FACULTY OF SCIENCE, ENGINEERING AND COMPUTING

School of *Computing and Information Systems*

BSc DEGREE

IN

Computer Science

PROJECT DISSERTATION

Name: Mahmoud Hachich

ID Number: K1421103

Project Title: Design and Implementation of a e-commerce website for Hachich Tiles

Project Type : Build

Date: 20/04/2017

Supervisor: Professor James Orwell

Kingston University London

Did you discuss and agree the viability of your project idea with your supervisor?	Yes
Did you submit a draft of your project dissertation to your supervisor?	Yes
Did you receive feedback from your supervisor on any submitted draft?	Yes

Plagiarism Declaration

Declaration

I have read and understood the University regulations on plagiarism and I understand the meaning of the word *plagiarism*. I declare that this report is entirely my own work. Any other sources are duly acknowledged and referenced according to the requirements of the School of Computer Science and Mathematics. All verbatim citations are indicated by double quotation marks ("…"). Neither in part nor in its entirety have I made use of another student's work and pretended that it is the own. I have not asked anybody to contribute to this project in the form of code, text or drawings. I did not allow and will not allow anyone to copy my work with the intention of presenting it as their own work.

Date: 20/04/2017

Signature

Abstract

"With online sales rapidly increasing throughout the UK, it is predicted that more than 89 billion in sales will be projected through e-commerce for the year 2017" (Statista, 2017). This is purely due to the fact that businesses are now available online and through different multimedia platforms, allowing a wider target audience to be reached. The purpose of this project is to develop a dynamic database driven website for the client. This e-commerce website will be developed through the use of MySQL and PHP, aiding communication with the database alongside HTML, CSS and JavaScript for front-end. The success of this website will contribute to the client's sales and hopefully deal with all his current problems.

The problem that the client is currently facing is that his company is not generating enough sales which have led to a cash flow problem. The client has a small tiles and ceramics store and wishes to expand to the online market. Customers are currently complaining due to the lack of communication that is taking place when a customer buys a product or requests products. All customers have to drive to the store to view the products on offer as he currently doesn't have a catalogue. Thus the build of this project will be based on the implementation of an e-commerce store for the client which will allow customers to browse the catalogue, message the client and perform other functions.-

The business sales are predicted to increase with the release of this e-commerce store. As all coding will be developed from scratch, all requirements will be implemented with respect to the client's needs. A customer and admin view will be created allowing the customer to perform basic functionality whilst the customer must be able to add, edit and update products in the shopping cart. The admin must also have control over editing and deleting customer accounts.

Contents

1. Project Introduction

In this section of the report, a detailed introduction into the clients business and his current problems and will be given. The client, who is a high-quality ceramics retailer contacted me during the summer of 2016 with his challenge and requirements and believed the development of an e-commerce website was the best possible solution.

1.1. Business Background

Hachich Tiles is a small tile selling business which is run by the client and his friends. Hachich Tiles was established in 2010 and has one branch. The business is located in Shepherds Bush which is an area in West London with great infrastructure and a famous shopping centre called Westfield. Being a location of high residency and with a large volume of shoppers, sales have been predicted to be high.

The client currently stocks product samples in his store and customers willing to purchase the products on offer have to visit the store to have a look at the products before purchasing. Orders are then placed and if the requested stock is found, customers are contacted to arrange a collection or delivery day.

Hachich Tiles provides its customers with a wide range of tiles and tiling accessories. The business has attracted many customers due to its unique selling point, the importing of rare tiles from Jordan and Turkey. This unique selling point has enabled Hachich Tiles to charge a premium price for its exceptional high quality products. The importing and selling of these unique tiles plays a major role in Hachich Tiles revenue stream. Hachich Tiles is currently generating all of its income through the selling of its tiles and ceramics. Brand recognition and awareness is developed through word of mouth, customer recommendations and an extensive network of contacts.

1.2. Business Problems

Hachich Tiles is currently experiencing a lot of problems that must be dealt with in order to remain in the tilling industry. One major problem facing the business is a decrease in total sales. Consequently, the business has experienced financial problems as their expenses exceed the total income being generated. As a result, the total profit has drastically reduced and a solution should be provided as soon as possible to avoid the total collapse of the business.

In addition, there is poor distribution of information and customers do not get adequate communication about the levels of stock and the availability of the products which are displayed in store. Some customers become irritated when they order goods and are told that they have to wait for some days for importation. A customer working in the construction industry will have to meet client deadlines, and delivery of late products can therefore jeopardise their reputation and therefore opt to purchase products from competitors such as Topps Tiles, B&Q and many more.

1.3. Proposed Solutions

There are many solutions to deal with the problem of low sales, including lowering costs, conducting marketing campaigns and offering promotions, increasing brand awareness and targeting a wider audience range. After a meeting with the client, a decision was made to create a company website as a way of reaching out to many people with the goal of increasing sales.

The proposed solution is therefore the development of a dynamic database driven website which will act as a tool in increasing sales and advertisement. The website will consist of a HTML and CSS driven interface followed by a PHP and MySQL database.

The website will give customers the ability to browse the product catalogue and make purchases through a simple interface.

Hachich Tiles Architectural Design:

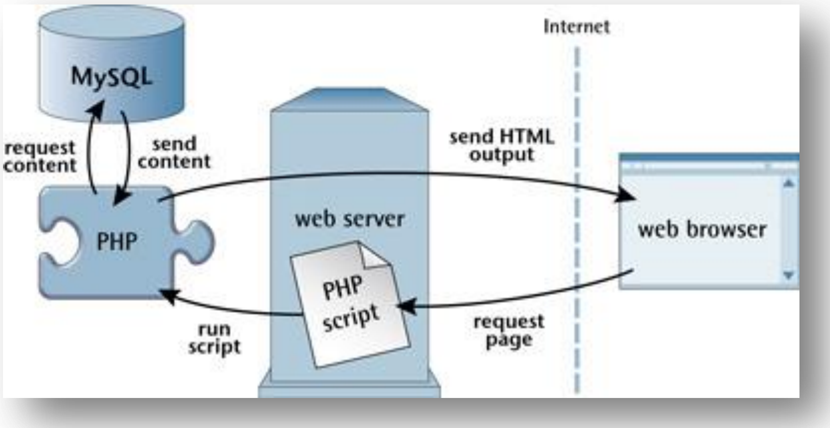


Figure 1: Communication between the web browser and MySQL database is conducted through the PHP script which is run to request or to send the needed content. (Yank, K. 2009)

2. Project Aims & Objectives

As mentioned above in chapter 1.3, the aim of the project is to design and build a fully functional database driven e-commerce website to help increase the sales for Hachich Tiles. The website will enable the exchange of goods and money between the customers and the clients business. The customers will be able to perform simple tasks such as, searching for products, adding a product to a cart, making enquires and viewing orders.

2.1. Objectives

"Objectives must be derived from, and consistent with, the intention of the identified goals" (Phil Bartle 2007).

O1. Offer a website which allows the customer to view products and place order 24/7 365 days a year.
O2. Improve Hachich Tiles cash flow problems by 14% before the end of 2017.
O3. Provide information on the products which are available in store on the website.
O4. Increase Hachich Tiles profits by 5% before the end of 2017.
O5. Allow users to make an inquiry on bulk orders before purchases.
O6. Increase the clients store brand awareness and increase his market share by the end of 2017.

2.2. Major phases

Literature review – This will contain information about the projects goals, target audience, competition and the e-commerce market. This review of literature should give me an insight into what makes the website successful, and how to meet all customer expectations.

Analysis – This will provide a clear definition of what the problem is followed by the methodology that is going to be used to achieve the aims of the project. The analysis phase should ensure a solid understanding of user requirements to ensure a successful website.

Design – A phase which will determine the way the website will look and feel for the user. This phase should place emphasis on the target audience as different websites adapt the website look to attract that particular target audience. The activity and sequences diagrams representing the websites operation will all be included in this section of the project. Wireframes will be created to give the client an idea of how the website will look

Implementation – Where the design goes live. Pictures of different code and explanations should be included in this phase of the project. Discussions on benefits of the code chosen to meet aims will also be included here.

Testing and evaluation – Testing of the website is essential to ensure it is error free and functional. This section will outline whether the project aims have been met and if not then what could be done to ensure that the aims are reached next time.

2.3. Project Milestones

The milestones below show the major stages that the project will undergo to ensure its success.

- Client understanding gained through interviews and from design feedback
- Approved wireframes
- First website prototype created
- Final prototype created
- Client changes implemented
- Pre website launch
- Final launch
- Website evaluation

----------The rest of this page was left blank on purpose ----------

3. Literature Review

The purpose of this section, the literature review, was conducted to gain a deeper understanding into the ways e-commerce websites work and what factors affect them. A discussion of the relevant technologies that currently exist in e-commerce will be offered alongside an evaluation of the relevant products and services. Alongside the discussions, the findings of the literature review should explore the ways in which companies acquire sales online and how business success can depend heavily on the number of sales made through the web.

3.1. What is E-commerce?

The electronic sales market has changed hugely over the last 30 years; at the start e-commerce was facilitated through electronic data interchanges and electronic fund transfers. The increased use of credit cards in the 1980's and automated teller machines (ATM's) showed an increase in the exchange of goods and services over the internet. The goods and services exchanged online can vary to the type of business; this includes retails sites, auction websites, music website and many more.

"Electronic commerce or e-commerce is defined as use of the internet to facilitate the transaction of business activities that include the selling of goods and services, purchasing of products (and services), and even the provision of after sales support over the internet" (Anita Rosen, 2000, p43). Rather than the traditional physical interaction of conducting business, e-commerce facilitates the transaction of a wide range of business activities to be conducted electronically by the relevant parties involved without having direct physical contact. Due to recent advancements in e-commerce, it comprehensively refers to the application of digital data processing technology systems coupled with electronic communications in business activities and transactions to facilitate the transformation and creation of refined relationships that enhance the creation of value between the organization and its customers and partners as well.

One of the main features of e-commerce is the use of electronic modes of payment such as the electronic funds transfer, credit or debit cards and so forth, eliminating the use of cash payments. Another key feature of e-commerce is the 24-hour availability of business services to customers regardless of geographical location. "In addition to providing reliable and efficient communication with clients and partners, e-commerce also facilitates the provision of pre-sales and aftersales support services to clients" (Mesenbourg, 2005, p89). "Equally important features of e-commerce are its marketing and advertising features with the capability of widening the reach of marketing the products and services offered by the business, and the inventory management features that automate the management of inventory to be more efficient "(Alexander Hutchinson, 2012, p102).

3.2. Impacts of E-commerce

Customers have been able to access the services offered by many different companies all over the world via the internet no matter where they are. Most products which can be purchased in store can be now found online, whether it be clothing, food, gadgets or tools.

As mentioned by Chris and Christopher, "e-commerce retail is the fastest growing trade sector and has outpaced every other trade and manufacturing sector since 1999" (G., C. and Jr., C. 2011). With such domination in the trade and manufacturing sector, businesses have been able to lower costs and increase profits through online selling. E-commerce has therefore made it easy for companies to have low cost high profit margins.

The past four years for the UK have shown the highest numbers in business to customer sales the country has ever seen. “E-commerce sales have increased in the United Kingdom from £54.97 billion in 2012 to a staggering £83.66 billion in 2016” (Statista, 2016). Such a huge increase in sales demonstrates the transformation that the web has had to the retail industry.

Even as e-commerce remains to be a relatively new concept that is emerging in the management of business operations, there is no doubt that e-commerce is creating a beneficial impact to organizations. One of the most notable impacts of e-commerce to businesses today is the significant reduction in the cost of operating business. “E-commerce helps businesses to save on the cost of using the traditional manual paper based system of information storage and retrieval by using modern electronic digital databases to create, process, store, distribute and retrieve information” (Turban, King &Chung, 1999, p79). Another “impact of e-commerce is the elimination of physical boundaries that previously fragmented markets into physical geographical locations” (Lallana, Quimbo & Whinston, 2007, p245). However, with the emergence of e-commerce, the traditional fragmented sections of physical geographical marketplaces have been replaced by a borderless international marketplace not limited or restricted by imaginary boundaries. This impact has invaluable benefits, particularly for start-ups and small and medium enterprises such as Hachich Tiles who are struggling with low sales by depending solely on the local area of Shepherds Bush in London, as it paves the way for such ventures to reach and penetrate into the global market relatively more easy.

Of significance importance as well, as it pertains to the impact of e-commerce, is the facilitation of ‘network production’. “Network production refers to outsourcing of production and business processes to partners located in different geographical locations through telecommunications networks” (Anita Rosen, 2000, p58). Another major impact of e-commerce is the elimination of hours of business that restrained business activities and transactions. As such, businesses are accessible to customers and suppliers for transactions 24 hours a day throughout the year. Above all, delivery of business processes and transactions has improved courtesy of e-commerce.

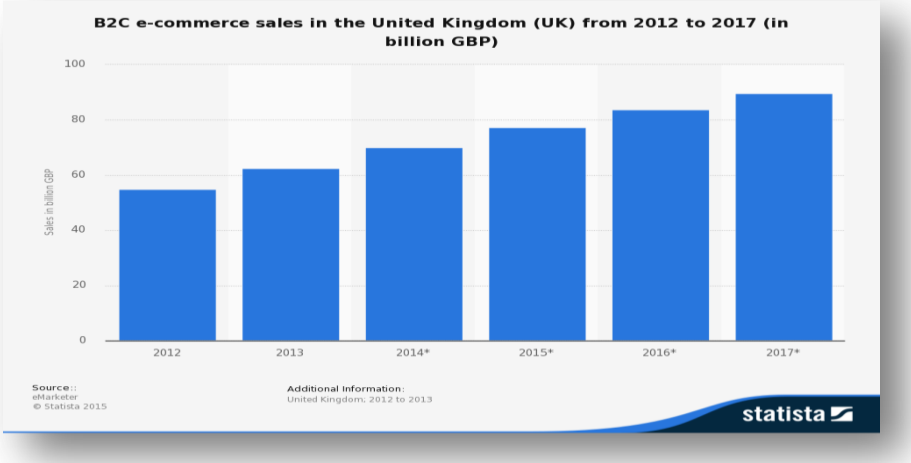


Figure 2 - Increase in the number of e-commerce sales for B2C from 2012 to 2017

3.3. Types of E-Commerce

There are various types of e-commerce and they are determined by the kind of transactions they conduct. They are categorized into the following classifications:

Business-to-Business (B2B): a type of e-commerce that conducts business transactions between and among other businesses. "B2B e-commerce creates relationships and facilitates dynamic interactions between business associates" (Andam, 2000, p106). To put it differently, the B2B e-commerce model is an electronic marketplace for businesses to sell and purchase products who then sell it to consumer at the retailer level.

Business-to-Consumer (B2C): a type of e-commerce, where the business transactions involve the business and the consumer as parties of the transactions. "In this model of e-commerce, businesses offer a wide range of products and services to the consumer for purchase and acquisition"(Turban, King &Chung, 1999, p178). Fundamentally, this model works by presenting consumers with a display of available products on the web page, from which they can select items and order.

Consumer-to-Consumer (C2C): a model of e-commerce that facilitates the carrying out of business transactions between and among consumers. "The C2C e-commerce empowers users by enabling them to sell products and services directly amongst themselves" (Zorayda Ruth, Andam, 2000, p299).

Consumer-to-Business (C2B): an electronic market that involves the selling of products and services by individuals to businesses. In a "C2B e-commerce model; the transactions are driven, and originate from the individuals who offer their products and services for sale to businesses" (Cronin, Mary, 1998, p128).

Business-to-Government (B2G): a type of e-commerce model defined by business transactions that involve private businesses and the public sector. In this electronic market, "the government plays the lead role by sourcing for needed products and services from businesses through the procurement, licensing and other government tender related transactions" (Timmers, 2000, p213).

3.4. Factors influencing E-commerce

The growth of e-commerce can be attributed to several significant and influential factors that have powered and fuelled its development. The key forces that have propelled e-commerce include the following factors:

3.4.1. Economic Forces among the most notable drivers of the influence of e-commerce are economic forces. "E-commerce has enhanced economic efficiency that has enabled business organizations to save significantly by cutting down on the overhead costs of business operations such as the cost of communication, the cost of marketing and advertising - to mention just but a few" (Mesenbourg, 2005, p95). In addition, improvement in the speed, reliability and cost efficiency of electronic transactions with partners and consumers has worked in favour of influencing e-commerce.

3.4.2. Technological Forces: one of the major factors influencing e-commerce is the technological advancement of information communication technology (ICT) systems. "To begin with, advances in the technology of digitizing content and the rising popularity of open source system technology have led to the convergence of communication into a single compatible platform across most of the communication devices, consequently making the setting up of communication infrastructure affordable besides enhancing its efficiency" (Zorayda Ruth, & Andam, 2000, p330).

3.4.3. Market Forces: The elimination of physical geographical markets which has been substituted and replaced with a new and accessible international market which is available and at the disposal of start-ups and small and medium enterprises regardless of the location of the business has worked in favour of e-commerce as a key force influencing its penetration in the business world. Equally important is that e-commerce can allow a business to "access their target audience with relative ease to promote, advertise, market and sell and purchase products and services" (Robert, Mitchell, 2012, p266).

3.4.4. E-commerce Payments Systems

Unlike the bricks and mortar type of businesses that use paper money as a means of making payment, the e-commerce business websites make use of electronic modes of payment when paying for products or service delivery. In essence, the electronic mode of payment refers to the payment of goods or services through the paperless monetary schemes or modes. "Making payments via the paperless electronic monetary methods has led to a significant reduction in the costs of labour, transactions and paperwork associated with the traditional means of payment using paper" (Kalakota & Whinston, 1996, p282). In addition to being user-friendly, electronic payment is less time consuming in comparison to the manual modes of payment.

Among the commonly used means of payment on e-commerce business platforms is the use of credit cards. Credit cards are small cards made out of plastic, which are fitted with a magnetic chip or strip that contains a unique identification number linked with the user's account. "The card is read via a card reader and allows the user to purchase goods and services on credit via the credit card issuer and pay back after a specified period" (European Commission, 2012, p189).

The debit card just like the credit card is used as a means of payment in e-commerce. Alongside credit cards, debit cards are also a common means of online payment. The debit card is also a small card made of plastic with a unique number identifier that is linked with the user's bank account. Unlike the credit card, that makes payment through credit, "the debit card makes payment by deducting the amount required for payment from the bank account should there be sufficient funds" (Business Software Alliance, 2001, p72). Another similar payment method is the smart card that stores money and deducts the required amount after conclusion of transactions. Equally popular in the payment of e-commerce transactions is the electronic Fund Transfer (EFT) that facilitates the transfer of money between bank accounts. "Transfer of funds can be accomplished using the ATM, computer or smart gadget devices" (FTI Consulting, 2011, p14).

3.4.5. Delivery

Once payments have been made to the business, the delivery process begins. To put it differently, the “order to delivery process begins as soon as the consumer expresses his or her intent to purchase a product (or a service) by submitting an order and concludes with the successful delivery of the purchased goods to the right consumer” (Oxford Economics, 2011, p62). While there are a set of mechanisms that work behind the delivery process as soon as a consumer places an order and completes their payment, it is best to recognize and acknowledge that: the process is not any different to the bricks and mortar business model. The distinguishing factor in transaction is that the consumers do not get to see the product purchased until delivered to them. Following a successful order placement, the quantity is allocated to the warehouse.

The order is processed in the warehouse where the necessary quality checks are conducted on the product after it is picked against the order list. The purchased product is then packed and labelled as the essential documents are printed and packed along with product for shipment. The consignment is handed over to the courier partners of the business who determine whether it will be sent by air, road or sea based on several factors such as weight, the urgency priority in delivery 1 day, 2 days, a week and so forth. “Once the consignment arrives at the consumer’s closest point/station of delivery, the package is once again submitted to an inspection, checking for any damage and the overall condition and quality of the package before it is handed over to the postman to complete the last mile delivery right to the consumer’s door step” (Nielsen, 1993, p308).

3.4.6. User Experience

Besides delivering the purchased goods to consumers, “e-commerce ought to provide users with a unique shopping experience” (European Commission, 2012, p100). As such, it is safe to say that user experience is important in e-commerce businesses and the transactions conducted as it pertains to the order placement process, product quality, and shipping and customer support services. An improved usability of an e-commerce website, gives business organizations leverage and competitive advantage. This is particularly true if the “e-commerce website offers the consumer with a much more satisfying shopping experience that goes beyond providing them with a catalogue of products to shop from” (Timmers, 2000, p154) users will recommend and continue using the website. For this reason it is highly recommended to build interactive ecommerce sites that engage with the consumer and offer a unique shopping.

Another important requirement for a unique user experience on business organizations e-commerce websites is navigation. It is extremely important for the users to be able to navigate through the different pages of the site with ease. It also imperative that the content should be incorporated creatively with the suitable interactivity and appropriate choice of media without leaving out a detailed description of the product, the product cost, delivery cost and any other necessary additional information for the transaction. Equally important is that the presentational dimensions and aspects of the e-commerce website are taken into account, paying attention to graphical design consideration. Above all, what remains of value to the user’s experience is that the “personal information they provide is protected and safeguarded and that it is also used only for the purpose for which it was collected” (Coward, Chris, 2002, p60).

3.4.7. The Security Systems of e-Commerce

Essential to the reliability of e-commerce is adequate security that can guarantee and ensure that the transactions carried out over the internet are secure and free from malicious attacks. Businesses are bound to lose consumer trust and confidence in e-commerce if the security systems are compromised in any way whatsoever. At this point, it is best to take into account that there are several requirements that are critical to the safe purchase or sale of a product or service, to the subsequent delivery of the product or service to the consumers. “Key, among the essential requirements for secure business transactions over the internet are confidentiality and integrity” (Business Software Alliance, 2001, p50). It is therefore imperative to ensure that personal user information is secured from interception while in transmission by making the information accessible only to authorized persons to maintain the integrity of the data.

Other key requirements that are critical to a safe and secure e-commerce transaction are availability and authenticity. Availability means that information be readily available at all times despite one’s physical location. On the other hand, “authenticity refers to the availability of mechanisms to verify the validity of the user prior to making personal information available” (Lallana, Quimbo & Whinston, 2007). Of similar importance is auditability, which requires that, the data is stored in a method that checks for integrity requirements.

The major measure that can be adopted to ensure the safety of e-commerce transactions include: “encryption, which protects data under transmission using a security code, generated by the sender and can only be decrypted by the designated receiver using a different or the same code” (European Commission, 2012). Digital signatures are also an effective measure to guarantee the authenticity of data through encryption security password, while security certificates are useful in the verification and identification of websites or users.

3.5. Relevant Technologies

There are many relevant technologies which can aid in the development of a fully functional e-commerce website. Relevant technologies cover the programming languages used, those of which effect the implementation and design of the website.

E-commerce websites can be generated and maintained by different platforms such as Shopify, Magento, WIX and many more. These platforms deal with issues mentioned in the factors influencing e-commerce above.

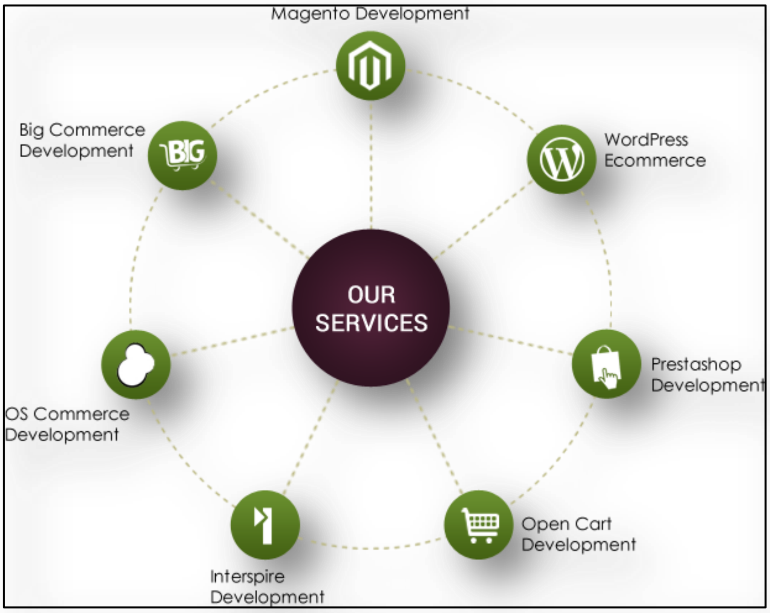


Figure 3 - The different types of platforms used to develop e-commerce websites.

3.5.1. HTML 5

HTML 5 is the latest Hypertext Mark-up Language which works alongside CCS and JavaScript. These 3 languages provide the structure, design and some sort of functionality to websites. The previous HTML4 differentiates from HTML5 in many different ways. HTML5 gives the ability to section web pages into articles, asides, section, summary, video, track, time and many more. Therefore, one’s website will improve in consistency, accessibility, semantics and mutuality.

3.5.2. CSS3

What is CSS?

CSS is the language for describing the presentation of web pages, including colours, layout, and fonts (W3.org, 2016). It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based mark-up language (W3.org, 2016).

3.5.3. PHP

PHP: Hypertext Pre-processor

PHP is a widely-used, open source scripting language (W3schools.com, 2016). PHP is able to add, delete and modify data in your database (phpMyAdmin). PHP is also able to send and receive cookies which will aid the development of an e-commerce website when it comes to creating a shopping cart.

Advantages:

- Runs on various platforms (Linux, Windows, MAC OS, Unix)
- Compatible with most current servers
- Free
- Runs efficiently once implemented correctly

3.5.4. JavaScript

As mentioned by Stephen Chapman in his article, JavaScript is a programming language used to make web pages interactive (Chapman, 2016). JavaScript is built into all major web browsers.

Advantages:

- Executed on the clients side not administrators
- Easy language
- Extends the functionality of webpages

Disadvantages

- Security issues
- Rendering varies (layout engines may differ)

3.5.5. Databases and SQL

Often abbreviated *DB,* a database is basically a collection of information organized in such a way that a computer program can quickly select desired pieces of data (Beal, 2016).

The language which allows access into these databases is SQL, which stands for Structure Query Language. Without SQL it's impossible to retrieve, insert, update, delete and create new data in your database. The communication between a database and a web browser is crucial when it comes to the development of e-commerce websites. The ability to store product information and client information; for example it is essential to aid with the completion of purchasing products. If there are no stored products in the database, customers will not be able to view or purchase anything.

3.6. Existing Websites

To build the perfect e-commerce website and ensure its global success, research was conducted on the following companies; Topps Tiles, B&Q and Screw Fix. These are a few of the companies which are currently dominating the hardware/tiles market.

To get information on what existing websites had, extra attention was spent on the structure, content present and features such as navigation and interaction. Knowledgeable in what offers a good usability experience, analysis of the competitor’s websites was conducted through personal usability tests. Interaction was made with the website in order to complete several tasks, the strengths and weaknesses encountered whilst performing the tasks was documented.

The Harvard industry dominator, ‘B&Q’ which is owned by Kingfisher gave some good impressions based on the colour; design and layout of the website. The positives were numerous but the main key points where the websites responsiveness and the image to ratio aspect when resizing the browser. When searching for a product during the usability tests, the navigation bar allowed easy access into different categories. Other information such as that the navigation-bar showed categories which offers the user an easier navigation. The ability to refine products based on colour, material, size and price made it simple for me to find products which matched the task criteria of the usability test. The ability to add the products to the basket looked like a necessity as all e-commerce websites need it. The navigation bar also had a ‘Thebasket’ option which directs the user to their current shopping basket. Store locators and customer support were also present to give the users more information. When selecting a product, B&Q was found to offer both delivery and in-store collection depending on what suited the customer. This is a great way to meet customer needs and ensure customers remain loyal. B&Q’s website overall was very useful when it came to product information and offers etc.

Despite all the findings above, they indicate that the website had too many products and was too informative. The ability to concentrate on one product at a time was difficult as there was too much information about it.

Characteristics of most hardware selling websites are:

- Searching for products
- Breakdown of different products through the use of a category navigation bar
- Locating of the stores near you
- Shopping basket where product quantity can be changed or removed
- Promotions and deals on the homepage
- About us page providing information of the business and the services it offers

4. Analysis and Methodology

In the analysis and methodology chapter of the report, a detailed discussion on the project methodology, followed by the project analysis will be offered. The analysis will consist of the user requirements, stakeholders, and their analysis, followed by SWOT and MoSCoW analysis. Using the mentioned tools will then allow me to identify what best procedures should be put in practice to ensure the development of an e-commerce website to fulfil the client's needs within the agree timeframe.

4.1. DSDM

Different projects have numerous different problems which can affect the outcome of the products in both the long run and the short run. In regards to the building of an e-commerce website, most projects will need to produce the same outcome, a functional website without the need to sacrifice major resources. Problems which can possibly affect the project are failing to deliver the website to the client on time, frequent change of the scope, lack of PHP experience, and going over the clients predicted budget. Therefore, when selecting the right project methodology, the project outcome was found to be determined by the factors mentioned above.

An agreement between me and the client was made leading to the selection of the dynamic system development method (DSDM) to ensure that the problems mentioned above will not occur. DSDM is an agile approach which brings in the agility and flexibility needed to successfully implement software. DSDM is an iterative approach which heavily involves user contribution to ensure the delivery of the project on time and in budget which is essential to the client. As the clients problems are rapidly growing and sales are decreasing, the e-commerce website has to be launched as soon as possible to fix the problems. If the client is to remain in the current tiles industry, action to improve his sales has to be taken as soon as possible with the least cost possible. DSDM includes the use of several practices such as, facilitated workshops, MoSCoW prioritisation and time boxes. All these practises aid in the delivery of the project on time and on

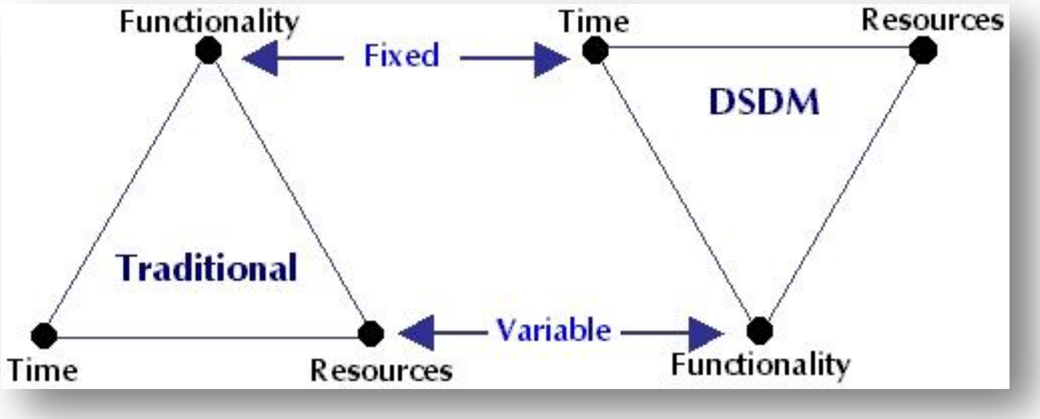


Figure 4- Showing a comparison of how functionality in DSDM is flexible in comparison to a traditional development method

Table 1- How DSDM deals with the clients business problems

Problems with projects	How DSDM deals with issues
Poor communication	DSDM has facilitated workshops where stakeholder interaction takes place instead of large chunks of written text, passed from one stakeholder to the other. Workshops all tend to have a clear objective and are led by a neutral.
Delivered product doesn't meet client needs	Drifting away from the project scope happens a lot. DSDM ensures that the stakeholders are part of the development team which encourages collaboration and communication to ensure the team is on track.
Lack of testing	Delivering a product with many errors at the time of the product deadline can become very frustrating and can cause a late product release or product failure. Therefore more time is needed to fix the errors which will inevitably delay the launch of the website. DSDM uses time boxes which write tests before a feature is implemented to ensure its correctness.
Late delivery of product	DSMD focus on meeting short-term goals thought in increments found in time boxes. Less important features can be left out instead of extending the deadline.
Going over budget	Through testing, delivery of important requirements and constant communication between the stakeholders and developers, the chance of project failure would be low. This therefore will prevent the need of spending money on fixing errors and the cost of releasing a product late.

DSDM ensures that the businesses costs are constant and that a high quality product will be delivered at the end. This is done by constantly changing the business requirements with approval by all major stakeholders. For this project in particular, DSDM will help with maintaining costs and delivering the project on time.

4.2. DSDM principles:

- Focus on the business needs – MoSCoW prioritisation
- Deliver on time – time box allows the focus of priorities and hits deadlines
- Collaborate – all team members have the same power to take decisions quickly
- Never compromise quality – ensure that the project quality chosen from the start is followed through – continuous testing to ensure error free product.
- Build incrementally from firm foundation – stakeholders part of the development team
- Develop iteratively –stakeholder feedback is used at each iteration
- Communicate continuously and clearly – workshops and stand-ups to ensure roles are clearly defined
- Demonstrate control – plans are visible to all members and tracking and reporting is used
- (Business Agile, 2000)

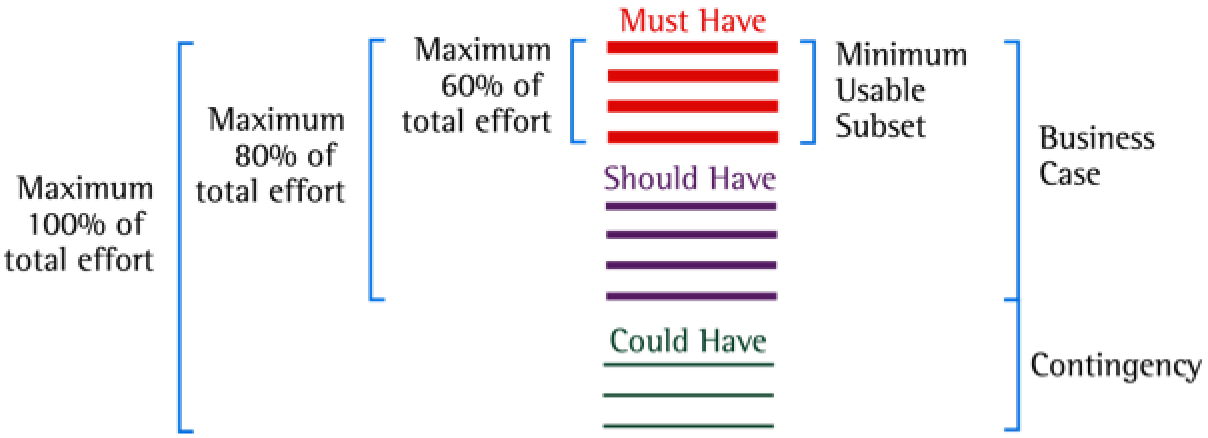

Must Have
Should Have
Could Have
Maximum 60% of total effort
Maximum 80% of total effort
Maximum 100% of total effort
Minimum Usable Subset
Business Case
Contingency

Figure 5 - Break down of the amount of work that will go into each requirement based on its position in the MoSCoW scheme

Figure 5 above gives a break down into how the implementation of these projects requirements will be grouped to ensure that they are delivered on time, with a fixed budget and of high quality. Most of the projects effort will be focused on the ‘must have’ requirements where the least of the time will be used to implement the ‘could haves’. Through the weekly meetings with the client, implementing the ‘must have’ requirements to the best of quality, within the predicted time frame and cost was agreed. If the project is delivered with all the ‘must haves’ correctly implemented before the project deadline, then the implementation of the ‘should haves and could haves’ will occur.

4.3. Functional and Non-Functional Requirements

4.3.1. Functional Requirements

Functional requirements deal with what the system should do or provide for the users (Sqa.org.uk, 2007).

End user requirements (Customer):

C1. Create account
C2. Login with an account
C3. Browse product catalogue
C4. Choose product category
C5. Add products to basket
C6. View the basket
C7. Edit basket
C8. View basket total
C9. Checkout
C10. View order details
C11. Delete order
C12. Request password reset
C13. Receive create account email confirmation
C14. View company information
C15. Message seller for stock enquiry
C16. Request potential delivery time of products
C17. Navigate through the website easily

Admin Requirements:

A1. Login
A2. Logout
A3. Add & remove products from catalogue
A4. Edit product information
A5. Manage customer account details
A6. Delete customer account

4.3.2. Non-Functional requirements

Non-functional requirements detail constraints, targets or control mechanisms for the new system (Sqa.org.uk, 2007).

NFR1. Access the same basket through a different session
NFR2. Address the number of transaction per day
NFR3. Online chat to provide fast communication between seller and admin
NFR4. Ensure that the pages load in under 1 second
NFR5. Provide an easy method to change user details
NFR6. Available on android/IOS stores as a mobile application
NFR7. Handle many payments from same account at once
NFR8. System must be available 24 hours a day
NFR9. Accessible for disabled or less abled users

4.4. MoSCoW Analysis

This is a prioritisation technique which is used to place importance on the requirements. The requirements are split into the four categories, must, should, could and won't have.

Table 2 - MoSCoW table containing all functional and non-functional requirements

MoSCoW Prioritisation	
Must Have	Requirements that must be implemented before the launch of th website.
	C1. Create account C2. Login with an account C3. Browse product catalogue C4. Choose product category C5. Add products to basket C6. View the basket C7. Edit basket C8. View Basket Total C9. Checkout C10. View order details C11. Delete order A1. Login A2. Logout A3. Add & remove products from catalogue A4. Edit product information A5. Manage customer account details A6. Delete customer account
Should Have	Requirements that should be implemented but not that important
	C12. Request password reset C13. Receive create account email confirmation C14. View company information C15. Message seller for stock enquiry
Could Have	Potential of implementing the requirements if there is more time
	C17. Navigate through the website easily C16. Request potential delivery time of products
Won't Have	Will not be included in the website
	Availability on IOS/Android as a mobile application A live instant messenger

Throughout the delivery of the e-commerce website, it is expected for the priority of the requirements to change. Some must have requirements could potentially drop to becoming should or could have requirements due to the lack of time and budget. On the other hand if the project is completed before schedule and to a high standard, some could have and should haves will be implemented.

4.5. SWOT

SWOT is an analysis tool which looks at the internal strengths and weaknesses of the business as well as the external opportunities and threats. SWOT analysis allows the business to understand how its strengths can potentially differentiate itself from its competition. Alongside the strengths, SWOT can help the management of potential threats and weaknesses.

Table 3 - SWOT analysis showing the internal and external business strengths and weaknesses

	Helpful	Harmful
Internal	**Strengths**	**Weaknesses**
	Unique Selling Point Lower Operation costs Recently built website Available on all browsers Integrating social media with e-commerce for free business marketing.	Unexpected costs No face to face selling Communication between seller and business Only one developer Need for an internet access device
External	**Opportunities**	**Threats**
	Growth of the e-commerce market Company sales to increase Ability to take advantage of technological advancements 24/7 shopping experience	Competition Security and Privacy Tax and Legislation Malicious attacks Weather changes

4.5.1. Strengths

Unique selling point – There is nobody in the current tiling industry selling tiles imported from Jordan and Turkey giving a competitive advantage. Being the only competition which has these tiles, ability to add value to the product is easy. Hence charging a higher price is easy without the worry of losing customers.

Lower Operation costs – With the current business problem not reaching targets sales and low profits, low operating costs will decrease the break-even point needed to make a profit. Costs of operating the website and maintaining it are relatively low.

Recently built website – Latest technology will give the website a new, modern and easy to use look and feel ensuring customer's satisfaction.

Available on all browsers – Websites are available to be accessed through many different platforms making the products available to all internet users. This gives the advantage or reaching different potential clients.

Online payments – Payments can be taken online securely through PayPal or other payment processing options to ensure that the customer's money and business transaction is safe. Secure payments will strengthen the relationship between the customer and business.

Integrating social media with e-commerce for free business marketing can be a method of advertising and creating brand awareness. Customers can also potentially gain an insight into the businesses objectives through a Facebook group or a twitter page giving customer live updates of their delivery status.

4.5.2. Weaknesses

Unexpected costs – Costs such as damaged products, unnecessary refunds, shipping costs and server maintenance all increase the business costs. If the business costs are constantly changing, the break-even point will fluctuate meaning it will cost more to make the same profit that one was making before, or even worse not making a loss.

No face to face selling – Customers complain that online websites lack the personal touch and relationship that is developed in a retail store. Helping a customer look for a size, serving at the cashier, thanking, and building a personal relationship with customers will be left out.

Communication between seller and business – Limited communication between the seller and the business can cause complications in the delivery of the correct products if mistakes or errors are made online. No direct employee available to help solve the problem.

Only one developer – Project can potentially launch after agreed time with client, quality of the programming and website layout/design can be very poor. Developer experience will determine the websites overall functionality.

Need for an internet access device – Some day-to-day shoppers have still not adapted to the 'online shopping' market and prefer to go in-store to make purchases. Despite the internet being such a big part of everyone's day to day lives some customers are still not able to access the internet, hence meaning they will not be able to access the business website.

4.5.3. Opportunities

Growth of e-commerce market – E-commerce sales are continuing to grow in the UK meaning there is a high increase in user spending for this market and online products will always be on demand.

Company sales to increase with launch of E-commerce website – Opportunity of the increase in sales as user's customers will be purchasing products online. Company can focus on making sales online and close store saving costs potentially lowering price of the products.

24/7 shopping experience – The website can be accessible 24/7 and customers can make purchases on products when regular stores are closed e.g. midnight.
Ability to take advantage of technological advancements will ensure that customers receive the best shopping experience online with no errors and faults.

4.5.4. Threats

Competition – Topps Tiles and B&Q are dominating the current tiling market; customer loyalty with competitors has already been established.
Security and Privacy – Customers are afraid that their data can end up in the wrong hands, with doing so it can impair the relationship between the customer and the business. Business reputation can also be ruined pushing customers away decreasing sales in the long term.

Tax and Legislation – UK has left the European Union so taxes must be paid on all imported and exported products. Products coming in from Turkey/Jordan have a high tariff cost.

Malicious attacks – Hackers and viruses can cause the business website to crash, leaving customer information at risk. This information will include their email address, age, credit card details and more.

Weather changes – Floods and lightening spells can potentially cause damage to servers where business and customer information is stored. This can impair the reputation of the business and cause the business to fail.

4.6. Stakeholder Analysis

Store owner (Admin):

The store owner is currently in need of an e-commerce website to amend his current financial state and to prevent him from closing shop. The client wants a basic website to facilitate the exchange of his goods for money and to ensure that customers are always satisfied.

Customers:

The customer is who the system will be designed and developed for, along with the store owner, who must edit content. Alongside with the main aim of the project, increasing sales and fixing cash flow problems. The end users are always looking for the lowest prices, the easiest websites and those with the best products. As mentioned in the literature review, customers show a great interest into quickly finding the products they want, making a secure payment, and having the most enjoyable shopping experience possible.

Employees (Admin):

Hachich Tiles hire small numbers of people to deal with basic customer queries. Jobs include delivering products, providing customer support and to ensure customer satisfaction is met at all times. One or two of the employees will have access to the admin panel of the e-commerce website. Being highly powered they will be able to make the changes above. The selected admin employees want to be able to make changes to customer account, and make changes to the products which are available or out of stock. The selected employees want to be able to access the e-commerce website and make changes to user information to ensure the correctness of the orders and to prevent any financial errors.

Suppliers:

As sales have decreased, the store owner has been delivering a small number of products to the store. The supplier's interests are in how successful the e-commerce website will be and its potential in driving Hachich Tiles sales up. The more products sold on the e-commerce website, then the more products he would have to supply from abroad. With a higher number of sales, the supplier can increase his revenue same as for the store owner.

4.7. Use case diagram (Joint customer and admin)

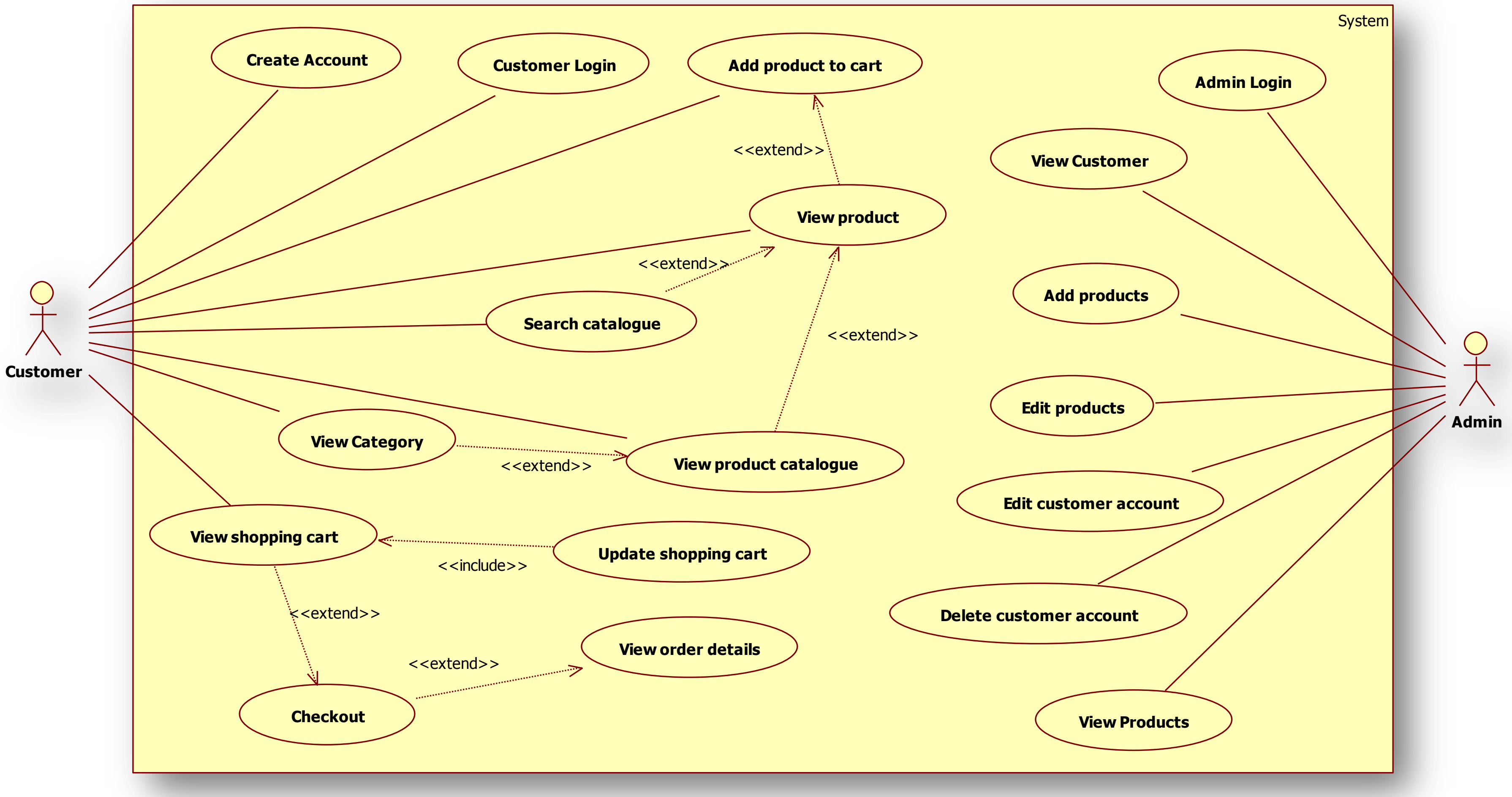


Figure 6 - Admin and Customer use case diagram

5. Design

This section of the report will house information on the design of both the front-end and the back-end of the e-commerce website. A series of sequence diagrams will be drawn from use case descriptions following a detailed activity diagram which will be attempting to show the operation of the full systems, front-end and back-end. The database design, data dictionary, system architecture and design narrative will also be included in reference to the back-end. A series of wireframes and prototypes will also be included to give a clear vision of what the developer intended to build.

5.1. Use Case Descriptions

Table 4- Table containing information on the use case text descriptions

Use case ID	Use case name	Primary Actor	Scope	Complexity (Low-med-high)	Priority (1-10) 10 being the highest
UC01	Create account	Customer	In	Low	10
UC02	Add to cart	Customer	In	Low	9
UC03	Send message	Customer	In	Low	6
UC04	View products	Customer	In	Low	8
UC05	Add product	Admin	In	Med	10

Table 5 - Create account use case text description

Use case name	Create account
Use case ID	UC01
Brief description	The use case descriptions will give information on a user willing to create an account
Primary actor	Customer
Frequency of use	100% of those who do not have an account and wish to make a purchase.
Triggers	User would want to make an account to make a purchase.
Preconditions	Customer will have access to the internet. Customer will be on the website.
Main flow	1. Customer will click the login button 2. The website would redirect the customer to the login page 3. The customer will not have an account so will click on the create account page. 4. The website loads the create account page, showing the fields required for entry. 5. Customer fills in the fields. 6. Customer clicks create account.
Alternative flows	AF1. Request forgot password if email already belongs to an account. AF2. Customer has added items to the basket and is attempting to checkout; customers then redirected to login page where they can create account.
Exception flows	EF1. User is already logged in
Post conditions	Success: The customer will receive a confirmation email with a link, upon activating account the customer will be redirected and logged in. Customer can then make purchases, send messages and more. Failure: Customer account is not created.

More use case text descriptions can be found in Appendix A

5.2. Activity Diagram

5.2.1. Search product/browsing simplified

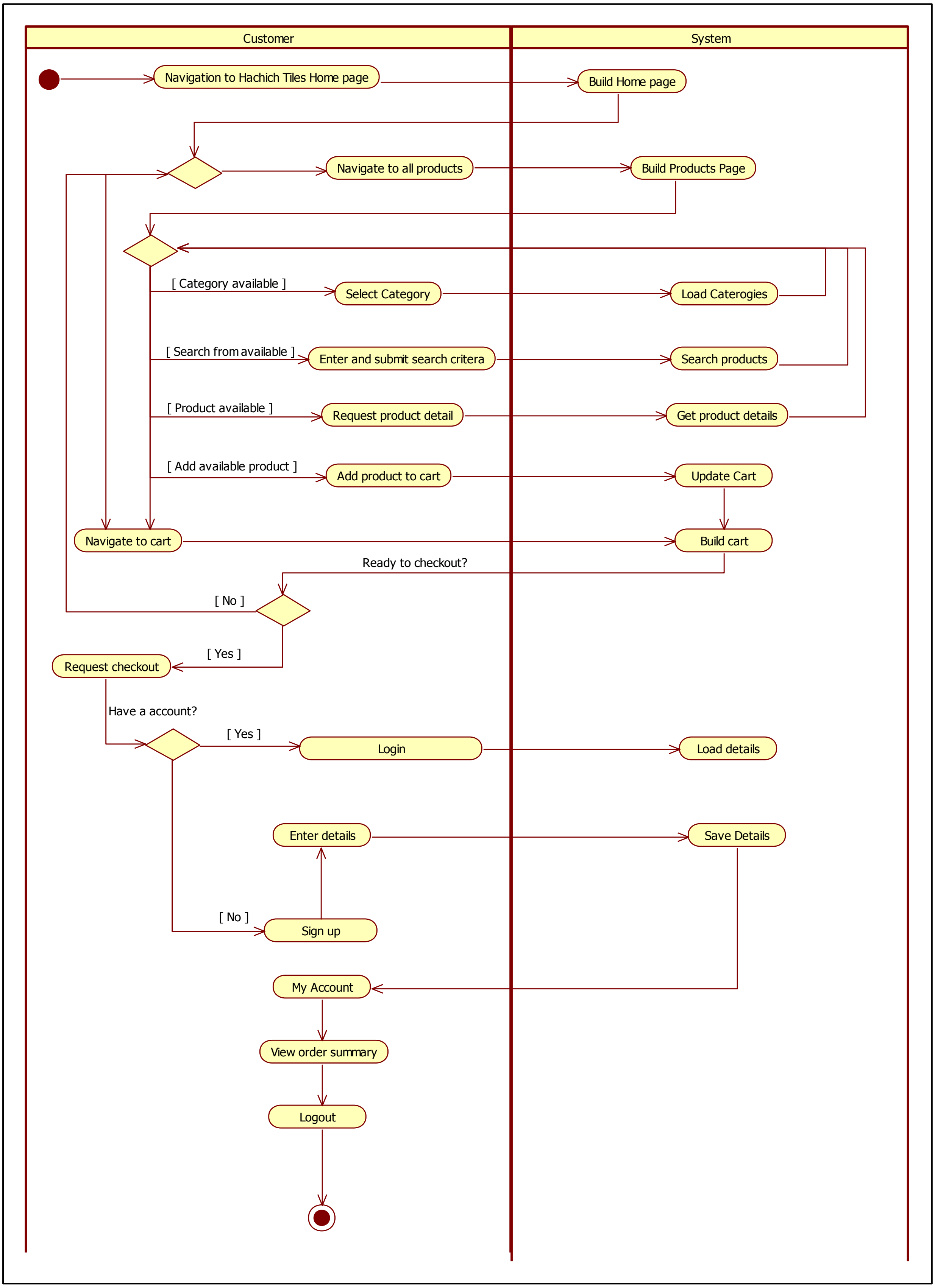


Figure 7 - Activity diagram for overall website

5.2.2. Login

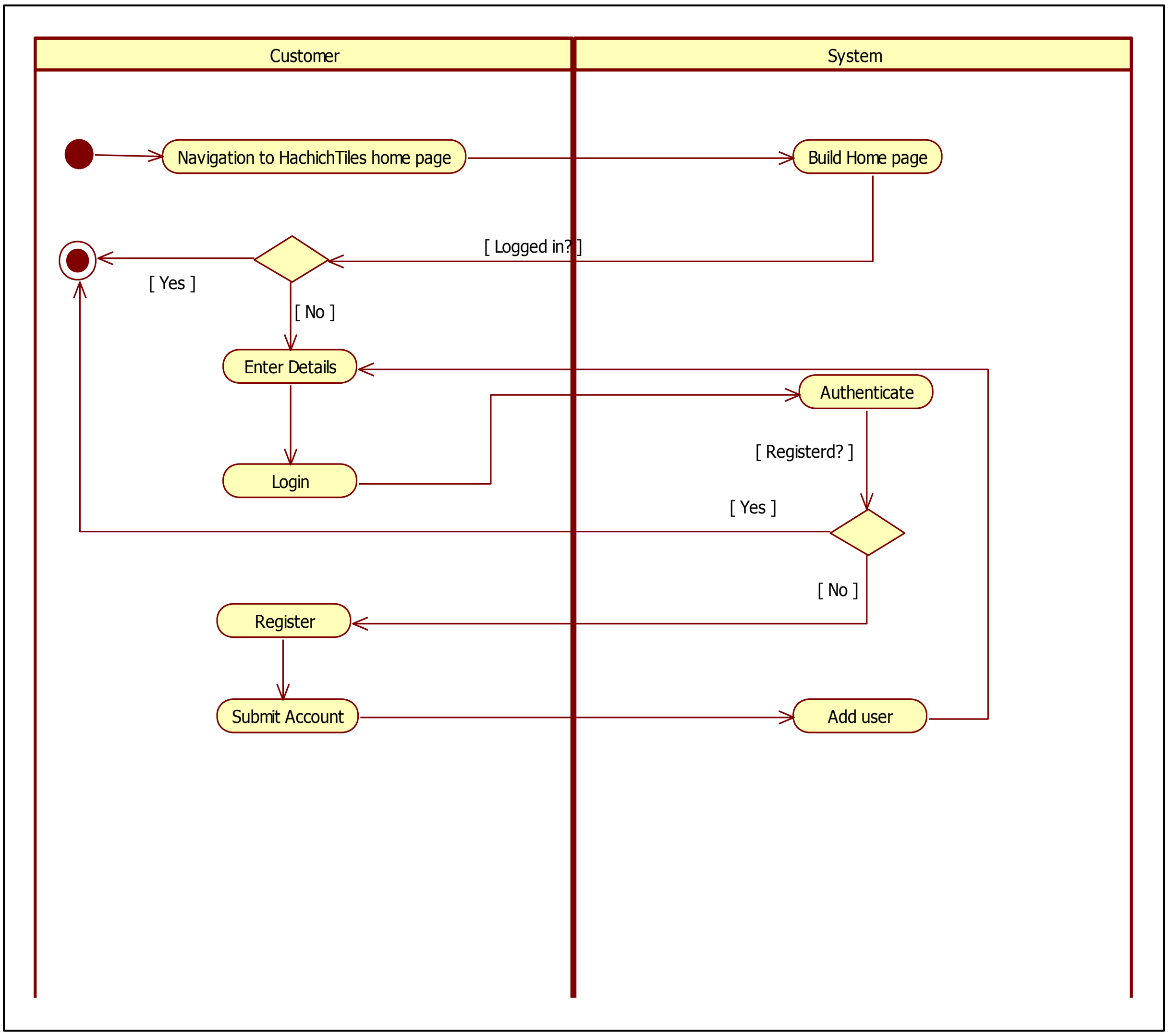


Figure 8 - Activity Diagram for the login procedure

5.2.3. Add to Cart

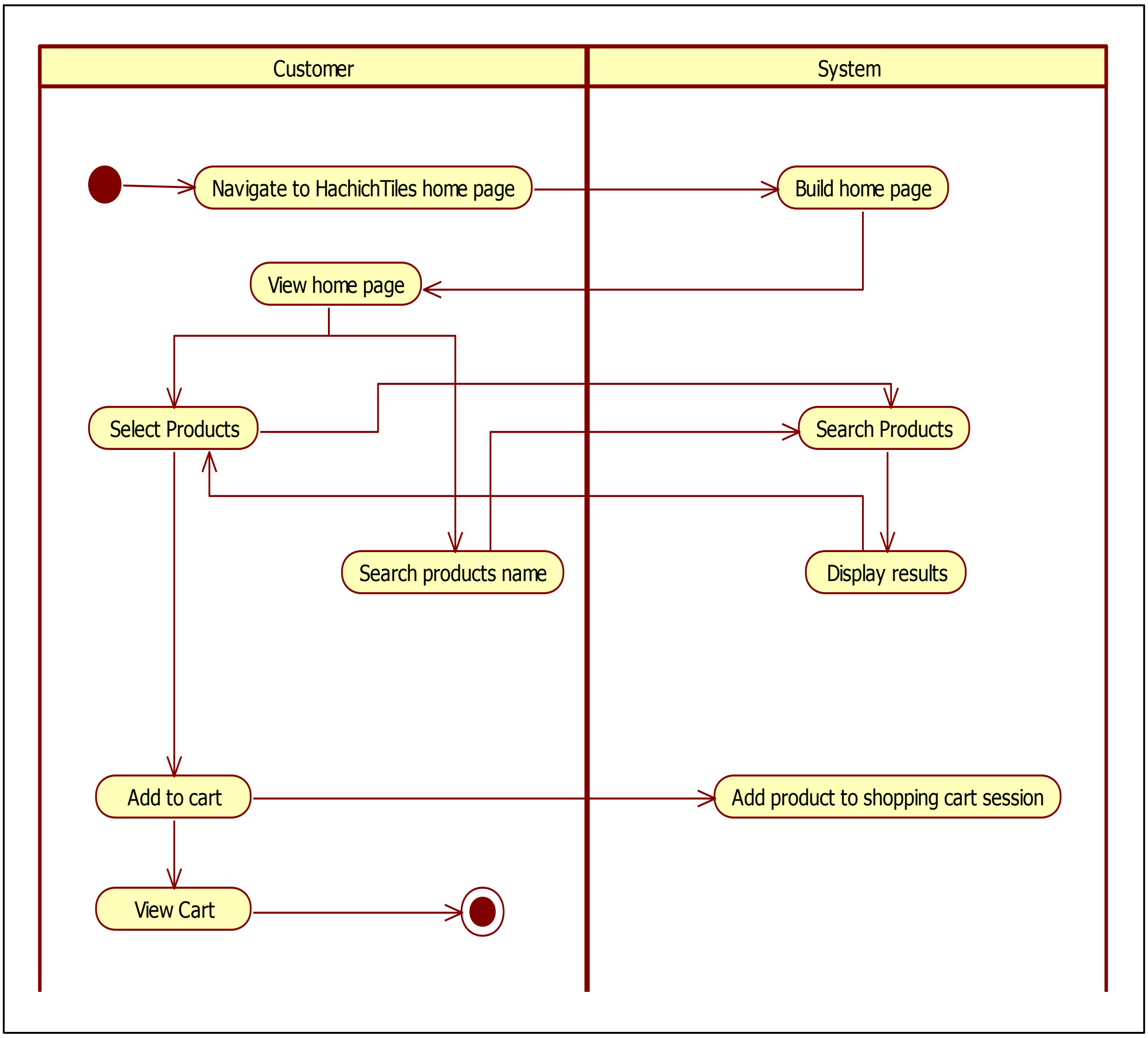


Figure 9 - Activity diagram for adding a product to the shopping cart

5.2.4. Create Account

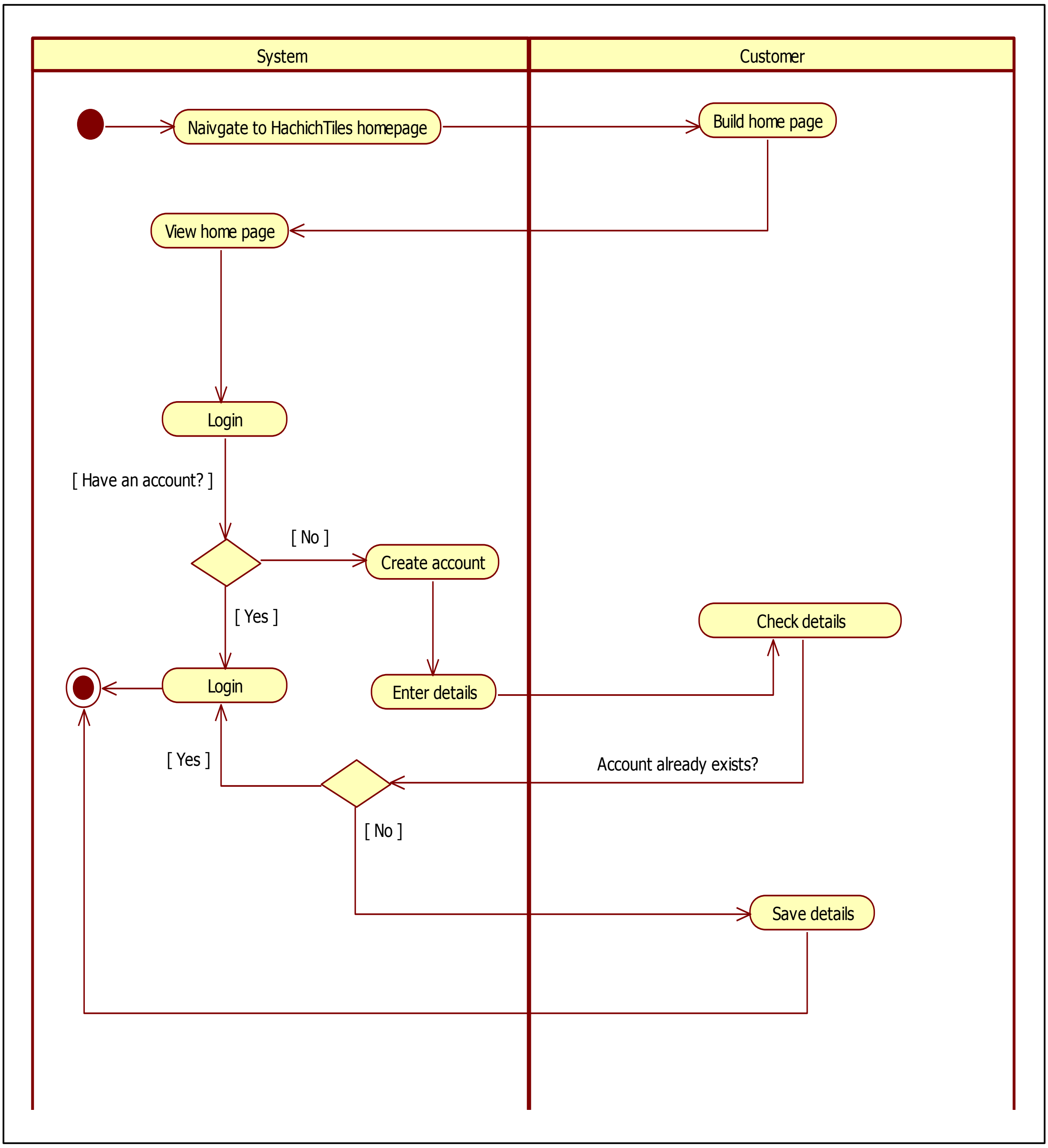


Figure 10 - Activity diagram for creating an account

5.3. Sequence Diagram

5.3.1. Create Account

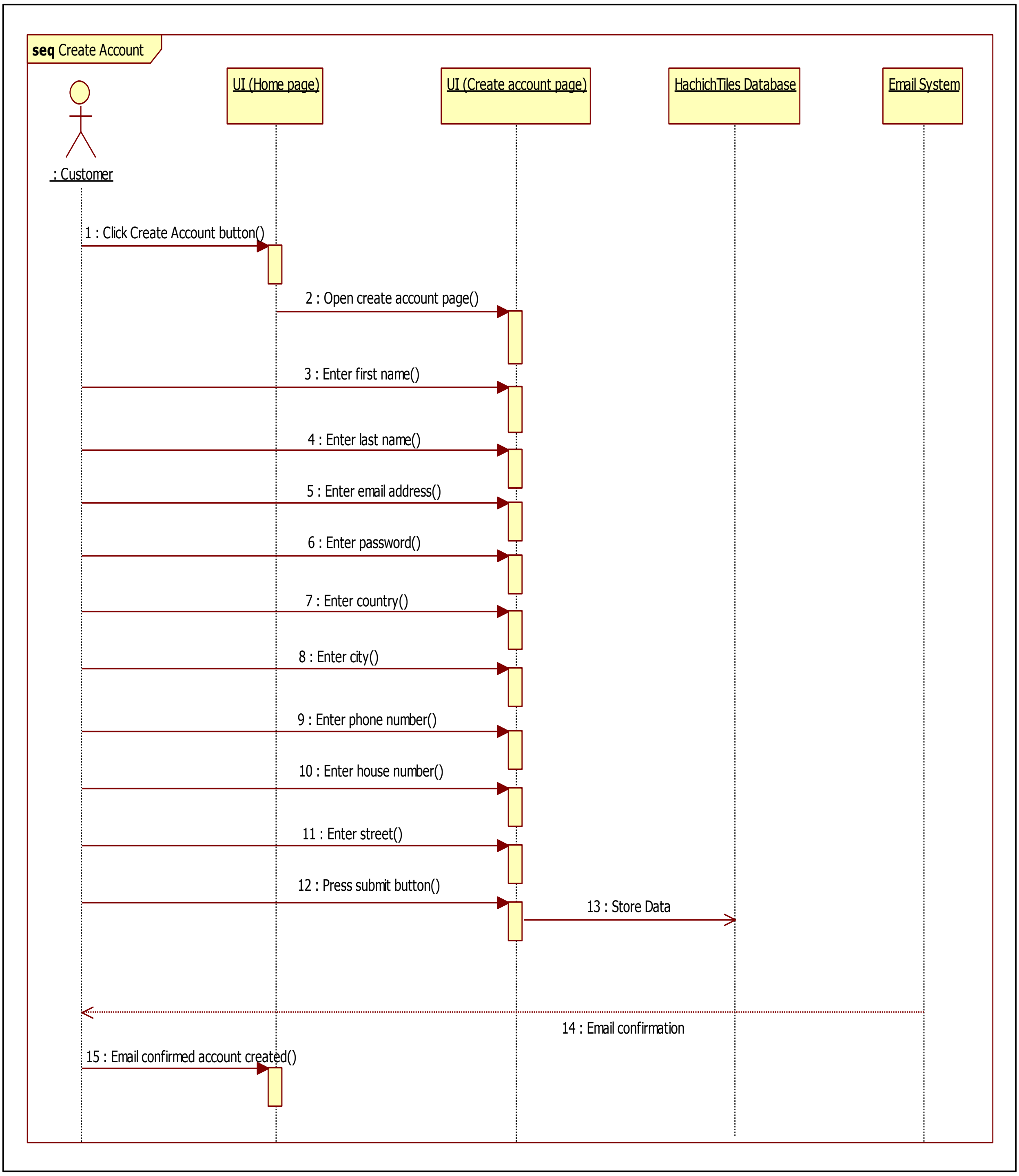


Figure 11 - Sequence diagram for creating an account

5.3.2. Login

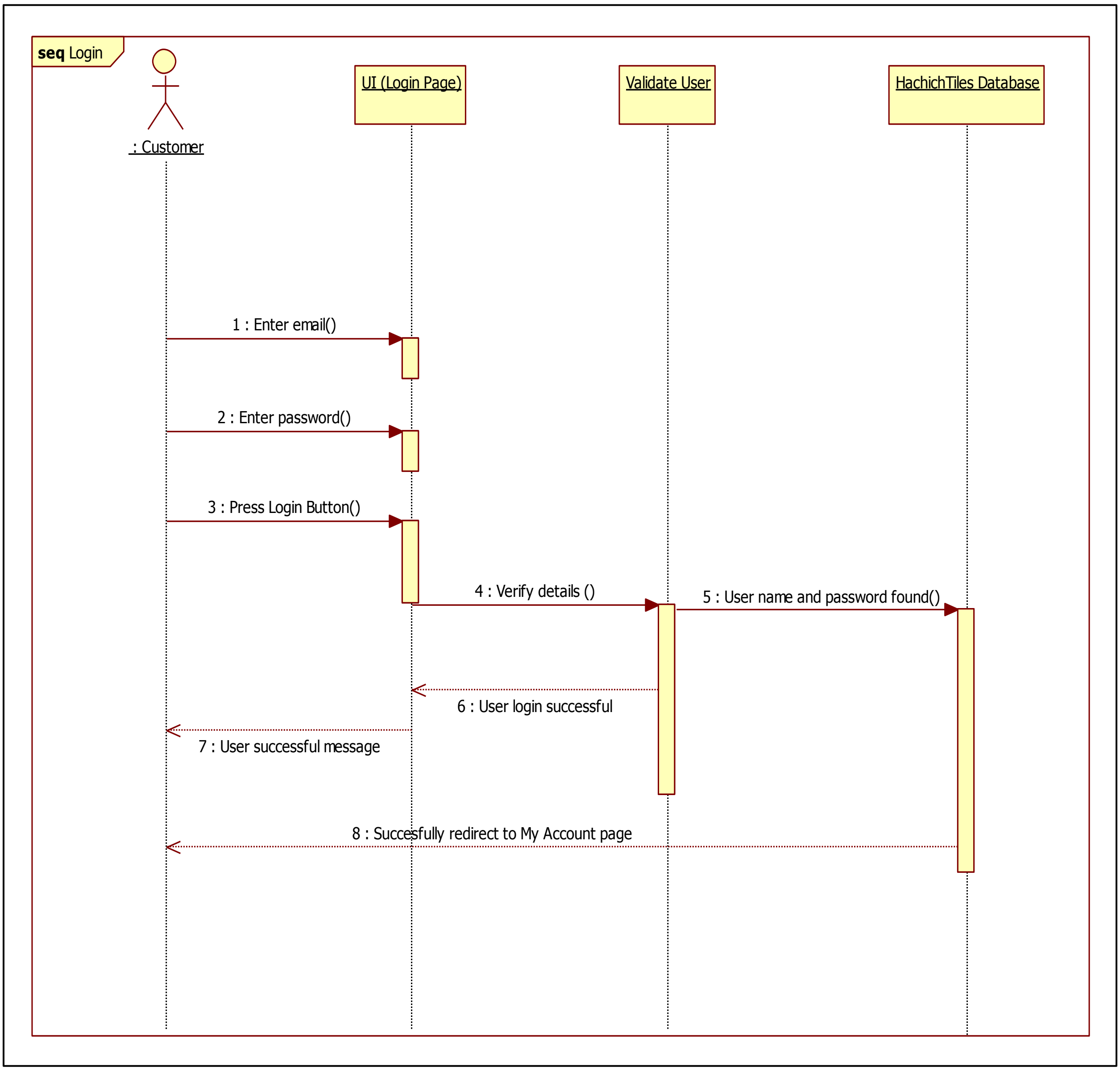


Figure 12 - Sequence diagram for when logging into the website

5.3.3. Add to Cart

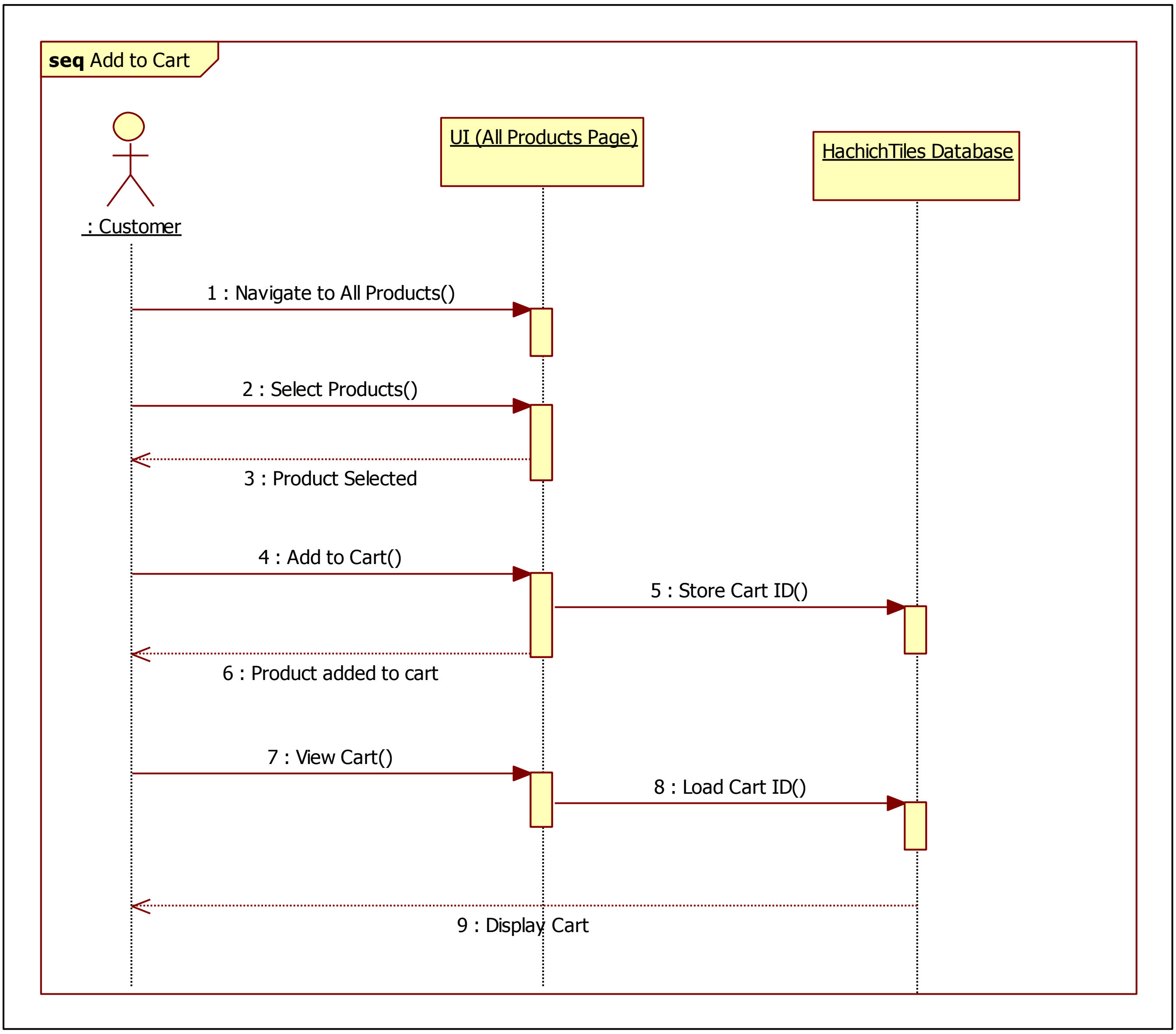
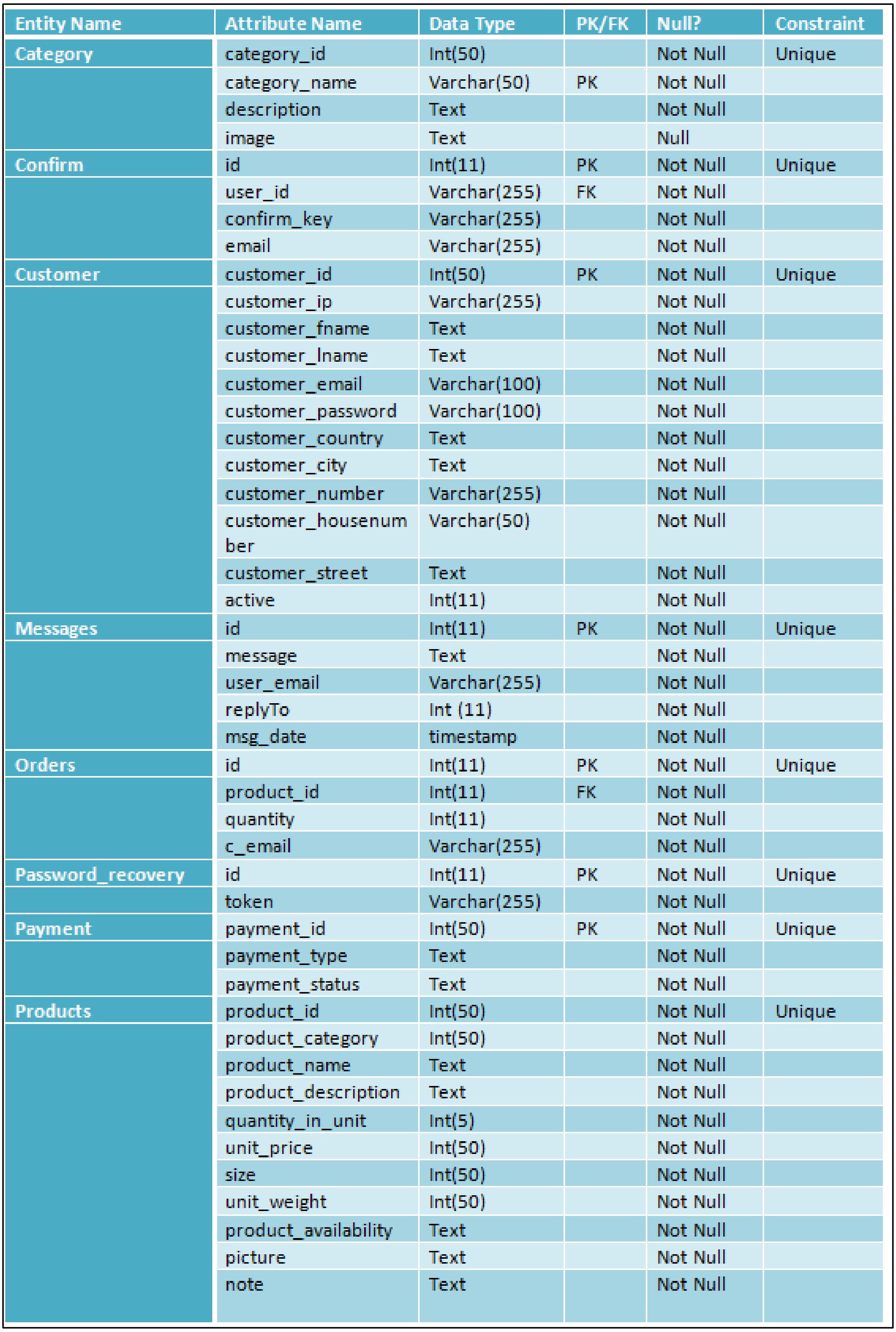


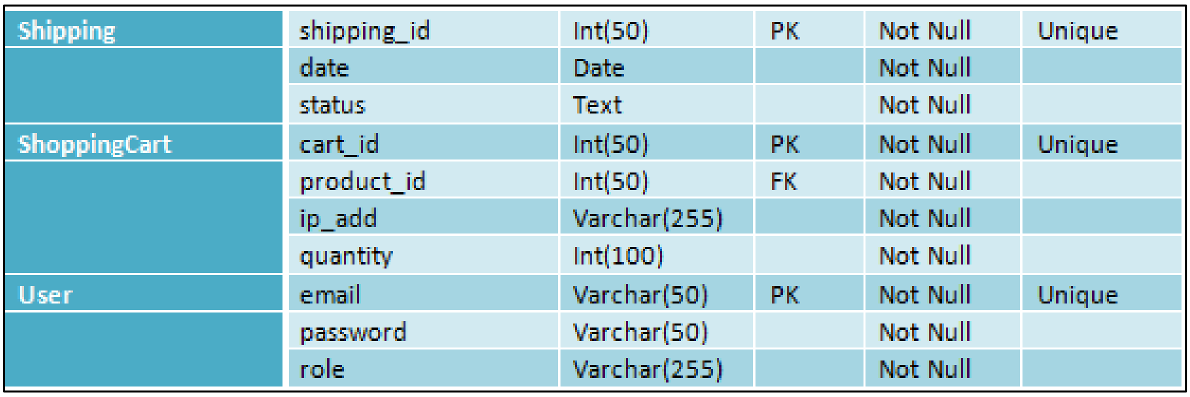
Figure 13 - Sequence diagram for adding a product to the cart

All low fidelity wireframes can be found in appendix 3

5.4. Data Dictionary



Entity Name	Attribute Name	Data Type	PK/FK	Null?	Constraint
Category	category_id	Int(50)		Not Null	Unique
	category_name	Varchar(50)	PK	Not Null	
	description	Text		Not Null	
	image	Text		Null	
Confirm	id	Int(11)	PK	Not Null	Unique
	user_id	Varchar(255)	FK	Not Null	
	confirm_key	Varchar(255)		Not Null	
	email	Varchar(255)		Not Null	
Customer	customer_id	Int(50)	PK	Not Null	Unique
	customer_ip	Varchar(255)		Not Null	
	customer_fname	Text		Not Null	
	customer_lname	Text		Not Null	
	customer_email	Varchar(100)		Not Null	
	customer_password	Varchar(100)		Not Null	
	customer_country	Text		Not Null	
	customer_city	Text		Not Null	
	customer_number	Varchar(255)		Not Null	
	customer_housenumber	Varchar(50)		Not Null	
	customer_street	Text		Not Null	
	active	Int(11)		Not Null	
Messages	id	Int(11)	PK	Not Null	Unique
	message	Text		Not Null	
	user_email	Varchar(255)		Not Null	
	replyTo	Int (11)		Not Null	
	msg_date	timestamp		Not Null	
Orders	id	Int(11)	PK	Not Null	Unique
	product_id	Int(11)	FK	Not Null	
	quantity	Int(11)		Not Null	
	c_email	Varchar(255)		Not Null	
Password_recovery	id	Int(11)	PK	Not Null	Unique
	token	Varchar(255)		Not Null	
Payment	payment_id	Int(50)	PK	Not Null	Unique
	payment_type	Text		Not Null	
	payment_status	Text		Not Null	
Products	product_id	Int(50)		Not Null	Unique
	product_category	Int(50)		Not Null	
	product_name	Text		Not Null	
	product_description	Text		Not Null	
	quantity_in_unit	Int(5)		Not Null	
	unit_price	Int(50)		Not Null	
	size	Int(50)		Not Null	
	unit_weight	Int(50)		Not Null	
	product_availability	Text		Not Null	
	picture	Text		Not Null	
	note	Text		Not Null	



Shipping	shipping_id	Int(50)	PK	Not Null	Unique
	date	Date		Not Null	
	status	Text		Not Null	
ShoppingCart	cart_id	Int(50)	PK	Not Null	Unique
	product_id	Int(50)	FK	Not Null	
	ip_add	Varchar(255)		Not Null	
	quantity	Int(100)		Not Null	
User	email	Varchar(50)	PK	Not Null	Unique
	password	Varchar(50)		Not Null	
	role	Varchar(255)		Not Null	

Figure 14 - Database dictionary including the attribute name along with the data type and length, constraint type, primary key or foreign key and if null.

5.5. Database design

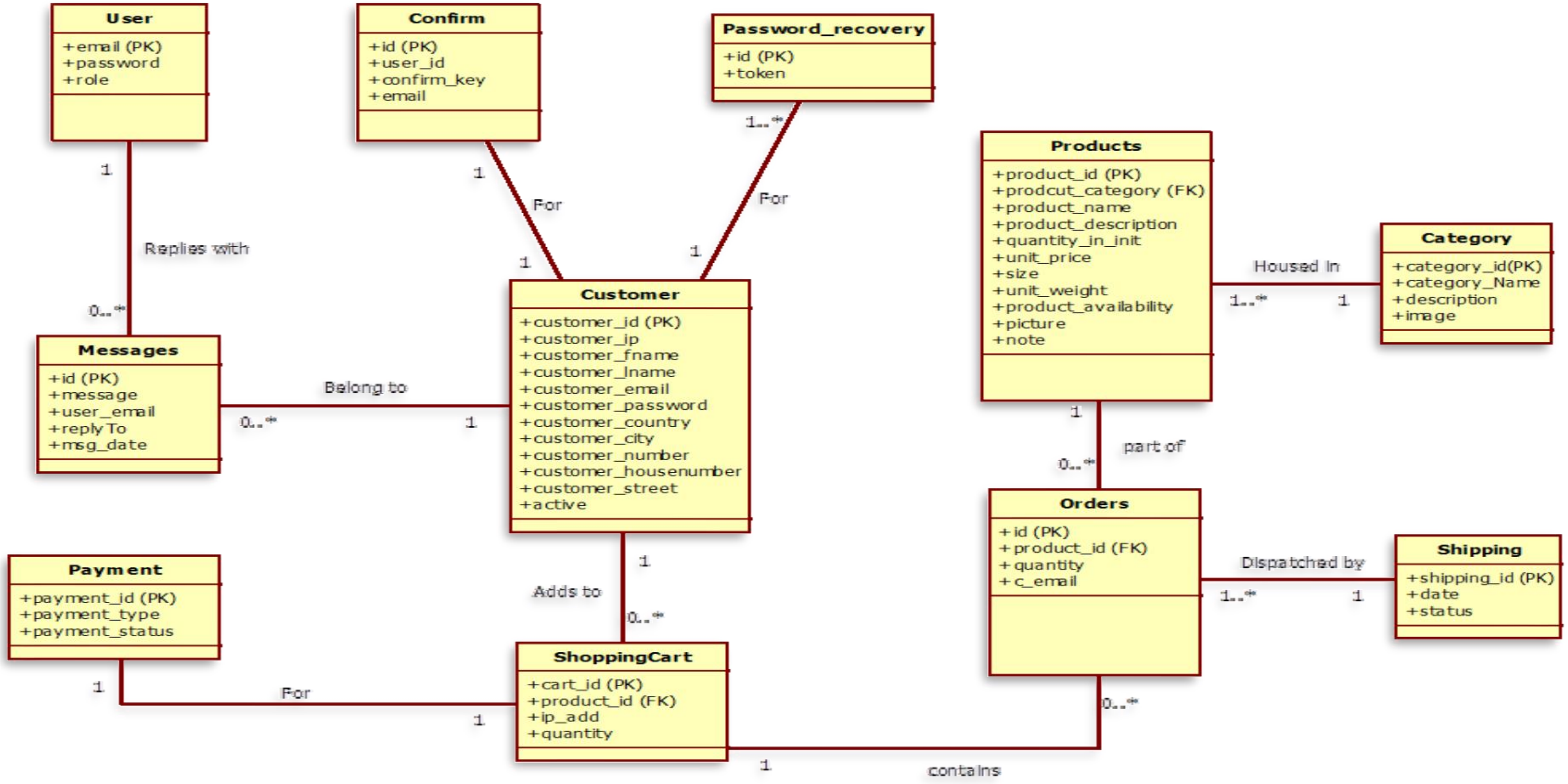


Figure 15 - Database design for the e-commerce website, including the relations and key integrities

5.6. Prototype

The wireframes below will give an insight into the design of the website and what each website should roughly have. The design of the website was developed through the software called AXURE. These design models should show an idea of what the website should look like. The website itself could and could not look like the designs shown below but each wireframe design below has attempted to show the way the website requirements would look on delivery.

5.6.1. Homepage

Figure 16 below shows the wireframe for the homepage. The homepage should consist of images which slideshow to show the different types of products available, offers and a side bar on the right. The navigation bar will be present at the top of the pages where it is needed. All boxes in the wireframes marked with an 'X' in the middle are there to represent images.

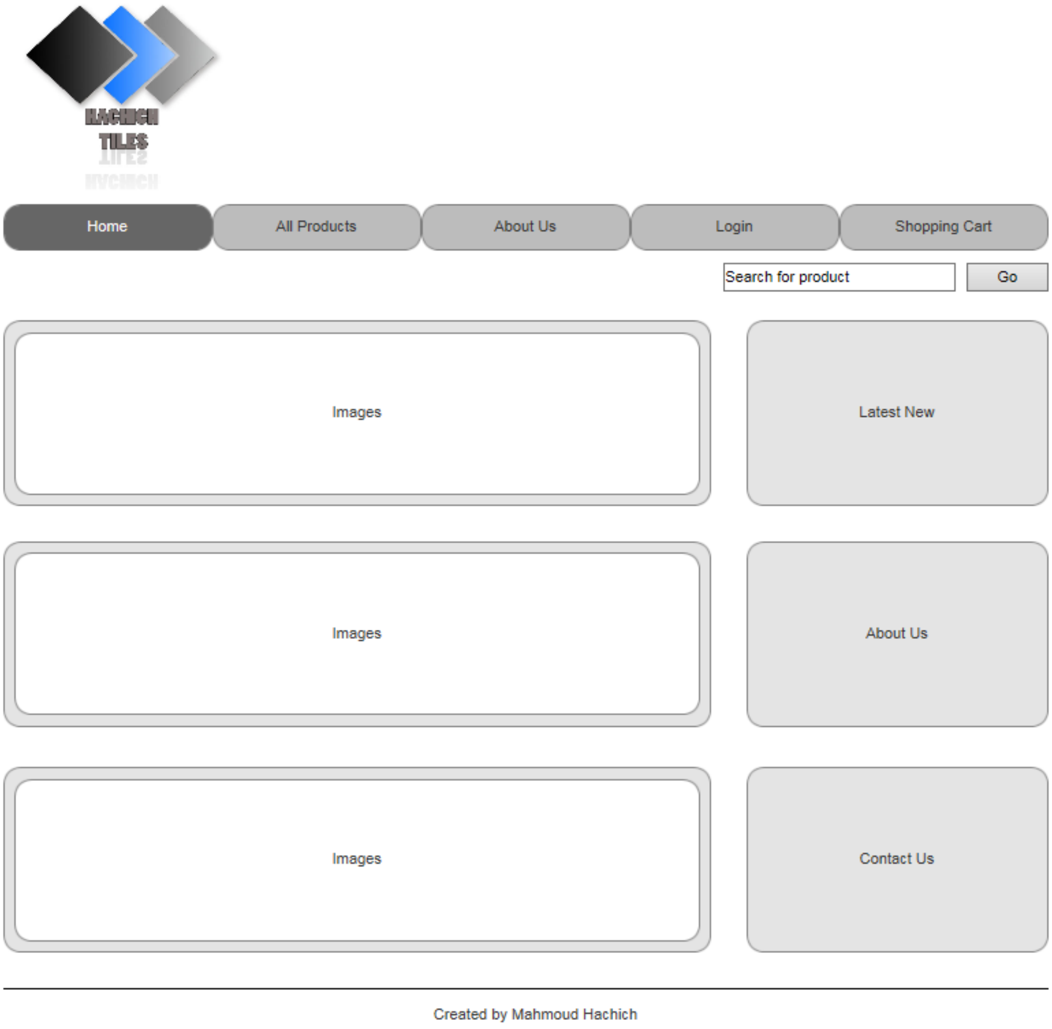


Figure 16 - Prototype for the homepage of the website

5.6.2. All Products

The all products page is one of the main features of the website. Here the customer should be able to view the products based on the different categories available. The categories will be displayed on the left-hand side of the page, whilst the products will be displayed centre-right. The user should be able to add the products to the cart and view more details for it. Figure 16 below gives a brief look of what the webpage would look like.

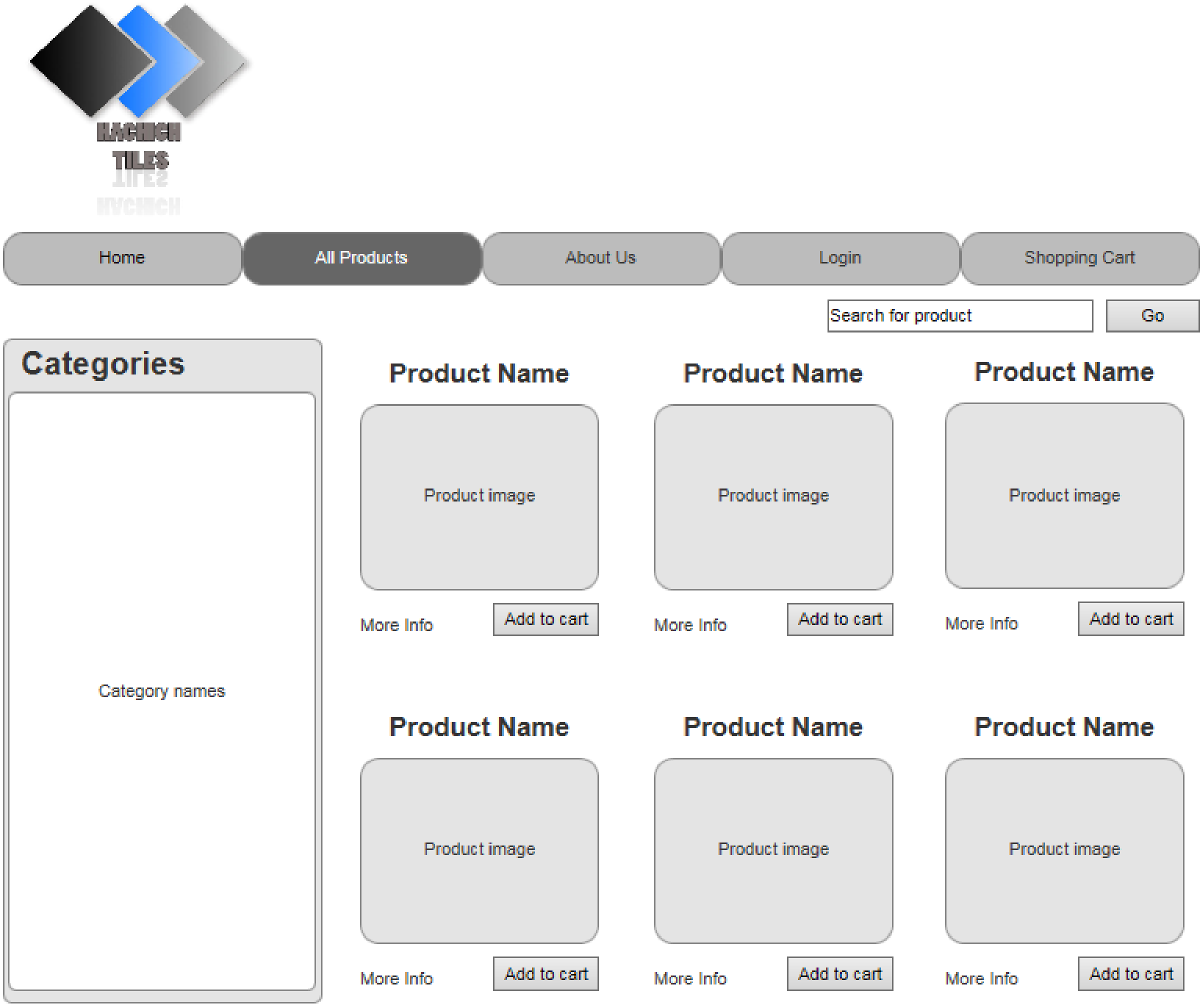
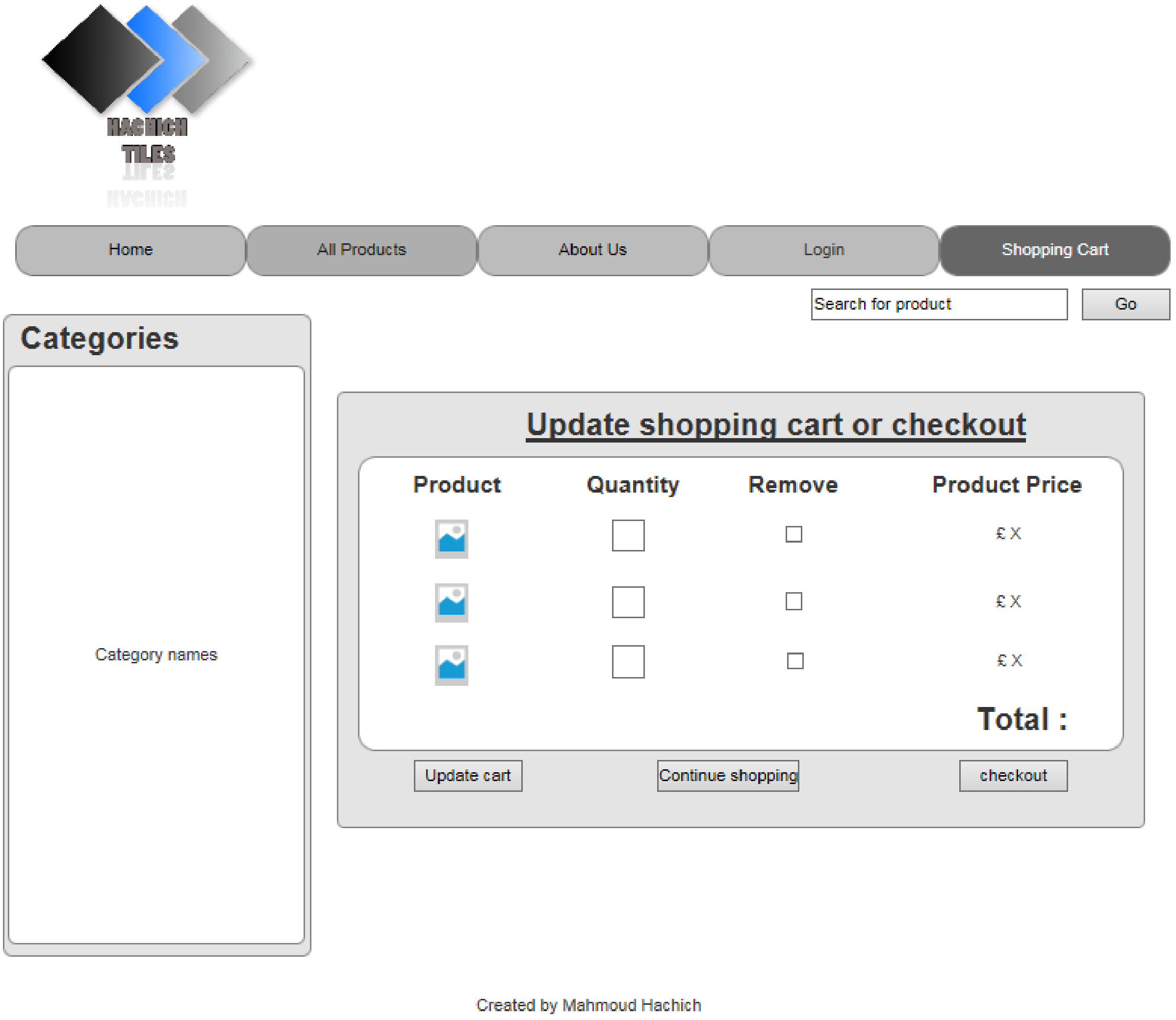


Figure 17 - Prototype of how all products should be displayed.

5.6.3. Shopping Cart

Figure 18 below shows the design of the shopping cart page. The customer should be able to make change to the products which are available in the cart; removing and changing the quantity of the products would also be present.



Product	Quantity	Remove	Product Price
	□	□	£ X
	□	□	£ X
	□	□	£ X
			Total :

Figure 18 - Prototype of the shopping cart, ability to update, checkout and login should be available when implementing

5.6.4. Account Page

When logged in, the customer should be presented with a page like the one presented below. Figure 19 shows the splitting of the account page into 3 different categories. This should give the user different levels of access as he/she wishes.

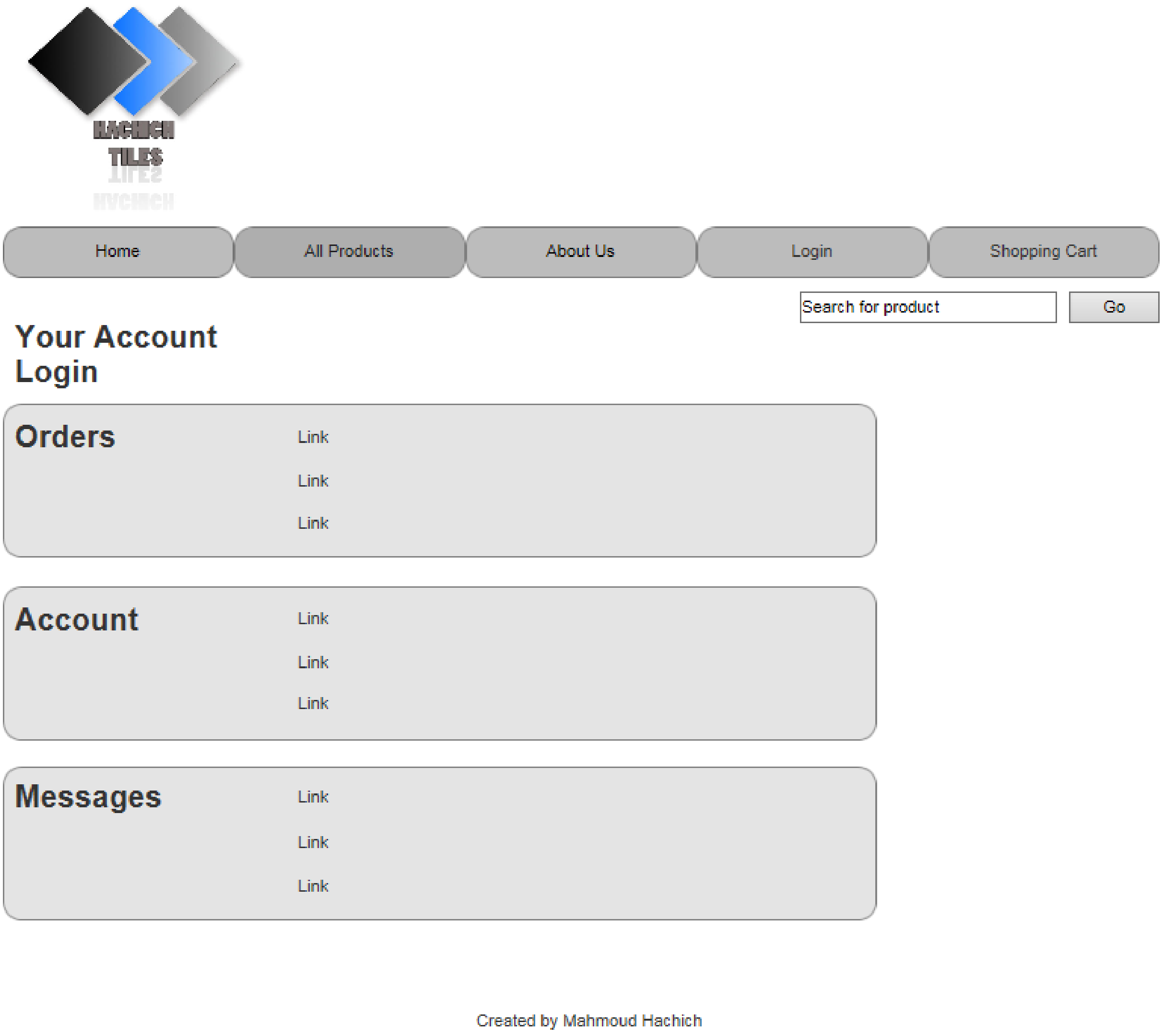


Figure 19 - Prototype for the account page, split into 3 different sections

5.6.5. Login

Figure 20 has been created on the simple basis of an easy login and logout page for the user. After the user has entered the correct credentials he is able to access his account information and so on. On successful login, the customer should be provided with an info window alerting them of the login.

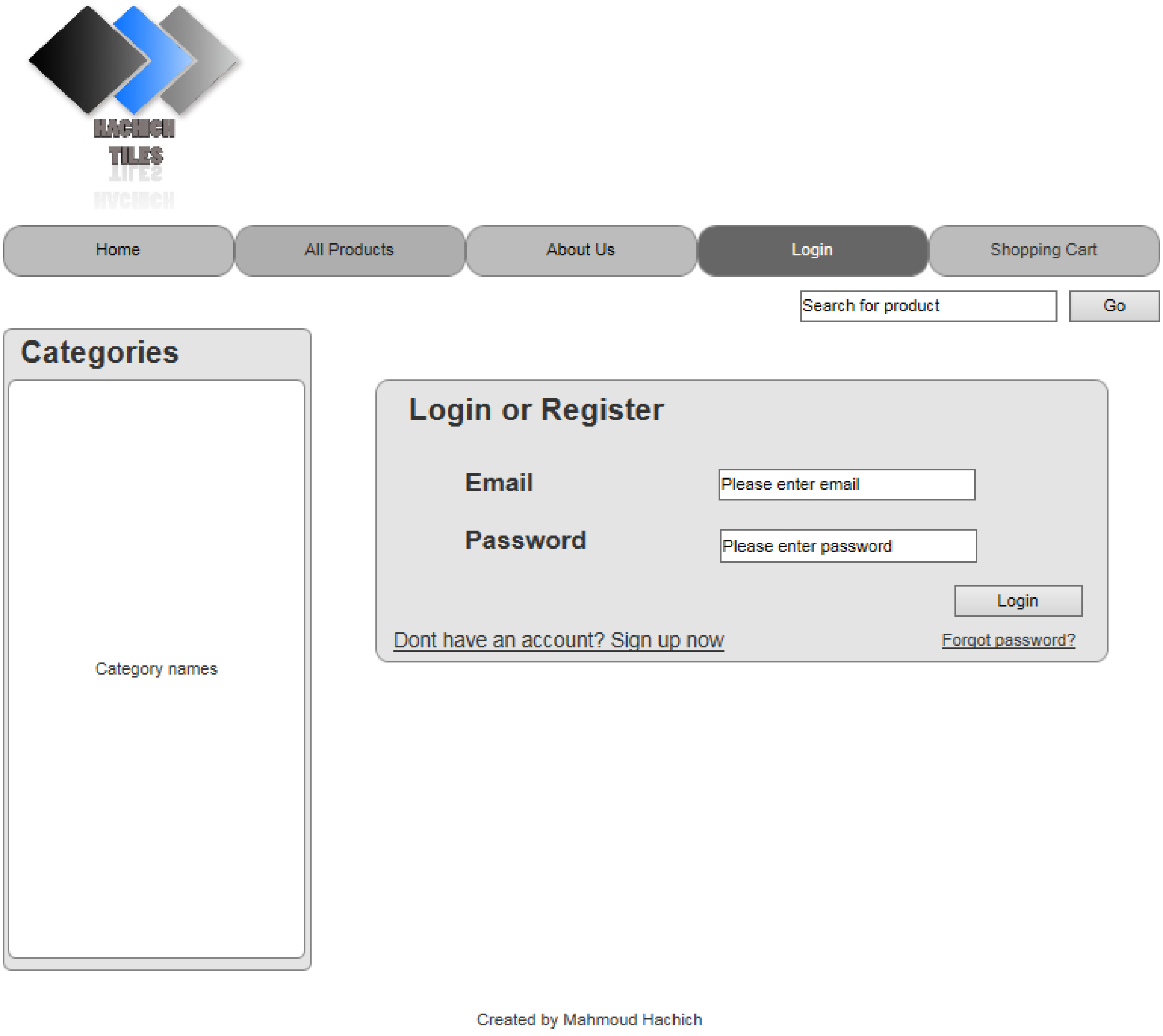
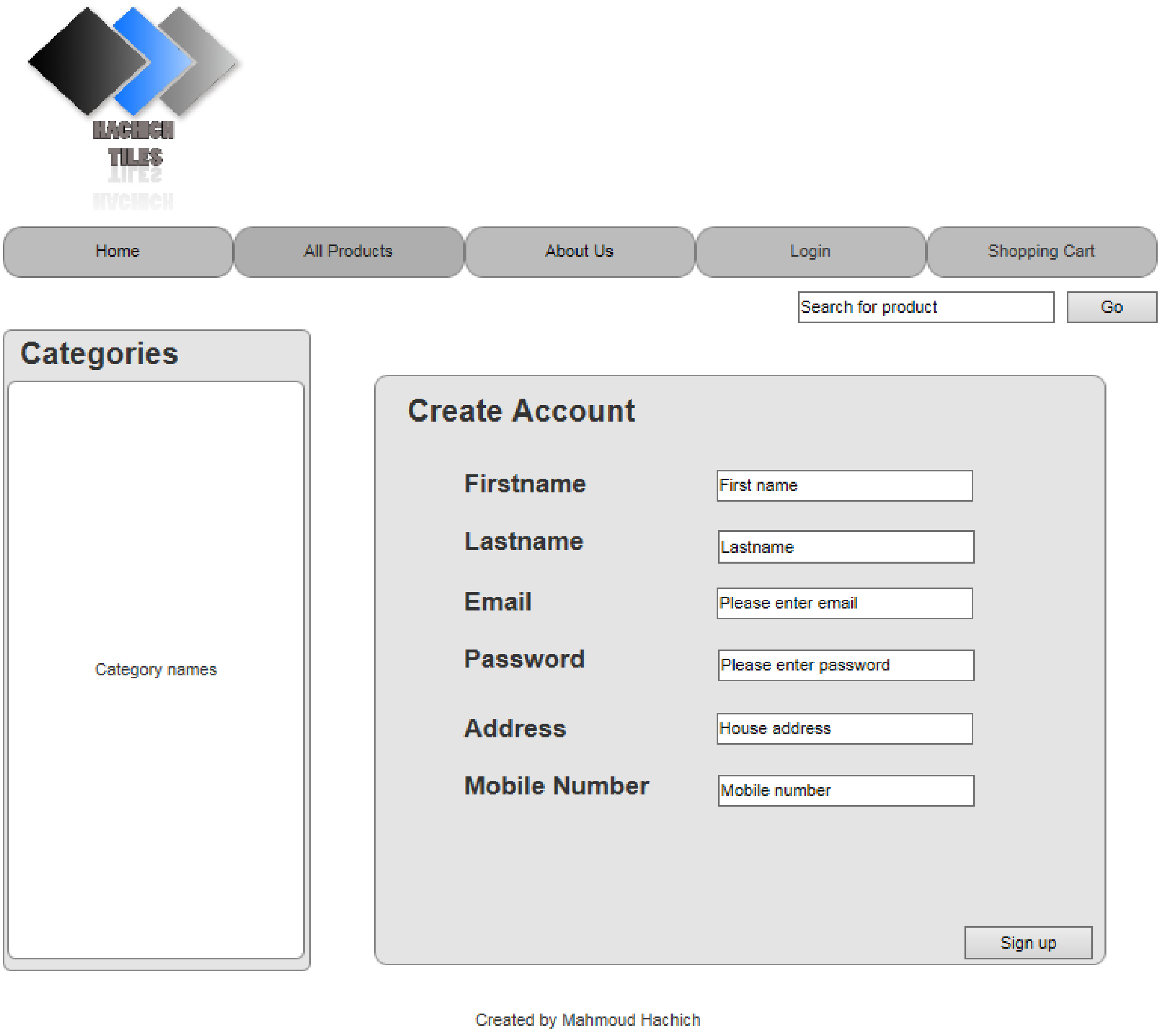


Figure 20 - Prototype for the login page, containing two text fields and a login button.

5.6.6. Create Account

Figure 21 below shows the design of what the page would look like when the customer is registering a new account. The user should be able to enter his details in the text fields and create a new account



Create Account

Firstname: First name

Lastname: Lastname

Email: Please enter email

Password: Please enter password

Address: House address

Mobile Number: Mobile number

Figure 21 - Prototype for creating an account, containing several input text fields so the user information gets stored.

5.6.7. Send Message

The wireframe below is designed to allow the customer to send messages to the seller related to both orders and available products.

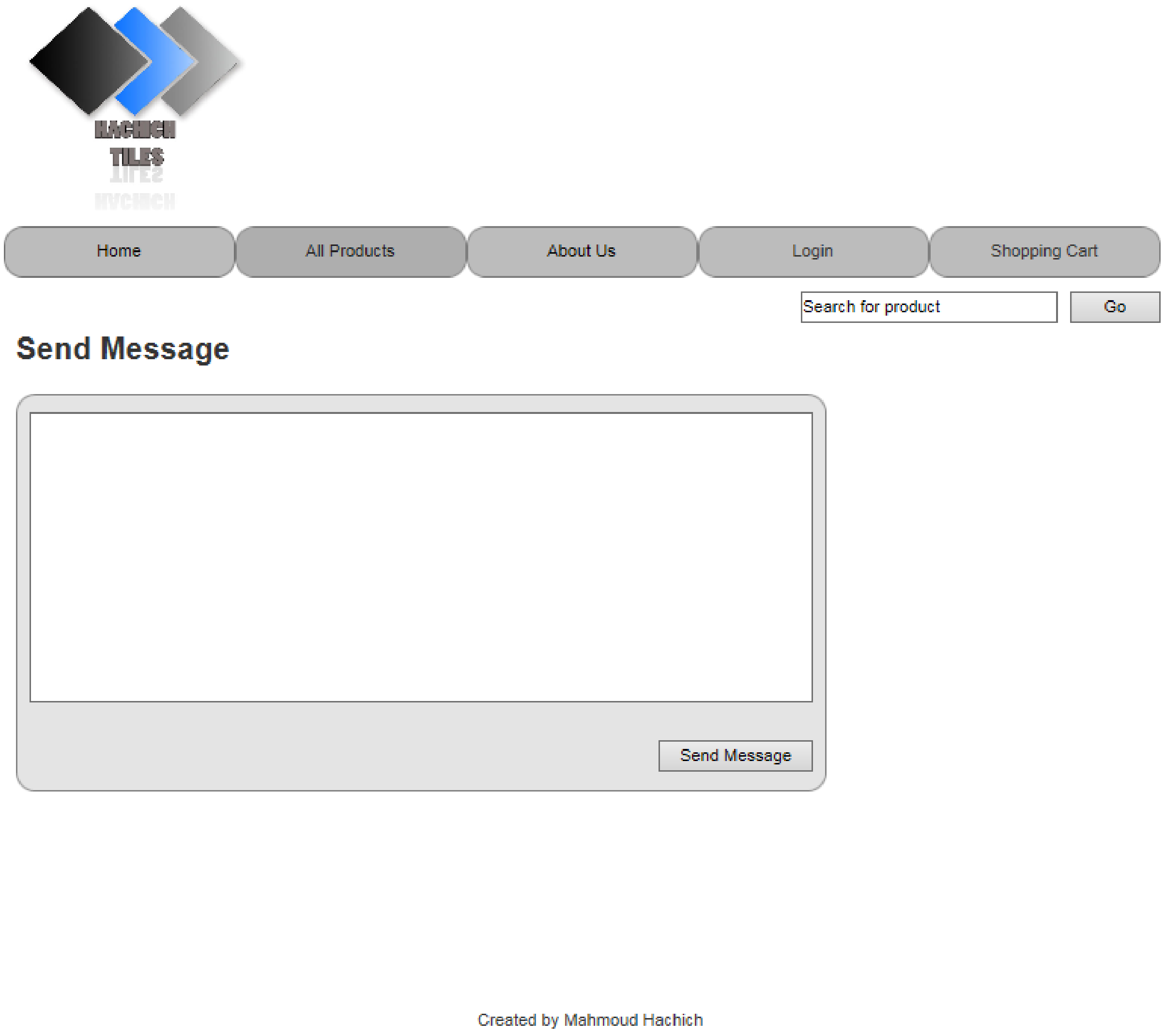
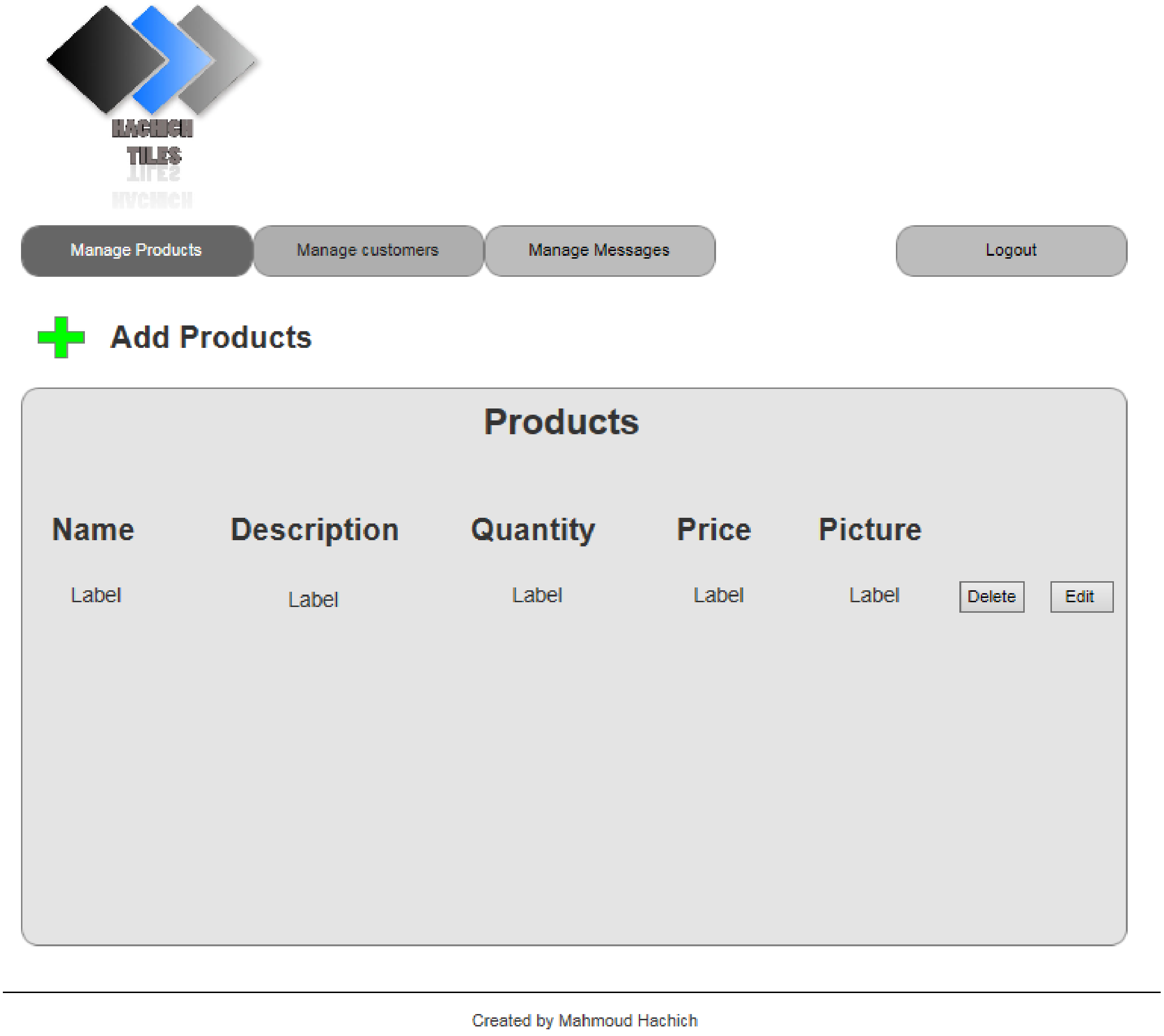


Figure 22 - Prototype for sending a message to the admin when in customer view.

5.6.8. Admin Products Page

This design of the admin panel homepage on login shows all the products that are available with the feature of adding products. The admin should also be able to somehow edit and remove products from the catalogue.



Products

Name	Description	Quantity	Price	Picture		
Label	Label	Label	Label	Label	Delete	Edit

Figure 23 - Prototype for the admin products page, feature such as adding products, deleting, and editing them should be available here.

5.6.9. Admin Add Products

Below is the design of figure 24 where the admin can add products. Fields should be filled in correctly and should be displayed on the customers end.

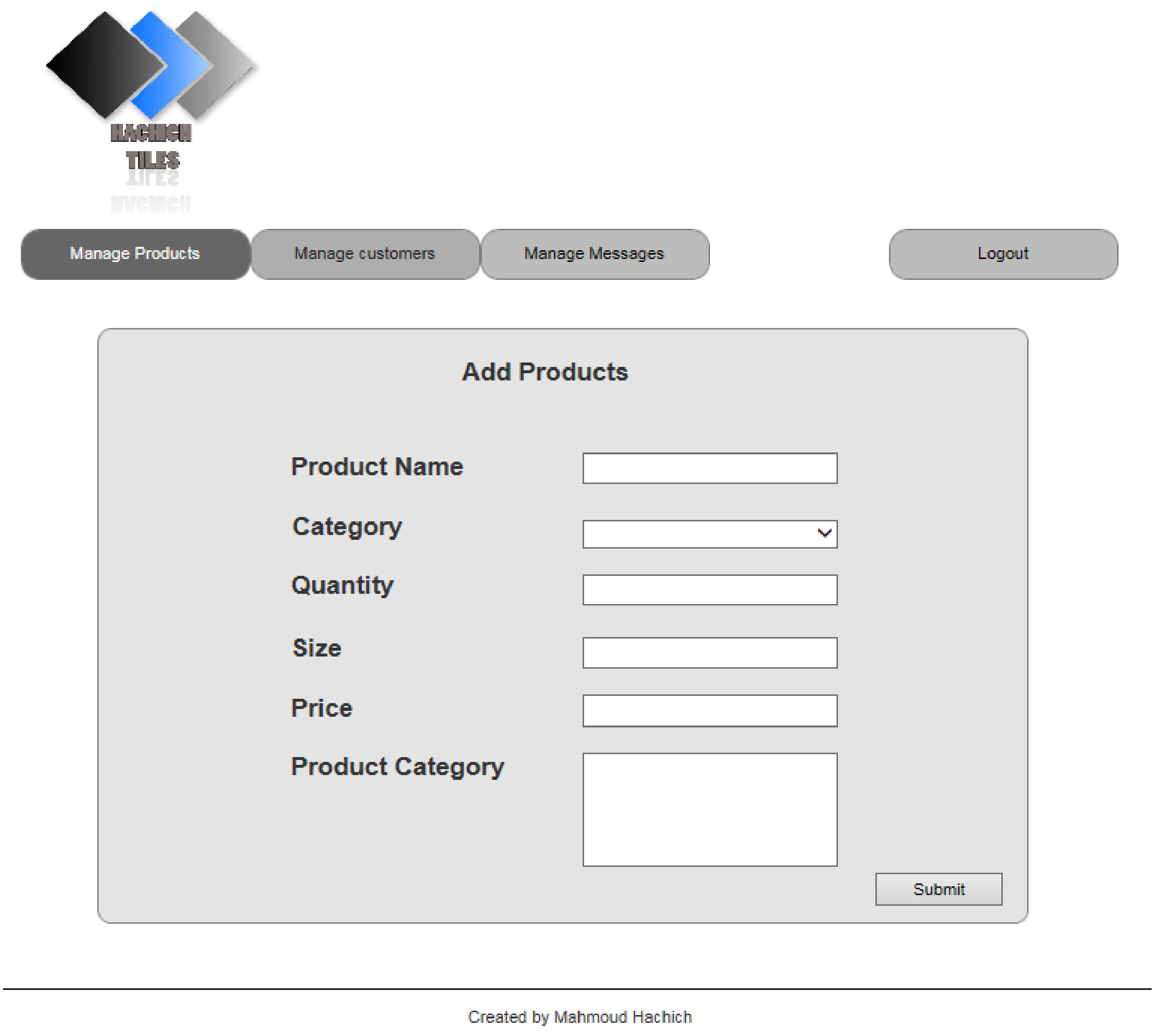
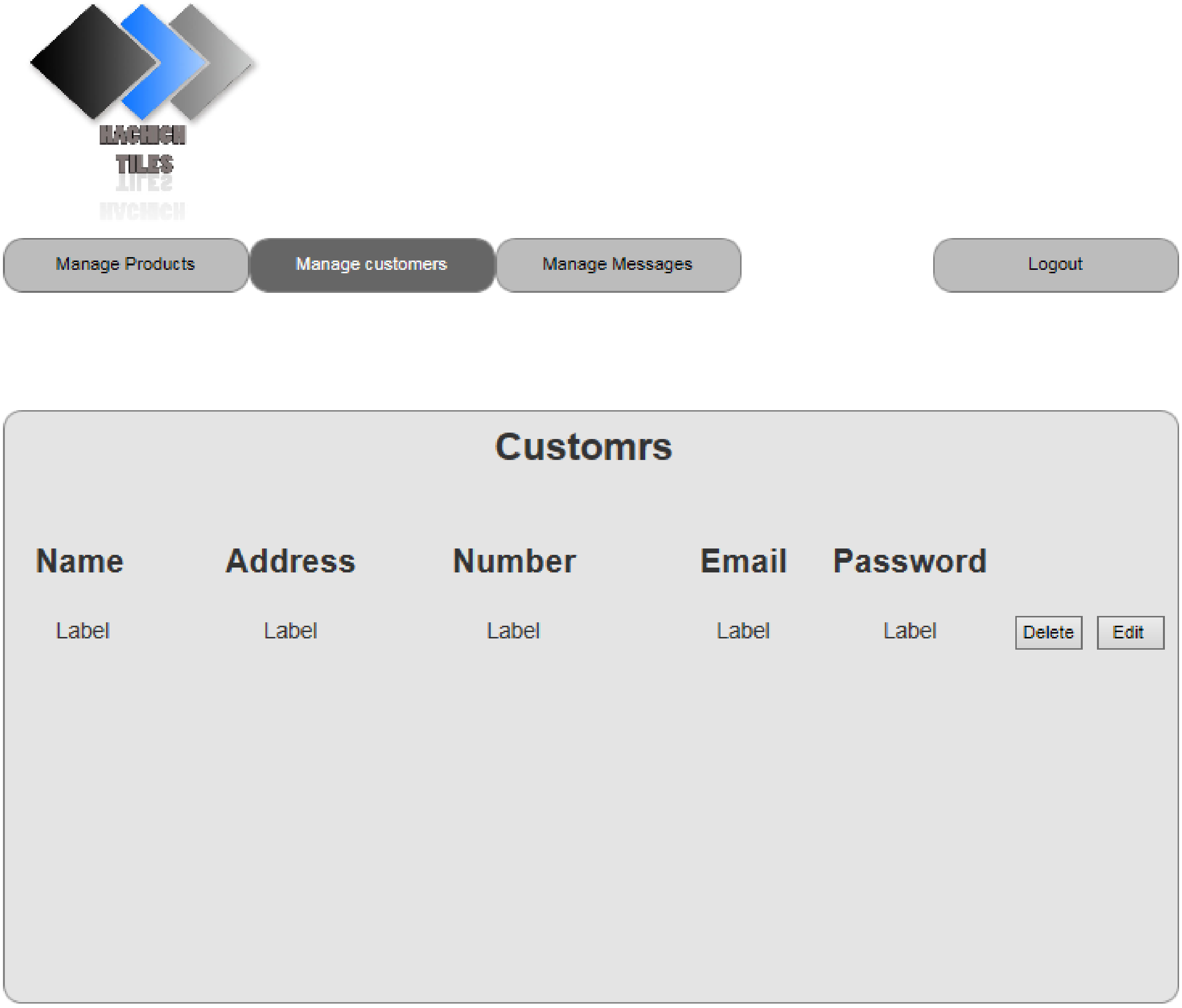


Figure 24 - Prototype for adding products page

5.6.10. Admin Manage Customer

The admin should also be able to edit and view customer information as a requirement. With each customer account available below, admin can have a look at all the customer information before making any changes or deleting account.



Customrs

Name	Address	Number	Email	Password
Label	Label	Label	Label	Label

Figure 25 - Prototype for the managing of customer’s page

5.6.11. About Us Page

Figure 26 below represents the prototype design for the about us page. This page should show the company information and map of the stores location integrated via Google Maps JavaScript API.

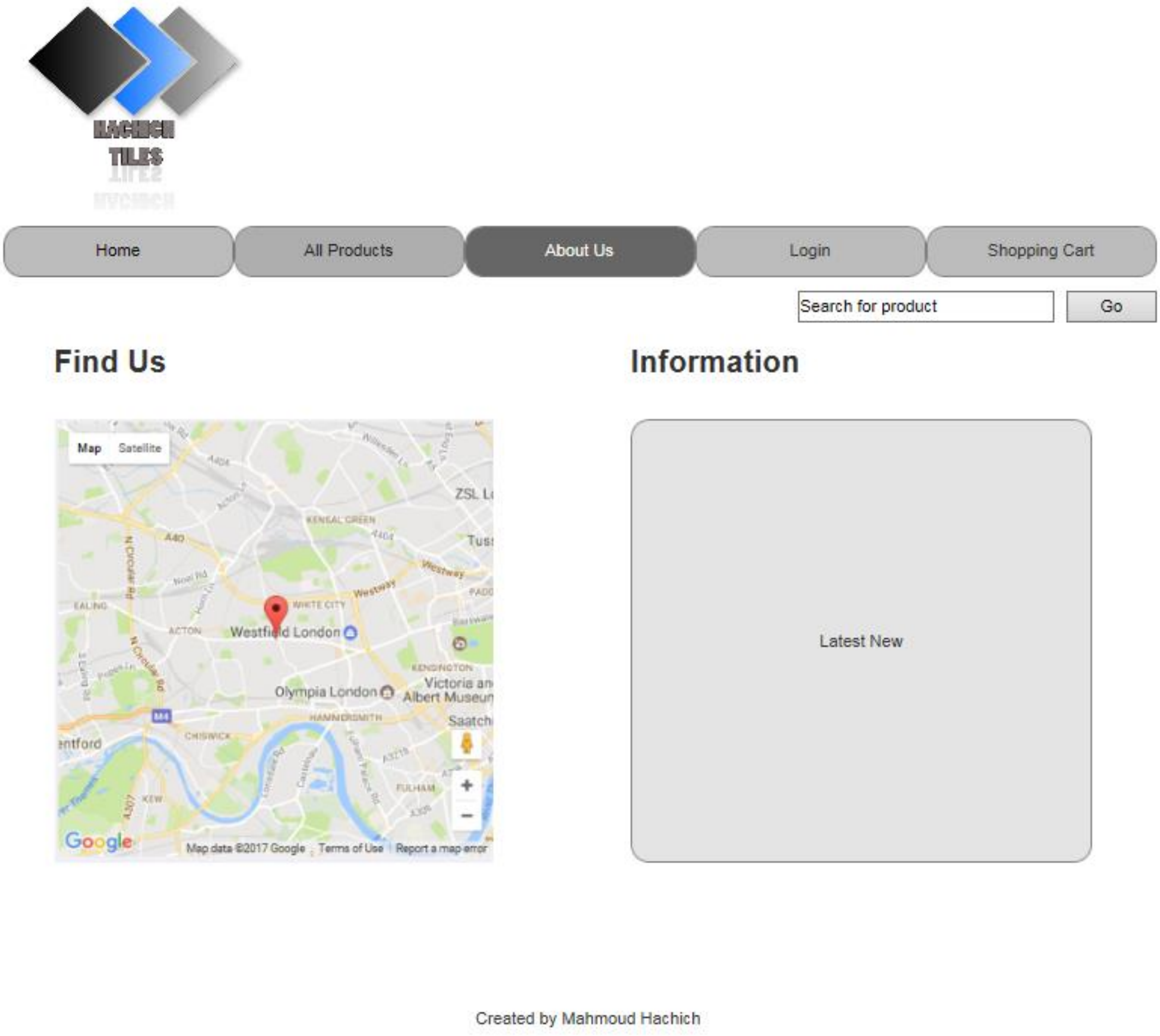


Figure 26 - Prototype for the admins editing of customers.

6. Implementation

When implementing the e-commerce website, there was many ways to implement it to achieve maximum functionality to meet the client's needs. This chapter contains information on the languages used, the tools used and the different methods used to accomplish the website functionality.

6.1. Development Languages

When conducting the literature review, information was found on the best programming language which allows communication with the database through MySQL is PHP. Hence a decision between me and client lead to programming through PHP; this will show evident later in the report. Amongst implementing the PHP code through both procedural programming and the successful completion of the website, technologies such as code ignitor and other MVC frameworks became familiar.

HTML 5 was used to create and implement web pages which are accessible by web browsers. The new version HTML5 has made creating and assembling web pages easier than the older version. HTML5 helps with organising the web pages' structure, field validation and code organisation. One major drawback which is noticeable to the client side is that certain features such as the viewing date and time do not work on different browsers; therefore, this feature has not been implemented.

Cascading stylesheets (CSS) have been used alongside the HTML 5 to bring colour and positioning of the different elements for the customer view. Different stylesheets have been used throughout the website but certain elements of the stylesheets have remained constant throughout the implementation of the website. Elements such as navigation bar, logo and categories are present in all style sheets. All stylesheets where scripted from scratch ensuring their responsiveness.

PHP was used to allow the communication between the HTML web pages and webservers. For this project, the webservers were connected to the database where the product and customer information was stored. Through the hosting company 1 and 1, access to the SFTP information, control panel and database administrator tools were accessed. The hosting came with a service called phpMyAdmin which allows information from the database to be sent, retrieved, and deleted from the e-commerce site.

6.2. Development Tools

When scripting all PHP code, the chosen text editor was one the programmer was familiar with, hence the choice of Notepad++. Notepad++ is a free tool which allows the developer to launch the project whilst developing. Notepad++ is also a free tool which meant implementation of the e-commerce store was conducted at a different location. Features such as line numbers and zooming in and out made it easy with the detection of programming errors.

To run all the PHP functions that this project required and to ensure that this website was functioning correctly, a local web server to test the webpages was used. The software used is XAMPP; XAMPP contains four primary components which are Apache, MySQL, PHP and Perl. The website was hosted by 1 and 1 due to their reasonable monthly price, database storage offered and the use of the new secure file transfer protocol. To ensure communication with 1 and 1's servers, the FTP client FileZilla was sued to allow communication to and from the server. This software is free of charge and was very easy to use.

6.3. Code and Functionality

This part of the report would be based on the implementation of the website using HTML, PHP and CSS. The implementation would be shown in accordance to the functional requirements which have been mentioned above in section 4.3.1. At the beginning of the project, all the PHP code was implemented via procedural programming, but throughout the different modules offered at university, one became familiar with the different frameworks offered. The tool code ignitor was used to implement object oriented PHP through the MVC framework. Although this has not been included in the report due to its incompleteness, the procedural PHP programming is included. For every functional requirement, a snippet of the code with an explanation of its operation will be included. Some HTML forms will be included to show how the PHP communicated with the forms created.

Having developed a dynamic website for the client, customers will be filling out many forms and submitting them whilst browsing the website. A customer willing to purchase products will need to have an account, therefore the 'create account form' will be submitted. When the submit button is pressed, the form data is sent with the HTTP "POST" method.

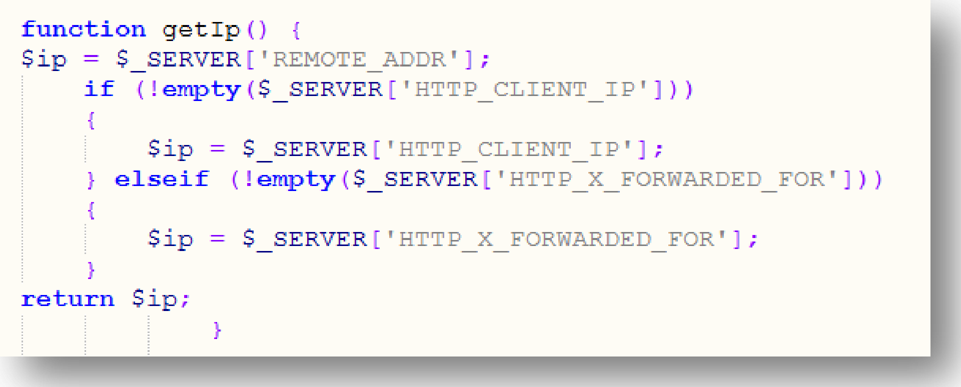
There are two different types of request methods in PHP where data can be sent when submitting a form. Both the "GET" and "POST" method operate in the same way. Both methods create an array which hold the names of the forms and values which are the input data from the user (W3schools, n.d.). Information sent from the servers which is then retrieved is often carried out by the "GET" method. This allows the user to resubmit the same URL repeatedly. On the other hand, the POST method is used for writing data to the servers.

A function which has been widely used is the "isset" function, which checks if the text-field has any data submitted in it. Once this value is true, all the code in the "if" statement containing the 'isset' function will be executed.

A way which is most suitable to store information based on the user's multipage access is through sessions. Sessions are used in this project to store user shopping cart information, against the unique cart id which is stored in the database. This will later be evident when the snippets of code are provided.

6.3.1. Create Account (C1 Functional Requirement)

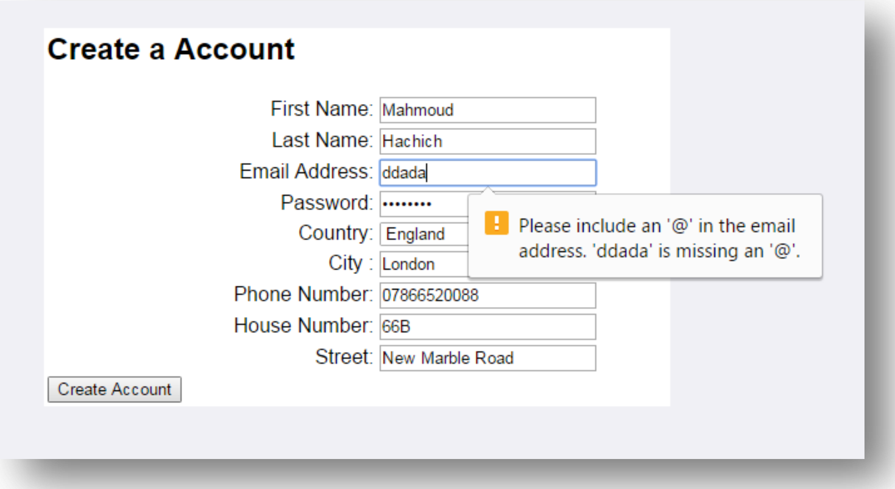
For each person that is willing to make purchases on the website, they must make an account where their personal details are stored. To be able to create a customer account, a form had to be created to parse information into the database. Upon the submitting of a specific form, the details submitted would generally be inserted, deleted, or updated from the database.



```
function getIp() {
$ip = $_SERVER['REMOTE_ADDR'];
    if (!empty($_SERVER['HTTP_CLIENT_IP']))
    {
        $ip = $_SERVER['HTTP_CLIENT_IP'];
    } elseif (!empty($_SERVER['HTTP_X_FORWARDED_FOR']))
    {
        $ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
    }
return $ip;
            }
```

Figure 27 – “getIp” Function, which gets the users IP address and stores it. Used a shopping “cart ID” for when the same user logs in.

Having never built a website before and with little to no experience in web development and the way e-commerce websites function, the different ways of developing a shopping cart where trailed. As shown in figure 27, the shopping cart was initially working via the customer IP address. This has shown to be very difficult for customers accessing the website through the same IP address. Those attempting to access and add products to the shopping cart under the same IP address would have found the same products in the shared carts. This feature had to be changed and it is not one that could have remained.

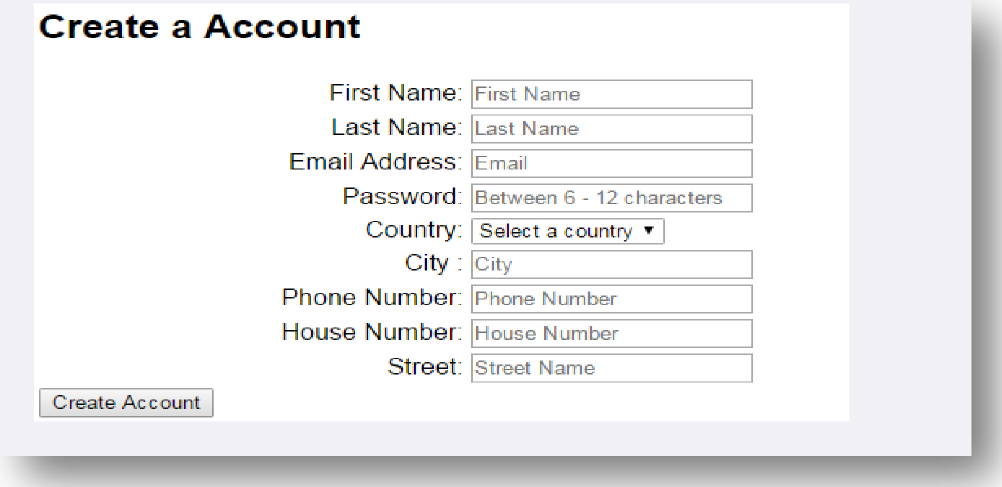


Create a Account

First Name:	Mahmoud
Last Name:	Hachich
Email Address:	ddada
Password:	••••••••
Country:	England
City :	London
Phone Number:	07866520088
House Number:	66B
Street:	New Marble Road

Please include an '@' in the email address. 'ddada' is missing an '@'.

Figure 28 - Create account front-end, with HTML5 email validation

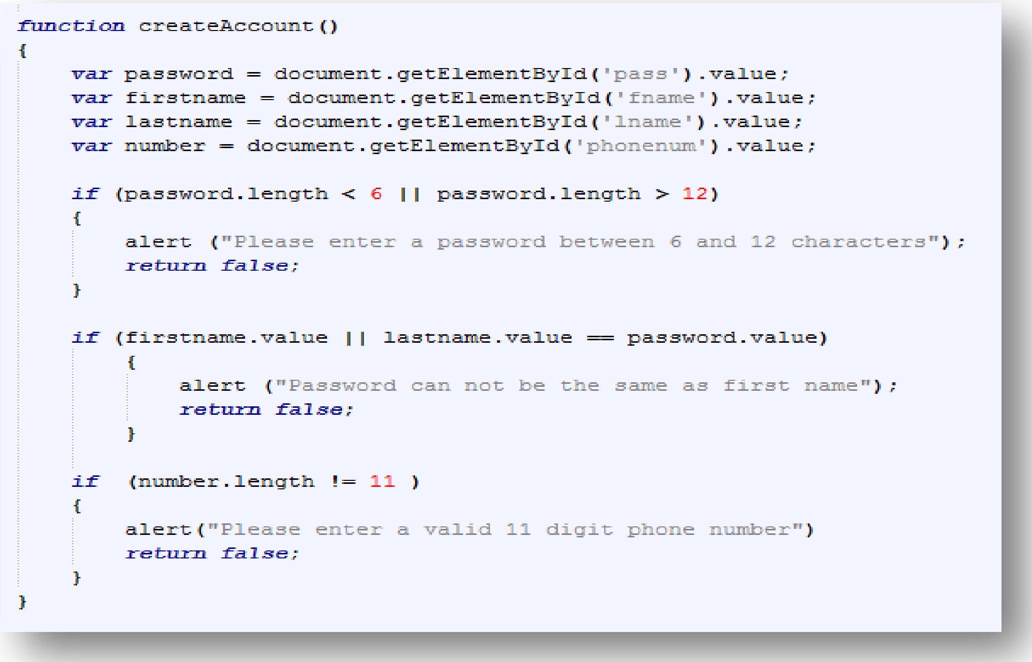


Create a Account

First Name: First Name
Last Name: Last Name
Email Address: Email
Password: Between 6 - 12 characters
Country: Select a country ▼
City : City
Phone Number: Phone Number
House Number: House Number
Street: Street Name

Figure 29 - Create account front-end empty

Using HTML 5 has many advantages to HTML 3; one of these advantages is the ability of assigning every input type to be required. This is shows in figure 28 and 29 above, where a pop-up appears for the email when the user is attempting to create an account and forgets the ‘@’ symbol in their email. This is implemented through the inclusion of the ‘required’ word.

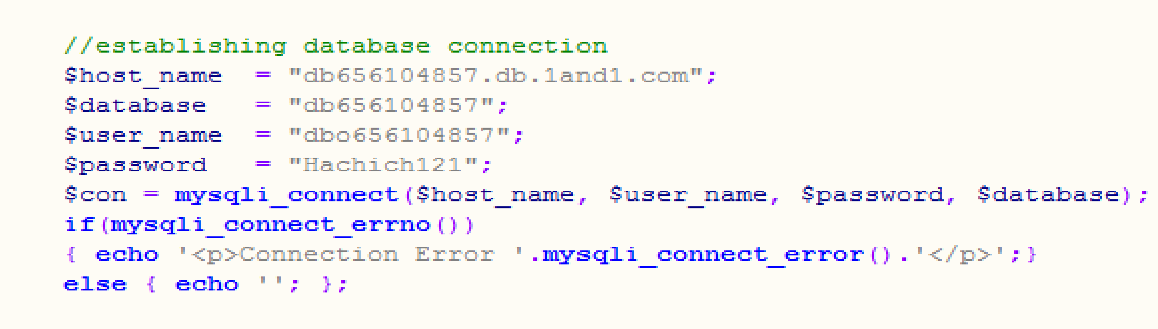


```
function createAccount()
{
    var password = document.getElementById('pass').value;
    var firstname = document.getElementById('fname').value;
    var lastname = document.getElementById('lname').value;
    var number = document.getElementById('phonenum').value;

    if (password.length < 6 || password.length > 12)
    {
        alert ("Please enter a password between 6 and 12 characters");
        return false;
    }

    if (firstname.value || lastname.value == password.value)
        {
            alert ("Password can not be the same as first name");
            return false;
        }

    if  (number.length != 11 )
    {
        alert("Please enter a valid 11 digit phone number")
        return false;
    }
}
```

Figure 30 - JavaScript create account validation

On the other hand, figure 30 shows a JavaScript function called “createAccount”. This figure shows that a user can only enter certain credentials to successfully make an account. A user must ensure that their password is between 6 and 12 characters ensuring a high level of password security. The user is also unable to use their first name or last names as their password. The password length should also be between 6 to 12 characters otherwise an alert message is shown.

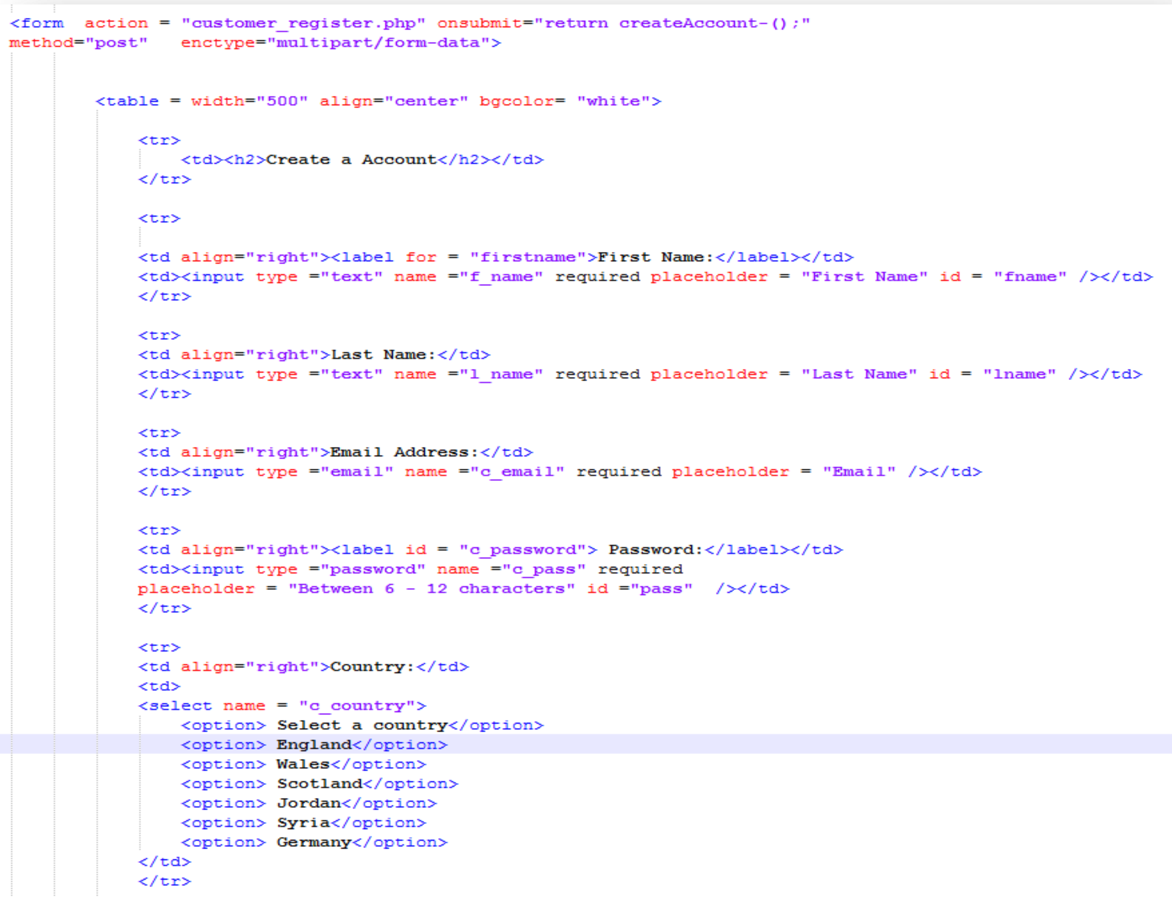
Figure 31 below show the connection to the database which is made. This connection ensures that when a customer requests a functionality which invloves reading, writing, updating and deleting data a error does not occur. The code below was included in every PHP file.



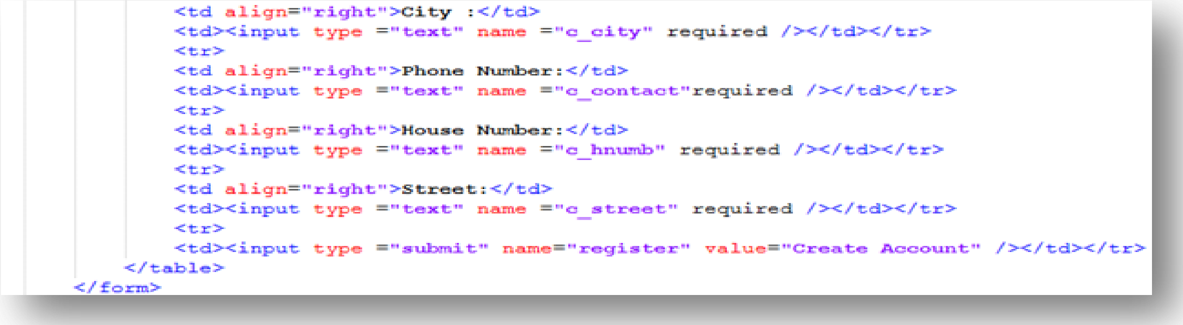
```
//establishing database connection
$host_name  = "db656104857.db.1and1.com";
$database   = "db656104857";
$user_name  = "dbo656104857";
$password   = "Hachich121";
$con = mysqli_connect($host_name, $user_name, $password, $database);
if(mysqli_connect_errno())
{ echo '<p>Connection Error '.mysqli_connect_error().'</p>';}
else { echo ''; };
```

Figure 31 - Establishment of database connection

Below shows the HTML code responsible for the creating of a table to organise all the input fields together to give the look that was shown in figure 26. All the input fields are required and this is where the user inputs their details. Each input type has a field name which is used during PHP to call on functions and assign values to what's in the names field to the database.



```
<form  action = "customer_register.php" onsubmit="return createAccount-();"
method="post"   enctype="multipart/form-data">


        <table = width="500" align="center" bgcolor= "white">

            <tr>
                <td><h2>Create a Account</h2></td>
            </tr>

            <tr>

            <td align="right"><label for = "firstname">First Name:</label></td>
            <td><input type ="text" name ="f_name" required placeholder = "First Name" id = "fname" /></td>
            </tr>

            <tr>
            <td align="right">Last Name:</td>
            <td><input type ="text" name ="l_name" required placeholder = "Last Name" id = "lname" /></td>
            </tr>

            <tr>
            <td align="right">Email Address:</td>
            <td><input type ="email" name ="c_email" required placeholder = "Email" /></td>
            </tr>

            <tr>
            <td align="right"><label id = "c_password"> Password:</label></td>
            <td><input type ="password" name ="c_pass" required
            placeholder = "Between 6 - 12 characters" id ="pass"  /></td>
            </tr>

            <tr>
            <td align="right">Country:</td>
            <td>
            <select name = "c_country">
                <option> Select a country</option>
                <option> England</option>
                <option> Wales</option>
                <option> Scotland</option>
                <option> Jordan</option>
                <option> Syria</option>
                <option> Germany</option>
            </td>
            </tr>
```

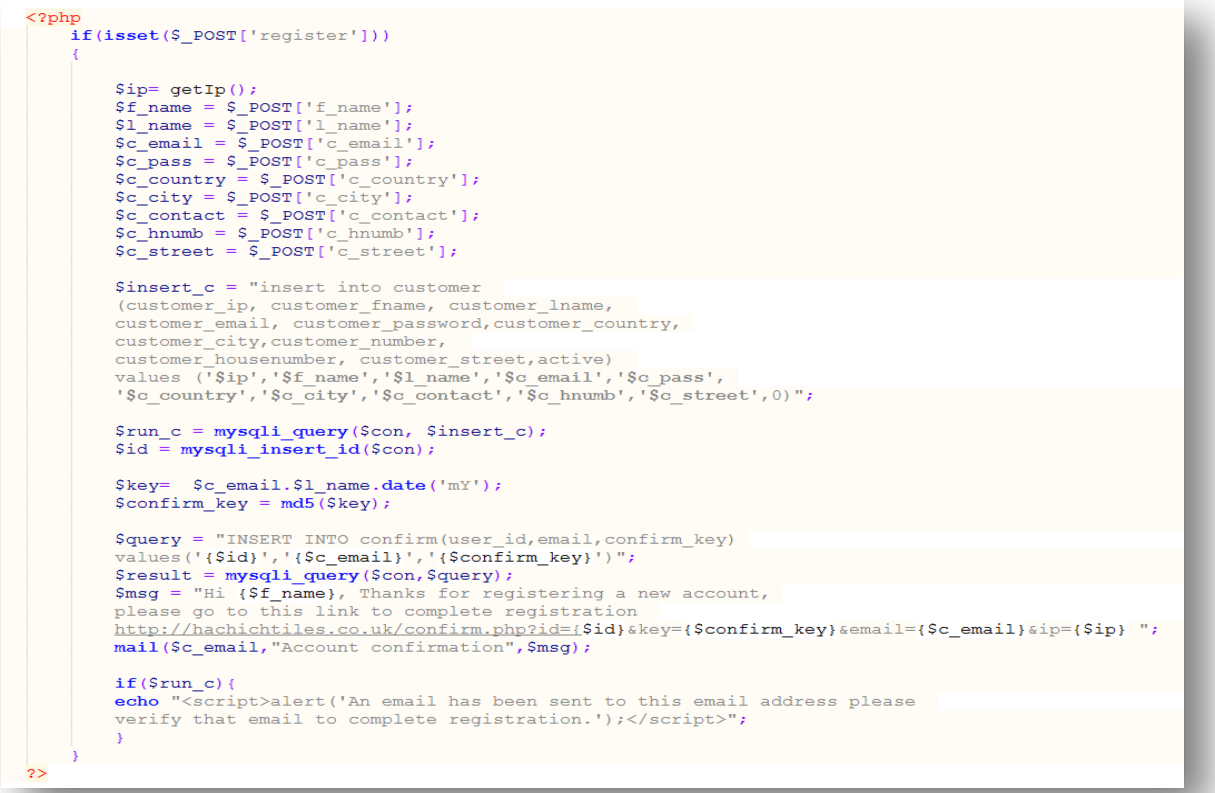


```
                <td align="right">City :</td>
                <td><input type ="text" name ="c_city" required /></td></tr>
                <tr>
                <td align="right">Phone Number:</td>
                <td><input type ="text" name ="c_contact"required /></td></tr>
                <tr>
                <td align="right">House Number:</td>
                <td><input type ="text" name ="c_hnumb" required /></td></tr>
                <tr>
                <td align="right">Street:</td>
                <td><input type ="text" name ="c_street" required /></td></tr>
                <tr>
                <td><input type ="submit" name="register" value="Create Account" /></td></tr>
            </table>
        </form>
```

Figure 32 - Create account HTML form

Figure 30 shows the PHP code which gets executed when the form is submitted. In the Function, itself the code does not have to include the database connection statement.

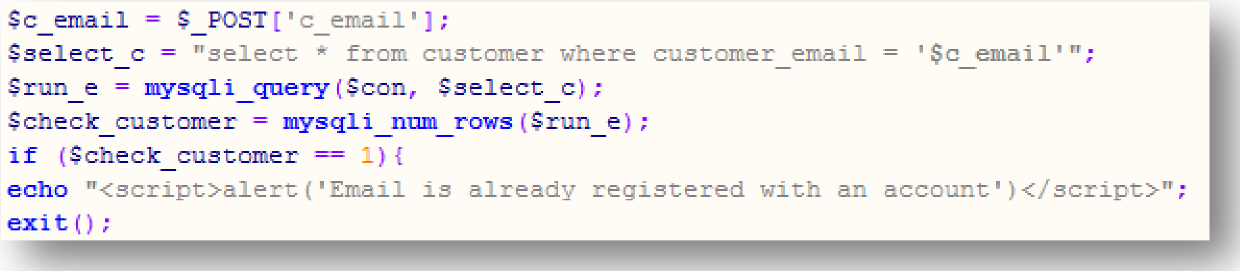
The figure below shows the code which is responsible for creating and storing the customer's information into the database. When a user is to submit the form called "register" variables are created which are assigned to the input fields assigned in figure 32. A SQL query is then used to insert the variables created to fill the customer's database table columns. This function ensures that when a customer clicks on the "create account" button that everything in the "if" statement gets executed so that the customer's details are saved safely in the database.



```
<?php
    if(isset($_POST['register']))
    {

        $ip= getIp();
        $f_name = $_POST['f_name'];
        $l_name = $_POST['l_name'];
        $c_email = $_POST['c_email'];
        $c_pass = $_POST['c_pass'];
        $c_country = $_POST['c_country'];
        $c_city = $_POST['c_city'];
        $c_contact = $_POST['c_contact'];
        $c_hnumb = $_POST['c_hnumb'];
        $c_street = $_POST['c_street'];

        $insert_c = "insert into customer
        (customer_ip, customer_fname, customer_lname,
        customer_email, customer_password,customer_country,
        customer_city,customer_number,
        customer_housenumber, customer_street,active)
        values ('$ip','$f_name','$l_name','$c_email','$c_pass',
        '$c_country','$c_city','$c_contact','$c_hnumb','$c_street',0)";

        $run_c = mysqli_query($con, $insert_c);
        $id = mysqli_insert_id($con);

        $key=  $c_email.$l_name.date('mY');
        $confirm_key = md5($key);

        $query = "INSERT INTO confirm(user_id,email,confirm_key)
        values('{$id}','{$c_email}','{$confirm_key}')";
        $result = mysqli_query($con,$query);
        $msg = "Hi {$f_name}, Thanks for registering a new account,
        please go to this link to complete registration
        http://hachichtiles.co.uk/confirm.php?id={$id}&key={$confirm_key}&email={$c_email}&ip={$ip} ";
        mail($c_email,"Account confirmation",$msg);

        if($run_c){
        echo "<script>alert('An email has been sent to this email address please
        verify that email to complete registration.');</script>";
        }
    }
?>
```

Figure 33 - PHP code for register a customer

Figure 34 below is part of figure 33 but was implemented later on preventing the duplication of the same account being created with the use of an “if” statement. The code below is used to check if the person trying to create an account has entered an email which belongs to an account in the customers table. The variable “$check_customer” runs through the number of rows in the customers table and checks if the input field email assigned to the variable “$c_email” contains an email which is already present in the table.

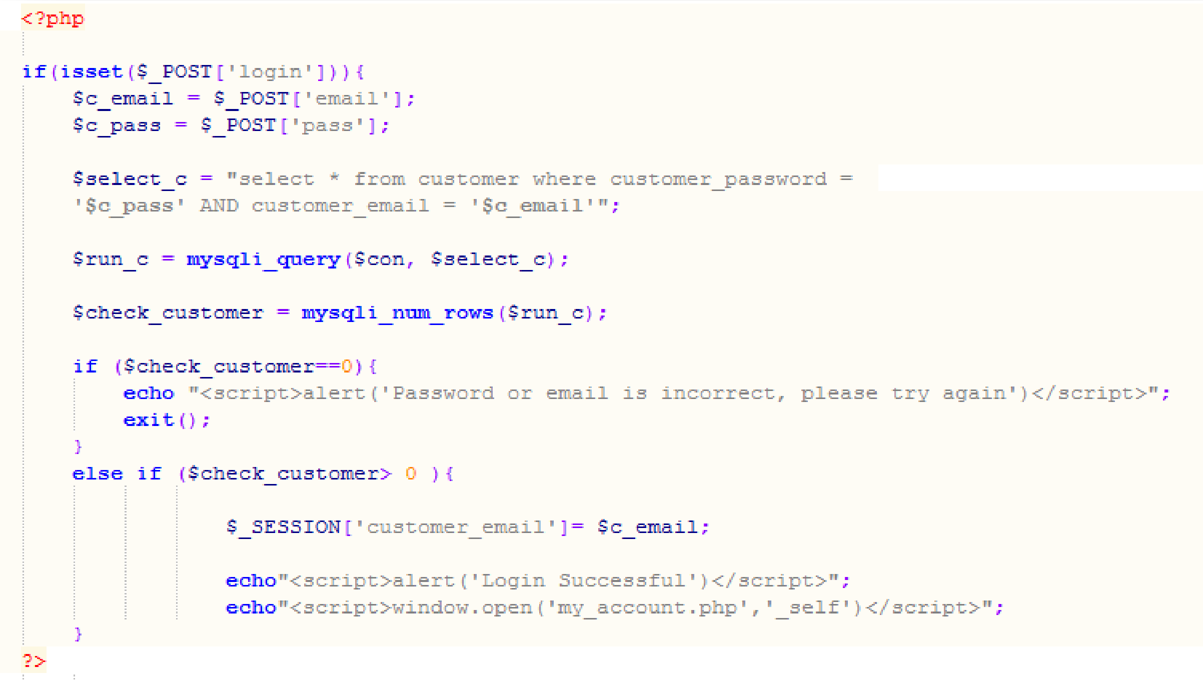


```
$c_email = $_POST['c_email'];
$select_c = "select * from customer where customer_email = '$c_email'";
$run_e = mysqli_query($con, $select_c);
$check_customer = mysqli_num_rows($run_e);
if ($check_customer == 1){
echo "<script>alert('Email is already registered with an account')</script>";
exit();
```

Figure 34 - PHP if statement preventing the same user creating an account again

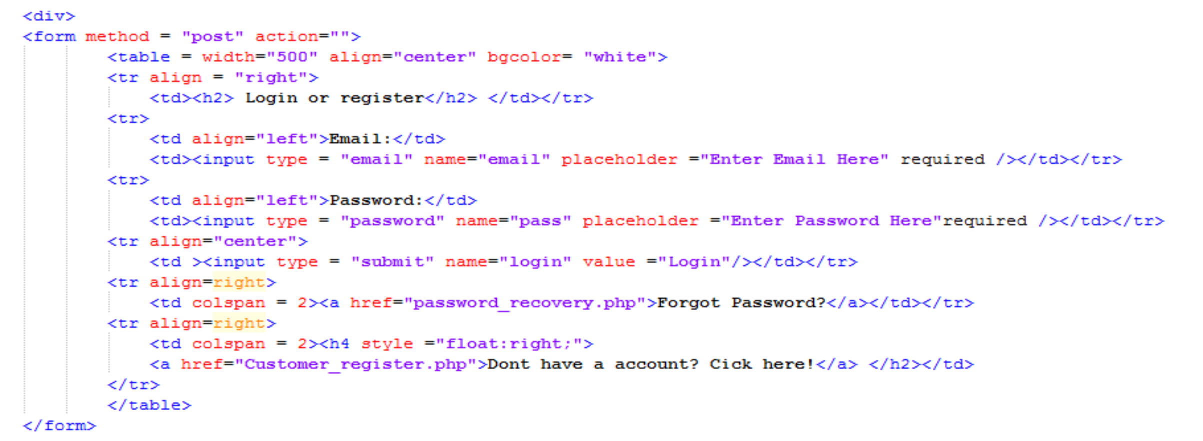
6.3.2. Login with an account (C2 Functional Requirement)

Once a customer makes an account they can login to access their account details, order history, browse products and perform more functionality. The code in figure 35 below shows two variables which are assigned to the input fields created in figure 36. One submission of the form called “login”, the email and password input field name are assigned to two variables which are then used to check if that information is in the customer’s database.” If” the email or password entered do not exist in the customers table show an error message, “else” successfully log the customer in.



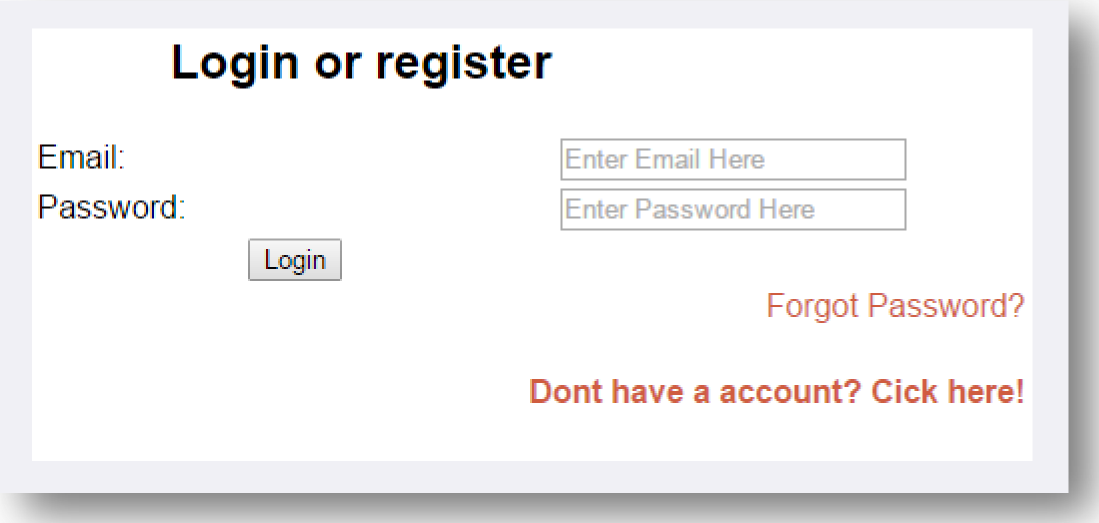
```
<?php

if(isset($_POST['login'])){
    $c_email = $_POST['email'];
    $c_pass = $_POST['pass'];

    $select_c = "select * from customer where customer_password =
    '$c_pass' AND customer_email = '$c_email'";

    $run_c = mysqli_query($con, $select_c);

    $check_customer = mysqli_num_rows($run_c);

    if ($check_customer==0){
        echo "<script>alert('Password or email is incorrect, please try again')</script>";
        exit();
    }
    else if ($check_customer> 0 ){

                $_SESSION['customer_email']= $c_email;

                echo"<script>alert('Login Successful')</script>";
                echo"<script>window.open('my_account.php','_self')</script>";
    }
?>
```

Figure 35 - PHP code for login



```
<div>
<form method = "post" action="">
        <table = width="500" align="center" bgcolor= "white">
        <tr align = "right">
            <td><h2> Login or register</h2> </td></tr>
        <tr>
            <td align="left">Email:</td>
            <td><input type = "email" name="email" placeholder ="Enter Email Here" required /></td></tr>
        <tr>
            <td align="left">Password:</td>
            <td><input type = "password" name="pass" placeholder ="Enter Password Here"required /></td></tr>
        <tr align="center">
            <td ><input type = "submit" name="login" value ="Login"/></td></tr>
        <tr align=right>
            <td colspan = 2><a href="password_recovery.php">Forgot Password?</a></td></tr>
        <tr align=right>
            <td colspan = 2><h4 style ="float:right;">
            <a href="Customer_register.php">Dont have a account? Cick here!</a> </h2></td>
        </tr>
        </table>
</form>
```

Figure 36 - Login HTML form



Login or register

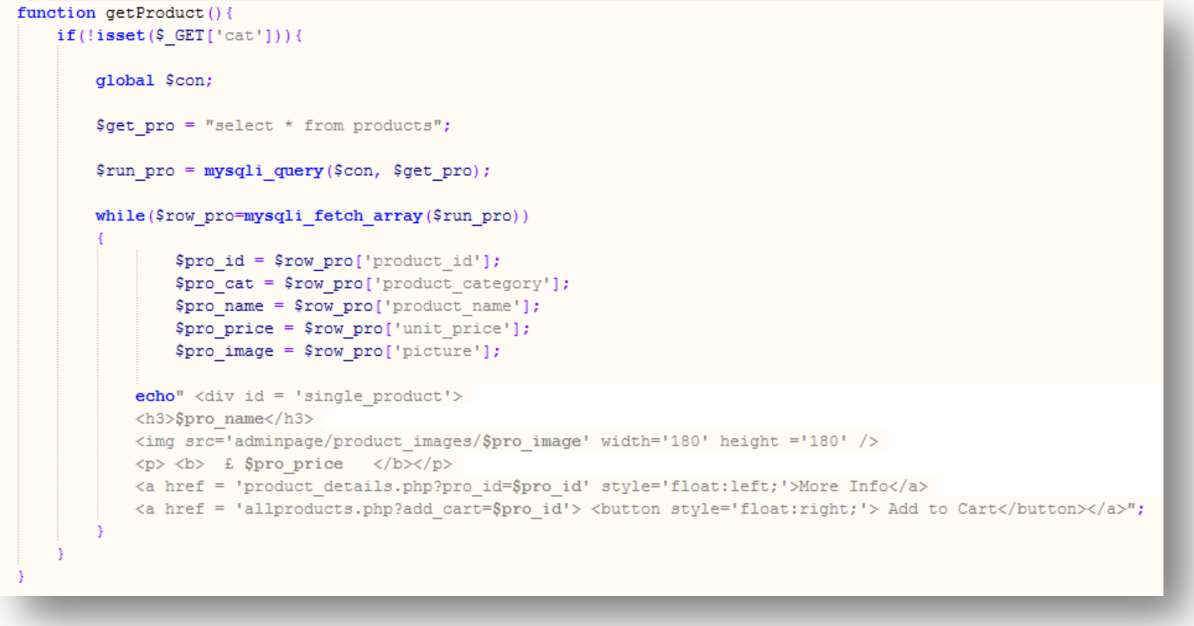
Email: Enter Email Here

Password: Enter Password Here

Figure 37 - Login front-end empty

6.3.3. Browse Products (C3 Functional Requirement)

The function “getProduct” which is shown in figure 38 is responsible for retrieving the products from the database and displaying them in the all products page. The variable “$get_pro” is declared which contains a SQL statement where all products in the products table are selected. A “while loop” is then used to store the products received from the database in an associative array. The “while loop” then assigns the variables to the database table products columns. The collected information from the array is then displayed in a div element called “single_product”.



```
function getProduct(){
    if(!isset($_GET['cat'])){

        global $con;

        $get_pro = "select * from products";

        $run_pro = mysqli_query($con, $get_pro);

        while($row_pro=mysqli_fetch_array($run_pro))
        {
                $pro_id = $row_pro['product_id'];
                $pro_cat = $row_pro['product_category'];
                $pro_name = $row_pro['product_name'];
                $pro_price = $row_pro['unit_price'];
                $pro_image = $row_pro['picture'];

            echo" <div id = 'single_product'>
            <h3>$pro_name</h3>
            <img src='adminpage/product_images/$pro_image' width='180' height ='180' />
            <p> <b>  £ $pro_price   </b></p>
            <a href = 'product_details.php?pro_id=$pro_id' style='float:left;'>More Info</a>
            <a href = 'allproducts.php?add_cart=$pro_id'> <button style='float:right;'> Add to Cart</button></a>";
        }
    }
}
```

Figure 38 - PHP function "getProduct" to retrieve all product information and display it on products page

Figure 38 below is how the product will be displayed for the customer. Both buttons are functional and perform dynamic functions to meet customers requirements.

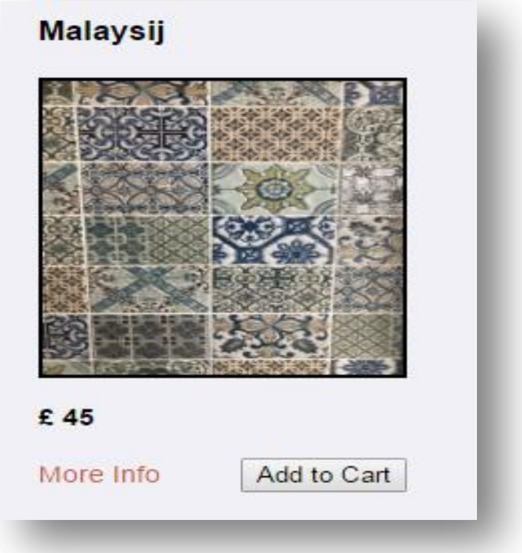
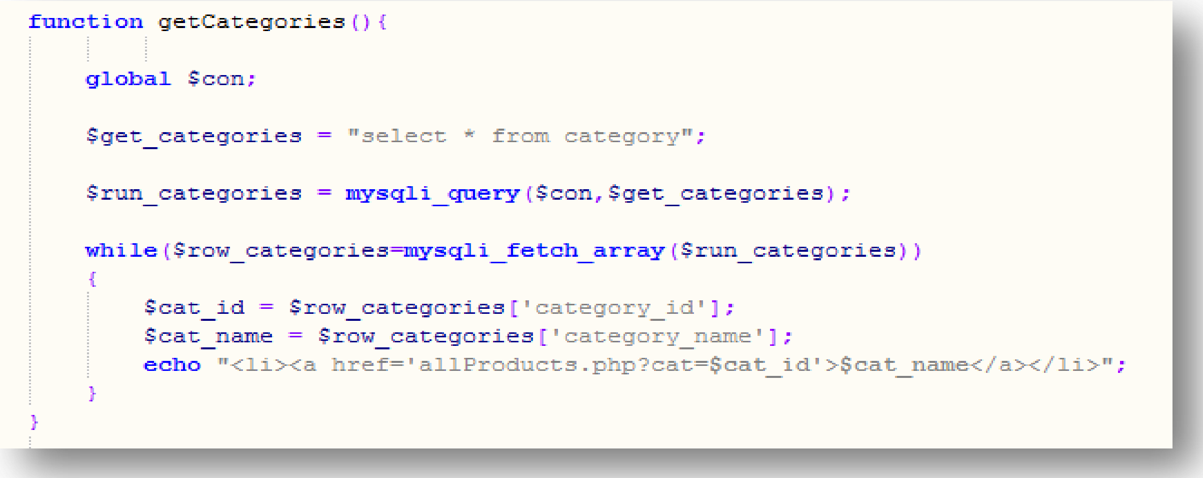


Figure 39 - Product front-end look on all products page

6.3.4. Choose product category (C4 Functional Requirement)

The function “getCategories” below is used to retrieve all the different types of categories stored in the database below. Only the administrator with access to the 1 and 1 panel is able to insert categories into the database table. Categories will remian static throughout the products page on the website. The function “getCategories” work by the use of a “while loop” which fetches all the information in the categories table.



```
function getCategories(){

    global $con;

    $get_categories = "select * from category";

    $run_categories = mysqli_query($con,$get_categories);

    while($row_categories=mysqli_fetch_array($run_categories))
    {
        $cat_id = $row_categories['category_id'];
        $cat_name = $row_categories['category_name'];
        echo "<li><a href='allProducts.php?cat=$cat_id'>$cat_name</a></li>";
    }
}
```

Figure 40 - PHP "getCategories" function which gets the category that each product belongs too

Part of the viewing of the products, a side bar will be present in all pages as showing in the design stage. This will allow the users to redirect to the specific product category as required. Every product will belong to a specific category.

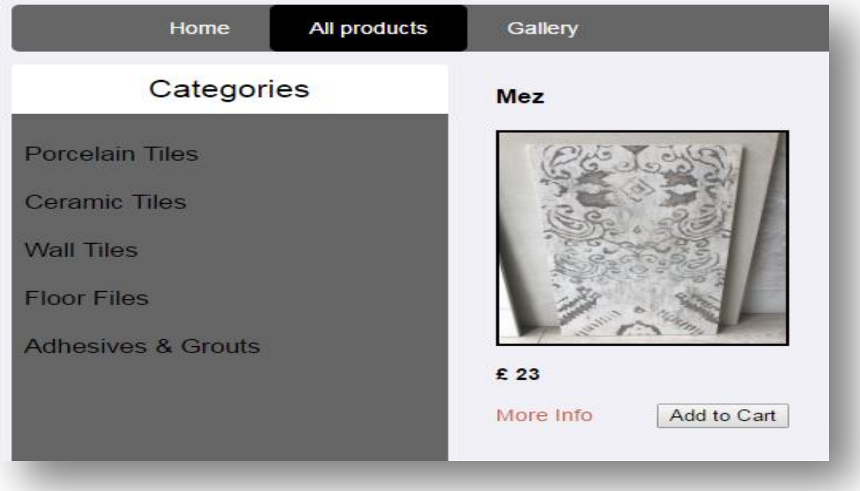
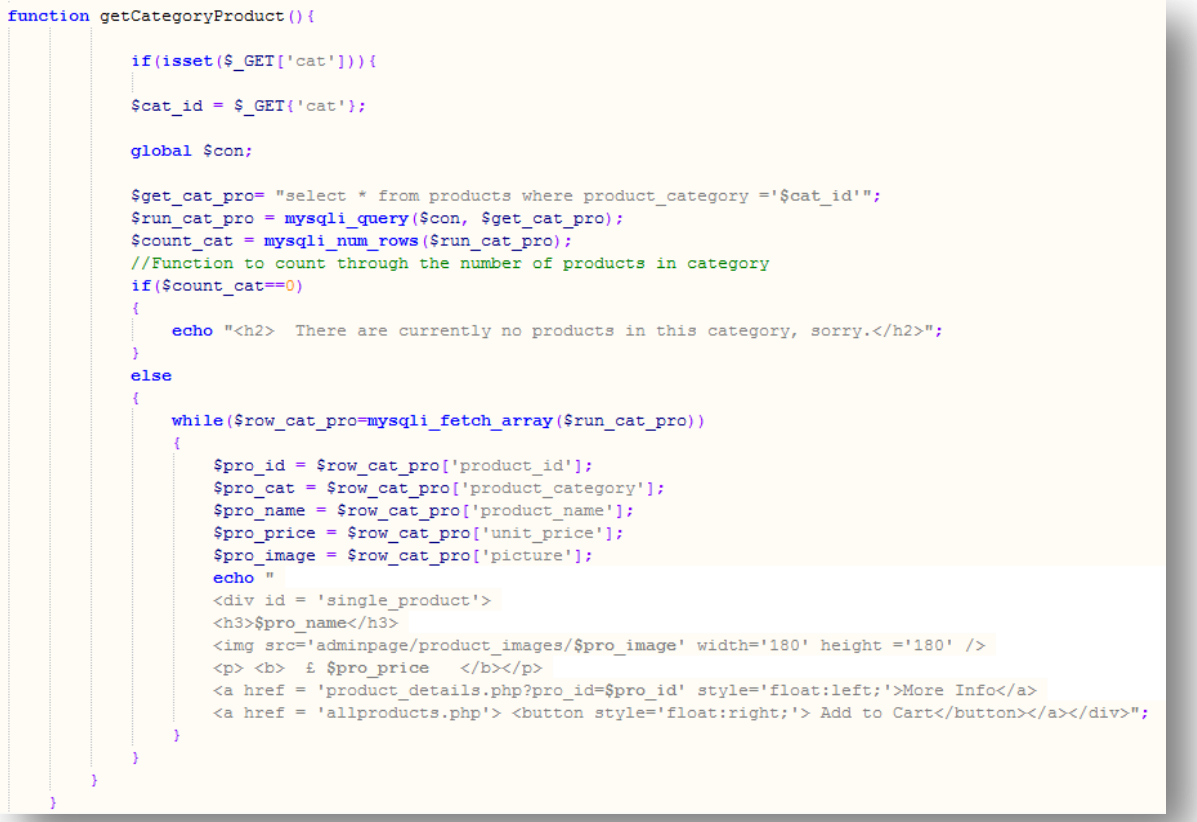


Figure 41 - Front-end view of categories and product

Figure 42 below is responsible for the assigning of different products to the different categories. This PHP if statement assigns the "$cat_id" variable to the "category_id" which was created in the "getCategories" function. A SQL query is then used to select all products where the "product_category" equals the "$cat_id". An array is then used to print out all the data in a "single_product" div element to given the look shown in figure 41.

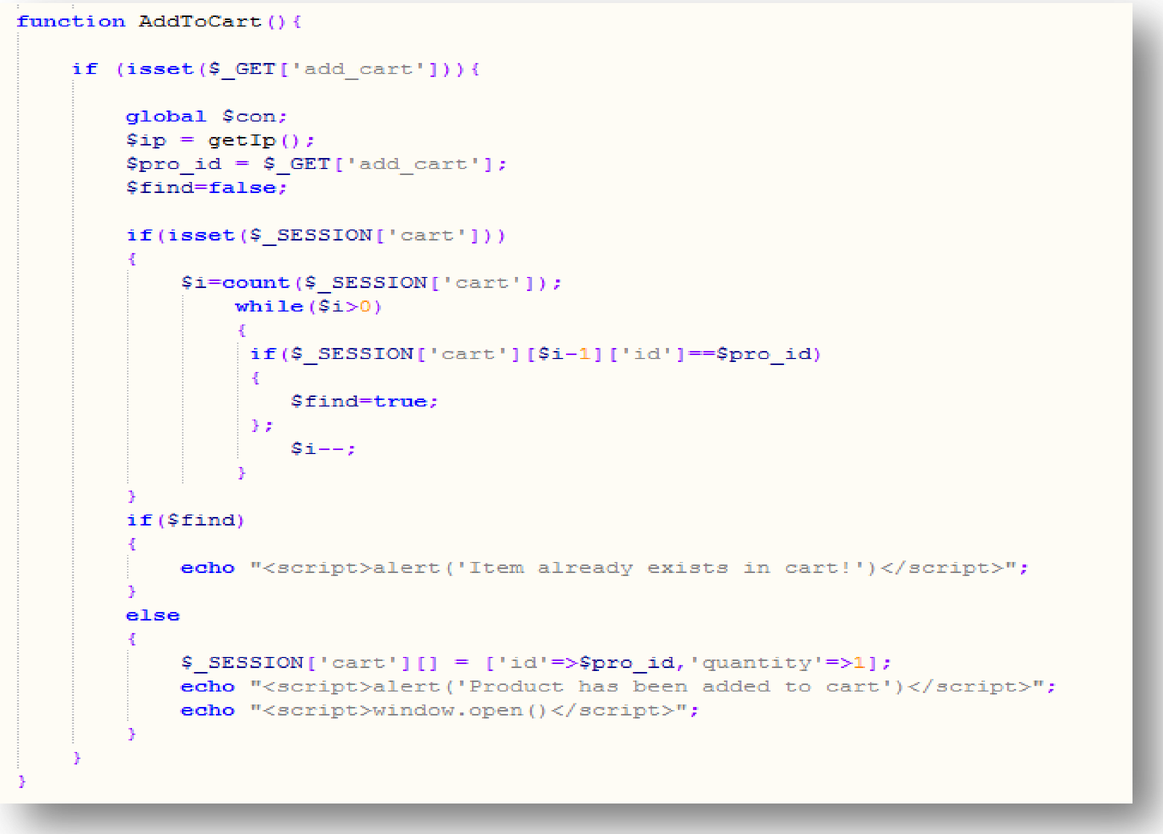


```php
function getCategoryProduct(){

            if(isset($_GET['cat'])){

            $cat_id = $_GET{'cat'};

            global $con;

            $get_cat_pro= "select * from products where product_category ='$cat_id'";
            $run_cat_pro = mysqli_query($con, $get_cat_pro);
            $count_cat = mysqli_num_rows($run_cat_pro);
            //Function to count through the number of products in category
            if($count_cat==0)
            {
                echo "<h2>  There are currently no products in this category, sorry.</h2>";
            }
            else
            {
                while($row_cat_pro=mysqli_fetch_array($run_cat_pro))
                {
                    $pro_id = $row_cat_pro['product_id'];
                    $pro_cat = $row_cat_pro['product_category'];
                    $pro_name = $row_cat_pro['product_name'];
                    $pro_price = $row_cat_pro['unit_price'];
                    $pro_image = $row_cat_pro['picture'];
                    echo "
                    <div id = 'single_product'>
                    <h3>$pro_name</h3>
                    <img src='adminpage/product_images/$pro_image' width='180' height ='180' />
                    <p> <b>  £ $pro_price   </b></p>
                    <a href = 'product_details.php?pro_id=$pro_id' style='float:left;'>More Info</a>
                    <a href = 'allproducts.php'> <button style='float:right;'> Add to Cart</button></a></div>";
                }
            }
        }
    }
```

Figure 42 - The PHP "getCategoryProduct" function which fetches the products responsible for each category

6.3.5. Add Products to Basket (C5 Functional Requirements)

The figure 43 below represents the function which allows the customers to add products to cart. If the customer is to press the “add to cart” button which is assigned to the HTML code “add_cart” shown in figure 38. This function checks if that product is already in the shopping cart based on the current session. The variable “$find” is used to figure out if the cart is empty, a “counter” loops through the products in the session. If the selected product is in the current session which matches the current “$pro_id” which represents the product id, the variable find is then assigned to true, and an alert message is shown. “Else” if the product in the session is less than quantity 1 an alert message is shown and product is added to cart.

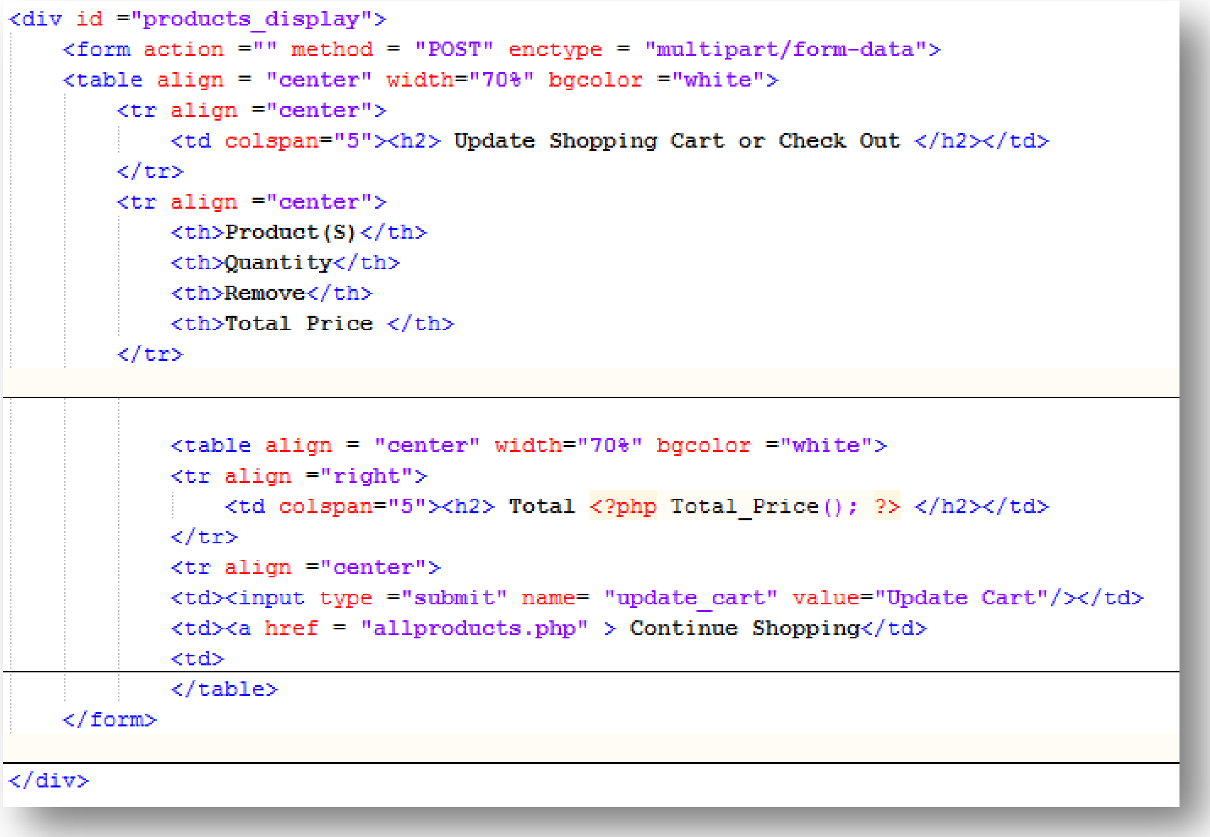


```
function AddToCart(){

    if (isset($_GET['add_cart'])){

        global $con;
        $ip = getIp();
        $pro_id = $_GET['add_cart'];
        $find=false;

        if(isset($_SESSION['cart']))
        {
            $i=count($_SESSION['cart']);
                while($i>0)
                {
                 if($_SESSION['cart'][$i-1]['id']==$pro_id)
                 {
                    $find=true;
                 };
                    $i--;
                }
        }
        if($find)
        {
            echo "<script>alert('Item already exists in cart!')</script>";
        }
        else
        {
            $_SESSION['cart'][] = ['id'=>$pro_id,'quantity'=>1];
            echo "<script>alert('Product has been added to cart')</script>";
            echo "<script>window.open()</script>";
        }
    }
}
```

Figure 43 - PHP "AddToCart" function giving the customer the ability to add products to the shopping cart

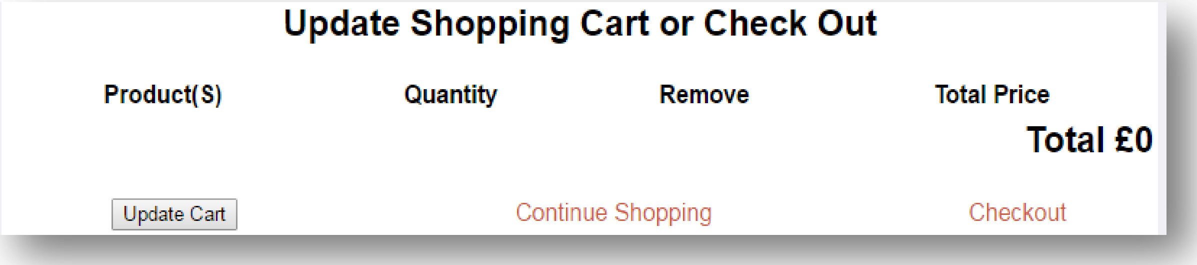
6.3.6. View Basket (C6 Functional Requirements)

When the customers wish to view their baskets, they will have the option to edit the quantity and remove the products from the shopping cart completely. Below figure 44 shows the HTML and PHP code responsible for displaying the products which are in the cart. The HTML below is what displays the products in a specific order. A basic table is created to centre all the products in the middle.



```
<div id ="products_display">
    <form action ="" method = "POST" enctype = "multipart/form-data">
    <table align = "center" width="70%" bgcolor ="white">
        <tr align ="center">
            <td colspan="5"><h2> Update Shopping Cart or Check Out </h2></td>
        </tr>
        <tr align ="center">
            <th>Product(S)</th>
            <th>Quantity</th>
            <th>Remove</th>
            <th>Total Price </th>
        </tr>

            <table align = "center" width="70%" bgcolor ="white">
            <tr align ="right">
                <td colspan="5"><h2> Total <?php Total_Price(); ?> </h2></td>
            </tr>
            <tr align ="center">
            <td><input type ="submit" name= "update_cart" value="Update Cart"/></td>
            <td><a href = "allproducts.php" > Continue Shopping</td>
            <td>
            </table>
    </form>

</div>
```

Figure 44 - HTML code responsible for the shopping cart page



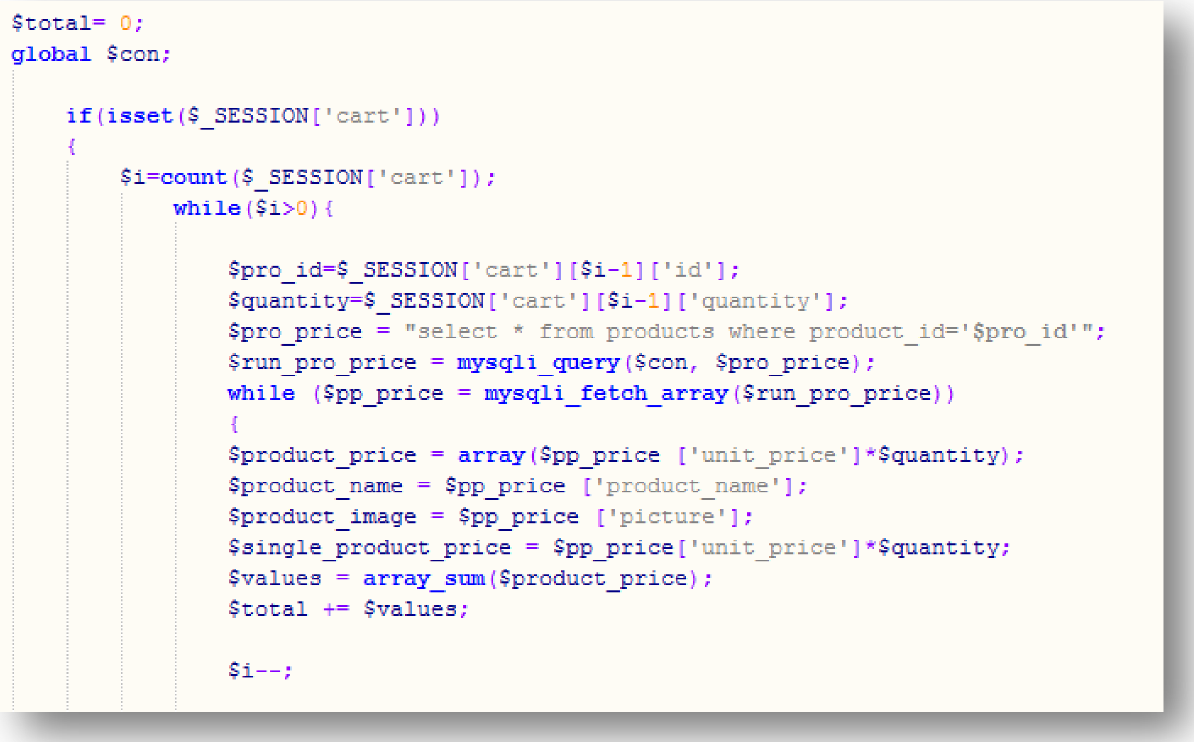
Update Shopping Cart or Check Out

Product(S) Quantity Remove Total Price

Total £0

Figure 45 - Front-end HTML of an empty shopping cart, ability to view quantity, product name, price and total cart price

The PHP function below showing in figure 46 is used to read all the products in the “cart session” and display it in the shopping cart page. A counter is assigned to the variable “$i” which counts through the available cart sessions. Products are then selected based on the “$pro_id” which was used in figure 43 when adding the products to the shopping cart. The variable “$pp_price” is then used to match the product id to retrieve an array of the product price, name, and image through the use of a “while loop”. The products in the cart session are then displayed in the div element above as shown in figure 44.

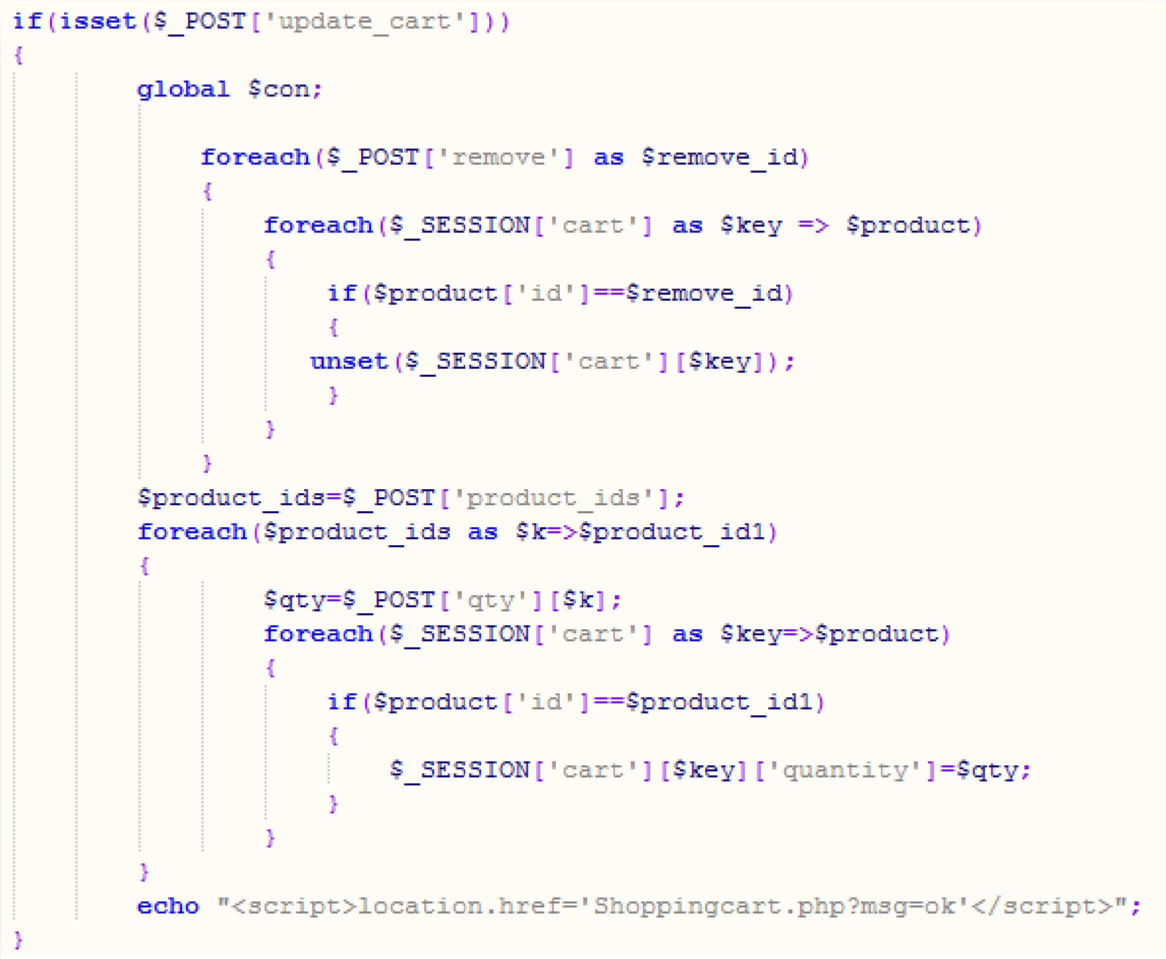


```
$total= 0;
global $con;

    if(isset($_SESSION['cart']))
    {
        $i=count($_SESSION['cart']);
            while($i>0){

                $pro_id=$_SESSION['cart'][$i-1]['id'];
                $quantity=$_SESSION['cart'][$i-1]['quantity'];
                $pro_price = "select * from products where product_id='$pro_id'";
                $run_pro_price = mysqli_query($con, $pro_price);
                while ($pp_price = mysqli_fetch_array($run_pro_price))
                {
                $product_price = array($pp_price ['unit_price']*$quantity);
                $product_name = $pp_price ['product_name'];
                $product_image = $pp_price ['picture'];
                $single_product_price = $pp_price['unit_price']*$quantity;
                $values = array_sum($product_price);
                $total += $values;

                $i--;
```

Figure 46 - PHP code responsible for displaying the products information in the shopping cart

6.3.7. Edit basket (C7 Functional Requirements)

The PHP code below works when the customer needs to update the cart. Removing or changing the quantity of any given product in the cart is a functional requirement. If a user is to check the remove box or change the quantity and click update cart, a message should appear upon success and the changes should be visible.



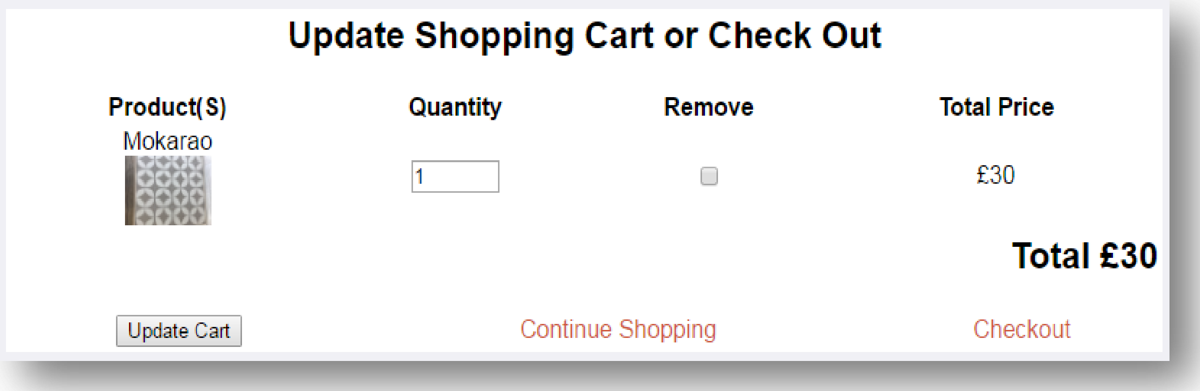
```php
if(isset($_POST['update_cart']))
{
        global $con;

            foreach($_POST['remove'] as $remove_id)
            {
                foreach($_SESSION['cart'] as $key => $product)
                {
                    if($product['id']==$remove_id)
                    {
                   unset($_SESSION['cart'][$key]);
                    }
                }
            }
        $product_ids=$_POST['product_ids'];
        foreach($product_ids as $k=>$product_id1)
        {
                $qty=$_POST['qty'][$k];
                foreach($_SESSION['cart'] as $key=>$product)
                {
                    if($product['id']==$product_id1)
                    {
                        $_SESSION['cart'][$key]['quantity']=$qty;
                    }
                }
        }
        echo "<script>location.href='Shoppingcart.php?msg=ok'</script>";
}
```

Figure 47 - PHP code if the customer wishes to make changes to the products in their shopping cart

As shown in figure 47, “update_cart” is the name of the input type shown in figure 44. When a user presses on the update cart button, the following code will be executed. Upon the checking of the remove checkbox, a “for each loop” is used which assigns the “remove” array to a variable called “$remove_id”. This variable is parsed into an “if” statement which checks if the “$product_id” equals the check boxed product and if so, then the product is then removed from that particular “cart session”.

As for the quantity, it operates in the same way, but instead of unsettling the session it updates the quantity chosen in the “$qty2” variable and the neq quantity is assigned to the current session cart array.

The figures below show the changing of the quantity for the Mokarao product on offer. The quantity is always registered as one when a product is added to the cart.



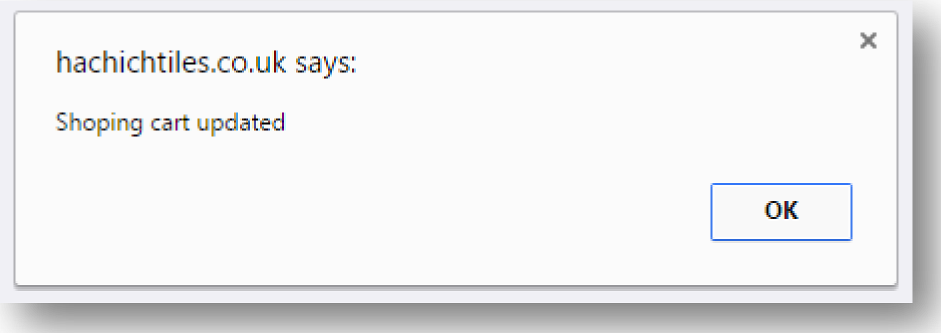
Update Shopping Cart or Check Out

Product(S)	Quantity	Remove	Total Price
Mokarao	1	□	£30

Total £30

Figure 48 - Front-end of the shopping cart page with one product of one quality

A pop-up script is then presented notifying the customer that the cart has been updated

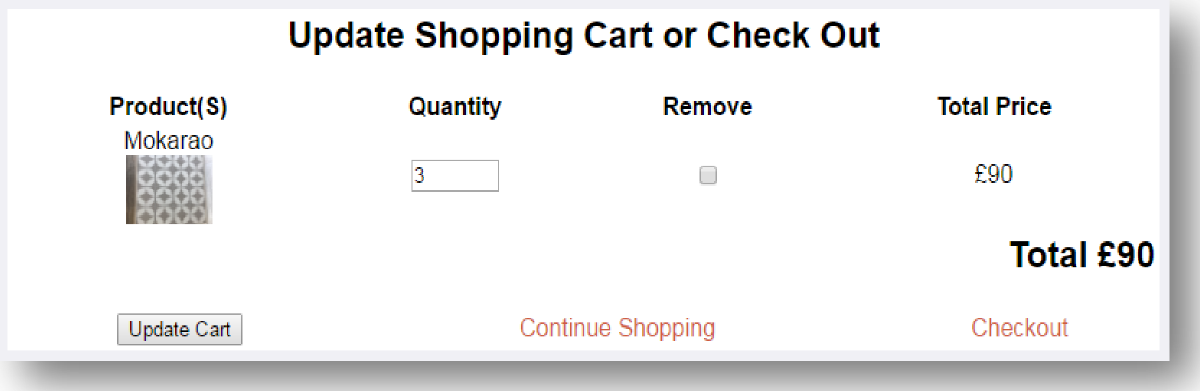


hachichtiles.co.uk says:

Shoping cart updated

Figure 49 - Alert message when changes have been made to the shopping cart

A quantity change from 1 product to 3 products is shown below. The total price is also shown for each product along with the total for all products.



Update Shopping Cart or Check Out

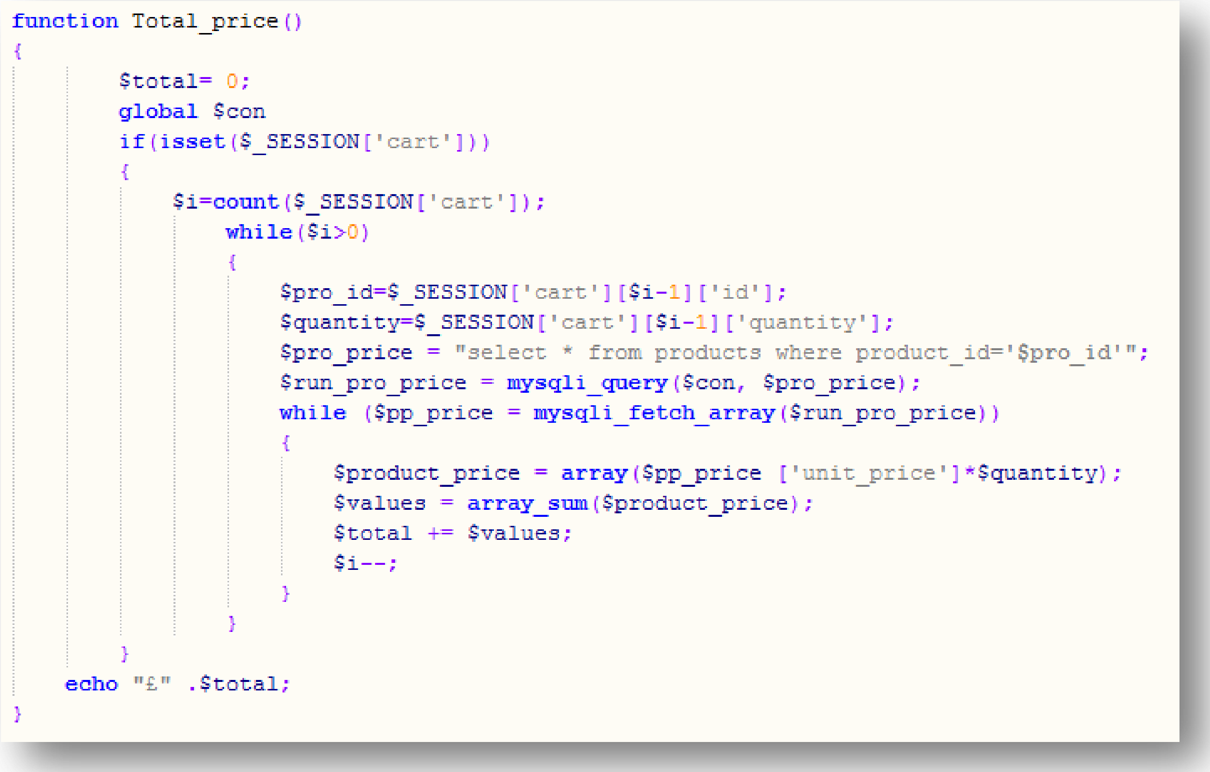
Product(S)	Quantity	Remove	Total Price
Mokarao	3	□	£90

Total £90

Figure 50 - Change in quantity for the shopping cart from 1 to 3

6.3.8. View Basket Total (C8 Functional Requirements)

For the customer to be able to see the total price of all the different products which are in the current “cart session”, the function “Total_Price” was created. The function starts by defining a variable called “$total” to zero. If the “$_SESSION” array is assigned to “cart” (which it is) then a counter is assigned to the variable “$i” which counts the number of sessions in the array. As the session is present, the variable “$i” will be greater than 0 and the code below will be executed. All the products in the shopping cart will be selected through the use of a SQL statement, this is followed by a “while loop” which assigns the variable “$product_price” to an array which multiplies the “unit_price” by the quantity.



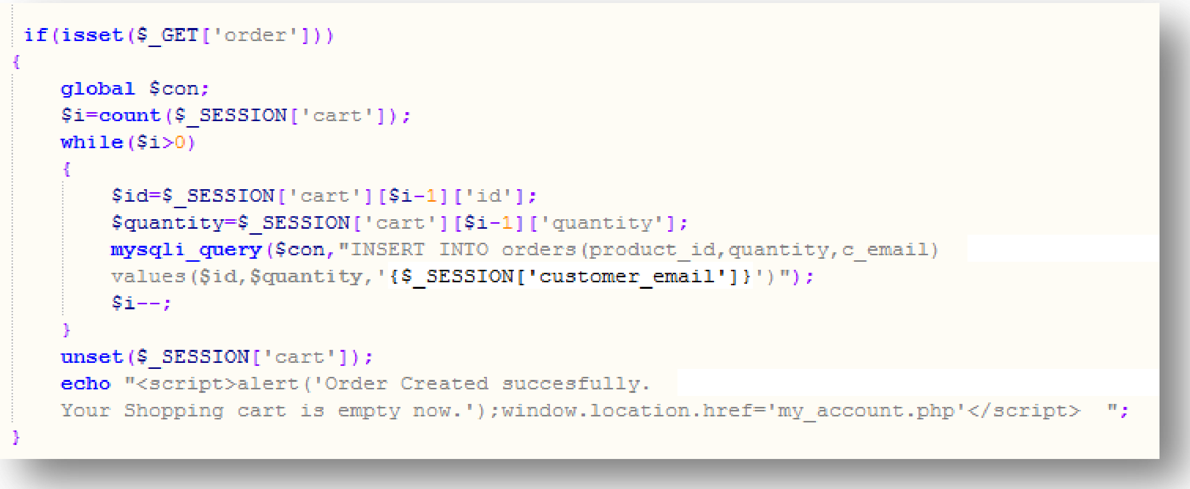
```
function Total_price()
{
        $total= 0;
        global $con
        if(isset($_SESSION['cart']))
        {
            $i=count($_SESSION['cart']);
                while($i>0)
                {
                    $pro_id=$_SESSION['cart'][$i-1]['id'];
                    $quantity=$_SESSION['cart'][$i-1]['quantity'];
                    $pro_price = "select * from products where product_id='$pro_id'";
                    $run_pro_price = mysqli_query($con, $pro_price);
                    while ($pp_price = mysqli_fetch_array($run_pro_price))
                    {
                        $product_price = array($pp_price ['unit_price']*$quantity);
                        $values = array_sum($product_price);
                        $total += $values;
                        $i--;
                    }
                }
        }
    echo "£" .$total;
}
```

Figure 51 - PHP function called “Total_price” which calculates the price of all products

6.3.9. Checkout (C9 Functional Requirement)

Then it came to the implementation of this function, the client wanted customers to commence all their transactions securely through the website. The initial plan was to allow all users to make payment via PayPal. As the project came together, the client changed his mind and wanted all payments to be processed by cash upon delivery of the products or as a bank transfer.

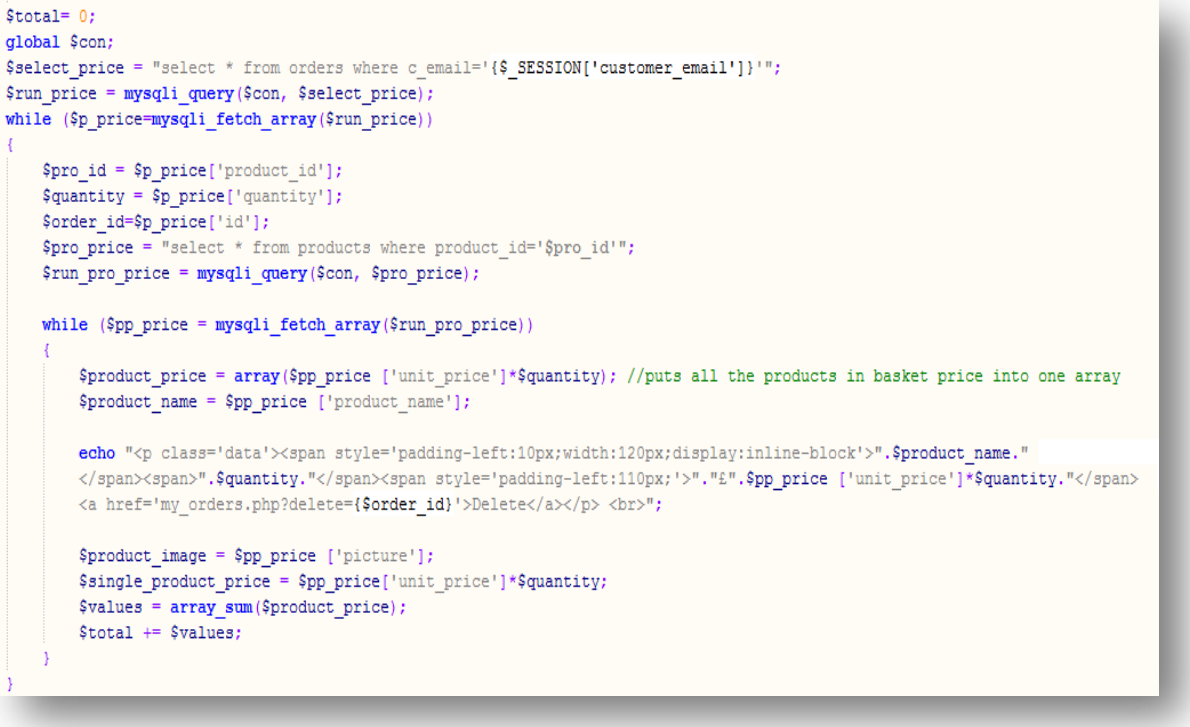
Therefore, figure 52 below shows how the customer will be able to checkout with the selected products, without making any payments on the website. The admin will then be able to view the order and contact the customer.



```
if(isset($_GET['order']))
{
    global $con;
    $i=count($_SESSION['cart']);
    while($i>0)
    {
        $id=$_SESSION['cart'][$i-1]['id'];
        $quantity=$_SESSION['cart'][$i-1]['quantity'];
        mysqli_query($con,"INSERT INTO orders(product_id,quantity,c_email)
        values($id,$quantity,'{$_SESSION['customer_email']}')");
        $i--;
    }
    unset($_SESSION['cart']);
    echo "<script>alert('Order Created succesfully.
    Your Shopping cart is empty now.');window.location.href='my_account.php'</script>  ";
}
```

Figure 52 - PHP code for storing the shopping cart information when checking out in the orders database table

6.3.10. View Order Details (C10 Functional Requirements)



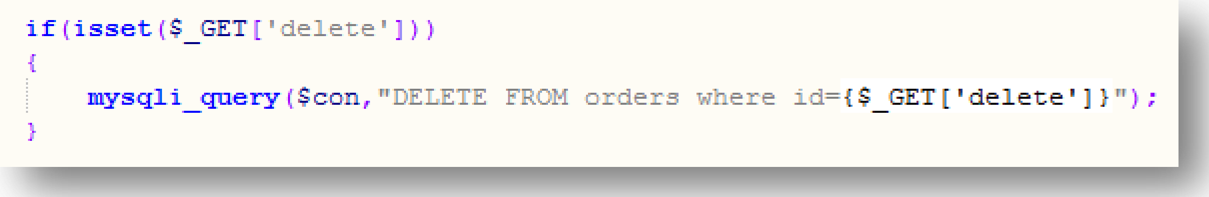
```
$total= 0;
global $con;
$select_price = "select * from orders where c_email='{$_SESSION['customer_email']}'";
$run_price = mysqli_query($con, $select_price);
while ($p_price=mysqli_fetch_array($run_price))
{
    $pro_id = $p_price['product_id'];
    $quantity = $p_price['quantity'];
    $order_id=$p_price['id'];
    $pro_price = "select * from products where product_id='$pro_id'";
    $run_pro_price = mysqli_query($con, $pro_price);

    while ($pp_price = mysqli_fetch_array($run_pro_price))
    {
        $product_price = array($pp_price ['unit_price']*$quantity); //puts all the products in basket price into one array
        $product_name = $pp_price ['product_name'];

        echo "<p class='data'><span style='padding-left:10px;width:120px;display:inline-block'>".$product_name."
        </span><span>".$quantity."</span><span style='padding-left:110px;'>"."£".$pp_price ['unit_price']*$quantity."</span>
        <a href='my_orders.php?delete={$order_id}'>Delete</a></p> <br>";

        $product_image = $pp_price ['picture'];
        $single_product_price = $pp_price['unit_price']*$quantity;
        $values = array_sum($product_price);
        $total += $values;
    }
}
```

Figure 53 – Extracting customers’ orders based on sessions which are assigned to their email.

The PHP above is responsible for displaying all the products that have been ordered on a separate page. A SQL statement is used to select all the orders where the “c_email” equals the email address of the one that the customer is currently logged in with. This is stored in a session array from when the customer logs in. “Mysqli_fetch_array” is then used to select all the information from the orders table and assign each one to its individual variable. An inner while loop is used to insert an array of the unit price multiplied by the quantity into a variable called “$product_price”.

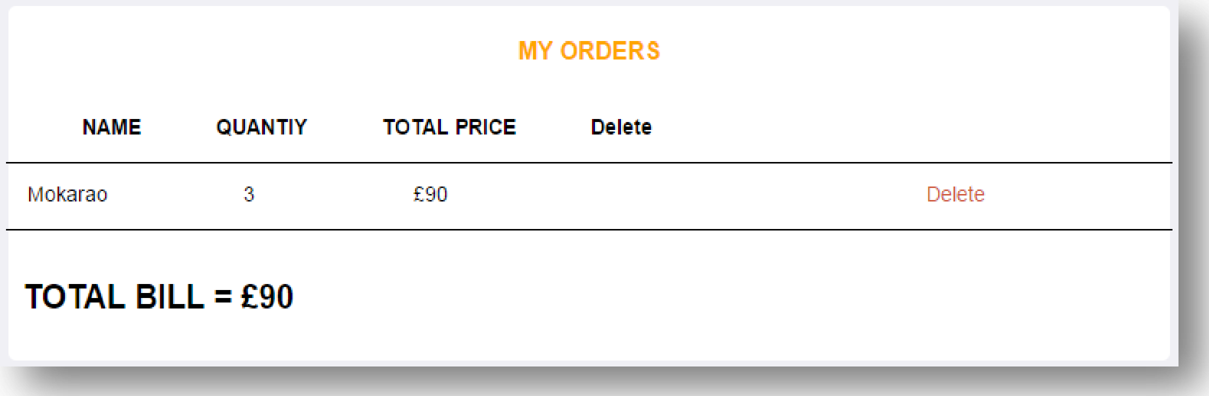
6.3.11. Delete Order (C11 Functional Requirements)

The figure below shows the code responsible for the removing of a product from the orders page. When the delete button is pressed a SQL statement is used to delete the order from the orders table.



```
if(isset($_GET['delete']))
{
    mysqli_query($con,"DELETE FROM orders where id={$_GET['delete']}");
}
```

Figure 54 - Customer deleting orders



MY ORDERS

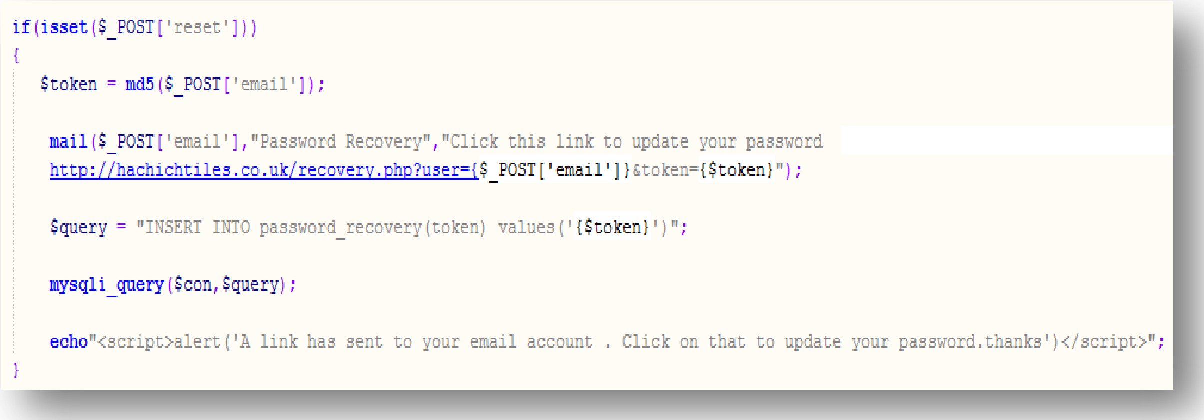
NAME	QUANTIY	TOTAL PRICE	Delete	
Mokarao	3	£90		Delete

TOTAL BILL = £90

Figure 55 - Front-end view of orders

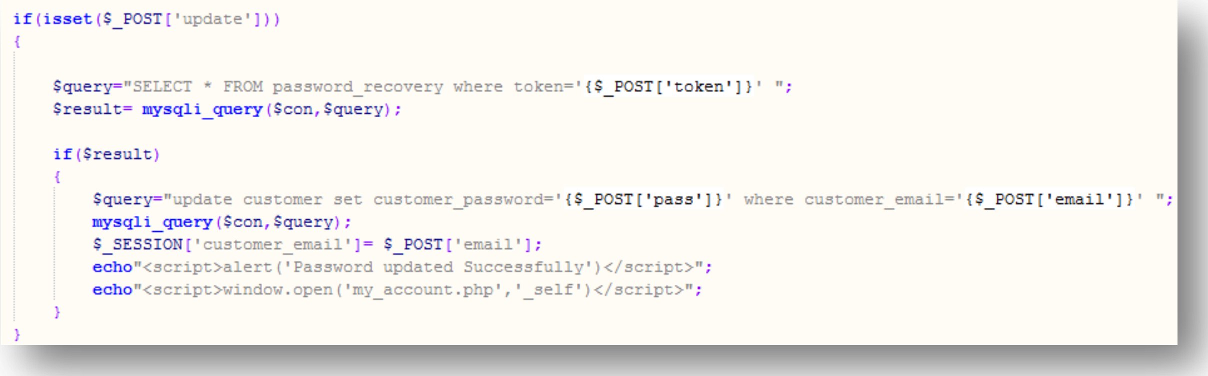
6.3.12. Forgot Password (C12 Functional Requirements)

The PHP code below shown in figure 56 is responsible for sending the user an email to allow them to change their password.



```
if(isset($_POST['reset']))
{
  $token = md5($_POST['email']);

   mail($_POST['email'],"Password Recovery","Click this link to update your password
   http://hachichtiles.co.uk/recovery.php?user={$_POST['email']}&token={$token}");

   $query = "INSERT INTO password_recovery(token) values('{$token}')";

   mysqli_query($con,$query);

   echo"<script>alert('A link has sent to your email account . Click on that to update your password.thanks')</script>";
}
```

Figure 56 - PHP code to reset password using md5



```
if(isset($_POST['update']))
{

    $query="SELECT * FROM password_recovery where token='{$_POST['token']}' ";
    $result= mysqli_query($con,$query);

    if($result)
    {
        $query="update customer set customer_password='{$_POST['pass']}' where customer_email='{$_POST['email']}' ";
        mysqli_query($con,$query);
        $_SESSION['customer_email']= $_POST['email'];
        echo"<script>alert('Password updated Successfully')</script>";
        echo"<script>window.open('my_account.php','_self')</script>";
    }
}
```

Figure 57 - PHP code for when customer clicks the update button

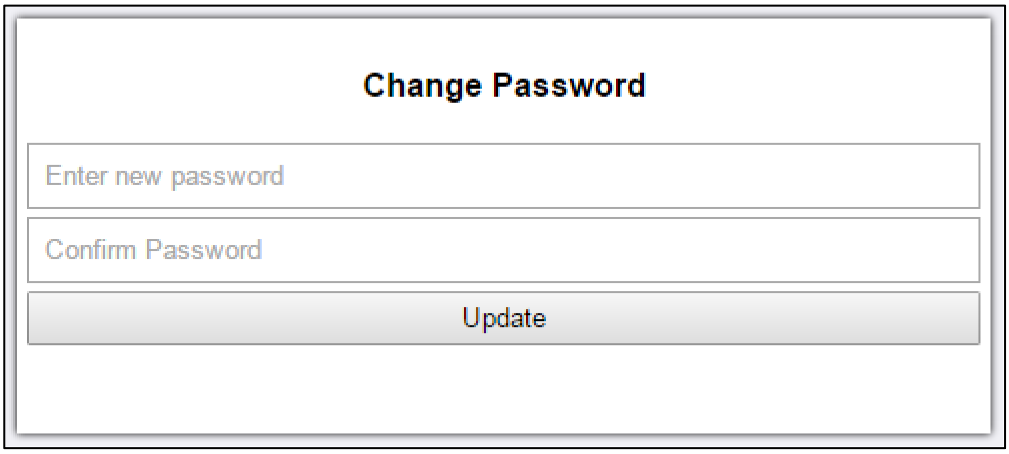
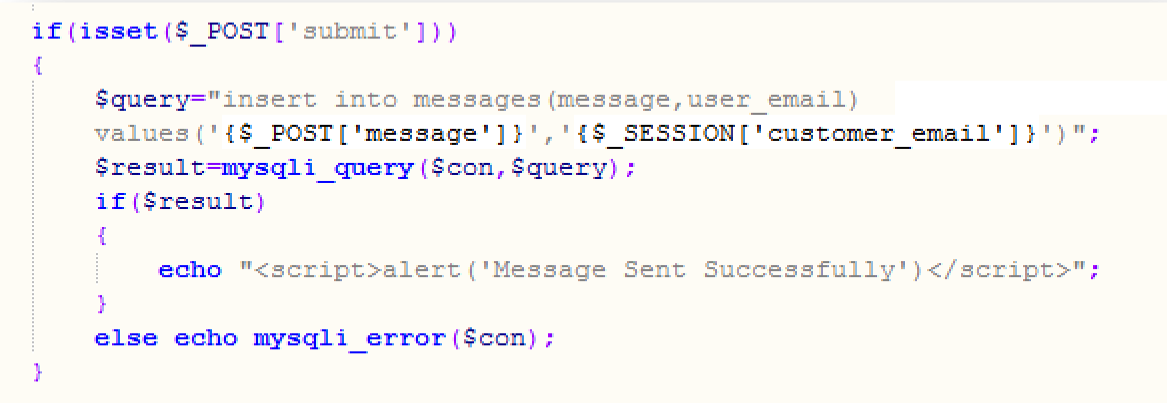


Figure 58 - Front-end for the forgotten password code after clicking on the email link.

6.3.13. Message seller for stock enquiry (C15 Functional Requirements)

The code below is responsible for sending messages to the admin. These messages are stored in the messages table in the database. After pressing the send button, an alert pop-up is presented and the message is successfully stored.



```
if(isset($_POST['submit']))
{
    $query="insert into messages(message,user_email)
    values('{$_POST['message']}','{$_SESSION['customer_email']}')";
    $result=mysqli_query($con,$query);
    if($result)
    {
        echo "<script>alert('Message Sent Successfully')</script>";
    }
    else echo mysqli_error($con);
}
```

Figure 59 - PHP code for sending message to seller

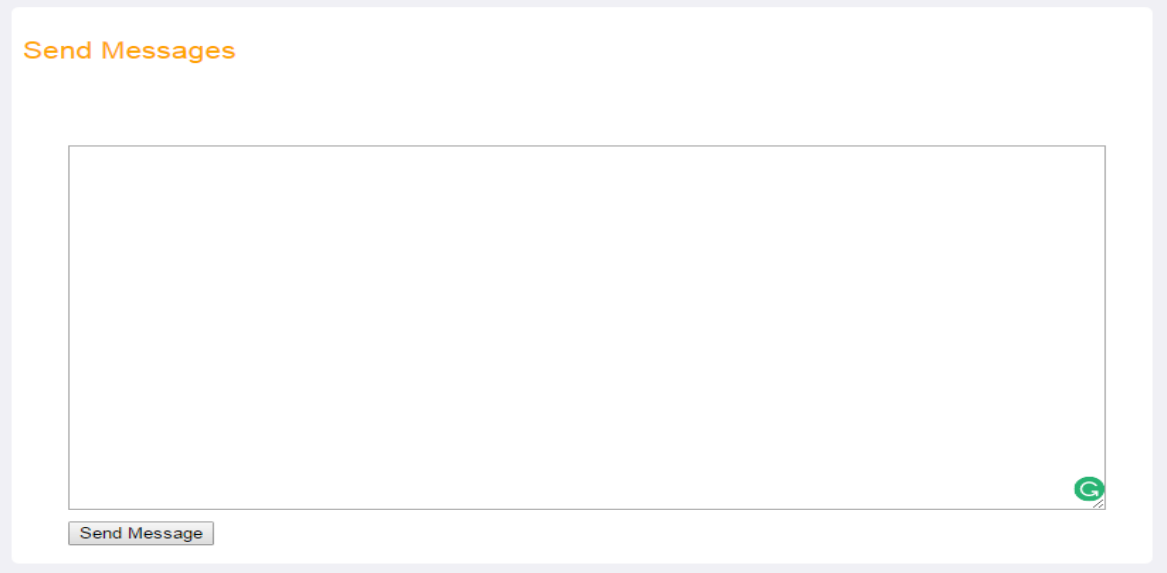


Figure 60 - Front-end for sending a message

6.3.14. Admin Panel

When the admin logs in with the correct credentials they are redirected to the homepage with a different view to the customer.

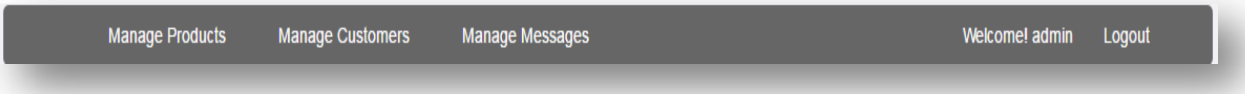
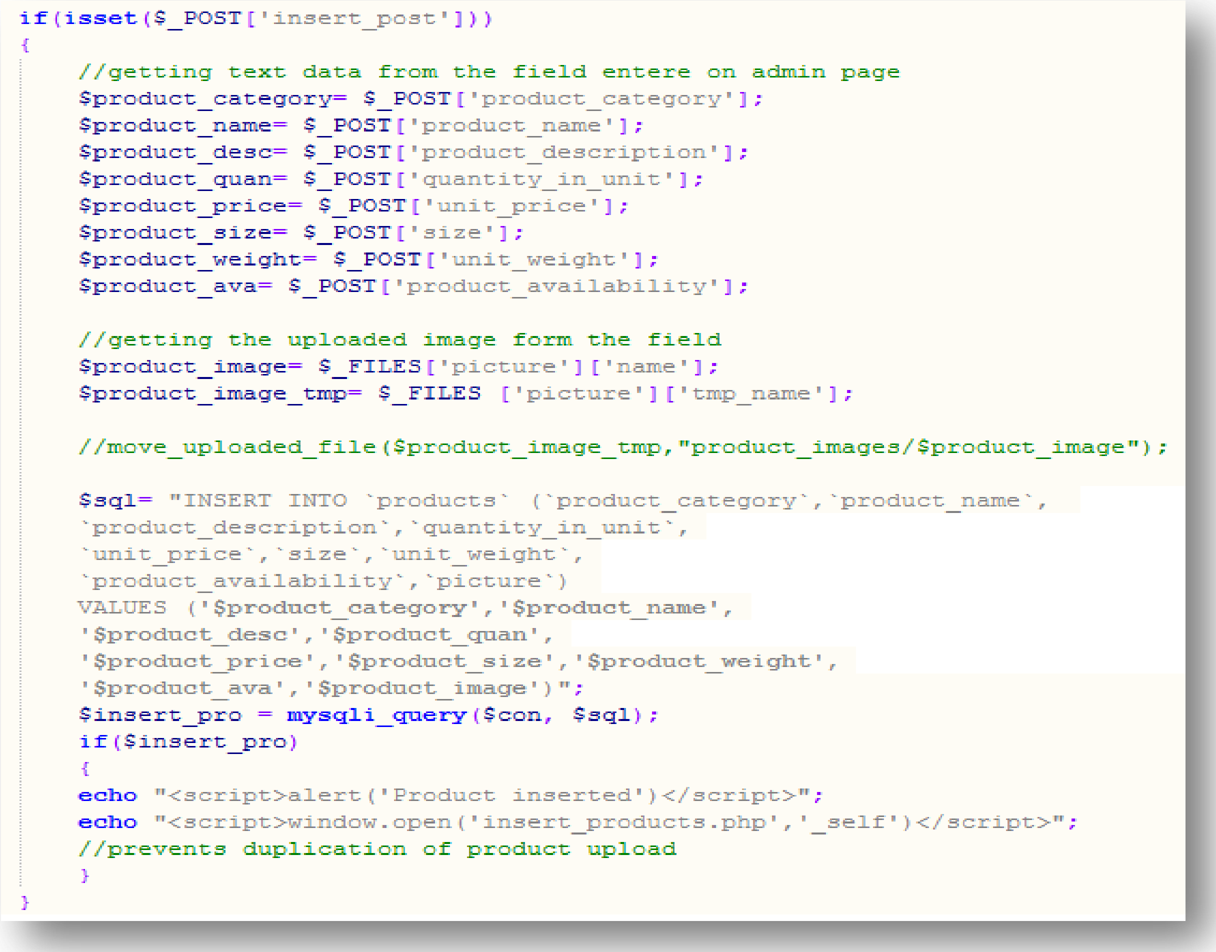


Figure 61 - Front-end for admin navigation bar

6.3.15. Add & remove products from catalogue (A2 Functional Requirements)

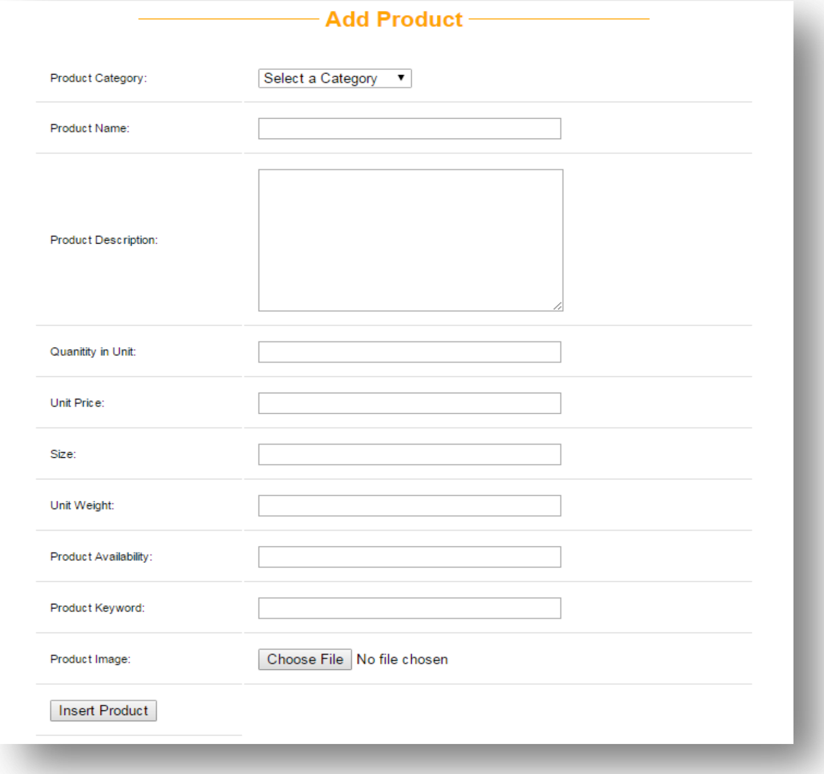
The code below is responsible for the adding of the products to the website catalogue. If the admin presses on the insert product button all the code in the “if” statement gets executed. Variables are created which are assigned to the field names of the HTML form. A SQL statement is then used to upload the entered data into the products table. An alert is then displayed to inform the admin that the process has successfully worked.



```
if(isset($_POST['insert_post']))
{
    //getting text data from the field entere on admin page
    $product_category= $_POST['product_category'];
    $product_name= $_POST['product_name'];
    $product_desc= $_POST['product_description'];
    $product_quan= $_POST['quantity_in_unit'];
    $product_price= $_POST['unit_price'];
    $product_size= $_POST['size'];
    $product_weight= $_POST['unit_weight'];
    $product_ava= $_POST['product_availability'];

    //getting the uploaded image form the field
    $product_image= $_FILES['picture']['name'];
    $product_image_tmp= $_FILES ['picture']['tmp_name'];

    //move_uploaded_file($product_image_tmp,"product_images/$product_image");

    $sql= "INSERT INTO `products` (`product_category`,`product_name`,
    `product_description`,`quantity_in_unit`,
    `unit_price`,`size`,`unit_weight`,
    `product_availability`,`picture`)
    VALUES ('$product_category','$product_name',
    '$product_desc','$product_quan',
    '$product_price','$product_size','$product_weight',
    '$product_ava','$product_image')";
    $insert_pro = mysqli_query($con, $sql);
    if($insert_pro)
    {
    echo "<script>alert('Product inserted')</script>";
    echo "<script>window.open('insert_products.php','_self')</script>";
    //prevents duplication of product upload
    }
}
```

Figure 62 - PHP code for inserting a product

The way the product is added will look like the below. Part of adding the product the admin can choose the category that the product belongs too.

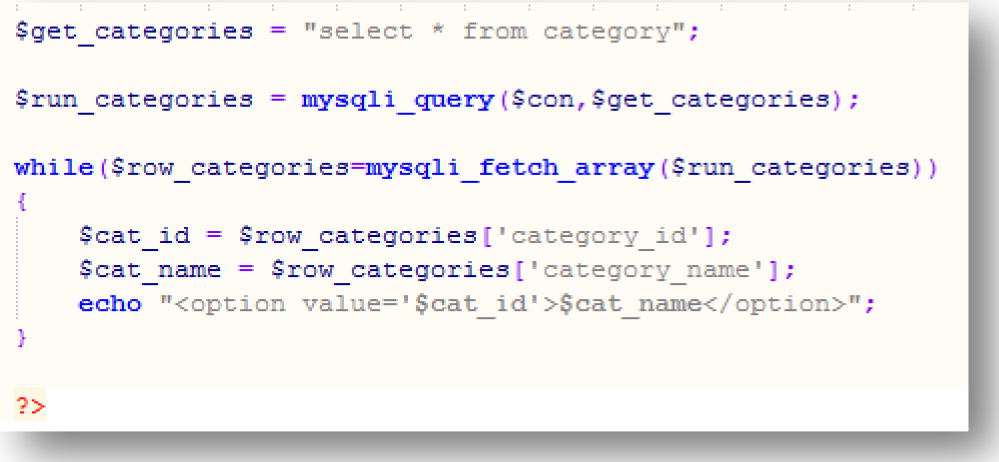


Add Product

Product Category:	Select a Category ▾
Product Name:	
Product Description:	
Quantity in Unit:	
Unit Price:	
Size:	
Unit Weight:	
Product Availability:	
Product Keyword:	
Product Image:	Choose File No file chosen

Figure 63 - Front-end for adding a product

Selecting the categories using a SQL statement and assigning it to a variable called get categories. A While loop is then used to go through the array run categories and assigns every category id and category name in the database to different variables which are then printed to the Admin.

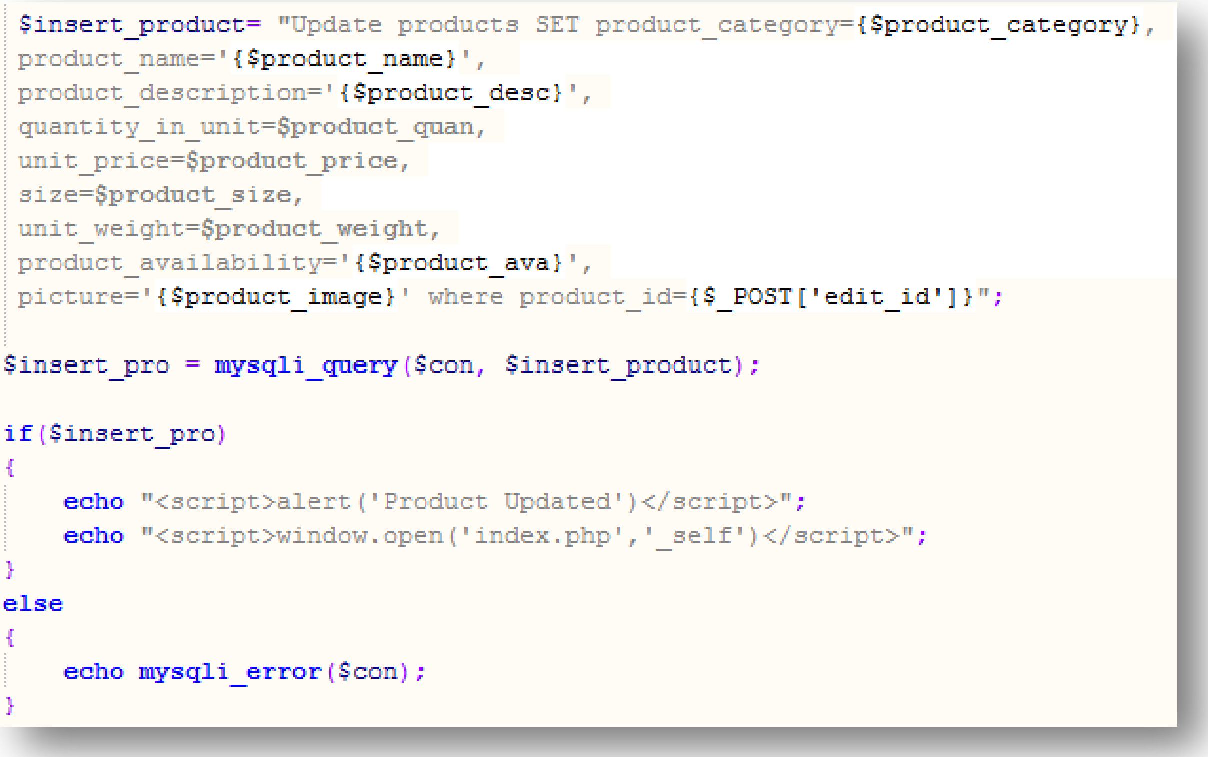


```
$get_categories = "select * from category";

$run_categories = mysqli_query($con,$get_categories);

while($row_categories=mysqli_fetch_array($run_categories))
{
    $cat_id = $row_categories['category_id'];
    $cat_name = $row_categories['category_name'];
    echo "<option value='$cat_id'>$cat_name</option>";
}

?>
```

Figure 64 - Assigning the products to the selected category

6.3.16. Edit Product Information (A3 Functional Requirement)

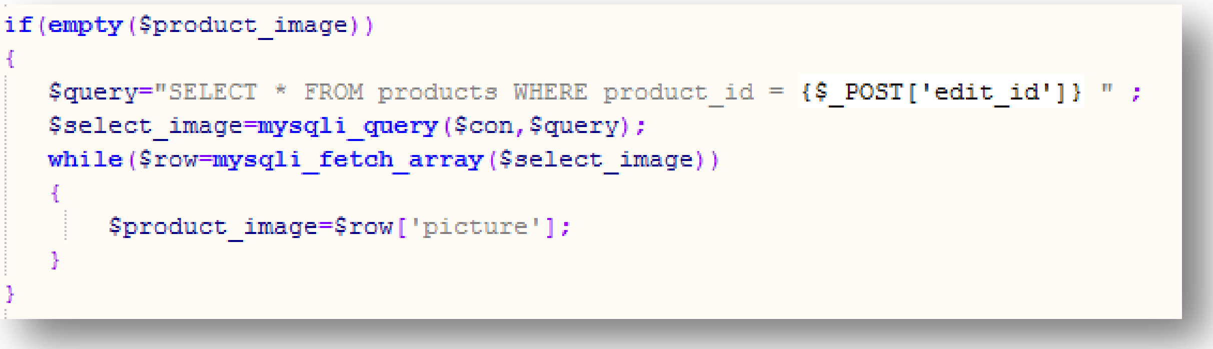
The code below is responsible for the editing of an existing product. A SQL statement is used to set each one of the fields provided for the product when pressing the update button.



```
 $insert_product= "Update products SET product_category={$product_category},
 product_name='{$product_name}',
 product_description='{$product_desc}',
 quantity_in_unit=$product_quan,
 unit_price=$product_price,
 size=$product_size,
 unit_weight=$product_weight,
 product_availability='{$product_ava}',
 picture='{$product_image}' where product_id={$_POST['edit_id']}";

$insert_pro = mysqli_query($con, $insert_product);

if($insert_pro)
{
    echo "<script>alert('Product Updated')</script>";
    echo "<script>window.open('index.php','_self')</script>";
}
else
{
    echo mysqli_error($con);
}
```

Figure 65 - PHP code for updating the products table

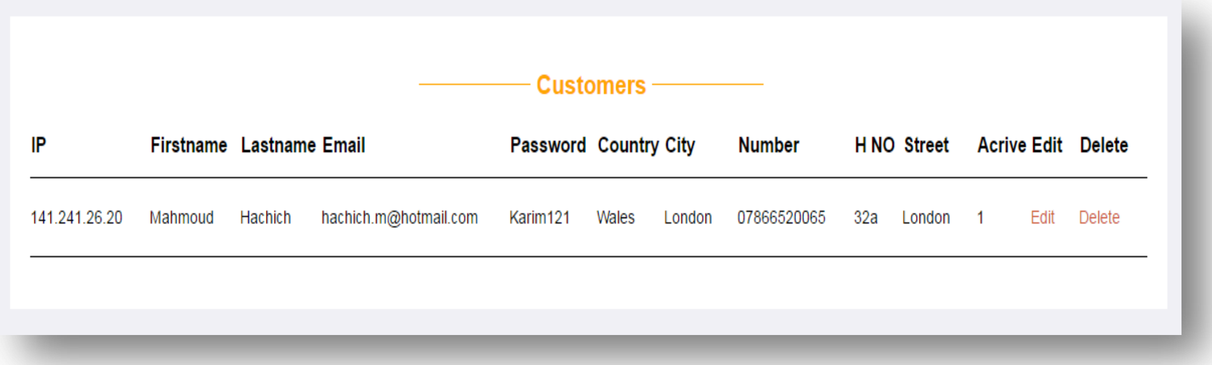
If the admin is not to change the product image, the code below ensures that the previous picture is still present after updating the product.



```
if(empty($product_image))
{
   $query="SELECT * FROM products WHERE product_id = {$_POST['edit_id']} " ;
   $select_image=mysqli_query($con,$query);
   while($row=mysqli_fetch_array($select_image))
   {
       $product_image=$row['picture'];
   }
}
```

Figure 66 - PHP code for keeping the same product image picture when updating products

6.3.17. Manage customer account details (A4 Function Requirement)



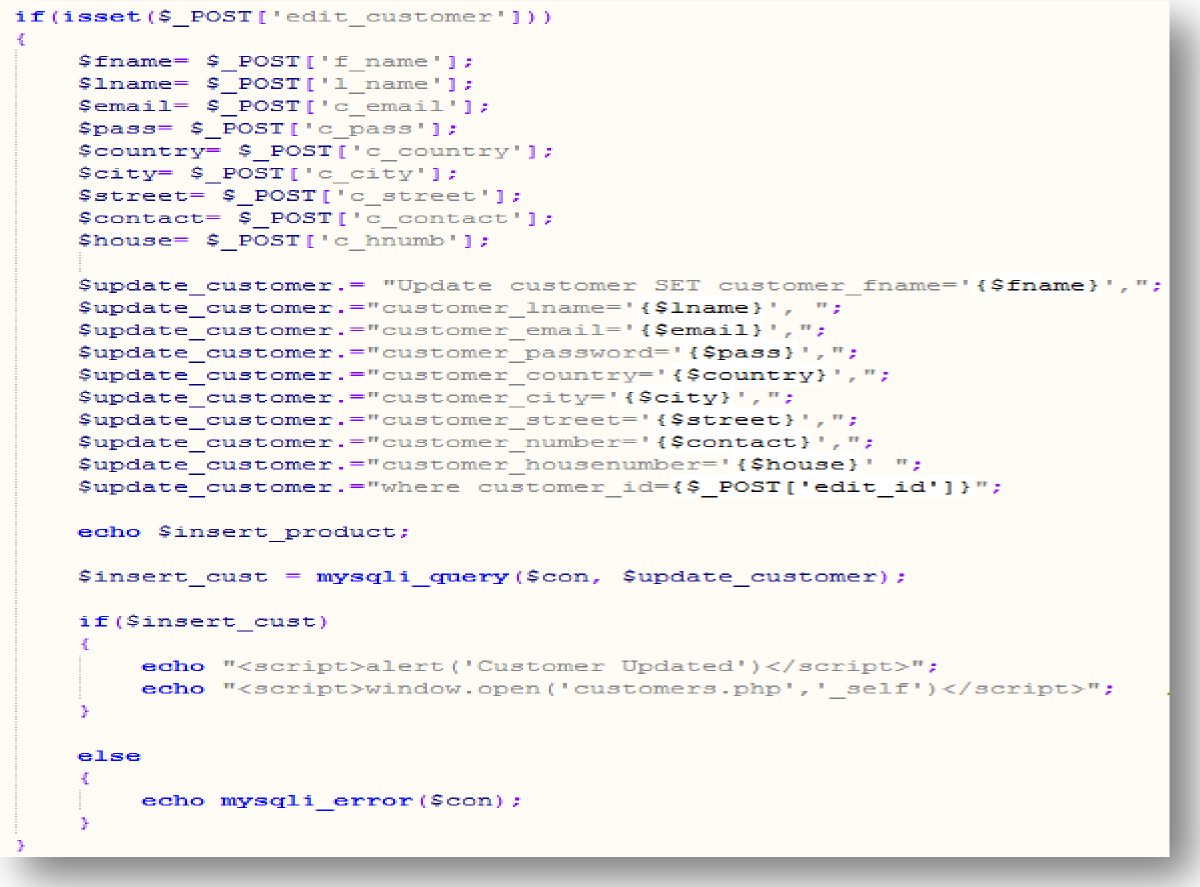
Customers

IP	Firstname	Lastname	Email	Password	Country	City	Number	H NO	Street	Acrive	Edit	Delete
141.241.26.20	Mahmoud	Hachich	hachich.m@hotmail.com	Karim121	Wales	London	07866520065	32a	London	1	Edit	Delete

Figure 67 - Front-end admin view of customers

The code below is responsible for giving the admin access to view and edit customer accounts.

The figure 68 below shows the creating of new variables which are assigned to the input types in the form. A variable called "update_customer" is then created which contains a SQL statement that assigns the columns from the database customer to the variables created earlier.

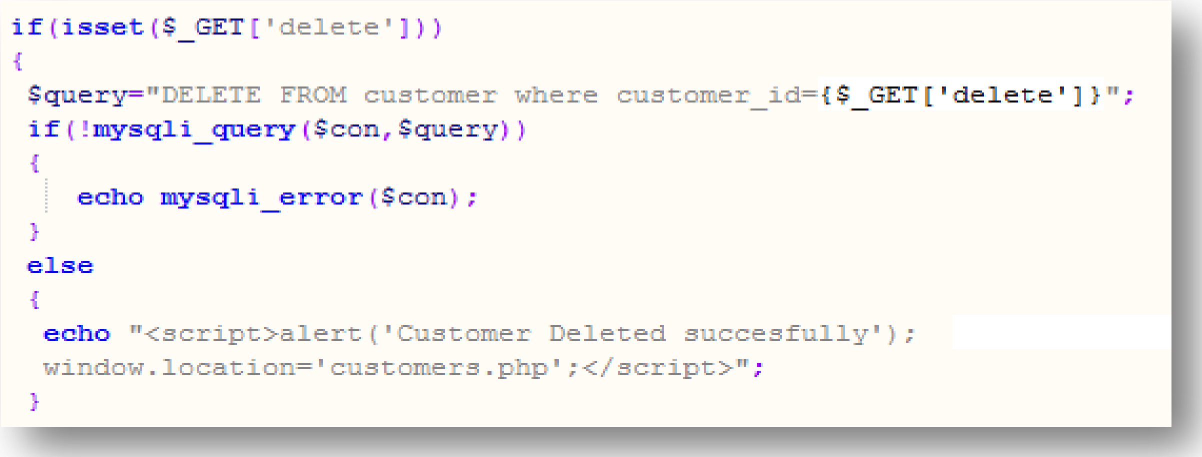


```
if(isset($_POST['edit_customer']))
{
    $fname= $_POST['f_name'];
    $lname= $_POST['l_name'];
    $email= $_POST['c_email'];
    $pass= $_POST['c_pass'];
    $country= $_POST['c_country'];
    $city= $_POST['c_city'];
    $street= $_POST['c_street'];
    $contact= $_POST['c_contact'];
    $house= $_POST['c_hnumb'];

    $update_customer.= "Update customer SET customer_fname='{$fname}',";
    $update_customer.="customer_lname='{$lname}',  ";
    $update_customer.="customer_email='{$email}',";
    $update_customer.="customer_password='{$pass}',";
    $update_customer.="customer_country='{$country}',";
    $update_customer.="customer_city='{$city}',";
    $update_customer.="customer_street='{$street}',";
    $update_customer.="customer_number='{$contact}',";
    $update_customer.="customer_housenumber='{$house}'  ";
    $update_customer.="where customer_id={$_POST['edit_id']}";

    echo $insert_product;

    $insert_cust = mysqli_query($con, $update_customer);

    if($insert_cust)
    {
        echo "<script>alert('Customer Updated')</script>";
        echo "<script>window.open('customers.php','_self')</script>";
    }

    else
    {
        echo mysqli_error($con);
    }
}
```

Figure 68 - PHP code for editing customer details

6.3.18. Delete Customer (A5 Functional Requirement)

The code below is used to delete customer accounts. This function is only present on the control panel of all customers on the admin page. Once an admin selects the delete button a customer will be deleted instantly. A SQL statement is assigned to a variable where all customers whose “customer_id” matches the delete submit button will be removed.

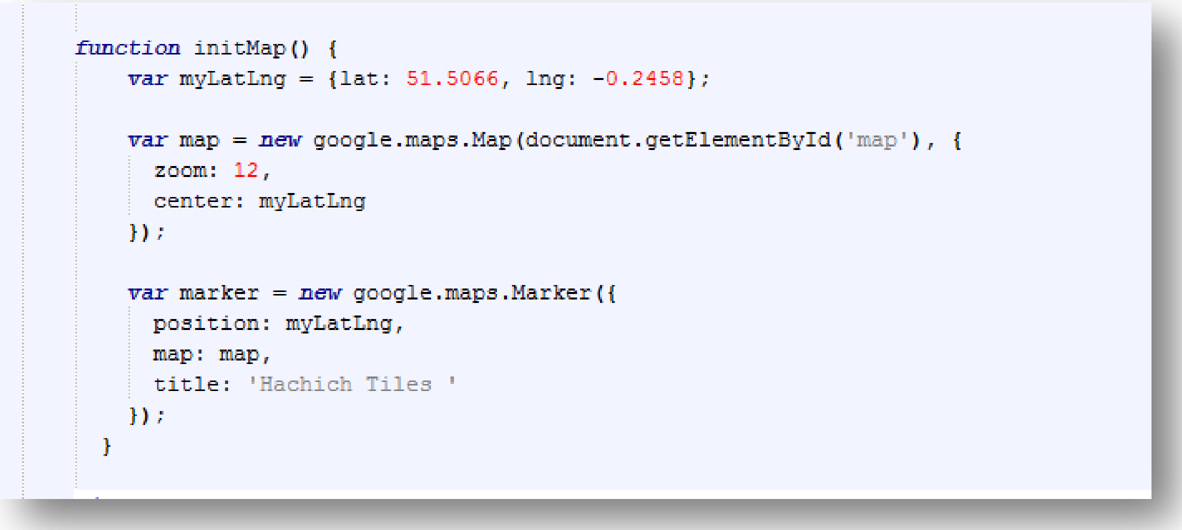


```
if(isset($_GET['delete']))
{
 $query="DELETE FROM customer where customer_id={$_GET['delete']}";
 if(!mysqli_query($con,$query))
 {
     echo mysqli_error($con);
 }
 else
 {
  echo "<script>alert('Customer Deleted succesfully');
  window.location='customers.php';</script>";
 }
```

Figure 69 - PHP code to delete customer accounts

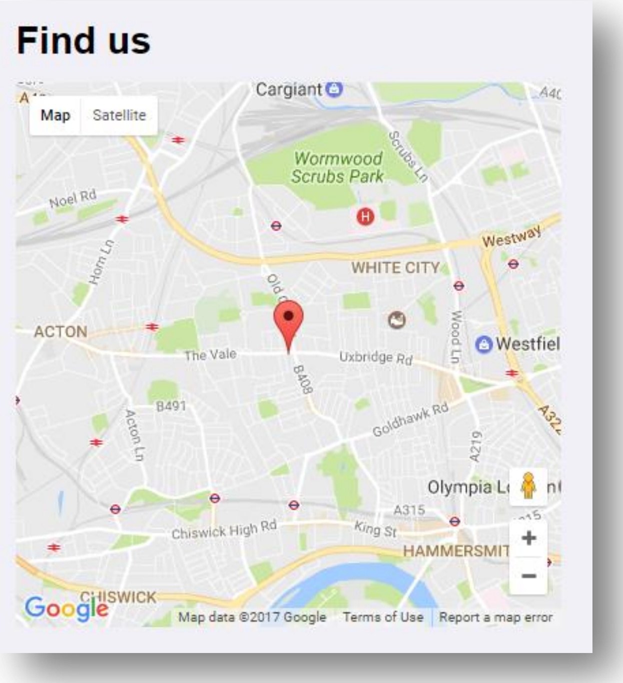
6.3.19. About us

As part of the Programming III module, JavaScript for Google maps was a learnt and implemented into the e-commerce store. On the about us page a JavaScript function was created to create a map and centre a marker to the stores geographical location.



```
function initMap() {
    var myLatLng = {lat: 51.5066, lng: -0.2458};

    var map = new google.maps.Map(document.getElementById('map'), {
      zoom: 12,
      center: myLatLng
    });

    var marker = new google.maps.Marker({
      position: myLatLng,
      map: map,
      title: 'Hachich Tiles '
    });
  }
```

Figure 70 - JavaScript function “initMap”



Find us

Figure 71 - Store location with a pointer using google maps API

7. Testing & Evaluation

In this part of the report, differently testing strategies will be undertaken to verify if the solutions implemented meet the target audience's needs. The first test type would be to conduct a usability test whilst the other test strategy would be white box testing. Both strategies would check for functionality, ease of use, and validation.

7.1. Usability Test

The usability test was conducted based on the CIF standard method which includes the different study methods used and an evaluation of the results used (Hachich, 2017). This usability report will therefore provide a deep evaluation of the design and experience of Hachich Tile's website.

7.1.1. Experimental Design

The overall objective of this usability test is to figure out what parts of the website will fail simple activities provided to the users. There are many tasks that can be undertaken on an e-commerce website, and to build the best atmosphere for the participants, the testing was done with the remote moderated method. Due to the client's budget and the project launch date, a lab-based approach would be expensive, time consuming and difficult to conduct given the time frame.

Tools provided in usability test:

- Laptop
- Web browser (Google Chrome)
- Task Sheets
- Pen and Paper

7.1.2. Participants

Table 6 - Table of participants involved in usability test

Participant ID	Occupation	Gender	Age	Previous online shopping experience?
HT01	Student	Female	21	Yes
HT02	Labourer	Male	38	No
HT03	Delivery Driver	Male	27	Yes
HT04	Student	Male	22	No
HT05	Lawyer	Female	32	Yes

7.1.3. Participant Task Sheet

Task Sheet

Please read the instructions on this sheet carefully.

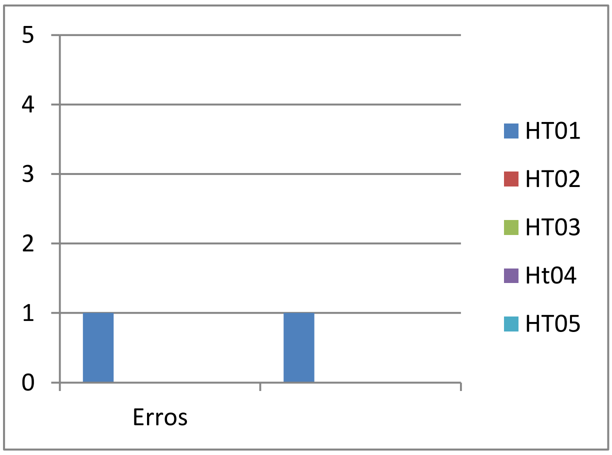
Task Number 1: Search for Mokarao Tile

Search for the Mokarao Tile and add the product to cart. Upon adding the product to the shopping cart, change the quantity of the product to 5. If you do not have an account, create an account, and proceed. After checking out, check your orders to see if the Makarao tile is there.

Note: Please do not contact or make any payments to the seller.

7.1.4. Usability Test Evaluation

Task Number 1: Search for Mokarao Tile	
Time (mean):	00:06:54
Assists (mean):	0.20
Errors (mean):	0.33
Success (%):	100%



Problems

Users found it difficult after login to access the shopping cart. Participant HT01 asked for help when the website redirected them to the “My Account” page. Participant HT01 attempted to add products but was not able to.

Figure 72 - Usability test results

In conclusion to the usability results, the study has shown that users are able to locate a product from the catalogue, add it to cart and checkout without any major technical problems. The navigation bar seems to offer simplicity and completeness for the users when they were attempting to complete different actions. All participants were also able to create an account successfully and login after with the ability of viewing their orders, messages and more. The only issue was that participant HT01 was expecting to be re-directed to the home page or the products page instead of the ‘the account’ page.

7.2. White box and functionality testing

The table below will show all the functions that have been tested; the method of testing that has been undertaken along with their results. White box testing was used as the expected results were known to the tester.

Table 7 - White box testing showing the method of testing and the expected outcomes of the requirements

Functional Requirement	Method of testing	Expected Outcome	Result Pass or Fail?
C1. Create account	Click on login Click on create account Enter valid credentials Submit form	Email the customer a confirmation email, a new row of data to be added to the customers table in the database.	Pass
C2. Login with an account (Valid)	Click on login Enter valid credentials Submit form	A successful login, alert box saying, “successful login” and redirecting the customer to Your Account page.	Pass
C2. Login with an account (Invalid)	Click on login Enter invalid credentials Submit form	Incorrect password or email alert box to show. Denny user access and remain on login page.	Pass
C3. Browse product category	Click on All products View all the products available	All products for sale to be displayed with the ability to view more details and add the product to cart.	Pass
C4. Chose product category	Click on All products Select the category porcelain tiles and floor tiles and view the available products	If products are available in selected category to display them and if not to say “there are currently not products in this category, sorry”.	Pass
C5. Add products to basket	Click on All products View all the products available Add any products to cart	Alert box saying product added to cart. Product should be in the shopping cart with a quantity of one. Product should not be in the	Pass
C5. Add products to basket (Product already exists in cart)	Click on All products View all the products available Add a product to the cart. Add the same product again	Alert message informing the user that the product already exists in the shopping cart.	Pass
C6. View the basket	Click on shopping cart View all the products in the cart	Product name, product quantity, product price and basket total should be visible	Pass
C7. Edit basket (Quantity)	Click on shopping cart Change the quantity of a product from 1 to 5. Click on update cart	Alert box saying, “shopping cart updated”. Product quantity to change to 5, product price to change and the basket total to also	Pass

		change.	
C7. Edit basket (Remove product)	Click on shopping cart Tick the remove box for a product. Click on update cart.	Shopping cart updated alert box. Product to no longer be in visible in the shopping cart.	Pass
C8. View basket total	Click on shopping cart	Total price displayed at the bottom of the cart and under all products.	Pass
C9. View order total	Click on The account panel Click on The Order	All order to be displayed along with the all orders total.	Pass
C10. Delete orders	Click on The account panel Click on The Order Delete a current order	Deleted order should no longer be there	Pass
C11. Request password reset	Click on login Click on forgot password Enter email and submit the form	An email should be sent to the entered email address. A reset password link should be present where the user can enter a new password.	Pass
C12. Receive create account confirmation email	Check email after signing	Account confirmation email should be present in the email address entered to create account.	Pass
C13. View company information	See company information on homepage.	Have company information such as location and contact details visible on the homepage.	Pass
C14. Message seller for stock enquiry	Click on login Click on The Account Click on Send Messages Enter message in text box Click on the send message to submit form	Alert box saying, “Message sent successfully”. Customer should be able to view the sent message by clicking on buyer/seller messages.	Pass
C15. Request potential delivery time of products	Click on login Click on The Account Click on Send Messages Enter message in text box Click on the send message to submit form	Alert box saying, “Message sent successfully”. Customer should be able to view the sent message by clicking on buyer/seller messages.	Pass
C16. Navigate through website easily	Click on the navigation links Return to the homepage	Customer should be able to direct themselves easily	Pass
A1. Add products to catalogue (Admin)	Login as admin with the valid details Select the green add a product button Enter all the product details Click insert product to submit product to catalogue.	Alert message showing product successfully inserted. Product to be added into the products database and should be visible on customers end.	Pass

A1. Remove products from catalogue (Admin)	Login as admin with the valid details Select the product you wish to delete. Click the delete button.	Alert box saying product is deleted. Product should no longer be in the database table of products. The customer should no longer see the product.	Pass
A2. Edit product information (Admin)	Login as admin with the valid details Select the product you wish to edit. Change the product name into "BROWN TEST". Click on the update product button.	Alert box saying product update should appear. Product name should change to "BROWN TEST" in the database. Customer should be able to see a product called "BROWN TEST".	Pass
A3. Manage customer account details (Admin)	Login as admin with the valid details Click on manage customers from the navigation bar. Select edit for the customer whose information you would like to change. Change customer name to "DAVID TEST".	Alert box saying account successfully updated. Customer name in the customer database table should now be DAVID TEST.	Pass
A4. Delete customer account (Admin)	Login as admin with the valid details Click on manage customers from the navigation bar. Click delete from the customer you would like to delete.	An alert box saying account successfully deleted. Customer account should no longer be visible on the manage customers panel. The customer's account should be deleted from the customer's database table.	Pass

8. Critical Review

In this section fo the report a review of theoverall website will be given alongside the problems which have been encountered in both the design and implementation stage.

8.1. Project Success

With the issues of the clients business addressed in chapter 1.0 of the report and the successful implementation of an e-commerce, the project is seen as a success. Customers are now able to view the products on offer, add them to cart and place an order for cash on collection. This helps with the client's initial cash flow problems as more sales can be made via the e-commerce website. This project success has also shown that even if the customer does not place an order, they are able to contact the client through the messaging service offered. The e-commerce website will also aid the client by allowing the range of products offered to be viewed online before any enquiries or orders are made. Customers are also now able to arrange visits to the store through the email and mobile number offered on the e-commerce website. The functionality of allowing customers to send messages has allowed the client and his staff to personally drive conversations and sales. The clients business is also able to attract new customers now through search engine visibility increasing his market share and sales. Meeting the objective of being available 24/7/365 days a year has meant the client is able to reach his audience at all times. With the e-commerce industry expanding, tools such as CrateJoy offer a free platform which will allow the client to analyse his website. This will most definitely allow the client to understand his customers better. The function edit products will allow the client to prioritise the most wanted produducts by placing them on the homepage.

In conclusion, the project's findings indicate that the client's problems will be resolved in both the short term and long term. Brand awareness will increase, along with a better understanding of his customers. New customers will be attracted through the general traffic from search engines. Communication between the customers and the client will be established through the e-commerce website.

8.2. Analysis, Design and Implementation

Based on the literature review conducted and the comparison of the similar websites, MoSCow prioritisation was used to ensure that the most needed functionality will be implemented. Most of the current competitors don’t have the functionality of paying by cash on collection on the ordered products, dealing with the problem of online frauds and payment security. HachichTiles also gives the customer the ability to communicate with the seller for any general stock enquiries or to ask detailed questions instead of calling the store or physically going in. These are all features which the competition do not have. This will allow the client to stand out in comparison to the current competitors.

The chosen methodology, DSDM was selected for reasons explained in chapter 4 of the report. With theclients cashflow problems, choosing an agile methodology ensured that his costs for the project are fixed and there will be no unpredicted costs. DSDM ensures that the project requirements were always flexible and easy to change which was a benefit to theclient more than myself. Features such as a slide show on the homepage and the ability to conduct a live chat with a admin had been implemented but the client had no longer needed them. This meant that more time had to be sacrificed to work on the remaining requirements, making it difficult to stick to the time frame.

Communication between myself and theclient was established on a regular basis but requirements were always changing due to agile methadology chosen.

8.3. Issues Faced

New to the languages PHP, HTML, CSS, JavaScript and any of the web development frameworks on offer today, it was very difficult to learn everything from the start. A lot of time was spent on learning the different ways in which PHP can be implemented and the differenecs the implmenetation can have on the results of the website. The website has the initial idea of keeping all items in a basket correspondent to its IP address, but later found out about sessions and how they can help customers in accessing their basket. This was implemented but meant that all thepresent code which was based on IP addressing had to be changed to work with sessions instead.

Other issues faced was connecting to the root directory through File Zilla. Having bought the domain through 1 and 1 hosting meant that the main domain was connected to a website builder which took some weeks before they were able to remove it and assign it to a subdmain. This meant that despite having a SSL certificate the website's original domain was secure but not the current sub-domain which meant that the websites security was low.

8.4. Future ideas and development

One feature which the client did not want to be implemeted was the ability to allow the customers to make payments for the products available online instead of paying for their products via cash. This feature can be implemented in the future if demand gets high and the client changes his mind.

With the studying of digital business as one of the third year modules offered by Kingston Unniversity, present knowledge on digital marketing was expanded. Through the use of metdata for SEO, PPC (Pay per click), social media and other tools, one can market their business easily. All these methods of marketing will help the client to achieve a high number of page visitors.

Another idea could be to use of websites such as wix or shopify where a monthly fee is charged in return for a simple click and drop simulator which allows the connection of a domain. Admin tasks are taken care of very easily and a simple, stylish design is offered.

8.5. Lessons Learned

Overlooking the results of the project and the amount of time spent on developing the e-commerce site. Having to manage a real-life project with a demanding client where requirements change is not as easy as it may seem, especially when attempting to balance other studies and assignments at the same time.

As mentioned above, at the start of the project there was little to no experience on PHP, HTML, CSS and JavaScript. The different ways in which JavaScript can be used to validate the data entered by the users have been learned. The more time and effort which was put into PHP, the more the understanding grew on the differences in coding from scratch or using a web development framework. As an educational project, coding was done from scratch and everything was implemented by me. Through the literature review conducted, an insight into the different factors affecting e-commerce websites have been understood along with the importance of selling online and its benefits.

8.6. Ethical and Legal Aspects

Having conducted a usabilty test, all users had to sign a pre-test and post-test consent form. The pre-test consent form and post-test consent form will be shown in the appendix at the end of the report.

Appendix B gives the moderator, myself, permission to use the participants results in thereport to collect and publish the data obtained.

------------------------ End of report ------------------------

9. References:

Alexander Hutchinson. (2012). Card payment implementation. *Guide for ASP.NET and PHP websites*, 34-3

Anita Rosen. (2000).The E-commerce Question and Answer Book. *American Management Association, 25*(6), 54-59.

Beal, V. (2016). *What is Database (DB)? Webopedia Definition*. [online] Webopedia.com. Available at: http://www.webopedia.com/TERM/D/database.html [Accessed 28 Dec. 2016].

Business Software Alliance. (2001). E-commerce and Developing Markets: Technology, Trade and Opportunity. *Management and Technology Journal*, *60*(1), 34-59.

Business, Agile. (2012). "Atern Principles". *Agile Business Consortium*. N.p., 2017. Web. 20 Feb. 2017.

Coward, Chris. (2002). Obstacles to Developing an Offshore IT-Enabled Services. *Harvard Business School Press.95* (7), 56-61

Cronin, Mary J. (1998). Banking and Finance on the Internet. *John Wiley & Sons*, 43-47

Chapman, S. (2016). *Are JavaScript and Java the Same Language?*. [online] About.com Tech. Available at: http://javascript.about.com/od/reference/p/javascript.htm [Accessed 28 Dec. 2016].

WikiDot, W. (n.d.). *Dynamic Systems Development Method (DSDM) - AGILE Methods of Software Development*. [online] Dsdmofagilemethodology.wikidot.com. Available at: http://dsdmofagilemethodology.wikidot.com/ [Accessed 14 Nov. 2016].

European Commission (2012).Green Paper -An integrated parcel delivery market for the growth of e-commerce in the EU. The European Commission, *15*,(4) 45-54

E. Turban, D. King & H.M. Chung.(1999). Electronic Commerce: A Managerial Perspective. *Prentice Hall*, 78-84

FTI Consulting. (2011).Intra-Community cross-border parcel delivery. A study for the European Commission. FTI Press, (7), 12-23

G., C. and Jr., C. (2011). *The economic impact of e-commerce – CSCMP's Supply Chain Quarterly*. [online] Supplychainquarterly.com. Available at: http://www.supplychainquarterly.com/columns/scq201102monetarymatters/ [Accessed 12 Dec. 2016].

Lallana, Quimbo &Andrew B. Whinston. (2007).Electronic Commerce: A Manager's Guide. *Addison Wesley Longman, Inc,* 19-20.

Mihai, M. (2008). *E-commerce*. [online] ProQuest. Available at: http://search.proquest.com.ezproxy.kingston.ac.uk/docview/1124361876?accountid=14557&rfr_id=info%3Axri%2Fsid%3Aprimo [Accessed 12 Dec. 2016].

Nielsen, J. (1993). Usability Engineering*. Boston: Academic Press,99-108*

Network &Service-oriented architecture, (5), 77-87

Oxford Economics (2011), The Economic Impact of Express Carriers in Europe. The Oxford Economic Journal, *46* (6), 26-39

P. Timmers.(2000).Electronic Commerce – Strategies and Models for Business-to-Business Trading.*John Wiley & Sons*, 63-91

Phil Bartle. (2007). SMART. Available at: http://cec.vcn.bc.ca/cmp/modules/pd-smar.htm. Last accessed 28/12/2016.

R. Kalakota and A.B. Whinston. (1996). Frontiers of Electronic Commerce. *AddisonWesley*, 19-27

Robert.L.Mitchell. (2012). Computer world, Case study: Integration in the cloud Microsoft developer

Sqa.org.uk. (2007). *Functional and Non-Functional Requirements*. [online] Available at: https://www.sqa.org.uk/e-learning/SDM03CD/page_02.htm [Accessed 12 Jan. 2017].

Statista. (2016). *B2C e-commerce sales revenue UK 2012-2017 | Forecast*. [online] Available at: https://www.statista.com/statistics/252189/b2c-e-commerce-sales-in-the-united-kingdom/ [Accessed 12 Dec. 2016].

Thomas L. Mesenbourg. (2005). Measuring Electronic Business: Definitions, Underlying Concepts, and Measurement Plans. *The Tech Journal, 2*(9), 12-34.

W3schools.com. (n.d.). *PHP 5 Form Handling*. [online] Available at: https://www.w3schools.com/php/php_forms.asp [Accessed 5 Mar. 2017].

W3.org. (2016). *HTML & CSS - W3C*. [online] Available at: https://www.w3.org/standards/webdesign/htmlcss [Accessed 28 Dec. 2016].

W3schools.com. (2016). *PHP 5 Introduction*. [online] Available at: http://www.w3schools.com/php/php_intro.asp [Accessed 28 Dec. 2016].

Yank, K. (2009). *The Big Picture*. [image] Available at: https://www.sitepoint.com/publishing-mysql-data-web/ [Accessed 28 Dec. 2016].

Zorayda Ruth B. Andam. (2000). *An Introduction to Electronic Commerce : SME e-Commerce in Three Philippine Cities. Transaction Publishers*, (2), 40-57.

10. Appendix

Appendix A

Use case name	Add to cart
Use case ID	UC02
Brief description	Use case showing how a customer will add a product to their shopping car
Primary actor	Customer
Frequency of use	100% of the time when willing to purchase a product.
Triggers	Customers view the product being sold.
Preconditions	Customer will have access to the internet. Customer will be on the website.
Main flow	1. User clicks on the all products page 2. User is then provided with all the products in the catalogue. 3. User sees a product of interest. 4. User clicks 'Add to Cart' button. 5. User sees an information window to inform him of the adding to the basket. 6. User then views product in the basket by selecting shopping cart on navigation bar
Alternative flows	AF1. User has already added the product to basket therefore they skip to main flow 6. AF2. Customer session is still active and products are still in cart
Exception flows	EF1. Product is out of stock and customer closes web page.
Post conditions	Success: The selected product by the customer will be in the shopping cart page. The customer will also be able to remove the product from the cart and adjust its quality. Failure: Product will not be found in the cart.

Figure (Add to cart use case text description)

Use case name	Send message
Use case ID	UC03
Brief description	How a user can message the seller for any type of enquiry
Primary actor	Customer
Frequency of use	When the buyer wishes to contact the seller.
Triggers	Clicking on the 'TheAccount' page wanting to send a message.
Preconditions	Customer will have access to the internet. Customer will be on the website. Customer will be logged in.
Main flow	1. The customer will be logged into the website 2. The customer will click 'TheAccount' on the navigation bar 3. The customer will then select the send message feature 4. The customer will write the message and press send 5. Information windows will pop up showing the success of the message sent
Alternative flows	AF1. Customer will select the request refund option on 'TheAccount'
Exception flows	EF1. Send the intended message to the company email address
Post conditions	Success: Information box saying, 'Message sent' and a notification for the administrator. View the message under 'buyer/seller messages' on 'TheAccount'. Failure: Customer does not send the message successfully and hence doesn't receive a reply.

Figure (Send message use case text description)

Use case name	View products
Use case ID	UC04
Brief description	This use case demonstrates the steps a user has to undertake to view a product
Primary actor	Customer
Frequency of use	When the user wishes to view the available products for purchase or just to have a look
Triggers	In need of a product
Preconditions	Customer will have access to the internet. Customer will be on the website.
Main flow	1. Customer will select 'Product' on navigation bar on the homepage. 2. Customer will then be redirected to a page where products and a category sidebar will be shown. 3. Customer then selects the wanted category 4. Customer then located the product 5. After locating the product, the customer then clicks on 'More Details' to be redirected to another page where product information is presented.
Alternative flows	AF1. Customer will search a product to view via the sear bar provided.
Exception flows	EF1. Product wanted is not in stock EF2. Site is down for maintenance.
Post conditions	Success: The customer can view the products available, including its price, details and image. Can then add the product to cart if willing too. Failure: Products are not shown and customers' needs are not meet.

Figure (View products use case text description)

Use case name	Add product (Admin)
Use case ID	UC05
Brief description	Demonstrate how an admin will be able to add a product to the store catalogue for sale.
Primary actor	Admin
Frequency of use	When a new product is in stock
Triggers	The need to maximise sales on a new product.
Preconditions	Admin will have access to the internet Admin will be on the website and logged in Admin will have access to the Add product feature
Main flow	1. Admin will login successfully 2. Admin will select to manage products on the navigation bar 3. Admin will then select Add to product 4. Admin will then insert all the product information 5. Admin then presses add product
Alternative flows	AF1. To insert more than one product at once trough the uploading of an Excel CSV file.
Exception flows	EF1. Edit the current out of stock products to be replaced with the new ones, keep database storage low.
Post conditions	Success: Customer and admin can view the new product via the normal view; all product information is stored in the database. Failure: The product is not uploaded and cannot be viewed by the Customer or administrator.

Figure (Add products use case text description)

Appendix B

Pre-test Consent and Ethics form

I ………………………………………………………… agreed to participate in the usability test which is conducted by Mahmoud Hachich.

I confirm that I am the age of 18 or over and I am not obliged to conduct this test.

I confirm that I have read and understood this information sheet and the invitation to participate.
I understand:

○ The purpose, risks, and benefits of taking part in this session.
○ What the involvement will entail and any questions have been answered to the satisfaction.
○ That the participation is entirely voluntary, and that I can withdraw at any time without prejudice.
○ That all information obtained will be confidential.
○ That research data gathered for the study may be published provided that I cannot be identified.

Contact information has been provided should I wish to seek further information from the investigator at any time for purposes of clarification.

Date: ……………………………..

Signature: …………………………………………..

(Please note this consent form has been used in theuser experience coursework)

Appendix C

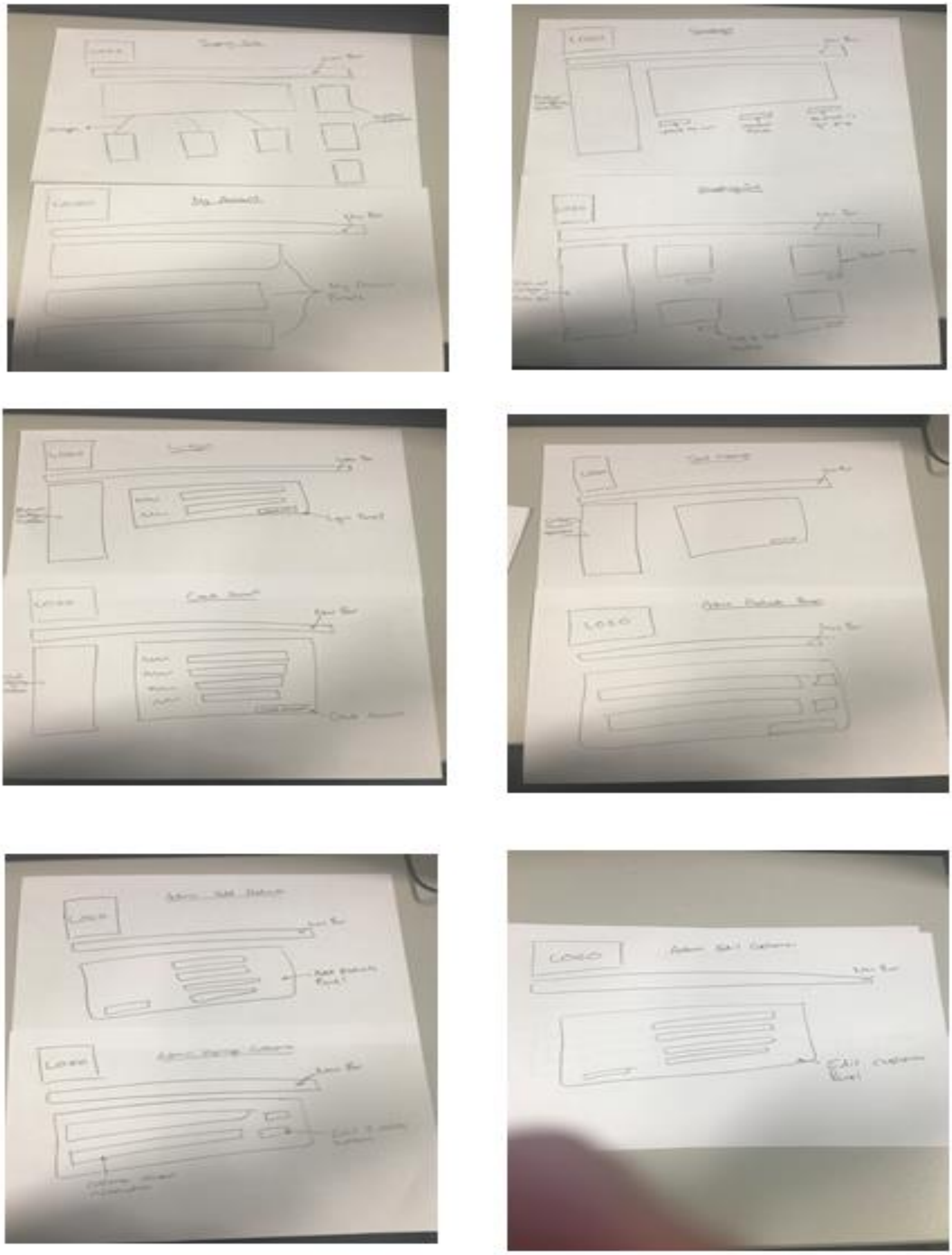


Figure 73 - Low fidelity wireframes

------------------------ End of report ------------------------