



浙江工业大学

本科毕业论文(设计)

论文题目: Design and Implementation of Forbidden
Word Recognition and Rendering with
Browser Environment

学 院: 计算机科学与技术学院
专 业: 软件工程(中外合作办学)
班 级: 2019 软件工程(中外合作办学)02
学 号: 201906150218
学生姓名: 唐晨宇
指导老师: 李小薪
提交日期: 2023 年 06 月

摘 要

违禁词检测指筛查并标记文本中的违禁词的一种自动化技术。违禁词检测是一种重要的信息安全措施,常见于媒体平台。目前,金融企业数字化转型不断升级,线上平台的用户数进一步增长,对于发布内容的风险管理需求日益增长。因此,建立企业合规性检测平台对降低企业内容发布风险至关重要。

本文设计并实现了一个基于浏览器环境的违禁词检测系统,用于在发布内容过程中进行自动化违禁词检测。该系统由文件类型解析、内容检测和浏览器文件渲染等三个模块构成:文件类型解析用于解析用户上传的文件类型,结合文件头十六进制数据与 MIME 类型进行类型检测,并通过配置文件确定与限制文件类型;内容检测用于提取文件中所需被检测的内容,基于云服务接口从输入文本中生成违禁词列表;浏览器文件渲染则将检测出的敏感文本进行定位并加上不同的颜色底纹进行高亮展示。此外,还探讨了文件违禁词识别中二进制文件的网络传输数据量较大的问题。由于二进制文件通常包含了富文本信息,文件通常较大,本文以文本内容传输取代二进制文件进行网络传输。首先对文件类型进行解析,根据不同的文件类型,采用不同的方式提取文本信息,进行网络传输。本系统能够对 PDF 文件、Office 系列文件等主流的文件类型进行文本内容提取,极大提高了大文件的违禁词检测速度。

系统基于 B/S 结构,提供了一个交互式图形界面,提升了系统的易用性。用户可以上传需要检测的文件,系统能够快速确认文件内容的合规性,为后续进行相应的人工审核与追责建立了可视化手段。

关键词: 违禁词检测,违禁词渲染,文件解析,Node.js

Abstract

Forbidden word detection refers to an automated technology that screens and marks sensitive text. Forbidden word detection is an essential information security measure commonly found on media platforms. Currently, as financial enterprises undergo digital transformation, the number of users on online platforms continues to grow, leading to an increasing need for risk management in content publishing. Therefore, establishing an enterprise compliance detection platform is crucial in mitigating content publishing risks.

This article designs and implements a browser-based forbidden word detection system for automated detection during the content publishing process. The system consists of three modules: file type parsing, content detection, and browser file rendering. File type parsing is used to analyze the file types uploaded by users, combining hexadecimal data from file headers with MIME types for type detection, and restricting file types based on a configuration file. Content detection extracts the required content to be checked from the file and generates a forbidden word list using cloud service interfaces. Browser file rendering locates the detected sensitive text and highlights it with different colored backgrounds. Additionally, the article explores the issue of large data transmission in the detection of forbidden words in binary files. Since binary files often contain rich text information and are typically large in size, this study proposes using text content transmission instead of binary files for network transmission. The file type is first parsed, and text information is extracted and transmitted over the network using different methods based on the file type. This system is capable of extracting text content from mainstream file types such as PDF files and the Office suite, significantly improving the speed of forbidden word detection in large files.

The system is based on a B/S architecture and provides an interactive graphical interface, enhancing usability. Users can upload files for detection, and the system can quickly confirm the compliance of the file's content, providing a visual means for subsequent manual review and accountability.

Keywords: Forbidden word detection, Forbidden word rendering, File parsing, Node.js

Contents

摘 要	I
Abstract	II
Contents	IV
List of Tables	V
List of Figures.....	VI
Chapter 1 Introduction.....	1
1.1 Background and Significance	1
1.2 Related Work	2
1.2.1 Content Security Review Technology	3
1.2.2 Content Review Platform.....	4
1.2.3 File Types and Recognition Techniques	5
1.3 Research Content.....	6
1.4 Outline	7
1.5 Conclusion	8
Chapter 2 Main Technical Basic.....	9
2.1 Node.....	9
2.2 Vue.....	11
2.3 TypeScript	13
2.4 Conclusion	14
Chapter 3 Content Detection.....	15
3.1 File Operation	15
3.1.1 File in Javascript	15
3.1.2 File Type Resolution	16
3.1.3 File Content Parsing.....	19
3.2 Recognition of Forbidden Words	22
3.2.1 Necessity.....	22
3.2.2 Sensitive Thesaurus	23
3.2.3 Service Selection	23
3.3 Conclusion	25
Chapter 4 Browser Rendering Technology.....	26
4.1 PDF.js	26

4.2	Rendering Scheme	31
4.3	Conclusion	33
Chapter 5	Content Review Platform.....	34
5.1	Analysis and Design	34
5.1.1	View Point.....	34
5.1.2	Requirement Analysis	34
5.1.3	System Module Design	36
5.1.4	Database Design.....	39
5.2	System Implementation	39
5.2.1	Engineering	39
5.2.2	Implementation	42
5.3	System Testing.....	47
5.4	Conclusion	48
Chapter 6	Summary and Outlook	50
6.1	Project Summary	50
6.2	Future Outlook.....	50
References	52
Acknowledgements	54
Appendix	56
附录 1	毕业设计文献综述	56
附录 2	毕业设计开题报告	56
附录 3	毕业设计外文翻译(中文译文与外文原文).....	56

List of Tables

Table 3-1 File type test	16
Table 3-2 Config of file type.....	18
Table 3-3 Result of file type resolution	19
Table 3-4 Lib of office.....	21
Table 3-5 Advertising sensitive words	23
Table 3-6 Service	24
Table 4-1 Lib of PDF.....	26
Table 5-1 Users of system	34
Table 5-2 Unit test.....	47
Table 5-3 UI test	48
Table 5-4 Function test	48
Table 5-5 Performance test.....	49

List of Figures

Figure 2-1 Memory of JavaScript	10
Figure 2-2 Vue model.....	12
Figure 3-1 File in browser	15
Figure 3-2 Flow of file type module	17
Figure 4-1 File hijacking.....	29
Figure 5-1 User module	36
Figure 5-2 Detection module	37
Figure 5-3 Audit module.....	38
Figure 5-4 Record module.....	39
Figure 5-5 System structure.....	40
Figure 5-6 Use case of content audit	41
Figure 5-7 Database of user control	42
Figure 5-8 Database of file	42
Figure 5-9 Login.....	43
Figure 5-10 Interactive page before uploading the file.....	44
Figure 5-11 Result after uploading the file.....	45
Figure 5-12 Problem list	46
Figure 5-13 Audit UI.....	46
Figure 5-14 File list.....	47

Chapter 1 Introduction

1.1 Background and Significance

In 2016, General Secretary Xi Jinping proposed a series of “Internet+” plans and called for the establishment of a good network environment at the National Conference on Cybersecurity and Informatization, emphasizing that websites should bear the “main responsibility” in online information management and enhance the sense of mission and responsibility of internet enterprises to jointly promote the sustainable and healthy development of the internet^[1]. With the continuous upgrading of modern financial enterprises’ digital transformation, the number of users on online platforms has further increased, posing challenges to many of the enterprise’s existing basic system architecture^[2]. Many banks in China have established fintech subsidiaries, while insurance and securities industries have also optimized their businesses through digital technologies, resulting in improvements in risk control and service levels^[3].

The successful integration of the internet and the financial enterprise has greatly improved the convenience and inclusiveness of financial services, but it has also brought significant risks. Given the complexity and specialized nature of financial products, ordinary consumers find it difficult to distinguish between genuine and fake products, and tend to rely on various media, especially influential ones, to obtain information^[4], but these most authoritative media outlets are often owned by financial enterprise operators.

As socially responsible entities, it is crucial to prioritize the truthfulness, reliability, and effectiveness of financial information in media and advertising campaigns while also emphasizing risk management needs. To enhance compliance in media promotion, enterprises typically introduce review mechanisms and adopt the principle of position isolation and transparency, whereby content creators and reviewers are mutually independent. However, manual review is plagued by numerous issues, such as high workload, labor costs, and inconsistent standards. To effectively control costs and fulfill regulatory requirements, a growing number of enterprises have chosen to combine machine and manual review.

Adhering to the concept of being a responsible enterprise, the author’s financial en-

enterprise has proposed the need for content compliance detection to enhance risk management in information publishing through machine auditing. Content compliance detection is mainly manifested in the recognition of forbidden words in practical operation, with the ultimate goal of ensuring that the information published by the enterprise's content publishers on public platforms complies with laws and regulations, while also alerting the publishers to any problems in the content and enabling them to make timely corrections and publish content on schedule.

Although there are many related research contents available online, including multiple technological approaches, including automated algorithms, deep learning, and natural language processing, there is still a significant gap in the actual project implementation. The fundamental reason is that different content platforms face different types and scales of content, and their specific implementation methods should be tailored to the specific business processes needed.

The significance of the project research is to effectively reduce the compliance risk of content published on the platform and prevent the occurrence of illegal events caused by improper content release. The research of forbidden word detection and rendering system is to enable enterprises to strengthen the risk management of enterprise media content in a low-cost way, and is an effective means to reduce the risk of enterprise content release. It also help enterprises manage risks, reduce false information, protect the legitimate rights and interests of users, alleviate the pressures faced by risk control departments, and enhance the enterprise's sense of mission and responsibility. All of these help to build the sustained and healthy development of the Internet environment.

1.2 Related Work

Numerous studies have focused on content security with artificial intelligence technology. Some studies have used natural language processing to analysis the content. Other researches were on the practical application, which use the technology including natural language processing falls to the ground. In this section, status of content review platform are stated, and the development of natural language processing in content review is also briefly introduced. Some other technology such as file type recognition is also introduced in this section, which is an expansion module help to check the file type.

1.2.1 Content Security Review Technology

In the content security review technology at home and abroad, web page information evaluation mainly applies four filtering technologies, namely, filtering based on Internet content grading platform (PICS), database filtering (IP library, URL library), keyword filtering and intelligent content understanding filtering^[5]. In this paper, keyword filtering and intelligent content understanding filtering are mainly used for content recognition, which mainly relies on the intelligent recognition technology of natural language processing technology NLP, deep learning and other technologies to achieve filtering^[6]. Through the content analysis and understanding to generate a document model to identify forbidden content.

1.2.1.1 NLP

NLP (Natural Language Processing)^[7] is a branch of artificial intelligence which designed to enable computers to understand, interpret and generate human language. Natural language processing is a cross-discipline of linguistics, computer science and artificial intelligence, which studies the interaction between computer and human language, especially how to program computer to process and analyze a large amount of natural language data. The goal is to generate a model that can “understand” the content of the document, and this technology can accurately extract the information and opinions contained in the document, and classify and organize the document itself. This means that the computer must be able to recognize the grammatical and semantic structures in the language, analyze the words, phrases and sentences in the text, and correctly understand their meaning.

When it comes to content security, NLP technology can be used in the following aspects:

1. Sensitive word filtering: NLP technology can be used to identify and filter sensitive words from text. It can recognize these words through rule-based methods or machine learning algorithms. For example, the content audit system uses NLP technology to detect keywords such as violence, pornography, politics, terrorism and so on;
2. Text classification: NLP technology can be used to classify text into different categories. For example, the content audit system can divide the text into normal,

illegal, and questionable categories, so that people can have a clear understanding of the audit objectives and make it easier to review.

3. Semantic analysis: NLP technology can be used to identify entities and concepts in text and understand the relationship between them. Because sometimes simple sensitive words do not violate the rules, only through semantics can we really find out the sensitive parts of the text.

Of course, rebuilding a content security audit model is not the focus of this article, but it is a necessary process to understand NLP. In a non-professional artificial intelligence research and development enterprise, we prefer to purchase related services for development rather than build our own natural language processing model from scratch. So this paper focuses on the applicability of these models in actual projects. Of course, we should also retain the possibility that it can be replaced.

1.2.1.2 Content Review Service

Aliyun content Security Service^[8,9] is an AI technology-driven content review service, mainly used for content security on Internet platforms. Content security products provide multimedia content risk detection capabilities such as pictures, video, voice, text to help users find risky content or elements such as pornography, violence and politics, which can greatly reduce manual audit costs and improve content quality, improve platform order and user experience. Its main services are: (1) text security detection; (2) picture security detection; (3) video security detection; (4) voice security detection; (5) custom security policy. The service classifies the results of the detected content and returns the sensitive content in the indicated content. Aliyun content security service has been widely used in many industries to help users achieve fast, accurate and intelligent content review, improve the content security level of the platform, reduce the compliance risk of the platform, and protect the rights and interests and security of users.

Considering the development cost and the perfection of existing service, we consider using cloud security service as an important tool for content audit in this system.

1.2.2 Content Review Platform

Content moderation mechanisms are commonly found on UGC platforms, such as Mango TV, which has its own proprietary content safety technology system. However, not all enterprises develop their own content safety review systems. To reduce review

costs, improve review quality and efficiency, many third-party content review agencies have entered the content moderation outsourcing service enterprise, mainly including traditional mainstream media, internet cloud service giants, AI technology enterprises, and specialized content moderation outsourcing enterprises^[10].

However, whether it is a self-developed platform or a third-party service, the audit process of “machine audit + manual audit” can generally be summarized as follow steps^[10]:

1. Machine marking classification: Make use of the content analysis ability of AI, tag the content, store it in different repositories, and enter the preliminary examination;
2. Machine preliminary examination: AI will match the contents of the library to the relevant check libraries, such as “sensitive word library” and “sensitive picture library”, and mark the possible problems, so as to provide reference for the follow-up review;
3. Manual review: Judge the tag and classification label of the first trial, and read through the work in a comprehensive way, and give a judgment on whether the content is in compliance.
4. Manual third trial: It is mainly to confirm and deal with the results of the review, and to be responsible for making decisions on the contents of the review;
5. Manual sampling inspection: The content works that have been examined three times need to be sampled manually at last.

The content review process has been very perfect. This paper also refers to the above steps to complete the research of this topic.

1.2.3 File Types and Recognition Techniques

In order to deal with the problem of attachments in formal notification, we need file recognition technology to determine the file type, so as to distinguish between different types of documents.

In Windows system, file manager identifies and determines the file type according to the suffix name of the file, that is, the last “.” and the back part. However, the file name is completely transparent to the user and can be modified at will, and it is easy

to encounter situations where the file name is tampered with or damaged, the file suffix name is missing or cannot be automatically recognized by the system, so relying on the extension name alone is not reliable^[11].

In addition to the suffix name, there is another way of file recognition called feature identification, which refers to a varying lengths hexadecimal string of character data stored in the file header to distinguish different file types.

In the browser environment, when a user uploads a file on a web page, the browser will use a series of file type identification algorithms, for example, the browser will read the byte stream of the uploaded file and parse the file feature identification information to determine the file type, and deal with it accordingly.

MIME type (Multipurpose Internet Mail Extensions type)^[12] is a standardized description of file types, browsers can use MIME types to determine file types. There are usually three ways to detect MIME types^[13]: (1) check the content-type carried in the request^[14]; (2) read the characteristic identification of the file; (3) judge according to the suffix name.

At present, there are indeed some developed MIME type recognition packages, but after research, it is found that it does not support the MS Office format we need^[15], which uses the open file format of Office Open XML (OOXML)^[16]. Its content identification is consistent with the compression type, so additional development is needed to complete this work.

1.3 Research Content

In the system designed in this article, the goal is to design a content audit system that meets the requirements of the enterprise, ensuring that content is audited as quickly and efficiently as possible through a fast and effective feedback mechanism. The expected research objectives of this topic are as follows:

1. In this study, we need to identify the files uploaded by users, and we need to develop a file recognition module in the browser environment. Multiple types of files imported through a variety of ways can be identify by this module, in which MS Office and PDF file formats should be paid more attention to;
2. Solve the problem of large file transfer in the network, the amount of data can be

reduced by extracting the contents of the file. So we need to design a module to parse the file given by the user, make it easier to transfer in the network;

3. Use Node to design the request interface for interaction with Aliyun, make it conform to Aliyun's structure invocation specification, and analyze the available return results;
4. Complete the implementation of the forbidden word recognition and rendering system in the browser environment. Use the module we completed above, users only need to input the content and files to be published, and the system can automatically extract the required data and upload it to the cloud service, and feedback the corresponding results in time;
5. Strive for multi-end adaptation, so that the system can meet a variety of needs of content release, improve the user experience.

1.4 Outline

This paper is a research and design of recognition and rendering of forbidden words in browser Environment. This paper consists of six parts, namely introduction, main technical basic, forbidden words detection, rendering technology, forbidden word recognition system and summary. The main content of each chapter is summarized as follows:

In *chapter one*, the background of the content security is introduced, and the importance of establishing a compliant content distribution platform is illustrated. Then, related research status is reviewed. In addition, the content of research is also presented.

In *chapter two*, the main technical involved in this study will be introduced, including the node.js, typescript and vue.

In *chapter three*, talks about how to realize the recognition of forbidden words, including the selection of cloud service, the design and implementation of interface and data structure. In addition, the identification module of file type and the parsing module of file is introduced, it include the necessity of module design, the exploration of the implementation method and the actual effect.

In *chapter four*, Talked about the browser's file rendering technology, how to display mainstream files in the browser. This paper mainly introduces the existence of files

in the browser, the application of PDF.js as an open source project, its existing problems and solutions, and finally introduces the scheme of coloring on forbidden words.

In *chapter five*, a forbidden word recognition and rendering system is designed and implemented. The requirements and structure of the system are analyzed and designed, the development techniques are briefly introduced, and the modules and functions of the system are demonstrated. In addition, the testing methods is also introduced to ensure that the system is working properly.

In *chapter six*, a brief summary and outlook of this project are stated.

1.5 Conclusion

Today, with the rapid development of the Internet, the number of users of the online platform is also increasing, the risk management needs of enterprises are also constantly improving, and the content compliance testing system has a lot of market space and social demand. Starting from the enterprise “compliance content detection”, this paper studies the detection and rendering of forbidden words in the browser environment, in order to promote the construction of enterprise content compliance. This business process proposed in this article is equally applicable to other content publishing platforms, such as government websites and school notice boards.

Chapter 2 Main Technical Basic

2.1 Node

Node.js^[17] is a JavaScript running environment based on Chrome V8 engine, which has advantages in development efficiency and performance, and is widely used in the field of Web development. Node.js provides some special modules, which make it easy for developers to carry out I/O operation, process management, timer, database access and other operations, making it more convenient and quick to use JavaScript for server-side development.

In addition to serving as a server, it is more used to quickly build extensible Web applications. For example, npm (Node Package Manager)^[18] package management system is now the most important tool in front-end work, and it is also the default package management module of Node.js. With the widespread use of Node.js in Web application development, the Node.js community is also very active. Npm provides a large number of third-party modules and tools to help developers develop and deploy applications more quickly. More and more developers download npm packages to the local software environment, and develop software on this basis^[19].

a) EventLoop

Node.js is an event-driven, non-blocking I/O model that allows developers to handle multiple connections simultaneously. This means that Node.js applications can handle a large number of requests without blocking the main event loop.

Javascript is single-threaded execution, this means it can only handle one task at a time. But eventloop make it work like Multithreading. In order to understand the event-loop, we need to know some knowledge points: (1) Asynchronous events are divided into macrotasks and microtasks, and micro task queues take precedence over macro task queues; (2) Macro tasks are executed individually, while violating tasks are carried out in teams.

Figure 2-1 shows how memory is allocated in Javascript. When the event loop is executed, the main process code is treated as a macrotask and pushed into the call stack, and whenever an asynchronous task is encountered, JavaScript registers a callback and

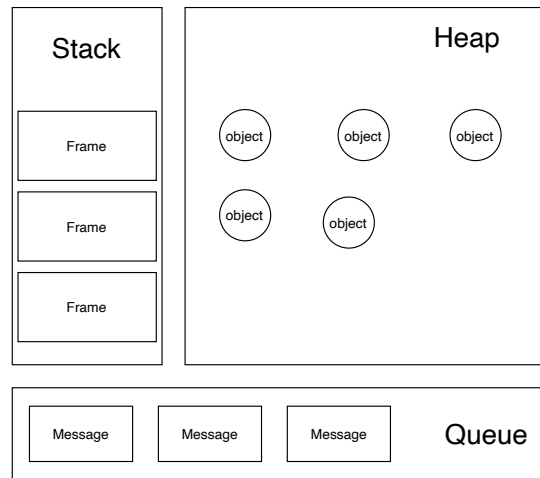


Figure 2-1 : Memory of JavaScript

pops the task from the call stack. The task is pushed into the corresponding task queue, which contains a macrotask queue and a plurality of micro task queues. When the call stack is empty, the event loop fetches the first task in the task queue, and the macrotask alternates with the microtask queue. If there are no tasks in the task queue, the event loop waits until a new task is joined.

In order to complete a highly available system, understanding the mechanism of the event loop improve the efficiency of the program execution and enable the program to return in the right way.

b) I/O

IO means Input/Output. In the computer field, IO is usually used to describe the process of interaction between the computer system and the external environment (hardware equipment, network, etc.). This is one of the most different between node and browser. The IO module in Node.js is used to handle file system and network communications and mainly includes the following sub-modules: fs, path, treame, os, net, http, gram, zlib. In this system, we pay more attention to fs and http modules.

“fs” meas File System, this module provides an API for working with the file system. The “fs” module allows developers to read, write, and manipulate files and directories on the server. The module provides both synchronous and asynchronous methods for working with the file system, allowing developers to choose the approach that best suits their needs. At the same time, it can also cooperate with the “path” module to create

static resources on the server for file storage, which is also one of the important contents of this project.

“http” module provides the ability to create HTTP servers and clients. Through this module, developer can easily handle HTTP requests and responses, and build Web applications such as RESTful API. This not only enables programs to communicate with certain standards on the Internet, but also improve the performance of Node as a back-end service, and makes it possible to provide micro-services and build middleware platforms.

The main benefits eventloop bring is its ability to handle I/O operations in a non-blocking way, which makes it ideal for building scalable and high-performance applications. Combined with it, it can take advantage of the file processing ability of javascript. The system will use node to work as a serve to handle request from front-end and interact with the database.

2.2 Vue

Vue.js is a popular JavaScript front-end framework for building interactive Web interfaces. It adopts a progressive development approach which allows developers to gradually apply new features or technologies to existing code.

a) Data-Driven

Data-driven is one of the core concepts of Vue.js framework, and it is also an important paradigm in modern Web application development. Its idea is to separate the UI from the underlying data and drive the update of the interface through the changes of the data, thus making the development more efficient, flexible and maintainable. Vue use MVVM to implement data-driven. MVVM means Model-View-ViewModel. Model layer is responsible for handling business logic and interacting with the server, View layer responsible for transforming the data model into UI display, which can be simply understood as a HTML page, view-model layer is used to connect Model and View, is the communication bridge between Model and View. The Figure 2-2 shows the model the Vue used. Different with React, the other front-end framework use JSX, Vue data-driven implementation mainly depends on the following parts: (1) Template syntax is a display mode like HTML, these templates can contain a variety of instructions and expressions to associate data with the interface through data binding, this enables data to render correctly on the interface. (2) Responsive systems hijacks the data object and automatically triggers the re-rendering of the interface when the data changes. This process is based on the getter

and setter functions of JavaScript and realizes responsive refresh by detecting changes in data. So when user change value in HTML page, the code will watch this change and trigger the corresponding function. It also will render the page after the value used by template changed, these two processes are called bidirectional data binding. In addition to bidirectional data binding, vue also provides rendering functions that allow developers to write JavaScript code directly to generate virtual DOM node trees. By doing so, you can control the rendering logic of the interface more flexibly and achieve more complex interactions.

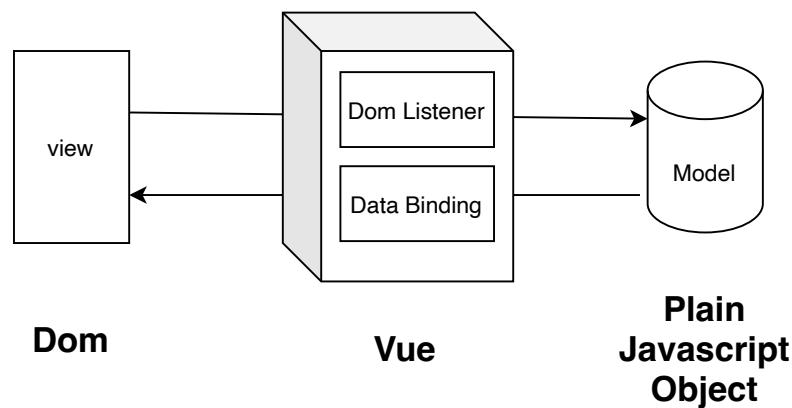


Figure 2-2 : Vue model

b) Components

Vue improves the reusability of data by means of components. Vue collects functional pages and combines html, js, css into a component to make it reusable and extensible. Each Vue component has its own independent scope, which means that the state and behavior of the components are isolated from each other and do not affect the rest of the page. At the same time, Vue also provides some special syntax and built-in functions to enable components to build into projects and enable them to communicate with each other.

A single page application (SPA) is a Web application based on front-end routing that does not need to reload the entire page, but instead switches content dynamically through JavaScript. Vue Router is the official router of Vue, which can help to build SPA applications. Vue Router matches components by listening for changes in URL addresses and supports advanced functions such as passing parameters to components and dynam-

ically modifying routing rules. Vue provides two common ways of passing data between parent and child components: props and events. Props allows the parent component to pass data to the child component, and the child component receives the data through the props and can display or modify it. The event mechanism allows the child component to pass messages to the parent component, which triggers events that the parent component listens to through the \$emit method and passes the data to the parent component. This "publisher-subscriber" mode is a common way of communication between Vue components, and can achieve efficient and reliable data transmission. Between non-parent and child components, Vue provides a state management library, Pinia. In Pinia, we can define the global state of the application by creating a store class, and then access and modify the state in a manner similar to the computed property and the methods method in the Vue component. This approach has the characteristics of simple, clear, flexible, extensible and type-safe, which can help developers manage the state of the application more efficiently, thus improving the quality and maintainability of the code.

2.3 TypeScript

JavaScript is a relatively poor language for developing and maintaining large applications, and there are some difficulties in using JavaScript in a complex code base: language abstractions that lack robustness, such as static types, classes, and interfaces, which can hinder programmers' productivity and undermine tool support^[20]. TypeScript is an extension of JavaScript designed to address this flaw, developed by Microsoft and a superset of JavaScript, so every JavaScript program is a TypeScript program^[21,22]. TypeScript adds optional static types and some new features, such as classes, interfaces and so on. So that developers can do type checking and better maintainability when writing code. TypeScript is designed to provide lightweight help to programmers, so the module system and type system are flexible and easy to use. At the same time, it can also adapt to existing JavaScript projects very well, without the need for overall project rewriting^[22].

TypeScript enforces that variables must be declared before they are used, which avoids the problems that may be caused by implicit declaration of variables in JavaScript. In addition, TypeScript adds support for object-oriented programming through the definition of interfaces and classes, making the code easier to read and maintain. Developers can create their own interfaces and classes, and use these interfaces or classes for type determination in the code, so as to improve the readability and maintainability of the code.

TypeScript performs type checking at compile time, and if a type error is found, it will be prompted at compile time to avoid type errors that may occur at run time. This feature can not only improve the standardization of the code, but also improve the development experience of developers. In addition, TypeScript provides many useful tools and features, such as Enum, Namespace, and Decorator, to further enhance the functionality and flexibility of TypeScript.

2.4 Conclusion

Node is one of the most important JavaScript running environment in the field of browser environment. It enables programmers to build a complete program using only Javascript. In this chapter, the important theoretical knowledge and useful tool related to Node are stated, which is an important basis for the follow-up work of this study. At the same time, the concept of Vue and TypeScript is introduced, the former will be used to build the project and the later will be used to make this project more easy to maintain.

Chapter 3 Content Detection

3.1 File Operation

3.1.1 File in Javascript

From a general perspective, it is common practice to process files on the backend, using languages such as Java that provide more comprehensive file handling capabilities and higher processing performance. The frontend's primary responsibilities include page rendering and logical interactions. However, with increased backend pressure and continuous frontend technological advancements, front-end systems have started taking on more data-processing tasks.

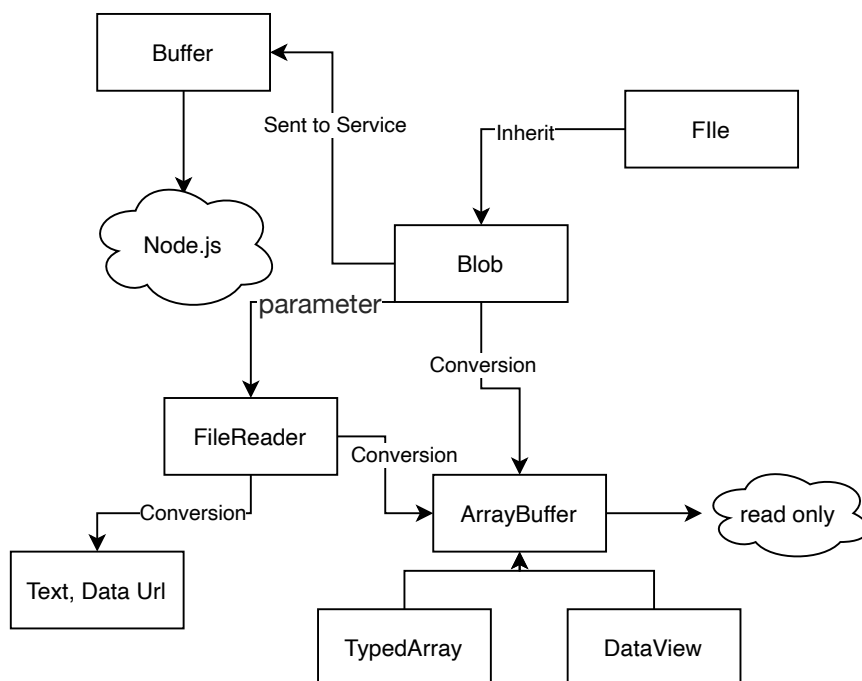


Figure 3-1 : File in browser

Thanks to HTML5, browsers support more methods of manipulating binary data. Before HTML5, we usually use streams for file transfer. Streams has good performance in network transmission. Figure 3-2 shows the all type in Javascript the file can be. In the browser environment, there are two main sources of files: local uploads from users

and files from remote servers. Blob is a object of HTML5 that is specifically used to support binary file operations. The object can increase the file transfer rate through data slicing. Blob provides identifiers for files through blob URL, which is different from data URL based on base64 encoding, which contains the complete data content. File inherits from Blob and provides attribute information such as file names. ArrayBuffer, by contrast, represents the complete binary readable data of the file, which can be written using TypedArray and DataView. In addition, files can be converted between multiple formats through the FileReader object.

An in-depth understanding of the transfer of files in browsers and networks will help to maintain the consistency of data and the unity of interfaces in the construction of the system. This in turn improves the maintainability and extensibility of the system.

3.1.2 File Type Resolution

3.1.2.1 Necessity

File name	Actual type	Result
testpptx2.jpg	pptx	image/jpeg
testjpg.pptx	jpg	application/vnd.openxmlformats-officedocument.presentationml.presentation
Testdoc.doc	docx	application/msword

Table 3-1 : File type test

It is important to identify file types in the system because different file types need to be handled differently. In this system, the identification of file types mainly serve the following three functions:

1. Limit the types of files uploaded by users to ensure file security.
2. Ensure that the system calls the correct function to operate on the file.
3. Protect servers and save network resources.

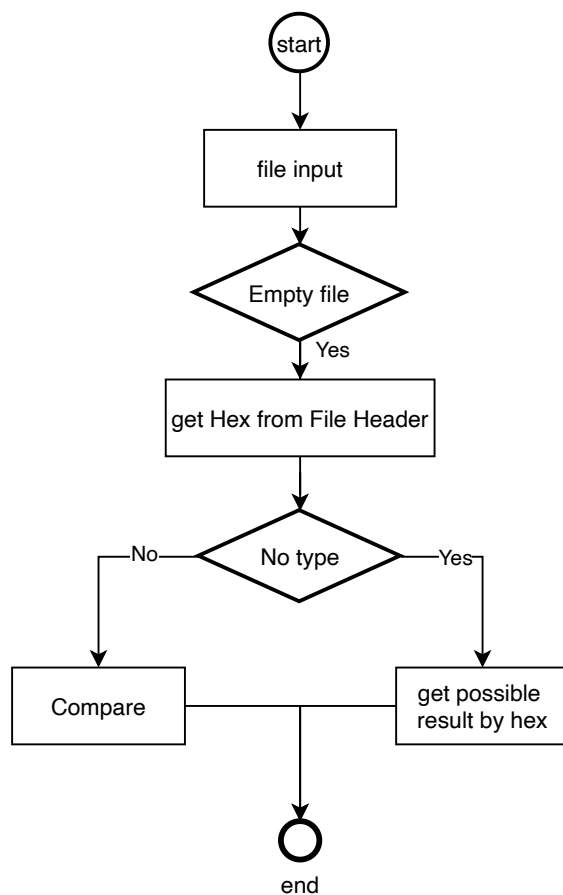


Figure 3-2 : Flow of file type module

In the section 1.2.3, we describe the existing applications of file recognition technology, there are mainly three common recognition schemes: file suffix, MIME type and magic number. The file suffix is the technology used in the existing system of the enterprise. In view of the transparency of the suffix name, only relying on the suffix name to identify the file type is not conducive to the stable operation of the system. Aiming at the recognition mode of MIME type, this paper uses the file upload component to identify the type in the browser environment, and the recognition results are shown in the Table 3-1. In terms of the results, the browser still mainly depends on the file suffix to identify the file type, so we decided to introduce Magic Number to take over the file recognition.

Magic Number mainly determines the type of the identified file by extracting the header of the binary file. The existing Magic Number file recognition module mainly supports the recognition of picture, video and other file types, but does not support MS

Office very well. At the same time, we also studied the format of domestic software WPS to increase the recognition support for this file type. In addition, it can also support the company's internal file format, can also be identified by the specified Magic Number. After considering the above factors, this paper thinks that the general module of file recognition has considerable development value.

3.1.2.2 Implementation

File type module including file type verification and type finding. The flow of module is show in Figure 3-2 . The module need a file and a optional config as input, the module will anysis the file, extract the MIME type and header of the file. If the file type exists, verify that MIME type matches Magic Number through the file header. If it does not exist, guess the possible type of the file through Magic Number. The module will return an array of file suffix string.

File type	MIME type	Hex
pdf	application/pdf	25 50 44 46
zip	application/zip	50 4b 03 04
wps	application/vnd.ms-works	0e 57 4b 53
pptx	application/vnd.openxmlformats-officedocument.presentationml.presentation	50 4b 03 04
docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document	50 4b 03 04
doc	application/msword	d0 cf 11 e0
		a1 b1 1a e1
ppt	application/vnd.ms-powerpoint	d0 cf 11 e0
		a1 b1 1a e1

Table 3-2 : Config of file type

The module device into two parts. One is config file, which can custom definition by developers, which is the first step of module. the module create an interface to de-

fine the struct of file type which contains hex, MIME type and suffix name. This struct contain the most of attribute we attention. The Table 3-2 shows part of config, developer can configure this file to implement restrictions on the acceptance of files, which is similar to the whitelist of files. The other part is execution function. Firstly, we use file type introduced in Section 3.1.1 to get 10 Byte data. This 10 Byte contains the full information of file type. Then we converse it into hex string and compare with Magic number we defined in the config file. After get the all Parameters we need, we firstly execute type verification, if file's MIME type match its Magic number, the type of file can be confirmed. If that's not the case, all possible types of the file will be returned, and the caller will need to handle this operation on his own, determine that the file does not comply with the regulations, or try to parse the file through the possible types.

File name	Actual type	MIME type	Result
test. wps	wps	application/vnd.ms-works	.wps
test.pdf	jpg	application/pdf	.jpg
test	pdf	plain/txt	.pdf
Test.jpg	pptx	image/jpeg	.zip .pptx .docx
test	pptx	image/jpeg	.zip .pptx .docx

Table 3-3 : Result of file type resolution

The Table 3-3 show the result of module. Obviously, this module can achieve the desired function, correctly identify the file type, and help the system to limit the file type.

3.1.3 File Content Parsing

3.1.3.1 Necessity

Prior to content auditing, it is necessary for us to clarify the required data types for the service. In section 3.2.3, we examined numerous content security services offered by

various enterprises. To ensure seamless replacement of services, employing the forbidden word recognition presented in the text is a more fundamental and universal solution. Another advantage of using text transmission instead of file transmission is reduced data transfer in the network, which decreases system wait time and enhances user experience.

To quantify the difference in latency between file and text transmission, this paper extracts all words from a demonstration presentation file into text. Without styles, the powerpoint file size was 82KB, whereas the text size was only 25KB. When additional media such as styled images are included, the file size difference becomes much more evident. In the same network environment, it takes more time to perform network delay and detection. Assuming the download of a 10Mb presentation template in a 4 Mb/s network environment, the amount of extracted text is only 8Kb. According to the calculation formula and substituting the result, the following conclusion can be drawn:

$$time = data \div rate$$

$$time = 10Mb \div 4Mb/s = 2.5s$$

$$time = 8Kb \div 4Mb/s = 2ms$$

It is apparent that using text detection rather than file detection can significantly improve efficiency and enhance user experience. Moreover, unified detection forms save interface development quantity, effectively reducing project time cost while improving efficiency.

3.1.3.2 Implementation

The number of functions in File content parsing module depends on the number of file types that need to be converted. In this system, we mainly operate on the five file formats of pdf,docx,doc,ppt,pptx. In each function, we will put a file as input and an array contains key-value pair of page-content as output. For pdf,Ms Office2007,Ms Office2003 (wps), we propose three different schemes respectively.

For pdf, PDF.js is a useful module, which will be introduced in Section 4.1. It provide the methods rendering pdf. After studying the use process of the pdf module, it is found that it contains a function to obtain the text, which can obtain the text when canvas rendering, and the pdf text can be easily obtained by using this function.

After pdf can complete the text acquisition, this paper first proposes to convert the other file types into pdf through the existing services, and then parse through the pdf

parsing module, but this brings great network overhead, and the time delay formula is as follows. In the experiment, this overhead is more than 10 seconds, according to the user feedback principle, this is an unacceptable waiting time, so this scheme is chosen as fallback plan.

$$Time = OfficeNetTransTime + PDFNetTransTime + ConversionTime$$

For the documents of Ms Office 2007, we also found many high-quality third-party libraries during the technology selection. Table 3-4 shows the libraries we have studied. Most of these powerful libraries rely on the Node environment, and complex types of binaries are difficult to parse in browsers. Although there are many tripartite libraries to deal with Office files, and there are also many powerful processing libraries, we only need simple text extraction function in the project, and we hope to control the project volume through using general basic library. Section 1.2.3 introduces that the Ms Office 2007 file type is a compressed package file composed of XML, so using Jszip decompression to operate in the browser can well solve the problem of complex Office file types. Through the analysis of the structure of the decompressed file, the content of the file can be extracted by regular expression, so that we can effectively extract the content needed in the Ms Office document.

lib	environment	feature
Office.js	Node.js	The official library, which can read the content, format, style and metadata of Office documents, can only be used in Microsoft Office environment.
Mammoth.js	Both	Provides a simple interface to implement the operation of docx documents, can be converted to HTML and get the content, but does not support ppt.
JSZip	Browser	Provide decompression solution and functional basis under the browser.
adm-zip	Node.js	Provide decompression solution under Node

Table 3-4 : Lib of office

In versions prior to Ms Office 2003, that is, files suffixed with doc and ppt. These file structure does not use the compressed package format. In view of the fact that these

two types of files account for less in the system, the former can basically obtain the contents of the files through UTF-16 decoding, while the latter adopts the process of converting them to pdf format and then carrying on the follow-up process. In addition, the format of domestic software wps is the same as that of doc, so the same solution can be adopted.

After implementation and testing, the file can basically get the corresponding content output, effectively achieving the optimization results proposed in Section 3.1.3.1.

3.2 Recognition of Forbidden Words

3.2.1 Necessity

In today's Internet era, content security management has become one of the important tasks to ensure user experience and maintain social order. Forbidden word detection for text content is one of the key technologies, which aims to identify and filter potential illegal, inappropriate or sensitive text information to ensure the health and safety of the network environment. Section 1.2.1.2 has described how some content security detection services perform forbidden word detection. Compared with the local forbidden thesaurus, the use of content security services to detect forbidden words has the following necessity and advantages:

(1) Rich thesaurus and real-time updates: Cloud's content security service maintains a large and rich forbidden thesaurus, covering a wide range of fields and languages, including common sensitive words, politically sensitive words, vulgar words, and so on. The service regularly updates the thesaurus to ensure the timely identification of new forbidden words, thereby improving the accuracy and timeliness of detection.

(2) Multi-dimensional detection capability: Cloud's content security service is not only limited to simple text matching, but also covers multi-dimensional detection capabilities. For example, it can identify contextual information and understand the meaning and semantic relations of words by analyzing the context around the text, so as to more accurately judge whether it belongs to forbidden content. In addition, it can also use pinyin and other technologies to detect forbidden words expressed in deformation, phonetic proximity, homophone and other ways, so as to improve the coverage and accuracy of detection.

(3) Scalable service capabilities: Cloud's content security service is based on the cloud platform and is highly scalable and flexible. It can handle large-scale requests and

return test results in a short time. It is suitable for high concurrency scenarios and large-scale text detection requirements. In addition, the service also provides a wealth of interfaces and tools to facilitate developers to integrate and customize to meet personalized business needs.

Based on the above advantages, this paper chooses the content security cloud service to develop the project, which reduces the development cost while improving the availability, thus reducing the cost and increasing the efficiency.

3.2.2 Sensitive Thesaurus

When designing and implementing a forbidden word detection system under the basic background of the Advertising Law, the collection and management of forbidden words need to be regulated and controlled in accordance with the laws and regulations to ensure the legitimacy and compliance of the relevant content. The table 3-5^[23] shows the part of forbidden words in Advertising Law. These forbidden words will be entered into the cloud service as an additional thesaurus to provide the project with the ability to identify forbidden words in subdivided areas. Of course, according to the needs of the business needs, we can also add other vocabulary for detecting.

most related	the best, the most beloved, the largest, the greatest, the highest ..
first related	ranking first, unique, first brand, NO.1, TOP1, unique ..
country related	the first in the country, First choice ..
time related	pre-order as soon as possible ..
grade related	national level, AAA, advanced ..
brand related	gold medal, famous brand, leading brand ..
authority related	time-honored brands, Chinese well-known trademarks ..

Table 3-5 : Advertising sensitive words

3.2.3 Service Selection

As we mentioned in Section 1.2.2, because of the high cost of developing a model for content security review, the emergence of third-party services provides enterprises with better choices. Through the survey of mainstream services, our most common ser-

vices come from Tencent Cloud and Aliyun. The features provided by both are shown in the table 3-6 .

Both of them can basically meet the service support for the project, but because Aliyun provides more services and the price is relatively low, and the resources purchased by Aliyun can be used for any series of content security services, Tencent Cloud has to pay separately for text, pictures, etc.

In text detection, Tencent Cloud provides keywords strings and tags as return results, and gives its confidence and detection suggestions, which are suitable for simple result display. Aliyun provides two versions of the detection service. The basic version returns a text after the banned words are blocked by *, the location and text of the forbidden words can be obtained by comparing with the original text. The enhanced version provides more subdivided tags and strings of forbidden words.

In this project, in order to achieve the purpose of text rendering, Aliyun can more accurately determine the location of the text, so using Aliyun for content security detection has become a better choice. In the specific selection of service content, this project puts forward various schemes:

1. Aliyun's file recognition is used to identify the files that need content recognition directly, but the file size it supports is only 5MB, so it puts forward the operation mode of converting the file to PDF and uploading it in pages for identification. However, because the development of PDF paging is too tedious and takes too long, it is not adopted finally.
2. The text in the file is extracted and detected by the text recognition service. The advantages of doing so have been described in Section 3.1.3.1, and after possibility

Service	Text	Picture	Video	File	Price
Tencent	yes	yes	yes	no	16 yuan / 10000 articles
Alibaba	yes	yes	yes	yes	15.5 yuan / 10000 articles

Table 3-6 : Service

verification, this method is more feasible and is the final solution adopted by this project.

However, since Aliyun needs enterprise certification to activate relevant services, this article can only be developed using Tencent Cloud. Thanks to the content security detection through text recognition and the transferability of the interface design, the two sets of services can be converted only by certain processing for the different returns. This also means that when there is a better service choice in the future, under the existing interactive interface design, the service migration can be carried out well. However, according to the return content of different services provided by different services, we need to adjust accordingly according to the determined text as input and keyword-page key-value pair as output. In the design, this project takes the text string that needs to be identified as input, processes the data returned by the cloud service, and returns an array of objects containing forbidden type tags, forbidden words, and page numbers. The data structure is designed as One-tier object array which can easily traverse all the data needed, which reduces the complexity of multi-tier data access.

In the development, the two types of services basically provide the corresponding API. For service security, access restrictions and localization of service keys are of great significance to improve code security and reduce the risk of source code leakage. After implementation and testing, the file can basically get the corresponding output of contraband words, which can meet the future development functions.

3.3 Conclusion

This chapter introduces how to realize the recognition of forbidden words, including the selection and limitation of cloud services, the design and implementation of interfaces and data structures. At the same time, it also introduces the pre-working document processing for the identification of forbidden words, mainly introduces the identification module of file type and the parsing module of file, from the necessity of module design, the exploration of the implementation method and the actual effect to describe.

Chapter 4 Browser Rendering Technology

4.1 PDF.js

lib	advantages	disadvantages
pdfjs	PDF rendering on the web page, providing an existing pdf presenter	Modifications to PDF are not supported
js-pdf	Generate a new pdf document, with a powerful function choices.	Relatively complex
pdf-lab	Used for the creation and operation of PDF, with strong maneuverability	No strong rendering support
react-pdf	Used to generate PDF and support JSX syntax	Limited function

Table 4-1 : Lib of PDF

Regarding PDF, we conducted research on existing PDF libraries, including pdfjs, pdf-lab, js-pdf, and react-pdf. Table ?? presents the advantages and disadvantages of these PDF libraries. Since our project requires frontend rendering and display of PDF files rather than generating a new PDF, pdfjs is the better choice. Moreover, pdfjs has already been extensively used in existing enterprise projects, so continuing to use this library can improve familiarity and understanding, reduce maintenance costs, and minimize code compatibility issues arising from differences between different libraries. PDF.js provides a comprehensive built-in PDF browser called viewer.html, which can effectively embed the PDF viewer into the project using an iframe interface, as applied in existing enterprise projects.

PDF.js displays PDF content on the page using two main layers: the canvasLayer and the textLayer. The canvasLayer is responsible for rendering the PDF content on the page, while the textLayer contains the text content, although it is transparent and not visible to the user. The textLayer is typically used to support text copying and selection in the PDF.

The canvas layer requires drawing operations to update and modify the content,

while the textLayer is directly displayed on the DOM, allowing the text content to be applied within specific areas of the page. The textLayer breaks down the text into spans displayed in the DOM. For example, the text “123Test” in this layer may be represented as two spans: “123” and “Test.” This significantly increases the complexity of text rendering. After in-depth research on the textLayer, we can extract and process the text content within it for application in Section 4.2.

Understanding and studying PDF.js has provided insights into the different roles and characteristics of the canvas layer and textLayer in page rendering. This understanding is crucial for utilizing PDF.js effectively and processing and applying text content. In our research, we will focus on analyzing and utilizing the textLayer to achieve efficient handling and utilization of PDF text.

In this project, we have identified four issues related to using pdfjs: how to use a file stream instead of a file path as an input parameter for PDF.js, highlighting forbidden words in PDFs, the issue of lazy loading in pdfjs causing page refresh and information loss, and cross-page interaction problems.

1) File Hijacking

PDF.js is a powerful tool used to retrieve PDF data from files and render them. In typical scenarios, it loads files by accepting a parameter containing the file URL, which is passed as payload to the src attribute of an iframe. This parameter represents the file path. After obtaining the parameter, PDF.js uses XMLHttpRequest or Fetch API to retrieve the actual file data from the server. Due to browser security restrictions, PDF.js cannot directly read and load files from a user’s system. To overcome this limitation in existing enterprise projects, a common approach is to upload files on a remote server and use the file path as input for PDF preview. Unfortunately, this method places a significant burden and overhead on the network and backend. Additionally, storing and managing these temporary files consumes server resources. To address these challenges, our research focused on using file streams as a file source for PDF.js and proposed a novel solution aimed at improving the performance and user experience of PDF.js.

To tackle this issue, our study analyzed PDF.js and identified the following steps in the file rendering process:

(1) Obtain the current page’s URL: Use the JavaScript window.location.href property to retrieve the complete URL of the current page.

(2) Parse the URL: Create a URL object and pass the current page's URL as a parameter to it. The URL constructor can be used to parse the URL.

(3) Retrieve query parameters: Use the URL object's `searchParams` property to obtain the query parameters from the URL. The query parameters are the part of the URL after the question mark. The `get()` method can be used to retrieve specific query parameter values, such as the value of the "file" query parameter.

(4) Load the PDF file using the file parameter value: Once the value of the file parameter is obtained, it can be used to load the corresponding PDF file and initiate the rendering process.

(5) Retrieve the actual file data from the server: When rendering remote files with PDF.js, the tool dynamically downloads and retrieves the actual file data from the remote server as needed. PDF.js uses a chunked loading and rendering approach, meaning it only loads and renders specific parts of the file when required. Specifically, when the renderer needs to access a certain part of the file, PDF.js initiates a network request to fetch the corresponding data chunk from the server.

By leveraging the file parameter provided by PDF.js, we can directly specify the file URL when loading the PDF file. This eliminates the need to retrieve the actual file data from the server and allows us to obtain the data directly from the browser cache. In our workflow, users can upload their local PDF files using a file upload component and pass the temporary URL to PDF.js for rendering. To facilitate cross-page communication, we utilize the window object to mount the actual rendered file and forge an arbitrary file path as input. By triggering a change in the URL, we initiate the PDF.js rendering process. Since the original file path is overridden by the content on the window object, the remote server URL string is transformed into a temporary URL object in the browser. When retrieving the actual file data, PDF.js fetches it directly from the browser cache, resulting in significantly faster speeds compared to fetching from a database via network requests. In conclusion, our solution successfully achieves PDF rendering using file objects by changing the pointers. This method is similar to a man-in-the-middle attack, as it intercepts web pages and provides file streams to PDF.js. It is important to note that during the file stream invocation, it is crucial to differentiate between file URL objects and file paths. When obtaining the current page's URL, the file parameter is treated as a file path string, which can be either a relative or absolute path. However, during runtime,

the file parameter is an object representing a file URL, serving as a class pointer object that points to the file's address rather than a string.

Figure 4-1 illustrates the implementation principle of this approach.

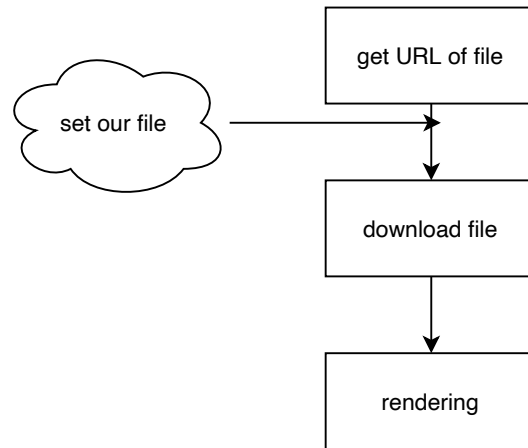


Figure 4-1 : File hijacking

This solution brings several advantages. Firstly, it reduces the burden on the server because files do not need to be uploaded and stored on the server. Instead, files are rendered directly on the client-side, reducing network transfers and server resource consumption. In our project, this translates to a reduction in storage space and management overhead for files containing forbidden content on the server. Secondly, it enhances the user experience by allowing users to immediately view their uploaded PDF files in the browser, without waiting for upload and server processing times.

Furthermore, this approach provides better privacy as files do not need to be transmitted over the network to the server. Users can handle and view their PDF files in a local environment, ensuring greater security and reliability. Additionally, this solution offers more flexibility as users can choose to selectively upload files to the server for further processing or sharing, or directly use PDF.js in the browser for viewing and manipulation.

In summary, our optimized solution utilizes the functionality of PDF.js and the convenience of the upload component to enable direct uploading and rendering of PDF files in the browser. By reducing server load, improving user experience, and enhancing privacy, our solution provides users with a convenient, efficient, and secure way to manage and view their PDF documents.

2) High light and Lazy render

Two issues encountered in the rendering solution are lazy loading and highlighting. Highlighting refers to the need to highlight forbidden words in the document, while lazy loading is a resource optimization technique. PDF.js incorporates this technique by only rendering the pages that will be displayed and hiding unnecessary content to conserve browser resources. Specifically, as the user scrolls or zooms the PDF pages, PDF.js dynamically loads and renders new pages and content as needed. This approach gradually downloads and stores the file's data based on the user's interactions, providing a smooth browsing experience. It reduces memory usage and ensures fast response when navigating through the file. However, this method leads to the loss of the original highlighting effect because previously rendered pages are destroyed when the number of renderable pages is exceeded and need to be re-rendered.

To address this issue, we have implemented a solution that synchronously applies highlighting during the rendering process. By analyzing the PDF.js code, we pass the highlighting function as a global object to PDF.js and incorporate it into the rendering process. Specifically, when PDF.js performs a new rendering of the interface, this function is invoked. This approach successfully overcomes the problem caused by lazy loading while introducing the challenge of multiple renderings. To avoid repeatedly executing the rendering function within a single scroll event, we typically use a timer to throttle the function. Throttling ensures that the function is only executed once within a certain time period. In our project, we utilize the Lodash library for throttling to mitigate potential risks associated with repetitive rendering.

3)Iframe transfer

Interface navigation was an additional requirement introduced during the development process. It aimed to display all the keywords and their corresponding page numbers in a table format, allowing users to click on the entries to navigate to the desired pages. Initially, the project attempted to modify the public page number property of PDF.js to achieve interface navigation. However, simply modifying the property would not trigger any page changes; it would only change the displayed value because JavaScript does not listen for property changes. Therefore, the project delved into the source code and identified the page navigation method within the PDFLinkService class based on property querying and debugging. By utilizing this method, the interface navigation functionality

was successfully implemented. As the operation involves inter-page interaction, VUE cannot directly call the methods of the iframe. To overcome this issue, the necessary functions were installed onto the browser object to facilitate cross-page communication effectively.

4.2 Rendering Scheme

This section explores the application of the rendering solution in two aspects: file presentation and forbidden word highlighting. Specifically, addressing the issues of complex frontend rendering and high development costs with minimal improvement in output for PPT and Word files, this project chose PDF.js as a mature PDF rendering solution for the final display rendering. Additionally, the workload was reduced by converting the files to PDF. Although we mentioned in Section 3.1.3.2 that this approach may incur additional time overhead, the number of problematic files is usually small. By leveraging the ability to analyze the presence of forbidden words in the files, users can quickly receive feedback. The file upload conversion process can be completed during the time users spend browsing the forbidden word results. This approach reduces user waiting time and enhances the overall user experience.

Regarding the design of forbidden word highlighting, this project proposes two sets of solutions: page overlay and text style addition.

1) Page Overlay

In the page overlay approach, an independent layer of color blocks is created on the Canvas layer to achieve forbidden word highlighting. This is similar to using different colored markers to highlight important content in a book. The specific implementation involves calculating the position and length of the text in the TextLayer layer to create corresponding color block tags and overlay them, thus achieving the highlighting of forbidden words.

When using the Alibaba Cloud service, the position of the text in the file can be quickly determined by counting the characters in the span text and matching it with the returned positions of the blocked text. To calculate the position, we first need to determine the offset and then the length of the color block. The offset is determined by the font position of the forbidden word in the span, and the width of the font can be determined by the font size. The length of the forbidden word color block is determined by the length of the word and the font size. The specific formula is as follows:

$$Offset = fontSize * wordsPositionr$$

$$WordLength = fontSize * wordsLength$$

In this approach, there are several advantages:

(1) No need to modify the content of the TextLayer layer itself, avoiding display issues caused by duplicate rendering. There are also fewer color operations performed on the DOM, eliminating the need for color removal.

(2) When dealing with situations where text is split into different tags, only calculating the initial position and length is needed to achieve rendering in one go.

However, this approach has a major drawback. Due to different font styles, characters may have varying widths. For example, the width of a digit may be different from that of a Chinese character. This can result in discrepancies when calculating the overall length of forbidden words and offset, affecting the alignment between the highlighted portion and the body of the forbidden word. This drawback is magnified in longer texts and may lead to suboptimal visual effects in practical display scenarios.

2) Child Span

The approach of using nested `` elements and applying inline CSS to change the text background is an effective way to achieve text highlighting. By wrapping the forbidden words in `` tags, it ensures that the highlighted sections align with the rendered text on the canvas layer. When using Tencent Cloud as the content detection service, quickly locating the text positions through keyword queries without complex calculations is possible. However, this approach has a limitation when dealing with text that spans across three or more `` elements. It would require further development to address this issue in the future.

There are some drawbacks to this approach as well. Firstly, it involves handling multiple renderings and frequent DOM operations. When performing multiple renderings, it is necessary to clear the previous styles; otherwise, the highlighted text length may increase with the font size. Secondly, when dealing with text that spans across multiple `` elements, even with the approach of finding the text position within the entire page content and determining its position within different `` elements, it still requires adding styles to each `` and calculating the specific position of the text within each span, which adds complexity to the implementation.

Both of the mentioned approaches can accomplish the task of text rendering, each with its own advantages and disadvantages. After comparing the practical results, this project has opted for the second approach, which is span-based highlighting. Although it incurs higher computational costs, the number of forbidden words within a single page is unlikely to exceed the browser's performance limit. Overall, the delay in user experience is manageable, and this approach provides more accurate visual results.

Since this project uses Tencent Cloud as the content safety detection service, there are limitations in obtaining the precise positions of the text. The approach of obtaining the position of forbidden words within the text loses the advantage of contextual detection, leading to a decrease in the precision of detection and rendering. Additionally, some methods of evading forbidden word detection, such as including spaces between the text, may bypass this rendering approach. Even if Tencent Cloud has detected the keyword, it may not be rendered properly. In future updates, more optimization and research can be done to address the challenges related to rendering text that spans across multiple `` elements.

4.3 Conclusion

This chapter provides an overview of the browser rendering techniques used in the project. It discusses the reasons for adopting the PDF.js library, identifies the issues encountered, and presents the corresponding solutions. Additionally, it explores the highlighting approach for forbidden words, outlining the advantages and disadvantages of each method and offering potential solutions to address the associated challenges.

Chapter 5 Content Review Platform

5.1 Analysis and Design

5.1.1 View Point

The forbidden words detection system is mainly aimed at the content authors. With the help of this system, the content published by users can be detected quickly, so the function of the system is mainly considered from the perspective of users. The Table 5-1 shows the users of system. At the same time, we also take into account the other stakeholders, which has introduced in section 3.2.2. The other things system should be consider is external systems, such as the content review service like Ali Cloud or Tencent cloud. They may provide different service and data format.

User	Background	Needs
Content authors	With the development of the Internet, more and more attention has been paid to the content compliance of network information.	Allow quickly identify the content uploaded by users.
	Many enterprises have put forward the requirement of content compliance testing to strengthen the risk management of information release by means of machine audit. .	Be able to efficiently display the location of prohibited words.
		Be able to raise objections to the system test results and request for review.
Manual auditor	In order to effectively control the cost and complete the compliance management requirements of enterprises, the combination of machine audit and manual audit has become the first choice of more enterprises. When there is something wrong with the machine audit, manual audit is necessary.	Allow the reviewer to view what needs to be reviewed.
		Establish a corresponding liability system.

Table 5-1 : Users of system

5.1.2 Requirement Analysis

5.1.2.1 Function Requirements

- *Login and permission checking*: Because this system involves the author and auditor, it is necessary to set up the login function to confirm the system user's identity

and permission. The author can enter the test page, and the auditor needs to enter the audit interface.

- *Content audit*: Users can use the system to upload and detect the content they need to publish, and at the same time provide users with intuitive visualization of forbidden words.
- *Examination and approval*: The auditor can check all texts and documents that need to be tested manually, approve and choose whether or not to pass the audit, and assume the corresponding audit responsibility.
- *History query*: The system will save the files uploaded by the user and display their audit status and forbidden status. Users can filter and query the uploaded files and make file calls in the unified file service through the file id. Only the audited files can be finally used in the file service system.

5.1.2.2 Non-functional Requirements:

- *Accuracy*: As a computer-aided system which can be used to check forbidden words, the accuracy of the results is very important. The test results of forbidden words should have good accuracy to ensure the compliance of the content published by enterprises. The Tencent Cloud detection used in this topic has high accuracy, the recognition result is reliable to a certain extent.
- *Processing speed*: The system should have the ability to quickly complete file parsing and forbidden word detection. Slow detection can lead to a poor user experience. Usually, the detection time of normal-sized files should be less than 1 s.
- *Usability*: The system should be easy to use, the interface and functions are clear at a glance, and there is no ambiguity. Users should be able to quickly master the use of the system through experience and learning in a short period of time.
- *Extensibility*: The system provides a scalable architecture and interfaces to adapt to changing needs and different business scenarios. Developers can customize rules and policies according to the characteristics of their own business, and carry out

customized forbidden word detection. In addition, the service also supports integration with other services and systems, such as message queues, logging, etc., to further expand its function and scope of application.

5.1.3 System Module Design

The project is based on a browser environment, so Javascript has become a better programming language choice. Thanks to the fact that Typescript and Node, Javascript can run as server-side languages and have good maintainability. As a whole, the system will adopt the way of Bhand S architecture, even if the Web system is used.

1) User Module

The module includes user login and permission control, which is responsible for verifying the identity of users. The user needs to enter the account password in the login interface, and the system verifies and returns to the available page and loads dynamically to ensure that the system function will not be accessed by those who cannot. Figure 5-1 is the flow chart of the user module.

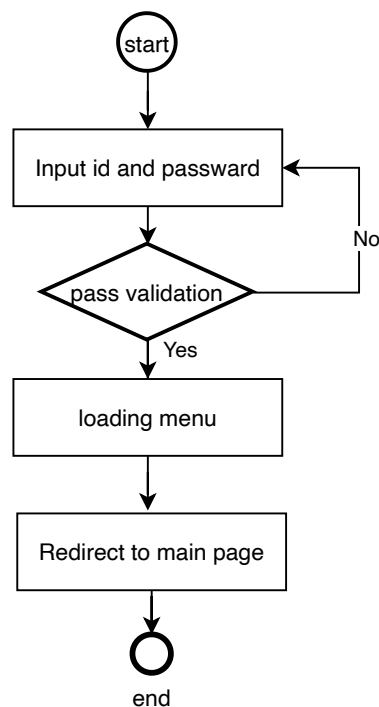


Figure 5-1 : User module

2) Detection Module

The module provides users with a clear, user-friendly, and intuitive graphical user interface. It includes a file upload and result viewing page, which serves as the main page of the system. On this page, users should be able to upload files, and once the system completes the forbidden word detection, the uploaded file will be displayed on the page. Users can click on the forbidden words listed in the table to quickly navigate to the corresponding sections in the document. Figure 5-2 illustrates the workflow of this process.

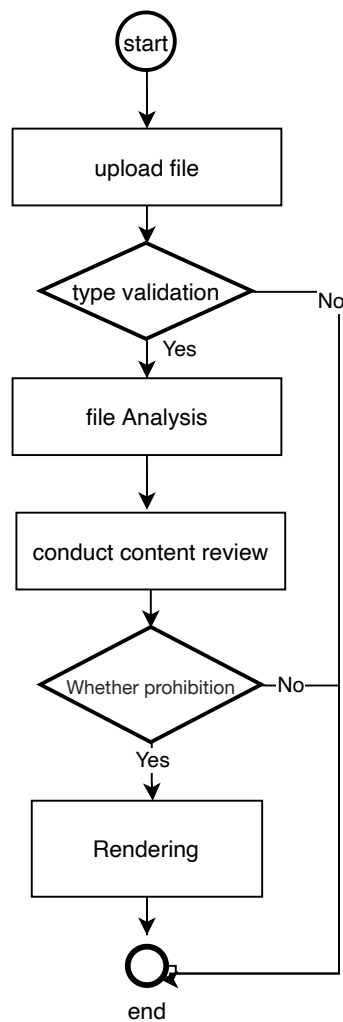


Figure 5-2 : Detection module

3) Audit Module

This module provides users with the function of manual audit. In view of the forbidden word errors found by the machine audit, the user can submit it to the audit for approval, and the auditor can review the errors in the machine audit by clicking the approve button according to the judgment. Figure 5-3 shows the flow of the process.

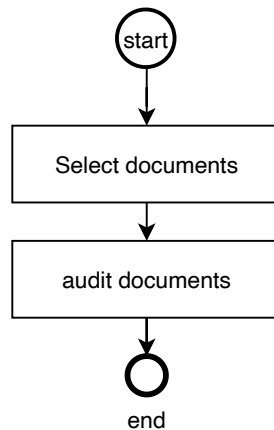


Figure 5-3 : Audit module

4) Record Module

This module provides users with records of uploaded files, through which users can view the approval process of uploaded files, and call the audited files in other systems through the recorded ID. It include a sub module called File module. This module mainly involves the static resource management of files on the server and database-related operations. Figure 5-4 shows the flow of the process.

The preliminary expected structure of the forbidden word detection and rendering system is shown in Figure 5-5 .

The central functionality of the system is content moderation. The use case diagram illustrating the interaction between the user and the system modules is depicted in Figure 5-6 . The user begins by uploading the file that needs to be reviewed. The system performs forbidden word detection by processing the file and invoking the forbidden word detection service. Users can then view the locations of the forbidden words through the interactive interface and gain a clear understanding of the detected content.

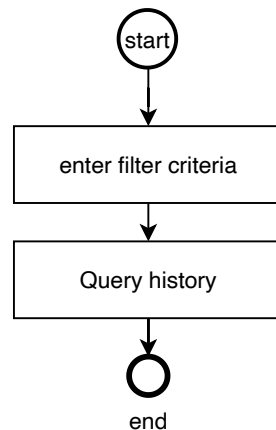


Figure 5-4 : Record module

5.1.4 Database Design

The system consists of two main table part: the user management part and the file management part. The user management part includes tables for users, roles, permissions, and menus. Users can have multiple roles, and a role can be assigned to different users. Roles have a many-to-many relationship with permissions, and permissions have a many-to-many relationship with menus. The file management part includes attributes for the uploader and the reviewer, which are associated with the user table. The entity relationship diagram for the user management part is illustrated in Figure 5-7 . The entity relationship diagram for the file management part is shown in Figure 5-8 .

5.2 System Implementation

5.2.1 Engineering

In the software development process, we emphasize an engineering mindset. Software engineering thinking refers to the systematic, standardized, and collaborative approach to handling software development tasks. This approach encompasses various aspects such as modular and component-based thinking, code reuse and the DRY principle, using version control tools, design patterns, code testing and quality control, as well as documentation and comments.

1)Modular and Component-based Thinking

In the project development, a complete application is typically composed of multiple sub-modules or components. By splitting the different parts into independent compo-

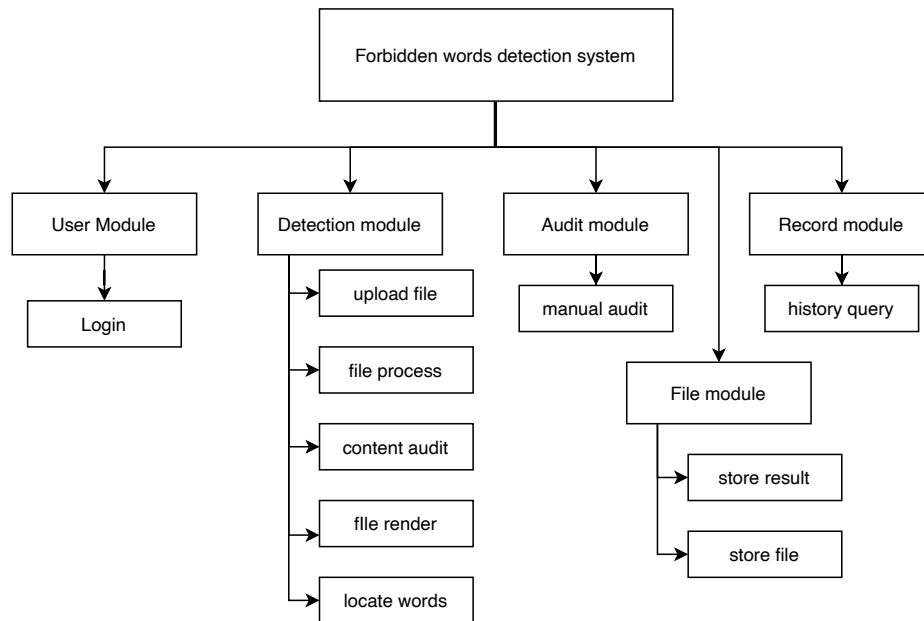


Figure 5-5 : System structure

nents and encapsulating the code into modules, we can reduce code coupling and make the code more maintainable and reusable. This project adopts a single-page application approach, integrating functional pages, modularizing and componentizing them to enhance system cohesion and enable rapid scalability to meet the growing business demands. For frontend API requests, the project uses a unified interface approach, modularizing and controlling all requests uniformly, greatly enhancing code maintainability.

2) Code Reuse and the DRY Principle

In project development, it is common to use existing frameworks and libraries to implement specific functionalities. Additionally, the DRY principle (Don't Repeat Yourself) is widely applied, which aims to avoid writing duplicate code. In this project, commonly used and shared modules, including styles and scripts, are extracted for code reuse to improve code quality and development efficiency.

3) Use of Version Control Tools

Developers typically use version control tools like Git to manage their code repositories. These tools allow developers to track changes in code and coordinate code modifications among multiple developers, ensuring code correctness and consistency. In this

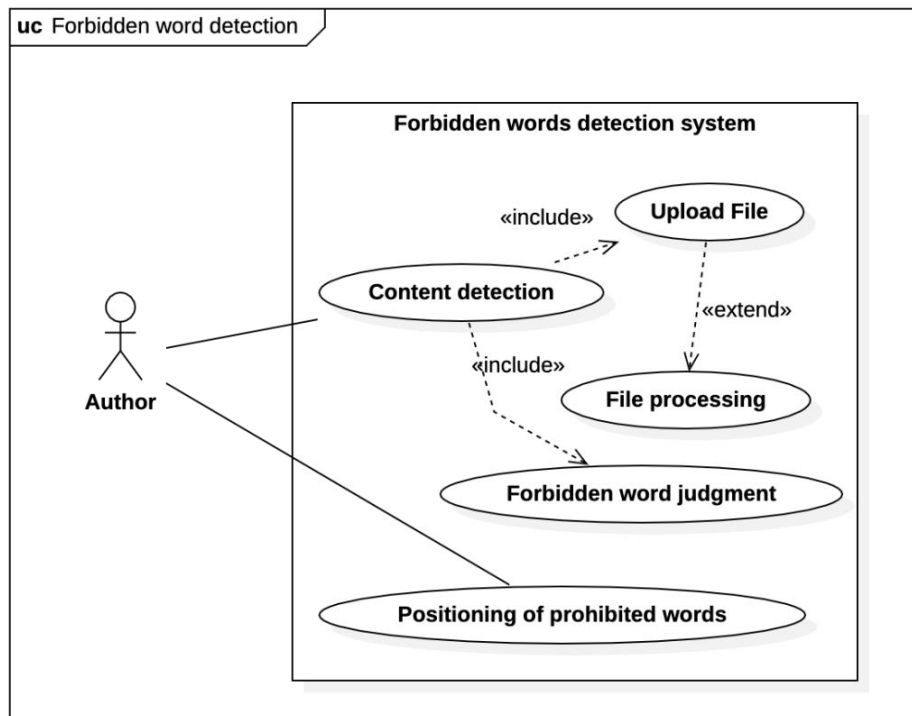


Figure 5-6 : Use case of content audit

project, version control software is used for tracking the development progress, enabling quick rollback to specific versions, and playing a crucial role in requirement changes and technology selection.

4) Design Patterns

In project development, design patterns such as MVC, MVVM, and the publish-subscribe pattern are widely used. These design patterns are integrated into the development frameworks and programming styles to help organize code logic, simplify complexity, and improve the maintainability of the application.

5) Code Testing and Quality Control

In project development, developers typically use automated testing tools to ensure the correctness of their code. This project utilizes the ApiPost integrated tool for project testing. Furthermore, in terms of code quality, linters like ESLint are used to check for syntax errors and potential issues in the code and minimize technical debt.

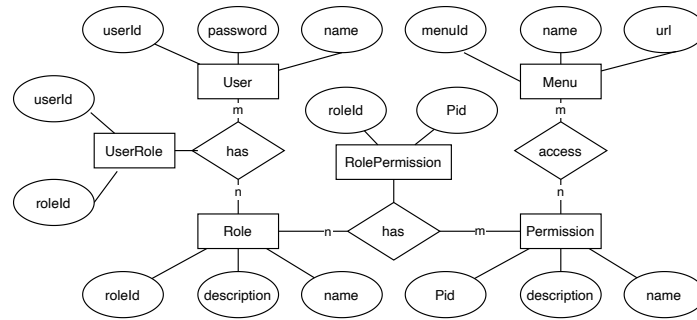


Figure 5-7 : Database of user control

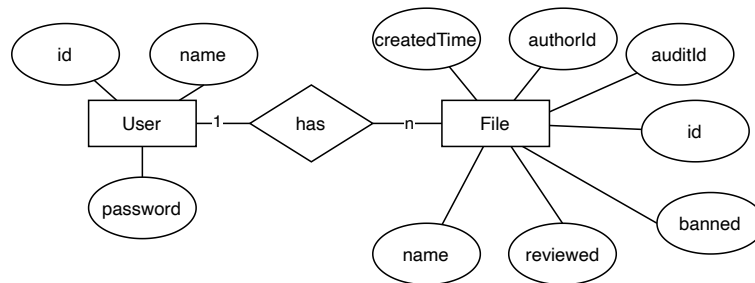


Figure 5-8 : Database of file

6) Documentation and Comments

To facilitate maintenance and future upgrades, this project emphasizes writing clear and informative code comments, as well as documenting how to use APIs or libraries. This helps other developers better understand the code logic and makes subsequent development and maintenance tasks easier.

7) Engineering Control and Environment Isolation

In project development, this project distinguishes between development, testing, and production environments to ensure isolation and quick switching between environments. This helps facilitate rapid deployment and bug fixes while mitigating risks associated with system failures.

5.2.2 Implementation

The project use TypeScript as the programming language for both frontend and backend development. Vue.js is chosen as the frontend framework, while Nest.js serves as the backend framework. TypeScript enhances the maintainability of the system by

providing strong typing and improved tooling support. The unified language ensures the coordination and consistency of the system, offering significant advantages in data integration.

MySQL is used as the database, primarily for storing file storage information and managing user permissions. It enables efficient data storage and retrieval, ensuring the system's functionality in terms of file management and access control.

To achieve system compatibility across different environments, the project adopts a B/S (Browser/Server) architecture, specifically in the form of a web application. Users can access the system through a web browser, making it easily accessible and platform-independent. For future development, frontend development tools will be considered to enable system adaptation for multiple platforms through packaging and bundling techniques.

1) User Module

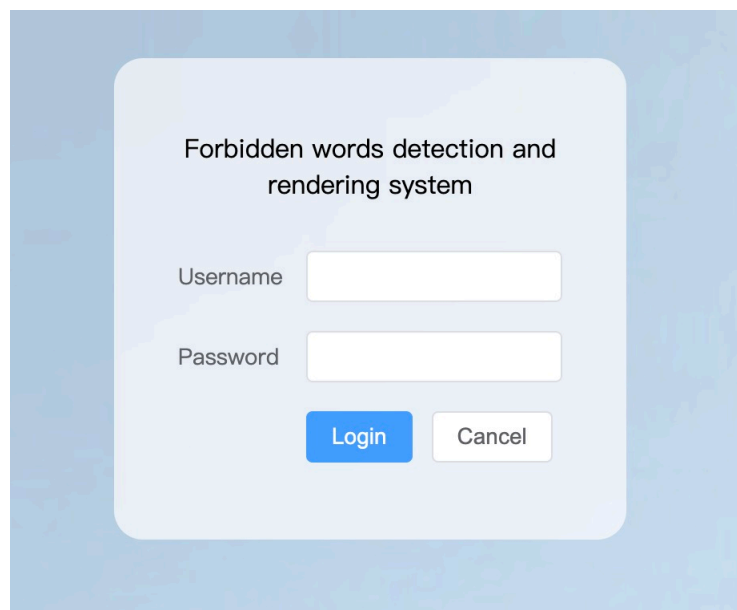
The image shows a login interface for a system titled "Forbidden words detection and rendering system". The interface is contained within a light blue rounded rectangle. It features two input fields: "Username" and "Password", both with white text and light blue borders. Below the "Password" field are two buttons: a blue "Login" button and a white "Cancel" button with a light blue border. The entire form is set against a background of a light blue gradient with a faint world map pattern.

Figure 5-9 : Login

The user module primarily consists of a user login page. When users attempt to access any page of the system without being logged in, they are redirected to the login page. To authenticate the identity and permissions of system users, they are required to enter their username and password. If the user inputs an incorrect account or password, the authentication will fail, and access will be denied. The system will display a prompt

stating "Account or password is incorrect" and redirect the user back to the login page for reentry. Please refer to Figure 5-9 for a visual representation of the login page.

This module primarily employs RBAC (Role-Based Access Control) for permission control. Users are assigned to role groups, and each role is associated with specific permissions that grant access to certain pages. After successful username and password verification, a token is generated for subsequent identity verification. Using this token, users can request access to authorized pages, which will be dynamically loaded by the frontend router. Even if a user attempts to access a page without proper permissions via URL, they will only see a blank page.

2) Detection Module

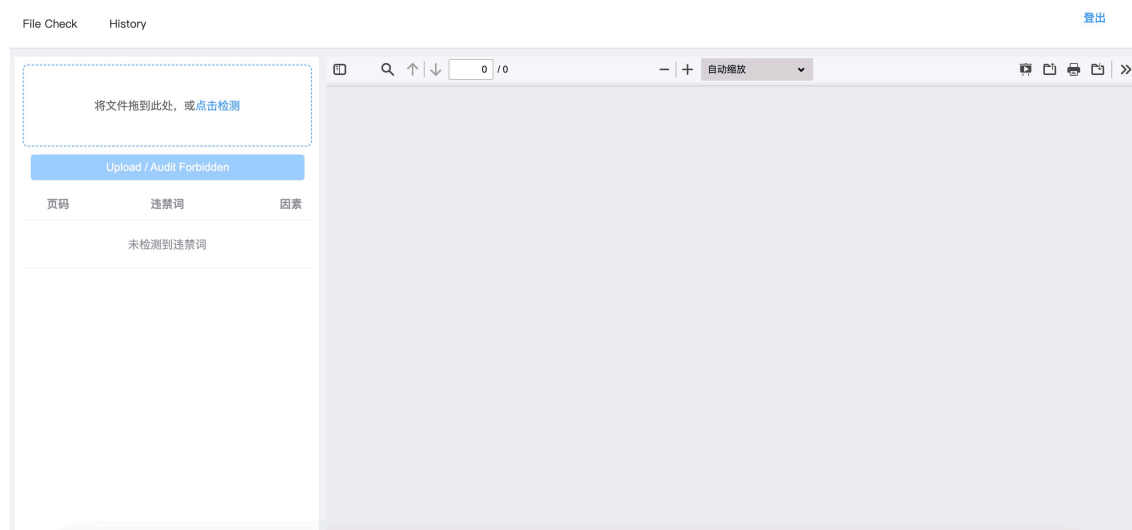


Figure 5-10 : Interactive page before uploading the file

After logging in, users will enter the interactive module where a graphical interface is provided for interactions. Please refer to Figure 5-10 for the interface preview. Users can upload detection files by clicking on the file upload box in the top left or by dragging and dropping files into the box. The backend will parse and analyze the uploaded files. If no forbidden words are detected, a pop-up window indicating "No forbidden words found" will appear. However, if forbidden words are detected and the file needs to be uploaded for conversion, a pop-up window containing the forbidden words will appear first. Once the file processing is complete, it will be rendered in the module on the right-hand side. Figure 5-11 illustrates the situation where a pop-up window with the forbidden

words and the rendering of the forbidden words are displayed after uploading a forbidden file.



Figure 5-11 : Result after uploading the file

Users can inspect the files according to their needs. By clicking on the forbidden words in the left-hand table, they can quickly navigate to the page where the forbidden words are located. After inspecting the file, if the user believes that there is an error in the machine review, they can click the "Upload" button to upload the detection result or submit the problematic file for manual review. If the file is deemed to be without issues, it can also be uploaded to the server for future use. Please note that this button is disabled until a file is uploaded.

开始日期	<input type="text" value="开始日期"/>	结束日期	<input type="text" value="结束日期"/>	<input type="button" value="查询"/>
文件名	文件ID	审核状态	创建日期	上传者ID
违禁词测试.doc.doc	DXM2MACBV6VZL7F10PSFE6	not detected	2023-06-03	2

Figure 5-12 : Problem list

3) Audit Module

The reviewers can access the review interface by logging in through the login module. They can browse the list of files awaiting review and select a file to review. Figure 5-12 shows the list of files awaiting review. By double-clicking on a specific item in the list or clicking on the corresponding action button on the right, they will enter the review page. The review page is similar to the interactive page but without the file upload module. It includes buttons for approving and rejecting the file. Figure 5-13 represents the interface for reviewing a specific file.

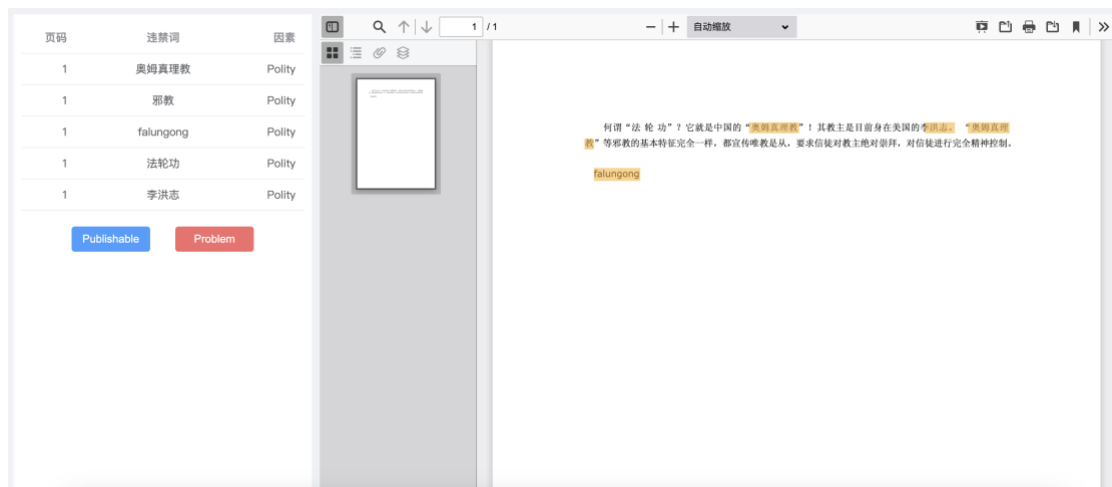


Figure 5-13 : Audit UI

4) Record Module

This module provides users with records of uploaded files, through which users can view the approval process of uploaded files, and call the audited files in other systems through the recorded ID and be used. Users can filter the list of files by the filter criteria above. Figure 5-14 shows the document record page.

开始日期	2023-06-01	结束日期	2023-06-08	审核状态	fail	查询
文件名	文件ID	审核状态	创建日期	审核人ID		
违禁词测试.doc.doc	4AMKMD1IQJPHV6RC8CJ18E	fail	2023-06-03	1		

Figure 5-14 : File list

5.3 System Testing

To ensure the quality and usability of the forbidden words detection and rendering system, standardized testing was conducted, including unit testing and integration testing.

Unit Name	Check Item	Result
file trans	Whether the file is successfully uploaded and converted	PASS
download	Whether to download the file and the type is file stream	PASS
textReview	Whether forbidden words are correctly identified	PASS
login	Whether the Token is returned correctly	PASS
completed	Whether to return the conversion result according to the file ID	PASS
permission	Whether to return the corresponding route according to token	PASS

Table 5-2 : Unit test

In unit testing, various modules of the forbidden words detection and rendering system were constructed and tested. The testing focused on the system interfaces, functionalities, and performance, with defined test contents and metrics. Separate testing was conducted for the frontend and backend programs. The testing was performed using the ApiPost tool for interface definition and test data preparation. By ensuring consistency between the frontend and backend data, they were tested independently. This approach effectively improved team development efficiency and facilitated documentation retention. Table 5-2 presents the results of the unit testing.

In integration testing, simulated user operations were performed after the frontend and backend systems were integrated. This aimed to simulate real-world usage scenarios in a realistic environment. During the testing process, a local server was deployed as the system's server-side, and access was simulated within a local area network to mimic an intranet environment. The testing mainly evaluated the user interface and the functionality of the system, as well as the runtime and performance, which were recorded.

Check Item	Result
Whether the interface elements are displayed properly	YES
Is the interface layout in line with the thinking model	YES
Whether the element style is reasonable	YES
Is it easy to touch by mistake	NO

Table 5-3 : UI test

The user interface testing primarily evaluated the layout of the interface, whether it was prone to accidental touches, and whether it aligned with the users' mental models. Table 5-3 presents the results of this testing.

Check Item	Result
Login	Successfully jump to the correct page according to the user type to correctly display the login failure information
File upload	The file was successfully uploaded to the server, and the contents of prohibited words were returned correctly and prompted accordingly.
File type check	The file type is correctly identified and the file type error message is displayed correctly.
PDF Interface jump	Successfully jump to the correct interface
Menu item	Successfully jump to the correct interface
Audit	After clicking on the list of problem files, get the documents correctly and go to the audit page, and the data will be recorded correctly after submitting the audit results.

Table 5-4 : Function test

The functionality testing focused on ensuring that all functions operated correctly. The evaluation criterion for this testing was whether the pages and buttons responded as expected. Table 5-4 documents the results of this testing.

Performance testing quantitatively assessed the critical response times of the system. The purpose was to ensure reliability and response speed. Table 5-5 presents the results of this testing.

5.4 Conclusion

In this chapter, the detection and rendering system for forbidden words is presented. It is built on the foundation of TypeScript and the Nest framework, allowing easy

Check Item	Result
Rendering accuracy	Offset < 10px; Coverage rate > 90%
Time for file trans	< 3s (1M)
Time for detection of prohibited words	< 1s
Time for PDF Render	< 1s

Table 5-5 : Performance test

access through web browsers. This system revolves around the core functionality of detecting forbidden words and provides users with an interactive graphical interface and features for managing the risks associated with forbidden words.

Chapter 6 Summary and Outlook

6.1 Project Summary

This article proposes the application of forbidden word detection in the browser environment and introduces innovative approaches to file recognition and rendering. The proposed solution includes an improved file recognition module and file conversion method, enabling efficient extraction of file content and reducing time delays in transmitting binary files over the network. Additionally, for rendering PDF files in the browser, the article suggests using interception and replacement of file streams, addressing the limitation of browsers not being able to directly access the local file system and expanding the possibilities for applying PDF files in the browser.

Based on this solution, a user-friendly interactive system for forbidden word detection has been developed. This system adopts a B/S architecture and can be accessed through web browsers. It provides enterprise content management users with functions such as file auditing, quick identification of forbidden words, and compliance application within a unified system. The system offers fast response times, high detection accuracy, and the potential to reduce content publishing risks and strengthen compliance management.

Moreover, as a graduation project in the field of software engineering, this article demonstrates the practical application of software engineering principles throughout the software development lifecycle, including project analysis, design, development, and testing.

6.2 Future Outlook

With the development of the Internet, compliance with internet content has become increasingly important. More and more companies are paying attention to the application of content security in their industries. The task of file content detection faces significant limitations in practical applications. However, the advancement of HTML5 technology has provided more possibilities for browsers to handle binary files. This article proposes a forbidden word detection and rendering system based on the browser environment, introducing file recognition and conversion to achieve efficient detection of forbidden

words. In terms of system development, there are several areas where forbidden word detection can be further improved and applied with better architecture:

Firstly, there is still significant room for improvement in the accuracy of forbidden word detection and rendering. The detection results provided by Tencent Cloud have certain limitations and require more processing time, while Alibaba Cloud's results are more compatible with file browsing. This article has made certain compromises in this aspect, resulting in some deficiencies in the implementation. Forbidden word detection, as an important aspect of natural language recognition, is in increasing demand for content compliance detection. As detection methods evolve over time, evasion techniques may also develop, making research on content detection an ongoing and continuous task.

Secondly, this article mainly focuses on file recognition and conversion in forbidden word detection and introduces innovative solutions for browser file rendering. It also develops a file prohibition detection system that can access resources within the accessible system through file numbering. However, there are still areas for improvement in integrating the system into actual application scenarios. For example, the review process for main body text and the overall workflow still need further enhancement. Due to the excellent portability of this system, integrating the review process into the workflow is a worthwhile task to explore. Additionally, for existing content publishing platforms like WeChat Official Accounts, studying their content publishing processes and creating a content publishing system with compliance checks through their provided APIs would be beneficial.

Lastly, the main operating environment of this article is on the PC side, primarily as a browser application. Considering the increasing use of mobile devices as content publishing tools, adapting the system for mobile devices is a significant direction for development. Additionally, for desktop platforms, many applications are packaged and deployed using Electron to meet various system integration and platform adaptation requirements. As for backend systems, adopting a microservices architecture is a more suitable direction for development, as it facilitates system scalability and adaptability, aligning with future development trends.

References

- [1] 陈力丹. 习近平在网络安全和信息化工作座谈会上的讲话 [J]. 新闻前哨, 2018(59-60).
- [2] 张倩. 现代金融企业数字化转型升级中存在的问题与解决对策 [J]. 文化创新比较研究, 2021(67-70).
- [3] 谭西梅. 金融行业数字化转型的现状、挑战与建议 [J]. 商业文化, 2022(94-95).
- [4] 张继红. 我国互联网金融广告行为的法律规制 [J]. 收藏, 2019, 5.
- [5] 彭昱忠, 元昌安, 王艳, et al. 基于内容理解的不良信息过滤技术研究 [J]. 计算机应用研究, 2009(433-438+447).
- [6] 王宏宇, 陈冬梅. 网络信息内容安全技术浅析 [J/OL]. 电脑知识与技术, 2018(51-52).
<http://dx.doi.org/10.14004/j.cnki.ckt.2018.0492>.
- [7] Wikipedia contributors. Natural Language Processing[EB]. 2023.
- [8] 阿里云. 阿里云内容安全文档 [EB]. .
- [9] 张建军, 孙滔, 孟方. 通过人工智能实现内容智能审核及在世界杯的实战 [J]. 现代电视技术, 2018(52-54+145).
- [10] 黄孝章, 童婷薇. 互联网第三方内容审核服务发展探析 [J/OL]. 北京印刷学院学报, 2023(50-56).
<http://dx.doi.org/10.19461/j.cnki.1004-8626.2023.01.011>.
- [11] 张润峰. 基于特征标识的文件类型识别与匹配 [J]. 计算机安全, 2011(6): 40–42.
- [12] MOZILLA. MIME types[EB/OL]. 2022.
https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types.
- [13] 轩蜗. Go 语言 MIME 类型介绍 [EB/OL]. 2020.
<https://xuanwo.io/2020/04-go-mime-intro/>.
- [14] FIELDING R, RESCHKE J. Hypertext transfer protocol (HTTP/1.1): Semantics and content[R]. 2014.

- [15] PARK B, PARK J, LEE S. Data concealment and detection in Microsoft Office 2007 files[J]. Digital Investigation, 2009, 5(3-4): 104–114.
- [16] van VUGT W. Open XML[J]. The markup explained, .
- [17] Node.js[EB]. 2022.
- [18] npm – build amazing things[EB]. 2022.
- [19] GOSWAMI P, GUPTA S, LI Z, et al. Investigating the reproducibility of npm packages[C] // 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME). 2020: 677–681.
- [20] RASTOGI A, SWAMY N, FOURNET C, et al. Safe & efficient gradual typing for TypeScript[C] // Proceedings of the 42Nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. 2015: 167–180.
- [21] BIERMAN G, ABADI M, TORGERSEN M. Understanding typescript[C] // ECOOP 2014–Object-Oriented Programming: 28th European Conference, Uppsala, Sweden, July 28–August 1, 2014. Proceedings 28. 2014: 257–281.
- [22] FENTON S, FENTON, SPEARING. Pro TypeScript[M]. [S.l.]: Springer, 2014.
- [23] 二爷. python 工具脚本, 网站广告违禁词检测脚本源码 [EB]. 2021.

Acknowledgements

In the blink of an eye, I am approaching the final task of my four-year academic journey. As I write these words, it seems that my student life is drawing to a close, marking the end of a significant chapter. During these four years at ZJUT, I have been incredibly fortunate to encounter numerous individuals who have helped me along the way.

First and foremost, I would like to express my gratitude to my advisor, Li Xiaoxin. You have been guiding me throughout the past six months. From the initial confusion in selecting a topic to the regular progress checks, without your assistance, perhaps my graduation project would not have been completed so smoothly. Our exchanges have provided me with invaluable inspiration and added many highlights to my project.

I also want to thank my colleagues and superiors at the company who have continuously supported me and provided guidance in my professional skills. It is through the leadership and mentorship of such experienced individuals that I have been able to find the right direction for my career.

I am grateful to my lecturers during my university years, including the foreign teachers from Sweden. Your teachings have laid a solid foundation for my future and allowed me to see the world with broader perspectives. I would also like to express my appreciation to the two counselors who provided assistance in my campus life, as well as my class advisor for offering guidance to me on a personal level. To my classmates, thank you for supporting me throughout these four years of university, as our collective efforts have fostered a conducive learning atmosphere.

I extend my gratitude to my roommates, who have shared the joyful experience of four years in our university dormitory. The presence of like-minded friends in this small space has been invaluable.

I would also like to express my sincere appreciation to all the frontline workers who have been combating the pandemic for the past three years. Thanks to your dedication, I have been able to maintain good health during my university life.

Lastly, I want to express my deepest gratitude to my family and my girlfriend. They have always been my strongest support and the warmest harbor in my life, providing me

with both practical and emotional guidance. No matter what challenges I face, they have always been there to accompany and encourage me.

To all the strangers who have helped me along the way and to all those who have shown me love, truly grateful.

Appendix

附录 1 毕业设计文献综述

附录 2 毕业设计开题报告

附录 3 毕业设计外文翻译(中文译文与外文原文)