Comp5130 Internet & Web System

Cheng Wei Liao

2018-11-30

Week 11

This will be the last report for this semester. For the next two weeks, there will be presentation and demonstration.

This week I was working on product detail page and message system.

Product detail page is simple, customers click the link from the product list page, and the system will direct the page to the detail page. This page shows the price and the name of the product on the top, product posted date and time and the category on the middle, and product description on the bottom. On the right-hand side of the middle section, there are seller name and three button options. “Message” button can directly send the message to the seller. “Make offer” button can have users send their ideal price for the product to the seller. Click on the “Buy” button can directly order this product. And that is all the functions and element in this page.

For message system, it is more complicated. The content can be separate into two parts. On the left, there is the users that you have talked before or now. On the right is the content for each conversation. I spent a lot of time working on this system and solved the problems that I have encountered. First, to filter the people that the user has talked and should not have duplicate link on the page, I need to execute some complicated SQL command to reach this goal. For example, the table below shows how I store each conversation to database.

sender	receiver	content
a	b	AAA
a	c	BBB
b	c	CCC
c	a	DDD

Assume the current user is A, then the system should only show user B and C as the users that A has talked. However, no matter my SQL command is

```
SELECT receiver FROM `chat` WHERE sender='a'
```

OR

```
SELECT sender FROM `chat` WHERE receiver='a'
```

The result will contain duplicate person C

sender	receiver	content
a	c	BBB
c	a	DDD

So, what I want is that only list B and C as the result.

After several tests, I combined two SQL commands with “union” [1] to get expected result.

```
(SELECT receiver FROM `chat` WHERE sender='a') UNION
(SELECT sender FROM `chat` WHERE receiver='a')
```

Now I get only B and C as result.

Second, because the system does not allow users to upload their account photo, I think it might be difficult to identify who you are talking to. So, I found a website [2] that can generate a picture based on the strings sent to them, and the picture represent the identity when users are chatting to each other.

Third, I design the chat bubble for each conversation so that it can easily found which message is belong to whom. The design method is from somewhere online but I lost the URL for that site.

Everything mentioned above was what I was working on this week. Although it took me a lot of time doing those things, the result was beautiful. Hope I can finish the whole project before presentation and demonstration.

References

[1] https://www.w3schools.com/sql/sql_union.asp

[2] https://ui-avatars.com

This is entirely my own work, except as disclosed in the documentation. I gave help to the following persons:

Signed _Cheng Wei Liao_