

# COMP3030J Software Engineering Project 2020-2021

## Group2 Doraemon

Ningxin Li 19206173  
Xuan Wu 19206179  
Yilan Wang 19206190  
Peixuan Dong 19206154  
Tingjun Wu 19206159  
Xuran Zhao 19206158

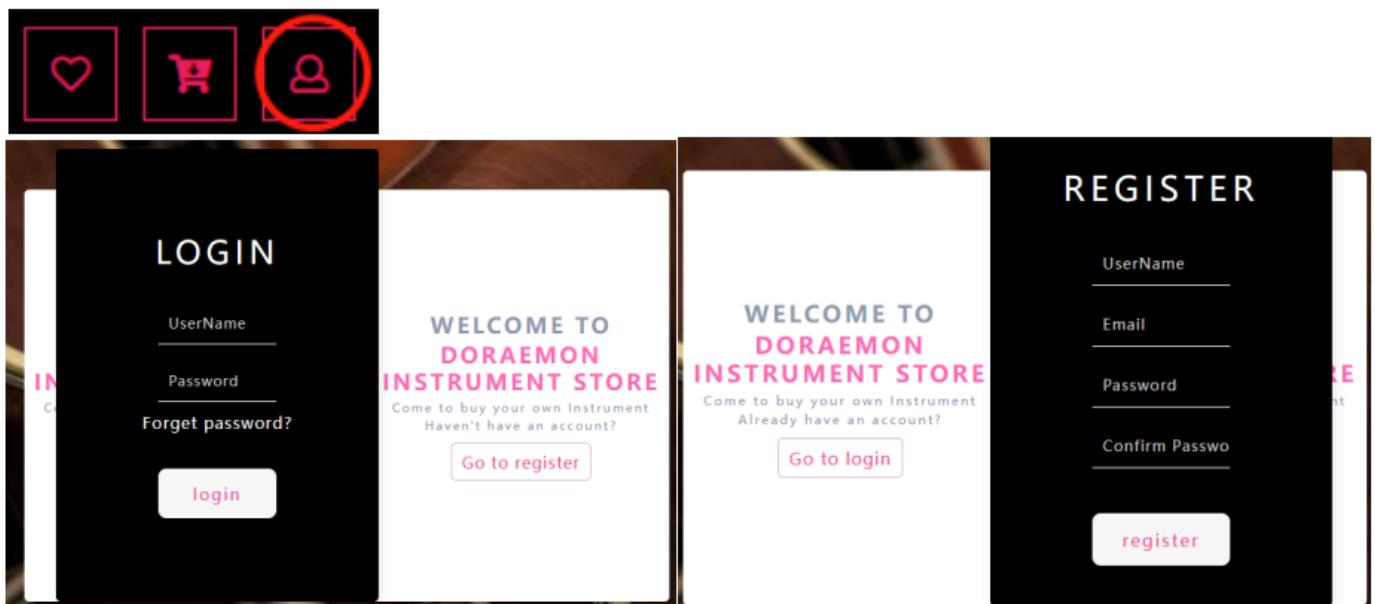
November 1, 2022

## 1 User documentation

Server address: <http://193.1.133.149:5002/>

### Login and register:

As a customer you can go to the login page by clicking the picture button in the upper right corner of the homepage, as shown in the following figure.



On this page you can enter a username and password to log in if you already have an account. If you don't have an account, click register on the right and the page will turn into a registration page. Enter the username, email and password you want to register. (Provide customer accounts:username1 password:A000000 ) As the staff in this page you can only login function, staff account is only provided by the developer, can not be registered. (Provide admin accounts:admin password:A123456)

### Forgot password:

If you forget your password, you can click Forget password. Then you need to enter your user name and the email address you signed up for and get a verification code from that email address. Then you need to enter a new password and re-enter the password to confirm it. If all the above inputs are correct the password will be changed.(Provide customer accounts:username1 password:A000000 email:test.doraemon@outlook.com email password:WASD13579) Modify Reason:After feedback from other groups, they believe that the account has security risks, and the account will be lost if you forget the password, so we added the function of forgetting the password.

# CHANGE PASSWORD

User Name

Email

Verification code

Get Code

New Password

New Password Again

Submit

Translate:

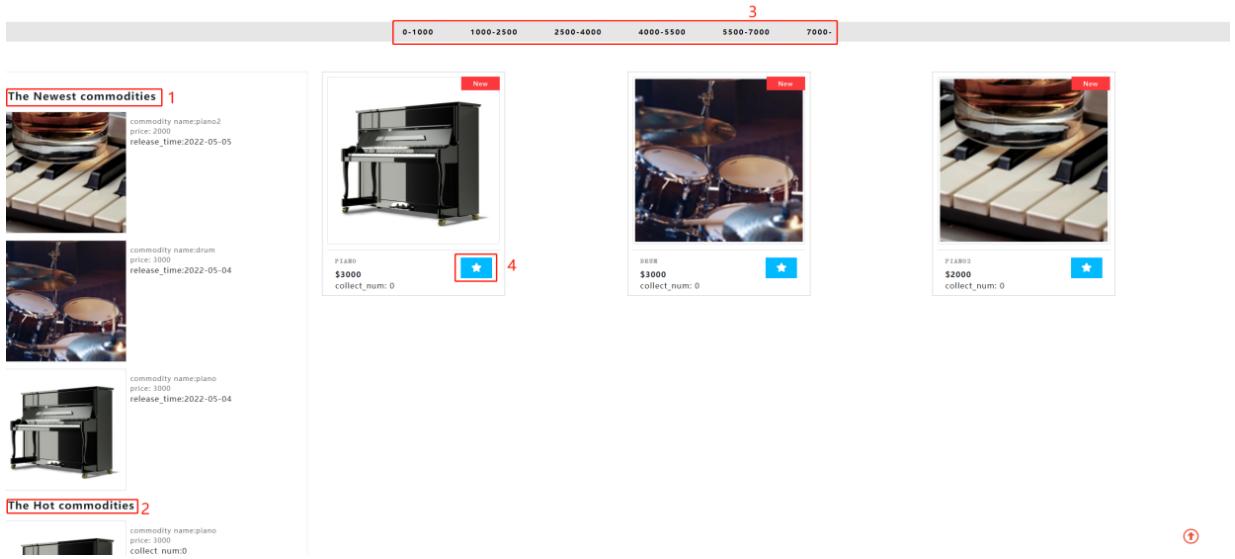
As shown in the above figure, click the place pointed by the arrow to realize the translation function of the page.  
Modify Reason: We have adopted the suggestions in the feedback, so that the web page can save the translation status, and at the same time, we can translate the contents in the database.

## 1.1 Customer



Click shop Now to see the image below. Modify Reason: To make the description clearer.

## Instruments Shop



### **The Newest commodities:**

In this column, the latest products will be displayed, as shown in sign 1 in the figure above.

### **The Hot commodities:**

The most collected items are displayed in this column, as shown in sign 2 in the figure above.

### **Price range:**

Click on different ranges to display items belonging to this range, as shown in sign 3 in the figure above.

### **Collect:**

As shown in sign 3 in the figure above, when browsing various commodities in the store page, customers can click the five-pointed star button in the commodity bar to collect, and the collected commodities can be viewed in the personal information page. Clicking the five-pointed star button again will cancel the collection.

### **Product details page:**

This screenshot shows a detailed view of a piano product. The top navigation bar includes 'Home // Single Page', 'DESCRIPTION', and 'REVIEWS'. The main content area features a large piano image, its name ('Piano'), price ('3000'), and stock level ('stock: 100'). It also includes a play sound button and a quantity selector ('Number : 1'). Below these are 'Add To Cart' and 'Buy Product' buttons. To the right, there's a review section with fields for 'TITLE' and 'YOUR COMMENT ...', and a 'SEND' button. Red arrows numbered 1 through 4 point to specific interactive elements: 1 points to the play sound button; 2 points to the quantity selector; 3 points to the 'Buy Product' button; and 4 points to the 'SEND' button in the review form.

When the customer clicks the commodity icon to enter the commodity details page, the interface will appear. The customer can view the picture, name, quantity and selling price of the commodity in this interface.

### **Play music 1.:**

After the customer clicks the play music icon, the website will play the music played by the instrument.

### **Add to cart 2.:**

After selecting the quantity you want to buy, click the add to cart button, the selected goods and the selected quantity will be saved in the customer's shopping cart, and the page will also jump to the shopping cart page.

My Shopping Cart					
Select All	Items	Unit price	Number of units	Sum	Operations
<input checked="" type="checkbox"/>	piano	3000	<input type="button" value="2"/>	6000	<input type="button" value="Delete"/> 2. (1)
<input checked="" type="checkbox"/>	drum	3000	<input type="button" value="6"/>	18000	<input type="button" value="Delete"/>
<input type="checkbox"/> Select All	<input type="button" value="Delete All"/> 2. (2)	items selected: 8 pcs	Total: \$ 24000.00	<input type="button" value="Pay"/> 2. (3)	

After the customer enters the shopping cart page, the interface will appear. In the shopping cart interface, customers can see the goods added to the shopping cart, the quantity of goods, and the total price of goods purchased.

#### Shopping cart delete single category goods 2.(1):

If customers want to remove a single kind of goods from the shopping cart, they can click the delete button on the right side of the line of such goods to remove such goods.

#### Delete all items from shopping cart: 2.(2):

If customers want to delete all items in the shopping cart, they can click the delete all button to remove all items.

#### Shopping cart to buy goods 2.(3):

If customers want to buy some goods in the shopping cart, they can select the goods and click the pay button. In the confirmation interface, click the pay button again to confirm the payment and jump to the purchase interface.

#### Purchase interface 3:

After selecting the quantity you want to buy, if you want to buy goods directly, you can click the buy product button to save the goods and purchase quantity information and enter the purchase interface.

**Order Details:**

Product: Piano-YDP144B

Number: 1

Delivery address: No. 100, Pingyuan park

Phone number: 13738597364

Consignee name: Jack

Transport: delivery home

#### Confirm purchase:

After jumping to the purchase page, the customer can input his receiving address, telephone number, recipient name and other information according to his own situation, and click confirm pay to complete the purchase.

#### Confirm purchase:

If the customer wants to cancel the purchase of the product, he can click the cancel pay button to end the purchase.

#### Comment 4.:

After the customer logs in and enters in the product details page. At the bottom of the page, customer will see the place where the comments are posted. Enter the title and content of the comments, and click the send

button to post the comments.Modify reason: after checking, we found that the comment part was missing on overleaf, and we added it again.

### Order list and order details page:

Customers can click the order button in the personal interface to enter the order page.In the order list interface, click the order to jump to the details page of the order. On the order details page, customers can see the product name, unit price, purchase number, delivery address and other information of the order.

### Edit orders 1.:

Customers can edit details of their own orders, include delivery address, phone number, consignee name and transport way they want (delivery home or pick up at the store). But can only edit before “Signed in”.

### Receive order 2.:

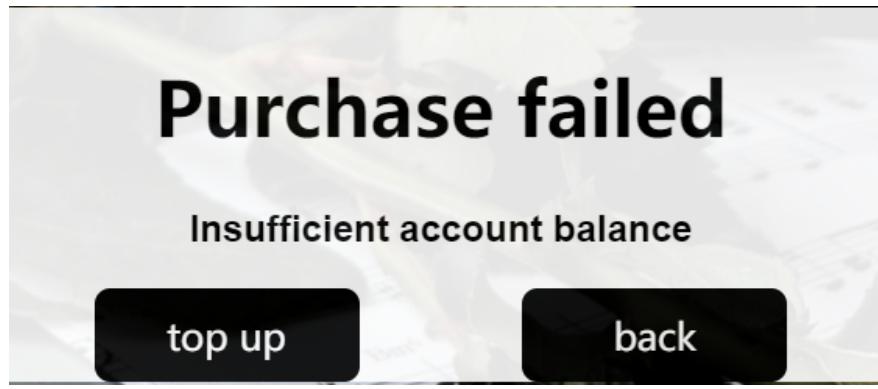
When the order status is delivered, the customer will see the confirm receive button on the order details page interface. After the customer successfully receives the goods, click the button to change the status of all orders of the order number to signed in, and the button disappears.

### Cancellation of order 3.:

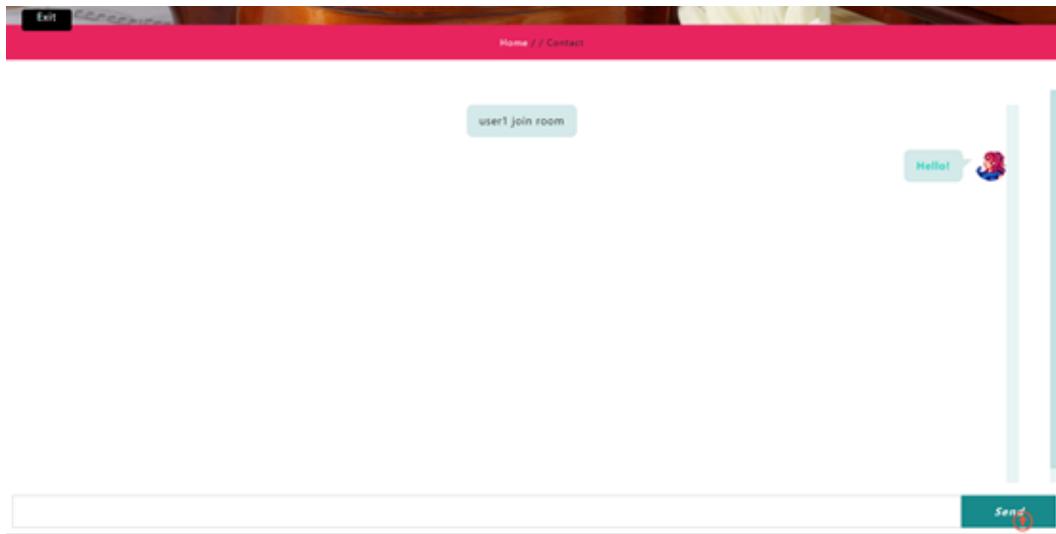
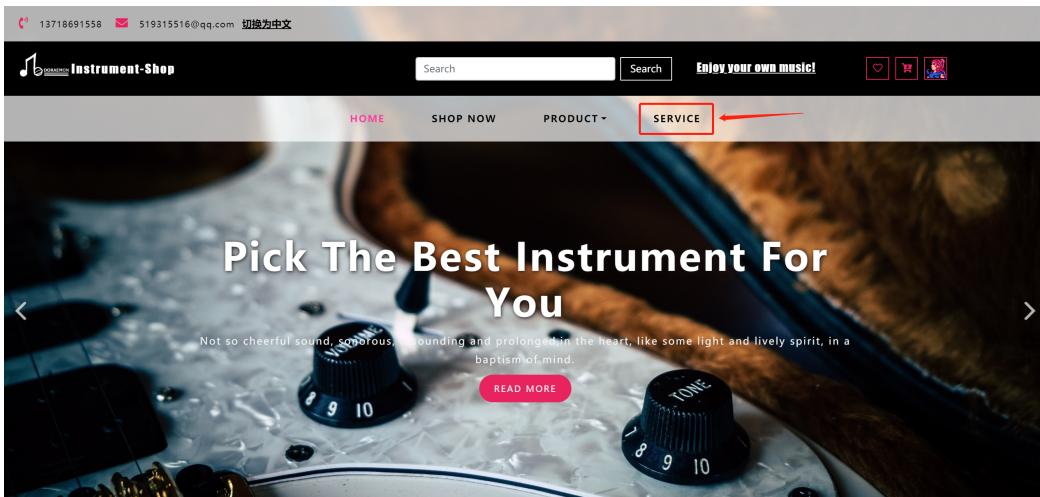
When the order status is not signed in, you can see the cancel order button. If you want to cancel the order, you can click this button to cancel all orders of the order number.

### Top-up:

As a customer, you can click the recharge button on the personal page to enter the recharge page, enter the amount you want to recharge and recharge,as shown the following figure.When the customer pays for goods, if the balance is insufficient, he can also click the recharge button on the page to enter the recharge page for recharge,as shown the following figure.**After you need to wait for the administrator to confirm your recharge, the recharge amount will arrive in the account**



Customer online chat:



On the main page, there is a button for the contact between customer and staff. after click the service button, customer will get into a chat box, but customer can only have conversation with staff.

Edit personal information:

The screenshot shows the admin's interface for managing a single order. On the left, there's a sidebar with a user icon and the name "admin". Below it are links for Home, Customer, Products, Orders, Services, and Top up List. Buttons for "Come back to Home Page", "切换为中文" (Switch to Chinese), and "Log out at here" are also present.

In the center, there's a large image of a black digital piano. Above the piano are two buttons: "Urgent" (red) and "Delivered" (blue). To the right of the piano, there's a section titled "Order5Details:" with the following information:

- Product:** Piano-YDP144B
- Price:** \$18270
- Num:** 5
- Delivery address:** None
- Transport:** delivery home
- Consignee name:** None
- Phone number:** None

Below this information are three buttons labeled 1., 2., and 3., each with a red arrow pointing to it:

1. A button labeled "Change the urgent status".
2. A dropdown menu labeled "Delivered" with a "Change order status" button next to it.
3. A button labeled "Cancel order".

The screenshot shows the customer's interface for managing their personal information. On the left, there's a sidebar with a user icon and the name "user1". Below it are links for Home, Collection, Shopping Cart, Orders, Services, and Top up List. Buttons for "Come back to Home Page", "切换为中文" (Switch to Chinese), and "Log out at here" are also present.

In the center, there's a form for editing personal information. It includes fields for changing the avatar (with a placeholder "选择文件 未选择任何文件"), Username (set to "user1"), Email (set to "12345678@qq.com"), Address (set to "None"), Phone number (set to "None"), Name (set to "None"), and buttons for "Submit" and "Back".

The background of the interface features a close-up image of a green harmonica surrounded by white flowers.

Customers can edit their personal information at the home page, by click the “Modify the information”.

## 1.2 Staff

### Staff manage single order interface:

Staff can click the orders button in their personal interface to enter the order management list interface. Click the specific order to enter the single order management interface. The staff can see the order commodity name, shipping address, signer's name, transport way and other information in the single order management interface.

**admin**

- [Home](#)
- [Customer](#)
- [Products](#)
- [Orders](#) (highlighted with a red circle)
- [Services](#)
- [Top up List](#)

[Come back to Home Page](#)

[切换为中文](#)

[Log out at here](#)

Normal	Not deliver	Normal	Not deliver
Order1: Piano-YDP144B Purchase time:2022-05-06 15:56:36.303425 Product information:Piano YDP144B Aries Series Digital Console Piano with Bench, Black	Order2: Piano-YDP144B Purchase time:2022-05-09 01:12:01.071613 Product information:Piano YDP144B Aries Series Digital Console Piano with Bench, Black		

**admin**

- [Home](#)
- [Collection](#)
- [Shopping Cart](#)
- [Services](#)
- [Orders](#)

[Come back to Home Page](#)

You can click here ~

[Log out at here](#)

**Order1 Details:**

**Product:** piano  
**Price:** \$3000  
**Num:** 1  
**Delivery address:** No. 100, Pingyuan Park, Chaoyang District, Beijing  
**Transport:** delivery home  
**Consignee name:** Tom  
**Phone number:** 13987652647

[Change the urgent status](#) → 1.

2. [Delivered](#) [Change order status](#) [Cancel order](#) → 3.

### Change order urgent status 1.:

When the transportation mode of the order is delivery home and the order status is not signed in, the button change the urgent status will appear. Staff can click this button to change the urgent status of all orders of the order number. If the order is in normal status, click this button to change the order to the urgent status. At the same time, all orders of the order number will also be listed in the front in the order management list interface. If the order is in urgent status, click this button, the order will change to normal status, and all orders of the order number will be cancelled from being placed at the top of the order management list interface. When the customer changes the order status to signed in, the expedited status of the order will automatically return to normal.

### Change order status 2.:

When the order status is not signed in, the change order status button will appear. The staff can select the order status in the selection column above and click this button to change the status of all orders of the order number.

Tips: The options in the selection column vary according to the order transportation mode.

### Cancellation of order 3.:

When the order status is not signed in, the employee can see that the cancel order button will appear. The staff can click this button to cancel all orders of the order number.

## Staff online chat:

Communication Lists				
User ID	Username	Time	Count of messages	Situation
	user1	2022-05-05 20:47:26.478953	1	unread
	user2	None	0	no message
	1920	None	0	no message
	rich	None	0	no message

On the main page, there is a button for the contact between customer and staff. After click the customer button, then staff will get into a chat list with all website customers, staff can contact with every customer.

## Release product:

  
**admin**  
[Home](#)  
[Customer](#)  
[Products](#)  
[Orders](#)  
[Services](#)  
[Top up List](#)  
[Come back to Home Page](#)  
  
[切换为中文](#)  
[Log out at here](#)

**Products**

Product List

[add product](#)

name	quantity	release_time	price	type	Operation
Piano-YDP144B	19	2022-05-08 15:56:36.286454	\$3654	piano	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Drum-LC178X025	12	2022-05-08 15:56:36.289240	\$620	drum	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Piano-KDP75	12	2022-05-08 14:40:22.031379	\$349	piano	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Drum-KD-5080-01	20	2022-05-08 16:23:15.224396	\$780	drum	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Electric-Drum-Set	5	2022-05-08 16:36:43.799057	\$3230	drum	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Horn-14D	15	2022-05-08 16:52:38.802058	\$6478	horn	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Horn-92D	21	2022-05-08 16:54:51.407667	\$2599	horn	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Trombone-1240RL-T	8	2022-05-08 16:59:29.382174	\$6144	trombone	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Trombone-620G	8	2022-05-08 17:00:39.823866	\$4362	trombone	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Trumpet-YTR-8335	19	2022-05-08 17:05:41.020514	\$1877	trumpet	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Trumpet-180S37	28	2022-05-08 17:07:41.534389	\$1149	trumpet	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>

  
**admin**  
[Home](#)  
[Customer](#)  
[Products](#)  
[Orders](#)  
[Services](#)  
[Top up List](#)  
[Come back to Home Page](#)  
  
[切换为中文](#)  
[Log out at here](#)

**New Product**

**Basic Information**

product name:	<input type="text" value="Please enter..."/>
stock:	<input type="text" value="Please enter..."/>
price:	<input type="text" value="Please enter..."/>
type:	<input type="text" value="piano"/>

**Commodity Picture**

Picture	<input type="file" value="选择文件"/>
	<input type="file" value="选择文件"/>
	<input type="file" value="选择文件"/>

**Product Description**

introduction	<input type="text"/>
--------------	----------------------

Staff can find a button named “add product” in the Product page, which is in home page. Staff can add a new commodity after fill product name, stock, price, type of this instrument, introduction and add pictures and music of the instrument.

### Edit product:

name	quantity	release_time	price	type	Operation
Piano-YDP144B	19	2022-05-06 15:56:36.286454	\$3654	piano	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Drum-LC178X025	12	2022-05-06 15:56:36.289240	\$620	drum	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Piano-KDP75	12	2022-05-08 14:20:22.031379	\$349	piano	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Drum-KD-5080-01	20	2022-05-08 16:23:15.224396	\$780	drum	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Electric-Drum-Set	5	2022-05-08 16:36:43.799057	\$3230	drum	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Horn-14D	15	2022-05-08 16:52:38.802058	\$6478	horn	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Horn-92D	21	2022-05-08 16:54:51.407667	\$2599	horn	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Trombone-1240RL-T	8	2022-05-08 16:59:29.382174	\$6144	trombone	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Trombone-620G	8	2022-05-08 17:00:39.823868	\$4362	trombone	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Trumpet-VTR-8335	19	2022-05-08 17:05:41.020514	\$1877	trumpet	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>
Trumpet-180S37	28	2022-05-08 17:07:41.534389	\$1149	trumpet	<a href="#">View</a> <a href="#">Delete</a> <a href="#">Edit</a>

Staff can find buttons named “Edit” and “Delete”, at the same page of Release product. After click this button staff can edit all the information of this commodity just as what they do when release product, or **click delete to delete a commodity**. (modify reason: Adopt feedback from other group)

### Staff recharge management:

Customer	Top up Time	Top up Amount	Top up Situation	Confirmation
user1	2022-05-09 01:11:28.039246	\$1000000000	Arrived to account	<a href="#">Confirmed</a>
rich	2022-05-30 18:22:39.612376	\$1000	Need to be dealt with	<a href="#">Confirm</a>

Click the Top-up list as shown in mark 1 in the following figure on the staff interface to view the Top-up record of the customer. On the Top-up list page, you can view the related information of the customer's top-up record. If you click "confirm" as shown in mark 3 in the following figure, the user's top-up is approved.

### Ban a user/unban a customer:

User ID	Username	Email	Register time	Balance	Operation
2	user1	12345678@qq.com	2022-05-06 15:56:36.290273	0	<a href="#">Ban</a>
3	user2	87654321@qq.com	2022-05-06 15:56:36.290273	0	<a href="#">Ban</a>
4	rich20	135792468@qq.com	2022-05-06 15:56:36.291268	0	<a href="#">Ban</a>
5	rich	richrich@gmail.com	2022-05-06 15:56:36.291268	10000000	<a href="#">Ban</a>

The staff can ban a customer's account so that the user cannot log in. If the banned customer is already logged in, he or she will not be able to use functions such as buying goods and collect, can only simply browse the web. Staff can also unban customers so that they can use the account normally.

## 2 System documentation

### 2.1 Abstract

This project is about the establishment of an online shopping mall for selling Musical Instruments. The website is divided into two ports: customers and employees. Customers can browse, buy Musical Instruments and consult questions, while employees can modify information about Musical Instruments and answer customers' questions. This documentation will explain the details of this system, applied technologies and group work.

### 2.2 Introduction

#### 2.2.1 Background

Due to the impact of the pandemic, most people gradually change their offline shopping to online shopping, and many businesses that only have offline stores have set up online shopping malls. There is a local musical instrument shop that, before the pandemic, had been selling Musical Instruments to individual customers who came to buy them. Due to the epidemic, people are going out less, sales of Musical Instruments have also fallen. Therefore, the store has decided to open an online mall to sell Musical Instruments.

#### 2.2.2 Detail

Here we provide two accounts for testing

- **IP: <http://193.1.133.149:5002>**
- **customer account:**  
user1 password:A000000 email:test.doraemon@outlook.com email password: WASD13579
- **employee account:**  
admin password:A123456

This project is to develop a website to sell musical instruments for this local store. Thus, the main relationship of the it is to provide a plat form to sell instruments for customers. It is divided into two portals: customers and employees.

- Customer
  - Register and login the personal profile
  - Browse the products
  - Purchase instruments
  - Change the personal information
  - Check the order
  - Modify the information of order
  - Consult the problem of purchase
  - Add the instruments into shopping cart
  - Translation of languages
  - Search the specific instruments

- Collection of products
- Listen the sound of instruments
- Employees
  - Check the order from customers
  - Translating the languages
  - Reply the message from customers
  - Add the information of products
  - Delete the information of products
  - Modify the information of products

Besides, a clean and beautiful layout is also important for a website, so our team also uses JavaScript to render a good layout. To show the information and credit of the store, this system shows the introduction of the store, and customers can read the comments of each products which are sent by other customers. Particularly, this website has a function of trying to listen the musical instruments. Customers can listen the sound of each instruments. It can help customers to choose the appropriate instruments.

### 2.2.3 Milestone

Milestone	Start Date	End Date
Completed work package	2022/03/01	2022/03/5
Design system	2022/03/02	2022/03/07
Completed database	2022/03/07	2022/03/14
Front-end development completed	2022/03/08	2022/05/12
First big test	2022/04/15	2022/04/17
Back-end coding success	2022/03/10	2022/05/17
Second test	2022/05/17	2022/05/19
Final version completed	2022/04/10	2022/05/20

### 2.2.4 Management

In order to complete the system on time with high quality, our team made the work package according to customer requirements and actual ability. In the process of the project, we will combine testing and programming, carry out project testing regularly, and make modifications when bugs are found, so as to avoid risks caused by the accumulation of bugs. Since customer requirements may change at any time during the development process, we will also modify our plan accordingly, so as to ensure timely completion of the project while meeting customer requirements.

## 2.3 Group Work

### 2.3.1 Group Structure

Our group is composed of Ningxin LI, Xuan WU, Yilan WANG, Peixuan DONG, Tingjun WU, Xuran ZHAO. The group has been divided into 3 subgroups: front-end group, back-end group, and test&maintenance group.

Subgroup	Member
front-end group	Xuan WU, Yilan WANG, Xuran ZHAO
back-end group	Ningxin LI, Peixuan DONG, Tingjun WU
test&maintenance group	Xuan WU, Yilan WANG, Xuran ZHAO

### 2.3.2 Changes

In the process of project development, changes often occur, including changes in the division of labor among members, changes in plans, and changes in requirements. Either way, we need a reasonable way to manage change. If changes are not handled properly, it is likely that there will be disagreements within the group, product quality degradation, time consuming, etc. Therefore, how to manage change is critical. Our teams manage changes using the following processes: requesting changes, analyzing impact, group voting, and implementing changes.

- **Requesting Changes:** Team members determine the changes that need to be made, submit a change request, and need to explain the areas where the change will affect, such as time, the interface of the corresponding change, etc., as well as the complexity and scale of the change.
- **Analyzing Impact:** Careful analysis and assessment of the possible impact of the change, especially in terms of manpower, cost, and time, provided the basis for a subsequent panel vote.
- **Group voting:** At the weekly group meeting, the group members unanimously decide whether or not to approve the changes. If it is a more urgent change, it can also be decided through WeChat or a conference call.
- **Implementing changes:** After the changes are confirmed, the team members will determine the timetable for the changes, and publish the content that needs to be changed on gitlab so that all members can view the progress.

### 2.3.3 Quality

During the development process, team members need to control the quality of the product, and finally deliver a product that meets the quality requirements. During the development process, we often encountered problems such as unsatisfactory front-end design, so that team members often had to repeat the work many times. Or the understanding of a function is not thorough enough, so that the development cannot meet the needs of the function. In order to solve these problems, we use some third-party platforms to help us better ensure the quality of our products.

- **Trello:** A platform for project management, each member can browse and add tasks for the week, and modify the progress of tasks.
- **GitLab:** Gitlab, as a git code repository, provides a simple, transparent and efficient way of working with git, combined with an issue tracking system. Gitlab has flexible configuration scripts, lightweight task lists and Kanban boards, and supports task status changes and email reminders for git submissions
- **Jishi Design:** A Chinese UI design tool, it can help front-end members to design the interface to be developed, and show the static interface to the whole team to prevent modification after it has been developed.

Through these tools, each team member can participate in various tasks, more accurately grasp the progress of the entire project, and make up for the lack of personal completion.

### 2.3.4 Communication

Good communication is essential when a team develops a project together. Communication can reduce time costs and support efficient and high-quality output. To effectively achieve our goals, we decided to hold weekly Scrum meetings on Tuesdays and Thursdays. Tuesday's meeting is about identifying everyone's tasks for the week, and Thursday's meeting is about summarizing work from previous days, discuss technical issues, and schedule tasks (usually planning content suitable for weekend development at home). On Wednesday, the team will also have a brief meeting with the TA, and absences should be requested in advance.

Team members need to share learning resources, help each other, jointly design plans, update progress, point out problems in a timely manner, and discuss to find solutions to problems. In addition to the fixed offline meetings twice a week, the team can also communicate through WeChat, email, telephone and face-to-face dialogue. Members need to check WeChat in time to obtain information in a timely manner, and log in to the mailbox and project management regularly to obtain any information from other teams. Information and notifications for members, teaching assistants, and mentors.

- **Meeting:** At a minimum, the meeting should cover sharing progress, summarizing work, assigning tasks, discussing problems and trying to find solutions. Meetings should be efficient and orderly, with team members taking turns speaking or asking questions. Each scrum lasts 15-20 minutes, and the purpose is to adjust the team's progress so that everyone knows the progress of the project, and it is also an opportunity to communicate with each other.
- **participate:** Each team member must participate in the development of the project and complete their assigned work. Everyone has a corresponding role and needs to be responsible for their own work.
- **decision making:** Team members cannot make decisions on their own, and every change that affects project delivery needs to be discussed with members before they can be decided. Team members need to be involved in decision-making together, and when solving a problem, the team should fully analyze, collect all opinions, and then make a final decision.
- **conflict:** Conflicts in group development are quite normal. Faced with conflicts, each member should try to explain the conflicts patiently and peacefully, and consider and respect others from the perspective of others.

## 2.4 Functional Requirements Implementation

The technical and design ideas encountered by the key features of the project will be introduced. For each feature, a flow chart will be included and an explanation will be given as to why I used this approach to design the feature. (Modify Reason:After feedback from other groups)

### 2.4.1 Registration/Login

**User Story:** "As a customer, I want to browse the products to choose the favorite product."

**Design Idea:**

The register accepts a user's E-mail, username, and password and stores them in a database, which are the basic information of the user. The user's email address must be correct. The password must contain at least six letters and digits. The password must be entered twice. A user can register only when the two passwords are the same. You need to enter your user name and password. If you enter the correct user name and password, the main page of the website is displayed. If you make a mistake, the user will be prompted to try again. If you forget your password, the user will need to enter their username, email address. The system then sends an email containing a verification code to the mailbox. After entering the verification code, you can change the password successfully. The website is divided into the user side and the employee side. The employee does not need to register when logging in, but only needs to enter a pre-designed employee account to enter the employee side.

**Flow chart:**

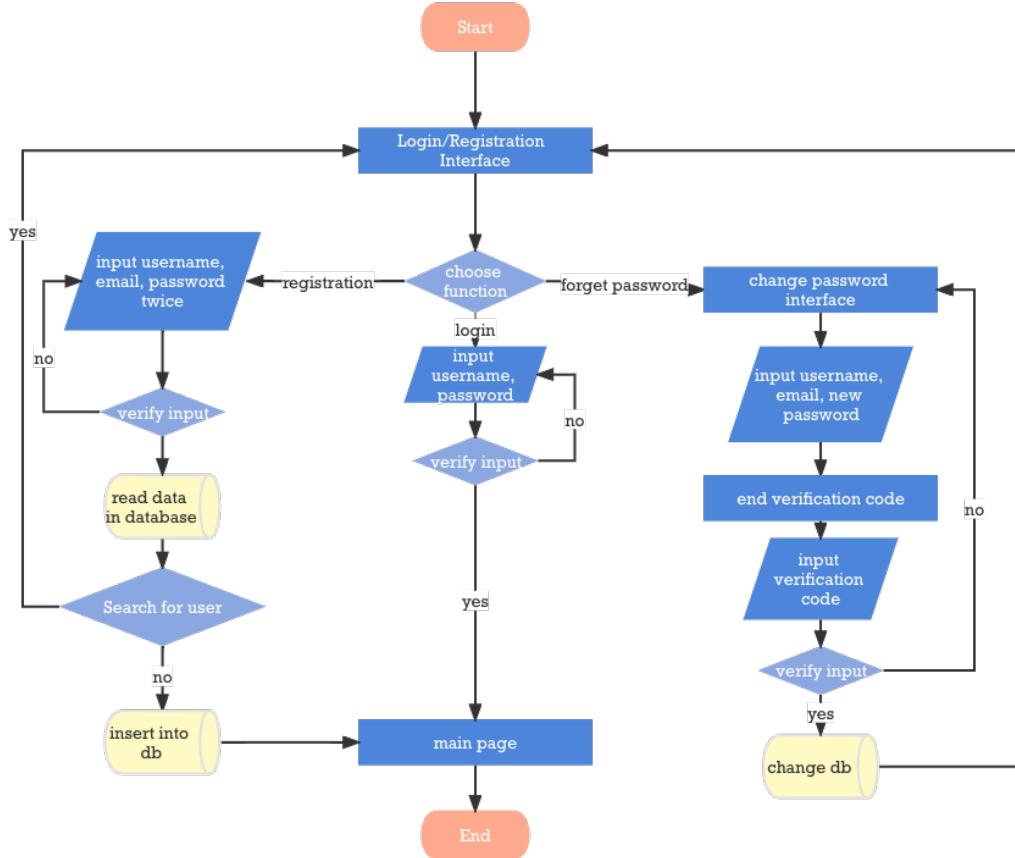


Figure 1: Flow chart: Login/Registration

#### 2.4.2 Browse product

**User Story:** "As a customer, I want to browse products to choose the favorite product"

**Design Idea:**

Customers can browse products as logged in or not logged in, and all products will be displayed on the Shop Now page. Users can see all commodities in this interface, and they can collect commodities in the login state. If they do not log in, the page will automatically jump to the login interface for users to log in. If the user wants to browse the detailed information of the product, he can click the corresponding product, and the page will jump.

**Flow chart:**

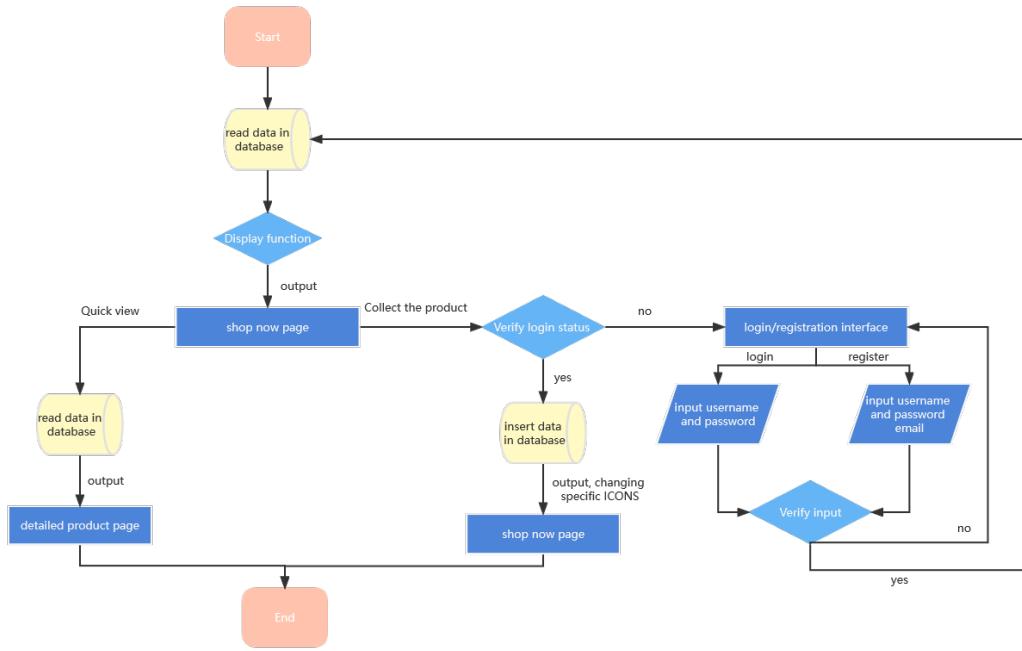


Figure 2: Flow chart: Browse products

#### 2.4.3 Search product

**User Story:** "As a customer, I need a search function so that I can find what I want."

**Design Idea:**

Design Idea: The search function is an important feature of user friendliness. Doraemon music Store supports product search. This feature uses JavaScript and Ajax to display search results, and users can click on the product title in the search results to browse for detailed product information. **Flow chart:**

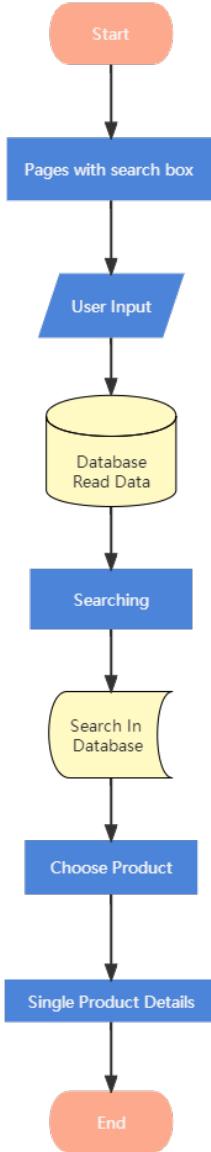


Figure 3: Flow chart: Search product

#### 2.4.4 View order

**User Story:** "As a customer, I want to view the order which I have ordered and edit the orders."

**Design Idea:**

Customers can browse the order of the goods they have ordered and view the details of the order. Detailed information includes commodity information, logistics information, receiving information, etc., to help customers better understand the status of the goods they buy. In addition, users can modify or cancel orders on the order details screen. To modify the order, you can modify the shipping address, consignee, mobile phone number and delivery method. Customers can only modify orders that have not been delivered. Customers can also cancel orders that do not arrive to make their shopping experience better.

**Flow chart:**

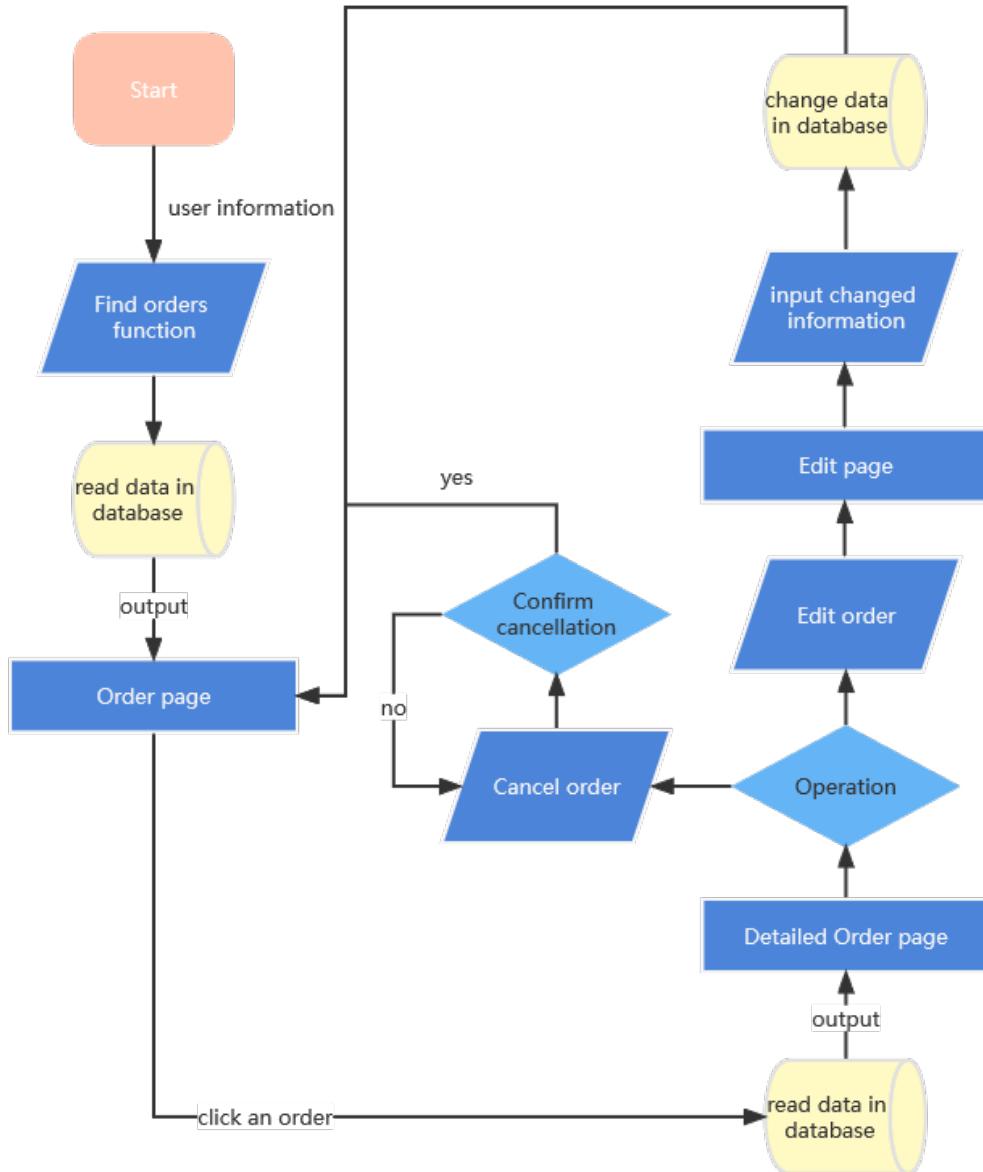


Figure 4: Flow chart: View order

#### 2.4.5 Purchase

**User Story:** "As a customer, I want to purchase products"

**Design Idea:**

Purchase and order generation is one of the core functions of our online mall project. The design of online shopping mall purchase process has two, the first is in the product page, directly purchase selected goods. The second way is to first add the selected product to the cart, and then place an order in the cart. The purchase process involves multiple pages and requires multiple data exchanges and modifications, such as verifying that an item is in stock and a user has an adequate balance. This is the most complex feature of the project.

**Flow chart:**

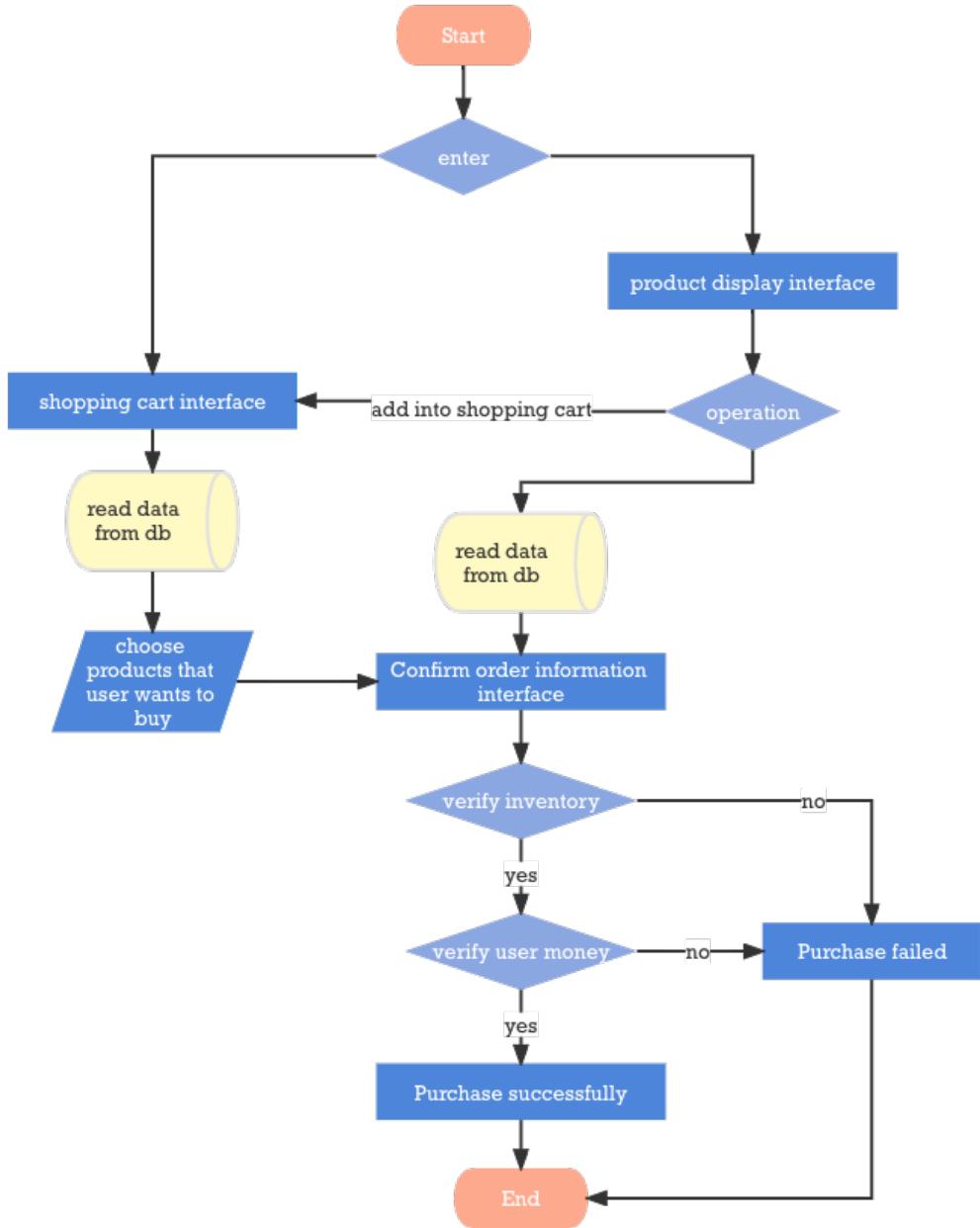


Figure 5: Flow chart: Purchase products

#### 2.4.6 Comment

**User Story:** "As a customer, I want to comment the products to share my opinions."

**Design Idea:**

Users can comment on the product details page and other users can see it. Comments are updated in real time. Each comment shows the title, content, and author. Flask and HTML are used to implement this feature.

**Flow chart:**

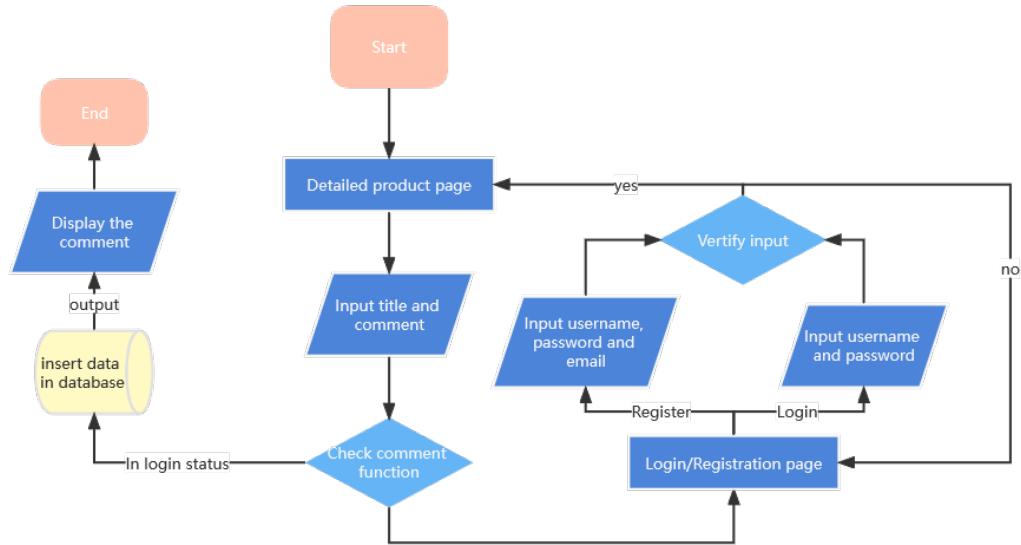


Figure 6: Flow chart: Comment

#### 2.4.7 Product Management

**User Story:** "As a staff, I want to add new products or editing existing products"

**Design Idea:**

Product management includes the modification of existing product information and the addition of new products. On the employee side, the system reads the information about the musical instrument in the database and enters the product information into the form. The employee can read the basic information of all products on the product management interface and edit the existing products. In addition, employees can add new products on this page. Whether editing an existing product or a new product, it involves reading, modifying and inserting from the database. Moreover, since the information of the musical instrument also includes music and pictures, this function not only involves the operation of the database, but also the storage and reading of the files.

**Flow chart:**

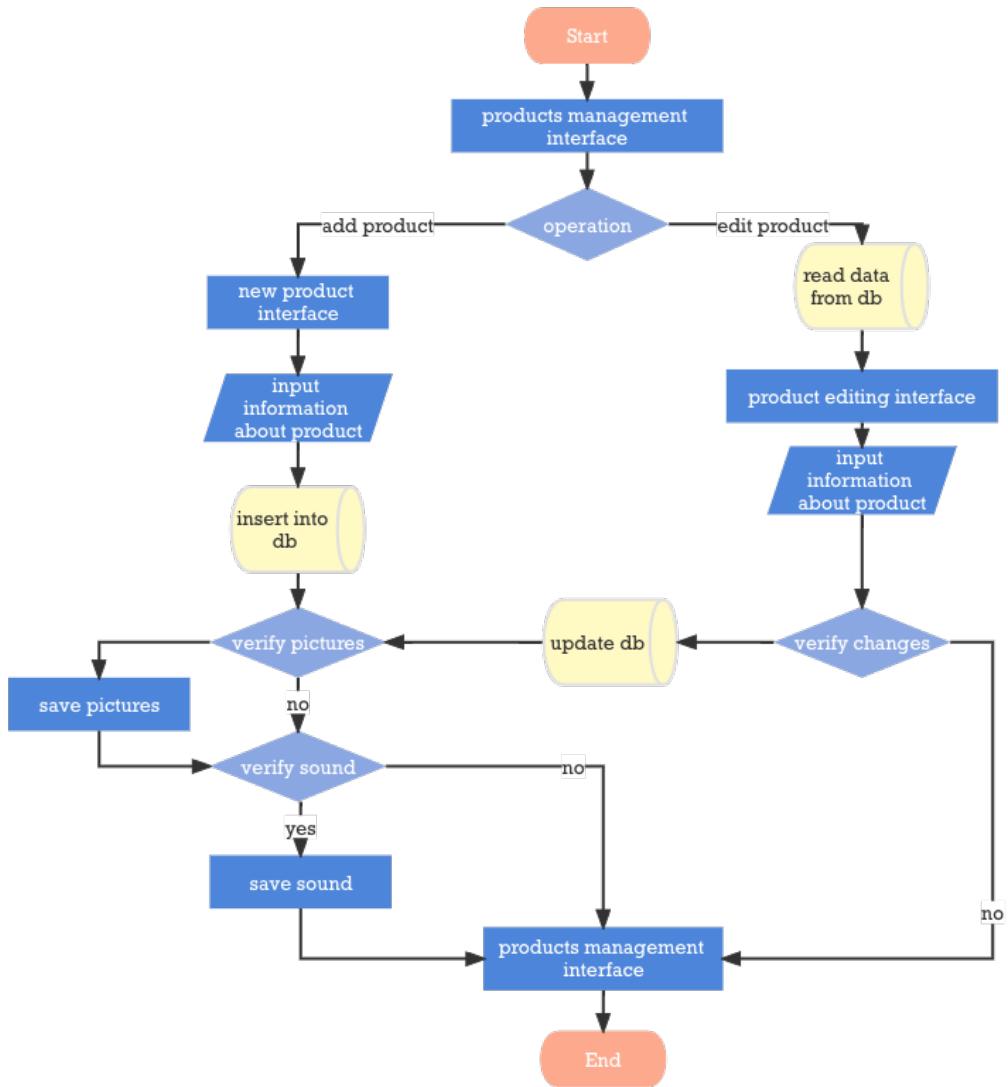


Figure 7: Flow chart:Product Management

#### 2.4.8 Order management

**User Story:** "As a staff, I want to view all orders and change their status and priority"

**Design Idea:**

Order management is another important function on the employee side. The data about the order stored in the background of the system includes order id, user id, product id, address-related information and order status. Staff can change the status of the order (shipped or not shipped), the priority of the order (higher priority will be shipped faster), and cancel the order.

**Flow chart:**

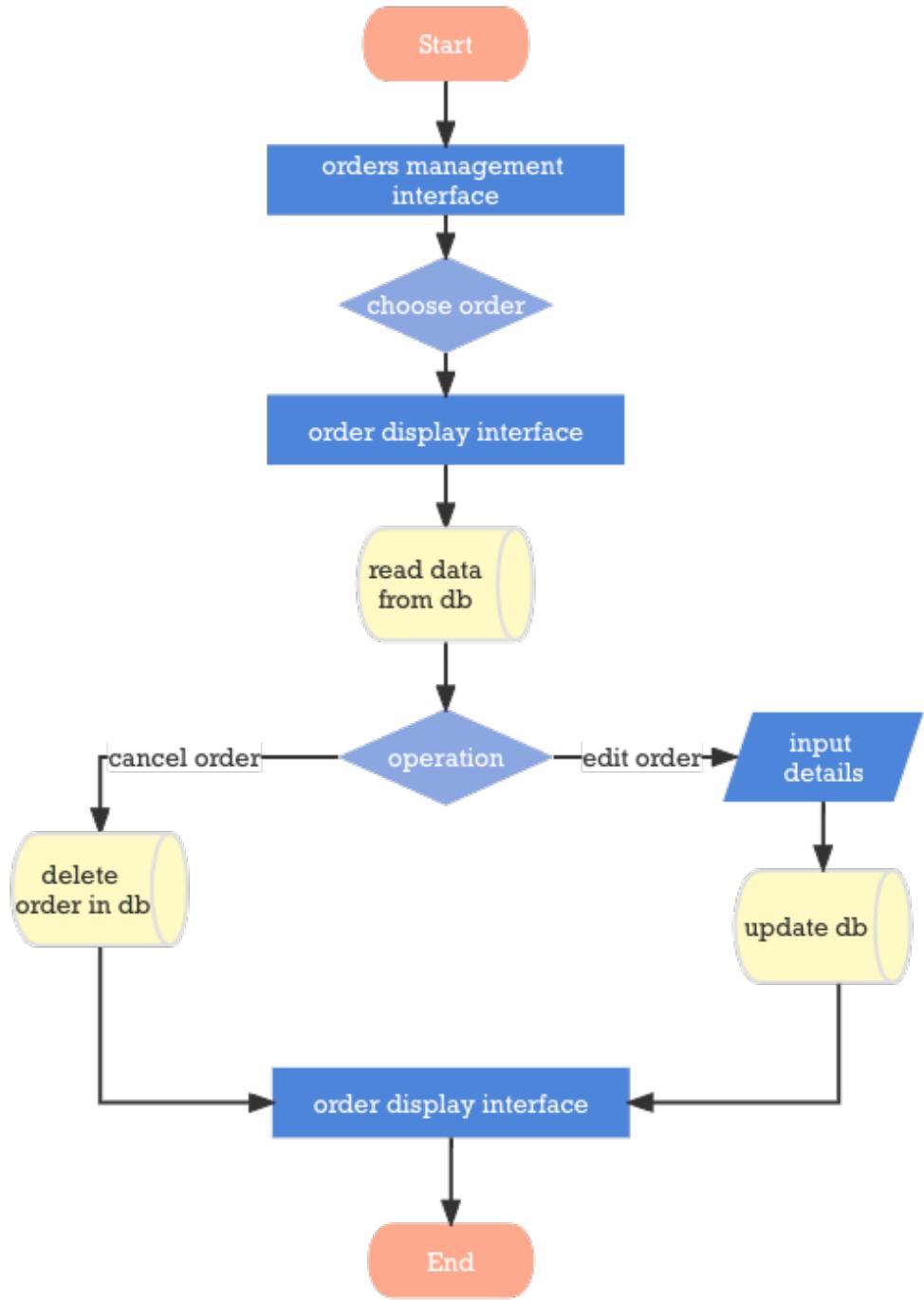


Figure 8: Flow chart:Order Management

#### 2.4.9 Online communication

**User Story:** "As a customer, I think there is an online way to communicate with employees online and ask questions about products; As an employee, I want to communicate with customers online so that I can reply to their questions."

#### Design Idea:

Our project provides the contact page to realize the online chat function between customers and employees. After entering the service page, users will enter a chat room where they can send messages or check historical messages. On the staff side, after entering the service page, the staff will first see the list of customers, and then enter the chat room after selecting a specific customer. Flask-socketio and database was used for online communication.

#### Flow chart:

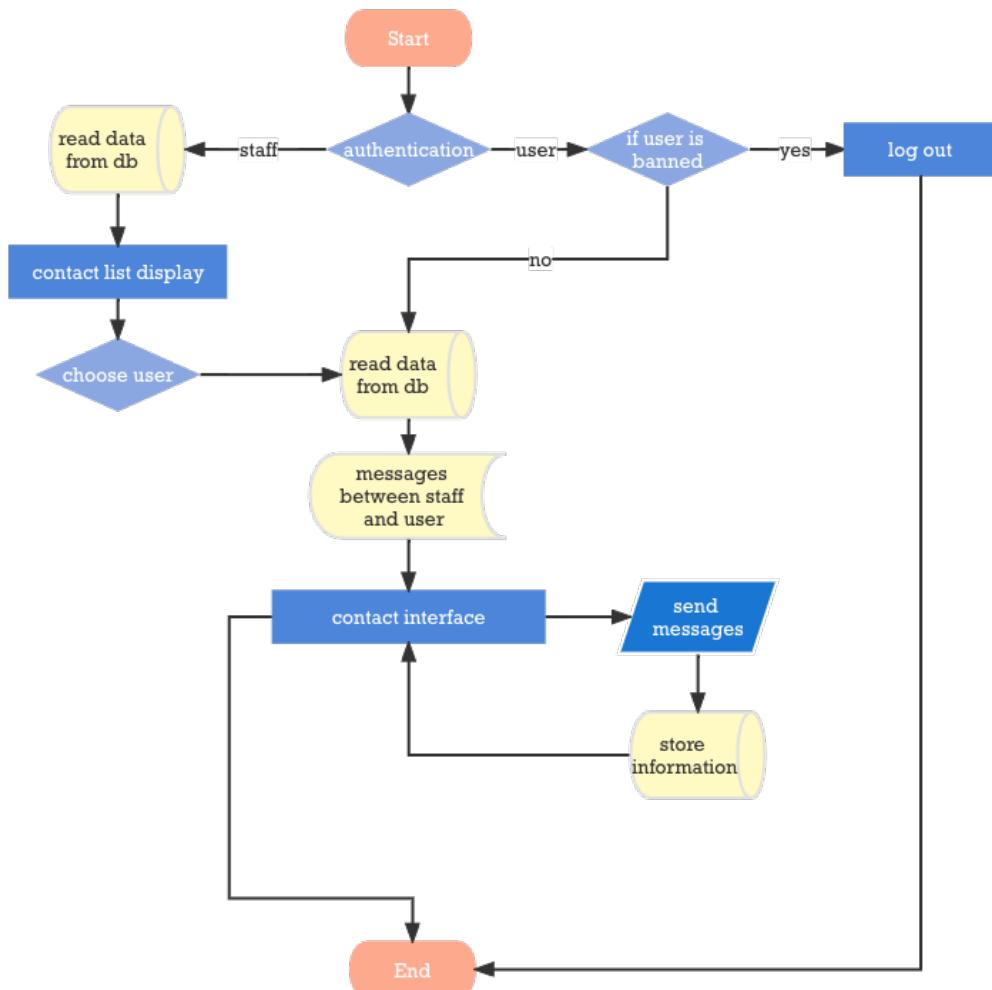


Figure 9: Flow chart:Communication

#### 2.4.10 Language switching

**User Story:** "As a customer, I need to switch between Chinese and English so that I can understand the content of the website."

**Design Idea:**

We added a language switch button to the top of the navigation bar on the home page, the top of the shopping cart, and the menu bar on the left side of the home page. The site is available in both Chinese and English. When a user logs in, the site automatically monitors the user's browser language preferences and follows them. In order to facilitate the users of both languages to better find the language switch button, we use Chinese in the language switch button when the page is displayed in English, and English title is used in the button when the page is displayed in Chinese. We use the I18n plugin in jquery and session judgment in Python to implement this functionality. **Flow chart:**

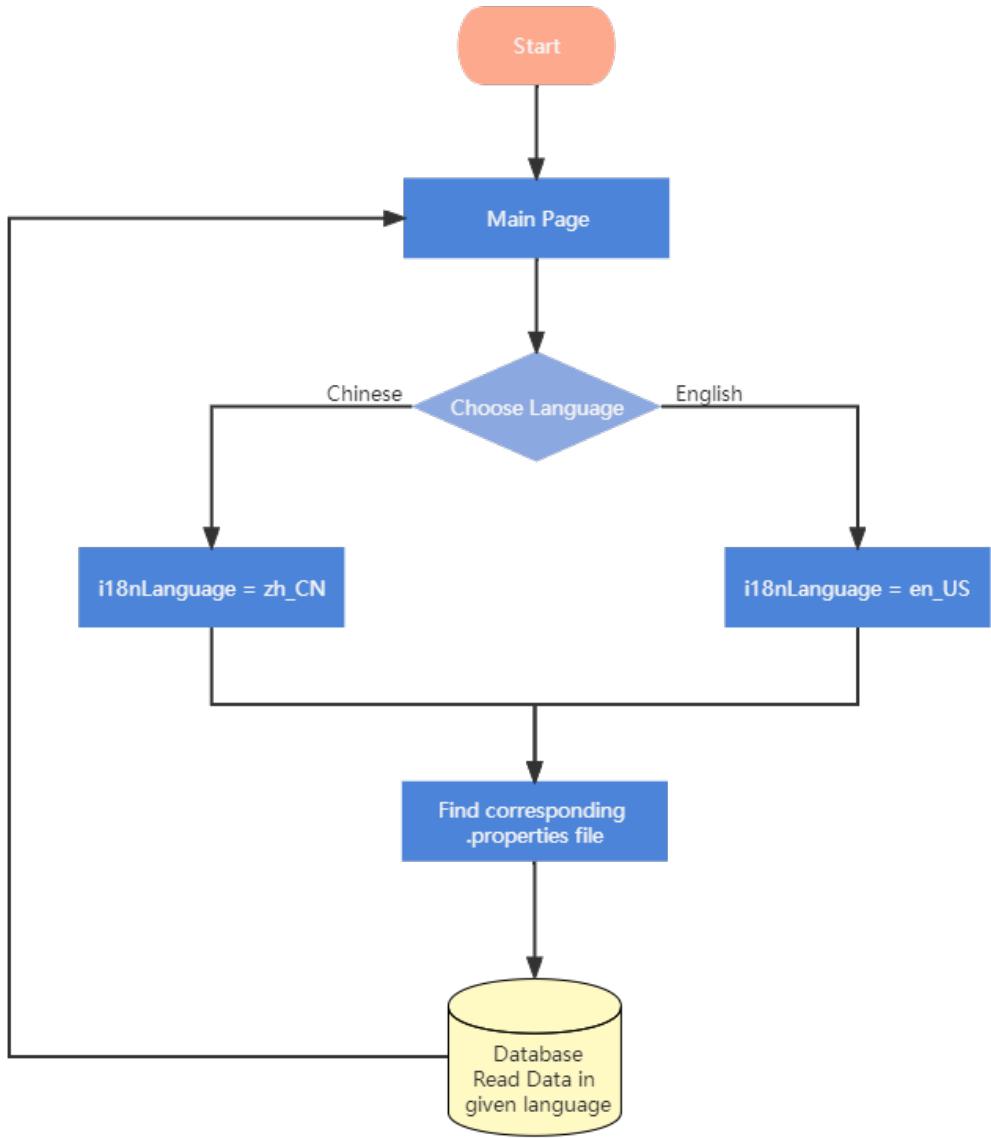


Figure 10: Flow chart:Language switching

## 2.5 Non-Functional Requirements Implementation

### 2.5.1 Security

Our project ensures user information security in two main ways. The first is that the user registration will have certain requirements on the password, to avoid the user to use a simple password and be cracked, and in the user's password stored in the database, will use hash encryption,which ensures that a user's information will not be compromised in the event of a malicious attack on the database. In addition, when a user requests to change the password, the system sends an email to the user's mailbox using SMTP for email transmission. The email includes a randomly generated verification code that changes the password only when the user enters the corresponding verification code, which improves the security of the account.

### 2.5.2 Reliability

The loading of our project page is stable, and the page jump is reasonable. For example, it is set that it will not return to the top after submitting a comment. After the user selects the page language, the language will not be changed even if the page is jumped. Users can use the website with confidence. In addition, The design style based on component development has the characteristics of high cohesion and low coupling, which is convenient for adding and modifying functions later. In terms of UI, several mainstream desktop browsers work well, among them, the Edge browser is the most stable and suitable for devices of different screen sizes.

## 2.6 Technical Implementation

### 2.6.1 Software Architecture

According to the structure of the forum logical model, the online mall system follows the Component-Based Development (CBD) design style. In the CBD design style, components are independent of each other and contain each other. Therefore, during the design process, each system operation is completed by the cooperation of components, but it is necessary to ensure that there is information exchange between the components. This information exchange is usually achieved through interfaces. For example, to add and modify products, it is necessary to identify the user's authority first, determine whether the user can add and modify products, and then open the function to the user.

In addition to the above CBD design style, this project also takes the database as the center design style. The reason for adopting this design style is that the data used in the system are all stored in the database, such as username and password, corresponding content of products and reviews, user's orders and order status, etc. in the database. When users operate in the system, the system reads and invokes data in the database and displays it in the front end. Therefore, the database is very important in this project.

Last but not least, event-driven design is also applied to the system, where a step is triggered by the completion of the previous step. For example, when a user places an order, the system needs to confirm that the user has logged in before making a purchase. This is called an event-driven design style application.

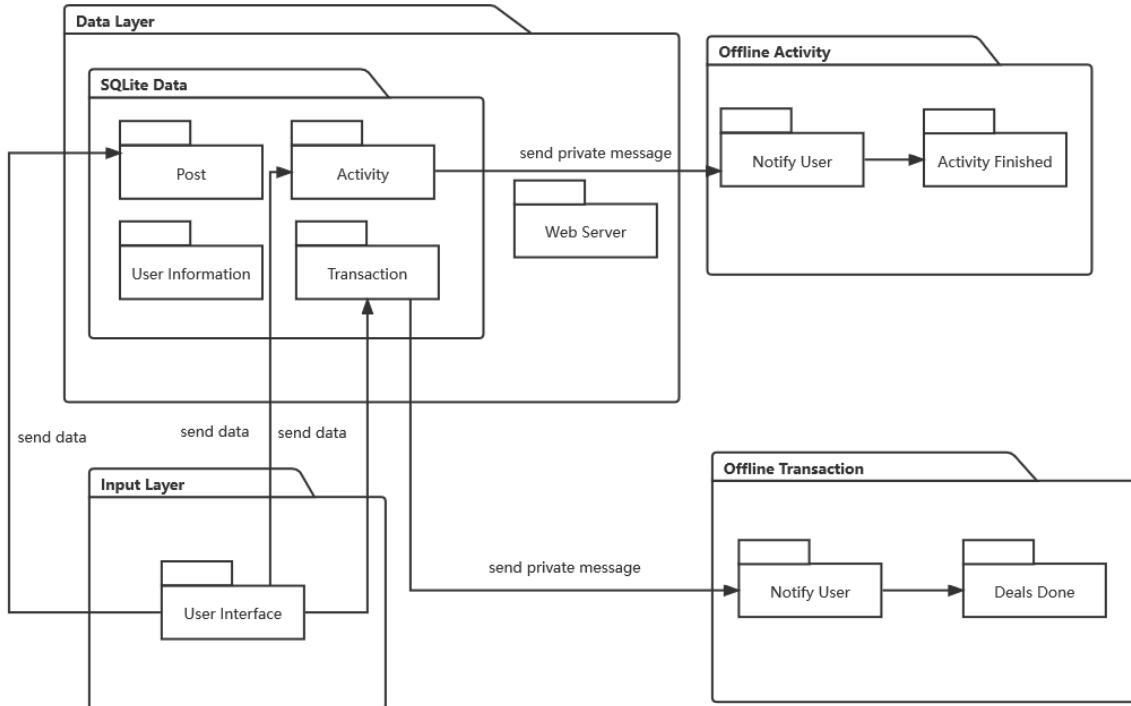


Figure 11: Architecture Diagram

## 2.6.2 Database

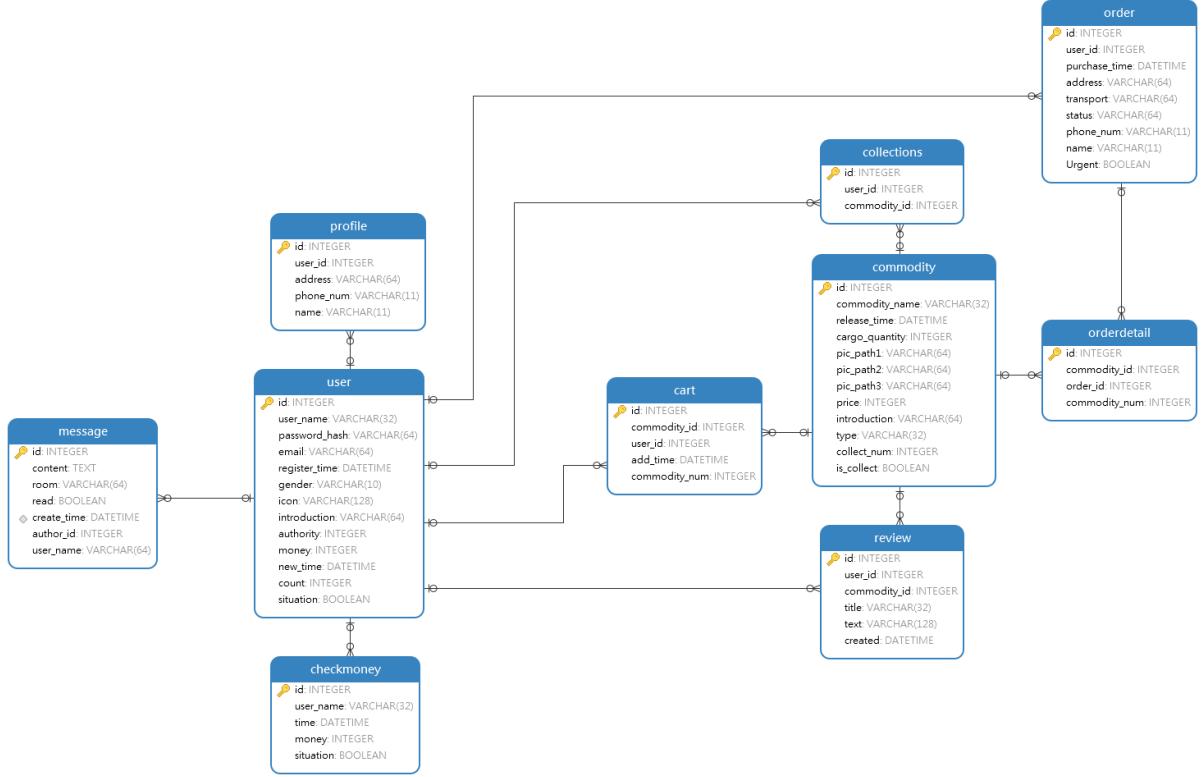


Figure 12: Database Diagram

- **users**: It stores the information of users, such as student\_id, email, username and so on.
- **commodity**: It stores the information of each type of commodities, such as name, price, quantity, introduction
- **profile**: It stores the user's personal information, such as address and contact number
- **cart**: It stores the information of the user's shopping cart, such as user ID, product ID
- **collection**: It stores information about the user's favorite items, such as user ID, item ID
- **message**: It stores information exchanged between customers and customer service, such as user ID, room number, whether it has been read, etc
- **order**: It stores the user's order information, such as user ID, order time, address, status and so on
- **orderdetail**: It stores the details of the order, such as the order number and item ID
- **review**: It stores information about product reviews, such as user ID, product ID, and comment content
- **checkmoney**: It stores information such as user ID and amount of money when a user initiates a charge

## 2.6.3 Global flow control

### • Time dependence

Time dependence exists in the system. It has real-time dependence and can store various operation information of users according to actual time. For example, when a user wants to publish an article, the system will record the time when the user publishes the article and display it on the main page according to the time of publication. User comments are also displayed on the main page of articles in the order they were published. For each article that receives comments, the system will also sort them according to the time of comments.

### • Orderliness

There's linear execution in the system. The user buys goods, generates order information, and the

administrator changes the order status until the final order is completed (that is, the buyer signs for the goods). Completion of the order means completion of the transaction. This will happen after each order becomes signed. The order can only be generated after the buyer has made a payment.

- **Concurrency**

There will be major concurrency in this system. Each item, review, shopping cart, user, and order has its own data store. Each has a corresponding ID. Goods have corresponding content, comments, collection number, etc.; Users have ids, email addresses, passwords, favorites, and so on. After the user completes the operation, the corresponding data changes in parallel.

#### 2.6.4 Requirements

**Hardware and software requirements.**

**The user should possess a PC which enables the basic web browser.**

- alembic: 1.5.8
- antiorm: 1.2.1
- bidict: 0.22.0
- blinker: 1.4
- click: 8.0.4
- colorama: 0.4.4
- Flask: 2.0.3
- Flask-Avatars: 0.2.2
- Flask-Mail: 0.9.1
- Flask-Migrate: 2.7.0
- Flask-SocketIO: 5.1.1
- Flask-SQLAlchemy: 2.5.1
- Flask-WTF: 1.0.0
- greenlet: 1.1.2
- itsdangerous: 2.1.0
- Jinja2: 3.0.3
- Mako: 1.2.0
- MarkupSafe: 2.1.0
- numpy: 1.22.3
- Pillow: 9.0.1
- PyMySQL: 1.0.2
- python-dateutil: 2.8.2
- python-editor: 1.0.4
- python-engineio: 4.3.1
- python-socketio: 5.5.2
- six: 1.16.0
- Werkzeug: 2.0.3

#### 2.6.5 Data Structures

The data structure we used in this project hash table. When user logged in the password used the Werkzeug to encode the password user input and store them as hashcode in the database in order to enhance the security of the website. This data structure is mainly used in user registration process and user login process when user input their password, the website will check whether the hash code matches the one stored in database to distinguish the user's identification.

```
1 @app.route('/login/login_mes')
2 def login_mes():
3     user_find = User.query.filter(User.user_name == request.form["username2"]
4         ]).first()
5     if not user_find:
6         flash('Incorrect username')
7         return redirect(url_for('login'))
8     if (check_password_hash(user_find.password_hash, request.form["password"]
9         )):
10         flash('Login success!')
11         session["USERNAME"] = user_find.user_name
```

```

10         session['Logged_in'] = True
11         session['uid'] = user_find.id
12         session['authority'] = user_find.authority
13         return redirect(url_for('main_page'))
14     else:
15         flash('Incorrect Password')
16         return redirect(url_for('login'))
17
18 @app.route('/login/reg_mes')
19 def reg_mes():
20     user_in_db = User.query.filter(User.user_name == request.form["username1"]
21         "").first()
22     if user_in_db:
23         flash('User has sign up!')
24         return redirect(url_for('login'))
25     else:
26         if request.form["password1"] != request.form["password2"]:
27             flash('Passwords do not match!')
28             return redirect(url_for('login'))
29         else:
30             passw_hash = generate_password_hash(request.form["password1"])
31             user = User(user_name=request.form["username1"], email=request.
32                 form["email"], password_hash=passw_hash)
33             db.session.add(user)
34             db.session.commit()
35             profile = Profile(user_id=user.id)
36             db.session.add(profile)
37             db.session.commit()
38             flash('User registered with username:{}.'.format(request.form[""
39                 "username1"]))
40             session['USERNAME'] = user.user_name
41             session['uid'] = user.id
42             print(session)
43             return redirect(url_for('main_page'))

```

---

## 2.6.6 Technique

### Front-end:

- JavaScript

JavaScript is the programming language of web. Most of modern HTML pages use JavaScript to realize dynamic function in webpages in order to provide users with more smooth and beautiful browsing effect. Our team used JavaScript to realize the dynamic addition of elements, the sliding window of login interface, the change of element style when clicking, etc., to ensure that users can quickly master the function when using our website.

- HTML

As an essential front-end language, we use HTML to construct the overall UI, combined with JavaScript and CSS to show users a beautiful website.

- CSS

Our team uses CSS to add complex styles to HTML elements, making the entire page more beautiful.

### Back-end:

- FLASK

FLASK is a very popular Web framework. FLASK is more flexible and secure than other Web frameworks and is suitable for small teams like us to build a feature-rich medium to small site in a short amount of time, which is enough for this online instrument store. In FLASK, the code is clean, easy to expand, and the plugin library is powerful enough to customize personalized websites. As a microframework, FLASK is more conducive to further development and maintenance.

- Python

Python is now the most popular programming language, suitable for rapid development on most platforms. In addition, FLASK is based on Python. We use Python language to create functions to achieve data processing and front-end interaction.

- SQLite

We use SQLite to store data. It has faster processing speed and is compatible with flask.

The following table shows functions and technologies used to implement them.

Num	Technique	Measurable objectives
1	FLASK_login, HTML, CSS, JavaScript	Customers who are not logged in can register or login to an existing account.
2	HTML, CSS, JavaScript	Customers can browser basic company information.
3	JQuery, CSS, HTML	The website can switch between English and Chinese.
4	SQL, HTML, CSS, JavaScript	Users could modify personal information.
5	Python, SQL, HTML, CSS, JavaScript	The home page can recommend products to customers according new arrival and popularity.
6	Python, SQL, HTML, CSS, JavaScript	Customers can check commodity information and purchase or collect commodities, or add to cart.
7	Python, SQL, HTML, CSS, JavaScript	Customers can combine payment for multiple orders in shopping cart.
8	Python, SQL, HTML, CSS, JavaScript	Customers can communicate to staff in real time.
9	Python, SQL, HTML, CSS, JavaScript	Customers can add, delete or modify commodity information.
10	Hash encryption	The system can ensure safety of user information.
11	CSS	The system realizes responsive layout.

### 2.6.7 Test and Maintenance

- **Test:**

At the end of each phase of the development process, we use white-box testing, which is either logic-driven testing or code-based testing. Since the test members themselves are development members, they understand the internal logical structure of the sequence and can test all logical paths. Our tests fall into two categories:

- **Static analysis:** Tests that can be performed without executing a program, such as code reviews, code scans. Group development often uses peer programming, in which one member writes the program and another reviews it to see if there are bugs caused by the member's subjective limitations or syntax errors.
- **Dynamic analysis:** tests that need to be executed, such as unit tests, coverage tests. At the end of each phase of development, members of the test team conduct unit tests. For example, when the functionality for employees to upload new products was developed in one week, the testers would run the program after development to see how the functionality was being developed.

- **Maintenance:**

In order to improve the quality of the site and give users a better experience, maintenance is still an important part of the project development process, so once the project is completed, some members of the development team stay on to maintain the project, while others are fired to work on other projects. The maintenance work includes: maintenance and update of the database, correction of system operation errors, and improvement of problems reported by users in the process of use.

## 2.7 Conclusion

Through the combination of various technologies at the front and back end, we finally realized a user-friendly online shopping store. Through the use of our website, customers can buy their favorite instruments without leaving home, and they can also communicate with the store staff online when they encounter problems. It also provides stores with a new sales channel, helping to boost the economy during the pandemic.