MULTIMEDIA VIDEO PLATFORM DEVELOPED BASED ON SPRINGBOOT AND VUE

REN ZHIPENG
ROSILAH HASSAN

Fakulti Teknologi & Sains Maklumat, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor Darul Ehsan, Malaysia

ABSTRACT

Video sharing platforms are services found in the realm of media that allow people to upload and watch videos with others online This platform has changed how people enjoy media by providing a space for discovering fun content, learning, and connecting with others. Video-sharing platforms store user-uploaded videos and make them available through sharing features on the platform and other social media sites. People interact with these videos by watching them and engaging through likes and comments while also sharing them to enhance community involvement and connection, among users. The initiative centers on creating a video sharing site aimed at encouraging user generated content that celebrates culture and encourages community engagement. The front end development and design utilize technologies such as HTML,CSS and JavaScript; the back end relies on Java programming language along, with the Spring Boot framework.Notable elements include video streaming,effective content moderation and interactive community management features. Building a content management system enables creators to easily upload and oversee their content to boost user interaction in line, with todays digital advancements by utilizing agile techniques for flexibility and quick deployment while adapting to user input to maintain the platforms relevance and longevity in a fast changing digital landscape. So far, the project has been completed. This project has passed usability testing, It is a video sharing platform with novel elements, for heightened user engagement and easier content exploration.
Key word: video platform,website,SpringBoot,Vue

INTRODUCTON

In the digital age, video-sharing platforms have emerged as a cornerstone of media interaction, shaping how content is consumed, created, and shared across the globe. This literature review aims to dissect the multifaceted influence of these platforms,

focusing on their technological evolution, user behavior dynamics, and the economic implications of their widespread adoption. By examining scholarly articles, industry reports, and empirical studies, this review will explore the mechanisms through which video-sharing platforms manage vast amounts of content, the strategies they employ to monetize that content, and the challenges they face in balancing user engagement with commercial interests. The insights garnered will inform the development of a new platform designed to address current shortcomings, particularly in content discoverability, advertising intrusiveness, and support for diverse content formats.

The rise of video-sharing platforms has transformed media consumption, where users can access diverse content and interact with communities from around the world. Viewers are often inundated with content, leading to decision fatigue and difficulty discovering content that matches their preferences.

Firstly, many advertisements are placed on the video platform. Users often must watch many advertisements during use. Users may lose patience while waiting for advertisements, which affects the user experience. Secondly, Long videos are at a disadvantage compared to short videos. Long videos usually require a longer viewing time, which may cause viewers to choose short videos when they are pressed for time or looking for quick entertainment. Short videos, with their fast consumption characteristics, can quickly attract and retain users' attention, especially in modern life where mobile devices and fragmented time are more common. Long-form videos are generally more expensive to produce and have longer production cycles, which limits the speed and frequency of content updates. Short videos, on the other hand, can be quickly produced and released, and can better adapt to market changes and the immediate needs of the audience. Short video platforms usually have low barriers to entry for users, and anyone can simply shoot and upload short videos. The production of long videos such as film and television works involve more complex technologies and is not easy for ordinary users to participate in. Finally, Videos posted on the website need to be reviewed more strictly to ensure the quality of the videos posted on the website and to prevent many low-quality videos from entering the website and damaging the user experience.

METHODOLOGY

To develop the video-sharing platform, we will adopt an Agile development methodology. This approach emphasizes iterative development, frequent feedback, and continuous improvement, which are essential for dynamic and user-centric platforms like video-sharing services. Agile allows for flexibility and quick adaptation to changing requirements and user feedback, ensuring that the final product meets user needs effectively.

The Agile development methodology is justified for the video-sharing platform due to its inherent flexibility, user-centric design, effective risk management, and

focus on improved quality. Agile's iterative process allows for ongoing changes in project scope and requirements, responding adaptively to user feedback and market trends, which is essential for maintaining the platform's relevance and user-friendliness. Frequent iterations and feedback loops ensure that the platform continuously evolves to meet user needs and preferences, thereby enhancing user satisfaction and engagement. Additionally, Agile methodology supports risk management by facilitating continuous testing and integration, which helps in identifying and addressing potential issues early in the development process. This proactive approach minimizes the risk of major problems moving forward. Furthermore, regular reviews and refinements lead to a higher quality product as they allow for the incorporation of user feedback and necessary adjustments, ensuring the final product meets high standards of quality and reliability.

About system development process, covering key steps and critical technologies involved in architecture design, technology selection, module implementation, and system integration. Initially, the overall architecture of the platform was planned, selecting Spring Boot as the back-end framework and Vue3 for the front-end, in order to achieve an efficient separation of front-end and back-end development. For the database, a combination of MySQL and Redis was used to enhance data storage security and improve access speed, while ElasticSearch was integrated to provide full-text search and personalized recommendation.Furthermore, front-end development emphasized user experience optimization, utilizing named components, dynamic style binding, and responsive layouts to ensure smooth and interactive user interfaces.

Compared to traditional CSS styling methods, Vue3 demonstrates significant advantages in modern front-end development. Firstly, Vue3 supports the use of <style scoped> within Single File Components (SFCs), enabling scoped styles that prevent potential conflicts associated with global CSS definitions. This modular approach to style management greatly enhances the maintainability and scalability of projects. Additionally, Vue3 facilitates dynamic style binding through the use of v-bind and inline styles, allowing developers to adjust styles in real-time based on component states and user interactions, thereby optimizing responsiveness and flexibility in design.Notably, Vue3 integrates CSS in JavaScript features, permitting the use of JavaScript expressions directly within styles. This enables a higher degree of dynamic control, reducing the complexity typically associated with manipulating the DOM via JavaScript in traditional development workflows. In terms of performance optimization, Vue3 supports on-demand loading of style files, ensuring that only the styles required for the current view are loaded. This contrasts sharply with the traditional approach of globally loading CSS files, which often leads to unnecessary style loading and increased performance overhead. Furthermore, Vue3 offers native support for preprocessors such as SASS, SCSS, and LESS, allowing developers to leverage variables, nested rules, and other advanced features to write more reusable and readable styles without requiring additional build configurations. These

capabilities make Vue3 a highly efficient solution in modern front-end development, particularly for complex applications, laying a solid foundation for modular architecture and performance optimization.

Building an advanced video streaming platform, the Spring Boot framework stood out for its excellent modularity, scalability, and efficiency, becoming a core component in our technology stack. The platform aims to provide users with a stable, efficient, and feature-rich video viewing and interactive experience, while ensuring the convenience of backend management and the maintainability of the system. Spring Boot's layered architecture provides solid support for the development of the video streaming platform. The data access layer (DAO) abstracts database operations and implements secure and efficient management of core data such as video metadata, user information, and viewing history. The design of this layer separates data access logic from business logic, improving the flexibility and scalability of the system. In the model layer (Bean layer), Spring Boot's IoC container manages the core entities of the platform, such as videos, users, and subscriptions. These entities encapsulate the data and business logic necessary for the operation of the platform, providing a solid foundation for upper-level services. By using beans, the platform achieves a high degree of modularity, making it easier to expand and maintain functions. As a bridge between users and the platform, the control layer is responsible for receiving and processing user HTTP requests, accurately routing user requests to the corresponding service layer, and generating responses that meet user needs. In the video streaming platform scenario, the control layer implements rich user interaction functions such as video search, content playback, comment management, and user profile editing. The service layer is the core of the platform's business logic, encapsulating key business processes such as video processing, subscription management, and content recommendation. As the boundary of transactions, it ensures the consistency and integrity of business logic. The design of the service layer adopts a clearly defined interface to make the business logic abstract and clear, which is convenient for unit testing and integration testing, thereby improving the stability and reliability of the system.

RESULTS AND DISCUSSION

The live broadcast platform adopts a modular architecture design concept during its construction, and achieves comprehensiveness and scalability of the system through refined functional decoupling. The platform's front-end architecture is developed based on a responsive framework, integrating video upload/player components, intelligent search, user system, and real-time comment area interactive interface. The system's core functions cover full scenario needs such as user registration and login, multimedia content management, accurate content retrieval, and community interaction. The relevant interfaces will be presented in detail in subsequent demonstrations.

For the front-end, we used Vue.js, which is very popular now. The interface we made is both beautiful and easy to use. We split the front-end code into many small modules, such as the module for video playback and the module for comments, so that the code can be reused, it is easy to change anything, and the interface can also be flexibly changed. For the back-end, we chose Spring Boot, which is really easy to use. Features such as dependency injection and AOP allow us to handle business logic more easily. The back-end code is also divided into small modules, some of which are dedicated to business, some are dedicated to processing user requests, and some are dedicated to interacting with the database. In this way, the data can be well saved and found quickly.

As shown in Figure 1, this is the homepage of the video platform. This page supports browsing as a visitor, but you need to log in to experience the full functionality. The interface has a powerful navigation bar, which contains a search box with powerful search functions and buttons for various functions. For example, clicking the login button will jump to the card with login and registration functions.

Figure 1 Home page of video platform

About the login card, as shown in Figure 2, this is a card with a simple design. It has the functions of login and registration. If you have already registered an account, you can enter your account and password to log in to the website. If you are a new user, you can click the register button to register an account to experience all the functions.

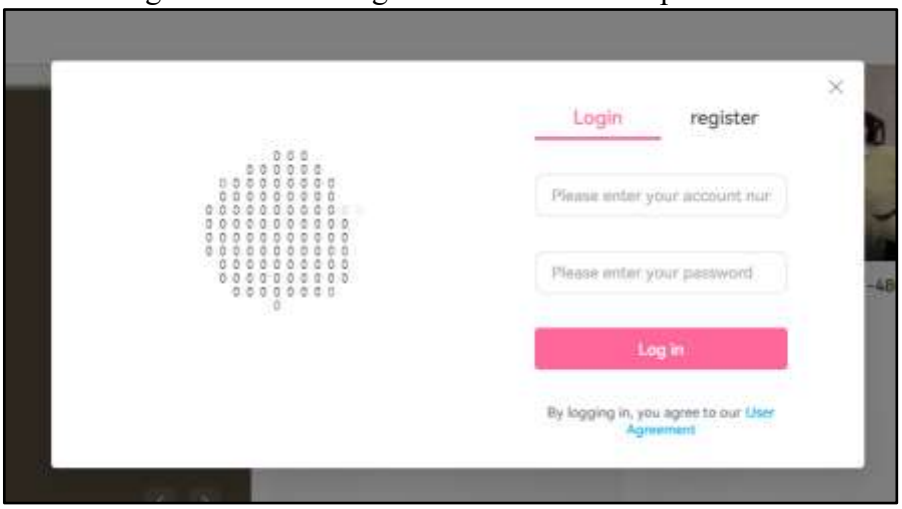

Figure 2 Login page of video platform

Figure 3 shows the search page that you will be redirected to by pressing the Enter key in the search box. Here you can experience the powerful in-site search function, which can accurately search for the information you need based on keywords. Of course, you can directly enter keywords in the search box in the navigation bar.

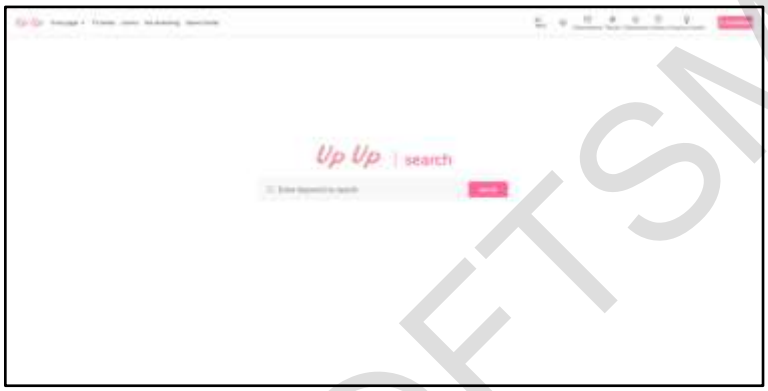

Figure 3 Search page of video platform

Figure 4 is a screenshot of the video playback page. I designed a user-friendly and powerful video playback page for Up Up. This playback page pays more attention to the interactive experience of users on the site, encouraging users to form an active and vibrant community. For the video itself, it has functions such as pause, full-screen playback, and double-speed playback. In addition, I also designed a barrage function for the video, which allows users to send real-time barrages while watching videos, allowing users to interact in real time while watching videos, effectively improving user activity.

A barrage list is added on the right side of the interface, which can clearly view the historical information of the barrages. I also added a video list on the right side of the interface. Click on the video cover to jump to another video. Moreover, you can also automatically play the first video in the video list when the current video ends. Just below the video window, there are functions such as like, dislike, and favorite. Below these buttons, there is also a comment area for the video, where users can also communicate and interact.

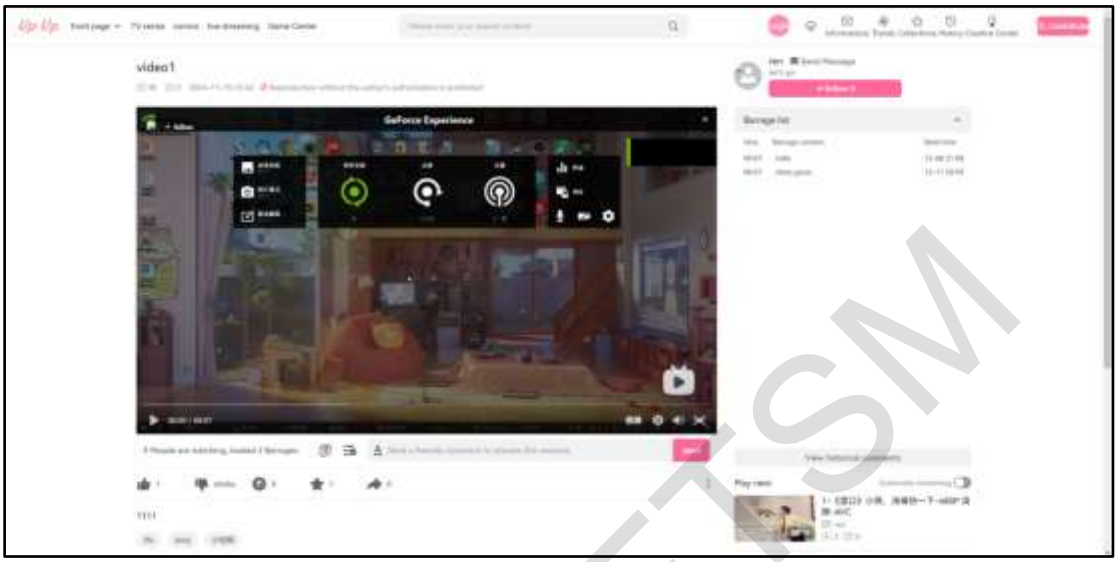

Figure 4 Video play page of video platform

The comment area is shown in Figure 5.

Figure 5 Video comment area of video platform

This website also provides private messaging function, as shown in Figure 6. Here, you can chat with your friends in real time. You can also receive offline messages.

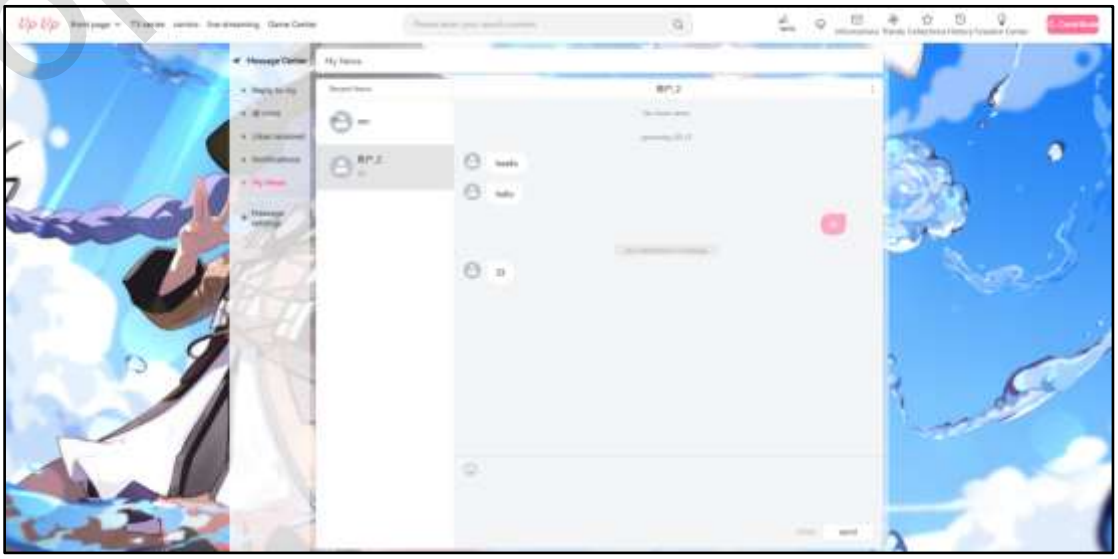

Figure 6 Chat Interface of video platform

As shown in Figure 7, we provide users with a personal page. On this page, users can view the videos they have posted and the videos they have collected. They can also modify their personal information.

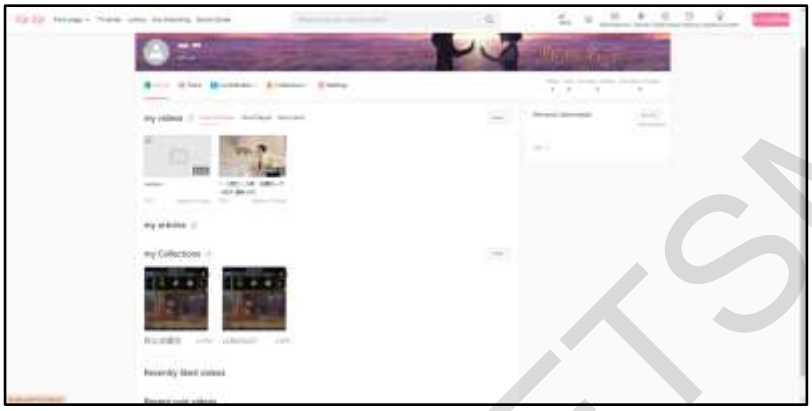

Figure 7personal information page of video platform

The personal information modification page is shown in Figure 8.

Figure 8 Personal information modification interface of video platform

You can also view other people's profiles and see the videos and favorites they have posted. You can also click the Send Message button to start a chat. Other people's personal pages are shown in Figure 9.

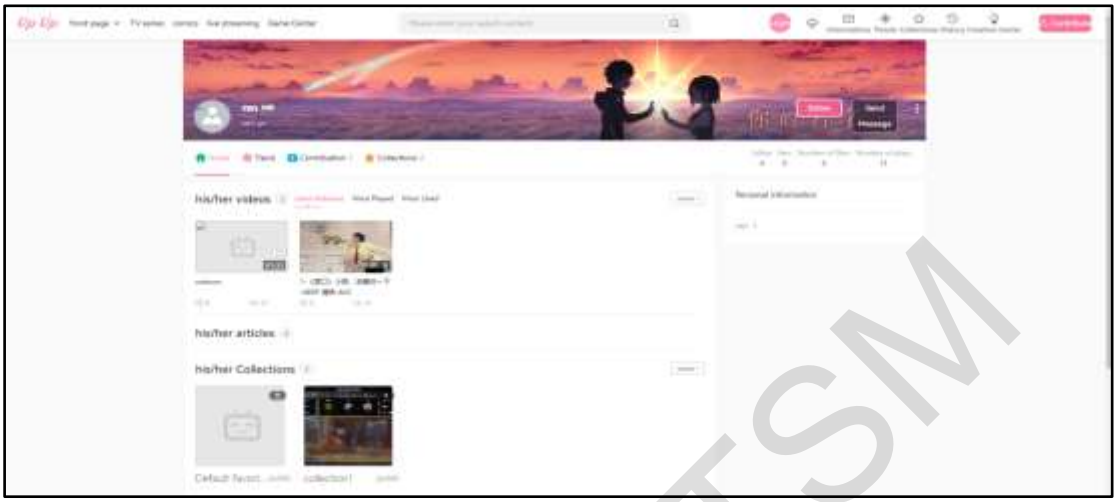

Figure 9 Other people's personal page of video platform

Different from the client, the administrator side is not allowed to be browsed by visitors, so you can only enter the administrator side after logging in. Only accounts with administrator privileges can log in to the administrator side. The administrator login interface is shown in Figure 10.

Figure 10 Admin login page of video platform

The administrator's homepage is very simple, and the functions are mainly integrated in the navigation bar on the left. As shown Figure 11.

Figure Figure 11 Admin home page of video platform

The video review page is shown in Figure 12. On this page, the administrator can view the video information waiting for review, the video information that has passed the review, and the video information that has not passed the review.

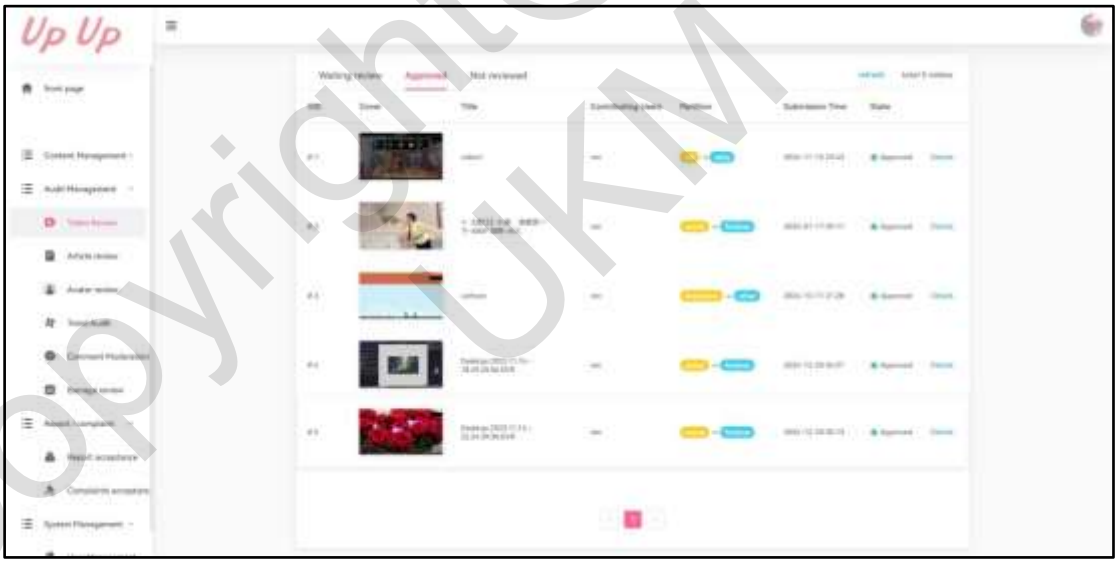

Figure 12 Video review page of video platform

Figure 13 shows the video details page. The administrator can jump to this page by clicking Details on the review page. On this page, the administrator can view the detailed information of the video and decide whether the video has passed the review and is published on the client.

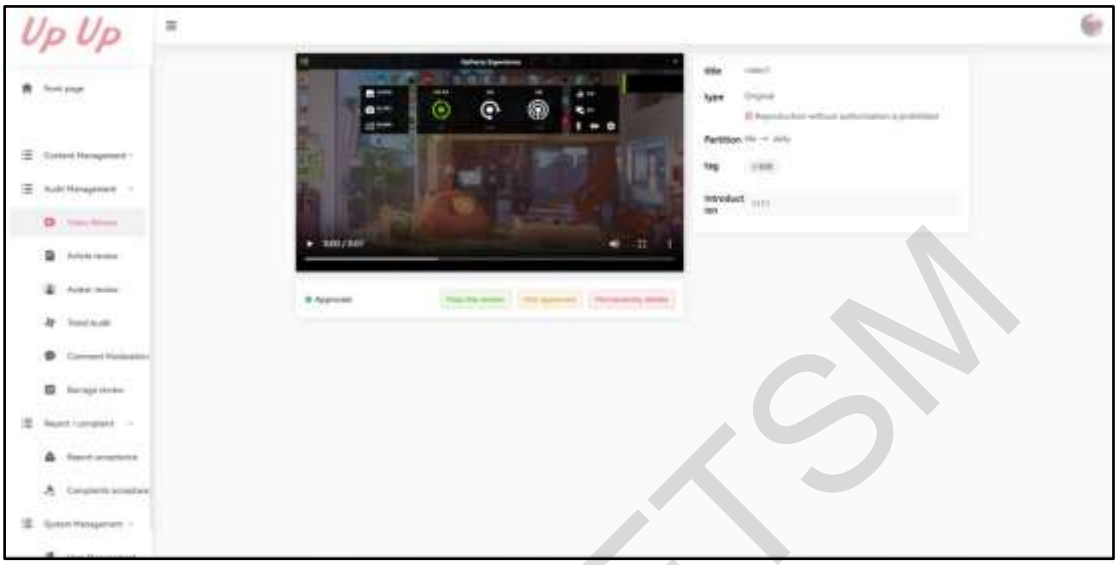

Figure 13 Video detail page of video platform

Usability Testing

Table 1 Result of general information

Question	Response Options	Response Count	Percentage
How frequently do you use video-sharing platforms?	Daily	10	50%
	Weekly	6	30%
	Monthly	3	15%
	Rarely	1	5%
How familiar are you with similar platforms?	Very familiar	12	60%
	Somewhat familiar	6	30%
	Not familiar	2	10%
How easy was it to navigate the platform?	Very easy	9	45%
	Easy	8	40%
	Neutral	3	15%
	Difficult	0	0%
	Very difficult	0	0%
Were you able to find the features you were looking for?	Always	8	40%
	Most of the time	9	45%
	Sometimes	3	15%

Question	Response Options	Response Count	Percentage
	Rarely	0	0%
	Never	0	0%
Were the error messages clear and helpful when something went wrong?	Always	7	35%
	Most of the time	8	40%
	Sometimes	4	20%
	Rarely	1	5%
	Never	0	0%
How would you rate the system's response time?	Very fast	8	40%
	Fast	9	45%
	Neutral	2	10%
	Slow	1	5%
	Very slow	0	0%
How easy was the registration and login process?	Very easy	11	55%
	Easy	7	35%
	Neutral	2	10%
	Difficult	0	0%
	Very difficult	0	0%
How would you rate the video upload experience?	Excellent	8	40%
	Good	9	45%
	Neutral	2	10%
	Poor	1	5%
	Very poor	0	0%
How smooth was the video playback experience?	Very smooth	10	50%
	Smooth	8	40%
	Neutral	2	10%
	Choppy	0	0%
	Very choppy	0	0%
Were the error messages clear and helpful when something went wrong?	Always	7	35%
	Most of the time	8	40%
	Sometimes	4	20%
	Rarely	1	5%
	Never	0	0%

Question	Response Options	Response Count	Percentage
How would you rate the system's response time?	Very fast	8	40%
	Fast	9	45%
	Neutral	2	10%
	Slow	1	5%
	Very slow	0	0%
How satisfied are you with the overall experience?	Very satisfied	9	45%
	Satisfied	8	40%
	Neutral	2	10%
	Dissatisfied	1	5%
	Very dissatisfied	0	0%
Would you recommend this platform to others?	Definitely	11	55%
	Probably	7	35%
	Not sure	2	10%
	Probably not	0	0%
	Definitely not	0	0%

The platform performed well in terms of usability, with an intuitive interface that is easy to navigate. Core functionalities such as video upload, video playback, and commenting demonstrated strong capabilities, meeting the required specifications. The system effectively handled standard use cases, and no major performance issues were observed during testing. The overall quality of the platform is high, with no significant functional failures or system crashes during the testing period. The platform consistently performed well, and key features operated as expected under anticipated usage conditions. However, to ensure the platform can scale and effectively handle a larger user base, further testing in extreme conditions and large data scenarios is recommended. And further improve the system based on user feedback from usability testing.

CONCLUSION

Summary of Video Platform

The Video Platform was designed to address modern challenges in video sharing, focusing on user engagement, content accessibility, and community building. This platform uses a front-end and back-end separation architecture, with technologies like Spring Boot, MySQL, Vue3, and ElasticSearch. Core features include robust video

upload and streaming functionality, efficient content moderation, and an engaging user interface. The system emphasizes performance, scalability, and user security, ensuring it meets the demands of a competitive digital landscape.

Strengths and Limitations of Video Platform

The Video Platform boasts several key strengths that contribute to its success. Its user-friendly design makes it accessible to a wide range of users, prioritizing simplicity, and convenience in navigating the platform. The advanced technology stack, utilizing Spring Boot and ElasticSearch, ensures the platform is both functional and scalable, capable of handling increasing user demands and content. One of its significant strengths is its comprehensive content moderation system, which guarantees the safety and quality of user-generated videos, thus fostering a trusted environment. Furthermore, the platform's focus on community engagement, through features such as commenting, liking, and sharing, cultivates an interactive and vibrant user community. However, the platform does have certain limitations. Due to resource constraints, more advanced features such as AI-driven content recommendations and live streaming were not implemented, limiting the platform's potential for personalized user experiences and real-time engagement. Additionally, the separation of the front-end and back-end requires careful synchronization, and this architectural choice could pose challenges as the system scales. Furthermore, budget limitations led to a reliance on local servers, which impacted the ability to implement more robust cloud solutions or advanced security features like multi-factor authentication.

REFERENCE

Williams, J., & Davis, R. (2020). Trends in video sharing platform advertising and their effects on user engagement. *Journal of Business Research*, 73(3), 320-329.

Wu, J. (2021). Study of a Video-sharing Platform: The Global Rise of TikTokNoam, E. (2019). Looking Ahead at Internet Video and its Societal Impacts. *In Moments in the Development of the Internet*.

Gómez, O. S., Rosero, R. H., & Cortés-Verdín, K. (2020). CRUDyLeaf: A DSL for Generating Spring Boot REST APIs from Entity CRUD Operations. *Cybernetics and Information Technologies, 20*(3), 3-14.

Habes, M., Elareshi, M., Youssef, E., & others. (2022). Social impact of videos at new media platforms on the eLearning acceptance during the Covid-19. *Information Technology & People.*

Sharma, S., & Conboy, K. (2022). Agile-as-a-tool and agile-as-a-culture: a comprehensive review of agile approaches adopting contingency and configuration theories. *Review of Managerial Science*.

WordStream. (2023). TikTok vs. YouTube: Which is best for your goals? fromhttps://www.wordstream.com/blog/ws/2023/11/08/tiktok-vs-youtube

We Do Your Group. (n.d.). Case Study: TikTok vs YouTube.from https://wedoyour.group/case-study-tiktok-vs-youtube/

Lifewire. (2023) . TikTok vs. You Tube: Which one is right for you ?

fromhttps://www.lifewire.com/tiktok-vs-youtube-6665455

Lemonlight. (2024). YouTube vs. Vimeo: The Key Differences in 2024. from https://www.lemonlight.com/blog/youtube-vs-vimeo/

Technical-tips.com. (2024). YouTube vs. Dailymotion - video platforms in comparison. Technical Tips. from https://technical-tips.com/blog/internet/youtube-vs-dailymotion--video-18634

Sun, Y. (2021). Comparison of the Development of TikTok and Bilibili. Journal of Electronic Research and Application, 5(4).from http://ojs.bbwpublisher.com/index.php/JERA/article/view/2507

Mishra, A. D., & Mustafa, K. (2024). Security requirements specification by formal methods: a research metadata analysis. *Multimedia Tools and Applications*, *83*(14), 41847-41866.

Medeiros, U., Vasconcelos, A., Silva, C., & Goulão, M. (2020). Requirements specification for developers in agile projects: Evaluation by two industrial case studies. *Information and Software Technology*, 117, 106194.

Makrickiene, Neringa, Gudas, Saulius, & Lopata, Audrius. (2019). Ontology and enterprise modelling driven software requirements development approach. *Baltic Journal of Modern Computing*, *7*(2), 190–210.

Wikipedia. (n.d.). Use case diagram.Retrieved May 13, 2023 https://en.wikipedia.org/wiki/Use_case_diagram

Wall, C., Macy, J., & Dorsey, C. (2019). *Spring Boot: Up and Running: Building Cloud Native Java and Kotlin Applications*. O'Reilly Media.

Wikipedia. (n.d.). *System requirements specification*.Retrieved June 3, 2023 https://en.wikipedia.org/wiki/System_requirements_specification

Wikipedia. (n.d.). *User interface design*.Retrieved June 7, 2023 https://en.wikipedia.org/wiki/User_interface_design

Jama Software. (n.d.). *How to write system requirement specification (SRS) documents*. , from https://www.jamasoftware.com

REN ZHIPENG(A184588)
Prof. Dr. Rosilah Hassan
Fakulti Teknologi & Sains Maklumat
Universiti Kebangsaan Malaysia