**Software Requirements Specification**

**for**

**TGDD**

**Version 1.0 approved**

**Prepared by Group 2**

**Group 2, HCM22\_CPL\_JAVA\_02- FRESHER ACADEMY**

**June 11, 2022**

**Table of Contents**

[Introduction](#_1fob9te) **4**

[Purpose](#_zesebu5bqcqk) 4

[Project Scope and Product Features](#_tyjcwt) 4

[Overall Description](#_1t3h5sf) **4**

[Product Perspective](#_4d34og8) 4

[User Classes and Characteristics](#_17dp8vu) 5

[Operating Environment](#_3rdcrjn) 5

[Design and Implementation Constraints](#_lnxbz9) 5

[Assumptions and Dependencies](#_35nkun2) 5

[Use Case](#_1ksv4uv) **6**

[Use Cases Diagram](#_htwy40sfomsm) 6

[Login](#_j0iql4gkzjnz) 6

[Logout](#_62bcn6eebdzl) 7

[View product](#_e3h1gx4vimls) 7

[View product detail](#_6q3s4ou5obnk) 8

[Add new product](#_510nkw4s1c8g) 8

[Update product](#_n4svf1di3zf9) 9

[Delete product](#_n21oloxgrhs6) 10

[View category](#_bu5b5bkw28h) 11

[Add category](#_41d1pd7j2a3n) 11

[Update category](#_e46zmre5jeqr) 12

[Delete category](#_q5gmxek50gt9) 12

[View order](#_pfl1rruqz82r) 13

[Update order](#_e8bxr5ll7tp8) 13

[Delete order](#_e9q5zhvp1p5e) 14

[View cart](#_ks5tx6pagtjy) 15

[Add product to cart](#_btqy8rqlhds6) 15

[Update product in cart](#_2ftw0m5kjroc) 16

[Delete product from cart](#_u3hwekrld1m5) 17

[Search product](#_gultqi8zh19g) 17

[Check out](#_jhipzse7mvm) 18

[**View order history**](#_ve76amvsrx6e) **19**

[System Features](#_u2smg2nzai2j) **20**

[Login](#_44sinio) 20

[Description](#_pp5ujcxf28f8) 20

[Functional Requirements](#_z337ya) 20

[Login.Show: Show a form for filling the login’s information](https://docs.google.com/document/d/1bR-kPE6Ud6Ku3lvFZXUOJDokYcixwKcpf7tAo9nxJdQ/edit#heading=h.dxe201wawpaw) [20](https://docs.google.com/document/d/1bR-kPE6Ud6Ku3lvFZXUOJDokYcixwKcpf7tAo9nxJdQ/edit#heading=h.dxe201wawpaw)

[View Order History](#_3j2qqm3) 20

[**Search**](#_4dt6no7l4ygz) **21**

[Add product to cart](#_1y810tw) 21

[Description](#_2jxsxqh) 21

[Functional Requirements](#_kklfvw7cpug7) 21

[Add product](#_4i7ojhp) 21

[View product](#_ket0ujehhfnk) 22

[Check out](#_8vpovxk4enza) 22

[Data Requirements](#_5i4cmcmnw0im) **23**

[Logical Data Model](#_1ci93xb) 23

[Class Diagram](#_q4g1esiq0y4s) 24

[External Interface Requirements](#_1pxezwc) **24**

[Communications Interfaces](#_49x2ik5) 24

[Screen Flow](#_8qo0vncn5rcg) 25

[Non-functional Requirements](#_g4deok68dpj) **25**

[Security Requirements](#_ihv636) 25

[Performance Requirements](#_1hmsyys) 25

[Assumptions and Dependencies](#_41mghml) 25

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| TGDD | 27/6/2022 | change usecase diagram format | 1.2 draft 1 |

# **Introduction**

## **Purpose**

This SRS describes the functional and non-functional requirements for software release 1.0 of the TGDD. This document is intended to be used by the members of the project team who will implement and verify the correct functioning of the system.

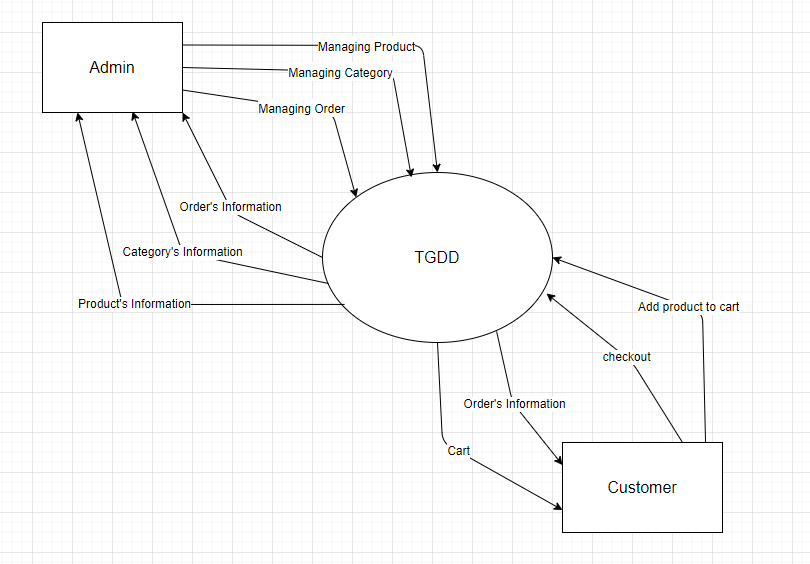
## **Project Scope and Product Features**

TGDD is a web application that provides online shopping for electronic devices. By shopping online, customers can reduce travel time and product selection time; With a simple, fast and safe checkout procedure, customers can shop at this website with peace of mind

# **Overall Description**

## **Product Perspective**

The TGDD is a web application system that replaces manual shopping processes. This system provides customers with the ability to buy electronic devices remotely, bypassing manual steps.



**Figure 1. Context diagram for the TGDD.**

## **User Classes and Characteristics**

| Customer | The customer is someone who wants to buy the electronic devices remotely. When the new system is launched, it is estimated that there are about 1000 potential customers, of which a large number of customers are expected to use TGDD regularly. For customers who do not have time to perform routine procedures, customers who need to shop manually will be potential customers. |
| --- | --- |
| Admin | Admin is the person who manages the TGDD. Admin will manage TGDD including the product (category) and customer’s order. Admin has the right to view customer’s order information, edit and delete user order from the system; Admin is the person who creates the product (category) and gives access to the TGDD for the employee; is responsible for monitoring and resolving invalid access to ensure system safety. |

## **Operating Environment**

OE-1: The TGDD shall operate correctly with the following web browsers: Microsoft Edge; Firefox (all versions); Google Chrome (all versions)

## **Design and Implementation Constraints**

CO-1: The system’s design, code, and maintenance documentation shall conform to the *Process Impact Intranet Development Standard, Version 1.3*.

CO-2: The system shall use the current corporate standard Oracle database engine.

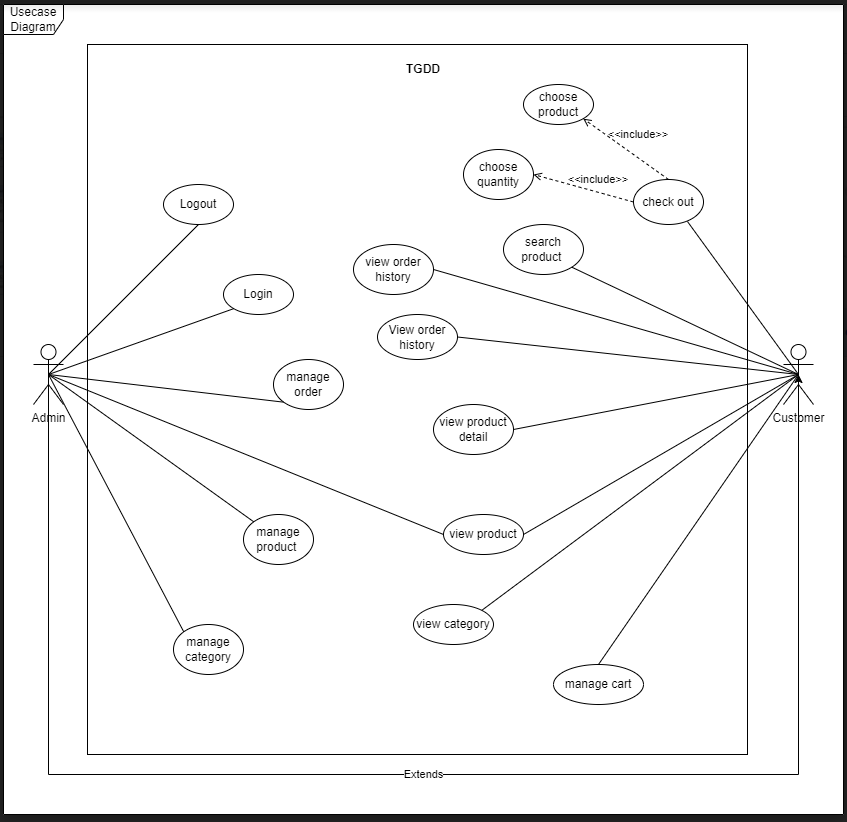
CO-3: All HTML code shall conform to the HTML 5.0 standard.

## **Assumptions and Dependencies**

AS-1: The TGDD is open 24/7 to ensure customer service and provide the best support

# **Use Case**

## **Use Cases Diagram**

****

### **Login**

| ID and Name: | **UC-1 Login** | | |
| --- | --- | --- | --- |
| Created By: | DungNT | Date Created: | 11/6/2022 |
| Primary Actor: | Admin | | |
| Description: | Admin login into the system by their username and password or by phone number and email | | |
| Trigger: | When admin click on Login button | | |
| Preconditions: | PRE-1. Have an account | | |
| Postconditions: | POST-1. System move user to admin page | | |
| Normal Flow: | **1.0 Login**  1. Admin click on the login button  2. System show a form for admin input the username and password  3. Admin input the information  4. System do the authentication  5. System move admin to admin page if they pass the authentication and return to login page if they fail | | |
| Alternative Flows: | **none** | | |
| Exceptions: | **1.0.E1 Username or password is wrong**  System return to login page with a warning about wrong username or password | | |
| Other Information: | None | | |

### **Logout**

| ID and Name: | **UC-2 Logout** | | |
| --- | --- | --- | --- |
| Created By: | DungNT | Date Created: | 11/6/2022 |
| Primary Actor: | Admin | | |
| Description: | Admin logout from the system | | |
| Trigger: | When admin click on Logout button | | |
| Preconditions: | PRE-1. Already login | | |
| Postconditions: | POST-1. System move to login page | | |
| Normal Flow: | **1.0 Logout**  1. Admin click on the logout button  2. System move admin to the login page | | |
| Alternative Flows: | **none** | | |
| Exceptions: | **none** | | |
| Other Information: | None | | |

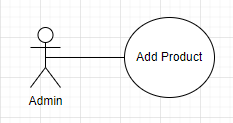
### **View product**

| ID and Name: | **UC-3 Customer View List Product** | | |
| --- | --- | --- | --- |
| Created By: | QuangNH | Date Created: | 11/6/2022 |
| Primary Actor | Customer | Secondary actor | Admin |
| Description: | Customer View List Product | | |
| Trigger: | The user accesses the Thegioididong website | | |
| Preconditions: | The user accesses the Thegioididong website | | |
| Postconditions: | System move list product page | | |
| Normal Flow: | **1. The customer accesses the home page.**  **2. The system redirects to list all products.**  **3. The system allows the user to perform other activities.** | | |
| Alternative Flows: | **none** | | |
| Exceptions: | **none** | | |
| Other Information: | None | | |

### **View product detail**

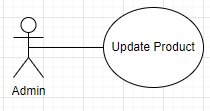
| ID and Name: | **UC-4 Customer View Detail Product** | | |
| --- | --- | --- | --- |
| Created By: | QuangNH | Date Created: | 11/6/2022 |
| Primary Actor | Customer | Secondary actor | Admin |
| Description: | Customer View Detail Product | | |
| Trigger: | The user accesses the Thegioididong website | | |
| Preconditions: | The customer selects a product to view detail. | | |
| Postconditions: | System shows product detail. | | |
| Normal Flow: | **1. The customer accesses the home page.**  **2. The system redirects to list all products.**  **3. The customer selects the product.**  **4. The system redirects to the product detail page.**  **5. The system allows the user to perform other activities.** | | |
| Alternative Flows: | **none** | | |
| Exceptions: | **none** | | |
| Other Information: | None | | |

### **Add new product**



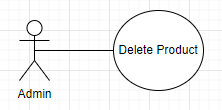
| ID and Name: | **UC-5 Add new product** | | |
| --- | --- | --- | --- |
| Created By: | ToanDD | Date Created: | 11/06/2022 |
| Actor: | ADMIN | | |
| Description: | ADMIN adds a new product | | |
| Trigger: | When ADMIN clicks on “Add product” button | | |
| Preconditions: | PRE-1. Already sign up and have an account role as (ADMIN) | | |
| Postconditions: | POST-1. System move ADMIN to add product form page | | |
| Normal Flow: | **Add a new product**  1. ADMIN clicks on the “Add product” button  2. System shows a form for ADMIN input the information of product  2.1 System shows a form with:  - Field: product name, type: text, length maximum 50 characters  - Field: product code, type: text, length maximum 15 characters  - Field: product price, type: number  - Field: product quantity, type: number  - Field: product category, type: list, text  - Field: product manufacturer, type: list, text  - Table field: product specification  - Field: product description, type: text  - Field: product image, type: file  3. ADMIN fills the form  4. System does the verification to check the form  5. System moves ADMIN to the main page if they pass the verification and return to the main page to do the next action. If they fail the verification, the system shows the error messages to show why they are having a failure. | | |
| Alternative Flows: | none | | |
| Exceptions: | **2.1.E1 Some field is null**  System shows a mess to warn which field doesn’t have any characters.  **2.1.E2 Product name is already existed**  System shows a mess to warning the product name is duplicate  **2.1.E3 Product image must be .png .pneg**  System shows a mess at product image field to warning this file is not acceptance | | |
| Other Information: | None | | |

### **Update product**



| ID and Name: | **UC-6 Update product** | | |
| --- | --- | --- | --- |
| Created By: | ToanDD | Date Created: | 11/06/2022 |
| Primary Actor: | ADMIN | | |
| Description: | ADMIN update a product | | |
| Trigger: | When ADMIN clicks on “Update product” button | | |
| Preconditions: | PRE-1. Already sign up and have an account role as (ADMIN)  PRE-2. Product is already existed | | |
| Postconditions: | POST-1. System move ADMIN to update product form page | | |
| Normal Flow: | **Update a product**  1. ADMIN clicks on the “Update product” button  2. System shows a form for ADMIN input the information of product  2.1 System shows a form with:  - Field: product name, type: text, length maximum 50 characters  - Field: product code, type: text, length maximum 15 characters  - Field: product price, type: number  - Field: product quantity, type: number  - Field: product category, type: list, text  - Field: product manufacturer, type: list, text  - Table field: product specification  - Field: product description, type: text  - Field: product image, type: file  3. ADMIN fills the form  4. System does the verification to check the form  5. System moves ADMIN to the main page if they pass the verification and return to the main page to do the next action. If they fail the verification, the system shows the error messages to show why they are having a failure. | | |
| Alternative Flows: | **none** | | |
| Exceptions: | **2.1.E1 Some field is null**  System shows a mess to warn which field doesn’t have any characters.  **2.1.E2 Product name is already existed**  System shows a mess to warning the product name is duplicate  **2.1.E3 Product image must be .png .pneg**  System shows a mess at product image field to warning this file is not acceptance | | |
| Other Information: | None | | |

### **Delete product**



| ID and Name: | **UC-7 Delete product** | | |
| --- | --- | --- | --- |
| Created By: | ToanDD | Date Created: | 11/06/2022 |
| Primary Actor: | ADMIN | | |
| Description: | ADMIN delete a product | | |
| Trigger: | When ADMIN clicks on “Delete product” button | | |
| Preconditions: | PRE-1. Already sign up and have an account role as (ADMIN)  PRE-2. Product is already existed | | |
| Postconditions: | none | | |
| Normal Flow: | **Delete product**  1. ADMIN clicks on the “Delete product” button  2. System shows a mess for ADMIN to confirm selection  3. System shows a mess to confirm the action is success | | |
| Alternative Flows: | **none** | | |
| Exceptions: | **2.E1 This product can’t be deleted**  System shows a mess that this action (del) is denied | | |
| Other Information: | None | | |

### **View category**

| ID and Name: | **UC-8 View Category** | | |
| --- | --- | --- | --- |
| Created By: | HuongTTD | Date Created: | 11/6/2022 |
| Description: | Users view categories | | |
| Primary Actor: | Customer and Admin | | |
| Trigger: | Customers hover on the Category menu or Admin access Manage Category page. | | |
| Preconditions: | Users logged in the system. | | |
| Postconditions: | List of categories is show | | |
| Normal Flow: | 1. The customer accesses the home page.  2. The customer hover on the Category menu on the Navigation bar.  4. The customer is able to see all visible categories in the system. | | |
| Alternative Flows: | 1. Admin access Manage Category page. 2. Admin is able to see all categories in the system. | | |
| Exceptions: | None | | |
| Other Information: | None | | |

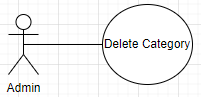
### **Add category**

| ID and Name: | **UC-9 Add Category** | | |
| --- | --- | --- | --- |
| Created By: | HuongTTD | Date Created: | 11/6/2022 |
| Primary Acto:r | Admin |  |  |
| Description: | Admin add new categories | | |
| Trigger: | Admin access Manage Category page. | | |
| Preconditions: | Users logged in the system as Admin. | | |
| Postconditions: | User is returned to the Manage category page, the new category is included in the all categories list. | | |
| Normal Flow: | 1. Admin access Manage Category page.  2. Admin click on Add new category page.  3. Admin inputs all necessary information.  4. Admin click Add. | | |
| Alternative Flows: | None | | |
| Exceptions: | * Validation errors. * Category already existed. | | |
| Other Information: | None | | |

### **Update category**

| ID and Name: | **UC-9 Update Category** | | |
| --- | --- | --- | --- |
| Created By: | HuongTTD | Date Created: | 11/6/2022 |
| Primary Actor: | Admin |  |  |
| Description: | Admin update an category | | |
| Trigger: | Admin access Manage Category page. | | |
| Preconditions: | Users logged in the system as Admin.  The update category must already existed. | | |
| Postconditions: | None | | |
| Normal Flow: | 1. Admin access Manage Category page.  2. On the list of all categories, admin clicks on one category.  3. The update page for that category is shown.  4. Admin input all necessary information.  5. Admin click Update. | | |
| Alternative Flows: | None | | |
| Exceptions: | * Validation errors. | | |
| Other Information: | None | | |

### **Delete category**



| ID and Name: | **UC-10 Delete Category** | | |
| --- | --- | --- | --- |
| Created By: | NgocDBT | Date Created: | 11/6/2022 |
| Primary Actor: | Admin | | |
| Description: | Admin delete category from the system | | |
| Trigger: | When admin click on delete category button | | |
| Preconditions: | PRE-1. Login using admin account  PRE-2 Must have categories | | |
| Postconditions: | POST-1. Delete selected category | | |
| Normal Flow: | **1.0 Logout**  1. Admin click on the delete category button  2. Admin choose which category to delete  3. System pop up the confirmation dialog  4. When the admin click confirm delete button, the system will display a successful delete message and delete order product | | |
| Alternative Flows: | **none** | | |
| Exceptions: | **Admin click on cancel option when the dialog is popped up**  **Return to view category** | | |
| Other Information: | none | | |

### **View order**

### 

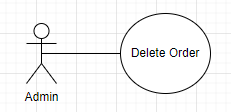
| ID and Name: | **UC-12 View Order** | | |
| --- | --- | --- | --- |
| Created By: | KhoiDT | Date Created: | 11/6/2022 |
| Primary Actor | Admin | | |
| Description: | Admin view order product of customer send to the system | | |
| Trigger: | When Admin click on Manager Oder button | | |
| Preconditions: | Login admin account | | |
| Postconditions: | none | | |
| Normal Flow: | 1. Admin accesses the Manager Order page 2. The system show list order of customer | | |
| Alternative Flows: | none | | |
| Exceptions: | none | | |
| Other Information: | none | | |

### **Update order**

### 

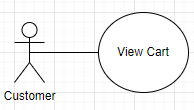
| ID and Name: | **UC-13 Update Order** | | |
| --- | --- | --- | --- |
| Created By: | KhoiDT | Date Created: | 11/6/2022 |
| Primary Actor | Admin | | |
| Description: | Admin update order product of customer on system | | |
| Trigger: | When Admin click on update button | | |
| Preconditions: | * Login admin account * Must have order product | | |
| Postconditions: | Update order product | | |
| Normal Flow: | 1. Admin click on update button 2. Admin select order product of customer to update 3. System show information of order product 4. Admin change information of order product 5. Admin click save button 6. The system displays a message ask admin to confirm the update information 7. When the admin click confirm update button, the system will display a successful update message and store the new information | | |
| Alternative Flows: | none | | |
| Exceptions: | 2a. Admin click go back button  Cancel save and system move to Manager Order page  7a. Admin click cancel button  Cancel save order product and exit table confirm | | |
| Other Information: | 5b. Open table confirm when admin click save button | | |

### **Delete order**



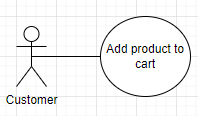
| ID and Name: | **UC-14 Delete Order** | | |
| --- | --- | --- | --- |
| Created By: | KhoiDT | Date Created: | 12/6/2022 |
| Primary Actor | Admin | | |
| Description: | Admin delete order product of customer on system | | |
| Trigger: | When Admin click on delete button | | |
| Preconditions: | * Login admin account * Must have order product | | |
| Postconditions: | Delete order product | | |
| Normal Flow: | Admin click on delete button  Admin select order product of customer to delete  Admin click ok button  The system displays a message ask admin to confirm the delete information  When the admin click confirm delete button, the system will display a successful delete message and delete order product | | |
| Alternative Flows: | none | | |
| Exceptions: | 3a. Admin click cancel button  Unselect order product  4a. Admin click cancel button  Cancel delete oder product and exit table confirm | | |
| Other Information: | 3b. Open table confirm when admin click ok button | | |

### **View cart**



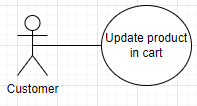
| ID and Name: | **UC-15 View cart** | | |
| --- | --- | --- | --- |
| Created By: | AnhNL | Date created | 12/06/22 |
| Primary Actor: | Customer | | |
| Description: | Customer view all his items (if available) in the shopping cart | | |
| Trigger: | When customer clicks on “View cart” button | | |
| Preconditions: | none | | |
| Postconditions: | none | | |
| Normal Flow: | **View cart**  1. Customer clicks on the “View cart” button.  2. List of items that were added to cart is displayed along with item details and total cost. | | |
| Alternative Flows: | none | | |
| Exceptions: | In case the cart is empty, a system msg ‘cart empty’ is displayed and the customer has the option to return to the home page by clicking on “back to homepage”. | | |
| Other Information: | None | | |

### **Add product to cart**



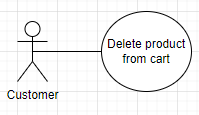
| ID and Name: | **UC-16 Add product to cart** | | |
| --- | --- | --- | --- |
| Created By: | AnhNL | Date Created: | 12/06/2022 |
| Primary Actor: | Customer | | |
| Description: | Customer add item(s) to his shopping cart | | |
| Trigger: | When customer clicks on “Add item to cart” button in view item page | | |
| Preconditions: | PRE-1. Customer must be on a view item/item details page | | |
| Postconditions: | Item(s) added to cart | | |
| Normal Flow: | **Add product to cart**  1. Customer clicks on the “Add item to cart” button.  2. Added items to cart, will remain in the same page. | | |
| Alternative Flows: | **none** | | |
| Exceptions: | none | | |
| Other Information: | In case of adding an item that is already in the cart, the quantity of that item will be increased by 1 in the cart | | |

### **Update product in cart**



| ID and Name: | **UC-17 Update product in cart** | | |
| --- | --- | --- | --- |
| Created By: | NgocDBT | Date Created: | 12/06/2022 |
| Primary Actor: | Customer | | |
| Description: | Customer change quantity of his item(s) in the shopping cart | | |
| Trigger: | When customer clicks on some specified button | | |
| Preconditions: | PRE-1. Customer must have products on his shopping cart. | | |
| Postconditions: | POST-1. Cart updated | | |
| Normal Flow: | **Update product in cart**   1. To change the quantity of items, customers click on ‘+’ to increase quantity by 1, or ‘-’ to decrease quantity by 1. 2. Changes reflected in the shopping cart page based on activities performed. | | |
| Alternative Flows: | none | | |
| Exceptions: | none | | |
| Other Information: | None | | |

### **Delete product from cart**



| ID and Name: | **UC-19 Delete product from cart** | | |
| --- | --- | --- | --- |
| Created By: | NgocDBT | Date Created: | 12/06/2022 |
| Primary Actor: | Customer | | |
| Description: | Customer delete/remove item from his shopping cart | | |
| Trigger: | When customer clicks on “Delete” button | | |
| Preconditions: | PRE-1. Customer must be on his shopping cart page. | | |
| Postconditions: | none | | |
| Normal Flow: | **Delete product**  1. Customers click on the delete item button present next to the item to delete the item from the cart.  2. Confirmation dialog will be popped up  3. Customer click on confirm button  4. The item(s) will be removed from the page if it is deleted | | |
| Alternative Flows: | none | | |
| Exceptions: | **Customer click on cancel option when the dialog is popped up**  **Return to view cart** | | |
| Other Information: | None | | |

### **Search product**

| ID and Name: | **UC-19 Search product** | | |
| --- | --- | --- | --- |
| Created By: | AnhNL | Date Created: | 12/06/2022 |
| Primary Actor: | Customer | | |
| Description: | Customer search product(s) | | |
| Trigger: | When customer clicks on “Search” button | | |
| Preconditions: | none | | |
| Postconditions: | none | | |
| Normal Flow: | **Search**  1. Customer type in the keywords related to the item like item name/ brand etc.  2. Customer clicks on the “Search” button.  2. List of products related to the item searched for will be shown | | |
| Alternative Flows: | none | | |
| Exceptions: | none | | |
| Other Information: | Displays:   * No results for the searched term/keyword. * Try checking your spelling or use more general terms. | | |

### **Check out**

| ID and Name: | **UC-20 Check Out** | | |
| --- | --- | --- | --- |
| Created By: | QuangNH | Date Created: | 11/6/2022 |
| Primary Actor | Customer |  |  |
| Description: | User make checkout when buying products | | |
| Trigger: | When user click on check out button | | |
| Preconditions: | Must have product in cart before pressing check out button | | |
| Postconditions: | None | | |
| Alternative Flows: | **1. The customer performs checkout when adding products to the cart**  **2. The system redirects to the checkout page and shows a list of all products added to cart.**  **3. The customer fills in personal information**  **4. The customer clicks the checkout button to proceed with the payment**  **5. The Payment accepted , allows the user back list product view page to perform other activities** | | |
| Exceptions: | **1.0. Information cannot be left blank**  System return to checkout page with a warning about information cannot be left blank | | |
| Priority: | High | | |
| Other Information: | None | | |

### **View order history**

| ID and Name: | **UC-21 View order history** | | |
| --- | --- | --- | --- |
| Created By: | ToanDD | Date Created: | 18/6/2022 |
| Primary Actor: | Customer | | |
| Description: | Customer must login by phone number into the system to view order history | | |
| Trigger: | When customer clicks on “Lịch sử mua hàng” button | | |
| Preconditions: | PRE-1. Customer must already login to the system | | |
| Postconditions: | POST-1. System moves the user to view the order history page. | | |
| Normal Flow: | **1.0 View order history**  1. Customer clicks on the “Lịch sử mua hàng” button  2. System shows purchase time and a table with orders  2.1 Purchase time : from date to date  2.2 A table with orders: have 1 or more orders   * Order code * Status * Product name * Price * Order date * View product detail   2.3 A table with orders: empty | | |
| Alternative Flows: | **none** | | |
| Exceptions: | **none** | | |
| Other Information: | None | | |

### **Find product by location**

| ID and Name: | **UC-22 Find product by location** | | |
| --- | --- | --- | --- |
| Created By: | ToanDD | Date Created: | 21/6/2022 |
| Primary Actor: | Customer | | |
| Description: | Customer chooses new location to find products | | |
| Trigger: | When customer clicks on “Xem giá, tồn kho tại: (vị trí)” button | | |
| Preconditions: | none | | |
| Postconditions: | POST-1. System shows a table with a list of locations for the user to choose. | | |
| Normal Flow: | **1.0 Find product by location**  1. Customer clicks on the “Xem giá, tồn kho tại: (vị trí)” button  2. System shows a table with a list locations  2.1 Field: find quick location, type: text  2.2 A table with a list of cities: city  3. System displays new location at the the “Xem giá, tồn kho tại: (vị trí)” button and load new list of products | | |
| Alternative Flows: | **none** | | |
| Exceptions: | **none** | | |
| Other Information: | None | | |

# System Features

## **Login**

### **Description**

ADMINs login into the system.

### **Functional Requirements**

Login.Show: Show a form for filling the login’s information

.Displays: Display a form for user input their username and password to login into the TGDD

**Login.Verify: Check for correctness**

.Verify : The TGDD does the authentication, shows a warning message if login’s information is wrong, and moves to the admin's main page if the user passes the authentication.

## **View Order History**

### **Description**

Admin needs to see the order history list of all customers so that they can manage orders and provide the best possible assistance for customers.

### **Functional Requirements**

**ViewOrderHistory.Request:**

.System: The system confirms that the user has logged into the system with the role Admin.

.Users: When you want to see customer order history.

**ViewOrderHistory.List:**

.Displays: displays the list of view order history for all customers.

.Sort: Sort list order history by order date

**ViewOrderHistory.View:**

.Display: Display customer’s name, address, phone number; order date, Purchase order.

## **Search**

### **Description**

Admin and customers need search tools to find products more easily, quickly and accurately.

### **Functional Requirements**

**Search.Show: Show list of products according to the entered keyword**

.Display : Display a list of product follow by keyword

**Search.Sort: Sort product list follow by category**

.Category : Sort the product list by the category that the admin and customer chooses

.Display : Display a list of products followed by category.

## **Add product to cart**

### **Description**

Customer click on the add product to cart button, the product will be added to the cart

### **Functional Requirements**

|  |
| --- |
| **Add product to cart.View: View product page**  .Display : system display a product page for customers to select the product to add to shopping cart |
| **Add product to cart. Add: Add product to cart**  Click : Customer clicks on the button add to cart at the product view page. If there is already this product in the cart, increase the quantity by 1.  **Add product to cart. Show message: Show successfully added message** |
| .show: The system displays the successfully added message on the customer screen.  .store: Store the product in a cart session. |
|  |

## **Add product**

### **Description**

The feature “Add product” helps user ver ADMIN to add a new product with all specific information about the product into the database.

### **Functional Requirements**

| **AddProduct.Show: Show a form for filling the product’s information**  .Displays: Display a form for user input specific information about a product.  .Input : Collect the information in form: product name, product code, product price, product quantity, product category, product manufacturer, product specification, product description, product image. |
| --- |
| **AddProduct.Add: Add product to database**  .Check : The system does the checking function for the .Input function.  .Failure: The system shows a warning message if the product's information is wrong.  .Save : The information of the product after checking will be saved to the database.  .Exception : Lost connection to database or server. |
| **AddProduct.Done: When function add already success, the system will do the following**  .Success : show a mess that this product is already in database  .Store : Store the product with all of the details, and can be ordered by the customer.  .Category : Update Category, this product will show when a user searches by Category.  .Manufacturer : Update Manufacturer, this product will show when a user searches by Manufacturer. |

## **View product**

### **Description**

Admin need to see all products currently existing in the system to manage the store’s catalogue.

Customers need to see all products to choose and buy.

### **Functional Requirements**

**ViewProduct.Request:**

.System: The system confirms that the user has logged into the system with the right role to view the product pages.

.Users: When you want to see the products.

**ViewProduct.List:**

.Displays:

If the current role is Customer, display all available products.

If the current role is Admin, display all products no matter their availability.

.Search: Searching for a product based on its name.

**ViewProduct.View:**

.Display: Display all information related to the product, admin can see additional information as well.

## **Check out**

### **Description**

Customer checks out when they done choosing product they want to buy

### **Functional Requirements**

**CheckOut.Show: Show the list of products customer has chose**

.Displays: Display list of products with the details (name, quantity, image,...)

.Input: Display a form of text so customers can fill their shipping’s information

**CheckOut.Done: Calculate the left quantity of product and save the order’s information**

.Calculate: Minus the amount of product that customer have bought and save into the database

.Save: save the order’s information into the database

## **View order history by phone number**

### **Description**

Users need to see all the orders made with their phone number.

### **Functional Requirements**

**ViewOrderHistory.Request:**

.System: The system sends an OTP to the phone number user inputted.

.Users: When you want to see customer order history.

**ViewOrderHistory.List:**

.Displays: displays the list of view order history for the current user after successful OTP authentication.

.Sort: Sort list order history by order date

**ViewOrderHistory.View:**

.Display: Display customer’s name, address, phone number; order date, Purchase order.

## **Find product through location**

### **Description**

Find available products by the selected location.

### **Functional Requirements**

**FindProductByLocation.Search:**

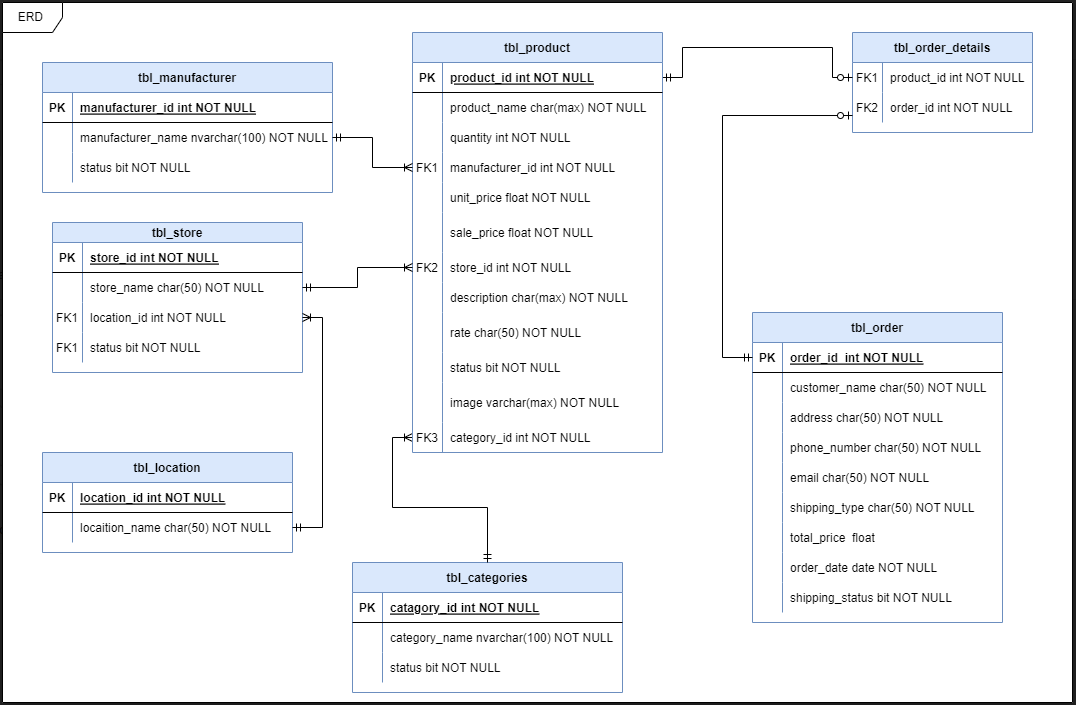
.System: get the list of products available by the selected location.

**ViewOrderHistory.View:**

.Display: Displays available products according to the selected location.

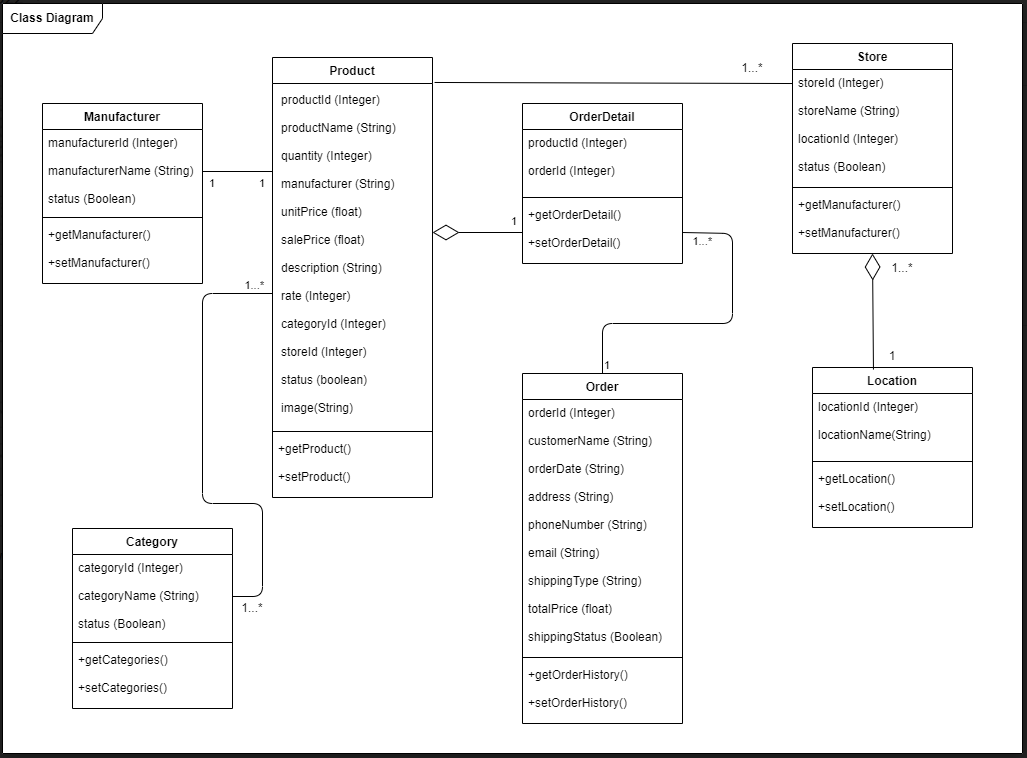
# **Data Requirements**

## **Logical Data Model**



**Figure 2. Partial data model of the TGDD.**

## **Class Diagram**



**Figure 3. Partial data model of the TGDD.**

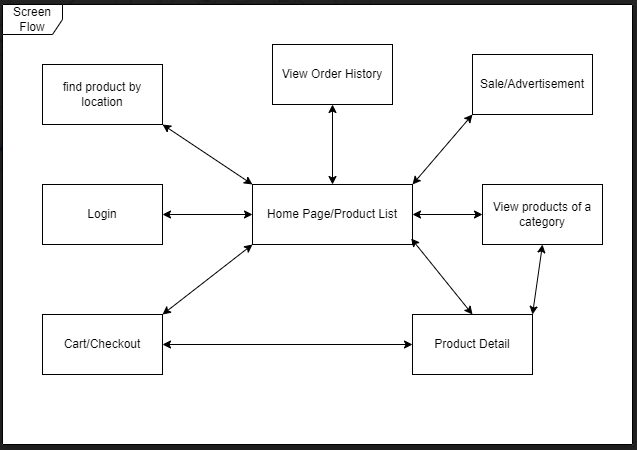
# **External Interface Requirements**

## **Communications Interfaces**

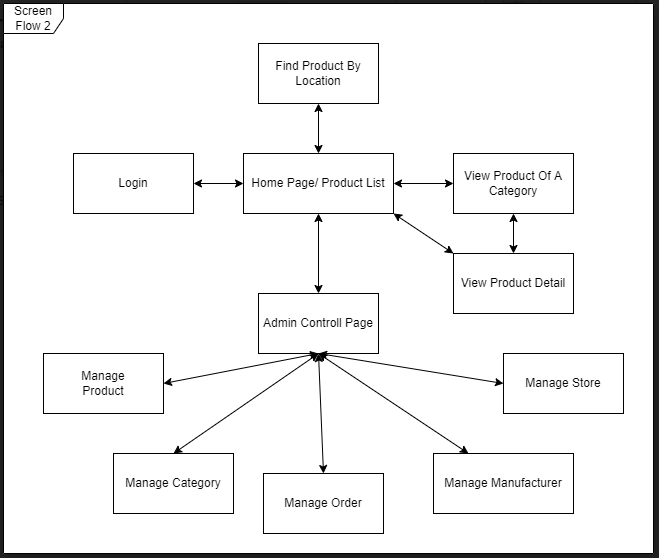
CI-1: The TGDD shall send an email or a text message to the Customer to inform the order

CI-2: The TGDD shall send an email or text message to the Customer to report any problems with the order.

## **Screen Flow**



**Figure 4. Screen Flow of Customer Feature of the TGDD.**

****

**Figure 5. Screen Flow of Admin Feature of the TGDD.**

# **Non-functional Requirements**

## **Security Requirements**

**-** SR-1: Admin accounts’ passwords shall be encrypted using SHA-256 encryption.

## **Performance Requirements**

- PER-1: 95% of the screens shall load with the maximum time of 3 seconds from the moment the customer send a request

## **Assumptions and Dependencies**

- AS-1: An admin can deny a Customer’s order.

# 