

UNIVERSITY OF PLYMOUTH

School of Computing, Electronics and
Mathematics

PRCO304

Final Stage Computing Project
2017/2018

Stock Exchange Themed Bar Application

BENJAMIN SHAFTO

STUDENT ID: 10487540

COMPUTER SCIENCE BSc (HONS)

Acknowledgements

Firstly, I would like to thank my project supervisor, Chris Johnson and Nigel Barlow for their continuous support and guidance through this project and my degree. I would also like to thank my parents for their moral support over the last 4 years of my degree.

Abstract

This report describes a software development project to design and develop a new stock exchange themed bar system for local bars and nightclubs, with the intention to add to the business's USP and broaden their customer base.

The report begins with market research into competitors, focusing on their (Unique Selling Point) USPs and methodologies. These results, furthered by an initial design, are presented and demonstrate how the project's core features were constructed. A general scope of the project's requirements is then outlined, followed by a detailed structure on the project management style.

The following processes were completed in incremental stages, each highlighting key stages in the project's growth from back-end conceptualisation, management orientated Graphic User Interface (GUI) and communication between the software, database and client-side website.

This was produced with programming languages such as Java, PHP, MYSQL, HTML, JavaScript and CSS. Further descriptions of this process are presented, followed by a usability and testing section. A final evaluation of the working system is compared to the real-world solution and the market research requirements to demonstrate all the core requirements.

A final project post-mortem is then presented, which verifies that the design and development approaches used were appropriate.

Details of project management issues, major bugs and miscellaneous deliverables are presented in various appendices.

Table of Contents

Acknowledgements.....	2
Abstract.....	2
Tables of Figures	5
1. Introduction	6
2. Background	6
3. Market Research	7
4. Method of Approach.....	8
5. Objectives.....	9
6. Deliverables.....	9
7. Legal, Social, Ethical and Professional issues	9
8. Project Management	10
8.1. Tools.....	10
8.2. Control Plan.....	10
9. Development Stages:	10
9.1. Stage 1: Research and Requirements Outline	10
9.2. Stage 2: Initial High-Level Design	11
9.3. Stage 3: Algorithm Design.....	11
9.3.1. Core Concepts	12
9.3.2. Bonus Features	13
9.3.3. How the algorithm works.....	15
9.3.4. Worked Example:.....	15
9.4. Stage 4: Database.....	18
9.5 Stage 5: Middleware and Integration	19
9.6. Stage 6: Website Design and Implementation	20
9.7. Stage 7: Validation and Verification Tests	20
9.8. Stage 8: Report.....	21
10. Project Evaluations	21
10.1. End-Project Report.....	21
10.2. Post-Mortem.....	22
10.3. Objective Evaluation	23
11. Conclusions	24

11.1. Lessons Learnt.....	24
11.2. Further Development.....	25
11.3. Final Thoughts.....	25
12. Statement of Word Count.....	25
13. Reference List.....	26
14. Appendices.....	26
14.1. Appendix A: Original Sketch of Website Functionality	26
14.2. Appendix B: Market Research Companies.....	27
14.3. Appendix C: Market Research Questionnaire.....	27
14.4. Appendix D: Live Tickers	28
14.5. Appendix E: Tablet Application	28
14.6. Appendix F: Database Research.....	29
14.7. Appendix G: Database ERD	29
14.8. Appendix H: Detailed Website Design	30
14.8.1. Login Screen	30
14.8.2. Drink Analysis.....	30
14.8.3. Main Menu.....	32
14.8.4. Settings.....	32
14.8.5. Stock History	33
14.8.6. Stock Page	33
14.8.7. Trend Analysis	34
14.9. Appendix I: Revenue Streams	34
14.9.1 Revenue Streams: Sponsorship	35
14.9.2. Revenue Streams: Initial Ideas.....	35
14.9.3. Revenue Streams: Competitor Analysis.....	35
14.9.4. Revenue Streams: Distribution Companies	36
14.9.5. Revenue Streams: Competitor Analysis.....	36
14.10. Appendix J: Website Screenshots	37
14.10.1. Live View	37
14.10.2. Net Profit.....	37
14.10.3. Category Comparison.....	37
14.10.4. Overall Comparison.....	38
14.11. User Guide	39

14.12. Project Management Artefacts:.....	42
14.12.1 PID.....	42
14.12.2 Highlights	48
14.12.2.1. Highlight 1	48
14.12.2.2. Highlight 2	49
14.12.2.3. Highlight 3	49
14.12.2.4. Highlight 4	50
14.12.2.5. Highlight 5	50
14.12.2.6. Highlight 6	51
14.12.2.7. Highlight 7	51
14.12.3. Exception Report: Stage 6.....	51
14.12.4. Resources Used	53
14.12.5. Installation Guide.....	53

Tables of Figures

Figure 1 - The Drink Exchange Website showing hardware	7
Figure 2 - Profitability of 'The Drink Exchange'	7
Figure 3- Reserve Bar Stock Exchange GUI	8
Figure 4 - Freeride Café GUI.....	8
Figure 5 - High-Level Design of the Database Architecture.	11
Figure 6 - Initial Pseudo Code for the Order Generation Algorithm.....	12
Figure 7 - Loop around Algorithm.....	13
Figure 8 – Smart Order Generation explanation with arrays and worked example.....	14

1. Introduction

Within any lucrative market, businesses are constantly searching for a USP to win over consumers and outdo their competitors. This stock exchange themed bar system (SEBS) project aims to satisfy that exact business demand by increasing profitability and broadening the demographic of consumers that a business would otherwise miss out on.

SEBS was designed with all the key features that would satisfy a real-world client, but for demonstration purposes some aspects of the system differ from how it would work in the real-world. In the real-world, SEBS would link with the business's point-of-sale (POS) system, then combine with the rest of SEBS. However, for demonstration purposes, orders are carefully created with a complex algorithm which manipulates the prices before linking with the other part of the project which performs analytics on the data whilst outputting the recalculated prices onto displays positioned around the bar.

The three main sections of the project are: Back-end order generation code, database storage and a managerial website to view the system at run time. Each of these components perform together to create a life-like simulation of how it would be operational in a bar, and how management staff would obtain extra information about how their bar was performing, both at real-time and offline.

2. Background

This project was inspired by a ski trip to France, during which an unusual and quirky ski pub was discovered which boasted a unique stock exchange themed bar and caught my interest immediately. Before noticing the bar I had never come across a stock exchange themed bar before and the bar seemed to suck in customers from the street and neighbouring bars.

A large part of this atmosphere was down to how immersed the customers were who had already been playing the game for a while. At every 5-minute interval customers could be heard from outside cheering as their favourite drink slumped down to an all-night low and the sound of air horns reverberated off the mountains and filled the valley. As more and more customers were drawn into the bar the atmosphere intensified with newer, more eager customer's queueing 5 rows deep, completely encapsulating the bar and the frantic bar staff in order to get to the front of the bar to grab a bargain before the drinks prices changed yet again.

This experience, along with research into the market, led to the creation of my own stock exchange themed bar system.

A stock exchange themed bar system has all the traits of a normal bar with additional features. These are primarily the fluctuating drinks prices and analytical website. It recalculates all drink prices every 5 minutes depending on quantity bought from the previous 5-minute interval. Customers then have a completely unique experience each time they use the system as they have complete control over how the drinks prices change. This process is described in further detail in section 5.

The system has been created for bars that are seeking an extra "Wow" factor for their bar; they are searching for that elusive USP that sets them apart from their competitors. By using the system, bars will draw in customers interested in the new concept that otherwise may have gone to a competitor or simply walked past. Customers will get a sense of enjoyment out of the system as they attempt to "play" and try to get the cheapest drinks that they can see displayed on the screens. The advantage to bars

using the system is increased footfall and higher profit margins through selling greater quantities of drinks that may not normally be so popular.

3. Market Research

During my market research, companies were reluctant to discuss the specifics of the algorithm as they didn't want to give too much information about their USP to a potential competitor. While there was an indication with the software and hardware they used, this meant the algorithm had to be designed from scratch. Appendix A shows an original sketch of the designs and how it could function.

Market research into stock exchange themed bars was varied. Online companies like *'The Drink Exchange'* offered a package deal whereby a bar would pay them for the setup, installation and

Figure 1 - The Drink Exchange Website showing hardware



unlimited customer support. The package was advertised through their website which recommended the hardware that could be used to best support their package and obtain the maximum from it.

Their USP was a mobile application which customers could order from as well as live tickers that were displayed around the bar along with TV displays that promoted the drinks prices at real time.

Figure 1 (Reference A) also shows a tablet application that managers could use to see trends emerging and

customer spending patterns. Over time, the bar application would have an average spending pattern on a user that could be desirable to drinks companies wishing to get more information about their consumers.

Figure 2 - Profitability of 'The Drink Exchange'

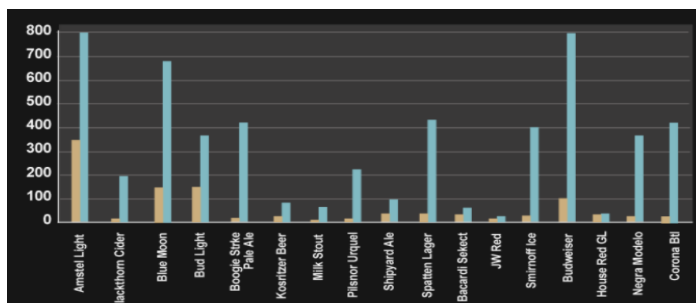


Figure 2 shows the number of sales of a drink. Orange shows sales before the system and blue shows sales of a drink when the system has been run over a night. This graph shows that not only are drinks more popular when the new system is running but customers were ordering more than they would normally would.

A more physical example of a stock exchange system would be *'The Reserve Bar Stock Exchange'* (Figure 3, Reference B) in London where they advertise drink prices on TV displays around the bar or via a mobile application where customers order drinks to their table. Their GUI is similar to *'Freeride Café'* (Figure 4, Reference C) as they both use bright colours to show the drink price and movement to the customer.

Figure 3- Reserve Bar Stock Exchange GUI



The 'Freeride C  fe' is situated within a small French ski resort and was the smallest bar that uses a stock exchange themed system. Their GUI has a feature which stops customers buying drinks within the last 5 seconds of every 5-minute interval so customers don't order a drink at one price and then pay a different one. This market research guided the project to realistic, achievable objectives.

Figure 4 - Freeride C  fe GUI



4. Method of Approach

Agile methodology has been used throughout the project and was essential for keeping up with the project schedule. Agile was deemed the most fitting methodology as initially the requirements weren't fixed and were subject to change pending the market research. These can be noticed when comparing the PID with the objectives in Section 5. After researching what functionality was in stock exchange themed systems in the current market, it was discovered that there was additional hardware that was

utilized to further the USP which wasn't necessary for the prototype. However, it did give a valuable insight into further work that could be implemented.

PRINCE2 was also a valuable project management tool when setting out the project objectives. The project was split into core stages: (i) Research; (ii) Software design & development; (iii) Hardware implementation; (iv) Testing and evaluation. These 4 core stages, along with the initial project plan (Section 14.4) showed that each segment of work was not to be started until the previous was finished. Learning from experience was another PRINCE2 principle that was incorporated into the project. From previous coursework I had learnt how to connect to a database efficiently using a RESTful API that uses GET and POST methods. This sped up implementation stage 5 which allowed more time to be delegated to tasks that hadn't been undertaken before, for example Google Charts.

When used effectively Agile is highly successful for development but project management can let the project down. To combat this, the project contains an initial project plan, change logs and incremental stage plans. Small, manageable segments of the project led to a higher work rate which gave the project overall stability.

5. Objectives

1. Research and analyse the current market to find the functionality being offered by bars with similar themes.
2. To research the current technologies that make similar systems attractive to various clients.
3. To review the profitability of the proposed system and the interest it would receive.
4. To improve the 'Wow' factor of potential client's bars to draw in more consumers.
5. To provide a user manual.
6. To produce an algorithm that dynamically changes the prices of drinks depending on consumer demand.
7. To provide the client with a simple online GUI to view the system and its additional features.
8. To provide the client with a consistent and reliable system.

6. Deliverables

- User Manual
- Software Package:
 - NetBeans order generation code.
 - Database – MYSQL.
 - Website – PHP.
- Project Initiation Document.
- This report.

7. Legal, Social, Ethical and Professional issues

Since none of the order generation code and other technologies use or store any sensitive data, there is no concern for this project breaching any laws regarding monitoring or data protection.

8. Project Management

8.1. Tools

During the build of this project a variety of software tools were used. A full list of technologies used can be seen in Section 14.7. However, the main tool used for most of this project was NetBeans IDE (version 8.2) for the development of both the order generation code and website. The order generation code was coded in Java whilst the website was coded in PHP, HTML and CSS. NetBeans was also utilized for connection between order generation to the database and the database to the website. MYSQL was used for the database setup and queries. Not all listed programs were ones selected to be used in the PID as some of the tools were not free to use.

8.2. Control Plan

The control techniques listed below have been implemented throughout the project's development.

- **Weekly highlight Reports:** These documents kept track of the progress that was being achieved each week as well as showing slows in the project that needed addressing. These also provided extra motivation as the list of tasks being achieved each week increased.
- **Weekly project supervisor meetings:** These meetings provided a sometimes much needed opinion on the work that was being carried about and the timeframe required to do it in.
- **End stage reports:** These ensured that all the deliverables and objectives that were originally set out were achieved to the standard that was necessary.

9. Development Stages:

9.1. Stage 1: Research and Requirements Outline

Stage 1 consisted of researching existing stock-exchange themed bar systems within the current market. This was achieved by outlining the core features of each businesses system as well as any “wow” factors that they used. Appendix B shows the full list of businesses that were researched as well as a questionnaire (Appendix C) which was used to analyse how each system was created and tailored to meet the needs of that company. Not all businesses researched were able to conduct the questionnaire due to out of season hours. The information gathered about those businesses was instead collected from their websites.

Another factor that wasn't foreseen was the difference between the sizes of companies that offered a stock-exchange themed night out. Companies like *'The Drink Exchange'* offered a fully immersive experience for the customer, with a mobile app to order drinks to their table, live tickers (Appendix D) and a tablet application (Appendix E) for managers to analyse the system whilst it was running. Some businesses only offered the most basic implementation of a themed stock-exchange night. *'Freeride Café'* offered customers the experience daily after 9pm when they would order drinks from the bar whilst the prices changed every 100 seconds. There were no live tickers, analytics or extra features to draw customers in. As *'Freeride Café'* was situated within a ski resort where there were a limited number of bars to compete against, there was less of a need to spend vast amounts on “wow” factors.

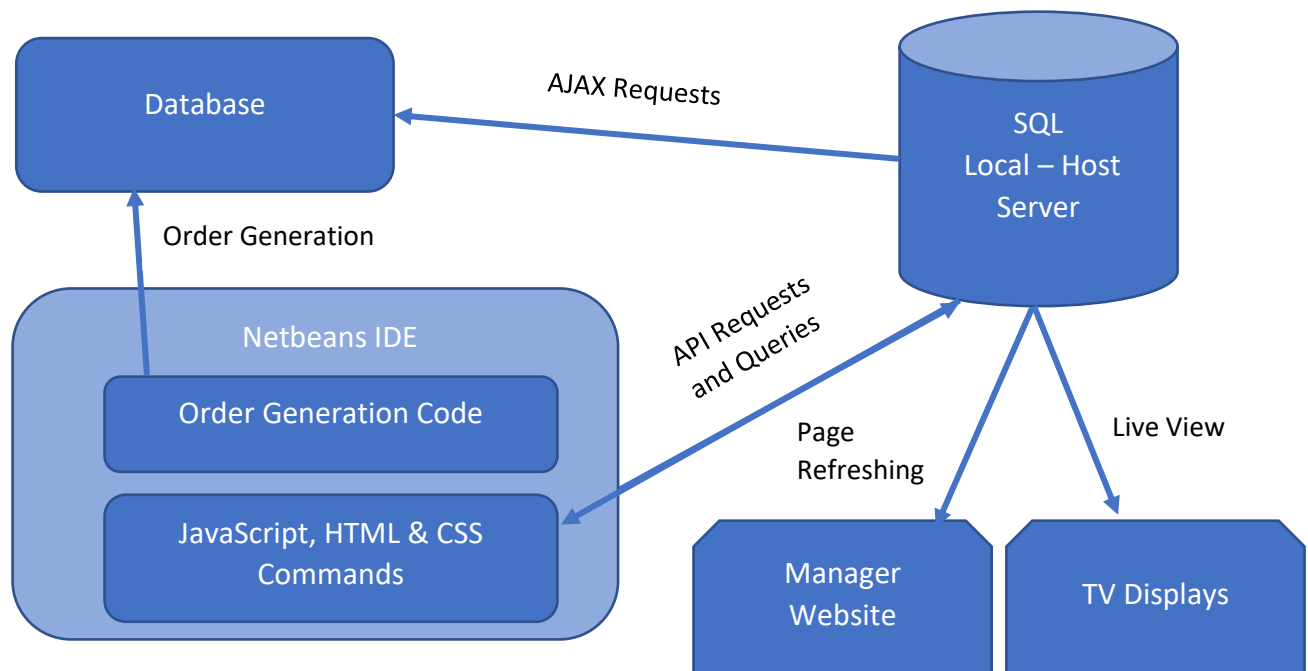
The conclusion gained from this research was that the capacity of the bar corresponded to the functionality that was offered.

9.2. Stage 2: Initial High-Level Design

The majority of the program was designed and built in a NetBeans IDE that contains two separate packages of code; the order generation and the website. In a real-world situation, the data would be sent every 5 minutes instead of every 5 seconds for the purposes of demonstration in this project.

Once the database has received the data it populates the relevant tables. The SQL local-server contains a connection pool which handles all communication between the server and external software. The server queries the database every 5 seconds to check if any new data has been added, before displaying recalculated drink prices on the TV outputs. The connection pool also handles requests to the website which managers would make when using the web application. Figure 5 shows the software, the direction the data travels and the contents of the data itself.

Figure 5 - High-Level Design of the Database Architecture.



9.3. Stage 3: Algorithm Design

The algorithm was the backbone of this project and was imperative to its success. To gain an understanding of what core features would be necessary market research was carried out on similar themed systems already in the market. The areas that were focused on the most were how accurately companies tried to mimic a real-life stock exchange so that the customers had a fully immersive experience and the level of functionality that the businesses offered. From this research a pseudo coded algorithm was loosely designed (Figure 6) with the premise that every drink ordered will affect every

other drink's new price in the next 5-minute period. This 6-part process would then be repeated every 5 minutes whilst the system is running.

Figure 6 - Initial Pseudo Code for the Order Generation Algorithm

```

Features of the Algorithm for calculating the drinks prices.
- The sum of all the drinks prices will always equal the same amount.
- The drinks will be divided up into sections that won't always equal the same (extra feature?)
- Spirit/mixers, beers, wines, shots, cocktails, etc...
- Updates every 5 minutes - with a 5 second countdown beforehand.
- Has a "market crash" feature whereby a certain section would have '50% off beers'.
- Each drink would have a minimum and a maximum cost.

- Algorithm for alternating the prices of the drinks:
1. Create a drinks and corresponding prices 'drinks', 'prices', 'prices' and 'allPrices' array
2. Create and start a timer for 5 minutes
3. Start loop of x (100-TP) amount of orders to start the After 5 minutes is up (maybe 4min55 for countdown?) process
   3.1. The number of times a drink was ordered is totalled for each drink (vodka coke, vodka lem, beer, cocktails...)
   3.2. The drinks are ordered and ranked in most-> least popular
   3.3. The top 50% of the drinks are matched oppositely (most with least, 2nd most with 2nd least)
       3.3.1. This means that no two drinks are always affected by eachother, mimicking a stock-exchange!
   3.4. newPrices = oldPrices
   3.5. newPrices are stored in the 'prices' and allPries' arrays
4. Start animation of 5 second countdown
5. 5 minute timer ends.
6. newPrices are also sent through the server to the database to be stored (this could be done at the end of every end of day?)

```

A concept which was considered for inclusion within the algorithm was a 'market crash' feature. During the market research some bars were found to have this feature, where a single drink or drink category would drop dramatically in price with flashing lights appearing on the displays saying, for example, "50% off Carling". This feature could potentially benefit both the consumer and the bar; it could be designed in the bar's favour by putting the 'market-crash' feature on drinks with high profit margins, so that they would make a profit even at 50% off, whilst being cheap for the consumer to purchase and therefore increasing sales.

The bar could then promote offers to drink suppliers so that their product could appear on the 'market crash' more frequently than their competitors.

9.3.1. Core Concepts

After completing market research and an initial design of how the algorithm could function a set of core concepts were forming. These ensured that the algorithm would give the consumer the most immersive experience whilst having enough functionality that any bar could utilize it.

The first core concept is that each drink ordered affects every other drink's price in the next 5 minute interval. This mimics a real-life stock-exchange market and will give the consumer a more immersive experience. This is explained in detail in section '9.3.3. How the algorithm works:' as every drink is calculated and ordered differently each 5 minute interval.

The second concept is that the most expensive drink decreases in price by the same amount that the least expensive drink increases. This ensures that no single drink changes price by a drastic amount without having a corresponding effect on another drink. It would also be noticeable to the consumers the impact they have on varying drinks prices, as if a lot of people are ordering, for example, Becks beer within a certain time interval, the price will soar upwards in the next interval whilst a drink that hasn't been purchased very much at all will drastically decrease in price.

Thirdly, an important feature is that a drink's price does not have to change every interval. This would happen if the last two drinks to be compared had been ordered exactly the same number of times within the last interval. In Figure 7 it shows which loop around the algorithm the drinks prices are updated.

Figure 7 - Loop around Algorithm.

1	2	3	4	5	6	7	8	9	10
1 st loop	2 nd loop	3 rd loop	4 th loop	5 th loop	5 th loop	4 th loop	3 rd loop	2 nd loop	1 st Loop

Each loop goes through the array of 'drinkQuantities' searching for the highest and lowest value that hasn't previously had their drink price updated.

The fourth concept is a bias variable that increases the amount by which the drinks prices are changed every interval. This was a concept that wasn't in the initial design but after the algorithm developed this variable made the system produce an output that was more suited to a bar. Prior to introducing this bias variable, the prices were only changing within the range of 20p-£1.40, which isn't that much of a change to incentivise consumers to feel like they're getting a bargain by buying another drink. Therefore, by having this modifier, the drink prices will change in the region of 30p-£2.10, which is much more of a difference in price therefore much more attractive to the consumer.

The fifth concept is that within each interval the most expensive 50% of drinks will drop in price and the bottom 50% of drinks will increase in price. This concept is superseded by concept three if the last two drinks to be compared have the same number of orders in the last interval. This links to the sixth concept whereby the drink prices change relative to the quantity of drinks bought only in the previous interval and no others. This was important as when conducting the market research section some bars calculated the new prices based off an average of the last 5 intervals. This tactic seemed less consumer friendly as the average price of a drink would move significantly slower, resulting in a slower paced system. When in comparison a stock-exchange bar should be fast-paced.

The seventh core concept is that the maximum, minimum and starting price of all drinks and initialised before the system is run. For this project simulation all drinks have a maximum price of £8.99 and a minimum price of £2.99, with a starting price of £5.99. This could be fully customizable to the bar where they could move these variables around for different nights to maximise profits according to demand. For example, on New Year's Eve the maximum and minimum price could be increased. On slow nights, the prices could be decreased to encourage customers to come into the bar. The final core concept is that there is no limit on the amount that a drink could increment or decrement within a single interval if they stay between the minimum and maximum prices. This allows the system to be more sporadic and dissimilar each time it's run.

9.3.2. Bonus Features

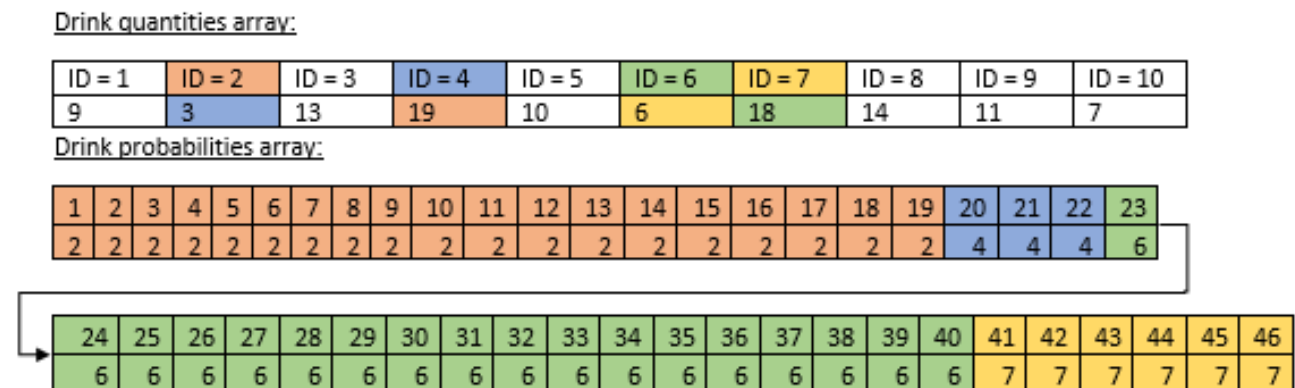
Completing the market research section gave me a previously unseen insight into the different forms of functionality that bars offered. Where one bar would offer functionality such as live tickers displayed around the bar, giving the consumer an extra way to absorb the drink information, others would have no bonus features apart from differing the amount of time between the intervals on different nights. From this, bonus concepts were constructed that would make the system unique and differentiate it from the others in the market.

The first bonus feature is smart order generation. In an artificial situation, this allows the simulation of customers placing drinks orders at the bar. This bonus feature wouldn't exist in a real-life product; however, it contains an algorithm to generate orders every 5 minutes, thus mimicking a real-life system.

The first time the system is run the prices are started at £5.99 and drink orders are chosen at random. However, after the first interval, every other interval uses smart order generation. This process is explained in further detail in Section 9.3.3 & 9.3.4. It works by populating an array called 'drink probabilities' with drink IDs. The amount that a single drink ID is added into the array is reliant on the drink it's being compared with in the order generation algorithm.

For example, if the most popular drink (ID = 8) has been ordered 18 times and the least popular drink (ID = 19) has been ordered 3 times. The 'drink probabilities' array will have the least popular drink ID being added 58 times and the most popular drink 9 times. This ensures that in the next interval the chances of the system randomly picking the previously most popular drink is a lot smaller than the least popular drink based off the number of times the drink ID's appear in the 'drink probabilities' array (Figure 8).

Figure 8 – Smart Order Generation explanation with arrays and worked example.



From the 'drink quantities' array, where ID = 4 (19) is the maximum and ID = 2 (3) is the minimum. The 'drink probabilities' array is populated with drink IDs. Drink 2 is added 19 times and drink 4 is added 3 times. Then, on the next loop of the algorithm drink 6 is added 18 times and drink 7 is added 6 times.

The rest of the bonus features are part of the analytics on the web application. The net profit feature allows users to switch between different categories of drink: Beer, Cider, Cocktails, Shots, Spirit / Mixers and Wine. The formula below shows how the net profit is calculated.

$$\text{Net Profit} = \text{Quantity} \times \text{Current Interval Price}$$

This information is fetched through an AJAX call to the database from the locally hosted server which is then read back into a: drink price, drink quantity and drink percentage movement array. These are then utilized within the different PHP pages.

Another analytical feature of the web application is the category comparison. The user gets the option of selecting the same categories of drink as they could within the net profit graph, with the added option of 3 views. The first view is the changing price of each drink in the category over the time the system is run. The second is the changing movement percentage of each drink in the category chosen.

Lastly, the changing quantity of each drink within the category chosen. Alternately, the user has the option to view an overall comparison of the drinks price, movement or quantity together on one chart.

These bonus features distinguish this project from similar systems on the market. It would benefit bars by providing them with additional information about sales and equip them with tools that allow trends to be spotted which would have previously gone unnoticed. The combination of the themed system, paired with the web application analysis tools provides a bar with a strong USP over its competitors.

9.3.3. How the algorithm works

1. The total quantity of orders for a 5 minute interval is calculated.
2. The amount of times each drink is ordered is also totalled within a 5 minute interval.
3. A variable called 'percMovement' is used to calculate the percentage of each drink ordered out of the total number of drinks ordered, which is then stored in an array. For example. 8 orders out of a total of 40 would be 20%.
4. Drinks are ranked highest to lowest according to the 'percMovement' percentage value.
5. A drink's new price is calculated by taking its old price + or – a movement factor. This movement factor is called 'result'.
6. To calculate 'result', the following steps occur:
 - 6.1. The drink with the highest 'percMovement' value has its corresponding price divided by 100 and stored as an independent variable 'perc' for all the drink IDs.
 - 6.2. For each drink ID, the same 'percMovement' value is multiplied by 'perc' and stored as a variable called 'result'.
 - 6.3. The variable 'result' is multiplied by a static variable of 1.5 to increase the amount that the drink prices move before being rounded to 2.d.p.
7. The highest and lowest 'percMovement' values have their drink IDs fetched.
8. The 'result' variable is added to the highest 'percMovement' drink price which has just been fetched.
9. The same 'result' is also subtracted from the lowest 'percMovement' drink price.
10. These two new drinks prices are then stored in an updated drinks price array.
11. The loop is repeated around again, this time for the second highest and lowest 'percMovement' values until all values have been compared and updated.
12. The new drink prices are then sent to be stored and manipulated in the database.

9.3.4. Worked Example:

1. The total quantity of orders for 6 drinks over a 5 minute period is 43.
2. The total number of times a drink is ordered with a corresponding drink ID is shown below.

ID = 1	ID = 2	ID = 3	ID = 4	ID = 5	ID = 6
8	2	5	13	4	11

3. A variable called 'percMovement' is used to calculate the percentage of each drink ordered out of the total. This is then stored in an array. For example. 8 orders out of a total of 43 is 18.6%.

ID = 1	ID = 2	ID = 3	ID = 4	ID = 5	ID = 6
PercMov = 18.6%	PercMov = 4.65%	PercMov = 11.63%	PercMov = 30.23%	PercMov = 9.3%	PercMov = 25.58%

4. Drinks are ranked highest to lowest according to the 'percMovement' value.

ID = 4	ID = 6	ID = 1	ID = 3	ID = 5	ID = 2
PercMov = 30.23%	PercMov = 25.58%	PercMov = 18.6%	PercMov = 11.63%	PercMov = 9.3%	PercMov = 4.65%

5. A drink's new price is calculated by taking its old price + or – a movement factor. This movement factor is called 'result'.

6. To calculate 'result', the following steps occur:

6.1. The drink with the highest 'percMovement' value has its corresponding price divided by 100 and stored as an independent variable 'perc' for all the drink IDs.

ID = 4	ID = 6	ID = 1	ID = 3	ID = 5	ID = 2
PercMov = 30.23%	PercMov = 25.58%	PercMov = 18.6%	PercMov = 11.63%	PercMov = 9.3%	PercMov = 4.65%
Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99
Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599

6.2. For each drink ID, the same 'percMovement' value is multiplied by 'perc' and stored as a variable called 'result'.

ID = 4	ID = 6	ID = 1	ID = 3	ID = 5	ID = 2
PercMov = 30.23%.	PercMov = 25.58%	PercMov = 18.6%	PercMov = 11.63%	PercMov = 9.3%	PercMov = 4.65%
Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99
Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599
Result = 1.81	Result = 1.53	Result = 1.11	Result = 0.70	Result = 0.56	Result = 0.28

6.3. The variable 'result' is multiplied by a static variable of 1.5 to increase the amount that the drink prices move before being rounded to 2.d.p.

ID = 4	ID = 6	ID = 1	ID = 3	ID = 5	ID = 2
PercMov = 30.23%.	PercMov = 25.58%	PercMov = 18.6%	PercMov = 11.63%	PercMov = 9.3%	PercMov = 4.65%
Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99
Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599
Old Result = 1.81	Old Result = 1.53	Old Result = 1.11	Old Result = 0.70	Old Result = 0.56	Old Result = 0.28
Result = 2.72	Result = 2.30	Result = 1.67	Result = 1.05	Result = 0.84	Result = 0.42

7. The highest and lowest 'percMovement' values have their drink IDs fetched.

ID = 4	ID = 6	ID = 1	ID = 3	ID = 5	ID = 2
--------	--------	--------	--------	--------	--------

PercMov = 30.23%.	PercMov = 25.58%	PercMov = 18.6%	PercMov = 11.63%	PercMov = 9.3%	PercMov = 4.65%
Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99
Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599
Old Result = 1.81	Old Result = 1.53	Old Result = 1.11	Old Result = 0.70	Old Result = 0.56	Old Result = 0.28
Result = 2.72	Result = 2.30	Result = 1.67	Result = 1.05	Result = 0.84	Result = 0.42

8. The 'result' variable is added to the highest 'percMovement' drink price which has just been fetched.
9. The same 'result' is also subtracted from the lowest 'percMovement' drink price.
10. These two new drinks prices are then stored in an updated drinks price array.

ID = 4	ID = 6	ID = 1	ID = 3	ID = 5	ID = 2
PercMov = 30.23%.	PercMov = 25.58%	PercMov = 18.6%	PercMov = 11.63%	PercMov = 9.3%	PercMov = 4.65%
Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99
Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599
Old Result = 1.81	Old Result = 1.53	Old Result = 1.11	Old Result = 0.70	Old Result = 0.56	Old Result = 0.27
Result = 2.72	Result = 2.30	Result = 1.67	Result = 1.05	Result = 0.84	Result = 0.42
New Price = £8.71					New Price = £3.27

11. The loop is repeated around again, this time for the second highest and lowest 'percMovement' values until all values have been compared and updated.

ID = 4	ID = 6	ID = 1	ID = 3	ID = 5	ID = 2
PercMov = 30.23%.	PercMov = 25.58%	PercMov = 18.6%	PercMov = 11.63%	PercMov = 9.3%	PercMov = 4.65%
Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99
Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599
Old Result = 1.81	Old Result = 1.53	Old Result = 1.11	Old Result = 0.70	Old Result = 0.56	Old Result = 0.27
Result = 2.72	Result = 2.30	Result = 1.67	Result = 1.05	Result = 0.84	Result = 0.42
New Price = £8.71	New Price = £8.29			New Price = £3.69	New Price = £3.27

ID = 4	ID = 6	ID = 1	ID = 3	ID = 5	ID = 2
PercMov = 30.23%.	PercMov = 25.58%	PercMov = 18.6%	PercMov = 11.63%	PercMov = 9.3%	PercMov = 4.65%
Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99	Price = £5.99

Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599	Perc = £0.0599
Old Result = 1.81	Old Result = 1.53	Old Result = 1.11	Old Result = 0.70	Old Result = 0.56	Old Result = 0.27
Result = 2.72	Result = 2.30	Result = 1.67	Result = 1.05	Result = 0.84	Result = 0.42
New Price = £8.71	New Price = £8.29	New Price = £7.66	New Price = £4.32	New Price = £3.69	New Price = £3.27

12. The new drink prices are then sent to be stored and manipulated in the database.

The working example shows the system working for one 5-minute interval with 6 drinks. The full system contains 35 drinks and runs for as long as is necessary.

9.4. Stage 4: Database

In Stage 4, several variations of potential databases were researched to assess how best to store the information about drinks and associated data. A MYSQL database was chosen for this project as it was the most fit for purpose. A further explanation why this database was chosen can be seen in Appendix F. The core details for each of the 35 drinks were added into the database at this stage and would be modified as the project progressed.

An initial implementation of the database was constructed for the middleware to send the NetBeans order generation code to the database. At this stage only, a small amount of data was sent to the database via the middleware to test the architecture of the proposed system. A brief design of the website page 'Live View' was set up so data could be received from the database and displayed onto the page statically. These webpages were created from the progressive designs in the previous stage.

As the database wasn't overly complex to create, the stage was completed before the 2-week allocated period. This extra period was instead used to validate the data that would be inputted into the database tables and complete the entity relationship diagram (ERD) (Appendix G). As this data was later going to be pulled out of the database to use on the website, it was essential to check the accuracy of the data in the tables early in the development stages. Queries that the website would require were also loosely designed in preparation.

Stage 4 was also the period where the final report was started. The structure of the report was laid out with the weekly highlight reports being added into the appendices. This report was recognised as a substantial task and starting it early revealed the sections that would be the most time consuming and that would require the most focus; one of these stages being the explanation of the algorithm for generating the orders.

The downside of this stage was that the work being carried out wasn't following the initial project plan. The website implementation wasn't supposed to be developed until the end stages of the project with most core requirements being focussed on the algorithm and connecting it to the database. After realising this flaw in the development, the design was primarily stripped down into smaller tasks to focus work and boost motivation for the project by completing numerous smaller tasks.

9.5 Stage 5: Middleware and Integration

Stage 5 was designated for the development and integration of the middleware that would link the database, order generation code and website together. The motivation behind completing this stage early was that the website could be viewed and tested with the actual data that was being produced by the order generation code. This would throw up any bugs in the system that could otherwise go unnoticed.

The middleware that was selected to use within the project was a MySQL localhost server for connecting the database to the NetBeans code and a XAMPP server to connect the website to the database and NetBeans code. The reason two servers were involved was because the project was completed in two parts and then merged later. The MySQL server allowed simple SQL commands that send the order generation data to the database every 5 seconds. This data was then extracted from the database with AJAX queries on the XAMPP server, populating relevant website pages. The website pages were coded with PHP as it's a free open source language and had been practiced in part from a previous module. PHP also allowed for easy integration with CSS and HTML. Both were crucial to create a dynamic website page that showed the prices being displayed and updated every 5 seconds. Starting the process of building the website early gave a longer period to research areas such as PHP, HTML and CSS that hadn't previously been learnt in detail but which would be necessary to gain a sufficient level of understanding to incorporate their use.

In the initial stages, setting up the structure of the website was essential to gauge how long it would take to get the functionality implemented. Once a basic framework was in place, the middleware was added and tested to see if the AJAX requests to the database were successful. After the first packet of information had been sent and displayed on the webpage, the next challenge was getting the data to be sent from the database every 5 seconds.

This dynamic webpage functionality caused a lot of problems for the project as setting up the connection pool was a skill that hadn't been previously learnt. Once researched and some basic tutorials had been looked over, adding a single AJAX request to send to the database was tested. Once one AJAX request was made, the other pages were easier to create as they contained similar requests. However, within each page the code was changed to tailor to the functionality that was being developed. This was then later optimized by creating a timer event which would be used to send a request to the database every 5 seconds.

By using a connection pool with the XAMPP server, it was possible to send a request to the relevant database table / query asking if any more recent information had been added than was currently being displayed. The connection pool handled all the requests to the database to display data on the analytics webpages as well as navigation between pages.

The MySQL server was used in conjunction with Java Database Connectivity (JDBC). JDBC can be used to access any kind of tabular data from a relational database and was chosen for these reasons. It was also very easy to implement to the project as only one JDBC driver was needed to add to the project to create a connection.

During the next week of this increment, the focus shifted from setting up the structure to creating aesthetically pleasing webpages that could be easily viewed by the end user. This consisted of the analytics pages that would be used by the bar staff whilst the system was running. The 3 main pages

that were created were: net profit, category comparison and overall comparison. This meant that the pages could be tested with the order generation data whilst the system was still being developed, saving the time of trialling the pages with test data beforehand.

9.6. Stage 6: Website Design and Implementation

This core features of the website designs had already been implemented in the previous stage so the middleware could be setup. A rough sketch of how the website could function can be seen in appendix A, with a later, more detailed design in appendix H. These designs weren't final but met the core requirements previously set out in section 5 whilst giving a clear sense of direction of how to proceed development.

This stage was then concentrated on adding the additional functionality and aesthetics to the web pages for the end user to be able to get the most out of the system. By having a website with all the analytics on one page, the user wouldn't have to waste time navigating between pages, more simply, they could browse all the views by selecting a clearly laid out combination of radio boxes and drop-down boxes that would be specific enough to show a drinks price changing or a categories percentage movement vary over the duration of the night.

Towards the end of the stage, the focus shifted onto removing bugs and ensuring that the program ran as smoothly as possible. A fault that was noticed was that the drink prices weren't changing at each interval as initially the duration was set to 5 seconds which didn't allow any lag-time. Occasionally they would be the same as the last interval because the most recent drinks details hadn't yet been sent to the database. By changing the speed that the orders were being generated and sent to the database to 4.5 seconds, drinks prices and other corresponding details would already be in the database before they were needed.

9.7. Stage 7: Validation and Verification Tests

This section shows the functionality of the website and the corresponding validation and verification details that show its purpose within the system.

- Live View

Verification: Being able to view the live prices is essential to the application that has been developed. This screen (Section 14.9.1) would be sent to the TV displays around the bar as to immerse the consumers in the application. The consumer can see the drink name, price and movement before it's updated at each interval. The movement is also colour coded. Green for a movement that makes a drink cheaper and red more a more expensive drink change.

Validation: The user is static on the webpage and cannot use the side menu to navigate off the page. The user can use the back button on the webpage or type another webpage into the URL. This is so the side menu isn't viewable to the consumers looking at the page.

- Trend Analysis

Verification: Trend Analysis is just as crucial to the project as the Live View. The user can navigate between 3 main features; net profit, category comparison and overall comparison. Each of these features can show the price, percentage movement or quantity in addition to each of the categories of drink; Beer, Cider, Shots, Spirit / Mixers, Cocktails and Wine.

Validation: The webpage controls the flow of the information; the user can absorb all the information on the Google Chart as soon as there is information in the database. Hovering the mouse over a data point on the graph allows the user to see the exact value at that 5-minute interval.

9.8. Stage 8: Report

The report was originally meant to be completed at the end of the build. However, due to the scale of the report it made sense to commence it at an earlier stage. This was discussed with the supervisor and it was agreed that it would be sensible to get a head start on the report.

The first stage of the report was setting out the general structure and what each section would consist of. This allowed the report to have a clear flow and lessened the length of time that the report would take to write up formally as all the points for discussion were already in place. After each week's objectives had been completed, the report was updated with the information about that stage. If there was a week where no highlight report was generated, it was noted that there was a backlog of work that hadn't been completed.

After most of the sections in the report had been drafted, there was more time to look over each one to update any areas that had been changed or were no longer relevant to the project. This was beneficial towards the end of the project, as more time could be focussed on refining certain areas in the project's code that needed work, in addition to adding documentation into the code where applicable.

10. Project Evaluations

10.1. End-Project Report

Throughout the duration of the project, many changes were made that impacted the project. Before each change was made, the possible outcomes were evaluated using risk management techniques. Below is a list of changes that were made that differed from the initial project plan:

- Changes from the plan

As aforementioned, there were numerous alterations to the project from the project initiation document. The changes that arose were treated carefully by the developer and were justifiable based on the experience of the developer and the market research that was undertaken in section 3 of this report. A Gantt Chart was created a few weeks into the project to gain a wider perspective of the changes undertaken so as to not incur any time management problems towards the end of the project. Contingency timing was then allocated to the new changes. For example, changing increment 3 from the PID (Appendix 15.4). This increment was initially thought to be very beneficial to the project when the PID was written. However, as the project progressed, working on the functionality of the website added greater value than having three different themes for the end user to choose from. This led to the aesthetics of the web application looking less developed, but the actual functionality of the pages being much higher, which is crucial to the projects USP; the USP being the "wow" factor that businesses are looking to invest in.

- Changes from the PID

Although the developer tried adhering to the PID, from an early stage it was apparent that the initial plan was not going to be as accurate as anticipated. When this was discussed with the supervisor, it was suggested that the developer lacked experience undertaking projects of this scale and because of this,

certain changes were made to ensure that the core aspects of the project were completed before any additional features started development. Inadequate time estimation skills inevitably occur when developers lack experience in projects of this scale. This is a skill that has improved over the course of the project and will be improved further in the future through more experience.

- Changes throughout

A major change that was implemented part way through the project was switching from a HTML based web application to PHP. During the development of dynamic web pages, PHP was found to be a much easier application to use when trying to refresh only part of the webpage so that the 'Live View' could be updated with new prices at each interval. This meant creating a whole new web application and starting the design again from scratch. The developer was hesitant to commit to the change as he had no experience in the language and half the allocated amount of time for this section had already been used trying to implement the dynamic functionality with HTML pages. Despite this, numerous hours were spent completing and following tutorials so that the changes could be implemented and the website's core functionality was achieved because of this new skill. This altercation did incur a time penalty and an exception report was completed to show this in Section 14.6.

10.2. Post-Mortem

On reflection, some of the initial outlined objectives of this project were not suitable and did not reflect how a stock market themed bar system should be designed and implemented. Before starting the build, the developer had minimal experience with this type of project, but had a strong passion for producing the system which stemmed from an immersive experience in a French bar that was running a unique variation of a stock exchange themed system. The revised project objectives, which included the removal of a PID objective, '*Design GUI with 3 different themes to pitch to potential clients*' and replacing it with '*Add further analytical functionality for the website*' were much more suitable to the final product that the project was aspiring to become. While some objectives still seemed ambitious in their scope, they inspired the developer to keep progressing the project to meet its targets to produce the best version of the system within the given timeframe.

When first commencing the project, objective 7 specified the website as "*To provide the client with a simple online GUI to view the system and its additional features*". While the specification was relatively broad, it left room for interpretation of what the additional features and general theme could encompass. This turned into a positive outcome, as when the amended objective was refined to benefit the website, the overall project also benefitted, as it furthered the project's USP; Increasing the "wow" factor for a bar's business.

If the project was undertaken again, the developer would not use a programming language in which he had little or no experience. During the development of the web application stage, the developer had little experience with HTML and no experience in CSS and PHP. If it could have been fully developed in HTML, this stage would have been less prolonged. However, learning a new programming language is a valuable skill and will help prevent this kind of problem occurring in future.

By developing the project using flexible Agile processes, the system was split into multiple, incremental stages. As the project consisted of 3 main parts; order generation, database and website, each with its

own sub-sections and development features, naturally the project was designed using Agile methodologies. Agile methodologies were especially practical when altering the objectives of the project as mentioned earlier in this section, as the new objectives were beneficial to the end user. Also completing these smaller incremental stages allowed the developer to see the amount of progress being completed each week, as Agile methodologies state that working software is the primary measurement of progress.

Whilst the project build ended up overrunning by several weeks, all the functionality outlined at the beginning of the project was achieved. This accomplishment was largely due to the successful implementation of several key PRINCE 2 principles. By detailing an end stage procedure, where the task that had just been completed was analysed and compared against the objective that it was trying to fulfil, it was quickly shown how accurately and efficiently the completed task benefited the project. As aforementioned, switching the website to PHP was a big change, but as it was documented and controlled in the appropriate manner, as without this process the development could have continued down the wrong path with no real direction or relevance.

10.3. Objective Evaluation

At the start of the project, objectives were outlined that would allow the developer to create a stock exchange themed system that could be used within a real bar. These objectives have been evaluated below:

- The first objective was to analyse the current market to find the functionality being offered by bars with similar themes. This was to gauge the amount of core features that were required to make the system a USP for bars. These core features branched off to include bonus features which weren't essential to the project, but would still be beneficial. This objective was accomplished by splitting the workload into core and bonus features, virtually all of which were implemented into the system. This can be seen in section 9.3.1 and 9.3.2. The market research section also shows how a variety of bars were chosen to be analysed which saw multiple themes being used to show off their product.
- The second objective was to research the current technologies that make similar systems attractive to various clients. The conclusion from the market research (section 3) was that the level of functionality corresponded to how immersive the consumers experience was. Through the use of live tickers, analytics and extra "wow" factors such as a "market crash" feature that randomly offered a certain drink's price to drastically drop, businesses were able to engross the consumer into the experience and get them to 'play' the system into getting a cheap drink. These technologies were a mixture of hardware and software implemented features.
- The third objective was to review the level of profitability of the proposed system and the interest it would receive. This objective was mainly addressed with a mind-map that showed all the various revenue streams that could be used to bring extra money into the bar, given the budget of a bar. This is shown in Appendix I.

- The fourth objective was to improve the “wow” factor of a potential client’s bars to draw in more consumers. The “wow” factor that was initially designed was aimed at the consumer and the bar. The consumer would get a clear view of the drink’s price and the movement of each drink since the last interval. This is shown in appendix J. The bar would also gain a “wow” factor of additional analytics that showed the bar the profit per drink, per interval or they could see details of the drink’s price, movement and quantity over the duration that the system was being run. The system provided the bar with an analytical tool that they wouldn’t have had before.
- The fifth objective was to provide a user guide which can be seen in Appendix 14.2.
- The sixth objective was to provide an algorithm that dynamically changes the prices of drinks depending on the consumer demand. This objective took the longest to implement but was the most complex and crucial objective of the entire project. By completing this in Stage 3, the worry of this task overrunning into subsequent tasks was diminished. Another benefit of achieving this task was the opportunity to continually improve its aesthetics and functionality as the rest of the project was progressing.
- The seventh objective was similar to objective 3, but also included adding an easy GUI that the bar staff could quickly learn how to use. This was achieved by taking a minimalist approach to the amount of text on the webpages and only including what was relevant. Too much clutter on the webpages would detract from the user’s positive experience.
- Lastly, the eighth objective was to provide the client with a consistent and reliable system. This was achieved by a connection pool within the XAMPP server that handled all the requests and queries that were needed to be made and storing the returned information in arrays that could be easily inserted into the Google Charts API.

11. Conclusions

11.1. Lessons Learnt

The main lesson that has been prominent throughout this project build is learning to be flexible to combat unforeseen issues. Before the start of the project I had little experience dealing with projects on this scale and even less in combating problems that arise during development. The project taught me how to formally log the exceptions that arose and find a solution to them that would have the minimal impact on the functionality of the project whilst still meeting the core and bonus objectives laid out at the beginning.

The project also refined the skills that I felt were weak / undeveloped at the start of the build. Even though I had worked with databases and servers in previous projects, I had not embarked on a project of this nature independently. This required me spending a vast number of hours on the internet following examples and tutorials on how to write complex functions that the system needed. This led to a much deeper understanding of the technologies used and more importantly, the role they played in joining the system together as a complete entity.

Another skill that was refined over the course of the project was time management. My degree has taught me the best practices and tools to employ to ensure a project meets its goals, but implementing these acquired skills myself into a project was crucial to further my knowledge of their importance. I also learnt which practices I, as a developer, work best with. For example, performing the sprints on the early

stages of the NetBeans order generation code showed me smaller milestones that were being accomplished and boosted my motivation to progress the project.

11.2. Further Development

This project has allowed me to turn an idea that I was passionate about into a product that I am proud of. It has hugely improved my skills as a programmer and a project manager – skills that will be further improved and developed should I take this project further. If I were to do this, there are a lot of new challenges that would have to be faced. For the project to be fully functional in a real bar, it would have to be integrated into a bar's point-of-sale (POS) system, which would replace the order generation code that was designed as a substitute for this. Bars also use a range of POS systems; each one of these would require a steep learning curve if I wanted to be able to merge and implement them.

Nearing the end of the project, it was noticed that as mentioned in the PID, Section 6: Initial Project plan, increment 3, that the developer would design 3 different themes for the system to use on the web application. This was removed from the objectives, as adding functionality to the website was worth more than the aesthetics. On reflection, it wasn't beneficial to have done this, because it would have been useful when pitching to future clients. This is an area that I would consider fulfilling in the coming months after the project is officially over as it could be a useful asset.

I am aware that the application could still benefit from future work to make it a more rounded and complete project. This holds true to the fact that a project never has just one version; most applications have updates and add-ons that are released in the future.

11.3. Final Thoughts

The project has been an appropriate challenge for me as a developer. Throughout the process there were multiple issues that arose and had to be fixed to maintain forward momentum. I have improved upon the skills that I started the project with and in addition have acquired new skills such as HTML, CSS and PHP. Although I did not have a real client as the driving force behind it, I have instead produced a working demonstration of how the project could be used in a real-life scenario which could be taken to future investors and used as a selling point. The project allows consumers to have an immersive experience which differs from a normal night out at a bar or club. This alternative concept for a night out could be a strong USP for businesses looking to grow their existing customer base or create a brand-new target audience for their bar.

12. Statement of Word Count

Word Count = 10,309.

Link to OneDrive = https://liveplymouthac-my.sharepoint.com/:u:/g/personal/benjamin_shafto_students_plymouth_ac_uk/EaVAwZPDBc5BibizbzQNBPOB2UsO2cb_oiPzbt5z2JlwzA?e=VETbEc

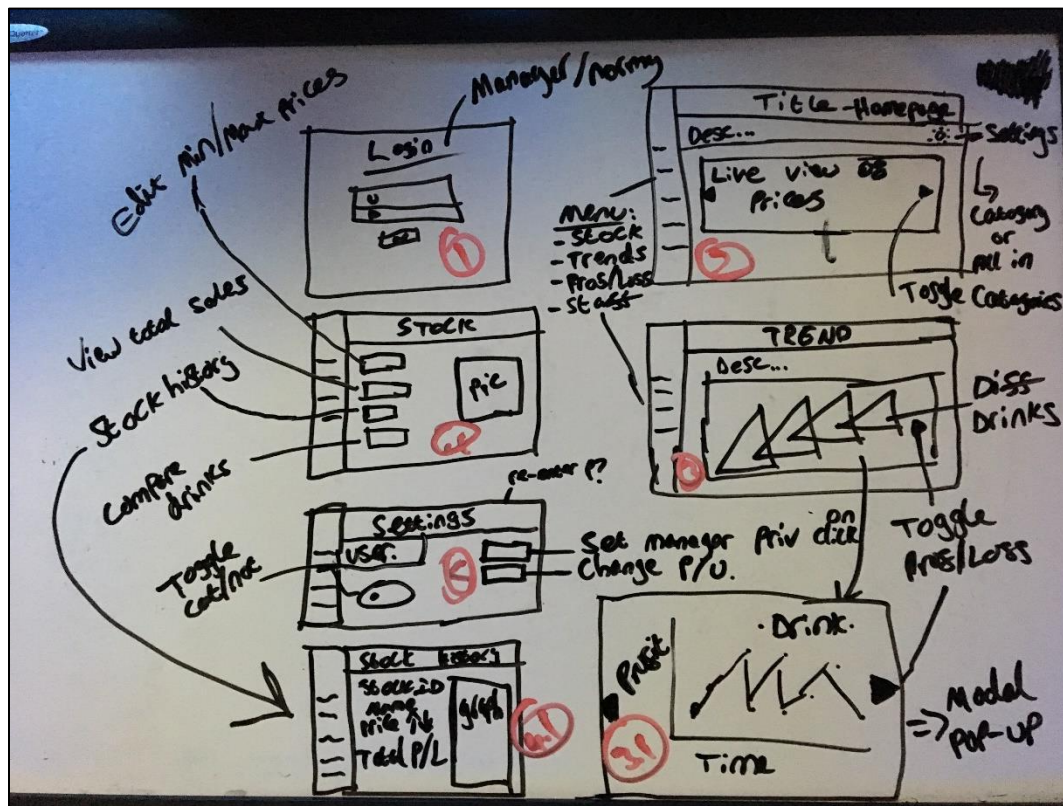
13. Reference List

- A. www.thedrinkexchange.com
- B. www.reservebarstockexchange.com
- C. <https://www.alpedhueznet.com/bars/freeride-pub-alpe-d'huez-centre-27389>

14. Appendices

14.1. Appendix A: Original Sketch of Website Functionality

This image shows a rough sketch of the functionality that the website could contain. Although this was an initial sketch, not all the functionality shown was carried through to the end of the project. For example, the login screen was dropped as there wasn't multiple users accessing the website.



14.2. Appendix B: Market Research Companies

Company	Location / Theme	Successful
ReserveBarStockExchange.com	Stock-Exchange in London	Yes
Free Ride Café – Alpe d'huez	Stock-Exchange in French Alps	Yes
TheDrinkExchange.com	Stock-Exchange online package	Yes

14.3. Appendix C: Market Research Questionnaire

The questions listed below were chosen to obtain the most information from bars that agreed to participate in discussions concerning the intricate details about their business model which they might be reluctant to disclose to an outsider.

Questions to ask bars who already have a similar system:

1. What reason led you to change to a themed bar?
2. What was your old system like?
3. Did you get a third party in to design it or did you make it yourself?
 - a. What hardware do you have with the system?
 - b. What software runs the application?
 - c. How does the system link together? Local-Host?
 - d. How does your algorithm for changing the drinks prices work? By drinks groups?
4. What inspired the design of the GUI? Was it themed?
5. Does the system draw in more customers, if so, was that an incentive to become a themed bar?
6. Do the customers prefer the themed bar to a normal working one?
7. What visual aids do you use to show the stock exchange system around the bar?
8. Do you have any extra features with the system like a "market crash" mode?
9. What feedback, if any, do you get from the system? E.g. Reports, Trends...
10. Is there anything else special about your system that hasn't been mentioned yet?

14.4. Appendix D: Live Tickers

Image taken from Reference A. The live tickers add to the immersive experience of the consumer by displaying the drinks prices and percentage movements on a small LED display, encompassing the bar.



14.5. Appendix E: Tablet Application

Image taken from Reference A. The tablet application shows the bar application running so that the manager can analyse the system live.



14.6. Appendix F: Database Research

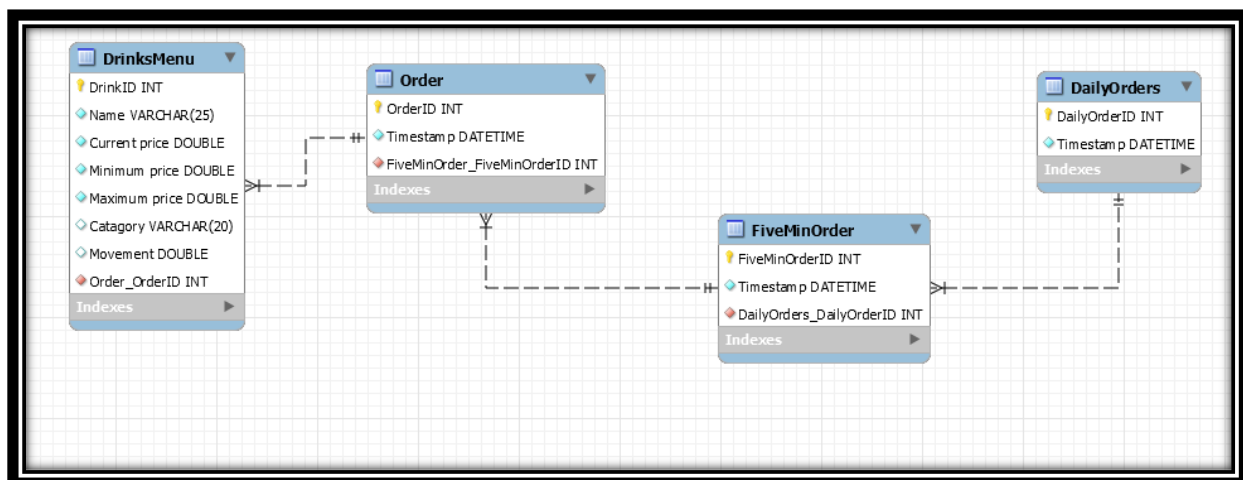
The database that was selected for the project was MYSQL via MYSQL Workbench. This application was chosen because it's a free-to-use, open-source application that facilitates effective management of databases. Some of the advantages of using MYSQL are listed below:

- **Data Security:** MYSQL is globally renown for being secure and reliable, which adheres to Section 5, objective 8 from the objectives section of this report which states that the developer will provide the end user with a consistent and reliable system.
- **Scalability:** MYSQL offers developers the ability to tiny of massive amounts of data which is flexible for growth. This feature particularly caught the eye of the developer as whilst the project is still being developed and is not yet ready to be released, if the need for more data storage is needed. The database provider won't have to change.
- **Cost:** The cost of renting a database within other applications can be expensive. As MYSQL is free, there is no initial or running costs. This means that if the project was to be left alone for a few months or years, the developer could come back to it without occurring any additional costs.

Other databases that were researched for the project were SQLite and Oracle. Oracle had many advantages such as the east import and export of schemas to the vast range of tools available with reviews boasting excellent customer support. The reason this database wasn't chosen was the cost of running it. SQLite was also researched and compared against its rivals and has may benefits such as storing big data, data analysis and storing file archives which could be beneficial to the project and was almost the first choice over MYSQL.

14.7. Appendix G: Database ERD

Below is a simple ERD for the database. As it stands there are currently 4 tables that store the order generation information being sent from the NetBeans application. This may be subject to change if further work is completed on the project at a later date.



The connections are as follows:

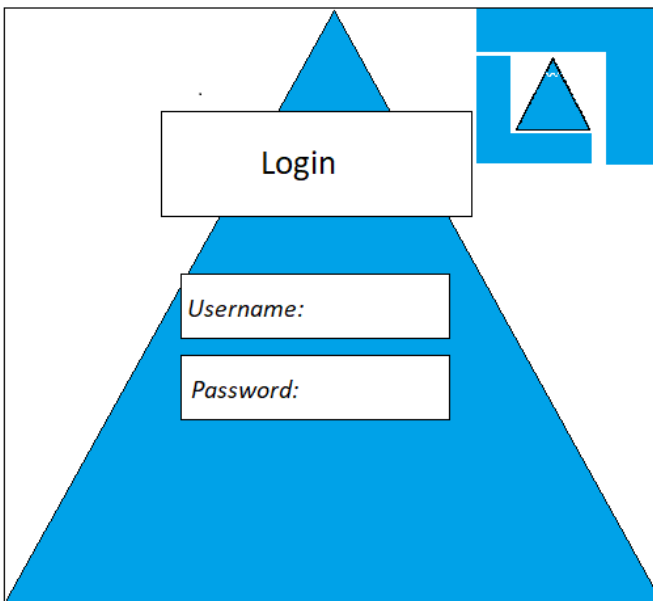
- There is a one to many relationship between the DailyOrders table and the FiveMinOrder table. This is because there are multiple 5-minute intervals that can be stored within a day.

- There is a one to many relationship between the FiveMinOrders table and the Order table. This is because there are multiple orders within a 5-minute order period.
- There is a one to many relationship between the Order table and the DrinksMenu table. This is because there are a random number of drinks between 1 and 8 that can be added to a single order.

14.8. Appendix H: Detailed Website Design

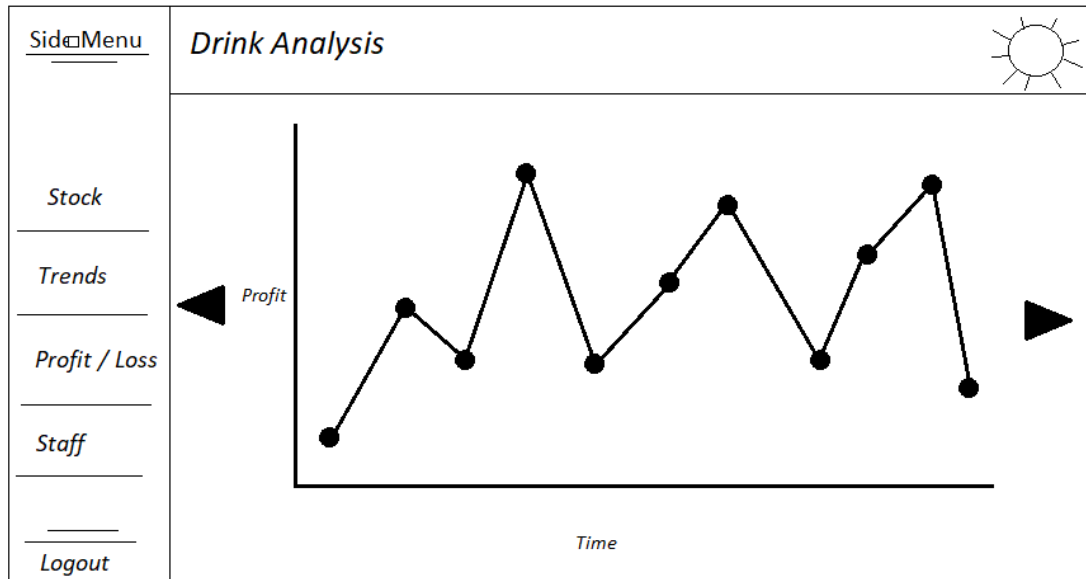
14.8.1. Login Screen

The initial design for the login screen with a logo in the top right-hand corner. This wasn't implemented into the project as the need for multiple users wasn't necessary.



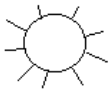
14.8.2. Drink Analysis

The drink analysis section was designed to show the details of drinks over a period of time with the profit levels being measures on the y-axis. The side-menu on the left-hand side shows the different pages that the user could navigate between.



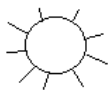
14.8.3. Main Menu

The main menu design is the first page the user sees when opening the project. The notes section would act as a calendar that the bar could possibly sync with their Google calendar. The main menu was also initially designed to have dynamic images automatically transitioning.

<u>Side Menu</u>	<div><div></div><div><h2>Main Menu</h2></div></div> <div><p><i>Description of stock levels, trends, profit/loss etc...</i></p><div><div>◀</div><div><p><i>Rolling screen of current stock prices being displayed on the tv outputs</i></p></div><div>▶</div></div></div> <div><p><i>Notes / Pinboard of important dates</i></p></div>
Stock	
Trends	
Profit / Loss	
Staff	
Logout	

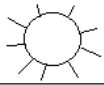
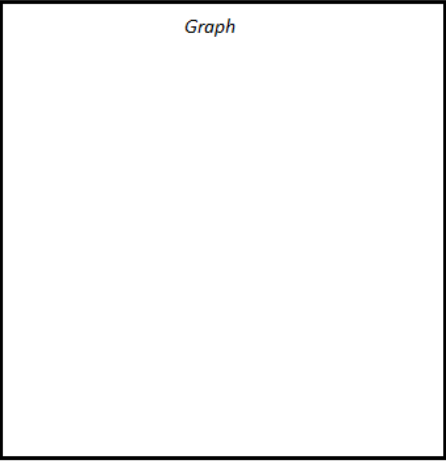
14.8.4. Settings

The initial design for the settings page was brief. This page wasn't implemented into the system as again, the need for multiple users was unnecessary.

<u>Side Menu</u>	<div><div></div><div><h2>Settings</h2></div></div> <div><div>User:</div><div><div>Change staff privileges</div><div>Change Username / Password</div></div></div>
Stock	
Trends	
Profit / Loss	
Staff	
Logout	

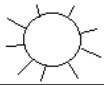

14.8.5. Stock History

The stock history page was designed for the user to save time checking stock levels on a separate application. The user would simply select the stock name or ID of their choice and be able to see the corresponding details of that stock item. The graph on the right-hand side was going to show details of that particular drink from the last time the system was ran.

<u>Side Menu</u> <i>Stock</i> <hr/> <i>Trends</i> <hr/> <i>Profit / Loss</i> <hr/> <i>Staff</i> <hr/> <hr/> <i>Logout</i>	<i>Stock History</i> 	
	<input type="text" value="Stock ID"/>	<div>Graph</div> 
	<input type="text" value="Stock Name"/>	
	<input type="text" value="Stock Price"/>	
	<input type="text" value="Overall price movement (up / down %)"/>	
	<input type="text" value="Total Profit / Loss on stock item"/>	

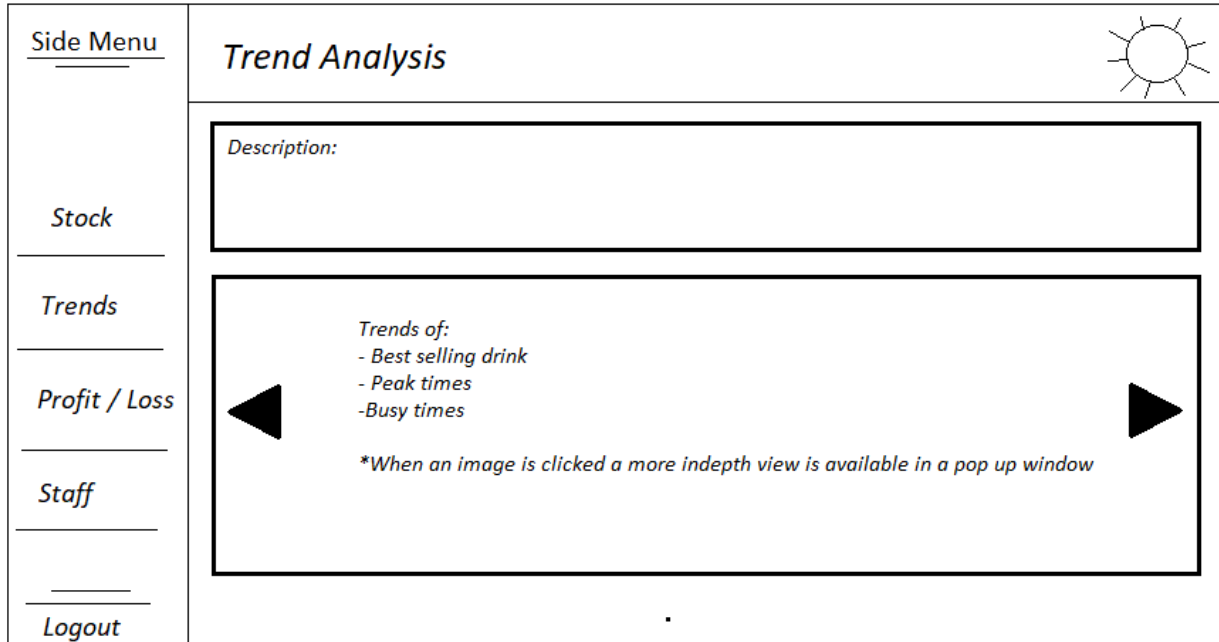
14.8.6. Stock Page

The stock page's design was similar to Stock History. This page however, allowed the user to edit the stocks details and save the user time by not having to use a separate application. This wasn't implemented into the database as the functionality proved to be too time consuming.

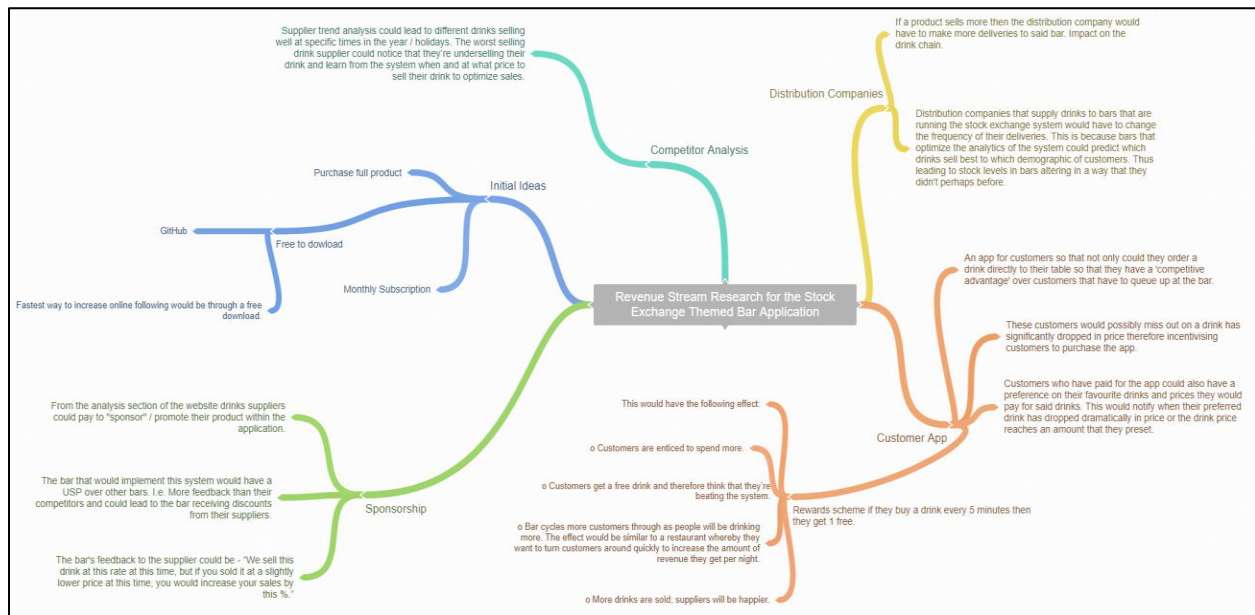
<u>Side Menu</u> <i>Stock</i> <hr/> <i>Trends</i> <hr/> <i>Profit / Loss</i> <hr/> <i>Staff</i> <hr/> <hr/> <i>Logout</i>	<i>Stock Page</i> 				
	<input type="text" value="Edit min/max values"/>	<div>Top 5 best selling drinks this week</div> 			
	<input type="text" value="View sales"/>				
	<input type="text" value="Stock history"/>				
	<div>Compare drinks</div> <table border="1"> <tr> <td>Vodka</td> <td>Gin</td> </tr> </table>		Vodka	Gin	
	Vodka		Gin		

14.8.7. Trend Analysis

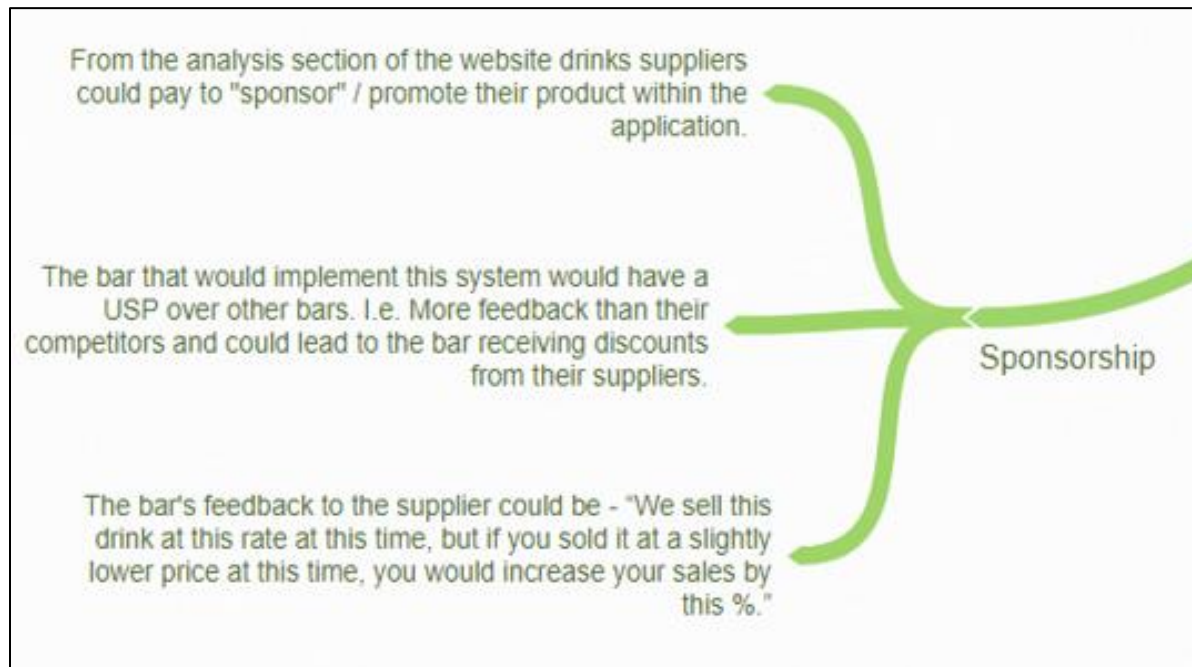
The trend analysis page was the main focus of the website and the design of this page is similar to the produced version. The trends of the page changed to this initial design but proved to be more beneficial for the end user.



14.9. Appendix I: Revenue Streams



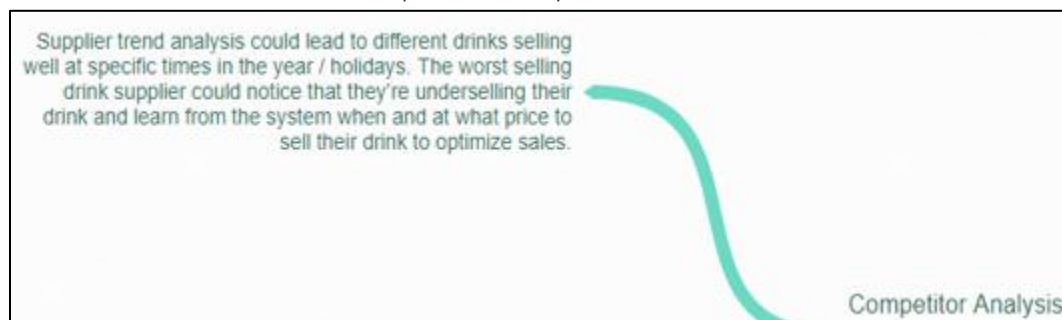
14.9.1 Revenue Streams: Sponsorship



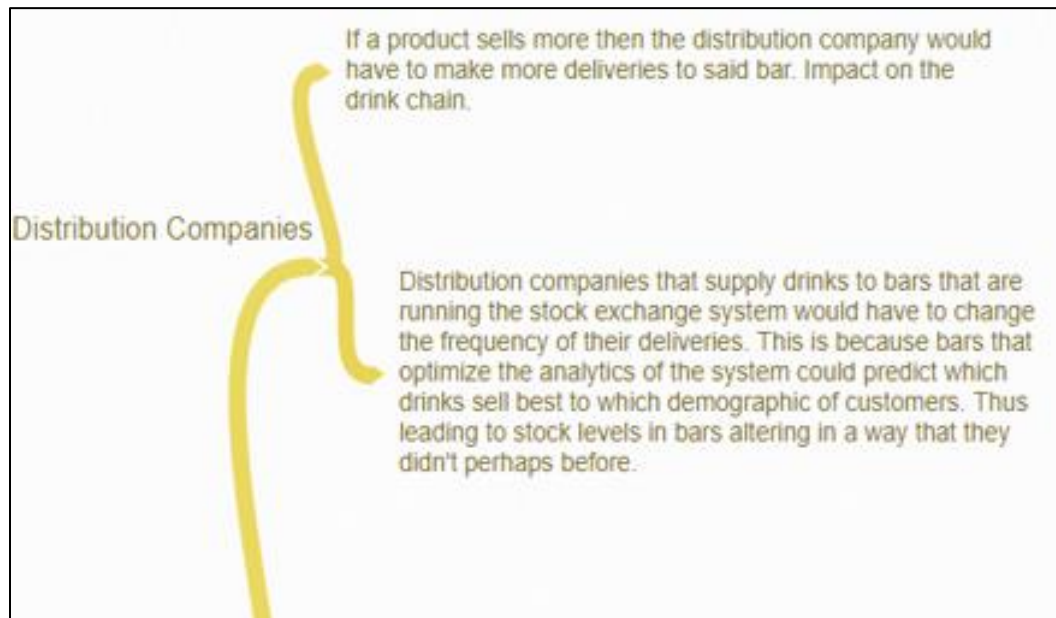
14.9.2. Revenue Streams: Initial Ideas



14.9.3. Revenue Streams: Competitor Analysis



14.9.4. Revenue Streams: Distribution Companies



14.9.5. Revenue Streams: Competitor Analysis



14.10. Appendix J: Website Screenshots

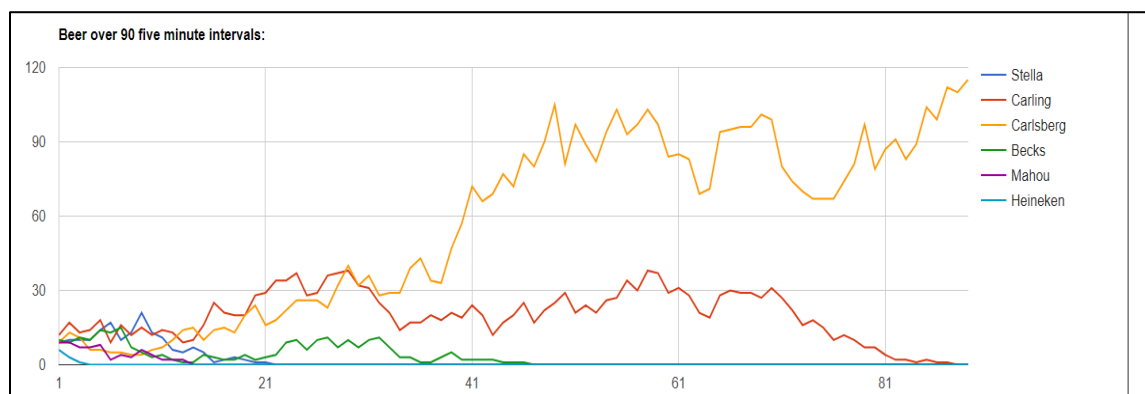
14.10.1. Live View

This screenshot shows the live view of all the drinks that exists within the mock bar system.

Beer			Cider			Spirit-Mixers		
Stella	£4.85	-3.00%	Strongbow	£4.85	-3.00%	Vodka Coke	£5.16	+3.20%
Carling	£5.19	+3.80%	Strongbow Dark Fruits	£5.16	+3.20%	Gin Tonic	£5.16	+3.20%
Carlsberg	£4.85	-3.00%	Bulmers	£5.16	+3.20%	Rum Coke	£4.77	-4.60%
Becks	£5.16	+3.20%	Magners	£4.85	-3.00%	Spiced Rum Coke	£5.24	+4.80%
Mahou	£4.85	-3.00%	Old Mout	£4.79	-4.20%	Gin Lemonade	£5.15	+3.00%
Heineken	£4.82	-3.60%				Malibu Coke	£5.16	+3.20%
Spirits			Cocktails			Wine		
Tequila	£4.82	-3.60%	Sex on the beach	£5.23	+4.60%	Chardonnay	£4.84	-3.20%
Jägerbomb	£5.18	+3.60%	Mojito	£5.21	+4.20%	Rioja	£4.82	-3.60%
Sambuca	£5.15	+3.00%	Long Island Iced Tea	£5.18	+3.60%	Rosé	£4.81	-3.80%
Vodka	£5.16	+3.20%	Martini	£4.84	-3.20%	Merlot	£4.84	-3.20%
Fireball	£4.84	-3.20%	Daiquiri	£4.84	-3.20%	Sauvignon Blanc	£4.76	-4.80%
Dark Rum	£4.84	-3.20%	Old Fashioned	£4.84	-3.20%	Pinot Noir	£5.18	+3.60%

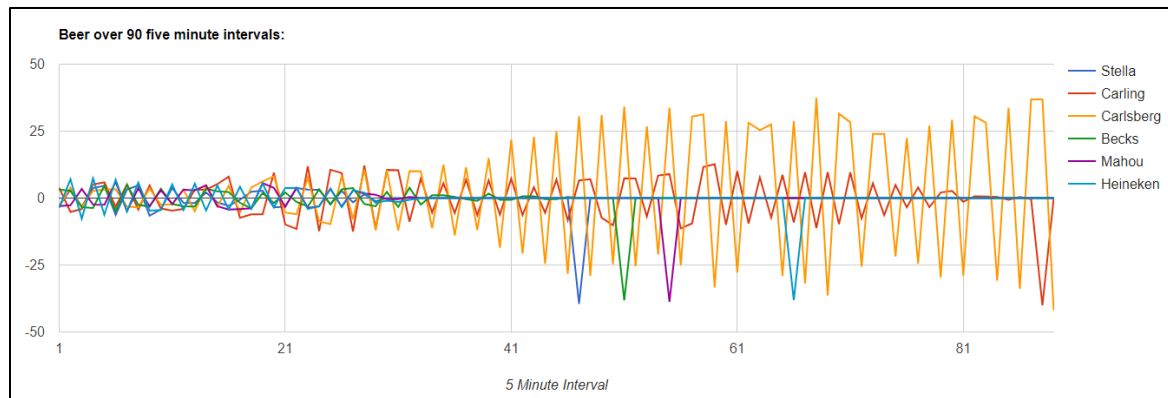
14.10.2. Net Profit

The Google Chart below shows the net profit for the Beer category selected from a combo box over the first 90 five-minute intervals. Cider, Cocktails, Shots, Spirit / Mixers and Wine are also available to view. The graph calculates the net profit by reading in the price the drink is selling for at a specific interval and multiplies it by the quantity its bought.



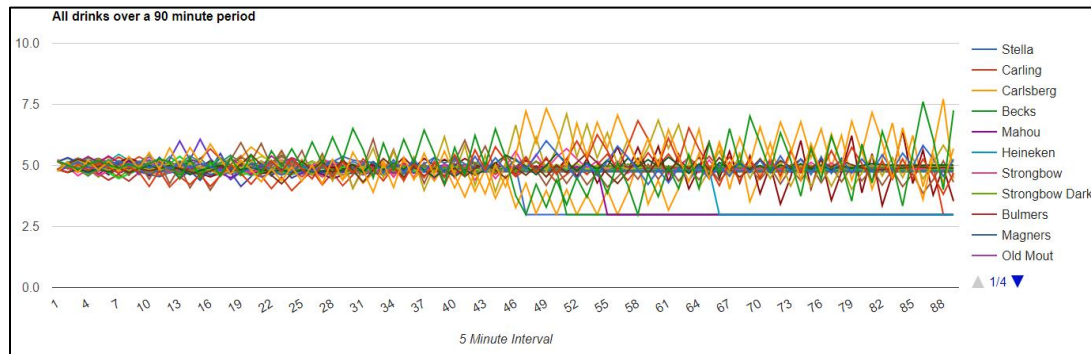
14.10.3. Category Comparison

The category shown below in the Google Chart is Beer. The user can choose to select to view the price, movement percentage or quantity for each category of drink. The chart is updated each time the user selects a different output or new data is received from the database from the connection pool.



14.10.4. Overall Comparison

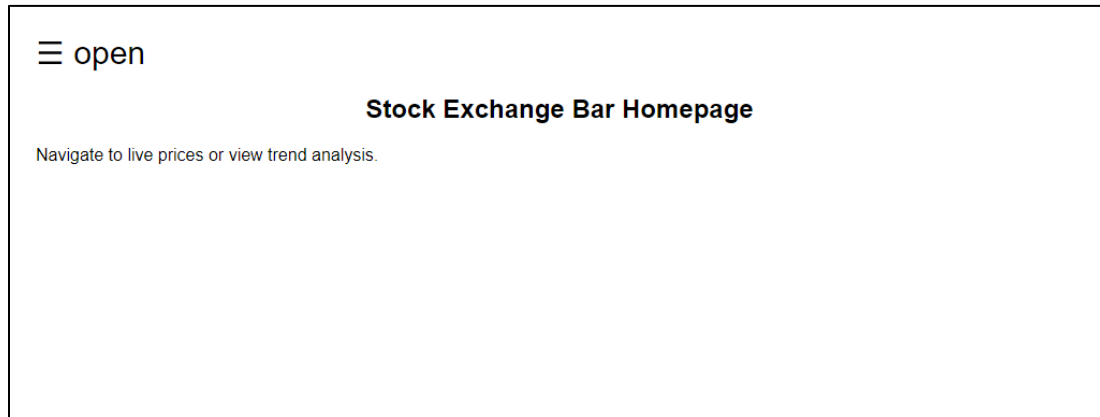
The Google Chart below shows every single drink price in the database being displayed. The legend on the right-hand side of the graph is scrollable and shows the colour associated with each drink. At any interval the user can hover the mouse over the graph and see the value of the drink price.



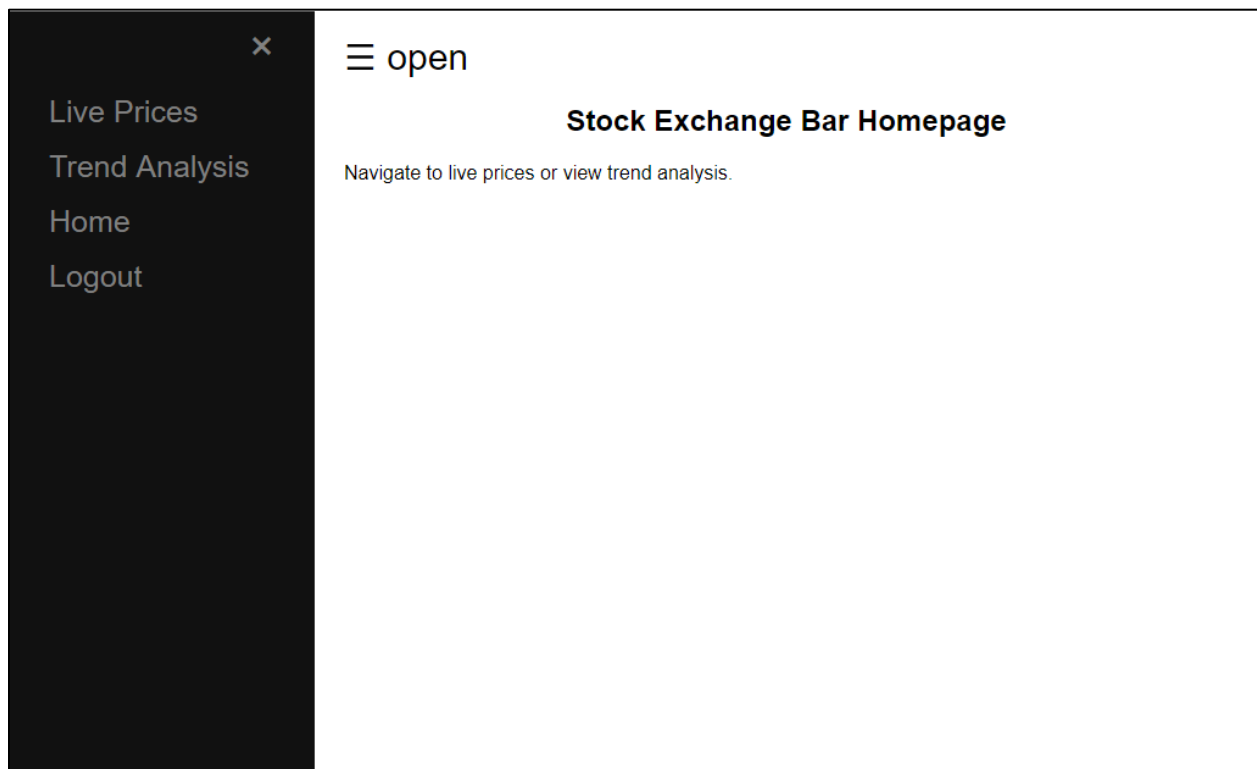
14.11. User Guide

The only section of the project that the user will access is the website. Below is its user guide:

The first page the user is welcomed to the is the main menu. The original design for this was more detailed but the functionality on the trend analysis had a higher priority. The webpage can be accessed through the URL <http://localhost/test/Home.php>



From this page the user can open the side menu on the left-hand side that animates open showing the other pages.



The below image shows the Live View page which would be displayed on various TV displays around the bar. The functionality included shows the drinks current price and movement percentage. Red shows the price has increased by that percentage and green shows the percentage decrease.

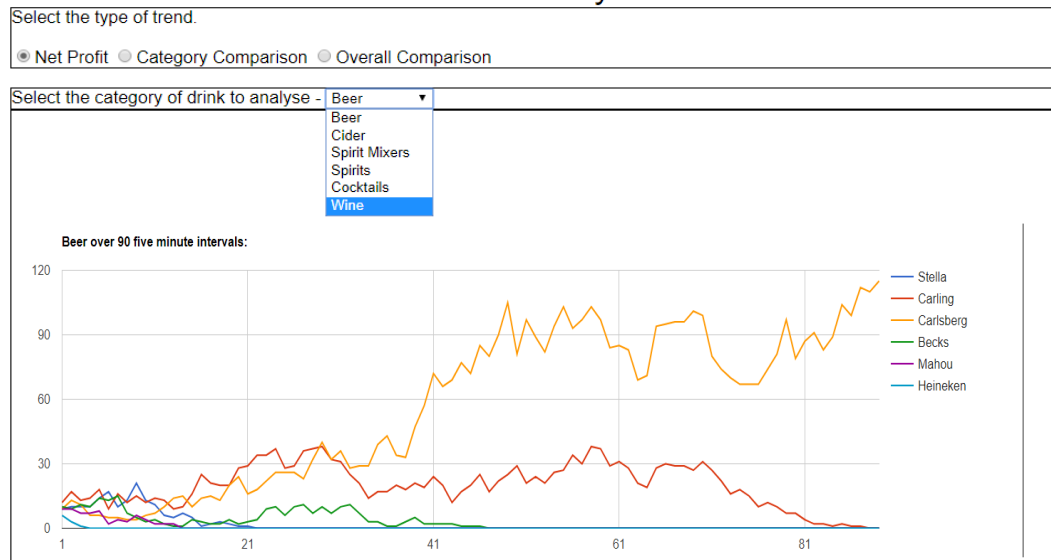
Beer			Cider			Spirit-Mixers		
Stella	£4.85	-3.00%	Strongbow	£4.85	-3.00%	Vodka Coke	£5.16	+3.20%
Carling	£5.19	+3.80%	Strongbow Dark Fruits	£5.16	+3.20%	Gin Tonic	£5.16	+3.20%
Carlsberg	£4.85	-3.00%	Bulmers	£5.16	+3.20%	Rum Coke	£4.77	-4.60%
Becks	£5.16	+3.20%	Magners	£4.85	-3.00%	Spiced Rum Coke	£5.24	+4.80%
Mahou	£4.85	-3.00%	Old Mout	£4.79	-4.20%	Gin Lemonade	£5.15	+3.00%
Heineken	£4.82	-3.60%				Malibu Coke	£5.16	+3.20%
Spirits			Cocktails			Wine		
Tequila	£4.82	-3.60%	Sex on the beach	£5.23	+4.60%	Chardonnay	£4.84	-3.20%
Jägerbomb	£5.18	+3.60%	Mojito	£5.21	+4.20%	Rioja	£4.82	-3.60%
Sambuca	£5.15	+3.00%	Long Island Iced Tea	£5.18	+3.60%	Rosé	£4.81	-3.80%
Vodka	£5.16	+3.20%	Martini	£4.84	-3.20%	Merlot	£4.84	-3.20%
Fireball	£4.84	-3.20%	Daiquiri	£4.84	-3.20%	Sauvignon Blanc	£4.76	-4.80%
Dark Rum	£4.84	-3.20%	Old Fashioned	£4.84	-3.20%	Pinot Noir	£5.18	+3.60%

The user can then use the webpages 'back' button to navigate back to the previous page or type in the URL of another page.

The below screenshot shows the Net Profit page and the Google Chart displaying the data.

≡ Menu

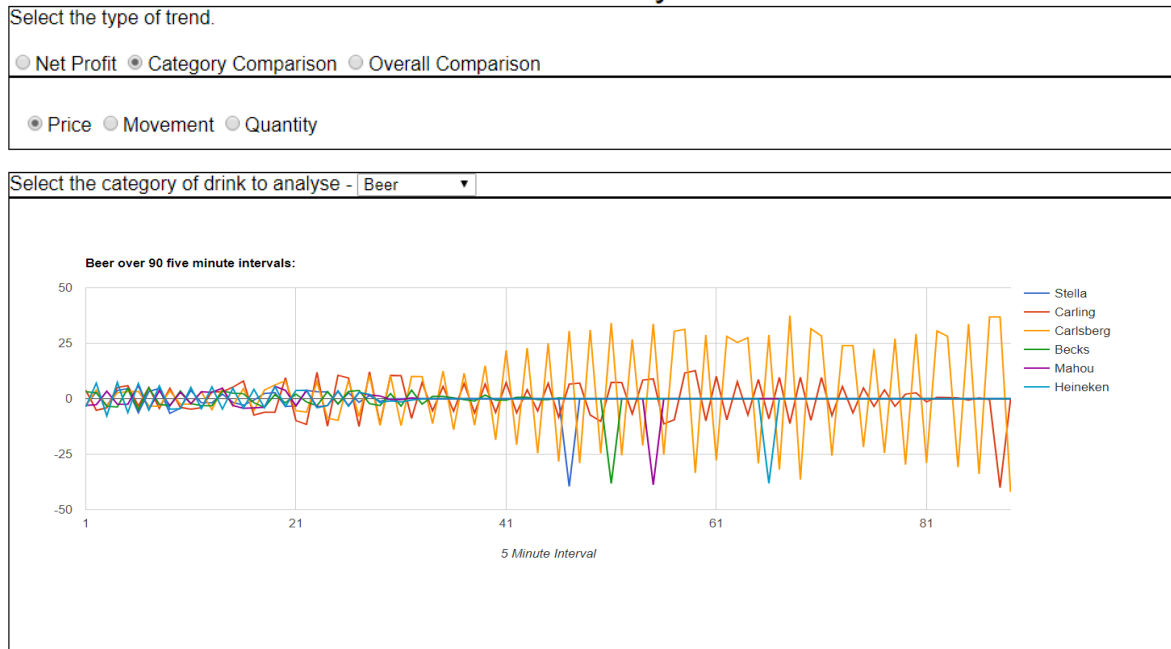
Trend Analysis



The next section of this webpage is the category comparison which give the user the tools to view the price, percentage movement or quantity of each drink category. The below screenshot shows the percentage movement for the Beer category.

≡ Menu

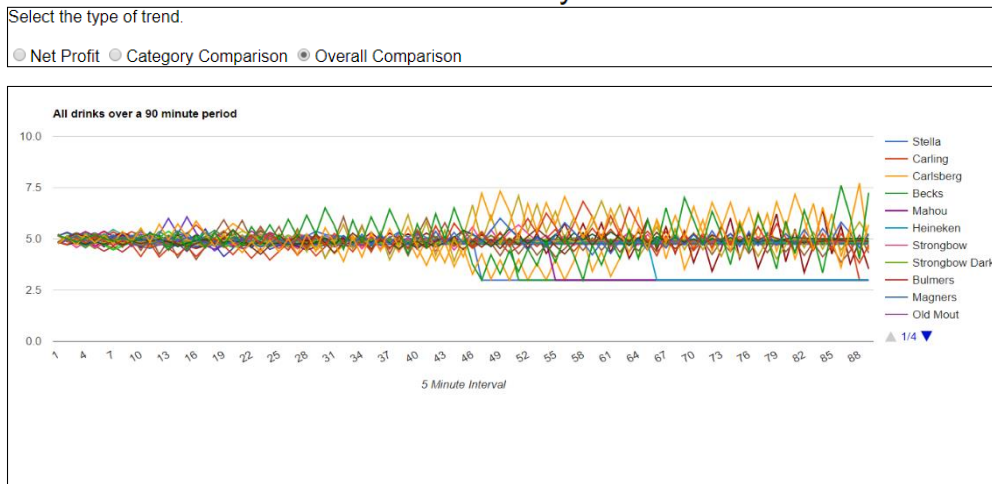
Trend Analysis



The final section of the webpage is the overall comparison of the drinks. This allows the user to see all of the data on one Google Chart. The side legend is scrollable and shows unique colours for each drink. The user also has the functionality of being able to hover the mouse of any interval being shown to get the exact price of the drink at that specific interval.

≡ Menu

Trend Analysis



14.12. Project Management Artefacts:

14.12.1 PID

Contents

1. Introduction – who are the potential clients?	3
2. Business Case – client’s possible business needs / benefits	3
3. Project Objectives	3
4. Initial Scope	4
5. Method of Approach	4
6. Initial Project Plan	5
7. Initial Risk List	6
8. Initial Quality Plan	6
9. Legal, Social, Ethical issues	7
a. Clients requirements	3
b. Benefits for the client	3
a. Control Plan	5
b. Communication Plan	5
c. Simulation Plan	6

1. Introduction

The project that I am proposing to design and market is a 'stock-exchange' themed bar system (BSES) that can be targeted at any bar / nightclub. The project would be designed so that a business's stock and point of sale (POS) system would be incorporated into the new project, but in addition to their old system the project would introduce an algorithmic GUI. This would show the drinks prices varying depending on the consumer demand and could be further tweaked by the client to maximise profits and specifically tailor it to their consumers. These potential clients would also be able to view the project running live on a web server that would report stock levels, profits and trends.

2. Business Case

a. Clients requirements

A client's requirements would be that the proposed projects system works in conjunction with the clients existing system. As the client would already have a current bar/stock system, it's important for my new project to incorporate features that work in conjunction with the old system. The client would also want the new system to be easy to learn and train managers to fully utilise the outputs of the system. For example, the emerging trends that would accumulate after the BSES has been used. The client would also require the investment into the BSES to bring in extra business into the bar and increase revenue made from selling the same stock. Finally, the client would also expect the system to be secure and reliable so that the overall operation of the bar or nightclub wouldn't be affected if the system was to malfunction.

b. Benefits for the client

The main benefit that the client would be receiving is the unique selling point it would gain over its competitors by offering a service that is rarely seen in the bar / nightclub environment. This project would offer a client the ability to draw in new customers from their competitors. It would also encourage customers to spend more on drinks because the BSES incentivises customers to buy a drink when its price is lower than it would normally be. However, because the drinks are always fluctuating the consumers never know when a they're really getting a good bargain. The client would also benefit from selling stock that wouldn't normally be being turned over very quickly. This saves money on not having any wasted stock as every item of stock would have the opportunity to become the cheapest selling drink at one point in the night. Lastly, a benefit that could have other ethical issues is that customers could get competitive about finding the best cheap drink deal, so they would consume more drinks than they would normally do on a night out to feel like they've 'beaten the system'.

3. Project Objectives

1. To research into the current market to find the level of service being done by other themed bars.
2. To research the current technologies that make other similar systems stand to attract various clients from different locations.
3. To review the current POS and stock systems already working at potential client's bars.
4. To improve the 'Wow' factor of potential client's bars to draw in more consumers.
5. Provide a user manual.

6. To produce an algorithm that dynamically changes the prices of drinks depending on consumer expenditure.
7. Provide the client with a simple GUI to view the system.
8. To provide the client with a secure and reliable system.

4. Initial Scope

1. Researching the current target market would include areas such as:
 - a. Current software being used.
 - b. GUI aesthetics.
 - c. Outputs. E.g. Reports, trend detection, automated stock orders.
 - d. Level of adoption of bar themed products in industry.
2. Current technologies that would be researched to bring the product up to the market standard:
 - a. Website hosting / local-servers.
 - b. App development.
 - c. Database hosting.
 - d. Visual elements.
3. The Analysis of current POS and stock systems will be done by interviewing both local and off-shore bars to gain an understanding into their scale and setup. This could also include the other licencing services that they pay for and other experiences they've had with similar proposed projects.
4. Improving a potential client's 'Wow' factor would involve interviewing local and off-shore potential clients and ask them what they think their brand is lacking or could benefit from.
5. The user manual will demonstrate all the systems functionality and how to fully utilise it.
6. The algorithm will be researched and compared to other similar algorithms on the market. Once an optimal algorithm has been designed. Trial and error will fully optimize it.
7. The GUI will be designed to tailor to individual clients and so each one will have a similar underlying principles, but aesthetically they will differ.
8. A secure and reliable system will be achieved as the client's current POS and stock management system will be left mostly unchanged. The BSES algorithm will calculate the next 5 minutes' prices and feed that back into the client's POS system. Using a local-host server will also mean no data is sent online.

5. Method of Approach

Software development will consist of an agile, Kanban style approach. This method was chosen as it helps identify where processes need improvement or more focus to meet the objectives deadline. Kanban will also help prevent the project from bottlenecking. This The main stages of the project can be compressed into the following stages: (i) Research; (ii) Software design & development; (iii) Hardware implementation; (iv) Testing and evaluation.

The possible technologies that I would implement with are: a Netbeans IDE, a SQL local-server, a database host like Oracle which is connected to the server with MySQL. This section will be further clarified by the research section of the project.

6. Initial Project Plan

Initial Project Plan			
Stage	Expected Start Date	Expected Completion Date	Products / Deliverables / Outcomes
Initiation	26th Jan	11th Jan	PID
Research and requirements outline	29th Jan	5th Jan	Research of current market competitor systems, outline core / bonus requirements, research current technologies in the market.
Initial High Level Design	5th Jan	12th Jan	Design documents (outline user manual, algorithm pseudo code, hardware functionality, mock templates of various themes)
Increment 1	13th Jan	1st Mar	Implement software and algorithm with dummy data – test.
Increment 2	2nd Mar	14th Mar	Implement researched hardware and algorithm with software – test.
Increment 3	15th Mar	23rd Mar	Design GUI with 3 different themes to pitch to potential clients.
Easter Vacation	23rd Mar	13th Apr	
System and User acceptance testing	16th Apr	20th Apr	Finalise system, test product of potential clients, optimize user training manual
Assemble & compete final report	23rd Apr	4th May	PRCO204 Report

a. Control Plan

The goals of the control plan are to support the processing and traceability of changes to a series of factors. To help me do this I will be using the following PRINCE 2 features:

- Learn from experience. Due to my previous experience with Android development, I have learnt the basic format of how to design and develop which will save me time when developing the BSES.
- Managing by stages. As my task has 3 main ‘increments’ I will be breaking those large sections down into smaller, more manageable chunks that will all be measurable.
- Focus on products. The details of the product from the research section will be made clear so that only relevant work is done on the project as to not waste time on unnecessary features.

Having incremental meetings with my project supervisor as dictated by the PRCO304 module will also ensure my targets are deemed realistic. There are also the below plans that add control variables to the project.

b. Communication Plan

The communication plan will be controlled meetings in line with the control plan. These could be arranged weekly or further ad-hoc meetings could occur when applicable. Potential clients may appear

towards the end of the project that would include lines of communication, but as for now no concrete client exists.

c. Simulation Plan

The simulation is for when the project is at a stage whereby the full features of the project need to be trialled and tested in a separate environment to which they were created. A local bar could be approached and asked if they would mind trialling the system free of charge. This would allow the reports and algorithmic charts to be tested and review what useful data can be predicted.

7. Initial Risk List

Risk	Management Strategy
Schedule overrun	Contingency has been introduced into the project plan over the Easter break. Highlight reports from objectives met and weekly meetings will provide a regular monitoring schedule. An exception plan will be developed, and approved by the project supervisor, in the event of more than 1 week's slippage.
Learning / Development issues with technologies	A core system will be designed to minimise extra workload to ensure that a basis system, if nothing else, can be developed.
Requirements breakdown	If the proposed requirements doesn't meet the PRCO304 brief then extra functionality like the app will be introduced to increase the difficulty of the project.
Technology Failure	System backups will be taken daily. Problems with hardware will be dependent on third parties and contingency planning will allow any temporary lapses in technology to be amended.

8. Initial Quality Plan

Quality Check	Strategy
Requirements	Requirements will be validated to make sure they are correct, relevant, achievable and measurable. This ensures no time is wasted on objectives that are not relevant. The requirements will also outline the product quality for the GUI and the user manual. A walkthrough of the system will be done on a separate system.
Hardware validation	During Stage 2 of the initial project plan, different variations of hardware setups will be documented and reviewed to configure the best setup for the scale of my project.
Design Validation	The design will be researched and split into specific deliverables. With Database normalisation, OO programming styles and Kanban development.
Algorithm Development	Research into the current market will show different ways of design. These will be compared and their core elements will be drawn out to optimise my own algorithm.
System V&V and user acceptance	To be conducted in accordance with Stage 7 of the initial project plan.

9. Legal, Social and Ethical Issues

During this stage the legal, ethical and social issues were taken into perspective. An ethical issue that was noticed was the gambling affect that arises when consumers would continue to purchase drinks at the bar. A person who could be categorized as a gambler would be at risk at spending more money than they should, which could have further consequences to the client. This could also affect the legal issue of licencing the final product to clients. During Stage 2 of the initial project plan, research will be undertaken into the different types of licences other similar products are selling as well as other avenues. This could be an issue to the lack of experience that I have in the legality of a product; however, the research and PRCO304 project supervisor will combat this issue.

14.12.2 Highlights

14.12.2.1. Highlight 1

PRCO304: Highlight Report 1	
Name: Benjamin Shafto	
Date: 09/02/2018	
Review of work undertaken <ul style="list-style-type: none"> • Research and requirements outline. • Initial method of implementation laid out. • Companies with similar systems contacted to conduct further market research. • Questionnaire completed ready for market research. • Mind map for the system completed. 	
Plan of work for the next week <ul style="list-style-type: none"> • Conclude market research. • Organise a definitive timetable for ALL activities to be carried out within the project. • Organise hardware together. • Pseudo code the test data to be entered into the algorithm. 	
Date(s) of supervisory meeting(s) since last Highlight:	07/02/2018
Brief notes from supervisory meeting(s) since last Highlight <ul style="list-style-type: none"> • Organise architecture layout for the project from a real-world scenario and a demo point of view. 	

14.12.2.2. Highlight 2

PRCO304: Highlight Report	
Name: Benjamin Shafto	
Date: 13/02/2018	
Review of work undertaken <ul style="list-style-type: none"> • Conclusion of market research (pending 1 more contact). • Pseudo code for the order generating code and the drinks algorithm. • Mock website design and functionality. • Mind map of product and brand. 	
Plan of work for the next week <ul style="list-style-type: none"> • Organise a definitive timetable for ALL activities to be carried out within the project. • Organise hardware together. • Implement software and algorithm with dummy data – test. • Outline user manual. • 	
Date(s) of supervisory meeting(s) since last Highlight:	09/02/2018
Brief notes from supervisory meeting(s) since last Highlight <ul style="list-style-type: none"> • Organise architecture layout for the project from a real-world scenario and a demo point of view. 	

14.12.2.3. Highlight 3

PRCO304: Highlight Report
Name: Benjamin Shafto
Date: 20/02/2018
Review of work undertaken <ul style="list-style-type: none"> • Organised architecture layout for a demonstration of the project. • Written the PRCO304 document up to the current date. • Started the implementation of the order generation code with test data.
Plan of work for the next week <ul style="list-style-type: none"> • Finish the order generation code and test thoroughly. • Start the algorithm code and link it with the order generation code with appropriate test data.
Date(s) of supervisory meeting(s) since last Highlight: 07/02/2018
Brief notes from supervisory meeting(s) since last Highlight <ul style="list-style-type: none"> • Organise architecture layout for the project from a real world scenario and a demo point of view.

14.12.2.4. Highlight 4

PRCO304: Highlight Report**Name:** Benjamin Shafto**Date:** 27/02/2018**Review of work undertaken**

- Finished the order generation code and test thoroughly.
- Started the algorithm code.

Plan of work for the next week

- Finish the algorithm code and link it with the order generation code with appropriate test data.

Date(s) of supervisory meeting(s) since last Highlight:

20/02/2018

Brief notes from supervisory meeting(s) since last Highlight

- Organise architecture layout for the project from a real world scenario and a demo point of view.

14.12.2.5. Highlight 5

PRCO304: Highlight Report**Name:** Benjamin Shafto**Date:** 09/02/2018**Review of work undertaken**

- Finished the order generation code and test thoroughly.
- Finished algorithm code

Plan of work for the next week

- Error and validation testing/

Date(s) of supervisory meeting(s) since last Highlight:

27/02/2018

Brief notes from supervisory meeting(s) since last Highlight

- Organise architecture layout for the project from a real world scenario and a demo point of view.

14.12.2.6. Highlight 6

PRCO304: Highlight Report**Name:** Benjamin Shafto**Date:** 15/02/2018**Review of work undertaken**

- Error and validation testing complete.

Plan of work for the next week

- Create host for database server and link to netbeans application.

Date(s) of supervisory meeting(s) since last Highlight:

27/02/2018

Brief notes from supervisory meeting(s) since last Highlight

- Organise architecture layout for the project from a real world scenario and a demo point of view.

14.12.2.7. Highlight 7

The work for this highlight was merged with highlight 6 due to an oversight in project management. This was explained in Section 14.6.

14.12.3. Exception Report: Stage 6

Project Name: Stock Exchange Themed Bar Application**Date:** 29/03/2018**Author:** Benjamin Shafto**Owner:** Benjamin Shafto

This document is subject to change. This document has been created in anticipation for the stage not being completed in time. At this current stage, this is not the case but is highly probable. There is still half a week left for me to make changes to the web application. This document is being created if the web application features are not completed in time.

Revision History

Revision Date	Previous Revision Date	Summary of Changes
29/03/2018	None	Initiation of the Exception Report
6/04/2018	29/04/2018	Web Application changed from a HTML to a PHP project. Dynamic webpage functionality completed.

Purpose

This Exception Report has been created in anticipation for a Stage exceeding its threshold for tolerance. The document has been provided by the Project Manager / Developer to offer recommendations of how to proceed this stage in the instance that the threshold is exceeded.

Initial Exception Report

Title

- Dynamic webpage implementation

Cause

- HTML doesn't support dynamic web pages.

Consequences

- Large proportion of the past week has been spent working on a web application that can't be used. Learning and implementing PHP and subsequent languages will incur further time penalties which has the potential to delay the project.

Options

- Revert to HTML. Change the functionality of the web page. However, this would severely affect the project's functionality as dynamic webpages are part of the core requirements.
- Learn from the situation, make sure future tasks that support core requirements are thoroughly researched before attempting to develop them to ensure that this issue doesn't repeat itself.

Recommendation

- Aim to find a work-around for the HTML pages, possibility there is another method that could be attempted within the allocated time period?
- Don't rush the development of the web application. Multiple core features follow the completion of this task and a contingency plan will allow this stage to be completed to the level required.

Lessons

- Ensure each stage is researched fully to check the software anticipated for use can handle the functionality required of it.
- This was an issue, but it was identified before it had a seriously adverse effect on the project. Whilst this could result in an overrun for this stage, the knock-on effect should be minimal due to recognising the problem early on.

14.12.4. Resources Used

Resource	Information	Resource Type
Sarah Shafto	sarahshafto@gmail.com	Editor
Chris Johnson	C.Johnson@plymouth.co.uk	Project Supervisor
Nigel Barlow	Nigel.Barlow@plymouth.ac.uk	Personal Tutor
Google Charts API	https://developers.google.com/chart.com	Analyse Tool
XAMPP	https://apachefriends.org	Hosting
NetBeans IDE 8.2	https://netbeans.org/downloads/	Development Software
MYSQL Workbench version 6.3	https://www.mysql.com/products/workbench/	Development Software
MYSQL local-host	https://dev.mysql.com/doc/refman/8.0/en/connecting.html	Supporting Documentation
Google Chrome	www.google.com/chrome/	Testing web functionality
Microsoft Excel	https://www.office.com/	Software for creating supporting documentation
Microsoft Word	https://www.office.com/	Software for creating supporting documentation
Stack Overflow	https://stackoverflow.com/	Supporting documentation

14.12.5. Installation Guide

- Using the link for NetBeans IDE 8.2 in the 'Resources Used' section above download NetBeans with all the bundles included.
- Using the link for MYSQL Workbench in the 'Resources Used' section, download the program in version 6.3.
- Unzip the packaged bundle called 'finalProject'.
- Once NetBeans is installed click 'open project' and navigate to the folder labelled 'NetBeans Files' and add the 'orderGeneration2' package into NetBeans.
- The 'orderGeneration' package requires a connector to link to the database. Find the file in the 'finalProject' folder called 'mysql-connector-java-5.1.46' and resolve the issues by adding the following connector's.
 - mysql-connector-java-5.1.46.
 - mysql-connector-java-5.1.46-bin.
- Once this is done, add in the 'test' file which is the PHP website into NetBeans.
- To start the server for the MYSQL database complete the following steps:
 - Windows key + r.
 - RUN services.msc
 - Find MYSQL57
 - Start the service (if not running already).
- To start the other server for the PHP website enter the 'finalProject' folder.
 - Click xampp.

- b. Click xampp_start and once the application opens, let it run in the background.
9. Run the 'test' script before the 'orderGeneration' script.
10. Run the 'orderGeneration' script.