

详细设计文档

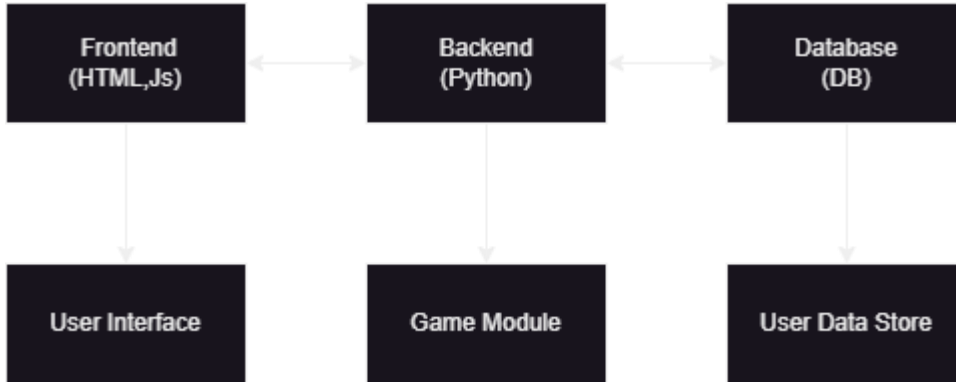
1. 系统架构设计

1.1 系统概述

本系统的核心目的是为用户提供一个富有趣味性和教育性的飞花令游戏体验。系统将通过前端和后端的紧密合作来实现游戏功能，采用模块化设计，确保系统具有较好的可扩展性和易维护性。系统由前端、后端、数据库三部分组成：

- 前端：**主要负责展示用户界面，接收用户输入，并将游戏反馈传递给用户。前端将采用现代化 Web 框架开发，以保证用户界面友好且具有响应性。
- 后端：**后端负责处理用户请求，控制游戏流程，调用诗词生成模块，进行用户数据存储等。后端将使用 Python 语言进行开发
- 数据库：**数据库主要用于存储用户信息、游戏历史记录、诗词数据等。数据库将采用 MySQL 或 PostgreSQL，以便于高效地管理结构化数据。

1.2 系统架构图



1.3 主要模块描述

- 前端模块：**
 - 前端负责提供用户界面(UI)和用户交互。通过 Node.js 框架，前端能快速响应用户的点击操作，进行输入输出交互。用户可以通过前端界面参与游戏、查看诗词解析和历史记录等。
- 后端模块：**
 - 后端主要通过 Web 框架来接收前端请求，进行数据处理，并将结果返回前

端。后端还负责调用诗词生成模块，验证用户输入，并更新数据库中的历史记录。

- 数据库模块
 - 数据库存储系统的所有数据，包括用户信息等。所有的数据都将通过后端与数据库进行交互，确保系统的数据安全性和持久性。
- 诗词生成模块
 - 诗词生成模块是本游戏的核心模块之一，基于预训练的语言模型（如 GPT 或其他诗词生成模型），生成符合飞花令规则的诗句。
- 推荐系统
 - 推荐系统分析玩家的历史行为，提供个性化的诗词、问题或学习资源推荐。通过推荐，玩家可以更加深入地了解中国古典诗词，并提高游戏体验。

2. 数据库设计

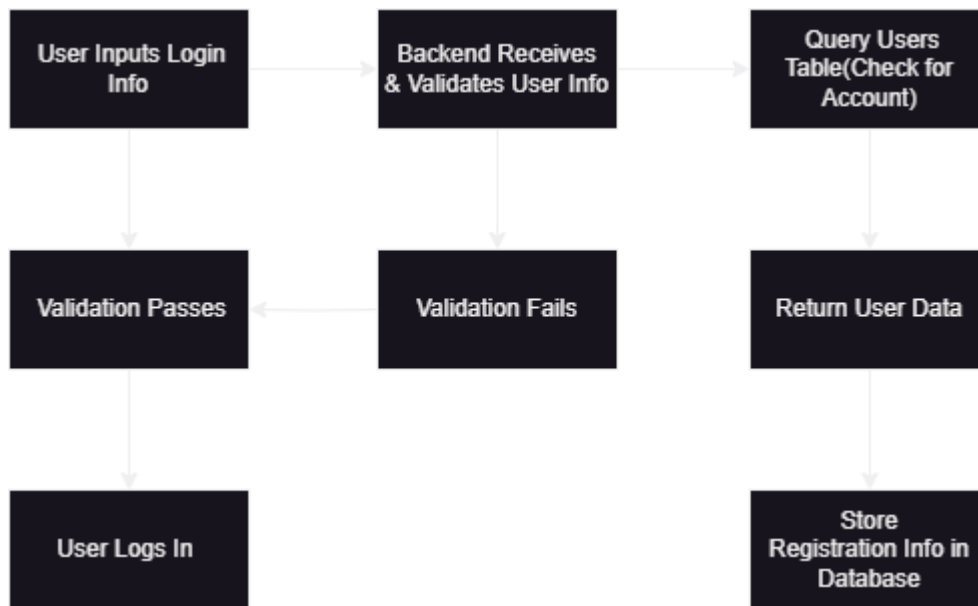
User Data
Name
Password
Email

2.1 数据库表结构

为了管理用户信息、游戏历史和推荐内容，我们设计了以下数据库表结构：

- 用户表 (Users): 存储用户的基本信息，包括用户名、密码、邮箱等。

2.2 数据存储与访问



- **数据存储:**
 - 所有用户信息数据都存储在数据库中。使用 SQL 数据库可以方便地进行数据检索、更新和管理。
- **数据访问:**
 - 后端通过 ORM 框架与数据库进行交互。系统通过 API 接口实现前端与后端的交互，数据通过 JSON 格式传输。

3. 关键功能模块设计

3.1 飞花令游戏模块

- **功能描述:** 飞花令游戏模块是本系统的核心部分，负责处理游戏的逻辑和用户的输入，生成符合飞花令规则的诗句，判断玩家输入的诗句是否合法，并根据规则判断胜负。
- **详细设计:**
 1. **游戏初始化:** 游戏开始时，系统会生成一个题目（一个字或词），玩家需要基于此字或词作答。
 2. **诗句生成:** 玩家输入后，系统通过诗词生成模块生成符合规则的诗句。系统会根据输入的字词生成多样化的响应，确保游戏不单一。
 3. **输入验证:** 系统会校验玩家输入的诗句是否符合飞花令规则，包括押韵、平仄、字词搭配等。如果输入不符合要求，系统会提示玩家错误并要求重新输入。

4. **游戏结束**：当玩家连续答错三次或时间用尽，游戏结束。系统根据答题情况计算得分，并展示玩家的历史成绩。

• **模块流程图：**



3.2 诗词生成模块

- **功能描述：** 诗词生成模块利用自然语言处理技术，通过训练好的大语言模型（如 GPT、星火）生成符合飞花令规则的诗句。该模块需要根据玩家输入的字或词生成多样化的响应诗句。
- **详细设计：**
 1. **输入解析：** 系统首先将用户输入的字或词进行解析，转化为适合生成诗句的格式。
 2. **诗词生成：** 通过调用外部 API（如 GPT 模型），系统根据输入的字词生成符合押韵、对仗、平仄等规则的诗句。
 3. **多样化生成：** 系统将根据输入生成多个备选答案，以便游戏能够提供多样化的体验。
 4. **优化生成：** 随着用户参与的增加，系统会收集用户的选择，优化诗句生成的质量和个性化程度。

3.3 推荐系统模块

- **功能描述：** 推荐系统根据用户的历史行为数据和兴趣，向用户推荐相关的诗词、学习资源、问题或网址。推荐系统能够根据玩家的输入和历史记录，逐步优化推荐

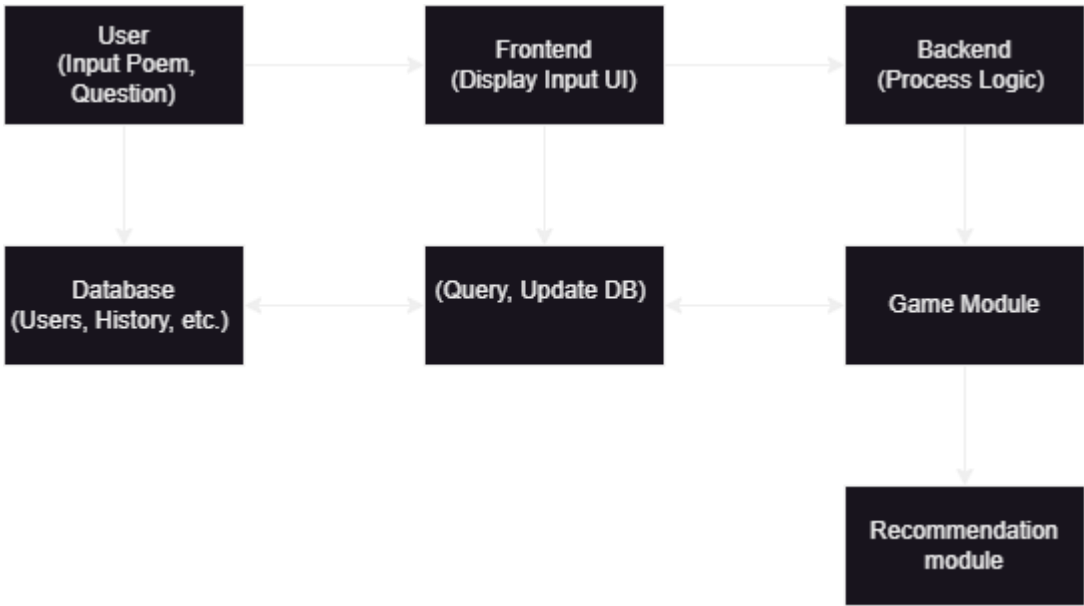
策略，提高个性化推荐的精准度。

• 详细设计：



1. **行为数据收集**：系统实时记录用户的行为数据，包括输入的诗句、答题情况等。
2. **兴趣分析**：系统会分析用户的行为，识别其兴趣点（如偏好的诗词风格、历史时期等）。
3. **推荐生成**：基于用户兴趣，系统将从诗词数据库中筛选出相关内容，推荐给玩家。
4. **推荐优化**：随着用户行为的增多，系统将使用机器学习算法不断优化推荐内容，提升用户的游戏体验。

3.4 数据流输送框图



4. 性能与安全设计

4.1 性能设计

- **响应时间要求：**系统应确保每次用户提交输入后，1-3 秒内返回结果，保持游戏流畅性。
- **负载均衡：**为适应高并发请求，后端将采用负载均衡技术，如 Nginx、Docker 等，确保系统在高负载下仍能平稳运行。
- **缓存机制：**对于频繁访问的数据，系统会通过缓存技术提高数据响应速度，例如使用 Redis 缓存诗词生成的结果，减少计算开销。

4.2 安全设计

- **数据加密与保护：**用户数据（如密码、个人信息）采用加密存储，传输过程中使用 SSL/TLS 协议确保数据安全。
- **权限控制：**系统将实现严格的权限控制机制，确保只有授权用户才能访问其个人信息和历史记录。

5. 结语

本详细设计文档详细描述了飞花令游戏系统的架构、功能模块、数据库设计、性能要求以及安全措施。通过模块化设计和优化的技术方案，系统能够高效、稳定地提供游戏和教育服务。随着项目的不断推进，我们将继续优化诗词生成质量、用户体验以及推荐系统的精准度，以确保为用户带来更加丰富、个性化的诗词互动体验。