University of Plymouth

School of Engineering,

Computing, and Mathematics

PRCO304

Final Stage Computing Project

2019/2020

Corner Shop Specialist

William Butler

10575638

BSc (Hons) Computer Science

**Acknowledgments**

I would like to acknowledge the support and guidance of my supervisor, Steven Furnell, throughout this project making sure I was always making progress and pushing me to achieve as best I can.

I would also like to thank my family, friends and colleagues on this course for all of their support throughout my degree.

**Abstract**

This report outlines a software development project to build a system that can be used by staff and customers of corner shops to assist both staff to more effectively carry out their jobs. The system allows customers to check the stock and products of their nearby shop and has a click and collect service for item reservation. For staff, the system allows managers to handle stock, deliveries and their staff.

The report begins by outlining the background, deliverables and objectives of this project, delving into the industry area, what is currently used and the drawbacks of the current solutions. Moving on from that the report will then cover the approach taken during the project and the social legal and ethical concerns related to this project. I will next then describe the architecture and design process used to set out the framework for the project. The report describes how the project was managed to ensure its success.

The next section of the report will cover the development phases of the project where a full stack system to solve the proposed problem with retail systems and deliver a system that could be used in industry to make working and using shops better for both staff and customers.

To conclude, this report will cover a post-mortem of the project to evaluate what went well and what could be improved on for future projects of this manner.

Contents

[1 Introduction 5](#_Toc41046673)

[2 Background, Objectives and Deliverables 6](#_Toc41046674)

[2.1 Motivation 6](#_Toc41046675)

[2.2 Project Objectives 6](#_Toc41046676)

[2.3 System Requirements 7](#_Toc41046677)

[2.4 Deliverables 7](#_Toc41046678)

[3 Legal, Social, Ethical and Professional issues 8](#_Toc41046679)

[3.1 Legal Considerations 8](#_Toc41046680)

[3.1.1 Licencing 8](#_Toc41046681)

[3.1.2 Data Handling 8](#_Toc41046682)

[3.1.3 GDPR 8](#_Toc41046683)

[3.2 Ethical Considerations 8](#_Toc41046684)

[3.3 Social Considerations 8](#_Toc41046685)

[4 Project Management 9](#_Toc41046686)

[4.1 Agile Project Management 9](#_Toc41046687)

[4.2 GitHub 9](#_Toc41046688)

[4.3 Trello 9](#_Toc41046689)

[4.4 Vertical Slice Development 9](#_Toc41046690)

[5 Stages 10](#_Toc41046691)

[5.1 Stage 0 – Architecture research and requirements engineering 10](#_Toc41046692)

[5.1.1 Objectives 10](#_Toc41046693)

[5.1.2 Outcomes of this stage 10](#_Toc41046694)

[5.1.3 Stage Review 14](#_Toc41046695)

[5.2 Stage 1 – Log in and sign up 14](#_Toc41046696)

[5.2.1 Aims 14](#_Toc41046697)

[5.2.2 Objectives 15](#_Toc41046698)

[5.2.3 Outcomes of this stage 15](#_Toc41046699)

[5.2.4 Stage Review 17](#_Toc41046700)

[5.3 Stage 2 – Stock and Deliveries 18](#_Toc41046701)

[5.3.1 Aims 18](#_Toc41046702)

[5.3.2 Objectives 18](#_Toc41046703)

[5.3.3 Outcomes of this stage 18](#_Toc41046704)

[5.3.4 Stage Review 19](#_Toc41046705)

[5.4 Stage 3 – Catchup from Technical Issues 20](#_Toc41046706)

[5.4.1 Aims 20](#_Toc41046707)

[5.4.2 Objectives 20](#_Toc41046708)

[5.4.3 Outcomes of this stage 20](#_Toc41046709)

[5.4.4 Stage Review 22](#_Toc41046710)

[5.5 Stage 4 – Staff Creation and Click and Collect 22](#_Toc41046711)

[5.5.1 Aims 22](#_Toc41046712)

[5.5.2 Objectives 22](#_Toc41046713)

[5.5.3 Outcomes of this stage 22](#_Toc41046714)

[5.5.4 Stage Review 22](#_Toc41046715)

[5.6 Stage 5 - Holiday, payroll, shifts and further deliveries 23](#_Toc41046716)

[5.6.1 Objectives 23](#_Toc41046717)

[5.6.2 Outcomes of this stage 23](#_Toc41046718)

[5.6.3 Stage Review 23](#_Toc41046719)

[6 End-project report 24](#_Toc41046720)

[6.1 Review of project objectives 24](#_Toc41046721)

[6.2 Changes during development 24](#_Toc41046722)

[6.2.1 Technology Stack Changes 24](#_Toc41046723)

[6.2.2 Functionality changes 24](#_Toc41046724)

[7 Project Post-mortem 25](#_Toc41046725)

[7.1 What went well 25](#_Toc41046726)

[7.2 What could be improved 25](#_Toc41046727)

[7.3 Lessons learned 26](#_Toc41046728)

[8 Conclusions 27](#_Toc41046729)

[9 Reference List 28](#_Toc41046730)

[10 Appendices 29](#_Toc41046731)

[10.1 User Guide 29](#_Toc41046732)

[10.2 Project Management 29](#_Toc41046733)

[10.2.1 Trello Boards 29](#_Toc41046734)

[10.2.2 Sprint plan 29](#_Toc41046735)

Word Count:

Code Submission Link: <https://github.com/wbutler99/PRCO304>

# Introduction

The idea for this project came from the developer’s experience working in retail for a year and using the systems provided to him to carry out his job. This project has no specific client in mind but has the potential to be commercialised as an off the shelf system for businesses.

In a retail environment, there is historically the case that the systems they use to store corporate information such as stock information, staff details and other corporate details on a shop to shop level are being handled by decades old software and hardware that is not kept up to date due to cost and other corporate concerns. This system has been built to modernise the systems used by shop managers and staff to help enhance the potential of these systems and add features to make working in retail more convenient and easier on both the staff and customers.

This software has been developed to be both a website and desktop application with the website providing an interface for staff and customers and the desktop application primarily for managers to handle corporate information and manage their individual shop.

# Background, Objectives and Deliverables

This section of the report will cover an overview of why the project was chosen and the desired outcomes of the project by its completion.

## Motivation

The main reason this project was conceived was to make working in retail more straightforward for the staff by alleviating some of the issues they currently encounter.

One major issue is the use of very outdated software for back office computer systems that is often slow and normally running on outdated operating systems that are no longer supported by their providers with updates thus making them very unsecure to recent security exploits which could be crippling for a company’s corporate information that is stored on these computer systems.

Another issue with these currently used systems they do not accommodate areas such as staff shifts and holidays as these are still done via paper. This provides the limitation of staff members having to physically come into the shop to see when they are working or to request and check their holiday. Having this on an online platform would allow staff to check shifts online for instance if they have been off work for holiday.

Another problem that shop staff face when working on the shop floor is questions from customers regarding if they stock a certain item or when items will be back in stock. This is something that can be moved to an online platform to free up staff members to more effectively carry out their duties and responsibilities.

The final issue that has been identified is pertaining to the usability of the systems currently employed by companies as most of them are ageing systems they are using very outdated usability techniques that provide a poor user experience for their end users with some systems seen using an almost command line system for certain sub-systems of the software.

The solution to be developed to address the issues outlined above to provide a better user experience for staff and allow customers to access information about their local corner shop online so that they do not need to travel to the shop to get the desired information. It will also provide a more secure system for corporate information on an up to date operating system to prevent data leaks.

## Project Objectives

The objectives outlined below consist of the desired outcome of this project and are:

1. For the developer to gain experience and understanding in full stack development and specifically vertical slice development.
2. Build a system that solves as many of the problems outlined to be resolved by the system as best as possible for the end users.
3. To provide a good user guide for the end users to be able to effectively use the system produced.
4. For the developer to gain insight into how commercial systems are produced and the requirements surrounding them in a business environment.

## System Requirements

A set of system requirements has been produced to quantify if the system is considered complete. These requirements were derived from requirements engineering performed in section 5.1 of this document. These will be broken down in more detail in the form of user stories in section 5.1 of this document.

1. Allow staff members to access information about their work remotely.
2. Allow customers to access information about shops and their products remotely.
3. Allow managers to manage stock and deliveries to their shop.
4. Have the ability for an administrator to manage aspects of all shops covered by the system.
5. Have functionality for staff members to perform basic work tasks from home that are not required for their presence in store.
6. Allow customers to check stock and reserve products for collection from their desired shop.
7. Have functionality for the management of newspaper delivery services if shops require them.

## Deliverables

Outlined below is a set of deliverables at a high level that the system should have upon completion:

1. A Desktop application to be used by both managers as an in-store back office computer system and by administrators in a head office to manage corporate information for shops.
2. A website for customers and staff to use to access information remotely about shops and staff information.
3. A backend Rest-API to securely provide data to the front-end systems
4. A database to store corporate and customer information.
5. User Guide
6. Project Report

# Legal, Social, Ethical and Professional issues

Legal, social and ethical issues is something that needs to be considered to ensure that the project is conducted in a manner that

## Legal Considerations

This section will cover any legal concerns that have to be considered for this project to make sure it is in accordance with regulations and guidelines in industry.

### Licencing

**Mongo DB Compass**

As a free plan of Mongo DB compass is used, the developer must adhere to the software terms of use. (Mongo DB, 2020)

**JavaScript Node packages**

Most of the JavaScript libraries used fall under the MIT licencing under the open source initiative making them freely available and are thus suitable for the intended use on this project.(The Open Source Initiative, 2020)

**Visual Studio**

Microsoft Visual studio is licenced through the Microsoft image program by the university. This allows for Visual Studio to be used for this project. However, this does mean that this system cannot be published for profit under this current licence and a full Visual studio licence would need to be acquired for this to be done. (Microsoft Imagine, 2020)

### Data Handling

### GDPR

## Ethical Considerations

The only human participants used to get feedback on this project were colleagues on the developer’s course. This is covered by the generic ethical approval submitted on behalf of the department for this module and covers all the ethical issues that would have been faced during the project.

## Social Considerations

This system was developed purely for the completion of the PRCO403 module to fulfil the requirements outlined. This means that the system is not intended to be deployed

# Project Management

## Agile Project Management

For this project, an agile approach has been used to manage the project and ensure its success. This is outlined by the PRCO304 module specification as the recommended project management style to use for the project.

Agile Project management as detailed in the agile manifesto has the key principle of continuous delivery of software to end users/clients at regular intervals. (Agile Manifesto, 2001) This was established by developing according to iterations or sprints with the design allowing for something to be ready to be shown to the end user at the end of each iteration.

## GitHub

GitHub has been used to manage the code base of the project

## Trello

## Vertical Slice Development

Vertical slice development was employed as part of an agile approach to maintain quality of the code produced and to verify completeness of functionality.

# Stages

In this section, the stages of development throughout the project will be documented to show what was produced at a given time and how successful each stage was.

## Stage 0 – Architecture research and requirements engineering

### Objectives

The objectives of this stage of the project are as follows:

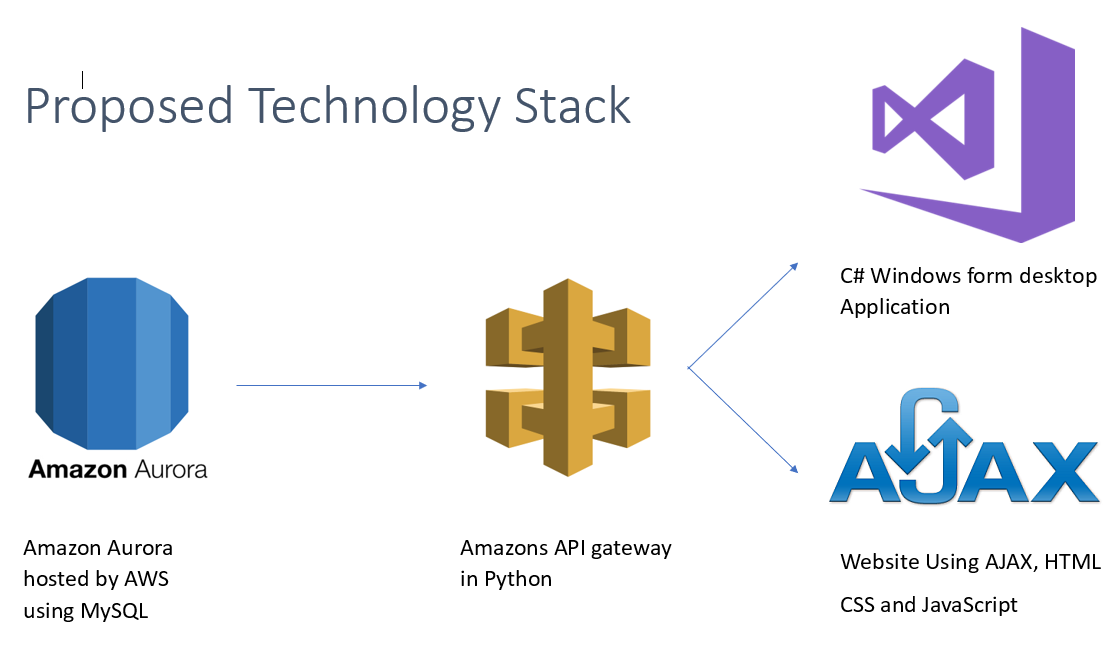
1. Research into potential technologies to use and analyse their effectiveness for producing a solution.
2. Identify relevant requirements.
3. Draw up user stories to be completed throughout the project.
4. Prioritise those user stories into what is most critical to the system and what is optional.
5. Draft a sprint plan to plan out how the development cycle will go over the coming stages.

### Outcomes of this stage

**Technology Stack**

Initially, it was decided to research into a cloud hosting solution for the project not only to give the developer more experience with this kind of framework, but also since it would allow for more seamless deployment of the software across multiple devices. To this end Amazon Web Services (AWS) was identified as a potential back-end technology stack as it has a free plan that would be suitable for the needs of the project in terms of performance and compatibility with the developer’s skills.

In terms of the front-end systems to consume the database and API, I decided to build a website using HTML, CSS and JavaScript due to the developer’s experience using these languages from previous study. To handle the requests to the server jQuery and specifically AJAX was identified as a suitable solution. To create the desktop application, C# windows forms were chosen to be used due to the developer’s previous knowledge of the software and the “drag and drop” nature of the practice allowing for effective user interface design.

Figure 1: Proposed technology stack to be researched and used at the start of stage 0.

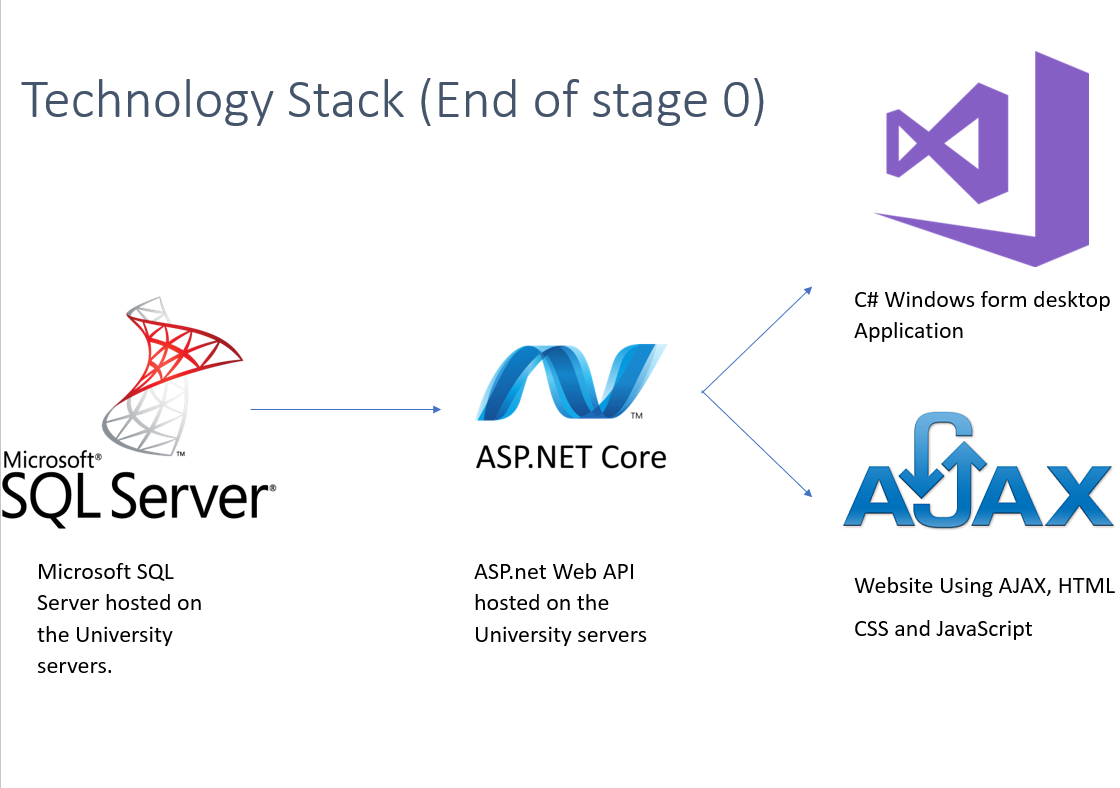
After further research into how AWS works and integrating it into software present on the developer’s devices. It was established that there were technical issues with connecting MySQL Workbench to an AWS MySQL database. This had the knock-on effect of using Amazon’s API Gateway much more complicated to use as it works most effectively with Amazon hosted services. This meant that the back-end technology stack was changed during this stage to move to technologies that the developer had used in the past and was more confident would work. The technologies for the Database and API were replaced with a Microsoft SQL Server and ASP.net Web API respectively.

Figure 2: Technology Stack post stage 0 research.

**Requirements**

During this stage requirements were elicited from both the developers experiences and desires for the system in terms of improving user experience for shop staff and also through consultation with family who have worked in management rolls of shops in the past to see what they would require out of the system.

From this a set of functionalities in the form of user stories was devised and prioritised into 3 categories. These categories include: Core, Stretch and Optional. Core functionality is functionality that is critical to the system functioning as intended and must be completed by the time the project is completed. Stretch functionality is functionality that ideally would be completed by the end of the project as it is important to be included. Optional functionality is user stories that are not as important to be included in the context of the overall system but would be welcome inclusions to extend the overall functionality of the system.

The following is the user stories identified sorted into these three categories:

Core:

* As a customer, I want to be able to set up an account so that I can use the website
* As a customer, I want to be able to log into the website so that I can access the features on the systems
* As a manager, I want to be able to log into the back-office system so that I can manage my shop
* As a Staff member, I want to be able to log onto the systems to access my features
* As a manager, I want to be able to set up accounts for my staff so that they can access the system
* As a customer, I want to be able to update my account details so that they are accurate
* As a customer, I want to be able to search for products so that I can see if my shop sells them
* As a manager, I want to be able to see what Items are on my next delivery so that I know what stock is due into the store
* As a manager, I want to be able to see how much stock of each item we sell we have so that I can know what to order for delivery
* As a staff member, I want to be able to change my payroll details so that I am paid into the right bank account
* As a staff member, I want to be able to request holiday so that I can have a break from work
* As a manager, I want to be able to approve holidays so that my staff can have time off when they have requested
* As an admin, I want to be able to add deliveries to the system so shops know when their deliveries will arrive
* As a manager, I want to be able to set up rota's so that my staff know when they are working
* As a staff member, I want to be able to look at the shop rota so that I can see when I have to work.

Stretch:

* As a customer, I want to see if the product I want is in stock so that I know if I can go and get it.
* As a customer, I want to be able to reserve the product that I want so that I can go and pick it up
* As a manager, I want to be able to view click and collect orders so that the items can be put aside for customers to collect
* As a customer, I want to see if the product I want is in stock so that I know if I can go and get it.
* As a manager, I want to be able to add staff members to the system so that they can access the system.
* As an admin, I want to be able to add managers to the system so that they can manage their shops

Optional:

* As a customer, I want to be able to set up a news delivery so that I can have papers delivered to me
* As a customer, I want to be able to cancel my newspaper deliveries If I do not want them delivered anymore
* As a customer, I want to be able to edit my newspaper deliveries so that I can get the paper I want delivered
* As a manager, I want to be able to view newspaper rounds so that I can organise papers for paper staff to deliver.
* As a customer, I want to be able to pay for my newspapers online so that I don't have to go into the shop to pay regularly

**Sprint Plan**

After the above requirements were elicited, they were organised into 2-week sprints to be completed throughout the project. The functionality was organised based on both its priority to the system and its connections to other user stories so that the system could be build up more efficiently layer by layer. The full sprint plan can be seen in appendix 11.2.

### Stage Review

Overall this stage was a success.

This stage identified key issues with the technology stack that if they had not been picked up before development could have cause significant delays to the overall project timeline due to the need to decide on and change technology stacks once issues arose. This also had the added benefit of moving development to more familiar architecture that would have then allow for a more efficient development process.

The requirements elicited were detailed and gave a clear vision for the specific functionality points that would need to be developed for this system to be fit for purpose in a real retail business environment. The prioritising of the functionality also allowed for the sprint plan to be more focussed on the core functionality and reduced the amount of gold plating that could creep into the project over time.

## Stage 1 – Log in and sign up

### Aims

The main aim of this stage was to get the basic framework of the system in place and to start work on some of the more basic pieces of functionality that would be required for the rest of the system to be built on to of. This included the ability for end users to sign up for accounts and log into the system which is required for any information to be provided to end users if the information is sensitive.

### Objectives

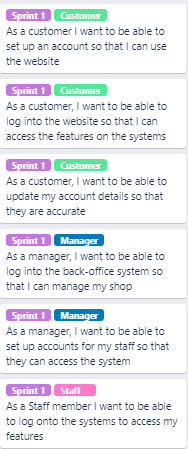


Figure 3: User stories to be completed in sprint 1

Other non-functional objectives to be completed in this stage:

1. Set Microsoft SQL Server database.
2. Set up ASP.net web API.
3. Set up the GitHub repository with a file structure.
4. Implement security measures for logging in and accessing information.
5. Implement encryption for passwords at rest.

### Outcomes of this stage

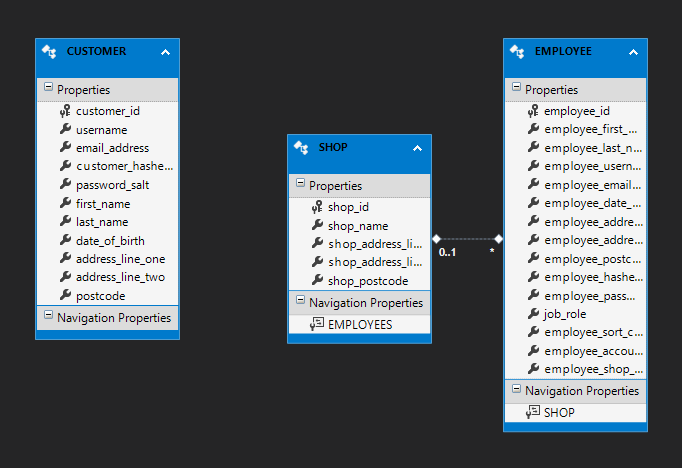
The back end of the system was designed around vertical slice development. To this end only functionality for 3 tables for the database were developed. Those tables being Customer, Staff and shops. 

Figure 4: Database entity relationship diagram generated by ASP.net.

The endpoints for these tables were then developed in the form of ASP.net controllers to expose the data.

In terms of implementing security features, there were 2 services/libraries that were used to ensure that data was only access by people who had authority to access it and that data was secure in transit and at rest.

Oauth was used as a token-based authenticator to ensure that endpoints could only be access by users with a token already granted to them by logging into the system. The .NET frameworks security libraries were used for the hashing and salting of passwords received by the system so that they are secure at rest. Also, as part of the implementation password or their salts are not sent to the client side in any way to ensure that the security is maintained.

The user interface for logging in and signing up on both the website and the desktop application were also produced to allow for the end users to interact with the back-end functionality.

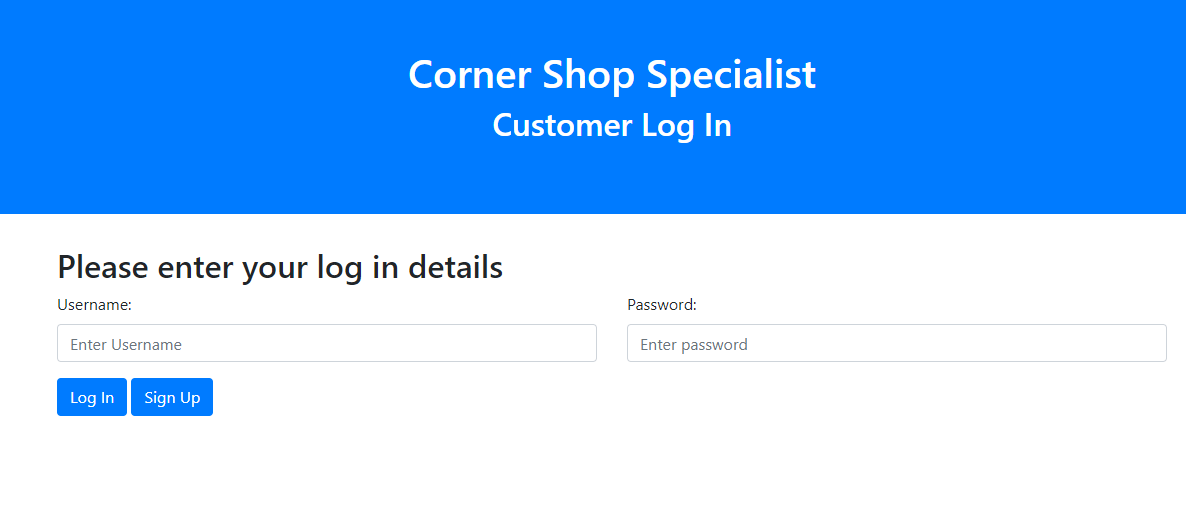
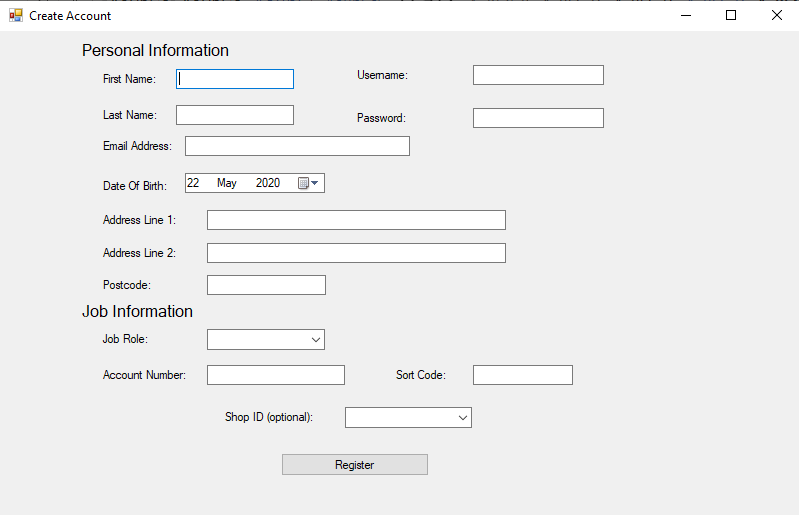


Figure 5: User interface for the website log in page

Figure 6: User interface for adding staff to the system

### Stage Review

Overall this stage was good in terms of implementing the front-end systems as the user interface for all functionality due to be completed in this sprint was completed to a good standard and was ready for the connection to the database to be fully functional.

However, it was at this point that an issue arose around the connection between the database and the API causing an authentication error to be thrown upon completing a request on the client side for either requests for data or for posting data to the server. This issue was to be investigated further in stage 2 in the hopes of finding a solution.

## Stage 2 – Stock and Deliveries

### Aims

The aim of this stage was to implement the groundwork for the stock management section of the system. This included implementing functionality for products to be view by customers and for stock values to be stored and retrieved in the desktop application for managers. Another aim was to fix any issues that arose in stage one that were hampering development progress.

### Objectives

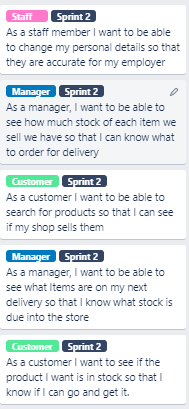


Figure 7: User stories to be completed in sprint 2

In addition to the above user stories there were also other tasks that needed to be completed:

1. Identify and find a solution to the connection issues between the API and Database
2. Finish the back-end functionality for sprint one that could not be completed due to technical issues.

### Outcomes of this stage

Before any development occurred on the sprint 2 functionality that needed to be implemented, the issue regarding the authentication error encountered in stage 1 was investigated. This problem had a large amount of time dedicated to it during this stage with contact going between lecturers to try and identify the problem and multiple attempts by the developer to find solutions to the problems.

In the end, as this problem was taking too much time to fix and there was no clear solution in sight. To ensure that the project did not fall further behind than it already had, the decision was made to change technology stack for the back-end functionality to MongoDB and NodeJS. These technologies were decided upon due to the developers very recent experience working with them during previous studies.

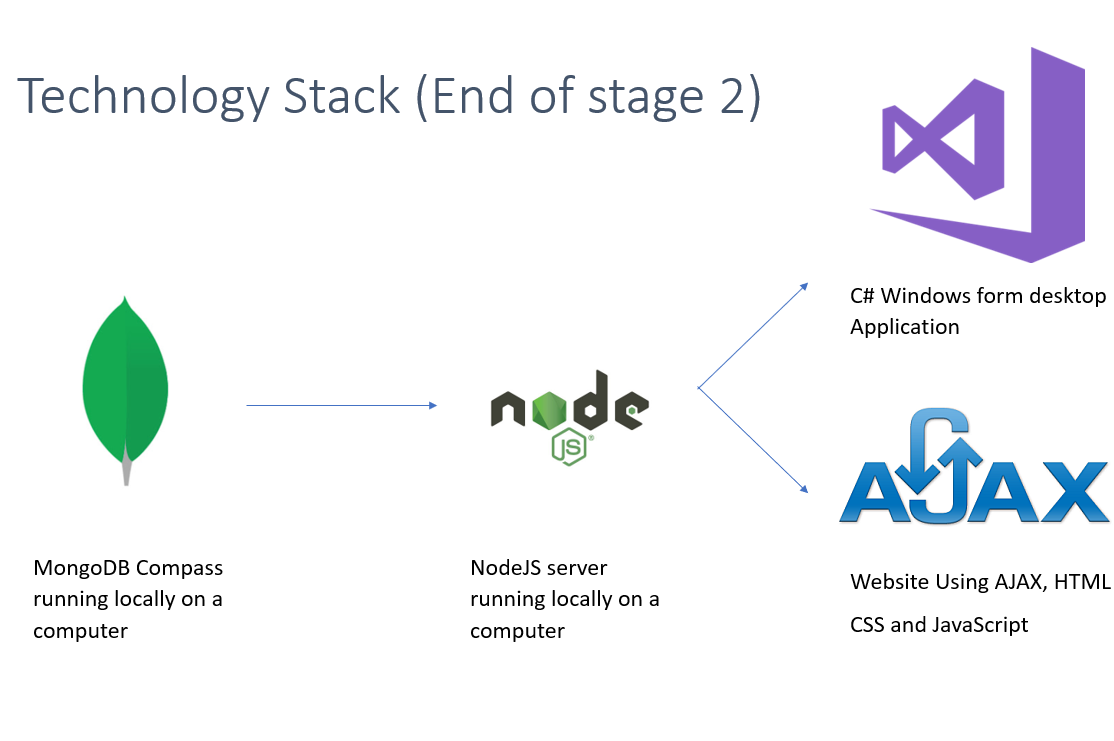


Figure 8: Technology stack after review in stage 2

At this stage it was also evaluated as to whether to reduce the amount of functionality that should be completed as a large amount of development time had been lost due to technical issues at this point. To this end, it was decided that the optional objectives identified at the start of the project, would be removed from the planned development cycle and would only be completed if significant progress was made and time was available at the end of the project.

In terms of new features developed these were very limited due to the technology changes requiring significant time and large amounts of refactoring to be completed. This stage saw no user interfaces created for the user stories set to be addressed in this stage and the back-end functionality that had already been created was all scrapped in favour of changing to the new technology stack.

### Stage Review

Overall this stage was not a success. This was due to the technical issues causing significant delays to the project and meaning that none of the features slated for development in this stage got developed and had a knock on effect of pushing back subsequent sprints in order to allow time for the incomplete features to be developed and for the project to stay on track for completion.

## Stage 3 – Catchup from Technical Issues

### Aims

The main purpose of this stage is to bring the project back on track after the technology stack changes made at the end of stage 2. This will entail completing all the sprint 1 and 2 functionality fully so that the project can move forward onto the more complicated functionality due to be completed next. This is planned to be longer than the rest of the stages due to the scale of the work required for this stage to be deemed complete.

### Objectives

During this stage the following needs to be completed:

1. Set up structure for the NodeJS server and Mongo database schemas
2. Complete back end functionality for sprint 1
3. Refactor current sprint 1 front end systems to work with new technology stack
4. Implement user interfaces and business logic for sprint 2 functionality
5. Research and implement security features in NodeJS

### Outcomes of this stage

During this stage MongoDB was set up locally on the developer’s computer to mitigate the risk of Mongo’s hosting service being overloaded as some hosting services were seeing overloads due to the Covid-19 pandemic and people working remotely.

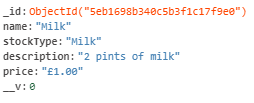


Figure 9: An example of a MongoDB record

A node server was then created in order to expose the schemas created in MongoDB. Once this was created, an alternative security library needed to be located as the security measures previously used were specific to ASP.net. Bcrypt was identified as the most viable solution as a hashing and salting algorithm in Node as it stored all the required information in one variable making authentication possible in one built in function call. Unfortunately, no alternative to the token-based authentication planned to be used could be found in the time frame so this feature was removed from the project.

The user interfaces for the sprint 2 functionality were then produced to allow the end users to access the products and stock information and the changing of personal information. There was a small delay with the desktop application when working out an effective method to view an unspecified amount of records to the end user once they are requested from the database. This was solved by using a “Data Grid View” that allows for a table to be constructed based off a C# class that the data could be passed into via JSON object mapping provided by Newtonsoft.json.

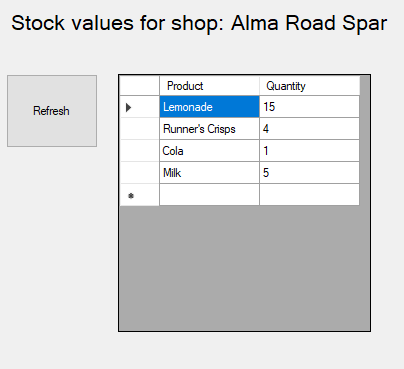


Figure 10: The desktop page for stock using a Data Grid View to map the values to a table

When developing similar features for the website, a technique of table iteration was used on a HTML table with the headers already laid out on the page. Bootstrap styling was used in order to make the tables styling consistent with the rest of the interface design such as the navbar thus making the interface easier to follow for the end user.

The search bar was implemented by searching the database for any matching product names in the products collection. However, this did come with the drawback that it could only find exact matches for records so this could prove a problem for end users.

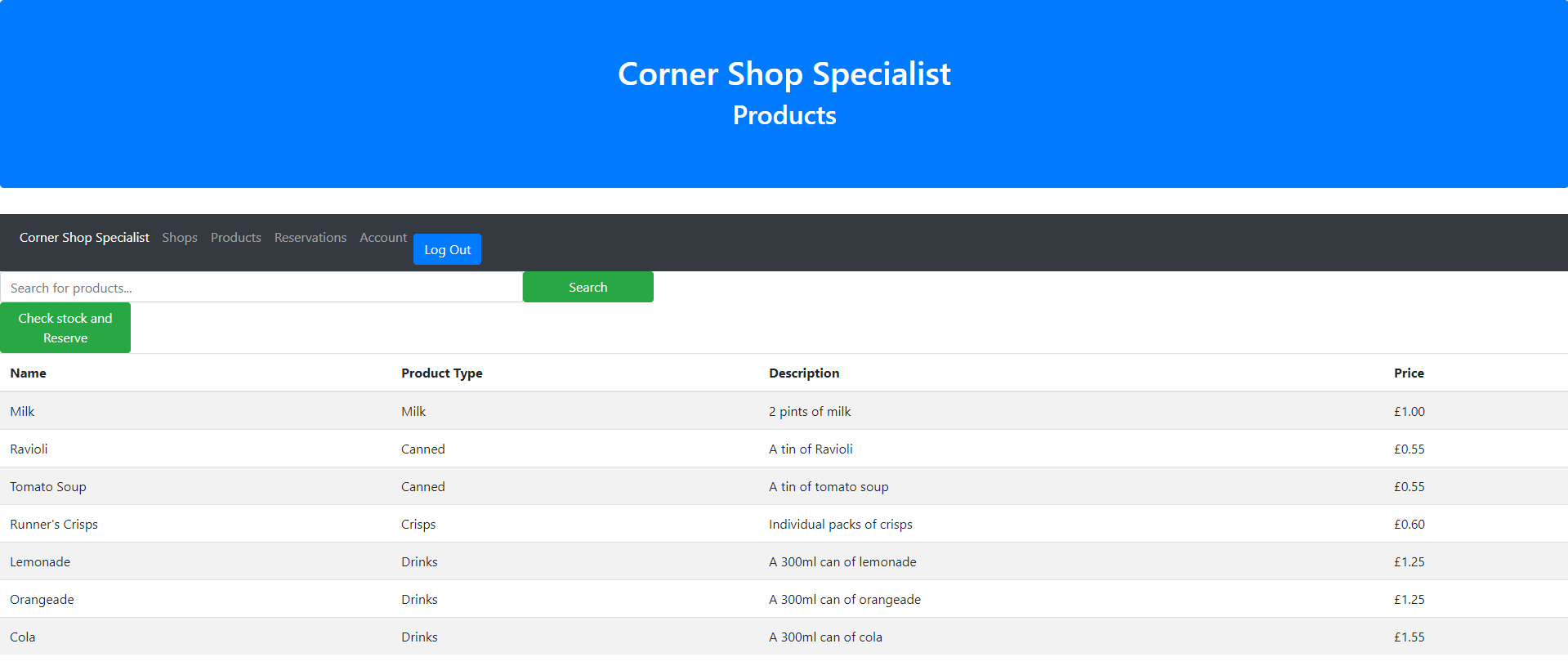


Figure 11: The products page with search bar from a logged in customers perspective

### Stage Review

This stage was a resounding success. All the desired outcomes were achieved, and this stage allowed for the project to get much nearer to its original timeline. This stage did take a long time to complete and could have been avoided if problems had been identified sooner and options were evaluated sooner regarding technologies.

However, it was tough in places to refactor the code that had already been written for the front end to access the database to work with new JSON values as some of the variables got renamed in the transfer of the back end. This created problems where data was not reaching the user interface and could sometimes take a long time to find and fix.

## Stage 4 – Staff Creation and Click and Collect

### Aims

This stage will address the user stories from sprint 3 that separates the account creation for staff so that admins can create all staffs and managers can only add staff that work at their shop. It will also add the ability for reservations to be created and viewed by managers.

### Objectives

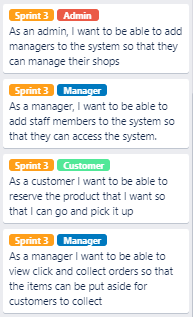


Figure 12: User stories to be completed in sprint 3

Other requirements were:

1. Implement a way to retain data across multiple HTML pages for users to view product information on a specific product.
2. Implement a routing server for the website’s navigation
3. Ensure that there was no way for any user to set up an account on the desktop application

### Outcomes of this stage

For this stage, two separate account creation pages were developed for the desktop application. These were similar in design to the account creation page in stage one but had more restrictions on what data the user would be able to put in as some data would be filled in when the data is sent to the server. For example, when a new staff member is added to the system the shop is assigned on the server based on the manager that is adding the account.

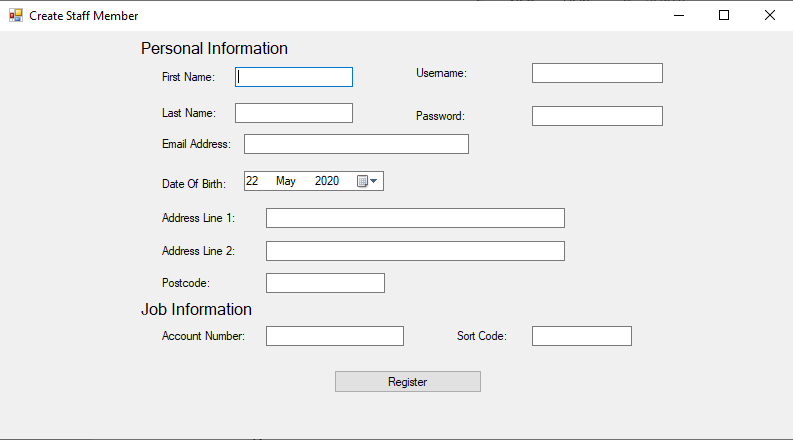


Figure 12: The form for creating a staff member that managers have access to.

There were issues encountered when it came to implement the click and collect service regarding data retention and processing the user inputs for reserving a product. It was intended at the start of the stage that the product information could have been transferred when a product is viewed and then the stock of that product could have been checked on that page before a reservation is made. However, this was changed in favour of having everything handled by drop down menus to make the back end less complex with data transfers due to the way the navigation had been decided to work. The way a customer would now reserve a product is by selecting the product and shop from 2 separate drop down menus and then clicking the reserve button.

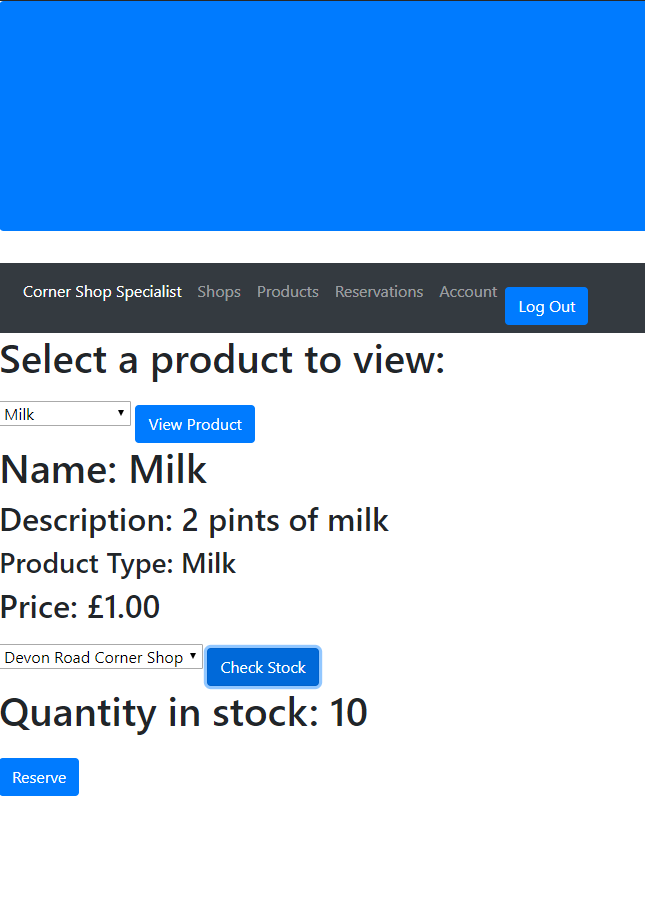


Figure 13: Reservation page for customers on the website.

It was decided when the click and collect service was being implemented that a routing server would not be possible to implement as there were issues with getting the navigation to work with the business logic already present in the website. So, it was decided that the navigation would be done by locally navigating to the HTML files directly to reduce complexity.

### Stage Review

This stage was very successful even though there were only a few user stories to implement as important infrastructure decisions were made to make sure that the code produced was consistent as the lack of a decision being made would have caused some code to be inconsistent with the rest causing bugs to occur that would require fixing. There were however the limitations with the click and collect service which was implemented but not to the desired quality.

## Stage 5 - Holiday, payroll, shifts and further deliveries

### Aims

The aim of the stage was to implement the features surrounding the management of staff such as shifts and holidays so that staff and managers could both see and edit staff and holidays and the remote access services so this data could be accessed on the website.

### Objectives

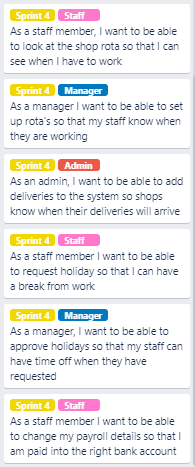


Figure 14: User stories to be completed in sprint 4

Other tasks to be completed in stage 5:

1. Code quality and cohesion checks to ensure code is consistent.
2. Integration testing on the node server.
3. Insertion of appropriate test data.
4. Fix a bug with date time objects sent from the desktop application
5. Add log out functionality for all web pages as it only works on index pages at this moment.

### Outcomes of this stage

Produced during this stage, were the systems to handle the staff portion of the program. To this end, a desktop section for viewing and approving holidays was produced that allows a manager to view all the holiday requests on the system, then go to another page in order to approve or deny any holidays that currently have the status of “Pending”.

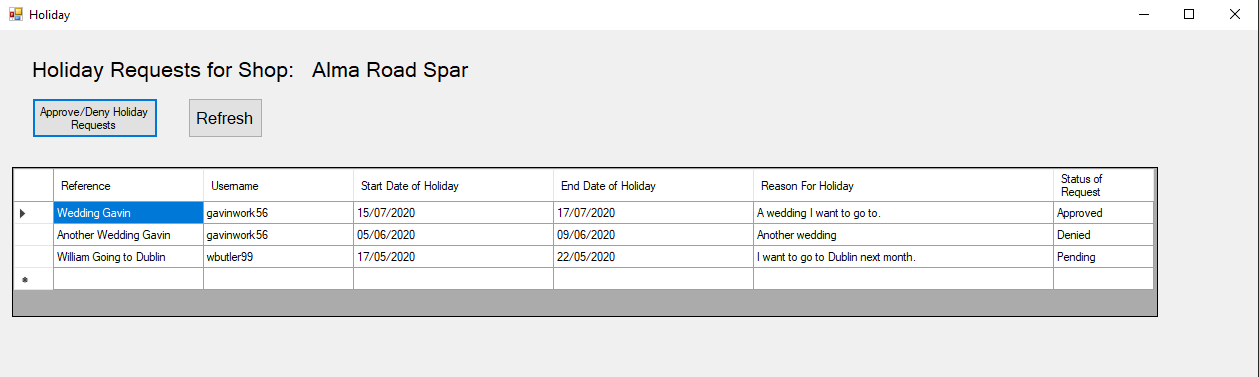


Figure 15: Desktop page to view all holidays a manager has on the system for his staff.

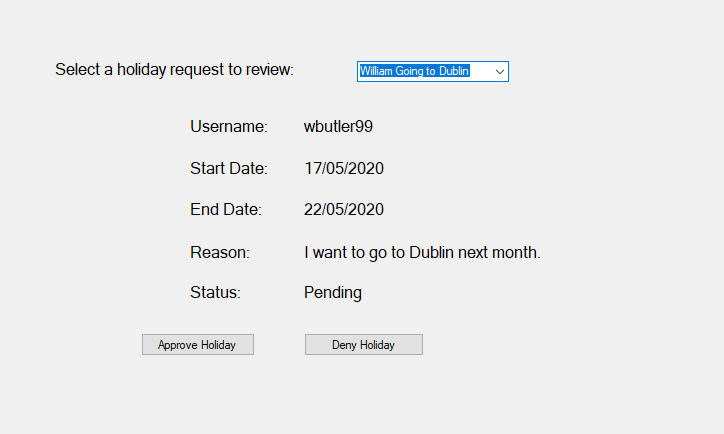


Figure 16: Desktop page to allow managers to approve or deny pending holiday requests.

The last user story for administrators was added in this stage giving them the ability to add new delivery dates and types for shops so that the managers can see when their next delivery is to be able to prepare both the shop and the staff to process it.

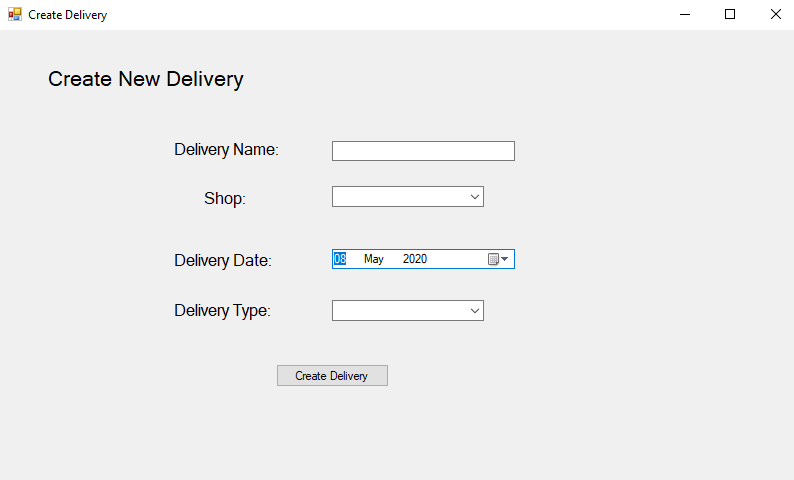


Figure 17: Desktop application page for admins to add new deliveries.

On top of the holiday request system, there was the system for shifts to be added to the system and accessed from the website so that staff could see when they are working from home to prevent the need for trips to work just to see when they are working next. Managers could also view all the shifts for their staff so they can see who is working on what day.

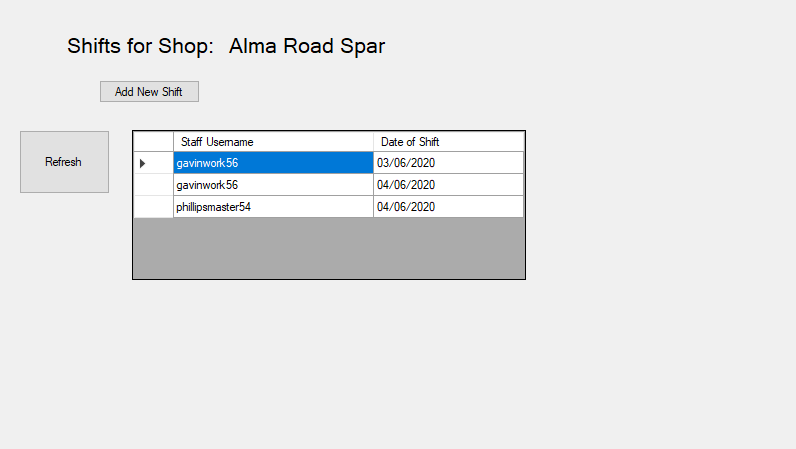


Figure 18: Desktop page to view all the shifts for the managers shop.

After these final pieces of functionality were implemented, the focus of the stage turned to polish and quality assurance by removing all inaccuracies and bugs that have appeared in the project throughout development in order to make the system look more professional and work as intended.

One of the significant bugs that needed fixing was the fact that when any date is sent from the desktop application to the server, there seemed to be an issue where the two types of date-time objects used by both C# and JavaScript behaved differently resulting in 1 hour being removed when saving to the database. This was resolved by adding an hour to any dates in the desktop before being sent to the server to ensure that the date stayed correct as the system does not at any point specifically require the time. This means so long as the date is correct the system will function as intended.

Another bug that has appeared was the fact that the log out buttons on the website would not function on any other pages other than the customer and staff home pages. This was quickly established to be down to conflicting JavaScript files with the server calls being housed separately to the log out script meaning that only the first loaded one would work. This was resolved by moving all of the JavaScript code into one script to remove the conflict and allow all the code to be executed.

After this it was intended that a round of integration testing using the mocha and chai NodeJS packages was to be performed to verify that the functions on the server are working as intended. However due to time constraints when reaching the end of the project, this was not completed as other items not associated to the final system needed to be completed with more urgency.

### Stage Review

Overall this stage went fairly well as by the end of the stage, the system was feature complete and all functionality had been checked for system breaking bugs and was working as intended. The major downside to this stage is the fact that the integration testing was not performed which gives a significant downside as there will be no automated testing performed on the system potentially allowing unforeseen bugs to show up during the products lifecycle that could have been avoided with testing.

# End-project report

## Review of project objectives

## Changes during development

There was a large amount of changes that were made during the development of this system and there were impacts and limitations to the system caused by them which will be analysed below.

### Technology Stack Changes

One of the major issues and changes that had to be made during the process of developing this system were the sweeping changes that needed to be made to the back end of the system. This had to be done at the end of the second stage of development after large amounts of time being dedicated to finding the source of the issues with the previous technology stack.

This led to the change to NodeJS and MongoDB to be used for the back end of the final product that provided much more in the way of limitations as to the quality of the functionality produced.

The main benefit of this change was that a new technology stack was not required to be learnt from scratch thus reducing the amount of time required to refactor the already existing parts of the system. There was also the benefit that the developer had used a similar framework more recently that the previous technology stack so was confident that it would function as intended with no unexpected issues.

There were however downsides to this change.

One of the major limitations of using MongoDB was that it is not a relational database. This had the impact of making relations between data much harder to define and having to be implemented by repeating a unique variable from the document that I want to associate. This makes it much more difficult to get all the data required during a request as multiple queries of the database must be made whereas in a relational database it would be one request containing all the required information.

There were also limitations with the framework used to make the business logic. Although NodeJS is very powerful when it comes to

### Functionality changes

DATA RETENTION

REPEATING DATA

ROUTING SERVER

# Project Post-mortem

This section of the report will review how the project went from the developer’s perspective and evaluate their own performance.

## What went well

I believe that overall this project was a success. I have created a system the achieves what I set out to do to a good standard within the time frame required.

All functionality classified in the core and stretch sections of my user stories have been achieved to some degree. This is a great achievement given the disruption and issues encountered throughout this project.

I feel that the use of vertical slice development really helped me focus in on one specific sub section of the system to implement that would then be easy to integrate smoothly into the wider system at play once completed.

The final technology stack that was used was a good choice to allow me to quickly and efficiently catch up from the lost time due to technology stack issues even if the final product was not as polished as I would have liked it. Using technologies that I had used previously allowed me focus less on technical issues that could arise and more on delivering functionality for the end users and satisfying the objectives of the project.

## What could be improved

I feel that the technology changes could have been made a little more proactively as I was stuck not making any progress for much longer than I probably should have been due to the issues encountered. This resulted in around 1-2 weeks of development time being lost being focussed on one issue when a technology stack change could have been made then the quality of the system could have been better come the end with more time to work in the new architecture.

I believe that I may have set myself too higher goal with the project given my tendency to set myself very high standards for my work. This then has the knock-on effect of making me very stressed and anxious that I will not achieve what I set out to do and reducing productivity.

Another improvement that could be made is more thorough research of the technology stacks proposed to be used on the project before development. This could include a full stack prototype to prove that the intended functionality can be implemented in the current framework and to iron out any large technical issues that were out of my control early on before they drew too much time away from the project.

I also feel that I could have documented the code better if I had the time to properly document it at the end of the project. This leads to the wider issue of time management when adapting to working from home during the current climate. I was becoming very anxious in the current situation as I was not used to this kind of event and had not accounted for this kind of thing happening. This had the effect of making me less focussed on my work having to worry about other things going on and the quality of the system produced I think reflects that lack of focus at times.

Another thing I feel that was lacking was any form of testing of the system both from a technical and user perspective.

When it comes to usability testing it was very difficult to organise as the country was in lockdown and the university was closed for face to face events. This meant that I could not organise to meet up with people to perform usability testing and setting up virtual testing was not possible for me given when I would have been ready to show something off due to the technical issues.

From a technical standpoint it was more of a case of development was already overrunning so there was not enough time to get testing performed. Plus I was not very familiar with how to test both a website and a desktop application as there was very little in the way of business logic on the client side and I did not have the time to learn the frameworks and perform the testing.

## Lessons learned

# Conclusions

# Reference List

Agile Manifesto, 2001. Manifesto for Agile Software Development**.** *Agile manifesto website.* Available at: <https://agilemanifesto.org/> [Accessed on 25th April 2020]

Microsoft Imagine, 2020. Microsoft Azure Dev Tools User Agreement. *Plymouth Microsoft Imagine portal.* Available at: <https://www.fose1.plymouth.ac.uk/MicrosoftImagine/> [Accessed 11th May 2020]

Mongo DB, 2020. Terms of service for Mongo DB services. *Mongo DB Website.* Available at: <https://www.mongodb.com/legal/terms-of-use> [Accessed 11th May 2020]

The Open Source Initiative, 2020. The MIT Licence. *The Open Source Initiative.*  Available at: <https://opensource.org/licenses/MIT> [Accessed 11th May 2020]

# Appendices

## User Guide

### Setting up the system

### Using the system

## Project Management

### Trello Boards

### Sprint plan