



华南师范大学

本科毕业论文

论文题目: 基于 Vue.js 的网络教育学院资产管理系统

指导老师: 单志龙

学生姓名: 罗玉珠

学号: 20161180147

学院: 计算机学院

专业: 计算机科学与技术

毕业时间: 2021.06

基于 Vue.js 的网络教育学院 资产管理系统

摘 要

随着科技的发展，数字化办公给人们的工作带来越来越多的便利，它可以帮助人们存储大量的办公数据，快速处理办公事务。目前网络教育学院的资产管理尚处于线下状态，需要工作人员整理并打印资产申请单，交由部门领导和主管领导审批，在党政会议通过后由采购人员进行采购。该过程复杂繁琐，增加了资产管理的时间人力成本。因此我们开发了一套基于 Vue.js 的网络教育学院资产管理系统。

网络教育学院资产管理系统使用前后端分离的开发模式，前端使用 Vue.js 框架进行开发，使用 axios 与后端进行通信，在页面设计上使用 Element UI 组件库进行页面的构建。根据网络教育学院教职工的需求，我将使用系统的用户分为 4 类，分别是管理员、普通用户、部门领导以及主管领导，按账号类别进行功能模块的划分。通过账号类别功能的划分，系统可以将申请单从创建到经历层层审批的过程转移到线上执行，大大提高了资产申请的速度。

关键词：资产管理；数字化办公；Vue.js；Element UI

VUE.JS-BASED ASSET MANAGEMENT SYSTEM FOR NETWORK EDUCATION COLLEGES

ABSTRACT

With the development of science and technology, digital office brings more and more convenience to people's work. It can help people store a large amount of office data and quickly process office affairs. At present, the asset management of the Network Education Institute is still in an offline state. It requires staff to sort out and print the asset application form, and submit it to department leaders and supervisors for approval. After the party and government meeting is approved, the procurement staff will purchase. This process is complicated and cumbersome, which increases the time and labor cost of asset management. Therefore, we have developed a set of Vue.js-based asset management system for online education colleges.

The asset management system of the Network Education Institute uses a development model that separates the front and back ends. The front end uses the Vue.js framework for development, axios is used to communicate with the back end, and the Element UI component library is used to build the page in the page design. According to the needs of the faculty and staff of the Network Education Institute, I divide the users who use the system into 4 categories, namely, administrators, ordinary users, department leaders, and supervisor leaders. The functional modules are divided according to account types. Through the division of account types and functions, the system can transfer the application form from creation to through layers of approval to online execution, which greatly improves the speed of asset application.

Key Words: asset management; digital office; Vue.js; Element UI

目录

摘 要	I
1. 绪论	1
1.1 选题背景.....	1
1.2 选题意义.....	1
2. 可行性分析.....	3
2.1 技术可行性.....	3
2.1.1 前端开发语言分析.....	3
2.1.2 Vue.js 框架分析	4
2.1.3 Element UI 组件库分析	4
2.1.4 前后端分离架构分析.....	4
2.1.5 MVVM 框架及开发模式优势	4
2.1.6 axios 技术	5
2.2 经济可行性.....	5
2.3 社会可行性.....	5
2.4 开发可行性.....	6
2.5 操作可行性.....	6
3. 需求分析.....	7
3.1 功能需求.....	7
3.2 安全需求.....	8
3.3 性能需求.....	8
4. 总体设计.....	9
4.1 系统结构设计.....	9
4.2 系统功能设计.....	9
5. 详细设计.....	12
5.1 系统开发环境.....	12
5.2 通用组件的开发.....	12
5.3 数据库设计.....	12
5.4 系统功能模块实现.....	15
6. 测试	22
6.1 功能测试.....	22

6.2 安全测试.....	24
8. 参考文献.....	27
附录	28
致谢	32

1. 绪论

1.1 选题背景

由于日常工作需要，网络教育学院经常需要购买大量物资来供相关活动和课程使用，目前网络教育学院还是使用线下审批的形式，审批流程如图 1-1，即整理并打印物资申请单，将申请单交给部门领导审批，部门领导通过则在申请单上签字同意，部门领导同意后再将申请单交给主管领导审批，主管领导审批同意后签字，由工作人员将申请的物资分配给相关人员去采购。由此可见，这样的流程不仅繁琐而且还受到时间空间的限制，需要提交申请单的人和各个审批领导都在场时才能够较顺利地通过审批进入采购阶段，这中间还可能出现领导审批不通过驳回申请单的情况，这样就造成了许多人力物力以及时间的浪费。随着科技的发展，越来越多的公司单位已采取数字化办公的形式来代替纸质文件层层提交的流程，因此，网络教育学院也需要一个线上资产管理系统，该系统能够满足所有线下层层审批一直到采购完成的要求，为网络教育学院的教职工带来一个稳定的、界面友好的系统。

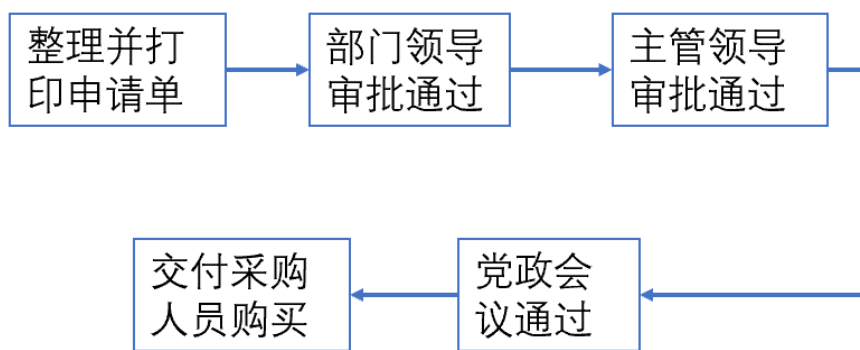


图 1-1 网络教育学院资产审批流程

1.2 选题意义

通过对网络教职工的访谈，我们决定打造一个网络教育学院资产管理平台。

网络教育学院资产管理系统旨在为网络学院的教职工提供一个线上物资采购管理平台，针对网络教育学院教职工的需求，该系统必须囊括工作人员整理物资、提交物资申请单、部门领导审批、主管领导审批、采购人员购买等一系列流程，并且在此基础上能够有一个操作友好，稳定的界面。即便申请人、部门领导、主管领导和采购人员不在同一场合时也能够完成各自在资产审批中的任务，使得资产从形成申请单到购买等一系列流程的时间大大缩短，为网络教育学院的教职工带来工作上的便利。

2. 可行性分析

2.1 技术可行性

网络教育学院资产管理系统的开发使用 JavaScript、html、css 三种编程语言，使用 MVVM 框架来进行开发，保证逻辑层与视图层的分离。在此基础上使用轻量级的 Vue.js 框架进行开发，引入 vue-router 插件来进行单页面应用的构建，使用 axios 向后台发送请求和接收响应。项目使用 Element UI 组件库进行开发，使得组件的复用得到大大提升，减少了前端开发的工作量。下面将对以上技术进行更详细的分析。

2.1.1 前端开发语言分析

(1) JavaScript 语言

JavaScript 是一种具有函数优先的轻量级、解释型或即时编译型的编程语言。它于 1995 年问世，在当时主要用途是代替 Perl 等服务器端语言处理输入验证，从此以后，JavaScript 逐渐成为市面上所有主流浏览器的标配。如今，JavaScript 的应用也不再局限于数据验证，而是渗透到 浏览器窗口及其内容的方方面面。完整的 JavaScript 包含三个部分，分别是核心（ECMAScript）、文档对象模型（DOM）、浏览器对象模型（BOM）。

(2) HTML 语言

HTML 的全称为超文本标记语言，是一种标记语言，它包括一系列的标签，通过这些标签可以将网络上的文档格式同意，使得分散的因特网资源连接为一个逻辑整体。HTML 文本是由 HTML 命令组成的描述性文本，HTML 命令可以说明文字、图形、动画、声音、表格、链接等。

(3) CSS 层叠样式表

CSS（Cascading Style Sheets）层叠样式表是一种用来表现 HTML 或 XML 等文件的计算机语言。它能够对网页中元素的位置的排版进行像素级精确控制，支持几乎所有的字体字号样式，可以静态地修饰网页，也可以配合各种脚本语言动态地对网页各元素进行格式化。使用 CSS 语言可以使系统有一个用户

使用友好的网页。

2.1.2 Vue.js 框架分析

Vue.js 是一套用于构建用户界面的渐进式框架，是一个 MVVM 的 js 库。它被设计为可以自底向上逐层应用。Vue.js 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

2.1.3 Element UI 组件库分析

Element 是一套基于 Vue.js 的 UI 组件库。目前最新推出的版本是 Vue3.0，我们在项目中使用的是基于 Vue2.0 的版本。Element UI 封装了大部分组件，使得组件得到极大程度的复用，它具有以下四个特点：

- 1) 一致性：与现实生活的流程、逻辑保持一致，所有元素的结构保持一致。
- 2) 反馈：通过界面样式和交互动效使页面元素的变化展现出来，让用户可以清晰地感知自己的操作。
- 3) 效率性：通过简洁直观的操作让用户快速理解作出决策。
- 4) 可控性：根据场景给予用户操作建议或安全提示，用户可以自由进行各种操作。

2.1.4 前后端分离架构分析

网络学院资产管理系统被设计为一个单页面应用 SPA (single-page application)，而单页面应用 SPA 正是前后端分离的一个典型应用。单页面的应用的特点在于在页面发生改变时不需要重新渲染整个页面，只需要局部地去修改页面中需要改变的部分，这样使得页面加载渲染的速度加快，用户的体验更好，也为项目将来的维护降低了维护成本。前后端的分离使得前端的页面开发工作和后端的 api 接口实现可以同步进行，大大减少了项目的开发时间成本，而且前端只需要负责 UI 的交互、页面的逻辑的实现，可以把更多的精力放在页面的优化上去，而后端也是根据需求去开发相应的接口，通过接口去读取或写入数据，保证用户的操作可以稳定地进行，其余的工作交给前端即可。前后端的交互则使用 axios 来发送前端请求和接收后台的响应。

2.1.5 MVVM 框架及开发模式优势

在 MVVM 框架出现以前有一个众所周知的设计模式：MVC，也就是模型层-视图层-控制层。这个设计模式的主要目的是分离视图层和模型层，是一种比较优秀的开发模式，能够令开发者感到满意，尽管如此，MVC 设计模式还是存在模型代码少、控制器代码多而且性能测试难度大等等问题。为了解决这些问题，MVVM 设计模式就出现了。所谓的 MVVM 指的使模型层-视图层-视图模型层，MVVM 设计模型也将模型层与视图层进行分离，视图层与视图模型层绑定，视图层向视图模型层请求命令，视图模型层与视图层进行通讯，这样可以比较轻松地实现页面的构建。视图层还能从模型层中取出数据，能够处理视图层中的业务逻辑。可以说，MVVM 设计模型具备了重用性高、低耦合、可独立开发、易于测试等优点，自诞生以来就受到了很多开发者的欢迎。

2.1.6 axios 技术

axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中。IE8 以上版本和所有浏览器的最新版本都支持 axios，它可以从浏览器中创建 XMLHttpRequests，从 node.js 创建 http 请求，支持 Promise API，拦截请求和响应，转换请求数据和响应数据，取消请求，自动转换 JSON 数据，客户端支持防御 XSRF。

2.2 经济可行性

网络教育学院资产管理平台是由网络教育学院教职工提出请求，学院老师进行指导，由我和另外一名学生进行开发，其中我负责前端的开发工作，另一名学生负责后台的开发工作，开发中使用到的技术，如前端的 JavaScript、html、css、Vue.js、Element 以及后台使用的 SpringBoot 框架均由我与另一名学生在网上或书籍中自学完成，因此在学习过程中不需要使用到资金。项目开发使用到的设备均是我们的个人电脑，不需要在网上购买服务器，所使用到的框架 Vue.js、SpringBoot 以及 UI 组件库 Element 均可免费使用，所以在设备方面可以不需要资金投入。因此在网络教育学院资产管理系统的开发过程中基本不需要任何资金的投入，具有非常大的经济可行性。

2.3 社会可行性

目前网络教育学院的资产管理流程使用的是纸质申请单教职工间传递签名

的方式进行，教职工需要整理好各自单位需要购买的资产，打印申请单交给部门领导审批，部门领导同意后交由主管领导进行审批，主管领导同意后由管理人员将申请的资产交由工作人员进行购买，若购买者并非申请者，则采购的进度需要向购买者反复确认。这样的流程显然十分繁琐，尤其是在部门领导或主管领导不同意时，还需要对申请订单进行修改打印，重新走一遍流程。这样就耗费了大量的时间和人力，使得资产的管理受阻，降低了资产购置的效率。因此，开发一个线上的资产管理系统十分必要，将原本申请单层层传递审批到购买进度的所有流程统统在系统中实现，可以、减少纸张的浪费，无论审批的部门领导还是主管领导是不是在办公室，他们也可以在线上对申请单进行审批，申请人也可以在系统中查看资产的购置进度，大大加快网络教育学院资产购置的速度。

2.4 开发可行性

我们两位都是用各自的电脑进行开发，系统开发基于 windows 10 操作系统，本人使用的处理器是 AMD Ryzen 5 处理器，可以在 chrome、Edge、IE 浏览器中进行兼容性测试。网络教育学院资产管理系统对运行所需要的设备和环境要求不高，因此，本人以及另一名同学可以使用各自的设备进行开发。

2.5 操作可行性

本系统的前端采用 Vue.js 和 Element UI 组件库进行开发，页面布局采用顶部导航栏，下面为正文的布局进行开发。Vue.js 具有响应式原理，能够将用户对虚拟 DOM 的操作已尽量快和少的步骤渲染到真实 DOM 上。让用户以最快的形式看到自己操作带来的变化。Element 是基于 Vue2.0 的 UI 组件库，减少前端工程师的开发难度的同时也为用户提供了一个简约易操作的界面。

3. 需求分析

3.1 功能需求

网络教育学院资产管理系统要求能够在线上进行从资产申请单的产生到领导审批以及最后采购状态的实现，主要使用到的前端框架为 Vue.js，组件库为 Element UI，后台框架为 SpringBoot，数据库为 Mysql。根据网络教育学院教职工的需求，我们将需求按角色分为 4 类账号，分别是普通用户账号、部门领导用户账号、主管领导用户账号以及管理员用户账号，各个账号功能划分如图 3-1。

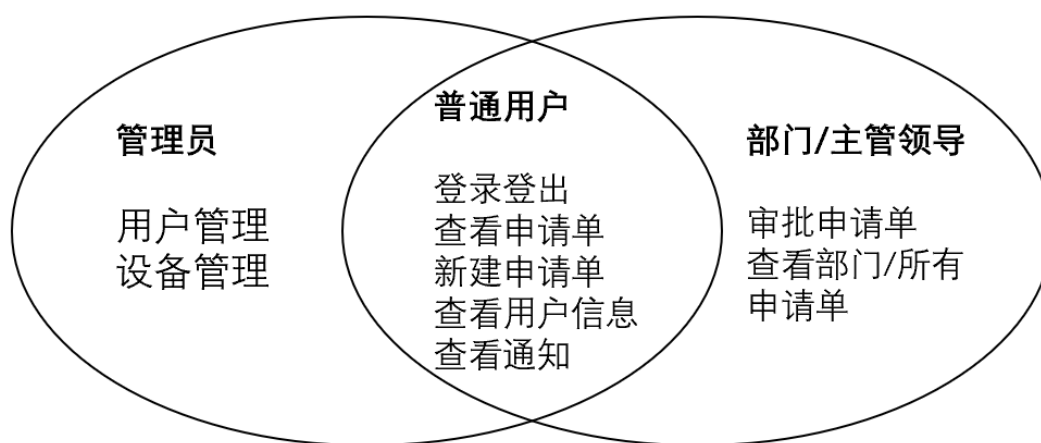


图 3-1 系统账号类别功能划分

由图可见，普通用户、部门领导、主管领导、管理员共有功能如下：

- 1) 登录登出：要求用户可以输入账号密码进入系统，在完成系统内所有任务时退出登录。
- 2) 新建申请单：要求用户可以在系统上输入申请单的部门、申请人、采购经费代码以及需要购买的资产的详细信息，最后生成一张申请单提交。
- 3) 查看个人申请单：要求用户可以看到自己曾经提交过的所有申请单。
- 4) 用户信息：要求用户可以在这里查看本账号的信息并修改密码。
- 5) 查看通知：要求用户可以在这里查看申请单进度的最新消息。

部门/主管领导的专属功能如下：

1) 审批申请单：要求部门/主管可以在这里通过或拒绝申请单。

2) 查看部门/所有申请单：要求部门领导可以在这里看到部门的所有申请单，主管领导则可以看到所有申请单。

管理员的专属功能如下：

1) 用户管理：要求管理员可以进行用户的单个/批量增加、修改。

2) 设备管理：要求管理员能够查看所有已通过主管领导审批的申请单中的设备，并选择设备生成采购单，任命某用户成为该采购单的负责人。

3.2 安全需求

为了保证网站不受到跨站伪请求（csrf）的攻击，项目在用户一开始登录时后台会生成一个 token 字符串以 Authorization 为键放在响应头中返回给用户，前端在收到这个请求的时候，会在 Vuex 中写入 token。Axios 拦截器会在每次发请求之前将 token 放入到请求头中。默认的 token 有效时间为 30 分钟，若超过 30 分钟，则 token 无效，会跳转到登录页面，若在 token 有效期的最后 10 分钟内发送请求，后台会返回一个带有新 token 的响应头，axios 拦截器检测到带有新的 token 响应头则用新的 token 代替旧的 token 进行后面的操作。

3.3 性能需求

根据网络教育学院教职工老师们的需求，所有的申请单都必须留底，所以无法对申请单进行删除，随着系统的使用时间越来越长，申请单累积的数量也会越来越多，因此，我们在查看申请单相关的页面使用分页请求或懒加载，使得一次性获得的数据有限，不必等到所有数据都返回的时候才能够进行查看等操作。

4. 总体设计

4.1 系统结构设计

网络教育学院资产管理系统前端采用 Vue.js 和 Element UI 组件进行开发，后台使用 SpringBoot 进行开发，数据库使用 Mysql，前端和后台的交互使用 axios 进行。前端负责页面的设计开发、发送页面的请求，数据组装和页面的渲染工作，后台负责匹配前端传来的路径并进行匹配，匹配成功则向数据库读写数据，并响应前端的请求。前端项目结构如下图 4-1 所示：

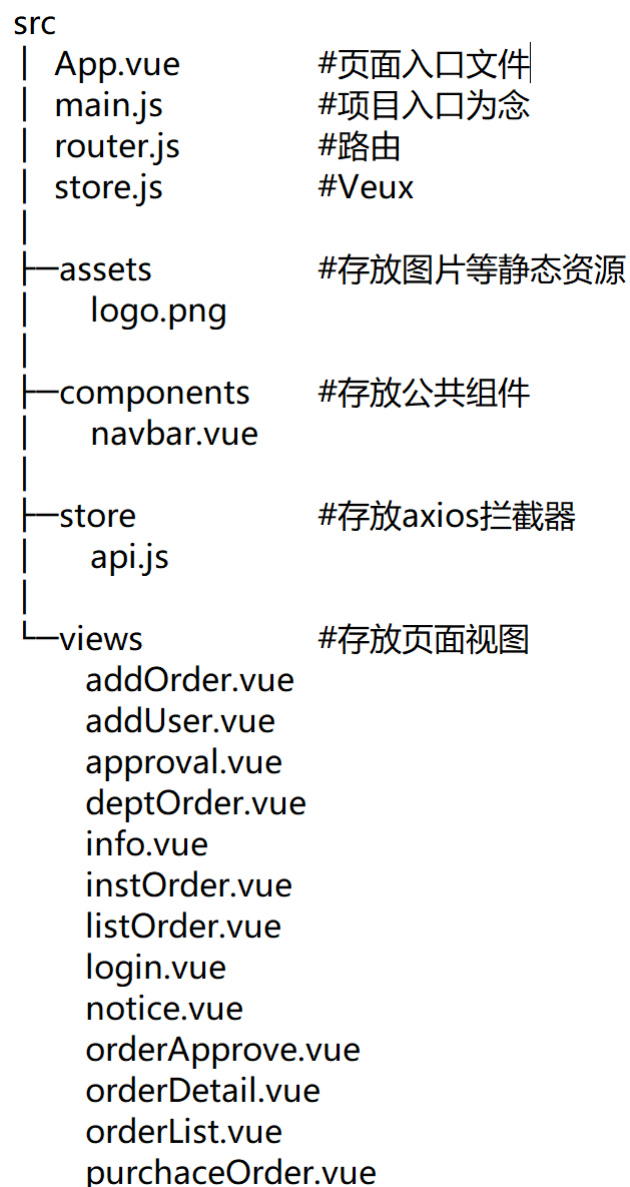


图 4-1 项目结构

4.2 系统功能设计

项目一共有 9 个功能模块：登录登出模块、新建申请单模块、查看个人申请单模块、用户信息模块、查看通知模块、审批申请单模块、查看部门/所有申请单模块、用户管理模块、设备管理模块。以下为功能模块的设计：

1) 登录登出模块：登录时要求用户输入正确的账号以及密码才能够进入系统，否则报错，用户登录成功的同时将后台返回的用户的相关信息如账号、名称、部门、角色、token 存入 vuex 并跳转到查看个人申请单页面；登出时将在发送请求后清空 vuex 中的数据并跳转到登录页面。

2) 新建申请单模块：在该模块可以输入部门、申请人、经费代码并且向申请单中添加物资条款，物资条款主要包括物资名称、配置参数、单价、数量、采购原因等，可以对添加的物资条款进行修改或删除，完成申请单的编辑后点击提交则将申请单状态改为 1（已提交），供主管领导审批。

3) 查看个人申请单模块：可以在该模块中看到用户本人提交的所有申请单，对于已提交部门领导未审批的申请单可以撤销提交后进行修改，对于领导已审批的申请单不能进行修改，只允许查看。可以通过申请单 id、申请部门、申请日期、经费代码、申请人对申请单进行搜索筛选。

4) 用户信息模块：用户可以在这里查看自己的账号信息，如账号、用户名，所属部门，也可以在这里修改密码，修改密码需要输入旧密码，并且两次输入新密码均成功时才会修改成功，修改密码成功后跳转至登录界面。

5) 查看通知模块：用户的申请单状态发生改变时系统会向用户发出一条消息，比如通过部门审核、通过主管领导审核等。

6) 审批申请单模块：待审批页面中列出所有该部门/主管领导的待审批申请单，点击待审批申请单进入申请单详情页面，在此页面中执行通过或拒绝操作，若拒绝申请单则需要输入拒绝原因。

7) 查看部门/所有申请单模块：该模块为部门/主管领导的专属模块，显示各自部门/主管的申请单，可以通过申请单 id、申请部门、申请日期、经费代码、申请人对申请单进行搜索筛选，也可以通过状态进行筛选。

8) 用户管理模块：管理员可以在这里添加单个用户、使用 excel 表格批量导入用户，批量导入的用户初始密码为 123456，也可以使用用户 id 对系统内所有用户进行搜索。

9)设备管理模块：将所有通过主管领导审批且还未进行采购的申请单内的物资条款以表格的形式罗列出来，管理员可以勾选相关的物资条目形成采购单交付给采购人员。

5. 详细设计

5.1 系统开发环境

硬件环境：处理器 AMD Ryzen 5 4600U with Radeon Graphics 2.10GHz，内存 16GB。

软件环境：操作系统 Windows10 64 位，编译器 WebStorm2020.1 x64，浏览器 Google Chrome。

5.2 通用组件的开发

在本项目中复用最多的组件是顶部导航栏 `navbar`，它在每一个页面中都需要用到。它位于项目目录中的 `components` 文件夹中，是基于 Element UI 组件库中的 `el-menu` 实现的，它获取 `vuex` 中的 `menu` 形成导航栏，封装有各个导航页面的跳转函数以及退出登录的方法，详细代码见附录。

5.3 数据库设计

通过对需求的整理分析，我们主要在数据库中设计了用户表、用户信息表、申请单表、申请单物资表，下面将对这三个表进行分析阐述。

(1) 用户表如表 5-1 所示：

表 5-1 用户表

字段名	类型	描述
id	char	用户账号
password	varchar	用户密码
is_active	tinyint	账号是否可用
role_id	int	角色 id
openid	varchar	OpenID

该表格主要用于在登录时实现账号与密码的匹配，从 `is_active` 判断该账号是否可用，再从 `role_id` 判断该返回哪些菜单项作为前端的导航栏。

(2) 用户信息表如表 5-2 所示：

表 5-2 用户信息表

字段名	类型	描述
id	char	用户信息表 id
username	varchar	用户名
account_id	char	外键，账号 id
department_id	char	外键，部门 id

该表记录了用户的基本信息，有 id、用户名、账号 id、部门 id 组成，在登录完成以及查看用户信息时返回用户的基本信息，在修改用户信时对该表进行写入。

(3) 申请表数据表如表 5-3 所示：

表 5-3 申请表数据表

字段名	类型	描述
id	char	申请单编号
apply_department	varchar	申请部门
apply_user	varchar	申请人
fund_code	char	采购经费代码
apply_date	datetime	申请日期
total	double	采购总金额
dept_leader_sign	tinyint	部门领导签名状态
dept_leader_sign_date	datetime	部门领导签名日期
inst_leader_sign	tinyint	主管领导签名状态
inst_leader_sign_date	datetime	主管领导签名日期
status	tinyint	申请单状态
file	mediumblob	签名文件
uid	char	申请人 id
withdrawal_reason	text	撤回原因

create_time	timestamp	创建时间
update_time	timestamp	更新时间
version	bigint	申请单版本

该表记录了一张申请单从产生到经过层层审批需要的各种信息，包括申请单的申请人、申请单位、总金额等基本信息，以及经过部门领导审批、主管领导审批后的各种状态的信息。

(4) 申请单物资信息如表 5-4 所示：

表 5-4 申请单物信息表

字段名	类型	描述
id	char	主键
name	varchar	物资名称
type	varchar	品牌型号
configuration	text	配置或技术参数
unit	char	单位
quantity	int	数量
budget_unit_price	double	预算单价
budget_total_price	double	预算总价
reason	text	申购原因及旧设备参数状态
new_user	varchar	新设备使用人
from_id	char	对应申请单编号
opinion	text	进入采购阶段后填写的设备列表各种情况或意见
status	tinyint	设备列表状态
purchase_id	int	对应采购单 id

该表记录了一张申请单中的物资条款情况。该表包括申请物资的名称、配置、型号等基本信息，用对应申请单编号来说明该条物资对应的是哪张申请单，以及

未来该物资被分配到某张采购单中需要经过的流程。

5.4 系统功能模块实现

1) 登录登出模块实现：登录界面如图 5-1，用户输入正确的账号密码，点击登录或按下键盘回车键就能够进入系统至查看申请单页面。



图 5-1 登录界面

2) 查看个人申请单模块实现：查看个人申请单模块界面如图 5-2 所示，该页面是登录成功之后的跳转页面，同时也可以通过顶部导航栏中的申请单->查看申请单进入。该页面主要由四部分组成，分别是顶部导航栏、搜索栏、个人申请单表格以及分页组成，可以通过在搜索栏相应字段输入相应的信息进行个人申请单的搜索，搜索为分页搜索，每一页有 10 条数据供用户查看，表格最底部显示当前页面中申请单总金额的总和，点击表格中的查看按钮可以对查看每条申请单的详细内容，如图 5-3 所示，可以对状态为已保存的申请单进行修改，对于非已保存状态的申请单不能进行修改，只能够进行打印。

用户

申请单

管理

设备列表

采购单

TestAdmin

申请单id:

申请部门: 请选择

申请日期: 选择日期 ~ 选择日期

申请经费代码:

申请人:

状态: 请选择

重置

搜索

编号	申请部门	申请人	申请日期	申请经费代码	总金额	状态	操作
20210201	财务部	a	2021年04月02日		24	已保存	查看
20210114	办公室	a	2021年03月25日	qq	6	主管领导已通过	查看
20210113	办公室	aaa	2021年03月05日	aa	6	主管领导已通过	查看
20210112	办公室	aaa	2021年03月05日	aa	6	主管领导已通过	查看
20210110	办公室	s	2021年02月04日	s	4	主管领导已通过	查看
总价					46		

<

1

>

图 5-2 查看个人申请单页面

用户

申请单

管理

设备列表

采购单

TestAdmin

编号: 20210201

采购经费代码:

申请日期: 2021年04月02日

申请部门: 财务部

申请人: a

采购总金额: 24

状态: 已保存

序号	物资名称	品牌型号	配置或技术参数	单位	数量	预算单价	预算总价	申请原因(请详细列明申请理由,并阐述采购必要性,要求不少于100字,如有旧设备,请列明)	新设备使用人	操作
2021020101	a	aaa	aaaaaaaaaaaa	g	4	6	24	aaaa	办公室用户1	<div>编辑</div> <div>删除</div>

添加物资条款

打印

提交

图 5-3 查看申请单详情页面

3) 新建申请单模块实现: 该页面位于顶部导航栏申请单->新建申请中, 界面如下图 5-4 所示。页面主要由顶部导航栏、横向表单、申请单条款表格以及底部的按钮组成。进入该页面时, 系统会自动将申请人的部门和申请人填写好, 申请部门使用选择器进行输入, 申请人使用模糊搜索返回相关用户名称方便用户的输入; 点击申请单条款中的编辑可对该条款进行编辑, 点击删除可进行删除; 点击底部的添加条款会弹出对话框以增加申请单中的条款, 如图 5-5a、图 5-5b 所示, 每张申请单最多只能有 6 个条目, 若已有 6 个条目点击添加条款则会弹出相关的消息提醒; 点击提交前系统会判断申请单以及相关条款是否填写完整, 若字段未填写完整则会将相关字段的输入框改为红色以提醒用户, 所有字段均填写完整才能够提交。

用户

申请单

管理

设备列表

采购单

TestAdmin

申请部门

办公室

申请人

TestAdmin

采购经费代码

采购经费代码

采购总金额: 4

物资名称	品牌型号	配置或技术参数	单位	数量	预算单价	预算总价	申请原因(请详细列明申请理由,并阐述采购必要性,要求不少于100字,如有旧设备,请列明)	新设备使用人	操作
a	a	a	个	2	2	4	a	TestAdmin	<div>编辑</div> <div>删除</div>

添加物资条款

提交

图 5-4 新建申请单

添加资产

物资名称

品牌型号

配置或技术参数

单位

请选择

数量

0

预算单价(元)

0

图 5-5a 添加物资条款

预算总价 0

原因(请详细列明申购理由, 并阐述采购必要性, 要求不少于100字, 如有旧设备, 请列明)

新设备使用人 请输入申请人

取消 确定

图 5-5b 添加物资条款

4) 用户信息模块：用户信息界面位于顶部导航栏用户->用户信息下，该界面显示用户的基本信息以及修改密码功能，修改密码点击确认修改后 token 会过期页面会跳转至登录界面。

用户 申请单 管理 设备列表 采购单 TestAdmin

用户信息

帐号: 12345678910

用户名: TestAdmin

所属部门: 办公室

是否采购负责人: 否

旧密码

新密码

确认新密码

确认修改

图 5-6 用户信息界面

5) 查看通知模块实现：查看通知模块位于用户名称->通知下，用户的申请单状态发生变化时系统会给用户发送一条消息，用户可以在此处查看。



图 5-7 查看通知模块

6) 审批申请单模块实现：该功能属于主管或部门领导的功能，位于主管/部门->待审批页面中，页面由待审批申请单表格组成，点击查看按钮可以跳转至审批界面，在审批界面进行通过或拒绝，若拒绝需要填写拒绝理由。

编号	申请部门	申请人	申请日期	采购经费代码	总金额	操作
20200106	办公室	申请人1	2020年12月11日	123456	0	查看
20200104	办公室	申请人1	2020年12月11日	123456	0	查看

图 5-8 待审批界面

序号	物资名称	品牌型号	配置或技术参数	单位	数量	预算单价(元)	预算总价(元)	申请原因(请详细列明申购理由,并阐述采购必要性,要求不少于100字,如有旧设备,请列明)	新设备使用人
1	a	a	a	套	3	3	9	a	TestAdmin

[通过](#)
[拒绝](#)

图 5-9 审批申请单界面

7) 查看部门/所有申请单模块实现：该功能为主管/部门领导拥有的功能，界面如图 5-10 所示，主要由上方搜索栏和下方申请单表格构成，点击查看可进入申请单详情页面查看申请单具体内容。

编号	申请部门	申请人	申请日期	申请经费代码	总金额	状态	操作
20210115	办公室	TestAdmin	2021年04月11日	123456	9	已提交	查看
20210114	办公室	a	2021年03月25日	qq	6	主管领导已通过	查看
20210113	办公室	aaa	2021年03月05日	aa	6	主管领导已通过	查看
20210112	办公室	aaa	2021年03月05日	aa	6	主管领导已通过	查看
20210111	办公室	w	2021年02月08日	w	4	已保存	查看
20210110	办公室	s	2021年02月04日	s	4	主管领导已通过	查看
20210109	办公室	a	2021年01月31日	a2	8	已保存	查看
20210108	办公室	a	2021年01月30日	a1	3	主管领导已通过	查看

图 5-10 查看部门申请单

8) 用户管理模块：用户管理模块位于管理员账号下的管理->用户管理下，其界面如图 5-11 所示，界面主要由上方搜索按钮和下方用户列表组成，管理员可以通过在输入框输入用户 id 点击搜索进行搜索；点击添加用户或批量添加用户进行添加；还可以对表格内的用户进行编辑，编辑内容如图 5-12 所示。

帐号	用户名	部门	职位	是否采购负责人	操作
12345678919	导入2	招生部	普通用户	否	编辑
12345678918	导入1	办公室	普通用户	否	编辑
12345678917	主管	办公室	主管院领导	否	编辑
12345678916	Boss	指导中心	主管院领导	否	编辑
12345678915	addOne1	教学部	主管院领导	否	编辑
12345678914	办公室领导	办公室	部门领导	否	编辑
12345678913	财务部用户1	财务部	普通用户	否	编辑
12345678912	办公室用户2	办公室	普通用户	否	编辑

图 5-11 用户管理界面

编辑用户

×

帐号

12345678919

用户名

导入2

密码

部门

招生部

▼

职位

普通用户

▼

取消

任命为采购负责人

确定

图 5-12 编辑用户

9) 设备管理模块实现：设备管理界面如下图 5-13 所示，页面由已通过主管领导审批的申请单中的条款表格以及底部的“形成采购单”按钮组成，勾选表格条款后点击形成采购单任命给系统中的采购人员相应的采购任务。

用户 ▼ 申请单 ▼ 管理 ▼ 设备列表 ▼ 采购单 ▼

TestAdmin ▼

<input type="checkbox"/>	序号	物资名称	品牌型号	配置或技术参数	单位	数量	预算单价	预算总价	申购原因	新设备使用人	申请部门
<input type="checkbox"/>	2021011401	qq	q	q	个	3	2	6	qq	qq	办公室
<input type="checkbox"/>	2021011201	a	a	a	a	2	3	6	aaa	aa	办公室
<input type="checkbox"/>	2021011000	s	ss	s	s	2	2	4	ss	ss	办公室
<input type="checkbox"/>	2021010800	a	a	a	a	1	3	3	asdf	asdf	办公室
<input type="checkbox"/>	2021010100	w	w	w	w	1	10000000	10000000	w	w	办公室
<input type="checkbox"/>	2020011203	w	w	w	w	2	2	4	w	w	办公室
<input type="checkbox"/>	2020011202	b	b	b	b	2	3	6	b	b	办公室
<input type="checkbox"/>	20200112	a	a	a	a	1	2	2	1	a	办公室
				视频输入接口支持3种3G SDI、H					必要性及用途：录像盒子主要是录制周末面授辅导课，每个学期300		

形成采购单

图 5-13 设备管理界面

6. 测试

对于前端开发来说，页面功能是否能够正常使用、系统是否安全至关重要，所以在测试方面，我就功能测试、安全测试两方面进行。

6.1 功能测试

(1) 登录登出功能：主要对登录表单以及登出按钮进行测试，测试结果如下：

表 6-1 登录登出功能测试

测试功能	测试操作	测试结果	测试时间
登录表单校验	多次输入正确、错误或空的表单	功能正常	2021/4/10
登出后退出系统到登录界面	点击登出按钮	功能正常	2021/4/10

(2) 申请单相关功能：对新建申请单和查看申请单相关功能进行测试，测试结果如下：

表 6-2 申请单功能测试

测试功能	测试操作	测试结果	测试时间
添加物资条款功能	多次只输入部分字段以及输入所有字段	功能正常	2021/4/10
编辑物资条款功能	点击编辑物资条款按钮，多次对物资条款进行修改	功能正常	2021/4/10
删除条款功能	点击“删除”按钮	功能正常	2021/4/10
新建申请单提交功能	在字段不全或字段完全的情况下点击提交按钮	功能正常	2021/4/10
查看申请单分页表	点击查看申请菜单	功能正常	2021/4/10
查看申请单细则	点击申请单表格中的查看	功能正常	2021/4/10

打印申请单	点击申请单详情总的打印按钮	功能正常	2021/4/10
上传申请单图片	点击申请单详情中的上传按钮	功能正常	2021/4/10
下载申请单图片	点击申请单详情中的下载申请单图片按钮	功能正常	2021/4/10

(3) 用户相关功能模块：主要对用户信息、管理员账号中的用户管理功能进行测试，测试结果如下：

表 6-3 用户管理功能测试

测试功能	测试操作	测试结果	测试时间
用户信息显示	点击用户信息	功能正常	2021/4/10
用户信息内修改密码	多次输入正确或错误的新旧密码，点击确认修改	功能正常	2021/4/10
用户管理中的搜索功能	多次输入正确、错误或空的id，点击搜索	功能正常	2021/4/10
用户管理中的添加用户	多次输入不完整的字段以及完整的字段，点击确认按钮	功能正常	2021/4/10
用户管理总的批量添加用户	点击批量添加按钮，下载模板，点击打开文件	功能正常	2021/4/10

(4) 审批相关功能模块：主要对部门/主管领导对申请单的拒绝或通过进行测试，测试结果如下：

表 6-4 审批功能测试

测试功能	测试操作	测试结果	测试时间
待审批列表的显示	点击部门/主管->待审批菜单	功能正常	2021/4/10

部门/主管通过申请单，跳回待审批页面	点击查看申请单详情，点击通过按钮	功能正常	2021/4/10
部门/主管拒绝申请单，跳回待审批页面	点击拒绝按钮，输入拒绝理由，点击确认	功能正常	2021/4/10

(5) 设备列表功能模块：主要对设备列表进行测试，测试结果如下：

表 6-5 设备列表功能测试

测试功能	测试操作	测试结果	测试时间
设备列表的显示	点击设备列表->所有设备列表菜单	功能正常	2021/4/10
勾选设备条目，形成采购单	点击表格左侧的多选框，点击形成采购单	功能正常	2021/4/10

6.2 安全测试

安全测试主要通过登录各个不同角色的账号，如管理员、普通用户、部门领导、主管领导的账号进行权限的测试。后台将 token 的时间设置为 30 分钟，若在 token 过期前 10 分钟用户发送了请求，则会在响应中放入新的 token，前端也将用新的 token 代替旧的 token。

表 6-6 安全测试

测试内容	测试操作	测试结果	测试时间
登录不同用户查看权限	分别使用管理员、普通用户、部门领导、主管领导的账号进行登录	功能正常	2021/4/10
在 token 过期后不能进行操作	登录 30 分钟后点击页面中任一可发送请求的按钮	功能正常	2021/4/10
新 token 代替旧 token	在登录后 20 分钟发送请求，查看 token 是否更换	功能正常	2021/4/10

7. 总结与展望

数字化办公能够对日常的办公数据进行管理，开发资产管理系统可以帮助网络教育学院的教职工更高效地进行资产的购置与管理，节约大量时间以及人力成本。本文根据网络教育学院教职工的实际需求，实现了基于 MVVM 的开发模式、使用 Vue.js 搭配 Element UI 组件库的资产管理平台。在本文依次分析了网络教育学院教职工目前的资产管理现状以及对未来资产管理数字化的要求，对资产管理系统进行可行性分析与需求分析，对资产管理系统进行总体设计和具体实现，并对已实现功能进行测试。目前资产管理系统前端部分的主要工作内容如下：

(1) 通过对网络教育学院教职工的访谈，对网络教育学院教职工的需求进行整理分析。通过访谈，我们确定了使用该系统的目标人群、平台预实现功能以及用户角色分类下的各种功能，根据功能设计资产管理系统网页的页面设计。

(2) 分析目前的前端主流技术，决定使用 Vue.js 进行前端的开发。使用前后端分离的方式进行开发，可以使得前后端的工作进度同时进行，大大提高开发效率，同时，将前端和后台进行分离可以使得错误定位更准确，整个系统易于维护。使用主流的 MVVM 框架 Vue.js 以及基于 Vue.js 的 UI 组件库 Element UI 进行开发，可以大大减少前端的工作量，提升开发速度。

(3) 根据网络教育学院教职工的需求，设计通用的功能模块，如登录模块、申请单模块、用户管理模块等等。

(4) 根据系统的需求，搭建前后端分离的开发环境，利用 Vue.js 和 Element UI 组件库具体实现系统的每一个模块页面，并开发通用的组件，如顶部导航栏，进行了健壮性测试。

(5) 对网络教育学院资产管理系统进行了模块化的黑盒测试，针对系统的每个模块进行了功能测试与安全测试，系统通过测试进行查漏补缺。最终测试结果满足了系统设计需求。

(6) 将系统向网络教育学院教职工进行展示，尝试将系统在真实场景中实践。结果表明，该系统能够满足网络教育学院教职工日常资产购置的需求，其实用性与可行性得到了有效的验证。如后续将系统放置到真实场景中使用，还需要

新增采购模块与学校官方系统进行对接。

基于 Vue.js 的网络教育学院资产管理系统较好地完成了以上六项工作，但是，系统还存在一些可提升的地方，因此，在此对该系统做出以下基点展望：

（1）由于工作量较大，该系统中有一些细节的部分做得还不是很完善，如网络教育学院的教职工老师曾希望申请单原因的括号部分字体呈红色以提醒用户谨慎输入该部分内容，部分页面的内容复用次数多，但并未设计通用组件进行页面的复用等，这些内容还需后续做出优化。

（2）对用户需求的搜集不够完善。目前系统完成了资产从提出到各级领导完成审批的各个流程所需要的功能，但是对网络教育学院教职工后续提出的采购功能没有实现，这一部分的功能需要对接学校的采购系统，还望后续能够补足这一部分的功能，让系统投入到实际的使用当中。

（3）页面设计兼容性不足。在目前的系统设计中，仅部分页面使用了 rem、vh、vw 等单位进行页面开发，页面在不同尺寸的设备上显示的适配度不高，这一部分还望后续能够进行优化。

（4）跟进当前主流的技术趋势，将最新的前端设计思想应用到该系统中，让系统的用户得到更舒适的用户体验，这样系统才能够在以后的更新迭代中继续工作，形成一个良好迭代生态循环。

8. 参考文献

- [1]王志任. 基于 Vue.js 的开发平台的设计与实现[D]. 广东工业大学, 2018.
- [2]旷志光, 纪婷婷, 吴小丽. 基于 Vue.js 的后台单页应用管理系统的研究与实现[J]. 现代计算机(专业版), 2017(30):51-55.
- [3]朱二华. 基于 Vue.js 的 Web 前端应用研究[J]. 科技与创新, 2017(20):119-121.
- [4]李广宏. vue.js 前端应用技术分析[J]. 中国新通信, 2019, 21(20):115.

附录

navBar 组件实现代码:

```
<template>
  <div >
    <el-menu
      style="position: fixed;width: 100%;height: 60px;z-index: 9"
      :unique-opened="true"
      @select="handleSelect"
      :default-active="activeIndex"
      class="el-menu-demo"
      mode="horizontal">
      <el-submenu v-for="(item, index) in menu"
        :key="index" :index="item.name">
        <template slot="title">
          <span slot="title">{{ item.name }}</span>
        </template>
        <el-menu-item v-for="(child, index) in item.children" :key =
"index" :index="child.path">{{ child.name }}</el-menu-item>
      </el-submenu>
      <el-dropdown trigger="click" class="account"
        @command="handleCommand">
        <el-badge :is-dot="showDot" class="UserMenu">
          <span class="el-dropdown-link" style="cursor: pointer">
            {{ loginName }}
            <i class="el-icon-arrow-down"></i>
          </span>
        </el-badge>
        <el-dropdown-menu slot="dropdown">
```

```

        <el-badge :value="noticeCount" class="item" :hidden="!showDot">
            <el-dropdown-item icon="el-icon-plus" command="notices">通知
        </el-dropdown-item>

        </el-badge>

        <el-dropdown-item icon="el-icon-switch-button" command="exit">注
销</el-dropdown-item>

    </el-dropdown-menu>

</el-dropdown>

</el-menu>

</div>

</template>

<script>
export default {
  name: "navBar",
  data(){
    return{
      activeIndex: '1',
      menu: this.$store.state.menu,
      username:"",
      showDot: false,
      noticeCount: 0,
      loginName: this.$store.state.loginName
    }
  },
  created() {
    this.username = this.$store.state.userName
    this.$getAxios(true).get("/notice/count",{

```

```

      params: {
        uid: this.$store.state.userName
      }
    }).then((res)=>{
      this.noticeCount = res.data.data
      if (res.data.data){
        this.showDot = true
      }
    })
  },
  methods: {
    handleSelect(key) {
      this.$router.push({path:key})
      console.log('this is path',key)
    },
    handleCommand(command){
      if(command==='exit'){
        this.exit()
      }else if (command === 'notices'){
        this.goToNotices()
      }
    },
    goToNotices(){
      this.$router.push({path:'/notices'})
    },
    exit(){
      this.$getAxios(true).get('/user/logout',{
        params: {
          uid: this.$store.state.userName

```

```
    }
  }).then(res=>{
    if(res.data.success){
      sessionStorage.clear()
      this.$store.commit('tokenClear')
      this.$store.commit('usernameClear')
      this.$store.commit('loginNameClear')
      this.$store.commit('menuClear')
      this.$router.push({path:'/login'})
    }
  }).catch(err=>{console.log(err)})
}
}
}
</script>
```

```
<style scoped>
```

```
.account{
  position: fixed;
  right: 10px;
  top: 20px;
}
```

```
</style>
```

致谢

本人毕业设计能够顺利完成，首先要感谢网络教育学院的老师耐心地与我们沟通需求，提供场地供我与小组成员进行沟通和开发。他们每一次对系统提出的问题都使我们开发的网络教育学院资产管理系统向更好的地方迈出了坚实的步伐。特别要感谢我们的毕业设计指导老师单志龙老师，他每一次开会中对我们的教诲，都使我醍醐灌顶，对我起到了极大的帮助。

其次，要感谢在资产管理系统中负责后台开发的李仲贤同学。我们从互相不认识到因在同一个小组中做毕业设计结缘，该系统用户角色众多，他每一次都能在开会之后及时对会议内容做出反馈和总结，以及提出下一步应该完成的内容，能够与他合作完成毕业设计实在是本人的荣幸。

最后，要感谢一直在背后支持我的家人和朋友们，是他们一如既往的爱让我能够坚持到现在，一直努力，一直向上，在遇到困难时也不轻言放弃，最终完成了毕业设计。