2016

E-Commerce Project

JOHN DWYER

S00130070

Table of Contents

Chapter 1 Overview

As part of 4th year at Institute of Technology, Sligo you are required to create an individual project. This project can be anything and the concept I choose for this project to build an e-commerce website that would allow users to purchase products. The concept for this project came from my own shopping habits and how little I actually purchase from local stores. With the convenience and variety that comes with purchasing products online more and more people are doing so. This has led to many local business going online as to reach a large number of prospective customers

This report will show the technologies behind the e-commerce site. As well as the research that was required to learn how to use these technologies. It will also cover any issues that I encountered throughout the project, how the project was but together and the end result of the project.

Chapter 2 Goal and Scope

2.1 Goal

The ecommerce Project aims to be a website that delivers an easy way for the user to purchase items that they need. Ideally the System will have a simplistic and comprehensive UI that is easy to understand so that all ages will be able to find and purchase the products they require without any hassle. The project will allow for the owner of the website to have administrator authority to create, edit and delete new products with ease. Of course as with all shopping websites all sensitive data such as communication over the website network will be secure.

2.2 Scope

The ecommerce project is a website that provides all required functionality for the customers and administrator of the site. The main part of the project is the way the customers are able to look through the product catalogue and filter by category to find the Item they require. Following this the next area of importance is for the customer to be able to add an item to the cart. Once the customer has added an item to the cart they will now be able to view there cart so that they can decide if they have everything that's is important and remove any unwanted items. Next the customer will enter there shipping details before finishing their purchase. Once an item has been added to the cart the cost will be calculated and displayed at a glance in the navbar. As items are added to the cart the total will be recalculated and displayed for when the customer finally wishes to complete their transaction that they know how much it will cost.

On the administrator side they will be able to create edit and delete products. When creating a new product the administrator will specify the products name, category, type, brand, colour, description and the quantity that is in stock. The administrator can also upload an image of the product if he so chooses.

Chapter 3 Areas of Research

3.1 E- Commerce

The first area of research I look into was examining pre-existing e-commerce websites. I looked at these sites as these are popular commercial websites that have had their kinks worked out and are a good basis to build a site on. The sites I visited were amazon (amazon, 2016), skate hut (skatehut, 2016), River Island (riverisland, 2016) from studying these sites I learnt that I must try keep the overall design as simplistic as possible as to make it appeal to all age groups. I also learnt that navigation must be simple and always easily viewable and accessible. I learnt that constant feedback is a must with the previously mentioned sites as amazon constantly informs the customer how many items are in the cart (shown in fig.1).

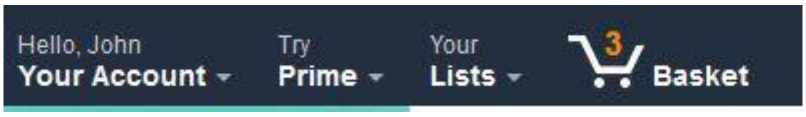


Fig.1 Amazon navbar (amazon, 2016)

While River Island chooses to inform the customer of the overall cost of their cart (shown in fig.2).

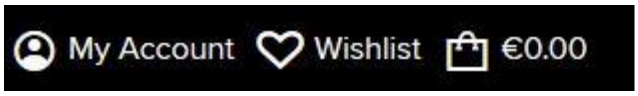


Fig.2 River Island navbar (riverisland, 2016)

From looking at river Island that a clean interface is a must so that the user can focus on the products they want and the importance of displaying just enough information beside the product but not to much as to distract them from the actual item itself. They do this by just displaying an image of the item itself with a short description and the price (shown in fig.3).

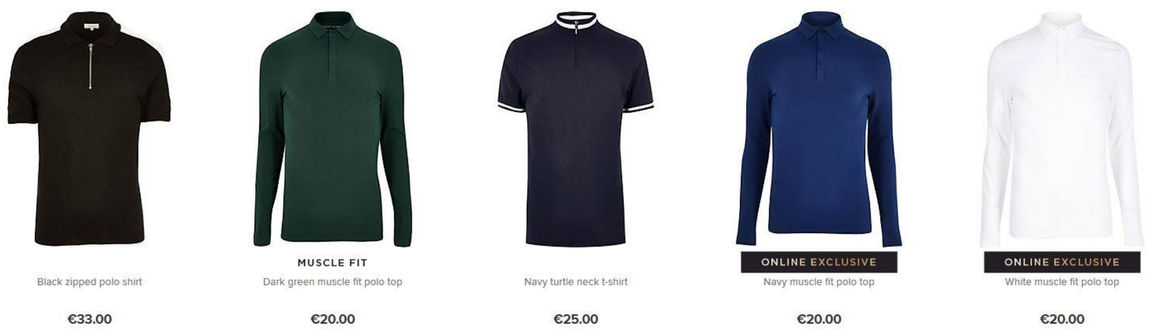


Fig.3 River Island navbar (riverisland, 2016)

While looking at skate hut I realised a nice feature which was when the user goes onto the site they are displayed straight away with the categories they can visit. This allows the user to easily visit the categories they wish to view but also displaying what other items they could purchase from the site. The customer might not of thought they needed these items or that the site stocked them but now they are informed of what the site has to offer(shown in fig.4).

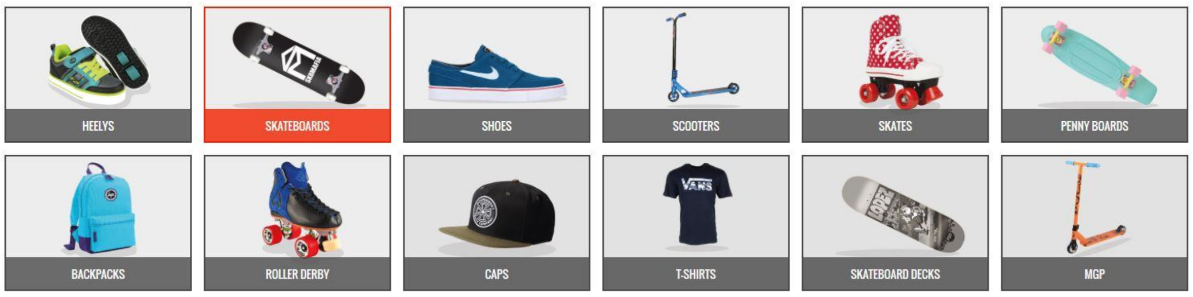
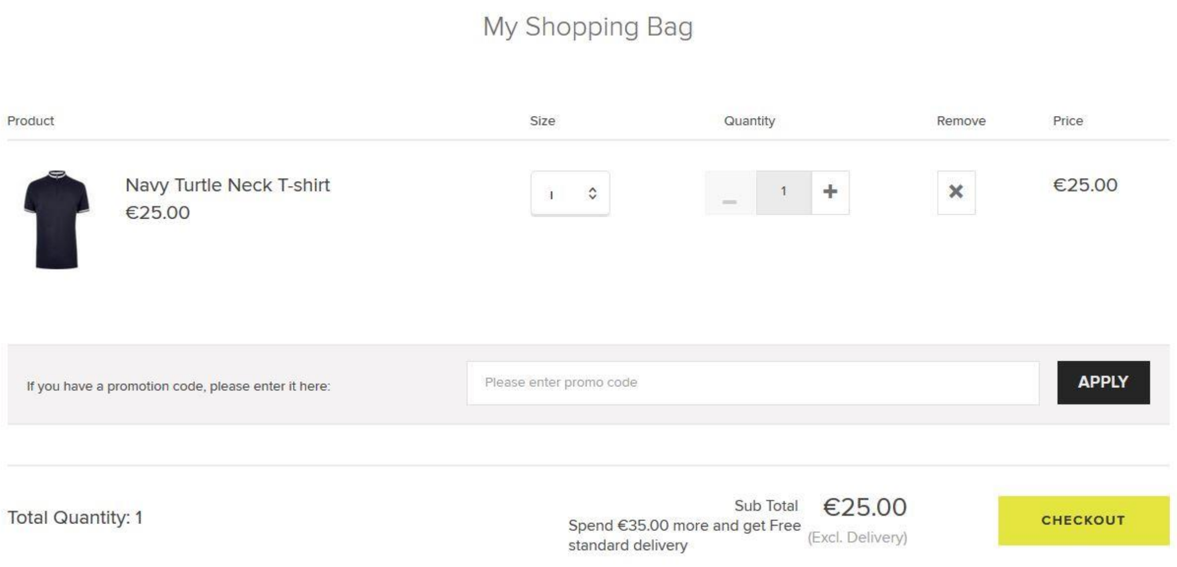


Fig.4 Skate hut catalogue (skatehut, 2016)

Another area of interest on these sites was the shopping cart which River Island kept clean and simply with just an image of the product, its name and size, the quantity they wish to purchase and the ability to remove anything they don't wish to purchase from the cart. They also kept the user informed of the price and the subtotal of all items in the basket (shown in fig.5).



My Shopping Bag

Product	Size	Quantity	Remove	Price
Navy Turtle Neck T-shirt €25.00	1	1	✖	€25.00

If you have a promotion code, please enter it here: Please enter promo code APPLY

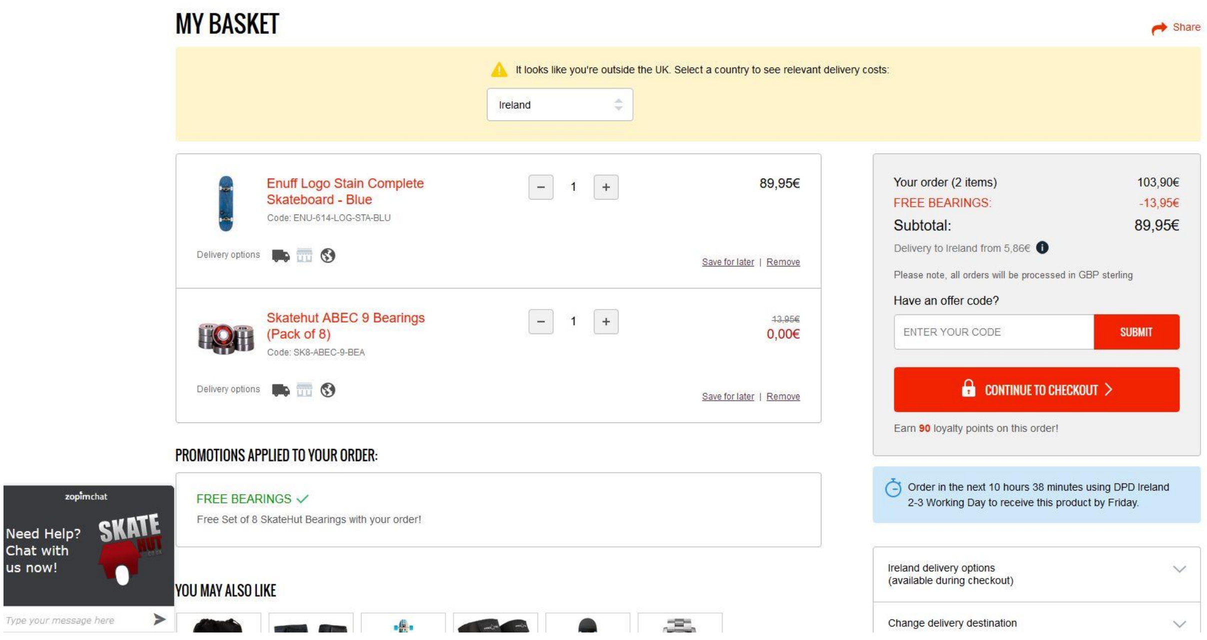
Total Quantity: 1

Sub Total €25.00
Spend €35.00 more and get Free standard delivery (Excl. Delivery)

CHECKOUT

Fig.5 River Island cart (riverisland, 2016)

While in comparison with skate huts basket you can notice it feels cluttered compared to River Island's clean design. The choice of colour, spacing and amount of things simple displayed in the cart page leads to a less satisfied user experience (shown in fig.6).



MY BASKET

Share

It looks like you're outside the UK. Select a country to see relevant delivery costs:

Ireland

Enuff Logo Stain Complete Skateboard - Blue
Code: ENU-614-LOG-STA-BLU
1
89,95€
Delivery options
Save for later | Remove

Skatehut ABEC 9 Bearings (Pack of 8)
Code: SK8-ABEC-9-BEA
1
~~13,95€~~
0,00€
Delivery options
Save for later | Remove

Your order (2 items)	103,90€
FREE BEARINGS:	-13,95€
Subtotal:	89,95€

Delivery to Ireland from 5,86€

Please note, all orders will be processed in GBP sterling

Have an offer code?

ENTER YOUR CODE SUBMIT

CONTINUE TO CHECKOUT

Earn 90 loyalty points on this order!

Order in the next 10 hours 38 minutes using DPD Ireland 2-3 Working Day to receive this product by Friday.

Ireland delivery options (available during checkout)

Change delivery destination

PROMOTIONS APPLIED TO YOUR ORDER:

FREE BEARINGS ✓
Free Set of 8 SkateHut Bearings with your order!

YOU MAY ALSO LIKE

zopimchat
Need Help? Chat with us now!
Type your message here

Fig.6 Skate hut catalogue (skatehut, 2016)

3.2 Management Methodology

For my management mythology I looked at the Waterfall and Agile Project management Methodology. I decided early on that I wouldn't be using the traditional waterfall project management as it handles things sequentially and is difficult to adjust a project as development progresses. Rather I would be taking aspects from the agile approach to project management. The aspects I took were sprints, daily stand ups and product backlog. As I was a one man team I didn't do daily stand ups in front of anyone instead I would write down what I had accomplished and what I aimed to accomplish for that particular day. At the end of every sprint I would re adjust my product backlog which I stored in a word document on my pc to try and make sure that the fundamental tasks of the project such as displaying and purchasing products would be accomplished be the end of the project.

3.3 User cases

I created use cases to help me identify the system requirements. These user cases were made up of possible sequences of interactions between the system and its users to reach their required goal. The first step was identifying who is going to be using the system which in my case is the customer and the administrator. After I identify the users I had to identify their needs I have written out these out below.

<u>Customer</u>

As a customer I want to be view all products.

As a customer I want to be able to look at categories.

As a customer I want to be able register an account.

As a customer I want to be able login an account.

As a customer I want to be able to connect my mobile to my account.

As a customer I want to be able to be able to view my account.

As a customer I want to be able to view the products in my cart.

As a customer I want to be able to purchase a product.

As a customer I want to be able to remove items from my cart.

Owner

As an owner I want to be able to create products.

As an owner I want to be able to edit products.

As an owner I want to be able to delete products.

As an owner I want to be able to upload photos of products.

3.4 Wireframes

A question many people wonder is why should you spend time doing wireframes when you could be making progress on actually creating the product. There are many reason to create these rather than rushing head first into a project as people can waste more time later on without a clear structure of what the end product should look like. At the start of a project people have a lot of creative thoughts that sound good and look nice in there head but don’t work in the real world . They can also get easily overwhelmed when actually putting the product together as they have to try and remember everything from where content will be placed, the colours and the fonts and so on. Wire frames make the creator think as a user and ask themselves what I want the user to do when they are on my site and would one design choice actually imped or cause extra hassle for the user to use.Once you have decided on what you want the user to do its time to construct the layout and placement of everything.

For building my wireframes I looked at many solutions some website based and some client based wireframe.cc, gomockingbird and justinmind. I picked justinmind over the other because its ease of use, long list of widgets and it has the ability to simulate your website and run through your design so you can quickly see missing feature and spot design flaws.

3.5 Version Control

For version control I looked at GitHub (github, 2016) and Bit bucket (bitbucket, 2016) . Bit bucket and Github are very closely matched in terms of features with both allowing you to push up multiple “versions” of a project, which show the changes that were made to the code over time, and allows you to backtrack if necessary and undo those changes. In the end I decided to go with Github because as a student I have 5 free private repository’s and I find it has a nicer interface compared to bitbucket.

3.6 Tech diagram

There was a great deal of research that went into choosing which technologies would be used in my project. In my early sprints I created a tech diagram using Astah (astah, 2016) to make sure I was organised and knew what I needed for each part of the project before I started working on my code (shown in fig.7).

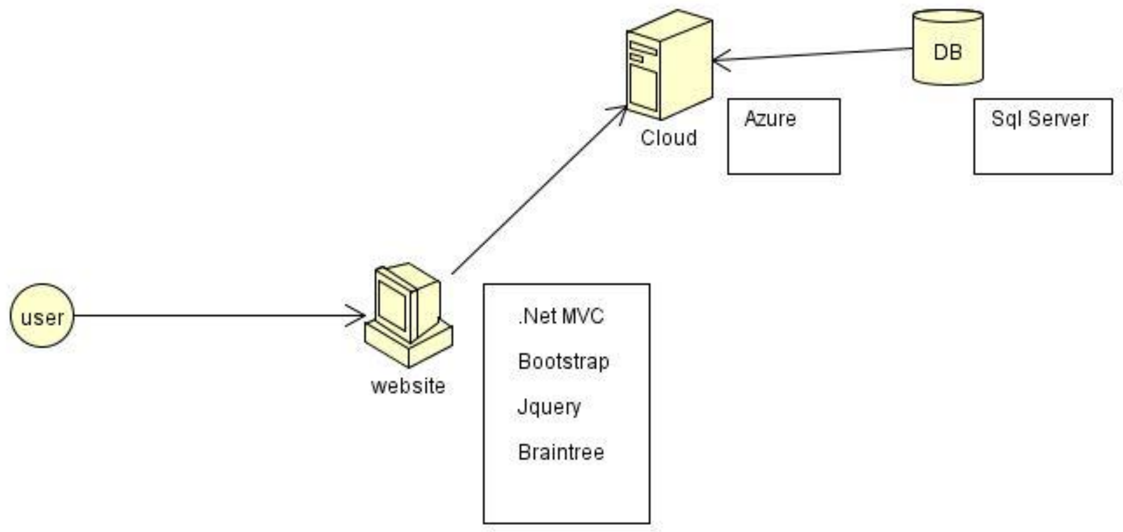


Fig.7 project start tech diagram

But over the duration of the project some of these technologies changed to deal with issues that occurred and features that I thought would be required and in the end appeared more like fig.8.

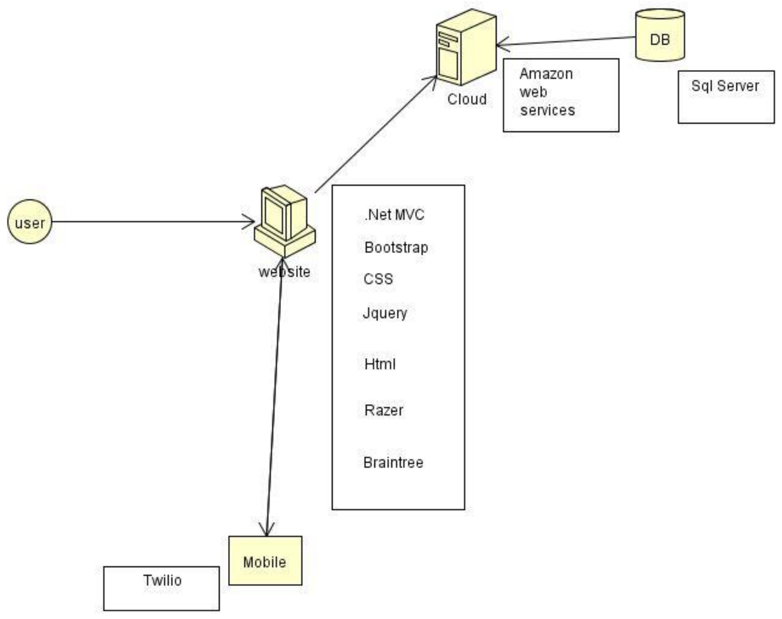


Fig.8 project end tech diagram

3.7 Framework

A framework is a supporting structure for developing code. I looked into two frameworks ASP.NET MVC and laravel. Both of these follow the Model–view–controller (MVC) a software architectural pattern with the Model representing the application core (for instance a list of database records), the View displays the data (the database records) and the Controller handling the input (to the database records). I choose to use ASP.NET MVC due to all the support it has and how well it fits with some of the other technologies I have picked to use out of the Microsoft stack.

3.8 Frontend Framework

I looked into two of the most popular front-end frameworks Bootstrap (Bootstrap, 2016) and Foundation (Foundation, 2016). Bootstrap and Foundation are both front-end framework for faster and easier web development and gives you the ability to easily create responsive designs. They include HTML and CSS based "design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins" (w3schools, 2016) (wikipedia, 2016). After much taught I choose to use bootstrap due to it being preinstalled in visual studio, quicker set up time due to having to do less customisation and the ease of availability of themes from sites such as bootswatch (bootswatch, 2016) and cod snippets from bootsnip (bootsnip, 2016).

3.9 IDE

There are many reasons for using an Integrated Development Environment (IDE) over a text editor such as sublime or VS code. IDE'S provide debugging, Intelisense and error checking as you are typing out your code. I choose to use Microsoft Visual Studio 2015. Visual studios is Microsoft developing suite for developers to enable them to be able to create software. The tools in Visual studio are designed to work seamlessly with other Microsoft products. This IDE can be used to develop products for multiple platforms and by default visual studio has support for C# (microsoft, 2016). The reason I choose to use Visual Studios was because of how easy it is to install packages through nuGet package manager, c# Intellisense and how well it works with other Microsoft products such as Azure and Sql Server.

3.10 Database

A database is a collection of information (data) that is organised so that it can be easily accessed, managed and updated. For the database I looked into MySql and Sql server 2012. MySql is the most popular open-source SQL database currently. This is because it is fast, reliable and scalable. One of the drawbacks with MySql compared to Sql Sever is that it can be more prone to data corruption. Although Sql server 2012 sever is known to have a poorer performance compared to MySQL it is resistant to data corruption which I think is a greater advantage.

Another advantage of Sql server is that it works extremely well with other Microsoft products. SQL Server also has better support for foreign keys compared to MySql, which means it deals with relational databases better (Boney, 2013).

In the end the reason I choose Sql server 2012 for my database over MySql is due to the better tooling support it has for .NET (Entity Framework + other ORM) and Visual Studio support.

After I set up my Sql server database I created three tables in sql server. These tables were user_details which handles all user information, order_info which handles all information related to orders and product_infoes which handles all information related to the products.

These tables changed as the project evolved until I ended up with the finished product shown in fig.9.

The next step was creating the database connection in Visual Studio. Luckily this was very straight forward using the Server Explorer window and clicking Connect to Database option allowed me to enter the database name that I wished to use in my project.

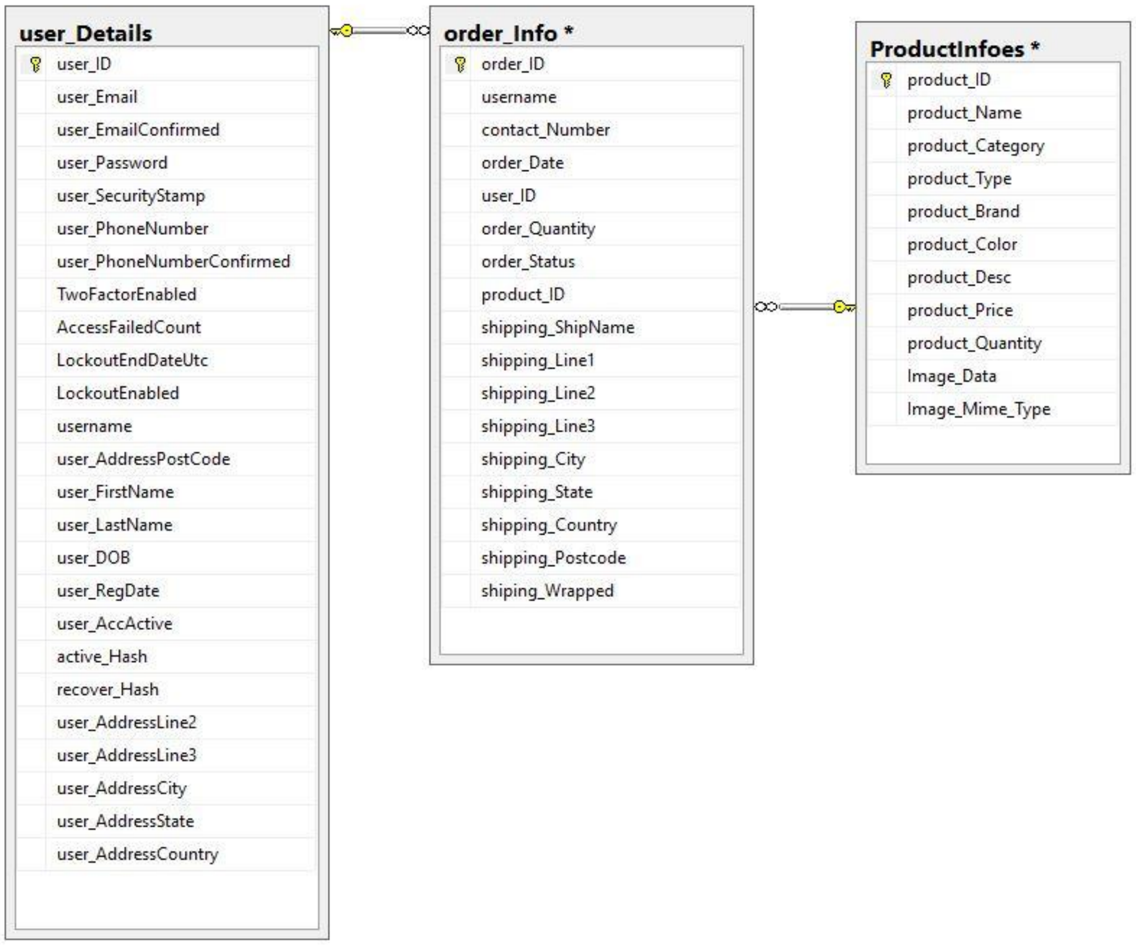


Fig.9 Database Design

3.11 Payment Method

The main reason I choose to use a third party for payments was because it reduces the security risk of storing user's payment information. When I was researching possible payment methods I discovered two that stud out above the rest these were Braintree

(Paypal, 2016) and stripe (stripe, 2016). I choose Braintree over stripe as Braintree allows PayPal transaction whereas Stripe doesn't. Although setting up Braintree was not as easy as I first thought it would be Braintree does have some great .net examples which helped me along the way. During the development of this project I used Braintree's sandbox environment. This allowed me to make dummy transactions for testing purposes using a dummy credit card and view if they went through successfully or not (shown in fig.12).

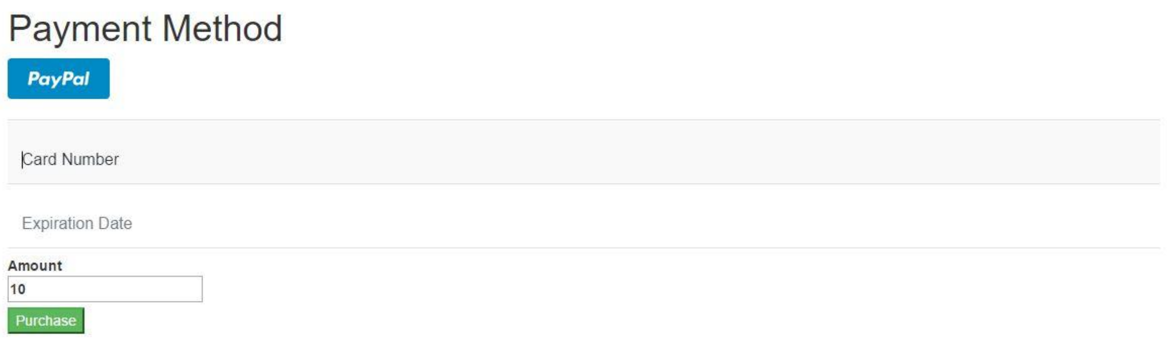


Fig.10 option for users to enter their credit card information

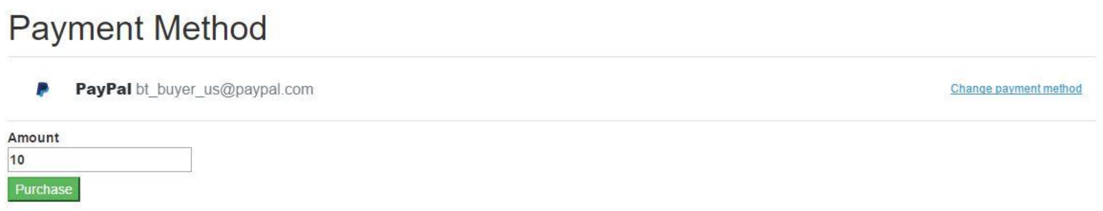
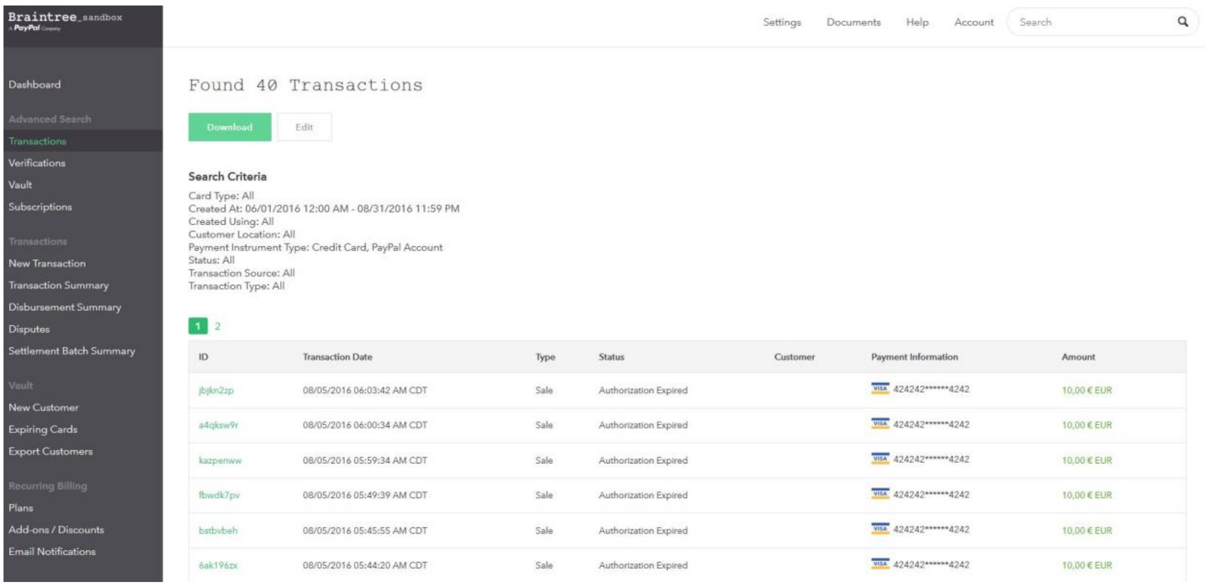


Fig.11 Selected paypal account



Found 40 Transactions

Search Criteria
Card Type: All
Created At: 06/01/2016 12:00 AM - 08/31/2016 11:59 PM
Created Using: All
Customer Location: All
Payment Instrument Type: Credit Card, PayPal Account
Status: All
Transaction Source: All
Transaction Type: All

ID	Transaction Date	Type	Status	Customer	Payment Information	Amount
jbjkn2zp	08/05/2016 06:03:42 AM CDT	Sale	Authorization Expired		VISA 424242******4242	10,00 € EUR
a4qksw9r	08/05/2016 06:00:34 AM CDT	Sale	Authorization Expired		VISA 424242******4242	10,00 € EUR
kazpenww	08/05/2016 05:59:34 AM CDT	Sale	Authorization Expired		VISA 424242******4242	10,00 € EUR
fbwdk7pv	08/05/2016 05:49:39 AM CDT	Sale	Authorization Expired		VISA 424242******4242	10,00 € EUR
bstbvbeh	08/05/2016 05:45:55 AM CDT	Sale	Authorization Expired		VISA 424242******4242	10,00 € EUR
6ak196zx	08/05/2016 05:44:20 AM CDT	Sale	Authorization Expired		VISA 424242******4242	10,00 € EUR

Fig.12 Braintree console dummy transitions list

```
    <add key="BraintreeEnvironment" value="sandbox" />
    <add key="BraintreeMerchantId" value="62vx549tvnz9jz39" />
    <add key="BraintreePublicKey" value="zvb9dz5xsyrwdc3y" />
    <add key="BraintreePrivateKey" value="0691b98a0b2f32a34a359d70c18fe04d" />
```

When I was setting up the Api work with my project I installed Braintree through NuGet package manger and then I stated what development environment I was using, my merchant ID, as well as my public and private keys for my account in the web config.

Next I created a Braintree folder which holds two configuration files these files manage the configuration settings for the payment gateway. Then I had to modify the Cart controller to allow for Braintree transactions. After this I created two views one which takes the customer's information and the second one which display's weather or not the transaction was a success and added there javascipt file.

3.12 Cloud

For my cloud service solution I looked into Azure, amazon web services and Digital Ocean. When I was looking into Digital Ocean I noticed that it was very much aimed for Linux developers and after setting up a droplet I realised I would need to do most of my work through the command prompt. Due to my lack of experience using linux and the added difficulty of doing everything threw the command prompt I decided this option wasn’t for me. Next I looked into Amazon Web services with this cloud provider I found the UI to be quite cumbersome for new users this combined with a steep learning curve to work how to set it up for hosting as well as some services not being available in Ireland made me choose not to go with this option at first. Finally I looked into Azure, although I had issues with documentation not being update from their previous Interface and some features being removed or renamed, there new UI was a lot easier to use than any of the other cloud services. Azure had handy quick start icons that guide you through whatever set up you needed to do. Although you may have less control over certain aspects of your cloud service using Azure, I found this one was perfect for my needs. Later on in my projects development I had to swap from Azure to amazon web services as my azure account key had expired. Although I was not happy to have to swap so late in the project AWS offered twelve months free usage for new users which was perfect for the duration of the project. Amazons services include Amazons EC2 (Elastic Compute Cloud) which “provides scalable computing capacity”. EC2 had a wide range of features such as virtual computing environments, known as instances which are perfect for hosting websites. When setting up Aws for hosting I had to create an instance. After this I had to open ports on the virtual machine and the server. I opened them in Aws console by creating a security group. Unfortunately I had issues with AWS which are mention later in problems encountered.

3.13 SMS authentication

I decided to use sms authentication for many reason being for easy account recovery, reducing number of fake accounts as well stopping people accidentally creating multiple accounts. For sms authentication I looked at both Twilio (twilio, 2016) and ASPSMS (aspsms, 2016). I decided to go with Twilio with it being the most popular sms authentication and I also because I got a chance to speak to Ben Nunney who is a manager in charge of EMEA Content Marketing at Twilio Inc when I was at the web summit. Ben told me a lot about this software and the services that they provided which is why I felt they were the better choice. I signed up to a free trial on twillos website which allows the user one free phone number for either voice or text Authentication. Using there documentation I worked out how to use their system with asp.net Identity and implement it into my project. I first had to create a phone number from the phone numbers section of their developers console. The next step was installing twillio from the NuGet Package Manager Console in visual studio using the "Install-Package Twilio" command to do so.

Next I had to find my Account SID, token and phone Number and store these values securely in the app settings.

```
  <appSettings>
      <add key="SMSAccountIdentification" value="ACac590f2ea94e15d4d47052d530558f92" />
      <add key="SMSAccountPassword" value="160729936751f26b79fa12f1efa851ac" />
      <add key="SMSAccountFrom" value="+353861802128" />
```

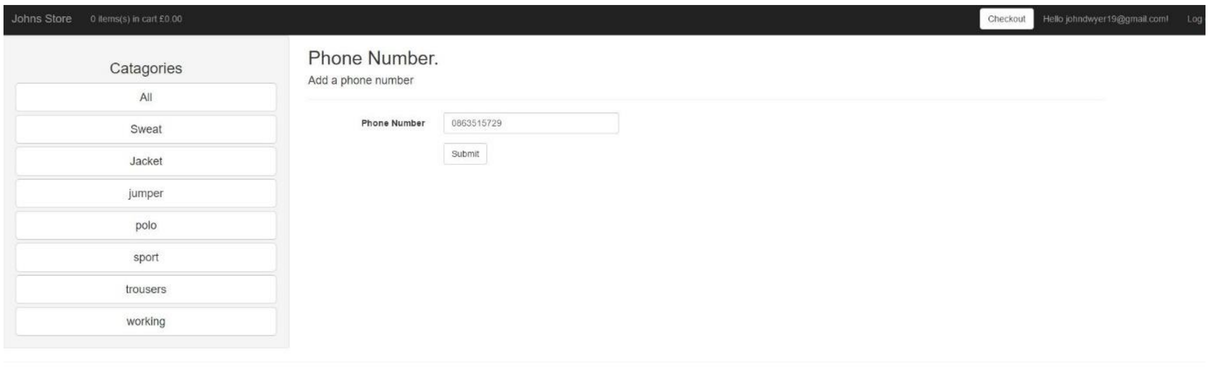
Following the set up in the app settings I had to configure the SmsService class in the *App_Start\IdentityConfig.cs* file (code shown below).

```
public Task SendAsync(IdentityMessage message)
        {
            // Plug in your SMS service here to send a text message.
            // Twilio Begin
            var Twilio = new TwilioRestClient(

System.Configuration.ConfigurationManager.AppSettings["SMSAccountIdentification"],
              System.Configuration.ConfigurationManager.AppSettings["SMSAccountPassword"]);
            var result = Twilio.SendMessage(
              System.Configuration.ConfigurationManager.AppSettings["SMSAccountFrom"],
              message.Destination, message.Body
            );
            //Status is one of Queued, Sending, Sent, Failed or null if the number is not
valid
            Trace.TraceInformation(result.Status);
            //Twilio doesn't currently have an async API, so return success.
            return Task.FromResult(0);
            // Twilio End

        }
```

SmsService

The last view steps I had to do was Update the View to support authenticaton and check that the EnableTwoFactorAuthentication and DisableTwoFactorAuthentication action methods in the ManageController had the ValidateAntiForgeryToken attribute.

When adding a number too your account you simply click on your email address in the top right which brings you to your account settings (shown in fig.16). From there you are given a number of options one of which is to add a mobile phone number. Once you click add number you are brought to a page to enter your mobile number (shown in fig.13).



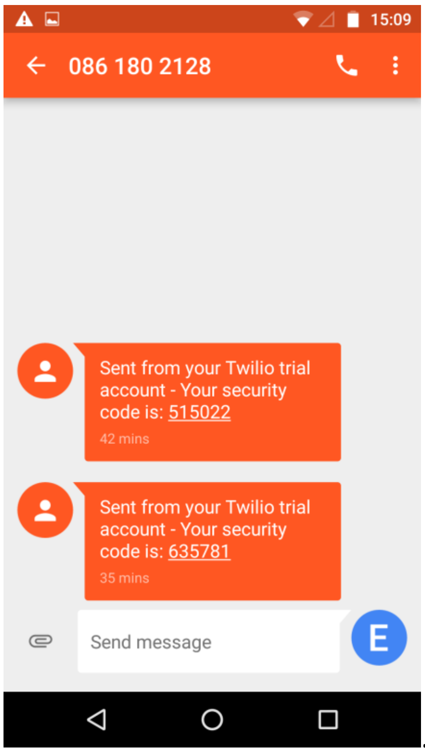
Phone Number.

Add a phone number

Phone Number: 0863515729

Fig. 13 entering your number

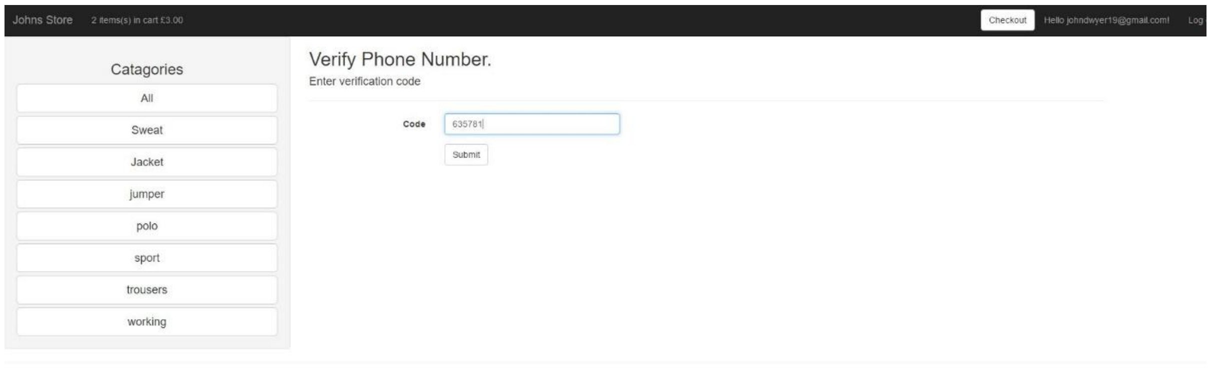
Once you enter your number you are then brought to a page (Fig.15) where you enter the security code that was sent to you by text shown in fig.14. This is the code you use to verify that your number is valid.



Sent from your Twilio trial account - Your security code is: 515022
42 mins

Sent from your Twilio trial account - Your security code is: 635781
35 mins

Fig.14 security code sent by Twilio


Johns Store 2 items(s) in cart £3.00 Checkout Hello johndwyer19@gmail.com! Log

Verify Phone Number.

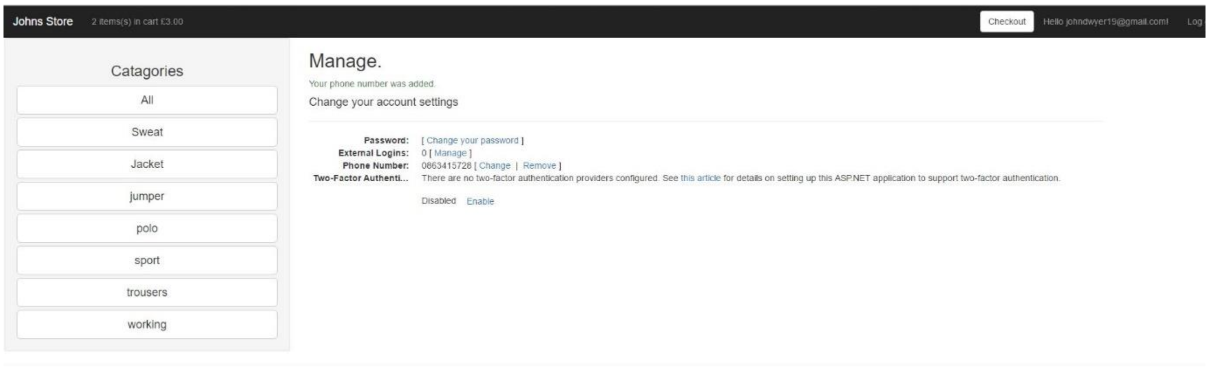
Enter verification code

Code: 635781

Submit

Fig. 15 verifying the phone number

After you enter the correct code it redirects you back to your account settings and gives you a message that your mobile number has been added. You are also able to change or remove this number if the case may be (shown fig.16).


Johns Store 2 items(s) in cart £3.00 Checkout Hello johndwyer19@gmail.com! Log

Manage.

Your phone number was added.

Change your account settings

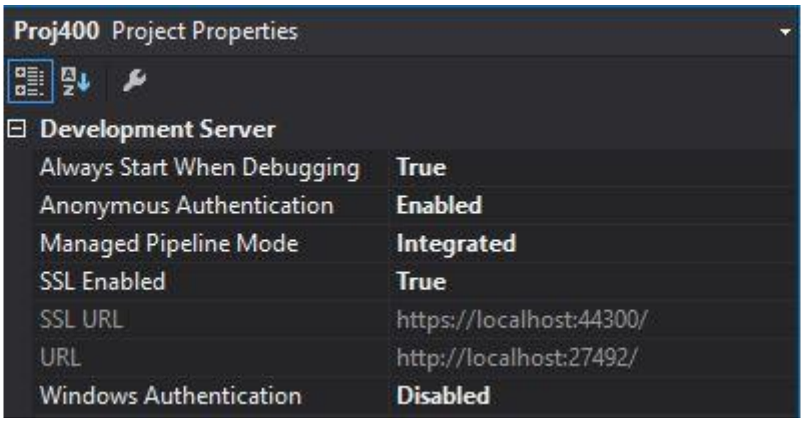
Password:	[Change your password]
External Logins:	0 [Manage]
Phone Number:	0863415728 [Change | Remove]
Two-Factor Authenti...	There are no two-factor authentication providers configured. See this article for details on setting up this ASP.NET application to support two-factor authentication.
	Disabled Enable

Fig.16 successful added number

3.14 SLL

As this is an e-commerce site setting up the IIS-Express to use SSL was very important. As the customers will be entering sensitive information on the site such as their payment information, name and address. It is important to keep using SSL after the user logs in and not to drop back to HTTP as without using SSL you're sending information in clear-text across the web. HTTPS is used for the protection of privacy and integrity of the exchanged data HTTPS is made up of HTTP with a connection encrypted by Secure Sockets Layer and is widely used across the internet (Freier, 2016).

The first thing I had to to enable SSL was to go to the project properties and change the SSL Enabled from false to True (shown Fig.17)and then go to the web tab in the project properties and change the project url to https://localhost:44300/ as to make it so when the project was running it would go to the https site.



Proj400 Project Properties

Development Server

Always Start When Debugging	True
Anonymous Authentication	Enabled
Managed Pipeline Mode	Integrated
SSL Enabled	True
SSL URL	https://localhost:44300/
URL	http://localhost:27492/
Windows Authentication	Disabled

Fig.17 enabling ssl

The final step I had to do was add the [RequireHttps] attribute to the ProductInfoController to make it a requirement that all requests must use HTTPS.

3.15 Dependency Injection

Dependency Injection is a software design pattern that is used for resolving dependencies**.** For my project I researched two different dependency Injections. The two I researched were Ninject and unity. I choose to use Ninject as it has a better fluent-configuration scheme and doesn’t need much configuration and unlike its Microsoft alternative it doesn’t require XML. When I was setting it up I installed two Packages Ninject MVC and Ninject.Web.Common. The next thing I did was add an Infrastructure folder. After the creation of this folder I added a class file called NinjectDependencyResolver.cs this is the custom dependency resolver. MVC uses a dependency resolver to create instances of the classes it needs to service requests. The next step was to create a link between the

NinjectDependencyResolver class and the MVC support for dependency injection in the App_Start/NinjectWebCommon.cs file.

```
private static void RegisterServices(IKernel kernel)
        {
            System.Web.Mvc.DependencyResolver.SetResolver(new
            Proj400.Infrastructure.NinjectDependencyResolver(kernel));
        }
```

Integrating Ninject in the NinjectWebCommon.

3.16 Moq

I used use a library called Moq which is used to create *mock objects*, which simulate the functionality of real objects from your project, but in a controlled way. I used Moq to test my code before using information from the database. This allowed me to insure that the code worked correctly before I used any database information. This allowed me to narrow the focus any errors the could be created with new functionality. Below is an example of where I mocked products from the database before I connected the program to the db.

```
        //Fake Database
            Mock<IProductsInfosRepository> mock = new Mock<IProductsInfosRepository>();
            mock.Setup(m => m.ProductInfos).Returns(new List<ProductInfo>(){
                new ProductInfo { product_Name = "Surf Board", product_Price=200},
                new ProductInfo { product_Name = "Surf Board 101", product_Price = 202 },
                new ProductInfo { product_Name = "Surf Board 202", product_Price = 203 } });

            kernel.Bind<IProductsInfosRepository>().ToConstant(mock.Object);
```

3.17 Image Uploads

In this project the Administrators are able upload product images and store them in the database so that they can be displayed in the product catalog .To do this I added two variable to the productinfoes table. These were Image_Data which data type was VARBINARY (MAX) and Image_Mime_Type which had a data type of VARCHAR (50). I also I needed to add two new fields to the productInfoes class that that correspond to the new database columns these were type byte for Image_Data and string for Image_Mime_Type. These had to exactly match the names I gave to the new columns in the database. I learnt that Web browsers will only upload files properly when the HTML form element defines an enctype value of multipart/form-data.

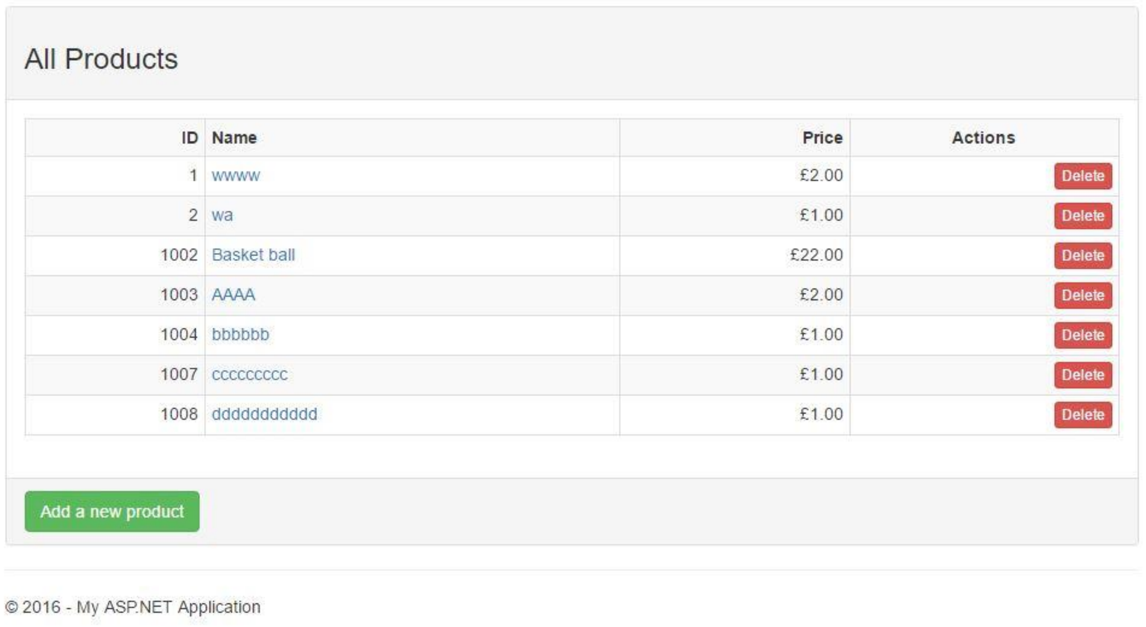
```
    @using (Html.BeginForm("Edit", "Admin", FormMethod.Post, new { enctype = "multipart/form-data" }))
```

Without the enctype attribute, the browser would pass on only the name of the file and not the actual image itself. In the Edit method I have a parameter which the MVC Framework uses to pass the uploaded file data to the action method. I copy this data and the MIME type from this parameter to the object so that the image is saved in the database.

Chapter 4 Design and User Interface

4.1 User Interface

The e-commerce site is aimed for both the owner and the customer with the aim of being a complete and simple system. The goal was to make a clean cut system which was easy to understand for both the user and the administrator at a glance. This is why I went with a clean white table design with colours that would be associated with everyday things such as traffic lights in the administration console shown in fig. 18. This is also why add a new product is green cause it gives the emotion that it is safe to go ahead and add a new product while delete is red as to give the user a warning that they shouldn't hit it unless they want to permanently remove this item. When deciding on colours throughout the project I would refer to the plutchik wheel(shown in fig.20) to make sure that the colours psychology match what the buttons accomplished for example red is harder to focus on therefore reducing the chance that the user will click on it. While green has a positive association meaning the user is more likely to click this button (Morin, 2016). If the administrator wish to edit an item they in the admin console they simply have to click on the name and they will be brought to a page where they can edit the details of that item.



All Products

ID	Name	Price	Actions
1	wwww	£2.00	Delete
2	wa	£1.00	Delete
1002	Basket ball	£22.00	Delete
1003	AAAA	£2.00	Delete
1004	bbbbbb	£1.00	Delete
1007	ccccccccc	£1.00	Delete
1008	dddddddddd	£1.00	Delete

Add a new product

© 2016 - My ASP.NET Application

Fig. 18 Admin Console

For the Home page I kept things simple with a slideshow of products that are in store and an image to the right and below it that would be images for new items in stock or sales (shown fig.19).

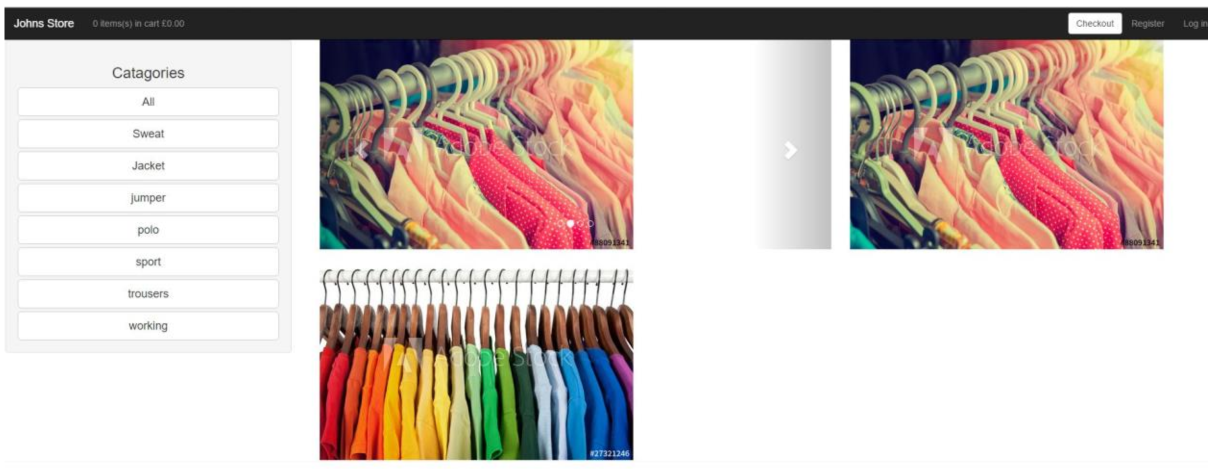


Fig.19 Home Page

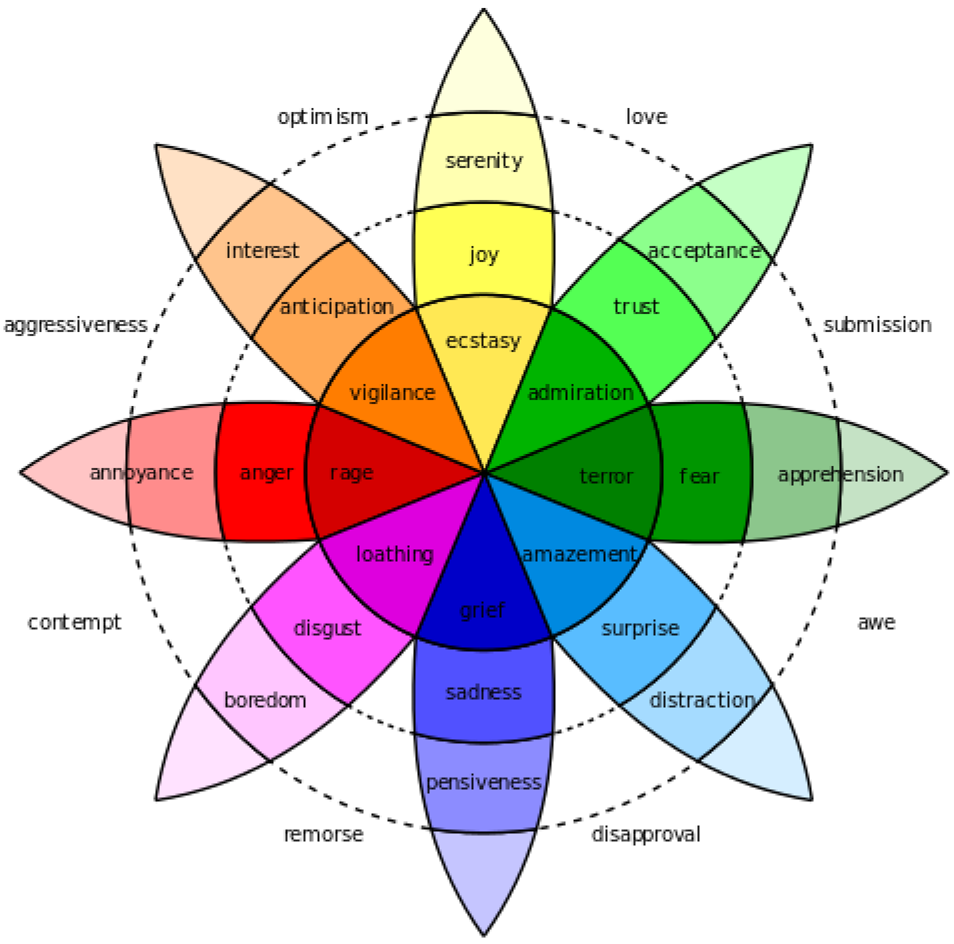
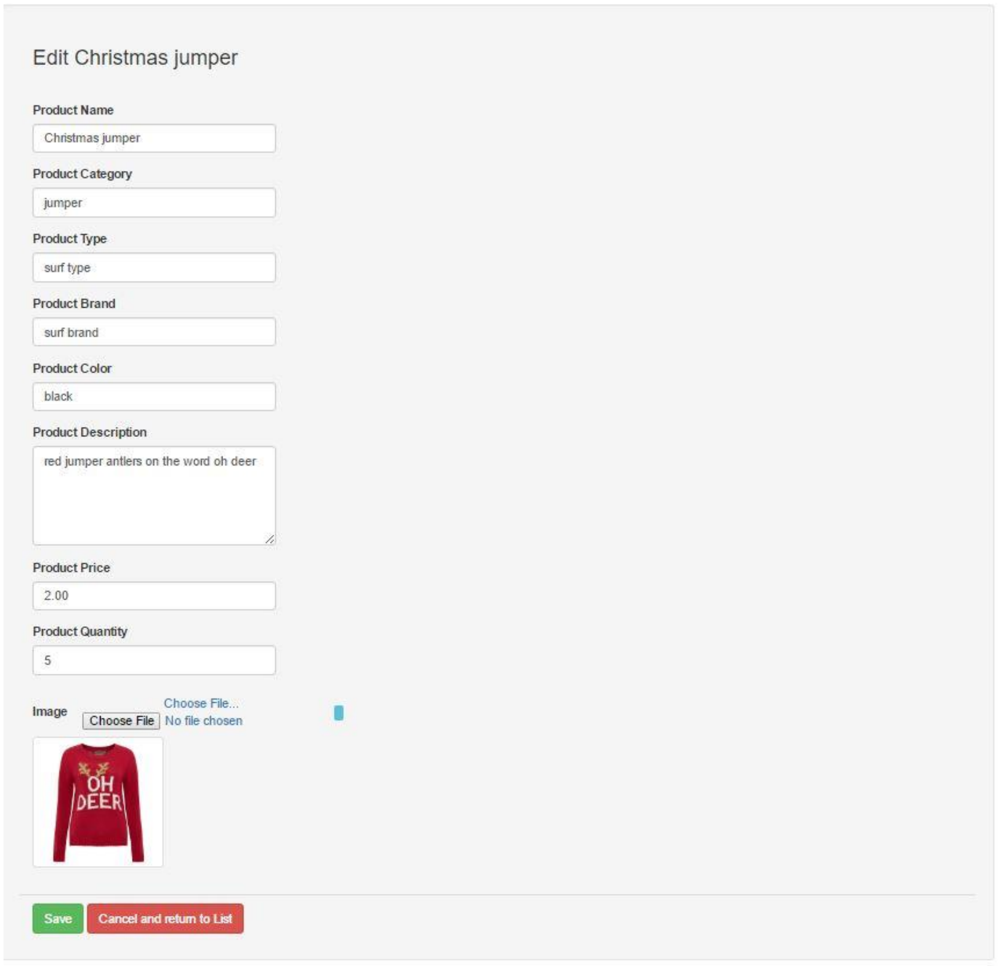


Fig. 20 Plutchik wheel

In the edit page (shown in fig.21) the administrator can change the products name, category, type, brand, colour, description and the quantity that is in stock and upload an image of the product. I have gone with a column style UI so that users can easily see what comes next and won't forget to enter any of the boxes. I used the Html.ValidationSummary for validation which displays a message for each of the properties to tell them if they were missing something that was required.


Edit Christmas jumper

Product Name
Christmas jumper

Product Category
jumper

Product Type
surf type

Product Brand
surf brand

Product Color
black

Product Description
red jumper antlers on the word oh deer

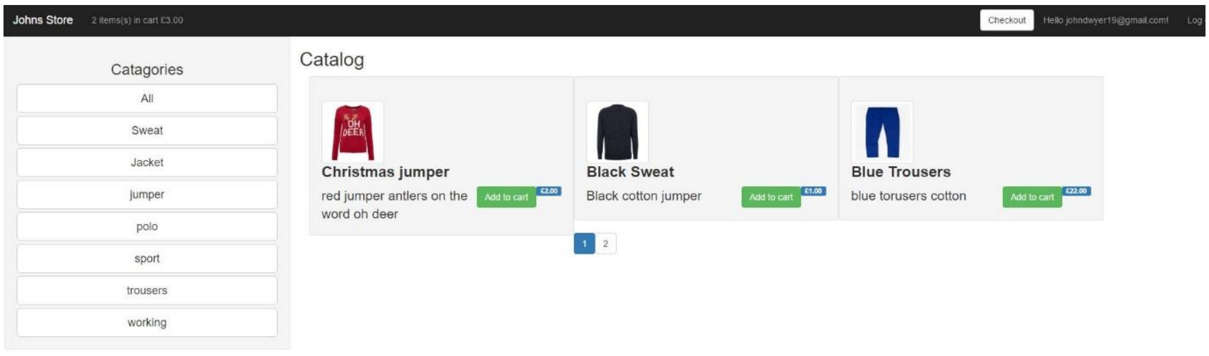
Product Price
2.00

Product Quantity
5

Image Choose File... Choose File No file chosen

Fig.21 editing an item in the developers console

For the catalogue(shown in Fig.22) I kept things simple by only displaying a limited number of items and allowing pagination which means that when the user is finished with this page they can go on to the next. There's a couple of advantages to this. Number one the users machine doesn't have to load a huge amount of images so it less intensive on their machine. It also allows for the user to not get lost scrolling in a ridiculously long page. I also have the categories displayed neatly in the corner so they can swap at a glance if they wished to do so. This also promotes the user to look at other categories and there for increase sales.



Catalog

Christmas jumper
red jumper antlers on the word oh deer
£2.00

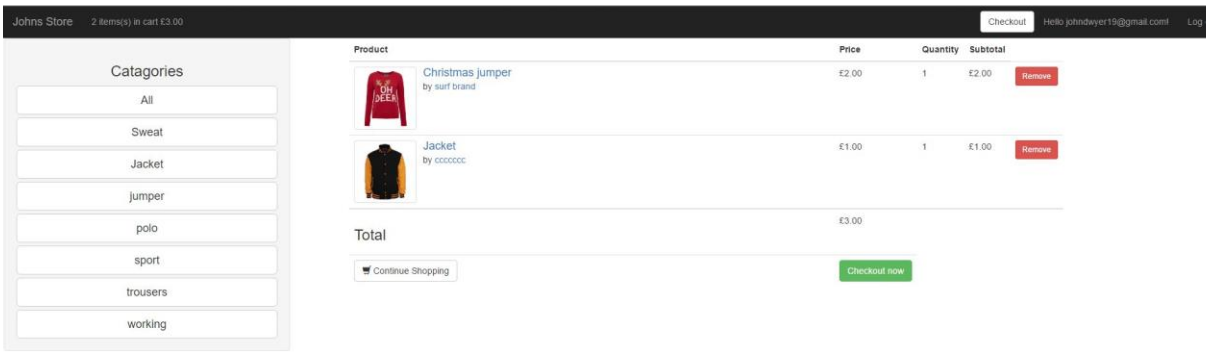
Black Sweat
Black cotton jumper
£1.00

Blue Trousers
blue torusers cotton
£22.00

Fig.22 Catalogue

For the shopping cart (shown in fig 23) I used a table to display all the items that are in the cart. Each products are separated into different rows with an image of the product on the left with the name of the product and the brand beside it. On the right hand side you can see the price, quantity, subtotal and the remove.

The total of all the items is displayed in a different row as to allow the customer to be able to see clearly how much the transaction will cost so that there is no surprises.



Product	Price	Quantity	Subtotal	
Christmas jumper by surf brand	£2.00	1	£2.00	Remove
Jacket by ccccccc	£1.00	1	£1.00	Remove
Total	£3.00			

Fig.23 shopping cart

I also have a running total that is displaying the number of items the user has in the cart and the total price (shown fig.24) displayed in the top left.

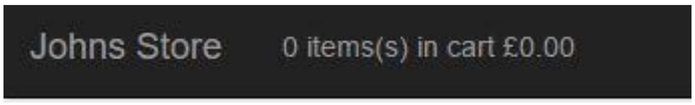

Johns Store 0 items(s) in cart £0.00

Fig.24 navbar running total

4.2 Class Diagrams

Class diagrams are a type of structure diagram that describes the structure of a system. It does this by showing the classes attributes, methods and their relationships to other objects (agilemodeling, 2016). In the class diagram shown in fig. 25 you will notice similarity's to the tables in the sql database. These Similarities are the Product_Infoes, Order_Info and user_Info tables but you will also notice 3 new tables these are CartRow, Cart and CartIndex. These tables are in charge of the cart. CartRow class represents a product that the customer has added to the cart. CartIndex is used to display the contents of the cart. The cart class is used for actual cart functionality such as adding or removing items from the cart or calculating the total value of all the items that are in the cart.

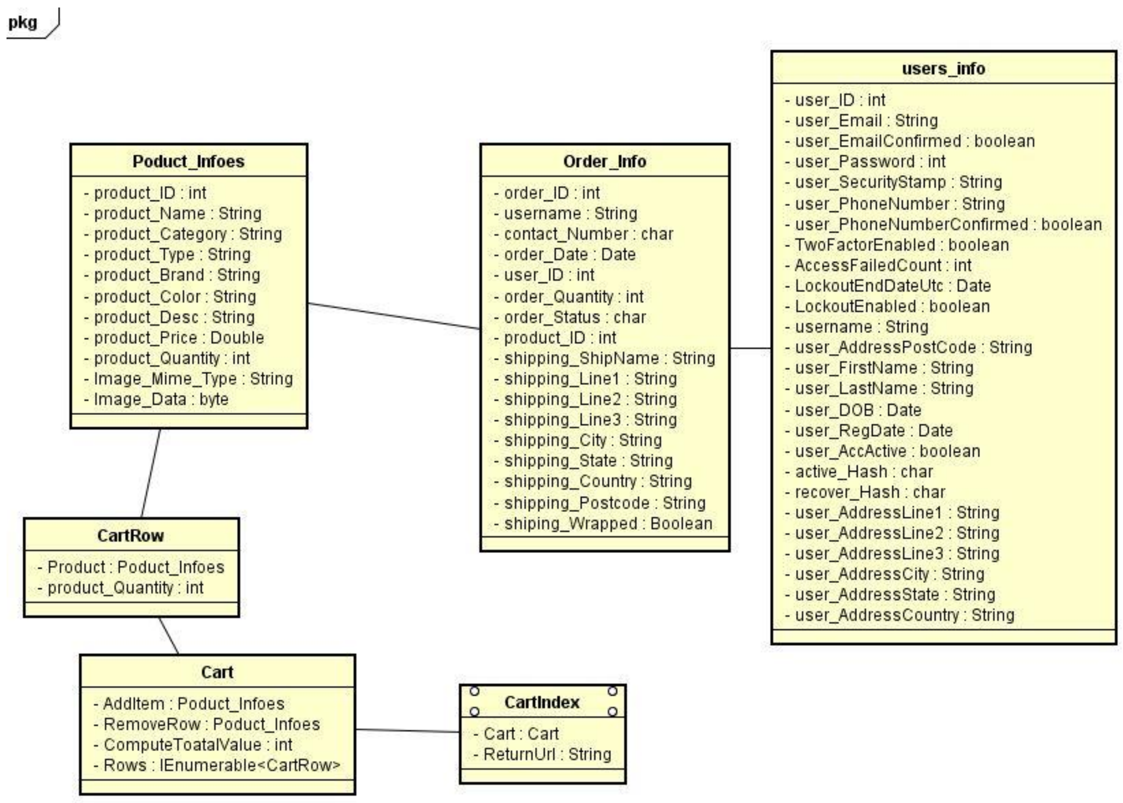

pkg
Poduct_Infoes
- product_ID : int
- product_Name : String
- product_Category : String
- product_Type : String
- product_Brand : String
- product_Color : String
- product_Desc : String
- product_Price : Double
- product_Quantity : int
- Image_Mime_Type : String
- Image_Data : byte
Order_Info
- order_ID : int
- username : String
- contact_Number : char
- order_Date : Date
- user_ID : int
- order_Quantity : int
- order_Status : char
- product_ID : int
- shipping_ShipName : String
- shipping_Line1 : String
- shipping_Line2 : String
- shipping_Line3 : String
- shipping_City : String
- shipping_State : String
- shipping_Country : String
- shipping_Postcode : String
- shiping_Wrapped : Boolean
users_info
- user_ID : int
- user_Email : String
- user_EmailConfirmed : boolean
- user_Password : int
- user_SecurityStamp : String
- user_PhoneNumber : String
- user_PhoneNumberConfirmed : boolean
- TwoFactorEnabled : boolean
- AccessFailedCount : int
- LockoutEndDateUtc : Date
- LockoutEnabled : boolean
- username : String
- user_AddressPostCode : String
- user_FirstName : String
- user_LastName : String
- user_DOB : Date
- user_RegDate : Date
- user_AccActive : boolean
- active_Hash : char
- recover_Hash : char
- user_AddressLine1 : String
- user_AddressLine2 : String
- user_AddressLine3 : String
- user_AddressCity : String
- user_AddressState : String
- user_AddressCountry : String
CartRow
- Product : Poduct_Infoes
- product_Quantity : int
Cart
- AddItem : Poduct_Infoes
- RemoveRow : Poduct_Infoes
- ComputeToatalValue : int
- Rows : IEnumerable<CartRow>
CartIndex
- Cart : Cart
- ReturnUrl : String

Fig.25 cart class’s connections

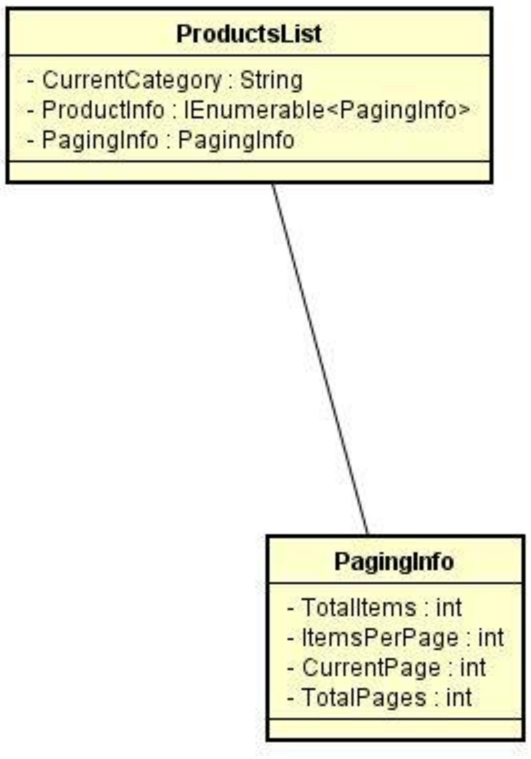


Fig.26 ProductList and PagingInfo class's

In fig.26 I show the connection between the ProductList and the PagingInfo class. ProductList is used to provide an instance of the PagingInfo view to the view. I use pagingInfo to pass information to the view about the how many items to display on the page, number of pages available, the current page and the total number of products in the available.

4.3 Time Management

I used Trello at the start to organise what I should do but quickly discovered that it is great for people working in teams allowing you to display and organise and revaluate priorities so that entire groups could see no matter their location. But I found it simply overkill for a single user to track time. As many of Trellos features would not be required and would off consumed extra time vs a notebook or sticky notes on a wall. Taking this into consideration I simply went with a blog and some sticky notes which I could quickly write down and re-organise my tasks as required. I found this method to be very beneficial in comparison with Trello as I was a one man team I had little reason to be sharing my tasks and re-organising with other and this kept my task ever visible and always on the top of my mind compared to being hidden in a tab or a website that I would only view a handful of times a day.

4.4 User accounts

For user accounts authentication I decided to use ASP.NET Identity and make the most of the .net framework. When I created my application I selected Individual User Accounts and visual studio set up the solution. This uses ASP.NET Identity which enables an easy set up that allows the users to register an account, using their email and password. ASP.Net Identity uses Entity Framework Code First, which allowed me to modify the tables and deploy any changes easily using Code First Migrations.

4.5 Problems Encountered

During the duration of the project there was many issues that appeared and delayed progress. For example the hosting. At the start of my project I was using azure and I found that a lot of their documentation was outdated as they were using example of the previous azure portal. Because of this they were mentioning features that names have been change or have been deprecated for example when using send grid Microsoft linked to the old azure portal store only to be told that it is no longer available.

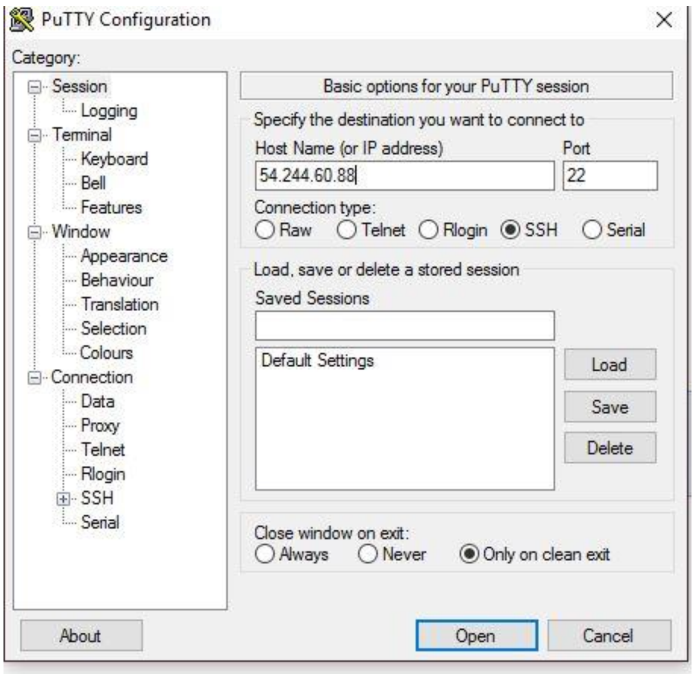
Another issue I had was with Bootstrap and jquery bootstrap has not been updated to work with the latest jquery version which caused issues with the drop down lists just not working. This was an issue as Brain tree requires a fairly new version of jquery to work.

Towards the end of my project my azure pass expired I went communicated with lectures to get a new key but unfortunately these were all expired as well. As a result I went onto

DreamSpark to get the student azure key pass only to discover that the student version is heavily restricted with access to certain items being stopped in the store as well as other restrictions. Due to this I decided Azure was no longer the way to go and as I had researched Amazon web services in an earlier sprint when I was deciding on a cloud hosting platform I knew this would be the correct choice.

Unfortunately this did not come without its own list of issues such as the learning the AWS interface unlike azure which you could have an idea of what to do at a glance using AWS requires a lot of reading from there documentation to be able to use it. AWS took a view weeks for me to understand what exactly I required for hosting.

The next issue I had with AWS was when I set up the windows virtual machine I set up Apache Tomcat but it refused to start for an unknown reason so I decide instead to use wamp but that claimed there was a missing dll called "mscvcr110.dll" I discovered this was a Microsoft Visual C++ Redistributable dll so installed the latest update yet it still complained of a missing dll. I then downloaded that dll by itself and attempted to put it in with all the already pre-existing dll's on the machine but the file already existed there. It was at this point that I realised I was encountering issues that I shouldn't be and decided to swap from the windows VM to the Ubuntu VM. When I swapped to an Ubuntu instance I noticed there was two different was to connect to the Ubuntu VM. Both were through a command shell which added an extra layer of difficulty to the set up. The first one was called PuTTy (shown in fig. 27) but I couldn't seem to get access to open the ports I required on the server yet I was able to open them on the AWS console but not on the VM.



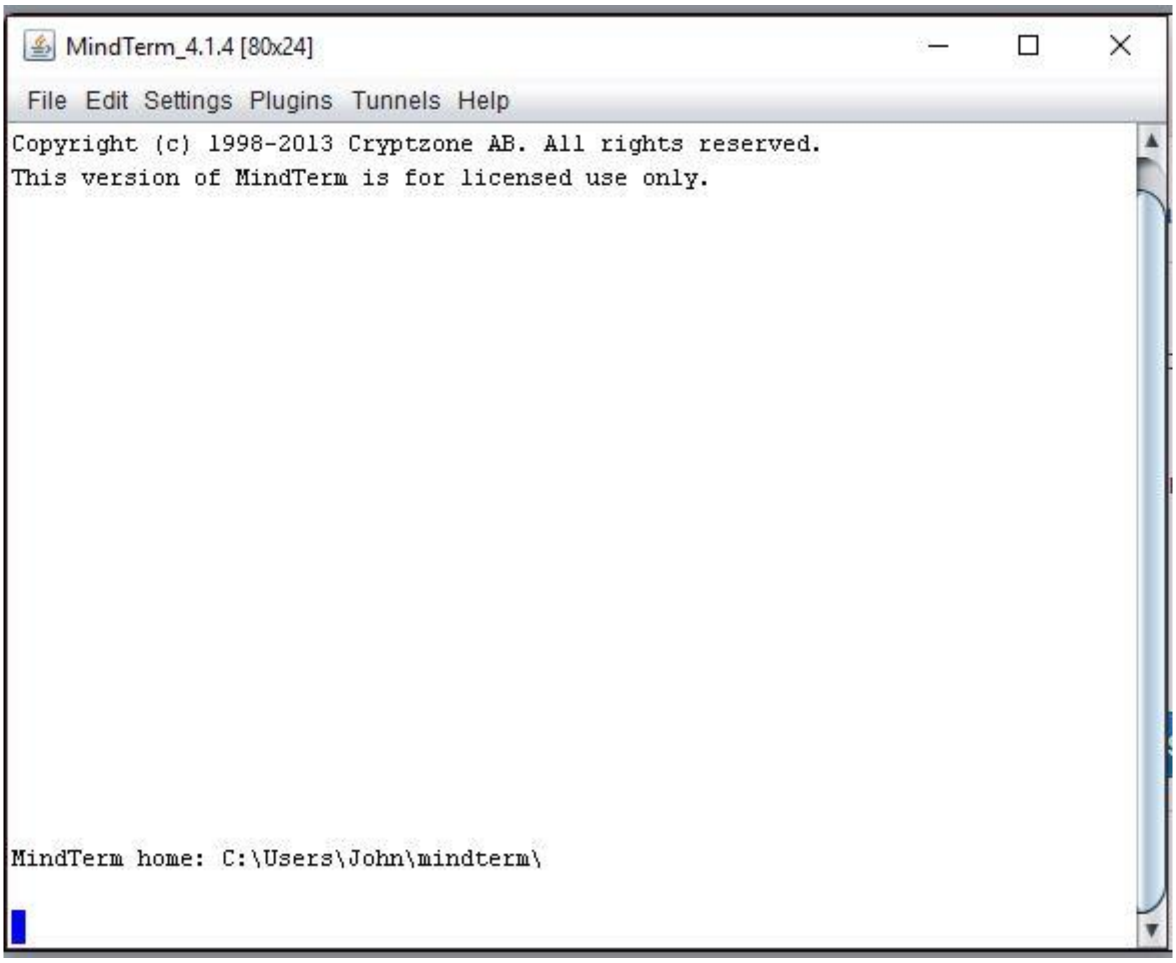
Specify the destination you want to connect to
Host Name (or IP address): 54.244.60.88
Port: 22
Connection type: Raw Telnet Rlogin [✓ SSH] Serial
Load, save or delete a stored session
Saved Sessions
Default Settings
Close window on exit: Always Never [✓ Only on clean exit]

Fig.27 PuTTy console

It was at this point I decided to use the second option which was a Java SSH Client directly from my browser called mind term (shown in fig.28). I had to update my java and at last this

```
sudo ufw allow 22
```

worked and I was able to open all the required ports on the Ubuntu uncomplicated firewall (ufw) (ubuntu, 2016) code to open a port below.But yet I was unfortunately still not able to access the server which was hosted at “http://54.244.60.88:8080/”.



```
Copyright (c) 1998-2013 Cryptzone AB. All rights reserved.
This version of MindTerm is for licensed use only.














MindTerm home: C:\Users\John\mindterm\

```

Fig.28 Mindterm console

4.6 Results

All though I didn't get a chance to accomplish everything I set out to do the main aims of the project were achieved. The application allows customers to register/login, connect their mobile number to their account, view all items that are in stock from the database, filter by specific categories, add items to the cart and calculate the total cost and make a transaction. While on the administrators side they are able to create, edit and delete products. Another piece of functionality that was successfully achieved was pagination of the products and the ability for the admin to upload images of there products.

4.7 Critical evaluation

The main aims of the project where achieved looking back I would have liked to be able to make everything fit together seamlessly. I feel as though overall the product that was produced could have used more work. But due to time constraint of 3 months and trying to balance my academic and personal life during the summer months proved quite troublesome as you can never plan for unforeseen circumstances. With this project I wanted to revisit and learn old topics as well as new to refresh my memory and touch up in areas I lacked before going out into the working world and although I didn't accomplished everything I would have liked I feel I have accomplished what I need for the future.

4.8 Further Reading

Blog: https://proj400summer.wordpress.com/

Github: https://github.com/s00130070/Proj400_Summer

Chapter 5 References

agilemodeling, 2016. *Conceptual Class Diagrams.* [Online]
Available at: http://www.agilemodeling.com/artifacts/classDiagram.htm
[Accessed 10 June 2016].
amazon, 2016. *amazon.* [Online]
Available at: https://www.amazon.co.uk/
[Accessed 2 June 2016].
aspsms, 2016. *aspsms.* [Online]
Available at: http://www.aspsms.com/
[Accessed 05 August 2016].
astah, 2016. *astah.* [Online]
Available at: http://astah.net/student-license-request
[Accessed 1 June 2016].
bitbucket, 2016. *bitbucket.* [Online]
Available at: https://bitbucket.org/
[Accessed 10 June 2016].
Boney, 2013. *Choosing Your DBMS: MySQL vs SQL Server.* [Online]
Available at: http://www.webnethosting.net/choosing-your-dbms-mysql-vs-sql-server/
[Accessed 31 May 2016].
bootsnip, 2016. *bootsnip.* [Online]
Available at: http://bootsnipp.com/
[Accessed 20 June 2016].
Bootstrap, 2016. *Bootstrap.* [Online]
Available at: http://getbootstrap.com/
[Accessed 4 June 2016].
bootswatch, 2016. *Free themes for Bootstrap.* [Online]
Available at: https://bootswatch.com/
[Accessed 22 June 2016].
dugan, j., 5. *Visual Studio Code vs. Sublime Text.* [Online]
Available at: https://john-dugan.com/visual-studio-code-vs-sublime-text/
[Accessed 5 November 2015].
Foundation, 2016. *Foundation.* [Online]
Available at: http://foundation.zurb.com/
[Accessed 4 June 2016].
Freeman, A., 2013. *Pro ASP.NET MVC 5.* Fifth ed. s.l.:Apress.
Freier, A., 2016. *SSL Protocol.* [Online]
Available at: https://tools.ietf.org/html/draft-ietf-tls-ssl-version3-00
[Accessed 25 July 2016].
github, 2016. *github.* [Online]
Available at: https://github.com/
[Accessed 10 June 2016].
microsoft, 2016. *Visual Studio IDE.* [Online]
Available at: https://msdn.microsoft.com/en-us/library/dn762121.aspx?f=255&MSPPError=-2147217396
[Accessed 31 May 2016].
Morin, A., 2016. *How To Use Color Psychology To Give Your Business An Edge.* [Online]
Available at: http://www.forbes.com/sites/amymorin/2014/02/04/how-to-use-color-

psychology-to-give-your-business-an-edge/#16a848d62e28
[Accessed 24 July 2016].
Paypal, 2016. *braintree.* [Online]
Available at:
https://sandbox.braintreegateway.com/merchants/62vx549tvnz9jz39/transactions/advanced_search
[Accessed 5 July 2016].
riverisland, 2016. *riverisland.* [Online]
Available at: http://eu.riverisland.com/
[Accessed 2 June 2016].
skatehut, 2016. *skatehut.* [Online]
Available at: https://www.skatehut.co.uk/
[Accessed 2 June 2016].
stripe, 2016. *stripe.* [Online]
Available at: https://stripe.com/ie
[Accessed 5 July 2016].
twilio, 2016. *twilio.* [Online]
Available at: https://www.twilio.com/
[Accessed 10 august 2016].
ubuntu, 2016. *Firewall.* [Online]
Available at: https://help.ubuntu.com/lts/serverguide/firewall.html
[Accessed 14 august 2016].
w3schools, 2016. *Bootstrap Get Started.* [Online]
Available at: http://www.w3schools.com/bootstrap/bootstrap_get_started.asp
[Accessed 3 June 2016].
wikipedia, 2016. *Foundation(framework).* [Online]
Available at: https://en.wikipedia.org/wiki/Foundation_(framework)
[Accessed 5 June 2016].