

# 基于AI辅助的就医问诊引擎

## 项目设计书

组长：蔡佳伟

组员：文豪 魏宸 徐浩 蔡云杉

日期：2024.10.20

版本：Version 5.0

### 修改历史

修订日期	版本号	修改人	修改内容	审核人
2024.10.16	Version 1.0	文豪	完成第一、第三、第四部分	蔡佳伟
2024.10.17	Version 2.0	徐浩	完成第二部分	蔡佳伟
2024.10.17	Version 3.0	蔡佳伟	完成第五、第六部分	魏宸
2024.10.18	Version 4.0	蔡云杉	完成第七、第八部分	蔡佳伟
2024.10.19	Version 5.0	魏宸	整体修改项目设计书	蔡佳伟

# 目录

修改历史

## 1.引言

- 1.1 项目背景
- 1.2 编写目的
- 1.3 软件系统名称
- 1.4 相关人员
- 1.5 读者对象

## 2.项目概况

- 2.1 前言介绍
- 2.2 项目介绍
- 2.3 名词介绍
- 2.4 产品成果
  - 2.4.1 产品
  - 2.4.2 手册
  - 2.4.3 服务

## 3.成本估计

- 3.1 硬件成本
  - 3.1.1 若自主训练和部署模型：
  - 3.1.2 若调用现有模型：
- 3.2 其他成本

## 4.开发技术分析

- 4.1 开发技术栈
- 4.2 技术分析

## 5.开发进度计划

## 6.角色分配

## 7.软硬件资源说明

- 7.1 服务器配置要求
- 7.2 客户端配置要求
- 7.3 软件依赖

## 8.专题计划要点

- 8.1 开发人员培训计划
- 8.2 质量保障计划
- 8.3 测试计划

# 1.引言

## 1.1 项目背景

在当今信息爆炸的时代，如何获取精准、可靠的医疗信息成为了社会广泛关注的焦点。然而，许多偏远或医疗资源匮乏的地区仍然面临医疗水平落后、医疗资源有限、就医成本高昂等问题。为了解决这一困境，我们提出了一个基于AI辅助的就医问诊引擎，以帮助大众更便捷地获取医疗服务。通过先进的人工智能技术，该平台能够快速解析患者复杂的症状信息和相关医疗数据，推荐适合的药物和治疗方案，并提供个性化的健康指导。借鉴2024年诺贝尔化学奖得主利用AI预测蛋白质结构的成功经验，我们的搜索引擎同样运用智能算法，提供最前沿、科学的医疗建议，帮助患者做出明智的医疗决策，推动医疗行业的智能化发展。

## 1.2 编写目的

编写该项目计划书的目的在于，详细说明基于AI辅助的就医问诊引擎的开发背景、技术实现、项目需求和未来发展规划，为项目开发和管理提供明确的指导依据。项目计划书旨在帮助团队成员、教师 and 同学们更好地理解本系统的设计初衷和技术目标，并为后续的项目执行、评审和优化提供依据。

## 1.3 软件系统名称

基于AI辅助的就医问诊引擎

## 1.4 相关人员

组长：蔡佳伟

组员：文豪 魏宸 徐浩 蔡云杉

## 1.5 读者对象

2024秋冬周二第3,4,5节/软件工程管理 全体同学 助教 老师

# 2.项目概况

## 2.1 前言介绍

本项目将会是将AI与医疗咨询相结合的一次有意义的尝试。目前的在线医疗知识平台，普遍存在着搜索结果不精确的情况，许多网站是以“特定疾病名称—典型症状”的教科书式的形式罗列相关信息，使得用户查询相关症状的难度加大，误判的概率也较高。将AI与相关功能相结合，可以使得用户能够在描述自身症状后得到更加精确的医疗建议，从而更加有效地活用现有的各类医疗知识信息。

## 2.2 项目介绍

本项目将会打造一个就医问诊领域的垂直领域搜索引擎。本引擎可以分析医疗数据与患者给出的复杂的不适症状，结合先进的AI技术，更加精确地分析出可能的病因、药物与治疗方案。

本项目的用户界面将以网页的形式呈现，并采取前后端分离的形式进行开发。用户通过访问指定网址的形式访问我们的服务，并在对应的网页上按照网页引导进行科目选择（内科/外科）、症状输入、用户与AI间交流、病情分析与提出建议等过程。其中，前端将采用Vue/React等动态JavaScript框架进行开发，后端将使用Django进行实现。

## 2.3 名词介绍

**AI**：即人工智能，指由人制造出来的机器所表现出来的智能，通常指用普通计算机程序来呈现人类智能的技术。在本项目中，AI模仿人类智能对症状与医疗数据进行分析，并给出尽可能精确的医疗与药物建议。

**Vue**：Vue是一款用于构建用户界面的JavaScript框架。它基于标准HTML、CSS和JavaScript构建，并提供了一套声明式的、组件化的编程模型，可以高效地开发用户界面。

**React**：React是一个支持采取声明式语法进行UI组件构建的JavaScript框架，用于基于UI组件构建用户界面。

**Django**：Django是一个使用Python写成的Web应用框架，主要目标是简化数据库驱动的网站开发，采取模型（Model）—视图（View）—模板（Template）的软件设计格式，比较适合敏捷开发。

**Git**：是一个分布式的版本控制软件，速度较快、使用灵活，适合分布式开发。本组将使用Git作为项目开发过程中各成员进行版本管理的方式。

## 2.4 产品成果

### 2.4.1 产品

本项目的产品是一款垂直领域就医问诊引擎，包括负责网页渲染与前端逻辑的前端程序、数据处理与算法支持的后端程序。产品客户在部署好相关服务器环境后分别启动前端程序与后端程序，并按照相关要求对网站进行配置。

### 2.4.2 手册

本程序的手册将包括以下内容：

- 部署说明。手册将说明运行前后端程序所需要的运行时环境（如数据库、Python引擎、JS引擎）的部署步骤，客户可以按照本部分说明将程序部署到指定的服务器中并正常启动。
- 管理说明。手册将对系统的后端配置方法进行说明，以便系统管理员进行后台维护。
- 使用说明。手册将对用户界面的操作方法进行讲解，以便用户使用本引擎。

### 2.4.3 服务

本项目将对客户提供就医问诊AI搜索引擎服务，供客户调取并返回结果给最终用户。

## 3.成本估计

### 3.1 硬件成本

为了保证AI辅助就医问诊引擎的高效运行，本项目需要配置一定的硬件资源。主要包括：

#### 3.1.1 若自主训练和部署模型：

- 服务器**：用于部署AI模型、处理大量用户请求及储存医疗数据。我们建议使用高性能云服务器，例如AWS、Azure等，能够弹性扩展，并满足AI计算的需求。
- GPU设备**：AI模型的训练和推理需要大量的计算资源，特别是深度学习模型。租赁云端GPU（如NVIDIA A100）能够显著加快训练速度，提高模型的响应效率。
- 数据存储设备**：医疗数据量较大，且需要高安全性存储和备份。采用云存储服务（如Amazon S3）来确保数据的可靠性和安全性，同时保障数据的快速读写能力。

此时预估的硬件成本取决于服务器租赁时长、数据存储需求以及计算资源消耗，随着用户量的增长，需要相应地扩展硬件配置。

#### 3.1.2 若调用现有模型：

- API调用费用**：调用现有的AI模型或第三方API服务，通常需要支付按使用量计费的API调用费用。比如使用OpenAI、Google Cloud AI或其他医疗AI平台的API，这些服务一般会按请求数、数据量或计算时长收取费用。随着用户使用频率的增加，这部分费用可能逐步上升。
- 人员成本**：由于无需开发和训练AI模型，团队的技术要求有所降低，可以集中精力于系统集成、用户界面设计、数据处理和系统维护上，但仍需专业人员管理API的调用和优化。
- 软件集成成本**：为了与现有的AI模型服务进行有效集成，可能需要额外的软件集成开发工作，确保API与系统无缝对接。这可能包括适配API接口、解析返回的结果并将其转换为用户友好的形式，以及优化调用效率。
- 合规和隐私成本**：即使调用现有模型，系统仍需处理用户的医疗数据。因此，依然需要遵守隐私保护和数据合规性标准。由于数据传输给第三方平台，隐私和安全风险增加，可能需要额外的法律合规和数据加密措施。

5. **模型优化和更新费用**：虽然模型由第三方提供，调用现有模型时，仍可能需要根据用户反馈和行业发展选择合适的模型版本或更新服务，以确保结果的准确性和先进性。

## 3.2 其他成本

**软件成本**：项目开发过程中可能涉及使用付费开发工具、库或API，例如医疗数据分析API、自然语言处理工具（如OpenAI API）等。此外，可能需要许可证来使用某些专业的医疗数据集。

**人员成本**：包括开发团队的工资、培训费用及其他支持团队的开销，例如测试人员、运维人员等。对于学术项目，这部分可能以学术支持或课程分配时间代替。

**数据获取成本**：在训练AI模型时，需要高质量的医疗数据集，这些数据集可能需要从第三方购买或通过授权获取。

**维护和升级成本**：随着项目的上线和推广，系统需要持续的维护和升级，包括软件更新、安全漏洞修复以及AI模型的再训练和优化。

**市场推广费用**：为了推广该AI辅助医疗平台，可能会涉及一定的市场营销费用，包括广告、用户引导等。

**法律和合规成本**：医疗数据涉及到高度敏感的信息，项目需要符合相关的隐私保护和数据合规标准，例如HIPAA（健康保险可携性和责任法案）等，这可能需要法律咨询和审计成本。

## 4. 开发技术分析

### 4.1 开发技术栈

**前端技术**：

- Vue.js**：Vue是一个轻量、高效的前端框架，适合构建灵活、响应式的用户界面。在本项目中，Vue.js用于设计和开发就医问诊引擎的用户界面，确保界面的快速响应和良好的用户体验。
- React.js**：React是另一个广泛使用的前端框架，具备强大的组件化开发和状态管理能力。我们可以结合React的虚拟DOM和单页面应用（SPA）的优势，用于处理复杂的用户交互，并实现高度动态化的页面。

**后端技术**：

- Django**：Django是一个高效、灵活的Python web开发框架，具有快速开发、安全性高、功能强大的特点。它为我们的就医问诊引擎提供了稳固的后端支持，用于处理用户认证、数据存储、模型调用等功能。
- Django REST Framework (DRF)**：用于创建和管理API接口，负责前后端的交互，使前端能够通过API获取医疗推荐结果和用户数据。

**版本控制**：

- Git**：我们使用Git进行版本控制，能够有效管理代码的更新和团队协作，确保代码的统一性和稳定性。通过GitHub或GitLab进行代码的托管与协作，便于团队成员同步开发进度并追踪问题。

**数据库**：

- MySQL/PostgreSQL**：作为Django的数据库后端，选择MySQL或PostgreSQL等关系型数据库来存储用户信息、历史查询记录和其他业务数据，以保证数据存储的稳定性和扩展性。

**部署技术**：

- Docker**：通过Docker进行容器化部署，确保开发环境和生产环境的一致性，提高系统的可移植性和扩展能力。

- **Nginx**: 用作反向代理服务器，提升系统的稳定性和安全性。

## 4.2 技术分析

**前后端分离架构**: 本项目采用前后端分离架构，前端使用Vue或React开发用户界面，后端采用Django提供API服务。通过这种分离架构，可以提高系统的开发效率和灵活性，同时便于前端与后端的独立开发和维护。

**组件化开发与复用**: React和Vue都具有组件化开发的特性，使得开发过程中可以将页面功能模块化，提升代码的复用性和可维护性。同时，React的虚拟DOM机制和Vue的双向数据绑定可以优化用户交互体验，确保系统的响应速度和性能。

**RESTful API设计**: 后端通过Django REST Framework为前端提供RESTful API接口，前后端通过API进行数据交互，保证了系统的可扩展性和灵活性。这一设计使得前端可以快速获取和提交数据，后台通过模型调用为用户提供实时的医疗建议和服务。

**版本控制与协作**: 通过Git的分支管理机制，项目开发团队能够并行开发不同功能模块，保证代码的稳定性和质量。Git的代码回溯功能也为后期项目的调试和优化提供了便利。

**安全性与合规性**: 在处理医疗数据时，系统需要遵循严格的隐私保护协议，如数据加密传输、用户认证和访问控制等措施，以确保用户隐私不被泄露。Django框架内置了安全性功能，如防护SQL注入、跨站请求伪造（CSRF）等，可以为系统提供基础的安全保障。

## 5.开发进度计划

项目阶段	持续时间	负责人	主要工作	输出内容
项目选定	10.8-10.22	蔡佳伟	项目进行选题，小组成员分工完成项目设计书等	项目设计书一份，开题ppt一份，开题展示一次
项目启动	10.23-10.29	文豪	准备启动项目，配置服务器，达成硬件、软件要求等	软件需求规格书一份
需求分析	10.30-11.5	蔡云杉	对项目各个需求进行解构，对各模块功能和大致实现进行分析	/
系统设计	11.6-11.13	徐浩	对项目具体实现进行分工，对框架进行适当改进，使编程实现更加容易	/
编程实现	11.14-12.10	蔡佳伟	小组成员分工，对软件进行实际开发，并且准备中期报告展示，在开发中解决实际问题	系统设计报告一份，中期展示ppt一份
系统调试	12.11-12.17	魏宸	对完成的系统进行调试，解决剩余的小问题，小组准备答辩报告，组内互评报告，小组成员准备个人部分报告	小组成员个人报告五份，答辩展示ppt一份，组内互评报告一份，系统测试报告一份

图表1：开发进度计划表

## 6.角色分配

编号	角色	角色职责	具体负责人
1	项目经理	负责整体项目的规划、进度控制和资源管理，确保项目按时交付并满足需求。	蔡佳伟
2	产品经理	负责产品需求的定义与优先级排序，确保产品符合市场需求并具有用户价值。	文豪
3	设计总监	负责指导和审核设计方案，确保产品的用户体验和视觉设计符合预期标准。	徐浩
4	测试团队经理	领导测试团队，确保软件在发布前经过充分的测试，发现并解决问题。	魏宸
5	美术监督	负责产品的视觉风格与美术方向，确保设计符合艺术标准并与产品定位一致。	蔡云杉
6	质量经理	负责监督产品开发过程中的质量控制，确保每个阶段都符合公司和行业的质量标准。	魏宸
7	开发团队	负责根据需求进行软件编码和实现功能，确保产品的功能性和技术可行性。	徐浩
8	测试团队	负责执行各种测试，发现和报告软件缺陷，确保软件质量和稳定性。	文豪
9	软件质量监督师	负责监控软件开发流程的合规性和质量，确保开发过程符合质量管理体系。	蔡云杉

图表2：角色分配表

## 7.软硬件资源说明

### 7.1 服务器配置要求

- 开发服务器：**至少8核CPU，16GB RAM，500GB SSD存储，用于开发环境的搭建和测试。
- 测试服务器：**至少4核CPU，8GB RAM，250GB SSD存储，用于软件的测试和质量保证。
- 部署服务器：**至少16核CPU，32GB RAM，1TB SSD存储，用于最终产品的部署和用户访问。

### 7.2 客户端配置要求

- 最低配置：**4核CPU，4GB RAM，100GB硬盘空间，用于基本的客户端操作。
- 推荐配置：**8核CPU，8GB RAM，500GB硬盘空间，用于流畅的用户体验和高级功能使用。

### 7.3 软件依赖



操作系统	Windows 10/11
数据库平台	MySQL 8.0+
Web 服务器	Nginx 1.18+
开发工具	Visual Studio Code, PyCharm
前端框架	Vue 3, React 17
后端框架	Django 3.2+
版本控制	Git

## 8.专题计划要点

### 8.1 开发人员培训计划

- 培训目标:** 确保团队熟悉 Vue, React, Django 等技术栈。
- 培训内容:** 包括但不限于框架基础, 高级特性和案例分析。
- 培训方式:** 自行学习线上课程, 并在团队内部进行分享。

### 8.2 质量保障计划

- 测试策略:** 单元测试, 系统测试, 用户验收测试。
- 测试工具:** Jest, pytest, Selenium。
- 代码审查:** 定期进行代码审查, 确保代码质量和一致性。

### 8.3 测试计划

- 静态测试:** 代码审查, 静态代码分析。
- 单元测试:** 针对前端组件和后端模块进行测试。
- 系统测试:** 集成所有模块进行系统级测试, 确保系统稳定性和性能。
- 用户测试:** 作为用户进行测试, 及时反馈用于优化产品。