WXES3182 Latihan Ilmiah Tahap Akhir 11

Framework to Build Component Based E-Manufacturing

Goh Ki Wei WEK010081

Supervisor : Mr.Teh Ying Wah
Moderator : Puan Maizatul bt. Akmar Ismail

Abstract

This documentation report is a fully description about our project. The project title is to build a technique and framework to develop a component-based e-manufacturing system. This project is undertaken by a group of four members.

The framework consists of four modules, which are Sales module, Material Requirements Planning (MRP) module, Production module, and Material Management module. Fulfilling minimum business requirements for a manufacturer is the ultimate goal of this project.

The essence of the framework is able to support reusable approach. Furthermore, framework is extensible or easy to evolve as the business requirement is changing.

We have conducted survey and research upon the cutting edges technologies, which could be leveraged and will be adopted in the development of the project. We have came across the literature review and study about the current available technology for operating system, database management system, data access paradigms, system architecture, system development tool, and programming language.

The software development life cycle of this project is actually following a particular development methodology. Incremental and iterative development approach will be ideal to develop the component-based framework. This methodology comprises of a sequence of steps along the software development phases, including preliminary investigation, system analysis, system design, implementation, system integration, and system testing. Meanwhile, Unified Modelling Language (UML) is being used to modelling and visualise the physical and conceptual of the software development process phases.

Acknowledgement

I would like to gratefully acknowledge the contributions of several people who have directly or indirectly supported me through the development of my project.

First of all, I would like to thank Mr. Teh Ying Wah, for being my supervisor and for the constructive advice, generous guidance, encouragement, support, dedication and supervision along the project.

Special thanks are also due to Puan Maizatul Akmar Ismail as the project moderator for the precious time to attend the viva presentations and contributed constructive advice, suggestion and ideas to further enhance value of this project.

Not forgotten also, I would like to take this opportunity to express my appreciation to my friends and course mates that have been accompanying me all the time in completing this project.

The appreciation is also dedicated to others who have giving me valuable supports and encouragement for this project.

Last but certainly not the least; I would like to take this opportunity to acknowledge the resource support of the FSKTM which greatly facilitated the process of the project.

Thank you.

List of Figures

List of Tables

Table of Contents

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

1.1 Project Introduction

E-manufacturing is about the connection and the use of the Internet and e-business technologies in manufacturing industries[1]. It allows manufacturers to speed up and slim down everything from design to manufacturing in a single system. E-manufacturing actually enables integration of all aspects of manufacturing from sales, marketing, customer service, material management, supplier relationships, logistics, and production management to decision making, scheduling and planning for top level management and so on.

Manufacturers are providing with real-time information for on-going improvement and decision making in an E-manufacturing system. Furthermore, integrated manufacturing execution system allowed efficient and effective method of information sharing and it also streamline the daily operation. Not only does it reduces cost and shortens the time-to-market, but also brings the competitive advantages.

The main objective of this project is to develop a framework for manufacturer that leaded them to build a component-based E-manufacturing system. Components will be built and embedded in the framework to assist manufacturer in building an E-Manufacturing system.

The framework comprises of four main modules: Scheduling module, Production module, Inventory module and Sales module. Information of each modules can be shared and transfer to each others. So this will meet the main aim of E-manufacturing which is to bridge the gap between the shop-floor and the management level.

1.2 Project Definition

Before proceeding to next chapter, importance terms that are used in the following chapters will be explained in below:

- **Framework**

  A framework is a set of classes providing a general solution that can be refined to provide an application or a subsystem and it is generally targeted for a particular application domain, such as user interfaces, business and data processing systems, telecommunications, or multimedia collaborative work environments. A framework can be more than a class hierarchy. It is a reusable semi-complete application with several dynamic and static components that required only little customization in order to produce different user-specific applications. (Fayad *et al*, 1999)

- **Component**

  Components are a physical and replaceable part of the system that complies with interfaces and they can be plugged into a system. There are four essential characteristic of a software component model which is the component, the component socket, the ability to cooperate with other components, and the components' user.

- **MES (Manufacturing Execution System)**

  MES is an on-line extension of the planning system with an emphasis on execution or carry the plan. (McClellan, 2001) The word MES literally implies the transition of the plan into implementations. Therefore, 'execution' means producing products, running the machine, update order priorities, making and measuring parts and so on.

1.3 Project Objective

Below are the objectives of our project:

- Our main objective is to develop a framework for creating a component-based e-Manufacturing system.
- The framework must support reusable approach.
- The framework must capable to build an e-Manufacturing system which is able to fulfil minimum business requirement for different types of manufacturers.
- The component-based framework able to provide faster delivery of system compare with paged-based system.
- The component-based framework has to be suitable and applicable in different kind of manufacturers.

Below are the objectives of my module:

- The main objective is to develop a material management module in the framework which allows the user to create an inventory/material management system.
- The inventory module should support reusable approach.
- The inventory module must performed information sharing with scheduling, sales module and production modules.
- The inventory system created from the framework is able to fulfil minimum business requirement for general inventory system.

1.4 Project Scope

Database generation

System will use a wizard to lead user to create database for material management module and others modules in the e-Manufacturing system. User can able to select fields from a provided list for each table. User is allowed to perform add, edit and delete fields functions.

User Interface Builder

In material management module and others module, system will show user with a list of interface templates to be selected. The user-interfaces for the material management module will be automatically generated according to the templates and fields selected by user at the early stages. This mean user does not need to design and create user-interfaces for each module, which will surely save a lot of time.

Report Generator (graphical chart)

After the interfaces generator has been completed, report generator will generate different kind of reports for different modules automatically based on the user specification. Material Management module could generate supplier report, purchasing report, and product summary report. Beside, user can chose type of graph to present such as pie chart, line chart, bar chart or text report.

Authentication

The authentication module enables system developer to determine the accessibility among each module by using the administration control panel to monitor the entire system effectively. The system developer will decide the accessibility of one module to other modules by using a checklist.

Material Management Module Scopes

Material management consists of three sub modules, which are:

Inventory management:

- System shall provide the key functions necessary to update and maintain for items and materials.
- System provides complete control and visibility of raw materials as it moves through the manufacturing process.

Purchasing management:

- System shall able to handles purchase order writing and the tracking of supplier information.
- System shall promote efficiency and accuracy in material purchasing

Shipping and receiving:

- System shall provide a central application within vantage to monitor incoming and outgoing items.

1.5 Project Motivation

E-manufacturing as a new generation of product development solution allows manufacturers all over the world to speed up and slim down everything from design to manufacturing. The main objective of e-Manufacturing is to bridge the gaps betweens top level management and shop floor. Below are the motivations that are initialising in this project:

- The software 'reuse' approach is commonly apply in software development industry (Sommerville, 2001).
- Customisation solution for small/medium-sized manufacturer system does not significantly show cost-effective when compare to software reuse approach.
- A well understanding of object structures is needed in Object Oriented approach where only expertises are able to handle this software solution. But, this is not happened in Component-based approach because it required no understanding about the programming needs.
- Component-based approach is much more reliable when compare to the web-based solution. It reduces the risk of hacking, cracking, virus attacking, intrusion or internet down.
- Component-based approach actually promise faster delivery of systems compare to customisation approach.
- Component-based has the edge over page-based system in terms of speed of the output.

1.6 Target User

Our system's main target user is the small-scale manufacturer. In this system, we only consider two types of user which is system developer and the end system user. This segregation is to ensure that there will no be any contradiction in terms of roles played by different users. In the mean while, we also enhance the data integrity and reliability of the system.

System Developer / Administrator

At this stage, system developers can use the framework developed from this project to create material management module and the others module, which is Sales Modules, MRP Modules and Production Modules. Besides that, system developers are entitled to create database for each modules by using database generator component. They are also required to create user interfaces base on their organisation business rules. Apart from that, system developers have to set on authentication mode for the whole system before it is delivered to end user. This is to ensure the security of the system has not been breach.

End System User

There will be two types of end system user, namely, high-level users and low-level users. Management employees are considered as high-level users, only they are provided with the authority to performing add new, update and delete relevant information or records. Besides, they are allowed to view all information of other modules (e.g. Sales, Scheduling, Production or Inventory). As for the low level users such as shop floor labours, they are just permitted to view certain information or performed certain input into the system. Modifying and deleting information are not inclusive in their tasks.

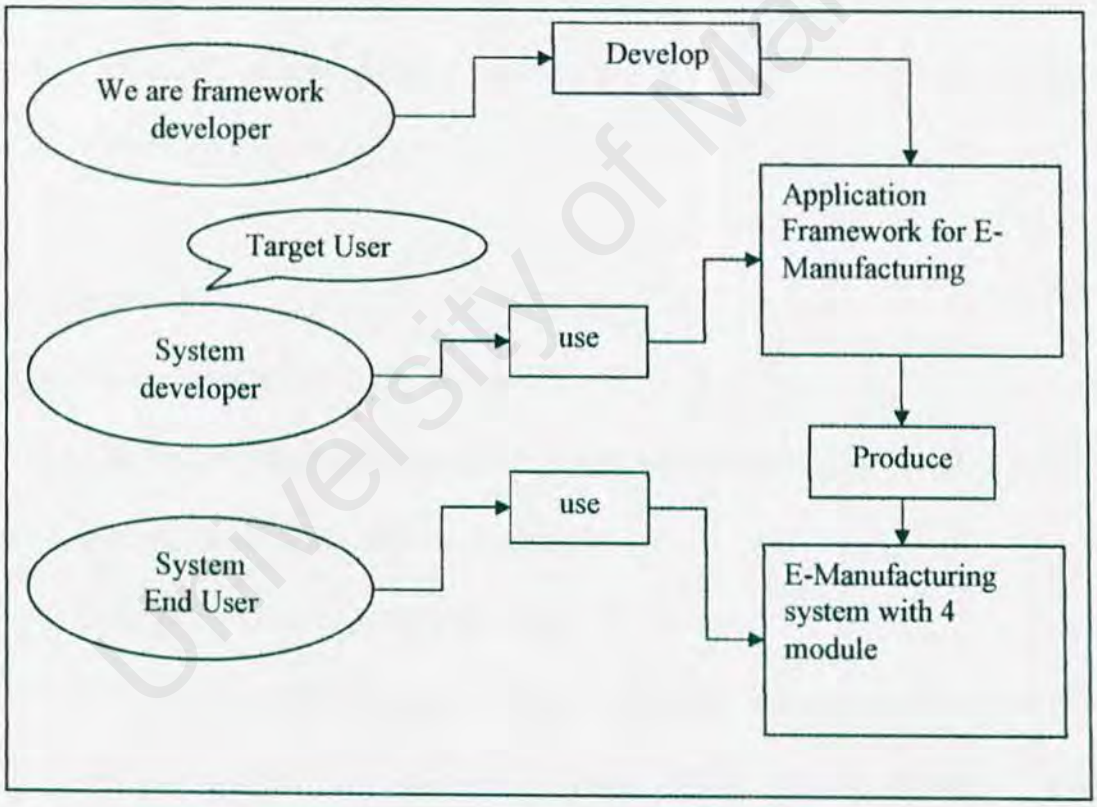

Figure 1.0 target user

1.7 Strength

Below are the strengths of the framework:

- The framework can be used or run in different desktop or computer because it is reusable.
- The e-Manufacturing system managed to evolve to satisfy the customer requirement needs changing.
- Information sharing is performed among each module.
- The delivery of E-Manufacturing system is promised in a short period of time without any hassle and dealing with the cumbersome of coding.
- The developed system is reconstructed base on the user required changing.
- Technical and programming expertise are no required when dealing with the system development.

1.8 Constraints

Below are the weaknesses of the framework

- The framework is well tailored for Small and Medium Manufacturer, but it is not applicable to big scale manufacturer.
- Debugging function is not providing in the system.
- The module in the framework does not cover the entire scope and element of e-manufacturing. For example financial module is not inclusive in this framework.

1.9 Expected Outcome

Below are the expected outcomes of the project:

- Developer is able to use the framework with important components to build a small e-Manufacturing system.
- Framework can be reused and implemented across a range of computers.
- The frameworks able to perform basic functions such as files saving, open files, create new files and so forth.
- Framework enables developer to generate database and create interfaces by using component.
- All modules (Scheduling, Inventory Management, Production and Sales) are able to link with each others and perform information sharing.
- Developer is able to generate graphical report such as chart, schedule, graphs and more.
- System able to perform basic functions which meets the important criteria such as stability, consistency, reliability and user friendly.
- Authentication module enhances the system security where only authorized user can access to the system according to their accessibility.

1.10 Project Schedule

ID	Task Name	Start	Finish
1	Preliminary Investigation	6/16/03	6/27/2003
2	Literature Review	6/30/2003	7/25/2003
3	Development Methodology	7/28/2003	8/1/2003
4	System Analysis	8/4/2003	8/22/2003
5	System Design	8/25/2003	9/15/2003
6	Implementation	10/17/2003	1/1/2004
7	System Integration and Testing	12/15/2003	2/20/2004
8	Documentation	6/16/03	2/27/2004

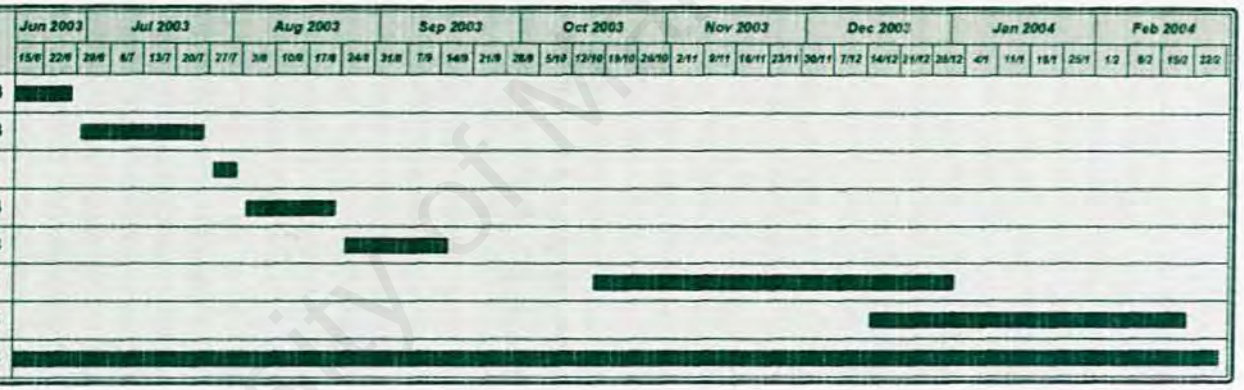

Figure 1.1 Project Schedule

2.0 Literature Review

Literature review is a research for knowledge and information that needed to develop this project. This helps to acquire a better understanding on the development tools that can be deployed during development. Besides, it also helps to gain better knowledge about development methodologies that can apply in this project.

Basically, this chapter is about some review on existing system that got the same or similar function. A study is carried on existing system about certain aspects such as the features, capabilities, the strengths and the weakness of the existing system. On the other hand, literature review enables the developers to study, analyse, and compare the relevant existing systems in the market.

2.1 Definition of Framework

A framework is a semi-complete application which integrated with dynamic and static components that can be customised to produce user-specified applications. A component framework is a dedicated and focused architecture of a few key mechanisms and the component level (By Jianxing Ruan). Component framework is also considered as a component where it allowed others component functions to "plug in".

To develop a component based framework for e-manufacturing system, a set of dynamic and static components which containing great variety of reusable method or function must be provided as a software development tools in a dynamical environment. This framework enables users to generate user-defined database, user-interface, and assign the codes to relate the database to user-interface by using the components inside. Formula can be assigned to the component so that it can performed certain calculation or manipulation of the data set.

There are quite a number of frameworks which are using object-oriented (OO) approach, but since the attention has shifted from framework and basic object-oriented paradigms by combining the benefits object-oriented design and programming with broad-scale of architecture viewpoints. Component-based software architecture and programming play a critical role in this trend. (Fayad, M. E. *et al*, 1999) Designing and developing frameworks, rather than classes, gives a much more flexible set of components to keep in your library. Ideally, to design a system, you select or build a set of frameworks, and fit them together. Therefore we have adopted the component-based approach as the core of the framework.

A framework also is defined as a skeleton of an application, where developer is given space to customize the application. The interface in the framework should define in between the components to ensure that these components can fit to each other and fit into the framework itself.

2.1.2 Application Framework Classification

There are three categories of application frameworks which based on their domain scope. Normally, a system or an application is developed by integrated a number of frameworks. (Fayad, M. E. *et al*, 1999)

These three classes of frameworks are:

- **System Infrastructure Frameworks** – This framework actually assist system developer to develop system infrastructure, such as communications, user interfaces and compilers.
- **Middleware Integration Frameworks** – A middleware integration framework actually support communication and information exchange among the components where it consists of a set of standards, library, and object classes. For example, CORBA, Java Beans, COM and DCOM.
- **Enterprise Application Frameworks** – These frameworks are dedicated on certain domain, such as telecommunication, financial systems. The framework can embed the domain knowledge and supports the development of end-user system or applications. This will allowed a wider range of system or application to be developed.

2.1.3 Benefit of Using Framework

There are many application frameworks that available in the market now, for example: agent-oriented application frameworks, system application frameworks, artificial intelligence application frameworks, business application frameworks, and etc.

Framework has become a mainstream and all levels of developer are likely to adopt frameworks technologies as their software development tool. (Fayad, M. E. *et al*, 1999)

Below are some of the benefits by using framework as the software development tool, where it can:

- Greatly reduce the software development time.
- Make the configuration connectivity and communications among the components or objects easier.
- Eliminate the cumbersome when dealing with error and coding during the system development.
- Embedded plenty of facilities, features, functions, and knowledge into the targeted domain.
- Can modified user-specific application by using the framework itself.
- Can provide framework with help, search engine, and example application for the particular domain as a guide for newcomer.
- Can adapt with the previous version application since they are using the same framework.

Others advantages of framework develop are:

- A framework itself is extensible; this skeleton of program can be extended to a complete application by add-in necessary components.
- Most of the time, applications developed by framework are more stable and reliable than ordinary application (develop without using a framework).
- All the applications developed by framework would have a standard pattern and maintainable.

2.2 Introduction of Component-Based Development

Component-Based Development is gaining ground for developing a high-quality, evolvable, and large software system. This reuse-based development emerged in the late 1990s. This technique is not a new chapter in software industry by now. Originally it was presented as early as 1968 on the NATO Conference on Software Engineering in a paper called "Mass-Produced Software Components" by Douglas Mclloroy. (Vaughn T., 1990) It will be much easier to combine the components to construct a software application rather than develop it from the beginning. Constructing an application under this new approach involves the composition of reusable and independent components.

2.2.1 What is a component

There are a lot of terms explaining components and its technology which might lead to confusion. This presentation attempts to reduce those terms by eliminating redundancies and explain and justify them. Below is the definition of component made by Christiansson which is widely accepted,

"A software component...

- *is independent and reusable,*
- *offers explicitly specified services through an explicitly specified interface,*
- *can affect/be affected by other software components,*
- *should have one documented specification*

Can have several independent implementation, i.e. one component can be implemented in several different programming languages and can have several executable (binary) shapes." (Christiansson B., 1999)

"A component is a binary unit of independent production, acquisition and deployment that interact to form a functional system". (Szyperski, course text) Component is very abstract as it encapsulated the functionality and then provides through interfaces. A collection of components are needed to develop an application. Generally, components are much easier to use than class libraries because software developers just needed to know how the components functions to use it. This save a lot of time where software developer are not required to study its structure ('black box') before the component is use. Beside, the component are reusable in others application where it can be applied in future version.

2.2.2 Why Component-Based

These are reasons why component-based approach is widely applied in software development:

- Portability – Component can be plugged into or unplugged from system due to system required needs.
- Support across applications – Component is able to support different applications.
- Applications are immovable – Most application consists of vast features which made them hardly to remove or upgrade independently where this situation can be avoided in component-based development.
- Simple to use – Component is much more easy to use than object class because it does not required software development to know detail knowledge of the object class.

2.2.3 Benefits of Component-Based Software Development

Below are the benefits of using component-based development:

- Component functionality can be reused in many applications.
- Assure higher quality and more reliable result.
- Component can be created with any languages or tools.
- Faster applications development process.
- Third party can not access to construction details

2.2.4 Drawbacks of Component-Based Software Development

The main difficulty with component-based development is the problem of maintenance and evolution. The source code of components is not available, this made component impossible to change to reflect any requirement changes. This leads to the increasing of the maintenance costs as a lot of additional works is required in order to reuse the components.

2.2.5 Component-Based Software Development Model

Figure 2.1 illustrates the model of Component-Based Software Development. Clients required no knowledge of how the component implements its interfaces. If the interface implementation is changed it will not affect the clients. Clients will only be affected if the interfaces are changed.

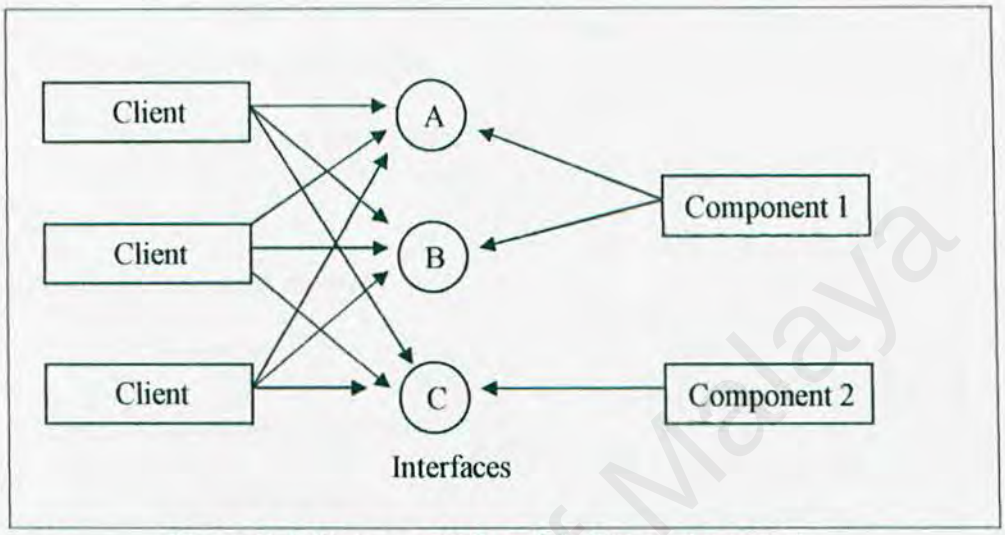

Figure 2.1 Component-Based Software Approach

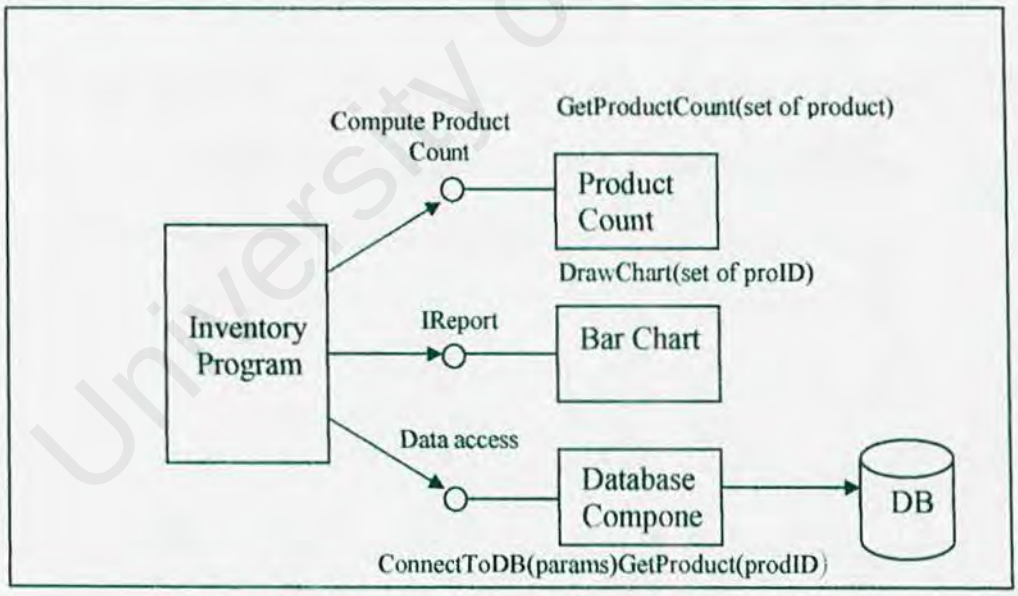

Figure 2.2 Services are provided by the component

2.2.6 Object-Oriented Vs Component-Based

Table 2.1 Component-Based Vs Object Oriented (ncst, 2001)

Object Oriented	Component-Based
▪ Can only be written by using Object-Oriented language. ▪ Highly dependencies such implementation, inheritance etc. ▪ Fine-grained units of composition. ▪ Designed to obey Object-oriented principles.	▪ Can be written in any language. ▪ More loosely coupled than objects. ▪ Can access multiple interfaces, and interface-oriented design. ▪ Designed to obey rules of the underlying component framework.

Microsoft's COM, COM+, DCOM, .NET framework, Sun's JavaBeans, EJB (Enterprise Java Beans), J2EE specifications, and COBRA specifications are the three popular software component technologies.

2.3 E-Manufacturing

"E-Manufacturing integrates customers, e-commerce systems and suppliers into manufacturing to provide an Internet-based strategic framework for factory." [2.1]

"E-manufacturing is concerned with the use of the Internet and e-business technologies in manufacturing industries." [2.2]

"A responsive manufacturing model that optimises the use of production assets based on information exchange from shop floor operations, across the enterprise and the extended supply chain." [2.3]

E-manufacturing as a new product development solution allows manufacturers to speed up and slim down everything from design to manufacturing in a system where it actually covers all the aspects such as financial, material management, production, sales, suppliers, scheduling, and management decision making and so on. E-Manufacturing used the e-business technologies to help manufacturers to gain competitive advantages by optimize the efficiency, productivity, and minimized the costs of production.

Make-to-order means that products are made when an order is placed. This is the objective that most manufacturers try to achieve now. As compared to older manufacturing model, promotes concept of mass production which always overloaded the inventory.

To avoid the inventory blooming, produce only what's needed and maintaining minimal or zero end-products is what Agile and lean manufacturing models been participating. On the other hand, constraint-based manufacturing tried to optimise the material flow in order to eliminate any production delaying.

Unpredictable and unforeseen event in the manufacturing process is a threat to manufacturer because it will cause them unable to fulfil the demand of the customer. Manufacturing forecasts and real-time information from shop-floor are important information for manufacture take decision making and action in order to minimise the possibility of unpredictable event occurs in manufacturing process. For example, a well schedule planning might reduce the possibilities of delay in delivery time where by effective process control can minimise the variation in the process.

Below are the benefits of implementing E-manufacturing system.

- **Increased Agility:** Quick response to customer demand as it is one of the key factors to gain competitive advantages.
- **Increased Profitability:** More effective control process where numbers of scrap and non-conformant products can be reduced as the fine quality products increased. Thus, this will increase the productivity with minimal cost.
- **Reduced Lead Times:** It can minimise the lead time required by having well scheduling and planning that slim down the main production and delivery flows.
- **Improved communications channel and knowledge sharing:**

  Top level management are able to share critical data or information with lower lever so that they can be always aware of what next steps needed to be carried on. This critical data is required in order to portray an accurate account of the day-to-day or hourly product manufacturing operations for continuous improvement and decision support.

2.4 Operating System

Software is designed to control the hardware of a specific computer system in order to allow users and application programmer to employ it easily. The operating system mediates between hardware and software (application programs). It handles the details of sending instructions to the hardware and allocating system resources in case of conflicts. As a result, an operating system is the program that controls all the other parts of a computer system – both the hardware and the software.

2.4.1 Windows 2000 Professional

(http://www.microsoft.com)

Windows 2000 Professional is the windows operating system for business desktop and laptop systems. It is used to run software applications, connect to the Internet and Intranet sites, and access files, printers and network resources. Windows 2000 professional aimed at individuals and businesses of all sizes. Windows 2000 Professional gives business users more flexibility. The integrated web capabilities let users connect to the Internet from anywhere, anytime – giving user company access to host of flexible, cost effective communication options. It includes security and mobile use enhancements where broad peripheral and mobile computer support is make. The advantages of Windows 2000 Professional are as follows:

- Rely on user PC to be up and running with enterprise level quality.
- Combine the ease of Windows 98 with the manageability, reliability, and security of Windows NT at speed of 30 percent faster than Windows 98.
- New peripheral support and features that extend notebook capabilities.

2.4.2 Windows 2000 Server

(http://www.microsoft.com)

This is the most economical choice. The Windows 2000 Server operating system enable users to increase the value of their existing investments while lowering overall computing costs. The Windows 2000 Server operating system aimed at small-to-medium size businesses. Specifically, Windows 2000 Server is easier to deploy, configure and use. This Operating system is provided with centralized, customizable management services to reduce TCO (Total Cost of Ownership). Further, the management services can work with existing management solutions and mixed-platforms distributed networks. Besides, it can function as a web server and/or a workgroup (or branch office) server. The advantages of Windows 2000 Server are as follows.

- User are provided with services that help them to built deploy servers more quickly. This will reduce the time required to create new websites, create virtual directories, manage security settings and manage security certificates.
- Group policy, based on the Active Directory helps admin to control user accessibility to desktop settings and applications by group rather than by individual user and computer. Windows Management Instrumentation (WMI) provides unified access.
- Windows Script Host (WSH) allows users to automate and integrate common tasks using a variety of scripting environments. Microsoft Management Console (MMC) gives user a common user interface presentation tool where user can integrate all the necessary windows-based and web-based administration components needed to fulfill a specific task.

2.4.3 Windows NT

(http://www.microsoft.com)

Window NT (which may originally have stood for "New Technology," although Microsoft doesn't say) is Microsoft's popular operating system for all types of networks ranging from workgroups to enterprise-level installations. Microsoft's Windows NT, which provides secure, scalable, reliable 32-bit operating system platform, is a component of Microsoft BackOffice and provides the foundation upon which all other BackOffice applications run.

Windows NT is actually two products: Microsoft NT Workstation and Microsoft NT Server. The Workstation is designed for users, especially business users, who need faster performance and a system a little more fail-safe than Windows 95 and Windows 98. The Server is designed for business machines that need to provide services for network-attached computers. The Server is required, together with an Internet server such as Microsoft's Internet Information Server (IIS), for a Windows system that plans to serve Web pages. Windows NT networking and security features:

- An easy-to-use Microsoft Windows 95-style desktop interface.
- Integrated Internet services and tools, including Internet Information Server.
- Administration tools such as System Policy Editor and user profiles allow system administrators to manage and maintain users' desktops in a consistent manor
- Support both of the server and client sides Domain Name System (DNS) protocol.
- Support for the Distributed Component Object Model (DCOM).
- Improvements to the core operating system services and components.

2.4.4 UNIX

An operating system developed at AT&T Bell Laboratories in 1669 and widely used today in the enterprise-networking environments. UNIX was used on mini computers and workstations in the academic community because of it openness. UNIX is a multi-user, multi-tasking operating system and it was designed as a time-sharing environment, in contrast to the batch environment of most computing system of the time. It is suitable for very large, massively symmetric multiprocessing systems, systems with greater than eight CPUs. Using UNIX, multiple users could simultaneously access the system and run their programs, communicating interactively with the system using remote terminal. Users could thus collaborate in real time on computing projects by sharing files and resources. (UNIX, 2001)

Table 2.2 UNIX Pros and cons

Pros:	*Cons:*
▪ Already exists for over 30 years ▪ available for almost any hardware platform ▪ Variety of utility programs that called tool. ▪ made to keep on running ▪ secure and versatile ▪ scalable ▪ a powerful and mature OS	▪ on proprietary systems: bundling and system specific implementation of commands/packages ▪ not user friendly, confusing for beginners ▪ proprietary hardware is expensive

2.4.5 Linux

Linux is a UNIX-like operating system that was designed to provide personal computer users a free or very low-cost operating system comparable to traditional and usually more expensive UNIX systems. It runs on a wide variety of hardware, ranging from 386's/ 486's/ Pentium/ Pentium II's to more exotic hardware such as Digital Alpha computers, PowerPCs and Silicon Graphics workstations. Probably the most unique characteristic of Linux is that it is freely distributable. This means that the source code for the kernel and most software cannot be withheld. (Kok, C. E., 2001)

Table 2.3 LUNIX Pros and cons

Pros:	*Cons:*
▪ free ▪ portable to any hardware platform ▪ made to keep on running ▪ secure and versatile ▪ scalable ▪ very short debug-time	▪ many different distributions ▪ not user-friendly, confusing for beginners ▪ unsafe

- Linux is a complete operating system that is:
    - Stable – the crash of an application is much less likely to bring down the operating system under Linux.
    - Reliable – Linux servers are often up for hundreds of days compared with the regular reboots required with a Windows system.
    - Extremely powerful as a very efficient and fast-performing system.

- Comes with a complete development environment, including C, C++, FORTRAN compilers, toolkits such as Qt and scripting language such as Perl and Awk. C compilers for Windows alone would set a user back hundreds of dollars.
- Excellent networking facilities, allowing user to share CPUs, share modems and etc; all of which are not included or available with Windows 95.
- The ideal environment to run servers such as a web server (e.g. Apache), or a FTP server.
- An operating system that is easily upgradeable. After any length of time, a typical installation of Windows and software will get into a complete mess. Often the only way to clear out all the debris is to re-format the hard disk and start again. Linux, however, is much better for maintaining the system.
- Support multiple processors and standards.
- True multi-tasking; the ability to run more than one program at the same time.

2.5 Reviewing Database Management System

DBMS known as Database Management System is a set of computer programs that controls the creation, maintenance, and utilization of the databases. Database is the core or the backbone for an application. In this section, some of the popular DBMS are reviewed at below.

2.5.1 Microsoft Access 2000

(*http://www.microsoft.com/office/access/default.asp*)

Microsoft Access 2000 is a relational database application that allows desktop users to create and manipulate megabytes of data, with 50MB being the average size database. Multi-user access to the same database is accomplished via file-server architecture, rather than client/server architecture. Since its first introduction in 1992 as Microsoft Access, the product has become a leader in the desktop database category both for its power and ease of use. This popularity continues in late 1995 with the introduction of Access 95, which was the first 32-bit RDBMS. Access 97, which followed in January 1997, combined the product's strong database capabilities with the web technologies, offering users the ability to share static and dynamic data via the corporate Internet. Access 2000, introduced in April 1999, provides stronger integration with other Offices applications as well as with enterprise-level database.

Strengths:

- Simplifies the skill set needed to create simple databases for desktop users.
- The improved interface offers more consistency with other applications.
- Adds Data Access Pages, which are HTML pages that let user interact with data over the web from any location.

- Maintain live links to the database.
- Its support of OLE DB, can act as a front end to high end database engines
- More scalable as business needs grow.

Limitations:

- A desktop relational database management system intended for individuals or small workgroups only.
- The Jet and MSDE engines available with Access 2000 support only 2GB per database.

2.5.2 Oracle

Oracle is a powerful database management system. It allows multi–user in a single database. Oracle is also a software package that specializing in managing a single, shared set of information among many concurrent users. Oracle performed extensive transactional control where it allows thousands of users to access the same data at the same period without any problems. Besides that, it is capable to handle huge amounts of data. Oracle maybe very complex and difficult to administer, but its complexity surely increase the system stability and makes it become very robust.

Oracle is also an excellent database server choice for client/server computing. Oracle can be plugged into a client /server equation. It works to efficiently manage database information, among the multiple clients requesting and sending data in the network.

Oracle supports all major operating system for both clients and servers, including MSPos, UnixWare, Netware, OS and SQL*net. It also support all major network communication protocols, including TCP/IP, SDX/IPX, named pipes and DEC-Net. Oracle can be the link that joins the many data stores and networks

throughout the heterogeneous computing system prevalent in most corporations. (Yee, C. P., 2001)

Oracle 9 I database

Oracle 9 I database provides transparent applications scalability by sharing cluster – wide cache for coordinated data access. Oracle 9 I database also includes business intelligence capabilities, and it also provides programmatic access, centralized management and multi-channel delivery of internet services. One advance feature of the Oracle 9 I database is the capability for recovering from disaster situations. Besides that, this database is designed with integrated manageability that creates a complete business view of all components powering e-business processing. Oracle 9 I also provide multiple layers of security where it can prevent unauthorized user from access to the database. Oracle 9 I is compatible with UNIX and Window NT. (Yee, C. P., 2001)

2.5.3 Microsoft SQL Server 7.0

(http://www.microsoft.com/msdn)

Microsoft Corporation releases SOL Server for relational database management system (RDBMS) on Windows platforms. Each platform has its own version of the product, which is 100 percent code-compatible with others.

Strengths:

- Integration with Microsoft Exchange Server provides Internet and Intranet collaboration and messaging-supporting SQL Server-initiated trigger – and other stored-procedure-based messaging and replication of Exchange public folders.
- Distributed Heterogeneous Query can be used to access data from multiple sources at one time; tables within multiple, distributed heterogeneous data sources can be referenced as if they were local tables within SQL Server. SQL Server packages all the data into a single result set and returns the answer to the client PC.
- Distributed Merge Replication lets a remote system periodically connect to a central server and synchronize table changes, an important feature for its use in mobile computing environments.
- Dynamic Resource Allocation lets server, disk and memory usage scale up and down to meet changing database demands.

Limitations:

- SQL Server is limited to Windows platforms, which may limit its scalability in many environments.

- When the database detects corruption or a hardware problem that cannot be repaired during runtime, it immediately shut down, taking the engine offline. This can significantly affect the reliability and availability the system.
- Not all features Available on all platforms. Some features of the SQL Server 7.0 Enterprise Manager are not available on Windows 95/98 implementation. This is due to the limitations inherent in desktop OS.
- SQL Server 7.0's MMC-based Enterprise Manager cannot connect to SQL Server 6.x databases because of SQL-MDO incompatibilities. Thus, users must run two different sets of database management tools until conversion is complete.
- It requires Internet Explorer 4.0 to function. This requirement can take up server disk space with superfluous code.

2.5.4 Microsoft SQL 2000

Microsoft SQL Server is a relational database management and analysis system for e-commerce, line-of-business, and data warehousing solutions. SQL Server 2000, the latest version, is the most robust database for the Windows family, RDBMs is the choice for a broad spectrum of corporate customers and independent software vendors (ISVs) building business application. Customers' needs and requirements have driven significant product innovation in ease of use, reliability and scalability and data warehousing. SQL Server 2000 runs on Window NT 4.0 or Windows 2000. By using SQL Server 2000, modern application can be developed that separate the client application and the database service. SQL Server Transact-SQL support ANSI-92 standard and provides extension to the SQL language. It also includes support for XML and HTTP;

performance and availability features to partition load and ensure uptime and advanced management and tuning functionality to automate routine tasks and lower total cost of ownership. [2.5.4] Microsoft SQL Server supports a set of features that result in the following benefits: - (Fah, L. L., 2002)

- Fully web enabled
    - Web enabled analysis
    - Web access to data
    - Integration with .Net Enterprise Servers
    - Full text search\ensure the application are secure of any networked environment
- High Scalable and Reliable
    - High Availability
    - Scalability
    - Partition your workload among multiple servers for additional \scalability.
    - Gain performance from you existing hardware by storing query results
- Faster time to market
    - Simplified Database Administration
    - Improved Developer Productivity
    - Analysis OLSAP Service
    - OLAP Flexibility

2.6 Reviewing Programming Language

2.6.1 Visual Basic .Net

(http://www.microsoft.com/net/basics/framework.asp)

Visual Basic 4 was one of the remarkable software programming application which released by Microsoft. It had brought the Object Oriented (OO) programming to reality. However, only few OO features were available at Visual Basic 4. Obviously, it was lacking of inheritance which was one of the key defining for any Object Oriented languages. Furthermore, secondary features such as overriding, overloading, and constructor were not included in VB.

Since Visual Basic was not a fully Object Oriented language, subsequently Microsoft released Visual Basic .NET. It is a full object oriented language. A programming language only can be considered as Object Oriented language when it supports four main features: Abstraction, Polymorphism, Encapsulation and Inheritance.

With Visual Basic 4, user is provided with the capability of object and COM component creation. This is contrast to Visual Basic .NET which has the capability of creating object and the defining how objects worked. In .NET, component-oriented features such as component-level scooping via Friend keyword and implements interface with Implements keyword are retained. Component level scooping means that users are capable to develop classes or methods which can be shared among others code within a component. On the other hand, the latter allows classes to have several identities.

In addition to these existing features, Visual Basic .NET provides user with inheritance features which made Visual Basic.NET much more remarkable.

Below are some of Visual Basic .NET significant advantages:

- It helps software developers to build robust Windows-based applications. It is easier and faster to build an interface for applications with the rich user interface features.
- It resolves versioning problems which hunted the previous VB version. Issues regarding component overwrites and Component Object Model registration are eliminated.
- Faster deployment with X-COPY features

2.6.2 Java 2 Platform Enterprise Edition

(http://java.sun.com)

Java 2 Platform Enterprise Edition (J2EE) is platform neutral, developer who adopted it as system platform is restricted to Java language. J2EE technology and its component based model have simplified the enterprise development and deployment. Besides of that, web services are also provided by J2EE in terms of framework and environment to enable development of interoperable, robust and secure enterprise applications.

The diagram below illustrates the relationship between the elements of J2EE Platform:

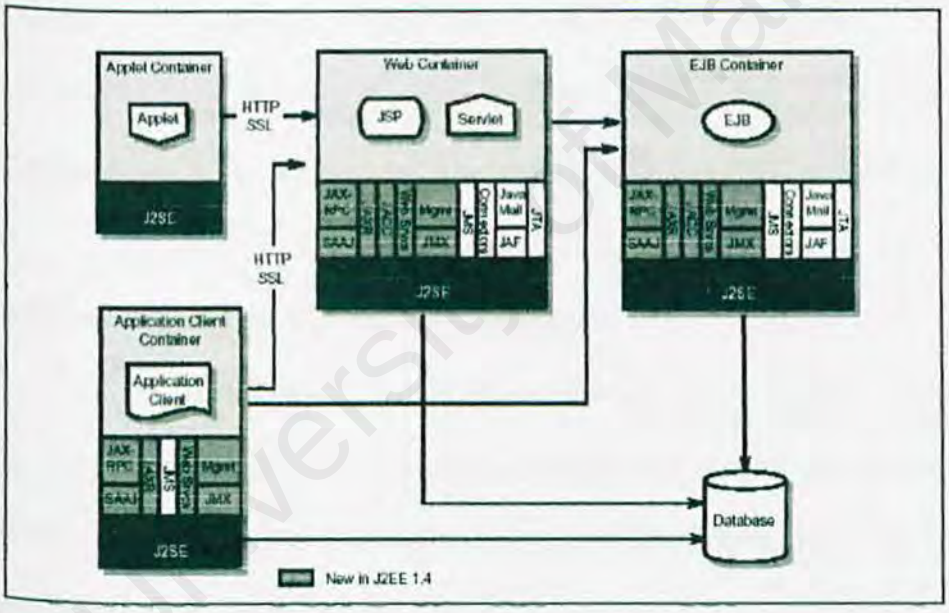

Figure 2.3 J2EE Architecture Diagram (Java, 2003)

Containers

J2EE products have to contain and support the four application components that defined by the J2EE runtime environment:

- Application clients – Graphic User Interface (GUI) programs that are typically execute on the client-side's desktop computer.
- Applets – A GUI components that could be execute by the web browser. Besides of web browser, an application or devices can execute an applet as long as it compatible with Java applet programming model.
- Servlets – This application component consist of JSP pages, web event listeners, and filters that execute by the web container. This container may respond to the web client if there are any HTTP requests from them.
- Enterprise JavaBeans (EJB) – A components that execute in a transactions environment. EJB always contain the business logic of a J2EE application.

J2EE Server Support for Application Components

The J2EE servers have offered a set activities to support the development of application components, including execution, management, support, and deployment. There are three categories of application components which have their own way of depending on a J2EE server:

- Web components and EJB components that are executed, managed, and deployed on a J2EE server.
- HTML pages and applets are executed on client-side's computer desktop, it actually deployed and managed on a J2EE server.

- This category of components are not completely managed and deployed on a J2EE server. An example of this category of components is Application Clients.

Interoperability

Most of the Application Programming Interface (API) within the J2SE has provided the interoperability with other components that are not a part of the J2EE platform. The example for this external component is external web or CORBA services. (Java 2 Platform Enterprise Edition Specification, v1.4, 2003)

J2EE Servers

A J2EE Product Provider could implements and executes server-side functionalities by using a transaction processing infrastructure together with Java 2 Platform, Standard Edition (J2SE) technology. In fact, J2EE client functionalities are also adopting J2SE technology.

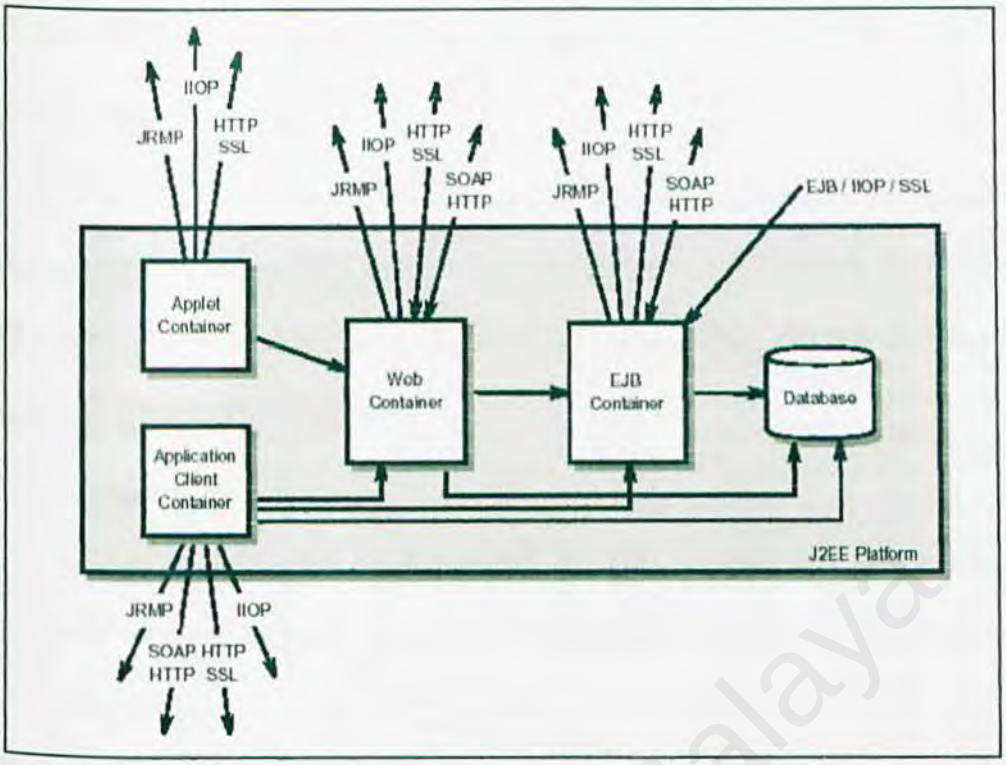

Figure 2.4 J2EE Platforms (Java, 2003)

The figure above has illustrated the interoperability facilities within the J2EE platform. (The directions of arrows have shown the client/server relationships of the components.)

components are divided into two key groups. The first one is designed specifically for regular data access and the other one is for SQL Server. Connection, Command, Data Reader and Data Adapter are the key managed-provider components. [2.4]

2.7.2 ODBC

(http://msdn.microsoft.com/library/odbc/htm.asp)

Open Database Connectivity (ODBC) is a commonly used application-programming interface (API) for database access. It uses Structured Query Language (SQL) as its database access language.

ODBC is designed for greatest interoperability which; it enables a single application to access different database management systems (DBMSs) with the same source code. Database applications call functions in the ODBC interface that implemented in the database-specific modules called drivers. It is used to isolate applications from database-specific calls in the same way which; printer drivers isolate word processing programs from printer-specific commands. Due to drivers are loaded at run time, it is not necessary to recompile or link again the application because a user is required to add a new driver to access a new DBMS.

Nowadays, many misconceptions about ODBC exist in the IT world. For the end user, it is an icon in the Microsoft Windows Control Panel. However, for the application programmer, it is a library containing data access routines. For many others, it is the answer to all database access problems ever expected.

ODBC is a specification for a database Application Programming Language (API). It is independent of any one DBMS or operating system. The ODBC API is

based on the CLI specifications from X/Open and ISO/IEC. ODBC 3.*x* totally implements these earlier versions of ODBC that based on preliminary versions of these specifications but did not completely implement them. It also adds features that normally required by developers of screen-based database applications, such as scrollable cursors.

Developers of DBMS specific drivers will apply the functions in the ODBC API. Applications call the functions in these drivers to access data in a DBMS-independent method. A Driver Manager will responsible to manage communication between applications and drivers. Even though Microsoft provides a Driver Manager for computers by using Microsoft Windows NT Server/Windows 2000 Server, Microsoft Windows NT Workstation/Windows 2000 Professional, and Microsoft Windows® 95/98, has come out with a number of ODBC drivers, and calls ODBC functions from some of its applications, anybody is able to write ODBC applications and drivers. In fact, other companies produce the vast majority of ODBC applications and drivers available for computers running Windows operating systems. In addition, ODBC drivers and applications are also available on the Macintosh and a variety of UNIX platforms.

2.7.3 OLEDB

Object Linking and Embedding Database (OLE DB) is the next generation of data access technology. It is a high performance database access method developed by Microsoft. Microsoft SQL Server has using the advantages of OLE DB within its own components. Besides of that, Microsoft has also includes an OLE DB provider for Oracle 7.3 with SQL Server. (Kok, C. E., 2001)

OLE DB is a strategic system-level programming interface that used to control and manages the data across the organisation. OLE DB is an open specification designed to build on the features of Open Database Connectivity (ODBC). The functionality of ODBC is to gain the accessibility of relational database, and OLE DB not only designed to access relational database, but it also provides the functionality to access the non-relational information sources, such as hierarchy databases and mainframe ISAM/VSAM, e-mail, custom business objects, text, graphical and geographical data, and file system stores. (Kok, C. E., 2001)

OLE DB has defined a collection of COM interfaces that encapsulate various database management system services and allows the creation of software components that implement the similar services. There are three OLE DB components, including data consumer, service components (processing and transport data), data providers (that contain and expose data). (Kok, C. E., 2001)

The main purpose of OLE DB interfaces is to allows the smoothly of integration components so that other OLE DB component vendors have the capability to develop a standard and high quality components rapidly. OLE DB also offered a bridge to ODBC to enable the continued compatible with a wide range of ODBC relational database drivers' availability. (Freeze, W. S., 2000)

2.8 Reviewing Component Technologies

2.8.1 COM

(www.msdn.microsoft.com)

COM is a component software architecture that allows applications and system to be built from components that have been developed by various software vendors. [2.5] COM defines an Application Programming Interface (API) that allows diverse of components to interact with each other.

Components need to have similar binary standard in order to interact. COM defines a binary structure for the interface between client application and component object. This binary standard provides interoperability between components written in any programming languages that support call function via pointers.

In COM, application interacts with components through interfaces. Interfaces define as a collection of functions provide a small and useful operation. Interface is differs from a class but a class can be instantiated to form a component object. An interface does not carry any implementations and inability to instantiated by itself.

Component objects can implement multiple interfaces, if the class has more than a set of services to provide. For example, a class might provide client to access data and performing calculations on the data. Each of these services requires different interfaces to be implemented.

Every interface has its own interface identifier. Thus, it will eliminate any chance of collision or versioning issue would occur. A new identifier is needed to define when a new component is defined.

The advantages of using interface in COM are listed below:

- Fast and simple object interaction.
- Interfaces are reusable.
- The ability for functionality in applications (clients or servers of objects) to evolve over time.
- Supports local and remote transparency.
- Programming language independence.

Figure below illustrates a component object support three different interfaces

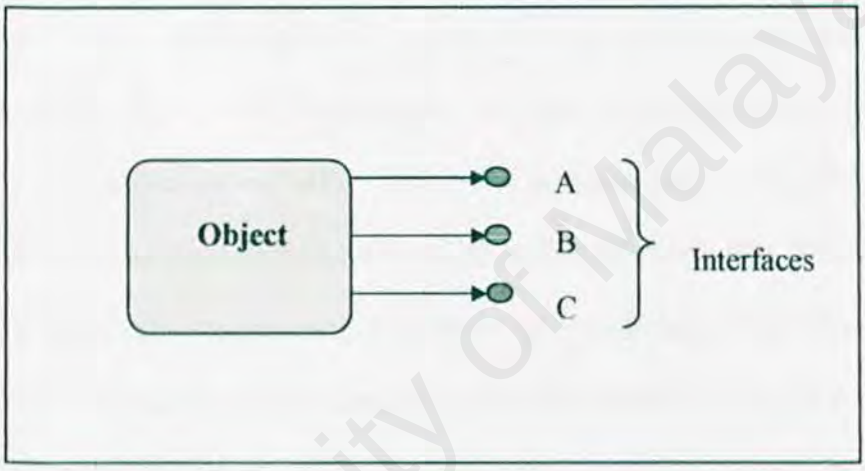

Figure 2.5 Component object with three interfaces (Williams, S. *et al*,1994)

2.8.2 COM+

(www.msdn.microsoft.com)

COM is a software architecture which stands for Component Object Model. COM enables formation of the foundation for higher-level software services. An OLE service is one example services. Compound documents, inter application scripting, data transfer, custom controls and other software interactions are commonly needed system functionality which OLE services can provide. (Microsoft)

COM+ builds on COM's integrated services and features, making it easier for developers to create and use software components in any language, using any tool. COM+ is defined as further evolution of the Microsoft Component Object Model and Microsoft Transaction Server (MTS). COM+ was designed dedicated to provide server-side component services and to fix some of DCOM's deficiencies. COM+ able to increase the system applications scalability by having thread pooling, object pooling and just-in-time object activation. Applications currently using COM technology will work in the COM+ environment. Besides, it also contributes in protecting the integrity of the data by providing transaction support.

2.8.3 DCOM

The Distributed Component Object Model (DCOM) is a protocol that enables software components to communicate directly over a network in a reliable, secure, and efficient manner. Previously called "Network OLE," DCOM is designed for use across multiple network transports, including Internet protocols such as HTTP. DCOM is based on the Open Software Foundation's DCE-RPC spec and will work

with both Java applets and ActiveX® components through its use of the Component Object Model (COM).

Distributed Component Object Model (DCOM) is a superset of the Component Object Model (COM) that allows the distribution of objects over the local area and wide area network. COM technology only allows the developer to create object, but DCOM's capability expand more than COM, DCOM allows the developer to create object across the network. (Grimes, R., 1997)

While discussing about the distribution an application across a network, there are several issue need to be considered, including object locater, communication, state persistence, reliability and availability. Each of this issue is being describe below:

- **Objects Locator** – A distribution system should have a mechanism that able to indicate the location of an activated object. The mission of the object locator should ensure the developers are able to find the particular object.
- **Communication** – Once the object has been located and activated, there are three areas of communication need to be consider:
    - **Data Typing** – When talking about the issue of distribution, the flexibility of data typing is very important and needed to be taking into account. A well design distributed system should allow the developer to declare the contracts in between the objects and the clients by using the special data types, these contracts are often called *interfaces*.
    - **Data Representation** – Since there are a number of objects distributed across several machines, there would be a possibility of applying the same method and data representation by the distributed system. So it is critical to have a definition of data types for the marshaling of data. DCOM is based

on the mechanism of Remote Procedure Calls (RPC), and RPC has defined Network Data Representation (NDR) to enable the data to be transmitted in terms of independent of actual data.

- **Synchronous and Asynchronous Communication** – Asynchronous communication are invoked when a user makes a request to a particular objects, while the process is being blocked until it was replied. Sometimes it is necessary to have a callback mechanism to indicate the client once the request has been processed.
- **State Persistence** – A client could gain the authority to access an object by create the objects, use its services, and then release it. The system (object server) must have the ability to associate with the client regarding to the issue of object connection, so that the server would know when to release the object's resources once the object has been finish used.
- **Security** – The object server administrator should aware who has used or attempted to gain accesses the objects. The username and password authentication should be implemented in order to have a secure distribution mechanism. The users should gain the security authority from the administrator before they could activate any object's services. Thus, DCOM is designed together with security build in.
- **Reliability and Availability** – A distributed object have the ability to offer transparency for the client, one of its purposes is to ensure the object connections is reliable. Reliability of DCOM could be achieved by using the Microsoft's Transaction Server (MTS).

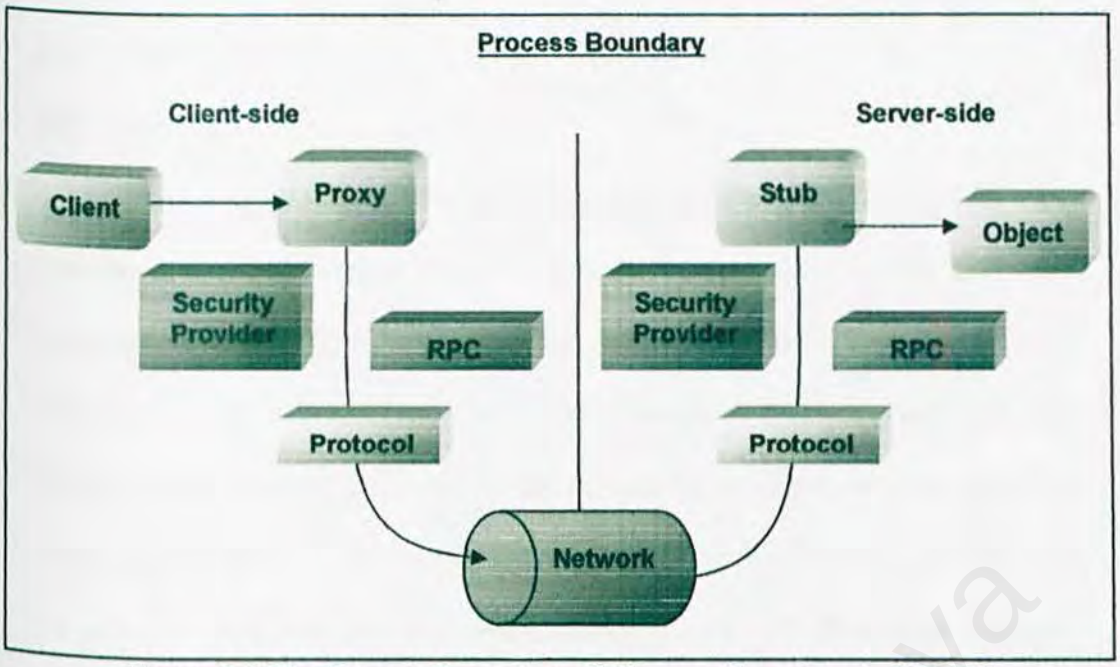

Figure 2.6 The Remote Procedure Calls over the network, DCOM

(Grimes, R., 1997)

2.8.4 CORBA

(http://www.omg.org)

The Common Object Request Broker Architecture (CORBA) is an emerging open distributed object-computing infrastructure being standardised by the Object Management Group (OMG). The OMG is a consortium of companies, including major players like Sun and IBM. The CORBA standards define a generic machine-independent approach to distributed object computing. A number of implementations of this standard by different vendors have been developed. CORBA implementations are available for UNIX and Microsoft operating systems. CORBA automates many common network-programming tasks such as object registration, activation and operation dispatching.

2.9 Reviewing Client/Server Architecture

2.9.1 Client / server architecture

Recently, client-server architecture has been introduced in order to replace the file server architecture. The users' queries could be handled by using a relational database management system (DBMS). Basically, client-server architecture has provided a query response to reduce the network traffic. It also improves multi-user updating through a Graphical User Interface (GUI) front end to a sharing database. The communication in between client and server is being practiced by using the Remote Procedure Calls (RPCs) and Standard Query Language (SQL) statements in a client-server architecture. (Schussel 96, Edelstein 94)

2.9.2 Two tier architectures.

Developed in the 1980s from the file server software architecture design, the Two-Tier architecture is where a client talks directly to a server, with no intervening server. Usually, it is for the small and medium environment.

In most of the two tier designs, most of the application portion of processing is in the client environment. The database management server usually provides the portion of the processing related to accessing data. However, there is a situation where it splits the processing management between client and server.

The approach that prototype an application in a small, two-tier environment, and then scale up by simply adding more users to the server will usually result in an ineffective system, because of the server becomes overwhelmed.

The two tier architecture is suitable for following circumstances:

- when the number of users is expect to be less than 100
- for non-real-time information processing in non-complex systems that requires minimal operator intervention

2.9.3 Three-Tier Architecture

(http://www.sei.cmu.edu/str/descriptions/threetier.html)

To overcome the limitations of the two tier architecture, the three tier architecture (a.k.a. three layer architectures) emerged in the 1990s. The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides process management where business logic and rules are executed.

The three tier architecture is used when an effective distributed client/server design is needed to provide functions such as queuing, application execution, and database staging, while hiding the complexity of distributed processing from the user. Three layer architectures is a popular choice for Internet applications and net-centric information systems.

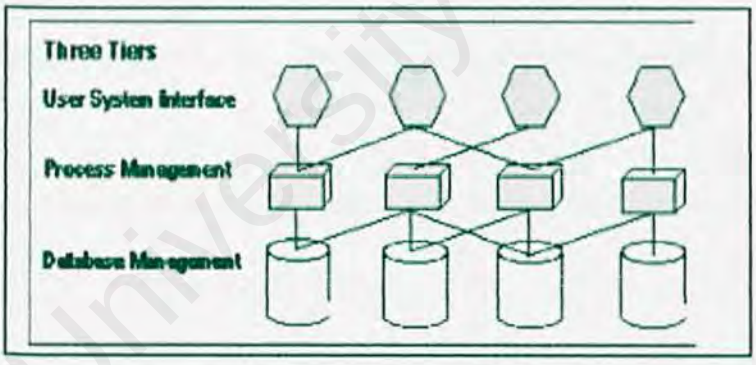

Figure 2.7 Three Tier Architecture

The middle tier provides process management services (such as process development, process enactment, process monitoring, and process resourcing) that are shared by multiple applications. For example, there are a number of middle tier products that link a database system to a Web server. This allows users to request

data from the database using forms displayed on a Web browser, and it enables the Web server to return dynamic Web pages based on the user's requests and profile

The advantages of three tier architecture are listed below:

- Allowing application to evolve easily. This is because any changes on presentation logic or business logic are highly independent.
- Increasing maintainability - altering business logic might not need to recode the GUI. Updating server is required.
- Business logic can be shared among applications with different User Interface.
- It will minimise the network bottleneck because application layer only transmit the required data or information.
- Client can access data easily and faster because client is insulated from database. Client does not need to know where the data resides in the database server.
- Besides business logic, database connection can be pooled and shared among clients.
- Application layer can be created with third or fourth generation languages such as Java, COBOL, C, and so on.

2.10 Case Study:

2.10 What is Inventory management?

Inventory is defined as a detailed list of all the terms in a place. More generally, inventory could be regarded as a resource that has economic value. An inventory is made up of one or more items where each item is a unique supply item, raw material, purchase of manufactured part, assembly, or final product.

Inventory management involves the control of the assents that are produced to be sold in the normal course of the firm's operation.

An Inventory Management must provide necessary functions for updating and maintaining raw materials and finished good inventory quantities and costs. Inventory issues reduce inventory quantities on hand, which are continually replenished through the processing of purchased into inventory.

2.10.1 What is shipping and receiving?

Shipping/Receiving monitor incoming and outgoing items, whether it is shipments against an order, sending subcontract parts to a vendor, receiving raw material from a purchase order to a job or into inventory, or filling an order from stock. All activity relating to shipments and receives can be performed and tracked through this module.

2.10.2 What is Purchasing Management ?

Purchasing Management handles Purchase Order writing and the tracking of supplier performance. Purchase Order processing updates detailed purchase history files, which provides continual reference to aid in making purchasing decisions. Purchasing Management provides a tool for a purchasing agent to request quotes for raw materials.

2.10.3 Case Study: Material Management Module

2.10.3.1 MARIAN Material Management System

(www.intergraph.com/ppo)

System Overview

We launched a research upon MARIAN material management system in order to gather more information about the material management system which consists of the major functionalities such as **Inventory Management, Shipping/Receiving, Purchasing RFQ**(Request for Quotation)**Management, Purchasing Management** , and **Advanced Materials Management.**

MARIAN material management system is divided into several modules to fit the work flow of its system. All the module will work together to implemented the whole material management process. The modules are:

a) Bulk Material Life cycle library

b) Engineering & Procurement integration module

c) The Material Supply Chain Management Module

d) The Site Management Module

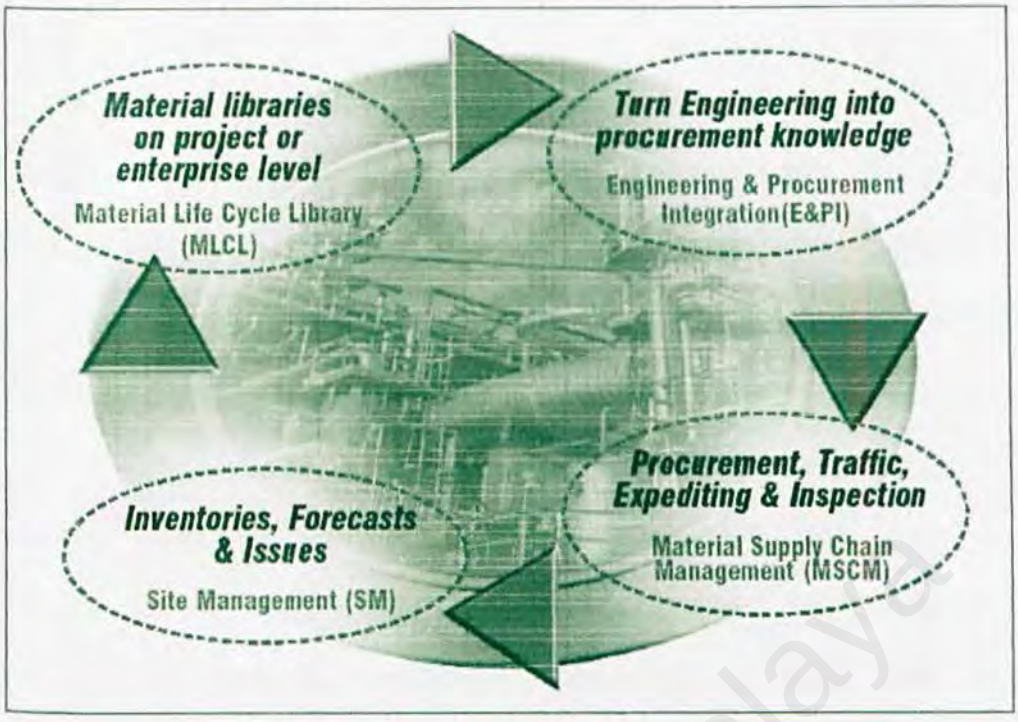

Figure 2.8 MARIAN material management module (Intergraph, 2003)

Inventory management

The Bulk Material Life Cycle Library of the MARIAN has several functionalities such as materials revision, change management and project estimating. All these functionalities are enabling at the corporate and project levels. Rapid estimate generation for new project bids can easily be done by using project estimating functionality. The Library also enables corporate (project-independent) material data standardization, especially of bulk material.

Advanced Materials Management

Engineering & Procurement Integration Module is using by MARIAN to presents the most recent status of any material item being used within the project. Besides, when duel with materials shortages, this module enables the elimination of surplus materials and delays. All engineering and design quantities are stored, combined, and change-controlled in highly flexible project structures.

Purchasing Management and Purchasing RFQ Management

The Material Supply Chain Management Module of MARIAN integrates interchanges with suppliers, manufacturers, fabricators, and freight forwarders. Decentralized processes such as supplier management, procurement, progress control, tracking and tracing, event management, supplier performance and history management can be synchronized. Same database are being used to store all procurement and relevant supply chain information. By using the same database, this enables paperless integration in purchasing management such as supplier performance, history measurement, supplier management, inquiry cycle, tabulation, purchase order, scheduling, milestone control, and tracking/tracing.

Shipping/Receiving

The Site Management Module in the MARIAN system enables balancing on-site personnel with material availability. This means all the materials can be assigned to site inventories in the shortest time. All warehouses can be control and manage efficiently by having critical processes such as automated material receiving, forecasting, reservation, and allocation routines materials implemented in each

warehouse. All the site inventory data are available when it is needed in sophisticated planning and simulation.

Advantages:

- Reducing labor hours and eliminating materials surplus and shortages can lower the total project and installed costs.
- Increases competitiveness through reduced project bidding time, compressed schedules, and reduced man-hours
- Enables sharing and execution by using correct, complete, and consistent data in global project work
- Allowing data reuse

Disadvantages:

- The automated Material management module requires accuracy of input. The technology cannot guard against human beings who enter the wrong information.
- System can not provide flexibility that only human beings can give. There are some small process may not need automated.
- User required training to enable them to use the system correctly.

2.10.3.2 Regional Hazardous Material Management System (RHMMS)

(https://www.denix.osd.mil)

System Overview

A research is carrying on RHMMS to retrieve more information about the Material Management System which consists of the major functionalities of **Inventory Management, Shipping/Receiving, Purchasing RFQ**(Request for Quotation) **Management, Purchasing Management ,** and **Advanced Materials Management.**

RHMMS are developed based on six modules to ensure HM (Hazardous Material) life-cycle management and control, are based on regional requirements. All the six modules will work together as the whole material management process. RHMMS provides the following operational modules:

- RHMMS Regional/Global Inquiry
- Inventory Tracking
- Demand Processing
- Replenishment
- Redistribution
- Financial

Inventory Management

Inventory Tracking module allow access to the information that contained in the RHMMS databases. The Inventory Tracking module contains reporting, data entry, and utility functions. Using this module, the Regional Inventory Manager can

monitor the availability of excess for each materials inventory levels and apprise customers.

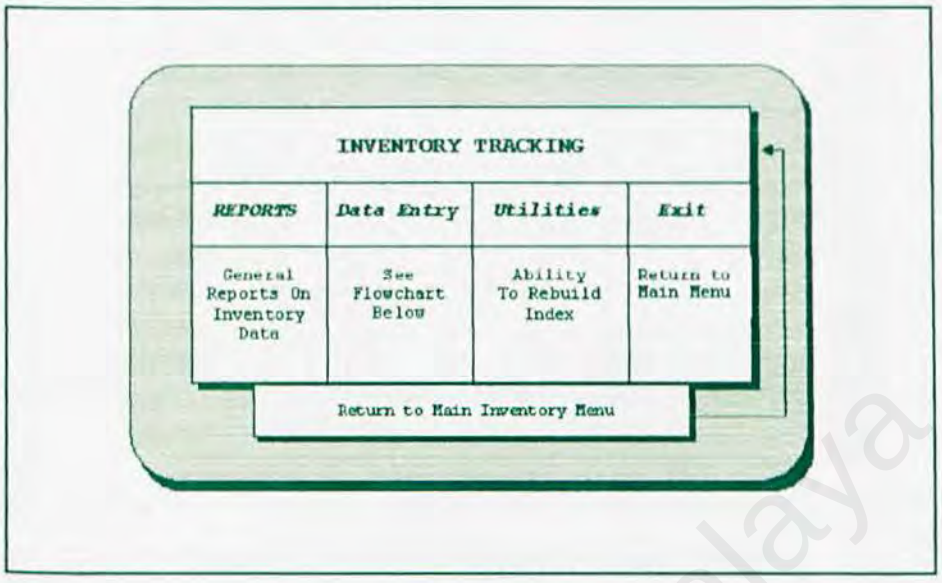

Figure 2.9 Advanced Materials Management (RHMMS, 2002)

RHMMS Regional or Global Inquiry module allows you to view the availability and location of the HM inventories from participating HAZMINCEN. Batch mode is used daily in updating data for the system. These updates are performed electronically via the LAN, WEB or modem to the main file server. The main file server will receives information daily from each HAZMINCEN via their HSMS or HICS inventory system(s). The file server then runs an update process and this module allows you to view the upload and update the inventory data.

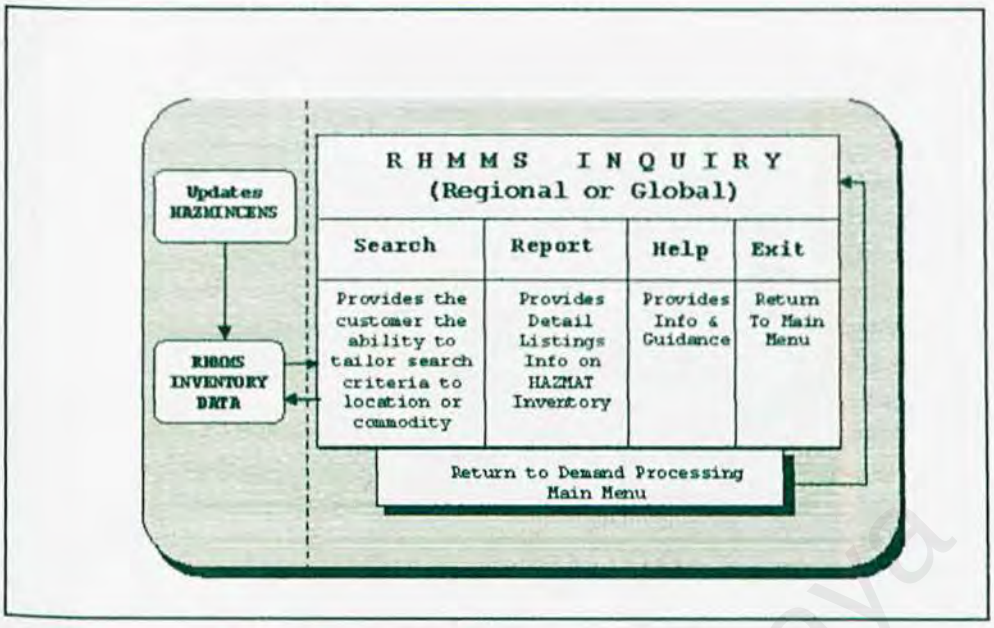

Figure 2.10 RHHMS Inquiry (RHMMS, 2002)

Shipping/Receiving

Redistribution was designed to ensure that HM (Hazardous Material) are redistributed in response to customer requests and it is been utilized by the Regional Inventory Manager. The availability of HM may not be aware due to geographic dispersion of locations and this is where redistribution needed in the inventory.

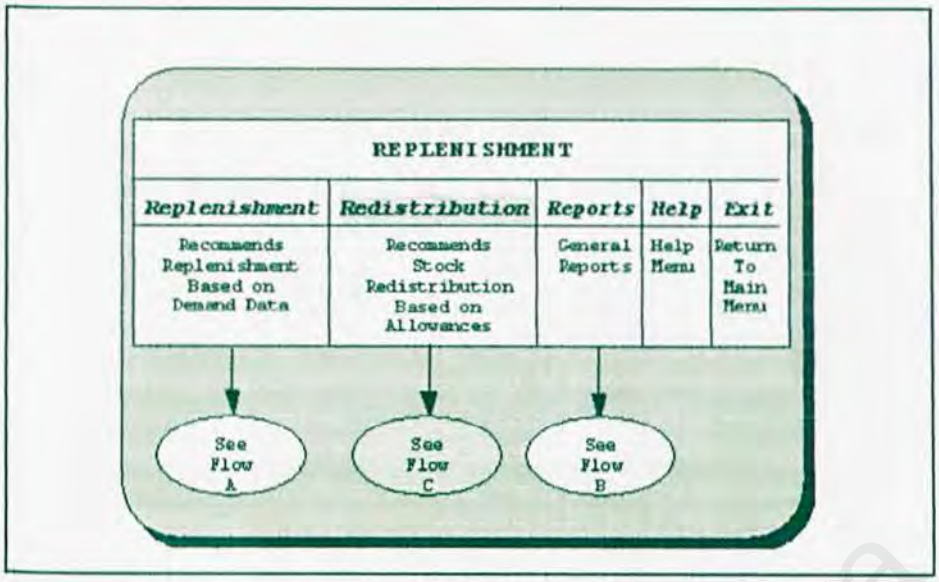

Figure 2.11 RHMMS Replenishment (RHMMS, 2002)

Purchasing Management

Demand Processing module is used to accomplish two major processes which are, first, provides the interfaces between transactions/files and RHMMS and second, processes the transactions (or demands) within RHMMS. There are two situations to consider in this module. One is RHMMS can fulfill the requirement completely, then there will be transactions that post the customer's requirements and the subsequent RHMMS issue action to the Requisition Status File (RSF). Second situation is the RHMMS could not fulfill the requisition, and then the unaltered transaction will be passed back. At this point, the requisition trail ends and it is considered completed.

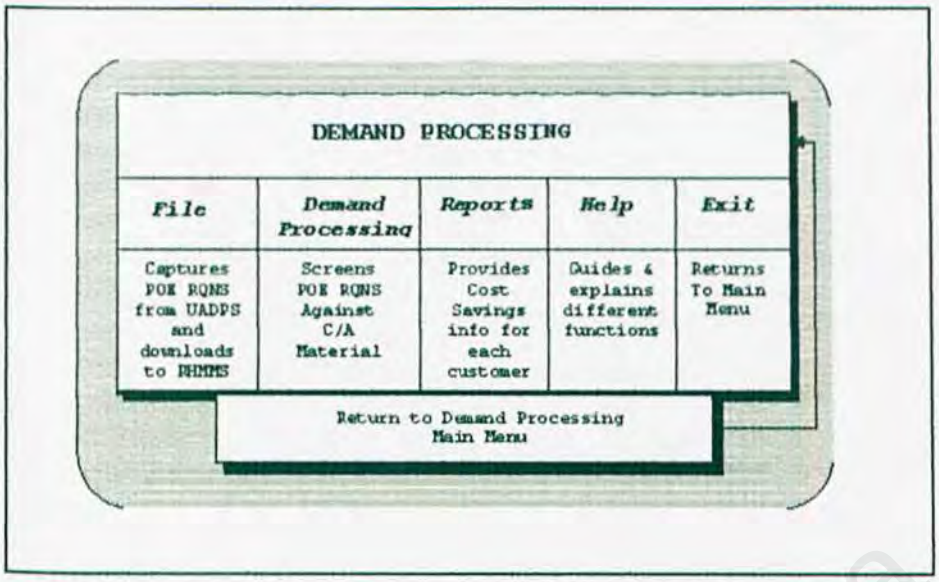

Figure 2.12 RHMMS Demand Processing (RHMMS, 2002)

Purchasing RFQ Management

Replenishment module recommends replenishment and redistribution actions. This module is based on regional demand history and inventory position for each item that stocked within the region. Factors like demand levels, realistic forecasting methodologies, operation and maintenance considerations are consider in generating replenishment requests.

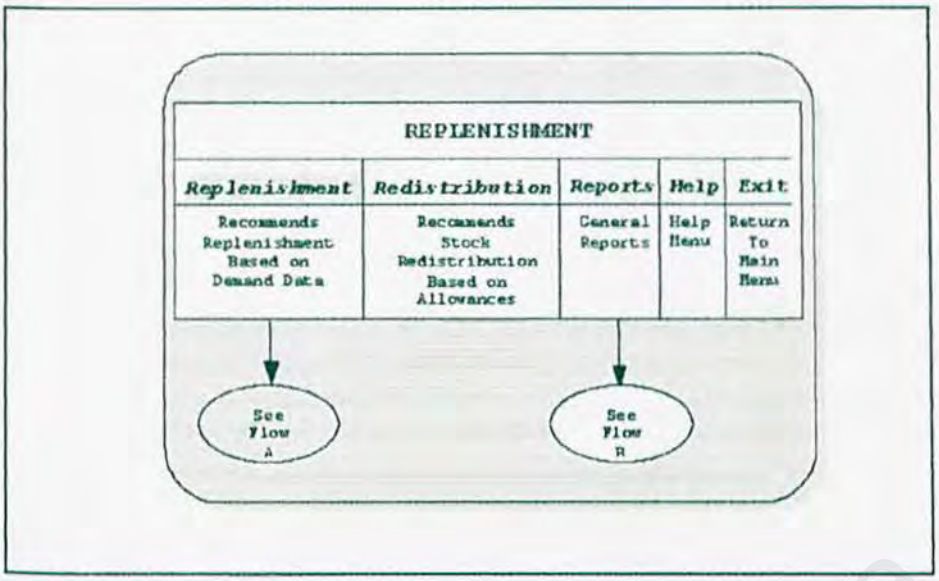

Figure 2.13 RHMMS Replenishment (RHMMS, 2002)

Advantages:

- Reduces lead times and managing resources better
- Demand forecasting methods also helped better estimation in inventory levels.

Disadvantages:

- The major drawback of the **RHMMS** was that it provided no flexibility for meeting actual customer orders. If the forecasts and subsequent base plans were not accurate (as was often the case), manufacturing would end up wasting cost to purchase unwanted material.

3.1 Development Methodology

A system development methodology is system development process that defines a set of activities, methods, best practices, deliverables, and automated tools for system developer and project manager used to develop and maintain most or all information systems and software. Developer should be careful in choosing the suitable development methodology for their process modelling because it would be a guideline for software life cycle. (Pfleeger, S. L. *et al*, 2001)

By building up a process model and its sub processes, this will help system developer to visualize and analyze the possible problem that would occurred, define the system requirement requirements, and design information system. The reasons why a suitable process modelling is important during software development are stated below:

- A process model establishes a basic foundation of understanding of the objective, scope, requirements, design, and implementation of the system. It consists of description for each development phase, activities, and method.
- A process model would provide the team member a guideline to identify the inconsistencies, redundancies, and omissions in its constituent parts and the process. The process becomes more effective if most of the problem and hiding error are noted and corrected. (Pfleeger, S. L. *et al*, 2001)
- The selected process model would reflect the goals of development. The development team are able to evaluate the candidate activities or solution for their appropriateness in address the goals. (Pfleeger, S. L. *et al*, 2001)
- The selected process should be tailored for the special situation in which it will be used. This is to help the development team understand where toleration is needed. (Pfleeger, S. L. *et al*, 2001)

There are many process models with different methods and techniques which used to conduct the life cycle of a software development. Different project with different criteria will require different process model even though most models would have similar goals and common task. Popular models used in today are: Waterfall Model, V Model, Prototyping Model, Increments and Iterative, and Spiral Model.

3.2 Methodology Consideration

In the early year of software development, customers would have to wait a long time for software system to be ready where sometimes it would take more than a year to produce the system. However, in today's competitive business environment, long delays are not allowed. Customers nowadays are looking for a system with new quality and functionalities with fast delivery of the system. Consequently, new process models were developed to fulfil the purpose of reducing cycle time.

Phased development is one way to reduce cycle time. The system would be divided into several pieces (phases), enable the user to use first release version while continue to enhance the rest release. (Shari Lawrence Pfleeger, 2001) figure 3.1 show the activities preformed by the phased development.

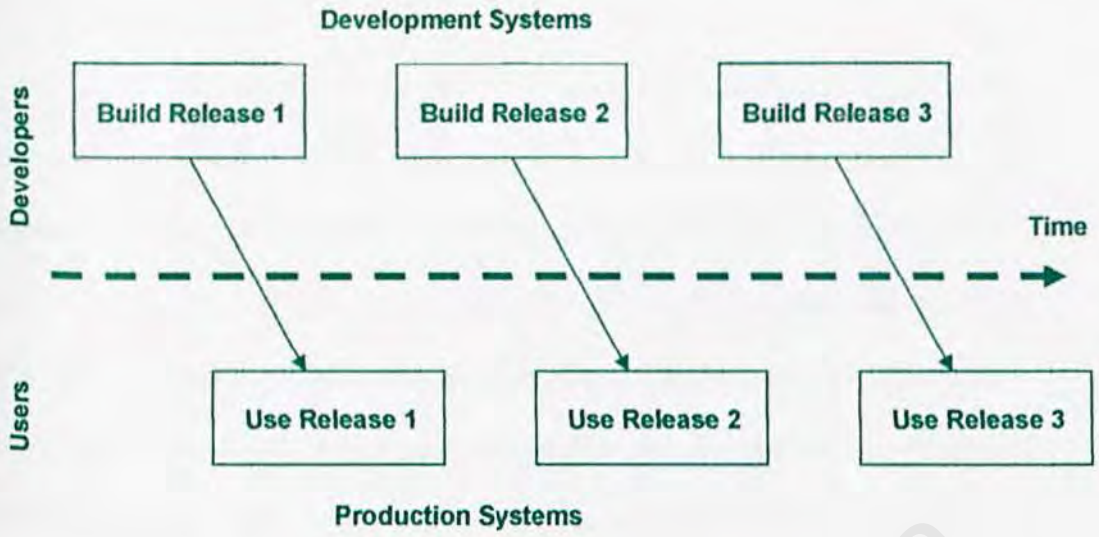

Figure 3.1 the phased development model.

The allocation of phases to increments depends on the priority of the subsystem. The highest priority subsystems are delivered first to the user. This means that the most important subsystem receives most testing and the users are likely to encounter software failures in the most important subsystem. (Sommerville, I., 2001)

Implementation of phased development offer greater flexibility and reduce the possibilities of failure to meet user requirements. The developer is able to collect valuable feedback from user to make further changes to satisfy the users in the future system. It just required minimum overhead cost if the developed phased of system does not satisfy user in terms of correctness, even through we have to re-develop the entire phase again. The two most popular approaches of phased development are **incremental development** and **iterative development**.

3.2.1 Incremental Development

Incremental development is the ability to recompile portions of a project and save the resulting object code. In incremental development, the requirements of the system are divided into subsystem by functionalities. Each of the subsystem would be developed according to the phases and each phase will be schedule and developed at different period. The deliverables for each phase is a subsystem with its specific functionality, and then by adding new functionality with next deliverable. Functional Developer always performs incremental compilations when it can, to keep build times as short as possible.

The Figure 3.2 shows the incremental development slowly build up the entire system by joining up the new deliverable with the previous one.

Figure 3.2 the incremental development model.

3.2.2 Iterative Development

Iterative development develops the entire system at the very early stage where it is no complete. Then the enhancement and modification functionalities of each subsystem are performed with next deliverable. The development team are able to demonstrate the overview system and obtain the valuable feedback. Interactive development provides the ability to execute code fragments, including definitions and redefinitions, in a running program. In fact, each iteration is a mini-*Waterfall* process with the feedback from one phase providing vital information for the idea and design of the next phase. Figure 3.3 illustrate the phases of using the iterative development model.

Figure 3.3 the iterative development model.

3.3 Implementing Incremental and Iterative Development Model

There are several reasons to combine incremental and iterative development model and use these two process models in this project:

- **Dedicated model** – Most of the process models were design to provide a guideline and a method for achieving success only under a certain set of situations, means it might only a dedicated model for certain type of project. So, it is advisable to adopt or combine different models to accommodate the new criteria when project situations are not covered beyond the boundaries of the single model.

- **Nature of the application** – It is vital for us (development team) divide the e-manufacturing framework into several phases as the frameworks considered as a large project in terms of scope and scale. Our rational of combining these models is actually based on the concept of *Command and Conquer*. Our project is emphasizing and using the technique of component-based, we could actually divide the framework into several components (phases), it is easier for us to concentrate and conquer (develop). Incremental model is suitable and adaptable with the concept above.

- **Uncertainties** – The development team do not have enough experience on develop a framework or e-manufacturing system, so plenty of uncertainties and unexpected problem would be countered during the system design and development. As a result, we choose a model like incremental and iterative which enable us to build certain component (phase) without fulfil all its requirements due to the uncertainties. Besides, we are given a flexibility to enhance its functionalities on the next release for previous version.

- **Time constraint** – The main obstacle for us is to complete the project in the limitation of time. We are only given 6 months to complete all task from system analysis, design, development, implementation to testing. Incremental and iterative model that provide fast result, require less-up front information, and offer greater flexibility is really the suitable choice for us.

One can model the general principles of iterative and incremental development by a fractal-spiral model. Figure 3.5 show the fractal-spiral model:

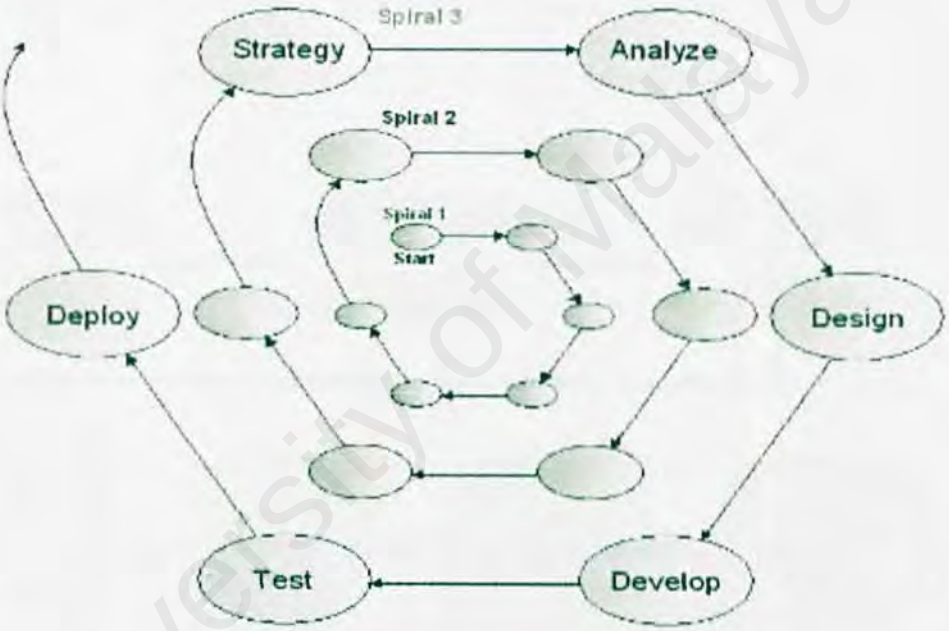

Figure 3.4 fractal–spiral models

- **Strategy** - this is primarily a management step. The results of the previous spiral are evaluated, new goals are identified, tasks are articulated and prioritized, resources are allocated, and schedules are laid out for the spiral.
- **Analyze** - focuses only on what has to be done, not how it is to be done.
- **Design** - considers how to do what was identified in the Analyze step.
- **Develop** - this step implements that design.

- **Test** - verifies conformance between the analysis, design and implementation to the satisfaction of the stake holders.
- **Deploy** - releases the results of the spiral to the client,

This emerging development methodology has inherited the advantages from both incremental and iterative model. At each phases, besides of adding new functionalities at next deliverable, the previous functionalities might being modified and enhanced based on further feedback form the user. The figure 3.4 has depicted the software development phases and processes based on the combination of incremental and iterative model.

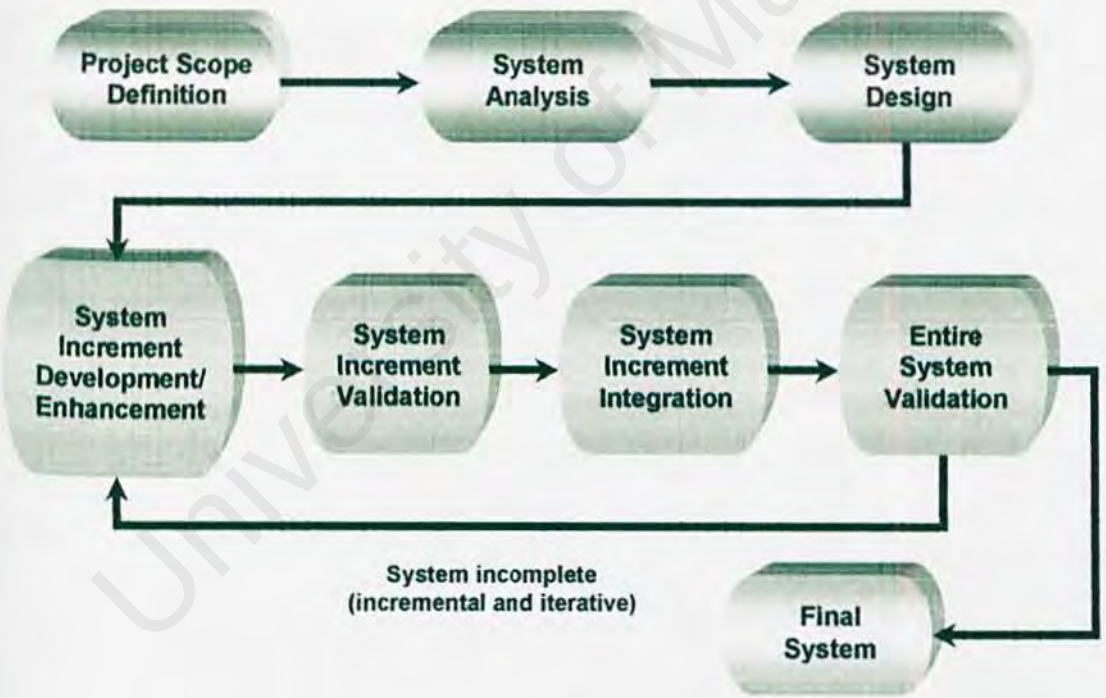

Figure 3.5 Incremental and iterative development model.

Since combination of the incremental and iterative development methodology has been applied to the project – developing a framework of e-manufacturing system, thus all the software development phases and required tasks have to according to the development model in sequence. The project started by negotiates the project scope, followed by system analysis and system design, then come to the process of incremental and iterative. There are several increments (sub-system) that have to be following the iterative process in sequence; firstly an increment would be developed, followed by increments validation, entire system integration and validation. Then, the second increment is being carried in the iterative process, following by the third and fourth increments, and so on, until final system has completed. The first increment might be modified or enhanced while the second increment is being developed; this is the unique characteristic of the combination of the incremental and iterative development methodology.

3.3.1 Preliminary Investigation Phase

This the first phase, we have to conduct a survey and planning at the beginning of the project. During preliminary investigation phases, the activities that have to be involved and took part by the development team including the determining of:

- **Project scope** – It is to negotiate the project scope and defines the objective. The project scope would act like a guideline so that development process of the system would not bias and go out of the project boundaries.
- **Project plan** –this is identifying the required task to complete the project, and then divide these responsibilities and required task equally among the team member.
- **Project participants** – Identify the project participant involve in this project development process.
- **Project schedule** – Each of the required tasks stated within the project plan were scheduled according to the task duration, priorities and dependency.
- **Constraint** –this is to identify all the constraint and limitation to the project. Most of the constraint cannot be changed.

3.3.2 Analysis Phase

System analysis is actually a problem-solving technique. it would decomposes or divides a system into several components pieces, and then study and analyse each of pieces in terms of functionalities, interactivity and effectiveness. This analysis phase consist of problem analysis, requirements analysis, and decision analysis. Each of these analyses would be discussed and explained into more detail.

Problem analysis is to conduct a study and analysis upon the existing problem that is currently being faced by most of the manufacturer. *"Don't try to fix it unless you understand It."* is the main concept stress on in problem analysis. Thus, the problem analysis enables the project team more understand about the problems that triggered the project. After conducting the problem analysis, the team member would have a visualisation of the whole project (framework of an e-manufacturing system) and the team should start consider and identify the problem to be solved. The scope of the project will be discussed here in order to be reduced or expanded.

Once the existing problems regarding to manufacturing has been identified, the requirements analysis is carried on to produce functional and non-functional requirements of the system. During this analysis, the capabilities and functionalities for the framework of e-manufacturing system are determined. A use case diagram has been adopted in capturing the system requirements. Use case diagram is one of the models specify within the Unified Modelling Language (UML), which is used to specify the behaviour of the system.

The purpose of decision analysis is to identify and analyse the candidate solution for feasibility. There are several types of solutions towards the functional and non-functional requirements in terms of technologies. There are numerous of relational database, system platform, software development tools, system

architecture, and system development methodology need to be taken into account and make decision. The key deliverable of this decision analysis is a system proposal. System proposal is a summary that specifies describes all the chosen relational database, system platform, system architecture and etc.

3.3.3 Design Phase

System design is complementary problem-solving technique to the system analysis. It reassembles those components pieces (decompose during system analysis) back into one complete system. System design will mainly focus on the technical and implementation of the project.

Since the framework of e-manufacturing system is using the reusable component based as the system architecture, the model-driven design has been adopted as a strategy in performing system design. Model-driven design fully emphasizes on the drawing of system model in the documentation for the technical and implementation of the system. UML diagrams have been used in modelling the classes (relationship), process flow (interaction), and components of the e-manufacturing system framework. All the notation and modelling for the entire framework should be in a uniform manner in order to eliminate the confusion in between the team member. The UML diagrams that used in modelling the system design are class diagram, sequence diagram and components diagram.

3.3.4 Implementation and Testing Phase

System implementation always referred as the system construction with activities of development, modification, enhancement, and integration of the system components. According to the incremental and iterative methodology, the implementation and testing phase may repeat several times.

Once the components within the e-manufacturing framework have been identified, implementation and testing phase will be carried onto each of these functionalities components in sequence. The implementation phase is about the coding and programming upon the functionalities component. These developed components need to be validated (testing phase) of its performance and functionalities (correctness). Components would be integrated with the previous developed components after they passed the testing phase. The relationship, interaction, and connectivity between these components need to be taking into account during the components integration.

The entire framework would be tested again after the integration regarding its correctness and efficiency. If there are still any remain of functionalities components have not yet be completed, the development process route would be continue through the implementation phase again. This is to perform enhancement to the existing components or to develop that particular component. Process of implementation and testing phase would not be terminated until all the required functionalities components have being developed and modified (optional) to produce the final system.

3.4 Unified Modelling Language (UML)

The system architecture of the framework of e-manufacturing system could be defined in five views by using the Unified Modelling Language (UML) as depicted in the diagram below:

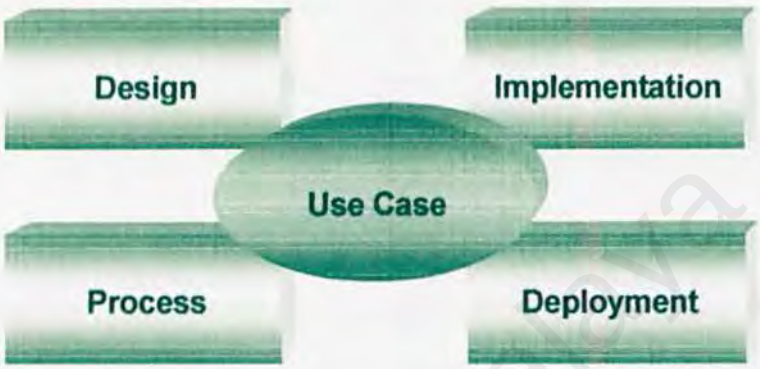

Figure 3.5 UML five views. (Scott, 2001)

The following is the description for the five views of a system:

- **Use Case View** – This UML diagram is actually focusing on the scenarios that involve actors such as human users or external system.
- **Design View** – UML played an important role in focuses on the vocabulary of the problem that needed to be solved by the system, and also the elements of the solution to particular problem during the system design phase.
- **Process View** –Some UML diagrams were used to describe the aspect of timing and the flow control of the system.
- **Implementation View** – The UML diagram will be focusing on the sub-elements that assembled to form the physical system during implementation phase.

- **Deployment View** – The UML diagram focuses on the configuration of the software, hardware and other physical elements or devices geographical distribution. (Scott, 2001)

The reasons of adopting UML technique as a tool to visualise the use case and component model for the framework of e-manufacturing system, are:

- UML is a recognised expressive visual language for modelling.
- It provided several types of diagrams to supports the conceptual of high-level development.
- It can eliminate the possible confusion among the development team by using a set of unified notation to well represent the system.
- The UML diagram could be reused as the component is being reused.

4.0 System analysis

System analysis is an essential and important phase in system development process. It is used to determine all the necessary requirements and fully document in detail what should take place before proceeding into subsequent phase. It is indeed an important feature of the system or a description of something that makes the system capable to proceed while fulfilling the system's intention. Emphasis will be on the functional and non-functional requirements of the system development needs.

System requirement analysis

Use case modeling is used to identify the various requirements that need to meet in this system framework development. The outcome of the analysis will serve as a guideline for the developer during the design on what and how the system framework has to function in order to fulfill the user requirement.

There are two types of requirements which are:

- Functional
- Non-Functional

4.1 Functional requirement analysis

The Functional requirement of the framework for E-Manufacturing system is modeled by using the UML use case. There are 4 components that provide general functionalities that shared by all modules. Following are the component that must be built in order to make the framework operates perfectly.

4.1.1 Database generator

When users have selected the targeted module to be built, the database generator will provide wizard to guide users to build the system database. Users are allowed to select fields for a targeted table. After all fields have been specified by user, the database generator will start to build the database according to user pre-defined requirement.

4.1.2 User Interface Builder

After the developer has defined and generated a database, a selection of user interface templates will be provided to user to choose. Developers are allowed to choose the template that suited to the target module. Business logic play an importance role here, where it provides several functionalities for particular module where will not exist in others module. For example, production module required the functionality for SPC sub-module where it is not need in sales module. The interfaces that generated will specific for the production module only.

Most of the interfaces generated will be in a form method. This is to allow the end system user to access to the system easier. Developers are allowed to edit the forms interfaces.

4.1.3 Report Generator

Report generator is one of the functional modules that enable the user to generate all sort of professional-looking report. This module provides an easy-to-use interface and guidance through a wizard that allows the developer to rapidly design and preview ad hoc report.

- Choose from report layout options ranging from standard reports to form letters, or build user defined report from scratch.
- There are different type chart available to enrich the option to the developer, including column chart, bar chart, line chart, pie chart, pyramid chart, cylinder chart, and etc.
- After chosen the type of chart, developer have to specify the data range from certain data source, either in a static manner or dynamic manner, or from a database as well.
- The developer can edit and change the chart properties in terms of chart title, X/Y axis value, location of legend, axes properties, colour, gridlines, and data label.
- The generated chart or report are editable even though the developer have already export the report, the developer to adjust the particular report by changing its properties based on certain condition.

4.1.4 Authentication

The authentication enables the user to determine the connections between modules using the administration control panel to monitor the entire system effectively by controlling the accessibility for each module to others module. This part can only be done after all the module have completed be built. The accessibility checklist will function as an assistant tool for system developer in planning and decided the accessibility for each module by using the password authentication.

The framework application will separated the end user system into two kinds of users which are

- High level user(administrator, supervisor) and
- Low level user (operator, end system user).

This means that after the system is completed, system developer will provide the password to the administrator. After then it is the administrator duty to determine the accessibility for each user by using the accessibility checklist during the add new user function.

Different users have different accessibility to the system where administrator have a wider accessibility than the low level user. Administrator is able to add new user, edit accessibility user, delete user and etc. This means different password will lead them to different accessibility to the system.

4.2 Functional Requirements - Use Case Driven Analysis

A use case is often referred to the main functionalities of a system, including the behaviour of a system or part of a system. It is description of a set of sequences of actions, including the alternative ways to yield an actor desirable output. Once the use case has been identified, it will be involve to the development process, it so called user case driven.

The actors are external factors that interact with a system to achieve a particular goal. The actor might be the system user, a printer, or other external system that interact with the system. The relationship between the use cases and the actors are actually modelled in a use case diagram.

The use case diagrams will be showed part by part according to the modules.

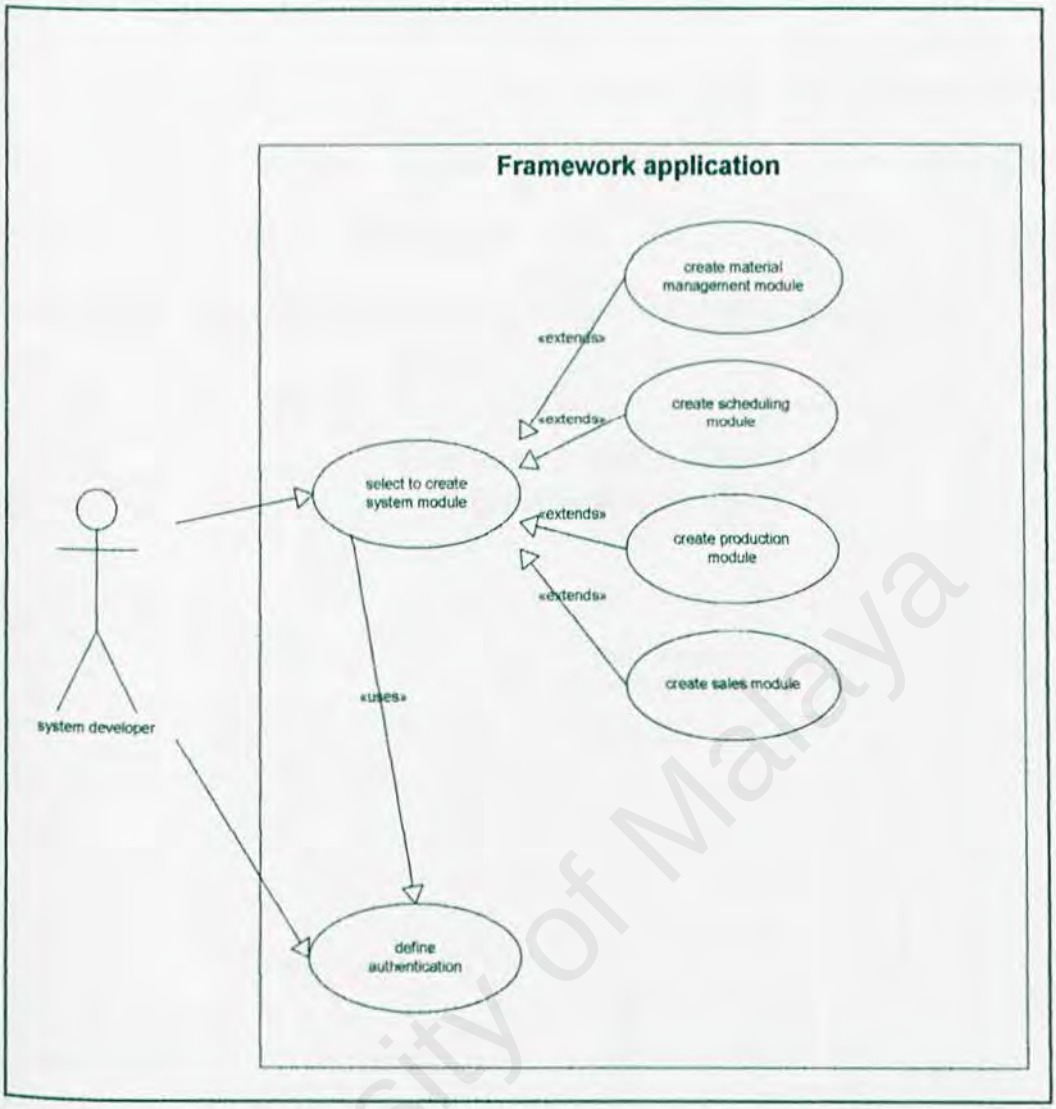

Figure 4.1 Use case for framework application

- System developer is able to use framework application to create the E-Manufacturing system. System developer is able to create 4 module which is production module, schedule module, material management module and sales module.
- After create the entire module, system developer is able to define the administrator user for the system.

4.2.1 Use Case for Material Management Module

There are several sub-modules that composed the material management module, including inventory management, shipping/receiving and purchasing management. Each of the functionalities within these sub-modules and material management module would be represented by a use case diagram as describe below:

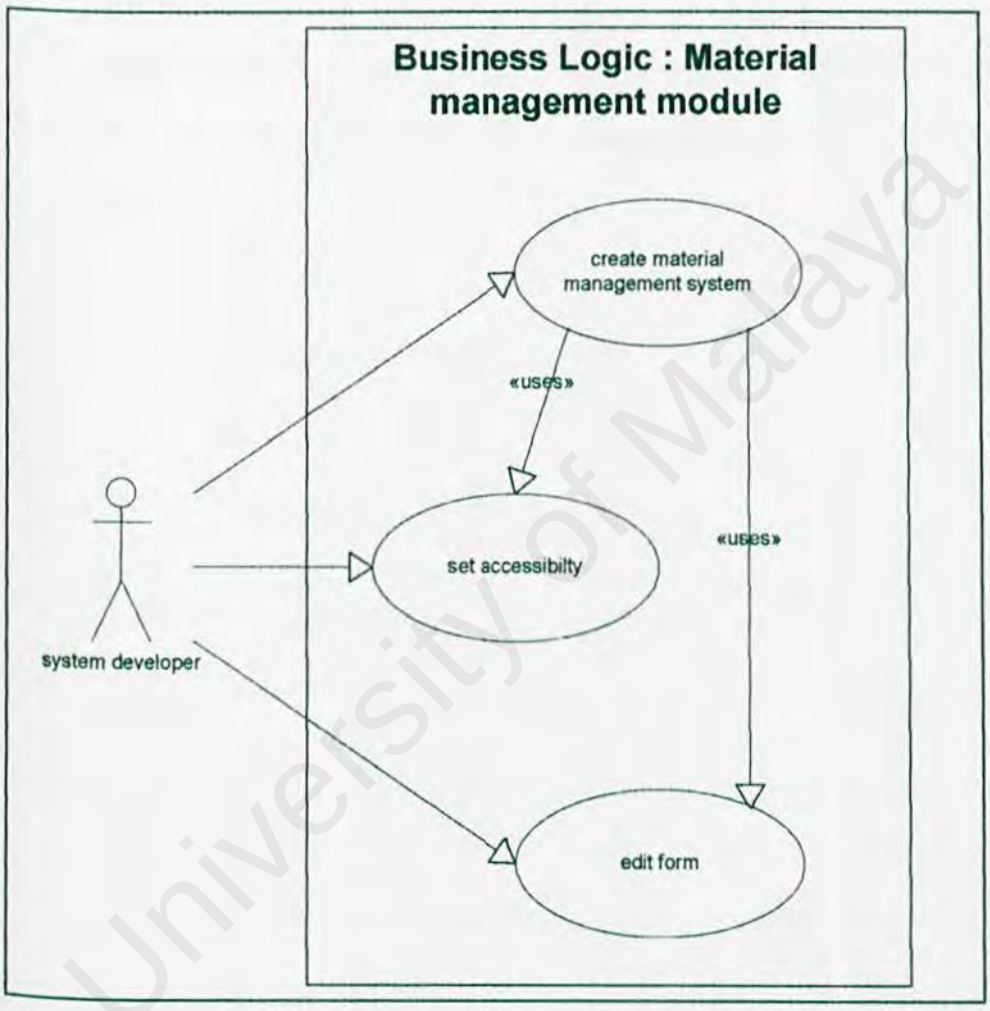

Figure 4.2 Use case for create material management module

Use case diagram above illustrates the functional requirements for material management module.

- System developer can chose to build the material management system by using the framework application.
- After the system has been build, system developer can set accessibility of this module with others module.
- System developer is then allowed to edit the form in the system, where system developer can decide either to change the form layout or delete the form.

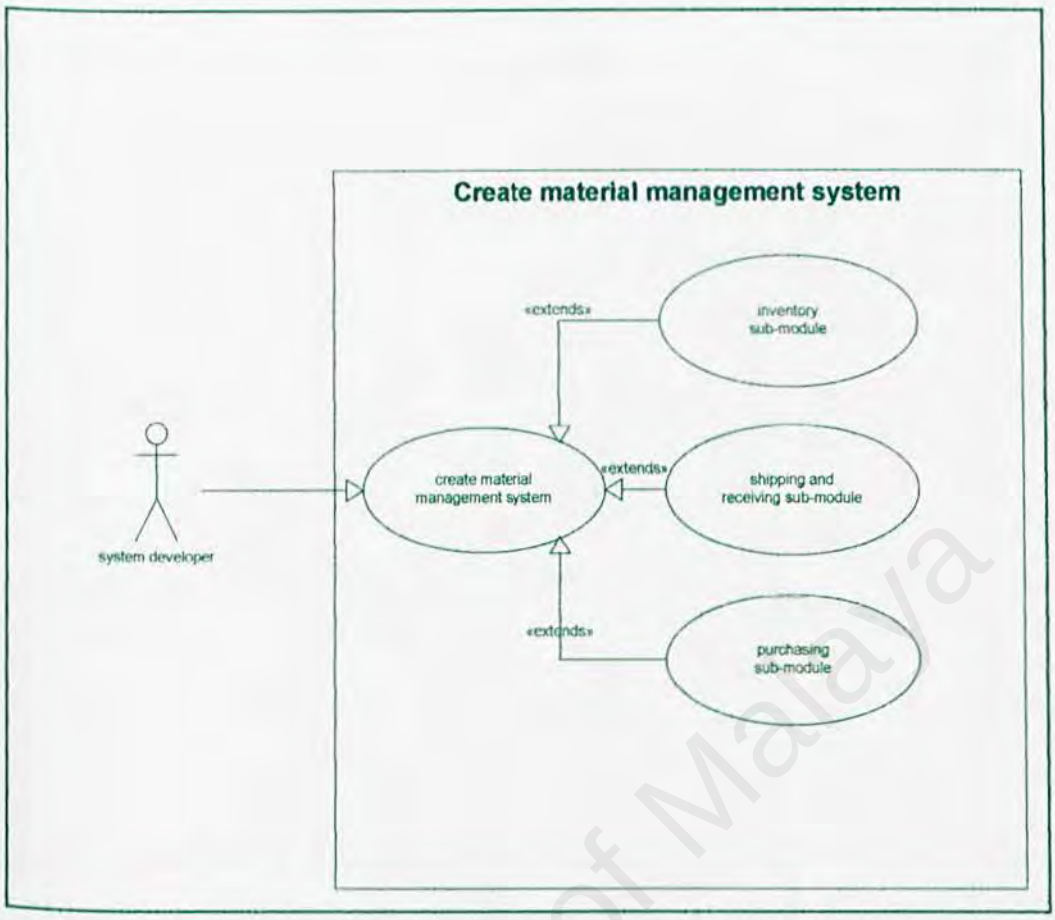

Figure 4.3 Use case diagram for create material management sub-module

Use case diagram above illustrates the functional requirements for material management module.

- There are 3 sub modules in material management module, which is inventory, shipping and receiving and purchasing. This sub-module will be build together when system developer is creating the material management system.
- Developer can specify the interfaces layout by choosing the template from the provided list.

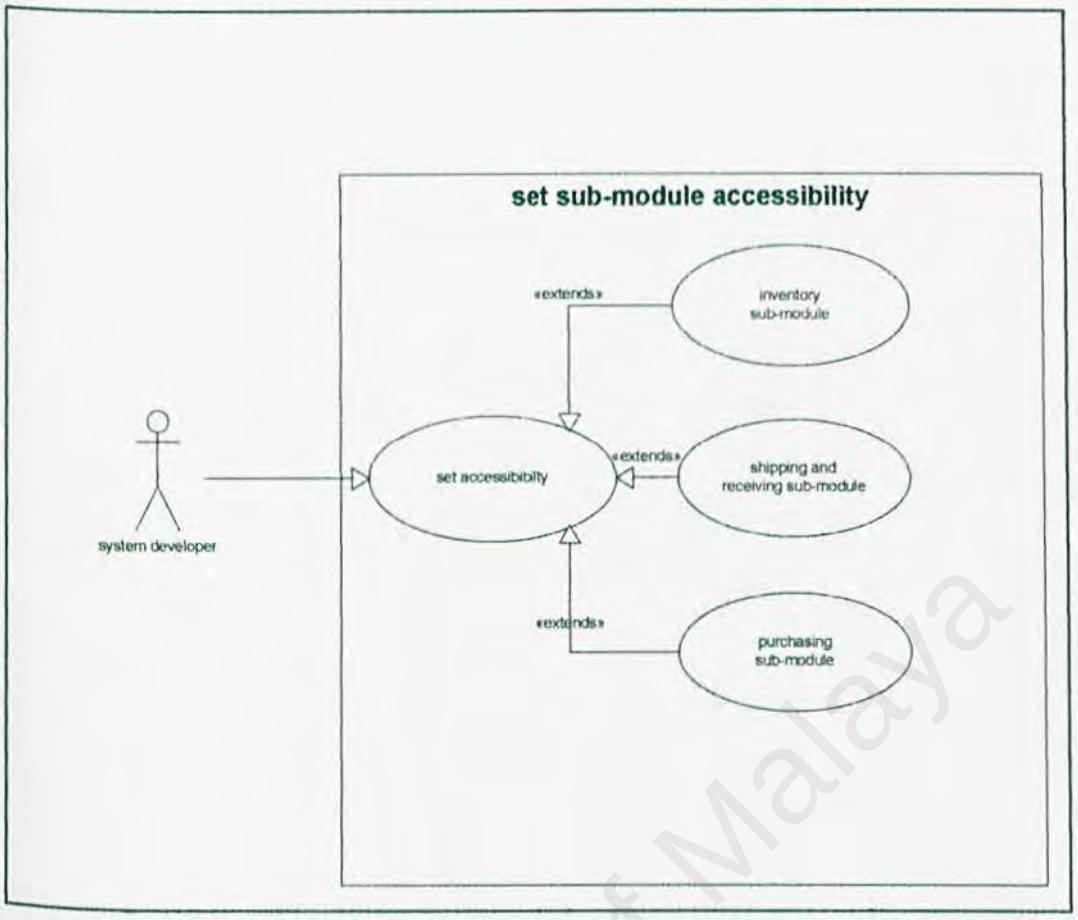

Figure 4.4 Use case diagrams for setting accessibility in material management module.

Use case diagram above illustrates the functional requirements for material management module.

- After create the sub module, system developer are able to view and set accessibility and linking for each sub-module.
- System developer can set a password for security purpose for the system.

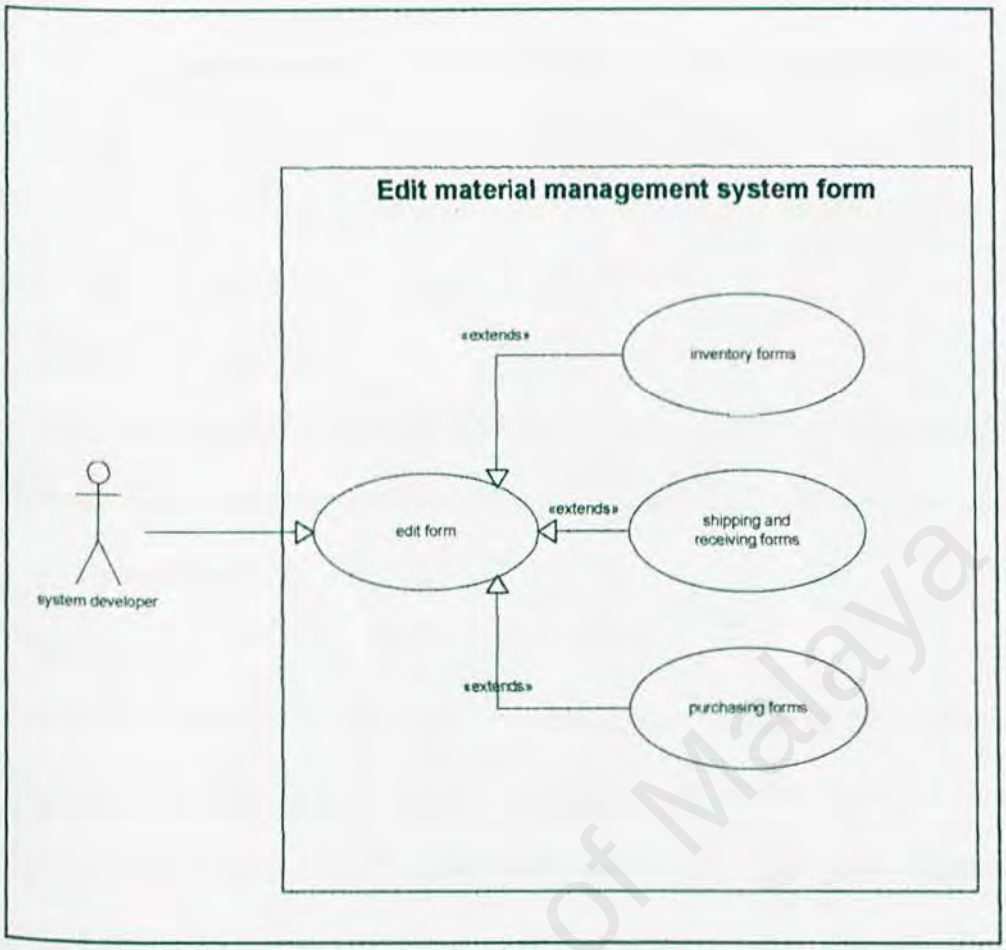

Figure 4.5 Use case diagrams for editing form in material management module.

Use case diagram above illustrates the functional requirements for material management module.

- System developer is allowed to view and edit each form in the system.
- The forms are separated according to the sub-module.

4.3 Non-Functional Requirements

A non-functional describes a restriction of the system that limits our choices for constructing a solution to the problem. These requirements are very subjective but are as important as the functional requirement. They must be fulfilling in order to avoid or minimize the failure of the system.

a) Flexibility

The framework is able to intend by adding new component in the future. This means that the manufacturer can add in more functionality that they needed into the framework.

b) Portability

The framework must be able to suit for various types of manufacture system. The portability of the system will ensure its capability to operate in various kind of manufacture system with little modification and redesign of the components.

c) Scalability

System must promise the framework serving the utmost performance at different machine specification. System must ensure that the framework still can operate normally when transfer to a machine that is greater or lesser power.

d) Modularity

Modularity involves breaking the framework into logical, manageable modules so that distinct functions of components could be isolated from one another. This will make testing and maintenance much easier.

e) Security

The system should be equipped with sufficient security. Accessibility of each user should be authenticated and validated by the system and the entire password should be encrypted. System must also show no potential leakage of information.

4.4 Run Time Requirements

Supplementary requirements are the general non-functional requirements of the framework application requirements. There are the hardware requirement constraints and software requirement constraints.

Hardware Requirements for Developer

Developer should have the following hardware and software requirements so that the process of the development will swing smoothly.

Table 4.1 Hardware Requirements for Developer

Hardware Requirements	
Processor	Intel Pentium II, 450MHZ (minimum) or higher
Memory	128MB or higher
Hard Disk Capacity	8GB (minimum)
Monitor	VGA or higher resolution
Others PC standard devices	Required

Software Requirements for Developer

Table below shows the software requirements for developer to develop application with Visual Studio .NET.

Table 4.2 Software Requirements for Developer

Software Requirements	
Operating System	Microsoft Windows 2000 or higher
Programming Tool	Visual Studio .NET 2000 or higher version
RDMS	Microsoft SQL Server 2000

Hardware Requirements for User

Hardware requirements have to take into consideration because it will give effect to the framework application that runs on it. In order to have a better performance, user has to meet the minimum hardware requirements which are stated at the table below.

Table 4.3 Hardware Requirements for User

Hardware	Server	Client
Processor	Intel Pentium II 450Mhz or higher	Intel Pentium 166Mhz or higher
Memory	192MB	64MB
Hard Disk Capacity	10GB	8GB
CD-ROM	Required	Optional
Pointing device & keyboard	Required	Required

Software Requirements for User

Below is the list of software requirements for user.

Table 4.4 Software Requirements for User

Server	Client
Windows Server 2000	Windows 98 or higher version
SQL Server 2000 .NET Framework	SQL Server Client .NET Framework

Why we choose Windows 2000 Professional?

- It's easy to use. One key feature is it is easier used and performed faster than Windows 95, Windows 98, and Windows NT 4.0.
- It is reliable where we can modify the operating system core to prevent crashes.
- It save time and increase productivity because it offers time-saving and mobile users key productivity features. It enable user to take files and folders offline and ability to hibernate and restart the system without reboot it.
- It is easy to support, manage and deploy.
- Security features in Windows 2000 Professional are applicable and used to protect sensitive data from unauthorized user.

Why choose Microsoft SQL Server 2000?

- Scalability – same database engine can be used across platforms ranging from laptop compiler running Microsoft Windows 95/98/2000 to large, multi processor servers running Microsoft Windows NT.
- Ease of installation, deployment, and use – SQL Server includes a set of administrative and deployment tools that improve ability to install, deploy, manage and use SQL Server across several sites. SQL Server 2000 is also capable of operating efficiently on a small, single-user system with minimal administrative overhead.
- Data warehousing – it includes tools for extracting and analysing summary data for online analytical processing. It also includes tools for usually designing database and analysing data using electronic based questions.

Why .Net Framework is practiced?

The Microsoft .NET Framework is an important new component of the Microsoft's operating systems. They are reliable, ease of use and deploy, and able to connect to other systems that the .NET Framework helps bring to computers.

The .NET Framework helps software developers and systems administrators easily build and maintain systems with improvements toward performance, security, and reliability.

There are many benefits of .Net Framework:

- Facilitates the development of software with improved reliability, scalability, performance, and security.
- Helps developers be more productive by:
    - Making it easier to reuse existing code.
    - Enabling developers to more easily integrate components written in any of the more than 20 supported programming languages.
    - Helping developers more easily build software for a wide range of devices using same skills and tools.

Why Visual Basic .NET?

- Since SQL Server 2000 has been chosen as the DBMS of the application. VB .NET will provide higher integration with SQL Server 2000 because both are Microsoft solutions.

- Visual Basic .NET 2003 provides developers with the award-winning Visual Studio .NET integrated development environment (IDE), which now includes faster startup time, enhanced Smart Listing for faster and more accurate coding, flexible Task Lists, property editors, IntelliSense® improvements, forms designers, and much more. In terms of performance, Visual Basic .NET byte codes are much faster than Java bytes codes because java bytes codes need to be interpreted before execution.

- IntelliSense is now available within the Immediate Window for providing assistance while debugging applications. In addition, the Visual Basic .NET IDE now offers a simplified Debug Window to provide only the most pertinent information for quickly debugging Visual Basic .NET applications. This allowed Visual Basic .NET produces high quality application in the shortest time

- The Microsoft Windows® Forms Designer included in Visual Basic .NET 2003 is an enhanced version of the forms designer that Visual Basic developers have been using for years. Features include control anchoring and docking to eliminate the need for complex resize code, an in-place menu editor to deliver WYSIWYG menu creation, the tab order editor to provide rapid application development (RAD) organization of controls, and visual inheritance.

- Visual Basic .NET 2003 developers can now leverage even more of their existing investments in code and skills. The improved upgrade wizard enables developers to migrate up to 95 percent of existing code to Visual Basic .NET. The upgrade wizard is now available in Visual Basic .NET 2003 Standard edition and Visual Studio .NET 2003 Professional, Visual Studio .NET 2003 Enterprise, and Visual Studio .NET 2003 Enterprise Architect editions.

- Visual Basic .NET provides developers with both the ActiveX® Data Objects (ADO) data access programming model for backward-compatibility, plus XML-based ADO.NET. With ADO.NET, developers gain access to more powerful components, such as the Dataset control and a strongly typed programming model that provides IntelliSense statement completion for data access code.

- Visual Basic .NET enables developers to easily deploy and consume XML Web services from within any application. Web services reduce development time by enabling software integration with applications on any platform.

Chapter 5.0 System Design

5.1 System Design

In System design stages, the functional and non-functional requirements are transforms from the system analysis phase into a formal and systematic representation of the system. The objective is to build a design model that could be used as reference to describe the conceptual and physical realisation of the requirements specification by using the use case diagram and analysis model. Then the design model created would be serving as an abstraction in the implementation phase.

5.2 System Architecture

The system architecture of the e-manufacturing framework is actually based on the three-tier architecture, which consists of three main separate layers:

The presentation layer

- This would be the Graphical user Interface (GUI) that plays a role as a terminal to interact with the system user.
- Basically this layer is responsible to capturing input and publishing output to the system user.

The logical layer

- This is the second layer, which also referred as the middle tier. This layer stores the entire business logic components that could be reused and shared among the users who interact through the presentation layer.

The data layer

- The is the third layer of three tier architecture which responsible to comprise a relational database management system that associated with logical layer by using the infrastructure of data access paradigms (ODBC or OLE DB).

The consideration and determination of technologies to be used to implement the three main layers should be covered in the system architecture, which are presentation layer (interface), logical layer (process), and data layer (data). The software development tool that is being used to develop the presentation layer and logical layer is .Net Framework, whereas the Visual Basic .Net would be the chosen programming language for system development. Since Visual Basic .Net is an ideal language in developing a software reusable component, thus it has been the main leverage in the development of component-based logical layer. Meanwhile, .Net Framework has provided the user with a Class Library that encourages the object reuse and reduces development time. A user interface library within the Class Library which enables the user to build interface easily, this facilities reduced the time of developing the presentation layer. On the other hand, Microsoft SQL Server is the chosen relational database management system in handling and monitoring the data layer.

There reasons to choose three-tier architecture as the system architecture are:

- Maintainability – It reduces the burden of maintenance, since the modification of logical layer does not affect the rest of the layers (presentation layer and data layer).
- Extensible – The logical layer could be extend by add in new business logic component without modifying interface and data structure.

- Reusable – Each business logic component can being reused or shared by different user access or request from different user interface (presentation layer).

The diagram below illustrates the system architecture of the e-manufacturing framework:

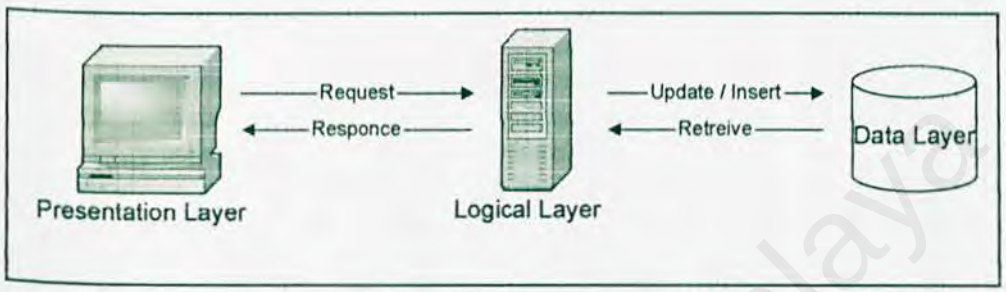

Figure 5.1 Three-Tier Architecture

5.3 Sequence Diagram for Material Management Module

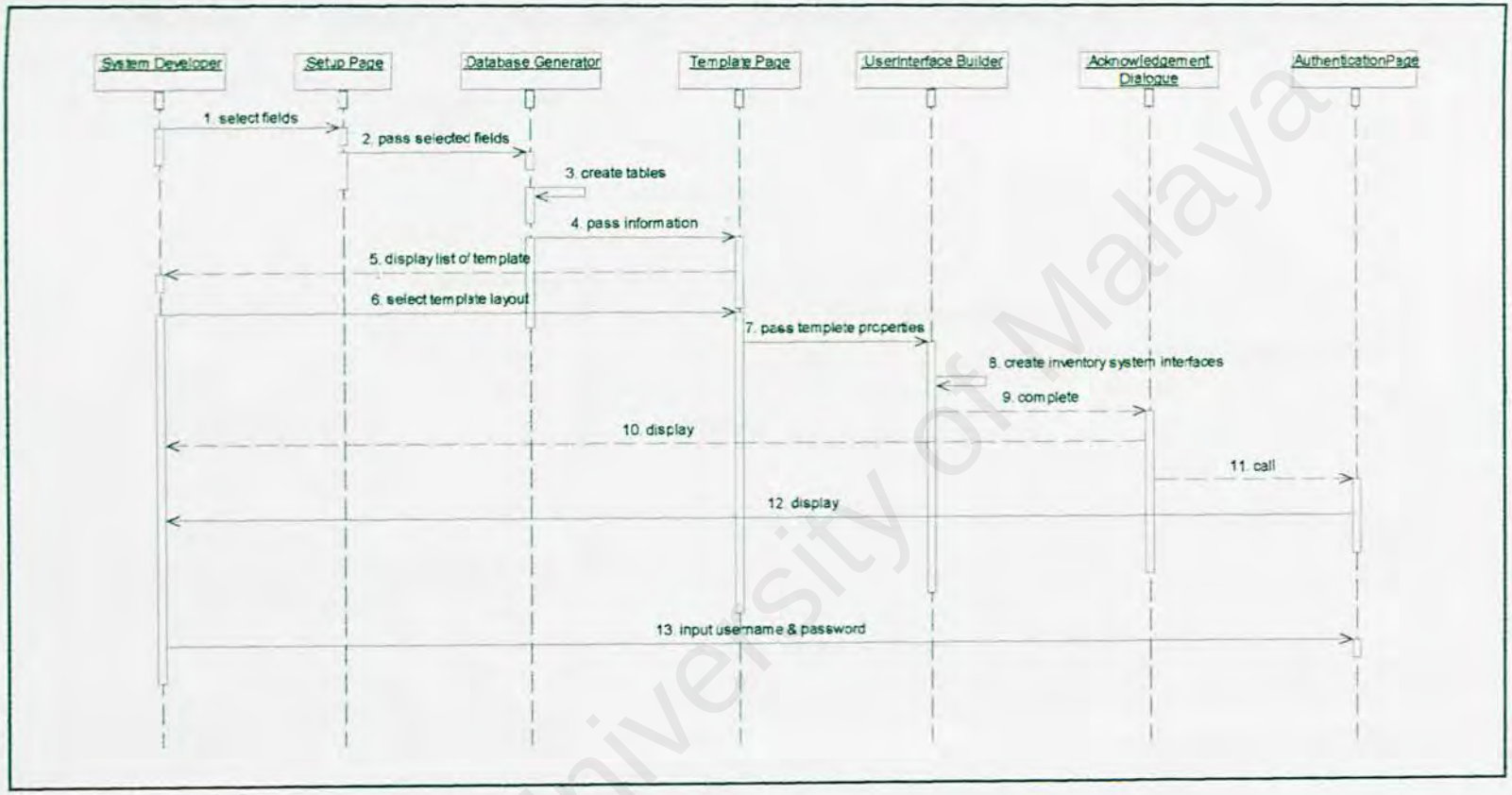

Figure 5.2 Sequence Diagram for Create Inventory Sub-Module

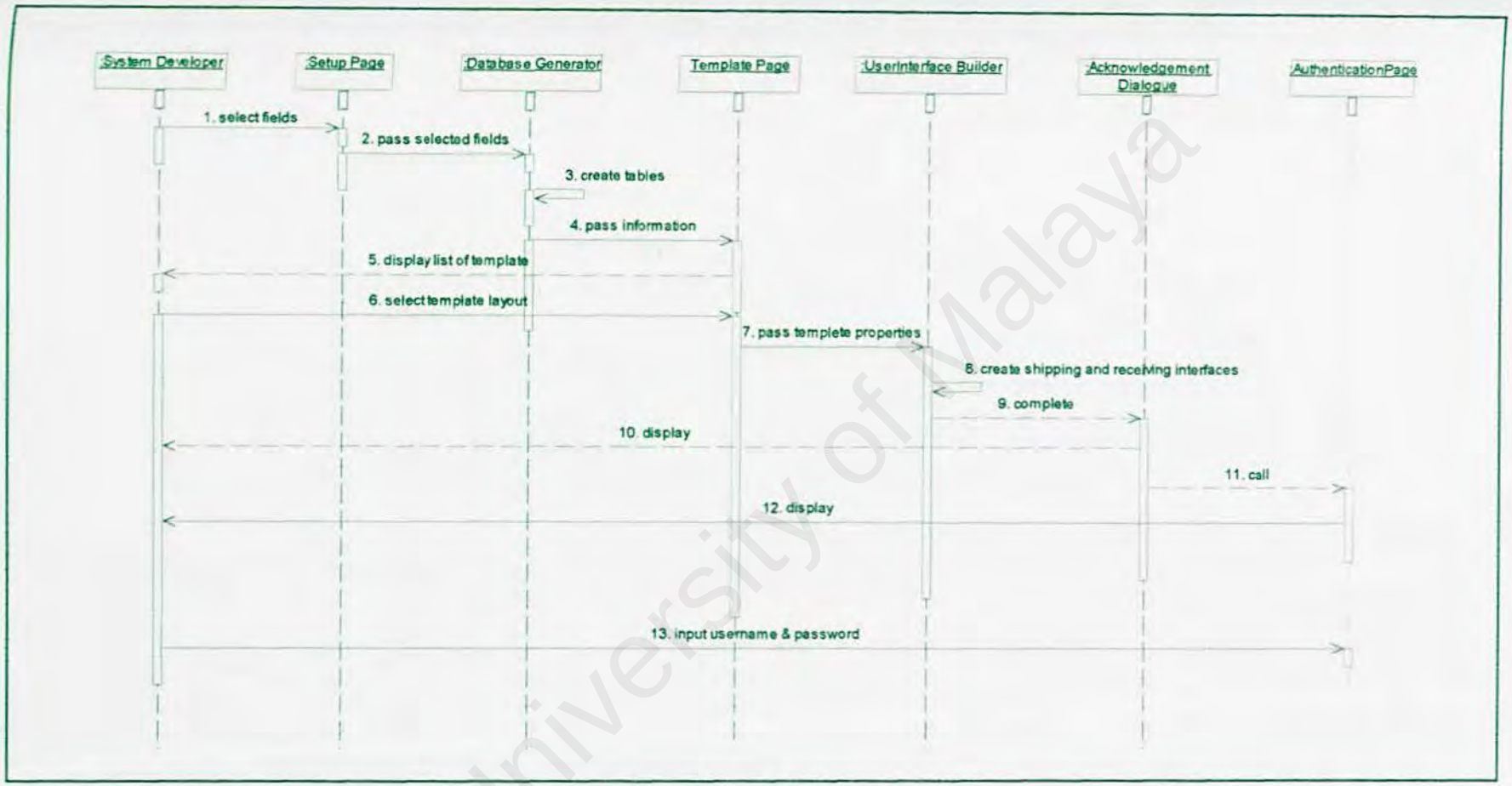

Figure 5.3 Sequence Diagram for Create Shipping and Receiving Sub-Module

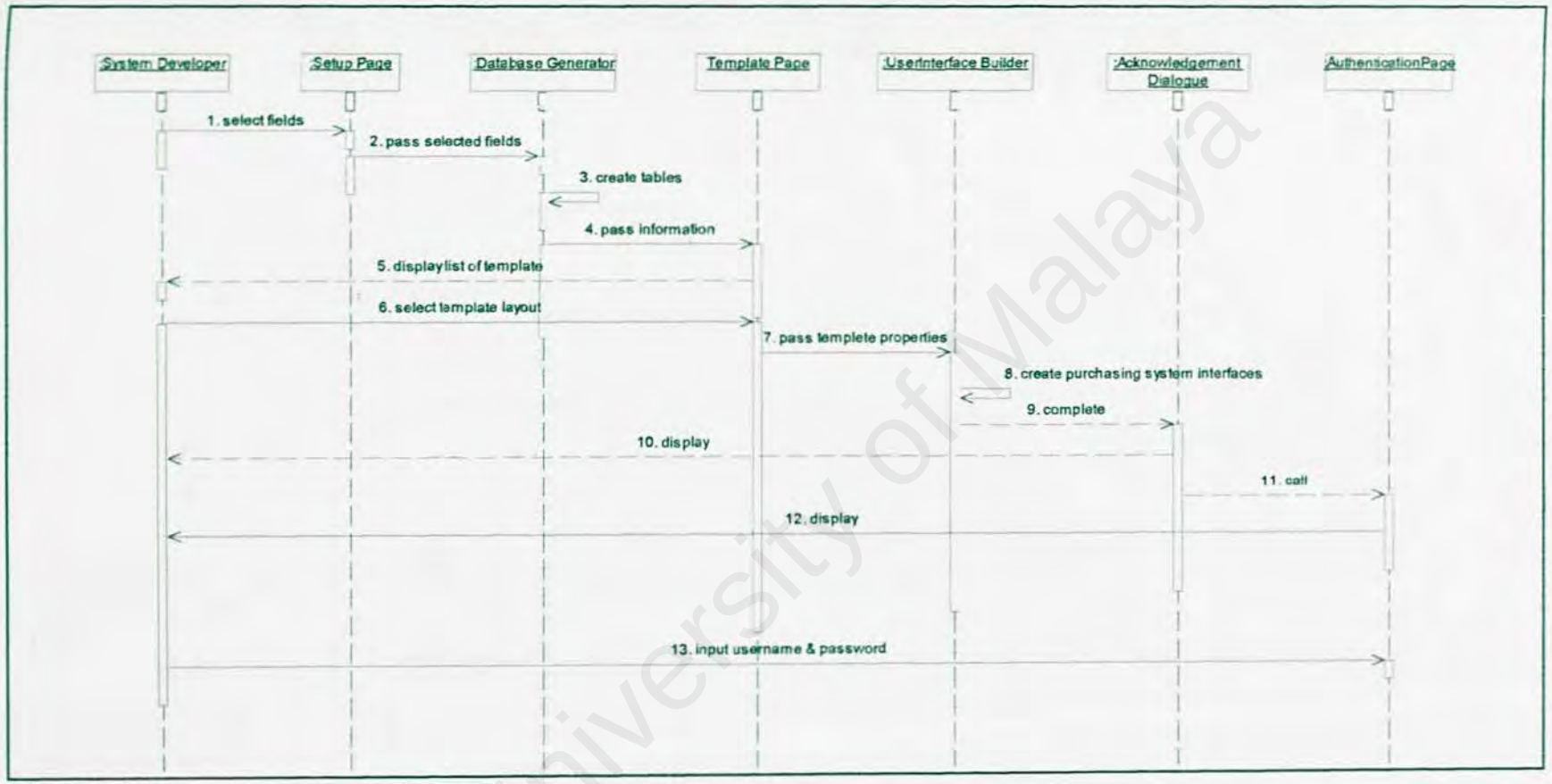

Figure 5.4 Sequence Diagram for Create Purchasing Management Sub-Module

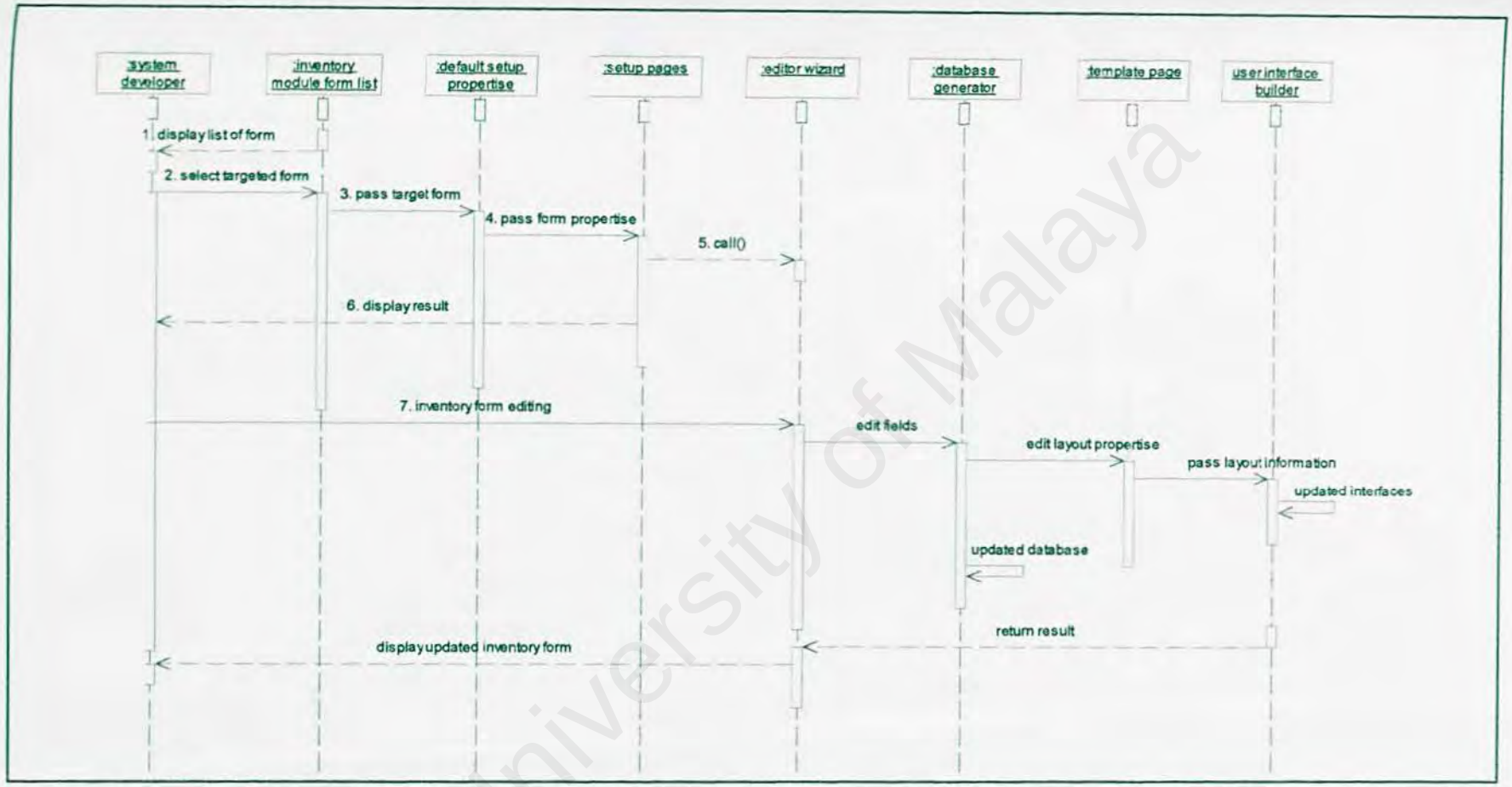

Figure 5.5 Sequence Diagram for Editing Form in Inventory System

Figure 5.6 Sequence Diagram for Editing Form in Purchasing Management System.

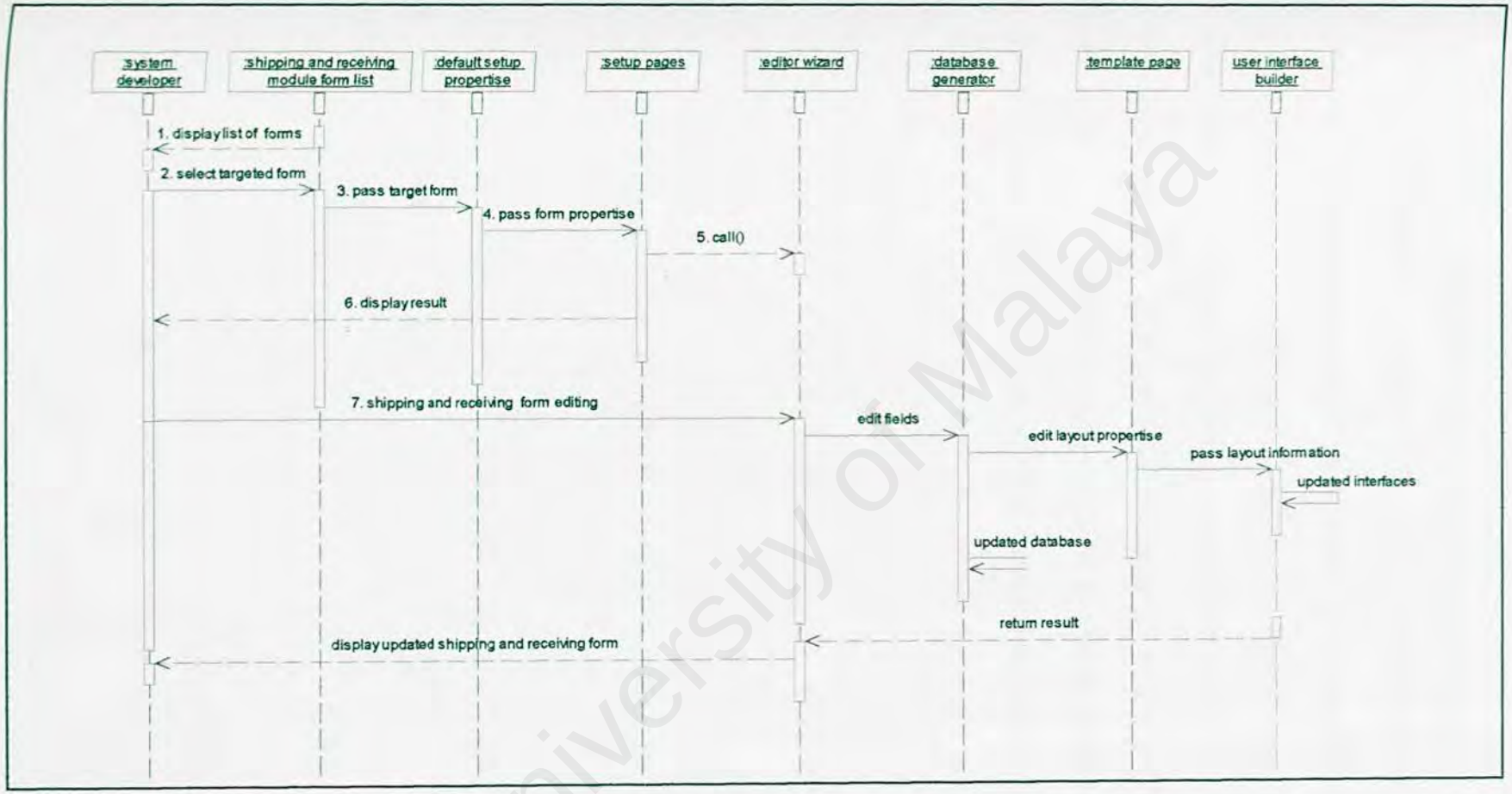

Figure 5.7 Sequence Diagram for Editing Form in Shipping and Receiving System

5.4 Class Diagram

Class design is one of the important elements that could be represented by using one of the UML diagrams, which is class diagram. This diagram contains all the required classes and objects, a class represent a collection of objects that share the same characteristics. An attribute is referred to a named property of a class; a class usually has a number of attributes. Whereas an operation of is referred to the implementation of a service or function that can be called from any object of the class. Meanwhile, each class have its own responsibilities need to be fulfilled by the associations in between the attributes and the operations of the class. A relationship need to be assigned among the classes in order to provide an abstraction of the system in terms of collaboration, it would be the stepping stone for the structure of the system.

5.4.3 Class Diagram (Material Management Module)

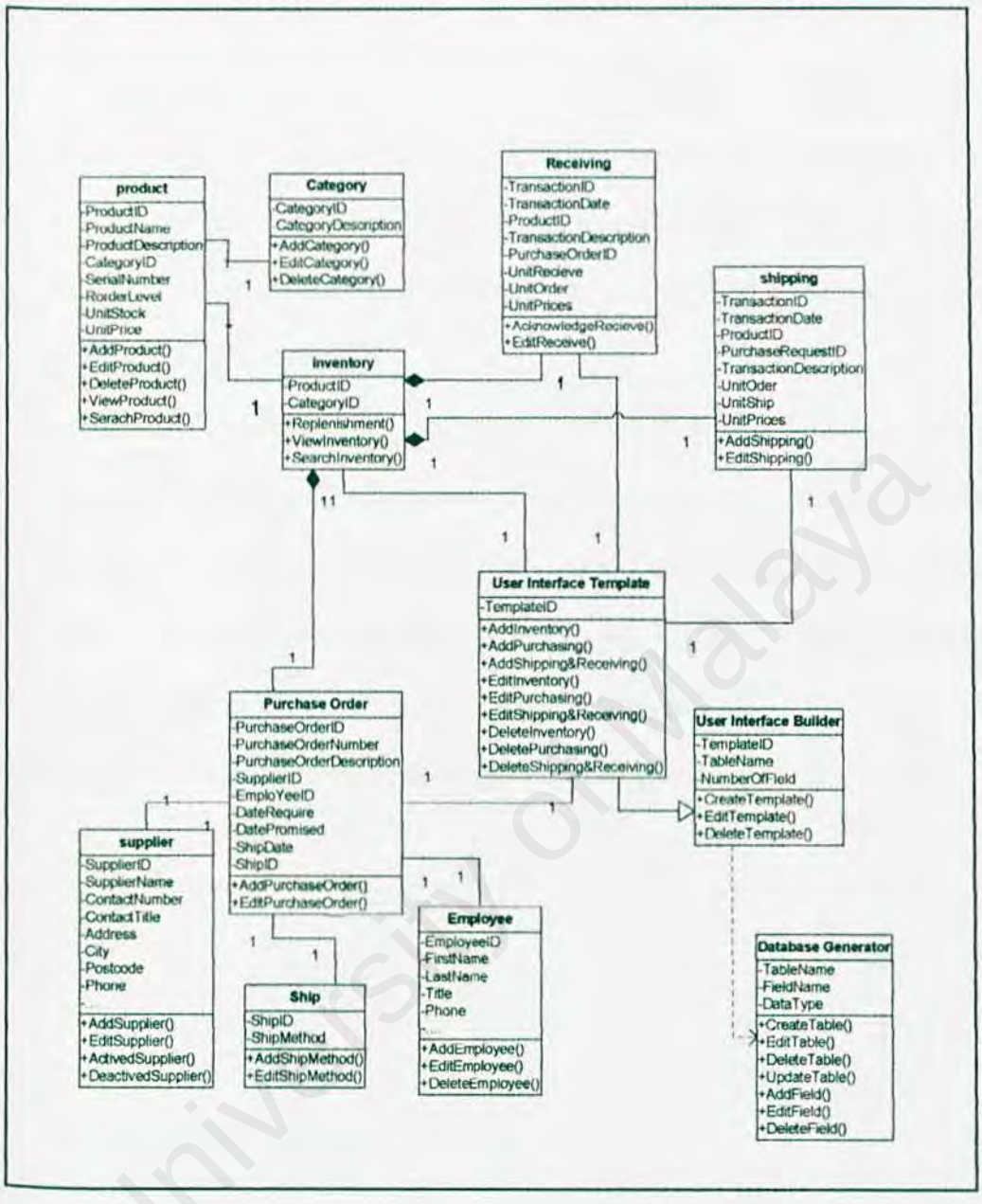

Figure 5.8 Class Diagram for Material Management Module

5.5 Chapter Summary

System design is system development phase that transforms the problems into a workable solution. These problems are actually based on the requirements that specified in the previous chapter. System design phase enable the developer to visualise and consider all the possible solutions, and determines the most appropriate solutions to solve those problem.

System design is also the key guidance towards the implementation phase, the framework, module and components that stated in the system design would act as a conceptual design which will be a guideline to developer during the construction of the system according to the requirements specification. In short, system design plays an important role in ensuring all the functional requirements and non-functional requirements are well fulfilled by the design specifications.

6.0 System Implementation

System implementation is the process that converts the system requirements and design into program code. It focuses on implementing the solution as software. This stage involves both application and database implementation where database will be created, coding of the system will be written before debugging step takes place. The effort spent in this phase will actually determines the success of the system and ease the processes of modification, debugging, testing, verification, system integration and for future enhancement.

6.1 platform development

The platform development includes setting up the development environment, create project in visual studio.net and create several databases in SQL 2000 server. Servers and development tools installations are the very first step before starting off with any development work.

6.1.1 Operating system

As mentioned the platform or operating chosen for the development of the project is Microsoft windows 2000 XP. The installation of windows 2000 XP is differences form the previous version of windows Operating system. The previous require DOS environment to install the operating system while windows 2000 offer a friendlier computing environment. when the system disk is booted, local installation file are being copied into the system and user are being offered a simple interfaces which is the menu prompting the user to input their installation options whether to clean install anew version of windows 2000 into the system or upgrade/repair the current Operating system into windows 2000.

6.1.2 Database management system

The chosen database management system is Microsoft SQL server 2000 which facilities the various data retrieving, storing, backup, restoring and others information manipulation activities. The installation is simple where user just needs to follow few steps of configuration. Since the system is connects to DBMS by using the SQLconnection, user have to make sure that the login and password is correct because problem will arise if the login and password of the user does not match. User is allowed to create more than one login and password.

A database named MM was created after the SQL server had finally installed. Then, the tables are created according to the database design. The database will become the database storages for the Material management module. When the framework system is complete, the tables that user created will also save in this database.

6.2 development environment

6.2.1 Actual hardware requirement

The hardware used to develop the system are listed as below

- Intel Pentium IV processor and above
- 128 RAM and above
- 2 GB hard disk with minimum of 500MB of free space
- keyboard and mouse
- Others standard desktop PC components.

6.2.2 Actual software tools requirement

During the system development, a lot of software tools were used. Table below show the list of software which used to develop the system.

Table 6.1 software tools lists

Microsoft windows XP	system requirement	Operating system
Microsoft SQL server	database	DBMS
Microsoft visual studio.net	system development	code development tools
Adobe Photoshop 5.0	interfaces design	button, background and icon design

6.3 program development and coding

Program development and coding is the phase which takes the longest time in the **development life cycle**. Using the right tools and the methods to develop the system are crucial in determining the success of the project.

Microsoft visual studio.net is the most suitable programming platform when developing the visual basic.net pages. This tool enables user to develop a system via a graphical user interfaces (GUI) and enable easy performance of many complex programming.

The language chosen to develop the system is Visual studio.net

Before starting on the coding process or any others detailed works on the program, a review on the program documentation need to carried on before keep going with the coding design of the system.

6.3.1 Review the program documentation

The first step taken in system development phase is to review the program documentation that was prepared during the early phases. The program documentation consists of architectural view, concepts and controls, module flow diagram, data dictionary and also the sample layout of the interfaces. The documentation provides clear guide and understanding of the works that need to be done in the coding phase. Besides, the documentation also consists of several programming references book such as visual basic .net programming and visual studio .net professional. This is to give us references about the coding part.

6.3.2 Designing the program

After reviewing the program documentation, design the program is the next following step. At this stage, we have to determine how the program can accomplish all the features and functions that need to be existed as describe in the program documentation. Then a logic solution for the programming problem is developed when we follow step and step to develop the system.

6.3.3 Coding approach

Coding the program is the process of writing the program instruction that implement the program design which translate the design specifications into a machine- readable format. It is an iterative process whereby it is done until the programmer obtains the desired results. If design is performed in a detailed manner, coding can be accomplished mechanically.

For this system, coding is done with the bottom up approach. The bottom up approach is begin by coding some complete lower components and leaves the high modules merely as skeletons that used to call the lower modules. Each lower -level function and procedure was develop individually which are then integrates into appropriate high level modules accordingly. Bottom up approach offers some advantages such as:

- Testing can be conduct on some of the modules while the others are still under construction and faults are easier to be detected.
- Critical functions can be coded initially to test their efficiency.
- Increase the development process as the lower-level modules or functions can be builds independently and smulationeously without waiting or delaying the others.

6.3.4 Style adopted

The coding paradigm adopted by the system is oriented at giving reliability and performances a balance

Maintainability

Codes are well organized. Task oriented codes are centrally located.

Readability

Readability is importance for the ease of the future enhancement. Codes are formatted to enhance understanding such as meaning variables and labels names, line brake or spacing in between diffrent section, comment and proper identification.

Reusability

Task oriented codes are easily developed for reuse, especially for task that cross application boundaries. Reusability is important as a method for improving quality throughout the system development process. It is powerful to create class' component that to be reused in others subsequent and related applications. It also reduces the coding times as well as reducing the errors occur.

Testability

Modules can be tested easily. The system has the ability to checks on the system input to ensure correct data is process and to protect system integrity

Speed

Module code can sagely be optimized without affecting the calling procedures.

The coding style is very important attribute to determine the readability and maintainability of the source codes. With a clear and systematic coding style, it helps the programmer to see the codes easier in order to help the programmer on maintaining and also debugging the system. Elements taken into considerations while coding an easy to maintain and enhance system are internal documentation, standard naming convention and etc.

Spacing and functional segment

Indent the code according to functional segment (for example, the loop or the if-else statement). Spacing or lines break in between different section of the code to enhance readability.

Internal documentation

Internal documentation is a description material written directly within the code. It means that internal documentation refers to comments within the code. It provides line-by-line explanation of what the program is doing such as the decryption about data structures, algorithms and so on. Besides, it also breaks the code into phases that represent major activities. This is needed to enhance readability of the code by someone else than a programmer. It will also help the programmer to recall what the function does or where the functions locate in the coding.

Different programming languages uses distinguish comment syntax. In Microsoft visual studio.net, the ' is use as the comment syntax. For instances

'Delete the fields selected

All the word after the symbol 'wills automatic change to comment syntax with the green text colour. For the next line, the symbol ' also has to put the font of the comment.

For example:

'Connect to database

'Retrieve all data from table product

Naming conversions

Naming conversions is an abbreviation of the control name or the object name. Standard naming conversion provides programmers with easy identification of variables. This system uses a naming conversion to ensure uniformity of the control and object name. Usages of these standards perform as a mean towards coding consistency and standardisation. Besides, it can also increase the readability of the codes

Follows are the naming conversions being applied:

Table 6.2 naming conversions

Control	control name	example
Command button	cmd_	cmd_close
SQLconnection	db.	db.openconnection db.closeconnection
Textbox	txt_	txt_categories

Sample code

Below is some of the sample code written in this system.

Visual basic example:

Selected and pass value

```
Dim DeleteItems As String
Dim table As String

DeleteItems = CheckedListBox1.SelectedItem
table = ListBox1.SelectedItem

If DeleteItems = "" Then
   MessageBox.Show ("Please select field to delete.")

End If
```

Xml example:

```
<?xml version="1.0"?>
<PropertiesPage xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <PropertiesList>
  <FormProper>
   <ControlLblName>Products</ControlLblName>
   <ControlLblText>ProductID</ControlLblText>
   <ControlLblSize>120,23</ControlLblSize>
   <ControlLblLocation>16,16</ControlLblLocation>
   <ControlType>TextBox</ControlType>
   <ControlName>Products</ControlName>
   <ControlFormat>txtProducts</ControlFormat>
   <ControlLocation>152,16</ControlLocation>
   <ControlSize>100,20</ControlSize>
  </FormProper>
```

Sqlexample:

```
Public Function UpdateData(ByVal TableName As String, ByVal
ColumnName As String, ByVal Value As String, ByVal ID As String,
ByVal IDValue As String)
        Dim SqlStatement As String

        SqlStatement = "UPDATE " + TableName + " Set " + ColumnName
+ "='" + Value + "' WHERE " + ID + "='" + IDValue + "'"
        CommandObject = New SqlCommand(SqlStatement,
ConnectionObject)
        CommandObject.ExecuteNonQuery()
    End Function
```

Encryption

The .NET framework provides several classes that enable user to perform cryptographic transformation on data using several standard algorithms.
For this system, encryption is used to protect user's login id and password from being read.

Exception handling

Exceptions, which also called as anomalous situations may occur during the execution of the program. These exceptions are handling by code that is outside the normal flow of control.
To ensure that the system handles errors gracefully, exception handling technique was implemented in the codes. The proper catch block is determined by matching the type of the exception to the name of the exception specified in the catch block. Besides, throw statement had been used to explicitly throw an exception to provide information when debugging.

Database connection

The database connection is an importance step that must be done before start the coding of this system because the system needs to retrieve and insert data into database. all importance information such as database sever , database name, username, and the [password needed to access to the database must be specified correctly. after the configuration, the coding can directly connect to the server and communication with the database begin. actually a database generator component is build for the purpose of manipulating the data between the system and database.

this component perform several functions such as connect to database, open

the sqlconnection,close sqlconncetion, build new table, delete table, edit table, update table, rename fields, retirve information for the table and so on.

Following are the two steps in order to open a database connection by using the database generator component.

```
Dim db As New DBGenerator.DatabaseGenerator
CreateConnection(Server, Database, Username, Password)
db.OpenConnection()
db.DeleteData("MMFields", "FieldName", "" & DeleteItems)
db.CloseConnection()
```

6.3.5 Module implementation

There 4 main modules in e-manufacturing system. The module presented here is material management module.

Authentication

1. Login

When user comes to the framework of the e-manufacturing system, they have to login by entering tier username and password. System will check whether they are validating user and system will also check whether they belong to administrator level or just low level user.

2. Logout

When user intends to leave the system, they click the logout button and then exits.

3. New staff profile

This section allows administrator only to add new user of the system.

4. Change server.

This section allow administrator to change the server to the target database to avoid information duplication between the user side and administrator side.

Create table

1. Select table and fields.

System developer can actually selected the table list provide and check the fields given.

2. Add new fields

System developer can add fields after selected the particular table if they feel than the fields provided is enough.

3. Delete existing fields

System developer can delete any fields form the table.

4. Edit fields name

System developer can actually rename the field name if the fields name is not suitable.

Selected template

1. Select template and view

System developer can selected template for interfaces display from a list and preview the template in picture.

Product maintained

1. Register product

System administrator and user can both add product through this section

2. Delete products

The features are used to remove any existing item record form the database.

3. Edit product

These features allow user to edit the previous dataset that already exists.

Inventory control

1. Stock check in

The features are designed for receiving stock into the system as it is delivered and received.

2. Stock adjustment

The sections allow the user check and edit stock on hand.

6.4 Chapter Summary

The system implementation focuses on various aspects which governs the functionality of the system to fulfil the functional and non-functional requirements of the system. During the system implementation, system requirement and design were converted into program codes. Besides, it also involves development environments setting such as the operating system and the database server. Several software tools were used to deploy the design into machine - readable language and then in turn to produce the required applications.

This chapter shows the details on approaches used in writing codes, scripts languages used to enhance the whole system and algorithm used in implementing the system. Error checking is importance to make sure that the system runs smoothly and without showing unnecessary error massages.

Chapter 7.0 System testing

After coding the system component, we usually examine the code to spot faults and eliminate them immediately. Testing is a process that focused on finding faults along the development of the system. A test is considered successful only when a hiding fault is discovered or new failure occurs as a result of our testing procedures.

An error means the system is in the state such that further processing by the system will lead to a failure. Faults, which also called bug, are the mechanical or algorithmic cause of an error. Fault identification is a process determining what faults caused the failure, and fault correction or removal is the process of making changes to the system so those faults are removed.

Software testing is crucial for software quality assurance and to increase the software reliability. In this chapter, software testing fundamentals, testing strategies and debugging methods will be presented.

7.1 types of faults

There are commonly 3 types of faults, namely, algorithmic faults, syntax faults, and documentation faults.

Algorithm faults occurred when a program algorithm/logic does no produce the proper output for a given input. This situation may cause by something is wrong with the process step. This fault is easy is spot, just by reading through the program, which also call as desk checking. It also can be track by submitting input data from each of the different classes of the data that we expect the program to receive during its regular working.

Syntax fault can be checked while parsing for algorithm fault. This is to ensure that the construct of programming language is properly used.

Documentation fault occur if the documentation does not match what the application does, and such fault can lead to others faults because of wrong implementation.

7.2 Testing strategies

Testing is a process of exercising or evaluating a system manual or automatic means to verify that it has satisfied the requirement or to detect differences between expected and observed results.

There are 4 types of testing strategies which is unit testing, integration testing, system testing, user acceptance testing.

7.2.1 Unit testing

Unit testing is performed for each of the smallest units of code. Each program component is test by it own isolated from the others components. This process verifies that the component can functions properly with the types of input expected.

Unit resting is start by examine the program code by reading through it and try to spot algorithm, data and syntax faults. It is done by comparing the code with the predefine specification to ensure that all relevant cases has been consider. Finally, a test case is produce where expected inputs are converted into desired output.

Following steps are used in carry out the unit testing:

1. Examine the code of the program by reading through it. Try to spot for possible algorithmic, data and syntax faults.
2. Control objective are tested to ensure its functionality.
3. Different data types are used to test the error handling functions.
4. Test cases are developed to ensure that the input is properly converted into the desired output.

Examine the code

In this stage, walk through the code is carried out to identify new possible faults. The code and the accompanying documentation are presented during reviewing. There are also a project review team which consist of my group member. This team will comment on their correctness.

Control object testing

Control object such as button and textbox are tested for the functionality. Button is clicked to make sure that they perform expected response, while textbox are tested with different data type input to perform validations.

Different data testing

Different data types such as number, character, date, symbols are used as test to ensure that invalid data for certain function can be tracked by the system without causing serious error. Apart from the normal and error data, NULL and space bar are also use to detect faults.

Choosing test case

To test a component, input data and condition has to be chosen. Then the component is allowed to manipulate the data, and the output is observed. The input is pre-selected so that the output demonstrates something about the behaviour of the code.

A test point/test case is a particular choice of input data to be used in testing a program. To perform test on component, we must first determine the test objective. Then, select the test case and define a test designed to meet the specific objective. Some data are purposely chosen to be improper. This is to check that the component can handles incorrect data gracefully.

Unit test example

Each table in database has at least associated with one trigger program. Unit resting carried out on each trigger program once it has completed. For examples:

1. add new fields into database
2. delete fields form database

Try the update/edit function, which mean edit/updating the existing record in database. Unit testing make sure each record has been updated/edited/ correctly.

1 Make change to the existing value.

2 Perform the updating to the record

3 Then, check from the database and see whether the value is change.

7.2.2 Module testing

When individual components are correctly working and meet their objectives, these components are combined into a module. A module is a collection of independent components. Module testing enables each module to be tested independently. This will ensure that the module calling sequence in this project is systematic.

Before allowing administrator to access the entire system, he\she has to undertake on authentication and password. Testing is carried out to ensure that the system perform verification properly and granted the aces only to the proper and valid person.

1. At first, try to login with the password provided.
2. Key in the userID and password.
3. Check whether system performs validation and then return user accessibilities, which mean user are lead to the specified pages.

Module

1. After success login into the main pages, try the button at the menu bar.
2. Click on certain button and check for the module pages appear.

7.2.3 Integration testing

As each module was able to perform their functions successfully, these components are then combined into a framework system. In others words, integration testing is the process of verifying that all the module components able work together as a whole system.

This integration is planned and coordinated to uncover errors associated with combination of modules. Interfaces as the module calling sequence are systematized and connect to the right form or pages. Several integrations testing approach exists. However, sandwich integration was used in this system. This approach combines top down integration with bottom-up integration. The system is viewed as 3 layers, which is developer framework, target user system, end system user system.

The testing begin from the main pages and then move into the backend or front-end of the system framework. Then it continues to access the functions of each module. Several test is perform to ensure the system working properly.

Integration testing example

Testing is performed several times on the same scenario as well as different situation with various data types. Therefore a lot of test case involved.

Example 1

1. Click on product button, product form will show up.
2. Enter new product information as specified in the form.
3. Click the ok button to submit the form.
4. Check the database and see whether the new information is updated.

7.2.4 System testing

It is the final testing procedure done in system testing. The objective of unit testing and integration testing is to ensure that the code has been implemented properly. In system testing, the testing is to ensue that the system does what users want it to be done.

7.2.4.1 Performance testing

Performance testing addressed the non-functional requirement of the application. Once the functions are convinced work as specified, the performance test compares the integrated components with the non-functional system requirement. The types of performance test carried out for this application are:

1. **compatibility tests**

This test was performed to find out that the interfaces functions perform according to the requirement. Results clarify that the accuracy of data retrieval was high in this system.

2. **Security tests**

This test ensures that the application fulfils the security requirements. Only the valid users are granted the access to the secure zone

3. **Volume tests.**

The fields and records are checked to see if they can accommodate all the expected data.

7.2.4.2 Functional testing

Function testing is based on the system functional requirement. In the others words, a functional test issued to check that whether the integrated system performs its functions as specified in the system requirement. Each module is tested individually to determine whether the system performs as required.

7.2.5 User Acceptance Testing

The final phase of the system testing is on the user acceptance testing. This is to ensure that system is able to functions normal in run time environment. The acceptance testing mechanism approach is duly focusing on the simulations of a real time business event for an organization. The target user now is the end system user which used the e-Manufacturing system that developed from the framework we developed.

This testing is importance to get the view of this system and any comment form them is useful for future enhancement. The more we understand the user requirement needs, the more we put the system in a valuable situation.

Some of the comments that are given by the users are stated below.

- More attractive interfaces are need.
- User required more templates.
- It is better to include the help file in the system.

7.3 Debugging Strategies

There are several debugging strategies that applied in this system which is:

1. **Using command prompt**

When errors occurred, we use a command prompt to view the error massage. This is a very effective way in debugging the system.

2. **Reviewing the algorithm used**

If a program is running well, but the information is not what as intended, this means that there may be a logic/database error. Reviewing algorithm and computations for their correctness and efficiency is needed for this purpose. Sometime, by using different algorithms, the efficiency of the program will be increasing.

3. **Display on screen the passing value**

One of the possibilities of retrieving wrong information is that the wrong value being passed or extracted. To ensure that the right value has been retried or passed, the passing value is prompted on screen for reviewing by using massage box.

4. **Check success status**

Some processes are dependent where failure in the previous process will affect many others process. In order to avoid chain reaction form this kind of process, success status is purposely set to return a true or false value. This can help to determine that where the errors occurred and make coding checking more effective.

5. Using query analyser provided by SQL server 2000

When the database transactions error occurred, the error massage will be displayed if it was caused by the error executing the SQL statement. Therefore, the SQL statements that return wrong results can be tested in SQL server by using the SQL query analyzer. The analyzer can actually provide more information about the error massage. Other than that, Query analyzer help in correcting the SQL which is wrong. If the SQL statement performed successfully, we can directly see the outcome in the database immediately. Then the correct SQL statement can be applied in the system code.

7.4 Chapter Summary

During the testing phase, which started when we started building the system until the last day of the implementation, several testing strategies ware being used to ensure that the system integrated was developed successfully. Unit, module, integration, system and user acceptance testing has been used carried out for this system. These testing have influence the system quality and reliability to become better.

Chapter 8 - System Evaluation

System evaluation is basically accessing and reviewing the system from the time of conception until it is tested and released. Evaluation was related to user environment, attributes, information priorities and several others concerns that are to be consider carefully before effectiveness can be concluded. At all phase of the system approaches, evaluation is a process that occurs continuously, drawing on a variety of sources information.

During the system evaluation, problems pertaining to software and hardware were observed, and suggestions for improvements are noted down and reviewed. Changes are made, whereby the system's strengths and limitations were evaluated by variety of users. Future enhancements are carried out if the systems are then based on the evaluation results from them.

There are many evaluation techniques that can be used it evaluate the final system. The following section will explain in details about the system strengths and limitations.

8.1 Problems Encounter and Solutions

8.1.1 Scope is not clearly defined.

In the initial of development, several problems ware encounter in specifying the scope of the system. The scope must be clearly defined before the coding can actually start. Moreover, scope is also required in the design of the system. The solution that have been done to capture the system requirements are through the interviews with all the potential users, check the similar system in market and form a discussion groups. We also find lecturers, and programmer friend for more advice.

8.1.2 Lack knowledge on SQL server

Using the extensive and relational database like sq server indeed requires much understanding of its concept and its features. The way of administering and manipulating database server are far different form ordinary stand alone database system. Therefore, more reading and database research have been done in order to have a more understanding on how to use the feature provided by SQL server.

8.1.3 Unfamiliarity in development tools used

During the early stage of system implementation, problems were encounter in using the development tool to develop the system. The Microsoft studio.NET is quite new in the markets and we never in touch with this tool. In short time, I have to learn and master the tool. However, with constant use and reference to the tools development, I manage to get more familiar with the tool in order to develop the system.

8.1.4 Lack of resources.

Due to lack resources such as hardware, people and time, the system could not be tested on a wider scale. However, the system were tested sufficiently to ensure quality, and conformance to user' requirement, guidelines and standards.

8.2 System Strengths

8.2.1 Simple and user friend interface

E-Manufacturing framework interfaces shows some of the advantages in term of its usability. Windows features such as consistent user interfaces are available in this system. E-Manufacturing framework provides standard interfaces appearance through out the whole system. This is to ensure that user can easily use the system without any briefing training.

8.2.2 Database maintenance

Authorize user are able to do housekeeping for database maintenance. They can create, add, modify and delete the information in the database. Therefore, the database will not accumulate with lots of useless data.

With the capability if the Microsoft SQL server 2000, it provide the functional of database backup. All the records and information in the table and database can be restore form the backup file if the system is corrupts. This feature increase the powerful and performance of the system.

8.2.3 Security protection

E-Manufacturing is a password-protected system, which prevents unauthorized users form accessing and changing the system data stored in the database. With the access level of control, the authorized users only have limited access to the system. Besides, the database is secured because all sensitive information is stored in an encrypted format.

8.2.4 Data validation and exception handing

Data validation is performed before record is inserted into the database. The purpose of this feature is to make sure that invalid data will not insert into the database and cause error. All fields in the form will be checked for null value or invalid data type. Error massage will be displayed when the system encounters exceptions and it will not terminate suddenly.

8.2.5 Report generating

System is able to generate summary report by using crystal reports. User is able to view their reports based on their selection criteria on the recorded exists. These summary reports are in graph format and text format.

8.3 System Constraints

There are some system limitations due to the system constraints, facilities constrains, limitation of programming itself and the project boundaries

8.3.1 Help file is not provide

Current version E-Manufacturing did not provide a help file that shows the user how to fully utilize the system.

8.3.2 Rebuilds constraints

If user wants to rebuild an e-manufacturing system in the same pc, it required user to set for new database or user will need to delete tables create by previous e-manufacturing framework. This is because the system will be confuse if the table it required to build is already exists in database.

8.3.3 Limited templates

The template provide in the system are limited for several choices. For future enhancement, more templates will be inserted.

8.3.4 Input constraints

The input methods in end user system are more to textbox method because at this phase, the system is only able to generate textbox form XML as the input methods for the system.

8.4 Future Enhancements

Below are the suggestions for the enhancement to add more value to the current version of E-Manufacturing.

8.4.1 Provide helps file

Helps file will be insert into the system functions to help the user so that they can fully utilize the system functions.

8.4.2 Textbox input

Provide more input method than textbox, such as compobox and listbox. This is to provide more effective way of keying data.

8.4.3 More system templates

More system template is builds for the future version so that it can provide user with more selection of system interfaces.

8.4.4 System Design

User will able to change the design of the system preview and interfaces, such as changing the system button style, change the system color and so on.

8.5 Knowledge And Experience Gained.

8.5.1 The importance of all phase in SDLC

System analysis is an importance phase in the system development life cycle (SDLC). This phase capture the user requirement and the goal of the system. It this phase is wrong defined. A lot of problems and errors will occurred and cause faulty to the system development. System will fulfill all the requirements and achieve its goal if the system analysis is able to complete successfully.

8.5.2 Development tools knowledge

The project is using the VB.NET, Microsoft Visual Basic.NET and Microsoft SQL server as development tools. By applied all them in the application, it's able to improve the understanding about the language themselves as well as their integration. I also improve my skill in programming language and knowledge about component based development.

8.5.3 Self expression

Developing E-Manufacturing system really given me a greater change to express myself in design and coding of the system. I have the chance to build application software by myself. Doing this project has greatly improved my self esteem and self confidence. I also was being exposed to real system development environment. This helps me learn how to management a project in time and resource and how to corporate in develop a system. I am setting experience on how to set up and configure various technologies to be able to serve as a live system.

8.6 Reviews on Goal

At final stage of the project, there were certain expectations on what would be achieved; the following is the expectation that achieved.

Expectation achieved

In overall, the system had fulfilled the expectations stated by the project. The basic foundation of the system was designed and implemented. It was eligible for future growth and implementation. The E-Manufacturing framework met the criteria like reliability, user friendly, and expandability.

Objective achieved

The project has successfully created a system that supported the information need for a material management system for E-Manufacturing. It could be concluded that the objective to establish the application had been achieved.

8.7 Chapter Summary

Overall, the material management for E-Manufacturing has achieved and fulfills its objective and functional requirements. The feasibility of the system depends on how much the organization will benefits form its implementation.

This project gives me opportunity to build an application from scratch. Here, theories and knowledge gained throughout the course of Software engineering were literally put into practice. A lot of programming knowledge was gained. This project has been a very useful experience which exposes the idea of research work to me.

Finally, there are much more rooms for improvement in this system, enhancement still could be made and more features and functions can be added for future version

Reference

Christianasson B. (1999) *Komponentbaserad systemutveckling genvag eller senvag?* In Nilsson, A. G. & Pettersson, J. S. (eds.) *Om metoder for systemutveckling i professionella organisationer Karlstadskolans syn pa informatikens roll i samhallet,* Studentlitteratur, Lund, (in Swedish)

Christiansson, B. *et al*, Component – Based Software Development Life Cycles

Coss, (2003) [online].
Available WWW<http://www.coss-system.com>

Edelstein, Herb. "Unraveling Client/Server Architecture." *DBMS 7,* 5 (May 1994): 34(7)

Evans, J. R. (1991). *Statistical Process Control for Quality Improvement.*
Prentice-Hall, Inc.

Fah, L. L. (2002) *Electronic Sales and Electronic Marketing.*
Degree Thesis, University Malaya

Fayad, M. E., *et al* (1999). *Implementation Application Frameworks.*
Wiley Computer Publishing.

Freeze, W. S. (2000). *Visual Basic Developer's Guide to COM and COM+.*
SYBEX Inc.

Grimes, R. (1997). *Professional DCOM Programming.*
Wrox Press Ltd.

henningsoftware (2003) [online].
Available WWW<http://www.henningsoftware. com>

Herzum, P., Sims, O. (1999). *Business Component Factory.*
Wiley Computer Publishing

Intergraph (2003) [online].
Available WWW<http://www.intergraph.com/ppo>

Jacobson, I. *et al*, (1999) *The Unified Modeling Language User Guide*,
Addison Wesley

Java™ 2 Platform Enterprise Edition Specification, v1.4 (2003).
Proposed Final Draft 3.
Sun Microsystems, Inc.[Online].
Available WWW <http://java.sun.com/webapps/download/Display>

Kok, C. E. (2001) *Integrated Hospital System Emergency Care Services System.*
Degree Thesis, University Malaya

Leon, A. (2000) *ERP Demystified*
Tata McGraw-Hill Publishing Company Limited

Louis [online].
Available WWW <URL: http://www.softis.is/> (1995).

McClellan, M. (2001). *Introduction to Manufacturing Execution Systems.* **3**. 1-12.

ncst (2001) *Introduction to Component Based Software Development* [online].
Available WWW <URL:
http://www.ncst.ernet.in/education/apgdst/comfac/lecture/Introduction%20to%20CBSD.pdf/> (1995).

Pfleeger, S. L. (2001). *Software Engineering Theory and Practice.* 2nd Ed.
Prentice Hall International, Inc.

Reynolds, M., Willis, T., Blair, R.,and Crossland, J.(2002) *Beginning VB.NET.* 2nd ed. Wrox.

RHMMS (2002) [online].
Available WWW<https://www.denix.osd.mil>

Schussel, George. *Client/Server Past, Present, and Future* [online].
Available WWW<URL: http://www.dciexpo.com/geos/> (1995).

Scott, K. (2001). *UML Explained.*
Addison-Wesley.

Sommerville, I. (2001). *Software Engineering.* 6th Ed.
Addison-Wesley Publisher Limited.

Toomey, J. W. (1996). *MRP II - Planning for Manufacturing Excellence.*
International Thomson Publishing.

Unix (2001) [online].
Available WWW<http://www.hu.mtu.edu/~awysocki/courses/ HU101-F99/joel2/page3.html>

Vaughn T. (1990) *Issues in Standards for Reusable Software Components. (November, 1996)* [online].
Available WWW <http://ruff.cs.umbc.edu>

Wetherill, G. B. *et al* (1991). *Statistic Process Control, theory and practice.*
Chapman and Hall

Whitten, J. L., *et al* (2002). *Systems Analysis and Design Methods.* 5th Ed.
The McGraw-Hill Companies, Inc.

Williams, S. *et al The Component Object Model: A Technical Overview* [online].
Avaible WWW <URL: http://msdn.microsoft.com/library/default.asp?Url=/library/en-us/dncomg/html/msdn_comppr.asp>

Yee, C. P. (2001). *Enterprise Resource Planning Inventory Control System.* Degree Thesis. University of Malaya

[1.1] http://www.cheshirehenbury.com/emanufacturing/eman.html

[2.1] http://business.cisco.com/prod/tree.taf%3Fasset_id=57815&MagID=57722 &public_view=true&kbns=1.html

[2.2] http://www.cheshirehenbury.com/emanufacturing/eman.html

[2.3] http://www.acs.co.kr

[2.4] http://www.developer.com/net/vb/article.php/1540311

[2.5] http://msdn.microsoft.com/library/default.asp? Url=/library/en us/dncomg/html/msdn_comppr.asp