



BSc (Hons) Computer Science

**Customer Relationship Management(CRM) Software Package for
SME**

**Jolene Frances Sweeney
B00285373**

23rd March 2018

Supervisor: Mark Stansfield

Declaration

This dissertation is submitted in partial fulfillment of the requirements for the degree of Computer Science (Honours) in the University of the West of Scotland.

I declare that this dissertation embodies the results of my own work and that it has been composed by myself. Following normal academic conventions, I have made due acknowledgement to the work of others.

Name:

JOLENE FRANCES SWEENEY

Signature:

Date: 23rd March 2018

Form to accompany Dissertation

To be completed in full

Surname Sweeney	
First Name Jolene Frances	Initials JFS
Borrower ID Number B00285373	
Course Code	
Course Description	
Project Supervisor Mark Stansfield	
Dissertation Title Customer Relationship Management(CRM) Software Package for SME	
Session 2017/18	

**Please ensure that a copy of this form is bound with your dissertation
before submission**

COMPUTING HONOURS PROJECT SPECIFICATION FORM
(Electronic copy available on Moodle Computing Hons Project Site)

Project Title: Customer Relationship Management (CRM) Software Package for SME

Student: Jolene Frances Sweeney

Banner ID: B00285373

Supervisor: Mark Stansfield

Moderator: Carolyn Begg

Outline of Project: Develop a cost effective template CRM software package that can be customised for SME businesses to suit their requirements.

Design and implement Customer Relationship Management(CRM) Software for Small-to-Medium Enterprise's(SME's). This project will be supported by research into the various methods in designing and implementing a custom CRM in the most effective way possible for a SME. Customer Relationship Management(CRM) is a process that can be adopted by any customer facing business whether that be direct customers, indirect or even business to business. To enhance that process there are various CRM software packages available to businesses that allow them to electronically store any data or information relevant to their customer relationship management.

With the increased usage of CRM systems with businesses, for SME's to keep up with their competitors, a cost effective but useful solution, is required to implement a CRM. Most vendors market CRM systems pre-designed with general conceived ideas of what a business type needs.

In theory the process of designing a CRM system can follow a set pattern to fit any company. Where this may work for larger enterprises with the budget and resources needed it may not work in a start-up or small to medium enterprise. Throughout this project research will identify areas of the process that may need adapting when developing and implementing within an SME context. Further exploration of what the SME requires from the CRM and how that fits with the process of design, development and implementation to ensure the most appropriate solution is achieved for the SME.

Software package will include but not be limited to:

- Database to manage customer records for SME
- Website user interface to allow the users to access the software package from anywhere
- Application tools to extract data to analyse for business trends etc.

A Passable Project will:

- i. Carry out a literature review relevant to how design, develop and implement a CRM solution.
- ii. Develop a basic CRM system based on typical SME requirements.
- iii. Evaluate some basic elements of the CRM package with an SME.

A First Class Project will:

- i. Conduct a more in-depth review of relevant literature and a critical understanding of using the key issues associated with developing a CRM solution with an SME context.
- ii. Conduct detailed requirements and design analysis with a SME in relation to identifying user needs and requirements, exploring problems and issues of undertaking requirements and design within an SME context.
- iii. Produce a fully working and integrated CRM solution for an SME that can be customised and accessed via a custom designed company website with social media link up
- iv. Provide a detailed evaluation of the CRM solution addressing some of the issues that are identified by the SME client.
- v. Provide a set of recommendations aimed at helping address some of the key issues, problems and challenges of developing a CRM solution within an SME context.

Reading List:

Coleman, G. and Verbruggen, R. (2017). A quality software process for rapid application development. *Software Quality Journal*, [online] 7(1998), pp.107-122. Available at: <https://pdfs.semanticscholar.org/47ec/a10e940fc16e2909d3c77bc26fdaea8f480f.pdf> [Accessed 4 Mar. 2017].

Charland, A. and Leroux, B. (2011). Mobile application development. *Communications of the ACM*, 54(5), p.49.

Li, C. and Ma, L. (2015). Operation Design of Customer Relationship Management System. 2015 8th International Symposium on Computational Intelligence and Design (ISCID), [online] pp.536-536. Available at: <http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=7469191> [Accessed 6 Mar. 2017].

Shull, F., Singer, J. and Sjøberg, D. (2010). *Guide to advanced empirical software engineering*. 1st ed. London: Springer, pp.285-311.

Resources Required:

- *SQL Server*
- *JetBrains Webstorm Development Software*
- *SQL Management Studio*
- *Laptop*
- *Web Browser*

Marking Scheme:

	Marks
Introduction	10
Literature/Background Review	15
Requirements and Design	15
Development	30
Implementation and Evaluation	20
Recommendations/Conclusions	15
Critical Appraisal	5

Table of Contents

Acknowledgements	9
Abstract	9
Chapter 1. Introduction.....	10
1.1 Introduction	10
1.2 Target SME	10
1.3 Aims & Objectives	11
1.4 Justifications	11
1.5 Ethical Considerations.....	13
1.6 Report Structure	13
Chapter 2. Background Information/Literature Review	15
2.1 Introduction.....	15
2.2 What is CRM.....	15
2.3 Role of CRM Software	16
2.4 Current CRM Applications	17
2.5 Technical Details	18
2.6 Research Methodology	21
2.7 Developmental Approach	22
2.8 Implementation Platforms	24
2.9 Summary	25
Chapter 3. Requirements & Design	26
3.1 Introduction.....	26
3.2 Gathering User Requirements.....	26
2.8.1 Information Gathering.....	26
2.8.2 User Needs Identification	27
2.8.3 Envisioning and Evaluation	27
2.8.4 Requirements Specification	28
3.3 Requirements & Features	29
3.4 Use Case Diagram.....	30
3.5 Use Case Descriptions	31
3.6 Website Application Layout.....	37
3.7 EER Diagram.....	48
3.8 Relational Schema.....	50
3.9 Data Dictionary.....	51
3.10 Summary	59
Chapter 4. Development.....	60
4.1 Introduction.....	60
4.2 Development of Database	60
4.2 Development of Website	67
4.4 Connection of Database to Website	74
4.5 Summary	79
Chapter 5. Implementation & Evaluation.....	80

5.1 Introduction.....	80
5.2 Implementation.....	80
5.2.1 Test Cases	80
5.2.2 Testing.....	81
5.2.3 Implementation of CRM System	83
5.3 Evaluation.....	84
5.3.1 Evaluation by Company.....	84
5.3.2 Evaluation by Users.....	84
5.3.3 Evaluation of Software	85
5.3.4 Evaluation of Software Development Process	86
5.4 Summary	86
Chapter 6. Recommendations & Conclusions.....	87
6.1 Introduction.....	87
6.2 Recommendations.....	87
6.2.1 Recommendations on Software Development Process	87
6.2.2 Recommendations on CRM Software	87
6.2.3 Recommendations on Upgrading CRM Software	88
6.3 Conclusions.....	89
6.3.1 Handover of CRM System & Documentation	90
6.4 Summary	91
Chapter 7. Critical Appraisal	92
References	94
Appendices	97
Appendix 1.....	97
Appendix 2.....	98
Appendix 3.....	99

Acknowledgements

I would like to thank both my supervisor Mark Stansfield and moderator Carolyn Begg for their continuous support throughout this project it has been invaluable. The involvement of the target SME Seraphim has been the core of this project as without them this system wouldn't have been developed. Lastly, I would like to thank my partner Dionne Mair for supporting me personally throughout this time, I really wouldn't be where I am today without her.

Abstract

This report is an in-depth review of the process of designing and developing Customer Relationship Management (CRM) software for a Small-Medium-Enterprise (SME) in a cost-effective way. It looks in to the background of a CRM and any current solutions available on the market. The full design process of the software is documented in the chapters throughout this report from the design and development to the testing stages. There are recommendations on the software development process, the software itself as well as how to upgrade the software in the future. By the end of this report it concludes the cost-effective way in which this solution can be implemented for a SME.

Chapter 1. Introduction

1.1 Introduction

The purpose of this project is to design and implement Customer Relationship Management(CRM) Software for a Small-to-Medium Enterprise(SME) for which a target company has been chosen. This project will be supported by research into the various methods in designing and implementing a CRM.

Customer Relationship Management(CRM) is a process that can be adopted by any customer facing business whether that be direct customers, indirect or even business to business. To enhance that process there are various CRM software packages available to businesses that allow them to electronically store any data or information relevant to their customer relationship management.

With the increased usage of CRM systems with businesses. for SME's to keep up with their competitors a cost effective but useful solution is required to implement a CRM. Most vendors market CRM systems pre-designed with general conceived ideas of what a business type needs.

The aim of this research is to discover a solution that allows the design and implementation of a CRM system within a SME that is customizable and cost effective. As most SME's operate on a smaller budget than your larger enterprises, cost is a huge factor when implementing systems even if the system can prove to be effective to the business.

1.2 Target SME

The chosen target SME for this project is Seraphim which is a start-up SME that works in the employment sector as an agency for zero-hours contract employees. As a start-up company, there is zero funds to develop a CRM therefore they are an appropriate candidate for this project. The development of this CRM will allow the SME to start running their business and making it a profitable success in the future. With no existing software in place this project will allow for a custom-built CRM designed alongside the SME and to fit the needs of each user group.

Seraphim will solely be managed off the CRM developed for them and will aim to supply staff initially in the Security, Catering and Events Crew industries. The company holds details of employers, employees and open jobs with management selecting appropriate staff to complete any open jobs. For the SME to run this business effectively they require a CRM to implement the storage, retrieval and management of such data as well as a portal for different types of users to access in a safe and secure manner.

1.3 Aims & Objectives

Aims:

- Determine that there is a cost-effective way to implement a CRM solution to the SME
- Design a CRM solution based on findings
- Implement CRM solution based on design

Objectives:

- Show that there is a role for a cost-effective CRM solution for a SME
- Compare current CRM solutions available on the market
- Design & implement CRM
- Implement complete CRM solution to SME

1.4 Justifications

This developmental area has been chosen for this research proposal because of the continuous growth of the Customer Relationship Management(CRM) system within a business software environment. According to a report by Capterra in 2015, “Predicted to be a [\\$36 billion market](#) by 2017, the CRM market is on pace to surpass Enterprise Resource Planning as the [largest grossing enterprise software sub-segment](#), and it is already the fastest growing type of business software” (Capterra, 2015). The CRM market share is dominated by 4 main companies/vendors: Salesforce, SAP, Oracle & Microsoft with the rest of the market share being held by other vendors as displayed in Figure 1.

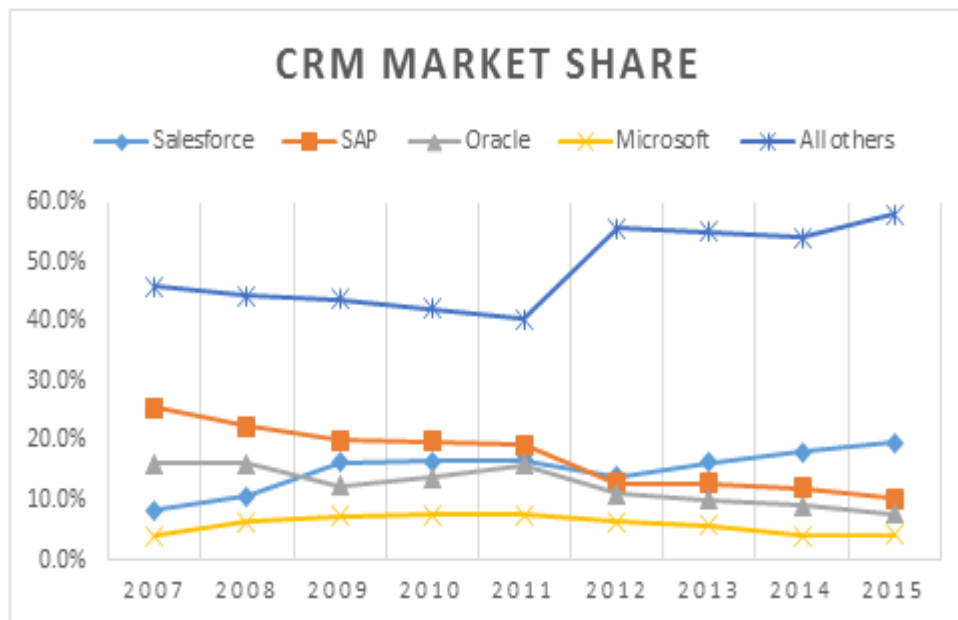


Figure 1. CRM Market Share Graph (Source: CRM Market Share Report 2017)

Other vendors can vary from smaller enterprises supplying off-the-shelf CRM software to an individual company employing a project team to develop a CRM specific to their business requirements. The various types of CRM software available reflects the various types of businesses that exist today creating a need for custom-built CRM software to suit a business's specific requirements.

Per author Hubbard 2013 he proposed that there are 5 main reasons to create a custom-built CRM software package for your business:

1. Custom Data Model

Most businesses operate a data model specific to their data requirements of the business and operate a multi-tenant database. A custom data model allows the data to be relational and highly accessible, incorporating business processes and functions.

2. Custom Workflows

A workflow typically is a step by step guide to a business process, and again most businesses adopt their own workflows depending on their business requirements. A custom workflow allows you to tether the CRM to be more specific to your day to day business activities and prevents any competitors from potentially following the same workflow as the business.

3. The right level of automation

Automation is the part of the CRM that takes care of behind the scenes tasks such as sending out newsletters to subscribers etc. this allows the business to operate efficiently with all behind the scenes tasks automatically taken care of. When using a vendor, who provides this automation without the ability to customise it there are occasions the automation could hinder the business and decrease efficiency instead

of increasing it. Therefore, a custom CRM allows the business to achieve the right level of automation for optimum operating efficiency.

4. Easier to use

Most off the shelf CRM systems layout etc. is generally decided and programmed by the vendor meaning that it may not necessarily suit the workflows, processes or needs of the business. With a custom system, the business can define how they want the layout etc. to look and feel like making it overall easier to use.

5. Killer analytics

Analytics reports on off the shelf software are generally a prescribed portfolio based on the most used reports over a variety of businesses. Not every business requires the same analytical data in a report therefore with custom software the business can specify which reports they require to be produced and how frequently they require them.

Along with various types of systems to implement there are varying costs depending on the system chosen. An overriding factor in any implementation of a system will be influenced by the budget available to the business. This again creates an increasing need to find cost effective solutions that can be applied across a variety of SME's who do not necessarily have a budget available to them.

1.5 Ethical Considerations

As this project is based on a real-world company there are a few key ethical considerations to take in to account throughout the project.

A CRM is based on a variety of organizational data that is confidential to several different parties. No data will be used for any other purpose than this project and will be held in a secure environment. If any of the information used for the purposes of this project is viewed by any parties outside the project would be a breach of the Data Protection Act and could result in prosecution under the current law applicable.

Any parties able to view or use this data will be authorized to do so and will follow strict guidelines otherwise if this is not possible "dummy" data will be used to allow the project to continue. Where the data is transmitted over the server this will be done in a secure way by employing various security and encryption techniques.

When any parties are being interviewed, observed, or assessed as part of this project they will be made fully aware and any consent gained expressly beforehand. Any data gathered from research following any of these activities will be kept under the guidelines of the Data Protection Act. Ethical consent will also be gained if any video techniques are used to observe the above activities.

1.6 Report Structure

The chapters following this introduction chapter make up the main body of the report and consist of the following:

- Chapter 2: Background Information/Literature Review which consists of an in-depth review of any background information and literature relating to this subject. Topics such as what a CRM is, current solutions available, technical details are discussed in length to ensure the project has a solid underpinning before being developed.
- Chapter 3: Requirements & Design which looks at the underpinning knowledge needed to capture requirements successfully for this type of project. It then discusses the stages conducted during the requirements and design phases of the project development process.
- Chapter 4: Development chapter considers the process followed during the development stages which is closely linked to the technical details section in Chapter 2. The chapter will describe in detail the development of all aspects of the full CRM system.
- Chapter 5: Implementation which discusses the process of implementation which for the purposes of this project the company did not want this done therefore the testing has been complete with advice offered on how to implement in the future.
- Chapter 6: Recommendations on the project are given with a conclusion to the report.
- Chapter 7: Critical self-appraisal conducted by the author on the overall process of this project.

Chapter 2. Background Information/Literature Review

2.1 Introduction

This chapter will discuss in detail background information that will assist in making this project a success as well as an in-depth review of current literature. Information on what a CRM is and what the role of CRM is within business environments is reviewed. A sample set of current CRM applications is investigated alongside any technical details that should be considered when developing the CRM system. Lastly, a review of relevant research methodologies appropriate to this project, developmental approaches and implementation platforms that will be required to complete this project.

2.2 What is CRM

CRM stands for Customer Relationship Management and is a combination of marketing techniques used by businesses to manage their relationship with customers. Its main purpose is to help build the relationships with customers by being able to manage their data, track interactions customers have with the business as well as being a central hub for customer account management. This process and design should help to improve customer relationships and in turn increase customer satisfaction, value and profitability.

According to (Salesforce.com, 2017) “More commonly, when people talk about CRM they are usually referring to a [CRM system](#), a tool which helps with [contact management](#), sales management, workflow processes, productivity and more. Customer Relationship Management enables you to focus on your organisation’s relationships with individual people – whether those are customers, service users, colleagues or suppliers. CRM is not just for sales. Some of the biggest gains in productivity can come from moving beyond CRM as a sales and marketing tool and embedding it in your business – from HR to customer services and supply-chain management.”

A CRM and a CRM system are separate entities with a CRM system being a tool used to implement CRM in a business. The components of a CRM system differ depending on the business’s needs or what is already in a vendor system, each of these components and sample vendor systems will be explored further on in this project. Customer relationship management regardless of how that is implemented can increase productivity when managing customers and allow that time to be spent focusing on the relationship of the individual customer.

Retrieving up to date information regarding your business can be a time-consuming task when a CRM is not being used as data can be stored in a variety of places and formats. A CRM can provide a clear overview of such information meaning customer data and interactions are managed from one central place.

A very common notion is that a CRM is just for sales but there is a proven record of various business types benefiting from a CRM within their business. CRM has evolved a great deal in recent years initially being aimed at the sales industry but the technology is now breaking through in to other industries as shown in figure 2 below.

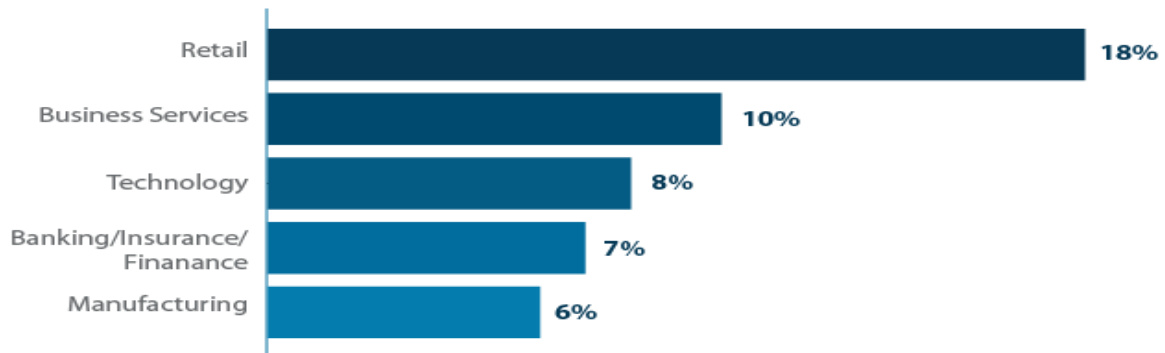


Figure 2. Top industries that use CRM software (Capterra, 2017)

Sales(retail) are still a large sector of the CRM market with them benefiting the most from a CRM to manage their sales process however the next four sectors are increasingly needing CRM solutions to manage their various processes which in nature can be more complex than a sales process. According to (Capterra, 2017) "It makes sense that accountants, consultants, software companies, insurance providers, equipment manufacturers, and the like would want to keep track of all their customer interactions over time, given that it usually takes many months and many people's input to make a large purchase like those listed above. And even once someone buys one of these products or services, a CRM is still necessary to track customer communications, help requests, contracts, invoices, and subsequent purchases."

To conclude the need for a CRM within a business has grown over recent years with more industries employing these processes. As such software to implement CRM within a business has become a large sector of the software market and the demand for existing and new solutions is increasing.

2.3 Role of CRM Software

The variety of data now stored by businesses has grown increasingly over the years, per statistics reported by LinkedIn there will be 4 billion connected people and 50 trillion gigabytes of data by 2020 (as shown in figure 3). This is a huge volume of data that must be managed by businesses.

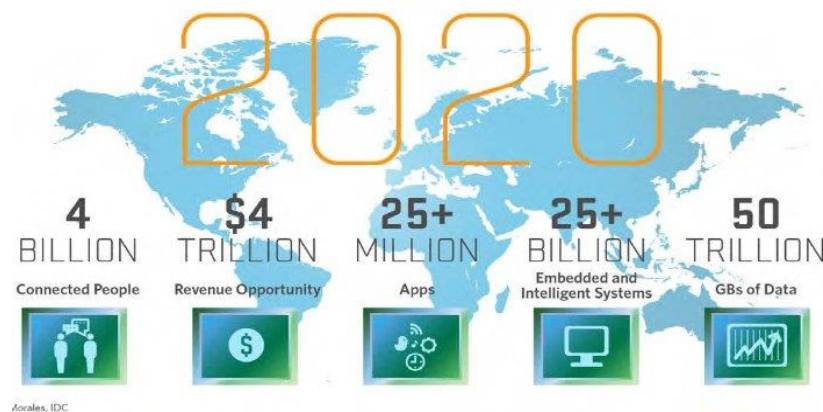


Figure 3. Data Centre Growth (Source: LinkedIn, 2016)

For this data to be managed in an efficient manner CRM software can and should be deployed, allowing a business to store, analyze, and manipulate their data to provide various tools such as reports, services, marketing etc.

The role of a CRM can be approached from two ways, theory and practice:

1. When viewed theoretically, CRM represents an operational strategy and management concept of modern enterprises. It is closely associated with corporate culture, emphasizing customers as one of the most important resources for an enterprise. The enterprise needs to provide sound customer service to meet the customer needs and improves its own profit margin through the provision of such service. (Li and Ma, 2015)
2. When viewed practically, CRM strives to establish a real-time, highly efficient and prompt enterprise-to-customer communication platform through optimizing and reorganizing enterprises' business procedures to develop a whole set of application software system based on modern computer network technology, information technology and communication platform, encompassing three major modules: marketing, sales management and customer service. (Li and Ma, 2015)

From this the role of CRM software can be defined as being a hub to customer communications ensuring relationships are managed in a professional manner, ensuring all departments of a business can communicate between each other and centralize all organizational information. This allows one department to see what contact another department has had with a customer, for example the complaints department can see that the sales department processed a sales order for a customer. This will reduce the amount of times information is lost or miscommunicated via the various communication channels a business may operate.

2.4 Current CRM Applications

There are many CRM applications available on the market today from various vendors, each of these vendors offer a variety of applications to adapt to suit a business. There are four main vendors in the market today some of which are:

- Salesforce
Global leader offering CRM technology in the cloud to a variety of businesses of all sizes enabling them to connect to their customers using the latest innovation technologies in mobile, social and cloud technology. Salesforce (Salesforce.com, 2017) offer a variety of different products depending on the business type with a tailored offering for small businesses. Their cloud based CRM system for small businesses offers several features such as:
 - Does not require any hardware or software making it available to all businesses regardless of their IT setup.
 - Offers 3 free upgrades each year to keep the software up to date, reducing maintenance costs.
 - Integration with other Salesforce products allowing the mix and match of products to suit the business needs.

Pricing plans range from:

- £20 per user per month for an off the shelf CRM for up to 5 users

- £60 per user per month for a complete CRM for any number of users
- £120 per user per month for a customizable CRM

If you have an organization of 10 users requiring a customizable CRM, costing £14,400 per year this can be a costly solution.

- SAP Anywhere

SAP Anywhere is a combined e-commerce and CRM software package that helps small businesses seamlessly manage marketing, sales, customer service, and inventory activities. Easy to use, this cloud-based front-office solution is available at an affordable subscription price (SAP, 2017). Features offered by SAP Anywhere include:

- Powerful CRM insights allowing a business to see all the ways a customer interacts with them.
- Omni-Channel experience providing a great customer experience across all channels.
- Mobile-first design allowing a mobile friendly application to be accessed anytime, anywhere.
- Cloud-based offered at a lower monthly cost and easy to set up.
- Painless integration with other systems and can be up and running within a few weeks.

Pricing plans:

- Starter plan - up to 5 users, 2 channels, 1GB of storage at \$534 per month and \$52 per user per month extra for additional users. One of start-up fee of \$1098.
- Plus plan - up to 10 users, 2 channels, 5GB of storage at \$908 per month and \$52 per user per month extra for additional users. One of start-up fee of \$1868.
- Premium plan - unlimited users, 5 channels, 100GB of storage at \$1818 per month. One of start-up fee of \$3,738

All plans include the ability to add extra channels at the cost of \$106 per channel per month.

If you have an organization of 10 users implementing this CRM solution in your first year will cost \$12,764 and for every year after that it would cost \$10,986

To implement any of these solutions would be at a huge financial burden to the organization with none of these solutions being purpose built for the organization. From these examples of solutions, we can conclude there is a gap on the market to build a custom-built CRM solution for a SME that is cost effective to the organization.

2.5 Technical Details

During the review of existing solutions currently available it has been identified that a cost-effective solution would be useful to SME's. There are many ways to keep costs down when developing software but still provide a useful and efficient solution.

To develop a relational SQL database to store organizational data for use by the CRM a DBMS system that can be used to create and maintain the database is required. Most DBMS vendors charge for their systems:

- Oracle Database Personal Edition - £74.00 (Shop.oracle.com, 2017)
- MySQL (now owned by Oracle) Enterprise Edition - £4,020 (Shop.oracle.com, 2017)
- Windows Server 2016 Essentials Edition - \$501 (Microsoft.com, 2017)
- MySQL Community Edition – Free (Mysql.com, 2017)

Each DBMS contains different advantage and disadvantages to using them, for the purposes of keeping cost as low as possible MySQL Community Edition which is a free software purchase is the choice of DBMS for this project. With the growing need of software being widely available across multiple platforms and operating systems, MySQL which is available on over 20 platforms allows this flexibility to be catered for.

For users to interact with the data and perform customer relationship management activities a CRM user interface is required, which again needs to be a non-native application that can be accessed from a wide variety of platforms. Every platform can run a web browser therefore a non-native web application will be developed to act as the user interface.

To develop a non-native web application, HTML5 will be utilized allowing the application to function on any platform that runs a HTML5 compliant browser. The European Commission state that, "Whereas all browsers – including Internet Explorer 7 and 8 – fully support HTML4 for several years, support for HTML5 varies very much among browsers. In Internet Explorer 7 and 8, support for HTML5 is very poor, whereas the latest version of Google Chrome, Mozilla Firefox and Apple Safari cover coherently most of the features offered by HTML5." (Ec.europa.eu, 2017)

The market share of browsers is split among various providers and these statistics have had a direct influence on the choosing of HTML5 standard to develop the CRM web application. As per the figures in table 1 Google Chrome and Mozilla Firefox hold the biggest share in the market which cover most of the HTML5 features offered.

Browser	Market Share
Google Chrome (latest version)	20% - 25%
Mozilla Firefox (latest version)	9% - 14%
Internet Explorer (version 7, 8, 9, 10)	15% - 28%
Internet Explorer 10	2% - 5%
Internet Explorer 9	4% - 9%
Internet Explorer 8	5% - 11%
Internet Explorer 7	2% - 3%
Safari (latest version)	2% - 5%
Mobile Safari (latest version)	2% - 5%

Table 1. Browser Market Share (Source: Ec.europa.eu, 2017)

JavaScript combined with PHP has been nominated as the preferred programming language for the user interface because it allows for interfaces to be created to improve user experience without having to wait for the server to reload and show another page (W3.org, 2017). Due to the CRM being used for the input and manipulation of data it is essential the web application can react to the user's interactions which will minimize data entry errors. As JavaScript is a web standard programming language it is compatible with other open web standards such as HTML and CSS.

It is extremely difficult to achieve a true cross-platform application as each platform comes with its own requirements to run certain features. Authors (Charland and Leroux, 2011) state that, "What makes things even more complicated are the differences among the actual platform SDKs (software development kits). There are different tools, build systems, APIs, and devices with different capabilities for each platform. In fact, the only thing these operating systems have in common is that they all ship with a mobile browser that is accessible programmatically from the native code." Due to a browser being the common thing not only across mobile platforms but any platform, a web application seems like the right choice when implemented a user interface that can work on any platform.

To implement a user interface using these technologies several different development environments are available to use such as JetBrains WebStorm, Komodo IDE and Eclipse. As the user interface requires little maintenance from the implementation stage there is more flexibility when choosing an IDE to develop the web application in. JetBrains provide a range of accounts available to students to allow access to their IDE allowing WebStorm to be used as the IDE for this project. This IDE provides support for JavaScript, Node.js, HTML, & CSS alongside a variety of frameworks. As shown in figure 4 WebStorm provides smart code completion reducing error rates, on-the-fly error detection, powerful navigation and refactoring. (JetBrains, 2017).

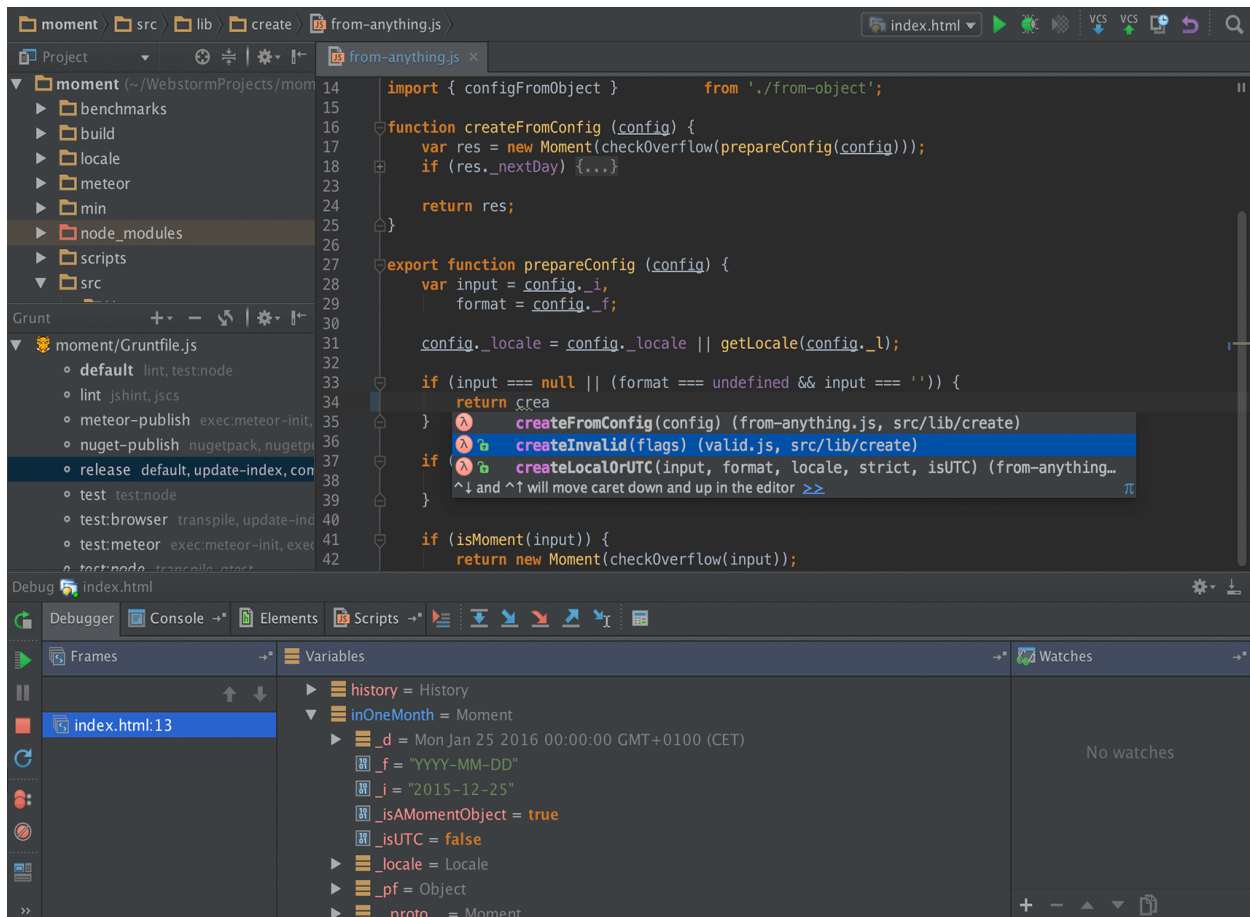


Figure 4. JetBrains WebStorm IDE Screenshot (Source: JetBrains, 2017)

2.6 Research Methodology

As the software, will be developed following the RAD developmental approach, there are several research techniques that can be adopted following a mixed methodology approach: Qualitative techniques

- Unstructured interviews with client and users to gather requirements
- Structured interviews with users to test features of software
- Observation of users to gain an insight into how the software is being used and any changes that could be made

Quantitative techniques

- Surveys to gain an understanding of the general features a SME would require in a CRM

As this research is being conducted to develop a CRM solution for a SME a mixed methods approach has been adopted to allow for the collection of a small sample of data primarily but the general features will be drawn from research conducted from a larger sample set. This allows a CRM to be developed that fits the general needs of a SME but can also be easily customized as will be displayed via the features implemented from the results of the qualitative research.

Shull, Singer and Sjøberg, 2010 believe that many of the research methods that are appropriate to software engineering are drawn from disciplines that study human behavior. Capturing user requirements is a study of human behavior and how the user interacts with the system. Therefore, the qualitative methods chosen, allow for these requirements to be explored in depth with a small number of people at their level.

Per author (Dybå et al., 2011) a main advantage of using qualitative methods such as interviews is that it allows the researcher to deal with the complexity of the issue rather than abstract it away, thus the results can be more informative. What one person may define as their user requirements may not necessarily be the requirements of another.

Unstructured interviews in the beginning allow for the user to freely discuss what they require at this stage of the project; this will allow the user to pass on their own ideas and opinions. Structured interviews however would be more useful in the testing stage of the project to allow a set criterion to be answered based on whether certain features are working in the manner that the requirements specified.

Observations also conducted at the prototype and testing stages of the project will allow findings to be made based on the actions of the users when trailing features of the software. As a user is generally not aware of the technical capabilities of the software, they may not always be able to communicate a requirement that would suit the business process but this may become obvious to the development team when the user is being observed.

Lastly focus groups conducted throughout each stage of the project will allow data to be gathered at a more larger scale from the users to gain a generalized concept of what they require from the system. Many features will be common to each employee and this will eliminate the need for repetition throughout the system.

Qualitative research techniques help to gain most information required from a small sample set of data and in this case a potential SME. However, again due to many features being common to each employee, many features are also common to many SME's. To gain a wider understanding of these features and requirements, a quantitative technique of surveys sent out to a larger sample set of data would be useful.

2.7 Developmental Approach

As with any software development there are many developmental approaches to choose from depending on what approach suits the needs of the project. The approach that will be adopted for this project is Rapid Application Development(RAD) due to its focus on developing quality applications faster allowing the project to run to its strict timescale but still involving the user.

RAD software development was first introduced in the 1970s and per (Naz and Khan, 2015), "In 1970 and onward years, traditional lifecycles were rigid. That result in poor and unusable systems. RAD is a solution for those rigid, stage wise or waterfall models of development." Traditional cycles follow a strict stage by stage approach which does not take in to account the ongoing changes throughout the project. Extracting user requirements from the client in

the initial planning stages generally only capture the concept of what the client wants. As the project progresses so do the user requirements therefore the planning stage is not a stage that can be complete entirely before the development stage begins, instead the planning stage after the initial phase is ongoing throughout the entire project. This will enhance the project efficiency and reduce the need to spend too much time on one phase.

Per a report published by (Coleman and Verbruggen, 1998) the advantages and disadvantages of RAD are as follows:

Advantages:

- Ease of implementation
- Improved user satisfaction
- Shorter time to reach the client

Disadvantages:

- Speed of development may be at the detriment to the design of the software
- Development team need to have more experience to produce software on time
- A strong control and management of the project is required to keep it on track

As the CRM is for a SME the RAD approach allows the software to go into production early on with little functionality to let the SME realize the benefit of implementing the CRM software to their business. Typically, these businesses may not have implemented such software before therefore by implementing the software early on with little functionality it can be built on bit by bit as the client gets used to each function. This will limit the impact on the business when the software is fully deployed as most functions will have been explored and used in advance with the most complex functions being the latter to implement.

RAD is divided up into four main phases as described below and shown in figure 5:

1. Requirements planning phase - Users, managers, and IT staff members discuss and agree on business needs, project scope, constraints, and system requirements. It ends when the team agrees on the key issues and obtains management authorization to continue. (Omics Group, n.d.) This allows the main requirements of the CRM to be agreed upon by all parties.
2. User design phase - during this phase, users interact with systems analysts and develop models and prototypes that represent all system processes, inputs, and outputs. (Omics Group, n.d.) This allows the users to be fully influential in the design of the software ensuring that the product is exactly as they require.
3. Construction phase - focuses on program and application development task like the SDLC. In RAD, however, users continue to participate and can still suggest changes or improvements as actual screens or reports are developed. Its tasks are programming and application development, coding, unit-integration and system testing. (Omics Group, n.d.) This allows for the work to go on and continuous updates to be

displayed to the user allowing changes to be made early on keeping the project on track.

4. Cutover phase - resembles the final tasks in the SDLC implementation phase, including data conversion, testing, changeover to the new system, and user training. Compared with traditional methods, the entire process is compressed. (Omics Group, n.d.) This allows the system to be built, tested and handed over in a much quicker time.

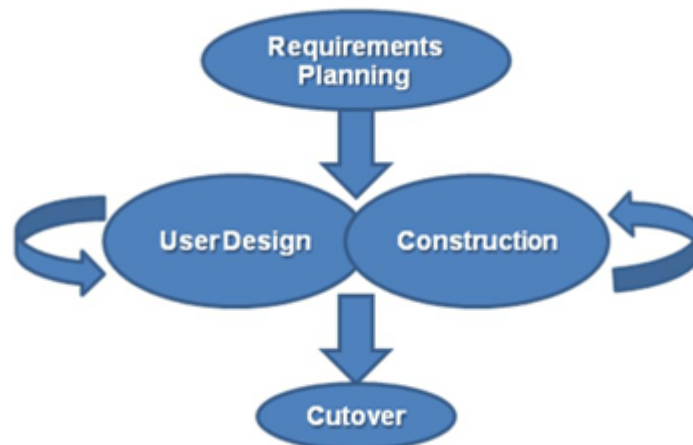


Figure 5. Rapid Application Development Process (Emersonprocessxperts.com, 2017)

2.8 Implementation Platforms

The proposed implementation platform for the storage of data will be a SQL relational database implemented via MySQL Community Edition. This has been chosen down to a cost related issue when developing the database as other DBMS platforms such as Oracle require the purchase of software to use it. As discovered in the Technical Details of this report MySQL Community Edition is currently free to download and use. The following are advantages and disadvantages of using a SQL relation database:

Advantages:

- Data is stored once in a table eliminating the need for multiple record changes and achieving more efficient storage.
- When one record is updated the change is reflected throughout any related tables
- Complex queries can be run on the data to allow the creation, insertion, modification or deletion of records. The queries can also be run to retrieve data and display it in several ways useful to the business
- Improved security due to the data being in tables, access to those tables can be limited to only those who require access.
- Easier to maintain the data and make any future changes to the data model.

Disadvantages:

- Can be expensive to implement and maintain due to the complex software and knowledge required. A business may need to hire personnel to implement the

database and purchase various security products such as encryption tools to secure the data being stored.

- Some fields are required to be limited in length meaning you need to specify in advance how many characters can be entered in to a field leading to a loss of data.
- The more complex the database and the queries being run the more powerful the hardware needs to be to process the data. A complex database may require a more powerful server to produce results in an efficient time.

For the CRM to be accessed by potential users, a user interface must be implemented and as proposed a Closed Web Application utilizing HTML5 & CSS and JavaScript will be implemented. This will limit the need for any specialist software being installed on a user's machine reducing the overall cost for the business and allows users to access the CRM software regardless of their location or machine being used.

The advantages of this approach mean that the CRM can be accessed from any machine that runs a HTML5 compliant browser which allows the business to deploy the software to a variety of employees including mobile workers. There will be a reduction in production costs as the software does not need to be produced on to any software installation materials. The software is easy to maintain and update, any updates to the software can be immediately implemented and it can be used on a variety of operating systems.

The disadvantages of this approach mean that the CRM cannot be accessed without a working internet connection, security can be compromised as the application is hosted via the web app and the web app must be programmed to work with the database.

2.9 Summary

To summarize this chapter has provided more than enough underpinning knowledge to allow the project to commence. This now allows the requirements and design phase to begin which will be discussed in great length in the next chapter.

Chapter 3. Requirements & Design

3.1 Introduction

During this chapter you will find the process of gathering requirements which has been fully researched in to the best methods on gathering such requirements. With adequate research being conducted and reported in this chapter, the process followed to carry out the requirements gathering has been documented. The design stage also forms the other half of this chapter with areas such as use case diagram, use cases, database design and website layouts being documented.

3.2 Gathering User Requirements

Gathering user requirements is a crucial part of any software process as it helps to understand exactly what the user requires the software to do. According to authors (Baxter, Courage and Caine, n.d.) requirements can come from any form of user and will be valid requirements and must be taken into consideration by any design team. To collect requirements correctly you must differentiate each type of user whether that be an end user or management user. User requirements are not set in stone they can change at any point throughout a project and as such the software should be able to support such changes. The gathering of such requirements can be crucial to the success of any interactive software design project.

2.8.1 Information Gathering

The first stage of gathering user requirements is the process of information gathering which is a stage to gather information about the users of the software, and any processes that currently take place. To gather such information many methods can be performed. One method that can be performed is stakeholder analysis.

The purpose of stakeholder analysis is to identify any users or stakeholders that could influence or impact the system in any way to ensure everyone's needs are met. According to authors (Varvasovszky, 2017) "Stakeholder analysis identifies for each user and stakeholder group, their main roles, responsibilities and task goals in relation to the system."

Another method of information gathering is context of use analysis. A system can be used differently by different types of users and as such the context in which they use the system can be different. According to authors (Varvasovszky, 2017) "The quality of a system, including usability, accessibility and social acceptability factors, depends on having a very good understanding of the context of use of the system." For example, depending on the user there are many things that can impede on how the user is using the system such as their environment, background distractions, disabilities, user workload etc. By conducting context of use analysis, you can capture this information in advance and ensure the system is usable to these groups of users.

A final example of a method for gathering information is task analysis. According to authors (Varvasovszky, 2017) "Task analysis involves the study of what a user is required to do in terms of actions and/or cognitive processes to achieve a task. A detailed task analysis can be conducted to understand the current system, the information flows within it, the problems

for people, and opportunities that indicate user needs.” Task analysis can be undertaken to discover how each task is broken down and performed by users. There are many ways in which you can do this such as hierarchical task analysis which breaks down high level tasks into smaller tasks which are more detailed. Another common method which can be used is flow charts which show the flow of each task in sequence of how the user interacts with it.

2.8.2 User Needs Identification

User needs can be identified to ensure that each need for each type of user is met, there are many methods to gather such information.

One method for identifying user needs is user surveys. User surveys can be conducted to gather a wide spread of information from few to many users and collation of the data can produce a general opinion on a pre-defined set of questions. The survey could be limited to a proportion of the user population or to all users directly if possible. Questions should be pre-defined and aimed at the needs of the user such as what is their role etc.

Another method for identifying user needs is interviewing. Interviewing allows for a wider scope of questioning techniques to be used to a smaller audience usually one to one. Questions can either be pre-set or can be free-flowing depending on how the interview goes, either way a structure for the interview should be established and interviewer should be careful about going off track when questioning.

A final example of a method for identifying user needs is use case diagrams & descriptions. Use case diagrams provide a high-level view diagram of the system (boundary box) with which tasks (use cases) can be inside the system for users(actors) to interact with. These diagrams are useful for explaining the use of the system in a non-technical visual manner. Use cases should always begin with a verb and describe an action and actors can be any user from a person to a system (one actor describes a group of actors such as manager, employee etc.). Use cases are further described in their use case descriptions where a pseudo-code of the use case is described including any pre-and post-conditions for the use case. Use case descriptions allow for a step-by-step guide to a use case to be detailed.

2.8.3 Envisioning and Evaluation

When needs have been gathered for the user, a proto-type or mock-up of the design should be created to give a visual representation to see how it is going to look and feel. There are many methods to achieve this.

One method to give a visual representation is to create storyboards. Storyboards typically show sequence of images that describe the relationship between the actions users take and any system inputs or outputs. According to authors (Varvasovszky, 2017) “A typical storyboard will contain many images depicting features such as menus, dialogue boxes and windows.” This method provides a visualization of user interactions etc. to allow for them to be presented to necessary stakeholders.

Another method to give a visual representation is prototyping which can be achieved in two ways:

Paper prototype

A paper prototype also commonly known as a mock up design can be used as a visualization of the layout to allow stakeholders to view how the software works and how interactions are responded to by the system etc. According to authors (Varvasovszky, 2017) "When a paper prototype has been prepared, a member of the design team sits before a user and 'plays the computer' by moving the paper and card interface elements around in response to the user's actions. The difficulties encountered by the user and user comments, are recorded by an observer. "This allows for changes to be made to the layout to make the difficulties encountered less difficult in the final release of the software.

Software prototype

A software prototype provides a working visualization of the software in a real-feel environment. It has a more realistic feel to it than a paper prototype and can provide rapid prototyping to establish an appropriate design with the user. According to authors (Varvasovszky, 2017) "Some design processes are based on a rapid application development (RAD) approach. Here a small group of designers and users work intensively on a prototype, making frequent changes in response to user comment. The prototype evolves into the full system." This process allows for the working application to be updated as it is worked on meaning the prototype evolves over time into a working application that the users have agreed upon.

2.8.4 Requirements Specification

When a requirement has been gathered fully it can be documented within a requirements specification. Many methods can be used to document it within a specification and ensure the correct information is recorded.

One method to document a requirement in a specification is requirements categorization. Categorizing requirements is important to establish which requirement is needed for which user and for which purpose it is needed. It can also be categorized in a priority order to ensure high priority requirements are developed first. Requirements can be categorized into categories such as user requirements, usability requirements etc. Prioritization can be particularly important to allow for effective resource management. Author (Varvasovszky, 2017) states that "The DSDM development method uses 'time boxes' where the functions and features in each phase of a system's release are defined by the resources available. This helps control the risks in system development, and allows the customer to redirect future effort to meet the user's needs more closely." If requirements are correctly prioritized it will allow for an effective system to be developed with the critical requirements developed first and the requirements of a low priority can be developed within time or planned as future development work.

Another method to document a requirement in a specification is task/function mapping. This process maps the tasks and which functions those tasks will involve. Author (Varvasovszky, 2017) states that "By showing the relationship between the tasks and the corresponding functional requirements linked in matrix form, trade-offs can be made between different functions, or to add and remove functions depending on their value for supporting specific tasks." Functions can be developed that support the high priority tasks

meaning that the development risks will be lower and the chance of a project success is higher.

3.3 Requirements & Features

During an open interview/discussion with the company a proposed list of requirements for the CRM has been gathered:

- Store data of employees including any vetting documentation such as 2 references, 5-year address history and disclosure documentation.
- Store data of employers
- Store data of past, present and future jobs and their outcome
- Website application user interface for users to access the CRM through
- System authentication by username and password
- New employees can register their details on the CRM
- Different levels of user profiles for management, employees and employers ensuring only the features appropriate to the profile are available.
- Messages centre that will have an inbox and sent items.
- Data can be updated through forms on the CRM to ensure it is being updated safely and securely.
- Secure storage and transfer of data

The CRM software will be accessed by 3 types of users with each user having different features available to them:

- Management User
 - Message centre will allow for communication between management to employees and management to employers.
 - Manage database through various forms such as update employee or employer details, issue employee warning, confirm job match, confirm registration of employee or employer, remove employee or employer from system, mark employee or employer as blacklisted
 - Analyse statistics of the data (details of which statistics will be needed to be confirmed later)
 - See past jobs
 - See past jobs that they have offered and their outcome (as a list)
 - See scheduled work
 - See scheduled work with employee details (as a list)
 - See unscheduled work
- Employee User
 - Message centre allows communication from employee to management
 - Update employee details through a form
 - See past work
 - See scheduled work

- See available work
- Apply for available work from available work calendar
- Employer User
 - Message centre allows communication from employer to management
 - Add a new job through a form
 - Remove a job through a form
 - Update employer details through a form
 - See past jobs
 - See past jobs that they have offered and their outcome (as a list)
 - See scheduled work
 - See scheduled work with employee details (as a list)
 - See unscheduled work

3.4 Use Case Diagram

Figure 6 below is the use case diagram for the Seraphim Customer Relationship Management System. The purpose of this diagram is to define user requirements as explained in Chapter 4 Section 2.7.

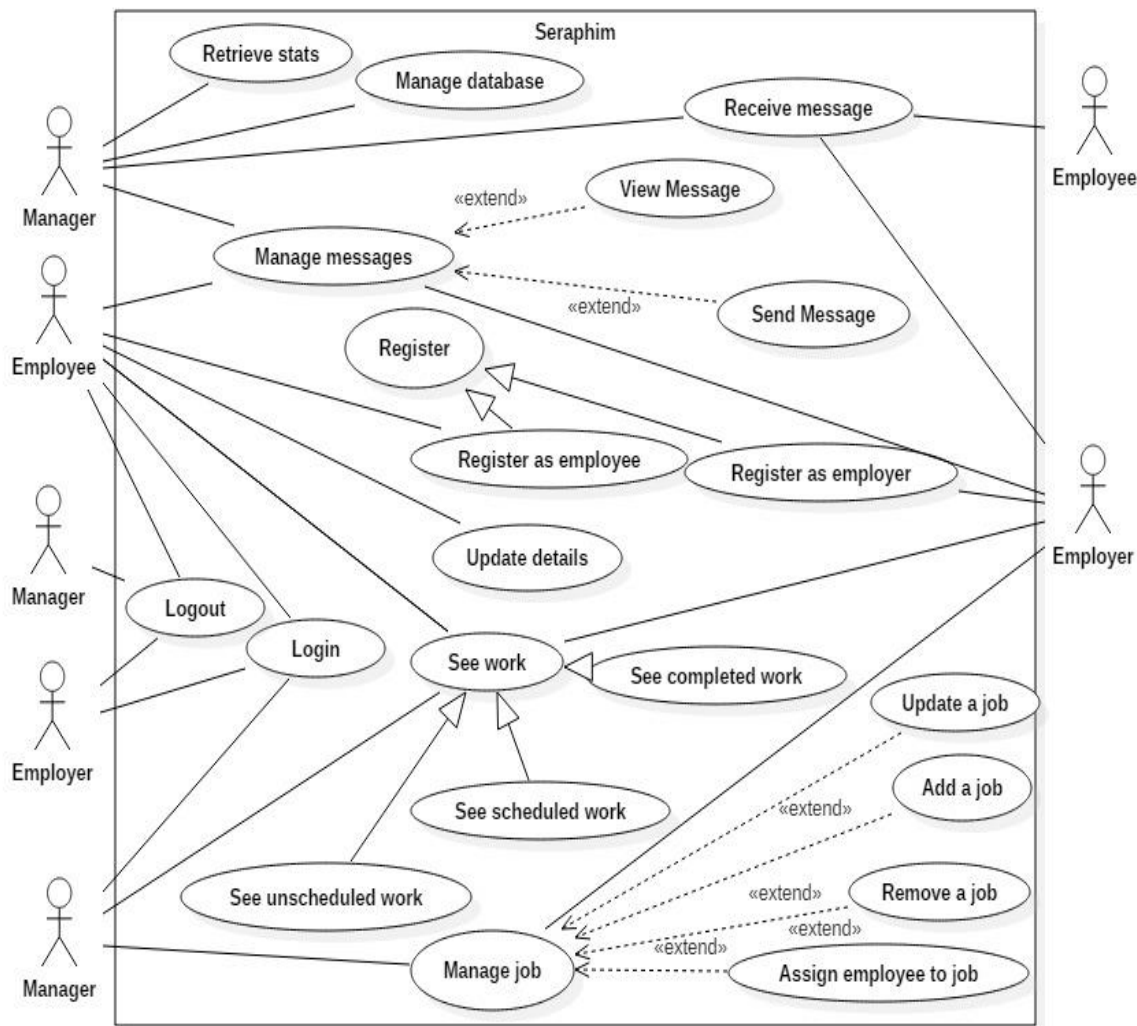


Figure 6. Sephora Use Case Diagram

3.5 Use Case Descriptions

Use case descriptions below describe the use cases in Figure 5 for the Sephora use case diagram. These descriptions define:

- Use Case Name
- Pre – Conditions (Anything that must happen for this use case to trigger)
- Post-Conditions (Anything that happens because of the use case being complete)
- Purpose (What the use case is for)
- Description (The steps of the use case in plain English)

Register Use Case Description

Use Case Name	Register
Pre-Conditions	Relevant button on website application has been clicked
Post-Conditions	Either employee or employer registration is complete
Purpose	Register a new user on the CRM
Description	New user clicks register button for relevant user IF user clicks register employee button THEN start Register Employee use case ELSE IF user click register employer THEN start Register Employer use case ELSE return error message Return to home page END

Table 1. Register use case.

Register Employee Use Case Description

Use Case Name	Register Employee
Pre-Conditions	Employee option selected during registration process
Post-Conditions	New record created on the database for employee table
Purpose	Allow new user to register their details on the database as an employee
Description	Employee registration form displayed User fills in registration form User clicks submit IF any mandatory fields are incorrect or blank THEN return to form and highlight areas for correction ELSE submit form END New record created in database with form details Pop up box announces user is registered successfully and can now log in.

Table 2. Register Employee use case.

Register Employer Use Case Description

Use Case Name	Register Employer
Pre-Conditions	Employer option selected during registration process
Post-Conditions	New record created on the database for employer table

Purpose	Allow new user to register their details on the database as an employer
Description	Employer registration form displayed User fills in registration form User clicks submit IF any mandatory fields are incorrect or blank THEN return to form and highlight areas for correction ELSE submit form END New record created in database with form details Pop up box announces user is registered successfully and can now log in.

Table 3. Register Employer use case

Login Use Case Description

Use Case Name	Login
Pre-Conditions	Must be a registered user
Post-Conditions	User logged in to system
Purpose	Users can log in to system
Description	User clicks login button Login page displays with login form User fills in username and password User clicks login IF username and password match username and password stored on database THEN user is logged in user dashboard is displayed ELSE error message is displayed user returned to home page END

Table 4. Login use case

Logout Use Case Description

Use Case Name	Logout
Pre-Conditions	User must be logged in
Post-Conditions	User is logged out of system
Purpose	User can log out of system
Description	User clicks log out button IF user is logged in THEN user is logged out of system homepage is displayed ELSE display homepage END

Table 5. Logout use case

Retrieve Stats Use Case Description

Use Case Name	Retrieve stats
Pre-Conditions	User must be a manager profile
Post-Conditions	Requested stats are displayed

Purpose	To display a range of statistics for management
Description	User clicks on retrieve stats tab Retrieve stats dashboard displayed User chooses from pre-set statistic options SWITCH option CASE 1: retrieve and display statistic option 1 CASE 2: retrieve and display statistic option 2 CASE 3: retrieve and display statistic option 3 Etc. END

Table 6. Retrieve Stats use case

Manage Database Use Case Description

Use Case Name	Manage database
Pre-Conditions	User must be a manager profile
Post-Conditions	Relevant changes are made to database
Purpose	Allows management to manage the database using preset queries to maintain the underlying integrity of the data
Description	User clicks manage database tab Manage database dashboard is displayed User chooses from pre-set manage options SWITCH option CASE 1: update employee details Display update employee form CASE 2: update employer details Display update employer form CASE 3: add employee form Display add employee form CASE 4: add employer details Display add employer form Etc. User fills in form user submits form While form contains errors IF form contains no errors THEN database updated ELSE return to form and highlight errors END END

Table 7. Manage Database use case

Manage Messages Use Case Description

Use Case Name	Manage messages
Pre-Conditions	User must be logged in
Post-Conditions	Message dashboard is displayed
Purpose	Allows user access to their message dashboard
Description	User clicks on message tab Message dashboard is displayed

	list all received messages IF view message is clicked THEN start view message ELSE IF send message button is clicked THEN start send message use case END
--	--

Table 8. Manage Messages use case

Send Message Use Case Description

Use Case Name	Send message
Pre-Conditions	Message dashboard must be displayed and user must be logged in
Post-Conditions	Message is sent
Purpose	Allows users to communicate with each other via sending messages
Description	Send message form displayed in new window User fills in form User submits form While form contains errors IF form contains no errors THEN send message to recipient ELSE return to form and highlight errors END END Close send message form window

Table 9. Send Message use case

View Message Use Case Description

Use Case Name	View message
Pre-Conditions	Message dashboard must be displayed and user must be logged in
Post-Conditions	Chosen message is displayed in new window
Purpose	To view a selected message from the user's message dashboard
Description	Open new window displaying users selected message IF user clicks reply button THEN start send message use case ELSE IF user clicks close message button THEN close selected message window ELSE window remains open END

Table 10. View Message use case

Receive Message Use Case Description

Use Case Name	Receive Message
Pre-Conditions	User must be registered
Post-Conditions	Message received on system and user notified
Purpose	Message is received on system and stored to be viewed when user logs in
Description	Message received on server from user Message stored in database IF recipient user is registered THEN store message in database

	END
--	-----

Table 11. Receive Message use case

Update Details Use Case Description

Use Case Name	Update Details
Pre-Conditions	User must be registered
Post-Conditions	Relevant data is updated
Purpose	Allows users to update their details stored within the database safely without affecting the data's integrity
Description	User selects update details tab Update details form displays User fills in form user submits form While form contains errors IF form contains no errors THEN database updated ELSE return to form and highlight errors END END

Table 12. Update Details use case

See Work Use Case Description

Use Case Name	See Work
Pre-Conditions	User must be logged in
Post-Conditions	List of work is displayed
Purpose	To display a list of work depending on option selected
Description	User selects see work dashboard See work dashboard is displayed END

Table 13. See Work use case

Manage Job Use Case Description

Use Case Name	Manage Job
Pre-Conditions	Must be either management or an employer user
Post-Conditions	Job management dashboard is displayed
Purpose	Allows employers and management to manage jobs within the system including adding new jobs, removing jobs or updating jobs
Description	User selects job management tab Job management dashboard is displayed User chooses from pre-set job management options SWITCH option CASE 1: Add Job start add job use case CASE 2: Remove Job Start remove job use case CASE 3: Update Job start update job use case END

Table 14. Manage Jobs use case

Add a Job Use Case Description

Use Case Name	Add Job
Pre-Conditions	Add job option must be selected during manage job option
Post-Conditions	New job is added to the system
Purpose	To add a new job to the system which is stored and available to be seen by the employer, employees and management
Description	New window is opened with add job form displayed User fills in add job form User submits add job form While form contains errors IF form contains no errors THEN database updated ELSE return to form and highlight errors END END Display job added message Close window and return to job management dashboard

Table 15. Add a Job use case

Remove a Job Use Case Description

Use Case Name	Remove Job
Pre-Conditions	Remove job option must be selected during job management option
Post-Conditions	Job is removed from system
Purpose	Allows a job to be completely removed from the system if it is cancelled before the job is scheduled to go ahead
Description	New window is opened with remove job form displayed User fills in remove job form User submits remove job form While form contains errors IF form contains no errors THEN IF start date of job is after current date THEN remove job from database ELSE display error message and return to job dashboard END ELSE return to form and highlight errors END END Display job removed message Close window and return to job management dashboard

Table 16. Remove a Job use case

Assign Employee to Job Use Case Description

Use Case Name	Assign Employee to Job
Pre-Conditions	Assign employee to job option must be selected during job management option and user must be a manager
Post-Conditions	Employee assigned to a job
Purpose	Allows management to assign an employee to an unscheduled job
Description	New window opens with assign employee to job form

	user fills in form user submits form While form contains errors IF form contains no errors THEN IF employee ID is available and not blacklisted THEN assign job Send message to employee and employer update database update job to scheduled ELSE display error message and return to job dashboard END ELSE return to form and highlight errors END END Display job assigned message Close window and return to job management dashboard
--	---

Table 17. Assign Employee use case

Update a Job Use Case Description

Use Case Name	Update a Job
Pre-Conditions	Update a Job option must be selected during job management options
Post-Conditions	Job details updated
Purpose	Allows management or employers to update the details of a job
Description	New window opens with assign update a job form user fills in form user submits form While form contains errors IF form contains no errors THEN IF job ID on form matches job ID in system THEN update job details Send message to employer and management update database ELSE display error message and return to job dashboard END ELSE return to form and highlight errors END END Display job updated message Close window and return to job management dashboard

Table 18. Update a Job use case

3.6 Website Application Layout

In the early stages of the design a prototype of each web-application page has been created to format the layout of items. This allows the company to have an input from an early point

about what the finished product will eventually look like as described in Chapter 2 Section 7. Detailed below is a sample of each page of the web-application and how it is proposed to look like from one user's perspective (manager, employee or employer. Where the page is viewable by another user type the view usually only alters by the navigation bar which has options specific to each user category.

➤ Home Page



Figure 7. Home page of Serpahim CRM (any user view)

➤ Register Employee Page

Figure 8. Register employee page of Serpahim CRM

➤ Register Employer Page

The screenshot shows the 'Employer Registration' page of the Seraphim CRM. At the top left is the Seraphim logo, and at the top right is a 'Logout' button. The title 'Seraphim CRM' is centered at the top, with 'Employer Registration' below it. The form contains the following fields: Company Name, Contact Name, Email, Contact Number, Address, Postcode, username, Password, and Confirm Password. Each field is represented by a white input box. At the bottom of the form are 'Submit' and 'Cancel' buttons.

Figure 9. Register employer page of Serpahim CRM

➤ Login Page

The screenshot shows the 'Login' page of the Seraphim CRM. At the top left is the Seraphim logo, and at the top right is a 'Logout' button. The title 'Seraphim CRM' is centered at the top. The form contains two fields: 'Username' and 'Password', each with a white input box. At the bottom of the form are 'Login' and 'Cancel' buttons.

Figure 10. Login page of Seraphim CRM

➤ Management Logged in Home Page



Figure 11. Management home page of Seraphim CRM

➤ Employer Logged in Home Page



Figure 12. Employee home page of Seraphim CRM

➤ Employee Logged in Home Page

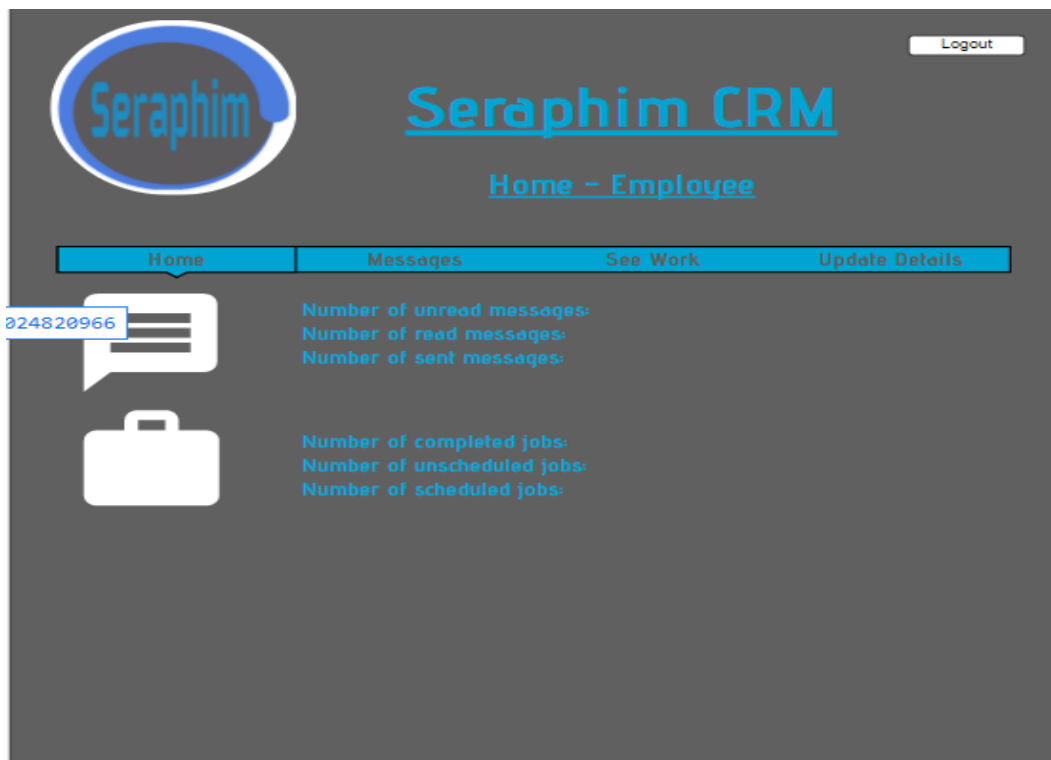


Figure 13. Employee home page of Seraphim CRM

➤ Manage Messages Page



Figure 14. Manage messages page of seraphim CRM (employer view)

➤ Send Message Page

Figure 15. Send message page of Serpahim CRM (employer view)

➤ View Message Page

Figure 16. View message page of Seraphim CRM (employer view)

➤ Job Management Page



Figure 17. Manage jobs page of Seraphim CRM (employer view)

➤ Add A Job Page

Figure 18. Add a job page of Seraphim CRM (employer view)

➤ Remove A Job Page

Logout

Seraphim

Seraphim CRM

Remove a Job

Home Messages See Work Manage Jobs Update Details

Job ID:

Job Name:

Remove Job Cancel

Figure 19. Remove a job page of Seraphim CRM (employer view)

➤ Update A Job Page

Logout

Seraphim

Seraphim CRM

Update a Job

Home Messages See Work Manage Jobs Update Details

Job ID: Search for Job

Name of Job:

Job Description:

Rate of Pay:

Job Start Date:

Job End Date:

Job End Date:

Job Type: Options

Update Job Cancel

Figure 20. Update a job page of Seraphim CRM (employer view)

➤ Assign Employee to A Job Page

Logout

Seraphim CRM

Assign Employee to a Job

Home Messages See Work Manage Jobs Manage Database Stats

Job ID: Search for Job

Name of Job:

Employee ID:

Employee First Name:

Employee Last Name:

Job Start Date:

Job End Date:

Assign Employee Cancel

Figure 21. Assign an employee to a job page of Seraphim CRM (manager view)

➤ Manage Database Page

Logout

Seraphim CRM

Manage Database

Home Messages See Work Manage Jobs Manage Database Stats

Update Employee Details Update Employer Details Update User Details

Add New Employee Add New Employer Remove User

Add Job Industry Add New Document Add New Disclosure

Update Disclosure Update Document View Document

Figure 22. Manage database page for Seraphim CRM (manager view)

➤ Manage Database Example Options Form Page

The screenshot shows the 'Add New Employee' form in the Seraphim CRM manager view. The interface includes a top navigation bar with a 'Logout' button and a main header with the Seraphim logo and 'Seraphim CRM'. Below the header is a sub-header 'Add New Employee' and a navigation menu with tabs: Home, Messages, See Work, Manage Jobs, Manage Database, and Stats. The form itself contains several input fields for employee details: First Name, Last Name, Address, Postcode, Date lived at address from, Mobile Number, Home Telephone Number, email, username, Password, and Confirm Password. At the bottom of the form are 'Submit' and 'Cancel' buttons.

Figure 23. Example manage database option (add new employee) of Seraphim CRM (manager view)

➤ Retrieve Stats Page

The screenshot shows the 'Manage Statistics' page in the Seraphim CRM manager view. The interface includes a top navigation bar with a 'Logout' button and a main header with the Seraphim logo and 'Seraphim CRM'. Below the header is a sub-header 'Manage Statistics' and a navigation menu with tabs: Home, Messages, See Work, Manage Jobs, Manage Database, and Stats. The page displays a grid of 12 statistics cards arranged in 4 rows and 3 columns. The statistics are: Number of Jobs per Overall, Number of Completed Jobs, Number of Uncompleted Jobs, Average Number of Jobs Overall, Average Number of Completed Jobs, Average Number of Uncompleted Jobs, Number of Jobs By Employer Overall, Number of Completed Jobs By Employer, Number of Uncompleted Jobs By Employer, Number of Jobs By Industry Type Overall, Number of Completed Jobs By Industry Type, and Number of Uncompleted Jobs By Industry Type.

Figure 24. Manage statistics page of Seraphim CRM (manager view)

➤ Retrieve Stats Example Options Display Page

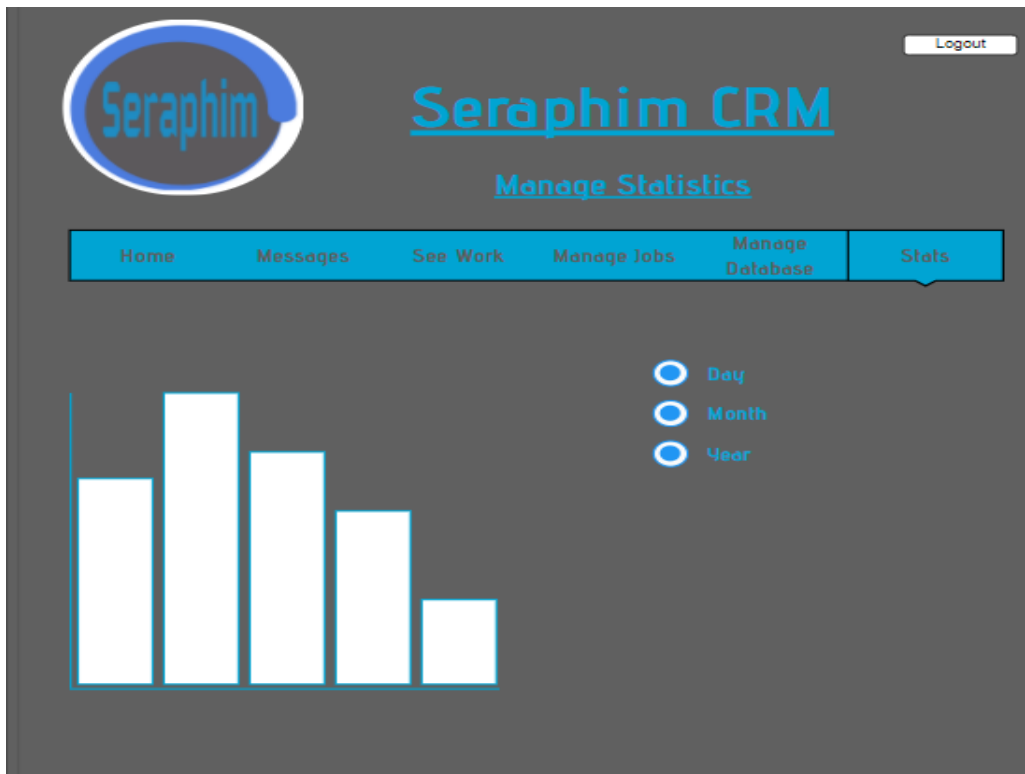


Figure 25. Manage statistics sample option page for Seraphim CRM (manager view)

➤ See Work Page

Job ID	Name	Employer ID	Status	Description	Pay Rate	Start Date	End Date	Industry Type	Employee ID
Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Cell 7	Cell 8	Cell 9	Cell 10
Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Cell 7	Cell 8	Cell 9	Cell 10
Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Cell 7	Cell 8	Cell 9	Cell 10
Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Cell 7	Cell 8	Cell 9	Cell 10

Figure 26. See work page for Seraphim CRM (manager view)

➤ Logout Page



Figure 27. Logout page for Seraphim CRM (any user view)

3.7 EER Diagram

An Enhanced Entity Relationship(EER) diagram is a visual representation of a database design and how each entity(table) will look and how these entities relate to each other. The diagram can be read as follows:

- Each entity is described by a box with a title and attributes
- Each attribute is described by a name and type
- Each entity has a primary key to uniquely identify it described with a gold key
- Attributes that can be null are described with a white diamond next to them
- Attributes that can't be null are described with a blue diamond next to them
- Attributes that are foreign keys are described with a red diamond next to them
- Relationships are from one entity to another and are described by a dashed line
- Relationships have a participation which is described by a diamond in the middle of the relationship for example:
 - Employee must be a User is described by white diamond half on User side of relationship
 - User can be an Employee described by black diamond half on Employee side of relationship
- Relationships have a cardinality(most) and ordinality(least) amount of times an instance of one entity can relate to an instance of another entity, for example:
 - Employee must be a User 1 time
 - User can be an Employee 0 to many times
 - 0 = zero times, 1 = 1 time, * = many times

Figure 28 shows the EER diagram for the Seraphim database using the notation explained above.

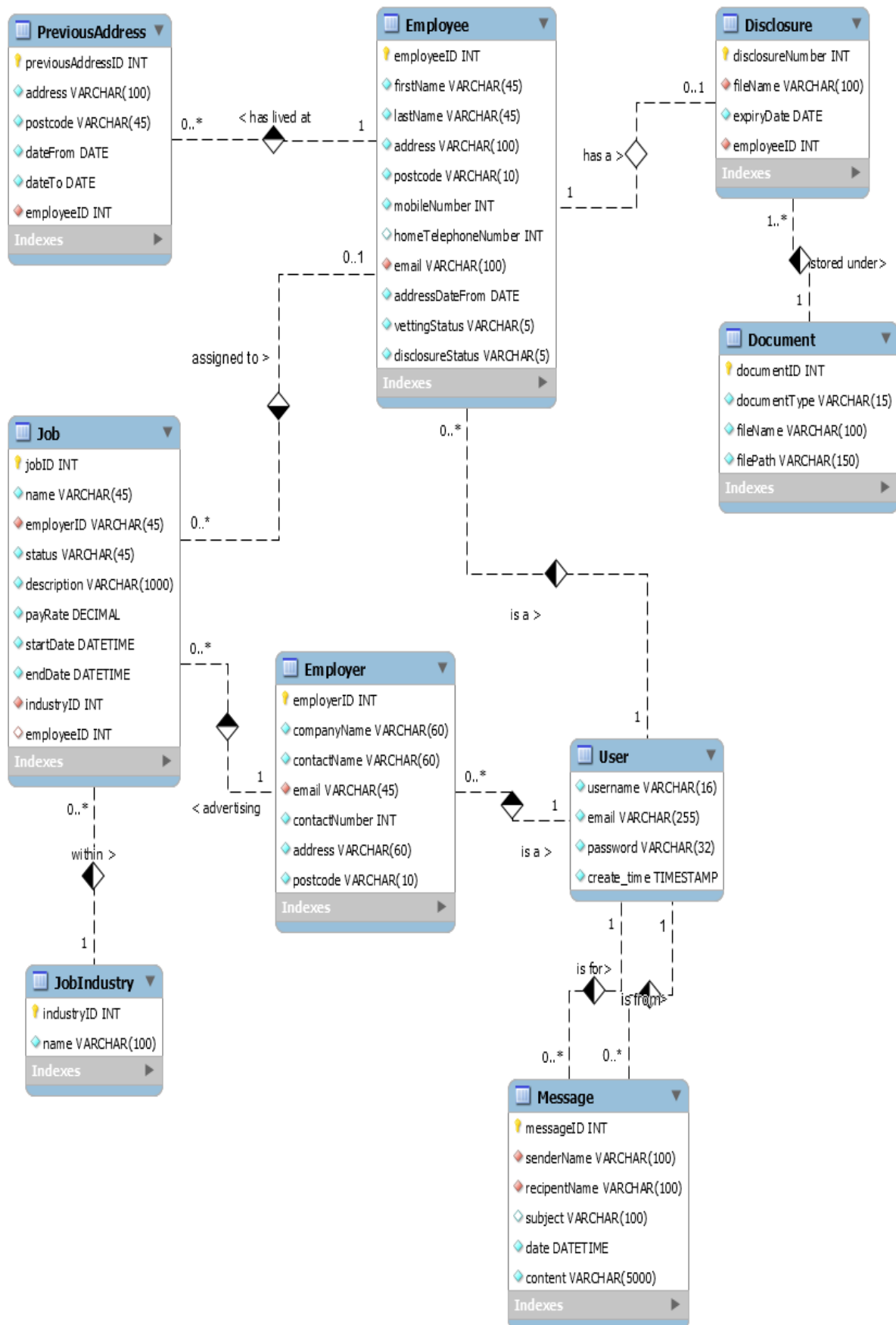


Figure 28. Seraphim Database EER Diagram

3.8 Relational Schema

The relational schema allows you to identify where there is a relationship within the data and how records are identified via what their primary and foreign keys are. Primary keys uniquely identify each record and foreign keys uniquely identify a record in a relationship.

Relation
User (username, email, password, create_time) Primary Key username
Employee (employeeID, firstName, lastName, address, postcode, mobileNumber, homeTelephoneNumber, email, addressDateFrom, vettingStatus, disclosureStatus) Primary Key employeeID Foreign Key email references User(email) ON DELETE no action ON UPDATE cascade
Employer (employerID, companyName, contactName, email, contactNumber, address, postcode) Primary Key employerID Foreign Key email references User(email) ON DELETE no action ON UPDATE cascade
Job (jobID, name, employerID, status, description, payRate, startDate, endDate, industryID, employeeID) Primary Key jobID Foreign Key employerID references Employer(employerID) ON DELETE no action ON UPDATE no action Foreign Key industryID references Industry(industryID) ON DELETE no action ON UPDATE no action Foreign Key employeeID references Employee(employeeID) ON DELETE no action ON UPDATE no action
JobIndustry (industryID, name) Primary Key industryID
PreviousAddress (previousAddressID, address, postcode, dateFrom, dateTo, employeeID) Primary Key previousAddressID Foreign Key employeeID references Employee(employeeID) ON DELETE no action ON UPDATE no action
Disclosure (disclosureNumber, fileName, expiryDate, employeeID)

Primary Key disclosureNumber Foreign Key fileName references Document(fileName) ON DELETE no action ON UPDATE no action Foreign Key employeeID references Employee(employeeID) ON DELETE no action ON UPDATE no action
Document (documentID, documentType, fileName, filePath) PrimaryKey documentID
Message (messageID, senderName, recipientName, subject, date, content) Primary Key messageID Foreign Key senderName references User(username) ON DELETE no action ON UPDATE no action Foreign Key recipientName references User(username) ON DELETE no action ON UPDATE no action

Table 19. Design stage relational schema

3.9 Data Dictionary

The database component of the CRM requires a data dictionary to describe exactly what data it will store, what type that data will be, any relationships that data has with other data (foreign keys), whether the data is a primary key, which table each data member is stored under. This can be referred to when wanting a description of the structure of the database and what format any data been transported in to the database must be in. Detailed below is each table and the structure of the tables including:

- Column names
- Data types
- Whether is a primary and/or foreign key
- Whether the column fields can be null
- Whether the column fields are unique
- Any default values of the column fields
- A description of the column
- Any constraints and/or properties of the column

EmployeeTbl

Column Name	Data Type	Primary Key/Foreign Key	Null	Unique	Default	Description	Constraint/Column Property
employeeID	INT	PK	N	Y		Unique employee ID to identify	Set identity specification seed as 1 and increment as

						each employee, set to auto increment	1.
firstName	VARCHAR (45)	N/A	N	N		First name of employee	
lastName	VARCHAR (45)	N/A	N	N		Last name of employee	
Address	VARCHAR (100)	N/A	N	N		Current address of employee	
Postcode	VARCHAR (10)	N/A	N	N		Current postcode of employee	
mobileNumber	INT	N/A	N	Y		Mobile telephone number of employee	
homeTelephone Number	INT	N/A	Y	Y		Home telephone number of employee	
email	VARCHAR (100)	FK	N	N		Email address that employee used to register	
addressDateFrom	DATE	N/A	N	N		Date employee moved	Date must be before current date

						in to current address	
vettingStatus	VARCHAR (10)	N/A	N	N	"Incomplete"	Management can update vetting status to complete when 5 year address history and 2 references have been provided	Can only hold values "Complete" or "Incomplete"
disclosureStatus	VARCHAR (10)	N/A	N	N	"Unverified"	Management can update disclosure status to verified when relevant disclosure Scotland certificate has been uploaded to the system	Can only hold values "Verified" or "Unverified"

Table 20. Employee Table

EmployerTbl

Column Name	Data Type	Primary Key/Foreign Key	Null	Unique	Default	Description	Constraint/Column Property
employerID	INT	PK	N	Y		Unique identifier for employers with an auto	Set identity specification seed as 1 and increment as 1.

						increment.	
companyName	VARCHAR(60)	N/A	N	N		Name of employer company	
contactName	VARCHAR(60)	N/A	N	N		Name of contact person at employer company	
email	VARCHAR(100)	FK	N	Y		Email address of contact person at employer company	
contactNumber	INT	N/A	N	Y		Contact number of employer company	
Address	VARCHAR(100)	N/A	N	N		Address of employer company	
Postcode	VARCHAR(10)	N/A	N	N		Postcode of employer company	

Table 20. Employer Table

UserTbl

Column Name	Data Type	Primary Key/Foreign Key	Null	Unique	Default	Description	Constraint/Column Property
username	VARCHAR(30)	PK	N	Y		Username of user profile	
emailAddress	VARCHAR(255)	N/A	N	Y		Email address of user profile	
Password	VARCHAR(32)	N/A	N	N		Password of user profile	
Create_time	TIMESTAMP	N/A	Y	N		CURRENT_TIMESTAMP	

Table 21. User Table

JobTbl

Column Name	Data Type	Primary Key/Foreign	Null	Unique	Default	Description	Constraint/Column Property
-------------	-----------	---------------------	------	--------	---------	-------------	----------------------------

		ign Key					
jobID	INT	PK	N	Y		Unique identifier of the job with an auto increment	Set identity specification seed as 1 and increment as 1.
Name	VARCHAR(45)	N/A	N	N		Name of the job	
employeeID	VARCHAR(45)	FK	N	Y		ID of the employer for the job	
Status	VARCHAR(45)	N/A	N	N	"Unscheduled"	Status of the job	Can only hold values "Unscheduled", "Scheduled", "Complete"
Description	VARCHAR(1000)	N/A	N	N		Description of the job	
payRate	DECIMAL	N/A	N	N		Rate of pay for the job	
startDate	DATETIME	N/A	N	N		Date and time of the start of the job	
endDate	DATETIME	N/A	N	N		Date and time of the end of the job	
industryID	INT	FK	N	N		ID of industry for the job	
EmployeeID	INT	FK	Y	Y		ID of employee assigned to job	

Table 22. Job Table

JobIndustryTbl

Column	Data Type	Primary	Nul	Uniqu	Defau	Descripti	Constraint/Colu
--------	-----------	---------	-----	-------	-------	-----------	-----------------

Name		Key/Foreign Key	Null	Unique	Default	Description	Column Property
industryID	INT	PK	N	Y		Unique identifier of the job industry with an auto increment	Set identity specification seed as 1 and increment as 1.
Name	VARCHAR(100)	N/A	N	N		Name of job industry	Can only hold values "Security", "Event Crew", or "Catering"

Table 22. Job industry Table

DocumentTbl

Column Name	Data Type	Primary Key/Foreign Key	Null	Unique	Default	Description	Constraint/Column Property
documentID	INT	PK	N	Y		Unique identifier of document with an auto increment	Set identity specification seed as 1 and increment as 1.
documentType	VARCHAR(45)	N/A	N	N		Type of document either word document or pdf or jpeg	Can only hold values "Word doc", "PDF", or "JPEG"
fileName	VARCHAR(100)	N/A	N	Y		Name of file that document is stored under	
filePath	VARCHAR(150)	N/A	N	Y		Filepath for where the document is	

						stored	
--	--	--	--	--	--	--------	--

Table 23. Document Table

DisclosureTbl

Column Name	Data Type	Primary Key/Foreign Key	Null	Unique	Default	Description	Constraint/Column Property
disclosureNumber	INT	PK	N	Y		Number of disclosure Scotland certificate	
fileName	VARCHAR(100)	FK	N	Y		Name of file that disclosure is stored under	
expiryDate	DATE	N/A	N	N		Date of expiry for the disclosure	
employeeID	INT	FK	N	Y		ID of employee that the disclosure relates to	

Table 24. Disclosure Table

PreviousAddressTbl

Column Name	Data Type	Primary Key/Foreign Key	Null	Unique	Default	Description	Constraint/Column Property
previousAddressID	INT	PK	N	Y		Unique identifier of previous address with auto increment	Set identity specification seed as 1 and increment as 1.
address	VARCHAR(100)	N/A	N	N		Address	

	00)					of previous address	
postcode	VARCHAR(10)	N/A	N	N		Postcode of previous address	
dateFrom	DATE	N/A	N	N		Date previous address is from	
dateTo	DATE	N/A	N	N		Date previous address is to	
employeeID	INT	FK	N	Y		ID of employee of previous address	

Table 25. Previous Address Table

MessageTbl

Column Name	Data Type	Primary Key/Foreign Key	Null	Unique	Default	Description	Constraint/Column Property
messageID	INT	PK	N	Y		Unique identifier of message with auto increment	Set identity specification seed as 1 and increment as 1.
senderName	VARCHAR(100)	FK	N	N		Username of sender	
recipientName	VARCHAR(100)	FK	N	N		Username of recipient	
Subject	VARCHAR(100)	N/A	Y	N		Subject of message	
date	DATETIME	N/A	N	N		Date and time message was sent	Set to current date time
content	VARCHAR(50)	N/A	N	N		Content	

	00)					of message	
--	-----	--	--	--	--	---------------	--

Table 26. Message Table

3.10 Summary

To conclude this chapter, the methods that can be chosen to gather user requirements was fully researched with the appropriate methods for this project chosen and then carried out. The full design stage has been documented with all areas of design carried out included within the chapter. Following on from this the development of the CRM System can now take place which is discussed in the next chapter.

Chapter 4. Development

4.1 Introduction

During this chapter the components of the development stages are discussed in an overview format. The areas discussed serve the purpose of giving the reader an understanding of how each stage was developed with more detail being found within the complete CRM solution itself. The CRM solution can be found in the files included with this report in a USB or electronically in the same folder as this report. A video was also produced with a walkthrough demonstration of the working solution.

4.2 Development of Database

The development of the database was done entirely through MySQL workbench which is a free to install product as mentioned in chapter 2 of this report. The MySQL workbench allows a MySQL57 instance server to be connected and administered via the workbench making the PC that holds the connection the main server. By running the server, you can create and manage the database via the workbench which is an easy to use graphical user interface.

The MySQL workbench allows for the creating of the database schema in the first instance before any real tables are added to the database. The schema as described in the design section allows you to design and develop the EER diagrams and ensure any relations are correct and present within the database. Once a schema is created the database tables can be created by adding tables to the database schema or inputting SQL code to the schema directly. The seraphim database was created using SQL code as this ensures the exact constraints and properties of each table are configured as need be. Figure 29 below shows an example of the MySQL workbench where the SQL data was executed.

Each table was created in turn before any foreign key constraints were added to ensure the foreign key existed in the table it was being linked to. SQL code ensures that no foreign key can be linked to a primary key or unique key of a table that does not exist. As a foreign key serves the purpose of linking one table to another it would not make sense to be linking to a table that does not exist.

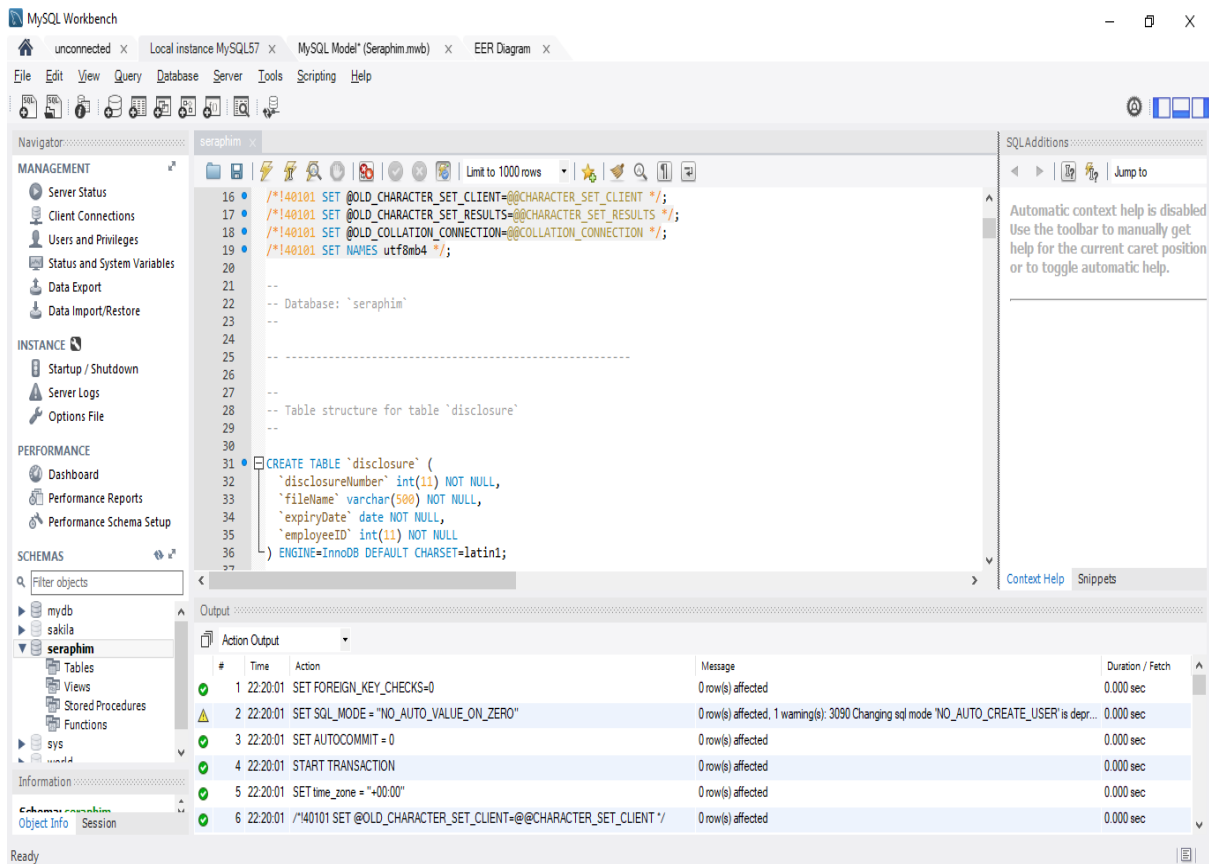


Figure 29. MySQL Workbench (Source: MySQL Workbench Seraphim Database)

Code is input in to the SQL editor and executed via the lightning bolt execute button, provided all the code is in the correct syntax the execution will run successfully. As seen in the above figure there is the option to input SQL to create tables which was done in this instance to create all 9 tables for the Seraphim database. Sample code for creating one of these tables:

```

CREATE TABLE `jobindustry` (
  `industryID` int(11) NOT NULL,
  `name` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

This code allows a table of name jobindustry to be created with the following columns:

- industryID with a datatype of integer and a length of 11 characters, it is also set to not allow nulls and to auto increment which starts at 1 and automatically increments by one number for each new record added to the table. The primary key value is also set to be this column.
- name with a datatype of varchar and a length of 100 characters, this again is also set to not allow nulls.

Each table for the database was created using a similar format as described above depending on the table requirements. Where a foreign key was needed which is a link to another table, the parent table had to be created first to allow for the foreign key to be added in the child table. Tables with foreign keys and indexes can be found in the data

dictionary which is included below, please note this has been updated from the design data dictionary to reflect any changes made since the initial design stages.

disclosure

Column	Type	Null	Default	Links to	Comments	MIME
disclosureNumber (<i>Primary</i>)	int(11)	No				
fileName	varchar(500)	No		document -> fileName		
expiryDate	date	No				
employeeID	int(11)	No		employee -> employeeID		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	disclosureNumber	3	A	No	
disclosureNumber_UNIQUE	BTREE	Yes	No	disclosureNumber	3	A	No	
employeeID_UNIQUE	BTREE	Yes	No	employeeID	3	A	No	
fk_disclosureDocument_document1_idx	BTREE	No	No	fileName	3	A	No	

document

Column	Type	Null	Default	Links to	Comments	MIME
documentID (<i>Primary</i>)	int(11)	No				
documentDescription	varchar(100)	No				
fileName	varchar(100)	No				
filePath	varchar(150)	No				
employeeID	int(11)	No		employee -> employeeID		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	documentID	16	A	No	
documentID_UNIQUE	BTREE	Yes	No	documentID	16	A	No	
fileName_UNIQUE	BTREE	Yes	No	fileName	16	A	No	

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
	E							
filePath_UNIQUE	BTREE	Yes	No	filePath	16	A	No	
document_employee	BTREE	No	No	employeeID	8	A	No	

employee

Column	Type	Null	Default	Links to	Comments	MIME
employeeID (<i>Primary</i>)	int(11)	No				
firstName	varchar(45)	No				
lastName	varchar(45)	No				
address	varchar(100)	No				
postcode	varchar(10)	No				
mobileNumber	varchar(11)	No				
homeTelephoneNumber	varchar(11)	Yes	NULL			
email	varchar(255)	No		user -> email		
addressDateFrom	date	No				
vettingStatus	varchar(5)	No	False			
disclosureStatus	varchar(5)	No	False			

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	employeeID	4	A	No	
employeeID_UNIQUE	BTREE	Yes	No	employeeID	4	A	No	
fk_employee_user	BTREE	No	No	email	4	A	No	

employer

Column	Type	Null	Default	Links to	Comments	MIME
employerID (<i>Primary</i>)	int(11)	No				
companyName	varchar(60)	No				
contactName	varchar(60)	No				
email	varchar(255)	No		user -> email		
contactNumber	int(11)	No				

address	varchar(60)	No				
postcode	varchar(10)	No				

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	employerID	5	A	No	
employerID_UNIQUE	BTREE	Yes	No	employerID	5	A	No	
companyName	BTREE	Yes	No	companyName	5	A	No	
fk_Employer_user1_idx	BTREE	No	No	email	5	A	No	

job

Column	Type	Null	Default	Links to	Comments	MIME
jobID (<i>Primary</i>)	int(11)	No				
name	varchar(45)	No				
employerID	int(11)	No		employer->employerID		
status	varchar(45)	No	"Unscheduled"			
description	varchar(1000)	No				
payRate	decimal(10,0)	No				
startDate	date	No				
endDate	date	No				
industryID	int(11)	No		jobindustry->industryID		
employeeID	int(11)	Yes	NULL	employee->employeeID		
removalReason	varchar(255)	Yes	NULL			

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	jobID	14	A	No	
jobID_UNIQUE	BTREE	Yes	No	jobID	14	A	No	
employeeID_UNIQUE	BTREE	Yes	No	employeeID	2	A	Yes	

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
	E			D				
fk_Job_Employer1_idx	BTREE	No	No	employerID	7	A	No	
fk_Job_jobIndustry1_idx	BTREE	No	No	industryID	14	A	No	

jobindustry

Column	Type	Null	Default	Links to	Comments	MIME
industryID (Primary)	int(11)	No				
name	varchar(100)	No				

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	industryID	4	A	No	
name	BTREE	Yes	No	name	4	A	No	

message

Column	Type	Null	Default	Links to	Comments	MIME
messageID (Primary)	int(11)	No				
senderName	varchar(255)	No		user -> username		
recipientName	varchar(100)	No		user -> username		
subject	varchar(100)	Yes	NULL			
content	varchar(5000)	No				
status	varchar(10)	No	unread			
create_time	timestamp	No	CURRENT_TIMESTAMP			

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	messageID	5	A	No	
messageID_UNIQUE	BTREE	Yes	No	messageID	5	A	No	

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
	E							
fk_sender_username	BTREE	No	No	senderName	5	A	No	
fk_recipient_username	BTREE	No	No	recipientName	5	A	No	

previousaddress

Column	Type	Null	Default	Links to	Comments	MIME
previousAddressID (<i>Primary</i>)	int(11)	No				
address	varchar(100)	No				
postcode	varchar(45)	No				
dateFrom	date	No				
dateTo	date	No				
employeeID	int(11)	No		employee -> employeeID		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	previousAddressID	0	A	No	
previousAddressID_UNIQUE	BTREE	Yes	No	previousAddressID	0	A	No	
fk_previousAddress_employee1_idx	BTREE	No	No	employeeID	0	A	No	

user

Column	Type	Null	Default	Links to	Comments	MIME
username (<i>Primary</i>)	varchar(100)	No				
email	varchar(255)	No				
userPassword	varchar(255)	No				
userType	varchar(10)	No				
status	varchar(20)	No	false			
token	varchar(255)	No				
create_time	timestamp	No	CURRENT_TIMESTAMP			
removalReason	varchar(255)	Yes	NULL			

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	username	10	A	No	
email	BTREE	Yes	No	email	10	A	No	

In the data dictionary above the indexes of each table are included alongside the table itself, indexes are created for primary keys, foreign keys and unique keys. After the design of the database the DBMS was not used to upload any data as this was done in the next stage of development during the design of the website to allow for features to be tested as and when they were complete. This reflects how the company will mainly access the database to keep the integrity of the database and prevent unauthorized access or compromising of the data. The overall cost of developing the database was kept at £0 with only time being spent and no cost to the software itself. The database files will be included in the USB flash drive included with this report.

4.2 Development of Website

The website itself is a custom company website designed to act as the main interface for the CRM where managers, employees and managers can access and perform the functions they require. As described in the technical section of this report the software used to develop the website was JetBrains IDE mainly the PhpStorm component of the IDE. Using this platform all programming languages needed to develop the website where able to be used and the ones selected for this website where as follows:

- HTML – Hypertext Mark-up Language. This language is used to layout the main content of the website pages such as headings, titles, paragraphs and forms.
- CSS – Cascading Style Sheet. The underlying language used to develop the design components of the website such as fonts, colours, where certain elements should be on the page etc.
- JavaScript which is a language used to add client-side behaviour to the html such as what happens when a button is clicked etc.
- PHP which is a server-side scripting language used to tell the server what you need it to do.

To ensure each page was developed in the same way a page template was initially created which was then re-created for each individual page needed throughout the website. The below sample of code in figure 30 displays this template:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Add New Employee Page</title>
  <link rel="stylesheet" type="text/css" href="CSS.css">
  <link href="https://fonts.googleapis.com/css?family=Advent+Pro"
rel="stylesheet">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
</head>
```

```

<body>
  <header>
    <div class="logo">
      
    </div>
    <div id="header">
      <h1>Seraphim CRM</h1>
      <h3>Add New Employee</h3>
    </div>
    <div id="logout_button">
      <button type="button"
onclick="location.href='Logout%20Page.php'">Logout</button>
    </div>

  </header>
  <div class="nav_bar" id="nav_employee">
    <ul>
      <li><a href="Logged%20In%20Page.php">Home</a></li>
      <li><a href="Manage%20Messages%20Page.php">Messages</a></li>
      <li><a href="See%20Work%20Page.php">See Work</a></li>
      <li><a href="Update%20Employee%20Details%20Page.php">Update
Details</a></li>
    </ul>
  </div>
  <div class="nav_bar" id="nav_employer">
    <ul>
      <li><a href="Logged%20In%20Page.php">Home</a></li>
      <li><a href="Manage%20Messages%20Page.php">Messages</a></li>
      <li><a href="See%20Work%20Page.php">See Work</a></li>
      <li><a href="Job%20Management%20Page.php">Manage Jobs</a></li>
      <li><a href="Update%20Employer%20Details%20Page.php">Update
Details</a></li>
    </ul>
  </div>
  <div class="nav_bar" id="nav_admin">
    <ul>
      <li><a href="Logged%20In%20Page.php">Home</a></li>
      <li><a href="Manage%20Messages%20Page.php">Messages</a></li>
      <li><a href="See%20Work%20Page.php">See Work</a></li>
      <li><a href="Job%20Management%20Page.php">Manage Jobs</a></li>
      <li><a class="active" href="Manage%20Database%20Page.php">Manage
Database</a></li>
      <li><a href="Retrieve%20Stats%20Page.php">Stats</a></li>
    </ul>
  </div>
  //where the main body of the page goes.
  <footer>
    <div class="footer">
      <p>&copy; Designed & Developed By: Jolene Sweeney <a
href="https://www.linkedin.com/in/jolene-sweeney-5026a5150/"
class = "fa fa-
linkedin"></a>
| Seraphim Contact Number: 00000000000 | Seraphim Facebook: <a
href="#"
class="fa fa-facebook"></a><p>
    </div>
  </footer>
</body>
</html>

```

Figure 30. Sample template of Seraphim Website page.

This above template remained the same throughout each page except for the login and register pages where the navigation bar components where removed as they do not come in to effect until a user is logged in. There are also 3 navigation bars displayed and this

remains the case throughout each page due to there being 3 different types of users that use the CRM, depending on which user is logged in will depend on which navigation bar is displayed. The code used to determine which navigation bar is displayed is in figure 31 below:

```
<?php
/**
 * Created by PhpStorm.
 * User: jolen
 * Date: 26/02/2018
 * Time: 01:35
 */

require('DBConfig.php');

session_start();
$userType = $_SESSION['user'];
if($userType == 'Employee'){
    echo "<script type='text/javascript'>window.onload =
function(){document.getElementById('nav_employer').style.display = 'none';
document.getElementById('nav_admin').style.display = 'none';}</script>";
}else if($userType == 'Employer'){
    echo "<script type='text/javascript'>window.onload =
function(){document.getElementById('nav_employee').style.display = 'none';
document.getElementById('nav_admin').style.display =
'none';}</script>";
}else if($userType == 'Manager'){
    echo "<script type='text/javascript'>window.onload =
function(){document.getElementById('nav_employee').style.display = 'none';
document.getElementById('nav_employer').style.display =
'none';}</script>";
}
```

Figure 31. Code to determine which navigational display to use.

The navigational menu code above gets the user type from the session variables which were stored when the user initially logged in. Using this user type the navigational menu's which are not needed for that user are then hidden using a mixture of PHP code and JavaScript to amend the style and display of the navigational menus. Using this technique it allowed for a custom navigation menu to be displayed for each user type and ensured only the functions their roles allowed access to are available to them.

Once the initial template design of each website page was designed the main body of the pages could be developed in a custom manner individually. A selection of the final developed pages can be seen in the below figures with the rest being available to view via the files on the accompanied USB flash drive.



Figure 32. Home page of Seraphim CRM Final Development.

Figure 32 displays the main home page of the CRM which opens when the user initially directs to the CRM. From this page there are 3 buttons available to click which allows the user to either register as an employee or employer or for a pre-registered user to login. Each of these buttons navigate to the correct page and the code that allows this functionality to be performed is displayed in figure 33.

```
<div class="manage_buttons">
  <p>Either register as an employee or employer or login.</p>
  <button type="button"
onclick="location.href='Login%20Page.php'">Login</button>
  <button type="button"
onclick="location.href='Register%20Employee%20Page.php'">Register Employee</button>
  <button type="button"
onclick="location.href='Register%20Employer%20Page.php'">Register Employer</button>
</div>
```

Figure 33. Sample code for a button when it is clicked.

There are many pages throughout the website with similar functionality such as manage database page, retrieve statistics page or the job management page. These pages each have many buttons that can be clicked to direct the user to the relevant function and are displayed and coded in a very similar way as can be seen in figures 32 and 33.

Another main function of this website is the use of HTML forms which are used for many purposes such as registering a user, updating user details, sending a message etc. Each of these forms are again coded in a very similar manner therefore only one will be shown here in figures 34 and 35 with the rest being available to view in the files on the USB.

Enter details and click submit to register as an employee to database and create new user profile.

* First Name
 * Last Name
 * Address
 * Postcode
 * Date lived at address from
 dd/mm/yyyy
 * Mobile Number
 Home Telephone Number
 * Email
 * Username
 * Password
 * Confirm Password

Submit Cancel

© Designed & Developed By: Jolene Sweeney | Seraphim Contact Number: 0000000000 | Seraphim Facebook

Figure 34. Register Employee form for Seraphim CRM.

The above form gives a variety of input fields for a user to type in data with the type of data being needed displayed in a label above the field. Each field that is required is marked with a red asterisk (*) to aid the user in knowing which fields to fill in. There are 2 buttons alongside the form one to submit the form which runs the PHP server-side code in the background (discussed more in section 4.3) and one to cancel the form which resets the form back to blank. Figure 35 below describes the code needed to display these forms on a web-page including marking fields required, any patterns needed such as postcode and telephone patterns, and lengths of the fields:

```
<div class="register_employee_form">
  <p>Enter details and click submit to register as an employee to database and
  create new user profile.</p>
  <form action="../../PHP/RegisterEmployee.php" method="post">
    <label class="required">
      First Name:<br>
      <input type="text" name="firstName" required><br>
    </label>
    <label class="required">
      Last Name:<br>
      <input type="text" name="lastName" required><br>
    </label>
    <label class="required">
      Address:<br>
      <input type="text" name="address" required><br>
    </label>
    <label class="required">
      Postcode:<br>
      <input type="text" name="postcode" required pattern="[a-zA-Z]{1,2}[0-
      9Rr][0-9a-zA-z]? ( | ) [0-9][a-zA-z]{2}"
      title="Valid UK Postcode"><br>
    </label>
    <label class="required">
```

```

        Date lived at address from:<br>
        <input type="date" name="dateLivedAtAddressFrom" required><br>
    </label>
    <label class="required">
        Mobile Number:<br>
        <input type="tel" name="mobNumber" required pattern="[0-9]{11}"
maxlength="11" title="11 Digit Mobile Number"><br>
    </label>
    <label>
        Home Telephone Number:<br>
        <input type="tel" name="homeTelNumber" pattern="[0-9]{11}"
maxlength="11" title="11 Digit Home Telephone Number"><br>
    </label>
    <label class="required">
        Email:<br>
        <input type="email" name="email" required><br>
    </label>
    <label class="required">
        Username:<br>
        <input type="text" name="username" required><br>
    </label>
    <label class="required">
        Password:<br>
        <input type="password" name="password" required minlength="8"
title="Password must be at least 8 characters"><br>
    </label>
    <label class="required">
        Confirm Password:<br>
        <input type="password" name="confirmPassword" required minlength="8"
title="Password must be at least 8 characters"><br>
    </label>
    <button type="submit" name="submit">Submit</button>
    <button type="reset" name="cancel">Cancel</button>
</form>
</div>

```

Figure 35. Code to display register employee form.

Another main feature that appears in a few pages throughout the website is a table of data. A table is displayed in the main body of the page with several rows containing data from the database. This data is pulled using PHP and displayed in the table rows to allow the user to see the data related to them. This is the case for seeing jobs, messages, documents etc. The layout can be seen in figure 36 with the code being described in section 4.3.



Figure 36. Table of messages displayed in the manage messages page of Seraphim CRM

One of the last main features of the website is the display of charts for the manager to analyze statistics relating to the CRM. For the data to be displayed in a quick and easy manner for the manager to get the information they require a bar chart was used rather than just displaying the data itself. To display the charts Fusion-Charts XT suite was used as the interface, running off a pure JavaScript API on a trial basis which can be used for as long as necessary. The charts are predefined in a library which is stored in the website directory and can be used to customize each chart to be displayed (Anon, 2018). There are many different chart options to choose from such as line charts or pie but for simplicity a bar chart was chosen. Figure 37 below will display an example of a bar chart taken from one of the website pages.

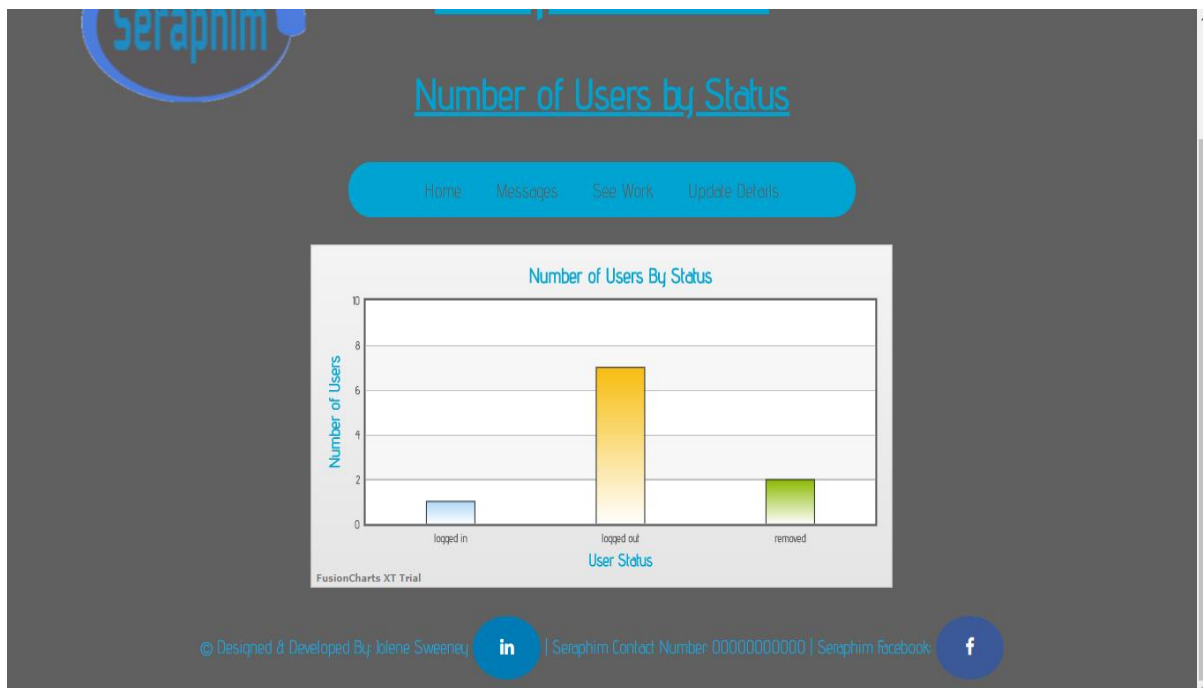


Figure 37. Bar chart displaying number of users by status in Seraphim CRM

Each of the options described in this section have been replicated and modified for each requirement but none-the-less follow the same structure. By doing this for each page for the website a fully developed website for the CRM was completed and using the software described to develop it allowed the costs to be only that of time for the developer keeping the solution cost-effective for the SME. Section 4.3 details the connection of the database to the website allowing for the data to be created, stored, retrieved and updated as required.

4.4 Connection of Database to Website

The connection of the database to the website forms the core part of the CRM as without this functionality the website and database cannot function together. PHP is a server-side scripting language and works seamlessly alongside a database allowing you to create SQL statements directly in the PHP and use them to perform functions. The main component of this CRM is the use of forms to input or update data within the CRM and PHP allows this to happen when the submit button is clicked. Upon submission of a form a PHP file is called and the code within that file is run to perform the updates to the database.

Each web-page within the CRM has its own PHP file to be called except for the pages that contain only buttons like the home page. When the PHP file is called the script is ran and when complete JavaScript code within the PHP is used to refresh the page to the required destination rather than defaulting the refresh to the PHP file page. The PHP file can be called within the page in a few ways:

- Forms - form action is set to the PHP page link:

```
<form action="../../PHP/RegisterEmployee.php" method="post">
```

Which calls the page upon the submit button being clicked.

- Within the main body of the HTML:

```
<?php
```

```
include '../PHP/NumberMessagesByStatus.php';
```

```
?>
```

Used to display an element coded through PHP such as a chart or table, PHP include is put in the exact place you wish the element to be displayed on the web-page.

- At the top of the main web-page before any other code to run the PHP file on initialisation:

```
<?php include '../PHP/NavigationMenu.php' ?>
```

This has been used to perform functions such as the display of the navigation menu which must happen upon the page opening.

For the database connection to be established in each PHP file and for the MySQL commands to be used within PHP a database configuration file was created to store the database configuration details and is then included within each PHP file to save the code being repeated each time and reducing an error in this code being created. The code for the database configuration is displayed in figure 38 below:

```
<?php
/**
 * Created by PhpStorm.
 * User: jolen
 * Date: 26/02/2018
 * Time: 18:54
 */

$link = mysqli_connect("localhost", "root", "andrew21", "seraphim");

//check connection
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
```

Figure 38. DBConfig.php code.

This config file is then called by using one line of code:

```
require('DBConfig.php');
```

When a connection to the database has been established code can then be used to perform various SQL functions with the web-pages described in section 4.2. A form page like the register employee form runs the PHP file to submit the data to the database as described in figure 39 below:

```
<?php
/**
 * Created by PhpStorm.
 * User: jolen
 * Date: 14/02/2018
 * Time: 11:29
 */

require('DBConfig.php');
```

```

$firstName = htmlspecialchars(mysqli_real_escape_string($link,
$ REQUEST['firstName']));
$lastName = htmlspecialchars(mysqli_real_escape_string($link,
$ REQUEST['lastName']));
$address = htmlspecialchars(mysqli_real_escape_string($link,
$ REQUEST['address']));
$postcode = htmlspecialchars(mysqli_real_escape_string($link,
$ REQUEST['postcode']));
$dateLivedAtAddressFrom = mysqli_real_escape_string($link,
$ REQUEST['dateLivedAtAddressFrom']);
$mobNumber = htmlspecialchars(mysqli_real_escape_string($link,
$ REQUEST['mobNumber']));
$homeTelNumber = htmlspecialchars(mysqli_real_escape_string($link,
$ REQUEST['homeTelNumber']));
$email = htmlspecialchars(mysqli_real_escape_string($link, $ REQUEST['email']));
$username = htmlspecialchars(mysqli_real_escape_string($link,
$ REQUEST['username']));
$password = htmlspecialchars(mysqli_real_escape_string($link,
$ REQUEST['password']));
$confirmPassword = htmlspecialchars(mysqli_real_escape_string($link,
$ REQUEST['confirmPassword']));

//check employee doesn't exist in database
$sql = "SELECT * FROM employee WHERE email = '$email'";
$result = mysqli_query($link, $sql);
$numRows = mysqli_num_rows($result);

//check username doesn't exist
$sql1 = "SELECT * FROM user WHERE username = '$username'";
$result1 = mysqli_query($link, $sql);
$numRows1 = mysqli_num_rows($result1);

if($numRows1 == 0) {
    if ($password === $confirmPassword) {
        $hashed_password = password_hash($password, PASSWORD_DEFAULT);
        $sql2 = "INSERT INTO user (username, email, userPassword, userType, status)
VALUES ('$username', '$email',
'$hashed_password', 'Employee', 'Logged Out')";
    } else {
        $message = "Passwords do not match";
        echo "<script type='text/javascript'> alert('$message');
window.location.href='../HTML/Register%20Employee%20Page.php';</script>";
    }
} else {
    $message = "ERROR: Username already exists please choose another username.";
    echo "<script type='text/javascript'> alert('$message');
window.location.href='../HTML/Register%20Employee%20Page.php';</script>";
}

if($numRows == 0) {
    //insert into employee table
    $sql3 = "INSERT INTO employee (firstName, lastName, address, postcode,
mobileNumber, homeTelephoneNumber, email,
addressDateFrom, vettingStatus, disclosureStatus) VALUES ('$firstName',
'$lastName', '$address', '$postcode', '$mobNumber',
'$homeTelNumber', '$email', '$dateLivedAtAddressFrom', 'False', 'False')";
} else {
    $message = "ERROR: Employee email already exists.";
    echo "<script type='text/javascript'> alert('$message');
window.location.href='../HTML/Register%20Employee%20Page.php';</script>";
}

if (mysqli_query($link, $sql2)) {
    $message = "User created.";
    echo "<script type='text/javascript'> alert('$message');</script>";
    if (mysqli_query($link, $sql3)) {
        $message = "Employee registered.";
    }
}

```

```

        echo "<script type='text/javascript'> alert('$message');
window.location.href='../HTML/Register%20Employee%20Page.php';</script>";
    } else {
        echo "ERROR: Could not able to execute $sql2. " . mysqli_error($link);
    }
} else {
    echo "ERROR: Could not able to execute $sql1. " . mysqli_error($link);
}
mysqli_close($link);

```

Figure 39. Register employee php code.

Upon a form being submitted the data stored in each field is stored in a variable, validation is then performed to ensure data isn't being committed and breaking the database integrity rules for example where an employee with the same email already exists. After any required validation is performed and if the validation is correct then the SQL statements are run and the data is uploaded to the database before confirming to the user it is successful or not and then refreshing back to the desired page.

Where it is a table being created the format of code differs from that of a form and this can be seen in figure 40. The data needed for the rows of the table must first be requested from the database and then input in to the rows of the table which is created within the PHP code. The table is created by echoing html code rather than the HTML always being there it is only created when the table is actually needed.

```

<?php
/**
 * Created by PhpStorm.
 * User: jolen
 * Date: 28/02/2018
 * Time: 00:00
 */

//store logged in username as recipient to pull correct messages for user from
database
$recipient = $_SESSION['username'];

//select messages for user that aren't removed and order by datetime
$sql2 = "SELECT * FROM message WHERE recipientName = '$recipient' AND status <>
'removed' ORDER BY UNIX_TIMESTAMP(create_time) DESC";
$result2 = mysqli_query($link, $sql2);

//create table and store messages in table
echo "<table border = '1'>
<tr>
<th>Sender</th>
<th>Subject</th>
<th>Message</th>
<th>View Message</th>
</tr>";

while($rs2 = mysqli_fetch_array($result2)){
    $messageID = $rs2['messageID'];
    echo "<tr>";
    echo "<td>" . $rs2['senderName'] . "</td>";
    echo "<td>" . $rs2['subject'] . "</td>";
    echo "<td>" . $rs2['content'] . "</td>";
    echo "<td><a href='../HTML/View%20Message%20Page.php?id=" . $messageID . "' >View

```

```

Message</a></td>";
    echo "</tr>";
}
echo "</table>";

//close database
mysqli_close($link);

```

Figure 40. Manage messages table code.

The last main feature is the creation of the charts for the displaying of statistics within the database. This proved the most difficult of features to implement as the correct SQL must first be established to get the required results from the database for the statistic being displayed like a count for example. When the desired data has been retrieved it is then populated in to the fusion charts code which displays the final chart as defined. Figure 41 below details the code used for this feature:

```

<?php
/**
 * Created by PhpStorm.
 * User: jolen
 * Date: 01/03/2018
 * Time: 17:17
 */

require ('DBConfig.php');

include '../FusionCharts/fusioncharts.php';

$sql1 = "SELECT COUNT(username) AS numberOfUsers, status FROM user GROUP BY
status";
$result1 = mysqli_query($link, $sql1);

if($result1){
    $arrayData = array(
        "chart" => array(
            "caption" => "Number of Users By Status",
            "showValues" => "0",
            "theme" => "ocean",
            "captionFont" => "'Advent Pro', sans-serif",
            "captionFontColor" => "#00a4d3",
            "captionFontSize" => "18",
            "xAxisName" => "User Status",
            "yAxisName" => "Number of Users",
            "xAxisNameFont" => "'Advent Pro', sans-serif",
            "xAxisNameFontColor" => "#00a4d3",
            "xAxisNameFontSize" => "15",
            "yAxisNameFont" => "'Advent Pro', sans-serif",
            "yAxisNameFontColor" => "#00a4d3",
            "yAxisNameFontSize" => "15",
            "labelFont" => "'Advent Pro', sans-serif",
            "labelFontColor" => "#606060",
            "labelFontBold" => "1",
            "valueFont" => "'Advent Pro', sans-serif",
            "valueFontColor" => "#606060",
            "valueFontBold" => "1"
        )
    );

    $arrayData["data"] = array();
}

```

```

while($rs = mysqli_fetch_array($result1)){
    array_push($arrayData["data"], array(
        "label" => $rs['status'],
        "value" => $rs['numberOfUsers']
    ));
}

$jsonEncodedData = json_encode($arrayData);

$columnChart = new FusionCharts("column2D", "numberUsersChart", 600, 300,
"chart-1",
    "json", $jsonEncodedData);

$columnChart -> render();
}

mysqli_close($link);

```

Figure 41. Number of users by status chart code.

The same as when the main web-pages were developed each feature described above can be replicated and customised for the desired feature. Where a form is created a similar format is followed as seen in figure 39 to script the form to the database and the same applies for a table and a chart. Each page of PHP code can be seen in the files included within USB attached to this report. There is also error-handling built-in to the coding of the website which catches errors that may occur when the user is using the CRM and an appropriate error message is displayed.

The connection of the database to the website via the PHP code was the last stage of developing the CRM solution. When this was complete the testing phase of the full solution could begin which is detailed in the next chapter. As PHP is a free to use coding language and the developer learned this language prior to this development project the cost of developing this section of the solution was again kept at £0. As the overall aim of this project is to design a cost-effective solution for the SME this has been achieved successfully up until this point.

4.5 Summary

To summarise this chapter a high-level overview of the development of the CRM solution has been provided with examples of key development areas being discussed. As mentioned in the website section the functionality is repeated throughout a lot of the website therefore only an example of each development technique has been discussed. With development being complete the implementation of the system can begin which is discussed in the next chapter with an evaluation of the system.

Chapter 5. Implementation & Evaluation

5.1 Introduction

During this chapter the implementation of the system which wasn't required by the company is discussed from an advice point of view for when they are ready to complete it. As implementation normally takes place by the developers this is merely just an overview of advice that they should take in to account during this stage. The system has also been evaluated by the developer, the company and the test-users with the results being discussed in the evaluation sections.

5.2 Implementation

The implementation of the CRM System involves fully testing the solution and implementing it and the process of this is described in this section.

5.2.1 Test Cases

Before testing could begin a test case for each individual test had to be created to test the functionality of the CRM completely. Test cases are used to describe the test and the expected outcome and then mark how the test was performed. In total for the CRM 188 test cases were designed to test each individual feature of each page from the page opening and displaying to the form submitting correctly. The test cases can be found within the test log file included with this report. A sample set of test cases for one page is detailed below:

Test Case	Description	Test Steps	Expected Result	Date Tested	Tested By	Outcome
Add a Job Page - Opens & displays correctly	Page should open and point to the appropriate add a job page for the desired user, with all elements displayed as predicted.	Click on the add a job button in the manage jobs page.	add a job page displays with logo in top left corner, title in the middle of the top of the page, logout button in top right corner, correct nav bar for desired user displayed, add a job form displayed.			
Add a Job Page - Nav bar links	Nav bar links direct to correct pages	Click on each nav bar link to check where it directs to. Go back a page on the browser after clicking each button to then click the next until all buttons have been checked.	Links should navigate to the page described in the button clicked.			

Add a Job Page - Form submits	Form submits correctly	Input required fields and click submit	Form should submit and message pops up confirming job added. Entry to job table in database made.			
Add a Job Page - Form doesn't submit	Form doesn't submit when error on form	Input data into some of the required fields and leave some blank.	Form wont submit, error message displayed and no entry made to database.			
Add a Job Page - Form resets	Form resets when cancel button is clicked	Enter data into any of the fields and click cancel button	Form resets and page refreshes back to add a job page.			
Add a Job Page - Logout button	Logout button logs user out.	Click logout button	Logout page should be displayed and a message pops up confirming user is logged out. User status is changed to logged out in the database.			

Table Test 27. Sample Test Case

As can be seen from the sample test case above it fully explores the test being performed ensuring it performs the way it was intended to. These test cases can be repeated numerous times until the expected result matches the outcome. The next section on testing will detail how these test cases were used to perform the testing of the CRM.

5.2.2 Testing

Testing has been an ongoing process of the CRM during the development stages with each feature that was implemented tested as it was complete. By doing the testing this way which followed the software development process chosen for this project it reduced the errors produced during the testing stage as most errors were caught and rectified beforehand.

The first round of testing performed using the test cases created was a developer test performed by the developer of the whole CRM platform. The main reason for the developer testing the platform first is to ensure that errors found during the testing could be caught and rectified before the sample users tested it. During this round of testing only a few errors were found such as a spelling mistake in the code and this was rectified instantly and the test re-run until it was successful. This round of testing proved that the platform was developed well and the software development process proved successful due to the testing of each feature happening as it was developed. Out of 188 test cases only 4 errors were found which is an error rate of 2.13% and an example of an error found is displayed below:

Test Case	Description	Test Steps	Expected	Date	Tested By	Outcome
-----------	-------------	------------	----------	------	-----------	---------

			Result	Tested		
Register Employee - Form Submits	Form should submit after being filled in and appropriate entry to database made.	Fill in the register employee form filling in all required fields: First name, last name, address, postcode, date lived at address from, mobile number, email, username, password and confirm password. Click submit when form has been filled.	Form should submit correctly when all required fields have been filled in. One record should submit to database in the employee table and one in the user table. A pop up message should come up for each record added to confirm they have been added.	18/03/2018	Developer - Jolene Sweeney	Passwords don't match error displayed - checked code and spelling error in code caused this error. Test re-run and form submitted correctly pop ups displayed to confirm this and entries made to database.

Table 28. Complete Test Case

The next round of testing to be performed was by the manager of the CRM system who performed tests against the test cases related to manager functionality of the system. The manager has the most amount of functionality with them being able to perform most of the functions throughout the system such as retrieve statistics and manage the database. As a result, there was 176 test cases to be performed. This testing was done over the course of a full day with each test case needing to be understood first and then performed by the manager. During this testing it was realized that every test case was marked as successful and no errors were found with the functionality.

Both a sample employee and employer were used to test the features directly related to these user roles. Initially a sample set of users were going to be used to test the features however due to time constraints there was only one user for each user role used. In both these cases they followed the same pattern as the manager and tested each feature in turn

and marking it successful or not. By the end of both sets of testing no further errors were produced. The error rate for all 3 user roles remained at 0% and no further development work was needed on the platform.

The results of the testing can be found in the test log included with the files alongside this report with developer testing having an outcome and the user testing just being marked successful or not. Each user also completed an evaluation of the platform and testing via a survey which will be discussed further in sections 5.2.1 and 5.2.2. It is also worth noting at this stage no company social media has been set up so a link has been created but not directly linked to any Facebook profile.

5.2.3 Implementation of CRM System

Implementation of the CRM system is the final step in the software development process and involves the set-up and integration of the system into the company. During this stage the system is hosted on the companies registered web-domain and becomes live for them to use as and when required. The SME chosen for this project is not ready for this stage to be complete therefore this section will discuss how this should take place.

A domain should be registered with a hosting company that gives the SME a custom domain for them to access the CRM through for example www.seraphim-crm.co.uk. Many companies offer a complete package that includes domain registration alongside a web-hosting package. As the SME will not have an IT department or anyone with direct technical knowledge, allowing a company to take care of the hosting of their website is the easiest option for them. One such company that could do this is fasthosts who offer a variety of packages such as:

- Ignite for £2.50p/m: 1 website hosted, 24/7 technical support, 10GB smart SSD web space, 2x 1GB databases, basic SEO tools, 5 x mail extra (2GB) mailboxes and free 1-year domain registration. Over one year this would cost £30 (Fasthosts.co.uk, 2018).

Many companies offer very similar solutions and the SME can choose one that suits their budget and needs when they are ready to host the system live for use. Until then the system will remain as files that can only be accessed via the system they lie on. It is also worth noting at this stage that when this report is complete the developer will do no further work for the SME in terms of the system or implementation as they have chosen not to have it done at this stage.

During implementation the SME must also bear in mind that they do not own the rights to the underlying code behind the system and it cannot be sold on or reproduced without direct permission from the developer. They however do own the rights to the software itself and will be fully responsible for anything directly linked to the software. The company should fully research current data protection laws that apply when the system is live and client data is stored upon it. Information on current laws can be found on www.gov.uk.

When the solution is implemented, the database server must be live and running for the website to work correctly as it relies on a stable connection to the database. Without a database connection errors will be thrown and functionality will be greatly reduced. The implementation also relies on the correct database password being in the DBConfig.php file stored within the PHP folder of the seraphim CRM folder (F:\Honours

Project\Website\Seraphim\PHP\DBConfig.php). To change this password to your own MySQL root user password simply remove 'andrew21' from the 9th line of code and input your own password. Please note when a new install of MySQL has been carried out the root password is usually defaulted to blank unless the installer changes it.

5.3 Evaluation

This section will contain an evaluation of the entire CRM system by the company and the users that tested the system as well as an evaluation of the software development process by the developer.

5.3.1 Evaluation by Company

After the testing of the software was complete the company was asked to complete an evaluation survey on their overall experience (can be found in appendix 1). This survey asked a range of questions on both the overall development experience of their system and the system itself. This evaluation was used to gauge whether their expectations have been met and if the project has been successful for them.

The survey results determined that overall the company was more than satisfied with the usability, the website and the full process as they gave high scores on each point. With scores between 7-10 on these points there is still room for improvement in certain areas however this reflects how successful the project has been. The company commented that they found it hard to know what they wanted to begin with but with the developer doing a design process with them this was easily established in the end. All functionality that they requested in the design stage was achieved and the company is happy this can allow their company to function.

They also noted that they would recommend this process to other companies in a similar position to them and that the software being built in this manner has saved them as a company a lot of money. They feel they have learnt a great deal from going through this process and are in a better position as a company going forward with this software.

5.3.2 Evaluation by Users

The sample users that tested the software also evaluated it in the same manner as the company did via an evaluation survey. They were both asked a shorter range of questions to gauge an understanding of how they felt about the software (can be found in appendices 2 & 3).

Both users within the survey concluded that the software was easy to use with it being easier to use than other software they have used. They both also felt they would not need any training to use the software which concludes that the usability is built with the user in mind. Scores of both 8 & 9 were given on the usability of the software which supports this theory. It was also noted by both users that the features and software would be of use to them and would help them in their user role.

As the usability of the software was something the users highly rated no training will be provided along with the software as it does not seem necessary. However, they did feel there were some added features they would like to see such as being able to change their

passwords on their own rather than via management. One user did also note they would like the option of selecting jobs on the actual job rather than having to message management to register their interest.

From both surveys the users would recommend this software to other users and they are satisfied overall with it. This again proves that the process overall has been successful from all user view points and not just on behalf of the developer or company.

5.3.3 Evaluation of Software

The software itself was a particularly difficult piece of software to build for a few reasons:

- Cost: As the aim was to keep costs to an absolute minimum, solutions had to be sought that were free to use to develop the software. This made the software a lengthy piece to develop as each individual component had to be developed and integrated with the rest of the solution using only the tools available that were free.
- Time: Time was a huge factor in this project as it had to be complete on time due to its nature. As the main developer faced some personal issues in the main stages of development the project was delayed but this was rectified by the developer working harder in the latter stages to complete on time.
- Customisation – As the software had to be custom built for the company no templates were able to be used to develop the software.
- Usability – Software had to be built with the users in mind that they would have little technical experience.

With the above issues being taken into account the overall software was built successfully and to specification. Only the features requested in the design stages were included in the final development and no features were missed out. The database connects seamlessly with the website via PHP code and security was kept in mind with the hashing of passwords and the validation of data being input into HTML forms. By validating the data before it is input into the database there is a reduction on the chances of a SQL injection.

As customer data will be stored in the database the security of this data is paramount in ensuring it is not compromised. The software is built with security in mind and if the software is used for its intended purpose this should not be breached. Customization was the biggest part of this software as the SME needed a solution solely designed with their business in mind. The software was built to ensure it was fully customized to the company's needs and can continue to be customized as the company grows.

Overall the software itself has proven extremely successful and highly rated by both the company and users of the software. However, testing of the software was a difficult process to put in practice with users not willing to spend the time to do any testing hence the limited users used. In any future developments of this type a similar process could be

followed but ensuring there are test users secured and available in the beginning to be involved at each stage.

5.3.4 Evaluation of Software Development Process

The software development process chosen for this development project was the Rapid Application Development (RAD) Process. This process worked particularly well with this project due to the advantage of developing each feature and testing them as they were developed to ensure it was working before moving on to the next feature. Time was a huge factor that was working against this project and as the name of the process states it allowed the application to be developed quickly but not neglecting the needs of the software.

With each phase working in tandem with the other when the design phase was complete a feature was developed and then tested before going back to the design stage to amend the design to suit the testing outcomes and so on. This catches flaws in the system in the development stage rather than at final testing when time may not allow for the flaws to be rectified. The time element of the project eventually became less of a factor as the development process chosen made this easier to manage as the project went on.

Where the process worked against the project was due to the quickness of which each phase moves, this was an advantage to a certain extent but also a major disadvantage. The same process was going around in a circle many times until complete to allow the SME to see the feature as it was developed but then needing further work done. As a development this is a huge disadvantage to the project as it can lose the trust of the SME due to them expecting to use more as they are being shown it. Whereas only the feature complete and ready to be used was being shown at any one time until the final testing stage.

Even with the process having both advantages and disadvantages to it, the process worked extremely well overall. The project was completed on time and to specification with huge credit given to the process being followed.

5.4 Summary

To conclude this chapter the implementation has not been completed as a part of this project therefore advice has been given should the company wish to do this in the future. Testing of the system was completed fully by the developer, the company and 2 sample test users. Due to the size of the system and time constraints no further test subjects were willing to take part. Following the testing of the software an evaluation was carried out and documented which leads in to the recommendations provided in the following chapter.

Chapter 6. Recommendations & Conclusions

6.1 Introduction

Following on from the evaluation there are numerous recommendations made by the developer to answer some key issues raised by the evaluations. This chapter will discuss those recommendations in detail relating to the software development process, the CRM system and any future upgrades to the system. To conclude the report a conclusion including the handover of the CRM system to the company is discussed.

6.2 Recommendations

These recommendations are based on the evaluations in the previous chapter as well as the opinion of the developer and any issues the developer feels may be of importance.

6.2.1 Recommendations on Software Development Process

The development of the CRM was a complex process to follow due to it being developed for a SME. The software process followed fitted the target SME and the timeframe that was set for this development. However, this may not be the case in any future developments of this kind as each SME's requirements will be different along with the project requirements being different.

Taking both these factors in to account it is recommended that a variety of software development processes are researched thoroughly to ensure the one chosen suits the project. Each process goes through a lifecycle and that lifecycle should suit the project and business needs to ensure a successful outcome is achieved. Where the RAD process is chosen there are a few recommendations to take in to account:

- Ensure to talk through the process with the business making sure they understand that this is quicker development process, aimed to put features in to place as and when they are developed. This will allow them to see the benefits of their software as it is being developed.
- Project plans should match the lifecycle of the development process to ensure the process is followed correctly.
- Always remember a software development process makes the development of software more consistent and ensures the entire process is seamless for the developer, the company and the users. So, follow it!

6.2.2 Recommendations on CRM Software

The CRM system developed for the SME takes in to account the business requirements they have at present. Due to this it is highly recommended the software is only used for the purposes it was designed for. To achieve this the SME must ensure that users register for the correct roles they require therefore an employee or employer should never be given admin privileges. The roles developed match the requirements of the types of users in the system and this business rule should never be broken in any circumstances.

It is also recommended that the security of the system is maintained by running the server on a secure environment with adequate firewall and anti-virus protection to detect any

potential breaches to the system. As the database is a SQL coded database it can be subject to such an attack called SQL injections where malicious users inject their own SQL code in to the database to breach the data. Any such breach would have a serious impact on the company and could result in a breach of data protection laws.

Ensure to read and understand the data protection laws and what you as a business can do to protect the data stored within the CRM. At present the CRM application can only delete data via the underlying database as the functionality was not requested any other way. As such the company should ensure to only keep data on this system that is relevant and not out of date. By not following this the company could find themselves facing prosecution for breaking the data protection laws. It will be the responsibility of the company to maintain the data correctly not the responsibility of the developer.

When the system is put in to implementation it is recommended the company ensure someone with the relevant technical experience is used to ensure it is done correctly. Poor implementation can cause the system to be either unsafe or for it not to function the way it was intended to. The recommended approach for implementing the system is outlined before, if this approach is not followed the system may not function as intended and may require further development:

- Download and MySQL workbench with MySQL server.
- Start the MySQL Server
- Import the seraphim database from the seraphim database folder of the USB drive (F:\Honours Project\Seraphim Database\Seraphim)
- Upload the website files to the domain.
- Website will be live in the domain and the servers should be run on a reliable system to ensure there is minimum downtime. Remember, no servers will mean no system.

The system developed was intended to be user friendly and have an easy to use user interface however this may not be the case for all users. Therefore, this company should ensure that users whilst using the system know how to use it correctly. Where users are unsure of functionality the company should provide training if required to these users.

When the system was developed, the intended user audience was less than 100 users therefore, this system was built to handle this audience. As the business grows this system may require further upgrades as outlined in the next section to cope with the additional strain. It is also highly recommended this software is only used within the SME it was designed for. Under no circumstances should it be reproduced or sold to any other environment.

6.2.3 Recommendations on Upgrading CRM Software

There are many aspects of this software that could be upgraded in the future if the SME wishes to. The software was built to allow for the ease of any upgrades to be implemented without changing the entire system. Many features could be implemented and a few will be discussed and recommended as a starting point for the SME.

At present the passwords of users can only be changed via the admin user who has access to update user details. As the user base grows this may not be a function the admin can cope with maintaining for everyone. Therefore, it would be recommended at this stage to add a forgot password feature to the login page along with an update password feature in the update details page. Forgotten passwords should only be updated if relevant security criteria are met to allow the user to update the password. Whereas updating a password should require the current password to then update to the new one.

As described in previous sections the website does not allow for deletion of data from the database as it wasn't a required functionality at this moment. However, it would be recommended that the business reviews this rule regularly and upgrades the software to include this functionality from an admin role to make the process of deletion of data simpler.

The viewing of data on tables at present shows a list of all relevant data within the one table on the one page. As the records grow this table could become extremely large making the process of loading it on to one page and searching through it tedious and long. As the records grow the system should be upgraded to include pagination links which allow for the results to be split in to smaller result sets with links to reach the other results set in the form of page 1, page 2 etc.

At present employees can only request to be assigned to a job via messaging the admin to ask for this to be actioned. What may be a simpler process is adding in of a link on the relevant job that the user can click. When clicked it will automatically send a message to the admin advising the user that clicked the link is requesting to be assigned to the job.

Lastly the statistic results produced for the admin users are a limited result set based on what the company required at present. It is recommended that as further statistics are needed the charts are replicated with the relevant data needed to provide further results. The functionality of the charts can also be extended to include different chart types, change of results based on day, month or year etc. More information in the chart functionality can be found at www.fusioncharts.com.

The set of recommendations laid out above is not extensive nor is it inclusive it merely provides a starting point for any upgrades to be performed based on development recommendations. The software itself can be upgraded in any way the company wishes to but it should only be done by someone with the relevant skills to maintain the integrity and security of the software.

6.3 Conclusions

To conclude the report at this stage the development of the CRM for the SME has been fully developed as outlined in all the chapters previously. This section will recap the aims and objectives outlined in the introduction and comment on how they were achieved:

- Aims
 - There is a cost-effective way to implement a CRM solution for a SME. As outlined in the literature review the cost of current CRM solutions like

salesforce run in to the thousands per month. With the solution sought in this report and developed as a result it came in at a budget of £0 and all that is left for the company to source is a domain server to run the solution on. This domain server could be purchased for as little as £2.50p/m. With a cost of £30 in the first year of implementation including the cost of development this is comparatively lower than the thousands of pounds that current CRM solutions are retailed at.

- The CRM solution was designed based on the findings of this report, with it being custom-built for the SME to the requirements outlined in the design stage.
- Implementation of the CRM was something the SME was not ready for therefore advice was offered within this report in how this should be achieved.

➤ Objectives

- It was shown that there was a role for a cost-effective CRM solution within a SME. The target SME chosen are a start-up company with no start-up funds that require a CRM to carry out their basic business functions. Without this solution they cannot begin to trade as a business.
- Current CRM solutions such as salesforce and SAP were compared in the literature/background review section of this report.
- The design of the CRM was complete fully with advice offered for the implementation as requested by the SME.
- As already advised in this report the implementation was something the SME did not require at this stage therefore the solution has been fully tested and handed over as a file structure for the SME to implement when ready.

6.3.1 Handover of CRM System & Documentation

The final part of this project was to handover the CRM and this report to the SME who will now take responsibility for the solution and any further work or maintenance they feel it requires. A list of files is given below that have been handed over to the CRM in the form of a USB flash drive:

- Seraphim Database Folder – Requires MySQL to be installed and the database to be imported.
- Website – Folder structure of all website files which must be kept within this structure. Either run website on a domain or install a development IDE that supports PHP such as JetBrains PHPStorm IDE.
- B00285373 Jolene Sweeney Final Report
- Evaluation surveys
- Presentation
- Specification form
- Test log

6.4 Summary

To summarize the final main chapter of this report recommendations on the full project from the developmental process to upgrading the software have been provided. The recommendations are purely an academic opinion of the developer and do not need to be followed if not deemed appropriate for the company or future developers. A conclusion of the report has confirmed the structure of the files handed over with this report to the company and the readers of the report. The following chapter is a critical self-appraisal by the author on themselves and discusses how the author feels the project when and how it helped to develop them as an academic.

Chapter 7. Critical Appraisal

This dissertation project proved to be the most rewarding yet challenging piece of work I have undertaken since being at university. When I started the project, I did not fully envisage how in depth this would become and how much work I had given myself to complete. However, with sheer determination I intended to fully complete the project to the best of my ability which I feel I have excelled at doing.

I have undertaken many development projects in the past but none of them where to this magnitude or did they require the skillset I had to develop to achieve the results that I achieved. With languages such as PHP being used within this project I had to learn more about this language before I could begin to use it. This proved the most challenging element of the project as 5 languages where used in conjunction with each other and I had to ensure I did not mix-up one language with another. I do feel however I could have learned more regarding PHP in the beginning stages rather than waiting until development.

Time was not on my side, the workload seemed to keep growing with less and less time to complete it. On top of an ever-challenging deadline I had several other module deadlines to juggle as well as having to take several weeks off due to an operation on my hand. This operation set the project back by at least a month and at one stage I was sure it wasn't going to be complete on time. However, with the support of my supervisor I was given the encouragement to keep going and I worked through the night for 4 weeks to ensure I caught up on the time lost.

Even after initial recovery from my operation my hand has now lost part functionality and simple tasks like typing have become difficult for me to complete. I would not let this impact my ability to complete work to the same standards I am used to producing and ensured I fought through the physical and emotional factors to reach my goal. By doing this I have not only academically excelled in my opinion, but I have also personally excelled my abilities and feel like I have grown as a person.

For most this type of project is a means to an end, for me it marks the end of a long hard road to reach where I wanted to be in life. Therefore, I feel the project reflects this journey for me by having its ups and of course its downs. Many bumps in the road caused me to think I was going to crash and burn but with a bit of corrective steering I was back on the right road and the finished result is what you have just been reading.

The feedback received during this project proved invaluable in ensuring the project was kept on track throughout. The interim report handed in at the mid-way mark received a positive response with only minor points needing reviewed. The next main stage to receive feedback was the presentation which I could not have been prouder of the grade or the comments I received from my supervisor and moderator. Both sets of feedback provided me with the confidence and the drive I needed to continue to complete this project.

Overall, I feel that this project served as a platform for me to push my boundaries and excel at the skills I already had as well as develop new ones. From the beginning to the end of the

project my focus has been in an out of the project but I feel I was able to spend more than enough time required to complete it.

References

- Ec.europa.eu. (2017). Browsers support - European commission. [online] Available at: http://ec.europa.eu/ipg/standards/browsers/index_en.htm [Accessed 6 Mar. 2017].
- Charland, A. and Leroux, B. (2011). Mobile application development. Communications of the ACM, 54(5), p.49.
- Coleman, G. and Verbruggen, R. (2017). A quality software process for rapid application development. Software Quality Journal, [online] 7(1998), pp.107-122. Available at: <https://pdfs.semanticscholar.org/47ec/a10e940fc16e2909d3c77bc26fdaea8f480f.pdf> [Accessed 4 Mar. 2017].
- Salesforce.com. (2017). CRM Company Information - Salesforce UK. [online] Available at: <https://www.salesforce.com/uk/company/> [Accessed 4 Mar. 2017].
- Linked In. (2016). Data Centre Growth. [online] Available at: <https://www.linkedin.com/pulse/data-center-growth-iot-think-beyond-2016-dave-latourelle> [Accessed 5 Mar. 2017].
- Dybå, T., Prikladnicki, R., Rönkkö, K., Seaman, C. and Sillito, J. (2011). Qualitative research in software engineering. Empirical Software Engineering, [online] 16(4), pp.425-429. Available at: <https://link.springer.com/article/10.1007/s10664-011-9163-y> [Accessed 5 Mar. 2017].
- Hubbard, C. (2013). Why Customized CRM is Better - Capterra Blog. [online] Blog.capterra.com. Available at: <http://blog.capterra.com/why-customized-crm-is-better/> [Accessed 4 Mar. 2017].
- Li, C. and Ma, L. (2015). Operation Design of Customer Relationship Management System. 2015 8th International Symposium on Computational Intelligence and Design (ISCID), [online] pp.536-536. Available at: <http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=7469191> [Accessed 6 Mar. 2017].
- Mysql.com. (2017). MySQL :: MySQL Community Edition. [online] Available at: <https://www.mysql.com/products/community/> [Accessed 6 Mar. 2017].
- Naz, R. and Khan, M. (2015). Rapid Applications Development Techniques: A Critical Review. International Journal of Software Engineering and Its Applications, [online] 9(11), pp.163-176. Available at: http://www.sersc.org/journals/IJSEIA/vol9_no11_2015/15.pdf.
- Shop.oracle.com. (2017). Product Category - Oracle Database. [online] Available at: https://shop.oracle.com/apex/product?p1=Database&p2=OracleDatabase&p3=&p4=&p5=&intcmp=ocom_database_oracledatabase [Accessed 6 Mar. 2017].
- Omics Group. (n.d.). Rapid application development. [online] Available at: http://research.omicsgroup.org/index.php/Rapid_application_development [Accessed 4 Mar. 2017].

SAP. (2017). SAP CRM Product List | A-Z Directory. [online] Available at: <https://www.sap.com/uk/product/crm.html> [Accessed 6 Mar. 2017].

Schaeffer, C. (2017). CRM Market Share Report 2017. [online] Crmsearch.com. Available at: <http://www.crmsearch.com/crm-software-market-share.php> [Accessed 1 Mar. 2017].

Shull, F., Singer, J. and Sjøberg, D. (2010). Guide to advanced empirical software engineering. 1st ed. London: Springer, pp.285-311.

JetBrains. (2017). WebStorm: The Smartest JavaScript IDE. [online] Available at: <https://www.jetbrains.com/webstorm/> [Accessed 6 Mar. 2017].

W3.org. (2017). What can you do with JavaScript - Web Education Community Group. [online] Available at: https://www.w3.org/community/webed/wiki/What_can_you_do_with_JavaScript [Accessed 6 Mar. 2017].

Microsoft.com. (2017). Windows Server 2016 Licensing & Pricing | Microsoft. [online] Available at: <https://www.microsoft.com/en-gb/cloud-platform/windows-server-pricing> [Accessed 6 Mar. 2017].

Emersonprocessxperts.com. (2017). [online] Available at: <https://www.emersonprocessxperts.com/wp-content/uploads/2013/10/rapid-application-development.png> [Accessed 18 Nov. 2017].

Baxter, K., Courage, C. and Caine, K. (n.d.). *Understanding your users*.

Varvasovszky, Z. (2017). *A stakeholder analysis*.

Salesforce.com. (2017). What is CRM. [online] Available at: <https://www.salesforce.com/uk/learning-centre/crm/what-is-crm/> [Accessed 23 Nov. 2017].

Capterra.com. (2017). CRM Industry User Research Report. [online] Available at: <https://www.capterra.com/customer-relationship-management-software/user-research> [Accessed 24 Nov. 2017].

Group, D. (2018). About the Apache HTTP Server Project - The Apache HTTP Server Project. [online] Httpd.apache.org. Available at: https://httpd.apache.org/ABOUT_APACHE.html [Accessed 24 Mar. 2018].

Apachefriends.org. (2018). About the XAMPP project. [online] Available at: <https://www.apachefriends.org/about.html> [Accessed 24 Mar. 2018].

Anon, (2018). Feature Tour. [online] Available at: <https://www.fusioncharts.com/tour/> [Accessed 24 Mar. 2018].

Fasthosts.co.uk. (2018). Web Hosting | SSD powered hosting packages from the UK | Fasthosts. [online] Available at: https://www.fasthosts.co.uk/web-hosting?source=GooglePPC&gclid=Cj0KCQjwkd3VBRDzARIsAAdGzMC_yjcMRUkhY_X5gezw4OIWaW650JAxGnOccxXu941Gkm3VcJgiMqUaAswkEALw_wcB [Accessed 25 Mar. 2018].

Appendices

Appendix 1

Evaluation Survey – Company

1. Does the CRM system meet the expectations laid out to you in the design stages?
In the beginning I wasn't sure exactly what I wanted but with a lot of questions being asked my expectations were laid out in the design stages and then exceeded in the final development.
2. Does the CRM involve all functionality requested?
Yes – It allows my company to be started as without this software I had no way of running my company to begin with.
3. How would you rate the usability of the software from 1-10?
7
4. Would you recommend other companies to have their CRM system developed this way?
Yes, I would because software that could have cost me thousands of pounds has cost me nothing to be built and allowed me to start my company without any initial expense on this.
5. How satisfied are you with the custom-built website of the CRM from 1-10?
10 as the website is exactly as agreed only better and fits with the company theme.
6. Are you satisfied with the registration and login process of the software?
Yes, in the beginning this is all we will need.
7. Are there any recommendations you would make to future companies looking to follow this process?
I would recommend they find either a student looking to do a similar project like this or a developer looking to extend their portfolio and willing to keep costs to a minimum for them. It has been a great process for the company as well as a learning curve as I had no idea to begin with how any of this would even happen.
8. Are there any recommendations you would make to the development of this system?
Anyone looking to have similar software should sit down and consider what they need first as this was something I was unsure of.
9. Any further comments?
I would like to thank the developer for doing this for my company in the way they have and allowing me to have this software at no cost to myself.
10. Overall how satisfied are you with the full process you have been involved in from 1-10?
9

Appendix 2

Evaluation Survey – User

1. How easy was it to use the software?
After looking around the various pages the software was easy to use
2. How user-friendly would you rate the software from 1-10?
8
3. Do you think the software would be useful to you as a user?
Yes, I can use this as a central place to find work
4. Do you think the features of the software match what you would need?
Yes, I can find jobs and keep in contact with colleagues etc.
5. Are there any recommended features you would like to see?
I would like to be able to sign up for a job direct on the job itself rather than messaging management, and change my password myself if forgotten.
6. Do you feel like you would need training to use the software?
No
7. Overall how satisfied are you with the software?
Very
8. Would you recommend it to other users?
Yes I would recommend it to my colleagues.

Appendix 3

Evaluation Survey – User

1. How easy was it to use the software?
Compared to other software used this is comparatively easier
2. How user-friendly would you rate the software from 1-10?
9
3. Do you think the software would be useful to you as a user?
Yes, I can log jobs I need covered easily and allow someone else to find me employees
4. Do you think the features of the software match what you would need?
Yes
5. Are there any recommended features you would like to see?
Change of passwords.
6. Do you feel like you would need training to use the software?
No
7. Overall how satisfied are you with the software?
Really satisfied.
8. Would you recommend it to other users?
Yes.