



VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

AKM Mahfuzul Haque

# E-COMMERCE WEB APPLICATION USING PAYPAL PAYMENT GATE- WAY

Technology and Communication

2017

## **ACKNOWLEDGEMENT**

I would like to thank God for showing me the way of my success. I am indebted to my parents (Mr. Karam Ali and Mrs. Khaleda Akhter) for their significant effort to support me all the way from my childhood until now. Their happiness is my success. My deepest gratitude goes to my thesis supervisor Dr. Ghodrat Moghadampour who has given his best to help me to complete my thesis. His guidance is excellent, and without his assistance, it would be quite difficult to fulfill my thesis. It is an honor to thank the Head of Degree Programme Dr. Seppo Mäkinen who is always friendly and helpful to me. It is my pleasure to thank all the teachers and staffs of Vaasa University of Applied Sciences who has made a great support for my whole study period.

Haque AKM Mahfuzul

May 2017

VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES  
INFORMATION TECHNOLOGY

## **ABSTRACT**

Author	AKM Mahfuzul Haque
Title	E-commerce Web Application Using PayPal Payment Gateway
Year	2017
Language	English
Pages	82
Name of Supervisor	Dr. Ghodrat Moghadampour

---

The main idea of the thesis was to develop an E-commerce Web Application by using payment gateway (PayPal payment technology) for an online shop which was developed by ASP.NET framework (version 4.6.2) and Microsoft SQL server management studio, where a seller (Owner of the Web shop) will be able to put all the products in the online shop, and a customer who is a buyer will be able to buy items through online payment. This application has two different views: Public view and Administrator (in short Admin) view. Admin view refers to the owner of the online shop who can manage the application as well maintain products and customers. The public view is made for clients or users for viewing and ordering products.

A user can search and view items from the online store. Only a registered user can buy products from the shop by using PayPal payment technology. Without the registration, it is not possible to check out through PayPal payment. This application has registration and login features for a user. A registered user can manage the own account. Registered customers can see their purchase history and can update profile information from their own account page.

Administrator of the web shop can add, edit and delete products from the store. He can update and remove a registered customer's information. As well, all the orders made by the customers will be available to the admin. Admin can update shipping information. Item stock and sale report will be shown to the admin.

---

KEYWORDS                      ASP.NET, SQL, E-COMMERCE, PAYPAL

# CONTENTS

## ABSTRACT

1 INTRODUCTION .....	1
1.1 Background .....	1
1.2 Objectives.....	2
2. RELEVANT TOOLS AND TECHNOLOGIES.....	3
2.1 ASP.NET.....	3
2.2 SQL Server Management Studio .....	3
2.3 SQL Database .....	4
2.4 C# Programming Language .....	4
2.5 HTML5.....	4
2.6 CSS3.....	5
2.7 PAYPAL Technology.....	5
2.8 Visual Studio.....	5
3. APPLICATION DESCRIPTION .....	6
3.1 Application Main Modules.....	6
3.1.1 Administrator Interface .....	6
3.1.2 User/Customer Interface .....	7
3.2 REQUIREMENT ANALYSIS.....	10
3.2.1 Requirements Specification .....	10
3.2.2 Use Case Diagram Analyse .....	11
3.2.3 Class Diagram of the project.....	15
3.2.4 Sequence Diagram of the project .....	20
3.2.5 Deployment Diagram of the project .....	26
3.2.6 Component Diagram of the project.....	27
3.2.7 Architectural Design of The System .....	30
4. DATABASE AND GUI DESIGN .....	31
4.1 Database Design.....	31
4.2 Graphical User Interface(GUI) Design .....	35
4.2.1 Home Page Design .....	36
4.2.2 User Registration Page Design .....	37
4.2.3 Login Page Design .....	38

4.2.4 Shopping Cart Design .....	40
4.3 Admin Interface (GUI) Design .....	49
4.3.1 Admin Registration Page.....	50
4.3.2 Admin Login Page.....	51
4.3.3 Admin Management Page .....	52
5. IMPLEMENTATION .....	53
5.1 Description .....	53
5.2 Implementation of Functions.....	53
5.2.1 Available Products in The System .....	54
5.2.2 User Registration .....	60
5.2.3 User Login .....	61
5.2.4 Payment Through PayPal .....	63
5.2.5 Mail Transport System .....	67
5.2.6 Search Products by Keyword .....	69
5.2.7 View Orders by Admin .....	70
6. APPLICATION TESTING .....	72
6.1 User Login Testing.....	72
6.2 User Shopping Cart Testing.....	73
6.3 Admin Login Testing.....	75
6.4 Adding Products Testing .....	77
7. CONCLUSIONS .....	80
7.1 FUTURE WORKS .....	81
REFERENCES.....	82

## LIST OF FIGURES

<b>Figure 1.</b> Admin Panel Use Case Diagram .....	12
<b>Figure 2.</b> Registered User Use Case Diagram.....	13
<b>Figure 3.</b> New User Use Case Diagram .....	14
<b>Figure 4.</b> Class Diagram of the entire project. ....	15
<b>Figure 5.</b> ShoppingCart Class diagram .....	16
<b>Figure 6.</b> Customer Class diagram .....	17
<b>Figure 7.</b> Administrator class diagram .....	18
<b>Figure 8.</b> Orders Class diagram.....	19
<b>Figure 9.</b> OrderDetails Class diagram .....	19
<b>Figure 10.</b> NVPCaller Class diagram .....	20
<b>Figure 11.</b> Sequence Diagram of the entire system.....	21
<b>Figure 12.</b> User Registration Function Sequence Diagram.....	23
<b>Figure 13.</b> Customer Login Function Sequence Diagram.....	24
<b>Figure 14.</b> Administrator Login Function Sequence Diagram .....	25
<b>Figure 15.</b> Deployment diagram of the system .....	26
<b>Figure 16.</b> Component diagram of the application.....	27
<b>Figure 17.</b> Admin Management Module Diagram .....	28
<b>Figure 18.</b> Customer Module Diagram .....	29
<b>Figure 19.</b> Project Architecture of the application .....	30
<b>Figure 20.</b> Database structure of the system .....	32
<b>Figure 21.</b> Customer details database table .....	33
<b>Figure 22.</b> Store procedure for SaveCustomerDetails.....	34
<b>Figure 23.</b> Registration database table .....	35
<b>Figure 24.</b> Home Page Design.....	36
<b>Figure 25.</b> User Registration Page Design .....	37
<b>Figure 26.</b> Customer Login Page Design .....	38
<b>Figure 27.</b> Successful Login View Design.....	39
<b>Figure 28.</b> Shopping Cart Design .....	40
<b>Figure 29.</b> Payment and Transaction Design Order Summary.....	41
<b>Figure 30.</b> Payment and Transaction Design .....	42
<b>Figure 31.</b> Payment by PayPal Account.....	43
<b>Figure 32.</b> Payment with Debit/Credit Card Part1 .....	44
<b>Figure 33.</b> Payment with Debit/Credit Card Part2 .....	45

<b>Figure 34.</b> Order Review Design.....	46
<b>Figure 35.</b> Complete Order Design .....	47
<b>Figure 36.</b> Email Order Notification Design view .....	48
<b>Figure 37.</b> Complete Order Database Design .....	49
<b>Figure 38.</b> Admin Registration Page Design.....	50
<b>Figure 39.</b> Admin Login Design .....	51
<b>Figure 40.</b> Admin Management View Design .....	52
<b>Figure 41.</b> Login Test Case Without Username and Password.....	72
<b>Figure 42.</b> Login Test Case Invalid Username and Password.....	73
<b>Figure 43.</b> Test Result after Adding Product to the shopping Cart.....	74
<b>Figure 44.</b> Test Result after Deleting Products from Shopping Cart .....	75
<b>Figure 45.</b> Login Test without Login Credentials .....	76
<b>Figure 46.</b> Invalid Login Attempt By Admin.....	76
<b>Figure 47.</b> Available Products from Database .....	77
<b>Figure 48.</b> Test Case Products Added By Admin .....	78
<b>Figure 49.</b> Products Added by Admin.....	79

# 1 INTRODUCTION

E-commerce means an electronic form of buying and selling process over the internet using common web browsers. Online shopping is getting so popular nowadays because the consumer can pay from home, products are cheap, and goods are delivered at home without any hustle. Developed countries are more familiar with E-commerce systems whereas online shopping is growing rapidly to the developing countries. /1/

An entrepreneur first invented E-commerce shopping idea in 1979. Before that people were used to buying products visiting the shops. Technology is growing at a rapid pace. Developers are working days and night to make life easier. They are building technologies which provide easy solutions to the consumers of this century. Consumers do not need to go to the shops to do their shopping. They can choose their own way of shopping. In Ecommerce shopping the consumer browse a web page, buy an item using a payment gateway. Seller and the buyer both will get an email confirmation of the purchase. Items will be delivered to the destination address after receiving the order. This is called Business to Consumer system(B2C).

/2/

## 1.1 Background

Ecommerce web-shops are getting popular nowadays in the developing countries. Many people are interested in getting their own online website for the business, as it is possible to sell products over the internet around the world. Consumers are also curious at online shopping because they do not want to waste their valuable time for shopping. This e-commerce web shop is designed and implemented by a real-time payment technology called PayPal payment technology, where a consumer can use a debit or credit card authorized by PayPal payment technology to pay for the products. The product is shipped to the buyer's address after a successful transaction.



## **1.2 Objectives**

The purpose of this application is to build an e-commerce store where a seller can sell his goods over the internet, and a buyer is able to buy items using PayPal payment gateway. This application contains administration (Admin in short) and consumers. The owner of the website is the Admin. Admin will have a login page. Login credentials will be known to the Admin by default. Customers are not able to view the Admin section. Admin can add, update and delete products and customer's information from the web shop. Products stock availability will be shown to the Admin. Admin can check the sales report daily, monthly and yearly basis. Admin can update the shipping status of a successful purchase.

The user can browse through the web store over the internet. He can browse the available products in the shop by searching through the search box or by category. After a successful login, the user can order a product from the shop. To log in to the webshop, he must sign up first. Registration details will be saved to the account page of the user. He can update his details after a successful login. A customer can track his order status of a product. The user must create a PayPal account by using their debit or credit card to buy a product from this web shop. After a successful transaction, the user will get an email confirmation from the seller containing a transaction number and an order number. By using order number, the user will be able to track the items' shipping status from the website.

## **2. RELEVANT TOOLS AND TECHNOLOGIES**

This project is built with Visual Studio 2015, ASP.NET Framework (version 4.6.2), C# programming language, SQL Server 2014 Management Studio, CSS, HTML5, PAYPAL TECHNOLOGY

### **2.1 ASP.NET**

ASP refers to Active Server Pages. ASP.NET is used as an open source framework for server-side web application to build dynamic web pages, web services, and web applications which are a part of .NET Framework. In January 2002 Microsoft first introduced .NET Framework version 1.0. ASP.NET allows the programmer to write code by using any ASP.NET supported programming languages such as VB.NET, C#. ASP.NET web forms are used to build ASP.NET applications. This framework is slightly HTML based. ASP.NET Framework 4.6.2 is the latest version of .NET Framework. An extension of .asp is used for ASP.NET web forms. ASP.NET communicate with the Internet Information Server(IIS) to provide contents in a request to the client-side programs. /3/

### **2.2 SQL Server Management Studio**

SQL Server Management Studio (SSMS) is used as an integrated environment to manipulate SQL infrastructure. It manages SQL system from SQL server to SQL Database. It is an open source tool introduced by Microsoft in 2005. Tools provided by SSMS are used to manage, configure and administer SQL instances within Microsoft SQL Server. SSMS uses components to upgrade, deploy and develop the data-tier components to build scripts and queries. /4/

### **2.3 SQL Database**

SQL stands for Structured Query Language which refers to a domain specific language used to manipulate, store and retrieve data which is stored in a relational database system. SQL database is used as a standard language for many relational database management systems such as MySQL, Oracle, and SQL server management system. SQL is a set of data manipulation language, data definition language, and data control language. /5/

### **2.4 C# Programming Language**

C# is an object-oriented, general-purpose programming language which allows developers to develop a variety of applications as secure and robust to run on the .NET Framework. It is designed for the Common Language Infrastructure. C# is used to build Database applications, Windows client applications, client-server applications, XML Web services, distributed components and so on. Visual C# has a convenient UI designer, an advanced code editor, integrated debugger and a lot of tools to build applications easily based on C# programming language and .NET Framework. /6/

### **2.5 HTML5**

HTML5 stands for Hypertext Markup language is the latest version of HTML. It is used to structure and publish contents like web pages on the World Wide Web (WWW). Nowadays HTML5 is supported by most of the web browsers. It is a standard language which can be used for web applications with all the programming languages. It has a variety of new attributes, new features, 2D/3D graphics, audio and video support, new HTML elements, full CSS3 support which makes it easier for users and web developers to develop web applications. /7/

## **2.6 CSS3**

CSS3 stands for Cascade Style Sheet is the latest release of style sheet language. It is used to describe and to set the layout of web pages, user interfaces for many web and mobile applications. Most of latest browsers support it. It has fonts, layout, and colors. World Wide Web Consortium(W3C) maintains the specifications of CSS. /8/

## **2.7 PAYPAL Technology**

PayPal is an American E-commerce company which allows any individual or any business in a safer as well in a convenient way to send and receive money using an email address through online payment systems. PayPal gives authorization to all individuals to create a personal or business account on its site, which connects to an existing bank account or debit cards or credit cards of a customer. PayPal is now widely used for most of the e-commerce sites for selling products like household and electronics items, digital contents and so on. PayPal is promoted as a money transfer company in 1999, before that it was security software developing company. Nowadays it is the most popular payment technology to the online users. A user does not need to pay for transferring money from PayPal account. /9/

## **2.8 Visual Studio**

VISUAL STUDIO is an IDE used to build apps for different devices or desktop apps, in the cloud and on the web. It is utilized to write code for Android, Windows, and iOS using this IDE. Easy code navigation, great IntelliSense, quick builds and fast deployment are common features of this IDE.

### **3. APPLICATION DESCRIPTION**

The requirements of the application depend on the system analysis. UML (unified modeling language) is used to draw and visualize the architectural design of the whole projects. UML is a standard and unique representation. The whole project is described as many different modules as well all the requirements are analyzed within this section.

#### **3.1 Application Main Modules**

This e-commerce web application has two main modules: one is Admin module which is a short form for administrator, and another is Customer/user module. Admin module has many essential functions. Login, adding/adding/deleting products, adding/adding/deleting registered customers' information, checking products stock, updating order status, adding a new category, checking customers order and sale reports are the pages which are used for Admin module. Customer/user module is the front-end interface. This module also includes many key features.

##### **3.1.1 Administrator Interface**

Administrator module is the system's management module which provides an interface to manipulate Admin control and client control. Management control of this system allows the admin to add products, to update product information, to remove products from the database, to update or delete customer information, to check products' inventory, to get sale report and all the orders made by the registered customers. It is also known as back-end module.

###### **3.1.1.1 Administrator Login Module**

USERNAME and PASSWORD are the mandatory fields to log in for the admin view module. Username and password are pre-set for this system. Only admin can login to the back-end system by using login credentials.

### **3.1.1.2 Members Management Module**

This module includes managing all the registered customers' information by the admin of the application. System admin can view all the registered clients' information. Admin can remove any customer if the account of the client is not active for a long time. Admin can update the information of customers on the request from a customer.

### **3.1.1.3 Product Management Module**

This module allows an admin to manage all the information related to products. An admin can add products, edit products information and remove products from the database. A stock inventory of the products is made in this module to count products' availability after every successful order completion.

### **3.1.1.4. Sale Report Module**

A daily, monthly and yearly sales report will be available in this module. It is an easier way to the seller to get informed about the sales statistics.

### **3.1.1.5. Order Details View Module**

This module provides all the payment details made by the customers. It is possible to view the purchased items, quantity, price, and shipping address.

### **3.1.1.6. Admin Sign Out Module**

This function is used for an Admin to log out of the system. Admin needs to log in again to enter the admin management system.

## **3.1.2 User/Customer Interface**

This is the front-end module of this scheme. It allows a user to browse products, view products, and purchase products by registering to the system. Many sub-modules involved under this module will be discussed below.

#### **3.1.2.1. Registration Module**

A user can register himself as a registered member with this module. A registration form will be shown to the user after clicking SIGN UP button. The user must fill all the mandatory fields to complete the registration process. Only a registered member can purchase products from this shop.

#### **3.1.2.2. Login Module**

A registered member of the system will be able to log into his own account by using his account credentials such as username and password. This module also provides a link to the customer to reset his username and password if he does not remember his login credentials.

#### **3.1.2.3. Browsing and Searching Module**

Any user has the right to browse and search on the websites for the desired products. The user can view all the goods on the page searching by keywords or by categories.

#### **3.1.2.4. Track Order Module**

A registered user can track his order status by an order id. Shipment status will be available to him after the product is delivered from the store. On the other hand, Admin can update shipping status after a successful purchase from the admin panel.

#### **3.1.2.5. Edit Profile Module**

This function allows to the registered users to edit their personal information such as address, email address, phone number, changing the password and so on. This feature will be available only after a successful login.

#### **3.1.2.6. Purchase History Module**

Any registered member will be able to view all the purchase details in this module. After a successful login, the customer will be able to check order history.

### **3.1.2.7. Checkout Module**

This module is used only for the logged in customers. Checkout module connects to a payment gateway. This system uses the PayPal checkout process. The customer can add or remove products before checkout. Customers can pay with PayPal account or debit/credit cards processed by PayPal.

### **3.1.2.8. Shopping Cart Module**

The shopping cart is the place where a user can view all the products and product details, he has added from the online shop. He will be able to add or modify products quantities in the shopping cart. By default, the shopping cart is empty. The user must add an item to the shopping cart to checkout. The shopping cart will display a review section in which the user can check the purchase price and products' quantity. Every user must register before buying products.

### **3.1.2.9. Logout Module**

A registered user will be able to log out of the system by clicking the logout button. After a logout process, the user will not be able to buy products from the store without the login again.

### **3.1.2.10. Contact Module**

Contact support displays all the contact details to make contact with the seller or the responsible person for this online shop. A contact request will be answered within three working days by the seller.



## **3.2 REQUIREMENT ANALYSIS**

It is important to analyze all the requirements needed for the application to understand, design and implement the project

### **3.2.1 Requirements Specification**

This section will give a clear description of the diverse types of requirements needed for the project. Two types of requirements are discussed here: Must have requirements and basic requirements.

Must have requirements for a registered customer:

- Login with username and password
- Buy products using PayPal account
- Updating personal information
- Changing password
- Viewing personal information includes card information and purchase history
- Email notification of a successful order completion

Must have requirements for Admin:

- Login with username and password
- Add/edit/delete products from store
- Edit/delete customer information
- Add/delete category from store
- Update order status
- View order details
- Checking products stock
- Checking Sale report

Basic requirements for the users (not registered in the system)

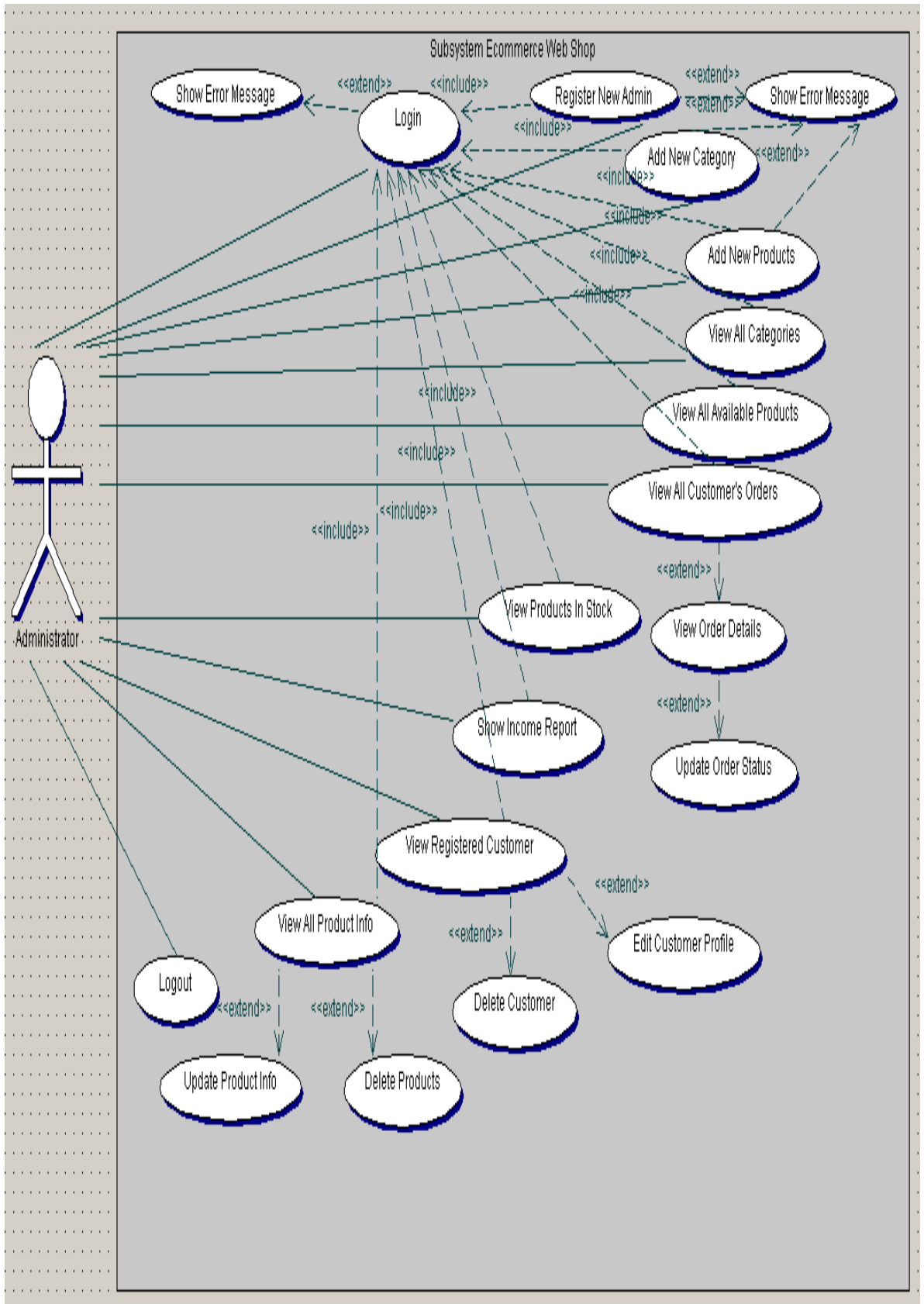
- Search items by keyword
- View products by category
- Sign up as a new customer
- Contact the seller
- Add items to the shopping cart
- Delete items from the shopping cart
- Tracking Order status

### **3.2.2 Use Case Diagram Analysis**

In a system analysis, a developer could show all the actors involved in the project using use case diagram. This project has two main phases: Admin interface and User interface. These two interfaces will be analyzed by use case diagram.

#### **3.2.2.1. Admin Panel**

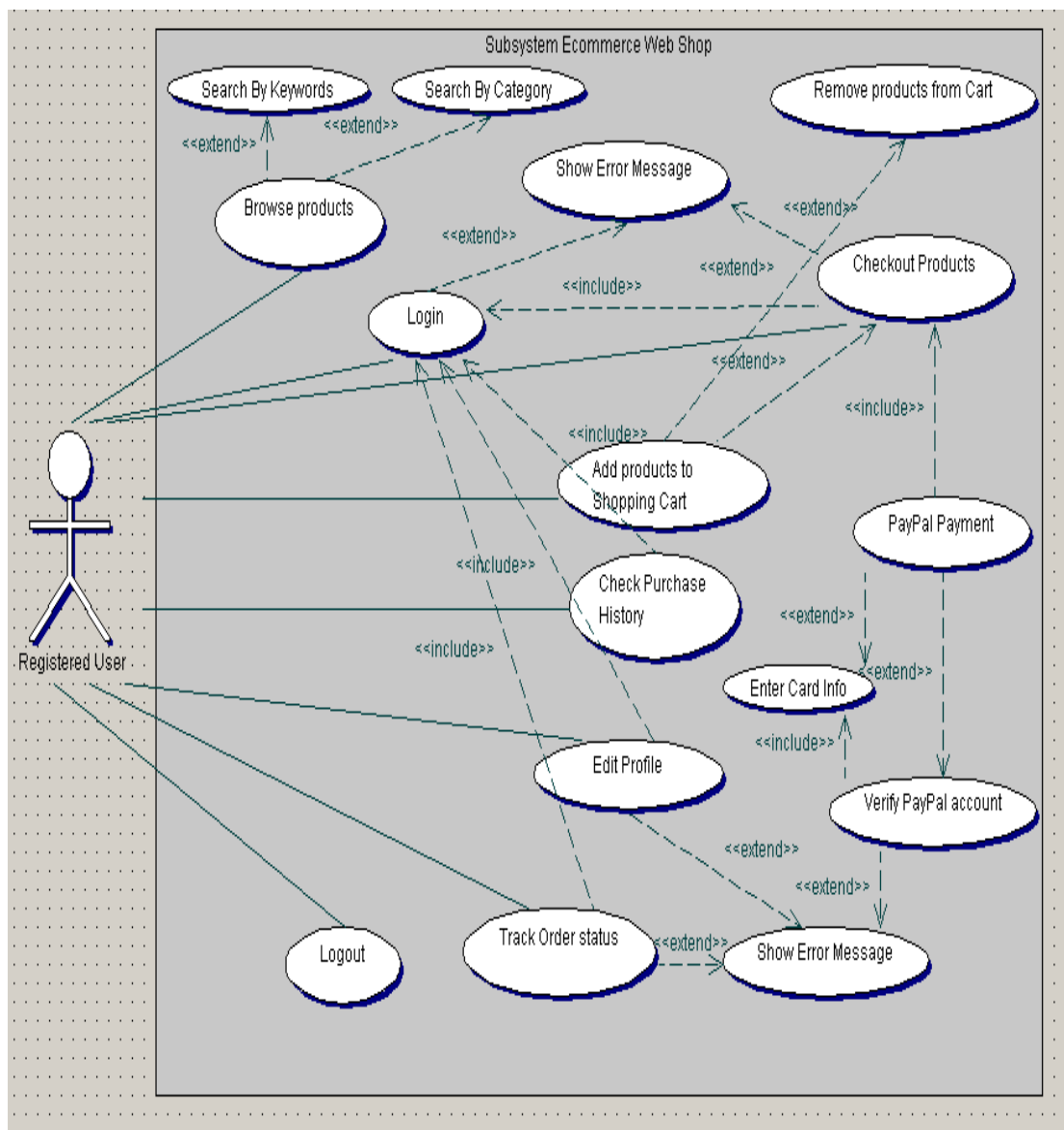
Figure 1 describes the Use Case Diagram of the administrator panel. An admin will be able to login with his username and password. He can add or edit items into the system. He can also remove products from the server. He will be able to update and delete customer's information. He can see products' stock and availability, all the transactions made by the customers, and sale report for a daily, monthly, and yearly basis. He will be able to add new category into the server. He can update the shipping status of a purchase. He will be able to log out from the system.



**Figure 1.** Admin Panel Use Case Diagram

### 3.2.2.2. Registered User

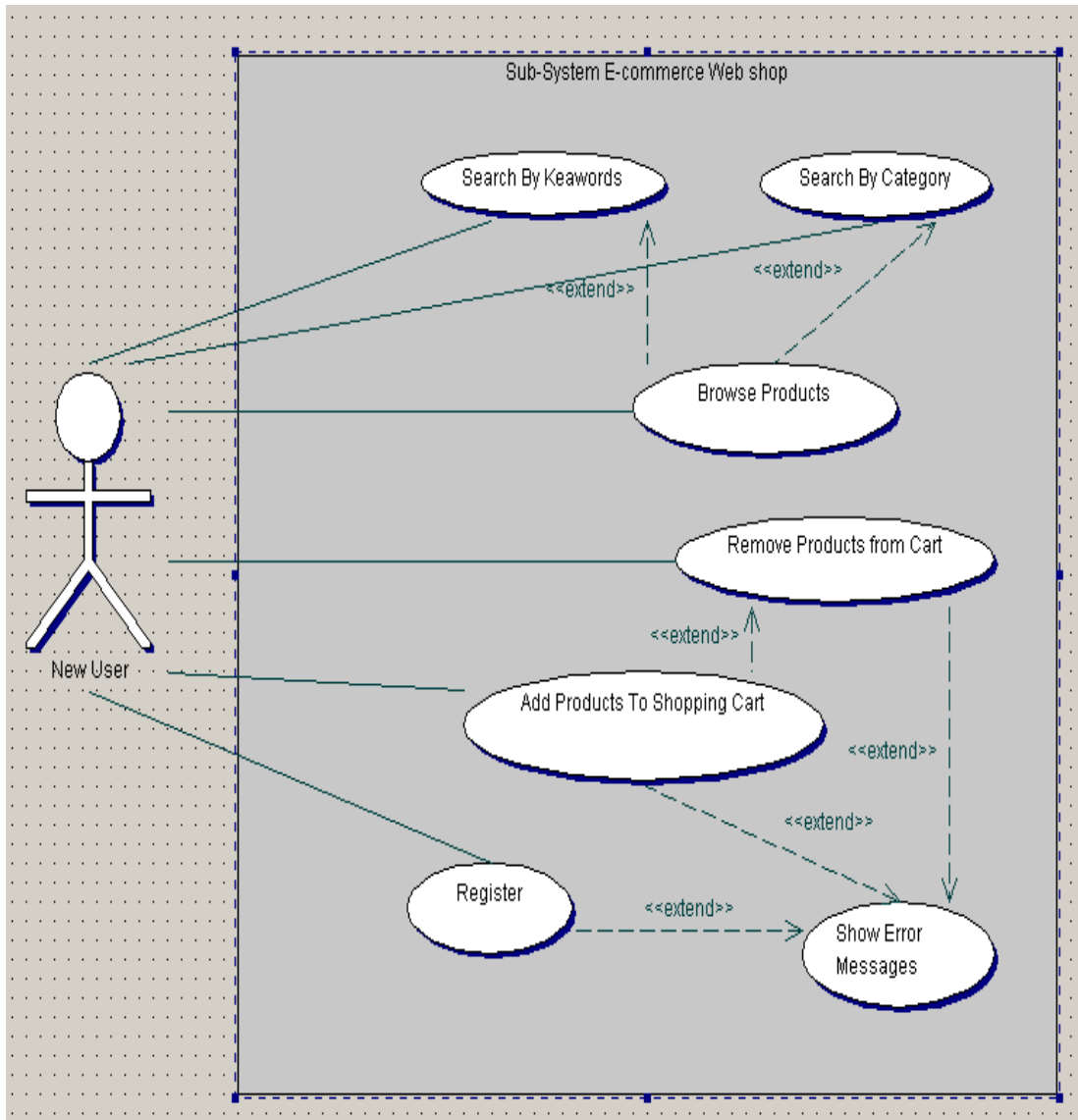
Figure 2 describes the Use Case Diagram of the registered user. A client or a registered user can do the same like a new user as well he will be able to log in, maintain a personal account which includes updating personal information, change the password, check purchase history. A customer can track the order status of a product if he has a transaction number.



**Figure 2.** Registered User Use Case Diagram

### 3.2.2.3 New User

Figure 3 describes the use case diagram of a new user. A new user can browse products, search products by category or by keywords, and add products to the shopping cart. But a new user cannot purchase an item without the registration. A browser can sign up as a new member of the system.



**Figure 3.** New User Use Case Diagram

### 3.2.3 Class Diagram of the project

Figure 3 shows the class diagram of the application. It shows the declaration of every parameter and functions of different classes of the system, and it also defines the relation between the classes. The purpose of this diagram is to model the static structure of this system.

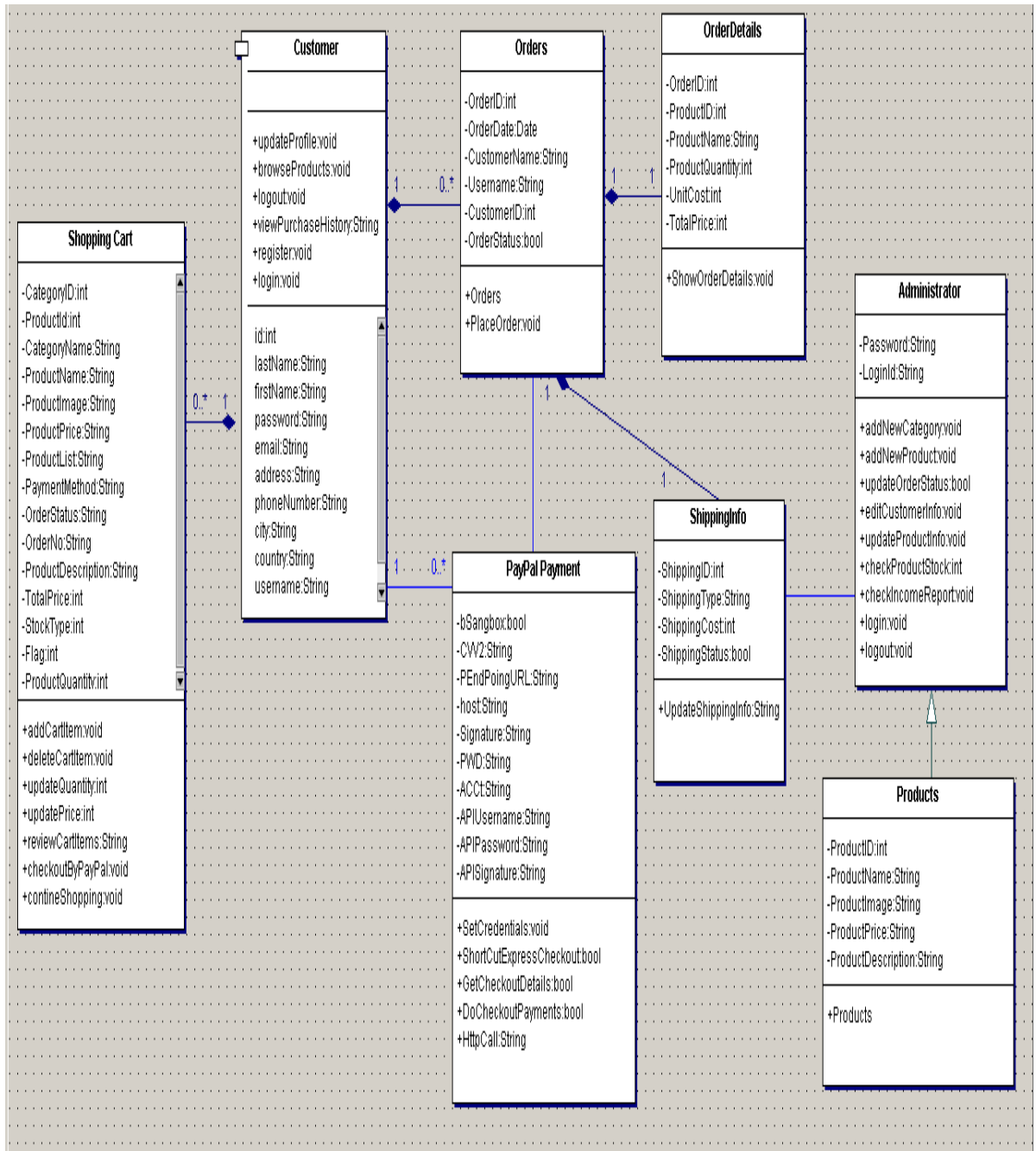
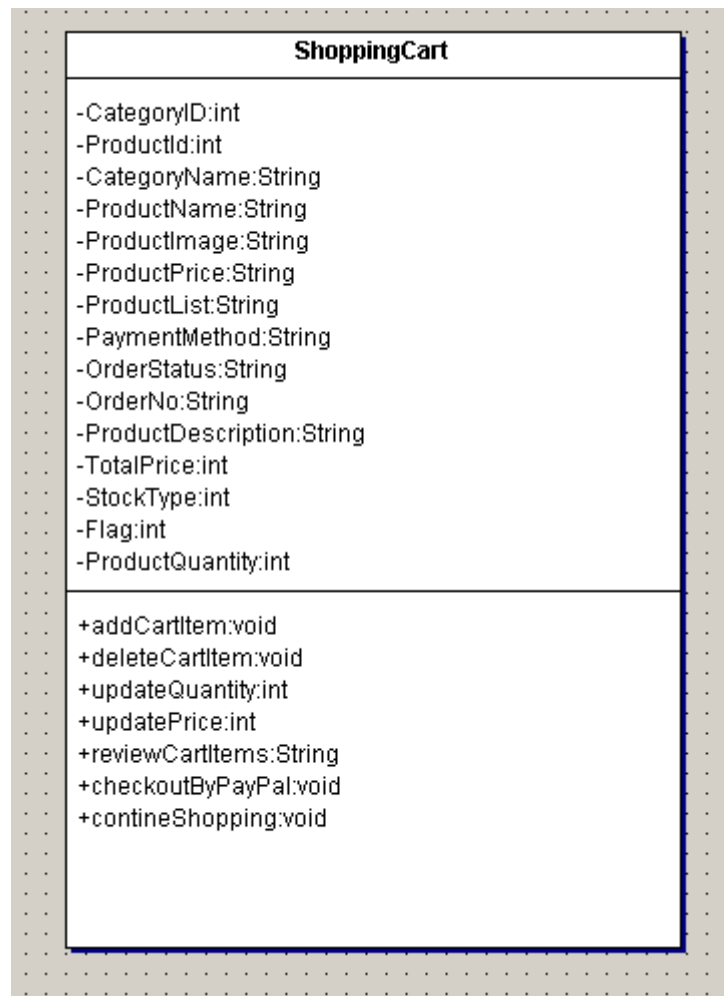


Figure 4. Class Diagram of the entire project.

All the important class diagrams are described below in short:

### 3.2.3.1 ShoppingCart Class

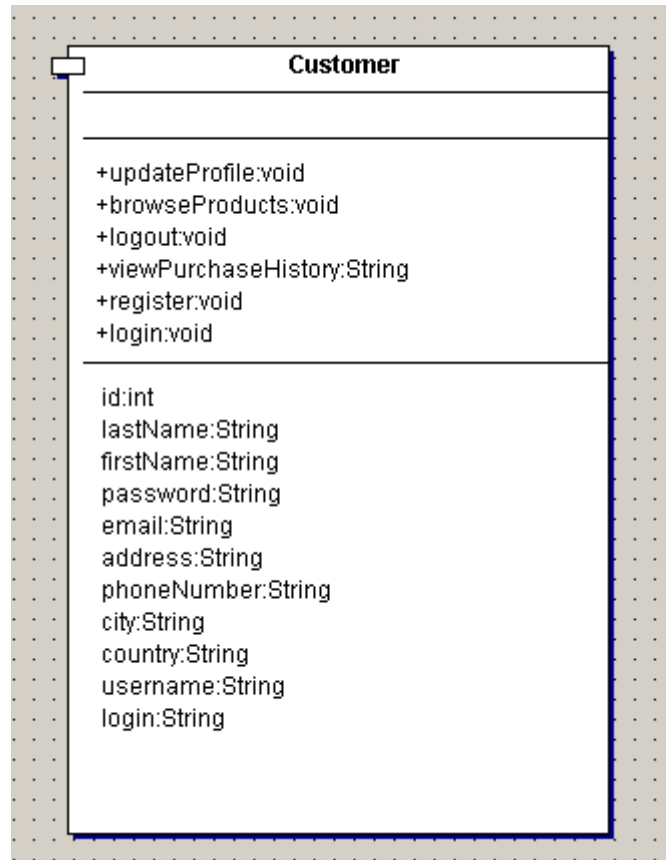
Figure 5 shows the attributes and the operations of the ShoppingCart class. A ShoppingCart class includes methods which are: adding products to the cart, removing items from the cart, updating cart products and prices, reviewing cart items and prices, checking out by PayPal method, and continuing shopping without check out.



**Figure 5.** ShoppingCart Class diagram

### 3.2.3.2 Customer Class

Figure 6 shows the attributes and the operations of the customer class. Customer class consist of functions which are: Updating profile information, browsing products by keywords or by category, logging in, logging out, registering as a member of the system, viewing purchase history.

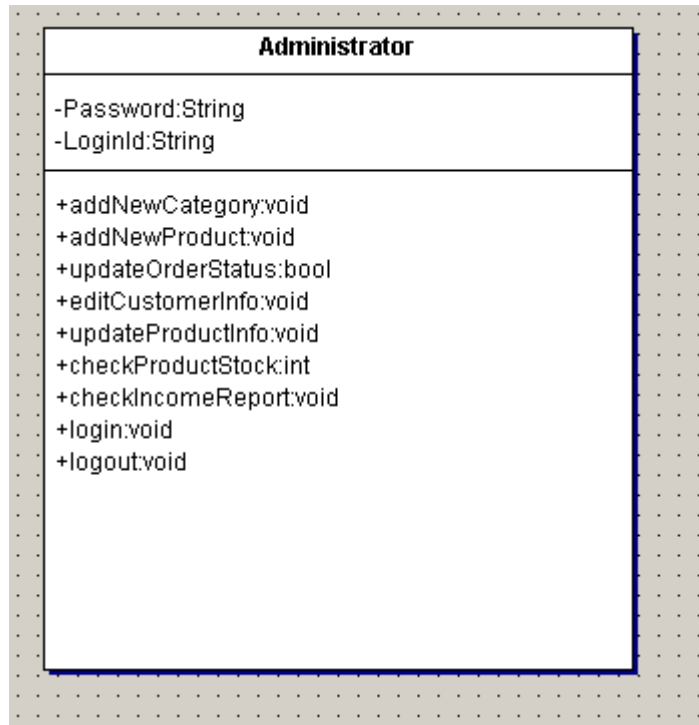


**Figure 6.** Customer Class diagram

### 3.2.3.3 Administrator Class

Figure 7 shows the attributes and the operations of the administrator class. This class includes methods which are: adding new product and category to the system, editing or deleting customer information, updating product information, checking products' stock and availability, checking income report in daily, monthly, or yearly basis, logging in, logging out, and updating shipping status.

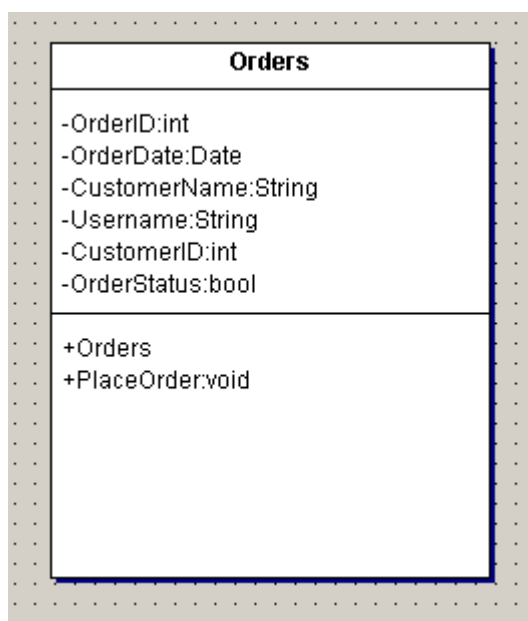




**Figure 7.** Administrator class diagram

### 3.2.3.4 Orders Class

Figure 8 shows the attributes and the operations of the orders class. This class includes method which is responsible for placing an order by a customer in the application.



**Figure 8.** Orders Class diagram

### 3.2.3.5 Order Details Class

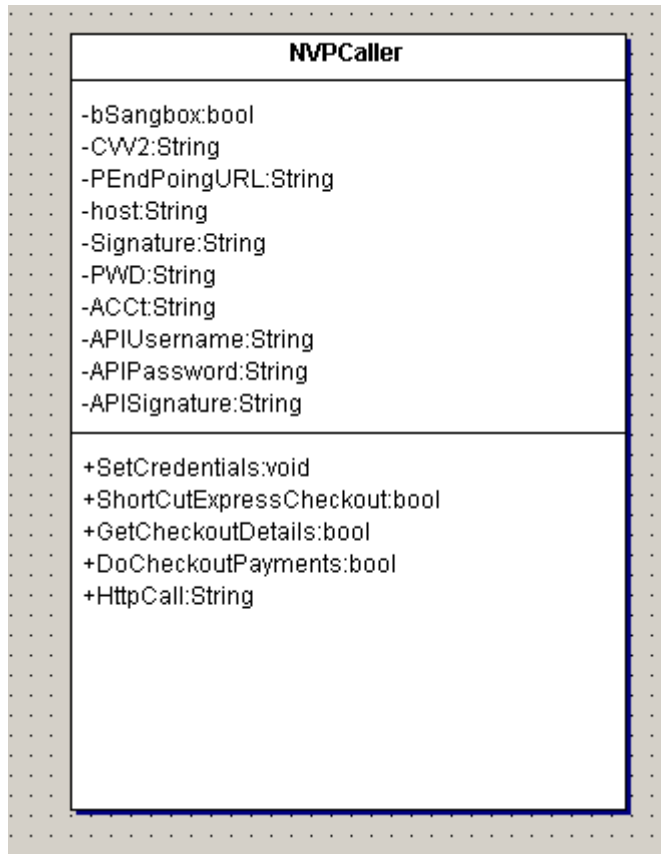
Figure 9 shows the attributes and the operations of the Order details class. This class has a method called ShowOrderDetails which shows the details of the order when a customer buys a product.



**Figure 9.** OrderDetails Class diagram

### 3.2.3.6 NVPApiCaller Class

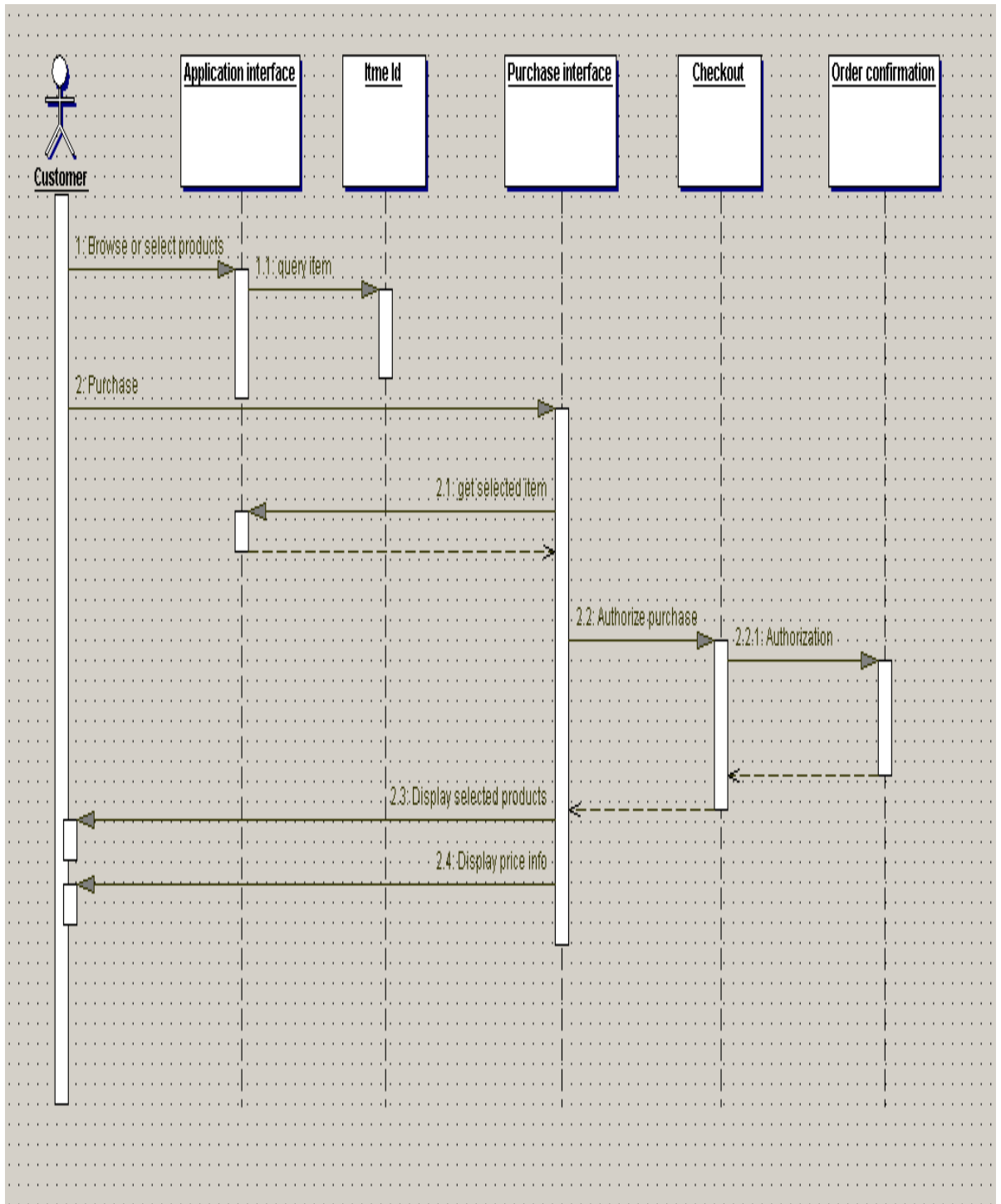
Figure 10 shows the attributes and the operations of the NVPApiCaller class. It has four essential functions which make communication to the PayPal site. This class make a connection to the PayPal payment gateway, and all the payment related methods are executed in this class.



**Figure 10.** NVPCaller Class diagram

### 3.2.4 Sequence Diagram of the project

Figure 11 describes the sequence diagram of this application for the front-end interface where a customer will be able to purchase products through a payment gateway.

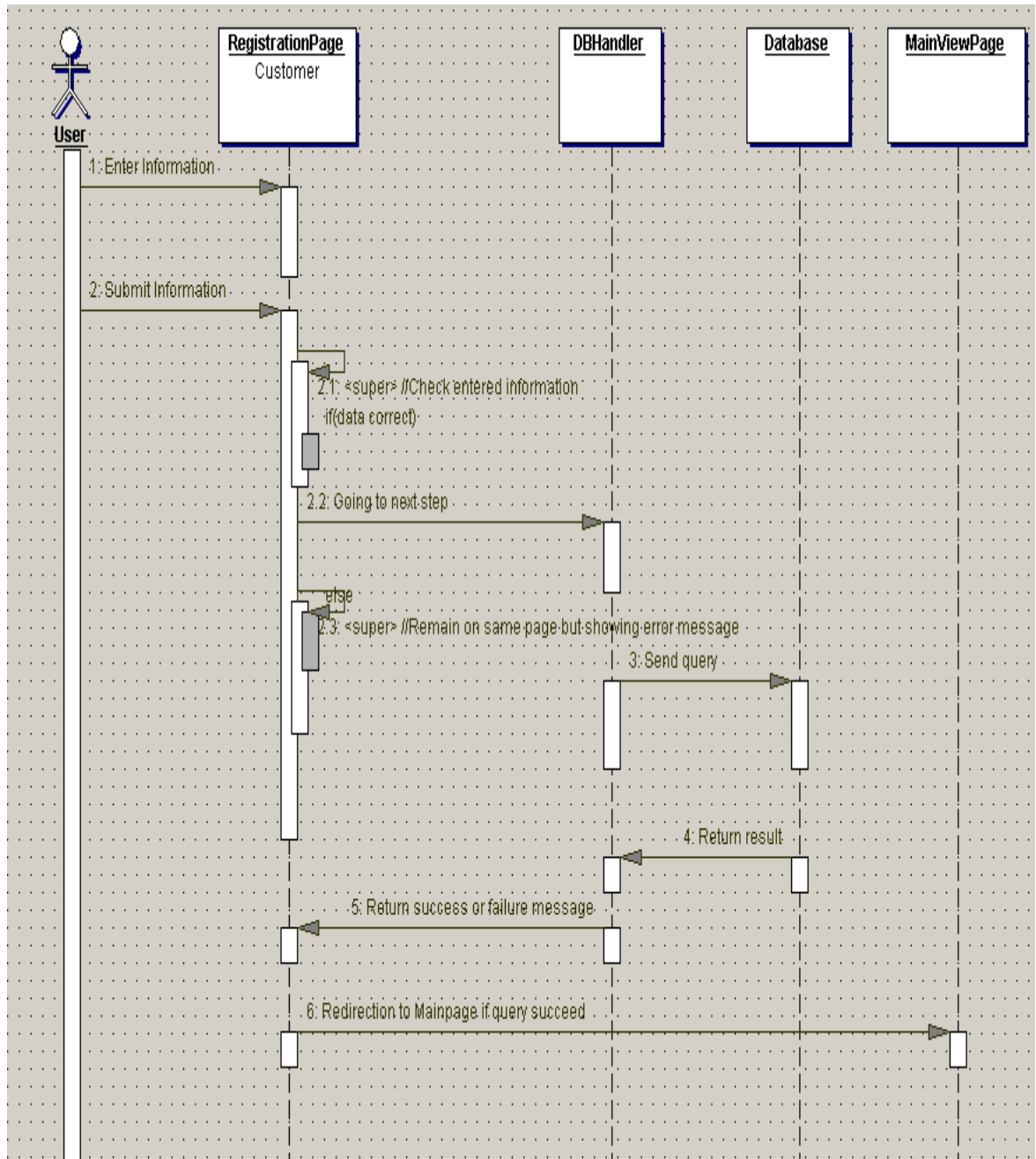


**Figure 11.** Sequence Diagram of the entire system

Figure 12-14 displays the Sequence diagrams of few functions of this project.

#### **3.2.4.1. User Registration Function**

Figure 12 explains the sequence diagram of the user registration process of this application. A user must register to purchase something from this web store. The registration procedure is quite simple. At first, the user needs to fill up all the required fields from the registration form. Then he will be able to submit the form by clicking the submit button. The backend controller behind registration page will confirm that all the required fields are correct. Otherwise, the application will send an error message to the user showing the exact error details and user will be redirected to the registration page again. If all the required data are correct, the user will be redirected to the default shopping page, and the user will be logged in automatically at the same time as a member.

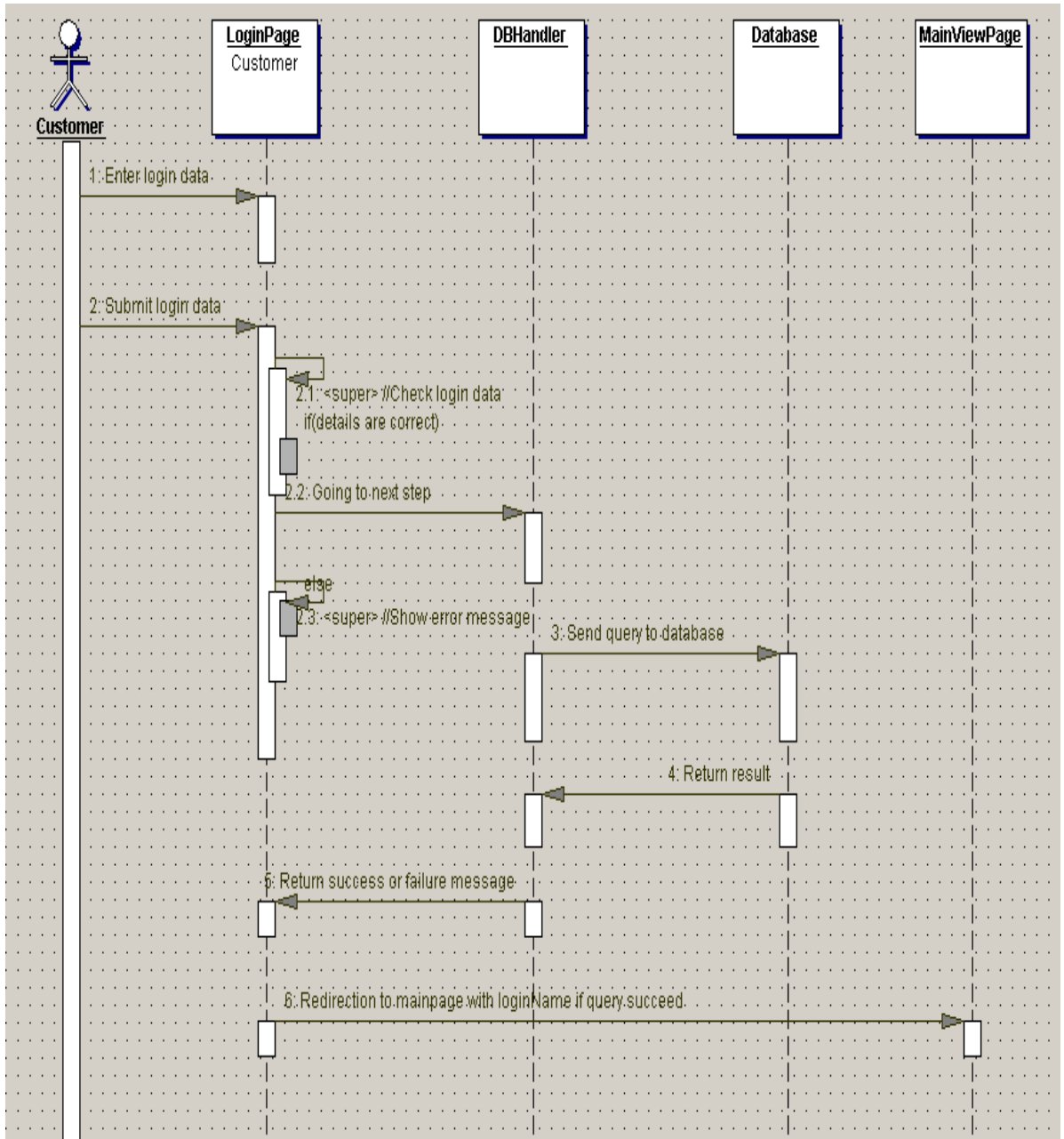


**Figure 12.** User Registration Function Sequence Diagram

### 3.2.4.2. Customer Login Function

Figure 13 describes the sequence diagram of client login procedure of this application. To log in to the system, client needs a username and password. A login form with two required fields will be shown for the client. When a customer submits his account credentials and presses the login button, login method will check if the

provided data matches to the client's login details stored in the database. If it does not match, an error message will be displayed to the client for the corresponding field. If the submitted data matches with the credentials in the database, the customer will be redirected to the main page with a login name.



**Figure 13.** Customer Login Function Sequence Diagram

### 3.2.4.3. Administrator Login Function

Figure 14 describes the sequence diagram of the administrator login procedure of this application. It follows the same sequence procedure as the client's login function. Without the log in an admin will not be able to enter the system. The Admin panel is separated from the client side.

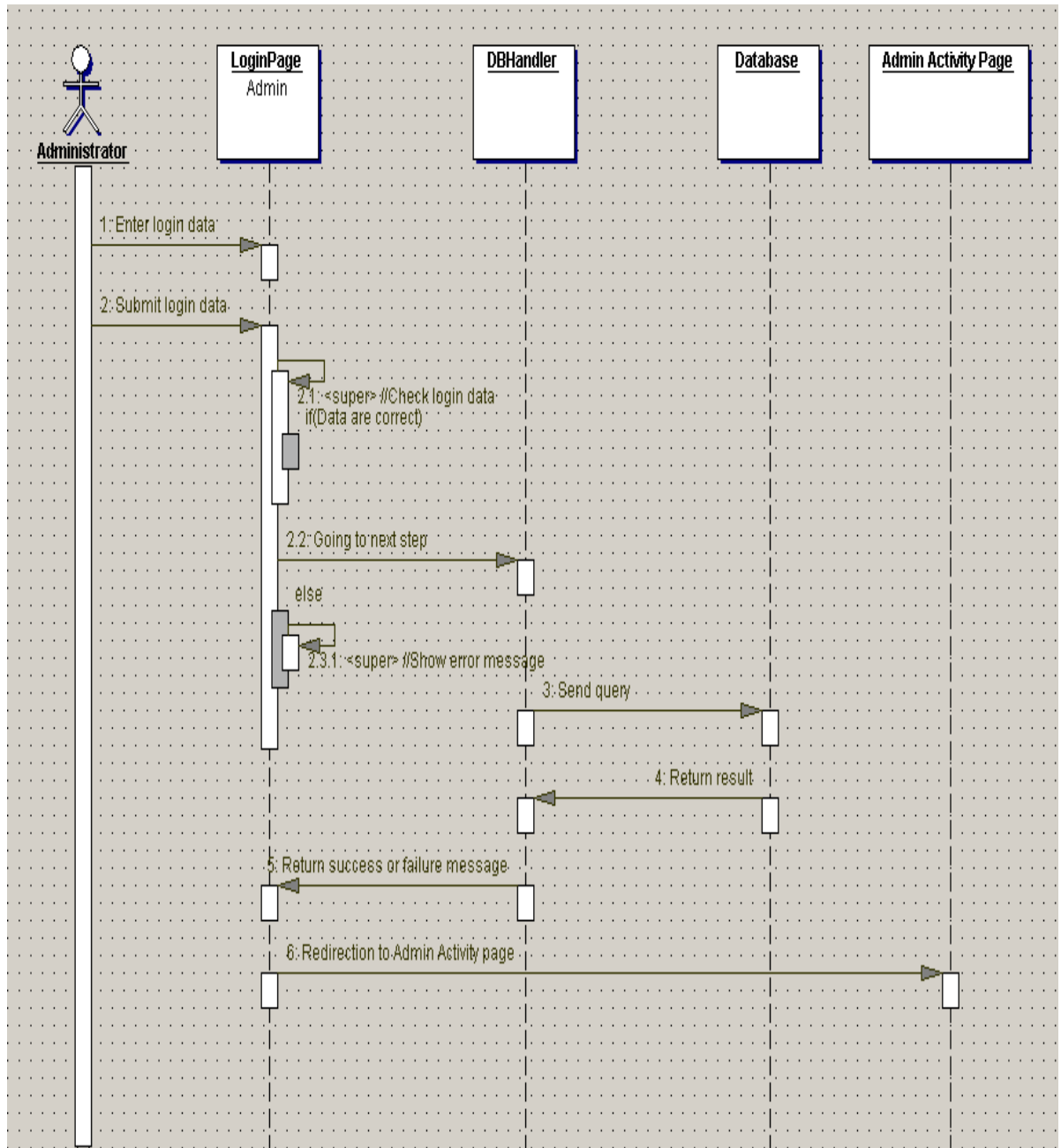
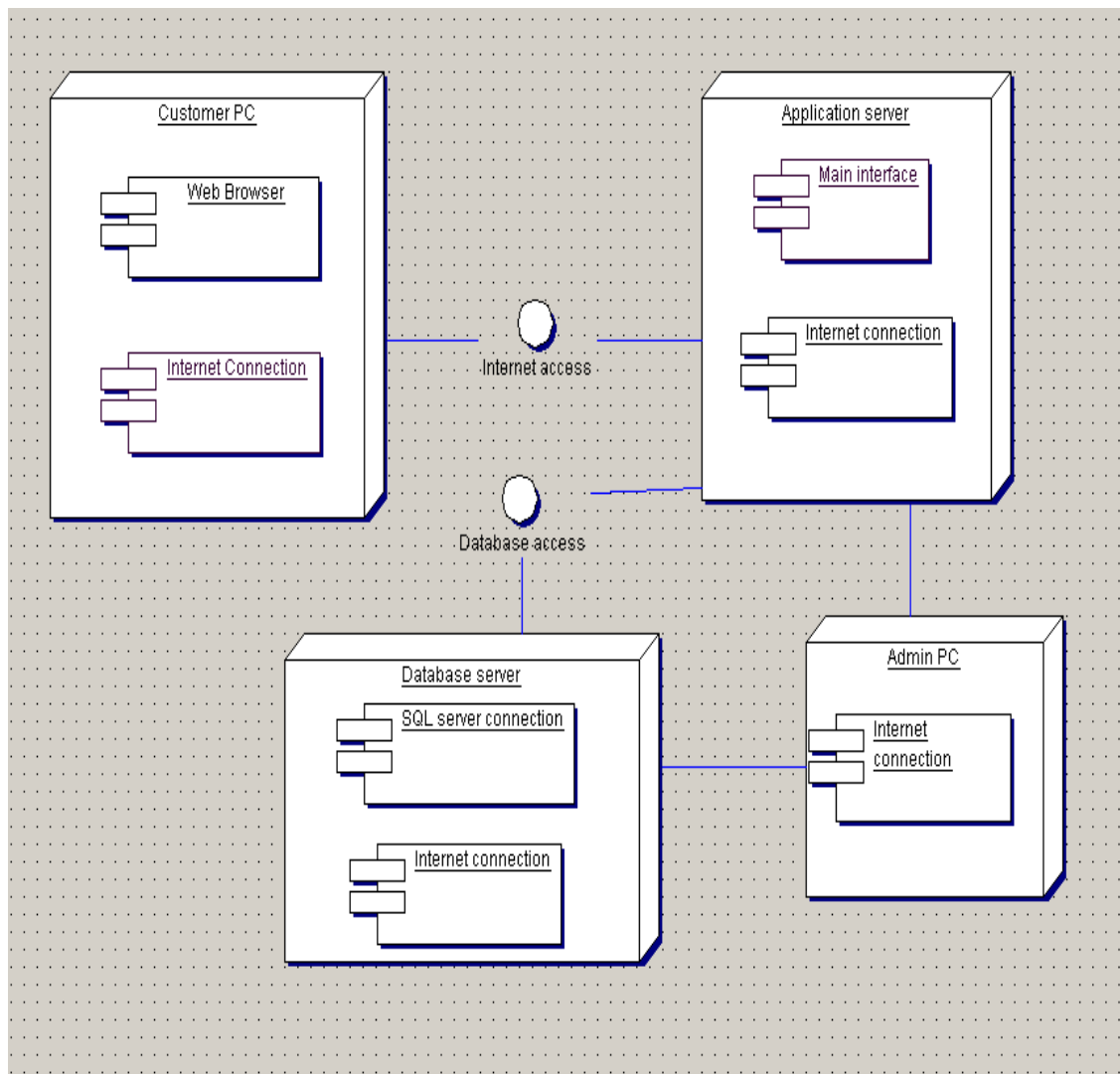


Figure 14. Administrator Login Function Sequence Diagram



### 3.2.5 Deployment Diagram of the project

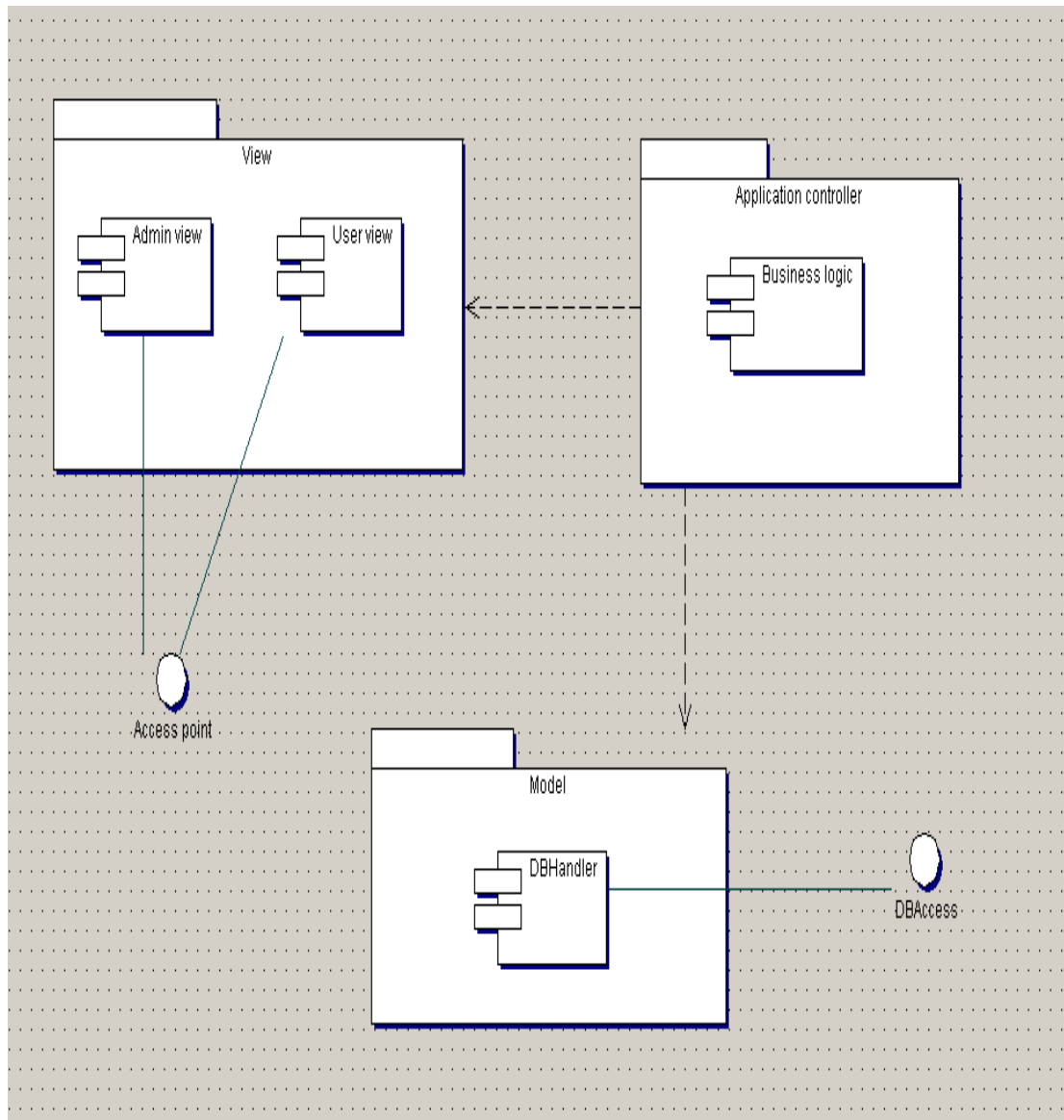
Figure 15 illustrates the deployment diagram of the application. It is the physical diagram of the system. It explains what type of hardware components needed for the system, what software component required to operate on each node of the system and how the various parts have connected each other. This application needs an internet connection for the customer or a web browser to browse to the site, as well to the application server. Application server is connected to the Database access point, and this point make connection to the database server side. Admin needs an internet connection to connect to the application server as well to the database server.



**Figure 15.** Deployment diagram of the system

### 3.2.6 Component Diagram of the project

Figure 16 illustrates the component diagram of this application. This diagram explains how this system is divided into many sub-systems. This is a three-layer application. Model, view and application controller are the layers of this application. The view layer has two parts: admin view and user view. The application controller includes all the business logic of the application which is the middle layer between view and model. The controller has the connection between view and model. At last, Model is responsible to connect to the database server.

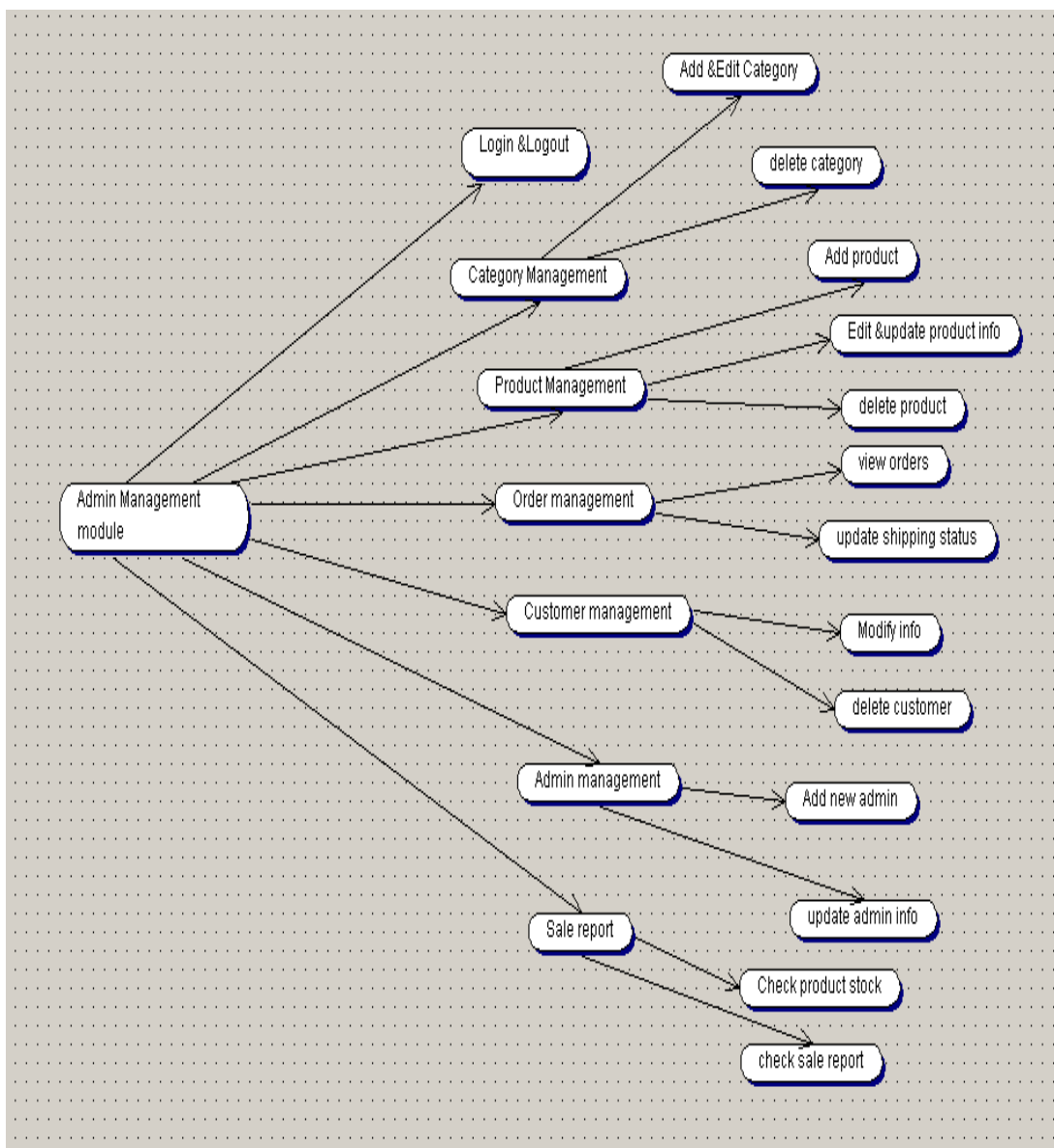


**Figure 16.** Component diagram of the application

This application is divided into two parts: Admin management module design and customer module design.

### 3.2.6.1. Admin Management module

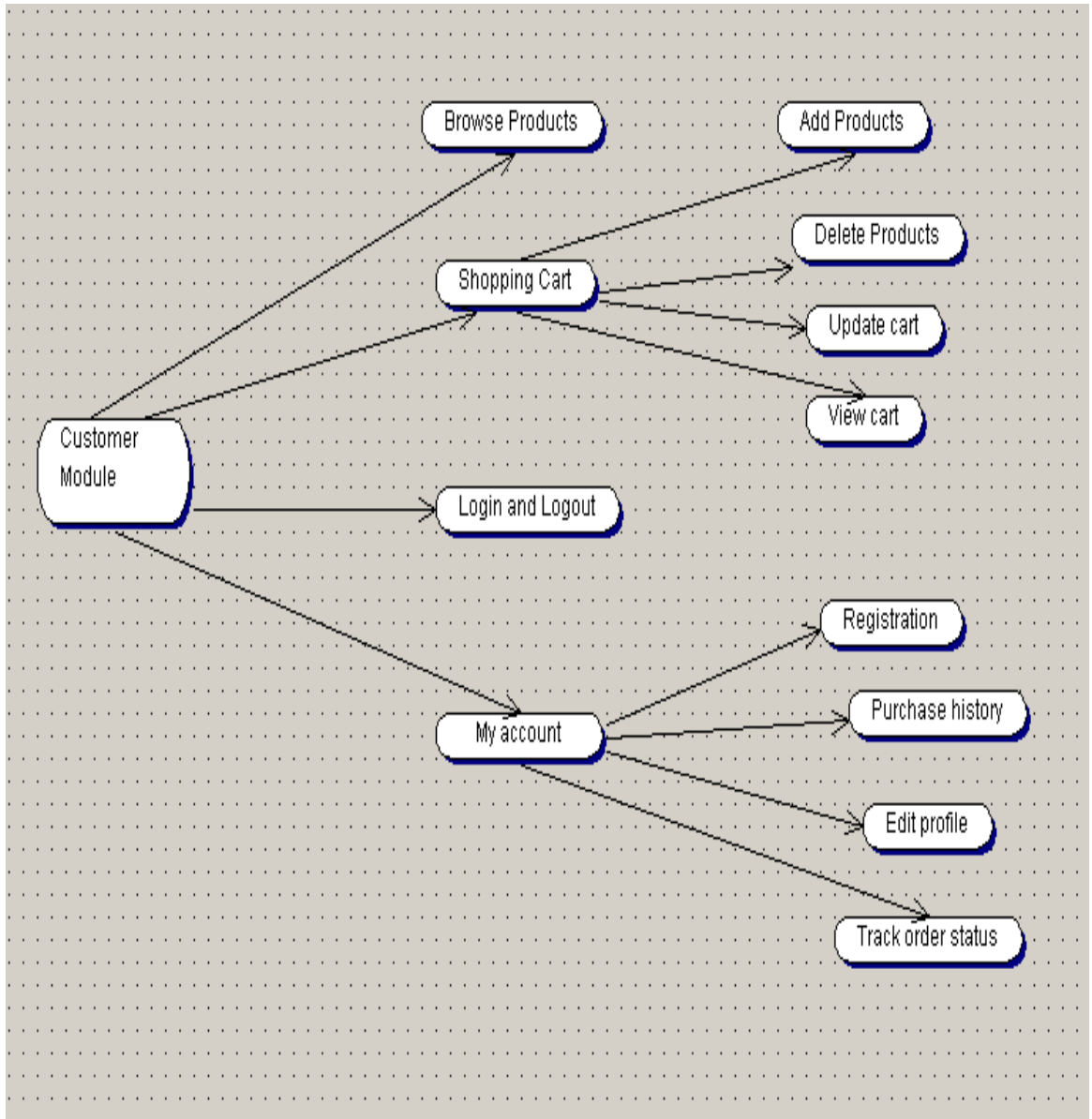
Admin management module includes admin login and logs out, member management, product management, and many other sub-modules. Figure 17 illustrates the Admin management module of the system.



**Figure 17.** Admin Management Module Diagram

### 3.2.6.2. Customer Module

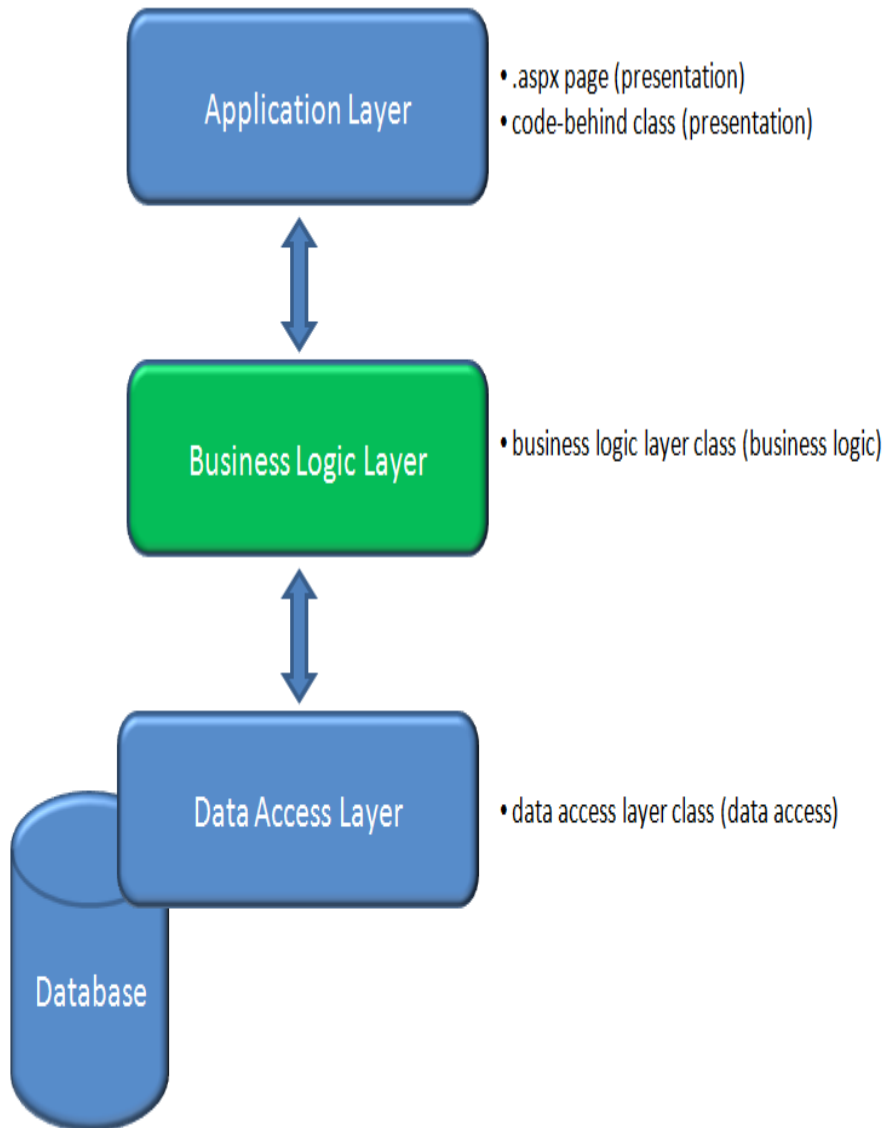
Customer module includes login, logout, shopping cart, my account settings. The Customer will be able to add, update and delete products in the shopping cart module. Figure 18 illustrates the Customer module design of the system.



**Figure 18.** Customer Module Diagram

### 3.2.7 Architectural Design of The System

Architectural design is the physical model transformation of a logical model. All the logical modules will be designed with structure in this section. Figure 19 illustrates the project architecture of this application.



**Figure 19.** Project Architecture of the application

## **4. DATABASE AND GUI DESIGN**

The database is one of the key element for an online e-commerce shop. The database is the place in where data is stored or saved. There are diverse types of Database Management Systems (DBMS), which are very helpful for the web application. DBMS manages and controls the function of a database. Most of the DBMS has user-friendly functions for GUI application. This section contains the design of the database and the design of GUI.

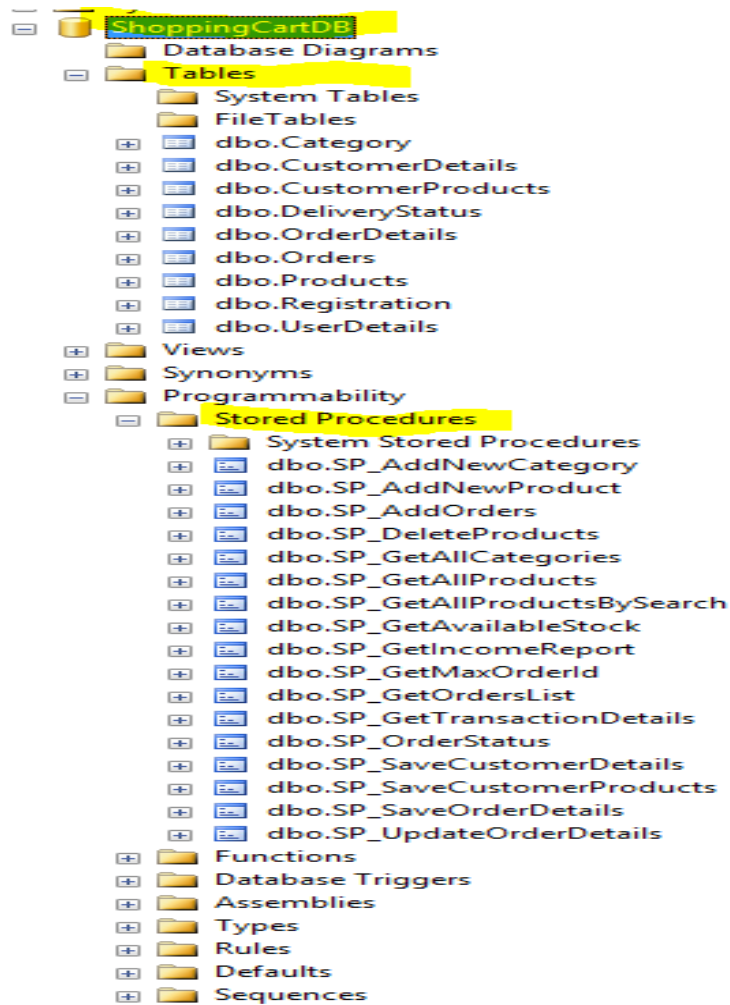
### **4.1 Database Design**

SQL server database is used for this web application. Microsoft SQL server 2014 management studio is an independent DBMS which has been utilized to create SQL server database and store procedure for the database. A connection string as well windows authentication has been used to connect to the web application.

“ShoppingCartDB” is the database name for this project. This database consists of 8 tables and 17 store procedures. Store procedure is the bridge between database tables and the back-end server/controller of the application. Store procedure is basically the logic implemented in applications to save memory and time. Store procedure can be used for a complex or extensive processing which needs execution of many SQL statements. “ShoppingCartDB” is made with the tables named as Category, CustomerDetails, CustomerProducts, DeliveryStatus, OrderDetails, Orders, Products, Registration and UserDetails.

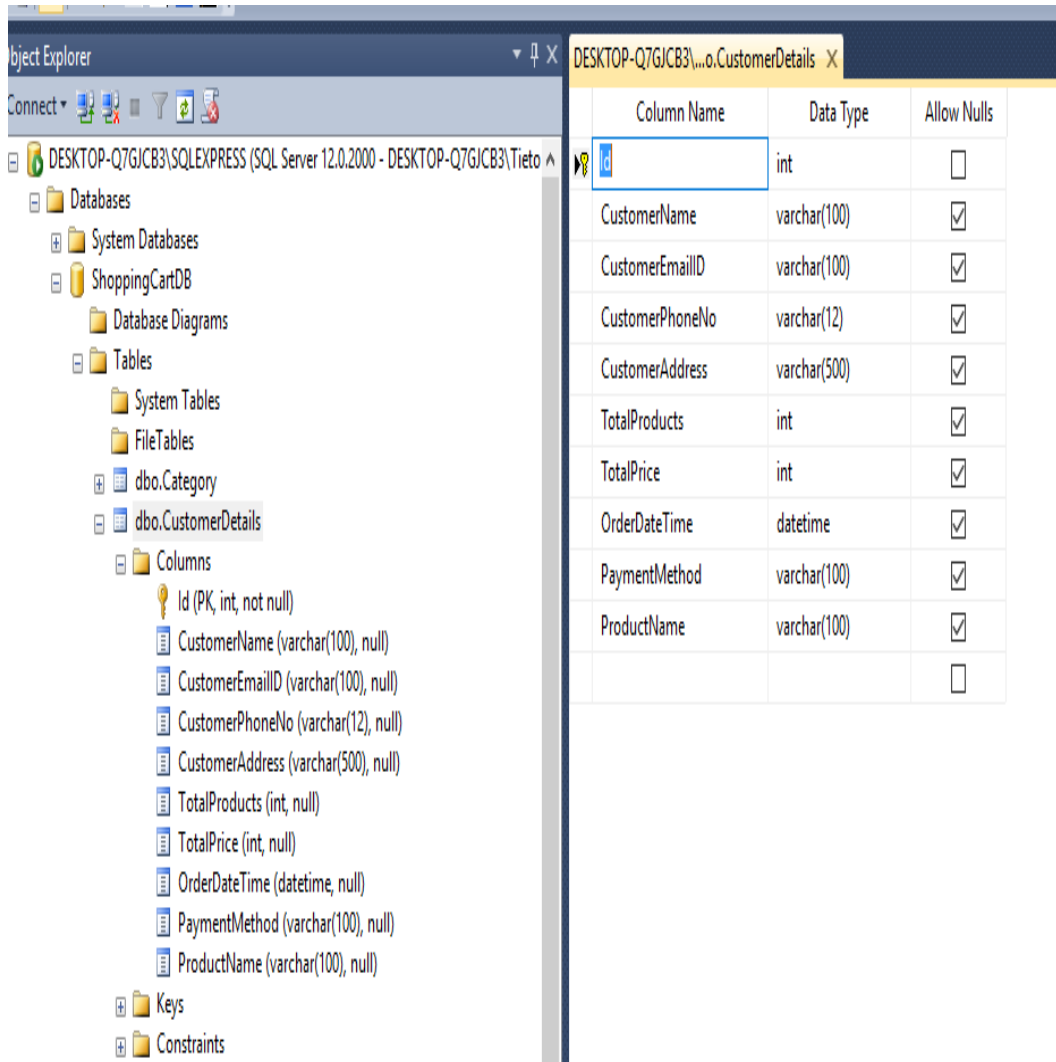
Name of the store procedures inside ShoppingCartDB are SP\_AddNewCategory, SP\_AddNewProduct, SP\_AddOrders, SP\_DeleteProducts, SP\_GetAllCategories, SP\_GetAllProducts, SP\_GetAllProductsBySearch, SP\_GetAvailableStock, SP\_GetIncomeReport and so on.

Figure 20 shows the database structure of the application.



**Figure 20.** Database structure of the system

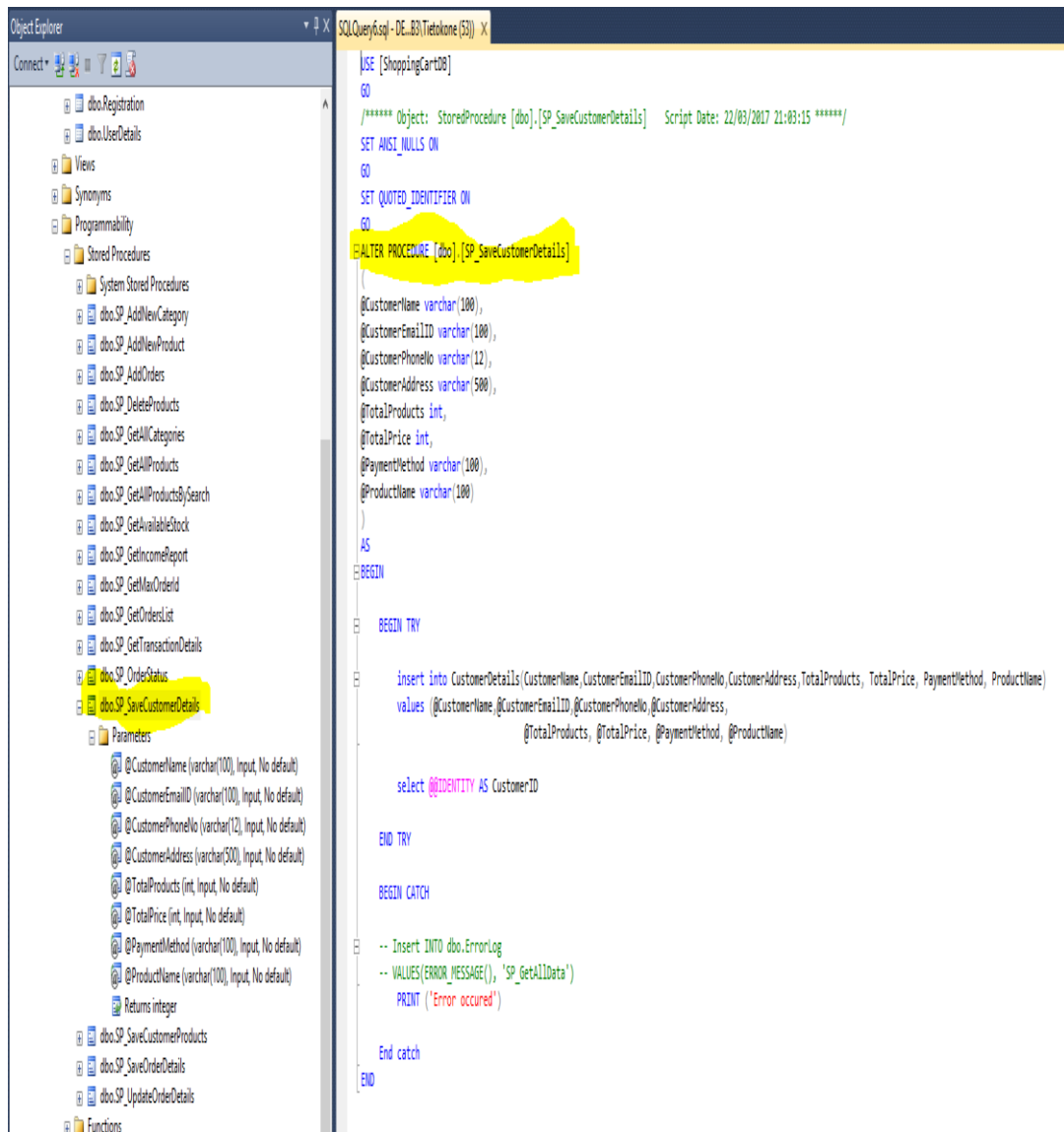
A database table has a unique name, and it includes rows and columns. Every column has own name and datatype. SQL query has been executed to make a database table. “CustomerDetails” is the name of a database table in where all the details of the customers (Customer name, email address, phone number, address, products’ quantity, total amount, order date, product name, payment method), who are ordering products, has been stored. A primary key (id) is used as a unique id of the table. Figure 21 shows the database table of Customer Details.



**Figure 21.** Customer details database table

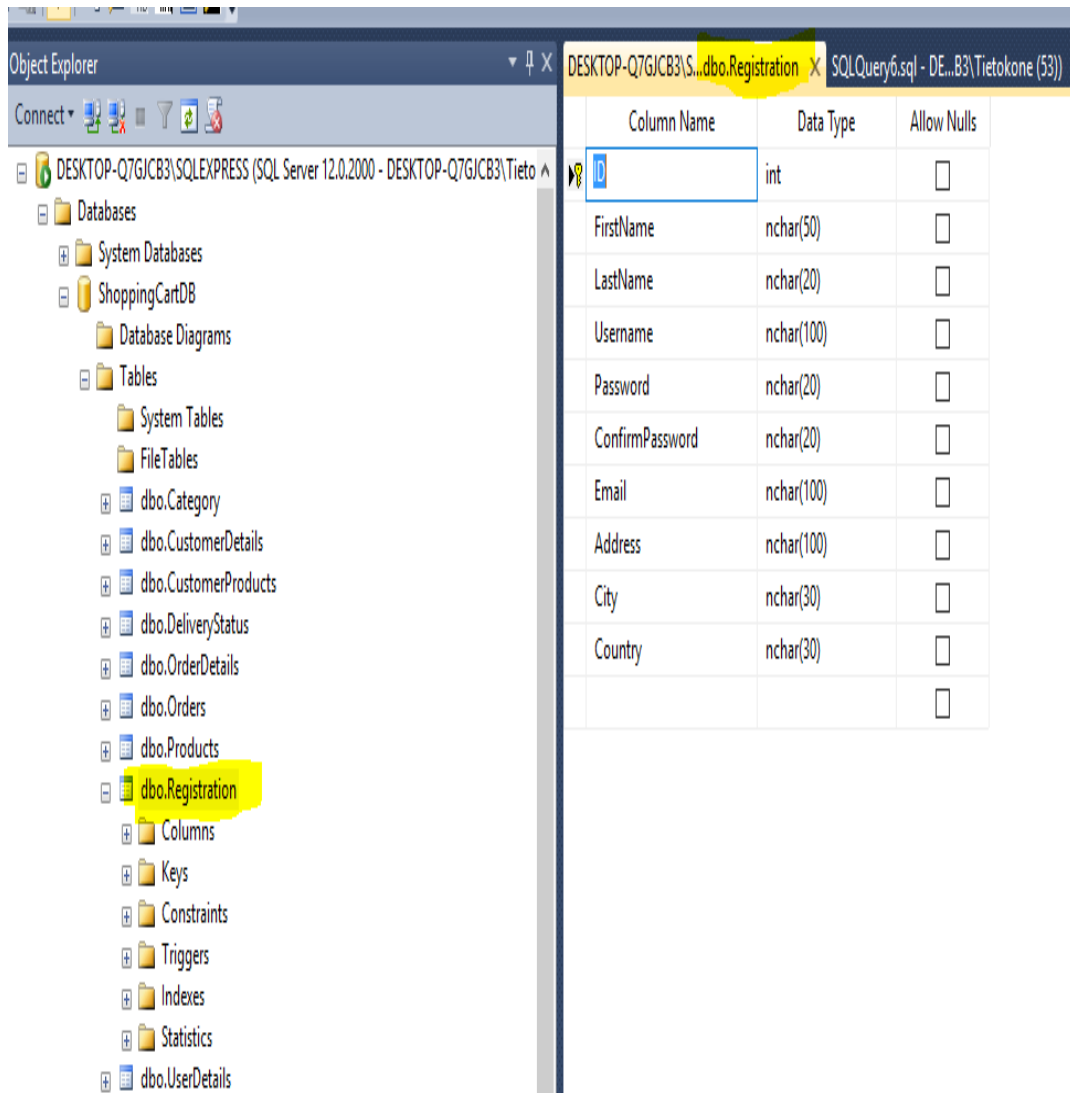
A store procedure called “SP\_SaveCustomerDetails” is executed to save the customer details provided in the web application to the “CustomerDetails” database table. Basically, it uses a SQL script as a query language. Figure 22 describes the store procedure for saving customer details in the database.





**Figure 22.** Store procedure for SaveCustomerDetails

Another table named as “Registration” is used to store registered customers’ data. Customer’s login details are saved in this table, which are essentials to login into the system. Figure 23 shows the Registration table of this application.



**Figure 23.** Registration database table

## 4.2 Graphical User Interface(GUI) Design

Graphical user interface is a kind of user interface which gives permission to the users to interact with the electronic machines/devices through visual indicators or graphical icons.

This system has been divided into two interfaces: Admin interface and User/Customer interface. A user can browse to the web pages, add items to the shopping cart. Only a registered user can purchase from the shop by using PayPal account or debit/credit cards processed by PayPal. Admin has the authorization to add, edit and delete products, to update customers' information.

## 4.2.1 Home Page Design

The homepage is the official page for this online shop. When a client browses the internet and searches for the domain name of the company, then the home page of this shop will be available to the users or customers displaying products' name, categories, products' images, search options, login and registration links, help desk support. Any user can search products by inserting keywords into the search box or by clicking the category's name. A user can add products to the shopping cart by clicking the "Add To Cart" button. A shopping cart icon is shown to the user, which is empty until someone adds a product to the cart. Figure 24 illustrates the home page design of this application.

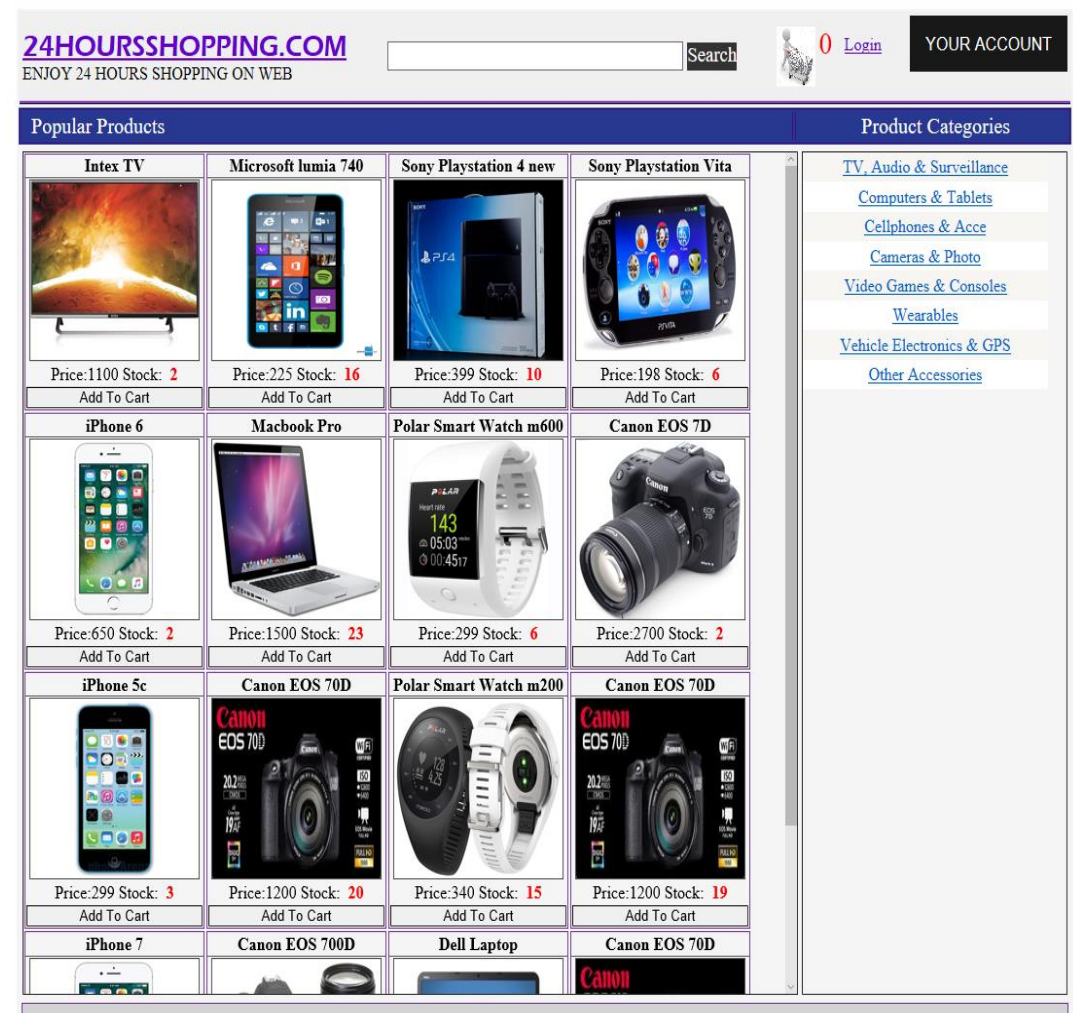
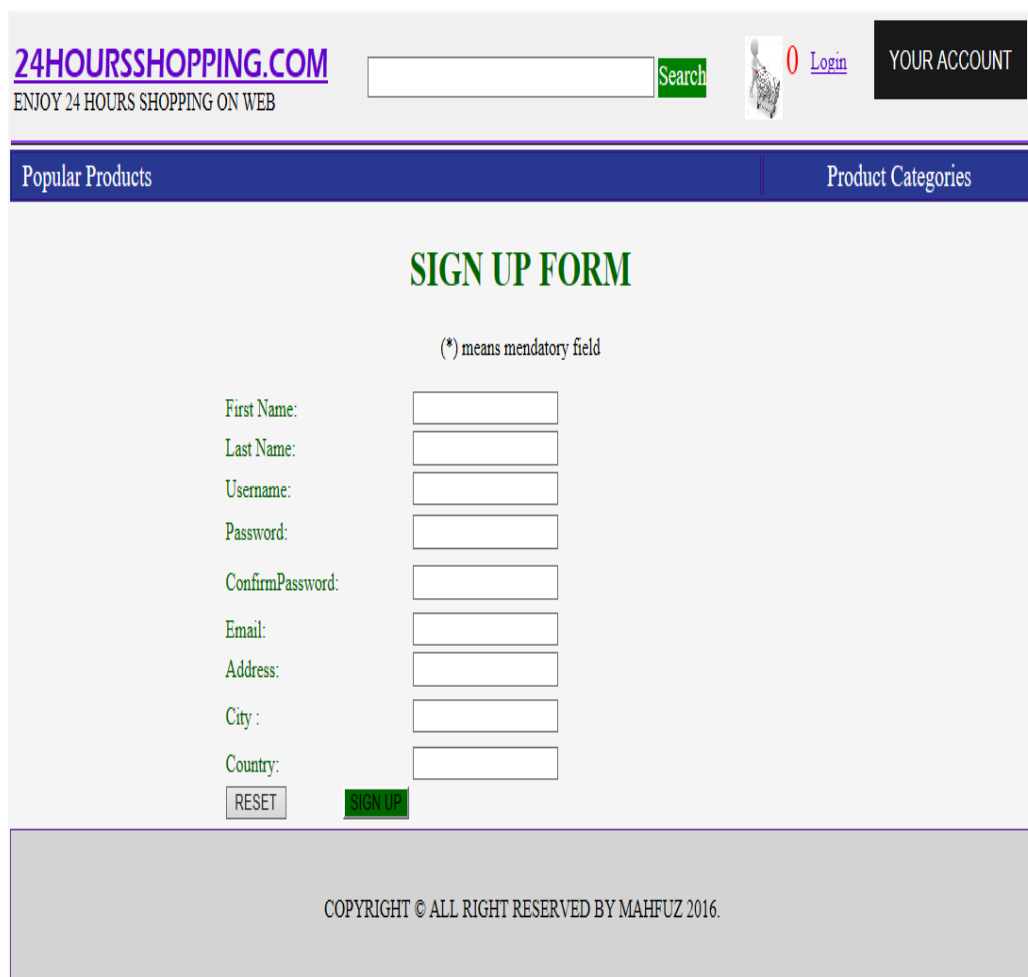


Figure 24. Home Page Design

## 4.2.2 User Registration Page Design

After clicking the SIGN UP button, a sign-up form will appear to the users. After completing all the required fields, the user will be registered successfully. When a user clicks the sign-up button, all the data will be saved to the database table called “Registration.” If the user leaves all the required fields incomplete, a red (\*) sign will appear beside each unfilled parameter or field.

“Registration.aspx” is the sign-up page which is displayed to the user to fill up to be a registered member of the system. “Registration.aspx.cs” is used as code behind “Registration” page to link to the database. Figure 25 illustrates the User Registration page design of this application.



**24HOURSSHOPPING.COM**  
ENJOY 24 HOURS SHOPPING ON WEB

Search

Login

YOUR ACCOUNT

Popular Products

Product Categories

### SIGN UP FORM

(\*) means mandatory field

First Name:

Last Name:

Username:

Password:

ConfirmPassword:

Email:

Address:

City :

Country:

RESET SIGN UP

COPYRIGHT © ALL RIGHT RESERVED BY MAHFUZ 2016.

**Figure 25.** User Registration Page Design

### 4.2.3 Login Page Design

A login form will be available to the user after clicking the login button shown on the home page. A user will be logged in automatically after completing the registration form successfully. A customer can log in by using his login details such as username and password. If the customer provides the wrong login details, the system will show an error message to the customers. Figure 26 illustrates Login page design before a successful login to the system.

The image shows a web page layout for 24HourShopping.com. At the top left is the logo '24HOURSSHOPPING.COM' with the tagline 'ENJOY 24 HOURS SHOPPING ON WEB'. To the right is a search bar with a green 'Search' button. Further right is a 'LOGIN' link and a 'YOUR ACCOUNT' button. Below this is a dark blue navigation bar with 'POPULAR PRODUCTS' and 'Product Categories'. The main content area is titled 'LOGIN' and contains a form with 'USERNAME:' and 'PASSWORD:' labels, two input fields, and a green 'LOGIN' button. Below the form is a link: 'ARE YOU NEW USER? [CLICK HERE TO SIGN UP](#)'. The footer contains links for 'ABOUT US', 'SHIPPING AND RETURNS POLICY', 'CONTACT', and 'SUPPORT', along with the copyright notice 'COPYRIGHT © ALL RIGHT RESERVED BY MAHFUZUL 2017'.

**Figure 26.** Customer Login Page Design

When a customer provides the correct username and password, he will be able to log in. After a successful login, the customer can see their name on the home page, and a logout option will be displayed under the name of the client. Figure 27 illustrates Login view design after a successful login to the system.

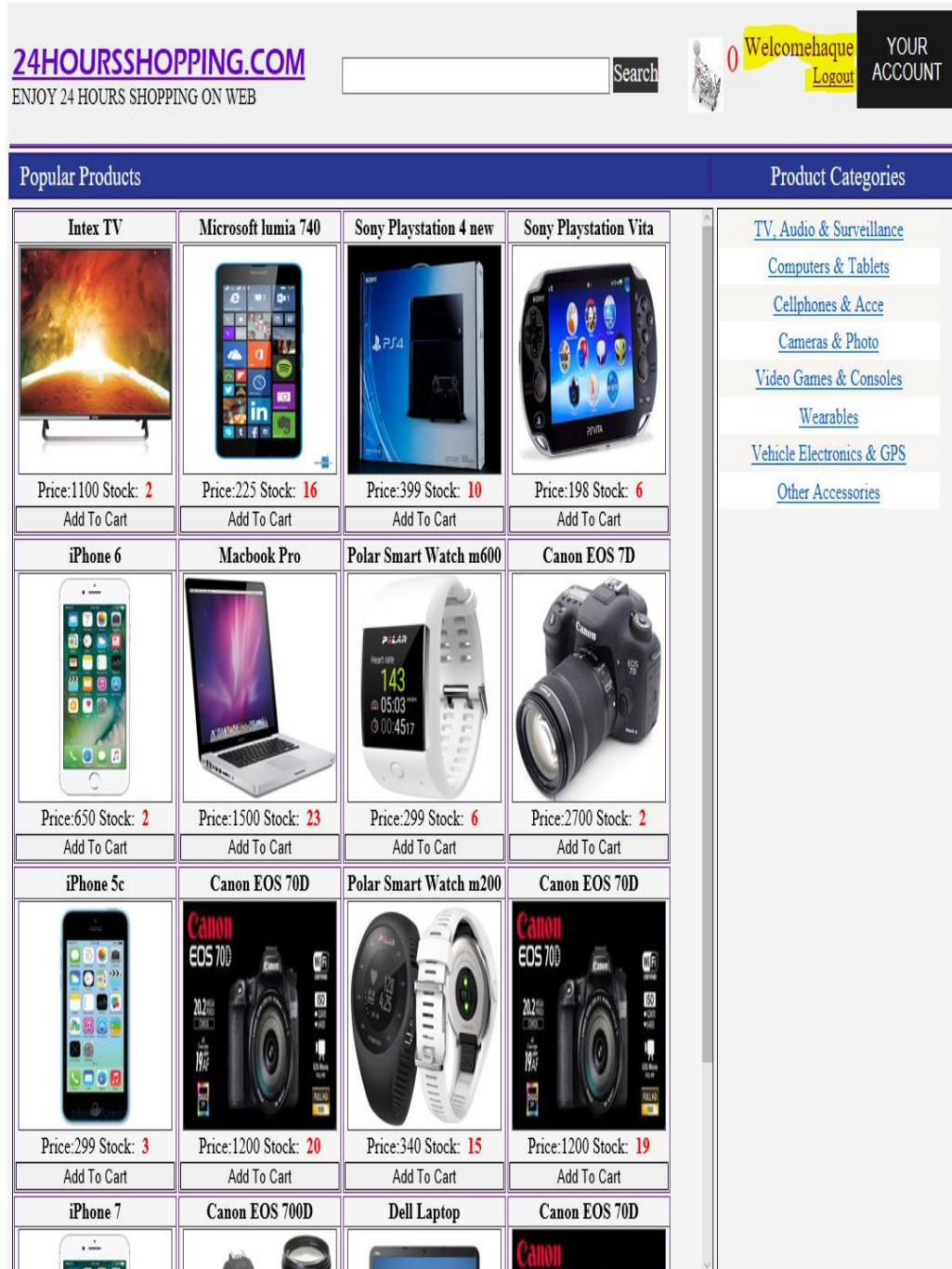


Figure 27. Successful Login View Design

## 4.2.4 Shopping Cart Design

The shopping cart is an essential part of a web shop application. A user can add products to the cart, update the quantity of the goods, delete products from cart, checkout with PayPal account. Customer must fill their contact details to be able to check out with PayPal in this web application. The customer can review the order details and total amount to pay for the order. Figure 28 illustrates the shopping cart view in the system.

**24HOURSSHOPPING.COM**  
ENJOY 24 HOURS SHOPPING ON WEB

Search

Login

YOUR ACCOUNT


Your Shopping Cart Products

Customer Details

**Please review your order!**

1 \* Macbook Pro 1500 €  
Total To Pay: 1500 €

Macbook Pro



AvailableStock: 23  
Price:1500 \* 1

RemoveFromCart

PayPal Check out  
The safer, easier way to pay

Continue Shopping

Full Name:  
PhoneNo:  
EmailID  
Address

Total Products :  
1

Total Price :  
1500

Product Name :  
Macbook Pro

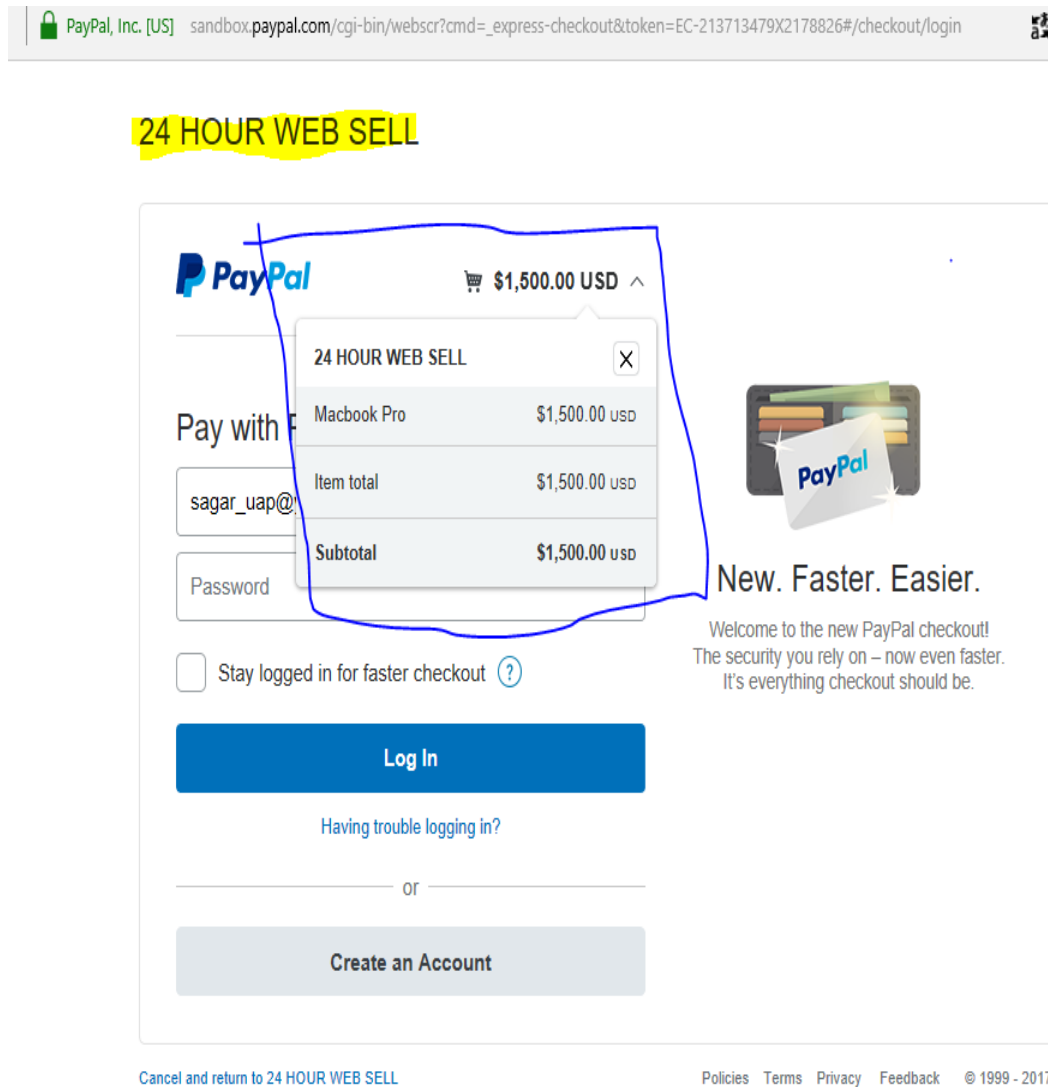
Shipping and Handling :  
 FREE SHIPPING  
 EXPRESS DELIVERY  
 REGULAR DELIVERY

Payment Mode :  
 1. PAYPAL PAYMENT  
 2. BANK TRANSFER  
 3. CARD PAYMENT

**Figure 28.** Shopping Cart Design

#### 4.2.4.1. Payment and Transaction

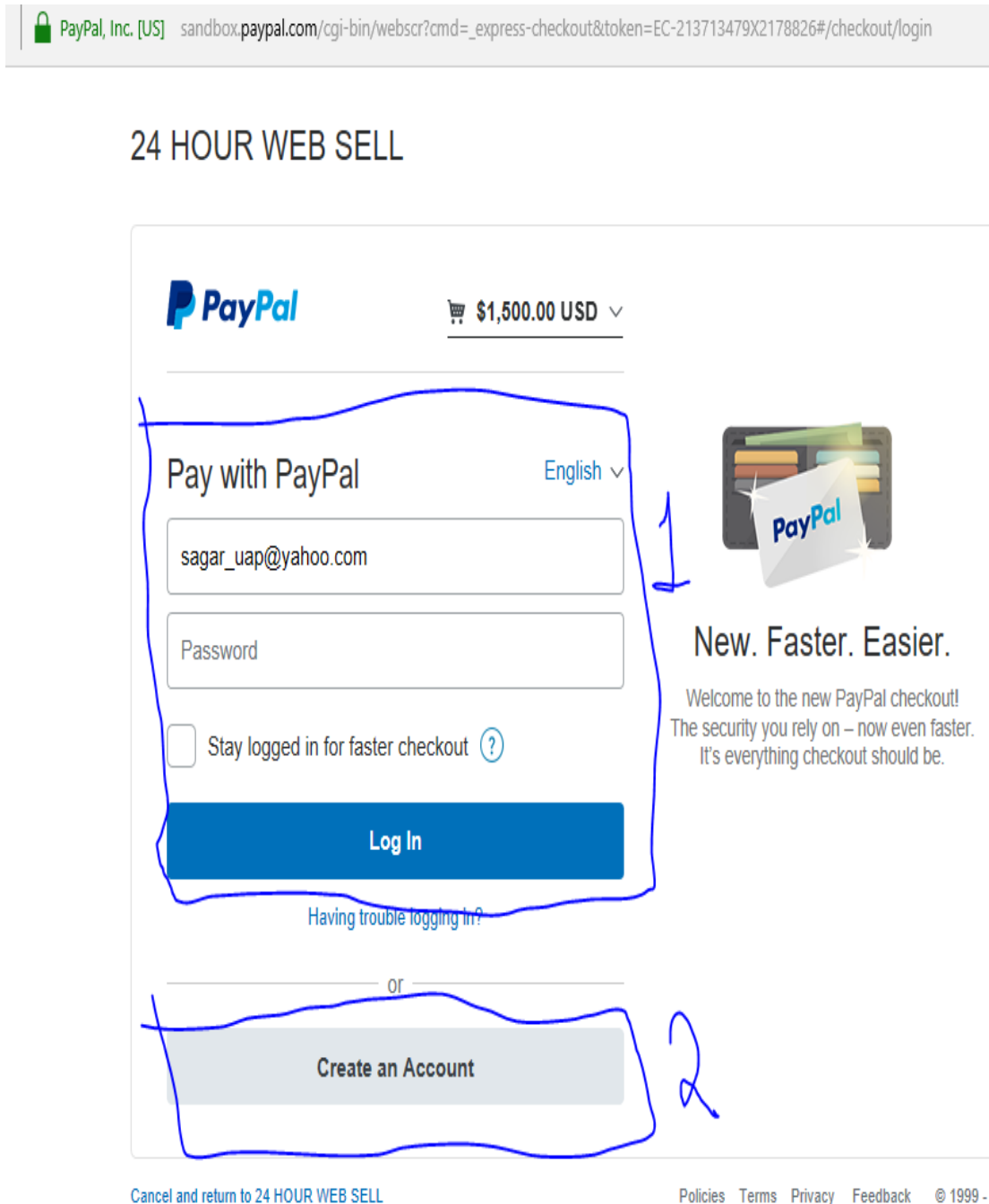
After completing all the customer details fields, a user can go through PayPal checkout process. When a user clicks the “PayPal Checkout” button, user authentication required to be able to buy products through this webshop. After a successful login, a user is able to check out with PayPal payment process. When PayPal checkout button is clicked, the order will be sent to the PayPal site, and PayPal sandbox will return the order summary which includes products’ name, products’ quantity and total amount to pay with the heading of “24HOUR WEB SALE”. Figure 29 illustrates an order summary sent to PayPal sandbox account.



**Figure 29.** Payment and Transaction Design Order Summary



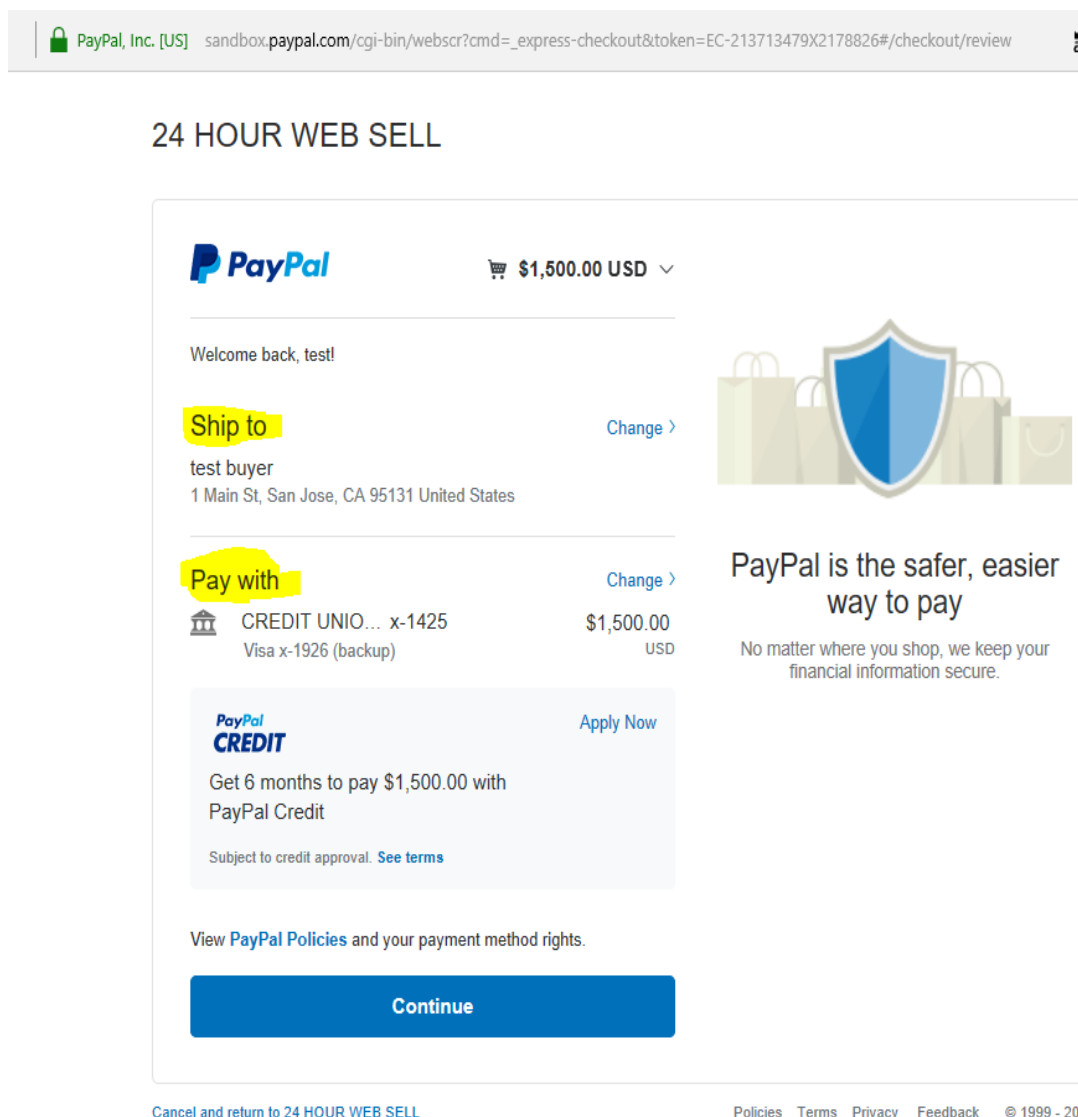
The customer can log in with the PayPal account credentials and pay the bill. Otherwise, they can create a new account to the PayPal system. Figure 30 illustrates how to pay using the PayPal account or how to make a new PayPal account.



**Figure 30.** Payment and Transaction Design

#### 4.2.4.2. Payment by PayPal Account

If a customer owns a PayPal account, he will be able to pay bills using option 1 displayed in Figure 24. He needs an email address and password to sign in to the PayPal system. Otherwise, the customer needs to create an account to pay the bill through the PayPal payment system. After a successful login, the customer can see the shipping address as well as payment card details. The Client needs to click the continue button to go through the process. Figure 31 illustrates how PayPal is showing shipping address and credit card details to a buyer.



**Figure 31.** Payment by PayPal Account

#### 4.2.4.3. Payment Without PayPal

If a customer does not own a PayPal account, he can create a new one by completing all information details shown in the following figure. He must provide his credit card or debit card information and the billing address. An email address and password are the necessary options to make a PayPal account. Part1 and part2 from Figure 32-33 explain the details how to make a PayPal account. After submitting all the information, customer needs to click “Agree and Continue” to give authorization to the PayPal to make an account for the client. Figure 32-33 illustrates how a buyer can purchase by using a credit card through PayPal payment system.

The image shows a PayPal checkout interface. At the top left is the PayPal logo. To its right is a shopping cart icon with the text "\$1,500.00 USD". Below the logo, the text "Pay with debit or credit card" is highlighted in yellow. Underneath this, it says "We don't share your financial details with the merchant." To the right of the form is a graphic of shopping bags with a shield in front of them, and the text "PayPal is the safer, easier way to pay" followed by "No matter where you shop, we keep your financial information secure." The form itself contains several input fields: a "Country" dropdown menu set to "United States"; a row of credit card logos (MasterCard, VISA, DISCOVER, AMERICAN EXPRESS); a "Card number" field; "Expires" and "CSC" fields; "First name" and "Last name" fields; "Phone type" (set to "Mobile") and "Phone number" fields; a "Billing address" section with fields for "Street address", "Apt., ste., bldg.", "City", "State" (dropdown), and "ZIP code"; and a checked checkbox for "Ship to my billing address".

Figure 32. Payment with Debit/Credit Card Part1

## Create your PayPal account

Get **PayPal Purchase Protection** on [eligible purchases](#).

Email ?

Password

Confirm password

I have read and agree to PayPal's [User Agreement](#), [Privacy Policy](#) and [Electronic Communications Delivery Policy](#). If you provide a mobile phone number, you agree that PayPal may contact you to service your account, investigate fraud, or collect a debt, using autodialed or prerecorded calls or texts. You may decline to receive autodialed or prerecorded calls or texts and still use PayPal by contacting [customer support](#).

**Agree & Continue**

[Cancel and return to 24 HOUR WEB SELL](#)

[Policies](#) [Terms](#) [Privacy](#) [Feedback](#) © 1999 - 2017 [English](#) ▼

**Figure 33.** Payment with Debit/Credit Card Part2

#### 4.2.4.4. Order Review

Order review is the final check by the customer before PayPal takes the money from the buyer's account. Order review includes checking product's name, quantity, price, subtotal and shipping address. The Buyer can complete the order by clicking the "Complete Order" button. Figure 34 illustrates the order review process in the system.

24HOURSSHOPPING.COM  
ENJOY 24 HOURS SHOPPING ON WEB

Search

Welcomehaque  
Logout

YOUR ACCOUNT

POPULAR PRODUCTS

Product Categories

## Order Review

Products:

ProductID	ProductName	ProductPrice	ProductQuantity
15	Sony Playstation 4 new	399	1

Shipping Address:

test buyer  
1 Main St  
San Jose CA 95131  
US

Order Total:

\$399.00

Complete Order

[ABOUT US](#) [SHIPPING AND RETURNS POLICY](#) [CONTACT SUPPORT](#)  
COPYRIGHT © ALL RIGHT RESERVED BY MAHFUZUL 2017.


**Figure 34.** *Order Review Design*

#### 4.2.4.5. Complete Order

After a successful order completion, a transaction number and an order id will be displayed to the customer. The Customer will be able to track his order status by using the order number. A link with the order number will be sent to the client's email address which is provided during the checkout process. Figure 35 illustrates the checkout complete view design of this application.

**24HOURSSHOPPING.COM**  
ENJOY 24 HOURS SHOPPING ON WEB

Search



Welcomehaque  
[Logout](#)

**YOUR ACCOUNT**

---

Popular Products

Product Categories

## Checkout Complete

*Happy shopping 😊*

You Have Placed Order Successfully.

Your Order Number:

1144

Payment Transaction ID:

8FU36376W31280455

Thank You!

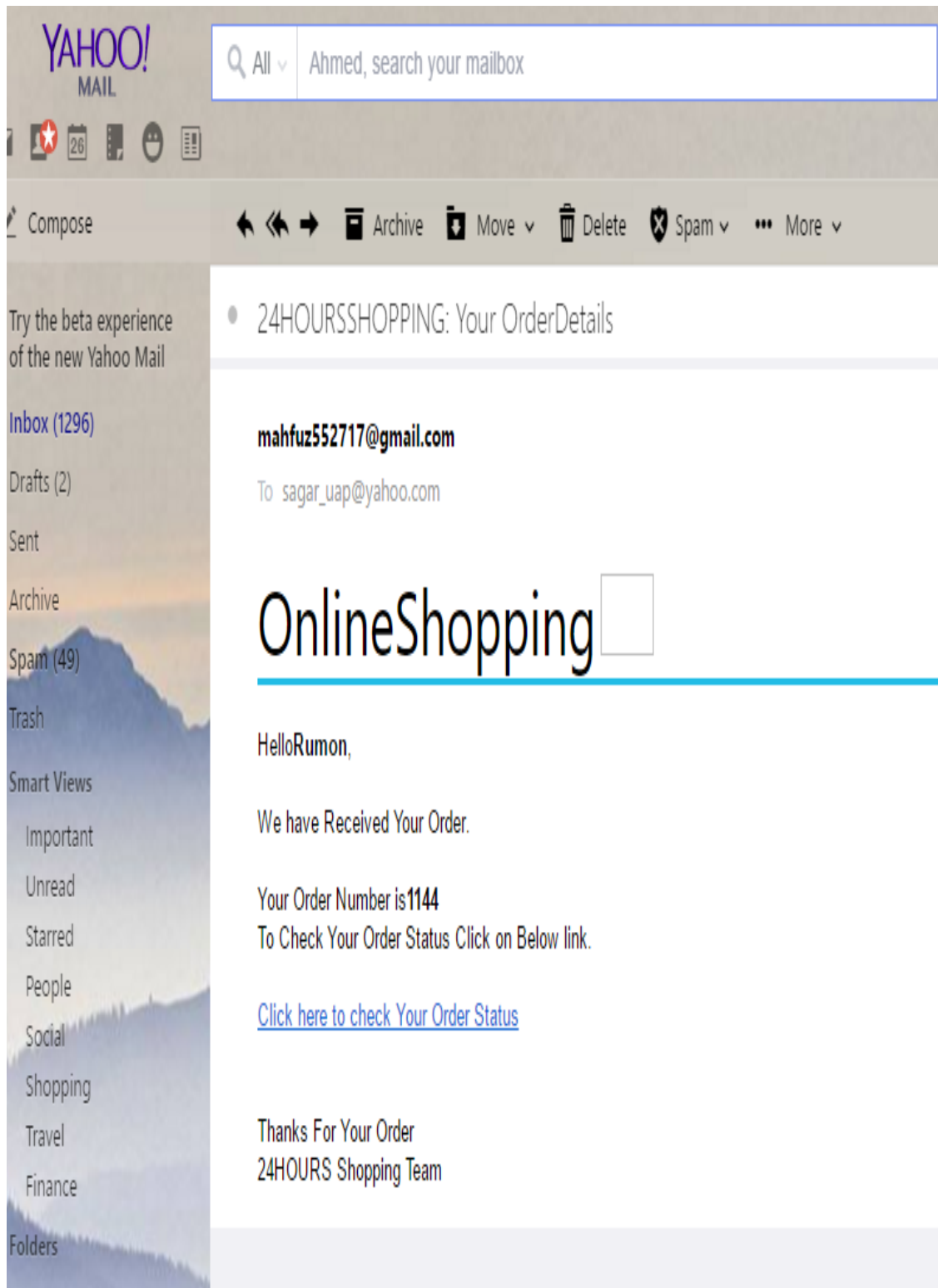
[Continue Shopping](#)

---

COPYRIGHT © ALL RIGHT RESERVED BY MAHFUZ 2016.

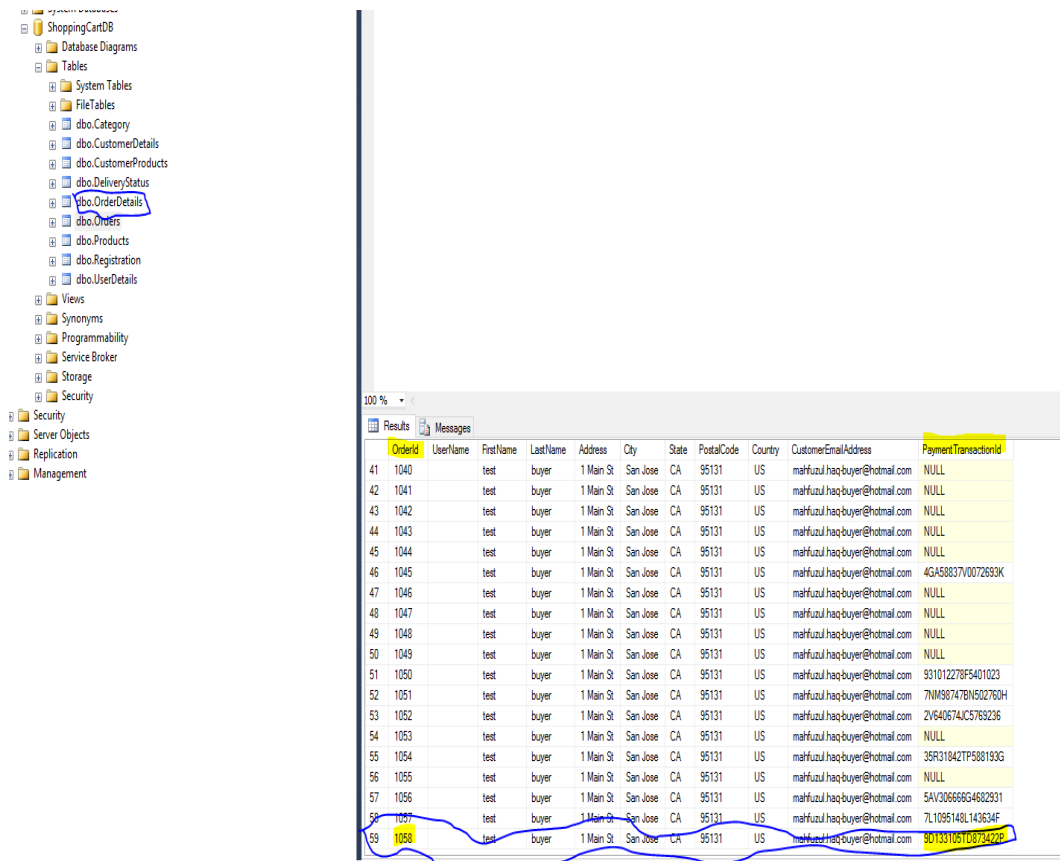
**Figure 35.** Complete Order Design

Figure 36 illustrates an email confirmation for a successful order from a customer's email.



**Figure 36.** Email Order Notification Design view

Order number and transaction number will be stored in the database called “Order-Details.” The Customer’s shipping address will also be saved in the same database. A data table named as “Orders” will store all the sold products’ information. Figure 37 displays the details from the database table called “OrderDetails.”



**Figure 37.** Complete Order Database Design

### 4.3 Admin Interface (GUI) Design

Admin interface has a different GUI than the user interface. Admin must log in to view the admin management panel. Admin login credentials will be stored in a database named “Admin.” After successful login, admin can see the management panel.

Admin can add categories, products to the database. Admin has the right to edit and delete products from the database. He can check products’ stock and income report.



He can update customers' information. He will be able to view all the orders made by the customers.

### 4.3.1 Admin Registration Page

Admin of the website can add other administrators. New admin could be registered by filling an admin registration form. All the data is stored in the database table named as "Admin." Figure 38 illustrates the Admin registration page design of this application.

ADMIN [Logout](#)

ENJOY 24 HOURS SHOPPING ON WEB

PANEL

[AdminRegister](#) [AddCategory](#) [AddProducts](#) [AllCategory](#) [AllAvailableProducts](#) [CustomerOrders](#) [ProductInStock](#) [IncomeReport](#) [RegisteredCustomer](#) [AllProductInfo](#)

## ADMIN REGISTRATION

First Name:

Last Name:

Username:

Password:

ConfirmPassword:

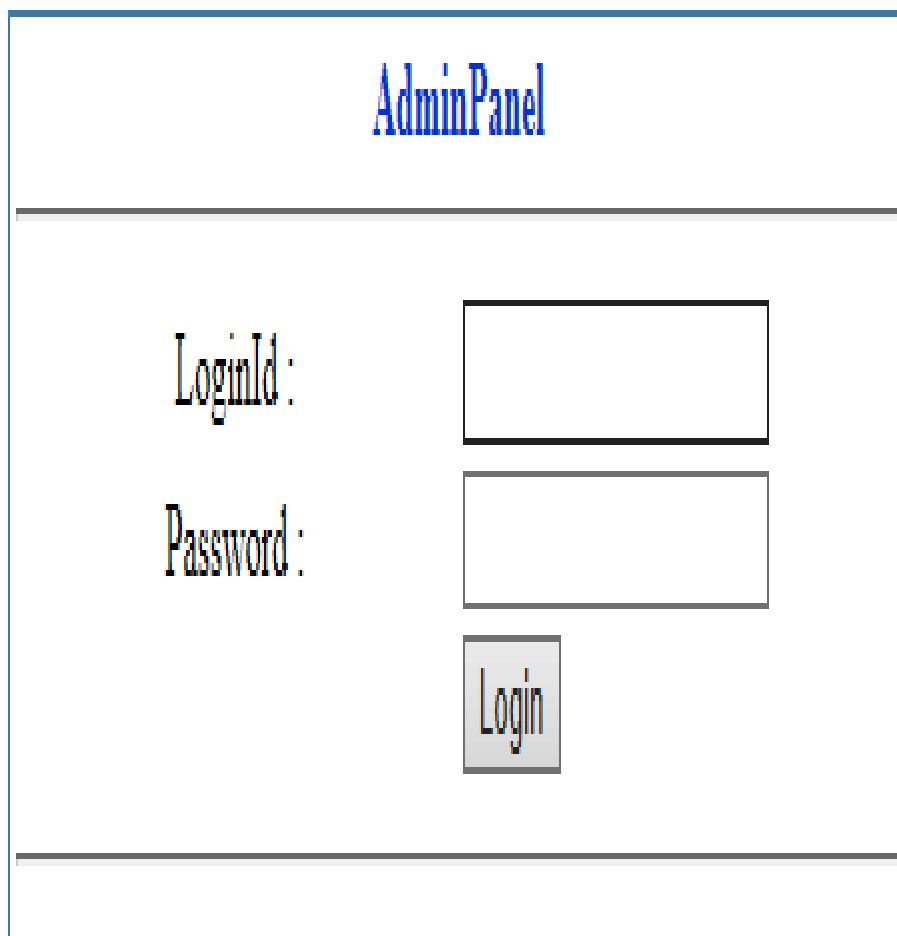
Email:

COPYRIGHT © ALL RIGHT RESERVED BY MAHFUZUL 2017.

**Figure 38.** Admin Registration Page Design

### 4.3.2 Admin Login Page

Admin panel is made for the admin only. It has LoginId and password fields. Without the valid LoginId and password, the admin will not be able to enter admin management panel. In case he forgets the login credentials, he will be able to check the database table called as “Admin.” Figure 39 illustrates Admin login page design of the system.



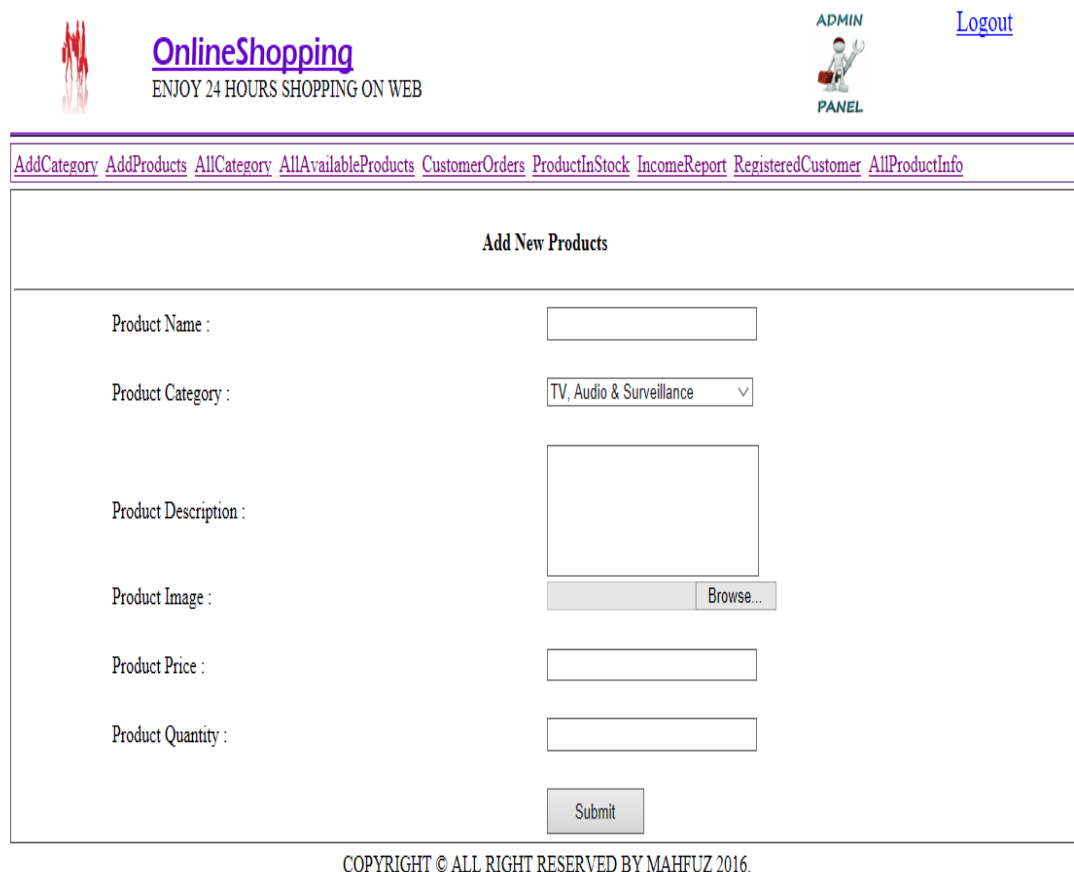
The image shows a web form titled "AdminPanel" in blue text at the top center. Below the title is a horizontal line. The form contains two input fields: "LoginId:" followed by a rectangular text box, and "Password:" followed by another rectangular text box. Below the password field is a grey button with the text "Login". At the bottom of the form is another horizontal line.

**Figure 39.** Admin Login Design

### 4.3.3 Admin Management Page

Admin management panel is vital for any online shop. This web application has an admin management panel. Only admin can manage this interface, but only after a successful login. Admin will be able to add a new category and new products. He can delete old or sold products from the database. Even he can update products information and customers' information. He can check all the available products and their stock. He will be able to view all the purchased products, and he can update shipping status. It also gives the possibility for the admin to get the income report for daily, monthly or yearly basis from this application. Figure 40

shows the admin management panel after a successful login.



The screenshot displays the Admin Management Panel for an online shopping application. At the top left, there is a logo for 'OnlineShopping' with the tagline 'ENJOY 24 HOURS SHOPPING ON WEB'. At the top right, there is a 'Logout' link and a 'ADMIN PANEL' icon. Below the header, a navigation menu contains links for 'AddCategory', 'AddProducts', 'AllCategory', 'AllAvailableProducts', 'CustomerOrders', 'ProductInStock', 'IncomeReport', 'RegisteredCustomer', and 'AllProductInfo'. The main content area is titled 'Add New Products' and contains a form with the following fields:

- Product Name :
- Product Category :
- Product Description :
- Product Image :
- Product Price :
- Product Quantity :

At the bottom of the form is a 'Submit' button. Below the form, the text 'COPYRIGHT © ALL RIGHT RESERVED BY MAHFUZ 2016.' is displayed.

**Figure 40.** Admin Management View Design

## **5. IMPLEMENTATION**

A well-planned implementation defines the main objectives to acquire the system's requirements and the required timeframe to accomplish the project.

### **5.1 Description**

The Implementation phase provides a complete strategy on how to plan, manage and analyze the whole project in a clear, structured manner. The implementation plan includes implementation strategy, business process analysis, scheduling identification, resources or requirements identification and integration to the system, customer requirements analyze and determination.

This web application will be implemented based on the online shopping system on both the perspectives of a customer and admin interface. A step by step process is maintained to implement all the functions required to complete the project. A complete guideline has been followed for each step during the implementation process. A database system has been designed and implemented before to start the process to apply the Admin and the consumer phase. All the functions have been developed by measuring the possible risk factors and applied during the required timeframe. All the functionality has been developed and implemented based on an ASP.NET framework and a SQL server database management system.

### **5.2 Implementation of Functions**

The implementation of this application is divided into two main phases: consumer interface and admin interface. The consumer or user interface has been implemented with html5, ASP.NET form and a CSS file are used to style and format the front side interface. The database has been carried out with SQL server database management system. The admin interface is also implemented based on ASP.NET framework.

### 5.2.1 Available Products in The System

Any consumer can browse or search products in this system. All the goods could be searched by category or by keywords. All the available products are stored in the database named as “Products.” All the goods will be accessible to the customers when they write the domain name of this shop to a web browser. A function and a connection class have been used to fetch the data from the database table.

A function called “GetProducts ()” will be called inside the PageLoad method, when the application will launch the browser. The code is given below:

```
if (!IsPostBack)
{
    lblCategoryName.Text = "POPULAR PRODUCTS ";
    GetCategory();//Get all Category
    GetProducts(0);//Get all Products
}
```

**Code Snippet 1.** Code for Getting all the products

This method uses a categoryID as a parameter. When a user clicks on a category name, all the products will be displayed to the user under the category name. Snippet 2 shows the code details of the method “GetProducts ()” which uses a categoryID.

```
private void GetProducts(int CategoryID)
{
    ShoppingCart k = new ShoppingCart()
    {
        CategoryID = CategoryID
    };

    dlProducts.DataSource = null;
    dlProducts.DataSource = k.GetAllProducts();
    dlProducts.DataBind();
}
```

**Code Snippet 2.** Code for Getting products using categoryID

The method called “GetCategory ()” will fetch all the category name from the database. Snippet 3 shows the code details of the method “GetCategory ().”

```

private void GetCategory()
{
    ShoppingCart k = new ShoppingCart();
    dlCategories.DataSource = null;
    dlCategories.DataSource = k.GetCategories();
    dlCategories.DataBind();
}

```

**Code Snippet 3.** Code for Getting category name

To fetch the data from the database “GetProducts” and “GetCategory” method use an instance of a class called “ShoppingCart.cs” and bind the data to display in the default page of this application. The ShoppingCart class uses a function named as “GetAllProductsBySearch,” which returns all the products searched by the keyword. The code is given below for GetAllProductsBySearch method:

```

//Get Products searching by Name
public DataTable GetAllProductsBySearch()
{
    SqlParameter[] parameters = new SqlParameter[1];
    parameters[0] = DataLayer.DataAccess.AddParameter("@ProductName", ProductName, System.Data.SqlDbType.VarChar, 100);
    DataTable dt = DataLayer.DataAccess.ExcuteDTByProcedure("SP_GetAllProductsBySearch", parameters);
    return dt;
}

```

**Code Snippet 4.** Code for the GetAllProductsBySearch return data table

The GetAllProductsBySearch method uses a connection class called “DataAccess.cs.” The DataAccess class uses a connection string to connect to the database. A store procedure called “SP\_GetAllProductsBySearch” has been declared to execute SQL queries into the database, and a data table is returned to the “GetAllProductsBySearch” function in response to the SQL query request. The code for DataAccess class is given below:

```

public class DataAccess
{
    public static string ConnectionString
    {
        get
        {

```

```

        Return ConfigurationManager.ConnectionStrings["MyConn"].ConnectionString.ToString();
    }
}
// Method to add parameter
public SqlParameter AddParameter(int size, string parameterName, SqlDbType DbType, object value)
{
    SqlParameter param = new SqlParameter();
    param.ParameterName = parameterName;

    param.Value = value;
    param.SqlDbType = DbType;
    param.Size = size;
    param.Direction = ParameterDirection.Input;
    return param;
}

// Method to connect Store procedure to the Database
public DataTable ExcuteDTByProcedure(SqlParameter[] Params, String ProcedureName,)
{
    SqlConnection conn = new SqlConnection(ConnectionString);
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = conn;
    cmd.CommandText = ProcedureName;
    cmd.Parameters.AddRange(Params);
    cmd.CommandType = CommandType.StoredProcedure;
    SqlDataAdapter adopter = new SqlDataAdapter(cmd);
    DataTable dTable = new DataTable();

    try
    {
        adopter.Fill(dTable);
    }catch(Exception ex)
    {
    }
    finally
    {
        //Disposing objects
        adopter.Dispose();
        cmd.Parameters.Clear();
        cmd.Dispose();
        conn.Dispose();
    }
    return dTable;
}
}

```

**Code Snippet 5.** Code for DataAccess connection class

“SP\_GetAllProducts” is a store procedure which executes a SQL query to fetch the data from the database table called as “Products” and return the result. SQL query for store procedure “GetAllProducts” is given below:

```

CREATE PROCEDURE [dbo].[SP_GetAllProducts] (@CategoryID INT)
AS
BEGIN

    BEGIN TRY
    IF(@CategoryID <> 0)
    Begin
    select * from (select P.CategoryID,
                        P.ProductID,
                        P.Name,
                        P.Price,
                        P.ImageUrl,
                        C.CategoryName,
                        P.ProductQuantity,
                        Isnull(Sum(CP.TotalProduct), 0)
                        as ProductSold,
                        (P.ProductQuantity - Isnull(Sum(CP.TotalProduct), 0))
                        as AvailableStock from Products P
    Inner join Category C
    on C.CategoryID = P.CategoryID
    left join CustomerProducts CP
        on CP.ProductID = P.ProductID
    Group by P.ProductID,
            P.Name,
            P.Price,
            P.ImageUrl,
            C.CategoryName,
            P.ProductQuantity,
            P.CategoryID) StockTable
    where AvailableStock > 0
    and CategoryID = @CategoryID

    End
    Else
        Begin
            select *
            from (select P.CategoryID,
                        P.ProductID, P.Name,P.Price,
                        P.ImageUrl,C.CategoryName,
                        P.ProductQuantity,
                        Isnull(Sum(CP.TotalProduct), 0)as ProductSold,
                        (P.ProductQuantity - Isnull(Sum(CP.TotalProduct), 0))
                        as AvailableStock from Products P
    Inner join Category C

```



```

on C.CategoryID = P.CategoryID
left join CustomerProducts CP
on CP.ProductID = P.ProductID
Group by P.ProductID,
        P.Name,P.Price,
        P.ImageUrl,C.CategoryName,
        P.ProductQuantity,
        P.CategoryID) StockTable
        where AvailableStock > 0

End

```

END TRY

**Code Snippet 6.** SQL query for getting all the products from database

An ASP data list has been implemented to display the products and its attributes in this web shop. Inside the data list tag, an Item template is used and an ASP.net built-in “Eval()” function has been utilized to fetch the data from the back-end server and to display products’ information to the user. The code for displaying products on the web page is given below:

```

<asp:DataList runat="server" Width="700px" RepeatColumns="4" Font-
Italic="False Font-Bold="False" " Font-Overline="False" Font-Strikeout="False"
Font-Underline="False" ID="dlProducts" >

    <ItemTemplate>

        <div align="left">

            <table class="style4" cellspacing="1" style="border: 1px ridge #9900FF">

                <tr>

                    <td style="border-bottom-style: ridge; border-width: 1px; border-color:
                    #000000">

                        <asp:Label Text='<%=# Eval("Name") %>' style="font-weight: 700"
                        runat="server" ID="lblProductName" ></asp:Label>

                    </td>

                </tr> <tr>

                    <td>

                        <img runat="server" alt="" src='<%=# Eval("ImageUrl") %>'
                        style="border: ridge 1px black; width: 173px; height:160px;"
                        id="imgProductPhoto" />

                    </td>

                </tr>

            </table>

        </div>

    </ItemTemplate>

</asp:DataList>

```

```

        </td>
    </tr>
    <tr>
        <td>
            Price:<asp:Label ID="lblPrice" runat="server" Text='<##
            Eval("Price") %>'></asp:Label>
            <asp:Image ID="imgStar" runat="server" Visible="false" Im-
            ageUrl="~/Images/GreenStar1.png" style="width:15px;
            height:15px;" />
            Stock:&nbsp;
            <asp:Label runat="server" Text='<## Eval("Available-
            Stock") %>' ToolTip="Available Stock" ID="lblAvailableStock"
            ForeColor="Red" Font-Bold="true"></asp:Label>
            <asp:HiddenField runat="server" Value='<##
            Eval("ProductID") %>' ID="hfProductID" />
        </td>
    </tr>
</tr>
<td>
    <asp:Button ID="btnAddToCart" runat="server" CommandArgu-
    ment='<## Eval("ProductID") %>' OnClick="btnAddToCart_Click"
    Text="Add To Cart" Width="100%" BorderStyle="Inset" Border-
    Width="1px" BorderColor="Black" CausesValidation="false" />
</td>
</tr>
</table>
</div>
</ItemTemplate>
<ItemStyle Width="33%" />
</asp:DataList>

```

**Code Snippet 7.** Code for displaying products

## 5.2.2 User Registration

The user needs to be a registered member to be able to buy products from this web shop. A registration form has been used for the user. After a successful registration, the user's data will be stored in the database table named as "Registration." A registration button called "Sign up" is available under the "Your Account" tab. When a user clicks the "Sign up" button, a sign-up form will appear to the user.

A function called "RegisterUser" will be responsible for processing and validate all the data submitted in the registration form before storing to the database table named as "Registration." The user must fill in all the required fields into the form to be able to register. Otherwise, it will show an error message. The code is given below:

```
public string RegisterUser(User user)
{
    //Check if User exists
    string query = string.Format("select count(*) from
Registration where Username='{0}'", user.Username);
    command.CommandText = query;

    try
    {
        conn.Open();//opening connectionstring
        int amountOfUsers = (int)command.ExecuteScalar();

        if(amountOfUsers < 1)
        {
            //User does not exist, create a new User
            query = string.Format("insert into Registration val-
ues ('{0}', '{1}', '{2}', '{3}', '{4}', '{5}', '{6}',
'{7}', '{8}')" , user.FirstName, user.LastName,
            user.Username, user.Password, user.Confirm-
            Password, user.Email, user.Address, user.City,
            user.Country);

            command.CommandText = query;
            command.ExecuteNonQuery();

            return "User has been registered Success-
fully!";
        }
        else
        {
            return "User already exists!";
        }
    }
}
```

```

    }
}
finally
{
    //closing connection
    conn.Close();
    command.Parameters.Clear();
}
}"/11/

```

### **Code Snippet 8. User Registration Method**

After a successful registration, a display message will be shown to the user that he is registered successfully. A query string has been executed to save data to the respective database table.

#### **5.2.3 User Login**

A user must log in to be able to check out into the system. The user must provide a valid user id and password. Otherwise, ASP.net validation properties will not allow the user to login to the system. When a user clicks the login button, a page called “Login.aspx” will be executed, and a login form will be available to the user. A function named as “UserLogin” will handle all the logics and process to make a connection to the database and to make the login process successful. The user login code is given below:

```

“public User UserLogin(string Username, string Password)
{
    //Check if user exist
    string query = string.Format("select
count(*) from Registration where Username='{0}' and
Password='{1}'", Username, Password);

    command.CommandText = query;

    try
    {
        conn.Open();

        int amountOfUsers = (int)command.ExecuteScalar();

        if(amountOfUsers > 0)
        {

```

```

        //as User Exists, check if Password match
        query = string.Format("select Password from
Registration where Username='{0}'", Username);
        command.CommandText = query;
        string dbPassword = command.ExecuteScalar().ToString();

        if (Password.Equals(dbPassword))
        {
            //As Password match. Password and Login are
            known to us
            //Retrive rest of the user data from database
            query = string.Format("select Email, FirstName,
LastName, ConfirmPassword, Address, City, Coun-
try from Registration where Username='{0}'",
Username);
            command.CommandText = query;

            SqlDataReader sdr = command.ExecuteReader();

            User user = null;

            while (sdr.Read())
            {
                string Email = sdr.GetString(0);
                string FirstName = sdr.GetString(1);
                string LastName = sdr.GetString(2);
                string ConfirmPassword = sdr.GetString(3);
                string Address = sdr.GetString(4);
                string City = sdr.GetString(5);
                string Country = sdr.GetString(6);
                user = new User(FirstName, LastName,
Username, Password, ConfirmPassword, Address,
City, Country, Email);

                }//closing of while

                return user;
            }//closing inner if
            else
            {
                //Password does not match
                return null;
            }
        }//closing of outer if
        else
        {
            //user does not exist
            return null;
        }
    }//try close
finally

```

```

    {
        conn.Close();//closing connectionstring
    }
}”/11/

```

### **Code Snippet 9.** User Login method

A string query is executed to fetch the correct login credentials from the database to match to the submitted data by the customer. After a successful login, the user is redirected to the homepage of this application. The user will be able to view his name and a logout button.

### **5.2.4 Payment Through PayPal**

In this project, the user has two options to pay by using PayPal. If a user owns a PayPal account, he can pay through the PayPal account by using a valid email and password. Otherwise, PayPal accepts credit card or debit card. The PayPal payment system is so secure in this sense that seller could never see the payment details from the buyer.

A class named as “NVPAPICaller” has been defined for this project. All the essential functions have been declared inside this class named as “ShortcutExpressCheckout,” “DoCheckoutPayment,” “GetCheckoutDetails” and “buildCredentials-NVPString.” /10/ PayPal important variables are given below:

```

“private const string CVV2 = "CVV2";
private string host = "www.paypal.com";
private string pEndPointURL_SB = "https://api-3t.sand-
box.paypal.com/nvp";
private const bool bSandbox = true;
private string pEndPointURL = "https://api-3t.pay-
pal.com/nvp";

private string host_SB = "www.sandbox.paypal.com";
private const string SIGNATURE = "SIGNATURE";
private const string PWD = "PWD";
private const string ACCT = "ACCT";”

```

### **Code Snippet 10.** PayPal function variables.

Live strings connect to the PayPal domain and sandbox strings communicate to PayPal's sandbox API and domain. APIUsername, APIPassword, and APISignature have been used to make a facilitator or user account into PayPal sandbox.

The below three functions are utilized to make purchases:

- SetExpressCheckout method
- DoExpressCheckoutPayment method
- GetExpressCheckoutDetails method

The ShortCutExpressCheckout function receives the product details and the test purchase information from the shopping cart. Then it will call the SetExpressCheckout function. Purchase details will be confirmed by GetCheckoutDetails method, and this method calls GetExpressCheckoutDetails method before completing the test purchase. DoExpressCheckoutPayment will be called from the testing environment to complete the purchase by method DoCheckoutPayment. When a consumer clicks the PayPal checkout button from the shopping cart page, PayPal checkout process will start. After a successful checkout, the customer will receive a transaction number made by PayPal which will be stored in the database table named as "Orderdetails." The code details for the ShortcutExpressCheckout is given below:/10/

```
public bool ShortCutExpressCheckout(DataTable dtResult,
string amt, ref string retMsg, ref string token)
{
    if (bSandbox)
    {
        pEndPointURL = pEndPointURL_SB;
        host = host_SB;
    }

    string returnUrl = "https://localhost:44355//Check-
out/CheckoutReview.aspx";
    string cancelURL = "https://localhost:44355//Check-
out/CheckoutCancel.aspx";

    NVPCodec encoder = new NVPCodec();
    encoder["METHOD"] = "SetExpressCheckout";
    encoder["RETURNURL"] = returnUrl;
    encoder["CANCELURL"] = cancelURL;
```

```

encoder["BRANDNAME"] = " 24 HOUR WEB SELL";
encoder["PAYMENTREQUEST_0_AMT"] = amt;
encoder["PAYMENTREQUEST_0_ITEMAMT"] = amt;
encoder["PAYMENTREQUEST_0_PAYMENTACTION"] = "Sale";
encoder["PAYMENTREQUEST_0_CURRENCYCODE"] = "USD";

Session["Currency"] = "$";

// Get the Shopping Cart Products

DataTable dtCartProducts = (DataTable)Session["MyCart"];

for (int i = 0; i < dtCartProducts.Rows.Count; i++)
{
    encoder["L_PAYMENTRE-
QUEST_0_NAME" + i] = dtCartProd-
ucts.Rows[i]["Name"].ToString();
    encoder["L_PAYMENTRE-
QUEST_0_AMT" + i] = dtCartProd-
ucts.Rows[i]["Price"].ToString();
    encoder["L_PAYMENTRE-
QUEST_0_QTY" + i] = dtCartProd-
ucts.Rows[i]["ProductQuantity"].ToString();
}

string pStrrequestforNvp = encoder.Encode();
string pStresponsenvp = HttpCall(pStrrequestforNvp);

NVPCodec decoder = new NVPCodec();
decoder.Decode(pStresponsenvp);

string strAck = decoder["ACK"].ToLower();
    if (strAck != null && (strAck == "suc-
cesswithwarning" || strAck == "success"))”
    {
        “token = decoder["TOKEN"];
        string ECURL = "https://" + host + "/cgi-bin/web-
scr?cmd=_express-checkout" + "&token=" + token;
        retMsg = ECURL;
        return true;
    }
else
{
retMsg = "ErrorCode=" + decoder["L_ERRORCODE0"] + "&" +
        "Desc=" + decoder["L_SHORTMESSAGE0"] + "&" +
        "Desc2=" + decoder["L_LONGMESSAGE0"];
        return false;
}”
}”

```



```
}
```

**Code Snippet 11.** Code for the ShortcutExpressCheckout method.

The code details for the GetCheckoutDetails is given below:/10/

```
“public bool GetCheckoutDetails(string token, ref string
PayerID, ref NVPCodec decoder, ref string retMsg)
{
    if (bSandbox)
    {
        pEndPointURL = pEndPointURL_SB;
    }

    NVPCodec encoder = new NVPCodec();
    encoder["METHOD"] = "GetExpressCheckoutDetails";
    encoder["TOKEN"] = token;

    string pStrrequestforNvp = encoder.Encode();
    string pStresponsenvp = HttpCall(pStrrequestforNvp);

    decoder = new NVPCodec();
    decoder.Decode(pStresponsenvp);

    string strAck = decoder["ACK"].ToLower();
        if (strAck != null && (strAck == "suc-
cess" || strAck == "successwithwarning"))
    {
        PayerID = decoder["PAYERID"];
        return true;
    }
    else
    {
        retMsg = "ErrorCode=" + decoder["L_ERRORCODE0"] + "&" +
            "Desc=" + decoder["L_SHORTMESSAGE0"] + "&" +
            "Desc2=" + decoder["L_LONGMESSAGE0"];

        return false;
    }
}”
```

**Code Snippet 12.** Code for the GetCheckoutDetails method.

A checkout return URL has been used to return to the shopping cart page. The customer can cancel the purchase by using checkout cancel URL. Two object variables called encoder and decoder have been used for receiving and sending data in

this function. The encoder is used to send data from the web application to the PayPal service and decoder is used to receive data returned from the PayPal. A data table and a Session have been utilized to receive data from this application. The code details to receive data from the application is given below:

```
DataTable dtCartProducts = (DataTable)Session["MyCart"];

for (int i = 0; i < dtCartProducts.Rows.Count; i++)
{
    encoder["L_PAYMENTREQUEST_0_NAME" + i] = dtCartProducts.Rows[i]["Name"].ToString();
    encoder["L_PAYMENTREQUEST_0_AMT" + i] = dtCartProducts.Rows[i]["Price"].ToString();
    encoder["L_PAYMENTREQUEST_0_QTY" + i] = dtCartProducts.Rows[i]["ProductQuantity"].ToString();
}
}
```

**Code Snippet 13.** Code to receive data from the application

### 5.2.5 Mail Transport System

An ASP.NET built-in mail transport system has been used to send a notification email to the customer's email address after a successful purchase. An order number and a link will be forwarded to the email of the client to track the order status. A method called "SendEmail" inside a class called "EmailEngine" has been developed to use for the mail transport system. The client must enter a valid email address to get the notification to his email address.

The code is given below:

```
public static class EmailEngine
{
    public static void SendEmail(String recipientEmail,
    string subject, string body)
    {
        using (MailMessage mailMessage = new MailMessage())
        {
            mailMessage.From = new MailAddress(Configuration-
            Manager.AppSettings["username"]);
            mailMessage.Subject = subject;
            mailMessage.Body = body;
        }
    }
}
```

```

        mailMessage.IsBodyHtml = true;
        mailMessage.To.Add(new MailAddress(receipientEmail));
        SmtplibClient smtp = new SmtplibClient();
        smtp.Host = ConfigurationManager.AppSettings["Host"];
        smtp.EnableSsl = Convert.ToBoolean(ConfigurationManager.AppSettings["EnableSsl"]);
        System.Net.NetworkCredential NetworkCred = new System.Net.NetworkCredential();
        NetworkCred.UserName = ConfigurationManager.AppSettings["UserName"];
        NetworkCred.Password = ConfigurationManager.AppSettings["Password"];
        smtp.UseDefaultCredentials = true;
        smtp.Credentials = NetworkCred;
        smtp.Port = int.Parse(ConfigurationManager.AppSettings["Port"]);
        smtp.Send(mailMessage);
    }
}
}

```

**Code Snippet 14.** Code for the SendEmail method

A file called “Web.config” has been modified to complete the mail transport system. As the SendEmail method needs a sender email address, this application must provide a sender email address.

```

<add key="Host" value="smtp.gmail.com"/>
<add key="EnableSsl" value="true"/>
<add key="UserName" value="mahfuz552717@gmail.com"/>
<add key="Password" value="#####"/>
<add key="Port" value="587"/>

```

**Code Snippet 15.** Web-Config code

Under the app settings tag, several keys such as Host, EnableSsl, UserName, Password and Port number have been applied to support the SendEmail method to complete the process.

### 5.2.6 Search Products by Keyword

A method called “GetProducts” will call an instance of class “ShoppingCart.” If the keyword searched in the search textbox match or compares the product name stored in the database, “GetProducts” will display the result to the user. Otherwise, it will display a message to the browser that no products are available for the searched keyword. The code for the method is given below:

```
private void GetProducts(string ProductNameSearched)
{
    ShoppingCart k = new ShoppingCart()
    {
        ProductName = ProductNameSearched
    };

    dlProducts.DataSource = null;
    dlProducts.DataSource = k.GetAllProductsBySearch();
    dlProducts.DataBind();
}
```

#### **Code Snippet 16.** Search products by keyword

A store procedure called “SP\_GetAllProductsBySearch” will be executed into the database and it will return the compared or matched products’ details to the front-end user. SQL query for “GetAllProductsBySearch” is given below:

```
CREATE PROCEDURE [dbo].[SP_GetAllProductsBySearch]
(
    @ProductName varchar(100)
)
AS
BEGIN

BEGIN TRY

SELECT * FROM (SELECT P.ProductID, P.ImageUrl, P.CategoryID,
P.Price, P.Name, C.CategoryName, P.ProductQuantity,
        Isnull(Sum(CP.TotalProduct), 0 as ProductSold,
        (P.ProductQuantity - Isnull(Sum(CP.TotalProduct), 0))
        as AvailableStock from Products P
        Inner join Category C
        on C.CategoryID = P.CategoryID
        left join CustomerProducts CP
        on P.ProductID=CP.ProductID
        Group by P.ImageUrl, P.ProductID,
```

```

        P.ProductQuantity,P.Price,
        C.CategoryName, P.Name,
        P.CategoryID) StockTable
        where AvailableStock > 0
        and Name like '%' + @ProductName + '%'

END TRY

```

**Code Snippet 17.** Store procedure for GetAllProductsBySearch method

### 5.2.7 View Orders by Admin

Admin can check all the orders, the products' quantity, sub-total, and details of the products made by all the customers. An ASP page called "CustomerOrders" has been implemented to view orders by the customers. An ASP grid view tag has been used for displaying all the data in the admin management panel. A code page behind "CustomerOrders" has been implemented to fetch the required data from the database. A function named as "GetOrdersList" uses an instance of the Shopping-Cart class to communicate to the database. The code for viewing all the orders is given below:

```

private void GetOrdersList()
{
    ShoppingCart k = new ShoppingCart
    {
        Flag = 0
    };

    DataTable dt = k.GetOrdersList();
    gvCustomerOrders.DataSource = dt;
    gvCustomerOrders.DataBind();
}

```

**Code Snippet 18.** Code for the GetOrdersList method

A store procedure called "SP\_GetOrdersList" fetch the data from the database and return the result for display. The code is given below:

```

USE [ShoppingCartDB]
GO

```

```

/***** Object: StoredProcedure [dbo].[SP_GetOrdersList]
Script Date: 29/03/2017 21:40:22 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[SP_GetOrdersList]
(
@Flag int
)
AS
BEGIN

    BEGIN TRY

        if(@Flag <> 0)
        Begin
        select * from CustomerDetails where Id=@Flag;
        end
        else
        begin
            select * from CustomerDetails
        end

    END TRY

    BEGIN CATCH

-- Insert INTO dbo.ErrorLog
-- VALUES(ERROR_MESSAGE(), 'SP_GetAllData')
        PRINT ('Error occured')

    End catch
END
GO

```

**Code Snippet 19.** Store procedure for the GetOrdersList method

## 6. APPLICATION TESTING

It is crucial to test the online application to ensure the reliability and the accuracy of the functions. Tests have been applied extensively to all the components, on the integration system and at a different level of the application's development. Without the test, there is a possibility that the application could crash anytime. Tests have been performed on both functional modules of this application, which is User module and Admin module.

### 6.1 User Login Testing

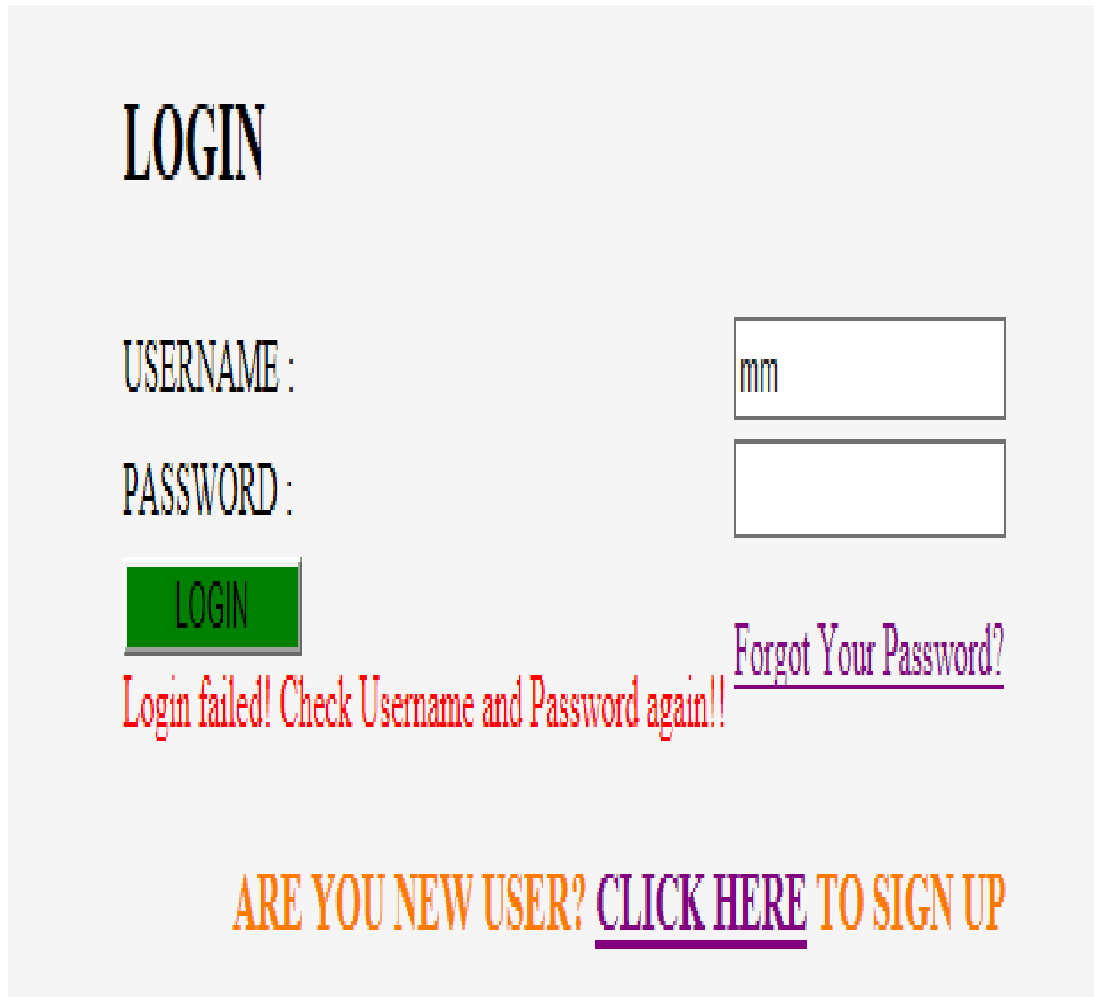
The user must log in to purchase products from this system. The user must provide a valid username and password to login into the system. When a user clicks the login button without submitting login credentials, a red (\*) sign will be displayed after each field. Figure 41 illustrates the test result.



The screenshot shows a login form titled "LOGIN". It contains two input fields: "USERNAME :" and "PASSWORD :". Both fields are empty and have a red asterisk (\*) to their right, indicating a validation error. Below the "PASSWORD :" field is a green "LOGIN" button. To the right of the "PASSWORD :" field is a purple link that says "Forgot Your Password?". At the bottom of the form, there is a line of text in orange and purple: "ARE YOU NEW USER? [CLICK HERE](#) TO SIGN UP".

**Figure 41.** Login Test Case Without Username and Password

If a user submits the wrong username and password, a warning message will be shown to the user. Figure 42 illustrates the test result for invalid login credentials.



**Figure 42.** Login Test Case Invalid Username and Password

## 6.2 User Shopping Cart Testing

The user is able to add products to the shopping cart by clicking “Add To Cart” on the web page. Figure 43 shows the test result after adding a product to the shopping cart.



24HOURSSHOPPING.COM  
ENJOY 24 HOURS SHOPPING ON WEB

Search

Login

YOUR ACCOUNT

---


Your Shopping Cart Products

Customer Details

**Please review your order!**


1 \* Macbook Pro                      1500 €  
**Total To Pay:**                      **1500 €**

**Macbook Pro**



AvailableStock: 21  
Price:1500 \*

RemoveFromCart



The safer, easier way to pay

Continue Shopping

Full Name:

PhoneNo:

EmailID

Address

Total Products :

Total Price :

Product Name :

Shipping and Handling :

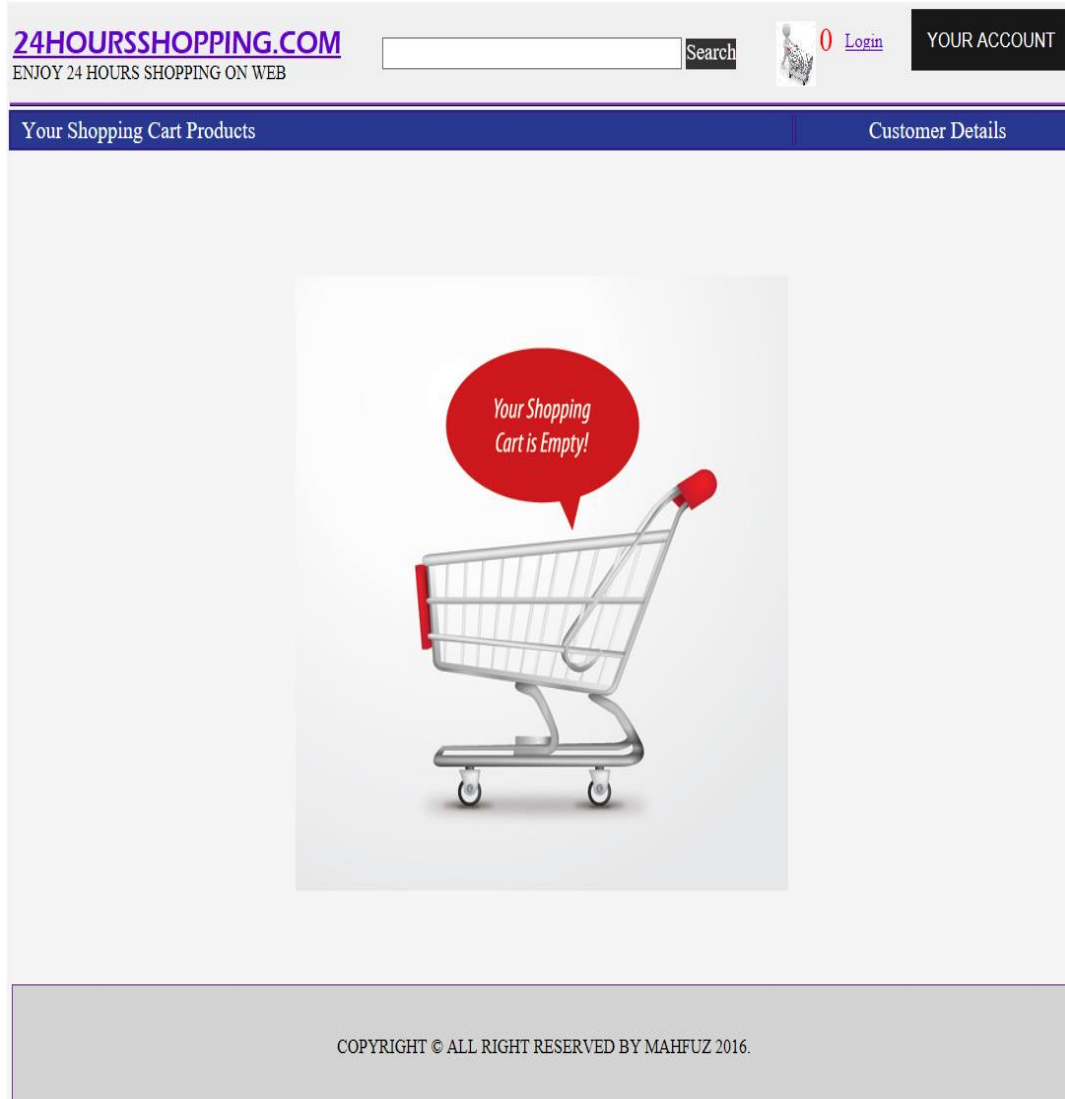
FREE SHIPPING  
 EXPRESS DELIVERY  
 REGULAR DELIVERY

Payment Mode :

1. PAYPAL PAYMENT  
 2. BANK TRANSFER  
 3. CARD PAYMENT

**Figure 43.** Test Result after Adding Product to the shopping Cart

The customer can delete products from the shopping cart or continue shopping without checkout process. After removing the product from the shopping cart, an empty shopping cart will be displayed to the user. Figure 44 illustrates the test result after removing products from the shopping cart.



**Figure 44.** Test Result after Deleting Products from Shopping Cart

### 6.3 Admin Login Testing

Admin must log in to enter the Admin management panel. Admin login testing is like user login test. If an admin tries to log in without submitting the LoginId and Password, a red (\*) sign will appear after each field. Figure 45 shows the login attempt view without submitting login credentials.

The image shows a web form titled "AdminPanel" in blue text at the top center. Below the title, there are two input fields. The first is labeled "LoginId :" and the second is labeled "Password :". Both fields are empty. To the right of each input field is a small red asterisk (\*). Below the password field is a grey button with the text "Login".

**Figure 45.** Login Test without Login Credentials

If an admin submits the wrong username and password, a warning message will be shown to the user. Figure 46 shows the login attempt view using invalid login credentials by an Admin.

The image shows the same "AdminPanel" login form. The "LoginId" field now contains the text "mm" and has a small 'x' icon in its top right corner. The "Password" field remains empty. The "Login" button is still present. At the bottom of the form, centered, is a red error message that reads "Incorrect LoginId/Password".

**Figure 46.** Invalid Login Attempt By Admin

## 6.4 Adding Products Testing

Only an admin can add products to the system. Even admin can edit and delete products from the system. Figure 47 shows the available products from the database server.

ProductID	Name	Description	Price	ImageUrl	CategoryID	ProductQuanti...
1	Samsung LED TV	Full HD ReadyS...	1300	~/ProductImag...	1	6
2	Dell Laptop	13,3 inch i5 Inte...	900	~/ProductImag...	12	12
3	Dell Laptop2	15 inch Intel i3 ...	800	~/ProductImag...	12	5
4	Intex TV	55 inch HD Rea...	1100	~/ProductImag...	1	3
5	iPhone 6	iPhone 664 gb i...	650	~/ProductImag...	23	10
6	iPhone 5c	iPhone 5cDispl...	299	~/ProductImag...	23	8
8	iPhone 7	128 gb 8gb ram	900	~/ProductImag...	23	18
9	Microsoft lumi...	8gb internal me...	225	~/ProductImag...	23	25
10	Macbook Pro	Macbook Pro i/ ...	1500	~/ProductImag...	12	50
13	Canon EOS 70D	Sensor size:337...	1200	~/ProductImag...	45	20
14	Canon EOS 700D	Megapixel: 17.9...	900	~/ProductImag...	45	10
15	Sony Playstatio...	CPU:Low powe...	399	~/ProductImag...	56	12
16	Polar Smart Wa...	Operating syste...	299	~/ProductImag...	89	6
17	Polar Smart Wa...	Battery type: 18...	340	~/ProductImag...	67	15
18	Dell Laptop	Windows home...	1190	~/ProductImag...	12	5
19	Sony Playstatio...	Sound:Speakers...	198	~/ProductImag...	56	8
20	Canon EOS 7D	18MP APS-C C...	2700	~/ProductImag...	45	4
21	Canon EOS 70D	Sensor size:337...	1200	~/ProductImag...	45	20
23	Canon EOS 70D	Sensor size:337...	1200	~/ProductImag...	45	20
NULL	NULL	NULL	NULL	NULL	NULL	NULL

**Figure 47.** Available Products from Database

The product has been added to the system by Admin. The product will be available to the web page immediately after it is added to the database. Figure 48 shows the test result for adding a product to the database of this system.

	ProductID	Name	Description	Price	ImageUrl	CategoryID	ProductQuanti...
▶	1	Samsung LED TV	Full HD ReadyS...	1300	~/ProductImag...	1	6
	2	Dell Laptop	13,3 inch i5 Inte...	900	~/ProductImag...	12	12
	3	Dell Laptop2	15 inch Intel i3 ...	800	~/ProductImag...	12	5
	4	Intex TV	55 inch HD Rea...	1100	~/ProductImag...	1	3
	5	iPhone 6	iPhone 664 gb i...	650	~/ProductImag...	23	10
	6	iPhone 5c	iPhone 5cDispl...	299	~/ProductImag...	23	8
	8	iPhone 7	128 gb 8gb ram	900	~/ProductImag...	23	18
	9	Microsoft lumi...	8gb internal me...	225	~/ProductImag...	23	25
	10	Macbook Pro	Macbook Pro i/...	1500	~/ProductImag...	12	50
	13	Canon EOS 70D	Sensor size:337....	1200	~/ProductImag...	45	20
	14	Canon EOS 70D	Megapixel: 17.9...	900	~/ProductImag...	45	10
	15	Sony Playstatio...	CPU:Low powe...	399	~/ProductImag...	56	12
	16	Polar Smart Wa...	Operating syste...	299	~/ProductImag...	89	6
	17	Polar Smart Wa...	Battery type: 18...	340	~/ProductImag...	67	15
	18	Dell Laptop	Windows home...	1190	~/ProductImag...	12	5
	19	Sony Playstatio...	Sound:Speakers...	198	~/ProductImag...	56	8
	20	Canon EOS 7D	18MP APS-C C...	2700	~/ProductImag...	45	4
	21	Canon EOS 70D	Sensor size:337....	1200	~/ProductImag...	45	20
	23	Canon EOS 70D	Sensor size:337....	1200	~/ProductImag...	45	20
	24	Canon EOS 6d	The EOS 6D DS...	2000	~/ProductImag...	45	20
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**Figure 48.** Test Case Products Added By Admin

The product details marked with yellow is shown from the database table. Now this application will check if the product is available on the web page. Figure 49 shows the test result.

POPULAR PRODUCTS				Product Categories
Price:650 Stock: <b>2</b> Add To Cart	Price:1200 Stock: <b>20</b> Add To Cart	Price:340 Stock: <b>15</b> Add To Cart	Price:1200 Stock: <b>19</b> Add To Cart	<a href="#">TV, Audio &amp; Surveillance</a> <a href="#">Computers &amp; Tablets</a> <a href="#">Cellphones &amp; Acce</a> <a href="#">Cameras &amp; Photo</a> <a href="#">Video Games &amp; Consoles</a> <a href="#">Wearables</a> <a href="#">Vehicle Electronics &amp; GPS</a> <a href="#">Other Accessories</a>
<b>iPhone 5c</b>	<b>Canon EOS 700D</b>	<b>Dell Laptop</b>	<b>Canon EOS 70D</b>	
				
Price:299 Stock: <b>3</b> Add To Cart	Price:900 Stock: <b>8</b> Add To Cart	Price:1190 Stock: <b>5</b> Add To Cart	Price:1200 Stock: <b>18</b> Add To Cart	
<b>iPhone 7</b>	<b>Sony Playstation 4 new</b>	<b>Sony Playstation Vita</b>	<b>Canon EOS 6d</b>	
				
Price:900 Stock: <b>14</b> Add To Cart	Price:399 Stock: <b>10</b> Add To Cart	Price:198 Stock: <b>6</b> Add To Cart	Price:2000 Stock: <b>20</b> Add To Cart	
<b>Microsoft lumia 740</b>				
				
Price:225 Stock: <b>16</b> Add To Cart				

COPYRIGHT © ALL RIGHT RESERVED BY MAHFUZ 2016.

**Figure 49.** Products Added by Admin

A product circled by a blue marker illustrates that admin has successfully added the product to the system.

## 7. CONCLUSIONS

E-commerce web application is very popular nowadays not only to the consumers but also to the seller. People are more addicted to the online shopping in this decade. Almost everything is possible to buy from the online shop. The Internet is the largest source for online shopping around the world. An online shop is not complete without a payment method. The PayPal payment method has been used for this application, which is quite easy to use. A seller must make a merchant/ business account to PayPal to get the money from the buyer to the seller's PayPal account. On the other hand, a buyer must make a personal account through PayPal. The Buyer needs to provide his debit/credit card information to the PayPal to make an account. When a buyer creates an account through PayPal, he can buy products from any online shop which supports PayPal payment process by using only email address and password. Seller will never be able to get the buyer's account information.

This e-commerce web application has been developed with ASP.NET and SQL server database system. PayPal payment gateway has been added to this application as the payment method. The user can browse and add products to the shopping cart, but will not be able to buy without being registered to the system. A registered customer can login to the system and buy products through the PayPal account. Customers' order information and shipping details will be saved in the SQL server database. An automatic email will be sent to the buyer's email address after a successful purchase.

An admin can add, edit and delete products in this system. Admin will update the shipping status for a completed purchase. Admin will have the authorization to update and remove a customer's information. All the orders made by the client will be shown to the admin management panel. And a sale report will be displayed to the admin on a daily, monthly and yearly basis. The sales report will be updated after each purchase.

This application has been implemented successfully based on the requirements made for the project. The purpose of the project was successful, and the goals were achieved. The application took a bit longer time than I expected, because of the difficulties to combine the PayPal payment method into the system. The Implementation of PayPal gateway in this application was one of the most challenging development processes. After spending time to study the PayPal payment method, the challenge was handled successfully. Mail transport system has been implemented to send a confirmation email to the users after a successful purchase, which was another difficult feature of this application.

## **7.1 FUTURE WORKS**

Registration and login to the system by using social media will be easier to the users. As mobile applications are more demanding to the consumers nowadays, a mobile application will be developed in future for this application. The shipping method will be updated later depending on where this application will be used. Currently, this application uses a free shipping method. A more powerful security level will be implemented in future to the application payment gateway as well to the database server. Recovering a forgotten password and sending a new password to the user's email address will be implemented later to the login method of the application.



## REFERENCES

- /1/ InnovationPei. E-commerce: Purchasing and selling online. Accessed 09.03.2017. [http://www.gov.pe.ca/photos/original/IPEI\\_ebiz\\_ecomm.pdf](http://www.gov.pe.ca/photos/original/IPEI_ebiz_ecomm.pdf)
- /2/ Tutorialpoints 2017. E-Commerce B2C model. Accessed 9.3.2017. [http://www.tutorialspoint.com/e\\_commerce/e\\_commerce\\_b2c\\_mode.htm](http://www.tutorialspoint.com/e_commerce/e_commerce_b2c_mode.htm)
- /3/ Chauhan, S. 2013. A brief history of ASP.NET. Accessed 10.3.2017. <http://www.dotnettricks.com/learn/aspnet/a-brief-version-history-of-aspnet>
- /4/ Microsoft 2017. SQL SERVER MANAGEMENT STUDIO. Accessed 11.03.2017. <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms>
- /5/ W3Schools 2017. Introduction to SQL. Accessed 11.03.2017. [https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp)
- /6/ Wagner, B. 2017. Introduction to the C# Language and the .NET Framework. Accessed 11.03.2017. <https://msdn.microsoft.com/en-us/library/z1zx9t92.aspx>
- /7/ W3Schools 2017. HTML5 Introduction. Accessed 11.03.2017. [https://www.w3schools.com/html/html5\\_intro.asp](https://www.w3schools.com/html/html5_intro.asp)
- /8/ Konrud. 2017. Mozilla Developer Network. CSS3. Accessed 11.03.2017. <https://developer.mozilla.org/en-US/docs/Web/CSS/CSS3>
- /9/ PayPal 2017. About us. Accessed 11.03.2017. <https://www.sandbox.paypal.com/fi/webapps/mpp/about>
- /10/ Raitan, E. 2014. Checkout and Payment with PayPal. Accessed 26.03.2017. <https://docs.microsoft.com/en-us/aspnet/web-forms/overview/getting-started/getting-started-with-aspnet-45-web-forms/checkout-and-payment-with-paypal>
- /11/ Mondal, R. 2015. Online Car Selling Application using PayPal. Accessed 18.03.2017. <http://theseus.fi/bitstream/handle/10024/87267/Online%20Car%20Selling%20Application%20Using%20Pay-Pal.pdf?sequence=1>