**软件需求规格说明书**

# 诗词生成系统

(Software Requirements Specifications)

**April 10, 2024**

**Version 0.3**

**Prepared by**

**Name:**

**杨鑫月**

**胡馨月**

**曾诗涵**

**陈文洁**

目录

[诗词生成系统 1](#_Toc9842)

[1 介绍 3](#_Toc32099)

[1.1目的 3](#_Toc9383)

[1.2项目背景 3](#_Toc5378)

[1.3范围 3](#_Toc4650)

[1.4概述 3](#_Toc28158)

[1.5定义、缩略语和缩写词 3](#_Toc26318)

[2总体描述 4](#_Toc1255)

[2.1项目概述 4](#_Toc18606)

[2.2现有系统 4](#_Toc15382)

[2.3项目前景 6](#_Toc6875)

[2.4用户特征 6](#_Toc13315)

[2.5约束 6](#_Toc26323)

[3 功能需求 7](#_Toc14203)

[3.1用户登录 7](#_Toc7790)

[3.2用户注册 7](#_Toc18206)

[3.3用户输入 7](#_Toc4938)

[3.4基础诗词生成 7](#_Toc30836)

[3.5藏头诗生成 8](#_Toc29309)

[3.6自定义诗词生成 8](#_Toc29280)

[3.7结果显示 8](#_Toc26782)

[4 非功能需求 8](#_Toc16983)

[4.1性能要求 8](#_Toc23478)

[4.2 可用性需求 8](#_Toc27640)

[4.3 安全性需求 9](#_Toc28481)

[4.4 可维护性和可扩展性需求 9](#_Toc11690)

[4.5用户界面要求 9](#_Toc11346)

[5 验收标准及测试 9](#_Toc2724)

[5.1验收标准 9](#_Toc2297)

[5.2测试 9](#_Toc2442)

[6 风险评估及应对措施 10](#_Toc32474)

[6.1技术风险 10](#_Toc15568)

[6.2用户接受度风险 10](#_Toc24689)

# 1 介绍

**1.1目的**

本需求文档详细介绍诗词生成系统的功能和特征，以便顾客能够确认对产品的确切需求，开发人员能够根据需求设计编码。下文将结合文字描述、数据流图和ER图等来描述诗词生成系统的功能、性能、用户界面、运行环境、外部接口以及针对用户操作给出的各种响应。本文档的预期读者有需求分析人员、设计人员、开发人员、项目管理人员、测试人员和用户。

**1.2项目背景**

近年来，随着人工智能技术，尤其是深度学习领域的飞速发展，自然语言处理（NLP）技术取得了前所未有的进步。诗词，作为中华文化的瑰宝，一直以来都是文人墨客表达情感、抒发怀抱的重要方式。然而，传统的诗词创作需要深厚的文化底蕴和长时间的磨练，对于普通人来说，创作出一首合格的诗词并非易事。

在这样的背景下，利用人工智能技术来辅助或自动生成诗词成为了一个热门的研究方向。诗词生成系统应运而生，它不仅能够降低诗词创作的门槛，让更多人体验到创作的乐趣，还能在一定程度上传承和弘扬中华传统文化。诗词生成系统作为一个近年来发展迅速的研究领域，随着深度学习技术的成熟，已经取得了一些令人瞩目的成果。

**1.3范围**

本系统是基于深度学习技术的诗词生成系统，具有基础诗词生成、藏头诗生成、自定义风格与形式等功能。系统采用前后端分离架构，后端使用Flask框架，前端使用Vue.js框架。该系统会向用户提供以下功能：

·用户登陆

·用户注册

·藏头诗生成

·自定义风格诗词生成

·用户信息管理

·查看历史记录

**1.4概述**

本文档以下内容从总体描述和具体要求两个部分描述满足所有系统需求的规范。总体描述提供了关于系统要求的背景资料，以及影响系统设计、体系结构和要求的一般因素；具体要求部分通过使用软件工程技术和工具，详细、准确地给出了行为、交互和体系结构方面的系统规范，以便设计人员能够设计出符合所有需求的系统，测试人员可以起草测试计划，从而检查系统是否实现所有功能和特性，符合所有要求。

**1.5定义、缩略语和缩写词**

无。

# 2总体描述

**2.1项目概述**

本项目旨在构建一个功能全面、用户友好的智能诗词生成系统。该系统基于深度学习技术，能够生成具有格律规范性、语言流畅性、创造性和想象力的诗歌。系统集成了基础诗词生成、藏头诗生成以及自定义风格与形式等多项功能，满足了用户在不同场景下的创作需求。

在技术上，本系统采用了前后端分离的项目架构，后端基于Flask框架提供API服务，前端则使用Vue.js框架构建用户界面。这样的架构设计使得系统具有良好的可扩展性和可维护性，同时也为用户提供了流畅、直观的操作体验。

**2.2现有系统**

目前市场上已经存在多个智能诗词生成系统，以下是对其中几个系统的简要介绍和与本项目的比较：

·九歌系统：

介绍：由清华大学开发，基于深度学习技术，能够生成不同风格的古诗词。

示例：



图1 九歌系统生成七言绝句《中秋》示例

比较：与本项目相似，都采用了深度学习技术。但本项目在用户自定义风格和形式方面可能更为灵活。

·乐府系统：

介绍：由华为诺亚方舟实验室开发，同样基于深度学习，注重风格多样性。

示例：



图2 乐府系统生成七言绝句《中秋》示例

比较：乐府可能在风格多样性上有所突破，而本项目在用户体验和界面设计上可能更为出色。

·AI作诗：

介绍：由诗三百团队开发，主要基于规则的方法来生成诗词。

示例：



图3 AI作诗生成七言绝句《中秋》示例

比较：与本项目不同，AI作诗更注重规则的制定，而本项目则更多地依赖于深度学习模型，两者在生成诗词的方法和效果上有所不同。

**2.3项目前景**

随着人工智能技术的不断进步和普及，以及人们对传统文化兴趣的日益浓厚，智能诗词生成系统有着广阔的市场前景和应用空间。

1.教育领域：系统可以作为教学辅助工具，帮助学生更好地理解诗词的结构和韵律，提升对传统文化的认识和兴趣。

2.文化创意产业：生成的诗词可以用于文化创意产品的开发，如诗词主题的文创产品、贺卡、装饰品等，丰富文化市场。

3.个性化服务：为用户提供个性化的诗词创作服务，如定制化的藏头诗、节日祝福诗等，满足用户的个性化需求。

4.技术研究与发展：项目的实施和运营将为深度学习、自然语言处理等技术在诗词生成领域的应用提供宝贵的实践经验和数据支持，推动相关技术的进一步发展。

考虑到目前市场上已经存在多个智能诗词生成系统，本项目需要在用户体验、诗词质量、风格多样性等方面不断创新和提升，以在竞争中脱颖而出。同时，随着技术的不断进步和用户需求的变化，本项目还有望在更多领域得到应用和推广。

**2.4用户特征**

系统的主要用户是对诗词感兴趣的人群，包括学生、诗词爱好者、文化艺术从业者等。

**2.5约束**

系统运行的环境需求如下：

服务器端：

操作系统：Linux

Python 3.7+

Flask框架

深度学习框架（如TensorFlow、PyTorch等）

客户端：

现代浏览器（Chrome、Firefox、Safari等）。

# 3 功能需求

**3.1用户登录**

·功能描述：用户可以通过此功能登录系统，以便使用个性化服务和保存个人设置。

·输入：用户邮箱和密码。

·处理：验证用户邮箱和密码是否匹配数据库中存储的信息。

·输出：登录成功则进入系统主界面，失败则提示错误信息。

·安全性：密码传输需加密，登录失败次数过多应暂时锁定账户。

**3.2用户注册**

·功能描述：新用户可以通过此功能创建账户。

·输入：用户名、用户邮箱、密码。

·处理：检查输入信息的合法性（密码合理性、用户名是否已存在），并将信息存储到数据库。

·输出：注册成功提示或错误信息。

**3.3藏头诗生成**

·功能描述：根据用户提供的若干文字，并选择诗词格式，生成一首藏头诗。

·输入：用户输入的文字内容。

·处理：系统需确保生成的每一句诗的首字与用户输入相匹配。

·输出：一首藏头诗，每句诗的首字组成用户输入的序列。

**3.4智能诗词生成**

·功能描述：用户可以选择特定的诗词合适，并输入关键词进行生成。

·输入：用户选择的诗词格式和输入关键词。

·处理：系统根据用户选择的风格调整模型参数或选择相应的训练数据。

·输出：一首符合用户指定风格的诗词。

**3.5用户信息管理**

·功能描述：用户可以修改用户名，也可以在通过验证后修改密码。

·输入：修改后的用户名或密码。

·处理：系统会验证修改的密码或者用户名是否合法。

·输出：修改成功或失败的提示。

**3.6查看历史记录**

·功能描述：用户可以查看之前生成的结果记录。

·输入：无。

·处理：无。

·输出：显示该用户的生成记录。

# 4 非功能需求

**4.1性能要求**

·响应时间：系统应在用户提交输入后的极短时间内（例如5秒内）生成诗词，确保用户体验的流畅性。

·准确性：生成的诗词应保证质量，符合语言规范，且与用户输入的主题或关键词高度相关。

·系统稳定性：在高并发场景下，系统应保持稳定运行，不出现崩溃或性能大幅下降的情况。

**4.2 可用性需求**

·用户界面：系统界面设计应简洁明了，布局合理，色彩搭配舒适，确保用户能够轻松找到所需功能。

·易操作性：系统功能操作应简单直观，减少用户学习成本。对于复杂操作或高级功能，应提供必要的提示和引导。

·用户指南和帮助文档：系统应提供详尽清晰的用户指南和帮助文档，包括功能介绍、操作步骤、常见问题解答等，以便用户快速上手并解决使用中遇到的问题。

**4.3 安全性需求**

·数据保护：系统应采取加密措施保护用户数据，防止数据在传输和存储过程中被窃取或篡改。

·访问控制：系统应实施严格的访问控制策略，确保只有授权用户才能访问敏感数据和执行关键操作。

·防止恶意攻击：系统应具备防御常见网络攻击（如SQL注入、跨站脚本攻击等）的能力，确保系统的安全性和稳定性。

**4.4 可维护性和可扩展性需求**

·模块化设计：系统应采用模块化设计思想，将不同功能划分为独立的模块，便于后续的维护和升级。

·可扩展性：系统应预留扩展接口和扩展点，以便在需要时能够轻松地添加新功能或集成其他系统。

·日志记录与监控：系统应建立完善的日志记录和监控机制，便于追踪问题、定位故障和优化性能。

·兼容性：系统应兼容多种主流浏览器和设备，确保用户在不同平台上都能获得良好的使用体验。

**4.5用户界面要求**

·用户界面应具有良好的响应性，适应不同屏幕尺寸和设备。

·用户界面应具备美观的设计和良好的用户体验。

# 5 验收标准及测试

**5.1验收标准**

系统应满足以下标准方可视为验收合格：

·系统能够准确根据用户输入生成诗词；

·生成的诗词应符合韵律和平仄要求；

·系统性能稳定，响应时间快；

·用户界面友好，易于操作。

**5.2测试**

系统应通过以下测试：

·功能测试：测试系统各项功能是否正常运行；

·性能测试：测试系统的响应时间和稳定性；

·可用性测试：测试系统是否易于上手和使用；

·安全性测试：测试系统的安全性和隐私保护能力。

# 6 风险评估及应对措施

**6.1技术风险**

风险：seq2seq模型和注意力机制的实现难度较大，可能影响项目进度。

应对措施：提前进行技术预研，加强团队间的沟通与协作，确保技术难题得到及时解决。

**6.2用户接受度风险**

风险：用户可能对生成的诗词质量和韵律满意度不高。

应对措施：收集用户反馈，持续优化模型算法和韵律规则，提高用户满意度。