

Design and Implementation of Webpage Instant Messaging System of E-Commerce Platform of Characteristic Agricultural Products

Fucheng Wan^{1,2}, Xiangzhen He^{1,2}, Yun Zhang¹ and Yu Kou¹

¹ Gansu Yinglan Computer Technology Co., Ltd., China

² China National Information Technology Research Institute, Northwest Minzu University, Lanzhou 730030, Gansu Province, China

Abstract. This paper uses the three-layer architecture of B/S as the basic framework for system development. The front-end interface uses HTML5 to design the overall architecture; CSS3 and JavaScript are combined to complete the rendering and interaction of the interface. The WebSocket protocol is adopted to establish asynchronous and full-duplex link mode for the browser and the server side in the communication aspect, and it avoids using the HTTP long polling and the long poll to make the server process too slow. The entire system uses jQuery write an Ajax engine to achieve pages without refresh show. Multi-language function is primarily achieved by the localization and table level. MySQL is used as the backend database to complete the system's main database table.

Keywords: e-commerce, instant messaging, multi-language, WebSocket, LAMP.

1. Introduction

With the development of the Internet, web instant messaging has begun to gradually replace client-side instant messaging; the main problem it faces is how the browser client can use system calls to communicate with another client browser in other places. In the past, the working ways of desktop applications establish a TCP connection with a process on the other end of the remote host by using a socket. Unlike the browser, the browser has been going the client to request the server, and the server returns the results, therefore, in order to realize the communication between the two clients, the information must be forwarded through the server. In order to solve this problem, the most used one is the WebSocket technology at present; which is not only a web communication method, but also an application layer protocol. It provides native full-duplex cross-domain communication between the browser and the server. It establishes the WebSocket connection (actually a TCP connection) between the browser and the server, and data transmission from client to server and server to client can be realized at the same time.

2. Research Content

2.1 Server-side Functional Requirements

In this web instant messaging system, the core part is the WebSocket server, which is mainly responsible for the mutual communication between the client and the server. This paper uses Node.js to build HTTP and the WebSocket servers, and with Socket.IO to assist construction. The WebSocket server makes it easy to establish a full-duplex communication connection between the client and the server. Once the connection is successful, the server can push the latest data to the client in time and remain open until it is explicitly closed. The main supported functions of the Websocket server are divided into the following modules, as shown in Figure.1.

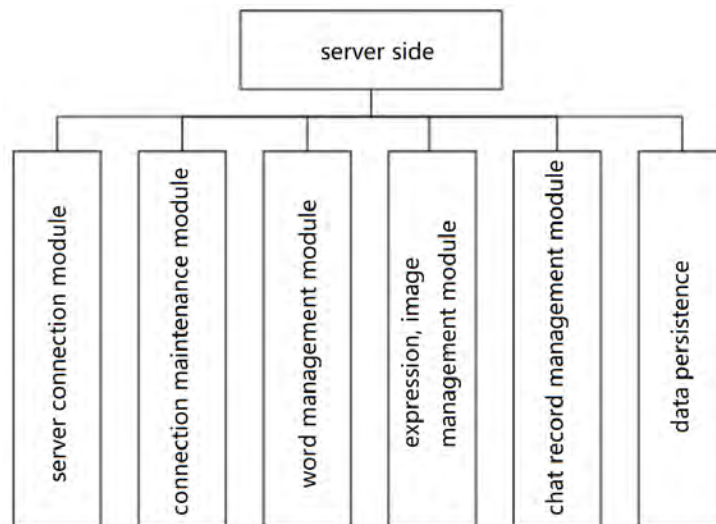


Figure.1 server-side function diagram

(1) Server connection module

It is mainly responsible for the connection between the server and the browser, when the user logs into to the platform, the server needs to verify the user's personal information, after successful login, the server can directly establish the WebSocket connection with the browser, if there is unprocessed information before, the server will send it directly to the user.

(2) Connection maintenance module

It is mainly responsible for maintaining the WebSocket connection between the server and the browser, and handle exceptional cases that may cause the connection to fail. When a message is sent, the server should send it to the browser in time, and the user cannot fail to accept the message due to instability of system.

(3) Word management module

It is mainly responsible for managing text messages sent from the browser. During this process, the server sends the received message to the user. If the user is offline at this time, the server temporarily stores the message and waits for the user to go online and send it again.

(4) Expression, image management module

It is mainly responsible for processing the expressions and image messages sent by the browser. In this process, if the user is online, the image can be sent directly to the browser through the server and handed to the user for handling; if the user is offline, the space occupied by the expression and the image is large, in order to reduce the pressure on the server storage, The server will prompt the sender to wait for the receiver to go online and then resend.

(5) Chat record management module

It is mainly responsible for saving chat records generated during the user's use of the system. The server automatically saves the user's chat record to the database and clears the record every 30 days.

(6) Data persistence

The server performs data persistence on the data operations of all users of the client. Database operations are required for each function.

2.2 Functional Analysis of Client

This system mainly has the functions of the following five modules, as shown in Figure.2:

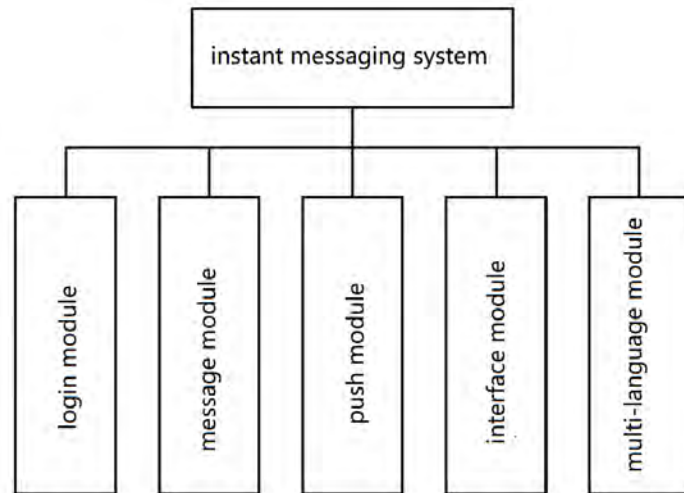


Figure.2 client function module diagram

(1) Login module

The login module is primarily intended to verify the user's personal information. When the user logs in successfully, the authentication is passed, at this time, HTTP and the Websocket protocol perform a handshake to complete the link between the browser and the server side, so that the user can communicate with the merchant at any time, and the server will also send received message to the user the first time.

(2) Message module

The message module is mainly responsible for processing messages generated during the users' chat process. This module includes: sending messages, receiving messages, querying chat records, saving chat records, and withdrawing messages.

1) Receiving messages and sending messages: this module is a function which realizes the communication between users and merchants, different merchants provide different chat windows, after the chat ends, the user can choose to close this window. In this window, the message input box and the message display box are included. In the input box, the user can choose to send text information, expressions and images according to their needs. When the users' input is complete, click the send button or the enter button, it shows that the message was sent successfully. Successfully sent messages are displayed in the message display box, including the content and time sent by the sender, as well as the content and time sent by the recipient. In addition, when the network is busy and not sent to the other side, it will prompt the transmission failed.

2) Query chat records: when the user uses the instant messaging system to send the message, the server will automatically save the chat record in the database, during the saving process, the system will save the information according to the nickname of the chat object and the time of sending the message for later look. When the user clicks to view the chat records, the system will recall the user's chat records with the merchant for the user to view.

3) Save chat history: after the user chats with the seller through instant messaging and purchases goods, it is inevitable that some disputes will occur, the platform can save the chat records in the form of text as a reference for the evidence. The user does not enjoy this function.

4) Message withdrawal: when chatting in normal times, because sometimes it is busy, the information sent may be the wrong object, or when we send information to others, we may find that the information already sent may cause other to misunderstand, in order to avoid unnecessary troubles and misunderstandings, the message withdrawal is a very useful function. Within two minutes of sending a message, the user can choose to withdraw the sent message.

(3) Push module

1) Information push: here it mainly refers to the push of advertisements, according to the records of the goods that the users usually browse, the Internet advertisements are pushed to the proper consumers in an appropriate manner.

2) Order Push: when the user successfully submits the order, completes the payment, and completes the receipt, it will automatically push a message to the buyer, and let the buyer know the status of the current order more clearly.

(4) Third party interface

This system is a multi-language e-commerce platform web instant messaging system, although it has most of the functions used for chat, in order to enhance the user experience and meet the various needs of some users, the platform provides two third-party interfaces for users to more easily exchange with merchants, there are QQ interface and Ali Wangwang interface, respectively. This design requires the user to download and install these two clients on the PC side in advance, when the user clicks these two icons, the corresponding login interface will pop up, and after logging in, the chat window with the merchant can be directly popped up.

(5) Multi-language module

Multi-language platform, as its name suggests, is an e-commerce platform that can support multiple languages (rather than single language), serving different users with multiple languages is a highlight of this platform, make users who use different languages to get the same information from the same website. Therefore, instant messaging systems should also support multi-language function. This system needs to design the display of five languages: Chinese, English, Tibetan, Mongolian and Uyghur to meet the demand.

In addition, the system is only designed and implemented for Chinese and Tibetan at present, and the multi-language display of system follows the entire platform. When the platform is in the Chinese interface, the instant messaging system that is opened is also the Chinese version; when the platform is in the Tibetan interface, the instant messaging system is also a Tibetan version, and the user only needs to download and install the Himalayan input method, and can input and send Tibetan

3. System Design

3.1 System Architecture Design

According to the above analysis, the system is divided into three parts according to the B/S structure, and the overall block diagram of the system is shown in Figure.3.

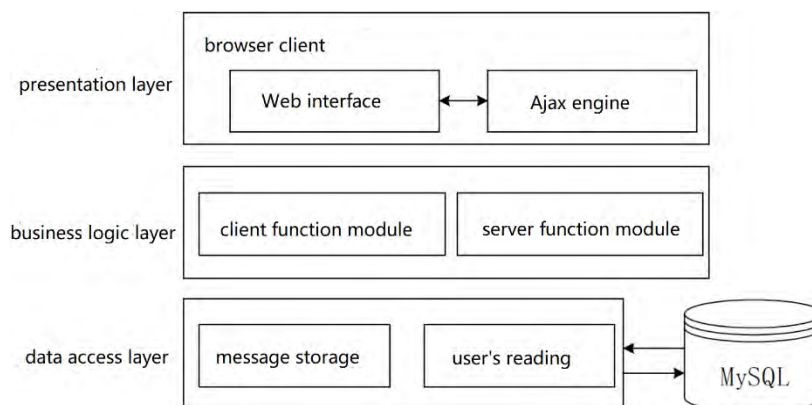


Figure.3 overall block diagram of system

Presentation layer: the front-end interface is mainly built by using HTML5, CSS3 and JavaScript. HTML5 is mainly used to build the structure, and the page is rendered by CSS3. JS is mainly used to realize the interaction of the page. The <link> and <script> tags are only used to import it into the structure of the HTML in implementation aspect. In addition, the jQuery framework is used to write the Ajax engine, which uses the asynchronous way to realize the interaction between the browser and the server. The Ajax engine can handle it by itself for some simple operations. If user's some operations need to call data from the server, the engine will asynchronously work without delaying the user's use of other features. The Ajax functional structure diagram is shown in Figure.4. The page designed has the characteristics of high portability and flexibility.

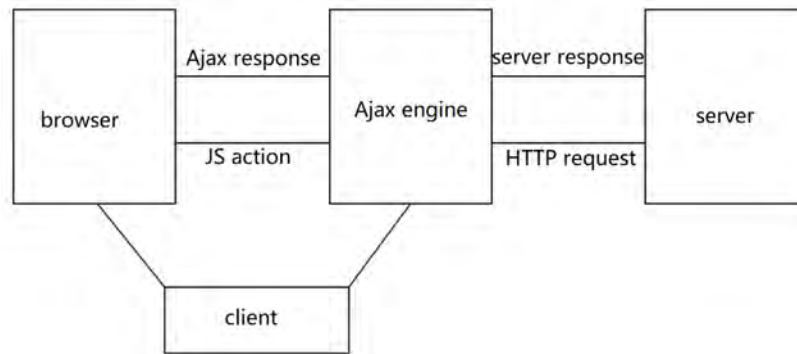


Figure.4 Ajax functional structure diagram

Business logic layer: various interfaces of the system are realized through PHP; various functions of the server are achieved. There are many ways to write an XML file system in PHP, the simplexml method can be used in this system. First, the simplexml_load_file function is used to load xml file \$myxml= simplexml_load_file ('config.xml'), and then you can use \$myxml to access each node. If a node is not repeated, it is unique and can be directly output, if a node is repeated and many node names are the same, the obtained value is an array. Therefore, this method is as easy to use as its name. The chat records generated during the chat process will be saved to the XML file.

Data storage layer: the data storage layer is mainly used to save related data such as chat records and user information, because the user's chat record is saved in the XML file system, then the users' information and other related data are saved in the relational database. This design allows the user's information and other related data to fully use other systems, as long as it is connected to the corresponding read and verification interface, the whole can be easily embedded into other website systems.

3.2 Function Implementation of System Core

(1) Function of sending text message

The sending of text messages is the most basic function of instant messaging. This system is implemented by click event and uses the addEventListener method to add click event to the button. When the user opens the instant messaging system, enter what they want to say in the input box, click the send button, the content will be sent to the server, at this time, it will trigger the socket.IO already encapsulated by the system, so the server can send the message to other side. Figure.5 shows the transmission flow diagram of sending text.

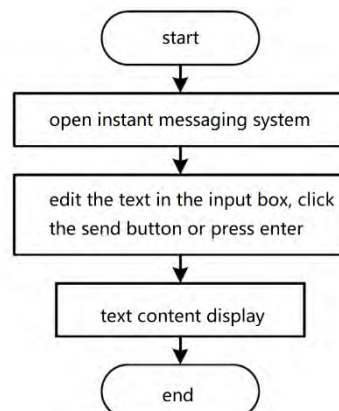


Figure.5 flow diagram of text transmission

(2) Function of receiving message

Because this system uses HTTP and the Websocket server, it avoids some complicated operations of the previous ajax polling and long poll, and avoids the way the client continuously sends message

requests to the server to accept messages. When the Websocket connection is successful, the server can actively push the information to the client once it receives the message. However, during the push process, the server will connect to the database to compare the user information, at this time; the value of the is_get field is 0, each time one is called, the value of is_get is recorded. After the comparison is successful, the AJAX callback function is used, DOM is used to display data in chat interface, each message is spliced back among them, and avoid overriding the previous message. Figure.6 shows a flow chart of receiving a message.

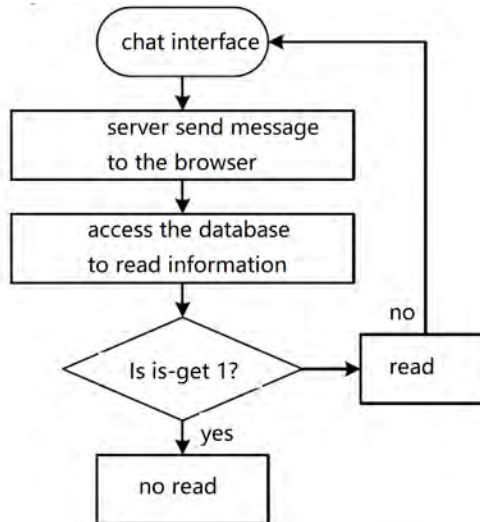


Figure.6 flow diagram of receiving message

(3) Function of sending expression

In order to achieve this function, we must first download the js related content and images of emoji, after download completed, the pages need to display the character of unicode coding, the database saves the utf8mb4 encoding, so the encoding conversion is required. After that, first, three files are introduced, emoji.css, emoji.js, and emoji.png. Use \$('remark-edit') to display the div of the expression (note: the div contains the text that displays the expression and the string, there can be no other tags in it), then we can introduce the Avalon method in JS, emoji.unifiedToHTML(html) is used to parse the expressions in emoji.js, so we can achieve function of sending expressions. The functional flow diagram of sending expression is shown in Figure 7.

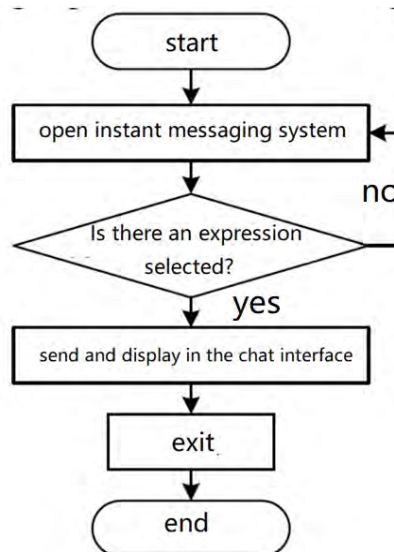


Figure.7 flow diagram of sending expressions

(4) Function of sending image

The sending of the image is slightly more troublesome than sending the text; and the FileReader object in JS will be used here. The FileReader object allows a web application to asynchronously read the contents of files (or raw data buffer zone) saved on the users' computer. When the user clicks on the image option, the local file selection window will pop up, and the users can select the image to be sent according to their needs. During the sending process, the image will be sent as a character string, but when the image is sent successfully, it will still be displayed in the image form when it appears in the message display box. The FileReader gets the base 64 code of the image, previews and send it. Once the file begins to read, no matter what success or failure, the result property of instances will be filled. If reading fails, the value of result is null, otherwise it is the result of reading, and most programs will grab this value when they successfully read the file. The flow diagram of sending image is shown in Figure.8.

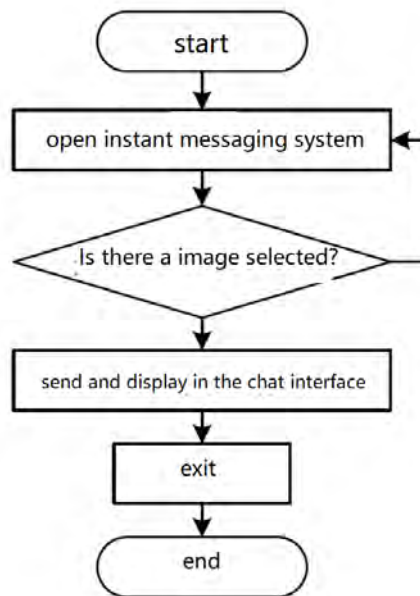


Figure.8 functional flow diagram of sending image

4. Conclusion

This system uses the LAMP development environment and finishes development based on the B/S three-layer architecture. The B/S architecture makes the whole system more layered and structure clear, and it is conducive to the later expansion and maintenance. The server uses efficient Apache, the database selects the most used MySQL, and the Websocket protocol can be used in communication with the client and server in time and effectively, the whole design is well arranged and the code is concise. After experimental testing, the system basically meets the user's needs, but there are still some problems, for example, achievement of high concurrency needs to be improved in the later stage and so on.

Acknowledgements

This paper is supported by the Science and Technology Program project of Chengguan District Science and Technology bureau in Lanzhou City, Gansu Province (2017-4-5).

References

[1]. Liu Wei. Discussion on the Development of Instant Messaging Software and E-commerce[J]. Telecom World, 2013 (13): 20-22.

- [2]. Huang Chongchun. An Experimental Method to Analyze How the HTTP Protocol Works[J]. *Computer Knowledge and Technology*, 2008 (11): 287+302.
- [3]. Tang Lun. Chen Xiaofeng. Development Techniques of Enterprise WebGIS Application[J]. *China Data Communications*, 2005(03): 28-34.
- [4]. Liu Xiaohuan. Using ASP Technology to Write a Simple Forum[J]. *Computer Programming Skills and Maintenance*, 2001 (01): 60-68.
- [5]. Fan Ye. Design and Implementation of Web Instant Messaging System Based on XMPP Protocol[D]. Xidian University, 2014.
- [6]. Zhang Ling, Zhang Cuixiao. The Research of WebSocket of Push Technology[J]. *Journal of the Hebei Academy of Sciences*, 2014, 31(02):49-53.
- [7]. Shen Gang. Design and Implementation of Multi-server Instant Messaging System Model[D], Shanghai Normal University, 2004.
- [8]. Wang Wenli, Wang Zhijun. System Development of BBS with Three-layer B/S Structure Based on ASP.NET[J]. *Journal of Chifeng University (Natural Science)*, 2008, 24(11):24-26.
- [9]. Wang Zhenxing, Huang Jing. Web Instant Messaging System Based on Server Push Technology and PHP[J]. *Computer Systems*, 2012, 21(12): 17-21.