Shoe Sales System

Nguyen Thanh Tuan – NGU0120

1. Task Specification

WHY?

I want to create an information system to manage a website about shoe sales system. This system describes the buying and billing process of customers and product suppliers who can supplement the product.

FOR WHAT?

Shoe Sales System will solve the selling problem, and users will choose which shoes they want to buy. This system will make it easier for the buyers to purchase and access products for payment.

WHO?

The main role of the administrator of Shoe Sales System is to check income and manage the shoes on the website (When the seller wants to sell shoes, the seller posts to the system, and the administrator will check the product on the pictures they posted).

The system has 2 types of user:

The first is the customer who wants to buy the shoe and the second is the seller.

INPUT

The system is interested in products uploaded by the seller. Products are paid and the users will be recorded in the payment history.

In case of the seller, we are interested in the product name, shoe type, size, brand, price, picture of the product, and other details. Only the seller is possible to do this, and after the seller uploads the product, the administrator will check and accept the product to be posted through the images and information provided by the seller.

In case of customers, we are interested in the username, surname, first name, address, and phone number. Users can search, choose, and buy the product. After he had chosen the product, that he wanted to buy and, proceeded with payment. The payment information will be passed to the seller, and the seller will deliver the product to the user.

OUTPUT

The output will available to everyone, even the user that is still having not created an account. The system will automatically sort the shoe and display information of the most purchased products. The user after creating an account, logging into the account, the user can select the product. If the shoes are sold out, the seller will mark the product as sold out (SOLD OUT). After the product is selected and paid, the user will be recorded in the payment history.

FUNCTION

The administrator will check the content and value of the product, if the content is spam or the product is not related to shoes, the administrator can remove it before being posted. (Example: if the seller doesn't add the shoe, but he adds e.g. clothes, watches or other things not related to the shoe, the administrator can check them and delete them). The user can search, choose, buy the product and every user can update their data.

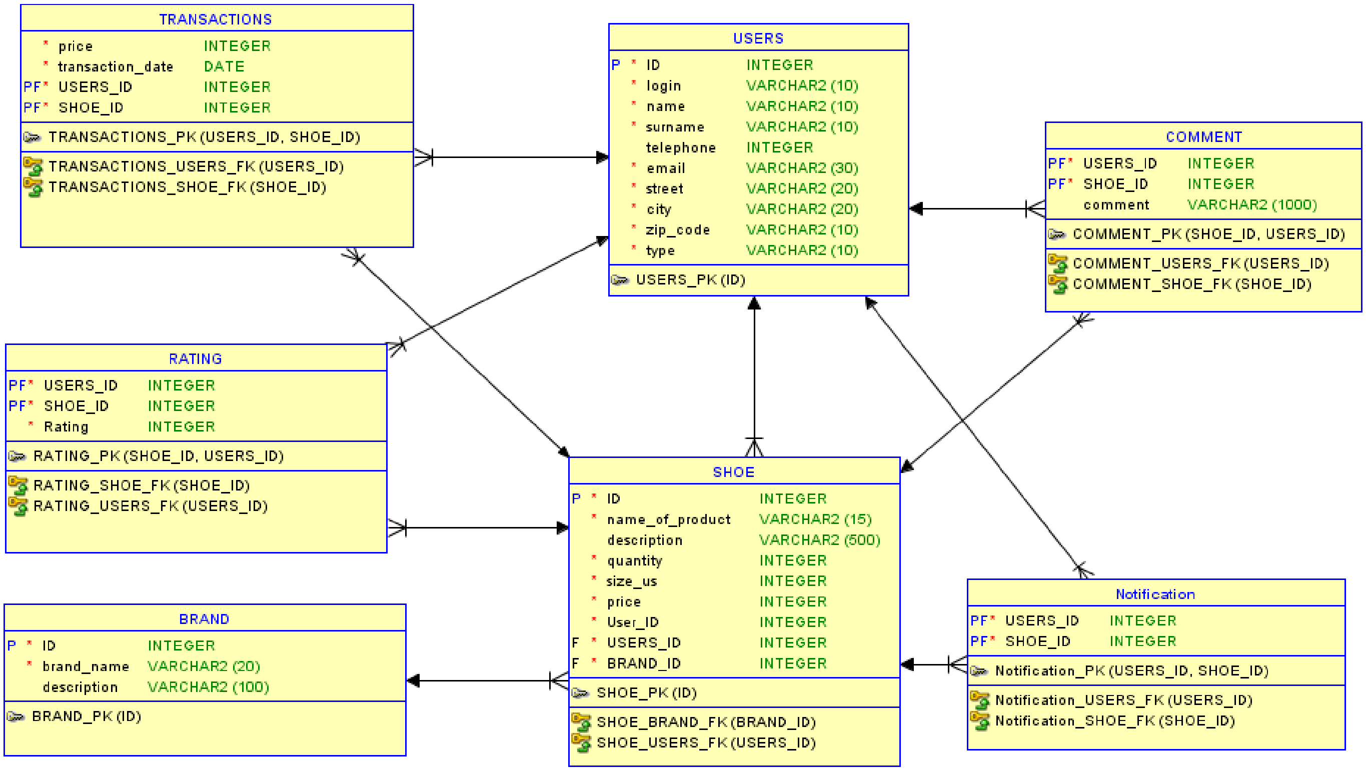
The seller can upload the product they want to sell, or remove if they do not sell it anymore, and the seller will deliver the product after the customer order on the system. When the user pays, the money will be transferred into the system. When the user receives the product, the money will be transferred to the seller.

The system will keep the money of the customer until the administrator confirms that the customer has received the product and the money will be given to the seller, if the customer does not receive the product, the money will be refunded to the customer.

The seller will set the product price, the product price can not be changed. If the seller wants to change the price, the seller will have to delete the current product and post the deleted product with the new price.

2. Conceptual Model

ER Diagram



TRANSACTIONS
```
   *  price             INTEGER
   *  transaction_date  DATE
PF*   USERS_ID          INTEGER
PF*   SHOE_ID           INTEGER
TRANSACTIONS_PK (USERS_ID, SHOE_ID)
TRANSACTIONS_USERS_FK (USERS_ID)
TRANSACTIONS_SHOE_FK (SHOE_ID)
```

USERS
```
P  *  ID          INTEGER
   *  login       VARCHAR2 (10)
   *  name        VARCHAR2 (10)
   *  surname     VARCHAR2 (10)
      telephone   INTEGER
   *  email       VARCHAR2 (30)
   *  street      VARCHAR2 (20)
   *  city        VARCHAR2 (20)
   *  zip_code    VARCHAR2 (10)
   *  type        VARCHAR2 (10)
USERS_PK (ID)
```

COMMENT
```
PF*  USERS_ID   INTEGER
PF*  SHOE_ID    INTEGER
     comment    VARCHAR2 (1000)
COMMENT_PK (SHOE_ID, USERS_ID)
COMMENT_USERS_FK (USERS_ID)
COMMENT_SHOE_FK (SHOE_ID)
```

RATING
```
PF*  USERS_ID   INTEGER
PF*  SHOE_ID    INTEGER
  *  Rating     INTEGER
RATING_PK (SHOE_ID, USERS_ID)
RATING_SHOE_FK (SHOE_ID)
RATING_USERS_FK (USERS_ID)
```

SHOE
```
P  *  ID               INTEGER
   *  name_of_product  VARCHAR2 (15)
      description      VARCHAR2 (500)
   *  quantity         INTEGER
   *  size_us          INTEGER
   *  price            INTEGER
   *  User_ID          INTEGER
F  *  USERS_ID         INTEGER
F  *  BRAND_ID         INTEGER
SHOE_PK (ID)
SHOE_BRAND_FK (BRAND_ID)
SHOE_USERS_FK (USERS_ID)
```

BRAND
```
P  *  ID           INTEGER
   *  brand_name   VARCHAR2 (20)
      description  VARCHAR2 (100)
BRAND_PK (ID)
```

Notification
```
PF*  USERS_ID   INTEGER
PF*  SHOE_ID    INTEGER
Notification_PK (USERS_ID, SHOE_ID)
Notification_USERS_FK (USERS_ID)
Notification_SHOE_FK (SHOE_ID)
```

Entity Types

Legend: **Table**, Primary Key, *Foreign Key*, attribute

USER(ID, Login, Name, Surname, Sreet, City, Zip_code, Telephone, Email, Type)

SHOE (ID, Name_of_product, Description, quantity, Size_US, Price, *User_ID*, *Brand_ID*)

RATING (*User_ID, SHOE_ID,* rating**)**

BRAND (ID, brand_name, Description)

COMMENT (*User_ID*, *SHOE_ID*, Comment)

TRANSACTION (*User_ID*, *SHOE_ID*, Price, Transaction_date)

NOTIFICATION (*User_ID,SHOE_ID*)

3. Data Model

Description of tables is depicted in the following tables.

Table **USERS**

	Data Type	Length	Key	Null	IC	Description
ID	Integer		Primary	N		
login	Varchar	10		N		Login of a user
name	Varchar	20		N		User name
surname	Varchar	20		N		Last name of the user
street	Varchar	20		N		Street of the user
city	Varchar	20		N		City of the user
zip_code	Varchar	10		N		Zip Code of the user
telephone	Number	12		A		Telephone of the user
email	Varchar	30		N		Email of a user
type	Varchar	10		N	1	User type

Table **SHOE**

	Data type	Length	Key	Null	IC	Description
ID	Integer		Primary	N		
name_of_product	Varchar	15		N		Name of product
description	Varchar	100		A		A short description of product
quantity	Integer			N		Quantity of product
size_us	Integer			N	3	Size of product
price	Integer			N	2	Price of product
User_ID	Integer		FK(USER)	N		
Brand_ID	Integer		FK(BRAND)	N		

Table **BRAND**

	Data type	Length	Key	Null	IC	Description
ID	Integer		Primary	N		
brand_name	Varchar	15		N		Name of brand
description	Varchar	100		A		A short description of brand

Table **COMMENT**

	Data type	Length	Key	Null	IC	Description
User_ID	Integer		PK, FK(USER)	N		
SHOE_ID	Integer		PK, FK(SHOE)	N		
comment	Varchar	100		N		Content of the comment

Table **RATING**

	Data type	Length	Key	Null	IC	Description
User_ID	Integer		PK, FK(USER)	N		
SHOE_ID	Integer		PK, FK(SHOE)	N		
rating	Integer			N	4	Rating of the shoe

Table **TRANSACTIONS**

	Data type	Length	Key	Null	IC	Description
User_ID	Integer		PK, FK(USER)	N		
SHOE_ID	Integer		PK,FK(SHOE)	N		
price	Integer			N	2	Price of the product
transaction_date	Date			N		Date of transaction

Table **NOTIFICATION**

	Data type	Length	Key	Null	IC	Description
User_ID	Integer		PK, FK(USER)	N		
SHOE_ID	Integer		PK, FK(SHOE)	N		

Integrity Constraints:

1. type: Admin, seller or customer
2. price >= 0
3. size_us >= 4 and size_us <= 15
4. rating >= 1 and rating <= 5

5. State Analysis

- **We define these status of shoe systems:**
    - **Stocking - The stock is still there.**
    - **Sold – a customer bought the shoe.**

We define these types of shoe sales system from Customer and Seller point of view

- **Own – The new shoe is created by the seller.**
- **The Latest Shoe – The customer can view the latest shoe in main page.**
- **Choose The Brand – The customer can choose the brand which the customer want to choose.**
- **Buy A Shoe – The customer can buy the new shoe in the SHOE table.**
- **Another Shoe – The customer can view other shoes while customer is viewing current shoes.**
- **Choose Size – The customer can choose size of shoe.**

6. Functional Analysis

5.1 List of Functions

1. User Management (User: Customer, Seller)

Table: User, Responsibility: Admin

1.1 User Insert

1.2 User Update

1.3 User Delete

1.4 List of Users with a definition of a filter to search users

1.5 User Details

Responsibility: Admin

2. Shoe Management

Table: Shoe, Brand

2.1 Upload New Shoe – The new shoe will be uploaded only if price >0 and 4 <= size_us <= 15 of the shoe.

Responsibility: Seller

2.2 Shoe Update – Seller can update their data (Name shoe, description, brand,…) – price can't be changed.

Responsibility: Seller

2.3 Shoe Delete

Responsibility: Admin, seller – Admin can delete some shoe (SOLD OUT) or many product doesn't like shoe. Seller can delete some shoe (SOLD OUT) or delete to change price.

2.4 Shoe Details

Table: all, Responsibility: User(Customer, Seller)

3. List of shoes

Table: all

3.1 The Latest Shoes - a list of the latest shoes for selected brands.

3.2 List of sold shoes

3.3 List of shoes without comment

3.4 List of shoes with comment

3.5 List of shoes with rating

4. Brand Management

Table: Brand, Responsibility: Admin

4.1 New Brand

4.2 Brand Update

4.3 Brand Delete

4.4 List of Brands

5. Comment Management

Table: Comment, Responsibility: Seller, Customer

5.1 New Comment

5.2 Comment Update

5.3 Comment Delete

5.4 Comment of Seller

6. Other Functions

6.1 User Notification – when the quantity of shoe down to 0, then system automatically notify to the seller. In our system, mails are not sent, only a record is inserted in the table **Notification**. This function is described in Section 5.2

6.2 Finding Item By Keyword – when the user want to find a keyword in the store, the system automatically return results which the user want to find. This function is described in Section 5.2.

5.2 Detail Description of Functions

Function 3. List of shoes

Input: The user id $idcustomer for function 3.2-.3.4

3.1 The Latest Shoes

```
select s.* from shoe s, brand b
where s.end < CURRENT_TIMESTAMP
and b.id=idbrand and b.id=s.brand order by (CURRENT_TIMESTAMP-end) asc;
```

3.2 List of the sold shoes

```
select s.* from shoe s
where s.description = “SOLD”;
```

3.3 List of shoes without comment

```
select s.* from shoe s, brand b
where s.shoe_id =$idcustomer
and s.id not in (select shoe_id from Comment where ID_seller =$idcustomer)
and b.id=idBrand and b.id=s.subbrand;
```

3.4 List of shoes has comment

```
select s.* from shoe s, brand b
where s.user_id =$idcustomer
and s.id in (select shoe_id from Comment where ID_seller =$idcustomer)
and b.id=idBrand and b.id=s.brand;
```

3.5 List of shoes has rating

```
select s.* from shoe s, brand b
where s.user_id= $idcustomer
and s.id in (select shoe_id from Rating where ID_seller =$idcustomer)
and b.id=idBrand and b.id=s.brand;
```

Function 1.1: User Insert

Parameter: $login

Function Description: The user can create account. The account can only be inserted into the if all required information for registration is satisfied and it is a transaction.

1. The system will check the login account if it is already registered.

   Select count(*) from User where login = $login

   If result is 0, it means the login can be registered.

2. If the login can be registered, the login account can be inserted into the system.

Function 6.1: User Notification

Parameter: $idUser, $idShoe,

Function Description: This function is automatically executed in the case of insertion of a new record in the table **Transaction**. This function will change value quantity in the table **Shoe** and it is a transaction.

1. Read attribute *quantity* of the Shoe:

   Select quantity from Shoe

   Where shoeID = $idShoe

2. If quantity higher than 0 then the system will decrease it by 1

3. If quantity > 0

   then update Shoe

   set quantity = quantity -1

4. After that, if quantity is 0, then notify to seller *where userId = $idUser*. In a real system, an e-mail is sent.

Function 6.2: Finding Item By Keyword

Parameter: $keyword

Function Description: This function will execute when the user want to find the shoe but the user don't know the name of shoe, so they want to find by keyword like Nike, Adidas,… in the table Shoe and it is a transaction.

1. The user will find the shoe by the keyword.

   Select name from Shoe

   Where name like %$keyword%

The system will check the quantity of the shoe, the quantity of the shoe which the user wants to find the expected shoe.

And where name =

(Select name, quantity from Shoe

Where name like %$keyword%)

2. The user will find the shoe by the expected price.

   Select price from Shoe

   Where price <= $expectPrice

3. Depend on the number of data provided by users, it is more uncomplicated for both sides, the system can find out an expected product as soon as it can also user can receive more product.

6. Design of User Interface

6.1 Menu

1. Shoe (Responsibility: Admin, User: Seller, Customer)

(a) The Latest Shoes – action: 3.1 The Latest Shoes

(b) List of the shoe

(c) Famous brand – action: 4.4 List of Brand

(d) Popular shoe

2. New Shoe (Responsibility: User: Seller) – action: 2.1 Upload New Shoe

3. My Profile (Responsibility: User: Seller, Customer) – action 1.5: User detail

4. Administration (Responsibility: Admin)

(a) User management (User: Seller, customer)

i. User insert – action 1.1: User insert

ii. List of User – action 1.4: List of User

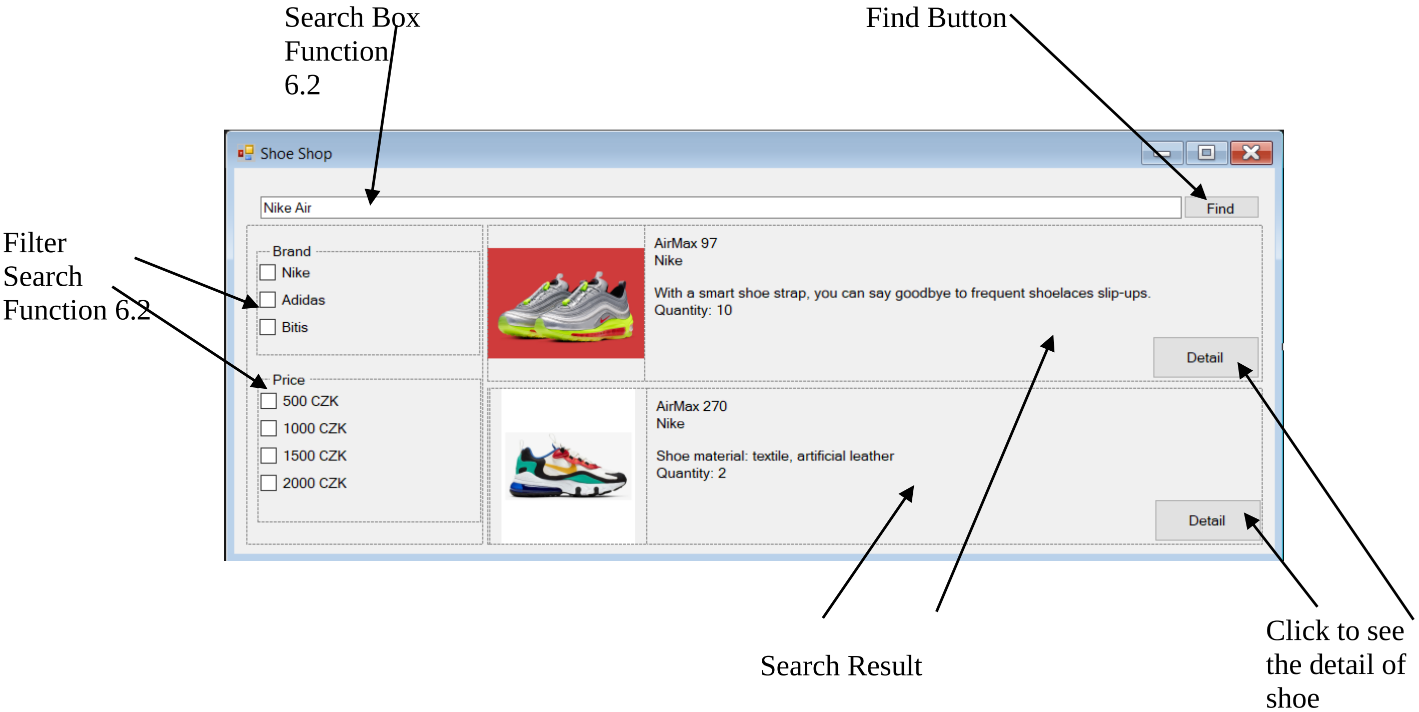
These action are offers for each users: 1.2 User update, 1.3 User delete, 1.5 User detail

(b) Brand management – similar to the user management

6.2 Shoe Detail Form

Function 6.2: Finding Item by Keyword

Description: When you want to search for a product related to a similar name you are looking for, you can enter the search field to find the product you are looking for. Or if you have a little money and want to buy shoes at the requested price, you can choose the required price of shoes. If you want to know details about the shoe, you can click the Detail button.



Nike Air

□ Nike
□ Adidas
□ Bitis

□ 500 CZK
□ 1000 CZK
□ 1500 CZK
□ 2000 CZK

AirMax 97
Nike

With a smart shoe strap, you can say goodbye to frequent shoelaces slip-ups.
Quantity: 10

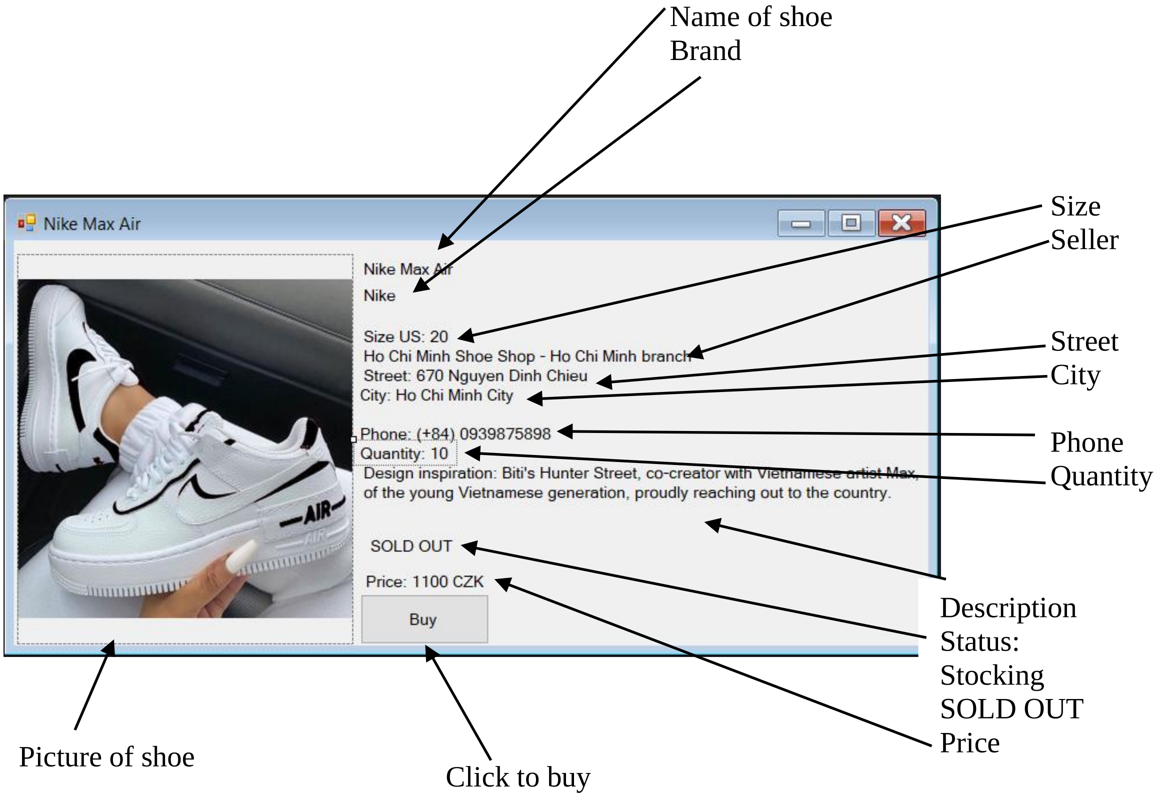
AirMax 270
Nike

Shoe material: textile, artificial leather
Quantity: 2

Picture 1: Finding Item by Keyword

Function 2.4: Shoe Detail

Description: This is the UI of the detail. You can see the detail of the shoe, you can see more information of shoe like Name, Brand, Size of shoe,... If you like, you can click to Buy button. After that, the user will be redirected to the checkout page, but the payment method is not supported in our system. So we will record this transaction into the Transaction table instead.



Nike Max Air

Nike Max Air

Nike

Size US: 20

Ho Chi Minh Shoe Shop - Ho Chi Minh branch

Street: 670 Nguyen Dinh Chieu

City: Ho Chi Minh City

Phone: (+84) 0939875898

Quantity: 10

Design inspiration: Biti's Hunter Street, co-creator with Vietnamese artist Max, of the young Vietnamese generation, proudly reaching out to the country.

SOLD OUT

Price: 1100 CZK

Picture 2: Shoe Detail