VIETNAM GENERAL CONFEDERATION OF LABOR TON DUC THANG UNIVERSITY

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL REPORT**

SERVICE-ORIENTED ARCHITECTURE

*Supervisor*: **DUONG HUU PHUC** *Student*:  **VO NGUYEN LONG – 519H0027**

**PHAN LE MINH NHUT – 519H0211**

Class**: 19H50204**

Course**: 23**

**HO CHI MINH CITY, 2022**

VIETNAM GENERAL CONFEDERATION OF LABOR TON DUC THANG UNIVERSITY

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL REPORT**

SERVICE-ORIENTED ARCHITECTURE

*Supervisor*: **DUONG HUU PHUC** *Student*:  **VO NGUYEN LONG – 519H0027**

**PHAN LE MINH NHUT – 519H0211**

Class**: 19H50204**

Course**: 23**

**HO CHI MINH CITY, 2022**

**ACKNOWLEDGEMENT**

First of all, I would like to express my deepest gratitude to teacher Duong Huu Phuc, who directly guided and enthusiastically supported me during the implementation of this final report. The teacher helped me to know the presentation of reports and review for me. I appreciate Mr. Phuc who is the advisor of the Society of Actuaries course for imparting knowledge in the second semester to me. The knowledge acquired during the course of study is not only the foundation for the reporting process but also valuable baggage for me to apply in my future work. Therefore, the report topic is a summary of theory and practice, between the knowledge that I had learned at Ton Duc Thang University.

The way I present my answers is still limited. Therefore, inevitably, I am looking forward to receiving valuable comments from you and lecturers in the faculty to improve and perfect my knowledge in this subject, to better serve my future work.

Finally, I hope you and teachers in the Faculty of Information Technology have good health and enthusiasm to continue to carry out their noble mission of imparting knowledge to future generations, always successful in careers.

Thank you for reading !

**THE REPORT IS COMPLETED**

**IN TON DUC THANG UNIVERSITY**

I assure that this is my own project and is guided by Mr. Phuc; The research contents and results in this topic are honest and have not been published in any form before. The data in the tables for analysis, comments, and evaluation are collected by the author himself from different sources, clearly stated in the reference section.

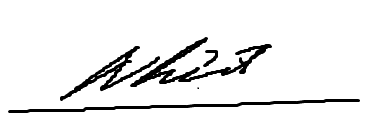
In addition, the project also uses a number of comments, assessments as well as data of other authors, other agencies, and organizations, with citations and source annotations.

**If the project is found to have any fraud, I would like to take full responsibility for the content of my project. Ton Duc Thang University is not related to copyright and copyright violations caused by me during the implementation process (if any).**

*Ho Chi Minh City, 22th July 2022 Author*

**

*Vo Nguyen Long*

**

*Phan Le Minh Nhut*

**Comment**

Ho Chi Minh City, Date Month Year (Sign and write full name)

**TABLE OF CONTENTS**

[1. INTRODUCTION 8](#_Toc120958547)

[1.1 Structure of the document 8](#_Toc120958548)

[1.2 System Capabilities 8](#_Toc120958549)

[2. REQUIREMENT SPECIFICATIONS 9](#_Toc120958550)

[2.0 Use case model: 9](#_Toc120958551)

[2.1 Textual description for each use case 10](#_Toc120958552)

[*2.1.1 Textual description for use case ‘Login’* 10](#_Toc120958553)

[*2.1.2 Textual Description for use cases ‘Create order’, ‘Create payment’, ‘Pay card’* 12](#_Toc120958554)

[*2.1.3 Textual Description for use cases ‘Search products’, ‘Add to cart’* 13](#_Toc120958555)

[*2.1.4 Textual description for use cases ‘Delete products’* 13](#_Toc120958556)

[*2.1.5 Textual Description for use cases ‘Update product’* 14](#_Toc120958557)

[*2.1.6 Textual Description for use cases ‘Sign up’* 14](#_Toc120958558)

[2.2 Functional requirements/ Non-functional requirements 15](#_Toc120958559)

[2.3 ERD diagram 15](#_Toc120958560)

[2.4 Class diagram 16](#_Toc120958561)

[3. WEBSITE 18](#_Toc120958562)

[3.1 API docs 18](#_Toc120958563)

[3.2 UI designs 19](#_Toc120958564)

[4. DATABASE DESIGN 27](#_Toc120958565)

[4.1 Login Screen 27](#_Toc120958566)

[*4.1.1 Main flow* 27](#_Toc120958567)

[*4.1.2 Validation rule/Data mapping* 28](#_Toc120958568)

[*4.1.3 Design Database* 28](#_Toc120958569)

[4.2 Homepage Screen 29](#_Toc120958570)

[*4.2.1 Main flow* 29](#_Toc120958571)

[*4.2.2 Validation rule/Data mapping* 29](#_Toc120958572)

[4.3 Registration Screen 30](#_Toc120958573)

[*4.3.1 Main flow* 30](#_Toc120958574)

[*4.3.2 Validation Rule/Data mapping* 30](#_Toc120958575)

[*4.3.3 Design database* 31](#_Toc120958576)

[4.4 Cart screen 32](#_Toc120958577)

[*4.4.1 Main Flow* 32](#_Toc120958578)

[*4.4.2 Validation rule/Data mapping* 32](#_Toc120958579)

[4.5 Payment screen 33](#_Toc120958580)

[*4.5.1 Main flow* 33](#_Toc120958581)

[*4.5.2 Validation rule/Data mapping* 33](#_Toc120958582)

[4.6 Admin’s login screen 34](#_Toc120958583)

[*4.6.1 Main flow* 34](#_Toc120958584)

[*4.6.2 Validation rule/Data mapping* 34](#_Toc120958585)

[*4.6.3 Design Database* 34](#_Toc120958586)

[4.7 Admin Homepage screen 35](#_Toc120958587)

[*4.7.1 Main flow* 35](#_Toc120958588)

[*4.7.2 Validation rule/Data mapping* 35](#_Toc120958589)

[4.8 Admin brand management screen 36](#_Toc120958590)

[*4.8.1 Main flow* 36](#_Toc120958591)

[*4.8.2 Validation rule/Data mapping* 36](#_Toc120958592)

[4.9 Admin categories management screen 37](#_Toc120958593)

[*4.9.1 Main flow* 37](#_Toc120958594)

[*4.9.2 Validation rule/Data mapping* 37](#_Toc120958595)

[4.10 Add product screen 38](#_Toc120958596)

[*4.10.1 Main flow* 38](#_Toc120958597)

[*4.10.2 Validation Rule/Data mapping* 38](#_Toc120958598)

[*4.10.3 Design database* 39](#_Toc120958599)

[4.11 Add brand screen 41](#_Toc120958600)

[*4.11.1 Main flow* 41](#_Toc120958601)

[*4.11.2 Validation Rule/Data mapping* 41](#_Toc120958602)

[*4.11.3 Design database* 41](#_Toc120958603)

[4.12 Add category screen 42](#_Toc120958604)

[*4.12.1 Main flow* 42](#_Toc120958605)

[*4.12.2 Validation Rule/Data mapping* 42](#_Toc120958606)

[*4.12.3 Design database* 42](#_Toc120958607)

[5. PROJECT MANAGER PLAN 43](#_Toc120958608)

# INTRODUCTION

## 1.1 Structure of the document

This document includes 5 parts:

|  |  |  |
| --- | --- | --- |
| Part | Content | Detail |
| 1 | Introduction | - Structure of document  - System capabilities |
| 2 | Requirement specifications | - Specify the functions included in the system, draw the use case diagram, ERD diagram and class diagram.  - Determine the functional and non-functional requirements of the system. |
| 3 | Website | - API docs  - UI design |
| 4 | Database design | Main flow, validation rule and data design |
| 5 | Project Organization | Contribution of each members |

Table 1 - Structure of the document

## 1.2 System Capabilities

- Connecting via bank and authorizing the payment.

- Collecting information about products

- Increase sales productivity with detailed searching products.

- Back up and restore when the web is down.

- Saving time and being easy to use.

- Maintain correct information of customers and products.

# REQUIREMENT SPECIFICATIONS

# 

## Use case model:

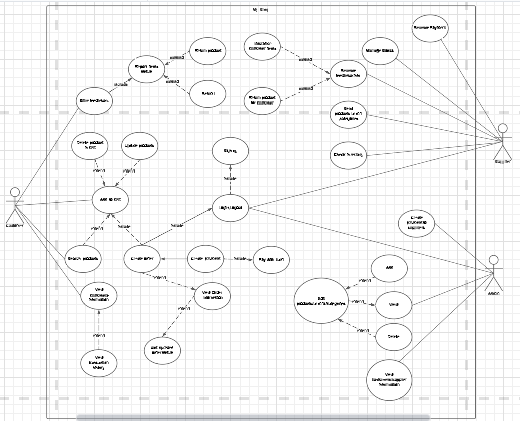
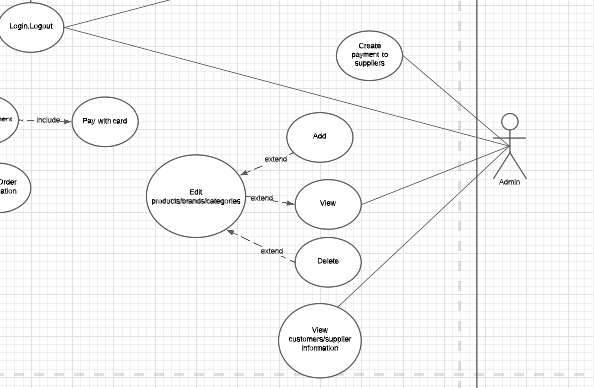
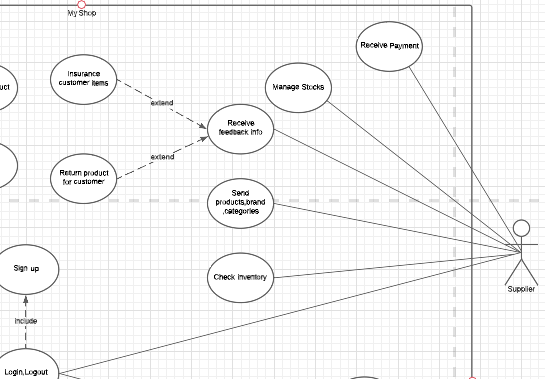
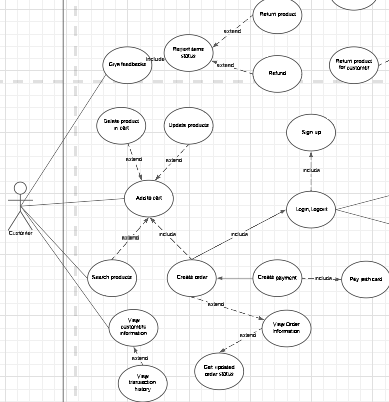


Figure 2 – Use case diagram







## 2.1 Textual description for each use case

***2.1.1 Textual description for use case ‘Login’***

|  |  |  |
| --- | --- | --- |
| Use case name: | Login | |
| Participating actors | User | |
| Entry condition(s) | The website used to place the order must be active  The database system must be avaiable | |
| Normal flow of events | Actor  Access Login interface  Input email address and password then click on the Login button | System  Display a login interface.  System checks whether the information is valid or not.  Displays a homepage interface |
| Exit condition(s) | Login successful |  |
| Exceptions (Alternate flow of events) | If the information is invalid, display an error message back to the login interface and display notification requires the user to input email address and password again. |  |
| Special requirements | None |  |

Table 4 - Textual description for use case ‘Login’

***2.1.2 Textual Description for use cases ‘Create order’, ‘Create payment’, ‘Pay card’***

|  |  |  |
| --- | --- | --- |
| Use case name: | Create order, Create payment, Pay card | |
| Participating actors | User | |
| Entry condition(s) | The website used to place the order must be active  User’s account must exist | |
| Normal flow of events | Actor | System |
| 1. Access cart 2. Customer clicks on the button “Order now”. 3. Customer clicks on the button “Pay with Card” 4. Customer enter car information then click “Pay” button | 1.1 Display cart interface.  2.1 System displays payment interface.  3.1 Display Card popup window  4.1 Display successful notification |
| Exit condition(s) | Ordering is completed | |
| Exceptions (Alternate flow of events) | None | |
| Special requirements | None | |

None

Special requirements

Table 5 - Textual description for use cases ‘Create order’, ‘Create payment’, ‘Pay Card’

***2.1.3 Textual Description for use cases ‘Search products’, ‘Add to cart’***

|  |  |  |
| --- | --- | --- |
| Use case name: | Search products, add to cart | |
| Participating actors | Users | |
| Entry condition(s) | The website used to place the order must be active  Users’s account must exist | |
| Normal flow of events | Actor | System |
| 1. Access Home page 2. Search item 3. Click Add to cart button 4. Click Details button   4.2. Select quantity and color than click Add to cart | 1.1 Display Home page  2.1 Display results  3.1 Update cart  4.1 Display details information of product  4.3 Update cart |
| Exit Condition(s) | Customer can add product into cart | |
| Exceptions (Alternate flow of events) | None | |
| Special requirements | None | |

Table 6 - Textual description for use cases ‘Search item’, ‘Add item’

***2.1.4 Textual description for use cases ‘Delete products’***

|  |  |  |
| --- | --- | --- |
| Use case name: | Delete products in cart | |
| Participating actors | Users | |
| Entry condition(s) | The website used to place the order must be active  Users’s account must exist | |
|  | Actor | System |

|  |  |  |
| --- | --- | --- |
| Normal flow of Events | 1. Access Cart page   1. Click Delete button to remove one line in cart 2. Click Clear All to remove all product in cart | 1.1 Display Cart page  2.1 Update cart  3.1 Update cart and return Home page |
| Exit condition(s) | Customer can remove product from cart | |
| Exceptions (Alternate flow of events) | None | |
| Special requirements | None | |

Table 7 - Textual description for use cases ‘Remove item’

***2.1.5 Textual Description for use cases ‘Update product’***

|  |  |  |
| --- | --- | --- |
| Use case name: | Update product | |
| Participating actors | Users | |
| Entry condition(s) | The website used to place the order must be active  Users’s account must exist | |
| Normal flow of events | Actor | System |
| 1. Access Cart page   1. Select quantity of product    1. Click Update button | 1.1 Display Cart page  2.2 Update cart |
| Exit condition(s) | Customer can update product in cart | |
| Exceptions (Alternate flow of events) | None | |
| Special Requirements | None | |

Table 8 - Textual description for use cases ‘Remove item’

***2.1.6 Textual Description for use cases ‘Sign up’***

|  |  |
| --- | --- |
| Use case name: | Sign up |
| Participating actors | Users |

|  |  |  |
| --- | --- | --- |
| Entry condition(s) | The website used to place the order must be active  Users’s account must exist | |
| Normal flow of events | Actor | System |
| 1. Access Registration interface   1. Fill in the form to create a new account    1. Click Register button | 1.1 Display Registration interface  2.2 Save account in database, display Login interface and the message Welcome. |
| Exit condition(s) | Customer can update product in cart | |
| Exceptions (Alternate flow of events) | 2 If customer input invalid information, the system will display notice message for them | |
| Special requirements | None | |

Table 9 - Textual description for use cases ‘Sign up’

## 2.2 Functional requirements/ Non-functional requirements

**Functional:**

The system must allow the user to choose the product (include: remove/add/ update items to cart).

The system must allow the user to place the order. The system must allow the user to pay by card.

The system provide user with warranty and refunds service.

The system must allow the admin to edit/add/delete all the products on the website.

The system must allow the admin to have control over user accounts or supplier accounts.

The system must allow the supplier to mange stocks of products on the website.

The system must allow the supplier to registry products to sell.

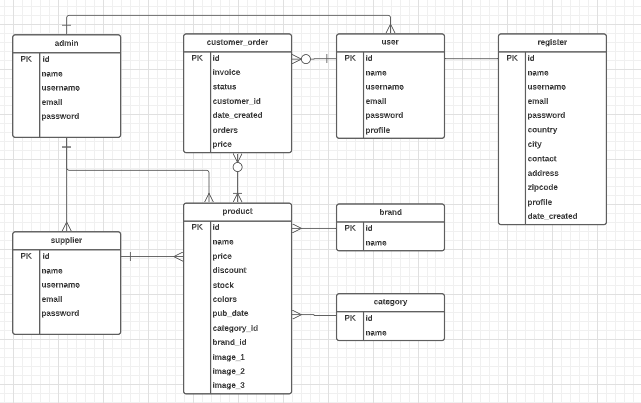
**Non-functional** :

The system must allow users to log in/ log out/ registration

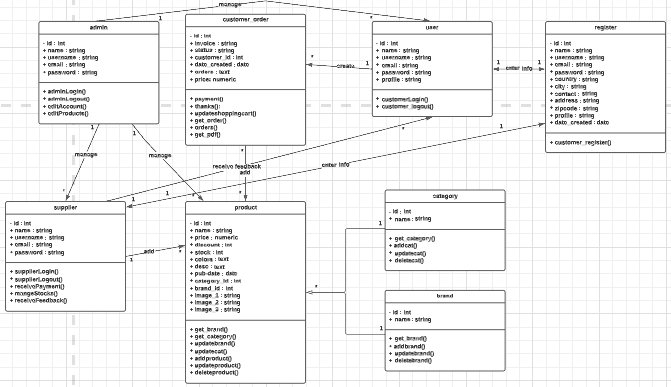
System resources must be secured. User-friendly interface.

The system operations must take no more than 2 seconds.

## 2.3 ERD diagram

****

## 2.4 Class diagram



# WEBSITE

## 3.1 API docs

|  |  |  |
| --- | --- | --- |
| **URL** | **Method** | **Description** |
| /customer/login | GET/POST | The user login interface. When user enter his/her information and click on “Login” button they will be logged in (with correct user account) |
| / |  | Homepage of the online shop. This page include the navigation bar and all of the products available for purchase |
| /customer/register | GET/POST | The user register interface. User enter his/her information to create an account that can be used to purchase item(s) on this site |
| /carts |  | The user’s cart. This page will show every item that user has put into his/her cart for purchase |
| /updatecart/<int:code> | POST | Update the change user make to items in cart. |
| /deleteitem<int:id> |  | Choose to delete items in cart. |
| /addcart | POST | Put items user choose to cart |
| /orders/<invoice> |  | The payment page. This page will show current item in user’s cart and when click on “Pay with cart” button, a payment form will show up for user to enter payment information such as card number, cvc,… |
| /payment | POST | Create a payment function with Stripe |
| /get\_pdf/<invoice> | POST | Create a transaction informations in pdf form |
| /login | GET/POST | The login screen for admin. This screen include the login button that when enter a valid account and click “Login” will take admin to admin page |
| /admin |  | Homepage of the online shop. This page include the navigation bar and all of the products available for purchase |
| /brands |  | Display every brands that has been added by admin |
| /categories |  | Display every categories that has been added by admin |
| /product |  | Display every products that has been added by admin |
| /addproduct  /addbrand  /addcat | GET/POST | The admin add product/brand/category interface. Here admin can add new product/brand/category |
| /updateproduct  /updatebrand  /updatecat | GET/POST | The admin edit product/brand/category interface. Here admin can edit product/brand/category informations. |
| /deleteproduct  /deletebrand  /deletecat | GET/POST | The admin delete the chosed product/brand/category. |
|  |  |  |

## 3.2 UI designs

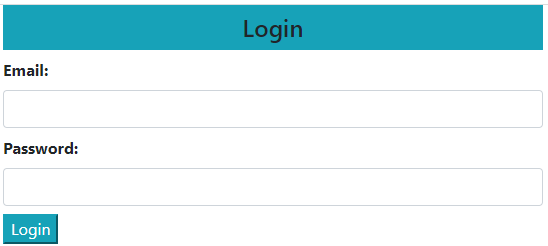


Figure 3 – Login screen

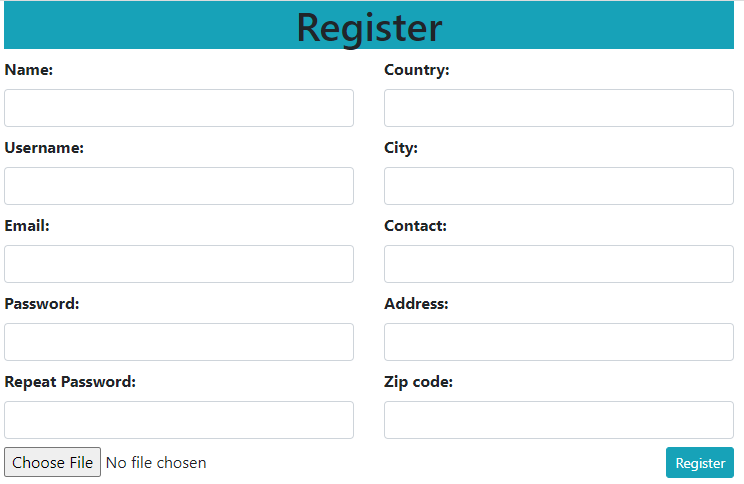


Figure 4 – Register screen

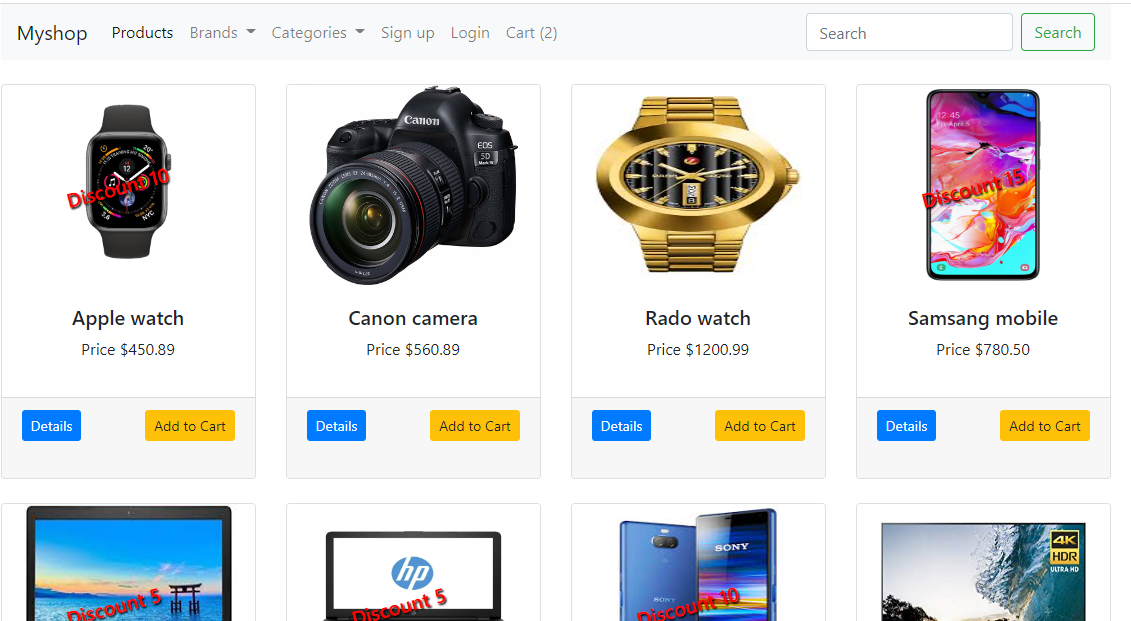


Figure 5 – Homepage screen

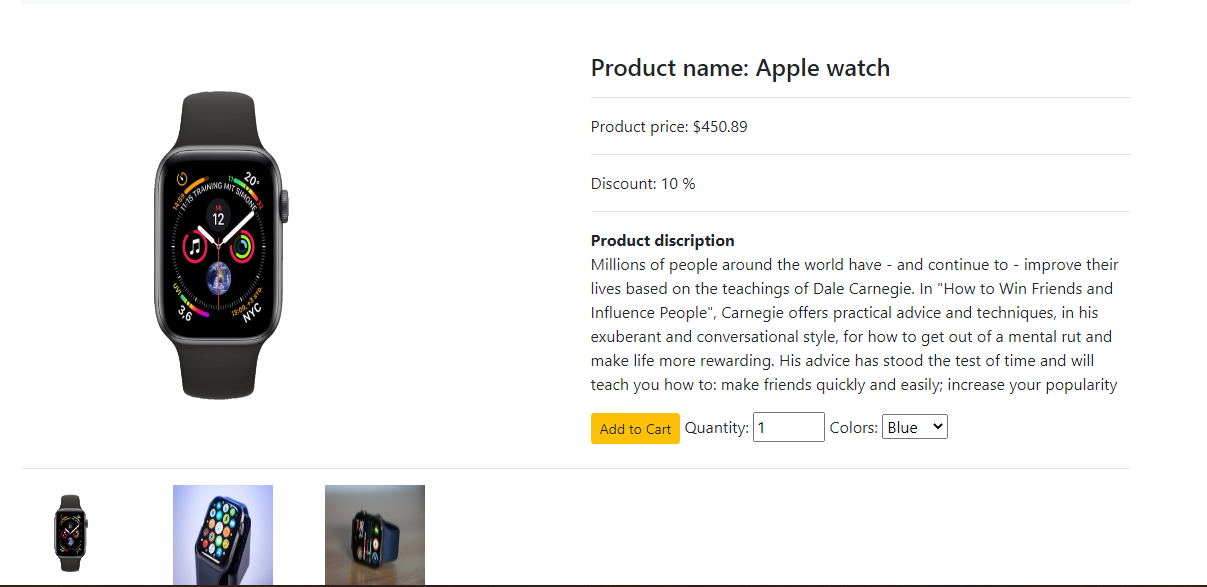


Figure 6 – Detailed product

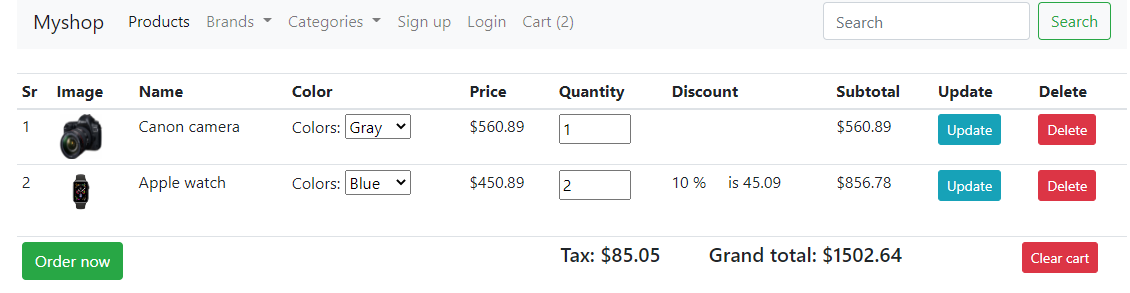


Figure 7 – Cart Screen

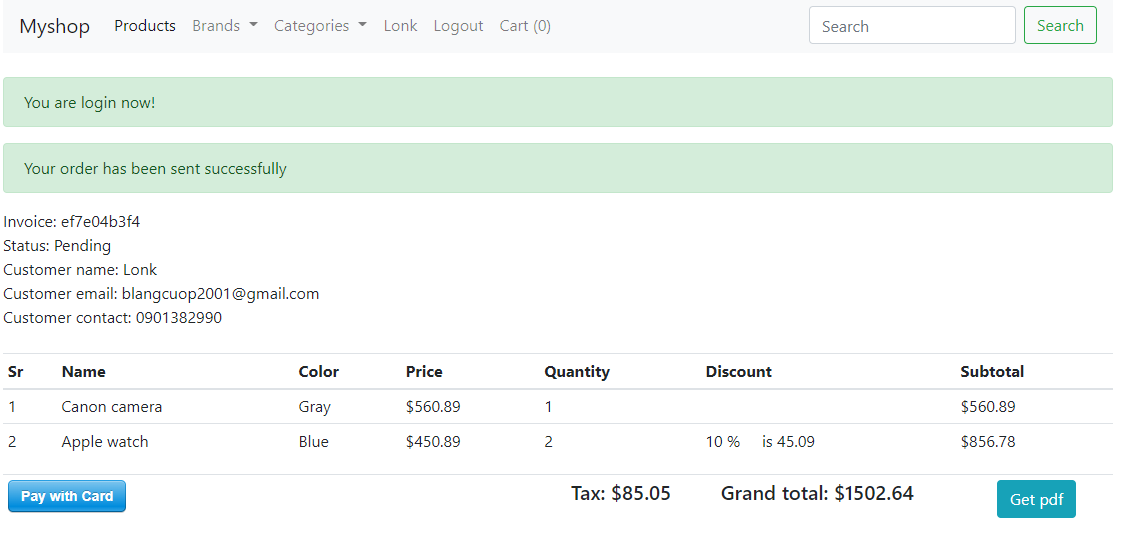


Figure 8 – Order now Screen

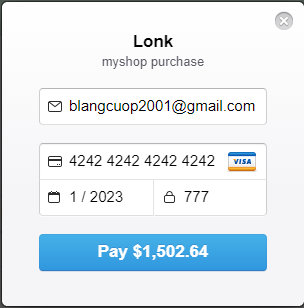


Figure 9 – Pay with Card screen

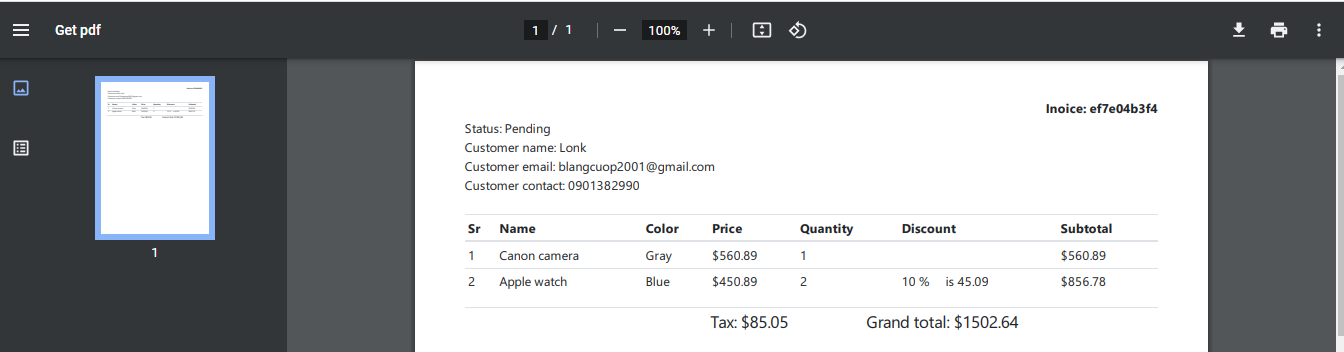


Figure 10 – Get pdf

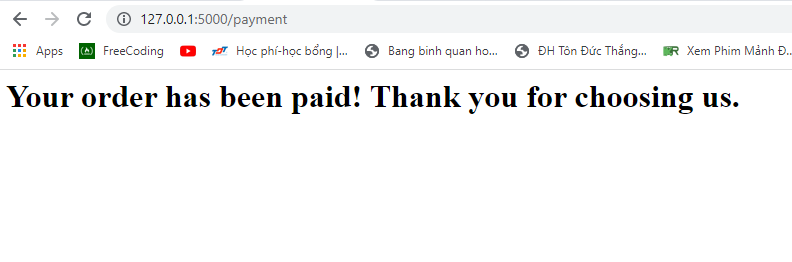


Figure 11 – Paid successful

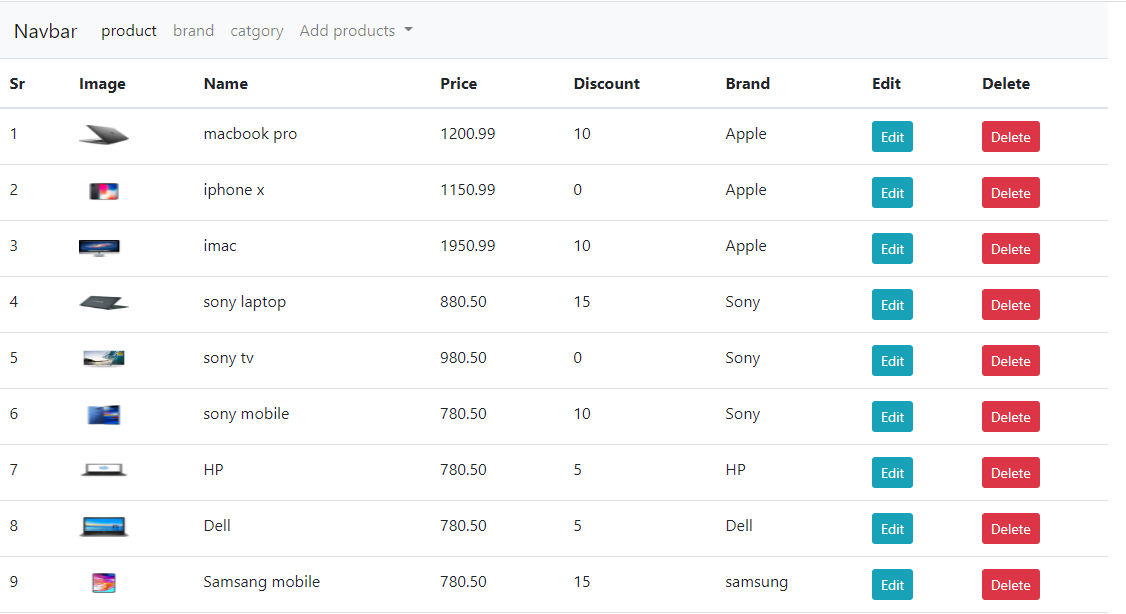


Figure 12 – Admin product screen

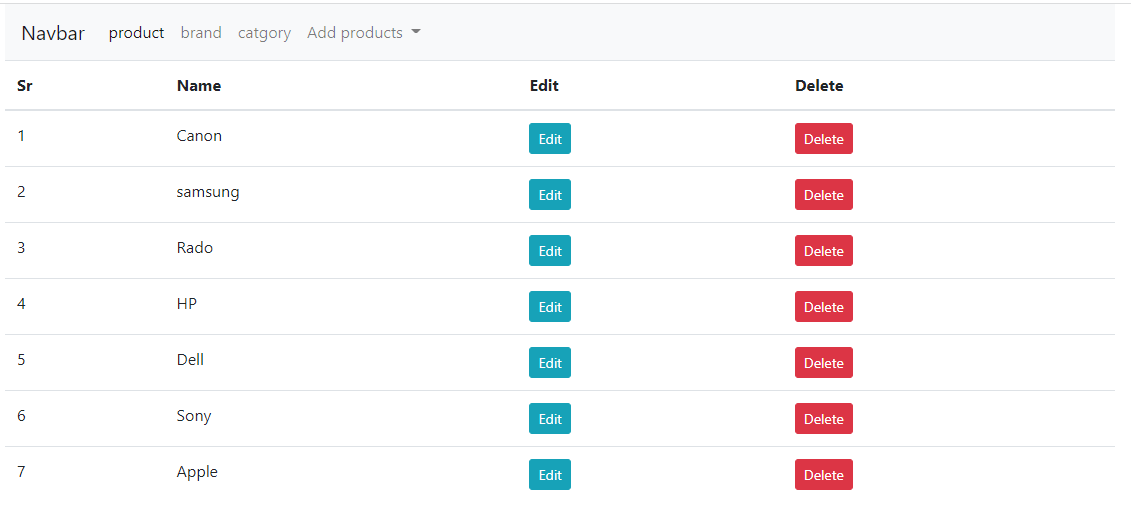


Figure 13 – Admin brand screen

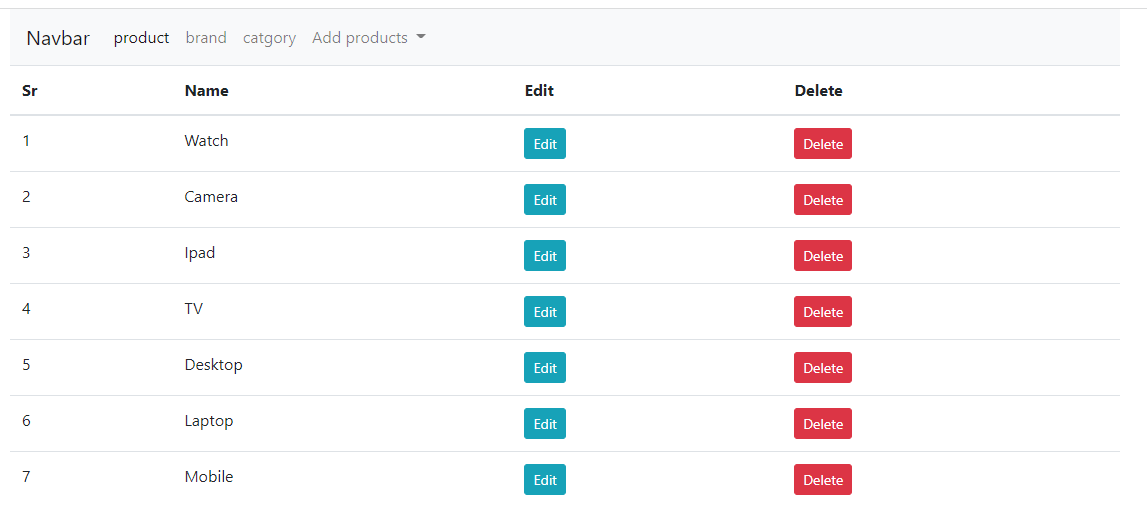


Figure 14 – Admin category screen

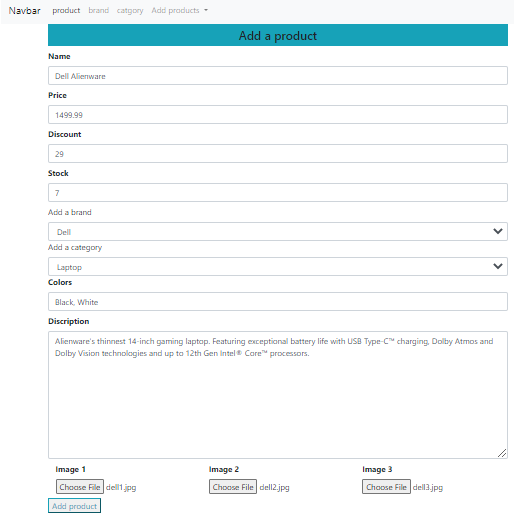


Figure 15 – Add product screen

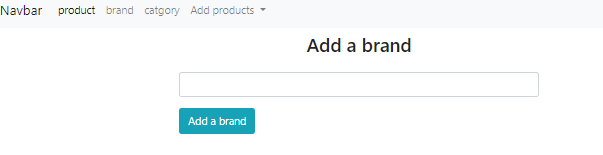


Figure 16 – Add brand screen

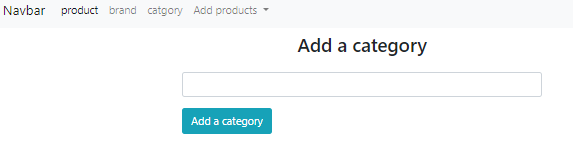
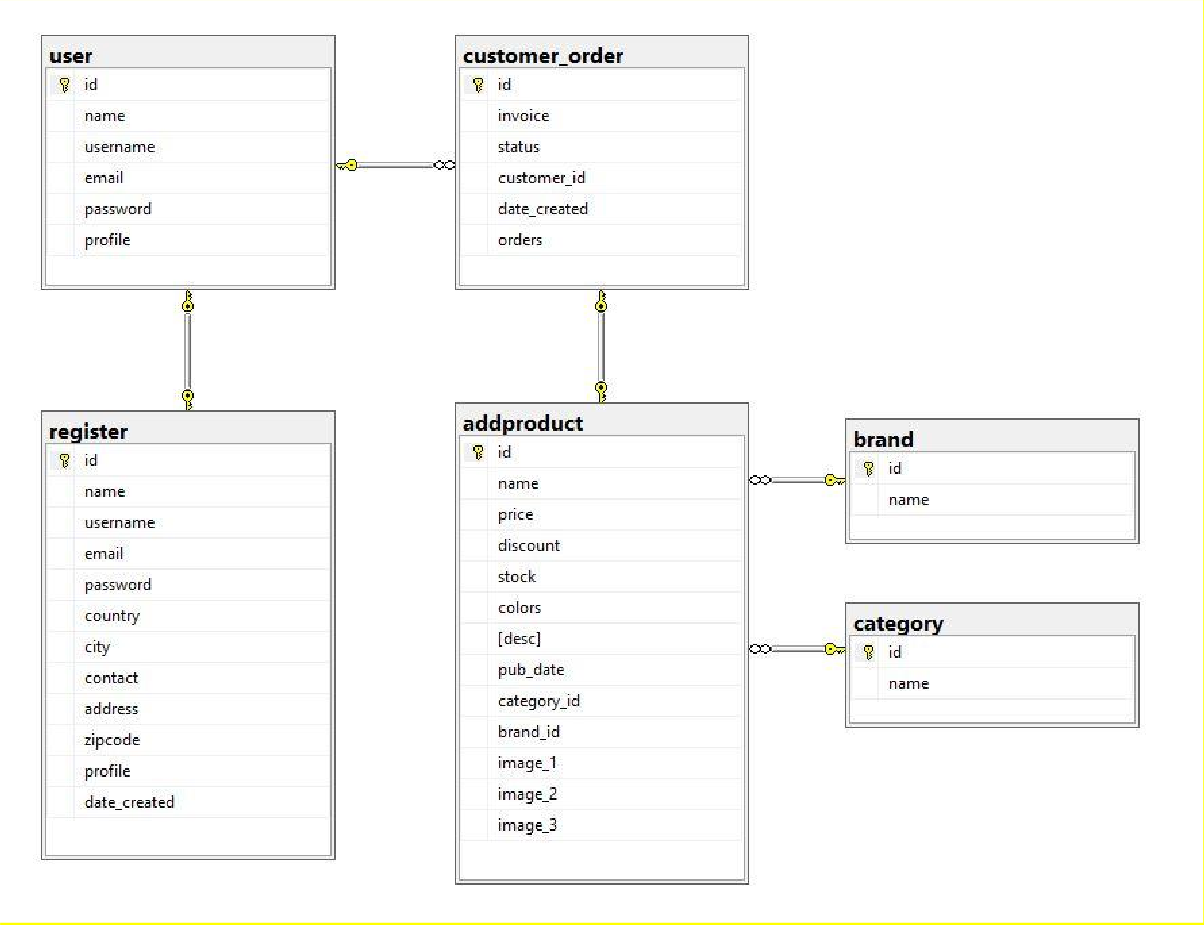


Figure 17 – Add Category screen

1. **DATABASE DESIGN**



**4.1 Login Screen**

### *4.1.1 Main flow*

|  |  |
| --- | --- |
| **Screen** | Login interface |
| **Description** | Allow users to log in to the software. |
| **Screen access** | Users input email and password then click on the login button on the login screen |

*4.1.2 Validation rule/Data mapping*

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Email | Textbox | User account  Must be String and unique  Insert to the field in Database: account. Email |
| Password | Textbox | User password Must be String  Insert to the field in Database: account. Password |
| Login | Button | Login button  Allow users access to the software |

### *4.1.3 Design Database*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Default** | **Description** |
| Email | String (50) |  |  | User account |
| Password | String (200) |  |  | User password |

## 4.2 Homepage Screen

### *4.2.1 Main flow*

|  |  |
| --- | --- |
| **Screen** | Online shop homepage |
| **Description** | Display the main interface of the website  From this screen, the user can access the website |
| **Screen Access** | Users click Product/Brands/Categories/Register/Login/Cart |

### *4.2.2 Validation rule/Data mapping*

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Product | Icon button | Product icon button  Allow users access to the Product interface |
| Brands | Icon button | Brands icon button  Allow users access to the Brands interface |
| Categories | Icon button | Categories icon button  Allow users access to the Categories interface |
| Register | Icon button | Register icon button  Allow users access to the Register interface |
| Login | Icon button | Login icon button allow users access to the Login interface |
| Cart | Icon button | Cart icon button  Allow users access to the Cart interface |

## 4.3 Registration Screen

*4.3.1 Main flow*

|  |  |
| --- | --- |
| **Screen** | Register interface |
| **Description** | Allow user to create an account |
| **Screen access** | User enter their informations to create an account |

*4.3.2 Validation Rule/Data mapping*

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Name | Textbox | The ID of the received note Must be String  Insert to the field in Database: import. Name |
| Username | Textbox | The ID of the received note Must be String and unique  Insert to the field in Database: import. Username |
| Email | Textbox | The ID of the received note Must be String and unique  Insert to the field in Database: import. Email |
| Password | Textbox | The ID of the received note Must be String  Insert to the field in Database: import. Password |
| Repeat Password | Textbox | The ID of the received note Must be String  Insert to the field in Database: import. Password |
| Country | Textbox | The ID of the received note Must be String  Insert to the field in Database: import. Country |
| City | Textbox | The ID of the received note Must be String  Insert to the field in Database: import. City |
| Contact | Textbox | The ID of the received note Must be String  Insert to the field in Database: import. Contact |
| Address | Textbox | The ID of the received note Must be String  Insert to the field in Database: import. Address |
| Zip code | Textbox | The ID of the received note Must be String  Insert to the field in Database: import. Zip code |

*4.3.3 Design database*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Null** | **Default** | **Description** |
| Enter Name | String (50) | Not null |  | The Name |
| Enter Username | String (50) | Not null |  | The Username |
| Enter Email | String (50) | Not null |  | The Email |
| Enter Password | String (200) | Not null |  | The Password |
| Enter Repeat Password |  | Not null |  |  |
| Enter Country | String (50) | Not null |  | The Country |
| Enter City | String (50) | Not null |  | The City |
| Enter Contact | String (50) | Not null |  | The contact |
| Enter Address | String (50) | Not null |  | The address |
| Enter Zip code | String (50) | Not null |  | The Zip code |

## 4.4 Cart screen

*4.4.1 Main Flow*

|  |  |
| --- | --- |
| **Screen** | Cart |
| **Description** | User can make a purchase |
| **Screen access** | User click on “Order now” button to proceed to payment page |

*4.4.2 Validation rule/Data mapping*

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Product | Icon button | Product icon button  Allow users access to the Product interface |
| Brands | Icon button | Brands icon button  Allow users access to the Brands interface |
| Categories | Icon button | Categories icon button  Allow users access to the Categories interface |
| Register | Icon button | Register icon button  Allow users access to the Register interface |
| Login | Icon button | Login icon button  Allow users access to the Login interface |
| Order now | Icon button | Order now icon button Allow to proceed to purchase |

## 4.5 Payment screen

*4.5.1 Main flow*

|  |  |
| --- | --- |
| **Screen** | Payment interface |
| **Description** | Allow user to purchase item(s) |
| **Screen access** | User click “Pay with card” to proceed purchase |

*4.5.2 Validation rule/Data mapping*

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Pay with card | Button | Log out button  Allow accountant to log out of the system |

## 4.6 Admin’s login screen

*4.6.1 Main flow*

|  |  |
| --- | --- |
| **Screen** | Admin login interface |
| **Description** | Allow admin to login into admin control page |
| **Screen access** | Admin input email and password then click on the login button on the login screen |

*4.6.2 Validation rule/Data mapping*

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Email | Textbox | Admin account  Must be String and unique  Insert to the field in Database: account. Email |
| Password | Textbox | Admin password Must be String  Insert to the field in Database: account. Password |
| Login | Button | Login button  Allow users access to the software |

*4.6.3 Design Database*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Default** | **Description** |
| Email | String (120) |  |  | Admin account |
| Password | String (120) |  |  | Admin password |

## 4.7 Admin Homepage screen

*4.7.1 Main flow*

|  |  |
| --- | --- |
| **Screen** | Admin homepage |
| **Description** | Display the main interface of the admin interface From this screen, the admin can access the website |
| **Screen Access** | Admin click Product/Brands/Category/Add Products |

### *4.7.2 Validation rule/Data mapping*

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Product | Icon button | Product icon button  Allow admin access to the Product interface |
| Brands | Icon button | Brands icon button  Allow admin access to the Brands interface |
| Category | Icon button | Categories icon button  Allow admin access to the Categories interface |
| Add product | Dropdown list | List what the admin want to add to the shop |
| Edit | Icon button | Edit icon button  Allow admin to edit a product |
| Delete | Icon button | Delete icon button  Allow admin to delete a product |

## 4.8 Admin brand management screen

### *4.8.1 Main flow*

|  |  |
| --- | --- |
| **Screen** | Admin brand screen |
| **Description** | Display the brand interface of the admin interface From this screen, the admin can access the website |
| **Screen Access** | Admin click Product/Brands/Category/Add Products/Edit/Delete |

*4.8.2 Validation rule/Data mapping*

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Product | Icon button | Product icon button  Allow admin access to the Product interface |
| Brands | Icon button | Brands icon button  Allow admin access to the Brands interface |
| Category | Icon button | Categories icon button  Allow admin access to the Categories interface |
| Add product | Dropdown list | List what the admin want to add to the shop |
| Edit | Icon button | Edit icon button allow admin to edit a brand |
| Delete | Icon button | Delete icon button  Allow admin to delete a brand |

## 4.9 Admin categories management screen

### *4.9.1 Main flow*

|  |  |
| --- | --- |
| **Screen** | Admin category screen |
| **Description** | Display the category interface of the admin interface From this screen, the admin can access the website |
| **Screen Access** | Admin click Product/Brands/Category/Add Products/Edit/Delete |

*4.9.2 Validation rule/Data mapping*

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Product | Icon button | Product icon button  Allow admin access to the Product interface |
| Brands | Icon button | Brands icon button  Allow admin access to the Brands interface |
| Category | Icon button | Categories icon button  Allow admin access to the Categories interface |
| Add product | Dropdown list | List what the admin want to add to the shop |
| Edit | Icon button | Edit icon button  Allow admin to edit a category |
| Delete | Icon button | Delete icon button  Allow admin to delete a category |

## 4.10 Add product screen

### *4.10.1 Main flow*

|  |  |
| --- | --- |
| **Screen** | Add product interface |
| **Description** | Allow admin to add new product |
| **Screen access** | Admin enter new product’s informations |

*4.10.2 Validation Rule/Data mapping*

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Name | Textbox | The ID of the received note Must be String  Insert to the field in Database: import. Name |
| Price | Textbox | The ID of the received note Must be Numeric  Insert to the field in Database: import. Price |

|  |  |  |
| --- | --- | --- |
| Discount | Textbox | The ID of the received note Must be Integer  Insert to the field in Database: import. Discount |
| Stock | Textbox | The ID of the received note Must be Integer  Insert to the field in Database: import. Stock |
| Add a brand | Dropdown list | The ID of the received note Must be String  Insert to the field in Database: import. Brand |
| Add a category | Dropdown list | The ID of the received note Must be String  Insert to the field in Database: import. Category |
| Colors | Textbox | The ID of the received note Must be Text  Insert to the field in Database: import. Colors |
| Description | Textbox | The ID of the received note Must be Text  Insert to the field in Database: import. Description |
| Image 1 | Textbox | The ID of the received note Must be String  Insert to the field in Database: import. Image 1 |
| Image 2 |  | The ID of the received note Must be String  Insert to the field in Database: import. Image 2 |
| Image 3 |  | The ID of the received note Must be String  Insert to the field in Database: import. Image 3 |

*4.10.3 Design database*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Null** | **Default** | **Description** |
| Name | Integer | Not null |  | Product’s name |
| Price | String (80) | Not null |  | Product’s price |
| Discount | Integer | Not null |  | Discount |
| Stock | Integer | Not null |  |  |
| Add a brand |  | Not null |  | Product’s band |
| Add a category |  | Not null |  | Product’s category |
| Colors | Text | Not null |  | Product’s color |
| Description | String (Text) | Not null |  | Product description |
| Image 1 | String (150) | Not null |  | Image 1 |
| Image 2 | String (150) | Not null |  | Image 2 |
| Image 3 | String (150) | Not null |  | Image 3 |

## 4.11 Add brand screen

*4.11.1 Main flow*

|  |  |
| --- | --- |
| **Screen** | Add brand interface |
| **Description** | Allow admin to add new brand |
| **Screen access** | Admin enter new brand’s informations |

*4.11.2 Validation Rule/Data mapping*

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Brand name | Textbox | The ID of the received note Must be String  Insert to the field in Database: import. Brand Name |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Null** | **Default** | **Description** |
| Brand Name | String (30) | Not nul |  | Brand’s name |

*4.11.3 Design database*

## 4.12 Add category screen

\

*4.12.1 Main flow*

|  |  |
| --- | --- |
| **Screen** | Add category interface |
| **Description** | Allow admin to add new category |
| **Screen access** | Admin enter new category’s informations |

*4.12.2 Validation Rule/Data mapping*

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Category name | Textbox | The ID of the received note Must be String  Insert to the field in Database: import. Category Name |

*4.12.3 Design database*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Null** | **Default** | **Description** |
| Category Name | String (30) | Not null |  | Brand’s name |

# PROJECT MANAGER PLAN

|  |  |  |  |
| --- | --- | --- | --- |
|  | **ID** | **Name** | **Task Percentage** |
| 1 | 519H0027 | Vo Nguyen Long | 50% |
| 2 | 519H0211 | Phan Le Minh Nhut | 50% |