Table of Contents

[1.0 Introduction 2](#_Toc503889119)

[1.1 Objectives, Scope, Targeted Audience 2](#_Toc503889120)

[1.2 Project Schedule 3](#_Toc503889121)

[1.3 Summary of Major Functions 5](#_Toc503889122)

[2.0 Requirement Specification 5](#_Toc503889123)

[2.1 Audience Modelling 5](#_Toc503889124)

[2.1 Audience Classification 5](#_Toc503889125)

[2.2 Audience Characterization 5](#_Toc503889126)

[2.2.1 Administrator 5](#_Toc503889127)

[2.2.2 Visitor 6](#_Toc503889128)

[2.2.3 Customer 6](#_Toc503889129)

[2.2 Major Functions 6](#_Toc503889130)

[3.0 Design and Modelling 7](#_Toc503889131)

[3.1 Database Schema 7](#_Toc503889132)

[3.2 Interface Design 8](#_Toc503889133)

[3.2.1 Admin Pages 8](#_Toc503889134)

[3.3 Website Architecture 23](#_Toc503889135)

[3.3.1 Navigational Structure 23](#_Toc503889136)

[3.3.2 Activity Diagram 24](#_Toc503889137)

[3.3.3 Concurrent Task Tree 27](#_Toc503889138)

[4.0 Implementation 30](#_Toc503889139)

[4.1 Login & MasterPage 30](#_Toc503889140)

[4.2 Admin Add Product 32](#_Toc503889141)

[4.3 Product Catalogue 35](#_Toc503889142)

[4.4 Product Description 39](#_Toc503889143)

[4.5 Cart 41](#_Toc503889144)

[4.6 Delete 43](#_Toc503889145)

[4.7Payment 44](#_Toc503889146)

[5.0 Conclusion 45](#_Toc503889147)

[6.0 References 46](#_Toc503889148)

# 1.0 Introduction

Sportify is a sportswear retailer based in Malaysia which was founded in 2014 and it is known as one of the biggest retailer which sells some of the biggest sports **brands** such as Nike, Adidas, and Puma. With its fashion-based retailing, it stocks items suitable for **professionals** and **novices** who wants to remain on trend. A few examples of the products it sells are sports apparel, equipment and sports shoes. Currently, despite having 10 retail stores in Malaysia, it does not have an online presence. With the increasing number in competitors such as JD Sports, Stadium and other sport retailers, it would be hard for Sportify to compete if it does not keep up with the demand and trends. Therefore, there is a need for the company to build a website and incorporating different technologies to sell their products and cater to a larger audience who are potential customers and keep up with the market trends and demands.

## 1.1 Objectives, Scope, Targeted Audience

One of our main objective is to develop a website which incorporates an **online store** to allow customers to have access to Sportify’s famous product which is its footwear and make an **online purchase** through a different platform to target **a wider range of audience**. Since technology has become a trend, usage of smartphones increases, which means there is an increase number of user online and Sportify must keep up with the demand. Furthermore, we want to incorporate interactive features such as allowing customers to submit **reviews**, contacting customer **support**, performing transactions and interaction with the information of the product offered. Aside from face-to-face interaction in shops, Sportify can interact with users online who are potential customers and increase sales. Lastly, we want to design the website structure and **layout of information** accordingly to ensure a good user experience. Instead of customer asking information about the products from sales assistants or browsing through the product tags at the store, a website can provide an increase in information richness, with its contents such as image slider, images and hyperlinks.

The project revolves around building a web application which **displays** the main product that Sportify is famous about; footwear, whereby customers may browse through and select an item, make the required **specifications** such as quantity and then it would be saved in a **shopping cart** which may contain multiple types of items to be checked out once signed in. The checkout is equipped with online **payment**. Customers can **filter** the items based category or type of footwear, such as sneakers and running shoes, and it will also provide information to customers about the **company** background, **store** locations, and **contact** information for customer service support.

The web application is targeted to **age groups** from 12 years old and above but focusing on **every-day use, running and training as few of the type of footwears available for purchase** due to trends of fashion in sneakers and gym training in Malaysia. The web app can be used to view products and make purchases. For youths **aged 12 and above** who do not have the **technical knowledge** or **disposable income** to purchase sporting shoes on their own can **have their parents** or guardian make the purchase on their behalf.

## 1.2 Project Schedule

|  |  |
| --- | --- |
| Week | Tasks |
| Week 1 | Assignment hand out day (10 October 2017).  First group meeting to decide on topic and scope of the system (12 October 2017). |
| Week 2 | Second group meeting to discuss what webpages to include and task division among group members (16 October 2017).  Working on introduction part of documentation, such as goals, mission, project scope, and targeted audience (17 - 20 October 2017). |
| Week 3 | Third group meeting to discuss audience modelling (23 October 2017).  Working on audience modelling and major functions of the system (24 - 27 October 2017). |
| Week 4 | Fourth group meeting to discuss about the diagrams, storyboard, and structure of the website (30 October 2017).  Working on the diagrams (31 October - 5 November 2017). |
| Week 5 | Working on code implementation based on work breakdown (6 – 12 November 2017). |
| Week 6 | Working on code implementation based on work breakdown (13 – 19 November 2017). |
| Week 7 | Working on code implementation based on work breakdown (20 – 26 November 2017). |
| Week 8 | Working on code implementation based on work breakdown (27 November – 3 December 2017). |
| Week 9 | Fifth group meeting to update on progress and discuss issues (4 December 2017)  Working on code implementation based on work breakdown (5 – 10 December 2017). |
| Week 10 | Working on code implementation based on work breakdown (11– 17 December 2017). |
| Week 11 | Working on code implementation based on work breakdown (18– 22 December 2017). |
| Week 12 | Year end break, no activities during the week (25 – 31 December 2017). |
| Week 13 | Year end break, no activities during the week (1 – 7 January 2018). |
| Week 14 | Testing and debugging of the system (8 – 12 January 2018).  Working on implementation part of the documentation (8 - 14 January 2018). |
| Week 15 | Sixth group meeting to conclude and compile documentation (15 – 16 January 2018). |

## 1.3 Summary of Major Functions

One of the major functions of the website is to enable customers to purchase Sportify’s footwear products online by browsing through the choices offered, adding it to a Wishlist or shopping cart if they want to purchase it with the available payment method offered. Aside from that, it allows customers to send inquiries or feedback to customer service support. Aside from customer as the main user, administrator is another type of user which would have a different set of functionalities such as managing products and transactions occurred.

# 2.0 Requirement Specification

## 2.1 Audience Modelling

### 2.1 Audience Classification

This website mainly targets millennials (20 to 35 years old) who are mainly sports participants as well as those who wears sports-inspired fashion. The target consists of both men and women in Malaysia from the middle class (those in college and working with an income of RM3000 and above). Whilst the smaller group includes anyone aged 14+ who purchase sportswear for hobbies or PE subject in school.

Thus, there are three audience classes that would be involved, which are Administrator, Direct User and third-party Online Payment Services provider.

### 2.2 Audience Characterization

Other than the Administrator who manage the content of the website, there are 2 distinguished direct users who use this application for a purpose and they are divided into 2 class variants which are Visitor and Customer. Below is the list of characteristics for each class.

### 2.2.1 Administrator

Description: Employees of Sportify who are in charge of managing content on the website.

Functionalities:

* Login to the system.
* View customer personal details and order details
* View, add, delete and modify item information
* View feedback from customer

### 2.2.2 Visitor

Description: users who express interest on the company by visiting the website

Functionalities:

* Register to the system.
* View products
* View company information
* Send inquiry/feedback

### 2.2.3 Customer

Description: users who have an account and attempt to purchase the company’s product(s) on the website

Functionalities:

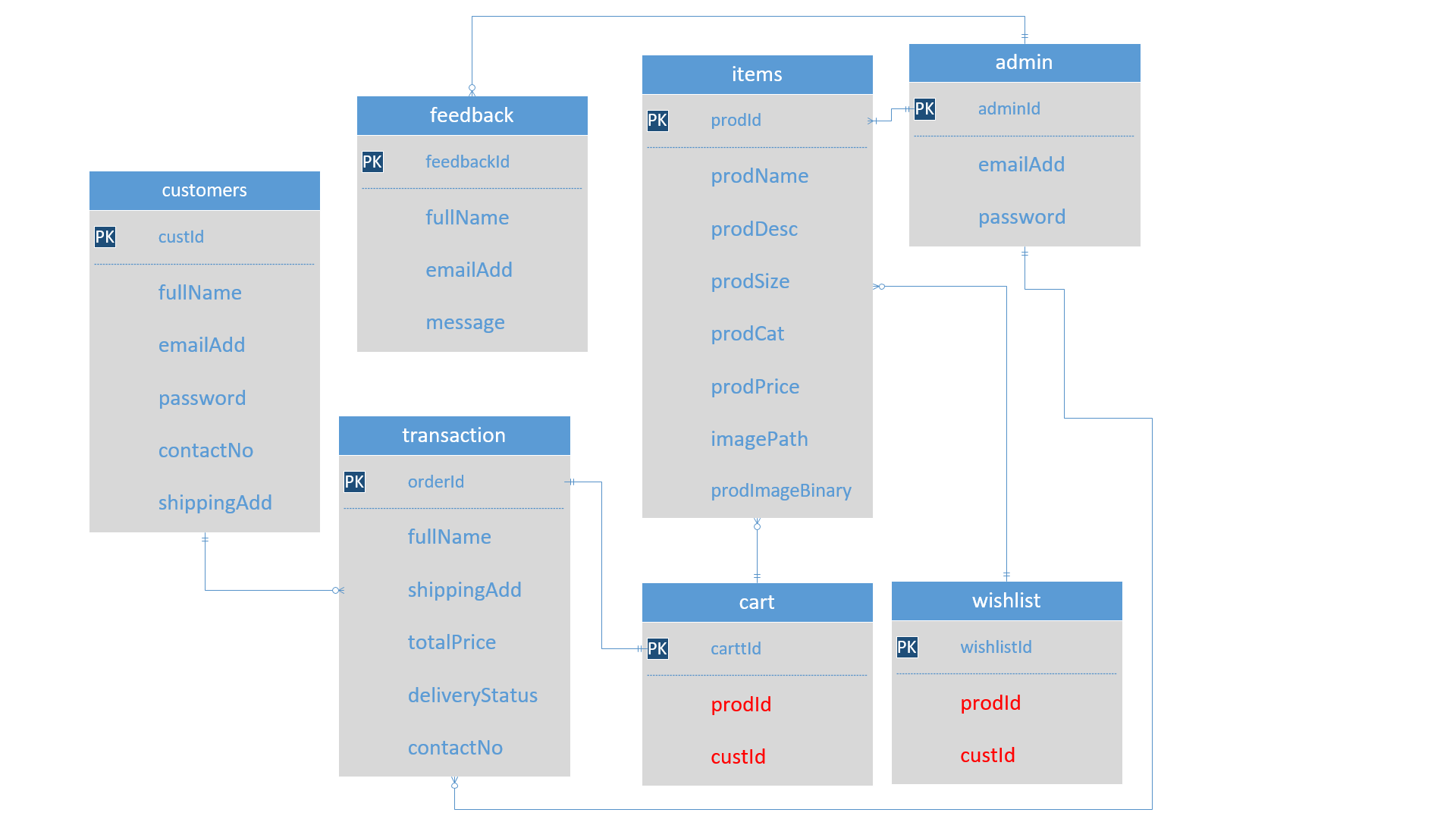
* Login to the system.
* Add and modify personal information (delivery address, payment details, etc)
* View products
* View, add, and delete items from wishlist and cart
* View receipt of purchase
* Make payment by purchasing online or opting for COD
* Submit feedback

## 2.2 Major Functions

Both visitors and customers can view the product catalogue based on category and filter the items based on price. They can then click on a particular item to view more details of the item. However, visitors can only view the products but not add it to their wishlist or cart to be purchased. In order to do so, they need to login or sign up to be a registered user to purchase items. Once their account has been created, they can login using the correct credentials which grants them member privileges such as adding product into wishlist and cart and proceeding to checkout to purchase the items. At checkout, customer will fill in the necessary details for payment and delivery, and once the item is purchased a receipt will be displayed to them. Staff of the website can view, add, update, and delete products; view all orders and update the delivery status; view customer details and feedback received.

# 3.0 Design and Modelling

## 3.1 Database Schema



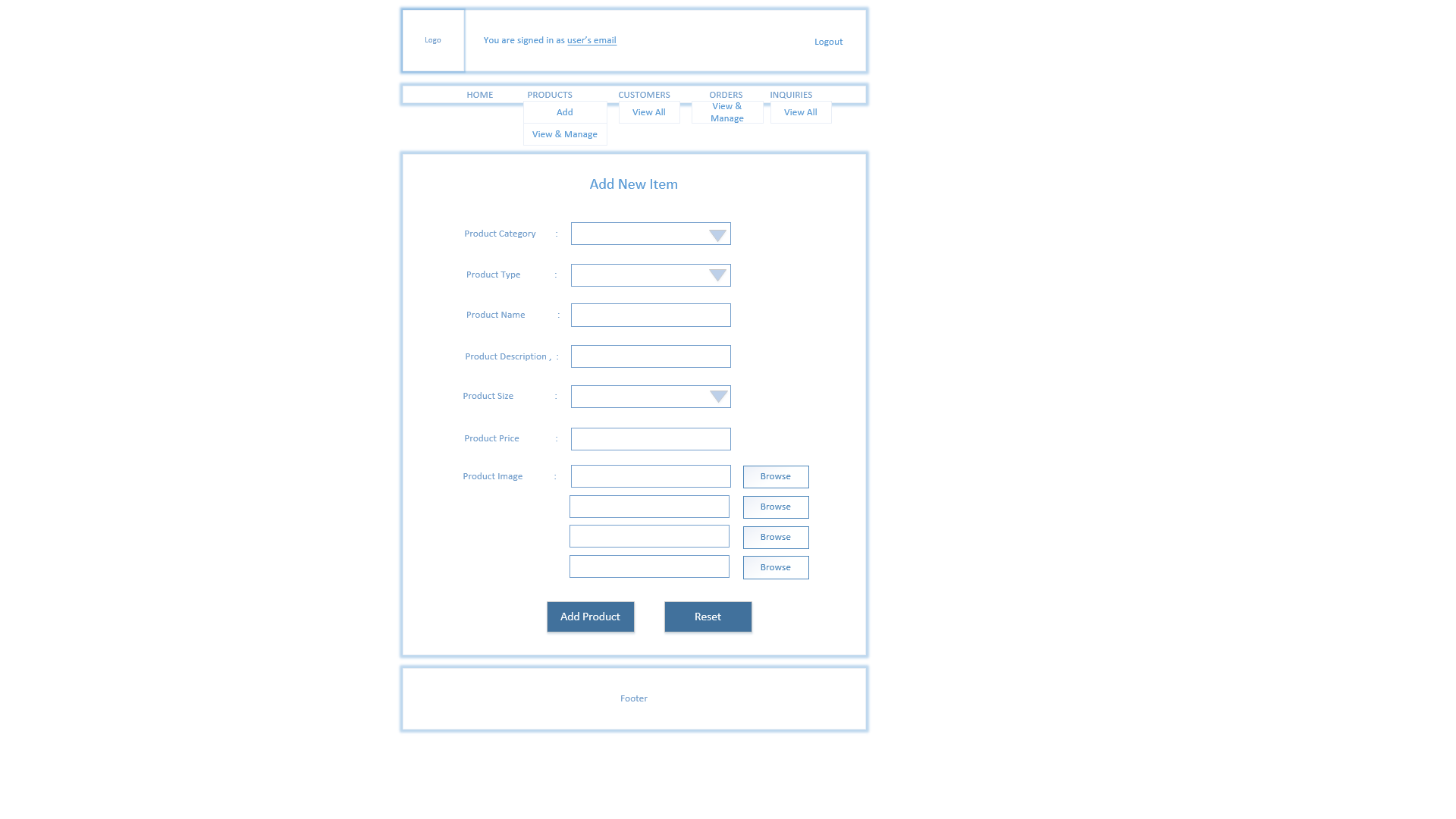
## 3.2 Interface Design

### 3.2.1 Admin Pages

Admin Home



Admin Add Item



Admin Manage Item



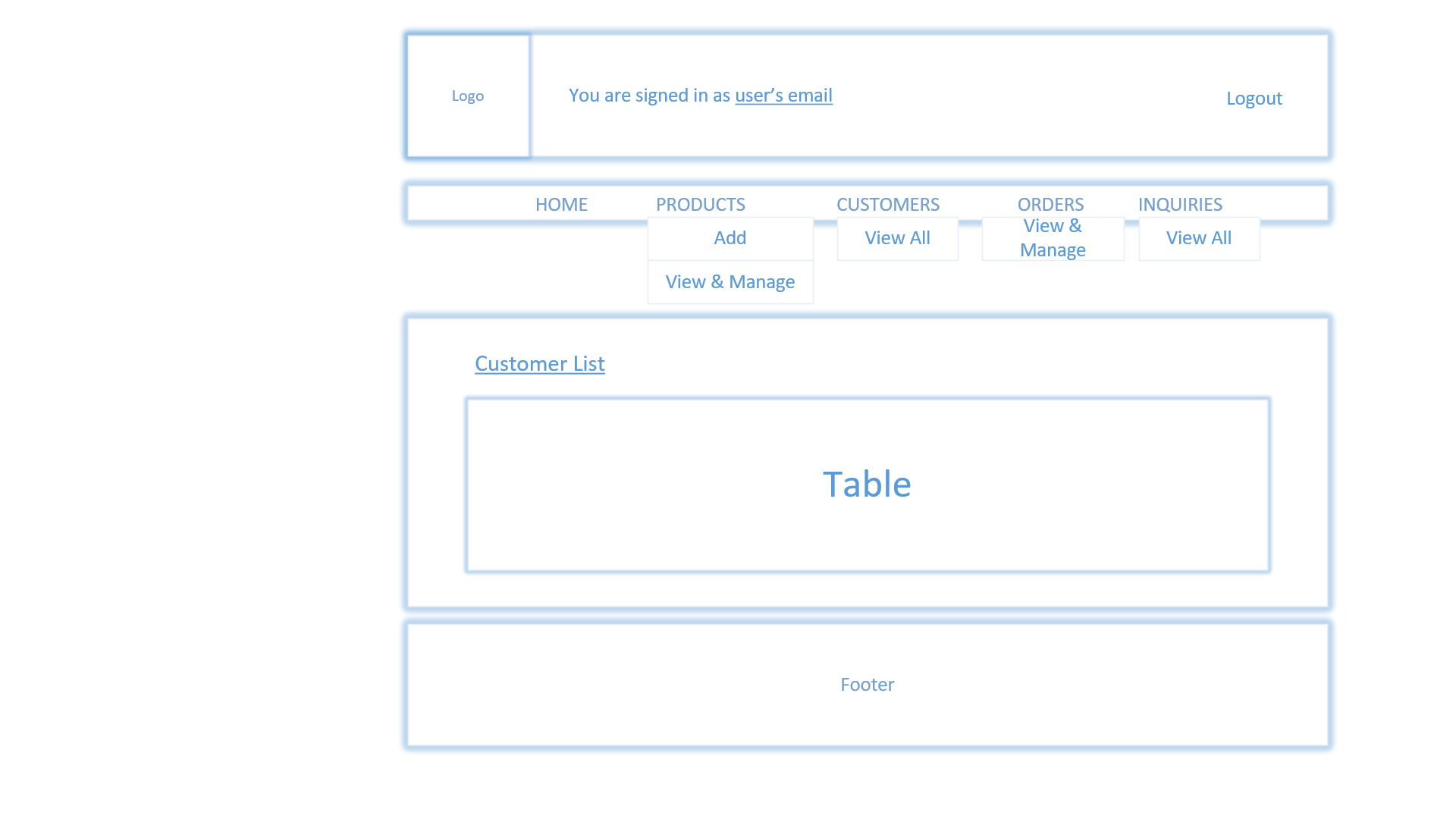
Admin View & Manage Products



Admin View & Manage Customer Orders



Admin View Customers



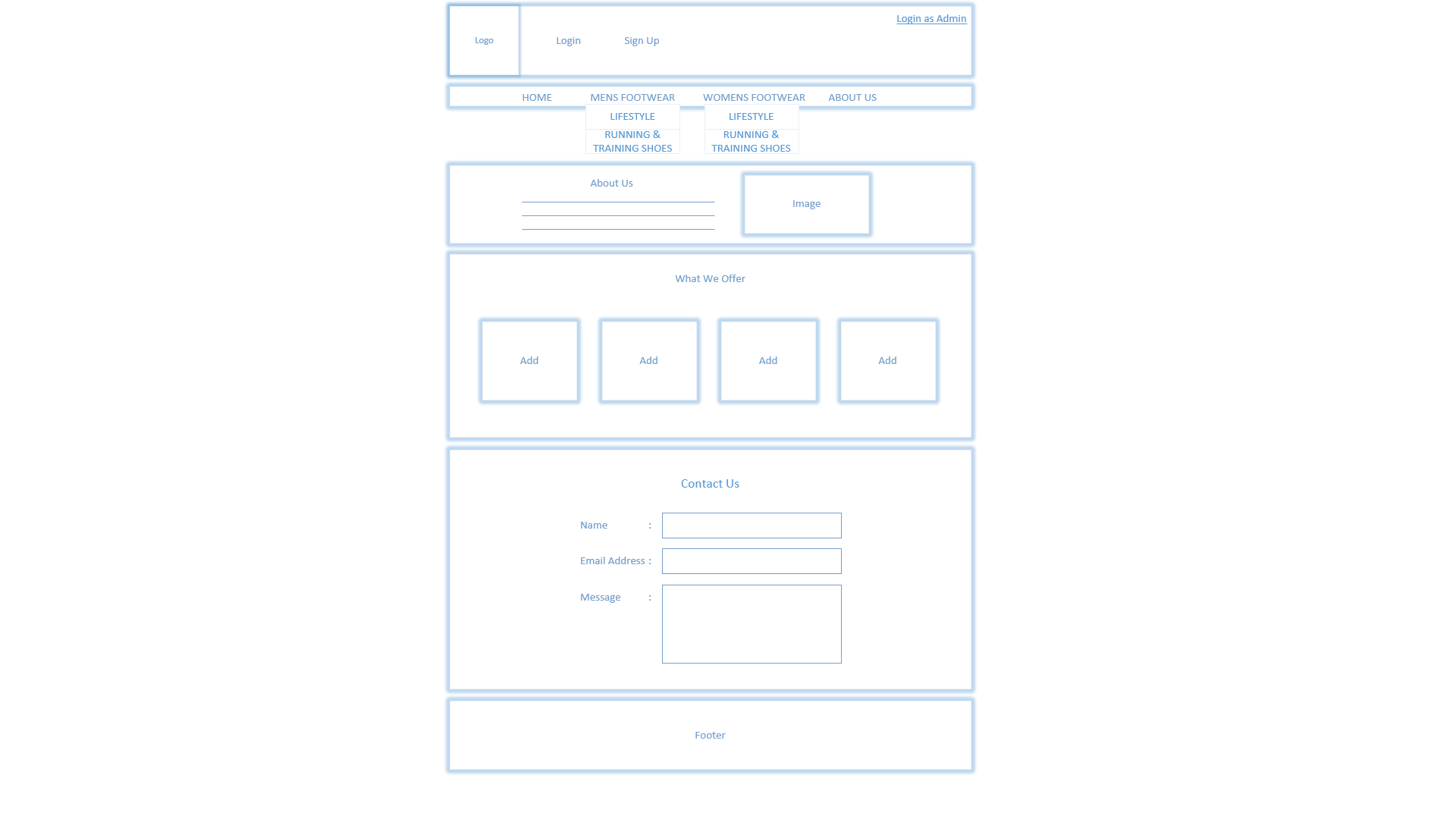
Admin View Customer Enquiry



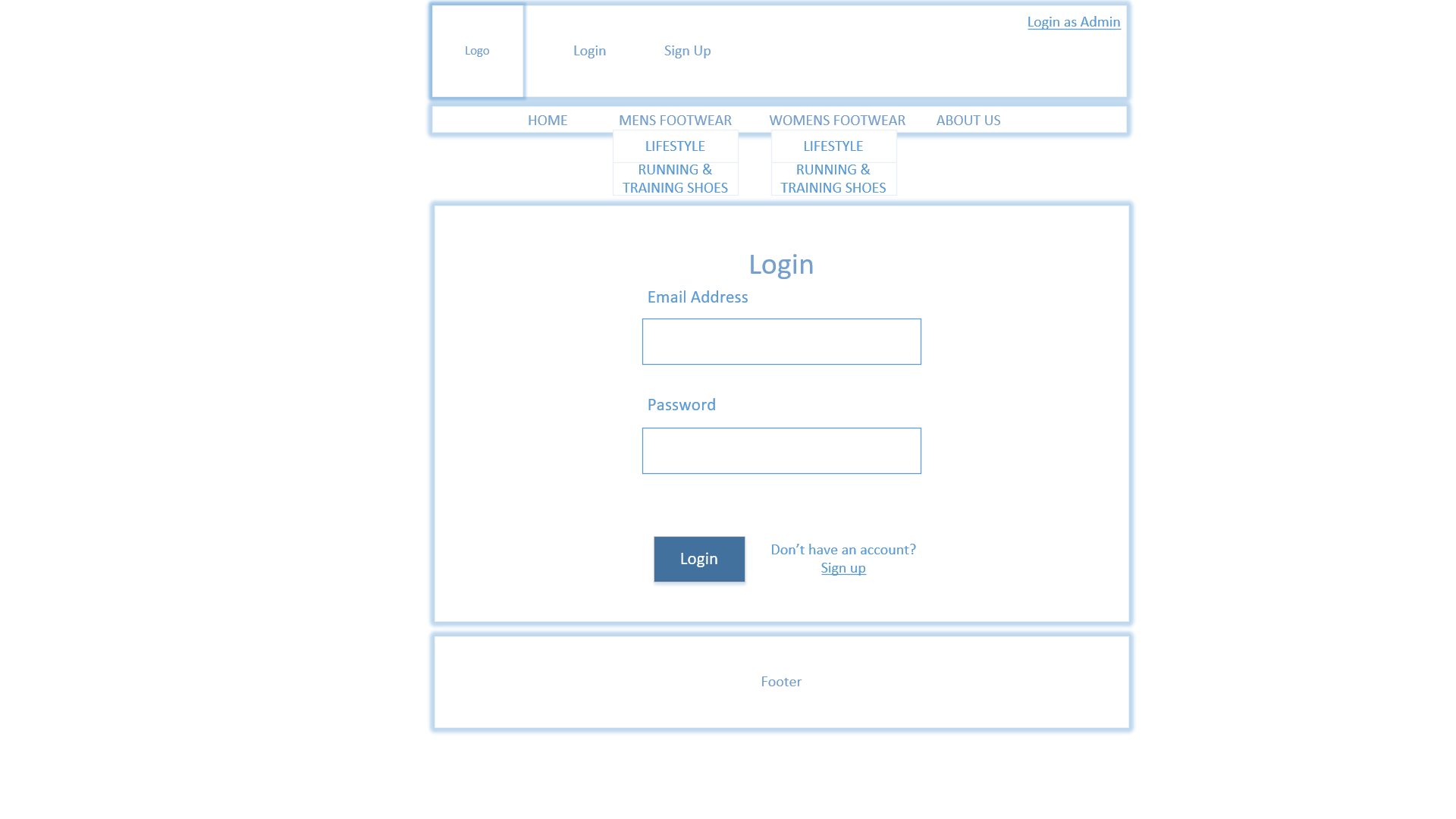
Default Homepage



About Us



Login Page

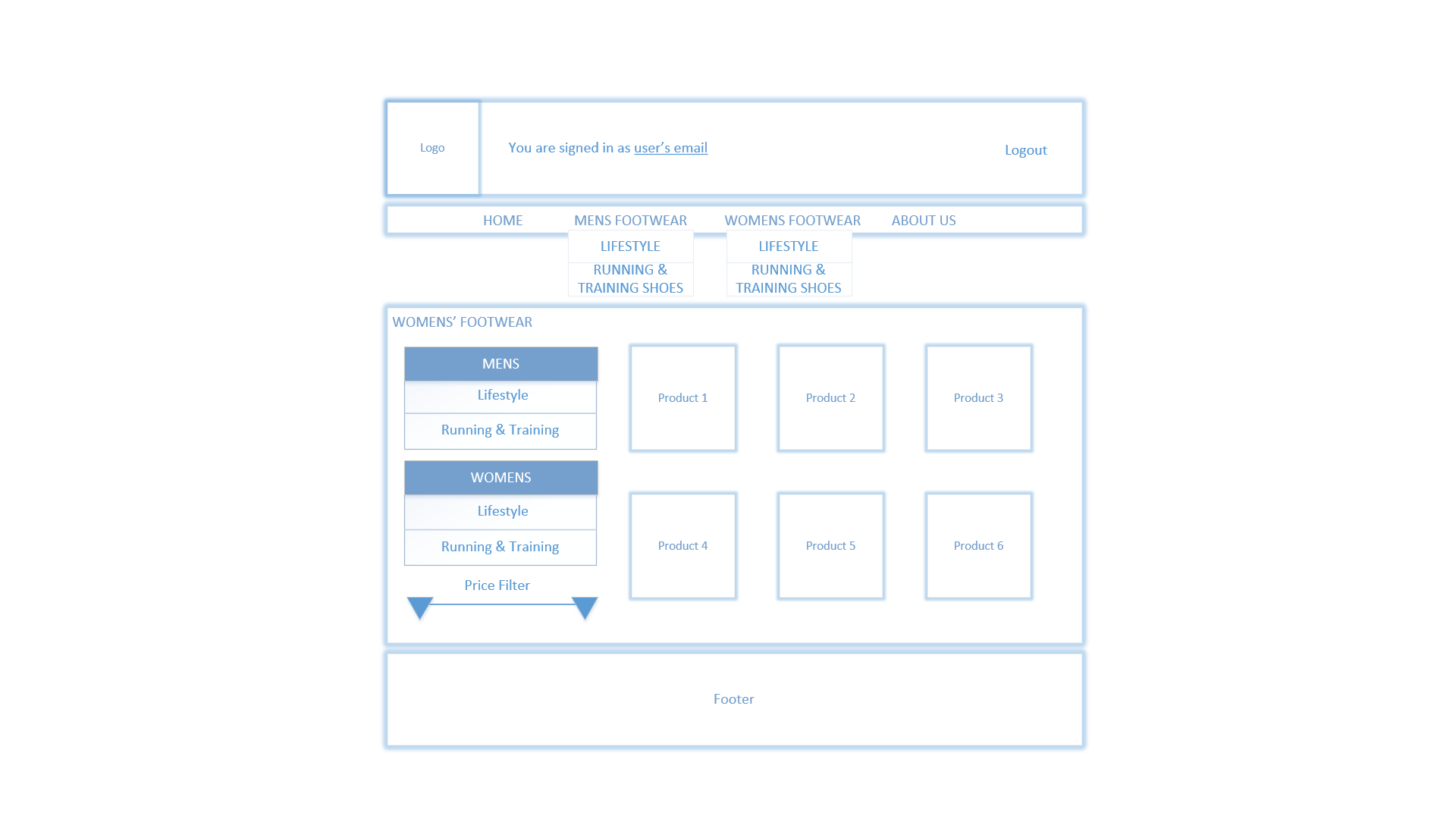


Customer Sign Up



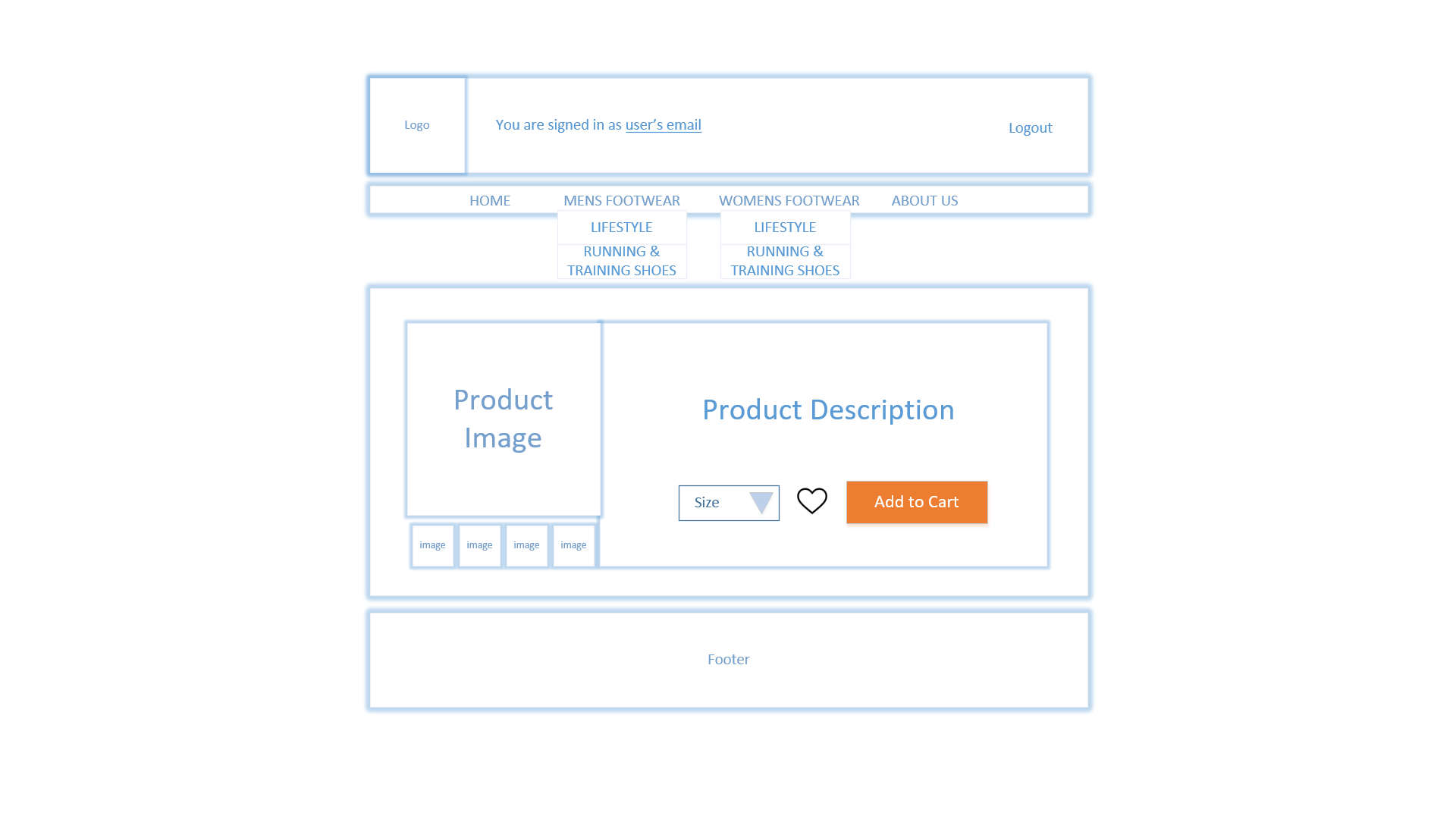
Product Catalogue

A picture containing vector graphics

Description generated with high confidence

Product Description

A picture containing vector graphics

Description generated with high confidence

Customer Wishlist

A picture containing vector graphics

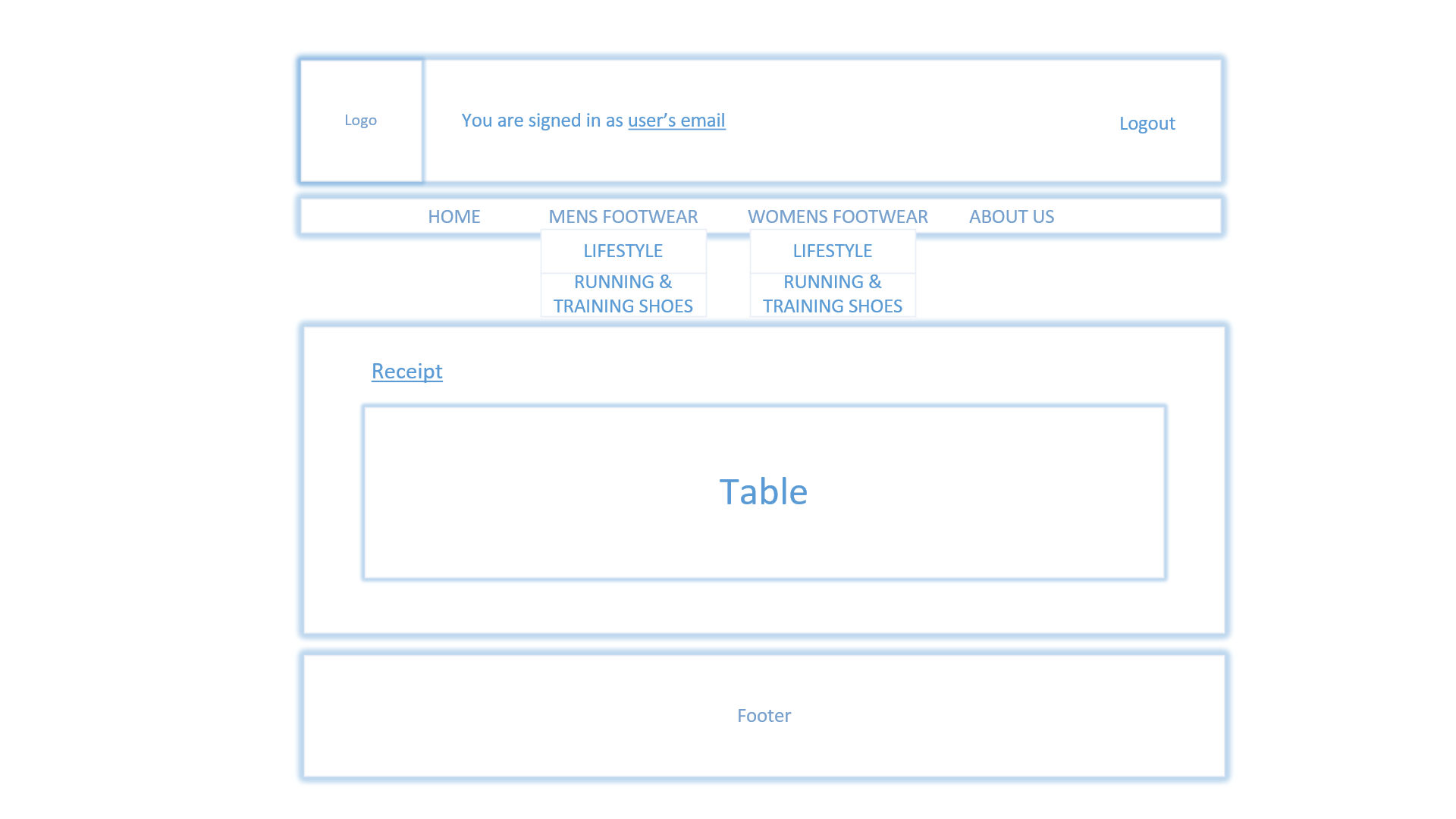
Description generated with high confidence

Customer Product Cart

A picture containing vector graphics

Description generated with high confidence

Customer Receipt

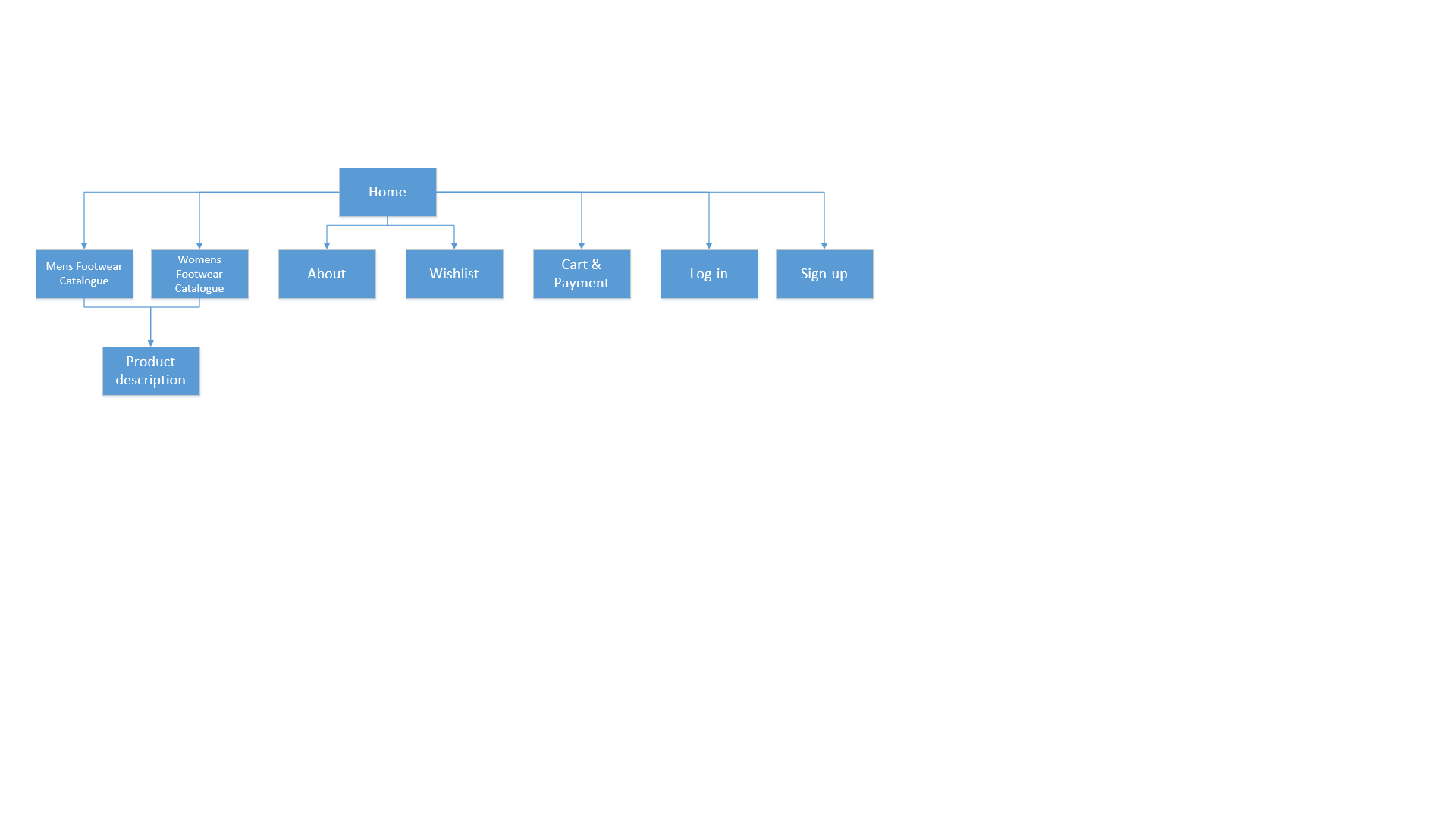




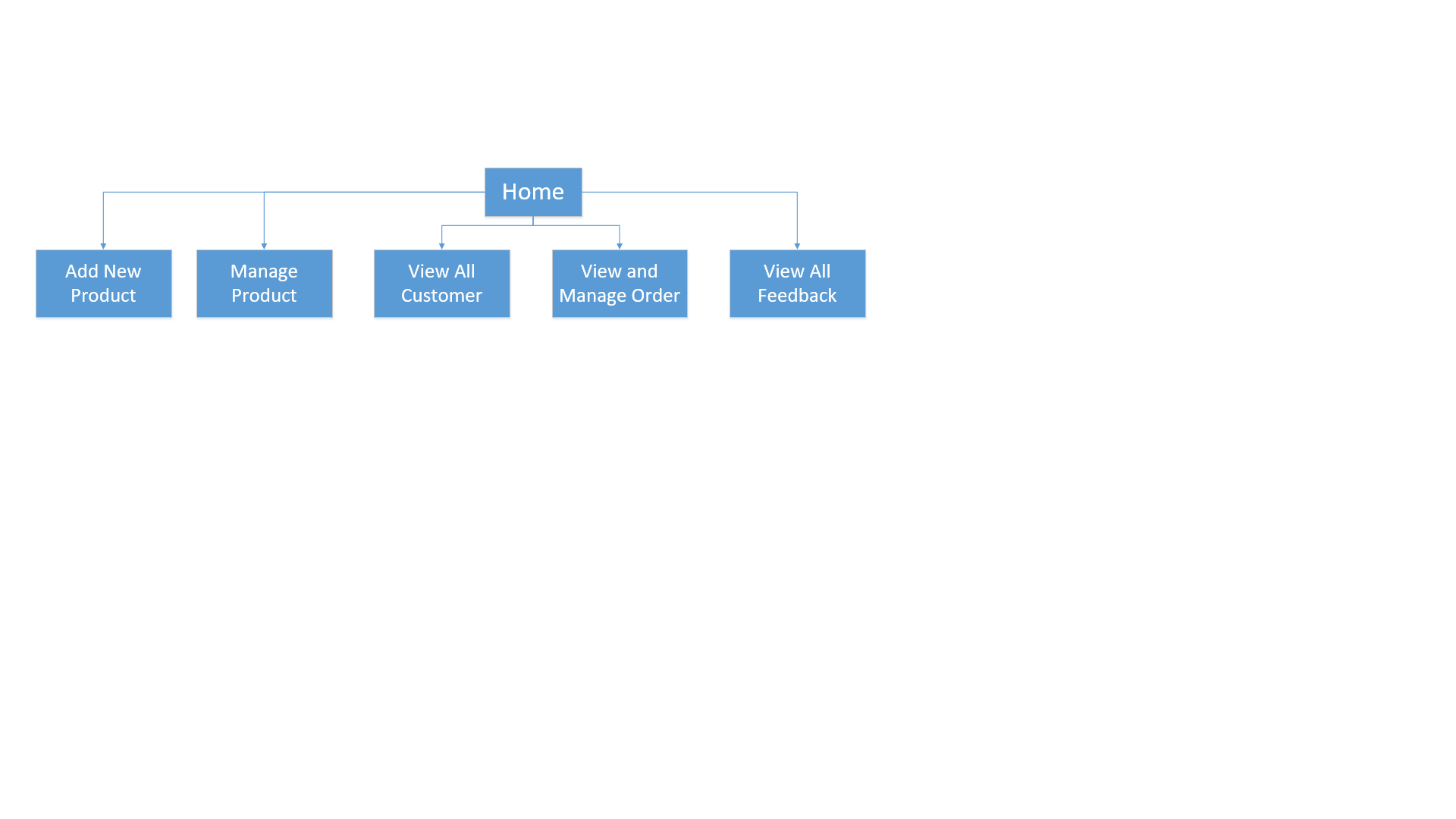
## 3.3 Website Architecture

### 3.3.1 Navigational Structure

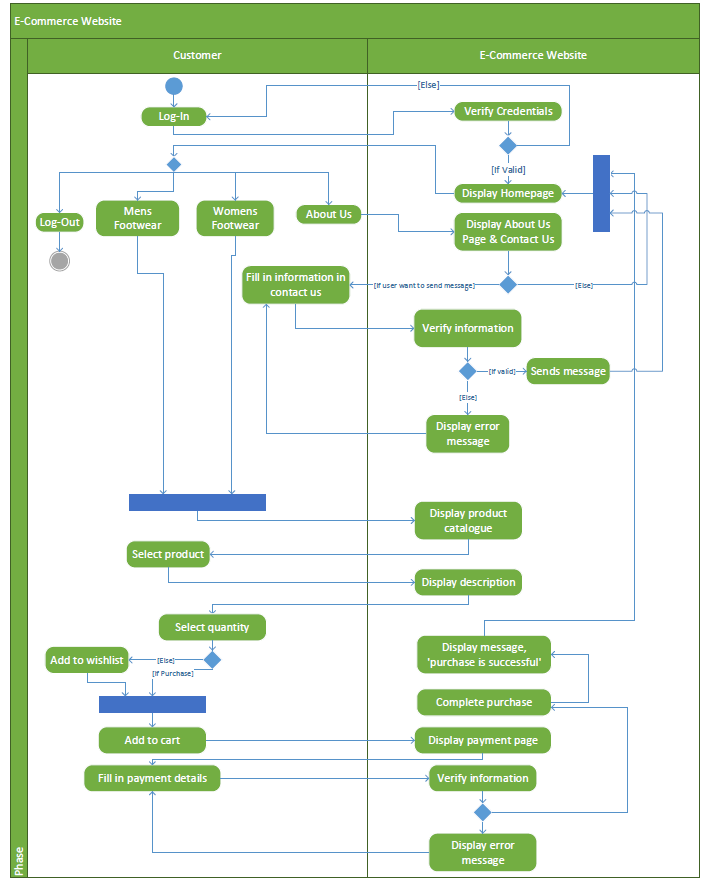
Customer

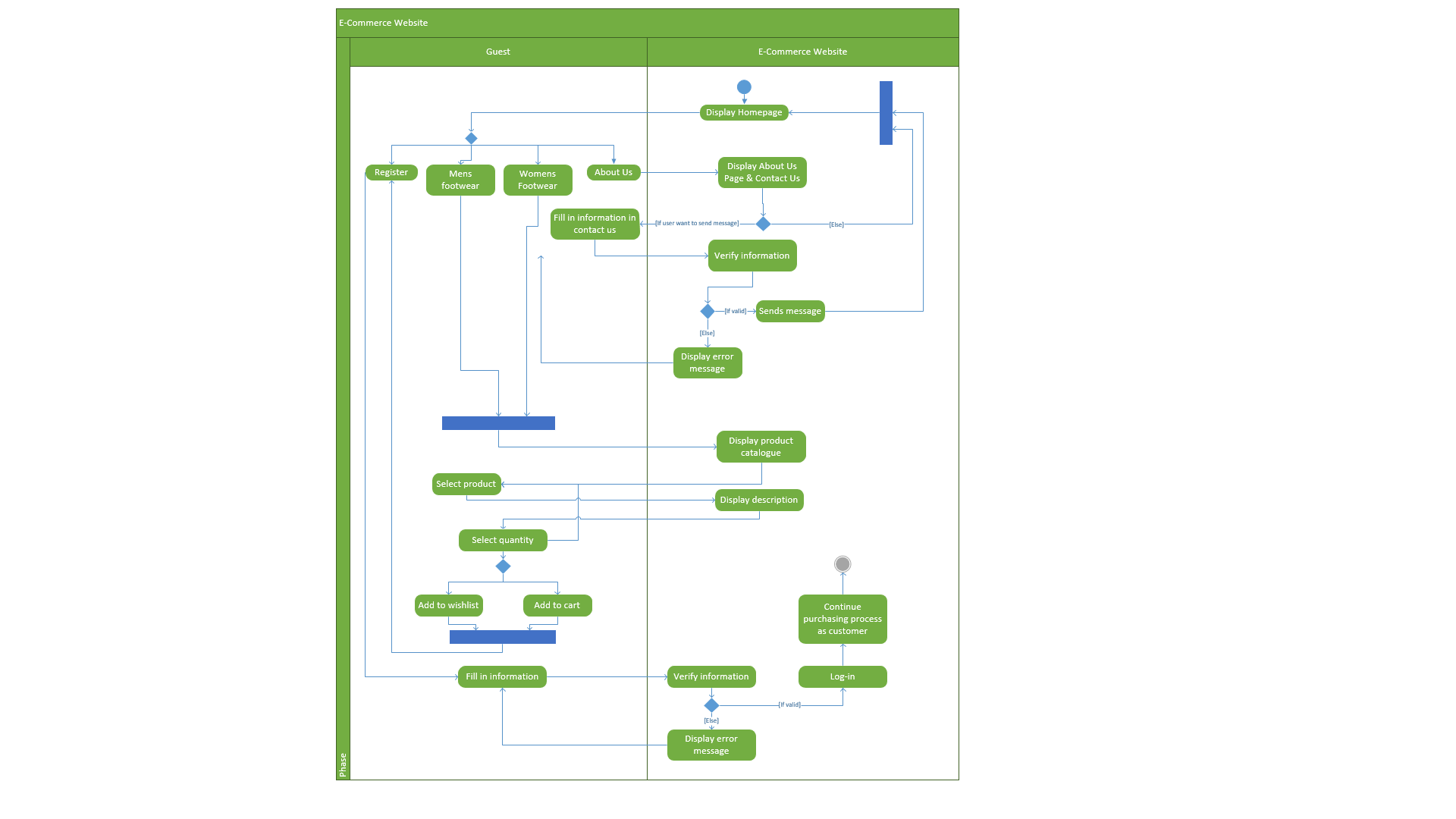


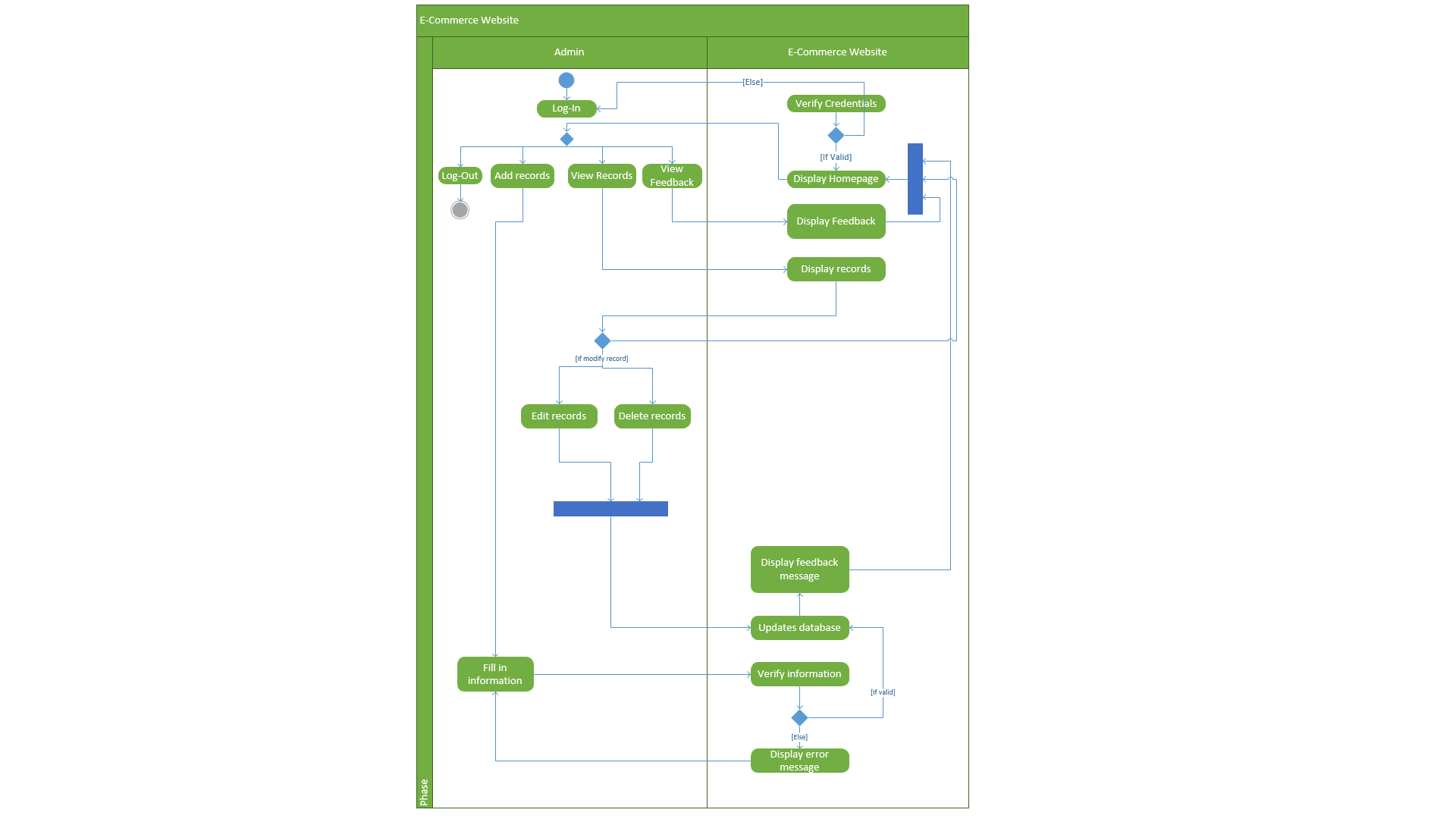
Administrator



### 3.3.2 Activity Diagram







### 3.3.3 Concurrent Task Tree

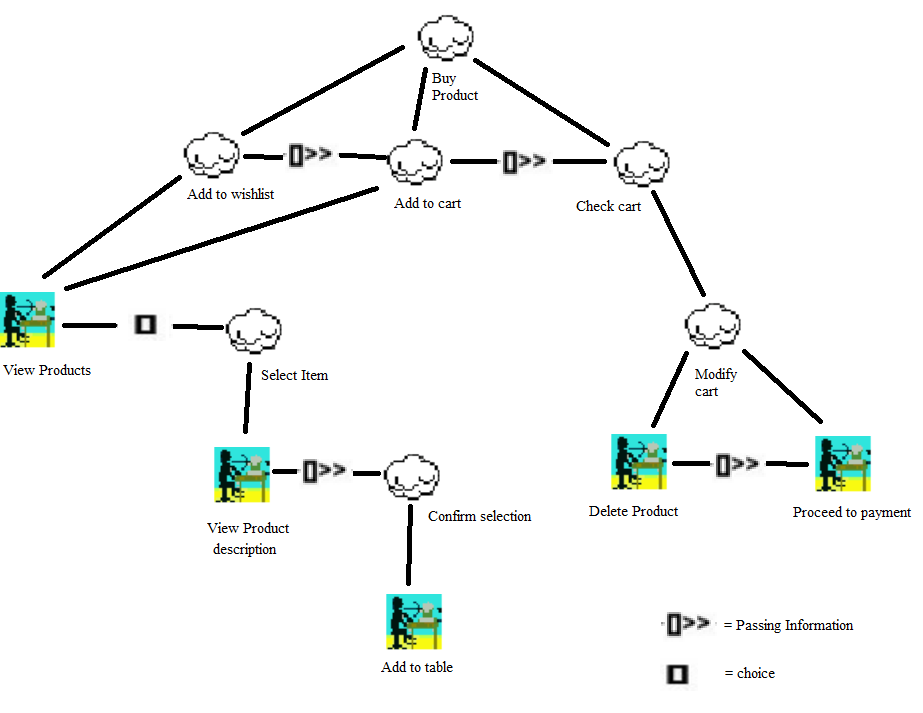


Figure 3.3.3.1: CTT for customer purchase

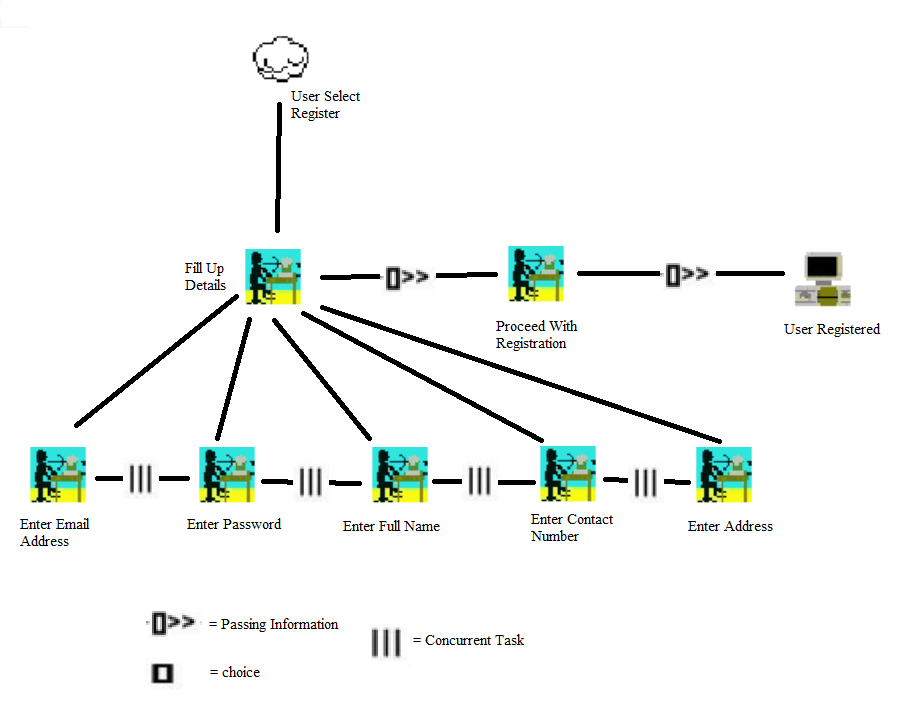


Figure 3.3.3.2: CTT for user registration

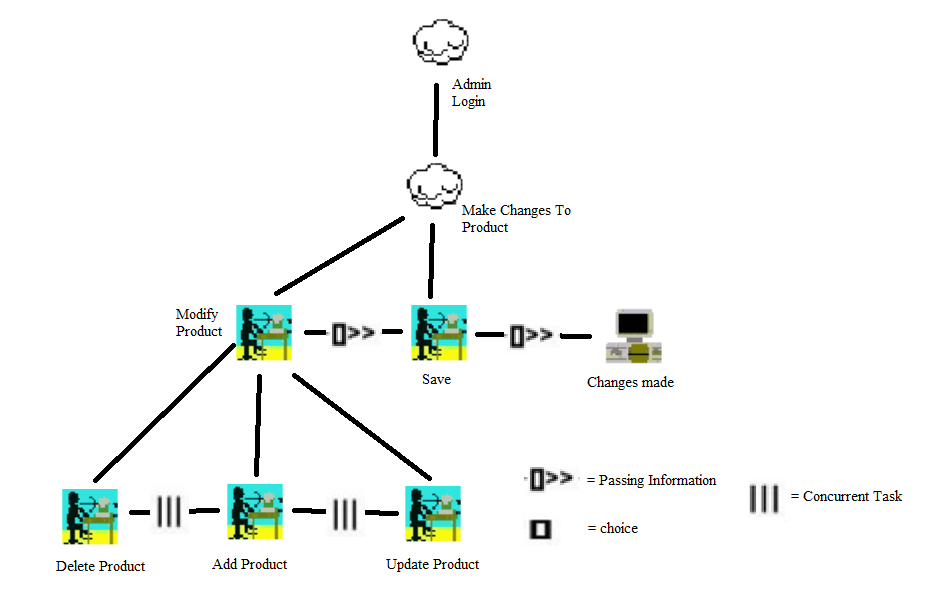


Figure 3.3.3.3: CTT for Admin Product Management

# 4.0 Implementation

## 4.1 Login & MasterPage

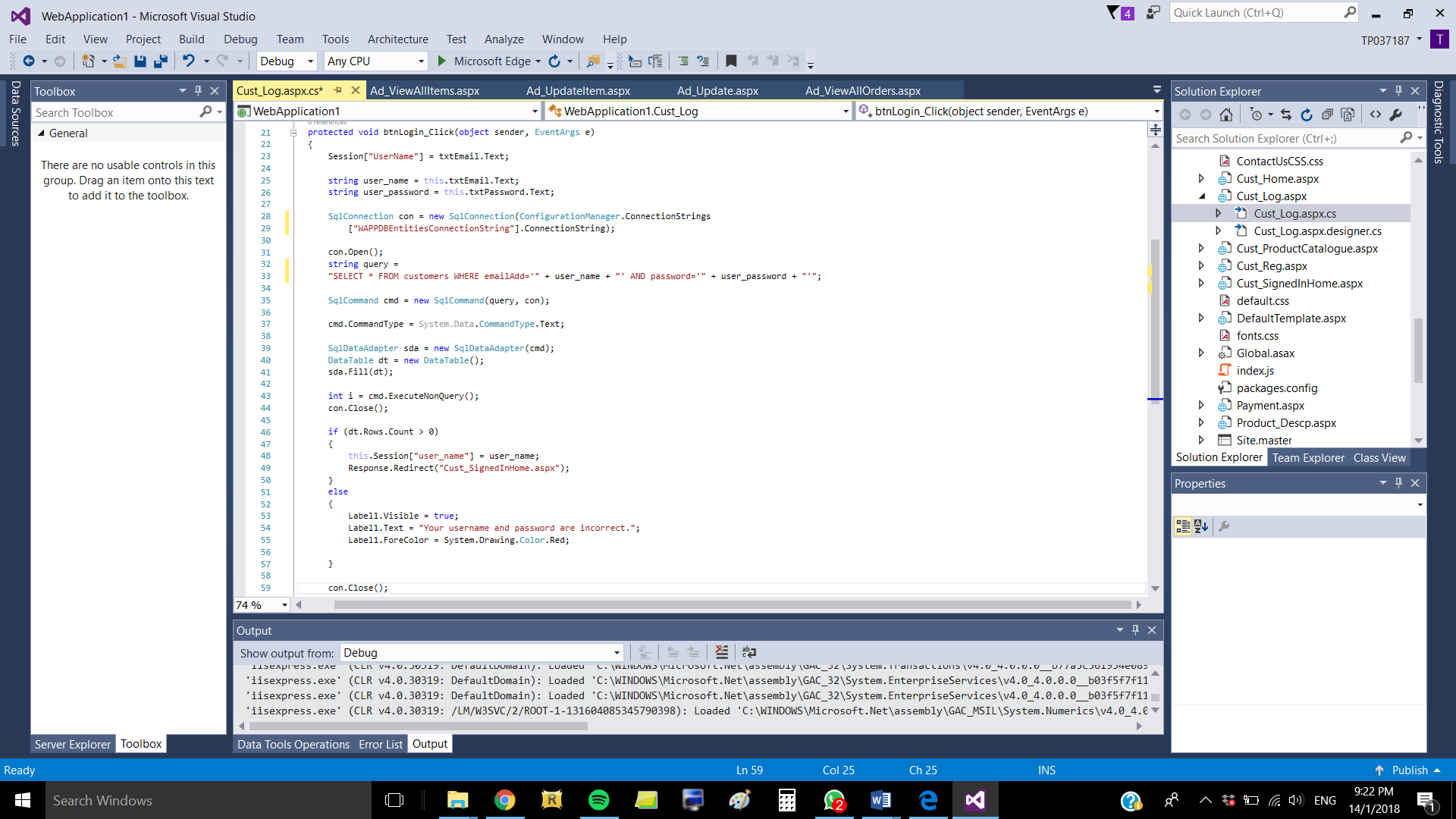


Figure 4.1.1: Function for Login button in Customer Login form

The figure above shows the code implementation for Customer Login form. The user has to enter their registered email address and password in txtEmail and txtPassword textboxes accordingly then click on the ‘Login’ button, in which the system will validate if the credentials match the ones in customers table, as seen in the query string. If they match, then the system will save customer’s email in session state and pass that information to the next page which is the Homepage for customers who have successfully logged in, hence in that homepage header there is a label that indicates that the user is signed in as [user’s email]. If the credentials do not match, the login is unsuccessful, and a message will appear to inform users that their email and password are incorrect.

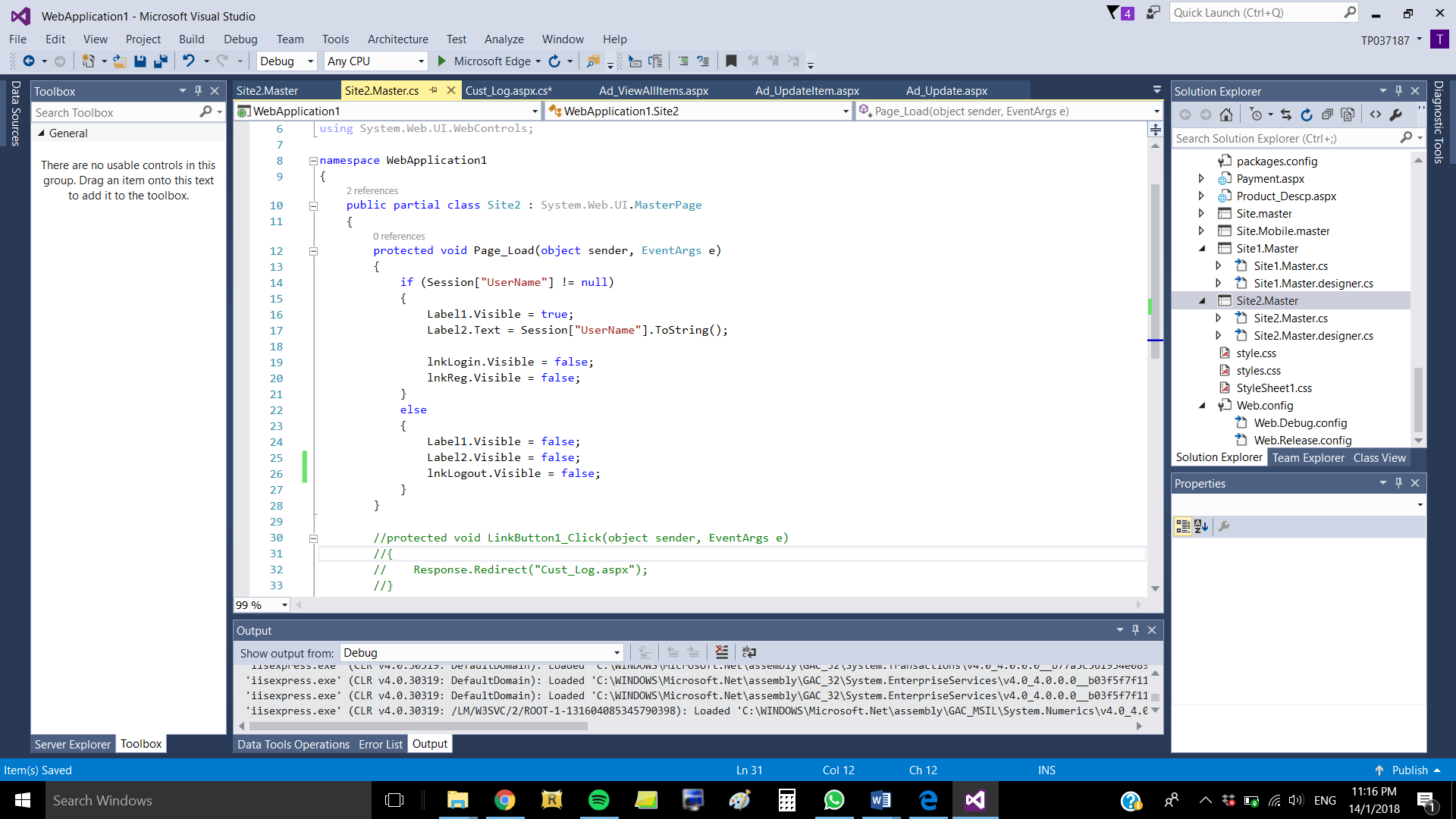


Figure 4.1.2: Function for header in Customer Masterpage

In the Customer Masterpage that contains the header and footer for all customer and visitor webpages, there is an if-else statement on page load which is used to determine which header to show user based on whether they are logged in or not. If user is logged in, that means the session state contains user’s email (as retrieved from Login form), hence the header should display the label that says “You are signed in as [user’s email]” and hide the Login and Register linkbuttons. Meanwhile, if the session state is empty, then the Login and Register linkbuttons should be displayed whilst the user identification label and Logout linkbutton should be hidden.

## 4.2 Admin Add ProductA screenshot of a cell phone Description generated with high confidence

Figure : Product category, product type and product name

A screenshot of a social media post

Description generated with very high confidence

Figure : Product description, product size and product price

A screenshot of a social media post

Description generated with very high confidenceA screenshot of a social media post

Description generated with very high confidenceFor add product, it will require the admin to select the product category, type, name, description, size, price and 4 images to be displayed in product catalogue and description. Therefore, it uses required field validation by putting **required = “required”** in every field to ensure there are no empty fields when it is entered in the database. Furthermore, input type validation is used in the price field to ensure only numbers are entered by putting **input type = “number”,** which means if the admin tries to enter text, there will be no result.

Figure : Upload product images part 2

Figure : Upload product images

A screenshot of a social media post

Description generated with very high confidence

Figure : Get image path and binary

A screenshot of a cell phone

Description generated with high confidenceA screenshot of a cell phone

Description generated with very high confidence

Figure : Saving data to database

Based on the above, a method called CreateProduct() is created which gets the image path and binary when an image is uploaded in the FileUpload field. Once it reads the file uploaded, it will determine the type of file using switch case, if it is in jpg/png/gif then it will convert the file into binary format, if it is in other formats then an error message will show. The system will assign the value from the variables in the fields into the database. In the button, it will call the ItemModel class and method CreateProduct() and AddProduct() from the class to insert data into the database.

## 4.3 Product Catalogue

A screenshot of a social media post

Description generated with very high confidence

Figure 4.3.1: Dropdown menu

Based on the dropdown menu, it uses the IF statement. If men footwear is chosen, it will display men’s footwear products and if the user further chooses the type, it will show products based on the category and type of product. Based on the IF statement, it calls different methods from the ItemModel class.

A screenshot of a cell phone

Description generated with very high confidence For the display of products from the database, it uses foreach statement to repeatedly display products if it can read and retrieve data from database. Once it reads, it assigns the field from the database and appends it in the respective elements.

Figure 4.3.2: Display product from database

A screenshot of a social media post

Description generated with very high confidence

Figure 4.3.3: if-else statement for categorization of products in the menu bar

Based on the code, it uses IF statement which gets the selected product from the product catalogue. Once product is selected, it will assign the variables and database field to the respective HTML elements by calling the method GetProduct() from the ItemModel class such as image to display the four-image path of the selected products from the database.

A screenshot of a cell phone

Description generated with high confidenceA screenshot of a social media post

Description generated with very high confidence

Figure 4.3.5: javascript for price filter

Figure 4.3.4: javascript for drop down menu

To show products from database, the dropdown menu is created for users to pick whether they want to view women or men’s footwear based on the type of products offered; lifestyle or running & training shoes. For the animation of the drop-down menu, JavaScript is used where the dropped down menu will be cloned and append to the menu and display when user clicks the HTML element. For the price filter, it also uses Javasript sets the range between the value of 190 to 728 as the minimum and maximum values respectively and uses the filter method to specify the criteria, and returns data based on user’s filter/specified criteria.

## A screenshot of a social media post Description generated with very high confidenceA screenshot of a social media post Description generated with very high confidence4.4 Product Description

Figure 4.4.1: Getting product selected to be displayed in product description page

Based on the code, it uses IF statement which gets the selected product from the product catalogue. Once product is selected, it will assign the variables and database field to the respective HTML elements by calling the method GetProduct() from the ItemModel class such as image to display the four-image path of the selected products from the database.

## 4.5 Cart



Figure 4.5.1: Cart table html

The figure above shows the code implementation to create a table to display the values or data depending on the needs. In this case, the table is created for cart page to display the items a customer purchased. The design of the table is taken from bootstrap. The columns created are such as Product, Category, Type, Size, Price and remove. At the top of the figure, the codes are used to create a confirm box function which will be using during delete process.

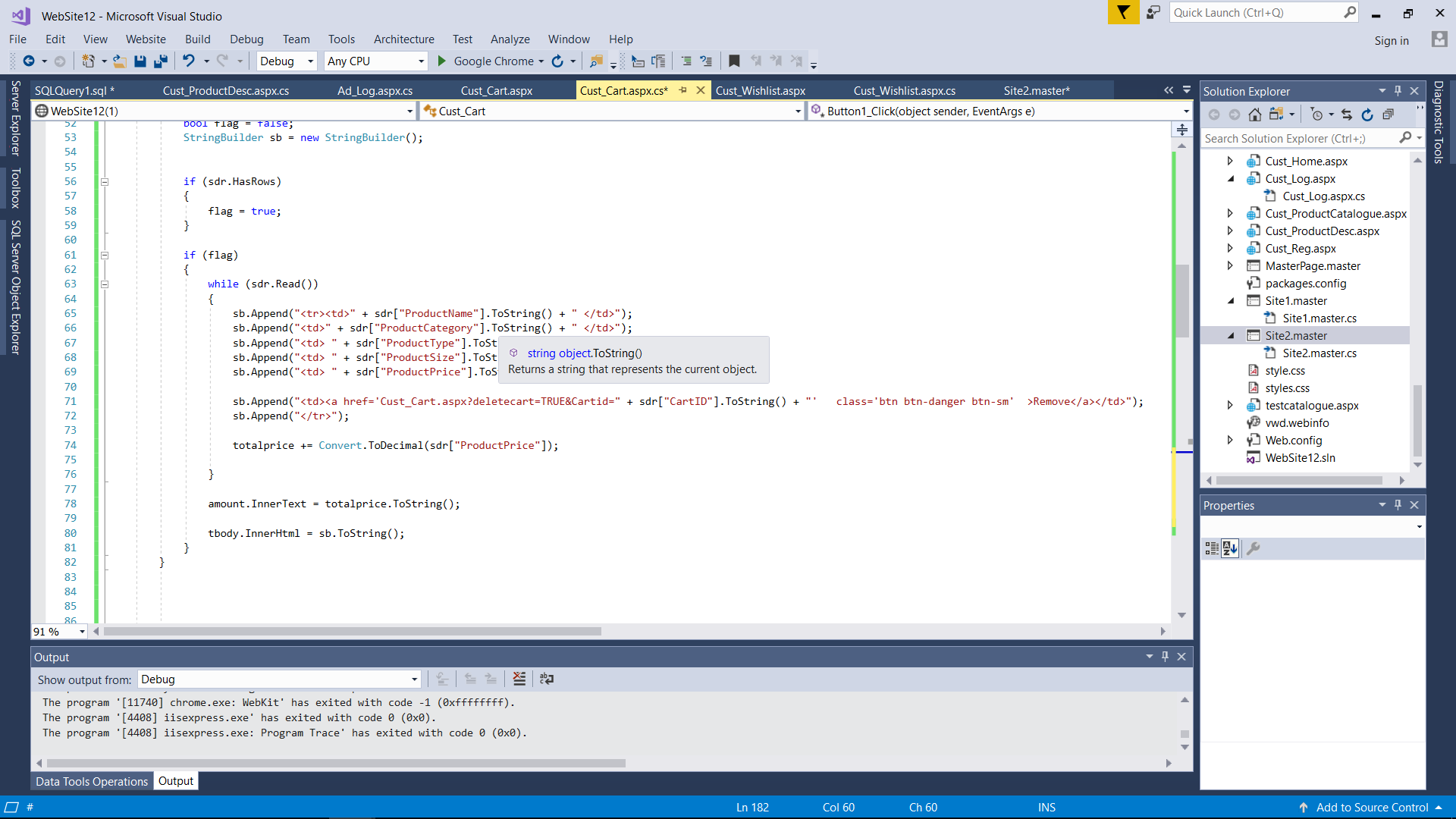


Figure 4.5.2: Calculate sum from cart table

This shows the code to calculate the sum of prices displayed in cart table and displayed by the label. The sum is calculated using looping function to sum up all the prices displayed in the table. The figure also shows how the data is received from the cart table and displayed. The customer id in the cart must be the same to the id customer who logged in.

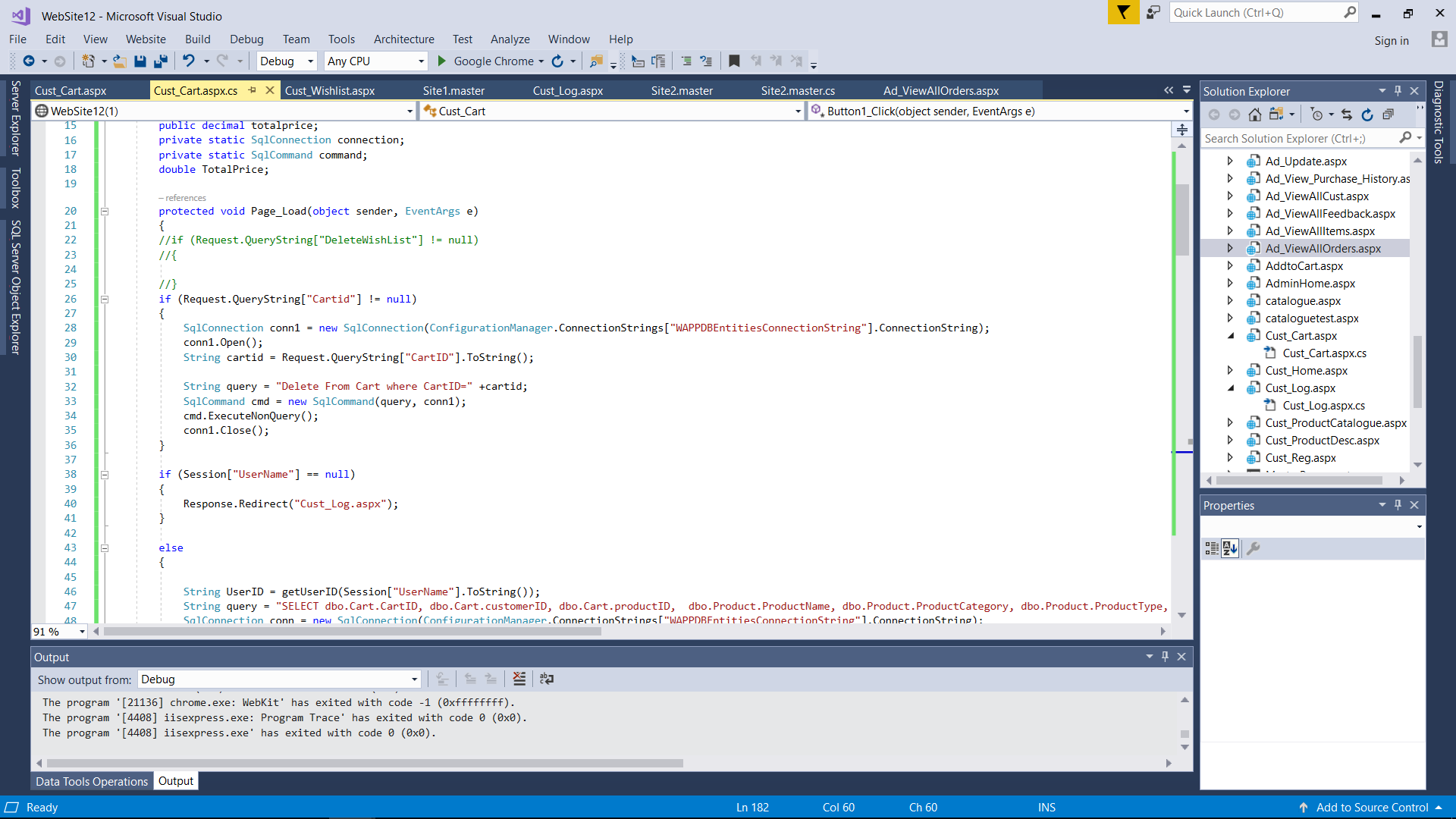
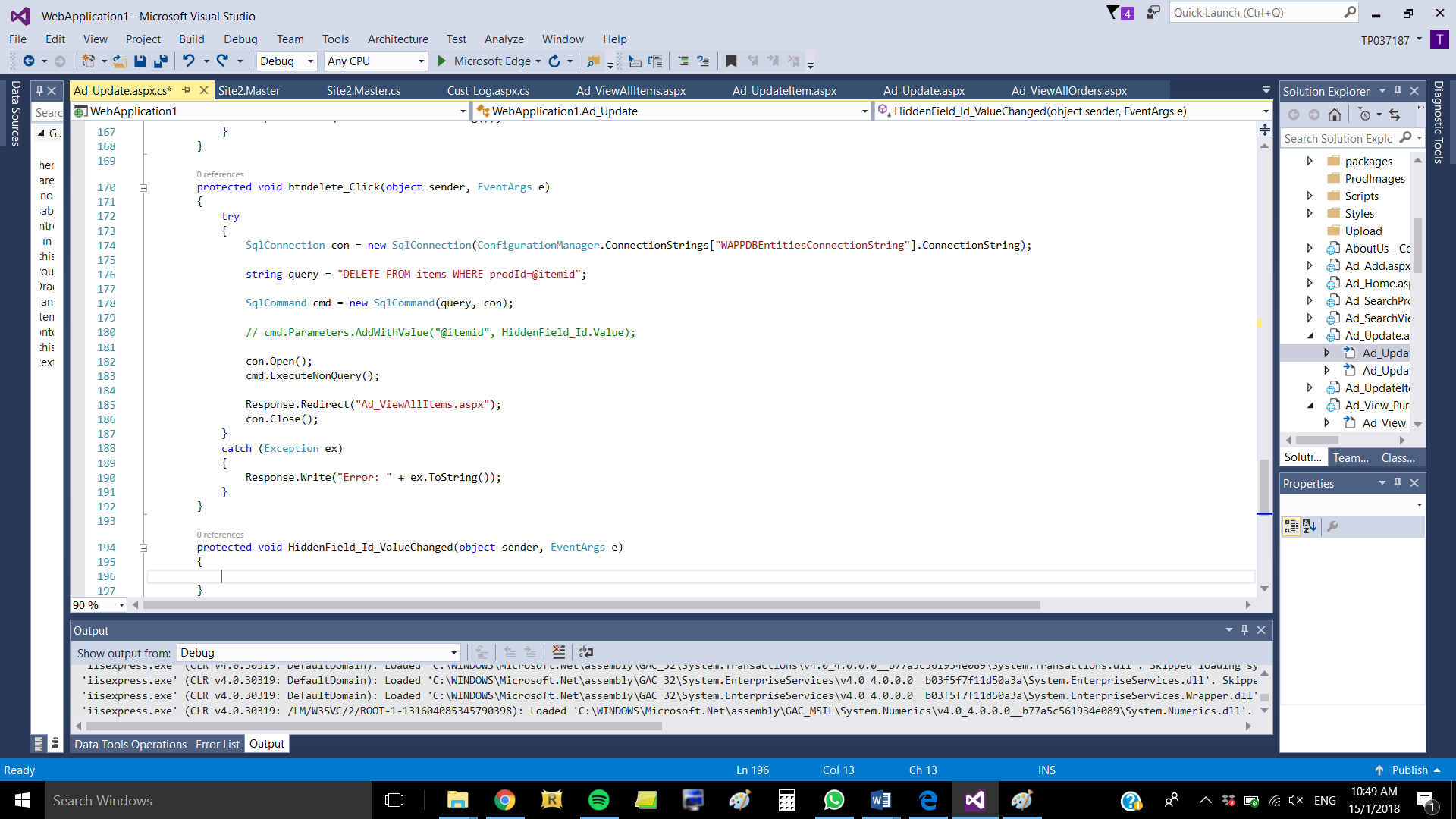


Figure 4.5.3: Delete product from cart table

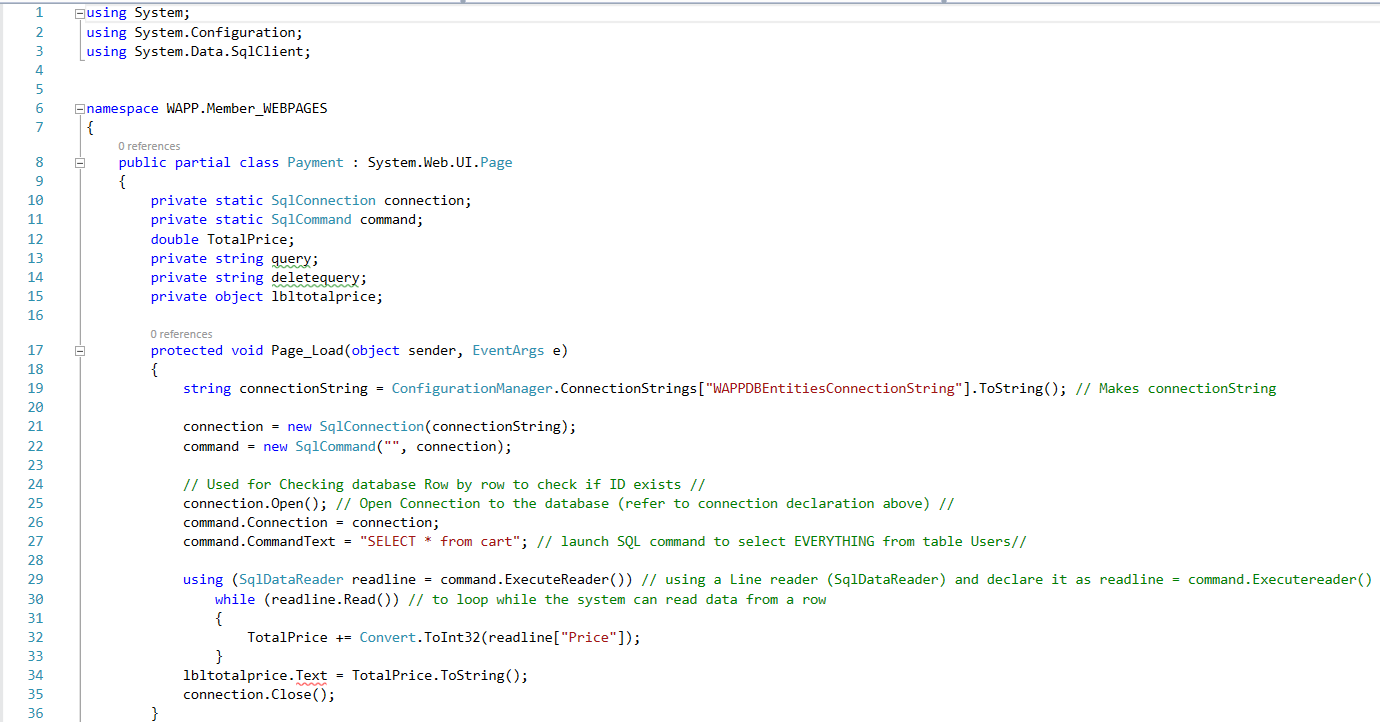
Figure xx shows the delete function of the cart table where customer can remove a product from their cart list if they changed their mind. The data will then be deleted both from the table and the database table. Before the delete function is carried out, a message box will appear to ask whether the user is certain with their decision. The code will delete based on the cartID which is the primary key of the Cart table in the database.

## 4.6 Delete



The above figure shows the code implementation for the Delete function in the Admin Manage Items form. When the form loads, it will retrieve the fields belonging to a certain product ID stored in the session state. Then user can click on ‘Delete’ button to delete the record. The system will validate if the product ID exists in the items table. If the product name is available, the selected product will be removed from the database. The string query command which is DELETE FROM items means that the selected ID will be deleted from the database.

## 4.7Payment





To identify the total price of the cart, the readline command is used so that the system will read all the data in “Price” field row by row in order to sum up the price. The user is required to key in the field in the Payment Page. The TotalPrice += TotalPrice.ToString(); means that the price be sum up in the Payment Page. When the user clicks on the ‘Submit Button’, the page will proceed to the Home Page. When the user clicks on the ‘Submit’ button, there will be a pop up message which shows option ‘Yes’ or ‘No’, if the user clicks on ‘Yes’ button, the data will be saved in the transaction table in database using the code ‘SqlCommand insertcommand’.

# 5.0 Conclusion

Overall, we developed an e-commerce website for Sportify which sells specifically footwear as one of their well-known products. We incorporated functionalities which allows customer to filter, browse and purchase the product by adding features such as adding to cart and performing payment activities. Aside from the customer, we also incorporated functionalities for the admin which is another type of user who manage the products, user and transactions. Based on that, we added features such as adding and updating products in the database and viewing feedback sent by customers.

However, there are many rooms for improvements such as developing a delivery tracking system which allows customers to track their delivered products which they have purchased online, to increase customer’s satisfaction due to the information transparency. Furthermore, we could incorporate more attributes for customers to select from when purchasing their product. Such as size, quantity and colour to increase the range of choices and flexibility for users.

Moreover, while this website allows registered users to view and add products into the shopping cart, it still lacks a secure platform for them to do online payment. Thus, future works may include connecting the website to PayPal or other online payment gateways. Search function can be added on customer side to search for products and admin side to search for product and customer. Furthermore, another improvement for the website will be allow Admin to add new staff into the website the website currently only has one Admin.

# **6.0 References**

Evarevirus, 2017. *Shop.* [Online]   
Available at: https://bootsnipp.com/snippets/5MzD5  
[Accessed 16 January 2018].

Khan, M., 2009. *Display images from SQL Server Database in ASP.Net GridView control.* [Online]   
Available at: https://www.aspsnippets.com/Articles/Display-images-from-SQL-Server-Database-in-ASP.Net-GridView-control.aspx  
[Accessed 20 December 2017].

Kingsley, A., n.d. *30 Free Responsive jQuery Image Sliders and Slideshows.* [Online]   
Available at: https://dcrazed.com/free-responsive-jquery-image-sliders/  
[Accessed 5 November 2017].

Msurguy, 2017. *Simple registration form.* [Online]   
Available at: https://bootsnipp.com/snippets/featured/simple-registration-form  
[Accessed 16 January 2018].

Silkalns, A., 2018. *50 Free HTML5 and CSS3 Login Forms For Your Website 2018.* [Online]   
Available at: https://colorlib.com/wp/html5-and-css3-login-forms/  
[Accessed 11 January 2018].

Silkalns, A., 2018. *Top 16 Free HTML5 & CSS3 Contact Form Templates 2018.* [Online]   
Available at: https://colorlib.com/wp/free-html5-contact-form-templates/  
[Accessed 7 January 2018].

Stackoverflow, 2018. *Allow only numbers to be typed in a textbox [duplicate].* [Online]   
Available at: https://stackoverflow.com/questions/7295843/allow-only-numbers-to-be-typed-in-a-textbox  
[Accessed 16 January 2018].

Stackoverflow, 2018. *How to Display Selected Item in Bootstrap Button Dropdown Title.* [Online]   
Available at: https://stackoverflow.com/questions/13437446/how-to-display-selected-item-in-bootstrap-button-dropdown-title  
[Accessed 16 January 2018].

Uthmansy, 2017. *Ecommerce Product Detail.* [Online]   
Available at: https://bootsnipp.com/snippets/featured/ecommerce-product-detail  
[Accessed 16 January 2018].

W3schools, 2018. *Bootstrap Dropdowns.* [Online]   
Available at: https://www.w3schools.com/bootstrap/bootstrap\_dropdowns.asp  
[Accessed 16 January 2017].