

VALIDATION

SHARK Second-hand Trading Platform

Group 4

舒梓轩 姜浩然 徐源松

Table of Contents

T1: Unit Test.....	3
T1.1: seller_buyer_product Unit Test.....	3
T1.1.1 Test User.signin.....	3
T1.1.2: Test User.login.....	3
T1.1.3: Test User.logout.....	4
T1.1.4: Test Product.add_product.....	4
T1.2: Test administrator.....	5
T1.2.1: Test administrator.login.....	5
T1.3 Test UI.....	5
T1.3.1 Test login_UI:	6
T1.3.2 Test administrator login UI:	6
T1.3.3 Test show products UI:	7
T1.3.4 Test Buyer Cancel Receipt.....	8
T1.3.5 Test upload product:	8
T1.3.6 Test seller deliver UI.....	9
T2 :Integration Test.....	10
T2.1: SystemProcessor+SystemDB+mainBoardUI Integration.....	10
T2.2: SystemProcessor+SystemDB+mainBoardUI+ FuctionsUserConfirm Integration....	11
T3 Function Test.....	13
T3.1 Test "Get the Like List" and "Add Like"	13
T3.2 Test "Get the watching List" and "Watch"	14
T3.3 Test "Get the Selling Inf"	15
T3.4 Test "Get the Buy Inf" and "buy"	16
T3.5 Test "Get the Trade Inf" and "Get the Trade_buyer Inf"	17
T3.6 Test "Chat"	18
T3.7 Test "Invest money"	19
T4: Model Checking.....	20
Check Properties.....	20
P4.1.....	20
P4.2.....	21
P4.3.....	21
P4.4.....	21

T1: Unit Test

T1.1: seller_buyer_product Unit Test

T1.1.1 Test User.signin

```
def test_sign_in(self, username, password, email):
    self.sign_in(username, password, email)
    user_inf = self.users_inf.find({'email':email})
    if len(user_inf) != 0:
        return [user_inf[0][1], user_inf[0][2], user_inf[0][3]]
    else:
        return user_inf
```

	Test Case T1.1.1.1	Test Case T1.1.1.2	Test Case T1.1.1.3
Coverage item	Tcover1.1.1.1	Tcover1.1.1.2	Tcover1.1.1.3
Input	'Zhangxiaoming','12345678','Zhangxiaoming@163.com'	'Lidachui','87654321','Lidachui@163.com'	'Lixioachui','87654321','Lixiaochui@163.com'
State	self.signin_or_login = False	self.signin_or_login = False	self.signin_or_login = True
Expect Output	['Zhangxiaoming','12345678','Zhangxiaoming@163.com']	['Lidachui','87654321','Lidachui@163.com']	[]

Test coverage: 3/3=100%

Test result: 3 passed

T1.1.2: Test User.login

```
def test_login(self, email_add, password):
    return self.login(email_add, password)
```

	Test Case T1.1.2.1	Test Case T1.1.2.2	Test Case T1.1.2.3
Coverage item	Tcover1.1.2.1	Tcover1.1.2.2	Tcover1.1.2.3
Input	'Zhangxiaoming@163.com','12345678'	'Lidachui@1633.com','87654321'	'Lixioachui@163.com','8765'
State	self.signin_or_login = False	self.signin_or_login = False	self.signin_or_login = False
Expect Output	0	1	2

Test coverage: 3/3=100%

Test result: 3 passed

T1.1.3: Test User.logout

```
    return self.login(email_addr,password)

def test_logout(self):
    self.logout()
    if len(self.like_list) != 0:
        self.like_inf = sql.Table(self.id + '_like_table', header=['product_number'])
        for item in self.like_list:
            print('like_inf',self.like_inf.find({'product_number':item}))
    if len(self.watching_history) != 0:
        self.watching_inf = sql.Table(self.id + '_watching_table', header=['product_number'])
        for item in self.watching_history:
            print('watching_inf',self.watching_inf.find({'product_number': item}))
    if len(self.purchase_history) != 0:
        self.purchase_inf = sql.Table(self.id + '_purchase_table', header=['product_number'])
        for item in self.purchase_history:
            print('purchase_inf',self.purchase_inf.find({'product_number': item}))
    if len(self.selling_list) != 0:
        self.selling_inf = sql.Table(self.id + '_selling_inf', header=['product_number'])
        for item in self.selling_list:
            print('selling_inf',self.selling_inf.find({'product_number': item}))
    if len(self.sold_history_list) != 0:
        self.sold_history = sql.Table(self.id + '_sold_history', header=['product_number'])
        for item in self.sold_history_list:
            print('sold_history',self.sold_history.find(item))

class Product():
    def __init__(self):
```

	Test Case T1.1.3.1
Coverage item	Tcover1.1.2.1
Input	-
State	-
Expect Output	like_inf [[product_id_772e949b473dad6efe39733cd53a966c]] watching_inf [[product_id_772e949b473dad6efe39733cd53a966c]] purchase_inf [[product_id_772e949b473dad6efe39733cd53a966c]] selling_inf [[product_id_d4e1c16ef860b6ce7ff6fee27391623]] sold_history [[product_id_f68724543c5d83c6b52471e822cf74b9]]

Test coverage: 1/1=100%

Test result: 1 passed

T1.1.4: Test Product.add_product

```
def test_add_product(self,price, seller_id, product_name, image, description, amount, tag, if_banned=False):
    product_id = self.add_product(price, seller_id, product_name, image, description, amount, tag, if_banned=False)
    print(self.product_inf.find({'product_id': product_id}))
```

	Test Case T1.1.4.1	Test Case T1.1.4.2	Test Case T1.1.4.3
--	--------------------	--------------------	--------------------

Coverage item	Tcover1.1.4.1	Tcover1.1.4.2	Tcover1.1.4.3
Input	'123' ' 6ea37e7114945 7be0acc9a1214e55264 ' ' tree' '\trees.jif' ' good' ' 1' ' tool'	'111' ' 6ea37e711 49457be0acc9a1214e 55264 ' ' picture' ' u=366400839,312589 9153&fm=26&fmt=au to.webp' ' good' ' 1' ' tool'	'10' ' 6ea37e71149 457be0acc9a1214e552 64 ' ' picture' ' u=41189723,29089586 73&fm=26&fmt=auto. webp' ' This is product1' ' 1' 'food'
State	self.if_banned = False	self.if_banned = False	self.if_banned = False
Expect Output	[['product_id_b27c6734cf9f 2bcf57e605cc47d0611a', '123', '6ea37e71149457be0acc9a 1214e55264', 'tree', \trees.jif', 'good.', 'FALSE', 'tool', '1']]	[['product_id_f1417a03 7709d3496114279e95 7b09d6', '111', '6ea37e71149457be0a cc9a1214e55264', 'picture', ' u=366400839,312589 9153&fm=26&fmt=au to.webp', 'good.', ' u=41189723,29089586 73&fm=26&fmt=auto. webp'; ' This is product1', 'FALSE', ' food', '1']]	[['product_id_d4e1c16e f860b6ce7ff6ffee27391 623', '123', '6ea37e71149457be0ac c9a1214e55264', ' u=41189723,29089586 73&fm=26&fmt=auto. webp'; ' This is product1', 'FALSE', ' food', '1']]

Test coverage: 3/3=100%

Test result: 3 passed

T1.2: Test administrator

T1.2.1: Test administrator.login

```
# 0: successful login    1: no email    2: wrong  
password
```

```
def test_login(self, email_address, password):  
    print(self.login(email_address, password))
```

	Test Case T1.2.1.1
Coverage item	Tcover1.2.1.1
Input	'shuzx@shanghaitech.edu.cn' 'szx010105'
State	self.signin_or_login = False
Expect Output	0

Test coverage: 1/1=100%

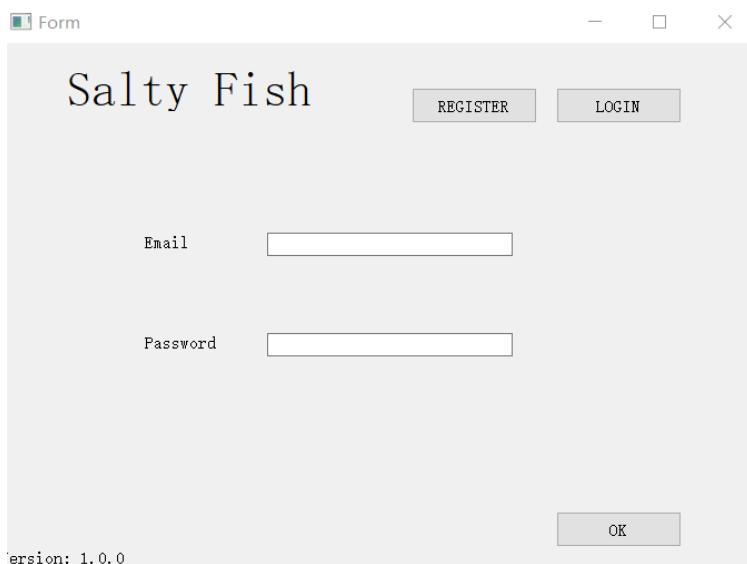
Test result: 1 passed

T1.3 Test UI

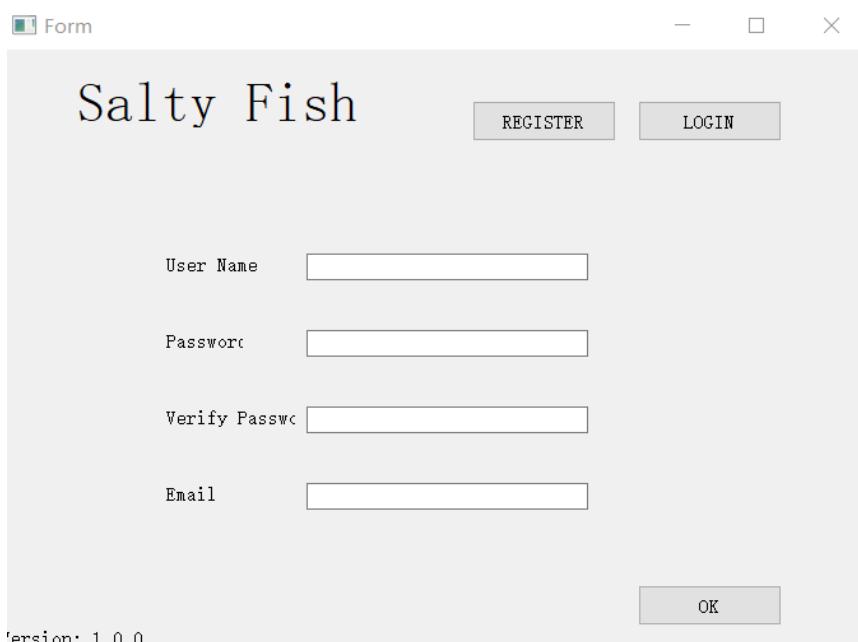
T1.3.1 Test login_UI:

```
def test_login_UI(usr: seller_buyer_product.User, admin: administrator.Admin):
    app = QApplication(sys.argv)
    ui_main = LOGIN.UI_Main(usr, admin)
    ui_main.show()
    app.exec_()
```

Expected output:



After clicking REGISTER:



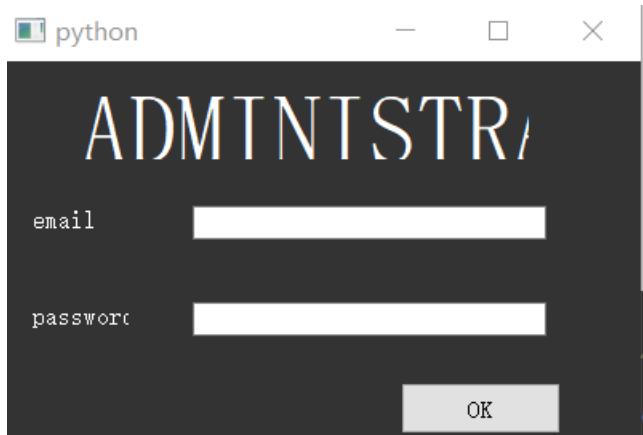
Test coverage: 1/1=100%

Test result: 1 passed

T1.3.2 Test administrator login UI:

```
def test_admin_page(admin: administrator.Admin):
    app = QApplication(sys.argv)
    wa = ADMIN_PAGE.UI_Main(admin)
    wa.show()
    sys.exit(app.exec_())
```

Expected Output:



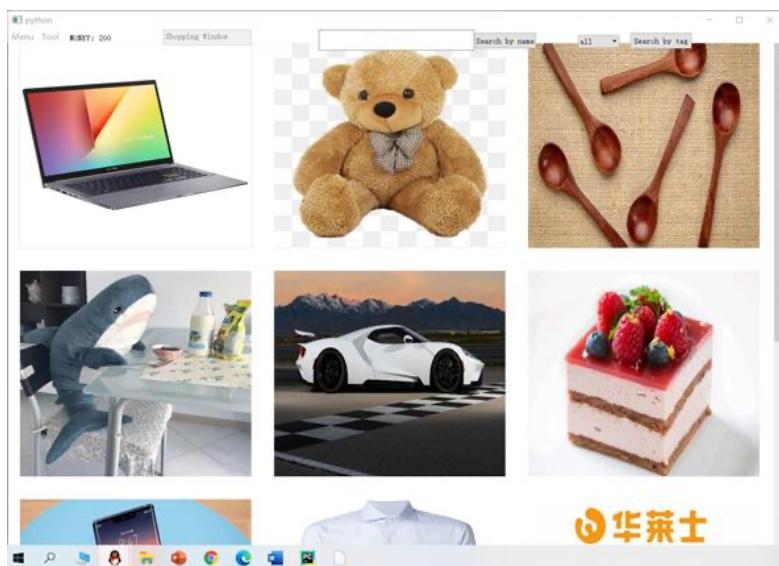
Test coverage: 1/1=100%

Test result: 1 passed

T1.3.3 Test show products UI:

```
def test_show_products(usr):
    app = QApplication(sys.argv)
    wa = USER_WINDOW.UI_Main(usr)
    wa.show()
    sys.exit(app.exec_())
```

Expected Output:



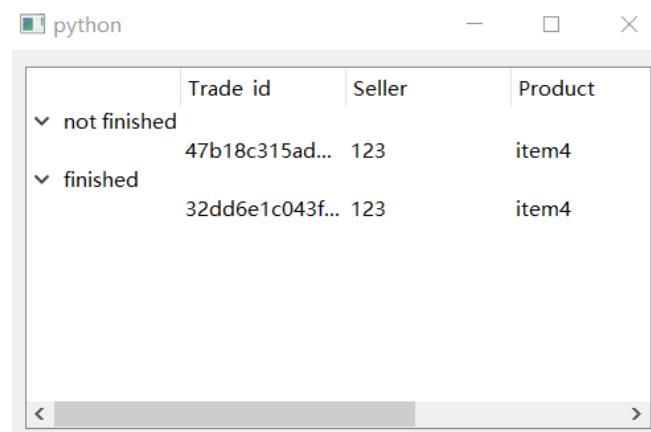
Test coverage: 1/1=100%

Test result: 1 passed

T1.3.4 Test Buyer Cancel Receipt

```
def test_Buyer_Cancel_Receipt():
    user = User()
    user.id = '6ea37e71149457be0acc9a1214e55264'
    app = QApplication(sys.argv)
    wa = Buyer_Cancel_Demo(user)
    wa.show()
    sys.exit(app.exec_())
```

Expect Output:



	Trade id	Seller	Product
not finished	47b18c315ad...	123	item4
finished	32dd6e1c043f...	123	item4

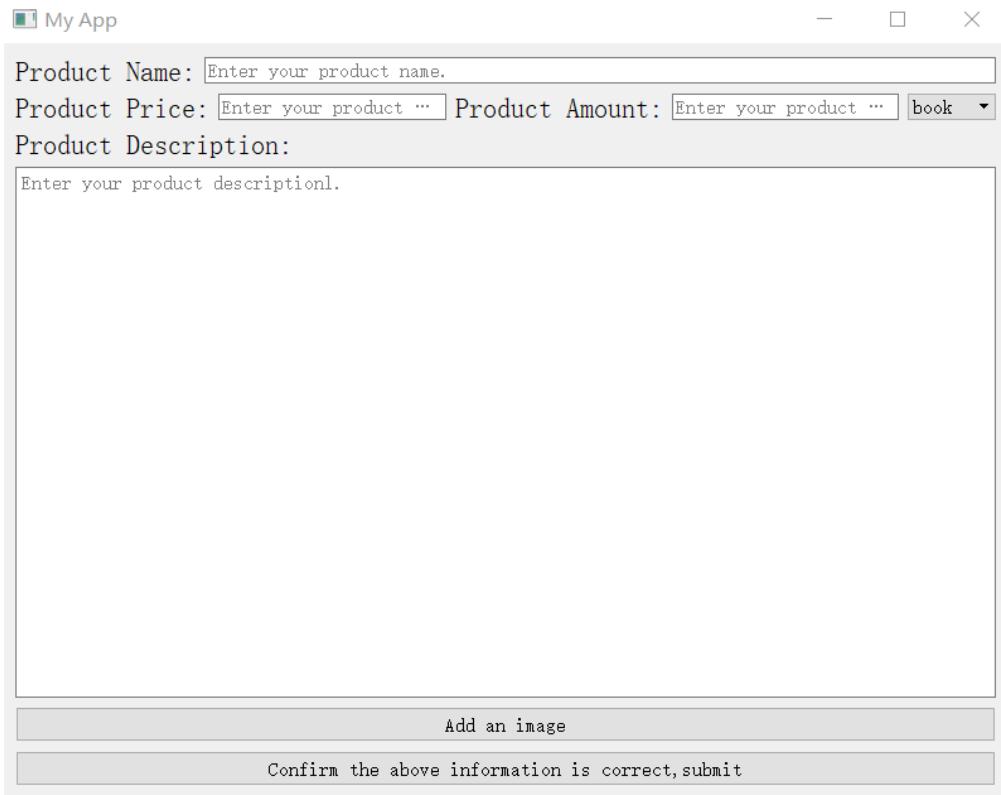
Test coverage: 1/1=100%

Test result: 1 passed

T1.3.5 Test upload product:

```
def test_upload_product:
    m_user = seller_buyer_product.User()
    m_user.username = 'spiderman'
    app = QApplication(sys.argv)
    window = MainWindow(m_user)
    window.show()
    app.exec_()
```

Expect Output:



Test coverage: 1/1=100%

Test result: 1 passed

T1.3.6 Test seller deliver UI

```
def test_seller_deliver():
    user = User()
    user.id = '123'
    user.username = 'jianghr'
    app = QApplication(sys.argv)
    wa = Demo(user)
    wa.show()
    sys.exit(app.exec_())
```

Expect Output:

Seller	Trade id	Buyer	Product	Price	Status
jianghr	47b18c315ad...	6ea37e71149...	item4	\$40	1
	32dd6e1c043f...	6ea37e71149...	item4	\$40	8

Test coverage: 1/1=100%

Test result: 1 passed

T2 : Integration Test

T2.1: SystemProcessor+SystemDB+mainBoardUI Integration

```
def Integration_Test_T2_1(DB, press, product):
    tc.user_email = "jianghr@shanghaiitech.edu.cn"
    tc.user_password = "1234qwer"
    tc.database = DB
    tc.press = press
    if te.press:
        print(tc.get_id(product))
```

Test case:

	Test Case T2.1
Input :	Press all the Product in the Main Window.
State:	Normally Login in
Expected Output:	product_id_90228c2c3205625d1e38b6e6fa6aef5d product_id_8dab908db229e452f2a1addb9c4c436d product_id_10b77035fa97b674633b23a645673777 product_id_910634ff261ab6e750c06800e34c5035 product_id_5f740f9878789b43a14db970584c7e27 product_id_cd211c089ad4a194f3d6410d0bae3870 product_id_09c9d5d28cd06ce4bb6ba57517e9bf2d product_id_8195c85e930e69520630229a9b529458 product_id_229e6294d02a7b1a78521925d6d1fe51 product_id_048b0cb5afa184bbb934ca51647092a3

Actual Output:	product_id_90228c2c3205625d1e38b6e6fa6aef5d product_id_8dab908db229e452f2a1addb9c4c436d product_id_10b77035fa97b674633b23a645673777 product_id_910634ff261ab6e750c06800e34c5035 product_id_5f740f9878789b43a14db970584c7e27 product_id_cd211c089ad4a194f3d6410d0bae3870 product_id_09c9d5d28cd06ce4bb6ba57517e9bf2d product_id_8195c85e930e69520630229a9b529458 product_id_229e6294d02a7b1a78521925d6d1fe51 product_id_048b0cb5afa184bbb934ca51647092a3
----------------	--

- Test coverage: 10/10 = 100%
- Test result: All passed

T2.2: SystemProcessor+SystemDB+mainBoardUI+ FuctionsUserConfirm Integration

```
def Integration_Test_T2_1(DB, press):
    tc.user_email = "jianghr@shanghaitech.edu.cn"
    tc.user_password = "1234qwer"
    tc.database = DB
    tc.press = press
    if te.press:
        print(tc.get_user_email())
```

Test case:

	Test Case T2.2
Input :	Username: jianghr@shanghaitech.edu.cn Password: 1234qwer Press all the function buttons
State:	Normally Login in
Expecte d Output:	jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn

	<u>jianghr@shanghaitech.edu.cn</u>
Actual Output:	A:\Anaconda3\python.exe C:/Users/86136/Downloads/team4-repo-master/Code/main_0.py jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn jianghr@shanghaitech.edu.cn

- **Test coverage:** 9/9 = 100%
- **Test result:** All passed

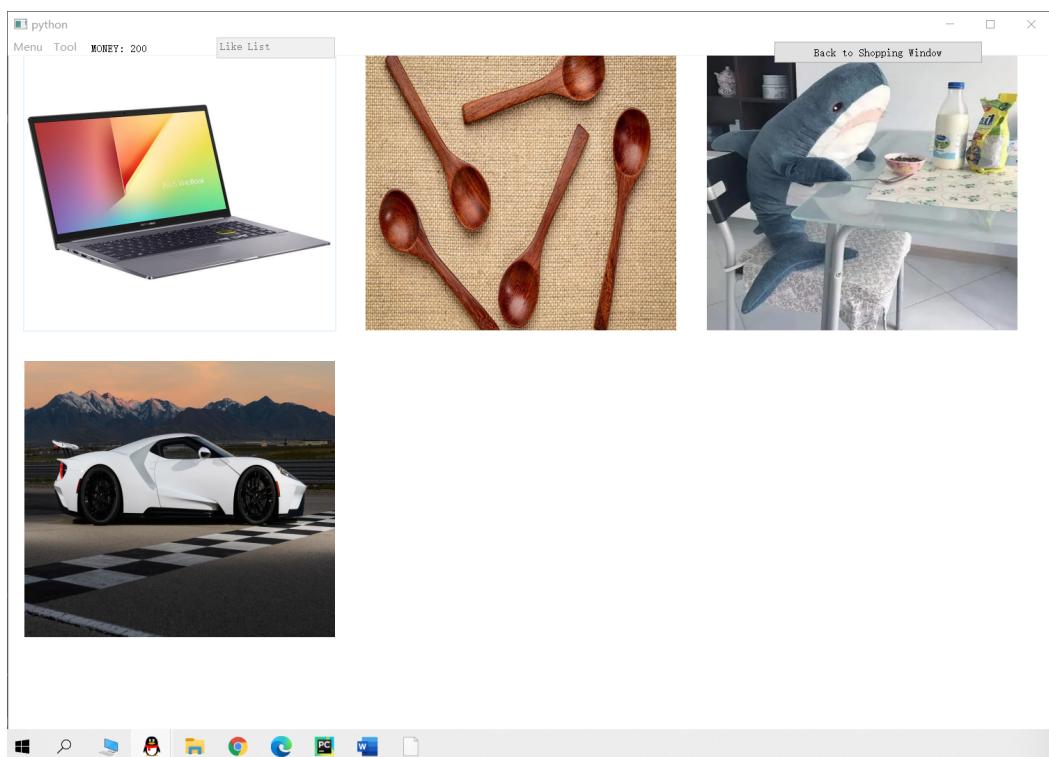
T3 Function Test

T3.1 Test “Get the Like List” and “Add Like”

- Test case

	Test Case T3.1
State	Normally Login in, Like List is empty
Operation	Add like with: product_id_90228c2c3205625d1e38b6e6fa6aef5d product_id_8dab908db229e452f2a1addb9c4c436d product_id_10b77035fa97b674633b23a645673777 product_id_910634ff261ab6e750c06800e34c5035 And check Like List
Expected Behavior	Show all products that have been liked

Output:



```
[['product_id_90228c2c3205625d1e38b6e6fa6aef5d'],
 ['product_id_8dab908db229e452f2a1addb9c4c436d'],
 ['product_id_10b77035fa97b674633b23a645673777'],
 ['product_id_910634ff261ab6e750c06800e34c5035']]
```

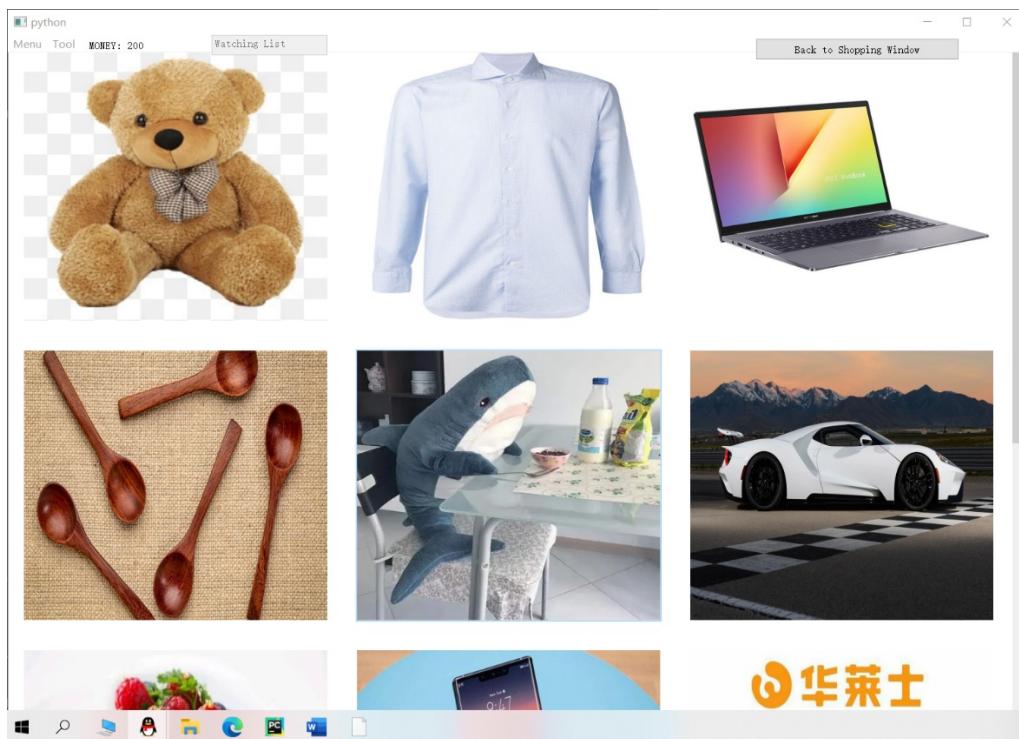
- Test result: All passed

T3.2 Test “Get the watching List” and “Watch”

- Test case

	Test Case T3.2
State	Normally Login in, Watching List is empty
Operation	<p>Watch:</p> <p>product_id_90228c2c3205625d1e38b6e6fa6aef5d product_id_8dab908db229e452f2a1addb9c4c436d product_id_10b77035fa97b674633b23a645673777 product_id_910634ff261ab6e750c06800e34c5035 product_id_5f740f9878789b43a14db970584c7e27 product_id_cd211c089ad4a194f3d6410d0bae3870 product_id_09c9d5d28cd06ce4bb6ba57517e9bf2d product_id_8195c85e930e69520630229a9b529458 product_id_229e6294d02a7b1a78521925d6d1fe51 product_id_048b0cb5afa184bbb934ca51647092a3 And check Watching List</p>
Expected Behavior	Show all products that have been watched

Output:



- `[['product_id_048b0cb5afa184bbb934ca51647092a3'], ['product_id_09c9d5d28cd06ce4bb6ba57517e9bf2d'], ['product_id_90228c2c3205625d1e38b6e6fa6aef5d'], ['product_id_8dab908db229e452f2a1addb9c4c436d'], ['product_id_10b77035fa97b674633b23a645673777'], ['product_id_910634ff261ab6e750c06800e34c5035'], ['product_id_5f740f9878789b43a14db970584c7e27'], ['product_id_cd211c089ad4a194f3d6410d0bae3870'],`

```
['product_id_8195c85e930e69520630229a9b529458'],
['product_id_229e6294d02a7b1a78521925d6d1fe51']]
```

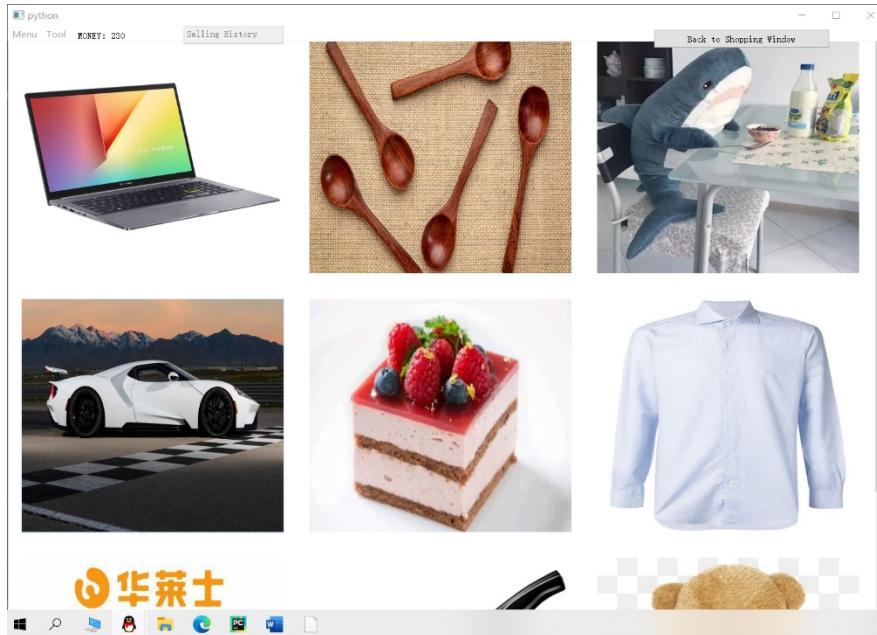
- **Test result: All passed**

T3.3 Test “Get the Selling Inf”

- Test case

	Test Case T3.3
State	Normally Login in
Operation	Check Selling Inf
Expected Behavior	Show all products that is selling

Output:



```
[['product_id_90228c2c3205625d1e38b6e6fa6aef5d'],
['product_id_8dab908db229e452f2a1addb9c4c436d'],
['product_id_10b77035fa97b674633b23a645673777'],
['product_id_910634ff261ab6e750c06800e34c5035'],
['product_id_5f740f9878789b43a14db970584c7e27'],
['product_id_09c9d5d28cd06ce4bb6ba57517e9bf2d'],
['product_id_8195c85e930e69520630229a9b529458'],
['product_id_229e6294d02a7b1a78521925d6d1fe51'],
['product_id_048b0cb5afa184bbb934ca51647092a3']]
```

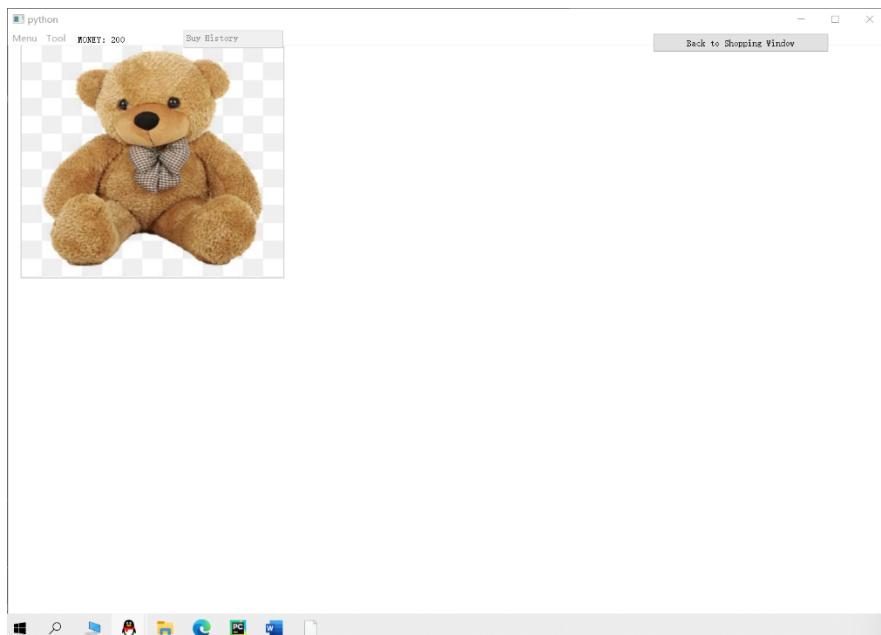
- **Test result: All passed**

T3.4 Test “Get the Buy Inf” and “buy”

- Test case

Test Case T3.4	
State	Normally Login in
Operation	Buy: product_id_048b0cb5afa184bbb934ca51647092a3
Expected Behavior	Show all products that is bought

Output:



[product_id_048b0cb5afa184bbb934ca51647092a3]

- **Test result: All passed**

T3.5 Test “Get the Trade Inf” and “Get the Trade_buyer Inf”

- Test case

	Test Case T3.5
State	Normally Login in
Operation	product_id_048b0cb5afa184bbb934ca51647092a3 been bought and send and received and returned
Expected Behavior	Show all trade that user acts as Seller with status 7

Output:

The image shows two windows of a Python application. Both windows have a title bar labeled "python".

The top window displays a table with columns: Trade id, Buyer, Product, Price, and Status. The data shown is:

	Trade id	Buyer	Product	Price	Status
return request					
receive return	4854292de8e...	jianghr	Teddy Bear	¥30	7
wait for deliver					
delivered					

The bottom window displays a table with columns: Trade id, Seller, Product, Price, and Status. The data shown is:

	Trade id	Seller	Product	Price	Status
not finished	4854292de8e...	d3026a37c42...	Teddy Bear	¥30	7
finished					

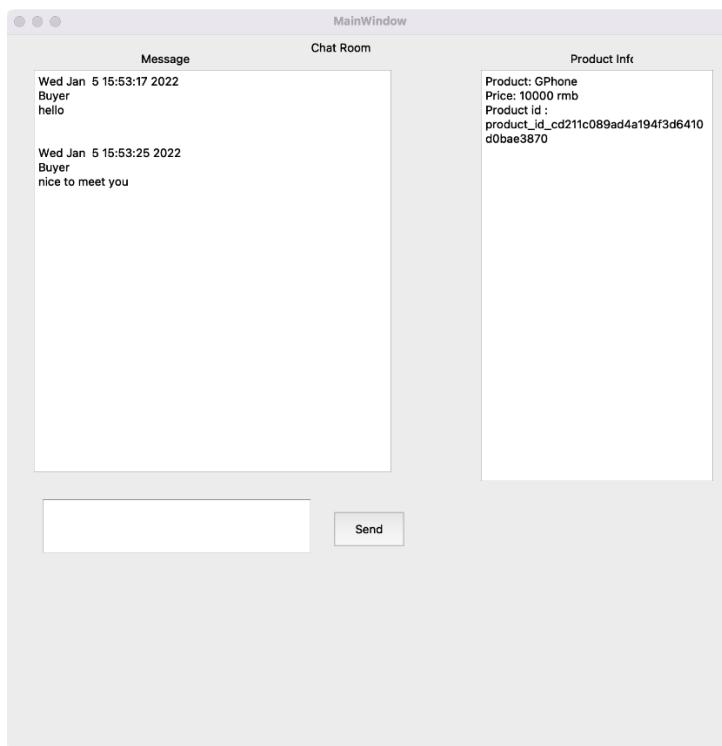
- Test result: All passed

T3.6 Test "Chat"

- Test case

Test Case T3.6	
State	Normally Login in, Chat History is empty
Operation	Send: "hello" "nice to meet you"
Expected Behavior	Show all messages that had been sent

Output:



- **Test result: All passed**

T3.7 Test "Invest money"

	Test Case T3.7
State	Normally Login in. Money = 200
Operation	Invest 200 yuan
Expected Behavior	Show MONEY = 400

Output:

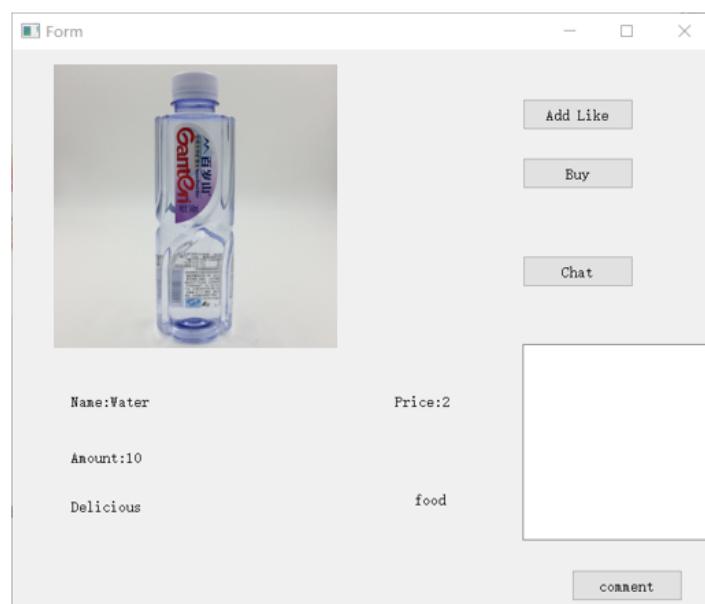


- **Test result: All passed**

T3.8 Test "Upload Product"

	Test Case T3.8
State	Normally Login in
Operation	Upload Product with: Name: Water Price: 2 Tag: food Amount: 10 Description: Delicious Image:src=http__8859372.s21i.faiusr.com_2_ABUIABACGAAg5La43QUo5qyyRjCQEziQEw.jpg&refer=http__8859372.s21i.faiusr
Expected Behavior	Show MONEY = 400

Output:



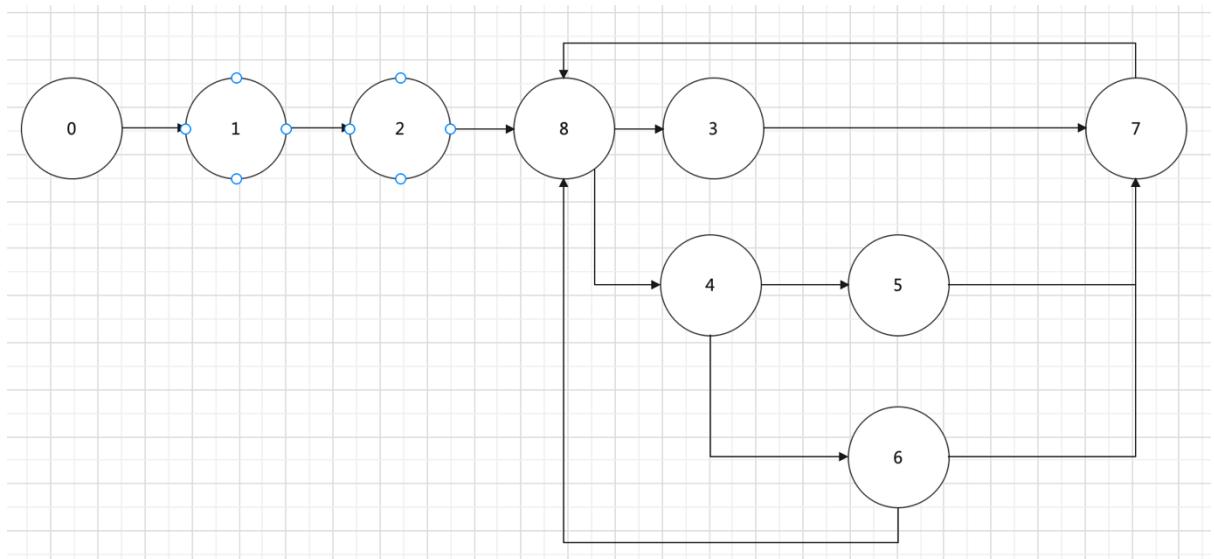
- **Test result: All passed**

T4: Model Checking

Trade

trade_status:

- 0: buyer: buy, buyer--money-->admin
- 1: seller: deliver
- 1.5: product on the way, seller--product-->buyer
- 2: buyer: receive, admin--money-->seller
- 3: buyer: return purchase with no reason(i.e. in 7 days), bank--money->buyer
- 4: buyer: require for return purchase
- 5: seller: accept the requirement, bank--money->buyer
- 5.5: product on the way, buyer--product-->seller
- 6: seller: reject the requirement
- 7: seller: receive the product, seller--money->bank
- 8: trade done!



A trade is created when buyer press ‘buy’ button in the product page and his/her money is enough. The state of trade at this time is 0, money buyer paid will store in bank. After seller deliver the product, the state change to 1. After buyer confirm receive the product the state will change to 2, then seller will get the money, state change to 8.

When return product, check weather trade within 7 days.

Trade within 7 days will directly change its state to 3, after send back money, state will change to 7 and it will change to 8 after seller receive the product.

Trade out of 7 days will be more complicated. The state will be directly change to 4 when buyer want to return the product. If seller reject the require, state will change to 6 then to 8. If seller accept the require, state will change to 5 and then 7 after send back money. Also state will change to 8 after seller receive the return product.

Check Properties

P4.1

Property	A complete trade from buyer buy the product to buyer receive product
----------	--

Description	State change: 0->1->2->8 Money from buyer to bank at state 0, from bank to seller at state 2.
Result	Pass

P4.2

Property	A complete return in 7 days flow path
Description	State change: 8->3->7->8 Money from bank to buyer at state 3, from seller to bank at state 7.
Result	Pass

P4.3

Property	A complete return out of 7 days flow path with seller reject the return requirement.
Description	State change: 8->4->6->8
Result	Pass

P4.4

Property	A complete return out of 7 days flow path with seller accept the return requirement.
Description	State change: 8->4->5->7->8 Money from bank to buyer at state 5, from seller to bank at state 7.
Result	Pass