Database Design

Report Document

彭鹏 201830570231

郭泓树 201830570088

林师言 201830570194

陈秉亮 201830570019

School of Computer Science and Engineering

SCUT

2020.9---2020.12

**Database Course Design Report**

1. **Introduction**

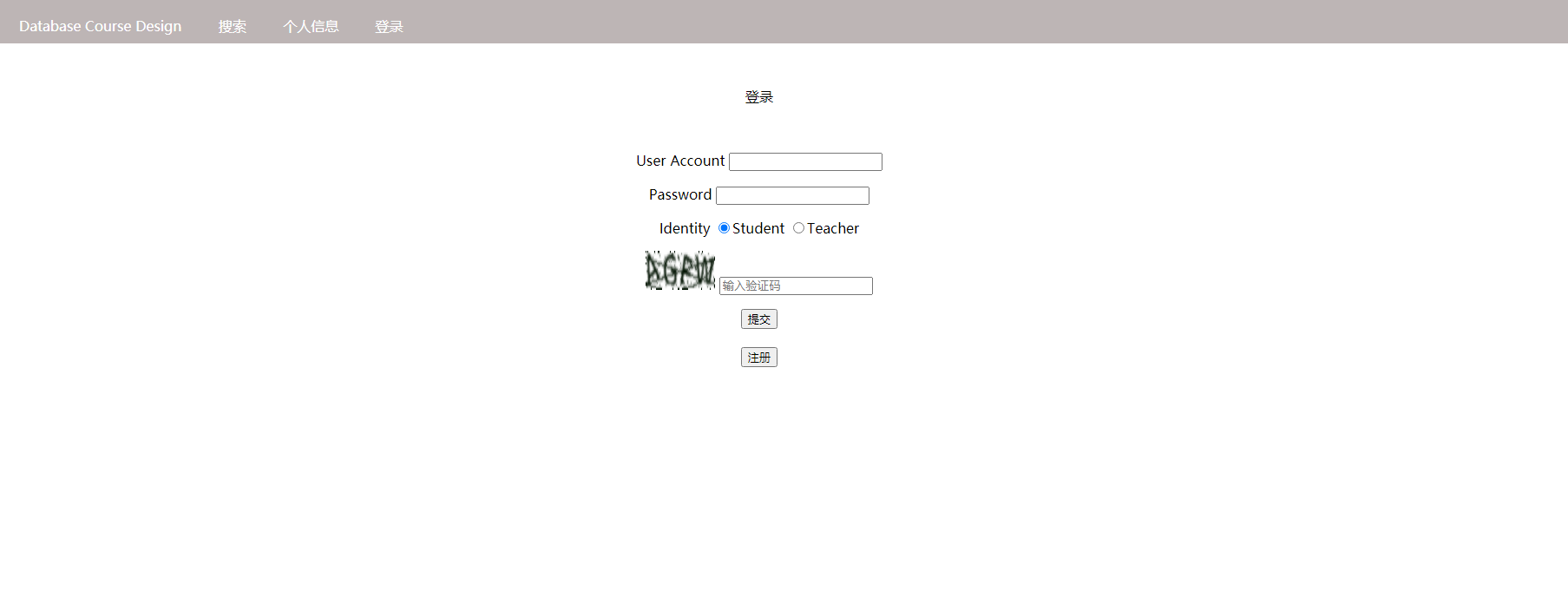
**In this course design, we use Django to build a campus e-commerce website for SCUT. The website is focus on the back-end design. You can visit** [**http://xinshi.cool:8000/**](http://xinshi.cool:8000/) **to try the website. We mainly achieve the login function, the search function, the commodity edits function and the transaction process. We use MySQL as our database and ORM to manage the database in the program.**

1. **User Guide**
2. **To visit our website, you can visit** [**http://xinshi.cool:8000/**](http://xinshi.cool:8000/)**. We have deployed the resources on our server with domain name. Since the domain name is not register for an e-commerce website, this might cannot reach. You can run this Django project on your own computer. The database resources are stable running at the server.**
3. **The start page of the website provides a login interface and the search interface.**



**Users can input search key to searching for items, and can press the “登录” button to login or register.**

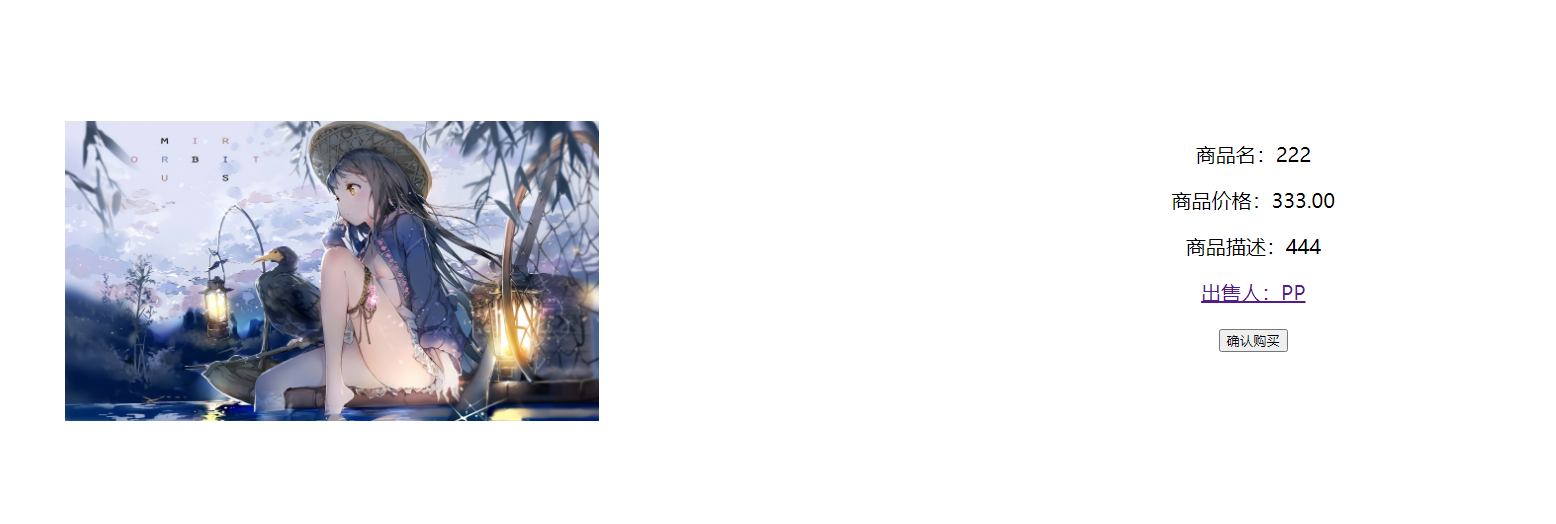
1. **At the login page, the user need provide their user account and the password to login. We also achieve the verification code function to limit the unsafe login action. At the register page, the new user needs to provide some detail messages like user account and the password. The following messages like name and college is optional.**

****

1. **The searching result page shows all the search result from the searching keywords. We also support “space” to split the keywords, means that using “A ‘space’ B” can get items which name includes both A and B.**

****

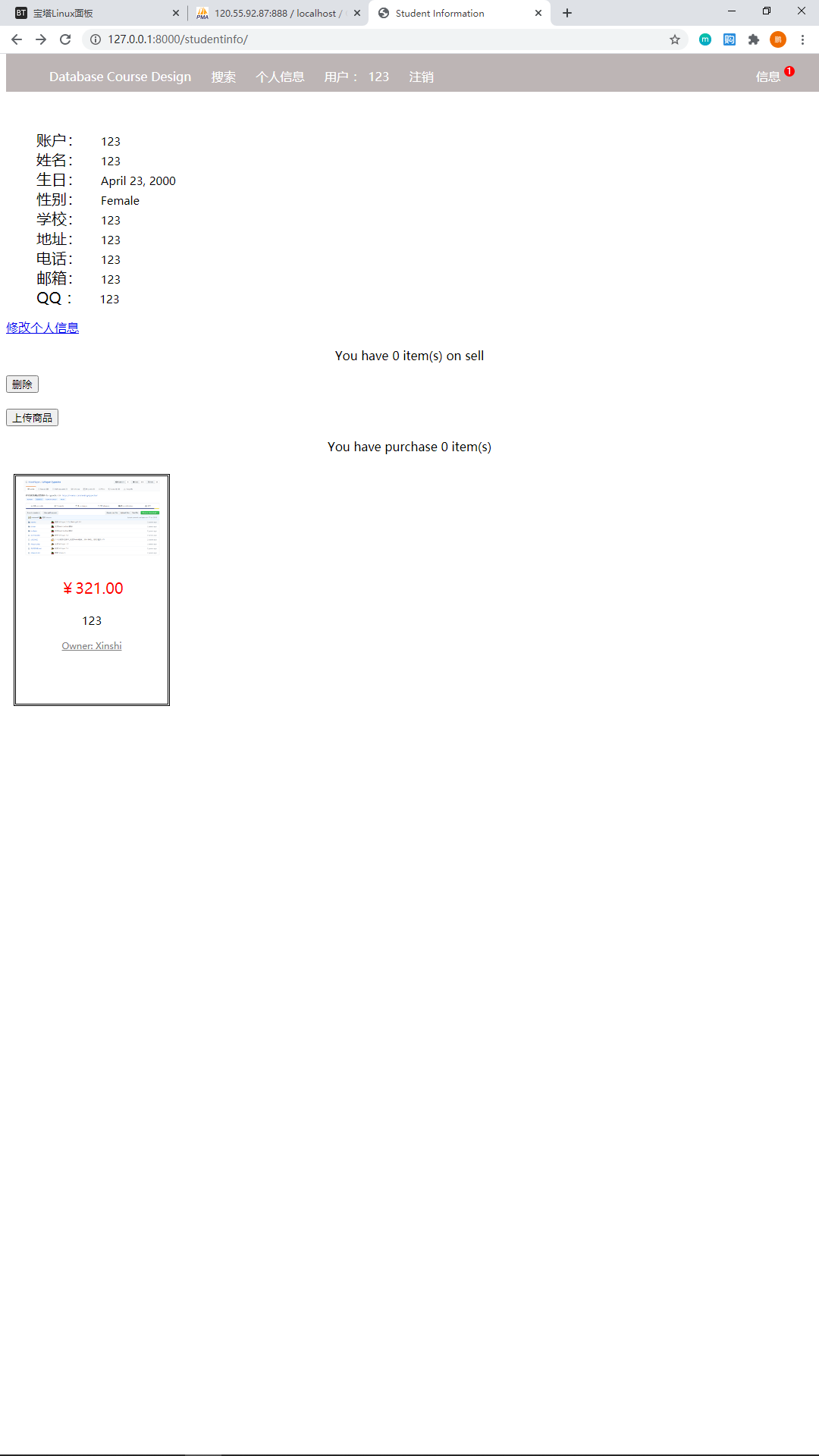
1. **At the items detail page, the user can see larger picture about the item. The name, price, descriptions and owner are listed on the right side. After press the “确认购买” button, the server will handle the request. The seller will receive a message about the purchase request. The buyer needs to wait for seller’s confirmation. Also, the buyer can visit the seller’s personal information page to get the contact methods.**

****

1. **At the personal information page, the user can modify the messages of himself and upload items for sale. The item should contains the name, price, description and the picture.**

****

**Also all the items purchased will be shown at this page.**

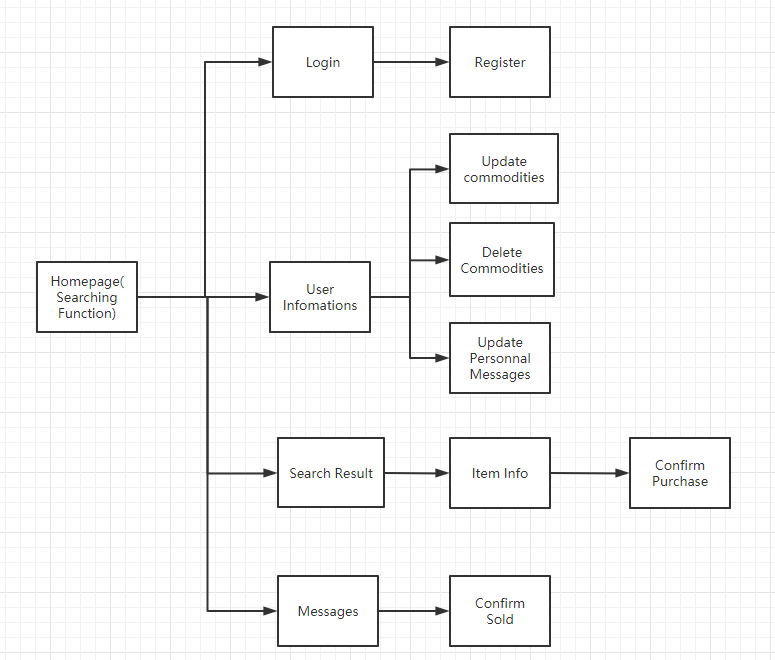
****

1. **The message page shows all the purchase request. When someone confirm purchase the user’s item, it will show a message at this page. The seller can confirm sale and communicate with the buyer via the link to buyer’s personal message page.**

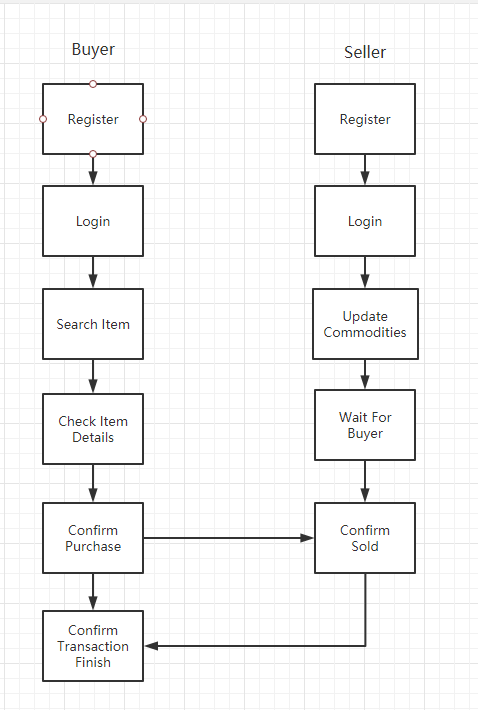
****

1. **Flow Chart**

**Webpage flow chart:**

****

**User flow chart:**

****

1. **Database Settings**

**With the ORM functions, we can create the database tables as a class and use migration function built-in Django. Also, add, delete, modify and search can be easier to achieve via ORM functions and process the database as a normal class of object.**

**Commodity**

****

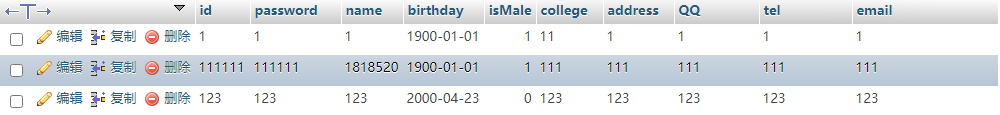
**Teacher(Manager)**

****

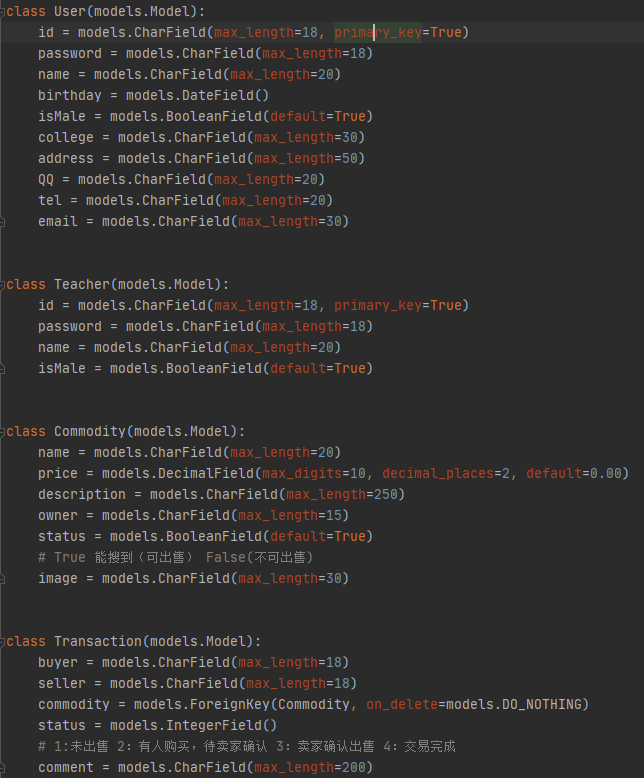
**Transaction**

****

**User**

****

**Program of ORM**

****

**Example of add, delete, modify and search**

**Add: Create a commodity from the input settings.**

****

**Delete: Delete one commodity**

****

**Modify: Use update function to modify the data.**

****

**Search: Search from user table where “id=input account” and “password = input password”**

****

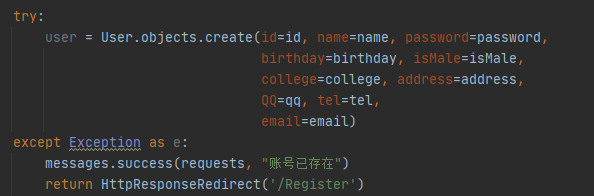
1. **Achievement Details**

**In this course design, we use the Django framework to build our websites back end. Django is an MVC framework based on python that provides an easily manage tools of the website construction. The Django database manager uses ORM to make all the data in the database to be visited as an object and the tables with constraint to be created and modified like a normal class.**

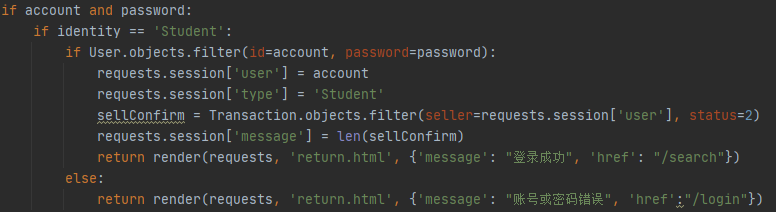
1. **User account managing**

**The user’s account in our design sets the account as primary key. For the register, the controller will check whether the inputs follow the standard and return the corresponding messages. Then we call the create function to add the messages to database. If failed, it means the primary key conflict and return the error warning.**

****

****

**For the login, we get the account and password from post request and check whether it is in our database. If not been found, return login error.**

****

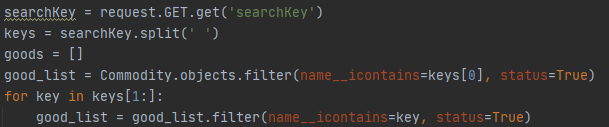
1. **Searching Function**

**The searching function we use the filter function provided by Django to achieve. For this function, it can receive params with modifier. For example, “name\_\_icontains” means filter out all the data which name attribute contains some keywords.**

****

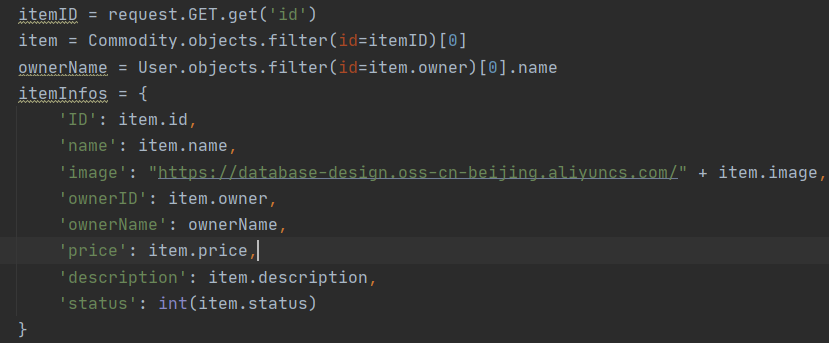
Example

**With this function, the server firstly get inputs from the front end, then split this words at space position. Then the program will recursively filter out the data user want.**

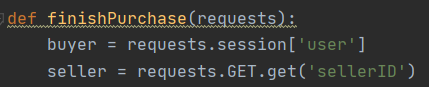
****

1. **Purchase process**

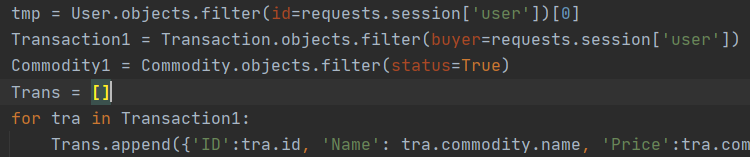
**After a user find things willing to buy, the user will click the itemInfo interface and show detail information of item.**



**User click button “purchase”, then the status of item will be status 2 (item is purchased, seller haven’t confirmed to sell).**

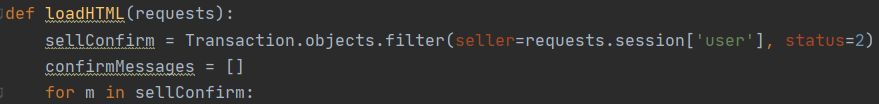


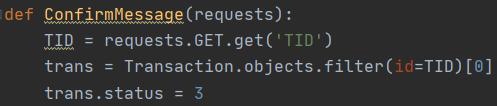
**After purchasing for an item, the buyer users can find their orders in the personal profile. The order interface will show the items purchased by user can provide a function for user to comment for a particular order.**



**When an item in status 2, seller can confirm to sell this item. And the status will change into 3 (seller confirm to sell).**

**In the seller account interface, a message about purchasing issue will be in the message box, the seller can confirm to sell this item by clicking “confirm” button. Function confirmMessage will change the status into 3.**





**After the seller confirm sell the item, the buyer can confirm in the profile interface. Then the status of this transaction will turn to 4 (transaction finished), and the whole process of purchasing is finished.**

1. **Admin platform**