Detailed Design Document

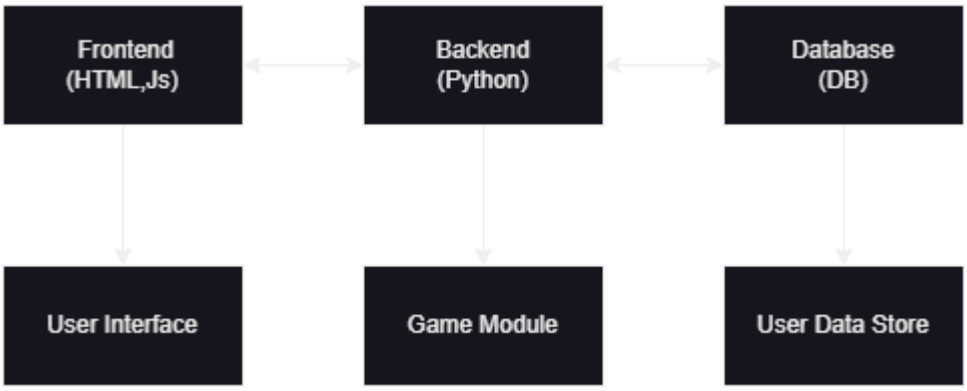
1. System Architecture Design

1.1 System Overview

The core purpose of this system is to provide users with an engaging and educational *Fei Hua Ling* game experience. The system achieves this by closely integrating the front-end and back-end functionalities, utilizing modular design to ensure scalability and maintainability. The system consists of three main parts:

- **Front-End**: Responsible for displaying the user interface, receiving user input, and delivering game feedback to the user. The front-end will be developed using modern web frameworks to ensure a responsive and user-friendly interface.
- **Back-End**: The back-end handles user requests, controls game flow, invokes the poem generation module, and stores user data. The back-end will be developed using Python.
- **Database**: The database is mainly used for storing user information, game history, and poem data. It will use MySQL or PostgreSQL to efficiently manage structured data.

1.2 System Architecture Diagram



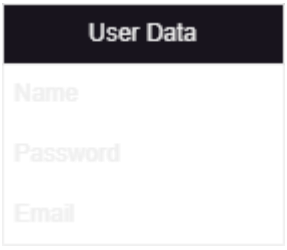
1.3 Main Module Descriptions

- **Front-End Module**:
  The front-end provides the user interface (UI) and handles user interactions. Using the Node.js framework, the front-end can quickly respond to user clicks and manage input/output interactions. Users can participate in the game, view poem analyses, and check historical records through the front-end interface.
- **Back-End Module**:
  The back-end handles the reception of requests from the front-end, processes data, and sends results back to the front-end. The back-end also calls the poem generation module, validates user input, and updates historical records in the database.
- **Database Module**:
  The database stores all data in the system, including user information. All data

interactions occur through the back-end with the database, ensuring data security and persistence.

- **Poem Generation Module**:
  The poem generation module is one of the core parts of the game. It uses pre-trained language models (such as GPT or other poem generation models) to generate poems that comply with the *Fei Hua Ling* rules.
- **Recommendation System**:
  The recommendation system analyzes players' historical behavior, providing personalized recommendations for poems, questions, or learning resources. These recommendations help players deepen their understanding of classical Chinese poetry and enhance the game experience.

2. Database Design

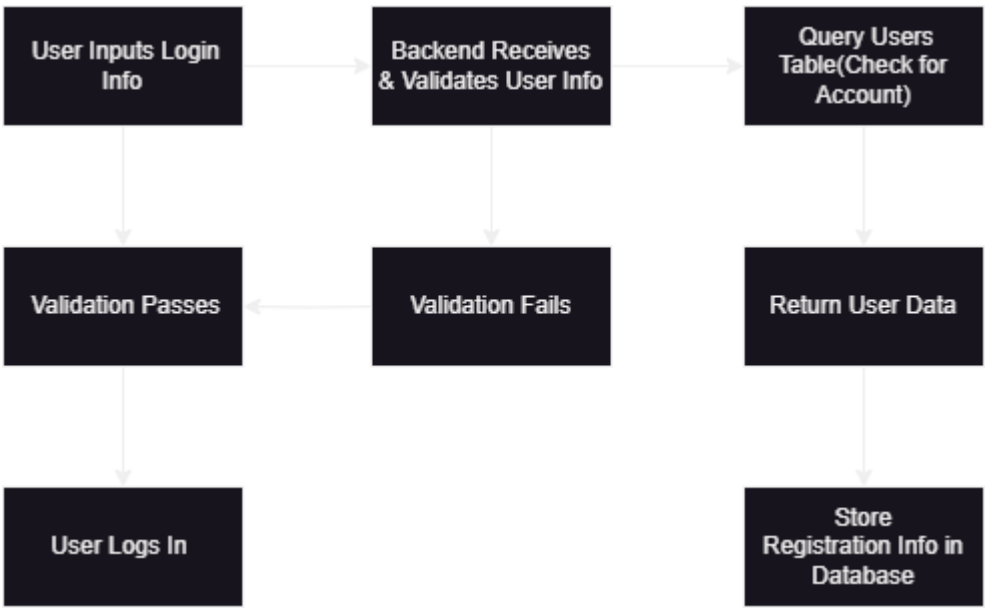


2.1 Database Table Structure

To manage user information, game history, and recommendation content, the following database table structures are designed:

- **User Table (Users)**: Stores basic user information, including username, password, email, etc.

2.2 Data Storage and Access



- **Data Storage**:
  All user data is stored in the database. Using SQL databases enables easy data

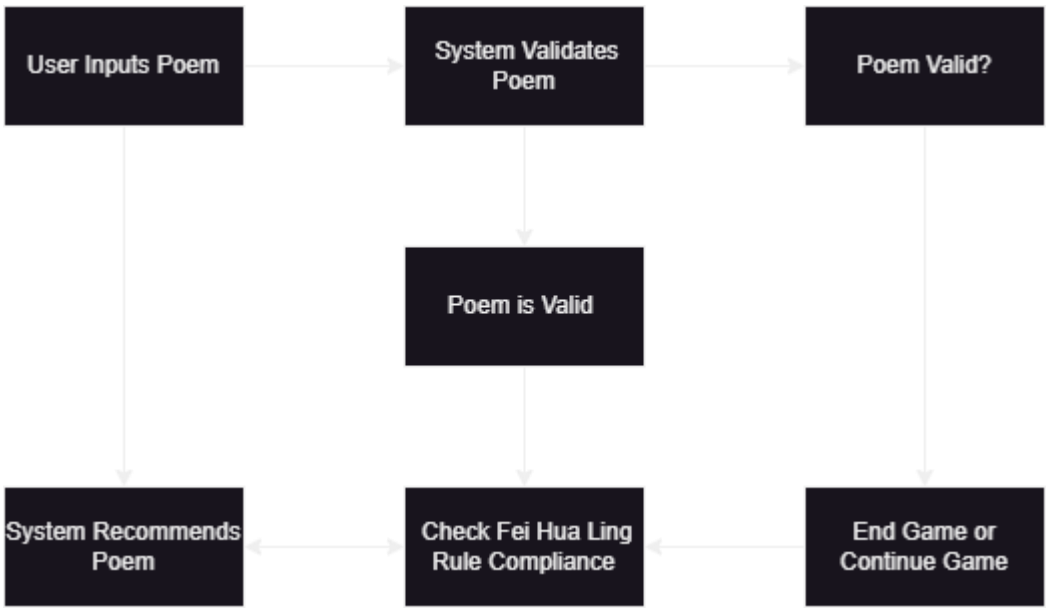
retrieval, updates, and management.

- **Data Access**:
  The back-end interacts with the database through an ORM framework. The system uses API interfaces to handle communication between the front-end and back-end, transmitting data in JSON format.

3. Key Function Module Design

3.1 *Fei Hua Ling* Game Module

- **Feature Description**:
  The *Fei Hua Ling* game module is the core component of the system. It is responsible for handling game logic, user input, generating poems that meet *Fei Hua Ling* rules, validating user input, and determining the winner based on the rules.
- **Detailed Design**:
  1. **Game Initialization**: At the beginning of the game, the system generates a topic (a character or word), and players must respond based on that character or word.
  2. **Poem Generation**: After the player submits their input, the system uses the poem generation module to create a valid poem. The system will generate diverse responses to ensure the game remains engaging.
  3. **Input Validation**: The system validates whether the player's poem meets *Fei Hua Ling* rules, such as rhyme, tone, word pairing, etc. If the input is invalid, the system will prompt the player to re-enter the answer.
  4. **Game End**: When a player answers incorrectly three times in a row or the time runs out, the game ends. The system calculates the score based on the number of correct answers and displays the player's historical performance.
- **Module Flowchart**:



3.2 Poem Generation Module

- **Feature Description**:
  The poem generation module utilizes natural language processing technology and pre-trained language models (like GPT or StarFire) to generate poems that comply with *Fei Hua Ling* rules. This module needs to generate diverse poem responses based on the player's input.
- **Detailed Design**:
  1. **Input Parsing**: The system first parses the player's input (character or word) and converts it into a format suitable for poem generation.
  2. **Poem Generation**: By calling an external API (such as GPT), the system generates poems that meet the rules of rhyme, parallelism, tone, etc.
  3. **Diverse Generation**: The system generates multiple alternative responses based on the input to offer a diverse experience.
  4. **Generation Optimization**: As users participate more, the system collects user choices and optimizes the poem generation quality and personalization.

3.3 Recommendation System Module

- **Feature Description**:
  The recommendation system analyzes user behavioral data and interests to recommend related poems, learning resources, questions, or websites. Based on player input and historical records, the recommendation system optimizes over time to enhance the accuracy of personalized recommendations.
- **Detailed Design**:
- 

  1. **Behavior Data Collection**: The system records user behavior in real-time, including submitted poems, answers, and other interactions.
  2. **Interest Analysis**: The system analyzes the user's behavior to identify their interests, such as preferred poetry styles or historical periods.
  3. **Recommendation Generation**: Based on the user's interests, the system selects relevant content from the poem database and recommends it to the player.
  4. **Recommendation Optimization**: As user behavior increases, the system uses machine learning algorithms to continually improve recommendation content, enhancing the user's game experience.

3.4 Data Flow Diagram

4. Performance and Security Design

4.1 Performance Design

- **Response Time Requirements**: The system should ensure that every user input returns a result within 1-3 seconds to maintain smooth gameplay.
- **Load Balancing**: To handle high concurrent requests, the back-end will employ load balancing techniques, such as Nginx or Docker, ensuring that the system operates smoothly under heavy load.
- **Caching Mechanism**: For frequently accessed data, the system will use caching technologies, such as Redis, to improve data response times, reducing computational overhead.

4.2 Security Design

- **Data Encryption and Protection**: User data (such as passwords and personal information) will be stored encrypted, and SSL/TLS protocols will be used for secure data transmission.
- **Access Control**: The system will implement strict access control mechanisms to ensure that only authorized users can access their personal information and game history.

5. Conclusion

This detailed design document thoroughly describes the system architecture, key functionality modules, database design, performance requirements, and security measures for the *Fei Hua Ling* game system. With modular design and optimized technological solutions, the system will efficiently and stably provide both gaming and educational services. As the project progresses, we will continue to optimize the quality of poem generation, enhance user experience, and improve the accuracy of the recommendation system to ensure a more rich and personalized poetry interaction for users.