|  |
| --- |
| **成绩** |
|  |



中国农业大学

课程论文

（2022-2023 学年春季学期）

**论文题目：** 档案归档详细设计文档

**课程名称：** 软件工程

**任课教师：** 李林

**小 组：** 档案管理小组

**班　　级：** 计算机201

**学　　号：** 2020308250101

**姓　　名：** 张维轩

目录

[档案归档系统详细设计文档 3](#_Toc139415935)

[一、用户组别与页面关系表 3](#_Toc139415936)

[二、档案归档页面设计 4](#_Toc139415937)

[2.1 登录页面Login 4](#_Toc139415938)

[2.2 档案管理页面Main 5](#_Toc139415939)

[2.3 档案归档页面 7](#_Toc139415940)

[三、档案归档Controller设计 8](#_Toc139415941)

[3.1 Controller设计要求 8](#_Toc139415942)

[3.2 档案索引规则Controller 8](#_Toc139415943)

[3.3 档案分类Controller 9](#_Toc139415944)

[3.4 档案架Controller 9](#_Toc139415945)

[四、档案归档SpringMVC配置表 10](#_Toc139415946)

[4.1 档案归档SpringMVC配置表 10](#_Toc139415947)

[4.2 档案索引规则设置 10](#_Toc139415948)

[4.3 档案分类设置 10](#_Toc139415949)

[4.4 档案架设置 11](#_Toc139415950)

[五、数据类型转换表设计 11](#_Toc139415951)

[5.1 数据类型转换表 11](#_Toc139415952)

[5.2 档案索引规则DTO 11](#_Toc139415953)

[5.3 档案登记表DTO 11](#_Toc139415954)

[5.4 档案分类表DTO 12](#_Toc139415955)

[5.5 档案架信息表DTO 12](#_Toc139415956)

[六、数据校验表设计 12](#_Toc139415957)

[七、国际化设计 13](#_Toc139415958)

[7.1 多语言资源文件 13](#_Toc139415959)

[7.2 日期和时间格式本地化 14](#_Toc139415960)

[7.3 动态切换语言 14](#_Toc139415961)

[八、 系统验证说明表 14](#_Toc139415962)

[九、 操作日志拦截器设计 15](#_Toc139415963)

[十、 问题件操作日志设计 15](#_Toc139415964)

[十一、SpringMVC标签库设计 16](#_Toc139415965)

[十二、Mybatis参数表设计 17](#_Toc139415966)

[十三、JDBC数据源用户角色表设计 17](#_Toc139415967)

[十四、档案归档会话Mapper设计表 18](#_Toc139415968)

[十五、domain与数据表或视图的对应关系 18](#_Toc139415969)

[十六、mybatis用户权限表 19](#_Toc139415970)

[十七、每一用例界面元素设计表 19](#_Toc139415971)

[十八、界面设计，界面元素说明 20](#_Toc139415972)

[十九、状态改变表 21](#_Toc139415973)

[二十、数据元素结构设计 21](#_Toc139415974)

[20.1 档案索引规则元素结构 21](#_Toc139415975)

[20.2 档案信息元素结构 21](#_Toc139415976)

[20.3 档案架信息元素结构 21](#_Toc139415977)

[20.4 选择档案架元素结构 22](#_Toc139415978)

[二十一、系统运行设计 22](#_Toc139415979)

[21.1 系统架构 22](#_Toc139415980)

[21.2 技术栈 22](#_Toc139415981)

[21.3 开发流程 22](#_Toc139415982)

[21.4 数据交互流程 23](#_Toc139415983)

[二十二、主要算法设计 23](#_Toc139415984)

[22.1 校验算法 23](#_Toc139415985)

[22.2 输入检测算法 23](#_Toc139415986)

[22.3 档案编码确认算法 23](#_Toc139415987)

档案归档系统详细设计文档

张维轩

2020308250101

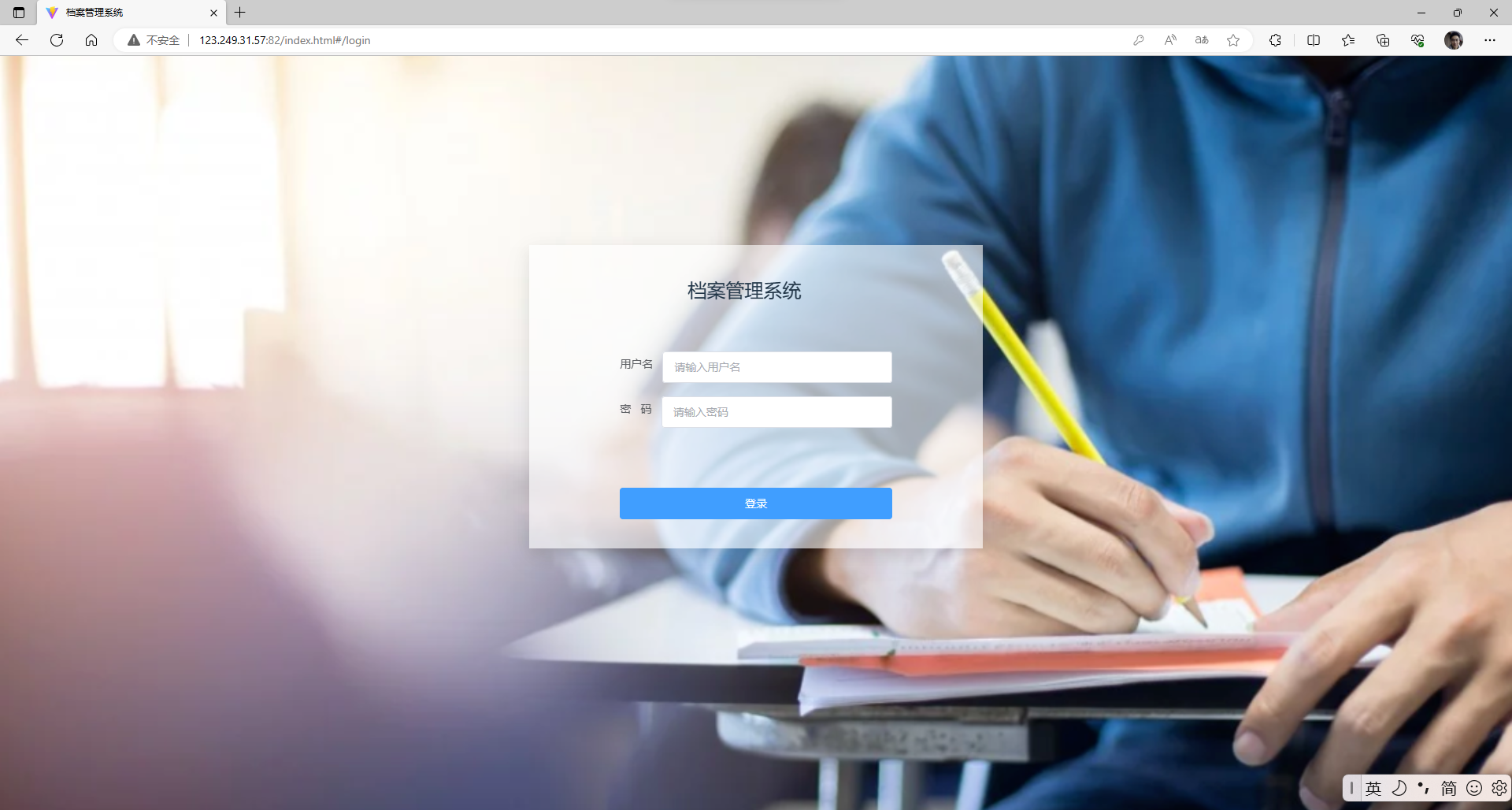
# 一、用户组别与页面关系表

在本次项目中，通过与小组成员交流，我们的档案管理系统最终采用了动态的用户权限管理方式，其中管理员（admin）可以对其他用户进行权限配置，而用户的访问权限是根据管理员的配置而动态确定的。由于权限与用户组别之间没有固定的对应关系，因此无法建立用户组别与页面的静态关系表，唯一固定的用户组别是管理员admin，其对应的页面为用户管理页面。

通过动态配置权限，管理员可以根据实际需求对不同用户进行个性化的权限设置，使系统能够灵活适应不同用户的需求。当系统中新增用户或修改用户权限时，管理员可以直接进行相应的权限配置，无需修改用户组别与页面的关系表或重新定义用户组别。这样可以提高系统的可扩展性和维护性。动态权限管理可以有效地保护系统的安全性。管理员可以细粒度地控制用户对系统中各个页面和功能的访问权限，从而减少潜在的安全风险。

# 二、档案归档页面设计

## 2.1 登录页面Login



1. 页面文件分析：

文件位置：archive\_management\_front/src/components/home/Login.vue

页面设计：

模板部分 (<template>)：包含了登录页面的 HTML 结构和元素。通过使用 Element Plus UI 组件库的 <el-form>、<el-form-item> 和 <el-input> 组件实现用户名和密码的输入框。

使用 <el-button> 组件实现登录按钮。

使用 <span> 元素展示页面标题。

应用内联样式 (style 属性) 来设置页面布局和样式。

脚本部分 (<script>)：包含了页面的逻辑和行为。

（2）页面所用技术点设计：

VUE 框架：该页面使用 Vue 框架来构建前端应用程序。

Element Plus UI 库：使用 Element Plus 提供的 UI 组件来构建表单和按钮等交互元素。

Axios：通过使用 $http 对象进行 HTTP 请求，从后端获取用户登录验证结果。

CSS 样式：使用内联样式和定位属性来实现页面元素的布局和样式。

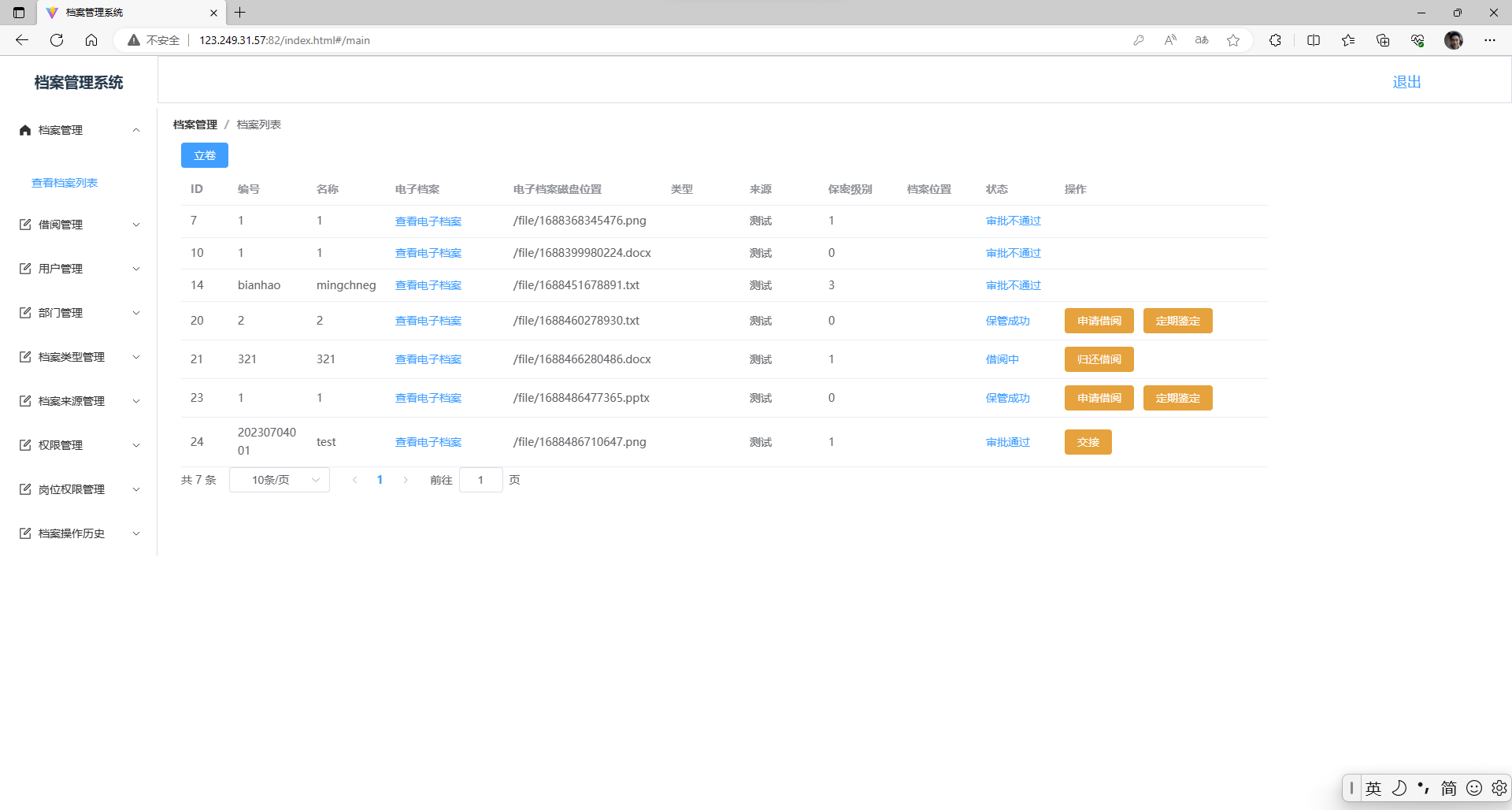
（3）业务校验

在登录表单提交时，使用 Axios 发送 HTTP GET 请求到 /user/login 接口，携带用户名和密码作为参数。后端返回一个状态码和相应的数据。

如果状态码为 1，表示登录成功，将返回的 token 存储到 sessionStorage 中，并通过 $router.push('/main') 跳转到主页面。

如果状态码不为 1，表示用户名或密码不正确，会使用 Element Plus 的 ElMessage 组件弹出错误提示消息。

## 2.2 档案管理页面Main



1. 页面设计

文件位置：archive\_management\_front/src/components/home/Main.vue

页面设计：

页面结构：使用 el-container 组件创建一个容器。

容器布局：分为两部分，左侧是导航栏 (el-aside)，右侧是内容区域 (el-container 和 el-main)。

导航栏 (el-aside)：包含了一个标题和多个菜单项。

使用 el-menu 组件作为菜单容器，通过设置 default-active 属性指定默认选中项。

使用 el-sub-menu 组件作为一级菜单项的容器，通过设置 index 属性指定菜单项的索引值。

使用 el-menu-item-group 组件作为菜单项的容器，通过设置 index 属性指定菜单项的索引值。

使用 el-menu-item 组件作为具体的菜单项，通过设置 index 属性指定菜单项的索引值，使用 @click 事件监听菜单项的点击事件。

内容区域 (el-container 和 el-main)：根据菜单项的选择展示对应的组件。

根据 menu 变量的值动态显示对应的组件，例如 dangan、jy、user 等。

1. 页面所用技术点设计：

Vue.js：Main.vue 是一个 Vue 单文件组件，使用 Vue.js 框架进行开发。

Element Plus：该页面使用 Element Plus UI 框架进行页面布局和组件渲染。包括使用 el-container、el-aside、el-menu、el-sub-menu、el-menu-item-group、el-menu-item 等组件进行布局和导航菜单的展示。

Vue Router：页面通过 this.$router.push() 方法实现路由跳转，具体代码中的示例为 this.$router.push('/main')，用于跳转到指定的路由路径。

动态组件渲染：根据 menu 变量的值，动态展示不同的组件，例如 dangan、jy、user 等。通过条件渲染的方式在 <el-main> 区域展示对应的组件。

Element Plus 图标组件：页面中使用了 Element Plus 提供的图标组件，包括 HomeFilled 和 Edit，通过 el-icon 和图标组件实现菜单项的图标展示。

档案类型来源分页



## 2.3 档案归档页面

(1) 页面设计

页面设计是一个简单的档案归档页面。页面的主要内容是一个表单，用户可以输入档案的名称、归档日期和归档位置，并通过点击提交按钮将表单信息提交给后端进行处理。

（2）页面应用技术

Vue.js：页面采用 Vue.js 框架进行开发，利用 Vue 的数据绑定和组件化能力构建页面。

Element Plus：使用 Element Plus UI 组件库，通过 el-form、el-form-item、el-input、el-date-picker、el-button 等组件实现表单的布局和输入控件。

数据绑定：通过 Vue 的双向数据绑定，将表单输入的值与 archive 对象中的属性进行关联，实现数据的实时更新和同步。

表单提交：通过点击提交按钮，触发 submitForm 方法，进行表单的提交操作。可以在该方法中发送请求将归档信息提交到后端进行处理。

（3）业务校验

根据后端返回的结果，进行业务校验和处理。可以使用条件判断、错误提示等方式来处理校验不通过的情况，保证提交的归档信息的合法性。

# 三、档案归档Controller设计

## 3.1 Controller设计要求

当分析和设计档案归档控制器时，我们需要考虑以下几个方面：

（1）路由和请求方法：根据业务需求，确定需要支持的路由和相应的请求方法（POST、GET、PUT、DELETE）。

（2）身份验证和授权：确定是否需要进行身份验证和授权，以确保只有经过身份验证的用户才能执行敏感操作。在档案归档Controller中，使用了JWT（JSON Web Token）进行身份验证，但你也可以根据你的需求选择其他身份验证机制。

（3）参数传递和数据验证：确定需要从请求中获取的参数，并进行相应的数据验证。在档案归档Controller中，使用了@RequestBody和@PathVariable注解来获取请求体和路径参数，并对数据进行验证。

（4）业务逻辑调用：调用相应的业务逻辑方法来执行档案归档的操作，例如创建档案、获取档案、更新档案和删除档案等。在档案归档Controller中，通过注入ArchiveService和UserService，并调用相应的方法来处理档案相关的操作。

（5）错误处理和返回结果：处理可能发生的错误情况，并返回适当的错误信息和HTTP状态码。在档案归档Controller中，通过返回Message对象来封装结果信息，包括消息、状态码和数据。

档案归档的三个步骤（档案索引规则的设置、档案分类的设置和档案架的设置）中的每个步骤都对应一个Controller时，我们可以对每个Controller的方法进行详细分析：

## 3.2 档案索引规则Controller

（1）editIndexRule(Model model)方法：

作用：处理GET请求，用于展示编辑档案索引规则的页面。

参数：model对象用于传递数据给视图。

返回值：返回字符串 "editIndexRule"，表示视图的名称。

（2） saveIndexRule(ArchiveIndexRule rule)方法：

作用：处理POST请求，用于保存或更新档案索引规则。

参数：rule对象包含用户在前端页面上输入或选择的档案索引规则数据。

返回值：返回字符串 "redirect:/archive/indexRule/edit"，表示重定向到编辑档案索引规则的页面。

## 3.3 档案分类Controller

（1） editClassification(Model model)方法：

作用：处理GET请求，用于展示编辑档案分类设置的页面。

参数：model对象用于传递数据给视图。

返回值：返回字符串 "editClassification"，表示视图的名称。

（2）saveClassification(ArchiveClassification classification)方法：

作用：处理POST请求，用于保存或更新档案分类设置。

参数：classification对象包含用户在前端页面上输入或选择的档案分类设置数据。

返回值：返回字符串 "redirect:/archive/classification/edit"，表示重定向到编辑档案分类设置的页面。

## 3.4 档案架Controller

（1）editShelf(Model model)方法：

作用：处理GET请求，用于展示编辑档案架设置的页面。

参数：model对象用于传递数据给视图。

返回值：返回字符串 "editShelf"，表示视图的名称。

（2）saveShelf(ArchiveShelf shelf)方法：

作用：处理POST请求，用于保存或更新档案架设置。

参数：shelf对象包含用户在前端页面上输入或选择的档案架设置数据。

返回值：返回字符串 "redirect:/archive/shelf/edit"，表示重定向到编辑档案架设置的页面。

# 四、档案归档SpringMVC配置表

## 4.1 档案归档SpringMVC配置表



## 4.2 档案索引规则设置

Controller：IndexRuleController是负责处理档案索引规则设置的控制器组件，它接收前端请求并将其转发给后续处理程序。

View：IndexRule.jsp是用于展示档案索引规则设置页面的视图，负责接收必要的请求参数和展示页面内容。

Model：IndexRuleService是处理档案索引规则设置业务逻辑的服务组件，它负责校验输入数据、更新数据库等操作。

URL Pattern：档案索引规则设置的URL Pattern被设置为/archive/indexRule，用于匹配前端发起的档案索引规则设置请求。

## 4.3 档案分类设置

Controller：ClassificationController是负责处理档案分类设置的控制器组件，它接收前端请求并将其转发给后续处理程序。

View：Classification.jsp是用于展示档案分类设置页面的视图，负责接收必要的请求参数和展示页面内容。

Model：ClassificationService是处理档案分类设置业务逻辑的服务组件，它负责校验输入数据、更新数据库等操作。

URL Pattern：档案分类设置的URL Pattern被设置为/archive/classification，用于匹配前端发起的档案分类设置请求。

## 4.4 档案架设置

Controller：ShelfController是负责处理档案架设置的控制器组件，它接收前端请求并将其转发给后续处理程序。

View：Shelf.jsp是用于展示档案架设置页面的视图，负责接收必要的请求参数和展示页面内容。

Model：ShelfService是处理档案架设置业务逻辑的服务组件，它负责校验输入数据、更新数据库等操作。

URL Pattern：档案架设置的URL Pattern被设置为/archive/shelf，用于匹配前端发起的档案架设置请求。

# 五、数据类型转换表设计

## 5.1 数据类型转换表



## 5.2 档案索引规则DTO

该DTO用于在不同层之间传递档案索引规则的数据。它包含与档案索引规则相关的属性，例如规则ID、规则名称等。通过使用该DTO，可以将档案索引规则的数据从业务层传递到表示层，或者从表示层传递回业务层。

## 5.3 档案登记表DTO

该DTO用于在不同层之间传递档案登记表的数据。它包含与档案登记表相关的属性，例如登记ID、文件名称等。通过使用该DTO，可以将档案登记表的数据从业务层传递到表示层，或者从表示层传递回业务层。

## 5.4 档案分类表DTO

该DTO用于在不同层之间传递档案分类表的数据。它包含与档案分类表相关的属性，例如分类ID、分类名称等。通过使用该DTO，可以将档案分类表的数据从业务层传递到表示层，或者从表示层传递回业务层。

## 5.5 档案架信息表DTO

该DTO用于在不同层之间传递档案架信息表的数据。它包含与档案架信息表相关的属性，例如架ID、位置等。通过使用该DTO，可以将档案架信息表的数据从业务层传递到表示层，或者从表示层传递回业务层。

# 六、数据校验表设计



这个数据校验表是为档案归档而设计的，用于验证传入的档案数据是否符合规定的规则和要求。以下是对数据校验表的分析：

（1）档案编号（档案编号）：这个字段的数据类型是字符串，用于表示档案的唯一编号。在数据校验时，可以对档案编号进行长度校验和字符集校验，确保档案编号符合预定的格式要求。

（2）档案名称（档案名称）：这个字段的数据类型是字符串，用于表示档案的名称。在数据校验时，可以对档案名称进行长度校验和字符集校验，以确保档案名称的合法性。

（3）档案分类（档案分类）：这个字段的数据类型是字符串，用于表示档案的分类信息。可以根据具体的业务需求，在数据校验时对档案分类进行枚举值或指定值域的校验，确保档案分类的准确性和一致性。

档案架位置（档案架位置）：这个字段的数据类型是字符串，用于表示档案在档案架上的位置信息。在数据校验时，可以对档案架位置进行格式校验，确保档案架位置的格式符合预定的要求。

（4）创建日期（创建日期）：这个字段的数据类型是日期，用于表示档案的创建日期。在数据校验时，可以对创建日期进行范围校验和格式校验，以确保创建日期的合法性。

（5）创建人员（创建人员）：这个字段的数据类型是字符串，用于表示档案的创建人员信息。在数据校验时，可以对创建人员信息进行长度校验和字符集校验，确保创建人员信息的准确性。

（6）档案状态（档案状态）：这个字段的数据类型是字符串，用于表示档案的状态信息。可以根据具体的业务需求，在数据校验时对档案状态进行枚举值或指定值域的校验，确保档案状态的准确性和一致性。

通过对这个数据校验表的设计，可以在档案归档过程中对传入的数据进行严格的验证，避免无效或错误的数据进入系统。数据校验的规则和要求可以根据具体的业务需求进行调整和扩展，以满足实际应用场景的需求。同时，数据校验还可以提供错误提示和反馈，帮助用户更好地理解和解决数据输入错误，提高系统的用户友好性和易用性。

# 七、国际化设计

档案归档系统的国际化设计方案是为了实现系统界面、文本、日期格式等内容的本地化，以便适应不同语言和地区的用户需求。

## 7.1 多语言资源文件

创建一个资源文件，例如 messages.properties，用于存储系统中的文本、标签、错误信息等内容的默认语言版本（例如英语）。

创建其他语言版本的资源文件，例如 messages\_zh\_CN.properties（中文）和 messages\_fr\_FR.properties（法语），分别存储对应语言的内容。

在资源文件中以键值对的形式存储文本，例如 welcome.message=Welcome to the archive filing system!。

## 7.2 日期和时间格式本地化

在系统的配置文件中，设置默认的日期和时间格式，例如 yyyy-MM-dd 和 HH:mm:ss。

根据用户的语言设置，动态加载对应的日期和时间格式。

在需要显示日期和时间的地方，使用合适的日期和时间格式进行格式化，例如使用 SimpleDateFormat 类进行日期和时间的格式化操作。

## 7.3 动态切换语言

在系统的用户界面中，提供一个语言选择的下拉菜单或按钮，用于切换系统的显示语言。

当用户选择不同的语言选项时，通过触发相应的事件或请求，将用户选择的语言信息发送到后台。

在后台通过将用户选择的语言信息保存在用户会话或配置中，以便在系统的其他部分获取和使用该语言信息。

在需要显示文本的地方，通过读取用户的语言设置，加载对应的资源文件，并根据资源文件中的键值对替换相应的文本内容。

# 系统验证说明表



该验证说明表可以在系统中用于验证用户角色和权限的访问控制，确保只有具有归档权限的用户可以执行档案归档操作，而其他用户则被限制在系统的其他功能或只具有只读权限。

实际实现中，可以将该验证说明表与用户角色和权限的管理模块结合起来，以实现动态权限分配和验证功能。例如，可以在用户登录时通过验证用户角色和权限表来确定其访问权限，并根据权限配置显示或隐藏相关功能按钮或页面。

# 操作日志拦截器设计



在档案归档操作中，ArchiveController 是被拦截的 Controller 类，拦截的方法为 archiveOperation，拦截的时间点是在方法执行前。当用户进行档案归档操作时，拦截器会在方法执行前触发，记录相应的操作日志。这样可以方便地追踪和审计档案归档操作，增加系统的安全性和可追溯性。

# 问题件操作日志设计



在以上表格中，ArchiveController 是被拦截的 Controller 类，拦截的方法为 problemArchiveOperation，拦截的时间点分别是方法执行前和方法执行后。在方法执行前，拦截器会记录问题件档案归档操作的相关信息，例如涉及的档案ID和问题描述。而在方法执行后，拦截器会记录问题件档案归档操作的结果，包括是否成功和失败原因等信息。

这样设计问题件操作日志表可以帮助系统管理员和开发人员快速定位和解决档案归档过程中可能出现的问题，提高系统的可靠性和稳定性。同时，也有助于监控和优化档案归档系统的性能和用户体验。

# 十一、SpringMVC标签库设计



这些标签提供了在档案归档系统中常见的表单元素和功能，可以方便地在SpringMVC的视图页面中使用。通过这些标签，开发人员可以更快速地构建表单页面，并实现表单数据的输入、验证和提交等功能。同时，还提供了显示验证错误信息、国际化消息和主题等辅助功能，提升用户体验和系统的可定制性。

# 十二、Mybatis参数表设计



这些参数涵盖了档案归档流程中的关键信息和操作记录，包括档案的基本信息（如ID、名称、类型、状态、存放位置等）、索引规则、分类、档案架号、编码，以及创建和更新的用户信息、时间戳和备注。这些参数将被传递给MyBatis的SQL语句中，用于进行数据库操作，如插入、更新、查询等。

# 十三、JDBC数据源用户角色表设计



在这个设计中，我们将"archiveDataSource"的JDBC数据源用于档案归档系统的数据访问。为了保证数据的安全性和权限控制，我们定义了两个使用角色及其对应的操作权限：

ROLE\_ADMIN：该角色拥有全部数据源的读写权限，即具有对数据源中所有表的增删改查操作的权限。

ROLE\_USER：该角色只拥有部分数据源的读权限，即只能对数据源中的某些表进行查询操作，而无法进行数据修改操作。

因为本档案管理系统使用了用户角色权限动态分配的功能，所以系统中仅存在两个角色。而user的权限会因admin的分配而不同。

通过为不同的用户分配不同的角色，我们可以实现对档案归档系统数据的精确控制和安全管理。根据用户的角色，系统可以在执行相关操作前进行权限验证，以确保用户只能访问其具有权限的数据。

# 十四、档案归档会话Mapper设计表



# 十五、domain与数据表或视图的对应关系

‘

# 十六、mybatis用户权限表

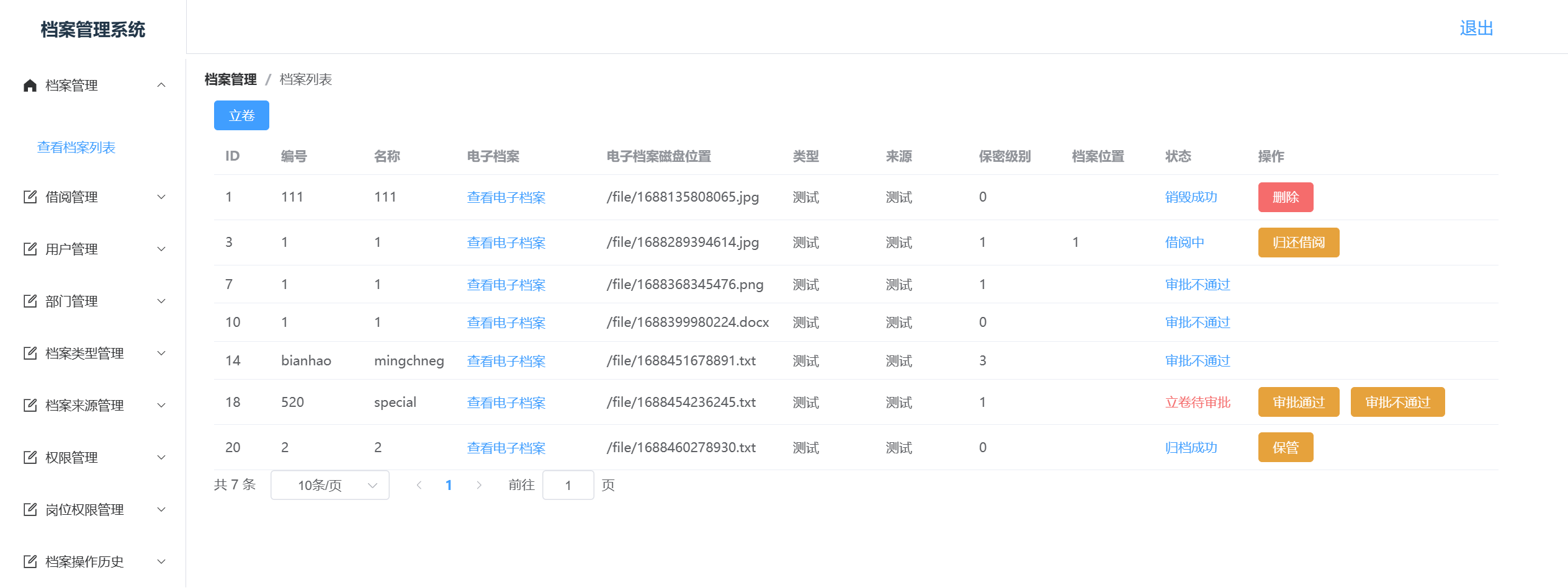


# 十七、每一用例界面元素设计表



# 十八、界面设计，界面元素说明

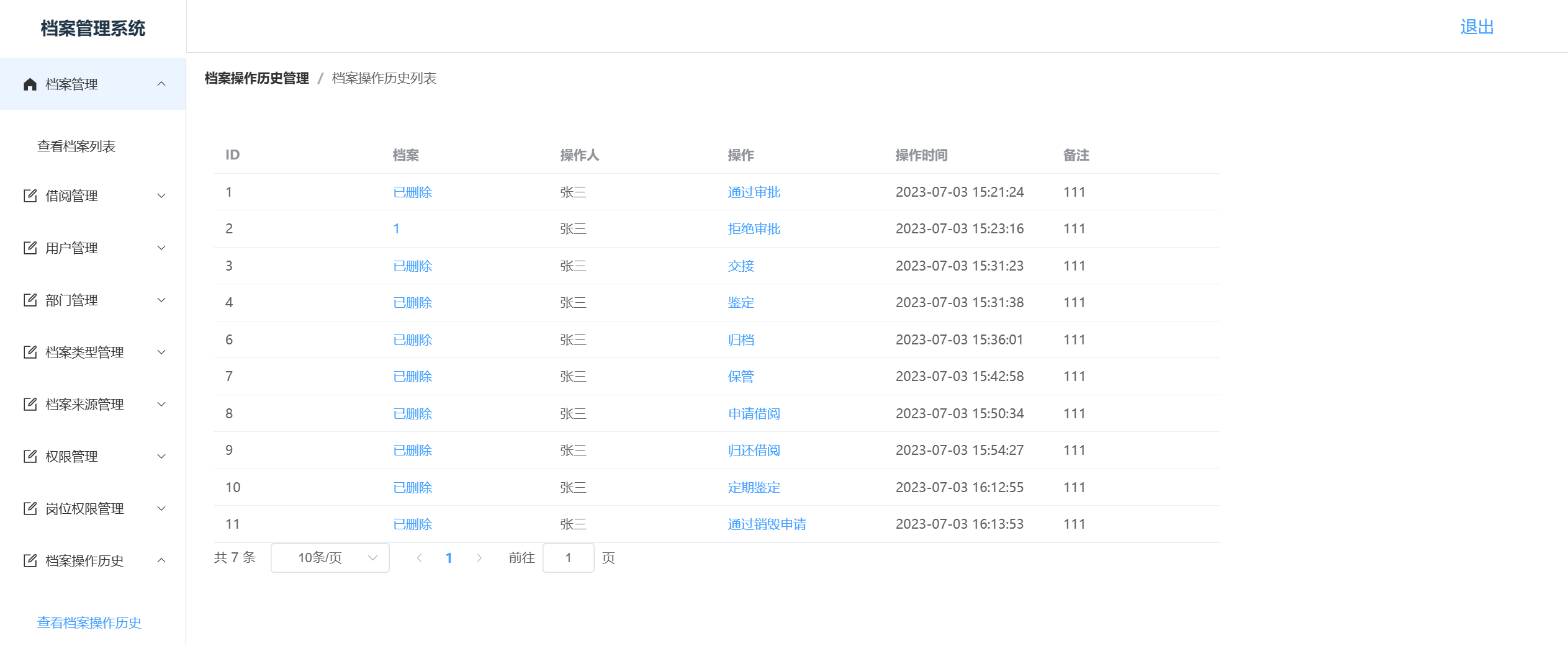
立卷之后会显示ID、编号、名称、电子档案、电子档案磁盘位置、类型、来源、保密级别、档案位置、状态等，经过审批、归档等操作就可以进入保管操作，会有一个保管功能按钮，如下：



在鉴定之后，点击档案归档共能够可以进行分类，输入档案分类信息：

档案完成分类后，可以对档案的档案架进行分配：

可以在档案操作历史看到记录，包括ID、档案、操作人、操作、操作事件和备注，如下：



# 十九、状态改变表



# 二十、数据元素结构设计

## 20.1 档案索引规则元素结构

规则类型：内容、存储形式、保密等级

规则内容：科研文档、认识文档、财务文档等

规则编码：A、B、C等

## 20.2 档案信息元素结构

档案名称：文档1、文档2、文档3等

其他档案信息字段：例如档案类型、创建日期、文件大小等

## 20.3 档案架信息元素结构

档案架剩余空间：表示档案架还能容纳的档案数量

档案架所属分类：表示档案架所属的分类类型

档案架编号：唯一标识档案架的编号

## 20.4 选择档案架元素结构

档案架剩余空间

档案架所属分类

档案架编号

档案名称

其他档案信息字段

# 二十一、系统运行设计

## 21.1 系统架构

该系统采用前后端分离的架构模式，前端使用Vue框架，后端使用Spring MVC框架。前端和后端通过RESTful API进行数据交互。部署在服务器上时，可以采用Tomcat容器来运行应用程序。

## 21.2 技术栈

* 前端技术栈：Vue.js、Vue Router、Vuex、Element UI、Axios等
* 后端技术栈：Java、Spring MVC、MyBatis、MySQL等

## 21.3 开发流程

（1）定义数据库数据表和实体类，并编写对应的DAO类和Mapper映射文件；

（2）设计前端UI界面，并定义前端数据模型和视图模型；

（3）在后端编写Controller类和Service类，提供相应的API接口并实现业务逻辑；

（4）在前端编写组件，并使用Axios库调用后端API接口获取数据；

（5）整合前端和后端，并进行相关测试和调试；

（6）部署应用程序到服务器上，并配置相关依赖和环境变量。

## 21.4 数据交互流程

前端和后端通过HTTP协议进行通信，前端通过Axios库向后端发送请求，后端通过Controller类接收请求，并将请求转发给Service类进行处理。Service类进行相应的业务逻辑处理，之后返回数据给Controller类。Controller类将处理结果封装成JSON格式，并返回给前端。

# 二十二、主要算法设计

## 22.1 校验算法

前端校验：前端通过一系列验证规则对用户输入的档案信息进行校验，例如必填字段、格式验证等。

后端校验：后端接收前端传递的档案信息，对其进行进一步校验，例如权限验证、合法性验证等。

## 22.2 输入检测算法

前端输入检测：前端对用户输入的档案信息进行检测，确保输入的合法性和完整性。

后端输入检测：后端对前端传递的档案信息进行再次检测，确保数据的有效性和完整性。

## 22.3 档案编码确认算法

根据档案归档流程中的要求，通过某个方法或算法生成档案的编码。

输入档案的信息，例如权限等级、部门、立卷事件等。

根据指定的编码规则，生成对应的档案编码。