**Unit Test**

1. **Tools**

Junit

1. **Test cases :**

Class: CUSTOMERSERVICE

**UT-1**

Public Customer Login(Customer customer)

**Description**

This test case tests Customer Login(Customer customer) method to see that user can login with valid and matched username and password and can not login with invalid to unmatched username and password.

**Test cases**

| ID | description | Input | Excepted output |
| --- | --- | --- | --- |
| 1 | the method can enable user to login with valid and matched username and password | username=‘[se5421115511@vr.camt.info](mailto:se5421115511@vr.camt.info)’  password=‘542115511’ | pass |
| 2 | the method can not enable user to login with invalid username | username=‘se542115511camt.info’  password=‘542115511’ | fail |
| 3 | the method can not enable user to login with invalid password | username=‘[se5421115511@vr.camt.info](mailto:se5421115511@vr.camt.info)’  password=‘54’ | fail |
| 4 | the method can not enable user to login with unmatched username and password | username=‘[se5421115511@vr.camt.info](mailto:se5421115511@vr.camt.info)’  password=‘542115512’ | fail |

**UT-2**

Public List<Product> SearchProductByName(String productName)

**Description**

This test case tests List<Product> SearchProductByName(String productName) method to see it can find the product list form the persistent by the existed product name and can not find with not existed product name.

**Test cases**

| ID | description | input | expected result |
| --- | --- | --- | --- |
| 1 | the method can find the products in persistence by the existed product name | productName=‘Iphone 4s’ | pass |
| 2 | the method can not find the products in persistence by the not existed product name | productName=‘AA’ | fail |

**UT-3**

Public void AddToShoppingCart(Product product)

**Description**

This test case tests void AddToShoppingCart(Product product) method to see it can add a product to shopping cart.

**Test cases**

| ID | description | input | expected result |
| --- | --- | --- | --- |
| 1 | the method can add one product to database | product A | pass |

**UT-4**

Public Customer CheckOut(Customer customer)

**Description**

This test case tests Customer CheckOut(Customer customer) method to see it enables user to check out the account.

**Test cases**

| ID | description | expected result |
| --- | --- | --- |
| 1 | this method can logout and redirect to login page | pass |

**UT-5**

Public Customer SaveShoppingCart(Customer customer)

**Description**

This test case tests Customer SaveShoppingCart(Customer customer) method to see it can save the shopping cart to database or not.

**Test cases**

| ID | description | expected output |
| --- | --- | --- |
| 1 | this method can add a shopping cart to user’s account | past |

**UT-6**

Public Customer EditProfile(Customer customer)

**Description**

This test case tests Public Customer EditProfile(Customer customer) method to see it can edit user’s profile picture or not.

**Test cases**

| ID | description | input | expected output |
| --- | --- | --- | --- |
| 1 | this method can edit user’s profile picture. | customer A | pass |

**UT-7**

Public Customer Register(Customer customer)

**Description**

This test case tests Customer Register(Customer customer) method to see it can add a new customer with validate username and password to the system and can not add a new customer to system with invalidate inputs.

**Test cases**

| ID | description | input | expected output |
| --- | --- | --- | --- |
| 1 | this method can add a new customer to system with valid username and password | username=“781832265@qq.com”  password=“chenyuke”  address=“chiangmai, Thailand”  picture=test.jpg | pass |
| 2 | this method can not add a new customer to system with invalid username | username=“781832265qq.com”  password=“chenyuke”  address=“chiangmai, Thailand”  picture=test.jpg | fail |
| 3 | his method can not add a new customer to system with invalid password | username=“781832265@qq.com”  password=“c”  address=“chiangmai, Thailand”  picture=test.jpg | fail |

**UT-8**

Public Customer getCurrentCustomer()

**Description**

This test case tests Customer getCurrentCustomer() method to see it can get the user’s information according to the username.

**Test cases**

| ID | description | expected output |
| --- | --- | --- |
| 1 | this method can get the information of the user that login currently. | pass |

**UT-9**

Public Customer setCurrentCustomer(Customer customer)

**Description**

This test case tests Customer setCurrentCustomer(Customer customer) method to see it can set the customer login currently to be the current customer or not.

**Test cases**

| ID | description | expected output |
| --- | --- | --- |
| 1 | this method can set the customer who login currently to be the current user | pass |

**UT-10**

Public ShoppingCart getSaveShoppingCart(Customer

customer)

**Description**

This test case tests ShoppingCart getSaveShoppingCart(Customer customer) method to see it can get the shopping cart in persistence according to customer login currently or not.

**Test cases**

| ID | description | expected result |
| --- | --- | --- |
| 1 | this method can get the shopping cart saved form the database fort the current customer | pass |

Class: ADMINISTRATORSERVICE

**UT-11**

Public Product EditProductInfo(Product product)

**Description**

This test case tests Product EditProductInfo(Product product) method to see it can edit the product information in database or not.

**Test cases**

| ID | description | input | expected output |
| --- | --- | --- | --- |
| 1 | this method can change the information of the product in database | product test | pass |

**UT-12**

Public List<ShoppingHistory> ScanShoppingHistory()

**Description**

This test case tests List<ShoppingHistory> ScanShoppingHistory() method to see it can get all the shopping histories in the database or not.

**Test cases**

| ID | description | expected result |
| --- | --- | --- |
| 1 | this method can get all the shopping histories inside the database | pass |

**UT-13**

Public Administrator Login(Administrator admin)

**Description**

This test case tests Public Administrator Login(Administrator admin) method to see it can enable admin to login with valid and matched username and password and can not login with invalid inputs.

**Test cases**

| ID | description | input | expected output |
| --- | --- | --- | --- |
| 1 | this method should enable admin login with valid, matched username and password | username=‘[se5421115507@vr.camt.info](mailto:se5421115507@vr.camt.info?subject=)’  password=‘542115507’ | pass |
| 2 | this method should not enable admin to login with invalid username | username=“542115507vr.cant.info”  password=‘542115507’ | fail |
| 3 | this method should not enable admin to login with invalid username | username=‘[se5421115507@vr.camt.info](mailto:se5421115507@vr.camt.info?subject=)’  password=‘5’ | fail |
| 4 | his method should not enable admin to login with invalid username | username=‘[se5421115507@vr.camt.info](mailto:se5421115507@vr.camt.info?subject=)’  password=‘542115510’ | fail |

Class CustomerDAO

**UT-14**

Public Customer getCustomerById(String customerID)

**Description**

This test case tests Customer getCustomerById(String customerID) method to see it can get the customer form the database according the customer’s id or not.

**Test cases**

| ID | description | input | expected output |
| --- | --- | --- | --- |
| 1 | this method can retrieve a customer form database with valid and existed customer id | 1 | pass |
| 2 | this method can retrieve a customer form database with invalid customer id | 4e | fail |
| 3 | this method can retrieve a customer form database with not existed customer id | 123 | fail |

**UT-15**

Public Customer addCustomer(Customer customer)

**Description**

This test case tests Customer addCustomer(Customer customer) method to see it can add a customer to database or not.

**Test cases**

| ID | description | input | expected output |
| --- | --- | --- | --- |
| 1 | this method could add customer to database | customer A | pass |

**UT-16**

Public Customer updateCustomer(Customer customer)

**Description**

This test case tests Customer updateCustomer(Customer customer) method to see it can update the information of current customer.

**Test cases**

| ID | description | input | expected output |
| --- | --- | --- | --- |
| 1 | this method can update the information current customer | customer B | pass |

**UT-17**

Public List<Customer> getCustomers()

**Description**

This test case tests List<Customer> getCustomers() method to see it can get all the customers in the database or not.

**Test cases**

| ID | description | expected output |
| --- | --- | --- |
| 1 | this method can get all the customers in the database | pass |

Class :ShoppingCartDAO

**UT-18**

Public shoppingCart getShoppingCart(Customer customer)

**Description**

Test the getShoppingCart(Customer customer) method to see that it can get a customer’s shopping cart with the parameter, and it cannot get a customer’s shopping cart if input wrong parameter.

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can get a customer’s shopping cart by using the current customer | Current customer | pass |
| 2 | The method can get a customer’s shopping cart by using another the customer | Another customer | fail |

**UT-19**

Public ShoppingCart addProduct(Product product)

**Description**

Test the addProduct(Product product) method to see that it can Put the product that customer chose to the customer’s shopping cart or not.

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can Put the product that customer chose to the customer’s shopping cart. | The product that customer chose | pass |

**UT-20**

Public ShoppingCart DeleteProduct(Product product)

**Description**

Test the DeleteProduct(Product product) method to see that it can Delete the product that inside the current customer shopping cart or not.

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can Delete the product that inside the current customer shopping cart. | the product that has be inside the current customer shopping cart | pass |
| 2 | The method cannot Delete the product that inside the current customer shopping cart. | the product that isn’t inside the current customer shopping cart | fail |

**UT-21**

Public getProducts(ShoppingCart shoppingCart)

**Description**

Test the List<Product> getProducts(ShoppingCart shoppingCart) method to see that it can get the all product that inside the shopping cart or not.

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can get the all product that inside the shopping cart. | The current customer’s currently shopping cart | pass |
| 2 | The method cannot get the all product that inside the shopping cart. | Other customer’s currently shopping cart | fail |
| 3 | The method cannot get the all product that inside the shopping cart. | the current customer’s other shopping cart | fail |

**UT-22**

Public ShoppingCart clearShoppingCart(ShoppingCart shoppingCart)

**Description**

Test the clearShoppingCart(ShoppingCart shoppingCart) method to see that it can delete all products in the shopping cart or not.

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can delete all products in the shopping cart. | The current customer’s currently shopping cart | pass |
| 2 | The method cannot delete all products in the shopping cart. | Other customer’s currently shopping cart | fail |
| 3 | The method cannot delete all products in the shopping cart. | the current customer’s other shopping cart | fail |

Class: PRODUCTDAO

**UT-23**

Public Product getProductByName(String productName)

**Description**

Test the getProductByName(String productName) method to see that it can get a product that have same name with the parameter or not.

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can get a product that have same name with correct parameter | productName=”chi 2” | pass |
| 2 | The method cannot get a product that have same name with the wrong parameter. | productName=”&abc123” | fail |

**UT-24**

Public Product getProductByID(String productID)

**Description**

Test the getProductByID(String productID) method to see that it can get a product that have same id with the parameter or not.

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can get a product that have same id with correct parameter | productID=002 | pass |
| 2 | The method cannot get a product that have same id with the wrong parameter. | productID=”abc123” | fail |
| 3 | The method cannot get a product that have same id with the wrong parameter. | productID=”abcd” | fail |
| 4 | The method cannot get a product that have same id with the wrong parameter. | productID=”$abcd” | fail |

**UT-25**

Public Product addProduct(Product product)

**Description**

Test the addProduct(Product product) method to see that it can add a new product to the e-shopping website system or not.

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can add a new product to the e-shopping website system | The product that not inside the system | pass |
| 2 | The method cannot add a new product to the e-shopping website system | The product that inside the system | fail |

**UT-26**

Public boolean DeleteProduct(Product product)

**Description**

Test the DeleteProduct(Product product) method to see that it can delete a product in the e-shopping website system

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can delete a product in the e-shopping website system | The product that inside the system | pass |
| 2 | The method cannot delete a product in the e-shopping website system | The product that not inside the system | fail |

**UT-27**

Public product updateProduct(Product product)

**Description**

Test the updateProduct(Product product) method to see that it can change a product of e-shopping website system

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can change a product of e-shopping website system | The product that inside the system | pass |
| 2 | The method cannot change a product in the e-shopping website system | The product that not inside the system | fail |

Class: ShoppingHistoryDAO

**UT-28**

Public ShoppingHistory getShoppingHistoryById(String customerID)

**Description**

Test the getShoppingHistoryById(String customerID) method to see that it can get the shopping history belong to special customer that customer id same as the parameter.

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can get the shopping history belong to special customer that customer id same as the parameter. | customerID=002 | pass |
| 2 | The method cannot get the shopping history belong to special customer that customer id same as the parameter. | customerID=”abc123” | fail |
| 3 | The method cannot get the shopping history belong to special customer that customer id same as the parameter. | customerID=”abcd” | fail |
| 4 | The method cannot get the shopping history belong to special customer that customer id same as the parameter. | customerID=”$abcd” | fail |

**UT-29**

Public List<ShoppingHistory> getShoppingHistories()

**Description**

Test the getShoppingHistories() method to see that it can get all of customers shopping histories in the e-shopping website

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can get all of customers shopping histories in the e-shopping website. | - | pass |

**UT-30**

Public ShoppingHistory AddShoppingHistory(ShoppingCart shoppingCart)

**Description**

Test the AddShoppingHistory(ShoppingCart shoppingCart) method to see that it can make a shopping cart that have check out to be a shopping history with the customer name and customer id

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can make a shopping cart that have check out to be a shopping history with the customer name and customer id | The Shopping cart that the customer has check out | pass |
| 2 | The method cannot make a shopping cart that have check out to be a shopping history with the customer name and customer id | The Shopping cart that the customer didn’t check out | fail |

ClassName: Register

**UT-31**

Public Boolean validateUsername(String username)

**Description**

Test the validateUsername(String username) method to see that it can Validate the format of username before user register a new account or not.

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can Validate the format of username before user register a new account by using email format | Username=”54511@gmail.com” | Return true |
| 2 | The method can Validate the format of username before user register a new account by using not email format | Username=”54511g.com” | Return false |

**UT-32**

Public Boolean validatePassword(String password)

**Description**

Test the validatePassword(String password) method to see that it can Validate the format of user password before the user register a new account.

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can Validate the format of user password before the user register a new account with password format is 5-10 characters | password =”12345678” | Return true |
| 2 | The method can Validate the format of user password before the user register a new account with password format less than 5 characters | password=”1234” | Return false |
| 3 | The method can Validate the format of user password before the user register a new account with password format more than 10 characters | password=”1234678901” | Return false |

**UT-33**

Public Boolean validateRe-password(String rePassword)

**Description**

Test the validateRe-password(String rePassword) method to see that it can Validate re-password is same as password or not.

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can Validate repassword is same as password by using same password | password =”12345678” | Return true |
| 2 | The method can Validate repassword is same as password by using different password | password =”123456” | Return false |

Class: Login

**UT-34**

Public Boolean validateUsername(String username)

**Description**

Test the validateUsername(String username) method to see that it can Validate the format of username before check the username match with user password or not.

**Test cases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can Validate the format of username before check the username match with user password by using email format | Username=”54511@gmail.com” | Return true |
| 2 | The method can Validate the format of username before check the username match with user password by using not email format | Username=”54511g.com” | Return false |

**UT-35**

Public Boolean validatePassword(String password)

**Description**

Test the validatePassword(String password) method to see that it can Validate the format of user password check the username match with user password.

**Testcases**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expected Result |
| 1 | The method can Validate the format of user password check the username match with user password with password format is 5-10 characters | password =”12345678” | Return true |
| 2 | The method can Validate the format of user password check the username match with user password with password format less than 5 characters | password=”1234” | Return false |
| 3 | The method can Validate the format of user password check the username match with user password with password format more than 10 characters | password=”1234678901” | Return false |