

西安交通大学

毕业设计（论文）

题 目 基于 HIS 系统的门诊病历书写器的设计与实现

电信学部计算机 学院 计算机 专业 计算机 82 班

学生姓名 邓嘉铭

学 号 2183610318

指导教师 张亚明

设计所在单位 西安交通大学电信学部计算机学院

2022 年 6 月

摘 要

近年来，电子书写器得到了各个领域的广泛应用，但是在医疗行业，医生一直难以获得一款使用方便，上手简单，同时又能和医生的个人信息绑定的一款医疗电子系统。因此本人开发出一款新型病历编辑器系统，在本文中展示设计和实现的过程，以及测试各项功能。本文中的电子病历编辑器的主要工作和创新重点是以下三点。

首先设计了不同的编辑模式，实现了权责划分。病历编辑器的模式分为三种，设计模式，书写模式和只读模式，旨在保证权限分隔，帮助医生、设计病历的专家各司其职。其中，医生使用书写模式，调用病历模版并填写病历的内容；专家使用设计模式，调用本文提供的接口完成病历的设计。

其次设计了大量的控件，很大程度上减少了医生的工作量。控件是病历编辑器的核心功能之一，解决了两个难题。第一是给医生提供了一个电子的病历框架，医生只需要在框架里填写内容，而不需要从头写内容，第二是提供了自动纠错机制，控件内部的函数会根据医生填写的内容自动判断填写的正确与否，从而减少了医生自己查错的时间。

最后设计了完整的医疗系统，帮助医院管理患者信息。本项目是一个全栈项目，有 Web 版和 Windows 版，既可以在网页上使用，也可以安装客户端离线使用。本系统有存储病历，查看病历，调用模版，编辑模版的功能。病历模版的调用和病历的存储和查看是用后端在云服务器的 SQL Server 数据库里取出数据后，和前端交互的结果。

关 键 词： HIS 系统；智能控件；结构化存储；MVC 架构

ABSTRACT

In recent years, electronic editors have been widely used in many fields, but in the medical industry, it is difficult for doctors to get a medical electronic system that is easy to use and can be linked with their personal information. Thus, I develop a new medical record editor system, and in this paper I will show the design and implementation process, as well as test the functions. The main work and innovations of the electronic medical record editor in this paper focus on the following three points.

First of all, different editing modes are designed to realize the division of authority and responsibility. The medical record editor is divided into three modes, design mode, write mode and read-only mode, which aim to ensure the separation of authority and help doctors and the specialists who design the medical record template and the patient to do their own jobs. Doctors use the writing mode to use the medical record template and fill in the contents of the medical record; experts use the design mode to use the interface provided in this paper to complete the design of the medical record; patients use the read-only mode to read only their own medical records.

Secondly, a large number of control units are designed, which largely reduce the workload of doctors. The controls are one of the core functions of the medical record editor and solve two problems. The first one is to provide an electronic medical record framework for doctors, who only need to fill in the content in the framework instead of writing the content from scratch, and the second one is to provide an automatic error correction mechanism, in which the function inside the control will automatically determine whether the content is correct or not, therefore reducing the time for doctors to check the errors.

Thirdly, a complete medical system is designed to help hospitals manage patient information. This project is a full-stack project with Web version and Windows version, which can be used both on the web page and offline by installing the client. The system has the functions of storing medical records, reading medical records, choosing templates, and editing templates. The use of medical record templates and the storage and reading of medical records are the result of the interaction between the back-end and the front-end after handling the data from the SQL Server database in the cloud server.

KEY WORDS: HIS system; intelligent controls; structured storage; MVC structure

目 录

1 绪论.....	1
1.1 研究背景和意义.....	1
1.2 国内外研究现状.....	2
1.2.1 国外病历数据结构化存储的研究现状.....	2
1.2.2 国内病历数结构化存储的研究现状.....	2
1.3 本文主要研究工作.....	2
2 技术概述.....	4
2.1 前端技术.....	4
2.1.1 React 框架.....	4
2.1.2 富文本编辑器 UEditor.....	5
2.2 后端技术.....	5
2.2.1 Go 语言和 Gin 框架.....	6
2.2.2 MVC 三层架构.....	6
2.2.3 Microsoft SQL Server 2017 数据库.....	7
2.3 本章小结.....	7
3 系统需求分析.....	8
3.1 医疗行业病历编辑器发展现状及问题.....	8
3.1.1 医疗行业的发展现状.....	8
3.1.2 电子病历编辑器系统存在的问题.....	8
3.2 功能性需求分析.....	9
3.2.1 病历编辑功能.....	9
3.2.2 病历结构化存储功能.....	10
3.2.3 病历模板制作功能.....	10
3.2.4 部门个性化服务功能.....	10
3.2.5 样例图.....	10
3.3 本章小结.....	11
4 系统设计与实现.....	12
4.1 总体业务流程设计.....	12
4.2 功能模块设计.....	13
4.2.1 信息展示.....	13
4.2.2 编辑系统.....	13
4.2.3 病历结构化存储系统设计.....	14
4.3 数据库设计.....	14
4.3.1 数据结构设计.....	15

4.4 运行环境.....	17
4.4.1 用户界面.....	17
4.4.2 设备要求.....	17
4.5 后端的设计与实现.....	18
4.5.1 Gin 框架.....	18
4.5.2 分布式锁.....	18
4.5.3 门诊业务流程.....	19
4.6 前端的设计与实现.....	21
4.6.1 编辑器主页面.....	21
4.6.2 控件.....	21
4.7 本章小结.....	22
5 系统测试.....	24
5.1 电子病历主页面.....	24
5.2 电子病历编辑器功能.....	24
5.3 控件.....	26
5.4 本章小结.....	30
6 结论与展望.....	31
6.1 本文主要工作总结.....	31
6.2 未来工作展望.....	31
致 谢.....	33
参考文献.....	34
附 录.....	35

1 绪论

近年来，由于计算机的使用者越来越多和网络的发展壮大，各行各业对于电子编辑器系统的要求越来越高，且每个行业都需要满足各自要求的编辑器，电子编辑器的应用前景越来越好^[1]。本文中设计和实现的电子编辑器系统主要是应用于医疗行业的电子病历编辑器系统，通过和医生的多次讨论，了解了医生对功能的个性化需求，从而设计出了本文中基于 HIS 系统的电子病历书写器系统。

1.1 研究背景和意义

最早的文本编辑器只能通过终端使用，没有可视化窗口，技术人员使用一些简短的命令，例如把文本打印下来，选择编辑位置等^[2]。如果想看到文本的变化就必须要把打印出来。相对于最早用手写来完成文字的编辑，这毫无疑问是一项具有革命性的进步，标志着人类从使用纸记录内容变为使用电子设备记录内容。这项技术在很多的领域得到了广泛的应用，在很大程度上减少了纸张的使用，降低了成本，推动了可持续发展^[3]。

随着各行各业对编辑效率需求的持续上升，对可见性好，交互性优，功能全面的电子编辑器的研发需求也迫在眉睫。因此人们在 1967 年开发出了最早的全屏编辑器 O26。与此同时，在 20 世纪 70 年代末开发出来了一款新型编辑器 vi，至今还内置于 Linux 和 Unix 操作系统中，不过人们更多地使用改进版本 vim^[4]，改进的内容在于有了屏幕显示，可见性得到了极大地改善，不需要再打印出来才能看见其内容和提供了更全面的功能，改善了其交互性。

随着时代的发展，各行各业都对编辑器的个性化设计提出了更高的要求，比如在高新科技领域，写计算机程序的编辑器 IDE 就推出了很多种 (Visual Studio Code, IntelliJ IDEA, Eclipse, GoLand, Pycharm)，它们会使用脚本自动安装语言环境且提供种类繁多的个性化功能的拓展库来给技术人员提供便利。不仅如此，还可以提示技术人员一些 API 的语法，极大地减轻了工作难度。

但是在医疗行业，目前的功能仅仅是提供一个简易的模板，在 Word 中进行编辑，没有提示，没有模板管理，没有病历数据库，没有权限划分，没有清晰的示意图。这样会造成书写麻烦，模板管理混乱，历史病历丢失，权责不明，看患者病情走势不明显的问题。本文中研究的基于 HIS 系统的电子病历编辑器的期望是解决上述提出的五个问题。综上，本文设计的电子病历编辑系统需要满足以下四个特性：

- 1) 自动检查和提示，检查输入内容的正确性和提示可能需要输入的内容；
- 2) 根据身份给予权限，根据不同账户给予不同权限，权责分明；
- 3) 病历和模板有序管理，保存病历模板和患者的历史病历到数据库中；
- 4) 患者体检数据可视化，将患者的一些身体数据用图表的形式呈现；

以上四个目标对于整个需要个性化编辑器的行业都有着非常重要的科学研究意义

和实用的价值。

1.2 国内外研究现状

目前，国内电子病历系统的种类很少，功能很不完善，更不用说优化一些操作和存储。近几十年，国外特别是西方发达国家的医院建立了 HIS 病历系统，确定了以患者为中心的理念，可见国外在电子病历编辑器方面的研究暂时在一定程度上超越了我^[5]。对我国而言，电子病历编辑发展到今天面对的最大的问题之一是如何把数据进行结构化存储，目的是提高数据存储，查找的效率。

1.2.1 国外病历数据结构化存储的研究现状

最早对电子病历编辑器的结构化存储开始研究的是美国斯坦福大学的一项名为 Lore 的项目，该项目的主要研究方向是 XML 文件在数据库中的存储问题，针对 DTD 模式 XML 文档的存储和检索技术进行更加深入的研究，可以检索半结构化数据，建立了半结构化存储的数据库^[6]。法国信息与自动化研究所开展了一个项目，研究了基于 XML 的异构数据集成技术，可以把多种数据格式转化为 XML 的格式，这样方便了把其他的数据存储格式转化为 XML 进而存储到数据库，提高了电子病历编辑器的实用性和拓展性^[7]。

1.2.2 国内病历数据结构化存储的研究现状

国内虽然不如国外的病历编辑器发达但是也在逐渐发展。很多三甲医院都在自主设计病历编辑器，目标是建设数字化的医院，也有少部分顶级医院已经建成了数字化医院。

中科院软件所开发出 Xtrans 系统实现了 XML 和数据库进行双向转换^[8]。以往一般是把 XML 的数据提出，存储到数据库中，但是没有技术能把数据库的数据存储到 XML 中，但是病历的使用既需要把编辑好的病历存储到数据库中，又需要从数据库中取数据存储到 XML 中，病历编辑系统再通过 XML 来呈现历史病历。

1.3 本文主要研究工作

本文的工作目标是根据对国内电子病历编辑器发展现状的分析，找出主要的问题，从头开始设计一个电子病历编辑器系统，功能包括显示患者的基本信息，调用模板，编辑模板，查看历史病历，编辑历史病历，显示上一次的修改内容和编辑病历的医生姓名等。

本文的主要工作分为以下几部分：

1) 技术讨论

为了实现一个新的电子病历编辑器，本人学习了其他医院的电子病历编辑器，和医生进行了多次深入得到讨论，进行需求分析和系统设计，根据需求确定了编程语言、前端和后端框架、数据库等技术，最后得出结论，本文的电子病历编辑器使用 C/S 架

构、Gin 框架为后端框架、React 框架为前端框架、SQLServer 数据库、阿里云服务器、百度的 ueditor 富文本编辑器作为主要使用的技术。

2) 需求分析

通过对医院的就诊流程的观察以及和医生的讨论，总结了目前电子病历编辑器的问题，确定了本文中的电子病历编辑器的功能。

3) 系统设计

根据电子病历编辑器的功能需求，采用前后端分离的设计思想，前端用 React 为框架实现，后端用 Gin 框架和 SQL Server 数据库。

4) 系统实现

在系统设计的基础上，编程实现前端和后端，并进行各个模块的调试，保证写的代码是正确且符合要求的。

5) 系统测试

选择对电子病历系统进行测试的技术和工具，开始调试系统，调试完成后部署到云服务器上并使用特定的技术生成客户端，完成 Web 版本和客户端版本，方便了客户的使用和未来的维护和更新。

2 关键技术

本章中首先介绍前端框架 React、百度的 Ueditor 富文本编辑器、后端框架 Gin、SQLServer 数据库，介绍清楚后说明使用这些技术的原因和意义。

2.1 前端技术

为了使医生和患者可以随时随地查看的病历，本文设计的电子病历系统可以通过浏览器进行访问，因此前端技术非常重要。

本文选择了使用 React 框架作为前端框架，因为这个框架是目前最主流的前端框架，技术非常成熟和可靠，而且可以实现本文中电子病历编辑器的一切需求。使用百度的富文本编辑器 Ueditor 的目的是当做电子病历编辑器的文本编辑原型，因为这项技术同样也是一个广受追捧的成熟技术，而且和 Word 的使用方式一致，又提供了足够的接口可以进行个性化的修改，因此本文在 Ueditor 的基础上设计了满足电子病历编辑器的需求的样式。

2.1.1 React 框架

React 是一套非常前沿的前端框架，主要用于完成前端设计模式。在 2015 年一项针对前端框架的调查表明，React 有着很高的开发者满意度^[9]。

React 的设计重视在视图上的工作，只需要调用 React 提供的 API 就可以完成数据和组件的绑定，因此比很多框架在使用上都更加方便和快捷。

即使 JQuery 也可以完成此电子病历编辑器，但是本文仍然选择使用 React 的原因是 JQuery 是用封装好的各种 DOM 方法来操作 HTML 的 DOM 结构而没有实现数据和组件的绑定，会使得在做复杂的网页时写代码会非常复杂，网页也会很卡。简单来说网页是通过组件的排列和叠加组成了基本的视图结构，通过 CSS 修饰，让视图更加柔和美观，看起来更舒服自然，最后通过 JavaScript 来接收用户的请求并返回对应的响应。如果把数据和视图放在一起，就像传统的 HTML 一样，处理起来是很麻烦和费劲的。因为几个结构间可能因为数据而产生联系。例如一个音乐播放器，进度条的时间和音乐播放的时间应该是一样的，但是如果手动波动进度条，那么由于数据混杂，可能会造成音乐播放和进度条的不同步问题，而且写 JQuery 会非常长，十分麻烦。

但是使用 React 可以很好地解决这个问题，因为有声明式，可以完成数据和组件的绑定。这样有两个好处。第一可以完成数据的实时变化，很多时候为了响应请求，网页需要刷新一次才能获得更新的数据，但是 React 的实时响应能在客户发出请求后立即给出响应，能让用户获得更好的体验和更好地留出活跃用户。第二可以控件的状态和数据绑定在一起，这样更新控件只需要更新数据源，这样既可节省更新控件代码的时间也可以完成同一个数据在不同控件的同步更新，不会造成数据不一致和为了保证同

步性而写出非常复杂的代码的问题。

2.1.2 富文本编辑器 UEditor

富文本编辑器是可以对文字、图片等进行编辑的产品。随着需求的日益增长和技术的大幅度进步，富文本编辑器的功能已经不局限于文字和图片，还包括了视频、表格、代码块、公式、附件、XML 等高级功能^[10]。本文使用的富文本编辑器，实际上就是利用了现有的封装好了的、技术成熟的富文本编辑器，这样可以节约开发者的时间，把主要精力都用在新技术的研发和实现而不是纠结于已经成熟的技术的开发。

本文的电子病历编辑器使用的富文本编辑器是 UEditor。UEditor 推出后由于很多特性都做的很出色而且是一个轻量级的编辑器，受到了人们的欢迎和追捧，可以根据用户的需求进行个性化定制，因此用户体验十分不错，并且这是一款免费的开源编辑器，从 Github 上可以直接获得其源代码和文档。不仅如此，这款编辑器的功能非常全面，插件也很多，而且可以很方便地和其他百度的产品相结合，稳定性十分强，接入也非常简单快捷且支持不同的环境。总来的说，UEditor 稳定性强，功能齐全，和 Word 的操作相似，开源免费，文档详细，而且是本国企业研发的产品，不会面临西方的“卡脖子”难题。综上，选择 UEditor 既可以方便开发人员的开发又对用户的上手使用十分友好，所以本文中电子病历编辑器选择了 UEditor 作为富文本编辑器。Ueditor 的界面如图 2-1 所示。

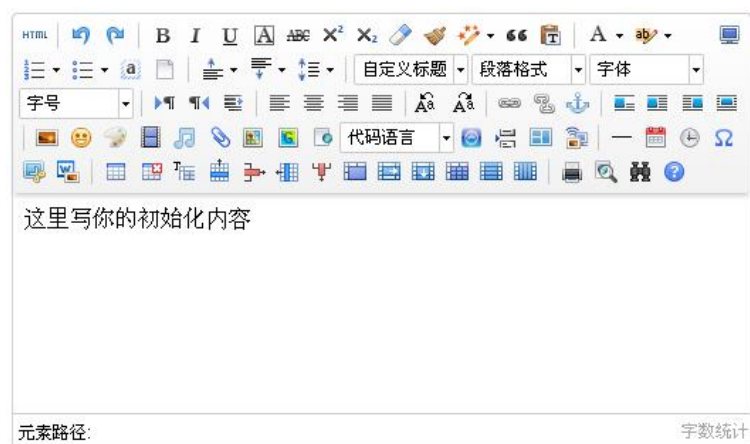


图 2-1 Ueditor 示意图

2.2 后端技术

本节主要介绍后端的技术，后端的技术在一个项目中的非常关键的，因为前端只是展示数据和给用户交互，而后端是负责提供展示的数据和确定交互的逻辑。

本文选择了使用 Go 语言作为后端的语言，因为这个语言虽然开发出来的时间不算很长，但是已经产生了很大的影响，在腾讯里 Go 语言的使用已经成了主流、成熟和可靠的编程语言，有着成熟的框架 Web 应用开发框架 Gin，可以满足完成电子病历编辑器的需求。使用 MVC 三层架构当作后端的架构，这同样也是一个广受追捧的成熟技术，

本文把后端的业务逻辑分为控制层、服务层和实体层，分别负责业务逻辑的实现，每个服务的具体实现以及和数据库的交互。使用 Micorsoft SQL Server 是因为总的来说 Micorsoft SQL Server 是一个很成熟稳定的数据库，使用起来方便简单，而且和 Windows 系统一样都是 Micorsoft 开发的，因此在同一家公司开发的操作系统上的切合度更好，稳定性更高，是不二之选。

2.2.1 Go 语言和 Gin 框架

Go 语言起源于二十世纪初，由 Google 正式对外宣布发布。Go 的主要目标是既有 python 这样的脚本语言的快速开发的优势，又有和 Java、C++ 等需要编译的编程语言一样的效率，Go 在并发上做了很大的改进，相对其他语言可以用较短的代码实现效率很高的并发^[1]，不仅如此，Go 还是一门开源的语言。

选择 Go 语言作为本次项目的语言的主要原因如下：

1) 对并发型的支持很好。电子病历编辑系统对并发的要求比较高，因此本文选择语言的时候希望语言在并发编程的时候能尽可能简洁，而 Go 语言完美地符合这个条件，Go 的并发执行单位是一种称为 goroutine 的协程，是在其他语言的并发控制的设计理念的基础上增加了很多的改进而成的。

2) 可应用于搭载 Web 服务器。本项目是需要做 Web 端的，所以 Go 的可应用于搭载 Web 服务器的功能非常重要。Go 应对大量事件时表现良好，可以完成并发支持的任务，当大量用户访问一个网页时，使用 Go 的网页不容易崩掉。

3) Gin 框架是用 Go 语言开发的 Web 应用后端框架，是广受欢迎的后端框架之一，由于路由相应速度很快以及支持 JSON 解析和验证，非常适合本项目的开发。

2.2.2 MVC 三层架构

目前，MVC 框架在当今工业界应用十分广泛，为了能够应对 GUI 类型的应用的拓展将程序划分为以下三层，如图 2-2 所示。

- 1) 控制层 (Controller)：处理前端发来的请求，并返回对应的数据。
- 2) 视图层 (View)：前端的一部分，是对视图界面的设计。
- 3) 实体层 (Model)：程序员对接口的实现以及和数据库的交互。

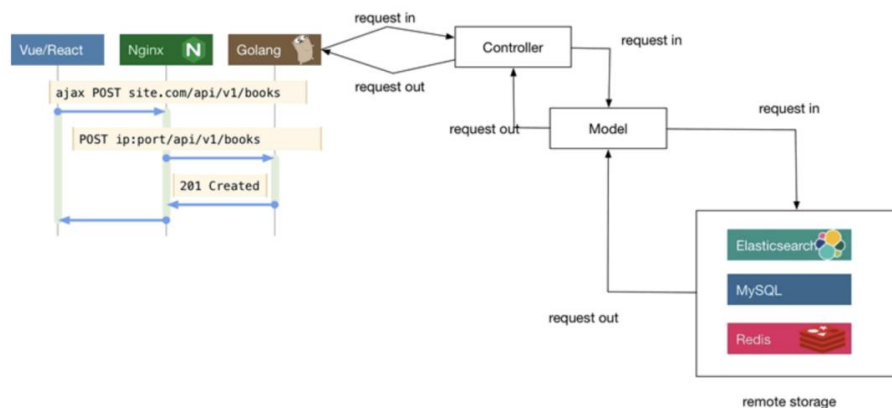


图 2-2 MVC 三层框架逻辑图

其中视图层被放到前端，而后端在本项目中细分为控制层、服务层和实体层，把接口的具体实现从实体层放到服务层中，也就是本项目的后端分为以下三层：

1) 控制层 (Controller)：处理前端发来的请求，并返回对应的数据，换句话说就是调用服务层提供的接口，按照业务逻辑完成业务。

2) 服务层 (Service)：负责功能接口的实现，本层就是提供接口和实现接口的内容，为了让控制层更高地专注于业务逻辑设计，把具体功能放到本层来实现。

3) 实体层 (Model)：负责和数据库交互，服务层会调用本层的接口来调用数据库的数据和存储数据到数据库中。

综上，本项目的后端也分为三层来实现，这样相当于把一个很困难的任务分割为多个小任务，如果遇到了问题需要维护的时候，可以找到具体在哪层出现了问题，方便技术人员更快地定位问题和解决问题，因此最终选择了三层架构来完成本文的电子病历编辑器。

2.2.3 Microsoft SQL Server 2017 数据库

目前，大多数医院采用的是 Windows 操作系统，作为同一家公司的产品，Microsoft SQL Server 和 Windows 系统由同一家公司研发，因此在 Windows 系统上运行 Microsoft SQL Server 的稳定性比其他平台更高，整体的结合程度也很好。不仅如此，Microsoft SQL Server 的使用难度并不高，比主流的数据库 MongoDB 和 MySQL 要简单不少。因此无论是软件工程师还是医生都会觉得这是一个合适的选择，所以最终选择了 Microsoft SQL Server。Microsoft SQL Server 2017 虽然不如 Microsoft SQL Server 2021 的特性多，但是运行起来很稳定。主要的优点如下：

1) 索引服务。SQL Server 数据库有 Columnstore 索引功能，为此在进行数据的处理方面，能提高数据的查找效率。

2) 自定义服务器权限服务。可以修改服务器的权限，提供更多个性化的设置接口。

3) 恢复服务。SQL Server 数据库有 AlwaysOn 功能，可以实现对数据库镜像功能改进，数据库出现破坏的时候可以对单组数据或全部数据进行恢复。

4) 回写式告诉缓存服务。SQL Server 数据库有 WriteBack Cache 功能，可以保留很可观的物理内存，在写密集的操作中，可以减少 I/O 提高效率和减少卡顿。

2.3 本章小结

本章中先是介绍了前端的技术框架和富文本编辑，接着介绍了后端的编程语言、框架、架构、数据库，最终确定出了前端和后端使用的语言和技术框架：

1) 通过对病历展示的分析 and 病历数据存储的需求，确定了使用 React 前端框架和富文本编辑器 UEditor。

2) 通过对医院的门诊住院的业务逻辑和病历数据的存储分析，确定了使用 Go 编程语言、Gin 框架和 MVC 三层架构以及 SQL Server 数据库。

3 需求分析

目前，国内大多数病历编辑器系统都是非结构化存储的，而且没有自动纠错和检查功能。本文重点从行业现状到医院信息化现状开始，对新型电子病历编辑器系统的需求进行了深入的分析，并把需求分析结果进行总结，为下一步的系统设计打下坚实的基础。

3.1 医疗行业病历编辑器发展现状及问题

本节重点介绍国内医疗行业的发展现状和主要存在问题，我国现阶段的医疗行业，特别是非一线城市的三甲医院，主要以非结构化病历为主，病历编辑器的发展停滞不前，和国外发达国家的成熟病历编辑器系统的差距较大。本节完成医疗行业的电子病历编辑器系统的发展现状调查，经过和医生的讨论和思考得出现阶段对电子病历编辑器的需求，为下一节的系统设计打下坚实的基础。

3.1.1 医疗行业的发展现状

现阶段，大多数医院的信息系统也就是电子病历编辑器系统的发展仍不成熟，功能十分有限，但是优点是已经基本实现了以患者为中心的系统，这点和国际上是很接近的。

以患者为中心的系统需要具有记录患者在医院内进行的所有医疗的历史病历的功能，而这个也是我国电子病历编辑器的核心问题所在。一个完整的病历编辑器系统，需要兼备病历的书写和保存，历史病历的管理和控制，以及医生账号信息的保存管理以及权限分割，以及要满足使用的简易性和清晰性。只有具有了这些功能，电子病历编辑器才会更好地适应时代的发展和医疗行业对电子病历编辑器的越来越高的要求，在医疗行业做好了电子病历编辑器系统，同时也可以在乎民生根本的医疗系统中减少欧美的卡脖子问题，增加我国的社会凝聚力和民族的自信心，可谓是一举多得的事，也是医疗行业电子病历编辑器系统的大势所趋。

3.1.2 电子病历编辑器系统存在的问题

目前，我国的电子病历编辑器系统存在的第一个大的问题是系统不统一。我国的医疗行业的病历编辑器系统一直都没有一个国家标准，所以人们每次去不同的医院就诊会面对不同的医疗信息系统，比如去三甲医院，往往能得到比较好的体验，去小城的普通诊所，往往体验不佳，甚至没有现代化的医疗信息系统。系统不统一的问题会造成不同的医院发展的很不均衡，患者的就医质量得不到保障。不仅如此，各个医院都在自己发展自己的医疗病历编辑系统，没有合作和共赢。因为标准不统一，合作的难度非常大，各个医院的设计标准和技术都有很大的区别。

第二个主要问题是系统的功能不全面。现阶段，大部分医院都没有自主研发的电

子病历编辑器系统，但是对电子病历编辑器系统的需求又与日俱增，因此很多医院会购买粗糙的电子病历编辑器系统来使用，特别是小城的医院。虽然这项举措解决了燃眉之急但是却不能从根本上解决问题，当然同时也造成了居民就诊的不便利和医生写病历和医院管理病历的不方便。我国目前大部分医院的电子病历编辑器能完成的功能比较单一，仅可以完成简易的病历书写及打印功和一些基本的权限控制。其余电子病历编辑器的功能均无法实现，这导致了大量的临床需求无发得到满足，医院也迫切希望通过更新电子病历编辑器系统来满足医院对就诊、减少数据存储控件和科研等方面的需求。

第三个主要问题是系统功能没有侧重点。目前，医院的电子病历编辑器只能满足基本的需求，但是很多医院都在某个领域有深入的研究领域和高超的医疗水平，而电子病历编辑系统却没有提供一些个性化的功能来满足各个医院在某个领域里的进一步科研需求。

3.2 功能性需求分析

新型电子病历编辑器系统的建设是一个复杂的工程。需要有明确且丰富的文档支持才能设计出一个功能全面的系统，因此需求分析的工作是非常关键的。本人的导师特意请了一位信息技术比较好的临床医生来和本人一起充分调研和考察了当前医生所在医院以及国内主流病历编辑器的优缺点，结合医院的现状，充分考察了国家的政策和法律，以及未来医院在发展过程中可能遇到的问题和困难，最后，在我们的多次讨论和功能性需求分析后得到了新型电子病历编辑器对功能的需求，帮助下一阶段系统设计打下了基础。

3.2.1 病历编辑功能

目前，我国的电子病历编辑器主要的问题是编写病历的编辑器的智能程度不够，具体表现在编辑器不能自动检查医生写的内容是否正确，体现在填写的内容是否符合规范和填写的内容是否犯了明显的错误，例如系统中这位患者的性别是女，填写的内容是男；本模块内容应该填写一个日期，而日期的填写是有一定格式要求的。经过和医生的讨论，得出结论解决这个问题的办法是，增加控件功能。电子病历编辑器中，提供给医生填写的每个单元都应该有一个内置的检查函数，功能是检查医生填写的内容不发生系统可以自动判断的错误。

控件的功能设计应该不止于此，为了提高医生写病历的速度和减少医生的负担，病历编辑器中应该增加自动提示医生填写内容的功能，例如填写日期，当医生点击填写区域时自动弹出一个日历，医生点击日历上的日期就完成了日期的填写；填写性别，点击区域时自动弹出男和女两个选择按钮，医生只需要选择一个；医生的主要精力可以不花在填写患者的基本信息填写中，而是花在病情的判断中和医疗方案的填写上。

如果患者不是第一次来就医，而是在系统中有历史病历，医生可以调取患者以往的病史，查看患者的患病记录，并且可以直接打开患者的历史病历，接着编辑本次的

病情，可以直接跳过填写患者基本信息的环节，更能很大程度上节约医生和患者的时间。

3.2.2 病历结构化存储功能

在上文的技术分析中设计一个新的电子病历编辑器系统的初衷就是为了解决病历的结构化存储问题，和发达国家的标准接轨，把我国的病历编辑器技术提升一个档次。因此病历的展示和存储的功能的设计尤其重要。结构化存储的另一个意义是把数据存储到数据库中很大程度上节省了物理空间。以前每张病历都需要以 word 文件的形式存储它的格式和内容，占物理空间较大，本文的电子病历编辑器把医生填写的数据存储在数据库中，格式以病历模版的形式存储在数据库中，每份病历模版都可以重复使用，因此省下了格式占有的物理空间。不仅如此，把数据存入数据库的另一个好处是方便技术人员调用数据进行机器学习算法的训练和测试，为将来设计给医生更佳体验的电子病历编辑器打下一定的基础。

3.2.3 病历模板制作功能

电子病历编辑系统不仅包含病历编辑器的功能，还需要兼备模板制作的功能，模板需要分不同的科室来制作，保持一定的个性化能更好地服务不同科室的医生。制作模板需要用到编辑器的基本功能，例如插入文字，删除文字，制作表格等。不仅如此，还需要使用控件。控件是电子病历编辑器中核心的部件，是填写病历中有共性的数据的组件，例如填写患者编号、姓名、性别、入院日期、症状等信息，我把它们分为复选框、文本输入框、下拉框、日期选择框这四类主要控件。

3.2.4 部门个性化服务功能

通过前面的分析和与医生的讨论，需要面对不同需求的患者提供不同的服务，诊断流程分为门诊和住院两大类。所有患者来到医院前台挂号后，医生给患者诊断的过程就是门诊，这一阶段只使用电子病历编辑系统的门诊部分，如果患者需要住院，那么接下来的医疗过程就由住院系统部分来负责。这样能有效区分门诊和住院的患者，而且因为不同种类的患者在具体的医疗上和付费上有不同的需求，因此，本文设计的电子病历编辑系统，把门诊和住院分开。

3.2.5 样例图

通过上文的讨论，我们得到了对电子病历编辑器的功能性需求，总结为两张样例图，分别是医生用户的样例图和设计病历的技术人员的样例图，经过上文的分析得出医生用户需要的功能是查看和编辑病历模版，查看和编辑历史病历，查看患者基本信息，查看上一次修改记录，保存和提交病历；设计人员需要的功能是查看和编辑所有病历模版，调用控件来设计病历模版，调用 UEditor 的基本编辑功能来设计病历模版，保存和提交病历模版，如图 3-1 所示。

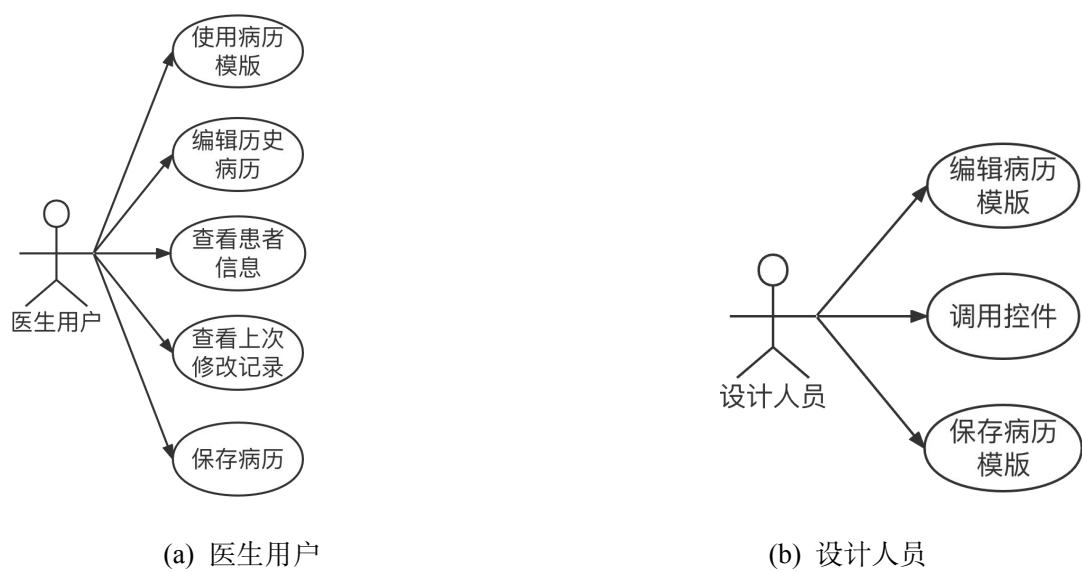


图 3-1 医生和设计人员需求分析样例图

3.3 本章小结

经过上面二个小组的讨论，本文得出了现阶段国内电子病历编辑器系统的问题和需求，现总结如下：

1) 我国电子病历编辑器的问题：通过和医生的讨论和信息的收集，经过上文的分析，得到了系统的标准不统一，功能不全面，功能没有侧重点三个主要的问题，找到了我国的电子病历编辑器系统的需求。

2) 我国电子病历编辑器的需求：根据需求的功能划分，本文得出了四个主要的需求，病历智能编辑，主要有对编辑内容的正确性的检查和利用选项填充内容；病历的结构化存储，主要是从以往的利用 word 来存储病历改为存储病历到数据库中，实现结构化存储；制作模板，提供编辑器的基本功能和控件，帮助设计人员快速设计出满足多个科室要求的病历模板；个性化服务，主要是把对患者的看病服务分为门诊和住院两大类，因为两类服务的收费是不同的，而且需要记录的患者信息也不同，提供给患者的服务也有差异。

4 系统设计与实现

本章的主要内容是电子病历编辑器的系统设计理念和具体的各个模块的系统设计。系统设计是工程开发中非常重要的一个环节，包括总体架构的设计，功能模块的设计和数据库的设计，是在需求分析的基础上，把需求落到实现的重要环节。在本章的电子病历编辑器系统中，由于牵涉到很多功能模块的调用和流程的处理，考虑到使用者的便捷程度，统一采取用首页导航的操作界面，在界面左侧是一个树形的菜单栏，最上面显示患者的基本信息，包括姓名，年龄，性别，挂号的 UID 等等基本信息，下面有两个树形的拉条，第一个是患者的历史病历和医生新建的病历，第二个是本科室的所有病历模版。右边是一块比较大的区域，作用是显示医生点开的病历和病历模版，最后侧显示留痕，记录上一次编辑病历的医生名字和编辑内容。本章主要对部分重点和有意义的功能模块做出一些关键的展示。

4.1 总体业务流程设计

这一节主要介绍电子病历编辑器系统的总体业务流程设计。电子病历编辑器系统的总体业务流程分为以下二种业务：

- 1) 设计模板。设计人员在系统中制作模板并把模板上传到数据库中。
- 2) 医生诊断。医生打开一个可用的模板或患者的历史病历，编辑完成后尝试提交，如果没有错误则提交成功，否则提交失败，系统自动把病历的数据导出并存储到数据库中，完成结构化存储，如图 4-1 所示。

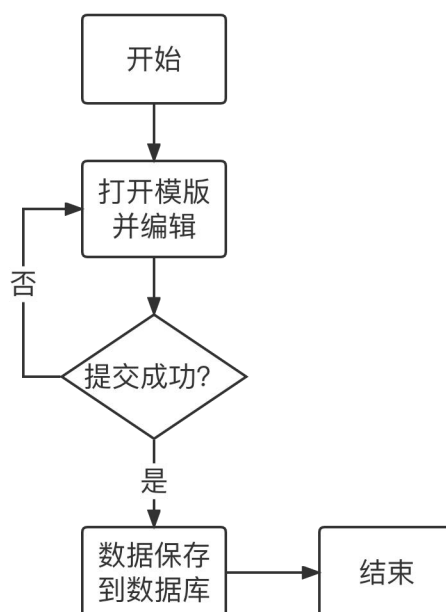


图 4-1 业务流程图

4.2 功能模块设计

这一节主要是介绍电子病历编辑器的各个功能模块的系统设计，是整个系统设计环节中最重要的一部分。功能模块主要分为两个部分，即病历编辑系统和病历存储系统。功能模块结构设计图，如图 4-2 所示。

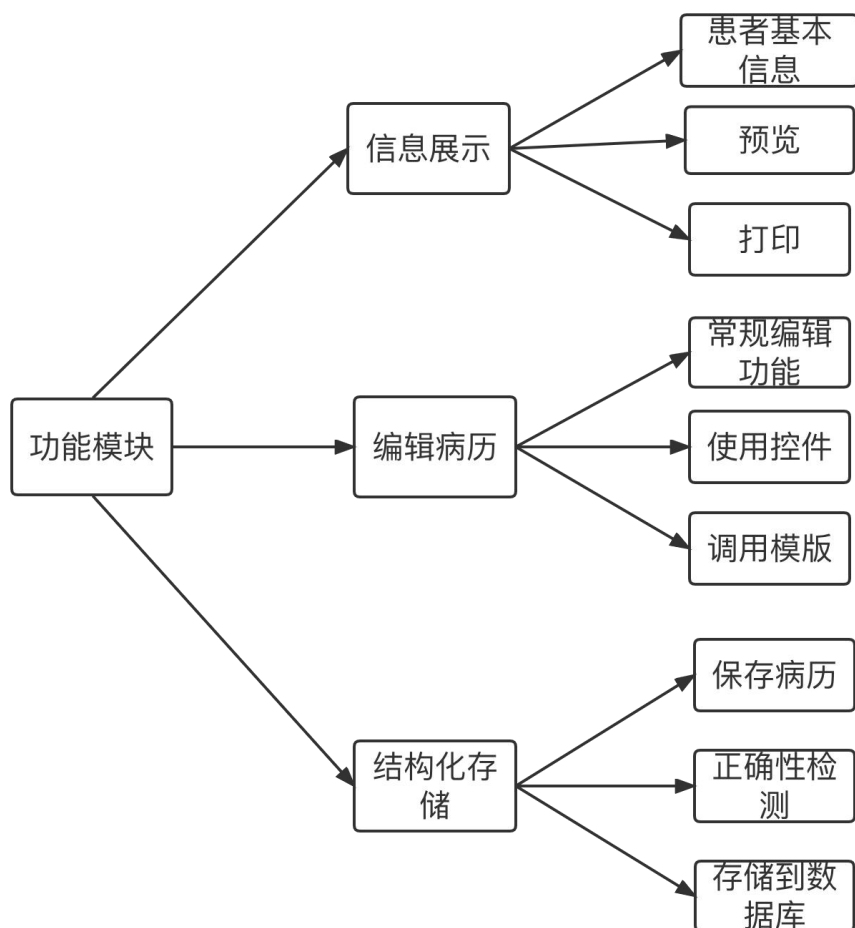


图 4-2 功能模块设计图

4.2.1 信息展示

信息展示是电子编辑器的重要功能，通过门诊挂号序号参数向后端请求患者的信息，并在前端展示出患者的基本信息、历史病历和病历模板。不仅如此，本系统还可以通过 Ueditor 提供写好的病历预览功能和通过 windows 系统的打印接口提供打印病历的功能，实现电子和实体的信息展示。

4.2.2 编辑系统

编辑系统是整个系统设计中非常重要的一环，编辑系统是记录患者基本信息和医生针对患者的临床症状做出的诊断的系统。编辑系统包括医生调用电子病历模版库中的模版和在电子病历编辑器中显示模版以及在电子病历编辑器中写病历，还包括调取

历史病历，保存病历和打印病历等诸多功能。

1. 诊断结果录入

1) 录入需要遵循以几点规则：如果是患者有同样的疾病历史病历，则在历史病历中增加最新的病状以及治疗方案，如果没有，则在新建的病历中写上病状和治疗方案。如果需要住院，则必须写上建议住院治疗，后续可能以此为依据，从门诊系统转入住院系统。

2) 保存病历：在医生填写好患者的个人相关信息后，通过点击给定的保存按键，系统自动检查编辑的内容是否正确，如果正确则会自动把将相应的信息录入数据库中，否则提示医生输入错误的位置，不保存数据到数据库中。

3) 提交病历：在完成病历后，保存完成后就自动把病历提交到系统中了，医生不需要额外的操作。

2. 查看病历

1) 患者有权利查看自己的病历，但是没有权限查看别人的病历。

2) 医生可以查看自己负责的全部患者的病历，但无权查看其他患者的病历。

3. 打印病历

1) 保存的病历可以在系统的左边菜单中查看，接下来需要选中病历。

2) 在查看完病历并确认病历的内容无误后，点击打印按键即可打印。

3) 打印完成后，查看打印内容，最后确认一次保证没有问题。

4.2.3 病历结构化存储系统设计

存储系统是病历编辑器系统中很重要的一部分，因为结构化存储病历是本系统相对国内主流的非机构化病历存储系统是一项重要的技术进步，同时当然也是本病历编辑器系统的重点技术难点。本文中病历编辑器的存储系统的实现主要是靠后端实现的，技术主要是使用 JavaScript 的 DOMParser 函数把病历的数据导入 XML 中，后端会接收 XML 文件，把 XML 的数据用自动解析后存到数据库中；需要打开病历查看时，前端发送请求后，后端会从数据库中取出数据，自动生成 XML 文件，前端会利用 DOMParser 函数解析 XML 文件为 HTML 文件并渲染。

4.3 数据库设计

数据库是为了解决系统研发对数据存储和查询的需求，可以提供多种接口来对数据进行一些规范操作的技术。明确使用数据库的类型和版本以及数据库表的设计，是数据库设计的重点。本文中的数据库选择 Microsoft SQL Server 数据库，因为 Microsoft SQL Server 和大多数医院的主流操作系统 Windows 系统都出自于同一家企业 Microsoft，使用起来的稳定性较好，而且操作简单，搭载方便，对软件工程师和使用者的都很方便。

4.3.1 数据结构设计

数据结构设计分为四个模块，分别是门诊病历模板模块、历史病历记录模块、门诊挂号记录模块、分布式锁模块。

1) 门诊病历模版表

电子病历模版表存储电子病历模版的信息，模板是制作病历的基础，每一份病历都是在模板的基础上填充信息获得，包括模版名称、模版内容、创建人、部门、使用类型、创建时间、状态、模版编号、备注等信息，如表 4-1 所示。

表 4-1 门诊病历模板表

字段名称	数据类型	字段说明
template_name	nvarchar(50)	模板名称（主键）
template	nvarchar(MAX)	模板内容
creator	nvarchar(50)	创建人
department	nvarchar(50)	部门
usage_type	nvarchar(50)	使用类型
creation_time	datetime	创建时间
status	nvarchar(50)	状态
template_no	varchar(20)	模板编号
comment	nvarchar(50)	备注

2) 门诊历史病历记录表

历史病历模块存储患者的历史病历，每份病历有且仅有一份，因此把门诊挂号序号选作主键，包括门诊挂号序号、患者编号、记录类型、记录内容、姓名、性别、出生时间、就诊科室、电话、最后修改人、最后修改时间、修改日志等信息，如表 4-2 所示。

表 4-2 门诊历史病历记录表

字段名称	数据类型	字段说明
mzghxh	char(12)	门诊挂号序号（主键）
cdno	varchar(24)	患者编号
record_type	nvarchar(50)	记录类型

表 4-2 （续）

字段名称	数据类型	字段说明
xm	nvarchar(50)	姓名
xb	char(1)	性别
cssj	datetime	出生时间
jzks	varchar(10)	就诊科室
tel	varchar(20)	电话
update_by	char(5)	最后修改者
update_time	datetime	最后修改时间
change_log	nvarchar(MAX)	修改日志
record	nvarchar(MAX)	病历内容 (.html)
record_xml	nvarchar(MAX)	病历内容 (.xml)

3) 门诊挂号记录表

门诊挂号信息记录表存储挂号的患者的个人信息，包括门诊挂号序号（主键）、姓名、科别、出生时间、年龄、科别、挂号日期、患者编号、身份证号码、电话、录入人员、诊断类别等信息，如表 4-3 所示。

表 4-3 门诊挂号记录表

字段名称	数据类型	字段说明
mzghxh	char(12)	门诊挂号序号（主键）
xm	nvarchar(50)	姓名
xb	char(1)	性别代码
cssj	datetime	出生时间
nl	int	年龄
kb	varchar(10)	科别
df	char(10)	大夫
rq	datetime	挂号日期
cdno	varchar(24)	患者编号

表 4-3 (续)

字段名称	数据类型	字段说明
sfzhm	varchar(20)	身份证号码
class	nvarchar(20)	家庭住址
tel	varchar(20)	电话
lry	varchar(10)	录入员
zdlb	char(1)	诊断类别

4) 分布式锁表

为了确保在同一时间只有一位医生拥有编辑病例的权限，本文使用分布式锁表来记录锁的归属信息，如表 4-4 所示。

表 4-4 分布式锁表

字段名称	数据类型	字段说明
mzghxh	char(12)	门诊挂号序号（主键）
owner	nvarchar(50)	医生姓名
lock_time	datetime	加锁时间

4.4 运行环境

4.4.1 用户界面

本文的电子病历编辑器系统是支持 Web 端也就是在浏览器上通过 URL 访问，也支持客户端也就是在 Windows10 操作系统进行使用。用户界面都是左侧为一个树形的下拉框系统，其中保存着病历和病历模版，右侧是显示病历和病历模版的编辑器部分。编辑器部分在输入上的操作和 Word 一模一样，很大程度上减少了医生学习新的编辑器需要的时间，大大提高输入病历的效率，而且简洁高效。

4.4.2 设备要求

本系统在阿里云服务器上开发，通过前后端分离的架构来设计，数据库使用 Microsoft SQL Server，设备需要满足以下要求：

1) 硬件要求

CPU: Intel 要求 i5 以上，AMD 要求 R5 以上

GPU: 无要求

2) 软件要求

操作系统: Windows 10

开发工具: Microsoft Visual Studio Code

编译环境: Go 语言, Node.js

数据库: Microsoft SQL Server 2017

服务器: 阿里云服务器

4.5 后端的设计与实现

4.5.1 Gin 框架

本系统的后端通过 Go 语言和 Gin 框架实现,作用是通过后端函数的处理正确地返回前端请求的内容,主要的处理的业务是病历模板管理、门诊病历管理和门诊信息管理。具体的设计流程分为设计数据库、路由管理和接口设计。

1) 设计数据库: 使用 yaml 文件进行数据库的账号密码以及云端 ip 的配置,并进行初始化测试。

2) 路由管理: 使用 Gin 框架来进行路由管理。Gin 封装了 Go 语言的网络操作相关的语句并提供给用户简洁的接口。首先使用 gin.Default()生成 Engine 对象,接着调用 Engine 的 Use()方法加载自定义跨域请求,配置了 HTTP 请求等信息。最后调用 Engine 的 Group()函数创建路由组,配置函数和路由的映射。路由的定义如表 4-5 所示。

表 4-5 路由定义

HTTP 请求	路由	说明
GET	/outpatient/login	登录并获取门诊信息
GET	/outpatient/logout	退出登陆停止门诊
POST	/record-template	增加病历模板
GET	/record-template	获取病历模板
POST	/medical-record/update	更新门诊病历
POST	/medical-record/insert	增加门诊病历

4.5.2 分布式锁

分布式锁的作用是当多个线程同时访问同一个变量,可能同时修改这个变量,但是该变量最终的值是最后一个线程提交的值,因此会出现脏读和不可重复读等问题。在电子病历编辑器中,如果同一个病历同时被多个医生打开并修改,那么医生同时修改病历,后面提交的医生会覆盖前面医生写的内容,这样有可能会造成严重的医疗纠

纷，因此需要使用分布式锁来保证每次只有一个医生能书写病历。

在电子病历编辑器系统中，医生使用的都是自己的电脑，而后端也是在云端部署的，通过在数据库中增加一张表 `emr_dist_lock` 来引入分布式锁，这张表的记录门诊挂号序号、医生、加锁时间其中门诊挂号序号是主键。业务流程是，每当医生请求打开并编辑病历时，尝试在表中执行 `Insert` 语句，由于键是唯一的，只能允许第一个请求的医生操作成功，其他医生操作失败。如果操作失败，医生也可以进入病历的编辑界面，但是会给医生提示：有其他医生正在编辑，并且设置编辑模式为只读模式。当医生结束编辑后，点击关闭界面的就会自动触发删除数据库的这条语句的操作，从而释放锁，其他医生可以获取锁，开始新的编辑。具体逻辑如图 4-3 所示。

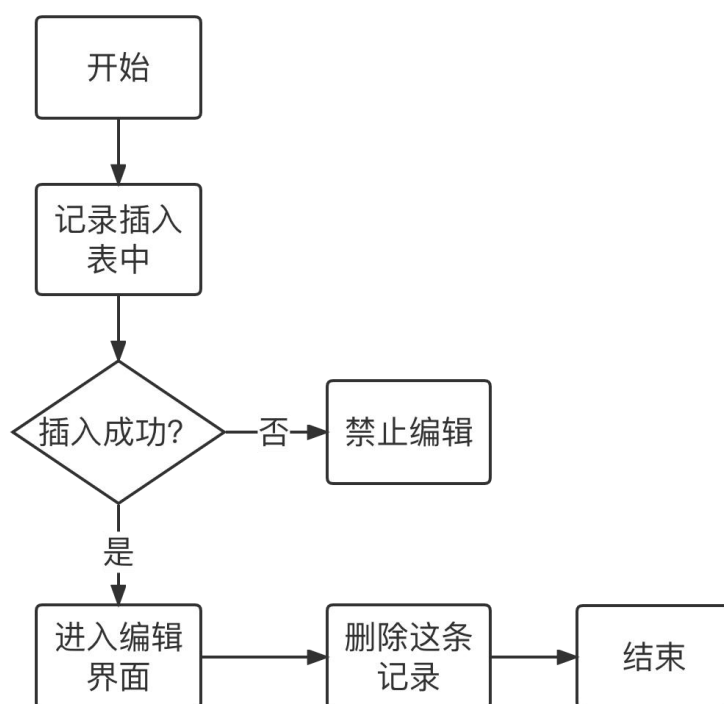


图 4-3 分布式锁实现原理

4.5.3 门诊业务流程

1. 挂号

患者在前台挂号，录入个人信息，系统自动生成唯一的门诊挂号序号，在门诊挂号表中插入一条记录，包括门诊挂号序号（主键）和患者录入的信息，包括患者姓名、性别、年龄、籍贯、民族、住址等。

2. 诊断

医生经过诊断得到患者的病理信息和病情判断，得到诊断信息、疗程信息和费用信息。

3. 基本信息录入

1) 在初步判断患者的病情后调用相应的电子病历模版。

2) 录入基本信息, 基本信息种类繁多, 总结如表 4-6 所示。

表 4-6 患者基本信息分类

基本信息类别	信息内容说明
患者信息	患者姓名、编号、年龄、性别、身份证号、籍贯、住址等
医生信息	主治医生姓名、住院医生姓名、护士姓名、科别等
诊断信息	诊断时间、损伤或病症的专业名词等
疗程信息	用药的学名、用药的数量、用药的方式、手术名称等
住院信息	入院途径、治疗类别、入院科别、入院病房等
费用信息	挂号费用、影像学诊断费、临床物理治疗费、手术费等

3) 当信息填写完成需要导入数据库时, 前端通过 POST 请求把这些信息整理为 JSON 形式的字符串, 如果信息为空则对应为空字符串, 把信息传到后端请求更新数据, 后端通过处理 JSON 得到对应信息, 根据 POST 请求对应的表把数据插入数据库中。

4) 进一步确定是否有住院需求, 如果有则转到住院业务流程模块。详细的业务流程图如图 4-4 所示。

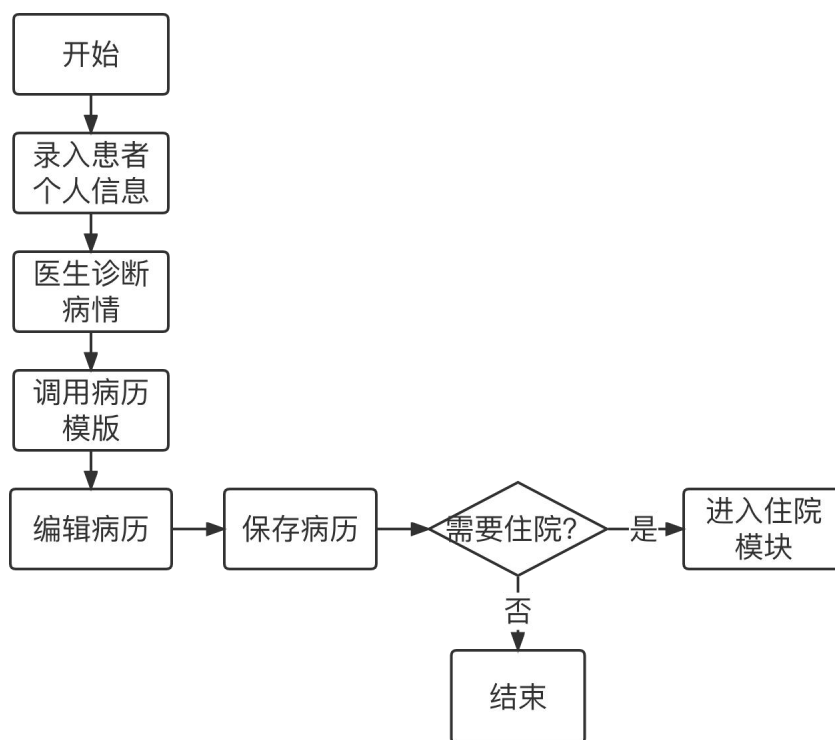


图 4-4 门诊业务流程图

4.6 前端的设计与实现

React 作为目前最流行的前端框架，可以用于快速开发 Web 应用。在电子病历编辑器中，医生的编辑功能是核心，也就是在富文本编辑器上的编辑，需要监听编辑的每一个 DOM 来实现留痕和检查内容的正确性。因此本文引入 React 的组件化技术把各个模块分成一个个组件，把复杂的问题分成一个个小问题来解决，既方便后续的维护和增加功能又方便查错和调试。本文把用户界面拆分为两块，一个是侧边栏组件和病历编辑组件，侧边栏组件又分为患者基本信息组件和病历及模版列表组件，病历编辑组件又分为富文本编辑器组件和留痕组件。技术实现细节如下：

4.6.1 编辑器主页面

1) 患者基本信息组件：本文通过 `componentDidMount` 组件调用后端提供的接口把获得的数据存入组件的 `state` 对象中，然后把数据通过函数传入 HTML 主页面中利用 Ant Design 中的 Descriptions 组件显示。

2) 病历及模版列表组件：本文同样通过 `componentDidMount` 组件调用后端接口获得数据，存入 `state` 对象中，然后通过 Ant Design 的 Menu 组件显示

3) 数据库同时通过 `componentDisMount` 组件传入了 `readOnly` 变量来判断是否可以编辑，如果 `readOnly` 是 1，页面会弹出禁止编辑的提示框。

4) 富文本编辑器组件：使用病历编辑器需要打开一个历史病历或者打开一个病历模版，医生可以点击侧边栏组件来完成这个操作，如果是 `readOnly=1`，那么编辑器会自动设置为只读模式，否则设置为编辑模式。本文使用 UEditor 作为源富文本编辑器。

5) 留痕组件和正确性检测：需要保留的痕迹包括：编辑医生、时间、控件名称、编辑后的内容。不仅如此本文还实现了实时显示历史和本次编辑的区别，因此需要实时监听每一个控件的变化。在富文本编辑器组件中的 `state` 对象中使用数组记录所有控件的编辑信息。当病历初始化时遍历所有控件并记录这些控件的值，由于每个控件填写的内容保存在 `className` 为 `oak-field-value` 的 `span` 中，因此本文在这个 `span` 中增加一个 `EventListener`，用于监听键盘输入。每次发生键盘输入后，获取控件内容和 `state` 原来的内容比较，如果有变化就把原始内容放入历史记录中，把数据传递给页面，由于 React 框架每次数据更新都会刷新一遍渲染，因此可以实时显示更新的过程。同时，由于每个控件都自带内容检测函数，在保存时会遍历控件并调用内容检测函数，如果有错误就会把错误信息传递给页面并显示，同时保存会被取消。

4.6.2 控件

控件的 `css` 配置统一放到 `oak-form.css` 文件中，每个控件都是主要以表格的形式呈现，因此 `html` 表头单元格元素 `th` 和标准单元格元素 `td` 使用较多，控件名称、控件 ID、排列方式都被统一放在 `tr`、`td` 中。`checkbox` 是控件中很重要的一部分，它负责设置本内容是否必须填写。

下面重点介绍复选框、文本输入框、日期输入框控件的实现原理。

1) 复选框。复选框可以给医生提供若干个选项来通过鼠标左键单击的方式填写病历的某项内容，从而控制填写的内容正确且准确。在设计病历模板的时候，设计人员通过点击添加选项来增加一个选项，通过 `onclick="fnAdd();"` ，点击时调用 `fnAdd()` 函数来添加选项，`fnAdd()` 函数传入一个特定的 `id` 调用 `fnAddComboTr()` 函数来生成一列表格，表格有三个需要填写的内容，分别是 `checkbox`: 是否必填，`text`: 选项的内容，最后一个给设计人员下移、上移和删除这列表格的按钮。

2) 文本输入框。文本输入框可以给医生提供一块写内容的区域，一般用作写患者姓名、性别、年龄、门诊挂号序号、身份证号等信息。单行文本可以被设计为填写三种数据类型，分别是普通文本、数字和身份证号，其中身份证号被严格限制为 18 位，如果不等于 18 就会拒绝医生的提交病历请求并报错。实现原理是通过 `html` 的 `input` 元素实现输入框，通过 `checkbox` 实现是否必填的选项。

3) 日期输入框。日期输入框给医生提供了输入患者入院或出院日期的工具。提供几种日期格式的输入，包括 `yyyy-MM-DD`，`yyyy 年 MM 月 dd 日`，`yyyy-MM-dd HH:mm:ss` 等日期格式，点击这个控件会弹出一个小型的日历，医生可以点击日期来完成日期的输入，避免了输入错误的问题。实现原理是通过 `js` 库的迷你日历作为给医生点击的日历模板，时间从 `1900-01-01` 到 `2099-12-31`，医生点击控件是触发函数打开日历，选择就诊日期后，把年月日时间传入把日期转为正确形式的函数，返回正确的日期及格式。

4.7 本章小结

经过了上面三个小结的讨论，本文完成了整体架构的设计，功能模块的设计和数据库的设计，以及前后端具体的实现过程，总结如下：

1) 总体架构的设计是业务分为两块，设计人员负责设计病历模板、医生调用病历模板填写病历后提交把数据存储到数据库中。

2) 功能模块的设计是分为三个部分。信息展示通过传入门诊挂号序号向后端请求患者的基本信息并在前端展示。编辑系统主要负责模版的调用，病历的显示和编辑，病历的存储，打印病历，其中界面的设计思想是，左侧上面根据病历显示患者的基本信息，左侧下面做一个树状的下拉框，可以在这里选择患者的病历和科室的病历模版，打开病历或者模版后，在右侧显示病历，开始编辑，编辑结束后，点击保存按钮后，病历自动保存并提交到系统内部，点击打印按钮就可以打病历。

3) 数据库的设计是先设计一个整体的概念设计，也就是把现实中的概念转化为抽象的实体，接下来设计了四张表格，分别是门诊挂号记录表、历史病历记录表、病历模板记录表和分布式锁记录表。

4) 后端主要使用的技术是 `Gin` 框架，给前端提供了六个接口来处理登录并获取门诊信息，登出并停止门诊，增加病历模板，获取病历模板，增加门诊病历，获取门诊病历，在数据库中增加了一张记录门诊挂号序号、医生、加锁时间的表实现了分布式

锁，保证了同一时间同一张病历只有一位医生在编辑。

5) 前端主要使用的技术是 **React** 框架，通过整合富文本编辑器实现了病历编辑器的主要编辑模块和选择，操作病历或病历模板的菜单栏模块，并且实现了记录最后一次编辑的改动的留痕模块以及通过控件实现了检查正确性的功能。每个控件都是一个 **html** 文件，实现和调试完成后被主页面调用。

5 系统测试

5.1 电子病历主页面

医生使用的电子病历编辑器系统的主页面如图 5-1 所示。

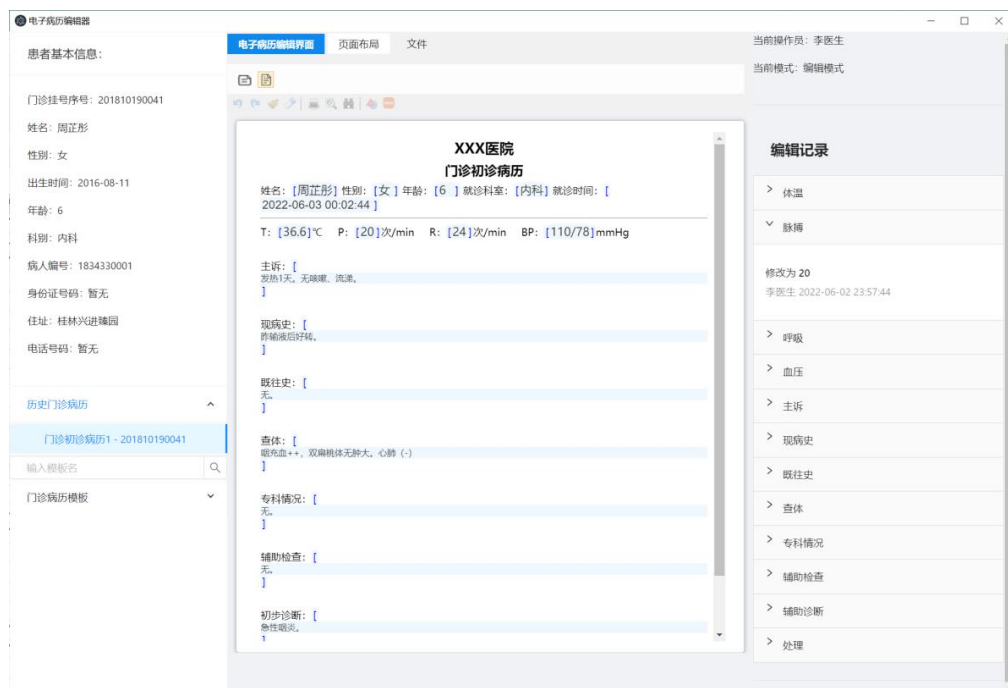


图 5-1 电子病历主界面

可以看出，主页面的左侧是显示了患者的基本信息和两个下拉框，分别表示患者的历史病历和本科室的病历模版，实现的原理是前端使用 React 的 TreeData 插件制作的下拉框，前端通过给后端发送请求，后端给前端返回一个 json 文件，前端从 json 中解析出数据，显示出来。主页面的右侧显示了电子病历编辑器的功能框和病历，有一个类似 Word 的竖直的忽隐忽现的位置就是编辑的位置，病历的显示原理和上面一样，都是前端给后端发送请求，得到数据并显示。

5.2 电子病历编辑器功能

提供各种文字编辑功能，以及电子病例编辑器的模式切换，填写控件内容等功能。该电子病历编辑器在电子病历模版的制作中表现出色，还可以完成电子病历的查看和书写以及存储等功能。

1) 基本的录入功能。可以满足基本的文字输入需求，快捷键和所有主流编辑器一致。根据具体情况可以设置不同的字体、字号、加粗、倾斜等，支持复制粘贴，撤销剪切等主流功能，如图 5-2 所示。



图 5-2 录入功能

2) 表格和图片的插入。支持插入格式为 jpg 或 png 等主流格式的图片。编辑图片的功能包括：旋转图片、拉大和缩小图片、剪裁图片等，方便分析症状。可以通过建立表格，填入患者的数据来更好地分析患者的病情，如图 5-3 和图 5-4 所示。



图 5-3 插入图片功能



图 5-4 插入表格功能

3) 留痕。支持留痕功能，留痕功能可以录入不同医生对病历的增删改查的记录，目的是帮助不同的医生查看患者在其他医生处的诊断记录，方便做出更简单地做出诊断，提高诊断的准确率，如图 5-5 所示。

编辑记录
> 体温
> 脉搏
修改为 20 李医生 2022-06-02 23:57:44
> 呼吸
> 血压
> 主诉
> 现病史
> 既往史

4) 输入提示与正确性检验。控件功能是本文中电子病历编辑器系统的很重要的一个功能，控件是病历的组成部分，通过封装好的内置函数完成对病历书写内容的自动纠错和输入提示。例如需要输入患者的入院日期，点击控件会打开一个日历，点击对

应日期就可以自动完成日期的输入，如图 5-6 所示。

记录时间: [2017-07-21 08:00:00]

日	一	二	三	四	五	六
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

时间 8 : 0 : 0

清空 今天 确定

图 5-6 日期输入提示

5) 分布式锁。通过数据库增加一张记录门诊挂号序号、医生姓名和加锁时间实现，保证同一个病历在同一时间只有一位医生在编辑，如图 5-7 所示。

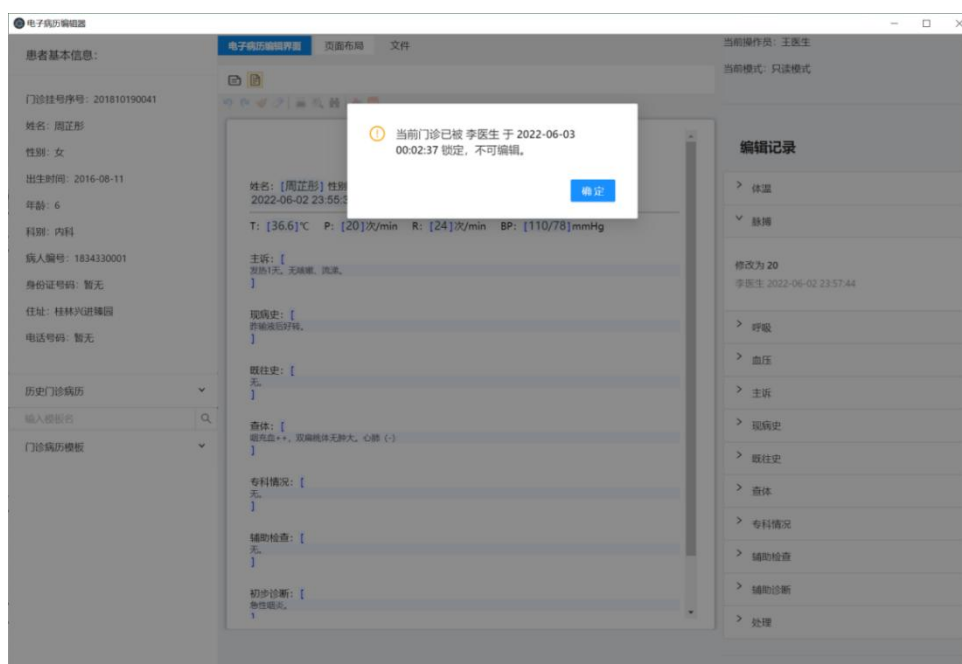


图 5-7 分布式锁

5.3 控件

1) 复选框

模板设计人员需要填写控件名称、控件 ID、排列方式、是否必填以及设计可以选择的内容，如图 5-8 所示。

复选框

控件名称 *

复选名称

控件ID

复选id

排列方式

☒横排 ☐竖排

☐是否必填

固定选项

动态数据

添加选项

选中	选项值	操作
<input type="checkbox"/>	晴天	<div>下移上移删除</div>
<input checked="" type="checkbox"/>	雨天	<div>下移上移删除</div>
<input type="checkbox"/>	多云	<div>下移上移删除</div>

确定

取消

图 5-8 设计复选框

医生点击白色方框选择填写的内容，如图 5-9 所示。

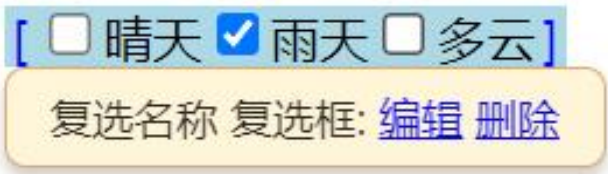


图 5-9 使用复选框

2) 单选框

设计人员负责填写控件名称、控件 ID、排列方式和是否必填，和复选框的区别是可以添加的值只有两个，如图 5-10 所示。

单选框

控件名称 *

单选名称

控件ID

单选id

排列方式

☒横排 ☐竖排

☐是否必填

固定选项

动态数据

添加选项

选中	选项值	操作
<input checked="" type="radio"/>	是	<div>下移上移删除</div>
<input type="radio"/>	否	<div>下移上移删除</div>

确定

取消

图 5-10 设计单选框

医生点击圆心框选择填写内容，如图 5-11 所示。

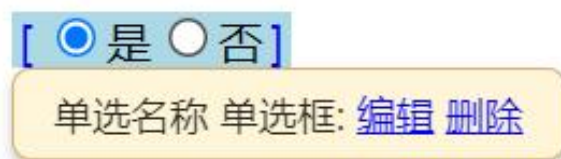


图 5-11 使用单选框

3) 下拉框

设计人员负责填写控件名称、控件 ID、选择字体颜色、是否必填，以及添加数据，如图 5-12 所示。

选中	选项值	操作
<input type="radio"/>	承认有冠心病	下移 上移 删除
<input type="radio"/>	否认有冠心病	下移 上移 删除

图 5-12 设计下拉框

医生点击下拉框，选择一个选项作为填写内容，如图 5-13 所示。

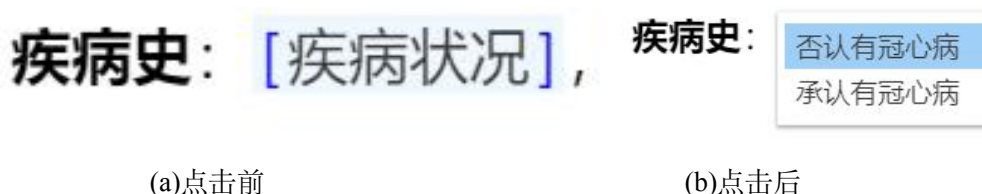


图 5-13 使用下拉框

4) 文本输入框

设计人员负责填写控件名称、控件 ID、默认值、允许输入的最小值和最大值、字体大小和是否必填，选择数据类型、字体颜色，其中数据类型有普通文本、数字和身份证号，如图 5-14 所示。

插入文本域输入

控件名称 *	控件ID	默认值
<input type="text" value="现病史"/>	<input type="text" value="xbs"/>	<input type="text" value="现病史"/>
数据类型	最小值	最大值
<div>普通文本</div>	<input type="text" value="请输入最小值"/>	<input type="text" value="请输入最大值"/>
字体颜色	最小宽度 & 字体大小	
<div></div>	<div><div>20</div>&<div>16</div>px</div>	<div><input type="checkbox"/> 是否必填</div>

确定

取消

图 5-14 设计文本输入框

医生点击改控件鼠标光标会出现在编辑位置，可以删除文字和添加文字，如图 5-15 所示。

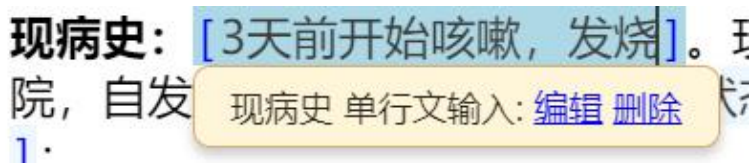


图 5-15 使用文本输入框

5) 日期输入框

设计人员负责填写控件名称、控件 ID、字体大小和是否必填，以及选择格式、默认时间、最小时间、最大时间和颜色，如图 5-16 所示。

插入日期输入框

控件名称 *	控件ID
<input type="text" value="记录时间"/>	<input type="text" value="控件ID"/>
格式	默认时间
<div>yyyy-MM-dd HH:mm:ss</div>	<div>2017-07-21 08:00:00</div>
最小时间	最大时间
<div>最小时间</div>	<div>最大时间</div>
字体颜色	最小宽度 & 字体大小 & 是否必填
<div></div>	<div><div>150</div>&<div>16</div>px <div><input type="checkbox"/> 是否必填</div></div>

确定

取消

图 5-16 设计日期输入框

医生点击日期控件在日历中选择日期和时间，如图 5-17 所示。



图 5-17 使用日期输入框

5.4 本章小结

经过上面三个小结的讨论，测试了电子病历编辑器的主页面布局，编辑器功能和控件的设计和使用，测试全部通过，电子病历编辑器基本实现成功。

6 结论与展望

6.1 本文主要工作总结

目前, 我国的电子病历编辑器系统和国外有着较大的差距, 结构化病历存储和智能纠错和提示的病历输入都存在一定的缺口, 因此制作一个国有的电子病历编辑器系统是非常重要和有意义的。在张亚明老师的指导下, 和银海医院的李医生的讨论下本人对电子病历编辑器系统的功能需求有了一定的了解, 成功制作出了一个新型电子病历编辑器系统, 本人的主要工作有以下四点:

1) 实现了结构化的病历存储。本文中的电子病历编辑器系统使用了结构化的病历存储, 原理是使用 XML 格式文件为中介, 保存时数据使用 XML 导出, 后端解析出 XML 的文件数据, 保存在数据库中, 实现结构化存储, 比以往的病历编辑器的存储的效率更高而且节约了很大的物理内存空间, 并且由于使用的数据库 Microsoft SQL Server 具有故障恢复的功能, 因此发生了重大事故也可以把病历的数据进行恢复, 使得保存病历的安全性得到了提升。

2) 实现了智能纠错和输入提示。智能纠错和输入提示主要是利用了控件。控件的实现是电子病历编辑器的主要的进步技术, 本文中电子病历编辑器封装好可以使用的控件库中有单行文本控件、多行文本控件、日期控件、单选框控件、复选框控件、下拉按钮控件等实用的控件。这些控件保证了设计人员设计病历时不需要写代码和改代码, 只需要拉动本文给出的控件库中的控件就可以使用控件来设计病历, 医生使用只需要点击控件并根据提示选择或输入文字就可以完成编辑, 使用起来很便捷。

3) 实现了前后端分离的系统。本文中电子病历编辑器系统是使用前后端分离的设计理念来完成的, 这样设计的好处是符合工业界对大部分应用的主流设计方式, 日常维护的技术很多, 运维也很方便。因为耦合性不强, 需要增加需求和修改需求时, 只需要把对应的功能改一下就可以, 不需要修改一个接口还要牵扯到其他接口的修改, 开发起来很快, 符合国际主流敏捷开发的要求。综上, 本文的电子病历编辑器系统的设计、投入使用和运维都是符合工业届需求的, 可以很好地投入实际使用中。

4) 实现了留痕等编辑器高级功能。电子病历编辑器在编辑功能上和其他普通的编辑器相比, 增加了留痕这个比较高级的功能。留痕是指类似于 IDE 中的工作流中显示改了哪些代码的功能, 显示医生在本次修改中, 对病历做了哪些增删改查, 好处是帮助医生一目了然自己的诊断和医疗方案的制定过。

6.2 未来工作展望

本文制作的电子病历编辑器系统为其他想制作功能更强大的项目提供了一些技术支持和代码实现, 虽然这些技术以及实现确实能解决当前病历系统基本和普遍的一些

问题，但是远远还没达到能实际应用到的顶级医院对于不同种类的疾病的病历输入的要求。由于本人的只是本科生学识有限，针对本文中未完成的工作和不足之处提出以下几点展望：

1) 根据文献调研，如果能够在各大医院把各个疾病的数据收集起来，制作成一个知识库，利用知识库和 NLP（自然语言处理）的技术，给不同领域的医生提供非常个性化的编辑器提示和纠错服务。如果可行，这样就能大大减轻医生的负担。因此制作出知识库有待进一步政策的发布和实践。

2) 可以把 API 部署到云服务器中。秉持着开源精神，应该把动态展示患者生理数据的接口提供给各大病历编辑器的制作人，能够帮助他们展示患者生理数据的变化状态，有助于医生判断患者的病情走势，做出最佳的医疗方案。

3) 更先进的技术使用还不够完善，如果有时间，可以使用最先进的 MongoDB 数据库，Redis 缓存，消息队列等工业界广泛使用的先进技术，完善后端系统。不仅如此，本文在设计控件种类还不多，应该设计更多的控件来更方便地给医生使用，而且应该把设计控件的 API 提供给其他技术人员，这个控件库应该开源给所有人，让大家一起给这个开源的控件库贡献自己的控件代码，然后一起使用这套控件库，做到事半功倍。

参考文献

- [1] Jun Liang, Ying Li, Zhongan Zhang et al. Adoption of Electronic Health Records (EHRs) in China During the Past 10 Years: Consecutive Survey Data Analysis and Comparison of Sino-American Challenges and Experiences[J]. J Med Internet Res 2021;23(2):e24813
- [2] Rachel R, Rupali G, Roberto AR. A Clinical Rule Editor in an Electronic Medical Record setting: Development, Design, and Implementation[J]. AMIA Annu Symp Proc. 2019;2019:537-541.
- [3] 舒婷, 赵犇, 刘海一. 2020年本文国医院电子病历系统应用水平分析[J]. 中国卫生质量管理, 2022,29(01).
- [4] 张洪伟, 梅传发. Vim 编辑器自动补全功能设置与使用[M]. 北京: 实验室科学, 2019, 22(04).
- [5] 秦丽丽. 基于 HIS 的结构化护理不良事件管理系统的研发及应用[G]. 第四届上海国际护理大会论文汇编. 上海: 上海护理出版社, 2019.
- [6] Edward P. Ambinder. Electronic Health Records[J]. J Oncol Pract.2020;1(2): 57-63.
- [7] 韩圣亚, 严莉, 刘萌, 徐浩, 朱韵松. 基于 XML 的自动化异构系统数据一致性校验方法[J]. 电子设计工程, 2021,29(13).
- [8] Maria Luisa Ventura 1, Anna Maria Battan, Chiara Zorloni, Laura Abbiati, Marco Colombo, Stefano Farina, Paolo Tagliabue. The electronic medical record: pros and cons[J]. J Matern Fetal Neonatal Med. 2021.
- [9] 李晓薇. React 前端应用技术分析. 网络安全技术与应用[J], 2022,(04).
- [10] 吕景美. 基于 ueditor 富文本编辑器的实验报告在线编辑系统[M]. 北京: 考试周刊, 2013, (04).
- 涂腾飞. Go 语言中的并发问题研究设计[M]. 北京: 北京邮电大学, 2019.