provide an interface for reviewing moves step-by-step as they occurred in the original game.

5. Graphical User Interface (GUI)

5.1 **Tic Tac Toe Board**

- The system shall display the Tic Tac Toe board through a graphical user interface.
- The system shall allow players to interact with the board by clicking on cells to make their moves.

5.2 **User Management Interface**

- The system shall provide GUI elements for user login and registration.
- The system shall display user profiles, including options for managing account details.

5.3 **Game History Interface**

- The system shall provide GUI elements for viewing and interacting with game histories.
- The system shall include features for selecting and replaying past games.

6. Testing and Quality Assurance

6.1 **Unit Testing**

- The system shall include unit tests for validating the functionality of individual components.
- The unit tests shall be implemented using the Google Test framework.

6.2 **Integration Testing**

- The system shall include integration tests to ensure that all components work together seamlessly.
- The integration tests shall cover interactions between game mechanics, AI, GUI, and user management.

7. CI/CD Integration

7.1 **Automated Testing**

- The system shall use GitHub Actions to automate the execution of unit and integration tests.
- The system shall trigger automated tests upon each code commit to ensure continuous quality assurance.

7.2 **Automated Deployment**

- The system shall use GitHub Actions to automate the deployment process.
- The system shall ensure that updates are deployed smoothly and maintain high code quality through continuous integration and deployment practices.

These functional requirements outline the key features and behaviors expected of the Advanced Tic Tac Toe Game, ensuring a robust and user-friendly application.

3.2 External Interface Requirements

1. User Interfaces

1.1 **Login Screen**

- The system shall provide a login screen where users can enter their username and password.
- The login screen shall include options for navigating to the registration screen.

1.2 **Registration Screen**

- The system shall provide a registration screen where users can create a new account by entering a username and password.
- The registration screen shall include input validation to ensure the username is unique.

1.3 **Main Menu**

- The system shall provide a main menu screen after successful login, offering options such as starting a new game, viewing game history, and logging out.
- The main menu shall be intuitive and easy to navigate.

1.4 **Game Screen**

- The system shall provide a game screen displaying the Tic Tac Toe board.
- The game screen shall allow players to interact with the game by clicking on cells to place their symbols.
- The game screen shall display the current player's turn and update the board after each move.

1.5 **Game History Screen**

- The system shall provide a game history screen where users can view a list of their past games.
- The game history screen shall include details such as the date of each game, the opponent, and the game result.
- The game history screen shall allow users to select and replay past games.

2. Hardware Interfaces

2.1 **User Devices**

- The system shall be compatible with standard input devices such as a mouse and keyboard for user interactions.
- The system shall display correctly on screens with a resolution of at least 1024x768 pixels.

3. Software Interfaces

3.1 **Database**

- The system shall interface with a SQLite database (or a similar lightweight database system) for storing user data and game histories.
- The system shall perform CRUD (Create, Read, Update, Delete) operations on the database to manage user accounts and game data.

3.2 **Operating System**

- The system shall be compatible with major operating systems, including Windows
- The system shall utilize standard system libraries for networking, file I/O, and graphical display.

4. Communication Interfaces

4.1 **Network Communication**

- The system shall use standard network protocols (e.g., TCP/IP) to communicate with remote servers for any online functionality (if applicable).
- The system shall secure all network communications using encryption protocols such as TLS. (If needed)

4.2 **Inter-Process Communication**

- The system shall handle inter-process communication where necessary, such as between the main application and background services for tasks like saving game history or managing user sessions.

These external interface requirements outline the necessary interactions between the Advanced Tic Tac Toe Game and its external environment, ensuring compatibility and user accessibility.

3.3 System Features

1. Interactive Tic Tac Toe Gameplay

1.1 **Two-Player Mode**

- The system shall support a two-player mode where two human players can take turns to play the game on a 3x3 grid.
- The system shall display the current player's turn and update the board accordingly after each move.

1.2 **Single-Player Mode with AI**

- The system shall provide a single-player mode where a human player can play against an AI opponent.
- The AI opponent shall use strategic algorithms to make decisions and provide a challenging gameplay experience.

1.3 **Win and Tie Detection**

- The system shall detect and announce a win condition when a player achieves three of their symbols in a row, column, or diagonal.
- The system shall detect and announce a tie condition when all grid cells are filled and no player has won.

2. User Authentication and Profile Management

2.1 **Secure Registration and Login**

- The system shall provide secure registration and login functionality for users to create and access their accounts.
- User credentials shall be securely stored using hashing algorithms to ensure data protection.

2.2 **Profile Management**

- The system shall allow users to manage their profiles, including updating personal information and changing passwords.
- The system shall ensure that profile updates are securely saved and reflected in the user's account.

3. Game History and Replay

3.1 **Save Game History**

- The system shall automatically save the details of each game session, including the players, moves, and results.
- Game histories shall be securely stored in a database or a customized file format for later retrieval.

3.2 **View Game History**

- The system shall provide an interface for users to view their past game histories, displaying details such as the date, opponent, and outcome of each game.
- The game history view shall be organized and easily navigable.

3.3 **Replay Past Games**

- The system shall enable users to replay past games from their history, allowing them to review each move step-by-step.
- The replay feature shall visually simulate the original game, showing moves in the order they were made.

4. Intelligent AI Opponent

4.1 **AI Strategy Implementation**

- The system shall implement an AI opponent using the minimax algorithm with alpha-beta pruning, ensuring strategic and challenging gameplay.
- The AI opponent shall evaluate the game state and make optimal moves to challenge the human player.

5. Graphical User Interface (GUI)

5.1 **User-Friendly Design**

- The system shall provide a user-friendly GUI that displays the Tic Tac Toe board and allows for intuitive interactions.
- The GUI shall include visual indicators for player turns, game status, and game results.

5.2 **User Management Interface**

- The GUI shall include screens for user registration, login, and profile management.
- The user management interface shall be visually appealing and easy to navigate.

5.3 **Game History Interface**

- The GUI shall provide a game history screen where users can view and interact with their past game records.
- The game history interface shall include features for selecting and replaying past games.

6. Testing and Quality Assurance

6.1 **Automated Testing**

- The system shall include automated unit and integration tests to ensure the functionality and reliability of all components.
- Automated tests shall be executed using GitHub Actions to maintain continuous quality assurance.

6.2 **Manual Testing**

- In addition to automated tests, the system shall undergo manual testing to verify usability and overall user experience.
- Manual testing shall involve scenarios covering typical user interactions and edge cases.

7. Performance Optimization

7.1 **Efficiency Metrics**

- The system shall monitor and optimize performance metrics such as response time, memory usage, and CPU utilization.
- Performance optimizations shall ensure that the game runs smoothly on supported hardware and software environments.

These system features describe the core functionalities and interactions expected in the Advanced Tic Tac Toe Game, providing a detailed overview for developers and stakeholders.

3.4 Non Functional Requirements

1. Performance Requirements

1.1 **Response Time**

- The system shall respond to user inputs within 1 second to ensure a smooth and interactive user experience.
- The system shall process AI opponent moves within 2 seconds to maintain game flow and user engagement.

1.2 **Concurrency**

- The system shall handle up to 1000 concurrent users without performance degradation, ensuring scalability for a large user base.

2. Reliability Requirements

2.1 **System Uptime**

- The system shall have an uptime of 99.9%, ensuring high availability and minimal downtime for users.

2.2 **Error Handling**

- The system shall gracefully handle unexpected errors and provide meaningful error messages to users.

3. Security Requirements

3.1 **Data Protection**

- The system shall use secure hashing algorithms to store user passwords, ensuring that plaintext passwords are never stored or transmitted.
- The system shall encrypt sensitive data to protect against eavesdropping and man-in-the-middle attacks.

3.2 **Access Control**

- The system shall ensure that only authenticated users can access their personal data and game histories.
- The system shall implement role-based access control to restrict administrative functions to authorized personnel only.

4. Usability Requirements

4.1 **User Interface**

- The system shall provide an intuitive and user-friendly interface that is easy to navigate for users of all ages and technical backgrounds.
- The GUI shall be visually appealing and designed to enhance the user experience.

5. Maintainability Requirements

5.1 **Code Quality**

- The system shall adhere to the Google C++ Style Guide to ensure consistent and readable code.
- The system shall include comprehensive documentation for all code, including inline comments and external documentation.

5.2 **Modularity**

- The system shall be designed with modularity in mind, allowing for easy updates and maintenance of individual components without affecting the entire system.
- The system shall use object-oriented programming principles to enhance code reusability and maintainability.

6. Portability Requirements

6.1 **Operating System Compatibility**

- The system shall be compatible with major operating systems, including Windows
- The system shall leverage cross-platform development tools (e.g., Qt) to ensure consistent behavior across different environments.

6.2 **Hardware Compatibility**

- The system shall run efficiently on standard desktop and laptop hardware, without requiring specialized or high-performance components.
- The system shall display correctly on screens with a resolution of at least 1024x768 pixels.

7. Scalability Requirements

7.1 **Database Scalability**

- The system shall use a database that can scale to accommodate growing user data and game history records.
- The system shall optimize database queries to ensure quick retrieval and storage of data even as the database grows.

7.2 **Application Scalability**

- The system shall be designed to scale horizontally, allowing for the addition of new servers to handle increased load as the user base grows.
- The system shall support load balancing to distribute traffic evenly across multiple servers.

8. Legal and Regulatory Requirements

8.1 **Data Privacy**

- The system shall comply with relevant data privacy regulations, such as GDPR and CCPA, ensuring the protection of user data.
- The system shall provide users with the ability to manage their data, including viewing, updating, and deleting their personal information.

8.2 **Software Licensing**

- The system shall comply with the licensing terms of all third-party software and libraries used in the development process.
- The system shall include proper attribution and licensing information for all third-party components.

These non-functional requirements ensure that the Advanced Tic Tac Toe Game is not only functional but also reliable, secure, and user-friendly, meeting the expectations of users and stakeholders alike.