Advances in Computer, Signals and Systems (2024)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2024.080506
ISSN 2371-8838 Vol. 8 Num. 5

Contract Management System Based on SpringBoot and Vue

Yuanrun Zhu

College of Mathematics and Computer Science, Guangdong Ocean University, Zhanjiang, 524088, China

Keywords: Contract Management System, Java, MySql, SpringBoot, Vue

Abstract: Contracts, as legally binding agreements between enterprises and external entities, carry significant weight concerning corporate interests, norms, and accountability. Their diligent management is vital to maintaining favorable operational dynamics and robust risk mitigation strategies for any business. Traditional contract management practices, however, are marred by inefficiencies, opacity in information flow, and a propensity for errors, rendering them inadequate for the sophisticated demands of contemporary enterprise management. To bridge this gap and address the inadequacies of manual contract administration, this study delves into the meticulous design and implementation of a cutting-edge contract management system. Leveraging the robust capabilities of SpringBoot and the flexibility of the Vue.js framework, the proposed system aims to revolutionize contract handling by introducing automation, enhancing transparency, and minimizing the potential for human error. This digital solution focuses on streamlining every facet of contract lifecycle management, encompassing creation, negotiation, execution, storage, and analysis. By adopting a technology-driven approach, it aspires to empower enterprises with real-time insights, ensuring compliance, and facilitating data-driven decision-making. Ultimately, the system is meticulously crafted to elevate contract management efficiency, thereby reinforcing the overall resilience and competitiveness of modern businesses in an increasingly complex commercial landscape.

1. Introduction

With the rapid development of information technology and the improvement of enterprise management needs, contract management plays a crucial role in enterprise operation. As a legal document agreed between an enterprise and an external entity, the contract involves the interests, norms and responsibilities of the enterprise, and its management is of great significance to the operation status and risk control of the enterprise [1-3]. However, the traditional way of contract management relies on manual operations, including manually creating, approving, signing and filing contracts. This approach is inefficient, prone to delays and errors, and consumes a lot of time and resources. Moreover, in traditional contract management, contract documents are usually stored in paper form or exist in different electronic files. This leads to opacity and dispersion of information [4-5], making it difficult to locate and access contract information, which may lead to loss or omission of information. All of these problems with traditional contract management methods can no longer meet the needs of modern enterprises for contract management.

In order to solve the problems of traditional contract management methods, the system studied in this paper is based on Spring Boot [6-9] with Vue [6-7] framework, which designs and implements an efficient, reliable and visualized contract management system. Based on the simple auto-configuration and rich ecosystem of Spring Boot framework, this system introduces MyBatis [6,10] technology for controlling the operation of Java side and database side. Meanwhile, Shiro [11-12] framework is introduced to realize permission management and ensure the security of the system. In addition, the system also uses Redis [13-14] non-relational database as a data cache, thus improving the response speed and stability of the system. Spring Boot's module-independent design features make the system development of functional modules are independent of each other and do not affect each other, to achieve the purpose of high cohesion and low coupling. This design makes the maintenance and expansion of the system more flexible and convenient. And Vue framework has the characteristics of responsive design, lightweight and high efficiency, which makes the data of the front-end page can be efficiently and asynchronously updated and rendered views, providing new technical support for contract visualization. By introducing the Vue framework, the contract management system can realize an interactive and user-friendly front-end interface, enabling users to view and operate contract data intuitively.

The system can substantially improve the efficiency and accuracy of contract management, reduce the occurrence of human errors, and digitalize and standardize the contract management process. This will help enterprises to enhance their management level, optimize internal processes and reduce risks, so that they can better adapt to the rapidly changing business environment and maintain their competitive advantages.

2. Current status of domestic and foreign research

The adoption of contract management systems has significantly benefited enterprises, with developed nations pioneering their implementation due to mature business processes and established legal frameworks. Since the 1980s, advances in microcomputers led to the concept of paperless offices, spurring the development of automated contract management solutions in these countries. The U.S., a global leader in this field, saw early adoption by companies like General Motors in the 1990s, enhancing operational efficiency. Similarly, the Netherlands' Philips Electronics innovated with a system featuring contract early warning functionality.

China's journey with contract management systems began later but gained urgency post-WTO entry, as companies recognized the limitations of manual methods amidst growing trade volumes. There's now a pressing need for integrated information systems to streamline contract handling, minimize business risks, and reduce costs, especially with the rise of B2B and B2C transactions in the 21st century. While still in its relative infancy, China's contract management landscape holds vast potential for growth, fueled by ongoing technological innovations and the increasing recognition of the value these systems bring to enhancing operational efficiency and competitiveness.

3. System requirements analysis

3.1. System objectives

The significance of the contract management system designed in this paper is to solve the problem of difficulty in finding and accessing contract information when enterprises manage contracts, and to improve the transparency and centralization of contract information while making enterprises reduce delays and errors in the management of contracts. The system's interface design is simple, convenient and easy to use, and it can safeguard information security, avoid leakage of personal information and ensure the normal operation of all functions. The main users of the system are divided into two parts: company employees and administrators. As shown in Figure 1 below.

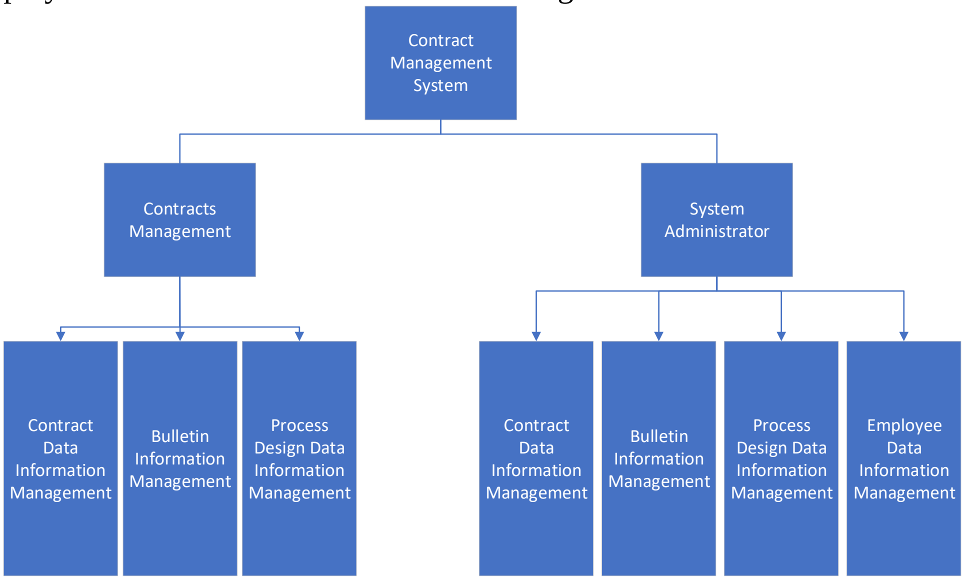

Contract Management System
Contracts Management
System Administrator
Contract Data Information Management
Bulletin Information Management
Process Design Data Information Management
Contract Data Information Management
Bulletin Information Management
Process Design Data Information Management
Employee Data Information Management

Figure 1: Functional structure of the contract management system

3.2. Functional Requirements Analysis for Company Employees

When a company employee enters the system, he/she can perform the following operations:

1) Operation of contract information: adding contract information, modifying contract information, deleting contract information and searching contract information.

2) Operation of announcements: adding announcements, modifying announcements, deleting announcements and searching for announcements.

3) Operation of process design: adding process design, modifying process design, deleting process design and searching process design.

As shown in Figure 2 below.

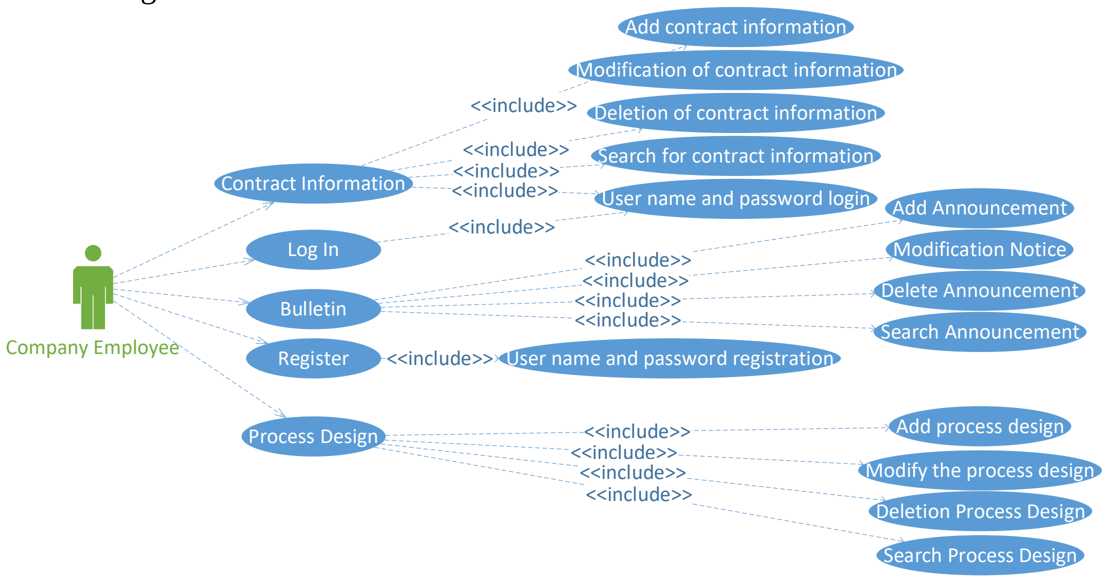


Figure 2: Company Employee Needs Analysis Use Case Diagram

3.3. Analysis of functional requirements for the Administrator

System administrators play an extremely important role in the operation and maintenance of the entire system and their presence is an essential part. When they enter the system, they can perform a variety of operations to ensure the normal operation of the system, these operations include:

1) System management: System administrators can carry out all-round security management of the whole system, regularly check system security, promptly find and deal with system vulnerabilities, prevent hacker attacks and so on. They are also responsible for maintaining the stability of the system to ensure that the system can operate normally in various environments.

2) Data management: System administrators are also responsible for managing the data in the system, including data detection, analysis and storage. They can check whether the data in the system is normal, whether there is any abnormal data, whether there is any problem such as data loss and solve these problems. They can also backup important data to prevent data loss.

3) Monitoring, operation and maintenance: System administrators also need to monitor and operate and maintain the registered microservices. They will check the operation of the microservices on a regular basis, and once they find any problems, they will immediately deal with them to ensure the normal operation of the microservices. In addition, they will also upgrade and optimize the microservices as needed.

As shown in Figure 3 below.

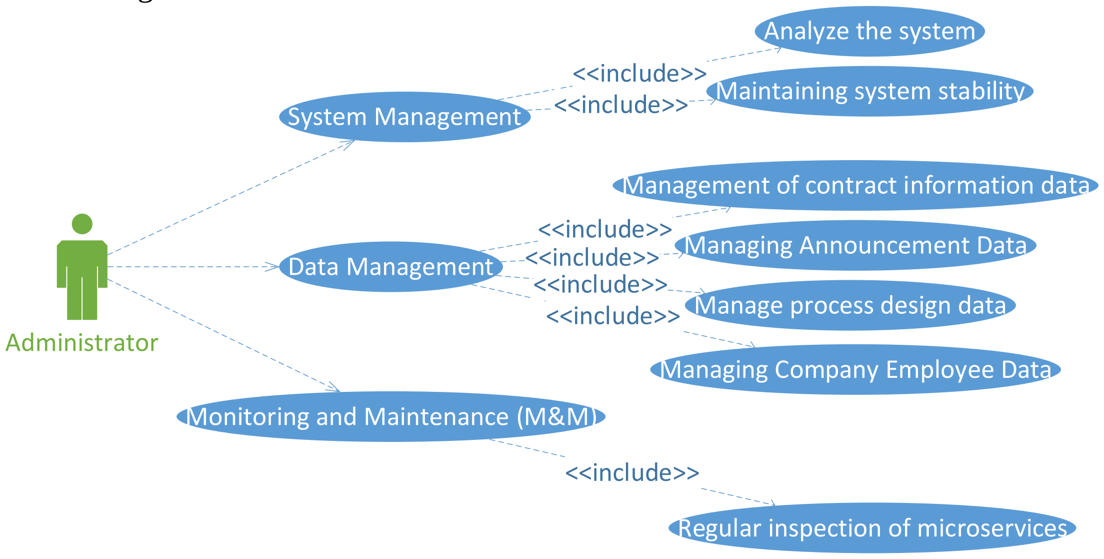

Analyze the system
<<include>>
Maintaining system stability
<<include>>
System Management
Management of contract information data
<<include>>
Managing Announcement Data
<<include>>
Data Management
<<include>>
Manage process design data
<<include>>
Administrator
Managing Company Employee Data
Monitoring and Maintenance (M&M)
<<include>>
Regular inspection of microservices

Figure 3: System Administrator Requirements Analysis Use Case Diagram

4. System design

4.1. Overall system framework

In order to improve the speed and adjustability of development, the SSM development model is adopted, the interface display adopts Vue components to improve the aesthetics of the interface, and the back-end adopts the SpringBoot framework and MyBatis driver, so as to build a system with B/S architecture.

The system architecture diagram is shown in Figure 4.

In order to improve the speed and adjustability of development, the SSM development model is used, the interface display uses Vue components to improve the aesthetics of the interface, and the back-end uses the SpringBoot framework and the MyBatis with SpringBoot Redis driver, so as to build a system with B/S architecture.

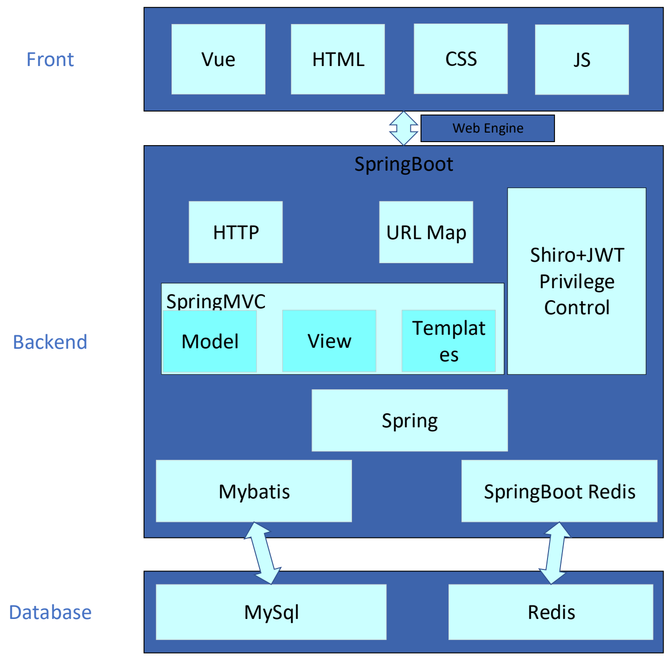


Figure 4: System Architecture Diagram

4.2. Database design

When designing a database we should first consider the user requirements, based on the user requirements for the specific design of the database, the design of the database usually includes the following parts.

4.2.1. Design of the physical structure of the database

The conceptual structure in the previous paragraph needs to be translated into the actual database model in the database system, which is simply the logical structure model of the database. The design of the physical structure of the tables in the Data Forms Library, which contains the user's binding information and login, online query of contract information, online query of announcements, etc., mainly refers to the rational design of a specific database structure. The results of the design of the physical structure of each table in the data form library are shown in the table below.

Table 1: Department table

Column	Data Type	Whether NOT NULL
Id	varchar(32)	NOT NULL
(1:Company,2:Department)Dept_Type	int	NOT NULL
Parent_id	varchar(32)	NOT NULL
Dept_name	varchar(255)	NOT NULL
Dept_code	varchar(255)	NOT NULL
Sort	int	NOT NULL
Create_time	datetime	NOT NULL
Update_time	datetime	NOT NULL
Create_by	varchar(255)	NOT NULL
Update_by	varchar(255)	NOT NULL
Data_flag	int	NOT NULL

Table 2: Contract Document Information Table

Column	Data Type	Whether NOT NULL
Id	varchar(20)	NOT NULL
Create_by	varchar(50)	NOT NULL
Update_by	varchar(50)	NOT NULL
Create_time	datetime	NOT NULL
Update_time	datetime	NULL
Title	varchar(255)	NULL
File_type	varchar(20)	NULL
Doc_type	varchar(20)	NULL
(Document status: 1:Draft, 2:Pending review, 3:Reviewed, 4:Reviewed not approved)Status	varchar(20)	NULL
File_url	varchar(500)	NULL
View_url	varchar(500)	NULL
Failure_reason	varchar(500)	NULL
Download_power	varchar(10)	NULL
View_power	varchar(10)	NULL
View_count	bigint	NULL
Download_count	bigint	NULL
Remarks	varchar(2000)	NULL
Doc_from	varchar(255)	NULL
Key_word	varchar(500)	NULL
Cover	varchar(500)	NULL

Table 3: Announcement table

Column	Data Type	Whether NOT NULL
Id	varchar(32)	NOT NULL
Title	varchar(255)	NULL
Content	longtext	NULL
State	int	NULL
Create_time	datetime	NOT NULL
Update_time	datetime	NOT NULL
Create_by	varchar(255)	NOT NULL
Update_by	varchar(255)	NOT NULL
Data_flag	int	NOT NULL

Table 4: Local file upload configuration table

Column	Data Type	Whether NOT NULL
Id	varchar(32)	NOT NULL
Local_dir	varchar(255)	NULL
Url	varchar(255)	NULL

1)Department table. A department is an entity that categorizes users, and the department table records information about the department, which covers the meaning and type of each field as shown in Table 1.

2)Contract document Information Table. The contract document information table records information about the contract document, which covers the meaning and type of each field as shown in Table 2.

3)Announcement table. The announcement form records information about the contract document, which covers the meaning and type of each field as shown in Table 3.

4)Local file upload configuration table. The local file upload configuration table records the path to store the locally uploaded file, providing a storage path for the file_url field in the contract document information table, which facilitates the subsequent conversion of the document using the LibreOffice plug-in. As shown in Table 4

5. System implementation

5.1. Implementation of the uploading of contract information function

Uploading contract information is shown in Figure 5.
Its code implementation is shown in Figure 6 and 7.

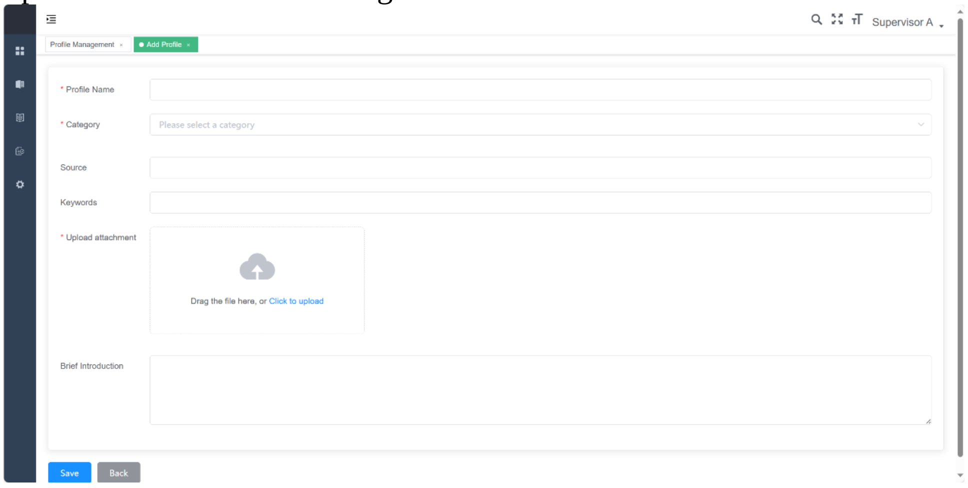
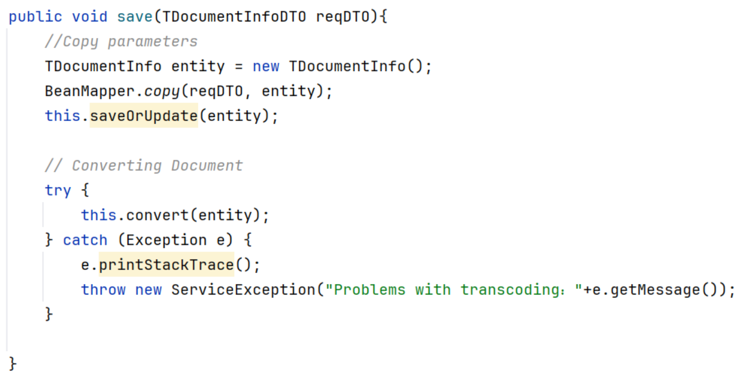
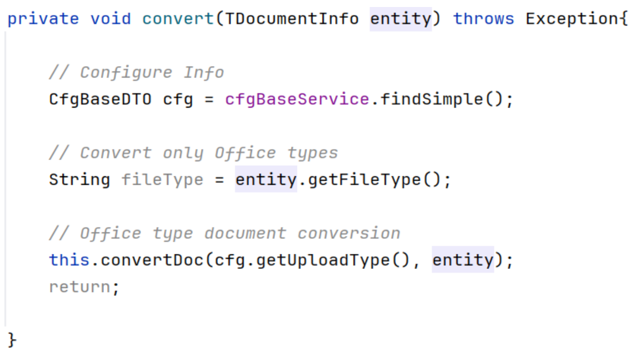


Figure 5: Contract data upload interface



```
public void save(TDocumentInfoDTO reqDTO){
    //Copy parameters
    TDocumentInfo entity = new TDocumentInfo();
    BeanMapper.copy(reqDTO, entity);
    this.saveOrUpdate(entity);

    // Converting Document
    try {
        this.convert(entity);
    } catch (Exception e) {
        e.printStackTrace();
        throw new ServiceException("Problems with transcoding: "+e.getMessage());
    }

}
```

Figure 6: Contract data upload interface back-end Service layer code



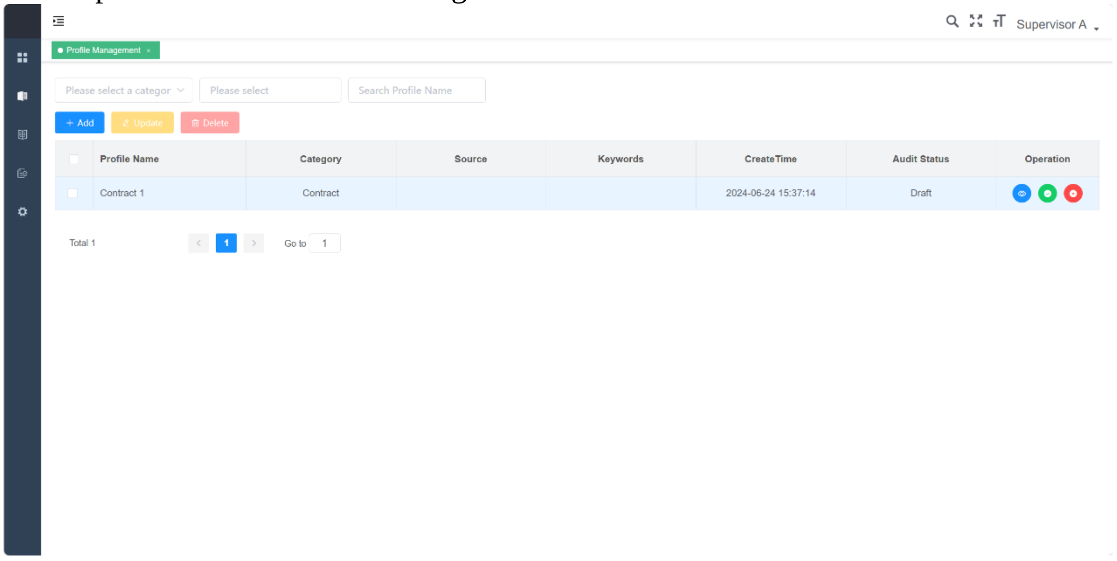
```java
private void convert(TDocumentInfo entity) throws Exception{

    // Configure Info
    CfgBaseDTO cfg = cfgBaseService.findSimple();

    // Convert only Office types
    String fileType = entity.getFileType();

    // Office type document conversion
    this.convertDoc(cfg.getUploadType(), entity);
    return;

}
```

Figure 7: Contract data upload interface back-end Service layer code

Code to achieve the function of uploading contract information, controller layer to receive TDocumentInfoDTO class and pass it into the service layer baseService save method, save method using springboot comes with BeanMapper. TDocumentInfoDTO property value for copying and save to the database, and then call the service layer convert method will be converted to pdf, and will save the document information to the database.

5.2. Functional realization of contract audits

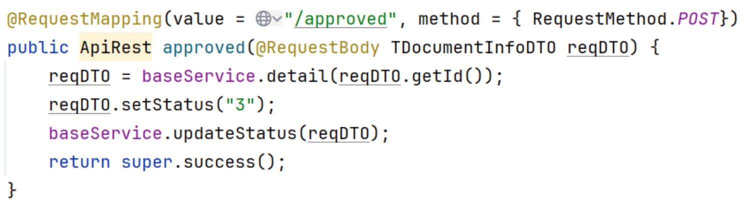
The contract audit is shown in Figure 8.
Its code implementation is shown in Figure 9.



	Profile Name	Category	Source	Keywords	CreateTime	Audit Status	Operation
	Contract 1	Contract			2024-06-24 15:37:14	Draft	

Total 1   1   Go to 1

Figure 8: Contract Review and Approval Screen



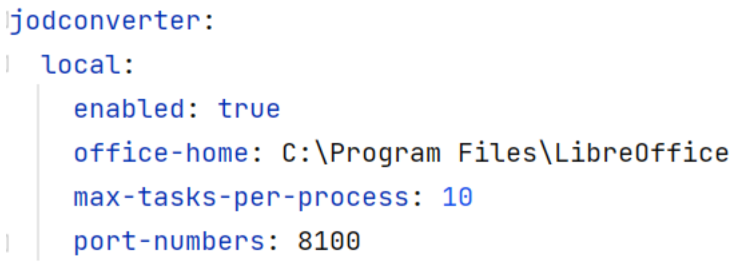
```java
@RequestMapping(value = "/approved", method = { RequestMethod.POST})
public ApiRest approved(@RequestBody TDocumentInfoDTO reqDTO) {
    reqDTO = baseService.detail(reqDTO.getId());
    reqDTO.setStatus("3");
    baseService.updateStatus(reqDTO);
    return super.success();
}
```

Figure 9: Back-end Service layer code for the contract review interface

5.3. LibreOffice plug-in calls

Call LibreOffice plug-ins in SpringBoot need to first install LibreOffice, and then follow the code below to achieve the operation.

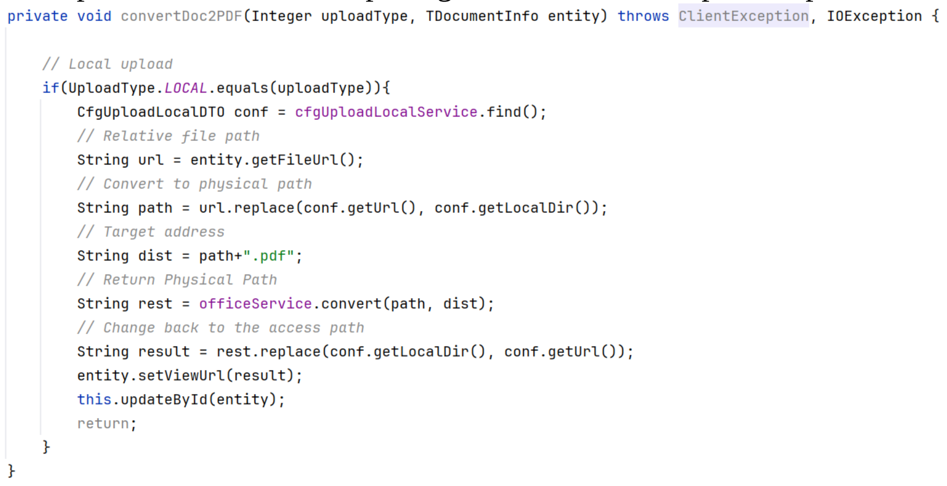
Code implementation shown in Figure 10, Figure 11, Figure 12.



```yaml
jodconverter:
  local:
    enabled: true
    office-home: C:\Program Files\LibreOffice
    max-tasks-per-process: 10
    port-numbers: 8100
```

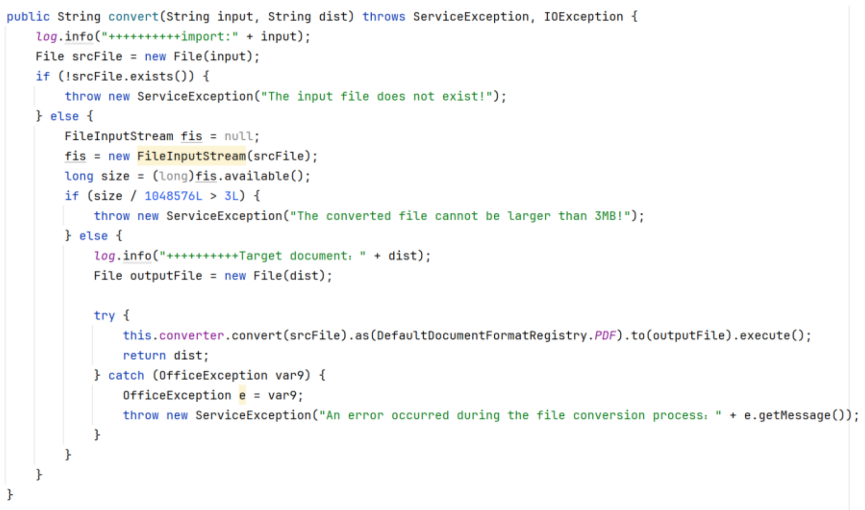
Figure 10: LibreOffice plug-in yml configuration code

Figure 10 is LibreOffice in SpringBoot written in yml configuration, where enabled for whether LibreOffice enabled (true for yes, false for no), office-home for the computer to install LibreOffice path, max-tasks-per-process for the LibreOffice process restart the maximum number of processes before the process, port-numbers for the opening of the LibreOffice process port.



```java
private void convertDoc2PDF(Integer uploadType, TDocumentInfo entity) throws ClientException, IOException {

    // Local upload
    if(UploadType.LOCAL.equals(uploadType)){
        CfgUploadLocalDTO conf = cfgUploadLocalService.find();
        // Relative file path
        String url = entity.getFileUrl();
        // Convert to physical path
        String path = url.replace(conf.getUrl(), conf.getLocalDir());
        // Target address
        String dist = path+".pdf";
        // Return Physical Path
        String rest = officeService.convert(path, dist);
        // Change back to the access path
        String result = rest.replace(conf.getLocalDir(), conf.getUrl());
        entity.setViewUrl(result);
        this.updateById(entity);
        return;
    }
}
```

Figure 11: LibreOffice plugin calling code in springboot



```java
public String convert(String input, String dist) throws ServiceException, IOException {
    log.info("++++++++++import:" + input);
    File srcFile = new File(input);
    if (!srcFile.exists()) {
        throw new ServiceException("The input file does not exist!");
    } else {
        FileInputStream fis = null;
        fis = new FileInputStream(srcFile);
        long size = (long)fis.available();
        if (size / 1048576L > 3L) {
            throw new ServiceException("The converted file cannot be larger than 3MB!");
        } else {
            log.info("++++++++++Target document: " + dist);
            File outputFile = new File(dist);

            try {
                this.converter.convert(srcFile).as(DefaultDocumentFormatRegistry.PDF).to(outputFile).execute();
                return dist;
            } catch (OfficeException var9) {
                OfficeException e = var9;
                throw new ServiceException("An error occurred during the file conversion process: " + e.getMessage());
            }
        }
    }
}
```

Figure 12: LibreOffice plugin wrapper code in springboot

Figure 11 in the officeService.convert method is to call Figure 12 in the convert method, Figure 12 in the convert method in the call libreoffice in the jodjodconverter in the dependency package in the convert method will be the user's incoming documents into pdf documents.

6. Conclusion

The SpringBoot-Vue.js contract management system elevates operational efficiency, streamlining contract processes from creation to completion. It fosters a well-structured database, enhancing compliance and reducing manual administration costs. However, improvements are needed: the system's inefficiency in managing high concurrency and processing large datasets under heavy loads necessitates scalable solutions and advanced data algorithms. Furthermore, the absence of a dedicated mobile interface, amid a mobile-dominated work environment, hampers adaptability. Enhancing compatibility and implementing responsive design would boost cross-platform accessibility. In conclusion, while the system has made significant strides in corporate efficiency and cost reduction, addressing concurrency limitations and integrating mobile optimization are vital for its full potential realization. These advancements will reinforce its role in risk mitigation and resource optimization within today's competitive business landscape.

References

[1] Tang Zhengyin. Introduction to risk management of international contract under DB model[J/OL].Highway, 2024(02):204-208[2024-03-05].http://kns.cnki.net/kcms/detail/11.1668.U.20240126.1720.096.html.
[2] Lu Jing. Analysis of Contract Risk Management and Prevention and Control Countermeasures of State-owned Enterprises [J].Investment and Entrepreneurship, 2024, 35(02):141-143+147.
[3] Duan Yuqing, Liu Bin, Zhai Huihui. Discussion on contract risk control and related issues in public hospitals[J]. Hospital Management Forum, 2023, 40(12):44-45+43.
[4] TANG Ping, AI Hua, YI Shuping. Web-based contract management system[J]. Journal of Chongqing University (Natural Science Edition), 2005(06):26-29.
[5] Chen Li. WEB-based contract management system [J]. Economist, 2007(08):217-218.
[6] Chen Qianyi, He Jun.Analysis of Vue+Springboot+MyBatis Technology Application[J]. Computer Programming Skills and Maintenance, 2020(01):14-15+28.DOI:10.16184/j.cnki.comprg.2020.01.005.
[7] Shengsheng Chao. An analysis of MVVM model based on front-end and back-end separation under SpringBoot microservice architecture[J]. Computer Knowledge and Technology, 2021, 17(23):128-129+141.DOI:10.14004/j. cnki.ckt. 2021.2412.
[8] Xiong YP. Analysis and research on application development technology based on SpringBoot framework[J]. Computer Knowledge and Technology, 2019, 15(36):76-77.DOI:10.14004/j.cnki.ckt.2019.4290.
[9] Lv Yuchen.SpringBoot framework in web application development[J]. Science and Technology Innovation Guide, 2018, 15(08):168+173. DOI:10.16660/j.cnki.1674-098X.2018.08.168.
[10] Rong Yandong. Research on the application of Mybatis persistence layer framework[J]. Information Security and Technology, 2015, 6(12):86-88.
[11] Xu Xiaocheng. Design and implementation of web application security framework based on Shiro[J]. Computer Knowledge and Technology, 2015, 11(16):93-95.DOI:10.14004/j.cnki.ckt.2015.1199.
[12] Wang, Suan-Wen. Implementation of permission management based on SpringBoot+Shiro[J]. Computer Programming Skills and Maintenance, 2019(09):160-161+173.DOI:10.16184/j.cnki.comprg.2019.09.057.
[13] Zeng Chaoyu, Li Jinxiang.Application of Redis in Caching System[J]. Microcomputer and Applications, 2013, 32(12): 11-13. DOI:10.19358/j.issn.1674-7720.2013.12.004.
[14] Ma Yuxing.Redis database characterization[J]. Internet of Things Technology, 2015, 5(03):105-106.DOI:10. 16667/j. issn. 2095-1302.2015.03.032.