**MINISTRY OF EDUCATION AND TRAINING**

**UNIVERSITY OF TECHNOLOGY AND EDUCATION**

# **FACULTY OF INFORMATION TECHNOLOGY**

# **--------------------**

**Web Programing**

**ONLINE SHOP**

**STUDENT LIST:**

1. **NGUYỄN MẠNH TIẾN 17145370**
2. **LÊ TUẤN ĐẠT 17110019**
3. **NGUYỄN HOÀNG TRƯỜNG MINH**

**LECTURER: Nguyễn Đức Khoan**

# **Ho Chi Minh City** **– 2019**

**MINISTRY OF EDUCATION AND TRAINING**

**UNIVERSITY OF TECHNOLOGY AND EDUCATION**

# **FACULTY OF INFORMATION TECHNOLOGY**

# **--------------------**

**Web Programing**

**ONLINE SHOP**

**STUDENT LIST:**

1. **NGUYỄN MẠNH TIẾN 17145370**
2. **LÊ TUẤN ĐẠT 17110019**
3. **NGUYỄN HOÀNG TRƯỜNG MINH**

**LECTURER: Nguyễn Đức Khoan**

# **Ho Chi Minh City** **– 2019**

**Lecturer's evaluation**

*Ho Chi Minh City, Jun, 2019*

**Lecturer**

# **Thanks**

The success of a student more or less always accompanies a lecturer. We would like to express our sincere thanks to Mr. Nguyen Duc Khoan, who directly supported the group, give us suggestions, comments and suggestions as well as provide tips to help us make the best project. Thanks to his instructions, help our team understand the knowledge to do, the presentation as well as the implementation of the project so we have completed the schedule with a lot of experience that we learn. Again, our team would like to thank the lecturer.

Projects are made within 8 weeks, just enough to complete it. However, due to many new knowledges as well as the time we do through each week is not optimal, the project will have some errors, that is inevitable.

We are looking forward to receiving all the comments of our teachers to help our limited knowledge better. Sincerely thanks.

# **Preface**

The purpose and objective of this training and particularly the content is really time-being and with this training we have gained some confidence regarding to introduce the application. We also belief that way we gained some sorts of IT knowledge and if we practice much and having some expertise in the field then we will be able to survive smartly in today’s competitive environment.

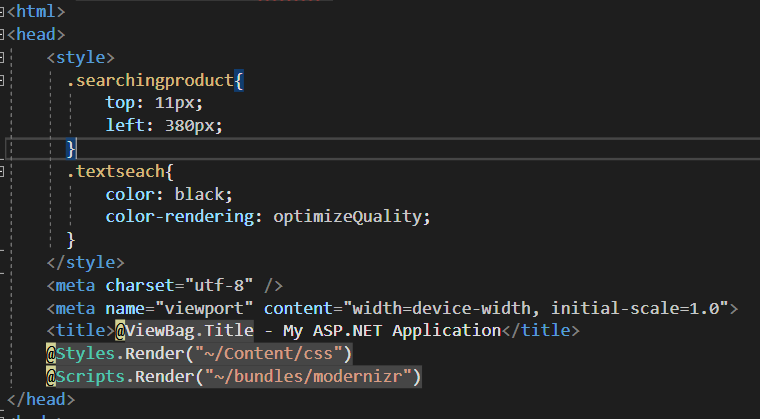
The effort to write the report is a partial fulfillment to complete the course. In the report I try my best to represent all the content which we learnt in a great deal in the program in a systematic and presentable order. I divided each of the topics as an individual chapter to reflect the entire topic more prominently and clearly. In the reference I have used citation method in the entire report. Finally, I am very hopeful that the structure and topic of the report will be a useful print material to all the reader especially to the user.

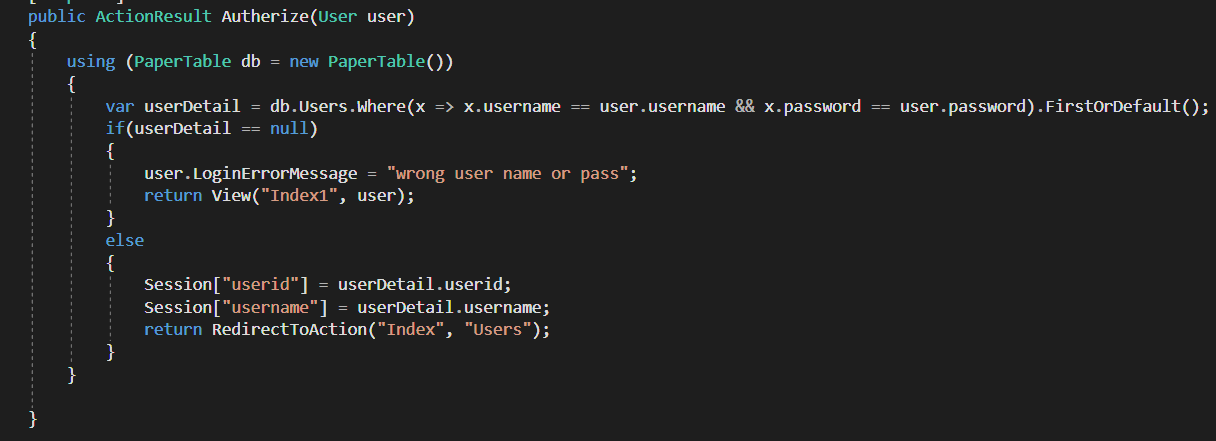
**Chapter 1: Basic Knowledge**

* 1. **Web design language**
     1. **Html**
* HTML (Hyper Text Markup Language), also known as hypertext markup language, was created to structure a website with pieces of information presented on the World Wide Web.

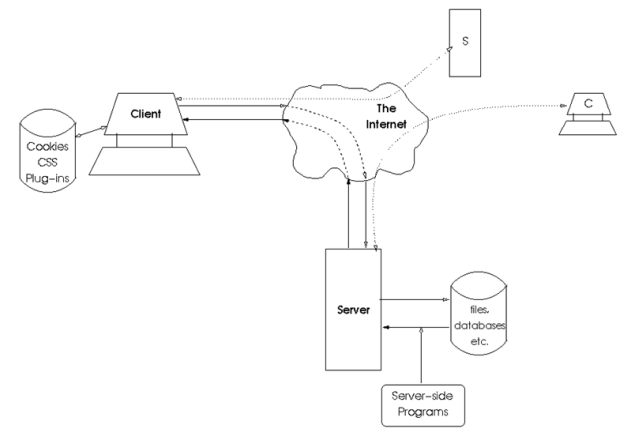
|  |
| --- |
| <html>  <head>  <title>Hello world </title>  </head> <body>  <p> Hello world! </p> </body> </html> |

* A web page starts with an <html> opening tag and ends with a closing </html> tag. Which includes two main content: header information is declared in <head> </head> tag, website content is declared in <body> </body> tag.
  + 1. **CSS**
* CSS (Cascading Style Sheet). The website is made up of html tags but with html tags only shows the frame of the website. To align and present beautifully, we need to use the CSS language. This is a language used a lot in web programming, often associated with html language.
* We have 3 ways to insert CSS into the HTML page
* Insert CSS content into the <style> </style> tag in the <head </head> section of the webpage.
* Insert directly inside the HTML tag.
* Link to an external .css file.
* In fact, the third way is the most used by programmers due to its usability and flexibility.
* Example I use the first insert CSS into products View

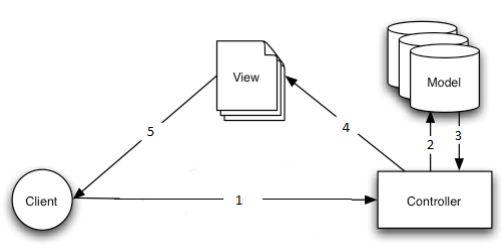


* + 1. **C#**
* C# (C-Sharp) is a programming language developed by Microsoft that runs on the .NET Framework.
* C# is used to develop web apps, desktop apps, mobile apps, games and much more.
* Example in LoginController
  1. **Architecture**

**1.2.1 Client/Sever**

****

Client / Server architecture is a famous architecture in computer networks, most websites operate based on this architecture. Where the Client is the client that sends the request to the Server. Here, the Server listens for requests from the Client, receives information from the Client, then processes and returns the results to the Client.

**1.2.2 Model MVC**

MVC model (Model - View - Controller) is one of the design models used in software development techniques, helping developers to split the application into three parts - Model - View - Controller. Each component has a separate task and function, help developing applications fast, easy to maintain, and upgrade the system

* **Model:** The component contains all the logic business, processing methods, database access, data description objects, relational constraints ...
* **View:** Ensuring the display of information, interaction with users, which contains all GUI objects such as textboxes, images ...
* **Controller:** Is the most important part in MVC model. Play the role of receiving requests from the client, processing information and then returning information to the client.

**How the MVC model works**

* When a client machine interacts with a view (using browsers on PC or Mobile). Send a request to the server. The controller will receive the request, process the request, if the information is related to the database, the controller calls the model to retrieve data. Then return the result after processing to View. The view displays information as HTML tags to the user.

**Chapter 2: System Analysis**

**2.1 Function of the website**

**2.1.1 Function for user/guest**

|  |  |  |
| --- | --- | --- |
| STT | Function | Description |
| 1 | View | User can view colors, products, categories, order and users table. |
| 2 | Order | User can order products on website and click order the product user want. |
| 3 | Cart | allow users to edit their orders, update the number or order new product |

**2.1.2 Detail of function for user**

1/ View

User login the username and password and choose what type user want to see

* Input: The page that user click
* Output: The view of the page that user want to see

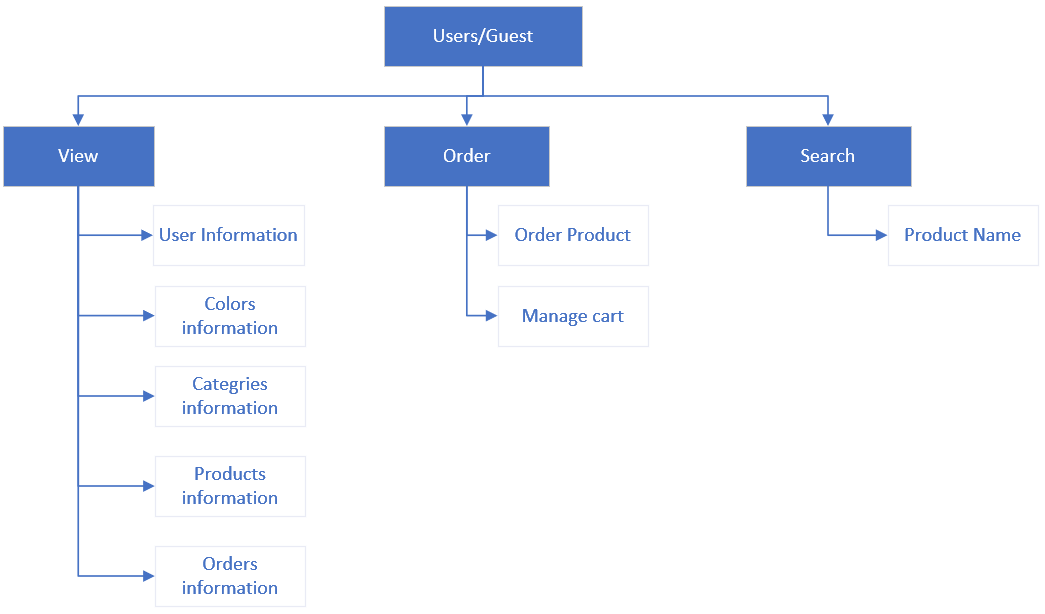
2/ Order

* Input: Information about the product that customers want to buy
* Output: The information of the item is in the cart

3/ Cart

* Input: Information of the items that customers have ordered, quantity products that customers want to buy
* Output: Cart information including product, quantity, total price ...

**2.1.3 Function Diagram for user**

****

**2.1.4 Function for admin**

|  |  |  |
| --- | --- | --- |
|  | Function | Description |
| 1 | Product Management | Information management of product images, administrators can upload product images, enter feature description information product, price, quantity |
| 2 | Categories management | Information management of category each product |
| 3 | Customer management | Add, edit, delete customer information |
| 4 | Order Detail | Display order details including information  About customers, information about the products in the order, order status, administrator notes |

**2.1.5 Detail of function for admin**

1/ Products management

* Input: Information of products name, price…… (add, edit, delete each product)
* Output: Information of each product after modify it.

2/ Categories management

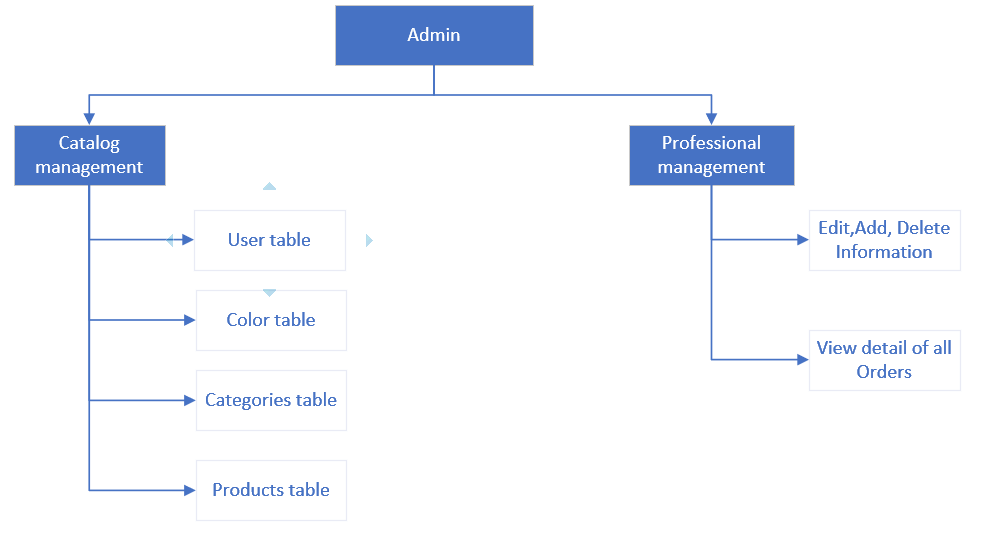
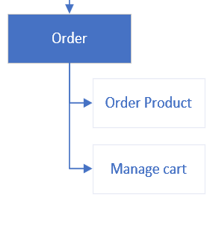
* Input: Information of categories (add, edit, delete each category)
* Output: Information of category after modify it

3/Customer management

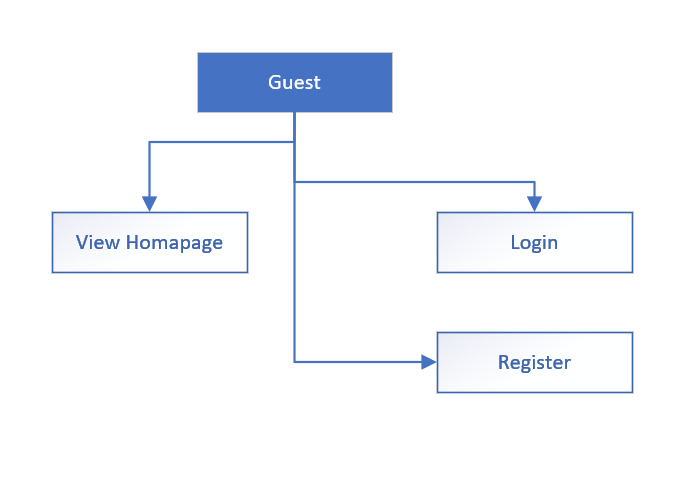
* Input: Information of users (username, full name, address, …)
* Output: Information of users after modify it

4/Order detail

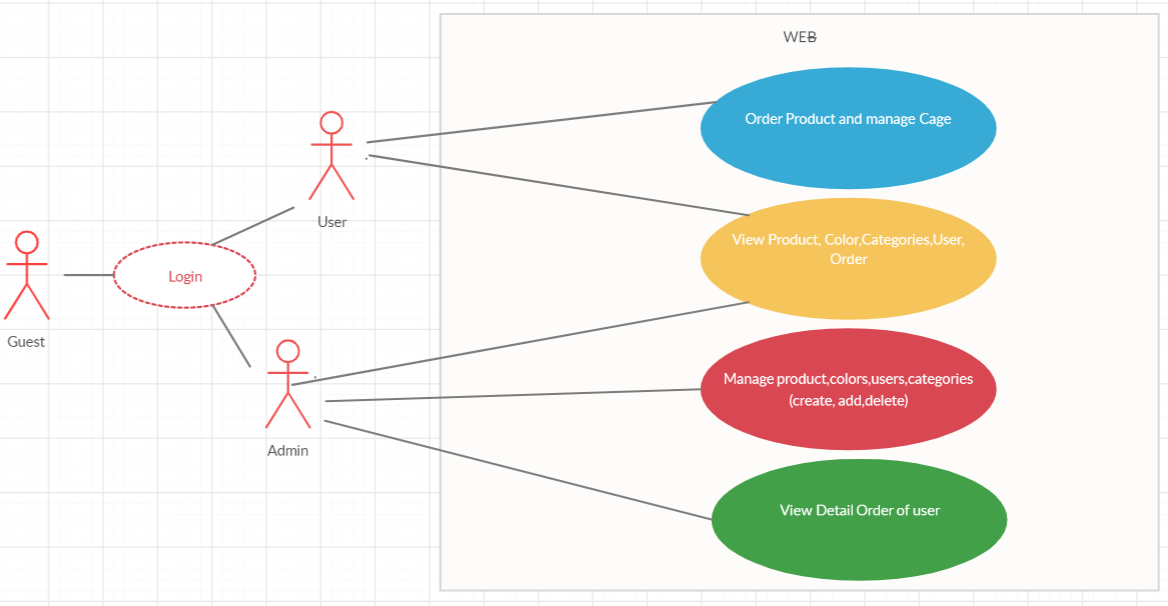
* Input: The information of order (address, number of products the user wants to buy)
* Output: The information of order

**2.1.6 Function Diagram for admin**

**2.1.7 Function Diagram for Guest**

****

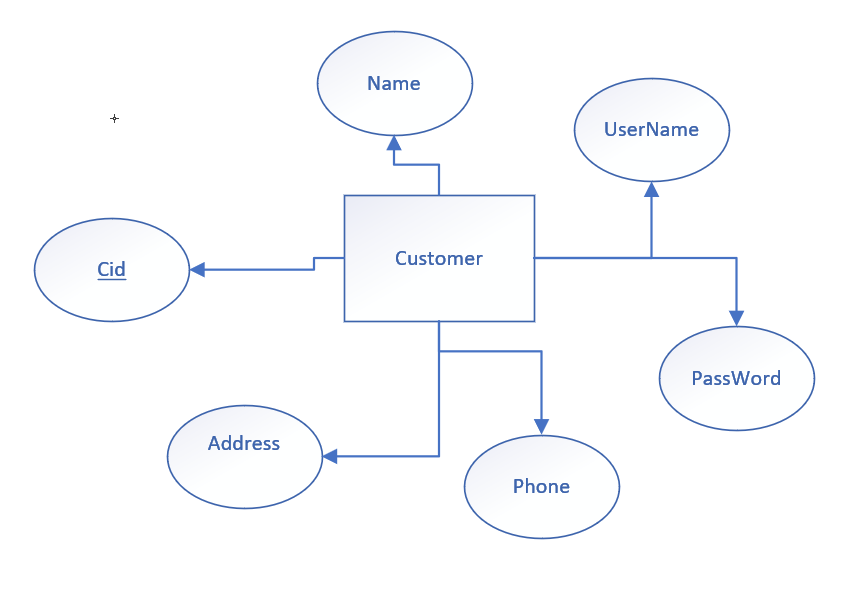
**2.1.8 Use case Diagram**



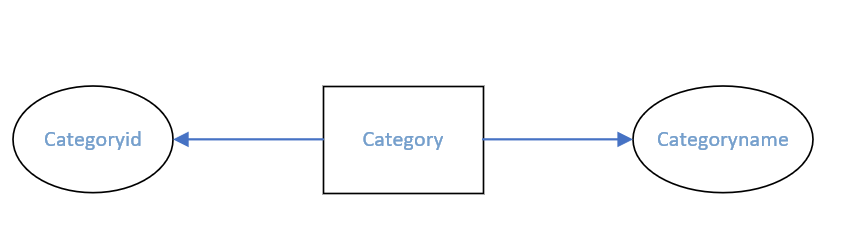
**Chapter 3: Database Design**

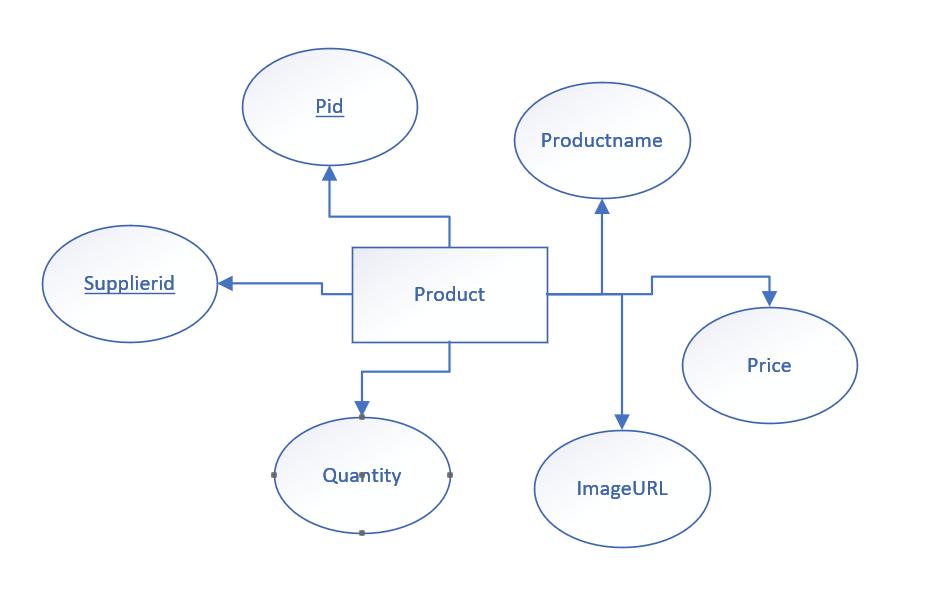
**3.1. Entity–relationship model**

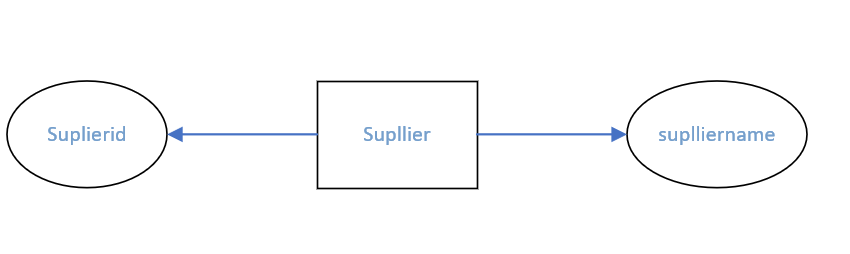
**3.1.1 List Entity**

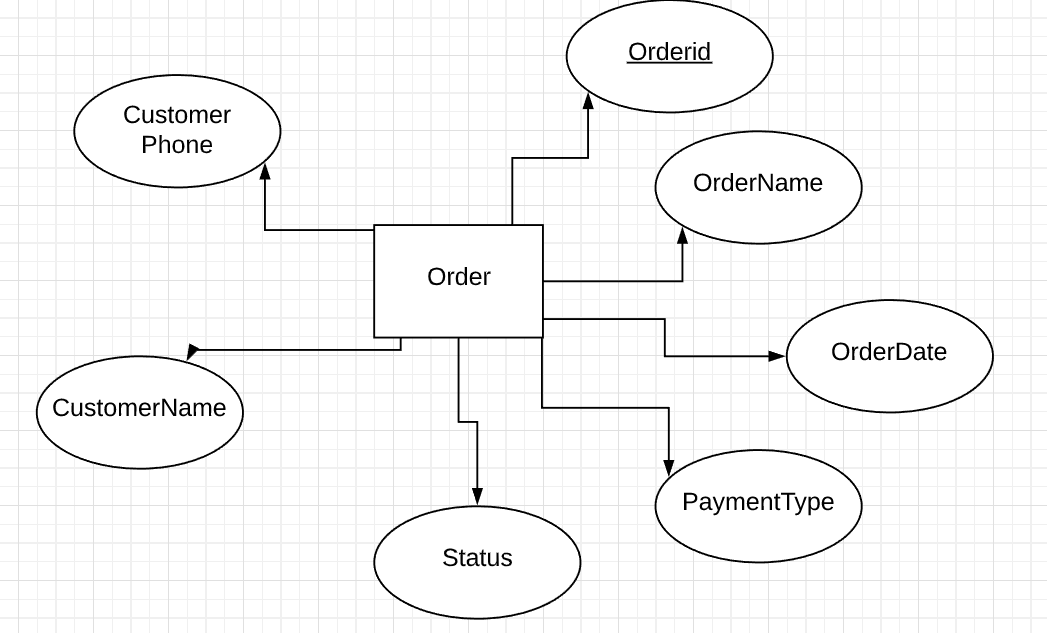
a/ Customer entity

b/Category entity

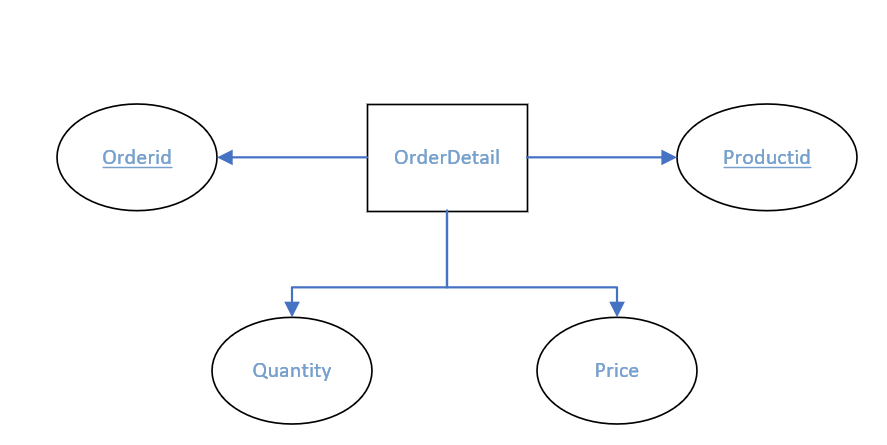


c/Product entity

e/Staff entity

f/Order entity

g/ OrderDetail entity



**3.1.2 Entity Relationship model**

Cart

Search

Products

User

**n**

**1**

Create

Customer

**1**

**N 1**

Have

**1 1**

Send

Buy

**1 n**

**1**

Depart Store

**n**

**1**

View Order

Include

Order

**11 n**

**n 1**

Selling products

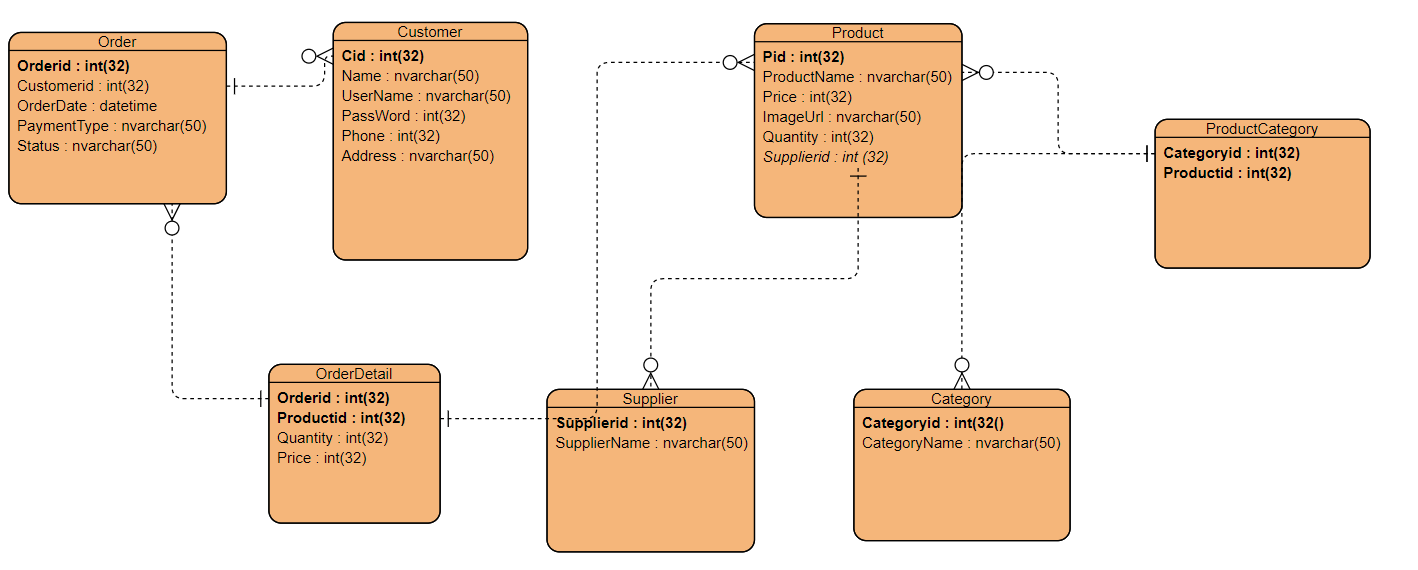
**n**

Admin

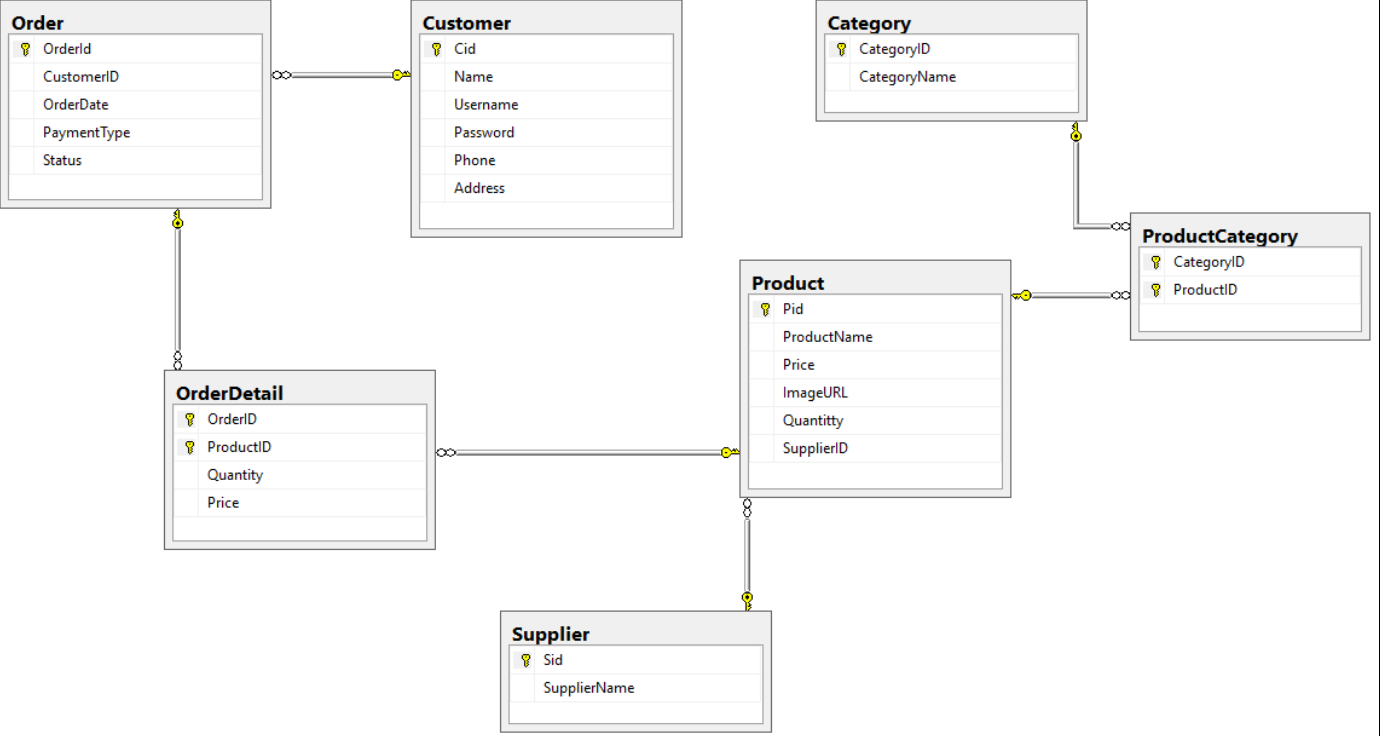
**3.2 Design database**

- Database Table

|  |  |  |
| --- | --- | --- |
| STT | TABLE NAME | SUBSTANCE |
| 1 | Customer | Customer table info |
| 2 | Supplier | Detail Supplier |
| 3 | Order Detail | Detail orders table of each customer |
| 4 | Order | Order Table of each customer |
| 5 | Category | Category of product |
| 6 | Products | Detail content of each product |
| 7 | ProductCategory | Connect 2 table Product and category |

**3.2.1 ERD**

**3.2.2 Connect the tables in the database**

****

**3.2.3. Design Table**

Order Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Datafield name | explain | Datatype | Key |
| 1 | OrderId | Id of each order | Int(32) | Primary key |
| 2 | CustomerID | Id of each customer | Int(32) | Foreign key |
| 3 | OrderDate | Date of each order | Nvarchar(50) |  |
| 4 | PaymentType | Type of payment you choose | Nvarchar(50) |  |
| 5 | Status | Status of the order | Nvarchar(50) |  |

Customer table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Datafield name | Explain | datatype | key |
| 1 | Cid | Id of each customer | Int(32) | Primary key |
| 2 | Name | Name of each customer | Nvarchar(50) |  |
| 3 | Username | Username to enter web | Nvarchar(50) |  |
| 4 | Password | Password to enter web | Nvarchar(50) |  |
| 5 | Phone | Phone number of customer | Int(32) |  |
| 6 | Address | Address of each customer | Nvarchar(50) |  |

Product table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Datafield name | Explain | datatype | key |
| 1 | Pid | Id of each product | Int(32) | Primary key |
| 2 | ProductName | Name of each product | Nvarchar(50) |  |
| 3 | Price | Price of each product | Int(32) |  |
| 4 | Quantity | Quantity of product | Nvarchar(50) |  |
| 5 | ImageURL | URL of image for each product | Nvarchar(50) |  |
| 6 | SupplierID | Supplier of product | Int(32) | Foreign key |

ProductCategory

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Datafield name | Explain | datatype | key |
| 1 | Pid | Id of product | Int(32) | Primary key |
| 2 | CategoryID | Id of category | Int(32) | Primary key |

Category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Datafield name | Explain | datatype | key |
| 1 | CategoryID | Id of category | Int(32) | Primary key |
| 2 | CategoryName | name of category | Nvarchar(50) |  |

Supplier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Datafield name | Explain | datatype | key |
| 1 | Sid | Id of each supplier | Int(32) | Primary key |
| 2 | SupplierName | Name of supplier | Nvarchar(50) |  |

Orderdetail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Datafield name | Explain | datatype | key |
| 1 | Orderid | Id of each order | Int(32) | Primary key |
| 2 | Pid | Id of each product | Int(32) | Primarykey |
| 3 | Quantity | Quantity | Int(32) |  |
| 4 | Price | Price of each product times quantity | Int(32) |  |

**Chapter 4: Design**

**4.1 Design Controller**

|  |  |  |
| --- | --- | --- |
|  | Controller | Purpose |
| 1 | Cart Controller | Add, delete, update the shopping cart. Check out. Process order when user want to buy products. |
| 2 | Category controller | List all item in category table.  Add, edit, delete any item the list. |
| 3 | Customer controller | List all item in customer table.  Add, edit, delete any item the list. |
| 4 | Login controller | To login the web and web use role to choose who admin who user |
| 5 | Order controller | To make the user order the product they want |
| 6 | Product controller | List all item in product table.  Add, edit, delete any item the list. |
| 7 | PublicPage controller | Show the product of each category. |
| 8 | Register controller | In order the for guest create their account. |
| 9 | Supplier | List all item in supplier table.  Add, edit, delete any item the list. |

**4.2 Handel Function**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Method** | **Purpose** | **The line contains the declaration** |
| 1  2  3  4  5  6 | [HttpPost]  public JsonResult LoginUser(UserViewModel userViewModel)  public ActionResult AddItem(int? id)  private int isExistingCheck(int? id)  public ActionResult ProcessOrder(FormCollection frc)  public ActionResult UpdateCart(FormCollection frc)  public JsonResult Register(UserViewModel userViewModel) | Login to the website, the system will check if an account has role 1 you will be admin and with the role 0 you will be customer  Add item into cart  Check if the product id that exist in the cart or not  Save cart information into order table and and order detail table  Update product into cart when user add item  Register for guest | LoginController(line37)  CartController(line 30)  CartController(line 63)  CartController (line 107)  CartController(line 88)  RegisterController(line36) |

**Chapter 5:** **Installation and testing**

|  |  |  |
| --- | --- | --- |
|  | **Situation** | **Purpose** |
| 1 | Input: data from database  Output: show data from index | Check the program can assign the database yes or not |
| 2 | Input: the data edit, delete the data, add the data  Output: change of data when adding, editing or deleting data | Check the code make sure that user can edit, add and del the data, and the data must change when user do that |
| 3 | Input: Login with admin or user  Output: decentralized user | Make sure that admin can add, del, edit some table that support for admin and user only view table |
| 4 | Input: Login with user  Output: User can order products | Make sure that user can order and modify item in the Cart |
| 5 | Deployed the application on IIS | Make sure that the application can work perfectly on browser without Visual Studio |

**CHAPTER 6. Conclude**

**6.1 Difficulties and solution**

|  |  |
| --- | --- |
| **Difficulties** | **Solution** |
| Create a shopping cart | <https://www.youtube.com/watch?v=h_jfcLICk_8&fbclid=IwAR3HXmtyNYmvubMNBnyP1MzMmwwyfpYiIsRwjHjTfhepx5rDUHOA4DHSH0Y> |
| Assign authority to controller and controller | <https://lethanhname.blogspot.com/2016/12/aspnet-mvc-5-authentication-filters.html>   * And with teacher help |

**6.2. For further implementation**

1. Users can sort products alphabetically
2. Make the theme more beautiful and easier for user to use

**6.3. Advantages**

1. Simple easy-to-use software for drawing genealogy
2. Know more about how to solve the problem
3. Users can manage their products

**6.4. Limit**

1. The IU is not good-looking

**Chapter 7: Task distribution**

|  |  |  |
| --- | --- | --- |
|  | Name | assignment |
| 1 | Lê Tuấn Đạt | Homepage, Category, PP Report, Delete, write report. |
| 2 | Nguyễn Mạnh Tiến | Login (user authentication), Add, Listing item with category, write report. |
| 3 | Nguyễn Hoàng Trường Minh | Order (add to card),  Register, Detail, write report. |