



# 西南科技大学

Southwest University of Science and Technology

## 本科毕业设计（论文）

题目名称：酒诗文化数据治理与可视化系统

设计与实现

学院名称	计算机科学与技术
专业名称	计算机科学与技术
学生姓名	李 胜
学 号	5120216892
指导教师	王桂娟（讲师）

二〇二五年五月

# 西南科技大学

## 本科毕业设计（论文）学术诚信声明

本人郑重声明：所呈交的毕业设计（论文），是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名：李杜

日期：2025 年 6 月 5 日

# 西南科技大学

## 本科毕业设计（论文）版权使用授权书


本毕业设计（论文）作者同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权西南科技大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本毕业设计（论文）。

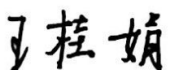
保密☐，在\_\_\_年解密后适用本授权书。

本论文属于

不保密☐.

（请在以上方框内打“√”）

作者签名：

指导教师签名：

日期：2025 年 6 月 5 日

日期：2025 年 6 月 5 日

# 酒诗文化数据治理与可视化系统设计与实现

**摘要：**中华传统酒诗文化源远流长，蕴含着丰富的历史、文化与艺术价值，是研究古代社会风貌与文人精神世界的重要窗口。然而，酒诗文化数据分散、格式多样且存在大量噪声，数据治理难度大，同时如何将复杂的数据以直观易懂的方式呈现，也是当前面临的一大挑战，本文设计并且实现了一套针对中华传统酒诗文化的数据治理以及可视化系统。此系统运用分层架构来设计，前端是基于 Vue 3 框架并结合 TypeScript 语言进行开发，还集成了 ECharts 可视化组件，后端运用 Node.js 与 Express.js 技术栈搭建服务器环境，同时配合 MySQL 数据库以及 AI 分析模块。在数据治理方面，系统构建了覆盖数据采集、标准化以及质量评估的全流程治理机制，切实保障了数据的一致性与准确性，还开发了多样化的可视化交互平台。在可视化方面，通过直观的图表和图形展示，方便用户理解和分析中华传统酒诗文化数据，为研究者和爱好者提供了一个高效、便捷且功能全面的平台，有助于深入挖掘中华传统酒诗文化的内涵与价值。

**关键词：**酒诗文化数据；数据治理系统；可视化系统；人工智能分析

# Design and Implementation of a Data Governance and Visualization System for Wine Poetry Culture

**Abstract :** The traditional Chinese wine poetry culture has a long history and contains rich historical, cultural and artistic values, and is an important window for the study of ancient social features and the spiritual world of literati. However, the data of wine poetry culture is scattered, diverse in format and has a lot of noise, which makes it difficult to govern the data, and how to present the complex data in an intuitive and easy-to-understand way is also a major challenge at present. The system is designed using a hierarchical architecture, the front-end is based on the Vue 3 framework and combined with the TypeScript language, and also integrates the ECharts visualization component, and the back-end uses Node.js and Express.js technology stack to build a server environment, and cooperates with MySQL database and AI analysis module. In terms of data governance, the system has built a whole-process governance mechanism covering data collection, standardization and quality assessment, effectively ensuring the consistency and accuracy of data, and has also developed a diversified visual interaction platform. In terms of visualization ,through intuitive charts and graphs, it is convenient for users to understand and analyze the data of traditional Chinese wine poetry culture, providing an efficient, convenient and comprehensive platform for researchers and enthusiasts, which is helpful to deeply explore the connotation and value of traditional Chinese wine poetry culture.

**Key words:** Wine and Poetry Culture; Data Governance; Visualization System; Artificial Intelligence Analysis

# 目 录

第 1 章 绪论 .....	1
1.1 研究背景 .....	1
1.2 研究意义 .....	2
1.3 国内外现状 .....	2
1.3.1 国内现状 .....	2
1.3.2 国外现状 .....	4
1.4 研究目标 .....	4
1.5 本章小结 .....	5
第 2 章 诗文化数据治理与可视化系统分析 .....	6
2.1 系统介绍 .....	6
2.1.1 系统定位与目标 .....	6
2.1.2 系统分层架构 .....	6
2.2 功能需求 .....	7
2.2.1 业务流程 .....	7
2.2.2 系统用例及关系图 .....	10
2.3 数据需求 .....	10
2.3.1 数据来源 .....	10
2.3.2 数据规模 .....	11
2.3.3 储存需求 .....	11
2.3.4 数据预处理需求 .....	12
2.4 性能需求 .....	12
2.5 系统用例分析 .....	13
2.5.1 酒诗文化数据治理与可视化系统的主要用例 .....	13
2.5.2 用例优先级矩阵 .....	14
2.6 本章小结 .....	15
第 3 章 诗酒数据治理方法设计与实现 .....	17
3.1 数据架构设计 .....	17

3.2 数据质量控制 .....	18
3.2.1 数据验证触发器 .....	18
3.2.2 数据一致性检查存储过程 .....	19
3.3 AI 自动评分 .....	20
3.3.1 用户交互设计 .....	21
3.3.2 实际应用效果 .....	21
3.4 性能优化 .....	21
3.4.1 索引优化 .....	21
3.4.2 分区策略 .....	22
3.4.3 查询优化 .....	22
3.5 本章总结 .....	23
<b>第 4 章 诗酒数据治理与可视分析系统详细设计与实现 .....</b>	<b>24</b>
4.1 技术栈详细设计 .....	24
4.2 数据流转流程 .....	24
4.3 系统界面介绍 .....	25
4.3.1 主界面介绍 .....	25
4.3.2 数据质量管理界面 .....	26
4.3.3 标准层管理界面 .....	29
4.3.4 数据可视化界面 .....	30
4.4 可视化实现方式 .....	31
4.5 系统功能介绍 .....	32
4.5.1 元数据列表的操作 .....	32
4.5.2 字段列表展示 .....	32
4.5.3 添加/编辑字段对话框 .....	33
4.5.4 数据同步 .....	34
4.5.5 生命周期钩子 .....	35
4.5.6 其他功能 .....	36
4.6 本章总结 .....	37
<b>第 5 章 诗酒数据可视分析与数据治理系统测试 .....</b>	<b>39</b>
5.1 功能性测试 .....	39

5.1.1 后端功能测试 .....	39
5.1.2 前端功能测试 .....	42
5.2 性能测试 .....	45
5.3 本章总结 .....	46
结论 .....	47
致谢 .....	48
参考文献 .....	49



## 第1章 绪论

### 1.1 研究背景

酒文化身为中国传统文化体系里的核心部分，与诗歌艺术于中华文明的漫长历史进程中一直维系着紧密的共生关联，唐宋时期，酒这个物质载体深入渗透到社会生活的各个方面，还在诗歌创作里转变成有丰富象征含义的美学意象，变为诗人用来抒发豪迈气魄、孤寂心绪、真挚情谊、离别哀愁以及生命感悟等多种情感的艺术手段。像“举杯邀明月”“莫使金樽空对月”等经典诗句，经过千年文化的积淀依旧保持着生命力，已然成为中华民族集体文化记忆里的关键符号，这些把酒写进诗里的文学作品以及与之相关的多维度信息，呈现出碎片化分布的特点，广泛散布在古籍文献、数字资源库、学术论著以及图像资料等异构化载体当中<sup>[1]</sup>。

在人工智能和数据科学快速发展的时代环境中，传统文化资源的数字化整理以及学术研究有了很大的进步，就拿高校和研究机构建立的古典诗歌数字资源库来说，这些平台为研究诗歌文本的语言特点和情感表达提供了关键基础，自然语言处理等人工智能技术被引入后，诗歌文本的自动化解析能力得到了提升，涉及语义关联分析、情感识别、智能创作以及跨语言转换等关键部分，推动了传统文学和现代科技的交叉融合<sup>[3][4][5]</sup>。凭借数字可视化呈现、元数据架构设计以及多模态情感计算等前沿技术方法，研究者在整合古代酒诗包含的文本、语境和图像等多维度信息，这为中华诗酒文化的系统性展示提供了创新方式，也为其当代价值重构和国际传播注入了新的活力<sup>[6][7]</sup>。

当前对于酒诗文化资源所开展的系统性整理工作存在着比较十分突出的欠缺，在跨模态资源整合以及标准化数据治理方面遭遇到了诸多困境，具体呈现为数据规范有所缺失、标注体系较为混乱以及异构数据难以达成互联互通等情况，这些技术方面的瓶颈对学术研究的深入进行以及文化保护工作的有效推进形成了制约，而且对中华文化在全球范围内的传播范围与认知程度产生了负面作用。这一现状为中华优秀传统文化的创造性转化与创新性发展提供了新的研究机会，也为“中华文化走出去”战略的实施开拓了潜在的发展空间<sup>[2]</sup>，利用数字人文技术与人工智能手段来达成诗酒文化资源的系统整合、标准化处理以及多语言跨文化传播<sup>[6]</sup>，会成为未来学术界和产业界急需解决的关键研究课题。

## 1.2 研究意义

本研究构建了一个综合性平台，融合信息系统、数据库技术、人工智能以及 Web 可视化等多学科方法，实现中华酒诗文化数据的系统化治理与可视化呈现，促进该文化遗产的数字化传承以及学术价值挖掘，其核心价值有三个维度：其一，运用结构化数据处理与智能分析技术，对传统酒诗文化资源进行高质量治理，保证文化数据有检索便捷、管理规范、分析科学以及应用可行等特点，其二，引入情感计算与自动评分等 AI 技术，建立人文学科与人工智能的跨学科研究范式，给诗歌分析方法论提供创新性技术支持，其三，采用多层次可视化技术体系，达成复杂酒诗数据的多维度知识表征与传播，有效提高文化资源的公众认知度与应用转化效率。

## 1.3 国内外现状

随着学界针对中华优秀传统文化研究程度的持续加深，古典诗歌身为关键的研究对象变得日益受到广泛的关注，研究视角从早期对诗词格律以及语言艺术的表层赏析，逐渐延伸至对其历史文化内涵的深入解读，这种转变体现出学术界和社会大众对于诗歌所蕴藏的人文精神与思想价值的认知有所提高。在当代社会快速变化的背景之下，古典诗词凭借其独特的精神慰藉功能和文化认同价值，正变成人们寻觅心灵栖息地的关键文化资源 [8]。

当下学术界和产业界正渐渐突破传统研究范式的限制，积极推动跨学科融合创新，诗歌研究领域已经出现了一系列带有高效检索功能的技术平台，这些系统整合了知识图谱、自然语言处理技术以及可视化呈现方式，达成了对海量诗歌数据的快速查询与智能筛选，这种构建诗歌领域数字化知识体系的研究路径，已经发展成数字人文研究的关键趋势。

### 1.3.1 国内现状

近十年间我国在数字人文领域古典诗词数据库建设方面有了较大进展，像中国知网、爱诗网以及各地数字方志等平台，逐渐搭建起了相对完备的古典诗词基础数据服务体系，给学术研究、教学实践以及公众文化需求给予了方便的数据检索途径，这些平台重点关注诗词文本的数字化采集、基础检索功能达成以及可视化呈现，对传统诗词文化的数字

化保存与传播起到了关键作用<sup>[9]</sup>。然而随着学科交叉研究的不断发展以及创新应用需求的持续增加，现有诗词数据库在专题分类体系构建、数据治理标准统一、深层数据结构化处理、智能语义解析以及开放可视化交互等方面存在着一定的局限性：

当前已有的研究显示，针对以酒文化作为主题的诗词亚类专题数据库展开的建设工作，至今依旧存在着十分突出的欠缺之处，在相关领域范围内，一直都缺少那种有权威性以及系统性的专业资源库，而这样的一种现状，对该领域朝着纵深方向的研究以及创新成果的呈现，都造成了颇为严重的妨碍<sup>[10]</sup>。

目前主流数据集中普遍存在着标注粒度较为粗糙、分类标准不太统一的情况，在题材分类、情感倾向、地域特征以及人物属性等关键维度方面，数据标注要么严重缺失，要么处于混乱状态，这样的结构性缺陷极大地限制了语义解析的精确程度以及多维度数据挖掘的潜在价值<sup>[11]</sup>。

当前的相关研究显示，借助情感分析、领域 AI 评分以及诗品自动评价等智能化处理手段，目前尚处在初步的探索进程以及实验验证环节，其核心存在的局限主要体现在两个方面，其一尚未创建出可适配古典诗歌文体特征的专门优化模型，其二缺少经由大规模人工标注所形成的高质量数据集，以此作为算法训练的基础<sup>[12]</sup>。

当前现有的数据库以及平台系统在可视化呈现这一方面，存在着形式较为单一的问题，并且其交互功能的设计以及场景应用的多样性，都以及待提高，这样的状况导致其难以对科研分析、教育教学以及文化旅游等不同领域用户的专业化需求，提供有效的支撑<sup>[13]</sup>。

当前各平台在数据版权管理、开放开发后续环节以及学术成果应用转化等关键领域，还没有构建起系统化的资源共享机制以及高效的成果转化体系，这种状况对资源的可持续开发利用以及创新驱动发展形成了一定制约<sup>[14]</sup>。

当下诗词数字资源建设遭遇不少挑战，像数据收录比较分散、标注标准未统一以及跨平台兼容性欠缺等情况，这些状况极大限制了知识图谱构建与跨模态融合等前沿技术的实际运用，在酒诗等特定领域体现得更为突出，很有必要促使诗词数据库建设达成从规模积累到质量提升的范式转变，借助实施精细化标注、深度结构化处理、智能化情感分析以及多元化可视化呈现等关键技术，来应对数字人文研究与中华诗词文化现代化传承的急切需求<sup>[12][14]</sup>。

### 1.3.2 国外现状

国际学术界在古典文学数据化工程这个领域，发展历程是比较久的，已经形成了相对成熟的数字人文实践体系，像北美地区的“Perseus 数字图书馆”以及欧陆的“Europeana”计划，是比较典型的代表，这些规模很大的数字化平台，整合了语义网络技术、海量数据处理以及数字遗产保护等前沿科技手段，实现了对古典文学典籍、图像资料以及元数据信息的系统性数字化管理，还促进了文献资源的跨语言传播、多模态呈现以及跨学科应用<sup>[15][16]</sup>。展开来说，Perseus 项目依靠其深度语义标注功能和开放式数据接口，给古希腊罗马典籍研究提供了关键的技术支撑和数据资源，而 Europeana 项目主要关注欧洲文化遗产的语义关联和数字化整合，借助联合图书馆、博物馆以及档案馆等多种类型文化机构，构建了统一的开放式知识服务生态系统<sup>[15][16][17]</sup>。

当前的相关研究显示，西方数字平台在处理中国古典诗歌酒诗这类有特色的题材之时，遭遇了文化语境存在差异、技术标准无法兼容以及应用场景有特殊性等多方面的限制，深入剖析其背后原因可知，当下的知识表示模型一方面没能充分呈现出中国诗歌特有的意象系统、情感表达以及礼俗典故之间的深层联系，另一方面缺少针对中文文本复杂性的适应性处理机制，这其中涉及了词法结构、语法特征以及意向表达的独特之处<sup>[18][19]</sup>。展开来说，这些平台还没有构建起可系统解析酒诗文化里宴饮礼仪、历史地理信息以及人物关系等关键要素的数据模型，致使难以达成对中国传统“诗酒风流”美学意境及其跨领域知识关联的数字化呈现与创新应用。

## 1.4 研究目标

本研究构建一套针对诗酒文化数据的可视化分析及治理系统，最关键的是搭建标准化酒诗元数据框架，对诗词作品、创作者以及时代背景等多方面信息加以规范化治理，系统运用自动化采集与人工审核相联合的多层级数据处理机制，还引入基于深度学习的情感分析算法与质量评估模块，提升了数据的判别精度和应用效能。基于此，系统借助多样化可视化呈现方式和交互功能，为学术研究者与普通用户给予直观的数据洞察与决策支持。

## 1.5 本章小结

在这一章节之中，对中华酒诗文化数字化研究的理论框架进行了系统性的构建，围绕研究背景、学术价值、国内外研究进展以及本文创新贡献这四个维度展开详细论述，一开始，从文化遗产的角度出发，阐释了酒诗作为中华优秀传统文化关键载体所有的独特价值，并且对数字化技术给这一文化遗产带来的转型机遇以及技术挑战展开了深入剖析。依据跨学科研究范式，明确了本研究借助数据治理与可视化技术达成酒诗文化数字化保护与创新应用的双重目标，在对国内外研究现状进行分析时，运用文献计量方法指出当前古典诗词数据库建设存在的结构性缺陷：国内在酒诗专题数据库领域仍然存在研究空白，而国际现有的平台普遍存在中文诗歌文化语境适配性不足的问题。本研究提出了包含诗酒数据可视化分析体系以及智能化治理框架在内的系统性解决方案，其创新性主要体现在三个方面：构建了端到端的酒诗数据治理工作流，开发了基于深度学习的情感分析集成系统，为中华酒诗文化的数字化保护与当代转化提供了创新性的方法论支撑。

## 第 2 章 诗文化数据治理与可视化系统分析

### 2.1 系统介绍

#### 2.1.1 系统定位与目标

本研究构建了酒诗文化数据治理与可视化集成平台，可达成数据全生命周期管理、人工智能辅助分析以及多元化可视化呈现的有机融合，该平台遵循“数据质量优化、安全可靠保障、智能治理实施、高效发布机制、知识协同共享”的核心原则，针对学术研究者、文化传播从业者、技术开发人员以及政策制定者等多元化用户群体，提供有精准性、智能化与交互性的酒诗文化数字化解决办法。

#### 2.1.2 系统分层架构

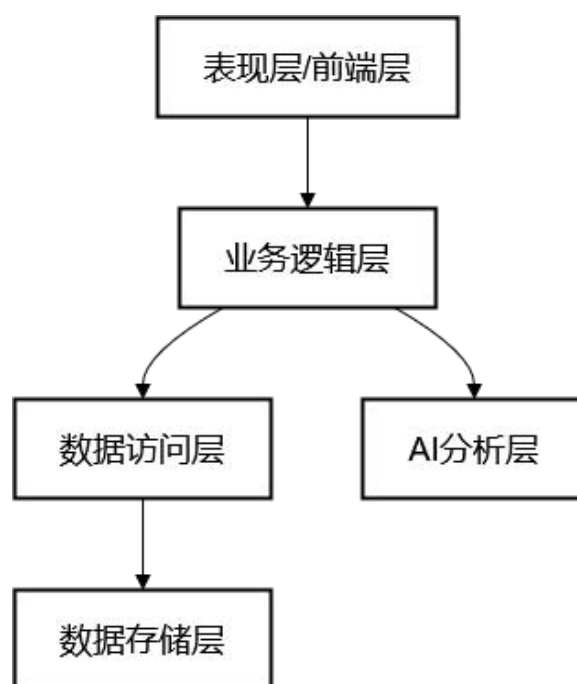


图 2-1 系统分层架构

本研究打造的诗词管理与分析系统运用分层架构设计思路，严格依照模块化和解耦原则来构建，系统架构囊括表现层、业务逻辑层、数据访问层、数据存储层以及 AI/数据分析层，形成了层次清晰的五层结构，表现层借助 Vue 3 框架得以实现，运用组件化开发模式来提高代码复用率，整合 Vuex 状态管理机制以保证数据共享，借助 Vue Router 达成页面路由控制。核心业务逻辑层着重处理诗词管理、诗人管理以及质量分析等核心

业务功能，凭借标准化业务接口规范业务流程，引入事务管理机制保障数据一致性，数据访问层封装了全部数据库操作，采用连接池技术优化资源利用，实施参数化查询防范 SQL 注入攻击，建立多级缓存体系提升数据访问效率。底层数据存储基于 MySQL 关系型数据库，凭借规范化设计保证数据结构合理，利用主外键约束维护数据完整性，建立索引机制优化查询性能，独立的 AI/数据分析层专注于诗词质量评估和文本分析功能，采用模块化架构支持算法扩展，结合异步处理与结果缓存提升性能表现，系统架构设计有这些特征：层间通信采用标准化接口协议，实现高内聚低耦合，建立自上而下的单向数据流机制，保证数据处理过程可追溯，实施分层安全防护体系，覆盖接口权限管控、数据访问审计等多重防护措施，凭借多级缓存策略、数据库优化及负载均衡技术保障系统性能，采用插件化设计支持功能扩展。在部署架构方面，系统采用前后端分离模式，前端由 Nginx 服务器承载，后端基于 Node.js 应用服务器，数据存储依靠 MySQL 数据库，并运用容器化技术实现环境隔离与自动化部署，这种分层架构设计明确了各层的职责边界，凭借标准化接口实现高效协作，在可维护性、可扩展性、可测试性等方面有优势，最终构建了一个功能完备、性能卓越且安全可靠的诗词管理与分析平台。

## 2.2 功能需求

本研究构建的系统达成了酒诗文化数据全生命周期管理机制，其核心功能模块有以下内容：运用标准化流程针对诗词文本、诗人信息以及相关多媒体资料开展结构化元数据建模并进行动态维护，借助人工智能技术达成数据质量智能评估体系，融合情感计算模型与自动评分算法，生成定制化的数据质量分析报告，在检索功能方面，系统支持多维度组合查询与全文检索技术，可动态构建诗歌、诗人以及酒文化实体间的语义关联网络与知识图谱，依靠集成 ECharts 等可视化工具，系统实现了涉及时空分布特征、流派演变趋势、情感标签聚类等多维度交互式数据可视化分析功能。

### 2.2.1 业务流程

本系统的业务流程主要围绕数据采集、处理分析、质量评估和可视化展示四个核心环节展开。

本研究搭建的数据采集系统运用多源异构数据获取机制，设有基于 Web 交互界面的用户手动录入功能，可支持单条诗词数据的增添、删除、修改以及查询操作，还达成了

符合行业标准的 API 接口协议，可和外部数据源进行无间隙对接，在数据预处理阶段，系统施行严谨的质量控制流程，先后开展数据格式校验、异常数据清洗以及标准化转换等关键步骤，以此在数据入库前切实保障其完整性与一致性。

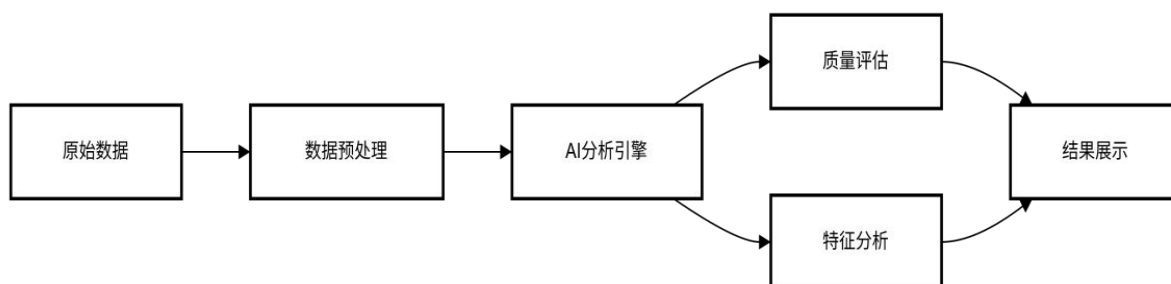


图 2-2 数据采集业务流程

完成数据预处理工作之后，系统会自动开启多维度评估分析机制，先是分析引擎针对诗词文本开展多层次特征抽取工作，其中涉及文学表现手法、语言风格特征以及文化内涵要素等关键维度，接着系统会调用集成的智能分析模块来进行语义深度解析，最终输出一份综合报告，里面包含质量评估指标以及结构化分析结果。整个分析流程借助异步任务队列架构达成自动化处理，以此保障系统有高并发响应性能。

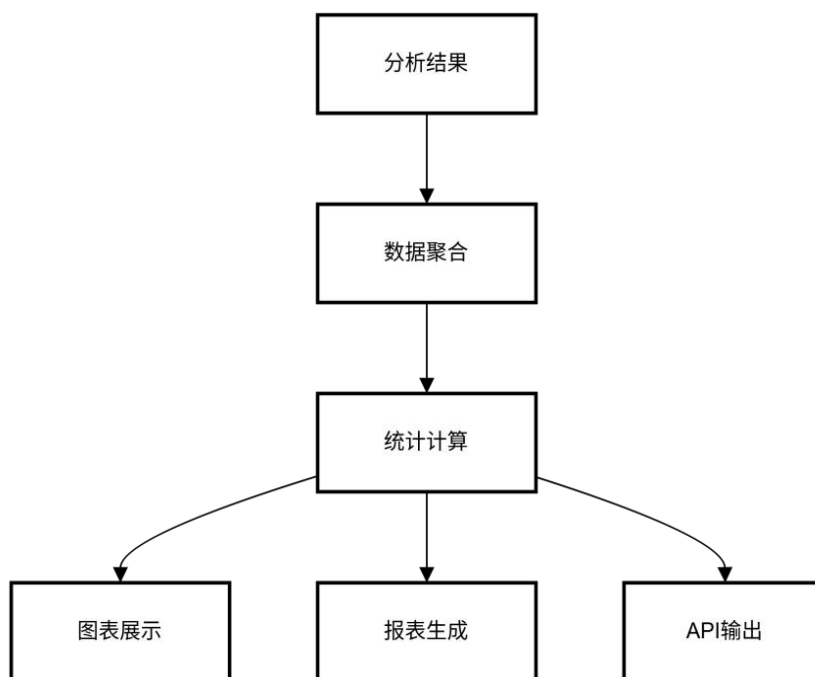


图 2-3 分析评估业务流程



本研究运用 ECharts 技术框架达成了分析结果的多维度可视化呈现，具体包含诗词数量统计、质量分布以及朝代分布等关键维度的数据展示，系统设计还整合了完善的数据导出功能，并且提供了标准化的 API 接口，用以契合跨系统数据交互与集成的要求。

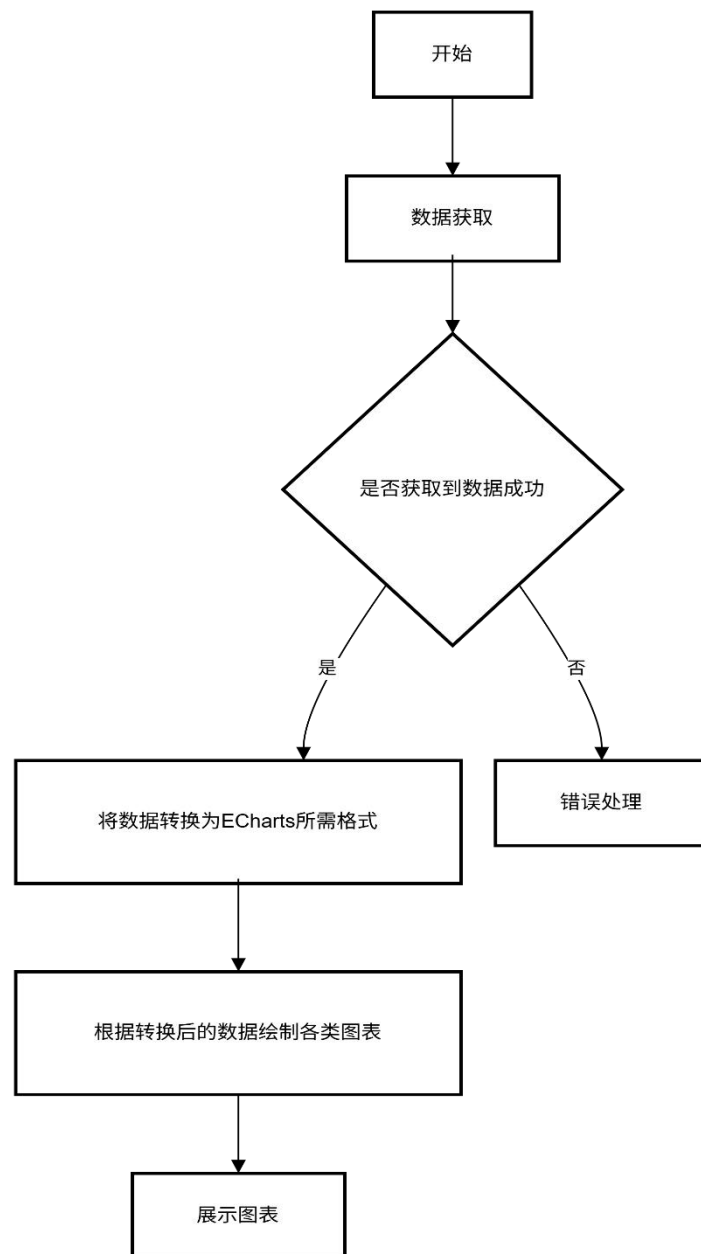


图 2-4 可视化呈现业务流程

### 2.2.2 系统用例及关系图

凭借对酒诗文化数据治理与可视化系统的功能需求以及用户特征展开详细剖析，本研究总结出如图 2-5 所示的核心用例以及与之相关的参与者。

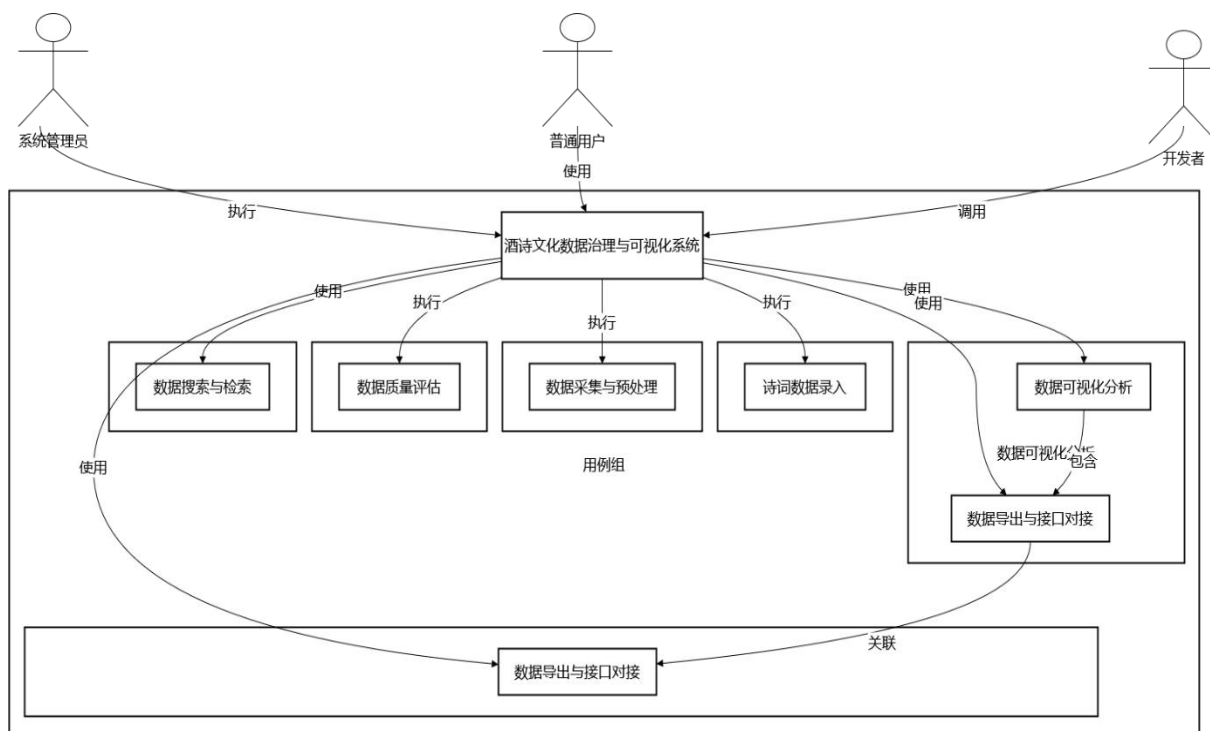


图 2-5 主要用例及关系图

此系统运用多角色协同架构展开设计，当中系统管理员肩负核心运维职责，具体负责诗词数据录入、用户权限配置以及数据备份恢复等基础工作，针对学术研究人员和文化爱好者群体，系统设置了数据质量评估、检索查询以及可视化分析等核心功能模块，企业开发者可借助标准化数据导出接口达成系统对接，在数据处理流程上，数据采集模块负责启动自动化采集与预处理任务，AI 分析模块有数据质量评估与可视化分析两项功能，最后由可视化组件完成分析结果的可视化展示。

## 2.3 数据需求

### 2.3.1 数据来源

本研究运用网络爬虫技术从政府机构以及学术机构等权威平台获取古典诗词和作者信息，以此保证数据来源有广泛性，原始数据拥有准确性，在质量评估环节，依靠前

沿人工智能大模型架构，借助优化设计的提示工程达成多维度自动化评分，保证评价体系有科学性与客观性，所有采集到的数据都要经过覆盖去重、纠错等内容的标准化清洗流程，并且进行多轮人工校验，用以保障数据的精确性与完整性，最终借助规范化存储机制把处理后的数据导入关系型数据库，为后续分析研究提供高质量的数据基础。

### 2.3.2 数据规模

本研究运用的原始数据集由诗歌基础数据、诗人关系数据、诗歌评分数据以及诗人评价数据这四个核心模块构成，经过整合处理后，总数据量大概有 5 万条，其中诗歌基础数据模块详细记录了每一首诗歌的元信息，覆盖诗歌标题、创作朝代、作者署名以及全文内容等关键信息，而诗歌质量评估数据模块则系统地呈现了针对每一首诗歌的量化分析结果，能为后续研究提供关键的评价参考。



图 2-6 诗词数据



图 2-7 诗人数据

### 2.3.3 储存需求

本系统在数据存储层面主要有三个维度的需求：原始文本数据存储、分析结果存储以及结构化关系数据存储，对于基础数据，系统要容纳大约 10 万首古代诗词的完整文本数据、3000 位诗人的详细资料以及相关文化元素数据，在分析数据方面，需存储诗词

质量评估指标、文本特征分析结果、时空分布特征等多维度分析数据，系统打算采用 neo4j 图数据库构建诗人社交网络、主题分类体系和文化元素关联等结构化数据模型。为了应对数据增长和系统扩展需求，设计预留 100% 的存储冗余空间，经过综合测算，系统总存储容量需求约为 23GB。

#### 2.3.4 数据预处理需求

本研究围绕多元数据源的诗词文本以及相关信息搭建了一套系统化的处理框架，一开始进行文本预处理工作，像把繁简字进行转换、将标点规范化并且把格式标准化，接着开展语言学分析，包含分词处理、词性标注以及命名实体识别，在诗人信息这一维度，着重做时空要素提取和标准化处理，具体有生卒年份归一化以及历史地名现代化转换。还建立了文本质量评估体系，开展错别字检测、格律校验以及内容完整性验证，借助机器学习算法构建主题分类模型，达成诗词的自动化分类与标签标注，最后设计数据清洗流程，经过去重、纠错、补全以及规范化操作，保障数据集的准确性与内在一致性。

### 2.4 性能需求

本系统的性能需求大致分为三个性能指标用于性能测试：

表 2-1 三个性能指标用于性能测试

性能指标类型	具体指标	预期值	最低要求	测试方法
响应时间	页面加载时间	< 2 秒	< 3 秒	单用户访问测试
	数据查询响应	< 1 秒	< 1.5 秒	单查询测试
	分析处理时间	< 5 秒	< 8 秒	批处理测试
	API 接口响应	<200ms	< 500ms	接口测试
吞吐量	系统吞吐量	1000 TPS	500 TPS	压力测试
	数据处理量	5000 条/分钟	3000 条/分钟	批处理测试
	峰值处理能力	2000 TPS	1000TPS	峰值测试

性能指标类型	具体指标	预期值	最低要求	测试方法
资源利用率	持续处理能力	800TPS	400TPS	持久性测试
	内存使用率	< 80%	< 90%	资源监控
	磁盘 IO 使用率	< 70%	< 80%	资源监控

2.5 系统用例分析

2.5.1 酒诗文化数据治理与可视化系统的主要用例

本系统的主要用例如下表 2-2:

表 2-2 酒诗文化数据治理与可视化系统的主要用例描述表格

用例编号	用例名称	参与者	前置条件	后置条件	描述
1	诗词数据录入	系统管理员、学术研究人员	用户已登录系统	诗词数据成功录入数据库	系统管理员或学术研究人员通过后端界面手动录入诗词文本、作者信息及相关元数据。系统支持单条数据录入或批量上传功能。
2	数据采集与预处理	数据采集模块	系统已配置好爬虫工具和 API 接口	采集的数据经过清洗、标准化后存储到数据库	系统通过网络爬虫从权威数据源采集诗词数据，并通过 API 与外部数据库同步。采集的数据经过格式校验、繁简转换、去重等预处理步骤。
3	数据质量评估	学术研究人员、系统用户	诗词数据已存储到数据库	生成数据质量评估报告	用户请求对特定诗词数据进行质量评估。系统调用 AI 分析模块，从完整性、准确性、一致性等维度生成评估报告。

用例编号	用例名称	参与者	前置条件	后置条件	描述
4	数据搜索与检索	学术研究人员、普通文化爱好者	系统已索引诗词数据	检索结果以列表形式展示给用户	用户通过关键词、作者、朝代等多维度条件搜索诗词数据。系统利用全文检索技术快速返回匹配结果。
5	数据可视化分析	学术研究人员、普通文化爱好者	诗词数据已存储到数据库	生成可视化图表并展示给用户	用户请求对诗词数据进行可视化分析。系统调用 ECharts 组件，生成诗词数量统计、朝代分布、诗人影响力分析等可视化图表。
6	自动生成质量报告	学术研究人员	数据质量评估已完成	质量报告以 PDF 或 Excel 形式生成并下载	用户请求生成质量报告。系统整合评估结果，生成包含图表、统计指标和改进建议的报告。
7	数据导出与接口对接	系统管理员、企业开发者	数据已存储到数据库	数据成功导出或对接到第三方系统	系统管理员将数据导出为 CSV 或 JSON 格式；企业开发者通过开放 API 获取元数据并集成到第三方应用。
8	数据备份与恢复	系统管理员	系统已配置备份策略	数据成功备份或恢复	系统管理员触发数据备份任务，将数据库内容备份到指定存储位置；在需要时执行数据恢复操作。

### 2.5.2 用例优先级矩阵

根据酒诗文化数据治理以及可视化系统的功能需求分析情况，此次研究依照用例优先级以及功能特性这两个方面，构建出了系统用例描述表格。

表 2-3 用例优先级矩阵

用例编号	用例名称	优先级	说明
UC1	诗词数据录入	高	系统核心功能之一，支持诗词文本、作者信息及相关元数据的录入。
UC2	数据采集与预处理	高	数据治理的关键步骤，通过网络爬虫和 API 接口采集并预处理数据。
UC3	数据质量评估	高	确保数据可靠性的重要环节，利用 AI 分析模块评估数据质量。
UC4	数据搜索与检索	中	用户高频使用功能，支持多维度条件搜索诗词数据。
UC5	数据可视化分析	中	提升用户体验的关键功能，生成诗词数量统计、朝代分布等可视化图表。
UC6	自动生成质量报告	中	专业化需求功能，整合评估结果生成质量报告。
UC7	数据导出与接口对接	低	面向开发者的功能，支持数据导出和对接第三方系统。
UC8	用户权限管理	高	系统安全性的保障，设置用户角色和权限。
UC9	数据备份与恢复	高	系统稳定性保障功能，支持数据备份和恢复操作。

本研究搭建的诗酒文化数据治理系统借助九大核心功能模块达成全生命周期管理，诗词数据录入用以创建基础数据源，数据采集与预处理负责完成原始数据的清洗及标准化工作，数据质量评估可保证信息的可靠性，数据搜索与检索可提供高效的查询服务，数据可视化分析可支持多维度的数据呈现，自动生成质量报告实现智能化评估，数据导出与接口对接保证系统的互联互通，用户权限管理维护数据安全，数据备份与恢复防范系统风险。这些功能模块共同协作，构建起稳定可靠的技术架构，同时形成完整的诗酒文化数据处理闭环，为学术研究、文化传播等应用场景打造强有力的数字化支撑平台。

2.6 本章小结

在这一章节当中，系统地对酒诗文化数据治理与可视化平台的整体设计框架进行了阐述，着重从系统定位、架构设计、功能模块、数据规范、用户画像以及性能指标这六个维度展开了论述，该平台的定位是构建一个综合性服务系统，它将数据治理、智能分析以及可视化呈现整合在一起，可为多元化的用户群体提供专业化的数字服务。在架构

层面，采用的是分层设计理念，借助标准化的接口协议来达成模块间的高内聚低耦合特性，这样一来，能保障系统的安全性与运行效率，又为后续的功能扩展以及前后端独立部署提供了技术方面的保障，功能设计依照数据全生命周期管理原则，构建起了一个完整的业务闭环，这个闭环覆盖了从数据采集、治理分析一直到可视化展示各个环节。在数据层面，明确了多源异构数据的采集规范、预处理流程以及存储方案，同时还对数据规模与存储需求进行了量化，基于用户角色特征所做的需求分析，为系统功能设计提供了精确的依据，而多维度的性能指标体系，则为系统开发与优化确立了量化的目标，最终保证平台可全面支撑酒诗文化数字化服务的各项业务需求。



## 第3章 诗酒数据治理方法设计与实现

### 3.1 数据架构设计

本研究构建了一种依据分层管理理念打造的多级数据架构体系，其核心设计思路是借助模块化组织形式达成数据资源的高效整合以及深入分析，如图所示，该系统的整体架构设计呈现出较大的层次化特性。

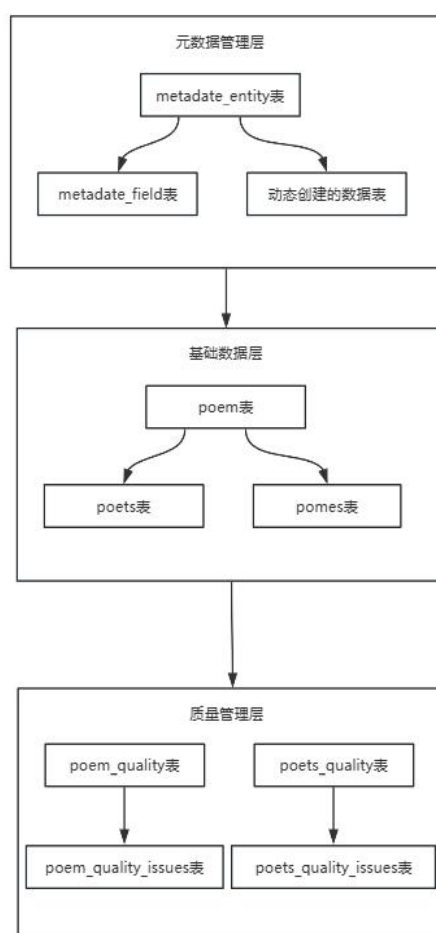


图 3-1 数据架构设计框图

本研究运用自底向上的分层架构设计方式，从基础数据存储开始逐步构建直至高级分析功能，在架构实现方面，先是借助元数据管理层对整体数据结构给予规范化定义以及动态管理，这里的动态表创建机制切实提高了数据模型的可扩展性，基础数据层作为核心存储单元，全面且系统地整合了诗词作品、诗人信息等关键业务数据。为保证数据

可靠性，质量管理层构建了基于评分机制的持续监控体系，在分析层面，集成了可视化展示模块用来支持决策分析。

为保证数据一致性得以达成，系统于各个层级都执行了严谨的数据完整性约束机制，并且对外键关联关系做了专门的优化。

## 3.2 数据质量控制

### 3.2.1 数据验证触发器

为保证系统数据有可靠性以及高质量，此次研究设计并且实施了一套带有多重验证层级的综合性数据校验机制，此机制贯穿于从最开始用户输入一直到最终数据存储的整个流程，数据要依次凭借三个关键的验证环节：

在应用层验证这一环节当中，本研究特意构建了一种前后端协同的双重校验机制，其中前端界面会施行动态表单验证策略，这个策略包含必填字段检测、输入格式校验以及内容长度限制等功能模块，并且会以实时交互的方式给用户提供验证反馈，而后端系统会执行基于业务规则的深度验证流程，借助逻辑约束条件来保证提交数据的业务合规性和完整性。

为了保证数据在入库之前可进行最终的安全性校验，此次研究运用数据库触发器验证机制当作第二道防线，这一机制借助预先设定好的触发规则，在数据执行插入、更新或者删除操作的时候会自动触发多重验证流程，具体包括数据格式标准化处理、业务逻辑合规性审查以及数据关联关系校验等功能模块，在数据库层面搭建起有效的数据质量管控屏障，防止不符合规范的数据写入行为。

为了给数据完整性提供最终的坚实保障，此项研究运用了多层次的数据库约束验证机制：借助实施主键约束达成数据记录唯一性的控制，依靠运用外键约束维持不同数据实体之间关联关系的完整性，并且搭配非空约束与检查约束的双重验证，以此构建起一套严谨的数据质量保障体系。

该验证体系借助多维度且全流程的质量控制机制，在保证数据精确性以及完备性的情况下，提高了系统的稳健性和可靠性，从图 3-2 可看出，这种分级验证架构可以有效阻挡数据污染风险，不断优化数据品质，为上层业务应用搭建坚实的数据支撑平台。



图 3-2 多层数据验证触发器机制

### 3.2.2 数据一致性检查存储过程

为保证数据完整性验证能做到全面，本研究搭建了多层次检测体系，在基础层面，借助数据总量统计分析来验证诗词库与诗人库的基本数据是否完备，接着运用左连接查询技术对评分数据做关联性检测，精确识别并统计出那些缺乏对应诗词记录的孤立评分数据，最后构建基于阈值判定的评分有效性验证机制，凭借系统检测各项评分指标有没有偏离预设的合理区间，以此实现对异常评分记录的高效识别与筛选。

#### 1) 基础数据完整性检查

本研究一开始便执行“SELECT COUNT(\*) FROM poems”以及“SELECT COUNT(\*) FROM poets”这两条 SQL 查询语句，把诗词表跟诗人表里面的记录总数分别存到 total\_poems 和 total\_poets 变量之中。如此操作能获取数据库里的基础数据规模，还为后续的数据处理与验证工作打下了必要的数据库基础，让系统拥有基本的分析条件。

#### 2) 孤立数据检查

本研究执行 SQL 查询语句 "SELECT COUNT(\*) INTO orphaned\_quality\_scores FROM poem\_quality pq LEFT JOIN poems p ON p.id = pq.poem\_id WHERE p.id IS NULL" 来对诗词评分表 poem\_quality 和诗词表 poems 进行左连接操作，依据诗词编号进行连接。这样做的来识别并且统计诗词表里面不存在对应记录的评分数据的数量，然后把结果存到变量 orphaned\_quality\_scores 里面，像这样不符合关联约束的数据记录被称作“孤立数据”，对这些孤立数据进行系统性检测，一方面可有效识别数据关联方面的异常情况，另一方面也可以为维护数据库的完整性和一致性提供关键的依据。

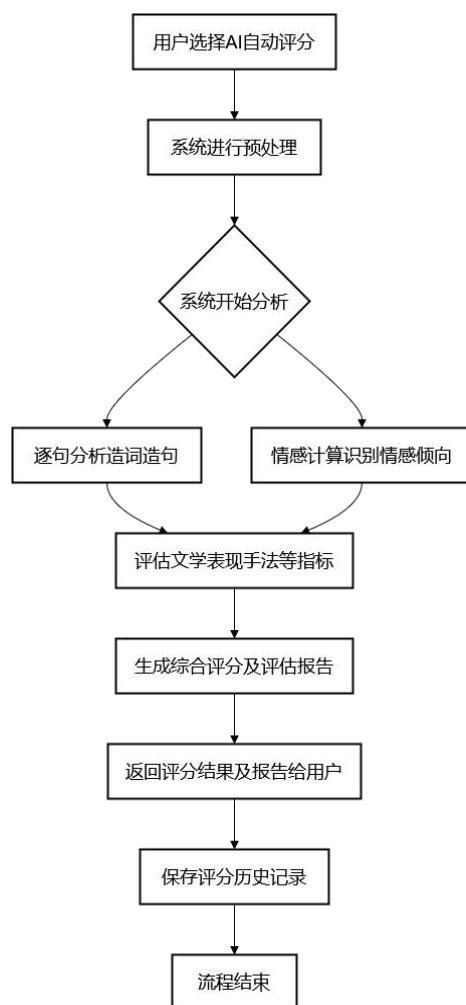
#### 3) 评分数据有效性检查

本研究执行 SQL 查询语句 "SELECT COUNT(\*) INTO inconsistent\_scores FROM poem\_quality WHERE quality\_score > 100 OR quality\_score < 0 OR complete\_score > 100

OR complete\_score < 0 OR accuracy\_score > 100 OR accuracy\_score < 0", 以此对诗词质量评分表里 quality\_score、complete\_score 和 accuracy\_score 这三个维度的评分数据展开有效性检验, 来识别并且统计超出 0 到 100 合理取值范围的异常数据记录。查询结果会存储在 inconsistent\_scores 变量中, 这一数据验证机制可有效识别异常评分, 还可以为后续数据清洗以及质量控制提供关键依据, 保证研究数据的准确性与可靠性。

### 3.3 AI 自动评分

随着数字技术在人文学科领域不断渗透, 本研究构建了一种智能化诗词质量评估系统, 该系统以深度学习架构为基础, 借助多维度特征分析框架, 达成了对文学作品语言特征、情感维度以及意象构建等核心要素的系统化量化评估, 为诗词创作提供了有较高



可信效度的客观评价指标。

图 3-3 AI 自动评分逻辑架构

本研究构建的系统采用多阶段处理流程，初始阶段借助文本预处理技术，涉及分词处理与词性标注，系统性提取词汇多样性指数、句法复杂度特征等关键文本特征，为后续分析筑牢数据根基，特征提取阶段，系统运用自然语言处理技术逐句解析诗词文本，着重考察修辞手法与语言表达特征。基于情感计算算法，系统实现文本情感极性的精准识别，并对情感表达的层次性与细腻度给予量化评估，最终评估阶段，模型整合文学修辞运用度、语言风格辨识度、文化内涵深度等多项评价维度，经加权计算得出诗词作品的综合评价分数。

### 3.3.1 用户交互设计

本研究于交互设计方面构建了颇为直观的评分界面系统，借由简化操作流程达成单次点击便可完成诗词文本提交与评分请求的目的，系统响应机制拥有毫秒级反馈能力，在返回评分数值之际生成结构化评估报告，此报告运用非专业化表述形式系统阐述评分维度及标准，能保证用户即时获取评分结果，又能推动其深入了解评价体系内在逻辑。系统创新性地集成了历史数据存储功能，可支持用户对历次评分记录展开回溯性分析，这一设计实现了创作成果的纵向比较，还为用户提供了可视化监测创作能力发展动态的有效途径。

### 3.3.2 实际应用效果

实证研究显示，这款人工智能自动评分系统呈现出相当出色的评分准确性，其该系统极大地提高了传统诗词评价流程的工作效率，还为创作者提供了有指导价值的反馈机制，这类反馈信息可有效地引导创作者改进创作技巧并提高作品的艺术表现力，在提升诗词创作质量上发挥着关键作用。

## 3.4 性能优化

### 3.4.1 索引优化

索引优化主要集中在四个层面：

在数据库表结构的设计工作当中，选用自增整型字段当作主键索引，可提升数据插入以及查询操作的执行效率。

本研究借助于在外键字段之上构建索引结构的方式，使得关联表之间的查询效率得到了提升，优化了数据库系统的整体性能表现。

为了可在高频查询场景里把检索效率给优化好，此次研究在诗词数据表当中针对朝代与作者字段的组合构建了复合索引结构，此索引策略是把两个有关联的字段进行联合索引，让多条件查询的响应速度得到了提升。

为了保证数据库表结构的完整性约束得以实现，本研究针对元数据表的关键字段执行了 UNIQUE KEY 约束机制，这种设计策略借助强制字段值的唯一性，切实有效地规避了数据冗余问题。

### 3.4.2 分区策略

分区优化主要集中在三个层面：

时序数据分区策略：对于那些包含时间戳字段的数据表而言，采用基于时间范围的分区方案是比较合适的选择，这样的设计可在一定程度上提升历史数据的归档效率，同时让管理更加便捷。

数据量调控机制方面：在本研究之中，借助 `cleanupExcessiveData.js` 脚本搭建起了一套智能化的数据量监控系统，此系统有实时检测数据存储规模的能力，当数据量超过预先设定的阈值时，会自动启动数据清理流程。

在海量数据处理场景里存在中间结果存储需求，本研究提出把临时表用作优化方案，此技术手段可有效提高数据处理效率，还可以降低系统资源消耗。

### 3.4.3 查询优化

运用 COALESCE 函数来对 NULL 值做指定替代值的转换，可切实保障数据处理的完整性和准确性，在数据库查询时，合理运用 JOIN 操作可达成多表间的关联查询，引入子查询可细化查询条件并限定结果范围，以此提升数据获取的精确度，为优化查询性能，在 SQL 语句里设置合适的 WHERE 条件可缩小数据扫描范围，提高查询效率。借助 LIMIT 与 OFFSET 参数配合使用，可实现查询结果的分页显示，这提高了结果的可读性和可控性，还为大规模数据处理带来便利。

### 3.5 本章总结

本章全面展示了基于 MySQL 平台的数据治理实施方案，主要从数据架构搭建、质量保障体系以及系统性能提高三个方面进行阐述，在架构设计方面，运用分层递进的构建方法，从底层数据存储到上层分析应用，都设置了严格的数据完整性约束机制和索引优化方案，对于数据质量管控，构建了包含应用层校验、数据库触发器验证以及约束条件检查的多层次保障系统，同时借助存储过程实现数据一致性的自动化校验。在性能优化阶段，重点研究了索引结构优化、数据分区策略以及查询语句调优等关键技术，以提高系统的数据处理吞吐量和查询响应速度。

## 第 4 章 诗酒数据治理与可视分析系统 详细设计与实现

在完成了前文所提及的系统需求分析以及数字治理模型构建之后，此章节将会着重阐述诗酒数据可视化分析系统的具体实现办法以及其在数据治理领域的应用设计。

### 4.1 技术栈详细设计

本项目采用现代化的前后端分离的技术栈，前端开发选用 Vue 3.x 框架并结合 TypeScript 语言，借助 Vuex 4.x 来达成应用状态管理，运用 Vue Router 4.x 完成路由控制，UI 组件库采用 Element Plus，数据可视化功能借助 ECharts 实现，HTTP 客户端采用 Axios 框架，项目构建工具选用 Vite。后端服务部署在 Node.js 运行环境中，将 Express.js 当作核心 Web 框架，数据库系统采用 MySQL，依靠 mysql2/promise 模块实现连接池优化，同时整合 CORS、Body Parser 以及 Express Router 等中间件，在数据存储层面，系统以 MySQL 关系型数据库为根基，设计了核心数据表结构，以此支持数据治理与分析功能模块的实现。

### 4.2 数据流转流程

Vue 3 框架构建的前端界面在接收用户操作请求后，会利用 Axios HTTP 客户端把请求传送到 Express 后端服务器，后端系统会先验证并预处理请求参数，接着按照业务逻辑需求，凭借 mysql2/promise 连接池实现的数据访问层来执行 MySQL 数据库的增删改查操作。对于需要深入分析的数据内容，系统会把它推送到 AI 分析层进行多维度处理，像构建诗词质量评价体系、量化诗人影响力以及识别酒文化特征等核心功能，经过业务逻辑层整合分析结果和原始数据后，系统会把最终数据包返回给前端，前端应用借助 Vuex 状态管理工具统一管理数据，并用 ECharts 等可视化技术将处理结果以多维图表、地理信息图谱及关联网络等形式展示给终端用户，实现完整的数据闭环流转。在整个流程里，系统依靠实施多层级验证机制、缓存优化策略以及异常处理方案，有效保证了数据传输的准确性和系统运行的高效性。



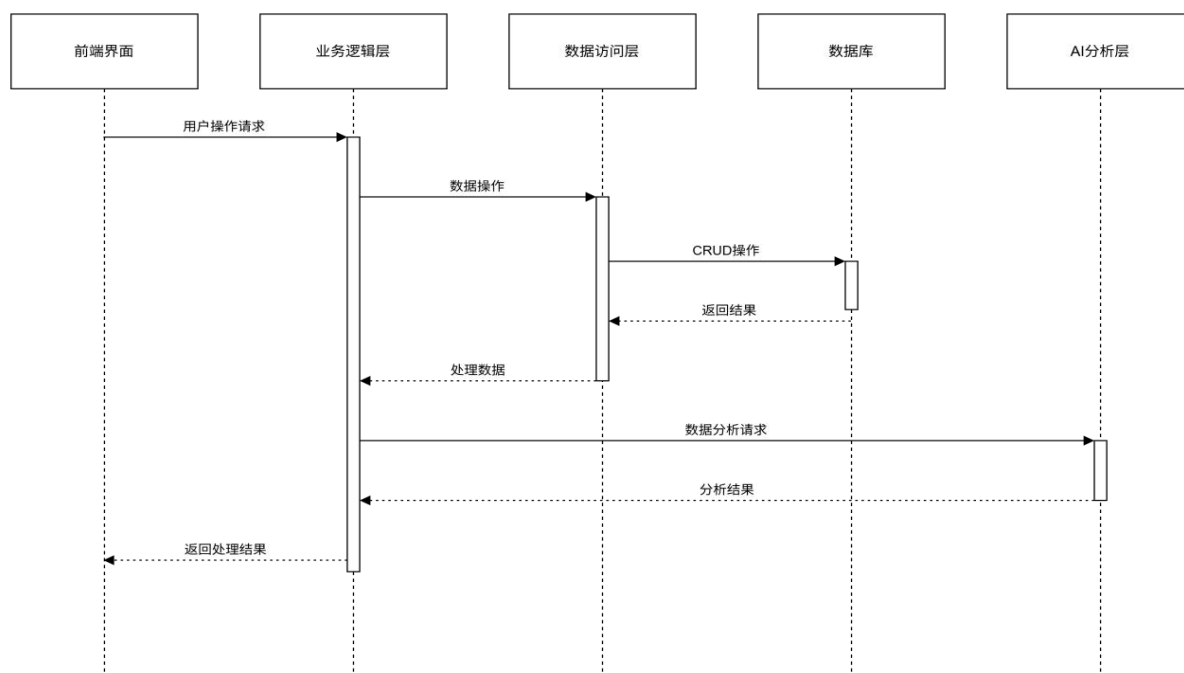


图 4-1 数据流转流程设计

## 4.3 系统界面介绍

### 4.3.1 主界面介绍

如图 4-2 呈现的那样，系统主界面运用元数据管理架构来设计，此界面可供用户针对诗人元数据以及诗词元数据等各种元数据字段开展可视化操作，全部的操作指令借助 node.js 技术实时同步到后端数据库，以此达成动态更新，系统拥有操作审计功能，可自动记录操作对象和相应时间戳，相关日志信息可查看图 4-3 所示的系统界面截图。

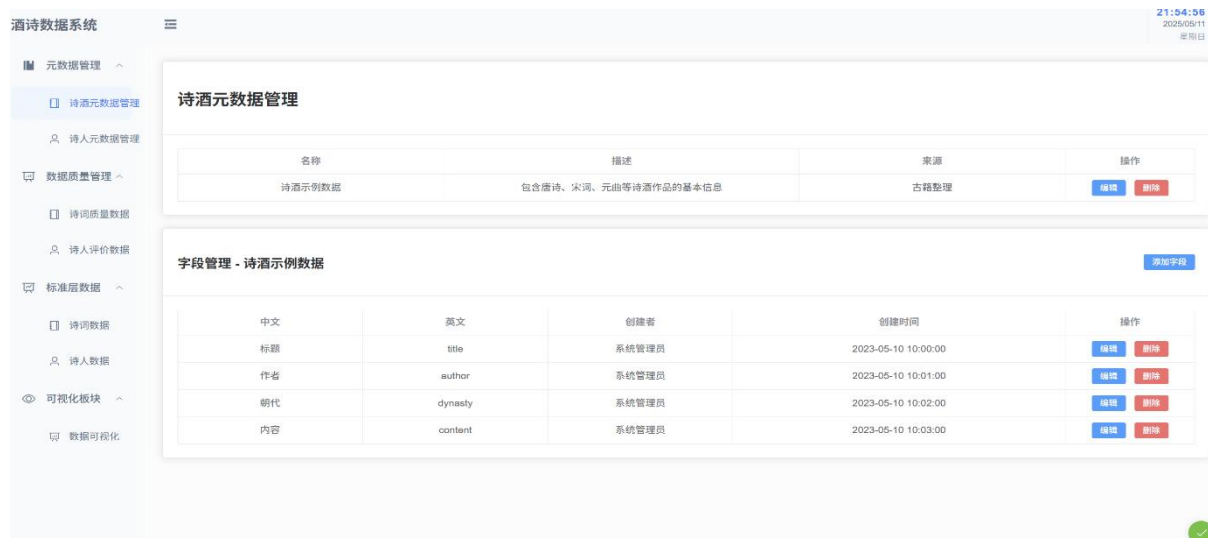


图 4-2 元数据管理界面

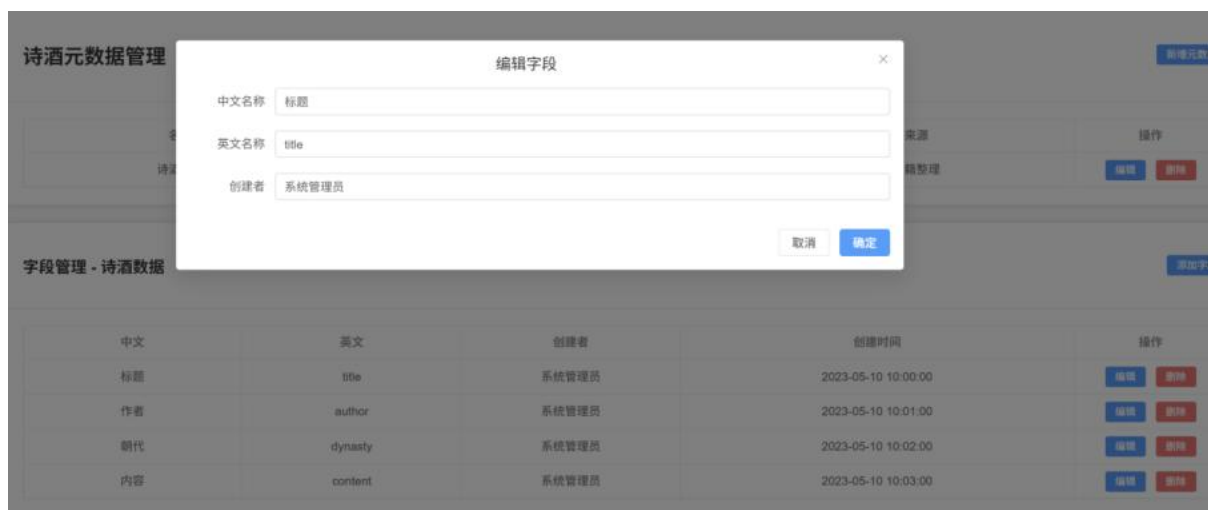


图 4-3 元数据管理操作界面

### 4.3.2 数据质量管理界面

本研究运用双维度数据管理体系，依靠诗词质量数据跟诗人评价数据这两个结构化数据表达成界面数据的质量控制，在构建评价指标体系时，诗词质量维度挑选完整性、准确性以及总体质量当作核心评价指标，诗人评价维度则重点考量创作水平、历史影响力以及时代价值等关键要素，鉴于当下缺少统一的评价标准，本研究创新地运用人工智能技术，依据“中国诗词大会”和“百家讲坛”等专业诗歌鉴赏节目的大量数据，借助

深度学习算法创建评价模型，最终形成有可操作性的量化评价体系。



图 4-4 数据质量管理界面

本研究创建了多维度的诗人筛选机制，借助关键词检索、朝代归属以及时代价值评估等核心维度达成精准数据提取，系统有可视化评价体系的详细展示功能，用户能依靠点击详情按钮查看完整的评估框架，平台还提供智能化质量报告生成服务，可以将当前页面数据的综合评价得分、各朝代评分分布特征以及时代价值构成比例以图形化形式呈现出来。



图 4-5 筛选数据



图 4-6 评价体系查看



图 4-7 质量报告查看

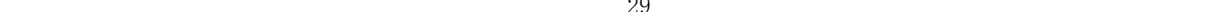


图 4-9 标准层管理-诗词数据-数据筛选操作

对于诗歌以及诗人这类操作对象而言，系统设置了编辑功能按钮来达成数据修改的目的，所有发生变更的信息都会实时同步到后台数据库，以此有效落实数据治理的基本准则，具体的实现方式如图 4-10 所示。

编辑诗人

\* 诗人姓名 马致远

\* 朝代 元

\* 生平事迹 (约1250年—1324年)号东篱，大都（今北京）人。早年热衷功名，曾任江浙行省务官。50岁退隐，作杂剧15种，《汉宫秋》写昭君出塞，列元曲四大悲剧。散曲《天净沙·秋思》"枯藤老树昏鸦"被誉为"秋思之祖"。曲词典雅清丽，《双调·夜行船》"百岁光阴一梦蝶"叹世警人。与关汉卿、白朴、郑光祖并称"元曲四大家"。卒年不详，约在泰定元年（1324）后。

取消 保存修改

图 4-10 标准层管理-诗人数据-数据编辑操作

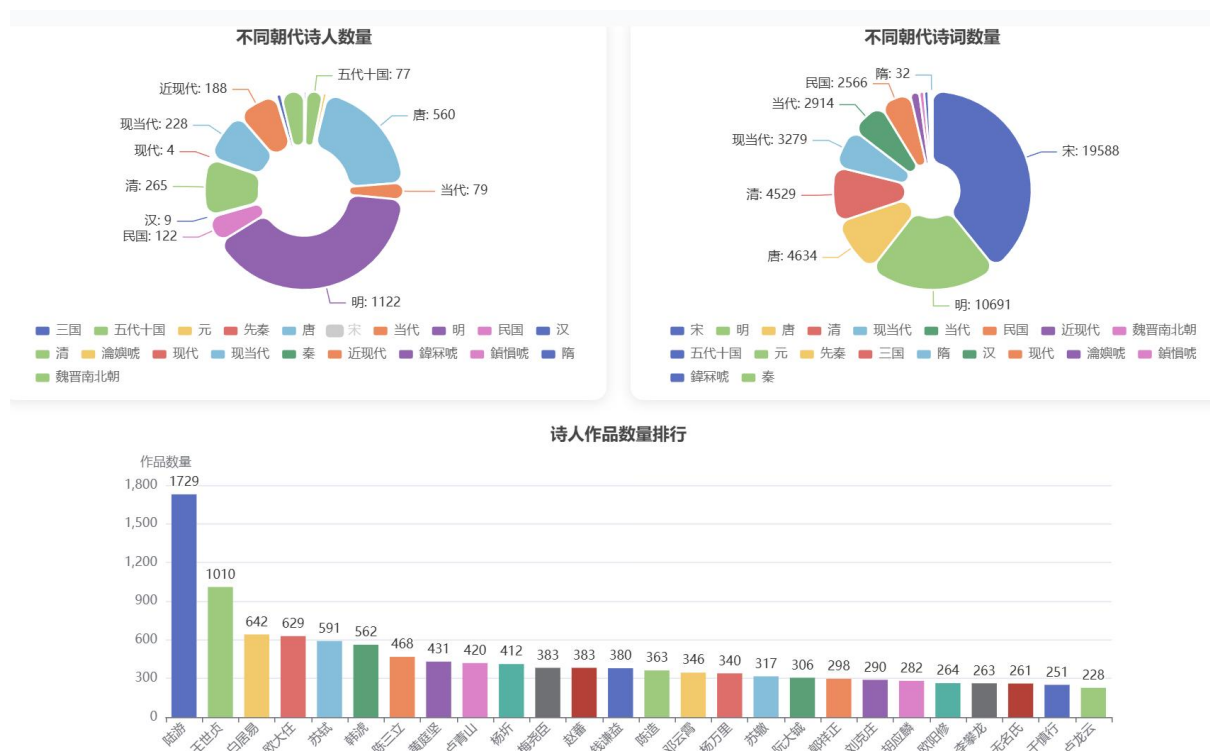
#### 4.3.4 数据可视化界面

本研究运用数据可视化技术，借助五幅交互式图表系统呈现了数据库中的核心指标，其中有诗词总量、诗人总数以及朝代分布等关键维度，以 ECharts 开源可视化库构建图形界面，达成了多维数据的直观展示，具体包含各历史时期诗人数量分布、不同朝代诗词创作总量统计以及诗人作品数量排序分析等内容。

由于 ECharts 组件拥有动态交互的特性，此次研究设计了图表点击响应机制，达成了数据可视化呈现里关键信息的动态聚焦以及差异化展示。

本研究运用环形饼图来强化视觉呈现效果，柱状图达成了颜色自定义以及动态渲染

功能，所有可视化组件都拥有响应式设计特性，其中数据卡片整合了悬停动画交互机制，



各类图表均可支持动态提示功能。

图 4-11 基于 echarts 库的数据展示

## 4.4 可视化实现方式

本研究运用 Vue 3 框架以及 ECharts 可视化库搭建了诗词数据分析仪表盘系统，该系统采用卡片式设计架构，界面顶部区域设置了三组数据统计卡片模块，分别动态展示诗词总量、诗人总数以及朝代数量等核心指标，各个卡片组件都集成了悬停动画特效并且在左侧辅以彩色边框装饰。在界面中央区域，创新性地采用了双饼图并置布局方案，左侧环形饼图直观呈现各朝代诗人数量分布特征，右侧标准饼图清晰呈现朝代诗词数量占比情况，这两个图都有图例交互功能以及动态数据标签显示机制，界面底部区域配备了横向渐变色柱状图，凭借色彩渐变和数值标注相结合的方式，有效突出了诗人作品数量排行榜单。系统采用响应式设计原则，所有可视化组件可根据浏览器窗口尺寸变化进行自适应调整，并且集成了数据悬停提示功能，在视觉设计方面，选用蓝绿柔和色调搭配纯白背景，营造出清新明快的视觉效果，数据交互层面采用实时请求机制，保证图表



可对数据更新做出动态响应，实时呈现最新分析结果。

## 4.5 系统功能介绍

诗歌元数据管理系统当中交互功能的核心模块是借助 poemMetadata.vue 组件来达成的，下面将对其具体的功能架构给予阐述：

### 4.5.1 元数据列表的操作

本研究运用 Element Plus 框架里的 el-table 组件达成元数据列表的动态可视化展示，此表格界面设计含有名称、描述以及来源这三个核心数据列，还集成了编辑和删除这两个功能模块，展开来说，当用户触发编辑功能时，系统会弹出模态对话框来支持元数据信息的修改操作，删除功能借助二次确认对话框机制保障数据操作的可靠性，经用户确认后执行前端数据移除。

```
<el-table :data="metadataList" border stripe style="width: 100%" @row-click="selectMetadata">
  <el-table-column prop="name" label="名称" min-width="150" align="center" />
  <el-table-column prop="description" label="描述" min-width="200" align="center" />
  <el-table-column prop="source" label="来源" min-width="150" align="center" />
  <el-table-column label="操作" width="180" align="center">
    <template #default="scope">
      <el-button size="small" type="primary" @click.stop="editMetadata(scope.row)">编辑</el-button>
      <el-button size="small" type="danger" @click.stop="deleteMetadata(scope.row)">删除</el-button>
    </template>
  </el-table-column>
</el-table>
```

图 4-12 元数据列表的关键代码



诗人	元	查询	重置	+ 新增
诗人序号	诗人	朝代	生平	操作
11	马致远	元	(约1250年—1324年)号东篱，大都（今北京）人。早年热衷功名，曾任江浙行省...	查看 编辑 删除

图 4-13 元数据列表操作界面

### 4.5.2 字段列表展示

本研究借助 Element Plus 框架里的 el-table 组件达成了动态数据表格展示功能，当中 currentFields 数据集经多列配置来呈现：利用 el-table-column 组件依次构建中文名称列、英文名称列、创建者列以及创建时间列，每一列都配置了最小宽度并且设置为居中对齐样式。系统特意设定了固定宽度的操作列，整合由 el-button 实现的“编辑”和“删除”功能按钮，其点击事件分别绑定 editField 和 deleteField 方法并传递当前行数据参数，展



开来说，editField 方法把当前行数据载入 fieldForm 并触发编辑对话框，deleteField 方法则借助确认对话框机制达成数据删除，在用户确认后执行 currentFields 数据移除以及后端同步操作。该设计方案在保证信息可视化呈现的有了高效的数据编辑与删除功能，切实维持了用户界面的简洁性以及操作流程的连贯性。

```
<el-table :data="currentFields" border style="width: 100%">
  <el-table-column prop="fieldNameCn" label="中文" min-width="100" align="center" />
  <el-table-column prop="fieldNameEn" label="英文" min-width="100" align="center" />
  <el-table-column prop="creator" label="创建者" min-width="100" align="center" />
  <el-table-column prop="createTime" label="创建时间" min-width="180" align="center" />
  <el-table-column label="操作" width="180" align="center">
    <template #default="scope">
      <el-button size="small" type="primary" @click="editField(scope.row)">编辑</el-button>
      <el-button size="small" type="danger" @click="deleteField(scope.row)">删除</el-button>
    </template>
  </el-table-column>
</el-table>
```

图 4-14 字段列表展示的关键代码

诗词序号	诗词名	作者	朝代	内容	操作
1	五子歌	无名	先秦	皇祖有训。民可近。不可下。民惟邦本。本固邦宁。予视天下愚夫愚妇。一能胜予。怨岂在...	<input checked="" type="checkbox"/> 查看 <input type="checkbox"/> 编辑 <input type="checkbox"/> 删除
2	送赵任卿芜湖丞	韩元吉	宋	青山照濡须，江驶不可渡。当年黄须儿，跨马识其处。奸雄有遗迹，草木尚西顾。孤城...	<input checked="" type="checkbox"/> 查看 <input type="checkbox"/> 编辑 <input type="checkbox"/> 删除
3	金楼子引殷时语	无名氏	先秦	车行酒。骑行炙。百二十日为一夜。	<input checked="" type="checkbox"/> 查看 <input type="checkbox"/> 编辑 <input type="checkbox"/> 删除
4	申叔仪乞粮歌	无名氏	先秦	佩玉蕊兮余无所系之。旨酒一盛兮余与褐之父睨之。	<input checked="" type="checkbox"/> 查看 <input type="checkbox"/> 编辑 <input type="checkbox"/> 删除
5	士冠辞	无名氏	先秦	令月吉日。始加元服。弃尔幼志。顺尔成德。寿考惟祺。介尔景福。吉月令辰。乃申尔...	<input checked="" type="checkbox"/> 查看 <input type="checkbox"/> 编辑 <input type="checkbox"/> 删除
6	和辉弟夕佳亭歌	陈达	明	只幅蒲帆回宦海，蝉鬓浮名陋华彩。小亭兀坐几忘年，万斛红尘难我洗。安身宁有不龟...	<input checked="" type="checkbox"/> 查看 <input type="checkbox"/> 编辑 <input type="checkbox"/> 删除
7	投壶辞	无名氏	先秦	有酒如淮。有肉如坻。寡君中此。为诸侯师。有酒如渑。有肉如陵。寡君中此。与君代...	<input checked="" type="checkbox"/> 查看 <input type="checkbox"/> 编辑 <input type="checkbox"/> 删除
8	清旷亭送子云得有字	韩元吉	宋	欢期苦难逢，离别乃易久。天涯老兄弟，况复亲白首。田园固不足，功名亦何有。今年...	<input checked="" type="checkbox"/> 查看 <input type="checkbox"/> 编辑 <input type="checkbox"/> 删除
9	国风·邶风·柏舟	诗经	先秦	汎彼柏舟，亦汎其流。耿耿不寐，如有隐忧。微我无酒，以敖以游。我心匪鉴，不可以...	<input checked="" type="checkbox"/> 查看 <input type="checkbox"/> 编辑 <input type="checkbox"/> 删除
10	别子云	韩元吉	宋	离觞不解醉，愁多令酒醒。别怀亦云恶，况此兄弟情。春风酿寒雨，十日未肯晴。回头...	<input checked="" type="checkbox"/> 查看 <input type="checkbox"/> 编辑 <input type="checkbox"/> 删除

图 4-15 字段列表展示

#### 4.5.3 添加/编辑字段对话框

本研究借助 Element Plus 框架里的 el-dialog 组件达成了字段管理对话框功能，此对话框含有中文名称、英文名称以及创建者这三个核心输入项，在用户交互这一方面，系统针对“添加字段”按钮的点击事件设定了元数据选择状态检测机制，经过验证后呈现

出空白表单对话框，对于“编辑”操作而言，会自动把选定字段的现有数据预先填充到表单当中。表单提交阶段施行必填项验证流程，依照操作类型来执行相应的数据处理：

```
const editField = (row) => {  
  Object.assign(fieldForm, row)  
  isEditingField.value = true  
  showFieldForm.value = true  
}
```

新增操作会自动生成唯一标识符以及创建时间戳，并且将数据状态标记为待同步，编辑操作则直接对现有字段信息进行更新。

图 4-16 添加字段按钮实现逻辑关键代码

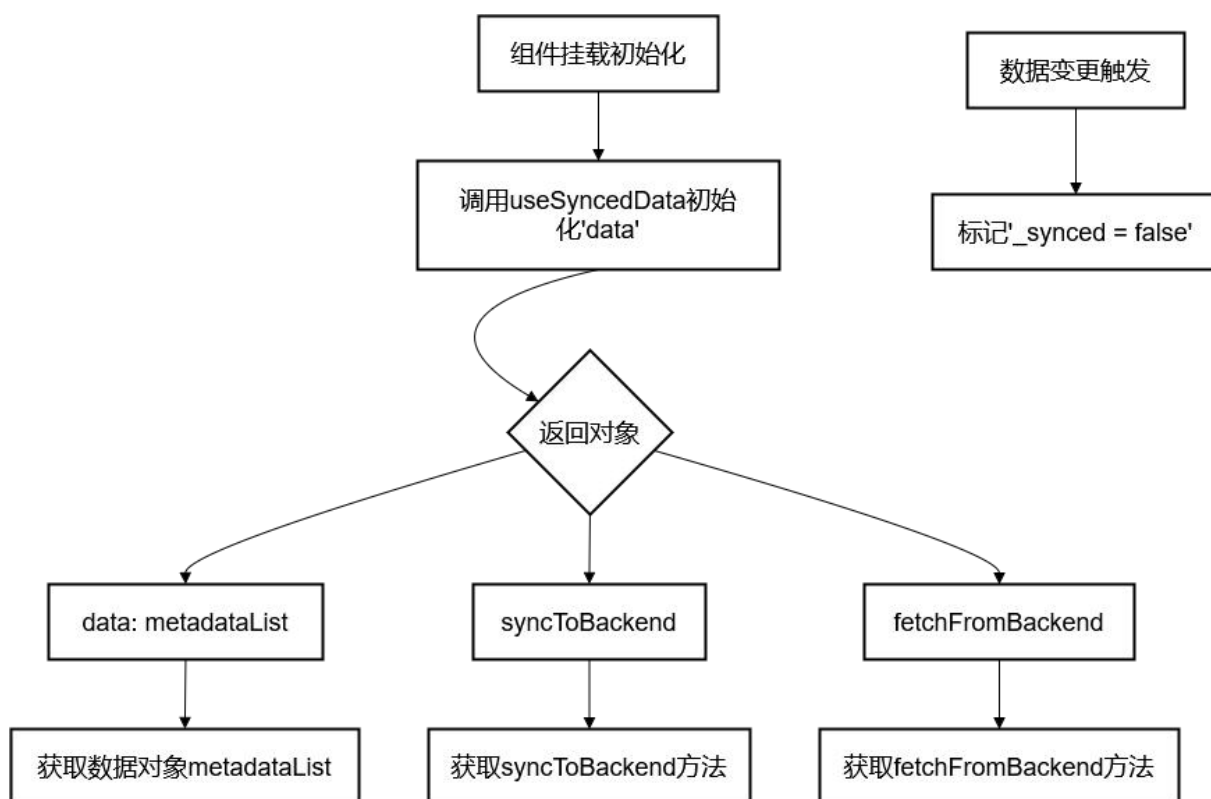
图 4-17 表单验证机制逻辑关键代码

```
const addField = () => {  
  if (!currentMetadataId.value) {  
    ElMessage.warning('请选择一个元数据')  
    return  
  }  
  resetFieldForm()  
  isEditingField.value = false  
  showFieldForm.value = true  
}
```

#### 4.5.4 数据同步

本研究借助自定义开发的 `useSyncedData` 工具来达成数据同步功能，此工具可返回有响应式特性的数据对象以及相应的同步操作办法，在具体的实现过程当中，借助调用 `const { data: metadataList, syncToBackend, fetchFromBackend } = useSyncedData('metadataList', initialData)` 这条语句来完成数据的初始化以及同步方法的获取。为了可有效地追踪数据状态的变化，系统设计了基于 `_synced` 和 `_action` 标记的状态跟踪机制，其中尚未同步的数据会在执行 `syncToBackend()` 方法的时候自动传输至后端，针对定时同步的需求，系统借助 `startAutoSync(30000)` 方法构建了间隔为 30 秒的自动同步机制，并且在组件卸载的时候调用 `stopSync()` 方法来终止同步过程。为了防止出现意外的数据丢失情况，系统还运用 `window.addEventListener('beforeunload', handleBeforeUnload)` 事件监听机制，在页面关闭之前借助 `handleBeforeUnload` 函数强制

执行 `syncToBackend()` 操作，以此保证所有等待同步的数据都可得到妥善的保存。该方案



在保障数据实时性的借助对同步频率进行优化，有效地降低了网络请求负载，并且建立了完善的数据安全保障体系。

图 4-18 数据同步的实现逻辑

#### 4.5.5 生命周期钩子

在 Vue 3 组合式 API 框架当中，`poemMetadata.vue` 组件运用生命周期钩子机制达成了高效的数据管理以及资源回收策略，展开来说，在 `onMounted` 生命周期阶段执行了如下关键操作：先是借助 `fetchFromBackend()` 方法从服务端异步获取最新元数据，接着运用 `selectMetadata(metadataList.value[0])` 完成默认元数据项的初始化选择，为保证数据实时性，依靠 `stopSync = startAutoSync(30000)` 构建了 30 秒间隔的自动同步机制，同时利用 `window.addEventListener('beforeunload', handleBeforeUnload)` 注册页面卸载前回调，切实保障了用户会话数据的持久化存储。与之相对应的，`onBeforeUnmount` 钩子实现了资

源的系统化释放：经过条件判断 `if(stopSync)` 后执行 `stopSync()` 终止定时同步任务，并且调用 `window.removeEventListener('beforeunload', handleBeforeUnload)` 解除事件绑定，以此避免潜在的内存泄漏问题。该设计方案借助严谨的生命周期管理，实现了组件数据的动态同步，还优化了系统资源利用率，提升了应用程序的运行稳定性与终端用户的操作体验。

```
onMounted(() => {  
  // 从后端获取最新数据  
  fetchFromBackend()  
  
  // 尝试从本地存储恢复状态  
  if (metadataList.value.length > 0) {  
    selectMetadata(metadataList.value[0]);  
  }  
  
  // 启动自动同步（每30秒同步一次）  
  stopSync = startAutoSync(30000)  
  
  // 添加页面关闭前同步数据  
  window.addEventListener('beforeunload', handleBeforeUnload)  
})  
  
onBeforeUnmount(() => {  
  // 停止自动同步  
  if (stopSync) stopSync()  
  
  // 移除事件监听  
  window.removeEventListener('beforeunload', handleBeforeUnload)  
})
```

图 4-19 生命周期钩子实现的关键代码

#### 4.5.6 其他功能

在 `poemMetadata.vue` 组件里，`getCurrentMetadataName` 和 `selectMetadata` 这两个核心辅助函数共同协作来完成元数据的选择以及展示功能，其中 `getCurrentMetadataName` 会对 `metadataList` 数组展开遍历，去匹配与 `currentMetadataId` 标识符相对应的元数据项，接着返回该项目的名称属性或者预设的提示文本，这样的机制切实保障了界面层对于当前操作元数据的可视化呈现。而 `selectMetadata` 函数主要是响应用户的元数据选择行为，它接收元数据行对象参数后，把该对象的 `ID` 属性赋予

currentMetadataId 状态变量，同时从 fieldsData 数据源里提取相应的字段列表并更新到 currentFields 状态，以此建立起元数据与字段之间的动态关联关系。这两个函数协同运行，保证了用户界面数据展示的一致性，还优化了元数据与字段间的逻辑关联机制，提升了系统在元数据管理方面的交互体验。

```
// 获取当前元数据名称
const getCurrentMetadataName = () => {
  const metadata = metadataList.value.find(item => item.id === currentMetadataId.value)
  return metadata ? metadata.name : '请选择元数据'
}

// 选择元数据
const selectMetadata = (row) => {
  currentMetadataId.value = row.id
  currentFields.value = fieldsData.data.value[row.id] || []
}
```

图 4-20 其他功能关键代码

## 4.6 本章总结

在这一章节当中，针对诗酒文化数据治理与可视化分析平台，对其整体架构设计以及技术实现方案进行了系统且全面的阐述，从技术架构方面来看，该系统采用的是前后端分离的设计模式，其中前端部分是基于 Vue 3.x 框架来构建的，在构建过程中集成了 TypeScript、Vuex 状态管理、Vue Router 路由以及 Element Plus 组件库，而数据可视化模块则选用 ECharts 来给予实现。后端服务是依托 Node.js 运行环境的，借助 Express.js 框架进行开发，同时配合 MySQL 关系型数据库以及配套中间件，最终构建起了一个高效且稳定的数据处理体系，此系统达成了完整的数据生命周期管理流程，涉及了用户请求处理、后端业务逻辑执行、数据库持久化操作以及前端可视化呈现等方面，切实保障了数据处理的时效性与准确性。其核心功能模块包含四大界面，元数据管理界面主要负责诗人及诗词基础信息的增删改查操作，并且可实现数据库实时同步，数据质量评估界面凭借构建多维度评价指标体系，再结合人工智能技术达成自动化质量分析以及报告输出，标准规范管理界面提供数据标准化处理与查询功能，以此保证治理流程的合规性，可视化分析界面基于 ECharts 技术实现动态交互式图表展示，支持响应式布局与实时数

据更新，为用户提供直观且高效的数据探索体验。这一章节从系统架构、技术选型一直到功能实现都做了阐述，充分呈现出数据治理与可视化分析技术在诗酒文化研究领域的创新应用。

## 第 5 章 诗酒数据可视分析与数据治理系统测试

本研究运用的是双维度评估体系，具体囊括功能测试以及性能测试这两大模块，在后续的章节当中将会围绕上述提及的测试维度展开详细的论述。

### 5.1 功能性测试

本系统功能测试分为前端功能和后端功能测试。

#### 5.1.1 后端功能测试

本研究打造了一套针对数据全生命周期的测试方案，其中覆盖了数据采集、预处理、存储管理、查询检索、分析计算、质量评估、更新同步、接口对接以及容灾备份等一系列核心功能模块，此测试体系可对后端架构的功能完备性展开系统验证，同时评估其对前端业务请求的稳定支撑能力，并且保证在数据处理的整个链条中，数据完整性与一致性的技术指标契合系统设计规范的要求。

表 5-1 后端功能测试：数据采集与预处理

用例编码	T-001			
功能描述	采集并预处理诗酒数据			
用例目的	测试后端数据采集与预处理功能			
用例前提	前端发送数据采集请求			
子用例编号	输入	期望输出	实际输出	状态
T-001-001	前端数据采集请求	从权威网站采集诗酒数据并完成预处理	从权威网站采集诗酒数据并完成预处理	正确

表 5-2 后端功能测试：数据存储

用例编码	T-002			
功能描述	存储诗酒数据至 MySQL			
用例目的	测试后端数据存储功能			
用例前提	后端获取到预处理后的诗酒数据			
子用例编号	输入	期望输出	实际输出	状态
T-002-001	预处理后的诗歌数据	数据成功存储至 MySQL 数据库	数据成功存储至 MySQL 数据库	正确

表 5-3 后端功能测试：数据查询与分析

用例编码	T-003			
功能描述	查询并分析诗酒数据			
用例目的	测试后端数据查询与分析功能			
用例前提	前端发送数据查询请求			
子用例编号	输入	期望输出	实际输出	状态
T-003-001	前端数据查询请求	返回查询结果并进行数据分析	返回查询结果并进行数据分析	正确

表 5-4 后端功能测试：数据质量评估

用例编码	T-004			
功能描述	评估诗酒数据质量			
用例目的	测试后端数据质量评估功能			
用例前提	后端获取到待评估的诗酒数据			
子用例编号	输入	期望输出	实际输出	状态
T-004-001	待评估的诗酒数据	返回数据质量评估报告	返回数据质量评估报告	正确



表 5-5 后端功能测试：数据采集与预处理

用例编码	T-005			
功能描述	更新并同步诗酒数据			
用例目的	测试后端数据更新与同步功能			
用例前提	前端发送数据更新请求			
子用例编号	输入	期望输出	实际输出	状态
T-005-001	前端数据更新请求	数据更新并同步至数据库	数据更新并同步至数据库	正确

表 5-6 后端功能测试：数据导出与接口对接

用例编码	T-006			
功能描述	导出诗酒数据并对接外部接口			
用例目的	测试后端数据导出与接口对接功能			
用例前提	前端发送数据导出请求			
子用例编号	输入	期望输出	实际输出	状态
T-006-001	前端数据导出请求	数据成功导出并完成接口对接	数据成功导出并完成接口对接	正确

表 5-7 后端功能测试：数据备份与恢复

用例编码	T-007			
功能描述	备份并恢复诗酒数据			
用例目的	测试后端数据备份与恢复功能			
用例前提	后端完成数据备份操作			
子用例编号	输入	期望输出	实际输出	状态
T-007-001	数据备份指令	数据成功备份并可恢复	数据成功备份并可恢复	正确

本研究精心设计的数据全生命周期功能测试体系，对数据采集、预处理、存储、查询、分析、质量评估、更新同步、接口对接以及备份恢复等核心环节的运行效能，进行

了全面且系统的验证，实验数据说明：在数据采集环节，系统可精准地获取权威数据源的信息，并顺利完成预处理工作，经过预处理的数据，可以高效地存入 MySQL 数据库，查询请求处理模块可精确地响应前端需求，并提供准确的分析结果，质量评估模块可生成有一定参考价值的评估报告，数据同步机制实现了实时更新的目标，接口对接功能运行顺畅，备份恢复系统呈现出可靠的容灾能力。测试结果充分证明了后端系统架构在功能完整性、运行稳定性方面有优越的性能，其数据一致性与完整性完全契合设计规范，为前端业务提供了坚实的底层支撑。

5.1.2 前端功能测试

本研究针对用户界面设计展开，着重探讨前端系统于诗酒文化数据展示和分析里的关键作用，借助构建涉及数据可视化呈现、交互式操作、数据检索与过滤等功能模块的综合性测试方案，以此验证前端界面在提高用户数据获取效率以及操作便捷性方面的实际成效。

表 5-8 前端功能测试：数据可视化展示

用例编码	T-008			
功能描述	诗词数据可视化展示			
用例目的	测试前端数据可视化展示功能			
用例前提	后端返回诗词数据			
子用例编号	输入	期望输出	实际输出	状态
T-008-001	返回的诗词数据	ECharts 生成的 可视化图表正 确展示诗词数 据	ECharts 生 成的可视化 图表正确展 示诗词数据	正确

表 5-9 前端功能测试：用户交互操作

用例编码	T-009			
功能描述	用户交互操作响应			
用例目的	测试前端用户交互功能			
用例前提	用户进行界面交互操作			
子用例编号	输入	期望输出	实际输出	状态
T-009-001	用户界面交互操作	系统响应并完成相应操作	系统响应并完成相应操作	正确

表 5-10 前端功能测试：数据查询与筛选

用例编码	T-010			
功能描述	诗词数据查询与筛选			
用例目的	测试前端数据查询与筛选功能			
用例前提	用户输入查询条件			
子用例编号	输入	期望输出	实际输出	状态
T-010-001	用户查询条件	返回符合查询条件的诗词数据	返回符合查询条件的诗词数据	正确

表 5-11 前端功能测试：质量报告生成与展示

用例编码	T-011			
功能描述	数据质量报告生成与展示			
用例目的	测试前端质量报告功能			
用例前提	后端返回质量评估数据			
子用例编号	输入	期望输出	实际输出	状态
T-011-001	质量评估数据	生成并展示数据质量报告	生成并展示数据质量报告	正确

表 5-12 前端功能测试：界面响应与适配

用例编码	T-012			
功能描述	界面响应速度与适配性			
用例目的	测试前端界面响应与适配功能			
用例前提	用户访问系统界面			
子用例编号	输入	期望输出	实际输出	状态
T-012-001	用户界面访问请求	界面快速响应 且在不同设备 上适配良好	界面快速响 应且在不同 设备上适配 良好	正确

表 5-13 前端功能测试：用户界面一致性测试

用例编码	T-013			
功能描述	界面适配性与兼容性			
用例目的	测试前端界面兼容性			
用例前提	用户登录系统界面			
子用例编号	输入	期望输出	实际输出	状态
T-013-001	在不同浏览器（如 Chrome、Firefox、Safari、Edge 等）中登录系统。	所有元素的显示效果和交互方式都符合设计规范。	界面快速响应且在不同设备上适配良好	正确

依据诗酒文化数据展示与分析系统的功能需求情况，本研究设计并且实施了包含多维度功能模块的综合性测试方案，着重考察了数据可视化呈现、交互操作响应以及数据检索过滤等核心功能，实验数据显示：在数据可视化方面，ECharts 图表组件可以准确呈现诗词数据分布特征，交互功能测试说明用户操作指令可得到及时响应并执行，检索系统测试验证了查询条件输入与结果返回的匹配情况，质量评估模块测试证实了数据质量报告的生成与展示功能，跨设备兼容性测试则说明系统界面有良好的响应速度与适配性能。这些实证结果充分证明了该前端系统在优化用户数据获取路径、提升操作效率方

面取得的成效，同时也验证了其在诗酒文化数据展示与分析应用中的关键价值，为用户提供了优质的交互体验解决方案。

5.2 性能测试

本研究围绕诗酒数据可视分析以及数据治理系统性能优化的需求展开，着重考察了首屏加载时间也就是 FCLT 以及核心 Web 指标这两个关键性能参数，这里面，FCLT 是用来衡量用户从发起页面请求开始一直到首屏内容呈现出来所需时间的一项关键指标，它和用户体验直接相关，核心 Web 指标是借助最大内容绘制即 LCP、首次输入延迟也就是 FID 以及累积布局偏移即 CLS 这三个维度，对网页的加载效率、交互响应速度以及视觉稳定性进行系统评估的。依据上述指标的测试数据，我们得到了下面这些实证结果：

表格 5-14 系统首屏加载时间（FCLT）测试

测试样例	测试结果（秒/s）	是否通过
T-FCLT-1	3.801	通过
T-FCLT-2	5.254	通过
T-FCLT-3	4.780	通过
T-FCLT-4	4.120	通过
T-FCLT-5	2.189	通过
T-FCLT-6	3.880	通过

依据表 5-13 的测试数据展开分析，研究系统于不同测试场景下的首屏加载性能颇为出色，全部测试样本的首屏内容加载时间都未超出 6 秒的界限，在 T-FCLT-5 测试场景里，系统呈现出极为突出的性能指标，其首屏加载时间仅为 2.189 秒，此数据充分证实系统在该特定应用场景下拥有高效的内容渲染能力，可给用户带来卓越的初始访问体验。而其余测试样本的加载时间都保持在合理范围，这一情况有力说明系统有稳定的响应能力，可以契合用户对数据信息的即时获取需求。

表 5-15 加载性能（LCP）测试

测试样例	测试结果（秒/s）	是否通过
T-LCP-1	1.745	通过
T-LCP-2	1.628	通过
T-LCP-3	0.815	通过
T-LCP-4	1.543	通过
T-LCP-5	1.832	通过
T-LCP-6	1.811	通过

依据表 5-14 所展示的加载性能测试数据，全部测试样例的 LCP 指标都优于 2.5 秒的基准值，当中 T-LCP-3 测试样例的表现颇为突出，其 LCP 值是 0.815 秒，此结果验证了系统在最大内容绘制方面有不少的响应速度，该性能优势体现了系统在资源加载优先级调度和效率优化方面有效，还凭借关键内容快速呈现改善了用户体验，让用户对系统性能的主观评价有了提升。

### 5.3 本章总结

本章围绕诗酒数据可视分析与治理系统开展了系统的功能和性能测试验证工作，在功能测试阶段，着重考查了后端数据处理的整个流程以及前端可视化交互系统的运行效率，测试数据显示，系统的各个功能模块都符合设计指标要求，有效保证了数据处理全生命周期的数据完整性和一致性，同时为用户给予了高效且直观的数据分析交互体验。

本研究对系统性能开展了全面测试分析，着重考量了首屏加载时间以及核心 Web 指标等关键参数，测试得出的数据说明，系统性能比预期标准要好：所有测试样本的首屏加载时间都没超过 6 秒，其中最好的记录是 2.189 秒，在核心 Web 指标方面，LCP 数值都符合低于 2.5 秒的行业规范，FID 指标处于良好区间，CLS 值稳定保持在 0.1 以下，这些数据证实了系统可给用户提供高效、流畅且视觉体验稳定的交互环境。

根据之前的实验验证，本章节借助系统化测试，证实了诗酒数据可视化分析及治理平台在功能完备程度与运行稳定状况方面取得了良好成效，该平台呈现出不错的人机交互体验和数据处理效率，而且成功搭建起来，为诗酒文化遗产的数字化保护与学术研究实践给予了有力的技术支持。

## 结论

针对酒诗数据分散且格式多样、存在大量噪声问题，设计实现了一套酒诗文化数据治理与可视化系统。系统基于多层架构设计理念，把系统划分成表现层、业务逻辑层、数据访问层、数据存储层以及 AI/数据分析层这五个功能层次，在提升系统的可维护性与可扩展性基础上，为各功能模块开发划定了清晰边界。

技术实现方面，前端开发用 Vue 3 框架和 TypeScript 语言，基于 Element Plus 组件库搭建用户交互界面，借助 ECharts 实现复杂数据可视化，后端服务基于 Node.js 运行环境，用 Express.js 框架构建服务器应用，凭借 mysql2/promise 库优化 MySQL 数据库连接池管理，保证数据操作高效及系统运行稳定。系统创新性整合了 AI 分析模块，该模块用深度学习技术对诗词文本做情感倾向分析和质量评估，提高了数据应用价值与深度挖掘潜力。

数据治理体系方面，运用多源异构数据采集办法，一方面借助 Web 爬虫以及 API 接口来获取权威数据源的原始诗词文本，另一方面还辅以人工录入机制来开展数据补充工作，在数据处理阶段，构建了标准化的清洗流程，以此有效提高数据质量的一致性。为保证数据质量，系统施行多层次验证策略，从前端输入验证、后端业务逻辑校验直至数据库触发器和约束条件设置，形成一套全面的质量控制体系，提高了数据的完整性与可靠性。

数据可视化方面，系统采用柱状图和饼图两种直观的可视化形式呈现酒诗文化数据。柱状图用于展示诗人酒诗的数量，通过直观的柱状对比，用户可以清晰地看到之间的差异。饼图则用于呈现酒诗中不同朝代诗人数量、不同作者作品的分布等，通过扇形区域的大小直观地展示各部分的比例关系。这些可视化图表为用户提供了直观易懂的分析视角。

本研究打造的酒诗文化数据治理和可视化系统，依靠智能化分析技术以及可视化呈现手段，给与用户清晰的数据呈现，该系统的技术架构与数据治理方案呈现出不错的通用性和扩展性特点。展望未来，随着数据规模不断扩大以及技术迭代发展，本系统的数据质量及数据治理能力将不断提高发展。

## 致谢

在毕业之际，以毕业论文的方式结束大学四年的美好时光。在撰写毕业论文的过程中，深刻体会到法学知识就像建房子，扎实的知识基础才会有牢固的地基，构建好论文的框架和清晰的逻辑思维才能写好一篇有质量的论文。此次的毕业论文也让我知道，独立的思考善于发现问题是作为一名法律人应具备的美好品质。

本次毕业论文能顺利完成首先最感谢的是我的指导老师王桂娟老师，从我们论文的选题、开题报告、中期检查、定稿和最终稿，王老师都以最负责、最认真的态度给我们进行指导，对学生的毕业论文进行严格把关，针对我们的论文进行专业的指导，针对我们存在的问题，给予有建设性的建议，给学生进行论文修改带来了很大便利。对王老师表示诚挚的感谢。其次也非常感谢对我予以帮助的同学，感谢你们在我写作的过程中，遇到困惑时给予我帮助，对于我的毕业论文完成也起到关键作用。

大学四年即将画上圆满的句号，在此也感谢各位任课老师大学四年的教导。对于论文老师指出存在的问题，我会虚心听取，积极的改进。在此再次向各位帮助过我的老师和同学们表示诚挚的感谢。



## 参考文献

- [1] 王子今. 中国诗酒文化论[J]. 社会科学家, 2010(6): 128-132.
- [2] 王天文. 论元数据在古籍数字化资源整合中的关键作用[J]. 图书情报知识, 2022, (2): 23-31.
- [3] DHQ数据组. Digital Humanities Quarterly: Reading the Quan Tang shi: Literary and Historical Perspectives[J]. Digital Humanities Quarterly, 2019, 13(4): 1-24.
- [4] Hou A, Frank A. Analyzing Sentiment in Classical Chinese Poetry[C]//Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH), Beijing, China. Association for Computational Linguistics, 2015: 15-24.
- [5] Li X, et al. Can AI Write Classical Chinese Poetry like Humans? An Empirical Study Based on Large Language Models[J]. arXiv preprint arXiv:2401.04952, 2024.
- [6] Long H. New Frontiers of Electronic Textual Research in the Humanities: Chinese Literary Allusions and Digital Philology[J]. Journal of Chinese Literature and Culture, 2018, 5(2): 411-431.
- [7] Wang P, et al. Editorial: Cross-Modal Learning: Adaptivity, Prediction and Interaction[J]. Frontiers in Neurorobotics, 2022, 16: 884970.
- [8] Scene clusters, causes, spatial patterns and strategies in the cultural landscape of Chinese poetry[EB/OL]. <https://www.nature.com/articles/s40494-023-01051-x>, 2025-05-07.
- [9] 李伟, 沈睿. 数字人文与古典诗词数据库建设研究[J]. 情报杂志, 2021, 40(5): 122-128.
- [10] 刘颖. 主题型古诗词数据库建设与应用分析——以“诗酒文化”为例[J]. 现代情报, 2023, 43(8): 143-150.
- [11] 王天文. 论元数据在古籍数字化资源整合中的关键作用[J]. 图书情报知识, 2022, (2): 23-31.
- [12] 王铮, 陆雅琦. 基于AI的古典诗歌情感分析研究进展[J]. 计算机科学, 2023, 50(4): 251-256.
- [13] 刘超, 李欣. 面向知识服务的古诗文可视化设计与评价[J]. 图书馆理论与实践, 2022, 44(1): 51-58.
- [14] 吴昊, 王素玉. 面向数字人文的中国古籍数字资源开放现状与改进路径[J]. 图书馆, 2022(6): 31-38.
- [15] Crane G. The Perseus Project and Beyond: How Building a Digital Library Challenges the Humanities and Technology[J]. D-Lib Magazine, 1998, 4(1): 1-25.

- [16] Europeana Foundation. Europeana Strategy 2020-2025[EB/OL]. <https://pro.europeana.eu/page/strategy>, 2024-06-01.
- [17] Gradmann S. Knowledge = Information in Context: On the Importance of Semantic Contextualisation in Europeana[J]. Liber Quarterly, 2010, 19(3-4): 185-203.
- [18] Liu J, Hill A, Cao Y, et al. Cultural Heritage and the Semantic Web: Encoding Chinese Poems in Europeana[J]. International Journal of Digital Humanities, 2020, 2(1): 33-54.
- [19] Li J, Li X. The Knowledge Graphs for Chinese Poetry and Their Application: A Perspective from Digital Humanities[J]. Data and Information Management, 2021, 5(3): 271–284.