Table of Contents

[1.0 Introduction 3](#_Toc525299724)

[1.1 Project Background 3](#_Toc525299725)

[1.2 Objectives 4](#_Toc525299726)

[1.3 Scope 4](#_Toc525299727)

[1.4 Targeted Audience 5](#_Toc525299728)

[1.5 Project Schedule 6](#_Toc525299729)

[1.6 Major Functions 8](#_Toc525299730)

[2.0 Requirement Specifications 8](#_Toc525299731)

[2.1 Features 8](#_Toc525299732)

[2.2 System Functions 9](#_Toc525299733)

[2.3 Software and Hardware Requirements 11](#_Toc525299734)

[2.3.1 Software Requirements 11](#_Toc525299735)

[2.3.2 Hardware Requirements 12](#_Toc525299736)

[3.0 Design 13](#_Toc525299737)

[3.1 Use Case Diagram 13](#_Toc525299738)

[3.2 Activity Diagram 15](#_Toc525299739)

[3.2.1 Customer 15](#_Toc525299740)

[3.2.2 Administrator 17](#_Toc525299741)

[3.2.3 Manager 18](#_Toc525299742)

[3.3 Entity Relationship Diagram 19](#_Toc525299743)

[3.4 Navigation Structure 21](#_Toc525299744)

[3.4.1 Customer 21](#_Toc525299745)

[3.4.2 Administrator 21](#_Toc525299746)

[3.4.3 Manager 22](#_Toc525299747)

[3.5 Presentation Design 23](#_Toc525299748)

[3.5.1 Visitor/Customer Homepage 23](#_Toc525299749)

[3.5.2 About Us 24](#_Toc525299750)

[3.5.3 Product Catalogue 25](#_Toc525299751)

[3.5.4 Product Details 26](#_Toc525299752)

[3.5.5 Checkout 27](#_Toc525299753)

[3.5.6 Order Success 28](#_Toc525299754)

[3.5.7 Check Order 29](#_Toc525299755)

[3.5.8 Submit Feedback 30](#_Toc525299756)

[3.5.9 Register 31](#_Toc525299757)

[3.5.10 Login 32](#_Toc525299758)

[3.5.11 Admin View & Manage Products 33](#_Toc525299759)

[3.5.12 Admin Add Product 34](#_Toc525299760)

[3.5.13 Admin Update Product 35](#_Toc525299761)

[3.5.14 Admin View & Approve Orders 36](#_Toc525299762)

[3.5.15 Manager View Customers 37](#_Toc525299763)

[3.5.16 Manager View Orders from a Selected Customer 38](#_Toc525299764)

[3.5.17 Manager View All Orders 39](#_Toc525299765)

[3.5.18 Manager View Orders for a Selected Product 40](#_Toc525299766)

[3.5.19 Manager View Feedback 41](#_Toc525299767)

[3.6 XML Tree 42](#_Toc525299768)

[3.6.1 Users 42](#_Toc525299769)

[3.6.2 Payment 43](#_Toc525299770)

[3.6.3 Feedback 44](#_Toc525299771)

[3.6.4 Country 45](#_Toc525299772)

[3.6.5 Category 45](#_Toc525299773)

[3.6.6 Ad Rotator 46](#_Toc525299774)

[3.6.7 Homepage – Top Products 46](#_Toc525299775)

[3.6.8 About Us - Team 47](#_Toc525299776)

[4.0 Implementation 48](#_Toc525299777)

[4.1 Register & Login (with XML) 48](#_Toc525299778)

[4.2 Checkout (with XML) 53](#_Toc525299779)

[4.3 Feedback (with XML) 57](#_Toc525299780)

[4.4 View & Delete Product 61](#_Toc525299781)

[4.5 Update Product (with XML) 63](#_Toc525299782)

[4.6 Test plan (test cases, pass/fail) 66](#_Toc525299783)

[4.7 Evaluation 68](#_Toc525299784)

[4.7.1 Evaluation by Customer 68](#_Toc525299785)

[4.7.2 Evaluation by Admin 69](#_Toc525299786)

[4.7.3 Evaluation by Manager 70](#_Toc525299787)

[5.0 Conclusion/Critical Appraisal 73](#_Toc525299788)

# 1.0 Introduction

## 1.1 Project Background

SoftwareBiz is a software retailer that caters to small-medium business needs in the areas of accounting, marketing, and procurement. Founded in 2009, they have become one of the leading software provider companies in Malaysia who sells in-house developed software solutions for SME businesses.

Vision

* To be recognized as one of the preferred software solution providers within the SME (Small-Medium Enterprise) sector across Asia.

Mission

* To empower customers and enhance their business growth through the utilisation of the latest technologies that streamlines their operations, monitors and improves performance.

SoftwareBiz focuses on selling technologies that are used for daily operations of Small-Medium Enterprises which are made to automate works and improve process efficiency. Currently, the way they do business is by selling ready-made off-the-shelf products in the form of CDs which customers can install on their computer. Each software unit can only run on one computer by one user and requires a product key that comes in the package, hence customers must purchase multiple quantity if multiple employees will be using it. Their customers are mainly companies consisting of 30-250 employees. They market their products through franchise in different Asian countries like Malaysia, Indonesia, China, India, and Singapore. They have representatives overseas who introduce their products to companies in that country.

SoftwareBiz continuously performs market research to understand the current situation and needs of small-medium businesses. The collected data is then analysed to form requirements for the next product development. During product development, the company consults with several external subject matter experts such as CEO, marketing and procurement managers of an SME in order to identify gaps and specify requirements. Then, a team of in-house developers will develop and test the software before launching into the market. SoftwareBiz always tries to get a good deal for their customers without compromising quality.

SoftwareBiz feels the need to change their business process by adapting to current digital trends which their competitors have been adopting. They want to bring their business activities online and sell their products in downloadable form instead of CD installation. This will greatly reduce their cost and increase sales. Therefore, the aim of this project is to develop a fully functional B2B e-commerce web application for SoftwareBiz to conduct their online business activities such as selling products through online catalogue, generating and managing (a.k.a reading and approving) orders, delivering software in downloadable form, and communicating with customers. It is essentially an online store that sells downloadable software products meant to be used by SME companies.

## 1.2 Objectives

There are several objectives for this project, such as:

* To conduct online business activities between SoftwareBiz (retailer) and its company-sized consumers
* To provide ease and convenience for retailer and customers in seeing the latest data about products and orders
* To reduce the time taken in generating and managing orders and products
* To provide reporting capabilities for managers regarding customers and sales data
* To integrate XML in certain aspects in order to prepare for communication with another system especially in payment process

## 1.3 Scope

The purpose of developing a B2B web application is to assist and streamline business processes between retailer and consumers, providing ease, convenience, and efficiency which ultimately leads to cost reduction, higher sales, and customer loyalty. The scope of the system are as follows:

Visitor

* To register an account
* To browse and view products
* To send a message to SoftwareBiz

Customer

* To login and logout
* To browse and view products
* To make a purchase
* To view purchase history
* To send a message to SoftwareBiz

Administrator

* To login and logout
* To add and manage products
* To view and approve orders

Manager

* To login and logout
* To view customer details and their orders
* To view all orders
* To view orders for a specific product
* To view feedback/messages received

In summary, the system will incorporate interactive features such as allowing Customer to contact customer support through feedback form, access information of the products offered, and performing a purchase with online payment. It should allow Admin to modify data related to products and orders. It will also provide reporting capabilities for the Manager regarding customers and sales in order to assess the company’s performance and build customer relationships in the future.

## 1.4 Targeted Audience

There are 4 user types in this system, as follows.

Visitor

The website visitors are those who express interest in the SoftwareBiz and the product offerings by visiting the website. Visitors do not have a user account or logged in yet, hence their information would not be recorded into the system. They could be anyone, such as SoftwareBiz’s competitors, market or product researchers, and company representatives.

Customer

The customers will mainly be company representatives who aim to purchase software solutions that can be used for day-to-day business operations on behalf of the company. They are most likely to be from the procurement department, IT department, or from the top-level management like COO and CTO. These companies would look for the best-priced and most suitable software according to their varied needs, hence they may be researching and considering different vendors (SoftwareBiz’s competitors). Usually, companies do not buy multiple types of software from the same retailer, they would only buy what they need at the moment and what the retailer’s speciality is, hence it would be fair to say they do not really need a cart feature. Since the web application can be accessed globally, customers may come from any country in the world. However, as stated in the company’s vision, SoftwareBiz will mainly target companies in Asian countries and focus their marketing efforts there, therefore most probably their customers are located in Asia.

Administrator

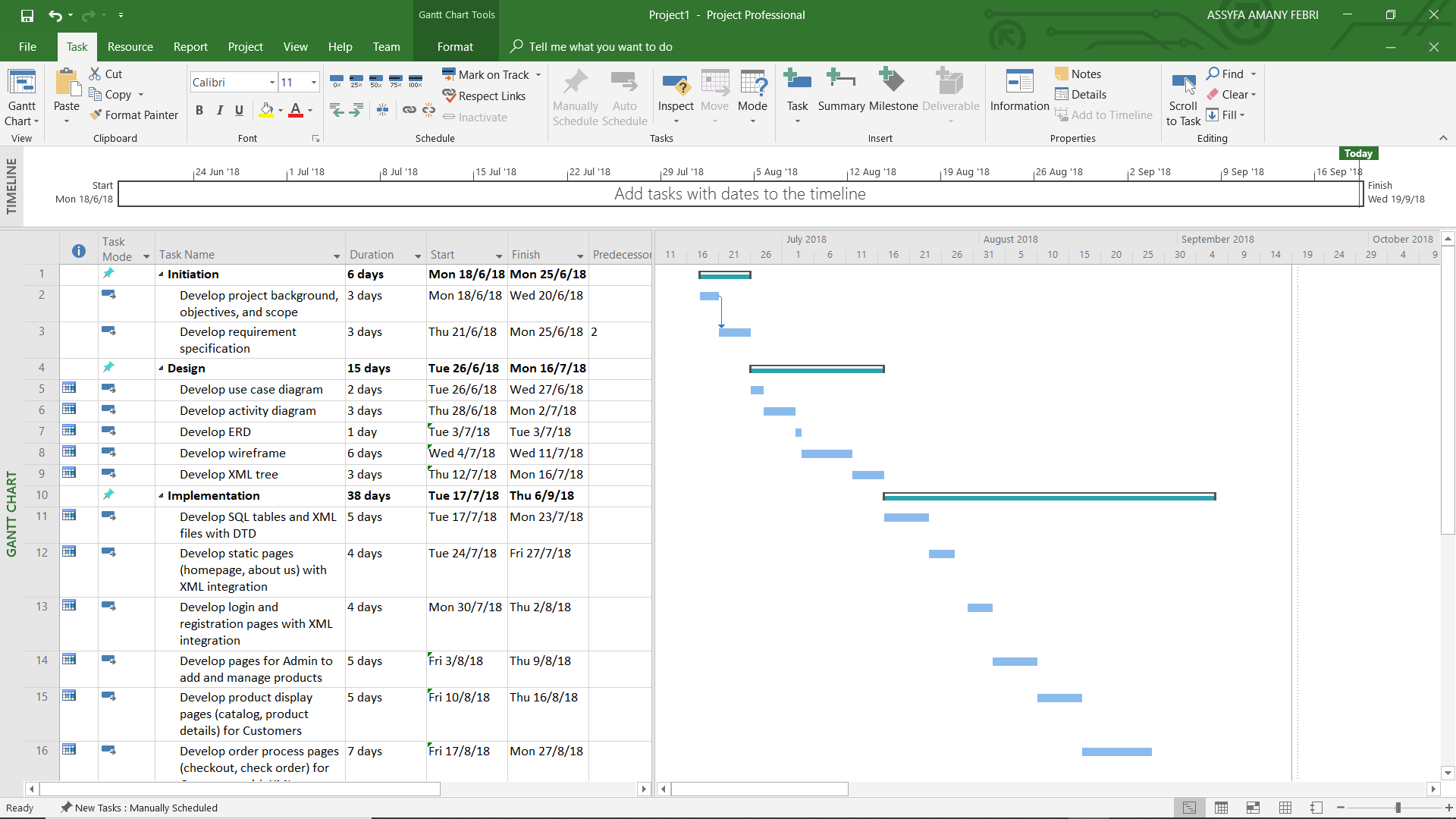
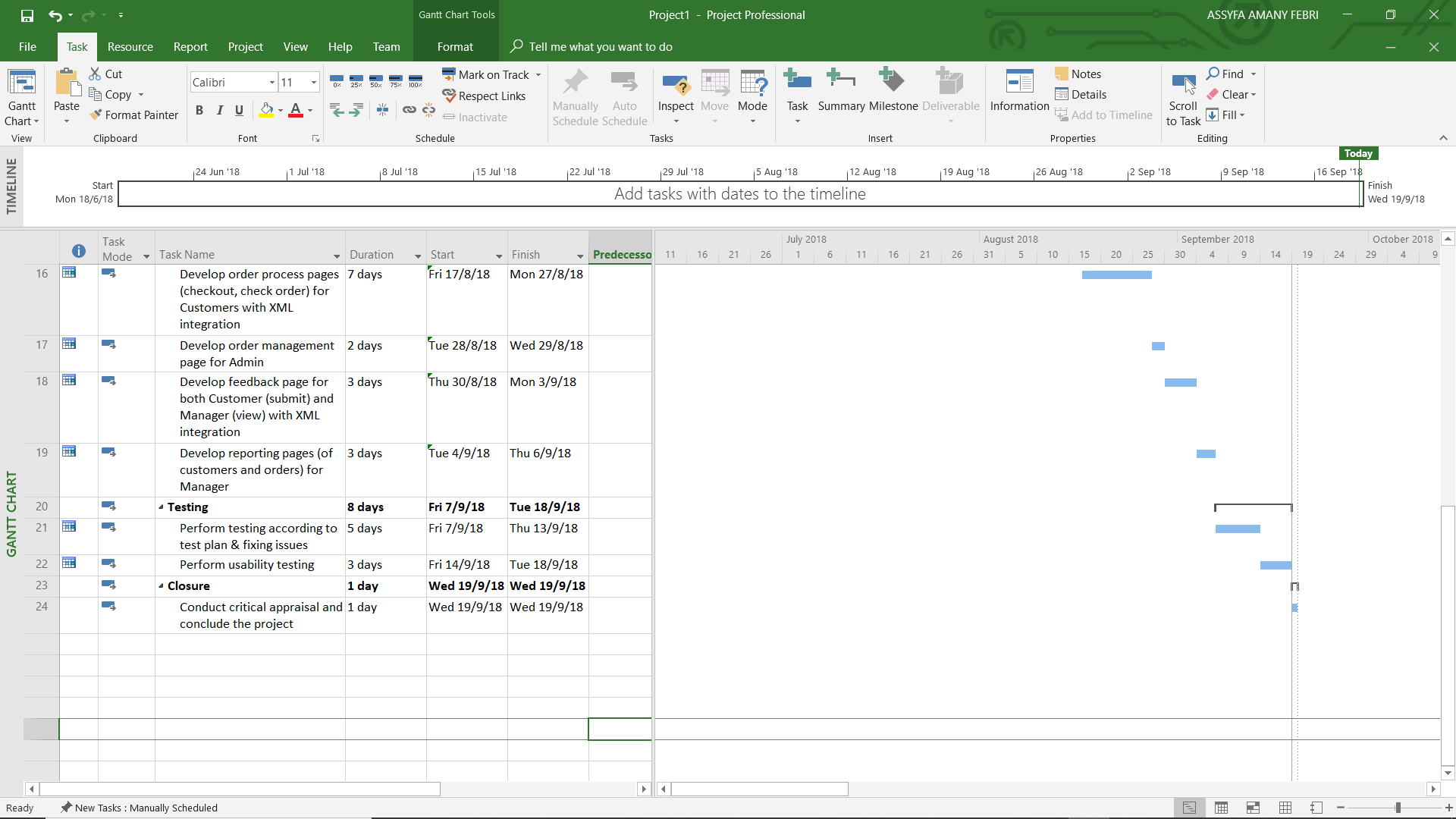
The website administrator is SoftwareBiz’s employee who is responsible in managing the website, providing content and updating it with the latest information. They are most likely to be a staff in the sales department who has access to the company products and in charge of managing orders.

Manager

The managers of SoftwareBiz will also be using the web application in order to assess company performance by accessing the data collected by the website so that they can form strategies and make better-informed decisions for the company. This user group consists of Head of Marketing, Head of Sales, Head of Customer Service, Head of Finance, and Head of Tech.

## 1.5 Project Schedule

The project schedule is as follows.



## 1.6 Major Functions

The major functions of the web application include product checkout with online payment, viewing purchase history and downloading products from approved orders, adding and managing products, approving orders, submitting feedback, and finally reporting capabilities on customer data and order data.

# 2.0 Requirement Specifications

## 2.1 Features

The four user types are Vistor, Customer, Manager, and Administrator. Visitors and Customers should be able to access the website and perform user registration and/or login. They can view the product catalogue and filter the items based on category: ‘accounting’, ‘marketing’, and ‘procurement’, in order to view products relevant to a specific business area.. When an item is clicked, they should be redirected to a page that displays further details of the chosen product. If they are interested in buying, they can enter the desired quantity on the product detail page itself and click checkout. If they have not logged in yet, they will be asked to login before continuing the purchase process. At checkout, Customer will see their order summary and be required to enter the necessary payment details. Once the order is confirmed and paid for, the transaction will be recorded in the system. Customer then has to wait for the order to be verified and approved by SoftwareBiz. They can come back later to view their purchase history and check the order status. If there are any approved orders, it will be shown in the “Approved Orders” list which comes with a link to download the software. After the product is downloaded, the order status will automatically change to “completed” and the whole order row will be moved to “Transaction History” list which has no download link.

Admin staff are the ones who are authorised to make changes to the content of the website, hence they will have the ability to access and modify products and orders. The products and their details will be shown in a grid for the Admin to get an overview of the existing products. They can also add new product by filling in all the required details, such as product name, description, category, price, and image. The system should allow Admin to edit product details or delete product and any changes will be updated in the database. Admin will be able to see a list of orders/transactions in detail, and have the power to update the order status from ‘processing’ to ‘approved’ to allow customers to download their purchased products.

Managers require reporting capabilities in order to make assessments and more informed decisions for SoftwareBiz. Hence, they should have access to customer information for the purpose of understanding their existing customers and build relationships, as well as access to orders/transactions in order to determine the sales trend. Manager will be able to view a list of registered customers and their details such as name, company name, role in the company, country, and email address. Furthermore, they can select a customer and view the orders made by that customer to understand what kind of products they were looking for and had chosen within a category. Additionally, they must be able to view a list of orders with the details to determine sales growth. They also require to be able to filter orders based on product, showing which companies in which countries purchased the selected product with the quantity.

The website structure and information layout will be designed in a way that ensures a good user experience that is easy to use by novice users. The homepage will be filled with an image slider that can be used to display news and advertisements. It will also have an overview of SoftwareBiz’s best-selling products to give customers a kickstart. For first-timers, they can visit the website’s About Us page to understand what SoftwareBiz does. Any visitor can contact the customer service through a feedback form where they can drop their details and message.

## 2.2 System Functions

Administrator

Description:

Administrators are employees of SoftwareBiz who are in charge of managing content of the website. They are usually from the sales department who has information regarding all the new and existing products. Their job is to add and manage products as well as processing orders.

Functionalities:

1. Login into the system as an Admin user
2. View an overview of listed products
3. Add new products
4. Edit product details
5. Delete products
6. View a list of orders
7. Update order status or approve an order

Manager

Description:

Managers of SoftwareBiz are those who hold a position in the middle or top management level and are responsible for making decisions within the company. They are usually the Head of Sales, Head of Finance, Head of Customer Service, CMO (Chief Marketing Officer), COO (Chief Operation Officer), and CEO (Chief Executive Officer).

Functionalities:

1. Login into the system as a Manager user
2. View a list of customers and their details
3. View a list of orders made by a selected customer
4. View a list of orders/transactions in detail
5. View a list of orders for a specified product

Visitor

Description:

Users who express interest on the company by visiting the website. They are most likely to be employees from an SME, looking to find candidates of suitable software to perform their daily operations.

Functionalities:

1. Register a Customer account into the system
2. View products
3. View company information
4. Send enquiry or feedback

Customer

Description:

Users who already have an account and attempt to purchase the company’s product(s) online. They would be a company’s representative (such as owner or manager) who has the authority to purchase software on behalf of the company.

Functionalities:

1. Login into the system
2. View products
3. Purchase different products separately, in multiple quantity
4. Check order status and transaction history
5. Download the software products that have been paid for and approved

## 2.3 Software and Hardware Requirements

### 2.3.1 Software Requirements

Operating System

Since the end-product is a web application, it can run on any operating software as long as there is a web browser. It can run on any Windows (preferably Windows 7 and above), Mac OS (preferably Mac OS 10.7 and above), or Linux on the client side. Since the system is made to be mobile-responsive, it can also run on any mobile operating system as well, whether it is Android or iOS, as long as there is a mobile web browser to access the system. However, the system development will be performed on a Windows operating system, due to its compatibility with the chosen programming language, ASP.NET C#, as they come from the same creator/provider, Microsoft.

Web Browser

The web application will look best in Google Chrome, Opera, and Mozilla Firefox because these browsers are compatible with JavaScript and HTML5. Running the system on Internet Explorer and Safari might have design and compatibility issues due to low JavaScript recursion limits by these browsers (Zakas, 2009).

Database

All the data will be stored in Microsoft SQL Server because it works natively with .NET due to belonging from the same provider which is Microsoft.

### 2.3.2 Hardware Requirements

Personal Computer

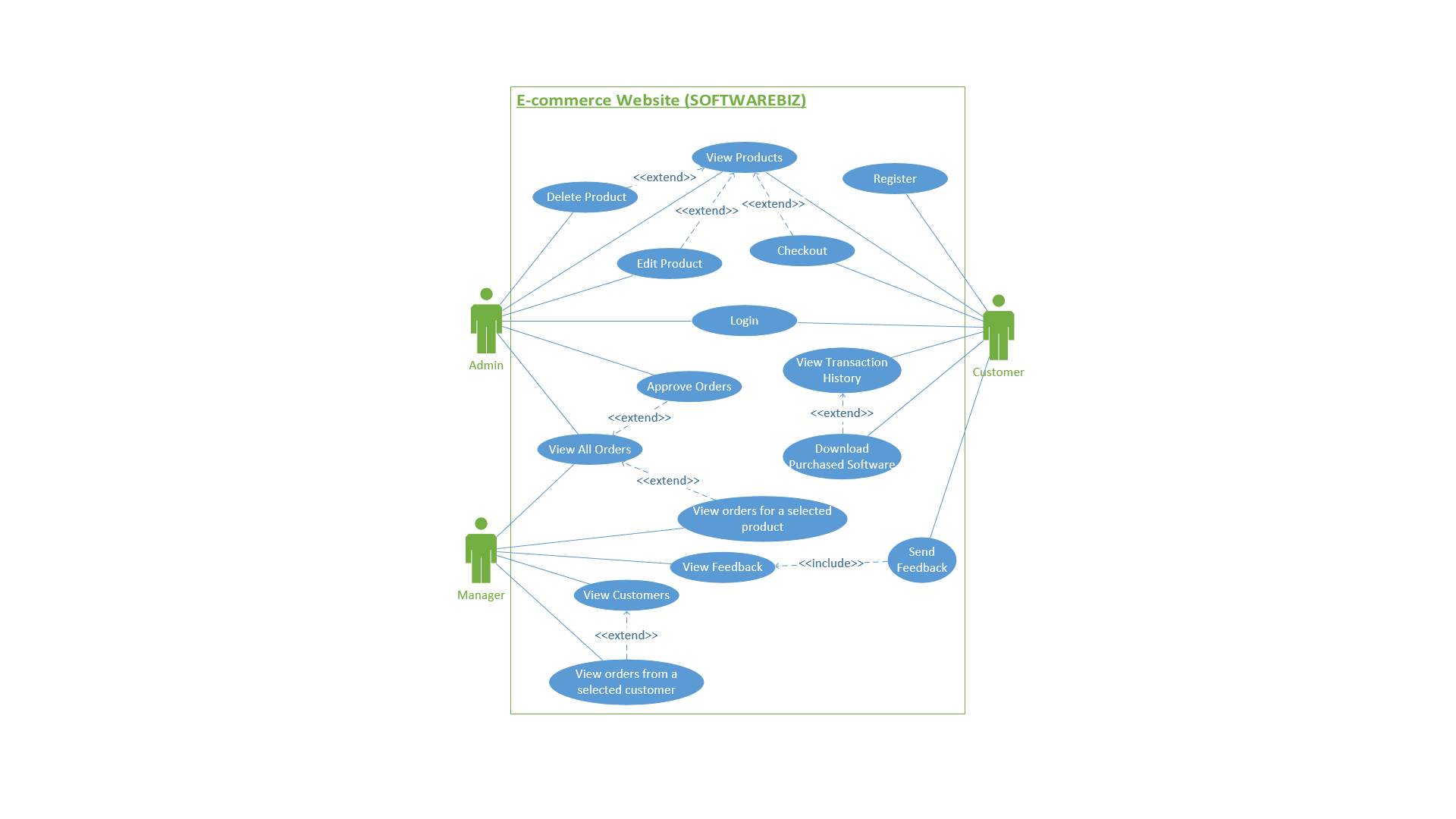
It requires a personal computer with an installed web browser to run the web application. The PC for client side should have a minimum of 512MB of RAM, 256MB of graphic card, and at least 10GB of free hard drive space. Meanwhile, for system development, in order to run ASP.NET it would require a minimum of Intel i3 Processor 2nd Gen or above, 4 GB of RAM, and 120GB of HDD (Deshmukh, 2016).

Server

During development stages, the web application will be hosted locally on IIS (Internet Information Services) created by Microsoft and made available on most version of Windows operating system. However, after handing over to SoftwareBiz, there needs to be a server that hosts the web application and keeps it online 24 hours for users to use. To ensure high availability, SofttwareBiz should invest in a capable server with good performance. Several options are Microsoft Azure and Amazon Web Services.

# 3.0 Design

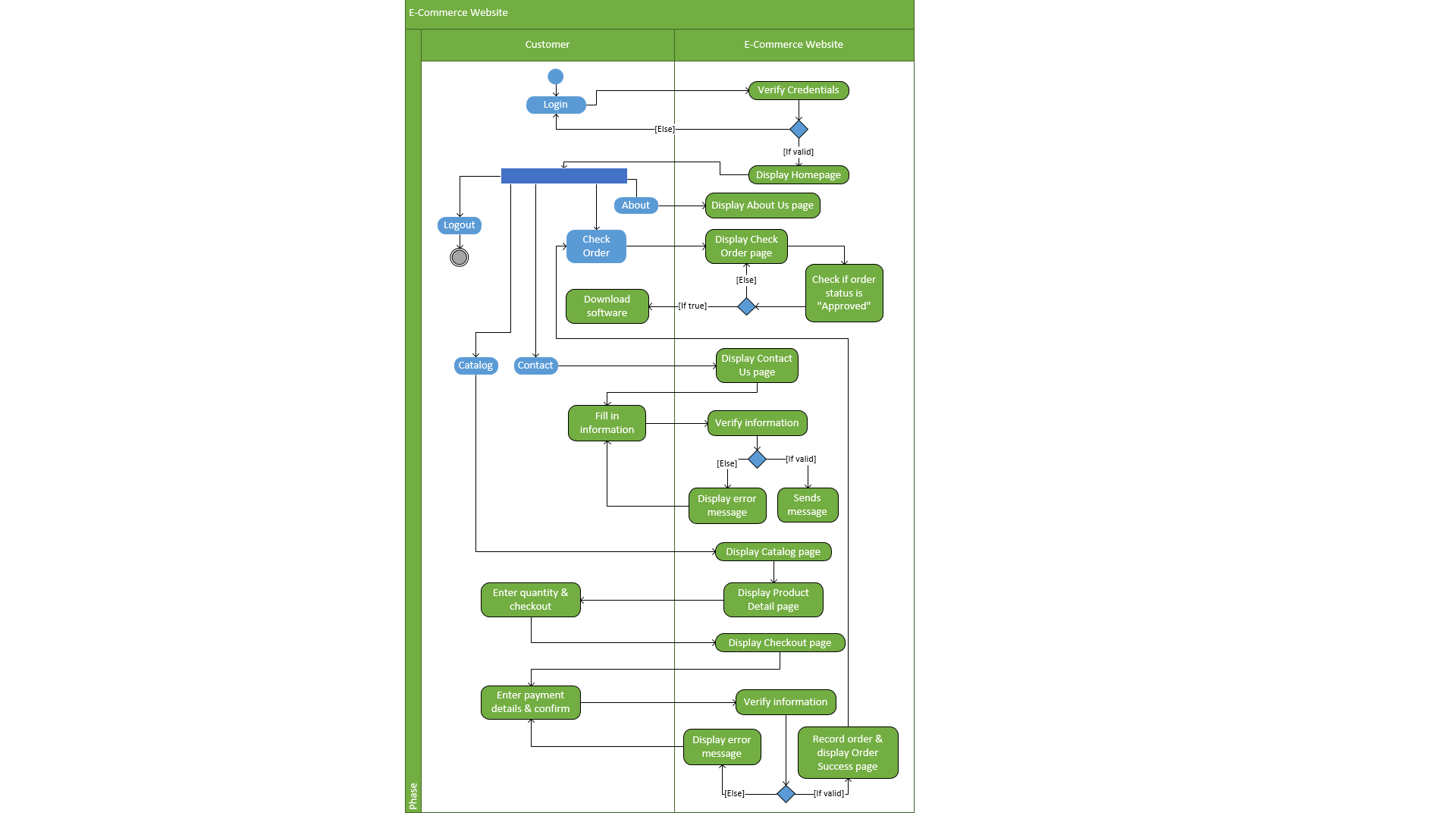
## 3.1 Use Case Diagram

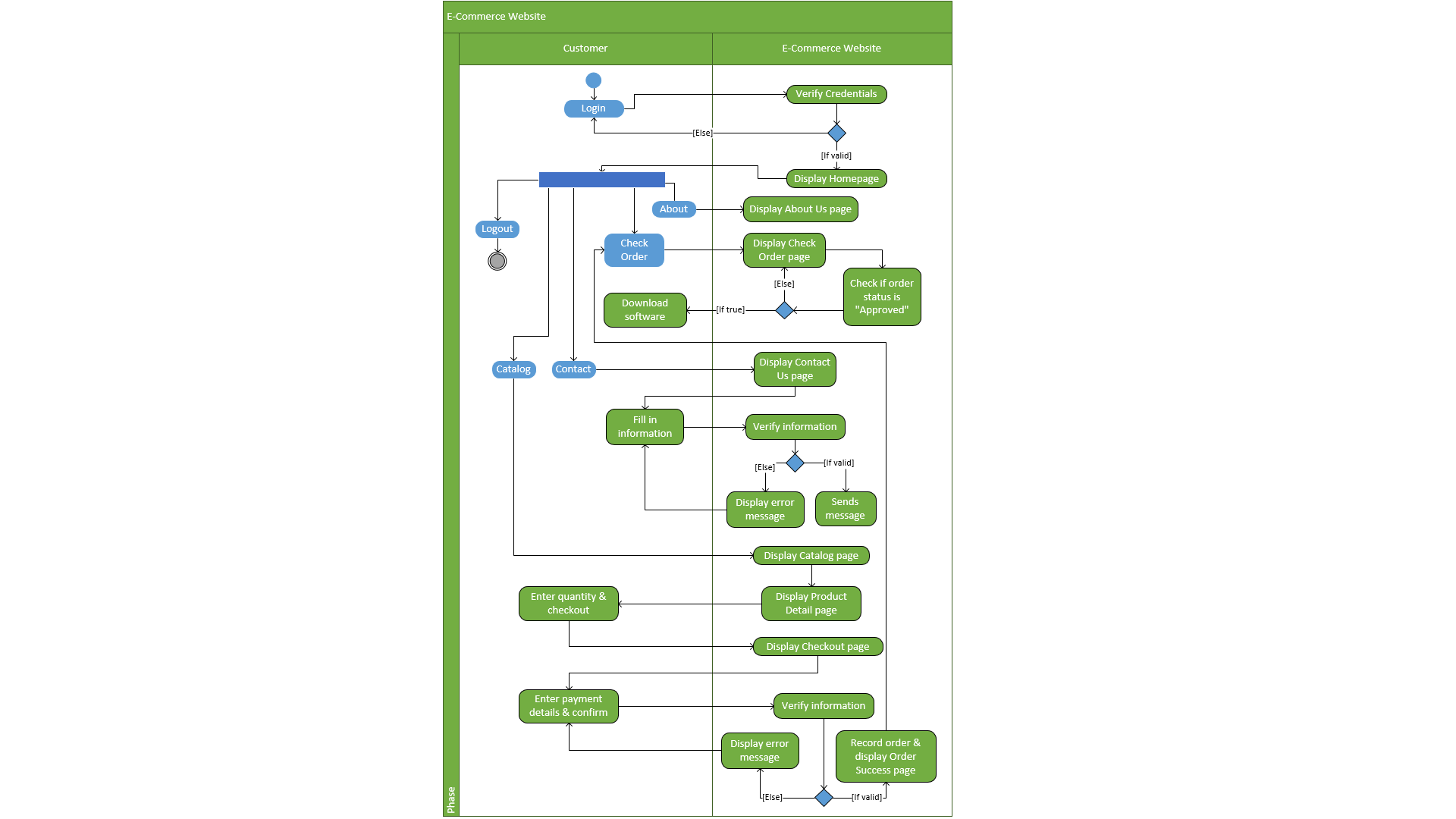


As can be seen in the use case diagram, Visitors will use the website to browse products. They cannot perform checkout because they do not have an account yet, and to perform that they must register an account. Customer who already has an account can login and potentially proceed to checkout if they like the product they are viewing. If they have made an order, they can view their purchase history to see order status. If any of the orders are approved then they can download the software. Both Visitor and Customer can send feedback to SoftwareBiz, which will be received and viewed by Manager. Manager can view customer data and from there they can select a customer and view their order history. They can also view all orders for all products and then select a product to view the sales for that particular product. Admin can view products, and upon any information changes they can choose to edit the product details. If the company has developed a new product, Admin will add that product into the system. If a product is no longer sold by the company, Admin will delete that product. Additionally, Admin is in charge of processing orders, hence they get to view all orders and approve the ones that have been processed.

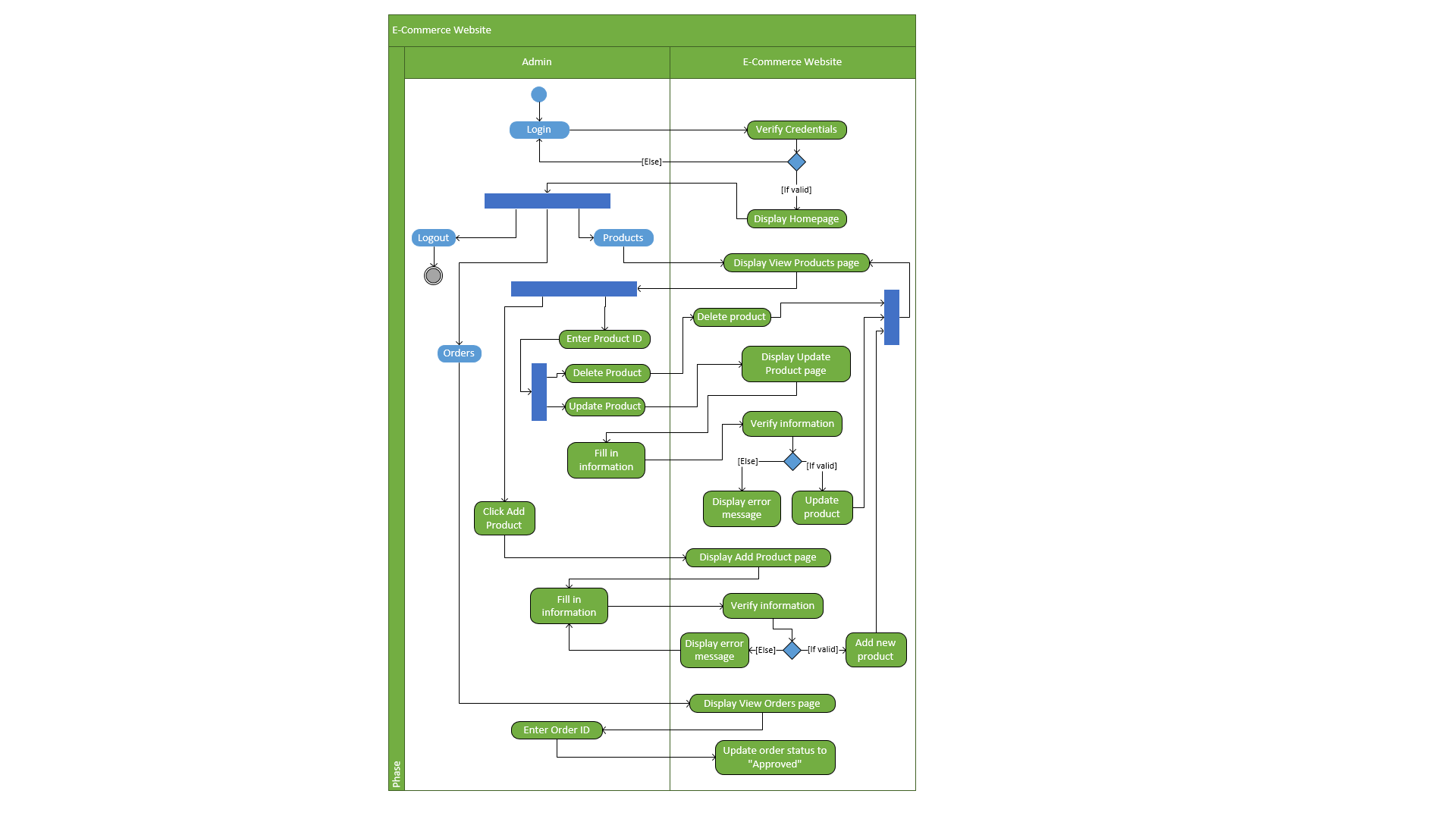
## 3.2 Activity Diagram

### 3.2.1 Customer

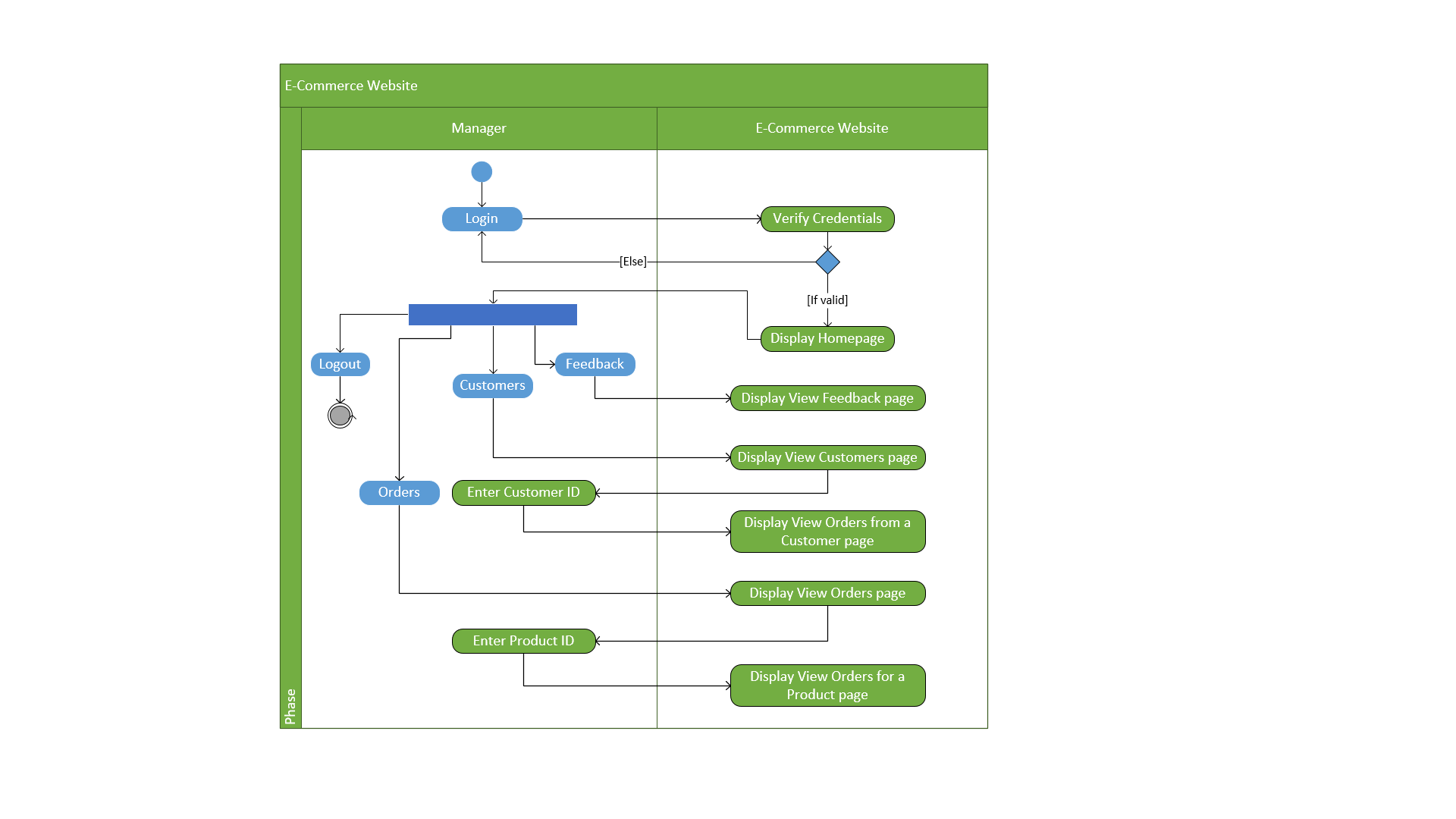




### 3.2.2 Administrator



### 3.2.3 Manager



## 3.3 Entity Relationship Diagram



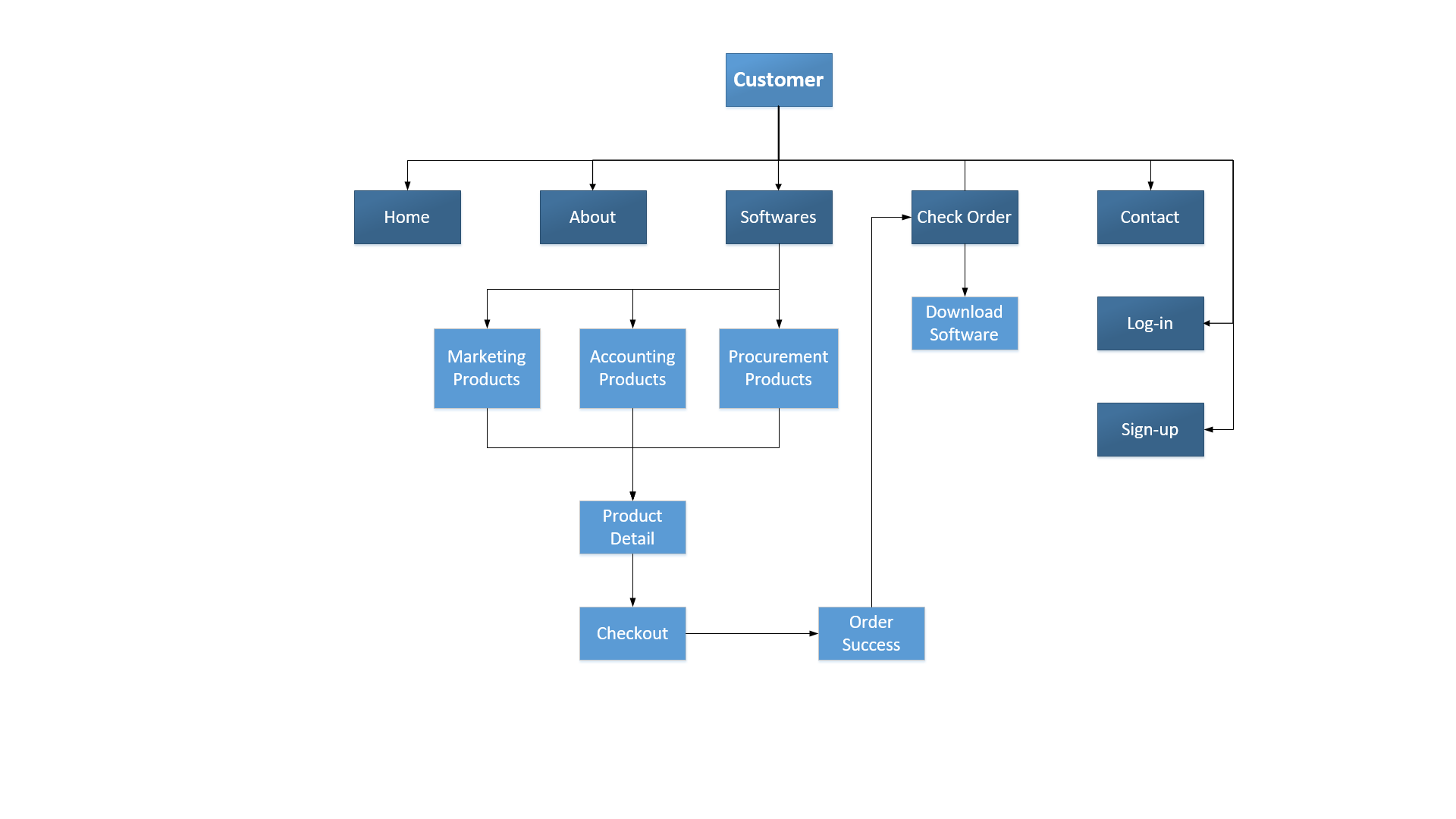
Based on the ERD, the ‘users’ table contains ‘userType’ field which defines whether the user is a Customer, Admin, or Manager. Based on the user type, they can perform different things. For Customer and Admin, one to many users can view one to many software products listed in the web application.

For all user types, one to many users can view zero to many orders/transactions stored in the ‘payments’ table. For instance, a Customer can only view orders made by themselves, while Admin and Manager can view all orders from everyone. The ‘payments’ table contain 2 foreign keys highlighted in red, which are the Product ID and User ID to show which user does the order belong to and which product was purchased within that order. The ‘users’ table also contains 2 foreign keys Payment ID and Product ID which will be called when Manager wants to view what orders have a particular Customer made.

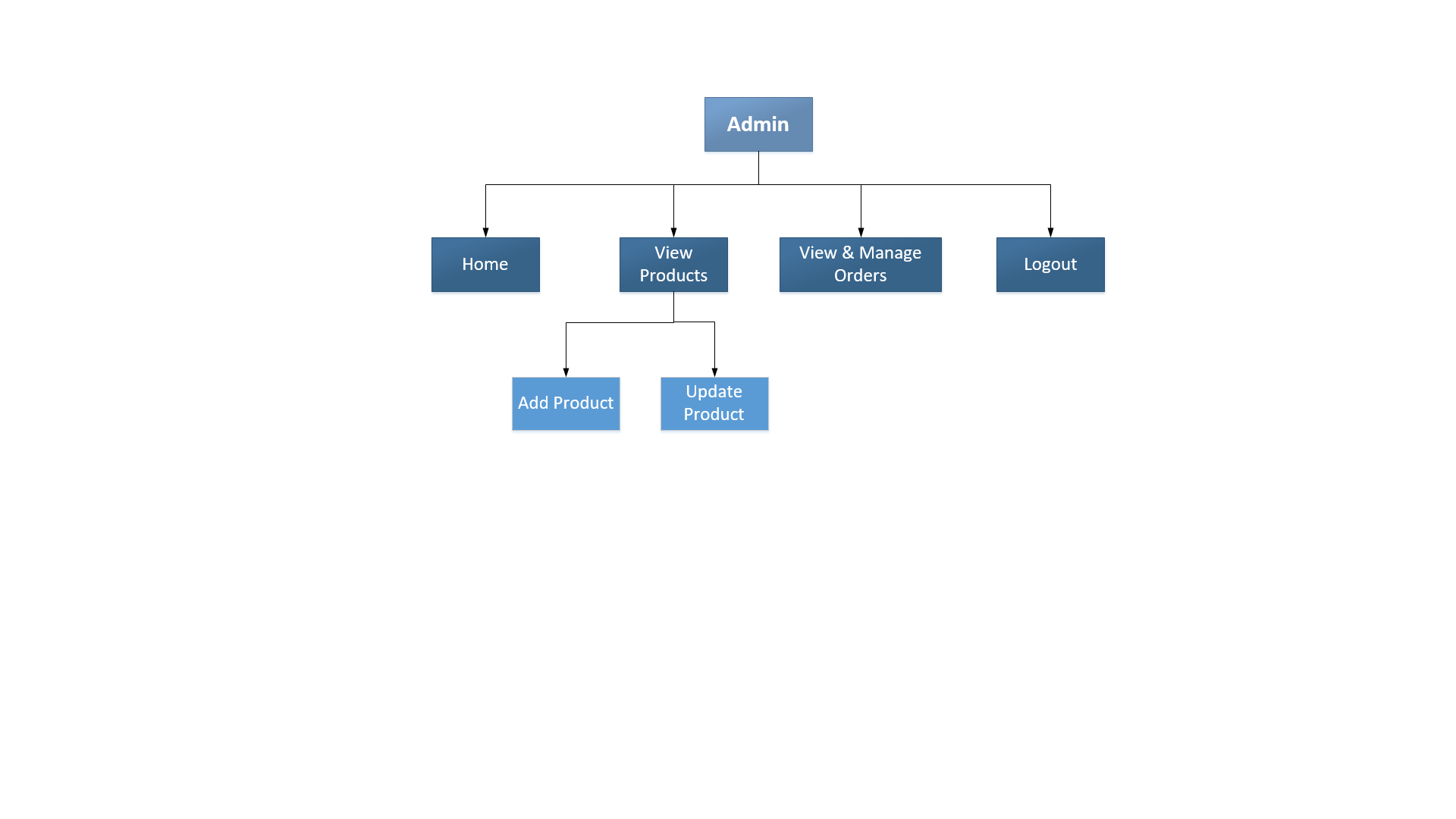
Furthermore, only one software can be included any one order, because there is no cart due to stakeholders’ request that they want the orders to be placed based on a particular product. Lastly, one to many managers can access zero to many feedback submitted by visitors. When logged in, Customer still uses the same feedback form as unregistered visitors. Hence they still need to enter their name and email manually instead of it being called from the ‘users’ table. Therefore, the only users who will read the data from ‘feedback’ table are the Managers.

## 3.4 Navigation Structure

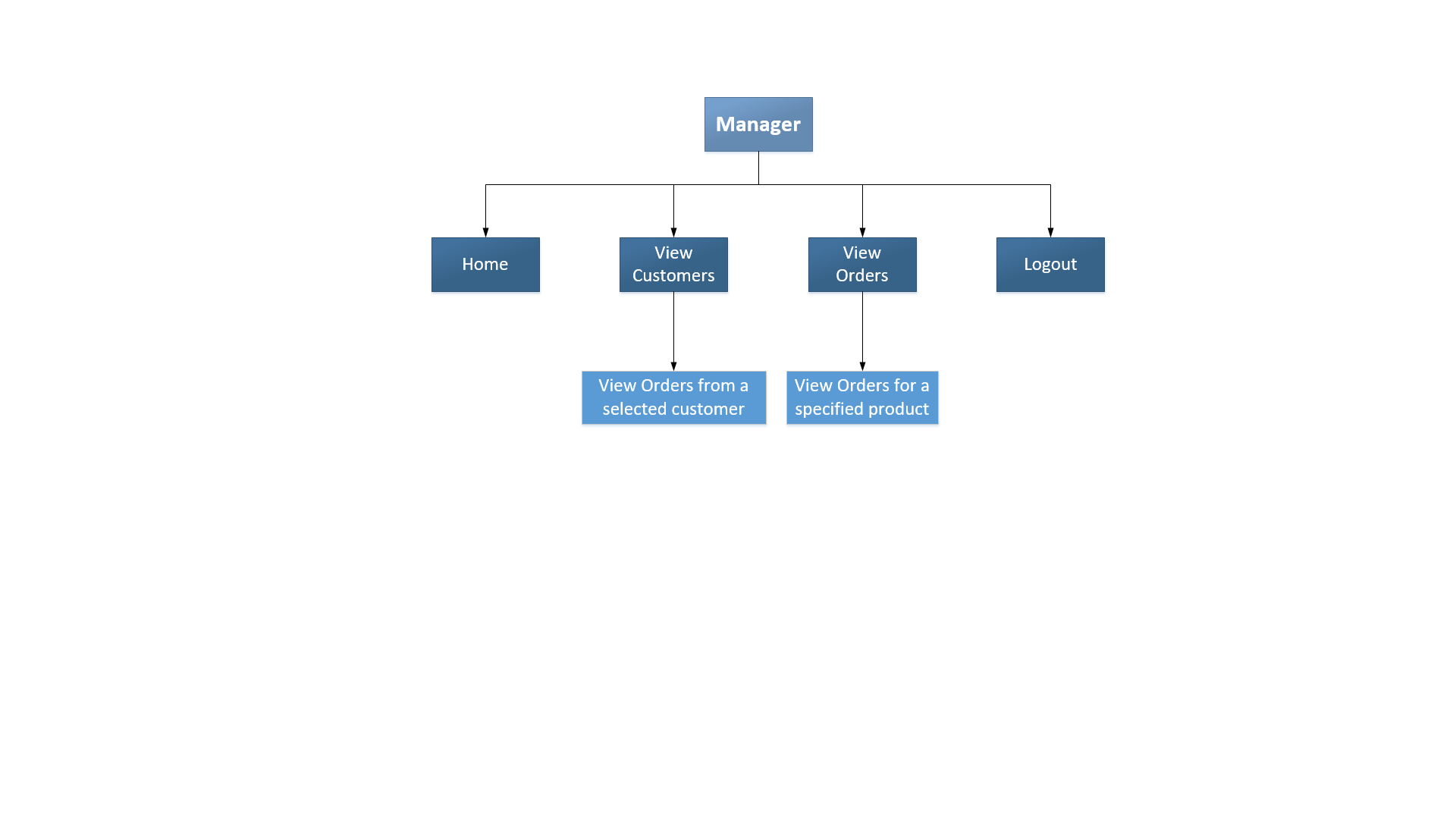
### 3.4.1 Customer



### 3.4.2 Administrator

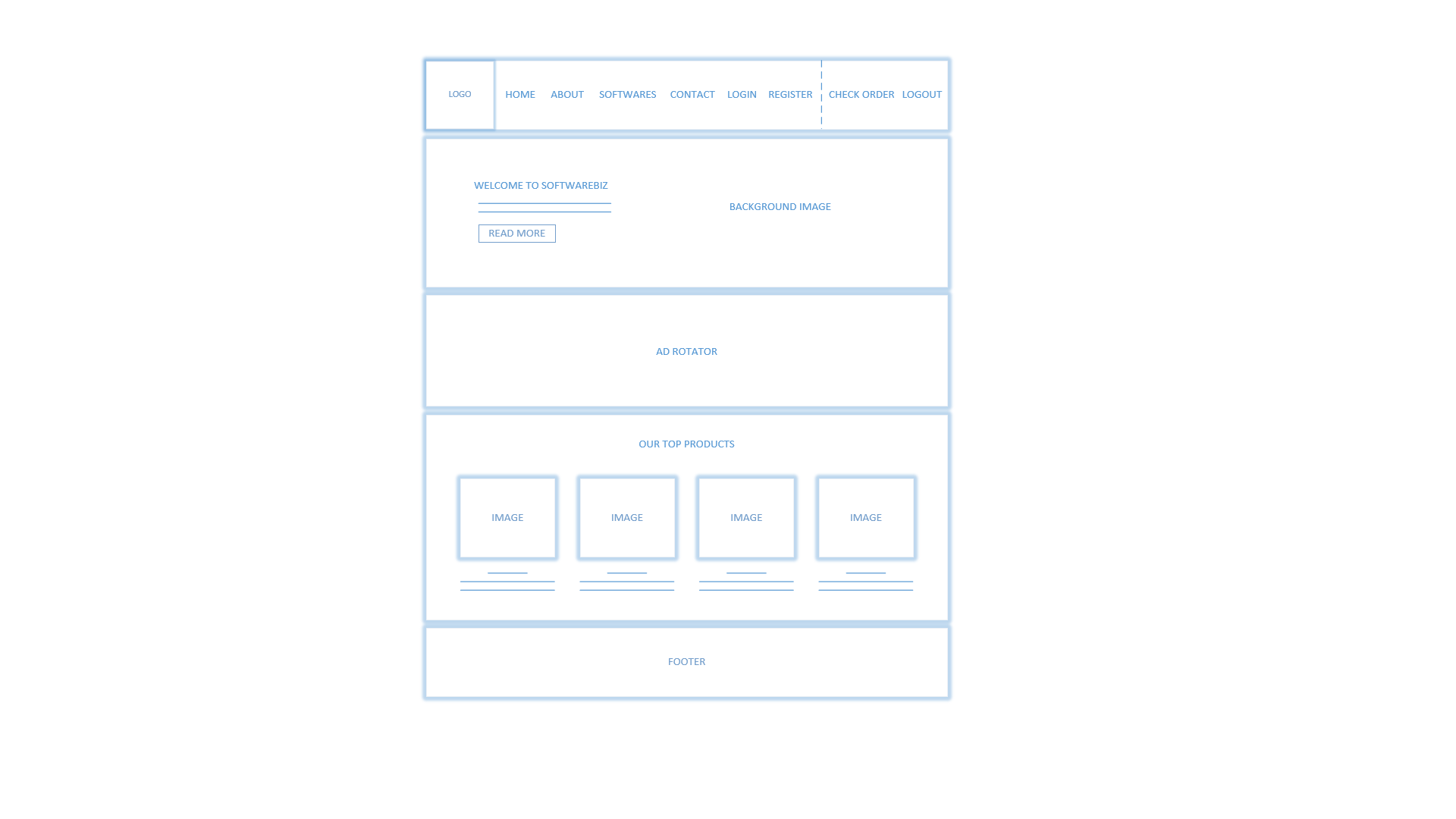


### 3.4.3 Manager



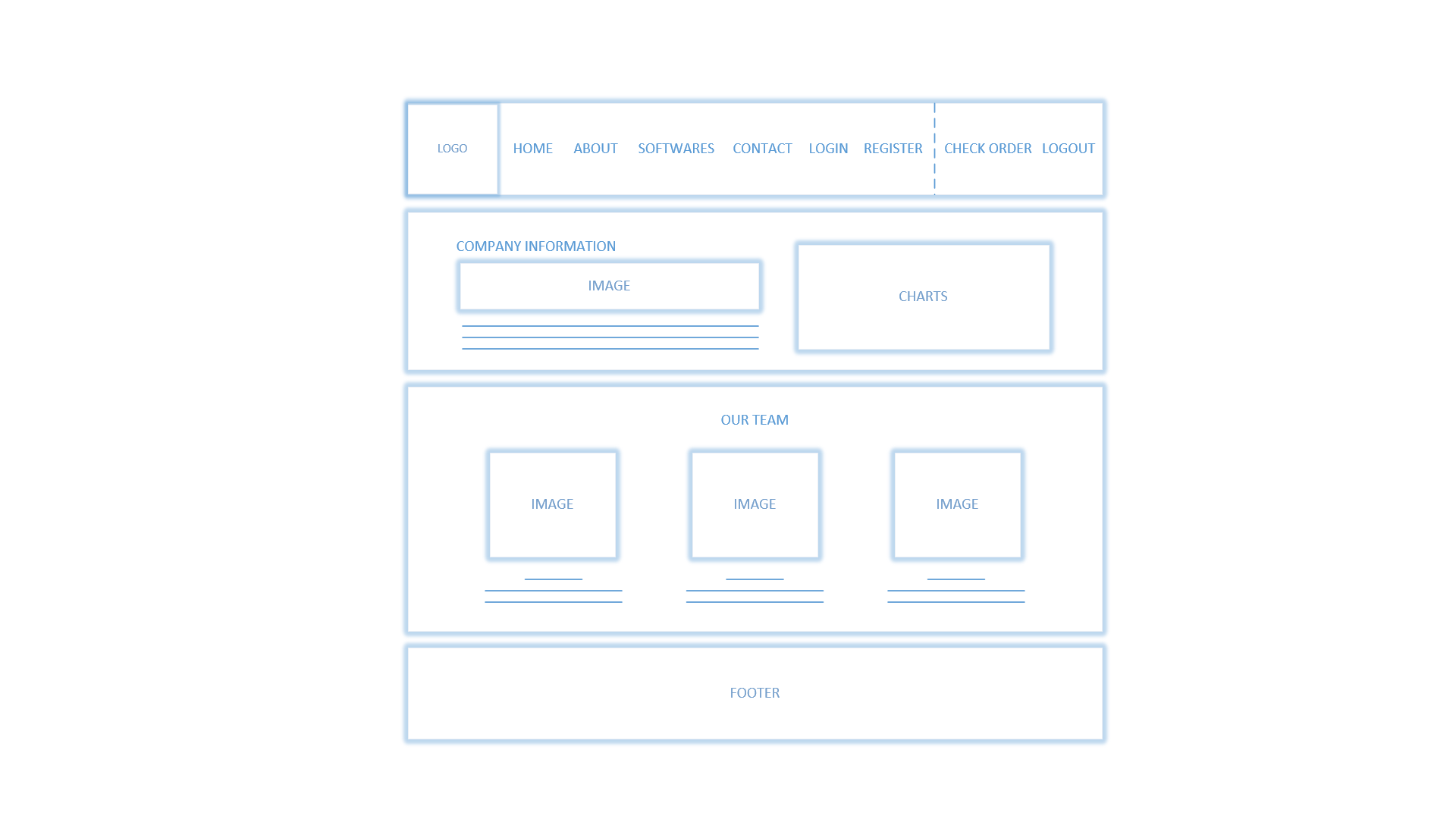
## 3.5 Presentation Design

### 3.5.1 Visitor/Customer Homepage



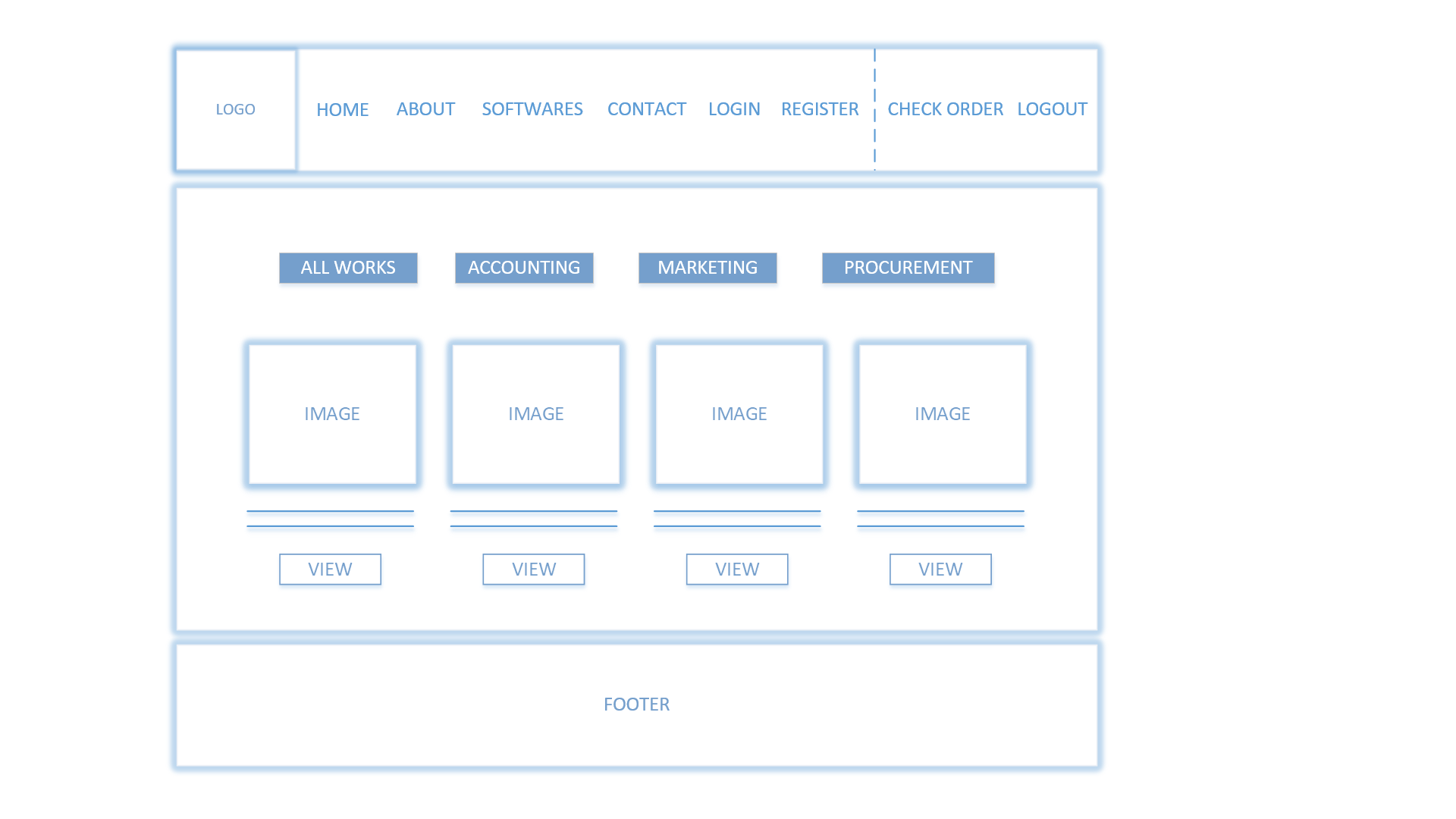
The homepage contains 2 types of XML integration. First is the Ad Rotator which reads data from ‘AdXml.xml’ to display images in sequence. Also, for displaying images and details of top products sold by SoftwareBiz, the ListView reads static data from another XML file that stores the information.

### 3.5.2 About Us



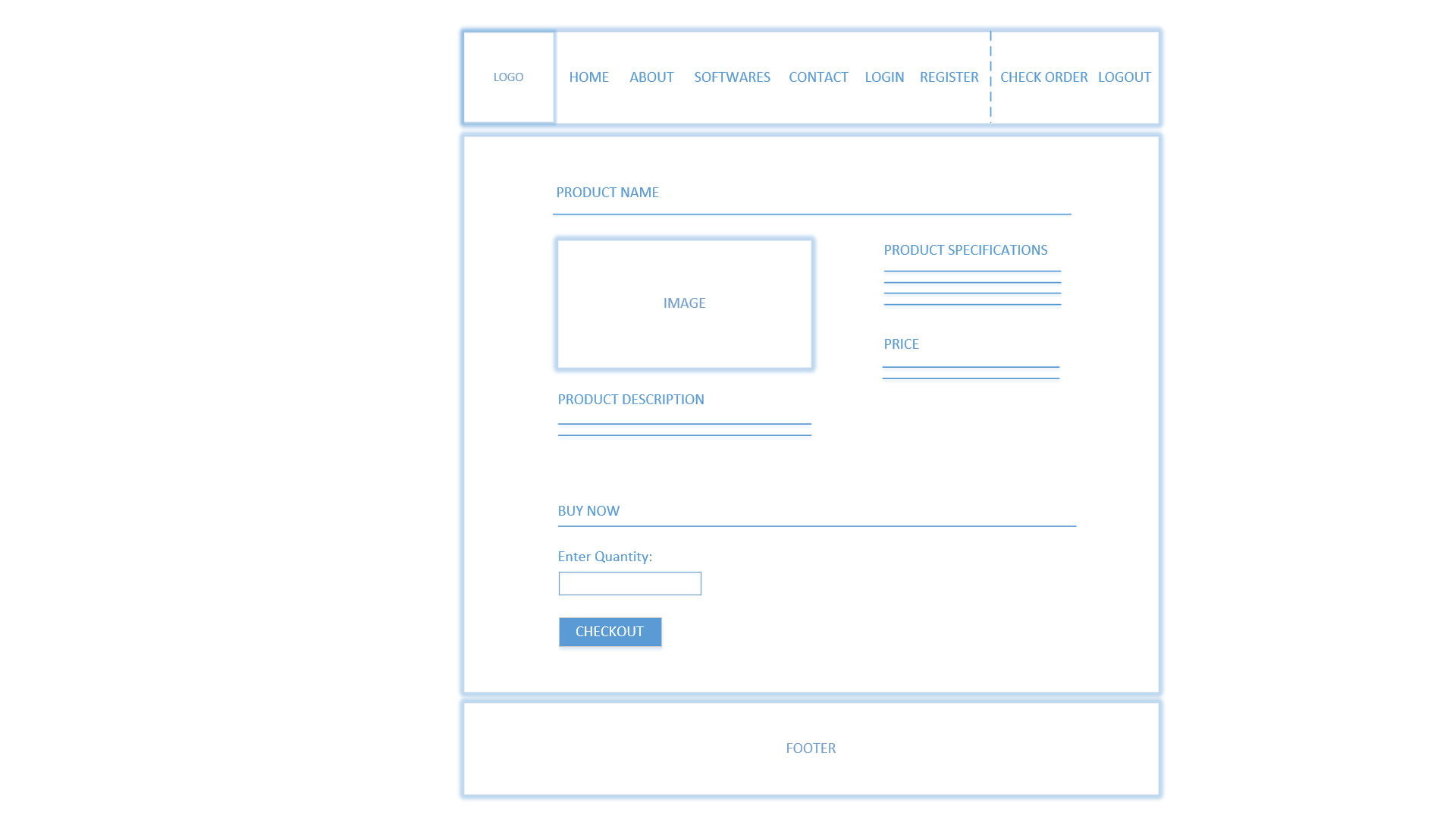
This page displays information regarding the company for first-time visitors. The charts are graphical representation of the company’s specialities. There is an XML integration used to display static information for ‘Our Team’, where the content for image, person name, and person description are coming from an XML file.

### 3.5.3 Product Catalogue



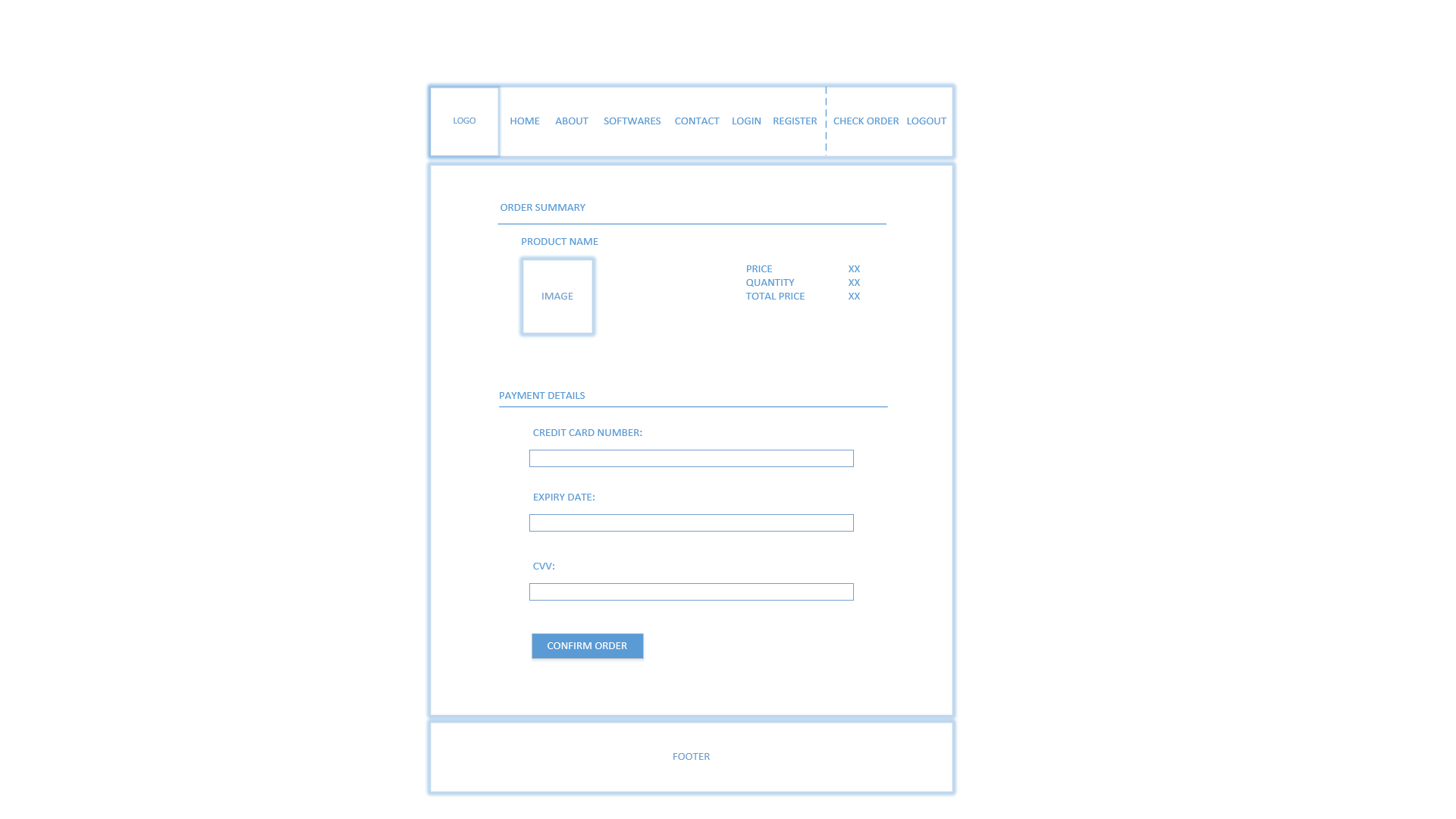
In product catalogue, user can filter products based on category by clicking those category buttons: All Works, Accounting, Marketing, and Procurement. Products are displayed from the ‘softwares’ table in the SQL database.

### 3.5.4 Product Details



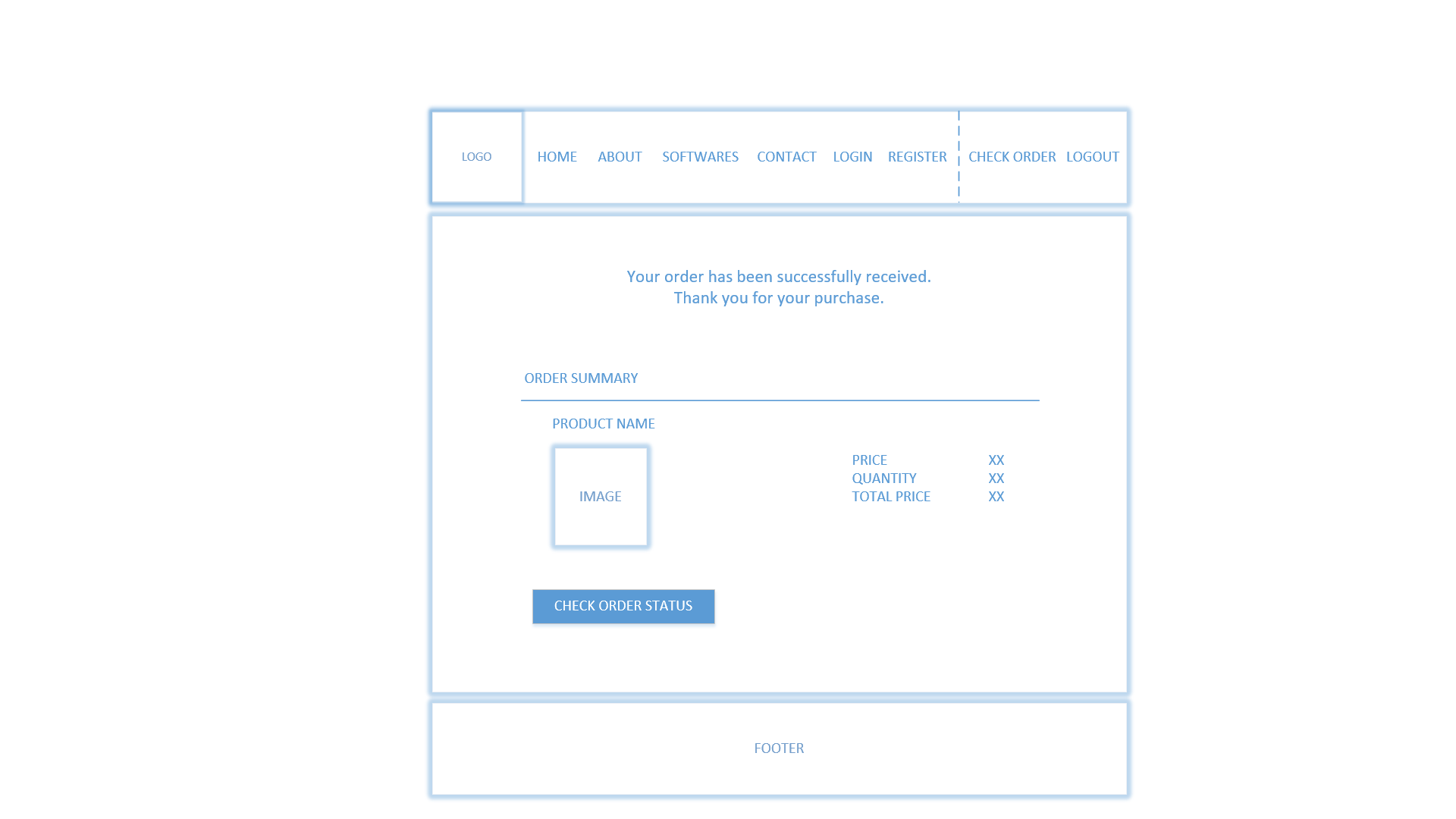
When user clicks on the selects a product from the catalogue, they will be redirected to this page which shows product details of the selected item. If user is interested in buying, they can enter the quantity that they want and click Checkout, which will redirect them to Checkout page if user has logged in, otherwise they will be redirected to Login page.

### 3.5.5 Checkout



This page shows order summary and requires customer to enter payment details in order to place an order.

### 3.5.6 Order Success



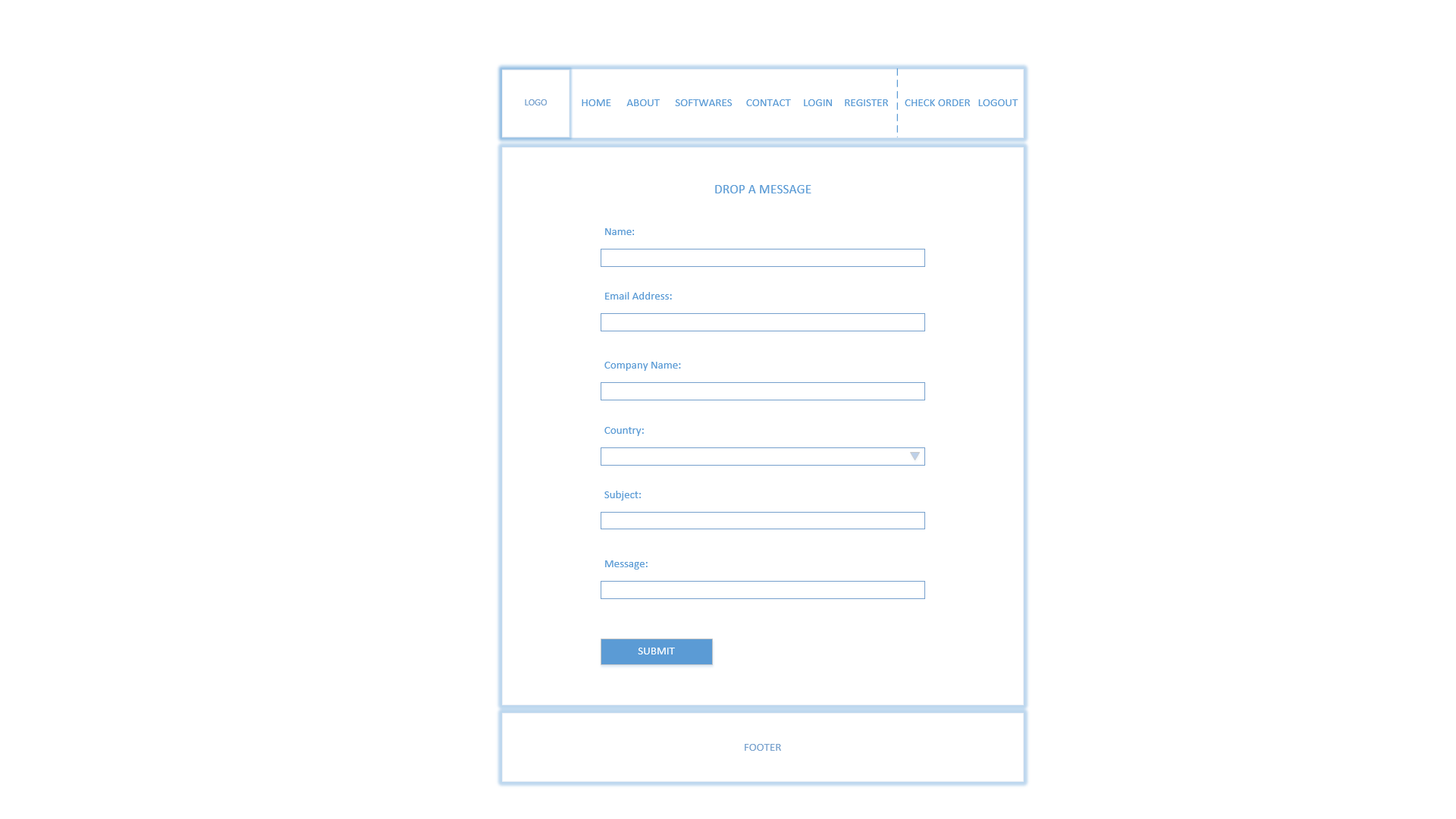
After the online payment had been done, customer is redirected to this page which indicates that their order (as shown in the order summary) was successfully submitted into the system and is being processed.

### 3.5.7 Check Order



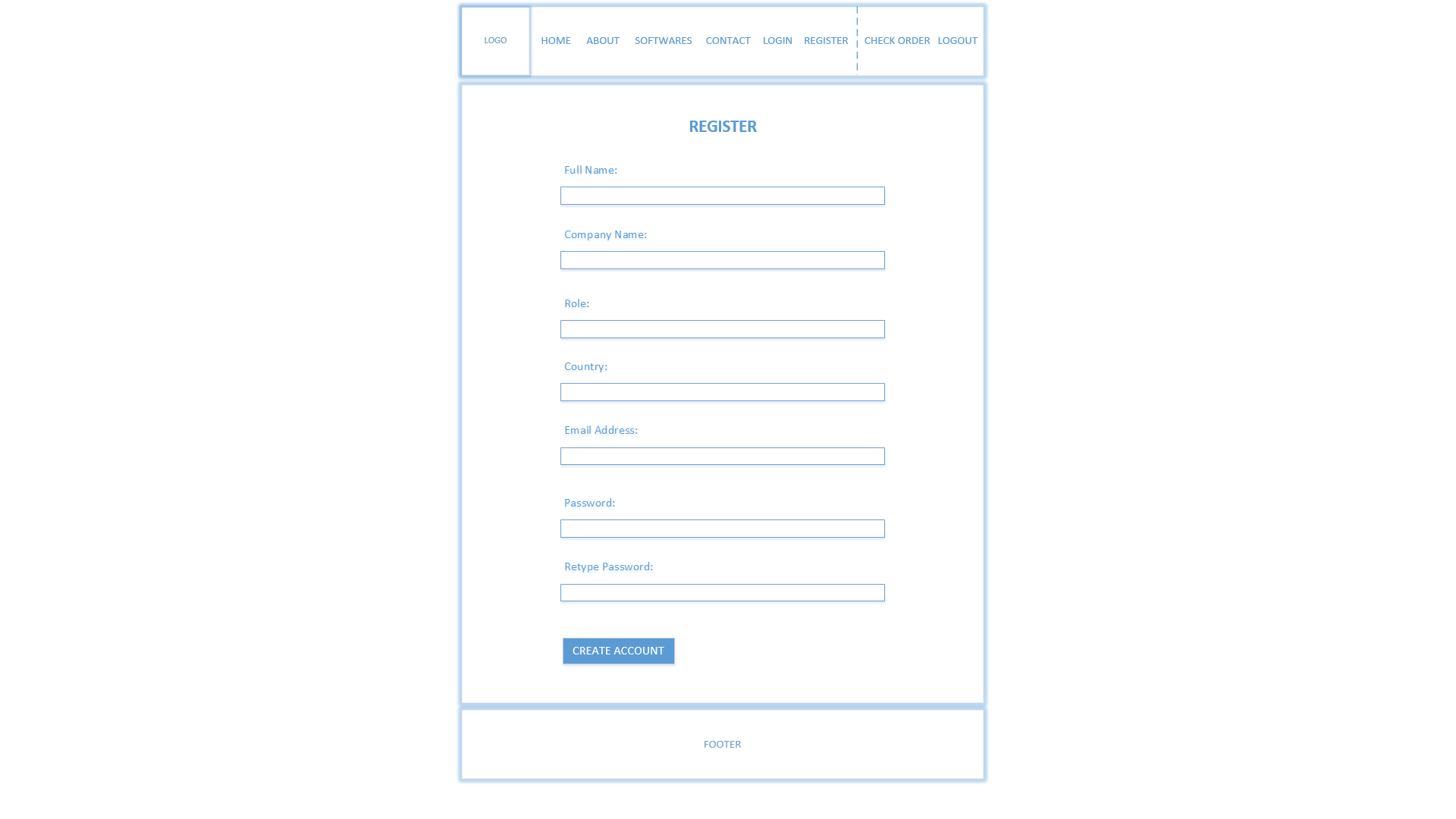
This page shows customer their transaction history in GridView. If any of their order has been approved, it will show in the list of approved orders with a Download button for customer to download the software. After it has been downloaded, the order will be moved to transaction history with the order status changed to ‘completed’.

### 3.5.8 Submit Feedback



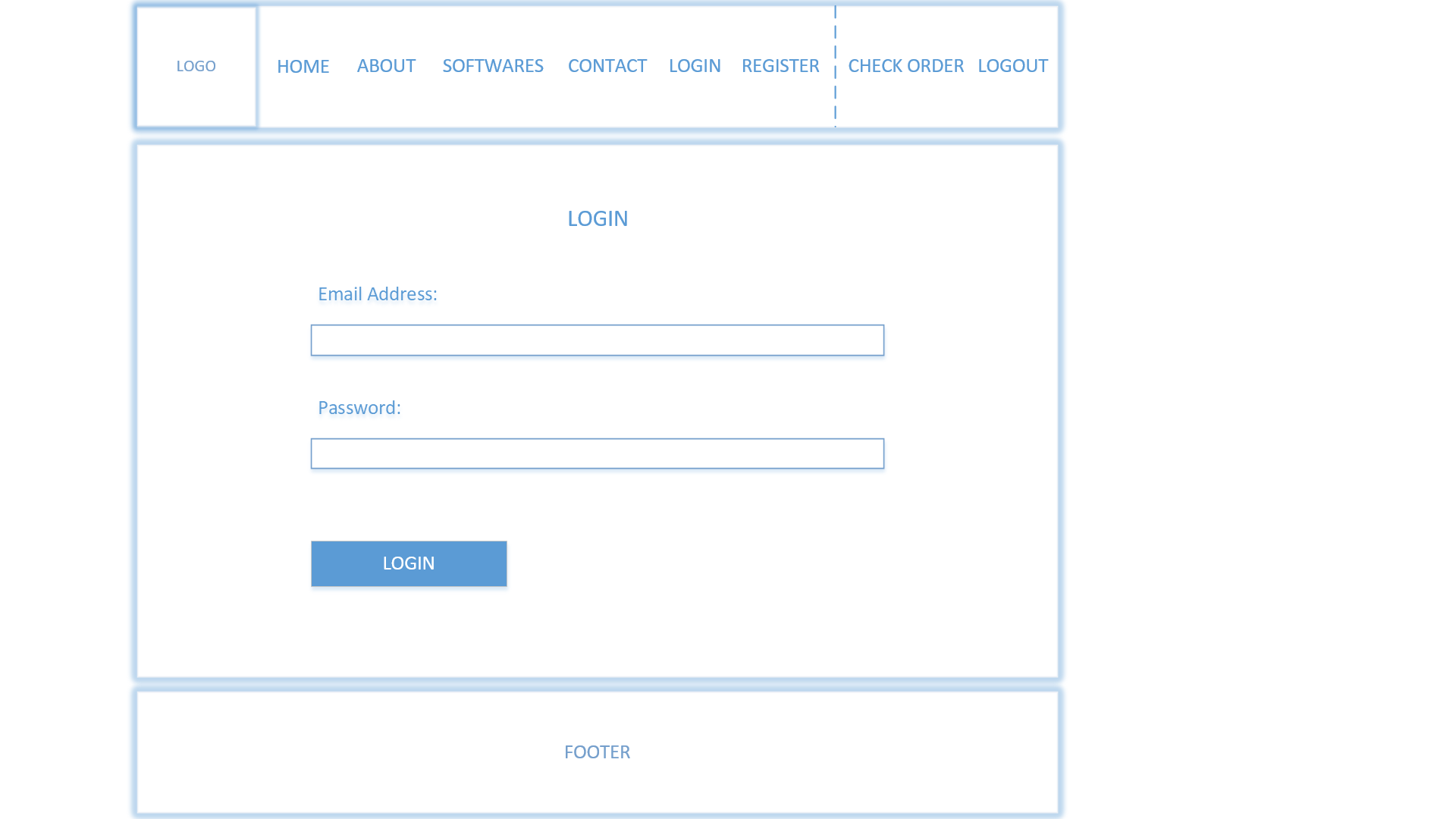
This page is shown when visitor/customer clicks the Contact link in the header. They can contact SoftwareBiz by entering this feedback form. All the fields are required to be filled before submission.

### 3.5.9 Register



Visitors can register a new customer account by entering all the required details on this page. The information will be saved into ‘users.xml’ file as well as ‘users’ table in the SQL database. To ensure user entered the correct password, user has to re-enter password in an additional field called ‘Retype Password’ which will match its content with the ‘Password’ field. Otherwise, an error message will show that prevents user from creating an account. After an account has been successfully created, the page will refresh with empty fields and a message that indicates successful registration.

### 3.5.10 Login



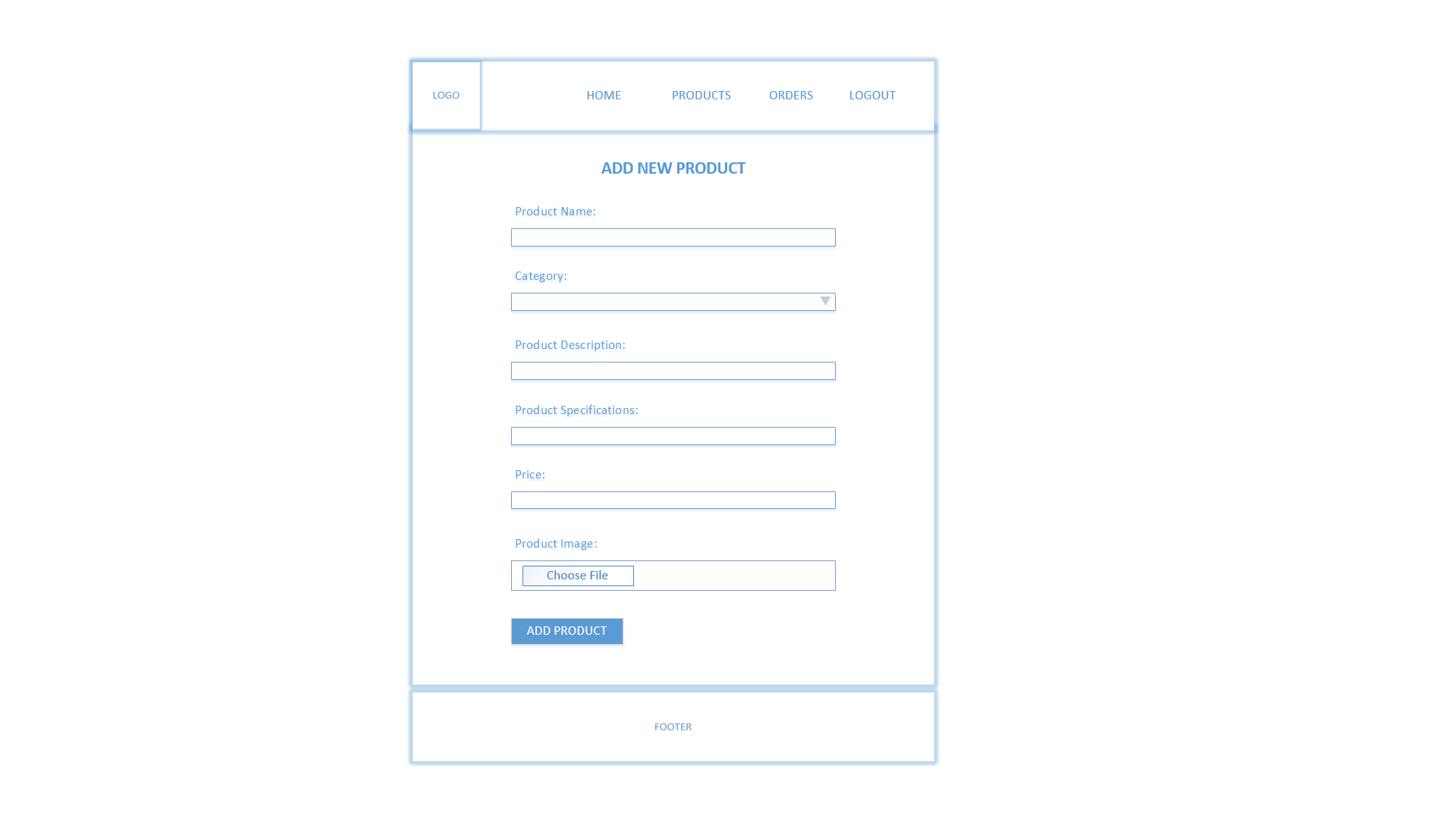
This web application uses single sign-on, which means all user types must use this page for login. It will verify the entered credentials against the XML file which contains all user accounts, if there is a match, it will check the user type and then redirect user accordingly. If there is no match, an error message will show.

### 3.5.11 Admin View & Manage Products



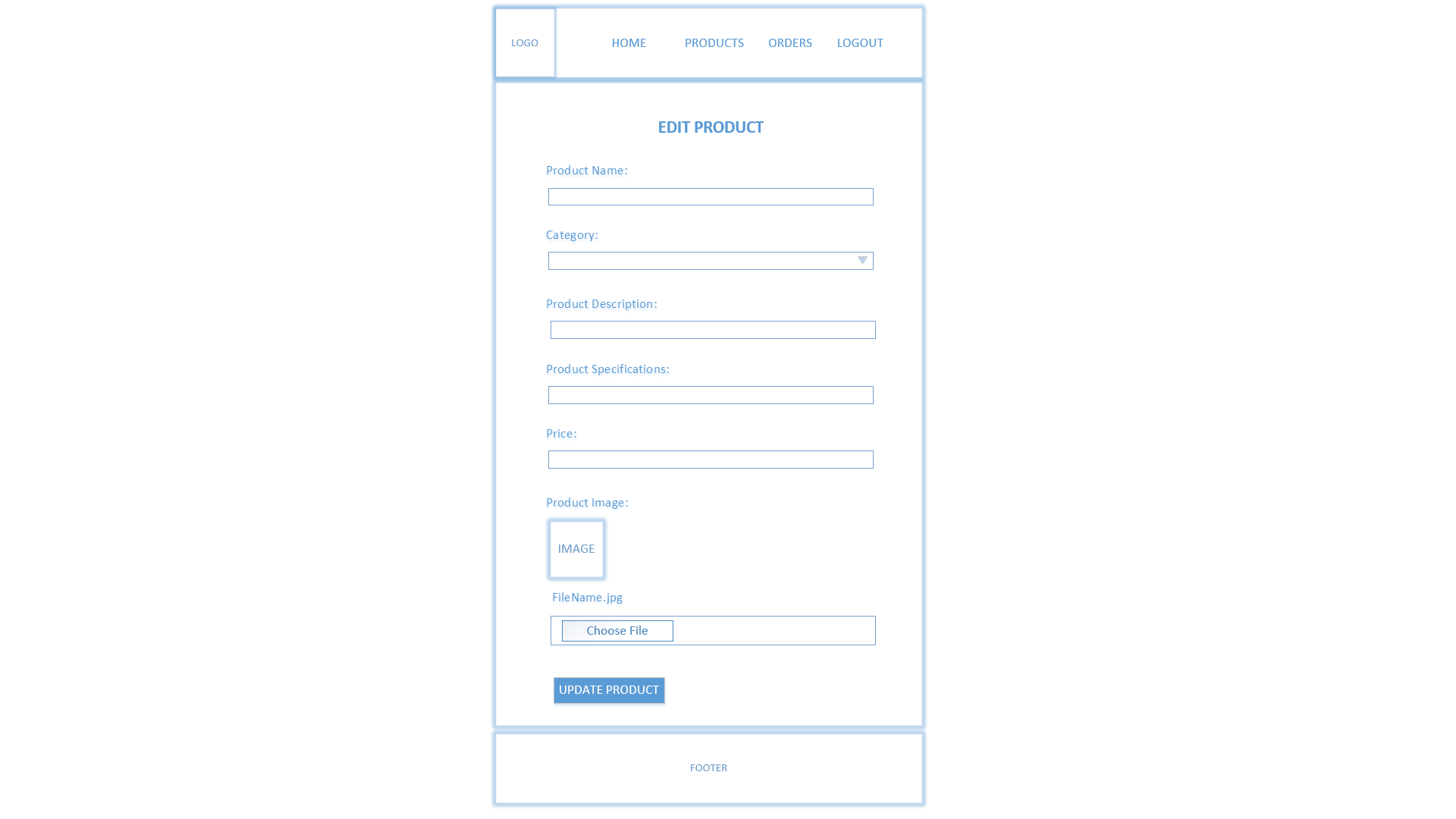
Notice that the main menu for Admin page is different from Customer page, as to adjust to what functionalities Admin needs from the web application. Firstly, they can view a list of existing products through this page, shown in a sortable GridView where user can click the any Header to sort items in ascending or descending order. It also has a button to redirect Admin to Add Product page. Admin can select a product to be edited or deleted by entering the Product ID and clicking Edit or Delete button. Delete button will immediately deletes the product from the list, whilst Edit button will redirect Admin to Update Product page.

### 3.5.12 Admin Add Product



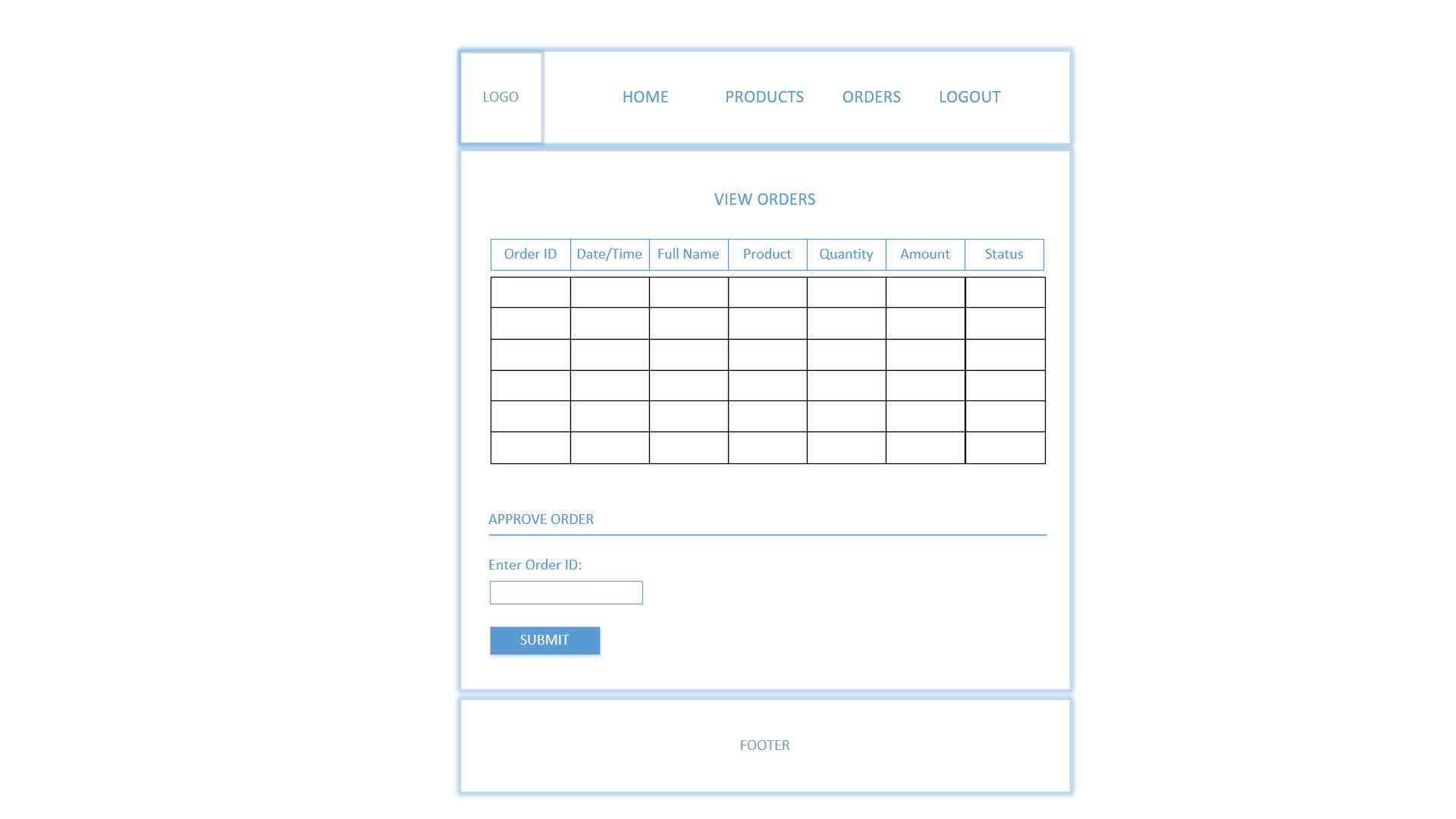
Admin has to enter all these fields to add a new product. The category options in the dropdown list come from ‘category.xml’. The product description and specifications are using multi-line textbox to allow Admin to adjust the textbox size so that they can write long paragraphs comfortably. This page uses File Upload control to upload product image and save it in a specific folder.

### 3.5.13 Admin Update Product



This page will be filled with existing details of the product that matches the Product ID entered in the View Products page. The ID is passed in query string, and referred to in the SQL query for this page. Any changes made on this page will be saved in the database, replacing the old data.

### 3.5.14 Admin View & Approve Orders



Another duty of Admin is to approve orders. They can view a list of orders in a sortable GridView and choose the one to approve by entering the Order ID and clicking the Submit button which will change the order status from ‘processing’ to ‘approved’.

### 3.5.15 Manager View Customers



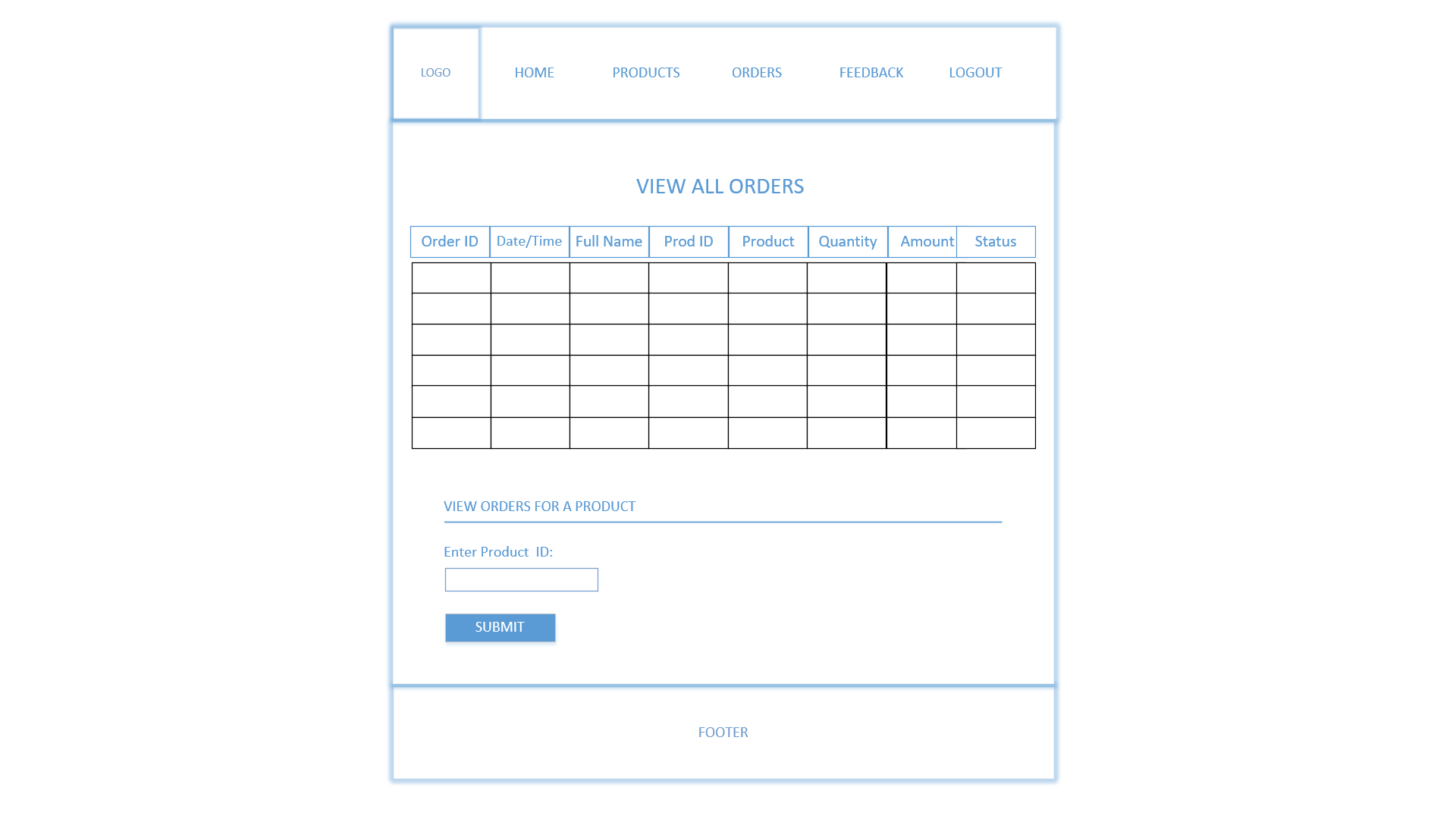
Manager can view all customers and their details in a GridView to get an overview of where their customers are from, which company, and what position do they hold within their that company. The GridView is sortable where Manager can click the any Header to sort items in ascending or descending order, therefore finding information faster and as preferred. Manager can also select a customer to view orders made by that customer in the past by entering the User ID and clicking the button.

### 3.5.16 Manager View Orders from a Selected Customer



According the User ID entered in the previous page, all transactions made by that customer are displayed in this page. This is to understand each customer a little better, identifying their preferences and needs based on purchase history. It can also be used to check a particular customer’s order details to cross-check with any complaints or miscommunication received from a customer’s feedback.

### 3.5.17 Manager View All Orders



Additionally, Manager should be able to view all sales made by the business, identifying the sales trend and assessing performance of the company. The data is shown in a sortable GridView therefore Manager can view information in an order that they prefer. Manager can select a particular product by entering the Product ID to check the sales for that particular product shown in the next page.

### 3.5.18 Manager View Orders for a Selected Product



Based on the Product ID entered in the previous page, the list of orders for a specific product is shown. Manager can sort the data according to Company Name and Country so that they can understand which company prefers this product and which countries use this product the most.

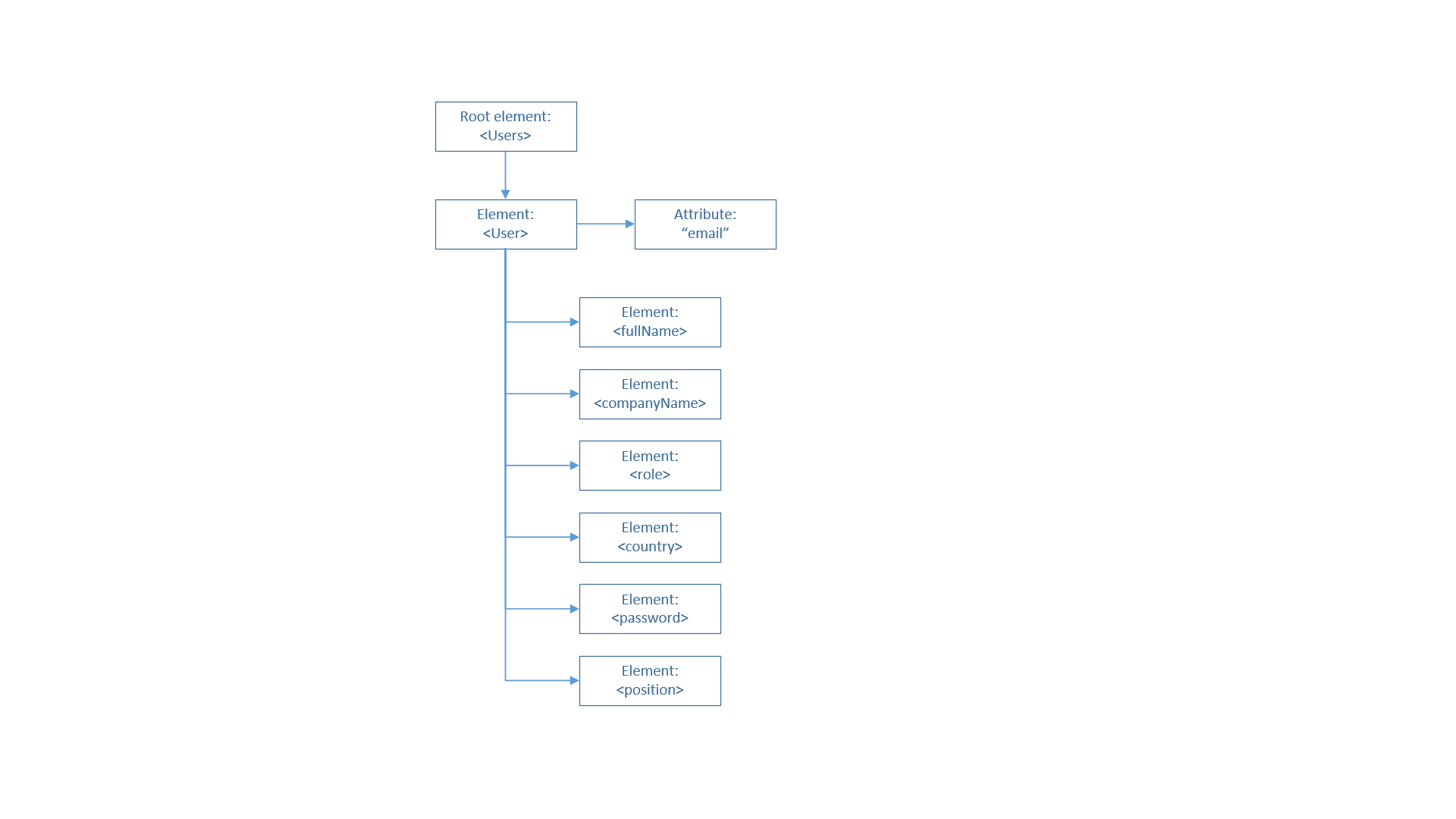
### 3.5.19 Manager View Feedback



This page fetches content from ‘feedback.xml’. It displays feedback in this manner. The boxes are borders of labels, not editable textboxes. It shows date/time of when the feedback was submitted so that the Manager can distinguish between the old ones and new ones that they must attend to. There is data paging which shows 3 feedback per page, hence Manager has to go to the next page to view more.

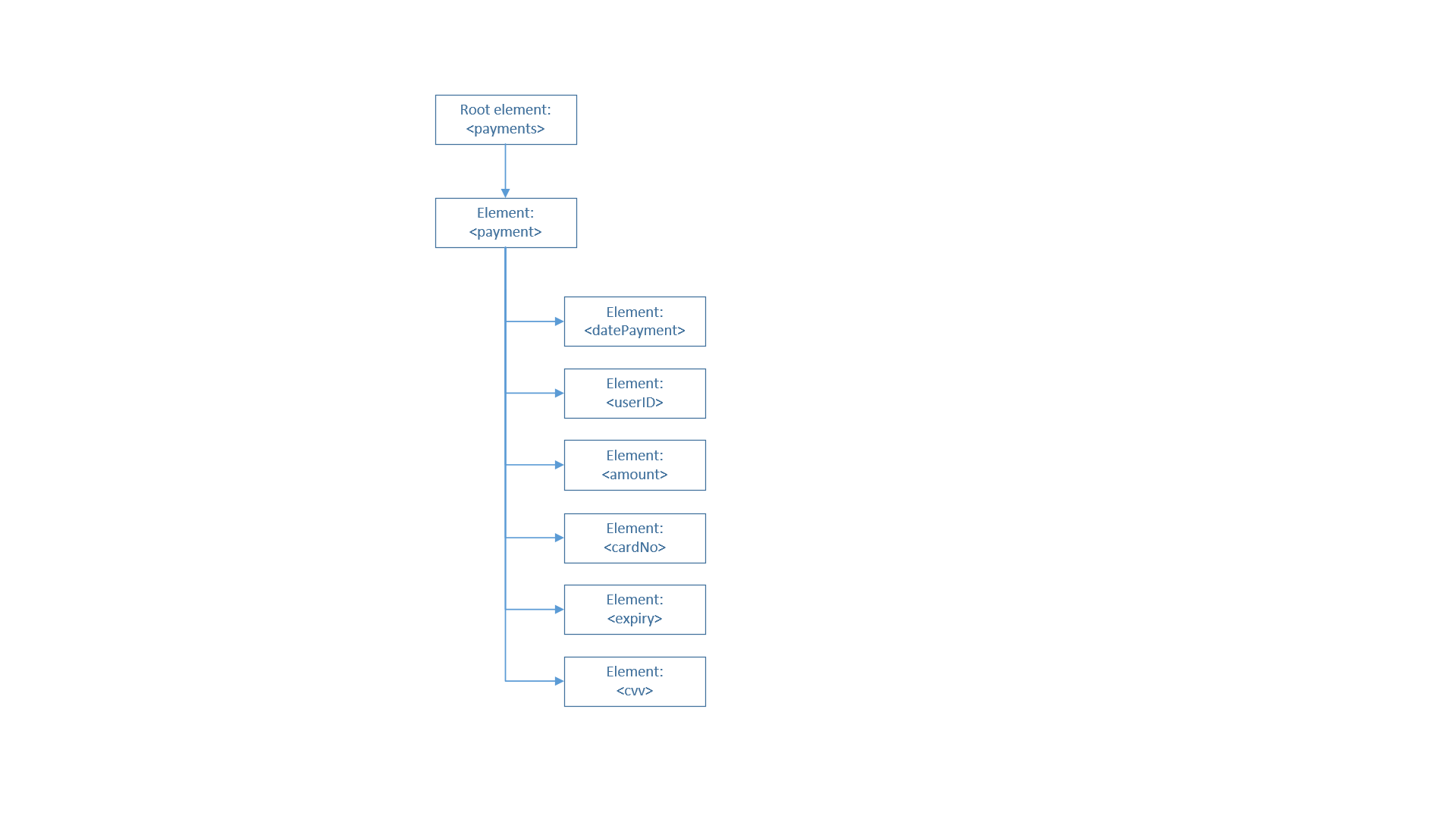
## 3.6 XML Tree

### 3.6.1 Users



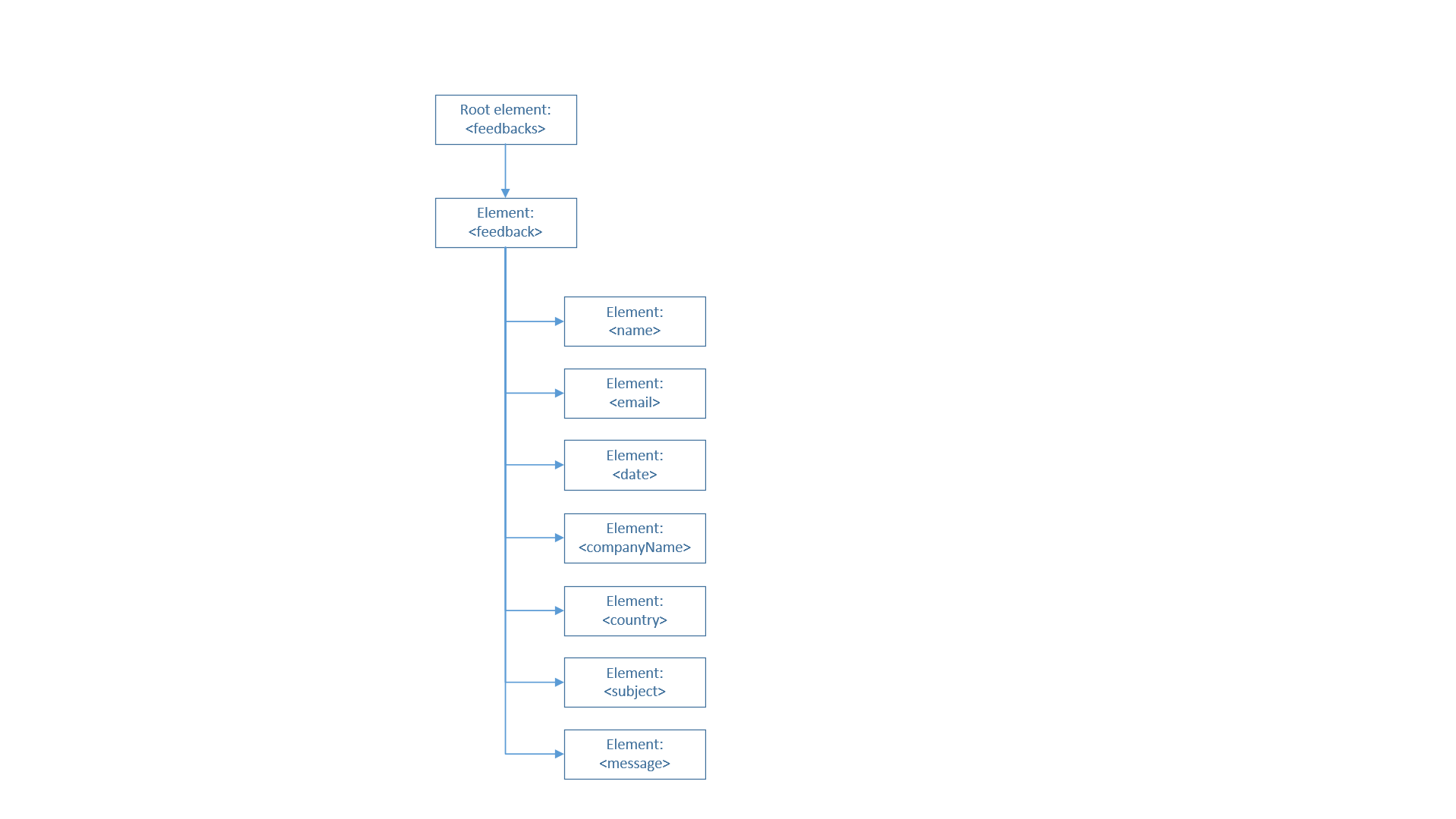
The ‘users.xml’ file will be structured this way. The function is to save users information which will be referred to for login verification. New user registration will add a new <User> element into this file. The <position> element defines the user type: customer, admin, or manager, hence during user will be redirected to their respective homepage upon registration. Since managers want to understand where customers are coming from, the information such as company name, role in the company, and country in which the customer is located in are recorded. To distinguish between <User> elements, the unique email attribute is used as identification.

### 3.6.2 Payment



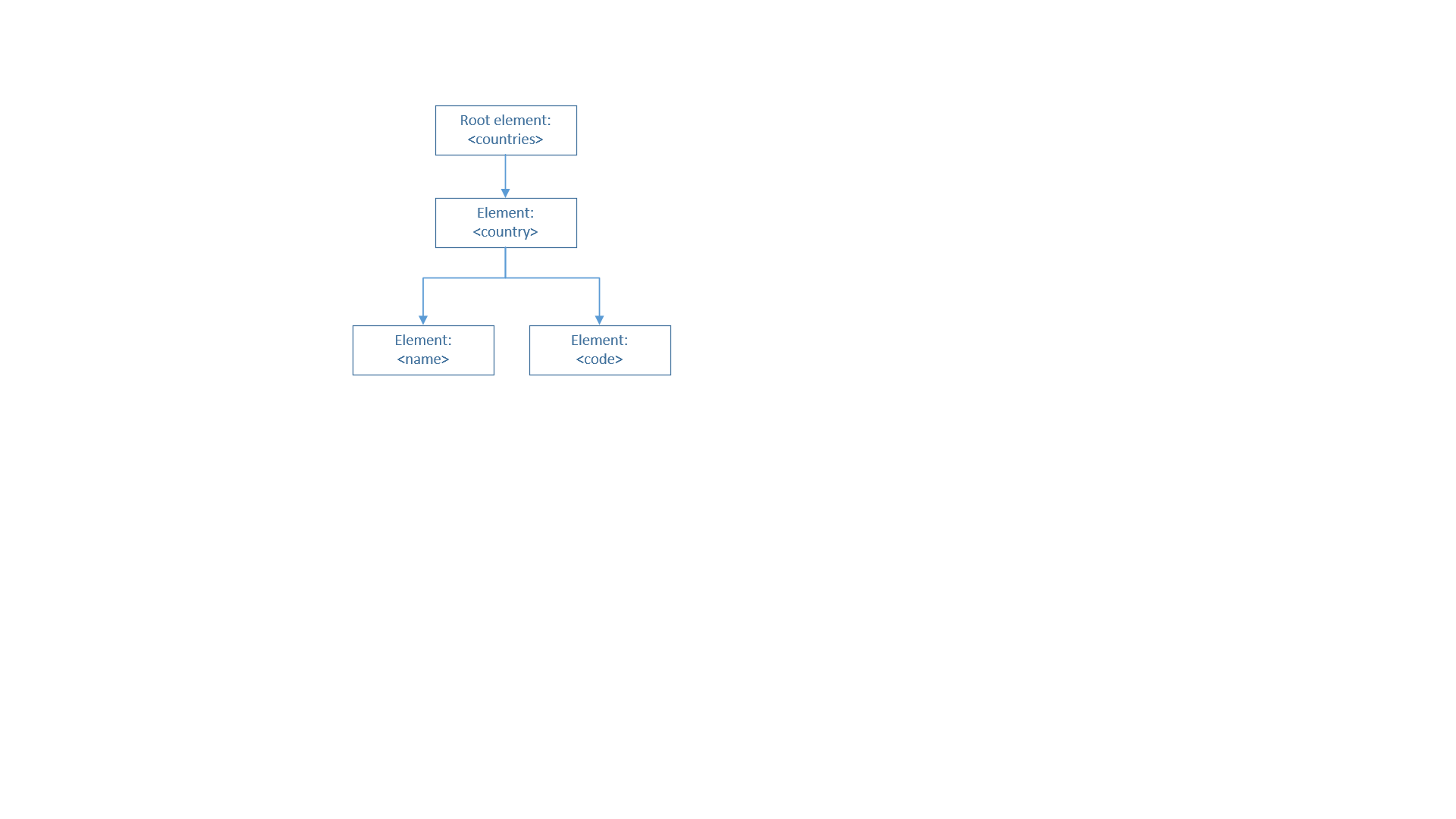
The ‘payments.xml’ will be structured this way. It includes all the essential information needed for payment verification with the Bank, such as card number, card expiry date, and CVV as found in most e-commerce websites.

### 3.6.3 Feedback



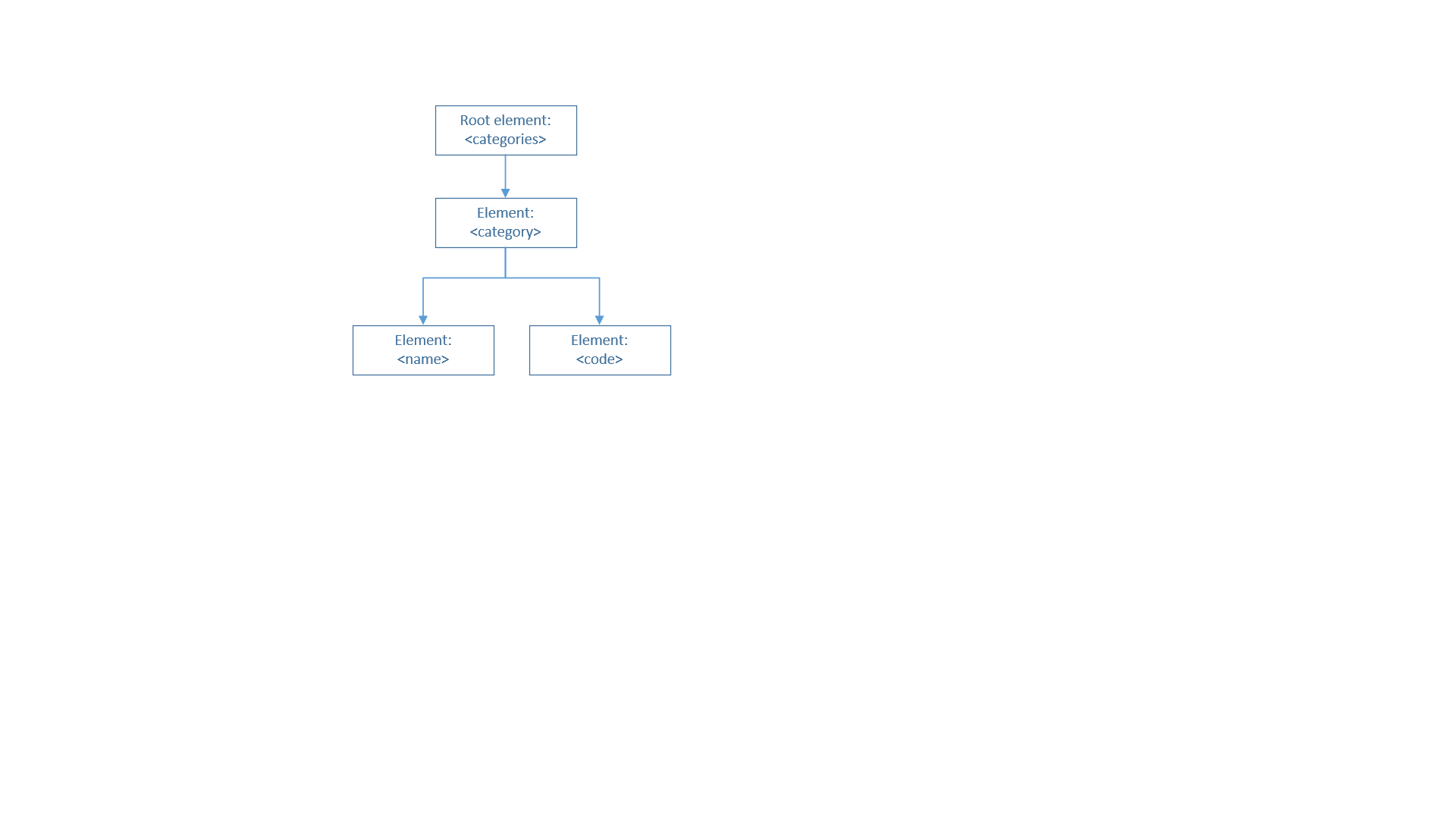
The ‘userfeedback.xml’ will be structured this way. It is used to save details of feedback submitted by visitor or customer. When Manager wants to view messages from customers, this file will be referred in order to retrieve the content. It includes name and email to allow SoftwareBiz’s staff to reply to the person giving the feedback. It includes date to identify whether the feedback was old or new. It also includes company name and country to understand where the person is coming from. The subject will give a quick heads-up of what the message is about.

### 3.6.4 Country



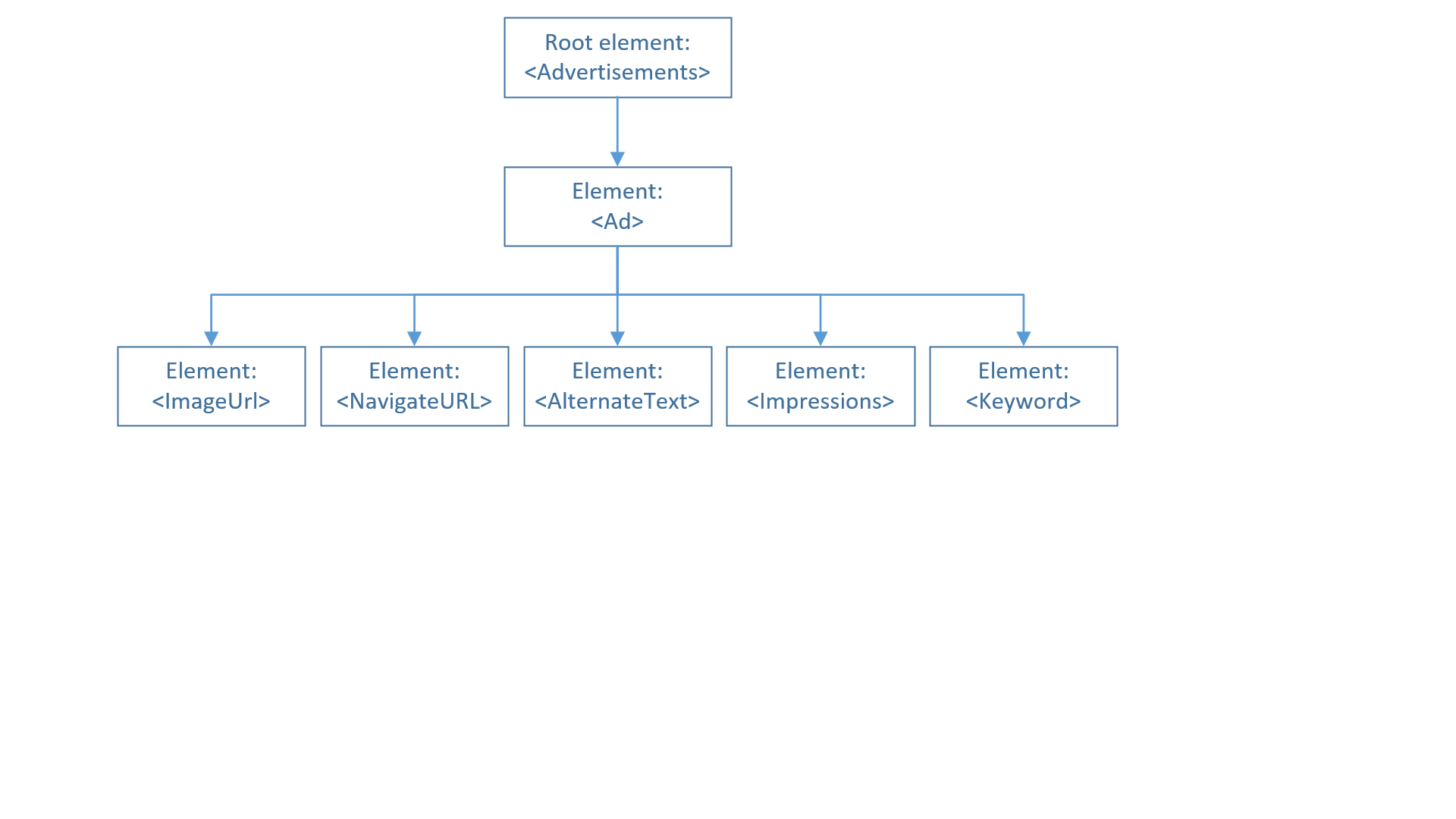
The ‘country.xml’ is used by the dropdown list in registration form and feedback form to identify where the customer is coming from as required by the stakeholder. The <name> of the countries will be shown on the frontend to customers, but the <code> will be recorded in the database for easier identification by the Manager.

### 3.6.5 Category



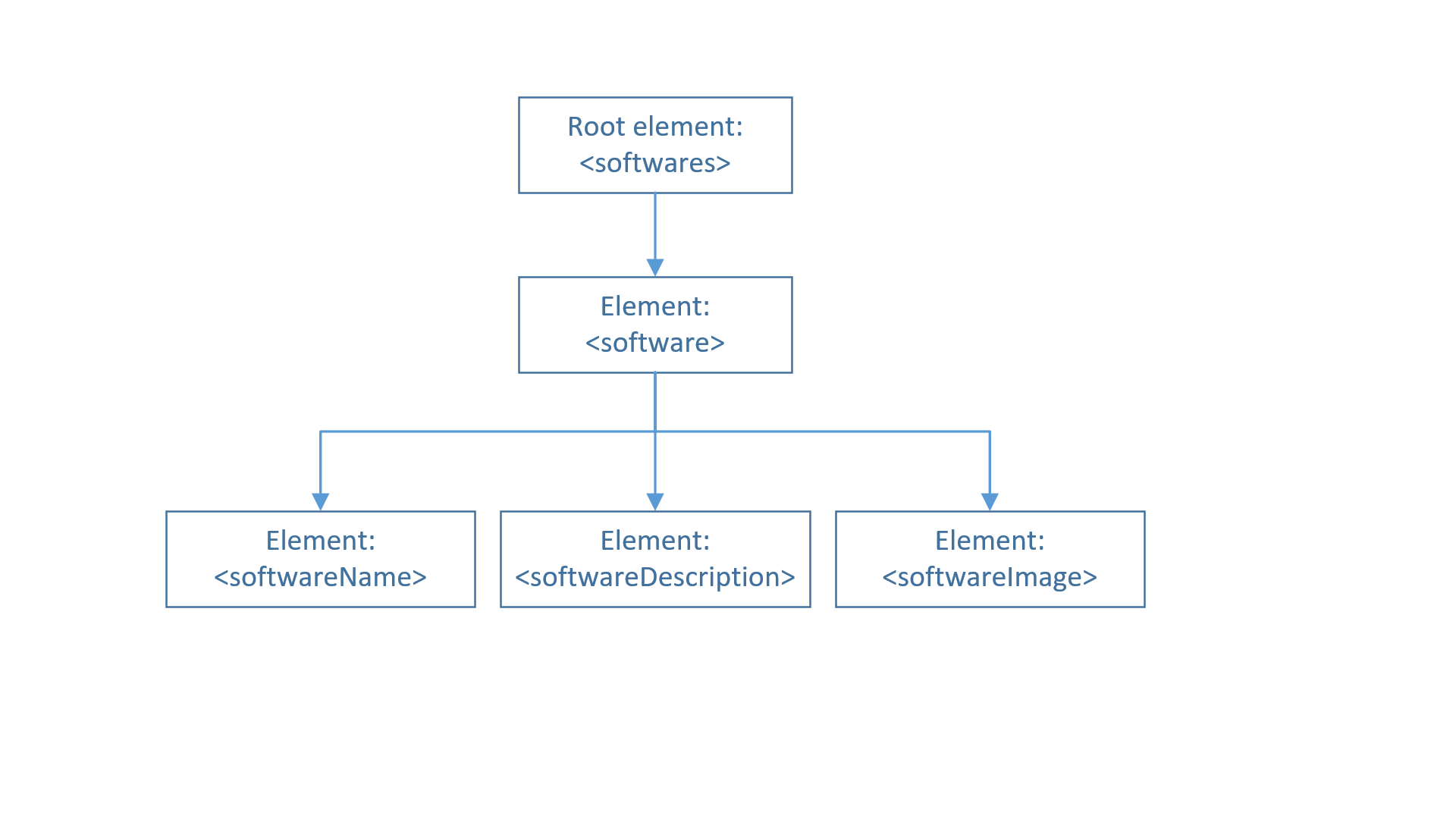
The ‘category.xml’ is used to store category information and used by other pages to display the available product categories. When Admin is adding or updating a product, they have to choose a category from the dropdown list which fetches content from this XML file.

### 3.6.6 Ad Rotator



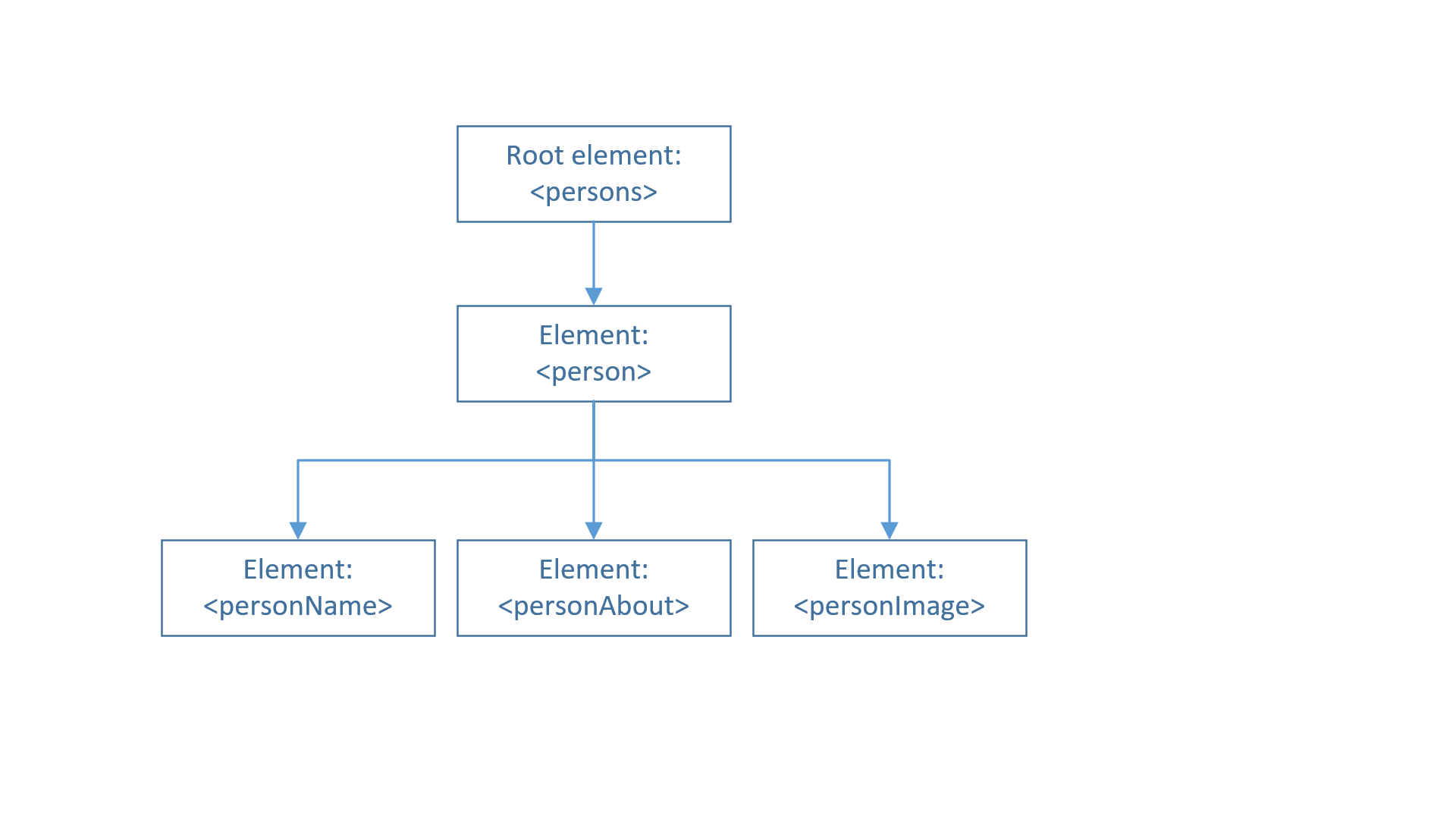
The Ad Rotator on the homepage will read the XML file that contains the above structure. The Ad Rotator will display a sequence of ad images as stored in the XML.

### 3.6.7 Homepage – Top Products



This project uses XML to display static data on the homepage. This XML file stores short description and image of the best-selling products from SoftwareBiz, which is displayed at the bottom of the homepage.

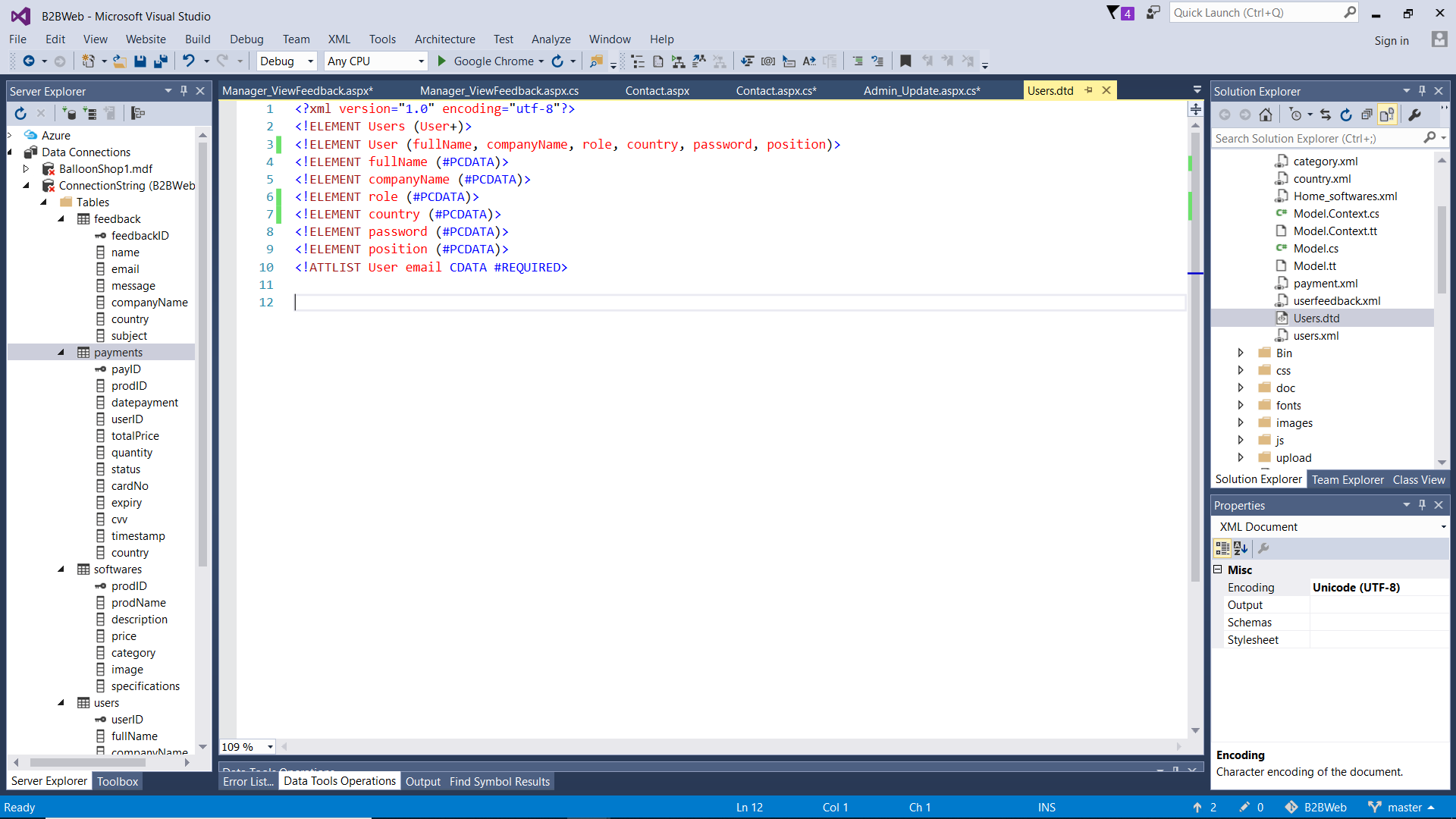
### 3.6.8 About Us - Team



On the About Us page, XML is used to display static data containing information about the team/employees of SoftwareBiz.

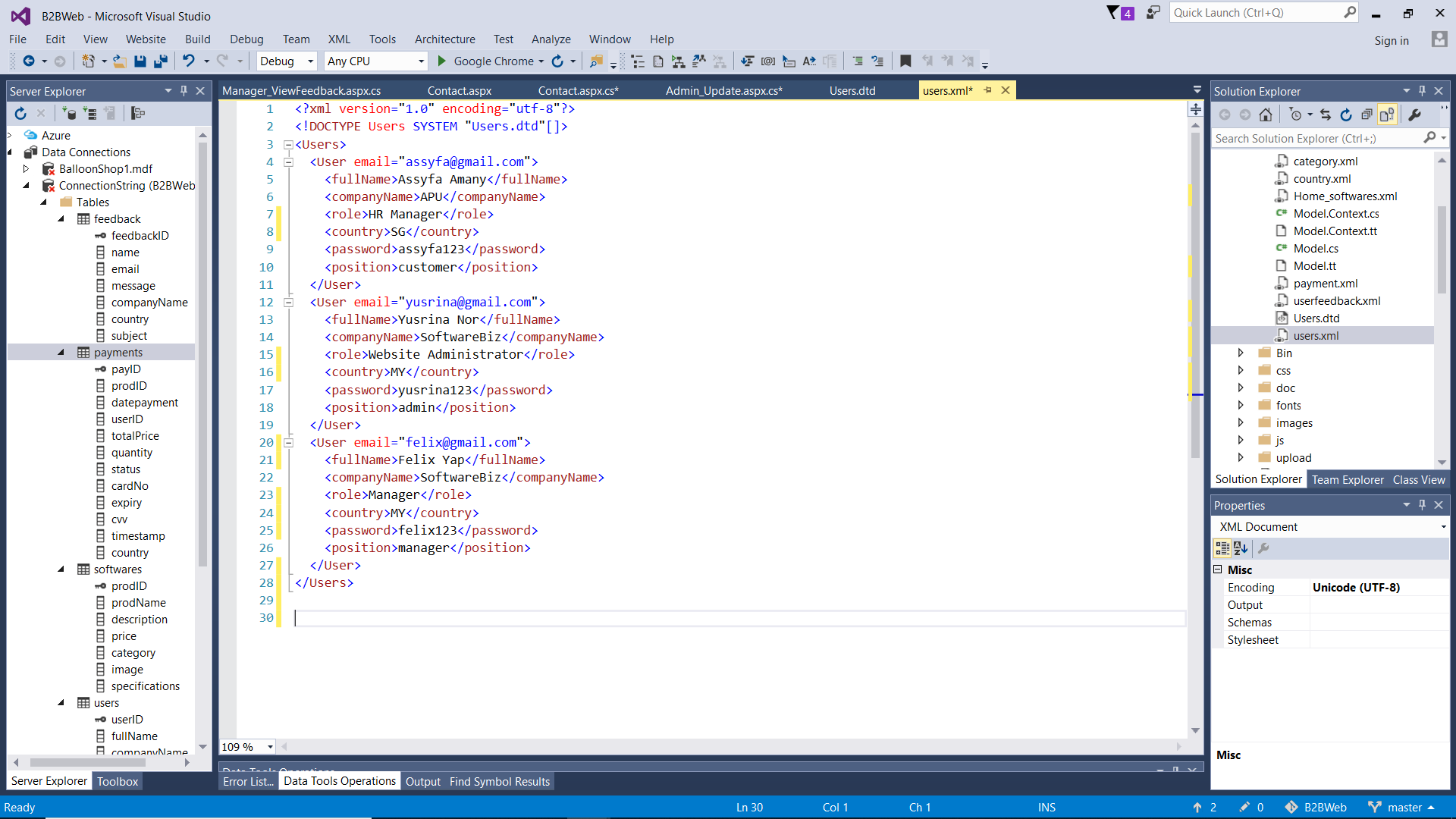
# 4.0 Implementation

## 4.1 Register & Login (with XML)



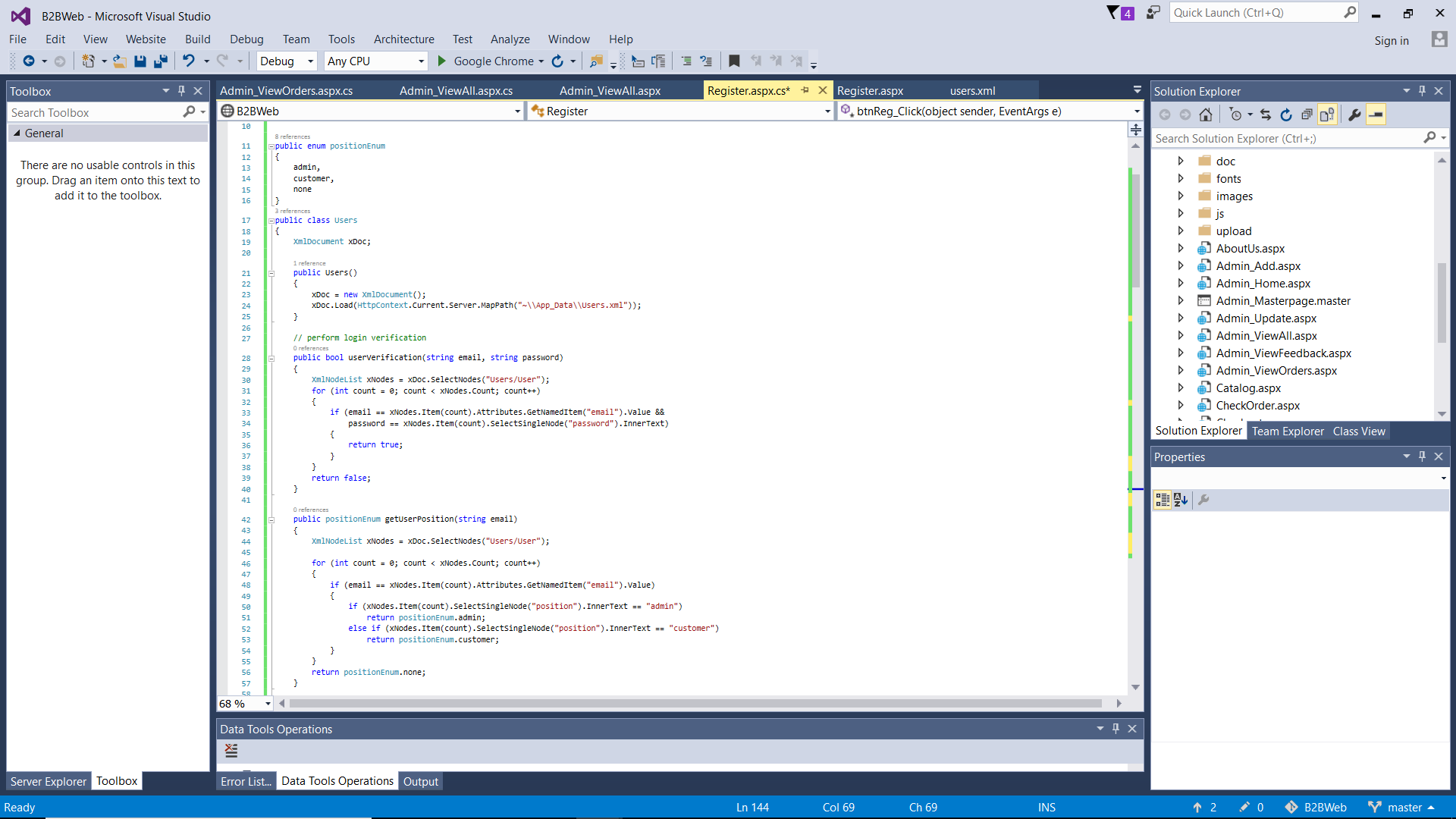
*Figure 4.1.1: users.dtd file for users.xml*

This is the Document Type Definition (DTD) meant to be used by ‘users.xml’. It defines the structure, constraints, and legal elements and attributes that must be followed by the XML file that refers to this DTD (w3schools, 2015). This means the XML document is validated against this DTD to ensure it is well-formed and valid. In this case, the DTD is saved in a separate .dtd file called ‘Users.dtd’. It defines that the root element is called <Users> which must contain 1 or more <User> element inside it. The <User> element must contain <fullName>, <companyName>, <role>, <country>, <password>, <position> elements which are marked as #PCDATA which means that it can contain any parseable text (Nikel, 2013). Each <User> element has an attribute called “email” and marked as CDATA #REQUIRED which means that it will not be parsed by a parser, hence can contain any special characters (Gerardnico, 2017).



*Figure 4.1.2: structure in users.xml*

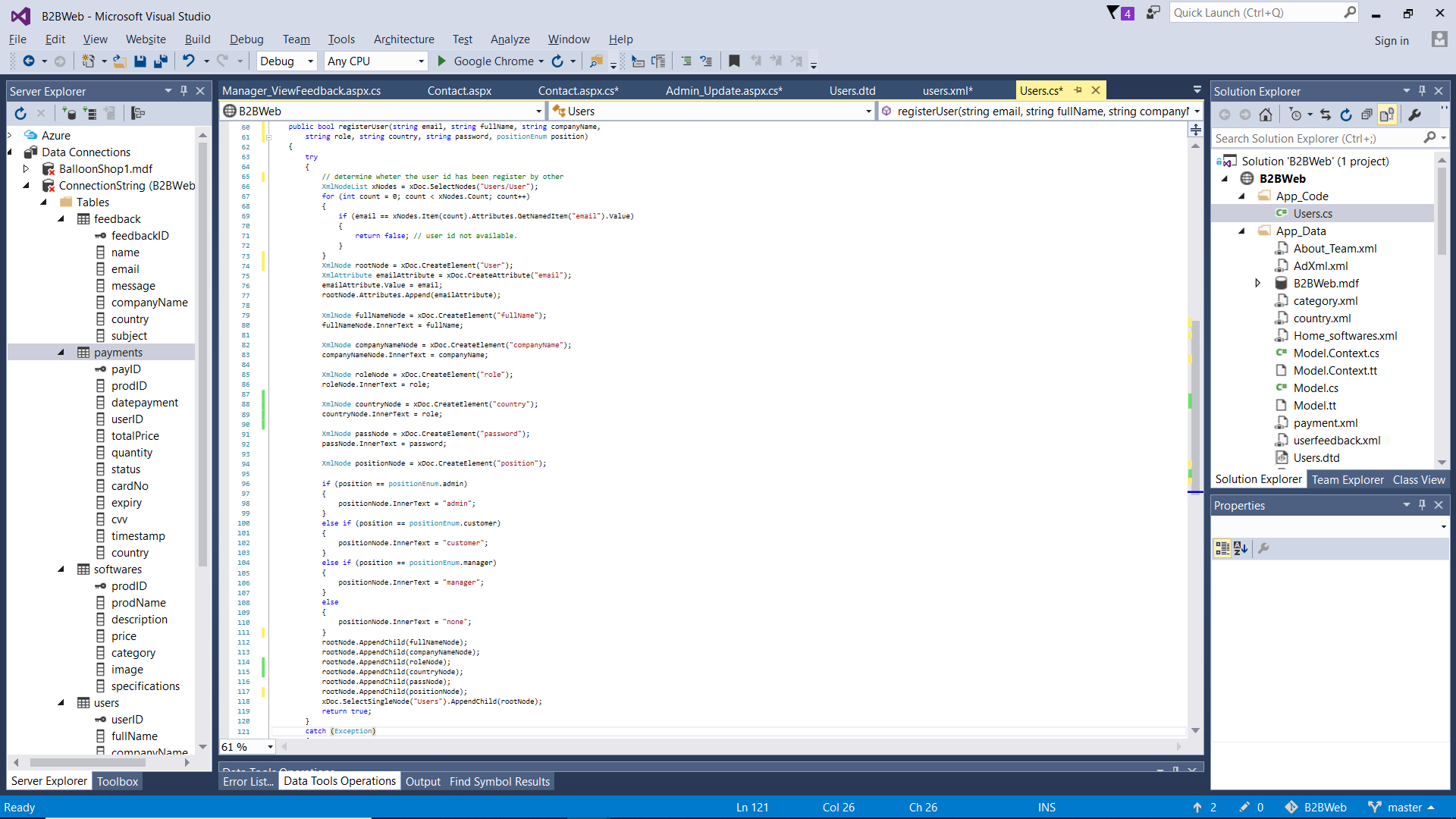
For login and registration, the website saves user data in an XML file called “users.xml”. It follows a DTD as specified in the first figure, which means the XML file must follow the DTD format.



*Figure 4.1.3: Users class for login and registration*

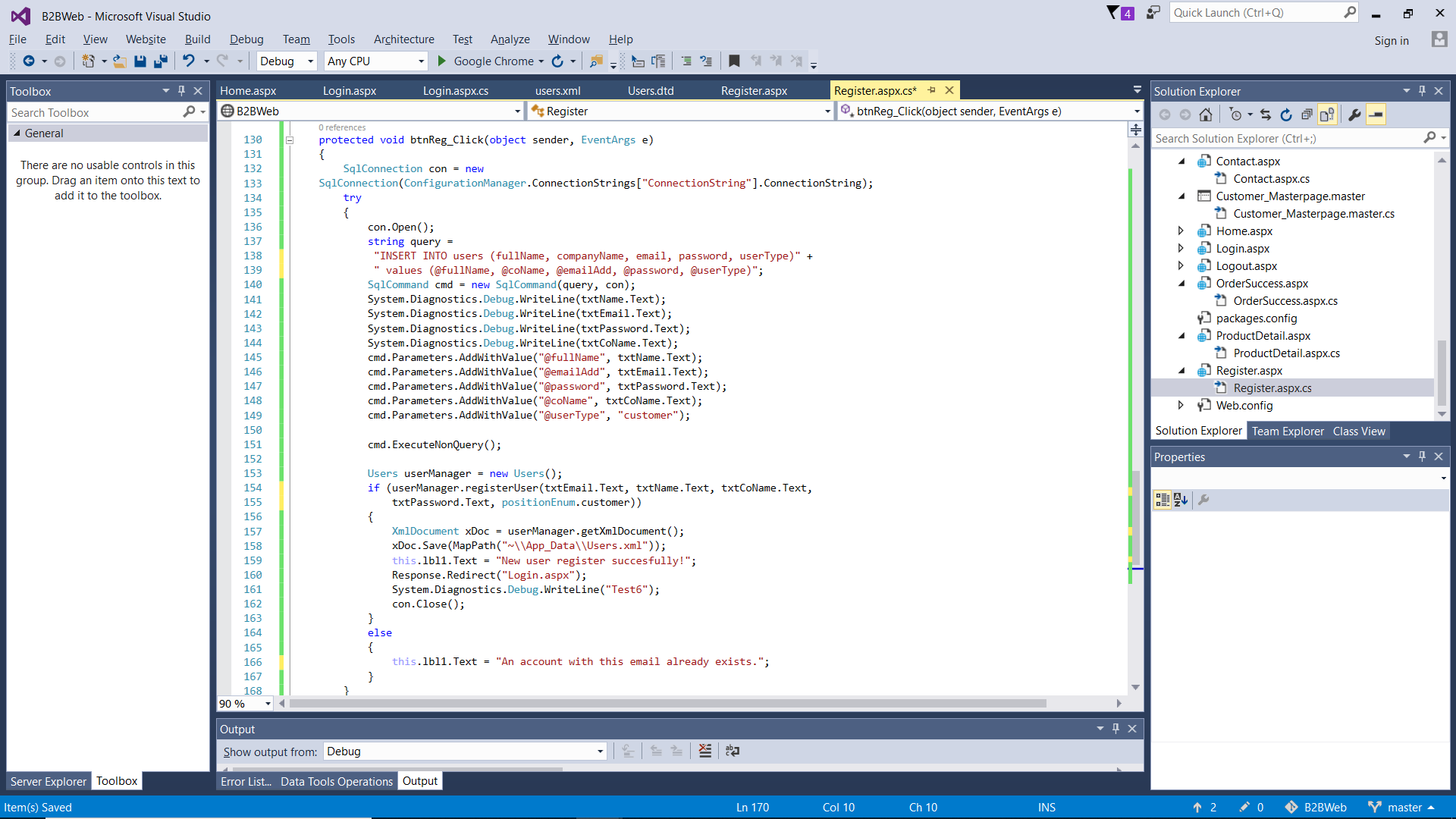
This part of the code used for verifying login credentials and getting user position, in order to redirect user to appropriate homepage suitable for their user type. Firstly, the types of users are defined under positionEnum, which are “admin”, “customer”, and “none”. Then, a class called “Users” is built to contain all the logic used for verification and user registration against the users.xml file. Methods of this class will be called when user clicks the Login button.

The first method, “userVerification”, checks if the email and password that user entered match any pair of <email> and <password> elements under a <User> element in the XML file. The “getUserPosition” method checks the <position> element of the matched <User> to determine whether the user is an Admin or Customer.



*Figure 4.1.4: registerUser method inside Users class is used for registration*

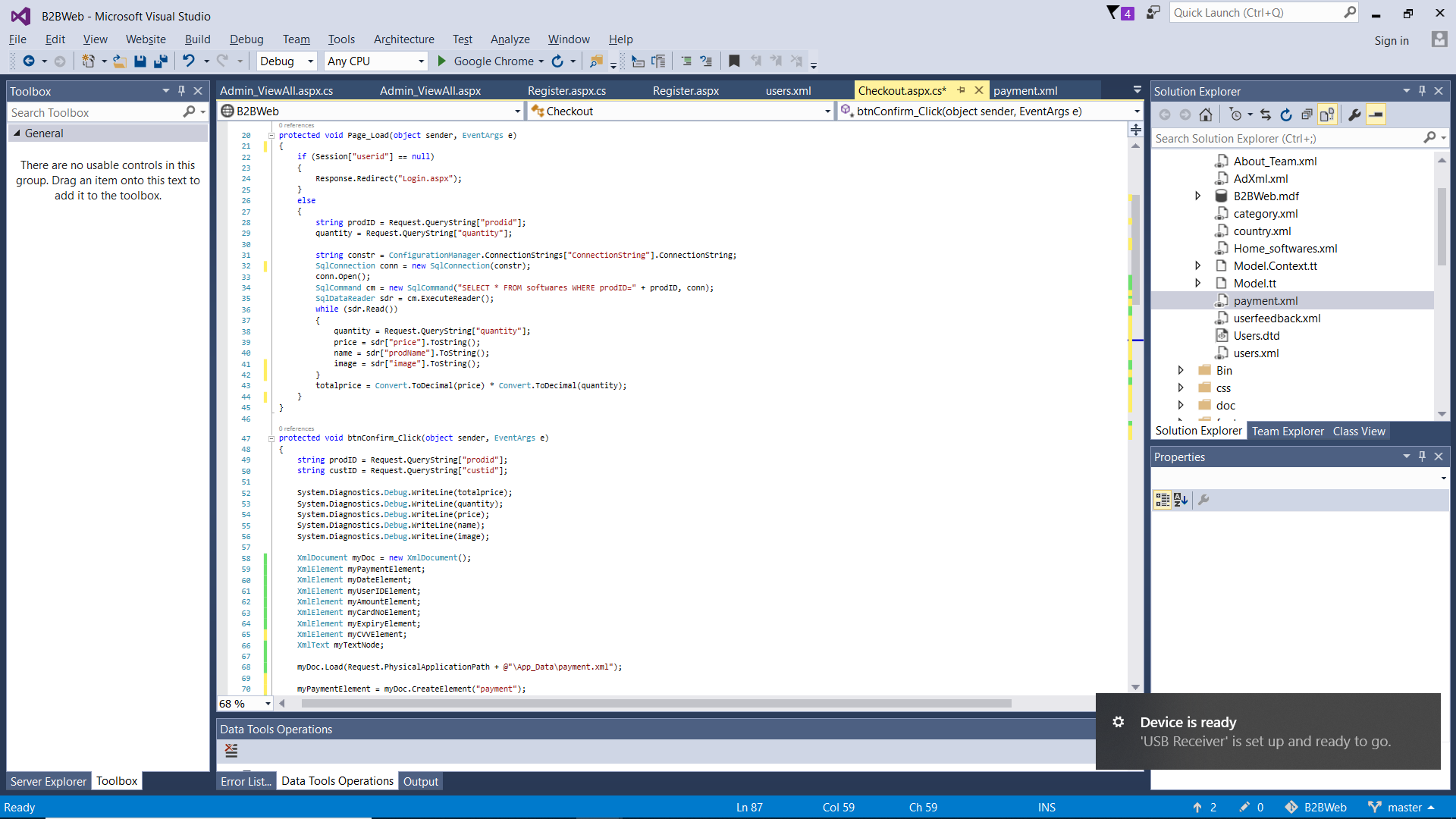
The class also has the “registerUser” method that creates a new user account by adding new User element into the XML document and populate it with content that user entered in the registration form. Firstly, it will check if the account has already existed by checking if the entered email matches the <email> element of any of the <User>. If the email does not exist in the XML file, it will add a new <User> element and populate it with the data that user entered. This is done by assigning the content into an XML Node, and then appending that Node into the rootNode one by one, following the format specified in the DTD.



*Figure 4.1.5: code to enter new user into the database*

When user clicks the Register button, the system will check if a user account with the entered email address already exists by calling the ‘userRegister’ method from the Users class explained earlier and passing the required parameters for checking. If it returns true, the new <User> element will be inserted into ‘Users.xml’ and a new row is inserted into the “users” table in the database to record new user account. User will then be redirected to Login page for login. Otherwise a message will show to inform user that the account already exists.

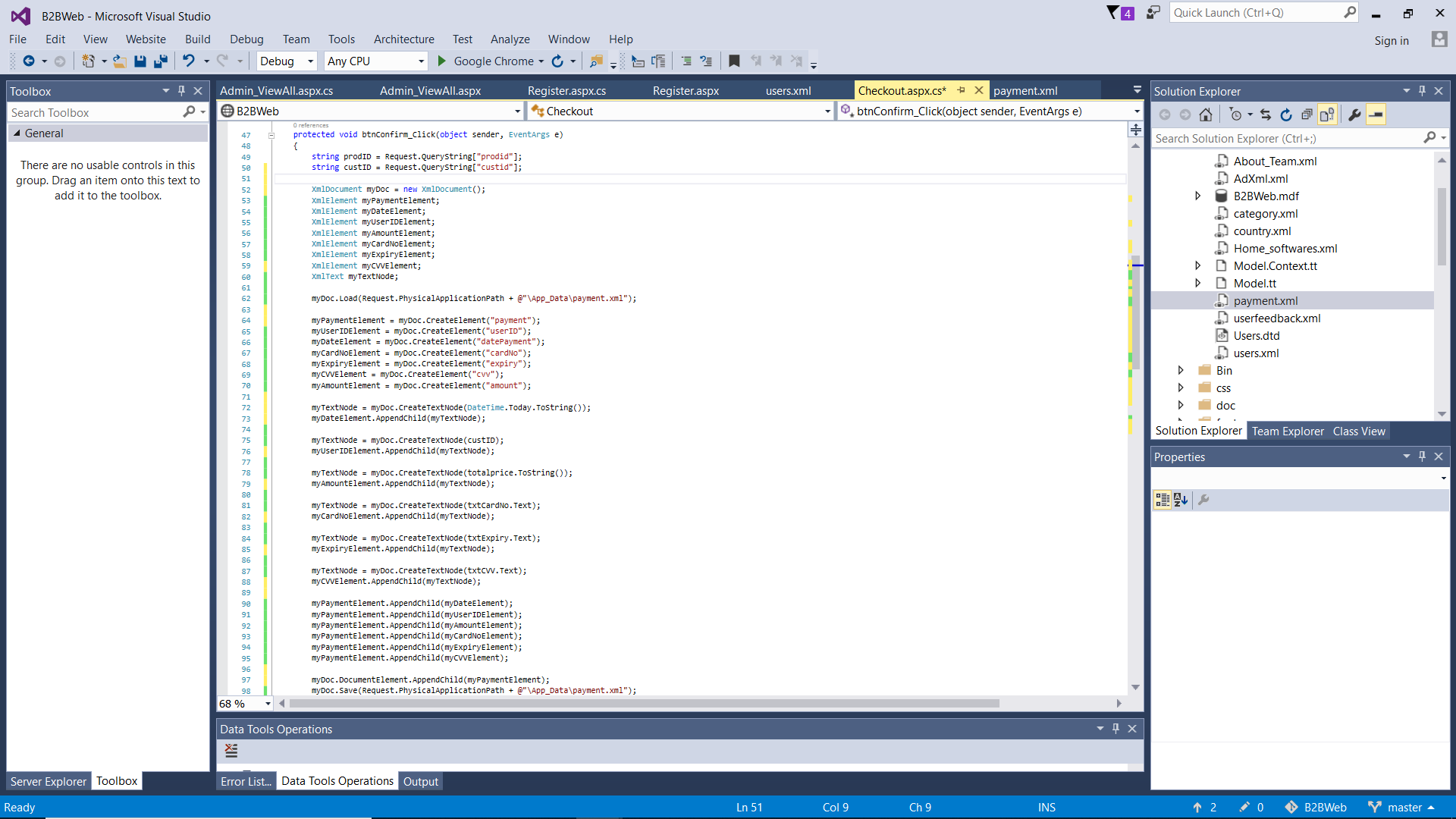
## 4.2 Checkout (with XML)



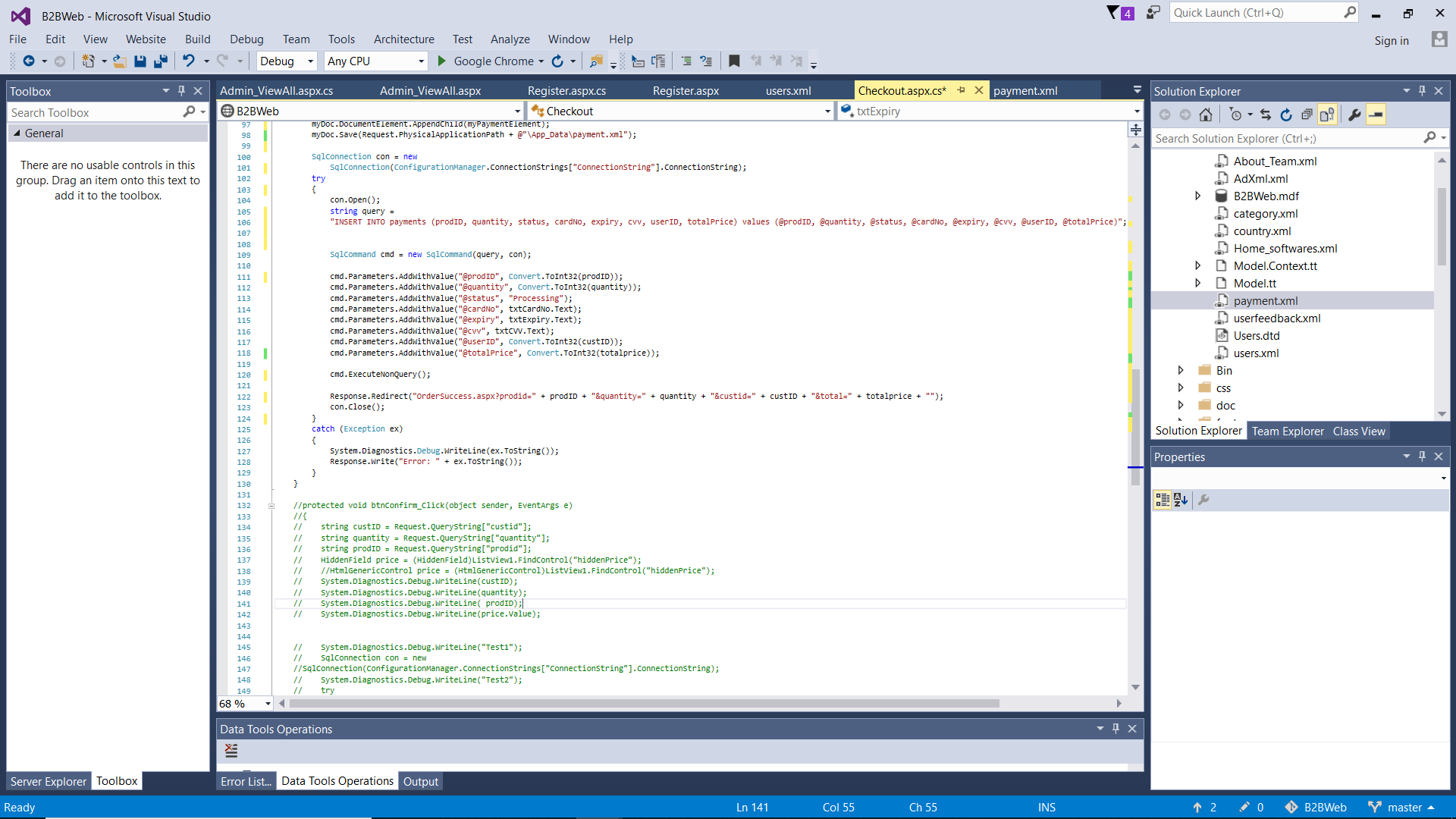
The checkout page is intended to show order summary and allow customer to enter their payment details. It cannot be accessed if user has not logged in, therefore if Session is null then user will be redirected to Login page. When the page loads, it shows order summary by taking query string of Product ID and Quantity from the previous Product Detail page, then the SQL Data Reader will read the content of the row that matches the Product ID, and the values will be displayed on the frontend using server-side script delimiters <% %> as shown. Furthermore, the backend also multiplies the product price with quantity to calculate total price of order which is shown on the frontend.



This code shows how information is displayed on the frontend. The variables defined in the backend are placed here using <% %> server-side scripting delimiters which allows for ASP scripts to be quickly referred to in the frontend (w3schools, 2010).

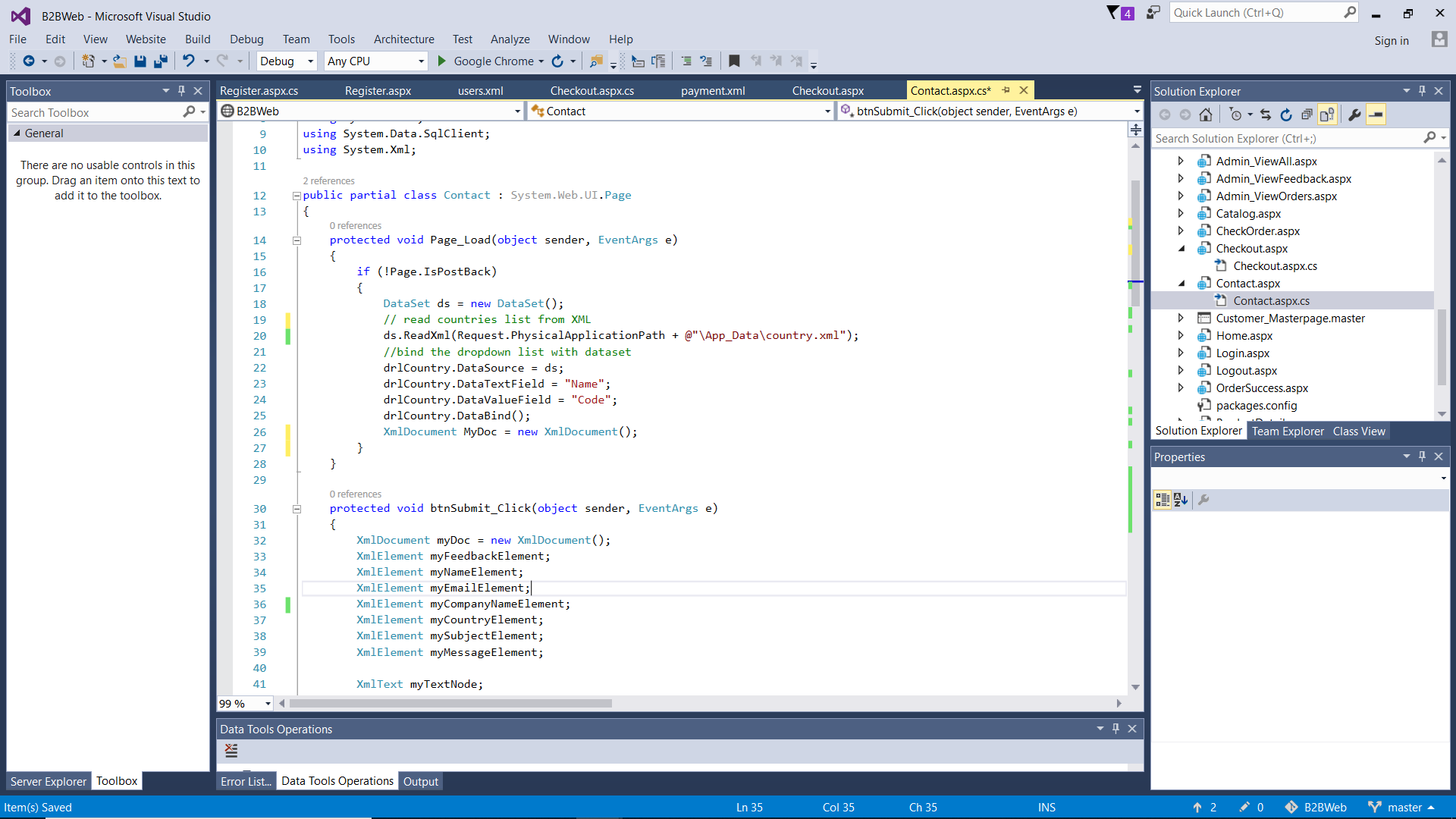


After customer has entered payment details, they are saved into XML file and database using the above code. The card details, date of payment (using DateTime function), Product ID and User ID (fetched using query string) are assigned to their respective Nodes and appended into the rootNode <payment> in the XML file to indicate a new transaction has taken place.

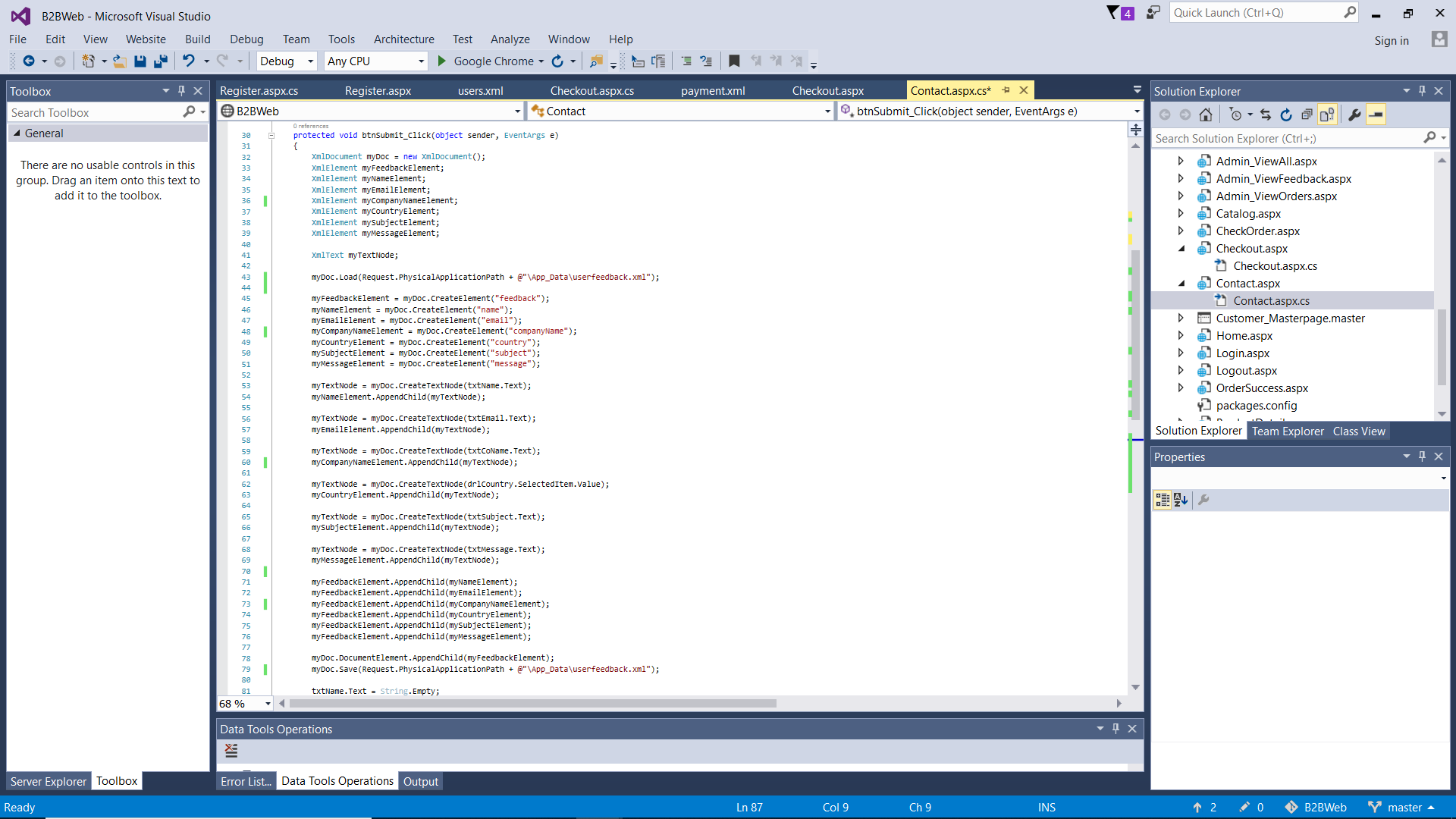


Additionally, order details are also inserted into the “payments” table in SQL database as shown in the code above using a connection string and SQL command. Afterwards, user is redirected into Order Success page with query string that carries Product ID, Quantity, and Total Price which will be used to display data in the next page.

## 4.3 Feedback (with XML)



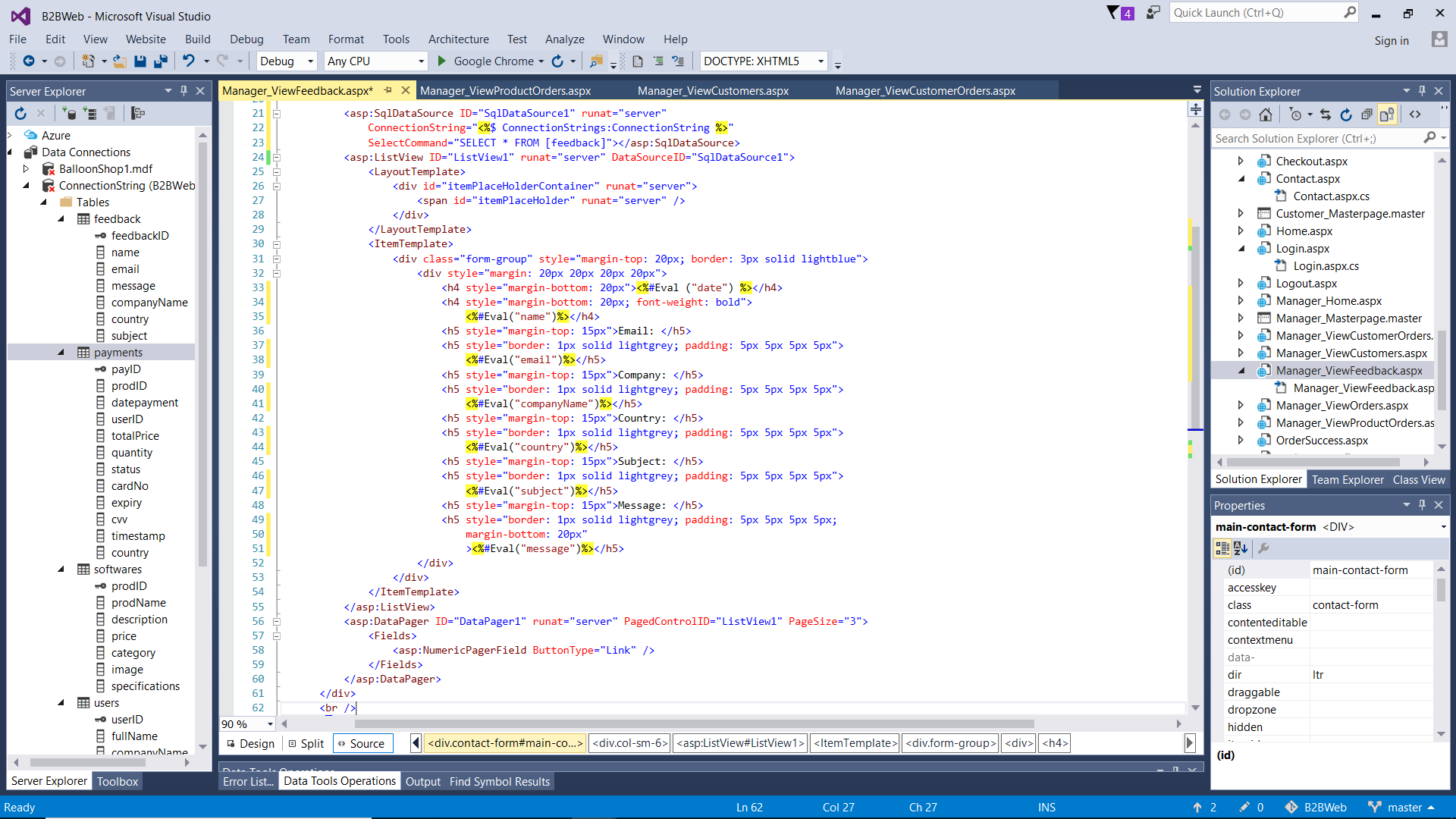
When feedback page is loaded, the content for dropdown list of countries is read from the XML file (country.xml) and bound with the dropdown list, showing the name of countries on frontend (using DataTextField) but the country code will be recorded in the backend instead (using DataValueField).



When feedback is submitted, it will be recorded in the XML file (userfeedback.xml) using the code above. The XML file is loaded, then the elements to be inserted are created and assigned a value, and then it is appended into the <feedback> root element which indicates a new feedback has been submitted, and the XML file is saved.

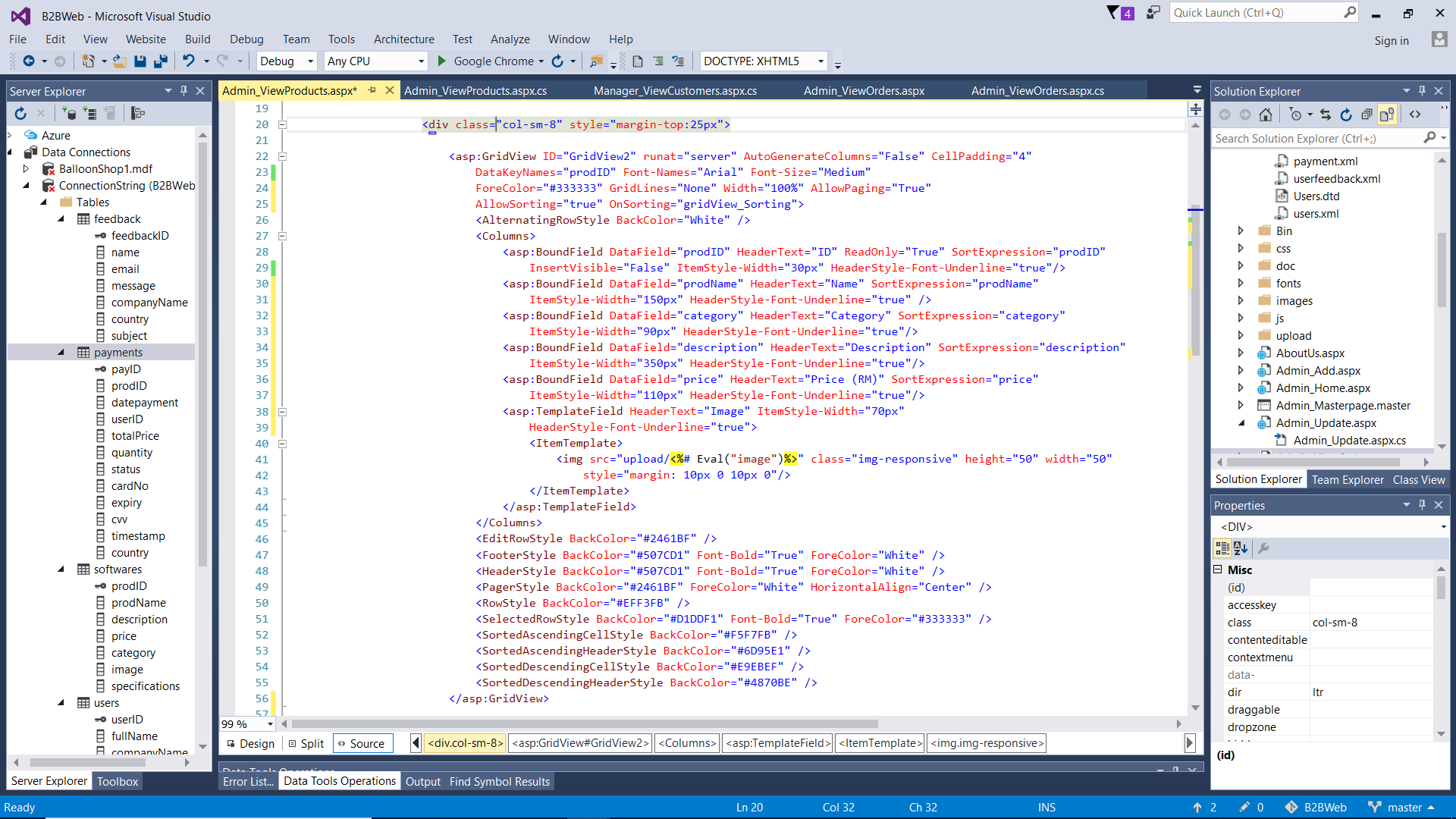


Additionally, the feedback data is inserted into the ‘feedback’ table in the database which will be used later to display submitted feedback in the frontend for the Manager to see.

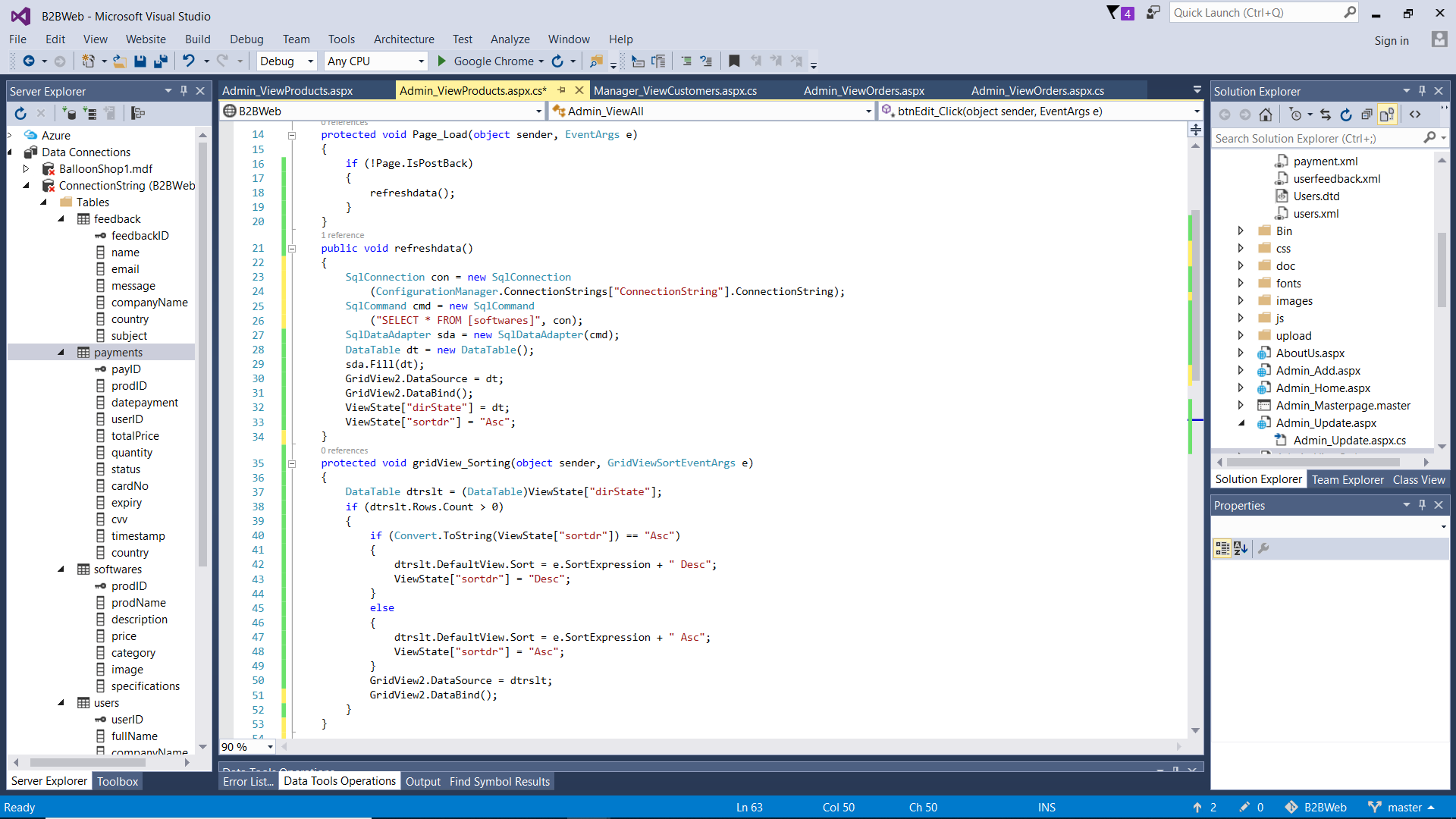


The above code shows how feedback is displayed in the frontend for Manager. Firstly, SQL data source is defined which contains the connection string and SQL command to pull the relevant data from the right table in the database. The feedback is shown in the ListView, which allows the use of ‘Eval’ to display content of a specific column in the database table. Additionally, a data pager is used which defines how many of the data-bound controls (a.k.a feedback) will be shown in a page and it displays navigation controls to go to other pages (Microsoft, 2007). In this case, it was set to 3 feedback in a page, hence to view more feedback the user must go to the next pages.

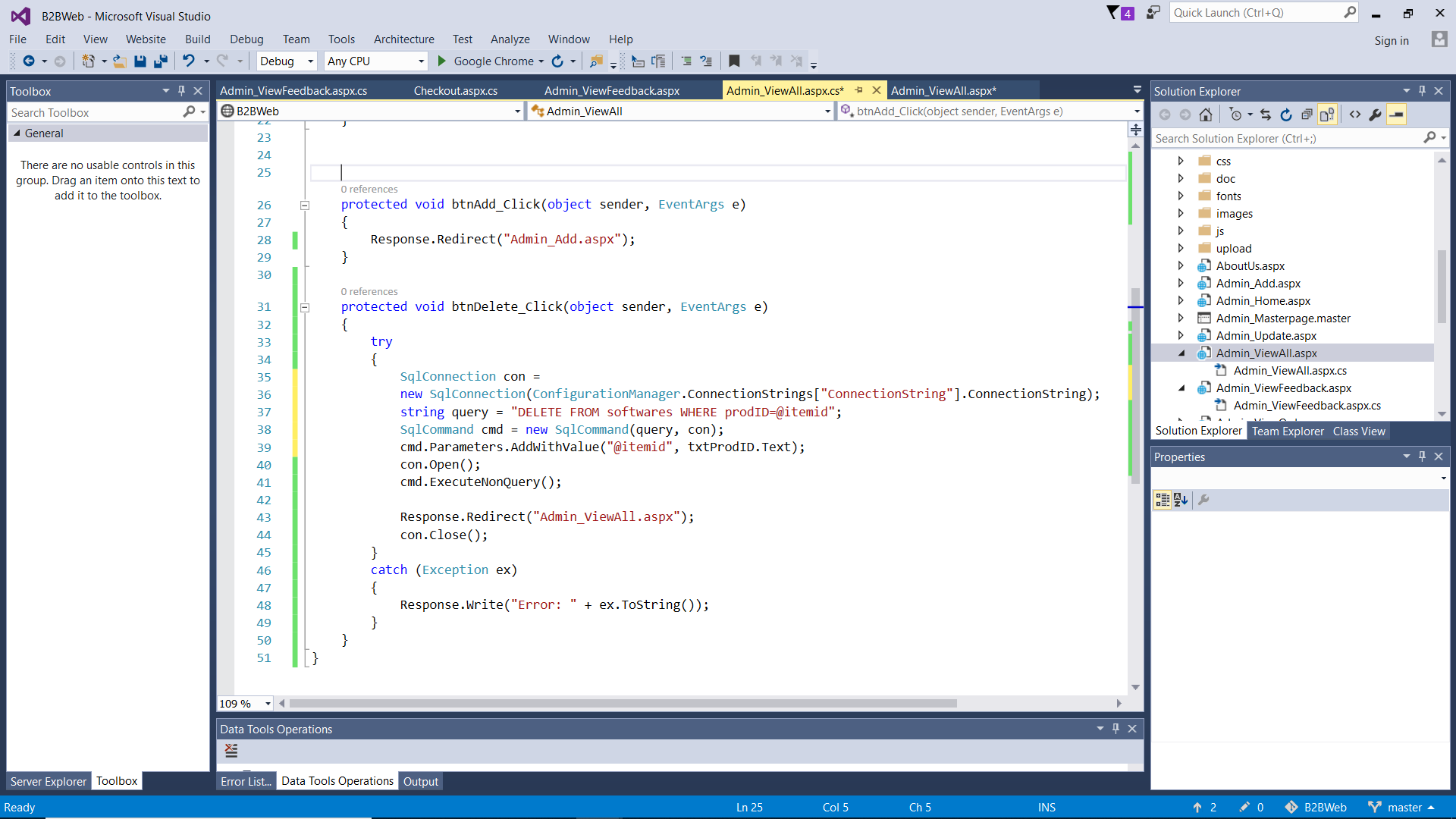
## 4.4 View & Delete Product



This is part of the frontend code for Admin to view all products. The content is shown in a sortable GridView which retrieves data from the data source and displays a list of data items by binding data fields to columns (Microsoft, 2017). The column header is also defined to identify the field and is clickable, when it is clicked the sorting will switch from ascending to descending or vice versa. Meanwhile, to display product image, a TemplateField is used in the GridView which allows the developer to insert <img> tag and specify the image URL using ‘image’ field in the database (VDWWD, 2017).

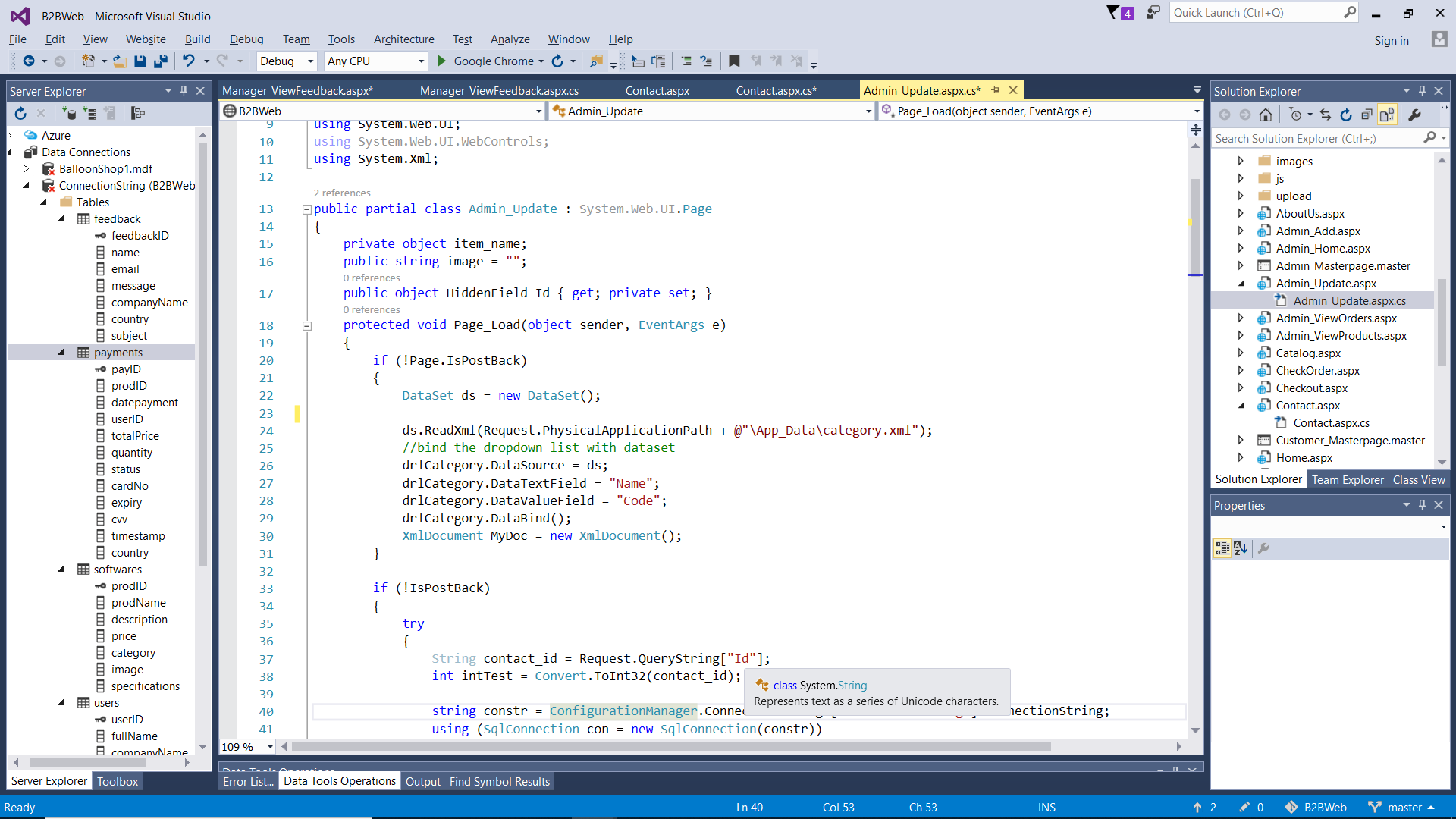


The above code is used to sort GridView items. As can be seen in the Figure 4.4.1, the <GridView> tag contains the attribute “OnSorting” which refers to the backend method called gridView\_Sorting. In this method, it determines what the current sort direction is, then switch the sort direction. The GridView is bound to the data source specified in ‘refreshdata’ method. If the page is not being loaded in response to a postback, the ‘refreshdata’ method will continuously be called. Hence, once user sorts the GridView, the list will be updated immediately.

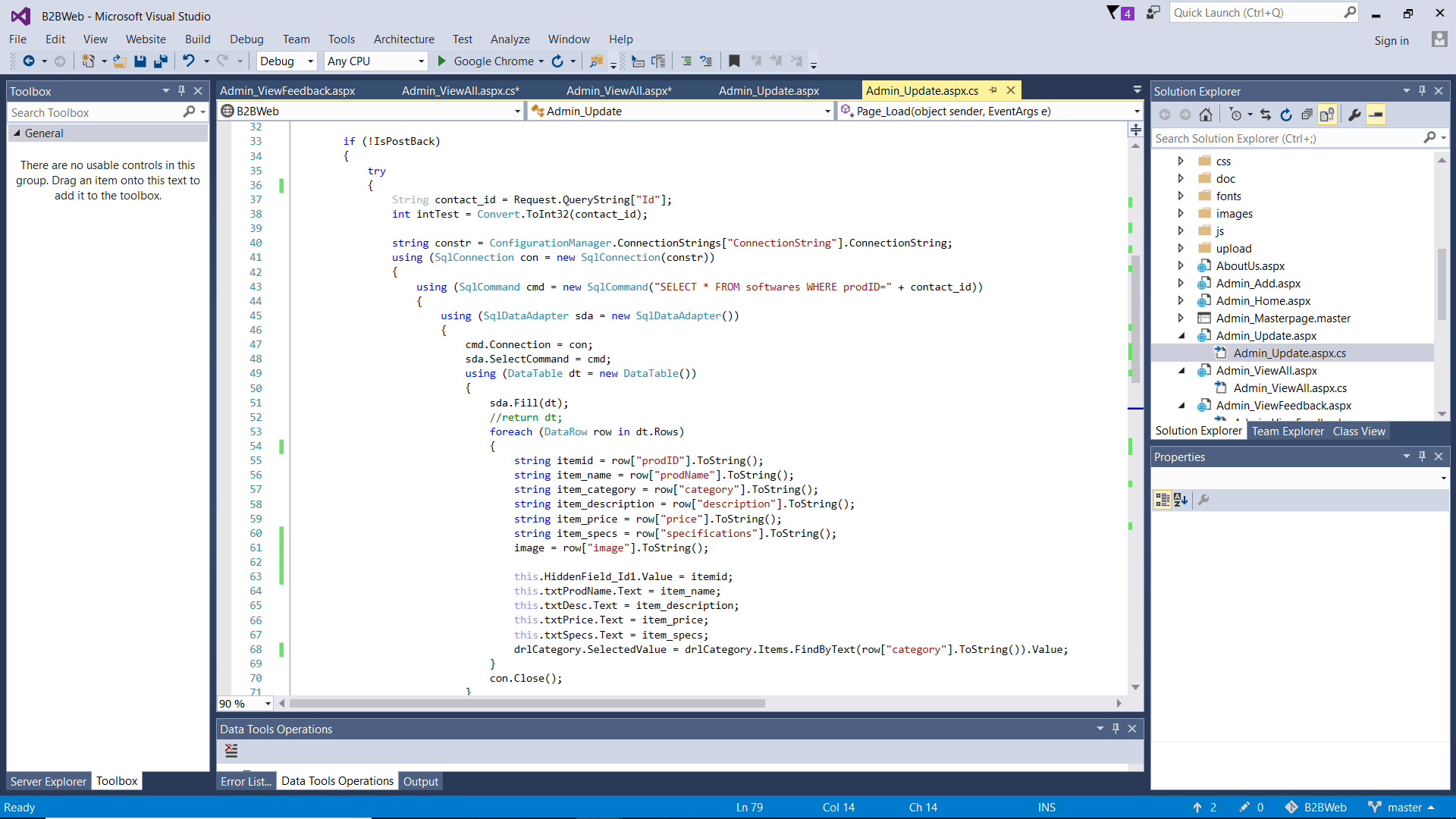


This is the code used to delete product. Firstly, user has to enter a Product ID to delete and click the delete button. Upon clicking, the backend will connect with the database and delete the entire row in the ‘softwares’ table that matches the Product ID entered. Afterwards, the page is refreshed to remove the item from the GridView.

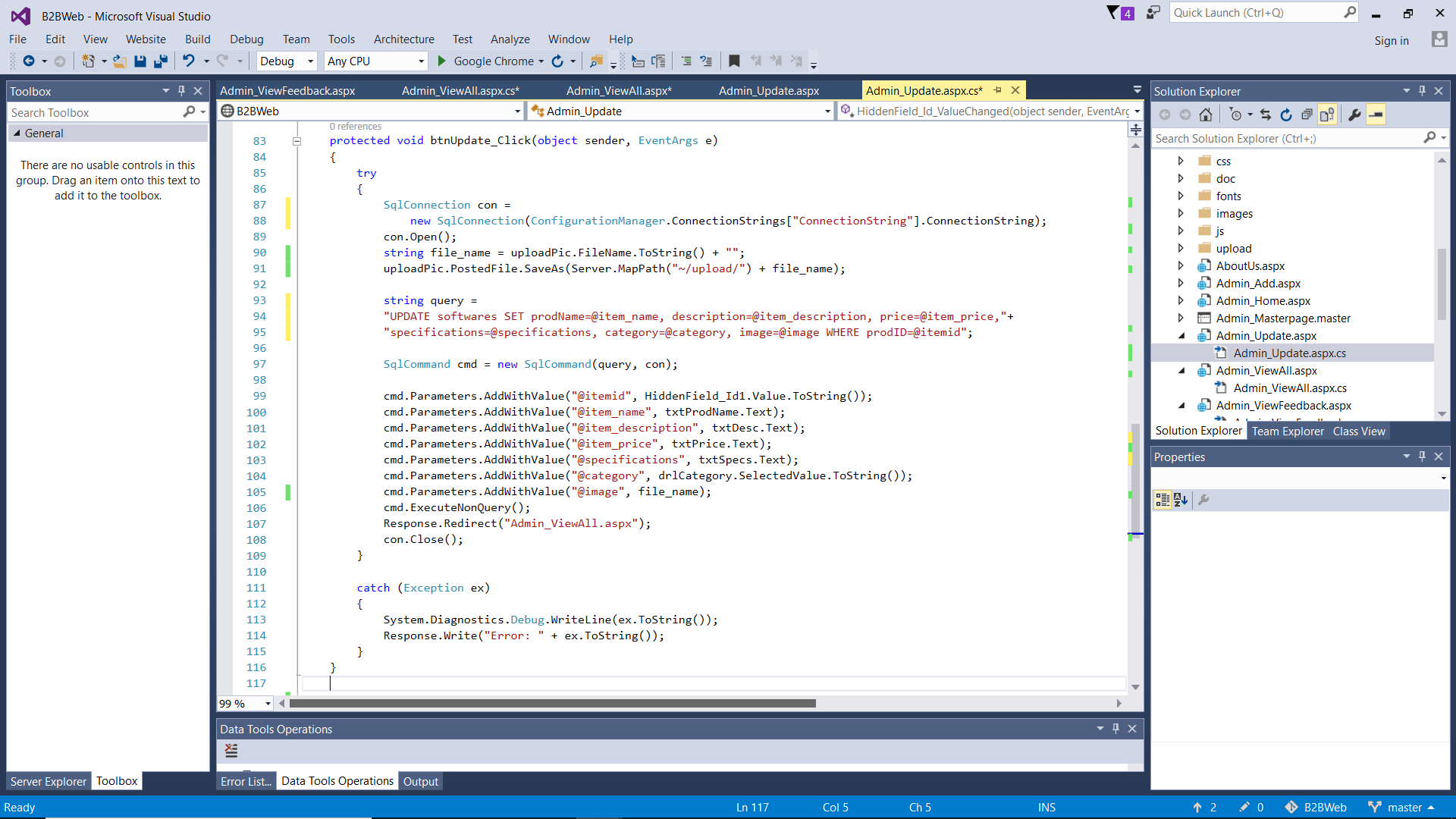
## 4.5 Update Product (with XML)



When the page loads, it contains fields where Admin can update the product details, including category. To display category options, ‘category.xml’ file is used to provide information that will be displayed in the dropdown list for categories. The code above shows how this was implemented. The DataSet.ReadXml method reads the data inside the specified XML file, and then the data is bound with the dropdown list, showing the name of category in the frontend whilst keeping the corresponding category code as the value which will be saved in the database (Microsoft, 2007).



When the page loads, it will show the details of the selected product to be edited. The selected product is defined by the Product ID entered by user in the previous page, which is passed to this page using query string. The query string is requested to be used within the SQL Command which selects a particular product from the ‘softwares’ table that matches the ID. Afterwards, SqlDataAdapter is used to retrieve data from the defined data source and populate tables within a DataSet using the Fill method which adds rows in the DataTable to match those in the data source (Godel, 2004). Hence, user can now assign a variable to contain the content of a specified row. That variable is then assigned to the textboxes shown in the frontend to display existing product details.



After Admin edits the product details, Admin clicks the Update button which will replace the current values in the database with the values entered by user in the textboxes, changing the product details as shown in the code above.

## 4.6 Test plan (test cases, pass/fail)

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Test Case | Pass | Fail |
| 1 | User can login into the system with the correct email and password |  |  |
| 2 | An error message shows when user enters wrong email or password |  |  |
| 3 | User can register a new Customer account, with the record saved into database and XML file |  |  |
| 4 | An error message shows when user enters an already registered email during registration |  |  |
| 5 | Customer can proceed to checkout page with correct quantity |  |  |
| 6 | If user has not logged in, they will be redirected to login page before they can proceed to checkout |  |  |
| 7 | If session expired, user will be redirected to login page before they can proceed to confirm order |  |  |
| 8 | If session expired, user will be redirected to login page before they can proceed to check order |  |  |
| 9 | User can logout and their session is terminated |  |  |
| 10 | User can filter the catalogue products based on category |  |  |
| 11 | User can view product details |  |  |
| 12 | User can only enter quantity in digits for checkout |  |  |
| 13 | The total price is shown correctly (price multiplied by quantity) |  |  |
| 14 | Customer can enter payment details and submit order successfully into the database and XML file |  |  |
| 15 | An error message shows if any of the required fields are left empty |  |  |
| 16 | Customer can check their orders (transaction history) |  |  |
| 17 | Only the approved orders will have a download button |  |  |
| 18 | Once the product is downloaded, the order status changes to "completed" and the product cannot be redownloaded |  |  |
| 19 | Visitor and Customer can submit feedback and it will be recorded into database and XML file |  |  |
| 20 | Admin can get an overview of the listed products in GridView |  |  |
| 21 | Admin can add a new product which will be saved into the database |  |  |
| 22 | Admin can update product details |  |  |
| 23 | Admin can delete product |  |  |
| 24 | Admin can view orders |  |  |
| 25 | Admin can approve selected order |  |  |
| 26 | Manager can view a list of customers and their details |  |  |
| 27 | Manager can view orders from a specific customer |  |  |
| 28 | Manager can view a list of orders and their details |  |  |
| 29 | Manager can view orders for a specific product |  |  |
| 30 | Manager can view all feedback |  |  |

## 4.7 Evaluation

All 3 types of users tested the website for usability and answered the following questionnaire.

### 4.7.1 Evaluation by Customer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Criteria** | | **Range (strongly disagree to strongly agree)** | | | | |
| **1** | **2** | **3** | **4** | **5** |
| **Accessibility** | | | | | | |
| **1** | The website was easy to navigate |  |  |  |  |  |
| **2** | The website was responsive in reacting to different screen sizes |  |  |  |  |  |
| **3** | The website loaded quickly |  |  |  |  |  |
| **4** | All the links worked well |  |  |  |  |  |
| **5** | It was easy to find the information I needed |  |  |  |  |  |
| **6** | I found the information I needed quickly |  |  |  |  |  |
| **Process** | | | | | | |
| **1** | The procedures were easy to understand and follow |  |  |  |  |  |
| **2** | The procedures were easy to remember |  |  |  |  |  |
| **3** | The website was able to do everything it was meant to do |  |  |  |  |  |
| **4** | I felt the various functions in this system were well integrated |  |  |  |  |  |
| **5** | The website allowed me to perform my tasks efficiently |  |  |  |  |  |
| **Design** | | | | | | |
| **1** | The page or information layout made sense |  |  |  |  |  |
| **2** | The visual website presentation was clean and pleasant |  |  |  |  |  |
| **3** | The design was consistent in terms of color, font, and structure |  |  |  |  |  |
| **Content Quality** | | | | | | |
| **1** | The content was accurate and showed latest information |  |  |  |  |  |
| **2** | The content was relevant |  |  |  |  |  |
| **3** | The content was complete |  |  |  |  |  |
| **Fault Tolerance** | | | | | | |
| **1** | The website was fault-tolerant by using validation |  |  |  |  |  |
| **2** | The error messages were clear |  |  |  |  |  |
| **3** | It was easy to find help and provide feedback |  |  |  |  |  |
| **User Satisfaction** | | | | | | |
| **1** | I felt very confident using the system |  |  |  |  |  |
| **2** | The system was easy to learn |  |  |  |  |  |
| **3** | I enjoy using the website |  |  |  |  |  |
| **4** | I feel comfortable conducting business with this website |  |  |  |  |  |

### 4.7.2 Evaluation by Admin

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Criteria** | | **Range (strongly disagree to strongly agree)** | | | | |
| **1** | **2** | **3** | **4** | **5** |
| **Accessibility** | | | | | | |
| **1** | The website was easy to navigate |  |  |  |  |  |
| **2** | The website was responsive in reacting to different screen sizes |  |  |  |  |  |
| **3** | The website loaded quickly |  |  |  |  |  |
| **4** | All the links worked well |  |  |  |  |  |
| **5** | It was easy to find the information I needed |  |  |  |  |  |
| **6** | I found the information I needed quickly |  |  |  |  |  |
| **Process** | | | | | | |
| **1** | The procedures were easy to understand and follow |  |  |  |  |  |
| **2** | The procedures were easy to remember |  |  |  |  |  |
| **3** | The website was able to do everything it was meant to do |  |  |  |  |  |
| **4** | I felt the various functions in this system were well integrated |  |  |  |  |  |
| **5** | The website allowed me to perform my tasks efficiently |  |  |  |  |  |
| **Design** | | | | | | |
| **1** | The page or information layout made sense |  |  |  |  |  |
| **2** | The visual website presentation was clean and pleasant |  |  |  |  |  |
| **3** | The design was consistent in terms of color, font, and structure |  |  |  |  |  |
| **Content Quality** | | | | | | |
| **1** | The content was accurate and showed latest information |  |  |  |  |  |
| **2** | The content was relevant |  |  |  |  |  |
| **3** | The content was complete |  |  |  |  |  |
| **Fault Tolerance** | | | | | | |
| **1** | The website was fault-tolerant by using validation |  |  |  |  |  |
| **2** | The error messages were clear |  |  |  |  |  |
| **User Satisfaction** | | | | | | |
| **1** | I felt very confident using the system |  |  |  |  |  |
| **2** | The system was easy to learn |  |  |  |  |  |
| **3** | I enjoy using the website |  |  |  |  |  |

### 4.7.3 Evaluation by Manager

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Criteria** | | **Range (strongly disagree to strongly agree)** | | | | |
| **1** | **2** | **3** | **4** | **5** |
| **Accessibility** | | | | | | |
| **1** | The website was easy to navigate |  |  |  |  |  |
| **2** | The website was responsive in reacting to different screen sizes |  |  |  |  |  |
| **3** | The website loaded quickly |  |  |  |  |  |
| **4** | All the links worked well |  |  |  |  |  |
| **5** | It was easy to find the information I needed |  |  |  |  |  |
| **6** | I found the information I needed quickly |  |  |  |  |  |
| **Process** | | | | | | |
| **1** | The procedures were easy to understand and follow |  |  |  |  |  |
| **2** | The procedures were easy to remember |  |  |  |  |  |
| **3** | The website was able to do everything it was meant to do |  |  |  |  |  |
| **4** | I felt the various functions in this system were well integrated |  |  |  |  |  |
| **5** | The website allowed me to perform my tasks efficiently |  |  |  |  |  |
| **Design** | | | | | | |
| **1** | The page or information layout made sense |  |  |  |  |  |
| **2** | The visual website presentation was clean and pleasant |  |  |  |  |  |
| **3** | The design was consistent in terms of color, font, and structure |  |  |  |  |  |
| **Content Quality** | | | | | | |
| **1** | The content was accurate and showed latest information |  |  |  |  |  |
| **2** | The content was relevant |  |  |  |  |  |
| **3** | The content was complete |  |  |  |  |  |
| **Fault Tolerance** | | | | | | |
| **1** | The website was fault-tolerant by using validation |  |  |  |  |  |
| **2** | The error messages were clear |  |  |  |  |  |
| **User Satisfaction** | | | | | | |
| **1** | I felt very confident using the system |  |  |  |  |  |
| **2** | The system was easy to learn |  |  |  |  |  |
| **3** | I enjoy using the website |  |  |  |  |  |

# 5.0 Conclusion/Critical Appraisal

In conclusion, this project was appraised as successful with a complete B2B web application that meets core user requirements being created and delivered. It has all the basic functions to conduct online B2B business activities, with XML integration that would increase the effectiveness tremendously. With this new system, partners can enjoy more ease in catalogue browsing, order, delivery, and communication processes, as a result of having a more organised system. This project has gone through the necessary system development stages, from requirements specification, to designing and modelling, then code implementation, passing all the testing and evaluation process, and now the system is ready to be released.

Based on the overall user evaluation, users seem to be quite satisfied with the design, content quality, fault tolerance aspects of the system. They found that the website presentation is well designed, clean, pleasant, and made sense. The content has good quality by being accurate, relevant, and quite complete. There is form validation in every field with clear error messages that inform users effectively, preventing faulty data from being recorded into the system. However, there are still several improvements that can be done in the future, especially in terms of accessibility and process.

According to the usability evaluation, the problems and areas of improvement for the website were identified. Customers seem to find that they are not performing their tasks efficiently enough. This may be due to the lack of search function, wishlist, and cart feature. The reason for the absence of cart feature is because managers found based on their past sales data that buyers do not usually buy more than one type of software at once, because they are companies who would purchase different software solutions from different vendors who offer the best deal. However, customers would appreciate if the mentioned features exist to help perform tasks faster, whilst some might find them essential, hence they should be implemented for future enhancements.

Other enhancements for customers include a Profile page, Remember Me feature for login, and Save Payment Details option. Due to the limited duration of the project, these additional features were not done. A Profile page would allow customer to edit their details or change password, it can also be used to collect more customer data for future CRM uses. Remember Me feature at login would make the login function faster for users, whilst with Save Payment Details customers do not have to re-enter information at checkout, hence improving task efficiency. Also, the site navigation can be improved by adding submenus to display product categories.

Furthermore, the filtering can be improved. Admin and Manager users should be given more options in filtering orders and customers. Currently, orders are filtered based on customer and product, but in the future there should be an option to filter orders based on date (to determine daily revenue), country (to determine sales trend in different countries), quantity and total amount (to identify big orders and most profitable customers). Admin should be able to filter orders based on order status, focusing on unapproved orders. Additionally, Manager should be able to filter customers based on country to understand the size of their customer base better. As for customers, they should be able to sort and filter the products in the catalogue based on price.

In the future, the system can be made capable of automatic reporting and providing insights for the Manager. For instance, there can be more calculation capability to sum up revenue based on date, customer, product, and country. Also, for faster and better feedback management in the future, Manager should view feedback in a GridView with a column that contains a checkbox to determine whether or not the issue had been handled as well as a remarks column to outline what had been done. This is to avoid confusion or duplication of tasks among different managers and to ensure that every feedback was attended to.

Overall, the project has accomplished its objectives. It will be able to help SoftwareBiz conduct their business in a much better manner which allows them to compete in the market, leading to business growth, higher customer loyalty, and more opportunities in the future.

6.0References

Afshari, M., 2009. *How to check for an Empty Gridview - Stack Overflow.* [Online]   
Available at: https://stackoverflow.com/questions/731461/how-to-check-for-an-empty-gridview  
[Accessed 23 August 2018].

Clark, R., 2012. *How to make Listview show only 1 record? - Stack Overflow.* [Online]   
Available at: https://stackoverflow.com/questions/8752179/how-to-make-listview-show-only-1-record  
[Accessed 23 August 2018].

Deshmukh, N., 2016. *What are the minimum hardware requirements to run asp.net? - Quora.* [Online]   
Available at: https://www.quora.com/What-are-the-minimum-hardware-requirements-to-run-asp-net  
[Accessed 30 August 2018].

EmailDrips, 2017. *Active Campaign - Free Trial Email Sequence - Emaildrips.* [Online]   
Available at: https://www.emaildrips.com/free-trial-email-sequence-active-campaign/  
[Accessed 23 August 2018].

FinancesOnline, 2018. *Pros and Cons of Procurify: Analysis of a Popular Procurement Management Software - Financesonline.com.* [Online]   
Available at: https://financesonline.com/pros-cons-procurify-analysis-popular-procurement-management-software/  
[Accessed 23 August 2018].

Gerardnico, 2017. *XML - Parsed Character Data (PCDATA) [Gerardnico].* [Online]   
Available at: https://gerardnico.com/markup/xml/pcdata  
[Accessed 30 August 2018].

Gerli, J., 2016. *sprout-social-logo-new | Industrial Marketer.* [Online]   
Available at: https://www.industrialmarketer.com/3-great-social-media-tools-for-business/sprout-social-logo-new/  
[Accessed 23 August 2018].

Godel, J. H., 2004. *DataAdapter In C#.* [Online]   
Available at: https://www.c-sharpcorner.com/article/dataadapter-in-C-Sharp/  
[Accessed 30 August 2018].

Griffiths, M., 2016. *Paging ASP.NET ListView Control with a DataPagerControl.* [Online]   
Available at: https://www.c-sharpcorner.com/blogs/paging-asp-net-listview-control-with-a-datapagercontrol  
[Accessed 12 September 2018].

Jackson, B., 2018. *Why I Use CallRail’s Call Tracking Software – CallRail Review.* [Online]   
Available at: https://woorkup.com/use-callrails-call-tracking-software-callrail-review/  
[Accessed 20 August 2018].

Jadav, N., 2015. *GridView Sorting in ASP.Net Using C#.* [Online]   
Available at: https://www.c-sharpcorner.com/UploadFile/009464/gridview-sorting-example-in-Asp-Net-using-C-Sharp/  
[Accessed 12 September 2018].

Khan, M., 2014. *Display (Show) data in ListView from database in ASP.Net using C# and VB.Net.* [Online]   
Available at: https://www.aspsnippets.com/Articles/Display-Show-data-in-ListView-from-database-in-ASPNet-using-C-and-VBNet.aspx  
[Accessed 22 August 2018].

Mein, J., 2009. *ASP.NET email validator regex - Stack Overflow.* [Online]   
Available at: https://stackoverflow.com/questions/1710505/asp-net-email-validator-regex  
[Accessed 19 August 2018].

Microsoft, 2007. *DataPager Class (System.Web.UI.WebControls) | Microsoft Docs.* [Online]   
Available at: https://docs.microsoft.com/en-us/dotnet/api/system.web.ui.webcontrols.datapager?redirectedfrom=MSDN&view=netframework-4.7.2  
[Accessed 8 August 2018].

Microsoft, 2007. *DataSet.ReadXml Method () | Microsoft Docs.* [Online]   
Available at: https://docs.microsoft.com/en-us/dotnet/api/system.data.dataset.readxml?view=netframework-4.7.2  
[Accessed 30 August 2018].

Microsoft, 2017. *GridView Overview | Microsoft Docs.* [Online]   
Available at: https://docs.microsoft.com/en-us/dotnet/framework/wpf/controls/gridview-overview  
[Accessed 30 August 2018].

Nikel, 2013. *PCDATA vs CDATA in XML DTD - StackOverflow.* [Online]   
Available at: https://stackoverflow.com/questions/20474113/pcdata-vs-cdata-in-xml-dtd  
[Accessed 30 August 2018].

SalesX, 2018. *X-Partners.* [Online]   
Available at: http://www.salesx.com/xpartners/  
[Accessed 12 August 2018].

Selmani, H., 2012. *Paging ASP.NET ListView using DataPager without using DataSource Control - DZone Web Dev.* [Online]   
Available at: https://dzone.com/articles/paging-listview-using  
[Accessed 30 August 2018].

Soofos, 2018. *ActiveCampaign Site Tracking - Online Cursus - Soofos.* [Online]   
Available at: https://soofos.nl/cursus/site-en-event-tracking-met-activecampaign/  
[Accessed 23 August 2018].

VDWWD, 2017. *c# - asp.net GridView ImageField set DataImageUrlField result of linq join - Stack Overflow.* [Online]   
Available at: https://stackoverflow.com/questions/45795114/asp-net-gridview-imagefield-set-dataimageurlfield-result-of-linq-join  
[Accessed 30 August 2018].

Vikrant, 2016. *Get current date only in C# - Stack Overflow.* [Online]   
Available at: https://stackoverflow.com/questions/6817266/get-current-date-only-in-c-sharp  
[Accessed 30 August 2018].

w3schools, 2010. *ASP Quick Reference.* [Online]   
Available at: https://www.w3schools.com/asp/asp\_quickref.asp  
[Accessed 30 August 2018].

w3schools, 2015. *DTD Tutorial.* [Online]   
Available at: https://www.w3schools.com/xml/xml\_dtd\_intro.asp  
[Accessed 12 September 2018].

Zakas, N. C., 2009. *JavaScript stack overflow error - Human Who Codes.* [Online]   
Available at: https://humanwhocodes.com/blog/2009/05/19/javascript-stack-overflow-error/  
[Accessed 30 August 2018].