Functionality

Introduction-The software is all the latest version (2019-12-31).

The project environment uses and installs: Chrome / Visual Studio Code / Git Bash / Github Desktop / MongoDB Compass.
The program environment uses: Node.Js / MogongoDB; Node.Js include: Vue / Vue-route / Vuex / element-ui / webpack / express / jsonwebtoken / socket.io / nodemon / nodemailer.

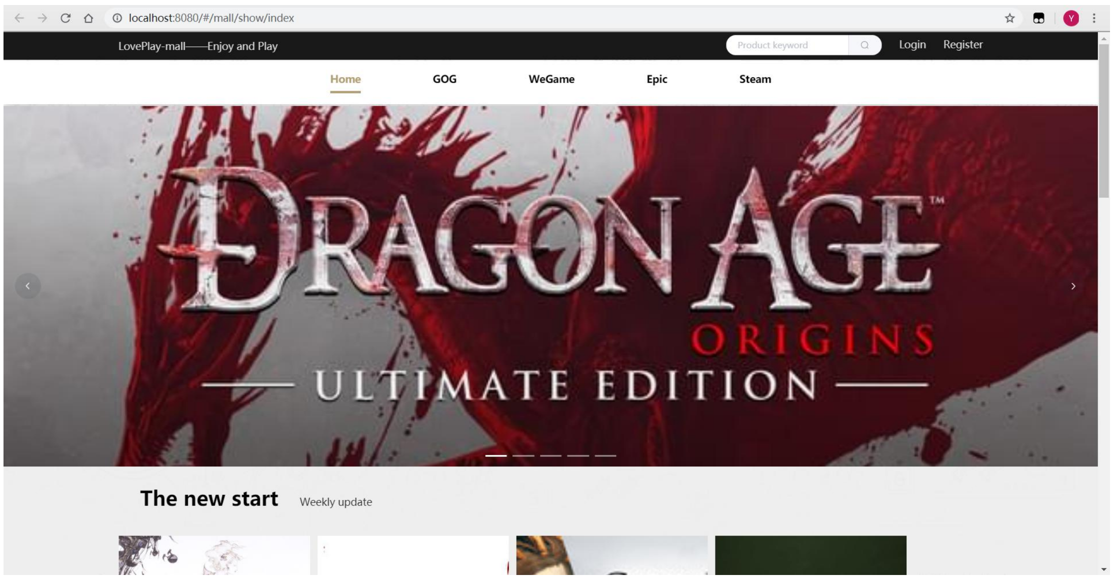
Node.js:
Node.js is a JavaScript runtime environment based on the Chrome V8 engine. Node.js uses an event-driven, non-blocking Istroke O model to make it lightweight and efficient.
Node.js = search engine optimization + first screen speed optimization + front and rear isomorphism
MongoDB:
A Document. based on JSON data model. The main feature is that the JSON data model is suitable for developers, and horizontal expansion can support a large amount of data and concurrency.
Express:
The core function is the routing system: can quickly and steadily enable the server to distribute a request packet to different logical units; focus on high performance; extensive test coverage. Help simplify HTTP operation support. 14 + template engine. Provide powerful scaffolding, through scaffolding to quickly create express applications.
Vue:
Vue is characterized by lighter20KB min+gzip . Progressive framework. Responsive update mechanism. The most important thing is the low cost of learning.

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.

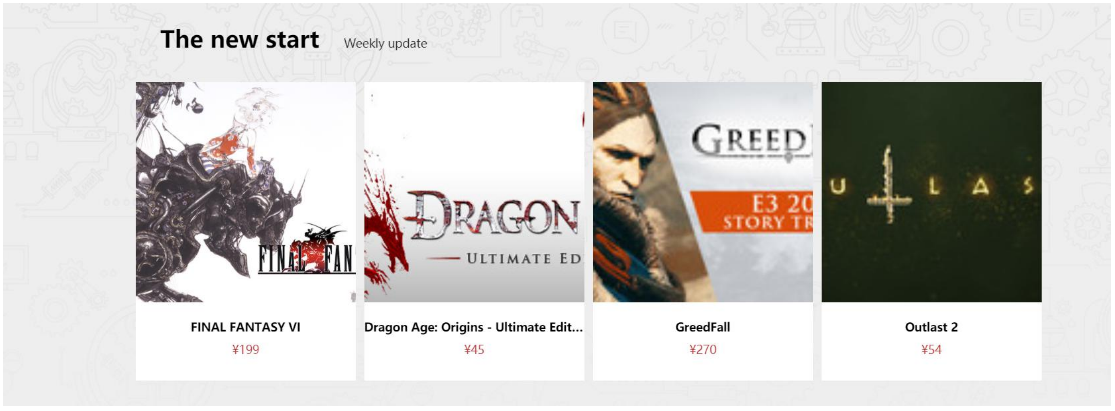
Front end:

Function: home page.

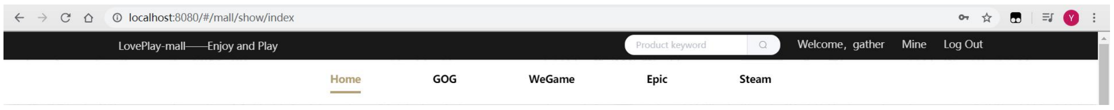

localhost:8080/#/mall/show/index
LovePlay-mall——Enjoy and Play
Product keyword
Login
Register
Home
GOG
WeGame
Epic
Steam
DRAGON AGE ORIGINS ULTIMATE EDITION
The new start
Weekly update

Function: search.

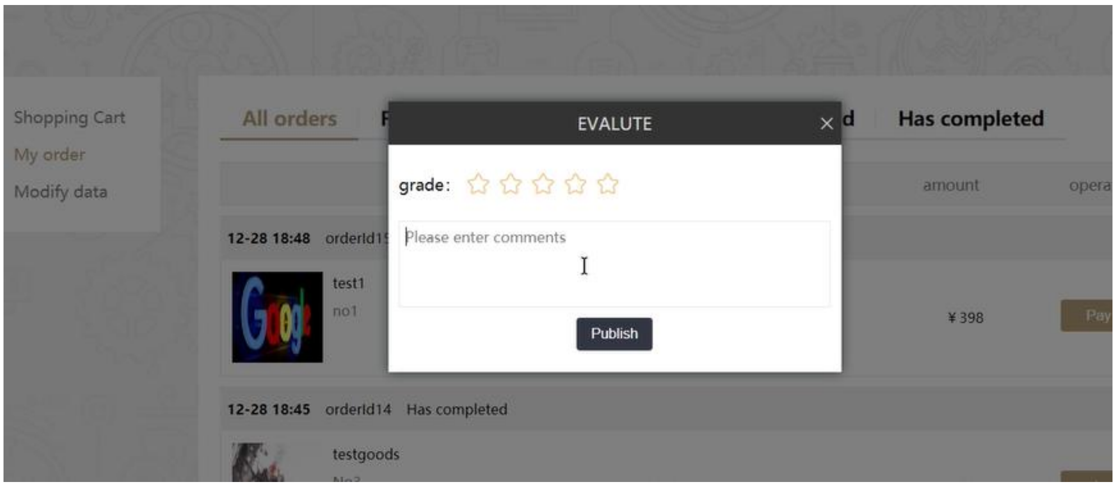
Function: new products on the market.
Reason: update products every week, which can stabilize the source of customers.



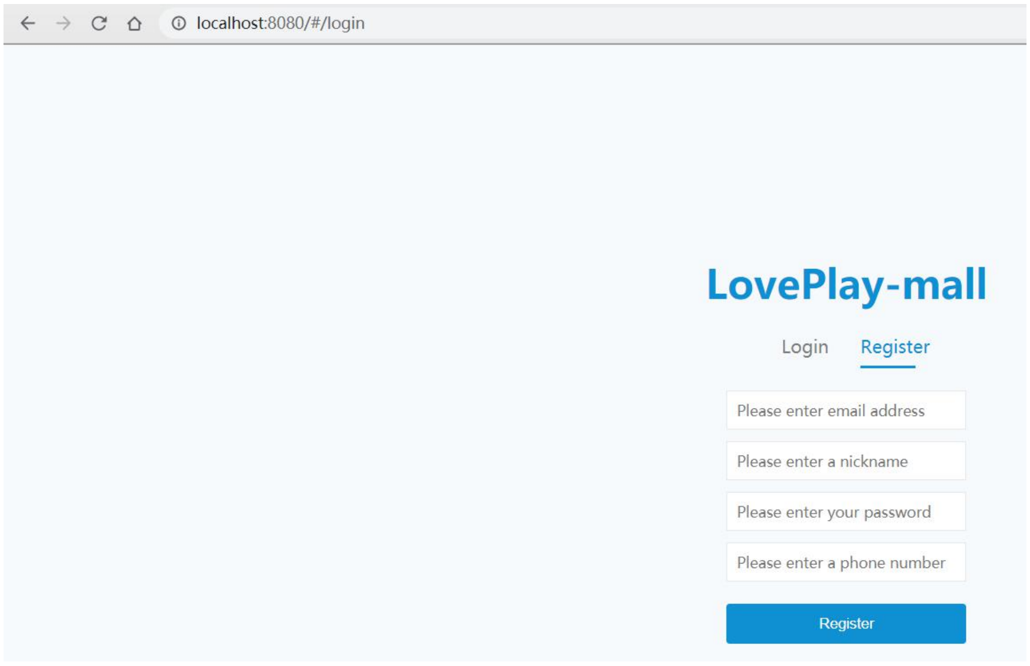
Function: game classification.
Function: it is convenient for customers to browse different kinds of games quickly.



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

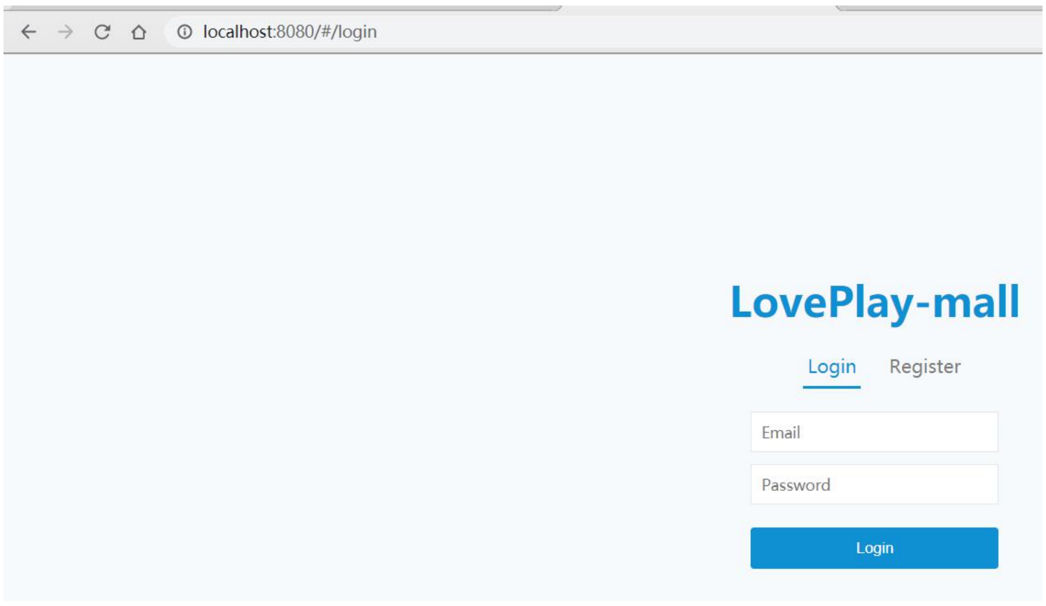


Function: registration.
Reason: user identity information is registered to the application platform


localhost:8080/#/login
LovePlay-mall
Login
Register
Please enter email address
Please enter a nickname
Please enter your password
Please enter a phone number
Register

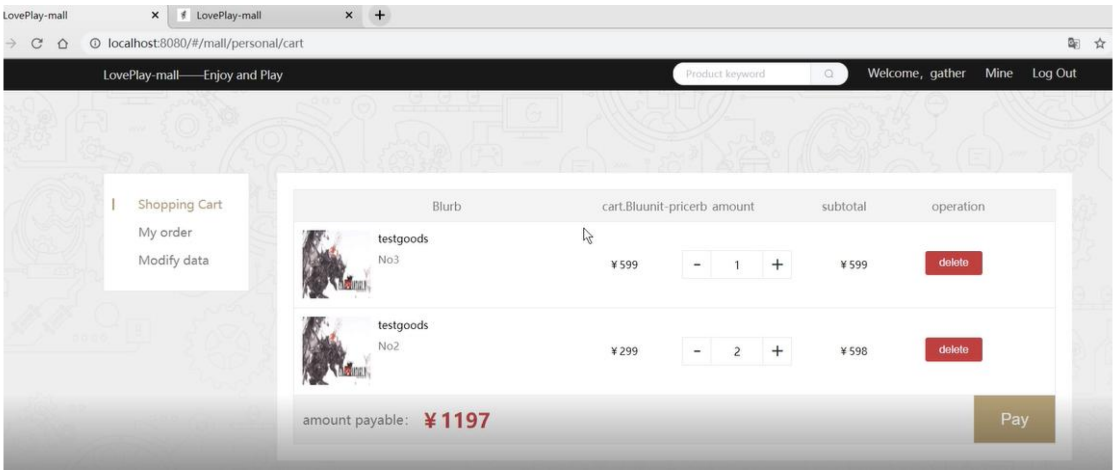
Function: login.

Reason: log in to the verification entry of the website



Function: shopping cart.

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

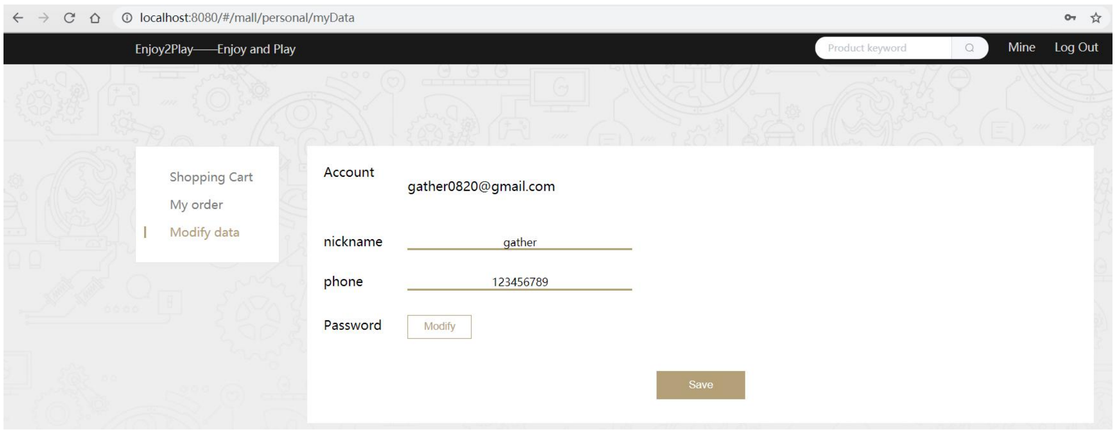


Blurb	cart.Bluunit-pricerb	amount	subtotal	operation
testgoods No3	¥599	1	¥599	delete
testgoods No2	¥299	2	¥598	delete

amount payable: ¥1197 Pay

Function: personal information.

Reason: view and modify personal information


localhost:8080/#/mall/personal/myData

Enjoy2Play——Enjoy and Play | Mine | Log Out

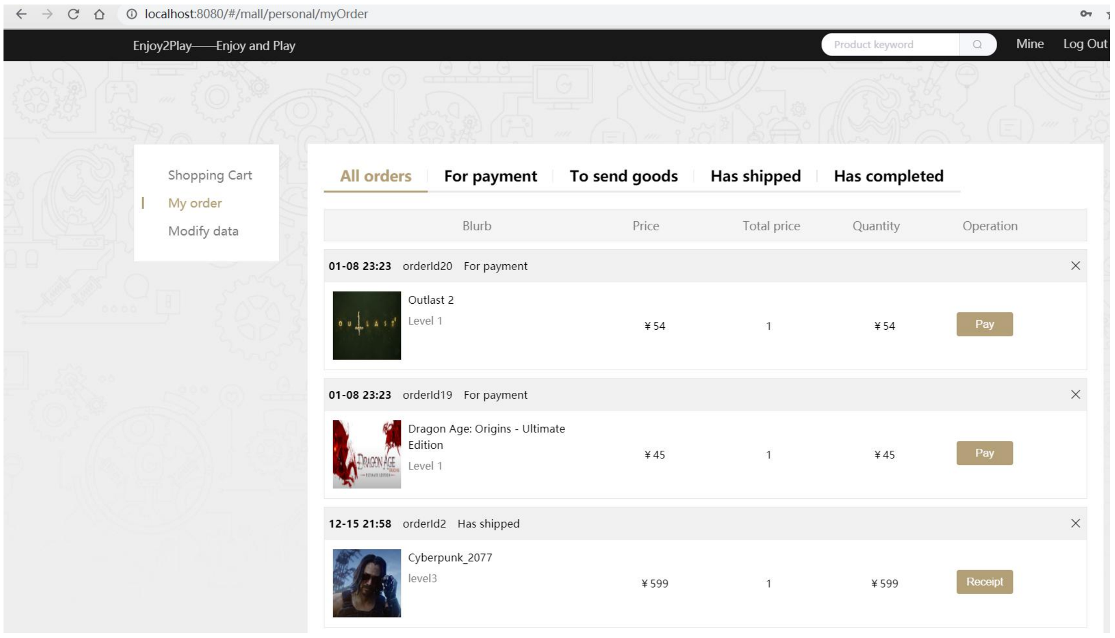
Shopping Cart
My order
Modify data

Account: gather0820@gmail.com
nickname: gather
phone: 123456789
Password: Modify

Save

Function: my order.

Reason: check order status and record order information


localhost:8080/#/mall/personal/myOrder

Enjoy2Play——Enjoy and Play | Mine | Log Out

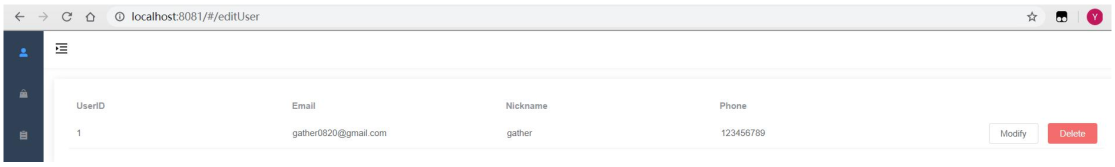
Shopping Cart
My order
Modify data

All orders | For payment | To send goods | Has shipped | Has completed

Blurb	Price	Total price	Quantity	Operation
01-08 23:23 orderId20 For payment				
Outlast 2 Level 1	¥54	1	¥54	Pay
01-08 23:23 orderId19 For payment				
Dragon Age: Origins - Ultimate Edition Level 1	¥45	1	¥45	Pay
12-15 21:58 orderId2 Has shipped				
Cyberpunk_2077 level3	¥599	1	¥599	Receipt

Backend:

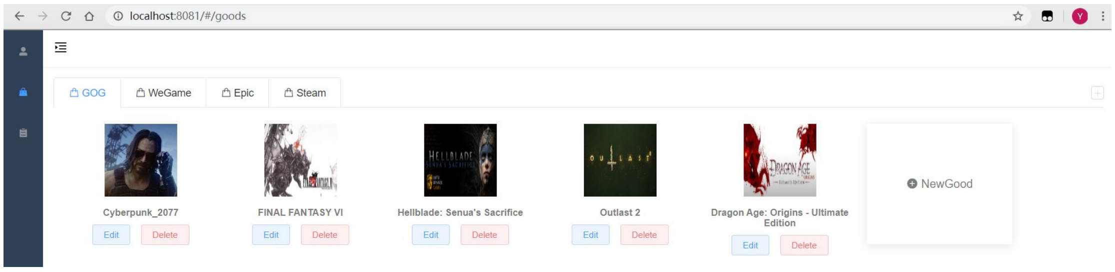
Function: user management.
Reason: you can manage user information, delete and modify query user information


localhost:8081/#/editUser

UserID	Email	Nickname	Phone	
1	gather0820@gmail.com	gather	123456789	Modify Delete

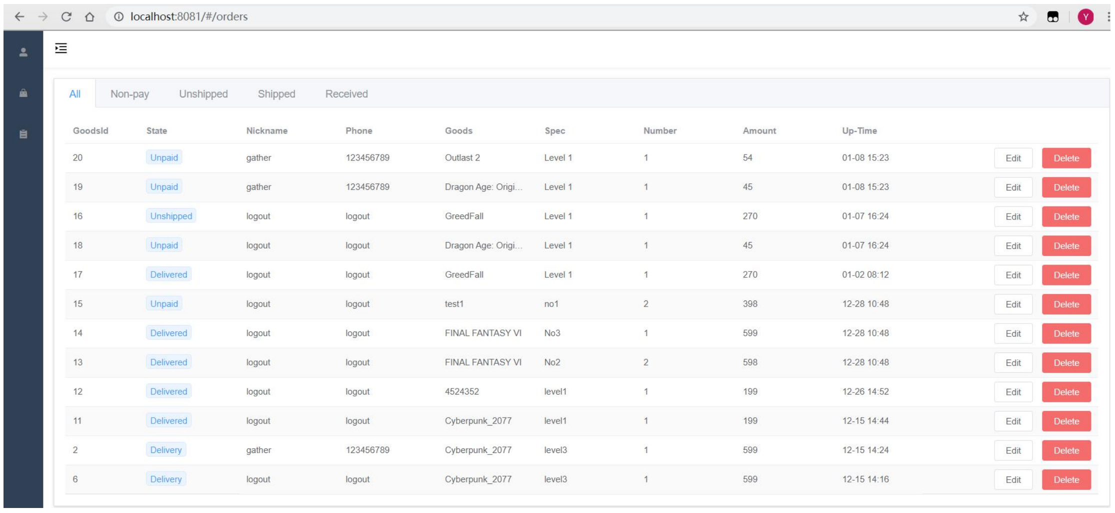
Function: goods on the shelf.
Reason: add game categories, add and modify game details and display them to the

front end.



Function: order list.

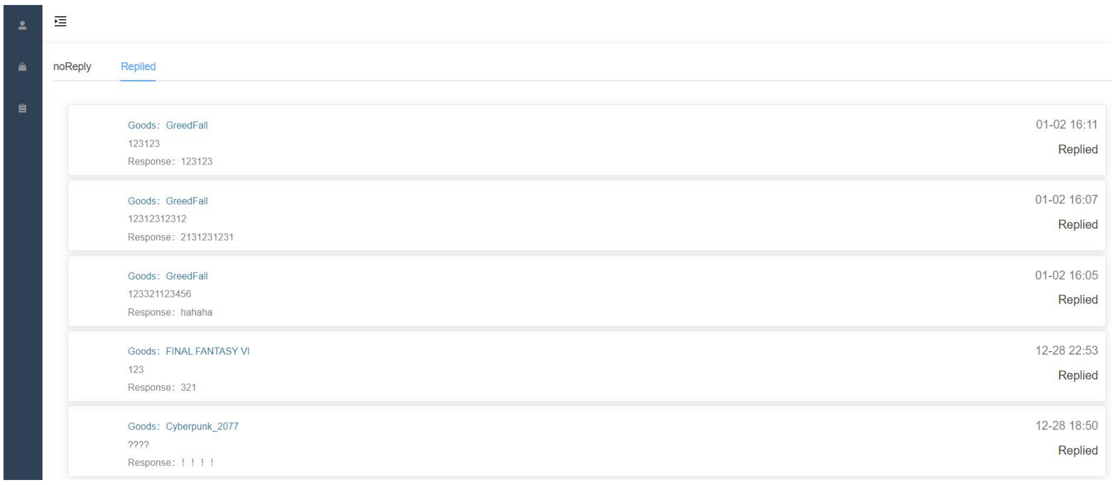
Reason: record the user's order information for delivery and modify the user's order information.



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

Function: Reply to message

Reason:Understand customer needs and answer customer questions



Goods：GreedFall
123123
Response：123123
01-02 16:11
Replied

Goods：GreedFall
12312312312
Response：2131231231
01-02 16:07
Replied

Goods：GreedFall
123321123456
Response：hahaha
01-02 16:05
Replied

Goods：FINAL FANTASY VI
123
Response：321
12-28 22:53
Replied

Goods：Cyberpunk_2077
????
Response：！！！！
12-28 18:50
Replied

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

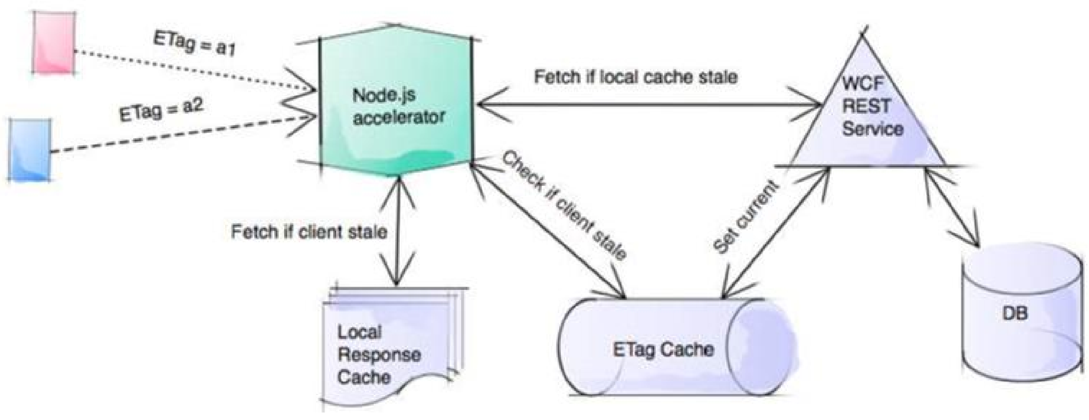
http servers such as apache, tomcat, nginx and iis. The front end of the platform adopts technology stack front end including Vue, Vue-route, element-ui, webpack.
The management side includes: Node, Mongodb database, nodemailer and realize the mail push function based on node.

Requirements

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:

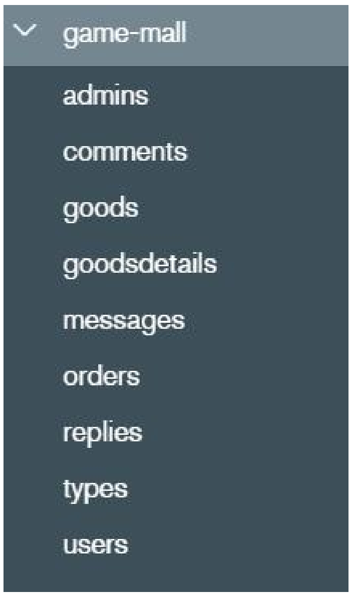


Process interaction

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:

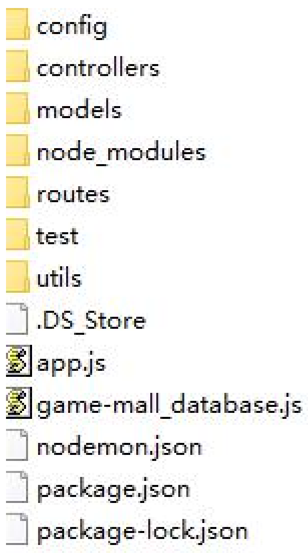


game-mall
admins
comments
goods
goodsdetails
messages
orders
replies
types
users

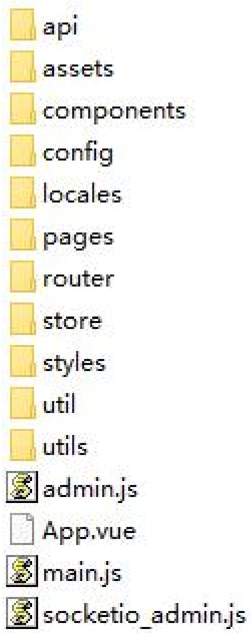
Admin table: responsible for storing the administrator's login information.
User table: responsible for storing user registration and login information.
Goods_ detail table: responsible for storing product details.
Comment table: responsible for storing user comments.
Goods table: responsible for storing commodity information.
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.

Code structure:

Server :

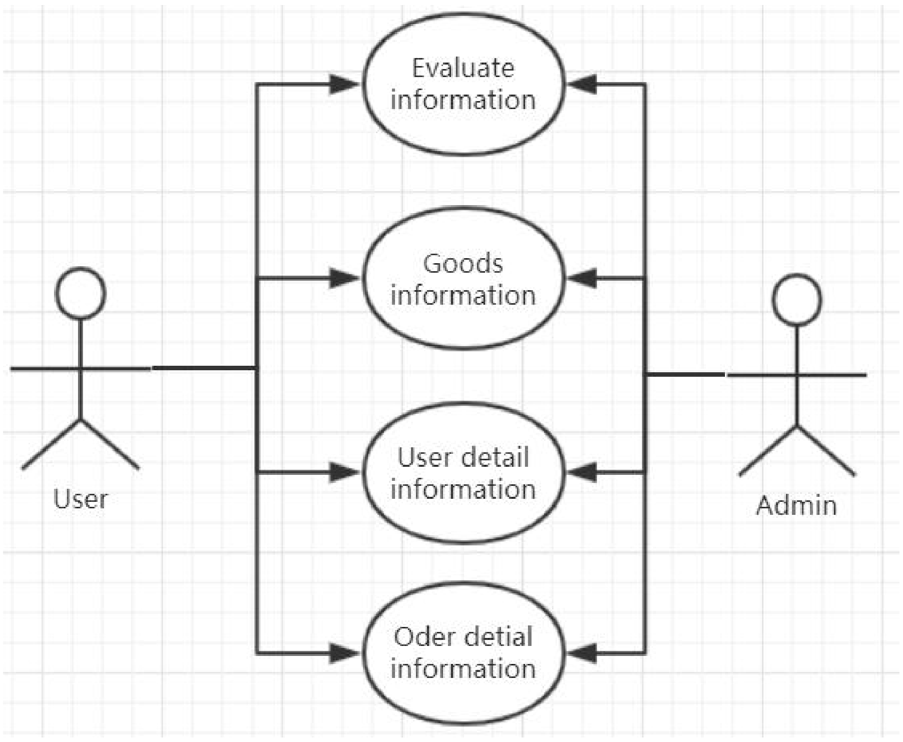

config
controllers
models
node_modules
routes
test
utils
.DS_Store
app.js
game-mall_database.js
nodemon.json
package.json
package-lock.json

Client:

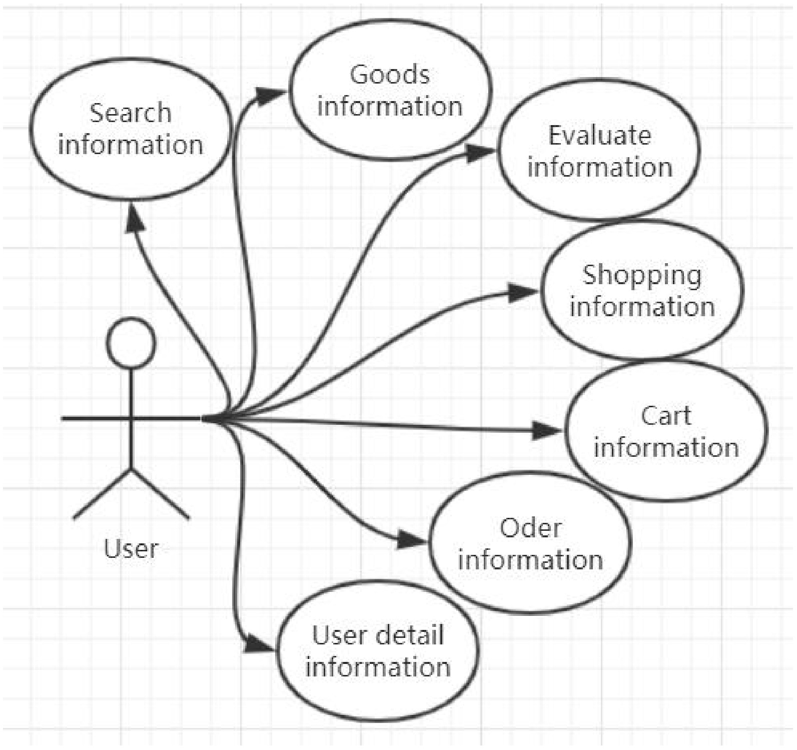

api
assets
components
config
locales
pages
router
store
styles
util
utils
admin.js
App.vue
main.js
socketio_admin.js

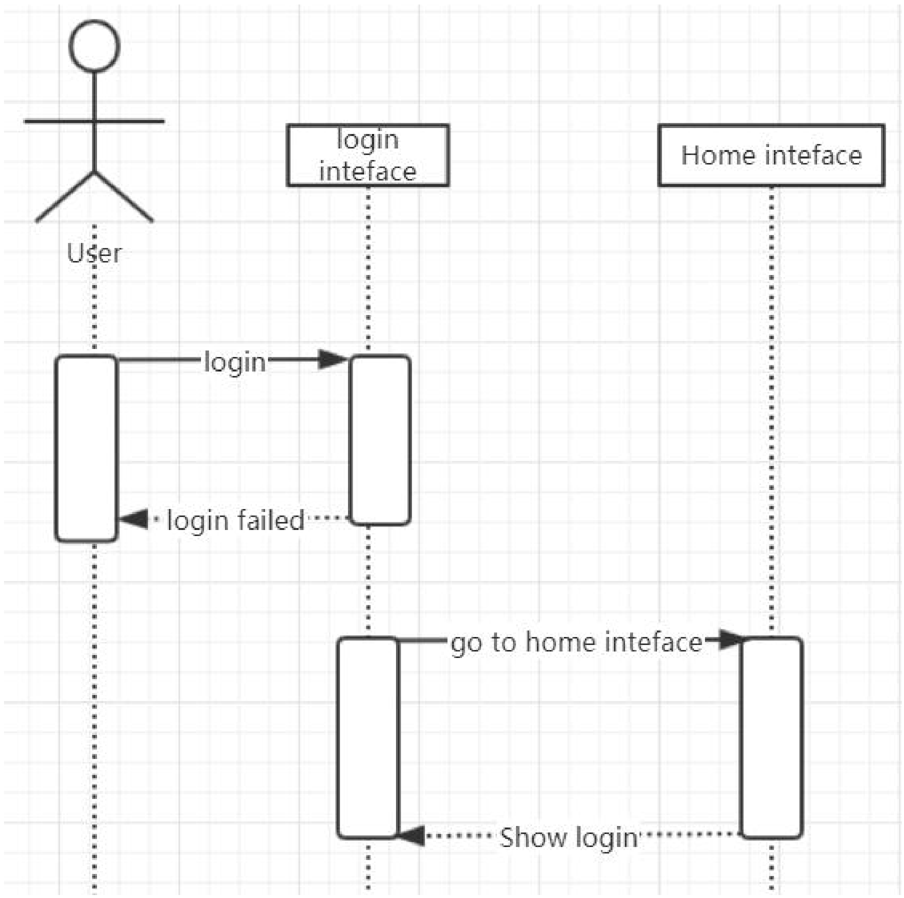
The above code structure is more general and in line with the standard development specifications, which is more conducive to later code maintenance and functional improvement.

System UML use case diagram：

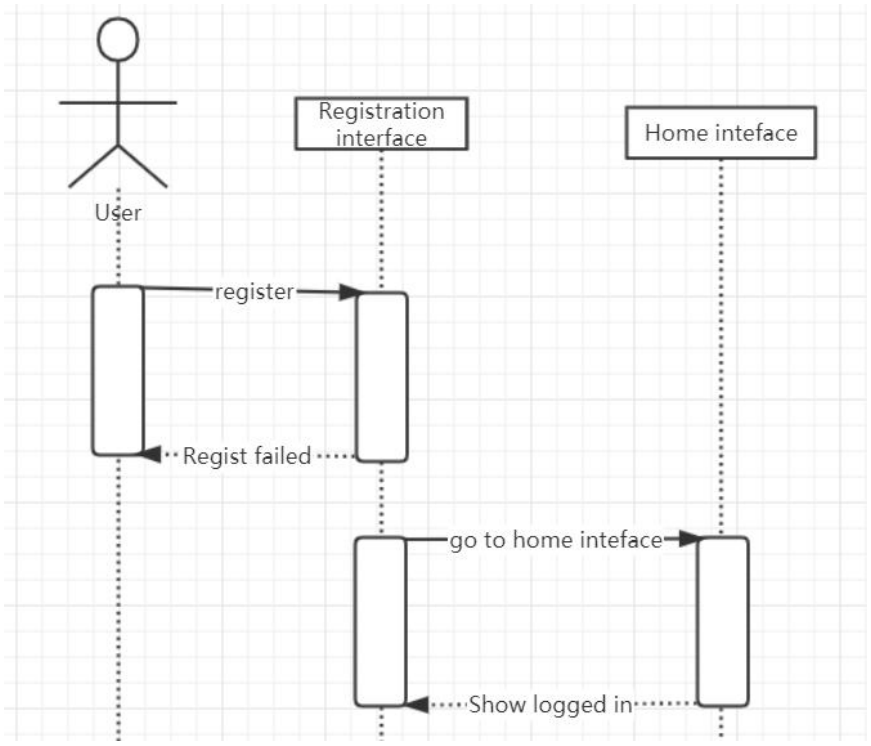


User UML use case diagram：

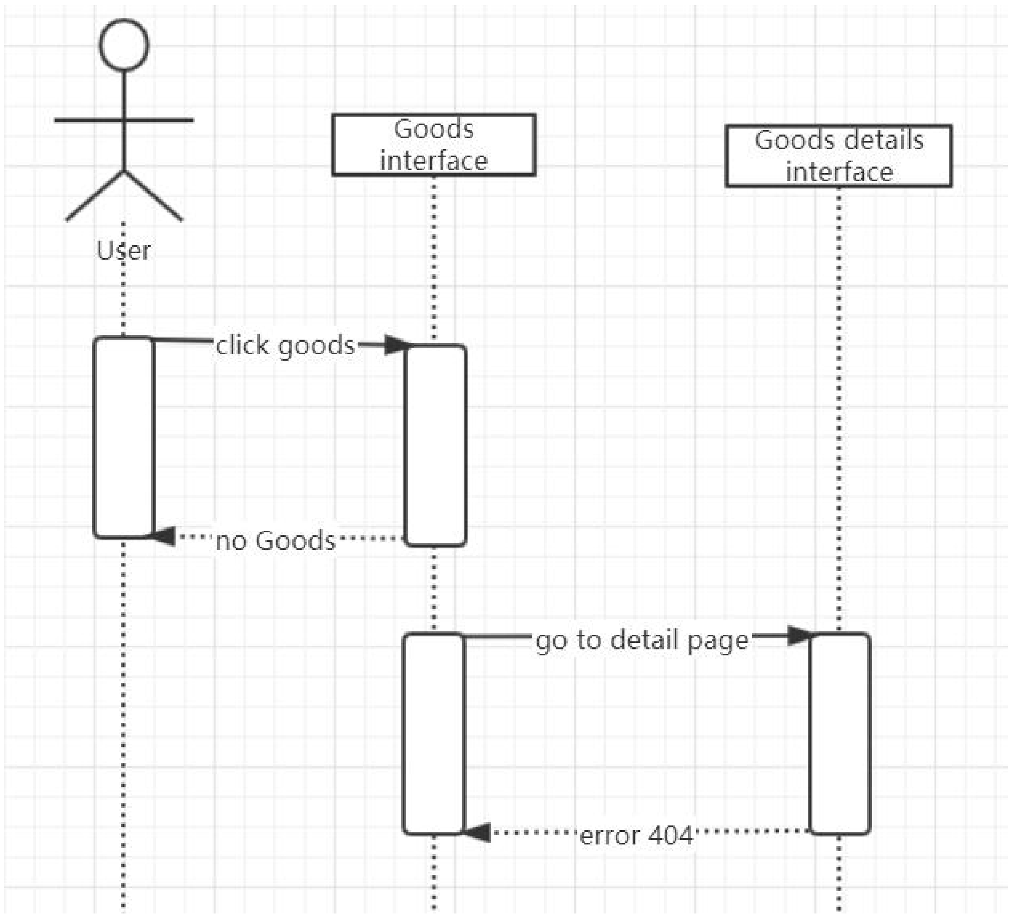




(Login)

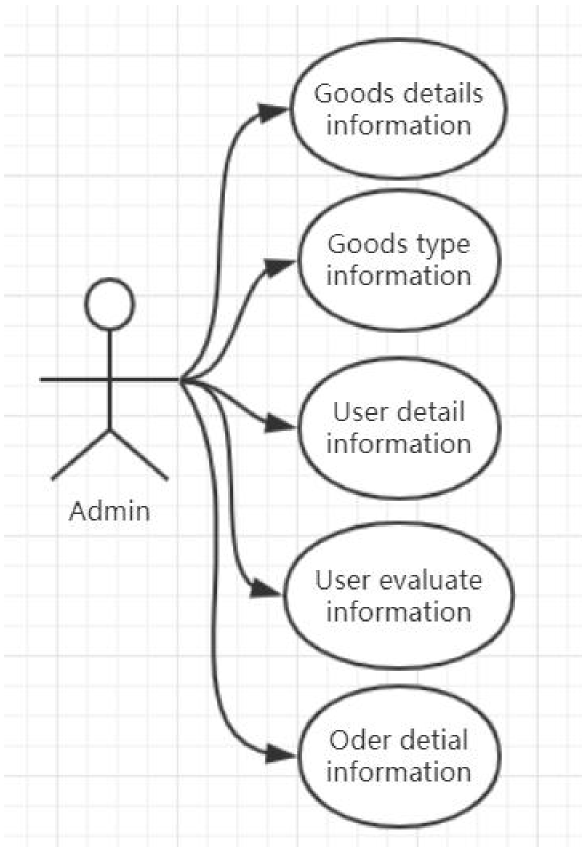


(Register)



(Go to the goods page

Admin UML use case diagram:



Testing

Test:

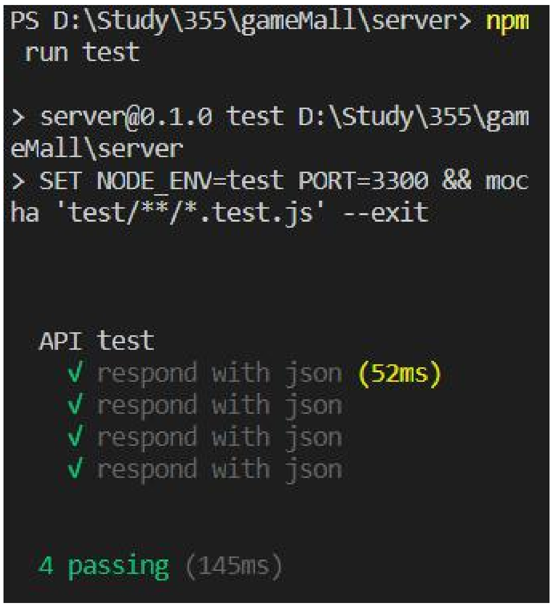
Perform automated text testing: use the automatic script function of comparing text and copy data to see if there is a text error message.
Test results: pass

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

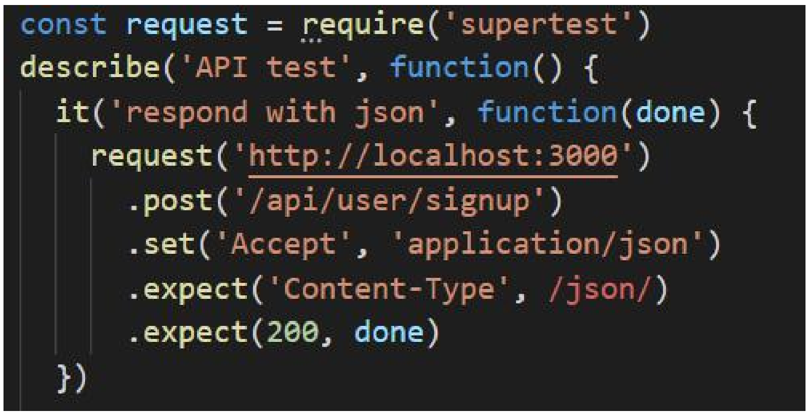
Perform usability testing of the shopping process: the most important process of the website is the shopping process: shopping cart, order reminder, order submission, unit testing.
Test results: pass

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%.



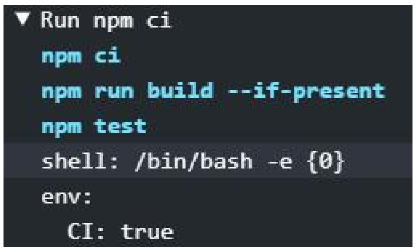
```
PS D:\Study\355\gameMall\server> npm
 run test

> server@0.1.0 test D:\Study\355\gam
eMall\server
> SET NODE_ENV=test PORT=3300 && moc
ha 'test/**/*.test.js' --exit



  API test
    √ respond with json (52ms)
    √ respond with json
    √ respond with json
    √ respond with json


  4 passing (145ms)
```



```
const request = require('supertest')
describe('API test', function() {
  it('respond with json', function(done) {
    request('http://localhost:3000')
      .post('/api/user/signup')
      .set('Accept', 'application/json')
      .expect('Content-Type', /json/)
      .expect(200, done)
  })
```

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



```
▼ Run npm ci
  npm ci
  npm run build --if-present
  npm test
  shell: /bin/bash -e {0}
  env:
    CI: true
```

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.