

齐鲁工业大学

本科毕业设计（论文）

基于 LLM 与多模态人工智能的健康管理与辅助诊疗系统的设计与
实现

学部（学院） 计算机科学与技术学部
专业班级 软件工程（软件开发）21-1
学生姓名 杜宇
学号 202103180009
导师姓名 姜文峰 李君

2025 年 06 月 03 日

本科毕业设计（论文）

基于 LLM 与多模态人工智能的健康管理与辅助诊疗系统的设计与

实现

学部（学院） 计算机科学与技术学部

专业班级 软件工程（软件开发）21-1

学生姓名 杜宇

学号 202103180009

导师姓名 姜文峰 李君

2025 年 06 月 03 日

齐鲁工业大学本科毕业设计（论文）

原创性声明

本人郑重声明：所呈交的毕业设计（论文），是本人在指导教师的指导下独立研究、撰写的成果。设计（论文）中引用他人的文献、数据、图件、资料，均已在设计（论文）中加以说明，除此之外，本设计（论文）不含任何其他个人或集体已经发表或撰写的成果作品。对本文研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示了谢意。本声明的法律结果由本人承担。

毕业设计（论文）作者签名： 杜宇

2025 年 06 月 03 日

齐鲁工业大学本科毕业设计（论文）

使用授权说明

本毕业设计（论文）作者完全了解学校有关保留、使用毕业设计（论文）的规定，即：学校有权保留、送交设计（论文）的复印件，允许设计（论文）被查阅和借阅，学校可以公布设计（论文）的全部或部分内容，可以采用影印、扫描等复制手段保存本设计（论文）。

指导教师签名： 晏文华 李君 毕业设计（论文）作者签名： 杜宇

2025 年 06 月 03 日

2025 年 06 月 03 日

目 录

摘 要	I
ABSTRACT	III
第 1 章 绪论	1
1.1 研究背景与目的	1
1.2 国内外研究现状	3
1.3 本文主要内容	5
1.4 本文组织及内容安排	6
第 2 章 关键技术研究	8
2.1 后端开发技术研究	8
2.1.1 Flask 框架	8
2.1.2 MySQL 数据库	8
2.1.3 RabbitMQ	9
2.2 人工智能技术研究	9
2.2.1 Qwen2.5-3B-Instruct 大模型及大模型微调技术	9
2.2.2 大模型输出增强策略	10
2.2.3 BioMedCLIP 多模态模型	11
2.2.4 基于文本增强的时间序列预测算法研究	11
2.3 前端开发技术研究	12
2.3.1 Vue3 框架	12
2.3.2 Element Plus 框架	13
2.3.3 前端其他技术综合研究	13
2.4 项目开发工具综述	14
第 3 章 系统需求分析	16

3.1 系统可行性分析	16
3.2 系统功能性需求分析	17
3.3 系统非功能性需求分析	19
第 4 章 系统设计	20
4.1 系统的构建	20
4.2 系统的框架及数据库建立	24
第 5 章 系统实现	34
5.1 总体设计论述	34
5.2 用户信息模块	35
5.3 多模态辅助诊断模块	42
5.4 医疗问答模块	45
5.5 医学与诊疗论坛平台模块	52
5.6 诊疗事项清单管理模块	56
5.7 病历诊断与资源中心	59
5.8 后台管理系统	62
第 6 章 系统测试	66
6.1 针对业务逻辑层 API 的单元测试	66
6.2 针对系统功能的黑盒测试	82
第 7 章 结论与展望	90
参考文献	92
致 谢	94

摘要

在新时代互联网应用与技术进一步普及与人工智能技术飞速发展的双重驱动下，计算机技术在医健领域的应用可谓愈加广泛，大众对医健日渐增加的需求难以被传统的诊疗与保健管理模式所满足，其面临的诸多如诊断效率低下、医资配置不均、患者不便利与决策依赖经验等问题已相对严重。因此，如何利用互联网与前沿的人工智能技术，特别是利用大规模语言模型（LLM）与多模态技术来提升医疗保健相关业务的数字化、智能化已成为一个重要的课题。为更好地探讨互联网技术以及大语言模型与多模态等 AI 技术在医疗领域的潜力，本研究设计并实现了明康慧医（Minh Khoe Tue Y）——基于 LLM 与多模态人工智能的健康管理与辅助诊疗系统，本人同时也为提高医患交流效率和优化诊疗的流程尽一份作为本科毕业生的微薄之力。

本平台是一个集注册登录、个人信息管理、多模态智能辅诊、医疗问答、诊疗论坛、病历管理、诊疗事项清单管理、资源中心及后台管理九大模块于一体的分布式系统平台。系统宏观架构采用前后端分离设计，业务逻辑层后端基于 Python Flask 框架，数据库采用 MySQL 的方案，RabbitMQ 实现完成业务逻辑端与智能服务端的异步消息通信，构建分布式微服务部署；前端页面组件化与交互效果采用 Vue3、axios 与 Element Plus 实现，系统鉴权通过 JWT 机制实现，保障数据的安全。

在 AI 智能服务端方面，“智能多模态辅诊”基于 BioMedCLIP 对比学习模型与 MarianMTModel 中英文神经机器翻译模型的级联架构，通过输入的医学影像，计算多条待判中文诊断描述为正确的相对概率分布。医疗问答、问题深度研究及其它语言生成任务均利用 MKTY-3B-Chat 大模型。该 LLM 以 Qwen2.5-3B-Instruct 为底座，采用 LLaMA-Factory 利用大量医学领域文本微调而成。问题深度研究模块基于“大模型讨论机制”，是为本人自研的一种 LLM 生成模式，可充分挖掘大模型内部的知识且可引导其推理。

“明康慧医”系统的具体设计与实现过程的描述在本文中得以完整呈现。这次研究首先明确了本系统开发的行业背景与选择上述技术路线的依据，然后在技术可行性角度，分层次解析了核心功能需求与实现方案，系统架构中各模块的工作原理与技术要点在文中以重点说明，全部性能指标在测试环节得到了覆盖测试。本人在文末总结了当前的成果，还作了后续改进方向的计划。这项医疗数字化的项目是为本人的一次探索，如能引

发学生们对 AI 医疗的关注，吸引更多同学参与该领域，这便是本研究最大的价值所在。

关键词：辅助诊疗；大规模语言模型；多模态；Vue3；Python Flask；

ABSTRACT

Driven by the further proliferation of Internet applications and technologies in the new era and the rapid advancement of artificial intelligence (AI), the application of computer technology in the healthcare domain has become increasingly widespread. The growing demand for public medical health can no longer be adequately met by traditional diagnosis, treatment, and health management models, which face numerous challenges such as low diagnostic efficiency, uneven distribution of medical resources, patient inconvenience, and experience-dependent decision-making. Therefore, leveraging Internet and cutting-edge AI technologies—particularly large language models (LLMs) and multimodal techniques—to enhance the digitization and intelligence of healthcare services has emerged as a critical research topic. To further explore the potential of Internet technologies, LLMs, and multimodal AI in the medical field, this study designs and implements Minh Khoa Tue Y (MKTY)—a health management and AI-assisted diagnosis system based on LLMs and multimodal artificial intelligence. Additionally, it aims to improve doctor-patient communication efficiency, optimize diagnostic workflows, and contribute to the digital transformation of China's healthcare industry as an undergraduate student's modest effort.

This platform is a distributed system integrating nine modules: user registration/login, personal information management, AI-assisted diagnosis, medical Q&A, diagnostic forum, medical record management, treatment checklist management, resource center, and backend administration. The system adopts a frontend-backend separation architecture: the backend, built on the Python Flask framework, handles business logic, while MySQL serves as the database. RabbitMQ facilitates asynchronous messaging between the business logic backend and the AI service backend, enabling distributed microservices deployment. The frontend employs Vue3, axios, and Element Plus for component-based page design and interactive effects, with JWT-based authentication ensuring data security.

On the AI service side, the "Intelligent Multimodal Diagnosis" module utilizes the BioMedCLIP contrastive learning model and the MarianMTModel neural machine translation model for Chinese-English translation. It calculates the relative probability distribution of multiple candidate Chinese diagnostic descriptions based on input medical images. The medical Q&A, in-depth problem analysis, and other text-generation tasks leverage the MKTY-3B-Chat LLM. This model is fine-tuned from Qwen2.5-3B-Instruct using LLaMA-Factory and extensive medical domain texts. The in-depth problem analysis module

employs a "LLM Discussion Mechanism"—a self-developed generation paradigm that thoroughly excavates the model's internal knowledge and guides its reasoning.

This paper investigates the design and implementation of the Minh Khoa Tue Y system. First, it elaborates on the research background and technology stack. Next, it systematically analyzes key technologies, requirements, and feasibility. Subsequently, it details the implementation principles and technical specifics of each module across all layers, followed by comprehensive testing and performance evaluation. Finally, the paper concludes with a summary and future prospects. This research represents a bold exploration of medical digitization, and the author hopes it will inspire more students to actively participate in "AI + Healthcare" studies.

Key words: AI-assisted diagnosis; Large language models (LLMs); Multimodal; Vue3; Python Flask;

第 1 章 绪论

1.1 研究背景与目的

伴随着计算机网络全球化的推进与人工智能技术的飞速发展，尤其是近些年在自然语言处理（NLP）等领域的突破，医健领域中人工智能的应用愈加广泛。与此同时，人口的老龄化、慢性病多发以及医疗需求的日益增长，促使传统医疗服务模式面临倍增的压力，基于 AI 的健康管理与辅助诊疗模式作为现代医疗健康服务高新技术的重要组成部分，已开始在全国乃至全球受到广泛关注。现有的医疗健康管理体系大多都依赖人工操作和传统的寻医问诊手段，存在信息不对称、诊断效率低、患者体验差等问题：偏远地区和基层医疗服务能力不足，导致患者难以获得及时有效的治疗；其次，患者在挂号、就诊和检查等环节中往往需要耗费大量时间，并且传统纸质的医学文书使得数据共享困难。根据世界卫生组织（WHO）的统计数据，全球老年人口占比到 2050 年预计可达 22% 之高，随着全球人口老龄化和慢性病发病率的上升，现代医疗系统承受的压力日益加剧，而医学在人工智能技术和互联网的加持下，能够在疾病诊断、医疗决策支持和健康管理等方面展现出显著优势，在一定程度上上述问题可以明显缓解。因此，利用现已广泛普及的互联网与先进的 AI 技术，特别是大语言模型与多模态 AI 来提升健康管理和辅助诊疗服务的智能化和自动化，正成为当前研究的热点，在此背景下，开发一个基于互联网技术和集成了人工智能技术的健康管理与辅助诊疗系统，不仅是技术创新的体现，也是应对当前医疗挑战的迫切需求。

本研究致力于设计实现一个基于大语言模型和多模态人工智能的健康管理与辅助诊疗系统，主要在临床辅助诊断、患者自诊、智慧健康管理、电子病历分析等方面进行研究。系统引入自然语言处理技术和多模态数据分析技术后，各智能模块能够分析病人的疾病信息、辅助提供保健建议并协助医生的临床决策，从而提升诊疗效率和健康管理效果。此外，系统中医疗论坛与个人主页等部分还可以使医患间和患者之间的交流更便利。

通过前文的分析可知本项研究的意义非凡。首先，项目紧跟时代脉搏与技术前沿，搭上了医疗智能化发展的“快车”。LLM 是一种非常厉害自然语言处理工具，经过本人针对临床医学与医药特定领域文本微调后的 LLM 能够理解也能生成生成专业的自然语言，在语言的严谨性、内容的正确性等方面一点也不比人类医师差，为我们这个时代的医疗领域打开了一扇新的大门。而多模态技术对齐了医学图像、文本信息特征，可“另

辟蹊径”地在另外一个角度辅助医师诊断疾病。本次研习将这两项尖端新技术应用到健康管理与辅诊系统中，不仅能够提高系统对健康数据的处理能力，还能实现更为精准的诊断和个性化的健康管理。除此之外，互联网还是一个提供医疗知识传播和交流功能的平台，使用者可以在在线论坛和资源库上获取最新医学与疾病相关的信息，医者可以借此学习新知、探讨交流，患友们也可以借此了解到最新的疾病信息。

第二，本项研究涉及提升医疗过程的效率与质量。就像前文所述，现在好多医院，特别是欠发达城市和地区，医生少病人多，看病得等好久，医生工作压力也非常大。为了解决这一问题，本研究提出的基于 LLM 和多模态人工智能的系统能辅助医生做疾病筛查、分析症状，还能给诊断建议，这样医生压力小了，看病速度也快了。而且患者可以用它自诊，也就是自己用这个系统在家治疗“小病”，根据自己的健康情况，平时随时就能查，有些小毛病自己在家就能解决，不用频繁往医院跑，某些情况下还可能提前发现大病，防患于未然，把疾病扼杀在摇篮里。

第三，本研究致力于推动个性化的健康管理。以往的健康管理，就是拿普遍的医学知识和经验来指导，根本没考虑到每个人的情况都不一样。本研究项目不一样之处是，系统是基于 LLM 和多模态人工智能的，它能用 LLM 和 CLIP 这些模型还有其它 AI 技术，把患者自己输入的数据深度挖掘与分析，然后量身定制，得出一套专属于某个病患的健康管理方案。

第四，本项研究可以促进临床决策辅助系统的智能化。老百姓俗语常说“一个大夫给你一个说法，不如多个大夫给你多个说法”。在医院看病的时候，医生临床做决策总是只能自己靠经验，可参考的东西太少了，资源有时候也很有限。本项研究中的 LLM 方法和多模态算法则可以系统地辅助医生，它已经“阅读”完了海量的医学文献，能把病例还有患者信息都分析一遍，在它的帮助下医生能把主观因素的影响排除到最低，帮医生判断诊断得是否正确，还能给出诊疗方案，让医生做的决策更靠谱、更科学。

第五，本研究立足于当下，但更顾及未来。本次研究谨以此文浅析了 AI 与医学深度结合越来越大的应用前景：本项目的研究目前来看非常具有实际应用的意义，而且还能给以后 AI 与医学融合领域的技术发展提供一点微薄之力。未来人工智能新技术、新算法会不断涌现，在这波人工智能大浪潮之下，其与医疗领域交叉的应用只能是越来越广。本项目的设计与实现，可以为后续研究提供经验和技术创新，本人真切希望能有更多人加入到 AI+医疗的研究当中，合力把医疗技术变得更好，让大家都能受益！

1.2 国内外研究现状

众所周知，近几年科技发展得迅猛，可以说人工智能和互联网技术的广泛应用已经把医健领域的老模式彻底颠覆了。在互联网、人工智能与大数据、多模态学习等技术的共同推动下，医疗健康管理和辅助诊疗系统得以飞快更新，现在不管是看病诊断的速度，还是日常健康管理，或者是医患间的沟通、信息的分享，都比以前强太多太多。尤其在疫情结束之后一段时间里，数字医疗的理念逐渐深入人心，像 AI 在线看病、用智能工具自诊、靠智能系统辅诊这些服务，都不觉得新鲜了，好多人都愿意接受，这便为“智慧医疗”相关系统的开发与落实开辟了一条大马路。国内外很多的研究机构、企业、高校和技术开发者纷纷入局，“一股脑地”投入到这一领域中，大量创新性的技术、系统和模型随之涌现。^[1]

而且不得不说的是，基于大语言模型训练和微调的技术在健康管理与辅助诊疗上的应用已经有了不少的突破，如 Google 的 BERT^[3]、Meta 的 LLaMA^[4]、清华大学与智谱公司的 GLM 系列模型^[5]等已在多个方面显露出了实力，比如说疾病预诊、智能问答和医学文献分析等方面。各种体量、各种架构的医学大模型如雨后春笋般涌现^[8]，国外 Google 公司的 Med-PaLM 大模型、Med-Gemini 多模态医学大模型^[7]和哈佛医学院团队的泰坦（TITAN）模型就比较有代表性，下文中将详细讲一下它们。

（1）Med-PaLM 大模型与 Med-Gemini 多模态大模型

2023 年，Google 公司发布了 Med-PaLM 大模型，经过在医学文本上的训练，这模型的水平都能和主治医生差不多“平起平坐”了。到了 2024 年中旬，Google 公司又“放大招”，推出了更厉害的 Med-Gemini 多模态医学大模型，这个 Med-Gemini 模型在 MedQA 和 USMLE QA 等好几个医疗测试里表现都特别好，甚至远远超过了博士的水平。Med-Gemini 功能强大，它拥有临床多模态理解和推理，还有长文本处理的能力，在皮肤科、放射科、电子病历解析等这些地方中都用得很广泛。Med-Gemini 可以进行医疗方面的多模态对话，什么事拿不准直接发图，使用起来非常方便。^[6]

（2）多模态病理模型 TITAN

在医学对比学习模型领域里，哈佛医学院在 2024 年 11 月搞了个大动作，他们发布了一款准确度很高的多模态病理模型 TITAN。它是一种多模态全切片基础模型，研究人员以 33 万多张全切片图像和对应的病理报告，用视觉语言对齐的算法给它做的预训练。不用微调，也不需要额外标签，TITAN 就能提取通用的切片表示，还能生成有泛化能力的病理报告，那怕临床资源是有限的。

我国国内的医疗大模型的研究更是一片欣欣向荣。我国中医，自成体系，为了发扬中医文化，“扁鹊”、“仲景”、“孙思邈”、“华佗”和“神农”等中医大模型层出不穷，被各大高校和企业研制出来。例如“学习强国”中的“中医智能健康助手”就是基于南京大经中医药信息研发的“岐黄问道大模型”，已受益千万家。在通用医学领域，商汤科技的“大医”大模型以“日日新·商量”为基础底座模型微调得到，“大医”的医学能力已超越 GPT-4 的水准。科大讯飞的“星火医疗”大模型在医疗诊断治疗推荐、医疗语言理解、专业医学文书生成、医学知识问答、医疗多模态等方面超越了 GPT-4o 的水平。上海人工智能实验室（OpenMEDLab）发布的“浦医”大模型基于海量医学数据以及医学专家知识微调，覆盖十余个临床方向，在多个医学具体领域达到了 SOTA 水平。^[14]在此着重介绍我校研发的一款中医领域大模型“扁仓”。

（3）扁仓中医大模型

“扁仓中医大模型”是我校齐鲁工业大学（山东省科学院）自然语言处理与认知计算团队联合山东省中医药大学附属医院临床研究中心一起研发的一款中医领域大模型。该模型以 Qwen-7B-Chat 为底座，利用中医药指令数据集和由心内科病历构建的中医辅助诊断数据集全参微调得到。模型在 TCM Syndrome Differentiation、TCM Disease Diagnosis、TCM Exam 等数据集上评测的得分之高已超过了 GPT-4，实属不易。

另外，互联网技术在医疗领域的应用得到了世界各国政府、企业和学术界的高度重视，许多知名平台和系统已广泛服务于公众。这些平台与系统不仅提供了医疗论坛、医患交流、诊断结果共享等基础功能，还为各大前沿 AI 模型的落地使用提供了一个载体。比较典型的例子有我国的“春雨医生”、“好大夫在线”以及国外的“沃森医疗”（IBM Watson Health）。

（4）春雨医生

春雨医生能代表中国线上医疗业的口碑，它提供的健康咨询、在线问诊和医生预约这些功能一应俱全。用户只需打开软件，就可以和专业医生进行网上当面交流，获得医生的疾病诊断和健康指导。便捷的服务模式和雄厚的医生资源是春雨医生成功的关键，它的出现，也给国内其他互联网医疗平台的发展提供了不少重要的参考。

（5）好大夫在线

互联网医疗平台中，好大夫在线也是其中的佼佼者之一，它专注在线问诊，也为医患搭建起了一座“桥梁”。患者能在上面预约专家看病、咨询健康疑惑，拉近了医患的距离。好大夫在线借助互联网技术实现了医患之间的高效联通，以前传统医疗资源分布

不均的大难题，也因它得到了些许缓解。

（6）沃桑医疗（IBM Watson Health）

美国 IBM 的沃桑医疗在 AI 医疗应用上号称“标杆”，它运用基于人工智能的医疗解决方案，比如用自然语言处理和机器学习技术，对医学文献、电子病历等数据作分析。不论是帮助医生做临床决策支持，还是在疾病诊断、药物研发和个性化治疗方面，沃桑都表现得亮眼，连美国总统都说它是 AI 在医疗领域里应用的“典型”。

1.3 本文主要内容

本篇论文的主要研究内容是“明康慧医”（Minh Khoa Tue Y，简称：MKTY）——基于 LLM 与多模态人工智能的健康管理与辅助诊疗系统设计与实现。现在的医疗场景越来越需要便捷、个性化的诊疗支持了，利用大语言模型和多模态人工智能技术给用户提供智能的辅诊和健康管理，正就是这个系统的创新之处。本文中，将详细讨论下列内容：

内容方面：系统共包含九大模块，注册登录、个人信息管理负责用户事务，智能辅诊、医疗问答动用了大模型，诊疗论坛、资源中心和诊疗事项清单管理方便了用户的交流，还有后管端是管理员登录管理系统的，每个模块都将挨个介绍其设计理念、数据模型建立、和代码里的那些必要的细节等。

技术选用方面文本将探讨以下内容：处理大批量前端 Web 请求涉及数据库读写的频繁执行，属于 I/O 密集型操作，而 AI 模型推理需要非常大的显存及算力资源，属于计算密集型操作，由于这两部分的逻辑性质不同，依赖的硬件资源要求也不相同，因而有必要设计系统采用前后端分离的架构和分布式架构，使系统的 SSR 前端、Web 请求处理逻辑、各 AI 模型推理逻辑、消息队列 RabbitMQ 等各组成部分分别置于不同的机器上运行。系统的所有 AI 模型均分别作为微服务模块，部署于带有 NVIDIA 计算卡的算力服务器；系统的 Web 请求处理后端采用 Python Flask 开发框架，并以 MySQL 8.x 作为数据库。分布式系统的“枢纽”采用 Rabbit MQ 消息队列，通过它来实现对 AI 模型调用请求消息的发布与消费、异步任务的处理以及高并发场景下 AI 模型调用请求洪峰的削峰，Python 后端使用 pika 库与 MQ 进行交互，使用 mysql 库与数据库进行交互；系统前端主体分别使用了 Vue.js 3 框架、axios 与 Element Plus UI 库实现页面组件化、交互效果与 UI 界面的美化，开发时基于 Vite 打包工具。另外，前端采用 highlight.js、marked.js 与 DOMPurify 库提供安全的 Markdown 代码渲染支持，并使用 ECharts 库绘制美观图表。在人工智能层面，本项研究基于 Qwen2.5-3B-Instruct 大模型，利用约 2.88GB 医学领域

语料数据微调得到了 MKTY-3B-Chat 大模型，并将其集成于系统中，同时利用本人自研的“大模型讨论机制”以及对知识库利用 RAG 方法均可以增强大模型的输出。系统引入了 MarianMTModel 中英文神经机器翻译模型与 BioMedCLIP 多模态模型以支持医学影像与文本的联合分析，以图像为依据计算多条语句正确的概率分布。此外，本人在本项研究中自研了一个文本增强的医学时间序列预测模型，该模型主要依靠门控循环单元（GRU）进行基础时间序列预测，而后计算历史时间序列的频域特征与文本描述的句子嵌入的交叉注意力分数向量，并利用其计算出一个差值波形，用于调整 GRU 输出的结果。系统还包含一个后台管理系统（后端），这一部分也是基于 Vue3 + Element Plus 与 Python Flask 框架，但逻辑上独立于其他部分，仅全权负责对数据库的管理。

系统测试、分析与总结方面：将对系统进行全面的单元测试与总体测试，分析测试结果，并作项目竣工总结、展望未来。

1.4 本文组织及内容安排

本篇论文共分为七个章节，各章节内容安排如下：

第一章为绪论。本章开门见山阐述了，本研究课题的提出源自于现今健康管理和辅诊领域在人工智能影响下发展的趋势和面临的许多挑战，这也便是研究背景和意义。紧接着，综述了对国内外相关的研究现状，包括 LLM 和多模态 AI 在医疗领域的应用进展。经此梳理，本研究的重要性、创新点及应用价值得到了明确。最后简要介绍了论文的整体结构与主要内容，以给后续章节建起一个框架。

第二章为关键技术研究。本章节详细介绍了开发系统中所用的各项主要技术，包括一些基本方法，诸如 Python Flask 后端框架、MySQL 数据库、Rabbit MQ 消息队列等，还有 Qwen2.5-3B-Instruct 大模型及基于 LoRA 算法和 LLaMA-Factory 软件的大模型微调技术，对比学习方面还有涉及 BioMedCLIP 多模态模型。有一些本人自研的方法也有浅析，如大模型输出增强策略和基于文本的时序预测算法。Vue3 前端、Element Plus 和前端其他技术与项目开发工具等都做了描述，对每项技术不仅深入探讨了它的原理、功能，还分析了在系统中的具体应用场景，从而筑牢了系统设计与实现的理论之基。

第三章为系统需求分析，“明康慧医”系统的全面需求分析在这一章完成。可行性分析方面有技术可行性、经济可行性和操作可行性，功能性需求分析依次讲解了系统九大模块的具体功能，还有非功能需求分析包括性能、安全性和用户体验。通过需求分析，我再次明确了系统的目标受众，定下来了它的核心功能，指导我后续开发这个项目。

第四章为系统设计。这一章提供了系统架构图和功能模块图，明确了系统的整体设

计。本章对各模块的功能、交互及技术实现草案还进行了详细的描述。同时，对于解决高并发和分布式部署的需求，这一章还对 Rabbit MQ 的分布式架构和前后端分离的具体设计策略做了研究。

第五章为系统实现。本章节基于第四章的设计，具体到代码，描述了怎么实现系统的各个功能模块，在实现过程中的各技术细节与难点解决方案都在这一节中详细介绍了。

第六章为系统测试。这一章分为两个部分，对 API 的测试和面向功能的测试。对功能的测试结合了实际应用场景，对系统的核心功能设计了一些用例测试，主要是黑盒测试和集成测试，试验模块间的协作如何。对 API 的测试就是单元测试，查看各个模块独立的功能是不是有问题。通过测试结果都达到了预期，验证了系统功能上的完备性和性能的可靠性。

第七章为总结与展望。本章节总结全文与本次的研究工作，归纳系统的技术成果与创新点，概括本研究所做的贡献。当然，技术与科学的研究是讲求实事求是的，我本人对自己尚未解决的问题和系统在研究过程中的不足之处也坦诚地写了出来，并提出了未来相应的改进方向，为后续的研究提供了参考。

第 2 章 关键技术研究

2.1 后端开发技术研究

2.1.1 Flask 框架

Flask 是一个轻量级的 Python Web 应用框架，广泛应用于快速开发构建轻量而灵活的 Web 应用程序和 API 服务，其基于 Werkzeug WSGI 工具包和 Jinja2 模板引擎，采用“微框架”的设计理念，虽然功能简洁，但具有高度的灵活性和可扩展性。它遵循“约定优于配置”的原则，为开发者提供了构建 Web 应用所需的“最小化”工具集，同时允许开发者根据项目需求自由扩展，具有简单易用、高度可定制等特点。

Flask 本身仅包含路由、HTTP 请求处理等核心功能，ORM、缓存、身份验证等其他功能可通过官方或第三方扩展（Flask-SQLAlchemy、Flask-Login 等）按需集成，这种设计不会使框架变得“臃肿”，能使项目结构更清晰，便于维护和扩展。另外 Flask 助于开发者高效开发，内置开发服务器和调试器，支持快速本地开发和实时调试；Flask 的 RESTful 请求处理功能可以很轻松地处理 HTTP 请求和响应，适合构建 API 服务；Flask 还包括一种上下文机制，通过请求上下文对象（`request` 对象、`session` 对象等）可以简化状态管理。Flask 框架的请求处理做的非常简洁但功能强大，支持 GET、POST、PUT 等 HTTP 请求的各种方法，其支持文件上传和表单处理等。

在本项目中，业务逻辑层 Web 请求处理端的 API 服务采用 Flask 框架进行构建，均使用 RESTful 架构，并以此实现了前后端分离的开发模式。

2.1.2 MySQL 数据库

MySQL 是一款功能强大且具备高安全性、高可靠性与高效性的开源关系型数据库管理系统，它广受各技术栈的开发者欢迎，已成为全球开发者及各类企业的首要选择。它的规模小巧轻量、运行效率高，并且是开源的，还具有优化了的查询执行引擎，支持索引、分区等加速技术，在处理大规模数据时表现很出色。它的核心优势为跨平台兼容性优良，开发者可在 Windows、Linux、macOS 等操作系统间无缝迁移数据库，使用 Python、Java、C++ 等常见高级编程语言可实现快速开发。开发者在某平台上编写的程序，几乎无需进行任何的代码修改，就能直接封装迁移至其他平台继续运行，这有效保障了其技术生态的广泛适应性。

MySQL 还是一种高性价比的数据库解决方案，MySQL 采用 GPL 开源许可证，对于企业而言，MySQL 开源的特性极大地削减了开发过程中的成本支出，也正因如此，

MySQL 在数量众多的中小企业里备受青睐。MySQL 8.x 版本于 2018 年推出，这一版本的意义重大，引入了诸多全新功能，比如添加了对 JSON 的支持，并对原有功能的效率进行了优化升级，让开发者在项目开发时更加便捷高效。

2.1.3 RabbitMQ

RabbitMQ 是一个在高级消息队列协议（AMQP）的基础上构建的开源消息代理软件，有着高性能、高可用性和扩展性的特点。它广受架构师们的青睐，这与其灵活的消息路由机制和稳定的消息传递能力分不开。它是用 Erlang 语言开发的，所以它的架构轻量，吞吐性能出色，还支持高并发，并且它的智能消息路由机制非常好，能支持直连和扇出等多种交换器类型，处理异步任务和解耦系统组件都表现的很好。

跨平台兼容性这方面，RabbitMQ 做得不错，Windows、Linux、macOS 这些主流的操作系统系统都能跑，Python、Java、Go、.NET 等十多种编程语言也都提供了访问 RabbitMQ 的客户端库。开发者写好的消息处理逻辑，核心代码几乎不怎么修改，就能直接迁移至不同技术栈的环境中使用，这极大地增强了系统集成的灵活性。

RabbitMQ 的高可靠性是有一套机制支撑的，消息持久化、集群化的部署方案和 ACK 确认机制，都是为了保证消息不丢。比如说 RabbitMQ 采用的镜像队列机制，会把消息自动同步到集群中的多个节点，精细地管控消息的生命周期，可以使用它内置的 TTL 和死信队列的特性。此外，它不仅完整实现了 AMQP 0-9-1 协议标准，还可以支持 STOMP、MQTT 等扩展协议，不同场景下的消息传输需求它都能满足。

2.2 人工智能技术研究

2.2.1 Qwen2.5-3B-Instruct 大模型及大模型微调技术

Qwen2.5-3B-Instruct 是阿里巴巴推出的一款小规模的对话模型，属于因果语言模型，体量大约为 30.9 亿参数。该模型的架构采用了带有 RoPE，SwiGLU 和 RMSNorm 的 Transformer。截至 2025 年 3 月，它是 Qwen2.5 系列的一款最新的模型，相较于前若干代模型，它的指令遵循能力有显著提升，在生成长度超过 8K token 的长文本、理解诸如表格的结构化数据以及生成结构化输出方面表现更佳。Qwen2.5 模型家族中包含各种尺寸的模型，通过充分的调研与对比，结合本次毕业设计项目研究所能获取到的硬件与算力资源支持，本人最终选定其 3B 模型进行微调。虽然比不上百亿参数级的巨大模型那样覆盖面广、上下文记忆强，但它体积小、响应快，部署起来也更灵活，本次研究受限于硬件资源而选用它进行下游任务。

项目开发中，本人采用 LLaMA-Factory 框架实现微调的工作。LLaMA-Factory 是一个简易但高效的大语言模型训练与微调框架平台。通过 LLaMA-Factory，开发者可以在无需编写任何代码的前提下，很容易地在算力服务器上甚至在本地完成部署，进而完成数百种预训练模型的微调。此外 LLaMA-Factory 框架支持针对各种算法、各种量化精度、各种模型架构的模型微调。

受限于计算卡显存资源，经综合考虑，本研究采用 LoRA 算法微调模型。LoRA 算法全称低秩矩阵适应算法，是一种用来微调大语言模型的巧妙方法。它的理念并不复杂：不用去动模型里所有的参数，而是通过加一些小的低秩的矩阵来调整模型，这样既能保持微调效果，又可省下不少计算资源和时间。算法原理大概为，将模型原权重矩阵加上一个小的变化量，这个变化量由两个低秩矩阵相乘得到。这样对于每个大矩阵，只需训练两个小低秩阵即可，可训练参数量可谓是“九牛一毛”。经过计算，假设用 LoRA 微调 GPT-3，参训参数量能缩减至万分之一，且性能几乎不折不扣。

本研究基于大量医疗领域语料，使用上述方法，微调得到了 MKTY-3B-Chat 明康慧医大模型，该模型尺寸与其底座 Qwen2.5-3B-Instruct 完全相同，在生物医疗领域的水平超过了 Qwen2.5-3B-Instruct，并针对本系统的特定模块做了优化。

2.2.2 大模型输出增强策略

本项目包含“明康慧医智慧问答”和“明康慧医智能体深度分析”两个显式调用 MKTY-3B-Chat 大模型能力的模块，医学领域的文本生成需求不同于普通的诗歌撰写、广告文案生成等，系统生成的医疗相关的文本应当是力求准确严谨的，而大模型生成文本过程中的“幻觉”现象截至目前以人类的手段又无法避免，那么本研究将使用两种方法解决这一问题。

检索增强生成（RAG）是目前人们公认的可以解决大模型幻觉问题的最好方法，同时也是当下 LLM 领域研究的一大热点，各种研究 RAG 方法的论文接连出现，但不论是哪种方法，基本原理万变不离其宗，不外乎利用某种方法将知识库中的某些语料片段取出，与用户 prompt 一起共同输入至 LLM，由 LLM 综合分析并组织语言生成。本项目中采用了一种比较基本的方法实现 RAG：由用户选定知识库，系统对知识库切片处理并计算每片的 TF-IDF 特征向量。用户输入文本时，计算 prompt 的特征向量，并计算各片语段的特征与其的欧几里得距离，将 top-k 的所有特征对应的语段与 prompt 拼接，输入至 LLM 进行解答。

本项研究中，本人设计了一种在一定程度上能够发掘大模型知识、降低知识性错误

率且能引导其做推理的 LLM 文本生成方案——大模型讨论机制。该机制的工作流程如下：模型权重相同（均为 MKTY-3B-Chat）但会话上下文不同，则不认为是同一个智能体，因而系统通过设置多个上下文数组模拟多个智能体，让每个智能体分别回答待深入研究的问题，然后由没有会话上文的“主持人”智能体总结各方发言，这是第一轮讨论的过程。以后每轮讨论，都将上轮主持人的总结和原问题拼接合并，并由各智能体基于自己的会话上下文再分别回答合并后的 prompt，最后主持人总结，周而复始，直至达到最大讨论轮次数。最后是“判敛”：用“BigBird”BERT^[12]将最后一轮讨论各方的输出计算句子嵌入向量，然后计算各向量两两之差的平均值，以此反应各方达成共识的程度，即讨论语义收敛程度，这个数值可供人类用户作参考。经测试发现，该方法的确可以激发 LLM 倾向于输出正确的内容并尽可能充分地利用其在各训练阶段学习到的知识。

2.2.3 BioMedCLIP 多模态模型

微软研究院在 2024 年发布的 BiomedCLIP，是一款性能比较好的生物医学视觉语言基础模型，它靠对比学习的算法，在 PMC-15M 数据集上进行预训练而成。PMC-15M 数据集是从 PubMed Central 的生物医学研究文章中提取的 1500 万对图文对。BiomedCLIP 使用 PubMedBERT 作为文本编码器，视觉 Transformer（ViT）作为图像编码器，并专门对特定领域做了调整。它能做各种类型的视觉语言处理（VLP）任务，如跨模态检索、图像分类和视觉问答都不在话下。BiomedCLIP 在一系列标准评测集上取得了新的最好的评测结果，而且显著比之前的视觉语言处理方法都优秀，但有个唯一的局限，由于该模型的预训练语料均来自 PubMed Central，因而其只支持英文。模型开源地址：

https://huggingface.co/microsoft/BiomedCLIP-PubMedBERT_256-vit_base_patch16_224

4

2.2.4 基于文本增强的时间序列预测算法研究

本项目本人在此次研究中尝试设计了一种医学时间序列预测模型，该模型基于深度学习算法预测时序数据，并能结合文本进行预测结果调整增强。具体来说，模型主要依靠门控循环单元（GRU）进行基础时间序列预测，而后通过 FFT 计算历史时间序列的频域，将频域中各频率序数对应的振幅向量与相位向量拼接得到频域特征，随后用 BigBird BERT 提取医学文本描述的句子嵌入，利用交叉注意力机制计算出频域联合特征向量与该句子嵌入的分数矩阵，从而得出加权频域联合特征。将此特征向量拆解并求逆 FFT 可得到一个差值时序数据，该数据可用于调整 GRU 输出的结果，作为模型最终的

输出。

下面简单介绍一下 GRU、交叉注意力以及 FFT 的基本知识：GRU 是一种擅长处理时序数据的循环神经网络，更新门和重置门等门控机制的引入控制了信息的传递，减轻了传统的 RNN 处理长序列梯度消失问题。更新门决定保留多少历史信息，重置门决定当前输入和历史状态融合的程度。具体关于 RNN 的内容本文从略。交叉注意力是一种可关联两种模态特征的注意力机制，它先是关注一种模态的每个特征维度与另一模态的每个特征维度，得到一个注意力分数矩阵，并利用该矩阵为第一种模态的特征向量加权。多模态任务中经常会用到交叉注意力和它的变体。FFT 全称快速傅里叶变换，用于从时域信号提取频域信息。

2.3 前端开发技术研究

2.3.1 Vue3 框架

本 Vue3 框架，全称 Vue.js 版本 3，是由我国独立开源开发者尤雨溪创建并由其团队成员维护的开源 MVVM 前端 JavaScript 框架。它基于标准 HTML、CSS 和 JavaScript 构建，采用“模型-视图-视图”模型（MVVM）模式，主要关注视图层。它提供组件化、声明式的编程模型，能够更科学地组织、简化 Web 前端构建。

Vue 是一款灵活的渐进式框架，开发者可以根据项目需求，按需选择不同的使用方式。它既可以在不需要构建步骤的情况下，用于对静态 HTML 进行渐进式增强，也可以作为 Components 嵌入现有页面，此外，Vue 还适用于开发单页应用（SPA）、服务端渲染（SSR）或全栈应用前端模块，甚至在开发 PC 桌面端和移动端应用软件、WebGL 项目乃至 CLI 命令行界面时，Vue 也有用武之地。

Vue 的核心特性中，声明式渲染基于标准 HTML 拓展模板语法，能描述 HTML 与 JavaScript 状态间的关系；响应性则能让 Vue 自动跟踪 JavaScript 状态变化，并实时更新 DOM；第三，Vue 还支持组件化开发，可以把页面拆分为独立可复用的一些组件，便于组合嵌套且具有复用性，符合软件工程学原理。Vue 通过虚拟 DOM 的思想将界面抽象为 JavaScript 对象树，经 Diff 算法比对后最小化以更新真实 DOM。此外，Vue 提供了丰富的生命周期钩子函数，让开发者容易通过对组件进行初始化、监听与销毁等各种操作来控制组件的生命周期。

Vue 简洁易用的语法、虚拟 DOM 的设计、组件化开发、生命周期钩子函数、数据驱动视图等特性，使其能够轻松与其他库或现有前端项目整合，基于 Vue 开发者可以高

效构建现代化 Web 应用，明显提升项目开发效率。

2.3.2 Element Plus 框架

Element Plus 是一款基于 Vue3，面向 UI 设计师和开发者设计的 UI 组件库，它由我国“饿了么”公司的前端团队开发，美观且好用，被很多开发人员称为“Vue 前端开发的伴侣”。它是一套给 Vue 用的“乐高积木”，目前市面上大量带表格、弹窗的网页，还有企业后台和数据看板等带表单的网页都是用这款框架快速搭建出来的。Element Plus 对开发者来说有三大好处：首先是省时间，按钮、表格、下拉框等组件它全写好了，并做了封装，无需开发者自己从头设计编写。其次是“颜值在线”，Element Plus 默认风格清爽大气，广受前端设计师和 UI 设计师的欢迎，第三，Element Plus 是由中国人开发的，它的文档也是中文的，很容易在国内找到社区讨论，这对于中国开发者非常有利。

2.3.3 前端其他技术综合研究

开发本系统前端的过程中，除了使用 Vue 和 Element Plus 之外，还引入了 Axios、highlight.js、marked.js、DOMPurify 和 ECharts 等重要的技术框架。这些库各负其责，共同支撑起了系统前端的运行。

首先介绍 Axios，它是一个基于 Promise 的 HTTP 客户端，主要用于发起网络请求。它实际上就是对 XMLHttpRequest 的封装，相当于“语法糖”，用起来比原生 JS API 更简单。但是它功能丰富，支持请求和响应拦截器、自动转换 JSON 数据、还可以控制超时，非常适合用在前后端交互通信模块的开发中。

在代码展示方面，本系统选择了 highlight.js。Highlight.js 是一个语法高亮库，可以起到自动识别和渲染代码片段高亮的效果。它支持一百种编程语言之多，还能根据代码内容自动判断语言类型，该库最大的特点就是能让代码显示的时候看起来更清晰、更专业。本项目中我特别选用了它的 rainbow 主题，用于渲染后台管理系统中的 markdown 及 JSON 代码渲染，管理员看着这些标记彩色的信息能舒服一些。

系统使用 marked.js 进行文档处理。marked.js 是一个 Markdown 语言解析器，能把 Markdown 格式的文本快速渲染成 HTML。它的优势明显，性能强，解析速度快，支持扩展语法，而且可以配合自定义的渲染规则来使用。在本系统中主要用于渲染大模型对话结果、医患互联电子邮件和电子病历。它的转换效果很好，且支持实时预览，适合本系统的技术需求。

DOMPurify 是一个用来“净化”HTML 的库，主要目的是防止 XSS 攻击。它能清

除<script>标签或者带有事件处理器的属性以及其他 HTML 中潜在的恶意脚本。为了系统安全，本系统引入了 DOMPurify，它与 marked.js 配合使用，在展示由 Markdown 转换出来的 HTML 时确保安全，以免 Vue 挂载该 HTML 时受到攻击。数字化的医疗系统中的数据安全特别重要，所以该库是不可或缺的。

ECharts 库是百度公司开源的一款数据可视化库，用来绘制各种交互式图表，比如折线图、柱状图和饼图等。ECharts 基于 Canvas，性能很好，支持很好的交互效果和响应式的布局。本系统采用 ECharts 库来做数据可视化，主要用它来绘制与展示智能多模态辅诊中概率分布的饼状图。使用它可以在视觉上让数据展示变得更为直观。

上述技术框架的搭配使用，使系统既保证了功能的完整性，又兼顾了美观度、性能和安全性。

2.4 项目开发工具综述

本项研究项目开发使用到的 IDE 有 PyCharm 与 VSCode，使用的代码管理工具和代码托管平台为 Git 以及 GitHub。下面将详细介绍之。

（1）PyCharm IDE

PyCharm 是一款由 JetBrains 公司开发的 IDE，专门为 Python 开发者设计，支持纯 Python 项目、Flask 或 Django 项目、还方便做科学计算和数据分析方面的研究。它之所以能够成为了 Python 开发者的首选工具，不仅因其强大的功能，还有它友好的用户 UI 界面。

（2）Visual Studio Code

微软公司开发的代码编辑器 VSCode 非常厉害，它全称 Visual Studio Code，能解决多种编程语言和环境下的开发问题。它有着轻量级的设计，功能非常丰富，还有极强的扩展性，是计算机各领域开发者的“瑞士军刀”。VSCode 的一大亮点是其集成了 GitHub Copilot 的功能。它是一个基于 LLM 的代码助手，可以根据当前打开的文件内容及项目中其他的文件生成代码建议，不用人插手，这个功能可以说能明显地提升编码速度，对学生也是很友好。

（3）Git 与 GitHub

Git 是一个使用非常广泛的分布式版本控制系统，发明初衷是帮助开发者更方便地进行代码管理和协作，其在 2005 年由来自荷兰的 Linux 内核创始人林纳斯·托瓦茨(Linus Torvalds) 开始开发。Git 提供了一种非常有用的功能，分支是一种可移动指针指向提交对象的，每位开发者可创建独属自己的工作线程，以便并行开发新功能、排错或实

验时不会影响项目主代码。

GitHub 是一个以 Git 命名的代码托管服务平台，它基于 Git 版本控制软件，是属于 Git 庞大生态系统的一部分。GitHub 是全球开发者交流和协作的重要平台，它不仅是开发者托管代码的地方，更是一个思想碰撞、技术交流的社区，大量开源项目在 GitHub 上生根发芽。本系统的开发借助了 Git 工具，并在 GitHub 平台托管代码，本系统开发的详细代码提交记录均由 GitHub 平台保留。截至撰写此文之时，本系统的代码仓库尚未公开，待本次研究结束并答辩通过后，本人将在征得各位指导教师的同意后开放源代码，开源地址：<https://github.com/duyu09/MKTY-System>

第 3 章 系统需求分析

3.1 系统可行性分析

在本系统开发初期，系统开发的工作能否顺利推进直接取决于是否开展了符合软件工程学的可行性分析。可行性分析的作用是，通过对系统的技术实现能力和资金成本投入进行系统性评估，论证系统开发与使用的合理性和必要性。各式各样的问题总会在系统开发及使用的过程中出现，有些是技术上的问题，还有的是资金的问题，故本节将结合明康慧医系统的特点，从技术可行性与经济可行性两方面展开分析。

（1）技术可行性

总体来看，本系统基于当下互联网和人工智能技术发展前沿，充分融合了大语言模型、多模态人工智能、基于 MQ 的分布式系统架构以及 Python Flask 后端框架与 Vue3 前端框架等多项成熟方案和技术，这些方案在业内均已得到充分论证并已由各大计算机互联网厂商部署使用多年，具有技术实现的理论基础和丰富的工程经验。

系统采用了 Python Flask 作为主要后端开发框架，该框架体量轻且具备灵活丰富的生态支持，能够满足本系统接口开发与部署的需求。数据库选用的是 MySQL8，如前文所述，MySQL 是全球开发者的数据库首选，非常成熟，并且稳定性高，支持复杂的增删查改及事务管理，完全适于本系统的开发，另外，高并发场景下，MySQL 足以支撑系统对海量数据的高效访问。此外，系统采用分布式架构，必要性已在 1.3 节中有说明（将性质不同、依赖硬件不同的模块部署于不同机器，具体请参见 1.3 节），为了实现分布式架构，系统引入了 Rabbit MQ 来进行异步消息队列管理，若非使用框架，这是业内的常见做法。

在人工智能模型方面，本项研究以 Qwen2.5-3B-Instruct 为底模，使用 LoRA 算法在 LLaMA-Factory 框架上进行了医学领域微调，事实证明明康慧医大模型进一步提升了 Qwen2.5-3B-Instruct 在医疗问答与诊疗辅助任务下的表现。近些年越来越多的对比学习模型已用于医学分析^[10]，BioMedCLIP 模型正是对比学习模型的一种，经严谨的测试，其完全具有辅助医师诊断疾病的能力^[11]，可用于实现本系统多模态诊疗的核心功能。

前端采用 Vue 3 框架配合 Element Plus、Axios 等组件，不仅使页面更加美观，而且实现了页面的高响应性，能够提升用户体验。

本系统所采用的技术方案均为当前行业广泛应用、成熟且社区支持活跃的开源技术，系统开发前本人已完成了技术预研，本次研究的技术栈能够充分满足系统的各项功能实

现需求，在技术上是完全可行的。

（2）经济可行性

从经济角度分析，系统整体采用的软件与工具均是开源的，Flask 框架、MySQL、RabbitMQ、Vue.js 等均无需额外为软件许可费用投资，能够有效降低运维成本。在生产环境中，系统的核心算力应当部署于 NVIDIA 显卡计算集群，但是系统中所有 AI 模型的参数量的加和不超过 6B，并且模型推理并不消耗巨大的算力资源，一个中低端 NVIDIA 显卡计算集群即可满足本项目所有模型推理及分布式运算所需的显存和算力的需求。在微调明康慧医大模型的过程中，LoRA 算法相较全参微调可大幅降低显存与计算量，本次研究使用了 2 张 11GB 显存的 NVIDIA GeForce RTX 2080 Ti 显卡便完成了对明康慧医大模型的微调工作。后续的研究与模型的升级几乎不需要购置大批硬件，无需大规模的资金投入。

整体来看，系统在资金投入方面主要集中在服务器硬件的购置和运维，这些对开发一个软件来说是必须的，因而资金需求量并不算大。但是系统在智能诊疗与数字化健康管理领域的市场应用前景很广阔，一旦系统稳定运行并投入市场使用，定能在医疗提效和辅诊等多个方面带来经济效益。因此，在经济层面上，系统具有可行性。

3.2 系统功能性需求分析

本研究的目的是构建一个包含健康管理、智能辅诊、医疗分析与问答、医患互动、资源共享等服务的智慧医疗平台。为了保证系统功能的科学性，项目中须对系统功能需求进行细致全面的分析。本系统主要面向的是医师用户与患者用户，同时也配有管理员端，整体功能可大致分为用户信息管理功能、智能诊疗功能、用户交流功能、资源管理功能，以及后台管理功能这么几大类。各部分功能需求如下：

（1）用户信息管理需求

客户端是用户使用系统的主要界面，需实现用户登录注册、个人信息管理、信息展示等基础功能。用户登录方式支持手机号、邮箱及系统分配随机的账号三种形式，满足不同用户习惯。系统还应满足用户修改完善个人信息，设置信息可见权限的需求，以尊重用户隐私。

（2）智能诊疗需求

临床实践中，对于医师来说大多数疾病诊断的难点和耗时点是病症仅有少数的几种可能性却难以定夺，毕竟但凡一名合格的医师不可能完全不懂医疗，那么医学影像学辅诊中，医师的需求就应当是能够上传 CT、MRI、X 光片等各类医学影像以及输入少数

几种诊断文本描述，系统应该能推理出各描述为正确的相对概率分布，从而辅助医师进行诊疗决策。由于该系统主要面向中国人开发，故应支持汉语，但医学文本使用英语描述更加严谨，故系统还需支持英文原生描述输入。另外，系统需将计算结果作图表化展示与表格数据展示，可提升诊断结果的直观性。

（3）医疗问答与深度分析需求

上述第（2）条需求中，分析了如何解决跨模态辅诊的问题，但是（2）中的解决方案无法解决纯知识性问题和用户没有任何头绪的问题，对非医学专业者也不友好。故系统需依托大模型能力，支持用户进行开放式问答和历史会话记录的保存与加载。

医学领域的机器问答需求有别于生活中常见的文本生成任务，医疗相关文本应当力求准确严谨，而大模型的“幻觉”现象目前无法避免，那么系统需要通过 RAG 方法增强对话输出。而有些医学问题不能浮于表面，需要推理思考，因此系统还需一种支持深入思考的机器问答模式。

（4）诊疗事项清单需求

老年患者及工作繁忙的患者用户常因客观原因忘记执行医嘱，当医嘱内容复杂繁多时，管理治疗事项的需求便更加迫切，因此本系统设计了诊疗事项清单管理功能，用户可创建各类型事项，并能够设置事项的完成状态与优先级。为了节省用户宝贵的时间，系统还要调用 LLM 能力，使机器辅助分析列表内容的合理性。

（5）资源中心需求

为了能共享资源，解决纸质医疗资料难以流通的缺点，系统应支持医患用户上传分享生物医疗相关文件。系统还需通过某种手段，使用户可以便捷的检索到资料，并利用资料知识增强 LLM 的输出。

（6）诊疗论坛需求

不论是医师之间还是患者之间，都需要多人在一起交流讨论诊疗知识和患病情况，系统开设医疗论坛平台的需求正因此显现。受现有论坛网站的启发，本项目构建的诊疗论坛平台应具备创建论坛、发布带图文帖子、评论点赞回复等交互功能。论坛还需支持分类、权限管理，确保论坛可达到预期效果。

（7）后台管理系统需求

任何一个应用软件系统都必须要有的是后台管理系统，也就是后管端，它仅由系统管理员登录，有全权查看、修改系统数据。本系统中，后管端需要包含对用户信息、各诊疗事项、论坛帖子、用户分享的资源、问答数据、病历等内容的全面管理与审核，系

统管理员可通过后管端查看修改系统数据。另外，为了降低系统架构耦合性，后管端须独立部署，在逻辑上与用户系统隔离。

3.3 系统非功能性需求分析

系统的非功能性需求事关平台稳定性、安全性和界面美观性，直接影响到用户体验与系统工程的质量。本章节将从代码性能、安全性和美化界面三个角度分析。

（1）性能需求

任何好的 Web 系统都须具备良好的响应速度与高并发处理能力，大模型推理、医学影像分析等涉 AI 推理的复杂计算的模块，应通过分布式架构及 MQ 任务队列的设计，保证系统整体运行的流畅性。前端代码效率也非常重要，根据软件工程师们的共识，4G 网络环境下，CSR 前端若能使用用户体验良好，除去图片字体等静态文件的包大小不应超过 2MB，^[26]这也要求开发时需时刻关注代码规模和注意选用的打包工具的打包质量。

（2）安全性需求

系统由于涉及敏感的医疗数据及用户隐私信息，必须严格执行权限管理和数据访问控制。具体来说，明康慧医系统应采用 flask_jwt_extended 框架提供的 JWT 完善的用户身份鉴权机制；后端所有 SQL 语句均要采用参数化查询，以免受 SQL 注入攻击；大模型生成的内容也需要先用 DOMPurify 过滤后再由 Vue 挂载渲染，避免受 XSS 攻击。

（3）美化界面需求

系统 UI 界面是系统的门面，设计应大气简约、直观流畅，符合医师与患者用户的使用习惯，降低用户学习使用本系统的成本。^[17]本系统还需支持用户跨终端访问，适配 PC 桌面端与手机移动端，满足用户多场景的使用需求。为了满足上述两点要求，前端须使用一套 UI 框架来完成界面主题的设计，而且还要注意视口切换时样式表的改变。

第 4 章 系统设计

4.1 系统的构建

本研究研发的系统“明康慧医——基于 LLM 与多模态人工智能的健康管理与辅助诊疗系统”采用多层分布式架构设计，遵循“前端后端分离、后端分布部署、各后端模块间解耦”的原则，系统宏观架构自顶向下包括经典的视图层、业务逻辑层、智能服务层、消息队列中间件与数据层，耦合度低，非常易于拓展，并且这样的设计模式相较于简单的前-后端架构甚至是前后端不分离的架构来说，极大地提高了系统的稳定性和容错性，也能有效满足医疗领域复杂数据处理的业务需求。而且本系统的 UI、请求处理、数据、模型推理四部分分离，高并发请求、大规模数据的存储以及 AI 模型推理计算的任务能够很好地被这种架构的设计支持处理。

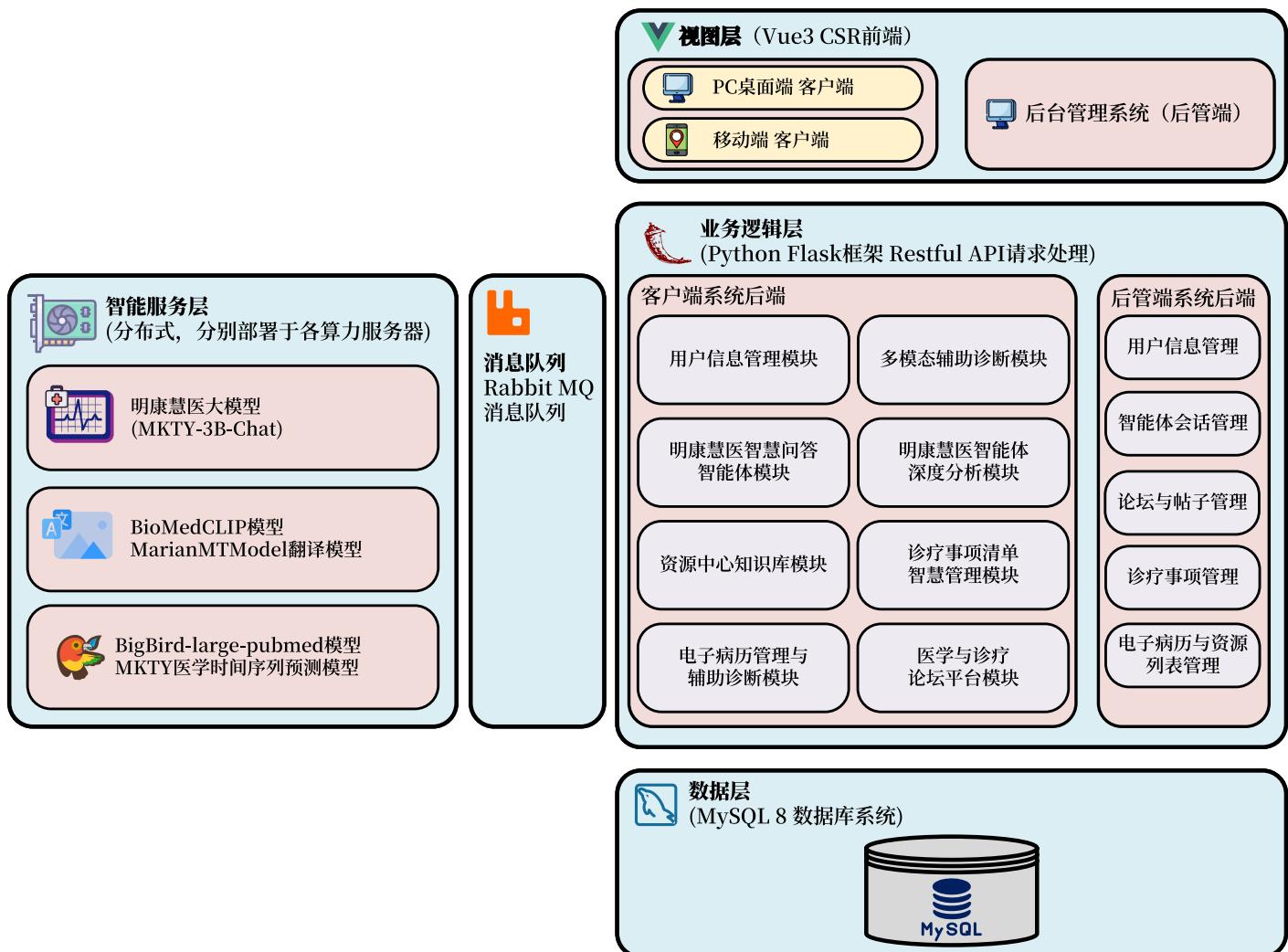


图 4-1 系统架构图

系统中的各模块均通过标准化接口进行通信：视图层于客户端展示，与业务逻辑层通过 HTTP 请求与响应作通信；业务逻辑层与数据层（即：MySQL 数据库管理系统）依赖 DBMS 内置的连接机制，本质上是基于 TCP 连接；智能服务层在物理上是分布式部署的多个 AI 模型，这部分都仅与业务逻辑层相关联而不与数据层或其他层关联以解耦。连接方式基于消息队列，各智能服务相当于业务逻辑来说均是透明的，业务逻辑层只需作为消息生产者调用函数发布消息，消息的调度由 MQ 自动完成，智能服务层的 AI 模型收到消息后便可开始推理模型消费消息，随后将推理结果返回至业务逻辑层，这样做不仅可非常方便往整个系统架构中添加业务逻辑后端和智能服务后端，还能消减请求洪峰，避免智能服务端并行计算压力过大。

（1）视图层

如果把一个 Web 应用比作一家公司，那么视图层便是公司的前台，是首先直接和“客户”打交道的人，是公司的门面。本系统中，视图层的主要职责是系统用户交互界面与数据展示界面的实现，其重要性不言而喻，项目中采用了 Vue3 作为前端主开发框架，结合 Element Plus UI 组件库，实现了清晰直观、操作便捷的多端适配界面，用户可通过 PC 桌面端或移动端登录系统并使用。事实上，Element Plus UI 库最初并不是为了多端适配而设计的，它的目标是帮助开发者快速构建表单界面或管理端界面，但经测试说明，由 CSS 媒体查询触发执行修改指定的 Element Plus 组件，可以很好地适配移动端竖屏的视口。另外，系统适当地引入了 JQuery 库来方便处理登录注册逻辑，该部分在具体编码时特地作着重关注，确保了 JQuery 库不会与 Vue 操作虚拟 DOM 的逻辑相冲突；

项目中还引入了 Axios、marked.js、DOMPurify、ECharts、highlight.js 等重要的框架，分别完成了 AJAX 请求发送、Markdown 解析、HTML 代码净化、图表绘制以及代码高亮等功能，具体说明请回见本篇论文第 2.3 节。同时，系统还配套设计了后台管理系统（后管端），该模块的视图层前端和业务逻辑后端均是独立于客户模块的。后管端用于支持平台运营管理、数据维护、智能体会话管理、论坛管理及诊疗事项监管等数据管理功能。前端采用客户端渲染（CSR）模式进行渲染，本次研究项目答辩时采用 Python 解释器自带的 http server 进行部署，后续可使用 Nginx 等任何 Web server 部署前端，同时考虑使用 Capacitor 生成 Java 代码，并由 Android Studio 将其打包为安卓 APK 包文件，供安卓手机当作应用安装使用。

（2）业务逻辑层

业务逻辑层是明康慧医系统功能实现的核心，同时也是系统的枢纽，联系着前端视

图、数据库层和各 AI 模型服务，通俗来说，它是本系统“狭义上”的后端，是系统后台的“第一层”，负责实现能够接收处理所有来自前端请求的 Web API。该层基于 Python Flask 框架开发，通过 Restful API 风格标准对外暴露接口，同时利用 mysql 库将位于数据层的 MySQL 8 数据库作连接，系统的业务逻辑层分为客户端系统后端与后台管理系统（后管端）后端两大子模块，承担各自不同的业务职责。概括的讲，本层客户端后台与后管端后台互不相关，各自实现业务逻辑处理视图层请求并从数据层取数据，必要时可向 MQ 发布消息调用各 AI 模型能力。

客户端系统主要面向的是医师和患者等普通用户，包含用户信息管理、多模态辅助诊断、明康慧医智能问答、深度分析、资源中心、诊疗事项管理、电子病历管理、医学论坛等功能模块，其中用户信息、资源中心和医学论坛三个模块为纯增删查改操作，不包含对智能服务的调用，智能问答、深度分析涉及调用明康慧医大模型会话的能力，事项管理和电子病历需要使用明康慧医大模型总结医疗领域不规则语段和基于知识的文本生成的能力，多模态辅诊则基于 BioMedCLIP 放射科影像对比学习模型和 MarianMTModel 神经机器翻译模型；此外，电子病历诊断功能界面中要附加基于医学文书的医学时序预测功能，大模型讨论机制（即深度分析功能）也需要依赖 BigBird BERT 计算讨论结果的句子嵌入。

再具体来说，用户信息模块即为系统中“个人主页”以及用户登录注册页面，除了注册登录外，还包含的子模块有信息展示、信息权限与修改、留言与查看留言和退出登录。除注册登录页，用户访问本系统任何页面都需要浏览器中 Cookie 存有 userId 以及 token，即保持登录状态，用户登录成功后会从后端得到 token 字符串，其中包含 userId 及其校验和等防篡改信息，若不考虑注册登录，业务逻辑后端所有 API 均使用 flask_jwt_extended 框架提供的 JWT 用户鉴权，任何调用 API 的请求都须在请求头里包含 token 字符串才可受权访问，否则将会被安全拦截。

（3）数据层

数据层是明康慧医系统的“最底层”，它一般包含数据库管理系统、缓存系统与文件系统，负责数据的 CRUD 等操作，当然本项目里没有显式地安装数据库地缓存。本系统采用 MySQL8 作为数据库管理系统，负责存储与管理本系统中的用户信息、诊疗事项、模型任务记录等所有结构化的表数据。系统数据库设计得可以确保所有字段只依赖于主键，不依赖其他非主键字段，因而范式可以达到 3NF，数据结构的规范性和可维护性可以确保。^[21]在数据安全方面，用户密码密文由业务逻辑层采

用 argon2 算法对用户密码明文进行高强度加密而产生，并由数据库存储。这个方法属于业内的共识，攻击难度巨大，理论上非常安全。在业务层控制权限机制配合下，数据访问的合法性得以保障。^[32]为提升查询性能与用户体验，系统启用了索引，能够优化减少数据库负载，高并发场景下的数据查询压力则能消减。

（4）智能服务层

智能服务层是本研究系统中重要创新点的载体，该层实际上是逻辑上的“分层”，各智能服务按功能划分独立部署，也就是层中各模块均应当分布式地部署于不同硬件配置的机器上^[20]，这些模块分别承载了多模态推理、大模型问答、深度讨论、时序预测等关键智能功能，项目中复杂模型计算任务的高效处理能力可通过如此架构得以保证，系统的灵活扩展能力也可得到显著体现。^[19]理论上各模型都应在具备高性能计算卡算力的集群节点上推理计算：明康慧医大模型（MKTY-3B-Chat）要求所部署的机器的 GPU 显存不可少于 10GB；对比学习模型 BioMedCLIP 与翻译模型 MarianMTModel 级联模型组合体所需 GPU 显存至少需要 6GB；BigBird-large-pubmed 与 MKTY 医学时序预测模型级联模型组合体所需至少 2GB 显存。具体部署的模型包括：

1. 明康慧医大模型（MKTY-3B-Chat）：基于通义千问 Qwen2.5-3B-Instruct 模型底座，通过 LoRA 算法和 LLaMA Factory 框架完成中文医学领域微调。微调算法采用的是增量预训练（Pretrain）与指令监督微调（SFT）两种，并分四个微调步骤进行，具体来说就是一轮增量预训练+一轮指令监督交替进行两次，这样做是考虑到底座模型规模不大，吸收知识的能力跟巨大规模模型相比稍逊一筹，若只进行一轮微调，那么经过 SFT 后大模型可能会遗忘掉其先在增量训练阶段学习到的知识，执行两轮微调可避免大模型的灾难性遗忘。^[15]微调所用语料的形式为医疗生物领域广泛文本、医学 QA、医学诊断、医学考试选择题、自我意识数据集等。经过训练，模型支持医学问答、病例分析、自我讨论等专业文本生成场景，具备较强的医健专业领域知识，训练详细过程此处从略，请参见第 5 章“系统实现”部分的论述。

2. BioMedCLIP 与 MarianMTModel 级联模型：实现医学影像与多个文本的多模态比对推理，用户可以上传医学影像与对应多项待诊断的文本描述，CLIP 模型的图像编码器将计算影像的特征向量，文本编码器可以计算各条文本的特征向量，随后计算影像特征与各文本特征的余弦相似度，随后通过 Softmax 得到各文本描述的正确性概率分布向量。由于 BioMedCLIP 的文本模块均使用英文医学期刊内容训练，故只支持英语输入，但系统主要面向中国用户使用，为解决此问题，级联架构使用 MarianMTModel 模型将

汉语翻译成英语，以支持系统的自动翻译，提升模型在汉语言语境下的可用性。

3. BigBird-large-pubmed 与 MKTY 医学时序预测模型：BigBird-large-pubmed 也是一款用医学期刊语料训练的 BERT，利用该 BERT 可以计算医健领域某语句的嵌入矩阵，从而可以计算出其句子嵌入。一方面，它是“智能体深度分析”功能的一部分，可以在大模型讨论的最后计算讨论结论的语义相似度；另一点，它是 MKTY 医学时序预测级联模型的组成部分，用户输入的语言通过它计算出特征向量，也就是句子嵌入，后续 MKTY 医学时序预测模型会拿到这个嵌入，进行基于多模态融合的推理预测。总结来说，该逻辑节点支持医学问题深度讨论的收敛度评估及诊疗事项的多模态时间序列预测。

本层中各节点、各模型通过且仅通过 Rabbit MQ 实现与业务逻辑层的通信和任务调度，业务逻辑层仅需向消息队列提交任务，由智能服务层自主接收、推理计算、回传结果。此机制“一箭三雕”，符合分布式计算系统的设计理念，既降低了业务层负载，又能提升了系统整体的任务并发处理能力，还能在逻辑上解除耦合，降低扩展难度。

4.2 系统的框架及数据库设计

前文已稍加详细地描述了系统整体的宏观结构和每一个层次的主要作用和大致实现方法，在本节中将着重讲解系统更加具体的设计和数据库的建立与设计。

下图展示了明康慧医系统的功能模块图：

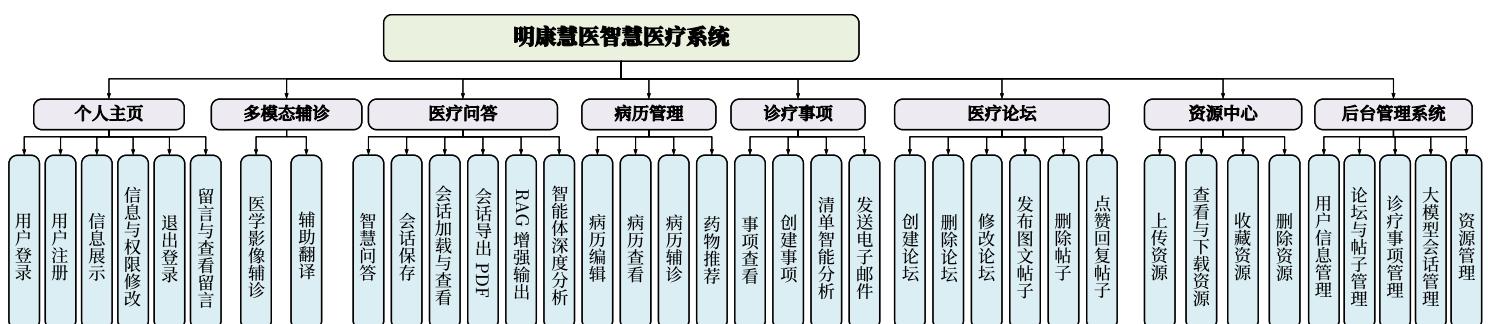


图 4-2 功能模块图

用户信息模块即为系统中“个人主页”以及用户登录注册页面，“个人主页”可支持查看他人和自己的基本信息，该部分信息可分别设置是否公开的权限，打开自己的主页有全权看到各条信息，打开别人的主页只能看到已公开的信息，每位用户还可方便的修改自己的信息和相应权限，当然也包括密码和头像的更改。此外，“我的主页”上还有查看用户留言的功能，该部分提供了留言及查看的功能，在自己的页面上，用户可看到收到的留言及自己发布的留言，单击头像可跳转至对应用户主页，在外人的页面上，可

以发送留言，自己发布的留言用户有权删除之。

登录页面支持多种登录方式，页面上有两个文本框以输入登录键和密码，还有一个选择框，对于登录键用户可以操作勾选或取消勾选选择框，选择使用用户 ID 或者是联系方式进行登录。除登录注册 API 外，系统全局采用 flask_jwt_extended 实现的 JWT 进行身份鉴权验证，客户端于 Cookie 存储 token 并在之后向业务逻辑层后端发起 Web API 请求中读取使用。

用户注册模块需要详细的个人信息以收集，收集内容有基本信息，包括：真实姓名、性别、用户类型（医师或患者）、年龄、联系方式（电话号码或电子邮箱，可用于登录，必须唯一）、来源地（医师为工作单位、患者为现居地）、用户自述、用户重要信息声明，还可以设置头像、密码，以及各字段的外人查看权限，除了头像、姓名和用户描述，其他每项都可设置权限是否公开，查看自己的信息时默认一定全部显示，查看别人的信息时按权限显示。上述提及的所有针对用户的信息都将存储于数据库中，下表展示了用户信息表的设计：

表 4-1 用户信息表（userinfo）

字段名	中文名	类型	非空	说明
userId	用户 ID	bigint	是	每个用户的唯一编码，注册时随机分配生成。
userCiphertext	用户密码密文	text	是	用户密码的加密值。
userName	用户姓名	text	是	用户姓名。
userType	用户类型	text	是	用户类型: 0 - 患者, 1 - 医师, 2 - 其他。
userSex	用户性别	tinyint(1)	是	用户性别: 1 - 男性, 0 - 女性。
userSexPermission	性别查看权限	tinyint(1)	是	0=公开; 1=仅自己可见
userAge	用户年龄	text	是	用户年龄，以文本存储。
userAgePermission	年龄查看权限	tinyint(1)	是	0=公开; 1=仅自己可见

字段名	中文名	类型	非空	说明
userFrom	用户来源地	text	是	患者填写城市，医师填写机构名称。
userFromPermission	来源地查看权限	tinyint(1)	是	0=公开；1=仅自己可见
userContact	用户联系方式	text	是	电话或邮箱。
userContactPermission	联系方式权限	tinyint(1)	是	0=公开；1=仅自己可见
userDescription	用户自述	text	否	用户自述，限制 1024 字符。
patientImportantInfo	患者特有信息	text	否	慢性病、过敏、家族遗传病史记录等。
doctorImportantInfo	医师特有信息	text	否	科室与擅长专业。
userImportantInfoPermission	重要信息权限	tinyint(1)	是	0=公开；1=仅自己可见
userAvatarId	用户头像 ID	text	否	用户头像 ID (GUID)。
userRegisterTime	注册时间	bigint	是	以秒为单位的 Unix 时间戳。

下表展示了用户留言信息表的设计，其中留言者用户 ID 与被留言者用户 ID 是指向用户信息表（userinfo）的外键。

表 4-2 用户留言表（mailitem）

字段名	中文名	类型	非空	说明
mailItemId	留言信息 ID	bigint	是	唯一标识留言信息。
mailFromUserId	留言者用户 ID	bigint	是	留言发起者 ID。
mailToUserId	被留言者用户 ID	bigint	是	留言接收者 ID。

字段名	中文名	类型	非空	说明
mailItemContent	留言内容	text	是	留言信息文本内容。
mailItemStatus	留言状态	int	是	0=正常; 1=已删除; 2=其他。
mailItemSendTime	发送时间	bigint	是	以秒为单位 Unix 时间戳。

多模态辅诊功能模块面向医师，支持基于医学影像的决策分析，其实现了医学 CT、X 光和 MRI 影像的分析与文书文字描述的推理比对，显著提升医师定夺诊断结论的效率。用户首先从视图层（前端）依次输入文本创建待确定的医学文书列表，可以选择源语言为中文或英文，若直接使用英文医学描述则不会造成语言转译造成的语义损失，可获得更好的推理效果，然后上传图片，单击图片还可以放大查看。开始分析后一段时间内应该可以得到机器辅诊结果，结果首先以 ECharts 饼图展示，也可打开表格查看概率预测数值百分比、文书原内容与翻译后相应的英文结果。这一部分功能是无使用记录实时使用的，因而系统没有设立多模态辅诊模块的数据库。

关于“医疗问答”功能：该功能基于 MKTY-3B-Chat 大模型，具体关于该模型的内容请参考前文，此处从略。“医疗问答”模块分为两部分，第一是明康慧医智慧问答部分：普通的大模型对话功能，可以进行 RAG 增强，前端渲染支持将 markdown 渲染为 HTML 并挂载到 Vue 的会话展示组件，每轮成功的会话后，自动都保存更新会话历史，用户可加载会话历史以查看或继续对话。第二部分，明康慧医智能体深度分析：基于“大模型讨论机制”的医疗问题深度研究。原理简介：第一轮讨论中，前端模拟多个智能体，让每个智能体分别回答待研究的问题，然后由“主持人”总结各方发言，以后每轮讨论，都将上轮主持人的总结和原问题拼接，并由各智能体再分别回答，最后主持人总结，周而复始，最后是“判敛”：用 BigBird 将最后一轮讨论各方的输出计算句子嵌入向量，然后计算各向量两两之差的平均值，以此反应各方达成共识的程度，即讨论语义收敛程度。后台只暴露出会话任务提交和轮询两个接口，“讨论机制”的逻辑实际上位于前端，调用这两接口实现的。这两大功能共用一个数据库表，表中有字段记录会话类型，可以区分普通对话和深度讨论；8.0 版本后的 MySQL 数据库支持 JSON 数据类型，故数据库的会话内容字段采用 JSON 存储，对于普通对话，JSON 中有 role 和 content 两个键，role 为

“user”或“assistant”，表示会话方，content 为对话内容渲染后的 HTML，可直接挂载；对于 AI 深度讨论，JSON 有 title 和 desc，title 为会话方描述，比如“智能体 A 讨论”，desc 为具体内容（纯文本）。以下表格展示了存储医疗问答功能模块的数据库设计：

表 4-3 大模型问答历史记录表 (llmhistory)

字段名	中文名	类型	非空	说明
sessionId	会话 ID	bigint	是	唯一标识一个会话。
isSessionDM	是否启用讨论机制	tinyint(1)	是	是否启用大模型讨论机制。
sessionSaveTime	保存时间	text	是	Unix 时间戳(秒)。
sessionUserId	所属用户 ID	bigint	是	会话所属用户 ID。
sessionContent	会话内容	json	是	存储会话内容 (JSON 格式)。

关于“诊疗论坛”功能：论坛平台首页显示论坛列表，包含论坛名称、编号、创建时间、隶属类别、论坛权限、创建者姓名及头像。用户可通过输入关键字以及选定论坛类别与权限来筛选论坛，实现快速查找，通过论坛类别与权限的筛选在后台进行，通过关键字的筛选在前端进行。用户还可通过设置名称、类型、权限来创建论坛。进入论坛，若权限不符会被拦截；顺利进入后，可查看各条帖子，每条帖子上显示用户头像、姓名、类型、来源地、帖子发布时间、赞数，同时帖子支持文字+图片，单击图片可以放大查看并可滚动查看帖子内所有图片。用户可对所有帖子点赞、回复，对于自己发布的帖子还可以删除。用户可任意发布帖子，帖子可以配图，最多 3 张，不能只有图没有文；用户单击“添图”按钮后可选择图片文件添加，添加后会有一个列表，用户可放大查看每张图，也有一个删除按钮可一键清空图片。对于该模块，数据库的设计需要两张表，否则将不满足第三范式了，但总体并不复杂，首先是论坛列表，存储着系统中创建有哪些论坛，包含，论坛名称、创建时间、创建者 ID 等，创建者 ID 是外键，还有状态字段表明删除状态；另外一张表是论坛内容表，也就是帖子的列表，里面存有帖子所属论坛、

发帖人等，帖子内容字段是 JSON，其中 content 字段存储帖子文本，images 字段存储帖子图片 GUID 列表，每个列表也是一个字符串，有特定字符分隔各 GUID。以下两张表格是诊疗论坛表的具体设计：

(1) 论坛列表：

表 4-4 论坛列表 (forumsummary)

字段名	中文名	类型	非空	说明
forumId	论坛 ID	bigint	是	唯一标识论坛。
forumCreator	创建者用户 ID	bigint	是	创建者 userId。
forumName	论坛名称	text	是	论坛名称。
forumType	论坛类型	tinyint(1)	是	0=医学知识论坛； 1=疾病论坛。
forumCreateTime	创建时间	text	是	Unix 时间戳(秒)。
forumPermission	论坛权限	tinyint(1)	是	0=不限类型； 1=仅医师； 2=仅患者。
forumStatus	论坛状态	int	是	0=正常； 1=已删除； 2=其他情况。

(2) 论坛帖子列表：

表 4-5 论坛帖子列表 (forumcontent)

字段名	中文名	类型	非空	说明
postId	帖子 ID	bigint	是	帖子唯一标识。
postCreateTime	帖子发布时间	text	是	Unix 时间戳(秒)。
postPosterId	发帖者用户 ID	bigint	是	发帖者的用户 ID。
postForumId	所属论坛 ID	bigint	是	帖子所属论坛 ID。

字段名	中文名	类型	非空	说明
postContent	帖子内容	json	是	含 content 文本与 images 列表。
postPraiseNumber	点赞数	int	否	被点赞数量，默认 0。
postStatus	帖子状态	int	是	0=正常； 1=删除； 2=其他情况。

“病历管理”功能实质就是一个基于 Markdown 的编辑功能，医师用户可创建病历，并绑定病患 ID。医师有编辑病历的权限，而患者只有查看病历的权限。此功能模块提供基于大模型的病历辅诊、药物推荐的功能，详细原理与“医疗问答”等模块完全相同，具体可见前文。数据库设计上有一点需说明：表中“病历资源列表”字段是系统为后续支持多模态病历而设置的，本次研究中因时间原因不再设计多模态病历（病历中图像与音频等可由外部链接链入，系统内部将不予存储，也不予分析）。以下是电子病历表的字段设计表：

表 4-6 电子病历表 (medicalrecord)

字段名	中文名	类型	非空	说明
medrecId	病历 ID	bigint	是	唯一标识电子病历。
medrecCreateTime	创建时间	text	是	Unix 时间戳(秒)。
medrecModifyTime	修改时间	text	是	Unix 时间戳(秒)。
medrecPatientId	所属患者 ID	bigint	是	病历所属患者 ID。
medrecMainDoctorId	主要医生 ID	bigint	是	病历主要负责人 (医师) ID。
medrecMinorDoctorId	其他医生 ID	bigint	否	病历其他负责人 (医师) ID。

字段名	中文名	类型	非空	说明
medrecAbstract	病历概要	text	否	简洁描述病历。
medrecState	病历状态	text	是	0=正常生效； 1=痊愈无效； 2=慢性病优先级降低。
medrecRight	病历权限	text	是	0=仅患者与医师可见； 1=公开可检索。
medrecEigenVecto r	病历特征值	json	否	存储病历 TF-IDF 特征。
medrecResList	病历资源列表	json	否	存储上传的资源 ID。

“诊疗事项”功能部分是为诊疗事项清单查看与编辑功能。每一项事项都有完成情况、时间、状态、事项类型、优先级等内容，每项展示各信息的概要，并以不同颜色醒目展示重要信息，单击进入后，会弹窗展示详细信息。用户可以将事项标记完成，也可以删除事项，事项类型分为一次性事项（时间段）、周期性事项（星期、时间点）、无时间要求（不设时间），前端读取当前服务器时间与各事项要求时间作对比，“时间状态”据此显示“未到时间”、“已到时间”、“已超时”；同时，紧急事项与未完成事项将在主页显示，前文已讲到，参见前文。模块中还有“医患互联”功能，实质是通过 markdown 编辑器编辑邮件，渲染为 HTML 并向指定邮箱发送。模块中还包含“AI 辅助分析”功能：基于 MKTY-3B-Chat 大模型，可判断计划合理性，并给出建议，大模型调用原理与“医疗问答”模块相同，可参见前文。对于本模块的数据库设计，包含系统中所有用户的每条事项，包含创建者（外键）、事项内容、时间信息以及事项类型等，以下是具体的数据表设计：

表 4-7 重要事项清单表（importantlist）

字段名	中文名	类型	非空	说明
listItemId	清单项 ID	bigint	是	唯一标识一条事项。

字段名	中文名	类型	非空	说明
userId	用户 ID	bigint	是	所属用户 ID。
listItemContent	医疗事项内容	text	是	事项具体内容。
listItemStartTime	事项开始时间	bigint	是	一次性事项开始时间（Unix 时间戳）。
listItemEndTime	事项结束时间	bigint	是	一次性事项结束时间（Unix 时间戳）。
listItemPriority	事项优先级	tinyint	是	0=正常；1=非常重 要。
listItemStatus	事项状态	tinyint	是	0=正常；1=删除； 2=其他。
listItemIsFinished	是否完成	tinyint	是	0=未完成；1=已完 成。
listItemTimeMode	时间模式	tinyint	是	0=一次性事项；1= 周期性事项；2=无 时间声明。
listItemTimeWeek	周期事项星期	tinyint	是	1=星期一；7=星期 日。

“资源中心”部分功能：用户上传资源，可以为 pdf 文件、doc 文件、txt 文件、ppt 文件等。后台存储原文件，同时提取文件中的文本并切片，计算每片的 TF-IDF 特征。用户可收藏资源并查看收藏的资源列表，收藏的资源也被称作“知识实体”，可用于大模型 RAG 使用。这部分主要由两个数据库表来实现，分别存储已上传资源和用户收藏。
下面是两张数据库表的字段设计表，分别表示上传的资源和用户的收藏：

表 4-8 知识实体表（knowledgeentity）

字段名	中文名	类型	非空	说明
-----	-----	----	----	----

字段名	中文名	类型	非空	说明
keId	知识实体 ID	bigint	是	唯一标识知识库实体。
keFileType	知识实体源文件类型	text	是	文件 MIME 格式。
keName	知识实体名称	text	是	展示时的标题。
keCreateTime	上传时间	text	是	Unix 时间戳(秒)。
keAbstract	知识实体概要	text	否	知识实体概要。
isKeMultimodal	是否为多模态实体	tinyint(1)	是	True=是。
keTextEigenVector	文本特征值 s	json	否	各文本特征。
keResList	资源列表	json	否	解析出的资源 ID。

用户收藏表：

表 4-8 用户收藏表 (knowledgeentity)

字段名	中文名	类型	非空	说明
recId	记录 ID	bigint	是	唯一标识一条收藏记录。
userId	用户 ID	bigint	是	表示该记录的用户。
keId	知识实体 ID	bigint	是	表示用户收藏的知识实体。

后台管理系统（后管端）：可由全权管理员登录，用以管理数据库各项字段，若某字段是 json 格式，还可利用 highlight.js 高亮显示。可新增、删除、查看系统各项内容。该部分逻辑上独立于系统，仅通过数据库与主系统关联，数据库中各项表均由后管端管理。

第 5 章 系统实现

本章的主要内容为“明康慧医”——基于 LLM 与多模态人工智能的健康管理与辅助诊疗系统的具体实现内容与过程，整项研究的基础工作以及系统项目的宏观结构、技术栈、各模块功能以及数据库建库逻辑已经在前文全面叙述，本章不再赘述。本章中将以系统功能来分节，具体到代码讲解每一部分的实现方法，并会附上该部分的系统运行效果示意图。

5.1 总体设计论述

本系统全称为明康慧医——基于 LLM 与多模态人工智能的健康管理与辅助诊疗系统，为了便于开发维护与管理、降低系统各部分耦合度，系统宏观采用前后端分离的分布式架构，前端视图、业务逻辑请求处理、各人工智能模型、数据库均各自独立部署于配置性能不同的机器，并且系统客户端分系统与管理端分系统也相互逻辑独立，仅通过数据库关联。关于系统架构的介绍可参见前文各部分。

再介绍一下系统名称与 LOGO 的有关情况。本项研究的系统名称为“明康慧医”，体现了系统“通过‘聪明与智慧’的技术医治患者使其健康”的初衷和宗旨，系统的英文名为“Minh Khoa Tue Y”（简称 MKTY），取自“明康慧医”四个汉字的越南文国语字拼写（Minh Khoe Tuę Y）。系统 LOGO 分为中英文两版，中文版左侧主体形状为一个心电图波形，左上角有一个小医药箱符号，寓意“智慧医疗”，LOGO 右侧为本系统繁体字名称与英文名称的字迹。本人设计该图像所使用的软件是 Adobe Illustrator，这是一款专门用于设计、制作矢量图的软件。LOGO 中的医药箱符号和心电图的符号均取材自阿里巴巴矢量图标库（网址：<https://www.iconfont.cn/>），字体使用了 Minh Nguyen、Gentium Basic、Minion Variable Concept 字体，特此说明。

以下是系统中文 LOGO 图像：



图 5-1 “明康慧医”系统中文版 LOGO

以下是系统英文版的 LOGO 图像，与中文版不同的是，其右侧均为系统的英文名称。



图 5-2 “明康慧医”系统英文版 LOGO

5.2 用户信息模块

就视图层而言，具体来说用户信息模块包含用户注册界面、登录界面、登录后的欢迎页以及个人主页，一共四个页面。技术栈上以 Vue.js 作为主框架，结合 Bootstrap 进行页面布局，使用 Element UI 的加载组件实现加载状态。jQuery 处理 DOM 信息的读取，具体是用以读取文本框的文本值，这个过程可以保证绝对不会修改任何 DOM，也就是不会对 DOM 树有影响，这样做不会与 Vue “不得修改 DOM”的理念冲突。登录页面使用采用了 Vue 单文件组件（SFC）格式进行代码编写，该页面支持两种登录方式，使用联系方式登录或使用账号登录，用户可以通过切换复选框来选择登录方式。登录逻辑：用户单击登录按钮后，系统会验证用户输入是否为空，若初步判定输入合法，系统将登录键、密码以及登录方式代码通过 loginVerification API 发送至后端，后端进行校验，若登录成功，则将成功代码以及 token 字符串返回给前端，前端将 token 和 userId 存入 Cookies 并跳转至登录欢迎页。

另外，登录页是系统第一个与用户打交道的页面，能非常深刻地影响用户对系统 UI 的印象，故系统使用不断更替的优美图片作为背景。当打开登录页时，登录页面的 Vue 组件挂载，触发 mounted 钩子函数，执行预加载背景图片，加载到浏览器的图片首先将被编码为 base64 数据，然后存入数组，避免每次更换图片都向后端发起请求，然后创建 JS 定时器，每隔一段时间生成一个图片张数内的随机数，作为下一张背景图片的序号，最后从数组中取出 base64 字符串，写入 background-image 样式，从而实现随机背景，通过 CSS 设置，背景切换有平滑的过渡动画效果，页面加载时也会有模糊到清晰的过渡动画，营造“沉浸感”。页面还包含注册入口，未注册用户可通过点击“注册账号”按钮在新窗口打开注册页面。

登录页代码还实现了响应式设计，会通过 CSS 媒体查询检测窗口宽度，当屏幕宽度小于 640px 时认为是手机，自动跳转到移动端登录页面。样式方面使用了自定义字体

"penxingshu"，按钮和输入框有交互效果，如悬停时边框变色和背景色变化。下面的一张截图展示了系统登录界面。



图 5-3 客户端登录页面

系统注册界面参考了 Bootstrap 的一个 demo 页面，页面上的所有组件均采用 Bootstrap 风格并由 Bootstrap 提供。该页面实现了一个功能完整的用户注册表单，包含多种信息的设定和隐私权限的设置。技术方面，组件引入了多个外部资源，包括 Bootstrap 的 CSS 和 JS 文件用于页面样式和交互，同时，页面从项目的 utils 和 api JS 文件中导入了错误处理、消息处理和注册 API 函数。组件利用计算属性定义了多个方法处理数据的显示格式，如 userSex_text 函数将数字类型的性别值（0 或 1）转换为文本（"男性"或"女性"），用以渲染显示，这些计算属性还负责根据隐私设置代码转为相应的显示文本，具体情况请查阅代码。

前端使用 Bootstrap 的栅格系统和表单组件构建了一个响应式的注册界面。表单包括基本信息填写、密码设定、隐私权限设定和用户承诺等多部分，使用了 v-model 指令实现数据双向绑定，v-if 和 v-else-if 指令实现条件渲染，以及@click 和@change 等事件绑定处理。页面还包含一个使用 el-dialog 组件实现的"注册须知"弹窗，注册须知即使用协议条款，具体参见代码。

着重说一下对于用户头像的处理：用户选择图片后，先将图片进行裁剪处理并转换为 Base64 数据。系统使用 FileReader API 读取用户选择的图片文件，检查文件大小是否

超过 10MB，并将图片内容转换为 Data URL 格式显示在页面上，也就是渲染出用户选择的图片。随后再使用 Canvas API 将图片裁剪为正方形并调整为 200*200 像素的尺寸，最后转换为 webp 格式图像的 Base64 字符串数据，发送注册请求时连同其他字段以 JSON 发送到后台。

提交注册请求时首先检查各必填字段是否已填写，包括用户名、用户类型、性别、年龄等，以及密码是否一致、是否同意注册须知等。初步验证通过后，将用户数据通过 register API 函数提交到后端，并根据返回结果显示成功或失败消息。

对于后端，系统使用了 argon2 算法进行密码密文生成。Argon2 算法是目前最先进的密码哈希算法，它是密码哈希竞赛 (PHC) 的获胜算法，专门为安全密码存储而设计，能抵抗侧信道攻击和暴力破解等多种攻击。Argon2 算法目前是事实上的行业标准，可以认为绝对安全，用户设置的密码将经 argon2 计算后存入数据库。另外，后端接收到前端传到的请求后将进行各项信息的解析，以存入数据库，其中用户头像的 base64 字符串会解码为 webp 文件，写入服务器硬盘，并为该文件分配 GUID，数据库中头像字段将以 GUID 存储该图。读取头像时执行相反操作，后端读取文件，解析为 Base64 并返回给前端。后端中，读取用户信息的 API 与读取头像的 API 是分开的，目的是考虑到头像信息过大，有些只需要用户基本信息而不需头像的需求则可避免较大的请求开销。

下图展示了系统的用户注册界面：



图 5-4 客户端用户注册页面

用户登录成功后即跳转到欢迎页，欢迎页面风格设计简约大气，整体以浅黄色为主，布局整体居中，从上至下依次是系统 LOGO、带阴影和彩色字的系统名称以及用户姓名

和重要事项参看按钮。系统保证后台（业务逻辑端）所在机器的时间是准确的，并提供了一个 `getCurrentTime` API 接口以提供当前准确时间的 Unix 时间戳，精确到秒，系统默认对客户端前端的时间不信任，组件挂载后首先会从服务器该接口读取当前 Unix 时间戳，并根据当前时间自动设置“早上好”、“晚上好”这样的问候语，然后读取用户信息，提取出用户名并展示。页面通过 `getImportantList` API 获取重要事项列表，并只渲染优先级为紧急的事项和标记为“未完成”的事项，事项清单具体内容参见下文。事项查看按钮上面显示有待渲染事项的个数，单击按钮会触发弹窗，显示事项列表，该弹窗是基于 Element Plus Dialog 组件开发的。下图展示了欢迎页的基本情况：



图 5-5 客户端欢迎页面

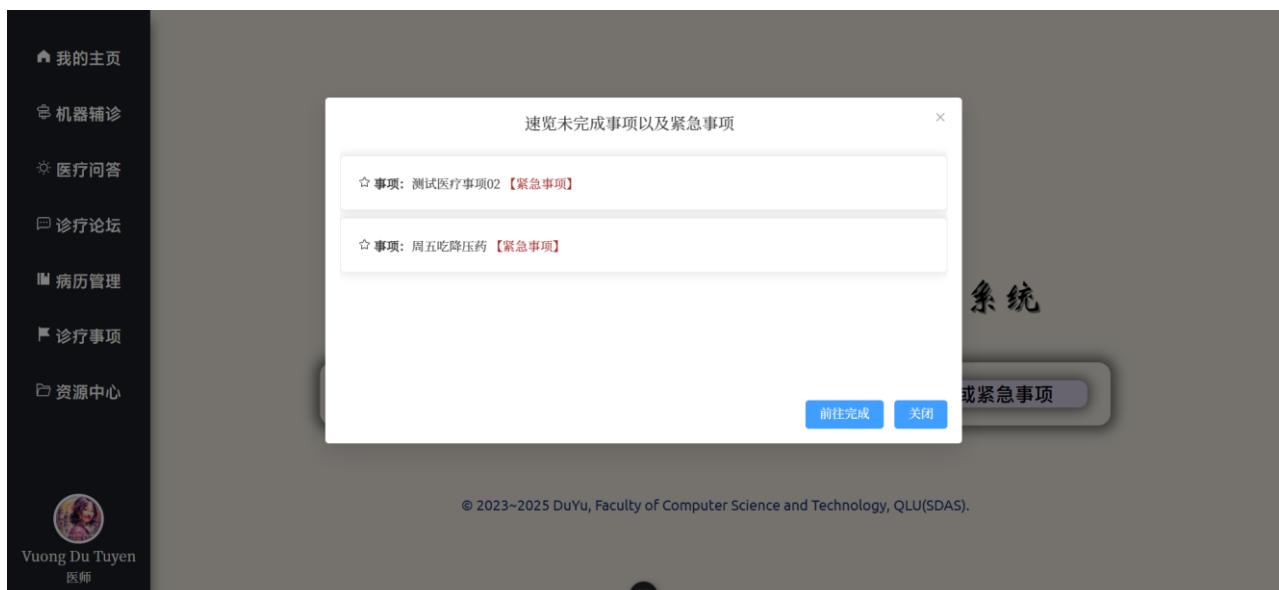


图 5-6 客户端欢迎页面 查看医疗事项

“我的主页”功能非常丰富，其仍然使用不断更替的好看的图片作为背景，更替原理与登录页面的相同。

首先是用户基本信息展示功能，调用后端 `getUserInfo` 和 `getUserAvatar` 接口读取用户的各项基本信息，如前文所述，`getUserAvatar` 拿到的是图像的 base64 字符串数据，可以将其直接赋给 `el-image` 组件的 `src` 属性显示图片，其中 `el-image` 组件是 Element Plus 提供的图片渲染组件，向其 `preview-src-list` 属性赋一个只有 `src` 属性值一个值的列表，还能实现单击组件放大查看。页面中间有个表格，UI 方面，对其 CSS 属性设置了特殊的 RGBA 值，使其透明度略降，可更好地与背景图片协调。该表格可显示用户名、用户类型、用户性别、用户年龄、用户来源、联系方式、用户描述、重要备注等信息，系统从后端读到的用户 JSON 信息包含若干 `userXXXXPermission` 变量，其中“XXXX”表示除了姓名和用户描述外的各字段名。这些变量是用户各项信息权限码，表示该项信息是否公开，并且若不公开，后端便会将该项信息内容设为 `null` 发往前端，此时前端会将该项目置为“<未公开>”字样，确保绝对安全。自己访问自己的主页，不论权限如何设置，都会显示全部信息；访问他人主页，则按权限限制显示。

`getUserAvatar` 接口和 `getUserAvatar` 接口均接收一个目标用户 `id`，以确定查询谁的信息。操作者的 `id` 是随 `token` 一同发往后端的，后端通过比对目标用户 `id` 和操作者 `id` 来确定信息查询者是否是本人，从而确定访问受限数据是否返回。

界面中，单击“修改信息”按钮出现弹窗，即可修改信息，前端调用 `modifyUserAvatar` 和 `modifyUserPassword` 接口来修改信息，只有访问本人主页时才有该按钮，若持有非本人登录信息的 `token` 强行发起请求，后端会因校验修改操作执行者身份不通过而拒绝执行。

页面上还有一个留言的功能，当访问自己主页时，显示“查看留言”按钮，否则显示“给 TA 留言”按钮，这一选择性渲染的实现（包括上文中提及的类似情况）均由 `v-if` 特性完成。单击按钮，窗体右侧会出现一个基于 Element Plus 的 `el-drawer`，这是一个“抽屉”组件，可以模拟抽屉“拉出”或“推进”面板。查看留言功能中，面板上包含“发给我的留言”和“我发的留言”两个选项卡，单击选项卡，会调用接口读取指定留言列表，并以 `v-for` 遍历显示，针对每个留言项还要再发请求读取用户名、类型、头像等详细信息，这部分的请求都是异步发起的，可以大幅度提高效率。此外，当光标落在留言项上时，会触发 `el-popover` 渲染，从而以小气泡框的形式显示用户简要信息，单击还能跳转至对应用户主页。对于发送的留言，自己有权删除，单击红色小垃圾桶符号，会

触发 el-popover 弹出确认框，单击确认可以发起删除请求，后台核实执行者与留言发布者是否为同一人，若相同，则执行删除操作，该“删除”并非真删除，而是将数据库中该项的状态字段改为“删除态”。“给 TA 留言”按钮单击后，会出现填写留言面板，填写留言后单击发送按钮，就会调用 API 发送留言。

页面得知当前目标用户 id 的方法是读取 GET 请求参数，打开主页时路径后拼接“?userId=xxx”，页面通过 this.\$route.query.userId 来读取用户 id，若没有请求参数，则默认使用当前登录用户，也就是展示自己的信息。

下图展示了个人主页的基本页面：

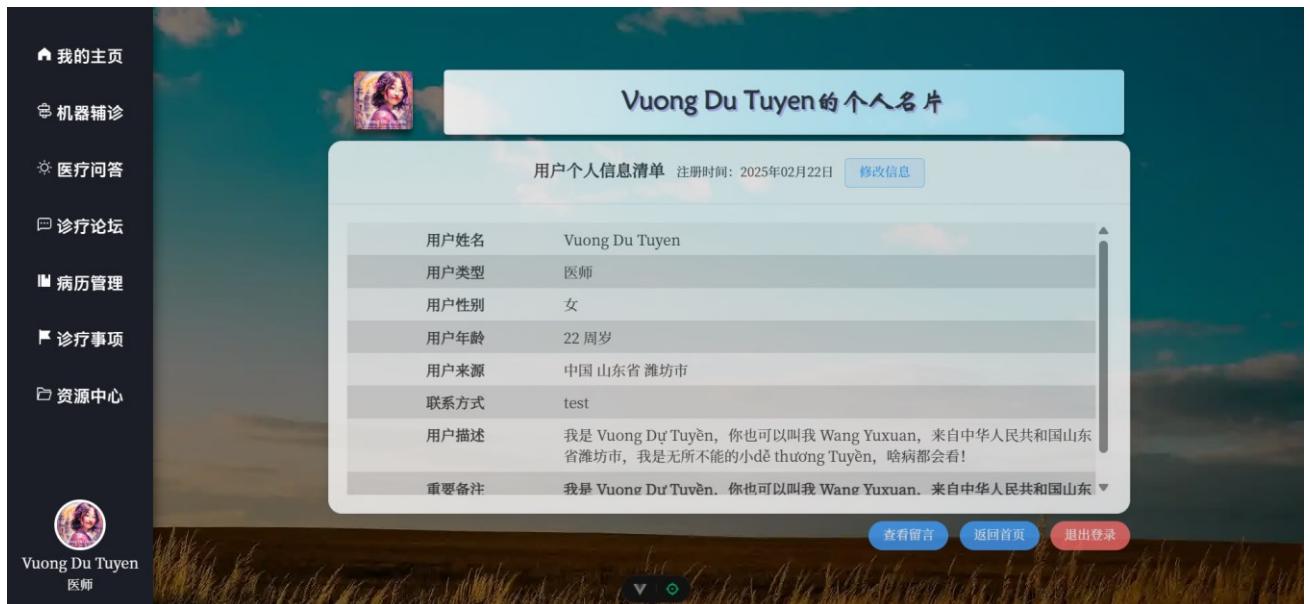


图 5-7 个人主页

下面展示了修改个人信息的页面：

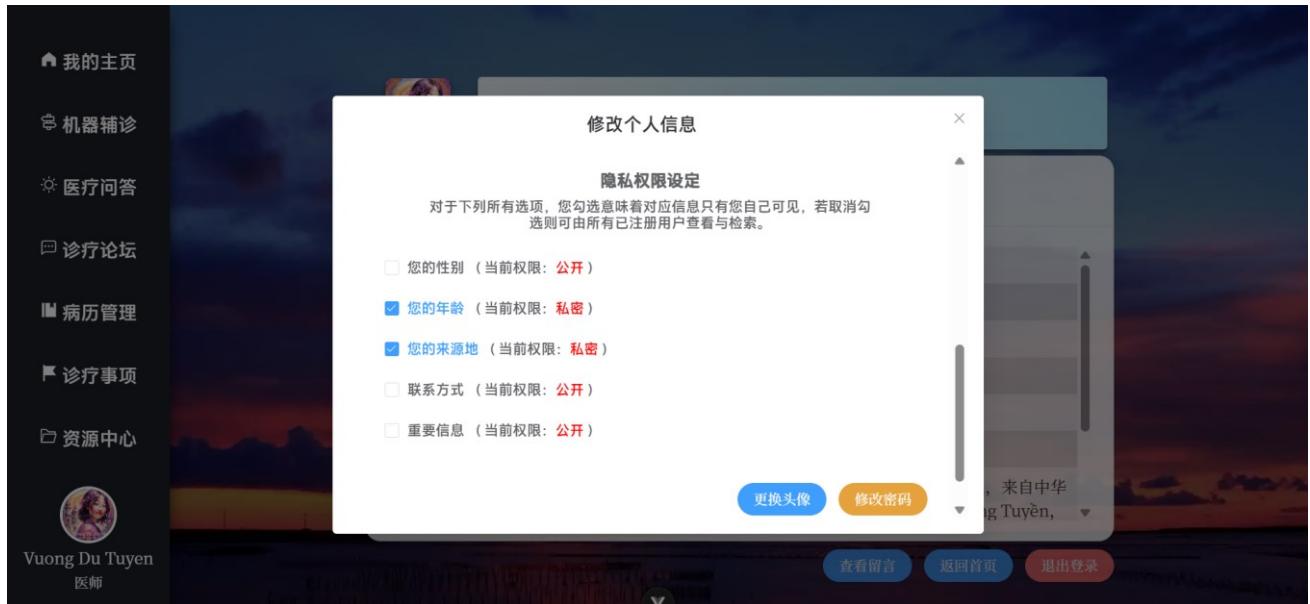


图 5-8 个人信息修改

下面展示了查看留言的界面，查看自己发送的留言以及发送留言的界面与其相似，不再作展示：

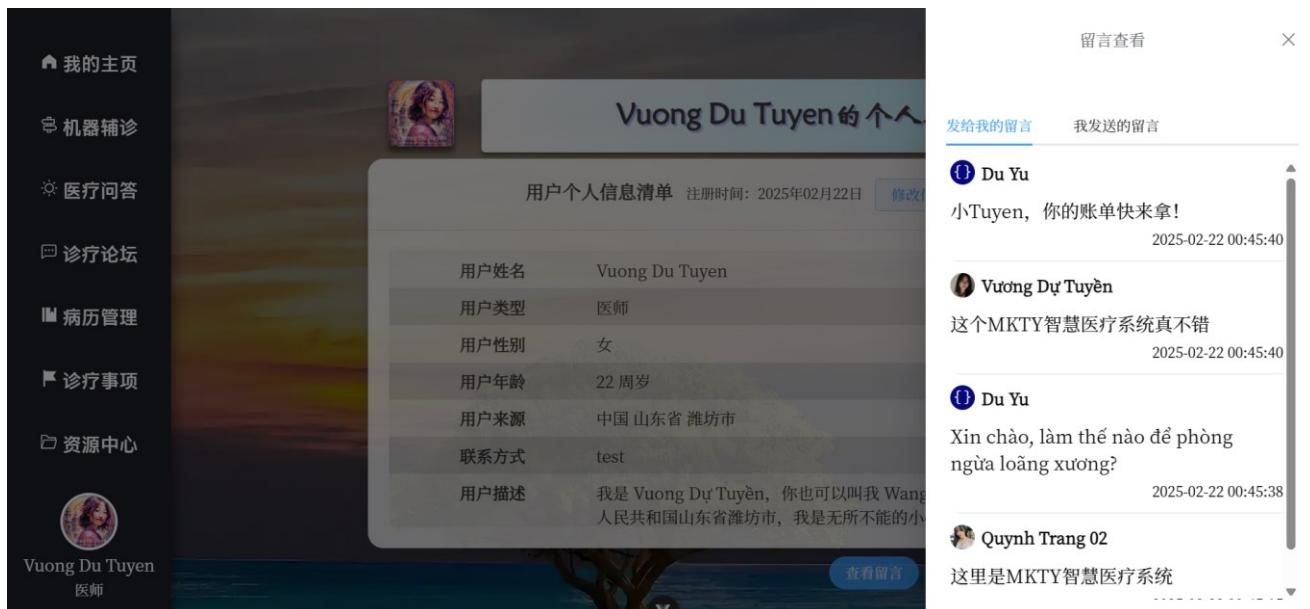


图 5-9 留言查看

5.3 多模态辅助诊断模块

首先介绍模块的视图层部分，组件首先导入了所需的依赖，包括 Element Plus 的图标组件、自定义 CSS 样式、API 函数和 ECharts 相关模块。核心功能实现在 getResult 方法中，该方法首先设置 loading 状态，该状态标志用于显示“处理中”的旋转图标，

然后收集用户输入的诊断文本和上传的图像，调用 `multimodalDiagnosisSubmitTask` API 提交任务，若提交成功则得到任务 GUID。提交之后创建定时器，周期性调用 `multimodalDiagnosisGetStatus` API 以轮询检查任务状态，每次轮询都会得到一个任务状态码，当读取到任务完成时，销毁定时器停止轮询，处理返回的诊断结果并更新界面展示到 ECharts 饼图，其中返回的处理结果包含用户原输入、对应英文翻译与相对正确概率。上述过程是通过 Promise 函数回调链来确保逻辑上的串行执行。处理图像上传时与用户头像的处理方式相同，使用 `FileReader` 将图像文件转换为 Base64 编码字符串进行传输和前端显示，不同的是为了提升检测准确度，默认不对图像作剪裁。诊断结果以 ECharts 彩色饼图形式展示各个诊断描述的概率分布，同时也提供表格形式的详细结果查看。

UI 方面使用 Element Plus 的布局组件构建了页面结构，包括顶部蓝色渐变色标题栏、主体内容区和结果展示对话框。主体内容区分为两部分：左侧显示待分析的诊断内容表，右侧是设置区域，包括文本输入、语言选择、图片上传控件和分析按钮等。用户单击右侧的选择文件按钮，浏览器打开文件选择器供用户选择图片；用户在文本框里输入医学文书，单击添加按钮，系统会将文本追加到文本列表中，由 Vue 的 `v-for` 在左侧列表中重绘，左侧列表还提供删除功能。结果展示使用对话框形式，包含饼图和详细的表格数据。

后台方面，系统为该功能模块提供了两个接口 `multimodalDiagnosisSubmitTask` 和 `multimodalDiagnosisGetStatus`。系统后台基于 pika 库编写了一个 `RPClient` 类，其封装了作为消息生产者连接 MQ、向 MQ 发布消息、接收消息的操作，具体是，调用 `BlockingConnection` 函数创建连接，以 `basic_publish` 函数发布消息，类中还维护一个字典类变量，存储任务 GUID 和任务结果的映射关系，当 MQ 消费者端计算完成后，即向业务逻辑端返回结果，存入字典，当前端通过 `multimodalDiagnosisGetStatus` 接口轮询请求任务结果时，根据 GUID 查找字典中的任务结果，若找到则说明任务已结束，返回结果，否则返回表示任务未结束的代码，为了释放内存，字典中的数据一旦被读取则要清除这一项。所发布消息的具体内容包含医学影像 base64 数据、文本列表、语言符号码，考虑到 MQ 双方均为 Python 语言环境，这些数据是以 Python 字典的序列化字符串发送的，智能服务端读到后采用 `ast` 安全反序列化这些数据结构。智能端若发现语言码代表汉语，则首先调用 `MarianMTModel` 神经机器翻译模型将文本列表中所有内容翻译为英文，然后再调用 `BioMedCLIP` 分别计算所传图像与各文本的余弦相似度，最后以 `Softmax`

函数将计算出的相似度向量求逻辑斯蒂归一化值，得到各语句“相对正确”的概率分布，连同英文翻译和原文一起返回给业务逻辑端。

下图展示了多模态辅诊模块的基本页面：



图 5-10 多模态辅诊基本页面

下面的截图展示了页面的饼图结果展示：

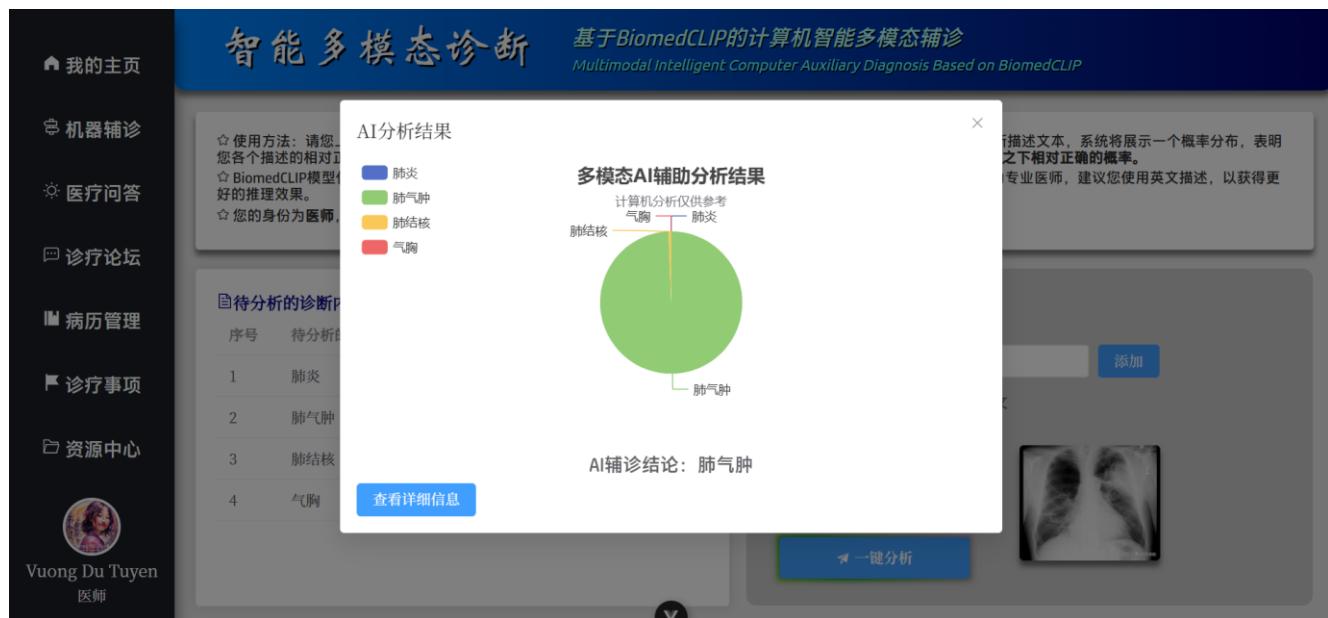


图 5-11 多模态辅诊饼图结果展示

单击“查看详细信息”按钮还可以查看对应英文翻译和概率值。



图 5-12 多模态辅诊表格结果展示

5.4 医疗问答模块

本小节主要讲述明康慧医系统的医疗问答功能模块，该模块分为两个部分——“明康慧医智慧问答”和“明康慧医智能体深度分析”，本节首先介绍关于明康慧医大模型训练的情况。

明康慧医大模型 (MKTY-3B-Chat) 参数量 3.09B，量化精度 BF16，经过充分调研，结合本研究硬件等客观条件，权衡利弊后决定在本研究中使用 Qwen2.5-3B-Instruct 大模型基于医学领域文本微调。微调时采用的框架为 LLaMA Factory，使用的算法为 LoRA（低秩矩阵适应算法），矩阵的秩设置为 8，LoRA 相较全参微调可显著降低计算量和显存。学校及指导教师为本次研究提供的算力硬件环境为 4 Intel Xeon CPU 核心 4 GB RAM 服务器，搭载 2 张 NVIDIA GeForce RTX 2080 Ti 显卡，每张显存为 11GB，共计 22GB。而底模 Qwen2.5-3B-Instruct 的参数规模也为 3B 左右，若采用全参微调，增量预训练阶段估计会使用 24GB 显存，这样估计的依据为，2080 Ti 显卡硬件上不支持混合精度训练，故训练阶段是必须使用 32 位浮点数精度，否则会造成数值不稳定，出现“nan”，此外反向传播时每个张量的梯度也要占用体量等于其参数的显存， $3B \times 4$ 字节 $\times 2 = 22.35GB$ ，外加 PyTorch 固有显存占用和当前训练数据的占用，则一共会消耗约 24GB 显存，上述简单计算还没有考虑训练过程中的中间激活值，而硬件条件为最多 22GB 显存，故本人决定使用 LoRA 算法降低显存消耗和加快训练速度，并适度减小 batch size。

微调算法采用的是增量预训练 (Pretrain) 与指令监督微调 (SFT) 两种，并分四个微调步骤进行，具体来说就是一轮增量预训练+一轮指令监督交替进行两次，这样做是

考虑到底座模型规模不大，吸收知识的能力跟巨大规模模型相比稍逊一筹，若只进行一轮微调，那么经过 SFT 后大模型可能会遗忘掉其先在增量训练阶段学习到的知识，执行两轮微调可避免大模型的灾难性遗忘。

训练数据方面：语料数据包含为生物领域广泛文本、医学诊断与问答、医学考试选择题以及自我意识等。在本项目中，MKTY 大模型的使用场景是医疗问答、大模型讨论、总结诊疗计划、根据病历诊断和推荐药物，本人针对这四条用途准备了数据集，医学生物广泛知识文本用以在增量预训练阶段增加大模型的医疗专业知识，医学问答数据集用于指令监督微调，增强大模型回答问题的能力，医学诊断用于增强大模型诊断病历的能力，使用医学考试选择题的目的是告诉模型一个问题及其回答正确答案的回答模式，在大模型讨论机制功能中，不论是独立智能体角色还是主持人角色，都应该针对某问题结合已有的答案做出自己的判断，而教会大模型做医考选择题即可达到训练大模型按这种模式来回答问题的目的。另外“自我意识”是指通过指令监督，使模型得知自己是谁，由谁开发等，这部分数据由我本人设定。

大模型所用训练数据总量约为 2.88GB（解压后约为 6.79GB），是为从全网各网站平台以及本人个人关系获得的，所有训练数据均为开源的，并且是在不违反开源协议的合法情况下使用的，由于数据来源数量非常大且难以统计，以下仅列出了主要的数据来源网址，所有数据用于训练前都做过二次清洗和规整格式等预处理：

1. <https://huggingface.co/datasets/Flmc/DISC-Med-SFT/tree/main>
2. <https://huggingface.co/datasets/Bolin97/MedicalQA/tree/main>
3. <https://huggingface.co/datasets/tyang816/MedChatZH/tree/main>
4. <https://huggingface.co/datasets/TigerResearch/MedCT/tree/main>
5. https://huggingface.co/datasets/hajhouj/med_qa/tree/main
6. https://huggingface.co/datasets/ChenWeiLi/Medtext_zhtw
7. 其它数据集（从略）

感谢上述开源数据集的提供者为本研究提供的帮助。另外，下方的损失值图展示了大模型在增量训练微调过程中交叉熵损失下降的过程。增量训练时设置了 3 个多 epoch，每个 epoch 需遍历训练 6000 个批次的数据，共训练 20000 个批次，下图可以直观的看出损失。

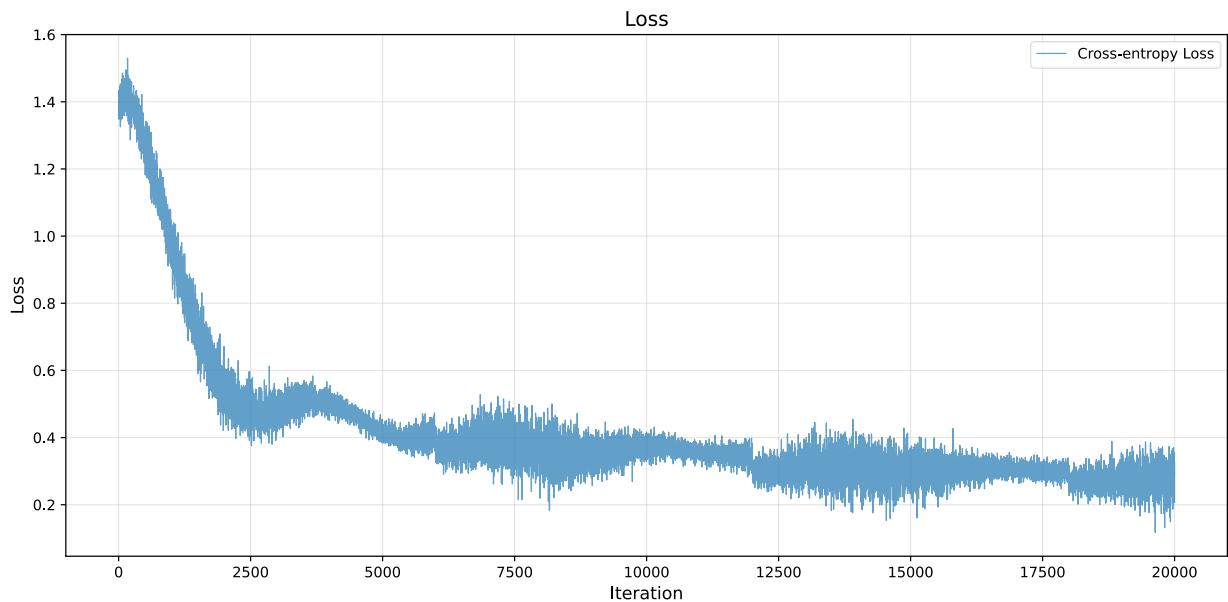


图 5-13 MKTY 大模型训练损失下降图

为了秉持学术研究的开放性精神和合作精神，待本次毕业设计项目答辩结束后，明康慧医大模型权重将开源，开源平台为 `huggingface`，预计的开源地址为：<https://huggingface.co/Duyu/MKTY-3B-Chat>。欢迎各位老师和同学下载使用并提出意见。

接下来讲解 MKTY 智慧问答部分的前端设计：这部分前端代码继承了项目全局技术栈，采用 `Vue` 开发，并引入了多个 `Element Plus` 的图标组件如 `Promotion`、`Avatar`、`Delete` 等，用于界面的图标美化展示。本页面的数据非常复杂，简而言之，页面定义了多个关键数据项：`PsyChat_userAvatar` 存储用户头像，`PsyChat_Context` 保存当前输入内容，`PsyChat_Generating` 标记是否正在生成回答，`PsyChat_SessionId` 记录当前会话 ID，`PsyChat_ChatArr` 存储聊天记录数组，`PsyChat_LlmSessionList` 保存历史会话列表等。

核心功能实现主要包括：聊天功能、会话管理和导出功能。聊天功能通过 `PsyChat_Send` 方法实现，该方法会将用户输入添加到聊天记录，调用 `llmInferenceSubmitTask` API 向服务器发送聊天记录和用户输入，然后通过 `setInterval` 创建定时器，定时调用 `llmInferenceGetStatus` API 轮询检查任务状态，最终获取大模型回答并渲染展示。请各位读者回看参考 5.3 节，上述过程与 5.3 节中讲到的 BioMedCLIP 模型调用的过程如出一辙，原理基本相同，在本次项目中，所有业务逻辑端调用智能端的方法皆与此类似，下文若还有相关功能，将略提及，不再详细赘述。关于 RAG 的功能，请参见下文“资源中心”模块。

同时，由于 MKTY 大模型的输出默认是 `Markdown` 格式的，代码为此引入了 `marked`

库用于 Markdown 渲染, DOMPurify 用于防止 XSS 攻击, highlight.js 用于代码高亮处理。具体流程是, marked 库将 Markdown 解析为 HTML, 而 HTML 可能包含恶意 JS 脚本, 不一定是安全的, 故用 DOMPurify 库清除 HTML 中可执行脚本或任何潜在的隐患, 安全的 HTML 字符串便可以通过标签的 v-html 属性, 利用 Vue 挂载, 从而渲染到界面上。

关于会话管理的功能。会话管理功能包括新建会话、加载历史会话列表、加载特定会话和删除会话等。新会话创建会重置 SessionId 为-1 并初始化聊天记录数组; 加载会话会根据会话 id, 从服务器获取历史对话并更新到界面; 删除会话则会调用 API 删除指定会话并刷新列表。若在页面中新建了会话或新打开页面, 那么 PsyChat_SessionId 的值为-1, 后台会将以此 id 值的会话新保存到数据库, 若 id 值不为-1, 那么说明前端是加载历史会话记录后又继续对话的, 后台会根据 id 值更新数据库中的记录。所有保存的会话均为 HTML 字符串, 可直接安全挂载。

页面还实现了聊天记录导出功能, 该功能通过调用 exportChatToPDF API 获取 PDF 文件流, 然后前端读取到文件流后创建虚拟[标签](#)并虚拟单击链接, 触发浏览器下载。后台先使用 HTML 代码设计了一种页面逻辑与样式, 然后从数据库读取到对话内容后会像前端一样在这个页面上渲染, 最后利用 weasyprint 库的 HTML 类将 HTML 导出为 PDF。weasyprint 库依赖 Python 外部环境, 在不同操作系统上要安装不同软件, 例如 Windows 系统中依赖 gtk3-runtime, Linux 等的暂不描述。

在 UI 设计上, 页面实现了响应式设计, CSS 代码里的媒体查询能在不同屏幕尺寸下调整样式, 这样使得在移动设备上也有良好的显示效果。此外页面被分成了顶部标题栏、功能按钮区、聊天主区域和输入区域一共 4 个区域, 聊天记录区使用了不同样式区分用户和 AI 消息, 用户消息右对齐, AI 消息左对齐, 就像 QQ 或 微信里那样。代码中还优化了许多的细节, 比如聊天消息发送后, 页面会自动滚动到最底部, 这个过程添加了防抖动处理, 生成回答时还有加载动画。下图展示了智慧问答功能的 UI 界面:

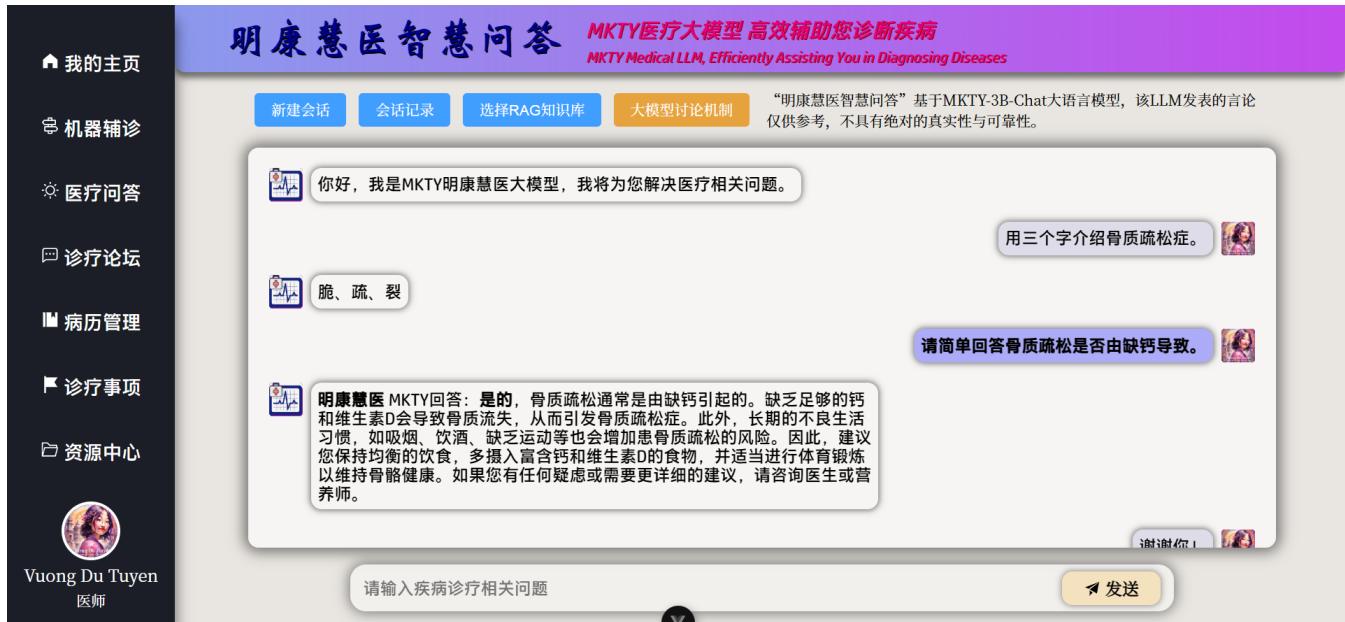


图 5-14 MKTY 智慧问答模块界面

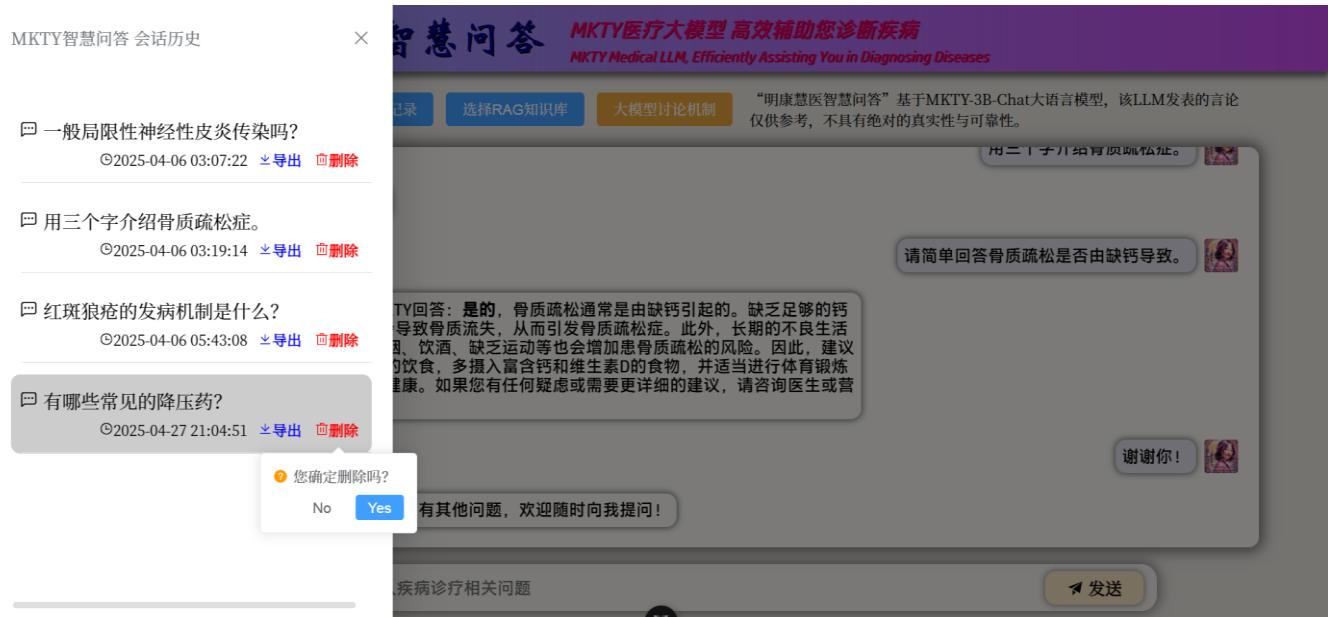


图 5-15 MKTY 智慧问答模块会话历史

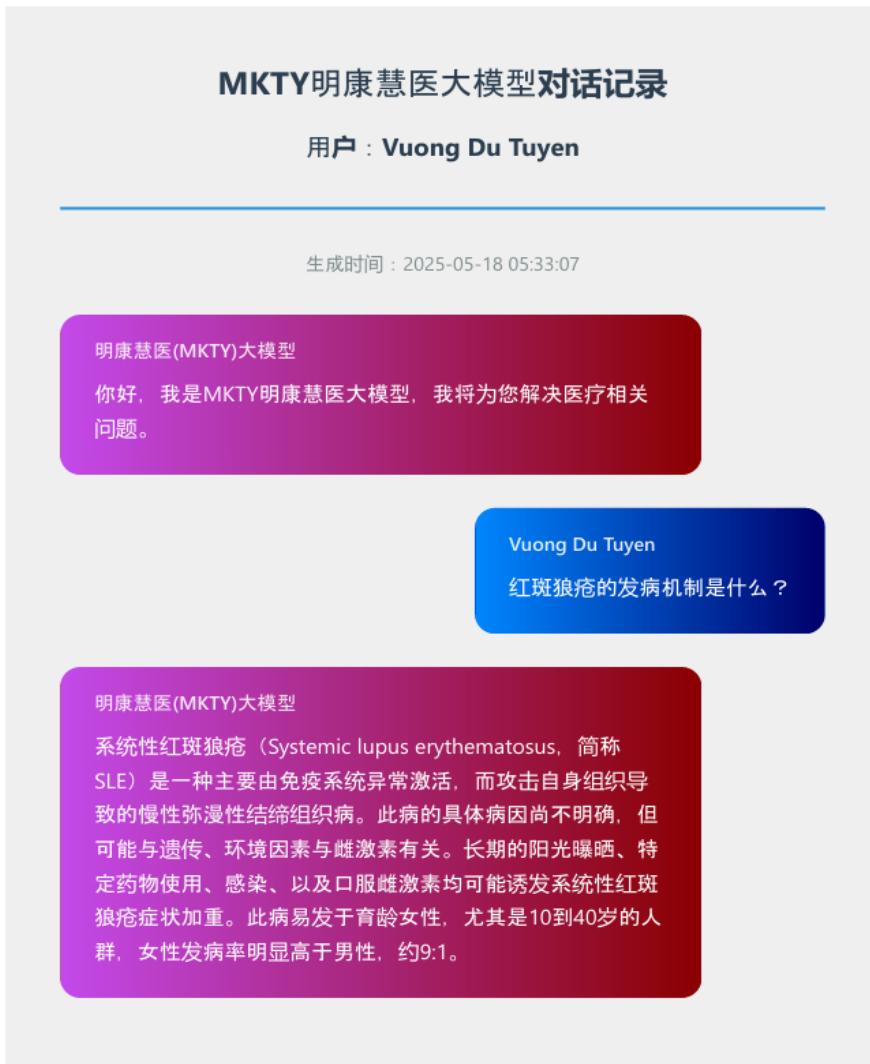


图 5-16 MKTY 智慧问答模块导出的 PDF 截图

接下来讲述明康慧医“智能体深度分析”功能：UI 设计上与“智慧问答”模块差别不大，特别说一下，前端使用了 el-steps 组件显示讨论进度，使用 el-collapse 组件实时展示讨论结果。另外，系统还添加了超参数设置弹窗对话框，通过滚动条设置各项超参数。

下面详细介绍基于大模型讨论机制的智能体深度分析功能：本功能有智能体个数、讨论回合数与判敛阈值三个超参数，它们可通过滑动滚动条来设置。完全相同的若干大模型（MKTЫ-3B-Chat）在会话上下文不同时不认为是同一个智能体。第一轮讨论过程是，系统通过设置多个上下文数组模拟多个智能体，让每个智能体分别回答待深入研究的问题，然后由没有会话上文的“主持人”智能体总结各方发言。以后每轮讨论，都将上轮主持人的总结和原问题拼接合并，并由各智能体基于自己的会话上下文再分别回答。

合并后的 prompt，最后主持人总结，周而复始，直至达到最大讨论轮次数。

然后是“判敛”的过程：用 BigBird 将最后一轮讨论各方的输出计算句子嵌入向量，然后计算各向量两两之差的平均值，以此反应各方达成共识的程度，即讨论语义收敛程度，这个数值可供人类用户作参考。对于大模型调用，后台只暴露出会话任务提交和轮询两个接口，“讨论机制”的实现逻辑实际上位于前端调用这两接口实现的，接口原理参见本节前半部分，此处从略。

下图展示了“智能体深度分析”基本页面效果和运行效果：



图 5-17 MKTY 智能体深度分析模块截图



图 5-18 MKTY 智能体深度分析模块运行效果



图 5-19 MKTY 智能体深度分析模块 超参数设置面板

5.5 医学与诊疗论坛平台模块

该模块共有论坛概览与论坛内容两个页面，首先介绍论坛概览。

从技术实现角度来看，组件采用了 Vue 的单文件组件结构，导入了 Element Plus 的多个图标组件和自定义 API 函数，其定义了多个状态变量，包括用户 ID、论坛数据列表、搜索框文字内容和分页信息等。当页面打开时，组件会在 mounted 钩子中调用 fo_loadForumList 方法加载论坛列表数据，这方法是组件的核心功能之一，它通过调用 getForumList API 获取系统中存续的所有论坛，并对每个论坛项进行处理以显示，包括转换时间格式、处理论坛类型和权限的显示文本，以及根据搜索框内容进行过滤。此外，该方法还会为每个论坛项加载创建者的信息和头像。

getForumList 函数的作用不复杂，前端传入论坛类型限制代码和论坛权限限制代码，该函数便在数据库中查询符合筛选条件并仍存在的论坛，将论坛信息返回给前端。同时，论坛展示列表中有创建者姓名及头像等信息，前端遍历 getForumList 返回的论坛列表时是并行发起查询用户信息请求的，这样可以提高执行效率。但是系统中有一个做的不是太好的地方，基于关键字检索的论坛筛选是在前端完成的，关键字完全为论坛名称的子字符串则认为是“匹配”，但如果论坛总数庞大，后端会向前端传输一个巨大的列表，这样做会消耗大量资源。

本模块中，论坛包含论坛名称、论坛创建者、论坛编号、隶属类别、创建时间和论

坛权限六大属性，其中论坛编号是论坛的唯一 id，论坛类别包含医学知识论坛和病情讨论区两种，论坛权限包含仅限患者访问、仅限医师访问和无限制三种类型。用户可以设置名称、权限和类型创建论坛，在页面显示的论坛列表中，自己创建的论坛有权修改，页面上会在自己创建的论坛旁显示修改论坛按钮，可修改的内容是论坛类型和删除论坛。在删除论坛或其它敏感操作前会通过弹窗确认用户意图，防止误触。修改论坛类型时后台会比较当前用户 ID 和论坛创建者 ID 来进一步控制权限。实现上述内容仅需后端实现数据库的 CRUD 操作，并由前端请求即可，不再详细介绍。

总结一下，页面实现了多个交互功能：1.通过搜索框和下拉选择器筛选论坛；2.点击论坛名称导航到论坛内部页面；3.论坛创建者可以修改论坛类型或删除论坛；4.用户可以查看论坛创建者的详细信息。这些功能均通过 Vue 的事件处理机制和方法调用实现。

在 UI 设计方面，整个页面使用了 Element Plus 的各种组件，如 el-input、el-select、el-button、el-pagination 等构建界面。页面通过 CSS 媒体查询实现了响应式设计，在小屏幕设备上调整显示效果。另外，组件还实现了分页功能，通过计算当前页的数据切片来显示适量的论坛项以免拥挤。

下图展示了论坛概览页面：

The screenshot shows the homepage of the 'MKTY Exclusive Medical and Treatment Forum Platform'. The header includes the platform name in Chinese and English, and a tagline: 'Physicians Can Discuss Academics, Patients Can Share Medical Conditions'. On the left is a sidebar with icons and labels for: '我的主页' (My Home), '机器辅诊' (Machine Assisted Diagnosis), '医疗问答' (Medical Q&A), '诊疗论坛' (Treatment Forum), '病历管理' (Medical Record Management), '诊疗事项' (Treatment Matters), '资源中心' (Resource Center). Below these is a profile section for 'Vuong Du Tuyen 医师' (Doctor Vuong Du Tuyen) with a small circular profile picture. The main content area has a title '论坛概览' (Forum Overview) and a sub-section '创建论坛' (Create Forum). It features a search bar with placeholder '搜索您参与的论坛' (Search forums you participate in) and dropdown filters for '请限定论坛类别' (Limit forum category) and '请限定论坛权限' (Limit forum permission). Below this are three forum entries, each with a thumbnail, the forum name, the creator's name, and a '修改论坛' (Edit Forum) button. The forums listed are:

- 论坛名称: 脊质疏松症讨论区
论坛创建者: Du Yu
论坛编号: 2
隶属类别: 病情讨论区
创建时间: 1970年01月01日 08时00分00秒
论坛权限: 不限人员
- 论坛名称: 近视防控论坛
论坛创建者: Vuong Du Tuyen
论坛编号: 3
隶属类别: 病情讨论区
创建时间: 2025年04月16日 07时22分43秒
论坛权限: 仅医师参与
- 论坛名称: 老年人保健综合
论坛创建者: Vuong Du Tuyen
论坛编号: 4
隶属类别: 医学知识论坛
创建时间: 2025年04月16日 07时22分43秒
论坛权限: 仅病患参与

At the bottom of the main content area is a pagination bar with page numbers 1, 2, and >.

图 5-20 诊疗论坛 论坛概览页面

下面是修改论坛弹窗：

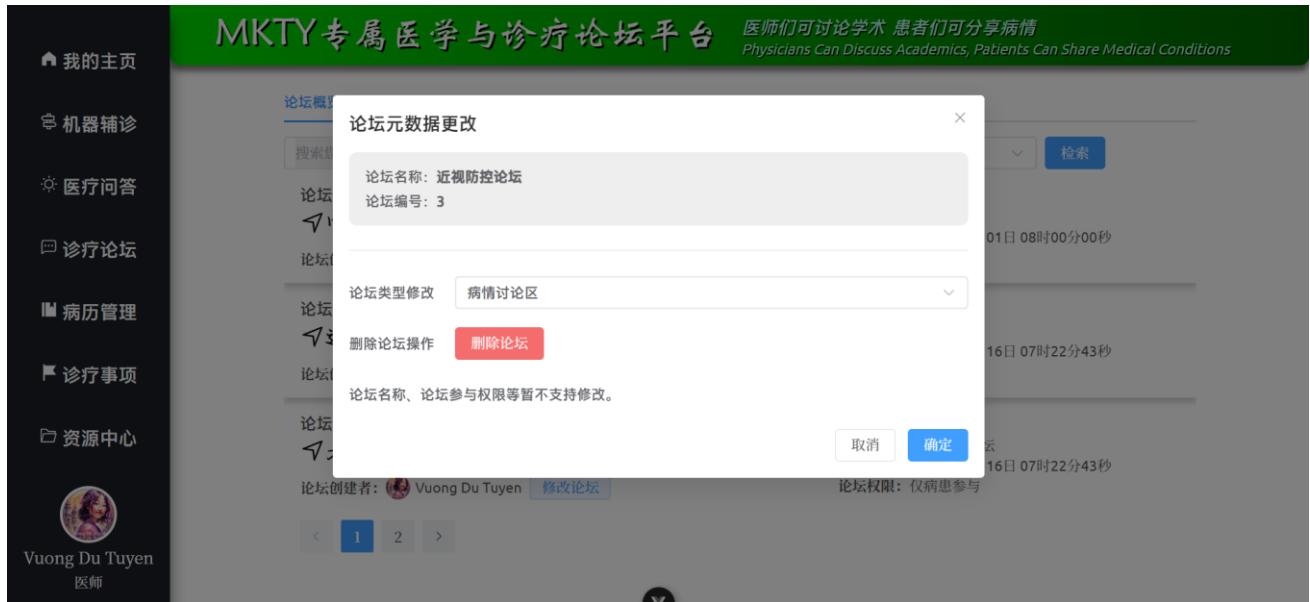


图 5-21 诊疗论坛 论坛数据更改弹窗对话框

下面是论坛的创建页面：



图 5-22 诊疗论坛 论坛创建页面

接下来介绍论坛内容页面，该页面组件负责显示特定论坛的详细信息、帖子列表以及允许用户发布新帖子和与其他用户互动。组件实现了多个方法来处理各种功能：`fi_loadPage` 方法用来加载论坛信息和帖子列表，它首先调用 `getForumInfo` API 获取论坛基本信息，然后调用 `getPostList` 获取帖子 ID 列表，接着遍历这些 ID，调用 `getPostContent` 获取每个帖子的详细内容，并通过 `getUserInfo` 和 `getUserAvatar` 获取发帖用户的信息和

头像，这些数据被整合到 ForumInner_Arr 数组中，用于在界面上显示。具体说来，每条帖子的完全显示需要 4 个串行的请求，首先读取论坛基本信息，比如论坛 id、名称等，然后读取论坛帖子 id 列表，该列表只包含若干 id 号，没有论坛的任何具体信息，随后遍历每个 id，请求帖子具体信息，包含帖子文字和图片、赞数、发布时间和发布用户 id 等，由于使用了 await 使请求函数串行执行，故此时已可以渲染基本信息了，减少用户等待时间。最后通过发布者 id，请求用户姓名和头像等信息。上文已讲，论坛有权限这一属性，checkPermission 方法用于检查用户是否有权限访问当前论坛，根据论坛权限设置和用户类型（患者、医师或其他人员）来判断。如果用户没有权限，会显示错误消息并立即重定向到论坛概览页面以阻止访问。

论坛的内容是以 JSON 存储于数据库的，其包含 “content” 和 “images” 两个键，前者存储帖子文本，后者存储帖子图像 GUID 列表，这个列表本质上也是一个字符串，是以 “\$^” 符号间隔开每个 GUID，图片是以 webp 文件的形式保存于服务器的。上传文件时，也使用 JSON，并且有相同的键，不同的是，images 中存储的是图片 base64 字符串，也是以 “\$^” 符号间隔开，限制最多传 3 张图片。用户每选择一张图片，组件就以 Canvas API 压制图片以限制尺寸，并将图像 base64 暂存数组，若用户选择了大于 1 张的图片，文本框上部就显示图片栏，用 el-image 组件渲染每张图片，利用 el-image 提供的功能，用户可放大预览图片，包括帖子里的图片，还可以轮播。图片栏中还有清空图片栏的按钮，原理是清空 base64 暂存数组。

用户还可以在系统上回复帖子（本质上就是添加“@用户名”字符串）和点赞帖子，对于自己发布的帖子下会显示删除按钮，还可以进行删除，这些操作都由后台基本操作数据库的 API 来实现。

在就 UI 界面而言，页面使用了嵌套的 div 结构来组织界面元素。主要区域包括论坛头部信息区域，显示论坛名称、帖子数量、属性和创建时间等论坛元信息；帖子列表区域，使用 v-for 指令遍历 ForumInner_Arr 数组，为每个帖子创建一个包含用户信息、帖子内容、图片、点赞数和操作按钮的卡片式界面；发帖区域，包含文本输入框、添加图片按钮和发布按钮，允许用户创建新帖子。样式上定义了响应式布局，使界面能够适应不同屏幕尺寸。

下方两张图展示了论坛内容这部分的 UI 效果：

The screenshot shows a forum post titled "近视防控论坛" (Myopia Prevention Forum) with 7 posts. The first post by user "Нодикний" (患者 | 越南社會主義共和國 富安省 緩和市) says: "快来看一看Quỳnh Trang! 摘掉眼镜就是好看!" (Come and look at Quỳnh Trang! It's good to take off your glasses!). The second post by "Vuong Du Tuyen" (医师 | 中国 山东省 潍坊市) says: "难道常挤眼就能防近视吗? 不应该少看手机吗?" (Is it true that squeezing your eyes often can prevent myopia? Shouldn't we reduce screen time?). There are buttons for "添加图片" (Add Image), "发布" (Post), and "删除" (Delete).

图 5-23 论坛内容界面

The screenshot shows a forum post titled "脱发医学问题研讨" (Androgenetic Alopecia Medical Problem Research) with 2 posts. The first post by user "Quynh Trang 02" (患者 | Việt Nam) says: "回复: @Нодикний: 嗯嗯确实! 你看我就痊愈了。" (Reply: @Нодикний: Yeah, I'm really healed! Look at me!). The second post by "Нодикний" (患者 | 越南社會主義共和國 富安省 緩和市) says: "脱发中一般脂溢性脱发比较容易治疗。" (In general, seborrheic alopecia is easier to treat). There are buttons for "添加图片" (Add Image), "发布" (Post), and "删除" (Delete).

图 5-24 论坛内容界面 (II)

5.6 诊疗事项清单管理模块

该模块实现了"明康慧医 MKTY"智慧医疗系统的重要事项清单管理的功能。该功能允许用户添加、查看、完成和删除诊疗相关事项，还可以调用明康慧医大模型智能分析事项计划和邮件联系医师。

页面实现了多个核心功能，首先是事项管理功能，通过 getImportantList API 获取事项列表，并支持添加、删除和标记完成操作，单击添加事项按钮，界面右侧会出现抽屉

弹窗，用户可以填写表单创建事项，删除和标记完成则在事件列表中完成。事项可分为一次性事项、周期性事项和无时间要求三种类型，其中一次性事项存储一个精确的时间范围，周期性事项存储待办事件的星期，无时间要求事项不存储任何时间。组件通过 updateCurrentTime 函数每半分钟轮询更新当前时间，并根据事项的时间属性计算时间状态（未到时间、已到时间、已超时）。用户单击事项信息文字，会出现弹窗，显示事项的详细信息。事项展示逻辑上，页面将事项按优先级和时间状态排序，高优先级和紧急事项会更加醒目，每个事项显示其内容、完成情况、时间状态、事项类型和优先级等信息，并提供标记完成和删除操作按钮。

该模块支持 AI 智能分析事项清单的安排合理性，前端通过 llmInferenceSubmitTask 和 llmInferenceGetStatus API 调用 MKTY 大语言模型，调用原理同 5.4 节前文所述，完全相同。系统将所有事项组织为 Markdown 的多级列表格式，然后拼接上“请分析合理性”的 prompt 而构建提示词，交由 MKTY 大模型，分析用户的医疗事项清单，提供专业的建议。分析结果使用 marked 库解析 Markdown 格式并通过 DOMPurify 进行安全处理，随后通过弹窗展示分析结果。

模块还支持医患沟通的功能，该功能的实现依靠电子邮件的发送。在前端，本人手写了一个 Markdown 编辑器，编辑器上方有按钮，支持快速输入 Markdown 语法，编辑器下方左侧是编辑区，右侧可以实时渲染 Markdown 文本为 HTML，这个组件核心也是靠 marked 库和 DOMPurify 库的。用户可填写收件邮箱、主题以及编辑 Markdown 邮件正文，单击发送按钮，邮件会以 HTML 格式发往指定邮箱。这个过程后端使用了 Python 的 email 库和 smtplib 库分别用于组织邮件内容和执行发送。

界面设计方面，页面组件使用了 el-dialog、el-drawer 及 el-form 等大量的 Element Plus 组件构建交互界面。样式部分使用了 scoped CSS 并实现了响应式布局，包括在小屏幕设备上的适配。具体界面效果请看以下几张图片：

图 5-25 诊疗事项清单管理

图 5-26 诊疗事项清单医患互联功能

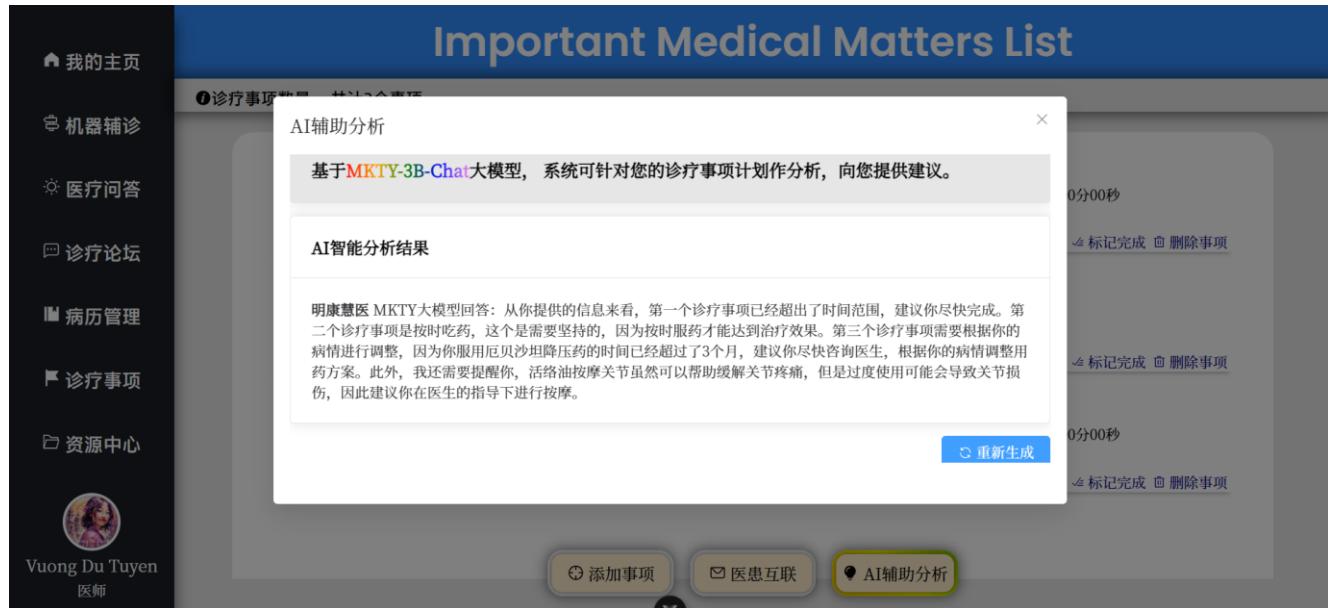


图 5-27 诊疗事项清单 AI 分析功能



图 5-28 用户收到邮件的样式

5.7 病历诊断与资源中心

因篇幅原因和时间关系，并且考虑到“病历诊断”和资源中心的前端设计与后端 CRUD 操作与前文所述功能所用技术栈高度形似，故本人决定本文中不再详细描述此两

者的功能，而是以简明扼要的语言着重讲解基于上传资源的 RAG 的实现以及可融合语言的时间序列预测模型的建立。

以下图片展示了系统的病历管理功能界面：



图 5-29 病历管理界面

以下图片展示了系统的资源中心界面：

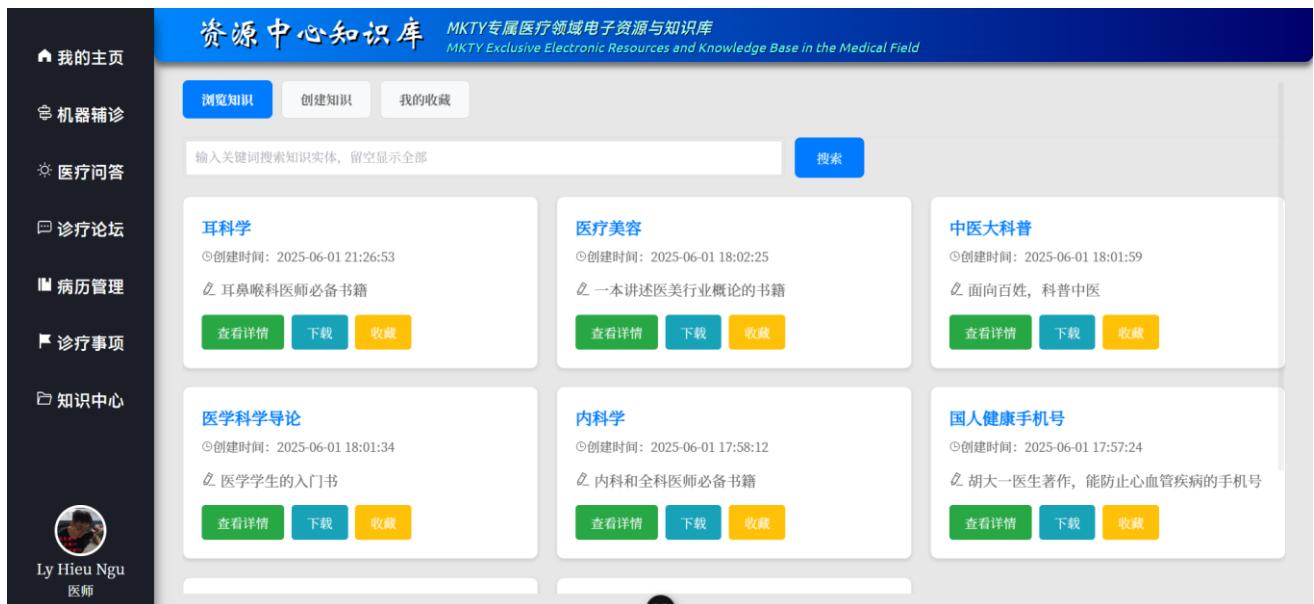


图 5-30 系统资源中心功能界面

以下图片展示了系统的资源中心界面：



图 5-31 资源中心功能 查看详情界面

关于 RAG 的实现：用户在资源中心模块上传任何文件后，后端都会为其分配一个资源 GUID 并建立以其为名建立一个目录，随后系统利用 `textract` 库提取 doc、ppt、pdf 等这类常见文件中的纯文本，为避免系统被攻击的风险，不支持上传压缩包及解压解析。提取文本后将对其分割切片，系统将整个文件的纯文本看作总数据集，并以此计算每个切片的 TF-IDF 特征，随后系统将计算后的模型、各文本切片以及原文件存储建立的目录，至此文件上传与解析的工作结束。当用户使用 MKTY 大模型时选定某资源，那么系统会先读取该资源对应的 TF-IDF 模型，并用它计算用户输入 `prompt` 的 TF-IDF 特征，然后计算该特征与各切片特征的余弦相似度，相似度 top-k ($k=5$) 的切片会返回前端，用以拼接在用户 `prompt` 前，从而增强 LLM 输出。

目前基于深度学习方法的各领域时间序列预测问题所使用最多的算法是 LSTM 或 GRU，直至去年（2024 年）也才有学者受 NLP 技术的启发提出基于 Transformer^[2]的时序预测模型，但这些方法都没有考虑到时间序列与多模态相结合。本次研究中，本人基于 GRU，尝试性地设计了一种基于医学文书的医疗时间序列预测模型，模型原理：主要使用门控循环单元进行初步的时间序列预测，而后通过 FFT 计算历史时间序列的频域，将频域中各频率序数对应的振幅向量与相位向量拼接得到频域特征，随后用 BigBird 提取医学文本描述的句子嵌入，利用交叉注意力机制计算出频域联合特征向量与该句子嵌入的分数矩阵，从而得出加权频域联合特征。将此特征向量拆解并求逆 FFT 可得到一个差值时序数据，与此同时将求逆 FFT 前的频域数据通过一个线性层，求得一个阈值向量，

利用门控的思想将这个阈值向量与求得的差值时序数据相乘，再加到基础 GRU 输出的结果上，作为模型最终的输出。这样设计的思想在于，时间序列的频域反映了序列的整体情况，而不像时域那样局限于局部时间，计算文本特征与序列频域特征的交叉注意力因而有意义，比如以心电图举例，文本描述“心跳加速”，那么从频域角度看，这句话代表的是整个心电图波形更高频部分的振幅增大，这很容易通过交叉注意力向某高频部分的振幅加权权重增大来反映出来，而时序数据理论上无法体现。

以下图片展示了模型的结构：

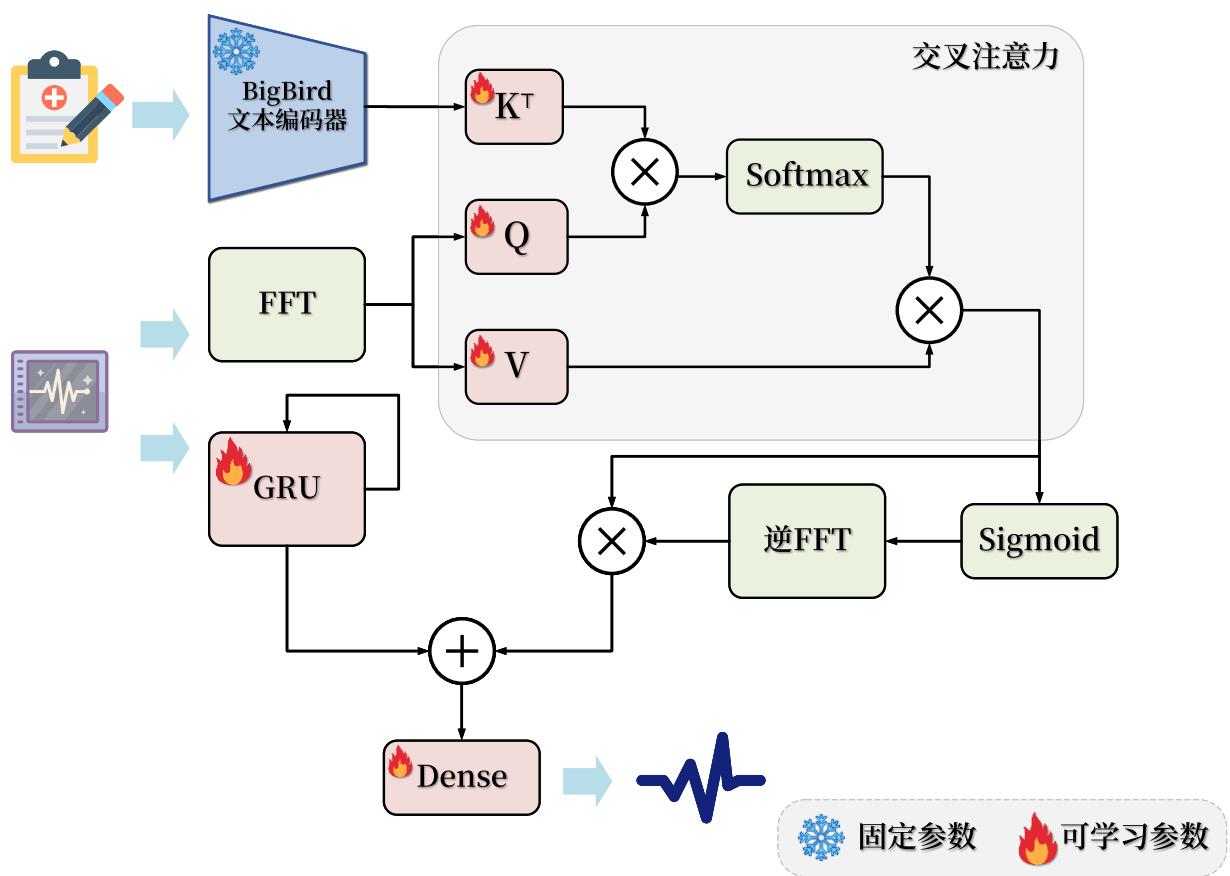


图 5-32 MKTY 医学时间序列预测模型结构图

5.8 后台管理系统

本项目的后台管理系统（后管端）逻辑独立于客户端系统，拥有独立的后台和 CSR 前端，但是两者所使用的技术栈是完全相同的。后管端直接对接数据库，全权管理系统的数据信息。此处不过多叙述，请看下列各图：



图 5-33 后台管理系统 登录界面

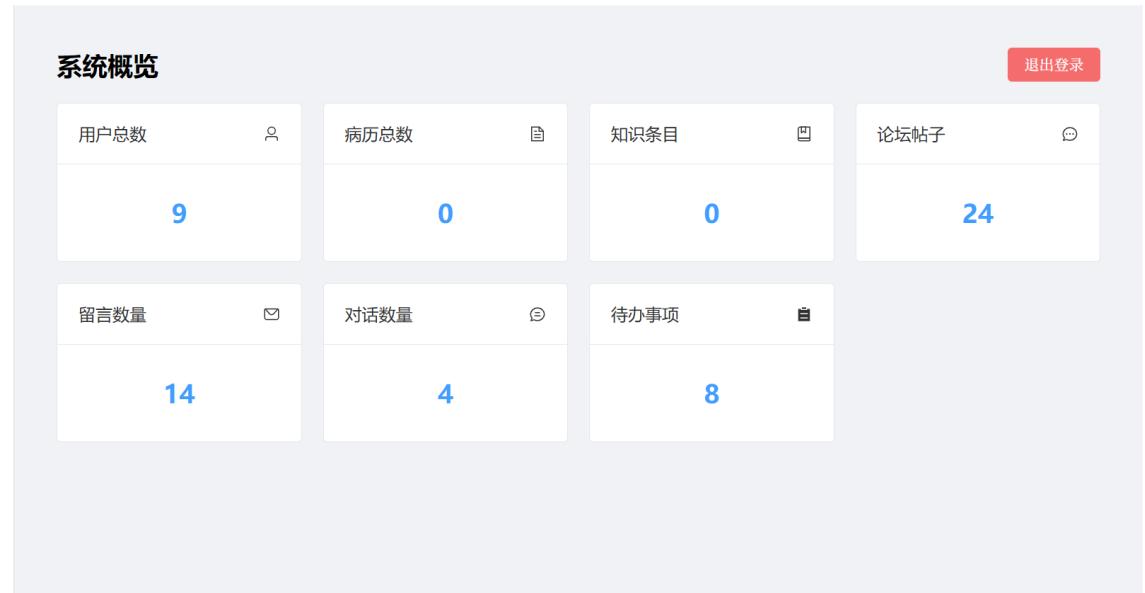


图 5-34 后台管理系统 数据看板

ID	姓名	用户类型	性别	年龄	来源地	联系方式	操作
1	Quynh Trang 01	患者	男	28	Tuy Hoa, Phu Yen, Viet Nam.	nocont_02	<button>编辑</button> <button>删除</button>
2	Quynh Trang 02	患者	男	28	Viet Nam	nocont_03	<button>编辑</button> <button>删除</button>
3	Du Yu	医师	女	23	山东省济南市	202103180009@st u.qlu.edu.cn	<button>编辑</button> <button>删除</button>
4	Du Yu	医师	女	23	山东省济南市	202103180009@te st_01	<button>编辑</button> <button>删除</button>
5	Du Yu	医师	女	23	山东省济南市	all_test	<button>编辑</button> <button>删除</button>
6	Vuong Du Tuyen	医师	男	22	山东省潍坊市	wyx	<button>编辑</button> <button>删除</button>
7	Vuong Du Tuyen	医师	男	23	山东省潍坊市	w	<button>编辑</button> <button>删除</button>

图 5-35 后台管理系统 用户管理

帖子内容详情

```
{
  "content": "测试帖子",
  "images": "8dd56703-22cb-4da3-9e80-f4e1fbba6c1a"
}
```

作者	帖子内容	回复数	状态	最后更新	操作
Tuyen	查看内容	0	已关闭	2025/4/18 07:30:22	<button>编辑</button> <button>删除</button>
Vuong Du Tuyen	查看内容	0	已关闭	2025/4/18 07:30:22	<button>编辑</button> <button>删除</button>
Нодикний	查看内容	939	正常	2025/4/18 07:30:23	<button>编辑</button> <button>删除</button>
Нодикний	查看内容	933	正常	2025/4/18 07:30:24	<button>编辑</button> <button>删除</button>
Vuong Du Tuyen	查看内容	0	已关闭	2025/4/26 02:48:49	<button>编辑</button> <button>删除</button>
Vuong Du Tuyen	查看内容	0	正常	2025/4/26 02:51:49	<button>编辑</button> <button>删除</button>

图 5-36 后台管理系统 论坛管理



图 5-37 后台管理系统 大模型会话管理

ID	用户	内容	优先级	状态	创建时间	操作
1	Vuong Du Tuyen	每周三上午10点进行康复训练	中	已完成	2025/2/16 23:10:10	<button>编辑</button> <button>删除</button>
2	Vuong Du Tuyen	每周三上午10点进行康复训练	中	已完成	1970/1/1 08:00:00	<button>编辑</button> <button>删除</button>
3	Vuong Du Tuyen	每周三上午10点进行康复训练	中	已完成	2025/2/16 23:10:10	<button>编辑</button> <button>删除</button>
4	Vuong Du Tuyen	每周三上午10点进行康复训练	高	已完成	2025/6/12 16:56:50	<button>编辑</button> <button>删除</button>
5	Vuong Du Tuyen	本周每天用活络油按摩关节	高	未完成	2025/3/26 00:00:00	<button>编辑</button> <button>删除</button>
6	Vuong Du Tuyen	晚上十点吃清开灵	中	未完成	1970/1/1 08:00:00	<button>编辑</button> <button>删除</button>
7	Vương Dư Tuyễn	晚上十点吃清开灵	中	未完成	2025/3/30 21:59:27	<button>编辑</button> <button>删除</button>

图 5-38 后台管理系统 重要诊疗事项清单管理

第 6 章 系统测试

给软件系统测试就像给马上上市的新车做全面路测——不仅要检查发动机参数，更要在真实路况中验证每个零件的协同工作。给系统一整个质量把关的过程，总要经历几个关键阶段，它远不止是按部就班的流程，本质上更像是一场开发者与需求的博弈，本系统内容繁多，涉及前后端与 AI 多个模块互联，模块内部逻辑复杂，模块间的相互调用还涉及到网络、消息队列等，“连接机构”众多，若不进行有效的测试，极易发生系统 Bug。

为此本人决定对系统做面向 API 的白盒单元测试和面向功能的黑盒测试两部分，一来这相当于双保险，双重测试下会尽最大可能减少软件错误，二来这种方法也合逻辑——首先站在开发者的角度测试内部所有函数，减小因“粗心大意”导致的问题，然后站在用户的角度，完全模拟用户使用系统，着重发现那些意想不到的问题。花这部分精力去做系统测试，除了能实现课本里教的“提升质量”、“减少缺陷”，本人觉得最重要的是作为一个医疗系统开发者对用户的负责。

6.1 针对业务逻辑层 API 的单元测试

该部分为明康慧医软件系统测试的第一步，属于白盒测试，具体测试方法为，仅在测试环境下启动系统后端，包括业务逻辑层以及各分布式 AI 智能服务层，然后使用 ApiFox 软件向各 API 发送请求（也可用 Postman 代替），核查返回值或执行的操作是否可达预期，若没有，则需通过打断点在函数内定位代码错误。

下列表格为针对业务逻辑层 API 的单元测试的测试步骤与用例表格。

表 6-1 /api/test (Token 验证) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	已获取有效 Token	携带有效 Token 发送 POST 请求	返回 code=0, msg=Token 验证成功	与预期结果一致
02	未携带 Token	发送 POST 请求	返回 401 Unauthorized 错误	与预期结果一致
03	Token 过期	携带过期 Token 发送 POST 请求	返回 401 Unauthorized 错误	与预期结果一致
04	Token 无效(篡改)	携带篡改后的 Token 发送 POST 请求	返回 401 Unauthorized 错误	与预期结果一致
05	Token 格式错误 (非 JWT)	携带非 JWT 格式的 Token 发送 POST 请求	返回 401 Unauthorized 错误	与预期结果一致

表 6-2 /api/loginVerification (用户登录) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	用户存在, 密码正确	传入 userLoginKey=用户 ID, userLoginKeyType =0, 正确密码	code=0, 返回 accessToken 和 userId	与预期结果一致
02	用户存在, 密码错误	传入 userLoginKey=用户 ID, userLoginKeyType =0, 错误密码	code=1, msg=登录失败, 密码错误	与预期结果一致
03	用户不存在	传入错误 userLoginKey, userLoginKeyType =0	code=1, msg=登录失败, 用户不存在或用户名有错误	与预期结果一致
04	用联系方式登录	传入 userLoginKey=有效手机号, userLoginKeyType =1, 正确密码	code=0, 返回 accessToken 和 userId	与预期结果一致
05	缺少必填参数	不传入 userPassword 参数	code=1, msg=用户登录标识或密码不可读取	与预期结果一致
06	无效登录类型	传入 userLoginKeyType =2 (非法值)	code=1, msg=未知用户登录标识类型	与预期结果一致

表 6-3 /api/register (用户注册) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	所有参数合法且唯一	传入完整合法参数 (含未注册的 userContact)	code=0, 返回 userId 和注册时间	与预期结果一致
02	userContact 已存在	传入已注册的 userContact	code=1, msg=您提供的联系方式已存在, 请您更换	与预期结果一致
03	缺少必填参数 (userName)	不传入 userName 参数	code=1, msg=表单数据出现严重问题	与预期结果一致
04	非法 userSex (值为 2)	传入 userSex=2 (非法值)	code=1, msg=表单数据出现严重问题	与预期结果一致
05	非法 userType (值为 3)	传入 userType=3 (非法值)	code=1, msg=表单数据出现严重问题	与预期结果一致
06	空密码	传入 userPassword	code=1, msg=表单	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
		为空字符串	数据出现严重问题	

表 6-4 /api/getUserAvatar (获取用户头像) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	用户存在且有头像	传入有效 userId (存在且有头像)	code=0, 返回带头部的 Base64 头像字符串	与预期结果一致
02	用户存在但无头像	传入有效 userId (存在但 userAvatarId 为空)	code=1, msg=该用户不存在 (或头像文件不存在)	与预期结果一致
03	用户不存在	传入无效 userId	code=1, msg=该用户不存在	与预期结果一致
04	缺少 userId 参数	不传入 userId 参数	code=1, msg=用户 ID 不可读取	与预期结果一致
05	userId 为负数	传入 userId=-1	code=1, msg=用户 ID 不可读取	与预期结果一致

表 6-5 /api/getUserInfo (获取用户信息) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	查看自己信息 (权限全开)	传入自己的 userId, 所有权限为公开	返回完整用户信息 (不含密码)	与预期结果一致
02	查看他人信息 (部分私密)	传入他人 userId, 对方设置性别为私密 (userSexPermission=1)	返回的 userInfo 不包含 userSex 字段	与预期结果一致
03	用户不存在	传入无效 userId	code=1, msg=该用户不存在	与预期结果一致
04	缺少 userId 参数	不传入 userId 参数	code=1, msg=用户 ID 不可读取	与预期结果一致
05	跨用户访问 (无权限)	传入他人 userId, 对方设置联系方式私密 (userContactPermission=1)	返回的 userInfo 不包含 userContact 字段	与预期结果一致

表 6-6 /api/modifyUserInfo (修改用户信息) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	合法修改 (无冲突)	修改 userName 和 userContact (新联系方式未注册)	code=0, msg=修改成功	与预期结果一致
02	userContact 冲突	修改 userContact	code=1, msg=你	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
03	缺少必填参数 (userName)	为已注册的他人联系方式 不传入 userName 参数	提供的联系方式已存在, 请你更换 code=1, msg=表单数据出现严重问题	与预期结果一致
04	非法 userType (值为 3)	传入 userType=3 (非法值)	code=1, msg=表单数据出现严重问题	与预期结果一致
05	修改他人信息 (越权)	修改他人信息 (越权) 传入他人 userId (尝试修改他人信息, 实际应修改自己的信息)	数据库仅修改当前用户信息, 不影响他人 与预期结果一致	与预期结果一致

表 6-7 /api/modifyUserAvatar (修改用户头像) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	合法头像 Base64 字符串	传入有效的带头部 Base64 头像字符串	code=0, msg=修改成功, 旧头像被删除	与预期结果一致
02	空头像数据	不传入 userAvatar 参数	code=1, msg=头像数据不可读取	与预期结果一致
03	非法 Base64 字符串	传入无效 Base64 字符串 (如文本内容)	抛出异常, code=1, msg=修改失败	与预期结果一致
04	用户不存在	通过篡改 token 传入无效 userId	code=1, msg=该用户不存在	与预期结果一致
05	头像文件删除失败	修改头像后手动保留旧文件 (模拟异常)	旧文件仍存在, 需检查代码逻辑 (预期应删除成功)	与预期结果一致

表 6-8 /api/modifyUserPassword (修改用户密码) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	旧密码正确	传入正确旧密码和新密码	code=0, msg=修改成功	与预期结果一致
02	旧密码错误	传入错误旧密码和新密码	code=1, msg=旧密码错误	与预期结果一致
03	空密码	传入 userOldPassword 或 userNewPassword 为空	code=1, msg=密码不可读取	与预期结果一致
04	新密码与旧密码相同	传入新密码与旧密码一致	允许修改 (需确认业务逻辑是否禁止)	与预期结果一致
05	数据库写入失败	模拟数据库连接	code=1, msg=后台	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
		中断（如断开数据库）	数据库写入失败	

表 6-9 /api/addMailItem（发送留言）测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	接收者存在，内容合法	传入有效接收者 userId 和留言内容	code=0, msg=发送成功	与预期结果一致
02	接收者不存在	传入无效接收者 userId	code=1, msg=留言接收者不存在	与预期结果一致
03	空留言内容	不传入 mailItemContent 参数	code=1, msg=留言内容或接收者 ID 不可读取	与预期结果一致
04	接收者 ID 为负数	传入 mailItemReceiver UserId=-1	code=1, msg=留言内容或接收者 ID 不可读取	与预期结果一致
05	发送者与接收者相同	传入自己的 userId 作为接收者	允许发送（需确认业务逻辑是否允许）	与预期结果一致

表 6-10 /api/getMailList（获取留言列表）测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	有发送的留言 (mode=0)	传入 mode=0, 用户有已发送的留言	返回非空 mailList 数组，包含已发送的留言	与预期结果一致
02	有接收的留言 (mode=1)	传入 mode=1, 用户有未读留言	返回非空 mailList 数组，包含接收的留言	与预期结果一致
03	无留言 (mode=0)	传入 mode=0, 用户未发送过留言	返回空 mailList 数组	与预期结果一致
04	无效 mode (值为 2)	传入 mode=2	code=1, msg=未知获取模式	与预期结果一致
05	留言已被删除 (status=1)	发送留言后标记为已删除 (status=1)	不返回已删除的留言	与预期结果一致

表 6-11 /api/deleteMailItem（删除留言）测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	留言存在且为自己发送	传入自己发送的有效 mailItemId	code=0, msg=删除成功，留言 status 标记为 1	与预期结果一致
02	留言不存在	传入无效 mailItemId	code=1, msg=留言不存在	与预期结果一致
03	尝试删除他人留言	传入他人发送的 mailItemId	数据库不执行删除操作（权限校	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
04	缺少 mailItemId 参数	不传入 mailItemId 参数	code=1, msg=留言 ID 不可读取	与预期结果一致
05	mailItemId 为负数	传入 mailItemId=-1	code=1, msg=留言 ID 不可读取	与预期结果一致

表 6-12 /api/multimodalDiagnosisSubmitTask (提交多模态诊断任务) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	所有参数合法	传入 language=zh, texts=[“症状 1”, “症状 2”], 有效 imageBase64	code=0, 返回 taskId (GUID 格式)	与预期结果一致
02	文本对比项不足 (1 个)	传入 texts=[“症状 1”]	code=1, msg=不可以只有一个对比项	与预期结果一致
03	非法 language (值为 fr)	传入 language=fr	code=1, msg=未知语言类型	与预期结果一致
04	缺少 imageBase64	不传入 imageBase64 参数	code=1, msg=图像信息不可读取	与预期结果一致
05	图像 Base64 无效	传入错误格式的 imageBase64 字符串	多模态服务返回错误, taskStatus=2	与预期结果一致

表 6-13 /api/multimodalDiagnosisGetStatus (获取任务状态) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	任务存在且进行中	传入有效 taskId (任务未完成)	code=0, taskStatus=1, msg=任务进行中	与预期结果一致
02	任务存在且已完成	传入有效 taskId (任务已完成)	code=0, taskStatus=0, 返回 taskResult	与预期结果一致
03	任务不存在	传入无效 taskId	code=0, taskStatus=1 (视为任务进行中, 需确认业务逻辑)	与预期结果一致
04	缺少 taskId 参数	不传入 taskId 参数	code=1, msg=任务 ID 不可读取	与预期结果一致
05	任务失败 (如参数错误)	提交非法参数任务后查询状态	code=0, taskStatus=2, msg=任务失败	与预期结果一致

表 6-14 /api/addImportantItem (添加重要事项) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
------	------	------	------	------

用例编号	前置条件	测试步骤	预期结果	实际结果
01	一次性事项，参数完整	传入 listItemTimeMode =0, listItemContent=“服药”， startTime=当前时间+3600, endTime=当前时间+7200	code=0, msg=添加成功，数据库写入记录	与预期结果一致
02	周期性事项，参数完整	传入 listItemTimeMode =1, listItemContent=“复诊”， listItemTimeWeek =2 (星期二)	code=0, msg=添加成功，数据库记录时间模式为周期性	与预期结果一致
03	无时间要求事项	传入 listItemTimeMode =2, listItemContent=“检查报告”	code=0, msg=添加成功， startTime/endTime 和 timeWeek 为 0	与预期结果一致
04	缺少必填参数（内容为空）	传入 listItemContent=“”	code=1, msg=重要事项内容不可读取	与预期结果一致
05	非法时间模式（值为 3）	传入 listItemTimeMode =3	code=1, msg=未知时间模式	与预期结果一致
06	周期性事项缺少星期参数	传入 listItemTimeMode =1 但不填 listItemTimeWeek	code=1, msg=后台数据库写入失败 (参数缺失)	与预期结果一致

表 6-15 /api/deleteImportantItem (删除重要事项) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	事项存在且未删除	传入有效 listItemId (状态为未删除)	code=0, msg=删除成功， listItemStatus 标记为 1	与预期结果一致
02	事项已删除	传入已删除的 listItemId (status=1)	数据库不执行操作，返回 code=0 (逻辑上视为删除成功成功，因状态已为删除)	与预期结果一致
03	事项不存在	传入无效 listItemId	code=1, msg=后台数据库写入失败 (记录不存在)	与预期结果一致
04	缺少 listItemId 参数	不传入 listItemId 参数	code=1, msg=重要事项 ID 不可读	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
05	尝试删除他人事项	通过篡改参数传入他人的 listItemId	数据库不执行删除（权限校验，仅允许删除当前用户事项）	与预期结果一致

表 6-16 /api/getImportantList（获取重要事项清单）测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	有未删除的事项	添加多个有效事项后调用接口	返回非空 importantList 数组，包含 status=0 的记录	与预期结果一致
02	无事项	用户未添加任何事项	返回空数组	与预期结果一致
03	事项已删除 (status=1)	添加事项后标记为删除 (status=1)	不返回已删除的事项	与预期结果一致
04	跨用户查询（越权）	通过篡改 token 查询他人事项	返回当前用户的事项，不包含他人数据	与预期结果一致
05	数据库查询失败	模拟数据库连接异常	code=1, msg=后台数据库查询失败	与预期结果一致

表 6-17 /api/finishImportantItem（标记事项完成）测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	事项存在且未完成	传入 listItemId=有效 ID, listItemIsFinished=1 (标记完成)	code=0, msg=标记成功，数据库更新为已完成状态	与预期结果一致
02	事项已完成	传入已标记完成的事项 ID, listItemIsFinished=1	code=0, msg=标记成功 (幂等性，允许重复调用)	与预期结果一致
03	事项不存在	传入无效 listItemId	code=1, msg=后台数据库写入失败 (记录不存在)	与预期结果一致
04	非法完成状态 (值为 2)	传入 listItemIsFinished=2	code=1, msg=未知完成状态	与预期结果一致
05	缺少 listItemIsFinished 参数	不传入 listItemIsFinished 参数	code=1, msg=未知完成状态 (参数缺失)	与预期结果一致

表 6-18 /api/getCurrentTime（获取当前时间）测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	正常请求	发送 GET 或 POST	code=0, 返回	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
02	无特殊条件	请求 多次调用接口	currentTime (Unix 时间戳, 数值>0) 每次返回时间戳 递增(或相等, 误 差在合理范围内)	与预期结果一致
03	服务器时间异常	模拟服务器时间 错误(如设置为 1970 年)	返回时间戳为 0 或 负数(需根据代码 逻辑处理, 预期返 回当前真实时间)	与预期结果一致
04	请求方法错误 (PUT)	发送 PUT 请求	返回 405 Method Not Allowed 错误	与预期结果一致
05	带无关参数	传入任意无效参 数(如 test=123)	忽略参数, 正常返 回时间戳	与预期结果一致

表 6-19 /api/llmInferenceSubmitTask (提交大语言模型推理任务) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	合法参数(含上下 文)	传入 prompt=“症 状分析”, context=[{'role': 'u ser', 'content': '头 痛'}]	code=0, 返回 taskId (GUID 格 式)	与预期结果一致
02	无上下文(context 为空)	传入 prompt=“天 气查询”, context=[]	code=0, 提交成 功(允许空上下 文)	与预期结果一致
03	空 prompt	不传入 prompt 参 数	code=1, msg=提 示词不可读取	与预期结果一致
04	context 非列表类 型	传入 context={"role": "u ser", "content": "错 误格式"}	code=1, msg=会话 历史不可读取	与预期结果一致
05	超长 prompt (>1000 字)	传入 prompt="a"*1001 , context=[]	大语言模型端可 能截断或报错, taskStatus=2(需视 服务端限制)	与预期结果一致
06	非法角色 (role=admin)	传入 context=[{'role': 'a dmin', 'content': ' 敏感指令'}]	大语言模型拒绝 响应, taskResult 包含安全提示	与预期结果一致

表 6-20 /api/llmInferenceGetStatus (获取大语言模型任务状态) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	任务存在且进行 中	提交任务后立即 查询	code=0, taskStatus=1, msg=任务进行中	与预期结果一致
02	任务已完成	提交简单任务(如 天气查询)后等待	code=0, taskStatus=0, 返	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
03	任务失败（参数错误）	提交含非法角色的 context 任务后查询	code=0, taskStatus=2, msg=任务失败, taskResult 包含错误信息	与预期结果一致
04	无效 taskId	传入随机字符串作为 taskId	code=0, taskStatus=1 (视为任务进行中, 需确认业务逻辑是否允许)	与预期结果一致
05	缺少 taskId 参数	不传入 taskId 参数	code=1, msg=任务 ID 不可读取	与预期结果一致

表 6-21 /api/saveLlmSession (保存大语言模型会话) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	新建会话 (sessionId=-1)	传入 sessionId=-1, 合法 sessionContent (至少 2 条对话)	code=0, 返回新 sessionId, 数据库新增记录	与预期结果一致
02	更新现有会话	传入已存在的 sessionId, 修改 sessionContent 后提交	code=0, 数据库记录更新, sessionId 不变	与预期结果一致
03	会话内容过短 (1 条)	传入 sessionContent 仅包含 1 条对话	code=1, msg=对话过短, 不可保存	与预期结果一致
04	非法 sessionId (非整数)	传入 sessionId="abc"	code=1, msg=对话 ID 不可读取(类型错误)	与预期结果一致
05	无 isSessionDM 参数	不传入 isSessionDM 参数	code=1, msg=对话时间不可读取 (实际为参数缺失)	与预期结果一致
06	跨用户保存会话	通过篡改 token 传入他人 sessionId	数据库仅保存当前用户会话, 不影响他人	与预期结果一致

表 6-22 /api/getLlmSession (获取指定会话内容) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	会话存在且为当前用户	保存会话后传入合法 sessionId	code=0, 返回 sessionContent (JSON 格式)	与预期结果一致
02	会话不存在	传入无效 sessionId	code=1, msg=该会话不存在或您无权读取该 ID 的会	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
03	跨用户查询	传入他人 sessionId (通过篡改参数)	code=1, msg=该会话不存在或您无权读取该 ID 的会话内容 (权限校验)	与预期结果一致
04	缺少 sessionId 参数	不传入 sessionId 参数	code=1, msg=对话 ID 不可读取	与预期结果一致
05	会话已删除 (假删)	删除会话后 (修改 status) 查询	code=1, msg=该会话不存在或您无权读取该 ID 的会话内容 (逻辑删除)	与预期结果一致

表 6-23 /api/getLlmSessionList (获取会话列表) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	有多个会话 (含 DM 和非 DM)	保存多个会话, 传入 isSessionDM=0 和 1 分别查询	返回对应类型的会话列表, sessionTitle 为用户首次发言前 16 字	与预期结果一致
02	无会话	用户未保存任何会话	返回空数组	与预期结果一致
03	非法 isSessionDM 值 (2)	传入 isSessionDM=2	code=1, msg=对话时间不可读取 (实际为参数值非法)	与预期结果一致
04	跨用户查询	通过篡改 token 查询他人会话	返回当前用户的会话列表, 不包含他人数据	与预期结果一致
05	会话内容解析失败	手动修改数据库中 sessionContent 为非法 JSON 格式	code=1, msg=未能成功获取会话列表 (JSON 解析错误)	与预期结果一致

表 6-24 /api/deleteLlmSession (删除会话) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	会话存在且为当前用户	传入合法 sessionId (当前用户创建)	code=0, msg=删除成功, 数据库记录被物理删除	与预期结果一致
02	会话不存在	传入无效 sessionId	code=1, msg=未能成功删除会话记录 (记录不存在)	与预期结果一致
03	跨用户删除	传入他人 sessionId (通过篡改参数)	code=1, msg=未能成功删除会话记录 (权限不足)	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
04	缺少 sessionId 参数	不传入 sessionId 参数	code=1, msg=对话 ID 不可读取	与预期结果一致
05	多次删除同一会话	重复调用删除接口	首次成功, 后续提示记录不存在	与预期结果一致

表 6-25 /api/addForum (创建论坛) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	合法参数 (医学知识论坛)	传入 forumName=“癌症研究”, forumType=0, forumPermission=0	code=0, 返回 forumId 和创建时间	与预期结果一致
02	权限仅限医师	传入 forumPermission=1 (当前用户为医师)	创建成功, 论坛仅限医师参与	与预期结果一致
03	非法 forumType (值为 2)	传入 forumType=2	code=1, msg=参数格式错误 (forumType 需为 0 或 1)	与预期结果一致
04	缺少 forumName 参数	不传入 forumName 参数	code=1, msg=参数格式错误 (forumName 为空)	与预期结果一致
05	跨权限创建 (患者创建医师论坛)	当前用户为患者, 传入 forumPermission=1	创建失败, 提示权限不足 (需业务逻辑校验用户类型)	与预期结果一致

表 6-26 /api/getForumList (获取论坛列表) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	筛选医学知识论坛	传入 forumType=0, forumPermission=3 (所有权限)	返回类型为医学知识的论坛列表	与预期结果一致
02	筛选仅限患者论坛	传入 forumType=1, forumPermission=2	返回类型为疾病且权限仅限患者的论坛	与预期结果一致
03	无筛选条件	传入 forumType=2, forumPermission=3	返回所有未删除的论坛	与预期结果一致
04	非法 forumType (值为 3)	传入 forumType=3	code=1, msg=参数格式错误 (forumType 需为	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
05	论坛已删除 (status=1)	创建论坛后标记 为删除 (forumStatus=1)	0-2) 不返回已删除的 论坛	与预期结果一致

表 6-27 /api/modifyForumType (修改论坛类型) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	合法修改 (医学→疾病)	当前用户为创建者, 传入 forumId 和 newType=1	code=0, msg=论坛类型修改成功	与预期结果一致
02	非创建者修改	传入他人创建的 forumId	code=1, msg=只有创建者才有权修改论坛类型	与预期结果一致
03	论坛已删除	传入已删除的 forumId (forumStatus=1)	code=1, msg=论坛不存在或已删除	与预期结果一致
04	非法 newType (值为 2)	传入 newType=2	code=1, msg=无效论坛类型值	与预期结果一致
05	缺少 forumId 参数	不传入 forumId 参数	code=1, msg=缺少必要参数	与预期结果一致

表 6-28 /api/deleteForum (删除论坛) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	论坛存在且为创建者	传入合法 forumId (当前用户创建)	code=0, msg=删除成功 (标记 forumStatus=1)	与预期结果一致
02	非创建者删除	传入他人创建的 forumId	code=1, msg=无删除权限	与预期结果一致
03	物理删除 (假删逻辑)	数据库实际执行逻辑为假删 (修改 status, 非真删除)	论坛仍存在于数据库, 但 status=1, 不被查询接口返回	与预期结果一致
04	缺少 forumId 参数	不传入 forumId 参数	code=1, msg=缺少 forumId 参数	与预期结果一致
05	forumId 为字符串	传入 forumId="abc"	code=1, msg=forumId 必须为整数	与预期结果一致

表 6-29 /api/getForumInfo (获取论坛信息) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	论坛存在且未删除 (status=0)	传入合法 forumId (status=0)	code=0, 返回 forumInfo (包含创建者、类型、权限等)	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
02	论坛已删除	传入已删除的 forumId (status=1)	code=1, msg=该论坛不存在或您无权读取该 ID 的论坛内容	与预期结果一致
03	跨用户查询 (非创建者)	传入他人创建的 forumId (status=0)	返回论坛信息 (公开字段), 但无法修改 (权限由修改接口控制)	与预期结果一致
04	缺少 forumId 参数	不传入 forumId 参数	code=1, msg=论坛 ID 不可读取	与预期结果一致
05	forumId 为负数	传入 forumId=-1	code=1, msg=论坛 ID 不可读取	与预期结果一致

表 6-30 /api/sendPost (发送帖子) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	合法参数 (含 3 张图片)	传入 forumId=合法 ID, postContent=“病情描述”, 3 张合法图片 Base64	code=0, 返回 postId, 图片转为 webp 格式保存, 数据库存储 GUID 列表	与预期结果一致
02	图片数量超限 (4 张)	传入 4 张图片 Base64 字符串	code=1, msg=帖子图片数量超过限制	与预期结果一致
03	图片大小超限 (301KB)	传入单个 Base64 长度为 300*1024+1 的图片	code=1, msg=帖子图片大小超过限制	与预期结果一致
04	无图片	不传入 postImagesBase64 List 参数	code=0, 返回 postId, 图片字段为空	与预期结果一致
05	跨论坛发帖 (权限不足)	传入 forumId 对应的权限为仅限医生, 当前用户为患者	发帖成功 (权限由论坛创建时控制, 发帖接口仅校验 forumId 存在性)	与预期结果一致
06	forumId 不存在	传入无效 forumId	code=1, msg=论坛 ID 不可读取	与预期结果一致

表 6-31 /api/getPostList (获取帖子 ID 列表) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	论坛存在且有未删除帖子	传入合法 forumId (存在多个 postStatus=0 的帖子)	返回包含帖子 ID 的 JSON 数组, 长度≥1	与预期结果一致
02	论坛无帖子	传入合法 forumId 但无有效帖子	返回空数组	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
03	论坛已删除 (status=1)	传入已删除的 forumId (forumStatus=1)	code=1, msg=论坛 ID 不可读取(需确 认业务逻辑是否 允许查询已删除 论坛)	与预期结果一致
04	缺少 forumId 参数	不传入 forumId 参 数	code=1, msg=论 坛 ID 不可读取	与预期结果一致
05	非法 forumId (字 符串)	传入 forumId="abc"	code=1, msg=论坛 ID 不可读取(类型 错误)	与预期结果一致

表 6-32 /api/getPostContent (获取帖子具体内容) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	帖子存在且未删 除	传入合法 postId (postStatus=0)	code=0, 返回帖 子内容、图片 Base64、发布者 ID 等	与预期结果一致
02	帖子已删除 (status=1)	传入已删除的 postId (postStatus=1)	code=1, msg=该 帖子不存在或已 被删除	与预期结果一致
03	图片文件不存在	手动删除服务器 上的图片文件(保 留数据库 GUID)	返回 images 为空 数组(接口自动校 验文件存在性)	与预期结果一致
04	缺少 postId 参数	不传入 postId 参数	code=1, msg=帖 子 ID 不可读取	与预期结果一致
05	跨用户查询敏感 内容	传入他人发布的 含敏感内容的 postId	正常返回内容(权 限由删除接口控 制, 查询无权限校 验)	与预期结果一致

表 6-33 /api/deletePost (删除帖子) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	帖子存在且为发 布者	传入合法 postId (当前用户发 布, postStatus=0)	code=0, msg=删 除成功, postStatus=1, 图 片文件物理删除	与预期结果一致
02	非发布者删除	传入他人发布的 postId	code=1, msg=该帖 子不存在或已被 删除或您无权删 除	与预期结果一致
03	多次删除同一帖 子	重复调用删除接 口	首次成功, 后续提 示帖子已删除	与预期结果一致
04	缺少 postId 参数	不传入 postId 参数	code=1, msg=帖 子 ID 不可读取	与预期结果一致
05	帖子无图片	传入无图片的	code=0, msg=删除	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
		postId	成功(无图片文件 需删除)	

表 6-34 /api/praisePost (给帖子点赞) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	帖子存在且未删除	传入合法 postId (postStatus=0)	code=0, msg=点赞成功, postPraiseNumber +1	与预期结果一致
02	重复点赞	对同一帖子连续调用点赞接口	code=0, msg=点赞成功(允许重复点赞, 无去重逻辑)	与预期结果一致
03	帖子已删除	传入已删除的 postId (postStatus=1)	code=1, msg=该帖子不存在或已被删除	与预期结果一致
04	缺少 postId 参数	不传入 postId 参数	code=1, msg=帖子 ID 不可读取	与预期结果一致
05	给自己的帖子点赞	传入当前用户发布的 postId	code=0, msg=点赞成功(允许用户给自己点赞)	与预期结果一致

表 6-35 /api/exportChatToPDF (导出聊天记录为 PDF) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	合法会话且为当前用户	传入合法 sessionId (当前用户创建, 含对话内容)	返回 PDF 文件流, 文件名包含会话标题和时间	与预期结果一致
02	会话内容含特殊字符	会话内容包含表情符号、Markdown 格式文本	PDF 正确渲染特殊字符(需依赖 export_chat_to_pdf 函数处理能力)	与预期结果一致
03	跨用户导出	传入他人 sessionId (通过篡改参数)	code=1, msg=会话不存在或您无权导出该 ID 会话内容	与预期结果一致
04	缺少 sessionId 参数	不传入 sessionId 参数	返回 400 错误, msg=会话 ID 不可读取	与预期结果一致
05	会话为 LLM 研讨机制	传入 isSessionDM=1 的会话 ID	PDF 包含研讨机制标识(需函数支持区分渲染)	与预期结果一致

表 6-36 /api/sendEmail (发送邮件) 测试

用例编号	前置条件	测试步骤	预期结果	实际结果
01	合法参数(纯文本)	传入 receiver=有	code=0, msg=发	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
02	含 HTML 内容 内容)	效邮箱， subject=“测试”， content=“邮件内 容”	送成功	
03	无效邮箱地址	传入 receiver=“abc@inv alid”	邮件发送失败， code=1, msg=发 送失败（需邮件服 务商校验）	与预期结果一致
04	缺少 receiver 参数	不传入 receiver 参 数	code=1, msg=收 件人邮箱地址不 可为空	与预期结果一致
05	内容为空	传入 content=“”， receiver=有效邮箱	code=1, msg=邮 件内容不可为空	与预期结果一致
06	超大附件（模拟）	传入 content=“a”10241 024（假设触发附 件大小限制）	邮件发送失败（需 服务端限制逻辑）	与预期结果一致

6.2 针对系统功能的黑盒测试

在本系统中，黑盒测试是系统全部开发完成后，以用户的身份对系统整体进行测试的过程，测试过程中不考虑运行细节，只是将自己当作系统的用户，将系统功能全部使用一遍，若发现严重错误，则将修改出错部分所在层和模块的代码，并重新做单元测试与黑盒测试，直到不出错误为止。下方表格展示了针对明康慧医系统功能的黑盒测试用例，这些用例覆盖 8 大模块，用例总数约 100 条，关键路径、正常路径、异常路径、权限验证、边界条件、交互行为、模型调用、消息队列交互、前后端通信等理论上可以做到全部覆盖。

表 6-37 【注册登录模块】测试用例表

用例编号	前置条件	测试步骤	预期结果	实际结果
01	无	进入注册页面，填 写完整有效信息， 选择联系方式注 册，提交注册	提示注册成功，分 配账号号码	与预期结果一致
02	无	进入注册页面，信 息不完整（缺手机 号）提交注册	提示信息不完整， 注册失败	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
03	用户已注册	使用正确联系方式登录	登录成功，跳转欢迎页	与预期结果一致
04	用户已注册	使用正确账号号码登录	登录成功，跳转欢迎页	与预期结果一致
05	用户已注册	使用错误账号号码或联系方式登录	登录失败，提示账号或密码错误	与预期结果一致
06	用户已注册	在登录页面输入正确账号，错误密码	登录失败，提示账号或密码错误	与预期结果一致
07	用户已登录	登录后点击“一键查看未完成事项”按钮	跳转到诊疗事项页面，显示未完成事项	与预期结果一致
08	用户已登录且有高优先事项	登录后点击“一键查看高优先事项”按钮	跳转到诊疗事项页面，显示高优先级事项	与预期结果一致
09	用户已登录	登录后观察欢迎页背景图片是否自动轮换	背景图片每隔一段时间自动变换	与预期结果一致

表 6-38 【我的主页模块】测试用例表

用例编号	前置条件	测试步骤	预期结果	实际结果
01	用户已登录	进入“我的主页”，查看个人信息展示	所有信息正确展示，头像、姓名、用户描述、其他按权限显示	与预期结果一致
02	用户已登录	进入他人主页，查看信息	按对方设置的权限展示信息	与预期结果一致
03	用户已登录	进入“我的主页”，尝试修改头像	修改头像成功，新头像正确显示	与预期结果一致
04	用户已登录	进入“我的主页”，尝试修改密码	修改密码成功，下次登录时使用新密码	与预期结果一致
05	用户已登录	修改个人信息中的联系方式、来源、备注等	信息修改成功，界面正确显示修改后的信息	与预期结果一致
06	用户已登录	修改信息公开权限(如联系方式设置为不公开)	他人查看时不显示该信息	与预期结果一致
07	用户已登录	进入“我的主页”留言列表，查看发送留言	正常显示所有已发送留言	与预期结果一致
08	用户已登录	进入“我的主页”留言列表，查看接收留言	正常显示所有接收留言	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
09	用户已登录	进入他人主页，点击发送留言，输入内容并提交	留言发送成功，对方留言列表显示该留言	与预期结果一致
10	用户未登录	直接访问他人主页	被重定向到登录页	与预期结果一致
11	用户已登录	点击他人主页头像	跳转到对应用户主页	与预期结果一致
12	用户已登录，设置信息为公开	他人查看主页，查看信息展示	信息全部显示	与预期结果一致

表 6-39 【智能机器辅诊模块】测试用例表

用例编号	前置条件	测试步骤	预期结果	实际结果
01	用户已登录	上传医学影像文件，输入多个英文诊断描述，提交进行推理	后端接收请求，开始推理，轮询后显示各描述对应概率分布	与预期结果一致
02	用户已登录	上传医学影像文件，输入多个中文诊断描述，提交进行推理	系统自动翻译为英文后推理，轮询后显示概率分布和翻译文本	与预期结果一致
03	用户已登录	上传非医学影像（如风景图），输入描述，提交进行推理	显示推理结果，但概率分布不具备医学意义，提示用户谨慎判断	与预期结果一致
04	用户已登录	不上传图片，仅输入描述，尝试提交	提示必须上传图片，无法提交	与预期结果一致
05	用户已登录	上传图片，输入描述列表为空，尝试提交	提示必须输入至少一个诊断描述，无法提交	与预期结果一致
06	用户已登录	上传图片，输入多个描述，提交后在轮询过程中关闭页面	后端继续处理，用户重新进入后不能直接看到结果	与预期结果一致
07	用户已登录	上传图片与描述，提交后轮询	当算力端返回结果时，前端自动停止轮询并显示饼图和表格	与预期结果一致
08	用户已登录	观察饼图展示效果	ECharts 饼图正确展示各描述的概率比例，点击图表可显示详细信息	与预期结果一致
09	用户已登录	观察表格展示效果	表格展示用户输入、翻译、概率百分比，并保持数据对应	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
10	用户未登录	直接访问机器辅助诊疗页面	被重定向到登录页	与预期结果一致

表 6-40 【医疗问答模块】明康慧医智慧问答功能 测试用例表

用例编号	前置条件	测试步骤	预期结果	实际结果
01	用户已登录	进入智慧问答页面，输入一个医学问题，点击发送	大模型生成医学相关回答，显示于对话区，支持 markdown 及代码高亮	与预期结果一致
02	用户已登录	进入智慧问答页面，输入非医学问题（如天气），点击发送	大模型生成回答，但提示结果仅供参考	与预期结果一致
03	用户已登录	在智慧问答页面点击“选择 RAG 知识库”，弹出用户收藏知识实体	弹窗正常显示用户收藏实体，用户可选中	与预期结果一致
04	用户已登录，已选 RAG 知识库	输入问题，使用知识增强	系统基于知识库做 RAG 增强，回答中引用相关知识片段	与预期结果一致
05	用户已登录，无知识库收藏	点击“选择 RAG 知识库”	弹窗提示无收藏实体	与预期结果一致
06	用户已登录	进行对话后，刷新页面	页面保留会话历史，可继续对话	与预期结果一致
07	用户已登录	在历史会话列表中加载过往对话	正常加载过往对话内容，界面显示正常	与预期结果一致

表 6-41 【医疗问答模块】明康慧医智能体深度分析 测试用例表

用例编号	前置条件	测试步骤	预期结果	实际结果
08	用户已登录	进入智能体深度分析页面，输入医学研究问题，点击开始分析	多个智能体进行第一轮回答，主持人总结，各阶段展示正常	与预期结果一致
09	用户已登录	进行第二轮、第三轮讨论	各智能体根据总结进行再讨论，主持人总结，每轮讨论正常显示	与预期结果一致
10	用户已登录	完成多轮讨论后，系统自动执行“收敛”	显示语义收敛程度结果，提示讨论收敛程度	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
11	用户已登录	在讨论中中途关闭页面	后台继续处理，用户需重新发起新的讨论流程	与预期结果一致
12	用户已登录	轮询时查看接口是否正确返回任务状态	轮询到结果后前端显示讨论总结，停止轮询	与预期结果一致
13	用户已登录	进行讨论时，输入非医学问题	智能体进行讨论，结果提示仅供参考	与预期结果一致
14	用户未登录	直接访问智能体深度分析页面	被重定向到登录页	与预期结果一致

表 6-42 【诊疗论坛模块】测试用例表

用例编号	前置条件	测试步骤	预期结果	实际结果
01	用户已登录	进入论坛首页，查看论坛列表	正常显示论坛名称、编号、创建时间、类别、权限、创建者姓名及头像	与预期结果一致
02	用户已登录	输入关键词搜索论坛	搜索结果正确筛选符合关键词的论坛	与预期结果一致
03	用户已登录	选择论坛类别、权限进行筛选	正确筛选对应类别、权限的论坛	与预期结果一致
04	用户已登录	点击创建论坛，输入名称、类型、权限，提交	成功创建论坛，论坛列表显示新论坛	与预期结果一致
05	用户已登录，创建者为自己	点击论坛项的“修改”按钮，修改论坛类别	修改成功，论坛类别更新	与预期结果一致
06	用户已登录，创建者为自己	点击论坛项的“删除”按钮，删除论坛	删除成功，论坛从列表移除	与预期结果一致
07	用户已登录，非论坛创建者	查看论坛项，确认“修改”按钮是否隐藏	按预期隐藏	与预期结果一致
08	用户已登录，权限不足	尝试进入限制访问的论坛	被拦截并提示无权限	与预期结果一致
09	用户已登录	进入开放论坛，查看帖子列表	正常显示帖子内容、头像、姓名、类型、来源地、发布时间、点赞数	与预期结果一致
10	用户已登录	对帖子点赞	点赞数增加 1，不能重复点赞	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
11	用户已登录	回复他人帖子	回复成功，显示在对应帖子下方	与预期结果一致
12	用户已登录，帖子发布者为自己	删除自己帖子	删除成功，帖子从列表移除	与预期结果一致
13	用户已登录	发布帖子，输入文字+图片（最多 3 张），提交	成功发布帖子，显示文字和缩略图，点击图可放大并滚动查看	与预期结果一致
14	用户已登录	尝试只上传图片，不输入文字	提示必须输入文字	与预期结果一致
15	用户已登录	点击“添图”按钮，添加图片，预览列表	正常显示上传图片，列表支持放大和删除	与预期结果一致
16	用户已登录	点击清空图片按钮	所有已添加图片被移除	与预期结果一致
17	用户未登录	直接访问论坛页面	被重定向到登录页	与预期结果一致

表 6-43 【病历管理模块】测试用例表

用例编号	前置条件	测试步骤	预期结果	实际结果
01	医师用户已登录	进入病历管理页面，点击创建病历，输入标题、正文、绑定患者，保存	成功创建病历，病历列表显示新病历	与预期结果一致
02	医师用户已登录	进入病历管理页面，选择已创建病历，点击编辑，修改正文，保存	成功保存修改，病历内容更新	与预期结果一致
03	医师用户已登录，患者已绑定	切换为患者用户，进入病历管理页面	可查看被绑定的病历，但不可编辑	与预期结果一致
04	患者用户未绑定病历	进入病历管理页面	提示无可查看病历	与预期结果一致
05	医师用户已登录	选择病历，点击“病历辅诊”，输入当前病历文本，提交	大模型返回辅诊意见，并显示在页面	与预期结果一致
06	医师用户已登录	选择病历，点击“药物推荐”，输入病历描述，提交	大模型返回推荐药物及理由，显示在页面	与预期结果一致
07	医师用户已登录	尝试删除一个病历	删除成功，病历从列表中移除	与预期结果一致
08	医师用户已登录，患者已绑定	患者尝试编辑已绑定病历	操作被禁止，提示无权限	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
09	医师用户已登录	病历编辑器支持 Markdown 语法编辑，并预览	正确渲染 markdown 格式，显示美观	与预期结果一致
10	医师用户未登录	直接访问病历管理页面	被重定向到登录页	与预期结果一致

表 6-44 【诊疗事项清单管理模块】测试用例表

用例编号	前置条件	测试步骤	预期结果	实际结果
01	用户已登录	进入诊疗事项页面，查看事项列表	列表展示事项名称、完成情况、时间、状态、事项类型、优先级等概要信息	与预期结果一致
02	用户已登录	点击某事项，查看详细信息弹窗	正确展示事项详细内容	与预期结果一致
03	用户已登录，存在未完成事项	在主页查看未完成事项提醒	正常显示未完成事项列表	与预期结果一致
04	用户已登录，存在高优先级事项	在主页查看高优先级事项提醒	正常显示高优先级事项列表	与预期结果一致
05	用户已登录	添加一个一次性事项，设置时间段，保存	添加成功，事项显示为“未到时间”或“已到时间”取决于当前时间	与预期结果一致
06	用户已登录	添加一个周期性事项，设置星期和时间点，保存	添加成功，根据当前时间显示相应的时间状态	与预期结果一致
07	用户已登录	添加一个无时间要求事项	添加成功，事项时间状态显示为空	与预期结果一致
08	用户已登录	将某事项标记为完成	状态变更为“已完成”，在列表中标识变化	与预期结果一致
09	用户已登录	删除事项	事项被删除，从列表移除	与预期结果一致
10	用户已登录	使用“医患互联”功能，填写邮件正文，输入邮箱地址，发送	邮件发送成功，收件方可收到邮件	与预期结果一致
11	用户已登录	使用“AI 辅助分析”功能，输入事项描述，提交	大模型返回计划合理性分析及建议	与预期结果一致
12	用户未登录	直接访问诊疗事项页面	被重定向到登录页	与预期结果一致
13	用户已登录，存在超时事项	在事项列表查看状态	正确显示“已超时”状态	与预期结果一致

用例编号	前置条件	测试步骤	预期结果	实际结果
14	用户已登录，事项设置紧急并未完成	在主页欢迎页点击“一键查看未完成和高优先级事项”按钮	正确跳转到事项列表，并自动筛选显示紧急及未完成事项	与预期结果一致

表 6-45 【后台管理系统模块】测试用例表

用例编号	前置条件	测试步骤	预期结果	实际结果
01	管理员用户已登录	进入后台管理首页	成功进入后台管理首页，显示可管理的各类数据库表	与预期结果一致
02	管理员用户已登录	查看数据库字段，字段为普通类型（字符串、数值）	正确显示字段名称、类型、内容	与预期结果一致
03	管理员用户已登录，字段为 JSON 格式	查看数据库字段	正确渲染 JSON 内容，highlight.js 高亮显示	与预期结果一致
04	管理员用户已登录	添加一条数据（输入合法数据）	添加成功，新数据出现在数据库列表中	与预期结果一致
05	管理员用户已登录	删除一条数据	删除成功，数据从数据库移除	与预期结果一致
06	管理员用户已登录	查看某表数据列表	正常显示数据列表	与预期结果一致
07	非管理员用户登录	尝试访问后台管理系统	被禁止访问，提示无权限或重定向到首页	与预期结果一致
08	未登录用户	直接访问后台管理系统	被重定向到登录页	与预期结果一致
09	管理员用户已登录，输入非法 JSON 格式	尝试保存字段	提示格式错误，保存失败	与预期结果一致
10	管理员用户已登录，查询条件为空或不存在	搜索字段数据	返回空列表或提示无数据	与预期结果一致

第 7 章 结论与展望

本项研究“明康慧医”项目核心目标就是开发一个智能高效的医疗健康管理系统，方便人们管理自己的健康，也能给医生提供靠谱的辅助诊疗工具。为了达成这个目标，本人以此次毕业设计研究为契机，学习、研究了各种前沿技术和 AI 算法，并且尽量整合到了这个系统中，希望能实实在在的尽自己一名本科生微不足道的力量来提升医疗服务的质与效。

在这个项目里，本人全程围绕用户需求来设计平台，借助现代信息技术打破传统医疗模式的限制，推动医疗行业创新发展。构建这个系统用到的关键技术非常多而又不“小众”，像 Python Flask 框架、MySQL 8 数据库、Rabbit MQ 消息队列，还有 Vue3 前端框架等等，它们被各大厂使用，并且技术文档详细，技术栈很稳定可靠。

需求上，本人深入分析了平台的功能性和非功能性需求，最终设计开发出了九个核心模块。不管是注册登录、个人主页管理，还是机器辅助诊断、医疗问题解答，包括诊疗论坛、诊疗事项提醒、资源中心以及后台管理，每个模块都经过严格测试和反复打磨，就是为了保证功能完整，让大家能用着顺手。

说到技术实现，项目采用的是分布式的系统架构，智能服务里的各模型部署在不同的算力服务器上，因此系统处理消息的响应就会更快更及时一些。研究中对阿里的 Qwen2.5-3B-Instruct 大模型进行了专门的优化，训练出了针对解决医学、医疗和生物学问题的 MKTY-3B-Chat 明康慧医大模型。还有 BioMedCLIP 模型能相对准确的分析医学影像，辅助医生做诊断决策，再加上 TF-IDF、RAG 这些技术和方法，系统的信息处理能力又提高了一大截。

虽然目前这个平台已基本满足了本研究最初设定的功能需求，但实际用起来还是发现了不少问题。比如可能是受数据集同质化的影响，MKTY 时间序列预测模型的表现并不佳，尽管它在理论上看起来是没有问题的。另外客观因素方面，由于本毕业设计硬件资源有限，只得选用 3B 规模模型进行训练，并且无法使用全参微调的方法，这极大的限制了大模型的能力空间，导致在复杂病情诊断时，大模型给出的推理结果有时候不太准确。并且项目中的 RAG 方法使用的是最基本的版本，后续还有待改进。还有一个最重要的问题便是，系统客户端模块的后台没有使用 ORM 框架，这个问题已“落后于时代”了。接下来的时间，我虽毕业，但从事计算机行业是我一辈子的事，我会继续不停钻研技术，把这些问题一个个解决掉，进一步提升自己的能力。

这个项目给当下的医疗健康管理辅助诊疗提供了一套新的思路，也算是能给计算机或医疗同行们做一些参考吧。不管是系统设计思路和功能模块划分，还是用户需求分析，都能给其他医疗机构或者相关单位开发类似平台提供借鉴。总的来说，我本人设计“明康慧医”这个项目不仅仅是为了满足眼前毕业设计的需求，更是希望大家都能踊跃的投入“AI+医疗”甚至是“AI+万物”的怀抱，众人合作便一定能推动医疗领域的创新发展，一定能为智能健康管理的进步出份力。我相信，随着基础科学的不断进步，在一众计算机专业和医学专业学生和研究人员的共同努力下，无所不能的“AI 医生”将不再是梦想，一定会出现。

参考文献

- [1] 王伟国,刘永萍,王生年,等. B/S 模式网上考试系统分析与设计[J].石河子大学学报（自然科学版）,2003,6(2):145-147.
- [2] Vaswani A, Shazeer N, Parmar N, 等 .Attention Is All You Need[J].arXiv, 2023. DOI: 10.48550/arXiv.1706.03762.
- [3] Devlin J, Chang M-W, Lee K, 等.BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J].arXiv, 2019. DOI: 10.48550/arXiv.1810.04805.
- [4] Touvron H, Lavril T, Izacard G, 等 .LLaMA: Open and Efficient Foundation Language Models[J].arXiv, 2023. DOI: 10.48550/arXiv.2302.13971.
- [5] Du Z, Qian Y, Liu X, 等.GLM: General Language Model Pretraining with Autoregressive Blank Infilling[J].arXiv, 2022. DOI: 10.48550/arXiv.2103.10360.
- [6] Singhal K, Tu T, Gottweis J, 等.Towards Expert-Level Medical Question Answering with Large Language Models[J].arXiv, 2023. DOI: 10.48550/arXiv.2305.09617.
- [7] Saab K, Tu T, Weng W-H, 等.Capabilities of Gemini Models in Medicine[J].arXiv, 2024. DOI: 10.48550/arXiv.2404.18416.
- [8] Ding T, Wagner S J, Song A H, 等.Multimodal Whole Slide Foundation Model for Pathology[J].arXiv, 2024. DOI: 10.48550/arXiv.2411.19666.
- [9] Radford A, Kim J W, Hallacy C, 等.Learning Transferable Visual Models From Natural Language Supervision[J].arXiv, 2021. DOI: 10.48550/arXiv.2103.00020.
- [10] He K, Mao R, Lin Q, 等.A Survey of Large Language Models for Healthcare: from Data, Technology, and Applications to Accountability and Ethics[J].arXiv, 2024. DOI: 10.48550/arXiv.2310.05694.
- [11] Lin W, Zhao Z, Zhang X, 等.PMC-CLIP: Contrastive Language-Image Pre-training using Biomedical Documents[J].arXiv, 2023. DOI: 10.48550/arXiv.2303.07240.
- [12] Zaheer M, Guruganesh G, Dubey A, 等.Big Bird: Transformers for Longer Sequences[J].arXiv, 2021. DOI: 10.48550/arXiv.2007.14062.
- [13] 于海涛,牟冬梅,王长聪.智能诊断模型的可解释性策略研究[J/OL].现代情报,1-16[2025-01-09].
- [14] 籍欣萌,昝红英,崔婷婷,等.中文医疗大模型综述：进展、评估与挑战[J].中文信息学报,2024,38(11):1-12.
- [15] 顾东晓,黄智勇,朱凯旋,等.医疗健康大模型知识体系构建、服务应用与风险协同治理[J/OL].情报科学,1-29[2025-01-09]. <http://kns.cnki.net/kcms/detail/22.1264.G2.20240925.0948.002.html>.
- [16] 余龙龙.基于对比学习的医学影像报告生成算法的研究与实现 [D].北京邮电大学,2024.DOI:10.26969/d.cnki.gbydu.2024.000560.
- [17] 胡雪晴,韩琪,付磊,等 .AI 虚拟医生在医疗行业的应用研究综述 [J].电脑知识与技术,2024,20(29):15-17.DOI:10.14004/j.cnki.ckt.2024.1447.
- [18] 曹建峰,徐艳玲.医疗领域多模态 AI 模型的机遇、挑战与治理应对 [J].中国医学伦理

- 学,2024,37(09):1023-1029.
- [19] 李苗苗.智能网联技术赋能的智慧医疗救护车改装设计与系统优化研究[J].汽车维修技师,2025,(08):52-54.
- [20] 王璐玮,陈家晟,陈蔚达,等.某妇儿医院儿童合理用药智能辅助决策系统的构建及应用[J].中国数字医学,2025,20(01):13-19.
- [21] 廖永珍,王晓辉,邱洁,等.2003—2023 年口腔癌智慧医疗文献循证可视化及对比分析[J].国际口腔医学杂志,2025,52(01):50-60.
- [22] 王安,章璐,刘丹青,等.基于“智慧医疗+中医药”系统的建设与实践[J].中医药管理杂志,2024,32(20):222-224.DOI:10.16690/j.cnki.1007-9203.2024.20.012.
- [23] 张志成,王静,张阳,等.OrthoGPT：面向精准诊疗的多模态骨科大模型[J].智能科学与技术学报,2024,6(03):338-346.
- [24] 王文硕.面向智慧医疗的联邦学习关键安全技术研究[D].烟台大学,2024.DOI:10.27437/d.cnki.gytdu.2024.000802.
- [25] 冯羽.“IoT+数据挖掘”下医院智慧医疗健康管理系统设计[J].中国新技术新产品,2024,(10):143-145.DOI:10.13612/j.cnki.cntp.2024.10.031.
- [26] 刘盈全.面向智慧医疗的大数据可视化管理系统开发[D].湖北大学,2024.DOI:10.27130/d.cnki.ghubu.2024.002024.
- [27] 黄涌,蕙娟霞,关成斌.基于 BERT-BiGRU 模型的智慧医疗问答系统[J].软件工程,2024,27(03):11-14+25.DOI:10.19644/j.cnki.issn2096-1472.2024.003.003.
- [28] Anay G ,Saiyed U ,Chandra B D , et al.Analyzing deep textual facial patterns for human pain sentiment recognition system in smart healthcare framework[J].Intelligent Decision Technologies,2024,18(3):1855-1877.
- [29] Abugabah A .An intelligent medical system using MRI to detect brain tumors utilizing enhanced computational efficiency and optimized segmentation[J].The Journal of Supercomputing,2025,81(5):699-699.
- [30] Wei J ,Yan H ,Shao X , et al.A machine learning-based hybrid recommender framework for smart medical systems.[J].PeerJ. Computer science,2024,10e1880-e1880.
- [31] Wang H ,Lin A .RETRACTED ARTICLE: Nursing intervention for patients undergoing percutaneous coronary intervention with ticagrelor based on intelligent medical systems[J].Soft Computing,2023,28(suppl 2):1-1.
- [32] 刘帅.基于云平台的远程智慧医疗架构体系研究[J].无线互联科技,2022,19(12):41-43.

致 谢

白驹过隙，时光荏苒，转眼间，我的毕业设计项目“明康慧医”研究与开发的工作已接近尾声，这意味着我大学四年的宝贵时光已走到了终点。到站了，就该下车了，心中却是万般不舍。四年前，满怀热情与鸿鹄志迈入齐鲁工业大学的校门，晨夕研习，几度春秋，矢勤矢勇，甘苦共度，斯情斯景，岂能尽述！以言语已无法表达我对学校之感激。一花一草，一枝一叶，好似指掌般熟悉，母校工大没有高楼，没有大厦，但有冬季的温暖夕阳的余晖，有老师们桃李天下之喜悦，有您的学生与您难以分离的感情！回眸课堂之上，三尺讲台，老师们循循善诱，谆谆教诲，而这一切，从这一刻起，将成为我人生历史书中的一页，浓墨重彩的一页。

回望这四年，不悔选择计算机这门专业，经过工大的培养，我对计算机已愈加热爱。感谢尊敬的姜文峰老师，您用您作为计算机专业学者的知识，不厌其烦地悉心指导我毕业设计，受您指导幸甚至哉，没有您的斧正，就没有此文的成果。感谢尊敬的贾瑞祥老师、鹿文鹏老师，本科四年，贾老师是您指导我参加了多届计算机赛事，从大一的懵懂启蒙到大四的熟练掌握，您是我进步的阶梯；鹿老师，在您的激励下，我得以迈过算法竞赛的门槛，接触到高层次的技术，谢谢您；尊敬的王迪老师，我深知您科研任务繁重，但您仍百忙之中抽出时间，带领我入门前沿研究，请收下我的敬意。老师们，您们批我谬误，励我精进，恩泽之深重，我铭刻五内。

感谢我的朋友李晓语，四年，与君或探讨学理，或共解难题，劳逸相伴，患难与共，或共图算法，或并研代码，风雨同行，披星戴月之时，常有共勉之语，我难以忘怀！

感谢我的家人，自幼含辛茹苦，养我育我，倾尽心力，惟愿我前程无虞。我在大学学习知识，而您们却辛苦工作，为我百般牵挂，筑牢坚实后盾，此如山重恩，怎敢忘却？谢家父之严厉，诲我以学，导我以德，促我在浩瀚书海里学习广博知识，若无家道之严明，岂有今日之微成？

不想结束，但篇幅告诉我不得不停止了。毕业在即，离开工大学府，心怀敬畏，但又志在四方，在此毕业论文严肃之地，我谨誓，秉持诚笃之风，践行科技之志，不忘初心，不负所学，勉力不息，夙夜匪懈，以报师恩、亲恩、国恩。

是为致谢。