Database Final Report

Group members:

Yutong Zhang	yuz192	yuz192@pitt.edu
Enmin Li	enl29	enl29@pitt.edu
Yijun Zhou	yiz209	yiz209@pitt.edu

1. overview of the e-commerce CD system

An E-commerce system is the product of rapid technological development Users, and its use greatly facilitates people's shopping life. Our group focuses on music records, hoping to develop a platform that allows music lovers to purchase CDs conveniently and quickly.

Our system can guarantee the regular use of users and managers at the same time. Users can create an account, log in to the website, purchase goods, and pay online. For the convenience of customers, all users are divided into two types-business and home, each type of user needs to fill in the different information. When we talk about managers, they are divided into three categories, each responsible for different scopes. A manager is responsible for the regular operation of the entire system; each region has a regional manager. Each store also has one and only one responsible manager called store-manager.

2. Assumptions

(1)region & store:

In the CD system, each store belongs to a specific region, and a region includes many stores. We create two regions: “America” and “Asia”; the region of “America” includes “New York” store and “Los Angeles” store; while the region of “Asia” includes “Tokyo” store and “Hong Kong” store.

(2) store & product:

The inventory of products in different stores is different, and users should choose the store first before purchasing goods.

3. E-R diagram

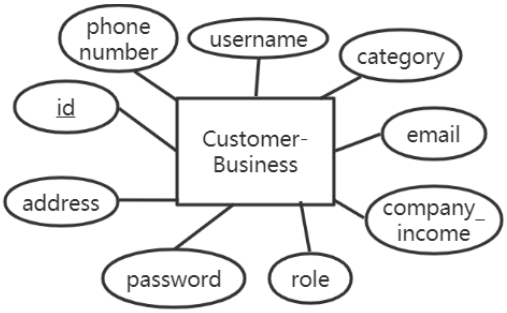


Figure 1: E-R diagram of Business Customer

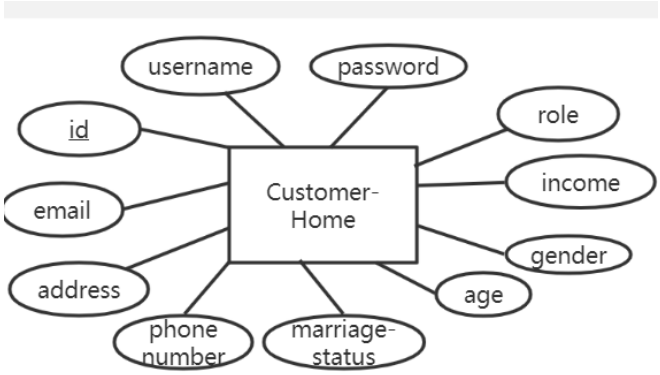


Figure 2: E-R diagram of Home Customer

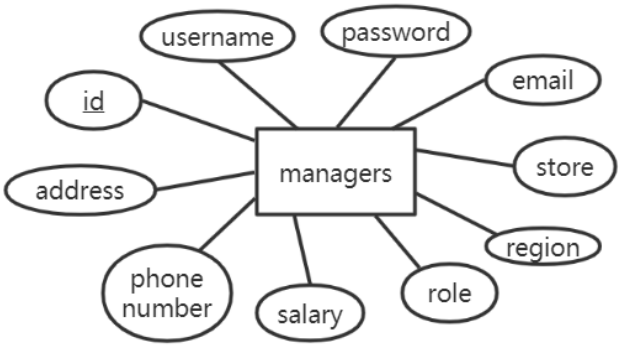


Figure 3: E-R diagram of managers

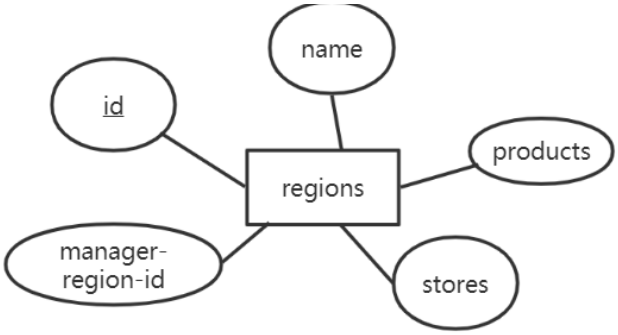


Figure 4: E-R diagram of regions

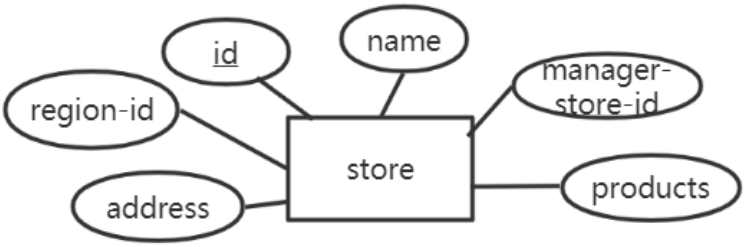


Figure 5: E-R diagram of stores

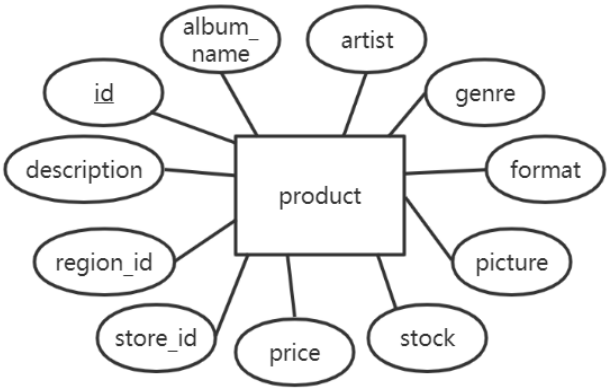


Figure 6: E-R diagram of products

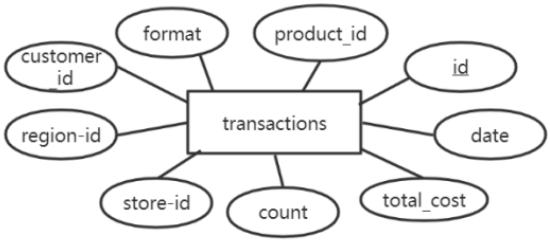


Figure 7: E-R diagram of transactions

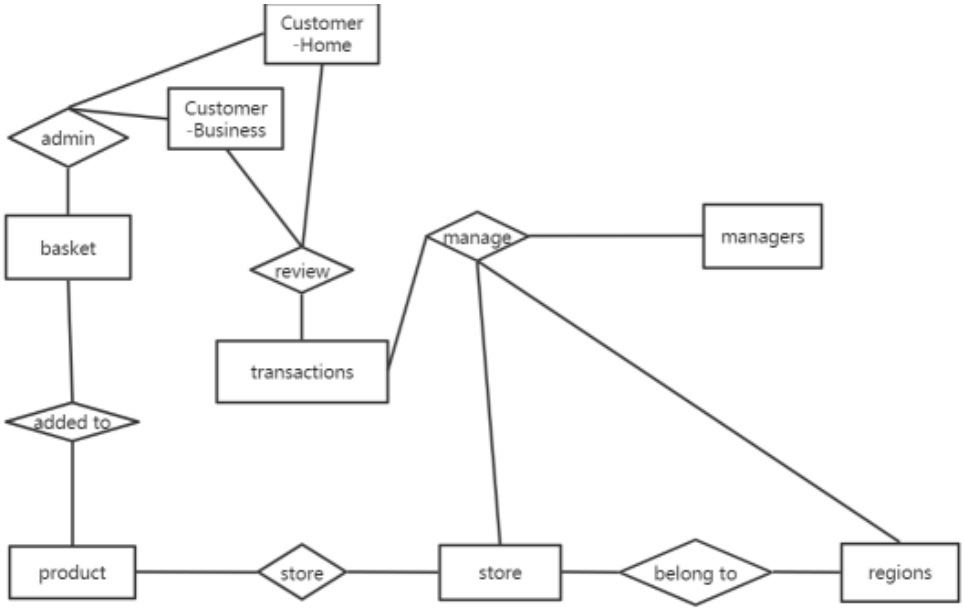
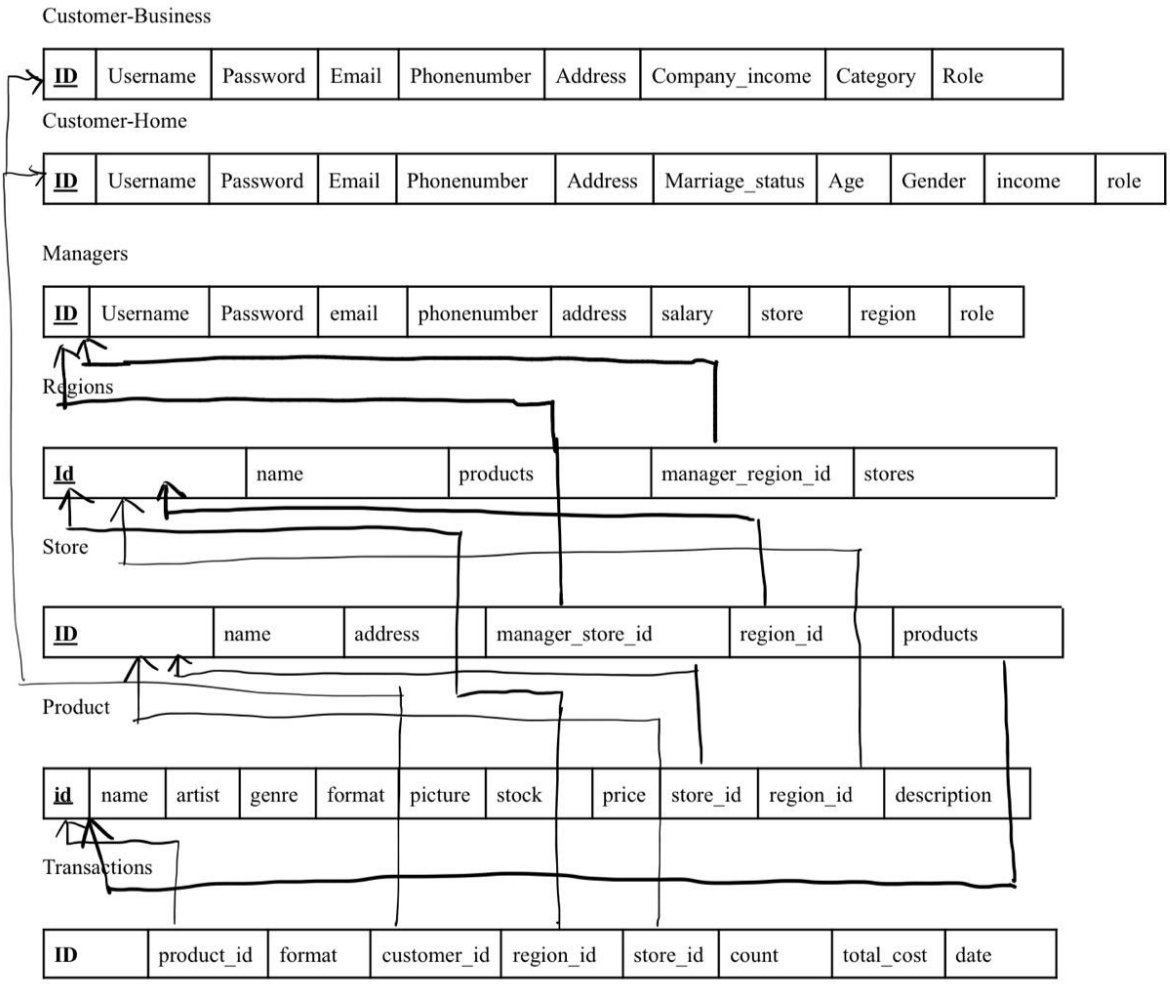


Figure 8: E-R diagram of system

4. relation schema



5. DDL statements with identification and justification of the Normal Form

```
CREATE TABLE IF NOT EXISTS ecommercecd.customers_business (
    id INTEGER NOT NULL,
    username VARCHAR(100),
    password VARCHAR(20),
    email VARCHAR(100),
    address VARCHAR(100),
    phoneno VARCHAR(100),
    category VARCHAR(100),
    company_income FLOAT, role VARCHAR(20),
    PRIMARY KEY (id),
    UNIQUE (email),
    UNIQUE (username)
);
```

In the table of customers_business,

the CANDIDATE KEY includes{'id','email','username'}

For each non-trivial FD x->y,if the relation is BCNF, x should be a super key.

In this relation,all x are super keys,

so the table customers_business is BCNF

```
CREATE TABLE IF NOT EXISTS ecommercecd.customers_home (
    id INTEGER NOT NULL,
    username VARCHAR(100),
    password VARCHAR(20),
    email VARCHAR(100),
    address VARCHAR(100),
    phoneno VARCHAR(100),
    marriage_status VARCHAR(100),
    age INTEGER,
    gender VARCHAR(100),
    income FLOAT, role VARCHAR(20),
    PRIMARY KEY (id),
    UNIQUE (email),
    UNIQUE (username)
);
```

In the table of customers_home,

the CANDIDATE KEY includes{'id','email','username'}

For each non-trivial FD x->y,if the relation is BCNF,x should be a super key.

In this relation,all x are super keys,

so it is BCNF

```
CREATE TABLE IF NOT EXISTS ecommercecd.managers (
    id INTEGER NOT NULL,
    username VARCHAR(100),
    password VARCHAR(20),
    email VARCHAR(100),
    address VARCHAR(100),
    phoneno VARCHAR(100),
    role VARCHAR(100),
    region VARCHAR(100),
    store VARCHAR(100),
    PRIMARY KEY (id)
);
```

In the table of managers,

the CANDIDATE KEY includes{'id'}

For each non-trivial FD x->y,if the relation is BCNF,x should be a super key.

In this relation,

there is a FD{store->region}, in which 'store' is not a candidate key,

so the table managers is not BCNF;

Besides, if a relation is 3NF, there is no transitive dependency.

For FD{id->store,store->region},'id' is a candidate key, but 'store' is not a primate attribute, so there is a transitive dependency in this relation,and the table managers is not 3NF;

Also, if a relationship is 2NF, there is no partial dependency.

In this relation, there is no partial dependency,

so it is 2NF.

```
CREATE TABLE IF NOT EXISTS ecommercecd.products (
    id INTEGER NOT NULL,
    album_name VARCHAR(100),
    artist VARCHAR(100),
    genre VARCHAR(100),
    cdformat VARCHAR(20),
    picture VARCHAR(200),
    description VARCHAR(1000),
    region_id INTEGER,
    store_id INTEGER,
    price FLOAT,
    stock INTEGER,
    PRIMARY KEY (id)
);
```

In the table of products,

the CANDIDATE KEY includes{'id'}

For each non-trivial FD x->y,if the relation is BCNF,x should be a super key.

In this relation,

there is a FD{store->region}, in which 'store' is not a candidate key,

so the table products is not BCNF;

Besides, if a relation is 3NF, there is no transitive dependency.

For FD{id->store,store->region},'id' is a candidate key, but 'store' is not a primate attribute, ,so there is a transitive dependency in this relation,and the table products is not 3NF;

Also, if a relationship is 2NF, there is no partial dependency.

In this relation, there is no partial dependency,

so it is 2NF.

```
CREATE TABLE IF NOT EXISTS ecommercecd.regions (
    id INTEGER NOT NULL,
    name VARCHAR(100),
    manager_region_id INTEGER,
    PRIMARY KEY (id),
    UNIQUE (name)
);
```

In the table of regions,

the CANDIDATE KEY includes{'id','name'}

For each non-trivial FD x->y,if the relation is BCNF,x should be a super key.

In this relation,

all x are super keys

so it is BCNF.

```
CREATE TABLE IF NOT EXISTS ecommercecd.stores (
    id INTEGER NOT NULL,
    name VARCHAR(100),
    manager_store_id VARCHAR(100),
    region_id INTEGER,
    PRIMARY KEY (id),
    FOREIGN KEY(region_id) REFERENCES regions (id),
    UNIQUE (name)
);
```

In the table of stores,

the CANDIDATE KEY includes{'id','name'}

For each non-trivial FD x->y,if the relation is BCNF,x should be a super key.

In this relation,

there is FD{manager_store_id->region_id}, in which 'manager_store_id' is not a super key,

so the table customers_products is not BCNF;

Besides, if a relation is 3NF, there is no transitive dependency.

In this relation,

there is FD{id->manager_store_Id,manager_store_id->region_id},in which 'id' is a candidate key but 'manager_store_id' is not a primate attribute,

so it is not 3NF.

Also, if a relationship is 2NF, there is no partial dependency.

In this relation, there is no partial dependency,

so it is 2NF.

```
CREATE TABLE IF NOT EXISTS ecommercecd.transactions (
    id INTEGER NOT NULL,
    product_id VARCHAR(100),
    cdformat VARCHAR(20),
    customer_id INTEGER,
    region_id INTEGER,
    store_id INTEGER,
    count INTEGER,
    total_cost FLOAT,
    trandate VARCHAR(30),
    PRIMARY KEY (id)
);
```

In the table of transactions,

the CANDIDATE KEY includes{'id'}

For each non-trivial FD x->y,if the relation is BCNF,x should be a super key.

In this relation,

there is FD{product_id-> cd_format, product_id->store_id, store_id->region_id}, in which 'product_id' and 'store_id' are not super keys,

so the table transactions is not BCNF;

Besides, if a relation is 3NF, there is no transitive dependency.

In this relation,

there is FD{id->produt_id,product_id->cd_format, product_id->store_id},in which 'id' is a candidate key but 'product_id' is not a primate attribute,

so it is not 3NF.

Also, if a relationship is 2NF, there is no partial dependency.

In this relation, there is no partial dependency,

so it is 2NF.

6. Description of front-end design and connection

The front-end uses Bootstrap, which is an open source toolkit for front-end development launched by Twitter.

The backend uses Flask, which is a micro-framework for Python development that relies on jinja2 templates and Werkzeug WSGI services. Werkzeug is used to process the Socket service, which is used in Flask to accept and process http requests; Jinja2 is used to process the template, render the template and data, and return it to the user's browser.

Combine Flask and Bootstrap through the correspondence between URL and Python function, change the IP address and port of the service provided by Flask, return the template through Flask, and other operations. Flask is responsible for processing user requests and calling some python programs. The Bootstrap framework provides HTML files to provide users with page display.

7. System implementation with example screen shoots

(1)users can create a new consumer account

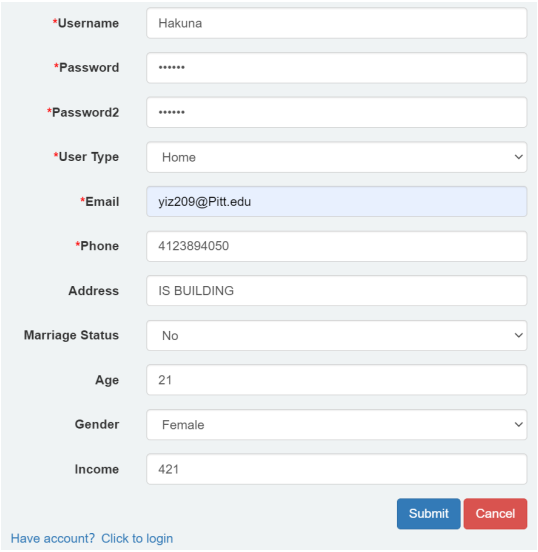


Figure 9 : registration interface

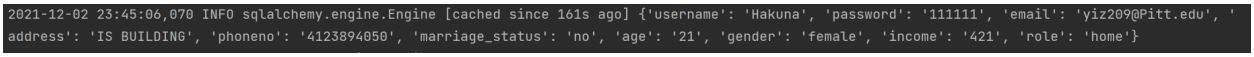


Figure 10 : background data of successful registration

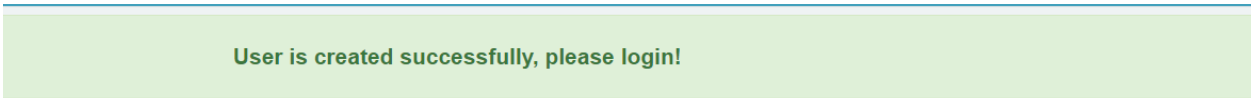
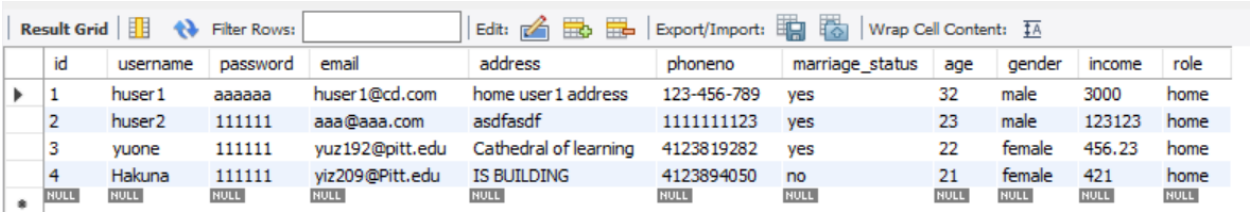


Figure 11 : successful registration reminder



id	username	password	email	address	phoneno	marriage_status	age	gender	income	role
1	huser1	aaaaaa	huser1@cd.com	home user1 address	123-456-789	yes	32	male	3000	home
2	huser2	111111	aaa@aaa.com	asdfasdf	1111111123	yes	23	male	123123	home
3	yuone	111111	yuz192@pitt.edu	Cathedral of learning	4123819282	yes	22	female	456.23	home
4	Hakuna	111111	yiz209@Pitt.edu	IS BUILDING	4123894050	no	21	female	421	home
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 12 : mySQL data of successful registration

(2)Users can update information and delete account

id	username	password	email	address	phoneno	marriage_status	age	gender	income	role
1	huser1	aaaaaa	huser1@cd.com	home user1 address	123-456-789	yes	32	male	3000	home
2	huser2	111111	aaa@aaa.com	asdfasdf	1111111123	yes	23	male	123123	home
3	yuone	111111	yuz192@pitt.edu	Cathedral of learning	4123819282	yes	22	female	456.23	home
4	Hakuna	111111	yiz209@Pitt.edu	320 McKee Place	4123894050	no	21	female	421	home

Figure 13: user “Hakuna” changed her address from ‘IS Building’ to ‘320 McKee Place’

id	username	password	email	address	phoneno	marriage_status	age	gender	income	role
1	huser1	aaaaaa	huser1@cd.com	home user1 address	123-456-789	yes	32	male	3000	home
2	huser2	111111	aaa@aaa.com	asdfasdf	1111111123	yes	23	male	123123	home
3	yuone	111111	yuz192@pitt.edu	Cathedral of learning	4123819282	yes	22	female	456.23	home
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 14: user”Hakuna” deleted her account

(3)Managers can create,update,delete new store

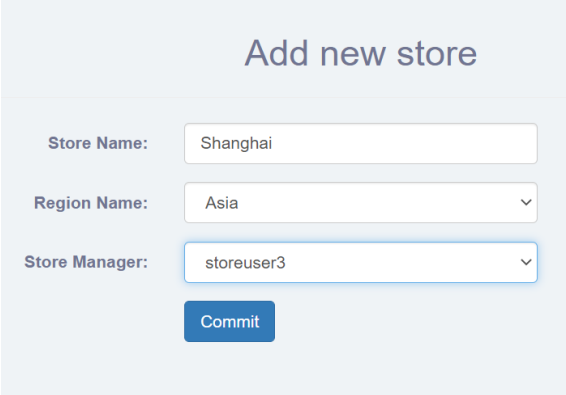
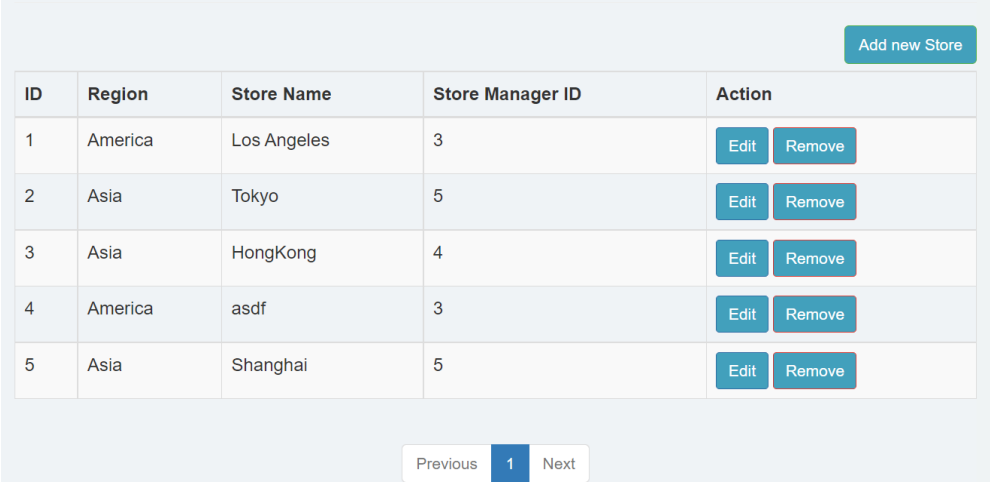


Figure 15 : New store adding interface



Store management interface

(4)Users can buy product

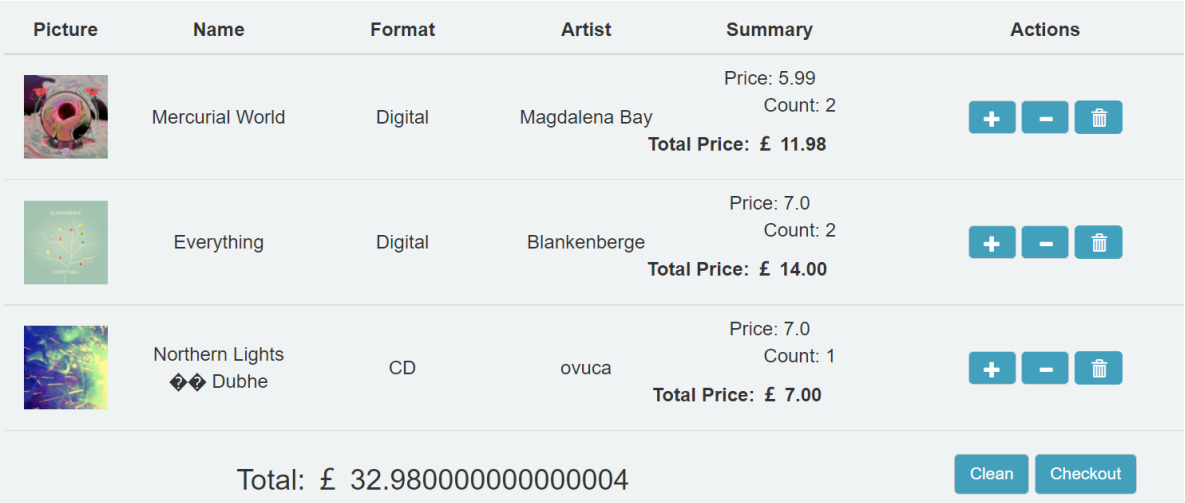


Figure 16 : basket interface

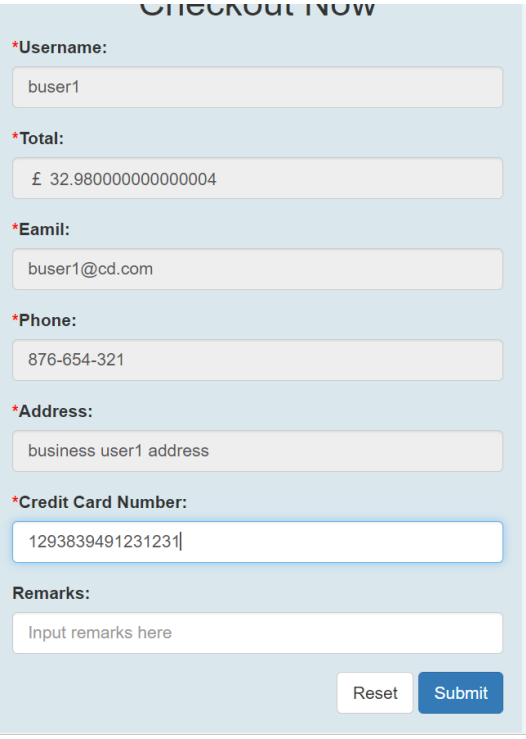


Figure 17 : Checkout interface

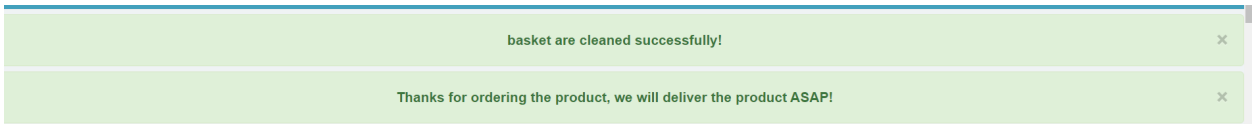


Figure 18 : Reminder of successful payment

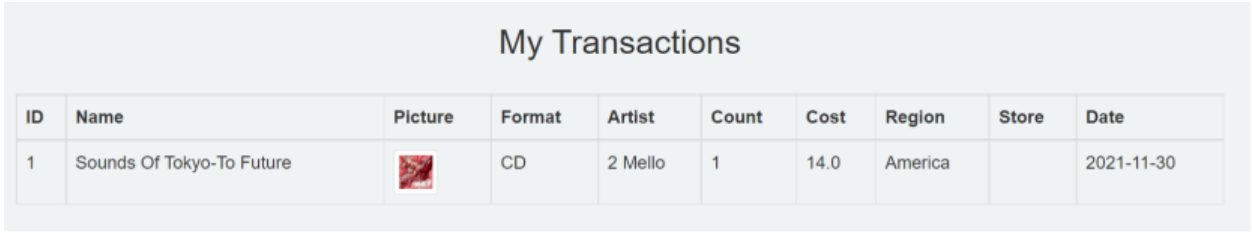


Figure 19 : the user's transaction after successful payment

(5)Users can review past orders

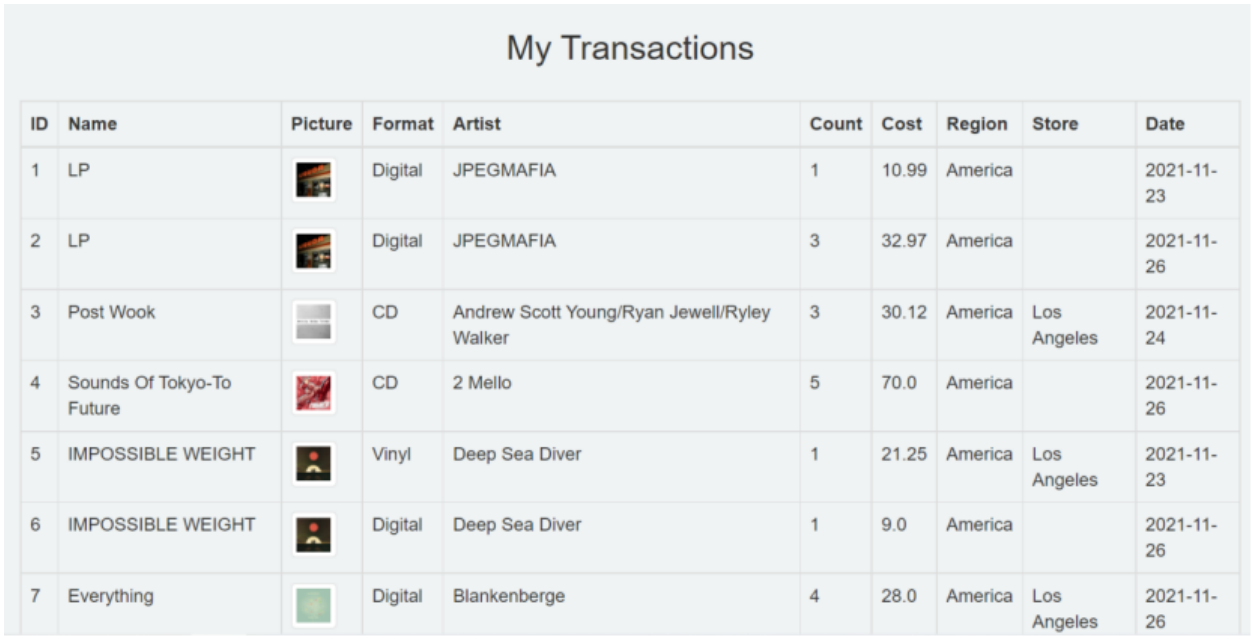


Figure 20: transactions reviewing interface

(6)Store managers can add,update,delete new products

Figure 21: product management interface

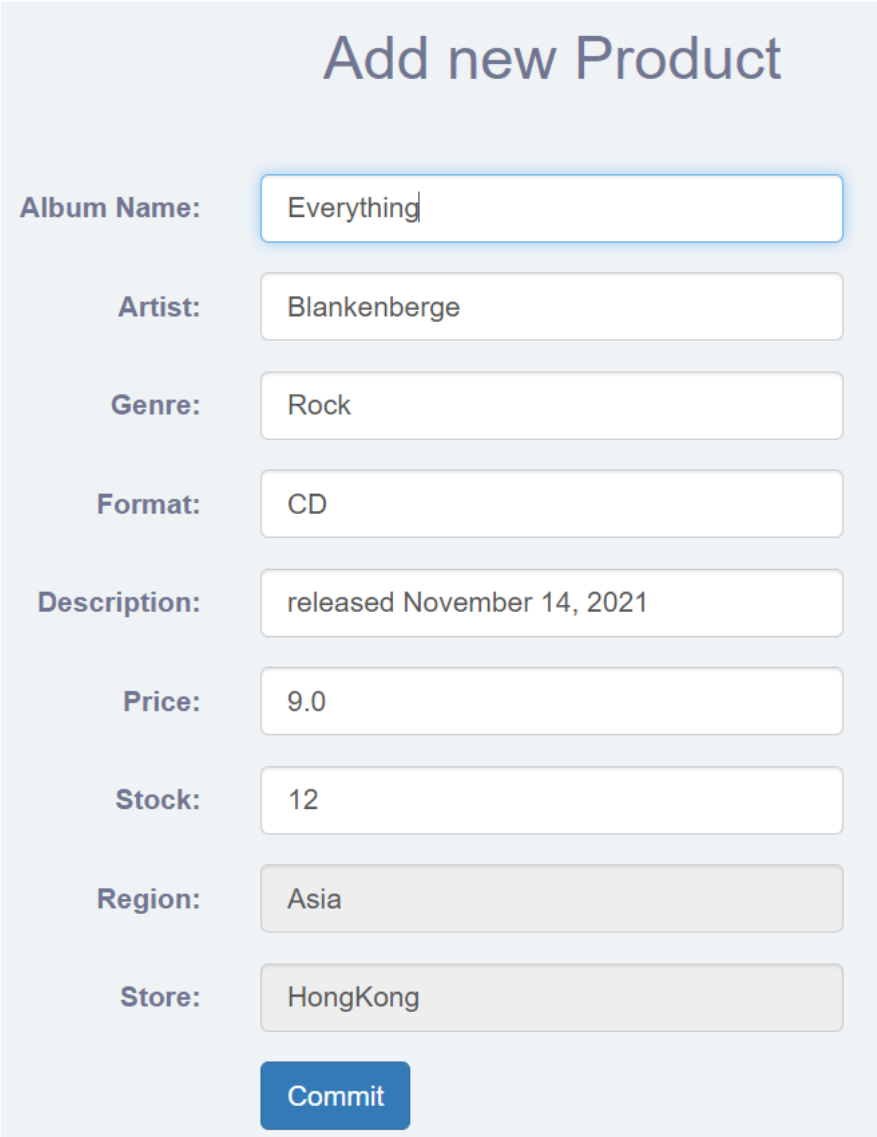


Figure 22: new product adding interface

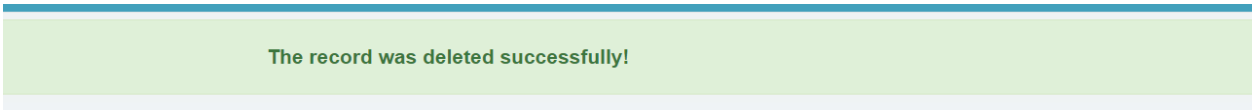


Figure 23: reminder of product removing

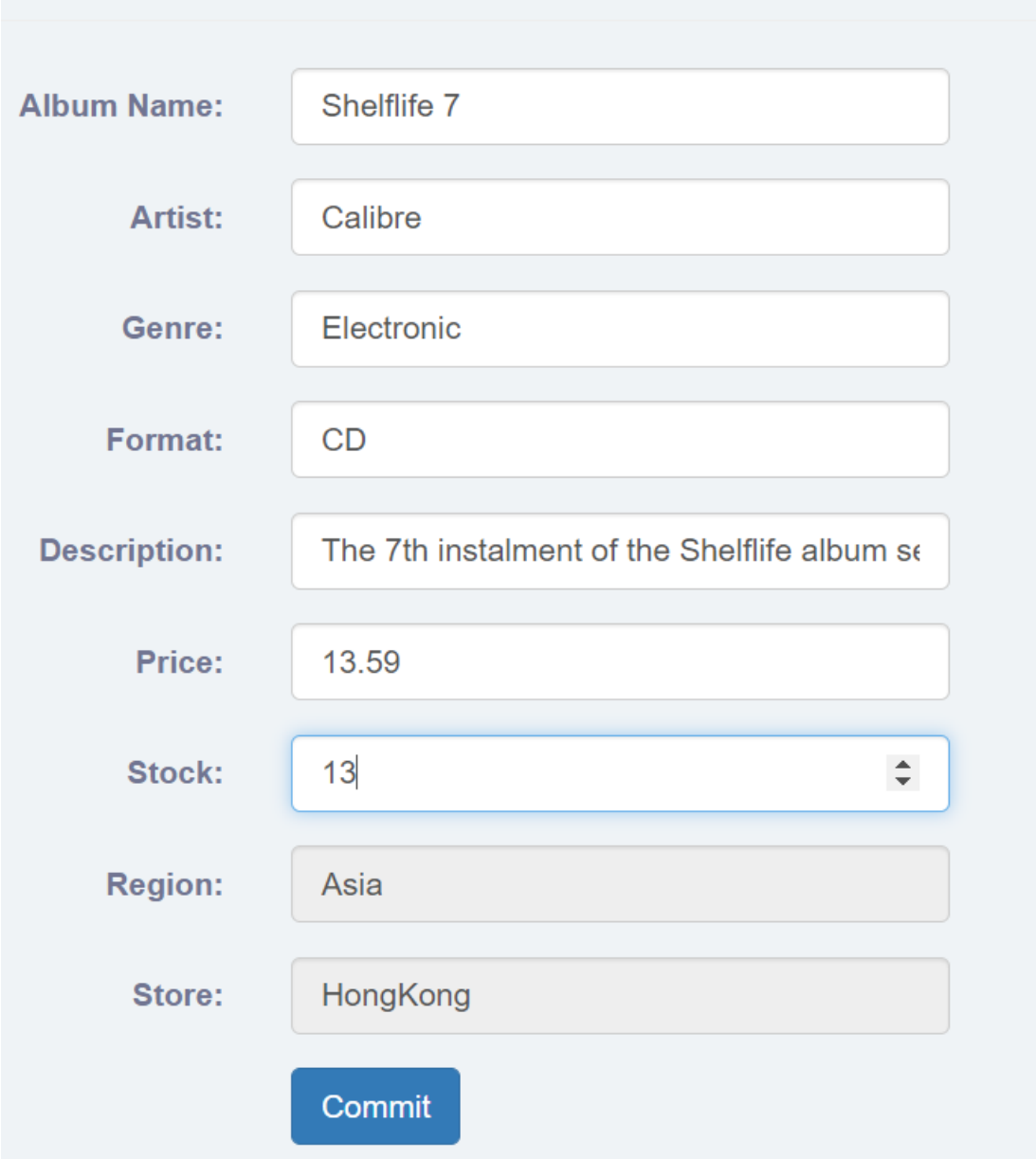


Figure 24: update the number of products

(7)If a kind of product in a store is sold out, the system should prompt the user that the store is out of stock

Figure 25: the product which we will test the function(7)

Figure 26 : the stock of this CD is 1, the user tries to buy 2 of it.

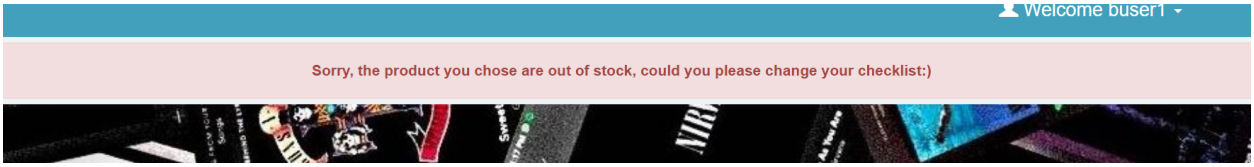


Figure 27: reminder of checkout failed

(8)The system should check whether the wrong information is entered during each creation process of the user

This function is used to check whether the user information input meets the specifications:

- the user name must be greater than three characters and less than fifteen characters;

- the password must be equal twice and contain uppercase and lowercase and numbers,
- the mailbox must conform to the correct format,
- the mobile phone number must conform to the correct format

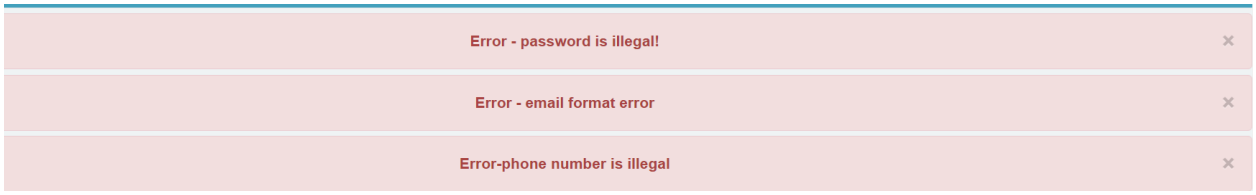


Figure 28: the reminder of registration failed

(9)The system can prompt error when a primary-foreign key has been changed

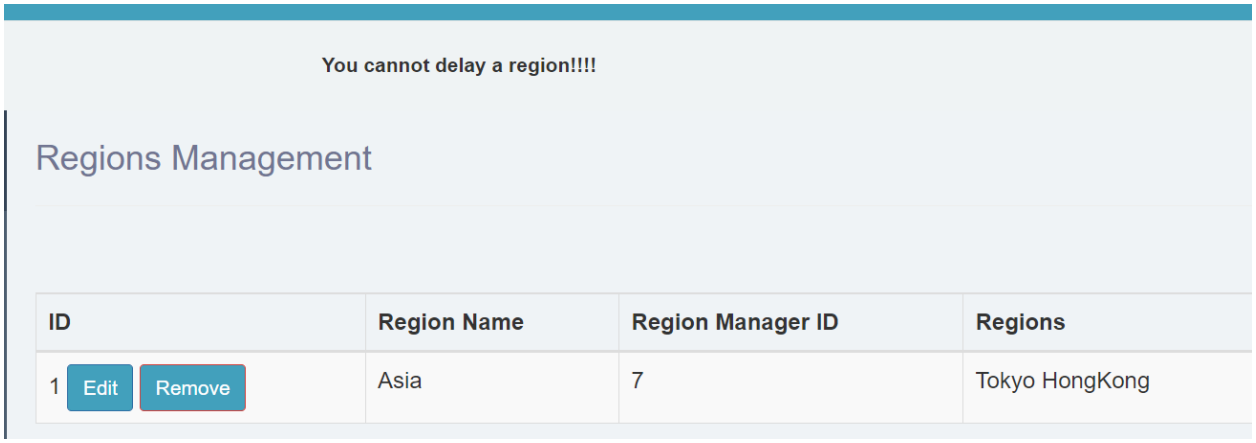


Figure 28: Region_id is foreign key for the table of store. If the manager want to remove a region,the system will stop this procession and provide a reminder

(10)The manager can get access to data aggregation query module

Total Sales Per Day for Region: All Region

ID	Trading Day	Total Sales
1	2021-12-03	$ 32.97999954223633
2	2021-11-27	$ 32.970001220703125
3	2021-11-26	$ 202.97000122070312
4	2021-11-24	$ 30.1200008392334
5	2021-11-23	$ 32.239999771118164

Figure 29: Total sales per day

(11)users can search products

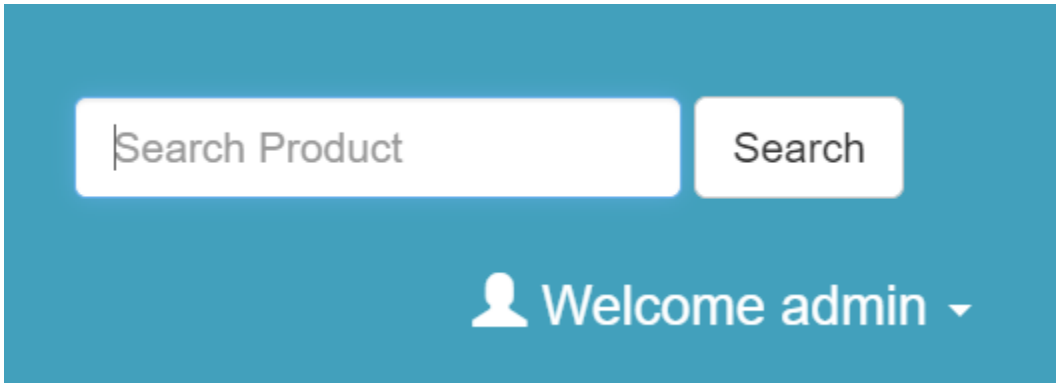


Figure 30: Search interface

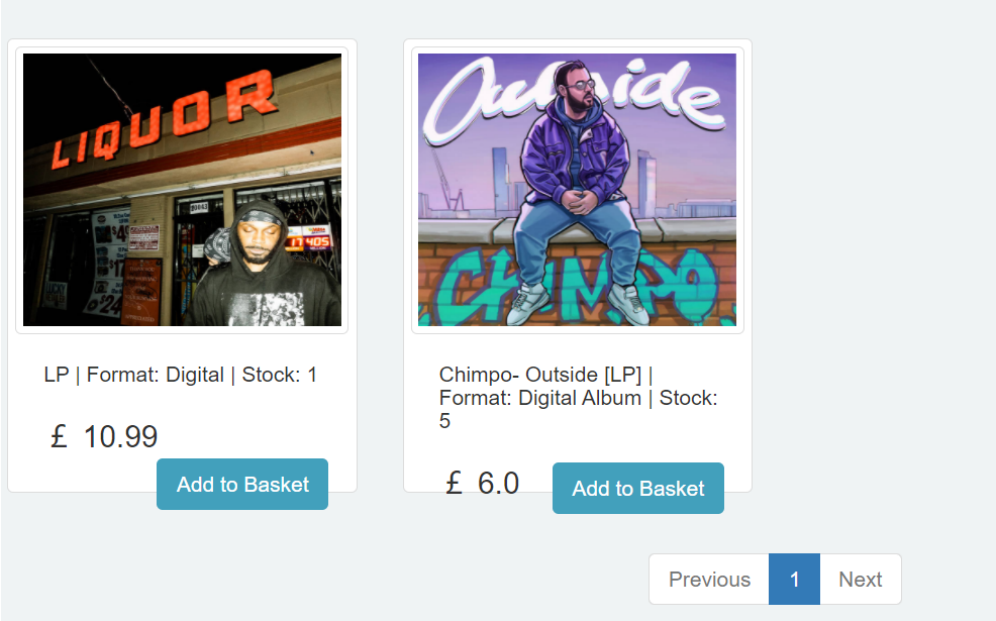


Figure 31: Search result

8. Testing efforts and erroneous cases

（1）Testing efforts

After the initial completion of the system, it will be tested functionally, starting with the registration function. When the user enters the required information, immediately use js to verify whether the form information entered by the user is legal.

During verification, judge whether the browser supports HtMLS. If it does, use the HTML5 form verification framework. If it does not, use js to verify the user information. After successful verification, click the "register" button to register. After successful registration, you will jump to login. Interface, you can log in with the account you just registered.

After the login is successful, it will automatically jump to the homepage. The homepage can view the product grouping list, the grouping corresponding product list, and the product pagination view.

The user enters the search criteria in the search box on the homepage and clicks the "Search" button to search.

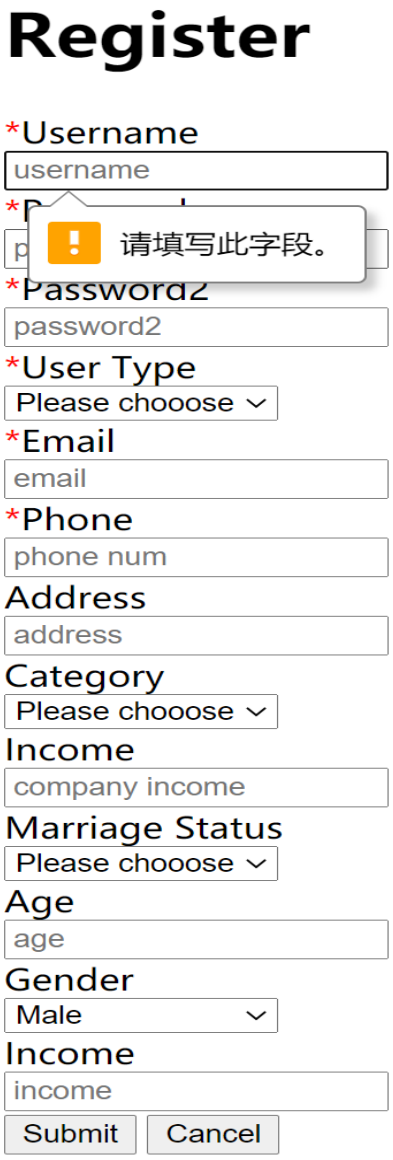
After clicking on a product, you can go to the "Product Details" interface to view the detailed information of the product.

Click the "Add to Cart" button on the "Product Details" interface to verify the inventory of the product. If the inventory is sufficient, the addition is successful and it will automatically jump to the "Add Successful" interface.

Click "My Shopping Cart" on the homepage or "Successful Addition" interface to view the shopping cart. When viewing the shopping cart, the total price is counted, and the items that have been added to the shopping cart can be deleted from the shopping cart**.**

（2）erroneous cases

All forms cannot be empty during registration; otherwise, the registration fails. The password must be the same as password 2; otherwise, an error message is displayed.The email address must meet the standard format XXX @. Otherwise, the system displays that registration cannot be performed.



9. Limitations and the possibilities for improvements

- Improve the design of the database, and modify the 2nf-compliant tables to 3nf and bcnf-compliant tables.
- The UI design can be more refined and increase the interface for CD audio playback.
- In addition, you can add a payment function on the checkout page to simulate reality more realistically.

- This system can only be run locally at present, and the domain name URL and virtual host can be bound later, and then the web page files can be uploaded to the virtual host for public access.
- Adopt front-end and back-end separation technology to reduce the pressure on the server.