

DEVELOPMENT OF A FILM CLUB WEB SYSTEM FOR DISCUSSING VIDEO CONTENT IN REAL TIME

Vinnitsia National Technical University

Abstract

An analysis of existing solutions in the field of film distribution was conducted and an algorithm for real-time user interaction was developed to implement the possibility of discussing content. Existing methods for implementing real-time interaction were analyzed and the feasibility of using a particular solution was justified. A diagram of the operation of the selected method is presented.

Keywords: cinema, web system, automation, discussion, content, films, real-time interaction.

Анотація

Проведено аналіз існуючих рішень у сфері кінопрокату та розроблено алгоритм взаємодії користувача в режимі реального часу для реалізації можливості обговорення контенту. Проаналізовано існуючі методи реалізації взаємодії в режимі реального часу та обґрунтовано доцільність використання того чи іншого рішення. Представлено схему роботи обраного методу.

Ключові слова: кіно, веб-система, автоматизація, дискусія, контент, фільми, взаємодія в реальному часі.

Introduction

Today, the film industry is not only about creating films, but also about distributing them. Everyone comes to the film business with different intentions, some to implement their own ideas, and some to develop their own business.

Due to the growing popularity of films and cinemas among the general population, the level of competition has increased, which requires the implementation of new solutions that could interest consumers.

The goal of the work is to increase the level of involvement of a new audience by implementing the development of a web system with the ability to discuss content in real time, which will increase the profitability of the business and increase the number of visitors.

The main task is to develop an algorithm for user interaction in real time for discussing the content of the web system, by analyzing existing solutions.

Comparative analysis of existing real-time interaction solutions and justification of protocol choice

There are many solutions for implementing real-time user interaction, but some of them have significant drawbacks to be useful.

The first of the available algorithms is "Server Sent Events" [1]. The main idea is to maintain communication with the server and allow receiving events from it on the client side. The main drawback that does not allow using this solution is its limitations, namely the lack of the ability to send messages to the server from the client side.

The second proposed algorithm is "Long Polling" [2]. Long polling is the simplest way of having persistent connection with server, that does not use any specific protocol like previously described "Server Sent Events". The main idea is that user sends request to the server and it does not close this connection until there is no at least one message to send back to the client. Each time when connection is closed client should reopen it. Fast server installation, because to receive up-to-date information, thousands of users need to send a request to the backend every second, which can slow down the system.

For sending requests, the HTTP protocol is well suited, which allows sending requests to the server from the client side. This does not require any third-party tools. But one of the main disadvantages is that users will need to send requests to the server themselves in order to receive current information. Thus, a delay of at least 1 second, or even more, is possible. There will also be an extremely high level of load on the server side

and at any moment the server may refuse the connection, which will affect the user experience of using the system.

The best option for implementing the possibility of discussing content in real time will be the “WebSockets” [3] network protocol. Compared to previous options, it has almost no drawbacks. The main idea is that the client initiates a connection of a certain type and the server will always notify the user on its own without the need to make additional requests. Therefore, “WebSockets” protocol provides a way to exchange data between browser and server via a persistent connection. The data can be passed in both directions as “packets”, without breaking the connection and the need of additional HTTP-requests.

Below there is the figure 1 where is shown the main difference between “HTTP” and “WebSockets” network protocols.

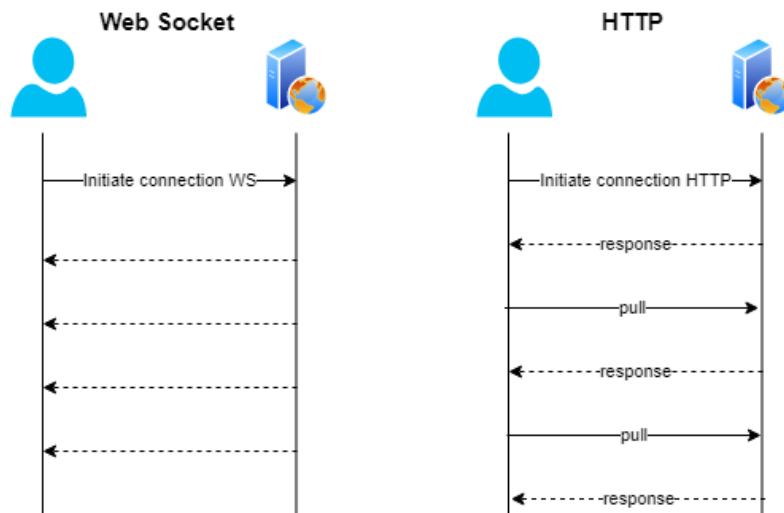


Figure 1. The main difference between “HTTP” and “WebSockets” protocols

From Fig. 1 it follows that when using the “HTTP” data exchange protocol, the user will be required to make constant requests, unlike “WebSockets”.

Figure 2 shows a state diagram of a module with real-time content discussion.

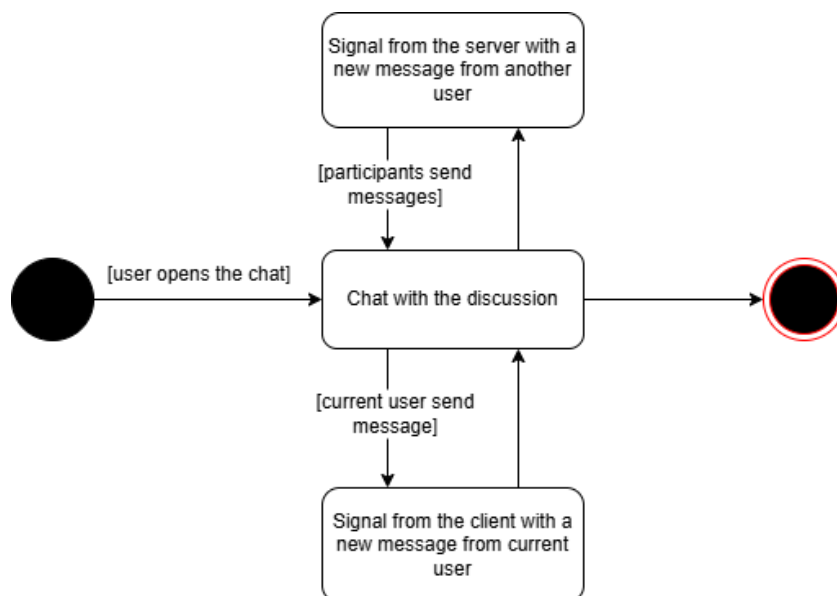


Figure 1. The main difference between “HTTP” and “WebSockets” protocols

Therefore, every time a message is sent, other discussion users will receive a signal and will be able to see the current information.

Conclusion

In the conclusion, the existing algorithms for implementing the possibility of exchanging messages between users who are participants in the discussion of a particular film were analyzed, and the feasibility of using “WebSockets” protocol was proven. The implementation of this functionality will allow to interest new users and increase the time they spend using the system, as a result of which the profitability of the film club will increase. An algorithm for using “WebSockets” protocol on the client was developed, and based on this, a state diagram of the module with content discussion was created.

REFERENCES

1. Server sent events, [Electronic resource]. Access mode: <https://javascript.info/server-sent-events> date of access: 18.03.2025.
2. Long pooling, [Electronic resource]. Access mode: <https://javascript.info/server-sent-events> date of access: 19.03.2025.
3. WebSockets protocol, [Electronic resource]. Access mode: <https://javascript.info/server-sent-events> date of access: 19.03.2025.

Ivan Y. Tsymbal – student of group 3PI-21b, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email: tsymbal.ivan2004@gmail.com.

Supervisor: **Liudmyla M. Magas** — FLD senior lecturer in English, Vinnytsia National Technical University, Vinnytsia, email : ludmag71@gmail.com

Цимбал Іван Юрійович – студент групи 3ПІ-21б, факультет інформаційних технологій та комп’ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: tsymbal.ivan2004@gmail.com

Науковий керівник: **Магас Людмила Миколаївна** — ст. викладач кафедри іноземних мов, Вінницький національний технічний університет, м. Вінниця, email : ludmag71@gmail.com