

|  |
| --- |
| **Graded Unit Evaluation** |
| Pete Hoben HND Computing - 30258915 |
| This document is my own personal evaluation of the completed project, which was to design a Stock Management System for Value Furniture |
|  |



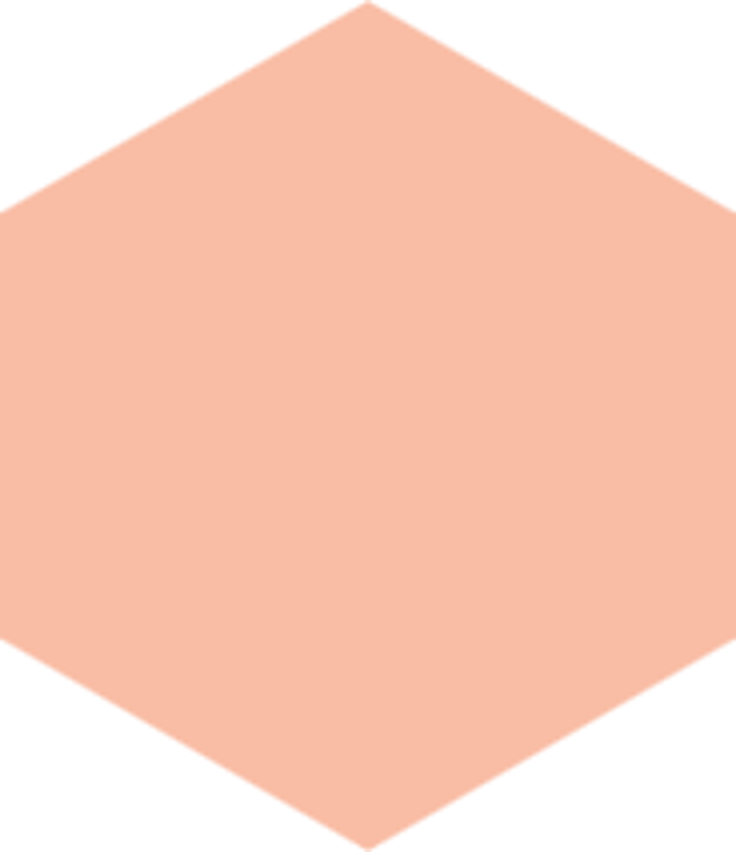
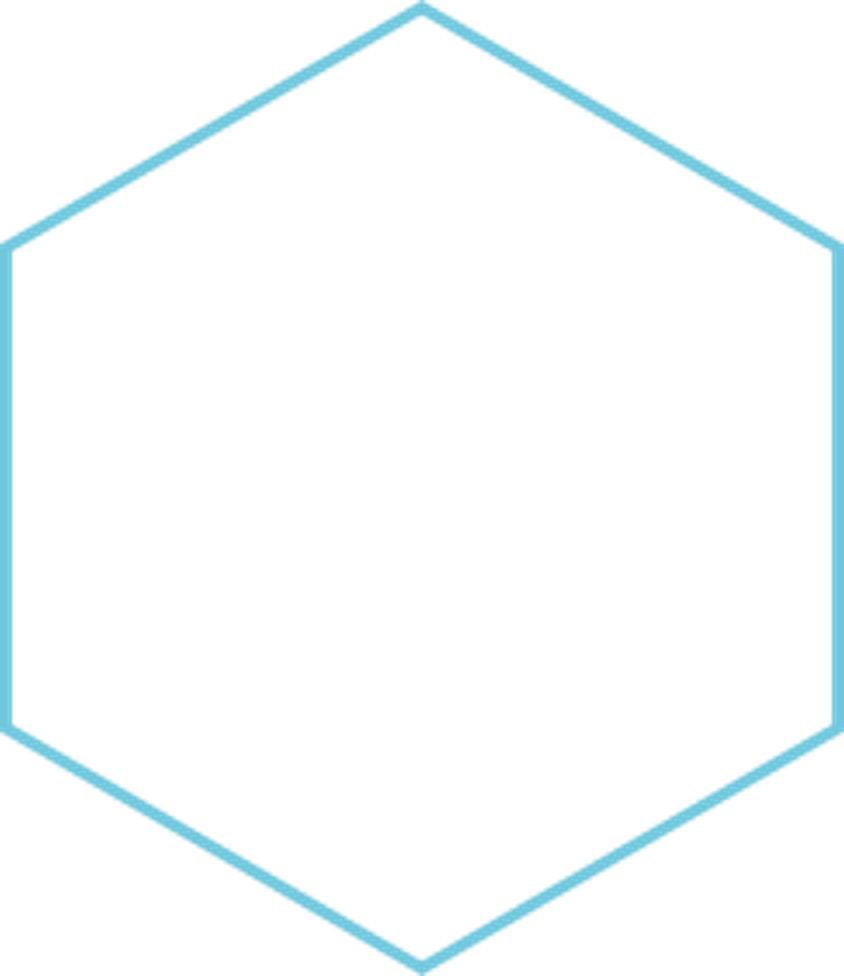


Table of Contents

Purpose of this report ………………………………………………………………………….. 2

Project Overview ………………………………………………………………………………...3

Analysis ………………………………………………………………………………………..…..5

Implementation ………………………………………………………………………………….6

Testing ……………………………………………………………………………………………...7

Strengths and Weaknesses……………………………………………………………………..8

Modifications……………………………………………………………………………………..10

Knowledge and skills…………………………………………………………………………….12

Recommendations………………………………………………………………………………15

Conclusion………………………………………………………………………………………...16

Resources Used…………………………………………………………………………………..17

Purpose of the Report

This report intends to show the process that I have taken in the creation of the Value Furniture Web Application for my Graded Unit. In this report, I will detail the project brief (giving details of what the requirements were as supplied to me by the client ), what I feel the strengths and weaknesses of the project are. I will also go further to detail what could have been done to improve the actual project based on the experience I have gained by doing this, from what my knowledge was when I began.

Project Overview

## Value Furniture

I was a supplied a project brief for a wholesale furniture retailer, called Value Furniture. This company were a trade supplier for furniture, who only sell to retail customers and not the general public. At present they are using a system that consists of only storing customer information on a computer system. All other information is held manually in a paper-based system. This system is not ideal, the system as it stands at the moment leads to many errors when staff forget to update figures. These errors lead to confusion and it ends up costing the company time and money to amend them.

The website being created for them requires multiple roles to be set up within it, each of these roles will be assigned to certain members of staff, this will allow them to perform their duties more efficiently and also lead to less errors occurring.

Each of the roles had to perform certain duties, with the shop manager role being an admin, which had access to every bit of functionality on the system. Also, we had the Sales assistants and Invoice Clerks Sharing responsibilities, so these roles had to be set up in order that they could do the others role also.

The client has requested that an IT system is created for them, that includes a Stock Control system. This system needs to allow users the ability to add products, edit products and delete products if necessary. The system also has to have a facility to create orders for customers and keep track of stock levels.

I was unable to implement the requirement of a live stock system. This requirement was meant to have the orders automatically update the current stock in real time. I failed to achieve this requirement due to issues I had with creating Quantities required on the actual Order. Due to time constraints I decided to move on with the project and come back and fix this later. This was a major mistake as I hadn’t realized the problems this would cause with the database.

The main functionality of the Website is listed below

* A main login Page
* Ability to add staff logins
* Ability to add a new customer to the system
* Add/Decrease the stock levels of a product
* Display Current Stock Levels
* Allow a new order to be added to the system
* Allow all orders placed by a customer to be added to system
* Display all completed invoices
* Have a secure login system for staff consisting of username/password
* Each user group should have access to their own set of actions
* Information about customers’ needs to be stored
* Facility to track an orders progress
* Payment Facilities online
* Allow an order to be cancelled
* System to allow for order returns
* System needs to be large enough to store at least 250 products
* Facility to email an invoice to a customer
* Sales assistants and Invoices clerks should share similar access
* System should generate reports for Stock
* System should generate reports for Invoices
* System should generate reports for Customer Orders
* If sales value is over £25,000 there must be an option for adding a discount

From the list above, all items displayed in red were not completed at the time of writing this evaluation ( ***25/07/2019*** )

Analysis

The Analysis and in fact planning has to be the first step in any Software Project. During this stage of the project certain essential tasks are required to have been completed in order to ensure the success of the project. This can be broken down into three main components

1 – Detailing the Functional and nonfunctional requirements

2- Creating a detailed design of how the system will look

3- Having a schedule for the project set up, commonly done by the creation of a Gantt chart that details every key task and highlights the project milestones.

Once the project brief was received, this was studied a few times to get an understanding of what exactly the client would be looking for as an end product. Once I had a bit more of an idea as to what I was trying to accomplish I then had to look at further fact finding techniques, in order to dig deeper into what sort of a system the client was needing. Sometimes from a project brief many bits of functionality are missed, as the client will not always know what they are needing. The may think it’s the developers job to know what they need rather than them.

This fact finding was done by means of Interview questions and me looking at similar systems.

Once this information was collated, the next step was to produce UML designs for key functions of the system. This section of the planning also required me to create many different diagrams ( use case, activity & class ).

Implementation

This was the stage when I began to code the application. Initially I started by creating the classes that I had detailed in the data model ( ER diagram ), then using Entity Framework in MVC I allowed this to scaffold all the models together and build the SQL Database.

In order to get the basic application set up with the database, I followed a Udemy course on ASP.NET, this course was for an online shop and showed how to get a product added to a shopping cart. This course helped to give me the basis for creating and linking all the further database tables together and provided me with a great place to begin. Unfortunately I biggest mistake within the project came from following this guide too closely. Which I will explain further later on.

Throughout the implantation Phase I learnt so much in regards to c#,SQL and designing web pages. To begin with I had a very limited knowledge of CSS, bootstrap and any sort of HTML

Testing

After the project was completed I conducted a series of tests on the application in order to fix any bugs, and to make sure that all the requirements list had been fulfilled.

I started off by creating a testing plan, this involved setting out tests against the requirements and then ensuring that the application responds as it should to any sort of input. I also had to make sure that all functions were performed to an acceptable timescale.

Documentation was created for each of the main tests that would detail whether it was a pass or a fail, and this was then recorded into a test data document along with a suggested fix that I could go back over and resolve.

This lead to many of the errors being fixed for the final version.

In hindsight I did not follow the testing process correctly for the application, if I could go back and do it again I would have followed a unit testing strategy and have included this in my project plan.

The benefits of unit testing would have meant that the application would have been more efficient and also the missing requirements would probably have been in the application for release now.

I left testing to the end with no real strategy in mind until I got to it.

Strengths and Weaknesses

The completed project has not ended as I expected it to. At the time of writing the evaluation so much is left to be done that should have been done in an orderly logical fashion. Which I have not realized the importance of until now. Parts of this program don’t work due to the coding not following a logical pattern with me jumping from one function to another in order to meet the end deadline.

The main weakness of this application is the inability for a customer or the sales agent to add a quantity to an order. This was skipped by myself during coding, where I believed I can go back and add that it easily. Firstly by not having this on the shopping cart. Invoices would not be able to be created, orders would not have a correct total and also stock levels would not update in real time. At the time I did not think of the repercussions that missing this main piece would be. When I did go back to fix this, I was presented with numerous database errors resulting from the migrations. Errors ranging from missing data, to problems with primary and foreign keys not matching up. Numerous attempts and many hours spent on stack overflow and google still did not give me the solution. Other than accepting that I built an incorrect data model and fixing this was out with my knowledge. I did manage to get the application to create totals on the order line, but not in a conventional way and not by any means a way that would provide much use for the end product.

I feel a good strength of the Web Application is the UI design and layout. Using bootstrap I have created a responsive website that is available on any internet ready device, such as desktop pc, laptop, tablet or smart phone. The design of the site looks modern and attractive. All the buttons and menus are easily identified, buttons are nicely colored, thanks to font-awesome I managed to integrate icons into the design.

The creation of Users on the system is another major weakness of this application. Although the functionality is there for the Admin user to create a customer, or create a staff member and assign a role. I had to have one page broken into two columns for registration. This meant that on the one page you could register either a customer or a staff member. I did adapt the layout of this page to make it look as clean and tidy as I possibly could but from a users point it would have been so easy to register a customer as a staff member and vice versa. I spent a lot of time on this, as I could not find a way to have two register pages set up, one to cater specifically for each user type. I do feel that there is probably a very simple way to do this, but could not find that solution so I tried to make it as easy as possible for the user.

This website is very strong in relation to error checking, both from user input and for the application itself to check the models being passed are correct. In all aspects of user input I have made sure that the inputs have to be of the type that it should be, such as if a name is to be entered, then it has to be of a certain length.

Moving further into the error checking, when data is being added to the database such as a product, if the product has not been given a Product Name, then the user will be advised and the product wont be added until a name is given.

Each time a model is passed, I have used isModelValid, in order to ensure it meets the requirements from the models class ( blueprint ), if the model is not correct then an error is displayed.

Try-catches have also been implemented where appropriate.

Modifications

Right from the start of this project, unfortunately I spent far too much time making modifications to it. The reason behind this was looking at far too much information online looking for assistance in getting me started on the project. With this being the first time I was undertaking a project of this type and lacking the confidence I now have I felt that this research would have been valuable to me, which it was not.

I had built my own data model, and then compared this to other similar ones on line and made amendments thinking that I needed to have these in my application because someone online posted one that has something I did not.

I initially intended to create the application using Microsoft.NET mvc and not asp.net as it has been done with. After viewing and completing many Microsoft tutorials for .NET I felt fairly confident in this, until I was recommended a Udemy course for ASP.NET. This course gave me the foundations to get started building an application.

Whilst writing the code for the Products, I realized that at this stage I should also create a further class for the Product Types, the intention for this was to create a sortable product catalogue on the site, so that users could see specific products rather than have to scroll through a complete list to find the product that they wanted. This lead me to create a further class for the Special tags, so that an item can be given a specific sales comment, such as Highest rated, best seller etc.

Since by this stage I had became confident with adding migrations, updating the database. I also saw how easy it was to create a table with Edit, delete & create views. I then decided I would create a new table and make this dedicated to a marketing list, and I named this “Request a brochure”. I got this idea from Howdens website as I saw that they had this functionality, and Value furniture was missing this from the requirements. This new function allowed me to capture email addresses, let them opt in or out of marketing. And in turn staff now had access to a mailing list, one that was updatable with check boxes, so when a brochure is sent it can be seen on the system. With more time I would have liked to expand this feature so that it created a list of customers needing brochures sent, and then letting the user update them all in the one go rather than individually.

ASP.Net applications start with Bootstrap 3, as I was going into the code and added in Identity to this. Bootstrap 3 was not compatible with it, and all the views were altered to make the pages unreadable. After a bit of research it seems that the version of ASP.NET I was using required Bootstrap 4 so I downloaded and installed that in my application to make sure the application looked good.

Another modification to the project was creating customers as a user type. This meant that admin could create the customer, assign them with a password and then they were able to log into the website, create an order or in fact view their previous order history. This was not as easy as I thought due to the differences with an application user, and a customer user. I would need to set customer pages into an area for Customers, and then make sure that the controllers were set up with the correct authorization, so that a customer could not say access an accounts controller and edit info that they should not have access to.

Knowledge and Skills

From the beginning of this project, until now the end I can say that my knowledge of Software Development and coding has increased so much. When I was given the project brief and from the first reading of it, I was excited but yet terrified. I had never undertook a project like this before and was unsure where to start, or if I could even complete it.

## C# ASP.NET, SQL

Having spent time learning .NET framework, the idea of changing to a completely different framework was daunting. But having completed a few tutorials and then had hands on experience of using it. I now feel fairly confident in this, also I have no second thoughts about writing in .NET also as they are both so similar that the switch from one to the other shouldn’t be difficult. Doing this project had me googling so much, and reading chapters from different books to find out where to begin, but now six weeks later I have so much confidence and belief in what I am doing. I have learnt about using different libraries within the framework. From the project one of the most valuable skills I have learnt is the use of SQL within Visual Studio. When I began the project I had limited knowledge, I knew how to set up a database and add the datasets to it. But by the end of this project I have learnt how to do that and a lot more with sql, I am now confident at adding-migrations, rolling back migrations, creating tables with primary and foreign keys. I can also easily delete them, rename them and also fill a database with test data. The trial and error with the database problems throughout the project made me spend a lot of time researching this until I felt I knew exactly what I was doing. To begin with I would just delete the database and create a new one, until I realized how to avoid this. Saving me a fair amount of time.

## CSS, HTML, BOOTSTRAP, JAVASCRIPT

At the beginning of the project I had little to no experience of creating a website, let alone a data driven website. This meant I had so much to learn , to create a site that looked good. I was confident I could create a plain html page with some text in it.

Now at the end of this, I am confident also in creating custom CSS, I know how to do that, I know how to create a minimized one for efficiency and I also understand HTML.

Initially I was baffled by the <div> </div> tag, Seeing a Div inside another div followed by class=”row” made little sense to me.

Now I understand very well about the div container, and why it is used and more importantly how to use it. I also introduced some JavaScript to the application in the form of drop down boxes for a calendar and times, as part of the order.

## SOFTWARE DEVELOPMENT LIFECYCLE

I learnt the importance of following the lifecycle as it is detailed, I have learnt that by not planning correctly leads to problems further down the line, which result in lost productivity and in turn an ineffective piece of software that would fail Quality Assurance.

Each step of the lifecycle is there for a reason, and that reason becomes clear when you work on a project that is not following it correctly. I didn’t follow it correctly as I was working to a tight deadline and I thought I can skip steps and do them later, which lead to the missing functionality.

With this project it gave me time to work on planning, designing, building, testing and then deployment of the software. These areas were covered in modules throughout the college year. When you are left on your own to work through them it isn’t as easy as it was in a class room environment being lead.

The main think I learnt here was how easy it is for a project to fail due to poor planning. That is something I will take away from this and never do again.

## TESTING

Testing was a valuable part of the project. I had my application complete and then prepared my testing plan. Each of these tests would ensure that my code was done correctly and all functions performing as expected.

I was wrong, tests were failing and the results were noted.

Everything was documented.

For me though, this part of the project was very beneficial, the test would tell me an error occurred and I knew exactly where to look for the error, so using the debugging in Visual Studio I could quickly locate the code that was causing the problem and fix it.

If I could do this again, at the beginning of the project I would prepare a Unit Testing plan and stick to it.

## GITHUB

I had never heard of GitHub or Version Control as its known. My understanding of this was saving the file then making a backup. SO every time I worked on my application I was using a saved version. This became confusing at times, and there was times I would load an old version and work on that. Unproductive.

I discovered GitHub and spend some time with tutorials on this, set up a small application and learnt how to commit files and make branches.

Using GitHub saved me so much time, whenever I hit a problem and the code wouldn’t work, instead of going back to the previously saved version on the hard drive I could just clear the unsaved changes and start again.

Recommendations

For a future release of this application, I would have like to have improved the navigation of the sites links. When a staff member logs into the site, rather than having menu bars at the top it would have been more appropriate to tailor the view to the users. For example when the store assistant logs in, the view would show a count of how many orders were waiting for dispatch, and perhaps only list the actual orders on screen that were needing dispatched. Currently the application shows every order on the system and the user needs to scroll down to see what is outstanding.

I would like to have improved reporting functions on the site also, rather than being simple reports that are just lists of data, the reports could have search filters and create charts for data. This could be used to show information such as what specific items were bought in what month/day. Utilizing reporting such as this could introduce Value Furniture to the advantages of big data, and how big data for trend analysis could help to improve the business.

Another function I would include would be changing the products page, rather than listing every product on the Database, this could instead focus on say 8 products. Reducing the list here could be beneficial as it could be the best sellers, or it could be used to generate products that are in high demand, or on sale. Furthermore, it could be used to show items that Value Furniture currently have a high number of in stock, and therefore would be looking to sell these items as quickly as possible. This could also incorporate an automatic discount on the price on these items to promote them.

Utilizing the mailing list that I set up on the site, to send out a monthly catalogue or even a monthly sales leaflet by email, that again uses the information stored on the database to focus on direct marketing. Perhaps promoting the best selling items or advising on sale items. Having a mailing list is a great way to reach both existing customers and potential new customers.

Conclusion

With the project now ended. This evaluation has given me some time to think over the past six weeks. Thinking about what I done well, what I didn’t do so well and what could be improved.

What I am taking away from this project, is a very valuable lesson on software development. It certainly isn’t all about being good at coding, so much more needs to be considered, such as time management and also a great deal of thinking outside the box.

This project gave me a view on everything, including testing. This was no easy challenge, but it was thoroughly enjoyable in every aspect.

What I have spent the past two years in college doing has all came together and I realize what I need to know in order to get a career in software.

I made many mistakes on this project, right from the start my interpretation was an online shop is being created. When in fact it wasn’t, the online shop was really an extra part of a Stock Management system. I should have focused on that part first and then built the shop around that, rather than the other way I about.

As much as I am not 100% happy with the end result of the application, the amount I did learn and the confidence that I now have was worth every minute of the last two years.

Resources Used

## Useful resources

Throughout the course of the project, I found myself doing a lot of research. Information was gathered and used from a variety of sources; these resources varied from books to YouTube video content. Below is a list of everything that was used by me to aid the completion of the project.

**Sandcastle – Is a Help File builder for Visual studio that creates documentation**

<https://blogs.msdn.microsoft.com/msgulfcommunity/2014/04/22/creating-documentation-in-c-using-visual-studio-and-sandcastle/>

**Udemy – Provides online training for a massive amount of topics**

Master ASP.NET MVC Core 2.1 using Entity Framework Core and deploy the final application on Azure and IIS

**Microsoft Tutorials – Microsoft provides tutorials for getting started with Visual Studio**

# Tutorial: Get Started with Entity Framework 6 Code First using MVC 5

# Tutorial: Get started with EF Core in an ASP.NET MVC web app

**City of Glasgow College Mycity – Colleges online learning facility & course notes and tutorials.**

C# Programming 3E by Barbara Doyle

**YouTube – Video Sharing platform**

Learn ASP NET MVC 5 Complete Tutorial 2019

**Amazon Prime – Video from amazon for a variety of topics**

The complete Course for Web Designer

The Complete Web Developer Course 2.0

**Reference Books**

Software Project Survival Guide by Steve McConnell

Applying UML and Patterns by Craig Larman

PRO ASP.NET MVC 5 Platform by Adam Freeman

Teach Yourself XML in 24 Hours by Michael Morrison

**Online Resources**

Font Awesome

GetBootStrap

Html – [www.w3schools.com](http://www.w3schools.com)

StackOverFlow