

目录

| | |
|------------------------|----|
| 第 1 章 项目概述..... | 1 |
| 1.1 项目背景..... | 1 |
| 1.2 项目开发意义..... | 1 |
| 1.3 本文组织结构..... | 1 |
| 第 2 章 系统需求..... | 3 |
| 2.1 引言..... | 3 |
| 2.1.1 编写目的..... | 3 |
| 2.1.2 背景..... | 3 |
| 2.1.3 定义..... | 4 |
| 2.1.4 参考资料..... | 5 |
| 2.2 任务概述..... | 5 |
| 2.2.1 目标..... | 5 |
| 2.2.2 用户的特点..... | 5 |
| 2.2.3 假定和约束..... | 6 |
| 2.3 需求规定..... | 6 |
| 2.3.1 对功能的规定..... | 6 |
| 2.3.2 对性能的规定..... | 15 |
| 2.3.2.1 精度..... | 15 |
| 2.3.2.2 时间特性要求..... | 15 |
| 2.3.2.3 灵活性..... | 16 |
| 2.3.3 输入输出要求..... | 16 |
| 2.3.4 数据管理能力要求..... | 17 |
| 2.3.5 故障处理要求..... | 17 |
| 2.3.6 其他专门要求..... | 19 |
| 2.4 运行环境规定..... | 20 |
| 2.4.1 设备..... | 20 |
| 2.4.2 支持软件..... | 21 |
| 2.4.3 接口..... | 23 |
| 2.4.4 控制..... | 24 |
| 第 3 章 系统设计..... | 26 |
| 3.1 引言..... | 26 |
| 3.1.1 编写目的..... | 26 |
| 3.1.2 背景..... | 26 |
| 3.1.3 定义..... | 26 |
| 3.1.4 参考资料..... | 26 |
| 3.2 总体设计..... | 27 |
| 3.2.1 需求规定..... | 27 |
| 3.2.2 运行环境..... | 27 |
| 3.2.3 基本设计概念和处理流程..... | 27 |
| 3.2.5 功能需求与程序的关系..... | 34 |
| 3.2.6 人工处理过程..... | 37 |
| 3.2.7 尚未解决的问题..... | 38 |
| 3.3 接口设计..... | 39 |
| 3.3.1 用户接口..... | 39 |
| 3.3.2 外部接口..... | 40 |
| 3.3.3 内部接口..... | 41 |
| 3.4 运行设计..... | 43 |
| 3.4.1 运行模块组合..... | 43 |

| | |
|--------------------------------|----|
| 3.4.2 运行控制 | 44 |
| 3.4.3 运行时间 | 46 |
| 3.5 系统数据结构设计 | 47 |
| 3.5.1 逻辑结构设计要点 | 47 |
| 3.5.2 物理结构设计要点 | 52 |
| 3.5.3 数据结构与程序的关系 | 53 |
| 3.6 系统出错处理设计 | 54 |
| 3.6.1 出错信息 | 54 |
| 3.6.2 补救措施 | 54 |
| 3.6.3 系统维护设计 | 55 |
| 第 4 章 系统评估 | 56 |
| 4.1 功能性属性验证 | 56 |
| 4.1.1 功能性属性验证的依据 | 56 |
| 4.1.2 功能性属性验证的主要内容 | 56 |
| 4.1.3 功能性属性验证的优先级 | 56 |
| 4.1.4 功能性属性验证的验收标准 | 57 |
| 4.1.5 功能性属性验证的测试方法 | 57 |
| 4.1.6 功能性属性验证的工具和方法 | 57 |
| 4.2 非功能性属性验证 | 57 |
| 4.2.1 性能需求验证 | 57 |
| 4.2.2 安全性需求验证 | 58 |
| 4.2.3 软件质量属性验证 | 58 |
| 4.2.4 业务规则验证 | 58 |
| 4.2.5 用户文档验证 | 59 |
| 4.3 其他需求验证 | 59 |
| 4.3.1 用户原创作品的版权保护 | 59 |
| 4.3.2 敏捷开发与 DevOps 思想的结合 | 59 |
| 4.3.3 用户协议与法律合规性 | 60 |
| 4.3.4 用户反馈与需求收集机制 | 60 |
| 4.3.5 系统日志与监控 | 60 |
| 4.3.6 系统兼容性与可扩展性 | 60 |
| 4.4 软件原型设计 | 61 |
| 4.5 软件运行界面 | 64 |

第 1 章 项目概述

1.1 项目背景

在信息化时代背景下，写作能力成为衡量一个人综合素质的重要指标之一。它不仅涉及到语言学科知识的整合，也是学生语言表达能力评估的重要组成部分，对学生学习其他学科具有重要的基础性作用。随着教育平台对创作课程的重视，专业老师对创作脉络、要素等知识点的讲解，对学生写作水平的提高起到了积极的推动作用。然而，写作能力的提高不仅仅依赖于专业老师的指导，还需要学生储备大量的词汇、经典诗文、名人名言等，以及对时事热点和民生百态的丰富收集，这是一个长期、循序渐进、与社会不断交互的过程。

在移动设备普及的今天，移动端学习平台应运而生，为学生的日常写作过程提供了便利。设计一款移动应用，陪伴学生的日常写作，具有极高的应用价值。本项目主题旨在通过人工智能技术，提高写作的效率和质量，生成契合文章主题内容的有吸引力的优质标题和摘要，开发出一套智能创作系统。

1.2 项目开发意义

本项目为智能创作平台应用，通过自然语言处理、自然语言生成、自然语言理解技术，能够自动生成与文章主题内容契合且具有吸引力的标题和摘要，极大地提高了写作的效率和质量。这不仅减少了人工处理的时间成本，也降低了由于主观性导致的误判风险。

该平台的开发和应用，将推动教育技术的发展，特别是在文章创作领域。通过智能化的工具辅助，学生可以更加专注于内容的创作和思想的表达，而不是被繁琐的标题和摘要撰写所困扰。

在新闻资讯创作、办公写作等场景中，用户可以利用该平台快速生成标题和摘要，提升工作效率。同时，合理的 UI 界面设计和友好的人机交互，使得用户在使用过程中获得更好的体验。

本平台的实现，将促进开源人工智能算法的应用和普及，推动人工智能技术在写作领域的应用，为未来的智能化写作提供更多的可能。

1.3 本文组织结构

第一章为项目概述，其中包含项目背景、项目开发的意义以及本文的组织结构说明。在项目背景部分，将详细介绍信息化时代背景下写作能力的重要性以及移动应用在辅助写作中的作用。项目开发意义部分则阐述本项目旨在通过人工智能技术提高写作效率和质量的目的。最后，组织结构部分则概述了本报告的布局，以便读者能够快速定位感兴趣的章节。

第二章系统需求，详细描述了软件需求的各个方面。首先，在引言部分阐明了编写需求说明书的目的和预期读者，接着任务概述部分描述了项目的目标、

用户特点以及项目的假设和约束。需求规定部分则详细列出了功能需求、性能需求、输入输出需求等。运行环境规定部分则说明了系统运行所需的设备、支持软件、接口和控制要求。

第三章系统设计，全面展示了软件体系结构的设计细节。引言部分提出了设计部分的目的和背景，总体设计部分描述了系统的总体架构和设计概念。接口设计部分详细说明了系统内部和外部的接口设计，运行设计部分讨论了系统的运行模块组合和控制机制。系统数据结构设计部分阐述了系统的逻辑和物理数据结构设计，系统出错处理设计部分描述了系统的出错信息、补救措施和维护设计。

第四章系统评估，对软件的功能性属性和非功能性属性进行了验证。功能性属性验证部分检查了软件是否满足需求规格说明书中定义的功能，非功能性属性验证部分则关注软件的性能、安全性等。其他需求验证部分包括了版权保护、敏捷开发与 DevOps 结合等其他需求的验证。软件原型设计部分展示了软件的原型设计和界面草图，软件运行界面部分则介绍了软件运行界面的实现和演示。

第五章自评成绩，项目成员根据自身的工作贡献和完成情况进行了自我评价。

第 2 章 系统需求

2.1 引言

2.1.1 编写目的

本节旨在阐明编写《易知智能创作平台软件需求说明书》的目的和预期读者。该文档作为项目开发和沟通的核心文件，具有多重重要目标：

明确需求：详细记录和描述易知智能创作平台的功能和非功能需求，确保开发团队、项目管理者、利益相关者和用户对系统的功能有一个清晰、一致的理解。

技术协议：作为用户和软件开发人员之间的技术协议书，明确双方对于软件产品的具体期望和承诺，为项目的顺利进行提供法律和技术上的依据。

设计和开发指导：为软件工程师和分析员提供详细的指导，确保设计工作能够准确反映用户的需求，并作为开发工作的基础和依据。

质量保证和测试基准：确立软件测试的准则和质量保证要求，帮助测试团队设计测试用例，验证软件是否满足文档中规定的需求。

维护和升级参考：为未来的软件维护和升级提供重要信息，使维护人员能够快速理解系统的设计和函数需求，便于进行必要的修改和优化。

沟通和协作平台：作为一个沟通工具，促进项目团队成员、用户以及第三方供应商之间的有效沟通和协作。

预期读者：

项目经理：利用该文档来理解产品的功能和目标，进行项目规划和管理。

设计和开发人员：依据该文档进行软件的架构设计和编码实现。

测试人员：根据需求说明书中的详细需求设计测试用例，确保软件质量。

运维人员：参考该文档进行软件部署后的维护和故障排除。

客户和最终用户：通过该文档了解软件的功能，判断是否满足其业务需求，并在项目后期提供反馈。

市场和销售团队：了解产品特性，进行市场推广和销售策略的制定。

2.1.2 背景

a. 待开发的软件系统的名称

本项目旨在开发一款名为“易知智能创作平台”的软件系统。该系统是一个以创作为主题的移动社交平台，专为小初高中生及创作爱好者设计，提供一个原创交流的平台，用户可以在该平台上汲取创作相关知识、锻炼写作能力、分享原创作品并获得激励、以及结交志同道合的笔友。

b. 本项目的任务提出者、开发者、用户及实现该软件的计算机中心或计算机网络

任务提出者：易知科技有限公司，一家涉足社交领域的企业，致力于通过技术创新提升教育体验。

开发者：由海南大学计算机科学与技术学院《软件体系结构》课程的专硕

软件工程班学生组成的开发团队，成员包括刘博文、刘俊、刘清扬、廖康顺和赵伟卿，由张雨老师指导。

用户：主要面向小初高中生及创作爱好者，包括但不限于学生、教师、家长以及任何对创作有兴趣的人士。

实现该软件的计算中心或计算机网络：软件将部署在云服务器上，以提高可用性和可扩展性，确保用户能够随时随地访问平台。服务器将采用 Windows Server 2008 及以上、Ubuntu 16 及以上、CentOS 7 及以上操作系统，以支持高并发和大数据量的处理需求。

c. 该软件系统同其他系统或其他机构的基本的相互来往关系

“易知智能创作平台”将与以下系统和机构发生交互：

第三方认证服务：为了实现单点登录功能，系统将集成第三方认证服务，如社交媒体账号登录等。

支付网关：平台内的付费内容和商城功能将需要与支付网关进行交互，以处理交易和支付流程。

内容审核系统：为了确保平台内容的合规性和安全性，系统将与内容审核系统对接，对用户上传的内容进行自动或人工审核。

数据分析和报告工具：系统将集成数据分析工具，以收集用户行为数据，用于改善用户体验和优化服务。

版权保护机构：为保护用户上传的原创作品版权，平台将与版权保护机构合作，确保用户权益得到合法保护。

2.1.3 定义

在本文档中，为了确保沟通的准确性和文档的清晰性，以下是一些专业术语和外文首字母缩写词的定义：

(1)AI (Artificial Intelligence): 人工智能，指由人制造出来的系统所表现出来的智能，通常由机器学习、自然语言处理等技术实现。

(2)API (Application Programming Interface): 应用程序编程接口，是软件系统中允许不同程序之间进行交互的一套规则和定义。

(3)DevOps: 开发与运维的组词，指的是一组过程、方法与系统的集合，用于促进开发、技术运营和质量保障部门之间的沟通、协作与整合。

(4)UI (User Interface): 用户界面，是人与计算机系统之间交流和交互的介面。

(5)UX (User Experience): 用户体验，指用户使用产品时的个人感受和态度，包括用户的情感、信仰、喜好、需求、感知、生理和心理反应等。

(6)OCR (Optical Character Recognition): 光学字符识别，是指通过扫描等技术将各种类型的文档转化为可由计算机识别和处理的格式。

(7)RDBMS (Relational Database Management System): 关系数据库管理系统，是一种基于关系模型的数据库管理系统。

(8)NOSQL (Not Only SQL): 非关系型数据库，指的是一系列旨在解决传统关系型数据库在某些场景下存在的限制的数据库管理系统。

(9)React Native: 一种用于开发移动应用的框架，允许使用 JavaScript 和 React 来开发原生应用。

(10)Vue: 一种构建用户界面的渐进式框架，用于构建单页应用。

(11)敏捷开发 (Agile Development): 一种以人为核心、迭代、循序渐进的软件开发方法。

(12)IEEE (Institute of Electrical and Electronics Engineers): 电气与电子工程

师协会，是一个国际性的电子技术与信息科学工程师的协会。

(13)鲁棒性 (Robustness): 指系统在异常和危险情况下生存的能力，即系统对错误输入、磁盘故障、网络过载或有意攻击等情况的抵抗能力。

(14)著作权法 (Copyright Law): 指保护文学、艺术和科学作品的作者的著作权的法律。

(15)UI/UX (User Interface/User Experience): 用户界面和用户体验的缩写，通常并用以强调软件设计中交互和视觉元素的重要性。

2.1.4 参考资料

[1].毋国庆,梁正平,袁梦霆,等.软件需求工程（第2版）[M].机械工业出版社,2019:1-247.

[2].[美]KarlWieggers,?Joy?Beatty.软件需求（第3版）[M].李忠利,李淳,霍金健,孔晨辉,译.清华大学出版社,2016:1-547.

[3].[美]JesseJames?Garrett.用户体验要素：以用户为中心的产品设计（原书第2版）[M].范晓燕,译.机械工业出版社,2011:1-164.

[4].[美]Joy Beatty, Chen A. 软件需求与可视化模型[M]. 第1版. 方敏, 朱嵘, 译. 清华大学出版社, 2017:1-359.

2.2 任务概述

2.2.1 目标

易知智能创作平台旨在为小初高中生以及创作爱好者提供一个创作原创移动交流平台。用户可以在该平台汲取创作相关知识、锻炼写作能力、分享原创作品并获得激励、以及结交志同道合的笔友。该平台的主要目标包括：

学习与创作：为用户提供丰富的创作素材和在线写作工具，帮助他们提升写作技能，创作高质量的创作。

社交互动：构建一个社区环境，让用户能够分享自己的原创作品，与其他创作者交流心得，互相激励，共同进步。

商业模式：通过提供免费的基础功能吸引大量用户，同时引入付费内容（如创作周边商品）以实现盈利，确保平台的可持续发展。

2.2.2 用户的特点

易知智能创作平台的用户群体涵盖了不同年龄段和兴趣层次的人群。核心用户包括小学生、初高中生和创作爱好者，他们对创作学习和创作有较高的需求，并且可能会频繁使用平台的各种功能。而大学生和其他用户则可能仅在特定情况下使用平台，社交需求相对较低。平台设计时应特别关注核心用户的体验，同时采取措施防止劣质用户带来的负面影响，确保社区环境的健康和安全。易知智能创作平台的目标用户群体及其特征如下：

重要用户:这些用户是平台的核心用户，对创作学习和创作有较高的需求，并且可能会频繁使用平台的功能。

(a) 小学生：创作和社交需求相对较低。主要由家长出于教育目的希望孩子通过该平台学习创作相关知识。只能使用家长的设备进行注册，操作通常在家长的监督下完成。可能需要更多的引导和支持，如家长或教师的帮助，以确保正确使用平台资源。

(b) 初高中生：超过半数的学生对创作相关的需求比较高，尤其是在准备

考试时。具有中等程度的社交需求，希望能够与同学或其他同龄人交流写作心得。普遍具有创作潜力，预测有自主注册的行为，会积极参与平台上的各种活动。可能会频繁使用在线写作、素材推荐等功能，并且乐于分享自己的作品，获取反馈和激励。

（c）创作爱好者：有强烈的创作学习和创作需求，对提高写作技能有很高的积极性。具有分享型的社交行为，愿意与其他创作者交流，寻找志同道合的朋友。属于自主注册的用户，可能会成为平台上的活跃成员，经常参与讨论、评价他人作品并上传自己的原创内容。

其他用户:这些用户虽然不是平台的主要目标群体，但也可能使用平台的部分功能。

（a）大学生：遇到创作方面的需求时，试图通过使用本产品获取对自己有利的知识。基于平时使用其他社交软件的情况，对本软件的社交需求不够明显，但仍属于自主注册。可能会偶尔使用平台的素材库或在线写作工具，但不太可能频繁参与社交互动。

（b）劣质用户：试图通过产品的社交功能进行广告或违法信息的散播以谋取利益。平台设计中需采取措施（如人脸识别）来限制此类用户的不当行为，保障平台的安全性和真实性。

2.2.3 假定和约束

a.技术约束：软件服务端必须使用 Java 语言编写，数据库需要用到 MongoDB、Redis、Neo4j、MySQL。Web 后台端使用 Vue 框架，移动端使用 React Native 框架。必须涉及到智能推荐算法和知识图谱的应用。编码需符合 Alibaba Java Coding Guidelines 编码规范和 JavaDoc 注释规范。

b.运行环境：服务器端、Web 后台端和移动客户端都有具体的硬件和操作系统要求。例如，服务器端需要 intel 酷睿系列 i5 四核 CPU、40G 硬盘、6GB 及以上内存，操作系统可以是 Windows Server 2008 及以上、Ubuntu 16 及以上、CentOS 7 及以上。

c.安全性要求：系统必须具备应对常见网络攻击的能力，敏感信息传输和存储需加密，接口调用需要安全认证，并实施严格的权限控制。

d.假设：文档所描述的软件需求获得了项目所有相关人员以及第三方评测机构的认可，且顶层需求基本稳定不变。如果客户提出新的需求，需求规格说明书将进行更新以反映这些变更。商城系统需要集成其他软件开发商提供的组件，假定这些组件能按时提供并满足需求，且与本系统的运行环境兼容。

2.3 需求规定

2.3.1 对功能的规定

易知智能创作平台的功能规定明确了系统应提供的各项功能及其具体要求，主要分为以下五个主要的功能模块：

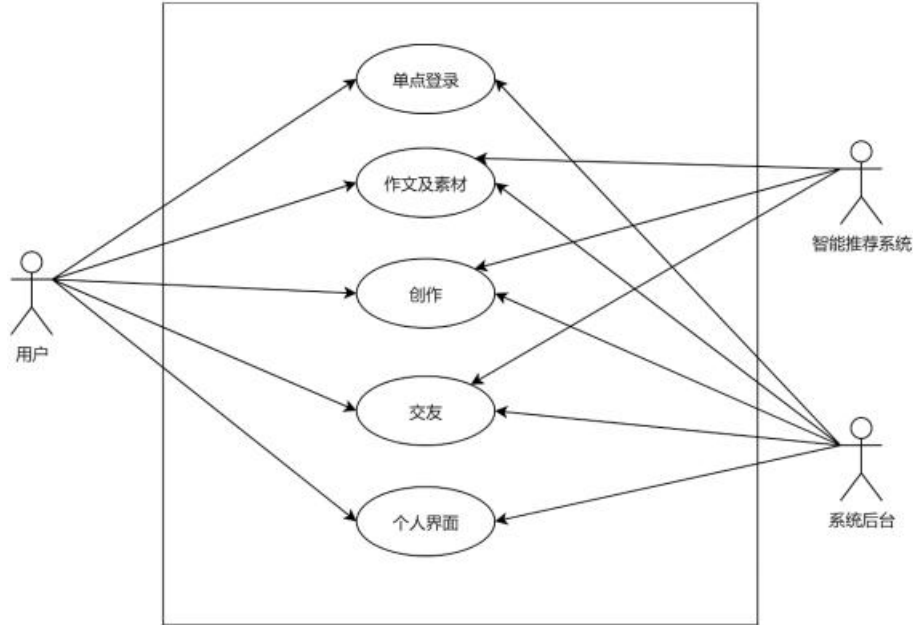


图 2.1 系统整体功能的用例图

a.单点登录功能（FC-1）

验证码登录：提供手机验证码登录功能，用户能通过验证码进行注册或登录。

个人资料补充：收集用户的性别、年龄、兴趣点、所在城市等基本信息，以便进行内容的模糊推荐。

人脸识别：增加人脸识别功能，以提高用户的真实性，防止劣质用户注册。

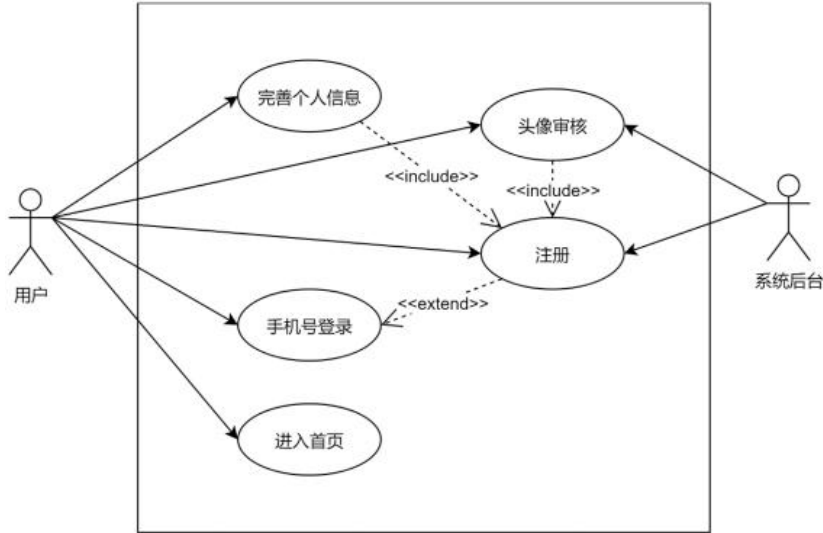


图 2.2 单点登录功能用例图

单点登录功能用例描述见表 2-1：

表 2-1 单点登录功能用例描述

| | | | |
|------|-----------|------|-----------|
| 用例名称 | 单点登录用例描述 | | |
| 相关信息 | | | |
| 所属模块 | FC-1 单点登录 | 相关需求 | RQ-1 单点登录 |
| 用例类型 | 功能测试 | | |

| | | |
|------|-------------------|--|
| 前置条件 | 用户打开作文纸条 APP 且未登录 | |
| 用例步骤 | | |
| 编号 | 步骤 | 预期 |
| 1 | 用户输入手机号，获取验证码 | 当用户获取验证码时，会进入验证码输入界面 |
| 2 | 用户输入验证码 | |
| 2.1 | 用户非法输入 | 应用继续等待输入验证码 |
| 2.2 | 用户输入六位验证码 | 验证码校对成功，应用跳转个人资料补充界面；验证码校对失败，应用继续等待输入验证码 |
| 3 | 用户进行个人资料补充 | |
| 3.1 | 用户上传头像 | 应用对人脸进行识别，若为人像则允许用户完成资料编辑；否则提示认证失败 |
| 3.2 | 用户完成填写其他个人信息 | 应用跳转到主页 |

b.创作相关功能（FC-2）

热点素材推荐：为用户提供最新的热点素材，帮助他们拓展视野，使文章更贴近生活。

分类素材推荐：根据用户的写作需求，提供分类素材搜索功能，便于用户动态查找所需材料。

创作推荐：根据用户的浏览历史和偏好，推荐优质的创作，帮助用户找到感兴趣的内容。

在线创作：提供在线写作工具，支持用户实时修改段落，并根据推荐的素材优化文章。

创作评价与优化：提交创作后，系统会自动进行优化建议和评价，帮助用户提升写作质量。

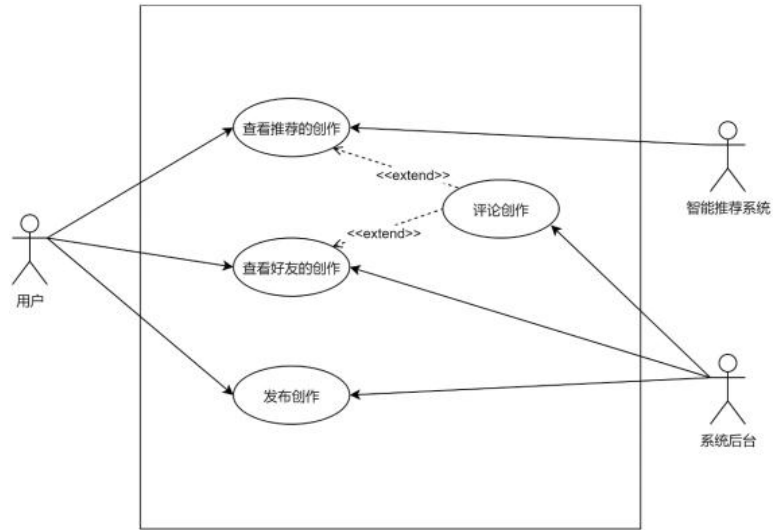


图 2.3 创作功能用例图

创作功能用例描述见表 2-2：

表 2-2 创作功能用例描述

| | | | |
|------|-------------------|----------------|-----------|
| 用例名称 | 创作功能用例描述 | | |
| 相关信息 | | | |
| 所属模块 | FC-3 创作功能 | 相关需求 | RQ-3 创作功能 |
| 用例类型 | 功能测试 | | |
| 前置条件 | 用户点击底部标签栏切换创作界面 | | |
| 用例步骤 | | | |
| 编号 | 步骤 | 预期 | |
| 1 | 推荐创作 | | |
| 1.1 | 用户滑动推荐创作子页面组件 | 页面显示创作内容列表 | |
| 1.2 | 用户点击某个创作项中的点赞按钮 | 点赞数+1，点赞按钮由灰变亮 | |
| 1.3 | 用户再次点击某个创作项中的点赞按钮 | 点赞数-1，点赞按钮由亮变灰 | |
| 1.4 | 用户点击评论按钮 | 页面弹出评论界面 | |
| 1.5 | 用户输入评论内容 | 评论数+1 | |
| 2 | 好友的创作 | | |
| 2.1 | 用户滑动关注创作页面 | 页面显示创作内容列表 | |

续表 2-2

| | | | |
|-----|-------------------|-----------------------|--|
| 编号 | 步骤 | 预期 | |
| 2.2 | 用户点击某个创作项中的点赞按钮 | 点赞数+1，点赞按钮由灰变亮 | |
| 2.3 | 用户再次点击某个创作项中的点赞按钮 | 点赞数-1，点赞按钮由亮变灰 | |
| 2.4 | 用户点击评论按钮 | 页面弹出评论界面 | |
| 2.5 | 用户输入评论内容 | 评论数+1 | |
| 3 | 发布创作 | | |
| 3.1 | 用户点击发布标签 | 页面切换至发布创作子标签页，显示标题框和创 | |

| | | |
|-----|----------------------|------------|
| | | 作编辑框 |
| 3.2 | 用户输入标题，输入创作内容，点击发布按钮 | 页面显示发布成功提示 |

c.社交功能（FC-3）

- 测试用户的喜好：通过问卷调查了解用户的兴趣范围，以便进行个性化内容推荐。
- 主动选择用户：提供随机用户推荐功能，用户可以通过左划右划来表示喜欢或不喜欢，拓展社交圈。
- 搜索附近用户：允许用户搜索附近的笔友，方便进行沟通和交流。
- 系统推荐相似用户：根据用户之间的相似度，推荐志同道合的笔友。
- 动态圈：用户可以发布和查看其他用户的动态，支持点赞、评论、收藏等功能。
- 创作圈：提供原创作品分享社区，用户可以上传自己的作品并获得激励。
- 短视频：支持用户制作和分享与创作相关的短视频，丰富创作形式。
- 消息中心：包括回复我的、@我的、点赞、私信、系统公告等分类，方便用户管理信息。

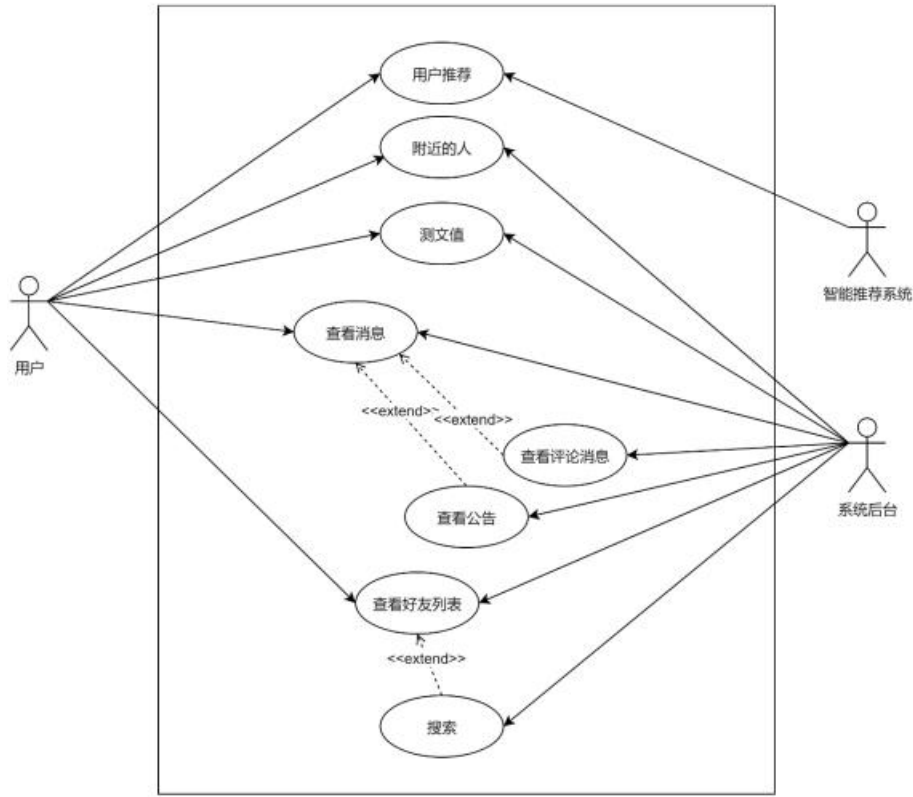


图 2.4 社交功能用例图

社交功能用例描述见表 2-3：

表 2-3 社交功能用例描述

| 用例名称 | 社交功能用例描述 |
|------|----------|
| 相关信息 | |

| | | | |
|-------|---------------------|-------------------------|-----------|
| 所属模块 | FC-4 社交功能 | 相关要求 | RQ-4 社交功能 |
| 用例类型 | 功能测试 | | |
| 前置条件 | 用户点击底部标签栏切换笔友界面 | | |
| 用例步骤 | | | |
| 编号 | 步骤 | 预期 | |
| 1 | 发现 | | |
| 1.1 | 用户点击测文值按钮 | 应用跳转到测文值界面并展示测试题 | |
| 1.2 | 用户点击附近的人按钮 | 应用跳转到搜附近页面，并展示附近用户头像和名称 | |
| 1.2.1 | 用户点击筛选按钮，改变筛选条件 | 界面展示对应结果 | |
| 1.3 | 用户点击今日最佳推荐组件 | 界面跳转至今日最佳推荐用户详情页 | |
| 1.4 | 用户点击推荐用户筛选按钮，改变筛选条件 | 界面展示对应结果列表 | |
| 1.5 | 用户点击推荐用户项 | 界面跳转至今日推荐用户详情页 | |
| 2 | 消息 | | |
| 2.1 | 用户切换到消息标签页 | 界面默认展示用户收到的所有消息 | |
| 2.2 | 用户点击消息类按钮 | 界面展示用户收到的对应类型的消息 | |
| 2.3 | 用户点击消息条目 | 应用跳转到消息详情界面 | |
| 3 | 联系人列表 | | |
| 3.1 | 用户切换到列表标签页 | 页面显示用户的联系人列表 | |
| 3.2 | 用户输入好友关键字搜索 | 页面显示好友搜索结果列表 | |
| 3.3 | 用户点击联系人列表子项 | 界面跳转至联系人信息详情页 | |

d.作文及素材相关功能（FC-4）

热点素材推荐：用户希望有热点素材推荐功能，有利于拓展视野，使文章与生活更加贴近。

开发分类素材功能：用户希望能增加分类素材功能，便于根据写作内容动态搜索相应素材。

作文推荐：用户希望能实现作文推荐功能，便于用户浏览自己喜欢类型的文章。

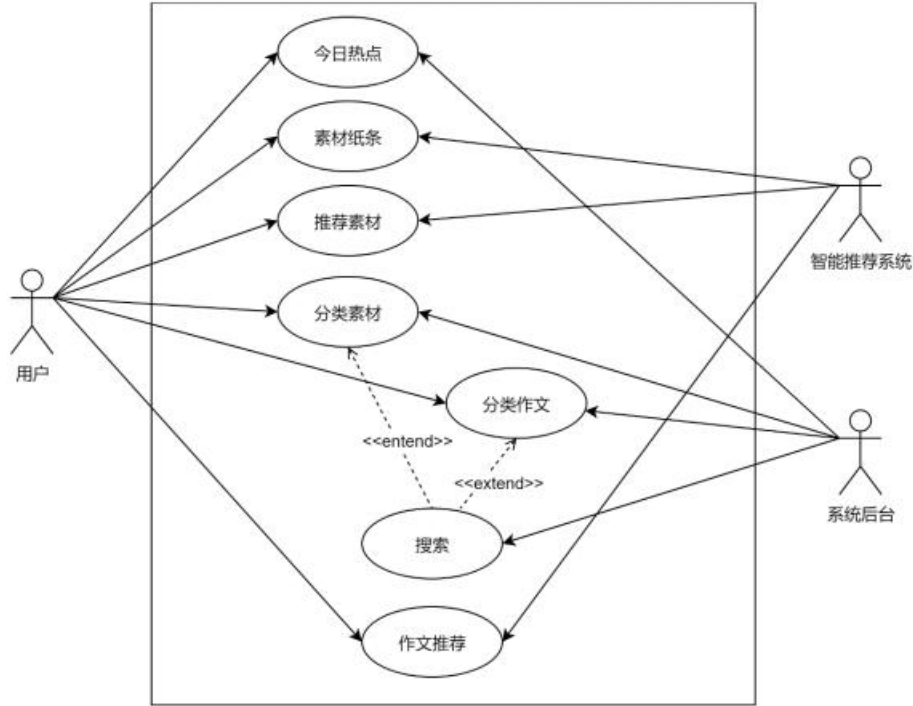


图 2.5 作文及素材相关功能用例图

作文及素材相关功能用例描述见表 2-4：

表 2-4 作文及素材相关功能用例描述

| 用例名称 | 作文及素材相关功能用例描述 | | |
|------|---------------------|-------------------|----------------|
| 相关信息 | | | |
| 所属模块 | FC-2 作文及素材相关功能 | 相关需求 | RQ-2 作文及素材相关功能 |
| 用例类型 | 功能测试 | | |
| 前置条件 | 用户点击底部标签栏切换到首页或作文界面 | | |
| 用例步骤 | | | |
| 编号 | 步骤 | 预期 | |
| 1 | 今日热点 | | |
| 1.1 | 用户点击今日热点菜单按钮 | 页面显示今日热点素材内容列表 | |
| 1.2 | 用户点击某个热点素材项 | 页面跳转至素材详情页 | |
| 2 | 素材纸条 | | |
| 2.1 | 用户点击纸条按钮 | 页面跳转至纸条子页面，显示素材纸条 | |
| 2.2 | 用户向左滑动素材纸条 | 该素材纸条被标记为用户不喜欢 | |

| | | |
|-----|---------------|-----------------------|
| 2.2 | 用户向右滑动素材纸条 | 该素材纸条被标记为用户喜欢 |
| 编号 | 步骤 | 预期 |
| 3 | 分类素材 | |
| 3.1 | 用户点击分类素材按钮 | 页面跳转至分类素材子页面，显示分类素材列表 |
| 3.2 | 用户选择某分类 | 页面显示相应分类素材列表 |
| 3.3 | 用户输入素材关键字搜索 | 页面显示素材搜索结果列表 |
| 3.4 | 用户点击某个素材项 | 页面跳转至素材详情页 |
| 4.3 | 素材推荐 | |
| 4.4 | 用户滑动推荐素材子页面组件 | 页面显示推荐素材内容列表 |
| 4.5 | 用户点击某个推荐素材项 | 页面跳转至素材详情页 |
| 5 | 作文推荐 | |
| 5.1 | 用户滑动推荐作文子页面组件 | 页面显示推荐作文内容列表 |
| 5.2 | 用户点击某个推荐素材项 | 页面跳转至素材详情页 |
| 6 | 分类作文 | |
| 6.1 | 用户切换至分类作文标签 | 页面切换至分类作文标签页，显示分类作文列表 |
| 6.2 | 用户选择某分类 | 页面显示相应分类作文列表 |
| 6.3 | 用户输入作文关键字搜索 | 页面显示作文搜索结果列表 |
| 6.4 | 用户点击某个作文项 | 页面跳转至作文详情页 |

e.我的界面（FC-5）

- 个人信息编辑：用户可以编辑和更新个人资料。
- 查看相互关注：用户可以管理自己的关注列表，增进与笔友的互动。
- 查看我的创作：用户可以查看自己发表的所有内容，包括动态、短视频和创作。
- 查看访客：用户可以查看谁浏览了自己的信息页，促进社交互动。
- 反馈功能：用户可以提交反馈信息，帮助开发团队改进产品。
- 系统设置：提供主题颜色切换、锁屏密码设置、第三方登录管理、注销等功能，增强用户体验。

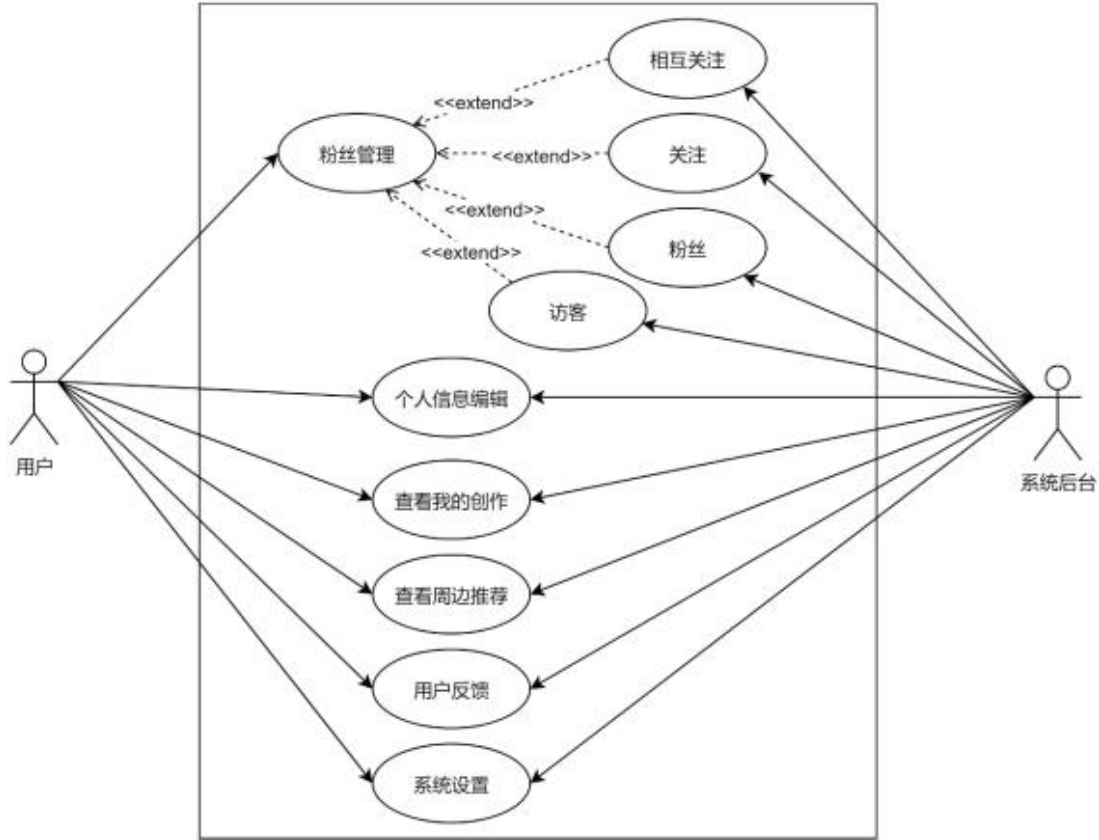


图 2.6 我的界面用例图

个人界面功能用例描述见表 2-5：

表 2-5 个人界面功能用例描述

| | | | |
|------|------------------|-----------------|-------------|
| 用例名称 | 个人界面功能用例描述 | | |
| 相关信息 | | | |
| 所属模块 | FC-5 个人界面功能 | 相关需求 | RQ-5 个人界面功能 |
| 用例类型 | 功能测试 | | |
| 前置条件 | 用户点击底部标签栏切换至个人界面 | | |
| 用例步骤 | | | |
| 编号 | 步骤 | 预期 | |
| 1 | 个人信息编辑 | | |
| 1.1 | 用户点击个人信息编辑 | 显示用户的基本信息，并提示修改 | |
| 1.2 | 用户点击提交 | 显示用户的信息提交结果 | |
| 2 | 粉丝管理 | | |

| | | |
|------|-------------|-------------------------|
| 2.1 | 用户点击相互关注按钮 | 显示相互关注的用户列表 |
| 2.2 | 用户点击关注按钮 | 显示关注的用户列表 |
| 2.3 | 用户点击粉丝按钮 | 显示用户的粉丝列表 |
| 2.4 | 用户点击访客按钮 | 显示访客列表 |
| 2..5 | 点击粉丝管理列表用户项 | 界面跳转至对应用户的具体信息页 |
| 编号 | 步骤 | 预期 |
| 3 | 查看我的创作 | |
| 3.1 | 点击查看我的创作按钮 | 显示我的创作列表 |
| 4 | 反馈功能 | |
| 4.1 | 点击反馈按钮 | 进入反馈信息页面 |
| 4.2 | 提交反馈信息 | 提示信息反馈情况 |
| 5 | 系统设置相关功能 | |
| 5.1 | 点击系统设置相关按钮 | 进入软件的基本设置选项页面，可进行注销等功能。 |
| 6 | 用户点击周边好物按钮 | 跳转至购物推荐独立子页面 |

2.3.2 对性能的规定

2.3.2.1 精度

a.数据准确性

系统中展示的所有数据必须准确无误。无论是用户的个人信息、创作内容、订单信息还是其他关键数据，都应确保其真实性和可靠性。用户在平台上提交的任何信息（如个人资料、创作、评论等）应被正确存储和显示。评价和反馈功能中的数据也应保持准确，避免误导其他用户。

b.内容完整性

系统需要包含当前需求所需的全部数据，以及可能用到的数据。这确保了用户在使用平台时能够获取完整的信息，避免因数据缺失影响使用体验。平台应提供全面的创作素材库，涵盖各类主题和风格，满足不同用户的创作需求。动态圈、创作圈等功能模块中的内容应保持更新，确保用户能够看到最新的作品和动态。

2.3.2.2 时间特性要求

a.响应速度

用户操作时，系统需要在 1 秒内实时返回准确结果，最多不超过 4 秒。这

确保了用户在使用平台时能够获得快速响应，提升整体体验。

页面加载、数据查询、提交表单等操作应在 1 秒内完成，最迟不超过 4 秒。在线写作、创作评价等需要后台处理的功能也应尽量缩短响应时间，避免用户等待过久。系统应具备高效的缓存机制，减少重复计算和数据传输的时间开销。

b.系统有效性

系统需要每天连续运行 24 小时，每周 7 天，包括公共假期和周末。这意味着平台必须具备高可用性，确保用户随时可以访问和使用。

系统应具备自动故障恢复机制，确保在发生硬件或软件故障时能够迅速恢复正常服务。定期进行系统维护和升级，但应选择在用户活跃度较低的时间段进行，以最小化对用户体验的影响。实施监控和报警系统，及时发现并处理潜在问题，确保系统的稳定运行。

2.3.2.3 灵活性

a.可扩充性

如果日后需要扩展数据，现有数据库要能够支持。这确保了系统能够随着用户增长和业务发展而平滑升级，不会因为数据量增加而出现性能瓶颈。

数据库设计应采用可扩展的架构，支持水平和垂直扩展，以应对未来数据量的增长。系统应具备良好的模块化设计，便于新增功能或修改现有功能，而不会影响整体性能。关键业务逻辑应与数据存储分离，确保在数据结构发生变化时，业务逻辑层不需要大规模修改。

b.容错性

正常按照需求内容，用户使用操作不会导致系统出错或者崩溃。即使发生异常情况，系统也应具备自我恢复能力，避免数据丢失或服务中断。

系统应具备完善的异常处理机制，捕获并记录所有异常情况，确保不会影响用户体验。对于可能导致系统崩溃的操作（如大文件上传、复杂查询等），应设置合理的限制和提示，防止用户误操作。实施定期备份策略，确保在发生数据丢失或损坏时能够快速恢复。

c.适应性

系统应能够适应不同的网络环境和设备类型，确保在各种条件下都能提供一致的用户体验。

支持多种网络协议（如 HTTP/HTTPS、WebSocket 等），确保在网络不稳定的情况下也能保持连接。优化移动端和 Web 端的适配，确保在不同屏幕尺寸和分辨率下都能正常显示和操作。提供离线模式，允许用户在网络断开时继续使用部分功能（如在线创作的离线保存），并在网络恢复后同步数据。

2.3.3 输入输出要求

输入输出要求确保了易知智能创作平台在与用户交互时的效率、准确性和用户友好性，同时也保护了用户数据的安全。

a.输入要求

用户输入：系统需要能够接收用户的基本信息，如手机号、验证码、个人资料（性别、年龄、兴趣点、所在城市等），以及在线创作时的创作内容。输入应通过简洁明了的界面进行，确保用户能够轻松地提供所需信息。

数据格式：用户输入的数据需要符合预定义的格式要求，例如手机号格式、验证码的数字格式等，以确保数据的有效性和准确性。

错误处理：系统应能够处理无效或错误的输入，提供清晰的错误提示，引导用户进行正确的输入。

b.输出要求

结果展示：系统需要在规定的时间内（1秒内，最多不超过4秒）向用户展示操作结果，如登录成功、创作提交反馈、搜索结果等。

数据准确性：输出的数据必须准确无误，无论是用户的个人信息、创作内容还是其他任何系统生成的数据。

界面友好性：输出的界面需要简洁大方，颜色柔和，功能按钮和关键字体系醒目，以提升用户体验。

实时反馈：系统需要提供实时反馈，如操作成功的提示、数据加载的进度显示等，让用户清楚地知道的操作结果。

安全性：对于敏感信息的输出，如用户个人信息，系统需要采取加密措施，确保数据传输和展示的安全性。

异常处理：系统在遇到异常或错误时，应提供明确的错误信息，并尽可能提供解决问题的途径或联系客服的选项。

多端一致性：由于系统包括服务器端、Web 后台端和移动客户端，输出的内容需要在不同端保持一致性，确保用户在不同设备上获得相同的体验。

2.3.4 数据管理能力要求

a.数据准确性要求

数据准确性：系统需求中需要展示的数据必须准确无误，以确保数据的可靠性。

b.数据管理响应时间

时间特性：用户操作时，系统需要在1秒内实时返回准确结果，最多不超过4秒，以提供快速响应。

c.数据管理灵活性

(a)扩展性的灵活性：数据库系统应该能够方便地进行扩展，以满足数据量增长和用户需求变化的要求，支持水平扩展和垂直扩展。

(b)数据安全的灵活性：数据库系统应该提供多种安全机制，以保护数据的机密性、完整性和可用性，包括用户认证、权限管理、数据备份和恢复，以及数据加密和审计功能。

(c)数据模型灵活性：数据库系统应该能够支持多种数据模型，如关系型、文档型、图形型等，以满足不同应用场景的需求。

(d)数据结构灵活性：数据库系统应该能够支持动态修改数据结构，包括添加、删除和修改表、字段等，以适应数据结构的调整和优化。

(e)数据访问灵活性：数据库系统应该提供灵活的数据访问方式，包括支持多种查询语言、查询方式和查询优化策略。

(f)数据存储灵活性：数据库系统应该能够支持多种数据存储方式，包括磁盘存储、内存存储和分布式存储等，以适应不同的存储需求。

(g)数据一致性和并发控制灵活性：数据库系统应该能够提供强大的一致性和并发控制机制，并允许根据实际需求选择不同的一致性和并发控制级别。

2.3.5 故障处理要求

1. 故障检测

a.实时监控

(a)系统性能监控：通过监控工具（如 Prometheus、Grafana 等）实时监测系统的 CPU 使用率、内存占用、磁盘 I/O、网络流量等关键指标，及时发现异常情况。

(b)应用日志监控：收集并分析应用的日志文件，识别错误信息、警告信息和异常行为。日志应包含详细的上下文信息，帮助开发人员快速定位问题。

(c)接口调用监控：对接口的响应时间、成功率、错误率等进行监控，及时发现 API 调用中的异常情况。特别关注高并发场景下的性能瓶颈。

(d)数据库连接监控：监控数据库的连接数、查询次数、慢查询等，确保数据库的稳定运行。对于长时间未响应的查询，应及时报警。

b. 自动化告警

(a)阈值设定：为各项监控指标设定合理的阈值，当指标超过阈值时，触发告警机制。例如，CPU 使用率超过 80%、内存占用超过 90%、接口响应时间超过 5 秒等。

多渠道通知：告警信息应通过多种渠道发送给相关人员，如短信、邮件、即时通讯工具（如钉钉、企业微信）等，确保告警信息能够及时传达。

(b)分级告警：根据故障的严重程度，设置不同的告警级别（如警告、严重、紧急），并采取相应的处理措施。例如，轻微问题可以通过自动恢复机制解决，而严重问题则需要人工介入。

2. 故障响应

a. 快速定位

(a)日志分析：当发生故障时，首先通过日志文件分析问题的根本原因。日志应包含详细的错误信息、堆栈跟踪、操作时间戳等，帮助开发人员快速定位问题。

(b)故障重现：如果可能，尝试重现故障场景，验证问题是否与特定的操作或环境有关。通过模拟用户的操作步骤，找出导致故障的具体原因。

(c)排查工具：使用调试工具（如 JProfiler、VisualVM 等）分析系统的运行状态，检查是否存在内存泄漏、线程死锁等问题。对于复杂的故障，可以使用分布式追踪工具（如 Jaeger、Zipkin）来跟踪请求的完整路径。

b. 临时应对

(a)降级服务：在故障无法立即修复的情况下，可以考虑启动降级服务，关闭部分非核心功能，确保系统的基本功能不受影响。例如，暂时关闭推荐算法、动态发布等功能，优先保证用户的登录、浏览等基本操作。

(b)限流与熔断：当系统负载过高或某个模块出现故障时，可以启用限流和熔断机制，限制请求的数量或直接拒绝某些请求，防止故障扩散。例如，当某个 API 的错误率超过一定比例时，自动熔断该 API，避免影响其他模块。

(c)缓存优化：增加缓存层，减少对后端服务的依赖。对于频繁访问的数据（如热点素材、热门创作），可以将其缓存到 Redis 中，减轻数据库的压力，提高系统的响应速度。

3. 故障恢复

a. 自动恢复

(a)重启服务：对于一些简单的故障（如进程崩溃、网络中断等），可以配置自动重启机制，确保服务能够在短时间内恢复正常。例如，使用 Supervisor 等工具监控服务的状态，当服务异常退出时，自动重启该服务。

(b)数据回滚：当数据库操作失败或数据不一致时，可以启用事务回滚机制，确保数据的一致性和完整性。对于长时间未完成的事务，可以设置超时机制，自动回滚以避免资源占用。

(c)重试机制：对于一些可重试的操作（如文件上传、消息推送等），可以在失败后自动重试，直到成功或达到最大重试次数。重试间隔应逐渐增加，避免短时间内频繁重试导致系统负载过高。

b.手动恢复

(a)备份恢复：当数据丢失或损坏时，可以从备份中恢复数据。备份策略应包括全量备份和增量备份，确保在不同情况下都能快速恢复数据。恢复操作应谨慎执行，确保不会覆盖最新的有效数据。

(b)代码修复：对于由代码缺陷引起的故障，开发人员应及时修复代码，并进行充分的测试，确保问题得到彻底解决。修复后的代码应尽快部署到生产环境，恢复正常服务。

配置调整：当故障是由配置错误引起的，可以调整相关配置项，如数据库连接池大小、缓存过期时间等，优化系统的性能和稳定性。

4. 预防措施

a.定期维护

(a)系统巡检：定期对系统进行全面巡检，检查硬件设备、操作系统、应用程序的状态，及时发现潜在问题。巡检内容包括服务器健康状况、磁盘空间、日志文件大小等。

(b)补丁更新：及时安装操作系统和应用程序的安全补丁，修复已知漏洞，防止黑客攻击。对于重要的安全补丁，应在测试环境中先行验证，确保不会影响现有业务。

(c)性能优化：定期分析系统的性能瓶颈，优化数据库查询、代码逻辑、网络通信等方面，提升系统的整体性能。可以使用性能测试工具（如 JMeter、LoadRunner）进行压力测试，评估系统的承载能力。

b.容灾演练

(a)灾难恢复计划：制定详细的灾难恢复计划，明确在发生重大故障时的应对措施和责任人。计划应包括数据恢复、服务切换、通知流程等内容，确保在紧急情况下能够迅速恢复正常服务。

(b)应急演练：定期进行应急演练，模拟各种故障场景，检验系统的容灾能力和团队的应急响应能力。演练内容可以包括网络中断、服务器故障、数据库损坏等常见故障，确保团队成员熟悉应急处理流程。

(c)异地备份：为了防止本地数据中心发生不可抗力事件（如火灾、地震等），应建立异地备份机制，将重要数据存储在不同的地理位置，确保即使本地数据中心受损，也能从异地恢复数据。

2.3.6 其他专门要求

用户在该平台上上传的所有原创作品应当受《中华人民共和国著作权法》相关规定保护，在网络平台上发表作品，版权理应归于原创作者所有。嵌入在用户协议中的“版权许可或转让”的内容是非法的，属于无效条款，所以在软件使用协议或条款中应注明相关条款，不得非法侵占用户权益。

此外，软件开发过程中应在敏捷开发的基础上增加 DevOps 的开发思想，透过自动化“软件交付”和“架构变更”的流程，来使得构建、测试、发布软件能够更加地快捷、频繁和可靠。

2.4 运行环境规定

2.4.1 设备

易知智能创作平台的运行环境规定中对服务器端、Web 后台端和移动客户端的设备提出了明确的要求，确保系统能够在不同类型的设备上稳定运行。通过合理的硬件配置和软件选择，平台能够提供高效、稳定的服务，满足用户的多样化需求。这些规定不仅有助于开发人员和运维人员选择合适的设备，也为系统的长期稳定运行提供了保障。以下是关于运行环境规定中设备规定的详细介绍：

1. 服务器端

服务器端是整个平台的核心部分，负责处理业务逻辑、数据存储和通信等任务。为了确保服务器端能够高效、稳定地运行，以下是对服务器端设备的具体规定：

a.硬件环境：

(a)CPU: Intel 酷睿系列 i5 四核或更高配置，确保足够的计算能力以应对高并发请求。

硬盘：典型安装需要至少 40GB 的存储空间，建议使用 SSD 固态硬盘以提高读写速度。

(b)内存：6GB 及以上，推荐 8GB 或更高配置，以支持多线程处理和缓存机制。

(c)网络：千兆以太网卡，确保高速稳定的网络连接，支持负载均衡和故障切换。

b.操作系统：支持的操作系统包括：Windows Server 2008 及以上,Ubuntu 16 及以上,CentOS 7 及以上.

c.数据库：

(a)MongoDB 3.4.4: 用于存储非结构化或半结构化的数据。

(b)Redis 3.2.1: 用于缓存频繁访问的数据，提升系统响应速度。

(c)Neo4j 4.2.3: 用于构建知识图谱，支持复杂的社交关系和推荐算法。

(d)MySQL 5.5 及以上：用于存储结构化数据，如用户信息、订单信息、商品信息等。

d.编程语言与框架：

(a)Java: 服务器端必须使用 Java 语言编写，确保跨平台兼容性和高性能。

(b)Spring Boot: 用于快速开发和部署微服务架构，简化配置和管理。

(c)Nginx: 作为反向代理服务器，负责负载均衡、静态资源分发和 SSL 加密等功能。

2. Web 后台端

Web 后台端是管理员和运营人员进行系统管理和操作的界面，主要通过浏览器访问。为了确保 Web 后台端在不同浏览器上的兼容性和性能，以下是对 Web 后台端设备的具体规定：

a.浏览器最低要求：IE 浏览器版本 12.0 及以上（不推荐使用 IE 浏览器，建议升级到 Edge 或其他现代浏览器）；Chromium 内核 78 及以上（如 Google Chrome、Microsoft Edge 等）；Firefox 68 及以上（可选，但建议使用）。

b.操作系统：支持的操作系统包括：Windows 7 及以上，macOS 10.13 及以上，Linux 各发行版。

c.屏幕分辨率：推荐屏幕分辨率为 1920x1080 或更高，确保界面布局清晰、

操作方便。

d.网络环境：稳定的互联网连接，带宽不低于 10Mbps，确保页面加载速度快，用户体验良好。

3. 移动客户端

移动客户端是用户与平台交互的主要方式，支持 Android 和 iOS 两大主流操作系统。为了确保移动客户端在不同设备上的兼容性和性能，以下是对移动客户端设备的具体规定：

a.手机最低配置要求：

(a)屏幕尺寸：5 英寸及以上，确保界面显示完整、操作便捷。

(b)操作系统：Android 7.0 及以上：支持最新的 Android 功能和 API，确保应用的稳定性和安全性。iOS 7.1 及以上：支持最新的 iOS 功能和 API，确保应用的稳定性和安全性。

(c)内存：2GB 及以上，推荐 4GB 或更高配置，确保应用运行流畅。

(d)处理器：八核或更高配置，确保应用在多任务处理时的性能。

(e)存储空间：16GB 及以上，确保有足够的空间安装应用和其他文件。

b.网络环境：

支持 Wi-Fi 和 4G/5G 移动网络，确保在网络条件不佳的情况下也能正常使用。

推荐使用 4G/5G 网络，确保低延迟和高带宽，提升用户体验。

c.传感器：

(a)GPS 模块：用于“搜附近”功能，帮助用户找到附近的笔友。

(b)摄像头：用于人脸识别和短视频录制，确保图像质量清晰。

(c)加速度计和陀螺仪：用于优化用户界面的交互体验，例如自动旋转屏幕。

2.4.2 支持软件

1. 服务器端

服务器端是整个平台的核心部分，负责处理业务逻辑、数据存储和通信等任务。为了确保服务器端能够高效、稳定地运行，以下是对服务器端支持软件的具体规定：

a.操作系统：Windows Server 2008 及以上，Ubuntu 16 及以上，CentOS 7 及以上。

b.编程语言与框架：

(a)Java：服务器端必须使用 Java 语言编写，确保跨平台兼容性和高性能。

(b)Spring Boot：用于快速开发和部署微服务架构，简化配置和管理。

(c)Maven/Gradle：用于项目构建和依赖管理，确保项目的自动化构建和持续集成。

c.数据库：

(a)MongoDB 3.4.4：用于存储非结构化或半结构化的数据，如用户动态、评论、点赞等。

(b)Redis 3.2.1：用于缓存频繁访问的数据，提升系统响应速度，如热点素材、热门创作、用户会话信息等。

(c)Neo4j 4.2.3：用于构建知识图谱，支持复杂的社交关系和推荐算法。

(d)MySQL 5.5 及以上：用于存储结构化数据，如用户信息、订单信息、商品信息等。

d.应用服务器：

- (a)Tomcat 8.5 及以上：用于部署 Java Web 应用程序，提供 HTTP/HTTPS 服务。
- (b)Jetty（可选）：轻量级的 Java Web 服务器，适用于微服务架构。
- e.反向代理与负载均衡：
 - (a)Nginx 1.10 及以上：作为反向代理服务器，负责负载均衡、静态资源分发和 SSL 加密等功能。
 - (b)HAProxy（可选）：用于高可用性和负载均衡，确保系统的稳定性和扩展性。
- f.消息队列：
 - (a)RabbitMQ 3.6 及以上：用于异步处理任务，如通知推送、日志收集等。
 - (b)Kafka 0.10 及以上：用于大规模分布式消息传递，支持高吞吐量和低延迟。
- g.日志管理：
 - (a)ELK Stack (Elasticsearch, Logstash, Kibana)：用于集中管理和分析日志，帮助快速定位问题。
 - (b)Graylog（可选）：另一个强大的日志管理解决方案，支持实时监控和告警。
- h.监控工具：
 - (a)Prometheus：用于监控系统性能指标，如 CPU、内存、磁盘 I/O、网络流量等。
 - (b)Grafana：用于可视化 Prometheus 采集的数据，生成实时监控图表。
 - (c)Zabbix（可选）：全面的监控解决方案，支持多种监控方式和告警机制。
- i.安全防护：
 - (a)WAF (Web Application Firewall)：用于防护常见的 Web 攻击，如 SQL 注入、XSS、CSRF 等。
 - (b)ClamAV：用于病毒扫描和恶意文件检测，确保上传内容的安全性。
 - (c)Fail2Ban：用于防止暴力破解攻击，自动封禁异常 IP 地址。
- 2. Web 后台端

Web 后台端是管理员和运营人员进行系统管理和操作的界面，主要通过浏览器访问。为了确保 Web 后台端在不同浏览器上的兼容性和性能，以下是对 Web 后台端支持软件的具体规定：

 - a.前端框架：
 - (a)Vue.js 2.6 及以上：用于构建用户界面，提供高效的组件化开发体验。
 - (b)Element UI：基于 Vue.js 的 UI 组件库，提供丰富的组件和样式，简化开发工作。
 - (c)Axios：用于发起 HTTP 请求，支持 Promise 和拦截器，方便处理 API 调用。
 - b.开发工具：
 - (a)Webpack：用于打包和优化前端资源，支持代码分割、懒加载等功能。
 - (b)ESLint：用于代码规范检查，确保代码质量和一致性。
 - (c)Prettier：用于代码格式化，保持代码风格统一。
 - c.浏览器最低要求：
 - (a)IE 浏览器版本 12.0 及以上（不推荐使用 IE 浏览器，建议升级到 Edge 或其他现代浏览器）
 - (b)Chromium 内核 78 及以上（如 Google Chrome、Microsoft Edge 等）
 - (c)Firefox 68 及以上（可选，但建议使用）
 - d.操作系统：
 - (a)Windows 7 及以上
 - (b)macOS 10.13 及以上
 - (c)Linux 各发行版

3. 移动客户端

移动客户端是用户与平台交互的主要方式，支持 Android 和 iOS 两大主流操作系统。为了确保移动客户端在不同设备上的兼容性和性能，以下是对移动客户端支持软件的具体规定：

a. 开发框架：

(a) React Native 0.63 及以上：用于开发跨平台移动应用，支持 iOS 和 Android，允许使用 JavaScript 和 React 构建原生应用。

(b) Expo（可选）：基于 React Native 的开发工具链，简化开发和调试过程，支持无配置启动项目。

b. 构建工具：

(a) Node.js 12.x 及以上：用于运行 React Native CLI 和其他开发工具。

(b) Yarn 或 npm：用于管理项目依赖，确保依赖项的一致性和可靠性。

(c) CocoaPods（仅 iOS）：用于管理 iOS 项目的依赖，简化第三方库的集成。

c. 开发环境：

(a) Android Studio 4.0 及以上：用于开发和调试 Android 应用，支持模拟器和真机调试。

(b) Xcode 11.0 及以上：用于开发和调试 iOS 应用，支持模拟器和真机调试。
操作系统：Android 7.0 及以上，iOS 7.1 及以上。

d. 网络通信：

(a) Axios 或 fetch：用于发起 HTTP 请求，支持 Promise 和拦截器，方便处理 API 调用。

(b) Socket.io（可选）：用于实现实时通信，如消息推送、在线状态同步等。

e. 多媒体处理：

(a) react-native-image-picker：用于选择和上传图片，支持相机和相册功能。

(b) react-native-video：用于播放视频，支持多种视频格式和控件。

(c) react-native-camera：用于访问设备摄像头，支持拍照和录制视频。

f. 地理位置：

(a) react-native-geolocation-service：用于获取设备的地理位置信息，支持 GPS 模块。

(b) react-native-maps：用于显示地图和标记位置，支持 Google Maps 和 Apple Maps。

g. 权限管理：react-native-permissions：用于请求和管理设备权限，如相机、麦克风、存储等。

h. 推送通知：

(a) Firebase Cloud Messaging (FCM)：用于实现推送通知功能，支持 iOS 和 Android。

(b) OneSignal（可选）：另一个流行的推送通知服务，支持多平台和自定义通知模板。

2.4.3 接口

接口是系统各组件之间以及系统与外部系统之间进行通信的桥梁。为了确保数据的正确传递和处理，以下是对接口的具体要求：

1. 用户界面（UI）接口

a. 启动界面：简洁大方，以最精简的文字描述软件的特性，给用户留下深刻印象。

b. 介绍界面：使用多个界面左右滑动以全面展示应用的功能概述，并提供跳过功能。

- c.登录界面：登录按钮醒目，第三方登录方式和条款置于页面底部。
 - d.主页：整体颜色一致，整体布局保持一致，底部导航栏醒目且便于切换子页面。
 - e.子页面：每个独立模块的功能应通过底部导航栏集中于每个独立的界面，模块的子功能应利用功能按钮或顶部导航标签进行跳转。
 - f.其他独立页面：如用到独立页面跳转，则页面顶部必须注明名称且提供返回按钮。
2. 硬件接口
- 无特殊需求：对于硬件接口，本项目没有特别的要求，主要依赖于移动设备的标准接口，如摄像头、GPS 等。
3. 软件接口
- a.OCR 识别与服务器接口：用于上传用户创作拍照识别信息。该接口应支持多种图片格式（如 JPEG、PNG），并能快速返回识别结果。
 - b.手机 APP 与推送信息服务器接口：用于给用户推荐创作信息及创作评价结果。该接口应支持实时推送，并能根据用户的偏好和行为进行个性化推荐。
 - c.数据库服务器与 APP 端接口：用于传送创作数据信息、用户信息等。该接口应支持高效的查询和更新操作，确保数据的一致性和完整性。
4. 通信接口
- a.HTTP/HTTPS：系统采用 HTTP/SSL 通信安全或加密，确保客户移动终端和服务器之间重要信息交换的安全性。所有敏感信息（如用户密码、支付信息等）都应通过 HTTPS 传输，防止中间人攻击。
 - b.WebSocket：用于实现实时通信功能，如消息推送、在线状态同步等。该接口应支持长连接，确保低延迟和高可靠性。
 - c.API 网关：作为统一的入口，负责路由、限流、认证等功能，确保 API 的安全性和稳定性。API 网关应支持版本管理、日志记录和监控功能，方便后续维护和扩展。

2.4.4 控制

控制机制是指系统为确保其正常运行而采取的各种管理和防护措施。为了保障系统的安全性、稳定性和性能，以下是对控制机制的具体要求：

1. 安全控制
- a.应对网络攻击：系统应具备常见的网络攻击防护能力，如防爬、防暴力破解、防 SQL 注入、防跨站脚本攻击（XSS）等。具体措施包括：
 - b.实施验证码验证，防止自动化工具的恶意攻击。
 - c.使用参数校验和输入过滤，防止 SQL 注入和 XSS 攻击。
 - d.配置防火墙和入侵检测系统，实时监控和阻止异常流量。
 - e.敏感信息传输：所有涉及敏感信息的传输和存储（如用户密码、支付信息、个人隐私数据等）都应采用加密技术，确保数据的安全性。常用加密算法包括 AES、RSA 等。
 - f.权限控制：系统应实施严格的权限控制，确保不同角色的用户只能访问和操作其授权范围内的数据。权限控制应覆盖所有功能模块，包括用户管理、创作管理、社交互动、商城管理等。特别是对管理员和运营人员的操作，应进行详细的日志记录，以便审计和追溯。
2. 性能控制
- a.负载均衡：通过 Nginx、HAProxy 等工具实现负载均衡，确保系统的高可用性和扩展性。负载均衡器应支持动态调整节点权重，根据服务器的健康状况自动分配流量。

b.缓存机制：使用 Redis 等缓存工具，减少对后端服务的依赖，提升系统的响应速度。对于频繁访问的数据（如热点素材、热门创作），可以将其缓存到 Redis 中，减轻数据库的压力。

c.限流与熔断：当系统负载过高或某个模块出现故障时，可以启用限流和熔断机制，限制请求的数量或直接拒绝某些请求，防止故障扩散。例如，当某个 API 的错误率超过一定比例时，自动熔断该 API，避免影响其他模块。

d.异步处理：对于一些耗时较长的操作（如文件上传、消息推送），可以采用异步处理机制，确保不会阻塞主线程，提升系统的并发处理能力。常用的异步处理工具包括 RabbitMQ、Kafka 等。

3. 数据控制

a.数据备份：定期进行数据备份，确保在发生意外情况时能够快速恢复数据。备份策略应包括全量备份和增量备份，以减少备份时间和存储空间的占用。数据备份应存储在安全的环境中，防止数据泄露。

b.数据清理：定期清理无效数据，如超过一定期限未登录的用户、过期的动态、被删除的创作等，以释放存储空间并提高系统性能。数据清理操作应谨慎执行，确保不会误删重要数据，并在清理前进行备份。

c.数据同步：确保不同数据库之间的数据一致性，特别是在跨数据库操作（如用户信息与订单信息的关联）时，避免数据丢失或不一致的问题。数据同步机制应支持双向同步，确保数据的完整性和一致性。

4. 操作控制

a.日志管理：系统应记录所有重要的操作日志，包括用户登录、注册、发布动态、提交创作、下单支付等。日志应包含操作时间、操作类型、操作对象、操作结果等信息，便于后续审计和问题排查。日志文件应定期归档，避免占用过多存储空间，并设置合理的保留期限。

b.版本控制：使用 Git 等版本控制系统，确保代码的版本管理和协作开发。每次发布新版本时，应进行全面的测试，确保新功能的正确性和稳定性。版本控制应支持回滚机制，允许在出现问题时快速恢复到之前的版本。

c.配置管理：通过配置文件或环境变量管理系统的各项配置项，如数据库连接字符串、API 密钥、缓存过期时间等。配置管理应支持热更新，允许在不重启服务的情况下修改配置项，提升系统的灵活性和可维护性。

第 3 章 系统设计

3.1 引言

3.1.1 编写目的

本节旨在作为用户与开发团队之间的技术协议，详细定义系统的功能和功能性需求，为系统设计、开发、测试及维护提供指导和依据。该文档特别强调了系统设计的要求，包括信息流、界面设计、功能特性、性能标准和安全措施，确保所有相关方对产品的特性和行为有一致的理解，并为后续的开发活动奠定基础。此外，它也为项目的验收提供了标准，保证最终产品满足最初定义的功能和性能要求，同时支持未来的软件维护和更新工作。

3.1.2 背景

本系统的背景在于应对当前教育市场中创作教学的需求变化，特别是在国家双减政策的大环境下，传统教育培训机构面临挑战，而线上教育平台则迎来了新的发展机遇。本项目旨在通过开发一款面向小初高中生及创作爱好者的移动社交平台——易知智能创作平台，来满足用户在创作学习和创作方面的需求，同时提供一个可以分享作品、交流心得并获得激励的社区环境。

从系统设计的角度来看，本节详细规划了该平台的技术框架和服务架构，以确保其能够高效稳定地运行，并为用户提供良好的体验。本节不仅定义了系统的整体功能模块，如单点登录、创作相关功能、社交互动、商城以及个人中心等，还深入探讨了各个功能的具体实现方式和交互逻辑。例如，单点登录机制的设计考虑到了提高用户体验的一致性和便利性；创作相关功能则结合智能推荐算法和知识图谱的应用，力求为用户提供个性化的内容和服务；社交功能强调了用户之间的连接与互动，鼓励原创内容的分享和传播；商城系统则集成了第三方组件，以支持商品交易的顺畅进行；最后，“我的界面”提供了个性化的设置选项，增强了用户的归属感和参与度。

为了保证系统的高性能和高可用性，本节对性能需求进行了明确规定，包括响应时间、连续运行能力、容错性和数据准确性等方面的要求。同时，在安全性方面，文档也提出了针对网络攻击的防护措施、敏感信息传输加密、接口调用认证以及权限控制等策略，以保护用户隐私和数据安全。

3.1.3 定义

见 2.1.3

3.1.4 参考资料

见 2.1.4

3.2 总体设计

3.2.1 需求规定

见 2.3

3.2.2 运行环境

见 2.4

3.2.3 基本设计概念和处理流程

从系统整体的角度分析用户的业务流程，用户通过单点登录后，可以访问应用的各个板块，具体如下图所示。

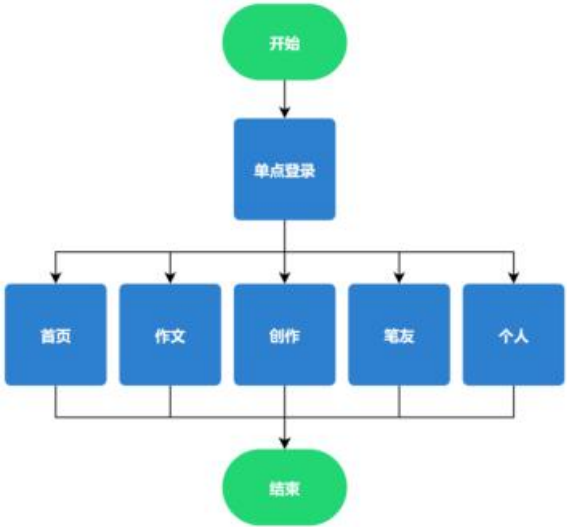


图 3-1 用户业务流程图

(1) 用户登录

用户从“开始”开始，首先进行“单点登录”尝试。如果单点登录失败，则需要“获取验证码并登录”。用户输入验证码后，系统会判断验证码是否正确。如果验证码正确，系统会进一步判断用户是否为新用户。如果是新用户，则需要“完善信息”，包括上传头像。系统会检查信息是否合法，如果信息合法，则用户成为注册用户，并跳转到“首页”。最后，流程以“结束”结束。

如果在任何一步出现问题，流程会回到相应的错误处理步骤，例如验证码错误会回到“获取验证码并登录”步骤，信息不合法会回到“完善信息”步骤。

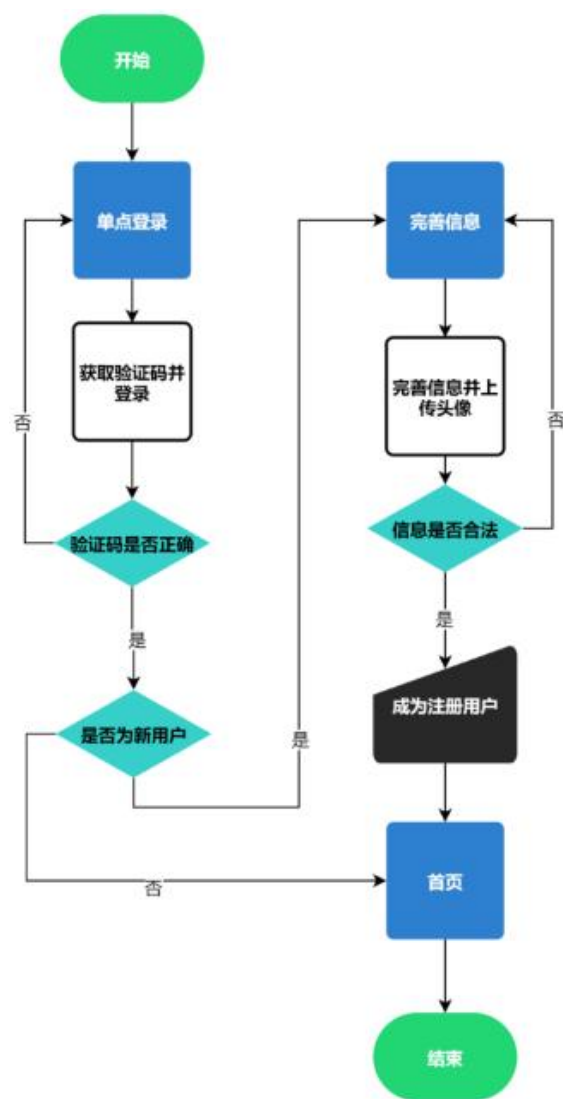


图 3-2 用户登录功能业务流程图

(2) 创作功能

下图展示了一个创作功能的业务流程。流程从“开始”开始，用户进入“创作”功能模块。用户可以选择不同的功能模块进行操作，包括“素材卡片”、“满分创作”、“分类素材”、“编辑”和“上传”。在“素材卡片”模块，用户可以查看推荐素材；在“满分创作”模块，用户可以查看高分创作；在“分类素材”模块，用户可以查看分类素材；在“编辑”模块，用户可以编辑创作；在“上传”模块，用户可以上传创作。

用户还可以选择“跳转在线写作界面”进行创作创作，或者“跳转其他页面”进行其他操作。此外，用户可以选择“查看今日热点”来查看今日热点信息。

在完成相关操作后，用户可以返回查看推荐素材、查看创作、查看分类素材或查看作品。最终，流程以“结束”结束。

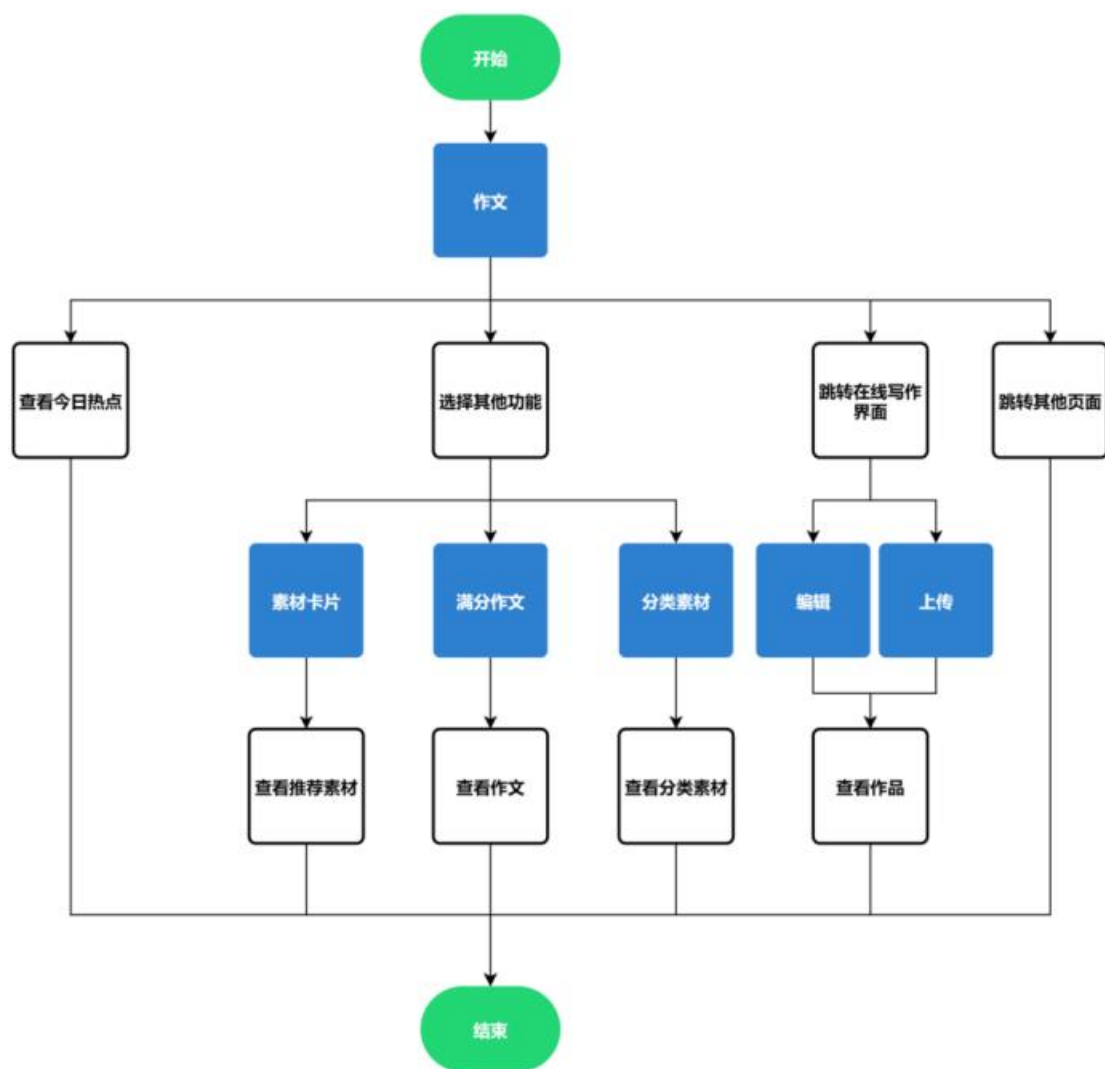


图 3-3 创作功能业务流程图

(3) 社交功能

下图展示一个创作功能的业务流程。流程从“开始”开始，用户进入“创作”功能模块。用户可以选择不同的功能模块进行操作，包括“素材卡片”、“满分创作”、“分类素材”、“编辑”和“上传”。在“素材卡片”模块，用户可以查看推荐素材；在“满分创作”模块，用户可以查看高分创作；在“分类素材”模块，用户可以查看分类素材；在“编辑”模块，用户可以编辑创作；在“上传”模块，用户可以上传创作。

用户还可以选择“跳转在线写作界面”进行创作创作，或者“跳转其他页面”进行其他操作。此外，用户可以选择“查看今日热点”来查看今日热点信息。

在完成相关操作后，用户可以返回查看推荐素材、查看创作、查看分类素材或查看作品。最终，流程以“结束”结束。

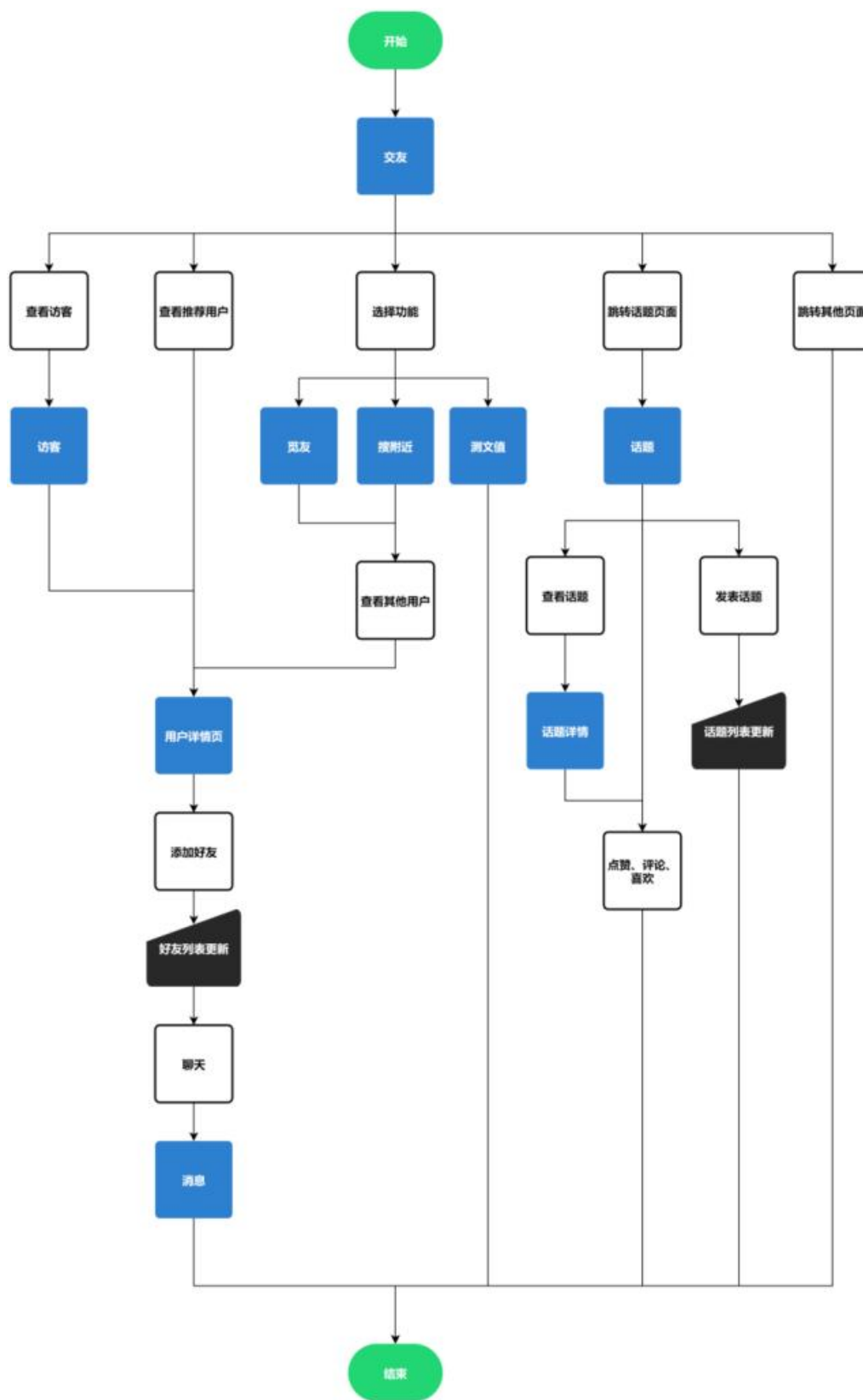


图 3-4 社交功能业务流程图

(4) 我的界面功能

下图展示了一个社交应用中“我的页面”业务流程。流程从“开始”开始，用户进入“交友”功能模块。用户可以选择多种操作路径：

用户可以“查看访客”，查看谁访问了自己的页面。

用户可以“查看推荐用户”，查看系统推荐的用户。

用户可以“选择功能”，进一步细分到“觅友”、“搜附近”和“测文缘”：

在“觅友”中，用户可以查看其他用户。

在“搜附近”中，用户可以搜索附近的用户。

在“测文缘”中，用户可以进行文字缘分测试。

用户还可以选择“跳转话题页面”，查看和参与话题讨论。

用户也可以选择“跳转其他页面”，进行其他操作。

在“查看其他用户”后，用户可以进入“用户详情页”，进一步查看用户的详细信息。用户可以选择“添加好友”，添加该用户为好友，这将触发“好友列表更新”。之后，用户可以进行“聊天”和查看“消息”。

在“跳转话题页面”后，用户可以“查看话题”或“发表话题”。查看话题后，用户可以进入“话题详情”，并进行“点赞、评论、喜欢”操作，这将触发“话题列表更新”。

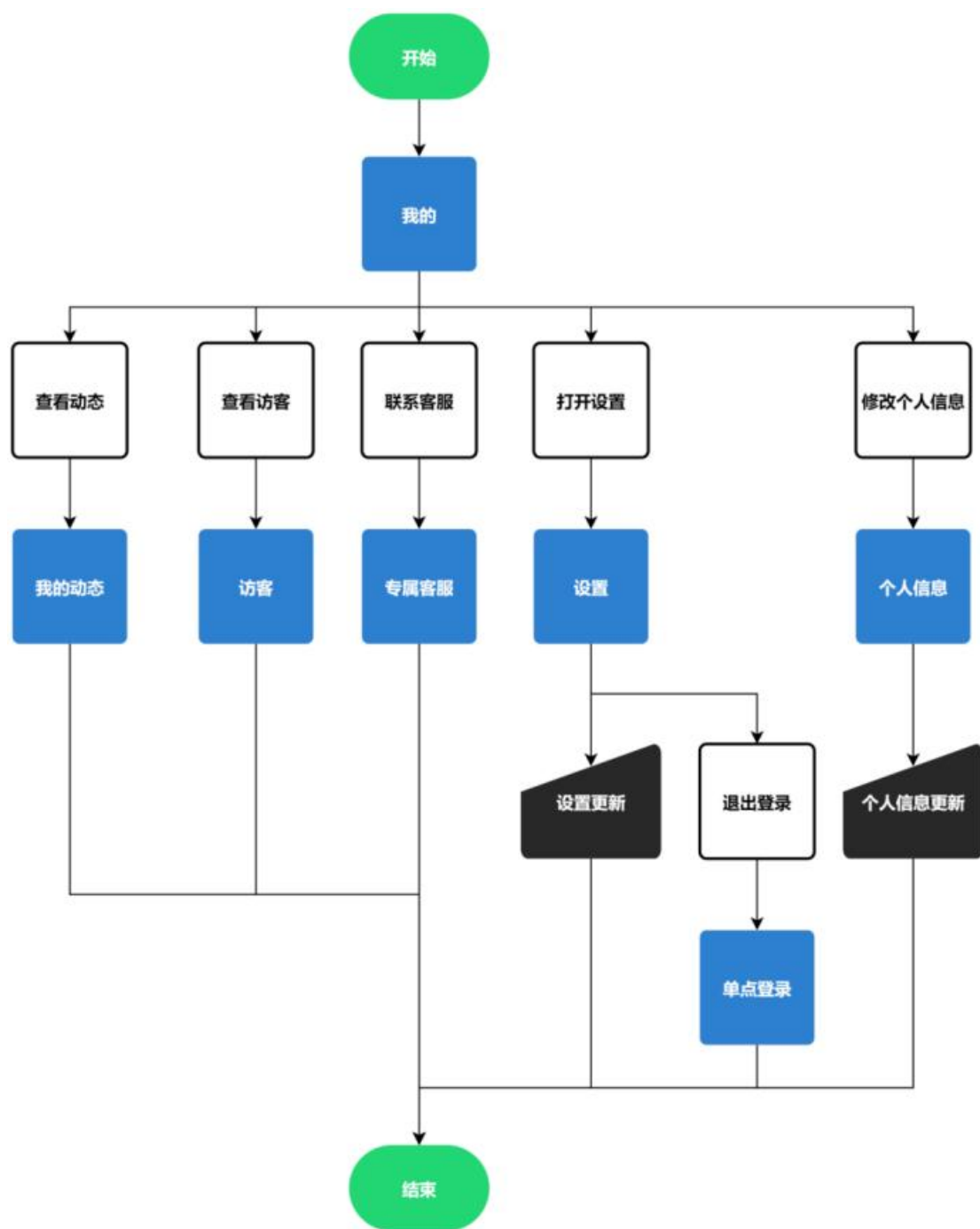


图 3-5 个人界面功能业务流程图

3.2.4 结构

易知智能创作平台的总体架构可以分为以下几个主要层次：

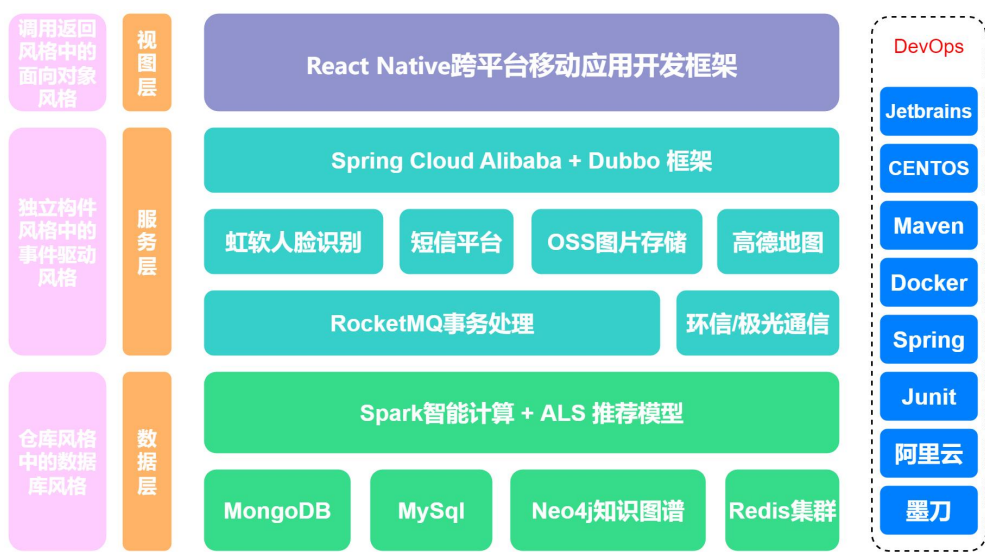


图 3-6 系统技术架构图

- a.展示层（Presentation Layer）：
包括移动客户端（iOS 和 Android）、Web 后台端。
提供用户界面，处理用户的输入输出，展示信息给用户，并与业务逻辑层进行交互。
移动客户端使用 React Native 框架实现跨平台支持；Web 后台端采用 Vue 框架开发。
- b.业务逻辑层（Business Logic Layer）：
负责处理核心业务规则和流程，如单点登录验证、创作推荐算法、社交互动机制等。
每个业务逻辑单元对应于特定的功能模块，如创作管理、社交网络、商城服务和个人中心等。
通过 API 接口与其他层通信，确保数据的一致性和安全性。
- c.数据访问层（Data Access Layer）：
包含数据库操作，用于存储和检索用户数据、创作内容、社交记录及商品信息等。
使用多种数据库技术来优化不同类型的数据存储：MongoDB、Redis、Neo4j 和 MySQL。
支持智能推荐算法和知识图谱的应用，以增强个性化服务。
- d.安全与认证层（Security and Authentication Layer）：
实施全面的安全措施，包括但不限于防爬虫、防暴力破解、SQL 注入防护、跨站脚本攻击（XSS）防御等。
管理用户的身份验证和权限控制，保证敏感信息的安全传输和保存。
- e.外部接口层（External Interface Layer）：
处理与第三方服务或系统的集成，例如支付网关、短信服务提供商等。
提供必要的 API 接口，以便其他应用程序或服务可以调用平台提供的功能。

- f.消息推送层（Notification Layer）：
负责向用户发送通知，如新消息提醒、订单状态更新等。
支持实时推送和延迟推送两种模式，确保消息及时准确地传达给用户。
- g.运维监控层（Operations Monitoring Layer）：
监控整个系统的运行状况，包括服务器性能、网络流量、应用错误日志等。
设立故障报警机制，一旦检测到异常情况立即通知相关人员采取行动。

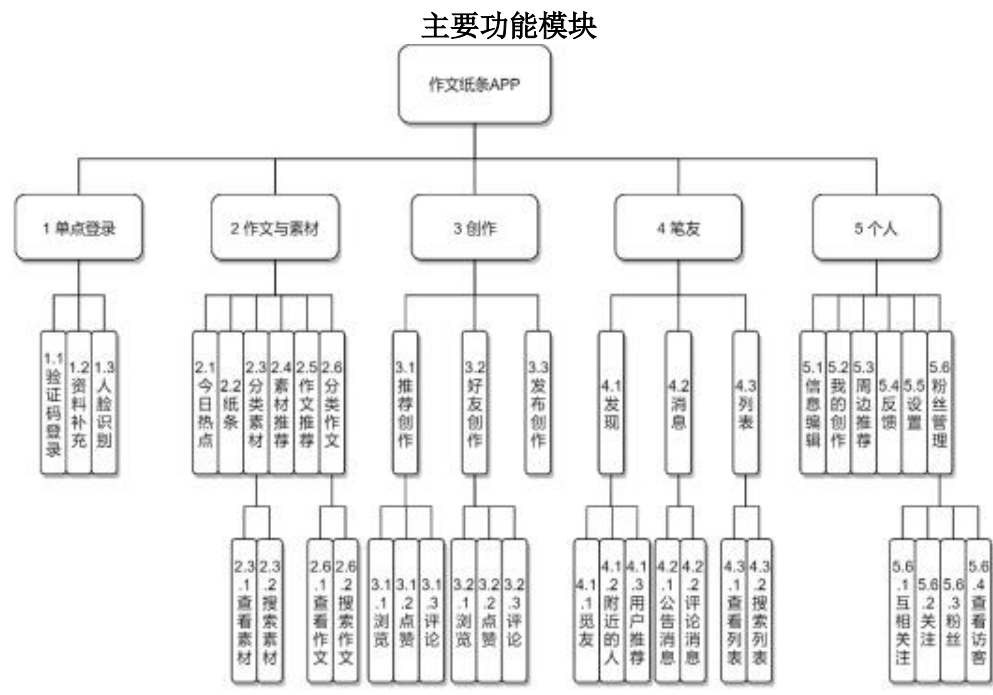


图 3-7 系统功能模块图

- a.单点登录模块（SSO Module）：提供统一的身份验证入口，简化用户的登录体验。
- b.创作管理模块（Composition Management Module）：支持使用大语言模型对内容生成摘要，创作的创建、编辑、评价等功能，同时提供素材推荐和在线创作工具。
- c.社交互动模块（Social Interaction Module）：构建用户之间的交流平台，包括动态发布、评论、点赞、私信聊天等社交行为。
- d.商城服务模块（E-commerce Service Module）：为用户提供购买创作相关周边商品的服务，包含商品分类、搜索、购物车、订单管理和支付等功能。
- e.个人中心模块（Personal Center Module）：允许用户管理个人信息、查看关注列表、编辑已发表的作品以及提交反馈意见等。

3.2.5 功能需求与程序的关系

- a. 单点登录（SSO）
功能需求：提供验证码登录、个人资料补充以及人脸识别等功能，增强账户安全性。

身份验证服务：处理用户注册、登录请求，生成并验证验证码，管理会话状态。

用户信息管理系统：收集和存储用户基本信息，支持用户资料编辑和更新。

人脸识别 API：调用第三方或自建的人脸识别服务，用于验证用户身份的真实性。

下面用相关类图展示单点登录模块的逻辑实现：

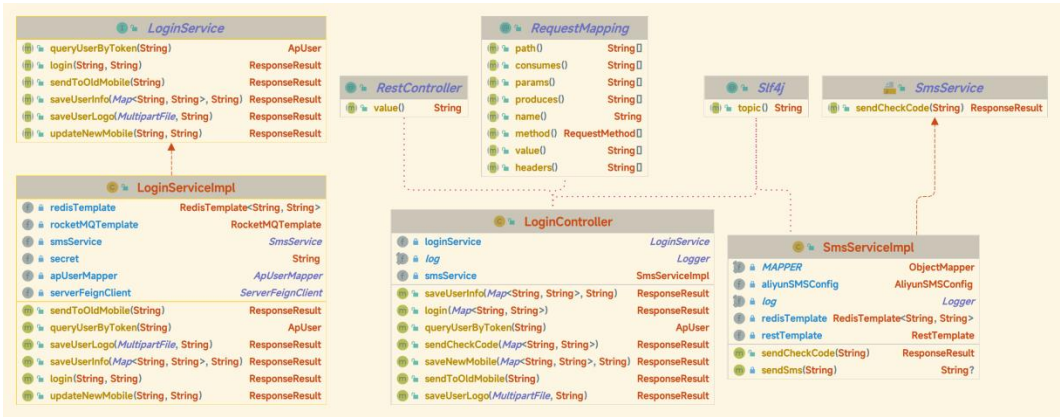


图 3-8 单点登录模块相关类图

b. 创作相关功能

功能需求：包括热点素材推荐、分类素材搜索、创作推荐、在线创作及优化评价等。

程序组件：

智能推荐引擎：基于用户的兴趣偏好和历史行为数据，为用户提供个性化的素材和创作推荐。

内容管理系统：管理和维护创作库，支持文章的创建、编辑、发布和删除操作。

在线编辑器：提供富文本编辑环境，允许用户实时预览并保存创作草稿。

创作评估工具：集成自然语言处理技术，对提交的创作进行自动评分和改进建议。

下面用相关类图展示创作模块的逻辑实现：

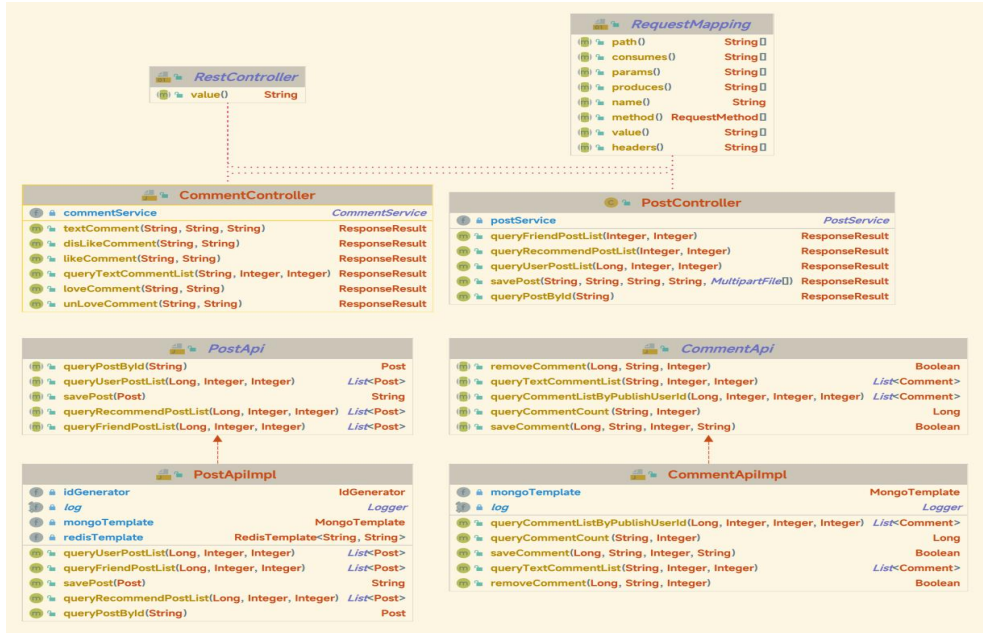


图 3-9 创作模块相关类图

c. 社交功能

功能需求：支持用户之间建立联系，分享动态，参与社区讨论，上传短视频等。

程序组件：

社交网络服务：构建用户关系图谱，管理好友列表，支持私信聊天和个人主页展示。

动态流管理：负责动态内容的发布、点赞、评论等功能，确保信息流的及时性和准确性。

视频处理服务：提供短视频的录制、上传、转码和播放支持，保证用户体验流畅。

d. 商城功能

功能需求：提供商品浏览、搜索、加入购物车、订单创建、支付等功能。

程序组件：

商品目录系统：维护商品信息，支持分类检索和详情页展示。

购物车管理：实现商品添加、数量调整、结算等功能，确保购物流程便捷。

订单处理系统：接收并处理用户的购买请求，生成订单记录，跟踪配送进度。

支付网关接口：集成多种支付方式，如支付宝、微信支付等，确保交易安全可靠。

e. 我的界面

功能需求：包含个人信息编辑、查看相互关注、我的创作、访客记录、反馈意见、系统设置等。

程序组件：

用户配置中心：提供个性化设置选项，如主题颜色切换、锁屏密码设定等。

互动统计面板：汇总用户的社交活动数据，如关注者数、粉丝数、获赞数等。

作品管理后台：帮助用户管理和整理自己的原创作品，支持编辑、删除和重新发布。

客服与反馈渠道：建立用户与开发团队之间的沟通桥梁，收集问题报告和改进建议。

下面用相关类图展示社交和个人界面模块的逻辑实现：

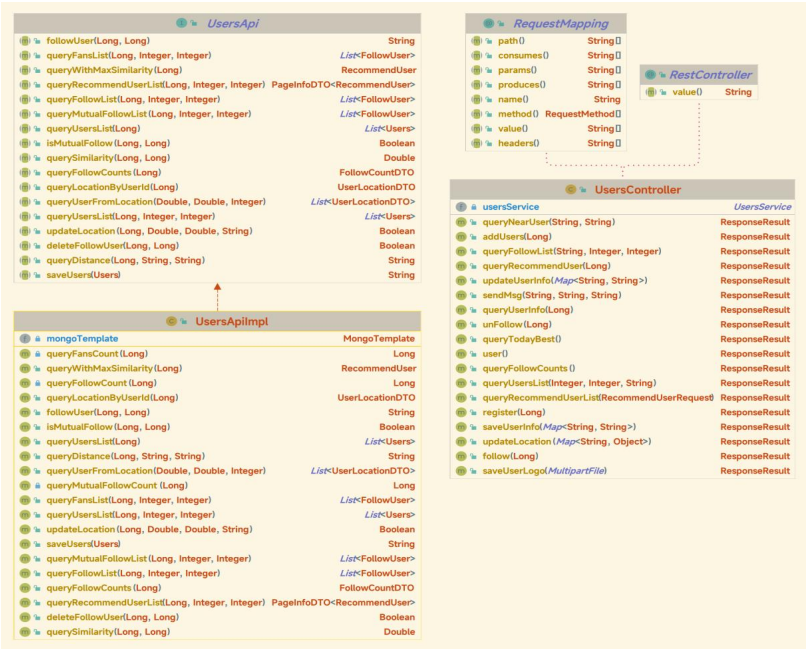


图 3-10 社交和个人界面模块相关类图

功能需求与程序的关系总结:

- a.一致性：确保每个功能模块的设计都严格遵循需求文档中的描述，从用户界面到后台逻辑均保持一致。
- b.模块化：将不同功能拆分为独立但又相互关联的服务或组件，便于单独开发、测试和部署。
- c.集成性：通过 API 接口和服务总线等方式，让各个模块可以无缝对接，形成一个完整的工作流。
- d.扩展性：考虑到未来可能新增的功能或业务变化，设计时预留足够的灵活性和扩展空间。
- e.安全性：在所有功能中嵌入必要的安全措施，保护用户隐私和系统稳定运行。

3.2.6 人工处理过程

a. 需求收集与分析

人工处理：项目经理或产品经理需要通过访谈、问卷调查、市场调研等方式，收集潜在用户的需求，并对这些需求进行整理和优先级排序。
重要性：确保最终产品能够准确地满足目标用户的期望，避免因需求不明确而导致的项目偏差。

b. 设计评审

人工处理：由开发团队、测试团队及业务专家组成的评审小组，对初步设计方案进行审查，提出改进建议，确保方案的可行性和合理性。
重要性：通过多方参与的设计评审，可以提前发现潜在问题，优化设计，减少后期返工的可能性。

c. 测试验证

功能测试：测试人员编写详细的测试用例，手动执行测试以验证每个功能是否按照需求文档正确实现。

用户体验测试：邀请真实用户参与体验，收集他们的意见和建议，评估系统的易用性和满意度。

重要性：保证产品质量，及时发现并修复缺陷，提高用户满意度。

d. 用户反馈处理

人工处理：设立专门的客服团队，负责接收并处理用户的咨询、投诉和建议；定期汇总用户反馈，向产品研发团队提供改进建议。

重要性：持续改进产品，增强用户体验，提升用户忠诚度。

e. 日常运维

监控与报警：运维人员实时监控服务器性能、网络流量等关键指标，一旦出现异常情况立即触发报警机制，并采取相应措施。

数据备份与恢复：定期进行数据库备份，制定灾难恢复计划，确保数据安全。

版本更新与维护：根据业务发展和技术进步，适时推出新版本，修复已知漏洞，添加新功能。

重要性：保障系统的稳定运行，防止意外停机和服务中断，为用户提供可靠的服务。

f. 内容审核

人工处理：对于用户上传的内容（如创作、评论、短视频等），需要安排专人或借助 AI 辅助工具进行审核，确保内容合法合规，符合社区规范。

重要性：维护良好的社区环境，防范不良信息传播，保护用户权益。

g. 安全事件响应

人工处理：建立应急响应团队，针对可能出现的安全威胁（如黑客攻击、数据泄露等）制定应急预案，快速响应并解决问题。

重要性：保护用户隐私和数据安全，维护公司声誉。

h. 商城商品管理

人工处理：商城管理员负责上架新品、调整库存、处理订单以及协调物流配送等工作，确保商品信息准确无误，交易流程顺畅。

重要性：保证商城服务的质量，促进销售额增长。

3.2.7 尚未解决的问题

尚未解决的问题如下表所示

| ID | 关键词 | 问题描述 |
|------|-----------|--|
| UQ-1 | 用户激励形式 | 给用户原创内容提供激励是考虑用积分还是代币仍需商榷 |
| UQ-2 | 用户名 | 在已有手机号登陆的情况下，用户名登陆是否还有必要 |
| UQ-3 | 商城管理后台的开发 | 本文档仅列出了商城客户端的需求，那么对于商城管理者来说，商城后台的需求还暂未确定 |
| UQ-4 | 内容审核 | 对用户上传的内容进行合法性审核 |

3.3 接口设计

3.3.1 用户接口

a. 启动与介绍界面

启动界面：展示品牌标志或应用特性简短描述，快速加载后自动跳转至介绍界面。

介绍界面：采用多页滑动形式，全面展示应用的主要功能和优势，提供“跳过”按钮方便用户快速进入主程序。

b. 登录/注册界面

登录界面：

手机号码输入框：用于输入用户的手机号。

验证码输入框：接收通过短信发送的验证码。

第三方登录选项：如微信、QQ、微博等快捷登录方式。

忘记密码链接：引导用户找回账户信息。

登录按钮：醒目的大按钮，点击后进行身份验证。

注册界面：

基本信息填写：包括性别、年龄、兴趣点、所在城市等，帮助系统更好地了解用户。

人脸识别认证：增加安全性，防止劣质用户的注册。

完成注册按钮：提交所有必要信息后，创建新账户。

c. 主页

顶部导航栏：

包含搜索框，方便用户查找特定内容。

设置个人中心入口，直接访问“我的”页面。

底部导航栏：

固定显示，包含四个主要标签：“首页”、“创作”、“社交”、“我的”，便于用户在不同功能之间切换。

内容区域：

显示推荐的热点素材、最新动态、精选创作等内容，吸引用户浏览。

d. 创作相关功能界面

创作创作编辑器：

提供富文本编辑功能，支持插入图片、链接等元素。

实时保存草稿，避免因意外断网而丢失数据。

推荐相似好词好句及素材，辅助用户创作高质量创作。

创作优化与评价结果展示：

展示系统给出的评分和改进建议，帮助用户提升写作水平。

e. 社交功能界面

交友模块：

测文值：通过回答一系列问题来了解用户的兴趣偏好。

随机匹配：左划右划选择是否喜欢随机推荐的用户。

搜附近：基于地理位置找到附近的笔友。

动态圈：

用户可以发布文字、图片或短视频动态，并与其他用户的动态互动（点赞、评论）。

创作圈：

分享原创作品，获得社区内其他用户的反馈和支持。

消息中心：

汇总来自私信、评论、点赞等多种类型的通知，保持用户活跃度。

f. 商城界面

商品分类展示：

以卡片式布局呈现各类商品，每张卡片包含商品图片、名称、价格等信息。

商品详情页：

更详细的描述，包括用户评价、库存状态等。

购物车管理：

查看已选商品，调整数量，一键结算。

订单跟踪：

显示订单状态，预计送达时间等信息。

g. 我的界面

个人信息编辑：

修改头像、昵称、简介等资料。

相互关注列表：

浏览并管理自己关注的人以及粉丝名单。

我的创作汇总：

集中展示已发表的所有作品，支持编辑或删除操作。

查看访客记录：

了解谁访问了自己的主页。

反馈意见提交：

提供表单让用户表达对产品的看法或遇到的问题。

系统设置：

调整主题颜色、开启/关闭锁屏密码等功能。

3.3.2 外部接口

a. 用户认证与授权接口

OAuth 2.0 授权服务器：

功能描述：提供第三方登录（如微信、QQ、微博）和单点登录（SSO）服务。

API 端点：

/auth/authorize：用于发起授权请求。

/auth/token：用于交换授权码获取访问令牌。

/auth/userinfo：用于获取用户信息。

安全措施：使用 HTTPS 协议，确保传输过程中的数据加密；对客户端应用进行注册管理，分配唯一的 Client ID 和 Secret。

b. 支付网关接口

支付服务提供商集成：

功能描述：集成主流支付渠道（如支付宝、微信支付），实现在线支付功能。

API 端点：

/payment/createOrder：创建订单，返回支付链接或二维码。

/payment/queryOrder：查询订单状态。

/payment/refund：处理退款请求。

安全措施：采用 SSL/TLS 加密通信，确保敏感信息的安全传输；对接口调用进行签名验证，防止篡改。

c. 推送通知服务接口

消息推送服务:

功能描述: 通过推送服务向用户发送即时通知, 如新消息提醒、订单更新等。

API 端点:

/push/sendNotification: 发送推送消息给指定设备或用户组。

技术选择: 考虑使用 Firebase Cloud Messaging (FCM) 或者 Apple Push Notification service (APNs), 根据目标平台选择合适的推送服务。

安全措施: 确保推送内容符合相关法律法规, 避免滥用推送服务; 限制推送频率, 提高用户体验。

d. OCR 识别接口

OCR 服务:

功能描述: 允许用户上传图片, 自动识别并提取其中的文字信息, 主要用于创作拍照识别。

API 端点:

/ocr/recognizeText: 接收图片文件, 返回识别出的文字内容。

技术选择: 可以选用百度 AI 平台提供的 OCR API 或其他成熟的第三方 OCR 服务。

安全措施: 确保上传的图片不包含敏感信息, 或者在传输过程中进行适当的脱敏处理。

e. 数据同步与备份接口

云存储服务:

功能描述: 定期将本地数据库的数据备份到云端, 保证数据的安全性和可恢复性。

API 端点:

/backup/uploadData: 上传增量数据至云存储。

/backup/downloadData: 从云存储下载备份数据。

技术选择: 考虑使用 AWS S3、阿里云 OSS 等可靠的云存储解决方案。

安全措施: 所有上传下载操作均需经过身份验证和权限控制; 对存储的数据进行加密保护。

f. 第三方商城组件接口

商品管理系统集成:

功能描述: 集成了其他软件开发商提供的组件, 用于管理商城的商品信息、库存管理等功能。

API 端点:

/product/getProductList: 获取商品列表。

/product/getProductDetail: 获取商品详情。

/inventory/updateStock: 更新库存数量。

安全措施: 确保接口调用的安全性, 例如通过 API 密钥、OAuth 等方式进行认证; 限制接口调用的频率, 防止恶意攻击。

3.3.3 内部接口

a. 用户认证与授权接口

功能描述: 处理用户注册、登录、密码找回等操作, 并提供统一的身份验证和权限管理。

API 端点:

/auth/checkToken: 验证访问令牌的有效性。

/composition/search: 搜索符合特定条件的创作。

/social/notification: 获取用户的最新通知列表。

/store/payment/process: 处理支付请求。

/profile/feedback: 提交产品使用中的问题或建议。

`/data/sync/start:` 启动数据同步过程。

`/data/backup/restore:` 从备份中恢复数据。

功能描述：用于审核用户上传的内容，确保其合法合规。

/content/review/submit: 提交待审内容。

/content/review/approve: 批准通过审核的内容。

/content/review/reject: 拒绝不符合标准的内容。

3.4.1 运行模块组合

用户管理子模块：负责用户的注册、登录、密码重置和个人资料维护。

权限控制子模块：管理用户角色和权限分配，确保只有授权用户可以访问资源。

创作创作子模块：提供在线编辑器，支持富文本格式，实时保存草稿，并相似素材。

创作评估子模块：集成自然语言处理技术，自动评分并给出改进建议。

创作推荐子模块：基于用户兴趣和行为数据，智能推荐相关创作。

创作存储子模块：管理和维护创作数据库，支持文章的创建、编辑、发布和删除操作。

动态发布子模块：允许用户分享文字、图片或短视频动态，并与其他用户的动态互动（点赞、评论）。

交友匹配子模块：通过测文值和随机匹配等功能帮助用户找到志同道合的朋友。

消息中心子模块：汇总来自私信、评论、点赞等多种类型的通知，保持用户活跃度。

社区管理子模块：构建用户关系图谱，管理好友列表，支持私信聊天和个人主页展示。

商品管理子模块：维护商品信息，支持分类检索和详情页展示。

购物车管理子模块：实现商品添加、数量调整、结算等功能，确保购物流程便捷。

订单处理子模块：接收并处理用户的购买请求，生成订单记录，跟踪配送进度。

支付网关子模块：集成多种支付方式，如支付宝、微信支付等，确保交易安全可靠。

个人信息编辑子模块：允许用户修改头像、昵称、简介等资料。

我的创作子模块：集中展示已发表的所有作品，支持编辑或删除操作。

查看访客子模块：了解谁访问了自己的主页。

反馈意见提交子模块：提供表单让用户表达对产品的看法或遇到的问题。

系统设置子模块：调整主题颜色、开启/关闭锁屏密码等功能。

f. 数据同步与备份模块

增量数据上传子模块：定期将本地数据库的数据备份到云端。

数据恢复子模块：从云存储下载备份数据，用于灾难恢复。

日志管理子模块：记录所有重要的操作日志，便于故障排查和审计跟踪。

g. 内容审核模块

内容提交子模块：接收用户上传的内容进行初步处理。

人工审核子模块：安排专人或借助 AI 工具进行最终审核，确保内容合法合规。

违规处理子模块：对于不符合标准的内容采取相应措施，如删除、警告等。

h. 安全与监控模块

网络安全子模块：实施防火墙、入侵检测等防护措施，防范网络攻击。

数据加密子模块：对敏感信息进行加密存储和传输，保护用户隐私。

性能监控子模块：实时监控服务器性能、网络流量等关键指标，及时发现并解决问题。

故障报警子模块：一旦检测到异常情况立即通知相关人员采取行动。

3.4.2 运行控制

a. 监控与报警

服务器性能监控：

CPU、内存、磁盘 IO、网络流量等关键指标：使用 Prometheus、Grafana 等工具进行实时监控，设置合理的阈值，一旦超过即触发警报。

应用程序健康检查：通过心跳检测、API 响应时间等方式监控应用的健康状态。

安全事件监控：

入侵检测系统（IDS）/入侵防御系统（IPS）：部署防火墙规则，防止未经授权访问；利用 WAF 保护 Web 应用免受常见攻击。

异常行为分析：通过机器学习算法识别异常登录尝试、暴力破解等恶意活动，并及时响应。

用户行为监控：

操作日志记录：详细记录用户的每一步操作，包括登录、发布内容、购买商品等，以便审计和问题追踪。

流量统计分析：分析用户访问模式，优化推荐算法和服务配置。

报警机制：

即时通知：当监控到任何异常情况时，立即通过短信、邮件或即时通讯工具发送告警信息给运维团队。

分级响应：根据问题严重程度设定不同的响应级别，如轻微警告、紧急处理等。

b. 日志管理

集中化日志收集：

ELK Stack（Elasticsearch, Logstash, Kibana）：用于收集、存储和可视化所有服务的日志数据，方便查询和分析。

日志分类：将日志分为不同类别（如应用日志、安全日志、访问日志），便于管理和检索。

日志保留策略：

定期清理：根据业务需求制定合理的日志保留周期，过期日志自动归档或删除。

归档备份：重要日志应定期备份至离线存储介质，以备不时之需。

c. 故障处理与恢复

自动重启机制：

进程守护：使用 Supervisor 等工具监控关键进程，一旦发现崩溃立即重启。

容器编排：采用 Kubernetes 等容器管理平台，实现服务的自动化部署、扩展和自我修复。

容错设计：

冗余架构：为关键组件提供多实例部署，确保即使单点故障也不会影响整体服务。

负载均衡：通过 Nginx 或其他 LB 解决方案分发请求，提高系统的并发处理能力和稳定性。

灾难恢复计划：

备份策略：每日全量备份数据库，增量备份频繁更新的数据表；同时备份配置文件和其他静态资源。

恢复演练：定期测试备份的有效性和恢复流程，确保在真实灾难发生时能够快速恢复正常运行。

d. 性能优化

缓存机制：

分布式缓存：引入 Redis 作为缓存层，减少数据库查询压力，提升读取效率。

页面静态化：对于不常变动的内容生成静态 HTML 页面，直接从 CDN 分发，减轻服务器负担。

代码优化：

性能瓶颈排查：利用 APM（Application Performance Management）工具定位慢查询、长耗时接口等问题。

算法改进：不断优化核心算法，降低计算复杂度，加快处理速度。

资源调度：

弹性伸缩：根据实际负载动态调整云服务器数量，避免资源浪费或不足。

任务队列：非即时任务（如邮件发送、报表生成）放入消息队列异步执行，改善用户体验。

e. 安全控制

身份验证与权限管理：

OAuth 2.0 协议：确保第三方登录的安全性，保护用户隐私。

细粒度权限控制：基于角色的访问控制（RBAC），精确划分每个用户的角色和权限。

数据加密：

传输层加密：所有对外接口均采用 SSL/TLS 加密通信，防止中间人攻击。

存储层加密：对敏感数据进行加密存储，即使数据库泄露也能保护用户信息安全。

安全审计：

定期审查：组织内部或聘请外部专家定期审查系统的安全性，查找潜在漏洞并及时修补。

合规性检查：确保系统符合相关法律法规要求，如 GDPR、CCPA 等。

3.4.3 运行时间

a. 日常运行时间

24/7 连续服务：

系统需保证每天 24 小时、每周 7 天的连续可用性，包括公共假期和周末，以满足用户的随时访问需求。

关键业务时段：

高峰时段：通常为工作日的晚上（如 18:00-23:00）以及周末全天，应特别关注性能监控和资源调度，确保响应速度不受影响。

低谷时段：非高峰期间可适当减少服务器资源分配，优化成本。

b. 定期维护窗口

周度维护：

每周选择一个用户活动较低的时间段（例如周一凌晨 00:00-02:00）进行例行维护，如更新补丁、清理缓存等。维护前应提前通知用户，并尽量缩短停机时间。

月度维护：

每月第一个周五的凌晨进行更大规模的维护工作，包括数据库优化、安全补丁安装等。此时间段内可能需要短暂中断部分服务，但应控制在最短时间内完成。

c. 数据备份与恢复

每日全量备份：

每天凌晨 02:00 启动一次完整的数据库备份任务，确保所有数据都能得到及时保存。

增量备份：

在白天每隔两小时进行一次增量备份，快速捕捉最新的数据变化，以便发生故障时可以迅速恢复到最近的状态。

定期恢复测试：

每月最后一个周六进行一次备份恢复演练，验证备份的有效性和恢复流程的可行性。

d. 性能监控与调整

实时监控：

使用 Prometheus、Grafana 等工具对系统性能进行 7x24 小时不间断监控，设置合理的阈值并配置自动报警机制。

高峰期性能优化：

在预估的高并发访问时间内，提前扩容服务器资源，启用更多的缓存节点，确保系统能够应对大量请求而不出现延迟或错误。

e. 内容审核

实时审核：

用户上传的内容应尽可能实现实时审核，利用 AI 辅助工具初步筛选后由人工复核，确保内容合法合规，不影响正常发布流程。

夜间批量处理：

对于未及时审核的内容，可以在夜间低峰时段集中处理，减轻白天的审核压力。

f. 推送通知

即时推送：

新消息提醒、订单状态更新等重要通知应在事件发生后的数秒内推送给用户，确保信息传递的及时性。

定时推送：
营销活动、促销信息等内容可以选择在特定时间点（如上午 9:00 或下午 5:00）统一发送，避免打扰用户休息。

g. 安全扫描与漏洞修复
每日安全扫描：
利用自动化工具每日对系统进行全面的安全扫描，检测潜在的安全隐患，如 SQL 注入、XSS 攻击等。

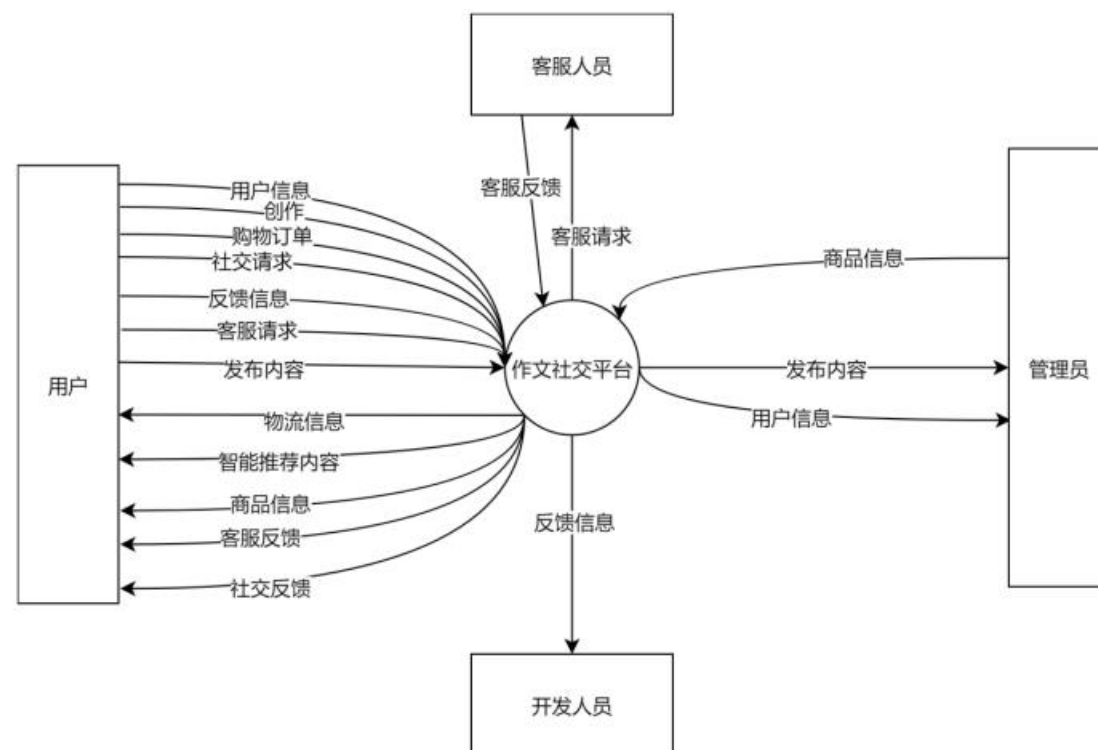
紧急修复：
发现严重漏洞时，立即启动应急响应机制，在最短的时间内完成修复并部署更新，确保系统的安全性。

3.5 系统数据结构设计

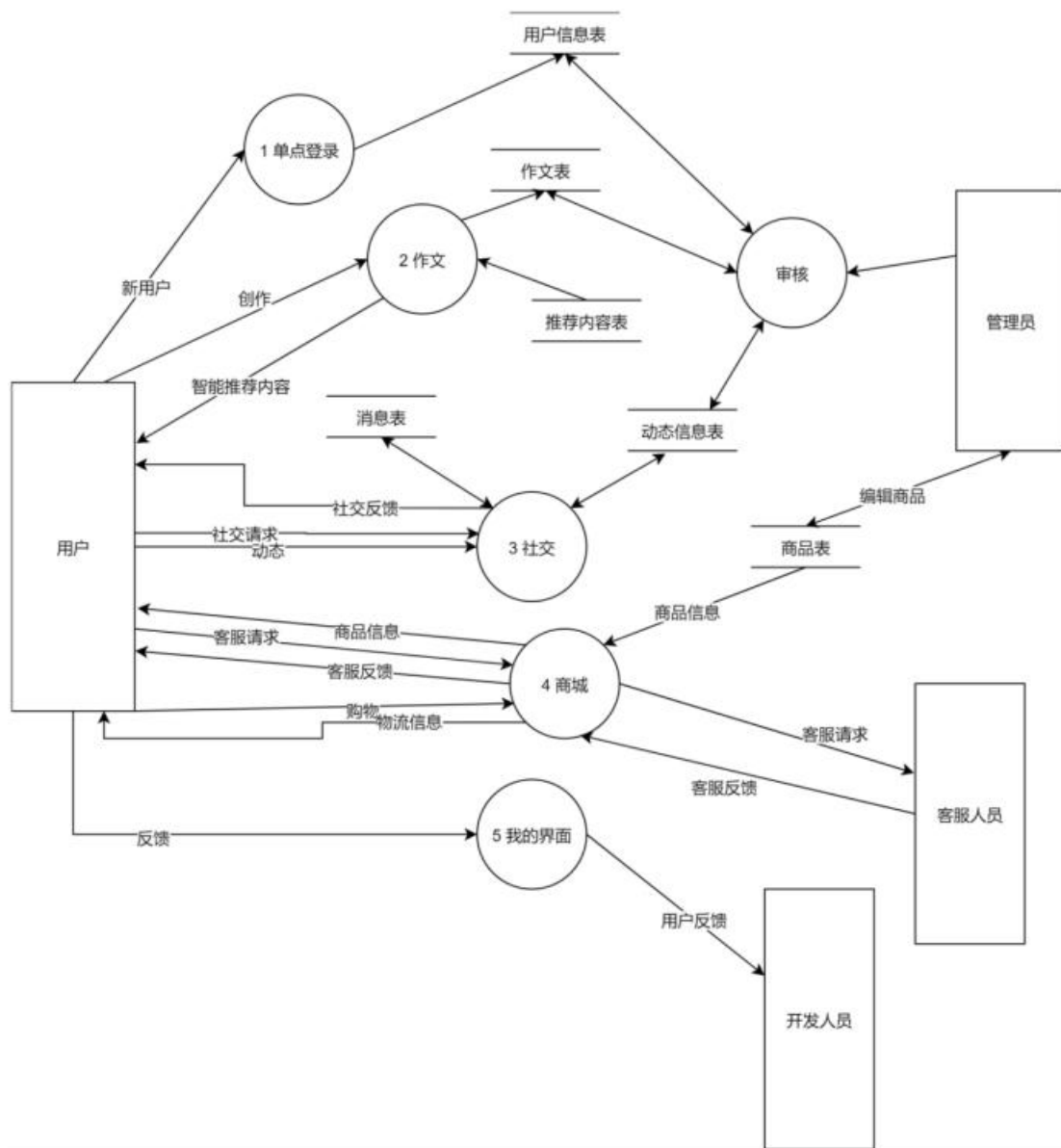
3.5.1 逻辑结构设计要点

逻辑结构设计是指从业务需求出发，定义数据对象之间的关系和结构，确保数据的完整性和一致性。

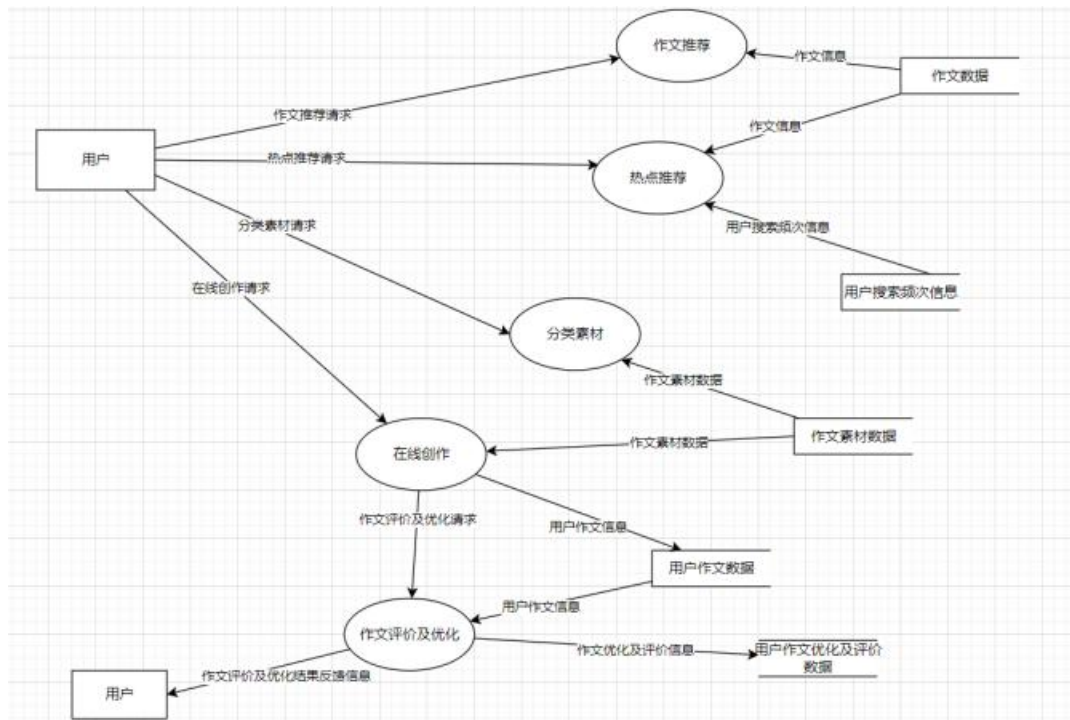
数据流图：
0-层数据流图：描述了系统的整体数据流动，包括用户、服务器、数据库等主要模块之间的数据交互。



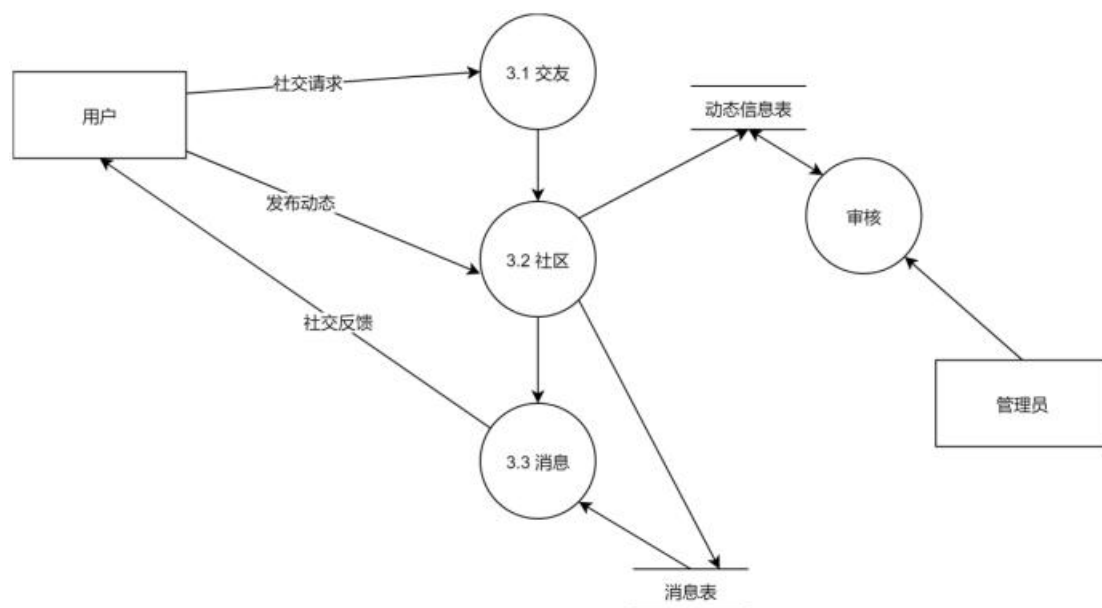
1-层数据流图：进一步细化了各个功能模块的数据流动，例如创作功能、社交功能、购物功能等。



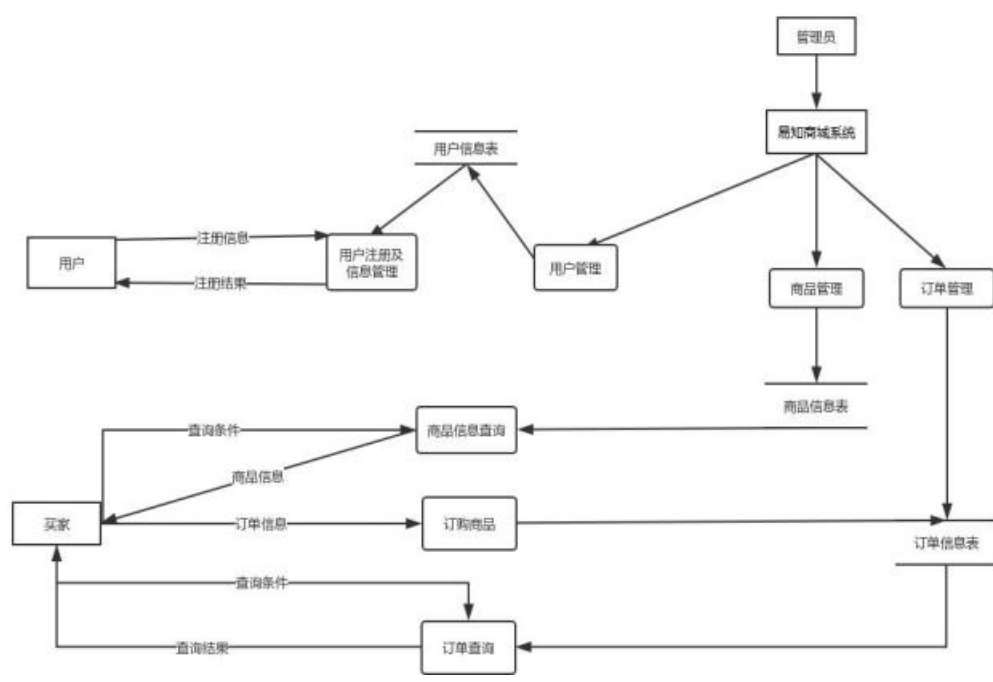
2-层数据流图：详细描述了具体功能模块的数据流动，例如创作功能中的用户提交创作、系统推荐创作等。



创作功能数据流图



社交功能数据流图



购物功能数据流图

数据字典：

数据条目：定义了系统中主要的对象及其组成。例如：

用户基本信息：包括用户 ID 和手机号。

用户详细信息：包括用户 ID、昵称、头像、性别、年龄、学历、居住城市等。

用户动态：包括动态 ID、点赞数量、评论、内容、创建日期等。

创作：包括创作 ID、内容、创建日期等。

商品信息：包括商品 ID、名称、类别、单价等。

订单信息：包括订单 ID、商品 ID、数量、用户 ID 等。

| ID | 数据流名 | 标识符 | 组成 |
|-------|--------|--------|--|
| DDE-1 | 用户基本信息 | YHJBXX | 用户基本信息=id+手机号 |
| DDE-2 | 用户详细信息 | YHXXXX | 用户详细信息=id+用户 id+昵称+用户头像+用户标签+性别+年龄+学历+居住城市+生日+封面图片 |
| DDE-3 | 用户动态 | YHDT | 用户动态=id+点赞数量+喜欢+评论+内容+创建日期 |
| DDE-4 | 短视频 | DSP | 短视频=id+类别+内容+创建日期 |
| DDE-5 | 创作 | ZW | 创作=id+内容+创建日期 |
| DDE-6 | 商品信息 | SP | 商品信息=id+日期+名称+类别+单价 |
| DDE-7 | 订单信息 | DDXX | 订单信息=id+订单 id+商品 id+数 |

| | | | |
|--|--|--|---------|
| | | | 量+用户 id |
|--|--|--|---------|

数据项：定义了每个数据对象的具体字段及其属性。例如：

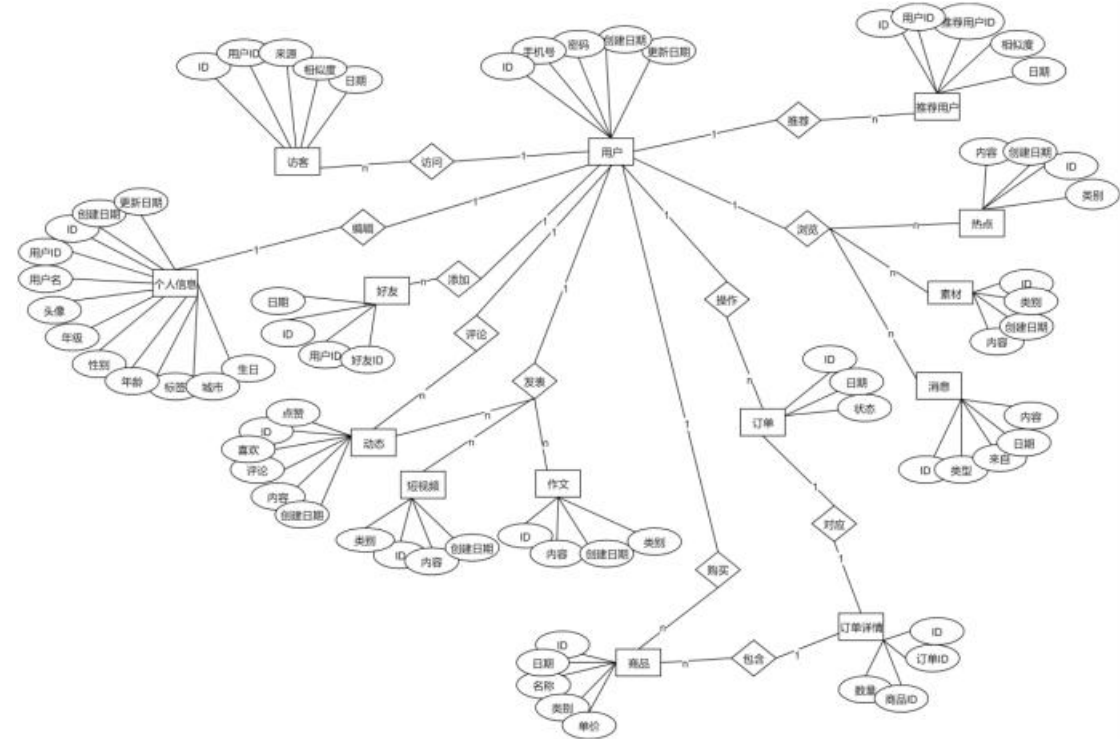
tb_user 表：包括用户 ID、手机号、密码、创建时间、更新时间等字段。

tb_user_info 表：包括用户详细信息字段，如昵称、头像、性别、年龄、学历、居住城市等。

| ID | 业务数据对象 | 字段名 | 数据类型 | 允许为空 | PK | 字段说明 |
|--------|--------------|-----------|--------------|------|----------------|-----------------|
| DDI-1 | tb_user | id | bigint(20) | NO | auto_increment | |
| DDI-2 | tb_user | mobile | varchar(11) | YES | | 手机号 |
| DDI-3 | tb_user | password | varchar(32) | YES | | 密码，需要加密 |
| DDI-4 | tb_user | created | datetime | YES | | |
| DDI-5 | tb_user | updated | datetime | YES | | |
| DDI-6 | tb_user_info | id | bigint(20) | NO | auto_increment | |
| DDI-7 | tb_user_info | user_id | bigint(20) | NO | | 用户 id |
| DDI-8 | tb_user_info | user_name | varchar(50) | YES | | 昵称 |
| DDI-9 | tb_user_info | logo | varchar(100) | YES | | 用户头像 |
| DDI-10 | tb_user_info | tags | varchar(50) | YES | | 用户标签：多个用逗号分隔 |
| DDI-11 | tb_user_info | sex | int(1) | YES | | 性别，1-男，2-女，3-未知 |
| DDI- | tb_user_info | age | int(11) | YES | | 用户 |

| | | | | | | |
|--------|--------------|-----------|--------------|-----|--|------|
| 12 | | | | | | 年龄 |
| DDI-13 | tb_user_info | edu | varchar(20) | YES | | 学历 |
| DDI-14 | tb_user_info | city | varchar(20) | YES | | 居住城市 |
| DDI-15 | tb_user_info | birthday | varchar(20) | YES | | 生日 |
| DDI-16 | tb_user_info | cover_pic | varchar(100) | YES | | 封面图片 |
| DDI-17 | tb_user_info | created | datetime | YES | | |
| DDI-18 | tb_user_info | updated | datetime | YES | | |
| ... | | | | | | |

实体关系图（ER 图）
描述了系统中各个实体之间的关系。例如：
用户与创作：用户可以发布多篇创作。
用户与动态：用户可以发布多条动态。
用户与订单：用户可以创建多个订单。
商品与订单：订单中可以包含多个商品。



3.5.2 物理结构设计要点

物理结构设计是指将逻辑结构映射到具体的存储介质（如数据库）中，确

保数据的高效存储和访问。

数据库设计

数据库类型:

系统使用了多种数据库类型, 包括关系型数据库 (MySQL)、NoSQL 数据库 (MongoDB、Redis、Neo4j)。

MySQL: 用于存储结构化数据, 如用户信息、创作信息、订单信息等。

MongoDB: 用于存储非结构化数据, 如用户动态、短视频等。

Redis: 用于缓存高频访问的数据, 如用户登录状态、验证码等。

Neo4j: 用于存储用户与用户之间的关系, 支持社交功能中的用户推荐和好友关系管理。

表结构设计:

每个表都定义了主键 (PK) 和外键 (FK), 确保数据的完整性和一致性。

例如, `tb_user` 表的主键是用户 ID, `tb_user_info` 表通过外键关联到 `tb_user` 表。

索引设计:

为常用的查询字段 (如用户 ID、商品 ID、订单 ID 等) 创建索引, 提高查询效率。

分区设计:

对于大数据量的表 (如创作表、动态表), 可以考虑按时间或用户 ID 进行分区存储, 提高查询性能。

数据存储优化

数据压缩:

对于大文本数据 (如创作内容), 可以考虑使用压缩算法减少存储空间。

数据缓存:

使用 Redis 缓存高频访问的数据, 减少数据库的访问压力。

数据备份:

定期备份关键数据, 确保数据的安全性和可恢复性。

3.5.3 数据结构与程序的关系

数据结构与程序的关系是指数据结构如何支持程序的功能实现, 确保程序能够高效地访问和操作数据。

数据访问接口

数据库接口: 程序通过数据库接口 (如 JDBC、ORM 框架) 访问数据库, 执行增删改查操作。

缓存接口: 程序通过 Redis 接口访问缓存数据, 提高数据访问速度。

文件存储接口: 对于大文件 (如短视频、图片), 程序通过文件存储接口 (如云存储服务) 进行存储和访问。

数据处理逻辑

智能推荐算法: 系统使用知识图谱和智能推荐算法, 根据用户的行为和兴趣推荐创作、用户、商品等。

数据分析: 系统通过分析用户行为数据 (如浏览记录、点赞记录、购买记录), 生成用户画像, 支持个性化推荐和广告投放。

数据同步: 系统需要同步多个数据库之间的数据, 例如用户信息在 MySQL 和 Redis 之间的同步。

数据一致性

事务管理：对于涉及多个数据操作的功能（如订单创建、支付），程序需要使用事务管理，确保数据的一致性。

分布式锁：对于高并发场景（如秒杀活动），程序需要使用分布式锁，避免数据冲突。

数据安全

数据加密：对于敏感数据（如用户密码、支付信息），程序需要进行加密存储和传输。

权限控制：程序需要根据用户的权限控制数据的访问范围，确保数据的安全性。

3.6 系统出错处理设计

3.6.1 出错信息

错误码设计：

系统应设计统一的错误码，用于标识不同类型的错误。错误码应能够快速定位错误的来源，并提供给开发人员和用户明确的错误信息。

例如，系统应为常见的错误类型（如网络错误、数据库连接失败、权限不足等）定义标准的错误码。

异常日志输出：

系统应记录详细的异常日志，包括错误发生的时间、地点、错误类型、调用链路等信息。

日志级别应根据错误严重程度进行分类，例如：

Error 级别：记录系统逻辑出错、异常或其他重要的错误信息。

Warn 级别：记录潜在的错误或警告信息。

Info 级别：记录系统正常运行的关键信息。

用户友好提示：

系统应向用户提供友好的错误提示信息，避免暴露技术细节。

例如，当用户操作失败时，系统应提示“操作失败，请稍后重试”或“网络异常，请检查网络连接”。

3.6.2 补救措施

失败设计思维：

系统设计应考虑异常情况，提前准备好从运行失败的场景中恢复。例如：

入参判空：在接口设计中，对输入参数进行判空和有效性校验，避免因非法输入导致系统崩溃。

业务边界处理：在设计主干功能时，充分考虑异常流程和业务边界。例如，用户在支付过程中，如果银行扣款成功但系统入账失败，系统应支持重试或人工干预。

断网处理：

系统应支持离线模式，例如用户在线创作时，如果网络中断，系统应允许用户选择离线保存，待网络恢复后再同步数据。

人脸识别失败处理：

如果用户在登录时人脸识别失败超过 3 次，系统应自动切换到验证码登录，避免用户无法登录。

游客访问提示：

如果用户以游客身份访问“我的”页面，系统应提示用户登录，避免游客访问受限功能。

3.6.3 系统维护设计

监控与告警：

系统应对关键接口和服务器进行实时监控，当发生故障时，及时发送短信或邮件提醒给运维人员。

例如，系统应监控数据库连接状态、接口调用成功率、服务器负载等指标，并在异常时触发告警。

日志分析与故障定位：

系统应提供日志分析工具，帮助运维人员快速定位故障原因。

例如，通过分析异常日志，运维人员可以确定是数据库连接问题、网络问题还是代码逻辑问题。

自动化运维：

系统应支持自动化运维工具，例如自动化部署、自动化扩容等，以提高系统的可维护性。

备份与恢复：

系统应定期备份关键数据（如用户信息、创作数据、订单数据等），并在发生数据丢失时能够快速恢复。

版本管理与回滚：

系统应支持版本管理，确保每次发布都有明确的版本号。如果新版本出现问题，系统应支持快速回滚到上一个稳定版本。

第 4 章 系统评估

4.1 功能性属性验证

4.1.1 功能性属性验证的依据

功能需求表：文档中详细列出了各个功能模块的需求（如单点登录、创作相关功能、社交功能、商城功能、我的界面功能等），这些需求是功能性属性验证的基础。

用例描述：每个功能模块都有对应的用例描述，详细说明了用户操作的步骤、预期结果以及验收标准，这些用例可以用于功能性测试和验证。

4.1.2 功能性属性验证的主要内容

a.单点登录功能：

验证用户通过手机验证码登录的功能是否正常。
验证人脸识别功能是否能够有效识别用户真实性。
验证用户在登录后是否能够顺利补充个人资料。

b.创作相关功能：

验证热点素材推荐、分类素材推荐、创作推荐等功能是否能够正常工作。
验证在线写作功能是否支持动态修改段落、推荐素材等功能。

c.社交功能：

验证用户通过测试题了解兴趣范围的功能是否有效。
验证用户是否能够通过系统推荐或搜索功能找到志同道合的笔友。
验证动态发布、点赞、评论、收藏等功能是否正常。
验证短视频功能是否支持用户上传、浏览、点赞等操作。

d.商城功能：

验证商品分类、搜索、购物车、订单创建、支付等功能是否正常。
验证用户是否能够正常管理收货地址、查看商品评价等。

e.我的界面功能：

验证用户是否能够编辑个人信息、查看相互关注、查看我的创作等功能。
验证访客功能、反馈功能、系统设置功能是否正常。

4.1.3 功能性属性验证的优先级

文档中对每个功能需求都标注了优先级（①、②、③、④），功能性属性验证应按照优先级进行：

优先级①：单点登录、创作推荐、在线写作、社交功能中的用户推荐、商城中的订单创建和支付等功能。

优先级②：热点素材推荐、分类素材推荐、社交功能中的动态发布、消息功能等。

优先级③：商城中的商品分类、搜索、收货地址管理等功能。

优先级④：我的界面中的访客功能、反馈功能等。

4.1.4 功能性属性验证的验收标准

每个功能需求都有明确的验收标准，例如：

单点登录功能：用户能够通过验证码登录，人脸识别系统能够识别用户真实性。

创作相关功能：用户能够通过热点素材推荐浏览热点素材，能够在线写作并获取推荐素材。

社交功能：用户能够通过测试题了解兴趣范围，能够发布动态并进行点赞、评论等操作。

商城功能：用户能够成功检索商品、创建订单并完成支付。

我的界面功能：用户能够编辑个人信息、查看我的创作等。

4.1.5 功能性属性验证的测试方法

功能测试：根据用例描述，模拟用户操作，验证系统是否能够按照预期响应。

边界测试：验证系统在极端情况下的表现，例如非法输入、网络异常等。

性能测试：验证系统在用户操作时的响应时间是否符合性能需求（如 1 秒内返回结果）。

4.1.6 功能性属性验证的工具和方法

墨刀原型工具：文档中提到使用墨刀建立高保真原型，可以用于功能性验证的前期测试。

自动化测试工具：可以考虑使用自动化测试工具（如 Selenium、Appium 等）对功能进行自动化验证。

手动测试：对于一些复杂的交互功能，手动测试仍然是必要的。

功能性属性验证的主要依据是文档中的功能需求表和用例描述，验证内容涵盖了单点登录、创作相关功能、社交功能、商城功能和我的界面功能等。验证应按照优先级进行，优先级①的功能应优先验证。验收标准和测试方法应结合功能需求和用例描述进行设计，确保系统功能符合用户需求。

4.2 非功能性属性验证

4.2.1 性能需求验证

时间特性：验证用户操作时，系统是否能够在 1 秒内返回准确结果，最多不超过 4 秒。

系统有效性：验证系统是否能够每天连续运行 24 小时，每周运行七天，包括公共假期和周末。

容错性：验证系统在用户正常操作时是否不会出错或崩溃。

可扩充性：验证系统是否能够支持未来扩展数据的需求。

内容完整性：验证系统是否包含当前需求所需的全部数据，以及可能用到的数据。

数据准确性：验证系统中展示的数据是否准确无误。

4.2.2 安全性需求验证

应对网络攻击：验证系统是否能够防护常见的网络攻击类型，如防爬、防暴力破解、防注入、防跨站脚本攻击等。

敏感信息传输：验证敏感信息的传输和保存是否加密。

敏感信息接口调用：验证敏感信息接口的调用是否需要安全认证。

权限控制：验证系统是否进行了访问控制和权限控制。

故障提醒：验证系统是否能够监控接口和服务器，并在发生故障时及时发送短信提醒。

4.2.3 软件质量属性验证

易用性要求：

状态可见：验证用户是否能够时刻清楚自己在进行什么操作，系统是否有反馈和进度显示。

环境贴切：验证系统是否模仿真实世界的产品或使用方式，降低用户学习成本。

一致性：验证系统是否遵循统一的设计规范和逻辑，保持结构、色彩、操作、反馈和文字的一致性。

个性化帮助：验证系统是否提供简化选项、向导或演示，是否允许用户跳过不必要的操作。

可用性要求：

可学习性：验证用户初次接触系统时是否能够轻松了解并使用该产品设计。

效率：验证用户是否能够在简单的几步操作后快速完成需求。

可记忆性：验证用户在一段时间未使用系统后是否能够快速再次熟练使用。

出错：验证系统在用户触发未知错误后是否能够友好提示用户信息，并协助恢复正常使用状态。

满意度：验证 90%的用户是否对系统达到满意程度。

鲁棒性要求：

设计规约：验证系统是否具备失败设计思维，能够处理异常流程和业务边界。

图示表达设计：验证系统是否使用图示表达设计，清晰展示业务关联关系和异常分支。

异常错误处理：验证系统是否具备异常错误处理机制，能够快速定位错误并输出错误码和日志。

4.2.4 业务规则验证

用户登录验证码有效期：验证用户登录的验证码是否在 5 分钟内有效。

人脸识别失败处理：验证人脸识别 3 次失败后，系统是否要求用户使用验证码登录。

在线创作断网处理：验证在线创作创作时，系统是否支持离线保存。

游客访问提示：验证游客身份访问“我的”页面时，系统是否提示用户登录。

搜附近功能 GPS 检测：验证搜附近功能启用时，系统是否检测 GPS 是否

开启，未开启时是否提示用户开启 GPS 定位。

4.2.5 用户文档验证

用户手册：验证是否提供了符合 GB8567-88 标准的纸质用户手册。

帮助文档：验证是否提供了在线帮助文档，用户是否能够通过帮助文档解决问题。

在线教程：验证是否提供了在线教程，用户是否能够通过教程快速上手系统。

总结：非功能性属性验证主要包括性能需求、安全性需求、软件质量属性、业务规则和用户文档的验证。验证内容涵盖了系统的响应时间、连续运行能力、容错性、安全性、易用性、可用性、鲁棒性等方面。验证时应结合文档中的具体需求和验收标准，确保系统在非功能性方面满足用户和业务的要求。

4.3 其他需求验证

4.3.1 用户原创作品的版权保护

需求描述：用户在该平台上上传的所有原创作品应当受《中华人民共和国著作权法》相关规定保护，版权归原创作者所有。软件使用协议或条款中不得非法侵占用户权益。

验证内容：

验证软件使用协议或条款中是否明确注明了用户原创作品的版权归属，确保符合《中华人民共和国著作权法》的相关规定。

验证系统是否采取了技术手段（如加密、水印等）保护用户上传的原创作品，防止未经授权的复制或传播。

验证系统是否提供了用户投诉和版权申诉的渠道，确保用户能够维护自己的版权权益。

4.3.2 敏捷开发与 DevOps 思想的结合

需求描述：软件开发过程中应在敏捷开发的基础上增加 DevOps 的开发思想，通过自动化“软件交付”和“架构变更”的流程，使得构建、测试、发布软件更加快捷、频繁和可靠。

验证内容：

验证开发团队是否采用了敏捷开发方法，是否定期进行迭代开发和交付。

验证是否引入了 DevOps 工具链（如 Jenkins、GitLab CI/CD 等），用于自动化构建、测试和发布流程。

验证是否实现了持续集成（CI）和持续交付（CD），确保软件的发布过程高效且可靠。

验证开发团队是否与运维团队紧密合作，确保软件的部署和维护过程顺畅。

4.3.3 用户协议与法律合规性

需求描述：软件使用协议或条款中应明确用户权益，不得包含非法侵占用户权益的条款。

验证内容：

验证用户协议或条款是否符合相关法律法规（如《中华人民共和国合同法》《中华人民共和国著作权法》等）。

验证用户协议中是否明确说明了用户在使用平台时的权利和义务。

验证用户协议中是否避免了不合理的条款（如强制用户放弃版权、不合理的数据收集等）。

4.3.4 用户反馈与需求收集机制

需求描述：系统应提供用户反馈功能，方便开发团队及时发现缺陷和不足，同时收集用户的新需求。

验证内容：

验证系统是否提供了用户反馈入口（如反馈按钮、在线客服等）。

验证用户反馈信息是否能够及时传递给开发团队，并得到有效处理。

验证系统是否定期分析用户反馈，并根据反馈内容进行功能优化或需求调整。

4.3.5 系统日志与监控

需求描述：系统应具备日志记录和监控功能，用于记录用户操作、系统运行状态和异常情况。

验证内容：

验证系统是否能够记录用户的关键操作（如登录、支付、发布内容等）。

验证系统是否具备实时监控功能，能够及时发现并处理异常情况（如服务器宕机、接口调用失败等）。

验证日志记录是否符合安全性和隐私保护的要求，避免敏感信息泄露。

4.3.6 系统兼容性与可扩展性

需求描述：系统应具备良好的兼容性和可扩展性，能够支持未来的功能扩展和数据增长。

验证内容：

验证系统是否能够在不同的运行环境中正常运行（如不同版本的浏览器、操作系统等）。

验证系统是否采用了模块化设计，便于未来功能的扩展和维护。

验证数据库是否支持未来的数据扩展需求（如增加新的表结构或字段）。

总结：“其他需求验证”主要包括以下内容：

用户原创作品的版权保护：验证版权归属和保护措施。

敏捷开发与 DevOps 思想的结合：验证开发流程的自动化和高效性。

用户协议与法律合规性：验证用户协议的合法性和合理性。

用户反馈与需求收集机制：验证用户反馈功能的有效性。

系统日志与监控：验证日志记录和监控功能的实现。

系统兼容性与可扩展性：验证系统的兼容性和扩展能力。

这些需求验证确保了系统在开发、运营和用户权益保护等方面的全面性和合规性。

4.4 软件原型设计

在需求分析阶段通过墨刀工具建立了抛弃型原型并绘制了项目草图。

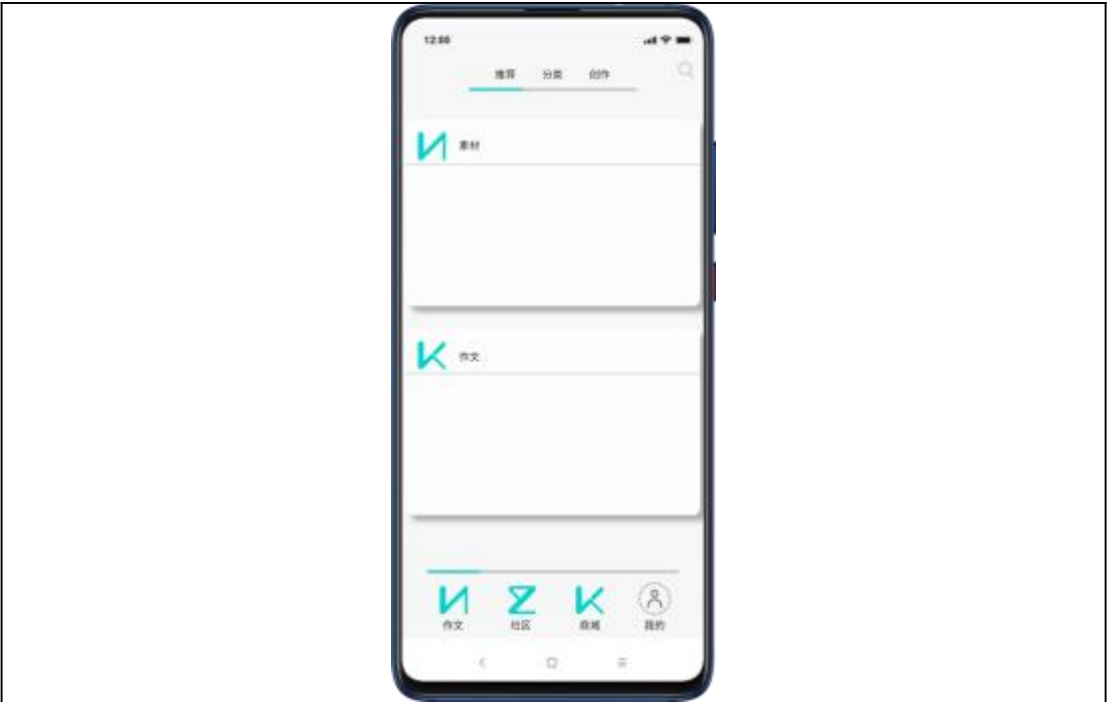
| | |
|--|---|
| <div><div>●</div><div>启动界面</div></div> | <div></div> |
| <div><div>●</div><div>介绍界面</div></div> | |



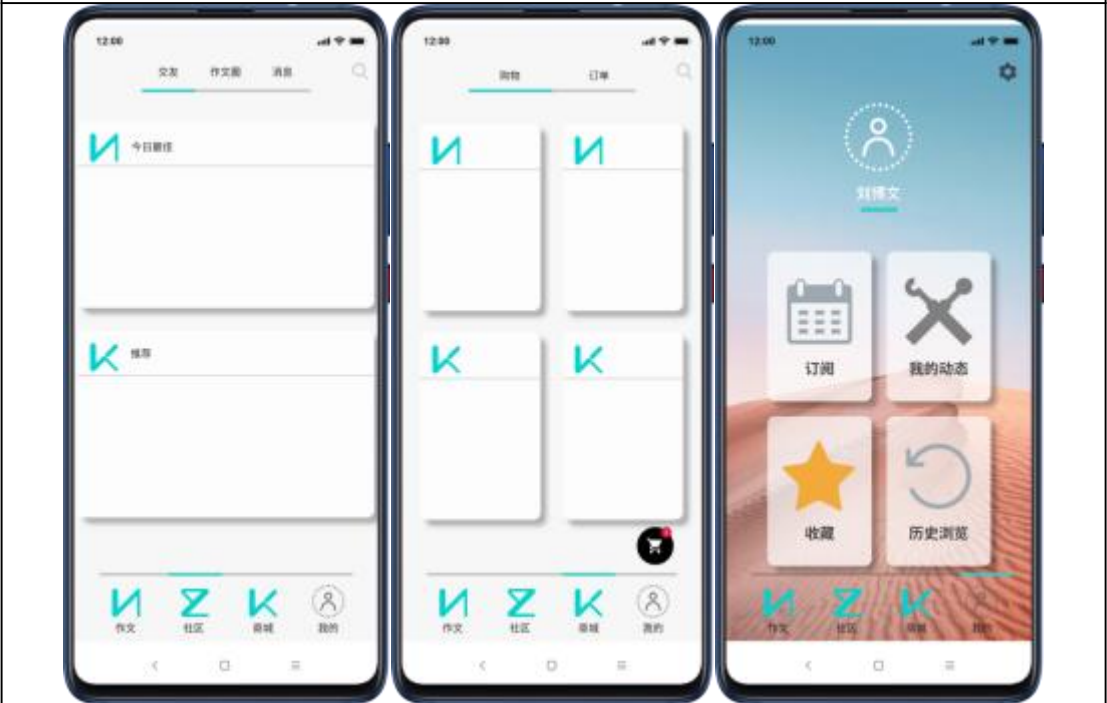
• 登陆界面



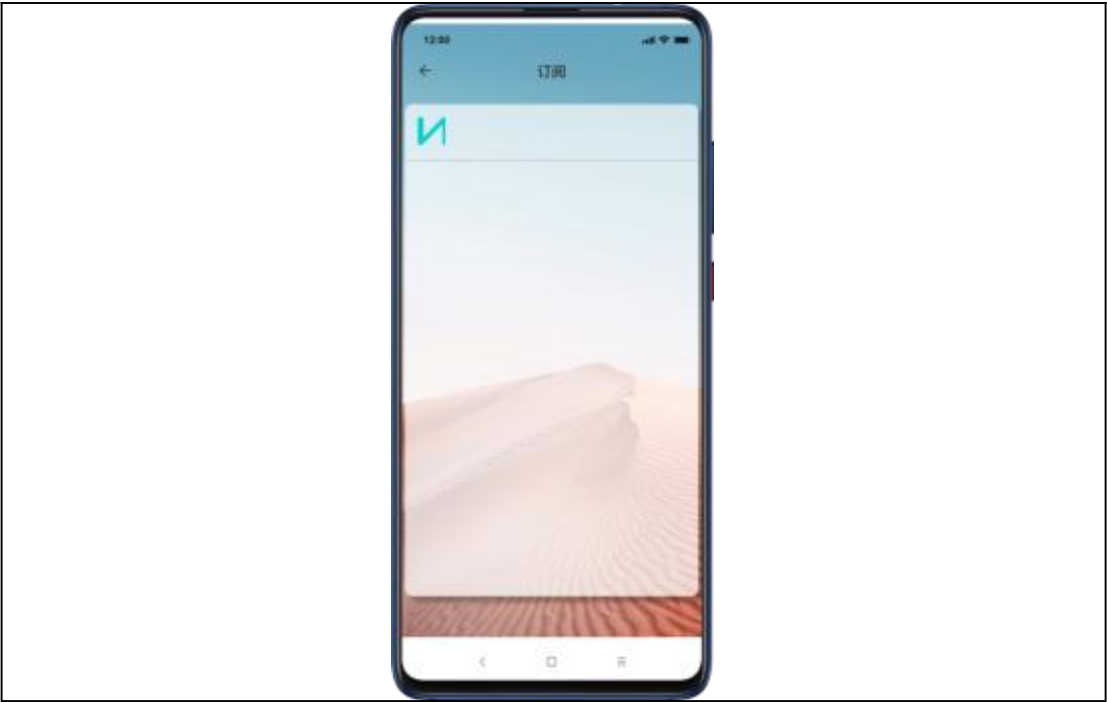
• 主页



• 子页面



• 独立页面



4.5 软件运行界面

在界面实现阶段需参考之前的草图进行界面代码的编写，下面将介绍前台项目组成结构和界面演示。



