**PROJECT:**

**Distributed online instant messaging system**

**Link: *123.206.62.224:8080***

|  |  |  |
| --- | --- | --- |
| NAME | TUT NUMBER | UQAC NUMBER |
| YANAN SONG | 20152578 | SONY03129604 |
| LINGFENG ZHANG | 20152649 | ZHAL17099706 |
| ZIQI JIN | 20152727 | JINZ14119602 |
| TIANYI GONG | 20152632 | GONT21079708 |

**Code Source:** ***https://github.com/syn326/vuechat***

**Content:**

1. introduction
2. roles’ responsibility
3. functions’ illustration
4. Plan to do and things achieved
5. my design
6. running samples
7. how to build up our project
8. API
9. technology illustration
10. conclusion
11. remarks
12. references

**Introduction:**

Our team has finished an online instant messaging system which can provide users with instant online chatting as well as leaving messages by using a variety of technologies such as Vue.js, Vue Router, Vuex, Axios, webpack ,vue-cli, Node.js, Express, CORS, Socket.IO ,MySQL, MD5 and Sass.

**Roles’ responsibility:**

1. YANAN SONG:
   1. revising LAB-ONE
   2. checking LAB-TWO
   3. Writing ‘PROJECT’: writing messages interacting (chatting part) & function of adding and deleting friends.
2. LINGFENG ZHANG:
   1. writing LAB-ONE
   2. testing LAB-TWO
   3. Writing ‘PROJECT’: writing messages Board part & report.
3. ZIQI JIN:
   1. revising LAB-ONE
   2. writing LAB-TWO
   3. Writing ‘PROJECT’: writing login page & register page.
4. TIANYI GONG:
   1. revising LAB-ONE
   2. writing LAB-TWO
   3. Writing PROJECT: writing ‘change theme’ part & ‘check My-Info’ part.

**Functions’ illustration:**

1. Login function: you can input your id and password to enter chatting room.
2. Register function: if you have not had one id yet, you would register one.
3. Add new friends function: you can search for new online friends according to ‘id’.
4. Delete friends function: you can delete your friends if you want to do so.
5. Leave messages function: you can leave messages to the public on the messages board.
6. Chatting function: you can chat with your friends instantly.
7. Check my information function: you can check your information.
8. Change Theme function: you can change the color of chatting room background according to your preference.

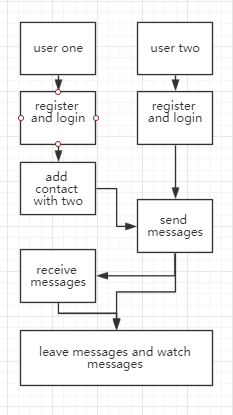
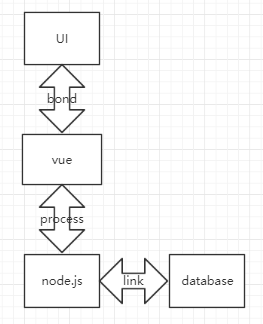
**Plan to do and things achieved:**

According to course-Distributed System, our team wanted to implement a chatroom by utilizing some technologies related to DS. If we want to implement DS, we will learn how to process cross origin, because we have to distribute front end and rear end to different servers. So we used CORS(mentioned later) to achieve it. To implement DS, we also used vue.js(focused on ViewModel layer of MVVM pattern),which connects the View and Model via two way data bindings, to dispatch logics from UI, both of which were connected by API. We also used vue-router to relocate pages in the front end rather than in the rear end. For communication between front end and rear end, we utilized Express middleware to request and response messages. Of course, there were XML transmissions according to Axios requesting HTTP. Overall, we achieved our goal of implementing DS in the project, finishing distributed online instant chatroom.

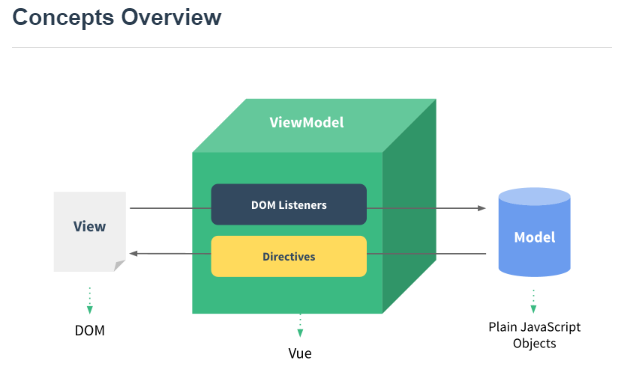
**My design:**

In this project, we strived for some functions of chatroom, such as adding friends, leaving messages, chatting and so on.

This is the flow chart of how to process & Tier model:

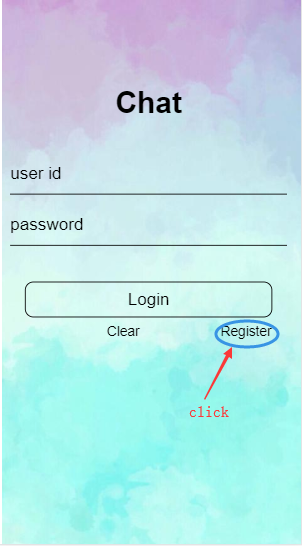
 

Vue.js concepts overview:

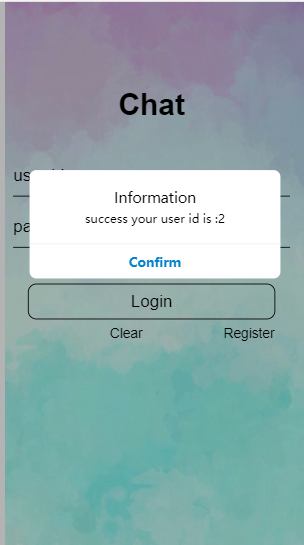
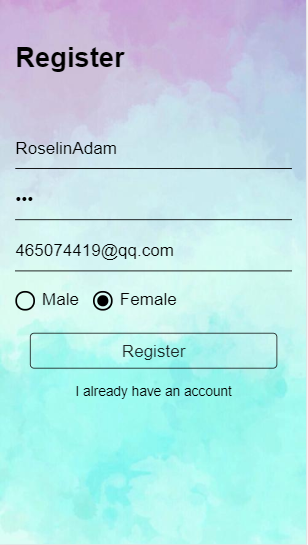


**Running samples:**

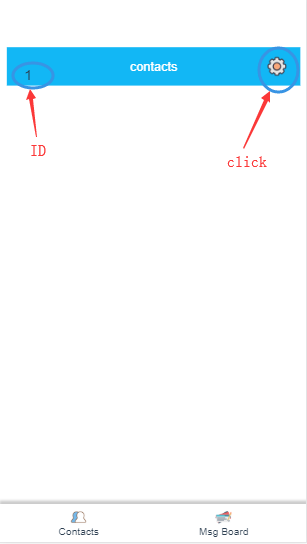
Firstly, you can access to the index page and click the ‘Register’ to register you own account if you haven’t had one before.



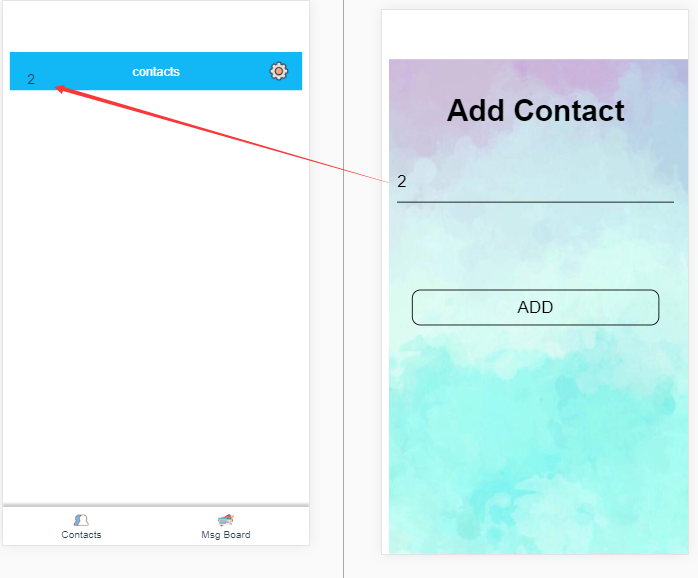
Secondly, you can fill up the information and have your own account .For example, your account is ‘2’, password is ‘234’. You can access it by yourself.



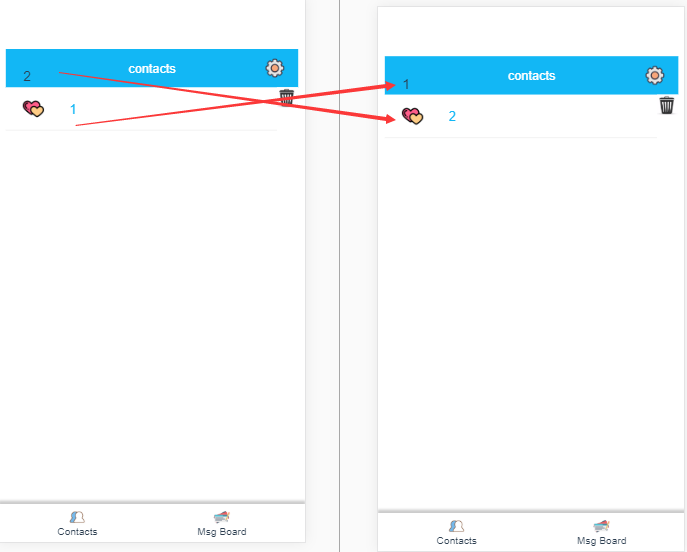
Then, you can login into your contacts page. You can register another account and login again in different pages, simulating two persons’ instant talk. Another one’s account is ‘1’ and password is ‘123’.After that, you will enter in your own page. You can just click the wheel-shaped button and then some function buttons will be shown. We chose ‘add contact’.



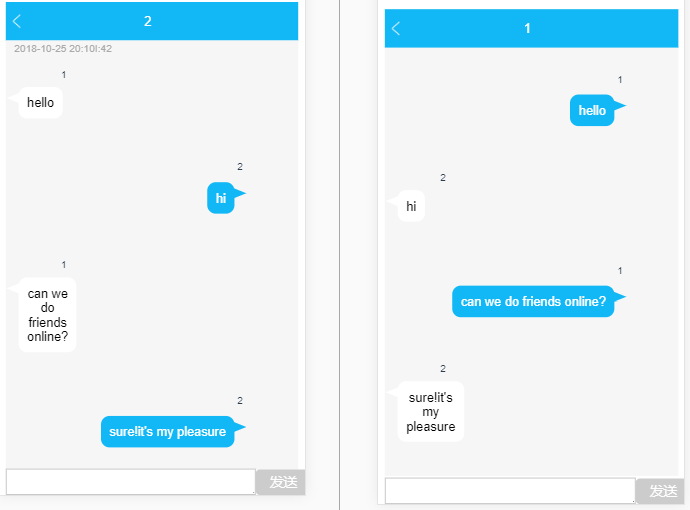
Then you just input an account number as well as click ‘ADD’ to add a new friend online.



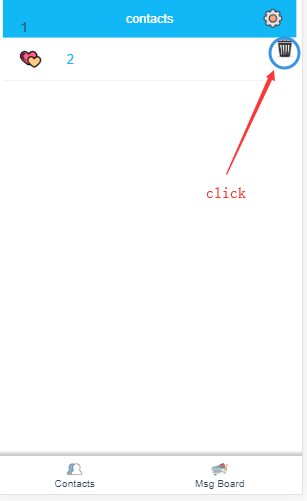
Of course, you have to refresh your page to show your new online friends because we couldn’t show the new friends directly without a refresh.



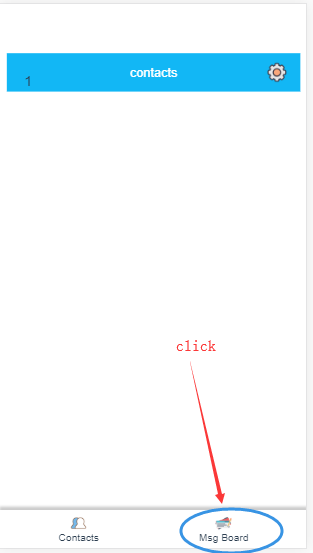
At the next step, we click one friend and send some messages to contact with him or her. It is instant online contacts. Both of different users can receive and send messages synchronously.



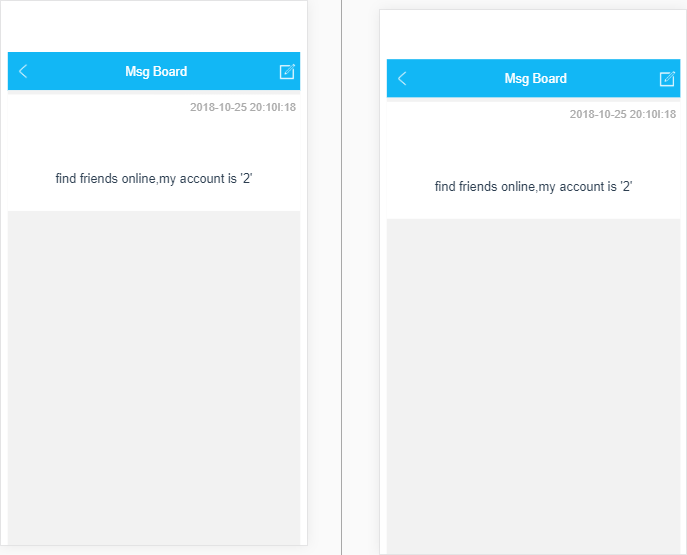
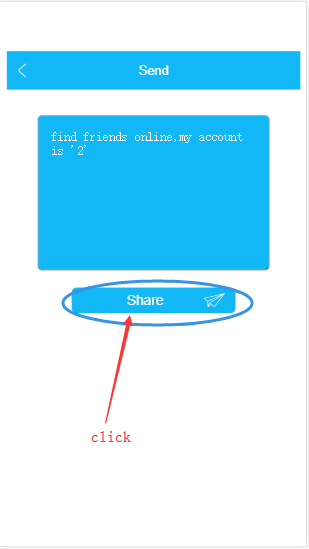
Of course, you can delete your friends if you do not want to chat with him or her. The method is clicking the garbage-bin-shaped button and you can delete them after refreshing. At the same time, your friends will also delete you automatically and synchronously, of course, also after refreshing pages.



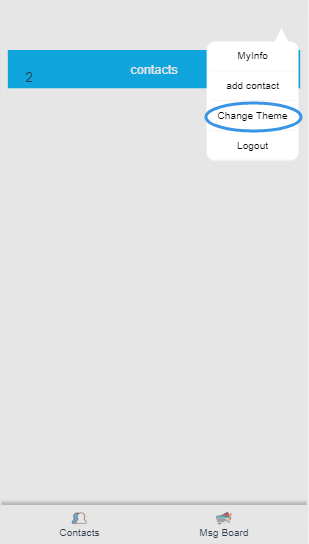
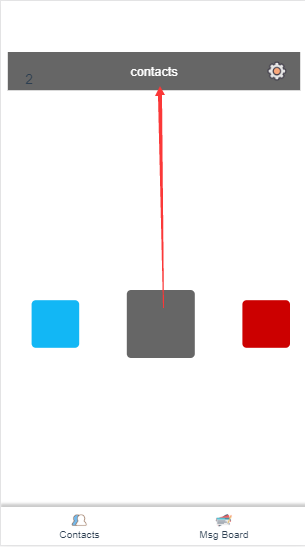
We also have another function: leaving messages in the Message Board. After you leaving some messages, all accounts have the right to watch it, which is just like a public forum. We only click the ‘Msg Board’ and then locate in the message board. If you want to write something or leave some messages on it, you can just click the note-shaped button and then edit your messages.



After finishing your editing, you can just send it to the public. As the consequence, all accounts can watch your leaved messages.



Of course, our chatting room is customized. We can change the theme (the color of the chatting room) according to wheel-shaped button as well as choosing ‘change theme’. At the next stage, you just choose your favorite color, and we provide you with three kinds of color: blue, grey and red.

**How to build up our project:**

1. Change ‘123.206.62.224’ to your IP at line 2 in the file vuechat/src/api/server.js.
2. Enter in the direction of our project by using terminal
   1. cd \*\*
   2. install node.js environment
3. Front-end environment:
   1. npm install
   2. npm run dev
4. Rear-end environment:
   1. npm install express
   2. npm install http
   3. npm install socket.io
   4. npm install cors
   5. npm install mysql
   6. npm install moment
   7. node server.js
5. Enter your IP(eg.localhost)+port:8080 in browser

**API:**

1. link front end to rear end:

const Server and then export default Server in src/api/server.js

1. link rear end and database:

mysql.createConnection in server.js

**Technology illustration:**

1. Vue.js: Vue is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries.[1]
2. Vue Router: Vue Router is the official router for Vue.js. It deeply integrates with Vue.js core to make building Single Page Applications with Vue.js a breeze. It can manage different routes and enable pages to show the direction file correctly.[2]
3. Vuex: Vuex is a state management pattern + library for Vue.js applications. It serves as a centralized store for all the components in an application, with rules ensuring that the state can only be mutated in a predictable fashion.[3]
4. Axios: Axios is a HTTP library which is based on the promise(an asynchronous resolution in JavaScript) and which can be used in Node.js. There are some features it included: create the XMLHttpRequests in browsers, create http requests from node.js, support Promise API, and so on.
5. webpack: webpack is a *static module bundler* for modern JavaScript applications. When webpack processes your application, it internally builds a dependency graph which maps every module your project needs and generates one or more *bundles*.[4]
6. vue-cli: A simple CLI for scaffolding Vue.js projects[5]
7. Express **middleware**: Express is a routing and middleware web framework that has minimal functionality of its own: An Express application is essentially a series of middleware function calls. Middleware functions are functions that have access to the request object (req), the response object (res), and the next middleware function in the application’s request-response cycle. The next middleware function is commonly denoted by a variable named next.[6]
8. Cross-Origin Resource Sharing(CORS) is a mechanism that uses additional HTTP headers to tell a browser to let a web application running at one origin (domain) have permission to access selected resources from a server at a different origin. A web application makes a cross-origin HTTP request when it requests a resource that has a different origin (domain, protocol, and port) than its own origin.[7]
9. Sockey.IO: Socket.IO is a library that enables real-time, bidirectional and event-based communication between the browser and the server.[8]
10. Sass(Syntactically Awesome StyleSheets): Sass is an extension of CSS that adds power and elegance to the basic language. It allows you to use variables, nested rules, mixins, inline imports, and more, all with a fully CSS-compatible syntax. Sass helps keep large stylesheets well-organized, and get small stylesheets up and running quickly, particularly with the help of the Compass style library.[9]
11. MD5(message-digest algorithm): The MD5 message-digest algorithm is a widely used hash function producing a 128-bit hash value. MD5 was initially designed to be used as a cryptographic hash function and it can still be used as a checksum to verify data integrity.[10] We use it to encrypt password.

**Conclusion:**

In this project, our team members finished it cooperatively. We used middleware to implement distributed communication. Our project is a typical 3-tier project: User Interface (client front end), server and database.

**Remarks:**

You can download our source code in GitHub:

<https://github.com/syn326/vuechat>

You can access to our server by IP:PORT to use our project directly:

123.206.62.224:8080

**Reference:**

1. <https://vuejs.org/v2/guide/index.html>
2. <https://router.vuejs.org/?tdsourcetag=s_pctim_aiomsg>
3. <https://vuex.vuejs.org/>
4. <https://webpack.js.org/concepts/>
5. <https://www.npmjs.com/package/vue-cli>
6. <http://www.expressjs.com.cn/en/guide/using-middleware.html>
7. <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>
8. <https://socket.io/docs/#What-Socket-IO-is>
9. <https://sass-lang.com/documentation/file.SASS_REFERENCE.html>
10. <https://en.wikipedia.org/wiki/MD5>