基于 SpringBoot 的古建筑数字化信息管理系统设计与实现*

左悦
南宁学院土木与建筑工程学院
南宁 530200

林宁
南宁学院信息工程学院
南宁 530200

摘 要 古建筑作为我国历史文化的重要载体，是前人文明和智慧的结晶，是我国乃至世界物质文化遗产的瑰宝，其具有信息量大、信息类型多样、资料繁杂等特点，为其保护和传承带来诸多难题。近年来，数字化技术高速发展，为古建筑领域的保护研究带来了新的机遇。本文针对古建筑数字化信息管理的需求，提出了一种基于 SpringBoot 的古建筑数字化信息管理系统设计方案。该系统采用前后端分离的技术架构，实现了古建筑信息的采集、存储、管理等功能，为古建筑的保护和传承提供了有效的技术支持。

关键字 Spring Boot，古建筑，数字化信息管理，前后端分离

Design and Implementation of the Digital Information Management System of Ancient Buildings Based on SpringBoot

Zuo Yue
School of Information Engineering
Nanning University, Nanning 530200, China
gladyz5487 @126.com

Li Ning
School of Information Engineering
Nanning University, Nanning 530200, China
bgy_2009@163.com

Abstract—As an important carrier of China's history and culture, ancient architecture is the crystallization of predecessors' civilization and wisdom, and the treasure of the material cultural heritage of China and even the world. It has the characteristics of large amount of information, diverse information types and complex data, which brings many problems for its protection and inheritance. In recent years, the rapid development of digital technology has brought new opportunities for the protection and research in the field of ancient architecture. According to the needs of the digital information management of ancient buildings, this paper puts forward a design scheme of the digital information management system based on Spring Boot. The system adopts the technical architecture of separating the front and rear ends, which realizes the functions of the collection, storage, retrieval and display of the information of ancient buildings, and provides effective technical support for the protection and inheritance of ancient buildings.

Key words—Spring Boot, Ancient architecture, digital information management, front and rear end separation

1 引 言

我国古建筑有其独特的建筑体系，作为我国历史文化的重要载体，承载着丰富的历史信息和文化内涵，对历史、文化、艺术和科学技术发展等领域都有着重要的研究价值。然而，由于自然环境和人为因素的影响，许多古建筑都遭受到了不同程度的破坏。近年来，国家层面越来越重视对我国文化遗产的保护和传承，增强人民的文化自信以及民族自豪感和认同感，各种相关保护政策的出台为古建筑领域的研究工作带来了机遇。我国古建筑结构复杂，形式多样，地域特征显著，使得其信息量巨大，信息冗杂，给保护工作带来了困难。为了更好地传承这些宝贵的文化遗产，对古建筑进行数字化信息管理显得尤为重要[1]。

数字化信息管理是指通过计算机技术和网络技术，对利用数字化设备采集的各类信息进行存储、检索、展示、分析等操作，以实现信息的智能化管理。而数据库技术是计算机数据处理和信息管理系统的核心，它解决了大量数据有效组织和存储的问题，并能够对数据进行高效获取和处理。国外关于历史文化遗产的数字化保护这项工作的开展较早，并在构建多媒体数据库、对文物进行数据收集、可视化文物方面取得了一定的研究成果[3]。而国内对历史文化遗产的保护和数字化管理工作相对起步较晚，虽然也取得了一

*基金资助：本文得到南宁学院教授培育工程项目(2019JSGC14)资助。

些标志性成果，但仍处于探索阶段。因此，本文提出了一种基于 SpringBoot 的古建筑数字化信息管理系统设计方案，旨在为古建筑的保护和传承提供更多的思路和有效的技术支持。

2 系统总体设计

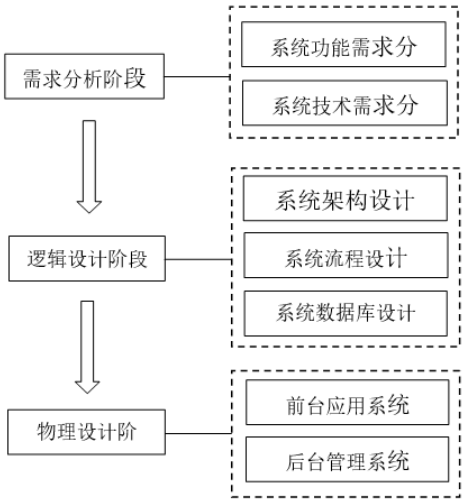

图 1 系统设计总体思路

3 系统需求分析

3.1 系统功能需求

系统的主要目的是对古建筑信息进行管理，因此系统应包含古建筑信息管理功能；角色管理功能；用户管理功能；系统管理功能。如图2所示。

3.2 系统技术需求

古建筑信息管理系统使用 JAVA 编程语言进行研究开发，操作系统用 Windows 系列，浏览器应用程序用谷歌浏览器和 Microsoft Edge 浏览器，后端环境要求：jdk 8 以上、maven3.6 以上、SpringBoot2 以上、开发工具：idea、git，前端环境要求：node.js、python3 以上、开发工具：WebStorm 或 VS Code、git。

（1）开发的框架

本项目基于 SpringBoot 框架进行开发，SpringBoot 是一个快速开发的 JavaWeb 框架，具有开发效率高、易于维护和可扩展性强等优点。同时 SpringBoot 还提供了一系列的插件和工具，使得开发过程更加高效和便捷。

（2）数据库

本项目采用 MySQL 数据库，MySQL 是一种开源的关系型数据库管理系统，具有开发成本低、性能稳定和兼容性强等特点。同时 MySQL 还支持高可用性和可扩展性，能够满足本项目的数据存储需求。

（3）前端框架

前端使用 Vue.js 框架进行开发，采用组件化的方式进行功能拆分。通过调用后端提供的 RESTful API 接口，实现用户界面的展示和交互。前端代码遵循 MVVM 模式进行编写，保证了代码的可维护性和可扩展性。

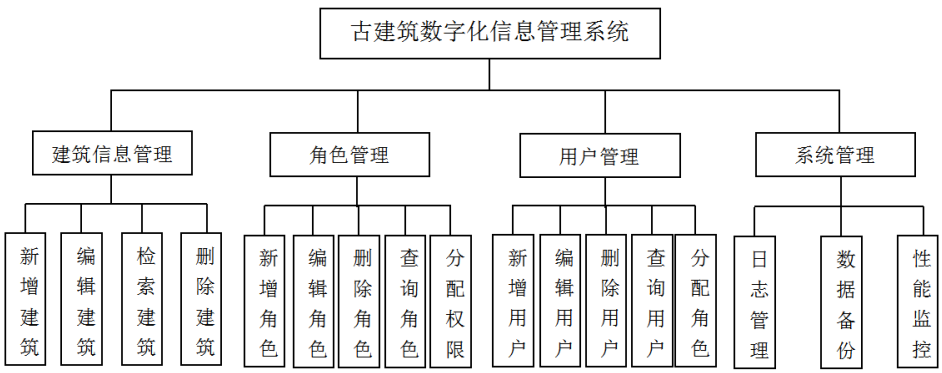

图 2 系统功能需求

（4）后端实现

后端使用 SpringBoot 框架进行开发，实现了一个基本的 RESTful API 接口。通过编写 Controller、Service、Dao 等层次的代码，实现了对古建筑信息的增删改查等操作。后端代码遵循面向对象的编程原则，保证了代码的可读性和可维护性。

（5）通信技术

本项目需要实现不同模块之间的通信和数据交换，可采用 HTTP、WebSocket 等通信技术实现。

综上所述，开发系统采用的开发框架、数据库、前端框架和通信技术都是目前比较成熟和稳定的技术手段，且具有良好的兼容性和扩展性，能够满足系统开发需求。

3.3 系统性能需求

（1） 系统响应时间要求系统应具有快速响应的特性，用户使用系统时简单的操作系统所响应的时间应该控制在 1 秒以内，而较为复杂的操作、需要后台数据处理的操作时间范围则应该控制在 3 秒到 5 秒之间。

（2） 系统安全性系统应当有一定的安全性，只能对合法用户做出登录和使用系统的反应,并且对每个用户做出权限设置，明确每个用户的权限。系统还应该保护用户的后台数据安全，比如用户的登录名、密码、以及用户重要信息，保证账号信息安全。

（3） 目标系统用户界面应简洁易懂，用户使用时操作应简单明了，充分考虑系统与用户交互简单化。所有操作时的语言均采用中文语言系统，界面为中文图形化，使非技术性人员在初步接触系统时也通过摸索掌握系统大致使用方式，再经过一些简单的训练就能熟练使用系统的大部分功能。

（4） 系统应拥有一定程度的可扩展性，须采用模块化设计的工作理念使其设计开发相对灵活，可根据后续用户体验，不断更新其模块与功能或应用户要求作出相应的调整，也可在已完成的系统基础上进行不同程度的再次开发。

4 古建筑数据库设计

4.1 概念模型设计

古建筑信息量大、数据类型较多且信息冗杂，通过构建数据库对基础信息、文史信息、空间信息进行系统地组织，构建数据之间的逻辑联系，便于信息的管理和应用。而要将古建筑保存到计算机中，首先需要将现实世界抽象为信息世界，既把现实世界中的客观对象抽象为某一种信息结构，这种信息结构不依赖于具体的计算机系统，称之为概念模型。因此要构建古建筑信息数据库，首先需要对古建筑进行概念模型设计。古建筑信息从内容上可以分为建筑要素信息和人文要素信息。

建筑要素信息包括建筑基本信息、建筑构件信息、建筑空间信息。

建筑要素信息：

建筑基本信息（建筑编号，建筑名称，地理坐标，建筑面积，建筑年代，建筑历史，建筑功能）

建筑构件（构件编号，建筑编号，构件位置，构件类型，构件名称，构件照片，构件描述）

建筑空间（建筑编号，测绘图，立面与细节照片，三维点云模型）

人文要素信息包括传统工艺信息、历史事件信息、民风习俗信息。

人文要素信息：

传统工艺（传统工艺编号，工艺名称，工艺介绍，工艺过程照片，工艺过程视频，工艺场所编号）

历史事件（历史事件编号，事件名称，事件介绍，事件照片，事件视频，事件场所编号）

民风习俗（民俗活动编号，活动名称，活动介绍，活动照片，活动视频，活动场所编号）

4.2 E-R 图设计

E-R图又称实体-联系图，实体指客观上可以相互区分的事物，联系则反映实体内部或实体之间的关联。E-R图是用来表示实体及其关联的一种信息结构图。联系有一对一、一对多及多对多三种类型。根据对古建筑信息的分析和描述，构建古建筑E-R图见图3。

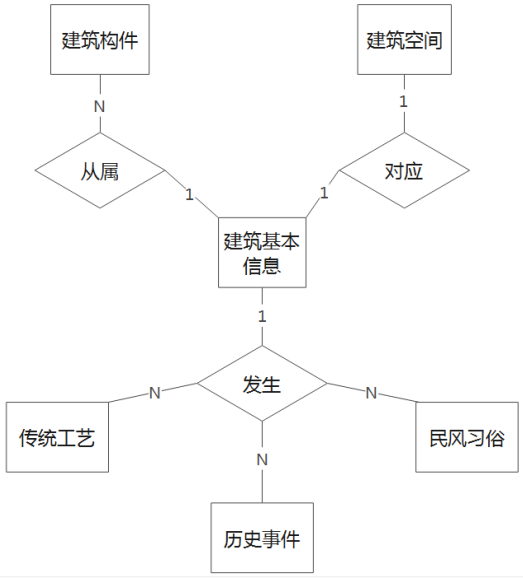

图 3 古建筑E-R图

4.3 数据表设计

在完成古建筑概念模型设计后，需要将其转换为某数据库管理系统支持的逻辑模型，才能够完成由现实世界向机器世界的转换，即将古建筑信息存放到数据库中。关系数据库作为主流数据库，采用关系模型作为数据组织方式，即以二维表的形式对数据进行存储，易于后期的操作和管理。根据古建筑 E-R 图，对数据表设计如表 1-表 6。

表 1 建筑基本信息表（JZJB）

字段名称	数据类型	是否允许空值	条件约束	说明
Bid	bigint	否	主键、自动递增	建筑编号
Bname	varchar	否		建筑名称
Bpos	text	否		地理坐标
area	double	是		建筑面积
age	varchar	是		建筑年代
fun	text	是		建筑功能
history	text	是		建筑历史

表 2 建筑构件表（JZGJ）

字段名称	数据类型	是否允许空值	条件约束	说明
Cid	bigint	否	主键、自动递增	构件编号
Bid	bigint	否	主键，外键	建筑编号
Cname	varchar	否		构件名称
type	varchar	是		构件类型
Cloc	text	是		构件位置
Cpic	text	是		构件照片
Cdes	text	是		构件描述

表 3 建筑空间表（JZKJ）

字段名称	数据类型	是否允许空值	条件约束	说明
Bid	bigint	否	主键，外键	建筑编号
Sdraw	varchar	否		建筑测绘图
ele	text	是		立面与节点照片
Bmod	text	是		三维点云模型

表 4 传统工艺表（CTGY）

字段名称	数据类型	是否允许空值	条件约束	说明
Gid	bigint	否	主键、自动递增	传统工艺编号
Gname	varchar	否		工艺名称
Gdes	text	是		工艺介绍
Gpic	text	是		工艺过程照片
Gvideo	text	是		工艺过程视频
Bid	bigint	是	外键	工艺场所编号

表 5 历史事件表（LSSJ）

字段名称	数据类型	是否允许空值	条件约束	说明
Hid	bigint	否	主键、自动递增	历史事件编号
Hname	varchar	否		事件名称
Hdes	text	是		事件介绍
Hpic	text	是		事件照片
Hvideo	text	是		事件视频
Bid	bigint	是	外键	事件场所编号

表 6 民风习俗表（MFXS）

字段名称	数据类型	是否允许空值	条件约束	说明
Fid	bigint	否	主键、自动递增	民俗活动编号
Fname	varchar	否		活动名称
Fdes	text	是		活动介绍
Fpic	text	是		活动照片
Fvideo	text	是		活动视频
Bid	bigint	是	外键	活动场所编号

5 系统功能实现

对于开发实现一个比较完善的古建筑信息管理系统，不仅对其安全性和系统性能有着高标准的需求，也还需要有着规范合理的可视化界面，为古建筑信息的查询、检索、模型还原、数据维护等功能。

5.1 系统用户登录的实现

（1）用户登录认证业务逻辑

此模块提供了用户的登录功能，用户进入系统网址后在必输项用户名、密码、验证码中输入正确的内容，页面 JS 校验账号、密码、验证码正确后，提交前端数据到 Controller 层，将密码与用户表的账户和密码对比一致后生成 token 安全认证码后完成登录。

（2）用户登录逻辑核心代码：

前端往后端发送用户信息代码：

```
// 向后端发送请求
```

```
var values = {};
values.Username = data.Username;
values.Password = data.Password;
values.Type = data.Type;
axios.post('loginSystem/Login', values).then(res => {
        var data = res.data;
        if (data.code == 0){
          //获取 token 存入
          localStorage.setItem("Token", data.Token)
          localStorage.setItem("UserId", data.UserId)
          localStorage.setItem("UserRealname",
data.UserRealname)
          localStorage.setItem("UserName", data.UserName)
          localStorage.setItem("UserType", data.UserType)


          layer.msg(res.data.msg, {icon: 1},function (){
              window.location = 'index.html'});
           }else {
              layer.msg(res.data.msg, {icon: 5});
           }
          })
// 后端代码实现
@RestController
@CrossOrigin
@RequestMapping("/loginSystem")
public class LoginController {
  private UserInfoService userInfoService;
  private AdminService adminService;
  private SuperAdminMapper superAdminMapper;
          private JsonObject jsonObject;

          @RequestMapping(value = "/Login", method =


  RequestMethod.POST)
  @ResponseBody
    public Map login(@RequestBody
    Map<String,String> map){
        Map UserMap = new HashMap();
        String Username= map.get("Username");
        String Password= map.get("Password");
        String Type= map.get("Type");
        char[] chars = Type.toCharArray();
        char myType = chars[0];
     // 根据前端发送的用户，判断该用户是否存储
    if (myType=="admin"){
        Map AdminMap = new HashMap();
        /*为管理员用户，跳转到管理员界面*/
    }
    else
    {
         Map UserMap = new HashMap();
        /*为普通用户，跳转到普通用户界面*/
    }
```

5.2 图片上传及动态树形下拉业务逻辑

前端使用 layui 文件上传模块对文件进行上传，且图片上传为单独请求不与整页数据进行提交，图片上传模块在上传文件后到 Contorller 层进行文件存储处理，使用 FileOutputStream 对象进行文件上传处理。动态树形下拉及是使用 layui 当中的 tree 模块和 java 的 tree 工具类，也是设置好父级 ID 与子 ID 即可。

图片上传及动态下拉核心代码：

```
    // 初始化插件
$("#zyupload").zyUpload({
    width: "650px",
    height: "350px",
    itemWidth:                                    "140px",
itemHeight: "115px",
    url:  "http://localhost:9990/system/headerImgUpload",// 上传
文件的路径
    fileType: ["jpg", "png"],
    fileSize: 51200000,
    multiple: true,
    dragDrop: true,
    tailor: true,
    del: true,
    finishDel: false,                          /
    onSelect: function (selectFiles, allFiles) {
        // 选择文件的回调方法  selectFile:当前选中的文件
allFiles:还没上传的全部文件
    console.info("当前选择了以下文件：");
    console.info(selectFiles); },
    onDelete: function (file, files) {
        // 删除一个文件的回调方法 file:当前删除的文件
files:删除之后的文件
    console.info("当前删除了此文件：");
    console.info(file.name); },
    onSuccess: function (file, response) {
        // 文件上传成功的回调方法
        console.info("此文件上传成功：");
        console.info(file.name);
        console.info("此文件上传到服务器地址：");
         console.info(response);
       //$("#uploadInf").append("<p>上传成功，文件地址是："
+ response + "</p>");
          $("#diPhoto").val(response);
      },
    onFailure: function (file, response) {
     //文件上传失败的回调方法
    layer.msg("文件上传失败",{icon: 5});
    console.info("此文件上传失败：");
    console.info(file.name); },
    onComplete: function (response) {
    // 上传完成的回调方法
        layer.msg("文件上传完成",{icon: 1});
        onsole.info("文件上传完成");
        console.info(response); }
});
```

5.3 建筑模型信息存储的实现

Revit 是一款强大的建筑信息模型软件，提供了三维建模功能，广泛应用于建筑设计、施工和维护领域。Revit 支持用户在三维环境中进行建筑设计，可以实时查看建筑物的外观、结构和系统，用户也可以通过参数化建模方式快速调整建筑物的尺寸和形状，实现灵活的设计变更。

在古建筑信息保护中，利用 Revit 软件构建古建筑信息模型，可以将古建筑信息转化为数字模型得到永久保存。但创建好的 Revit 建筑模型只能以单文件的形式进行存储，大量的信息模型文件，不利于查看、检索模型信息，也不能进一步利用计算机技术智能处理模型信息。因此，将古建筑模型信息保存到数据库表中，是实现模型信息检索、智能化处理的关键一步。

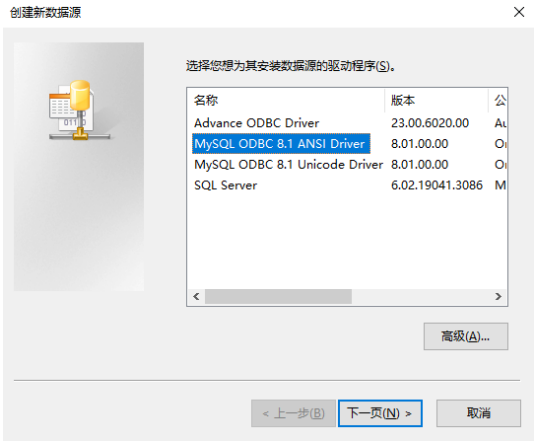

图 4 选择数据库驱动程序

图 5 信息模型导出到数据库

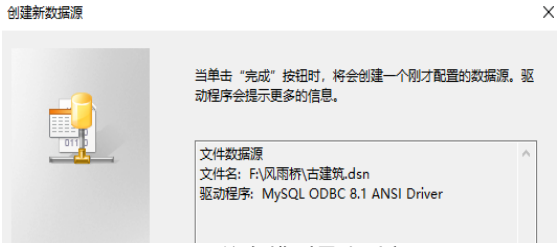

图 6 信息模型导出过程

Revit 信息模型存储的实现：

(1) 安装数据库服务器：首先，你需要安装一个数据库服务器，如 SQL Server、Oracle 或 MySQL 等，为数据库安装驱动程序，以 Mysql 为例。

(2) 创建数据库和表格：在数据库中创建的数据库和表格，用于存储 Revit 模型信息。如：建筑基本信息表、建筑构件表、建筑空间表、传统工艺表等，然后将信息模型内容导出并插入到数据库相应表格中，实现信息模型在数据库中存储。

6 结束语

本文提出了一种基于 SpringBoot 的古建筑数字化信息管理系统设计方案，该系统采用前后端分离的技术架构，具有较好的可扩展性和可维护性。本文设计实现了古建筑信息的存储、管理等功能，通过实际应用和测试，证明了本系统的有效性和可行性。为下一步，将对古建筑模型信息实现智能化处理，提供数据支撑，同时为古建筑的保护和传承提供有效的技术支持。

参 考 文 献

[1] 左悦.BIM技术在古建筑保护中的应用探究[J].城市建筑,2023,20(06):199-202.DOI:10.19892/j.cnki.csjz.2023.06.54.

[2] 杨钰琪. 潮州乡村建筑遗产保护数字化技术路径研究[D].广东工业大学,2023.DOI:10.27029/d.cnki.ggdgu.2022.000961.

[3] 宋昱. 传统建筑装饰数据库构建及应用研究[D].内蒙古工业大学,2022.DOI:10.27225/d.cnki.gnmgu.2021.000471