西南科技大学

Southwest University of Science and Technology

本科毕业设计（论文）

题目名称：LinkingDoctor 医疗影像阅片系统前端设计与实现

学生姓名：王 岚

学生学号：20131226

专　　业：计算机科学与技术

指导教师：刘 畅

学院(部)：计算机科学与技术学院

LinkingDoctor 医疗影像阅片系统前端设计与实现

摘要

“互联网+”目前已经成为了社会上一种主流的创作模式。随着“互联网+”越来越深地渗入人们的生活，日常生活、工作、学习变得极为方便，这不仅改善了人们的生活方式，也改变了人们的生活习惯。为打破医学影像长期封闭于医院围墙内这一不利现状，为医生群体提供更多更丰富的肿瘤医疗研究资料，LinkingDoctor 以 Web 应用的形式作为医生以及医疗机构的工作平台给医疗诊断和资料共享提供了条件。LinkingDoctor 采用 webpack 作为模块的加载器和打包工具，使用 AngularJs 和 Bulma 作为项目框架进行 web 前端开发。其中主要的模块使用了 Cornerstone.js 和 Xtk.js 等 js 库文件完成了二维医学影像阅片和三维影像阅片功能。在此 Web 应用中，主要为用户提供了影像查询、报告撰写、二维影像阅片、影像勾靶、三维影像阅片、影像上传和下载等功能，基本满足了医生及机构对影像阅片系统的需求。

关键词：LinkingDoctor，Cornerstone，Xtk.js，影像勾靶

DESIGN AND IMPLEMENTATION OF LINKINGDOCTOR MEDICAL IMAGE CLOUD PLATFORM'S FRONT END DEVELOPMENT

ABSTRACT

Now, the "Internet+" has become a mainstream culture in the social as a kind of creation mode. With the "Internet +" more and more deeply into people's lives, people's daily life, working and learning has become extremely convenient, which not only improved people's Lifestyle, but also changed people's habits. In order to break the adverse situation that medical images has been closed in the hospital fence for long, and to provide more and more abundant tumor medical research material for doctors.LinkingDoctor provides the conditions for medical diagnosis and data sharing in the form of Web applications as a medical platform for doctors and medical institutions.LinkingDoctor uses webpack as a module loader and packaging tool, using AngularJs and Bulma as a project framework for web front-end development.The main modules using the Cornerstone.js and Xtk.js and other js library files to complete the two-dimensional medical imaging and three-dimensional image reading function.. In this Web application, mainly for the user to provide image query, report writing, two-dimensional image reading, image hook target, three-dimensional image reading, image upload and download functions, basically meet the doctors and institutions on the image reading systemThe demand.

Key words: LinkingDoctor, Cornerstone, Xtk.js, Image Target Hooking

目 录

第 1 章 绪论

1.1 课题的目的和意义

随着“互联网+”的快速发展和不断盛行，大数据、云计算等科技不断发展，社会对市场、对用户、对产品、对企业价值链乃至对整个商业生态进行重新审视和思考方式的不断衍生的背景下，医疗行业这一传统而重要的行业也在“互联网+”的发展过程中得以发展。肿瘤治疗作为对医疗影像高依赖的医疗类型，在互联网应用广泛的时代能帮助医生对肿瘤疾病的治疗有更广阔的视野，加以更丰富的学习资料和经验。数以万计的癌症患者饱受肿瘤的给他们带来的生理痛苦和心里伤害。国内外先后出现了多家致力于治疗肿瘤的专症专治医院，但是由于病例仅限于院内流通，使得肿瘤治疗的发展遇到了很多阻碍和瓶颈。

“互联网+医疗”获得社会各界广泛关注，多种肿瘤病种的数据搜集和分析正在进行中，后续将建立大数据平台，用于肿瘤的疾病预防、诊断、治疗，以及药物的研发、保险服务等领域。不仅帮助了更多的人了解自身的身体状况，也帮助了医疗人员开阔视野，互相学习。互联网技术的出现使得医疗行业不论是在技术上还是在形式上都取得了重大的突破和变革。多媒体医学影像系统的应用使医疗卫生事业实现信息化、数字化、计算机化、无纸化、无胶片化管理,对于促进现代医疗的发展有着建设性的作用,也是现代医学发展的必然趋势。[1]

1.2 课题研究现状

首先是，“互联网+肿瘤”的模式缺乏标准化。通过对国内外有关“互联网+肿瘤”的网站、App等软件产品的试用和分析以及搜集分析相关的业内研究报告，可以看出，当前国内外的肿瘤数据平台不论是形式上还是内容上缺乏一个指定的标准，这相比于肿瘤医疗的桌面应用有着尤为明显的差距。很多的网站构建不符合医生的使用习惯，不利于肿瘤数据的分析而一味追求个性，突出产品差异。

其次，业务流程图图例“互联网+肿瘤”离普及医疗互联网的目标还有很长的路要走。虽然现在专注“互联网+肿瘤”的公司越来越多，但使用互联网产品进行肿瘤数据分析的医疗机构却寥寥无几，这一点在国内市场尤为明显。很多省内的大医院都仍旧桎梏于桌面应用，医院数据因而也无法得以全国共享，这对于肿瘤医治水平的提高是个不利条件。一些医生团体意识到“互联网+肿瘤”的重要性，已经开始自主参与使用此类互联网网站，而投资大企也十分看好“互联网+肿瘤”的模式，进行了较大规模的投资工作。

1.3 课题目标

系统针对医生群体或医疗机构，为用户提供了病人的肿瘤影像阅片系统。医生可以随时随地地通过互联网查看权限范围内的病人信息和影像并撰写报告，而此类病人信息不仅仅局限于医生自己的病人，还可以是其他医院的病人，这是LinkingDoctor阅片系统最重要的一点，可以做到医疗信息共享和医疗资源共享。资历老的医生可以通过该平台跨区域地帮助缺乏医

疗经验的医生诊断病人病情，从而达到最优化的治疗，给予病人更权威的治疗方案和手段。

1.4 课题实施方案

项目按照需求调研分，调研结果分析，项目概要设计，项目详细设计，系统开发与实现的顺序进行实施。

（1）通过与医生当面谈话和调查问卷的方法，调研医生在通过桌面应用查看病人的医疗影像时的使用习惯和必用或常用的功能；

（2）分析调研结果，给出项目的大体业务流程和业务逻辑，对项目的可行性和实现难度进行预估，整理出项目实现所需的必要技术支持和需要攻破的难题，并制定实现项目开发的技术方案。

（3）对项目进行概要设计，包括业务逻辑、模块设计、需要实现的功能等；

（4）制定项目的详细设计方案，绘制项目流程图、制定每个模块的功能点；

（5）根据之前的设计方案进行界面设计和代码编写，逐步实现项目各模块的功能；

（6）在实现项目功能的基础上，优化用户体验，保证网页性能，如响应请求的速度、文件下载的速度优化，并改善用户的使用舒适度等各类问题。

1.5 本章小结

本章主要介绍了当前课题的研究范围和国内外现状以及课题实现的目的意义和实现方案，随项目的研究反向进行了较为总体的概括。

第 2 章 医疗影像阅片系统前端开发关键技术介绍

为了在短期内搭建出结构清晰、可复用性好，且易于维护的 Web 应用程序，该系统使用了基于 MVC 的 AngularJs 框架进行项目技术，使用 Cornerstone 和 Xtk 技术进行阅片页面的开发,使用了 webpack 技术进行项目的构建和打包工作。该系统选择 Webstorm 作为开发工具,运用 git 技术使得项目开发便捷化。本章中将对 LinkingDoctor 医疗影像阅片系统前端开发实现所使用的关键技术进行简单介绍。

2.1 MVC 框架

MVC 程序结构是在 20 世纪 70 年代作为 Smalltalk 语言的一部分被引入的[2]，从那以后 MVC 一直在桌面开发环境中非常流行，但是却长期与 web 开发毫无联系。MVC 所奉行的将管理数据的代码（model）及应用逻辑代码（controller）和向用户展示数据的代码（view）清晰地分开，使得复杂的代码编辑工作变得有条理而分工明确，这不仅规范了代码的编写，还给开发人员带来了极大的方便。

这里需要区分一下 MVC 框架与 MVC 设计模式。框架，是在多个项目软件开发的过程中，根据开发者的经验积累，提取出特定领域的软件的共有且有效的部分，从而形成的体系结构。所以在不同的领域中，软件项目也有着不同的框架类型。要清楚，软件的框架不是一个现成可用的应用系统，而是说，框架相当于一个项目的半成品，它能提供了许多开发者需要的服务，供开发人员进行二次开发，以此来高效且规范地实现具体功能的软件应用系统。

2.2 AngularJs

AngularJS 主要用于构建单页面 Web 应用[3]，它是一个在 Web 前端开发工作中应用极为广泛的前端 MVC 框架，目前它也是网上使用量最大的框架（根据项目数量）。它于 2009 年由 Misko Hevery 和 Adam Abrons 等人创建，后来其公司被 Google 公司并购，AngularJs 得以继续发展，逐渐成为一款成熟而优秀的前端 MVC 框架。

AngularJs 扩展了 Web 应用中的 HTML 词汇，如 ng-app、ng-controller、ng-check 等，从而在应用程序中使用 HTML 语言进行动态内容申明，极大地方便了 Web 应用的开发。AngularJS 有着许多有趣而高效的特性，其中最为核心的就是运用了 MVC 设计模式，其具有模块化、还有模板和数据的双向数据绑定、依赖注入、语义化标签等等特性。其中最令人称道的是 AngularJs 的双向数据绑定，通过双向数据绑定，我们节省了从前运用 jquery 进行页面刷新的一大段代码。除此之外，AngularJs 还支持快速测试、ＵＲＬ 管理等，并且随着此框架的快速发展，github 上越来越多的框架相关的扩展功能，Web 前端开发变得工作越来越简洁，有条理，便于阅读。AngularJs 使用简单而清晰地声明式模板实现各种特性，并且使用大量的现有组件或自定义组件，扩展模板语言还提高了网页渲染的速度，帮助开发人员方便快捷地写出各种高性能的优美应用。尽管 AngularJs 有着强大的功能和特性，为大部分开发人员提供了一个更高层次的抽象来使得应用程序的开发变得简洁，但它同时也损失了一些灵活性，所以它也不是一个万能的框架，尤其是在 DOM 操作极为复杂的应用中，比如网页游戏、图形界面编辑器，AngularJs 就失去了用武之地。

本项目中，尤其是在影像查询模块和报告撰写模块中存在很多的异步加载操作，AngularJs 能够很大程度地提高网页渲染速度，极大地简便了代码的编写，再加上其在各大互联网门户中的广泛应用，可见其发展的成熟度较高，学习资料丰富，使用起来更加便捷有效。

2.3 Dicom

DICOM 标准是 Digital Imaging andCommunications in Medicine 的缩写,是用于临床使用和研究的图像交换的普遍标准[4]，其目的是支持从 CT，MRI 和其他医疗模式分发和观看医学图像[5]。Dicom 标准包括了医学的数字成像和通讯两个方面[6]，它定义了质量能够满足临床需求的可以用于数学化医学影像传送、显示和存储的医学图像格式，是 ACR（American College of Radiology）和 NEMA（National Electrical Manufactorers Association）为主制定的。1985 年，第一版 Dicom 协议发布，给放射学实践带来了巨大的改变，X 光胶片全部被数字化，医学影像的工作流程变得更加方便而易于存储和管理。Dicom 标准不仅仅支持医学放射图像,它是可扩展的,既支持放射学图像,也支持非放射学图像[7]。

在 Dicom 协议中，定义了关于医学影像以及其相关信息的格式和交换方法，利用这个协议，我们可以在各类影像设备上进行影像数据的传输工作，有效帮助医学影像设备之间的数字影像传输工作和医疗设备的数据获取、存储、显示、转换工作，其中包括的医学影像类型有 MR、CT、CR 和超声图等。

DICOM 的普及不仅提供了高效率的图像存取和使用服务,也为医院中包括图像信息在内的综合医学信息管理奠定了基础[8]。其中 DICOM 标准文件格式如图 2-1，DICOM 文件十六进制格式如图 2-2：

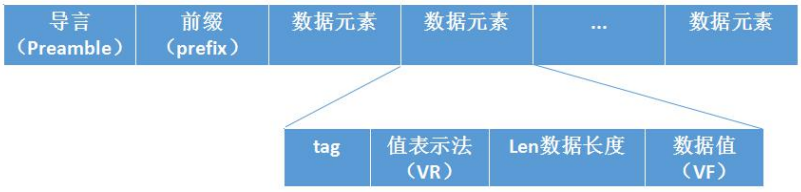


图 2-1 DICOM 文件格式

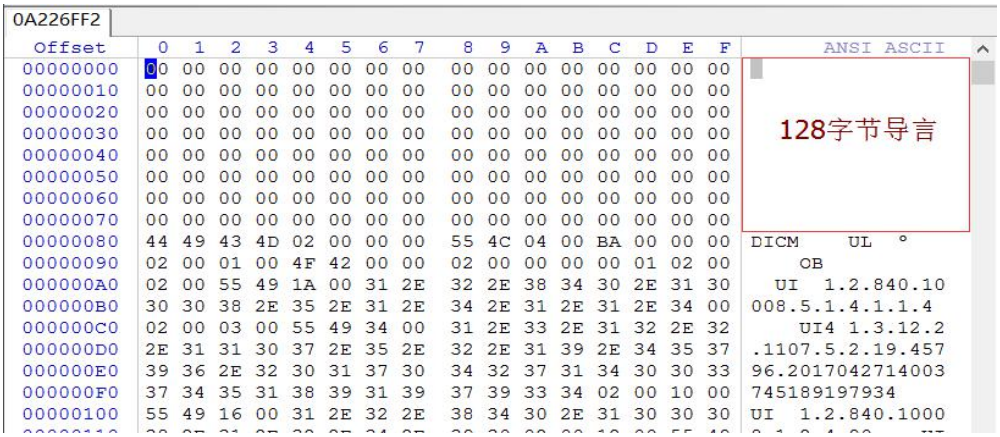


图 2-2 十六进制的 DICOM 文件

DICOM 文件中的第一部分是引言，可以不写，基本上所有的 DICOM 文件的前 128 字节均为 0。

第二部分是跟在引言后的 4 字节文件标识，图 2-2 的右侧有一个 DICM，它是 DICOM 文件标识，如果不是 DICM 则表示不是 DICOM 文件。

第三部分为数据集，一个数据集描述了现实世界信息对象的一个实例。数据集由数据元素 (Data Elements) 构成，数据元素是对对象属性值的编码[9]。

数据集中包含了文件的元数据、普通数据元素、像素数据数据元素、患者及检查的数据、私有数据以及图像、覆盖层等数据。数据元素中的 tag 就是 dicom 里定义的字典，所有数据

元素从前到后按 tag 又可简单分段。文件数据元素不受传输语法影响，总是以显示 VR 方式表示，因为它里面就定义了传输语法；普通数据元素受传输语法影响，显示 VR 表示方式还是隐式 VR 表示方式；像素数据数据元素是最重要也是最大的一个数据项，其实存储的就是图像数据。掌握以上内容才可以进行 DICOM 文件的解析。

2.4 Cornerstone

Cornerstone 是专门用于操作符合 Dicom 标准的医学影像图片的 js 包，他可以用于医学图像在网页端的获取、显示和各类医学图像操作，包括调整窗宽窗位、改变影像大小、移动影像位置、影像旋转、影像局部调整等。其影像的显示基于 HTML5 的 canvas，在网页上进行点的绘制，而影像的操作也是点的操作。通过 Cornerstone 绘制出的图像位置又不同于普通的 canvas 标签绘制的图像，其相对位置是基于所附属的 Div 内又库文件再动态加载的 Div 标签。Cornerstone 的获取方式也分为本地获取和远程获取，需要分别运用不同的方法去取得所需图像。

2.5 Xtk

Xtk（X ToolKit）是 github 上的开源库，它是主要用于三维医学影像显示和操作的 js 包，它能支持 OBJ、NII、VTK、STL 等格式的三维医学影像的大型文件，同时也能支持 PNG、JPG/JPEG 等二维图像小文件。Xtk.js 是基于 WebGL 的轻量且高效的 js 库。XTK.js 内置的三维影像操作功能十分强大，包括移动、旋转、放大缩小、透视等等，充分满足了医生在查看医学影像时的各种需求。Xtk 的使用简单而方便，能够快速地在浏览器端渲染出三维模型，与桌面应用相比毫不逊色。

2.6 webpack 打包

webPack 是一款好用的模块加载器，它也是近年来很火的一款打包工具，他能够把各种互为依赖的资源，包括 js、jade、png/jpg、less/sass、coffee 等都模块化，然后进行使用和处理。Webpack 根据 Commonjs 规范进行脚本的编写，使用 webpack 作为模块加载器的项目在开发工作中进行加载工作时非常便捷，它当中有很多健全的模块加载器（loader）用于模块的加载和编译，我们只需要以 require(XXX)的形式来引入各种模块，就可以方便地使用这些模块了。Webpack 的扩展性非常强大，它有着完善的插件机制，同时也支持 AMD/CMD 模式。Webpack 是模块化代码中不可或缺的。它比其他工具，例如 Bower 和 Browserify，更容易使用，并且不需要其他工具（如 gulp 或 grunt）来管理构建过程。使用 webpack 不仅能将客户端代码模块化并构建成一个捆绑包来传递给浏览器，还可以编译一些代码[10]。

2.7 本章小结

本章介绍了在项目开发过程中需要用到的关键技术，包括 AngularJs 框架作为项目开发的基本框架，然后使用 webpack 作为模块加载器和打包工具。重点使用了用于医疗影像的 dicom 影像协议以及能够操作 dicom 图片的 Cornerstone 库函数和操作三维影像的 xtk 库文件。

第 3 章 医疗影像阅片系统前端需求分析

3.1 可行性分析

通过对系统进行可行性分析，可以有效地降低对系统的投资失误率，是保证系统开发成功的基本前提。本课题对实现 LinkingDoctor 医疗影像阅片系统的前端开发进行了以下几方面的可行性分析：

（1）经济可行性：由于本系统的开发人员只有学生，而项目开发的指导教师对开发者进行义务指导，并且项目的开发环境都使用的是免费的开源软件，经过预算，该项目的开发无需任何经费，开发的经济成本为零。而本项目的实现可帮助肿瘤医疗事业获得革命性的改进，使医学影像在医生和机构之间得以形成资源共享、经验共享，并藉此提升肿瘤医疗服务水平，对于推进我国肿瘤医疗行业的改进有着重要的意义，因而本项目的实现在经济上是可行的。

（2）技术可行性：技术可行性，是通过分析项目开发的技术条件是否能满足开发的需求。本课题将实现基于 MVC 模式的医疗影像阅片系统，系统采用的开发技术包括 HTML/HTML5、LCSS、Javascript、Jquery、AngularJs、Bulma、Webpack 等技术。目前来看，这些技术都是现在比较流行的互联网 Web 前端开发技术，可以在网络、书籍中查询到大量的技术学习资料，因此本项目的实现在技术上是具备可行性的。

（3）操作可行性：本项目旨于服务肿瘤医疗行业中的医生群体和医疗机构，帮助其在工作生活中便捷地随时随地地查阅病人影像信息，提高医生的工作效率，提升肿瘤疾病治疗的治愈率。本项目的实现是基于医生群体对影像处理系统的使用习惯上进行设计的，在用户体验上，具备合理的使用条件，用户对于其操作和流程是比较熟悉的。用户只需要在有网络的情况下就可以在任何时间、地点进行影像的阅片工作，极大效率地利用了用户的碎片时间。且本项目使用 AngularJs 和 Bulma 实现了美观且友好的 Web 交互页面，据此，本项目的实现具有操作可行性。

（4）社会可行性：由于本项目的实现为各个机构的医学影像资源提供了一个便捷的共享平台，帮助医生和医疗机构获取更多的学习资源并提升了医疗工作的效率，打破了之前的医学影像资源困于医院围墙之内的问题，推动了医学影像行业的共享化发展，降低了工作成本，有效避免了传统影像传输中容易发生的错误，因此，本项目具有社会可行性。

结合以上四方面的分析，可以得知本项目的开发是完全可行的。

3.2 非功能性需求分析

非功能性需求是指软件在满足用户的业务需求以外的其他需求。非功能性需求是系统开发和后期维护中很重要的一个部分，它所涉及的问题包括系统的性能，可靠性，可维护性，健壮性等。

本课题所研发的医疗影像阅片系统的前端的非功能性需求定义如下：

（1）实用性：系统需要医生在影像阅片流程中的业务要求。

（2）易用性：网页的界面美观，并且页面的交互设计符合用户的操作习惯，用户经过简单的操作培训就能完全掌握平台的使用方法。

（3）可靠性：平台需要稳定地运行，尽量少得发生错误，避免用户在使用过程中出现网页崩溃或影响加载失败等现象。

（4）性能需求：网页加载的图像数据量大，普通情况下满足在 20 秒以内的时间中对 100 个指标进行加载。

3.3 功能需求分析

在对现有的肿瘤医疗影像阅片系统进行了充分的调研和分析后，提取了医生在进行医学影像阅片时的工作流程，并对本课题所研发的医学影像阅片系统进行了功能需求分析。

对医疗影像阅片工作流程描述如下：首先，是二维医疗影像通过医疗设备直接传到后台系统中，此时机构端和医生端都会出现这条患者的影像信息，机构端的此条信息状态为“已上传”，然后机构端用户对最新上传的影像进行确认和病人基本信息的反馈，这时，机构端的状态会变成“已确认”，而医生端的状态则会变成“处理完成”，与此同时医生端和机构端就都可以进行二维影像的阅片工作了。医生端和机构端都可以下载二维的病人影像集的 zip 文件。医生端可以下载二维影像进行三维重建后上传自己的三维影像文件和 pdf 影像报告文件到对应的患者信息中，同时对三维影像的重建进行反馈。此时机构端的状态会从“已确认”变成“处理完成”，医生端和机构端都可以进行三维影像阅片和二维影像阅片工作。

在用户登录时分机构端用户和医生端用户，输入用户名、密码即可登录。登录后可以退出登录状态。

在患者影像查询时，可以通过患者姓名、年龄、性别、影像上传日期、影像类型等进行查询。

在进行二维影像阅片时，医生（包括医生端和机构端的医生）可以进行的操作包括：对一个序列的影像的滚动查看，影像的自定义布局，影像窗宽窗位的调整，影像局部对比度调整，影像的放大缩小，影像的旋转，影像的上下左右翻转，在影像中绘制点、线、测量角、椭圆，影像的平移、影像序列的自动播放，影像反色，影像恢复原始状态，影像中角标的隐藏与显示，获取影像的 DCM/JPEG 格式，影像之间的定位线绘制，影像的勾靶操作和去除勾靶线条等功能。

在二维影像阅片界面中进行报告撰写时，医生（包括医生端和机构端的医生）可以进行的操作包括：与报告相关影像的选择，影像表现、诊断意见和备注的编写、报告模板的选择、新建、编辑，报告的保存和清空等功能。

在进行三维影像阅片时，医生（包括医生端和机构端的医生）可以进行的操作包括：影像旋转、影像的放大缩小、影像的平移、影像的颜色改变、影像的透视和 pdf 报告的查看和下载、打印功能。

3.4 用例分析

用例技术是通过用例、参与者与用例以及用例之间的关系来描绘系统外在需求的一种方法。作为 UML(统一建模语言)的一种重要表示法,用例分析方法在软件开发过程中占据着重要的地位。正确使用用例分析方法有助于项目的需求分析、体系结构设计、进度安排、测试和验证。文中简要介绍了增量/迭代式软件过程,通过实例探讨了软件开发中如何使用用例分析技术,包括项目风险分析、确定系统边界、细化事件流、图形化用例以及用例归档技术,从而为获取用例模型提供了有效的方法和途径[11]。要想从用例中捕获用户需求就需要使用合理的结构,正确的需求分析方法,把用户的行业业务整理成一个清晰的表述方式[12]。

系统用例分析是系统需求与在项目设计的中介，分析得出的结果是建立项目逻辑模型的根据，这一逻辑模型需要提供出满足客户要求的业务逻辑和功能。通过需求分析得出，该系统需要设置两类用户：机构用户、医生用户。

3.4.1 机构用户模块

机构用户是医疗机构端或医生群体中掌握患者影像上传信息的用户，只有机构用户能够

反馈上传用户的信息。机构用户模块主要包括用户登录与登出，影像上传确认及病人信息反馈，患者影像查询，二维影像阅片，三维影像阅片，影像下载等几个模块。模块功能简介：二维医疗影像通过医疗设备直接传到后台系统中，机构用户登录到平台，此时，机构端的影像信息状态为“已上传”，然后机构端用户对最新上传的影像进行确认并对患者的基本信息进行反馈，这样，机构端的状态会变成“已确认”，机构端用户可以进行二维影像的阅片和二维的病人影像集的 zip 文件下载。等到医生端下载二维影像进行三维重建并上传三维影像文件和 pdf 影像报告文件到对应的患者信息后，机构用户中的该条影像信息的状态会从“已确认”变成“处理完成”，机构端可以进行三维影像阅片。用例图如图 3-1 所示：

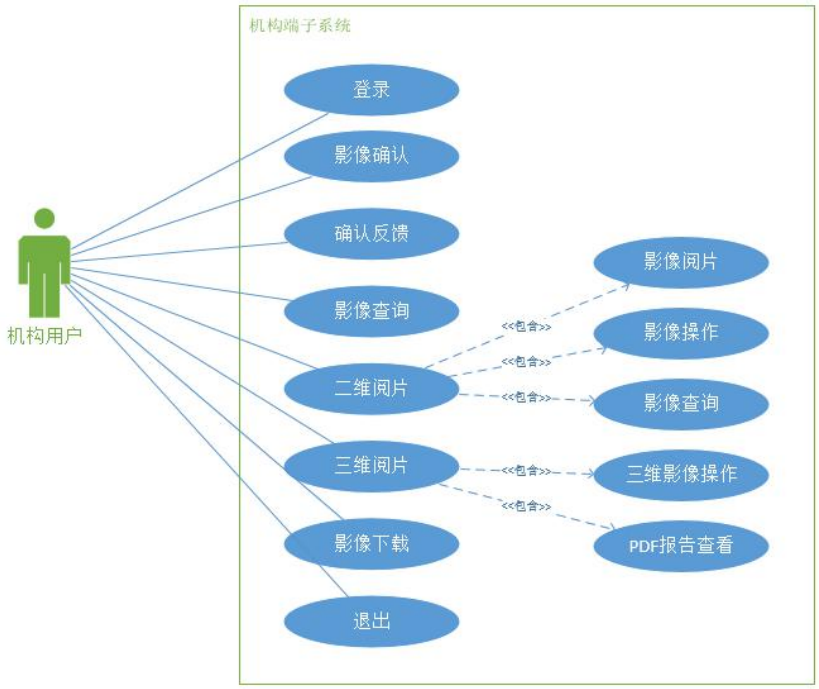


图 3-1 机构用户用例图

其中二维影像上传后的影像确认反馈的用例描述表如表 3-1 所示：

表3-1 影像确认反馈用例描述表

用例：影像确认反馈	
简单描述	用户在本平台已上传二维影像的情况下进行患者信息的影像确认和信息反馈
主参与者	机构用户
副参与者	数据库
前置条件	本网站被打开，二维影像已上传成功，且用户需要进行患者影像信息确认和反馈操作
主流	（1）用户打开本网站 （2）用户登录进入查询页面 （3）在查询页查看已上传的患者影像 （4）用户点击对应影像进行确认，弹出影像信息反馈框 （5）用户确认或修改病人信息并编辑反馈内容 （6）用户点击保存
后置条件	该患者影像状态变为已确认

3.4.2 医生用户模块

医生用户只能进行患者影像下载和三维影像重构，并进行影像阅片，不能够上传影像并反馈影像信息。医生用户模块主要包括用户登录与登出，患者影像查询，二维影像阅片，二维影像下载，三维影像和 PDF 文件上传，三维影像上传信息反馈，三维影像阅片，报告撰写等几个模块。模块功能简介：医生用户登入平台，可以进行已存患者影像查询工作和二维影像阅片工作。医生用户可以下载二维影像的 zip 文件，进行三维重建后上传三维影像 obj 文件、存储影像颜色和纹理的 mtl 文件和 PDF 报告文件，医生用户上传完成后进行三维影像的信息反馈。此时，医生用户可以进行三维阅片、二维阅片、登出等操作。用例图如图 3-2 所示：

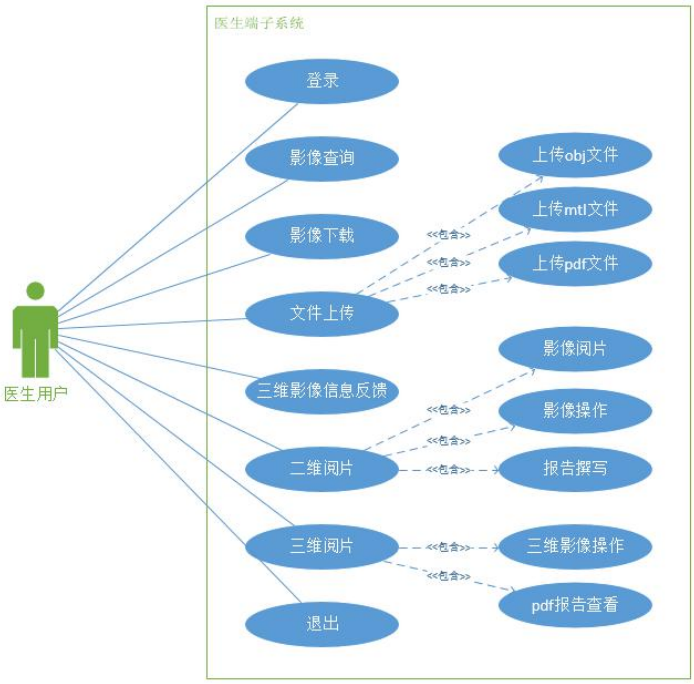


图 3-2 医生用户用例图

其中影像查询的用例描述表如表 3-2 所示：

表3-2 影像查询用例描述表

用例：患者影像查询	
简单描述	用户在本平台查询页上进行患者影像的多条件查询
主参与者	医生用户、机构用户
副参与者	数据库
前置条件	本网站被打开，且用户有影像查询需求
主流	（1）用户打开本网站 （2）用户登录进入查询页面 （3）在查询页填写或选择查询条件 （4）用户点击查询，完成查询操作
后置条件	用户获得相应查询结果
附加流	用户查询条件得到查询结果为 0 条时，页面显示“没查询到相应数据”

用户在进行二维影像阅片时的用例描述表如表 3-3 所示：

表3-3 二维影像阅片用例描述表

用例：二维影像阅片	
简单描述	用户在本平台查询页上进行患者影像的多条件查询
主参与者	医生用户、机构用户
副参与者	数据库
前置条件	本网站被打开，且用户有二维影像阅片需求
主流	（1）用户打开本网站 （2）用户登录网站进入查询页面 （3）用户点击“断层”或“断层图像”按钮进入二维影像阅片页面 （4）用户进行二维影像阅片操作
后置条件	用户按需操作二维影像

三维影像阅片的用例描述表如表 3-4 所示：

表3-4 三维影像阅片用例描述表

用例：二维影像阅片	
简单描述	用户在本平台已上传二维影像的情况下进行患者信息的影像确认和信息反馈
主参与者	机构用户
副参与者	数据库
前置条件	本网站被打开，且用户有三维影像阅片需求
主流	（1）用户打开本网站 （2）用户登录网站进入查询页面 （3）用户点击“3D”或“3D 图像”按钮进入三维影像阅片页面 （4）用户进行三维影像阅片操作
后置条件	用户按需操作三维影像
附加流	三维影像未上传时页面显示“没有 3D 图像数据，请上传。”

文件上传的用例描述表如表 3-5 所示：

表3-5 文件上传用例描述表

用例：文件上传	
简单描述	用户在本平台已上传二维影像的情况下进行影像的三维重建并上传其 obj 文件、mtl 文件和 pdf 报告文件
主参与者	医生用户
副参与者	数据库
前置条件	本网站被打开，且用户有三维影像重构和阅片需求
主流	（1）用户打开本网站 （2）用户登录进入查询页面 （3）用户点击下载二维影像 zip 文件用于三维重建 （4）用户点击上传三维影像 obj、mtl、pdf 文件
后置条件	三维影像上传成功
附加流	当用户选择了非 obj、mtl、pdf 文件进行上传时提示用户上传错误；上传完成后若有未上传成功的文件提示用户上传失败的文件信息。

3.5 本章小结

本章首先对项目进行了可行性的分析，指出了本项目在经济、技术、社会和操作方面的可行性。然后对系统进行了非功能需求和功能需求分析，提出了功能和非功能要求。最后对系统的进行用例分析，提出机构用户和医生用户两种用户角色，并分别对这两种用户角色进行了用例分析，并给出相应的用例图加以描述。

第 4 章 医疗影像阅片系统前端概要设计

在软件过程中,概要设计是在需求分析阶段完成之后进行的。需求分析和概要设计是两个极其重要的阶段。需求分析,完成的是找到软件做什么的问题,概要设计要完成的是整体项目怎么做的问题[13]。

4.1 项目开发环境

本课题所研发的 LinkingDoctor 医疗影像阅片系统需要实现用户通过网络与服务器进行数据交互，且具有一定的并发量。因此本项目将结合本系统的开发技术，来给出本系统所需的运行环境。

项目采用 visio 作为绘图工具，使用 windows 10 操作系统，后台采用 Java 技技术进行开发，因此具有跨平台的特性。服务器端需要安装数据库管理工具，本系统使用较为流行的 Sql Server2008，开发使用 tomcat 服务器，数据库使用 MySQL，后台开发工具使用 Eclipse 。本项目全部使用 GitLab 进行版本控制管理，前端开发使用 webstorm 2016.3.2 作为开发工具，并通过 chrome 57.0.2987.133 浏览器进行页面调试。

4.2 项目总体设计

本项目是单页面网站，其通过使用 AngularJs 框架，运用 AngularJs 的 ui-route 库，在页面之间进行单页面切换从而实现页面跳转功能。项目中使用指令（directive）作为项目模块开发的基本单位，在页面中直接使用指令实现页面。指令中的 link 和 controller 作为页面功能实现的接口，指令的模板为页面进行布局和渲染。本项目中一共有四个页面，其页面跳转流程如图 4-1 所示：

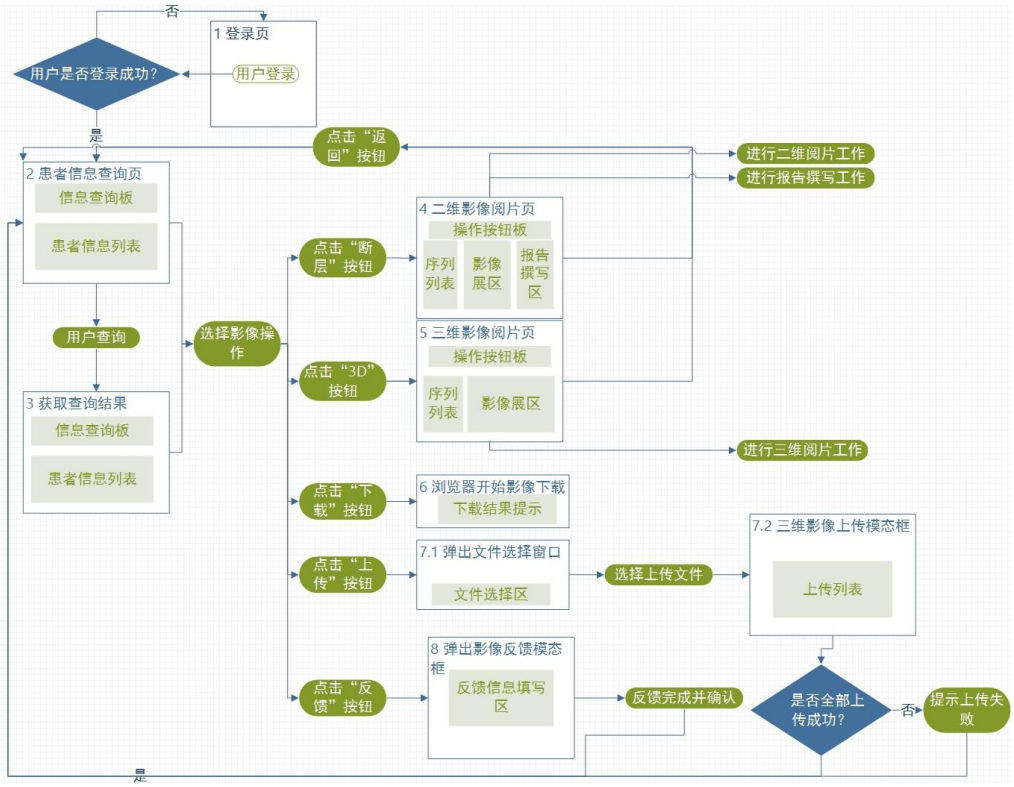


图 4-1 页面跳转流程图

本项目使用 AngularJs 框架，它是一种前端 MVC 框架。在运用 MVC 框架写出的项目中，视图（view）会从模型（model）中获取到数据，然后再展示给使用者，当用户通过键盘或鼠标与应用进行交互时，控制器（controller）会做出相应的响应，并操作修改模型当中的数据，然后数据的改变会使模型通知视图，从而改变视图，应用刷新，显示新的内容。MVC 非常灵活，使用 MVC 作为项目开发的设计框架会让应用更加易于扩展、维护和测试。选用 AngularJs 框架简化了代码编辑工作，也加快了页面渲染速度。

本项目将模块划分为用户登录模块、患者影像查询模块、二维影像阅片模块、三维影像阅片模块和阅片报告撰写模块共五个模块，其中患者影像查询模块中机构用户和医生用户的职能有所不同，机构用户能够确认二维影像上传并反馈病人信息，而医生用户能进行三维影像的上传并进行三维重建反馈。

项目中的模块划分如图 4-2 所示：

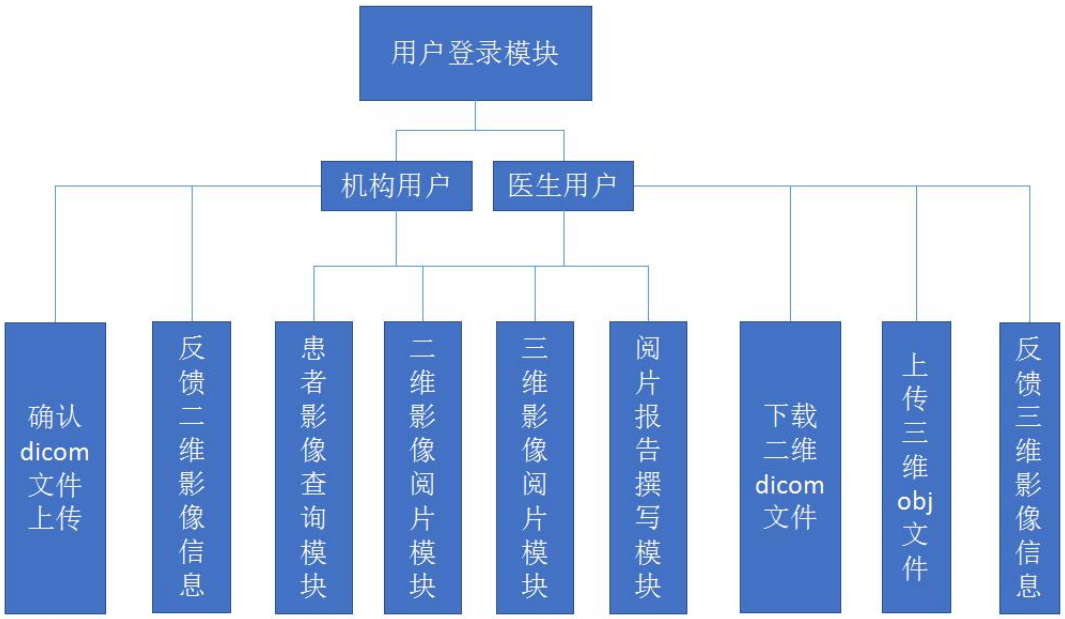


图 4-2 项目模块划分图

4.3 模块概要设计

4.3.1 登录模块

登录界面主要是作为机构用户和医生用户的入口，其页面跳转流程如图 4-3 所示：

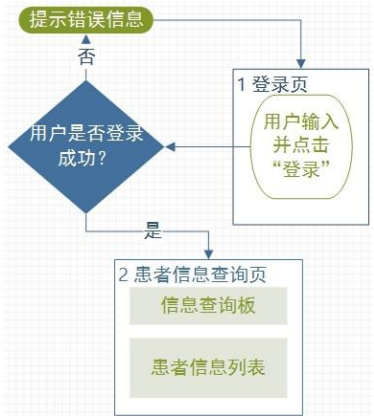


图 4-3 登录页面跳转流程图

用户登录模块只需要一个用户登录接口：UserLogin。

4.3.2 患者信息查询模块

用户界面是机构用户和医生用户进行影像上传、下载、信息反馈、信息查看以及影像查询和用户退出的界面，是平台登录后显示的主界面，其页面跳转流程图如图 4-4 所示：

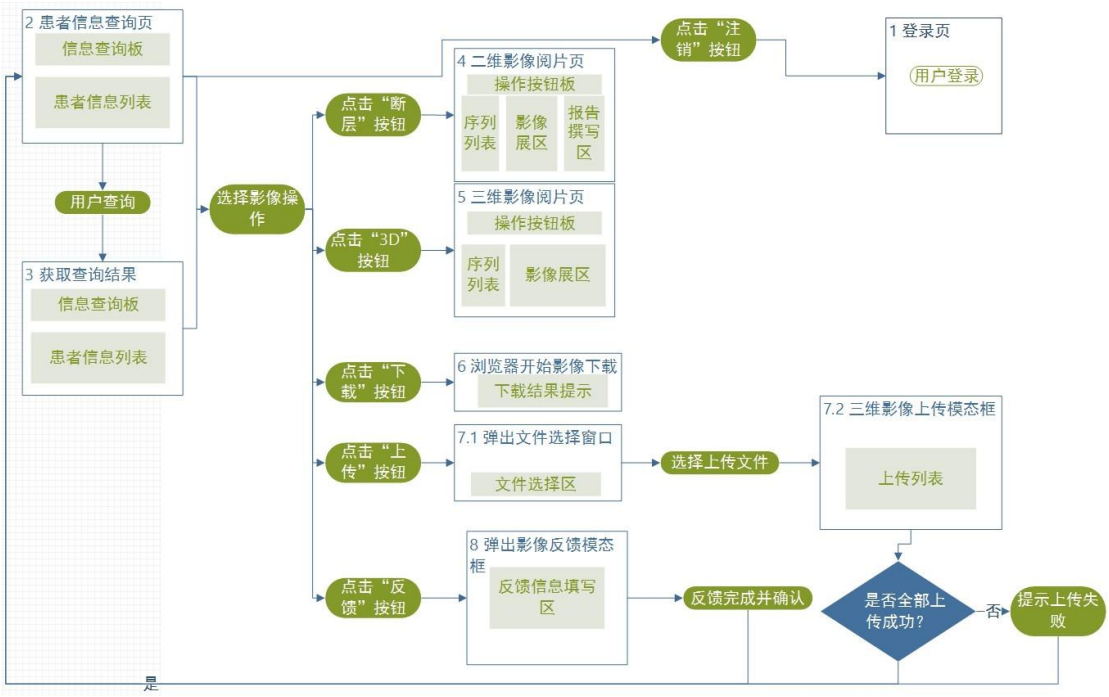


图 4-4 影像查询页面跳转流程图

患者查询模块需要通过在登录页进行的判断用户类型后获取相应的患者信息列表，在进行影像上传下载时需要在模态框内修改患者信息，在更改了患者信息后会更新患者信息列表和影像状态。患者信息查询模块的接口设计如下：

（1）影像获取接口：GetAsk；

（2）影像信息更新接口：UpdateImageInfo；

（3）更新影像接口：UpdateAsk；

（4）上传 3d 影像接口：Upload3dFile。

4.3.3 二维阅片模块

二维阅片界面中的阅片部分功能较多，是整个平台的重中之重，其包括的阅片功能比较丰富。其页面跳转流程图如图 4-5 所示：

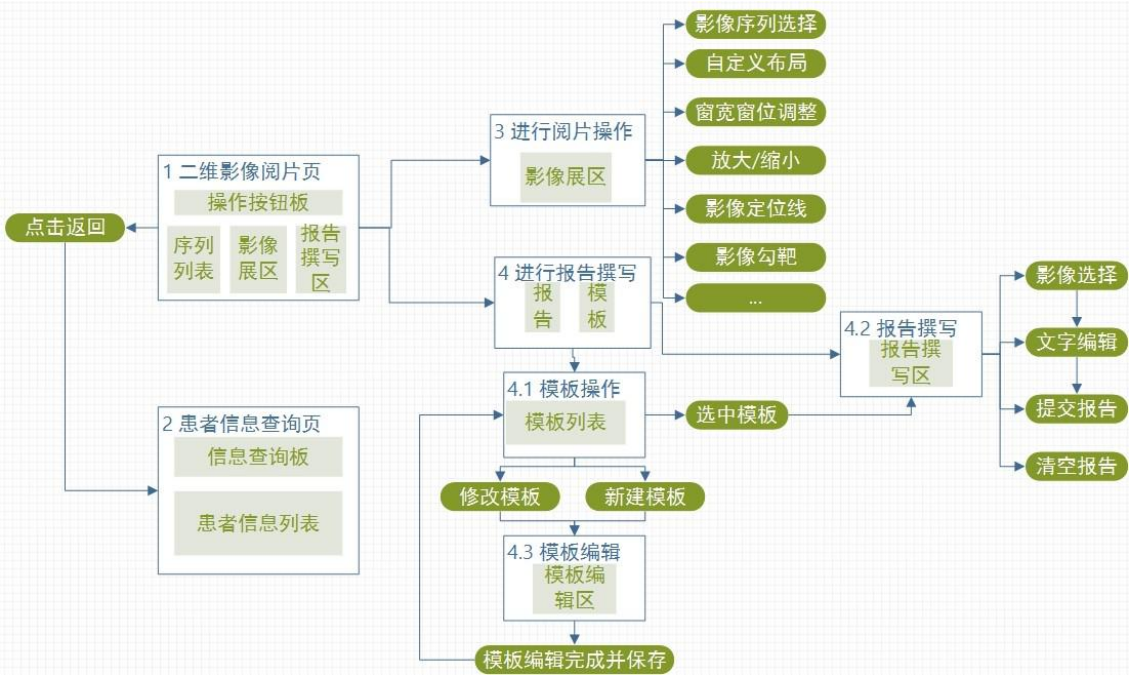


图 4-5 二维影像阅片页面跳转流程图

在进行二维影像阅片操作时，需要获取影像序列信息和单张 dicom 影像的信息。在进行阅片报告撰写时需要进行报告的提交和获取。二维影像阅片页面的接口设计如下：

（1）获取每个 Inatance 的 tag 信息的接口：GetFileMetaInfoServlet；

（2）获取 dcmurl 和 serverURL 的接口：GetConfig；

（3）获取病人基本信息的接口：GetPatientInfo；

（4）获取一次检查的所有序列的接口：getSeries；

（5）获取序列影像的接口：GetInstances；

（6）提交报告的接口：CommitReport；

（7）获取已存报告的接口：GetReport。

4.3.4 三维阅片模块

三维阅片界面是用户查看 obj 文件的三维影像时使用的界面，同时用户也能在三维阅片时查看此前上传的 PDF 报告文件。其页面跳转流程图如图 4-6 所示：

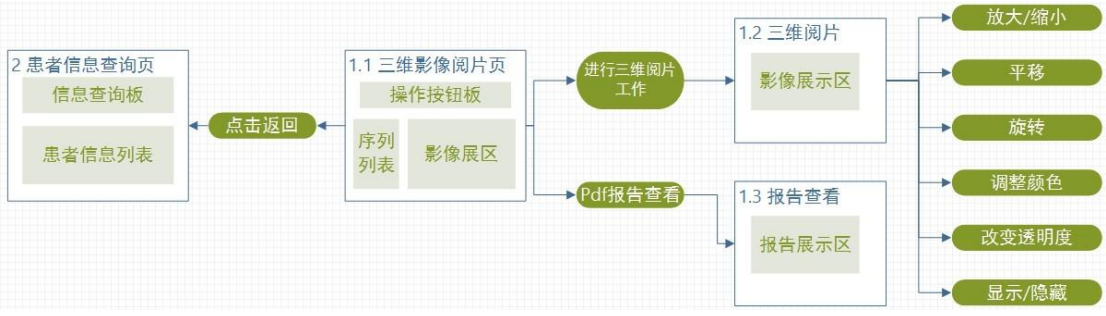


图 4-6 三维影像阅片页面跳转流程图

三维影像阅片页面的接口设计如下：

（1）获取后台文件地址的接口：GetFilesUrl；

（2）获取后台文件的接口：GetFiles。

4.4 本章小结

本章主要介绍了对项目的概要设计，包括对页面跳转流程的设计、实现项目使用的框架技术以及对前后端对接接口的概要设计。

第 5 章 医疗影像阅片系统前端详细设计与实现

5.1 登录模块

5.1.1 模块界面设计

机构用户和医生用户都在此处输入用户名和密码即可登录。登录页面的设计如图 5-1 所示：

图 5-1 用户登录界面风格

5.1.2 模块接口设计

用户登录模块只有一个用户登录接口'user/login'，其接口设计如表 5-1：

表5-1 用户登录模块接口设计表

接口名	接口描述	方法	请求路径	输入输出
UserLogin	用户登录时输入用户名和密码，点击登录	POST	'user/login'	向后台发送用户输入的用户名和密码，后台返回登录状态（成功、密码错误、用户不存在）

5.2 患者影像查询模块

5.2.1 模块界面设计

影像查询界面是用户登录平台后展现的主界面，主要用于患者影像信息的查询和进入影像阅片。此界面中，用户可以根据患者姓名、患者 ID、检查部位、检查设备、影像状态、上传日期进行患者影像的查询，查询到结果后，可以进行查看患者详情或进行患者信息反馈等工作，也可以进入二维阅片界面或者三维阅片界面进行阅片操作。影像查询界面的设计如图 5-2 所示：

图 5-2 用户界面风格

影像的查询工作可以通过多个选项进行筛选。在进行患者影像查询时，使用了 mbdatepicker.js 这一日期选择 js 库实现影像上传日期的选择功能，在选择影像类型、部位、状态时使用了 AngularJs 的选择框，直接给定选项数组就能实现用户选择功能，不需要写 js 进行数据绑定。查询时通过后台进行全局查询并向前台返回查询结果。其中查询时的关键代码如下：

```
"btn": [{
   "text": "查询",
   "method": (obj) => {
      searchChoose = {
         patientname: obj.text[0].choose || "",
         patientid: obj.text[1].choose || "",
         partsinstudy: obj.select[0].choose == "全选" ? "" : obj.select[0].choose,
         modality: obj.select[1].choose == "全选" ? "" : obj.select[1].choose,
         status: obj.select[2].choose == "已确认" ? 1 : obj.select[2].choose == "处理中" ? 2 :
obj.select[2].choose == "处理完成" ? 4 : "",
         uploadstartime: obj.picker[0].choose || "",
         uploadendtime: obj.picker[1].choose || "",
         pageNo: 1,
         pagesize: 20,
         username: username
      };
      if (role == 3) {
         obj.select[2].choose == "已上传" ? jigouGetList(searchChoose, {
               status: [0]
            }) : jigouGetList(searchChoose)} else if (role == 2) {
         doctorGetList(searchChoose);}
   }
}]
```

5.2.2 影像上传及模态框设计

三维影像上传和 pdf 报告文件上传时在选中指定上传文件后会弹出一个反应上传进度的模态框，如图 5-3 所示：

图 5-3 影像上传模态框

在影像上传的过程中，当用户选中了指定上传的 obj、mtl、pdf 文件后，项目中的压缩文件功能开始运作，将 obj 这一动辄上百兆的大文件进行无损压缩后再传到后台。压缩采用了 zip.js 这一 js 库，极为方便地实现了文件压缩工作，其中使用 zip.js 包进行文件压缩的关键代码如下：

```
function zipBlob(file,index, callback) {
    var fileName = file.name;
    zip.createWriter(new zip.BlobWriter(), function (writer) {
        writer.add(file.name, new zip.BlobReader(file), function () {
            writer.close(function (blob) {
                callback(blob, fileName,index)
            });
        }, function (currentIndex, totalIndex) {
            //console.log(currentIndex, totalIndex)
        });
    }, function (error) {
        console.log("zip Blob err in home.js" + error)
    });
};
```

5.2.3 模块接口设计

其接口设计如表 5-2 所示：

表5-2 影像查询模块接口设计表

接口名	接口描述	方法	请求路径	输入输出
GetAsk	根据用户信息获取对应患者影像列表	POST	'ask/getAsk'	用户登录时，向后台发送用户信名和密码，登录成功后跳转到影像查询页面获取后台返回的患者影像列表，包括患者的姓名、id、影像上传时间、

续表 5-2

接口名	接口描述	方法	请求路径	输入输出
GetAsk	根据用户信息获取对应患者影像列表	POST	'ask/getAsk'	影像类型、影像状态等。
UpdateImageInfo	用户根据当前影像状态进行影像信息反馈和影像信息的查看	POST	'ask/updateImageInfo'	用户在二维影像上传后向后台反馈或修改患者的基本信息和影像类型、临床诊断、基本病史等并提交给后台；在上传三维影像后用户需要进行三维重建反馈；用户还可以直接通过接口发出请求进行患者和影像信息的查看。
UpdateAsk	用户在进行影像确认和信息反馈时会改变影像的状态	POST	'ask/updateAsk'	用户进行影像确认和信息反馈工作会触发影像状态的更新接口，请求后台进行影像状态更新操作，并返回影像状态码。
Upload3dFile	用户通过点击“上传按钮”选择需要的文件，从而进行 3d 影像的上传操作	POST	'3dfile/upload3dFile'	用户将 3d 影像 obj 文件压缩处理后发送到后台，文件发送成功则显示“全部发送成功”，文件发送失败则显示发送失败文件信息，提示重新上传。

5.3 二维阅片模块模块

5.3.1 模块界面设计

二维阅片界面是用于多个系列影像同时阅片的界面。在此界面中功能丰富，不仅能进行阅片工作，还可以进行影像报告撰写。其中阅片操作包括影像的自定义展示、影像的放大缩小、影像平移、影像的旋转、影像长度测量、影像角度测量、影像点的定位、影像反色显示、影像窗宽窗位调整、影像还原、下标的显示与隐藏、影像定位线的显示、影像勾靶操作、勾靶线条去除、影像滚动播放和自动播放等。而报告撰写的操作包括报告模板的选择、新建、修改、影像的选择、患者情况的编辑、报告清空、报告提交等。二维阅片界面的设计如图 5-4 所示：

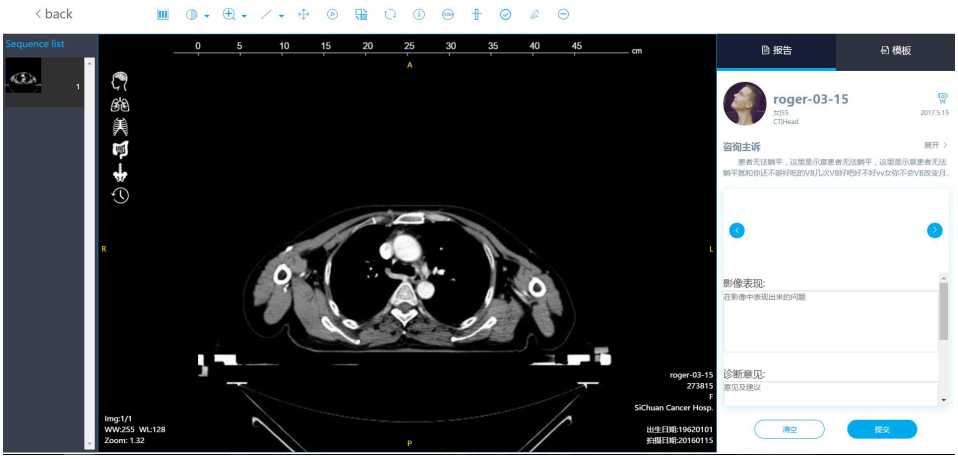


图 5-4 二维阅片界面

二维影像阅片中可以进行多种影像阅片操作，如图 5-5、5-6 所示：

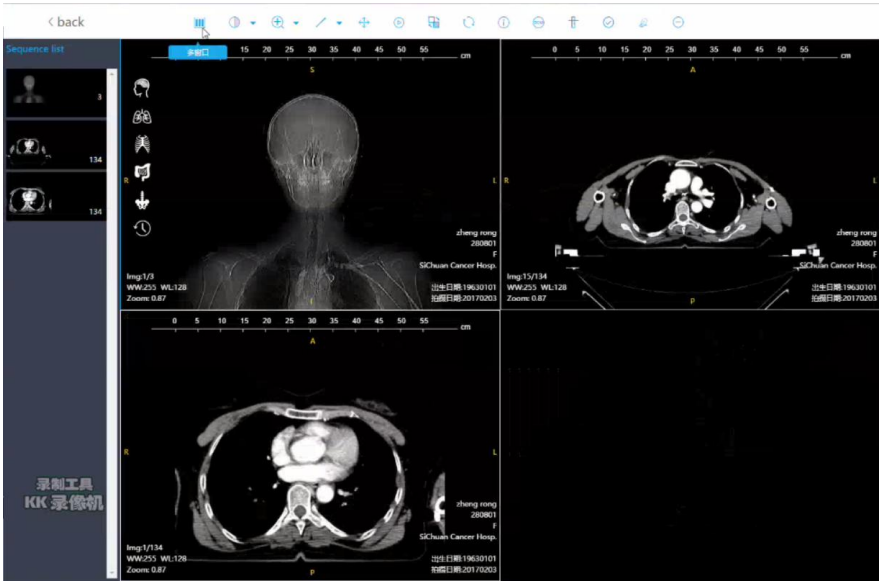


图 5-5 影像自定义布局

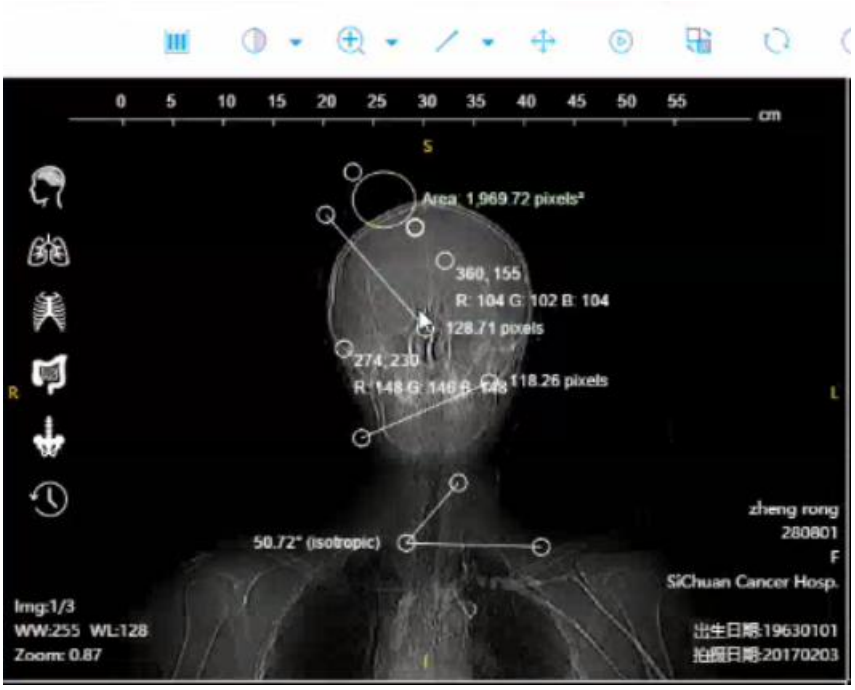


图 5-6 影像测量

二维影像阅片重点使用了操作 dicom 影像的 cornerstone.js，其丰富的功能函数满足了大多数影像操作的需求，但仍然有很多功能需要 DOM 操作，这是 cornerstone.js 满足不了的。其中关键的两个技术在下一节中会详细介绍。

点击左列的影像或点击获取 Dicom 图片和获取 jpg 图片时，右边的主界面会通过 ajax 进行替换并刷新的操作实现界面的局部刷新。

操作中的自定义布局需要构造一个空白表格，作为用户的布局选择框，当用户的鼠标悬停在选择框上时，选择框的 css 中设置 hover 样式，在鼠标移动时选择框颜色随鼠标位置改变，用户点击指定选框，触发 showSeries()函数进行界面重绘，界面布局随之改变，形成多视图阅片。

影像的滚动翻阅则是在触发鼠标滚动事件时对局部视图进行影像替换和刷新。

影像中的角标显示随着影像布局的改变而移动，这是通过将角标绑定到了 dicom 影像的父级 canvas 标签中实现的，当父级标签位置大小改变了，其子元素也就随之改变。而角标的隐藏与显示直接通过 jquery 中的 hide()和 show()函数实现。

5.3.2 模块参数设计

二维阅片模块中重要参数设置如表 5-3：

表5-3 二维阅片界面重要参数表

参数名	参数类型	注释
SeriesObj	Object	是当前页面所以影像信息的集合对象，其中包括患者信息 patientInfo、序列数 seriesNum、影像序列 imageSeries，其中 imageSeries 数组中有序列对象，其中包括影像数目 allPicNum、序列 id、序列信息 info、单张影像数组 images
imgBoxShowd	Array	当前展示的影像组的 id 数组
listObj	Array	二维阅片左侧列表对象数组，其中包含列表图片 imgUrl、序列描述 seriesDescription、序列中影像总数 numberOfInstances 和序列序号 seriseNumber

其中影像显示时的重要代码如下：

```
var imageId = imageIdforShow;
var element = dicomImage.get(0);
cornerstone.enable(element);
cornerstone.loadAndCacheImage(imageId).done(function (image) {
    cornerstone.displayImage(element, image);
}
```

5.3.3 模块接口设计

其接口设计如表 5-4 所示：

表5-4 二维阅片模块接口设计表

接口名	接口描述	方法	请求路径	输入输出
GetFileMetaInfoServlet	获取每个 Inatance 的 tag 信息的接口	POST	'GetFileMetaInfoServlet'	用户进入二维阅片界面后通过在查询界面点击列表后向后台发送患者的检查 id 从而获取每个序列的 tag 信息。
GetConfig	获取 dcmurl 和 serverURL 的接口	GET	'ImageViewer/getConfig'	获取 dicom 影像的地址和存储 dicom 影像的服务器的地址。
GetPatientInfo	获取病人基本信息的接口	GET	'image/' + imageId	获取当前选定的一次检查的病人信息。
getSeries	获取一次检查的所有序列的接口	GET	'getseries'	获取当前选定的一次检查的所有序列。
GetInstances	获取序列影像的接口	GET	'instances/getinstances'	获取当前选定的一次检查的所有影像。

5.3.4 影像勾靶的实现

我们在理解什么是靶区之前，先了解什么是肿瘤区。肿瘤区是指用一般诊断手段能确定出的肿瘤侵犯范围，它应有一定的形状和大小，并与临床分期标准一致。对转移到另一处的

肿瘤可认为是第二肿瘤区；对根治性手术切除的肿瘤可认为没有肿瘤区。而靶区，它比肿瘤区大，除包含肿瘤区外，还要包括有散在癌细胞存在的“亚临床灶”和肿瘤可能侵犯的范围。它是由主管医生根据病人的肿瘤分布情况、易扩散性和可能扩散的范围来确定的。因此，同一个肿瘤区可能出现两个以上的靶区：如乳癌中晚期，可能将腋窝和锁骨上作为第二、三靶区。[14]

影像勾靶是医生确认影像中靶区位置的操作，是癌症患者进行放疗前极为重要的一环，而在网页上进行靶区勾画需要克服一个难点，就是 DICOM 影像有其自己的坐标系，不能简单地运用 canvas 进行简单的勾画，在 canvas 上勾画的点坐标与 DICOM 影像中的点坐标是不重合的，那么勾画出来的靶区自然是错误的。为了克服这一难题，本项目中对绘制线条时的鼠标坐标进行了转换。

实现从鼠标坐标到对应像素点坐标的转换如下：

```
var pixelpoint = cornerstone.pageToPixel(element, e.clientX, e.clientY);
```

其中 cornerstone.pageToPixel()函数如下：

```
function b(b, c, d) {
    var e = a.getEnabledElement(b);
    if (void 0 === e.image)throw"image has not been loaded yet";
    var f = (e.image, b.getBoundingClientRect()), g = c - f.left - window.pageXOffset, h = d - f.top - window.pageYOffset, i = {
        x: g,
        y: h
    }, j = a.internal.getTransform(e);
    return j.invert(), j.transformPoint(i.x, i.y)
}
```

实现从像素坐标到 dicom 图像的 canvas 元素中相对坐标的转换：

```
var point = cornerstone.pixelToCanvas(element, {
x: instanceObj.checkTarget.lines[lineNum][0],
y: instanceObj.checkTarget.lines[lineNum][1]
});
```

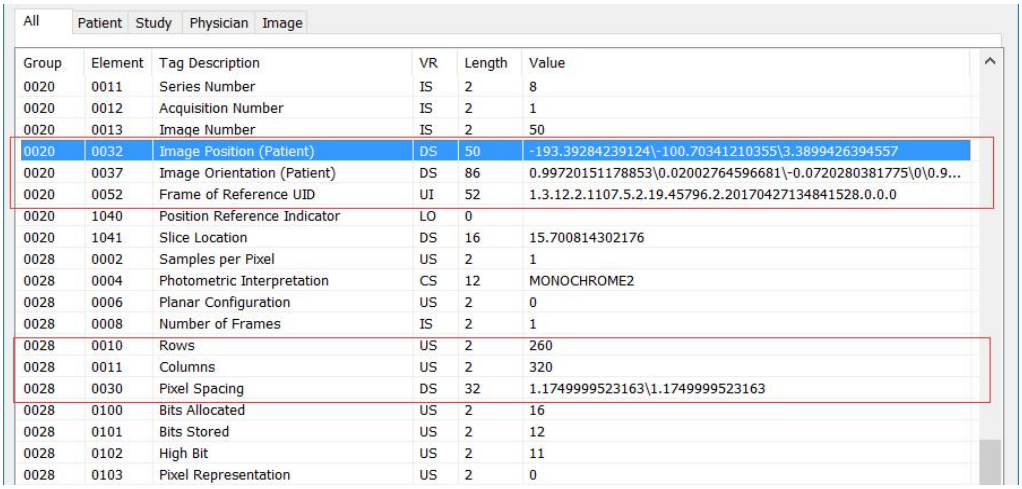
其中 cornerstone.pixelToCanvas()函数实现如下：

```
function b(b, c) {
    var d = a.getEnabledElement(b), e = a.internal.getTransform(d);
    return e.transformPoint(c.x, c.y)
}
```

在两个函数共有的 transformPoint()函数和 getTransform()函数是 cornerstone 中内置的用于坐标转换的函数，将页面位置转换为影像所在 canvas 中的像素点坐标。

5.3.5 影像定位线的绘制

在影像定位线绘制时，是通过对两张 DICOM 图像的坐标进行对比实现的。通过 sante DICOM Viewer 软件可以看到 DICOM 文件的信息，它包含了患者信息、影像坐标、像素空间、行列数等等。如图 5-7 所示：



All | Patient | Study | Physician | Image

Group	Element	Tag Description	VR	Length	Value
0020	0011	Series Number	IS	2	8
0020	0012	Acquisition Number	IS	2	1
0020	0013	Image Number	IS	2	50
0020	0032	Image Position (Patient)	DS	50	-193.39284239124\-100.70341210355\3.3899426394557
0020	0037	Image Orientation (Patient)	DS	86	0.99720151178853\0.02002764596681\-0.0720280381775\0\0.9...
0020	0052	Frame of Reference UID	UI	52	1.3.12.2.1107.5.2.19.45796.2.20170427134841528.0.0.0
0020	1040	Position Reference Indicator	LO	0	
0020	1041	Slice Location	DS	16	15.700814302176
0028	0002	Samples per Pixel	US	2	1
0028	0004	Photometric Interpretation	CS	12	MONOCHROME2
0028	0006	Planar Configuration	US	2	0
0028	0008	Number of Frames	IS	2	1
0028	0010	Rows	US	2	260
0028	0011	Columns	US	2	320
0028	0030	Pixel Spacing	DS	32	1.1749999523163\1.1749999523163
0028	0100	Bits Allocated	US	2	16
0028	0101	Bits Stored	US	2	12
0028	0102	High Bit	US	2	11
0028	0103	Pixel Representation	US	2	0

图 5-7 DICOM 文件信息

根据这些信息，可以对两张 DICOM 图片的交线进行计算。

如图 5-14 中，DICOM 文件信息中的 Row 是指影像中每一行中的像素点数，Columns 是指影像中每一列中的像素点数，Pixel Spacing 是指在每一行中和每一列中各个像素之间的物理距离，Image Position 是指影像的左上角第一个点的空间坐标，Image Orientation 是指影像第一行和第一列相对于病人的方向余弦，Frame of Reference UID 是指影像的参考系。

通过这些影像信息对于同一参考系的影像进行重合线条的绘制。计算方法如下：

（1）构造定位图空间坐标变换矩阵及其逆阵；

（2）通过方向余弦和影像起始点计算得到切片图四角位置和切片图位置；

（3）通过四角位置计算两张不同影像的交点；

（4）计算定位图上当前定位线的起点和终点坐标；

（5）在影像上绘制定位线。

构造定位图空间坐标变换矩阵及逆矩阵的关键代码如下：

```
for (var k = arr.length - 1; k >= 0; k--) {
    if (arrJS[k] != k) {
        for (kk = 0; kk < arr.length; kk++) {
            temp = swap([arr[k][kk], arr[arrJS[k]][kk]]);
            arr[k][kk] = temp[0];
            arr[arrJS[k]][kk] = temp[1];
        }
    }
    if (arrIS[k] != k) {
        for (kk = 0; kk < arr.length; kk++) {
            temp = swap([arr[kk][k], arr[kk][arrIS[k]]]);
            arr[kk][k] = temp[0];
            arr[kk][arrIS[k]] = temp[1];
        }
    }
}
```

5.4 报告撰写模块

5.4.1 模块界面设计

报告撰写模块主要有报告模板、和报告撰写，其中模板的选择功能如图 5-8 所示：

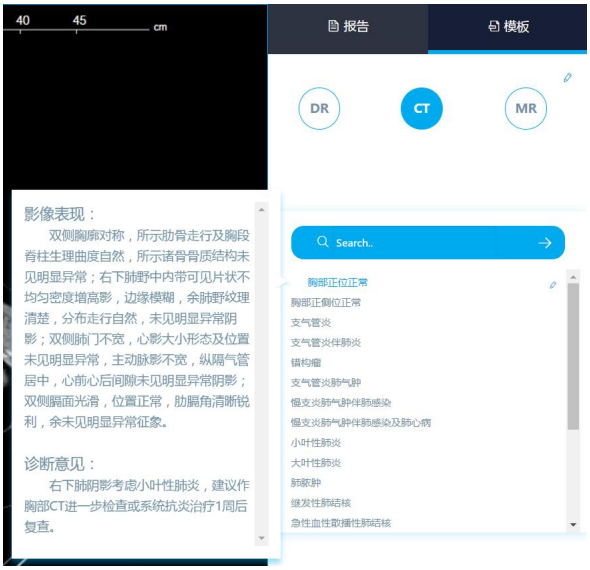


图 5-8 选择模板

模板的编辑功能如图 5-9 所示：

图 5-9 模板编辑

报告撰写时，报告可以进行影像选择和文字的编辑：

如图 5-10 所示：

图 5-10 报告模块

报告和报告模板的实现主要通过 AngularJs 框架，在模板的列表中，使用 ng-repeat 直接绑定模板数据，并使用 ng-filter 进行模板的搜索工作。模板详情和报告撰写慕课都采用了 DOM 操作进行部分展示。

其中，报告撰写时影像的选择是重点，将所选影像按照序列 id 和影像 id 进行顺序排列，显示在报告页中，其中选择影像时的重要代码如下：

```
$("#checkImage").click(function () {
    recoverButtonColor();
    clearAllIconsColor();
    for (var i = 0; i < imgBoxShowdNum; i++) {
        disableAllTools($("#dicomImageBoxId_" +
SeriesObj.imageSeries[imgBoxShowd[i]].id).children(0).get(0));
    }
    $('#checkImage').css({'color': 'white', 'background-color': '#00aaef'});
    var a = curImgBoxId.substring(23) - 1;
        if (SeriesObj.imageSeries[imgBoxShowd[0]].images[imgInsShowedArray[a]].checked ===
false) {
            SeriesObj.imageSeries[imgBoxShowd[0]].images[imgInsShowedArray[a]].checked =
true;
            drawChecked(curImgBoxId);
        } else {
            SeriesObj.imageSeries[imgBoxShowd[0]].images[imgInsShowedArray[a]].checked =
false;
            unbindDrawChecked(curImgBoxId);
        }
});
```

5.4.2 模块接口设计

其接口设计如表 5-5 所示：

表5-5 报告撰写模块接口设计表

接口名	接口描述	方法	请求路径	输入输出
CommitReport	提交报告的接口	POST	'report/commitReport?askid=' + $stateParams.askid	用户进入二维阅片界面后进行报告影像的选择和报告撰写操作，点击提交，将患者的影像表现、诊断意见和备注以及所选影像的 id 提交到后台的对应检查位置。
GetReport	获取已存报告的接口	GET	'report/getReport'	获取当前检查的报告内容，包括所选影像 id、影像表现、诊断意见和备注等。

5.5 三维阅片模块

5.5.1 模块界面设计

三维影像阅片界面是用于一组位于同一坐标中的三维影像显示和操作的界面。其中对三维影像进行的操作有：影像的显示、影像的旋转、影像的平移、影像的放大缩小、影像颜色的选择等，除此之外，还可以在该界面中查看此前上传的 pdf 报告。

三维阅片中可以进行 pdf 报告文件的查看，如图 5-11 所示：

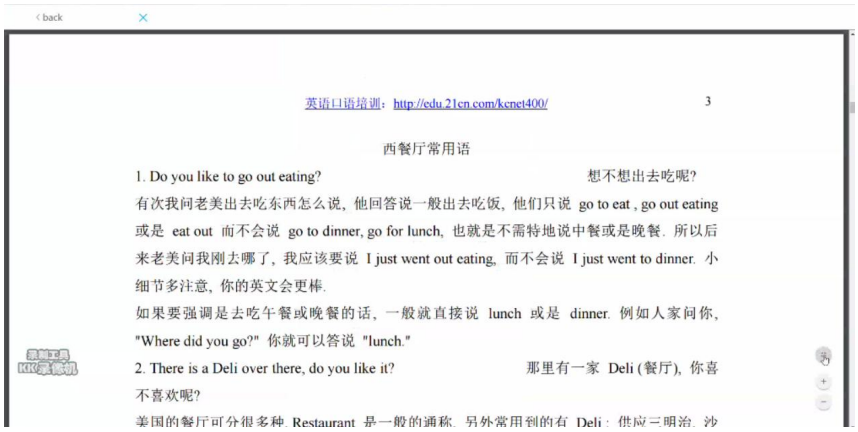


图 5-11 pdf 报告模块

此外，三维阅片还有许多功能操作，如图 5-12,5-13 所示：

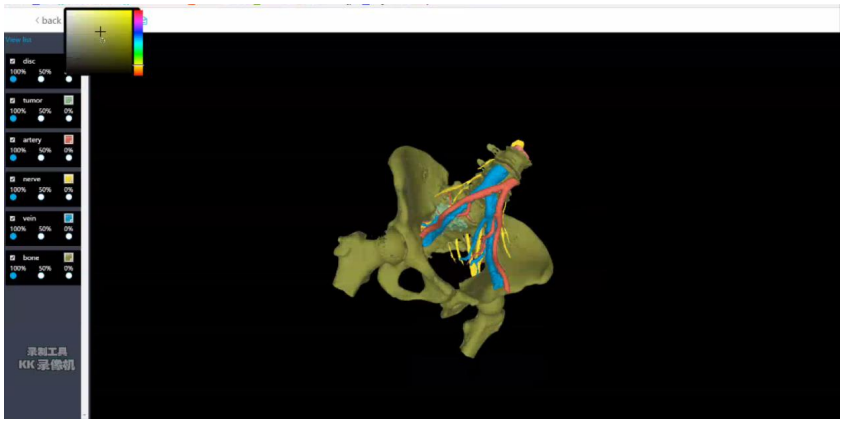


图 5-12 影像颜色改变

图 5-13 影像透明度改变

三维影像阅片重点使用了操作三维影像的 xtk.js，通过对 xtk.js 可以实现三维影像的放大缩小、影像旋转、影像平移。通过运用 xtk.js 的渲染原理可以改变影像颜色及透明度和显示与隐藏。

5.5.2 模块参数设计

三维阅片模块中重要参数设置如表 5-6：

表5-6 三维阅片界面重要参数表

参数名	参数类型	注释
ObjFilesGet	Array	是当前页面中所有 obj 文件对象的数组
mtlFilesGetUrl	Array	当前展示的影像组的默认 mtl 文件的获取地址数组，用于解析影像纹理、颜色，渲染影像。
ViewMesh	Object	当前展示影像数组的纹理颜色的数组对象

其中三维影像显示时的重要代码如下：

```
function read(files) {
    for (var i = 0; i < files.length; i++) {
        var fname = files[i].name.split(".")[0].toString().replace(" ", "");
        createData(fname);
        var f = files[i], _fileName = f.name,_fileExtension = _fileName.split('.').pop().toUpperCase();
        if (_fileExtension == _fileName.toUpperCase()) {
            _fileExtension = 'DCM';
        }
        if (_data[fname]['mesh']['extensions'].indexOf(_fileExtension) >= 0) {
            _data[fname]['mesh']['file'].push(f);
        }}
    var _types = Object.keys(_data[fname]), _numberOfFiles = files.length,_numberRead = 0;
    var loadHandler = function (type, file) {
        return function (e) {
            var data = e.target.result;
  _data[fname][type]['filedata'][_data[fname][type]['file'].indexOf(file)] = data;
            _numberRead++;
            if (_numberRead == _numberOfFiles) {parse(_data[fname]);}
```

```
    };
  };
var errorHandler = function (e) {};
   _types.forEach(function (v) {
      if (_data[fname][v]['file'].length > 0) {
         _data[fname][v]['file'].forEach(function (u) {
            var reader = new FileReader();
            reader.onerror = errorHandler;
            reader.onload = (loadHandler)(v, u);
            reader.readAsArrayBuffer(u);
         });}
```

其接口设计如表 5-7 所示：

表5-7 报告撰写模块接口设计表

接口名	接口描述	方法	请求路径	输入输出
GetFilesUrl	获取后台文件地址的接口	POST	'3dfile/get3dFile?askid=' + $stateParams.askid	用户进入二维阅片界面后通过 url 发送检查 id 到后台获取到存储在后台的 obj 文件、mtl 文件和 pdf 文件的文件路劲。
GetFiles	获取后台文件的接口	GET	serverIp+'/file/' + mtlFilesGetUrl[index].path + '?n=' +mtlFilesGetUrl[index].name	通过获取的文件路劲组装三维文件请求路劲向后台发送请求，获取压缩后的 obj 文件和 mtl、pdf 报告文件。

5.5.4 obj 文件加载的实现

在本项目中，影像下载速度一直是一个巨大的难题，由于 obj 文件动辄上百兆，为平台的流畅运行带来了难题，也让用户非常困扰，每次想要进行三维影像阅片时，等待时间都要几分钟甚至更长时间。

为了解决这一问题，本项目对 obj 文件进行了压缩处理上传文件，在请求 obj 文件时直接加载其压缩文件，进行解压缩操作，然后再显示影像。这样做的结果大大减少了文件加载的时间，仅需几十秒甚至几秒就能使影像加载完成。

5.6 本章小结

本章主要介绍了项目中各个模块的设计和实现，其中包括模块的界面设计、模块的参数设计、模块的接口设计、模块中的难点和解决方案等。其中对在项目实现过程中遇到的问题和解决方案进行了比较详细的说明。

第 6 章 医疗影像阅片系统前端系统测试

6.1 测试原则

随着用户对软件质量要求的不断提高，以及人们对软件质量的重视程度越来越高，软件测试在软件开发中的地位越来越重要[15]。

当项目开发人员已经将全部模块开发完成，各单元模块已经经过了开发人员代码检查和程序互测后，各单元模块集成在一起，进行独立测试。

项目测试是为了检测项目的需求有没有按照需求规格说明书中的结果输出，测试系统运行及业务处理是否与我们所期望的结果一致。

当所测内容都达到软件需求规格说明书的要求，所发现的问题也已全部解决。遗留问题中无致命性和严重性错误，且一般性错误不能超过总数的 2%，轻微性错误不能超过 8%，遗留问题总数不能超过问题总数的 10%时，测试可结束。当系统在部署期间或使用过程中发现严重错误或客户需求已发生重大变化时，测试可再启动。

6.2 测试环境

硬件环境：Intel(R) Core(TM) i7-6700HQCPU @ 2.60GHz 12.0GB DDR4 内存

软件环境：Windows10 家庭中文版

6.3 测试结果

6.3.1 功能测试

（1）“用户登录退出”测试

表 6-1“用户登录退出”功能测试

用例编号	UC-101			
功能描述	完成用户登录退出的操作			
用例目的	对用户登录退出进行功能测试			
用例前提	在此项目中用户登录情况下时测试退出			
子用例编号	输入/动作	期望的输出/响应	实际输出	状态
UC-101-01	用户填写正确的用户名和密码并点击登录，登录后点击退出。	用户登录成功，页面显示用户名； 用户点击退出时，回到登录界面。	页面显示用户名，用户点击退出时，回到登录界面。	正确
UC-101-02	用户填写错误格式的用户名和密码，用户点击登录。	页面根据用户输入给出相应的错误提示如用户不存在、密码错误。	输入为错误的账户名或密码时提示用户不存在或密码错误。	正确

（2）“用户影像查询”测试

表 6-2“用户影像查询”功能测试

用例编号	UC-102			
功能描述	完成用户查询影像的操作			
用例目的	对用户影像查询功能测试			
用例前提	在此项目中用户登录成功进行影像查询			
子用例编号	输入/动作	期望的输出/响应	实际输出	状态
UC-102-01	用户输入患者姓名、ID 并点击查询。	获得该姓名和 ID 对应的患者影像。	查询所得结果是用户输入的姓名和 ID 对应的患者影像。	正确
UC-102-02	用户选择检查部位、检查设备、状态、上传开始日期和上传结束日期并点击查询。	页面根据用户选择，显示出与用户查询信息相对应的影像。	查询结果是用户选择项对应的患者影像。	正确

（3）“影像的上传和下载”测试

表 6-3“影像的上传和下载”功能测试

用例编号	UC-103			
功能描述	完成用户上传和下载影像的操作			
用例目的	对用户上传和下载影像功能测试			
用例前提	在此项目中二维影像已从医疗设备上获取			
子用例编号	输入/动作	期望的输出/响应	实际输出	状态
UC-103-01	用户点击任一条影像操作区的下载按钮。	下载获得对应的影像 zip 文件。	通过下载得到了对应影像的 zip 文件。	正确
UC-103-02	用户点击对应患者的操作区的上传按钮，选择要上传的三维影像 Obj 文件或 PDF 文件并确认，上传文件时弹出模态框，点击完成。	页面弹出患者信息和上传文件信息的模态框，上传完成后点击完成则上传成功。	页面中弹出了上传文件信息的模态框，上传完成后点击完成，上传成功。	正确
UC-103-03	用户点击对应患者的操作区的上传按钮，选择 jpg 文件并确认。	页面弹出模态框，提示不能上传 jpg 格式的文件，请重传。	页面弹出了模态框，下方提示不能上传 jpg 格式的文件，请重传。	正确

（4）“二维影像阅片”测试

表 6-4“二维影像阅片”功能测试

用例编号	UC-104			
功能描述	完成用户二维影像阅片的操作			
用例目的	对用户二维影像阅片功能测试			
用例前提	在此项目中用户从用户界面点击断层进入二维影像阅片界面			
子用例编号	输入/动作	期望的输出/响应	实际输出	状态
UC-104-01	用户点击自定义布局按钮选择希望的布局。	影像根据用户自定义布局进行排列。	影像根据用户选择的布局进行排列显示。	正确
UC-104-02	用户在默认情况下在影像上滚动鼠标。	影像序列进行影像更新显示。	影像序列进行影像更新显示。	正确
UC-104-03	用户选择调整窗宽窗位按钮，按住鼠标左键在影像上移动。	影像的窗宽窗位改变； 影左下角数字显示当前窗宽窗位。	影像的窗宽窗位改变，其左下角数字显示当前窗宽窗位。	正确
UC-104-04	用户选择局部对比度按钮，按住鼠标左键在影像上移动。	移动鼠标时，影像上出现矩形影像出现选择框； 所选择的区域对比度明显。	移动鼠标时，影像上出现矩形影像出现选择框，所选择的区域对比度明显。	正确
UC-104-05	用户选择放大缩小按钮，按住鼠标左键在影像上移动。	移动鼠标时，鼠标向上移动，影像变小，鼠标乡下移动，影像变大。	移动鼠标时，鼠标向上移动，影像变小，鼠标乡下移动，影像变大。	正确
UC-104-06	用户点击左旋按钮。	影像逆时针转转 90 度。	影像逆时针转转 90 度。	正确
UC-104-07	用户选择上下翻转按钮。	影像上下 180 度翻转。	影像上下 180 度翻转。	正确
UC-104-08	用户选择左右翻转按钮。	影像左右 180 度翻转。	影像左右 180 度翻转。	正确
UC-104-09	用户选择长度按钮，按住鼠标左键移动。	出现以鼠标左键按下作为起点和以鼠标左键抬起作为终点的直线； 显示直线长度。	出现以鼠标左键按下作为起点和以鼠标左键抬起作为终点的直线，并显示直线长度。	正确
UC-104-10	用户像素值测量按钮，按住鼠标左键移动。	鼠标左键抬起位置出现一个小圆圈； 显示该点的 x 和 y 轴位置和 rgb 值。	鼠标左键抬起位置出现一个小圆圈并显示该点的 x 和 y 轴位置和 rgb 值。	正确

续表 6-4

子用例编号	输入/动作	期望的输出/响应	实际输出	状态
UC-104-11	用户选择椭圆框按钮，按住鼠标左键移动。	出现以鼠标左键按下作为起点和以鼠标左键抬起作为终点的椭圆框； 显示椭圆框的像素面积。	出现以鼠标左键按下作为起点和以鼠标左键抬起作为终点的椭圆框并显示椭圆框的像素面积。	正确
UC-104-12	用户选择角度测量按钮，按住鼠标左键移动，放开左键后再次点击。	出现以鼠标左键按下作为起点和以鼠标左键抬起作为终点的直线再次点击鼠标左键后出现一个角；显示该角的角度。	出现以鼠标左键按下作为起点和以鼠标左键抬起作为终点的直线再次点击鼠标左键后出现一个角并显示该角的角度。	正确
UC-104-13	用户选择移动按钮，按住鼠标左键移动。	影像跟着鼠标的移动而移动。	影像跟着鼠标的移动而移动。	正确
UC-104-14	用户选择反色按钮。	影像的颜色变成反色。	影像的颜色变成反色。	正确
UC-104-15	用户选择重置按钮。	影像恢复未操作时的状态。	影像恢复未操作时的状态。	正确
UC-104-16	用户选择角标按钮再次选择角标按钮。	影像左下角和右下角的角标消失；再次选择角标按钮时，角标出现。	影像左下角和右下角的角标消失，当用户再次选择角标按钮时，角标出现。	正确
UC-104-17	用户选择获取 dicom 图片按钮，用户再选择获取 jpg 图片按钮。	影像显示 dicom 图片； 获取 dicom 图片的按钮变成获取 jpg 图片按钮； 用户再选择获取 jpg 图片按钮，影像显示 jpg 图片； 获取 jpg 图片的按钮变成获取 dicom 图片按钮。	影像显示 dicom 图片，获取 dicom 图片的按钮变成获取 jpg 图片按钮.用户再选择获取 jpg 图片按钮，影像显示 jpg 图片,获取 jpg 图片的按钮变成获取 dicom 图片按钮。	正确

续表 6-4

子用例编号	输入/动作	期望的输出/响应	实际输出	状态
UC-104-18	在选择了两张图以上布局的情况下，用户选择定位线按钮。	滚动其中一张图片，另一张图片中的定位线随着滚动图片的变化而移动。	滚动其中一张图片，另一张图片中的定位线随着滚动图片的变化而移动。	正确
UC-104-19	用户选择勾靶按钮，按住鼠标左键移动。	出现随着鼠标移动轨迹绘制的红色线条。	出现随着鼠标移动轨迹绘制的红色线条。	正确
UC-104-20	用户选择删除勾靶线按钮，在勾靶线上点击鼠标左键。	所点击的勾靶线消失。	所点击的勾靶线消失。	正确

（5）“三维影像阅片”测试

表 6-5“三维影像阅片”功能测试

用例编号	UC-105			
功能描述	完成用户三维影像阅片的操作			
用例目的	对用户三维影像阅片功能测试			
用例前提	在此项目中用户通过在用户界面点击 3D 按钮进入三维影像阅片界面			
子用例编号	输入/动作	期望的输出/响应	实际输出	状态
UC-105-01	等待影像加载完成后，勾选对应影像的选择框。	三维影像在界面上加载显示出来。	三维影像在界面上加载显示出来。	正确
UC-105-02	用户点击影像的颜色选择框选择影像颜色。	三维影像变成用户选择的颜色。	三维影像变成用户选择的颜色。	正确
UC-105-03	用户点击影像透明度选择全透明、半透明、不投吗。	对应的三维影像变成对应的透明状态。	对应的三维影像变成对应的透明状态。	正确
UC-105-04	用户按住鼠标左键并移动鼠标。	三维影像随鼠标移动而旋转。	三维影像随鼠标移动而旋转。	正确
UC-105-05	用户按住鼠标中键并移动鼠标。	三维影像随鼠标移动。	三维影像随鼠标移动。	正确
UC-105-06	用户滚动鼠标。	三维影像随鼠标滚动而放大缩小。	三维影像随鼠标滚动而放大缩小。	正确

续表 6-5

子用例编号	输入/动作	期望的输出/响应	实际输出	状态
UC-105-07	用户点击 PDF 报告按钮；用户点击退出 PDF 报告按钮。	界面显示 PDF 报告；界面退出 PDF 报告显示并显示三维影像。	界面显示 PDF 报告点击退出 PDF 报告后，界面显示并显示三维影像。	正确

（6）“报告撰写”测试

表 6-6“报告撰写”功能测试

用例编号	UC-106			
功能描述	完成用户报告撰写的操作			
用例目的	对用户报告撰写功能测试			
用例前提	在此项目中用户从用户界面点击断层进入二维影像阅片界面			
子用例编号	输入/动作	期望的输出/响应	实际输出	状态
UC-106-01	用户点击选中图片按钮。	报告中出现选中的图片。	报告中出现选中的图片。	正确
UC-106-02	用户选择模板界面。	页面出现报告模板界面。	页面出现报告模板界面。	正确
UC-106-03	用户选择指定模板进行编辑。	页面出现。	页面出现空模。	正确
UC-106-04	指定模板进行编辑。	页面出现模板编辑界面。	页面出现模板编辑界面。	正确
UC-106-05	用户选择指定模板。	页面出现报告撰写界面并含有选中模板的内容。	页面出现报告撰写界面并含有选中模板的内容。	正确
UC-106-06	用户输入模板名称搜索模板。	页面出现用户搜索对应的模板。	页面出现用户搜索对应的模板。	正确
UC-106-07	用户撰写报告后点击清空。	报告界面清空。	报告界面清空。	正确
UC-106-08	用户撰写报告后点击提交。	报告提交到后台，刷新页面后报告仍在。	报告提交到后台，刷新页面后报告仍在。	正确

6.3.2 性能测试

（1）二维影像加载性能测试

表 6-7 二维影像加载性能测试

用例编号	UC-201
性能描述	完成二维影像加载的性能
用例目的	对二维影像加载用时进行测试
用例前提	在此项目中用户点击“断层”或“断层图像”按钮

续表 6-7

子用例编号	输入/动作	实际输出
UC-201-01	用户点击“断层”或“断层图像”按钮，dicom 影像加载 100 张	二维影像加载平均用时 11.16s
UC-201-02	用户点击“断层”或“断层图像”按钮，dicom 影像加载 200 张	二维影像加载平均用时 17.54s
UC-201-03	用户点击“断层”或“断层图像”按钮，dicom 影像加载 300 张	二维影像加载平均用时 26.23s
UC-201-04	用户点击“断层”或“断层图像”按钮，dicom 影像加载 400 张	二维影像加载平均用时 39.24s
UC-201-05	用户点击“断层”或“断层图像”按钮，dicom 影像加载 500 张	二维影像加载平均用时 48.62s

由测试结果进行性能分析，加载一张 dicom 影像平均用时为 139.2ms，事物成功率 100%，测试结果在用户可接受范围内，满足系统性能要求。

（2）三维影像加载性能测试

表 6-8 三维影像加载性能测试

用例编号	UC-202	
性能描述	完成三维影像加载的性能	
用例目的	对三维影像加载用时进行测试	
用例前提	在此项目中用户点击“3D”或“3D 图像”按钮	
子用例编号	输入/动作	实际输出
UC-202-01	用户点击“3D”或“3D 图像”按钮，obj 影像加载共 20 兆	三维影像加载平均用时 5.23s
UC-202-02	用户点击“3D”或“3D 图像”按钮，obj 影像加载 40 兆	三维影像加载平均用时 5.91s
UC-202-03	用户点击“3D”或“3D 图像”按钮，obj 影像加载 60 兆	三维影像加载平均用时 7.46s
UC-202-04	用户点击“3D”或“3D 图像”按钮，obj 影像加载 80 兆	三维影像加载平均用时 9.87s
UC-202-05	用户点击“3D”或“3D 图像”按钮，obj 影像加载 100 兆	三维影像加载平均用时 17.77s

由测试结果进行性能分析，加载一兆三维影像平均用时为 241ms，事物成功率 100%，测试结果在用户可接受范围内，满足系统性能要求。

6.4 本章小结

本章主要介绍了项目实现后进行的项目测试工作，首先介绍了软件测试原则，并说明了项目测试时的系统环境，制作了足够的测试用例进行用例测试工作并仔细填写了用例表。

第 7 章 结论

系统运用了 MVC、AngularJs、Dicom、Cornerstone、xtk、webpack 等关键技术，其中系统主要运用了 AngularJs 这一前端 MVC 框架作为系统开发框架，同时结合 jquery 进行部分网页的开发，系统关键网页中运用了 Dicom 医学影像协议，并通过 Cornerstone.js 和 Xtk.js 进行医学影像的处理操作，最后整个系统使用了 Webpack 作为模块加载器和打包工具，使用 git 作为版本控制管理工具。

在进行系统的设计和开发前，首先明通过实地调研明确了系统的非功能性需求和功能需求，非功能需求方面，从经济、技术、操作和社会这四个方面进行了着重分析，保证了该 Web 应用系统的可行性。功能需求方面，详细设计每个页面的功能点，充分考虑了用户痛点和必要需求，且抽象出了系统的业务流程图。通过对系统的需求分析，整理出清晰明朗的系统所需功能。为后期开发做好了准备工作。

在进行本次系统的设计时，选择了恰当的系统开发环境，然后根据之前列出的系统所需的功能进行由简入繁的业务流程设计。在进行详细的系统设计时，首先是根据之前设计的业务流程和功能点进行了页面设计和用户交互设计，然后将系统的业务流程分模块进行整理汇总，抽象出各个模块，进行详细的功能设计。除此之外，还对两个关键的技术难题进行了分析，提出了解决方案并展示关键代码。据此，整个系统基本完成。

系统也存在一些亟待解决的性能问题，比如以下问题：

（1）影像的加载速度比较慢，有时会因为数据传输量过大导致网页崩溃；因为在系统中使用的医学影像无论在文件大小上还是数据数量上都比较大，有时一个三维影像就有上百兆，而二维医学影像有时也有上百张，所以对网页和网速以及代码的稳定性都是一个很强的考验，尽管我在传输数据时进行了压缩和解压缩工作，但是影像的下载仍然需要较长的时间。

（2）在代码编写方面，系统中的部分代码的复用率比较低，重写代码造成整个系统存在一些冗余现象；

（3）系统中由于使用的技术都比较新，版本较高，所以对浏览器的性能要求也相对比较高，因此该应用的浏览器兼容不太好。系统的开发都是在 Chrome 浏览器上进行的，对于火狐浏览器，该系统还能做到较好的兼容，但是由于 IE 浏览器不能很好地支持一部分页面特效，所以在 IE 浏览器端看到的页面效果会与 Chrome 浏览器存在一些差距。

对于这些问题，系统有待进一步改进，完善代码并提高本 Web 应用的性能。

希望在以后的工作学习中对上述问题进行深一步的研究，逐渐完善系统功能，提升应用的稳定性和兼容性。

参考文献

[1] 杨眉, 简卓为, 李文璞,等. 多媒体医学影像系统[J]. 现代电子技术, 2000(7):75-76.
[2] 格林, 夏德瑞, 大漠穷秋. 用 AngularJS 开发下一代 Web 应用[J]. 中国科技信息, 2013(23):90-90.
[3] AriLerner, 勒纳, 赵望野,等. AngularJS 权威教程[M]. 人民邮电出版社, 2014.
[4] David A. Clunie. DICOM Structured Reporting and Cancer Clinical Trials Results[J]. Cancer Informatics,2007,4(Imaging):.
[5] S Babhalgaonkar,MV Waghmare. Image Analysis of Mammographic DICOM Images[J]. Volume 3, Issue ICASE13, April 2013, ISSN Online: 2277-2677.
[6] 何斌, 金永杰, 李玉兰. 按照 DICOM 标准制定核医学图像文件格式[J]. 核电子学与探测技术, 2001(6):440-443.
[7] 黄志聪, 庄天戈. DICOM 标准的发展及最新的变化[J]. 中国医疗器械杂志, 2004, 28(3):203-207.
[8] 梁存升, 冯骥. DICOM 标准分析及其应用[J]. 中国医学装备, 2006, 3(2):18-20.
[9] 全海英, 全海英, 杨源,等. DICOM 数据集与 DCM 文件格式[J]. 计算机应用, 2001, 21(z1):145-146.
[10] Subramanian V. Modularization and Webpack[M]. Pro MERN Stack. Apress, 2017.
[11] 叶斌. 软件开发中的用例分析技术[J]. 计算机技术与发展, 2004, 14(9):118-121.
[12] 胡树玮, 张修如. 使用用例分析技术捕获需求[J]. 计算机技术与发展, 2005, 15(7):4-6.
[13] 李亚. 面向对象软件概要设计过程[J]. 福建电脑, 2008, 24(6):48-49.
[14] CSDN 博客. 什么是肿瘤区、靶区、治疗区和照射区?http://blog.csdn.net/moyumoyu/article/details/8491621
[15] 范勇，兰景英，李绘卓.软件测试技术[M].西安电子科技大学出版社, 2009.

谢辞

时光荏苒，光阴似箭，一转眼，大学四年就这样过去了，这四年里，我过得十分充实。这学期，我的毕业设计和毕业论文是在繁忙的工作状态下完成的，这是我大学四年来学习生活的总结，是我学习专业知识所得成果的结晶。在此，我要感谢在我制作我的毕业设计这段时间以来，帮助我，关心我和支持我的老师、同学还有亲人和朋友们。

本毕业论文是我在我的指导老师，刘畅老师的指导下完成的，从我的课题名称的确定到任务书的撰写、修改我的开题报告，再到毕业论文的撰写时需要注意的格式、需要突出的内容和要解释清楚的概念等等，刘畅老师都事无巨细地指点我。在生活、学习、工作等各方面，刘畅老师也很关心我和其他同学，和我们相处十分亲切，甚至刘老师还帮着我去打印任务书。在此，我对刘畅老师报以最诚挚的感谢！

此外，我还要感谢我的同学和实习期间的同事们，在我进行项目编写的时候，从技术方便为我提供帮助，耐心地给我讲解我不熟悉的知识，还从我项目的各方面给我提出建议，包括业务逻辑、界面交互等等，是他们对我的帮助让我得以顺利地完成我的代码编辑工作，完成我的毕业设计。我真的非常感谢这群好朋友。

另外，我想感谢我的爸爸妈妈，这一路上，他们默默支持我的学习、工作，为我加油打气，给我以精神支持和经济支持，让我在学习工作之外没有后顾之忧，我才能安心地全心地去完成我的毕业设计。

最后，我还想感谢一下我的辅导员，周诚毅老师和甘佳学姐还有以前给我上过课的所有老师，是在他们的教导下才有了今天的我，他们不仅教给我文化知识，还教给我心灵的良药，做人的道德品质。我的任课老师中，特别是蒋勇老师，他教授我们班的编译原理课程，虽然他平时非常严厉，几乎所有的学生都怕他，但是我觉得我对他有着各位的亲切感和尊敬。他对待工作的认真负责，对学生的严格要求，都让我从心底里得到了很大的启发。