

# **COMP3013 Database Management System**

## **Course Project Report**

### **Project Title: Online Shoes Store**

**Instructor:**

**Goliath**

**Group Members :**

**CHEN Kexuan 1930026012**

**OUYANG Runcheng 1930026102**

**YUAN Hannan 1830024337**

**HUANG Shaoxuan 1930026048**

## **1.Introduction**

The project for database management system is aims to utilize knowledge learned from the course and apply to real life problem. As E-commerce becomes a hot trend in epidemic period, demand for better database management system on online transaction has increased. Here we are addressing the problem in real online commerce and associated with actual transaction process. In this project, we use tools from PHP, JavaScript and HTML to build up website as front-end while SQL and PHP are used in back-end. We establish a comprehensive database to support the recording of transaction data. Besides, we enable user in our website manipulate, view and add data within their scope. Sincerely hoping that this project can help buyer and seller have a better experience in online transaction.

In 2018, sports shoe's market size has increased by 23% and current size is over 430 billion yuan. Nowadays, in sporting good market, shoes play an important role and it has great opportunity to grow in the following five years. The trend is concentrated in online shopping that more and more factory can promote their goods directly via online shopping mall. What's more, in the past one year, the epidemic has stimulated people's desire in purchasing goods online. It encourages consumer to have a wider search for sports shoes via internet. Meanwhile, consumer have access to different types of shoes and able to do a wider comparison and discussions. Therefore, challenge for data storage has raised and needs to overcome the data process with various metadata. Pressure for seller to manage their data is even higher as consumer needs differs.

We draw some basic assumption to modify the real transactional case and help us build up a suitable database management system for shoe's online transaction. We have assumptions on user right, accessibility and business model. Assuming all seller are joining into one venture and sell shoes as one single seller from consumer's viewpoint.

Sellers share the profit generate from website's shop, inventory and relevant information stored in database. For seller, we also give limitation on access right on customer's privacy and comment. Customer's personal information is prohibited view by seller and comment given by customer is not allowed seller to manipulate in order to make a fake image on their products. For customer, they can view all posted comments but can not modify any published comments in the website. Meanwhile, they cannot check another buyer's personal information as well.

## **2.Function**

1. As a buyer you can create a new account by imputing the buyerID, nickname, E-mail address, phoneNumber, gender, password.
2. You can login in the system as a user or seller.
3. A buyer has their own collection list. (By pressing the collection button, then go into a new web page.)
4. In the collection list web page, you can browse all the shoes information (including the shoesID which is very useful for buyer to do some operation) in the system which haven't been reserved by anyone by input the brand you want to see (NIKE, Adidas, PUMA, LI-NING) and then press the view button.
5. In the collection list web page, you can see your own collection by pressing the collection button.
6. And also, you can add some new item(shoes) to your collection list.
7. And also, you can delete some item from your collection list, too.
8. And each web page has a function that you can exit this web page.
9. As a buyer you can enter the comment web page by pressing the comments button.

10. You need to input the content you want to input and also the shoes id and then press the add comments button. And you can see all the comments from all the user in the new page.

11. A buyer has a reservation list to help them reserve some shoes.

12. You can also see your reservation list by pressing the reservation button.

13. You can add some new item to your reservation list and for the corresponding shoes will be delete in the shoe's entity automatically, and also the system will add this shoe to your collection list.

14. When you delete some shoes in the reservation the system will also add this shoe back to the shoes entity.

15. If a buyer wants to buy some shoes from his reservation, then press the purchase button to enter to a new webpage.

16. The system will show buyer's shoppinglist and his reservation for the buyer who press the button add to shoppinglist.

17. In this new webpage if you confirm you want to buy the shoes from your reservation you can press the "generate you shoppinglist now!" button. And the system will automatically avoid duplicate items which has already exist in the shopping list.

-----up functions are all buyers' functions-----

18. As a seller you can create a new account by imputing the sellingID, nickname, E-mail address, phoneNumber, gender, password.

19. You can login as a seller and enter the seller webpage and you will see five buttons (AddShoes, SubtractShoes, Check Comments, SellingRecord, Exit).

20. As a seller you can add a new shoe to the shoes entity by input the shoesID, price, brand, size. If the shoesID is duplicate then the system will tell seller.

21. The subtractShoes function need the seller input a shoesID and delete the corresponding shoes in the shoes entity.

22. The Check Commentsfunction is to ask seller to input a shoesID and then we can check the comments for this specific shoes.

23. The SellingRecord function is that when a seller finfish one order then he can delete this order in the sellingRecord entity by imputing the orderID and this webpage will also show all the selling record.

### **3.Assumption**

1. We assume that each insert or delete in our system is legal (won't input a number when they need to input a letter) and if the input is not existing in the corresponding database, then system will give users a warning. (For example, if a buyer wants to add a comment to shoes, then when he input the shoesID which is not exist then the system will be warning him the shoes is not exist. And the user won't input a number in a place need letters.)

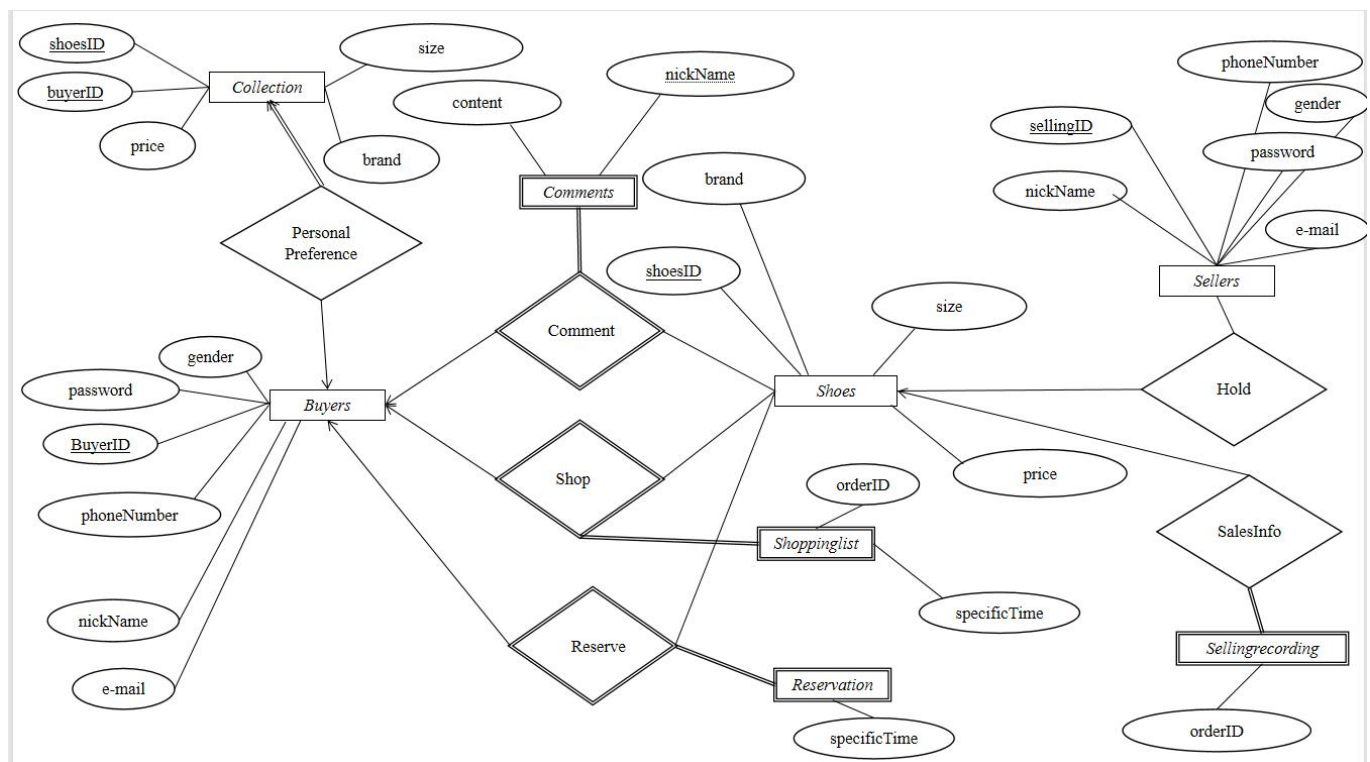
2. We assume that buyer won't delete a pair of shoes which are automatically added into collection by the system because they were added to the reservation. In the real life, in the most of case the buyer will also not delete the shoes after they add it to their reservation.

3. A buyer can only comment the same shoes for one time. Because in the real life, many app also regulate the users comment to make comment more representative and objective.

4. Seller can only see the comment but not change it. Because the sellers' comments are not objective.

5. We depart collection from reservation because consider that in real life some buyer might just like the shoes but don't intent to buy it. So, we have reservation list. Once, the buyer adds some item to reservation then other buyer cannot see them.
6. For the shopping list is that when user decide to buy something then the item must come from the reservation because the reservation means the items, they intent to buy.
7. To stimulate the real situation, we will get a specific time when we generate a new order. And because one order can contain more than one item so actually the orderID cannot be a primary key. We assume that the shoesID is the primary key. For the similar reason the sellingrecording use the shoesID as the primary, too.

#### 4. E-R diagram and description



E-R diagram

## 5. Functional dependencies and relational schemas

### Functional dependencies

(1)

Buyers = {buyerID, password, nickname, e-mail, phoneNumber, gender}

F = {buyerID → password, nickname, e-mail, phoneNumber, gender}

(2)

Sellers = {sellingID, password, nickname, e-mail, phoneNumber, gender}

F = {sellingID → password, nickname, e-mail, phoneNumber, gender}

(3)

Shoes = {shoesID, brand, size, price}

F = {shoesID → brand, size, price}

(4)

Comments = {shoesID, buyerID, nickname, content}

F = {shoesID, buyerID → nickname, content}

(5)

Collection = {buyerID, shoesID, price, size, brand}

F = {buyerID, shoesID → price, size, brand}

(6)

Reservation = {buyerID, shoesID, specificTime}

F = {shoesID → buyerID, specificTime}

(7)

Shoppinglist = {orderId, buyerID, shoesID, specificTime}

F = {shoesID → orderId, buyerID, specificTime}

(8)

Sellingrecording (orderId, shoesID)

F = {shoesID → orderId}

### Relational Schemas (underlined and in blue are primary key)

Buyers (buyerID, password, nickname, e-mail, phoneNumber, gender)

Sellers (sellingID, password, nickname, e-mail, phoneNumber, gender)

Shoes (shoesID, brand, size, price)

Comments (shoesID, buyerID, nickname, content)

Collection (buyerID, shoesID, price, size, brand)

Reservation (buyerID, shoesID, specificTime)

Shoppinglist (orderID, buyerID, [shoesID](#), specificTime)

Sellingrecording (orderID, [shoesID](#))

In SQL table will be like :

**buyerID   password   nickname   e-mail   phoneNumber   gender**  
(Buyers)

**sellingID   password   nickname   e-mail   phoneNumber   gender**  
(Sellers)

**shoesID   brand   size   price**  
(Shoes)

**shoesID   buyerID   nickname   content**  
(Comments)

**buyerID   shoesID   price   size   brand**  
(Collection)

**buyerID   shoesID   specificTime**  
(Reservation)

**orderID   buyerID   shoesID   specificTime**  
(Shoppinglist)

**orderID   shoesID**  
(Sellingrecording)



## **6. Normal form checking**

We can see from the 5 that all the functional dependence is in BCNF, so all the schema are in normal forms.

## **7.Describing the primary keys**

For Buyers: the buyerID is used to identify each entity of buyers

For Sellers: the sellingID is used to identify each entity of sellers

For Shoes: the shoesID is used to identify each entity of shoes

For Comments: the primary key is shoesID, buyerID, and this can identify the entity of comments that the specific buyers comment on the specific shoes.

For Collection: the buyerID and shoesID are the primary key, so that the specific buyer can collect the shoes. Combine this two attributes are primary key.

For Reservation: the shoesID is the primary key, and it is the only attribute distinct from other attributes.

For Shoppinglist: the shoesID is the primary key, similarly as Reservation, and each of the entity is based on the reservation of the specific shoes.

For Sellingrecording: the shoesID is the primary key and each of the specific relate to one orderID, but one orderID can contains many shoesID, which means buy many shoes for one order.

### 8.Workload for each member.

Name	ID	Work
Chen Kexuan(Oliver)	1930026012	Php function implement web page(sellers and partial buyers(connect with buyers and sellers)); Report writing; Build Database.
OuYang Runcheng(John)	1930026102	Php function implement web page(buyers); Report writing.
Yuan Haonan(Herman)	1830024337	Web page optimization(Java script, css); Report writing.
Huang Shaoxuan(Steven)	1930026048	Web page optimization(Java script, css).
JIN Baihui	1930026058	None.