OSD Product Management System

High-Level Test Plan

Team 5

Abdurrahman Mohammad
Phuc Phan
Ashley Ma
Camille Pina

Testing Methodologies

Unit test

We will write the unit test for every function and we will run all the unit tests after the development of every new feature to make sure the new feature meets the application requirement and does not interfere with the old ones. For example, when we write the function *getCustomer(customerId)* to get a customer from the database based on his/her id, we will write the unit test for this function to make sure that it works correctly. Later on, when we develop another function *addCustomer(customer)* to add a customer into the database, we also need to run the unit test of *getCustomer(customerId)* again, to make sure the new function does not affect the old features.

Integration test

After passing all the unit tests, we will do the integration tests to test the interaction between integrated units. For example, in the add item page, when an administrator types the item name, price, quantity and description, then clicks on the submit button, we will test to make sure the item information is sent by the browser to the webserver and saved into the database.

Test Metrics

We will measure the average render time of all the web pages.
Response time of the web application
The accuracy of information from database
Accurate calculation on the order and shipping options

Success criteria

The website meets the requirements and the average render time of all the web pages are less than 1 second.

The website functionality and interactivity:

- Users can browse and check featured products from the home page.
- Users can search products through searching bar with matching results
- Users can register account
- User can login to the customer system, review and update information
- User can purchase items
- User can choose different shipping options
- Administrator can modify inventory and update to the whole system in time

The backend handling requests

- The database can provide correct information
- The database can provide different information to the customer and administrator

- The backend can make effective calculation on shipping opinions
- The backend can cooperate with front end with timely and correct information

Functional testing

Visitors/ Customers/ Administrators view company information: business hours, locations, etc.

Goal: all users can see the company information, such as company name on the top and locations, phone number, business hours on the bottom of the home page.
Test: check if the information (company name, phone number, business hours and locations) displayed correctly on both Chrome and FireFox browsers.

Visitors browse items with/without an account

Goal: all users can browse items
Test: check if the item information (item names, prices, descriptions) displayed correctly on both Chrome and Firefox browsers.

Visitors search products with/without an account

Goal: all users can search for an item based on its category or name.
Test: check if the list of search items displayed correctly, such as when the search result is empty, a not found message should be displayed, or when the list is too long, it should display the pagination.

Visitors create account

Goal: visitors can create account
Test: make sure the username is not duplicated, the email and phone number is valid and the password is complicated enough.

Visitors sign into account

Goal: visitors can login into an account
Test: make sure only users who type username and password correctly can login.

Customers/Administrators change account information

Goal: Customers/Administrators can change account information, such as phone number, address, email address, and password.
Test: make sure the new information is saved correctly into the database.

Customers/Administrators log out

Goal: Customers/Administrators can log out by clicking on the logout button
Test: make sure after users log out, then refresh the page, they are still in logout status.

Administrators search for items

Goal: administrators can search for items based on their category and names

Test: check if the list of search items displayed correctly, such as when the search result is empty, a not found message should be displayed, or when the list is too long, it should display the pagination.

Administrators view an item information

Goal: administrators can view an item information, such as name, remaining quantity, price, and description.
Test: Check if these information above is displayed correctly on both Chrome and FireFox browsers

Administrators update items

Goal: administrators can update an item information, such as name, remaining quantity, price, and description.
Test: make sure the new information is saved correctly into the database.

Administrators remove items

Goal: administrators can remove an item
Test: make sure the removed item is no longer in the database

Administrators create item categories

Goal: Administrators can create item categories
Test: make sure the new item category is saved in the database.

Administrators remove item categories

Goal: Administrators can remove item categories
Test: make sure the removed item category is no longer in the database

Administrators manage customer accounts

Goal: Administrators can manage customer accounts, such as view and delete customer accounts
Test: Make sure the customer information is displayed correctly, such as usernames, phone numbers, email addresses, and addresses. And make sure the account is no longer in the database after deleting.

Customers search for items

Goal: Customers can search for items based on their category and names
Test: check if the list of search items displayed correctly, such as when the search result is empty, a not found message should be displayed, or when the list is too long, it should display the pagination

Customers view item information

Goal: Customers can view an item information, such as name, remaining quantity, price, and description.

Test: Check if these information above is displayed correctly on both Chrome and FireFox browsers

Customers save items to carts

Goal: customers can add an item into their shopping carts.
Test: make sure the adding items are still in the shopping carts after refreshing the pages, and are no longer there after checking out.

Customers remove items from carts

Goal: customers can remove an item from their shopping cart.
Test: make sure the removed items are no longer in the database.

Customers change the quantity of items in the cart

Goal: customers can change the quality of items in the cart
Test: make sure the changed quantity is saved to the database.

Customers select payment methods

Goal: Customers can select payment methods
Test: make sure the selected payment methods are saved in the database.

Customers select shipping methods

Goal: Customers can select shipping methods
Test: make sure the selected shipping methods are saved in the database.

Customers track current orders

Goal: Customers can keep track of the current orders
Test: make sure the shipping information is displayed correctly

Customers update delivery address

Goal: customers can update delivery address
Test: make sure the updated delivery address is saved into the database.

Customers cancel orders

Goal: customers can cancel orders
Test: make sure the order statuses are changed to “cancel” in the database.

Customers view previous orders

Goal: customers can view their previous orders
Test: make sure the orders are displayed correctly on both Chrome and Firefox browsers.