

**本科毕业论文**

**基于Web的跨专业互助学习系统**

**学 院:** 数学与信息学院

**专 业:**  计算机科学与技术

**姓 名:** 黄满德

**学 号:** 202129210105

**指导教师:**  姚金涛  **职称** 副教授

**提交日期：** 2025 **年**  4 **月** 14 **日**

华南农业大学本科毕业论文（设计）原创性声明

本人郑重声明：所呈交的毕业论文（设计），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： 日期： 年 月 日

华南农业大学本科毕业论文（设计）使用授权声明

本人完全了解学校有关保留、使用毕业论文（设计）的规定，同意学校保留并向国家有关部门或机构送交毕业论文（设计）的复印件和电子版，允许毕业论文（设计）被查阅和借阅。学校可以将本毕业论文（设计）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毕业论文（设计）。

作者签名： 指导教师签名：

日期： 年 月 日 日期： 年 月 日

摘 要

当今高等教育领域，专业划分愈发精细，这虽有利于在校学生针对特定领域深入钻研，却也致使学生的知识结构局限于本专业范畴内。为突破专业壁垒，助力在校学生构建多元且全面的知识体系，进而优化学习体验，本文着手开展了基于 Web 的跨专业互助学习系统相关研究与设计工作。

在前期准备阶段，通过查阅跨专业学习、互助学习的文献资料，并深入调研分析各类在线学习平台案例后，本文综合考量调研所得结果，明确了系统的开发方向和如下四个工作重点。

在技术层面，前端选用 Vue3 与 Vite 来快速搭建交互界面，后端则采用 Go 语言和 Gin 框架，借助其高效的路由与中间件功能提升系统性能。同时，选用 MySQL 存储结构化数据，利用 WebSocket 和 WebRTC 技术，并结合 Protocol buffer 协议，实现实时通信功能。

在产品设计阶段，本文从实用性的角度出发，确定了账户、文章、互助帮学、用户交流和后台管理多个核心模块，并在后续的开发过程中，注重代码质量与可复用性，严格按照设计方案进行代码编写。

最后，本文设计了全面的测试用例，对各个功能模块进行严格测试，确保系统稳定可靠，达到预期的设计目标。

本系统通过多样化的互动交流和学习方式，能够充分激发学生的学习兴趣，有效提高学生学习的积极性与主动性，能在校园或更大的学习群体中营造出良好的跨专业学习氛围。

关键词：Gin+Vue；跨专业学习平台；聊天系统；WebRTC技术

Web-based interdisciplinary mutual learning system

Huang Mande

（College of Mathematics and Informatics, South China Agricultural University,

Guangzhou 510642, China）

**Abstract：**In today's higher education field, the division of majors is becoming increasingly refined. Although this is conducive to students' in-depth research in specific fields, it limits their knowledge structure to the scope of their major. In order to break through professional barriers, assist students in building a diverse and comprehensive knowledge system, and optimize the learning experience, this article has initiated research and design work on a web-based cross disciplinary mutual aid learning system.

In the early preparation stage, by reviewing literature on interdisciplinary learning and mutual aid learning, and conducting in-depth research and analysis of various online learning platform cases, this article comprehensively considers the research results and clarifies the development direction of the system and the following four work priorities.

On the technical level, Vue3 and Vite are used for the front-end to quickly build interactive interfaces, while Go language and Gin framework are used for the back-end to improve system performance with their efficient routing and middleware functions. At the same time, MySQL is selected to store structured data, utilizing WebSocket and WebRTC technologies, and combined with Protocol buffer protocol to achieve real-time communication functionality.

In the product design phase, this article has identified multiple core modules from a practical perspective, including accounts, articles, mutual assistance and learning, user communication, and backend management. In the subsequent development process, emphasis is placed on code quality and reusability, and code writing is strictly carried out according to the design plan.

Finally, this article designed comprehensive test cases to rigorously test each functional module, ensuring system stability and reliability, and achieving the expected design goals.

This system, through diverse interactive communication and learning methods, can fully stimulate students' interest in learning, effectively improve their enthusiasm and initiative in learning, and create a good interdisciplinary learning atmosphere on campus or in larger learning groups.

**Key words:** Gin + Vue;cross-disciplinary learning platform; Chat system; WebRTC

目 录

[1 绪论 1](#_Toc195905828)

[1.1 课题来源及意义 1](#_Toc195905829)

[1.1.1 课题的来源 1](#_Toc195905830)

[1.1.2 课题研究的意义 1](#_Toc195905831)

[1.2 主要研究的目的及内容 2](#_Toc195905832)

[1.3 研究方法及设计思路 2](#_Toc195905833)

[1.3.1 研究方法 2](#_Toc195905834)

[1.3.2 设计思路 2](#_Toc195905835)

[1.4 本文的主要工作及论文结构组织 3](#_Toc195905836)

[1.4.1 主要工作 3](#_Toc195905837)

[1.4.2 论文结构组织 3](#_Toc195905838)

[2 相关技术简介 5](#_Toc195905839)

[2.1 前端技术简介 5](#_Toc195905840)

[2.2 后端技术简介 6](#_Toc195905841)

[2.3 数据库简介 7](#_Toc195905842)

[2.4 WebSocket简介 8](#_Toc195905843)

[2.4 开发工具及运行环境 8](#_Toc195905844)

[3 系统需求分析 9](#_Toc195905845)

[3.1 系统可行性分析 9](#_Toc195905846)

[3.1.1 技术可行性 9](#_Toc195905847)

[3.1.3 运行可行性 9](#_Toc195905848)

[3.1.4 时间可行性 10](#_Toc195905849)

[3.2 系统总体需求 10](#_Toc195905850)

[3.3 系统功能需求 12](#_Toc195905851)

[3.3.1 账户模块 12](#_Toc195905852)

[3.3.2 文章模块 12](#_Toc195905853)

[3.3.3 互助帮学模块 12](#_Toc195905854)

[3.3.4 用户交流模块 12](#_Toc195905855)

[3.3.5 后台管理 12](#_Toc195905856)

[3.4 本章小结 13](#_Toc195905857)

[4 系统整体设计 14](#_Toc195905858)

[4.1 系统整体设计 14](#_Toc195905859)

[4.1.1 Web系统设计准则 14](#_Toc195905860)

[4.1.2 系统总体结构 14](#_Toc195905861)

[4.2 具体功能模块设计 16](#_Toc195905862)

[4.2.1 用户层设计 16](#_Toc195905863)

[4.2.2 文章管理设计 16](#_Toc195905864)

[4.2.3 互助帮学设计 16](#_Toc195905865)

[4.2.4 用户交流模块设计 16](#_Toc195905866)

[4.2.5 后台管理模块设计 17](#_Toc195905867)

[4.3 数据库设计 17](#_Toc195905868)

[4.3.1 整体数据结构设计 17](#_Toc195905869)

[4.3.2 数据库物理结构设计 17](#_Toc195905870)

[4.4 本章小结 19](#_Toc195905871)

[5 系统详细设计与实现 20](#_Toc195905872)

[5.1 用户注册登录模块实现 20](#_Toc195905873)

[5.2 文章管理模块实现 23](#_Toc195905874)

[5.3 互助功能模块实现 27](#_Toc195905875)

[5.4 用户交流模块实现 28](#_Toc195905876)

[5.5 后台管理模块实现 31](#_Toc195905877)

[5.6 系统表单组件实现 32](#_Toc195905878)

[5.7 本章小结 33](#_Toc195905879)

[6 系统测试 34](#_Toc195905880)

[6.1 测试目的 34](#_Toc195905881)

[6.2 系统模块测试 34](#_Toc195905882)

[6.2.1 用户模块测试 34](#_Toc195905883)

[6.2.2 文章模块测试 34](#_Toc195905884)

[6.2.3 互助发帖模块测试 34](#_Toc195905885)

[6.2.4 用户交流模块测试 35](#_Toc195905886)

[6.2.5 后台管理模块测试 35](#_Toc195905887)

[7 总结与展望 36](#_Toc195905888)

[7.1 研究工作总结 36](#_Toc195905889)

[7.2 系统未来展望 36](#_Toc195905890)

[参考文献 38](#_Toc195905891)

[致谢 40](#_Toc195905892)

1 绪论

本章主要介绍本课题研究的来源及意义，国内外研究现状，主要研究的目的及内容，以及研究的方法和设计思路。

1.1 课题来源及意义

1.1.1 课题的来源

在当今大学教育环境中，互联网技术的迅猛发展极大地改变了信息传播与获取的方式。互联网具有信息量大、传播速度快、交互性强和影响力广泛等特性，为学生突破专业局限、开展跨专业学习提供了前所未有的条件。据调查，大学生通过网络查找资料、参与在线课程、论坛讨论已成为他们获取知识的主要途径。

同时，知识经济时代对人才的需求日益多元化，跨专业学习对于拓宽个人知识视野、提升综合素养以及增强职场竞争力的重要性愈发凸显(孙赤婴，2024)。然而，现实中跨专业学习面临着诸多阻碍。

一方面，高等教育专业划分细致，导致学生知识局限性明显。有研究指出，基于学科取向的科学研究和专业教育模式在高校中仍占主导地位，学科知识发展的高度分化与人才知识需求的高度综合之间存在矛盾。这种专业划分使得学生在跨专业学习时，难以找到系统全面且适合自身水平的学习资料，学习计划推进困难。

另一方面，现有的学习平台存在诸多问题。相关研究表明，现存的学习平台离散化、碎片化严重，忽略高阶能力培养。同时，这些平台课程关联度低，缺乏个性化学习引导，不同专业知识融合性差，无法满足学生跨专业学习的需求。

在此背景下，本课题以打造综合性跨专业学习网站为切入点，具有重要的现实意义。通过充分利用前沿的互联网技术，构建一个资源丰富、易于使用、学习模态多样的跨专业学习网站，有望解决传统学习资源分散与高效整合的矛盾，以及现有学习平台存在的问题，为学生提供更好的跨专业学习支持，满足知识经济时代对复合型人才的培养需求。

1.1.2 课题研究的意义

在高等教育专业划分细致的背景下，学生往往局限于自身所学专业知识范畴内。此跨专业互助学习系统，能让学生接触到其他专业同学分享的学习经验、学习方案、实践案例等，使其不再受限于单一专业知识体系，极大地拓宽了知识视野，帮助学生构建更全面、多元的知识架构。同时，系统提供多样化的互动交流学习方式，例如互助聊天，与文章作者交流学习，通过AI总结获取知识等等，这一系列设计有助于调动学生学习的积极性和主动性，使其更主动的投入到学习中。

1.2 主要研究的目的及内容

本课题的主要研究目的，旨在通过开发基于 Web 的大学生跨专业互助学习系统，帮助学生突破传统专业壁垒，使其接触其他专业核心内容、前沿动态与实践案例以激发创新思维，另一方面为学生学习中遇到的难题提供及时帮助并激发学习兴趣，进而促进不同专业领域学生间的交流学习，营造良好的跨专业学习氛围。在研究内容方面，涵盖系统架构搭建，前端选用 Vue3框架、后端采用 Gin框架，使用MySQL 数据库存储基本数据，借助 WebSocket构建聊天信息平台，使用Protocol buffer协议进行信息流的传输。同时接入豆包语言大模型提供文章AI总结功能；在功能模块开发上，着重打造文章、视频、热点管理，互助聊天、AI 总结问答等基础功能，支持用户点赞、收藏、评论文章及与作者聊天学习等互动功能；通过多样化互动交流和学习方式，激发学生兴趣，提高其积极性和主动性，最终实现提升跨专业学习效果与促进专业领域交流融合的目标。

1.3 研究方法及设计思路

1.3.1 研究方法

通过广泛查阅国内外有关跨专业学习、在线学习平台(郭思嘉，2018)等多方面文献资料，为系统设计开发提供理论支撑，确保研究科学合理。接着运用调查研究法，通过设计调查问卷并实地访谈高校学生群体，了解学生跨专业学习实际需求等情况，为功能模块设计找准现实依据。同时利用案例分析法，了解学习的在线学习平台案例，借鉴其优势经验融入到本系统开发中，提升系统质量。

1.3.2 设计思路

（1）用户需求导向：基于前期通过调查研究法所获取的大学生跨专业学习的实际需求，明确系统要解决的关键问题，如打破专业壁垒、提供个性化学习引导、优化互动交流体验以及方便学习资源获取等。以满足这些核心需求为出发点，来规划系统整体的功能框架和操作流程，确保系统能够切实贴合学生的使用场景，提高其对系统的认可度和使用率。

（2）功能架构规划：从功能层面进行系统架构设计，将其划分为多个功能模块，包括文章、视频、热点管理模块，用于整合和展示多专业的优质学习资源；互助聊天模块，方便不同专业的学生实时交流互动、答疑解惑；AI 总结问答模块，借助智能技术辅助学生快速获取知识要点、解答学习疑问等。同时，设计用户的点赞、收藏、评论等互动功能，增强用户参与感与学习趣味性，通过各功能模块之间的协同配合，构建起一个功能完备、操作便捷的跨专业学习平台。

（3）技术实现保障：根据系统功能需求，选择合适的技术框架来确保系统的稳定运行与高效实现。前端采用 Vue3 框架，注重打造简洁、美观且易用的用户界面，提升用户交互体验；后端运用 Gin 框架(肖睿，2019)，保障系统的高效处理能力和良好的扩展性。利用 MySQL 数据库存储如用户信息、学习资源等基本数据，借助WebSocket构建实时通信的聊天信息平台，使用Protocol buffer协议高效传输实时信息流，并且接入豆包语言大模型，为用户提供智能对话接口，运用这些技术手段共同支撑系统各项功能的顺利实现，为学生提供优质的跨专业学习服务。

1.4 本文的主要工作及论文结构组织

1.4.1 主要工作

（1）明确系统开发的背景和目标：针对高等教育专业划分细致导致学生知识局限的问题，确定开发基于 Web 的跨专业互助学习系统，旨在打破专业壁垒，为学生学习难题提供帮助，激发学习兴趣，促进不同专业学生交流。

（2）开展系统设计：前端采用 Vue3 + Vite 技术栈，利用 Vue3 的响应式机制和 Vite 的快速构建特性打造良好用户界面；后端使用 Gin 框架结合 Go 语言，借助其高效路由和中间件提升性能。选用 MySQL(Wei Z，2023) 存储结构化数据，Redis 实现缓存功能，通过 WebSocket 构建聊天平台并接入豆包语言大模型。设计账户、文章、互助帮学、用户交流、后台管理等功能模块。

（3）实现系统功能：完成用户注册登录、文章管理（创建编辑、浏览等）、互助功能、用户交流（私聊、音视频通话等）、后台管理等模块的开发，还基于 Formily 封装表单组件提高开发效率。

（4）进行系统测试：设计测试用例对系统各模块进行测试，涵盖用户、文章、互助发帖、用户交流和后台管理模块，确保系统功能正常、稳定运行。

1.4.2 论文结构组织

（1）绪论：介绍课题来源，阐述研究意义，说明主要研究目的和内容，介绍研究方法和设计思路。

（2）相关技术简介：分别介绍前端、后端、数据库和WebSocket的技术原理，并介绍了系统的开发工具和运行环境等。

（3）系统需求分析：进行技术、运行、时间可行性分析，明确系统总体需求，功能模块及大致功能，细分系统各模块具体操作。

（4）系统整体设计：阐述Web系统设计准则，介绍系统总体结构，描述具体功能模块设计和数据库设计。

（5）系统详细设计与实现：详细说明各功能模块如用户注册登录、文章管理、用户交流模块等核心模块的实现以及系统表单组件的实现过程和原理。

（6）系统测试：说明测试目的，展示各模块测试用例及结果，验证系统功能和稳定性。

（7）总结与展望：总结研究工作，包括系统开发过程、遵循原则和实现成果；对系统未来进行展望，提出在功能性、社交互动、性能、平台拓展性等方面的优化方向。

2 相关技术简介

本章主要介绍实现跨专业互助学习系统所使用到前后端技术栈，包括了Vue、Vite、Gin等等。

2.1 前端技术简介

在前端的技术选型上，整体使用Vue 3+Vite作为前端页面开发的工具。Vue 3 是渐进式 JavaScript 框架 Vue.js 的最新主要版本，而 Vite 是一个基于原生 ES 模块提供支持的构建工具。

Vue 3摒弃Vue 2 (Quan Y，2019)中 Object.defineProperty 的实现方式，转而采用 Proxy 来构建响应式机制。在复杂的数据结构频繁更新的场景下，Vue 3 的响应式系统能够更迅速地响应变化并更新相应的 DOM 元素，有效提升了应用的响应速度和流畅度。在学习曲线上，Vue 3 的语法和 API 设计相对简洁直观。Vue 3 在组件的定义和使用上，采用了类似 HTML 标签的方式来挂载组件，代码结构清晰，易于阅读。而 React 则依赖于 JavaScript 的函数式编程概念，像 JSX(JavaScript XML)语法，需要开发者对 JavaScript 函数、返回值等有更深入的理解，对于新手来说可能存在一定的学习门槛。Vue 3 的官方文档非常详细且条理清晰，示例丰富，从基础概念到高级应用都有详细的讲解和代码示例，方便开发者快速查找和学习相关知识。Vue3框架原理如图1所示。

图示

AI 生成的内容可能不正确。

图1 Vue原理

Vite是一个超快的前端构建工具。在传统的前端开发构建工具中，启动开发服务器往往需要对整个项目进行打包等预处理操作，这一过程随着项目规模的增大，耗时会显著增加。而 Vite 利用了现代浏览器原生支持 ES 模块的特性，在开发环境下省略了繁琐的打包过程，直接基于源文件启动开发服务器，从而实现了极快的冷启动速度，大大缩短了开发者等待项目启动的时间，提高了开发效率，让开发者能够迅速进入开发调试环节。Vite在小型项目中的热更新表现相当出色，当项目中的某个文件发生变化时，它能够精准地识别出变化的模块，并仅对该模块进行重新编译和更新，而无需像传统构建工具那样重新构建整个应用。Vite 采用按需编译的策略来优化项目的构建过程。不同于传统构建工具会一次性对所有模块进行编译处理，Vite 只会在浏览器实际请求某个模块时，才对其进行编译并提供给浏览器。这一策略避免了对那些暂时未被使用到的模块进行不必要的编译工作，有效减少了编译时间和资源消耗，特别是对于大型项目，按需编译能够显著提升整体的开发和构建效率。Vite框架ES模块原理如图2。

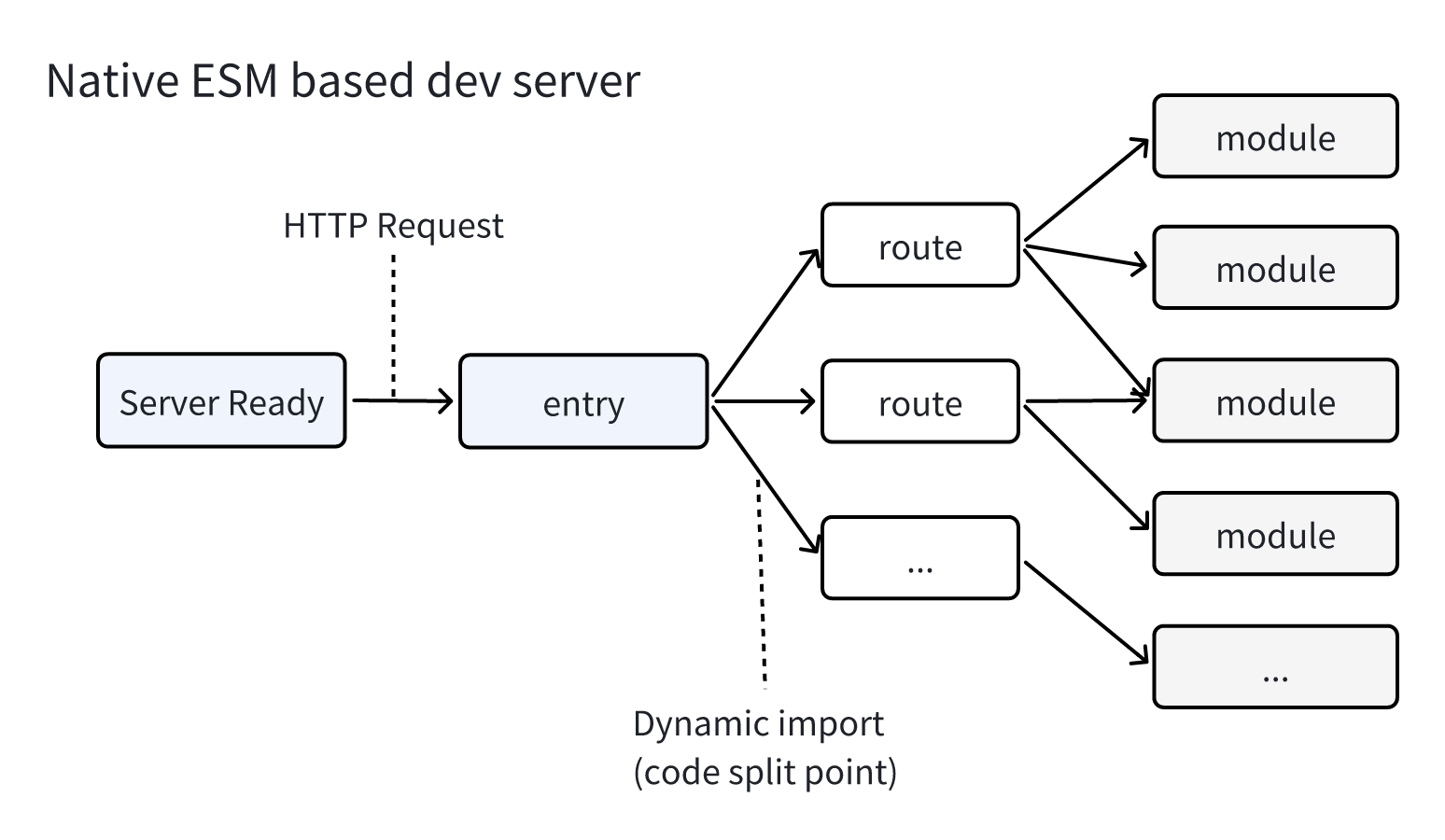


图2 Vite 原理

2.2 后端技术简介

Gin框架是一个用 Go 语言(王铮清和刘壮峰， 2022)编写的轻量级 Web 框架，由 Golang 开发者实现。它的设计目标是高效、快速且易于使用。Gin 基于 Radix 树实现路由，这种数据结构使得路由查找速度极快，能够高效处理大量的并发请求。并且它的中间件机制设计合理，开销小，有助于提升整体性能。此外，Gin提供了简洁明了的 API，开发者可以轻松上手。它采用了类似 Express.js 的路由定义方式，让熟悉 Node.js 的开发者也能快速适应。同时，Gin支持多种中间件，如日志记录、错误处理、跨域处理等。开发者可以根据需求自定义中间件，增强应用的功能。Gin 还内置了对 JSON 数据的验证功能，方便开发者对请求数据进行验证和处理。Gin框架整体架构如图3所示。

图示

描述已自动生成

图3 Gin框架结构

2.3 数据库简介

MySQL是一款开源的关系型数据库管理系统，具备 ACID（原子性、一致性、隔离性、持久性）特性，能确保在复杂的事务操作中数据的准确性和完整性，适用于对数据完整性、准确性要求较高的各类业务场景。

Redis(Guo Y，2012)是一款开源的高性能键值对（Key-Value）存储数据库，常用于缓存、消息队列、实时统计等场景，它的数据存储在内存中，因此读写速度极快。数据存于内存，使得其读写操作的响应时间极短，能够快速地获取或更新数据，非常适合作为缓存层来减轻后端数据库的压力，提高应用整体性能。

2.4 WebSocket简介

WebSocket 协议(李仲岐，2023)是一种实现客户端（通常是浏览器中的网页脚本等）与服务器之间的全双工通信，也就是允许服务器主动向客户端推送信息，同时客户端也能随时向服务器发送消息。WebSocket 建立连接后，只要连接保持打开状态，双方就能随时互相发送数据，这样极大地提高了通信的实时性和交互性。WebSocket凭借其独特的优势，在实时聊天应用、实时数据推送、协同办公与写作工具等诸多需要实时交互的应用场景，发挥着不可或缺的作用。

2.4 开发工具及运行环境

开发工具有：

VSCode：编程开发的集成工具，配合智能化插件高效开发系统。

Navicat：数据库管理工具，通过运行SQL语句管理数据信息。

Chorme浏览器：用于系统的查看和调试

系统的运行环境：

操作系统：Window 11 / Mac

Go版本：Go 1.23 及以上

Node版本“16.0 以上

数据库 MySQL 5.7以上

3 系统需求分析

3.1 系统可行性分析

3.1.1 技术可行性

本跨专业互助学习系统选用 Vue3、Gin 框架、WebSocket、MySQL等技术栈。在前端开发中本文使用Vue3，其采用组件化的开发模式，使得代码结构清晰明了，具备很强的可复用性。同时，基于响应式原理，能够打造出交互体验良好、流畅度颇高的用户界面，为学习者提供便捷、舒适的操作环境，十分契合系统前端展示以及交互操作的功能需求。而且 Vue3 在前端开发领域已经占据了重要地位，一直在不断更新迭代，性能和稳定性都有可靠保障，也经过了大量项目实践的检验，技术成熟度高，遇到常见问题很容易在相关技术社区找到解决方案。

Gin框架在后端开发中方面发挥关键作用，它是一个轻量级的 Web 框架，拥有高效的路由处理机制以及中间件支持功能，能够快速搭建起稳定可靠的服务端，轻松应对系统运行过程中的大量并发请求，保障系统后端服务的稳定运行，有力地支撑起系统各项业务逻辑的实现，在众多 Web 开发项目中被广泛应用，有着成熟的技术生态和丰富的开发案例可供参考。

WebSocket术专注于实现实时通信功能，对于跨专业互助学习系统来说意义重大。它能够让不同专业的学习者在系统中进行实时的交流互动，比如在线讨论问题、实时答疑解惑等，极大地增强了系统的交互性和实时性，并且其技术规范成熟，在各类需要实时通信的应用场景中都有着出色的表现。

MySQL具备强大的数据存储和管理能力，能够高效、稳定地存储系统中的用户信息、学习资料数据、交流记录等各类结构化数据，且有着完善的事务处理机制，保证数据的一致性和完整性，同时有着丰富的工具和插件支持，方便开发过程中的数据库操作与维护。

在硬件方面，常规的服务器设备就能满足系统开发以及初期运行的基本需求，随着系统用户数量的逐步增加以及功能拓展，可以依据实际情况灵活地进行硬件资源的扩展升级。开发过程中使用的集成开发环境为 Visual Studio Code ，能提供丰富多样的功能以及实用的插件。在AI辅助开发功能日益发展的今天，搭配上Copilot，MarsCode插件后，可显著提高开发效率。

3.1.3 运行可行性

从使用习惯上看，系统使用简洁直观、易于理解的页面设计，类似常见社交平台或者学习类APP的操作模式，便于使用者快速上手该系统。从人力配置来看，在系统运营初期，只需要少量的维护人员即可完成系统的使用，人力需求相对容易满足，运行可行性高。从成本上看，所用技术和工具均为开源免费的，成本主要集中在服务器的租赁和大模型接口的调用Token费用，后期运营维护成本在可控范围内。综上所述，本系统运行可行性高，值得开发。

3.1.4 时间可行性

从时间上看，系统开发花费的时间人力成本大概为90pd。在已经具备基本的技术栈背景下，在前两周，会进行详细的需求分析，明确系统的功能架构和模块区分。在后续的一个半月，可以根据前期设计，展开编码工作，优先实现核心功能模块，同时有序推进各个模块的开发。在最后的时间，进行系统的集成测试，漏洞修复，使得系统达到预期要求。

3.2 系统总体需求

通过跨专业互助学习系统，可以拓宽学生的视野，让学习了解各个学科中的独特魅力和价值。在功能模块上，系统包含用户管理、文章增删查改、文章收藏、点赞、评论、聊天等等功能，在系统用户层上，将用户分为管理员和普通用户，并赋予不同的功能，具体实现如图4、图5所示。

用户管理：系统支持通过邮箱验证注册的方式，提供账号密码找回功能，通过邮箱验证重置密码，未来也会支持通过社交账号的登录功能。

文章管理：登录后的用户可以通过文章创作中心进行文章的创建发布管理，在管理员审核后，文章即可被浏览、点赞、收藏、转发、评论，同时提供了文章访问、评论、点赞量等信息统计。

互助帮学模块：在互助帮学模块中，用户可以发布帖子到社区中，帖子审核通过后即可被所有用户浏览到。所有用户均可在帖子内进行评论，同时可以通过帖子中的发布人信息私聊用户。

用户交流模块：系统提供私聊功能，在聊天界面可以发送文本、表情、图片、文件、语音等内容。同时还可以和用户进行一对一的音视频聊天，达到一对一互助帮学的效果。

图示

AI 生成的内容可能不正确。

图4 用户用例图

图示

AI 生成的内容可能不正确。

图5 管理员用例图

3.3 系统功能需求

3.3.1 账户模块

账户模块主要负责用户注册、登录、退出、密码找回等操作。

用户没有登录时，为游客状态，只能浏览文章和用户发布的互助帮学内容，不能对文章进行任何操作，包括点赞、收藏等等。在注册界面用户填写完成用户名和密码即可正常登录系统。

在登录注册界面，提供重置密码功能，可以通过邮箱验证，完成对密码的重置。登录后，用户点击右上角自己的头像时，会出现退出选项，点击退出按钮就可以退出系统。

3.3.2 文章模块

文章模块是系统的核心功能之一，登录后的用户可以创建文章，并给文章打上标签、话题、摘要等内容，点击发布文章，待管理员审核通过后即可在主页浏览到对应文章。每篇文章都可以根据AI大模型生成一段概述，这段概述在浏览文章时候可以看到。

用户可以通过关键字搜索文章，文章可以进行点赞、评论、收藏、转发、沉浸式阅读。浏览文章的同时，也会展示了文章作者的相关信息，通过点击按钮作者和行私聊。

3.3.3 互助帮学模块

在请求帮学模块中，用户可以发布一个简单的帖子，内容可以是提问问题、资料查询等等、学习路线咨询等等。帖子中可以有图片表情、文字内容，用户发帖之后需要通过管理员审核后帖子方能被所有人浏览。

3.3.4 用户交流模块

在用户交流模块中，系统支持表情、图片等信息的发送功能，也提供了语音内容、文件内容的发送。同时系统也提供视音频聊天的功能，通过发起音视频聊天的按钮向在线用户进行音视频聊天。

3.3.5 后台管理

后台管理模块对网站的平稳运行、学习资源质量保障以及用户使用体验的提升起着不可或缺的关键作用。

只有管理员用户才可以进入后台管理页面，后台管理包括了用户管理、文章管理、互助内容管理、公告配置功能。在用户管理模块，可以修改、删除、恢复用户信息。在文章管理模块、互助管理模块可以审核、删除内容，只有审核通过的内容才可以被用户浏览到。审核通过的内容可以修改为审核不通过。在公告内容的配置中，管理员可在此功能模块中进行各类公告信息的编辑、发布与管理操作。

3.4 本章小结

本章对跨专业互助学习系统进行了细致的需求分析一是可行性分析，从技术、运行、时间三方面说明，所选技术成熟可靠，运行易上手且成本可控，开发时间人力成本有相应安排。二是总体需求，介绍系统功能模块及各模块大致功能，含用户、文章、帮学、交流、审核等方面，配有对应用例图。三是系统功能需求，细分账户、文章、互助帮学、用户交流、后台管理等模块，阐述各模块具体操作及管理员相应职责。

4 系统整体设计

本章主要介绍在设计跨专业互助学习系统时采用的设计思想和细节技术栈，同时阐述了选择该项技术的主要原因。根据网络上的最佳实践，从系统总体结构分析到各个功能的详细分析，细致的描述了一遍整个系统的技术路线。

4.1 系统整体设计

4.1.1 Web系统设计准则

作为一个基于Web的项目系统，应该满足易用性的要求，同时具备拓展性、兼容性，在性能方面也要达到反馈速度迅速且稳定，在代码方面，也需要有良好的维护性，以保证后续的维护。总的来说，整个系统的设计应该遵循以下的原则：

易用性：界面的设计要符合用户操作习惯，操作流程简洁明了，方便不同技术水平的用户轻松上手，降低用户的学习成本。

可维护性：代码结构清晰、易于理解和修改，方便开发团队后续进行功能优化、问题修复等维护工作，遵循良好的编程规范和设计模式。

安全性：对用户的隐私数据（如账号密码、个人信息、交易记录等）进行加密存储和传输，防止数据泄露。具备防范常见网络攻击（如 SQL 注入、跨站脚本攻击等）的能力(胡敏，2015)。

4.1.2 系统总体结构

系统基于前后端分离的设计理念，采用功能化分层的设计思想，将系统划分为，表现层、服务层、数据层，具备良好的维护性和可拓展性。系统整体结构如图6所示。

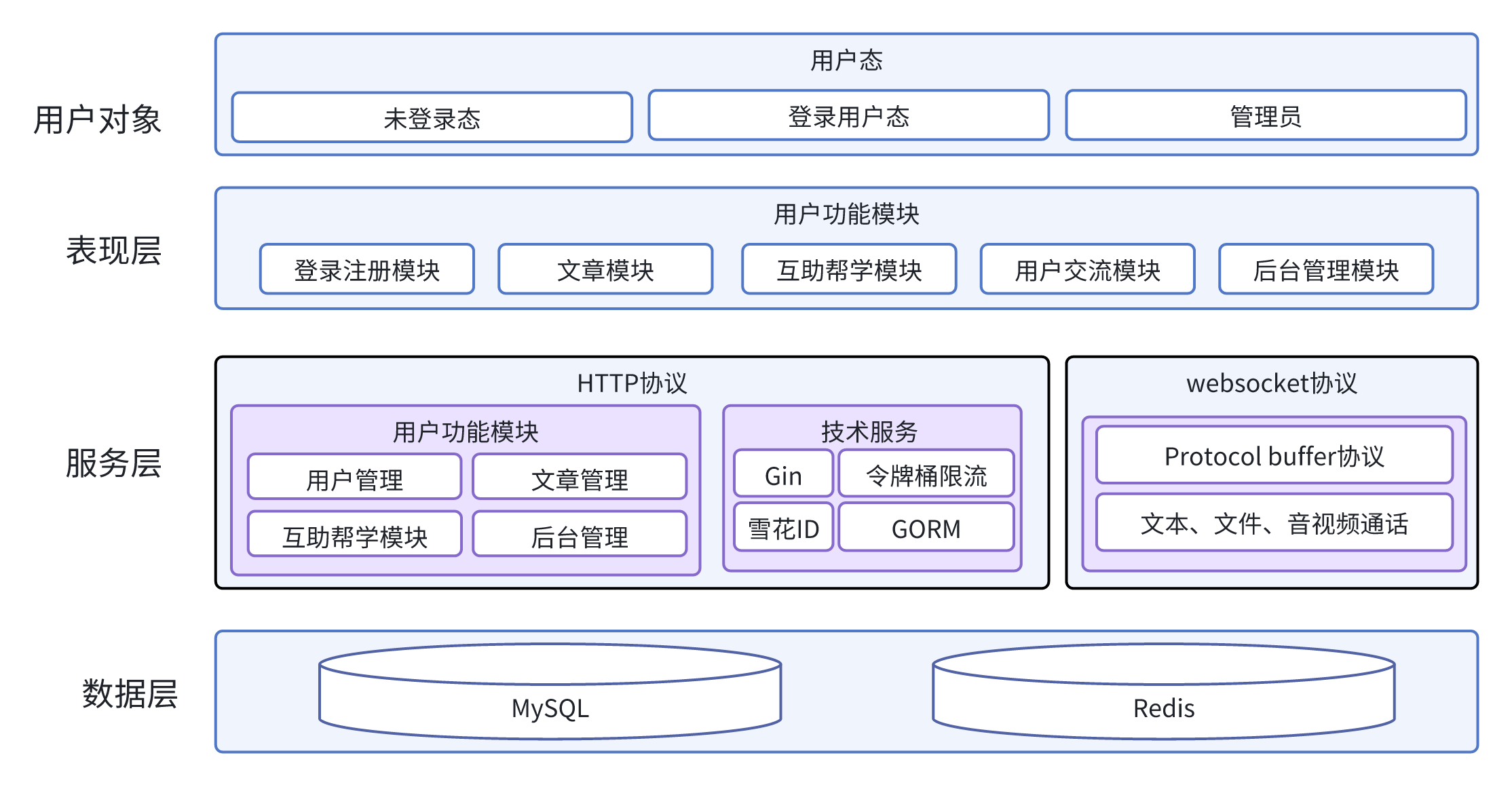


图6 系统架构图

表现层作为用户与系统接触的第一道 “桥梁”，是整个系统直观呈现在用户眼前的部分，也是用户与之产生交互的首要界面。表现层依托于Vue3框架构建，使用组件化，结构化的思想，对页面总体设计成上中下三栏布局，分别对应导航区域，内容区域，底部信息透出区域。使用Vue-Router实现内容区域流畅切换，使用Pinia作为全局的状态管理，封装了通用的请求响应拦截器，实现API的统一管理，以及更多的状态同步功能，以此应对复杂的交互场景。针对拓展性原则，使用了Formily表单库，封装了表单系统，对表单的生命周期进行全面的封装，包括表单初始化、回填、联动、校验等内容，通过一份JsonSchema即可渲染表单，提升表单的开发效率。

服务层整体采用Go语言编写，使用Gin框架，将各个功能模块拆分，使用Gin框架路由，基于功能模块做分组，接收客户端请求。同时使用令牌桶算法，限制了客户端请求数量。在服务端设计上，整体封层为Controller、Logic、Dao三层经典架构设计，这种分层架构设计使得服务层的代码结构清晰，各层职责明确，便于团队协作开发和系统的长期维护。同时，结合功能模块拆分和请求限流机制，为系统的高性能、高可用性和可扩展性提供了有力保障。

在数据层中，服务层借助 Gorm 这一强大的 ORM（对象关系映射）库与关系型数据库展开交互。Gorm 能把数据库中的表、行、列等概念，自然地映射为 Go 语言中的结构体、对象和字段，极大地简化了数据库操作的代码编写工作。在系统架构中，使用关系型数据库与 Gorm 交互。

4.2 具体功能模块设计

4.2.1 用户层设计

在跨专业互助学习系统前端页面中，用户存在有三种状态，未登录态、登录态、管理员。三种状态分别对应有不同的功能。对于需要用户信息的功能，例如创建文章，点赞、评论、视频、提问等都需要用户进行登录注册之后才可以进行操作。

对于用户操作时那些需要用户信息的操作时，会提醒用户进行登录注册。对已登录用户，则无限制。

登录页面中，提供了账户密码的登录功能，同时也提供找回密码功能，通过邮件信息获取验证信息，重置密码。在注册页面中，需要填写用户名、昵称、邮箱、头像等信息，需要通过邮箱获取验证码并验证后方可以注册成功。注册成功直接跳转登录界面。

4.2.2 文章管理设计

在文章管理中，用户在文章创建界面对文章进行编辑创作，发布前需填写文章分类信息、话题、封面、摘要等内容。文章发布后，系统会自动调用大模型接口，获取整篇文章的 AI 总结信息。待系统管理员审核通过后，文章才可以被用户浏览、点赞、评论等等。

4.2.3 互助帮学设计

在互助帮学模块中，用户登录后，即可在互助帮学界面发送互助信息，互助信息内容可以有文字、表情、图片信息等等，发布互助信息前也需要选择帖子所属模块标题等信息，在经过管理员审核之后便可在互助模块中被其他用户浏览、评论。

4.2.4 用户交流模块设计

在用户交流模块中，用户可以通过评论、文章、互助信息的用户简介跳转到聊天信息页面。进入聊天信息界面后发起WebSocket(Cao W ,Tan X，et al，2016)请求，链接到服务端中的连接池里，进行一对一的交流。选用Protocol buffer协议作为信息传输协议，因为Protocol buffer序列化后的大小是Json的10分之一，是xml格式的20分之一，但是性能却是它们的5~100倍。。对普通文本信息直接存储在数据库中，对与文件信息则存储在服务器后，获取对应URL后再存储到数据库中。对于视频聊天和语音聊天功能，使用WebRTC(唐超，2024)，通过getUserMedia方法获取用户的摄像头和麦克风媒体流，在经过交换SDP、交换ICE两次信息内容交换后，建立链接并传输媒体流，实现音视频通话功能。

4.2.5 后台管理模块设计

在后台管理模块中，有用户管理、文章管理、互助帮学管理、公告配置四大功能。在三个管理页面中，主要由上方的筛选框和下方的表格组成，通过各种筛选配置，进行数据的筛选，映射在表格中。表格中除了有基本的数据信息外，还有操作列，可以进行编辑、删除、审核等操作，点击编辑会弹出编辑弹窗表单，供给管理员进行编辑，点击删除会进行二次询问，确定后进行删除操作。

4.3 数据库设计

4.3.1 整体数据结构设计

本跨专业互助学习系统的数据结构设计，通过多个数据表以及严谨的外键关联关系，充分考虑了用户、内容（文章、互助学习资料）、分类、标签、话题以及用户互动等多方面的需求，实现了跨专业的知识分享、互助学习以及用户之间的交流互动等功，各个表的字段设计和约束设置也有助于提高数据的准确性、完整性以及查询效率等。

4.3.2 数据库物理结构设计

本系统涉及的表结构有用户信息表、文章信息表、文章评论表、互助信息表、互助信息评论表、点赞表、收藏表、关注表、用户聊天列表，聊天消息表一共10张数据表。下表1至表7是数据库主要数据表信息。

表1 用户信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 默认值 | 字段描述 |
| user\_id | varchar(20) | 0 | 用户ID |
| user\_name | varchar(20) |  | 用户名 |
| nickname | varchar(20) |  | 昵称 |
| email | varchar(320) |  | 邮箱 |
| phone\_number | varchar(11) |  | 手机号 |
| password | varchar(255) |  | 密码 |
| confirm\_password | varchar(255) |  | 重复密码 |
| overview | varchar(255) |  | 简介 |
| gender | tinyint | 0 | 性别 0:未知 1:男 2:女 |
| is\_email\_verified | tinyint(1) | 0 | 邮箱是否已验证 1为验证 |
| avatar | varchar(255) |  | 头像信息 |
| created\_at | datetime | currenttime | 注册时间 |
| updated\_at | datetime | currenttime | 更新时间 |

表2 文章信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 默认值 | 字段描述 |
| article\_index | int unsigned |  | 文章序号 |
| article\_id | varchar(20) |  | 文章ID |
| title | varchar(255) |  | 文章标题 |
| content | text |  | 文章内容 |
| user\_id | varchar(20) |  | 文章作者 |
| category\_id | int |  | 分类id |
| tag\_id | int |  | 标签id |
| abstract | varchar(200) |  | 文章摘要 |
| cover | varchar(200) |  | 文章封面 |
| gpt\_summarize | varchar(200) |  | gpt生成的文章摘要 |
| view\_count | int | 0 | 浏览量 |
| comment\_count | int | 0 | 评论数量 |
| like\_count | int | 0 | 点赞数量 |
| created\_at | datetime | currenttime | 创建时间 |
| updated\_at | datetime | currenttime | 更新时间 |
| article\_status | int | 0 | 文章状态0未审核1通过 |

表3 互助信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 默认值 | 字段描述 |
| assistance\_index | int unsigned |  | 互助序号 |
| assistance\_id | varchar(20) |  | 互助ID |
| title | varchar(255) |  | 互助标题 |
| user\_id | varchar(20) |  | 用户ID |
| category\_id | int |  | 分类id |
| tag\_id | int |  | 标签id |
| user\_id | varchar(20) |  | 用户ID |
| topic\_id | int |  | 话题id |
| cover | varchar(200) |  | 互助封面 |
| created\_at | datetime | currenttime | 创建时间 |
| updated\_at | datetime | currenttime | 更新时间 |
| assistance\_status | int | 0 |  |
| content | text |  | 互助内容 |

表4 文章评论表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 默认值 | 字段描述 |
| article\_id | varchar(20) |  | 文章ID |
| comment\_id | varchar(20) |  | 评论ID |
| user\_id | varchar(20) |  | 用户ID |
| content | text |  | 评论内容 |
| create\_time | datetime | currenttime | 创建时间 |
| update\_time | datetime | currenttime | 更新时间 |
| parenet\_id | Varchat(20) | 0 | 评论父级id |

表5 互助评论表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 默认值 | 字段描述 |
| assistance\_id | varchar(20) |  | 互助id |
| comment\_id | varchar(20) |  | 评论id |
| content | text |  | 评论内容 |
| create\_time | datetime |  | 评论时间 |
| parent\_comment\_id | varchar(20) |  | 父级评论id |
| update\_time | datetime | currenttime | 创建时间 |
| user\_id | varchar(20) |  | 评论用户id |

表6 点赞表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 默认值 | 字段描述 |
| like\_id | int |  | 点赞id |
| article\_id | varchar(20) |  | 文章id |
| user\_id | varchar(20) |  | 用户id |
| liked\_at | timestamp | currenttime | 点赞时间 |

表7 聊天消息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 默认值 | 字段描述 |
| content | varchar(2500) |  | 消息内容 |
| content\_type | smallint |  | 消息内容类型：1文字 2.普通文件 3.图片 4.音频 5.视频 6.语音聊天 7.视频聊天 |
| created\_at | datetime(3) | currenttime | 发送消息时间 |
| deleted\_at | bigint unsigned |  | 删除消息时间 |
| from\_user\_id | bigint unsigned |  | 发送端id |
| to\_user\_id | bigint unsigned |  | 接收端id |
| id | int |  | 消息id |
| url | varchar(350) |  | 文件或者图片地址 |

4.4 本章小结

本章对跨专业互助学习系统设计进行了详细介绍。在系统总体设计方面，基于 Web 项目特性明确设计准则，涵盖易用性、可维护性、安全性等原则。其总体结构采用前后端分离与功能化分层理念，表现层用 Vue3 等实现页面交互与布局；服务层以 Go 语言结合 Gin 框架保障高效协作；数据层通过 Gorm 与关系型数据库交互并引入MySQL丰富存储。具体功能模块上，整体介绍了用户模块、文章模块、互助模块、沟通交流模块、后台管理模块的整体设计，同时也通过表结构将系统的SQL表字段详细信息列出，全方位呈现了该学习系统的设计思路与架构。

5 系统详细设计与实现

本章主要阐述各个模块中的主要功能实现，以及实现功能的原理步骤，展示系统的实际效果。

5.1 用户注册登录模块实现

（1）前端实现

登录注册模块主要内容为两个表单，在登陆页面，只需要收集用户的账号密码即可登录。在注册页面则还需要收集用户昵称、邮件、性别、头像等内容。两个表单通过JSONSchema和Formily进行渲染，在提交信息之前会通过Validator进行内容校验，校验通过方能发起请求，下图7、图8为登录注册界面样式。

图形用户界面, 网站

AI 生成的内容可能不正确。

图7 用户登录界面

图形用户界面, 网站

AI 生成的内容可能不正确。

图8 注册界面样式

（2）服务端实现

在登录操作中，会通过用户的账号、密码到数据库进行匹配，匹配成功后服务端返回Token，后续请求需要带上Token进行身份确认，后端在需要用户信息的请求中设置了验证Token的中间件，以验证用户身份。登录流程伪代码如下所示。

|  |
| --- |
| // 登录逻辑  function login(username, password) {  // 从数据库中查询用户信息  user = queryUserFromDatabase(username, password)  if (user exists) {  if (password == user.password){  // 生成 Token  token = generateToken(user)  // 返回 Token 给客户端  return token  }  // 登录失败，返回密码错误  return "Login failed: Invalid password"  } else {  // 登录失败，返回错误信息  return "Login failed: Invalid username or password"  }  } |

在注册逻辑中，前端输入邮箱信息后，点击发送验证码，服务端生成6位随机数通过smtp包发送邮件给用户，并将用户邮箱作为Key，验证码作为Value存储到 Redis 中，设置过期时间为5min。用户收到验证码后，需要输入验证码后，方能发起注册请求，在服务端验证通过后，会将密码加密处理，存储到数据库中。注册流程伪代码如下所示。整体注册登录时序图如图9所示。

|  |
| --- |
| // 注册逻辑  function sendVerificationCode(email) {  // 生成 6 位随机验证码  verificationCode = generateRandomCode(6)  // 将验证码存储到 Redis 中，设置过期时间为 5 分钟  storeInRedis(email, verificationCode, 5 \* 60)  // 通过 SMTP 发送邮件给用户  sendEmail(email, verificationCode)  return "Verification code sent successfully"  }  function register(user, password, verificationCode) {  // 从 Redis 中获取存储的验证码  storedCode = getFromRedis(user.email)  if (storedCode == verificationCode) {  // 验证码验证通过，加密密码  encryptedPassword = encryptPassword(password)  // 将用户信息存储到数据库中  saveUserToDatabase(user, encryptedPassword)  return "Registration successful"  } else {  // 验证码验证失败，返回错误信息  return "Registration failed: Invalid verification code"  }  } |

图示

AI 生成的内容可能不正确。

图9 登录注册时序图

5.2 文章管理模块实现

（1）前端实现

在文章模块，前端主要实现文章的创建和浏览两个界面。在创建界面，前端使用WangEditor编辑器，通过基本配置能够实现文字加粗、下划线、斜体、文字颜色、图片插入等基本的功能。在完成文章内容编辑后，点击发布按钮，弹出表单，填写文章摘要，封面，选择文章分类后即可发布。文章创建界面样式如图10所示。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

图10 文章创建

在首页中会请求文章列表接口，可以通过关键字查询和分类列表进行文章的筛选，被审核通过的文章才可以被浏览。点击已发布的文章能进入到文章的浏览界面，浏览界面为三栏布局，左侧为操作栏，可以进行点赞、收藏、评论等操作。中间为文章内容界面，通过设置WangEditor为ReadOnly模式，渲染文章内容，右栏为文章作者信息，可以通过右栏关注作者或者与作者私聊。文章浏览界面样式如图11所示，首页样式如图12所示。

图形用户界面, 应用程序, Teams

AI 生成的内容可能不正确。

图11 文章浏览页面

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图12 首页

（2）服务端实现

在单篇文章浏览逻辑中，会调用多个接口，通过文章id获取文章内容信息、文章作者信息、点赞、浏览、 收藏、评论信息并逐一返回给前端。

在文章创建逻辑中，前端携带文章信息，请求文章创建接口，在创建接口中，首先会将基本数据存储到数据库中，然后开启协程调用doubao-1-5-vision-pro-32k模型接口，对文章内容进行AI总结，总结失败会重新进行内容总结，直到生成AI总结内容。最后会将AI总结内容存储到数据库中，伪代码如下所示。文章模块流程图如图13所示。

|  |
| --- |
| // 文章创建接口函数，接收前端传来的文章信息  function createArticle(articleInfo) {  //将基本数据存储到数据库  var articleId = saveBasicArticleDataToDatabase(articleInfo);  if (articleId === null) {  return "Failed to save basic article data to database";  }  //开启协程调用模型接口进行 AI 总结  go func() {  var summary = "";  var retryCount = 0;  var maxRetries = 3; // 最大重试次数  for (retryCount = 0; retryCount < maxRetries; retryCount++) {  summary = callDoubaoModelForSummary(articleInfo.content);  if (summary!== "") {  break;  }  // 可添加重试间隔时间，例如等待5秒  wait(5 \* time.Second);  }  if (summary == "") {  log("Failed to generate AI summary after multiple attempts");  } else {  // 将 AI 总结内容存储到数据库  saveSummaryToDatabase(articleId, summary);  }  }();  return articleId;  } |

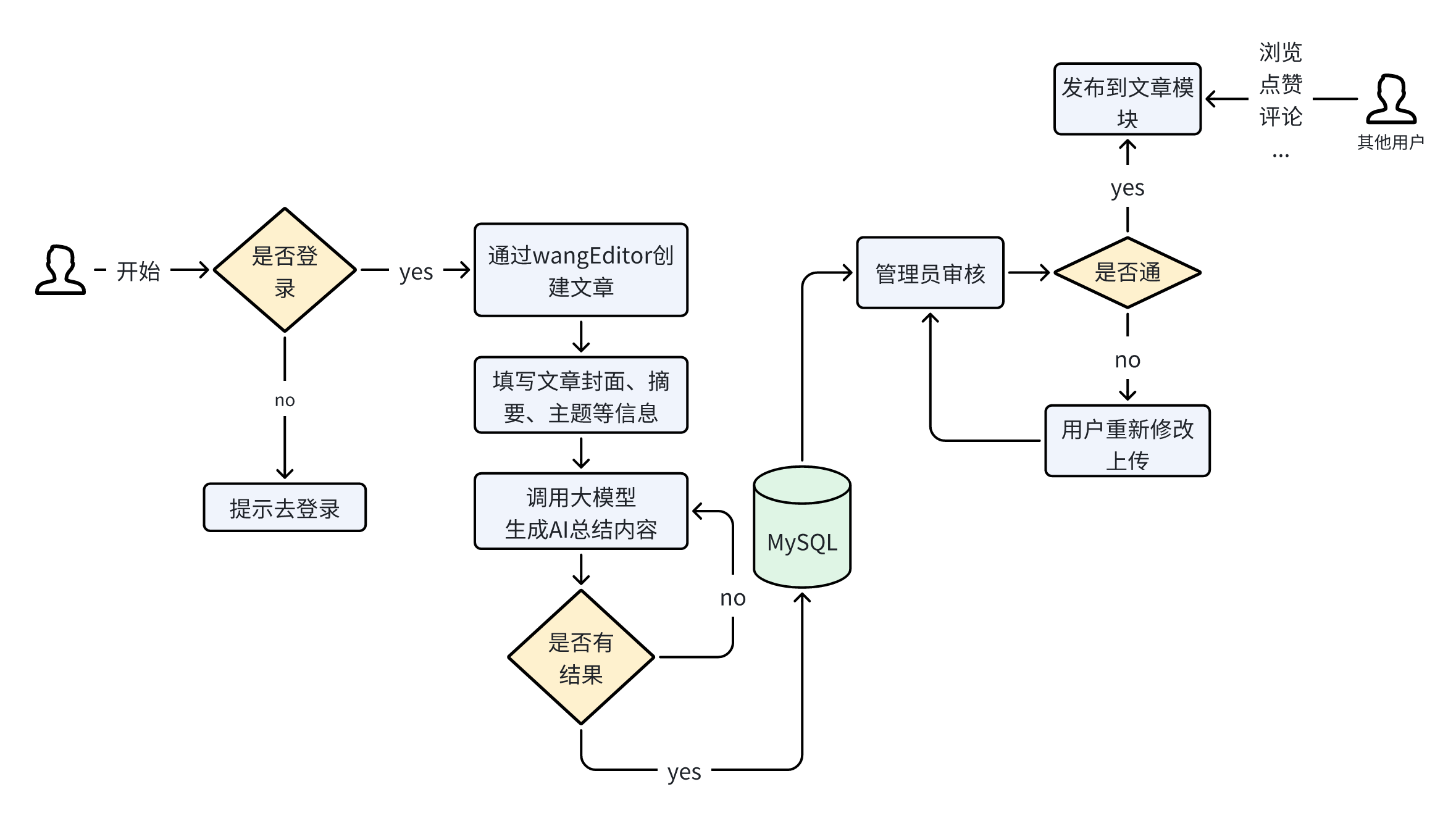


图13 文章管理设计

5.3 互助功能模块实现

（1）前端实现

互助帮学模块中，发布和内容浏览都在同一个界面，上方为互助发布模块，由输入框和发布按钮组成，点击发布按钮后需要填写互助内容信息后方可发布。下方为浏览界面，可以看到用户发布的互助内容，点击互助内容的评论Icon可以展开评论信息。互助帮学模块界面如图14所示。

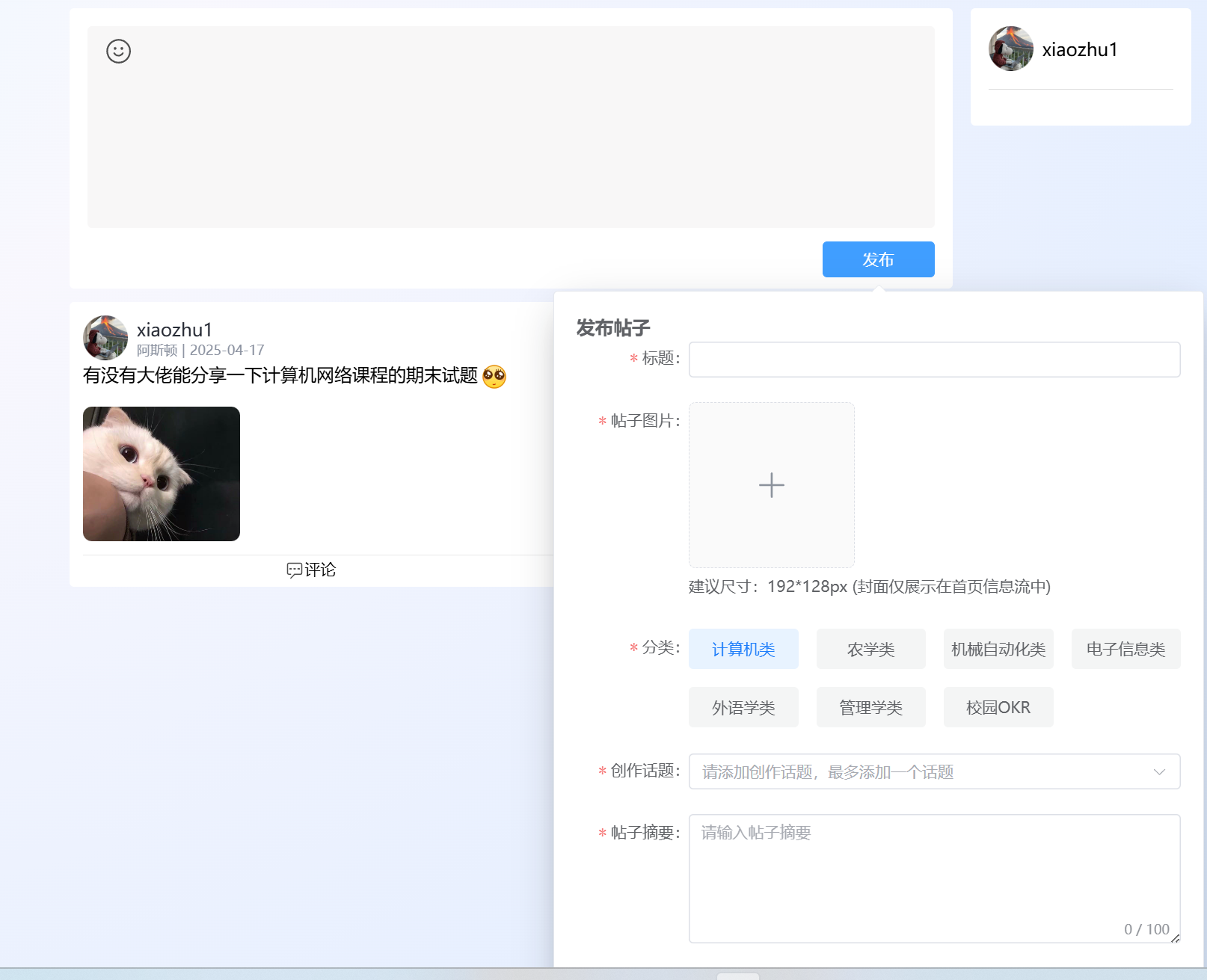


图14 互助帮学页面

（2）服务端实现

互助模块的服务端实现为简单的创建接口，服务端接收前端传送过来的信息后，直接存储到数据库中，管理员在后台审核过后，帖子方可在主页被浏览到。互助帮学模块设计流程图如图15所示。

图示

AI 生成的内容可能不正确。

图15 互助模块设计

5.4 用户交流模块实现

（1）前端实现

在用户交流模块中，可以通过文章浏览、互助信息浏览、评论内容中跳转到聊天界面。在聊天界面中，左侧为聊天列表，和你聊过天的用户展示在这个列表项上，右侧是聊天框和输入框。聊天框中的内容区分为普通文本信息，图片信息、语音信息、文件信息四大类信息，通过contentType字段区分信息种类。并选择不同的样式进行展示，对于普通文本信息通过Vue指令v-html进行内容渲染，对于图片信息通过ElImage组件展示，对于语音信息则通过audio标签展示，对于文件信息，则通过文件后缀名展示不同的icon。对于音视频通话功能拉起弹窗调用getUserMedia接口调用摄像头和录音设备，进行内容传输展示。

在输入框中，能通过icon进行不同信息的发送，包括文字、语音、文件、音视频通话功能。系统封装了sendMsg方法，在发送之前会通过Protocol buffer协议将发送内容转化为二进制流，发送给服务端。也封装了onMessage方法，对后端返回的信息进行统一处理，包括协议转化，展示处理，聊天界面样式如图16所示。音视频通话界面如图17所示。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

图16 用户交流界面

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图17 用户视频通话界面

（2）服务端实现

聊天功能主要区分为三大块，包括普通文本信息传递功能、图片等文件信息传递功能、音视频通话功能。

在用户进入聊天界面后，会向后端请求聊天列表接口，获取聊天列表，如果新的聊天对象，还会在聊天列表中添加用户信息。并通过uid进行WebSocket链接到连接池中，以进行后续的信息发送接收功能。每次发送信息都会携带toUid，服务端通过toUid 在连接池中查找用户并进行发送。

服务端在接收到来自客户端的信息后，会通过contentType字段区分文本信息类型，对于普通文本信息，会直接通过Websocket进行转发，并存储到信息数据表中，对于文件信息，例如语音、word文件、图片等先存储在服务器存储中，在获取url后再转发给用户。

对于音视频聊天功能，用户点击发起音视频聊天请求后，在前端会通过getUserMedia方法获取用户的摄像头和麦克风的媒体流，通过WebSocket向对方发送电话请求，待对方同意接通后，再通过WebSocket，进行SDP和ICE的交换，后续建立链接并传输媒体流，最终达到音视频通话的能力。服务端伪代码如下所示。整体流程图如图17所示。

|  |
| --- |
| // 服务端连接池  let connectionPool = {};  // 处理客户端 WebSocket 连接  function handleWebSocketConnection(socket, uid) {  connectionPool[uid] = socket;  socket.on('message', function(message) {  let parsedMessage = JSON.parse(message);  let toUid = parsedMessage.toUid;  let contentType = parsedMessage.contentType;  if (contentType === 'text') {  // 直接转发文本信息并存储到信息数据表中  forwardTextMessage(parsedMessage, toUid);  saveTextMessageToDatabase(parsedMessage);  } else if (contentType === 'file') {  // 存储文件信息，获取 url 后再转发给用户  let fileUrl = saveFileToServer(parsedMessage.file);  let fileMessage = {  contentType: 'file',  fromUid: parsedMessage.fromUid,  toUid: toUid,  url: fileUrl  };  forwardFileMessage(fileMessage, toUid);  } else if (contentType === 'call-request') {  // 转发音视频通话请求  forwardCallRequest(parsedMessage, toUid);  } else if (contentType === 'call-accept') {  forwardCallAccept(parsedMessage, toUid); // 转发音视频通话同意  }  });  socket.on('close', function() {  delete connectionPool[uid];  });  } |

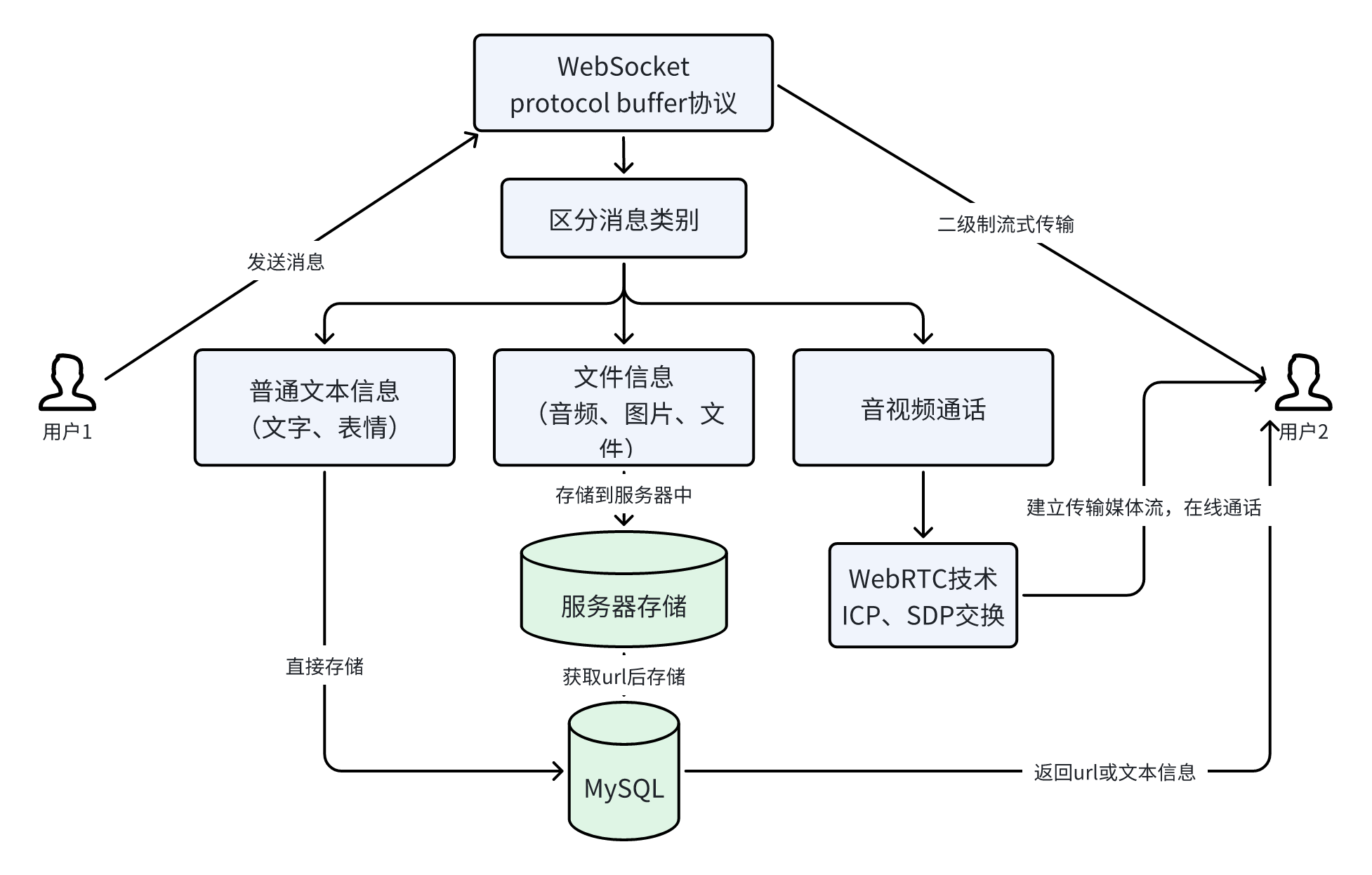


图17 用户交流模块功能

5.5 后台管理模块实现

（1）前端实现

在系统的后台管理模块中，提供了管理员管理用户账号、文章、求助信息、公告配置的内容。三个管理模块由筛选框表格页组成，在前端，封装了Filter-Bar过滤组件，通过配置项渲染筛选框，通过简单的配置信息即可渲染出顶层筛选框，每当筛选框内容改变的时候，会进行列表请求，获取新的数据。以此渲染到表格中。同样的，表格部分也是通过配置信息以及Table组件，进行渲染展示。表格除了基本的信息展示外，提供了操作列表，供给管理员进行编辑和删除操作，通过点击编辑按钮弹出弹窗表单，进行内容编辑。后台管理界面如图18所示。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图18 后台管理界面

（2）服务端实现

在后台管理中的服务端是相对与用户表、文章表、互助帮学表、配置表四张数据表的增删查改接口，通过Go语言中的结构体与数据表之间的映射，可以将Go语言中定义的结构体与MySQL中对应的表关联，再设置相应的查询条件结构体，通过Post请求，快速完成了四张数据表增删查改接口。

5.6 系统表单组件实现

本系统使用的表单基于Formily封装，Formily是阿里巴巴开源的一个表单系统，底层是对表单数据用 Es6 的 Proxy 代理，对数据进行劫持来做发布订阅，以达到更好的性能，无需做任何优化即可获得超高的性能优势，依赖追踪，精确更新，按需渲染，让表单真正做到了只需关注业务逻辑，无需考虑性能问题。Formily提供了 SchemaField 这样的协议驱动组件，同时是基于标准 JSON-Schema 的驱动，让表单开发可以变得更加动态化，可配置化。(纪乾坤，2023)基于Formliy，本系统在其基础上封装了一个通用表单组件，精简的整个表单系统的能力，只暴露出了表单初始化、回填、校验、提交、提交成功、提交失败等生命周期钩子，同时充分使用了JSON-Schema协议驱动表单的渲染，配置，将表单开发抽象成配置信息的开发，大大提高了表单能力的开发效率。

5.7 本章小结

本章详细的介绍了跨专业互助学习系统各个功能模块的实现。介绍了开发使用的技术栈以及详细的功能是如何开发，包括注册登录、文章管理、互助内容管理、后台管理等等。

6 系统测试

6.1 测试目的

系统测试是软件开发过程中的关键步骤，能够检查各个模块的功能是否正常运作，通过系统测试验证交互功能是否稳定运行，避免出现类似消息发送失败、无法及时回复等问题，保证跨专业学习交流环境的顺畅体验。本章节根据模块分区，进行了多次人工测试，通过设计测试用例，查询bug，并尽可能的保证系统核心功能的可用性，稳定性(郑荣茂，2005)。

6.2 系统模块测试

6.2.1 用户模块测试

表8 用户模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试功能 | 测试步骤 | 预期结果 | 实际结果 |
| 01 | 邮箱注册 | 1）用户信息少填  2）验证码不正确  3）信息填写正确 | 1）注册失败  2）注册失败  3）注册成功 | 符合预期结果 |
| 02 | 登录 | 1）错误填写用户名和密码  2）用户名和密码正确 | 1）登录失败  2）登陆成功 | 符合预期结果 |
| 03 | 个人信息修改 | 1）更新个人信息  2）重置密码 | 1）更新成功  2）修改成功 | 符合预期结果 |

6.2.2 文章模块测试

表9 文章模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试功能 | 测试步骤 | 预期结果 | 实际结果 |
| 01 | 文章创建/编辑 | 1）标题、摘要等信息少填  2）所有信息信息填写正确 | 1）创建失败  2）创建成功 | 符合预期结果 |
| 02 | 文章浏览 | 1）点赞、收藏  2）AI总结功能 | 1）符合预期  2）所有文章均有AI总结信息 | 符合预期结果 |
| 03 | 评论功能 | 1）评论信息正常展示 | 1）符合预期 | 符合预期结果 |
| 04 | 跳转私聊 | 1）点击作者、评论者头像 | 1）跳转到私聊页 | 符合预期结果 |

6.2.3 互助发帖模块测试

表10 互助发帖模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试功能 | 测试步骤 | 预期结果 | 实际结果 |
| 01 | 帖子创建/编辑 | 1）标题、摘要等信息少填  2）图片、表情、文字填写 | 1）创建失败  2）功能达到效果 | 符合预期结果 |
| 02 | 帖子浏览 | 1）图片、文字表情信息正常展示 | 1）符合预期 | 符合预期结果 |
| 03 | 帖子评论 | 1）帖子评论功能正常，用户可以进行评论，被评论用户收到信息提示 | 1）符合预期 | 符合预期结果 |

6.2.4 用户交流模块测试

表11 用户交流模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试功能 | 测试步骤 | 预期结果 | 实际结果 |
| 01 | 私聊功能 | 图片、表情、文字私聊  语音、视频、文件等内容发送 | 1）成功发送，对方接收成功，展示内容正常  2）成功发送，对方接收成功，可以下载 | 符合预期结果 |
| 02 | 音视频通话功能 | 发起音视频通话  接收方拒接/取消音视频通话要求 | 对方用户在线，拉起是否接通弹窗，接通后音视频传输正常。  挂断电话后停止调用系统设备信息 | 符合预期结果 |

6.2.5 后台管理模块测试

表12 后台管理模块测试结果表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试功能 | 测试步骤 | 预期结果 | 实际结果 |
| 01 | 用户管理相关 | 暂停/重启用户账号  编辑用户信息 | 1）用户账户被暂停后无法使用。  2）修改成功 | 符合预期结果 |
| 02 | 文章管理相关 | 1）审核文章  2）编辑文章  3）下架文章 | 1）审核成功，其他用户可浏览  2）编辑成功  3）用户查看不到已下架的文章 | 符合预期结果 |
| 03 | 互助帖子  管理相关 | 1）审核帖子  2）删除帖子 | 1）审核成功，其他用户可浏览修  2）删除成功 | 符合预期结果 |
| 04 | 公告配置 | 1）配置公告信息 | 1）前端页面正常展示公告 | 符合预期结果 |

7 总结与展望

7.1 研究工作总结

在毕业设计的选题过程中，本文聚焦于高等教育专业划分细致导致学生知识局限的问题，通过广泛查阅文献、调查研究和案例分析，敲定了一个基于Web的，涵盖了文章管理、资源分享、帖子发布、聊天室、音视频通话等功能的跨专业学习互助系统。

在进行需求的详细分析后，确定了以 Vue3、Gin 框架等为技术支撑的开发方案，在系统设计上，遵循易用性、可维护性和安全性原则，采用前后端分离和功能化分层架构。在数据库设计上，基于MySQL数据库，通过多个数据表及外键关联，确保数据的准确性和完整性。功能模块涵盖账户、文章、互助帮学、用户交流和后台管理，各模块分工明确，协同实现跨专业学习功能。

在系统实现阶段，前端方面，本系统充分落地了组件复用的思想，本系统封装了高复用性组件例如表单组件，过滤器组件、聊天框组件等，通过组件的服用，减少了前端开发的工作量，提高了开发的效率。在后端方面，本系统秉持模块化设计理念，将系统拆分为多个功能模块，如用户管理模块、文章管理模块、互助帮学模块、交流模块和后台管理模块等。每个模块具有清晰的职责和接口，便于开发、维护和扩展。对于一些常用的业务逻辑和工具函数，进行了封装和抽象，形成了可复用的代码库。例如，将用户认证、数据加密、日志记录等功能封装成独立的工具类，在不同模块中可以直接引用，避免了代码的重复编写，提高了代码的可维护性和可测试性。

最后，本系统根据前期的需求分析，列出了详细的测试用例，本依照测试用例对系统进行了全面的测试，以确保系统达到需求分析的目标，能够稳定、可靠的运行。

7.2 系统未来展望

目前系统仅仅实现了基础功能，对比市面上存在的系统，还存在有很多可以拓展优化的地方，具体如下。

在功能性方面，可以进一步丰富学习资源类型，提供更多学习能力，例如互动学习，AI总结问答，接入更强大的语言模型以此提高问答的准确性和深度，为学生一共更加智能的学习辅助能力。

在社交互动上，可以增加社交小组功能，根据学习方向进行功能分区，添加学习直播功能，实现线上的一对多教学功能，促进学生之间更深入的交流与合作，营造更活跃的学习氛围。

在性能方面，随着用户数量增加，对系统进行性能测试和优化，采用负载均衡、分布式存储等技术，确保系统稳定运行，提高并发处理能力。

在平台拓展性上，也以开发移动端应用，实现跨平台使用，方便学生随时随地进行跨专业学习，扩大系统的使用范围和影响力。

参 考 文 献

孙赤婴.人才培养需要发展跨学科教育[J].教育,2024,(17):1.

郭思嘉.企业内部在线学习平台的设计与实现[D].哈尔滨工业大学,2018.DOI:10.27061/d.cnki.ghgdu.2018.001050.

王铮清,刘壮峰.基于Go语言的内容管理系统的设计与实现[J].电脑知识与技术,2022,18(24):62-64.DOI:10.14004/j.cnki.ckt.2022.1433.

李仲岐.基于WebSocket即时通信系统设计与实现[C]//天津市电子学会.第三十七届中国（天津）2023’IT、网络、信息技术、电子、仪器仪表创新学术会议论文集.天津市电子仪表信息研究所有限公司;,2023:129-131.DOI:10.26914/c.cnkihy.2023.022867.

肖睿.基于Gin框架的营销活动公共类库的设计与应用[D].华中科技大学,2019.DOI:10.27157/d.cnki.ghzku.2019.005431.

胡敏.Web系统下提高MySQL数据库安全性的研究与实现[D].北京邮电大学,2015.

纪乾坤.面向化工领域动态表单多前端框架通用渲染机制的研究与应用[D].合肥学院,2023.DOI:10.27876/d.cnki.ghfxy.2023.000081.

郑荣茂.基于Web系统的测试[J].韶关学院学报(自然科学版),2005,(03):36-38+43.

唐超.WebRTC协议在线上教学中的应用实践[J].福建电脑,2024,40(07):41-47.DOI:10.16707/j.cnki.fjpc.2024.07.008.

Cao W ,Tan X ,Liu B , et al.Design and Implementation of a High Performance Event-Driven WebSocket Server[C]//Computing,Performance and Communication Systems(2016 Vol.1 Num.1).Computer Science Technology,Wuhan University of Technology;,2016:30-34.

Guo Y .Study on High-performance Distributed Cache Architecture based on Redis[C]//Information Engineering Research Institute,USA.Proceedings of 2012 2nd International Conference on Advanced Materials and Information Technology Processing(AMITP 2012) Volume 34.Department of Information Engineering,Beijing Information Technology College;,2012:25-29.

Quan Y .Design and Implementation of E-commerce Platform based on Vue.js and MySQL[C]//Wuhan Zhicheng Times Cultural Development Co., Ltd..Proceedings of the 3rd International Conference on Computer Engineering, Information Science & Application Technology(ICCIA 2019).The General Education College of Xi'an Eurasia University;,2019:460-465.DOI:10.26914/c.cnkihy.2019.055856.

Wei Z .The design of library database management system based on MySQL[C]//Department of Biomedical Engineering, Faculty of Engineering, ?ukurova University.Proceedings of the 2023 International Conference on Machine Learning and Automation.Electronic and Electrical Engineering,University College London;,2023:948-957.DOI:10.26914/c.cnkihy.2023.124752.

致 谢

行文至此，我的大学生活也即将画上句号。在撰写这篇毕业论文的过程中，我收获颇丰，也得到了许多人的帮助，心中满是感激，希望借这段文字表达我的谢意。

我要诚挚地感谢我的导师姚金涛副教授。从选题的迷茫，到系统架构设计的困惑，每一个关键节点，姚老师都给予了我悉心的指导。他耐心地解答我的问题，用丰富的学识和严谨的治学态度为我指引方向，让我在研究的道路上少走了许多弯路。在论文的撰写过程中，姚老师仔细审阅每一个章节，提出了宝贵的修改意见，让我的论文质量得到了质的提升。

我还要感谢数学与信息学院的各位授课老师。在大学四年的时光里，他们用精彩的课程为我打下了坚实的专业基础，让我在计算机科学与技术的领域中不断探索、成长。因为有了他们的教导，我才能拥有足够的知识和技能去完成这个项目。

在项目开发过程中，我得到了许多同学的帮助。我们一起讨论技术难题，分享彼此的想法。在遇到困难时，大家互相鼓励、共同攻克难关。那些一起熬夜调试代码、激烈讨论方案的日子，成为了我大学时光中最珍贵的回忆。他们的陪伴和支持，让我感受到了团队的力量，也让我在这个项目中不断进步。

此外，我要感谢我的家人。他们一直是我最坚强的后盾，给予我无条件的支持和鼓励。在我为学业忙碌的日子里，他们默默承担起生活的琐碎，让我能够全身心地投入到学习和研究中。无论遇到什么困难，他们总是相信我、支持我，让我有勇气不断前行。

感谢华南农业大学为我提供了良好的学习环境和丰富的学习资源，图书馆里丰富的藏书、先进的实验室设备，都为我的研究和学习提供了有力的保障。

在未来的日子里，我会带着这份感恩继续前行，将所学知识运用到实际工作中，努力为社会贡献自己的一份力量。再次向所有帮助过我的人表示衷心的感谢。