**BTEC Level 5 HND Computing and Systems Development**

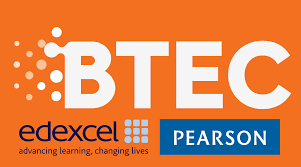


A Project Programming in .NET Presented

by

Nguyễn Đình Tùng(GC00952)

Assignment2



Submitted to the Graduate School of the

University of Greenwich Vietnam in partial fulfillment

Wednesday, 18th July 2018

Lecturer: Đỗ Quốc Bình

**Table of Contents**

[**Introduction** 5](#_Toc521156227)

[**Task 1: Be able to design .NET solutions.** 6](#_Toc521156228)

[1.2: Design .NET Programming solution to a given problem 6](#_Toc521156229)

[1. Use case diagram(s) 6](#_Toc521156230)

[2. ERD for the back-end database for this system 7](#_Toc521156231)

[3. Class Diagram 8](#_Toc521156232)

[4. Graphical user interface design (GUI) 10](#_Toc521156233)

[1.2: Explain the components and data file structures required to implement the given design 16](#_Toc521156234)

[1. Feature register 18](#_Toc521156235)

[2. Feature login 20](#_Toc521156236)

[3. View all products 22](#_Toc521156237)

[4. Feature search products 22](#_Toc521156238)

[5. Feature add products to cart 23](#_Toc521156239)

[6. Feature payment and cancel products in cart 24](#_Toc521156240)

[1.3: Evaluate potential delivery environment and interaction 26](#_Toc521156241)

[1. Delivery environment 26](#_Toc521156242)

[2. Interaction 30](#_Toc521156243)

[**Task 2: Be able to implement .NET solutions** 32](#_Toc521156244)

**Table of figures**

[Figure 1. Use-case diagram 7](#_Toc521156140)

[Figure 2. ERD diagram 8](#_Toc521156141)

[Figure 3. Class diagram 10](#_Toc521156142)

[Figure 4. Wireframe for feature register 11](#_Toc521156143)

[Figure 5. Wireframe for feature Login 11](#_Toc521156144)

[Figure 6. Wireframe for feature View Customers 12](#_Toc521156145)

[Figure 7. Wireframe for feature Order 12](#_Toc521156146)

[Figure 8. Wireframe for feature View Cart 13](#_Toc521156147)

[Figure 9. Wireframe for feature Management Categories 13](#_Toc521156148)

[Figure 10. Wireframe for feature Management Manufacturers 14](#_Toc521156149)

[Figure 11. Wireframe for feature Management Products 14](#_Toc521156150)

[Figure 12. Wireframe for feature Management Customers 15](#_Toc521156151)

[Figure 13. Wireframe for feature View Report 15](#_Toc521156152)

[Figure 14. Files structures 16](#_Toc521156153)

[Figure 15. Folder Models 17](#_Toc521156154)

[Figure 16. Structures Views 17](#_Toc521156155)

[Figure 17. Structures Controllers 18](#_Toc521156156)

[Figure 18. DB connect 30](#_Toc521156157)

[Figure 19. Add data to database 30](#_Toc521156158)

[Figure 20. Database update data 31](#_Toc521156159)

[Figure 21. Database delete data 31](#_Toc521156160)

[Figure 22. Interface Register 32](#_Toc521156161)

[Figure 23. Login Interface 32](#_Toc521156162)

[Figure 24. Interface Home page customers 33](#_Toc521156163)

[Figure 25. Interface Order 33](#_Toc521156164)

[Figure 26. Interface View Cart Customers 34](#_Toc521156165)

[Figure 27. Interface Management Categories 34](#_Toc521156166)

[Figure 28. Interface Management Manufacturers 34](#_Toc521156167)

[Figure 29. Interface Management Products 35](#_Toc521156168)

[Figure 30. Interface Add new product 35](#_Toc521156169)

[Figure 31. Interface Update product 36](#_Toc521156170)

[Figure 32. Interface Management Customers 36](#_Toc521156171)

[Figure 33. Interface View Report 37](#_Toc521156172)

**Table of tables**

[Table 1. Compare between environment 30](#_Toc521156187)

# **Introduction**

This report we will discuss about how to design .NET Programming at this project and explain about component in design, explain about use-case diagram, class diagram, ERD diagram and design wireframe for interfaces will display with user. Continue we will discuss about how to this program work and how to I write code for this program, technologies I used in program bring effectives how about the program and finally I will show all interfaces in this program.

# **Task 1: Be able to design .NET solutions.**

## 1.1: Design .NET Programming solution to a given problem

### 1. Use case diagram(s)

Same with reality a store allow customers can buy computers or computer accessories, so sure in my program must have role intended for customers. After then I had idea for customers role I was asked me, who will management store, management products, management categories, management manufacturers, write report for each month… I had an idea for managers roles to management store. So, sure in my program I was determined two main roles this is customers and managers. Below I will specific analysis for each role.

Role intended for Customers: with customers, they can register a new account when they want buy computers or computer accessories at store, after they were registered, they can login to system store online and search for themselves products that they like and they can buy this laptop or computer accessories depending on economic conditions. Why we must requirement customers register before buy products because from information customers were registered, we can deliver follow address that they were registered

Role intended for Managers: to management quantity of products in store as add new a product to system store online or update products about quantity, price, management categories, management manufacturers… role intended for managers is very important. Outside they can management status of account of customers. For example when customers want register new account must need have managers accept this accounts, because if to customers register free will lead to many problems for system and store. Or staffs can view report for result activities of store each month…

So, why we must divide roles for system of store . Because when we divide role to management will easier when not divide roles and increase effective for store, increase professionalism…

Below is use-case diagram I was designed for system store selling computers or computer accessories online

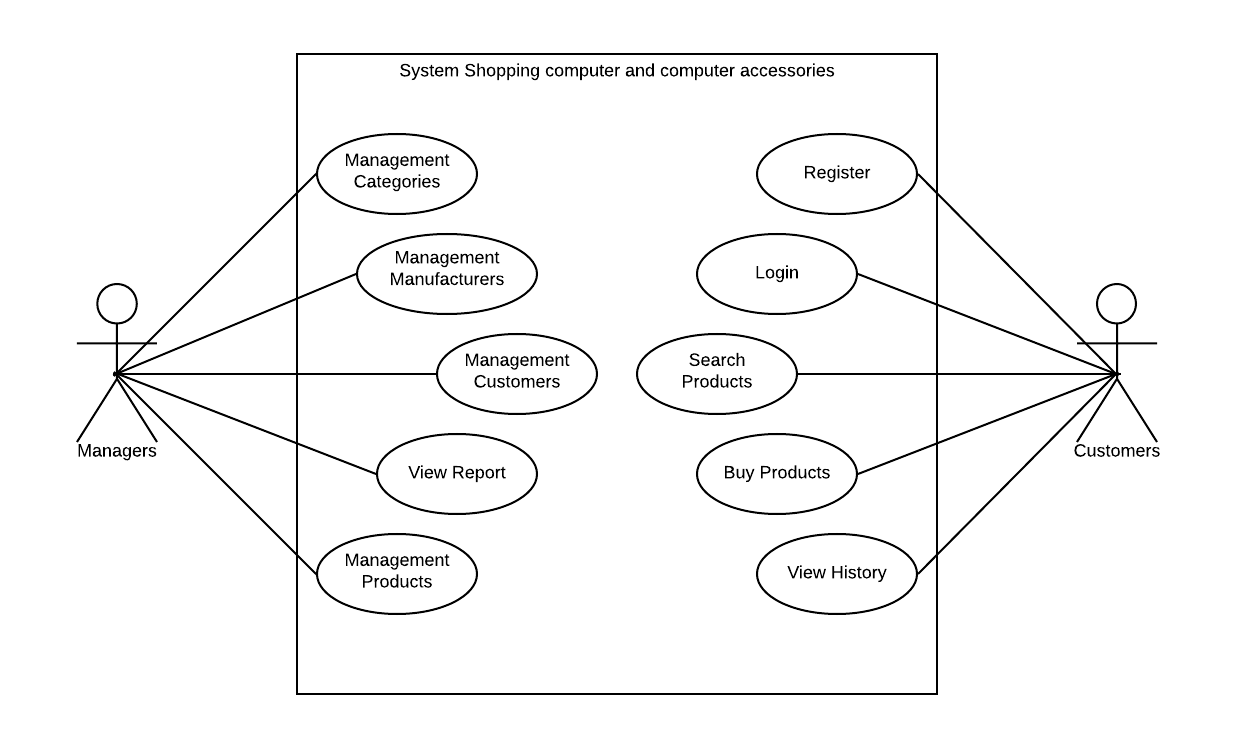


Figure 1. Use-case diagram

### 2. ERD for the back-end database for this system

From use-case diagram above I was designed a ERD diagram intended for database with relationship between tables includes primary key, foreign key…

I was designed seven tables for database this is: table Staffs, table Customers, table Categories, table Manufacturers, table Products, table Orders and table OrderDetail

Table Staffs includes attributes as Username, Password, FullName, Address, Gender, Email, Phonenumber, Birthday, Avatar, Created, Salary, Status with primary key is field Username

Table Customers same with table Staffs, but also have difference this field Salary at table Staffs be instead by field Level at table Customers.

Table Categories and table Manufacturers same together with attributes CatId, CatName intended for table Categories and MafId, MafName intended for table Manufacturers, field Description is have in both table. Two fields CatId and MafId will is primary key for each table.

Table Products includes attributes as ProductId, ProductName, CatId, MafId, PriceHire, Quantity, Description, ImageProd, Level, Status, Created, Creator with primary key is ProductId and three foreign keys this is CatId will foreign key with table Categories, MafId will foreign key with table Manufacturers and Creator will foreign key with table Staffs.

Table Orders includes attributes as OrderId, UsernameC, UsernameS, TotalAmount, OrderDate, Status with primary key is OrderId, there are two foreign keys this is UsernameC foreign key with field Username at table Customers and UsernameS foreign key with field Username at table Staffs.

Table OrderDetail includes attributes as OrderDetailId, OrderId, ProductId, UnitPrice, Quantity with primary key is OrderDetailId, there are two foreign keys this is OrderId foreign key with field OrderId at table Orders and ProductId foreign key with field ProductId at table Products.

Below is image ERD diagram for system Selling Computers and computer accessories.

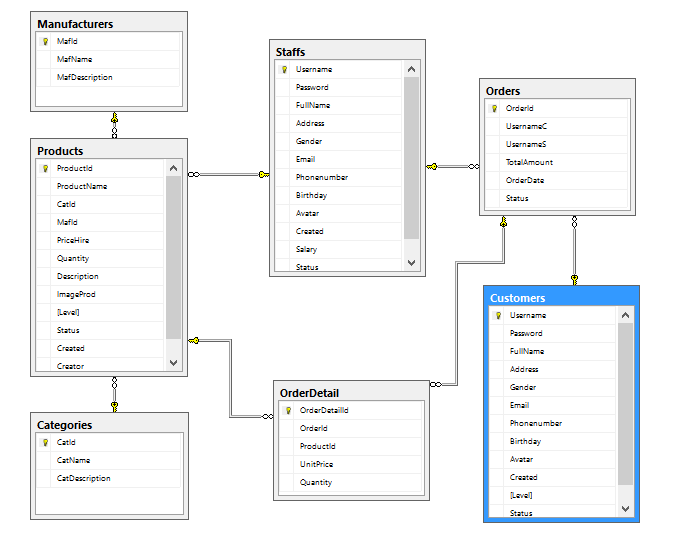


Figure 2. ERD diagram

### 3. Class Diagram

With ERD diagram above, I was designed for program structures classes as follows: one class for customers and one class for staffs, one class for categories and one class manufacturers, one class to save information for products (computers and computer accessories) one class to save information order of customers with system store selling computers and computer accessories online.

Why need have classes?, because information that customers were registered or buy computers and computer accessories will must be saved to database and I need create a class intended customers or order to can mediate between view users and controller and database. Problem use classes is very important, it will help increase performances for system and clear design help programmer easier management system and can easier in debugs problems. Classes will is an important link in the MVC design model. It represents the data (value) of an object…

Why need have class intended customers?, because class will represents the data (value) of an object , represents information from customers to system and opposite, help system determined clearly activities of customers, status of customers… That will help we management a way thoroughly avoid will happened bugs from system…

The other classes also same with customers, they only change about attributes in each class and other activities…

After we determined clearly classes and why need them, we will must set up relation for classes. For example in my program about relationship, I will create relation for class customers, products and class orders because when staffs want management orders from customers will need must know account this customers and code products to can easier in management report as management order of store. I has created relationship as follows class Staffs association with classes Categories, Customers, Products, it can management all of them, Categories association with class Products, Manufacturers association with class Products, because class Categories and Manufacturers will provides CatId and MafId for class Products and class Order will association with classes Customers and Orders, because an Order always be created by a Customer and CarIds.

Below is class diagram I was designed for my program selling computers and computer accessories

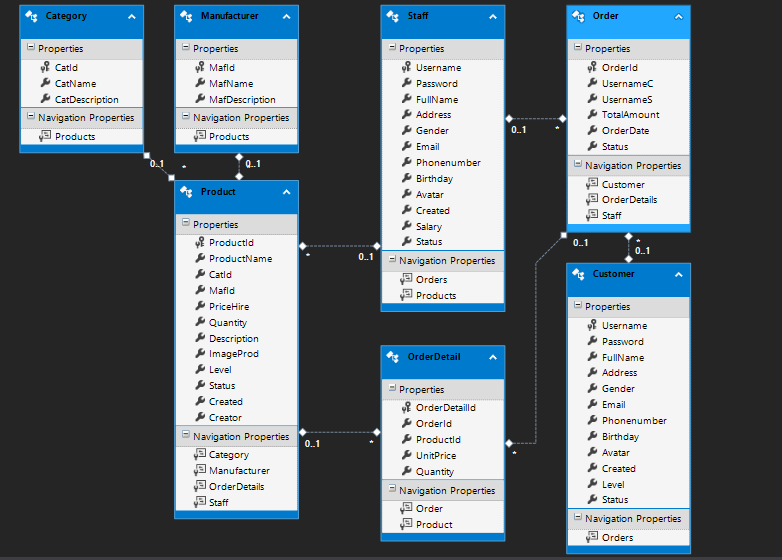


Figure 3. Class diagram

### 4. Graphical user interface design (GUI)

After I was designed use-case diagram, ERD diagram and class diagram I will continuous design interfaces will display with user. Below is some wireframe I was designed to simulation interfaces I will write code to display for users.

First I was designed for role Customers with features as register, login , order, view cart. Below is wireframe for feature register a new account of customers

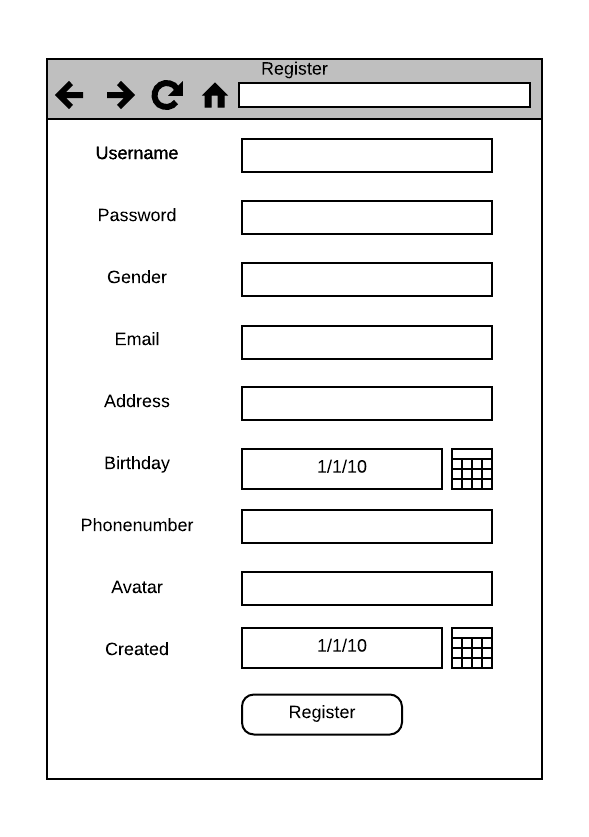


Figure 4. Wireframe for feature register

Next is wireframe for feature login to system of customers

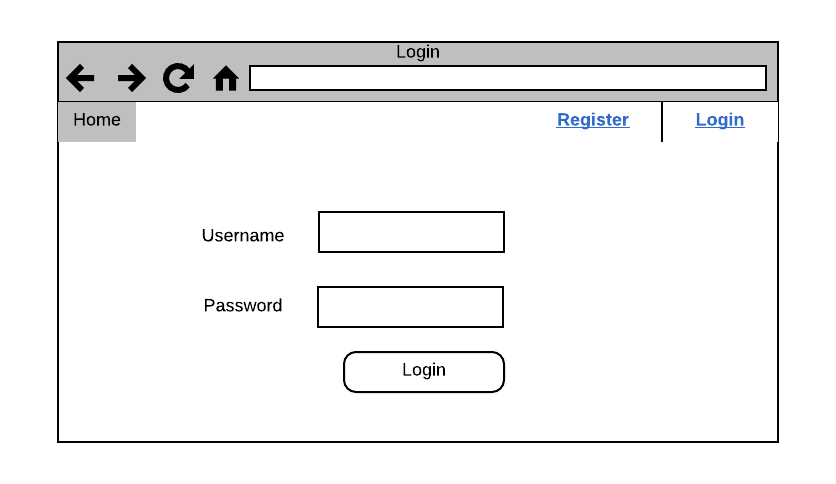


Figure 5. Wireframe for feature Login

Next is wireframe for feature view customers

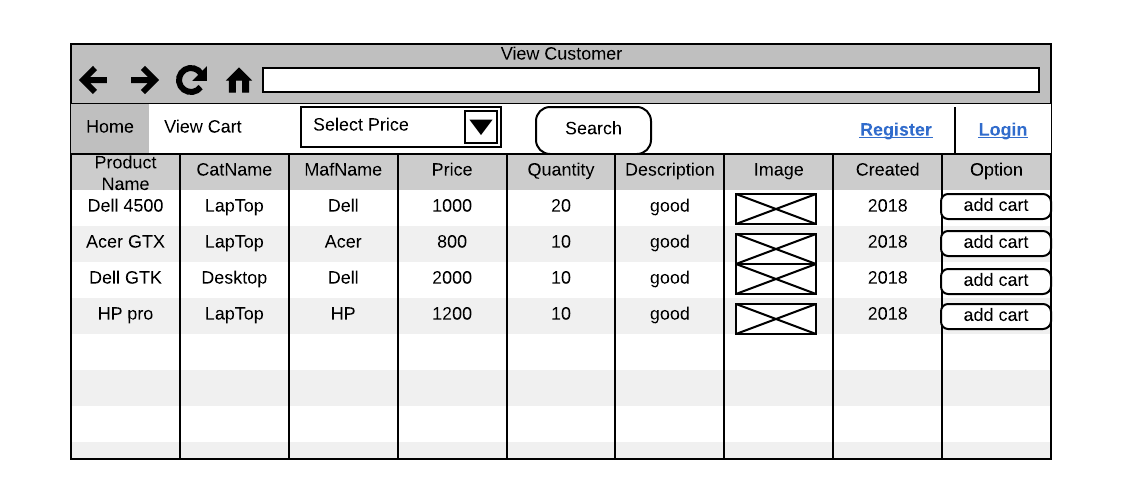


Figure 6. Wireframe for feature View Customers

Next is wireframe for feature order of customers

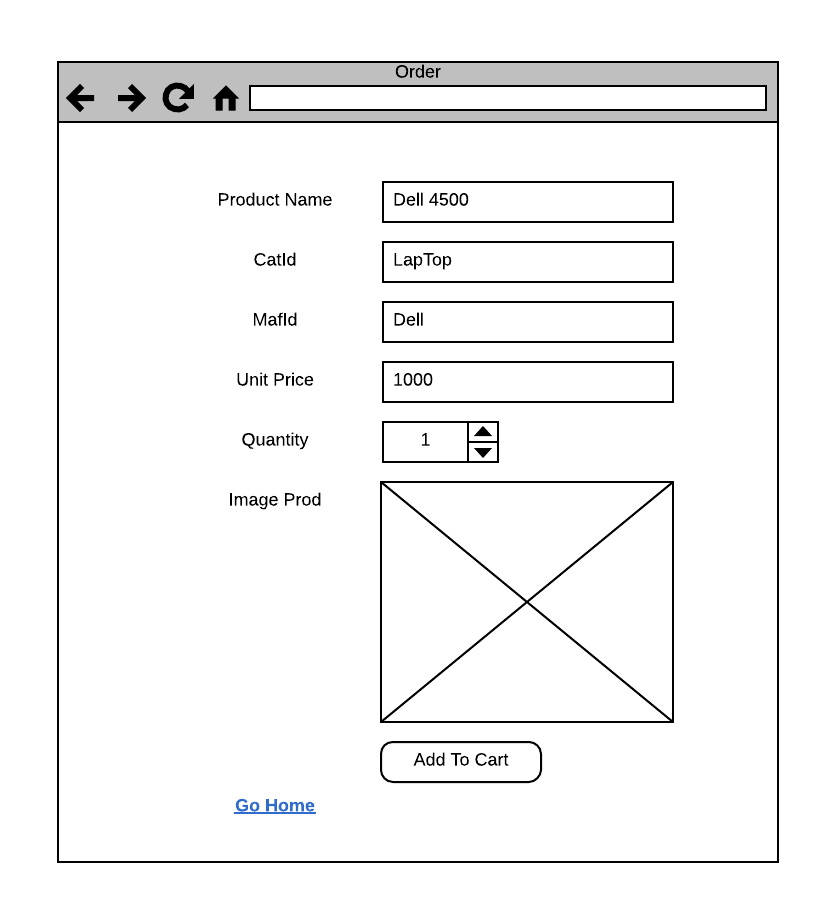


Figure 7. Wireframe for feature Order

Next is wireframe for feature view cart of customers

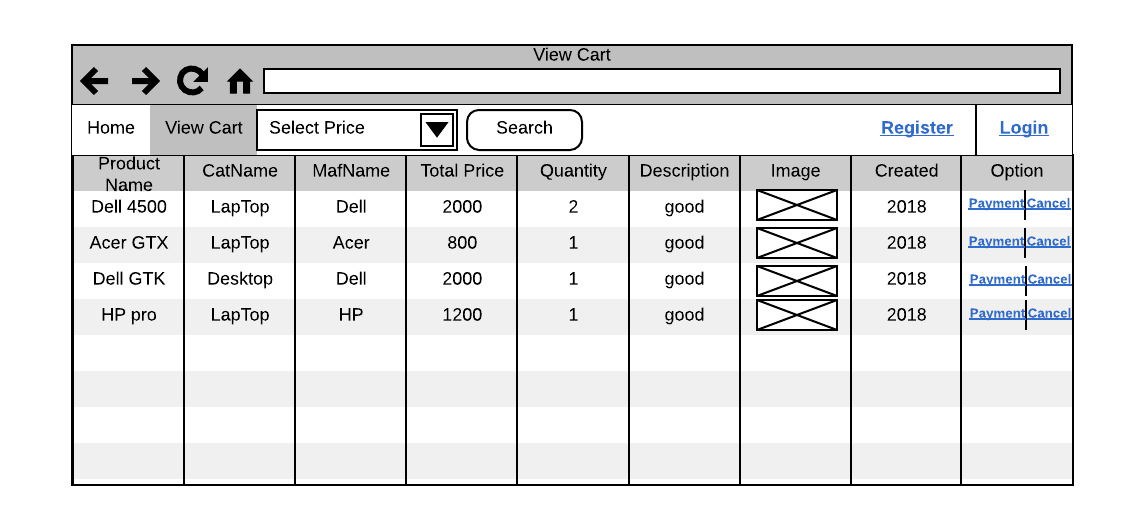


Figure 8. Wireframe for feature View Cart

Continuous I was designed for role Managers with features as management categories, management manufacturers, management products, management customers and view report

First is wireframe for feature management categories of managers

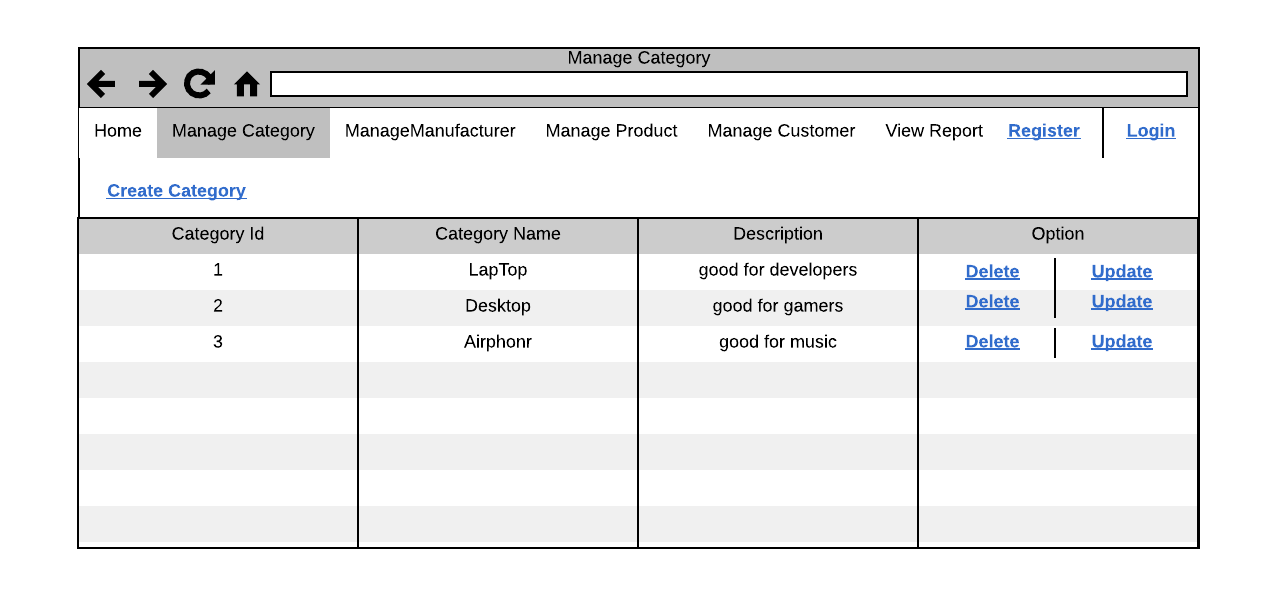


Figure 9. Wireframe for feature Management Categories

Next is wireframe for feature management manufacturers of managers

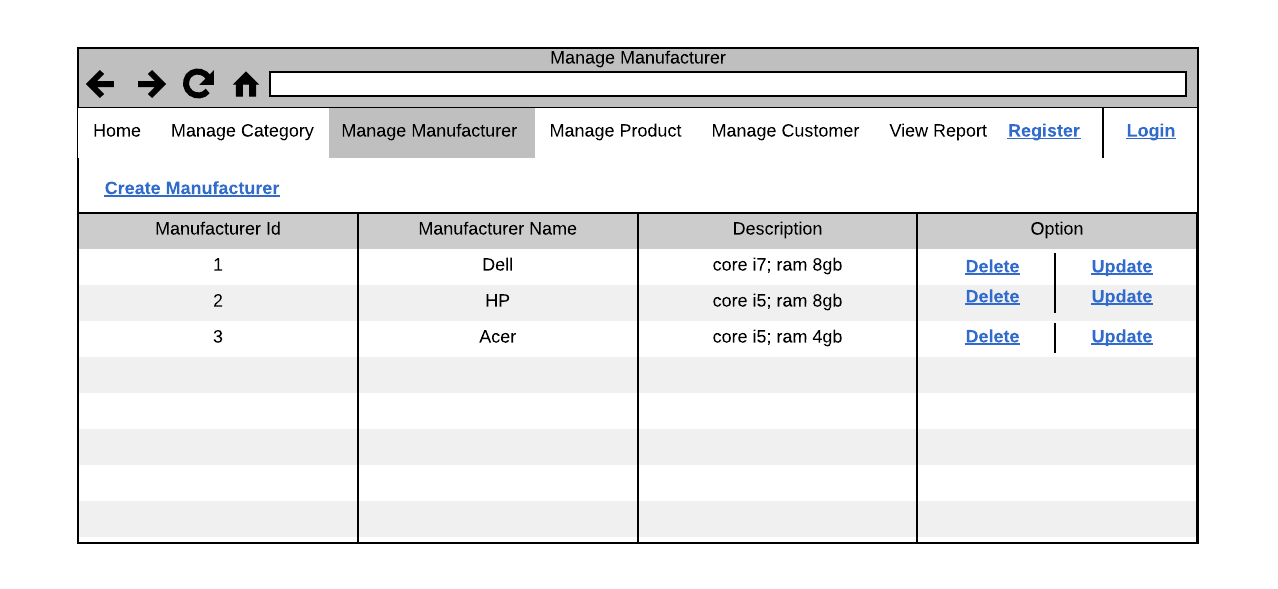


Figure 10. Wireframe for feature Management Manufacturers

Next is wireframe for feature management products of managers

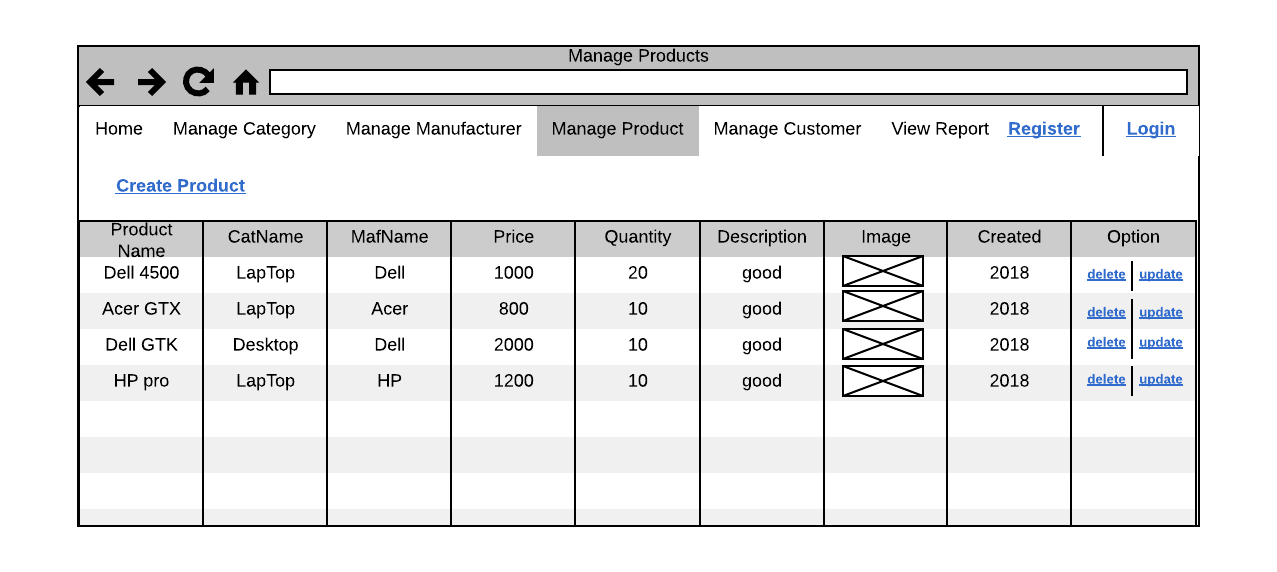


Figure 11. Wireframe for feature Management Products

Next is wireframe for feature management customers of managers

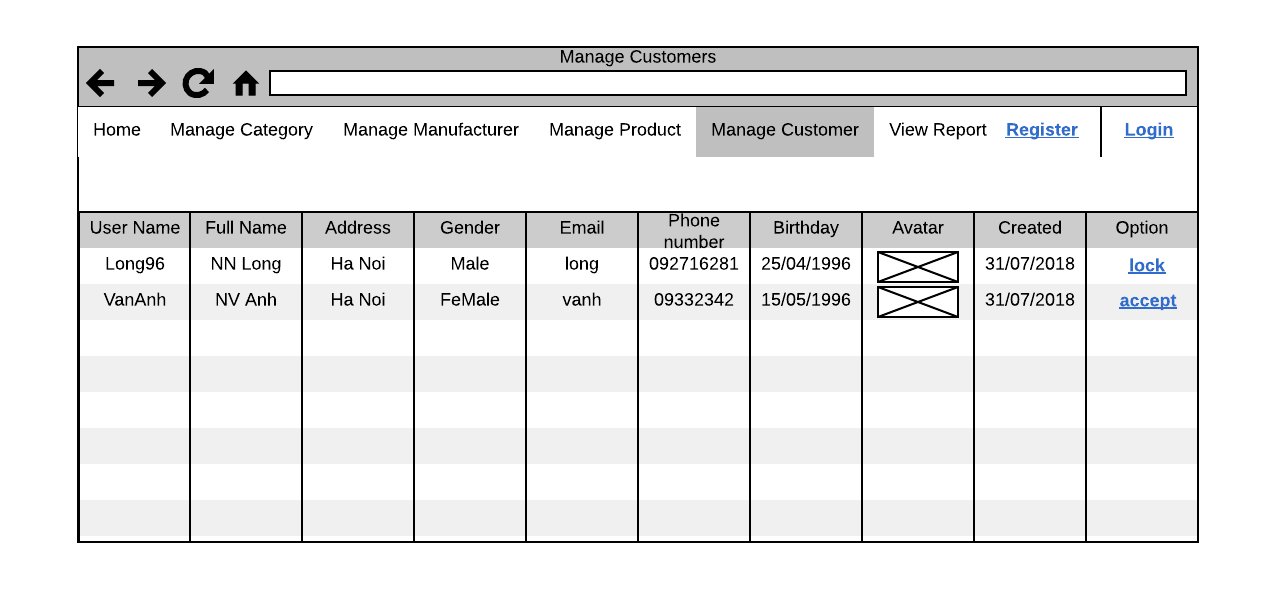


Figure 12. Wireframe for feature Management Customers

Next is wireframe for feature view report of managers

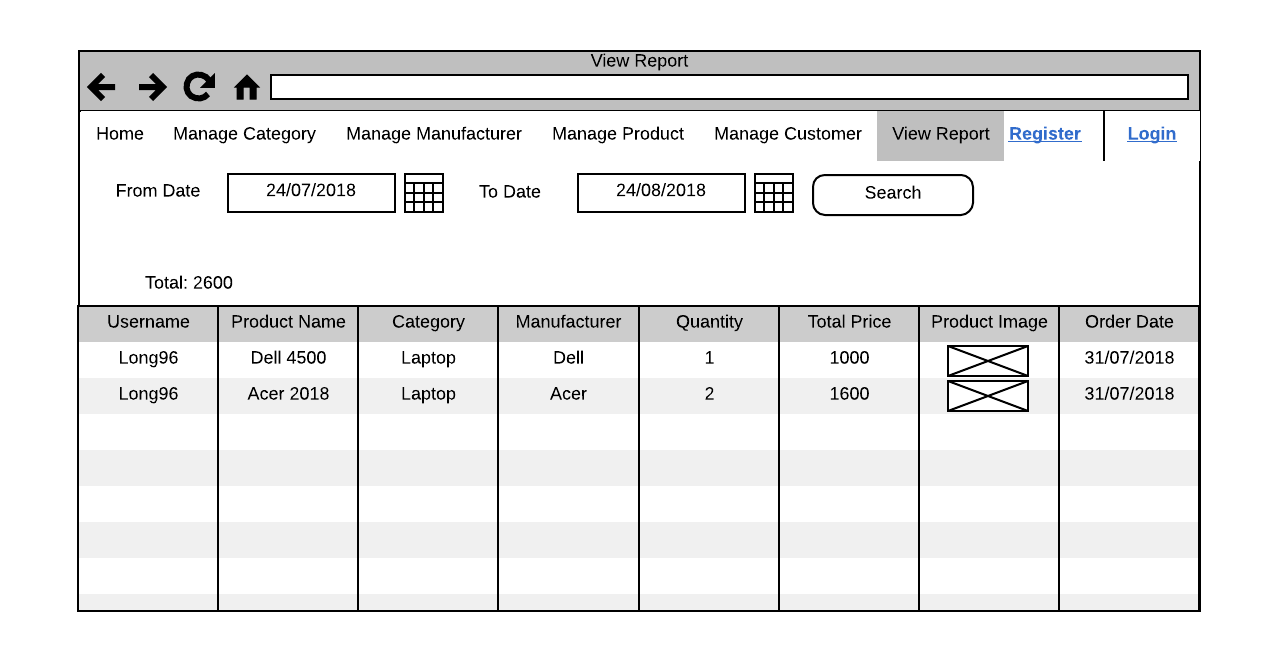


Figure 13. Wireframe for feature View Report

* **Conclusions:** This part I explain about problem How to I was designed for my program follow topic selling computers and computer accessories. I given use-case diagram, ERD diagram, class diagram and some wireframe and explain about them. Through that you can understand more about how to I design a program and how to analysis for a topic.

## 1.2: Explain the components and data file structures required to implement the given design

My program was designed follow model MVC with three parts very important this is Model, View and Controllers, beside my program also created some other folders to storage image and some content default. Below is image files structures after I was wrote code for my program.

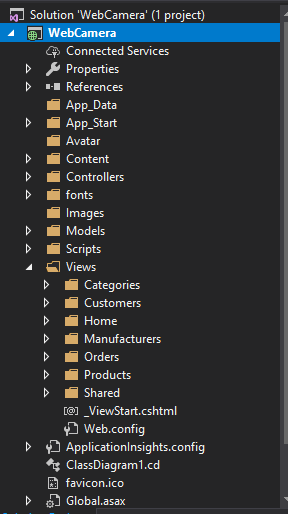


Figure 14. Files structures

In this picture you can easy see three folders to create model MVC in my program this is Model, View and Controller each folder will storage other classes and perform different tasks

First I will explain about folder Models, this folder storage files classes to connect with database and implement manipulates with database, at this program I used technology Entity framework to connect database and auto created classes corresponds to the tables in database. Here will receive customer or manager information through “ActionResult” at Controllers and perform features that users want with the database such as adding data, correcting data, or deleting unnecessary data. The system will retrieve data from the database through attributes in the classes and display them to the user. Below is image for files structures Models I created in my program

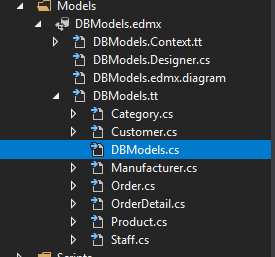


Figure 15. Folder Models

Next is folder Views are place storage files interfaces display with user. Here is place that user can manipulate with system as requirement system add new an account… user will enter data on interfaces and click to buttons to system implement manipulates. In here storage files have the .cshtml extension meaning is interfaces. The general view have task receive data from the user through interfaces and send to the controller through “ActionResult”. Below is image for files structures Views I created in my program.

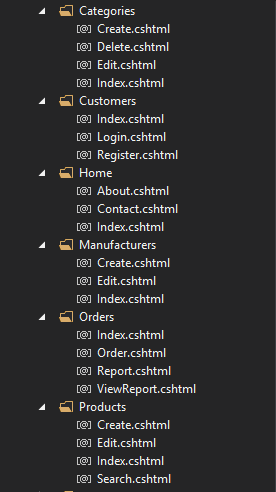


Figure 16. Structures Views

Next is folder Controller, it have task receive data or action from side views and passing data or action to models to save data or implement actions with database. In here storages files classes controller, in each controller can create “ActionResult” corresponds to with requirement of user as Create, Update, Delete … Below is image for files structures Controller in my program.

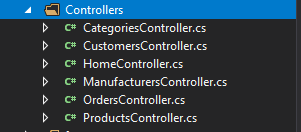


Figure 17. Structures Controllers

Outside I also create new some folders as folder Images to storage pictures of products in program and folder Avatar to storage pictures of customers when they register account in my program.

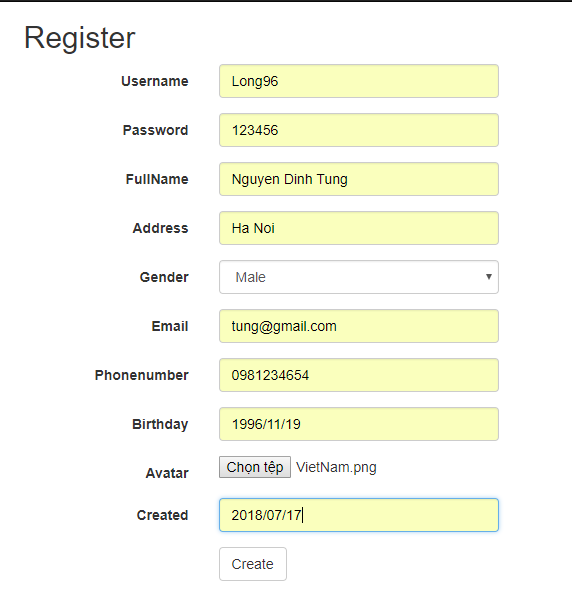
Use model MVC will easy manage my program when my program face with problems, help my program have performances better, it is more profession and help programmers reduce time to write code with big programs.

Continue we will discuss about how to program works and explain about main threading of program this is Customers will must register account, next is login to system, next is search products they want buy, next is add products they want buy to cart on system and payment products in cart of them. Below we will discuss from feature register to feature payment.

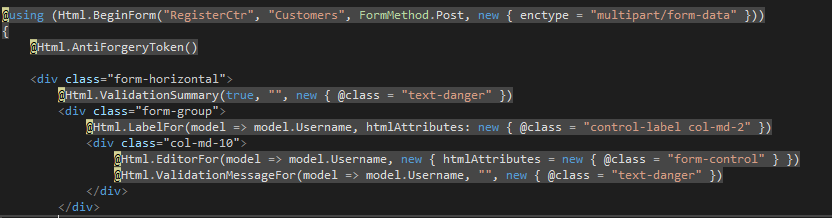
### Feature register

First users want login to system must register a new account, so to register a new account my program will how to work, below I will explain about components and how to connect between classes for you from model to view and controller.

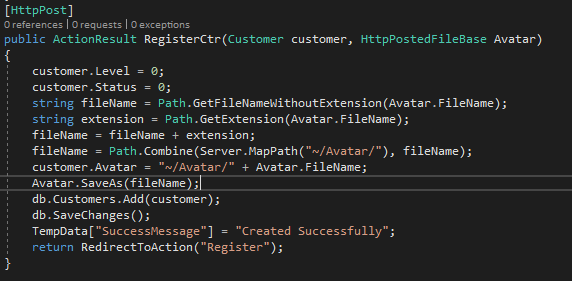
First system will requirement customers enter all attributes as image below



After customers fill all data to interfaces as above, customers will click button create to system passing information to controller through event form with event is “RegisterCtr” and at controller “Customers” as image below

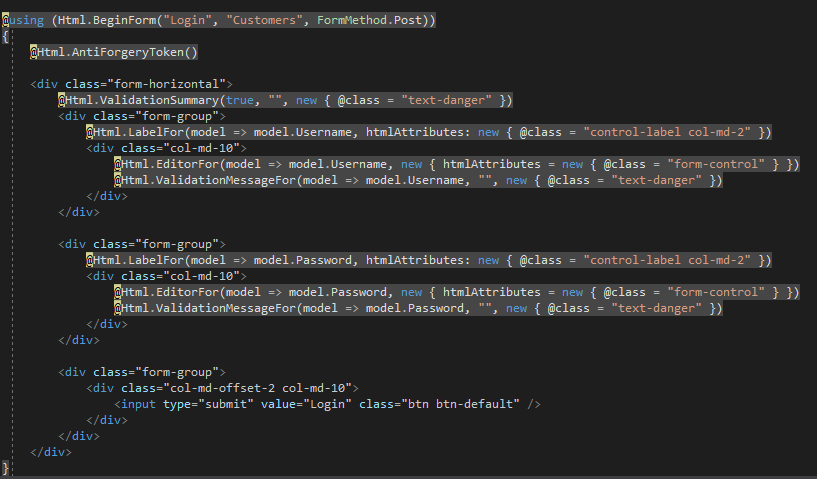


Continue at controller will receive data from interfaces and add to model “Customers” in save information to database as image below

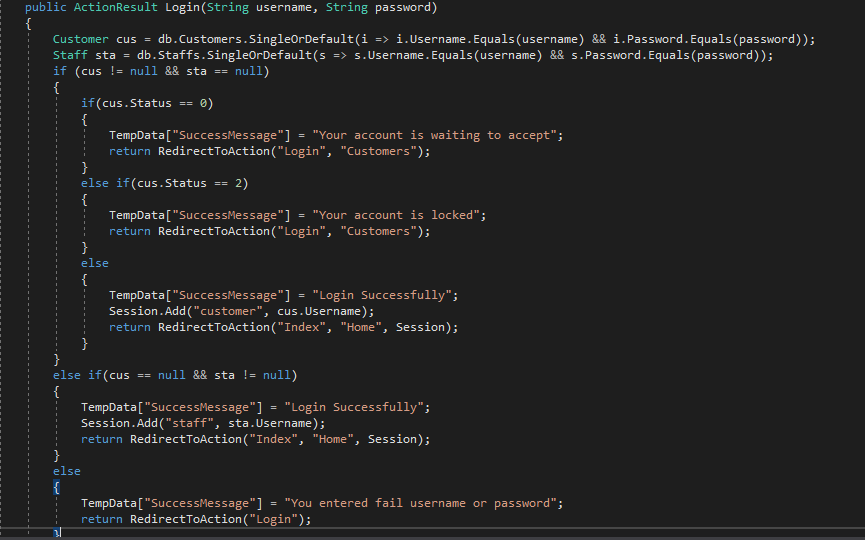


### Feature login

Continue is feature login, same with feature register. First users need enter Username and Password and send information to controller through event “Login” and at controller “Customers” as image below.

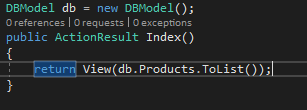


After passing information to controller, at controller I will get information from database with username and password received and check username and password, if username and password is exist already, if username and password not exist already system will info errors by dialog to users, if username and password same with account in table Customers, system will display view customers for users, if username and password same with account in table Staffs, system will display view intended for managers. Here I used session to passing data between interfaces as image below

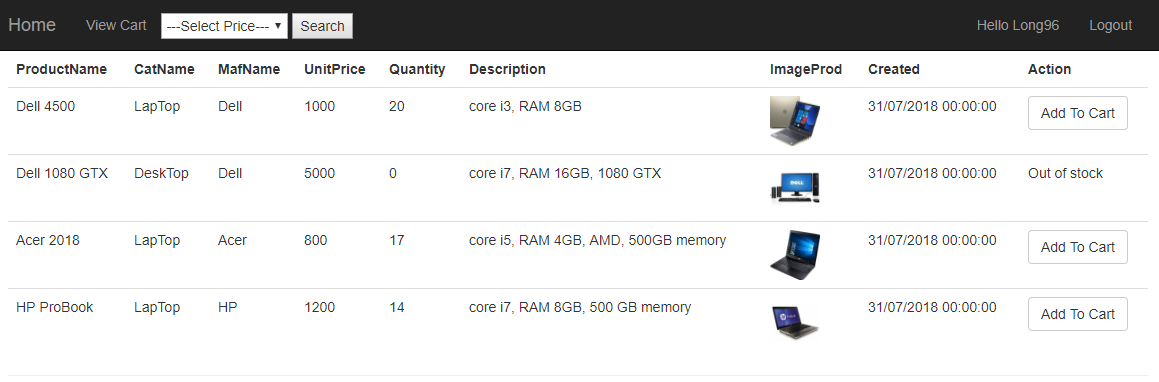
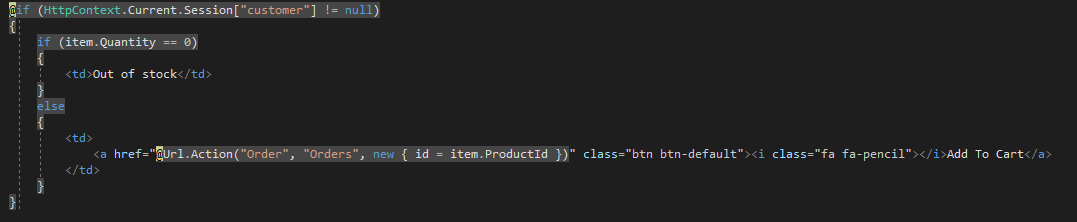


### View all products

After customers login successful to system intended for customers, system will show all products to customers. First I connected to database and requirement database return a list products as image below

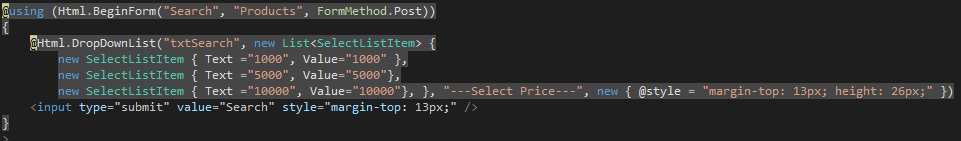


Continue I created an interface to display all products, I used foreach to display all products after controller send information and I checked have session username of customers will display button Add to cart and if quantity of product equal 0 will not display button add to cart that instead is text Out of stock as image below.

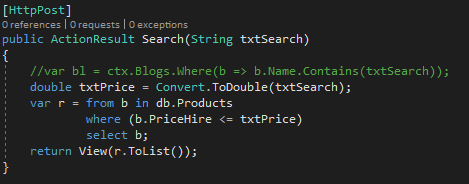


### Feature search products

At screen view customers and can search products follow price, first you need choose price on combobox and click to button search to system send data from combobox to event “Search” at controller “Products” as image below

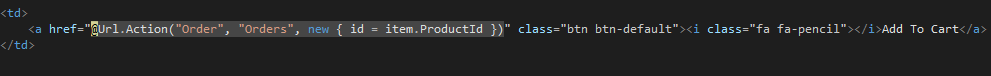


At controller I will use data be passed from interface to implement query search in database by query linq and return a list products with price less more price users chosen at combobox as below image

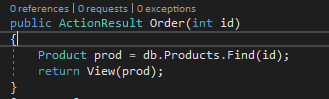


### Feature add products to cart

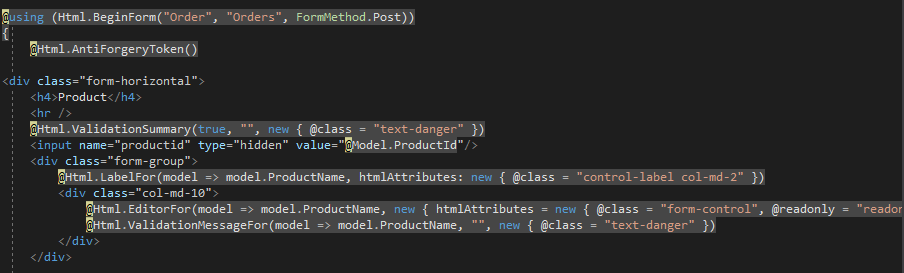
First when customers want buy a products at store, customers will must add to cart of them to implement manipulate add to cart customers will click to button “Add to cart” and system will passing ProductId of products chosen through event “Order” at controller “Orders” as image below



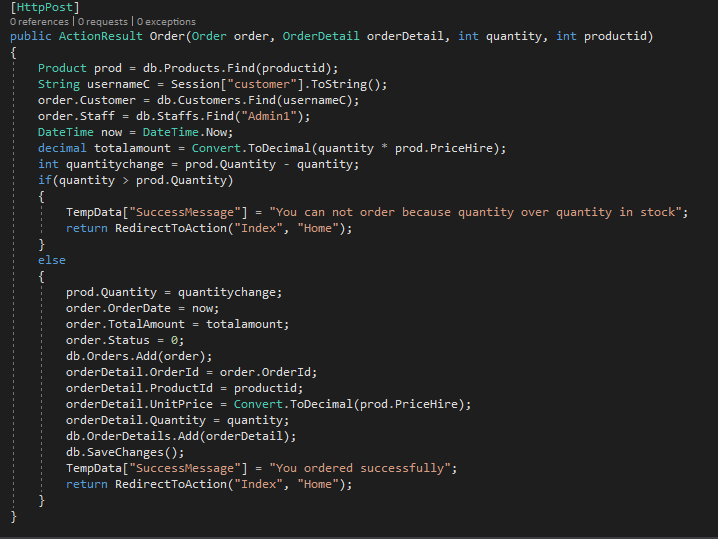
Continue at controller system will find Product follow ProductId passed from interface and return a Product with this ProductId and passing information in Product to interface Order as image below



Continue at interface order users will enter quantity they want buy this Product and send to controller through event “Order” and FormMethod [HttpPost] at controller “Order” as image below

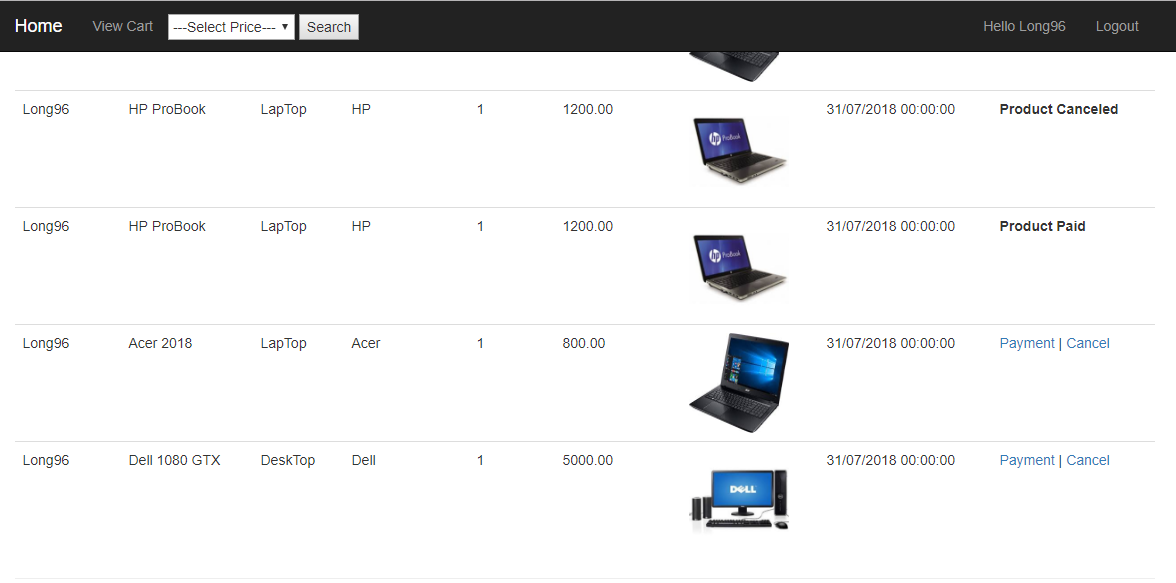
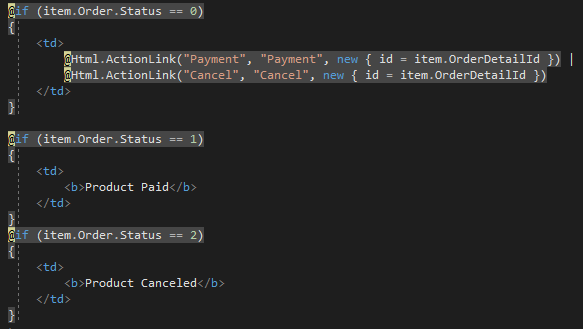
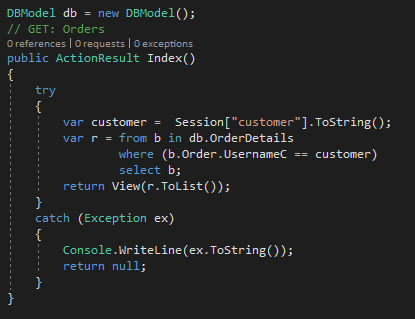


At controller Order [HttpPost] check quantity customers want buy and quantity in stock, if quantity customers want buy more than quantity in store (in database) system will info error to customers by message dialog, if quantity customers want buy less more quantity in stock (in database) system will simultaneous save information to table “Orders” and table “OrderDetails” to database and info success as image below

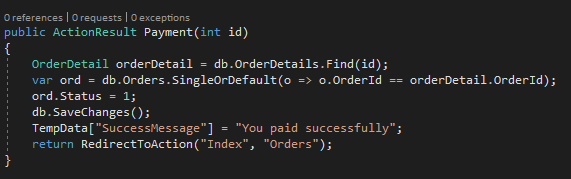


### Feature payment and cancel products in cart

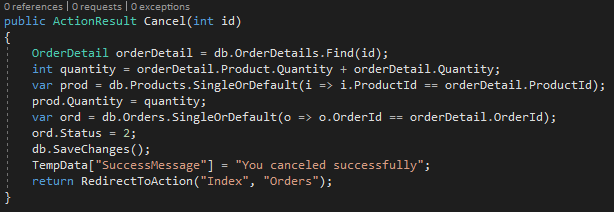
In order to pay or cancel a product ordered on the system, I need to display the products that the customer has ordered, I used the session to store the customer's username and I used the session. and execute the linq query to find the products that the customer has placed in the Order table in the database and return the list of products. At table Order I created a field Status to manage orders. I checked if Status equal 0 I will display two link buttons payment and cancel, if Status equal 1 I will display product paid, if Status equal 2 I will display product canceled as image below.



Continue if customers choose payment products system will passing OrderDetailId to controller through event “Payment” at controller “Orders”, at this controller I found OrderDetailId that customers want payment and change Status of order and info successful to customers as image below



If customers want cancel product system will passing OrderDetailId to controller through event “Cancel” at controller “Orders”, at this controller I found order from OrderDetailId and calculator quantity to return the quantity to the warehouse and change status become 2 mean is product canceled and save database as image below



## 1.3: Evaluate potential delivery environment and interaction

### 1. Delivery environment

This part I will compare three environments most popular today this is: Mobile Website, Web-base application, Desktop-base application and conclude on delivery environment that I use.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Mobile Website  environment | Web-base application environment | Desktop-base application environment |
| What is it? | The mobile web refers to browser-based Internet services accessed from handheld mobile devices, such as smartphones or feature phones, through a mobile or other wireless network. | A web-based application is any application that uses a website as the interface or front-end. Users can easily access the application from any computer connected to the Internet using a standard browser. | Desktop application (or Standalone app) is designed to run on single work stations or PC, so when you are about to test the desktop application |
| Advantages | ***Improved User Experience:*** Research on mobile website usability shows that mobile-optimized websites significantly improve user experience and satisfaction, which makes a positive impression when it counts.  ***Faster Download Speed:*** Mobile websites are specifically designed for mobile standards and optimal download speed, which means less waiting and more browsing.  ***Engagement and Context:*** A mobile website allows you to immediately engage users with mobile-specific features such as click-to-call, mapping functions. Furthermore, easy know user's location with website.  ***Improved Search/SEO performance:*** A mobile website provides improved rankings on mobile-friendly search engines such as Google and Yahoo, and also allows placement in a growing number of mobile and local directories.  ***Portability and Connectivity:*** A mobile website can be accessed anywhere, any time. This level of constant connectivity provides an unprecedented opportunity to connect with target audiences in new ways, wherever they may be.  ***Competitive Advantage:*** A mobile website ensures that you - not your competitors - capture visitor attention while you have it.  ***Integration with Offline Media:*** Mobile devices are unique in their ability to immediately bridge the gap between online and offline media through the use of QR codes. | ***Cost effective development:*** There is no need to develop and test it on all possible operating system versions and configurations. This makes development and troubleshooting much easier and for web applications that use a Flash front end testing and troubleshooting is even easier.  ***Accessible anywhere:***  Unlike traditional applications, web systems are accessible anytime, anywhere and via any PC with an Internet connection. This puts the user firmly in charge of where and when they access the application.  ***Easily customizable:***  The user interface of web-based applications is easier to customize than is the case with desktop applications and mobile website. This makes it easier to update the look and feel of the application or to customize the presentation of information to different user groups.  ***Accessible for a range of devices:*** In addition to being customizable for user groups, content can also be customized for use on any device connected to the internet. This includes the likes of PDAs, mobile phones and tablets.  ***Improved interoperability:*** It is possible to achieve a far greater level of interoperability between web applications than it is with isolated desktop systems.  ***Easier installation and maintenance:*** With the web-based approach installation and maintenance becomes less complicated too. Once a new version or upgrade is installed on the host server all users can access it straight away and there is no need to upgrade the PC of each and every potential user.  ***Adaptable to increased workload:*** Increasing processor capacity also becomes a far simpler operation with web-based applications. If an application requires more power to perform tasks only the server hardware needs to be upgraded.  ***Increased Security:*** Web-based applications are typically deployed on dedicated servers, which are monitored and maintained by experienced server administrators.  ***Flexible core technologies:*** Any of three core technologies can be used for building web-based applications, depending on the requirements of the application. The Java-based solutions (J2EE) from Sun Microsystems involve technologies such as JSP and Servlets. | ***Rapid Development:* Development Designed from the beginning to be a quick and easy solution to building graphical user interfaces (GUIs), especially when using Windows Forms in Visual Studio for Windows.**  ***Security:***Since the user keeps his or her data on their own computer systems (the program is running off the user’s desktop), this makes it harder for hackers to gain access to people’s data.  ***Available Controls:***Desktop application developers have a whole buffet of user-interactive controls to choose from. This goes for the out-of-the-box controls (e.g. Visual Studio for Windows) as well as 3rd-party controls.  ***Flexibility:***It is very easy to write desktop applications that take advantage of the user’s hardware (such as: scanners, cameras, Wi-Fi, serial ports, network ports, etc.)  ***Portability:***Desktop applications can be portable, but most are \*not\* portable and require manual installation from the user.  ***Maintenance:***Desktop applications usually need to be manually updated (or at least have manual approval) to install updates.  ***Performance:***Typically you will find that well-written desktop software running on a decent computer runs faster than web applications. |
| Disadvantages | - Mobile Application once made for a particular environment cannot be replicated to other environments  - User’s smart phone must connect to internet if they want to access data. It will  be trouble if has no Wi-Fi or 3G. | - Web Application needs active internet for it to work on your devices. Mobile App works without an active internet.  - Web Application cannot use the resources provided by the system (It can only use as much as provided to the browser), so it requires the service setup to the cloud to connect to the remote database for all the purpose  - Also there are many vulnerabilities of Web application such as cross site scripting , SQL injection and many other which mobile applications are pretty much immune. | - Not portable: You can’t access application without computer.  - Storage require: It may be get trouble if your computer not response enough storage. |

Table 1. Compare between environment

* **Conclusions:** In general, each environment has its advantages and shortcomings, so we should depend on the situation and choose to suit the requirements of the project. In this project I think the web-based application environment is most appropriate for the project because the security of this environment is better and more extensible than the other two environments, cost issues. In addition, the web-based application environment has more programming languages ​​to choose from than the other two environments. On the side of its limitations, the internet is very popular nowadays all over the world. We can see that bridesmaids can also connect to the internet for a computer. Moreover, with a laptop you can take it. Anywhere without fear of too cumbersome problem.

### 2. Interaction

In this program I used technology Entity Framework to connect with database:

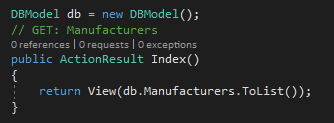


Figure 18. DB connect

You can see in the picture name of database this is DBModel and how to call database and use it in my program

Outside below is image to describe how to database add data into database, below is example about add an object Manufacturers in my program

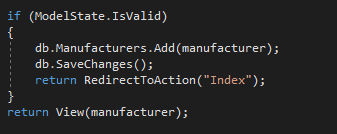


Figure 19. Add data to database

I just need write name variable of database called and select a table in database after add an object to database and save everything were changed

Below is how database implement manipulate update a manufacturer in my program

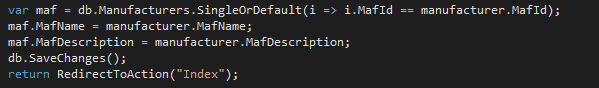


Figure 20. Database update data

I just need find in database with manufacturer that you want change with query linq as in image “SingleOrDefault” it will return an object manufacturer with correct manufacturer you want change and after I just need change information of this object and save change

Below is how to database delete data in table Manufacturers

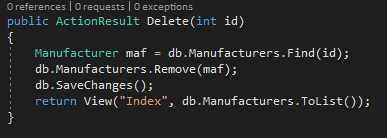


Figure 21. Database delete data

I just need find in database and write code “Remove” as in image and passing into an object to delete this manufacturer.

# **Task 2: Be able to implement .NET solutions**

In this task we will show some of interfaces in my project, includes for managers and customers:

1. Register for Customers

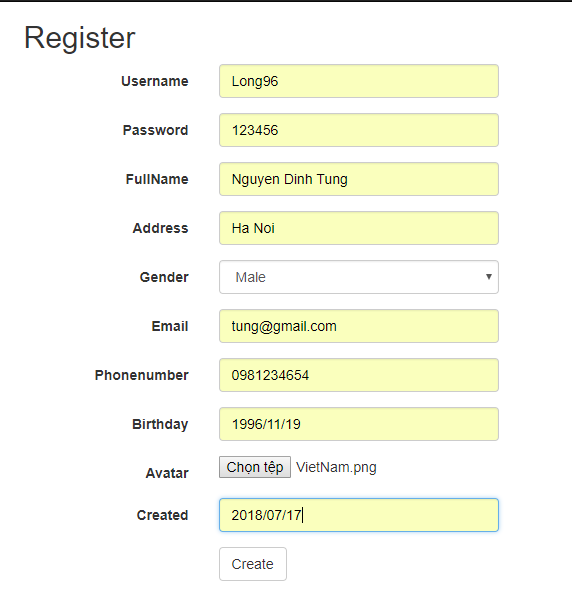


Figure 22. Interface Register

1. Login for Customers or Managers

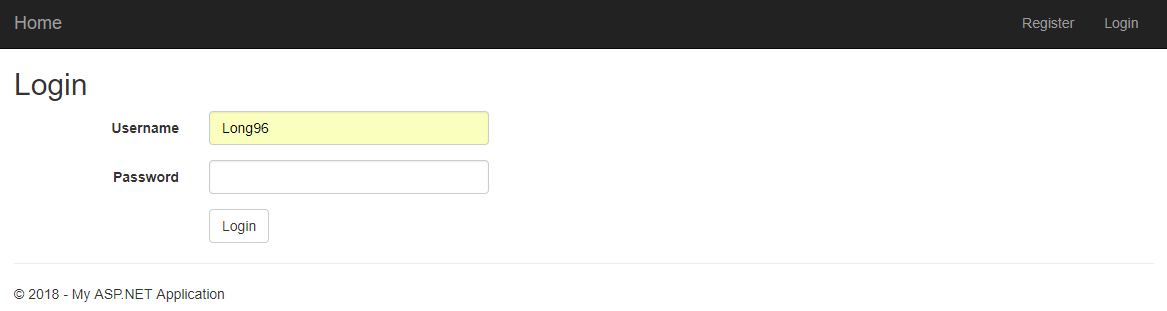


Figure 23. Login Interface

1. Home page for Customers

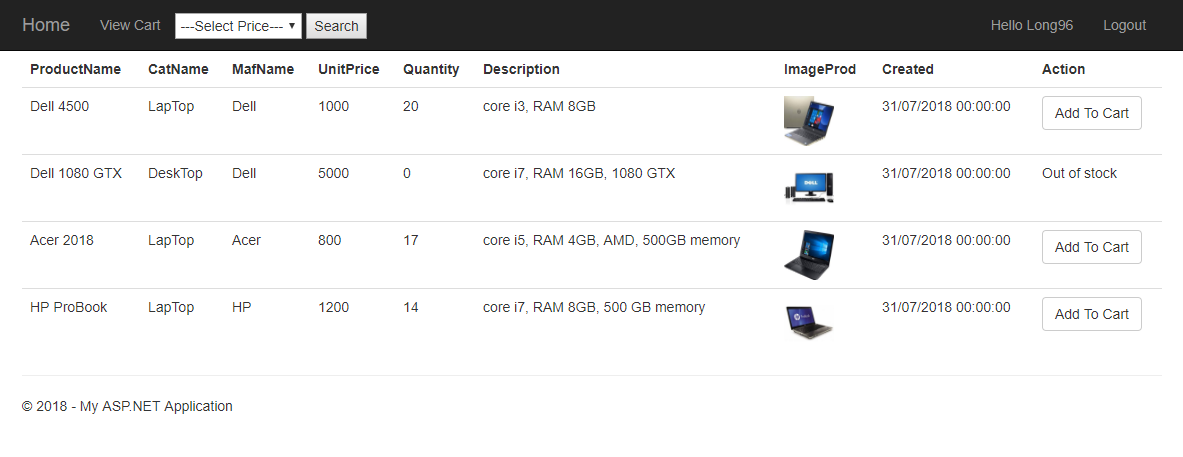


Figure 24. Interface Home page customers

1. Order for Customers

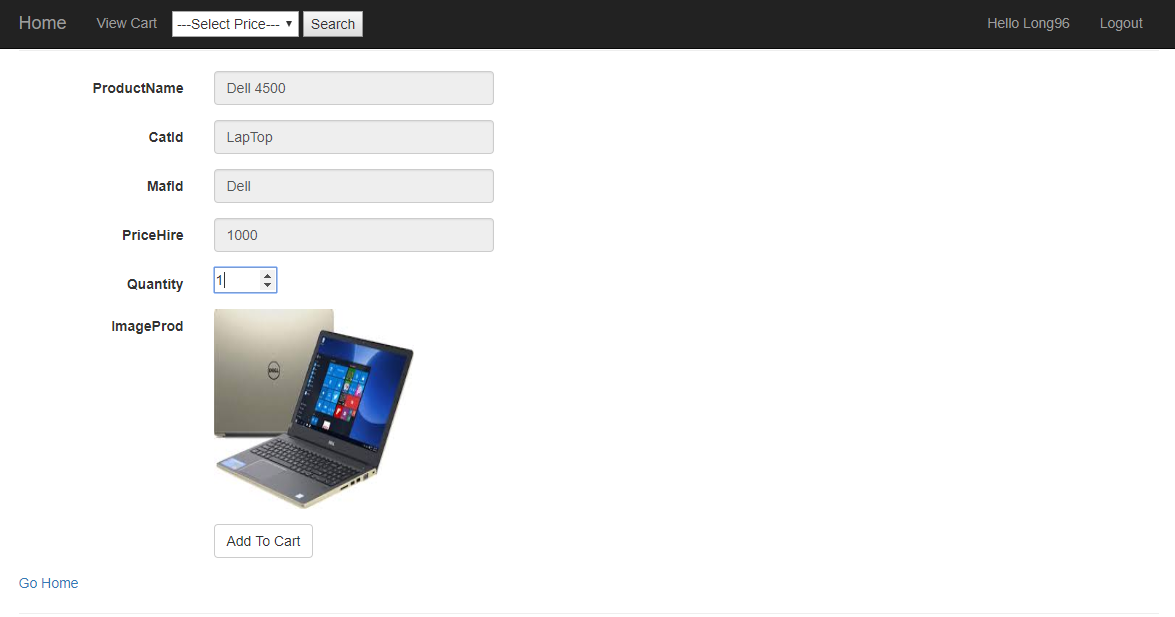


Figure 25. Interface Order

1. View Cart for Customers

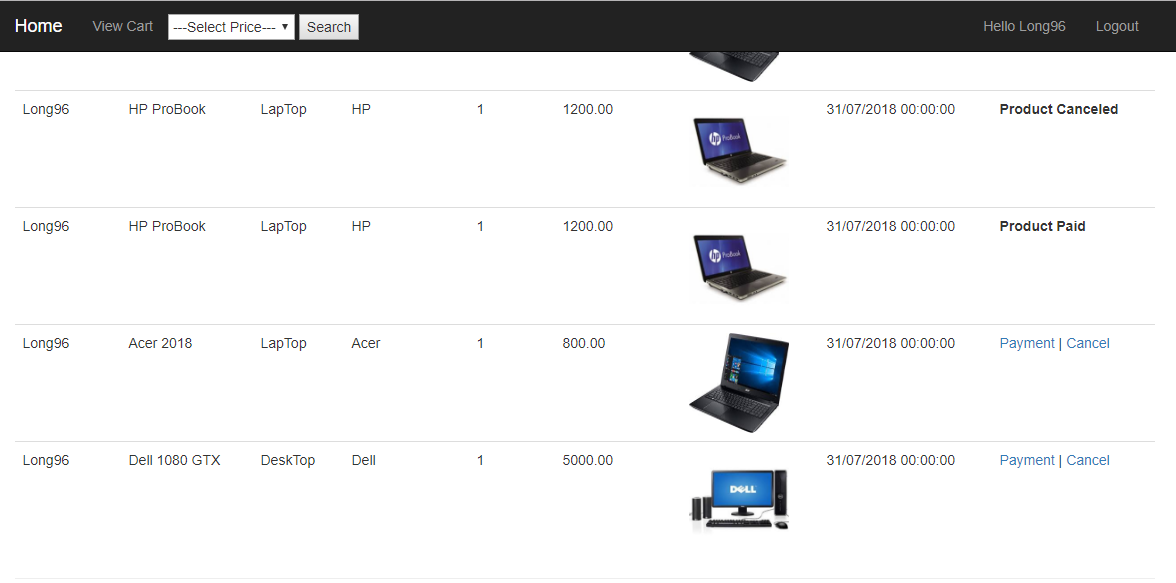


Figure 26. Interface View Cart Customers

1. Management Categories

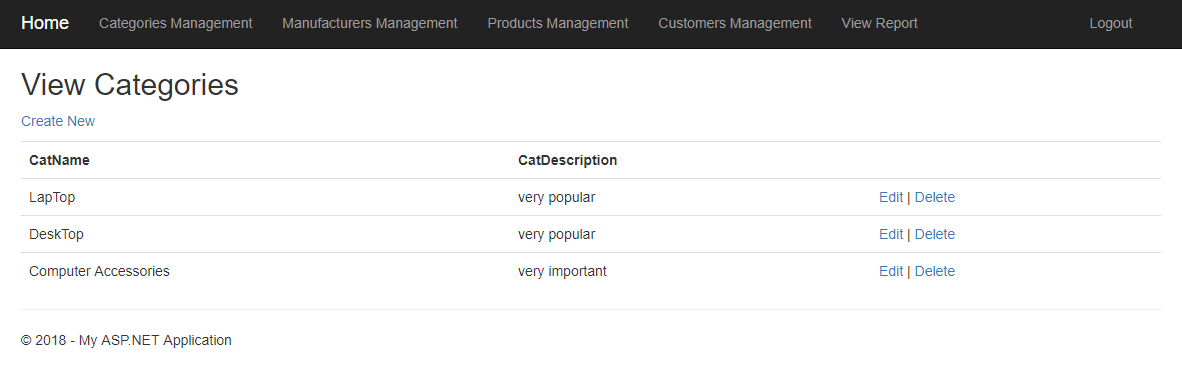


Figure 27. Interface Management Categories

1. Management Manufacturers

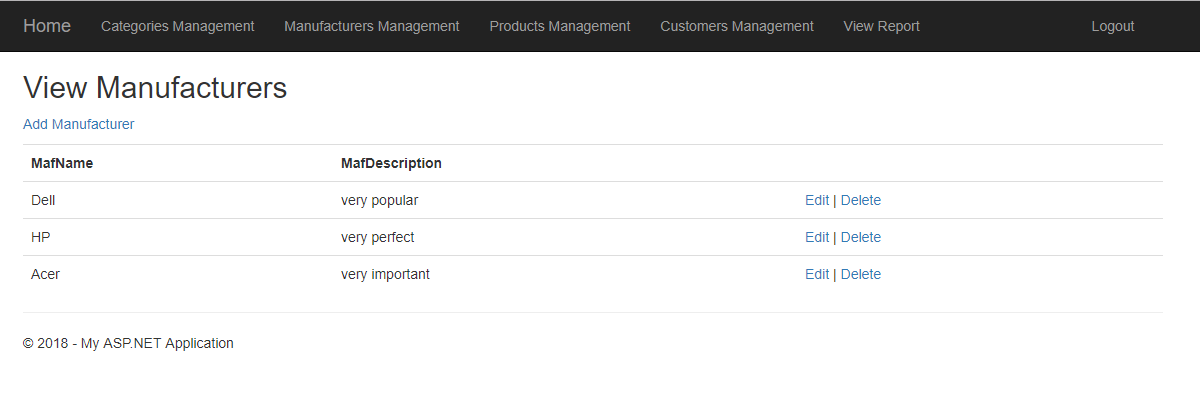


Figure 28. Interface Management Manufacturers

1. Management Products

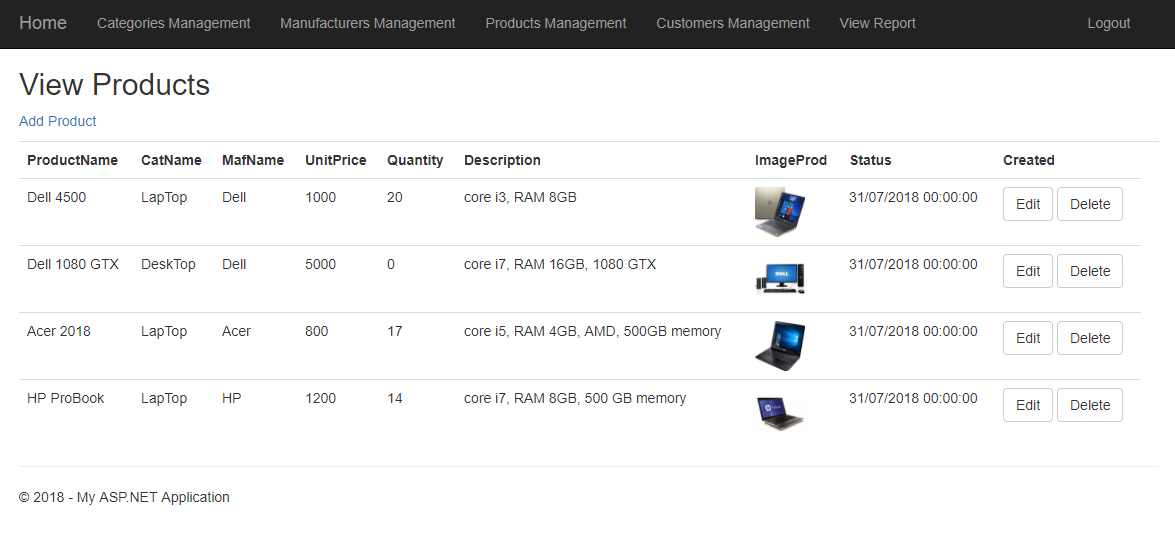


Figure 29. Interface Management Products

1. Create product from Managers

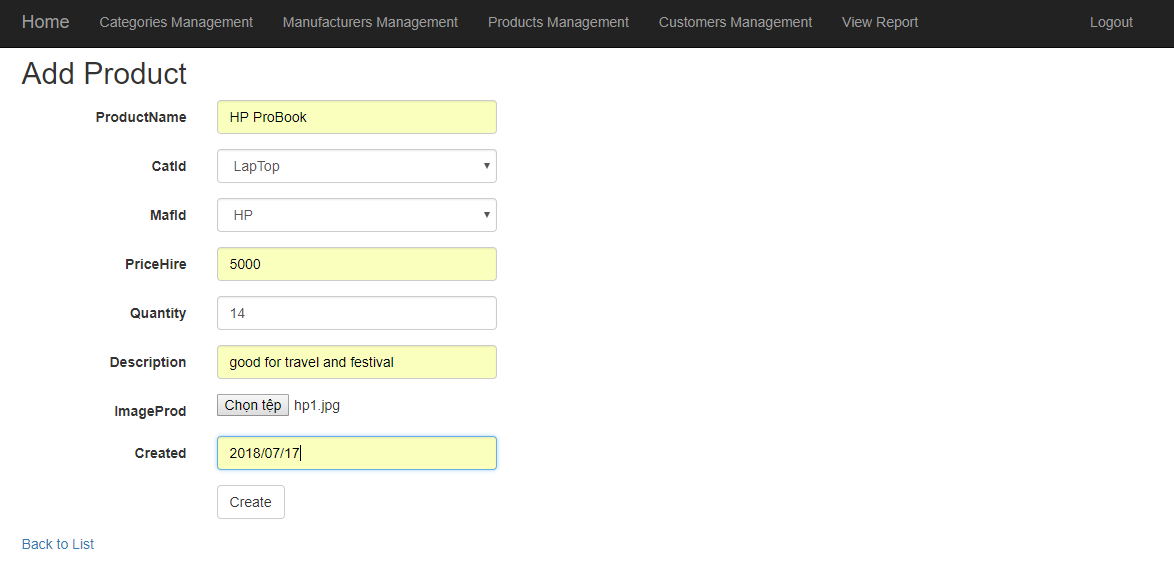


Figure 30. Interface Add new product

1. Update products from Managers

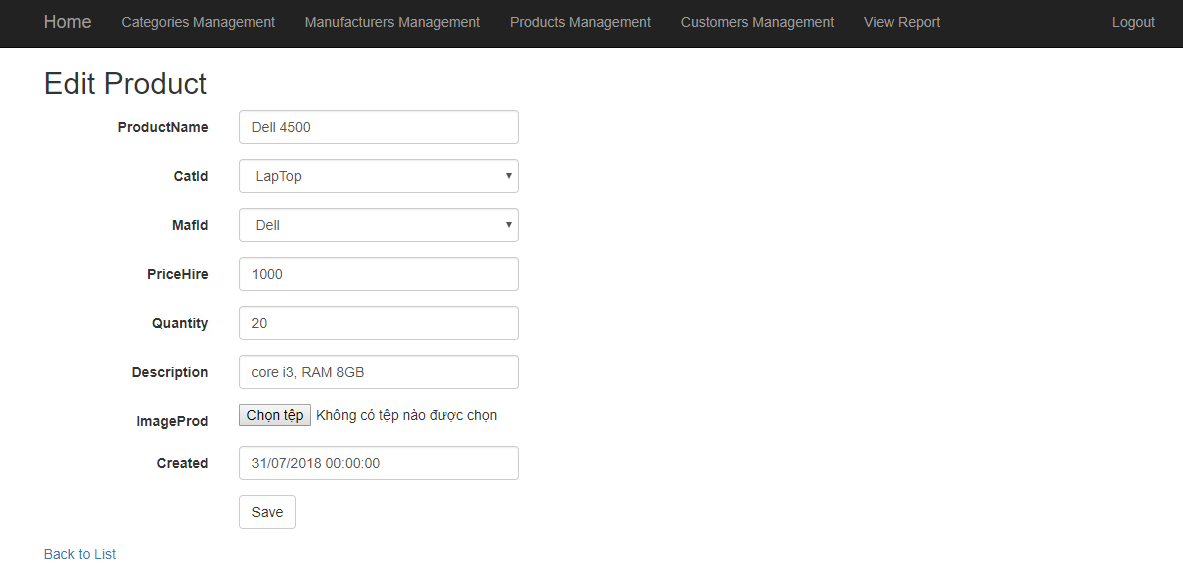


Figure 31. Interface Update product

1. Management Customers

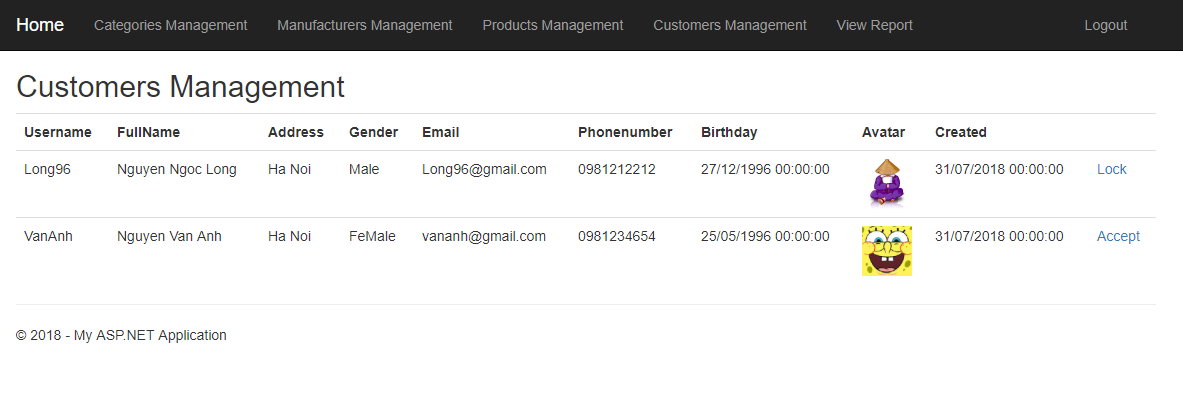


Figure 32. Interface Management Customers

1. View report for Managers

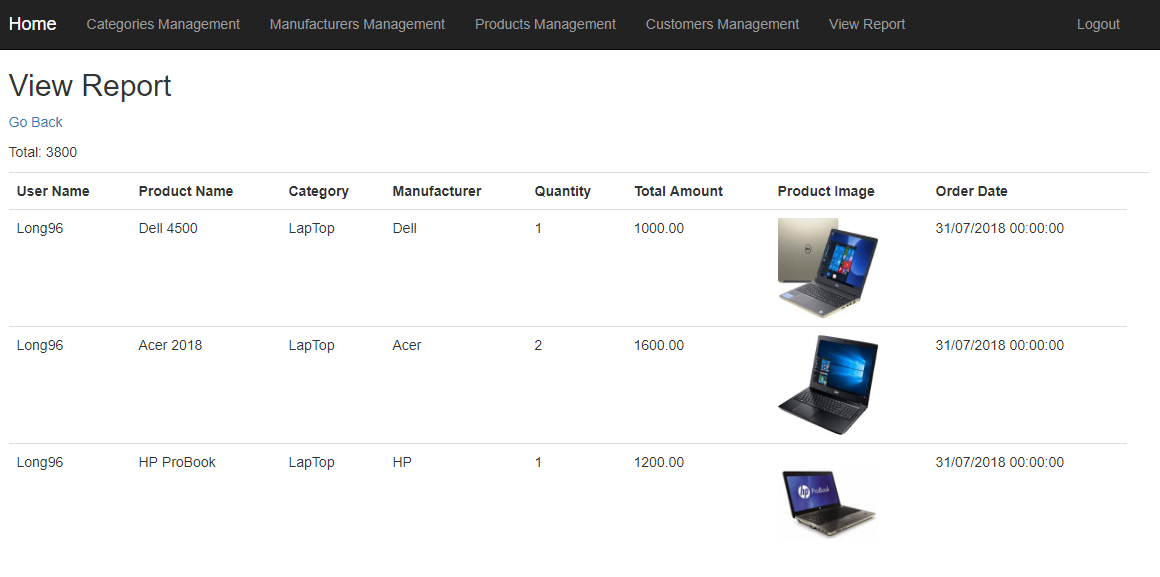
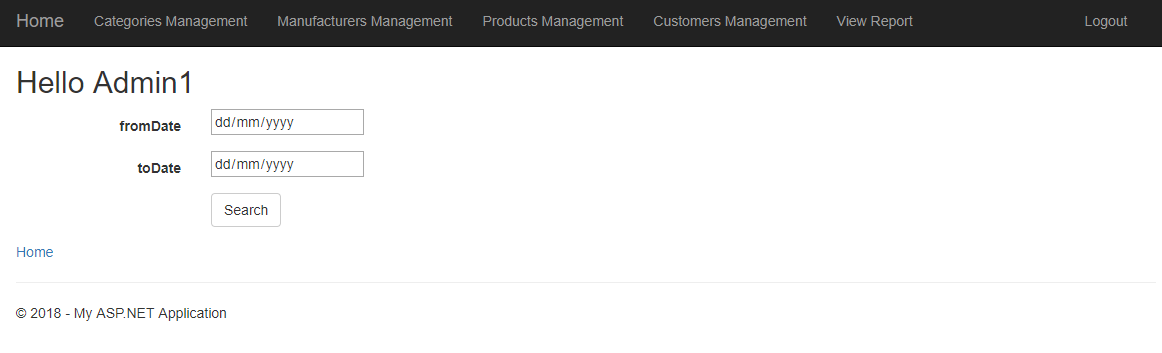


Figure 33. Interface View Report