

# CS307 Project 2

12111448 侯芳旻

12112701 刘子豪

Lab session: Lab 2

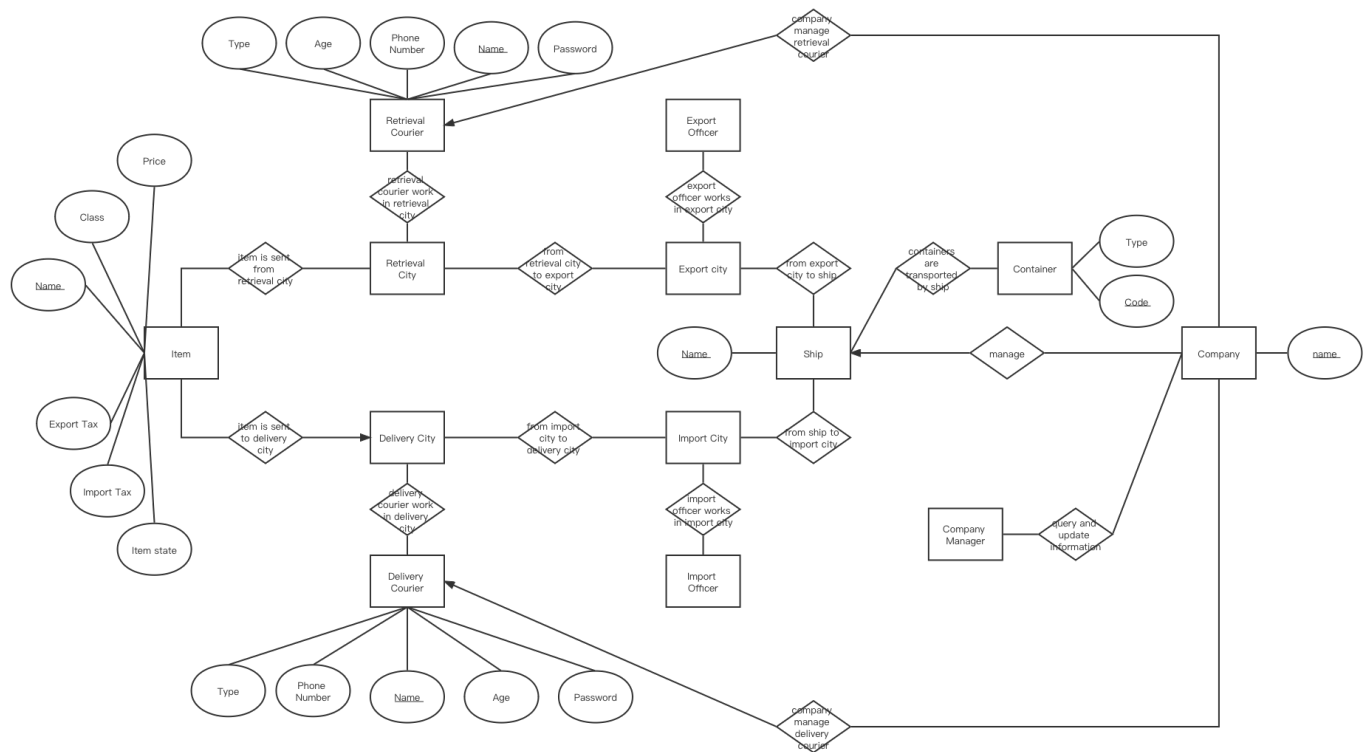
Lab teacher: Weiyu Wang

## contribution

task	侯芳旻	刘子豪
Database design	1	
Table Manipulation		1
SUSTC Department Manager User	1	
Courier User	1	
Company Manager User		1
Seaport Officer User		1
Advanced APIs and Other Requirements	1	
write report	1	
Contribution	50%	50%

## 1 database design

### ER diagram



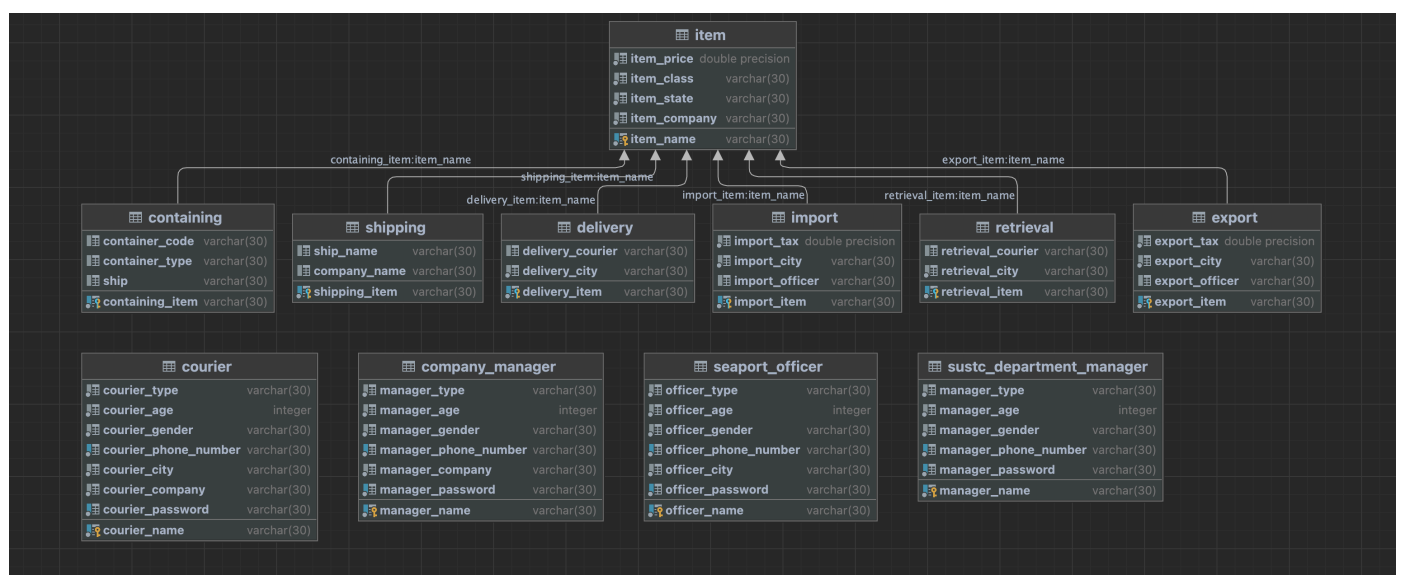
From the ER diagram created by our group, it is clear to see that the relationship between two different items are all different.

And as our design follows the third normal form:

- Unique rows
- Scalar columns
- Every non-prime attribute has a full functional dependency on a candidate key
- Every non-trivial functional dependency either begins with a superkey or ends with a prime attribute

Our design can meet the demand.

## show visualization



- **Item**: use its unique name as the primary key. It contains information about the item itself. Many other relations are use item name as the foreign key.

- Containing: use the item stored in it as the primary key.
- shipping: use the unique name of the item on the ship as the primary key.
- Retrieval: use the unique name of item as the primary key. store the information directly connected to the courier.
- Delivery: use the unique name of item as the primary key. store the information directly connected to the courier.
- Import: use the item being imported as the primary key. It stores information generated when the import happened.
- export: use the item being exported as the primary key. It stores information generated when the export happened.
- Seaport\_officer: use its unique name as the primary key.
- Courier: use its unique name as the primary key.
- company\_manager: use its unique name as the primary key. store the information directly connected to the person.
- Sustc\_department\_manager: use its unique name as the primary key. store the information directly connected to the person.

## database user creation and permission granting descriptions

```
create role Courier with password '1';
create role SeaportOfficer with password '2';
create role Company_Manager with password '3';
create role SUSTC_Department_Manager with password '4';
grant select, update on Item,Export,Import to seaportofficer;
grant select on Item,Import,Export,containing,shipping to company_manager;
grant update on containing,Item to company_manager;
grant select on Item,Retrieval,Delivery to courier;
grant insert on Item,Retrieval,Delivery,Export,Import,Containing,Shipping to courier;
grant update on Item,Delivery,Retrieval to courier;
grant select on
Item,Delivery,Retrieval,Export,Import,Containing,Shipping,Company_Manager,Courier,Seapo
rt_Officer,SUSTC_Department_Manager to sustc_department_manager;
```

We use the above sql statements to create 4 users: Courier, SeaportOfficer, Company\_Manager, SUSTC\_Department\_Manager.

According to the description of the project, we let every type of the staffs share the same user.

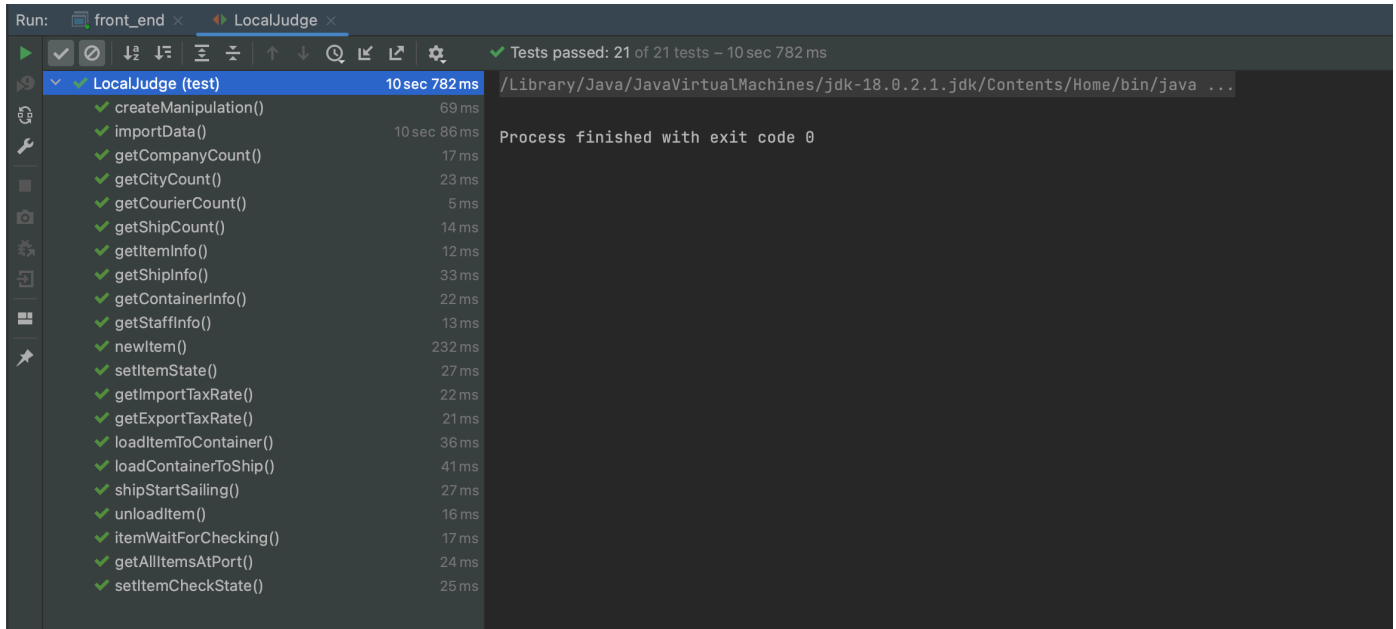
And we granted different permissions to different types of users according to the operations that they are described to be permitted to do before:

- Courier can

## 2 Basic API Specification

We successfully implemented ALL of the required basic APIs in this sections.

And the JUnit test are shown below:



From the above screenshot, it is clear to see that all of the APIs have passed the JUnit test, indicating that the APIs we have implemented meet the requirements of the description.

## 2.1 Table Manipulation

### 2.2-2.5

The thought of the implementation are similar.

- We first check the validity of the input login information
  - If it is valid, then connect to the database and do the required operation and return the result.
  - If is is invalid, then just return the output indicating the invalidity of the login information.

## 3 Advanced APIs and Other Requirements

### frontend

We designed a frontend that can login, logout and do the operations that certain users are permitted to do.

- In our frontend, first you can choose to exit or enter it by inputing a number:
  - If you choose to enter, a message will be shown, asking for the user type you want to choose to .
  - If you choose to exit, the operation will end.

Please choose the begin state:

[1] choose\_user

[2] exit

1

Please choose the authorization type:

[1] Courier

[2] Company Manager

[3] Seaport Manager

[4] SUSTC Department Manager

|

Please choose the begin state:

[1] choose\_user

[2] exit

2

Exit successfully!

Process finished with exit code 0

- After choosing the type of users you want to login, you will be asked to input your name and the password:
- After entering the name and password, for different type of users, you will be asked to choose from the operations that are permitted. The possible operations for different type of users are different.

Please choose the begin state:

[1] choose\_user

[2] exit

1

Please choose the authorization type:

[1] Courier

[2] Company Manager

[3] Seaport Manager

[4] SUSTC Department Manager

1

Please enter the user name:

*Xie Haoren*

Please enter the password:

*3417181758909436515*

Please choose the operation:

[1] newItem

[2] setItemState

|

Please choose the begin state:

[1] choose\_user

[2] exit

1

1  
Please choose the authorization type:

- [1] Courier
- [2] Company Manager
- [3] Seaport Manager
- [4] SUSTC Department Manager

2

Please enter the user name:

Yu Bao

Please enter the password:

2113353997898296736

Please choose the operation:

- [1] getImportTaxRate
- [2] getExportTaxRate
- [3] loadItemToContainer
- [4] loadContainerToShip
- [5] shipStartSailing
- [6] unloadItem
- [7] itemWaitForChecking

3

Please enter the item name:

|

Please choose the begin state:

[1] choose\_user

[2] exit

1

Please choose the authorization type:

[1] Courier

[2] Company Manager

[3] Seaport Officer

[4] SUSTC Department Manager

3

Please enter the user name:

*Yu Jiabao*

Please enter the password:

*6542608543306460801*

Please choose the operation:

[1] getAllItemAtPort

[2] setItemCheckState

|

Please choose the begin state:

[1] choose\_user

[2] exit

1

Please choose the authorization type:

[1] Courier

[2] Company Manager



[3] Seaport Officer  
[4] SUSTC Department Manager

4

Please enter the user name:

*Xia Lichen*

Please enter the password:

*248442857068449968*

Please choose the operation:

[1] getCompanyCount  
[2] getCityCount  
[3] getCourierCount  
[4] getShipCount  
[5] getItemInfo  
[6] getShipCount  
[7] getContainerInfo  
[8] getStaffInfo

2

29

Please choose the next state:

[1] change\_user  
[2] do\_operation  
[3] exit

2

Please choose the operation:

[1] getCompanyCount  
[2] getCityCount  
[3] getCourierCount  
[4] getShipCount  
[5] getItemInfo  
[6] getShipCount  
[7] getContainerInfo  
[8] getStaffInfo

- If the messages you provided are invalid, it will NOT provide the feedback you want, indicating the invalid of the operation:
  - An example is shown below. You can see that there are invalid information, so the return is FALSE.

Please choose the authorization type:

- [1] Courier
- [2] Company Manager
- [3] Seaport Manager
- [4] SUSTC Department Manager

1

Please enter the user name:

Xie Haoren

Please enter the password:

3417181758909436515

Please choose the operation:

- [1] newItem
- [2] setItemState

1

Please enter the item name:

cherry-3a393

Please enter the item class :

cherry

Please enter the price :

123

Please enter the item state :

PickingUp,  
ToExportTransporting,  
ExportChecking,  
ExportCheckFailed,  
PackingToContainer,  
WaitingForShipping,  
Shipping,  
UnpackingFromContainer,  
ImportChecking,  
ImportCheckFailed,

FromImportTransporting,

Delivering,

Finish

*Finish*

Please enter the retrieval city:

*hu*

Please enter the retrieval courier:

*yes*

Please enter the delivery city:

qwe

Please enter the delivery courier:

Please enter the export city:

sd

Please enter the export officer:

Please enter the export tax:

12.4

Please enter the import city:

ok

Please enter the import officer:

Please enter the import tax:

34.2

false

- After finishing an operation, you can choose to exit, do another operation or log into another user:

Please choose the next state:

[1] change\_user

[2] do\_operation

[3] exit

3

Exit successfully!

Process finished with exit code 0

```
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
```

Please choose the begin state:

[1] choose\_user

[2] exit

1

Please choose the authorization type:

[1] Courier

[2] Company Manager

[3] Seaport Officer

[4] SUSTC Department Manager

4

Please enter the user name:

Xia Lichen

Please enter the password:

248442857068449968

Please choose the operation:

[1] getCompanyCount

[2] getCityCount

[3] getCourierCount

[4] getShipCount

[5] getItemInfo

[6] getShipCount

[7] getContainerInfo

[8] getStaffInfo

1

12

Please choose the next state:

[1] change\_user

[2] do\_operation

[3] exit

1

Please choose the authorization type:

[1] Courier

[2] Company Manager

[3] Seaport Officer

[4] SUSTC Department Manager

3

Please enter the user name:

Please choose the begin state:

[1] choose\_user

[2] exit

1

Please choose the authorization type:

[1] Courier

[2] Company Manager

[3] Seaport Officer

[4] SUSTC Department Manager

4

Please enter the user name:

Xia Lichen

Please enter the password:

248442857068449968

Please choose the operation:

[1] getCompanyCount

[2] getCityCount

[3] getCourierCount

[4] getShipCount

[5] getItemInfo

[6] getShipCount

[7] getContainerInfo

[8] getStaffInfo

2

29

Please choose the next state:

[1] change\_user

[2] do\_operation

[3] exit

2

Please choose the operation:

[1] getCompanyCount

[2] getCityCount

[3] getCourierCount

[4] getShipCount

[5] getItemInfo

[6] getShipCount

[7] getContainerInfo

[8] getStaffInfo

## 4 Extra Information

---

### test environment

MacBook Pro:

- Apple M1 pro
- 8-Core CPU
- 14-Core GPU
- 16GB Unified Memory
- 512GB SSD Storage

## 5 Reflection

---

- Before creating tables, it is necessary to fully consider the relationships between different items. When creating the tables, we should consider whether it is convenient for users to query the informations stored in them and update some of them. All of the efforts are done to make it more fast and safe for us to do the operation and get the correct results that is needed.
- The relationship, the constructure of the project and all things about the frame of the project should be designed carefully. This can make further works clearly and more effective. And it will also make codes more maintainable.
- When coding the object-oriented programming model is useful. When some code will be reused for many times, it is better to write them as a separate method rather than repeat them more than once. And it will also help us to reivise these codes.