Table of contents

Software Requirements Specification

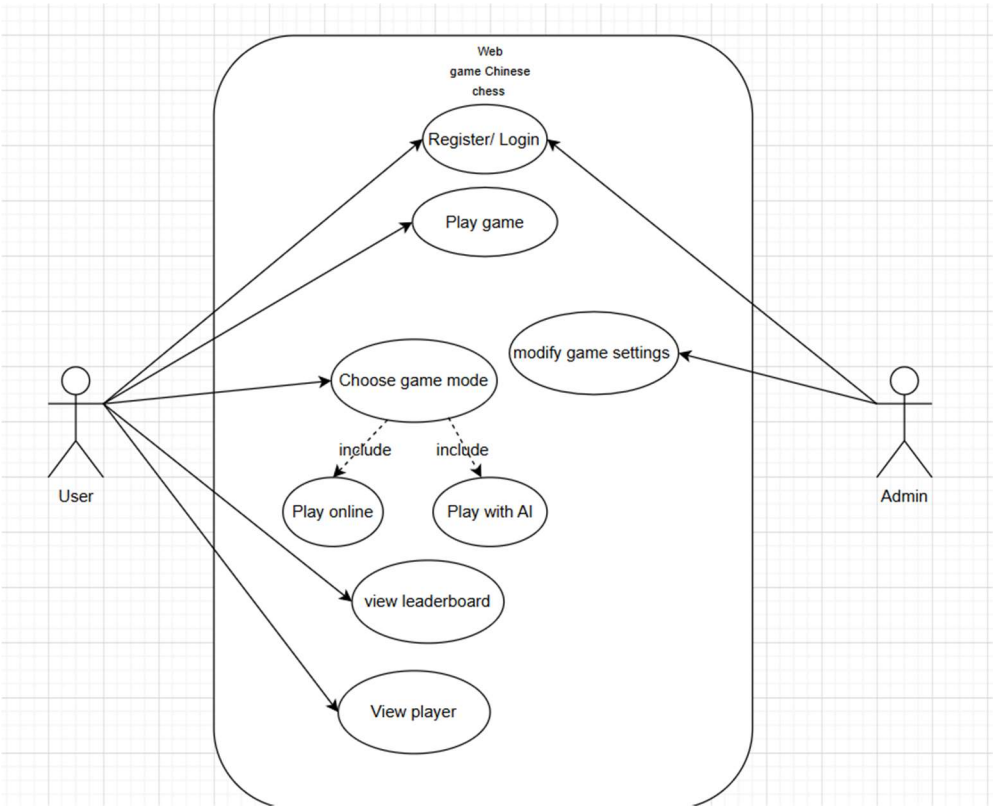
1. Purpose

- The purpose of this document is to define the software requirements for a chess application, ensuring that all functions, performance, and technical constraints are described clearly and accurately.
- The chess application will provide a platform for users to participate in chess online or offline, support playing with AI, playing with other people, and analyzing chess games. The main goals include:
    - Create an intuitive, friendly, and easy-to-use chess playing environment
    - Support multiple game modes, including playing with AI, playing with friends, or competing online
    - Ensure the accuracy of the rules according to international chess standards
    - Integrate analysis, save, and review features to help players improve their skills
    - Provide a ranking and statistics system to help players track their progress
    - Ensure good performance, stable operation on multiple platforms (Windows, Android, iOS, Web).

2. High Level Requirements

2.1.Actors and Use Cases Descriptions

2.1.1. Diagram



2.1.2. Actors Description

- **User (Player):** This represents a regular player who interacts with the game application.
- **Admin:** The admin is responsible for managing the system settings.

2.1.3. Use Cases Description

#	Code	Name	Brief Description
1	UC01	Register/Login	Allows actors to register/login to the system.
2	UC02	modify game settings	Allows admin to adjust parameters such as: game rules, difficulty level or server settings
3	UC03	Play game	Allows players to join a match.
4	UC04	Choose game mode	Allows players to choose game mode
5	UC05	Play online	Allows players to play online.
6	UC06	Play with AI	Allows players to play games with AI
7	UC07	View leaderboard	Allows players to check the rankings of top players.
8	UC08	View player	Allows players to browse player profiles and stats.

2.1.4. Use Case & Actor mapping

Actor / Use case	Admin	User (Player)
UC01: Resgiter/ Login	**x**	**x**
UC02: modify game settings		**x**
UC03: Play game	**x**	
UC04: Choose game mode	**x**	
UC05: Play online	**x**	
UC06: Play with AI	**x**	
UC07: View leaderboard	**x**	
UC08: View player	**x**	

3. Functional Requirements

3.1.UC01: Resgiter/ Login

- **Use Case Description**

Name	Resgiter/ Login	**Code**	UC01
Description	Allow actors to log in and register to the system		
Actor	Admin/ Users	**Trigger**	Actor clicks login button
Pre-condition			
Post condition	Go to default page with corresponding role		

- **Activities**

Actor	System
Main Flow: Login successful	

1	Actor enters username/password and clicks Resgiter/ Login button on login page		
		2	Check the username/password is correct then go to the corresponding default role page

System Message

N/A

3.2.modify game settings

- **Use Case Description**

Name	modify game settings	**Code**	UC02
Description	Admin can modify system data.		
Actor	Admin	**Trigger**	admin click on system settings
Pre-condition	must be admin to edit.		
Post condition	can modify game rules, rankings, match history, ...		

- **Activities**

Actor		**System**	
Main Flow: Admin modified the game successfully.			
1	admin click modify game settings to modify the game.		
		2	Modify the game according to the admin's programming.

System Message

N/A

3.3.Play game

- **Use Case Description**

Name	Play game	**Code**	UC03
Description	Users can play chess.		
Actor	User	**Trigger**	user click to play game
Pre-condition	User must log in to the system		
Post condition			

- **Activities**

Actor		System	
Main Flow:User play chinese chess successfully.			
1	User clicks on the game icon to play chess.		
		2	The system will automatically find a suitable opponent to play with the player.

System Message

N/A

3.4.Choose game mode

- **Use Case Description**

Name	Choose game mode	**Code**	UC04
Description	Users can choose the game mode.		
Actor	User	**Trigger**	user click to choose game mode.
Pre-condition	User must log in to the system		
Post condition			

- **Activities**

Actor		System	
Main Flow: User selects game mode successfully.			
1	User clicks on icon to select game mode.		
		2	The system will display 2 game modes for users to choose from: play online and play with AI.

System Message

N/A

3.5.Play online

- **Use Case Description**

Name	Play online	**Code**	UC05
Description	Users can play online with other players.		
Actor	User	**Trigger**	user click to play online.
Pre-condition	User must log in to the system		
Post condition			

- **Activities**

Actor		System	
Main Flow: The user successfully selected online play mode with other players.			
1	User clicks on icon to play online.		
		2	The system will find the player another player to play online.

System Message

N/A

3.6.Play with AI

- **Use Case Description**

Name	Play with AI	**Code**	UC06
Description	Users can practice with AI.		
Actor	User	**Trigger**	user click to play with AI.
Pre-condition	User must log in to the system		
Post condition			

- **Activities**

Actor		System	
Main Flow: User plays with AI successfully.			
1	User clicks on icon to play with AI.		
		2	The system will automatically connect the player with AI.

System Message

N/A

3.7.View leaderboard

- **Use Case Description**

Name	View leaderboard	**Code**	UC07
Description	Users can view the leaderboard of players with good performance.		
Actor	User	**Trigger**	user click to View leaderboard.
Pre-condition	User must log in to the system		
Post condition			

- **Activities**

Actor	System
Main Flow: User view leaderboard successfully.	

1	User clicks on icon to view leaderboard.		
		2	The system will display the ranking of the top player and the ranking of the user viewing.

System Message

N/A

3.8.View player

- **Use Case Description**

Name	View player	**Code**	UC08
Description	Users can view their own and other players' information.		
Actor	User	**Trigger**	user click to View player.
Pre-condition	User must log in to the system		
Post condition			

- **Activities**

Actor		**System**	
Main Flow: User view player successfully.			
1	User clicks on icon to view player.		
		2	The system will display user information and allow users to view other players' information.

System Message

N/A

4. Mockup Screens

5. Nonfunctional Requirements

5.1.Accessibility

- The website shall comply with WCAG 2.1 AA accessibility standards to support users with disabilities.
- The website shall support multiple languages, including Vietnamese and English.

5.2.Auditability

- The system shall log all user activities, including game moves, logins, and profile updates.
- An audit trail shall be maintained for at least six months for security and compliance purposes.

- The logs shall be tamper-proof and only accessible to authorized administrators.

5.3.Correctness

5.3.1. Accuracy

- The system shall enforce official Chinese Chess rules, ensuring valid move detection.
- The move validation algorithm shall correctly identify check, checkmate, stalemate, and perpetual check.
- Elo or alternative ranking calculations shall be accurate and follow the appropriate rating system.

5.3.2. Precision

- The game engine shall execute moves with a precision of ±50 milliseconds to ensure fair gameplay.
- The chess clock shall maintain an accuracy margin of ±50ms to prevent timing discrepancies.

5.4.Interoperability

- The website shall be compatible with modern web browsers, including Chrome, Firefox, Safari, and Edge.
- It shall support third-party Chinese chess engines via REST APIs.

5.5.Maintainability

- The system shall follow a modular architecture to allow for independent component updates and maintenance.
- The codebase shall be well-documented to facilitate future modifications and debugging.
- The website shall support automated deployments and updates to minimize downtime.

5.6.Performance

5.6.1. Capacity

- The system shall support at least 10,000 concurrent users without noticeable performance degradation.
- The database shall efficiently store and retrieve millions of game records with optimized indexing.

5.6.2. Response Time

- The website shall load within 3 seconds on a standard broadband connection (5 Mbps or higher).
- Moves shall be processed and displayed within 100ms of a player's action.

5.6.3. Throughput

- The system shall handle a minimum of 1,000 active game sessions per second without performance issues.
- The API shall process at least 500 requests per second during peak loads.

5.7.Portability

- The website shall be accessible on desktop, tablet, and mobile devices with a responsive design.
- The system shall be cross-platform, running on Windows.

5.8.Reliability

- The system shall maintain 99.9% uptime, ensuring a maximum annual downtime of 8.76 hours.
- Automatic failover mechanisms shall switch to a backup server within 10 seconds in case of a failure.
- The system shall support periodic automatic backups to prevent data loss.

5.9. Reusability

- The game engine, authentication module, and ranking system shall be designed as reusable components.
- The API shall follow RESTful architecture, allowing reuse in mobile apps and third-party platforms.

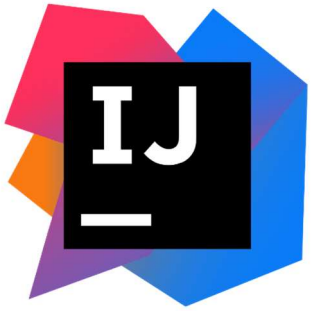
5.10. Robustness

- The system shall prevent illegal user inputs and game exploits through proper validation.
- It shall implement security measures against SQL injection, cross-site scripting (XSS), and denial-of-service (DoS) attacks.
- If a player disconnects, the game shall attempt reconnection within 10 seconds before declaring a forfeit.

6. Development Tools

- **Visual Studio Code:** Visual studio tool is used to code related to front-end of web game. Including front-end of home page, chess board, login and register.

- **IntelliJ IDEA:** IntelliJ tool is used to code the back-end for the game, such as: being able to play the game, being able to register and log into the game.



IntelliJ IDEA Ultimate

- **MongoDB:** MongoDB is used to store game data into the Database, including storing game registration accounts, game play history and player achievements.

7. Commercial and Development Direction.

- Developing and operating an online Chinese Chess platform requires a combination of technology and business strategy. Technically, the system can be built with a frontend using React.js or Vue.js, a backend using Node.js, Django or .NET, along with WebSockets to ensure a real-time gaming experience. Key features include multiplayer, AI opponents, a ranking system, automatic matchmaking, and a mobile-friendly interface. In terms of business strategy, the platform can adopt a freemium model, monetizing through advertising, membership subscriptions, or selling custom items such as chess boards and special chess pieces. The target audience includes the Chinese Chess enthusiast community in Chinese-speaking countries and globally. To grow users, marketing strategies can leverage SEO, partner with KOLs (influencers) in the Chinese Chess community, and organize online tournaments. The combination of modern technology and a reasonable business model will help the platform develop sustainably and attract players in the long term**.**