FACULTY OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA

CHUA FONG

WEK010059

e-Procurement for IT Stuff

Under the Supervision of

Assoc. Prof. Dr. Lee Sai Peck

Moderator

Pn. Nazean Jomhari

Session 2003/2004

A project proposal and report in partial fulfillment of the requirement for the Degree of Bachelor of Computer Science University of Malaya

ABSTRACT

All undergraduates of the Faculty of Computer Science and Information Technology (FSKTM), University of Malaya are compulsory to take course entitled "Projek Ilmiah Tahap Akhir" in their 3rd year of study.

Ilmiah Tahap Akhir brings out 9 credit hours of total 108 credit hours. Projek Ilmiah Tahap Akhir consists of 2 component; firstly, WXES3181/WXET3181: Projek Ilmiah Tahap Akhir I (Semester I - 4 credit hours), secondly, WXES3182/WXET3182: Projek Ilmiah Tahap Akhir II (Semester II – 5 credit hours). Content of WXES3181/WXET3181 are mainly bout the explanation of propose project supported by review of literature, paining and design.

The name of the propose system is e-Procurement of IT stuff. This project aims to develop a online purchase order tracking system which is appropriate for the usage of an organization to simplify their purchasing IT stuff processes such as printer, computer desk, mouse and extra. This system consists of 3 main modules. They are the administrator module, client module and vendor module each has their own functions.

Tools planning for system development are in Linux platform, PHP scripting language, JavaScript and DBMS PostgreSQL.

ACKNOWLEDGEMENTS

The project is successfully developed under the assitance and guidance of many parties.

Firstly, i would like to express my sincere gratitude to my project supervisor, Assoc. Prof. Dr. Lee Sai Peck upon her willingness to accept me under her supervision and her invaluable advice.

Secondly, i wish to forward my appreciation to Puan Nazean Johari for being my moderator and not forget Puan Siti Hafizah for her precious advice.

I also wish to thank my fellow friends, Ms. Heng Ai Shi, Ms. Nurul Sufiani Ahmad dan Ms. Cheang Li Fun for their encouragement and their helping hands whenever i faced difficulties. I would like to thank Senior Foo See Lee for her guidance and effective motivation.

Lastly, i would like to forward my appreciation to all the librarians of FSKTM library(Resource Room) for their contribution and quality service.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Overview of Internet Services

The Internet is a global computer network with an estimated 100 million users worldwide, and it is growing like crazy; according to one estimate, more than 300,000 new Web pages appear every seven days, and the total amount of information available on this worldwide network doubles every year. The Internet is not simply a new mass medium, it is the first mass medium that involves computers and uses digitized data. For this reason, it is more interactive than television, radio, and newspapers, which limit user interaction to content selection. With the Internet, people can create information as well as consume it. It is the first truly democratic mass medium.

Traditional media are not going away soon, but we have to count on one thing that the internet is transforming almost everything we do, including communicating, obtaining information, learning, exchange knowledge, looking for jobs, keeping up with a career or a profession, and even falling in love. In addition, it is equally indispensable for businesses; according to one estimate, electronic commerce, facilitated by the Internet, will create a market worth billions in coming years.

As the ever-changing business scenario has become more and more dependant on the Internet for any data transactions or for communication between applications. As a result, the focus of software development is shifting from desktop applications to applications that can access data through the Internet. These applications are mainly *distributed applications*—scalable applications in which data are share across applications.

Distributed applications consist of a client application, which has interactions with a middleware application that contain the business logic for entire business solution that has been created. This intermediate application in turn interacts with the underlying databases that store the data for the application. Therefore, a business solution comprises a number of applications and database. They may be present on a single computer, but in large-scale business operations, they generally are distributed across different computers connected over a network. In that case, these applications may created by using different programming languages and, even for certain situation, on different platform. It is paramount that all these applications are integrated in order to build a complete business. This integration is made simpler with the usage of Internet services.

1.2 Project Introduction

e-Procurement for IT Stuff is an online purchasing order tracking system appropriate for a middle or huge organization to simplify the process of purchasing IT stuff such as printer, computer table, mouse and extra. It processes and tracks all purchase orders manually. User can easily do purchasing application within a short time and it is faster than the normal manual system. This new system actually reduce time that been spend to complete an application that was required by every department in the organization.

1.3 Project Objective

Before starting system development, demonstration of project objective is necessary. The core objective of this project are as the following:

- Standardize and shorten the process of purchase order

- Promote paperless environment and fast data retrieval via internet services
- To develop a user friendly system
- Provide simple and interactive graphical user interface
- Decrease the time waste cause by technical problems (e.g. phone line engaged)

1.4 Project Scope

This system involved four main modules, which are super-administrator, administrator, client and vendor. Each of these modules has very close relationship in the procurement processes.

1.4.1 Super-administrator (super-admin)

- *Super-admin Module*: Super-admin can perform function such as add or delete system administrator. He/She may monitor the staff-in-charge of each application (purchase order) through the system. Furthermore, he/she can distribute task to administrator. Super-administrator can also perform basic function like change login password.

1.4.2 Administrator (admin)

- *Administrator Module*: Admin have the authority to add or delete client and vendor from the list. Admin can view each purchase order and the related vendor's record and offering price. Besides, admin shall keep track of each application and update the related information of each application. Administrator can also perform basic function like change login password.

1.4.3 Applicant

- *Applicant Module*: Applicant consists of staff-in-charge of each department that make purchase orders through the system. They also can view their application status through the system. Furthermore, user can also perform basic function like update and change login password.

1.4.4 Vendor

- *Vendor Module*: Vendors consist of IT vendor who has relationship with the organization. The system will allow vendor to view recent application and each item's specification. Vendor can perform basic function like Update Company's information and change login password.

1.5 Significance of Project

There is much to be gained and achieved by developing this system. It provides ways to keep track of all the processes of purchasing IT stuff, information of all the system users, and improve and provide services needed easier and in a short time.

a) Improvement of Services Quality

One of the main purposes of this project is to provide services for users. By applying web service in the system, make it possible to perform as required by the users. Ensuring users that availability of the latest information is met and reducing time spent in searching for it, makes the overall process simple and short.

b) User-friendly Application

One of the essential component to make a system successful is the system must be user-friendly to ensure that users are comfortable when using it. Therefore,

IT Stuff Procurement System, as an inventory system will be design to met this requirement.

c) Foster Better Working Environment

With *IT Stuff Procurement System*, the system users will safe a lot of time due to availability of this system. With just clicking on the mouse, they will be provided functions required without mingle up with plenty of processes, which are time consuming. As a result, it increases productivity and fosters a quality work-life-environment.

1.6 Expected Outcome

The summarization of the expected outcome of *IT Stuff Procurement System*:

- Simple and user friendly system
- A fully develop system that can fulfill all the requirements specified
- Effective and efficient way to store and achieve relevant information

1.7 Project Limitations

i. Inexperience skill

Since there is insufficient knowledge of PHP and Linux there would be limitation especially in personal skill when using them to develop the whole system.

ii. Time

Much time needed to analyze and study on the development of this online system and the tools use to develop the system.

iii. Resources

Resources such as coding, command and information about the tools can only be obtained through the Internet because reference books are too expensive. So much of the resources are limit to the Internet only.

iv. Human Resources

Actually, this project requires more than two people to complete it, as the scope is large. This limitation will cause the system to be developed in a rush in order to build it on time.

1.8 Project Schedule

The schedule of the project will be shown on Figure 1.1. Overall, this project starts on 23rd July 2003 till 17th February 2004.

	Task Name	Duration	Start	Finish
1	1 Preliminary Study and Planning	14 days	Wed 7/23/03	Tue 8/5/03
2	2 Literature Study	21 days	Wed 8/6/03	Tue 8/26/03
3	3 System Analysis	28 days	Wed 8/27/03	Tue 9/23/03
4	4 System Design	21 days	Thu 9/18/03	Wed 10/8/03
5	5 Development and Coding	114 days	Thu 10/9/03	Fri 1/30/04
6	6 Unit Testing	14 days	Thu 1/15/04	Wed 1/28/04
7	7 System Testing	21 days	Sun 1/18/04	Sat 2/7/04
8	8 Documentation	142 days	Mon 9/29/03	Tue 2/17/04

Figure 1.1: Project Schedule

1.9 Report Organization

This section contains brief introductory information of every chapter in this report in order to ensure a better understanding among readers about this project.

Chapter 1, "**Introduction**", a brief introduction to the whole project including the project overview, project motivation, project objective, project scope, significance of project and project schedule to show the timeline.

Chapter 2, "**Literature Review**", studies on the events and the problems that will occur before the project begins. In this chapter, also reviews the tools and languages that could be use to develop web services in an organization. Besides, this chapter will cover literature survey on existing or similar system, web services architecture.

Chapter 3, "**Methodology**", discuss the methodology use for system development, and techniques applicable in problems solving.

Chapter 4, "**System Analysis**", discussion on the system requirements such as functional requirements and non-functional requirements. Explanation of the consideration put in choosing various tools and programming language will be included.

Chapter 5, "**System Design**", provides details regarding the system design, database design and user interface design.

Chapter 2: LITERATURE REVIEW

2.1 Introduction

Literature review forms a paramount process in system development. System comparison, analysis and facts finding can be done in order to achieve better understanding of the system, which will be developed. Researches and evaluation can be done to available system or similar system for the purpose to have a closer view and clearer picture of the system before proceeding on system development.

2.2 Web Services

Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks.

2.2.1 Web Services Architecture

A web service can be an intermediate application that allows a Web service client application to access data from an underlying database. As shown in the figure 2.1, web service architecture internally consists of four layers.

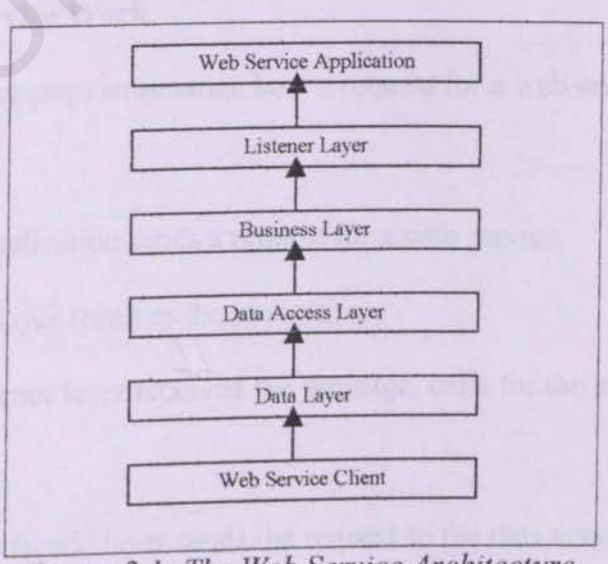


Figure 2.1: The Web Service Architecture

i) Data Layer

The data layer is the first layer in the web service architecture. This layer contains the data that the web client application needs to access.

ii) Data Access Layer

This layer contains the business logic or code that allows the web client application to access the data in the data layer. A part from storing data, data access layer is use to ensure that the data in the data layer is secure.

iii) Business Layer

Business layer consists of business logic and business facade layers. This layer contains the code required for implementing the web service.

Business logic layer: contains all the services provided in a web service.

Business facade layer: acts as an interface of the web service.

iv) Listener Layer

It receives the requests sent by users for a web service then parses the received messages and dispatches the request to the appropriate method in the business facade layer.

2.2.2 How Web Service Work

The following steps summarize how a request for a web service from a user is receive and process:

1. The client application sends a request for a web service.
2. The listener layer receives the request.
3. After the listener layer received the message, calls for the business façade layer.
4. The business façade layer sends the request to the data access layer.

5. The data access layer applies the business logic.
6. After applying the business logic, the data access layer sends the message back to the business facade layer.
7. The business façade layer sends the message to the listener layer.
8. The listener layer sends the data to the web services client.

2.3 Internet Research – Analysis on Similiar System

Research is done by surfing the Internet to search for existing management systems, web services system and latest technologies of software development tools. Using Internet search engines such as Yahoo, Google, MSN, and Lycos does it. There is much useful information, which is closely related to this project planning.

2.3.1 ACT! (Interact Commerce Corporation)

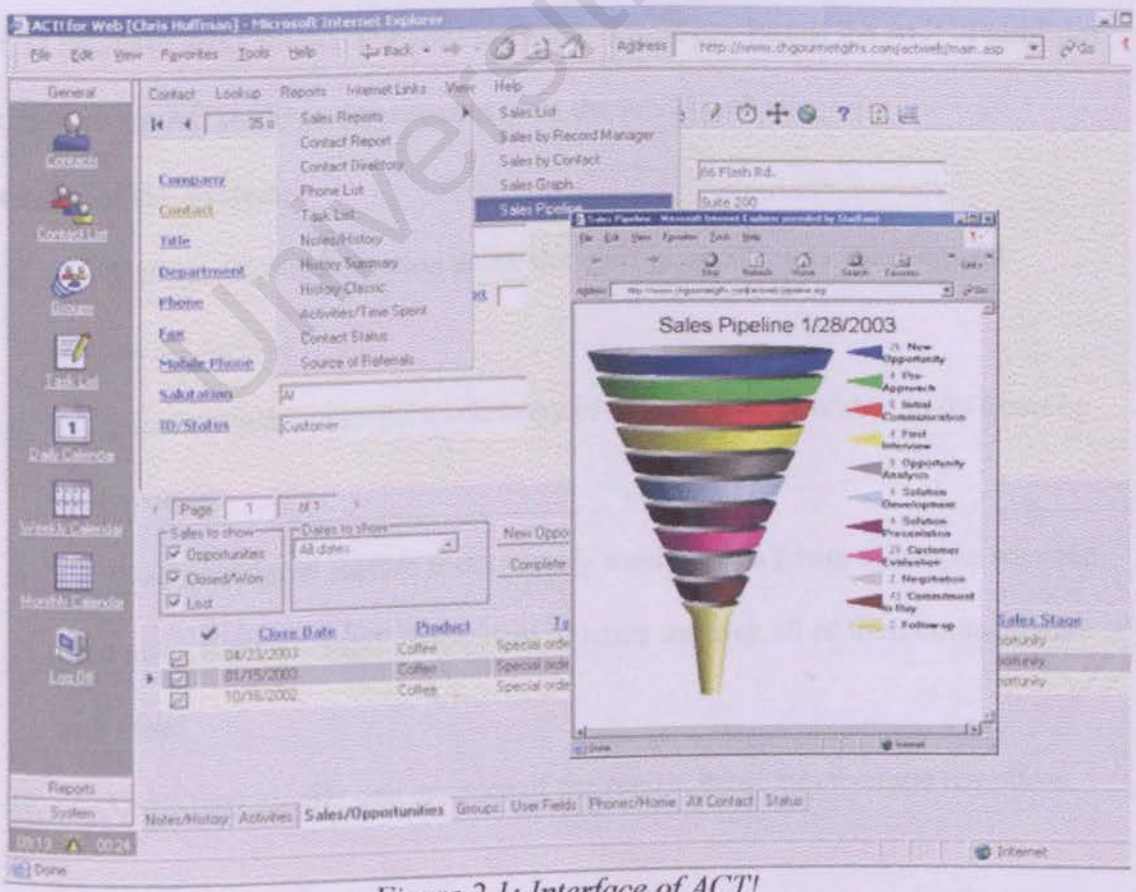


Figure 2.1: Interface of ACT!

ACT! 2000 is Sales Automation Software that automates business tasks such as inventory control, sales processing, and tracking of customer interactions, as well as analyzing sales forecasts and performance. Business may have a custom version developed specifically for their needs, or choose from among the increasing number of sales automation software products.

Dale Carnegie Training, the leader in business training solutions, can help users move closer to their sales goals. Dale Carnegie Sales Advantage, a proprietary selling process, defines the eleven stages in the Sales Development Cycle. Users can access the Dale Carnegie Training web site from within ACT! to learn more about the Sales Development Cycle.

User can link data with ACT! databases on Handheld Personal Computers (HPCs) running Windows CE version 1.0 or 2.0 (640x240 screen). If users use a Palm Computing device, users can link data between their Palm and their desktop ACT! database using the ACT! PalmPilot Link. Users can install the ACT! PalmPilot Link from the ACT! CD or download it from the ACT! web site.

Benefits of the System:

i. Manage sales information using the built in opportunity tracking and forecasting tools.

ii. Filter the task list to get a summary of calls, meetings and/or to do items by priority level.

iii. Synchronization feature is commonly used when a group of people who are often away from a central office want to share some or all of their contact data with other.

iv. Users may modify one or more of the report, label, or envelope templates.

v. View daily, weekly, and monthly calendars for individuals, selected users or the entire workgroup.

vi. Track up to 255 fields of customer information, creating instant access to every relationship.

vii. User can see all the activities that they have scheduled with all contact in the group in a single location.

Weaknesses of the System:

i. Restriction of data integration that limited database type for exporting or importing to or from others database's data.

ii. ACT! provide limited functionality for business planning and sales analysis.

2.3.2 sqlDESKTOP Management

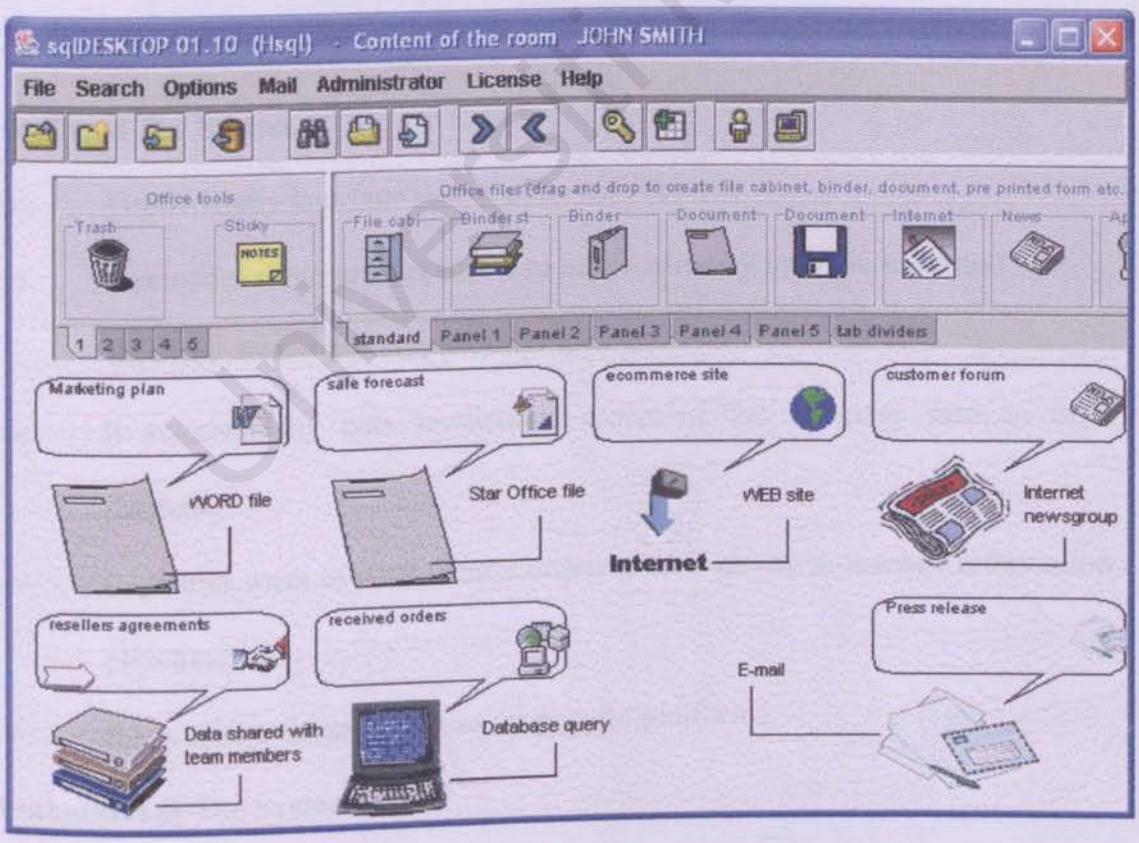


Figure 2.2: sqlDESKTOP Management

As figure 2.2 above, sqlDESKTOP is a icon-based interface oriented system. It is a computerized manual filing system arranged just like an ordinary office with pictures of real cabinets and real folders. It allows users to organize information they access regularly : files, multimedia data on CD ROM or DVD, WEB pages, database queries, emails, newsgroups and extra using familiar, everyday office supplies (folders, binders, filing cabinets, etc.). Easy to use, sqlDESKTOP makes extensive use of graphics, using pictures of real cabinets and real folders. With sqlDESKTOP, users can clip together documents related to the same topic as user would with normal office documents. Even if the documents were not created by the same program. Or if some are stored on hard disk and others on CD ROM or Internet sites. Documents are easy to find because user don't have to remember the exact name of the document or where they put it. Visually locate what user are looking for in a familiar office environment and click on it.

Benefits of The System

i. User-friendly interface that is understandable.

ii. It provides simultaneous access to information by multiple users and departments to minimize processing delay.

iii. It automatically puts intellectual assets of the company into an SQL database.

iv. It enables users to permanently improve their ability to manage information effectively.

v. It is a Java software that runs on popular platforms.

Weaknesses of The System

i. User need to training to understand the system and it takes time for them to familiar with the system.

ii. The interface is colorful, may causes confusion.

iii. It supports only a few latest version databases.

iv. It does not compatible with operating system other than Windows and Linux.

2.4 Reviews on the Latest Technology

2.4.1 Operating System

Operating system is much like a computerized version of a traffic cop, standing at the intersection of the computer's hardware, application programs and the user. To be specific, it manages programs, parcel out memory, deals with input/output devices and provides a means of interacting with the user.

2.4.1.1 Unix

Developed at AT&T's Bell Laboratories in the early 1970s. It begins as an open source project that became widely used in Universities, scientific labs, and by the U.S. government. Unix was the first operating system with preemptive multitasking, and it was designed to work efficiently in a secure, centrally administered computer network. Other important UNIX innovations include the concepts of file directories and path name. It also supports multiprocessing making it ideal for use with high-powered minicomputers equipped with several CPUs. Unix is not widely supported because difficulty of use. Unix defaults to a command-line user interface, which is challenging for beginner users.

2.4.1.2 Microsoft Window 2000

Windows 2000 is a multipurpose operating system with integrated support for client-server and peer-peer networks. The Window 2000 has been designed to increase reliability, deliver higher levels of systems availability, and provide for scalability from a small network to a large enterprise network.

Compared to Windows 98 Second Edition and Windows NT, Window 2000 is more stable. Moreover, it can perform much faster than other Microsoft windows operating systems. Because of its stability, it seldom causes system crash and less reboot.

2.4.1.3 Linux

Linux is a multitasking, multi-user operating system, which means that many people can run many different applications on one computer at the same time. This differs from MS-DOS, where only one person can use the system at any one time. Under Linux, to identify yourself to the system, you must log in, which entails entering your login name (the name the system uses to identify you), and entering your password, which is your personal key for logging in to your account. Because only you know your password, no one else can log in to the system under your user name.

Linux is an operating system kernel that behaves and performs similarly to the famous UNIX" operating system from AT&T Bell Labs. It has all of the features of a modern operating system: true multitasking, threads, virtual memory, shared libraries, demand loading, shared, copy-on-write executables, proper memory management, loadable device driver modules, video frame buffering, and TCP/IP networking.

Linux was written originally for Intel processor based PC's, using the hardware facilities of the 80386 processor and its successors to implement its features. The 80386 family includes the 80486, and all of the Pentium chips. However, there are now many ports to other hardware platforms.

Linux supports many file systems, including the standard MS-DOS partitioning scheme, so it can share disk with other operating systems. Linux supports all known versions of the Microsoft FAT and VFAT file systems, including those used by Windows 95, Windows 98, Windows NT, Windows 2000 and windows ME through loadable kernel modules. In a correctly configured system, they should load automatically when the partitions are mounted.

2.4.2 Web Server

A web server is a program that waits patiently for browsers to request a web page. Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks. When the server receives a request for a specific resource, it looks for the requested file and sends it to the browser. If the file is not found, the server sends an error message. Every computer on the Internet that contains a web site must have a web server program.

2.4.2.1 Microsoft's Internet Information Services (IIS)

IIS is a group of Internet servers (including a web or Hypertext Protocol server and a File Transfer Protocol server) with additional capabilities for Microsoft's Window NT and Windows 2000 Server operating systems.

IIS contains many new features along with performance and reliability enhancements. With IIS, Microsoft includes a set of programs for building and administering web sites, a search engine, and support for writing web-based applications that access databases. IIS is tightly integrated with the Windows NT and 2000 Servers resulting in faster Web page serving.

IIS is user-friendly because of easiness of configuration and can be used alone as a web server. It inherits Window NT features. IIS can help administer secure websites. A part of that, IIS support a variety of applications such as Common Gateway Interface (CGI), Active Server Page (ASP), connection to ODBC and Secure Socket Layer.

2.4.2.2 Apache

Often referred to as simply Apache, a public-domain open source Web server developed by a loosely knit group of programmers. The first version of Apache, based on the NCSA httpd Web server, was developed in 1995. Because it was developed from existing NCSA code plus various patches, it was called a patchy server - hence the name Apache Server.

The original version of Apache was written for UNIX, but there are now versions that run under OS/2, Windows and other platforms. The Apache httpd server is a powerful, flexible, HTTP/1.1 compliant web server implements the latest protocols, including HTTP/1.1 is highly configurable and extensible with third-party modules can be customized by writing 'modules' using the Apache module API provides full source code and comes with an unrestrictive license runs on Windows NT/9x, Netware 5.x and above, OS/2, and most versions of Unix, as well as several other operating systems.

2.4.2.3 Personal Web Server (PWS)

The Personal Web Server is ideal for intranets, homes, schools, small business workgroups and anyone who wants to set up a personal Web server. PWS is an entry-level/mid-range server for Windows 9x/ME/NT platforms. Users of Windows 9x/ME/NT can now share Web content as easily as they share folders on a network.

It is also a great cost-saving solution for small businesses and schools because they no longer need to dedicate a PC to a Web server. Instead, they can leverage their existing PCs simply by enhancing them with the Personal Web Server software.

2.4.3 Data Access Technology

In the early days of databases, developers needed an intimate knowledge only of the database product they were using. Nevertheless, database products and technologies have advanced quickly. From relational databases to non-relational data stores such as email and file systems, data-access technologies have had to keep up with rapid technology changes. With the advent of client/server and multi-tier application architectures, developers must now understand a variety of data-access technologies.

A few of the data-access technologies are reviewed and consideration is making to enable the communication and access to the database.

2.4.3.1 ODBC (Open Database Connectivity)

ODBC technology provides a common interface for accessing heterogeneous SQL databases. This interface provides maximum interoperability: One application can access different SQL database management systems (DBMSs) through a common

set of code. Thus, a developer can build and distribute a client/server application without targeting a specific DBMS.

Database applications call functions in the ODBC interface, which are implemented in database specific modules called drivers. The use of drivers isolates applications from database specific calls in the same way that printer drivers isolate word processing programs from printer specific commands. Because drivers are loaded at run time, a user only has to add a new driver to access a new DBMS, it is not necessary to recompile or relink the application.

2.4.3.2 JDBC (Java Database Connectivity)

JDBC is a Java API enables Java programs to execute SQL statements. This allows Java programs to interact with any SQL-compliant database. Since nearly all relational database management systems (DBMSs) support SQL, and Java runs on most platforms, JDBC makes it possible to write a single database application that can run on different platforms and interact with different DBMSs. JDBC is similar to ODBC, but is designed specially for Java programs, whereas ODBC is language-independent. Java Soft developed JDBC.

The JDBC API allows developers to take advantages of the Java platform's "Write Once, Run Anywhere" capabilities for industrial strength, cross-platform applications that require access to enterprise data. With a JDBC technology-enabled driver, a developer can easily connect all corporate data even in a heterogeneous environment.

2.4.3.3 Activex Data Object (ADO)

ADO is an application program interface introduced by Microsoft that lets a programmer writing Windows applications get access to a relational or non-relational database from both Microsoft and other database providers. ADO takes the objects based in DAO (Data Access Object) and RDO (Remote Data Object) and provides a much simpler object model than DAO and RDO. DAO and RDO are designed only for accessing relational databases. ADO contains several built-in objects that simplify the task of accessing data from data stores.

As Microsoft's other system interface, ADO is an object-oriented programming interface. It is also part of an overall data access strategy from Microsoft called Universal Data Access.

2.4.3.4 OLE DB

OLE DB builds on ODBC and extends the technology to a component architecture that delivers higher-level data-access interfaces. This architecture provides consistent access to SQL, non-SQL, and unstructured data sources across the enterprise and the Internet. (In fact, for access to SQL-based data, OLE DB still uses ODBC because it is the most optimized architecture for working with SQL.) OLE DB consists of three components: the data consumer (e.g., an application); the data provider, which contains and exposes data; and the service component, which processes and transports data (e.g., query processors, cursor engines). OLE DB is one API that operates against SQL data sources and non-SQL data sources such as mail and directories.

2.4.3.5 ADO .Net

Microsoft designed ADO.NET based on its experience with its successful ADO object model. But ADO.NET addresses three important needs that ADO doesn't address: providing a disconnected data-access model, which is crucial to the Web environment; providing tight integration with XML; and providing seamless integration with the .NET Framework.

2.4.4 Database Server

Database Server is a place to store structured collection of data. A database is any collection of information stored in an organized way and it exist before computers. Database stored information so that you can quickly locate, organize, and display information you need while keeping unwanted information out of your way. Database management plays a central role in computing, as stand-alone utilities, or as parts of other applications.

2.4.4.1 Microsoft SQL Server 2000

SQL Server 2000 is one of Microsoft's most popular database servers. It provides agility in data management and analysis, allowing any organization to adapt quickly and gracefully to derive competitive advantage on a fast-changing environment. It is designed to meet requirement of a distributed client-server environment.

The SQL Server driver enables application to access data in Microsoft SQL Server database through the Open Database Connectivity (ODBC) interface. Structure Query Language (SQL) is used to access data in a SQL Server database. All

the client workstations communicate with SQL Server across a star network with TCP/IP protocol. It is fully web-enabled database product, providing core support for Extensible Markup Language (XML) and the ability to query across the Internet and beyond the firewall.

SQL Sever 2000 extends the performance, reliability, quality, and ease-of-use of Microsoft SQL Server version 7.0. Microsoft SQL Server 2000 includes several new features that make it an excellent database platform for large-scale online transaction processing (OLTP), data warehousing, and e-commerce applications. OLTP is now called SQL Server 2000 Analysis Services, which is also, includes a new data-mining component.

2.4.4.2 Oracle

Based in redwood, California, Oracle Corporation is the largest software company whose primary business are database products. Historically, Oracle has targeted high-ended workstations and minicomputers as the server platforms to run its database systems. Its relational database was the first to support the SQL language, which has since become the industry standard.

Oracle server is a multi-user relational database management system (DBMS) that runs on numerous operating systems. Oracle 8.0, the worlds most powerful object relational database is the heart of the open, standards-based Network Computing Architecture. Network Computing Architecture allows IT organization to spend less time struggling with interoperability issues and more time focusing on deploying solutions.

Oracle 8.0 includes a fully integrated set of easy-to-use management tools, full distribution, and replication and web features. Oracle 8.0 also provides the highest

levels of availability through fast fail over, easier management, and zero data loss disaster protection, with Data Guard, the only complete data protection solution available on the market.

Oracle 8.0 delivers the entire major platform requires by network-based architecture which involve multiple hardware and software platforms, this includes UNIX, Linux and Windows platforms. However, it is expensive and separate licenses are required for each of its database engine. Oracle 8.0 data management, security, reliability, and ease of use, is unique designed to meet the demands of the network era. For mainframe system, parallel server's environments, or desktops, Oracle 8.0 is the database of choice.

2.4.4.3 PostgreSQL

PostgreSQL is an enhancement of the POSTGRES database management system, a next-generation DBMS research prototype. While PostgreSQL retains the powerful data model and rich data types of POSTGRES, it replaces the PostQuel query language with an extended subset of SQL. PostgreSQL is free and the complete source is available.

2.4.4.4 MySQL

MySQL is an open source relational database management system (RDBMS) that uses Structured Query Language (SQL), the most popular language for adding, accessing, and processing data in a database. Open Source means that it is possible for anyone to use and modify. Anybody can download MySQL from the Internet and use it without paying anything. Anything can study the source code and change it to fit their needs. MySQL is noted mainly for its speed, reliability, and flexibility. Most

agree, however, that it works best when managing content and not executing transaction.

MySQL stores data in separate tables rather than putting all the data in one big storeroom. User can define relations to link the tables, this will making it possible to combine data from several tables on request.

MySQL is a small, compact, easy-to-use database server, ideal for small and medium sized applications. Client/server implementation consists of a server and many different client programs. It is available on a variety of UNIX platforms, Linux, Windows NT, Windows 9x and Windows 2000.

2.4.5 Software Architecture

There are a few software architectures such as mainframe architecture, client-server architecture, file-sharing architecture, two-tier architecture, three-tier architecture, and Windows Distributed Internet architecture.

2.4.5.1 Mainframe Architecture

In mainframe system architecture, all operation is within the central host computer. User interacts with the host through a terminal that captures keystroke and sends that information to the host. Mainframe architecture is not tied to a hardware platform. User can interact with the host using PCs, and UNIX workstations. A limitation of mainframe architecture is that it does not easily supports graphical user interface or accesses to multiple databases from graphical dispersed sites. In the last few years, mainframes have found a new use as a server in distributed client/server architecture.

2.4.5.2 Client-server Architecture

Client-server is a network architecture in which each computer or process on the network is either a client or a server. Client-server architecture implies a cooperative processing of requests submitted by a client, or requester, to the server, which processes the request and returns the results to the client. The client manipulates the data and presents the result to the user.

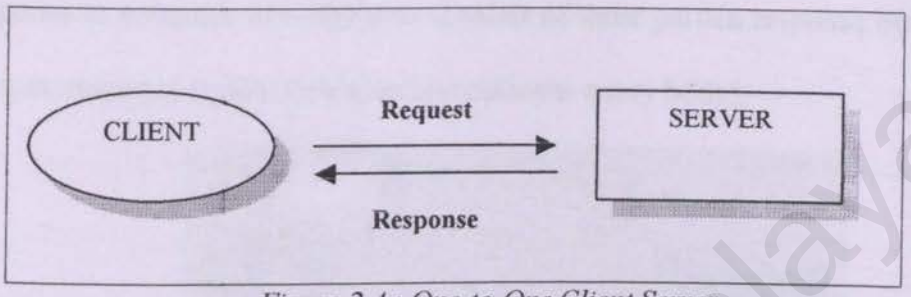


Figure 2.4 : One-to-One Client Server

Client-server solutions can be in a many-to-one design that is more than one client typically makes requests of the server.

2.4.5.3 File Sharing Architecture

The original PC network were based on file sharing architecture, where the server downloads files from the shared location to the desktop environment. File sharing architecture is centralized on a server to download files to several PCs or upload files from several PCs to the server. The requested user job is then run (including logic and data) in the desktop environment. File sharing architecture works if shared usage is low, update contention is low, and the volume of data to be transfered is low. The setback is the shared usage and the volume of the data to be transferred is low to ensure that it is successful.

2.4.5.4 Two-tier Architecture

Two-tier architecture refers to client/server architecture in which the user interface runs on the client and the database is stored on the server. The basic client/server model is a two-tier model because it has only one client and one server. A typical request message from a client to a server consists of three major parts(a request line, optional request headers, and an optional entity body). A server's response to a request message also consists of three parts(a response header line, one or more response header fields, and an optional entity body).

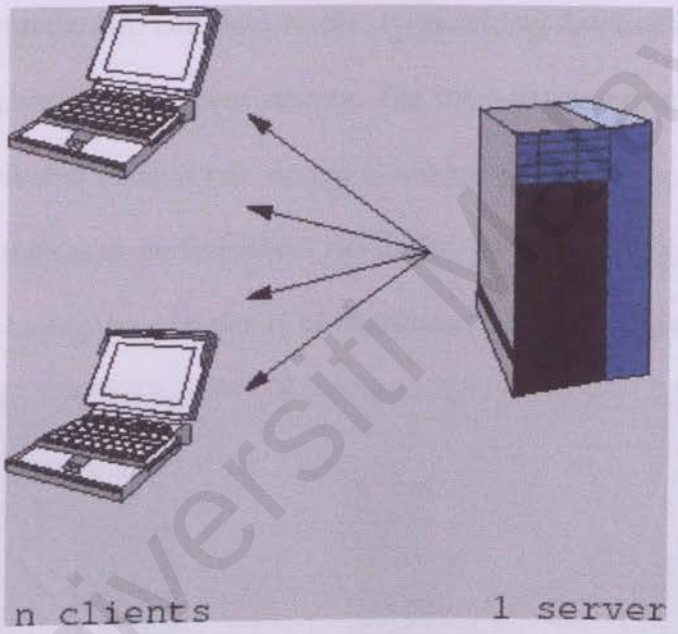


Figure 2.5 : Two-tier architecture

By two-tier architectures, the user system interface is usually located in the user's desktop environment and the database management services are usually in a server that is a more powerful machine that services many clients.

2.4.5.5 Three-tier Architecture

Due to the increasing performance needed in Distributed Computing, two tier architectures become more and more outdated by the huge load that each client

directly talking to the final server would cause. Intervening servers (and therefore on a logical layer a middle tier) enlargen the architecture to handle the task.

The three-tier software architecture (also known as three layer architecture or multi-tier architecture as the middle tier may consist of several tiers by itself) emerged in the 1990s to overcome the limitations of the two-tier architecture. 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 and can accommodate hundreds of users (as compared to only 100 users with the two tier architecture) by providing functions such as queuing, application execution, and database staging. The three-tier architecture is used when an effective distributed client/server design is needed that provides (when compared to the two tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user.

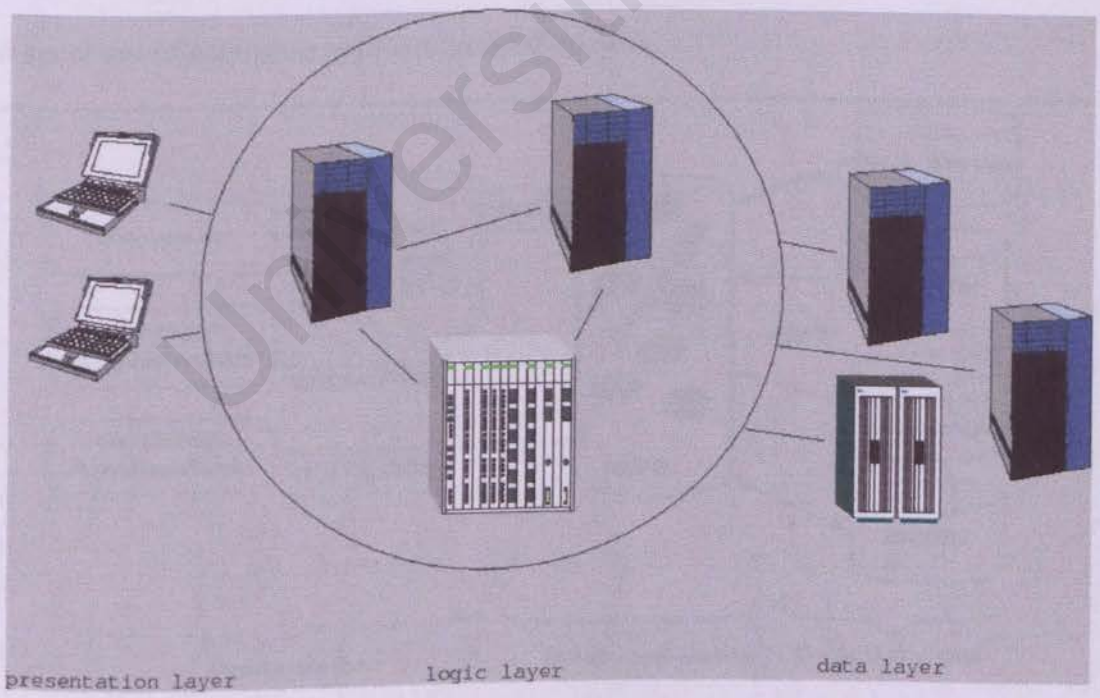


Figure 2.6 :Three-tier architecture

The three-tier consist of : client-tier(presentation layer), application-server-tier(logic layer), and data-server-tier(data layer). Higher-order architecture(those with

more than 3-tiers), are called 'n-tier architectures'. The limitation with three-tier architecture is that the development environment is reportedly more difficult to use than the visually-oriented development of 2-tier applications.

2.4.5.6 Windows DNA Architecture

Microsoft Windows Distributed interNet Applications is an architecture that enables developers to integrate web-based and client/server applications in a single, unified architecturee. Windows DNA is the name given to the combination of tradisional n-tier architecture with the intrinsic Windows 2000 services(including COM+, MSMQ, and Active Directoty).

The technologies within Windows DNA cover both the user's machines(clients) and the machines serving up the data from the Web(serves). It is thus Microsoft's Client-server architecture for the era of the internet, addressing the full spectrum of enterprise application development.

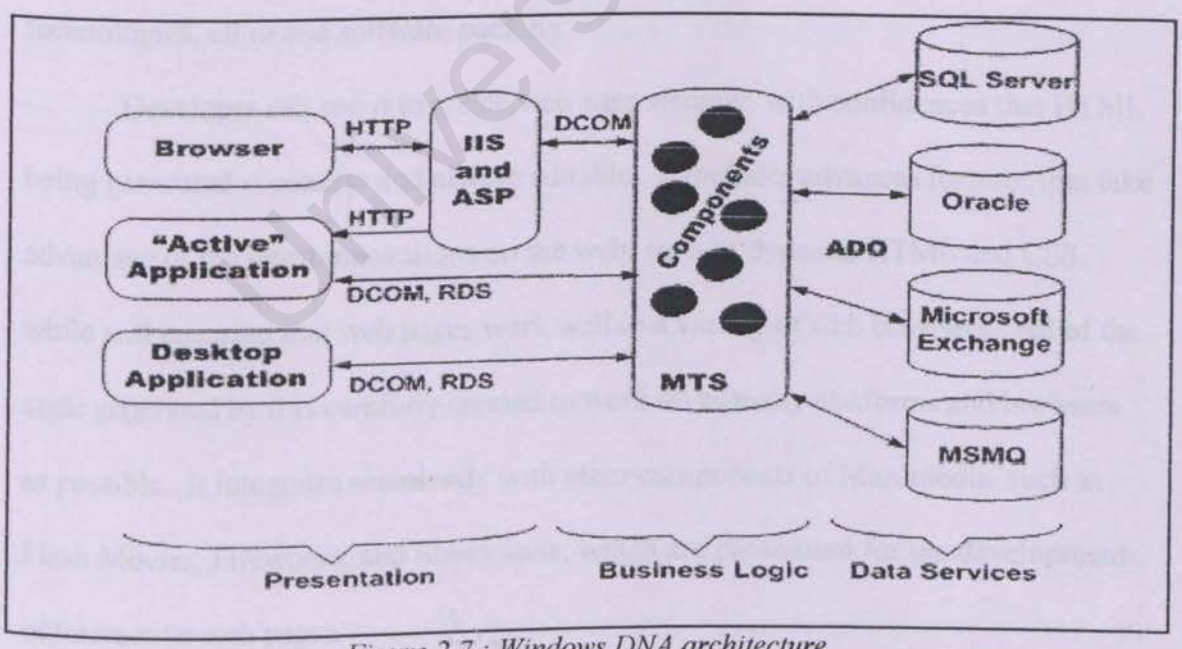


Figure 2.7 : Windows DNA architecture

Because Windows DNA is based on COM and open Internet standards, developers can use any language or tool to create compatible applications. COM

provides a modern, language-independent object model that provides a standard way for applications to interoperate at all tiers of the architecture. Through COM, developers can extend any part of the application via pluggable software components that can be written in C++, Visual Basic®, Java, or other language. Windows DNA supports a broad range of development tools today, including tools from Microsoft, Borland, and many other vendors.

2.4.6 Development Authoring Tools

With the assistance of authoring tools, the process of system development will become more interesting, easier and faster.

2.4.6.1 Macromedia Dreamweaver

Macromedia Dreamweaver gives developers the productivity of a visual web page layout tool, the control of an HTML text editor and support for a new web technologies, all in one software packing.

Developer can use it to create web sites visually, with confidences that HTML being generated is concise and always editable. It includes advances features that take advantage of the latest innovations on the web, such as dynamic HTML and CSS, while still ensuring that web pages work well in a variety of web browsers. All of the code generated by it is carefully created to work on as many platforms and browsers as possible. It integrates seamlessly with other components of Maromedia, such as Flash Movies, Fireworks, and Shockwave, which are paramount for the development of interactive web pages.

2.4.6.2 Macromedia Flash

Macromedia Flash is the professional standard authoring tool for producing high-impact Web experiences. Whether you are creating animated logos, Web site navigation controls, long-form animations, entire Flash Web sites, or Web applications, you will find the power and flexibility of Flash ideal for your own creativity. It can be effectivelly and easily intergrated in a web page to produce a dynamic web page. It is able to reduce contraints often faced by other web technology such as transaction time is long, low quality diplay and many more. Macromedia Flash is widely accepted as a popular interactive multimedia technology and is been widely used.

2.4.6.3 Adobe Photoshop

Graphic is much popular in many of the the information presentation process seems it fasten the process of presenting images and it is capable of giving a clear, consistant and to the point explanation. It allows users to create, modify, manipulate and produce images.

2.4.6.4 Microsoft Visual Studio.Net

Microsoft Visual Studio.net, is a development tool for developing web application and XML web services, desktop application, and mobile applications. It supports multi-language such as C# .net, Jscript .net, C++ .net and VB .net which use the same intergrated development environment(IDE), which allows to share tools and facilities in the creation of mixed-language solutions.

Visual Studio .net is the only development environment built from the ground up for XML Web Serveices. By allowing applications to share data over the internet,

XML web services enable developers to assemble applications from new and existing code, regardless of platform, programming language, or object model.

2.4.7 Development Language

The high progression of computer language form third generation to four generation and until now, it become a popular knowledge to all even to a person who is not an information technology person. Although fourth-genaration languages have given non-programmers useful tools for extracing information from databases, they are not general-purpose programming languages. Programmers kept on using third-generation languages. Below are most powerful and famous computer languages for developing a web-based system.

2.4.7.1 ASP .net

Active Server Page.net, or ASP.net, is Microsoft's latest version of its popular dynamic web programming technology, ASP .net is much more than a simple upgrade from classic ASP.

ASP.net pages execute on the server and generate markup such as HTML, WML, or XML that is sent to a desktop or mobile browser. ASP.net pages use a compiled, event-driven programming model that improves performance and enables the separation of application logic and user interface. ASP.net pages and ASP.net XML Web services files contain server-side logic (as opposed to client-side logic) written in Visual Basic .net, C# .net, or any .net-compatible language. Web applications and XML Web services take advantage of the features of the common language runtime, such as type safety, inheritance, language interoperability, versioning, and integrated security.

ASP.NET includes a set of controls that encapsulate common HTML user interface elements, such as text boxes and drop-down menus. These controls run on the Web server, however, and push their user interface as HTML to the browser. On the server, the controls expose an object-oriented programming model that brings the richness of object-oriented programming to the Web developer. ASP.NET also provides infrastructure services, such as session state management and process recycling, that further reduce the amount of code a developer must write and increase application reliability.

2.4.7.2 Java Server Page (JSP)

JSP allows web developers and designers to rapidly develop and easily maintain, information-rich, and able to support dynamic web pages that leverage existing business system.

JSP is comparable to Microsoft's Active Server Page (ASP) technology. Whereas a Java Server Page calls a Java Program that is executed by the Web Server, an Active Server Page contains a script that is interpreted by a script interpreter (such VBScript or Jscript) before the page is sent to the user.

The following are the popular features of Java Server Page (JSP):

- Rapid and fast web development, deployment, and maintenance
- Simplifying page development with tags
- Platform independent
- Emphasizing reusable components
- Open development and widespread industry support

2.4.7.3 JavaScript

A scripting language developed by Netscape for HTML documents. Scripts are performed after specific user-triggered events. JavaScript can interact with HTML source code, enabling web authors to spice up their sites with dynamic content. Browser such as Netscape and Microsoft supports it, but Internet Explorer supports only a subset, which Microsoft calls Jscript.

Scripting languages are easier and faster to code than the more structured and compiled languages such as C and C++. Even though scripting languages take longer to process than compiled languages, but they are useful for shorter program.

JavaScript is use in web site development to perform actions such as:

- To perform form validation
- Cause text or a graphical image to change during a mouse roll over
- Automatically change a formatted date on a web page
- Cause a linked-to page to appear in a new popup window

2.4.7.4 Hypertext Preprocessor (PHP)

Rasmus Lerdorf created PHP sometime in 1994. During mid 1997, PHP development entered the hands of other contributors. Two of them, Zeev Suraski and Andi Gutmans, rewrote the parser from scratch to create PHP version 3 (PHP3).

Self-referentially short for PHP: Hypertext Preprocessor, an open source, server-side, HTML embedded scripting language used to create dynamic Web pages. In an HTML document, PHP script (PHP has since evolved into a powerful server-side markup language with syntax that resembles a mix between Perl and C.) is enclosed within special PHP tags. Because PHP is embedded within tags, the author can jump between HTML and PHP (similar to ASP and Cold Fusion) instead of

having to rely on heavy amounts of code to output HTML. Seems PHP is executed on the server, the client cannot view the PHP code.

PHP can perform any task that any CGI program can do, but its strength lays in its compatibility with many types of databases. In addition, PHP can talk across networks using IMAP, SNMP, NNTP, POP3, or HTTP.

PHP offers excellent connectivity to most of the common databases (including Oracle, Sybase, MySQL, and many others). PHP also offers integration with various external libraries, which allow the developer to do anything from generating PDF document to parsing XML.

PHP is the natural choice for developers on Linux machines running Apache server software, but runs equally well on any other UNIX or Window platform, with Netscape or Microsoft web server software.

2.4.7.5 Cold Fusion

Cold Fusion is a Macromedia product and created by Allaire Corporation of Cambridge, Mass, that includes a server and a development toolset designed to integrate databases and web pages. Cold Fusion web pages include tags written in Cold Fusion Markup Language (CFML) that simplify integration with database.

Coding for Cold Fusion is much more straightforward and intelligible than VBScript, C++, or Java, providing high levels of functionality. The tags conform to the basic HTML syntax of tag name followed by tag attributes, and are enclosed in the familiar HTML brackets (<>). Most tags are two-sided, and can be combined with each other and with HTML elements to create customs tags for use in Cold Fusion application.

2.4.8 Web Services Technology

2.4.8.1 Extensible Markup Language (XML)

Extensible Markup Language, abbreviated XML, describe a class of data objects called XML documents and partially describe the behavior of computer programs which process them. XML is derived from the Standard Generalized Markup Language (SGML), and can be considered a meta-language: a language for defining markup languages. SGML and XML are text-based formats that provide mechanisms for describing document structures using markup tags (words surrounded by '<' and '>').

XML documents are made-up of storage units called entities, which contain either parsed or unparsed data. Parsed data is made-up of characters, some of which form character data, and some of which form markup. Markup encodes a description of the document's storage layout and logical structure. XML provides a mechanism to impose constraints on the storage layout and logical structure.

2.4.8.2 Web Services Description Language (WSDL)

Web Services Description Language is an XML-based language used to define Web services and describe how to access them. WSDL is XML schemas that describe network services as collection of communication endpoints that are capable of exchanging messages; WDSL service definitions provide documentation for distributed systems and automate the details involved in communications between applications. WSDL is extensible to allow for the description of endpoints and their messages, regardless of what message formats or network protocols are used to communicate.

2.4.8.3 Universal Description, Discovery, and Integration (UDDI)

UDDI is an industry effort started in September of 2000 by Ariba, IBM, Microsoft, and 33 other companies. Today, UDDI has over 200 community members. Like a typical yellow pages directory, UDDI provides a database of businesses searchable by the type of business.

The UDDI, define a set of specifications that will make it easier for business to accelerate the use of business-to-business and commerce over the Internet. UDDI does this by defining how companies can expose their business applications – like commerce, order management, inventory, marketing, and billing – as web services that can be directly and securely defined, discovered and integrated with business applications at trading partners and customers.

The UDDI Project is based on existing Internet standards, is platform and implementation neutral, and has generated considerable momentum. Most essentially, UDDI involves the shared implementation of web services based on the UDDI specifications. This web service, the UDDI Business Registry is an Internet directory of businesses and the applications they have exposes as web services for trading partners and customers to use. Business programs will use the UDDI Business Registry to determine the specifications for programs at other companies in a manner similar to how people use web search engines today to find websites. This automated application-to-application discovery and integration over the internet will help eliminate many of the configuration and compatibility problems that are preventing business from more widely adopting business-to-business, despite business-to-business's potential for cost savings and improved efficiency.

2.4.8.4 Simple Object Access Protocol (SOAP)

SOAP was developed by Microsoft, Develop Mentor, and User land Software and has been proposed as a standard interface to the Internet Engineering task Force (IETF). It is somewhat similar to the Internet Inter-ORB Protocol (IIOP), a protocol that is part of the Common Object Request Broker Architecture (CORBA).

SOAP enables a program running in an operating system (such as Windows98) communicate with a program in the same operating system or of another type of operating system (such as RedHat Linux 8.0) by using the Hypertext Transfer Protocol (HTTP) and its Extensible Markup Language (XML) as the mechanisms for information exchange. Since web protocols are installed and available for the use by all major operating system platforms, HTTP and XML provide an already at-hand solution to the problem of how programs running under different operating system in a network can communicate with each other. SOAP specifies exactly how to encode an HTTP header and an XML file so that a program in one computer can call a program in another computer and pass it information. It also specifies how the called program can return a response.

An advantage of SOAP is that program calls are much more likely to get through firewall servers that screen out requests other than those for known applications (through the designed port mechanism). Since HTTP requests are usually allowed through firewalls, programs using SOAP to communicate can be sure that they can communicate with program anywhere.

2.5 Summary

In this chapter, it can be conclude that any kinds of information that is related to this project are reviewed in order to get better understanding of the system is going

to be develop. A part from that, review has been done on a few similar systems to adopt some suitable and good features. As to ease the process of system development reviews of variety of tools are made to identity the suitable tools for system development.

Chapter 3: METHODOLOGY

3.1 Methodology

Methodology is a way of doing things includes technology, management, strategic, and last but not least, cost. It is necessary to investigate the needs of a development environment as to provide a better structure of managing resources, a better implementation stability of technology, a better-defined requirement and a well-structured of development phase.

Methodology is a collection of procedures, techniques, tools and documentation aids. It helps software developers to speed up and simplify the software development process. A methodology may consist of phases, which in turn may consist of sub-phases. The phases are guides, which helps the developers to choose the appropriate techniques. It also helps the developer to plan, manage, control and evaluate information systems projects. Different methodologies have different objectives. In order to develop a successful system, system developer has to be very clear on the purpose of the system to be developed.

Every system development process model (figure 3.1) includes system requirements (user, needs, resource) as input and a finished product as output.

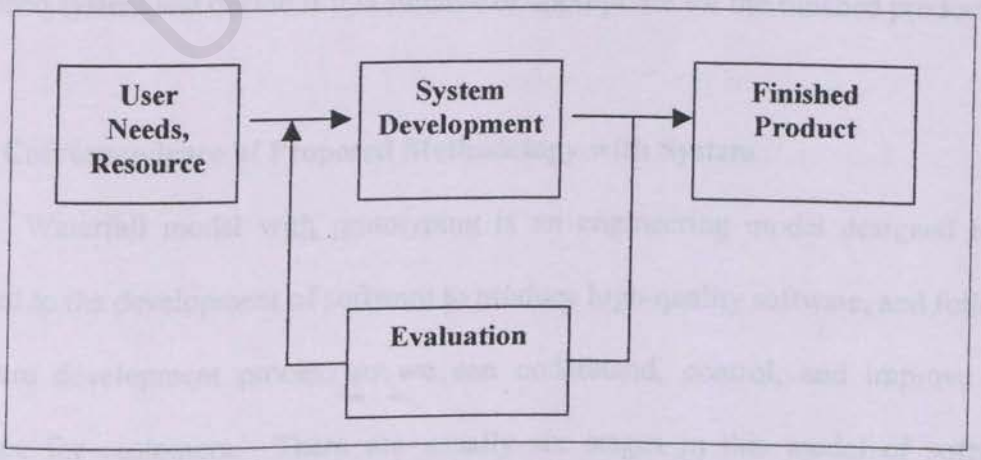


Figure 3.1: System Development Process Model

There are many kinds of development model in software engineering such as prototyping model, waterfall model or waterfall model with prototyping, spiral model and others. From investigations, it was found that different type of problems usually call for different problem-solving cycles and it is important to choose the most suitable cycle for a given problem. The choice is influenced by the nature of the problem.

In this chapter, I will emphasize on Waterfall model with prototyping, which to be used in my project development process.

3.1.1 Waterfall Model with Prototyping

The waterfall model was derived from engineering models to put some order in the development of large software products. It consists of different stages, which are processed in a linear fashion. Compared to other software development models it is more rigid and better manageable. The waterfall model is an important model, which is the basis of many other models.

Prototyping is such as a sub-process: a prototype is a partially developed product that enables customers and developers to examine some aspect of the proposed system and decide if it is suitable or appropriate for the finished product.

3.1.2 Correspondence of Proposed Methodology with System

Waterfall model with prototyping is an engineering model designed to be applied to the development of software to produce high-quality software, and follow a software development process so we can understand, control, and improve that produce for customers. There are usually six stages in this model of software development.

1. **Requirements Analysis and Definition**

In this stage, the requirements of the "to be developed software" are established. These are usually the services it will provide, its constraints and the goals of the software. Once there are established they have to be defined in such a way that they are usable in the next stage. This stage is often prelude by a feasibility study or a feasibility study is included in this stage.

2. **System Design**

This stage also involves outlining system functional by having feasibility studies or case studies on current system, determining and specifying hardware or software architecture and verifying system design.

3. **Program Design**

From the information of researches and studies, start determining and specifying program design and database design and verifying program design.

4. **Coding**

This stage involves programming, personal planning, tool acquisition, database development, component level documentation and programming management.

5. **Unit and Integration Testing**

After units are developed, start testing units separately and by different categories of users. Then integrating all tested units. After that, retest the integrated units.

6. **System Testing**

All the integrated units are combined into a system and how the whole is tested. When the combined programs are successfully tested the software product is finished.

7. **Acceptance Testing**

Testing on system completed. The system is delivered.

8. **Operation and Maintenance**

Most software products includes this stage of the development. It involves correcting errors that have gone undetected before, improvement and other forms of support. This stage is part of the life cycle of a software product, and not of the strict development, although improvements and fixes can still be considered as "development".

3.1.3 Prototyping

often, the user interface is built and tested as a prototype, so the users understand what the new system will like, and designer get a better sense of how the users like to interact with the system. Besides, prototyping is very useful for verification and validation.

- **Verification**

  It ensures that each function works correctly, checks the quality of the implementation and checks that a deliverable is complete (contains all requires information, follows standards).

- **Validation**

  It ensures that the system has implemented all of the requirements, so that each system function can be tracked back to a particular requirement in the specification. And makes sure that deliverables satisfy requirements specified in the previous stage or an earlier stage, and that the business case is met.

3.1.4 Strength of Waterfall Model with Prototyping

Waterfall model with prototyping is extension from waterfall model. It is one of most popular in system development environment. This is because it provided many benefit to developer. Below are strength of this model.

- **Stages by Stages**

Each process will implement in the stage by stage way. It also easy to understood.

- **Systematic and Sequential**

Each have been defined to avoid confusing to customer even developer.

- **Easy to Identify Project Milestone**

This is because each stage can separate with others stage.

- **Well Wide Used Method**

In system development environment, most system was developed by using this model.

- **Backtracking (feedback) and Iteration]**

The package has circulation factors so it allows changes in previous phase even through we are not in the current phase.

- **Low Project Risk**

Most developers are familiar and have more experience in this type of methodology, so the project risk will become low.

- **Advantages of Prototyping**

It can ensure the system meets the performance goals of constraints. Beside that, it also can ensure the system is practical and flexible. It also can ensure the system fulfill the users' requirement.

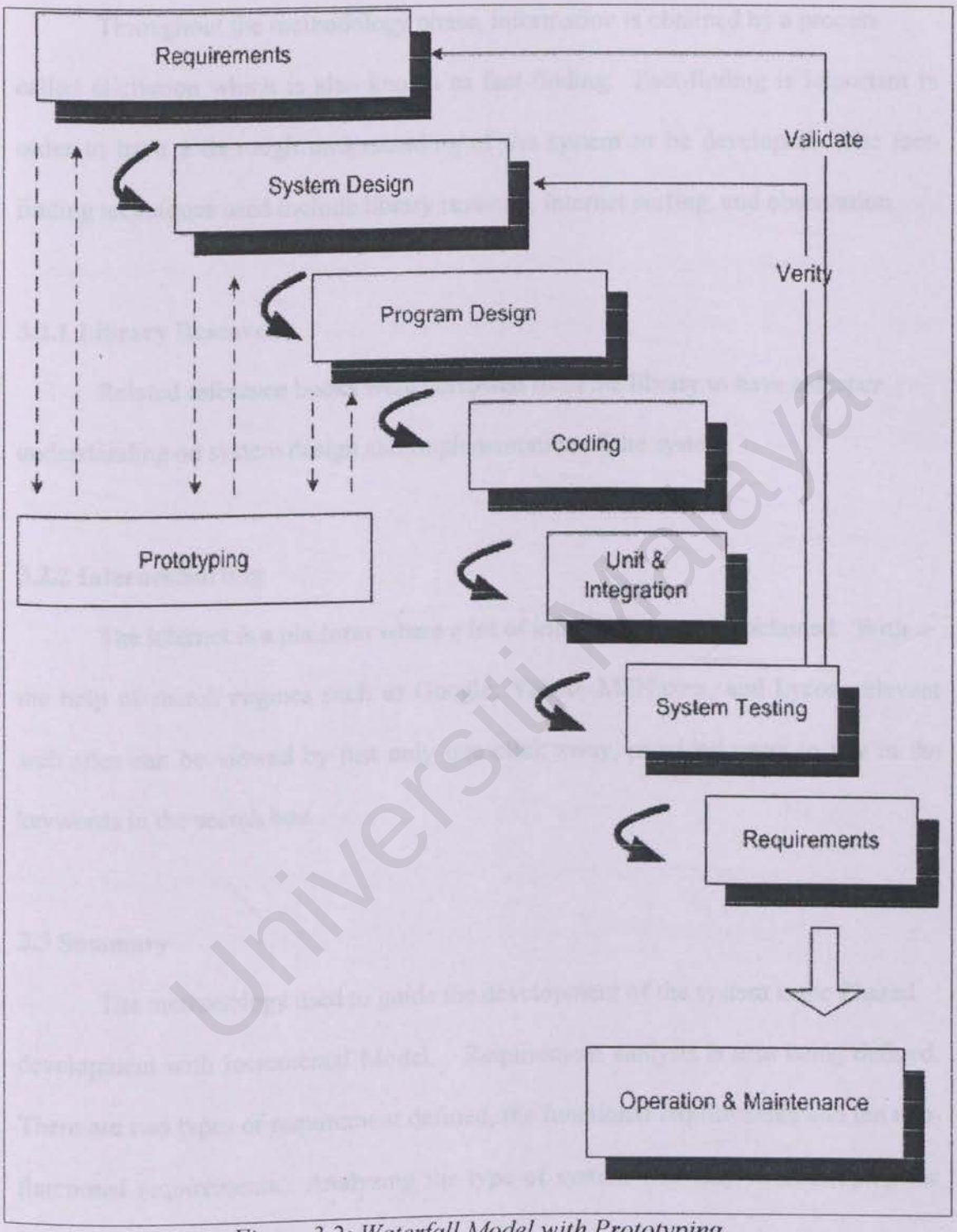


Figure 3.2: Waterfall Model with Prototyping

3.2 Fact Finding Techniques

Throughout the methodology phase, information is obtained by a process called elicitation which is also known as fact-finding. Fact-finding is important in order to have a thorough understanding of the system to be developed. The fact-finding techniques used include library research, Internet surfing, and observation.

3.2.1 Library Research

Related reference books were borrowed from the library to have a deeper understanding on system design and implementation of the system.

3.2.2 Internet Surfing

The internet is a platform where a lot of information can be obtained. With the help of search engines such as Google, Yahoo, MSN.com, and Lycos, relevant web sites can be viewed by just only one click away, provided users to key in the keywords in the search box.

3.3 Summary

The methodology used to guide the development of the system is the Phased development with Incremental Model. Requirement analysis is also being defined. There are two types of requirement defined, the functional requirements and the non-functional requirements. Analyzing the type of system will help in developing the system.

Chapter 4: SYSTEM ANALYSIS

4.1 Introduction

System analysis plays an essential role in order to ensure that the system met the requirements. The system definitely cannot be accomplished without the existence of these requirements. These requirements are further divided into two categories, functional requirement and non-functional requirement. A part from that, software and hardware requirements are identified to make the development process much easier.

4.2 Functional Requirements

Functional requirement are those requirements that the customer needs in order to do business. Furthermore, it captures the tasks that the business must perform and as such it does not include implementation details such as what hardware or software the system must use.

The function requirements for *IT Stuff Procurement System* consist of 4 modules: super-administrator, administrator, client, and vendor. Before accessing the system, there is one requirement to be met:

- **Log in**

  All users of the system need to provide their login ID and password for verification before accessing any provided services. They will then direct to a page where all the option of functionalities are provide.

4.2.1 Super-Administrator Module

It provides an easier way to super-admin to keep track of status of purchase orders and monitor administrators performance. It also enables super-admin to assign task (which mean each purchase order application will be handled by, let say one administrator) among administrators.

4.2.2 Administrator Module

It helps administrator to manage information of vendors and client. Information of each vendor assists administrator in making decision of which vendor is the best to deal with for a particular purchasing order. Each purchase order application will be process by administrator and the system help administrator to also update the status of each application from time to time.

4.2.3 Client Module

Clients manage to make purchase order application by provide some related information. They can view the latest status of each purchase order application.

4.2.4 Vendor Module

It offers information of current purchase order applications to vendor. Vendor can make changes to some of their company details. Beside, information of person to contact is provided.

4.3 Non-functional Requirements

Non-functional requirement are those requirements that are not directly needed by the business but are important. They represent constraints place on the

system such as performance, usability and also hardware and/or software the system must use.

4.3.1 User Friendliness

For a system to gain popularity, it must be easily understood by the users. The users need not to know what happen behind the system but through the system's user interface, uses are supposed to get whatever they want easily. Below are some 'schemes' provided to measure whether a system is user friendly:

- Consistent, in term of screen design and error messages displayed.
- Appropriate error handling with associated error messages.
- High degree of understandability and avoid too much of memorisation of events and commands for the users.

4.3.2 User Interface

The system is required to have an interface that is simple and allow the user to access the internal components of the system easily. A common interface must be created in order to reduce the learning curve of the users. Users of this system are usually busy and more attention is paid to their daily chores. Therefore, the computing features and learning commands should not be complicated and time consuming. The design of the user screens must be a systematic standard of Windows Graphical User Interface (GUI), module browser to allow users to shift between the main modules, and attrative icon buttons/toolbars with direct, short menu option.

4.3.3 Efficiency

A system is said to fulfill the efficiency requirement when its process or procedure can be called, accessed and functioning well to produce outcomes or output at a pace or speed acceptable by the users. Futhermore, all that has to happen in an unlimited of times after the system implementation whenever the users need it. The outcomes of the same process or procedure with the same input must be similar every time being called.

4.3.4 Reliability

A system is reliable if it does not produce costly or dangerous failures in the typical user environment and data integrity is preserved. All potential and possible failures and errors must be taken into account during the design and development stage.

4.3.5 Short Loading Time and Respond Time

As normal, users like the system to repond fast. Thus a system must be able to provide short loading time and respond time. Slow loading and respond time might cause the users to wait and discourage them from using the system again. However, the system's performance sometimes depends on the hardware used.

4.3.6 Modularity

Modularity means the system is broken into small modules so that distinct functions of objects could be isolated from one to another. This will make the system testing and maintenance process easier because the processes can be done portion by portion and not involving the whole system.

4.3.7 Maintainability

This may be defined qualitatively as the ease with which software can be understood, corrected, adapted, and enhanced.

4.3.8 Security

The system should be equipped with sufficient security. Each access by the user should be authenticated and validated by the system. System will ensure that sensitive information will be handled in a safe and professional manner. In addition, varios levels of functionality will be provided according to the user's status.

4.3.9 Functionality

The functinalities stressed here are the searching and retrieving capability, which is very essential in any web applications that deal with data retrieval from existing databese.

4.3.10 Reusability

The system can be enhanced or to use by other system as well.

4.4 Chosen Technology, Software, and Development Tools

4.4.1 Development Platform – Linux

Why choose Linux operating system?

- freely-distributable open source operating system
- multitasking, multiuser operating system
- provide access to virtual console

- use a standard layout for files so that system resources and programs can be easily located called the directory tree
- provides a mechanism known as file permissions, which protect user files from tampering by other users. In other words, it has highest level of security

4.4.2 Database Management System – PostgreSQL

Why choose PostgreSQL?

- Open source database, so it is free
- has most features present in large commercial DBMSs, like transactions, sub selects, triggers, views, foreign key referential integrity, and sophisticated locking. We have some features they do not have, like user-defined types, inheritance, rules, and multi-version concurrency control to reduce lock contention.
- has performance similar to other commercial and open source databases.
- Reliable
- mailing lists provide contact with a large group of developers and users to help resolve any problems encountered. While we cannot guarantee a fix, commercial DBMSs do not always supply a fix either. Direct access to developers, the user community, manuals, and the source code often make PostgreSQL support superior to other DBMSs. There is commercial per-incident support available for those who need it.
- there are several graphical interfaces to PostgreSQL available

4.4.3 Development Web Server – Apache

- feature-full open source web server

- fast, stable, powerful, flexible, HTTP/1.1 compliant web server implements the latest protocols
- unrestrictive license runs on Windows NT/9x, Netware 5.x and above, OS/2, and most versions of Unix, as well as several other operating systems
- DBM databases for authentication
- Customized responses to errors and problems
- Unlimited flexible URL rewriting and aliasing
- Content negotiation

4.4.4 Development Language – PHP (Hypertext Preprocessor)

- Open source programming language
- compatibility with many types of databases
- can be embedded inn HTML as has not limited to output HTML
- PHP can talk across networks using IMAP, SNMP, NNTP, POP3, or HTTP.
- saving the end user a considerable amount of load time
- available for all major platform
- easy to learn

4.4.5 Development Tools

Macromedia Dreamweaver UltraDev

The Macromedia Dreamweaver UltraDev will be used as the software for scripting of the system. It provides many features and is stable in performance. This software supports language that can be used for writing scripting. In addition, it is user-friendly and powerful. As a result, Macromedia Dreamweaver UltraDev will be the more suitable development tool and is chosen due to the familiarity.

4.4.6 Software Requirement

Software	Description
Linux Operating System	As a development platform
HTML, PHP, JavaScript, JavaApplet	Coding development
Apache Server	As web server
PostgreSQL	To store and manipulate data
Internet Explorer 6.0	As web browser
Macromedia Dreamweaver UltraDev	Tools to develop the system

Table 4.1: Software Requirements

4.4.7 Hardware Requirements

Hardware	Description
Normal PC 1. Intel or compatible Pentium 400 Mhz and above 2. 128 MB RAM and above 3. Windows 98 4. 64K color with 800x600 resolution 5. 5GB of hard disk apace	Minimum requirement for system development
Network card	To connect to the network.

Table 4.2: Hardware Requirements

Chapter 5: SYSTEM DESIGN

5.1 Intoduction

System design is the creative process of transforming the problem into a solution or its a process where all the conceptual ideas from requirements specification are converted into technical specification.

In the system design phase, the system requirements gathered during the analysis phase and research conducted earlier are transmitted into a representation of system. Initially, the representations that is close to source code.

In the system design phase, input, output, file and database were producted which include the designed of input forms, screen in order to gather input data, data dictionary, file specification and report design. The objectives of system design are listed below:

1.) Specify Logical Design Elements

Detailed design specifications that describe the features of an information system: input, output, files and database and procedures.

2.) Meet User Requirements

- Performing appropriate procedures correctly
- Presenting proper form of information
- Providing accurate results
- Using appropriate method of interaction
- Providing overall reliability

3.) Ease of use

- Favorable human engineering
- Ergonomic design that is physically comfortable to user effectiveness and efficiency

4.) Provide Software Specification

- Specific components and functions with adequate detail to construct application software

Following are the design principles from the aspect of software engineering in order to achieve a good design:

a) Modularity

A system is said to be modular if it is decomposed into simpler, well-defined modules and inter-modular interfaces. The reason why modularity is desirable is because a system that is modular is easy to understand, delegate, code, test, document, and maintain.

b) Cohesion

A cohesion module performs a single processing function and requires little interaction with other procedures being performed in other parts of the program. In other words, a component is cohesion if all elements of the component are directed towards and essential for performing the same task.

c) Span of Control

Modules should interact with and manage the functions of a limited number of lower-level modules. Sub-modules within a module should be accessed from the main module and these sub-modules should be interred-linked.

d) Shared Use

The same function should not be duplicated in separate modules but eatablished in a single module that can be invoked by any other module when needed.

5.2 Overview of System Architecture

The system architecture is besed on the traditional on the traditional client-server architecture. Basically, the software system is broken down into 3 tier; the Presentation-tier, Business-tier, amd data-tier.

The presentation-tier deals with the presentation of the system to the end user. This part is web-enabled and is developed using Hypertext Preprocess (PHP). Development in this section involves work in web design, web authoring and web programming. The business-tier does the processing behind the scenes. This part applies the business rules and formatting to the data that is to be sent to the presentation layer. The database-tier accesses the database to reak or write data. This tier deals with all the connectivity with the database.

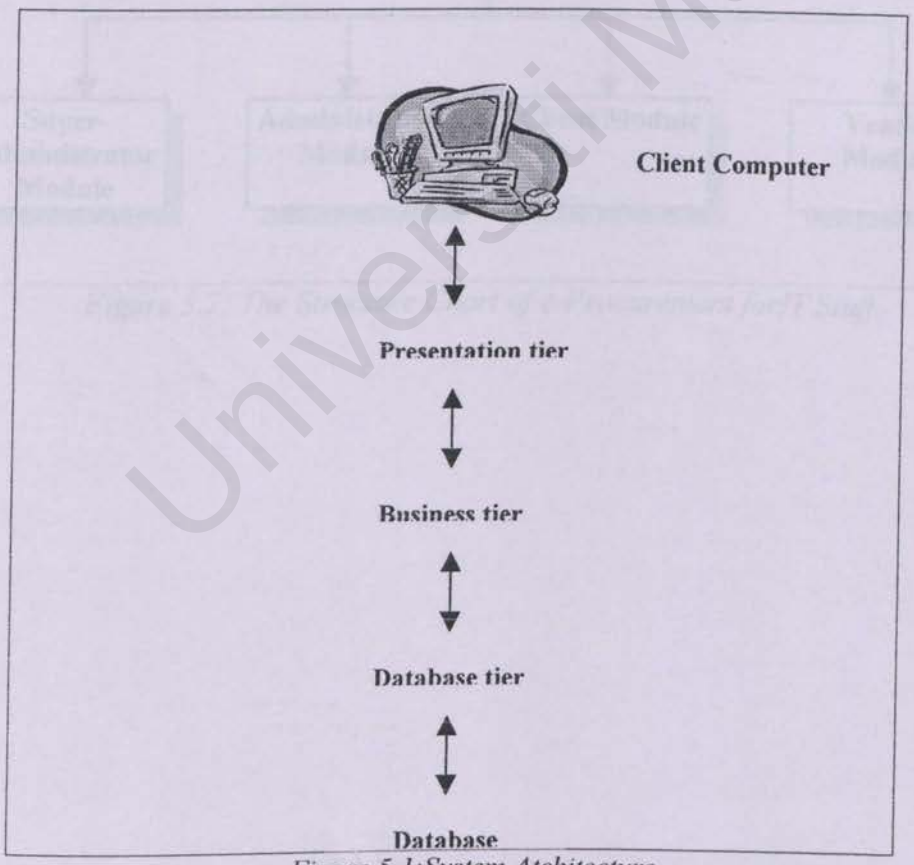


Figure 5.1:System Atchitecture

5.3 System Functionality Design

System functionality design is based on the system requirements stated in Chapter 3. It translates the system requirements into system functionality. This design focuses on the system structure design and data flow diagram.

5.3.1 System Structure Chart

The system structure is used to depict high-level abstraction of a specified system. The use of structure chart is to describe the interaction between independent modules. The full system structure chart can be represented as below.

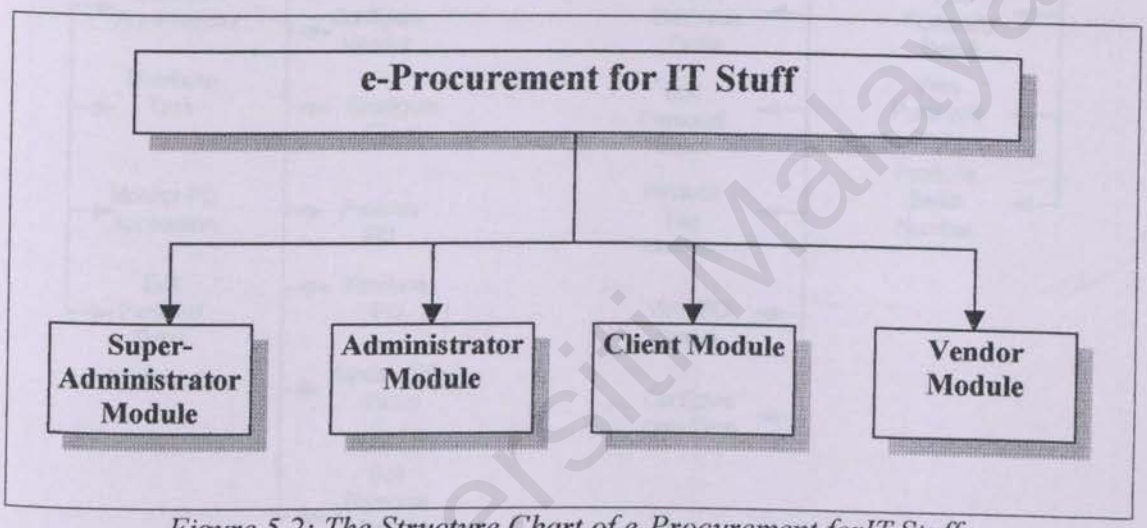


Figure 5.2: The Structure Chart of e-Procurement forIT Stuff

5.3.1.1 Structure Chart of e-Procurement for IT Stuff

5.3.1.1 Structure Chart of e-Procurement for IT Stuff

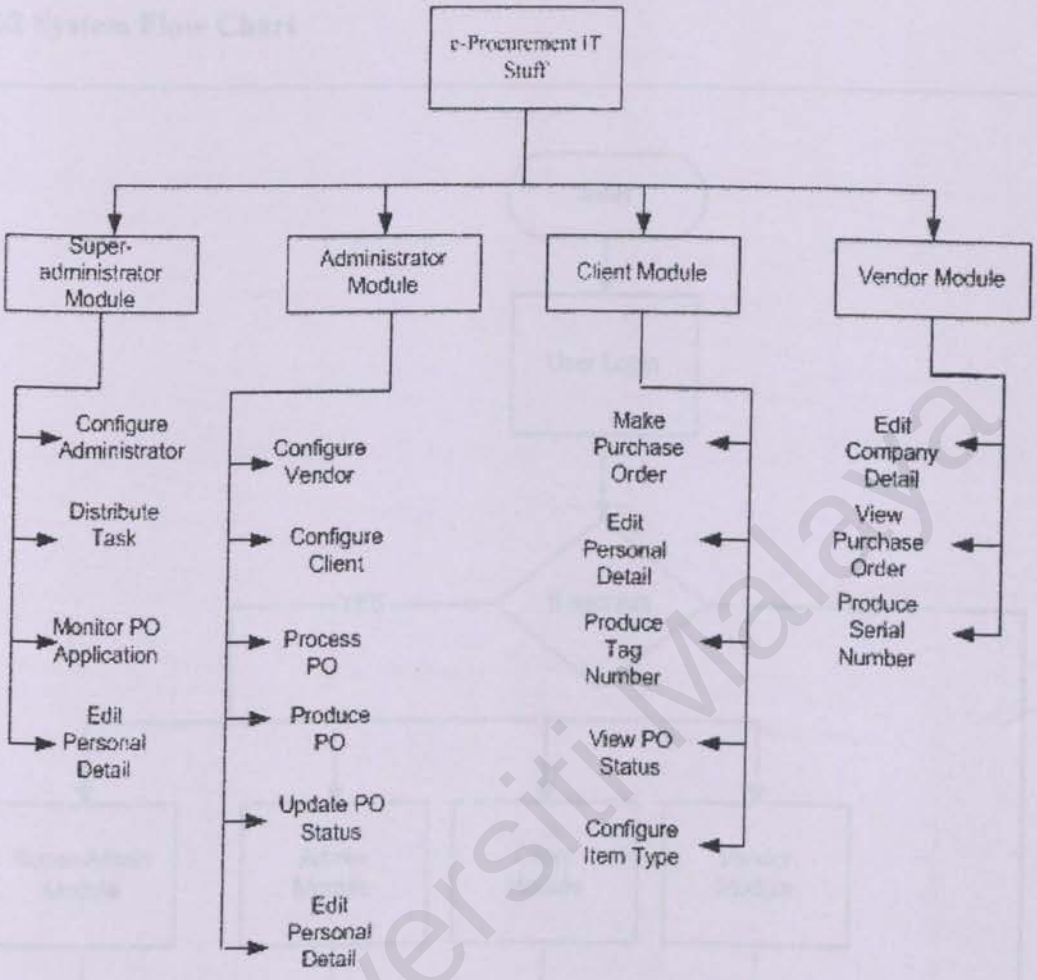


Figure 5.3: e-Procurement for IT Stuff Structure Chart

5.3.2 System Flow Chart

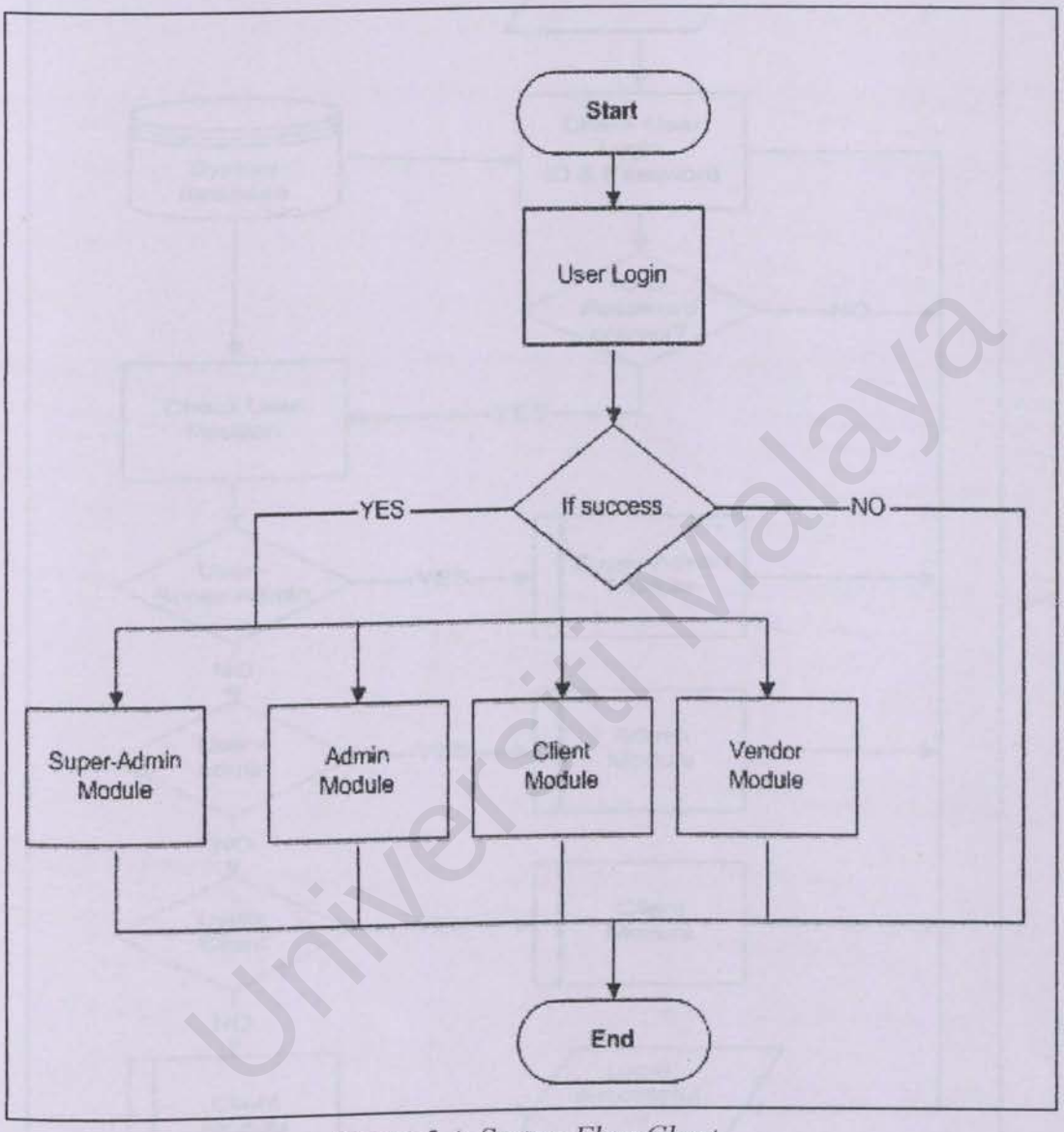


Figure 5.4: System Flow Chart

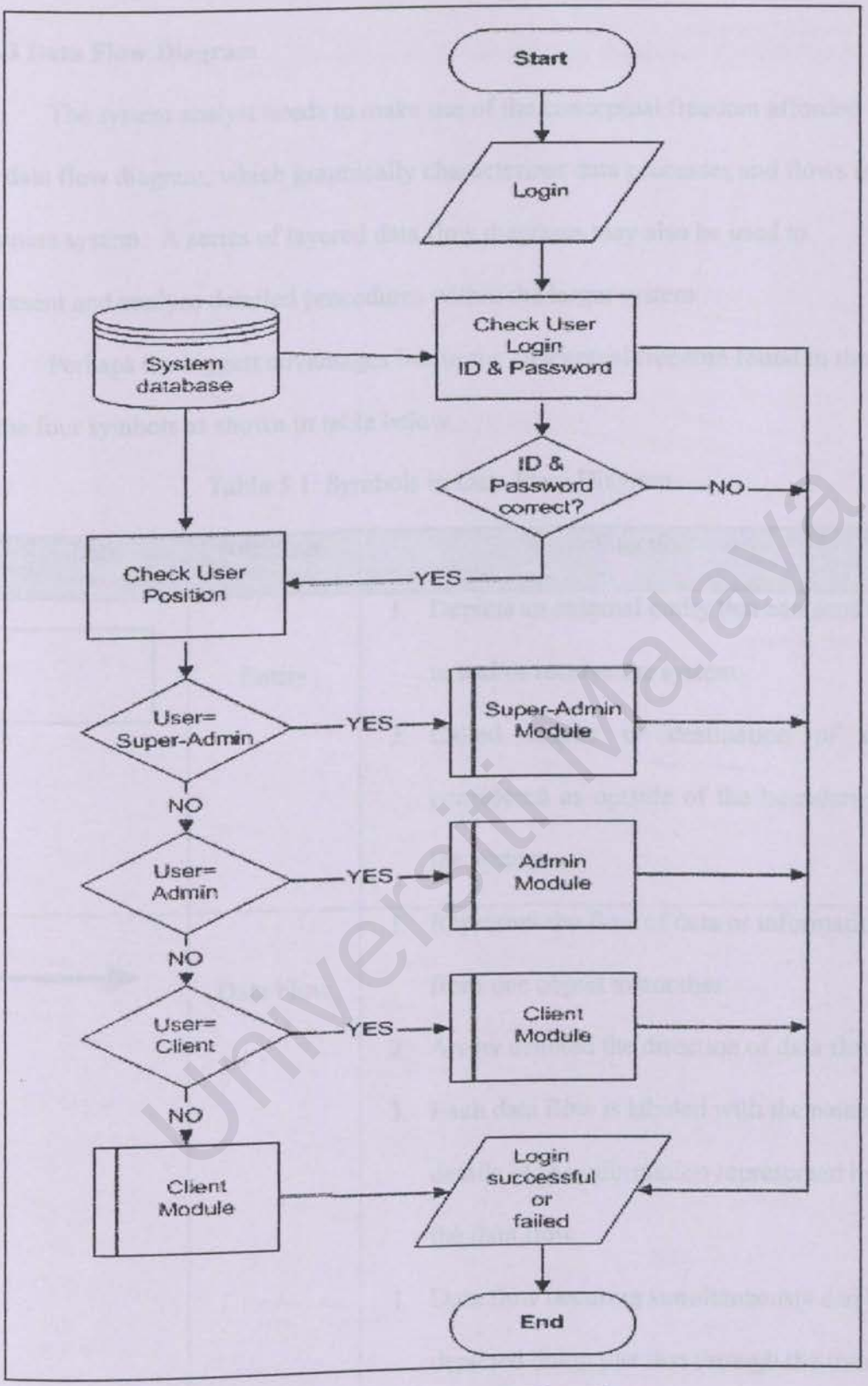


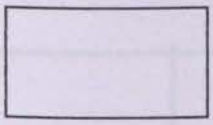
Figure 5.5: System Login Flow Chart

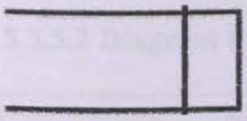
5.3.3 Data Flow Diagram

The system analyst needs to make use of the conceptual freedom afforded by the data flow diagram, which graphically characterizes data processes and flows in a business system. A series of layered data flow diagrams may also be used to represent and analyze detailed procedures within the larger system.

Perhaps the biggest advantages lies in the conceptual freedom found in the use of the four symbols as shown in table below.

Table 5.1: Symbols in Data Flow Diagram

Symbols	Attributes	Function
	Entity	1. Depicts an external entity that can send data to and/or receive the system. 2. Called source or destination of data, considered as outside of the boundaries of the system.
	Data Flow	1. Represent the flow of data or information from one object to another. 2. Arrow denoted the direction of data flow. 3. Each data flow is labeled with the name or details of the information represented by the data flow. 4. Data flow occuring simultaneously can be depicted doing just that through the user of parallel arrows.

	Process	1. Transform the input data to output data. 2. Represented by rectangle shape. 3. Comprise 2 / 3 section: a) Top section contain the identifier information. b) Center section contains a description of the process. c) Lower section contains the physical or computer program information.
	Data store	1. Represents data store and holds data for a given time within the system. 2. May represent manual store, such as filling cabinets or computerized files or databases. 3. Comprises of two section: a) Identifier reference number b) Description of the data stored

5.3.3.1 Context Diagram of e-Procurement for IT Stuff System

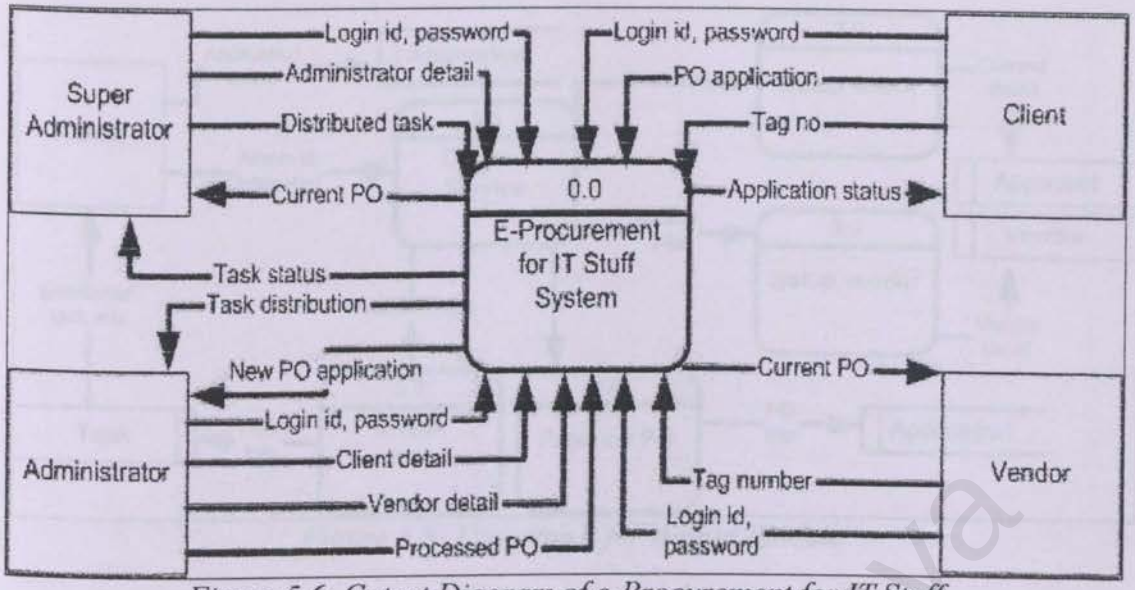


Figure 5.6: Cotext Diagram of e-Procurement for IT Stuff

5.3.3.2 Diagram 0 for Super-Admin Module

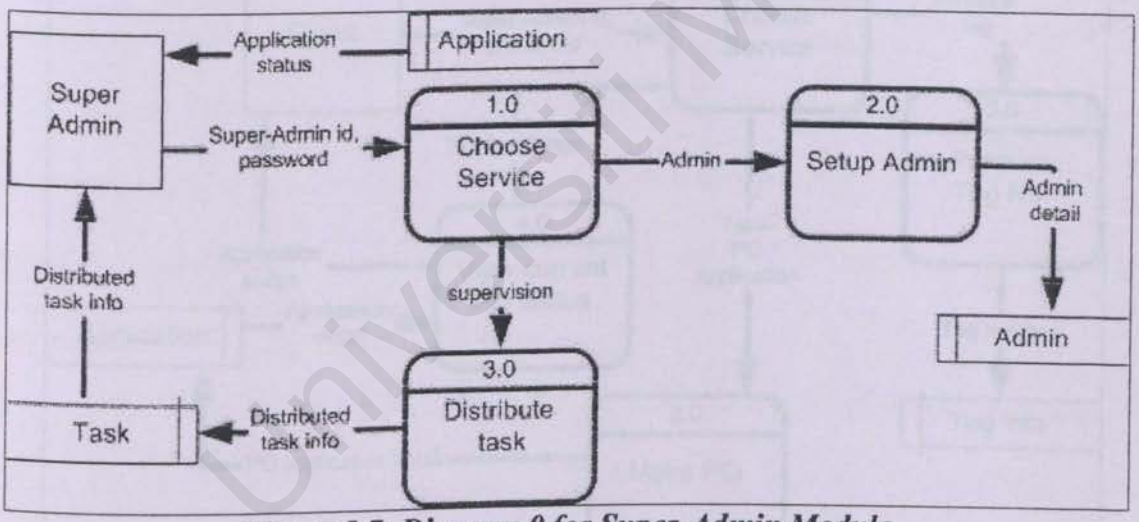


Figure 5.7: Diagram 0 for Super-Admin Module

5.3.3.3 Diagram 0 for Admin Module

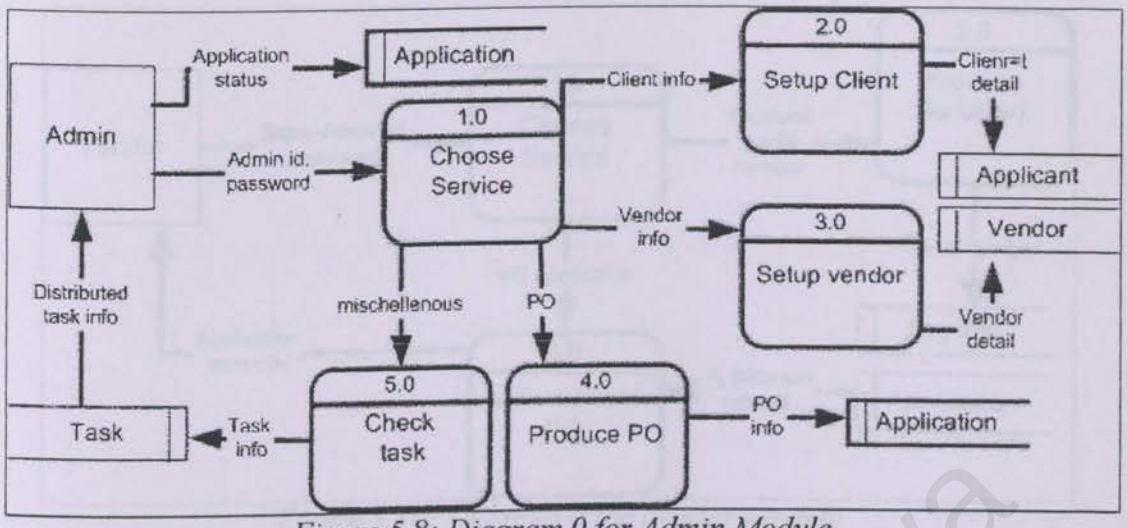


Figure 5.8: Diagram 0 for Admin Module

5.3.3.4 Diagram 0 for Client Module

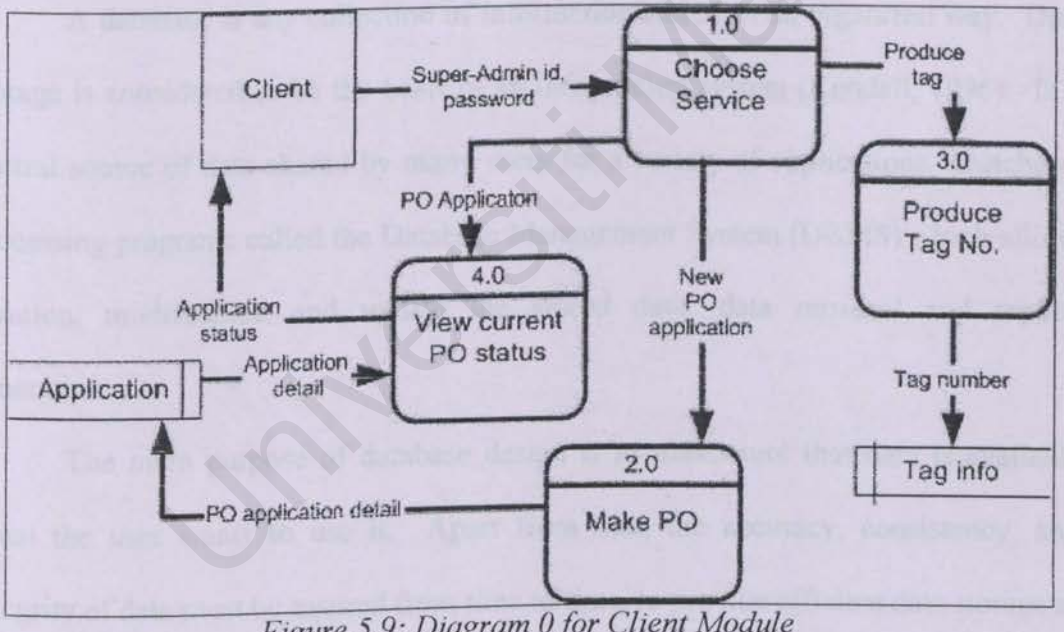


Figure 5.9: Diagram 0 for Client Module

5.3.3.5 Diagram 0 for Vendor Module

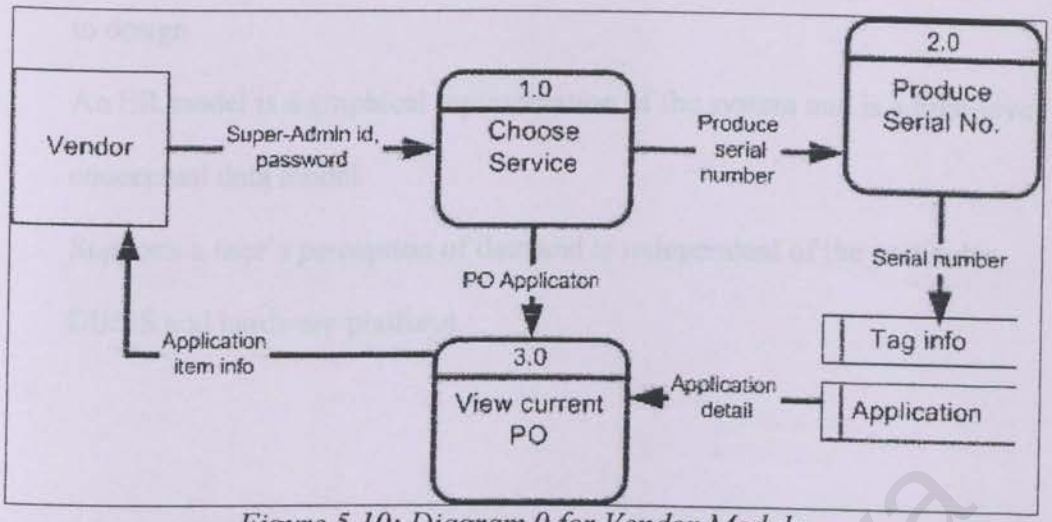


Figure 5.10: Diagram 0 for Vendor Module

5.4 Database Design

A database is any collection of information stored in an organized way. Data storage is considered to be the heart of an information system (Kendall, 1996). It is central source of data shared by many users for a variety of applications. Databese-processing programs called the Database Management System (DBMS) which allows creation, modification and update the stored data, data retrieval and reports generation.

The main purpose of database design is to make sure that data is available when the user wants to use it. Apart from that, the accuracy, consistency, and integrity of data must be assured from time to time, to provide efficient data storage as well as efficient updating and retrieval, to reduced data duplication, and easier representation of the users' perspectives.

In 1976, Peter Chan had introduced the use of the entity-relationship model (E-R Model). An E-R diagram contains many entities, many different types of relations, and numerous attributes. The benefits of Entity Relationship modeling are as follow:

1. Databases need to be designed and entity relationship (ER) modeling is an aid to design.
2. An ER model is a graphical representation of the system and is a high-level conceptual data model.
3. Supports a user's perception of data and is independent of the particular DBMS and hardware platform.

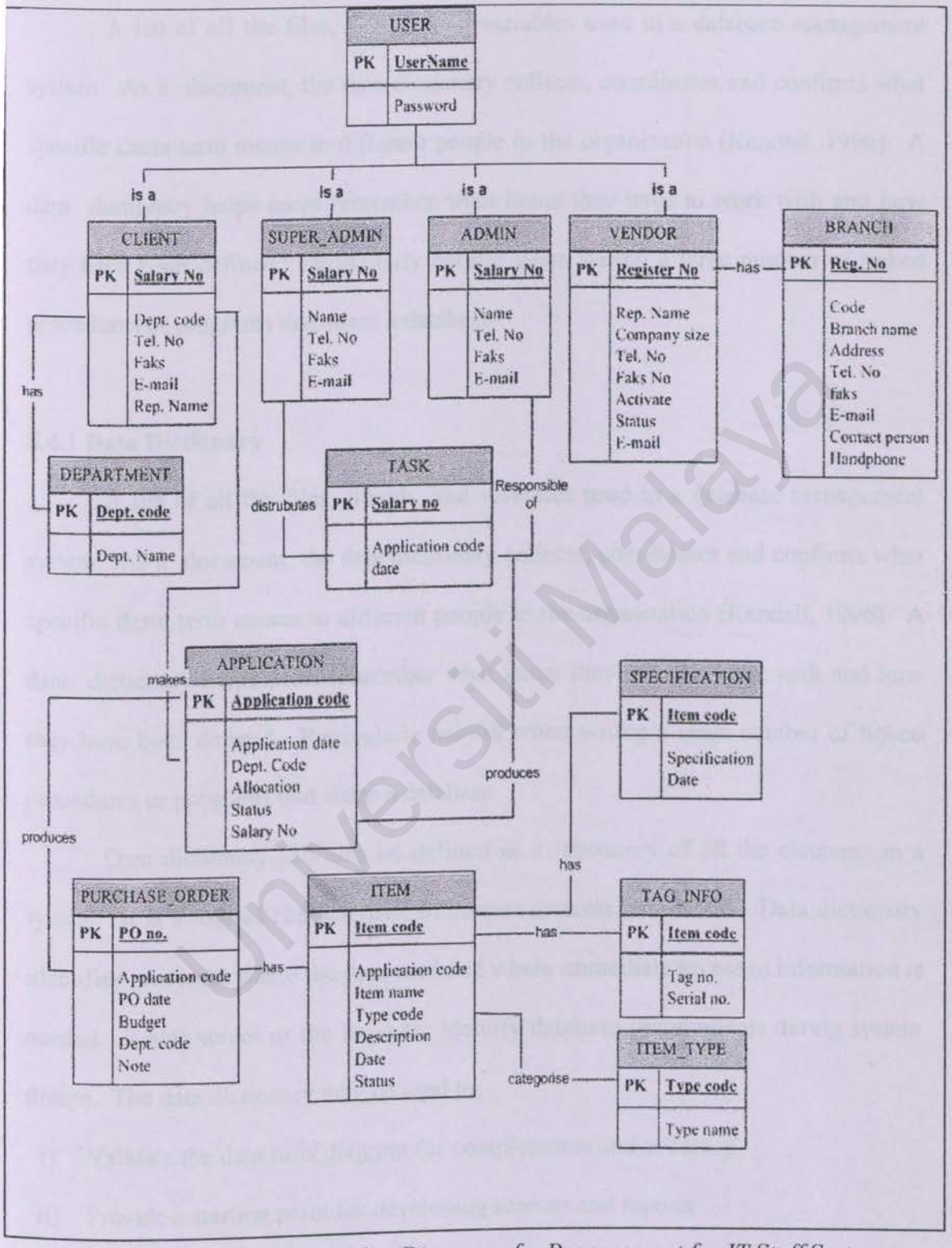


Figure 5.11: Entity Relationship Diagram of e-Procurement for IT Stuff System

5.4.1 Data Dictionary

A list of all the files, fienlds, and variables used in a database management system. As a document, the data dictionary collects, coordinates and confirms what specific dasta term means to different people in the organization (Kendall, 1996). A data dictionary helps users remember what items they have to work with and how they have been defined. Particularly helpful when writng a large number of linked procedures or programs that share a database.

5.4.1 Data Dictionary

A list of all the files, fienlds, and variables used in a database management system. As a document, the data dictionary collects, coordinates and confirms what specific dasta term means to different people in the organization (Kendall, 1996). A data dictionary helps users remember what items they have to work with and how they have been defined. Particularly helpful when writng a large number of linked procedures or programs that share a database.

Data dictionary can also be defined as a repository of all the elements in a system. It is a logical characteristic of current systems data stores. Data dictionary identifies processes where the data used and where immediate access to information is needed. It alse serves as the basic for identify database requirements during system design. The data dictionary may be used to:

i) Validate the data flow diagram for completeness and accuracy.

ii) Provide a starting point for developing screens and reports.

iii) Determine the logic point for data flow diagram process.

A data dictionary entry should contain specific categories of information including:

i) Name and aliases of the data item.

ii) Description of the data item.

iii) Permissible range of the data item.

iv) Its allowable length in information.

v) Any other pertinent editing information.

The data dictionary of *IT Stuff Procurement System* is shown as below:

Database Name: Procurement

1.) Table Name: applicant (Client)

Description: To store the details of each PO applicant

Table 5.2: Table Structure of Applicant (Client)

Field	Data Type	Length	Index	Description
appt_username	varchar	20		Applicant's login name
appt_id	varchar	20		Applicant's ID
appt_salaryno	varchar	8	Primary key	Applicant's salary number
appt_deptcode	varchar	5		Applicant's department code
appt_name	varchar	50		Applicant's name
appt_tel	char	12		Applicant's telephone
appt_faks	char	12		Applicant's faks number
appt_email	varchar	30		Applicant's e-mail address

2.) Table Name: super_administrator

Description: To store details of super-administrator

Table 5.3: Table Structure of Super-Administrator

Field	Data Type	Length	Index	Description
sadm_username	varchar	20		Super-admin's login name
sadm_id	varchar	20	Primary key	Super-admin's ID
sadm_name	varchar	50		Super-admin's name
sadm_salaryno	varchar	8		Super-admin's salary number
sadm_tel	varchar	12		Super-admin's telephone number

sadm_faks	varchar	12		Super-admin's faks number
sadm_email	varchar	20		Super-admin's e-mail address

3.) **Table Name: administrator**

Description: To store details of each administrator

Table 5.4: Table Structure of Administrator

Field	Data Type	Length	Index	Description
adm_username	varchar	20		Admin's login name
adm_id	varchar	20	Primary key	Admin's ID
adm_name	varchar	50		Admin's name
adm_ salaryno	varchar	8		Admin's salary number
adm_tel	varchar	12		Admin's telephone number
adm_faks	varchar	12		Admin's faks number
adm_email	varchar	20		Admin's e-mail address

4.) **Table Name: department**

Description: To store department's data

Table 5.5: Table Structure of department

Field	Data Type	Length	Index	Description
dept_code	varchar	5	Primary key	Department's code
dept_name	varchar	50		Department's name

5.) **Table Name: vendor**

Description: To store details of vendor

Table 5.6: Table Structure of vendor

Field	Data Type	Length	Index	Description
ven_username	varchar	20		Vendor's login name
ven_id	varchar	6		Vendor's password
ven_name	varchar	50		Vendor's name
ven_regno	varchar	20	Primary key	Vendor's register number
ven_status	varchar	2		Vendor's status
ven_activate	varchar	1		Vendor's activation

ven_size	varchar	10		Vendor's company size

6.) **Table Name: application (Purchase Order)**

Description: To record the application details

Table 5.7: Table Structure of Purchase Order Application

Field	Data Type	Length	Index	Description
app_code	varchar	8	Primary key	Purchase order application's code
app_date	Date			Application's date
app_salaryno	varchar	8		Refer to staff in-charge of the application
app_deptcode	varchar	5		Department code related to the application
app_allocate	Float			Amount of allocation for an application
app_status	varchar	1		Application status

7.) **Table Name: application_item**

Description: To record detail of item to be purchase

Table 5.8: Table Structure of Item to be Purchased

Field	Data Type	Length	Index	Description
apem_appcode	varchar	8	Primary key	Application code the item to be purchased
apem_itemcode	varchar	3		Code of the item
apem_typecode	varchar	4		Code for item type
apem_quant	char	7		Quantity to be purchased
apem_description	text			Item description
apem_statuscode	varchar	3		Item status code

8.) **Table Name: status_item**

Description: To record the latest status of items

Table 5.9: Table Structure of Status of Item

Field	Data Type	Length	Index	Description
stem_appcode	varchar	8		Application code related to item
stem_itemcode	varchar	3	Primary key	Code of item
stem_statuscode	varchar	3		Item status code
stem_salaryno	char	8		Staff in-charge
stem_date	timestamp			The date status item updated

9.) **Table Name: item_type**

Descriptioin: To store the item type and related data

Table 5.10: Table Structure of the Type of Item

Field	Data Type	Length	Index	Description
itype_itypecode	varchar	4	Primary key	Code of item type
itype_name	varchar	30		Name of item type

10.) **Table Name: Expertise**

Description: To store details of expertise

Table 5.11: Table Stucture of Expertise

Field	Data Type	Length	Index	Description
ex_code	varchar	4	Primary key	Code of expertise
ex_name	varchar	30		Name of expertise

11.) **Table Name: itype_expertise**

Description: To record expertise of each item type

Table 5.12: Table Structure of Expertise of Item Type

Field	Data Type	Length	Index	Description
itex_excode	varchar	4	Primary key	Code of expertise
itex_itypecode	varchar	4		Code of item type

12.) Table Name: branch

Description: To record details of vendor

Table 5.13: Table Structure of Branch (vendor)

Field	Data Type	Length	Index	Description
bra_venid	varchar	6	Primary key	Vendor's ID
bra_code	varchar	3		Code of branch
bra_name	varchar	30		Name of branch
bra_add	text			Branch's address
bra_tel	char	12		Branch's telephone
bra_faks	char	12		Branch's faks number
bra_email	char	30		Person-in-charge's e-email address
bra_cperson	varchar	50		Name of person incharge
bra_activate	varchar	1		Branch's activation
bra_hphone	char	12		Person-in-charge's handphone number

13.) Table name: vendor_expertise

Description: To store expertise of vendor

Table 5.14: Table Structure of Vendor's Expertise

Field	Data Type	Length	Index	Description
exven_venid	varchar	6	Primary key	Vendor's ID
exven_excode	varchar	4		Expertise's code

14.) Table Name: status

Description: To store status data

Table 5.15: Table Structure of Status

Field	Data Type	Length	Index	Description
sta_code	varchar	3	Primary key	Code of status
sta_staname	varchar	50		Name of status

15.) Table Name: ensure_spec

Description: To record specification of item

Table 5.16:Table Structure of Ensured Specification

Field	Data Type	Length	Index	Description
espec_appcode	varchar	8	Primaty key	Code of purchase order application
espec_itemcode	varchar	4		Code of item
espec_excode	varchar	4		Code of item expertise
espec_date	timestamp			Data specification made
espec_specification	text			Specification
espec_quant	char	7		Quantity of application
espec_itypecode	varchar	4		Code of item type
espec_status	varchar	3		Status of item

16.) Table Name: Item

Description: To store item name

Table 5.17: Table Structure of Item

Field	Data Type	Length	Index	Description
item_code	varchar	4	Primary key	Code of item
item_name	varchar	30		Name of item

17.) Table Name: produce_po

Description: To record details of purchase order

Table 5.18: Table Structure of Producing PO

Field	Data Type	Length	Index	Description
propo_appcode	varchar	8	Primary key	Code of application
propo_group	varchar	3		Group of application
propo_date	timestamp			Produce date of PO
propo_pono	varchar	10		Number of PO

propo_totalprice	float			Total price of PO
propo_vencode	varchar	6		Code of related vendor
propo_bracode	varchar	3		Code of branch
propo_deptcode	varchar	5		Code of department
propo_staffname	varchar	50		Name of staff involved
propo_note	varchar	100		Note for PO

18.) Table Name: produce_tag

Description: To record tag of item

Table 5.19: Table Structure of Produce Tag

Field	Data Type	Length	Index	Description
protag_appcode	varchar	8		Code of related application
protag_itemcode	varchar	4	Primary key	Code of item
protag_date	Timestamp			Date procuded tag
protag_status	varchar	3		Status of tag

19.) Table Name: task_distribution

Description: To record task distribution for each PO application

Table 5.20: Table Structure of Task Distribution

Field	Data Type	Length	Index	Description
td_appcode	varchar	8	Primary key	Code of application
td_adminid	varchar	20		Administrator's ID

20.) Table Name: tag_information

Description: To record tag information

Table 5.21: Table Structure of Tag Information

Field	Data Type	Length	Index	Description
ti_appcode	varchar	8		Code of application

ti_itemcode	varchar	4	Primary key	Code of item
ti_bilcode	varchar	3		
ti_serialno	varchar	30		Serial number of item
ti_tagno	varchar	30		Tag number of item

5.5 User Interface Design

User interface design describe how software communicates with the human user who uses it (Mundher, 1994). The web services system user interface design focuses on the effective general interaction between its user and the system. It also takes into account development of complete, unambiguous and easy-to-understand information displays.

Interface need to be usable and consist of certain characteristics in common:

i) They reflect the workflows that are familiar or comfortable

ii) They support the user's learning style

iii) They are compatible in the users' working environment

iv) They encompass a design concept (a metaphor or idiom) that is familiar to be users

v) They have a consistency of presentation (layout, icons, interactions) that makes them appear reliable and easy to learn

vi) The usaage of languages and illustrations are familiar to the users or are easy to learn

In short, usable interfaces fit in, simply and elegantly, with user's life and work needs.

5.5.1 Web Browser Screen Design

All user interface designs serve the same general purposes; to present informatin and to aid users in using the system. Graphical User Interfaces (GUIs), are by far the most popular. In a GUI, computer resourses (such as programs, data files, and network connections) are represented by small pictures, called icons. GUI will be in used because it provides a user-friendly environment to ease the administrative task. Other features to be consider during user interface development are as follow:

i) All screen displays are attractive and less congested with images

ii) Information on a single screen should displayed in a meaningful and logical order

iii) Screen presentation should be consistent

5.5.2 Example of User Interface Design

5.5.2.1 Example of User Interface for Main Page

Figure 5.12: User Interface of System Main Page

5.5.2.2 User Interface of Create New Admin

Figure 5.13: User Interface of Create New Admin Page

5.5.2.3 User Interface of Applicant Make New PO Application

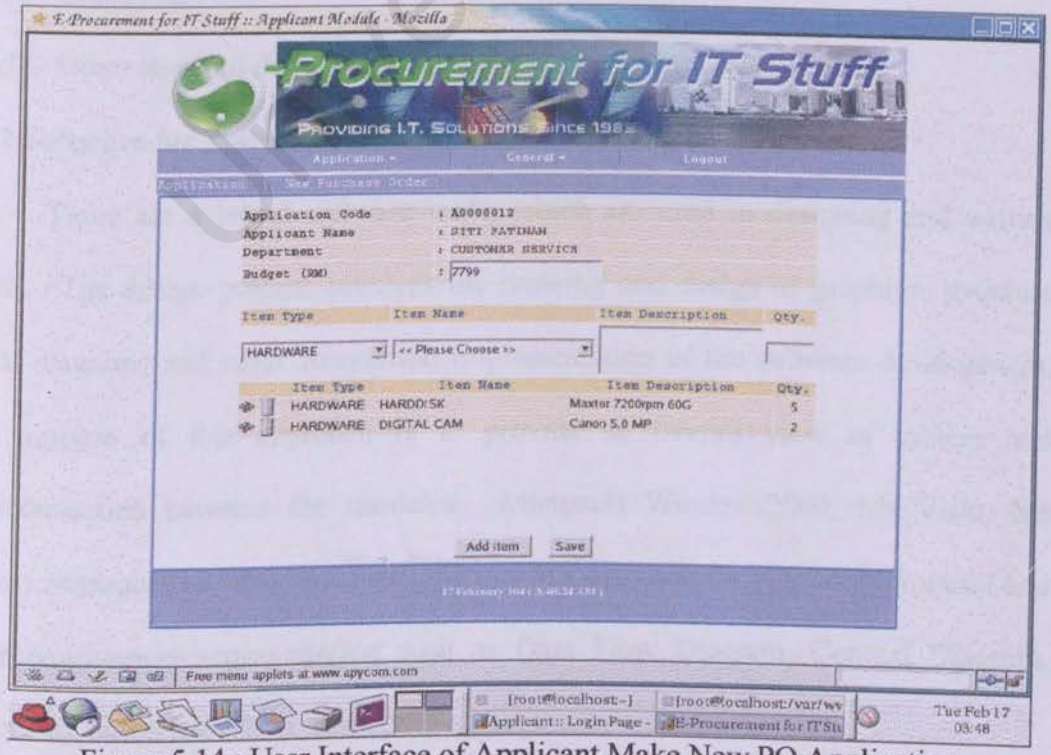


Figure 5.14 : User Interface of Applicant Make New PO Application

CHAPTER 6: SYSTEM IMPLEMENTATION

6.1 Introduction

System implementation is a process to convert the system requirement into program codes. The initial stage of system implementation involves setting up the development environment. This involves also setting up development tools to facilitate the system implementation. Using the suitable hardware and software will not only help to speed up the system development but also determine the success of the project. Setting up the development environment includes setting up development platform, web server, and database server to facilitate the system implementation.

6.1.1 Hardware Configuration

The following hardware specifications have been used to develop *IT Stuff Procurement System*:-

a). Pentium IV Processor

b). 256 DDRAM

c). 20GB Hard Disk

d). Other standard desktop PC component

6.1.2 Software for System Design and Report Writing

There are a lot of software tools, which are used in designing and writing report. The design process involves the drawing and design of graphics, structure chart, diagrams and other foundation implementation of the software development. The purpose of this approach is to provide an overall view of system and interconnection between the modules. Microsoft Windows2000, Ms Visio, Ms Project Manager and Macromedia Fireworks MX are used for preparing proposal and other requirement representation such as Data Flow Diagram, Context Diagram, Structure Chart, Flow Chart, and ER Diagram

6.1.3 Software for System Development

Table 6.0 shows the type of software used throughout the development process as pertaining to the specific usage:

Software	Usage	Description
RedHat Linux 9.0	Development/Implementation Platform	Operating System
Microsoft Windows 2000	Development Platform	Interfaces are designed using this platform
Apache 2.0.48	Web Server	To host the web page
PostgreSQL 7.3.0	Database Server	Database design, construction and implementation for data
Galeon Web Browser	Web Browser	Web page viewing

Table 6.0: Development Software

6.2 Report Writing

All the problems encountered, together with solutions found throughout the processes (from system implementation until system evaluation) were recorded.

6.3 Development Strategy

Generally, IT Stuff Procurement System is developed using top-down strategy. This approach allows the higher-level modules to be coded first before the lower-level module. Top down design strategy is chosen because it provides the ability to simplify the problems by repeatedly decomposing problems into smaller size. This approach may reduce the problems faced in each development phase and shorten the development time.

In addition, modular approach is used where each module is developed separately and later integrated into the fully functional system. Some critical programs are coded separately in components and are tested.

6.4 Coding Principle

During the development of IT Procurement System web service, there are some techniques being used in process of coding and developing to ensure the codes are understandability and easy for maintenance.

6.4.1 Coding Approach

Top-down approach is chosen to break the system into functions and procedures. All these modules or functions are built and developed separately.

6.4.2 Coding Style

Coding style is an important attribute of source code and it determines the intelligibility of a program. An easy to read source code makes the system easier to maintain and enhance.

6.4.3 Code Documentation

Code documentation begins with the selection of identifier (variable and labels) names, continues with the composition of commenting and ends with the organization of the program. In the coding process, there are some codes that are new and complicated for those who read the codes. Comments provide a clear guide to understanding during the maintenance phase of software development. They also provide the developer with a means of communication with other statement will be stated down just above or beside the codes explaining the codes' logic and usage. Comments must be included for any line of code if the meaning of the line is ambiguous. Therefore, this will ease the person who reads the program and also for those who will continue working on the piece of codes.

```
function ver_item()              //validate form
{
  var errMsg = "";
  var errStatus = "False";

if (document.frmApp.jenis.value == "0")
```

```
{
  errStatus = "True";
errMsg +="* Please select type of item.  \n";
}     //end if

if (errStatus == "True")
{
 alert(errMsg);          //alert message for validate form
  return "false"
}        //end if

else
return "true";

}      //end function
```

Figure 6.0: Using comment statement to explain code

6.4.4 Use Indent to Arrange Codes In Order

One of the good programming practice is to arrange codes in a very readability way to Ease person who reads the codes. Besides that, arrange codes properly and in order will enable programmer(s) to look for errors and easy for maintenance.

```
<table width="750" border="0" bgcolor="#0f87ff" align="center">
    <tr>
        <td height="15">
            <div align="left"><font size="2" face="Courier New, Courier,
mono" color ="#ffffff" ><b>Application | Update Purchase Order</b></font>
            </div>
        </td>
    </tr>
</table>
```

Figure 6.1: Using indent to arrange codes in readability order

6.4.5 Using Meaningful Name for Representing Variable's Name

For each variable that is being used in the system, meaningful word should be used to represent a variable. For instance, the textbox for inserting address for vendor will be given names as "txtadd1". This is important for easy understand and to avoid confusing the person who read the codes. Good naming techniques also include using the consistent style of naming.

6.4.6 Modularity

Reduce complexity, facilities change and thus, results in easier implementation by encouraging parallel development of different parts of a system. It is measured using 2 quantitative criteria, cohesion and coupling. Software with effective modularity is easier to develop because function may be compartmentalized and interfaces are simplified. Independent modules are easier to maintain because secondary effects caused by design or code modification are limited, error propagation is reduced and reusable module are possible.

6.5 Development of IT Stuff Procurement System Web Service

6.5.1 Web Page Layout Development

Since the HTML is the standard web-based scripting language that marks up a web page with formatting command, therefore it is widely used in the IT Stuff Procurement System web page layout development. By using HTML, presentable web pages and images have been developed. In addition, HTML is also used in IT Stuff Procurement System to generate forms that enable system to collect data from administrator and other system users. For instance, checkbox, textbox, radio button and others form component are inserted into forms to perform specific tasks or functions. By using PHP (Hypertext Preprocessor) that fully supports HTML features, presentable web pages have been built.

6.5.2 Program Development

a). Client Side Scripting

Besides using HTML, client side scripting such as JavaScript are embedded in the HTML codes for further enhances the functionality of web pages. Most of the client side scripting are employed to perform interactive tasks at the client

side such as checking form completeness and validate user's input information. Client side scripting is widely used in the most of the forms in the system because it can improve the overall performance of the application to the web server for further processing. By using the scripting technology, it may free up server resource for other processing tasks and improve the overall web server performance.

b). Server Side Scripting

In IT Stuff Procurement System, PHP (Hypertext Preprocessor), as server side scripting is embedded in the HTML scripts to enhance the web page functionality. PHP will be the default server side scripting for IT Stuff Procurement System to execute logic on the server and produce consistent results regardless the browser used by client. This is very important as PHP have the capability to pass parameter from one page to other web pages. Scripting delimiters <? ... ?> have to be inserted for the server side execution. Codes located within these delimiters are invisible to the client and are only executed in the server.

i. PHP Sessions

To further enhance the functionality of the system, PHP sessions is used to manipulate data that needs to be shared between different pages. The function we use to register a persistent variable for use with PHP sessions is *session_register("SessionID")*. Given the name (without the dollar sign) of a variable, it will make that variable and its contents persist for the duration of the current session. Users will be first being assigned session to be able to login the system before they can browse the content and use the services. After the session has been terminated,

user is required to log on to the system again to continue browsing the content again. As a result, session is used as a system control in IT Stuff Procurement System to control the system from unauthorized access.

ii. The Use of Include File

PHP also has a directive that is similar to Server Side Includes (SSI) which is supported by most Web servers. If the command include("all_admin.php") is encountered in a PHP block, the specified file will be inserted and parsed by PHP. The PHP Include_Path variable in php.ini (PHP configuration file) was set to "/usr/share/php/include". For E-Procurement for IT Stuff, this Include file is use to display banner of every page, database connectivity, and the drop down menu of each module. Include files is used to simplifies work done during modification and correction on pages. Figure 6.2 shows how a include file define in a page.

```
<?php
session_start();

include("all_admin.php");
.
.
.
?>
```

Figure 6.2: Include file

6.5.3 Database Development

Specifying all the fields for each table and the field's property using PostgreSQL will create all the related database tables. After the creation of all the

database tables, appropriate data or information will be inserted into some of these database tables to initialize the system.

i. Database Connection

PostgreSQL offers a high power command line front end called **psql** for a DBA to manage the database. It has excellent transactional support and rich in datatypes. These PHP statements do the following:

`pg_connect()`

Establish a connection to the grocery database on the local server.

`pg_exec()`

Execute a SQL query and stores the output in a variable named, for example $result.

`pg_numrows()`

Returns the number of rows extracted from the table by the previous SELECT.

`pg_close()`

Close the contents of the previous pg_exec, which can be important when running several consecutive large queries.

`pg_fetch_array()`

Returns the data of each row from the table by the previous SELECT.

```
<?
function pg_open()
{
  global $SERVER_NAME, $REQUEST_URI;
  //global $Gdbhost, $Gdbname;

      $conn = pg_connect("host=202.185.109.141 dbname=procurement user=postgres
              password=100");
     //$conn = pg_pconnect($Gdbhost,",",$Gdbname);

              if (!$conn)
              {
              error_log("Unable to open $Gdbname while processing
               $SERVER_NAME".$REQUEST_URI);
              } //end if
```

```
        return $conn;

} //end function pg_open

        function pg_send($sql)
        {
                global $sql_debug;
                $conn = pg_open();

                if ($sql_debug)
                {
                 print "Sending on $conn: $sql<br>";
                 } //end if

                return pg_exec($conn, $sql);

        }  //end function pg_send
?>
```

Figure 6.3: Database connection

```
        <?php
        session_start();

        include ("all_vendor.php");

        $conn = pg_open();

        if (!$conn)
        {
          print("Database connection failed.");
          exit;
        }

        $q8 = "SELECT ven_comname FROM vendor WHERE ven_id = '$vid'";
        $result9=pg_send($q8);
        $values=pg_fetch_array($result9,0,PGSQL_ASSOC);
        $comname = $values['ven_comname']."";

      ?>

.
.
.
<?
$q19 = "Select item_name, apem_itemcode, apem_appcode from
        application_item,  item where apem_itemcode = '$koditem' and
        apem_itemnamecode= item_code and apem_appcode = '$appcode' ";

$hasil19=pg_send($q19);

$values19=pg_fetch_array($hasil19,0,PGSQL_ASSOC);
$iname = $values19['item_name'];
$icode = $values19['apem_itemcode'];
$acode = $values19['apem_appcode'];

    echo"<table width=\"600\"> <tr>";
```

```
        echo"<td width=\"600\"><font size=\"2\" face=\"Arial, Helvetica,
sans-serif\" color=\"#000000\"><b>APPLICATION CODE : <font size=\"2\"
face=\"Arial, Helvetica, sans-serif\"
color=\"#0000FF\">$acode</font>   ITEM NAME : <font
size=\"2\" face=\"Arial, Helvetica, sans-serif\" color=\"#0000FF\">$iname</font>
   </b></font></td></tr> ";

    echo"</table>";
?>
```

Figure 6.4: Display data in a Web Page

CHAPTER 7: SYSTEM TESTING

7.1 Introduction

The main purpose of testing is to establish the presence of defects in a program and to judge whether the program is usable in real application. Even so, testing can only demonstrate and discover the presence of faults or errors. There are several types of errors like syntax faults, algorithmic faults, computation faults, and some time system configuration is needed. It is hard to show that there is no error in a program. Therefore, a more suitable approach must be chosen to reduce the possibility of errors in a program.

The approach of system testing for *e-Procurement for IT Stuff* is bottom-up approach. When this method is used, each component at the lowest level of the system hierarchy is tested individually first. Then, the next components to be tested are those that call the previously tested one. This approach is followed repeatedly until all the components are included in the testing and successfully tested.

7.2 Testing Process

In general, the testing process of *e-Procurement of IT Stuff* can be shown in the following figure. All the detail will be further explained in the subsequence sections:

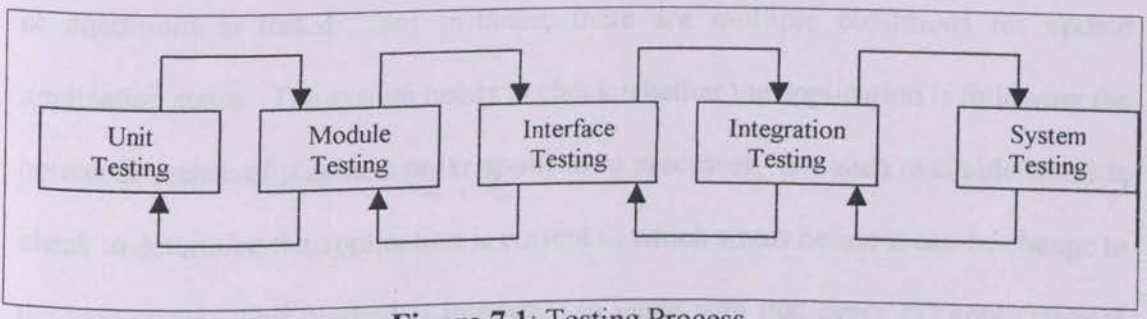


Figure 7.1: Testing Process

7.2.1 Unit Testing

Unit testing aims at the verification of the smallest unit within a program and ensure that they function properly. Each unit is tested independently without the interference from other system components. In *e-Procurement for IT Stuff*, most of the each modules contains sub-modules, which in turn consist of different function and units. These functions are individually tested before the entire application is tested.

7.2.1.1 White Box Testing

White box testing deals directly with the structure of codes within a module or segment and it requires the intimate knowledge of internal programs. Some of the code coverage is discussed below:

Segment Coverage Testing

Every segment of the code is rechecked to ensure that all of them are supposed to be executed at least once. In the login page, when user enters login identity and password, the system will authenticate his/her identity by checking through the database. Even though the identity might not concern other modules, but along the way the user might perform some other functions that relate to other modules as well.

Compound Condition Coverage

For multiple conditions that appear in the codes, every possible combination of conditions is tested. For instance, there are multiple conditions for update application status. The system needs to check whether the application is following the correct sequence of purchase order application processes. So, each available status is check to determine the application is current in which status before it can be change to the next status. This method is important to make sure that every PO application is following the correct sequence of PO application processes.

7.2.1.2 Black Box Testing

Black box testing involves testing of function of a module without knowing the logic structure of the code. Input is provided and out put is verified manually to check for accuracy. These conditions will fully exercise the entire functional requirement for this system. Black box testing uses an approach to uncover the different class of error compare to the white box testing method. Black box testing attempt to find error in the following categories:

- Interface error- for example, unable to establish a link to another page
- Performance errors-for instance, images appearing at the wrong path
- Initialize and termination error

Input Analysis

This approach is used when a set of input is given and its output is observed. In this strategy, different kinds of input are provided to test the compound statement. Every stated condition must be able to be executed and any unnecessary codes are eliminated. It is important to identify the variance between prototype and requirements.

For instance, a tester deliberately makes some mistakes or keys in value that is out of a acceptable range. This is done to see how the system will react and to determine whether the system will prompt the tester with the appropriate error message.

7.2.2 Module Testing

Module testing is performed without other system modules. A module consists of collection of dependent units to perform a particular task or function. Different possible test cases are applied to the module and the test results would be

verified. Unusual results will be analyzed and they would help in debugging sub-modules in order to produce the desired output.

7.2.3 Interface Testing

The interface should be user-friendly and not misleading. It is to ensure that the user understands what he/she is doing and what is the expected outcome. Instructions must be given in an appropriate manner and time. Error messages should be clear and straight to the point. However, this error messages should not incur any bad impact on the user, leaving discouragement to use the system.

Besides, the interface design should not lead the user to key in invalid entries. Data type like date format and date length should be controlled to avoid unnecessary problems. For example, the use of the JavaScript pop up calendar will help eliminate the problems of date format. For every input text, the maximum length must be set according to data property in the database.

7.2.4 Integration Testing

Integration testing is where combined modules that are dependent on one another are tested to determine if they function together as one. This is because integrated modules can be incorrect or inconsistent although the modules were individually proven satisfactory, as shown by the successful unit testing. Integration testing is specially aimed at exposing the problems that arise from the combination of modules. Variables and parameters passing are all tested during this phase.

Integration testing ensures that valid linking and dynamic relationship are established between sub-modules of the whole system. The testing is constructed and tested in small segments, where errors are easier to isolate and rectify.

Once functional test is completed, performance test is performed to compare the integrated modules with the non-functional system requirements. These

requirements include reliability, efficiency, maintainability, user friendliness and others to ensure that the system being developed is fully functional and optimized.

7.2.5 System Testing

System testing is to fully exercise the completed system. It is carried out on the entire integrated system as one unit to ensure that the entire system is validated. Its activities include testing of system performance, usability, data integrity, configuration sensitivity, and error handling and recovery.

The purpose of the overall system testing is to ensure and verifies that the system is function properly with all design and development objectives are met.

7.3 Conclusion

At the end of the testing phase, the system should be able to perform the tasks required and free of some errors. This is important to obtain user's confidence towards the system. However, there might have some problems and errors will occurred only after some time of using the system. Therefore, work of testing should not end up in this phase but have to keep on every now and then to make sure the system is functioning well if it is put fully in use.

CHAPTER 8: SYSTEM EVALUATION AND CONCLUSION

8.1 Introduction

Evaluation is the ultimate phase of developing a system and an important phase before delivery the system to the end user. At all phases of the system approaches, evaluation is a process that occurs continuously, drawing on a variety of sources and information.

Generally lots of technical and non-technical problems were encountered during the development stage. However, most of the problems were detected and resolved eventually but some are not.

8.2 Problems Encountered and Solutions

The *e-Procurement for IT Stuff* has unambiguous and straightforward deliverables and definitions. However, the process of developing the system is as challenging and demanding as any other development projects. Various difficulties ranging from minor setbacks to some considerable problems were encountered. The following subchapters will discuss the problems and approaches for solution in the *e-Procurement for IT Stuff.*

8.2.1 Problems in Development Environment, Language and Tools Selection

There are a lot of good and potential development tools available for the use of developing the system. However, not all these are suitable as each and every tool has its own strengths and weaknesses. The task of choosing the right development tool will always remain as the toughest question to answer. In searching for suitable combination of system development tools, some information finding methods and the system requirements are analyzed.

8.2.2 Lack of knowledge of the Development Environment, Language and Tools Chosen

Due to the time constraint, it is very difficult in learning the chosen language (PHP), operating system (Linux), and tools. Without a strong base of the language and the operating system, it need more time in looking for solutions to solve technical and non-technical problems that were encountered during the development of *e-Procurement for IT Stuff*. It consumes a lot of time in the beginning of development to learn the operating system commands. Some configurations needed to be done to PHP (php.ini), Apache web server (httpd.conf) and database (PostgreSQL-pg_hba.conf). All these need some research on the relative configuration files before knowing how to use them to do required configurations.

To solve these problems, Internet is the most vital source. There are lots of source codes, documentations and free tutorials in the World Wide Web. In addition, forum and mailing list is also another way to help to solve the problems occurred during the system development.

8.2.3 Problems in Reducing Script Size

System optimization is always an essential initiative in any system development. The purpose of system optimization is to increase its efficiency or to improve its performance without major upgrades on any hardware. As for *e-Procurement for IT Stuff*, these optimizations would be in the form of improving its execution time and response times. This can be theoretically achieved through a smaller program code size. The size of PHP files created range from 3 KB to 40 KB. The larger files are mainly related to the main pages, which consists of different access level, functionalities, verify and retrieval modules, and database controls. One of the reasons is that these pages contain complex multiple part coding with

comprehensive internal documentations and commenting codes. Moreover, the large file size is also contributed by various scripting conditions, extended SQL queries, and JavaScript event handlers.

8.3 System Strength

The *e-Procurement for IT Stuff* strength were recognized and identified. The *e-Procurement for IT Stuff* strength are described below:-

8.3.1 User-friendly Interface

The *e-Procurement for IT Stuff* has simple and easy to learn user interface, which could be, understand and used by new user. There are many features included to make the interface user-friendly such as text box, buttons, links and icons. These features will help the user to navigate and used the system. User-friendly interface will help to reduce the total cost of developing and implementation of the system because the training cost and training time required can be reduced. Most of the web pages are equipped with links and icons to help lessen user's effort in navigating through the system.

8.3.2 Provide Database Access

Maintenance tables are created to manage multi-value data. All the data collected are stored and organized by PostgreSQL postmaster. It provides a real-time database management. Changes made can be updated as soon as the changes being submitted. Besides, features and functions of *e-Procurement for IT Stuff* are easily expandable with tables added to its existing database design.

8.3.3 Simpler than Manual System

As what have been mentioned just now, the *e-Procurement for IT Stuff* is an automated version of the manual procurement system. But the *e-Procurement for IT Stuff* is much simpler to use and maintain compare to the manual system. *e-Procurement for IT Stuff* provides a paperless environment and one point data-entry. Data entered by user will be duplicated or retrieved whenever they are needed instead of requiring the user to re-type in the same information.

8.3.4 Efficient Data Manipulating

When accessing data or manipulating them, *e-Procurement for IT Stuff* is using stored procedures most of the time. This is done to improve performance for the system, where all the transaction is done at the database level, thus minimizing the waiting time for the user. Besides that, compare to the basic web application, using web service also minimize the process time at the client page where all the data transaction actually processed at the web service level.

8.3.5 System Transparency

The *e-Procurement for IT Stuff* system is transparent to the user. The user will have no idea how their request are being handle and processed. In fact, the user does not need to know the underlying structure of the system and database in making their request. All they need know is to submit data required and then view the results. This is important to avoid any confusion among users.

8.3.6 Reliability and Accuracy

The interface design is consistent throughout the whole system. A navigation bar is located on top for easier accessibility and brief instructions are provided for most of the processes.

e-Procurement for IT Stuff is also reliable because of the error tolerance provided by the system. For any fault created by user such as duplication of login ID, inappropriate keyword search or overlapping application, an error message for each process will be prompted to make user aware of the mistake.

8.4 System Constraints

Although much effort has been made to build the *e-Procurement for IT Stuff* to meet its requirements and minimizing the errors, the system still have it's own constraints. These constraints could not be avoided due to lack of experience in web-based system development and time limit. The system constraints are listed below.

8.4.1 Browser and Platform Limitation

It is recommended that this system run in Microsoft Internet Explorer 4.0 and above and Java Runtime Environment or Java Development Kit is required. The unsupport platform and browser combination is Mac and IE5 due to a known bug in the Mac version of IE. This limitation is due to usage of JavaScript calendar control. Web browser such as Netscape Navigator will also give different effect on layout design and some other functions may not work properly in this browser. In addition, Internet Explorer supports more scripting language such as JavaScript.

8.3.6 Reliability and Accuracy

The interface design is consistent throughout the whole system. A navigation bar is located on top for easier accessibility and brief instructions are provided for most of the processes.

e-Procurement for IT Stuff is also reliable because of the error tolerance provided by the system. For any fault created by user such as duplication of login ID, inappropriate keyword search or overlapping application, an error message for each process will be prompted to make user aware of the mistake.

8.4 System Constraints

Although much effort has been made to build the *e-Procurement for IT Stuff* to meet its requirements and minimizing the errors, the system still have it's own constraints. These constraints could not be avoided due to lack of experience in web-based system development and time limit. The system constraints are listed below.

8.4.1 Browser and Platform Limitation

It is recommended that this system run in Microsoft Internet Explorer 4.0 and above and Java Runtime Environment or Java Development Kit is required. The unsupport platform and browser combination is Mac and IE5 due to a known bug in the Mac version of IE. This limitation is due to usage of JavaScript calendar control. Web browser such as Netscape Navigator will also give different effect on layout design and some other functions may not work properly in this browser. In addition, Internet Explorer supports more scripting language such as JavaScript.

8.4.2 Lack of Advance Security Features

The security features implemented in the system are the most basic security features involving the session variables manipulation. Sessions variable are used to determine whether a user is granted to have access to the administrator functions or not. There are no advance security features implemented because of time constraint. In fact, for a web-based system, security issue is one of the most important issue in web based system development.

8.4.3 Limitation on Interface Design

As a formal application system, *e-Procurement for IT Stuff* is designed in such a way that suits the organization environment. The interface design is limited to avoid too many animation, sharp colors and funny images. It has to be impressive and yet professional looking. Features such as user friendliness have to be taken into consideration while developing on the user interface.

8.5 Future Enhancement

As *e-Procurement for IT Stuff* continue to evolve, future enhancement that will improve the overall system performance and functionality should be carried out. Therefore, it is hope that the following aspect could be considered in the future development of the system.

Urgent Notice Using E-mail

The super-administrator and administrator will send e-mail automatically to desire system user according to the e-mail address have been entered.

Announcement

Super-administrator or administrator can make some announcements to all the system users.

8.6 Current Enhancement

8.6.1 Changes in Database Table Design

The tables that have been designed previously are insufficient and some of them are incorrect due to the changes of system flow and logic. Most of the tables in the previous system design have been altered or totally changed in order to be compatible with the modularity of the web service. Some new tables are added due to system requirements. These changes are due to avoid redundancy of data.

8.7 Knowledge and Experience Gained

Knowledge (technical or non-technical) has been experienced through the entire development of *e-Procurement for IT Stuff*. Some of the knowledge and experience gained are listed below:-

8.7.1 Application Knowledge

The development of this project involves various applications such as Apache web server, PostgreSQL, PHP, Macromedia Fireworks, and Ulead PhotoExpress. A part from that, some knowledge on how to write PHP, JavaScript, and command of OS Red Hat Linux is also gained throughout the development phases of *e-Procurement of IT Stuff*.

8.7.2 Ability to Set Up Platform and Server

In order to get Linux packages (PHP, PostgreSQL and Apache) to work together, some configuration need to be done for each of the packages. From the system development, some basic knowledge of these configurations are gained.

8.8 Summary

Chapter 8 listed out the problems and solutions used to solved the problems. Most of the problems could be solved in order to make sure that the *e-Procurement of IT Stuff* will meet its requirements. Due to time constraint, some problems could not be solved at the moment. Problems that could not be solved have been expressed in the future enhancements. Future enhancements list out the suggested features, functions and changes need to be implemented if the *e-Procurement of IT Stuff* would be used as a full working system in the future.

Opinion and advice have been sought from few respondents during the system testing. Their opinions and suggestions have been taken into considerations in making the *e-Procurement of IT Stuff* more reliable. Some of the suggestions are listed in the future enhancements while some already being implemented.

The *e-Procurement of IT Stuff* strength and constraints are also being listed out. These strength and constraints are being identified during the system development and evaluation.

By developing the *e-Procurement of IT Stuff* system, a lot of new programming techniques and knowledge are gained. This knowledge can be used in solving future problems and in developing similar system.

8.9 Conclusion

Overall the *e-Procurement of IT Stuff* has achieved and fulfills all the requirements and objectives suggested. The significances of the *e-Procurement of IT Stuff* will be seen if the system could be implemented in a real working environment.

A lot of research and studies have been made during literature review. All the information gathered has been converted into a solution and could be used for future reference in developing a similar system.

Throughout the *e-Procurement of IT Stuff* development process, a lot of experience and knowledge acquired. And this will help in future effort in developing a web-based system and working aspect. This knowledge gained is very valuable and has improved the programming and ability to solve problems.

e-Procurement of IT Stuff system is designed in a proper manner so that it can be adopted at any time for any organization. It is hope that this system will be a success and provide a foundation upon which more innovative and comprehensive system may be built to perform multiple tasks and fulfill various user requirements.

REFERENCE

BOOKS

1. Choi, W., Kent, A., Lea, C., Prasad, G., Ullman, C., Blank, J., Cazzell, S. (2000). *Beginning PHP4.* Birmingham, U.K.: Wrox Press Ltd.
2. Kendall, Kenneth E., and Kendall, Julie E. (1996). *System Analysis and Design.* 4th edition. California: Prentice-Hall, International, Inc.
3. Pfleeger, Shari L. (2001) *Software Engineering: Theory and Practice.* 2nd Edition. New Jersey: Prentice-Hall, International Inc.
4. Powell, T. A., Schneider, F. (2001) *JavaScript: The Complete Reference.* New York: McGraw-Hill.
5. Tai, C.H. (2002). Sales Force Automation. Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur.

WEB SITE

1. http://www.w3schools.com
2. http://www.redhat.com
3. http://www.postgresql.org
4. http://www.php.net
5. http://www.apycom.com
6. http://www.webopedia.com
7. http://www.act.com
8. http://www.apache.com

APPENDIX A

USER MANUAL

USER MANUAL – TABLE OF CONTENT

1. INTRODUCTION

E-Procurement for IT Stuff system was developed to solve the problem of the current system facing throughout years which processes and tracks all purchase orders manually. *E-Procurement for IT Stuff* is an online procurement system for a middle or large size organization. If this system put into operation the process of purchasing IT related item(s) could be easily done. This system provide data access environment to the department and IT vendor. Each department and IT vendor can easily apply PO application in the short time. This is faster if compare to current system which is time consuming.

2. SUPER ADMIN

-Super admin must have username and password before login.

-Button in login page:

Button	Function
Submit	Link to **Create New Admin** page.
Cancel	Reset all the data.

-Super Admin Account login page.

Username	Type your username.
Password	Type your password.
Submit button	To login to your account.
Cancel button	To reset all the data.

-If login success, ***New Admin*** page will display. Else an error message will be display.

2.1 Admin

-From the drop down menu, click **Admin** button to get to the desire page.

-Links at **Admin** drop down menu:

Link	Function
New	Link to ***New Admin*** page.
Update	Link to page giving all the *admin name* and *department* to choose.

2.1.1 Admin - New

-Super admin can create new admin.

-This page will ask user to fill in a form for new admin's data.

Username	Create username for new admin. (This username is used for admin's login ID)
Password	Create password for new admin.
Confirm New Password	Retype the new password.
Department	Select the department for new admin.
Name	Type the name of new admin.
Salary No	Type the salary no for new admin.
Tel No	Type the telephone no for new admin.
Fax No	Type the fax no for new admin.
Email Address	Type the email address for new admin.
Cancel	To reset all the data.

-After finish fill the data about new admin, click **Update** button.

-Message "**New admin created successfully.**" will be display if data has successfully inserted into database.

All the field must be fill in except **Fax No (optional). **Username** and **Salary No** must be unique.

2.1.2 Admin - Update

-This page will display all the *Admin Name* and *Department*.

-Super admin can edit any admin by click at the  image.

-Then, user will be at the ***Update Admin*** page. User can update the data as shown in text box below.

Name	Display the name of the selected admin.
Department	Select the department for the existing user.
Salary No	This field is *disabled.*
Tel No	Display phone no.
Fax No	Display fax no.
Email Address	Display email address.
Status	Indicate the status of selected admin (active or inactive).

Cancel button	To reset all the data.

-After update data, click **Update** button to save changed.

-Message "**Admin's data has been updated.**" will be display if data are updated.

Admin can only login to their account if their status are **Active, else they cannot access to the system.

2.2 Task Distribution

2.2.1 Task Distribution - New

-User will first come to page entitled **New Task Assignment**.

-This page allows you to assign anyone of the administrator to in-charge of certain application.

-First of all, select an application from the drop down list.

-Then, select the administrator who will in charge of that particular application.

-Finally, click on button **Assign** to save this task distribution.

-User will see the list of task distribution at the bottom of this page.

-Repeat the steps above in order to assign more task(s).

2.2.2 Task Distribution - Update

-User will come to page entitled **Update Task Assignment**.

-This page allows user to edit or delete certain distributed task in the list mentioned above.

-Click on image  to edit or re assign administrator for a particular application.

-User will come to page entitled **Edit Task Assignment**.

-Select administrator from the drop down list to assign a particular application (task).

-Click on **Save** button to save changes.

-Click on image  to delete or to cancel task assignment from the list.

2.2.3 Task Distribution - View - By Department

-User will come to page entitled **Application Status | By department.**

-Select department from the drop down list to view related information.

-Click button **Submit** after select desire department.

2.2.4 Task Distribution - View - By Staff

-User will come to page entitled **Application Status | By staff.**

-Select department from the drop down list to view related information.

-Click button **Submit** after selection.

2.2.5 Task Distribution - View - By Status

-User will come to page entitled **Application Status | By status.**

-Select status from the drop down list to view related information.

-Click button **Submit** after selection.

2.3 Change Password

-When click on **Change Password** button at the drop down menu, user will see the ***Change Password*** page.

-User can change his/her old password to a new password.

Super Admin Name	This field is *disabled.*
New Password	Type your new password.
Confirm New Password	Retype your new password.
Cancel button	To reset all the data.

-After finish change your data, click **Update** button.

-Message "**Your password has been updated.**" will be display.

2.4 Logout

-Click **Logout** button to logout account each time before exit for security purpose.

3. APPLICANT

-Applicant (User) must own username and password before login.

-Button in login page:

Button	Function
Submit	Link to ***New Purchase*** page.
Cancel	Reset all the data.

-Login to Account Client

Username:	Type your username.
Password:	Type your password.
Click **Submit** button to login to your account.	
Click **Cancel** button to reset all the data.	
If successfully login, ***New Purchase*** page will be display.	

3.1 Application

-When click **Application** button you will see the drop down menu.

-Below are the links available:

Button / Link	Sub Link	Function
Application	New	Link to ***New PO Application*** page.
	Edit	Link to ***Update PO Application*** page.
	Delete	Link To ***Delete PO Application*** page.
Tag Number	Create/Edit	Link to ***Create/Edit Tag*** page.
Status List	By Code	Link to ***Find By Code*** page.
	By Status	Link to ***Find By Status*** page.

Take note of this button link:

	Image "scan files". (Edit)
	Question mark. (Not Available)
	Tick. (Available)
	Trash bin (Delete)

3.1.1 Application - Application

3.1.1.1 Application - Application - New

-The page will display applicant's data. Applicant can make a new PO Application by filling in the following information:-

Application Code:	Application code generate by database (disabled).
Name:	This field display applicant's name (disabled).
Department:	This field display applicant's department (disabled).
Budget (RM):	The amount of money that is budget for this purchasing.
Type:	Type of item that applicant want to purchase.
Item Name:	The item's name that applicant wish to purchase.
Item Description:	The specification of selected item.
Qty:	The quantity of item.
Add Item:	A button that allow applicant to add more item.
Save:	To save application.

-After applicant click save button then user are require to click a link to page for adding tag information of item(s) by click  . Click save button when done.

-  will appear if tagging information is not available and ✓ sign indicates tagging is done.

-User can also delete the item tagging information by clicking  and a prompt will ask whether applicant whether to delete it. If applicant wants to delete the info just simply click ok. Otherwise click button cancel to end action.

3.1.1.2 Application - Edit

-Applicant have to enter application code that he/she wants to edit then click submit. A new page will be appearing.

Application Code:	Display client's application code.
Name:	Display applicant's name (disabled).
Department:	Display user's department (disabled).
Budget (RM):	The amount of money that budget for purchasing.
Type:	Type of item that applicant want to purchase.
Item Name:	The related item's name.
Item Description:	The specification of selected item.

Qty:	The quantity of item.
Add Item:	A button that allow applicant to add more item.
Save:	To save application.
Scan Files image	An image that allow applicant to edit item.
Trash bin image	Button that allow applicant to delete (A prompt will ask whether applicant want to confirm the action. Click ok if applicant to proceed or cancel to cancel the action).

-If applicant clicks "edit" a new window will appear that need applicant to edit type of item, item name, item description and quantity of item. After all these information had been provided applicant must click button *save & back* that will save the changes and back to previous page.

3.1.1.3 Application - Delete

-Applicant must enter application code that need to be delete then click submit. New page will appear that display the info that related to the action.

Application Code:	Display client's application code.
Name:	Display applicant name (disabled).
Department:	Display applicant's department (disabled).
Budget (RM)	Display budget (disabled).
Type	Display type of item (disabled).
Item Name:	Display item's name (disabled).
Item Description:	Display Item description (disabled).
Qty:	Display the amount of item (disabled).
Button Delete:	This button will delete user application (a prompt will appear that ask whether user want to delete the application. Click *ok* to delete or cancel to cancel the action).

3.1.2 Application - Tag Numbers

3.1.2.1 Application - Tag Numbers - Create/ Edit

-When click tag number a page require applicant to enter application code that need tagging. After click button *submit* then a page contains these info will be opened. Following are some elements within this page:-

Application Code:	Display applicant's application code.
Name:	Applicant's name (disabled).
Department:	Applicant's department (disabled).
Budget (RM):	Budget of this PO application (disabled).
Type:	Type of item (disabled).
Item Name:	Related item's name (disabled).
Item Description:	Item's description (disabled).
Qty:	The quantity of item.
Save/Save tag:	Save the changes.
Print Tag:	This button enables to print the tag.
Tag:	To check whether user have done item tagging.
Image "Q's Mark"	This image will appear if tagging has not been done yet.
Image tick	This image will appear if tagging has been done.
Image "Scan Files"	Button that allow applicant to edit or do tag number and new window will appear where applicant are require to add tag number.
Image "Trash bin"	Button that allow applicant to delete tagging info (a prompt will appear that ask whether applicant want to delete the application. Click *ok* to delete or cancel to cancel the action)

3.1.3 Application - Status List

-Applicant can choose to display application status list by code or by status.

3.1.3.1 Status List - By Code

-Applicants must enter application code if they choose to view status list these way then click the submit button. A new page will appear that contains these following information.

Applicant code	Display applicant code.
Department	Display applicant's department.
Applicant Name	Display username.
Budget	Display budget.
Application Date	Display applicant's date.
Code	Display item code.
Item Type	Display item type.
Item Name	Display item name.
Item detail	Display item details.
Qty	Display quantity of item.
Tag list	Generate tag list.
View all status	A link that allow user to view all the status.
Type of tender	Display type of tender for item.

3.1.3.2 Status List - By Status

-For these function user just have to select a status then click search button then anew window will appear that contain.

Application code:	Display the codes that match the chosen status.
Date	The exact date the application been mad.
View	A link that show application status.

3.1.3.2.1 Status List - By Status - Application status

-This page will display this information.

Application code	Display applicant code (disabled).
Department	Display applicant's department (disabled).
Applicant Name	Display applicant's name (disabled).
Budget (RM)	Display applicant's budget (disabled).
Application Date	Display date application been made (disabled).
View All status	A link that view all status.
Type of tender:	Generate by database (ex: close tender & acceptance system).

3.1.3.2.2 Status List - By Status - Application status - Define specification

-This page will display this information.

Expertise:	Display expertise of company.
Date:	Date the application been made.

View status:	View whether client can view information that been made.
Item code:	The item code.
Item Name:	The name of item that user had choose.
Spec Name:	The specification name.
Spec Detail :	The specification details.

3.2 General

-When click **General** button, you will see the drop down menu.

-Link at drop down menu under Miscellaneous:

Link	Sub Link	Function
User	**Update**	Link to ***Update User*** page.
	Change Password	Link to ***Change Password*** page
Item Type	**New**	Link to ***New Item Type*** page.
	Update	Link to page showing all the *item type*.
Item	**New**	Link to ***New Item*** page.
	Update	Link to ***Search Item*** page.

3.2.1 General - User

3.2.1.1 General - User - Update

-The page will display applicant's data.

-Applicant can update his/her own data in the below text box.

Name	Display the name of the applicant.
Department	Display the department of the applicant.
Salary No	This field is disabled.
Tel No	Display phone no.
Fax No	Display fax no.
Email Address	Display email address.

-After data has been updated, click **Update** button to save changes.

-Message "**User's data has been updated.**" will be display if changes successfully saved.

3.2.1.2 General - User - Change Password

- Applicant can change his/her old password to new one.

User Name:	This field is disabled.
New Password:	Type applicant's new password.
Confirm New Password:	Retype applicant's new password.
Cancel button	To reset all the data.

-After finish change data, click **Update** button.

-Message "**Your password has been updated.**" will be display if changes successfully saved.

3.2.2 General - Item Type

3.2.2.1 General - Item Type - New

-Applicant can create new item type.

Item Type Code	This code is auto generate.
Item Type Name	Create new item type name.
Cancel button	To reset all the data.

-After finish create new item type, click **Save** button.

-Message "**New item type's data has been saved.**" will be display if data successfully saved.

3.2.2.2 General - Item Type - Update

-This page will display all the Item Type.

-Applicant can edit the item type by click at the image "scan files".

-Then applicant can update the data as shown in table below:-

Item Type Code	Item type code (disabled).
Item Type Name	Display the item type name that applicant want to edit.
Cancel button	To back to the previous page (page that display all the Item Type).

-After updatesd, click **Update** button.

-Message "Item Type's data has been updated." will be display if data successfully saved.

3.2.3 General - Item

3.2.3.1 General - Item - New

-Applicant can create new item.

Item Code	This code is auto generate.
Item Type	Select the item type for item.
Item Name	Type new item name.
Cancel button	To reset all the data.

-After new item created, click **Save** button.

-Message "**New item's data has been saved.**" will be display if data successfully saved.

3.2.3.2 General - Item - Update

-This page will ask applicant to select the item type, which applicant want to update the item name.

-Then click the **Search** button, all the *Item Code* and *Item Name* for that item type will be display.

-Applicant can edit the item name by click at the image "scan files".

-Now, can update the data in the below text box.

Item Code	This field is disabled.
Item Type	Applicant can reselect the item type.
Item Name	Display the item name that applicant want to edit.
Status	Applicant can check either one of the radio button (*Active or Non-Active*).
Cancel button	To reset all the data.

-After finish update item name, click **Update** button.

-Message **"Item's data has been updated."** will be display.

**If item's status is *Active*, then applicant can purchase the item, else applicant cannot purchase the item.)

3.3 Logout

-Click **Logout** button to logout each time before exit.

4. ADMIN

-Admin must have username and password before login.

-Button in login page:

Button	Function
Submit	Link to **Add Status** page.
Cancel	Reset all the data.

Login to Account: -

Username	Type your username.
Password	Type your password.

-Click **Submit** button to login to your account.

-Click **Cancel** button to reset all the data.

-If success login, ***Add Status*** page will display.

4.1 Vendor

-Links under the Vendor drop down menu: -

Link	Function
1. New	Link to page create new vendor.
2. Update	Link to page for update vendor's information.
3. Create Password	Link to page create login id for vendor.
4. Search	Link to page for searching vendor.

4.1.1 Vendor - New

-After click on **Vendor** button, you'll come to page, which Admin can create new vendor.

Or

- Admin will come to this page when you click on **Vendor** button then click *New* on the sub menu.

-This page required vendor to type in *Expertise Info* and *Company/Branch Info.*

Vendor Code	Vendor code is auto generated.
Vendor Name	Type in company's name.
Register No	Type in register number for this company.
Company Size	Select the company size from the drop down list, which has option like large, middle or small.
Expertise	Click on the "+" button to select company expertise.

-When admin come to page to select expertise for vendor, admin can click on the check boxes to select desire expertise.

-Click on *Select All* if admin want to select all the expertise.

-Click on *Remove All* to uncheck.

-Click on **Submit** button if done. Admin will visit the previous page but this time with the expertise name, which vendor had chosen.

-Click **Add Branch** button at the bottom of this page to add branch/branches information

before save.

- Admin will visit a page, which require admin to enter branch information.

Branch Name	Type in branch name (e.g. Kuantan)
Branch Address	Type in address of this branch.
Tel No.	Type in telephone number.
Fax No.	Type in fax number.
Email Address	Type in email address.
Representative	Type in representative vendor of this branch.
Hand phone No.	Type representative's hand phone number.

-Click on button **Save** after filled up all the required information.

-Then you will come to a page, which will display all the information you had entered.

- Admin can click on  button to edit branch information.

>>On this page, change related information and then click **Update** button. Then admin will go back to the previous page but with the updated branch information.

>>Click on **Cancel** button to reset.

>>Click on **Previous Screen** button to go back to the previous visited page.

Or

-Click on **Add Branch** button to add more branches.

-Click on **Save** button after all related branch information has been added

>>If admin had successfully save his/her data, then admin will receive message saying that branch info has been saved.

>>Else admin will receive a message, which state that his/her data CANNOT be saved.

-Click on **Cancel** button to cancel all procedure of adding this vendor.

** This act will erase all the information you had saved or cancel adding this vendor to the vendor list.

4.1.2 Vendor - Update

-Admin will come to this page when admin click on ***Vendor*** button then click ***Update*** on the sub menu.

-There two ways for searching vendor, either **(A)** Using **Search** or **(B)** Using **Advance Search.**

(A) Using **Search**: -

-Admin are requiring entering vendor's information: -

Vendor Name	Type in vendor's name, which Vendor wish to update it's Information.
Register No	Type in register number for that particular branch.

-Click on button **Search** after typed in the required information.

-Then admin will see vendor's info and branch info on the next page.

-Change related information then click on **Save** button to save any changes.

-Click on **Add Branch** button enables admin to add new branch information to this vendor.

-Click on **Save** button to save data.

-Click on **Previous Screen** button to go back to the previous page.

-Click on image pencil to edit branch information.

-On this page (edit branch), change related branch detail(s). Then click on **Update** button to save any changes.

-Click on **Previous Screen** button to go back to the previous page.

-Click on image trash bin to delete selected branch information.

-One prompt will ensure that admin want to delete selected branch information.

-Click on **Ok** button, all the selected branch information will delete and will lose those data permanently.

Or

-Click on **Cancel** button to cancel this act.

-Click on **Save** button to save all the data.

-Click on **Cancel** button to cancel all the information admin had change and return the original data.

(B) Using **Advance Search**: -

-This will provide much easier way to search for the desire vendor.

-Click on **Advance Search** button.

-Admin will visit a page, which require admin to type in vendor's name.

-Click on **search guide**, which will open a new window contains guide for searching vendor's name.

-Click on **Search** button to search for desire vendor.

-Then admin will come to a page showing all related vendor name.

-Click on image pencil to edit/update selected branch information.

-Change related information then click on **Save** button to save any changes.

-Click on **Add Branch** button enables admin to add new branch information to this vendor.

-Click on **Save** button to save data.

-Click on **Previous Screen** button to go back to the previous page.

-Click on image pencil to edit branch information.

-On this page (edit branch), change related branch detail(s). Then click on

-**Update** button to save any changes.

-Click on **Previous Screen** button to go back to the previous page.

-Click on image trash bin to delete selected branch information.

-One window will pop out to ensure that you want to delete selected branch information.

-Click on **Ok** button, you will delete all the selected branch information and those data will lose permanently.

Or

-Click on **Cancel** button to cancel this act.

-Click on **Save** button to save all the data.

-Click on **Cancel** button to cancel all the information you had change and return the original data.

4.1.3 Vendor - Create Password

-Admin will come to this page when he/she click on **Vendor** button then click **Create Password** on the sub menu.

-Admin are required to enter/select the following information: -

Vendor Name	Select vendor name from the drop down list.
User Name	Create/type in user name of vendor.
Password	Create/type in password of vendor.

-Click on **Save** button to save data.

-If admin see a new page displaying **Login account for this vendor successfully created** that's mean login id has successfully created.

Or

-If admin see a new page displaying **Login account already created for this vendor,** which means login id has been created for this vendor.

Or

-If admin see a new page displaying **This username already in use. Please enter other**

user name, which means he/she need to create a unique user name for this vendor.

-Please repeat the same steps to create login id for this vendor.

4.1.4 Vendor - Search

-Admin will come to this page when he/she click on **Vendor** button then click **Search** on the sub menu.

-This will provide much easier way to search for the desire vendor.

- Admin will visit a page, which require he/she to type in vendor's name.

-Click on **search guide**, which will open a new window contains guide for searching vendor's name.

-Click on **Search** button to search for desire vendor.

-Then admin will come to a page showing all related vendor name.

-Click on image pencil to edit/update selected branch information.

-Change related information then click on **Save** button to save any changes.

-Click on **Add Branch** button enables you to add new branch information to this vendor.

>>Click on **Save** button to save data.

>>Click on **Previous Screen** button to go back to the previous page.

-Click on image pencil to edit branch information.

>>On this page (edit branch), change related branch detail(s). Then click on **Update** button to save any changes.

>>Click on **Previous Screen** button to go back to the previous page.

-Click on image trash bin to delete selected branch information.

>>One window will pop out to ensure that admin want to delete selected branch information.

>>Click on **Ok** button, the selected branch information will be delete and will lose those data permanently.

Or

-Click on **Cancel** button to cancel this act.

-Click on **Save** button to save all the data.

-Click on **Cancel** button to cancel all the information admin had change and return the original data.

4.2 Application

-When click **Application** admin will see the drop down menu.

Link	Sub Link	Function
Application Form	**Update Application**	Link to ***Update Application*** page.
	Update Tag	Link to ***Update Tag*** page.
	Delete Application	Link to ***Delete Application*** page.
Application Status	**Add Status**	Link to ***Add Status*** page.
	Update Status	Link to ***Update Status*** page.
Application List	**By Department**	Link to ***Choose by Department*** page.
	By Status	Link to ***Choose by Status*** page.

Take note of this button link:

	Image "scan files" (Edit)
	Q's Mark (Not Available)
	Tick (Available)
	Trash bin (Delete)

4.2.1 Application - Application Form

4.2.1.1 Application - Application Form - Edit

-The page will ask you to input your *application code*. After you click **submit** button, the page will display *Update Purchase*. You can update your own data in the below text box.

Application Code	This field is *disable*.
Name	This field is *disable*.
Department's	This field is *disable*.
Department's Vote	Display the department of the admin.
Budget	Display Budget.
Type	Display type of item.
Item	Display item name.
Item Description	Display information about item that has been choose.

-Click on **image "scan files"** to edit item. When you click on **pencil button**, it will display *edit item* page. You can update your own data in the below text box.

Type	List of Type Item.
Item Name	List of item name.
Item Description	Information of item that has been choose.
Quantity	Quantity item that has been choose.

-After finished edit item, you must click **Save & back** button. It will go to the ***Update Purchase*** page.

-Click on **Trash Bin** to delete item. When you click on **Trash Bin**, it will ask you "Are you want to delete this item?". Click **Ok** if you want to delete item or click **Cancel** if not.

-After finish key in item, click button **Add Item** or you can add more item and click that button again.

-After finish update data, click button **save.** Message **Purchase application saved successfully** will be display. After that, you must click on **Need to create tag**

numbers to create tag numbers. It will display ***New Tag*** page. You can update your own data in the below text box.

Application Code	This field is *disabled.*
Department's	This field is *disabled.*
Allocation	Display allocation.
Type	Display type of item.
Item Name	Display item name.
Tag	Display or .

-Click on **image "scan files"** to edit tag. When you click on **image "scan files"**, it will ***display edit tag*** page. You can update your own data in the below text box.

Tag No	Display Tag No.

-After finished edit tag, click **Save** button or click **Previous Screen** if you do not want to edit tag.

-Click on **Trash Bin** to delete tag. When you click on **Trash Bin**, it will ask you "Are -you want to delete this tag?". Click **Ok** if you want to delete tag or click **Cancel** if not.

-After finished edit data for new tag, click **Save** button or click **Print Tag** button to print tag. When you click **Save** button, it will display **Tag numbers saved successfully**

4.2.1.2 Application - Application Form - Delete Application

-The page will ask you to input your *application code.* After you click **submit** button, the page will display ***Update Purchase*** page. You can update your own data in the below text box.

Application Code	This field is *disable*.
Department's	This field is *disable*.
Name /Type Of Vote	Display the type of vote of the admin.
Department's Vote	Display the department of the admin.
Allocation	Display allocation.
Type	Display type of item.
Item Name	Display item name.
Item Description	Display information about item that has been choose.
Qty	Display quantity of the item.

-The page will display **This application is NOT READY to process or This application is IN process.**

-Click **Delete** button if you want to delete that tag. Then you click **delete** button, it will ask you "Are you want to delete this tag?". Click **Ok** if you want to delete tag or click **Cancel** if not. If you click **Ok,** it will display **"Application A0000003 successfully deleted."**

4.2.2 Application – Application Status

4.2.2.1 Application - Application Status - Add Status

-The page will display ***New Application*** *page*. You can update your own data in the below text box.

Application Code	This field is *disable*.
Department's	This field is *disable*.
Applicant Name	Display name of that applicant .
Vote	Display the vote of the admin.
Allocation	Display allocation.
Application Date	Display date that application.
Item Code	Display item code of the item.
Item Name	Display item name.
Item Quantity	Display Item quantity.
Purchase Type	Display type of purchase.
Purchase Group	Display /List of the purchase group.

-Please tick (✔) at the checkbox to select one item or click **Select All** if you want to choose all item.

-Click **Remove All** if you want to remove all item.

-Click **Add** button if finished update data. It will display ***New Application*** page. You can update your own data in the below text box.

Application Code	This field is *disable*.
Applicant Name	Display name of that applicant .
Department's	This field is *disable*.
Vote	Display the vote of the admin.
Allocation	Display allocation.
Application Date	Display date that application.
Item Name	Display item name.
Close Tender :-	
Item Code	Display item code of the item.
Item Name	Display Item Name.
Item Quantity	Display Item quantity.
Purchase Type	Display Type of purchase.
Purchase Group	Display/ List of the purchase group.
Acceptance System:-	
Item Code	Display item code of the item.
Item Quantity	Display Item quantity.
Item Name	Display Item Name.
Purchase Type	Display Type of purchase.
Purchase Group	Display/ List of the purchase group.

-

Please tick (✔) at the checkbox to delete one item or click **Select All** if you want to delete all item. Then, click **Delete** button. You also can click **Remove All** to remove.

click **Delete** button. You also can click **Remove All** to remove.

-After finished, click **Save** button. When you click **Save** button, it will display ***Update Procurement Status***. You can update your own data in the below text box.

Application Code	This field is *disable*.
Department's	This field is *disable*.
Applicant Name	Display name of that applicant .
Vote	Display the vote of the admin.
Allocation	Display allocation.
Application Date	Display date that application.
Type of Tender	Display type of tender A) Close Tender. B) Acceptance Tender.
Purchase Group	Display/ List of the purchase group.
Please select status	List of the status. You must select status that following correct procedure sequence.

A) **Close Tender :-**

* Please tick (✔) at the checkbox to allow user view the status.

1 First, please select **Define Specification**, it will display ***Define Specification*** page.

You can update your own data in the below box.

Expertise	List of the expertise and you must select one.
Date	Date that you input the data.
Item Code	Display item code.
User spec	Display User specification.
Quantity	Display quantity of the item.
Specification	Display or .

-Click on **pencil button** to edit specification. When you click on **pencil button**, it will display ***Detail Specification*** page. You can update your own data in the below box.

Specification Name	This field ask you to input the specification.
Specification Description	This field ask you to input the specification description.

-After finished input your specification, click **Add Specification** button. When you click **Add Specification** Button, it will display :-

Specification Name	Display specification.
Specification Description	Display the specification description.

-Click on **pencil button** to edit specification. When you click on **pencil button**, it will display ***Edit Detail Specification*** page. You can update your own data in the below box.

Specification Name	This field ask you to input the specification .
Specification Description	This field ask you to input the specification description. or change the data.

-Click on **Trash Bin** to delete specification. . When you click on **Trash Bin**, it will ask you "Are you want to delete this Specification?". Click **Ok** if you want to delete Specification or click **Cancel** if not.

-After edit Specification, click **Edit Specification** button. It will display ***Detail specification*** page.

-Click on **Trash Bin** to delete specification. . When you click on **Trash Bin**, it will ask you "Are you want to delete this Specification?". Click **Ok** if you want to delete Specification or click **Cancel** if not.

-After finished edit specification, click **Save** button. When click **Save** button, it will display ***Define Specification*** page.

-After finished update specification, click **Save** button or you can click **Previous Screen** button if no data want to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

2 Please select **Document Tender,** it will display ***Document Tender*** page. You can update your own data in the below box.

Application Code	List of the expertise and you must select one.
Item Code	Display item code.

Quantity	Display the quantity of item.
Tender No	This field ask you to input tender no.
Date	Input your date.
User spec	Display user specification.

-After finished, click **Save** button or you can click Previous button. When you click **Save** button, it will display ***Update Procurement Status*** page.

3 Then select **Call Vendor**, it will display ***Call Vendor*** page. You can update your own data in the below box.

Date	This field ask you to input date.
Vendor Name	This field ask you to choose vendor. After choose vendor, click **Add Vendor**. Do this again to add more vendor

-Please tick (✔) at the checkbox to delete one vendor or click **Select All** if you want to delete all vendor. Click **Remove All** to remove.

-After finished, click **Save** button or you can click **Previous Screen** button if no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

4 Then select **Close tender**, it will display Close Tender page. You can update your own data in the below box.

Approved By	This field ask you to input the 1st witness.
Supported By	This field ask you to input the 2nd witness.
Date	This field ask you to choose date.

-After finished input your data, click **Save** button or you can click button **Previous Screen** if there is no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

5 Then select **Suggestion**, it will display ***Suggestion*** page. You can update your own data in the below box.

Date	This field ask you to choose date.
Vendor Name	This field ask you to choose vendor.
Price	This field ask you to input your price.

-After choose Vendor and Price, click **Add Vendor**. Do this again to add more vendor.

-Please tick (✓) at the checkbox to delete one vendor or click **Select All** if you want to delete all vendor. Click **Remove All** to remove.

-After finished, click **Save** button or you can click **Previous Screen** button if no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

6 Then select **Send to Bursary**, it will display ***Send to Bursary*** page. You can update your own data below box.

Date	This field ask you to choose date.

-After finished, click **Save** button or you can click **Previous Screen** button if no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

7 Then select **Receive Approval Letter form Bursary**, it will display ***Receive Approval Letter form Bursary*** page. You can update your own data in the below box.

Reference No	This field ask you to input the reference no.
Vendor Name	This field ask you to choose vendor with their price.
Date	This field ask you to choose the date.

-After finished, click **Save** button or you can click **Previous Screen** button if no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

8 Then select **Agreement**, it will display ***Agreement*** page. You can update your own data in the below box.

Date	This field ask you to choose date.

-After finished, click **Save** button or you can click **Previous Screen** button if no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

9 Then select **Produce PO**, it will display ***Produce PO*** page. You can update your own data in the below box.

No PO	This field ask you to input No PO.
Vendor Name	Display the vendor name.
Branch Name	This field ask you to choose branch of that vendor.
Send to	Display department
Officer name	Display the officer name
Date	This field ask you to choose date.
Note	This field ask you to input note.
Price/Unit	This field ask you to input price for 1 item.

Total Price	Display the total of price.

-Click **Print PO** button to print PO .

-After finished, click **Save** button or you can click **Previous Screen** button if there is no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

10 Then select **Call Chosen Vendor**, it will display ***Call Chosen Vendor*** page. You can update your own data in the below box.

Date	This field ask you to choose date.

-After finished, click **Save** button or you can click **Previous Screen** button if there is no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

-Then select **Send Item/ Installation**, it will display ***Send Item /Installation*** page. You can update your own data in the below box.

Receiver Name	This field ask you to input receiver name.
Invoice No	This field ask you to input Invoice No.
DO No	This field ask you to input Do No.

-After finished, click **Save** button or you can click **Previous Screen** button if there is no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

11 Then select **Produce Tag,** it will display ***Produce Tag*** page. You can update your own data in the below box.

Date	This field ask you to choose date.

-Click on **pencil button** to edit serial No and Tag No. When you click on **pencil button**, it will display ***Edit Serial No*** page. You can update your own data in the below box.

Tag No	Display Tag No.

-After finished, click **Save** button or you can click **Previous Screen** button if there is no data to be change. When you click **Save** button, it will display ***Produce Tag*** page.

-Click on **Trash Bin** to delete Serial No. When you click on **Trash Bin**, it will ask you "Delete this Specification?". Click **Ok** if you want to delete Specification or click **Cancel** if not.

-Please tick (✔) at the checkbox to allow user view status and print separate page or not.

-After finished, click **Save** button or you can click **Previous Screen** button if there is no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

12 Then select **Payment Verification,** it will display ***Payment Verification*** page. You can update your own data in the below box.

Date	This field ask you to choose date.

-After finished, click **Save** button or you can click **Previous Screen** button if there is no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

B) **Acceptance System:-**

* Please tick (✔) at the checkbox to allow user view the status.

1 First, please select **Define Specification**, it will display ***Define Specification*** page. You can update your own data in the below box.

Expertise	List of the expertise and you must select one.
Date	Date that you input the data.
Item Code	Display item code.
User spec	Display User specification.
Quantity	Display quantity of the item.
Spec	Display ✓ or ?.

-Click on **pencil button** to edit specification. When you click on **pencil button**, it will display ***Detail Specification*** page. You can update your own data in the below box.

Specification Name	This field ask you to input the specification.
Specification Description	This field ask you to input the specification description.

-After finished input your specification, click **Add Specification** button. When you click **Add Specification** button, it will display:-

Specification Name	This field ask you to input the specification.
Specification Description	This field ask you to input the specification description.

-Click on **pencil button** to edit specification. When you click on **pencil button**, it will display ***Edit Detail Specification*** page. You can update your own data in the below box.

Specification Name	This field ask you to input the specification.
Specification Description	This field ask you to input the specification description.

-Click on **Trash Bin** to delete specification. When you click on **Trash Bin**, it will ask you "Are you want to delete this Specification?". Click **Ok** if you want to delete Specification or click **Cancel** if not.

-After edit Specification, click **Edit Specification** button. It will display ***Detail Specification*** page.

-Click on **Trash Bin** to delete specification. . When you click on **Trash Bin**, it will ask you "Are you want to delete this Specification?". Click **Ok** if you want to delete Specification or click **Cancel** if not.

-After finished edit specification, click **Save** button. When click **Save** button, it will display ***Define Specification*** page.

-After finished update specification, click **Save** button or you can click **Previous Screen** button if no data want to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

2 Then select **Search Quotation**, it will display ***Search Quotation*** page. You can update your own data in the below box.

Date	This field ask you to choose date.
Vendor Name	This field ask you to choose vendor.
Price	This field ask you to input your price.

-After choose Vendor and Price, click **Add Vendor**. Do this again to add more vendor.

-Please tick (✔) at the checkbox to delete one vendor or click **Select All** if you want to delete all vendor. Click **Remove All** to remove.

-After finished, click **Save** button or you can click **Previous Screen** button if no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

3 Then select **Acceptance Document**, it will display ***Acceptance Document*** page.You can update your own data in the below box.

Reference No	This field ask you to input the reference no.
Vendor Name	This field ask you to choose vendor with their price.
Date	This field ask you to choose the date.

-Click **Print** button to print document.

-After finished, click **Save** button or you can click **Previous Screen** button if no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

4 Then select **Produce PO**, it will display ***Produce PO*** page. You can update your own data in the below box.

No PO	This field ask you to input No PO.
Vendor Name	Display the vendor name.
Branch Name	This field ask you to choose branch of that vendor.
Send to	Display department
Officer name	Display the officer name
Name of Vote	Display the name of vote.
Date	This field ask you to choose date.
Note	This field ask you to input note.
Price/Unit	This field ask you to input price for 1 item.
Total Price	Display the total of price.

-Click **Print PO** button to print PO .

-After finished, click **Save** button or you can click **Previous Screen** button if there is no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

5 Then select **Send Item**, it will display ***Send Item*** page. You can update your own data in the below box.

Receiver Name	This field ask you to input receiver name.
Invoice No	This field ask you to input Invoice No.
Invoice Date	This field ask you to input Invoice date.
DO No	This field ask you to input Do No.
DO Date	This field ask you to input DO Date .

-After finished, click **Save** button or you can click **Previous Screen** button if there is no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

6 Please select **Document Tender,** it will display ***Document Tender*** page. You can update your own data in the below box.

Application Code	List of the expertise and you must select one.
Item Code	Display item code.
Quantity	Display the quantity of item.
Tender No	This field ask you to input tender no.
Date	Input your date.
User spec	Display user specification.

-After finished, click **Save** button or you can click **Previous** button. When you click **Save** button, it will display ***Update Procurement*** *Status.*

7 Then select **Produce Tag,** it will display ***Produce Tag*** page. You can update your own data in the below box.

Date	This field ask you to choose date.

-Click on **pencil button** to edit serial No and Tag No. When you click on **pencil button**, it will display ***Edit Serial No*** page. You can update your own data in the below box.

Tag No	Display Tag No.
Serial No	This field ask you to input the Serial No

-After finished, click **Save** button or you can click **Previous Screen** button if there is no data to be change. When you click **Save** button, it will display ***Produce Tag*** page.

-Click on **Trash Bin** to delete Serial No . When you click on **Trash Bin**, it will ask you "Are you want to delete this Specification?". Click **Ok** if you want to delete Specification or click **Cancel** if not.

-Please tick (✓) at the checkbox to allow user view status and print separate page or not.

-After finished, click **Save** button or you can click **Previous Screen** button if there is no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

8 Then select **Payment Verification,** it will display ***Payment Verification*** page. You can update your own data in the below box.

Date	This field ask you to choose date.

-After finished, click **Save** button or you can click **Previous Screen** button if there is no data to be change. When you click **Save** button, it will display ***Update Procurement Status*** page.

4.2.2.2 Application - Application Status -Update Status

-The page will ask you to input your ***application code***. After you click submit button, the page will display ***Update Procurement Status*** page You can update your own data in the below text box.

-Click on **View All Status**. When you click **View All Status**, it will display ***Status List*** page.

-Click on any **Status at Status Name** to view that status.

-Click on **pencil button** to edit the latest status, the step is same with **Add Status** or click on **Trash Bin** to delete the latest status.

-Click **Previous Screen** button to go to ***Update Procurement Status*** page.

4.2.3 Application -Application List

4.2.3.1 Application - Application List - By Department

-The page will ask you to choose department. After you choose, click **Submit** button, the page will display ***Application list by department*** page. Click **Cancel** button to reset.

-You can click **view** to view the status of the application that you have been choose.

-When you click **view**, it will display ***Update Procurement Status*** page.

-Click **View All Status** to view all status list for that application code. You can click to the one of the **Status Name** to view that status. Then you click on **Previous Screen** button to go the *Status List* page.

-Click on **image "scan files"** to edit the latest status, the step is same with **Add Status** or click on **Trash Bin** to delete the latest status.

-Click on **Previous Screen** button, it will display ***Update Procurement Status*** page.

4.2.3.2 Application - Application List - By Status

-The page will ask you to choose Status. After you choose, click **Send** button, the page will display ***Application list by status*** page. Click **Cancel** button to reset.

-You can click **view** to view the status of the application that you have been choose. - When you click **view**, it will display ***Update Procurement Status*** page.

-Click **View All Status** to view all status list for that application code. You can click to the one of the **Status Name** to view that status. Then you click on **Previous Screen** button to go the *Status List* page.

-Click on **image "scan files"** to edit the latest status, the step is same with **Add Status** or click on **Trash Bin** to delete the latest status.

-Click on **Previous Screen** button, it will display ***Update Procurement Status*** page.

4.4 General

-When click **General** button, you will see the drop down menu.

-Link at drop down menu under **General**:

Link	Sub Link	Function
Admin	**Update**	Link to ***Update Admin*** page.
	Change Password	Link to ***Change Password*** page.
Applicant	**New**	Link to ***New Department Representative*** page.
	Update	Link to page showing all the *RC name* and *department*.
Department	**New**	Link to ***New Department*** page.
	Update	Link to page to select *department name*.
Item Type	**New**	Link to ***New Item Type*** page.
	Update	Link to page showing all the *item type*.
Expertise	**New**	Link to ***New Expertise*** page.
	Update	Link to page showing all the *expertise*.
Status	**New**	Link to ***New Status*** page.
	Update	Link to page showing all the *status name*.

Purchase Type	**New**	Link to ***New Purchase Type*** page.
	Update	Link to page showing all the *purchase type*.
Item	**New**	Link to ***New Item*** page.
	Update	Link to ***Search Item*** page.

4.4.1 General - Admin

4.4.1.1 General - Admin - Update

Admin can update his/her own data as shown in the below text box.

Name	Display the name of the admin.
Salary No	This field is *disabling*.
Tel No	Display tel no.
Fax No	Display fax no.
Email Address	Display email address.

-After finish update your data, click **Update** button.

-Message "**Admin's data has been updated.**" will be display.

4.4.1.2 General - Admin - Change Password

-You can change your old password to new password.

Admin Name	This field is *disabling*.
New Password	Type your new password.
Confirm New Password	Retype your new password.
Cancel button	To reset all the data.

-After finish change your data, click **Update** button.

Message "**Your password has been updated.**" will be display.

4.4.2 General - Applicant

4.4.2.1 General – Applicant - New

-Admin can create new applicant

This page will ask you to fill the data about new user.

Username	Create username for new user. (This username is used for login client)
Password	Create password for new user.
Confirm New Password	Retype the new password.
Department	Select the department for new user.
Name	Type the name for new user.
Salary No	Type the salary no for new user.
Tel No	Type the tel no for new user.
Fax No	Type the fax no for new user.
Email Address	Type the email address for new user.
Cancel	To reset all the data.

-After finish fill the data about new user, click **Update** button.

Message "**New Applicant created successfully.**" will be display.

All the field must be fill in except **Fax No (optional). **Username** and **Salary No** must be unique (different for each user).

4.4.2.2 General - Applicant - Update

-This page will display all the *RC Name* and *Department*.

-Admin can edit any user by click at the image pencil.

-Now, you are at the ***Update User*** page. You can update the data in the below text box.

Name	Display the name of the user.
Department	Select the department for the existing user.
Salary No	This field is *disabled*.
Tel No	Display phone no.
Fax No	Display fax no.
Email Address	Display email address.
Status	You can check either one of the radio button (*Active or Non-Active*).
Previous Screen button	To see the previous page (page that display all the *RC Name* and *Department*).

-After finish update user's data, click **Update** button.

Message "**User's data has been updated.**" will be display.

If user's status is **Active, this mean the user can login to make application, else the user cannot login to make application

4.4.3 General - Department

4.4.3.1 General - Department - New

-Admin can create new department for RC (***New Department***).

Department Code	This code is auto generate.
Department Name	Type new department name.
Cancel button	To reset all the data.

-After finish create new department, click **Save** button.

Message "**New department's data has been saved.**" will be display.

4.4.3.2 General - Department - Update

-This page will ask you to select the RC which you want to update the department.

-Then click the **Submit** button, all the *Department Name* for that RC (Faculty) will be display.

-Admin can edit the department by click at the image pencil.

-Now, you are at the ***Update Department*** page. You can update the data in the below text box.

Department Code	Display Department's code (disabled).
Department Name	Display the department name.
Previous Screen button	To see the previous page.

-After finish update department, click **Update** button.

Message "**Department's data has been updated.**" will be display.

4.4.4 General - Item Type

4.4.4.1 General - Item Type - New

-Admin can create new item type (***New Item Type***).

Item Type Code	This code is auto generate.
Item Type Name	Create new item type name.
Cancel button	To reset all the data.

-After finish create new item type, click **Save** button.

Message "**New item type's data has been saved.**" will be display.

4.4.4.2 General - Item Type – Update

-This page will display all the *Item Type*.

-Admin can edit the item type by click at the image pencil.

-Now, you are at the ***Update Item Type*** page. You can update the data in the below text box.

Item Type Code	This field is disabling.
Item Type Name	Display the item type name that you want to edit.
Previous Screen button	To see the previous page (page that display all the *Item Type*).

-After finish update item type, click **Update** button.

Message "**Item Type's data has been updated.**" will be display.

4.4.5 General - Expertise

4.4.5.1 General - Expertise - New

-Admin can create new expertise (***New Expertise***).

Expertise Code	This code is auto generate.
Expertise Name	Create new expertise name.
Cancel button	To reset all the data.

-Then select item type from **List Of Item Type** by click the check box.

-You can select all the item type by click **Select All**.

-You can remove all the item type by click **Remove All**.

-After finish creates new expertise, click **Save** button.

Message "**New expertise's data has been saved.**" will be display.

4.4.5.2 General - Expertise - Update

-This page will display all the *Expertise Name*.

-Admin can edit the expertise by click at the image pencil.

-Now, you are at the ***Update Expertise*** page. You can update the data in the below text box.

Expertise Code	This field is disabled.
Expertise Name	Display the expertise name that you want to edit.
Previous Screen button	To see the previous page (page that display all the *Expertise Name*).

-Then you can reselect item type from **List Of Item Type** by click the check box.

-You can select all the item type by click **Select All**.

-You can remove all the item type by click **Remove All**.

-After finish update expertise, click **Update** button.

-Message "**Expertise's data has been updated.**" will be display.

4.4.6 General - Status

4.4.6.1 General - Status - New

-Admin can create new status (***New Status***).

Status Code	This code is auto generate (S01).
Status Name	Create new status name.
Cancel button	To reset all the data.

-After finish creates new status, click **Save** button.

Message "**New status data has been saved.**" will be display.

4.4.6.2 General - Status - Update

-This page will display all the *Status Name*.

-Admin can edit the status name by click at the image pencil .

-Now, you are at the ***Update Status*** page. You can update the data in the below text box.

Status Code	This field is disabling.
Status Name	Display the item type name that you want to edit.
Previous Screen button	To see the previous page (page that display all the *Status Name*).

-After finish update status, click **Update** button.

Message "**Status data has been updated.**" will be display.

4.4.7 General - Item

4.4.7.1 General - Item - New

-Admin can create new item (***New Item***).

Item Code	This code is auto generate.
Item Type	Select the item type for item.
Item Name	Type new item name.
Cancel button	To reset all the data.

-After finish create new item, click **Save** button.

Message "**New item's data has been saved.**" will be display.

4.4.7.2 General - Item - Update

-You are at the ***Search Item*** page.

-This page will ask you to select the item type, which you want to update the item name.

-Then click the **Search** button, all the *Item Code* and *Item Name* for that item type will be display.

-Admin can edit the item name by click at the image pencil.

-Now, you are at the ***Update Item*** page. You can update the data in the below text box.

Item Code	This field is disabledg.
Item Type	You can reselect the item type.
Item Name	Display the item name that you want to edit.
Status	You can check either one of the radio button (*Active or Non-Active*).
Cancel button	To reset all the data.

-After finish update item name, click **Update** button.

Message "**Item's data has been updated.**" will be display.

If item's status is **Active, then user (client) can purchase the item, else client cannot purchase the item.

4.5 Logout

-Click **Logout** button to logout account each time before exit e-Procurement for security purpose

5.0 Vendor

5.1 Login

-Make sure u have your user name and password for e-Procurement.

-To login e-Procurement please visit : http://202.185.112.49/POT/main/main3.php

-Click *Vendor* to enter login page.

Button on *login* page: -

Button	Function
Submit	Enter e-Procurement account.
Cancel	Erase all entered data on login page.

-Please type into following text field: -

Username	Type in username of your e-Procurement account.
Password	Type in your password.

-Press **Submit** button to proceed to e-Procurement.

or

-Press **Cancel** button to erase all the data you had typed.

-If login success you will come to *Contact Us* page.

Repeat buttons on left hand side of each page :-

Button	Function
Update Branch	Link to ***Update Branch*** page.
Application Item	Link to ***Item of Application*** page.
Change Password	Link to ***Change Password*** page.
Produce Serial Number	Link to ***Produce Serial Number*** page.
Logout	Logout account and exit.

5.2 Update Branch

-Addition button/link: -

Button/link	Function
Submit	Click after select an application code from the drop down list. List of related item with be display.
Edit	Click to edit selected branch information.

-Now you will come to a page which require you to select as branch name from the drop down list.

-After select, click *Submit* button to display the related item to the application.

5.2.1 Edit branch information

-Click *Edit* to edit selected branch information which you want to update/change.

Addition button: -

Button	Function
Save	Click to save changes.
Previous Screen	Click to back to the previous page.

5.3 Application Item

-Addition button/link :-

Button/link	Function
Submit	Click after select an application code from the drop down list. List of related item with be display.
View	Link to page, which will display ***Item Specification***.

-Now you will come to a page which require you to select an application code from the

drop down list.

-After select, click *Submit* button to display the related item to the application code.

5.3.1 Access item specification

-Click *View* to link to the page which contains specification of the selected item.

Addition button: -

Button	Function
Print	Click to print ***Item Specification*** and related information.
Previous Screen	Click to back to the previous page.

5.4 Change Password

-Addition button :-

Button	**Function**
Save	Click to save new password.
Cancel	Click to erase all data entered.

-Please type into following text field: -

New Password	New password
Confirm Password	Re-enter new password and must be same as **New Password**.

-Press **Save** button to save changes.

-Press **Cancel** button to erase all the data you had typed.

-If you successfully change your password you will then come to page, which tells you that you new password has been saved.

5.5 Produce Serial Number

Addition button :-

Button	Function
Save	Click to save/update serial number.
Cancel	Click to erase all entered serial number.
Print	Click to print serial number and related information.
Previous Screen	Click to go to the previous page.

-Now you will come to a page which require you to select an application code from the drop down list.

-After select, click *Submit* button to display the related information to the application code.

5.5.1 Update/Add serial number

-Click button *Save* to save serial number(s).

5.5.2 Print serial number

-Click button *Print* to print serial number and related information.

-Click button *Previous Screen* to get back to the previous page.

5.6 Logout

-Click *Logout* button to logout account each time before exit e-Procurement for security purpose.

APPENDIX B

SCREEN SHOTS

Figure B (1): e-Procurement for IT Stuff Main Page

Figure B (2): Form Validation

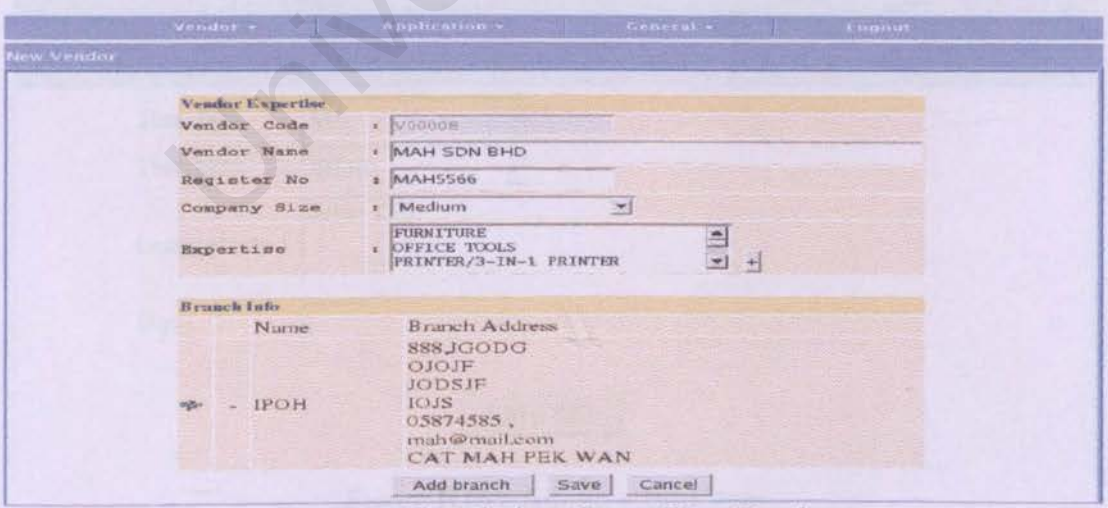


Figure B (3): Admin – Create New Vendor

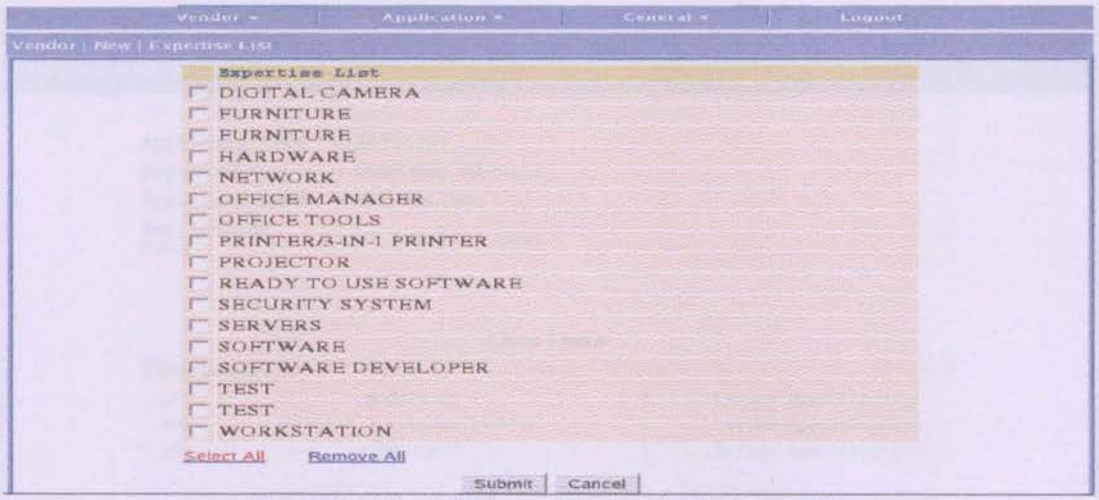


Figure B (4): Admin – Select Admin Expertise

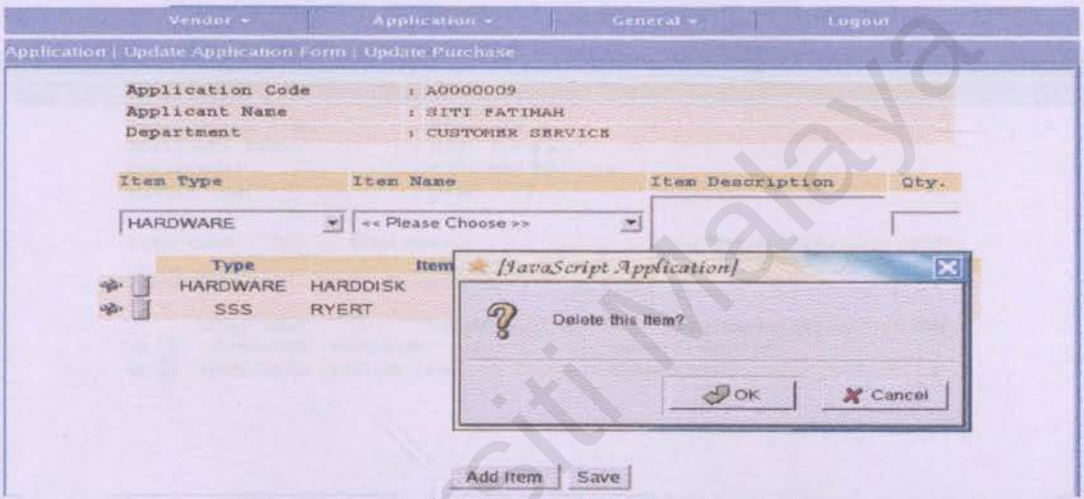


Figure B (5): Admin – Update PO Application

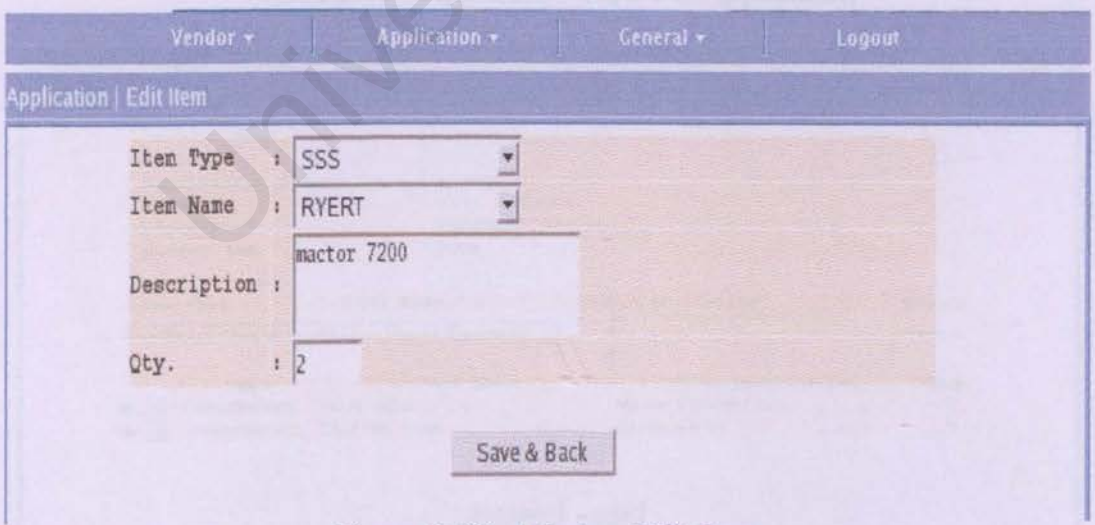


Figure B (6): Admin – Edit Item

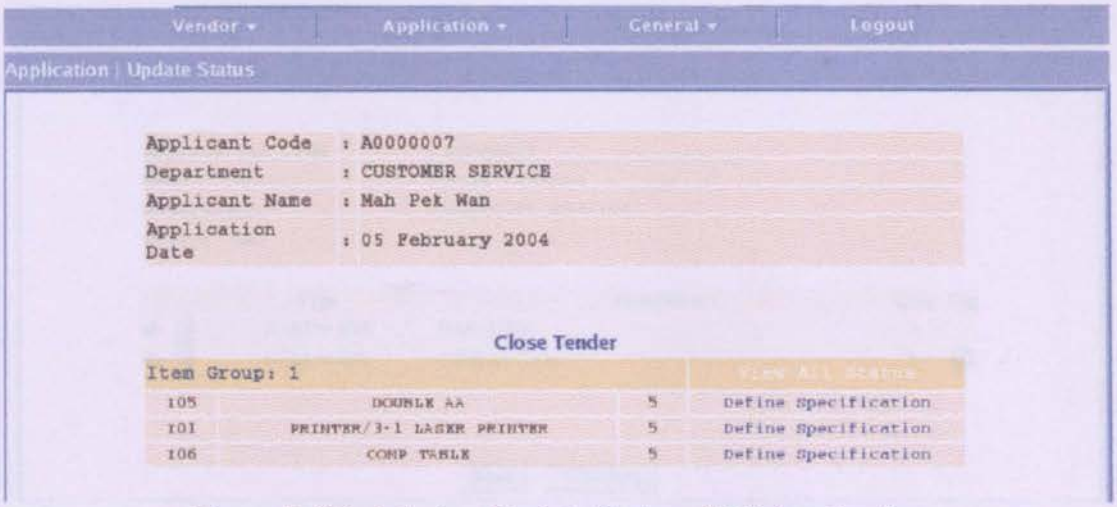


Figure B (7): Admin – Update Status of PO Application

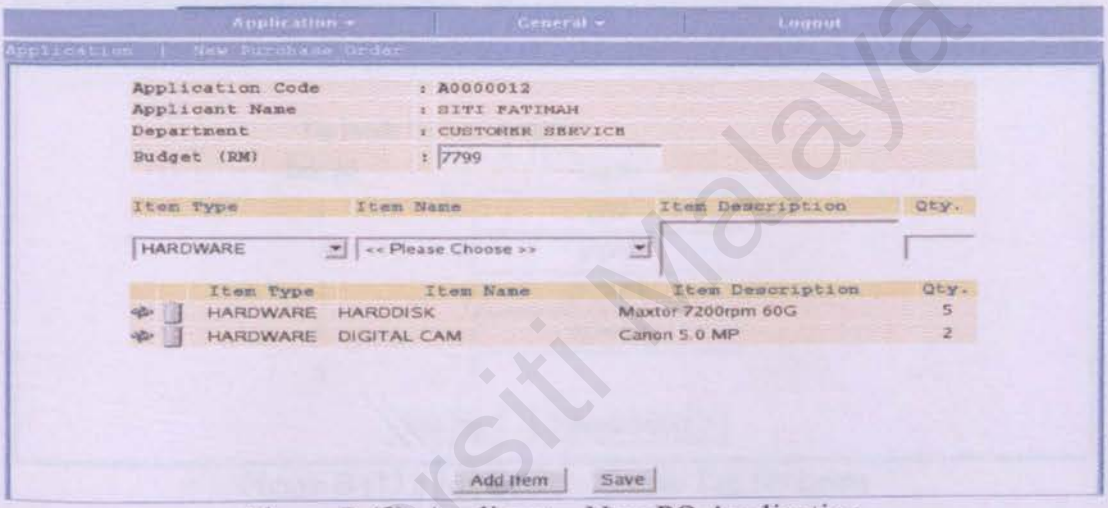


Figure B (8): Applicant – New PO Application

Figure B (9): Applicant – Update PO Application

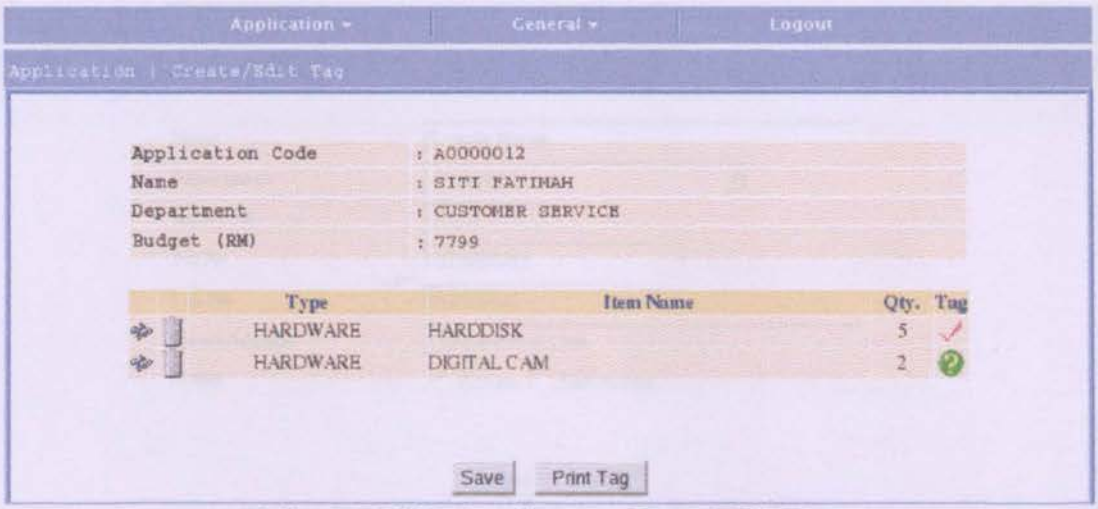


Figure B (10): Applicant – Create/Edit Tag

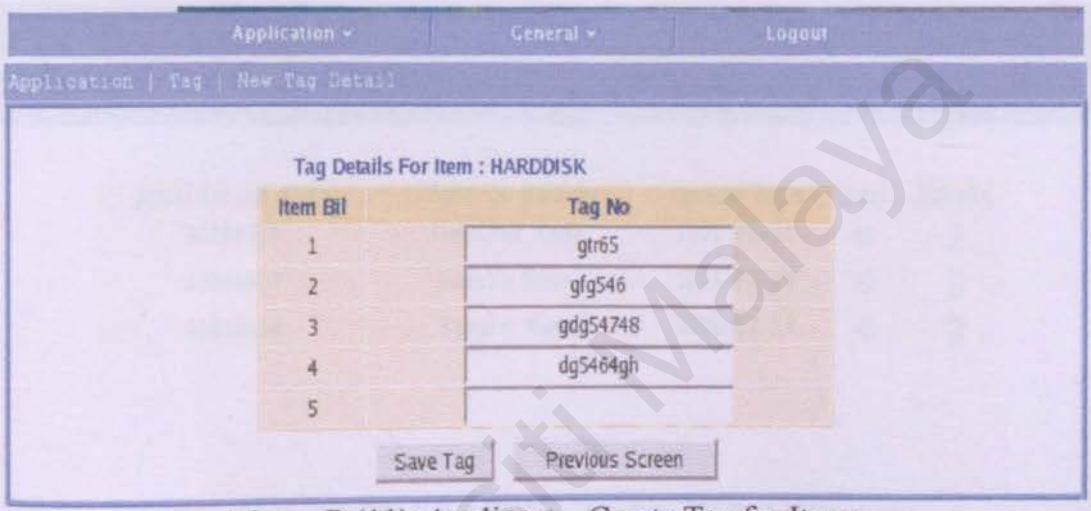


Figure B (11): Applicant – Create Tag for Items

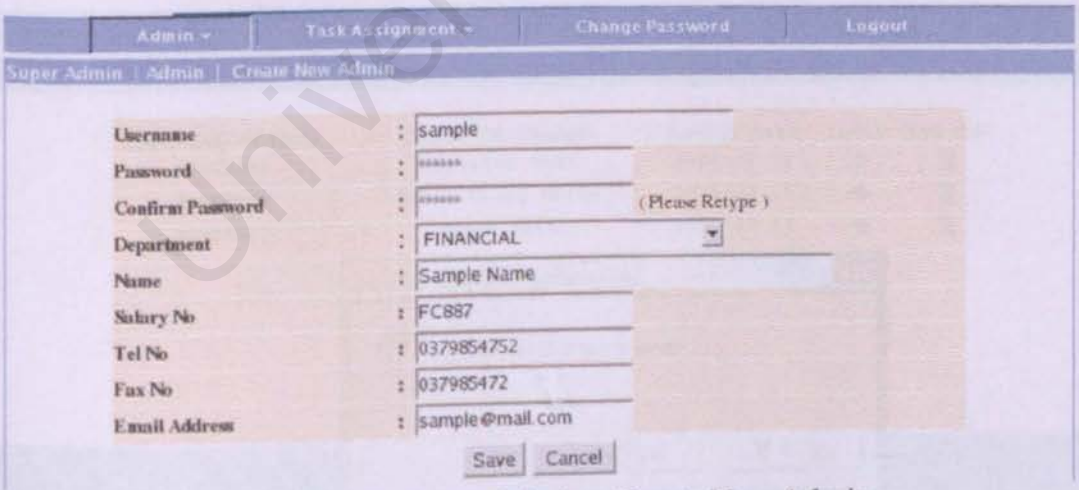


Figure B (12): Super Admin – Create New Admin

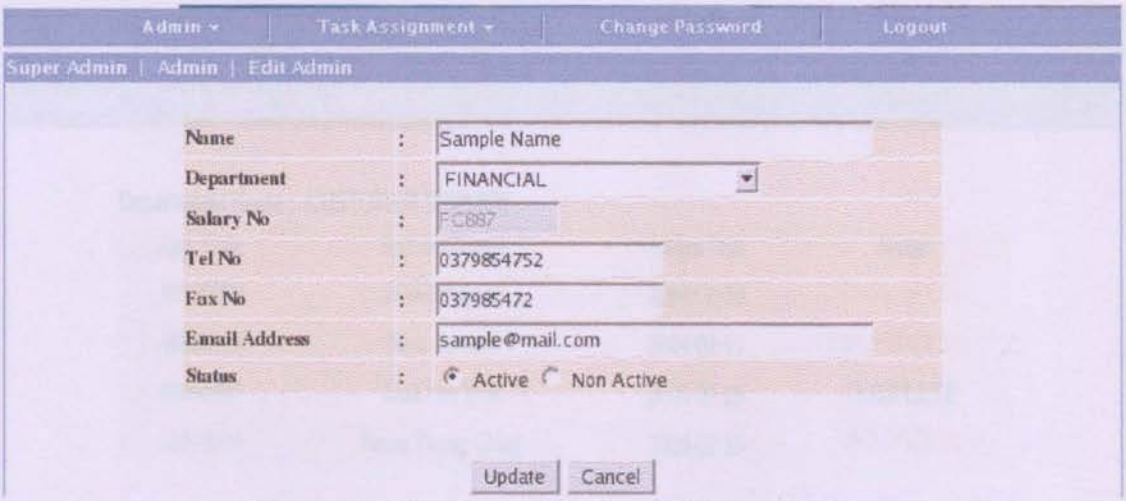


Figure B (13): Super Admin – Edit Admin data

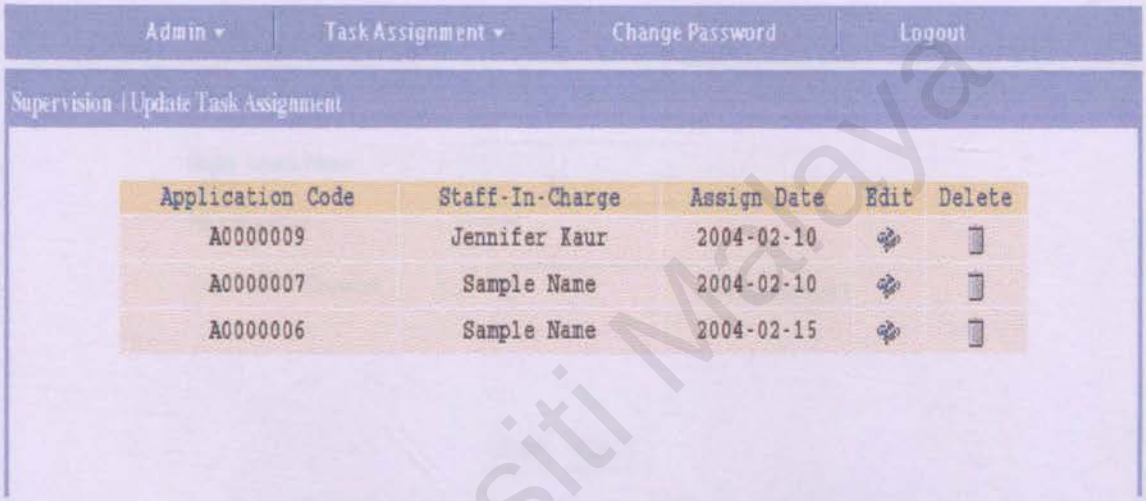


Figure B (14): Super Admin – Update Task Assignment

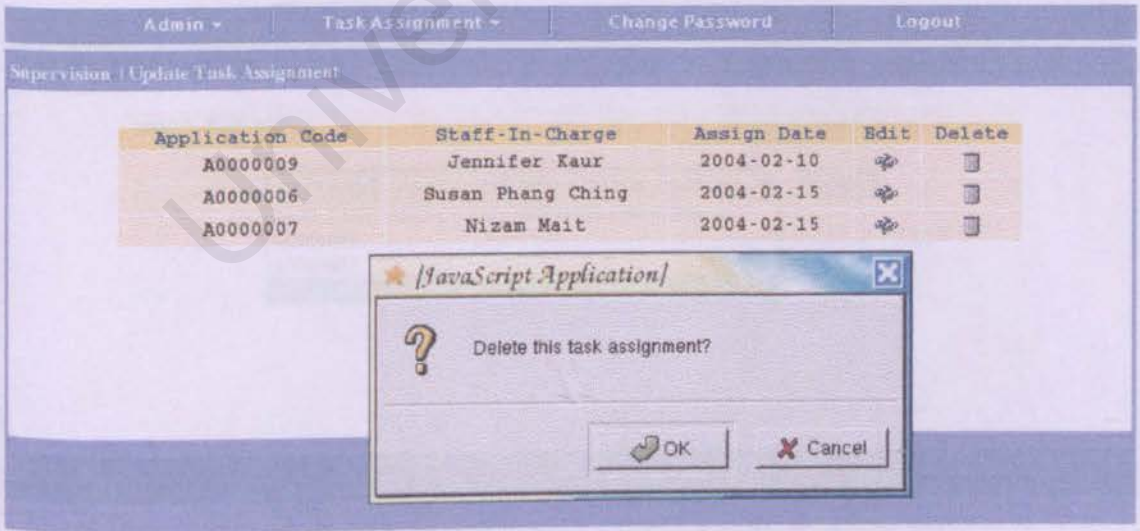


Figure B (15): Super Admin – Delete Task Assignment

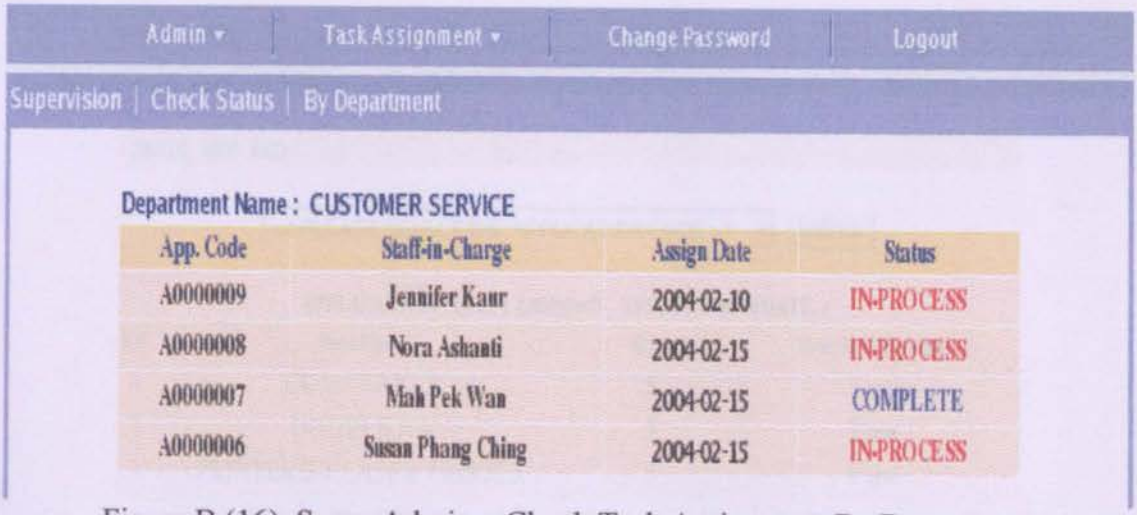


Figure B (16): Super Admin – Check Task Assignment By Department

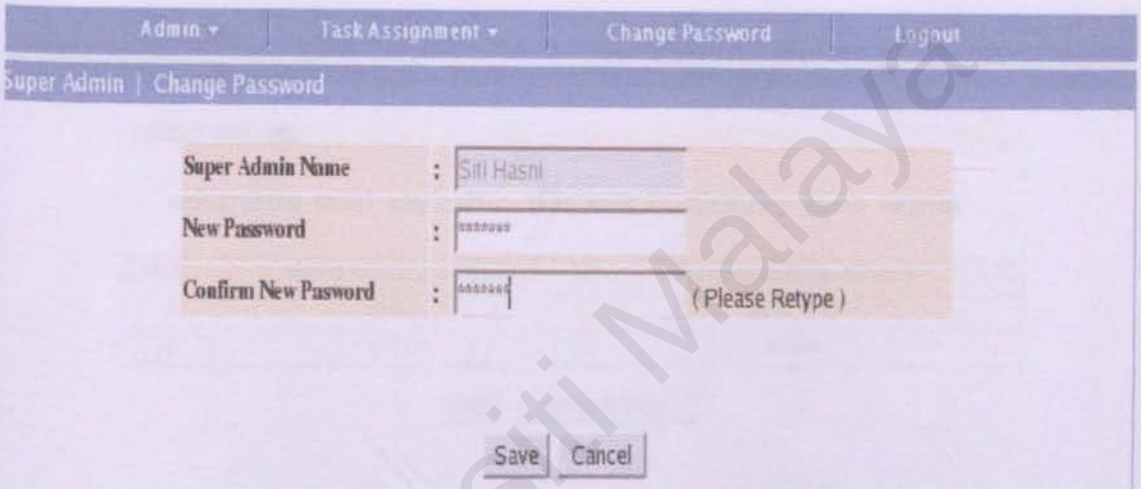


Figure B (17): Super Admin – Change Password

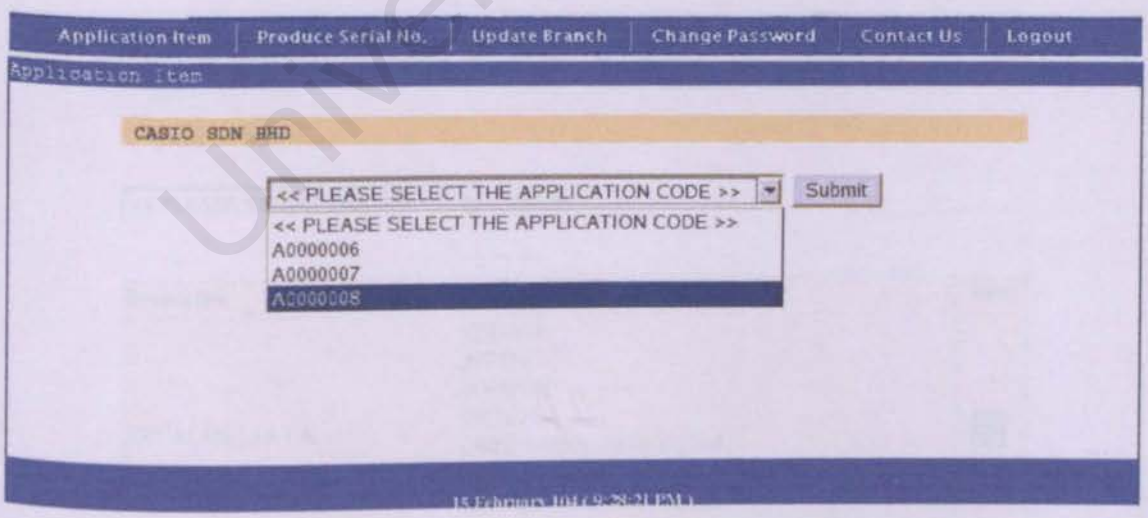


Figure B (18): Vendor – Select Application Code for Tender

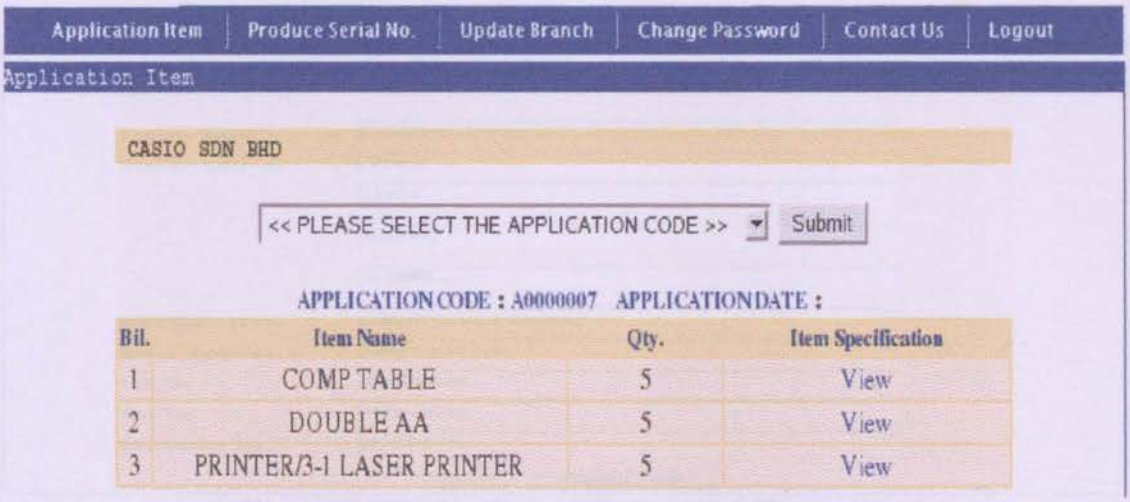


Figure B (19): Vendor – View Application Details

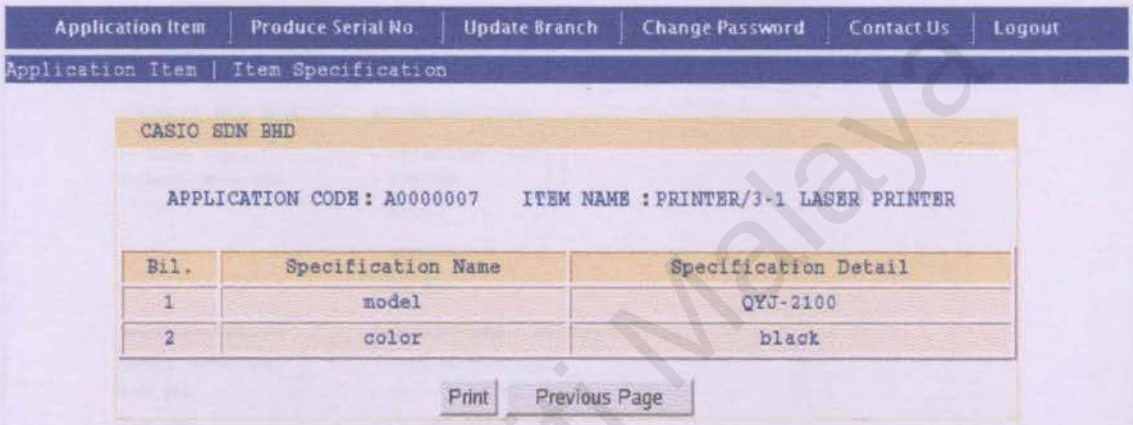


Figure B (20): Vendor – View Application Item Specification

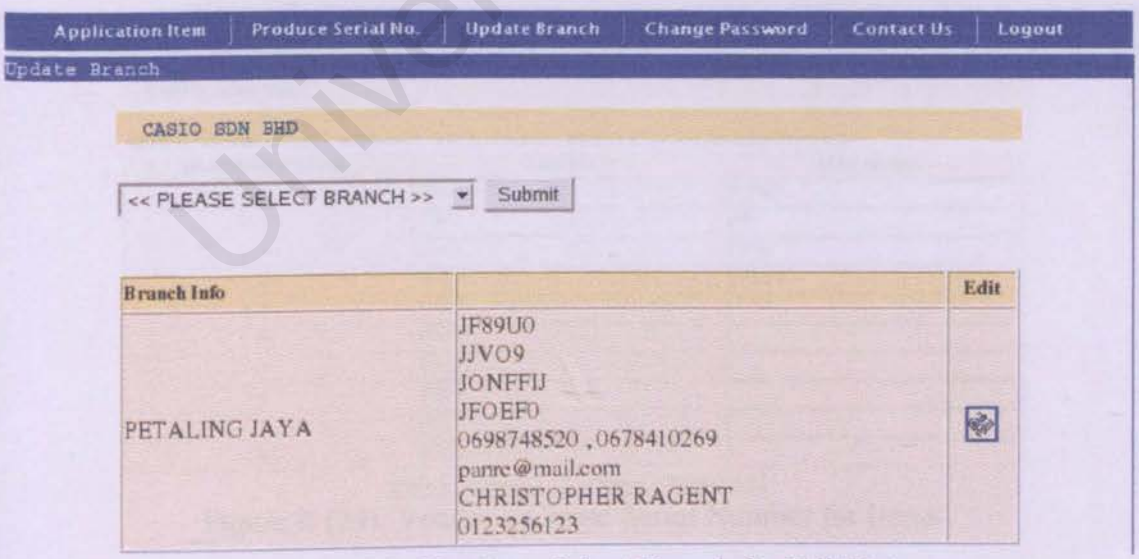


Figure B (21): Vendor – Select Branch for Edit Data

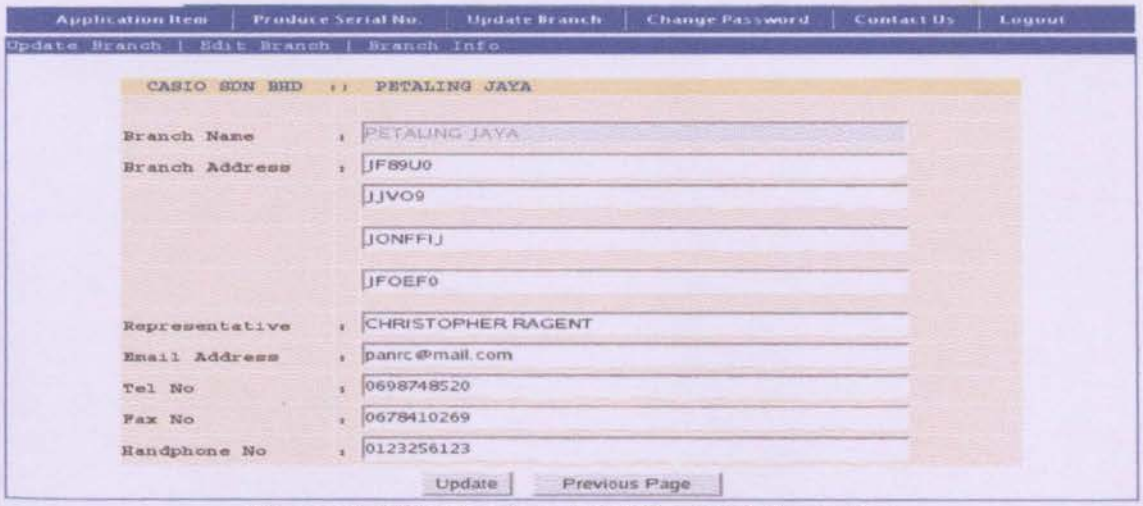


Figure B (22): Vendor – Edit Branch Information

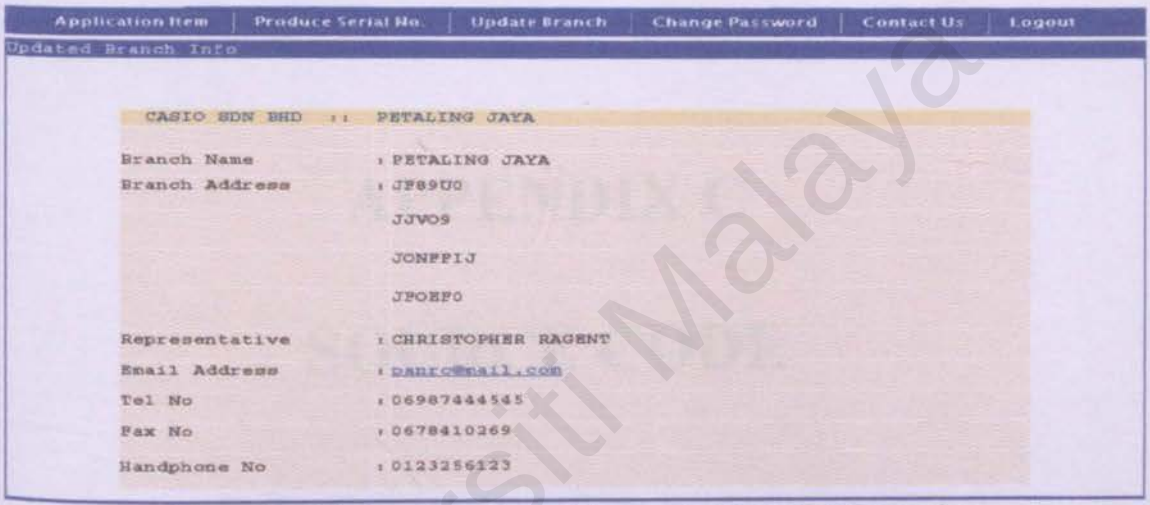


Figure B (23): Vendor – Save Changes to Branch Information

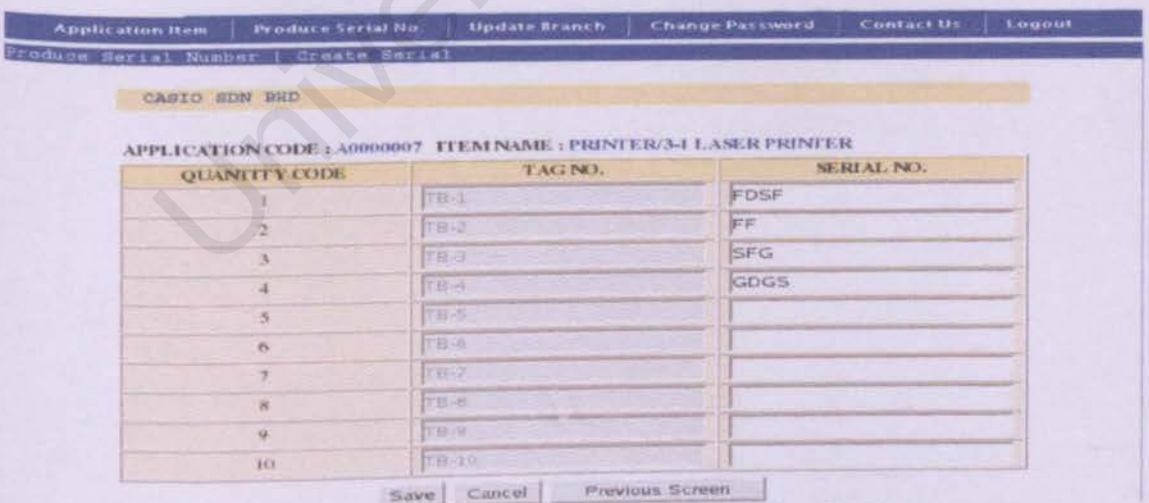


Figure B (24): Vendor – Create Serial Number for Items

APPENDIX C

SOURCE CODE

A: Update Application

```
<?php
session_start();

include("all_admin.php");

$conn = pg_open();

if (!$conn)
{
  print("Database connection failed.");
  exit;
}


$q1 = "Select  *, TO_CHAR(app_date,'DD Month YYYY') as date  from application where
app_code='$txtapp' ";
$r1=pg_send($q1);
$count1 = pg_numrows($r1);


if($count1 == 1)
{
        $values1=pg_fetch_array($r1,0,PGSQL_ASSOC);
        $KodJabatan = $values1['app_deptcode']."";
        $KodGaji = $values1['app_salaryno']."";
        $AppDate = $values1['date']."";

        $q3 = "Select * from applicant where appt_salaryno='$salaryc' ";
        $r3=pg_send($q3);
        $count3 = pg_numrows($r3);


        if ($count3==0)
        {

                $q4 = "Select * from administrator where adm_salaryno='$ salaryc' ";
                $r4=pg_send($q4);
                $values4=pg_fetch_array($r4,0,PGSQL_ASSOC);
                $AppName = $values4['adm_name']."";

        }
        else
        {
        $values3=pg_fetch_array($r3,0,PGSQL_ASSOC);
        $AppName = $values3['appt_name']."";
        }



        $q14 = "Select * from department where dept_code='$deptc' ";
        $r14=pg_send($q14);
        $values14=pg_fetch_array($r14,0,PGSQL_ASSOC);
        $Txtdept= $values14['dept_name']."";
}

$q2 = "Select * from application_item where apem_appcode='$txtapp' ";
$r2=pg_send($q2);
```

```
$count2 = pg_numrows($r2);
?>

<html>
<head>
<title>E-Procurement for IT Stuff :: Admin Module</title>

<style>
<!--
a {text-decoration:none;
  cursor:hand; }

-->
</style>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script language="JavaScript">
<!--

function send()
{
document.frmApp.action = "/PHP_WXES3182/app1/upform_main.php?status=1"
document.frmApp.method = "post"
document.frmApp.submit();
}


function valid_send()
{
var ErrMssg = ""
var ErrSet = "False"

count = parseInt(document.frmApp.gtotal.value);

for (var row=0; row<count; row++)
{
        if (document.frmApp.buytype[row].options[0].selected == true)

        {
         ErrSet = "True"
         ErrMssg +="* Please select purchase type. \n"
         break;
        }

}


if (ErrSet == "True")
 {
 alert(ErrMssg)
 return "false";
 }
else
 {
        return "true";

 }
}

function f(o)
```

```
{
 o.value=o.value.toUpperCase();
}
//-->
</script>
</head>

<body bgcolor="#FFFFFF" text="#000000"  vlink=white alink=white link=white
background="/PHP_WXES3182/Images/HOMES_Banner_Pale_Small_B&W.JPG">
<table width="750" border="0" bgcolor="#0f87ff" align="center">
 <tr>
  <td height="15">
   <div align="left"><font size="2" face="Arial, Helvetica, sans-serif" color ="#ffffff" ><b>
    Application | Update Status </b></font></div>
  </td>

 </tr>
</table>
<table width="750" border="1" bordercolor="#0f87ff" bgcolor="#0f87ff" align="center">
 <tr>

  <td valign="top" height="223" bgcolor="#FFFFFF">
   <form name=frmApp>
    <table width="600" border="0" align="center" >
     <tr>
      <td >
       <? if ($count1 == 1) { ?>

       <br>
       <table width="557" border="0" align="center">
        <tr bgcolor="#ffddbb">
         <td width="129"><b><font size="2" face="Courier New, Courier, mono">Applicant
          Code </font></b></td>
         <td width="10"><b><font face="Courier New, Courier, mono"
size="2">:</font></b></td>
         <td width="427"> <b><font face="Courier New, Courier, mono" size="2">
          <input type=hidden name=txtapp value="<? echo "$txtapp" ?>">
          <input type=hidden name=g value="<? echo "$g" ?>">
          <? echo "$txtapp" ?>
          </font></b></td>
        </tr>
        <tr bgcolor="#ffddbb">
         <td width="129"><b><font size="2" face="Courier New, Courier,
mono">Department</font></b></td>
         <td width="10"><b><font face="Courier New, Courier, mono"
size="2">:</font></b></td>
         <td width="427"> <b><font face="Courier New, Courier, mono" size="2">
          <? echo "$Txtdept" ?>
          </font></b></td>
        </tr>
        <tr bgcolor="#ffddbb">
         <td width="129"><b><font size="2" face="Courier New, Courier, mono">Applicant
          Name</font></b></td>
         <td width="10"><b><font face="Courier New, Courier, mono"
size="2">:</font></b></td>
         <td width="427"> <b><font face="Courier New, Courier, mono" size="2">
          <? echo "$AppName" ?>
          </font></b></td>
        </tr>
        <tr bgcolor="#ffddbb">
```

```
          <td width="129"><b><font size="2" face="Courier New, Courier, mono">Application
            Date </font></b></td>
          <td width="10"><b><font face="Courier New, Courier, mono"
size="2">:</font></b></td>
          <td width="427"> <b><font face="Courier New, Courier, mono" size="2">
            <? echo $AppDate ?>
            </font></b></td>
        </tr>
      </table>
      <br>
      <?
                $q17 = "Select * from application_item where apem_appcode='$txtapp' and
apem_buytype = 'D01'";
                $hasil17=pg_send($q17);
                $count17 = pg_numrows($hasil17);
                $q18 = "Select * from application_item where apem_appcode='$txtapp' and
apem_buytype = 'D02'";
                $hasil18=pg_send($q18);
                $count18 = pg_numrows($hasil18);

                              if($count17 > 0)
                              {
                               ?>
      <div align="center"><font color="#0000CC"><br>
        <b><font face="Arial, Helvetica, sans-serif" size="2">Close Tender
        </font></b> </font></div>
      <table width=557 align=center>
        <?


                              $q19 = "Select distinct apem_group from application_item where
apem_buytype = 'D01' and apem_appcode='$txtapp' order by apem_group";
                              $rB=pg_send($q19);
                              $count19 = pg_numrows($rB);


                               for ($row1=0; $row1<pg_numrows($rB); $row1++)

                              {
                               $values=pg_fetch_array($rB,$row1,PGSQL_ASSOC);
                                    $kumpulan=$values['apem_group']."";

                               echo "<tr BGCOLOR=\"#ffcc66\" height =20 >";
                          echo"<td width=400  colspan=3><font color=#0000cc size=\"2\"
face=\"courier new,courier,mono\"><b>Item Group: $kumpulan</b></td>";


                                                  echo "<td width=200 align=center><font
size=\"2\" face=\"courier new,courier,mono\" color=#0000cc ><a
href=\"slist_main.php?KodP=$txtapp&jb=D01&kumpulan=$kumpulan\" ><b
ONMOUSEOVER=\"style.textDecoration='underline'\"
ONMOUSEOUT=\"style.textDecoration='none'\">View All Status</b></a></td></tr>";

                                    $q20 = "Select apem_appcode, apem_itemcode,
apem_quant, apem_buytype, apem_group, apem_statuscode, sta_staname, sta_code,item_name from
application_item,status,item where  item_kod = apem_itemnamecode  and  apem_appcode='$txtapp'
and apem_buytype = 'D01' and apem_group='$kumpulan' and sta_code=apem_statuscode";
                                    $hasil20=pg_send($q20);
                                    $count20 = pg_numrows($hasil20);
```

```
                                        for ($row20=0; $row20<pg_numrows($hasil20);
$row20++)
                                                {
$values=pg_fetch_array($hasil20,$row20,PGSQL_ASSOC);
                                                $koditem=$values['apem_itemcode']."";
                                                $kuantiti=$values['apem_quant']."";
                                                $namaitem=$values['item_name']."";
                                                $namastatus=$values['sta_staname']."";


                                                echo "<tr BGCOLOR=\"#ffddbb\"
height =15>";
                                                echo"<td width=50><center><input
type=hidden name=KItem1[] value=$koditem><font size=\"1\" face=\"courier
new,courier,mono\">$koditem</center></td>";
                                                echo "<td width=300><center><font
size=\"1\" face=\"courier new,courier,mono\">$namaitem</center></td>";
                                                echo "<td width=50><center><font
size=\"1\" face=\"courier new,courier,mono\">$kuantiti</center></td>";
                                                echo "<td width=200><center><font
size=\"1\" face=\"courier new,courier,mono\" color=\"#0000cc\">$namastatus</center></td></tr>";

                                                } // end for
                                        } // end for
                                        ?>
        </table>
        <br>
        <?
                                        } // end tender

                                        if ($count18 > 0)
                                        {
                                        ?>
        <div align="center"> <br>
          <b><font color="#0000CC" size="2" face="Arial, Helvetica, sans-serif">Open
Tender</font></b> </div>
        <table width=557 align=center>
         <?


                                        $q19 = "Select distinct apem_group from application_item where
apem_appcode='$txtapp' and apem_buytype = 'D02' order by apem_group";
                                        $rB=pg_send($q19);
                                        $count191 = pg_numrows($rB);

                                         for ($row1=0; $row1<pg_numrows($rB); $row1++)
                                         {
                                          $values=pg_fetch_array($rB,$row1,PGSQL_ASSOC);
                                                $kumpulan11=$values['apem_group']."";

                                                        echo "<tr BGCOLOR=\"#ffcc66\" height =20 >";
                                                echo"<td width=400  colspan=3><font color=#0000cc
size=\"2\"
face=\"courier new,courier,mono\"><b>Item Group : $kumpulan11</b></td>";
                                                echo "<td width=200 align=center><font
size=\"2\" face=\"courier new,courier,mono\" color=#0000cc ><a
href=\"slist_main.php?KodP=$txtapp&jb=D02&kumpulan=$kumpulan11\" ><b
ONMOUSEOVER=\"style.textDecoration='underline'\"
ONMOUSEOUT=\"style.textDecoration='none'\">View All Status</b></a></font></td></tr>";
```

```
                                        $q20 = "Select apem_appcode, apem_itemcode,
item_name, apem_quant,
apem_buytype, apem_group, apem_statuscode, sta_staname, sta_code from application_item ,
status,item where
apem_itemnamecode=item_kod  and apem_appcode='$txtapp' and apem_buytype = 'D02' and
apem_group='$kumpulan11'
and sta_code=apem_statuscode";
                                        $hasil20=pg_send($q20);
                                        $count20 = pg_numrows($hasil20);

                                         for ($row20=0; $row20<pg_numrows($hasil20);
$row20++)
                                                {

$values=pg_fetch_array($hasil20,$row20,PGSQL_ASSOC);
                                                $koditem=$values['apem_itemcode']."";
                                                $kuantiti=$values['apem_quant']."";
                                                $namaitem=$values['item_name']."";
                                                $namastatus=$values['sta_staname']."";


                                                echo "<tr BGCOLOR=\"#ffddbb\"
height =15>";
                                                echo"<td width=50><center><input
type=hidden name=KItem1[] value=$koditem><font size=\"1\" face=\"courier
new,courier,mono\">$koditem</center></td>";
                                                echo "<td width=300><center><font
size=\"1\" face=\"courier new,courier,mono\">$namaitem</center></td>";
                                                echo "<td width=50><center><font
size=\"1\" face=\"courier new,courier,mono\">$kuantiti</center></td>";
                                                echo "<td width=200><center><font
size=\"1\" face=\"courier new,courier,mono\" color=#0000CC>$namastatus</center></td></tr>";
                                                } // for
                                } // end for
                                ?>
        </table>
        <br>
        <?
                                }


                        } // end of checking ada data

                else
                {
                 echo "<div align=\"center\"><font color=\"red\" size=\"2\" face=\"Arial, Helvetica,
sans-serif\"><b><br>No Data</b></font></div>";
                }

                ?>
      </td>
     </tr>
    </table>
   </form>
  </td>
 </tr>
<tr>
  <td height="79" bgcolor="#0f87ff" >
```

```
<p>
<!-- Date and time -->
<script language="JavaScript">
        <!--
                function showClock()
                        {

                        if (!document.all&&!document.getElementById)
                                return
                                        thelement=document.getElementById?
document.getElementById("tick2"): document.all.tick2
                        var Digital=new Date()
                        var hours=Digital.getHours()
                        var minutes=Digital.getMinutes()
                        var seconds=Digital.getSeconds()
                        var year = Digital.getYear();
                        var mon = Digital.getMonth();
                        var day = Digital.getDate();
                        var gmon= ""

                        if (mon==0)
                          gmon += "January"
                        else if (mon==1)
                                gmon += "February"
                        else if (mon==2)
                                gmon += "March"
                        else if (mon==3)
                                gmon += "April"
                        if (mon==4)
                                gmon += "May"
                        if (mon==5)
                                gmon += "Jun"
                        if (mon==6)
                                gmon += "Julai"
                        if (mon==7)
                                gmon += "Ogos"
                        if (mon==8)
                                gmon += "September"
                        if (mon==9)
                                gmon += "Oktober"
                        if (mon==10)
                                gmon += "November"
                        if (mon==11)
                                gmon += "December"

                        var dn="PM"

                        if (hours<12)
                                dn="AM"
                        if (hours>12)
                                hours=hours-12
                        if (hours==0)
                                hours=12
                        if (minutes<=9)
                                minutes="0"+minutes
                        if (seconds<=9)
                                seconds="0"+seconds

                        var ctime=day + " " + gmon +" " + year + " ( "
+hours+":"+minutes+":"+seconds+" "+dn
```

```
                                        thelement.innerHTML="<b style='font-
size:12;color:#ffffff;face:verdana'>"+ctime+" )</b>"
                                        setTimeout("showClock()",1000)
                                }
                        window.onload=showClock

                        //-->
                </script>
    </p>
   <div align="center"></div>
    <div align="center"><span id=tick2> </span> </div><br>
    </td>
  </tr>
</table>

<p> </p>
</body>
</html>
```

B: Create New Vendor

```
<?php
session_start();
include("all_admin.php");

if ($refresh == "true")
{
echo "<meta http-equiv='refresh' content='0;URL=ven_main.php?refresh=false'>";
}


$conn = pg_open();


if (!$conn)
{
  print("Database connection failed.");
  exit;
}

$query01 = "Select max(substr(ven_id,2)) as code from vendor  ";
$result1=pg_send($query01);


$cek_id  = "select * from expertise order by ex_name";
$result=pg_exec($conn,$cek_id);


 if(!$result || !$result1  )
 {
          "Error:" + pg_errormessage($conn);
          exit();
 }

$values=pg_fetch_array($result1,0,PGSQL_ASSOC);
$code = $values['code']."";


$new = chgcode($code + 1);

$newCode = 'V'.$new;

function chgcode ($a)
{

if ($a < 10 && $a > 0)
{$ans = '0000'.$a;}
else if ($a >=10 && $a<100 )
{$ans = '000'.$a;}
else if ($a>=100  && $a<1000)
{$ans = '00'.$a;}
else if ($a>=1000 && $a<10000)
{$ans = '0'.$a;}
else if ($a>=10000)
{$ans = $a;}
return $ans;
```

```
}
// ***************************************

if ($status == 0)
{
        $q3 = "delete from vendor_expertise_temp where texven_venid = '$newCode'";
                $hasil3=pg_send($q3);

        $q4 = "delete from branch_temp where tbra_venid= '$newCode'";
                $hasil4=pg_send($q4);

        $q5 = "delete from vendor_temp where tven_id = '$newCode'";
                $hasil5=pg_send($q5);
}

if ($status == 1)
{
$q1  = "select * from vendor_expertise_temp where texven_venid = '$newCode'  ";
$hasil1=pg_exec($conn,$q1);
$baris1 = pg_numrows($hasil1);
 if($baris1 == 0)
  {

                for ($row=0; $row<pg_numrows($result); $row++)
                {
                        if ($choice[$row] != "")
                        {
                                        $q2 = "Insert INTO vendor_expertise_temp values
('$newCode','$choice[$row]')";
                                        $hasil2=pg_send($q2);

                        }// if
                } // for

  }
 else
  {
        $q3 = "delete from vendor_expertise_temp where texven_venid = '$newCode'";
                $hasil3=pg_send($q3);
                for ($row=0; $row<pg_numrows($result); $row++)
                {
                        if ($choice[$row] != "")
                        {
                                        $q2 = "Insert INTO vendor_expertise_temp values
('$newCode','$choice[$row]')";
                                        $hasil2=pg_send($q2);

                        }// if
                } // for

  }

} // status =1

if ($status == 2)  // new_bra.php
{
$q4  = "select * from branch_temp where tbra_venid = '$newCode'  ";
```

```
$hasil4=pg_exec($conn,$q4);
$baris4 = pg_numrows($hasil4);

   $q5 = "Insert INTO branch_temp values
('$newCode','$txtKodCaw','$txtComname','$txtAdd1','$txtAdd2','$txtAdd3','$txtAdd4','$txtTel','$txtFax
','$txtEmail','$txtRep','A','$txtHp')";
        $hasil5=pg_send($q5);

}

if ($status == 3) // ebra.php
{

   $q8 = "UPDATE branch_temp set tbra_name = '$txtComname', tbra_add1 = '$txtAdd1', tbra_add2 =
'$txtAdd2', tbra_add3 = '$txtAdd3' , tbra_add4 = '$txtAdd4', tbra_tel ='$txtTel', tbra_faks = '$txtFax',
tbra_email ='$txtEmail', tbra_cperson = '$txtrep', tbra_hphone='$txtHp' WHERE tbra_venid =
'$newCode' and tbra_code = '$txtKodCaw' ";
        $hasil8=pg_send($q8);
}

if ($status == 4) //del_bra.php
{

   $q8 = "DELETE from branch_temp WHERE tbra_venid = '$newCode' and tbra_code =
'$txtKodCaw' ";
        $hasil8=pg_send($q8);
}



$q7  = "select * from vendor_temp where tven_id = '$newCode'  ";
$hasil7 = pg_exec($conn,$q7);
$baris7 = pg_numrows($hasil7);

if ($baris7 != 0 )
{
$values=pg_fetch_array($hasil7,0,PGSQL_ASSOC);
$NamaV = $values['tven_comname']."";
$RegV = $values['tven_regno']."";
$Saiz = $values['tven_size']."";
}

?>


<html>
<head>
<title>E-Procurement for IT Stuff :: Admin Module</title>

<style>
<!--

a {text-decoration:none;}

-->
</style>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script language="JavaScript">
<!--

function keluar(a,v,w)
{
  var agree=confirm("Delete this branch?");

 if (agree)
  {
link = "ven_main.php?status=4&newCode="+w+"&txtKodCaw="+v;
  document.frmdept.action=link;
  document.frmdept.target = "_self";
  document.frmdept.method="post";
  document.frmdept.submit();
  return true;
  }
 else
 return false;
}


function f(o)
{
  o.value=o.value.toUpperCase();
}

function vad()
{
var ErrMsgg = ""
var ErrSet = "False"


if (document.frmdept.txtVen.value == "")
{
 ErrSet = "True"
 ErrMsgg +="* Please enter vendor name.  \n"
}

if (document.frmdept.txtReg.value == "")
{
 ErrSet = "True"
 ErrMsgg +="* Please enter register no.  \n"
}

if (document.frmdept.txtSize.options[0].selected == true)
{
 ErrSet = "True"
 ErrMsgg +="* Please enter company size.  \n"
}

k=0;
for(i=0;i< document.frmdept.length;i++)
{
e=document.frmdept.elements[i];
if (e.type=='hidden' && e.name=='pakar')
k++;
}

if (k < 1 )
```

```
{
 ErrSet = "True"
 ErrMsgg +="* Please add expertise of this vendor. \n"
}


j=0;
for(i=0;i< document.frmdept.length;i++)
{
e=document.frmdept.elements[i];
if (e.type=='hidden' && e.name=='caw')
j++;
}

if (j < 1 )
{
 ErrSet = "True"
 ErrMsgg +="* Please add branch info of this vendor. \n"
}


if (ErrSet == "True")
 {
 alert(ErrMsgg);
 return false;
 }
else
 {
   document.frmdept.txtKod.disabled = false;
         return true;
         document.frmdept.txtKod.disabled = false;
 }
}

function vad_bra()
{
var ErrMsgg = ""
var ErrSet = "False"

if (document.frmdept.txtVen.value == "")
{
 ErrSet = "True"
 ErrMsgg +="* Please enter this company name.  \n"
}

if (document.frmdept.txtReg.value == "")
{
 ErrSet = "True"
 ErrMsgg +="* Please enter this register no.  \n"
}


if (document.frmdept.txtSize.options[0].selected == true)
{
 ErrSet = "True"
 ErrMsgg +="* Please select this company size.  \n"
}


if (ErrSet == "True")
```

```
 {
 alert(ErrMsgg);
 return "false";
 }
else
        {

        return "true";
        }
}


function ad()
{
document.frmdept.txtKod.disabled = false;
document.frmdept.action = "addex.php"
document.frmdept.method = "post"
document.frmdept.submit();
}

function adbra()
{
document.frmdept.action = "new_bra.php";
document.frmdept.method = "post";

action = vad_bra();

if (action == "true" )
 {
  document.frmdept.txtKod.disabled = false;
  document.frmdept.submit();
 }
}

function canl()
{
document.frmdept.txtKod.disabled = false;
document.frmdept.action = "ven_main.php?status=0"
document.frmdept.method = "post"
document.frmdept.submit();
}

//-->
</script>
</head>

<body bgcolor="#FFFFFF" text="#000000"
background="/PHP_WXES3182/Images/HOMES_Banner_Pale_Small_B&W.JPG">
<table width="750" border="0" bgcolor="#0f87ff" align="center">
 <tr>
   <td height="16">
<font size="2" face="Arial, Helvetica, sans-serif" color ="#ffffff" ><b>
     New Vendor</b></font>
 </tr>
</table>
<table width="750" border="1" bordercolor="#0f87ff" bgcolor="#0f87ff" align="center">
 <tr>
     <td valign="top" height="223" bgcolor="#FFFFFF">
    <form name="frmdept" method="POST" action="ven_save.php" onsubmit= "return vad()" >
      <table width="580" border="0" align="center" height="148">
```

```
<tr>
 <td height="152">
 <br>
  <table width="512" border="0" height="60" align="center">
   <tr border=1 borderclor=white>
    <td colspan=3 bgcolor="#ffcc66"><font color=#0000cc size=2 face=Arial, Helvetica, sans-
serif>
     <b>Vendor Expertise</b></font></td>
   </tr>
   <tr>
    <td width="139" height="11" bgcolor="#ffddbb"><b><font face="Courier New, Courier,
mono" size="2">Vendor
     Code </font></b></td>
    <td width="10" height="11" bgcolor="#ffddbb"><b><font face="Courier New, Courier,
mono" size="2">:</font></b></td>
    <td width="349" height="11" bgcolor="#ffddbb"><b><font face="Courier New, Courier,
mono" size="2">
     <input type="text" name="txtKod"  value = "<?echo "$newCode"  ?>"  disabled >
     </font></b></td>
   </tr>
   <tr>
    <td width="139" height="26" bgcolor="#ffddbb"><b><font face="Courier New, Courier,
mono" size="2">Vendor
     Name </font></b></td>
    <td width="10" height="26" bgcolor="#ffddbb"><b><font face="Courier New, Courier,
mono" size="2">:</font></b></td>
    <td width="349" height="26" bgcolor="#ffddbb"> <b><font face="Courier New, Courier,
mono">
     <?if ($baris7 == 0)
     echo "<input type=\"text\" name=\"txtVen\" size=\"50\" maxlength=\"50\" value=\"\"
onkeydown=\"f(this)\"  onkeyup=\"f(this)\" onblur=\"f(this)\" onclick=\"f(this)\" >";
                                    else
                                    echo "<input type=\"text\" name=\"txtVen\" size=\"50\"
maxlength=\"50\" value =\"$NamaV\" onkeydown=\"f(this)\" onkeyup=\"f(this)\" onblur=\"f(this)\"
onclick=\"f(this)\" >";
                                    ?>
     </font></b></td>
   </tr>
   <tr>
    <td width="139" height="27" bgcolor="#ffddbb"><b><font face="Courier New, Courier,
mono" size="2">Register
     No</font></b></td>
    <td width="10" height="27" bgcolor="#ffddbb"><b><font face="Courier New, Courier,
mono">:</font></b></td>
    <td width="349" height="27" bgcolor="#ffddbb"><b><font face="Courier New, Courier,
mono" size="2">
     <?if ($baris7 ==0)
     {echo "<input type=\"text\" name=\"txtReg\" size=\"20\" maxlength=\"20\" value=\"\"
onkeydown=\"f(this)\" onkeyup=\"f(this)\" onblur=\"f(this)\" onclick=\"f(this)\">";}
                                    else
                                    echo "<input type=\"text\" name=\"txtReg\" size=\"20\"
maxlength=\"20\" value =\"$RegV\" onkeydown=\"f(this)\" onkeyup=\"f(this)\" onblur=\"f(this)\"
onclick=\"f(this)\" >";
                                    ?>
     </font></b></td>
   </tr>
   <tr bgcolor="#ffddbb">
    <td width="139" height="27"><b><font size="2" face="Courier New, Courier,
mono">Company
     Size</font></b></td>
```

```
            <td width="10" height="27"><b><font size="2" face="Courier New, Courier,
mono">:</font></b></td>
            <td width="349" height="27"><b><font size="2" face="Courier New, Courier, mono">
             <?
                                          if ($baris7 ==0)
            {echo "<select name=txtSize><option value=0> << Please Select The Size >>
</option><option value=SMALL>Small</option><option
value=MEDIUM>Medium</option><option value=LARGE>Large</option>";}
                                          else
            {
                                          if ($Saiz == '0')
                                          echo "<select name=txtSize><option value=0
selected>Please Select The Size</option><option value=SMALL>Small</option><option
value=MEDIUM>Medium</option><option value=LARGE>Large</option>";
                                          else if ($Saiz == 'SMALL')
                                          echo "<select name=txtSize><option value=0
selected>Please Select The Size</option><option value=SMALL selected>Small</option><option
value=MEDIUM>Medium</option><option value=LARGE>Large</option>";
                                          else if ($Saiz == 'MEDIUM')
                                          echo "<select name=txtSize><option value=0
selected>Please Select The Size</option><option value=SMALL selected>Small</option><option
value=MEDIUM Selected>Medium</option><option value=LARGE>Large</option>";
                                          else if ($Saiz == 'LARGE')
                                          echo "<select name=txtSize><option value=0
selected>Please Select The Size</option><option value=SMALL selected>Small</option><option
value=MEDIUM>Medium</option><option value=LARGE selected>Large</option>";
                                          }
                                          ?>
             </font></b></td>
           </tr>
           <tr bgcolor="#ffddbb">
            <td width="139" height="27"><b><font size="2" face="Courier New, Courier,
mono">Expertise</font></b></td>
            <td width="10" height="27"><b><font size="2" face="Courier New, Courier,
mono">:</font></b></td>
            <td width="349" height="27"><b><font face="Courier New, Courier, mono" size="2">
             <?php
                                          if ($status != 0)
                                          {
                                                $kep1 = "";
                                                $q1  = "select * from vendor_expertise_temp
where texven_venid = '$newCode'  ";
                                                $hasil1=pg_exec($conn,$q1);
                                                for ($row=0; $row<pg_numrows($hasil1);
$row++)
                                                {

        $values=pg_fetch_array($hasil1,$row,PGSQL_ASSOC);
                                                $kodk = $values['texven_excode']."";


                                                $cek_id2  = "select * from expertise
where ex_code = '$kodk' ";
                                                $result2=pg_exec($conn,$cek_id2);

        $values=pg_fetch_array($result2,0,PGSQL_ASSOC);

                                                $kep = $values['ex_name']."";
                                                $kep1 .= "$kep \n";
```

```
                                                    echo "<input type=hidden name=pakar
>";
                                                    echo "<input type=hidden
name='kepitem[]' value= '$kep'>";
                                                }  // for
                                                echo "<textarea name=\"txtExp\"   cols=\"30\"
rows=\"2\">$kep1</textarea>";
                                                }
                                        else
                                        { echo "<textarea name=\"txtExp\"   cols=\"30\"
rows=\"2\" disabled> </textarea>"; }


                                ?>
            <input type="button" name="btnTambah" value="+" onClick="ad()">
            </font> </b></td>
          </tr>
        </table>
        <br>
        <?
$q6  = "select * from branch_temp where tbra_venid = '$newCode'  ";
$hasil6=pg_exec($conn,$q6);
$baris6 = pg_numrows($hasil6);
?>
        <table width =512 align =center >
          <tr bordercolor=white bgcolor="#ffcc66">
            <td id=ignore colspan =4><font color=#0000cc size=2 face=Arial, Helvetica, sans-serif>
              <b>Branch Info</b></font></td>
          </tr>
          <?
                                If (status != 0)
                                {

                                echo "<tr bordercolor=white > ";
        echo "<td bgcolor=#ffddbb onclick=\"adbra()\" colspan =4>";
                                echo "<font size=\"2\" face=\"Courier New, Courier, mono\"
color=\"#000000\"><u>Please
click for adding branch<u></font> </td></tr>";
                                } //if
                                else
                                {
                                        if ($baris6 > 0 )
                                {
echo "<tr bgcolor=\"#ffddbb\" bordercolor=white><font color=\"#000000\" size=\"2\" face=\"Courier
New, Courier, mono\"><td id=ignore width=20></td><td id=ignore width=20> </td><td  width=100
>Name</td> <td width=350 >Branch Address</td></font></tr>";
                                                for ($row=0; $row<$baris6; $row++)
                                                {


$values=pg_fetch_array($hasil6,$row,PGSQL_ASSOC);
                                                $tkod = $values['tbra_code']."";
                                                $tcaw = $values['tbra_name']."";
                                                $tadd1 = $values['tbra_add1']."";
                                                $tadd2 = $values['tbra_add2']."";
                                                $tadd3 = $values['tbra_add3']."";
                                                $tadd4 = $values['tbra_add4']."";
                                                $tTel = $values['tbra_tel']."";
                                                $tFax = $values['tbra_faks']."";
```

```
                                        $tEmail = $values['tbra_email']."";
                                        $tperson = $values['tbra_cperson']."";

                                        if($tadd2 =="" && $tadd3 =="" && $tadd4 ==
""){

                                        $add = $tadd1."<BR>";
                                        else if($tadd3 =="" && $tadd4 == "")
                                        $add = $tadd1."<BR>".$tadd2;
                                        else if($tadd2 =="" && $tadd4 == "")
                                        $add = $tadd1."<BR>".$tadd3;
                                        else if($tadd2 =="" && $tadd3 == "")
                                        $add = $tadd1."<BR>".$tadd4;

                                        else if($tadd4 == "")
                                        $add = $tadd1."<BR>".$tadd2."<br>".$tadd3;
                                        else if($tadd3 == "")
                                        $add = $tadd1."<BR>".$tadd2."<br>".$tadd4;
                                        else if($tadd2 == "")
                                        $add = $tadd1."<BR>".$tadd3."<br>".$tadd4;
                                        else
                                        $add =
$tadd1."<BR>".$tadd2."<br>".$tadd3."<br>".$tadd4;


                                        if($tFax == "")
                                        $contact = $tTel;
                                        else
                                        $contact = $tTel." , ".$tFax;

                                        if($tEmail == "")
                                        $last = $tperson;
                                        else
                                        $last = $tEmail."<br>".$tperson;

                                        echo "<tr  bgcolor=\"#ffddbb\" >";


        echo "<td id=ignore><a href
=\"ebra.php?KodP=$newCode&KodC=$tkod&NamaC=$NamaV\"><img
src=/PHP_WXES3182/Images/search.gif width=\"16\" height=\"14\" border=0 alt=\"Edit
Branch\"></a></td>";

 if ($baris6 >1)
echo "<td  width =20 ><input type=\"image\" border=\"0\" name=\"btndelete\"
src=\"/PHP_WXES3182/Images/del1.gif\"
width=\"20\" height=\"20\" alt=\"Delete\" onClick=\"return
keluar(this.form,'$tkod','$newCode')\"></center></td>";

            else
        echo "<td id=ignore><center> -</center></td>";

        echo "<td ><input type=hidden name=caw>$tcaw</td>";
        echo "<td >$add <br> $contact <br> $last</td>";
        echo "</tr>";
                                        }//for
                                    }//if

                                } //else
                                ?>
    </table>
```

```
        </td>
       </tr>
       <tr>
        <td height="26">
         <div align="center">
          <input type="button" name="btnadd" value="Add branch" onClick="adbra()"> 
          <input type="submit" name="btnSave" value="Save"> 
          <input type="button" name="btnBatal" value="Cancel" onClick = "canl()">
         </div>
        </td>
       </tr>
      </table>
     </form>
    </td>
   </tr>
<tr>
    <td height="2" bgcolor="#0f87ff" >
     <p>
      <!-- Date and time -->
      <script language="JavaScript">
            <!--
                        function showClock()
                              {

                               if (!document.all&&!document.getElementById)
                                     return
                                                thelement=document.getElementById?
document.getElementById("tick2"): document.all.tick2
                               var Digital=new Date()
                               var hours=Digital.getHours()
                               var minutes=Digital.getMinutes()
                               var seconds=Digital.getSeconds()
                               var year = Digital.getYear();
                               var mon = Digital.getMonth();
                               var day = Digital.getDate();
                               var gmon= ""

                               if (mon==0)
                                gmon += "January"
                               else if (mon==1)
                                     gmon += "February"
                               else if (mon==2)
                                     gmon += "March"
                               else if (mon==3)
                                     gmon += "April"
                               if (mon==4)
                                     gmon += "May"
                               if (mon==5)
                                     gmon += "June"
                               if (mon==6)
                                     gmon += "July"
                               if (mon==7)
                                     gmon += "August"
                               if (mon==8)
                                     gmon += "September"
                               if (mon==9)
                                     gmon += "October"
                               if (mon==10)
                                     gmon += "November"
                               if (mon==11)
```

```
                                        gmon += "December"

                                        var dn="PM"

                                        if (hours<12)
                                                dn="AM"
                                        if (hours>12)
                                                hours=hours-12
                                        if (hours==0)
                                                hours=12
                                        if (minutes<=9)
                                                minutes="0"+minutes
                                        if (seconds<=9)
                                                seconds="0"+seconds

                                        var ctime=day + " " + gmon +" " + year + " ( "
+hours+":"+minutes+":"+seconds+" "+dn
                                                thelement.innerHTML="<b style='font-
size:12;color:#ffffff;face:verdana'>"+ctime+" )</b>"
                                                setTimeout("showClock()",1000)
                                        }
                                window.onload=showClock

                                //-->
                        </script>
    </p>
    <div align="center"></div>
    <div align="center"><span id=tick2> </span> </div><br>
    </td>
  </tr>

</table>
</body>
</html>
```

C: Add Status Produce PO

```
<?php
session_start();

include("all_admin.php");

$conn = pg_open();

if (!$conn)
{
print("Database connection failed.");
exit;
}

if ($jb == 'D01')
{

$q29 = "Select * from application_item, ven_call_tender, vendor where apem_appcode=vpp_kodperm
and apem_buytype = '$jb' and apem_appcode='$KodP' and apem_group='$kumpulan' and
vpp_kodvendor=ven_id";
$hasil29=pg_send($q29);
$count29 = pg_numrows($hasil29);

$values1=pg_fetch_array($hasil29,0,PGSQL_ASSOC);
$value = $values1['vpp_price']."";
$ven = $values1['ven_comname']."";
$pembekal = $values1['ven_id']."";

$q3 = "Select * from application_item where apem_buytype = '$jb' and apem_appcode='$KodP' and
apem_group='$kumpulan' AND apem_statuscode='S13'";
$hasil3=pg_send($q3);
$count3 = pg_numrows($hasil3);

}

else
{
$q29 = "Select * from application_item,doc_agreement,vendor where apem_appcode=dst_kodperm
and apem_itemcode = dst_koditem and  apem_buytype = '$jb' and apem_appcode='$KodP' and
apem_group='$kumpulan' and dst_kodven=ven_id";
$hasil29=pg_send($q29);
$count29 = pg_numrows($hasil29);

$values1=pg_fetch_array($hasil29,0,PGSQL_ASSOC);
$value = $values1['dst_harga']."";
$ven = $values1['ven_comname']."";
$pembekal = $values1['ven_id']."";

$q3 = "Select * from application_item where apem_buytype = '$jb' and apem_appcode='$KodP' and
apem_group='$kumpulan' AND apem_statuscode='S10'";
$hasil3=pg_send($q3);
$count3 = pg_numrows($hasil3);

}

$q1 = "Select * from application,department where app_code='$KodP' and app_deptcode=dept_code";
$hasil1=pg_send($q1);
$values11=pg_fetch_array($hasil1,0,PGSQL_ASSOC);
```

```
$nogaji = $values11['app_salaryno']."";
$jab_kod = $values11['app_deptcode']."";
$deptName = $values11['dept_name']."";

$q21 = "Select * from applicant where appt_salaryno = '$nogaji'";
$hasil21=pg_send($q21);
$count21 = pg_numrows($hasil21);

if($count21 >0)
{
$values21=pg_fetch_array($hasil21,0,PGSQL_ASSOC);
$aptNm = $values21['appt_name']."";
$telpemohon = $values21['appt_tel']."";
}
else
{
$q22 = "Select * from administrator where adm_salaryno='$nogaji'";
$hasil22=pg_send($q22);
$values22=pg_fetch_array($hasil22,0,PGSQL_ASSOC);
$aptNm = $values22['adm_name']."";
$telpemohon = $values22['adm_tel']."";

}

$q2 = "Select * from application_item where apem_buytype = '$jb' and apem_appcode='$KodP' and
apem_group='$kumpulan' AND apem_statuscode='S11'";
$hasil2=pg_send($q2);
$count2 = pg_numrows($hasil2);

if($count29 < 1)
{
$keadaan = 1;
}


if ($count2 > 0)
{
$keadaan = 2;
}


if ($count3 < 1)
{
$keadaan = 3;
}



?>

<html>
<head>
<title>E-Procurement for IT Stuff :: Admin Module</title>

<style>
<!--
TABLE.Mtable TD {
BORDER-BOTTOM: #c1cdd8 1px solid;
BORDER-left: #c1cdd8 1px solid;
```

```
}

TABLE.Mtable1 TD {
BORDER-left: #c1cdd8 1px solid;
}


a {text-decoration:none;}

-->
</style>


<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<script language="JavaScript">
<!--

var gdCtrl = new Object();
var goSelectTag = new Array();
var gcGray = "#808080";
var gcToggle = "#cccc99";
var gcBG = "#ccffff";
var gcOn = "#ffffff";
var blue ="#0000ff";
var red="#ff0000";

var gdCurDate = new Date();
var giYear = gdCurDate.getFullYear();
var giMonth = gdCurDate.getMonth()+1;
var giDay = gdCurDate.getDate();

function fSetDate(iYear, iMonth, iDay){
VicPopCal.style.visibility = "hidden";
gdCtrl.value = iDay+"-"+iMonth+"-"+iYear; //Here, you could modify the locale as you need !!!!
for (i in goSelectTag)
goSelectTag[i].style.visibility = "visible";
goSelectTag.length = 0;
}

function fSetSelected(aCell){
var iOffset = 0;
var iYear = parseInt(tbSelYear.value);
var iMonth = parseInt(tbSelMonth.value);

self.event.cancelBubble = true;
aCell.bgColor = gcBG;
with (aCell.children["cellText"]){
var iDay = parseInt(innerText);
if (color==gcGray)
iOffset = (Victor<10)?-1:1;
iMonth += iOffset;
if (iMonth<1) {
iYear--;
iMonth = 12;
}else if (iMonth>12){
iYear++;
iMonth = 1;
}
}
```

```
fSetDate(iYear, iMonth, iDay);
}

function Point(iX, iY){
this.x = iX;
this.y = iY;
}

function fBuildCal(iYear, iMonth) {
var aMonth=new Array();
for(i=1;i<7;i++)
aMonth[i]=new Array(i);

var dCalDate=new Date(iYear, iMonth-1, 1);
var iDayOfFirst=dCalDate.getDay();
var iDaysInMonth=new Date(iYear, iMonth, 0).getDate();
var iOffsetLast=new Date(iYear, iMonth-1, 0).getDate()-iDayOfFirst+1;
var iDate = 1;
var iNext = 1;

for (d = 0; d < 7; d++)
aMonth[1][d] = (d<iDayOfFirst)?-(iOffsetLast+d):iDate++;
for (w = 2; w < 7; w++)
for (d = 0; d < 7; d++)
aMonth[w][d] = (iDate<=iDaysInMonth)?iDate++:-(iNext++);
return aMonth;
}

function fDrawCal(iYear, iMonth, iCellHeight, iDateTextSize) {
var WeekDay = new Array("S","M","T","W","T","F","S");
var styleTD = " bgcolor='"+gcBG+"' bordercolor='"+gcBG+"' valign='middle' align='center'
height='"+iCellHeight+"' style='font:bold "+iDateTextSize+" Courier;";          //Coded by Liming
Weng(Victor Won)  email:victorwon@netease.com

with (document) {
write("<tr>");
for(i=0; i<7; i++)
write("<td "+styleTD+"color:blue' >" + WeekDay[i] + "</td>");
write("</tr>");

for (w = 1; w < 7; w++) {
write("<tr>");
for (d = 0; d < 7; d++) {
write("<td id=calCell "+styleTD+"cursor:hand;' onMouseOver='this.bgColor=gcToggle'
onMouseOut='this.bgColor=gcBG' onclick='fSetSelected(this)'>");
write("<font id=cellText Victor='Liming Weng'> </font>");
write("</td>")
}
write("</tr>");
}
}
}

function fUpdateCal(iYear, iMonth) {
myMonth = fBuildCal(iYear, iMonth);
var i = 0;
for (w = 0; w < 6; w++)
for (d = 0; d < 7; d++)
with (cellText[(7*w)+d]) {
Victor = i++;
```

```
if (myMonth[w+1][d]<0) {
color = gcGray;
innerText = -myMonth[w+1][d];
}else{
color = ((d==0)||(d==6))?"red":"black";
innerText = myMonth[w+1][d];
}
}
}

function fSetYearMon(iYear, iMon){
tbSelMonth.options[iMon-1].selected = true;
for (i = 0; i < tbSelYear.length; i++)
if (tbSelYear.options[i].value == iYear)
tbSelYear.options[i].selected = true;
fUpdateCal(iYear, iMon);
}

function fPrevMonth(){
var iMon = tbSelMonth.value;
var iYear = tbSelYear.value;

if (--iMon<1) {
iMon = 12;
iYear--;
}

fSetYearMon(iYear, iMon);
}

function fNextMonth(){
var iMon = tbSelMonth.value;
var iYear = tbSelYear.value;

if (++iMon>12) {
iMon = 1;
iYear++;
}

fSetYearMon(iYear, iMon);
}

function fToggleTags(){
with (document.all.tags("SELECT")){
for (i=0; i<length; i++)
if ((item(i).Victor!="Won")&&fTagInBound(item(i))){
item(i).style.visibility = "hidden";
goSelectTag[goSelectTag.length] = item(i);
}
}
}

function fTagInBound(aTag){
with (VicPopCal.style){
var l = parseInt(left);
var t = parseInt(top);
var r = l+parseInt(width);
var b = t+parseInt(height);
var ptLT = fGetXY(aTag);
return !((ptLT.x>r)||(ptLT.x+aTag.offsetWidth<l)||(ptLT.y>b)||(ptLT.y+aTag.offsetHeight<t));
```

```
}
}

function fGetXY(aTag){
var oTmp = aTag;
var pt = new Point(0,0);
do {
pt.x += oTmp.offsetLeft;
pt.y += oTmp.offsetTop;
oTmp = oTmp.offsetParent;
} while(oTmp.tagName!="BODY");
return pt;
}

// Main: popCtrl is the widget beyond which you want this calendar to appear;
//       dateCtrl is the widget into which you want to put the selected date.
// i.e.: <input type="text" name="dc" style="text-align:center" readonly><INPUT type="button"
value="V" onclick="fPopCalendar(dc,dc);return false">
function fPopCalendar(popCtrl, dateCtrl){
gdCtrl = dateCtrl;
fSetYearMon(giYear, giMonth);
var point = fGetXY(popCtrl);
with (VicPopCal.style) {
left = point.x;
top  = point.y+popCtrl.offsetHeight+1;
width = VicPopCal.offsetWidth;
height = VicPopCal.offsetHeight;
fToggleTags(point);
visibility = 'visible';
}
VicPopCal.focus();
}

function fHideCal(){
var oE = window.event;
if
((oE.clientX>0)&&(oE.clientY>0)&&(oE.clientX<document.body.clientWidth)&&(oE.clientY<docu
ment.body.clientHeight)) {
var oTmp = document.elementFromPoint(oE.clientX,oE.clientY);
while ((oTmp.tagName!="BODY") && (oTmp.id!="VicPopCal"))
oTmp = oTmp.offsetParent;
if (oTmp.id=="VicPopCal")
return;
}
VicPopCal.style.visibility = 'hidden';
for (i in goSelectTag)
goSelectTag[i].style.visibility = "visible";
goSelectTag.length = 0;
}

var gMonths = new
Array("January","February","March","April","May","June","July","August","September","October","
November","December");

with (document) {
write("<Div id='VicPopCal' onblur='fHideCal()' onclick='focus()'
style='POSITION:absolute;visibility:hidden;border:1px ridge;width:10;z-index:100;'>");
write("<table border='0' bgcolor='#000099'>");
write("<TR>");
```

```
write("<td valign='middle' align='center'><input type='button' name='PrevMonth' value='<'
style='height:20;width:20;FONT:12 Fixedsys' onClick='fPrevMonth()' onblur='fHideCal()'>");
write("  <select name='tbSelMonth' onChange='fUpdateCal(tbSelYear.value,
tbSelMonth.value)' Victor='Won' onclick='self.event.cancelBubble=true' onblur='fHideCal()'>");
for (i=0; i<12; i++)
write("<option value='"+(i+1)+"'>"+gMonths[i]+"</option>");
write("</SELECT>");
write("  <SELECT name='tbSelYear' onChange='fUpdateCal(tbSelYear.value,
tbSelMonth.value)' Victor='Won' onclick='self.event.cancelBubble=true' onblur='fHideCal()'>");
for(i=1990;i<2015;i++)
write("<OPTION value='"+i+"'>  "+i+"  </OPTION>");
write("</SELECT>");
write("  <input type='button' name='PrevMonth' value='>'
style='height:20;width:20;FONT:16 Fixedsys' onclick='fNextMonth()' onblur='fHideCal()'>");
write("</td>");
write("</TR><TR>");
write("<td align='center'>");
write("<DIV style='background-color:white;'><table width='100%' border='0'>");
fDrawCal(giYear, giMonth, 18, 16);
write("</table></DIV>");
write("</td>");
write("</TR><TR><TD align='center'>");
write("<B style='color:#ffffff;cursor:hand' onclick='fSetDate(giYear,giMonth,giDay);
self.event.cancelBubble=true' onMouseOver='this.style.color=gcToggle'
onMouseOut='this.style.color=gcOn'>Today :  "+gMonths[giMonth-
1]+" "+giDay+",  "+giYear+"</B>");
write("</TD></TR>");write("</TD></TR>");
write("</TABLE></Div>");
}
//**********************

function printPO()
{
document.frmpermohonan.action = "PO.php";
document.frmpermohonan.target = "_blank";
document.frmpermohonan.method = "post";
action = valid();

if (action == "true")
{
document.frmpermohonan.submit();
}
}

function send()
{
document.frmpermohonan.action = "upform_main.php?status=14" ;
document.frmpermohonan.target = "_self";
document.frmpermohonan.method = "post";
action = valid();

if (action == "true")
{
document.frmpermohonan.submit();
}
}


function format(v,d)
{
```

```
var str="" + Math.round(eval(v) * Math.pow(10,d));
while (str.length <= d)
{
str = "0" + str;
}
var dc = str.length - d;
return str.substring (0,dc) + "."+ str.substring(dc,str.length);

}

function chgFormat(bil)
{
var a = document.frmpermohonan
var hargac=0;

var harga = eval("document.frmpermohonan.elements['seunit[" + bil + "]'].value")


if (isNaN(harga)==false)
{
hargac = format(harga,2)
}

return hargac
}//Kira jumlah seunit



function countT(bil)
{
var a = document.frmpermohonan
var total=0;

var harga = eval("document.frmpermohonan.elements['seunit[" + bil + "]'].value")
var unit1 = eval("a.elements['hkuantiti[" + bil + "]'].value")


if (isNaN(harga)==false)
{
jumlah = harga * unit1;
jumlah = format(jumlah,2);
}

return jumlah
}//Kira jumlah seunit



function countTotal()
{
total = 0;
for (w = 0; w < document.frmpermohonan.bilitem.value ; w++)
{
jumlahitem =  eval("document.frmpermohonan.elements['jumitem[" + w + "]'].value");
total = total + parseFloat(jumlahitem);
}

total = format(total,2);
document.frmpermohonan.jumharga.value=total;
```

```
}

//*****************

function valid()
{
var ErrorMsg1 = ""
var ErrFlag1 = "False"

var a = document.frmpermohonan

if (document.frmpermohonan.dc.value == "")
{
ErrFlag1 = "True"
ErrorMsg1 +="* Please enter date.  \n"
}

if (document.frmpermohonan.txtPO.value == "")
{
ErrFlag1 = "True"
ErrorMsg1 +="* Please enter Purchase Order Form No. (PO).  \n"
}

if (document.frmpermohonan.txtNamaCaw.options[0].selected == true)
{
ErrFlag1 = "True"
ErrorMsg1 +="* Please choose vendor's branch name.  \n"
}

i=0;
for (w = 0; w < document.frmpermohonan.bilitem.value; w++)
{
uom = eval("a.elements['UOM[" + w + "]'].value")
if (uom == null || uom == "")
{i=i+1;}
}

i = parseInt(i);
if (i > 0)
{
ErrFlag1 = "True"
ErrorMsg1 +="* Please enter UOM for each item.  \n"
}

if (parseFloat(a.pharga.value)!= parseFloat(a.jumharga.value))
{
//alert (a.pharga.value);
ErrFlag1 = "True"
ErrorMsg1 +="* Please make sure that the total item price is equal to tender value. \n"
}


if (ErrFlag1 == "True")
{
```

```
alert(ErrorMsg1)
return "false"
}
else
return "true"
}

function f(o)
{
o.value=o.value.toUpperCase();
}

function Back(a)
{

document.frmpermohonan.action="upform_main.php";
document.frmpermohonan.method="post";
document.frmpermohonan.target="_self";
document.frmpermohonan.submit();

}

//-->
</script>
</head>

<body bgcolor="#FFFFFF" text="#000000"
background="/PHP_WXES3182/Images/HOMES_Banner_Pale_Small_B&W.JPG">
<table border="1" cellSpacing="0" cellPadding="0" width="750" align="center" bgColor="#0f87ff"
bordercolor="#0f87ff">
<tr>
<td height="15">
<div align="left"><font size="2" face="Arial, Helvetica, sans-serif" color ="#ffffff" ><b>
<font color="#FFFFFF">Application | Produce P.O</font></b></font></div>
</td>


</tr>
</table>
<table border="1" cellSpacing="0" cellPadding="0" width="750" align="center" bgColor="#0f87ff"
bordercolor="#0f87ff">
<tr>
<td valign="top" height="560" bgcolor="#FFFFFF">
<form name="frmpermohonan" >
<? if ($keadaan==1)
echo "<br><br><br><center><font color=red size=2 face=\"Arial, Helvetica, sans-serif\">NO DATA
FOR ITEM APPLICATION </font></center>";
else if ($keadaan==2)
echo "<br><br><br><center><font color=red size=2 face=\"Arial, Helvetica, sans-serif\">ITEM HAS
GONE THROUGH PRODUCE P.O STATUS </font></center>";
else if ($keadaan==3)
echo "<br><br><br><center><font color=red size=2 face=\"Arial, Helvetica, sans-serif\">ITEM
NEEDS TO FOLLOW CORRECT PROCEDURE SEQUENCE.</font></center>";

else
{
?>
```

```
<br>
<table width="591" border="0" align="center" height="260">
<tr>
<td height="380">
<div align="center">
<br>
<input type=hidden name=txtPermohonan value ="<? echo"$KodP" ?>">
<input type=hidden name=kumpulan value ="<? echo"$kumpulan" ?>">
<input type=hidden name=jb value="<? echo"$jb" ?>">
<input type=hidden name=ppembekal value="<? echo"$pembekal" ?>">
<input type=hidden name=pharga value="<? echo"$value" ?>">
<input type=hidden name=kodJab value="<? echo"$jab_kod" ?>">
<input type=hidden name=telpemohon value="<? echo"$telpemohon" ?>">
</div>
<table width="600" border="1" align="center" bordercolor="#ffcc66" height="368" >
<tr bordercolor="#FFFFFF">
<td width=146 height="44"><b><font face="Courier New, Courier, mono" size="2"
color="#0000cc">Application
Code</font></b></td>
<td width="13" height="44"><b><font face="Courier New, Courier, mono" size="2"
color="#0000cc">:</font></b></td>
<td width="419" height="44"><b><font face="Courier New, Courier, mono" size="2"
color="#0000cc">
<? echo "$KodP" ?>
</font></b></td>
</tr>
<tr bordercolor="#FFFFFF" valign="baseline">
<td width="146" height="2"><b><font face="Courier New, Courier, mono" size="2">
No P.O.</font></b></td>
<td width="13" height="2"><b><font size="2" face="Courier New, Courier,
mono">:</font></b></td>
<td width="419" height="2"><b><font size="2" face="Courier New, Courier, mono">
<input type="text" name="txtPO">
</font></b></td>
</tr>
<tr bordercolor="#FFFFFF" valign="baseline">
<td width="146" height="2"><b><font face="Courier New, Courier, mono" size="2">Vendor
Name</font></b></td>
<td width="13" height="2"><b><font size="2" face="Courier New, Courier,
mono">:</font></b></td>
<td width="419" height="2"> <b><font face="Courier New, Courier, mono" size="2">
<input type="text" name="txtVen" size="40" value="<? echo"$ven" ?>" READONLY>
</font></b></td>
</tr>
<tr bordercolor="#FFFFFF" valign="baseline">
<td width="146" height="2"><b><font face="Courier New, Courier, mono" size="2">Branch
Name</font></b></td>
<td width="13" height="2"><b><font size="2" face="Courier New, Courier,
mono">:</font></b></td>
<td width="419" height="2"> <b><font face="Courier New, Courier, mono" size="2">
<?


$query23 = "Select * from branch where bra_venid = '$pembekal' order by bra_name";
$result23=pg_send($query23);


echo "<select name=\"txtNamaCaw\" size=\"1\" width=\"80\" >";
echo "<option value=\"0\"><< Please select branch name >></option>";
for ($row2=0; $row2<pg_numrows($result23); $row2++)
```

```
{
$values23=pg_fetch_array($result23,$row2,PGSQL_ASSOC);
$kod = $values23['bra_code'];
$nama = $values23['bra_name'];

echo "<option value=$kod >$nama</font></option>";
}
echo "</select>";

?>
</font></b></td>
</tr>
<tr bordercolor="#FFFFFF" valign="baseline">
<td width="146" height="6"><b><font face="Courier New, Courier, mono" size="2">Send
to</font></b></td>
<td width="13" height="6"><b><font size="2" face="Courier New, Courier,
mono">:</font></b></td>
<td width="419" height="6"> <b><font face="Courier New, Courier, mono" size="2">
<input type="text" name="txtPtj" size="40" value="<? echo"$deptName" ?>" READONLY>
</font></b></td>
</tr>
<tr bordercolor="#FFFFFF" valign="baseline">
<td width="146" height="2"><b><font face="Courier New, Courier, mono" size="2">Officer
Name</font></b></td>
<td width="13" height="2"><b><font size="2" face="Courier New, Courier,
mono">:</font></b></td>
<td width="419" height="2"> <b><font face="Courier New, Courier, mono" size="2">
<input type="text" name="txtStaff" size="40" value="<? echo"$aptNm" ?>" READONLY>
</font></b></td>
</tr>
<tr bordercolor="#FFFFFF" valign="baseline">
<td width="146"><b><font size="2" face="Courier New, Courier, mono">Date</font></b></td>
<td width="13"><b><font size="2" face="Courier New, Courier, mono">:</font></b></td>
<td width="419" ><b><font size="2" face="Courier New, Courier, mono">
<input name=dc readOnly style="TEXT-ALIGN: center">
<INPUT onclick="fPopCalendar(dc,dc); return false" type=button value=V>
</font> </b></td>
</tr>
<tr bordercolor="#FFFFFF" valign="top">
<td width="146" height="2"><b><font face="Courier New, Courier, mono" size="2">
Note</font></b></td>
<td width="13" height="2"><b><font size="2" face="Courier New, Courier,
mono">:</font></b></td>
<td width="419" height="2"> <b><font face="Courier New, Courier, mono" size="2">
<textarea name="txtNota" cols="50" rows="2"></textarea>
</font></b></td>
</tr>
<tr bordercolor="#ffcc66" valign="top" bgcolor="#ffffff">
<td colspan="3">
<?
$q19 = "Select * from application_item,item where apem_itemnamecode=item_kod and
apem_buytype = '$jb' and apem_appcode='$KodP' and apem_group='$kumpulan'  order by
apem_itemcode";
$hasil19=pg_send($q19);
$count19 = pg_numrows($hasil19);
echo "<input type=hidden value=$count19 name=bilitem>";

echo"<table width=600 align=center class=Mtable  cellpadding=0 cellspacing=0 style=\"border-
top:0px;border-right:0px;border-left:1px\">";
```

```
echo"<tr><td width=40 align=center><font size=\"2\" face=\"Arial, Helvetica, sans-serif\"
color=white><b>Code</b></font></td>";
echo"<td width=260 align=\"center\"><font size=\"2\" face=\"Arial, Helvetica, sans-serif\"
color=white><b>Item Name</b></font></td>";
echo"<td width=70 align=\"center\"><font size=\"2\" face=\"Arial, Helvetica, sans-serif\"
color=white><b>Unit</b></font></td>";

echo"<td width=90 align=\"center\"><font size=\"2\" face=\"Arial, Helvetica, sans-serif\"
color=white><b>Price/Unit</b></font></td>";
echo"<td width=90 align=\"center\"><font size=\"2\" face=\"Arial, Helvetica, sans-serif\"
color=white><b>Total</b></font></td>";
echo"</tr>";

for ($row1=0; $row1<pg_numrows($hasil19); $row1++)
{
$values19=pg_fetch_array($hasil19,$row1,PGSQL_ASSOC);
$ic = $values19['apem_itemcode'];
$in = $values19['item_name'];
$qty = $values19['apem_quant'];


echo"<tr bgColor=#ffddbb><td width=40 align=center><font size=\"2\" face=\"Arial, Helvetica, sans-
serif\" >$ic</font></td>";
echo "<td width=260 align=left><font size=\"2\" face=\"Arial, Helvetica, sans-serif\" >
$in</font></td>";
echo "<td width=70 align=center><font size=\"2\" face=\"Arial, Helvetica, sans-serif\" ><input
type=\"hidden\" name=\"hkuantiti[$row1]\" value=$kuantiti> $qty</font></td>";

echo "<td width=90 align=center><input type=\"text\" name=\"seunit[$row1]\" size=\"10\"
style=\"TEXT-ALIGN: center\"  onblur=\"this.value=chgFormat($row1)\"></td>";
echo "<td width=90 align=center> <input type=\"text\" name=\"jumitem[$row1]\" size=\"10\"
style=\"TEXT-ALIGN: center\" onblur=\"countTotal()\" readonly
onfocus=\"this.value=countT($row1)\" value=0></td>";

echo"</tr>";

}
echo"</table>";
echo"<table width=600 cellpadding=0 cellspacing=0 >";
echo"<tr> <td width=510 align=right><font size=\"2\" face=\"Arial, Helvetica, sans-serif\"
color=white><b> Total Price</b></font> </td><td width=90 align=center><input type=\"text\"
name=\"jumharga\" size=10 style=\"TEXT-ALIGN: center\" value=0 readonly></td></tr>";
echo"</table>";
?>
</td>
</tr>
</table>
</td>
</tr>
</table>
<table width="580" border="0" align="center">
<tr>
<td>
<div align="center">
<input type="button" name="btnSubmit" value="Save" onclick="send()">
<input type="button" name="btnPO" value="Print PO" onclick="printPO()">
<input type="button" name="back" value = "Previous Screen"
onclick="Back(document.frmpermohonan)">
</div>
</td>
```

```
</tr>
</table>
<? } ?>
</form>
</td>
</tr>
<tr>
<td height="2" bgcolor="#0f87ff" >
<p>
<!-- Date and time -->
<script language="JavaScript">
<!--
function showClock()
{

if (!document.all&&!document.getElementById)
return
thelement=document.getElementById? document.getElementById("tick2"): document.all.tick2
var Digital=new Date()
var hours=Digital.getHours()
var minutes=Digital.getMinutes()
var seconds=Digital.getSeconds()
var year = Digital.getYear();
var mon = Digital.getMonth();
var day = Digital.getDate();
var gmon= ""

if (mon==0)
gmon += "January"
else if (mon==1)
gmon += "February"
else if (mon==2)
gmon += "March"
else if (mon==3)
gmon += "April"
if (mon==4)
gmon += "May"
if (mon==5)
gmon += "Jun"
if (mon==6)
gmon += "Julai"
if (mon==7)
gmon += "Ogos"
if (mon==8)
gmon += "September"
if (mon==9)
gmon += "Oktober"
if (mon==10)
gmon += "November"
if (mon==11)
gmon += "December"

var dn="PM"

if (hours<12)
dn="AM"
if (hours>12)
hours=hours-12
if (hours==0)
hours=12
```

```
if (minutes<=9)
minutes="0"+minutes
if (seconds<=9)
seconds="0"+seconds

var ctime=day + " " + gmon +" " + year + " ( " +hours+":"+minutes+":"+seconds+" "+dn
thelement.innerHTML="<b style='font-size:12;color:#ffffff;face:verdana'>"+ctime+" )</b>"
setTimeout("showClock()",1000)
}
window.onload=showClock

//-->
</script>
</p>
<div align="center"></div>
<div align="center"><span id=tick2> </span> </div><br>
</td>
</tr>
</table>
<p> </p>
</body>
</html>
```