



Restaurant Management and POS System

Final Year Project
B.Sc.(Hons) in Software Development

BY
MARK GILMORE
JOSEPH GRIFFITH

MAY 21, 2020

Advised by Martin Hynes
DEPARTMENT OF COMPUTER SCIENCE AND APPLIED PHYSICS
GALWAY-MAYO INSTITUTE OF TECHNOLOGY (GMIT)

Contents

1	Abstract	2
2	Introduction	3
2.1	Chapter Overview	3
2.2	Executive Summary	3
2.3	Decision Making Process	4
2.4	Restaurant Management Systems	4
2.5	Point of Sale (POS) System	5
2.6	Management Systems	5
2.7	Current Flaws/Problem Statement	6
2.8	Project Proposal	7
2.9	Project Objective	7
2.10	User Authentication	9
2.11	Till GUI	9
2.12	Remote Ordering	9
2.13	Sales	10
2.14	Receipts/Hardware	10
2.15	Reports	10
2.16	Users	10
2.17	Categories	11
2.18	Products	11
2.19	Chapter Review - Methodology	11
2.20	Chapter Review - Tech review	12
2.21	Chapter Review - System design	12
2.22	Chapter Review - System evaluation	12
2.23	Section Review - Conclusion	13

3	Methodology	14
3.1	Chapter Overview	14
3.2	Meeting	15
3.3	Problem Solving	15
3.4	Version Control	16
3.5	Selection Criteria	17
3.6	Development Methodology	18
3.6.1	Waterfall	18
3.7	Conception	19
3.8	Initiation	20
3.9	Requirement Gathering and Analysis	20
3.10	System Design	20
3.11	Implementation	21
3.12	Integration and Testing	21
3.13	Deployment of system	21
3.14	Chapter Summary	21
4	Technology Review	22
4.1	Introduction	22
4.2	What is a Management System?	23
4.3	History of inventory management	24
4.4	Role of inventory management	26
4.5	Impacts of inventory management systems include:	27
4.5.1	Customer satisfaction	27
4.5.2	Pricing	27
4.5.3	Maximizing efficiency	27
4.5.4	Financial management	27
4.5.5	Menu planning	28
4.5.6	Forecasting and ordering	28
4.6	Advantages of a Management System	28
4.6.1	Speed and Efficiency	28
4.6.2	Document Generation	28
4.7	Disadvantages of a Management System	29
4.7.1	Reliance on Technology	29
4.7.2	Accuracy Issues	29
4.7.3	Risk of Fraud	29
4.8	Areas of further research	29
4.9	Users	30

4.10	Use Cases	30
4.11	Project Platform	31
4.12	Project Software	31
4.12.1	Xampp	32
4.12.2	Visual Studio Code	32
4.12.3	MySQL	32
4.12.4	LaTeX	32
4.13	Plugins	32
4.13.1	AdminLTE	33
4.13.2	Ion-icons	33
4.13.3	Bootstrap	33
4.13.4	Font Awesome	33
4.13.5	Bootstrap data tables	33
4.13.6	jQuery	33
4.13.7	jQuery Number	34
4.13.8	Fastclick	34
4.13.9	Sweet alert	34
4.13.10	TCPDF	34
4.13.11	Date Range Picker	34
4.13.12	Morris.JS Charts	35
4.13.13	Chart.js	35
4.13.14	Input mask	35
4.14	Measurable Goals and Requirements	35
4.14.1	Management	35
4.14.2	Staff	36
4.15	Non-Functional Requirements:	37
4.15.1	Management	37
4.15.2	Staff	37
4.16	Conclusion	37
5	System Design	39
5.1	Chapter Overview	39
5.2	Model-View-Controller	39
5.3	Component Diagram	39
5.4	Data Storage	40
5.5	Relational Database Management System (RDBMS)	41
5.5.1	Data Structure	42
5.5.2	Privileges	42

5.5.3	Data Recovery	42
5.5.4	Flexibility	43
5.5.5	Data Consistency	43
5.5.6	Speed	43
5.5.7	Ease of Use	43
5.6	Why RDBMS	43
5.7	Entity Relationship Diagram	44
5.8	Database Design	44
5.9	Graphical User Interface	46
5.9.1	Till GUI	46
5.9.2	Management GUI	48
5.10	Pricing	49
5.11	Flow Charts	50
5.12	Ajax/Java Script	50
5.13	Chapter Summary	51
6	System Evaluation	53
6.1	Chapter Overview	53
6.2	Project objectives	53
6.3	Testing	54
6.3.1	GUI testing	54
6.3.2	Functional Testing	54
6.3.3	Unit Testing	56
6.3.4	End to End Testing	56
6.4	Limitations and areas of Improvement in the project	58
6.4.1	Limitations	58
6.4.2	Improvements	58
7	Conclusion	59
7.1	Chapter Overview	59
7.2	Reflection	60
7.3	Learning Outcome	60
7.4	GitHub	60

List of Figures

3.1	Waterfall Methodology.	18
4.1	Bar chart showing the increase in value of the POS terminal market size.	23
4.2	Patent of the original bar-code scanner.	25
4.3	Inventory Control	26
5.1	Component diagram.	40
5.2	Database schema.	45
5.3	Till GUI	48
5.4	Side tabs.	49
5.5	Flow chart of the entire system.	50
6.1	GUI Testing.	55
6.2	Functional Testing.	55
6.3	Unit Testing.	56
6.4	End to End Testing.	57
6.5	End to End Testing. pt2	57

Chapter 1

Abstract

Today's businesses need efficient and effective methods of keeping track of daily transactions to help them with various tasks such as stock levels and cash flow. With competition stronger than ever business owners need reliable information to make well informed decisions that may effect their business in the short and long term. Storing and retrieving accurate and up-to-date information is crucial as is the ease in which it can be accessed and broken down into forms such as graphs and excel sheets where it can be easily read and analysed. This is where we saw the need to develop a Restaurant Management and Point-of-Sale (POS) system for a café or restaurant business environment. The main programming language for implementation is PHP coupled with javascript and a MYSQL database.

The system aims to make more efficient the workflow of a user by providing a fast and easy-to-use UI as well as very accurate, up-to-date and easily accessible reports for owners and management.

While this particular system has been designed for use in a restaurant or café it can be adopted, with ease, to be used anywhere a business needs the combination of easily accessible reports, stock management, customer integration and a working till system all in one.

Chapter 2

Introduction

2.1 Chapter Overview

This chapter provides an overview of the entire project from its inception, planning, building and ultimately its conclusion. An executive summary has been included in this chapter to provide an insight into the tasks at hand, the development of a restaurant Point of Sale and Management System. The insight that covers the problem states, existing solutions, proposed solutions and the development track of the project. Finally the chapter provides a brief summary of all the chapters with a run-down of what is expected in each one.

2.2 Executive Summary

Working in a team of two we have proposed and developed a web based Restaurant Management and Point of Sale (PoS) system. The system is an amalgamation of the two technological pieces in a bid to improve existing platforms to become more usable and efficient. By integrating the functionality of both into a single platform we believe we have raised the bar as far as ease-of-use and productivity goes.

This paper documents the entire build process, the wider consultative process, and future plans for the proposed system. In a bid to achieve the best in our dissertation, the build process is illustrative of research skills, academic excellence as well as an inquisitive for science. The paper therefore explores the unique learning curves as well as a viable problem statement. The study

has been segmented into different categories to be illustrative of the innovative space that we seek to explore. The Introduction provides an insight into the entire project, its milestones, goals and overall objective.

2.3 Decision Making Process

The decision-making process was based on solving recurrent, everyday technological challenges in our everyday life. We didn't just want to build something to get a good grade, we wanted to build something which we knew we could use in the real world that would greatly improve the productivity of ourselves and others. We narrowed this down based on our in-depth knowledge of the particular field of information and business management. As computer scientists, the quest for working in challenging and meaningful projects for current and future users was insatiable. This was the driving factor behind the team from conducting qualitative review of all proposals to ensure it was beneficial to the society as well as improving our own technical knowledge of key computing areas and developing our individual learning ability. With all this in mind we set about organizing our first meeting to brain-storm different ideas.

Quite early in the meeting, we decide to work on a Point-of-Sale and Stock Management System software custom developed for a shop. We identified a need for a new software that would challenge the status quo in the market by providing better solutions. The next step was a meeting with our project supervisor who illustrated the applicability of the idea of a Restaurant Management System (RMS). After further consultation we adopted the idea as our new project. We had a follow up meeting where we drew up different achievable goals that would lead to a successful project. This being an academic research work, we ensured that the decision-making process was informative, considerate and explored science based on current issues affecting society.

2.4 Restaurant Management Systems

As the name suggests, Restaurant Management System (RMS) is a software/hardware system used to manage the daily operations of a bar/restaurant

or cafe. Unlike a regular Point of Sale or a Management System, RMS is built specifically for restaurant operations with add-ons that make it more usable and scalable. Typically a RMS can therefore be regarded as a type of Point of Sale software.

According to Lime, A Restaurant Management System (RMS) is a type of Point-of-Sale (POS) software specifically designed for restaurants, bars, food trucks and others in the food service industry. Unlike a POS system, and RMS encompasses all back-end needs, such as inventory to staff management[14].

Developing an RMS system required the two of us to hone our skills in analytic thinking and innovative process. Due to the multifaceted fields of study in the project, we utilized a carefully laid out plan that ensured the learning curve was profound and balanced with the utilization of skills. This meant that out of the languages and skills we had to learn, we ensured that we had the necessary skills to learn the new languages within the speculated time.

2.5 Point of Sale (POS) System

A POS system is an application used to manage business transactions between a client and a company. Often a POS is facilitated with a cashless payment procedure that improves on speed and efficiency. The most common platform is either a debit or credit. Everywhere we go we see some sort of POS system in use. Be it to the shop or to the gym, businesses need a way to process payments with ease.

2.6 Management Systems

Management Systems are applications that allow the manager/owner of a business to keep track of sales, staff, customers and inventory. It allows for quick and up-to-date decisions that benefit not only the owners but staff too. Management systems are an integral part of the management of any business. Management systems rely heavily on the back-end development. This is a result of the main reports that the software has to compute. The

reports are developed to assist the management and staff in ensuring their business skills are successful while making effective changes in critical areas contributing to the overall success.

2.7 Current Flaws/Problem Statement

Similar to any other entity, current POS systems are not perfect. Furthermore the majority of them lag behind in terms of current technology. One of the challenges we identified was a lack of consideration of the end user. Part of this can be attributed to a poor communication channel between the developers and their clients or end users. As a result some of the add-ons, besides the basics, don't necessarily improve on the work flow of users. For instance, not being able to void anything other than the last product added leads to a lot of moving around the product to take it off the bill. The software must be able to address the customer's specific challenge [11].

The following is a list of a few of the problems current RMSs have:

- Voiding of only one item on the bill
- Concluding a transaction takes more than one click
- Customer purchase history not in system
- No voucher transaction button

A similar challenge is overlooking an important part of utility such as developing a one touch processing of a transaction. Often to process a credit/debit card payment or voucher transaction a user usually has to press at least three buttons. While this does not seem like much to the average person, on a busy night this can result in time wasted as well as mistakes being made which could cost the company money, or worse, it could cost them a customer [25].

The problem statement therefore has to be developed carefully by being inclusive and considerate. This meant that we conducted a number of site visits to different restaurants where we learnt the challenges of the different organizations. In these cases we only had to conduct simple oral interviews which included questions to frequenters of restaurant and other service oriented facilities. Besides this we had a sitting in which we discussed, in detail,

our experience working in spaces where a point of sale was critical to the survival of the business.

Finally, we consulted with different persons in the information technology industry to ensure we were working on a viable project. This includes our supervisor, fellow students as well as industry professionals. In order to collect qualitative and quantitative results we referred to editorials on the topic, the different technology platforms used and current solutions. After carefully crafting the problem statement we proceeded to develop our project proposal as outlined below.

2.8 Project Proposal

Our system is designed to address challenges in user productivity in a POS while maintaining accurate data on the transactions and business proceedings by developing a Restaurant Management System. The proposal was developed while maintaining the highest industrial standard as well as incorporate innovative technology. This will be achieved by developing a disruptive technology that will challenge the status quo as well as influence clients preference for POS systems.

The project is perfectly situated to resolve the challenges of separated utility between a point of sale and management system. Developing a system that has the utility and practicability of two entities is disruptive in nature while ensuring it resolves some of the fundamental challenges current systems have. The project illustrates an exploration of science as well as an interest in community development by conveying different technological platforms to resolve an impending challenge. This has been developed into a clear and concise project objective ensuring the project achieves its ultimate goal.

2.9 Project Objective

The main goal of our project is to create a universal web-based restaurant management and point-of-sale system with remote ordering capabilities (via a tablet). This system is applicable in a bar/restaurant and can easily be adopted to a café, shop or any premises that require a till system to pro-

cess payments, print receipts, use a barcode scanner as well as being able to generate reports. Our system aims to help managers and owners to make reliable decisions that will benefit the company both in the short and long term.

Considering the restaurant default applicability, there are so many of them offering closely related types of cuisines. With the high competition in place, the management and staff must ensure they please their customer during their short stay to their establishment. Often this is achieved by making prior arrangements and work orchestration. One process that must be carefully planned is ensuring the customer, while making payments, is dealt with in a timely manner. This is done by ensuring you optimize the speed and ease of making payments as well as avoiding service interruptions from machine or system downtime. An all round good experience will ensure that you retain most of your valid customers.

As pointed out earlier the an RMS can have a number of adds-ons custom built for specific customers. A good example is a loyalty platform that rewards customers based on their purchases and purchases dates. You can improve on customer services by regularly checking on customers based on their stored customer emails or customer phone numbers.

To accomplish the goals we had set out we opted to use the following programming languages due to their relevance to our case. This include PHP, JavaScript, MySQL as the database, CSS and HTML for the front-end design purposes. We had an appraisal for the project since both of us have worked in work spaces that require PoS systems. As a result we had sufficient insider information on the specific requirements for the task. We opted on improving the functionality of current systems by addressing their shortcoming while improving their workflow and usability. This should ensure the user can concentrate on serving the guest as opposed to constantly working a cumbersome till.

From the onset we knew that our project had to be worthy of the 15 credits on offer at level 8. This played a role in the decision to learn a new language to complete the task. After some research we decided we would learn PHP to develop our system as main language, while JavaScript was our main subsidiary language. PHP was chosen since it is an Object-Oriented programming language and we have no prior experience with the language. In

light of our problem statement, we thought it would be good challenge to try something new and broaden our own knowledge on the language. Further we opted to use MySQL for the database which will be discussed further in chapter 5 of this paper. Below is a list of features we have incorporated for this project in order for it be deemed successful.

2.10 User Authentication

User will be able to login and logout of the system selecting a predetermined username and password. This is an especially important task as it allows the owner or manager to see the individual sales of the employees which in turn allows the evaluation of performance, and in some cases, recognition for individuals that are performing well. This feature improves on employees' motivation and can further lead to better work ethic and productivity. This feature is also important to distinguish between an employee and an administrator. Employees, when logging in, will only be able to access limited features such as the till and the addition of customers. They will not be able to see sensitive information such as sales reports.

2.11 Till GUI

A good point of sale system, while appearing simple to a regular user, is very complex and has many moving parts. The main part of this system is the till. This has to be connected to every aspect of the system, constantly distributing and receiving data, all while maintaining a fast and efficient user interface that doesn't stall or slow at any point of the process.

A till has to have all the functionality a user needs to complete their job in a timely manner and as easy as possible.

2.12 Remote Ordering

A vital part of any restaurant POS system is the ability to take orders remotely. A large restaurant with many tables and customers can be very stressful for employees. This system allows users via a tablet to access all the functionalities of the main till and in turn alleviates the need for staff members to be behind the bar. This reduces commotion in the main floor

caused by serving and collection of bills. This provides efficiency to the daily running of the business.

2.13 Sales

The system allows staffers to choose products and save them to a table number. This is used to make payments as well as make follow up orders. The management is able to view the payment method used. This is based on the generated reports that are used to inform owners or other stakeholders of detailed transactions in the premises. Each transition also saves the staff members name as well as a transaction id so if anything goes wrong or there is a problem with a payment it is easy to review and access its details.

2.14 Receipts/Hardware

The user is able to print receipts of saved tables as well as already cashed off sales. The receipt includes all the details of the sale including staff member, receipt number, products, discount and price.

The system allows the printing of receipts using a receipt printer as well as a barcode scanner which allows staff members to either scan products and/or scan loyalty cards to get customer details and the discount amount.

2.15 Reports

Up-to-date and reliable reports are essential in the running of a business. The system offers full reports of sales that can be selected by based on different timelines, i.e. full day or a couple dates.

2.16 Users

The owner plays the role of an admin by adding users with usernames and passwords for login. The owner can also choose the role a staff member has in order to limit what he/she is able to access. This is very important as an owner does not want every staff member to be able to access the details of

the company.

This system will also be used by members of staff. With limited access to data they will still have everything they need to complete a sale from start to finish. Customers

This portal can be accessed by both the staff members and the owner. In the main till part of the system a staff member can add a new customer in order to sign them up for a loyalty discount. The discount can be set differently for each user. This loyalty program has been designed such that when typing in the customer number or scanning their loyalty card it comes up on screen and discount is deducted from the customers total before they pay.

2.17 Categories

The proposed system allows an owner to set up different categories of their products. For each category the user will be asked for a VAT and Tax rate which will be added on to the product price when it is added to the database. For our project we have set up food and drink categories with the current VAT and Tax rate for the goods. An owner can decide to break down the categories even further (i.e. Burgers, desserts, wines and spirits) but with the rates still being the same we decided to have just food and drink.

2.18 Products

Products can be added under two categories (food and drink). The product is then given a unique code that corresponds to its category (100's for food and 200's for drink). This is used when choosing the category and entering the buying price of the product. Hardware

2.19 Chapter Review - Methodology

In the Methodology Chapter we discuss some of the approaches we used towards different aspects of the project during the various stages of the project lifecycle. In this section we discuss the Waterfall approach and how we came to the decision that was the best this method to help us build our project. We will also discuss our choice of version control and how this helped us

progress through the various stages of construction. For a project of this size meetings had to be regular. In this chapter we will go through the frequency and structure of our meetings to give you a greater insight to how we pieced this project together.

Testing is a critical part of any project and in this chapter we explain how we rigorously tested our project using both black box and white box testing as well as what tools we used to help us achieve a satisfactory outcome with regards to the coverage and results of testing.

2.20 Chapter Review - Tech review

In the Technology Review chapter we will explain, in-depth, the vast research we done to come to a decision on all the aspects of this project from visual styling to the software design pattern used throughout development. We will also review the various back and front end technologies used that helped us achieve our goals and complete our project in an efficient and timely manner.

2.21 Chapter Review - System design

In this chapter we give an in-depth explanation and description of the overall system architecture and design of our POS system. We explain the decisions behind why we chose specific technologies and how they are implemented throughout our system. These will be illustrated by using a UML diagram as well as screenshots of our code and the working program.

We will also go into detail about the hosting of our project and why we came to the decision we did to host it in a particular cloud platform.

2.22 Chapter Review - System evaluation

In this chapter we will evaluate our system with regards to robustness, testing and scalability and provide results as well as an in-depth critical analysis of testing results and whether our initial objectives were met.

We will also discuss our POS systems limitations and where, if given more

time, we could have improved the system to make it better and fir for the use in a professional capacity in a real bar/restaurant.

2.23 Section Review - Conclusion

In this subsection we will narrate our conclusion and the experience as a whole. We will highlight our findings from previous chapters and list any outcomes we had as well as going into areas we have identified for future investigation. We will also give a reflection on our project, what we learned and whether or not this experience has benefited us. Further we will give a conclusion on the importance of team work in such large projects.

Chapter 3

Methodology

3.1 Chapter Overview

This chapter covers the methodology used in our project. It covers in detail the different steps that we took in the entire project. As discussed by Rachiele, methodology in software development is a process that includes a number of tasks by default being; planning, creating, testing and finally deployment [21]. Besides this, methodology in software development is professionally based on a prescribed process which develops use to be timely, organised and structured.

There is a number of methodologies that are suitable for this project. This includes but is not limited to Agile, Scrum and the Waterfall method. After careful consideration we agreed to use the Waterfall method for project management.

The Waterfall method was picked because of its applicability to develop a POS system where every phase is intertwined and rely on other completion of singular phase sequentially for resources to be released for the next phase. For example the products table could not be started unless the categories table was complete as it needs a category to be saved to the database. The project is full of scenarios similar to this rendering the Waterfall method as the most viable option.

3.2 Meeting

Meetings between team members took place every week. Initially we had meetings twice a week to develop quorum as well as a rudimentary plan which would then be adopted as the project methodology. We then settled for once a week meeting which was sufficient as we developed the work remotely. [fig]. The meeting would start with discussing the things we have learned since the last meeting. This part of the meeting was vital as we had to be on the same level when learning a new language. Very early on in the project we identified several online tutorials where we could learn the fundamentals of the PHP programming language [24]. At the end of every meeting we would set a couple of milestones we would have to have achieve development wise before the next time meeting. These meeting usually corresponded with certain parts of the language we needed to learn in order to complete the project.

3.3 Problem Solving

From the inception of the project we knew that learning a new language was going to cause a number of minor problems throughout the project. We therefore devised a problem solving process which would guide us and therefore we adhered to it throughout the development life-cycle. We therefore ensured that the learning curve was manageable and pointed towards the success of the project. We further established boundaries as far as our learning and problem solving capabilities are concerned to ensure we remained on course as we made relevant research and consultations.

Problems were solved by scheduling a meeting and discussing the problem at hand. The problem was then broken down into small steps where each team member was designated a task to research and solve. Another meeting would then take place to put the code together and test it to see if it worked. This approach really benefited us as we both learned a lot about the parts of the project in which we weren't directly involved. Working as a teams also ensured that we complemented and supplemented each other where necessary. The process also ensured we developed a continued learning culture while we furthered aid in developing updates and more ground breaking technologies.

Part of the critical steps in problem solving is constantly consulting with

the market players to ensure that we solve existing challenges. This was done after our research work indicated that often developers work resolved around assumed or inaccurate challenges. This is a result of poor problem solving skills. We therefore maintained regular consultations with our potential clients to ensure each and every build was oriented to solving their problems.

3.4 Version Control

GitHub

In order to work on the project as a team while maintaining industrial standards, we required a number of tools to control and manage our work. After researching on potential collaboration tools we agreed to use GitHub and Microsoft Teams. GitHub offered us various tools to aid in the planning, tracking and development of our project. Throughout our project we used GitHub as our version control tool.

One specific areas where GitHub was quite collaborative is the creation of the repository and adding the members of the team to work as collaborators the GitHub's task management tool. This ensured that both team members had valid contribution to the project. Further it enabled us to supplement our skills where necessary. Using GitHub also enabled us to consult on very specific areas of coding strictly on system engineering and system design. This ensured that the project maintained professional control framework as far as industry standard is concern.

We then proceeded to building our Kanban board, posting every task to it in order to keep track of where we were and what was expected next. This allowed us to break down our tasks into smaller, more manageable tasks. Through the utilization of its features we found GitHub to be an extremely useful tool in the research and development of our project as it allowed us to keep track of code, tasks and errors all in the one place. The Kanban board also assisted in the documentation of the project avoiding instances where different documentation structures and process would be required. This provides grounds for further research as well as future updates to the program by reviewing the challenges and steps taken to resolve them.

Use of GitHub also enabled us to be responsible developers in the information communication industry. By documenting our work in the platform, we provide grounds for other developers to research on similar projects. This meant that we were open to “public” scrutiny and opinion by contributing in a productive and illustrative manner. Overall GitHub plays a vital role in bring developers on-board and we were proud of contributing to the efforts. Using Microsoft Teams allowed us to keep in contact with each other and our project supervisor via messaging and video chats. Utilizing its share screen function we were able to both have an input into sections of the projects being covered by the other person. This was especially beneficial after the outbreak of the covid-19 virus which restricted us from having face-to-face meetings.

3.5 Selection Criteria

After laying out the problem statement, project proposal and problem objectives we finally had to start the actual build. We therefore embarked on a selection process, where we derived the best languages and additional tools that would be best for the project. This meant that we had to self-evaluate and reflect on the goals of the project. This stage was easily determined a vital step in the project milestones. The selection criteria for our project and indeed any project is key and something that must be laid out clearly. When selecting our languages, platforms and technologies we needed to ensure that all platforms would meet a certain standard that would enable us to tackle the challenges that were foreseeable in the different stages from beginning to completion.

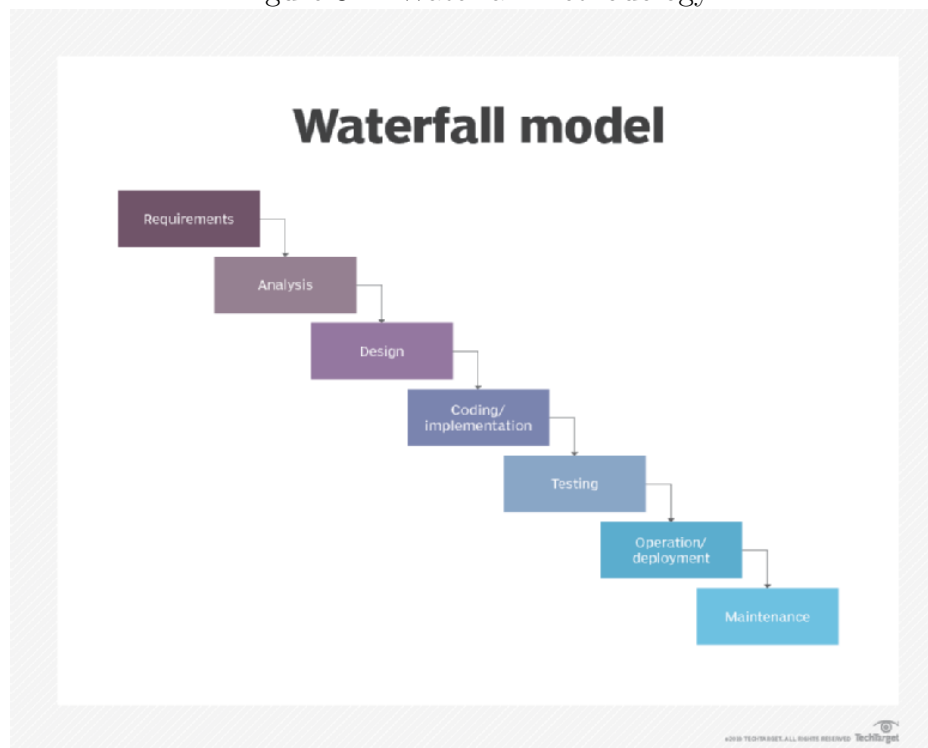
We had to make sure that the language chosen was not only a language worth learning but also one that would be able to help us build the project as efficiently as possible. The technologies chosen had to be capable of running what we were building and be compatible with future hardware and software developments.

3.6 Development Methodology

3.6.1 Waterfall

According to tutorial points editorials, the waterfall model is a simple to use approach where one phase must be completed before the next phase begins while avoiding overlapping [20]. The tutorial points outlines the bases of the model as a linear sequential flow. The Waterfall model requires the following of sequential phases throughout the software development cycle, starting with initial instructions regarding a customer's specific requirements followed by the practical implementation and product construction. Using the waterfall method leaves little room for change as the project proceeds therefore initial planning has to be thorough and every aspect of the project has to be designed and addressed before we could move on to the next stage.

Figure 3.1: Waterfall Methodology.



The following are the steps involved in the Waterfall process and how we incorporated them into the building of our project. We will give a brief

summary of the process as well as explaining which meetings we held to discuss which step.

3.7 Conception

This phase begins with the innovative idea we wish to develop. It involves a broad assessment of the project, whether or not it is viable and why it is beneficial to the individual team members. To begin with, we decided to have meetings twice a week to discuss the ideas for the project and what we would be interested and productive to work on. In these initial meetings, we discussed the various projects we could undertake and whether or not they would be worthy of a 15 credit, level 8 module.

A POS system was the first project that we proposed because of its complexity. The project was also viable because of the number years we spent working in areas using a till system, something we both had a strong understanding of, and knew the work around different systems and how to improve them.

We also looked at other ideas such as websites and applications that could buy and sell products among other things. These were all good ideas but we knew that, with our deep understanding and experience with POS systems, we could make a change and develop better systems that solve client problems. We didn't just want our project to be good, we wanted it to stand out and be worthy of a higher grade.

The next step in the process was the language. Learning a new language was always something that we put a lot of consideration in to as we knew learning something new would benefit us. In a lecture by Dr. John Healy he mentioned that we all should look at the possibility of learning a new language in order to broaden our knowledge in a chosen field. We therefore made a decision to explore the learning curve of a new programming language.

After a vast research into which programming languages we could use, PHP came up time and time again, as a great solution to what we were trying to achieve. Given that PHP is an Object-Oriented language we thought it ticked all the boxes that we needed to achieve our goal of building our POS

system.

3.8 Initiation

Once the idea is agreed upon we began to define objectives, scope, purpose and deliverables of the project. At this stage we put together a list of the different steps we needed to take to complete every step of the project and move on to the next phase. In this stage we made sure the potential clients for the system had made their submissions as far as a viable POS and RMS are concerned. After combining our joint experience of POS systems we put together a list of features the system must have to be considered viable and suitable for use. The list is found above under the objectives heading.

3.9 Requirement Gathering and Analysis

This part of the process required team members to research and identify all possible requirements of the POS system we aimed to develop. Specific functionality of each component must be identified, addressed and carefully thought out and documented before moving on to the next phase of production.

This process took time, with several meetings conducted to address different ways in which we would approach the build process. This was complimented using our own knowledge and experience of POS systems we researched extensively into what makes a good POS system and what parts of existing systems are better left out. This was a critical step in ensuring we develop a system that address challenges in the field as opposed to developing systems that solve developer quest for flashing software. We therefore made sure we incorporate the different actors' opinion on the issue in rational and professional manner.

3.10 System Design

In this phase the system specifications from the previous phase are addressed, and the system design is discussed and agreed upon. The system design helps specify the overall system architecture of the project whilst also specifying

hardware and system requirements needed to complete this project. In this phase we reviewed the complex requirements of a POS amalgamated with a RMS ensuring that all the characteristics are built into the systems. In the System Design chapter we will take a more in depth look at this process and how we approached it.

3.11 Implementation

Using inputs from the system design, the system is differentiated into smaller units. This smaller units are then build based on the systems design and system objectives. This phase carries the most regal room as far problem solving is concerned. We therefore made sure we paid attention to detail, constantly consulted and made sure development remained on course. We then integrated the units into the next phase of the project cycle. Each unit is developed and tested to ensure it is the best fit for the intended purpose and that its functionality runs as it is intended to run.

3.12 Integration and Testing

When all units are developed and tested, they are integrated into the main system. Post-integration, the entire system is fully tested to ensure that there are no faults. In this process we used Microsoft Teams to ensure both of us observed the results of these tests. We will delve more into the testing process in the testing chapter later in this paper.

3.13 Deployment of system

?????????? When the system was built and fully functional and all the testing was done the product was deployed onto the CLOUD???? Again we used Microsoft Teams to view this process as it happened.

3.14 Chapter Summary

Chapter 4

Technology Review

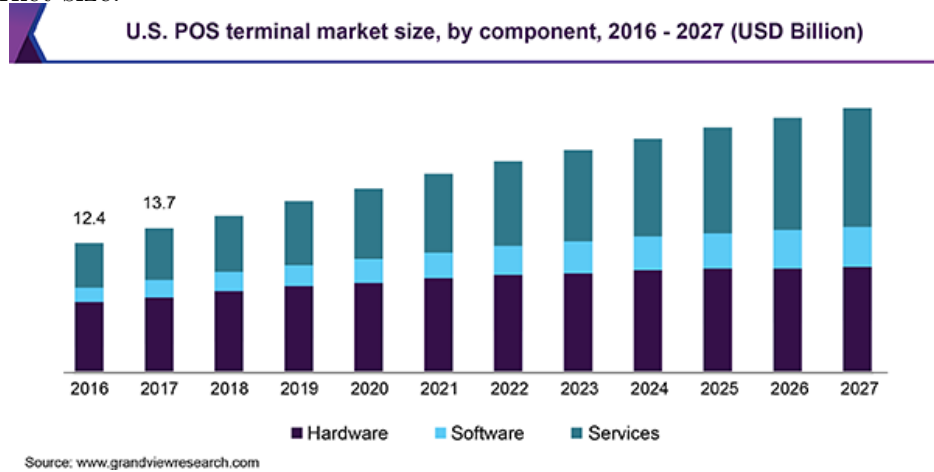
4.1 Introduction

The aim of the literature review section of this or any dissertation is to show that the author has indeed researched, understood, and has a good grasp of, the main work concerning the chosen topic or question in your field. This work can be done by searching online for websites, papers, and books that can be used to further enhance the knowledge of the topic at hand. The review is guided by one's research objective or by the issue or thesis someone is trying to argue out that will also provide the framework for one's further work.

This article focuses on precisely that. It provides a look into Point of Sale and inventory management systems and their relation and impact in the restaurant industry as well as other industries where they can be adopted too.

Today we are seeing a huge shift in restaurant technology. From touch screen ordering systems at tables to kiosks that you can order food and go to collect it, restaurateurs across the world are using new, modern devices to keep their restaurants on the cutting edge of the industry. According to Grand View Research, the global point-of-sale (POS) terminals market size in 2019 was USD 69.0 billion and is expected to reach USD 125.9 billion by 2027[fig]. This proves that the market shows no sign of slowing down. Where you can buy goods there is always a need for a stock management/POS system.

Figure 4.1: Bar chart showing the increase in value of the POS terminal market size.



With new technology, the interaction between customers and members of staff is dramatically reduced thus creating a new challenge for business owners. Customer service is still a vital part of a business and keeping customers coming back. Superior customer service is at the heart of every great restaurant, and new restaurant technologies can give restaurants an edge in responding to customer needs (Strong 2013). If a customer's only interaction with staff is to pay, then this part of the process has to be as efficient as the rest of the experience, and having a fast and reliable POS system allows this.

4.2 What is a Management System?

Management systems are systems that allow restaurant owners /managers to manage their stock levels through a piece of software linked from a computer to the point of sale system. They assist owners in tasks such as avoiding overstocking, avoiding running out of stock, overall sales, customer sales and staff sales. A well designed and reliable system is essential to any business. It therefore helps the user make key decisions on whether to keep products or not, whether to reach out to a customer that hasn't been to your restaurant in a while and to review staff sales to see if they are doing their work correctly. Being able to make these well-informed decisions leads to increased profits

along with greater stock management.

4.3 History of inventory management

Up until recently manual ordering in restaurants was used. Notepad and pens and basic point of sale systems were very common. Each order was taken then rang up on the till and the order sheet was then handed into the kitchen. This was a very time consuming and inefficient task which cost owners money and prolonged the time it took for customers to get their order. This was not too dissimilar to retail outlets from the past.

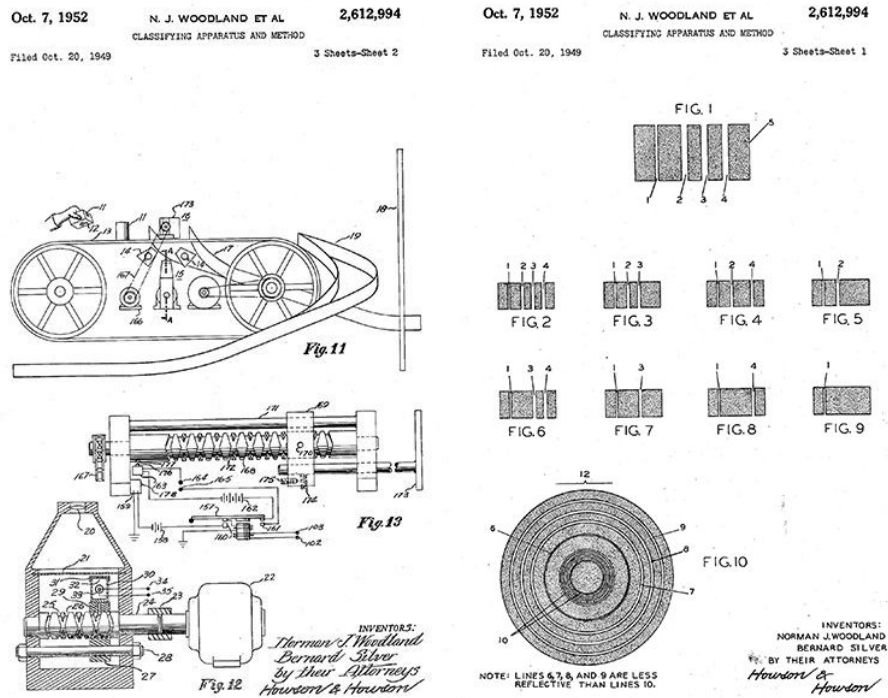
Many years ago, in retail outlets, workers wrote down every sale to keep a record and calculate it all up then check against stock levels at the end of every day. This was a time consuming and a very tedious task which was added to the end of every workday. This work was highly skilled as it had to be exact or there would be an imbalance. This imbalance could be found in small retail operations so the room for human error in a big business could be massive.

After the Industrial Revolution, efficiency and mass production became the main goals of businesses, along with improved customer experience at the point of sale (Miller,1997). In the early 1930s, the first modern check-out system was designed. Punch cards that corresponded with catalog items were used to process transactions. The cards were read through a computer which then passed the product information to the storeroom. Items were then brought from the back up to the customer finally the transaction was complete. This system revolutionized stock control. Now business owners were able to keep track of all sales as well as generate billing records and manage their inventory. This greatly reduced the risk of stolen stock as well as saving on man-hours.

While it saved owners money and time, the punch card system was an expensive one. Merchants needed a better system therefore researchers created one. In the late 1940s and early 1950s, researchers created the first bar-code system. It used ultraviolet light-sensitive ink and a reader to mark items for sale.

Once again, the system was too expensive and lacked the computing power

Figure 4.2: Patent of the original bar-code scanner.



needed to make it work. While retailers knew what they needed to keep tabs on their stock levels, society simply lacked the resources and computing power to make any great strides into the field, and thus the concept stalled, and progress was nonexistent.

In the 1960s the development of affordable laser technology revived the concept of an efficient stock management system. Lasers technology allowed for the manufacturing of smaller, faster, and cheaper scanners. In the late 1960s the bar code, or the Universal Product Code (UPC) as it was called at the time, was developed. With vast improvements in computing power, the ability of UPC codes to help track and manage inventory improved exponentially.

During the 1990s, due to advances in computer and software technology, retailers began implementing modern inventory management systems. Modern inventory management systems must have the ability to track sales and available stock levels. They also must be able to provide up to date informa-

tion for the owner to make key decisions regarding purchasing.

4.4 Role of inventory management

Inventory often represents a significant portion of a business's operational revenue. According to the Oxford dictionary, inventory control is defined as the fact or process of ensuring that appropriate amounts of stock are maintained by a business, to be able to meet customer demand without delay while keeping the costs associated with holding stock to a minimum.

Figure 4.3: Inventory Control



Stock levels must be kept adequately controlled to allow a business to offer customers the best possible service. If a customer visits any retail outlet to purchase a product but that product is sold out it could lead to them never returning. Owners and management need to ensure that adequate forecasting is maintained to complete all sales, especially in such a competitive market as we see today. This is especially important in the restaurant trade. An owner will always want enough items in stock to fulfill any orders it receives although they do not want to have excessive stock levels which will result in food being thrown out. This costs money which could inevitably destroy the business. The goal of any well-run business is to have all choices available to customers but not have excessive amounts of leftovers. Although the concept appears simple, it requires careful planning, standardized procedures, and monitoring to achieve desired results [3].

4.5 Impacts of inventory management systems include:

4.5.1 Customer satisfaction

If a management system is used to the best of its capabilities, then stock levels are always kept to a happy medium in which any product is there to be purchased at any time. Owners can analyze product sales in real-time and make key, well-informed decisions that let the running of business seem almost effortless. Owners can have their popular items in stock ready to instantly fulfill any order. They can also analyze past sales to adequately forecast what sells and on what occasion. This will result in gaining a good reputation among the general public as being reliable hence bringing in repeat custom.

4.5.2 Pricing

Having good stock management reduces the chance of stock waste. This will result in the saving of a lot of money. Keeping stock at a good operating level will keep business costs down and in turn, keep the costs of the customer down.

4.5.3 Maximizing efficiency

Maximizing efficiency is a key objective for this project. This was not only an important aspect of the POS part of the project but of the business management side of it too. A good management system will not only affect the price of items but will help in the handling and storage aspects of business too along with having the right staff on when deliveries are coming in. Knowing everything you need to know will result in products finding its way to the customer as quickly and as efficiently as possible. The inventory doesn't need to be counted.

4.5.4 Financial management

Improved financial control is one of the most important goals of inventory management systems. A well-designed Management System will ensure the maximum value is generated from purchased stock. In addition to the actual

cost of acquiring inventory, costs are associated with transporting and storing inventory[fig].

These costs are called carrying costs.

4.5.5 Menu planning

Menu planning is a key part of any restaurant. An owner needs to know what's selling and what isn't. There is no point in having stock that isn't selling left there and going out of date only to have to order more of the same stock in. Decisions need to be made on every under-performing item on the menu. This requires considerable planning to ensure that menus are as cost-effective as possible while also providing customers with what they want at prices they will purchase them at.

4.5.6 Forecasting and ordering

Forecasting can greatly affect inventory control. Bad forecasting can lead to excess ordering thus tying up a lot of money in inventory. This will, in turn reduce the cash flow of the business and lead to a lot of waste of stock.

4.6 Advantages of a Management System

Management systems offer up to date sales figures along which can be used to gauge the performance of all products on sale. While the need for hand counting stock will always be there to ensure your business isn't being robbed or people aren't making mistakes, management systems still make the life of the owner a lot easier.

4.6.1 Speed and Efficiency

An up to date sales form can be formed in seconds, giving the owner all the information, they need on anything. This saves on time, money, and man-hours that can be directed at other areas of the company.

4.6.2 Document Generation

Receipts can be generated at the click of a button allowing the user to document all sales, Secondly the user can generate excel sheets of data ranging

from reports and graphs that are created as sales and transactions are made.

4.7 Disadvantages of a Management System

4.7.1 Reliance on Technology

Using a computerized management system, while extremely useful, poses some disadvantages. One is an over-reliance on technology, Outside factors like power failures or the loss of Internet connection or network connectivity can render the system useless. This can lead to loss of earnings and customer dissatisfaction.

4.7.2 Accuracy Issues

A computerized management system alone does not ensure accuracy. A system and its data are only as good as the information passed into it. If a member of staff makes a mistake this could have serious ramifications down the line. Inaccurate stock and cash levels could be detrimental to a company as well as compromising the integrity of the reports it generates. This is why a plan must be put in place to be able to validate their data and check the figures reported by the system.

4.7.3 Risk of Fraud

Any computerized system carries the risk of fraud. An inventory management system is no different. The system could be hacked at the gain of someone looking to benefit from a company that relies on its management system to conduct its day to day business.

4.8 Areas of further research

Artificial intelligence could play a huge part in the future of Management and POS systems. The technology consolidates, standardizes and enriches data, providing the foundation for AI analytics to present data-driven recommendations that managers can either chose to accept, reject or revise[18] While it has already been implemented in a very small number of systems,

automatic stock ordering and forecasting using AI could be beneficial in helping owners run the business. Menu decisions could be made by the system which would take all the data it has access too and make well-informed decisions based on such data.

While this could be deemed an area of further research it could also be deemed an area of controversy. AI could take the decision making out of the hands of the owner and put it into the hands of the technology. Eliminating the human decision-making process could affect companies. While there are great strides in computing power you would always need a person to make the final call. Eliminating human interaction would be impossible as sales, stock, and money levels will always have to be cross-checked against any information a management system has. This leads to the question, is AI in inventory management an inevitability or a luxury we don't need?

4.9 Users

Business Management Systems are everywhere. You can easily spot these systems in areas such as shops, bars, clubs, and restaurants. They are needed by every business with a considerable amount of daily sales. While our Management System is intentionally built for restaurants it can be easily incorporated into any business. The software can easily be adopted anywhere there are needs similar to our system with no tweaking at all. This has been achieved by carefully reviewing different models and usability of the proposal in different case studies.

As a result, the system was designed to be flexible in its application thus allowing anyone to use it in any business premises. With all this said and done, the users of this particular system would-be owners, management, waiting for staff, chefs, and bar staff.

4.10 Use Cases

The following use case illustrates the usability of the software based on the users identified above. While the use of the system may vary, we have listed the predominant features of the system which are universally essential. This

further illustrates the importance of incorporating the functionality of POS systems and RMS into one. Not only does it provide grounds for reusable codes it as well provides grounds for upgrades involving:

- blockchain and the Internet of Things.
- Owner Management and Till Application
- Management Management and Till Application
- Waiting Staff Restaurant Application
- Chefs Kitchen Ticketing Application
- Bar Staff Till Application

4.11 Project Platform

A restaurant management system must have the ability to work on any machine regardless of settings and Operating System. We therefore built this system as a standalone system which could be installed any machine. This improves on re-usability of the code while easing the task of upgrading in the future. As illustrated above, we used PHP as the main language for the project. PHP is platform independent and runs on any Operating System. When deciding on the language, we reviewed the ease in which it is maintained, its performance and scalability. These were the deciding factors in the project review. PHP therefore was selected as the best language to learn and incorporate into the project based on our timeline and project requirements.

4.12 Project Software

The following is a list of the software we used for our project and why we decided to use it. For the most part we put together a list of what we needed at the start of the project and decided on what piece of software we would use for each specific job.

4.12.1 Xampp

XAMPP is a free and open-source cross-platform web server solution that we used to develop this project [10]. Throughout our four years in college we predominantly used WAMP for our web servers. While WAMP would be efficient enough to build on our Windows laptops we wanted to use a server that was cross platform so our program could be run on any Operating System. XAMPP offered us exactly this so it was the obvious choice.

4.12.2 Visual Studio Code

For our code editor we used Visual Studio Code. VS Code is a light weight software with an embedded Git control which is the main reason we chose to work with it [5]. We also looked at the idea of using sublime text but VS Codes reliability along with its ease of file creation made it our first choice. VS Code has everything we needed to build this project including Git integration and in-built terminal for an all in one experience.

4.12.3 MySQL

MySQL is an open-source relational database management system [19]. Its reliability and high performance were among the reason we chose it, along with the fact we hadn't used it in college in a while, so we decided to brush up on it in final year.

4.12.4 LaTeX

LaTeX is a document preparation system we have used for write-ups in college and was suggested to us for the write up of our dissertation [15].

4.13 Plugins

The following is a list of plugins used as part of our project. These were all decided on as the use of bootstrap makes the integration of these easy. Each plugin provides an added piece to the GUI of the project to make it more use-able and aesthetically pleasing.

4.13.1 AdminLTE

For the main user interface and design (Homepage) of the project we decided to use AdminLTE. AdminLTE is an open source administration dashboard template [1]. It is built on top of bootstrap which made it easy to adapt and shape to match our needs and goals. We chose this as it is the most popular administrative tool out there and is integrated and customized easily.

4.13.2 Ion-icons

Ion-icons are bootstrap icons we used for table icons throughout the project. Having used them in previous projects we knew they offered an added sense of professionalism and make the project look aesthetically pleasing [12].

4.13.3 Bootstrap

Another plugin used previously Bootstrap is an open-source CSS framework that we used for the buttons and used as part of AdminLTE for tables and forms [4].

4.13.4 Font Awesome

Again, this was used on last year's project, so we decided to use it again. It is a font and icon toolkit based on CSS [2].

4.13.5 Bootstrap data tables

We used the Bootstrap Data tables plugin for all of our tables. This plugin allowed for the sorting and searching of tables items without the need for any extra code [8].

4.13.6 jQuery

jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax [13].

4.13.7 jQuery Number

Used to format numbers to two decimal places. This plugin was decided on after we ran into issues with the formatting of numbers in the till area of the project and was not decided on at the beginning of the project.

4.13.8 Fastclick

Another plugin that was decided on during the project build, FastClick is a simple, easy-to-use library for eliminating the 300ms delay between a physical tap and the firing of a click event on mobile browsers. During research of plugins that might be used on projects such as ours we came across FastClick and decided to add it to our project to speed up the ordering process.

4.13.9 Sweet alert

A responsive, customize-able and accessible (WAI-ARIA) replacement for JavaScript's popup boxes. This wasn't a plugin we used previously but when researching pop-ups that would enhance the user interface we decided on sweet alert as it was easily incorporated into our project.

4.13.10 TCPDF

TCPDF is a PHP library for generating PDF documents on-the-fly. This plugin was essential in the development of generating receipts as a PDF, we decided to go with this to PDF plugin as it is now one of the world's most active Open Source projects for PHP to PDF.

4.13.11 Date Range Picker

Date Range Picker is a JavaScript component for choosing date ranges, dates and times. An essential plugin which we used to navigate to past reports and hone in on certain dates to be able track how sales have improved or went down over time. This was chosen as we felt it was of utmost importance for the user to be able to backtrack their sales logs.

4.13.12 Morris.JS Charts

Morris.js lets the user create aesthetic charts in next to no time, it is made very simple using the public api for each chart, We decided to go with this as it was shown as a chart in the AdminLTE dashboard that looked exactly what we were looking for. We used Morris.js to create line and bar charts

4.13.13 Chart.js

Like Morris.js, Chart.js is a free open-source JavaScript library for data visualization, Chart.js was also found on the dashboard we decided to go with this plugin as it's beautifully constructed charts such as their animated pie chart is what struck out to us.

4.13.14 Input mask

Used to format input (email, numbers, and dates). Most popular one out there to format certain inputs so we decided to use it.

4.14 Measurable Goals and Requirements

The following is a list of requirements that were identified which we needed to incorporate into our project for it to be deemed a success. Functional Requirements:

4.14.1 Management

- Ability to display the real time stock levels
- Ability to display the sales of a given date range
- Ability to display the sales reports and export them as Excel sheet
- Ability to add new staff members to the system
- Ability to edit staff members in the system
- Ability to add delete staff members from the system
- Ability to add new categories to the system

- Ability to edit categories in the system
- Ability to add delete categories from the system
- Ability to add VAT and Tax to the system
- Ability to edit VAT and Tax in the system
- Ability to add delete VAT and Tax from the system
- Ability to add meals and drinks to the menu
- Ability to edit meals and drinks on the menu
- Ability to delete meals and drinks from the menu
- Ability to add customers to the system
- Ability to edit customers on the system
- Ability to delete customers from the system
- Ability to add customer discount to the system
- Ability to edit customer discount on the system
- Ability to delete customer discount from the system
- Ability to edit the stock levels
- Ability to alter the price of products

4.14.2 Staff

- Ability to view table details
- Ability to add a new order to a table
- Ability to add a new order without entering a table number
- Ability to add another order to an existing order
- Ability to delete a wrong order from tables
- Ability to add a new customer
- Ability to print customer receipts

4.15 Non-Functional Requirements:

4.15.1 Management

- Ability to interact with the database
- Only accessible by management staff
- Ability to search the database

4.15.2 Staff

- Ability to interact with the database
- Ability to order remotely with a touch screen tablet
- Ability to disable meals and drinks that are out of stock
- Ability to view full or discounted price
- Development of a color coordinated stock buttons to visually warn the user of stock levels

4.16 Conclusion

This review aimed to gain an understanding of what inventory management is and the great impact they have had on restaurants and other businesses on a technological scale. We looked into the history of inventory management as well as its advantages and disadvantages as well as the role management systems play on the day to day running of businesses. While not directly interacting with a management system it affects us all daily. Inventory management has its impacts on many facets of a company from its owners and staff to the suppliers and distributors and also the customers. It affects the goods we receive and their quality. It also affects the price we pay therefore affecting how we live. While we did mention some disadvantages, having a management system helps organizations, restaurants included, to maximize profits and avoid losses due to factors such as stock waste, overstocking, under-stocking, and customer dissatisfaction due to lack of desired products. We also touched on methods to improve current systems by looking at areas

of potential future development. Artificial Intelligence will play a big part in how these systems are managed in the future, but how big remains to be seen. All things considered, we can conclude that management systems such as the one we have built have given businesses the tools it needs to grow and prosper.

Chapter 5

System Design

5.1 Chapter Overview

This chapter will focus on the design of the system. Using the different types of UML diagrams and Graphical User Interface (GUI) images, we aim to illustrate certain sections of the software explaining in detail how they were designed. This section outlines this in different categories.

5.2 Model-View-Controller

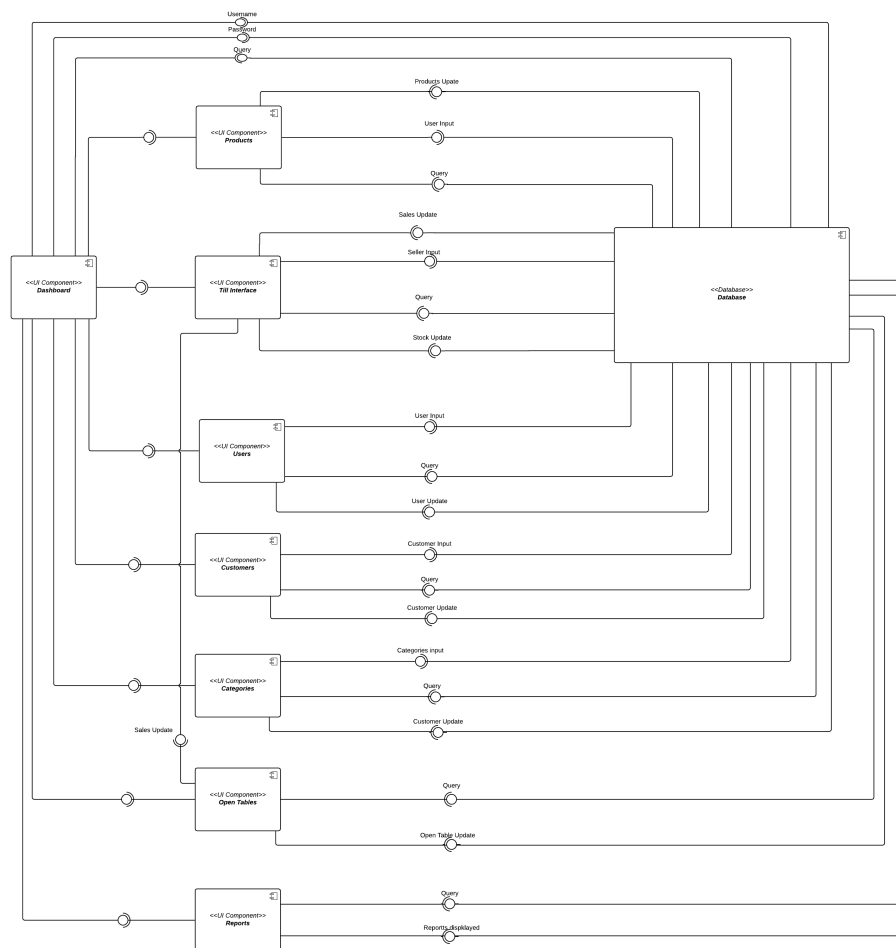
The Software design pattern that we chose was a Model-View-Controller(MVC). We felt it was best for the task ahead as the two of us had past experience using MVC design patter to develop work. We also believed that it would be perfect to connect the various parts of the project together while also keeping an internal separation for the representation of all of the different modules, finally we felt that since MVC is a pattern that is usually used in the designing layouts of the pages this would be essential in the presentation of the project.

5.3 Component Diagram

The component diagram in figure 5.1 shows the structural relationship between the components in the system. The diagram illustrates the usability

of the different components of derivatives of system requirements while installing the software.

Figure 5.1: Component diagram.



5.4 Data Storage

Data storage is a key part of this project. While choosing our database, we reviewed several storage solutions. This is a must-have in any system since we are dealing with offline complex data. The database choice is therefore critical for the preservation of data [22]. The following is a brief synopsis

of the storage solution we chose and why we decided to pick it: Relational Database Management System (RDBMS)

5.5 Relational Database Management System (RDBMS)

A Relational Database Management System (RDBMS) is a database management system based on the relational model of data. This particular type of database enables a user to access data based on its relation to another piece of data [6]. This ensures that is secure and a clear record of modification, read or write is maintained. It is fast and reliable and often used by some of the biggest companies in the world. It provides the developer with an option to use a structured query language (SQL) for querying and maintaining the database.

This is especially important since MySQL is an open-source programming language. The language is widely used by different developers and has been operation since 1995 [6]. This means that the code is widely re-usable and can get updates from different developers. It also assists the developers to raise a quorum on an impending challenge where they can get assistance from far wide in the information technology industry.

RDBMS's are often open-source and free to use. You can, however, opt to use paid solutions such as Oracle and SQL Server. The paid solutions versions are more utilitarian with options such as tech support. Oracle, for instance, is not an open-source software [7]. The database is expensive as well and is used often for huge amounts of data with very poor scalability. This, therefore, disqualifies these types of the database since they don't provide the necessary utilities for our project. Ideally, the paid solutions are unrestricted versions of the default open-source version.

RDBMS has several constraints that are beneficial to our project. This is; Not Null. This constraint ensures that there are no null/empty values in the columns. Unique. As the name suggests, the unique constraint is used to ensure all the data in the columns is unique. Check. The constraint is built to ensure that all the data in the individual columns meet a particular

predefined condition. Primary key. The primary key must be unique. This constraint must be unique since it is used to uniquely identify the different rows in a table. The key is also used to link a table to another. It, therefore, cannot have a null value. Foreign Key. This constraint is used to link two tables where one must behave the Primary key. Data Integrity. This constraint is used to ensure there is data integrity before modifying, writing, reading, or creating data [9]. There are several categories that a single task must pass to ascertain data integrity. The categories are; Entity Integrity, Referential Integrity, Domain Integrity, and User-Defined Integrity. [9] The above constraint contributed effectively to the advantages of the database. As a result, the RDMS database enshrines the following advantages:

5.5.1 Data Structure

As defined above, tables and columns are a defining factor in the database. This is an important figure since it provides the user with a natural space where data is structured and organized [23]. This makes it easy to structure queries to such through columns for matching entities.

5.5.2 Privileges

This feature is imported into our system. It assists the developer in categorizing access to sensitive information. The feature allows the database administrator to control activities in the database. This is achieved through integrity frameworks such as atomicity, consistency, isolation, and durability [17].

5.5.3 Data Recovery

Data loss is not a unique narrative in any information technology field. Data loss is, therefore, an impending risk while implementing any technological platform associated with data. RDMS leverage on this risk by providing the users with period backups of the system. This is preset by the developer to ensure the database periodically keeps a backup.

5.5.4 Flexibility

The database has been developed to accommodate different types of applications. This means that different applications based on different platforms can be used to access the platform. It further makes it easier for developers to handle big data.

5.5.5 Data Consistency

RDBMS databases are developed with a normalization feature. This ensures that all the tables maintain a proper structure of rows and columns. This also helps in sorting data where there predetermined methodologies of sorting data.

5.5.6 Speed

RDMS cannot be argued as the fastest database. The platform, however, provides several trades off that make it the best utilitarian for speed. RDMS provides a cheap data storage option. This means that by combining a large version of the same, the processor speed, decreasing memory requirements, and scaling makes RDBMS the most convenient database for the project. This is based on the tradeoff of the project as a whole [23].

5.5.7 Ease of Use

The platform supports MySQL. This is a very simple syntax language that eases the development work. The syntax used standard English language keywords and phrasing hence being very easy to use, learn, and intuitive.

5.6 Why RDBMS

SQL is an extremely flexible and robust querying language. When researching for a suitable query-based database, we come across RDBMS that provided all the utilities that we required. This was the best platform considering our application was largely a data computation and storage unit. WE, therefore, required a database that was easily designed while capable of handling complex systems. We, therefore, decided RDBMS is the best fit for our needs

The next step was to decide on the type of RDBMS to use. As mentioned above there are many open source RDBMSs available for developers. While there were rational versions of the same, we have used MySQL before and were aware of its capabilities. We, therefore, picked the same to manage the learning curve in the entire project.

While this was the case, there were several pointers that ruled out RDBMS as the best database option. This as listed by an online editorial include;

- Data Safety
- Fault-Tolerant
- Ease of use
- Scalability

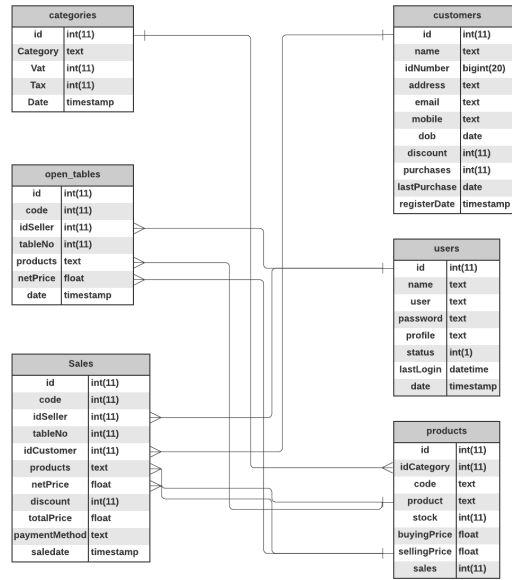
5.7 Entity Relationship Diagram

An Entity Relationship Diagram (ERD) shows the relationships of components stored in a database and is often used to represent a relational database and its requirements in a top-down view. We developed this diagram as a reference point considering the complexity of our project. Besides this, the diagram was developed as an industrial standard ensuring the code was readable to future users or introduced developers. The following ERD, figure 5.2, is the database schema for our POS system.

5.8 Database Design

At the start of the project, before a single line of code was written, a plan was put together for the database. The database is a key element to the system so the structure had to be designed well in advance of any coding. We had to make sure that the database captured all necessary information and was interconnected sufficiently to allow efficient querying. We also had to make sure the database was secure considering the environment that the software was to be placed in. Since the system was developed to manage the business as well as payments, the database design concepts were reviewed in different scenarios

Figure 5.2: Database schema.



Concepts of the database design are as follows;

- Ability to store product details.
- Ability to control the stock of the products when putting into an order.
- Ability to alter stock levels with new stock purchases.
- Ability to add and control customer details.
- Ability to add and control staff details.
- Ability to add and control category details.
- Ability to add and control the sale and open sale details.

5.9 Graphical User Interface

A major part of our system aims was to make it as efficient and as user friendly as possible. The graphical user interface (GUI) is the only component of the system that the user interacts with and therefore it is of great importance. The design had to be simple, clear, and concise but also have all of the required features needed at the users' fingertips. This is an important feature since it elevates the practicability of having features from two different platforms merged into one. We, therefore, had to make sure the two different components don't up complicate the system.

One of the main objectives of this project was to create a system that allowed users to complete orders in a fast and efficient manner. This was judged by the number of clicks required to get from the start of an order to the finish as well as the time taken to do so. Using AdminLTE, with a little tweaking offered us the ability to achieve our goals. The platform also enabled us to save time that would be spent developing the extensive user interface. Within our system, there are two different GUIs with each requiring a different design specification.

5.9.1 Till GUI

The system GUI is a critical part of this project. The usability and ease of usage, therefore, remained a part of the highest priority. The GUI is one of the main characteristics of the system. Through this portal clients would be required to make payments or orders for different assortments. The till portal was, therefore, design with the users' perspective in mind. We developed this section of GUI to be simplistic and intuitive. This means that a new user would have the easiest time figuring out how the portal works. We further made sure that the GUI did not require a lot of hardware resources which would end up slowing the entire system. We, therefore, developed several guidelines that would ensure we developed the correct till GUI.

The following is the specification for the till GUI:

Minimum clicks from start to finish. This would ensure that a new client would be able to easily navigate through the system without assistance. It would further ensure that a company is capable of maintaining its clients.

Every option is as little clicks away as possible. This would reduce the stress of constantly asking for help. It would also reduce the chance of clients finding the application cumbersome.

Adding products confined to half of the GUI while the payments process is confined to the other half. This ensures that the clients will have an easy time understanding the logic of the application. It further means that the clients will be able to navigate through the application by accessing portals that they require for transactions.

Accessing sales and open tables is in the sidebar of the GUI. This as well ensured that the clients did not have to scroll through different options to make an order. It further ensured that adding orders or modifying one was easy and convenient.

Ability to fit on any monitor size. Since practicability is one of our main objectives, we made sure that the system is scalable and compatible with different devices and operating systems. This enables clients with different preferential payments method to make payments from their different devices with ease. It also enables the developers to explore different platforms that the system can be upgraded to.

One-click payment method. This is one of the main objectives of the system. One-click payments reduce the amount of time and complexity of the system. Providing the user with complex computational power with the most friendly GUI. This improves on the operations at the different facilities while improving sales and management utilities.

show a design of the order the GUI uses to abide by the specifications listed above. Each part of the process is kept to one side of the till. Adding products on sale is situated on the right-hand side whereas the product list, table number, customer data, discount, and payment methods are all conducted on the left.

The sidebar allows the user to access sales and open tables to process table sales. Because of the need for use on tablets the system is designed to allow the GUI to fit all size screens. This allows the easy use of remote ordering.

Figure 5.3: Till GUI

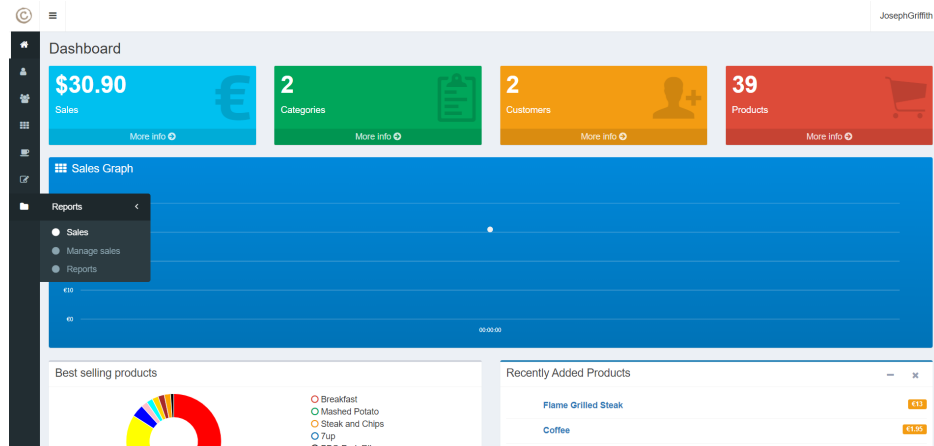
The screenshot displays the Till GUI interface. On the left, a sidebar contains navigation icons. The main area is divided into two sections. The top section shows a transaction summary for user 'JosephGriffith'. It includes a search bar with '2' entered, a list of items: '7up' (quantity 1, price € 2.00) and 'Beef Burger with Chips/wedges' (quantity 1, price € 12.50). Below this, a 'Discount' field is set to 0%, and the 'Total' is € 14.50. At the bottom of this section are buttons for 'Hold', 'Voucher', 'Card', and 'Cash'. The right section is titled 'Product' and shows a list of 38 items, with the first 10 displayed: '7up - 216', 'B.L.T. - 108', 'BBQ Pork Ribs - 111', 'Beef Burger with Chips/wedges - 106', 'Bowl of chips - 112', 'Bowl of Wedges - 113', 'Breakfast - 103', 'Buffalo Chicken Wings - 109', 'Burgers - 206', and 'Carlsberg - 205'. A search bar and a 'Show 10 entries' dropdown are at the top of this section. A pagination bar at the bottom indicates 'Showing 1 to 10 of 38 entries' with buttons for 'Previous', '1', '2', '3', '4', and 'Next'.

5.9.2 Management GUI

The management system requires a GUI to input data into various fields and therefore requires an appropriate input design. Using AdminLTE as our template we were able to re-use the same layout for each of the required fields (users, categories, products, and customers). The use of this platform enabled us to focus on the back-end ingenuity to provide the users with the best experience. Further, the platform enabled us to develop a complex GUI that is differentiated from the till GUI. The management GUI further was required to provide a detailed overview of the operations in the restaurant. This portal, therefore, required careful handling ensuring critical elements of the restaurant were easily accessible. Further, the management would have an easy time viewing the different entity values.

Figure 5.4 shows the design of the management application with the different functions appearing down the left-hand side as a tab. Each tab contains a sub-form with Figure 6.1/6.2 containing the structure of the generic data input form.

Figure 5.4: Side tabs.



5.10 Pricing

There are many factors throughout the system that affect the pricing of an item. The following is a list of the modules that affect pricing and how they all come together to make the price of a product: First of all, starting with categories, when adding a category, the owner/manager is asked to enter the percentage VAT and Tax rate of this particular category.

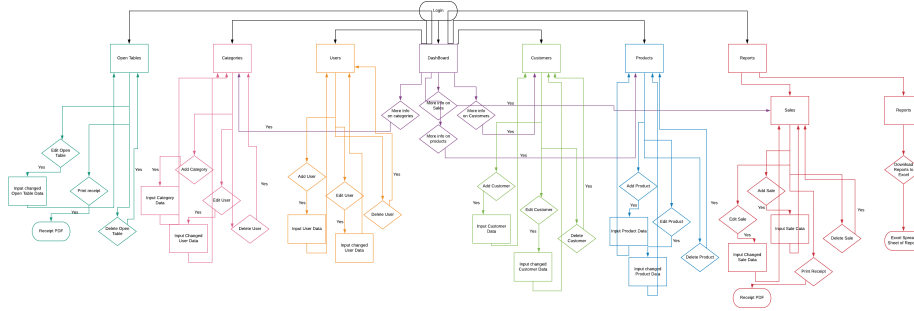
Secondly when adding a product the user chooses the product category and inserts the cost price. The VAT and Tax are automatically added to this price and are displayed to the user. The user then has the option of adding a percentage mark-up, chosen by themselves to the price or, they can choose to put their selling price into the system.

This process offers the user a lot of flexibility as well as efficiency. By just entering the cost price of a product the system computes this automatically. This process can save the user a lot of time spent calculating rates and mark-ups while reducing the risk of errors on their part. This further enables the management to work on different discounts and offers. By integrating the same in the system, this allows the user to feel more comfortable while making an order and to plan future visits to the restaurant. Using the system also enables the management to react to the demand in the market more easily.

5.11 Flow Charts

The following diagram is a flow chart of how the entire system works.

Figure 5.5: Flow chart of the entire system.



A flow chart is a diagram used to represent the process flow of an algorithm, problem, or some transaction within a business. Ideally, it is a diagram that depicts the process, system, and computer algorithm [16]. A flowchart is an essential tool for documenting the business system. It illustrates how we have managed to merge the POS and the RMS. It further enables developers in the future to improve or learn from the system. The flowchart has been designed to depict our system as it is indicating the different features as they are.

5.12 Ajax/Java Script

In the creation of the Restaurant Management system Ajax played an essential role in the development as it was used in unison with JavaScript in fetching, editing and deleting data from the database tables.

Here is a walk through of some of the functions that are used and what is called when creating, editing and deleting categories that relate to Ajax/Java Script in our system.

We initially use PHP calls from the model and controller where it adds data to the database. The controller asks for a response from the model as seen below to run the create category method which in turn adds a category.

Moving onto editing a category, when the edit category button is pressed the the categories.ajax.PHP file will fetch the category id for the specific category that is being edited from the database where the ID will then be used by the categories.js where once the fields are filled in the js file will send a POST to the database with the new edited information for the selected category.

Finally deleting a category starts off with pressing the delete category the categories.js file fetches the id of the desired category where once fetched the categories controller will take the category id and delete it in doing this the controller will once again send a request to the model to run the delete categories method, once successful the POS system will display a success message to the user.

5.13 Chapter Summary

This chapter covers system design in detail. Being a critical process in the entire build, the chapter details the same in the different sub-topics covering the entire planning, tools used consideration, and final structure. Diagrams have been used extensively in the chapter to illustrate the careful planning that ensured the project was a success. This includes a component diagram, entity-relationship diagram, till GUI, management GUI, and the flow charts. These diagrams further assist in the documentation of the project ensuring the entire build is represented and can be referred to for future research work or regular updates.

The chapter further covers important features such as data storage. This has been covered extensively including, the type of database that will be used, its selection criteria, and characteristics. All these variables assist in ensuring the success of the project. Further, the chapter explains the business end of the system, illustrating how different design concepts assist in ensuring the project is successful. There are factors such as GUI and price display. The GUI has been stressed since it is the only section of the system that clients will interact with. The GUI has therefore been developed to ensure the customers have an easy time interacting with the system. The pricing concept allows the management to indicate the price of different products and services. This presents the client with a friendly portal where they can

see the service or products that they opt for.

The chapter illustrates the importance of the management portal as well indicating how the complex back-end has been developed and simplified to a simple management GUI. This enables the business to be more streamlined by providing the management with clear statistical information about the enterprise. This chapter, therefore, illustrates the complex system design of the system and how it is integrated to ensure the business is successful. The next chapter, system evaluation documents in detail the expected performs of the system. The chapter will further extend the study and documentation of the system design indicating how the different selected tools and hardware contribute to the efficiency of the system. The chapter will evaluate the different design options take into consideration in this chapter illustrating their usability in a biased manner.

Chapter 6

System Evaluation

6.1 Chapter Overview

In the System Evaluation chapter we will touch on the overall design of the system while also comparing it to the project objectives that were stated in the Introduction chapter 2.9, the testing that went into the project will also be documented. We will finish off by reviewing any limitations and areas of improvement in the project

6.2 Project objectives

Initially we set out to create a universal web-based restaurant management and point-of-sale system that could be used on any premises that require a till system to process payments, print receipts, use a bar-code scanner as well as being able to generate reports. We felt that with the final product now complete we have achieved this goal as our web application is capable of all of the above, we feel it would be a great help to any site that needed a restaurant management and point-of-sale system. Other objectives included:

- User Authentication
- Till GUI
- Sales
- Receipts/Hardware

- Reports
- Users
- Categories
- Products

6.3 Testing

We felt that it was of utmost importance to fully test our web application in the development as well as in the completion of the project. Some of the testing we conducted included Graphical User Interface(GUI) testing, Functional testing, Unit Testing and finally End to End testing.

6.3.1 GUI testing

GUI testing is a software testing type in which tests the Graphical User Interface of the Project, This involves checking the screens with controls such as sidebar menus, buttons, icons , dialog boxes etc. The goal of GUI Testing is to make sure the UI works as designed.

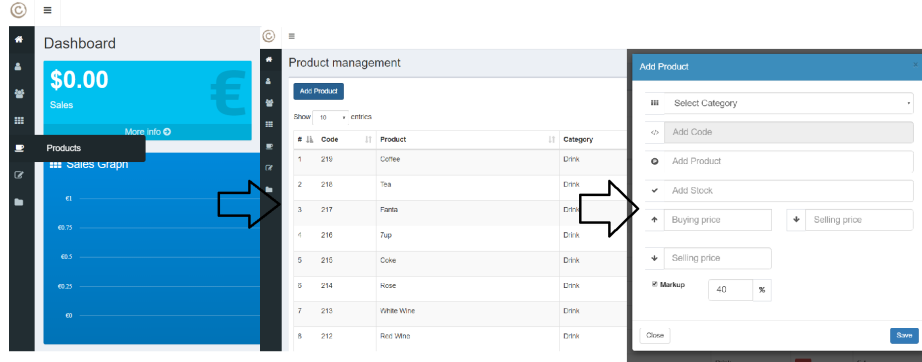
We decided to simply check some buttons and sidebar menu options in our bid to test the GUI functionality. In the following figure we show how the "Products" sidebar menu option leads to the products page then we test that the "Add Product" button works as designed.

As you can see both the menu option as well as the button work as designed as the Product options brings the user to the Products page while the Add products button directs the user to the creation of a new product.

6.3.2 Functional Testing

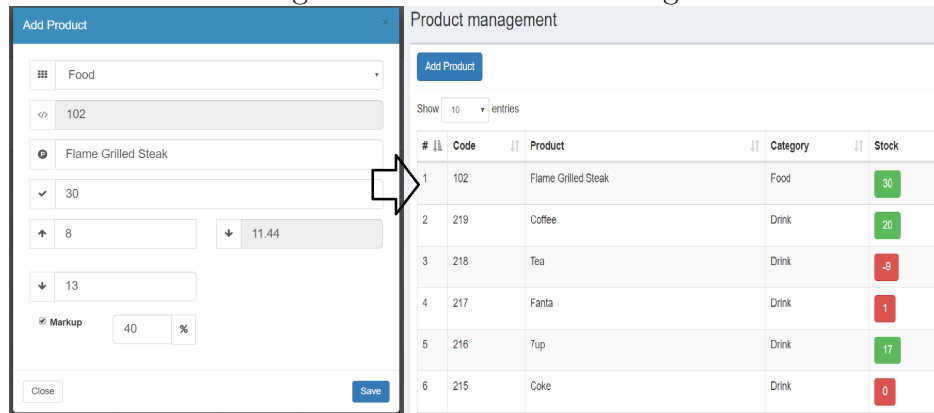
Functional testing is a type of software testing where you test the software system against the functional requirements. The purpose of one of these tests is to determine that the when inputting data into some parts of the software application the correct output is shown.

Figure 6.1: GUI Testing.



For our functional test we decided to stick with testing our products, once the add products button was clicked we moved onto inputting data to add a brand new product to verify that the functionality of add product was working as expected.

Figure 6.2: Functional Testing.



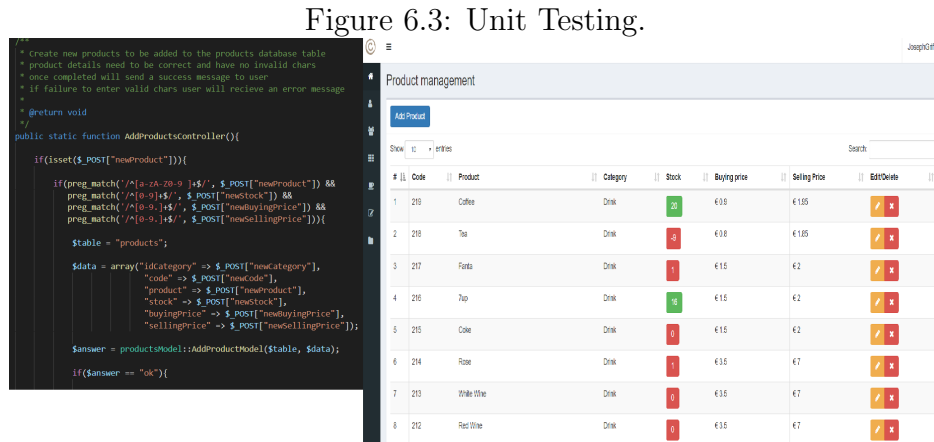
As displayed in the figure above you can see that once the user has input all the data needed for the creation of a product and clicks create product, the system will then redirect the user to the products page with

the newly created product displayed at the top of the Products table.

6.3.3 Unit Testing

Unit testing is the testing of individual units or components of a software. The reason that it is done this way is to validate that each unit of the software code performs as expected.

In our bid to do some unit test we decided to test the code that correlates to adding products to the database table.



As shown above the code snippet which is in charge of fetching and displaying the creation of the desired product and table displayed is working as intended.

6.3.4 End to End Testing

End to End testing is a type of software testing where the tester validates the software system along with its integration with external interfaces. The purpose to this specific testing is to test a production like scenario

In the end to end testing we just decided to go through logging in, creating a sale, printing a receipt then finally logging back out.

Figure 6.4: End to End Testing.

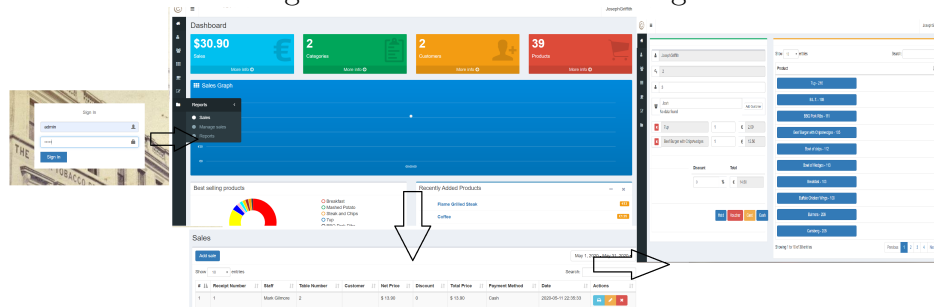
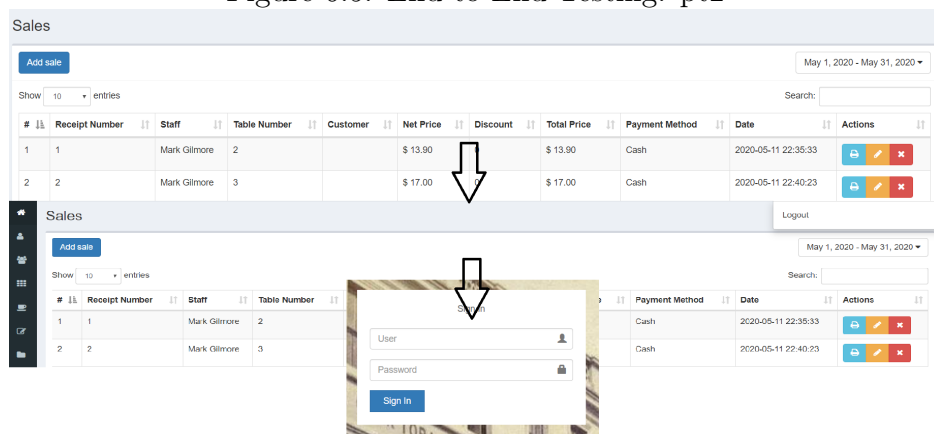


Figure 6.5: End to End Testing. pt2



As expected the system works as intended where the user can log into the system then create a sale, print the receipt for the sale then finally log out of the system.

6.4 Limitations and areas of Improvement in the project

6.4.1 Limitations

As of now the biggest limitation we are facing is not having the web application on the cloud this proves as a major set back as it is essential for us to have the application up and running on the cloud so that any businesses in the future would want to sample our software.

6.4.2 Improvements

Improvements that can be made to the current system could include adding a remote receipt printer, creating a loyalty system such as a loyalty code for returning customers where the customer would receive discounts for continued purchases or add item sales where some items on the menu would be on sale at different dates for example early birds deal where the customer gets a small discount on breakfast early in the morning.

Chapter 7

Conclusion

7.1 Chapter Overview

This final chapter is where we will conclude the project. We will talk once again about the aims and objectives of the project and evaluate whether or not they were met in the completion of the web application.

We believe we have achieved what we initially set out to create by developing a restaurant Point of Sale and Management System on the other hand we have also failed to meet all objectives as currently the restaurant Point of Sale and Management System is not hosted on the cloud resulting in remote order being impossible. Throughout the documentation we have shown the tools as well as methodologies that were needed in the developing the final product which allows the user to all the functionality that would be expected when using a restaurant Point of Sale and Management System. As listed in the introduction, objectives were defined :

- User Authentication
- Till GUI
- Sales
- Receipts/Hardware
- Reports
- Users

- Categories
- Products

7.2 Reflection

As mentioned earlier in the documentation System Evaluation, We are happy that we have achieved all that we initially set out to do. In conclusion we feel that we have provided an extensive report to the reader with insights into the technologies that were used in the development of our Web application along with the importance the technologies played in the development of the project.

7.3 Learning Outcome

Thought the development of this project both of us have acquired a large amount of knowledge on a variety of technologies and languages that went into the development of this project. Initially Php was foreign to us as it was something that we had never employed before, it was very challenging while also very rewarding learning new languages such as Php and Java Script. Java Script was another language which we felt our knowledge was inadequate with and was an extremely helpful to learn these two combined with using many of the plugins that were employed in the development of the project such as TC-PDF which allowed us to create PDF's with outputs of our database tables. The skills obtained in the development of this product such as the importance of working in a development team,we feel will help us immensely as we both move on in life as future software developers.

7.4 GitHub

Link to navigate to the GitHub where our project was developed:
<https://github.com/markgilmore10/Final-Year-project>

Bibliography

References

- [1] AdminLTE. *AdminLTE Bootstrap Admin Dashboard Template*. URL: <https://adminlte.io/>.
- [2] Font Awesome. *SDLC - Waterfall Model*. URL: <https://fontawesome.com/>.
- [3] T. Booth. *inventory management*. URL: <https://docplayer.net/7561467-Inventory-management-and-tracking.html>.
- [4] Bootstrap. *Bootstrap*. URL: <https://getbootstrap.com/>.
- [5] Visual Studio Code. *Code editing. Redefined*. URL: <https://code.visualstudio.com>.
- [6] codecademy. *What is a Relational Database Management System?* URL: <https://www.codecademy.com/articles/what-is-rdbms-sql>.
- [7] Oracle database. *Gartner Recognizes Oracle as a Magic Quadrant Leader in Database Management*. URL: <https://www.oracle.com/database/>.
- [8] DataTables. *DataTables*. URL: <https://datatables.net/>.
- [9] Educba. *What is RDBMS?* URL: <https://www.educba.com/what-is-rdbms/>.
- [10] Apache Friends. *What is XAMPP?* URL: <https://www.apachefriends.org/index.html>.
- [11] M. Georgiou. *User Experience Is the Most Important Metric You Aren't Measuring*. URL: <https://www.entrepreneur.com/article/309161>.
- [12] Ionicons. *Beautifully crafted open source icons*. URL: <https://ionicons.com/>.

- [13] jQuery. *What is jQuery?* URL: <https://jquery.com/>.
- [14] K. Kuligowski. *Restaurant management systems help you manage sales, staff, inventory and more.* URL: <https://www.business.com/articles/restaurant-management-system-guide/>.
- [15] L^AT_EX. *LaTeX – A document preparation system.* URL: <https://www.latex-project.org/>.
- [16] Lucidchart. *What is a Flowchart.* URL: <https://www.lucidchart.com/pages/what-is-a-flowchart-tutorial>.
- [17] M. and Anas. *What is RDBMS and How it is better than DBMS?* URL: <https://medium.com/@anas18/what-is-rdbms-and-how-it-is-better-than-dbms-c25b211454d4>.
- [18] Melanie. *AI.* URL: <https://www.unleashedsoftware.com/blog/how-inventory-control-can-benefit-from-artificial-intelligence>.
- [19] MySQL. *MySQL.com.* URL: <https://www.mysql.com/>.
- [20] Tutorial Points. *SDLC - Waterfall Model.* URL: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm.
- [21] G. Rachiele. *Software Development Methodologies.* URL: <https://medium.com/@gianpaul.r/software-development-methodologies-a856883a7630>.
- [22] M. Rouse. *Data Storage.* URL: <https://searchstorage.techtarget.com/definition/storage>.
- [23] D. L. Soltesz. *The Advantages of a Relational Database Management System.* URL: <https://www.techwalla.com/articles/the-advantages-of-a-relational-database-management-system>.
- [24] w3schools. *PHP Tutorial.* URL: <https://www.w3schools.com/PHP/DEfaULT.asP>.
- [25] C. Weisbach. *The Importance Of Improving User Experience.* URL: <https://www.forbes.com/sites/forbesagencycouncil/2017/06/15/the-importance-of-improving-user-experience/#50f474762b48>.