Design and Implementation of a Business Automation Solution

MSc Dissertation in Information Technology
(Software Engineering)

Author: Ullas Unnikrishnan Gupta
Reg. No: 091581794
Supervisor: Stephen Gill
Dedicated to my loving parents
DISCLAIMER

I hereby declare that the work presented in this report is done by me. All the references used while preparing this report have been stated appropriately. No other resources or documents have been used when preparing this report. I understand the consequences of not abiding by the rules of plagiarism. All the references have been furnished in the References section of this document.

Signature ____________________
ACKNOWLEDGEMENT

At first I would like to thank the all mighty for his blessings without which this dissertation would not have been possible. I would also like to sincerely thank my supervisor Prof. Stephen Gill for his support and valuable suggestions during the course of this dissertation. Finally, I would like to thank my parents and friends for their support. I would also like to specially Mr. Yad Swdad Tahir, my classmate for his suggestion to help me make this dissertation a success.
ABSTRACT

This dissertation is about designing and implementing a business automation solution for a company in Sharjah, UAE named Avalon Office Equipments. The company deals with distribution of office supplies to its customers located in various locations in the GCC like Oman, UAE, and Qatar etc. The business solution involves streamlining business data inflow through web and email, automating mailbox monitoring and cost analysis creation. It also allows quotation and invoice creation, database data modification along with graphical report generation with export options. This dissertation also involves the creation of a mobile application for quick and easy access to business data when away from office.

During the length of this dissertation, the current business process is studied and its weaknesses are identified. After the weaknesses are identified, the appropriate IT components required for improving business process efficiency and nullifying the weaknesses are finalised. These components are then designed, developed and tested. The developed system is then evaluated and critically analysed. The shortcomings are highlighted and future work is recommended.
# TABLE OF CONTENTS

DISCLAIMER .......................................................................................................................... 3  
ACKNOWLEDGEMENT ........................................................................................................... 4  
ABSTRACT ............................................................................................................................ 5  
TABLE OF CONTENTS ......................................................................................................... 6  
TABLE OF FIGURES ............................................................................................................ 8  
LIST OF TABLES .................................................................................................................. 10  
CHAPTER 1: INTRODUCTION ............................................................................................ 11  
  1.1 Introduction .................................................................................................................... 11  
  1.2 The Limitations and Solutions – an Outline ............................................................... 11  
CHAPTER 2: PRE-DEVELOPMENT STUDY & ANALYSIS ............................................... 14  
  2.1 Current Business Process .............................................................................................. 14  
  2.2 Problem Definition – Business Process ....................................................................... 16  
  2.3 Dissertation Goals ........................................................................................................ 17  
CHAPTER 3: LITRATURE REVIEW ................................................................................. 19  
  3.1 Proposed System Architecture ...................................................................................... 19  
  3.2 Related Available Technologies .................................................................................. 21  
    3.2.1 Desktop Application ............................................................................................... 21  
    3.2.2 Web Application Interface ................................................................................... 25  
    3.2.3 Mobile Application ............................................................................................... 30  
  3.3 Analysis of Technologies .............................................................................................. 31  
    3.3.1 The Desktop Application ...................................................................................... 31  
    3.3.2 The Web Interface ............................................................................................... 32  
    3.3.3 The Mobile Application ....................................................................................... 33  
  3.4 Software Development Model .................................................................................... 34  
    3.4.1 The Linear Sequential Model ............................................................................. 34  
    3.4.2 The Prototyping Model ....................................................................................... 34  
    3.4.3 The Rapid Application Development Model .................................................... 35  
  3.5 Risk Management ......................................................................................................... 36  
    3.5.1 Risk Identification ............................................................................................... 36  
    3.5.2 Risk Assessment ................................................................................................. 36  
    3.5.3 Risk Management ............................................................................................... 37  
  3.6 Professional, Legal, Social and Ethical Issues ............................................................ 38  
    3.6.1 Professional Issues .............................................................................................. 38  
    3.6.2 Legal Issues ......................................................................................................... 38  
    3.6.3 Ethical Issues ....................................................................................................... 38  
    3.6.4 Social Issues ....................................................................................................... 38  
  3.7 Conclusion .................................................................................................................... 39  
CHAPTER 4: SYSTEM DESIGN ........................................................................................... 40  
  4.1 Introduction .................................................................................................................... 40  
  4.2 Desktop Application Design .......................................................................................... 40  
    4.2.2 Textual Description .............................................................................................. 41  
    4.2.3 Class Responsibility and Collaborator diagram (CRC diagram) ....................... 46  
    4.2.4 Class Diagrams ................................................................................................... 48  
  4.3 Mobile Application Design ........................................................................................... 55  
    4.3.1 Use Case Diagram ............................................................................................. 55
4.3.2 Textual Description........................................................................................................55
4.3.3 Class Responsibility Collaborator Diagram (CRC Diagrams).................................58
4.3.4 Class Diagram ..............................................................................................................59
4.4 Servlet Classes Design...................................................................................................60
   4.4.1 Use Case Diagram......................................................................................................60
   4.4.2 Use Case Diagram......................................................................................................60
   4.4.3 CRC Diagram...........................................................................................................63
   4.4.4 Class Diagram .........................................................................................................65
4.5 Database Design .............................................................................................................67
   4.5.1 Conceptual Database Design ..................................................................................68
   4.5.2 Logical Database Design .........................................................................................69
4.6 Conclusion .......................................................................................................................70

CHAPTER 5: IMPLEMENTATION ......................................................................................71
5.1 Implementation Challenges ............................................................................................71
5.2 Implementation Highlights .............................................................................................76
   1. COMPONENT BASED DEVELOPMENT .......................................................................76
   2. FLEXIBLE DATABASE CONNECTOR ..........................................................................76
   3. FUTURE ENHANCEMENTS .........................................................................................77
5.3 Conclusion .......................................................................................................................78

CHAPTER 6: TESTING ......................................................................................................79
6.1 Desktop Application ........................................................................................................79
   6.1.1 Testing DML functions ............................................................................................79
   6.1.2 Testing Automatic Cost-Analysis Creation .................................................................85
   6.1.3 Testing Create Quotation .........................................................................................88
   6.1.4 Testing Send Quotation ............................................................................................89
   6.1.6 Testing Report Export ..............................................................................................92
   6.1.7 Testing Non-Super user functionality ......................................................................93
6.2 Web Application .............................................................................................................96
   6.2.1 Testing Appearance on Different browsers ..............................................................96
   6.2.2 Testing XHTML Validation ....................................................................................98
   6.2.3 Testing functionality ...............................................................................................99
   6.3.1 Functionality Test .................................................................................................101
   6.3.2 Speed Test .............................................................................................................102

CHAPTER 7: EVALUATION ..............................................................................................103
7.1 Product Evaluation .......................................................................................................103
7.2 User Evaluation .............................................................................................................104

CHAPTER 8: CONCLUSION AND FUTURE WORK ....................................................106
8.1 CONCLUSION ...............................................................................................................106
8.2 FUTURE WORK .............................................................................................................108

References .........................................................................................................................109

APPENDIX A: SOURCE CODE .........................................................................................111

APPENDIX B: EVALUATION FORM .............................................................................112

APPENDIX C: DB CONFIG XML ....................................................................................113

APPENDIX D: COST-ANALYSIS, QUOTATION, INVOICE (SAMPLE) .........................114
### TABLE OF FIGURES

- Figure 1: Current Business Process ................................................................. 14
- Figure 2: Proposed System Architecture ............................................................ 19
- Figure 3: The Linear Sequential Model [31] ....................................................... 34
- Figure 4: The Prototyping Model [31] ................................................................. 34
- Figure 5: The Rapid Application Development Model [31] ............................... 35
- Figure 6: Risk Assessment – Impact vs. Probability ........................................... 36
- Figure 7: Use Case Diagram – Desktop application ......................................... 40
- Figure 8: CRC Diagram – Desktop application ................................................. 47
- Figure 9: High Level Class Diagram – Desktop application ............................. 48
- Figure 10: Mail Manager Class Diagram ........................................................... 49
- Figure 11: Logger Class Diagram ..................................................................... 49
- Figure 12: Database Manager Class Diagram .................................................. 50
- Figure 13: PCS Class Diagram ......................................................................... 51
- Figure 14: Reports Manager Class Diagram .................................................... 52
- Figure 15: Cost Analysis Class Diagram ........................................................... 53
- Figure 16: Quotation Class Diagram ................................................................ 53
- Figure 17: Invoice Class Diagram ................................................................... 54
- Figure 18: Use Case Diagram – Mobile Application .......................................... 55
- Figure 19: CRC Diagram – Mobile Application .............................................. 58
- Figure 20: Class Diagram – Mobile Application .............................................. 59
- Figure 21: Use Case Diagram – Servlet ............................................................ 60
- Figure 22: CRC Diagram – Servlet ................................................................. 64
- Figure 23: Class Diagram – Servlet ................................................................ 65
- Figure 24: Class Diagram 2 – Servlet .............................................................. 66
- Figure 25: ER Diagram – Database design ...................................................... 68
- Figure 26: Products Table before Addition ....................................................... 79
- Figure 27: Application - Add Product ................................................................. 80
- Figure 28: Products Table after adding product .............................................. 80
- Figure 29: Update product ............................................................................... 81
- Figure 30: Application - Product Updated ........................................................ 81
- Figure 31: Products Table after product Updated ............................................ 82
- Figure 32: Application - Delete Product ............................................................ 82
- Figure 33: Products Table after product deleted .............................................. 82
- Figure 34: Application - Search products .......................................................... 83
- Figure 35: Application - Search Results ............................................................ 83
- Figure 36: Application - No result found ........................................................... 84
- Figure 37: Enquiry Form filled in ..................................................................... 85
- Figure 38: Mail Arrives .................................................................................... 85
- Figure 39: Mail read and Notification ............................................................... 86
- Figure 40: Auto Created Cost Analysis ............................................................. 87
- Figure 41: Application - Create Quotation ....................................................... 88
- Figure 42: Created Quotation .......................................................................... 88
- Figure 43: Application - Send Quotation ........................................................... 89
- Figure 44: Auto Create e-Mail .......................................................................... 90
- Figure 45: Create Quotation - Error Notification ............................................ 90
- Figure 46: Application - Generate Report ....................................................... 91
- Figure 47: Application - Generate Report (2) .................................................. 91
- Figure 48: Generated Report ........................................................................... 92
Figure 49: Application - Export Report .................................................................92
Figure 50: Exported Report ..................................................................................93
Figure 51: Application - Login Form.....................................................................93
Figure 52: Application Menu (non-Super User) ......................................................94
Figure 53: Application - Products Menu (non-Super User) .................................94
Figure 54: Application - Cost Analysis Menu (non-Super User) .......................95
Figure 55: Web Page in Internet Explorer ..............................................................96
Figure 56: Web Page in Firefox ...........................................................................97
Figure 57: Web Page in Opera .............................................................................97
Figure 58: XHTML validation result ......................................................................98
Figure 59: Web Enquiry Form Filled .................................................................99
Figure 60: Mail from web application .................................................................99
Figure 61: Created XML file attachment .............................................................100
Figure 62: Mobile Application .........................................................................101
LIST OF TABLES

Table 1: Functionality Comparison................................................................. 103
Table 2: Success Criteria Table ................................................................. 107
CHAPTER 1: INTRODUCTION

1.1 Introduction

In today’s highly competitive business world, Information Technology is an aspect that companies cannot do without. Large businesses today tend to depend on the power of IT to make critical decisions, manage finances, human resources, supply chain etc. The selection of the most appropriate IT solution for a business is a critical factor that determines the success of implementing IT solutions for the business. This is because the cost of implementing and supporting the IT infrastructure must not overweigh the revenues of the organisation. It is also very important that the IT solution implemented is flexible enough to evolve with the ever-changing business environment. The literature review covers the study done in order to design and implement an efficient and effective IT business solution that automates cost-analysis, quotation and invoice creation (pre-sales functionalities). The business solution also includes report generation and graphical report viewing. The implementation is done keeping in mind the balance between cost and productivity.

1.2 The Limitations and Solutions – an Outline

CISCO defines the 5 ways to improve business efficiency as the following

1. Providing Easy Access to Information
2. Delivering Anywhere, Anytime Access
3. Creating Effective Business Process with Partners
4. Ensuring Regulatory Compliance
5. Enhancing Employee Collaboration. [35]

Apart from these one can also understand that if the output of work done per input effort is increased, efficiency is increased.

The primary focus of this dissertation is to increase business process efficiency through reducing effort through automation. Some of the issues addressed in this dissertation are as follows

- Time and effort spent in reading generic emails and receiving multiple phone calls related to product enquiry.
- Limitations in current software such as updating product details in the database, sending emails with the newly created cost analysis as attachment etc.
- Updating current website of the company to overcome its limited functionalities i.e. the enquiry form in the website could only send a general email.
- Lack of access to critical data outside the office environment. For example, in a meeting with clients.
In order to overcome the above-mentioned limitations, and to improve business efficiency, information technology components such as a mobile application, a web application and a desktop application will be used. Some of the advantages of using the information technology components are listed below

• Business data inflow and capture will be streamlined thus avoiding the need to attend multiple phone calls and read general emails. This will allow improving business process efficiency, as the employees can use their time for other constructive work.
• Allow complete and easy access to information in the database as well as archive data such as old cost analysis, invoices etc.
• Automate redundant work such as reading emails of enquiry, creating cost analysis using the old software, email generation with the appropriate attachment etc.
• Allowing mobile access to business data anywhere and at anytime.

More details of the business process, its limitations and the solutions proposed and goals of this dissertation are discussed in the next chapter.
The following section consists of a brief description of all the chapters in this dissertation.

**Chapter One**

An introduction is to the dissertation is given. The problems and solutions of the current business process are briefly discussed at a high level.

**Chapter Two**

A study is done of the current business process. A thorough understanding of each step is gathered. The business process is analysed and gaps where IT components can be plugged in to improve business efficiency are identified. Next the components that will constitute the business solution are finalised.

**Chapter Three**

A study done of various technologies that can be used for the development of this business solution. Further, a study of the type of development methods available is done.

**Chapter Four**

The fourth chapter discusses the design of the system in detail. This includes Use Case diagrams, CRC diagrams, ER diagrams etc.

**Chapter Five**

This chapter highlights the implementation challenges. Here the technical difficulties faced during development of the application along with the solutions and work around used is highlighted. The main highlights of this dissertation are also furnished.

**Chapter Six**

This chapter discusses the testing done of the components.

**Chapter Seven**

The fifth chapter discusses the evaluation done along with the results and interpretation of the same. This chapter also contains a comparison of functionalities between the IT solution to be developed and existing softwares in the market.

**Chapter Eight**

Finally, the last chapter concludes this dissertation and also discusses the scope for enhancing this business solution.
CHAPTER 2: PRE-DEVELOPMENT STUDY & ANALYSIS

After obtaining a brief idea about the objectives of this dissertation in the last chapter, this chapter throws more light on the current business process, its limitations and the solutions to overcome these limitations. The solutions will be the goals of this dissertation.

2.1 Current Business Process

The company for which the IT business solution is being built is an office equipment distributor who caters to customers in UAE, Oman, Qatar etc. The company is named Avalon Office Equipments. The following figure shows the current business process.

![Current Business Process Diagram]

**Enquiries Received**

The current business process of the company begins at the enquiries received about the prices and availability of various products. These enquiries are usually sent as general emails or received by telephone calls. Thus there is a lack of a standard format or structure for the enquiries.

**Cost Analysis Generated**

The office staff then use accounting software named Tally to generate an initial cost analysis (pre-quotation). The cost analysis would contain details of the cost price, selling price, profit margin on each good etc. along with the total sales and margin for this order. Cost price is defined as the total value of the commodity, which replaces the price of production and labour cost involved [1] whereas the selling price is the price at which a good is sold.
Quotation Created

After the cost analysis is created, and the costs and margins are satisfactory and approved, a quotation is created. A quotation is an official document that is sent to the customer quoting the prices that that company is committed to for the specified goods or services. A quotation can contain the overall cost(price), break up of the costs (prices) involved, the time period for which the quotation is valid etc. [2]

Quotation Sent to Customer

This quotation is then sent out to the customer. It is possible that the customer may come back to request for reduced prices. In this case the cost analysis is re-created. Each time a cost analysis or quotation is created, it is printed out, scanned and then sent as the existing software lacks a ‘save as’ option as well as a mailing option.

Invoice Generated

Once the quotation is accepted by the customer, the suppliers for these goods are contacted and requests for the goods are placed. After the order of goods is received, it is then sent to the customer and a pro forma invoice is generated. A pro forma invoice is a bill of sale that is sent by the seller to the buyer during delivery of goods.[3]

Annual and Quarterly Report Generated

The company also creates reports at the end of the month to calculate the total sale, total profit or loss, the revenue generated etc. These reports are critical as they are analysed to determine the current position of the company and also support decisions to be made regarding the company’s growth.
2.2 Problem Definition – Business Process

After analysing the business process, the following limitations or areas where IT components could be plugged in were observed:

- Communication via phone – not reliable, very laborious and time consuming. There is also possibility for errors while noting down the products enquired.

- Communication via email – not structured. It is possible that the email could be large but the number of products being enquired is just one.

- Continuous monitoring of mailbox required as important and urgent e-mails can come at any time during office times.

- Accounting Software currently in use does not have flexibility. Updates to database are not possible. Hence if there is a price change for one product, that must be deleted and recreated within the current accounting software.

- Important mails and queries that come post working hours are missed at times and hence require monitoring even from home.

- The company has an enquiry form within the website but it only creates a general email.

- Reports generated are on paper. Hence time and effort is required to analyse the same.
2.3 Dissertation Goals

In order to mitigate the limitations mentioned in the previous section, to automate redundant work and to improve business process efficiency, the business solution being developed as a part of this dissertation would involve the following stages and functionalities.

1. Design and development of a database structure that can be used for this business.

2. Structuring the incoming enquires and requests using an enquiry form. Two enquiry forms would be created. The first one would be for users wishing to send enquiries via eMail. This enquiry form would be created in MS Excel (ref Appendix F). The second would be an online enquiry form (ref Appendix E). This would involve altering the existing online enquiry form to allow selecting products from the database as well as entering product details that are not available in the database.

3. Developing software using software design and development strategies and patterns. This would be a desktop application that would allow and perform the following functionalities.
   
   a. **Automatically** performing the following (Step i – v)
      
      i. Continuous monitoring of email inbox.
      ii. Reading relevant emails and picking up relevant attachments.
      iii. Reading the data in the attachments. (Excel form – sent via email and XML files sent via web enquiry form).
      iv. Creation of Cost Analysis (Pre-Quotation) which would include the total cost, individual cost, profit margin % etc. The supplier with the lowest supplier price is chosen to create the cost analysis for each product.
      v. Alert when a cost analysis is created (similar to the alert that comes when a new mail arrives in Google Mail).
   
   b. Options to alter the automatically created cost analysis. The alteration could be to the cost price, selling price, quantity etc. It is possible that one supplier is unable to supply the total requested amount (quantity) of goods. The desktop solution will allow addition of suppliers (who supply the remaining quantity) for the particular product. This added supplier could have a different supplying price. The application would automatically re create the cost analysis with two suppliers for the same product and provide new values for Total cost, Total sale, Profit Margin etc.

   c. Options to create a quotation from the cost analysis in the same format that the current accounting software creates. Hence there would be seamless integration of this software with business. If there are two suppliers for a product, the quotation would contain one entry for the product with the total quantity and the quoted price.
d. Options to automatically send created quotation without the need to create a mail and attach the file. Mail creation and attachment would be automated.

e. Options to create a final invoice from the quotation.

f. Options to automatically send created invoice without the need to create a mail and attach the file. Mail creation and attachment would be automated.

g. Add, Update and Delete the Products, Suppliers, and Customers data.

h. Generation of reports critical for business analysis – Sale report per customer, per goods sold, between a time period etc. This would provide an animated graph to allow visualisation and easy interpretation.

i. Option to export reports. Reports can be exported to excel sheets.

4. Design and development of a Web Interface for enquiry. This will involve redesigning the existing enquiry form in the live website to facilitate selection of the products being enquired. The website will also allow users to key in product details that don’t exist currently.

5. Design and development of a mobile phone application. The following are the uses.

   a. The mobile application will allow access of business data when away from office. It will allow searching for quotations and viewing the details of products quoted. It will allow searching for invoices and also get details of suppliers who supplied for a particular invoice. Also details of products can be enquired along with details of who are the current suppliers for a particular product.

   b. The mobile phone will also be useful to receive SMS from the desktop application if an enquiry arrives post work hours.

To conclude, this chapter introduced the current business process of Avalon Office Equipments. A study was done and the limitations were identified. The project goals were also furnished. In the next chapter a detailed study and analysis of the technologies that can be used to implement the same is done.
CHAPTER 3: LITERATURE REVIEW

After discussing the current business process of the company, and highlighting the limitations and project goals in the previous chapter, this chapter details the study done in order to choose the relevant technologies to implement the project goals. This chapter also explains the system architecture for the IT business solution being developed.

3.1 Proposed System Architecture

The initial idea for sketching the system architecture for the IT solution involved the existence of two databases. The reason for this was that one database would cater to the web and mobile applications where the table structure would facilitate rapid reading of data since reading is the predominant activity done by these two components and the other database would cater to the desktop application where the table structure would facilitate writing of data. However this idea was not implemented as it would increase the cost of maintenance. This would also increase the complexity of the application. The risk of data inconsistency is also high in this case.

The next idea had an architecture where the mobile application and the web application would interact with the database directly without the use of the servlet container server (as is the case in the final architecture – shown below). The disadvantage of this architecture is that the mobile application, web application and database would be very tight coupled. Hence expansion would be difficult. Also any change to the database could affect both components.

The third and final system architecture was as follows

![Figure 2: Proposed System Architecture](image-url)
As can be seen from Figure 1, the staff systems would connect to the database directly to perform updates. The automated pre sales cost analysis creation would run in only one system. This is done to avoid redundant/simultaneous reading of emails and avoid duplicate creation of cost analysis. It is the system that has the automated pre sales cost analysis creation running that would read and process emails for the enquiries and also connect to the email-SMS (Short Message Service) gateway to send SMS if an enquiry comes post office timings. The created cost analysis, quotations, and invoice files would be stored in a shared folder.

The web interface would be hosted on the existing web server. This server would connect to the servlet to retrieve data from the database. This 3-tier architecture is selected to separate the presentation layer, application layer and the back end. This model would also help in future enhancements to any section i.e. presentation, business logic or back end without affecting the other.

The mobile application would also connect to the servlet to retrieve data from the database. Thus the mobile application would not require APIs to connect to the database. It would only need the HTTP request response capabilities i.e. a GPRS or 3G connectivity.

The following section describes the technologies that can be used to develop the 3 components (mentioned above) with respect to the system architecture.
### 3.2 Related Available Technologies

#### 3.2.1 Desktop Application

##### 3.2.1.1 Microsoft DOT.NET 3.5

Microsoft DOT.NET 3.5 is a framework developed by Microsoft for developing applications such as Windows Application, Web Application, Web Service etc. Along with this framework Microsoft also developed a language name C#. This language was developed to eliminate issues that existed in languages such as Java, C++, Objective C etc. Features provided by the DOT.Net framework are as follows: [4]

- **Multiple language support** – The DOT.NET framework supports multiple languages such as C#, VC++, J# etc. It also supports other languages that comply with CLS (Common Language Specification). CLS is a sub set of CTS (Common Type System), which defines all the types and constructs. It also specifies the way these types interact with each other. [5]

- **Common runtime engine** – This framework is built on a common runtime engine known as the CLR (Common Language Runtime). Thus a programmer may choose to program in any language that he is comfortable with and it would still run as the framework would recognize and interpret the different types. [5]

- **Comprehensive base class library** – The base class library in DOT.NET provides an object model or a template that can be used by any DOT.NET compatible languages. [5]

- **Simplified deployment model** – DOT.NET allows multiple versions of dll (Dynamic Link Library). This allows programmers to keep multiple versions of the same .dll file and revert changes if required. [5]

In order to develop a desktop application (known as Windows Form) in DOT.NET, the framework offers an API (Application Programming Interface) named `System.Windows.Form` that facilitates the rapid development of a Windows Form. This API contains all the classes or types necessary for developing an interactive desktop GUI (Graphical User Interface). The classes provided by this API can be categorised as follows. [6]

- **Core Components** – The Windows Form API provides classes and types that are used to perform and manage the basic operations of a windows form. These types manage the form as an application and also control its behaviour with respect to the underlying operating system. [6]
• Controls – These are types that help make the GUI interactive. Some examples of controls are Buttons, List boxes, Radio Buttons etc. The Windows Form API comes with many such controls that helps make a windows application very user-friendly and easy to build and develop. [6]

• Components – Various additional components have been added with the Windows Form API. Some of them such as timer, image list etc run at the back ground and cannot be used or seen by the user, however they play a vital role in the development process. Some components are also visible. There are tool tips, error message controls and also validations. These aid users and promote overall user experience. [6]

• Dialogue Boxes – Dialogue boxes are ready-to-use components within the Windows Form API. These are File Browser, Printer Dialogue boxes etc. These allow developers to use them without working on the coding of their functionality. [6]

In addition to these API DOT.NET also provides a library named System.Drawing. This library allows programmers to create and manage 2-dimentional graphics within the desktop GUI. Along with the DOT.NET 3.0 framework Microsoft also released another package to improve the visual appearance of its windows form. This package is called WPF (Windows Presentation Foundation). WPF is discussed in the next section.

3.2.1.2 Windows Presentation Foundation

WPF or Windows Presentation Foundation was launched by Microsoft with the DOT.NET 3.0 version. This was to overcome the existing challenges for developers with respect to creating more interactive features such as video streaming, 2D graphics, 3D graphics etc. Before the introduction of WPF, programmers were forced to use third party APIs for obtaining these features. This not only was not very convenient, it also forced the programmers to change programming style based on the APIs they were using as each API had its own diverse nature. [7]

WPF completely eliminates this challenge faced by the programmers by providing APIs for video streaming, 2D and 3D graphics etc. WPF also introduces a mark up language named XAML (Extensible Application Markup Language). This allows total separation of the appearance of the UI with the business or working logic within. XAML is also compatible with other technologies that understand markup language. [7]

Apart from the above-mentioned features, WPF also provides other features such as the ability to select a particular “theme”. It also provides the ability to dynamically bind the data with the UI to get real time UI change with respect to data changes. One of the best features within WPF is the availability of vector graphics. This allows the GUI to automatically resize itself in accordance with the screen resolution of the system on which it is currently working. [7]
WPF thus combines the power of flexible programming that DOT.NET provides and bleeding-edge display technology to enhance user experience with desktop applications.

3.2.1.3 Java Desktop Applications

Java came into being in the early 1990s. The initial goal of this language was to develop applications that were platform independent. For example it was to be used in microwave ovens, remote controls etc. With the introduction of Java, the issue with C++ of having a specific compiler for each target which was expensive was mitigated. Java then evolved to a language for the internet and offered features like portability, security etc.[11].

Features offered by Java are as follows.

- **Security** – Java applets running on the web browser does not have access to local files. The environment available to the Java code is limited by the JVM (Java Virtual Machine).

- **Interpretability** – Java code is converted into Byte Code. This can then be interpreted and run on any operating systems as long as the JVM is installed. Thus Java desktop applications and other embedded applications can perform on multiple heterogeneous operating systems.

- **Portability** – The same reason for security results in portability of Java code. The byte code and JVM allow applications to be portable and reusable over various platforms.

With respect to desktop applications, Java has a package named AWT (Abstract Window Toolkit). This was initially developed and used to create Java applets. Java Applets are small executable programs that can be downloaded, has a user interface and runs within a Java compatible web browser. The AWT however can also be used to create many stand alone GUI applications that can run on systems that have the JVM installed. The AWT contains various classes in order create the user interface and also handle user events. Some of the classes that are available with AWT are Button, Checkbox, Listbox, Color, Choice, Container, Panel, Dialogue etc. [12]

AWT however had a few drawbacks or limitations with respect to its functionality and ability to meet users’ dynamic requirements. Java then introduced another package named Swing. This package contained more classes that could perform other functions. Some of the classes added in Swing are tabbed panes, trees, scroll panes etc. The components such as JScrollPane, JButton etc developed in Swing are developed completely in Java and are thus platform independent. AWT on the other hand was developed for specific platforms. The Swing package also provides added functionality to classes (GUI features) that are already available with the AWT package. One example of the additional advantage that Swing provides is the ability to use images on buttons (known as JButton in Swing) and alter them on particular user events. Thus this section discusses the possibilities and some features of using Java as the development environment for the windows application. [13]
3.2.1.4 C++ with Qt

Qt today is a fully grown application development framework that is now under the control of Nokia. It began as a GUI toolkit i.e. a set of classes that is used to create interactive user interface.[14] Qt in conjunction with C++ increases the breadth and capability of C++. It helps nullify the disability of C++ of not having many libraries. C++ with Qt however can be used to develop cross platform applications for variety of uses such as mobile phones, desktops etc.

C++ is a compiled language. This means that the C++ code is not interpreted like Java or converted to and intermediate language like the DOT.NET platform does. Each source code file is compiled and a object file is created. The source code file can have as many classes as required unlike Java where each class is one file. After the files are complied by the complier and then converted into object files, these are then combined to form the executable file. This is done by a program named ‘Linker’. [15]

Since C++ is a compiled language, it would be an ideal if the target environment is one where processing power of the system is low and also one where memory footprint could be a decisive factor.

With respect to developing a windows application, the combination of C++ with Qt offers various features.

- Libraries - These libraries help create interactive and user-friendly GUIs. Some of the available libraries are QDialogue, QPushButton, QLabel etc.

- Support for Extended Features – There exists support for extended features such as using ActiveX objects for windows and other platform specific functions.

- Appearance Alteration – The Qt framework allows developers to incorporate changes to the appearance of the GUI by using inheritance and also in the higher versions of Qt (versions above 4.2) HTML CSS can be used.[16] This approach with then split the programming logic with the appearance similar to WPF (discussed earlier).

- Support for 2D and 3D graphics – By combining OpenGL with QPainter from Qt framework, 2 and 3 dimensional graphics can be created. OpenGL is a cross-language, cross-platform framework used to create 2D and 3D graphics.[17]
3.2.2 Web Application Interface

3.2.2.1 PHP Hypertext Pre-processor

PHP is an open-source scripting language that is developed by volunteers from all over the world. It is a simple and effective language that is backed by availability of various APIs to perform functions such as email, database connections etc. Some of the unique features are as follows:

- **Portability** – PHP is a cross platform technology. Thus a PHP script that works on a windows platform will work on a Unix platform without any significant issues. This portability feature makes it very useful in cross-platform organisational environments.[18]

- **Ease Of Use** – PHP is a rapid development language. Its weak typing and extensive documentation of over 5000+ functions makes it very easy to develop effective and efficient web applications.[18]

- **External Application Support** – PHP supports various third-party databases like Oracle, MySQL, SQL Server 2008 etc. It also supports XML parsing using XPaths. [18] This is important as technologies are evolving with XML as their foundation.

3.2.2.2 Active Server Pages (ASP)

ASP or Active Server Pages is a server side scripting language developed by Microsoft. This server side scripting language works on IIS (Internet Information Services) which is Microsoft’s web server. ASP was developed to improve the interaction on the web. [xx] Features of ASP are provided below.

- **Component Availability** – ASP is packaged with various components that make interactive web development easier. Some of these components are Request and Response Object that can be used to send and receive HTTP request and response data, browser capability components which facilitates detection of the users’ browser and thus dynamic adaptation based on the browser type, permission checker component which allows restricting the web site user to access to certain files only. [19]

- **Cross Platform Functionality** – Chili!Soft’s Chili!ASP is an alternative ASP solution that can be run on operating systems such as HP-UX 11.0, IBM AX 4.3.1, Sun Solaris 2.5.1 etc. With this solution an ASP application developed in Window’s NT can be exported into another operating systems without issues. [20]
3.2.2.3 Java Server Pages (JSP)

JSP or Java Server Pages is Sun’s version of server side scripting language. It makes use of the power of Java to create efficient and powerful interactive web applications. JSP is built on servlets and designed to increase the efficiency. [21] JSP is the only server side scripting language that is strongly typed and therefore it improves readability and efficiency. JSP offers various features. They are discussed below.

- **Functionality** – JSP offers various functionalities as it uses the power of Java. JSP can use JDBC to access databases. It can also use Java Beans components and servlets to achieve other functionalities. JSP also has features that allow developers to work with XML data. [21]

- **Ease of development** – Unlike Java Servlets, JSP does not embed HTML code into the JSP program. Thus developers can develop JSP and HTML independently and then combine the same to obtain a dynamic web application. JSP also avoids the need for multiple re-compilation. Once compiled it automatically re-compiles if any changes are detected. [21]

3.2.2.4 CGI Scripting

CGI or Common Gateway Interface is an interfacing technology that allows the external world to interface with web servers through the HTTP protocol.[22] was designed to make the web more interactive. It is designed in such a way that it can be used on any operating system and developed in any language such as C++, Perl, Python etc. CGI scripts are compiled and kept in the cgi-bin folder of the server. These are activated when cited by an HTTP request.[23]

Some of the positive features are as follows

- **Operating System Independence** – The CGI programs are independent of the operating systems that they run on. Thus they can be used on large number of web servers and also in cross platform business applications. [23]

- **Language Independence** – Programming of CGI can be done using any language such as Perl, C++, Python etc. [23]

- **Open Standard** – CGI is an open standard technology. [22]

However, CGI has drawbacks that discourage its use. They are as follows

- **Limited Functionality** – Functionality of CGI are limited. It does not support cookies and sessions. [22]

- **Processing Overhead** – CGI has processing overhead. Each time an instance is invoked the web server creates a separate process for the execution of CGI script. [22] This model is not suitable when large number of invocations exist.
3.2.2.5 ASP.NET

ASP.NET is the latest version of web application from Microsoft. ASP.NET is built on the DOT.NET framework. As of today ASP.NET 3.5 is the latest version. ASP.NET removes classic ASP’s drawbacks. The drawbacks are that it is a typeless language and hence does not support the robust object oriented programming features. Lack of the Object oriented feature also forces reuse and repeat of code within the project. [24] Another disadvantage is that there is a mixture of two types of code i.e. the ASP code and the HTML code. Separating the business and presentation layers would result in structured format.

Some of the features offered by ASP.NET are as follows.

- **Code Separation** – ASP.NET allows separation of code into a .aspx file that contained the HTML and ASP code and a .vb or .cs file that contained the logic behind. .vb is for code written in visual basic and .cs is for code written in C#.[25]

- **Application Folders** – ASP.NET defines separate application folders that can be used during the development of a web application. Some of the folders are App_Data, App_Code, App_GlobalResource etc.[25]

- **Extensive Web Controls** – The DOT.NET framework provides ASP.NET with an extensive set of web controls that can be used to develop interactive. Some of the web controls available are Navigation Controls, Data Controls and Security Controls etc.[25]

- **Purely Object Oriented** – ASP.NET is purely object oriented and hence it is strongly typed, more structured and efficient. It also provides a Web.conf file which eases configuration.[25]

- **Themes** – ASP.NET also provides themes that can be used to alter the appearance of the web application. [25]
3.2.2.6 Java Servlets

The drawbacks of CGI (discussed in section 2.2.2.4) are nullified in Java Servlets. Servlets are Java server extensions that are used to make web applications more interactive and dynamic. They are also activated using the URL and by using HTTP protocol. Since it is based on Java, it is confined only to the JVM execution environment and thus is much safer than CGI. Also since Java is used, it can work on any operating system as long as the JVM is installed. [26]

The features of Java Servlets are as follows

- **Portable** – Java Servlets are portable. This means that a servlet can be developed on a windows platform and then easily deployed on another platform that has the JVM installed. [26]

- **Power** – Since the development is in Java, its wide range of APIs can be used by Java servlets. The API’s available would be JDBC for database connections, JNDI for naming directory lookups, EJBs for enterprise java beans etc. [26]

- **Safety** - Java’s garbage collection ability makes java safe from issues such as dangling pointers. It can also protect itself from exceptions by using the exception handling available from Java. Another safety aspect is access control i.e. the extent of access to the web site user can be controlled in java using Java Security Manager. [26]

- **Flexible** – Java servlets can create different types of content. This can be done using objects i.e. by treating the HTML page that is to be generated as an object. Servlets can also use XML-HTML transformation. Some outputs could be through simple ResponseStream objects. [26]
3.2.2.7 XHTML

XHTML is a standard developed by W3C as an enhancement to HTML 4. It follows the XML standard and thus enforces development of “neat” web pages. Advantages offered by XHTML are as follows: [27]

- XHTML offers the ability to add new extensions through XHTML modules and techniques.
- Since XHTML is XML based, a web application developed in this presentation technology can be accessed by various devices.
- XHTML also allows creation of “neat” web pages. This means that the web pages will be driven by certain rules such as well nested tags, proper quotations, attributes in lower case only etc unlike HTML 4.

3.2.2.8 AJAX

Asynchronous Javascript and XML, also known as AJAX, is not a technology in itself. It is a different approach to accessing resources and updating small bits of a web page without the need to refresh the entire page. This is achieved with the help of the XMLHttpRequest and its property named ‘onReadyStateChange’. This property is set to a function that is triggered once the state has changed. The response obtained by the object can be received using the ‘responseText’ or ‘responseXML’ property. [28]

Using AJAX one can access, XML files, Servlets, CGI scripts, as well as local files. This is an ideal way to create dynamic and efficient web applications while separating presentation logic with the working (business logic).
3.2.3 Mobile Application

3.2.3.1 J2ME

J2ME or Java 2 Mobile Edition is Java’s platform for developing small application for mobile devices. J2ME focuses on all small devices such as PDAs, mobile phones etc. The two major advantages of J2ME are as follows. [29]

- **Portability** - J2ME being built on the Java platform can be easily exported from one device to another without change as long as the JVM is installed.

- **Security** – Since the development environment is Java, any code that is downloaded will be secure as the file access rights would be restricted to the Java execution environment.

J2ME is divided into three major components. They are discusses below.

- **Configuration** – J2ME configuration refers to the arrangement of sub-components that make up the JVM for a particular device. This arrangement could include APIs from J2SE (Java 2 Standard Edition) and other APIs depending on the device capabilities and capacity.

- **Profiles** – A J2ME profile refers to a complete package of APIs required to run a particular application. Profiles are built on configuration and additional APIs.

- **Optional APIs** – Optional APIs offer additional features and functionality. These additional APIs are chosen based on the requirements of the application to be developed.

The combination of Configuration, Profiles and Additional APIs are called ‘stack’.

3.2.3.2 DOT.NET for Mobile Applications

DOT.NET for mobile application is Microsoft’s version of developing mobile applications. It runs on the DOT.NET framework, which depends on the CLR, CLS and CTS for its execution. The DOT.NET framework for mobile edition is a scaled down version of the actual DOT.NET framework. The mobile version may require as little as 2.5 MB space and 1.25 MB running memory to function. [30]

The DOT.NET version for mobile comes with support for various functions such as networking, Input/Output, XML wrapping etc. [30] The advantage of developing an application based on DOT.NET is the ability to program in different languages such as VB.NET, C# or J# etc and incorporate those components in the application. The disadvantage however comes form the fact that mobile applications built using the DOT.NET framework can function only on windows operating systems.
3.3 Analysis of Technologies

After discussing the available technologies that can be used to develop the windows, web and mobile application in section 2.2, this section critically analyses each technology. The suitable technologies for the requirement are then selected.

3.3.1 The Desktop Application

Developing the Windows application using Java provides the following advantage

- Deployment to multiplatform
- Support of extensive APIs
- Object oriented development and hence efficient and “clean” code.

However, the negative aspects of using Java for developing the windows application are given below:

- In order to achieve an improved visual appearance we must use external APIs. The capabilities of these APIs with respect to providing different themes are also limited.
- The staffs’ system at Avalon Office Supplies run on window operating systems, hence the advantage of multiple platform deployment also holds no value.

Hence Java for developing the windows application is not suitable with respect to the requirements and scenario at hand.

C++ with Qt can be used to develop desktop applications. The positives of this means of developing desktop applications are

- The use of Qt framework also enhances the ability to improve appearance.
- C++ also being an object oriented programming language would have code that is efficient and structured.
- Qt also allows support for 2D and 3D graphics that can be used to improve the overall application usability
- C++ with Qt also has the power of availability of libraries and plug in features.

However, C++ has some drawbacks.

- Developing GUI software gets very complicated. The effort and code required to create an interactive GUI application is much more as compared to using WPF or C# windows application.
- Also the lack of garbage collection in C++ makes programming more tedious as one has to manually free memory spaces. Otherwise this could affect the efficiency of the application.
- The use of pointers in C++ can be taxing.

Hence C++ with Qt is not a viable option with respect to the requirements at hand.
Using WPF or Windows Presentation Foundation for developing the windows application has the following advantages.

- Uses the power of DOT.NET framework in the application development.
- Greatly improves the appearance of the application thus improving usability.
- Separates the application logic from the presentation logic.
- Has the power of extensive number of APIs to perform various tasks.

The drawback can be that this technology can be used only on windows platform. Since the target systems for the desktop application run on Windows, this drawback is insignificant.

Hence using Windows Presentation Foundation would be the ideal technology to develop the windows application and Microsoft Visual Studio 2008 express can be used for the development.

### 3.3.2 The Web Interface

PHP is a simple and easy to use technology for developing web applications. PHP has the advantage of having various APIs for providing different functionality. PHP however is loosely typed and hence is not well structured and efficient. It is generally used for rapid web development. Further enhancements to the website could also be tedious. Also PHP combines presentation and business logic in one page, which is not the ideal approach for a web application. Further more, the use of PHP does not suit the proposed system architecture.

CGI or Common Gateway Interface has many drawbacks (as discussed in Section 2.2.2.4) and new technologies like Java Servlet etc have taken its place. Though the use of CGI supports the proposed system architecture, its drawbacks render it inappropriate.

ASP has also been superseded by ASP.NET, which offers many more advantages as compared to ASP. Though ASP still exists in many web applications, it is slowly on the verge of decline. Hence ASP would not be a suitable option.

JSP though having the advantage of being strongly typed and having the power of Java, is unsuitable for the proposed architecture. Hence it cannot be used for the web development.

ASP.NET uses the features of the DOT.NET framework such as multiple language support, common base class library etc. Also it allows easy configuration using the web.conf file. Further, it also separates the presentation layer with the business logic layer by defining .aspx page for the presentation and the .cs or .vb pages which has the business logic written in it. ASP.NET also supports extensive web controls and themes. However, aspx files can be hosted only on Window’s IIS server. The current website is not on an IIS server and hence developing the web application in aspx would increase the cost and effort to set up the application and migrate the existing website to another server.
Using XHTML, AJAX and Java Servlet in combination would have the following advantages.

- Separates the presentation logic from the business logic.
- Provides a “neat” presentation layer as XHTML is based on the XML format.
- Parts of web pages could be refreshed using AJAX without the need to refresh.
- Java Servlets can use the power of Java to create efficient and structured programs.
- The use of XHTML will allow deployment of the new web interface (web page) in the existing sever and can be attached to the existing website.
- This combination of XHTML, AJAX and Java Servlet also meets the requirements with respect to the system architecture.
- Using Java Servlet, the mobile application can also use the servlet architecture to interact with the database.

Thus the combination of XHTML, AJAX and Java Servlet is the ideal combination of technologies to develop the web interface. Servlet development will be done using Eclipse IDE and installed in the Glassfish Server. For the development of the front end i.e XHTML and AJAX, Adobe Dreamweaver CS4 will be used. The web page will be hosted on Apache Web Server.

3.3.3 The Mobile Application

Both J2ME as well as DOT.NET for mobile are technologies that can be used to develop mobile applications with respect to the requirement. However, since J2ME uses the Java platform, which allows it to be installed in any mobile platform including windows platform as long as JVM is installed, J2ME is suitable option. Also, since all members of the staff may not use a phone, which has windows operating system, J2ME is the ideal development environment. The development of the mobile application will be done using the Eclipse IDE with J2ME plug-in.
3.4 Software Development Model

After discussing and analysing the available technologies, this section discusses the software development models that can be used to develop the IT solution in this dissertation. Choosing the development strategy depends on factors such as, type of requirement, time frame, required quality of product etc. Three models are discussed.

3.4.1 The Linear Sequential Model

The Linear Sequential Model of software development involves a step-by-step process of software development. This model is also called the classic life cycle model or the waterfall model. The steps involved in development are analysis, design, coding, testing and support. [31] Figure 2 demonstrates the linear sequential model of development.

![Figure 3: The Linear Sequential Model][31]

3.4.2 The Prototyping Model

The Prototyping Model is used when the requirements of the software cannot be established or furnished in the initial discussions. [31] This means that there may be constant change in requirements or the requirements would be further clarified during development. Figure 3 illustrates this model of software development.

![Figure 4: The Prototyping Model][31]
3.4.3 The Rapid Application Development Model

Rapid Application Development Model or the RAD Model is a model that is used to develop an application in a short period of time. This model can be used if the time period for development is very short, the requirement for the software to be developed is very clear and precise and large number of resources is available. The idea of this model is to package the software into modules that can be plugged into each other and then use teams for parallel development of these modules. This model of development is illustrated in the figure below. [31]

As a part of this dissertation, the Linear Sequential Model will be used as the requirements for IT solution that is being built is clear and been given. Hence the need for using Prototype Model is eradicated. Also since using the RAD model would require large number of resources as development would of different components would be parallel, this model also cannot be used.
3.5 Risk Management

After the technologies available are analysed and the relevant technologies finalised, this section describes the risk analysis done as a part of this dissertation, it also describes risk mitigation alternatives.

3.5.1 Risk Identification

The possible risks that could come up were identified as follows:

1. Hardware malfunction.
2. Non-availability of Internet or LAN.
3. Application being developed interferes with ports that are being used by other applications.
4. Lack of subjects for evaluation.
5. Lack of clear and accurate information.
6. Lack of knowledge among staffs to use and maintain the new IT solution.

3.5.2 Risk Assessment

![Figure 6: Risk Assessment – Impact vs. Probability](image)

Risks on the whole can be categorised as shown above.

The risks that were identified as a part of this project were assessed and put into categories as shown below.

1. Hardware malfunction – This is a low probability risk as system used for development is a new system. The impact of such an event occurring would be high as development would be hindered.
2. Non-availability of Internet or LAN – This risk is of low probability as a reliable ISP is being used. The impact of such an event occurring would be high as development would be hindered.
3. Application being developed interferes with ports that are being used by other applications. – This is a low probability risk as the ports that are being used by
the system can be checked to avoid the same. The impact of this risk is low as
the ports of the application can be changed.
4. Lack of subjects for evaluation. – This is a low probability risk as it is rare that
we cannot find subjects who would like to co-operate. The impact of this is
low with respect to the project development and execution. This risk would
impact the accuracy of the evaluation.
5. Lack of clear and accurate information – This is a high probability risk as
there could be miscommunication between the analyst and staff at the office.
This however has a low impact as the requirement can be re-checked and
confirmed.
6. Lack of knowledge among staffs to use and maintain the new IT solution. –
This is a high probability risk as the IT solution being developed is new and
unrelated to the softwares being used at present. This thus has a high impact
with respect to the execution of the project.

3.5.3 Risk Management

The management and mitigation alternatives of the risks mentioned are given below.

1. Hardware malfunction. – This can be mitigated by doing the development
using a new system and taking regular backups.
2. Non-availability of Internet or LAN – This risk can be mitigated procuring a
VLAN software in advance.
3. Application being developed interferes with ports that are being used by other
applications.- This risk can be mitigated by changing the ports being used.
4. Lack of subjects for evaluation – This risk can be mitigated by arranging for
subjects in advance and ensuring that they co-operate during the evaluation
phase.
5. Lack of clear and accurate information – This risk can be mitigated by re-
checking the information gathered. It can also be mitigated if a prototype is
developed and shown to the clients so that changes can be incorporated easily.
6. Lack of knowledge among staffs to use and maintain the new IT solution. –
This risk can be mitigated by providing a training session to the staff.
3.6 Professional, Legal, Social and Ethical Issues

3.6.1 Professional Issues

The IT solution being developed is of the best quality possible. All standards for development and design are followed. The best solution for the requirements at hand is furnished and all major decisions are taken after discussion with the stake holder (in this case Avalon Office Equipments). Cost minimisation is kept as a criterion and also personal opinion is not given priority.

3.6.2 Legal Issues

Access to the software is controlled using username and password. Hence any unauthorised access can be prevented. If staff leaves the company, his access to the software can be delete by deleting his user details. The data communication over the network will not be encrypted. This is because no critical data such as credit card numbers, bank account numbers etc are used. Also logging will be done in order to track any intrusion. All business data would be in the database and data security would be the responsibility of the database administrator.

3.6.3 Ethical Issues

Human subjects have been used only for simple evaluation of the application. All company related data or information that has been received as a part of this development would be strictly confidential.

3.6.4 Social Issues

The stake holders have been informed that there would be a need to inform their current clients to avoid using phone calls and generic emails and use the given form (which can be downloaded from the website) for enquiries or they can use the online form given. This will help streamline data inflow. Also, the stake holders would have to request their suppliers to reply to the auto-generated mails being sent by the new application.
3.7 Conclusion

This literature review contains the study done in order to implement an IT solution for automating and streamline a business. It initially discusses the architecture for the IT solution that is being developed. Possible architectures were discussed and analysed.

Various technologies were then discussed and analysed to develop the desktop application, the web interface and the mobile interface. The technologies discussed for the desktop application were Microsoft’s DOT.NET, WPF, Java Desktop Application and C++ with Qt framework. It was decided that WPF would be used to develop the desktop application with C# as the coding language.

For the web interface, ASP, JSP, PHP, ASP.NET, CGI scripting, AJAX, XHTML and Java Servlet were discussed. It was decided that a combination of XHTML, AJAX and Java Servlet would be used to develop the web interface as this would suite the architecture proposed as well as provide a good platform for future enhancements.

For the web interface, J2ME and DOT.NET for mobile was discussed. After the analysis, the conclusion drawn was to use J2ME since this technology allowed cross-platform deployment.

After the technology and the architecture was finalised, the model for development was finalised and the risks studied.

Thus the current business process has been studied and the architecture along with the technologies that will be used have been finalised. The following section discusses the design of each component of the IT solution in detail.
CHAPTER 4: SYSTEM DESIGN

4.1 Introduction

In the previous chapter, the current business process was described. The proposed IT solution architecture was furnished and various technologies that could be used were discussed and analysed. In this chapter, the design of individual systems will be discussed. Namely, the database design, the desktop application design, the mobile application (J2ME Midlets) and the Servlet classes. The design would start by furnishing the use cases. After the use cases were finalised, the CRC diagrams are drawn. This would eventually lead to the creation of the class diagrams. After the design of the desktop, mobile and Servlet components, the database design for the same was created.

4.2 Desktop Application Design

In this section, the design of the desktop application design is created. The design will initialise with the creation of the use case diagram of the system and the actor (the user). The use case diagram is furnished below.

4.2.1 Use Case Diagram

![Use Case Diagram – Desktop application](image)

Figure 7: Use Case Diagram – Desktop application
4.2.2 Textual Description

Following the creation of the use cases, this section furnishes the textual description of the use cases. The textual descriptions consists of the goal of the use case, the preconditions, the success criteria, any exceptions that could occur as well as the post conditions.

**USE CASE**: Add Supplier
**GOAL**: To add a supplier record to the database

**Pre-Conditions**
The user must be logged in

**Main Success Scenario**
1. Supplier record is added to the database

**Exceptions**
1. The record was not added as the data validation failed at the client side
2. The record was not added due to connectivity issues.
3. The database is not running.

**Post Conditions**
Data of the new supplier can now be used by the system.

**Sub Use Cases**
None

**USE CASE**: Update Supplier
**GOAL**: To update a supplier record to the database

**Pre-Conditions**
The user must be logged in

**Main Success Scenario**
1. Supplier record is updated in the database

**Exceptions**
1. The record was not updated as the data validation failed at the client side
2. The record was not updated due to connectivity issues.
3. The database is not running.

**Post Conditions**
Data of the updated supplier can now be used by the system.

**Sub Use Cases**
None

**USE CASE**: Delete Supplier
**GOAL**: To delete a supplier record from the database

**Pre-Conditions**
The user must be logged in as a superuser

**Main Success Scenario**
1. Supplier record is deleted from the database

**Exceptions**
1. The record was not deleted as the data validation failed at the client side
2. The record was not deleted due to connectivity issues.
3. The database is not running.

**Post Conditions**
The deleted record does not exist in the database.

**Sub Use Cases**
None
USE CASE: Add Product
GOAL: To add a Product record to the database
Pre-Conditions
The user must be logged in
Main Success Scenario
1. Product record is added to the database
Exceptions
   1. The record was not added as the data validation failed at the client side
   2. The record was not added due to connectivity issues.
   3. The database is not running.
Post Conditions
Data of the new Product can now be used by the system.
Sub Use Cases
None

USE CASE: Update Product
GOAL: To update a Product record to the database
Pre-Conditions
The user must be logged in
Main Success Scenario
1. Product record is updated in the database
Exceptions
   1. The record was not updated as the data validation failed at the client side
   2. The record was not updated due to connectivity issues.
   3. The database is not running.
Post Conditions
Data of the updated Product can now be used by the system.
Sub Use Cases
None

USE CASE: Delete Product
GOAL: To delete a Product record from the database
Pre-Conditions
The user must be logged in as a superuser
Main Success Scenario
1. Product record is deleted from the database
Exceptions
   1. The record was not deleted as the data validation failed at the client side
   2. The record was not deleted due to connectivity issues.
   3. The database is not running.
Post Conditions
The deleted record does not exist in the database.
Sub Use Cases
None
USE CASE: Add Customer
GOAL: To add a Customer record to the database
Pre-Conditions
The user must be logged in
Main Success Scenario
1. Customer record is added to the database
Exceptions
   1. The record was not added as the data validation failed at the client side
   2. The record was not added due to connectivity issues.
   3. The database is not running.
Post Conditions
Data of the new Customer can now be used by the system.
Sub Use Cases
None

USE CASE: Update Customer
GOAL: To update a Customer record to the database
Pre-Conditions
The user must be logged in
Main Success Scenario
1. Customer record is updated in the database
Exceptions
   1. The record was not updated as the data validation failed at the client side
   2. The record was not updated due to connectivity issues.
   3. The database is not running.
Post Conditions
Data of the updated Customer can now be used by the system.
Sub Use Cases
None

USE CASE: Delete Customer
GOAL: To delete a Customer record from the database
Pre-Conditions
The user must be logged in as a superuser
Main Success Scenario
1. Customer record is deleted from the database
Exceptions
   1. The record was not deleted as the data validation failed at the client side
   2. The record was not deleted due to connectivity issues.
   3. The database is not running.
Post Conditions
The deleted record does not exist in the database.
Sub Use Cases
None
USE CASE: Generate Reports
GOAL: To generate reports
Pre-Conditions
The user must be logged in as superuser
Main Success Scenario
1. The required report is generated.
Exceptions
   1. The report was not generated as data does not exist
Post Conditions
Report is generated and stored in the location provided.
Sub Use Cases
None

USE CASE: Alter Cost Analysis
GOAL: To alter existing cost analysis.
Pre-Conditions
The cost analysis that is to be altered must exist.
Main Success Scenario
1. The cost analysis is altered and a new cost analysis document is created.
Exceptions
   1. The cost analysis was not created due to internal system errors.
Post Conditions
A new cost analysis document is created.
Sub Use Cases
None

USE CASE: Create Quotation
GOAL: To create a quotation from an existing cost analysis.
Pre-Conditions
The cost analysis for which the quotation is created, must exist.
Main Success Scenario
1. The quotation is created from the cost analysis.
Exceptions
   1. The quotation was not created due to database connectivity issues.
Post Conditions
A quotation was created for a cost analysis.
Sub Use Cases
None

USE CASE: Create Invoice
GOAL: To create an invoice from an existing quotation.
Pre-Conditions
The quotation for which the invoice is created, must exist.
Main Success Scenario
1. The invoice is created from the quotation.
Exceptions
   1. The invoice was not created due to database connectivity issues.
Post Conditions
An invoice was created for a quotation.
Sub Use Cases
None
USE CASE: Send Quotation
GOAL: To send a quotation to a customer
Pre-Conditions
The quotation must exist.
Main Success Scenario
1. The quotation was sent to the customer.
Exceptions
1. The quotation could not be sent due to internal system issues.
Post Conditions
The quotation was sent to the customer
Sub Use Cases
None

USE CASE: Send Invoice
GOAL: To send an invoice to a customer
Pre-Conditions
The invoice must exist.
Main Success Scenario
1. The invoice was sent to the customer.
Exceptions
1. The invoice could not be sent due to internal system issues.
Post Conditions
The invoice was sent to the customer
Sub Use Cases
None

USE CASE: Start and Stop Mail Processor
GOAL: To start or stop mail processor function
Pre-Conditions
The user must be logged in as a super user.
Main Success Scenario
1. The mail processor application is started if it was stopped.
   2. The mail processor application is stopped if it was started.
Exceptions
1. The application could not be started due to internal technical issues.
Post Conditions
The mail processor application was started or stopped if it was stopped or started respectively.
Sub Use Cases
None
4.2.3 Class Responsibility and Collaborator diagram (CRC diagram)

After the use cases have been identified and described, this section creates the appropriate high level classes with respect to the use cases. The classes identified with their responsibilities and their collaborators are as follows.

<table>
<thead>
<tr>
<th>GUI class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
</tr>
<tr>
<td>Display interface</td>
</tr>
<tr>
<td>Start and Stop automatic mail processor</td>
</tr>
<tr>
<td>create mails for sending</td>
</tr>
<tr>
<td>Add, Update, Delete Customer</td>
</tr>
<tr>
<td>Add, Update, Delete Products</td>
</tr>
<tr>
<td>Add, Update, Delete Suppliers</td>
</tr>
<tr>
<td>Generate and Display reports</td>
</tr>
<tr>
<td>After Cost Analysis</td>
</tr>
<tr>
<td>Create Invoice</td>
</tr>
<tr>
<td>Create Quotation</td>
</tr>
<tr>
<td>Login Functionalities</td>
</tr>
<tr>
<td>Log activities</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MailManager Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
</tr>
<tr>
<td>Read eMails from inbox</td>
</tr>
<tr>
<td>Process Attachments</td>
</tr>
<tr>
<td>Create eMails for sending with appropriate attachments (quotation/invoice) and body content</td>
</tr>
<tr>
<td>Create CostAnalysis</td>
</tr>
<tr>
<td>Log events</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CostAnalysis Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
</tr>
<tr>
<td>Create cost analysis and save to file</td>
</tr>
<tr>
<td>After Cost Analysis</td>
</tr>
<tr>
<td>Store data to database</td>
</tr>
<tr>
<td>Log events</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quotation Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
</tr>
<tr>
<td>Create quotation and save to file</td>
</tr>
<tr>
<td>Store data to database</td>
</tr>
<tr>
<td>Log events</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Invoice Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
</tr>
<tr>
<td>Create Invoice and save to file</td>
</tr>
<tr>
<td>Store data to database</td>
</tr>
<tr>
<td>Log events</td>
</tr>
</tbody>
</table>
The classes shown above are the high level classes. More classes were created as loose coupling and tight cohesion concepts would be used. Design pattern such as factory pattern, strategy pattern, observer pattern etc. were also be used. Hence interfaces as well as Factory classes were used around the primary classes.
4.2.4 Class Diagrams

The previous section describes the high level classes that were used in the development of the desktop application. This section highlights the association of these classes. The first part of this section shows the high level overall class diagram. The second part shows the detailed class diagram.

4.2.4.1 High Level Class Diagram

Figure 9: High Level Class Diagram – Desktop application
4.2.4.2 Detailed Class Diagram

Mail Manager Class

![Mail Manager Class Diagram]

Logger Class

![Logger Class Diagram]

Figure 10: Mail Manager Class Diagram

Figure 11: Logger Class Diagram
Figure 12: Database Manager Class Diagram
Product, Customer, Supplier Class

![PCS Class Diagram]

**Figure 13: PCS Class Diagram**
Figure 14: Reports Manager Class Diagram
**CostAnalysis Class**

```
<<Interface>>
ICostAnalysis

private String strFilePath
public void createCostAnalysis(arrayList dataValues)
public arrayList getCostAnalysisInfo()
public void alterCostAnalysis(arrayList dataValues)
```

```
Class CostAnalysisFactory
public ICostAnalysis getCostAnalyser()
```

![Figure 15: Cost Analysis Class Diagram](image)

**Quotation Class**

```
<<Interface>>
IQuotation

private String strFilePath
public void createQuotation()
```

```
Class QuotationFactory
public IQuotation getQuotationCreator()
```

```
QuotationImp
private string strFilePath
public void createQuotation()
```

![Figure 16: Quotation Class Diagram](image)
Invoice Class

Thus the classes for the development of the desktop application are created and the association between them have been indicated above. The next section, details the design of the mobile application, which is the second part being developed as a part of the IT solution for Avalon Office Equipments.
4.3 Mobile Application Design

The mobile application plays a critical role in enhancing accessibility to business data when away from office. This application allows the user to access business data when at a location such as a client office, in a common meeting. The availability of critical business data allows important decisions to be taken or better discussions to be held. A mobile application is also crucial as the internet connectivity is more easily available in mobile devices than in laptops when away from office. The functionalities available in the mobile application are described in the use cases below.

4.3.1 Use Case Diagram

![Use Case Diagram](image)

Figure 18: Use Case Diagram – Mobile Application

4.3.2 Textual Description

The textual description of the use cases shown above are described below.

**USE CASE:** Search Quotation

**GOAL:** To search for a particular quotation.

**Pre-Conditions**
The user must have internet connection.

**Main Success Scenario**
1. List of quotations are displayed.

**Exceptions**
1. The list of quotation could not be displayed as no data exists.
2. The list of quotation could not be displayed due to connectivity issues.
3. The list of quotation could not be displayed due to database issues.

**Post Conditions**
List of quotation is visible on the mobile screen

**Sub Use Cases**
None
USE CASE: Search Invoice
GOAL: To search for a particular invoice.
Pre-Conditions
The user must have internet connection.
Main Success Scenario
1. List of invoices are displayed.
Exceptions
   1. The list of invoice could not be displayed as no data exists.
   2. The list of invoice could not be displayed due to connectivity issues.
   3. The list of invoice could not be displayed due to database issues.
Post Conditions
List of invoice is visible on the mobile screen
Sub Use Cases
None

USE CASE: Get Quotation Details
GOAL: To retrieve the details of a particular quotation
Pre-Conditions
The user must have internet connection.
Main Success Scenario
1. Details such as date, product name, quantity of each product and price are displayed.
Exceptions
   1. The details of quotation could not be displayed as no data exists.
   2. The details of quotation could not be displayed due to connectivity issues.
   3. The details of quotation could not be displayed due to database issues.
Post Conditions
List of quotation details is visible on the mobile screen
Sub Use Cases
None

USE CASE: Get Invoice Details
GOAL: To retrieve the details of a particular invoice
Pre-Conditions
The user must have internet connection.
Main Success Scenario
1. Details such as date, product name, quantity of each product and price are displayed.
Exceptions
   1. The details of invoice could not be displayed as no data exists.
   2. The details of invoice could not be displayed due to connectivity issues.
   3. The details of invoice could not be displayed due to database issues.
Post Conditions
List of invoice details is visible on the mobile screen
Sub Use Cases
None
USE CASE: Browse Product Details
GOAL: To retrieve the details of a particular product

Pre-Conditions
The user must have internet connection.

Main Success Scenario
1. Details such as selling price, cost price, name and brand are displayed.

Exceptions
1. The details of product could not be displayed as no data exists.
2. The details of product could not be displayed due to connectivity issues.
3. The details of product could not be displayed due to database issues.

Post Conditions
Product details are visible on the mobile screen

Sub Use Cases
None

USE CASE: Get Suppliers for Particular products.
GOAL: To retrieve the details of Suppliers for Particular products.

Pre-Conditions
The user must have internet connection.

Main Success Scenario
1. List of Details such as supplier Name, Company, Phone Number and Extension is displayed.

Exceptions
1. The details of suppliers could not be displayed as no data exists.
2. The details of suppliers could not be displayed due to connectivity issues.
3. The details of suppliers could not be displayed due to database issues.

Post Conditions
List of details of suppliers is visible on the mobile screen

Sub Use Cases
None
### 4.3.3 Class Responsibility Collaborator Diagram (CRC Diagrams)

This section highlights the high level classes that will be used by the mobile application to communicate with the servlet and the database (through the servlet) in order to retrieve the required data.

<table>
<thead>
<tr>
<th>Display Class</th>
<th>Collaborators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Displays the welcome screen and company logo.</td>
<td>MenuForm</td>
</tr>
<tr>
<td>Creates command to enable users to navigate to the application menu.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MenuForm Class</th>
<th>Collaborators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Displays menu of the applications</td>
<td>Search Invoice, SearchQuotation, SearchProducts</td>
</tr>
<tr>
<td>Initiates appropriate form for search depending on use input</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SearchInvoice Class</th>
<th>Collaborators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Creates the form for searching invoices and displays it</td>
<td>HTTPServletTrigger</td>
</tr>
<tr>
<td>Initiates the servlet call with the appropriate parameters.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SearchProducts Class</th>
<th>Collaborators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Creates the form for searching products and displays it</td>
<td>HTTPServletTrigger</td>
</tr>
<tr>
<td>Initiates the servlet call with the appropriate parameters.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SearchQuotation Class</th>
<th>Collaborators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Creates the form for searching quotations and displays it</td>
<td>HTTPServletTrigger</td>
</tr>
<tr>
<td>Initiates the servlet call with the appropriate parameters.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ResultOptions Class</th>
<th>Collaborators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Displays the list of options that resulted from the first search of Invoice, Products, Quotation.</td>
<td>HTTPServletTrigger</td>
</tr>
<tr>
<td>Triggers the servlet to get details of options displayed in the list.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HTTPServletRequest Class</th>
<th>Collaborators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Creates appropriate query string</td>
<td></td>
</tr>
<tr>
<td>Triggers the HTTP servlet and collects response.</td>
<td>ResultOptions, FinalResultOptions</td>
</tr>
<tr>
<td>Calls appropriate class to display servlet call results</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ResultOptions Class</th>
<th>Collaborators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Displays the list of options that resulted from the 'get details' search of Invoice, Products, Quotation.</td>
<td></td>
</tr>
</tbody>
</table>

---

**Figure 19: CRC Diagram – Mobile Application**
4.3.4 Class Diagram

The detailed class diagram for the mobile application is shown below.

Figure 20: Class Diagram – Mobile Application
4.4 Servlet Classes Design

4.4.1 Use Case Diagram

As per the architecture design (described in Section 2.1), the servlet is the interface between the mobile and web application and the database. The following use case diagram describes the functionality of the servlet.

![Use Case Diagram – Servlet](image)

4.4.2 Use Case Diagram

The textual description of the use cases shown above are described below.

**USE CASE:** Get Supplier for particular product.

**GOAL:** To Get Supplier for particular product.

**Pre-Conditions**
The servlet must be loaded and the server must be running.

**Main Success Scenario**
1. Supplier details for a particular product are displayed.

**Exceptions**
1. The list of supplier details could not be displayed as no data exists.
2. The list of supplier details could not be displayed due to connectivity issues.
3. The list of supplier details could not be displayed due to database issues.

**Post Conditions**
List of supplier details is visible on the mobile screen

**Sub Use Cases**
None
USE CASE: Get List of Products  
GOAL: To Get List of Products  

Pre-Conditions  
The servlet must be loaded and the server must be running.  

Main Success Scenario  
1. List of Products are displayed.  

Exceptions  
1. The list of products could not be displayed as no data exists.  
2. The list of products could not be displayed due to connectivity issues.  
3. The list of products could not be displayed due to database issues.  

Post Conditions  
List of products is visible on the mobile screen  

Sub Use Cases  
None  

GOAL: To Get List of Quotation  
Pre-Conditions  
The servlet must be loaded and the server must be running.  

Main Success Scenario  
1. List of Quotation are displayed.  

Exceptions  
1. The list of Quotation could not be displayed as no data exists.  
2. The list of Quotation could not be displayed due to connectivity issues.  
3. The list of Quotation could not be displayed due to database issues.  

Post Conditions  
List of Quotation is visible on the mobile screen  

Sub Use Cases  
None  

GOAL: To Get List of Invoice  
Pre-Conditions  
The servlet must be loaded and the server must be running.  

Main Success Scenario  
1. List of Invoice is displayed.  

Exceptions  
1. The list of Invoice could not be displayed as no data exists.  
2. The list of Invoice could not be displayed due to connectivity issues.  
3. The list of Invoice could not be displayed due to database issues.  

Post Conditions  
List of Invoice is visible on the mobile screen  

Sub Use Cases  
None
USE CASE: Get Quotation details
GOAL: To Get Quotation details.
Pre-Conditions
The servlet must be loaded and the server must be running.
Main Success Scenario
1. Details of Quotation are displayed.
Exceptions
   1. The Details of Quotation could not be displayed as no data exists.
   2. The Details of Quotation could not be displayed due to connectivity issues.
   3. The Details of Quotation could not be displayed due to database issues.
Post Conditions
The Details of Quotation is visible on the mobile screen
Sub Use Cases
None

USE CASE: Get Invoice details
GOAL: To Get Invoice details.
Pre-Conditions
The servlet must be loaded and the server must be running.
Main Success Scenario
1. Details of Invoice are displayed.
Exceptions
   4. The Details of Invoice could not be displayed as no data exists.
   5. The Details of Invoice could not be displayed due to connectivity issues.
   6. The Details of Invoice could not be displayed due to database issues.
Post Conditions
The Details of invoice is visible on the mobile screen
Sub Use Cases
None

USE CASE: Capture webform data and send as attachment.
GOAL: To Capture webform data and send as attachment.
Pre-Conditions
The servlet must be loaded and the server must be running.
Main Success Scenario
1. The web form data is captured and an email is sent to the company id with the form data in an XML file.
Exceptions
   4. The form data could not be captured due to servlet connection errors.
   5. The XML file could not be created due to data exceptions.
   6. The email could not be sent due to mail server connection errors.
Post Conditions
The web page displays success message and email is sent with an XML file attachment containing the data
Sub Use Cases
None
4.4.3 CRC Diagram

After discussing the use cases, this section describes the servlet classes that will be used to develop the interface between the mobile application, web application and the database.

<table>
<thead>
<tr>
<th>Display Class</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Displays the welcome screen and company logo. Creates command to enable users to navigate to the application menu.</td>
<td>MenuForm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MenuForm Class</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Displays menu of the applications. Calls appropriate class depending</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quotation Class</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Creates appropriate queries to handle the requests to get details of Quotations. Establishes connection with the database</td>
<td>DatabaseConnection</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DatabaseConnection Class</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Connects to the database and executes the query passed and returns the result as String.</td>
<td></td>
</tr>
<tr>
<td><strong>MailManager Class</strong></td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Create and send email with XML file attachment</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>SMTPAuthenticator Class</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Creates a new authenticator type to and encapsulates PasswordAuthenticator class.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>XMLCreator Class</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Creates XML file that is to be used as attachment and returns it as a String.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>XMLAttachmentDataSource Class</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Creates a new DataSource type. This type creates the ByteArrayInputStream from a String.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>WebEnquiryProcessing Class</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Responsibilities</td>
<td>Collaborators</td>
</tr>
<tr>
<td>Creates the XML file with the form data</td>
<td>XMLCreator</td>
</tr>
<tr>
<td>Creates email with XML file as attachment and sends it</td>
<td>MailManager</td>
</tr>
</tbody>
</table>

**Figure 22: CRC Diagram – Servlet**
4.4.4 Class Diagram

The details of the classes and the interconnectivity of the classes are shown below in the class diagram.

Figure 23: Class Diagram – Servlet
Figure 24: Class Diagram 2 – Servlet
4.5 Database Design

The previous sections describe in detail the design of the desktop application, the mobile application and the servlet component. In this section the design of the database is discussed. It is initialised by analysing the entities involved and the relations between them. The Entity-Relation diagram or the ER diagram is used to create a conceptual database design. “The entity-relationship (ER) data model allows us to describe the data involved in a real world enterprise in terms of objects and their relationships and is widely used to develop and initial database design.” [34] After the conceptual design is created, the ER diagram is then converted into a logical design which would have the database schemas.
4.5.1 Conceptual Database Design

Figure 25: ER Diagram – Database design
4.5.2 Logical Database Design

**relation** Quotation
Quotation No: integer
Enquiry No: integer
Date: date
Customer Name: Names
PRIMARY KEY Quotation No

**relation** Invoice
Invoice Number: integer
Quotation Number: integer
Date: date
Total Sales: amount
Customer Name: names
Customer Company: names
PRIMARY KEY Invoice Number

**relation** Customer
Customer Id: integer
Name: names
Title: string
Company: names
Address: string
e-mail Id: string
Fax: integer
Telephone: integer
Extension: integer
PRIMARY KEY Customer Id

**relation** Products_Supplied
Supplier Id: integer
Product Id: integer
Date: date
Quantity: integer
Quotation Number: integer
PRIMARY KEY (Supplier Id, Product Id, Quotation Number)
FOREIGN KEY Products Id
REFERENCES Products (Products Id)
FOREIGN KEY Supplier Id
REFERENCES Suppliers (Supplier Id)
FOREIGN KEY Quotation Number
REFERENCES Quotation (Quotation Number)

**relation** Enquiries
Enquiry Number: integer
Customer Name: names
Attention: names
Date: date
Product Id: integer
Quantity: integer
Selling Price: amount
Cost Price: amount
Profit: amount
Profit Margin: amount
PRIMARY KEY Enquiry Number

**relation** Products_Enquired
Enquiry Number: integer
Products Id: integer
Date: date
PRIMARY KEY (Enquiry Number, Products Id)
FOREIGN KEY Enquiry Number
REFERENCES Enquiries(Enquiry Number)
FOREIGN KEY Products Id
REFERENCES Products (Products Id)

**relation** Suppliers
Supplier Id: integer
Name: names
Company: names
Address: string
e-mail Id: string
Fax: integer
Telephone: integer
Extension: integer
Notes: string
PRIMARY KEY Supplier Id

**relation** Products
Product Id: integer
Product Name: names
Brand: names
Description: string
Selling Price: amount
Cost price: amount
Notes: string
PRIMARY KEY Product Id
### 4.6 Conclusion

This chapter discussed in detail the design of the main components of the IT solution being developed as a part of this dissertation. Among the components discussed were the database structure, the desktop application, the mobile application and the servlets which will be the interface between the database and the mobile and web application.

As a part of the design, each component’s functionality was initially discussed using the use case diagrams. The textual descriptions of the same were furnished. After an understanding of the use cases, the classes were formulated using the CRC diagrams. This then facilitated the development of the class diagrams which highlighted the association between the classes.

Thus an overall design of the system is created. The next chapter details the implementation challenges of system.
CHAPTER 5: IMPLEMENTATION

5.1 Implementation Challenges

After the system design was concrete, the development of the IT business solution for Avalon Office Supplies was initialised. During the build, many challenges were faced. This chapter details some of the challenges faced and also describes the solution for the same.

1. MULTIPLE SUPPLIERS FOR A PRODUCT

The Challenge
One of the challenges that were faced was handling multiple suppliers. After discussing with the company staff, it was learnt that it is possible for 2 suppliers to supply the same product in a quotation. This happens when the supplier with the lowest supplier price does not have enough stock to meet the demand. The desktop application is programmed to automatically create the cost analysis by picking the supplier with the lowest supplier price. Hence it was required that there must be a way to handle the case where multiple suppliers supply products.

The Solution
The solution for this was to give an option in the GUI (Alter Cost Analysis button) to add more than one supplier for a product. On clicking the ‘Generate’ button a new cost analysis would be created which had multiple entries for one product with the corresponding supplier price and calculations of Profit Margin, Net Profit etc. Thus in the cost analysis, for one product there would be multiple suppliers with two different multiple price (cost price) and corresponding values of profit margin, profit value etc. However, the application was designed in such a way that the quotation created would have only one entry for the product that has two suppliers.

2. CONVERTING THE SERVLET REPLY TO MOBILE LIST IN MOBILE SCREEN

The Challenge
This challenge was to convert the reply stream from the Servlet to list of options on the mobile screen. This would be required when the user searches for a product, invoice, quotation etc using the mobile application and the request is sent to the Servlet. The Servlet then replies with the result of the search. The user then has an option to choose from the result of the search and get more details on one of the results.

The Solution
This challenge was overcome by allowing the Servlet to fix a delimiter between each option. The stream was then read within the mobile application, converted into string, and passed to appropriate classes depending of if the result was the initial result options or the final result options. Within the classes, the string was split at the delimiter and then appended to the display List class.
3. MOBILE APPLICATION DEVELOPMENT LIMITATIONS

The Challenge
Some of the challenges during the development of the mobile applications were limited memory, limited class types such as Arraylists, Treemaps etc.

The Solution
The solution for this was to design the application using minimum variables, and efficient patters. Instead of arraylists, the Vector was used.

4. DYNAMICALLY CREATING FORM ELEMENTS

The Challenge
Dynamically creating web form components was a challenge. In the new enquiry for developed for Avalon Office Equipments, one section was to select the products for which the enquiry is to be made. Based on the value selected in the ‘Item’ dropdown, the ‘Brand’ dropdown is populated and based on the ‘Brand’ and the ‘Item’ dropdown, the ‘Serial number’ dropdown is populated. Also dropdown rows can be added dynamically. Capturing and processing the right data was a challenge. This was because the element name had to be created dynamically in order to allow capturing the value.

The Solution
The solution for this was to use a variable with count pointer. This would point to the appropriate elements and based on the count values, the element name was created. Thus each dropdown ‘value change’ event was captured and appropriate brand dropdown or serial number dropdown values were changed.

5. WRITING TO EXCEL SHEET (JAVA)

The Challenge
The data captured in the web form had to be written to an excel sheet. The written Excel sheet had to be in the same format as the Excel sheet that would be filled manually for enquiry. This excel sheet would be sent to the email inbox that is monitored by the desktop application to allow automatic processing. However, writing to excel sheets was not successful using the jdbc:odbc drivers. It was observed that the performance was unreliable. This means that at times even though the process completed without errors, the data was not written. There were other available APIs but that would make the process complicated.

The Solution
The solution was to write the form data to an XML file and then send the same as an attachment to the email inbox. A new component was built and added to the desktop application to allow parsing and processing the data in the XML file. This component will also be useful if in future if the enquiry forms would change from Microsoft Excel to Microsoft Infopath forms. Microsoft Infopath forms saves data in the XML format.
Microsoft Excel was chosen over Microsoft Infopath as the technology to send enquiry forms as attachments by users as MS Infopath was only packaged with the premium edition of Microsoft Office package. Thus not every customer would have it.

6. XML FILE ATTACHMENT WITHOUT PHYSICALLY CREATING FILE

The Challenge
One of the most interesting challenges was to create an XML file attachment from the data entered in the web form and send the same without physically creating the attachment file. This means that the attachment was created from the data stream. The creation of physical file was discouraged because this action would be done at the servlet container and initiated by the website. It is possible that at the same time a web enquiry form would be sent by two different users. This would result in the creation of a file with the same name in one folder on the servlet container which would cause an error. Using GUID to create files would increase complexity of the system.

The Solution
This challenge was met by creating a class which was named XMLAttachmentDataSource. This class implemented the DataSource Interface and it was designed to accept the XML string as parameter to the constructor. This XML string was then internally converted into the required byte array stream. This was then passed to the datahandler of the mail message. Thus zero memory footprint and no physical file was created. The code is shown below.

```java
public class XMLAttachmentDataSource implements DataSource {

    byte [] bytStringBytes;

    public XMLAttachmentDataSource(String XMLString){
        bytStringBytes = XMLString.getBytes();
    }

    @Override
    public String getContentType() {
        return "application/xml";
    }

    @Override
    public InputStream getInputStream() throws IOException {
        return new ByteArrayInputStream(bytStringBytes);
    }

    @Override
    public String getName() {
        return "XMLAttachmentDataSource";
    }

    @Override
    public OutputStream getOutputStream() throws IOException {
        return null;
    }
}
```

Heriot – Watt University
MailManager

```csharp
XMLAttachmentDataSource XMLDStr = new XMLAttachmentDataSource
    (strXMLAttachmentString);
messageBodyPart.setDataHandler(new DataHandler(XMLDStr));
messageBodyPart.setFileName("Enquiry.xml");
multipart.addBodyPart(messageBodyPart);
```

7. A NEW LANGUAGE - XAML

XAML (Extensible Application Markup Language) is a language that is developed by
Microsoft in order to use with Windows Presentation Foundation. WPF was used in
the development of the desktop application of this dissertation. The advantage of
WPF is that it separates the presentation logic with the implementation logic. The
challenge was to master the new language and use the same in order to build the
desktop application.

8. APPROPRIATE INTERFACE FOR REPORT GENERATION

In the Report Generation module, reports would need different number of inputs from
the user. For example, the profit loss report requires just the ‘start year’ value where
as the product profit loss report would need the product name as well as the start year.

The Challenge
The challenge was to design an appropriate common interface function to enable
accepting different values for different reports. The options available were to create
static methods for each report type. The issue with this approach was that the
ReportGeneratorFactory was returning an IReport type and hence the GUI would not
be able to access the static methods. Another approach was to create different
methods within the IReport interface with different parameters. This approach was
rejected as it would not allow expansion i.e. if in future more reports are added, then
the interface will have to change thus resulting in changes in all the classes that
implement the IReport interface.

The Solution
The solution was to pass an ArrayList as the parameter and handle the order of
parameters within the class that implements the IReport interface. In this way any
number of parameters can be passed and allows future expansion.

```csharp
namespace ReportGenerator
{
    public interface IReport
    {
        void generateReport(System.Collections.ArrayList
            alParameterCollection);
        void export(System.Collections.ArrayList
            alParameterCollection);
    }
}
```
9. GUI DISPLAY

The Challenge
In the GUI, the user is allowed to click any button (like Add Customer, Delete Product, Generate Report etc) in any order. On clicking the buttons, the appropriate textboxes, buttons and labels must appear and the ones that existed on the GUI has to disappear. It is possible to control the visibility of each textbox, button or label using the visibility property. But this approach would increase the lines of code. The challenge was to perform this with minimum code.

The Solution
The solution was to group the textboxes, labels and buttons for each button in a group box and then control the visibility of the groupbox. The function used for the same is shown below.

```csharp
private static GroupBox gbCurrentlyVisible = new GroupBox();

private void setTrayElements(GroupBox gbSetElement)
{
    lblStart.Visibility = Visibility.Hidden;
    gbCurrentlyVisible.Visibility = Visibility.Hidden;
    gbSetElement.Visibility = Visibility.Visible;
    gbCurrentlyVisible = gbSetElement;
}
```

The above function is called on a particular button click as follows.

```csharp
private void btnAppMenuSearchCust_Click(object sender, RoutedEventArgs e)
{
    setTrayElements(gbSearchCustomer);
}
```

10. “THE CALLING THREAD MUST BE STA (Single Thread Apartment) BECAUSE MANY UI (User Interface) COMPONENTS USE THIS”

The Challenge
This error occurred when a WPF component was used with a System.Timers.Timer’s Elapsed event. The issue here was that the calling thread which in this case was a .dll file or a class library was not an STA. An STA is a logical container that contains only a single thread.

The Solution
One solution for the problem is to use a delegate to call the function within. The other solution was to use a System.Windows.Threading.Timer which creates a thread within STA. In this dissertation, the System.Windows.Threading.Timer was used.
5.2 Implementation Highlights

After discussing the challenges faced during the implementation, this section discusses the highlights of this dissertation.

1. COMPONENT BASED DEVELOPMENT

The main highlight of this dissertation is that the desktop application was built on a component-based model. This means that the entire application was built separately as different individual components and then plugged in at the end. For example, the Logger component was built separately. This logger component can be used with any other projects or applications. Similarly the DatabaseConnector, UserInfo are also individual components that can be used separately in other project, dissertations or applications. Thus about 50% of the components in this application can be reused.

2. FLEXIBLE DATABASE CONNECTOR

Another important highlight is the DatabaseConnector component. The speciality of this database connector is that it can be used to connect to any MySQL database. Also in future if the tables used in the current dissertation are altered, added, deleted or renamed, then no code needs to be changed. The only change that would be required would be to the XML file (Full file attached in Appendix C).

For example, to execute a Select statement on the products table, the following would be the code.

```java
DatabaseConnector.DbFactory.getImplementor("Select").
  executeQuery("products2", alValues);
```

where alValues is the values for the conditions.

This code would pick out the portion below from the whole XML file and use it to generate and execute the query.

```
  <products2>
    <select>
      <stmt>Select * from products where</stmt>
      <condition>product_id =</condition>
    </select>
    <delete>
      <stmt>delete from products where</stmt>
      <condition>product_id =</condition>
    </delete>
    <insert>
      <stmt>insert into products values</stmt>
    </insert>
  </products2>
```

If in future, during an enhancement, the name of the table changes from ‘products’ to ‘CustomerProds’ or if the primary key column name is changed from ‘product_id’ to ‘prod_id’ or even if the number of conditions needs to be added, then the only change that would be required would be to the XML file. The changed XML file would be as follows.

```
<products2>
  <select>
    <stmt> Select * from customerproducts where </stmt>
    <condition> prod_id = </condition>
    <condition> product_name = </condition>
  </select>
  <delete>
    <stmt> delete from customerproducts where </stmt>
    <condition> prod_id = </condition>
  </delete>
  <insert>
    <stmt> insert into customerproducts values </stmt>
  </insert>
</products2>
```

Also the InsertImplemetation class is designed in such a way that even if the number of columns is added or reduced, no change is required to code.

Thus the DatabaseConnector component was designed and created keeping in mind a flexible way to make changes and also keeping in mind future expansions.

### 3. FUTURE ENHANCEMENTS

This dissertation was built keeping in mind the possibilities of change and extensions in future. Hence various patterns that enable future extensions by promoting loose coupling concepts were created. For example the Factory Pattern was used extensively. For further clarity, let us take the example of the ReportGenerator component. The design is such that any number of new Report types can be added. All that is needed to be done is to create the class that implements the IReport interface and add the same in the ReportGenerator class.

Another example is that, if future enhancements requires the change in the implementation logic of a class say CostAnalysis, then a new class that implements the ICostAnalysis can be created and the same can be added to the CostAnalysisFactory.
5.3 Conclusion

This chapter highlights some of the challenges faced during the implementation of the IT business solution for Avalon Office Supplies. Among the challenges described were, writing to Excel sheets, creating an XML file attachment without creating the file physically, handling multiple suppliers for a product etc.

After the challenges were detailed, this chapter described the project highlights. The first of the highlights and areas of novelty was developing the desktop application using component based development in which 50% of the components can be reused in other projects, dissertations or application without any change of code. The next was the DatabaseConnector component that is flexible enough to allow the change of table names, column numbers, query conditions without any change in code. The only change would be in the DatabaseConnectorXML config file. The next highlight is that the entire enterprise solution was built keeping in mind future enhancements and extensions.

The next chapter discusses the evaluation and testing done as a part of this dissertation.
CHAPTER 6: TESTING

The previous chapter highlighted the challenges faced during the implementation of project. It also described the highlights of the project. In this chapter we discuss the testing that was done on the system. The components tested were the desktop application, the mobile application and the web application.

6.1 Desktop Application

The desktop application was tested by installing the application in a system. The first test was to check the data updates. A product was added, updated and deleted. Since the customer and supplier component’s functionalities were exactly the same only the result of the addition, updates and deletion of products is shown in this report. The rest of the screenshots for testing are available in Appendix E. All data updates were successful during testing.

6.1.1 Testing DML functions

Adding product to product table

Test Procedure

The following figure shows the Products table before data was added.

![Figure 26: Products Table before Addition](image)
A Product was added using the desktop application as shown below.

![Application - Add Product](image)

**Figure 27: Application - Add Product**

The following figure shows the product table after adding the product named ‘Printer’

<table>
<thead>
<tr>
<th>Product_id</th>
<th>product_name</th>
<th>brand</th>
<th>Description</th>
<th>Selling_Price</th>
<th>notes</th>
<th>SL_No</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Monitor</td>
<td>samsung</td>
<td>Computer Screen</td>
<td>1000</td>
<td>NULL</td>
<td>9922748</td>
</tr>
<tr>
<td>2</td>
<td>Monitor</td>
<td>samsung</td>
<td>Computer Screen</td>
<td>360</td>
<td>NULL</td>
<td>99225434</td>
</tr>
<tr>
<td>3</td>
<td>Monitor</td>
<td>Canon</td>
<td>Computer Screen</td>
<td>590</td>
<td>NULL</td>
<td>99225567</td>
</tr>
<tr>
<td>4</td>
<td>Keyboard</td>
<td>Epson</td>
<td>Computer Interface</td>
<td>110</td>
<td>NULL</td>
<td>5552234</td>
</tr>
<tr>
<td>5</td>
<td>Keyboard</td>
<td>Epson</td>
<td>Computer Interface</td>
<td>1000</td>
<td>NULL</td>
<td>992274</td>
</tr>
<tr>
<td>6</td>
<td>Toner</td>
<td>HP</td>
<td>Xerox</td>
<td>80</td>
<td>NULL</td>
<td>66329734</td>
</tr>
<tr>
<td>7</td>
<td>Toner</td>
<td>HP</td>
<td>Printer Toner</td>
<td>1000</td>
<td>NULL</td>
<td>6632934</td>
</tr>
<tr>
<td>8</td>
<td>Toner</td>
<td>Canon</td>
<td>Printer Toner</td>
<td>120</td>
<td>NULL</td>
<td>6632934</td>
</tr>
<tr>
<td>9</td>
<td>Stapler</td>
<td>Steal</td>
<td>Office Stapler</td>
<td>20</td>
<td>NULL</td>
<td>38443322</td>
</tr>
<tr>
<td>10</td>
<td>CPU</td>
<td>HP</td>
<td>CPU= AMD 9570, 4GB</td>
<td>159</td>
<td>NULL</td>
<td>59454322</td>
</tr>
<tr>
<td>11</td>
<td>Printer</td>
<td>HP</td>
<td>Laser printer</td>
<td>300</td>
<td>NULL</td>
<td>23796255</td>
</tr>
</tbody>
</table>

**Figure 28: Products Table after adding product**

*Test Result: Success*
Updating the selling price from ‘300’ to ‘350’

To test the update functionality, the product Id was entered and ‘Show’ button was clicked. All the data of that product was shown as can be seen in the figure below.

![Figure 29: Update product](image)

The relevant data is changed (Selling Price 300 to 350) and the ‘Update’ button is clicked.

![Figure 30: Application - Product Updated](image)
The following figure shows that the database was updated.

![Figure 31: Products Table after product Updated](image1)

**Test Result: Success**

*Deleting product ‘Printer’ from database.*

To test deleting data from the database, the product Id is entered and ‘Delete’ button is pressed.

![Figure 32: Application - Delete Product](image2)

The data was deleted as can be seen from the figure below.

![Figure 33: Products Table after product deleted](image3)

**Test Result: Success**
Searching for product id to perform ‘Update’ or ‘Delete’

To search for the product Id, one can use the search functionality as shown below.

Figure 34: Application - Search products

The Product name, brand and serial number is then entered into the spaces as shown and the Search button is clicked. The result appears as shown below.

Figure 35: Application - Search Results

Test Result: Success
If the data does not exist a message pops up notifying the unavailability.

![Figure 36: Application - No result found](image)

Test Result: Success
6.1.2 Testing Automatic Cost-Analysis Creation

Test Procedure

In order to test the automatic creation of cost analysis, the enquiry form (Excel Sheet) was filled as shown below.

![ENQUIRY FORM](image)

It was then attached and sent to the monitored mailbox.

![Mail Arrives](image)
The automatic mail processing system then picked the email, marked it as read, processed the attachment and notified with the alert as shown below.

Figure 39: Mail read and Notification
The cost analysis created is shown below.

![Figure 40: Auto Created Cost Analysis](image)

**Test Result: Success**

As one can see, the detail of the two products whose data is not present in the database was marked as N/A (Not Available). The unknown data can then be filled in manually or using the ‘Alter Cost Analysis’ function provided. However, before using the ‘Alter Cost Analysis’ function the Product data and the Product Supplier Data must be add to the database. This can also be done using the desktop application.
6.1.3 Testing Create Quotation

Test Procedure

To test the create quotation functionality, we shall use the cost analysis created in the previous section (Enquiry Number 57). Using the application under the ‘Quotation’ tab, the ‘Create Quotation’ button is clicked. In the Enquiry No section, ‘57’ is entered and ‘Create’ button is clicked.

Figure 41: Application - Create Quotation

The quotation was created. As you can see below the products whose data is not available (and was not entered manually or through the application) was excluded from the quotation.

![Created Quotation]

Figure 42: Created Quotation

Test Result: Success
6.1.4 Testing Send Quotation

The application also allows automatically creating an email with the fields such as ‘To’, ‘Subject’ filled. The body of the mail as well as the attachment is also automatically filled in.

Test Procedure

In order to test this, the quotation number is entered in the place for Quotation Number as shown below.

![Figure 43: Application - Send Quotation](image)

The Send Quotation button was clicked. The mail with the ‘To’ and 'Subject’ field was populated. The appropriate attachment was attached. The body of the mail was also populated.
If the quotation that is searched for does not exist an error message pops up as shown below.

![Create Quotation - Error Notification](image)

**Figure 45: Create Quotation - Error Notification**

**Test Result: Success**

Since the Invoice Creation had the same functionality the screenshot for the same are not shown in this report. The test was a success. The screenshot for the testing is available in Appendix E.
6.1.5 Testing Report Generation

Test Procedure

To test the report generator, the report manager section was selected. From the Reports Manager, the Product Report was selected as shown below.

![Application - Generate Report](image)

Figure 46: Application - Generate Report

The product whose report was required, was selected and the starting year was entered in the textbox as shown below.

![Application - Generate Report (2)](image)

Figure 47: Application - Generate Report (2)
The report was generated as shown below.

![Generated Report](image)

**Figure 48: Generated Report**

**Test Result:** Success

### 6.1.6 Testing Report Export

**Test Procedure.**

To test the export functionality, the Product Report was exported using the ‘Export’ button as shown below. A success message pops up on successful creation of the report.
6.1.7 Testing Non-Super user functionality

A User (not a Super-User) will have limited functionality. For example, the User Update is disabled, also the Delete functional of the Product, Customer and Supplier Section is disabled. Further, a ‘User’ is not able to alter cost analysis.

Test Procedure

To perform the test, the user logged in as ‘User’ i.e. a non-Super User.

Test Result: Success
As one can see from the above figure the ‘User Admin’ button which appears below the ‘Reports Manager’ button is invisible.

The above figure shows the ‘Delete Product’ and the ‘Delete Product Supply Options’ button is invisible. Similarly the ‘Delete Customer’ and ‘Delete Supplier’ button was also invisible.
The following figure shows that the ‘Alter Cost Analysis’ button is also invisible.

![Application - Cost Analysis Menu (non-Super User)](image)

**AVALON Office Equipments**

**Figure 54: Application - Cost Analysis Menu (non-Super User)**

**Test Result: Success**
6.2 Web Application

6.2.1 Testing Appearance on Different browsers

The web application was initially tested for its appearance in different browsers, mainly, Internet Explorer, Mozilla Firefox and Opera.

Test Procedure

The web page was opened in multiple browsers.

The following figure shows the web page on Internet Explorer.

![Figure 55: Web Page in Internet Explorer](image)
The following figure shows the web page on Mozilla Firefox.

![Figure 56: Web Page in Firefox](image)

The following figure shows the web page on Opera.

![Figure 57: Web Page in Opera](image)

Thus the appearance of the page on multiple browsers was the same.

**Test Result: Success**
6.2.2 Testing XHTML Validation.

Test Procedure

To test the validity of the XHTML page, the html file was loaded onto http://validator.w3.org/#validate_by_upload

The result was as follows.

![Figure 58: XHTML validation result](image)

**Test Result: Success**
6.2.3 Testing functionality.

Test Procedure

In order to test the functionality of the web application, the web form was filled as shown below and sent.

![Web Enquiry Form Filled](image)

On clicking the Submit Query, the data in the form was captured and an XML file was created as an attachment and an email was sent to the monitored mailbox as shown below.

![Mail from web application](image)

If the product you are enquiring about does not appear in the options above, please fill in the details below.

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Brand</th>
<th>Product Sl. No</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Printer</td>
<td>Hp</td>
<td>123432</td>
<td>5</td>
</tr>
</tbody>
</table>

Submit Query

Figure 59: Web Enquiry Form Filled

Figure 60: Mail from web application
The XML file created was as follows.

```xml
<WebFormValues>
  <CustomerInformation>
    <CompanyName>Al Zabeel</CompanyName>
    <ContactPerson>Mr. Rajiv Singh</ContactPerson>
    <Phone>983499827</Phone>
    <Email>rajiv@zabeel.com</Email>
  </CustomerInformation>
  <DDLInformation>
    <Product>
      <ProductName>Monitor</ProductName>
      <Brand>samsung</Brand>
      <ProductSINo>9927348</ProductSINo>
      <Quantity>10</Quantity>
    </Product>
    <Product>
      <ProductName>Toner</ProductName>
      <Brand>Hp</Brand>
      <ProductSINo>66324734</ProductSINo>
      <Quantity>2</Quantity>
    </Product>
  </DDLInformation>
  <TXTInformation>
    <Product>
      <ProductName>Printer</ProductName>
      <Brand>Hp</Brand>
      <ProductSINo>123432</ProductSINo>
      <Quantity>5</Quantity>
    </Product>
  </TXTInformation>
</WebFormValues>
```

Figure 61: Created XML file attachment

This XML file was then successfully processed by the Automatic Mail processing System.

**Test Result: Success**
6.3 Mobile Application

6.3.1 Functionality Test

Test Procedure

To test the mobile application functionality the J2ME simulator was used. The Quotation created in Section 6.1.3 was searched for in the mobile application as shown below.

![Figure 62: Mobile Application](image)

The data displayed was accurate as can be seen.

Test Result: Success
6.3.2 Speed Test

The mobile application was also packaged and installed in a Nokia N73 phone and tested for real life speed. Speed is an important factor in the success of a mobile application. The following figure shows the mobile application in the live environment.

![Mobile Application (2)](image)

Figure 62: Mobile Application (2)

The speed of the application was found to be good. There was no lag in moving from one screen to another.

Test Result: Success.
CHAPTER 7: EVALUATION

The previous chapter highlights the implementation challenges and the highlights of this dissertation. In this chapter the evaluation results are discussed.

7.1 Product Evaluation

The first evaluation done was of the solution developed as a part of this dissertation with the existing products in the market. For the study, two products, Focus and Tally were used. The results of the evaluation is shown below

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Functionality</th>
<th>IT solution being developed</th>
<th>Focus[32]</th>
<th>Tally[33]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Automatic reading of emails</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>Automatic processing of enquiries</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>Automatic creation of cost analysis</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Options to create quotation and pro forma invoice</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>Options to add delete and update database data of product, customer and suppliers</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>Options to customise GUI options</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Modularised functionalities</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>Profit and Loss Report Creation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>Report Creation per Product</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>10</td>
<td>Report Creation per Customer</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>11</td>
<td>Summation report</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>12</td>
<td>Graphical view of report</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>13</td>
<td>Pay Roll Module</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>14</td>
<td>HR Module</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 1: Functionality Comparison
7.2 User Evaluation

To perform the user evaluation of the project, the evaluation form (available in appendix B) was used. This was filled by the staff members of Avalon Office Supplies. After analysing the results and taking the average of the responses the following were the observations.

1. Is the new web enquiry form more helpful in streamlining enquiry data inflow?
   It was said that the new web enquiry form is better than the existing one. It will allow users to easily choose the products, the brand and the serial number. It also allows the users to input the name, brand and serial number of products that are not available from the drop downs. Thus the new web form was said to be better than the old one.

2. Will the mobile application help in improving business process.
   It was agreed that the mobile application will definitely help in if not improving at least assisting business process. However, it was also mentioned that business has not grown to a scale where true potential of the mobile application can be realised. Hence the true potential can be realised as business grows with time.

3. Will the automated mail processing reduce effort?
   The common reply to this question was that it greatly reduces efforts as now one will not have to read emails and attend phone calls and note the product details. There was a room for error when noting the product details for enquiries that comes via phone call. With the new application this room for error is nullified. It was also said that it will allows them to be free from reading about 30% of emails and also frees them from phone calls. The automated mail processing functionality was said to be “An excellent bit of innovation”.

   However a suggestion that was given was that the GUI could have a small sign (similar to unread email sign of Outlook Express) when a cost analysis is created as the notification may go unnoticed if the staff on whose system the Automated Mail Processor is running is not at his seat when the cost analysis arrives.

4. Does the GUI give full control over data in the database?
   There was a common agreement for this question. It was said that it does allow add, delete, update of customer, products, suppliers data.

5. Is the graphical report generator helpful?
   The animated graphical report generator was appreciated well. One of the comment given was “It is excellent work”.

6. Is the cost analysis creation accurate and please comment on the alter cost analysis.
   The cost analysis was said to be accurate and very helpful. The alter cost analysis is also very helpful.
However, a very important suggestion was received here. The suggestion was — *the view of the elements of the cost analysis could be better if it is in a writable grid format. This would help improve readability*.

7. **Is the GUI user friendly and easy to use?**
The GUI was said to be user-friendly.

8. **Separating the users as user and super user. Please comment.**
Having two sets of user was appreciated.

However an improvement suggested was *“it would be better if the super user was able to give users individual abilities. For example, a super user would like to give a user the ability to delete customer only and not the other functionalities such as delete products etc that the superuser can perform. At present, the only option is to make this user a super user and this would give the user access to all delete records and also alter cost analysis which is not required”.*

9. **How does this desktop application compare to Focus and Tally?**
The desktop application was said to be very competitive. There are various functionalities that are not available in focus and tally. But, on the other side, it was highlighted that Focus has other functionalities that are not available here for example modules such as payroll, HR, admin etc.

10. **Is the Quotation and invoice component useful?**
This question had a unanimous positive answer. It was said that it is useful. Especially the ‘send mail’ where the ‘from’, ‘to’, ‘subject’, body of the mail along with the corresponding file is attached and ready for send. The automatic creation of the same at the click of a button was highly appreciated.

After critically analysing the replies in the evaluation form, the conclusion drawn is that the dissertation is successful with respect to meeting its objectives. It allows reducing efforts of the staff by automating various processes like reading emails, creating cost analysis etc. It also gives complete control over business data. Further, the graphical display of reports has helped quick analysis of data. Creating Quotations and Invoices and preparing emails with the ‘from’, ‘to’, ‘subject’ and ‘body’ populated appropriately with these quotations and invoices as attachment was greatly appreciated.

However, there were valuable suggestions that were given. One of the suggestions was to have a small symbol in the task bar when a cost analysis is created. Another valuable suggestion was to be able to give users individual access abilities rather than having two states of access only. The third suggestion was to show the details of cost analysis in a grid view instead of text boxes as is now.

Hence the dissertation has moved in the right step with scope for improvements and enhancements. More details of the enhancements and conclusions are furnished in the next chapter.
CHAPTER 8: CONCLUSION AND FUTURE WORK

8.1 CONCLUSION

In the previous chapter, the dissertation project and its outcomes were evaluated and critically analysed. This chapter concludes this dissertation and also highlights the possible future enhancements.

This dissertation was a success in meeting the set goals. The following table illustrates the objectives and the success criteria.

<table>
<thead>
<tr>
<th>Objectives</th>
<th>Success Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design and development of a database structure.</td>
<td>Successfully Implemented</td>
</tr>
<tr>
<td>Structuring the incoming enquiries and requests using an enquiry form –</td>
<td>Successfully Implemented</td>
</tr>
<tr>
<td>MSExcel and Web form</td>
<td></td>
</tr>
<tr>
<td>Desktop Application to continuously monitor email inbox</td>
<td>Successfully Implemented</td>
</tr>
<tr>
<td>Reading relevant emails and picking up relevant attachments</td>
<td>Successfully Implemented</td>
</tr>
<tr>
<td>Reading the data in the attachments. (Excel form – sent via email and XML</td>
<td>Successfully Implemented</td>
</tr>
<tr>
<td>files sent via web enquiry form)</td>
<td></td>
</tr>
<tr>
<td>Automatic creation of Cost Analysis (Pre-Quotation)</td>
<td>Successfully Implemented</td>
</tr>
<tr>
<td>Alert when a cost analysis is created</td>
<td>Successfully Implemented. Was suggested to add</td>
</tr>
<tr>
<td></td>
<td>functionality of image in tool bar</td>
</tr>
<tr>
<td>Options to alter the automatically created cost analysis</td>
<td>Successfully implemented. However, a change is</td>
</tr>
<tr>
<td></td>
<td>required for the display of the cost analysis in the</td>
</tr>
<tr>
<td></td>
<td>GUI. Suggested to use a writable data grid.</td>
</tr>
<tr>
<td>Options to create a quotation from the cost analysis</td>
<td>Successfully Implemented</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Options to automatically send created quotation without the need to create a mail and attach the file</td>
<td>Successfully Implemented</td>
</tr>
<tr>
<td>Options to create a pro forma invoice from the quotation</td>
<td>Successfully Implemented</td>
</tr>
<tr>
<td>Options to automatically send created invoice without the need to create a mail and attach the file</td>
<td>Successfully Implemented</td>
</tr>
<tr>
<td>Add, Update and Delete the Products, Suppliers, and Customers data</td>
<td>Successfully Implemented</td>
</tr>
<tr>
<td>Generation of reports – Graphical</td>
<td>Successfully Implemented</td>
</tr>
<tr>
<td>Option to export reports to Excel</td>
<td>Successfully Implemented</td>
</tr>
<tr>
<td>Design and development of a mobile phone application.</td>
<td>Successfully Implemented</td>
</tr>
</tbody>
</table>

**Table 2: Success Criteria Table**

Apart from meeting the objectives set, the business solution that was developed as a part of this dissertation was appreciated by the staff at Avalon Office Equipments. The idea of automating redundant work such as reading emails that contained enquiries, creation of cost analysis etc were said to be innovative. The GUI was said to be user-friendly and the options allowed overcoming the limitations of the existing software. Further, the mobile application was said to be very useful and its true potential can be realised as business grows. Thus overall the dissertation can be said be a success.
8.2 FUTURE WORK

In order to extend this dissertation, the following areas were identified.

More components that help business automation, and increase efficiency can be created. Components such as HR, Pay Roll, Admin modules can be introduced by studying various business processes. This will allow this dissertation to turn into a complete office solution product that can be used from small to medium size businesses. The design of the components that constitute this dissertation facilitates expansion. Even the GUI was designed keeping in mind expansion in future. For example, instead of having all the buttons in one screen, it was divided into tabs and or distributed in the application menu.

More reports can be added as business requires. The design of the ReportGenerator module is such that it can easily accommodate adding reports (as discussed in Section 4.2 [3]). This is possible due to the use of the Factory pattern. The advantage of this is that it makes the product developed more general as different businesses require different reports.

Various configuration details can be moved to a configuration file. At present the location where the created invoice, quotation etc are placed is hard coded. Details such as these can be moved to a XML configuration file. This will allow configuration of this dissertation product. An installation wizard can also be developed and the configuration details can be entered through the wizard.

As was pointed out during the evaluation, the view of the elements of the cost analysis could be changed to a writable grid format. This would help improve readability as well as make the ‘Alter Cost Analysis’ section more user friendly. Further, a small image can be displayed in the taskbar when a cost analysis is created. This will allow the user to be notified of the creation of a cost analysis even if he misses the notification. Thirdly, the Super User functionality can be made more flexible i.e. a Super User should be able to give a User permissions on specific functions.

One aspect that can be improved greatly in this dissertation is that of security. Security aspects such as network security as well as encryption of data when travelling through the network can be implemented. Also password encryption within the database can be used. A password recovery mechanism can also be implemented.
References


[18] Vikram Vaswani Title: A beginners guide PHP. USA McGraw-Hill Companies USA 2009


[21] Hans Bergston Title: Java Server Pages California USA, O’Relly and Associates. 2004

[23] Deitel, Title: How to program 5th edition. Upper saddle river new jersey 2005 p 918


[27] W3C [home page on the Internet] Available from: http://www.w3.org/TR/xhtml1


[29] Sing Li and Jonathan Knudsen Title: Beginning J2ME 3rd Edition. USA, Apress 2005


APPENDIX A: SOURCE CODE

Available in CD.
APPENDIX B: EVALUATION FORM

Date __________________

I thank you for taking a few minutes of your time to fill this evaluation form. This would help me critically analyze the outcome if my dissertation. Please fill in the answers to the questions provided in the space below.

<table>
<thead>
<tr>
<th>1. Is the new web enquiry form more helpful in streamlining enquiry data inflow?</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Will the mobile application help in improving business process?</td>
</tr>
<tr>
<td>3. Will the automated mail processing reduce effort and increase efficiency?</td>
</tr>
<tr>
<td>4. Does the desktop application give full control over data in the database?</td>
</tr>
<tr>
<td>5. Is the graphical report generator helpful?</td>
</tr>
<tr>
<td>6. Is the cost analysis creation accurate and please comment on the alter cost analysis.</td>
</tr>
<tr>
<td>7. Is the desktop application user friendly and easy to use?</td>
</tr>
<tr>
<td>8. Separating the users as user and super user. Please comment.</td>
</tr>
<tr>
<td>9. How does this desktop application compare to Focus and Tally?</td>
</tr>
<tr>
<td>10. Is the Quotation and invoice component useful?</td>
</tr>
</tbody>
</table>
APPENDIX C: DB CONFIG XML

Available in CD.
APPENDIX D: COST-ANALYSIS, QUOTATION, INVOICE (SAMPLE)

Auto Created Cost – Analysis
# Quotation

**AVALON OFFICE EQUIPMENTS**  
P.O. BOX 38418, SHARJAH, U.A.E.  
MOBILE: 050-4820327 TEL: 06-5735561 FAX: 06-5722859  
e-mail: avalone@emirates.net.ae  
Visit: http://www.avalonsupplies.com

<table>
<thead>
<tr>
<th>QTN</th>
<th>12</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Bill No</th>
<th>Product</th>
<th>Item Name</th>
<th>Item Code</th>
<th>Quantity</th>
<th>Unit Price</th>
<th>Total</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Toshiba</td>
<td>Keyboard</td>
<td>4853234</td>
<td>17</td>
<td>110</td>
<td>1870</td>
<td></td>
</tr>
</tbody>
</table>

This is a computer generated quotation authenticated without sign.

Placing an Order With Us Saves You Time And Money

**Avalon Supplies - One Source ... Many Brands**  
Your Most Valuable Provider
Auto – Created Invoice

AVALON OFFICE EQUIPMENTS
P.O. BOX 38148, SHARJAH, U.A.E.
MOBILE: 050-4820327 TEL: 06-5735561 FAX: 06-5722859
e-mail: avalonee@emirates.net ae  visit: http://www.avalonsupplies.com

COPIERS, PRINTERS, FAX MACHINES & OFFICE SUPPLIES

PROFORMA INVOICE

<table>
<thead>
<tr>
<th>Si No</th>
<th>Brand</th>
<th>Item Name</th>
<th>Item Si No</th>
<th>Quantity</th>
<th>Unit Price</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Toshiba</td>
<td>Keyboard</td>
<td>4883234</td>
<td>7</td>
<td>110</td>
<td>770</td>
</tr>
</tbody>
</table>

This is a computer generated quotation authenticated without sign.

Placing an Order With Us Saves Your time And Money

Avalon Supplies - One Source ... Many Brands          Your Most Valuable Provider