

## **Functionality**

### **Introduction**

The electronic game shopping website is to establish a game platform to provide a safe and convenient trading platform for game enthusiasts, which can not only improve the sales volume of the website, expand the sales scope, but also save manpower and material resources, thus saving costs.

Game shopping system is a kind of online shopping.

With the rapid development of Internet technology, that physical sales began to decline because both sellers and buyers can get discounts when users buy goods from the website.

For this reason, it is very necessary to develop a game shopping website.

### **User interaction**

Customers can browse goods on the login web page. If you want to buy goods, you need to register and log in. Customers can buy goods directly or join the shopping cart and wait for the administrator to send the goods after checkout. After that, the registration email will receive an activation code. Customers can also evaluate this product.

### **Technical introduction**

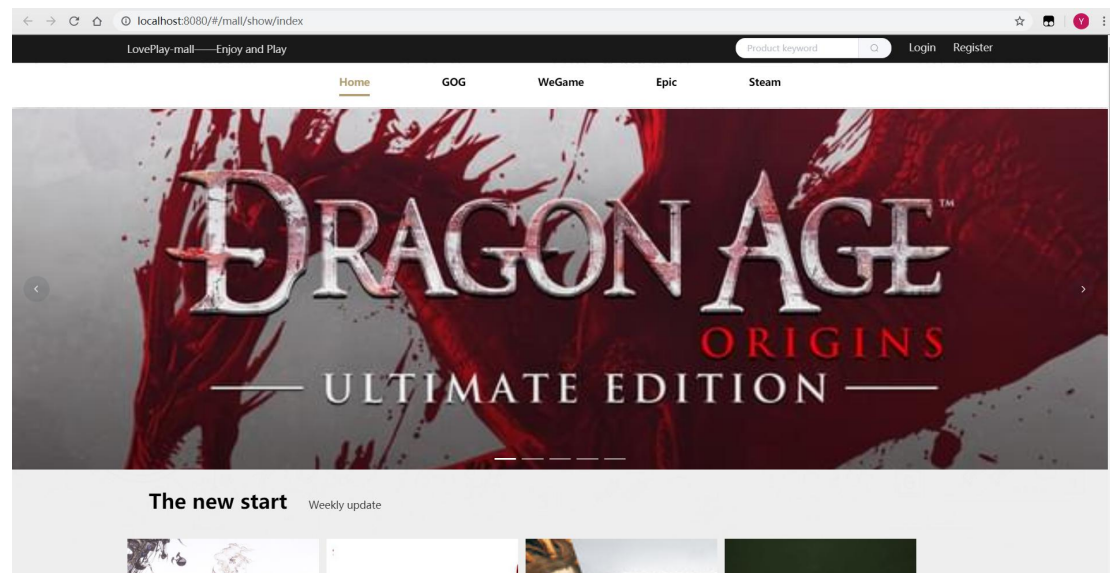
Node.js is not an independent language. Unlike PHP and Python, which is both a language and a platform, Node.js uses JavaScript for programming. Node.js skips http servers such as apache, tomcat, nginx and iis. The front end of the platform adopts technology stack front end including Vue, Vue-route, element-ui, webpack.

The management side includes: Node, Mongodb database, nodemailer and realize the mail push function based on node.

## Requirements

### Front end:

Function: home page.

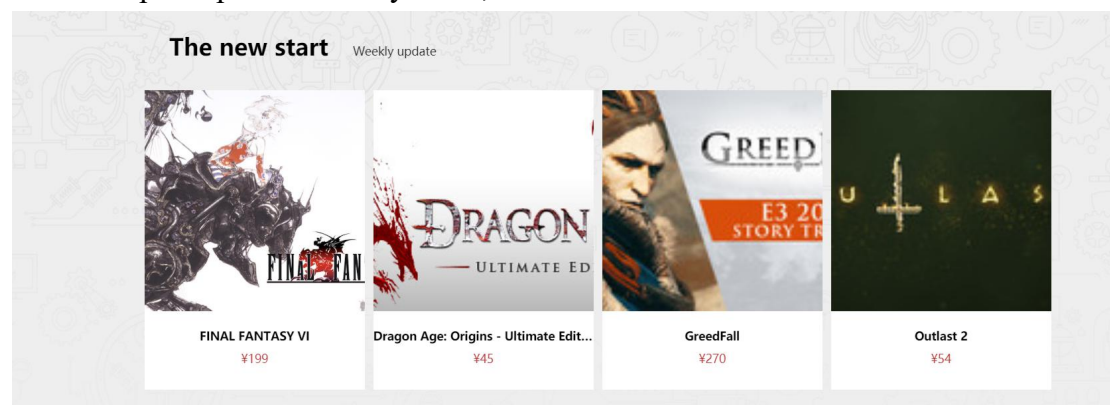


Function: search.



Function: new products on the market.

Reason: update products every week, which can stabilize the source of customers.



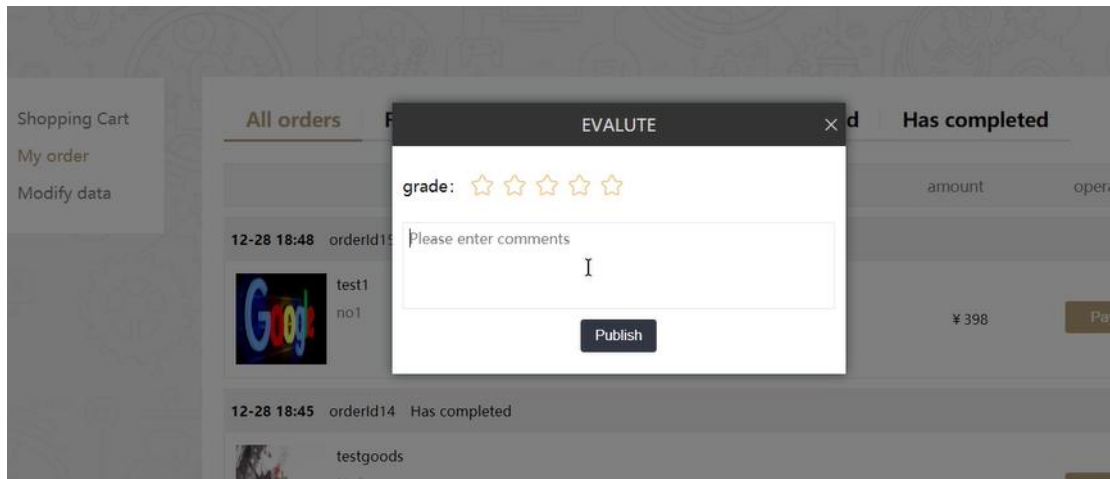
Function: game classification.

Function: it is convenient for customers to browse different kinds of games quickly.



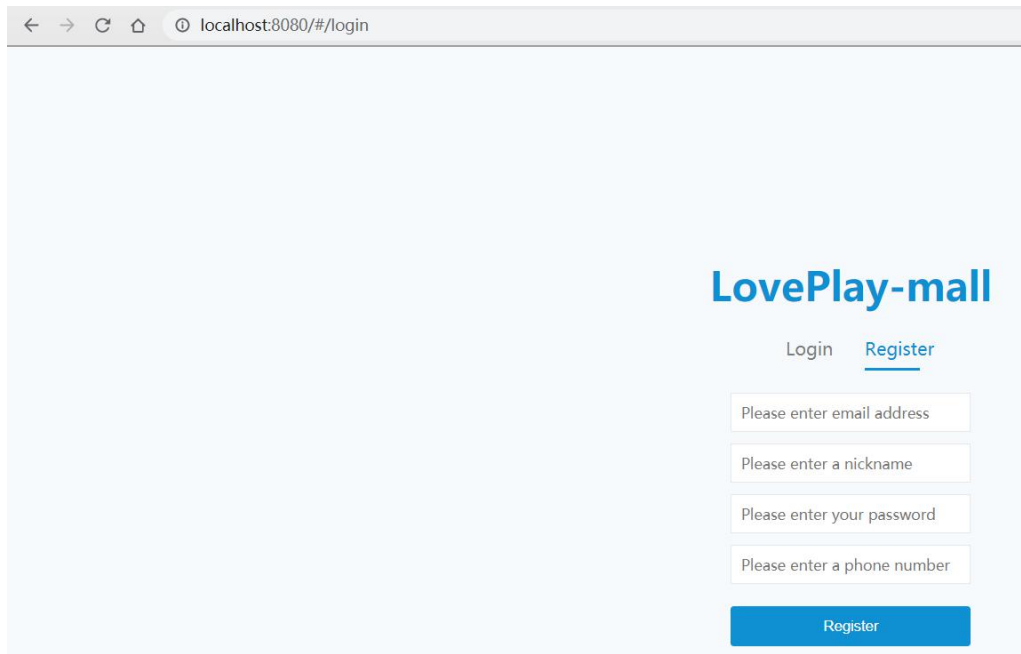
Function: commodity evaluation.

Reason: the feedback on the true feelings of goods can flexibly adjust the input of goods.



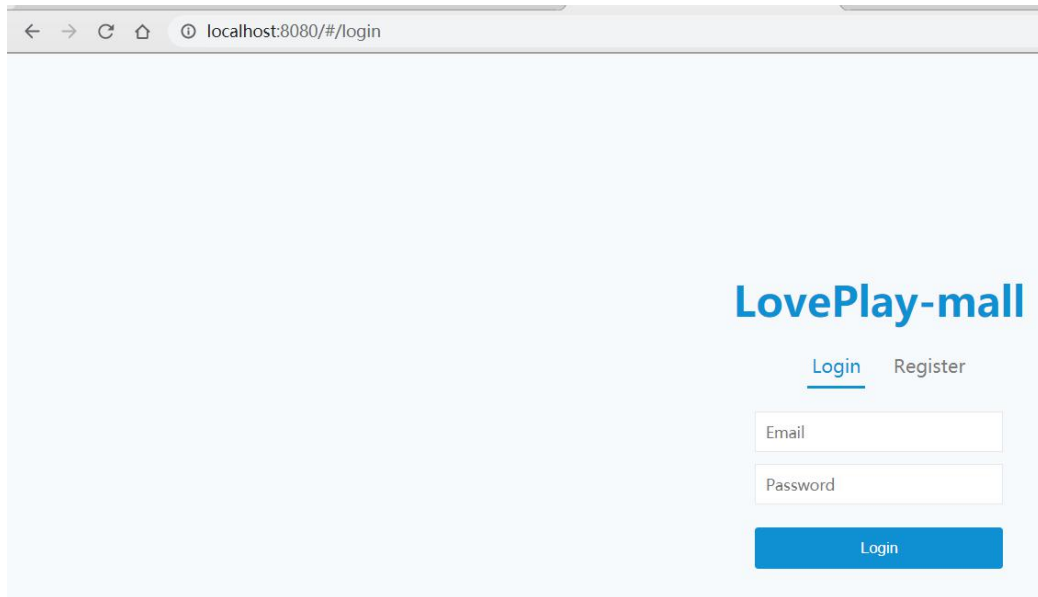
Function: registration.

Reason: user identity information is registered to the application platform



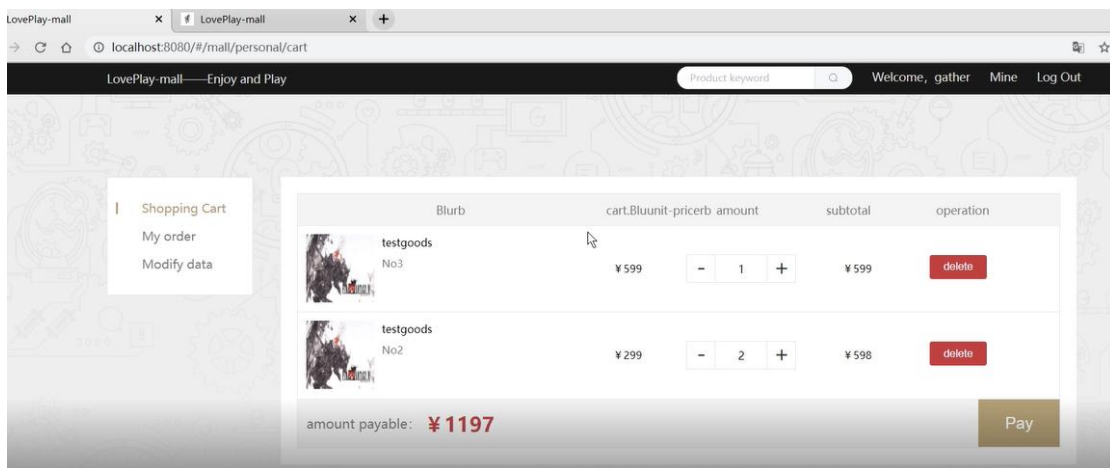
Function: login.

Reason: log in to the verification entry of the website



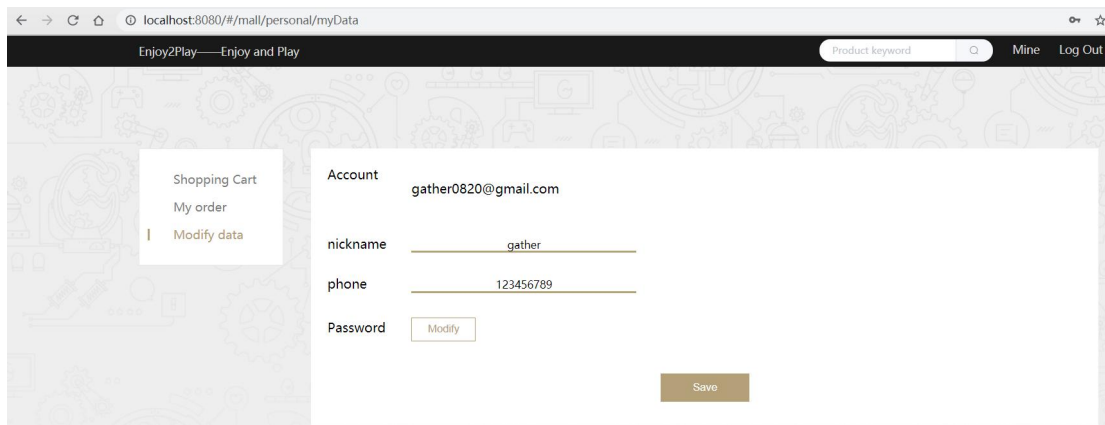
Function: shopping cart.

Reason: provide shopping behavior for users to join the shopping cart



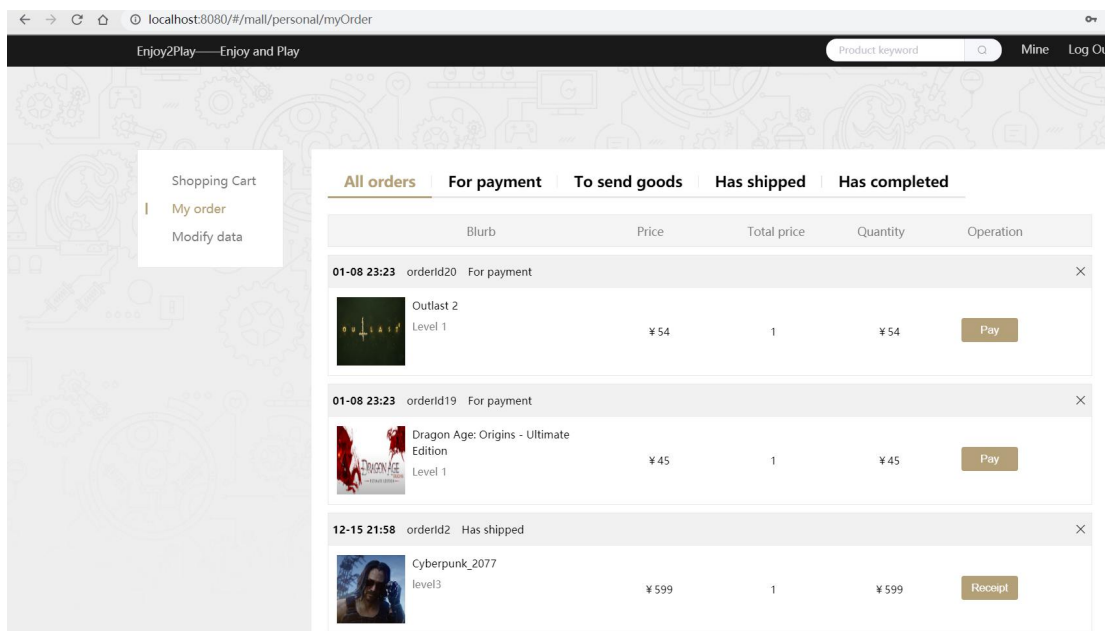
Function: personal information.

Reason: view and modify personal information



Function: my order.

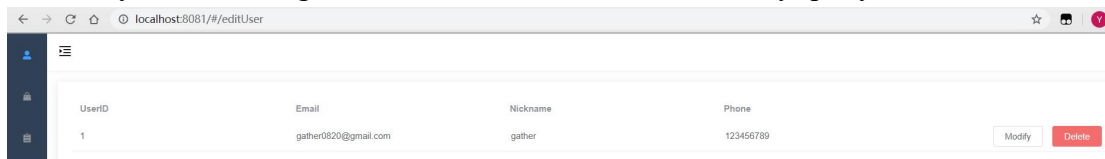
Reason: check order status and record order information



**Backend:**

Function: user management.

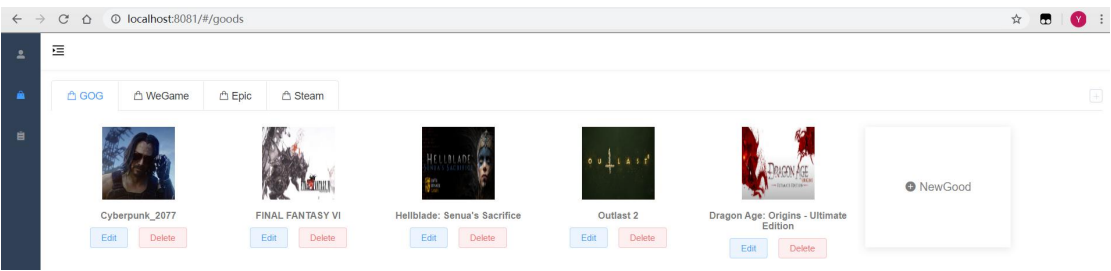
Reason: you can manage user information, delete and modify query user information



Function: goods on the shelf.

Reason: add game categories, add and modify game details and display them to the

front end.



Function: order list.

Reason: record the user's order information for delivery and modify the user's order information.

The screenshot shows a web browser at localhost:8081/#/orders. The interface includes a top navigation bar with tabs for All, Non-pay, Unshipped, Shipped, and Received. Below this, there is a table listing orders. Each row contains columns for GoodsId, State, Nickname, Phone, Goods, Spec, Number, Amount, and Up-Time. The table also includes 'Edit' and 'Delete' buttons for each order.

GoodsId	State	Nickname	Phone	Goods	Spec	Number	Amount	Up-Time
20	Unpaid	gather	123456789	Outlast 2	Level 1	1	54	01-08 15:23
19	Unpaid	gather	123456789	Dragon Age: Orig...	Level 1	1	45	01-08 15:23
16	Unshipped	logout	logout	GreedFall	Level 1	1	270	01-07 16:24
18	Unpaid	logout	logout	Dragon Age: Orig...	Level 1	1	45	01-07 16:24
17	Delivered	logout	logout	GreedFall	Level 1	1	270	01-02 08:12
15	Unpaid	logout	logout	test1	no1	2	398	12-28 10:48
14	Delivered	logout	logout	FINAL FANTASY VI	No3	1	599	12-28 10:48
13	Delivered	logout	logout	FINAL FANTASY VI	No2	2	598	12-28 10:48
12	Delivered	logout	logout	4524352	level1	1	199	12-26 14:52
11	Delivered	logout	logout	Cyberpunk_2077	level1	1	199	12-15 14:44
2	Delivery	gather	123456789	Cyberpunk_2077	level3	1	599	12-15 14:24
6	Delivery	logout	logout	Cyberpunk_2077	level3	1	599	12-15 14:16

Function: Reply to message

Reason: Understand customer needs and answer customer questions

The screenshot shows a web browser at localhost:8081/#/orders. The interface includes a top navigation bar with tabs for noReply and Replied. Below this, there is a table listing replies. Each row contains columns for Goods, Nickname, Phone, Response, and Up-Time. The table also includes 'Replied' buttons for each reply.

Goods	Nickname	Phone	Response	Up-Time
GreedFall	123123	Response: 123123	Replied	01-02 16:11
GreedFall	12312312312	Response: 2131231231	Replied	01-02 16:07
GreedFall	123321123456	Response: hahaha	Replied	01-02 16:05
FINAL FANTASY VI	123	Response: 321	Replied	12-28 22:53
Cyberpunk_2077	????	Response: ! ! ! !	Replied	12-28 18:50

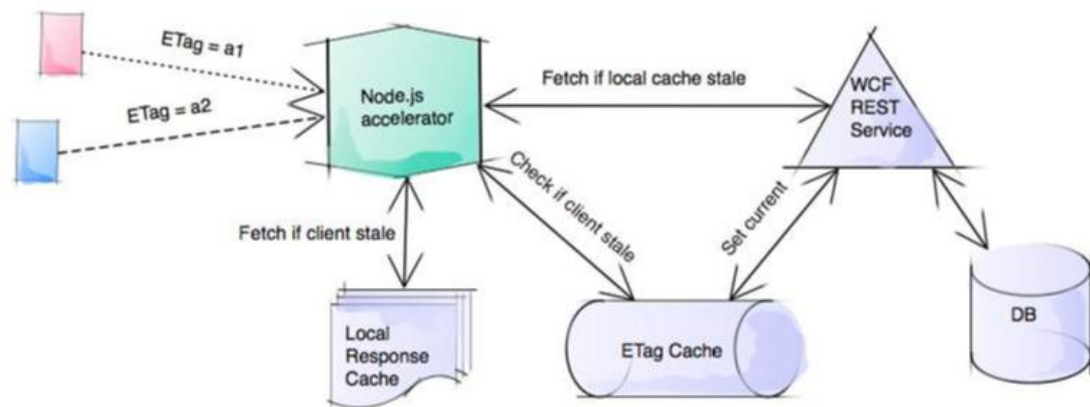
## Design

### System architecture

This system develops a distributed game shopping platform based on Bramp S architecture, which can provide convenient online game shopping for all kinds of game hobbies.

Client system architecture: Node.js construction and server-side connection of the client, used to construct a long connection, send and receive data, to achieve data transmission.

Server system architecture: the server uses Node.js language to construct the process: receiving requests, responding requests, caching processing, and obtaining data information from the MongoDB database to provide data and business support for the client. The system architecture is shown below:



### Process interaction

The system website through the foreground node.js language and the vue framework as the client of the whole system, the external interface service provided by the server side through http request mode adopts the MVC background design mode, and the game shopping application system is completed through the interaction between the front-end and background management process through the MongoDB database.

## Data and code structure:

Data structure:

Admin table: responsible for storing the administrator's login information.

User table: responsible for storing user registration and login information.

Perch.

Goods\_detail table: responsible for storing product details.

Comment table: responsible for storing user comments.

Goods table: responsible for storing commodity information.

Perch.

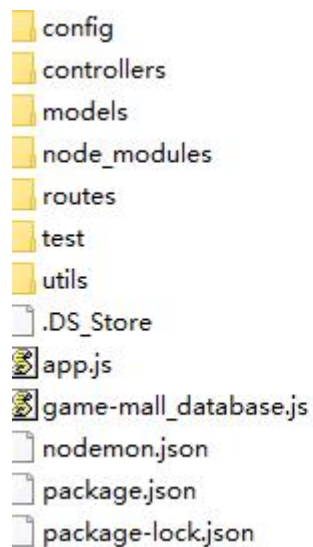
Order table: responsible for storing user purchase order information.

Shop table: responsible for storing the name of the buyer and the information that has been purchased.

Through the design of the above database table, the game online mall has stronger expansibility and maintainability.

## Code structure:

Server :



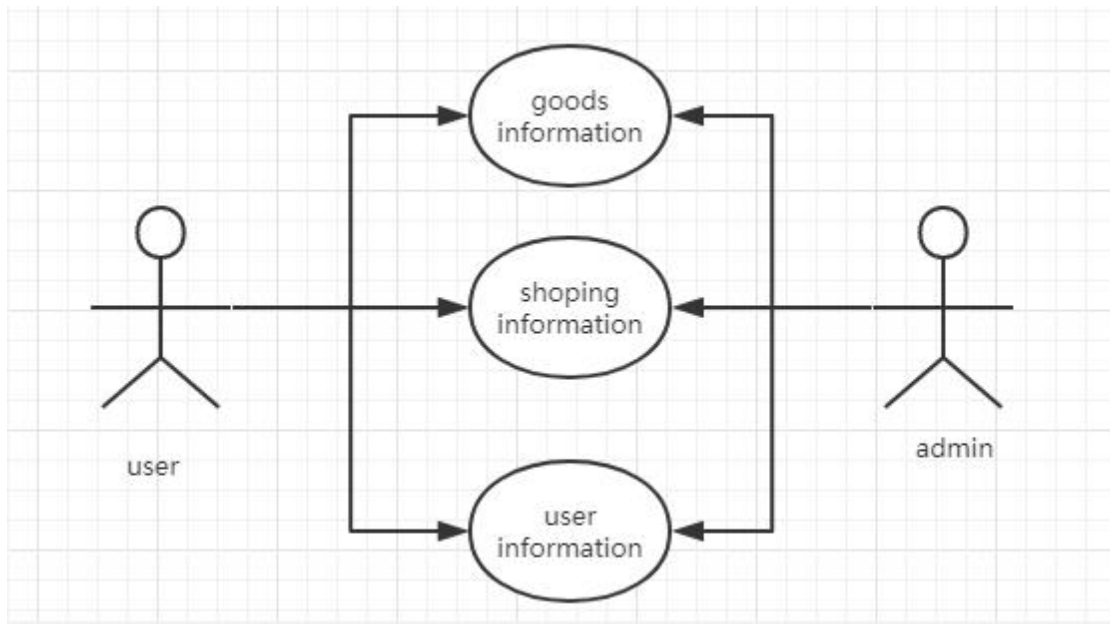
Client:



- api
- assets
- components
- config
- locales
- pages
- router
- store
- styles
- util
- utils
- admin.js
- App.vue
- main.js
- socketio\_admin.js

The above code structure is more general and in line with the standard development specifications, which is more conducive to later code maintenance and functional improvement.

### Overall use case diagram of the system



## **Testing**

### **Test:**

Perform automated text testing: use the automatic script function of comparing text and copy data to see if there is a text error message.

Test results: pass

The UE test was performed: the search results page, the correctness of the search results, and the display of the results page information for unit testing.

Test results: pass

Perform usability testing of the shopping process: the most important process of the website is the shopping process: shopping cart, order reminder, order submission, unit testing.

Test results: pass

Performance tests were performed: request stress tests were performed using Jmeter automation tools.

Test results: pass

Penetration tests were performed: penetration tests for user logins and game payments were performed using JUE automation tools.

Test results: pass

The mail push test is carried out: the mail collection of some users is verified by using the black box test.

Test results: pass, mail receipt rate is 100%.

### **Adopt the strategy to test the purpose:**

This website adopts the combination of black box testing and automated testing, which can find out all kinds of errors and defects in the system with the least time and manpower, and prove whether the function and performance of the system meet the system requirements by testing. whether the website has achieved the expected goal of the planning, whether it can meet the requirements of the business process, whether the interface is friendly, whether the operation is simple and convenient to verify.

## **DevOps pipeline**

**Environment description:**Is a JavaScript runtime environment based on the

Chrome V8 engine, by installing vscode development tools and using visual authoring such as Nodemon.

**Development environment:**The application system uses Windows or Linux operating system environment to download the Node.js installation package that nodejs, chooses according to different platform systems, and installs the installation package of nodejs to complete the environment needed to develop the system.

**Pipe description and use:**The DevOps approach is based on better collaboration and faster completion of the use of the application system. The "integration" of development and operation and maintenance is the bridging of skill sets and practices between developers and operation engineers and the combination of automated (DevOps) tools to create quality and complete the application system more quickly and reliably.

Through more communication and cooperation between the operation and maintenance team and continuous integration, the application system can be developed and released more quickly, securely and with high quality.

DevOps emphasizes the collaboration and communication between software developers and other information technology (IT) professionals in the process of automating software delivery processes and infrastructure changes.

It aims to establish a culture and environment so that construction, testing, and software release can be carried out quickly, frequently, and more stably.

#### **Four outcome indicators of DevOps:**

Deployment frequency: refers to how often applications and services deploy code to a production environment.

Change lead time: refers to the length of time from the time the code is submitted to running successfully in the production environment.

Service recovery time: refers to the time from the failure of online applications and services to the resumption of operation.

Change failure rate: the percentage of applications and services that fail to deploy in a production environment or result in service degradation after deployment.

#### **Personal reflection**

Through this program design, as far as technology is concerned, using the cutting-edge technical framework is that the application is more beautiful to provide user experience, reduce workload, and there are also shortcomings, because the cutting-edge technical framework needs to be studied and explored. It also increases

the development cycle, but has a new understanding of the overall business of the game shopping system and is more proficient in the use of technology.

Need to learn in the future more hands-on operation and preparation of in-depth understanding of the Vue front-end framework and Nodejs language, to combine theory with practice, to learn to think, this is a long process, but they will eventually gain and improve.