**Centre Name: ACE-HCMC-2-FPT.**

**Address: 590 Cach Mang Thang 8, District 3, Ho Chi Minh City, Viet Nam.**.



**FPT**

**ACADEMY**

**INTERNATIONAL**

**FPT**

**–**

**APTECH**

**COMPUTER**

**EDUCATION**

# 

# PETSHOP

**Design Document**

|  |  |  |
| --- | --- | --- |
| **Supervisor:** | **Mr. Ngo Phuoc Nguyen** | |
| **Semester:** | **4** | |
| **Batch No:** | **T1.2011.E1** | |
| **Group No:** | **6** | |
| **Order:** | **Full name** | **Roll No.** |
| **1.** | **Le Minh Hieu** | **Student1136226** |
| **2.** | **Tran Duy Phu** | **Student1136505** |
| **3.** | **Pham Khoi Minh Huan** | **Student1140055** |
| **4.** | **Luong Hoang Luu** | **Student1140020** |

**REVIEW 1**

## 1. Introduction

In this day and age, with the growth of the Internet and the advancement of technology, buying and selling online is faster and easier than ever.

Many people browse through popular e-commerce stores and search engines to find whatever they are looking for and find the best price.

As a seller on an eCommerce website, you can let customers know about your store, recommend products, and take online sales orders. PetShop Shop is an online store, specializing in providing dog and cat breeds and pet care.

So they need a website to sell online, manage their business such as orders, sales, profits, ...

Consumers will enjoy the convenience of being able to choose to buy the breeds of dogs and cats they want through data on supply stores, or for those who don't have time to shop. It is also possible to book a pet care schedule in the store through the website. .

## 2. Existing Scenario

Today, online shopping is an important and potential market. After gathering enough information about the product and the store, online customers can decide to buy or not without going to the store.

In order to expand the company's market and target online customers, PetShop Shop approaches online customers in an attractive, reliable and complete way.

## 3. Customer Requirement Specification

**For customers**, they can:

* Filter products by categories, categories and prices.
* Sign up for a member account, log in to the system to rate stars for services, leave reviews, order online.

**For developers**: have full access to (create, read, delete, update) all information on the website..

## 4. Functional Requirement Specification

### 4.1 Guests (unregistered customers)

### 4.1.1 Description

Guests can use basic functions of the website such as:

* Search products, filter products by category.
* - Access cart, check order.

#### 4.1.2 Functional Requirement

When customers (include guest and member) access the home page, there should be:

+ Customers can search for products by name, system process and respond to the page to display results..

+ Customers can register for a member account

Website provides a page for guests to register for a member account by providing the necessary information: email (unique), password, name, mobile, address.

+ Customers can add products to the shopping cart while surfing the website to create orders.

### 4.2 Member

#### 4.2.1 Description

* Guests can register on the site to become a member. Once they have an account, they will still be able to perform all the functions as a guest, but they'll be able to access more features after signing in:
* After logging in, members can:
* Update their information including password changes.
* Manage orders
* Members have access to the comment section of the product for review after purchase.
* Star rating for the product.

#### 4.2.2 Require function

### - Customers can log into the system with account information: email, password. After members login

### - Member can modify his / her profile: change password, name, mobile, address.

### Rating for product: rating for product (from 1 to 5 stars), if rated, they can see their rating and can update the rating.

### - Leave a comment

### 4.3 Administration

#### 4.3.1 Description

Manage information related to products and customers, track customer orders, diagnose store business status through sales numbers and charts.

#### 4.3.2 Functional Requirement

Through the admin pages, after logging in with email and password, the administrator can:

- Manage product-related information, admin:

• Create

• Read

• Update

• Erase.

- Manage service-related information, administrators:

• Create

• Read

• Update

• Erase.

- Manage information related to orders, administrators:

• Create

• Read

• Update

• Erase.

- Manage information related to members, administrators:

• Create

• Read

• Update

• Erase.

There is a page to change the status of your order with values (listed based on actual orders): processing (default - 1), shipping (2) and completion (3).

## 5. System Requirements

**5.1Hardware Requirements**

**5.2 Software Requirements** - Win 7, Win 8 or Win 10. - SQL Server 2017 or higher. - Netbean 7.4 or higher. JavaEE 7 SDK.

* Java Development Kit (JDK) 1.8 – Server TomCat 8.5.40.
* Google Chrome70.0 or higher.

## 6. Technologies

* JAVA EE
* JSP, Servlet
* HTML
* CSS
* Javascript
* Jquery
* Bootstrap

## 7. Scope of project

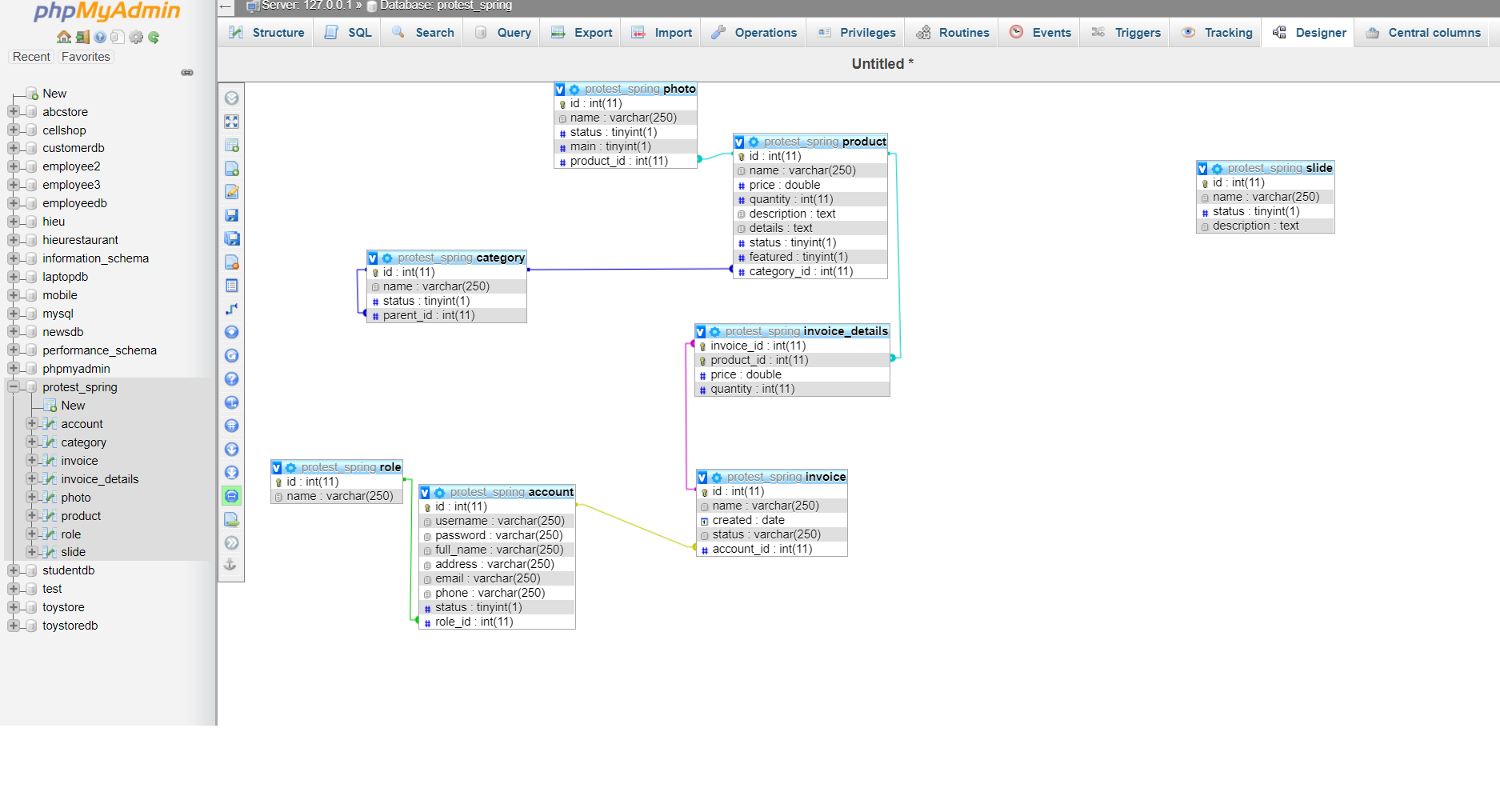
There are some functions not to be included in this project as follows:

* Manage the warehouse: all products with different sizes and colors will be always available.
* Manage promotions: No promotion will be apply.
* Manage the shipping.
* Manage payment methods via credit card.

## 8. Task sheet

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project** | | **Project Name:** |  | **Date of Preparation of Activity Plan** | | |
| **Ref. No.** | | **Tshop Computer** |  |  | | |
| No. | Task | Description | Start |  | Complete | Team member |
|  |  |  | Date |  | Date |  |
| 01 |  | Problem Definition |  |  |  | Le Minh Hieu  Pham Khoi Minh Huan |
| 02 | Customer Requirements |
|  | **Review 1** | Specifications | **24/11/**  **2020** |  | **08/12/2020** | Tran Duy Phu |
| 03 | Functional Requirement  Specification |
| 04 | System Requirements |
| 05 | Technologies |
| 06 | Scope of project |

\*Database Diagram :



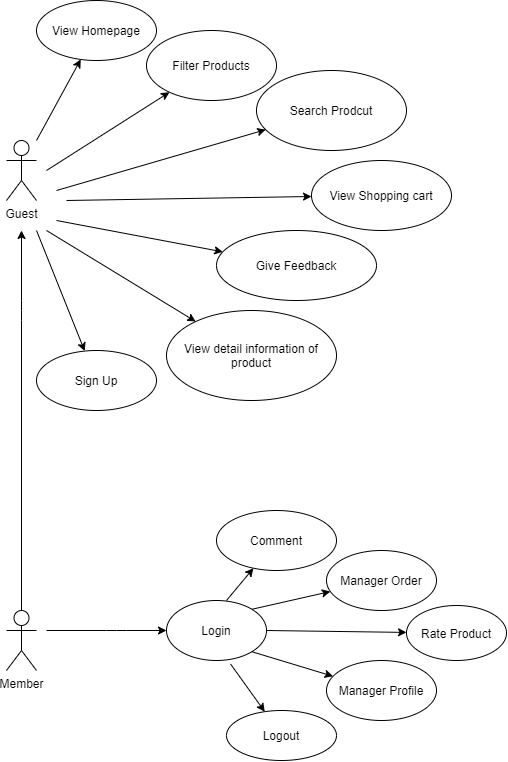
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Ref. No.:**  **eP/Advertisement**  **Portal Management**  **System/01** | | **Project Title:** | **Activity**  **Plan**  **Prepared By:** | **Date of Preparation of Activity Plan:** | | | |
| **Sr.No.** | **Task** | **Actual**  **Start**  **Date** | **Actual Days** | **Team**  **Mate**  **Name s** | **Status** |
| 1 | Database | S2 Pet Shop | 30/11/2020 | 08/12/2020 | 1 | Hieu | Completed |
| 2 | Login , Logout | 08/12/2020 | 3 | Phu | Completed |
| 3 | Shopping  Cart ,Check  Out | 10/12/2020 | 3 | Hieu | Completed |
| 4 | Search,Wishlist,Rating, Recommendation | 10/12/2020 | 3 | Phu | Completed |
| 5 | Design | 11/12/2020 | 3 | Huan | Completed |
| 6 | Admin Management | 11/12/2020 | 3 | Hieu, Phu | Completed |
| 7 | Database Design Diagram | 11/12/2020 | 2 | Hieu | Completed |
| 8 | GUI Design back- end | 14/12/2020 | 3 | Huan | Completed |
| 9 | Task sheet review | 15/12/2020 | 1 | Huan | Completed |
| 10 | staff decentralization |  |  |  |  | Luu |  |

**REVIEW 2**

**Entity – Relationship Diagrams Database**

**I.Use Case and Sequence Diagram:**

**Customer**

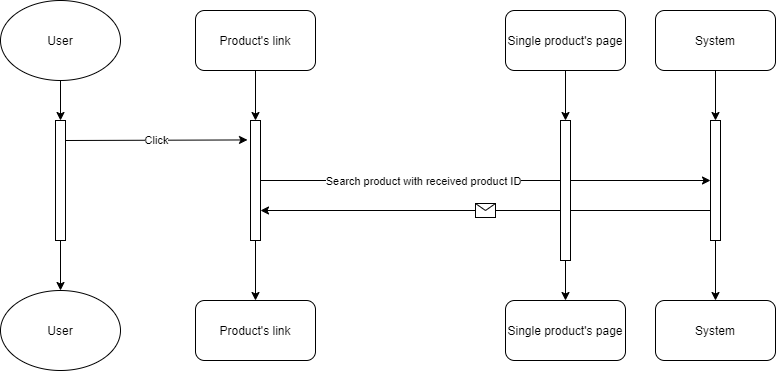


**Admin**

### 

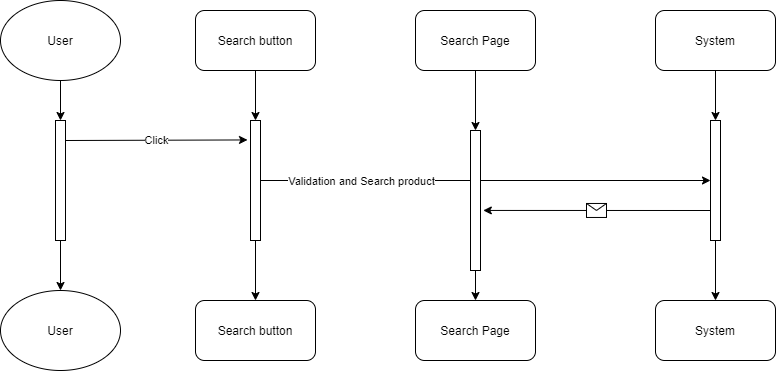
1. **Customer**
2. **Search**

|  |  |  |
| --- | --- | --- |
| Author | Tran Duy Phu | |
| Use Case Name | Search – Auto complete | |
| Actors | Guests and Members | |
| Description | Display products which have names contain string in search field | |
| Requirements | Search result appear in a panel right below the search field, which contains name and products’ image | |
| Pre-conditions |  | |
| Post-conditions |  | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Typing in search field | 1. Create an empty list to contain search result. 2. Search in Products table to find products which have names contain the search string (ignore case) and add them to result list. 3. Display the result list in a panel right below the search field. |
| Alternative flow |  |  |
| Exception |  |  |
|  |  |  |

****

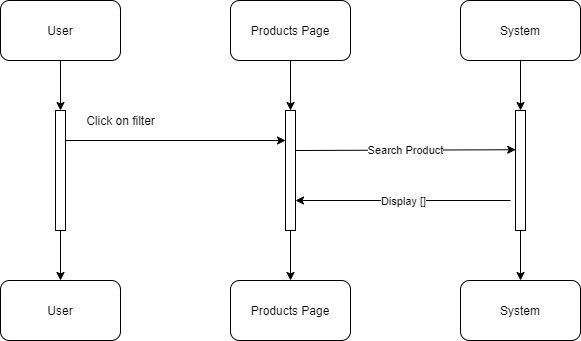
**a.1 Button**

|  |  |  |
| --- | --- | --- |
| Author | Tran Duy Phu | |
| Use Case Name | Search – Search button | |
| Actors | Guests and Members | |
| Description | When search button is clicked, search result page will be displayed with result list of products and their information. | |
| Requirements | Search string can not be blank. | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Actor input search string and click on search button. | 1. System check the search string [Exception 1] 2. System create an empty list to contain search result. 3. Search in Products table to find products which have names contain the search string (ignore case) and add them to result list.   4. Display the result list in the search result page. |
| Exception | 1. Actor left the search string empty | 1. An error message will be display:  “Search string can not be empty” |

****

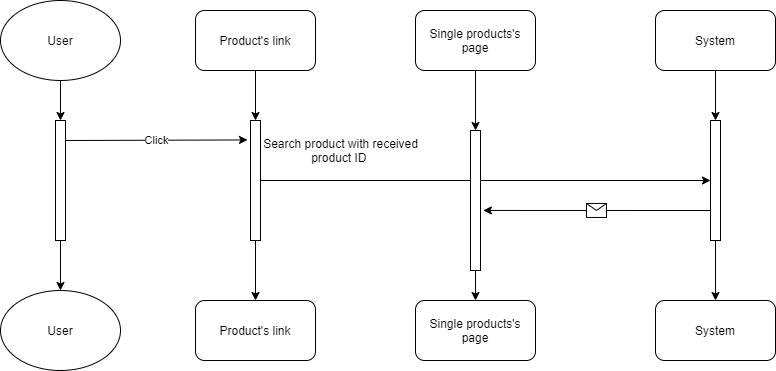
1. **Filter**

|  |  |  |
| --- | --- | --- |
| Author | Tran Duy Phu | |
| Use Case Name | Filter | |
| Actors | Guests and Members | |
| Description | Allows customers view all products which are in the same categories, brands. Products will be displayed with the pagination. | |
| Requirements | Links of Brands, Categories, Types must be clicked, or range of prices must be chosen. | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Clicks on links of Brands, Categories | 1. Receive parameters passing with the links and search products which have Brands or Categories with the parameters. 2. Display the list in the products page. |

****

1. **View Product Details**

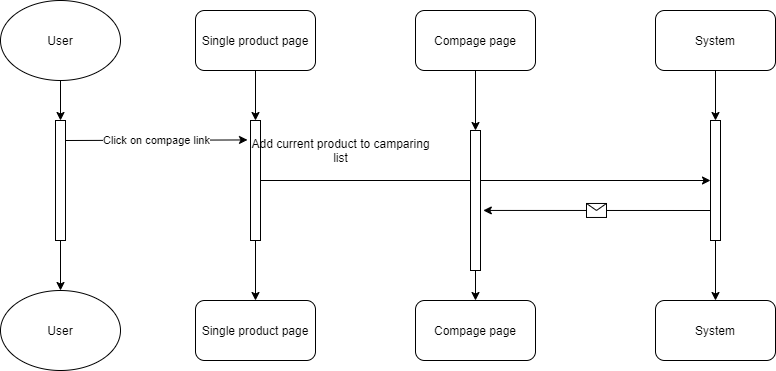
|  |  |  |
| --- | --- | --- |
| Author | Tran Duy Phu | |
| Use Case Name | View product details | |
| Actors | Guests and Members | |
| Description | When customers click on any products’ link, a page will display to show all products’ information. | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Clicks on products’ links | 1. Receive parameters passing with the links and search products which have productID satisfied with the parameters. 2. Display the details in the singleProduct page. |

****

1. **Compare**

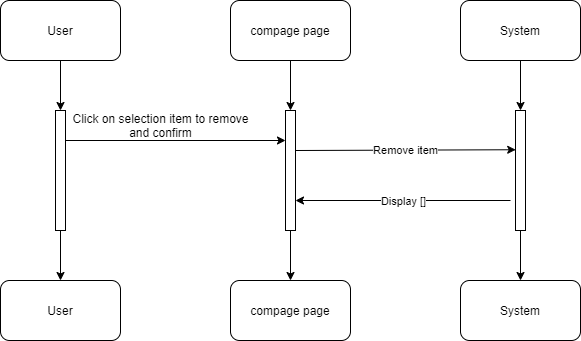
**d.1 Add:**

|  |  |  |
| --- | --- | --- |
| Author | Tran Duy Phu | |
| Use Case Name | Compare - Add | |
| Actors | Guests and Members | |
| Description | When a customer clicks on “Compare” link in singleProduct page, compare page will appear to display all compared products include added products. | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Clicks on “Compare” link in singleProduct page. | 2. Check the number of compared products:  + if it equals 5, system disable the compare link and display a text  “Reach the limit of 5 compared products”.  + else, system add the product to compared products and display compare page. |

****

**d.2 Remove:**

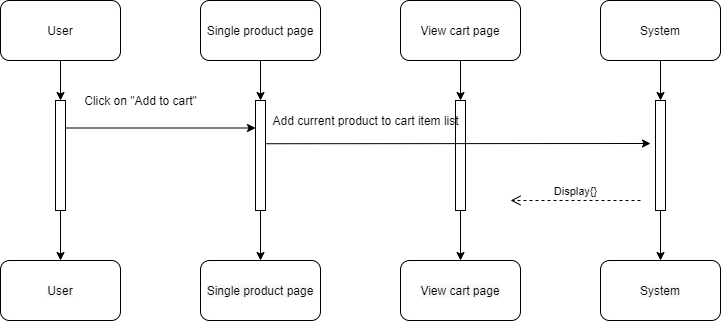
|  |  |  |
| --- | --- | --- |
| Author | Tran Duy Phu | |
| Use Case Name | Compare - Remove | |
| Actors | Guests and Members | |
| Description | When customer clicks on remove links and choose “Yes” to confirm, the selected product will be remove from the compared list. | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Clicks on remove link of compared product in the compare page.  3. Clicks on “Yes” to confirm. | 2. Display 2 buttons: “Yes” and “No” for customer confirm the removing.  4. Remove selected product from the compared list and display compare page with updated list. |
| Alternative flow | 5. Clicks on “No”. | 6. The 2 buttons disappear. |

****

**e.Cart**

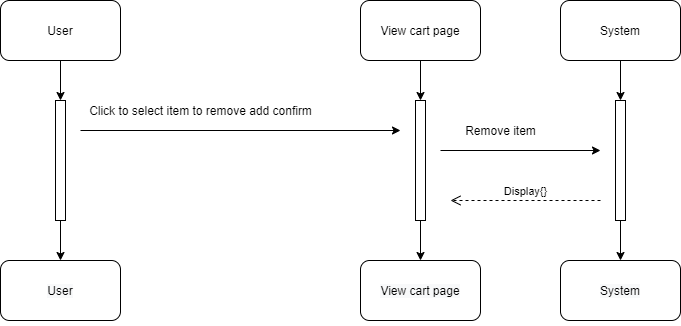
**e.1 Add**

|  |  |  |
| --- | --- | --- |
| Author | Le Minh Hieu | |
| Use Case Name | Cart – Add | |
| Actors | Guests and Members | |
| Description | When customers select the quantity and clicks on “Add to Cart” in singleProduct page, the current product will be added/updated into list of cart items, and cart page will be displayed with all cart items. | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Selects the quantity and clicks on “Add to Cart”. | 1. Check the added product:   + If existed in cart, system will add the quantity to the existed item’s quantity.  + Else, system will add the product with the selected quantity into the list of cart items.   1. Display the cart page with all updated cart items. |

****

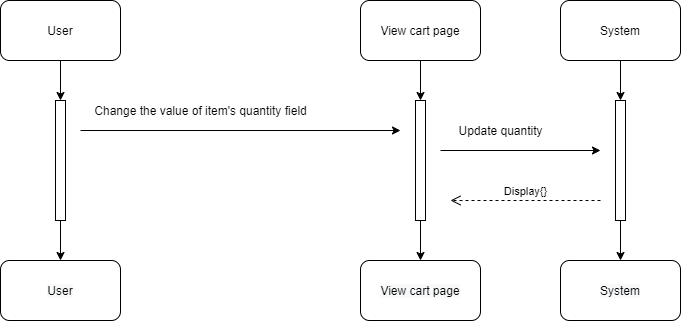
**e.2 Remove**

|  |  |  |
| --- | --- | --- |
| Author | Le Minh Hieu | |
| Use Case Name | Cart – Remove | |
| Actors | Guests and Members | |
| Description | When customers clicks on remove links and choose “Yes” to confirm, the selected product will be remove from the list of cart items. | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Clicks on remove link of cart items in the cart page.  3. Clicks on “Yes” to confirm. | 2. Display 2 buttons: “Yes” and “No” for customer confirm the removing.  4. Remove selected product from the list of cart items and display cart page with updated list. |
| Alternative flow | 5. Clicks on “No”. | 6. The 2 buttons disappear. |

****

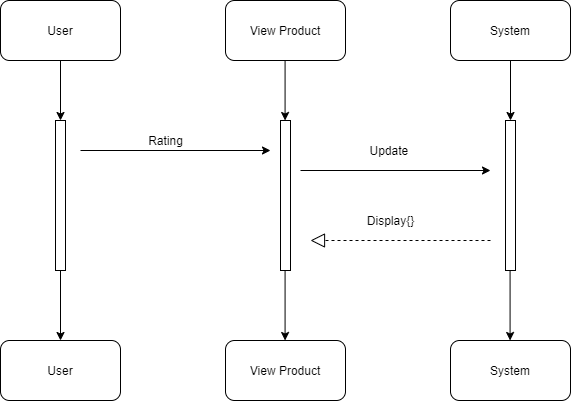
**e.3 Update quantity**

|  |  |  |
| --- | --- | --- |
| Author | Le Minh Hieu | |
| Use Case Name | Cart – Update quantity | |
| Actors | Guests and Members | |
| Description | When customers input quantity field of cart item, system will update the quantity, subtotal, total and display cart page with new information | |
| Requirements | Quantity must be greater than 0. | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Updates new number in the quantity field. | 1. Check the   new quantity [Exception 1]   1. Update new quantity, subtotal and total of cart items. 2. Display cart page with new information. |
| Exception | [Exception 1]   1. Actor input a number less than 0.      1. Actor input a string not in number format | * Display error message “Quantity must be greater than 0”. * Display error message   “Quantity is not in number format”. |

****

**F.Rating**

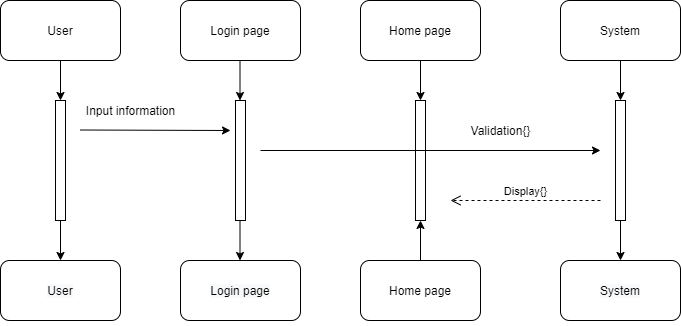
|  |  |  |
| --- | --- | --- |
| Author | Pham Khoi Minh Huan | |
| Use Case Name | Rating | |
| Actors | Guests and Members | |
| Description | Cart – Product reviews (Customers log in with their registered email and password.) | |
|  | Actor Actions | System Responses |
| Basic flow |  |  |
| Exception |  |  |

****

1. **Member**

**a.Login**

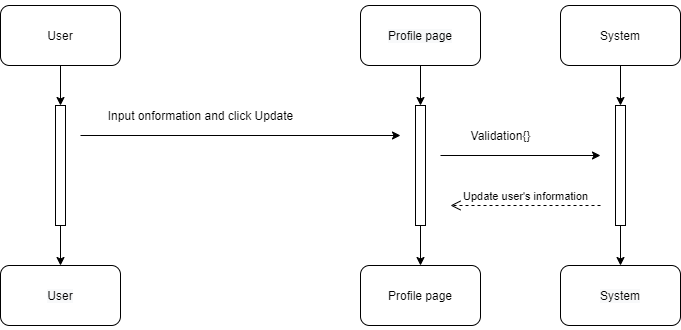
|  |  |  |
| --- | --- | --- |
| Author | Tran Duy Phu | |
| Use Case Name | Login | |
| Actors | Members | |
| Description | Customers log in with their registered email and password. | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Input email and password and clicks on  Submit button | 1. System check email and password [Exception 1] 2. System tracks the user’s information and display Home page |
| Exception | [Exception 1]   1. Actor left the Email blank 2. Actor left the Password blank 3. Actor inputs wrong email or password | Display error message:   * “Email can not be blank” - “Password can not be blank” * “Wrong email or Password” |

****

**b.Update private information**

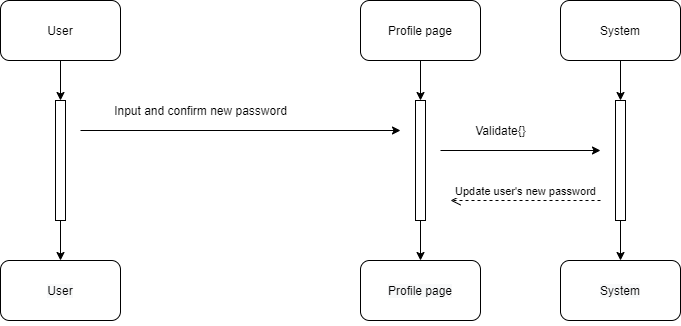
**b.1 Update information**

|  |  |  |
| --- | --- | --- |
| Author | Tran Duy Phu | |
| Use Case Name | Update information | |
| Actors | Members | |
| Description | Customers want to change their private information | |
| Requirements | Login as member | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Actors change Avatar, First Name, Last Name, Phone, Gender. | 1. System   check information  [Exception 1]   1. Update new information to data in Customers table. |
| Exception | [Exception 1]   1. Actor left First Name blank 2. Actor left Last Name blank 3. Actor left Phone Number blank 4. Actor left Gender blank | Display error message:   1. “First Name can not be blank” 2. “Last Name can not be blank” 3. “Phone Number can not be blank” 4. “Gender can not be blank” |

****

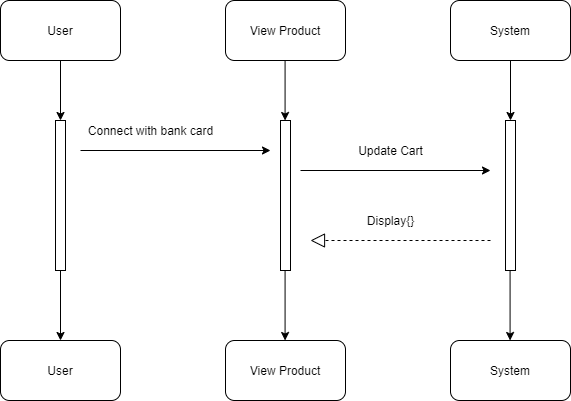
**b.2 Change password**

|  |  |  |
| --- | --- | --- |
| Author | Tran Duy Phu | |
| Use Case Name | Update information – Change password | |
| Actors | Members | |
| Description | Customer want to change the registered password | |
| Requirements | Login as members | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Input Old Password, New Password and Confirm Password | 1. Check   information [Exception 1]   1. Update New Password into Customers table |
| Exception | [Exception 1]   1. Old Password is wrong. 2. New Password length is less than 3      1. Confirm Password does not match with New Password | Display error message:   * “Old Password is wrong” * “New Password length is less than 3” * “Confirm Password does not match with New Password” |

****

**C.Online payment**

|  |  |  |
| --- | --- | --- |
| Author | Tran Duy Phu | |
| Use Case Name | Payment | |
| Actors | Members | |
| Description | Customer want to change the registered password | |
| Requirements | Login as members | |
|  | Actor Actions | System Responses |
| Basic flow | Bank account card connection | Check account connection |
| Exception |  |  |

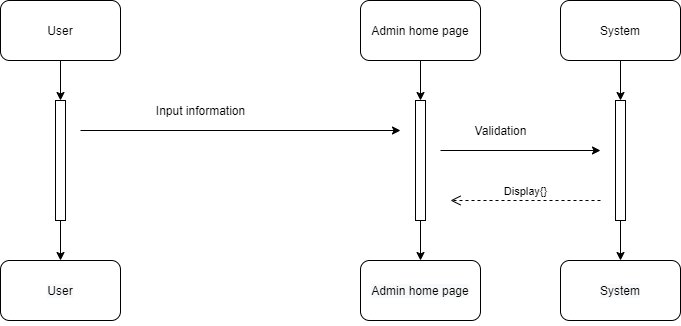
****

1. **Admin**

**a.Update account information**

**a.1 Update information**

|  |  |  |
| --- | --- | --- |
| Author | Tran Duy Phu | |
| Use Case Name | Update information | |
| Actors | Admins | |
| Description | Admins want to change their private information | |
| Requirements | Login as admin | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Change Full Name | 1. System   check information  [Exception 1]   1. Update new information to data in Admins table. |
| Exception | [Exception 1]  1. Actor left Full Name blank | Display error message:  1. “Full Name can not be blank” |

****

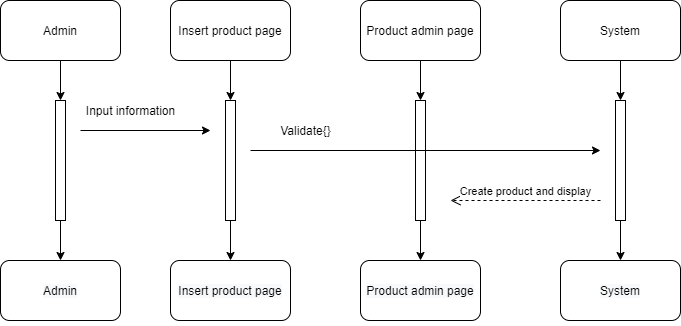
**a.2 Change password**

|  |  |  |
| --- | --- | --- |
| Actors | Tran Duy Phu | |
| Description | Admins want to change the registered password | |
| Requirements | Login as Admins | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Input Old Password, New Password and Confirm Password | 1. Check   information [Exception 1]   1. Update New Password into Admins table |
| Exception | [Exception 1]   1. Old Password is wrong. 2. New Password length is less than 3      1. Confirm Password does not match with New Password | Display error message:   * “Old Password is wrong” * “New Password length is less than 3” * “Confirm Password does not match with New Password” |

**b.Manage products**

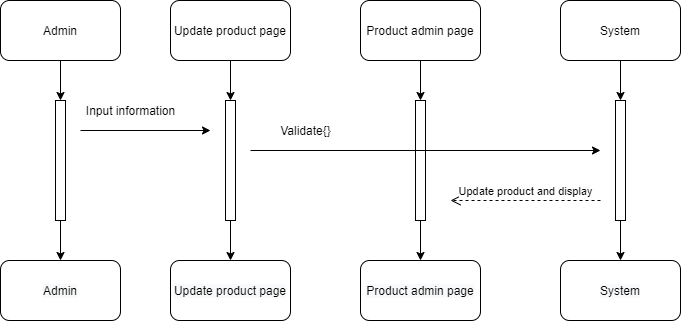
**b.1 Insert**

|  |  |  |
| --- | --- | --- |
| Author | Pham Khoi Minh Huan | |
| Use Case Name | Manage Products – Insert | |
| Actors | Admins | |
| Description | Admins want to insert new products | |
| Requirements | Login as Admins | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Input information to input fields: productID,ProductName,Price, Category, Brand, Images, Description and proStatus,proDate. Then clicks on Insert button. | 1. System check   the information [Exception 1]   1. Create new product in database 2. Display ProductAdmin page. |
| Exception | [Exception 1]   1. Actor left Product Name blank 2. Actor left Price blank 3. Actor left Description blank 4. Actor left Image | Display error message:   * “Product Name can not be blank” * “Price can not be blank” * “Description can not be blank”   - |

****

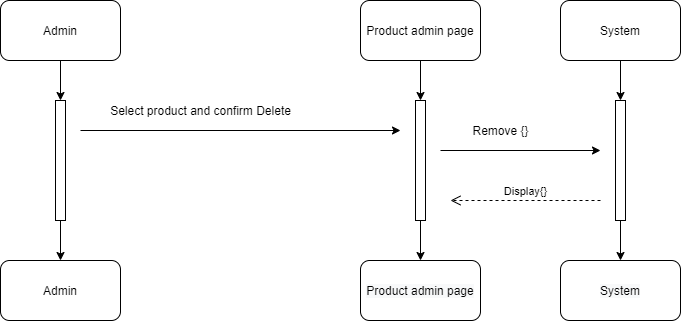
**b.2 Update**

|  |  |  |
| --- | --- | --- |
| Author | Pham Khoi Minh Huan | |
| Use Case Name | Manage Products – Update | |
| Actors | Admins | |
| Description | Admins want to update existed products’ information | |
| Requirements | Login as Admins | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Input information to input fields: productID,ProductName,Price, Category, Brand, Images, Description and proStatus,proDate. Then clicks on Update button.. | 1. System check   the information [Exception 1]   1. Update information in database 2. Display ProductAdmin page. |
| Exception | [Exception 1]   1. Actor left Product Name blank 2. Actor left Price blank 3. Actor left Description blank | Display error message:   * “Product Name can not be blank” * “Price can not be blank” * “Description can not be blank” |

****

**b.3 Delete**

|  |  |  |
| --- | --- | --- |
| Author | Pham Khoi Minh Huan |  |
| Use Case Name | Manage Products – Delete |  |
| Actors | Admins |  |
| Description | Admins want to disable existed products |  |
| Requirements | Login as Admins |  |
|  | Actor Actions | System Responses |
| Basic flow | 1. Clicks on Remove button  3. Clicks on “Yes” to confirm deleting    5. Clicks on “No” to cancel deleting | 2. Display 2 buttons: “Yes” and “No” 4. Change status value of chosen product to from Products table.   1. Hide the 2 buttons. 2. Display ProductAdmin page |

****

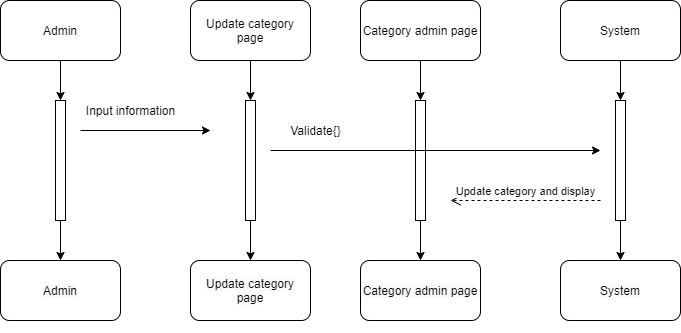
**c.Manage categories**

**c.1 Insert**

|  |  |  |
| --- | --- | --- |
| Author | Pham Khoi Minh Huan |  |
| Use Case Name | Manage Categories – Insert |  |
| Actors | Admins |  |
| Description | Admins want to insert new categories |  |
| Requirements | Login as Admins |  |
|  | Actor Actions | System Responses |
| Basic flow | 1. Input information to input fields: Category Name and Image. Then clicks on Insert button. | 2. System check  the information [Exception 1] 3. Create new category in database 4. Display Categories page. |
| Exception | [Exception 1]  1. Actor left Category Name blank | Display error message:  - “Category Name can not be blank” |

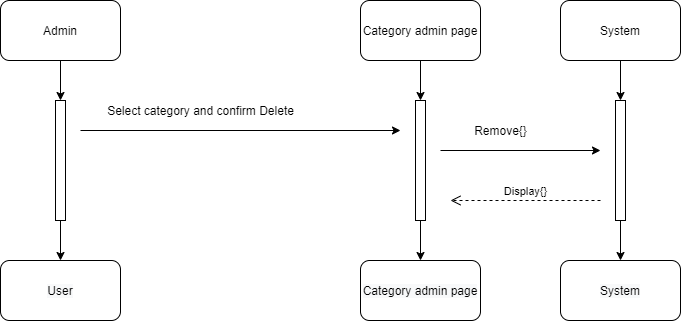
**c.2 Update**

|  |  |  |
| --- | --- | --- |
| Author | Pham Khoi Minh Huan | |
| Use Case Name | Manage Categories – Update | |
| Actors | Admins | |
| Description | Admins want to update existed categories’ information | |
| Requirements | Login as Admins | |
|  | Actor Actions | System Responses |
| Basic flow | 1. Input information to input fields: Category Name and Image. Then clicks on Update button. | 2. System check  the information [Exception 1] 3. Update existed category in database  4. Display Categories page. |
| Exception | [Exception 1]  1. Actor left Category Name blank | Display error message:  - “Category Name can not be blank” |

****

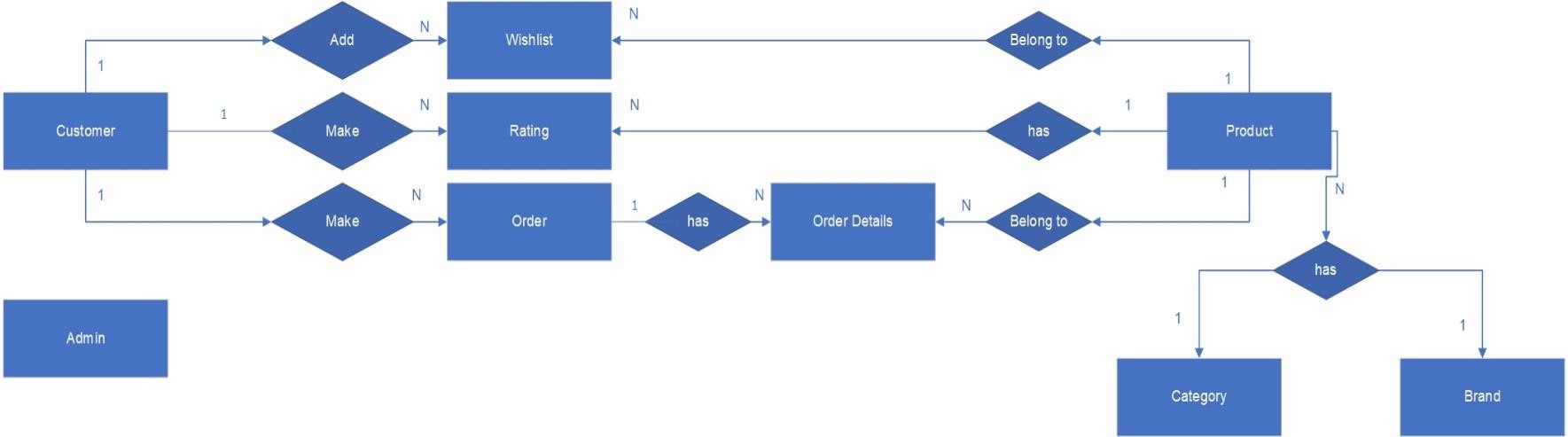
**C.3 Delete**

|  |  |  |
| --- | --- | --- |
| Author | Pham Khoi Minh Huan |  |
| Use Case Name | Manage Categories – Delete |  |
| Actors | Admins |  |
| Description | Admins want to delete existed categories |  |
| Requirements | Login as Admins |  |
|  | Actor Actions | System Responses |
| Basic flow | 1. Clicks on Delete button  3. Clicks on “Yes” to confirm deleting    5. Clicks on “No” to cancel deleting | 2. Display 2 buttons: “Yes” and “No” 4. Change status value of chosen product to from Categories table.   1. Hide the 2 buttons. 2. Display Categories page |

****

### II. Entity – Relationship Diagram (ERD):

**1.Logical**



**2.Tables:**

2.1Admin

2.2 Member

2.3 Guest

2.4 Product

2.5 Categories

2.6 Brands

2.7 Wishlist

2.8 Ratings

2.9 Orders

2.10 Order Details

**III. Task sheet:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Ref No. | | Project Name: | Date of Preparation of activity plan | | |
| No. | Task | Description | Start Date | Complete Date | Team Member |
| 01 | Review 2 | CART | 08/12/2020 | 15/12/2020 | Le Minh Hieu |
| 02 | COMPARE |
| 03 | CHECKOUT |
| 04 | MANAGE ORDERS |
| 05 | ORDER |
|  | ORDER DETAILS (INVOICE) |
| 06 | REPORT TOP PRODUCT SOLD |
| 07 | WISH LIST | Pham Khoi Minh Huan |
|  | RESET PASSWORD CUSTOMER |
| 08 | WISH LIST |
| 09 | VIEW PRODUCT DETAIL |
|  | RECOMMENDATION PRODUCT |
| 10 | REPORT TOP WISHLIST OF PRODUCT |
| 11 | MANAGE PRODUCTS |
|  | MANAGE CATEGORIES  UPDATE CATEGORY  UPDATE PRODUCT |
| 12 | SIGN UP | Tran Duy Phu |
| 13 | LOG IN |
| 14 | LOG OUT |
| 15 | LOGIN WITH GOOGLE ACCOUNT |
|  | UPDATE INFORMATION ACCOUNT |
| 16 | Rating |  |
| 17 | Banner Slide |
|  | Update Banner Slide |

**REVIEW 3**

1. **Table Definition** 
   * 1. **Customers:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Null** | **Default** | **Table-**  **Field-Ref** | **Description** |
| PK | Email | varchar(100) | No |  |  | Customers’ emails |
|  | Password | varchar(20) | No |  |  | Password to log in as member |
|  | FirstName | nvarchar(50) | No |  |  | Customers’ first names |
|  | LastName | nvarchar(50) | No |  |  | Customers’ last names |
|  | Gender | bit | No | 1 |  | Customers’ genders |
|  | Phone | varchar(20) | Yes |  |  | Customers’ phones |
|  | CreatedDate | Date | No | getDate() |  | Date which account is created |

* + 1. **Categories:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Null** | **Default** | **Table-**  **Field-Ref** | **Description** |
| PK | CategoryID | varchar(10) | No |  |  | Categories’ IDs |
|  | CategoryName | nvarchar(100) | No |  |  | Categories’ names |
|  | CategoryImage | varchar(max) | No |  |  | Categories’ images |

* + 1. **Brands:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Null** | **Default** | **Table-**  **Field-Ref** | **Description** |
| PK | BrandID | varchar(10) | No |  |  | Brands’ ID |
|  | BrandName | nvarchar(100) | No |  |  | Brands’ names |
|  | BrandImages | varchar(max) | Yes |  |  | Brands’ images |
|  | Descriptions | nvarchar(max) | Yes |  |  | Brands’ descriptions |

* + 1. **Products:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Null** | **Default** | **Table-Field-Ref** | **Description** |
| PK | ProductID | varchar(10) | No |  |  | Products’ ID |
|  | ProductName | nvarchar(100) | No |  |  | Products’ names |
|  | Price | float | No |  |  | Products’ prices |
| FK | CategoryID | varchar(10) | No |  | Categories (CategoryID) | Products’ categories |
| FK | BrandID | varchar(10) | No |  | Brands (BrandID) | Products’ brands |
|  | ProductImages | varchar(max) | No |  |  | Products’ images |
|  | Descriptions | nvarchar(max) | Yes |  |  | Products’ descriptions |
|  | proDate | date | Yes |  |  | Products’ date create |
|  | proStatus | int | No | 1 |  | Status of products’ activations |

* + 1. **Wishlist:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Null** | **Default** | **Table-Field-Ref** | **Description** |
| PK | WishlistID | int identity(1,1) | No |  |  | Wishlist’s ID |
| FK | CustomerEmail | varchar(100) | No |  | Customers (Email) | Customers’ emails |
| FK | ProductID | varchar(10) | No |  | Products(ProductID) | Added products’ IDs |

* + 1. **Ratings:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Null** | **Default** | **Table-Field-Ref** | **Description** |
| PK | RatingID | int identity(1,1) | No |  |  | Ratings’ IDs |
| FK | CustomerEmail | varchar(100) | No |  | Customers (Email) | Customers’ emails |
| FK | ProductID | varchar(10) | No |  | Products(ProductID) | Rated products’ IDs |
|  | Rate | int | No |  |  | Ratings’ values |
|  | RatingDate | date | No | getdate() |  | Rating dates |
|  | content | Varchar(100) | No |  |  | Content of  Customer’s rating |

**7. Orders:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Null** | **Default** | **Table-Field-Ref** | **Description** |
| PK | OrderID | varchar(10) | No |  |  | Orders’ IDs |
| FK | CustomerEmail | varchar(100) | No |  | Customers (Email) | Customers’ emails |
|  | Total | decimal(18,2) | No |  |  | Total payment |
|  | ShipAddress | nvarchar(200) | No |  |  | Receivers’s address |
|  | ShipDate | date | No |  |  | Shipping date |
|  | ShipNote | nvarchar(max) | Yes |  |  | Note for shipping |
|  | OrderDate | date | No | getdate() |  | Date order was made |
|  | PaymentMethod | nvarchar(100) | No |  |  | Payment method |
|  | ProcessStatus | nvarchar(50) | No |  |  | Orders’ status |
|  | OrderState | int | No | 1 |  | Status of orders’ activations |

* + 1. **OrdersDetails:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Null** | **Default** | **Table-Field-Ref** | **Description** |
| PK | OdID | int identity(1,1) | No |  |  | Orders details’ IDs |
| FK | OrderID | varchar(10) | No |  | Orders (OrderID) | Orders’ IDs |
| FK | ProductID | varchar(10) | No |  | Products(ProductID) | Products’ IDs |
|  | SellPrice | float | No |  |  | Products’ selling prices |
|  | Quantity | int | No |  |  | Products’ selling quantities |
|  | OdState | int | No | 1 |  | Status of orders details’ activations |

* + 1. **Admins:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Key** | **Field Name** | **Data Type** | **Null** | **Default** | **Table-**  **Field-Ref** | **Description** |
| PK | Email | varchar(100) | No |  |  | Admins’ emails |
|  | Password | varchar(20) | No |  |  | Password to log in as admins |
|  | FullName | nvarchar(100) | No |  |  | Admins’ full names |
|  | CreatedDate | Date | No | getDate() |  | Date account is created |

1. **Customers GUI Design**

Template - Home - Products - Contact

