SUPERMARKET INVENTORY SYSTEM

A Final Year Project Report

Submitted to the

Faculty of Computer Science and Information Technology

University of Malaya

Kuala Lumpur

Dissertation submitted by

GAN JIT SIN

WEK 000066

Supervisor

Miss Fazidah Othman

Moderator

Mrs Nurul Fazmidar

In partial fulfillment of the Degree of

Bachelor of Computer Science

Submission Date: 16 January 2003

Abstract

Supermarket Inventory System (SIS) is a client server system with the objective to computerize and enhance the inventory control system for the supermarket. The inventory control system has four users; they are cashiers, supervisors, clerks and managers. The cashiers will use this system to do the daily transaction. The supervisors can check the current status of the stocks by using the system. The clerks can do the usual clerical job by using this system such as insert, delete, view, check, update and maintain the inventory. The managers can check the daily and monthly sales or profit reports, latest in and out stock record, create new user and news by using this system. The inventory control system for the supermarket is developed using ASP.NET and SQL 2000 to design and manage the database. The inventory control system for the supermarket is developed with the purpose to enhance the effectiveness and efficiency of the existing manual inventory control system. It enables the company to achieve goals and objectives successfully with minimum resources and effort.

Acknowledgement

First and foremost, I would like to take this opportunity to express my gratitude and heartfelt appreciation to my supervisor, Miss Fazidah Othman. Without her support, guidance and advice, this project will not be a reality.

Next, I would like to thank my moderator, Mrs Nurul Fazmidar for being most considerate and kind.

Finally, I would like to thank my friends, especially Lor Chee Loong, Tham Hon Hoe, Ng Chee Wai, Lee Chyng Wei and Sim Kong Wei who have been very kind, helpful and provided me with many valuable advice and assistance during this project.

Table of Contents

List of Tables

CHAPTER 1 : Introduction

1.1 Overview

In this competitive, changing technology business environment, business and trading organizations complete deliberately to provide supreme and satisfactorily services besides the high quality products in order to retain the valued customers and attract new market. In order to accomplish this objective, an organization needs to ensure the completeness of the inventory and efficiency of the business operation. A comprehensive and efficient inventory control system is a key factor to manage the inventory well to avoid the inconvenience and problem of shortage of supply, thus ensures that the products reach the customers in time and as soon as possible. A well plan and design computerized inventory control system enhances the efficiency of the business operation as well. The business activities are undertaken smoothly with minimum effort and time, replacing the manual inventory system that involves traditional ledgers, repetition of inventory calculation and etc that is time consuming and troublesome. In addition computerized inventory system is error proofing to ensure the correctness, accuracy and authenticity of inventory and daily transaction data.

1.2 Problem Statements

Supermarket used to manage the inventory manually. At the end of a day's business, the clerk will need to calculate the total sales and they don't know the quantity of the goods sold on that day.

Besides this, in order to order goods, supervisor will have to check the remaining numbers of the goods manually and this sometime is done with the supplier each time when they need to order goods. This sometimes may cause some problems. This is because, if the supervisor do not pay attention to certain good and the good is out of stock, this may results in the customer complaints.

In order to know the suppliers list, a record is being use for reference manually to get the information of the supplier and the goods supplied by the supplier. If the clerk needs to get the information of the supplier, for example, the supplier's contact number and address, they will need to search for it manually. This is a waste of time and effort.

As a conclusion, there are some problems occur in manual inventory system:

- The records is managed manually
- No message is given when the supply of goods is shortage
- Staffs have to refer to the suppliers list manually to search for the required information
- Calculation of the stock cannot be done often because it is done manually.
- Errors always occur during the calculation

- It is very hard to modify changes of the in stock and out stock records.

With the appearance of these problems, the management level is very hard to make a good decision because they cannot get the latest information they needed in time. This will not happen if the company has a computerized inventory control system.

1.3 Objectives

1.3.1 General Objectives

Computerized inventory control system enhances the effectiveness and efficiency of the existing manual inventory control system. It enables the company to achieve goals and objectives successfully with minimum resources and effort. The company can manage the inventory system well and efficiently. In addition, it contributes to the fast, convenient and error proofing business environment that benefits both the company and the customers. It eventually will reduce or even dismiss the difficulty and error occurred in the manual inventory control process. Besides this, by having a computerized inventory system, it will increase the staff's productivity. The information also can be kept effectively and the information needed is also easily being retrieve when is needed.

1.3.2 Specific Objectives

In this competitive business world, an organization must ensure that it can provide the best service to the customers to retain the old customers and in the same time attract new customers. To fulfill this, the company must have a well-managed inventory system that can provide ample supply to its value customer in time. Instead of calculating the stocks manually, this proposed system also helps the company to make immediate decision when purchase goods needed by just viewing the latest stocks remaining that is updated automatically each time a transaction is made. Besides this, the information also can help the company to estimate its business progress and status at any time.

Besides that, the computerized inventory system also makes the business operation work easy, time-consuming, accurate and more reliable. Previously, the company's clerk will have to calculate the total sales and profit, stocks in and out manually. This manual work is time-consuming, troublesome and more error prone. However with this system, they will get the total sales and profit, stocks in and out easily by just checking it in the inventory system since it is updated automatically.

The managers of the company also can access to the system and get the latest information about the company's current sales and the figures of the stocks remaining in the company. This will make them can make a good decision while planning the company's operation plan.

In addition, the computerized inventory system also reduces the cost and effort needed to manage the inventory. It reduces the collections of the cabinet files and ledgers and builds a paperless business operation environment. Hence, the company able to manage the inventory with minimum effort.

1.4 Project Scope

The users of this Supermarket Inventory System (SIS) are cashiers, supervisors, clerks and managers. The cashiers will use this system to do the daily transaction. The supervisors can check the current status of the stocks by using the system. The clerks can do the usual clerical job by using this system such as insert, delete, view, check, update and maintain the inventory. Besides this the managers can check the daily and monthly sales or profit reports, latest in and out stock record by using this system.

The Supermarket Inventory System enable the users to record or modify the purchasing information, in stocks and return stocks information in the system by using the criteria that had been set. The system will automatically do the calculation that is needed. The system will allow the users to search and retrieve the information of the stocks by using the search function that had been provided by the system.

Besides that, the Supermarket Inventory System also enables the users to record or modify the information of the suppliers and then the users can search and retrieve the information of the suppliers by using the search function that had been provided by the system.

The Supermarket Inventory System also will automatically record the daily transaction into the database and print receipt to the customers. The information of the transaction will be saving in the database and can be retrieving by other authorized users.

The Supermarket Inventory System also will produce reports to the users. The reports are the daily or monthly sales and profit reports. Besides this, the quantity of the remaining stock also will be calculate automatically can be shown to the users when it is needed.

In addition, the system database will be updated automatically and will be backup automatically. Users will need to enter the login name and password in order to enter and use the system. This is mainly to prevent any unauthorized people to access the system and do unlawful things by using the system.

1.5 Significance

1.5.1 Important of the System

- More efficient and fast services can be provided to the customers
- A systematic recording will ease the inventory control of the stocks
- To reduce the manual works to update the inventory
- Makes the business operation work easy, time-consuming, accurate and more reliable
- Information can be retrieved whenever it is required at very high speed

1.6 Hardware / Software Specification

1.6.1 Hardware Requirement

The Minimum hardware requirements to run the Supermarket Inventory System are:

- 800 MHz Processor
- 256 MB Random Access Memory (RAM)
- CD-ROM
- 10 GB Hard Disk
- Other standard peripherals such as keyboard, mouse and monitor

1.6.2 Software Requirement

Microsoft SQL Server 2000

ASP.NET

1.6.3 Operating System

Microsoft Window 2000

1.7 Expected Result

By having this Supermarket Inventory System in the Supermarket, the company is expected to perform the management process more effective and efficient. The previous processes that are being done manually will be change into computerized. This system is expected to provide the latest information to the people in the management level in order to help them in making a good decision. It also expected to reduce the errors that had been made when the calculation is done manually.

Besides that, this system is expected to help the clerks in their daily jobs by simplify the jobs that they need to do. Now, they don't need to calculate and just let the system to do the calculation.

The system is also expected to provide high security to the information in the system.

1.8 Project Schedule

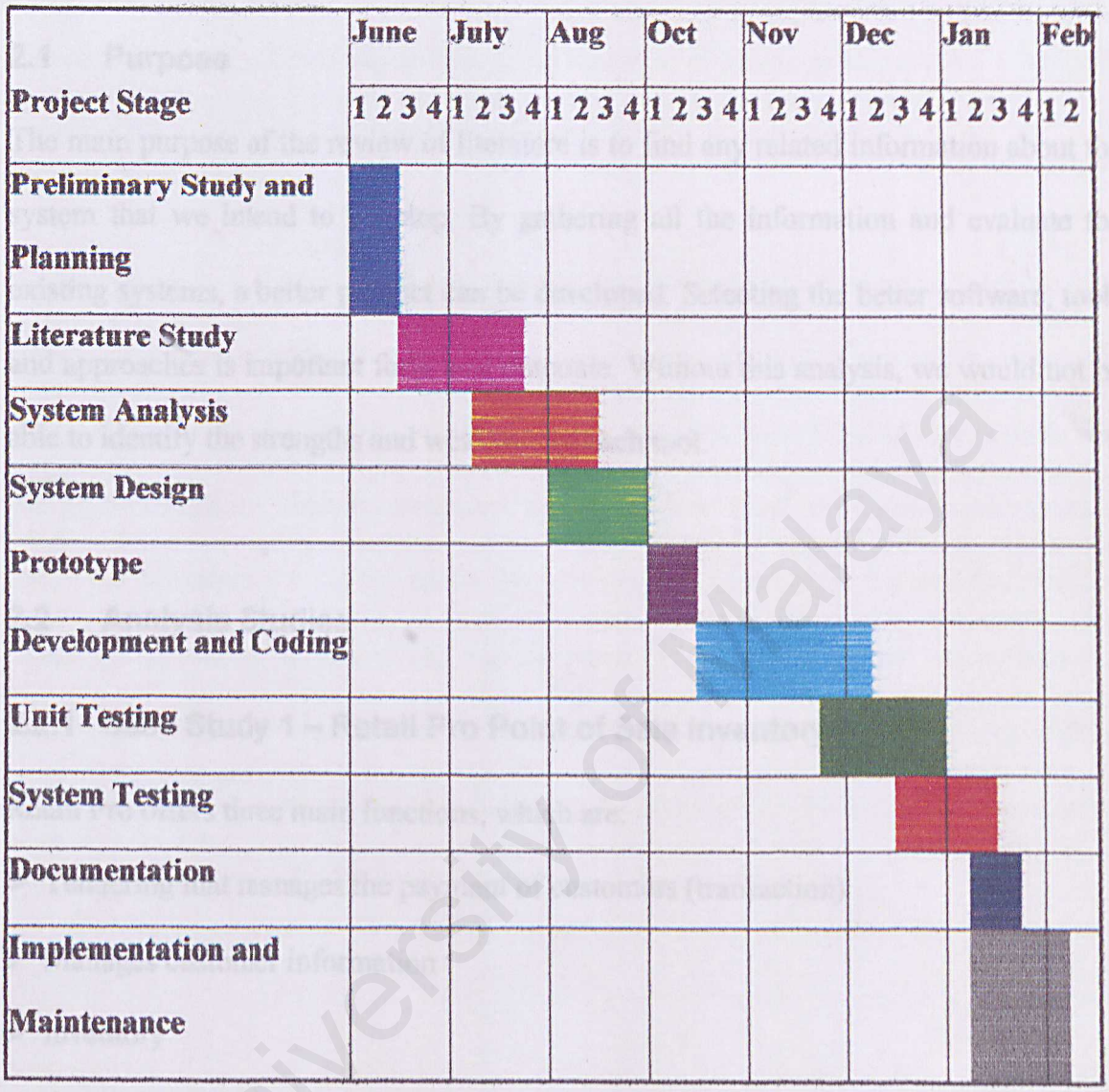


Figure 1-1: Project Schedule

CHAPTER 2 : Review of Literature

2.1 Purpose

The main purpose of the review of literature is to find any related information about the system that we intend to develop. By gathering all the information and evaluate the existing systems, a better product can be developed. Selecting the better software, tools and approaches is important for a best outcome. Without this analysis, we would not be able to identify the strengths and weakness of each tool.

2.2 Analysis Studies

2.2.1 Case Study 1 – Retail Pro Point of Sale Inventory

Retail Pro offers three main functions, which are:

- Tendering that manages the payment of customers (transaction)
- Manages customer information
- Inventory

Tender information window displays important details about the receipt being tendered, for example appropriate price displays important details about the receipt. There are three flexible ways to access to customer's information. Users enter customer's search data and the system will determine the method used to match the data to the particular customer. The methods are exact match, partial match or no match at all. For exact match, users enter the customer's ID number and system will list out the customer's

details. While partial match is when entering the first few characters of a last name, the system will display a user designed customer listing starting with the first possible record. All relevant fields are shown. If no match at all, users press button F3 that will provides the sales associate with a list of alternative look up choices.

Retail Pro keeps an inventory master data set that can instantly retrieve and display item information manually entering any of the codes. Users can check product's availability easily from the inventory data set. This system provides three flexible functions to find particular inventory: find, filtered view and style view. Find functions require users to enter the inventory's code to search the inventory. Filtered view focuses the range of items being search by specifying one or more descriptors (vendors, department and size). Style view presents inventory as a list of styles rather than items, which can be quickly searched by any of the data fields for the style level. This function is convenient for helping customers find a derived piece in a different size or color.

Discussion:

One distinguish feature in Retail Pro Point of Sale Inventory is the system maintaining a customer database that allows manipulation of customer information. The database manages the account payable of the customers. This system allows more flexible ways to retrieve item information, for example users can either enter the item's code or item's attribute to retrieve related information.

2.2.2 Case Study 2 – Idaho Computer Services, Inc's Inventory Control System

The Idaho Computer Services, Inc's Inventory Control System (JICON/3000) is a menu driven application. It is a real time system with point of sale inventory increases and reductions. Transactions are stored in a daily transaction data set. A complete transaction history is kept for all the inventory items. This allows for month and year-end reporting and also aid in the reordering and management of inventory. ICON/3000 is divided into separate locations. This allows for processing by the individual location. Each location is broken down further by item numbers. An item number is assigned to each item in the locations inventory. There are no restrictions on the number of locations or the number of items a location can have. All of the different options of a location can be set up in a monitor data set. This data set is set up when the system is installed and will not change until the location decides to change how the inventory is to be processed. Inventory Control maintains an inventory item master data set, which is established during set up, after the monitor data set is established. This item master data set contains one record for each item in the inventory. After the initial installation, user may make changes to the existing inventory to keep the item master data set current and efficient. History can be maintained for as many months of transaction as needed for the inventory management. The transactions are written from the daily transaction data set to the transaction history data set. These transactions can be used to show every operation, which occurred on an item over given period of time. At the end of a month, a month end program is run to update the month's sales history. Sales history is kept by the month while other information such as lost sales and item returns are kept by year-to-date fields. All

transactions, which apply to work order, are stored in a work order data set. All transaction for a work order is stored until the work order is completed.

Discussion:

One distinguish feature of this system is it is easy to understand and ease to use. ICON/3000 emphasizes on implicit part of the application. Inventory, transactions and order are kept in different data set (different records in database) that is more visible to users and systematic. Different data set also enable quick access and modification of data. However, there is no record of who is in chare of particular transaction. This is very important in order to identify the people who are responsible when an error is occurred.

2.2.3 Case Study 3 – Chemical Inventory System Pro (CISPRO 2000)

The Chemical Inventory System Pro (CISPRO 2000) is a high performance database system designed specifically for tracking chemicals and laboratory supplies. CISPRO 2000 can be used in a single user, networked or client/server environments. The system stores a variety of essential information about each chemical and supply item. CISPRO 2000 can store a tremendous amount of information about each chemical, but user has total flexibility as to how much of the information user use. CISPRO 2000 provides user define fields for reference ID's, and checkboxes for classifications. The physical and hazard tabs store a wealth of useful properties for each chemical. CISPRO 2000 implements a container-based tracking model, so user can have multiple lots and

containers of any chemical, supply, or formulation. CISPRO 2000 will automatically generate unique ID's for user, or user can assign by his own. These ID's can also be used in conjunction with the RIBM for remote inventory control purposes using a portable data terminal and bar code scanner. The Qty/Locations tab is where user keeps track of all containers for a specific chemical. User can have an unlimited number of containers, sizes, and vendors defined for each chemical. CISPRO 2000 can track all purchasing and receiving information for any given department or the entire facility. User can build requisition lists, as well as, purchase orders right from within the CISPRO 2000. CISPRO 2000 simplifies ordering and requisitioning for items in inventory by allowing user to place orders, view orders pending, orders received, or partial shipments with order items still pending. CISPRO 2000 now allows user to define Formulations, which have an Ingredient List. Once the ingredients have been specified, CISPRO 2000 can produce scaled batch records (these can also be used for material requirement reports). CISPRO 2000 can even adapt to user's needs with user-defined properties. If user needs a new field for some specific information, user can define a new property. Property types can be: text, number, date, or list of items. CISPRO 2000 has an improved searching and sorting interface. Everything user need is right there in front of him at all times. CISPRO 2000 provides an optional Transaction Logging mechanism, which allows facilities to monitor, or removal of an item. Item, date, location, user, and reason can be easily logged. The system can produce reports and warnings when an item is below a user-specified level. Using transactions, CISPRO 2000 can provide cost-accounting features such as removal and addition costs per account, and value of inventory reports. The user can maintain multiple databases, and import data from various data systems. An extensive array of reports is included for listing chemicals and

supplies, including reorder reports, hazard information reports and location reports. CISPRO 2000 now with a fully integrated ad-hoc custom report designer enables the user to modify any report, or even create totally new reports, right from within CISPRO 2000.

Discussion:

One distinguish feature of this system is it is easy to understand and ease to use. CISPRO 2000 has a well design interfaces that enable the user to see and get what they want fast and easily. Besides this, CISPRO 2000 can maintain a large volume of data in the database and enable the user to retrieve the information they need from the database easily. In addition, the system also make sure the system is secure and reliable by asking the user's ID before letting the user to sign in the system and get the information from the system. CISPRO 2000 also provides a lot of functions that will help the user in updating the system information.

2.3 System Architecture

There are a few software architectures available now: mainframe architecture, client-server architecture, two-tier architecture and three-tier architecture.

2.3.1 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 info to the host. Mainframe architecture is not tied to a hardware platform. User interaction can be cloning 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 graphically dispersed sites.

2.3.2 Client-Server Architecture

Client

Client is a networked information requester, usually a PC or workstation that can query database and/or other information from a server. Clients rely on servers for resources, such as files, devices, and even processing power.

Server

Server is a computer, usually a high-powered workstation, a minicomputer, or a mainframe, that houses information for manipulation by networked clients. Server is

dedicated to managing disk drives (file servers), database (database servers), printers (print servers), or network traffic (network servers).

Client-server

Client-server is 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 requests and returns the results to the client. The client manipulates the data and presents the result to the user.

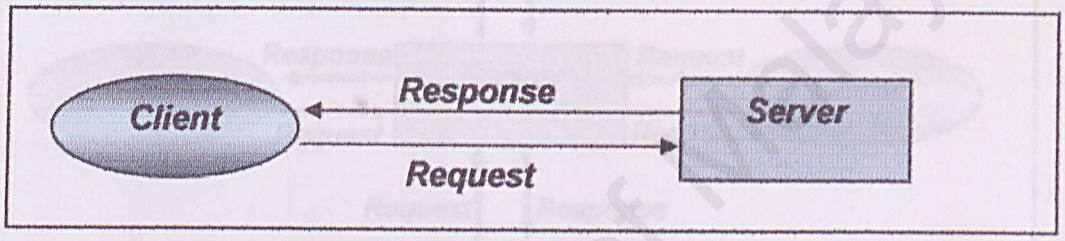


Figure 2-1: One-to-One Client Server

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

2.3.3 2-Tier Architecture

2-tier architecture refers to client/server architectures in which the user interface runs on the client and the database is stored on the server. The actual application logic can run on either the client or the server. There are only the architecturally tiered data server and client.

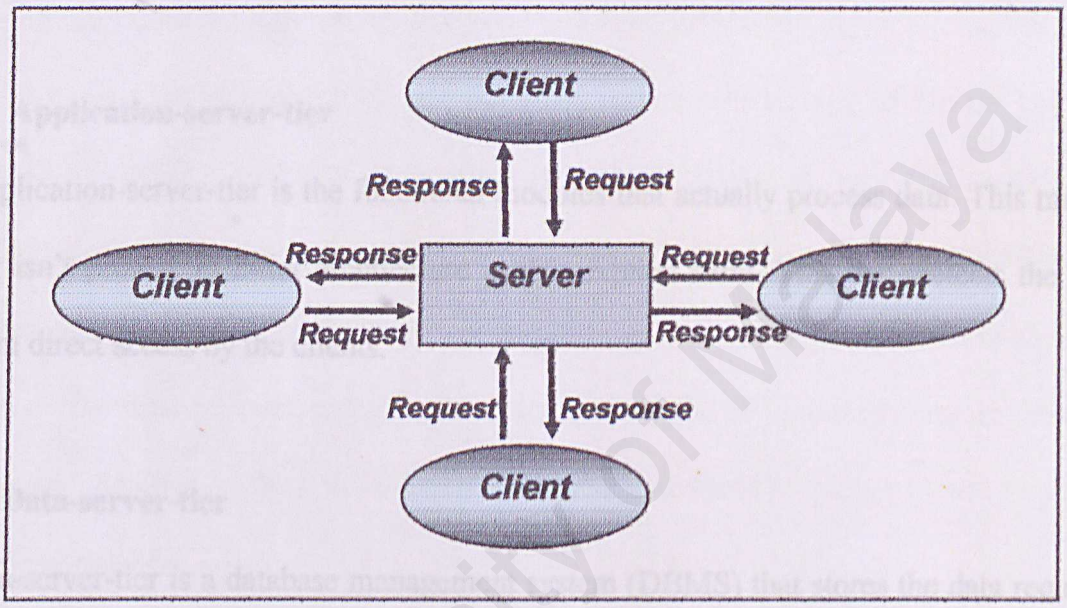


Figure 2-2: Many-to-One Client Server

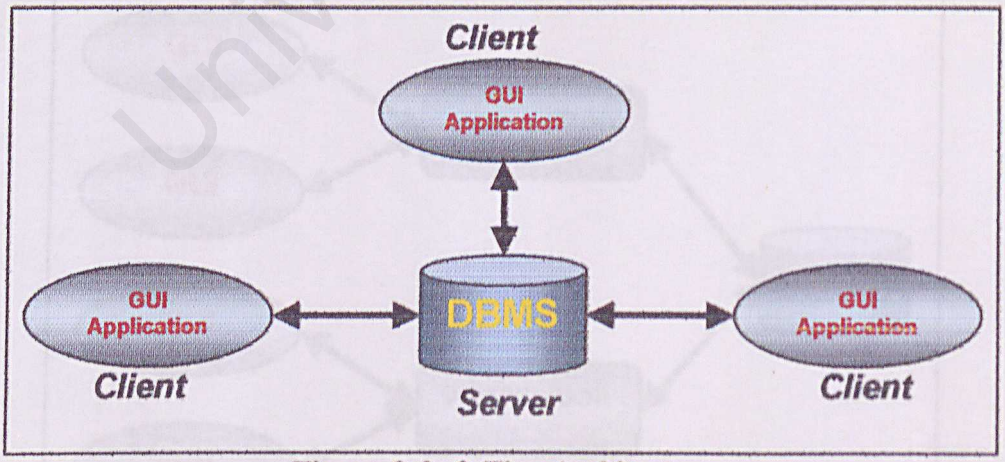


Figure 2-3: 2-Tier Architecture

2.3.4 3-Tier Architecture

Three-tier architecture is a special type of client/server architecture consisting of three well-defined and separate processes, each running on a different platform. The three tiers consist of:

1) Client-tier

Client-tier is the user interface, which runs on the user's computer.

2) Application-server-tier

Application-server-tier is the functional modules that actually process data. This middle tier isn't present in 2-tier architecture in this explicit form. This tier protects the data from direct access by the clients.

3) Data-server-tier

Data-server-tier is a database management system (DBMS) that stores the data required by the middle tier.

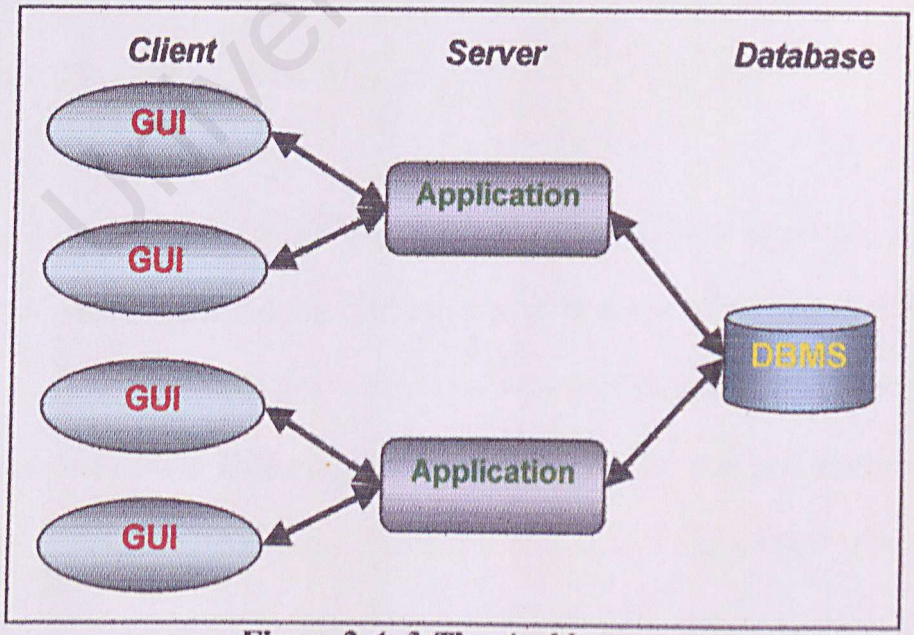


Figure 2-4: 3-Tier Architecture

2.3.5 Conclusion for Software Architecture

Client-server architecture is chosen for this project because it is easier to implement and design. The Client-server architecture has many advantages. The primary advantage of Client-server architecture arise from splitting the processing between the client system and the database server since the bulk of the database processing is done on the back-end , the speed of the DBMS is not tied to the speed of the workstation. As a result, the workstation need only be run the front-end software, effectively extending the life many order or smaller PCs which do not have the horsepower needed to run on complex DBMS. Another benefit of separating the client from the server is workstation independence; users are not limited to one type of system or platform. Besides that, Client/server is the preservation of data integrity. This division of work also reduces the load on the network connecting the workstation. Instead of sending the entire database file back and forth on the wire, the network traffic is reduced to queries to and responses from the database server.

2.4 Operating System

Operating system (OS) is a platform that performs basic tasks, such as recognizing input from the keyboard, sending output to the display screen, keeping track of files and directories on the disk, and controlling peripheral devices such as disk drives and printers.

Besides that, the OS makes sure that different programs and users running at the same time do not interfere with each other. For security, OS ensures that unauthorized users do not access the system. OS provides a software platform to allow application programs run on it.

2.4.1 Microsoft Windows 98 Server

Windows 98 is based on the popular Microsoft Windows 95 Operating System, and is designed for the consumer market. Windows 95/98 was designed for backward compatibility with older DOS and 16bit programs, as well as providing a platform for the newer (back in 1995) 32 bit programs.

Windows 98 works better by making it simple to access the Internet and by providing better system performance along with easier system diagnostics and maintenance. With Windows 98, users' system plays better as well with support for the latest graphics, sound, and multimedia technologies, the ability to easily add and remove peripheral devices with support for Universal Serial Bus (USB), and it also enables users to watch

TV on PC. Besides that, Windows 98 is compatible with more software (including games) and hardware.

2.4.2 Microsoft Windows 2000 Server

Windows 2000 is a multipurpose, entry-level server operating system Windows 2000 that can use to provide the network users with file, print, application or web services. The Windows 2000 Server operating system is designed to increase the value of the existing investments while lowering overall computing costs. Specifically, Windows 2000 Server is easier to develop, configure and use because it provides centralized, customizable management services to reduce TCO (Total Cost of Ownership). Moreover, these management services work with existing management solutions and mixed-platform distributed networks.

Some of the notable features in Windows 2000 Server are:

- A fully customizable administrative console that can be based on tasks rather than files, applications or users.
- A new file directory approach called Active Directory that lets the administrator and other users view every file and application in the network from a single-point-of-view.
- Dynamic Domain Name Server (DNS), which replicates changes in the network using the Active Directory Services, the Dynamic Host Configuration Protocol and the Windows Internet Naming Service whenever a client is reconfigured.

- The ability to create, extend or mirror a disk volume without having to shut the system and to back up data to a variety of magnetic and optical storage media.
- A Distributed File System (DFS) that lets user see a distributed set of files in a single file structure across departments, divisions or an entire enterprise.
- Close integration with and support for Microsoft's Message Queue Server, Microsoft Transaction and Internet Information Server (IIS).
- Configure Your Server Wizard reduces the time it takes to built a server and reduces the likelihood of error.
- Windows Script Host includes script for a number of commonly used administrative functions such as logon scripting.
- Plug and Play network adapters that helps reducing device configuring time.
- Connection Sharing Wizards provide an out-of-the –box network address management solution for small businesses.
- Comprehensive, standard based security services, including flexible authentication, data encryption, flexible and secure network access, protection of virtual private network (VPN) and security extensions for the development platform.

2.4.3 Microsoft Windows NT Server

Microsoft Windows NT Server is the only true multipurpose server operating system. It combines the case-of-use of Windows 95 with the power and reliability of Windows NT. Here are some of the advantages:

- It is easy-of-use interface that helps to work easier and faster.

- It is easy to manage and control as it includes remote management and troubleshooting tools and allow administrators to implement policies and standards for system-wide desktop configuration.
- It is productivity and compatibility as Windows NT ensures high performance for 32-bit programs. All Win16 Windows-based programs have the preemptive multitasking capabilities of Windows NT and can be run in a separate address space for better responsiveness and reliability.
- It meets the reliability standards to run critical line-of-business programs. It allocates separate memory space for 16-bits application, so if one 16-bit application fails it won't bring down other application. It also protects critical operating code, device drivers and data from application.
- It supports workgroup and networking. Windows has the built-in file sharing and print-sharing capabilities for workgroup computing. It also has open network system interface that is compatible with Banyan Vines, NetWare, Novell, UNIX and Macintosh as well as Microsoft Windows for workgroup, Windows 95 and standard x86 environment up to simultaneous connections can made to a Windows NT computer for sharing files and printers.
- It allows Object Linking and Embedding (OLE). In other words, it can combine information from several applications into one compound document using the special object linking and embedded capabilities of Windows-based application.
- It has built-in tools for internetworking and intranetworking like TCP/IP, Microsoft Explorer and Microsoft Peer Web Services.
- It enables the capabilities of integrating application on a single computer or even across multiple computers by using COM and DCOM.

2.4.4 Microsoft Windows 2000 Server versus Microsoft Windows NT Server

Windows 2000 Server is a newer version than Windows NT Server 4.0, surely it will has more advantages than Windows NT Server. Below are some of the differences:

- Windows 2000 was twice as fast as Windows NT 4.0.
- In Windows 2000, trust relationships are developed within domains.
- Able to easily delegate authority to change user accounts spread across several Windows 2000 Organizational Units.
- With Windows 2000, users are able to transparently access Microsoft Office application such as Word and Excel.

2.5 Database Server

A database is a structured collection of data. To add, access, and process data stored in a computer database, a database server is needed. There are several database servers available currently: Oracle, MySQL and Microsoft SQL Server 7.0, Microsoft SQL Server 2000 and Microsoft Access 2000.

2.5.1 Oracle

Oracle is a multi-user database. It provides unprecedented ease-of-user and is pre-tuned and pre-configured for today's dynamic workgroup and line-of-bus environment.

Oracle includes a fully integrated set of easy-to-use management tools, full distribution, replication and web features. Oracle also provides the highest levels of availability through fast failover, easier management, and zero data loss disaster protection, with Data Guard, the only complete data protection solution available on the market.

Oracle can runs on UNIX, Linux and Windows platform. However, it is expensive and separate licenses are required for each of its database engine.

2.5.2 MySQL

MySQL is a relational database management system. MySQL stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and

flexibility. The tables are linked by defined relations 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. It is client/server implementation that consists of a server and many different client programs. It is available on a variety of UNIX platforms, Linux, Windows NT, Windows 95/98 and Windows 2000.

MySQL is Open Source Software. 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. Anybody can study the source code and change it to fit their needs.

2.5.3 Microsoft SQL Server 7.0

Microsoft SQL Server 7.0 is a client/server based relational database management system. It is an ideal database engine for web because of its ability to queried and update via popular web browsers such as Internet Explorer. Microsoft SQL Server 7.0 is a significant tool in many regards from data warehousing to application that require not only a large amount of information but also many different simultaneous users, Microsoft SQL Server 7.0 is a key component data management requirement.

Microsoft SQL Server 7.0 is a perfect example of an n-tier system. The user can manipulate the data directly from the client-side. It also maintains referential integrity and security and ensures that operation can be recovered in the event of numerous types

of failure. Microsoft SQL Server 7.0 can control the access for the type of information that can be retrieved by the user.

Microsoft SQL Server 7.0 supports Internet database integration and allows the user to automate the publishing of database information in HTML documents. It allows us to build active websites and let us conduct processes on the Internet. It also provides the function for transparent distributed transaction. This means that it provides automatic distributed update capability across two or more SQL server transparent to the desktop application, making it a simple to use. It guarantees the integrity of transaction of update spanning multiple servers.

Microsoft SQL Server 7.0 makes giant strides in performance, reliability and scalability, giving organizations many opportunities to create intelligent, real-world business solutions. By voicing a need for more simplified and cost-saving features, organizations inspired the following innovations in Microsoft SQL Server 7.0:

- Scalable from laptop to multiprocessor cluster
- Dynamic row-level locking
- Dynamic self-management
- Wide array of replication options
- SQL Server Desktop
- Integrated OLAP services
- Data Transformation services
- Microsoft English Query
- Integrating with Microsoft Office 2000 and Microsoft Visual Studio

2.5.4 Microsoft Access 2000

Microsoft Access 2000 simplifies the skill set needed to create simple, useful database -- the improved interface offers more consistency with other office application, plus new features that increase productivity. Microsoft Access 2000 also adds Data Access Pages -- Web pages that let users interact with data over the web and maintains live links to a database. Microsoft Access 2000, through its support of OLEDB, can act as a front-end to high-end database engines.

Of particular importance, Access 2000 can act as a front-end client to corporate-level, back-end database, such as Microsoft SQL Server. Access can now be used in 2 ways: as a standalone application for creating databases for individual or departmental use or as an easy-to-use interface client to a more scalable and robust back-end database that was previously only available to professional database administrators (DBAs). This lowers the bar for creating true client/server application by allowing end users to take advantage of the ease-of-use of Access combined with the scalability and reliability of Microsoft SQL Server. Regardless of the back-end data source selected, end users will still have the same easy-to-use experience of the most popular desktop database client.

Microsoft Access 2000 makes it easy to get the information we need and provides powerful tools that help organize and share our database so our team and we make better decisions. Some of the features that Access provides as:

- Enable web collaboration and improve productivity

- Seamless integration between our data source and interactive web pages make building and sharing an Access database easier
- Access 2000 allows quick analyze details and see vital relationship
- Access 2000 includes built-in SQL Server integration that brings the power of high-end database to the familiar Access environment

2.5.5 Microsoft SQL Server 2000

SQL Server 2000 provides agility to company data management and analysis, allowing organization to adapt quickly and gracefully to derive competitive advantage in a fast-changing environment. From a data management and analysis perspective, it is critical to turn raw data into business intelligence and take full advantage of the opportunities presented by the Web. A complete database and data analysis package, SQL Server 2000 opens the door to the rapid development of a new generation of enterprise-class business applications that can give your company a critical competitive advantage. The record-holder of important benchmark awards for scalability and speed, SQL Server 2000 is a 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.

Microsoft SQL Server 2000 is a single process, multithreaded relational database server primarily intent for transactional processing. It is based on the client/server architecture, which divides processing into two components: a front-end, or client component, that run on a local workstation and a back-end, or server component, which runs on a remote computer.

As we make comparison between Oracle 9i and SQL Server 2000, a fact that Oracle 9i can support much more complex database compared to SQL Server 2000. Nevertheless, as SQL Server also incorporates a world-class feature set for distributed client-server computing, therefore it is still chosen as database management system as it is strongly believed that customers who use SQL.

2.6 Application programming Language

2.6.1 Microsoft Visual Basic 6.0

The first review is Microsoft Visual Basic 6.0. Its standard module enables production of stand-alone applications. It combines RAD and object-oriented programming.

The new tools are built on ActiveX Data Objects (ADO) 2.0, Microsoft's Standard high-level interface for data access. This has a simple object model, a common user interface for local and remote data access, support for hierarchical record sets and accessible data binding.

ADO components are created during development by dragging the required data elements, such as tables, views and fields from the new Data Environment Designer into screen forms to create data bound objects. The Data Environment Designer is used to define your project's data sources, and the tables, views and stored procedures that are available from each source. These can then be dragged into the appropriate positions on your forms.

Microsoft Visual Basic 6.0 was also an interpreted language system, so users could test and debug application on the fly from within the development environment.

Advantages:

- Component-based development aids and database access model
- Server-side transaction support

- Integration with web clients

Disadvantages:

- Performance remains second rate compared with C++ and other fully compiled language.

2.6.2 Microsoft Visual C++ 6.0

Visual C++ is Microsoft's version of the C++ programming language. Based on the programming language, C++ is an improved version of C that takes the C language to the next level of evolution of programming languages, those that provide object oriented programming.

The Enterprise Edition of Microsoft Visual C++ 6.0 provides many tools and components for building and validating enterprise level distributed Component Object Model (COM) applications. Visual C++ also provides ADO and OLEDB.

Advantages:

- Good performance

Disadvantages:

- Not really a RAD tool, so it is difficult to be used for window programming compares Visual Basic and Delphi.

2.6.3 Delphi

Delphi is Borland's best-selling rapid application development (RAD) product for writing windows applications. With Delphi, windows program can be written more quickly and more easily then was ever possible before. User can create Win 32 console applications or Win 32 graphical user interface (GUI) programs. When creating Win 32 GUI applications with Delphi, user has all the power of a time compiled programming language (Object Pascal) wrapped up in a RAD environment. What this means is that user can create the user interface to a program (the user interface means the menus, dialog boxes, main window, and so on) using drag-and-drop techniques for time rapid application development.

Borland built Delphi around its powerful Object Pascal language. Though Object Pascal's roots are in standard Pascal, it includes true object orientation with support for inheritance, encapsulation and polymorphism. The Pascal derivative in Delphi is sophisticated and powerful, yet easier to comprehend and manage than C++. Like Microsoft Visual Basic, Delphi uses dot notation to reference objects' properties and methods. Unlike Microsoft, however, the Delphi language relies consistently on objects and allows you to create your own classes by sub-classing from other objects and inheriting their properties and methods. This powerful feature, the cornerstone of OOP, gives user significantly more control in creating well-structured applications. It's also an excellent vehicle for code reuse.

Advantages:

- Distributed data set processing
- Full control of Microsoft transaction server interactions
- Speed comparable to most C++ compilers.

Disadvantages:

- Lacks Visual Studio's integrated Multilanguage packing.

2.6.4 ASP (Active server Pages)

ASP is a server-side scripting technology. ASP is indeed HTML page with an .asp extension. ASP allows for HTML and a scripting language such as VBScript, JScript or Perl to be interspersed in a Web page. When a browser requests an ASP page, the Web server generates a page with HTML code and sends it back to the browser.

One of the most important features about ASP is that it allows user to easily access data and put it on a Web page. User can simply display data from an ODBC-compliant database, or use ASP to make decisions about what to display on a Web page. User can then format the results in any way that they please.

Another important ASP feature is the ability to use cookies to store and retrieve information. The Request object has a Cookie collection, and user can use this in data processing.

2.6.5 ASP.NET

ASP.NET is a revolutionary programming framework that enables the rapid development of powerful web applications and services. Part of the emerging Microsoft .NET Platform, it provides the easiest and most scalable way to build, deploy and run distributed web applications that can target any browser or device. ASP.NET applications are a mixture of client side markup and code, and server side programming.

ASP.NET allows programmers and developers to work both VB.NET and C# within the same ASP.NET page. .NET is a milestone for Microsoft; its marks Microsoft entry into "run once, run anywhere" compiler market alongside Java and Ruby. .NET allows the programmer to use any number of .NET-compiler language to create its code and run anywhere through the robust .NET Framework. ASP.NET can take advantage of all that .NET has to offer, including supports for around 20 or more .NET languages from C# to Perl.Net, and the full set of .NET Framework software libraries.

2.6.6 JavaScript

JavaScript is a scripting language developed by Netscape to enable web authors to design interactive sites. JavaScript is different from Java. Although it shares many of the features and structures of the full Java language, it was developed independently. JavaScript can interact with HTML source code to enable web authors to spice up their sites with dynamic content. JavaScript is endorsed by a number of software companies and is an open language that anyone can use without purchasing a license. It is supported

by recent browsers from Netscape and Microsoft, though Internet Explorer supports only a subset, which Microsoft calls Jscript.

2.6.7 Java Server Pages™ (JSP)

Java Server Pages™ (JSP) is a web-scripting technology that can mix static HTML content with server-side scripting to produce dynamic output. By default, JSP uses Java as its scripting language; however, the specification allows other languages to be used, just as ASP can use other languages (such as JavaScript and VBScript). While JSP with Java will be more flexible and robust than scripting platforms based on simpler languages like JavaScript and VBScript.

JSP provides a number of server-side tags that allow developers to perform most dynamic content operations. So developers who are only familiar with scripting, or even those who are simply HTML designers, can use JSP tags for generating simple output. Advanced scripter or Java developers can also use the tags, or they can use the full Java language if they want to perform advanced operations in JSP pages.

CHAPTER 3 : System Requirements Analysis

3.1 Methodology

The system development methodology is a method to create a system with a series of steps or operations or can be defined as system life cycle model. Every system development process model (see Figure 3-1) includes system requirements (user, needs, resource) as input and a finished product as output.

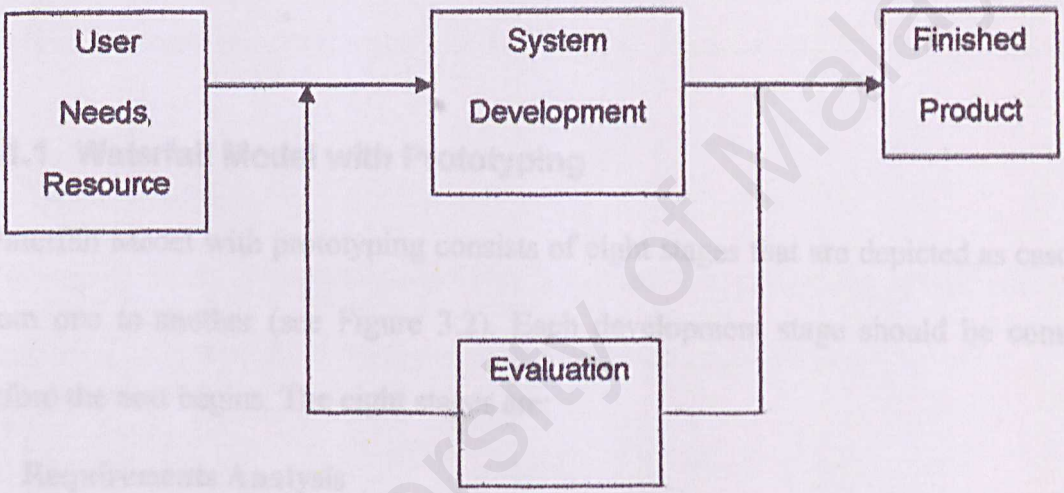


Figure 3-1: System Development Process Model

There are several process models in system development:

- Waterfall Model with prototyping
- V Model
- System Development Life Cycle (SDLC)
- Spiral Model

Waterfall Model with prototyping is chosen because:

- This model is useful in helping me to layout what I need to do
- It enables maintenance to be carried out at each stage due to its interactive nature. Changes can be done during any of the stages by returning to the previous stages.
- A good specification to begin with
- Easy to use
- Systematic
- Scope of project well understand
- Project risks have been accessed and are considered to be low

3.1.1 Waterfall Model with Prototyping

Waterfall Model with prototyping consists of eight stages that are depicted as cascading from one to another (see Figure 3.2). Each development stage should be completed before the next begins. The eight stages are:

1) Requirements Analysis

Understanding and determining users need by having brainstorming, eliciting and analyzing user requirements by having interview, survey or questionnaire session, collecting and specifying all the user requirements and validating requirements.

2) System Design

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

Determining and specifying program design and database design and verifying program design.

4) Coding

Involving programming, personal planning, tool acquisition, database development, component level documentation and programming management.

5) Unit and Integration Testing

Test units separately and integrate the tested units. Then, testing on the integrated units.

6) System Testing

Combining all the integrated units into a system. Testing on the system. Specifying, reviewing and updating of the system test and validating of system.

7) Acceptance Testing

Testing on system completed. The system is delivered.

8) Operation and Maintenance

Control and maintain the system. Revalidating of system.

The system has to be validated and verified during the stage of system testing. The verification is to make sure that the function in the Supermarket Inventory System works correctly and to check the quality of the implementation. The validation is to ensure that

Supermarket Inventory System has implemented all the requirements in the specification.

Prototyping is a sub-process and prototype is a partially developed product or a simple simulator of the actual system to examine the proposed system and overview on the functionalities. A prototype of Supermarket Inventory System will be built regarding to the project scope and the analysis of the system before start to build the actual system.

Prototyping is very important because:

- To ensure the system meet the performance goals or constraints.
- To ensure the system are practical and flexible.
- To ensure the system fulfill the users' requirement.
- To have an insight of how the module and sub-modules interact with each other.

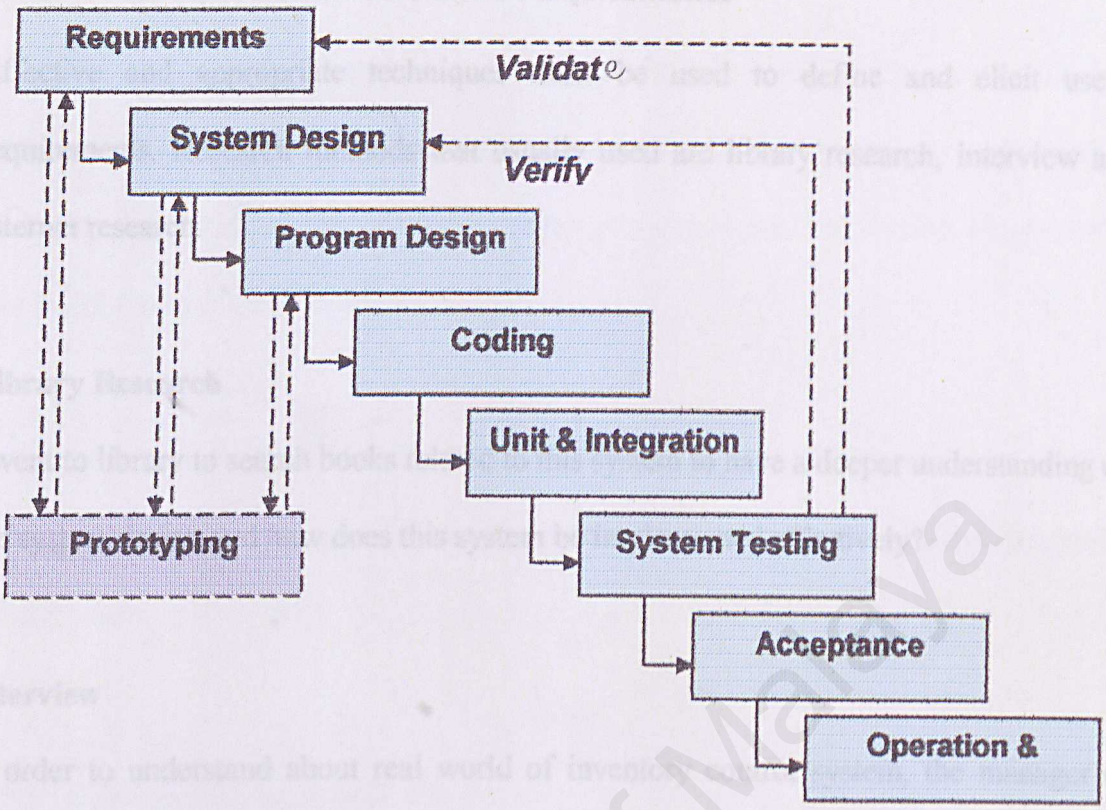


Figure 3-2: Waterfall Model with Prototyping

3.1.2 Techniques Used To Define Requirements

Effective and appropriate techniques must be used to define and elicit users' requirements. Research methods that usually used are library research, interview and internet research.

Library Research

I went to library to search books related to this system to have a deeper understanding on the system design and how does this system be implemented effectively?

Interview

In order to understand about real world of inventory control system, the manager of Gedung Harian Supermarket and KP Minimarket has been interviewed. From the managers, knowledge about inventory control management could be found.

Internet Research

I have surf around the net for sometime to gain deeper understanding about the supermarket inventory system.

3.2 Requirement Analysis

Requirement analysis will enables software developer to specify software function and build models of the data, functional and behavioral for the system analysis and design. A complete understanding of software requirements is essential to develop a system that can fulfill the user's need.

Requirement analysis is done during analyzing system needs. Requirement analysis activities include analyzing and determining functional requirements and non-functional requirements.

3.2.1 Problem recognition

From the analysis that I had done based on the supermarket, there are certain problems with the current manual system, which are:

- Inability to obtain status of inventory rapidly
- Cash transaction take long time and more error prone
- The lengthy time and effort used to managed the current inventory status
- disarrange able of the supplier's detail

3.2.2 Problem Analysis

A computerized inventory control system must have the following function to provide the resolution to the problem that the current manual system:

- Maintain the current status of inventory. Inventory decreases due to the cash transaction and the debit note (damaged stock, short supply or lost item) and increase with the added inventory based on the invoice
- Users' keys in some data and the system will automatically calculate the total amount and change for the transaction.
- Provide search function to retrieve the supplier's detail so that can contact the supplier as soon as possible when there is any shortage of the stock.
- Provide search function to retrieve the information about the latest status of the current stock in the inventory system.
- Automated the paper work done by clerk, for example insert, edit and delete item, insert, edit or delete supplier's detail.

3.2.3 Functional Requirements

Functional requirement is a statement of the service or functions that a system should provide how the system reacts to particular inputs, and how the system should behave in particular situations **[Sommerville, 1998]**. The functional requirements for Supermarket Inventory System are:

1) Add Record or Modify Record

To add new record or information about items or suppliers from the input that is given by the customer or supplier.

Data will be shown by the computer automatically when need to modify information about certain item or supplier by just key in the keywords needed.

2) Delete Record

To delete or terminate an existing record correctly by clerk.

3) Search Record

To search for a particular record based on the input from clerk, supervisor or manager for further operation.

4) Retrieve Record

To get or retrieve a particular record from the database and display the result to the users.

3.2.4 Non-Functional Requirements

Non-functional requirements that are included in the system in order to enhance system performance, and describe the constraints imposed on the system. Non-functional requirements are as important as the functional requirements. This section states the Non-functional requirements for inventory control system for supermarket.

1) User Friendliness

The system should have a user-friendly interface because users may be non-technical personal who would not be able to comprehend complex interface. The system should use the Graphical User Interface (GUI) approach in order to provide better understanding of how to use the system and better communication between the system and users. The interface should use meaningful captions and titles in order to help the users to use the system with more confidence. The system should provide a simple menu so that the user can obtain an overview o the system. The system should display a confirmation message for any non-trivial process such as deletion. The system should provide error handling and validation procedures as another mean to assist users in using the system. The system should also display an error message if an error occurs, such as an invalid data input, invalid password or unsuccessful process of a request.

2) Reliability

Reliability is the extent to which a program can be expected to perform its intended function with required precision **[Pressman, 2001]**. The system should not cause destructive error to user's machine. User's error should be recoverable.

3) Maintainability

System maintenance accounts would require more effort if the system is not designed according to good programming practices. Maintainability is the ease with which a program can be corrected if an error is encountered, adapted if its environment changes, or enhanced if the customer desires a change in requirements **[Pressman, 2001]**.

4) Correctness

A program or system must operate correctly or it provides little value to its users. Correctness is the degree to which the software performs its required function. To ensure this application quality, lots of testing and trial-and-errors will be carried out.

5) Efficiency

Undeniable, efficiency is the main key for implementing the new meetings management system. Efficiency is understood as the ability of a process procedure to be called or accessed unlimitedly to produce similar performance outcomes at an acceptable or credible speed **[Sommerwille, 1995]**.

6) Security

The proposed system has also security measures to minimize the risk of data exposure to unauthorized people.

3.3 Chosen Platform, Database, Application programming Language

3.3.1 Chosen Development Platform

As a web base client-server application, it provides a graphical user interface for users as well as administrator to perform administrative tasks over the Internet. The tools to be used in the development of this system are described in the following section:

For the Supermarket Inventory System, Windows is chosen as the development platform. Microsoft's Windows 2000 is built to work with a series of microprocessors from the Intel Corporation that share the same or similar sets of instructions.

The main reason for choosing Microsoft's Windows 2000 as the development operating system is because most of the computers in FSKTM are currently installed with Windows 2000. Therefore, the implementation of the new system can be done easily and effectively.

Benefits using Microsoft windows 2000 Professional

Reliable: Windows® 2000 Professional is built upon the rock-solid reliability of Windows NT® technology, which makes it significantly more reliable than either Windows 95 or Windows 98 technology. Reliability improvements in Windows 2000 Professional make it even more stable than Windows NT Workstation. Windows 2000 Professional offers high system uptime, dynamic system configuration and resilience to application failures.

Manageable and easy to use: The support for standards-based security in Windows 2000 Professional protects corporate data in stand-alone and networked environments. Windows 2000 Professional offers a set of built-in tools that make it easier to deploy and manage. In addition, Windows 2000 Professional offers an intelligent user interface that adapts to the way users work thereby making the users more efficient.

Internet-enable user business: Windows 2000 Professional is designed to make it easier for organizations to embrace the Internet. The built-in Internet Explorer (IE), a tightly integrated browser, provides users with a faster and richer Internet experience. With support for Dynamic HTML (DHTML) and Extensible Markup Language (XML), it offers a powerful platform for developers to create highly scalable end-to-end e-Commerce and line-of-business web applications.

Share information more efficiently using the Web: In the past, performing standard file operations on a network file share was much easier than performing similar operations on a remote Web site. Now, Windows 2000 Server technologies such as Web Distributed Authoring and Versioning (WebDAV) make it as easy to carry out standard file operations on a Web share.

Create Web-based business applications: Creating Web-based applications that integrate well into traditional business applications can be difficult. Windows 2000 Server overcomes this burden by sharing internet-aware application development tools with IIS, an efficiency that extends applications to the Web and eliminates awkward bridges between internal and external processes.

Bring server operating system functionality to the Web: In addition to allowing organizations to extend basic file and print services to the Web, Windows 2000 Server supports applications, media, and communications and networking services from a common server platform. This convergence means that everything a company can do with Windows 2000 Server is automatically supported in a fully integrated Web environment

Web server (IIS)

IIS is chosen as the web-publishing server its tight integration with windows 2000. IIS is a web server, a collection of software programs designed to service requests for information and other resources from client on the internet, World Wide Web or organizational Intranets. In a broader sense, IIS provides a comprehensive web server and web publishing system designed especially for Microsoft Windows server operating system. Apart from that, it also allows security features to be implemented on specific virtual directories to control access.

3.3.2 Chosen Database Management System

As we make comparison between Oracle 8i and SQL Server 2000, a fact that Oracle 8i can support much more complex database compared to SQL Server 2000. Nevertheless, as SQL Server also incorporates a world-class feature set for distributed client-server computing; therefore it is still chosen as database management system as it is strongly believed that customers who use SQL Server will enjoy benefits in these key areas:

- Reliable distributed data and transactions

- Centralized control of distributed servers
- Very high performance and scalability
- Support for very large databases

Benefits using Microsoft Server SQL 2000

Fully Web-Enabled: SQL Server 2000 provides extensive database programming capabilities built on Web standards. Rich XML and Internet standard support give user the ability to store and retrieve data in XML format easily with built-in stored procedures. Users can also use XML updategrams to insert, update and delete data easily.

Easy access to data through the Web: with SQL Server 2000, user can use HTTP to send queries to the database, perform full-text search on documents stored in database, and run queries over the Web with natural language.

Powerful, flexible Web-based analysis: SQL Server 2000 Analysis Services capabilities are extended to the Internet. Users can access and manipulate cube data by means of a Web browser.

Quick development, debugging, and data transformation: SQL Server 2000 features the ability to interactively tune and debug queries, quickly move and transform data from any source, and define and use functions as if they were built in to Transact-SQL. Users can visually design and code database applications from any Visual Studio tool.

Simplified management and tuning: With SQL Server 2000, it is easy to manage databases centrally alongside all enterprise resources. Stay online while easily moving and copying databases

Internet, Intranet and Commerce: SQL Server is very important in Internet, Intranet and electronic commerce strategy due to its cutting-edge features and seamless integration with Microsoft Windows NT and Microsoft BackOffice. Furthermore, the Web Assistant enhances the SQL Server Assistant to easily generate HTML and WML files from SQL Server data. It also supports Internet database integration and allows the users to automate the publishing of database information in the HTML documents, built active web sites and conduct the processes on the Internet.

Ease of Use: Features provided by SQL Server ensure easy-to-use for database administrators in building, managing and deploying business applications. For example, the Dynamic Self-Management automates many routine tasks. Besides, the profiling and tuning tools also help to simplify the process of finding the process of finding and fixing database problems by capturing and replaying server activity.

3.3.3 Chosen Application programming Language

ASP.NET has been selected as the web development tool for the proposed system. The reasons of choosing ASP.NET are as follows:

Enhanced Performance: ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.

World-Class Tool Support: The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.

Power and Flexibility: Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform is available to Web application developers. The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Web. ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages. Further, common language runtime interoperability guarantees that your existing investment in COM-based development is preserved when migrating to ASP.NET.

Simplicity: ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.

Manageability: ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Web applications. Because configuration information is stored as plain text, new settings may be applied without the aid of local administration tools. This "zero local administration" philosophy extends to deploying ASP.NET Framework applications as well. An ASP.NET Framework application is deployed to a server simply by copying the necessary files to the server. No server restart is required, even to deploy or replace running compiled code.

Scalability and Availability: ASP.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the ASP.NET runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps keep your application constantly available to handle requests.

CHAPTER 4 : System Design

4.1 Introduction

System Design is a phase of the waterfall that the entire requirements for the system are translated into system characteristics. The requirements for system are regarding to the analysis that had been discussed in the previous chapter. System design includes the following issues:

- Structure Analysis
- Structure Design
- Database Design

4.2 Structure Analysis

Once the problem had been identified and analyzed, the first technical activity that is performed as part of a software engineering begins with a series of model building tasks. Structured analysis is a classical modeling method used to construct several models of the system. By using this method, models that depict information content and flow are created. The models concentrate on describing what the system does that implies actually what the user requires. These models include Entity Relationship Diagram (ERD) and Data Flow Diagram (DFD).

4.2.1 Entity Relationship Diagram (ERD)

Data modeling describes the data objects to be processed by the system, the relationship between each object and other objects and the relationship between the objects and the processes that transform them. The Entity Relationship Diagram (ERD) is the notation used to conduct data modeling activity. ERD enables a software engineer to fully specify the data objects that are input and output from a system, the attributes that define the properties of these objects and their relationships. Please refer to Figure 4.8.

4.2.2 Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) enables the software engineer to develop models of the information domain and functional domain at the same time. As the DFD is refined into greater levels of detail, the analyst performs an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of data as it moves through the processes that embody the application. A DFD is a graphical technique that describes information flow and the transform that are applied as data move from input to output.

A level 0 DFD, also known as context diagram, represent the entire system elements as a single bubble with input and output data indicated by incoming and outgoing arrows and the external entity as a producer or consumer of information. The act of going from a single system to four component processes is called functional decomposition. Functional decomposition is an iterative process of breaking the description or

perspective of a system down into finer and finer detail. This process creates a set of hierarchically related charts in which one process on a given chart is explained in greater detail on other charts. Please refer to Figure 4.1 to Figure 4.7.

4.2.3 Narrative of Supermarket Inventory System

In the Supermarket Inventory System, there are four types of users, which are cashier, clerk, supervisor and manager.

For cashier, she will only need to do the daily transaction by just key in the needed item's information into the sold out page provided by the system. Then the system will automatically update the system's database by deleting the quantity of the sold out quantity and also record the cash transaction into the database.

For clerk, she will need to insert the information of the new item such as item's code, item's name, item's price, item's quantity and the supplier. If the item is an existing item, the clerk will need to refer to the hand out and check for the item's code, and then insert the item's information into the system's database. The information for the items is from the invoice received by the company. Besides this, when there are any damaged stock, return stock or shortage of stock (not the same as stated in the invoice), the clerk will need to delete the item's quantity from the database. In addition, the clerk also needs to insert or update the supplier's information such as name, address, phone number or fax number into the system. Then the clerk also can retrieve the supplier's information from the system when it's needed.

For the supervisor, he will be able to retrieve latest quantity of the balanced stock from the system only.

For the manager, he will be able to retrieve the information from the system, which is the balanced stock in the company and also the daily sales transaction. He also can get the supplier's information from the system.

For every users of the system, they will need to key in the username and password before they can enter the system.

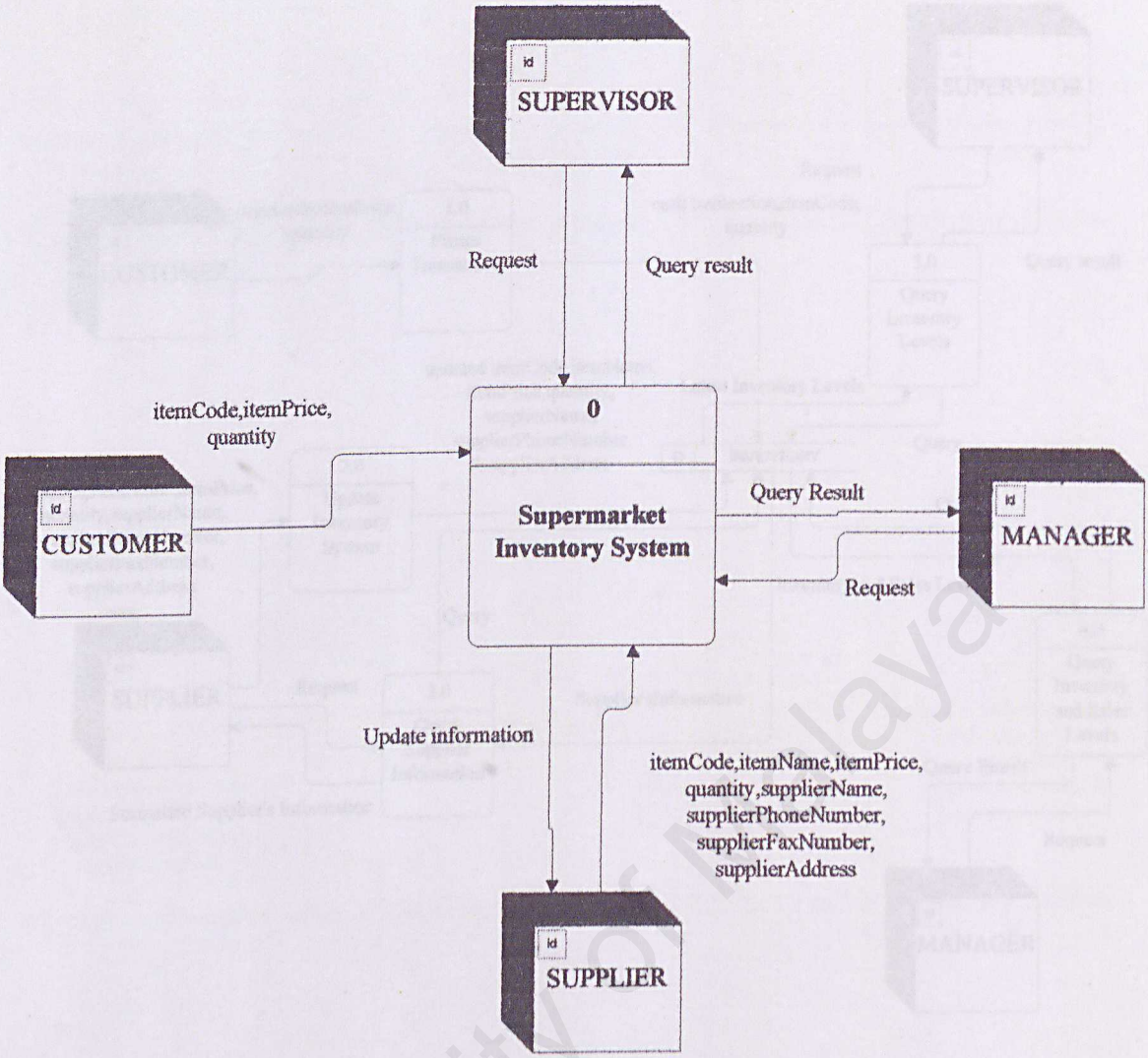


Figure 4-1: Context Diagram of Supermarket Inventory System

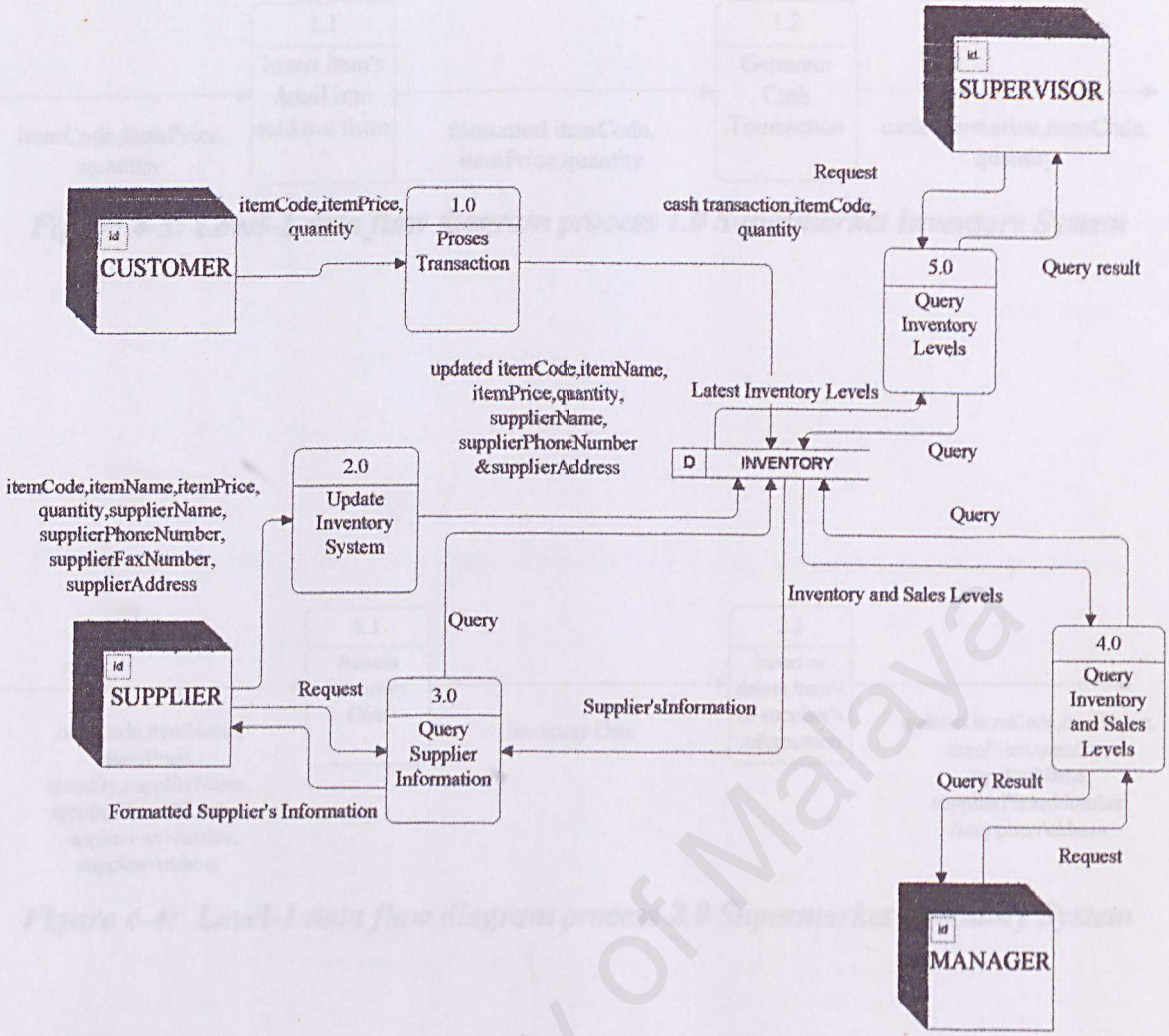


Figure 4-2: Level-0 data flow diagram for Supermarket Inventory System

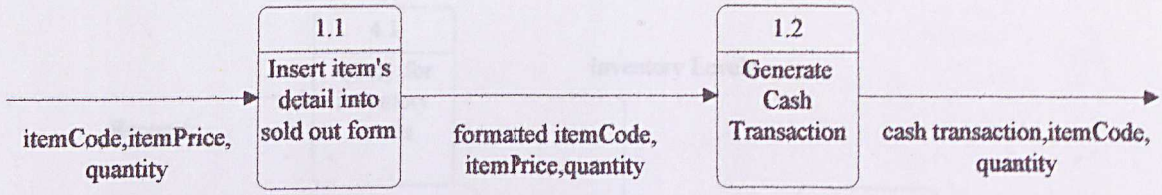


Figure 4-3: Level-1 data flow diagram process 1.0 Supermarket Inventory System

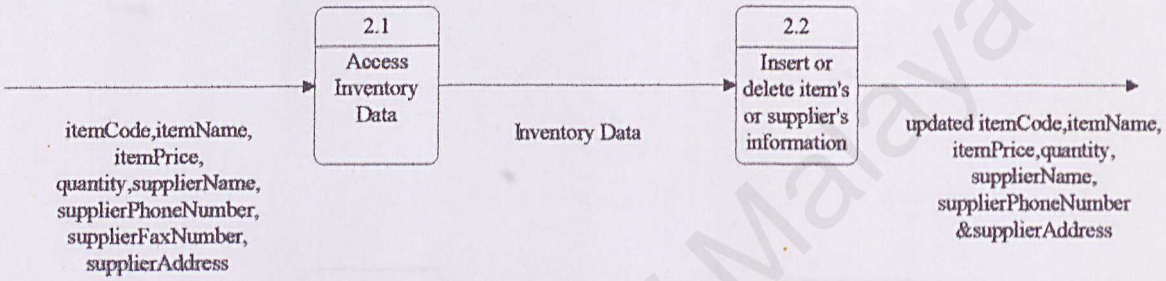


Figure 4-4: Level-1 data flow diagram process 2.0 Supermarket Inventory System

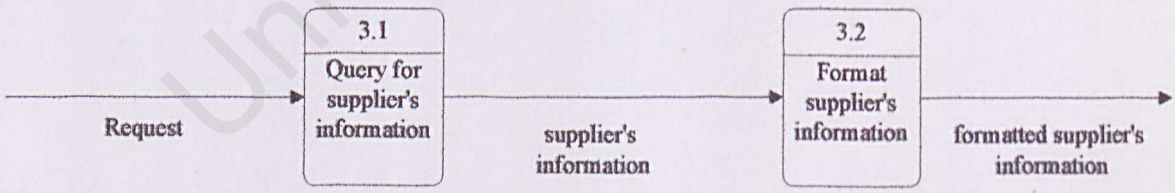


Figure 4-5: Level-1 data flow diagram process 3.0 Supermarket Inventory System

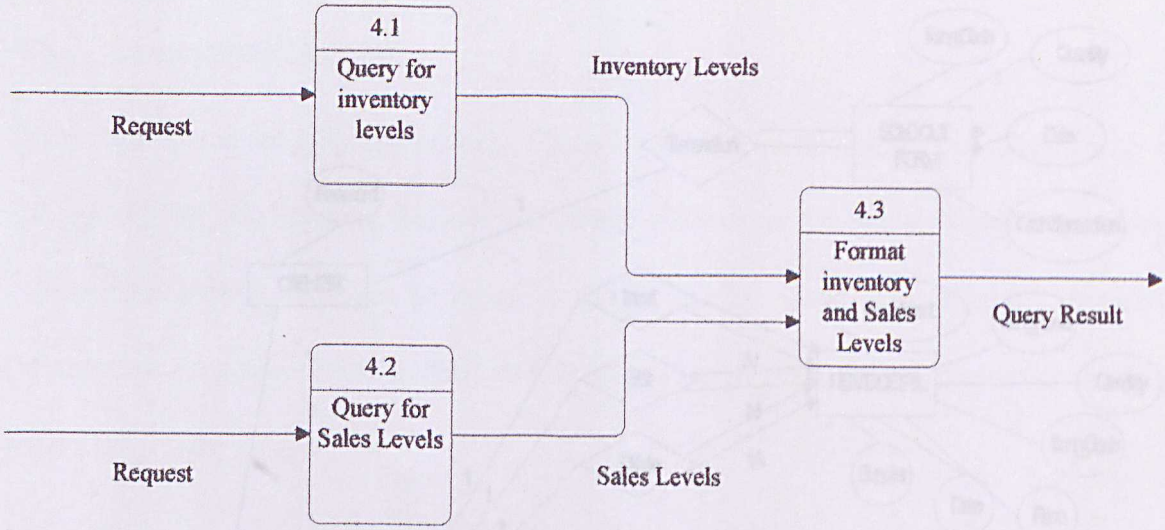


Figure 4-6: Level-1 data flow diagram process 4.0 Supermarket Inventory System

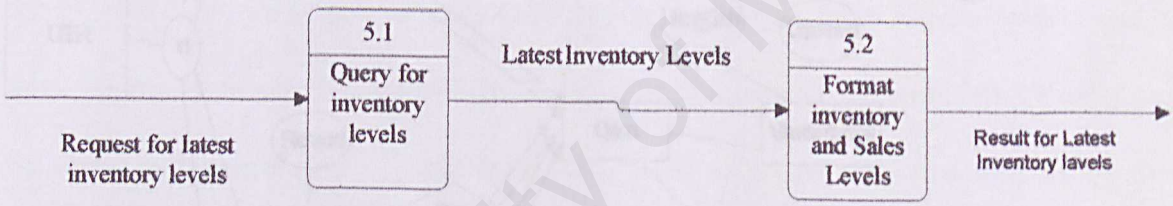


Figure 4-7: Level-1 data flow diagram process 5.0 Supermarket Inventory System

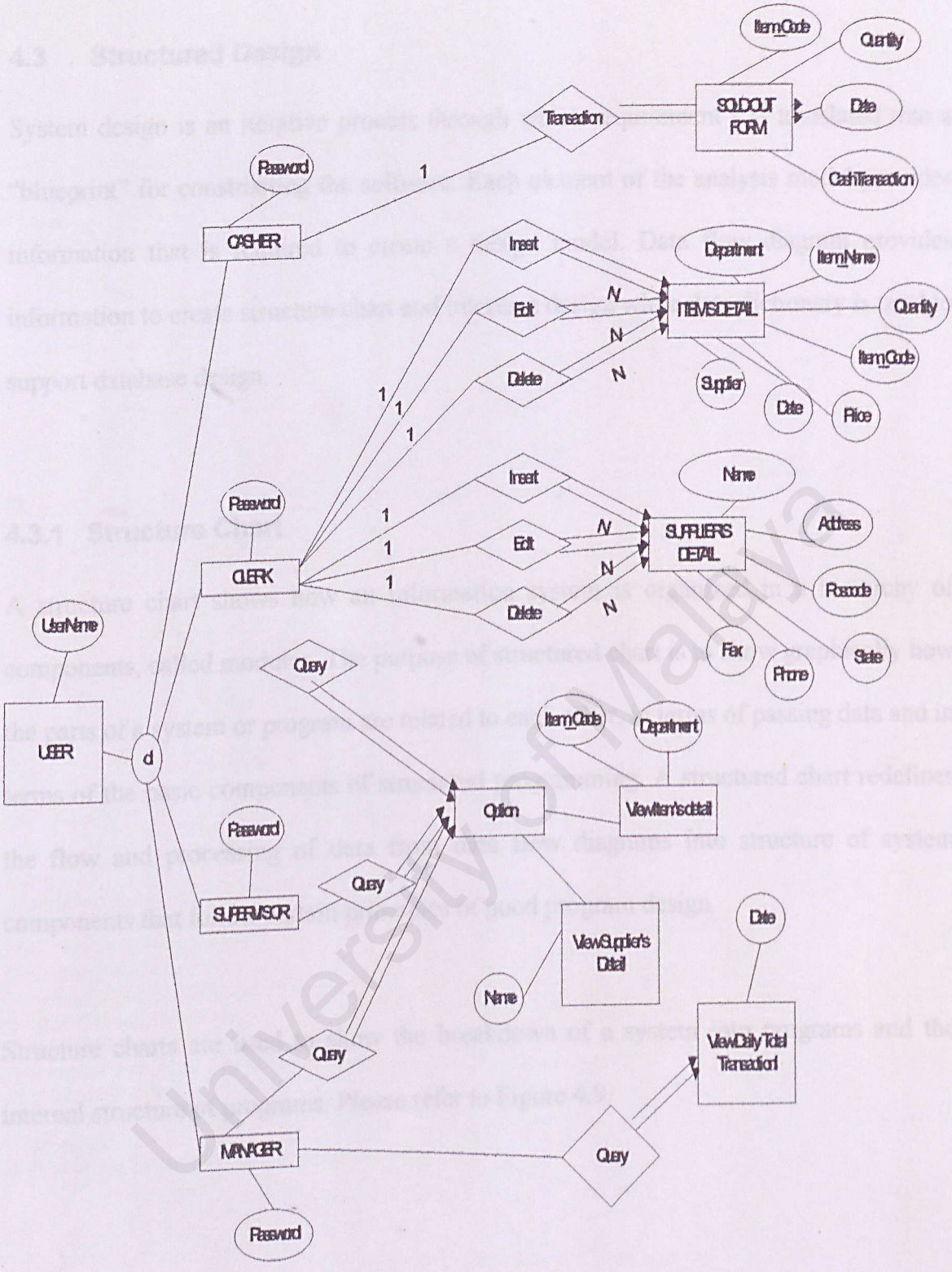


Figure 4-8: Entity Relationship Diagram (ERD)

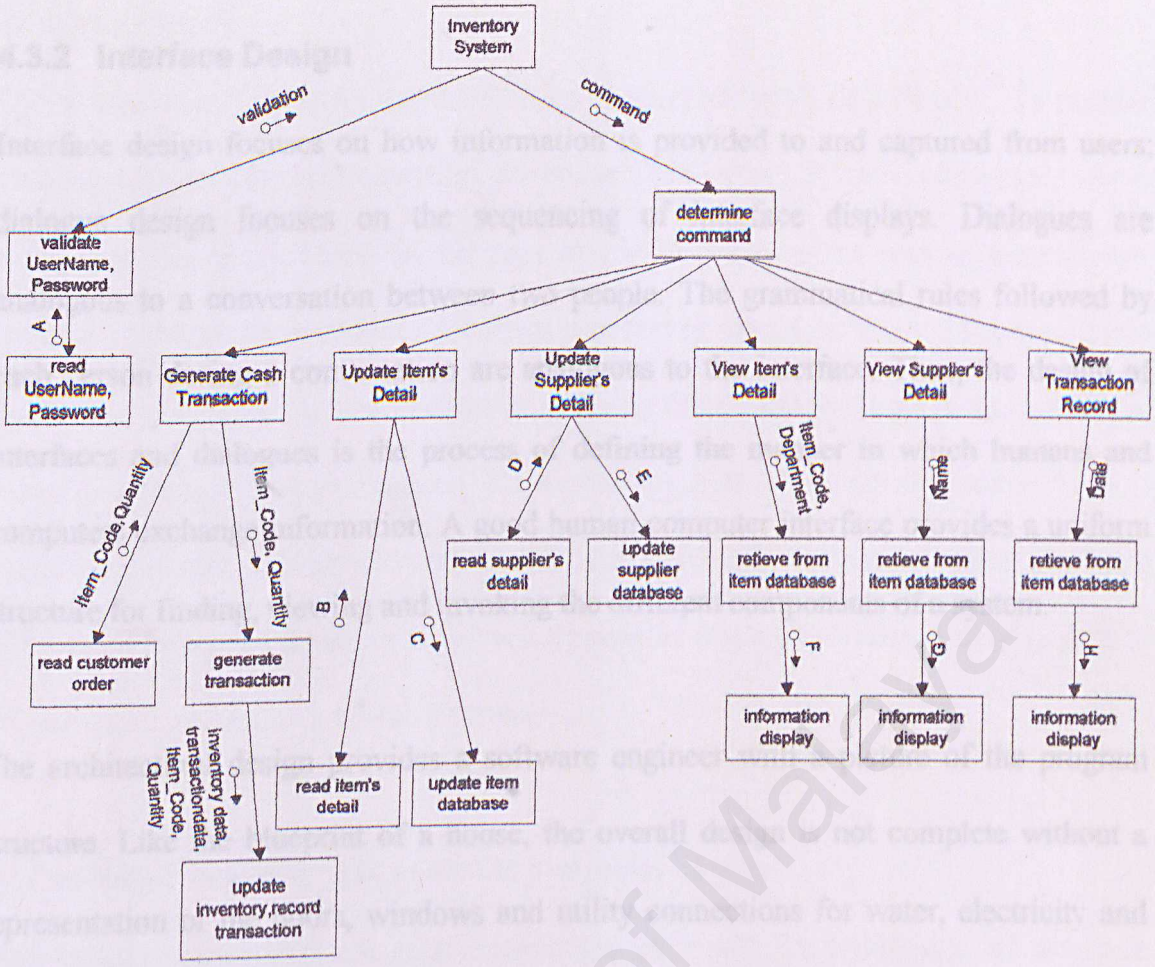
4.3 Structured Design

System design is an iterative process through which requirement s is translated into a "blueprint" for constructing the software. Each element of the analysis model provides information that is required to create a design model. Data flow diagram provides information to create structure chart and interface design while data dictionary is used to support database design.

4.3.1 Structure Chart

A structure chart shows how an information system is organized in a hierarchy of components, called modules. The purpose of structured chart is to show graphically how the parts of a system or program are related to each other, in terms of passing data and in terms of the basic components of structured programming. A structured chart redefines the flow and processing of data from data flow diagrams into structure of system components that follow certain principles of good program design.

Structure charts are used to show the breakdown of a system into programs and the internal structure of programs. Please refer to Figure 4.9.



Legend
A=username,password
B=Department,Item_Code,Item_Name,
Price,Quantity,Date,Supplier
C=formatted item's data
D=Name,Address,Poscode,State,Phon e,Fax
E=Formatted supplier's data
F=Data from database
(department,Item_Code,Item_Name,WQuantity,
Date,Price,Supplier)
G=Data from database (Name,Address, Poscode,State,
Phone,Fax)
H=Data from database (daily total transaction)

Figure 4-9: Structure Chart

4.3.2 Interface Design

Interface design focuses on how information is provided to and captured from users; dialogue design focuses on the sequencing of interface displays. Dialogues are analogous to a conversation between two people. The grammatical rules followed by each person during a conversation are analogous to the interface. Thus, the design of interfaces and dialogues is the process of defining the manner in which humans and computers exchange information. A good human-computer interface provides a uniform structure for finding, viewing and invoking the different components of a system.

The architectural design provides a software engineer with a picture of the program structure. Like the blueprint of a house, the overall design is not complete without a representation of the doors, windows and utility connections for water, electricity and telephone. The Doors, Windows and Utility connections for computer software comprise the interface design of a system.

Interface design focuses on three of concerns: the design of interfaces between software modules; the design interface between the software and other non-human producers and consumers of information (external entities); the design of the interface between a human (the user) and the computer.

In order to design and develop friendly and efficient interfaces, there are few Human Computer Interaction guidelines to be considered:

1) Be consistent

Use a consistent format for menu selection, command input, data display. To provide consistent format for menu selection, Supermarket Inventory System is designed to have entry to access menu based on the type of user. Each time the user of Supermarket Inventory System wants to quit the current function or screen, he/she just needs to clicks the "Back to Options Screen" button to return to the option menu page and select the menu to switch to another function. Textbox is used for data input meanwhile label is used to display the information that cannot be modifying by the user. Another graphical interface used in Supermarket Inventory System to display the data is data grid. It is primary used to display database information.

2) Use simple action verb to name the command button

Meaningful command name is easy to recognize and recall. For example, EDIT is to edit the current information and DELETE is to delete the current information.

3) Provide meaningful error message

Supermarket Inventory System will display message box that produce simple but meaningful error message. For example, "Please fill in username!" if the user do not key in the username field. "Invalid type of password!" is display when the password is invalid type.

4) Minimize the number of input actions required of the user

Instead of require the user to input the data; Supermarket Inventory System will display certain information after the user key in certain data. For example, in the cash

transaction page, when the user key in the code of the item sold, the name of the item will be shown and after the quantity of the item is being key in, the total price of the item will be display.

5) Use the right color for attractive appearance

Color on Web documents adds interest and holds a visitors' attention. In order to make sure that the colors are being used well and not overused, developers always have a purpose as to why part your document should be different color so they will be more noticeable. Using color to grab or hold attention is a valid use in web documents.

4.4 Database Design

By using the database approach, it has replaced the traditional file keeping of information. By using the database approach, data involve in a system will be more easily managed. But before any database approach can be used, the type of data that will involve will be identified first.

The list below is the list of entity and the attribute that involve with the database that will be developed for Supermarket Inventory System.

Entity	Attribute
Items	Department, ItemID, ItemName, Quantity, UnitPrice, IDate, Supplier
TblTransaction	TranDate, Total_Daily_Transaction
TblUserLog	LoginId, Password, Type, Name, Address, Poscode, State, Phone
TblNews	Date, NewsTitle, Content, UserName
TblSupplier	SupplierId, CompanyName, ContantName, Address, Poscode, State, Phone, Fax
Carts	CRecordNo, CartID, ItemID, Quantity
Counters	Type, Counter

Table 4-1: List of entity and attribute

4.4.1 Data Dictionary

The analysis model encompasses representations of data objects, function, and control. In each representation data objects and/or control items play a role. Therefore, it is necessary to provide an organized approach for representing the characteristics of each data object and control item. This is accomplished with the data dictionary.

Data dictionary is the repository of all data definitions for all organizational applications. The data dictionary is an organized listing of all data elements that are pertinent to the system, with precise, rigorous definitions so that both user and system analyst will have a common understanding of inputs, outputs, components of stored and intermediate calculations. [YOU89]

Data dictionary for Supermarket Inventory System is as below:

Data	Attribute	Field Name	Type
Items	Department	Department	Char
	Item's code	ItemID	Char
	Item's name	ItemName	Varchar
	Quantity	Quantity	Int
	Item's price	UnitPrice	Money
	Date	IDate	Date
	Supplier's name	Supplier	Varchar
TblSupplier	Supplier's Id	SupplierId	Char
	Company's name	CompanyName	Varchar
	Supplier's name	ContactName	Varchar
	Supplier's address	Address	Varchar
	Postcode	Postcode	Char
	State	State	Varchar
	Supplier's phone	Phone	Varchar
	Supplier's fax	Fax	Varchar
TblTransaction	Transaction's Date	TransDate	Datetime
	Cash transaction	Total_Daily_Transaction	Money
TblUserLog	User's login Id	LoginId	Char
	User's password	Password	Varchar
	User's type	Type	Varchar

	User's name	Name	Varchar
	User's address	Address	Varchar
	Poscode	Poscode	Char
	State	State	Varchar
	Phone	Phone	Varchar
TblNews	Date	Date	DateTime
	News's title	NewsTitle	Varchar
	News's content	Content	Varchar
	Write by	UserName	Varchar
Carts	Record's number	CRecordNo	Int
	Cart's Id	CartID	Char
	Item's Id	ItemID	Char
	Quantity	Quantity	Int
Counters	Type	Type	Char
	Counter	Counter	Int

Table 4-2: Data dictionary

CHAPTER 5 : System Implementation

5.1 Introduction

In the implementation phase of the system development life cycle, physical design specifications must be turned into working computer code, the code must be tested until the most of the errors have been detected and corrected, the system must be installed, user sites must be prepared for new system, and users must come rely on the new system rather than the existing one to get their work done.

Implementing a new information system into an organizational context is not a mechanical process. The organizational context has been shaped and reshape by the people who work in the organization.

Generally, the development environment is suited according to different development phases, which can be categorized into system design, system development and report writing process.

5.1.1 System Design

Although system design is clearly stated in chapter 4, nevertheless, during the initial stage of system development, a number of considerations and adjustments were done to the initial system design in order to match the actual needs and requirements.

5.1.2 System Development

The basic tools used for the system development are:

- ASP.NET (Program coding tool)
- Microsoft Window 2000 Server (Operating System)
- SQL Server 2000 Database (Database development tool)
- Adobe Photoshop 7.0 (Image/Graphic creation tool)
- Notepad and Edit plus (Editor for HTML)
- Macromedia Dreamweaver (Image design tools)

- Program coding tool

Table state program coding tool used and their purpose.

Tool	Purpose
ASP.NET	Create applications.

Table 5-1: Program coding tool and their purpose

- Operating System

Table state operating system used and their purpose.

Tool	Purpose
Window 2000	Needed when implementing database with Microsoft SQL Server.

Table 5-2: Operating system and their purpose

- Database development tool

Table state database development tool used and their purpose.

Tool	Purpose
Microsoft SQL Server	Database used for SIS
Microsoft Query in Microsoft SQL Server	Execute SQL statements
SQL Server Manager in Microsoft SQL Server	Set the SQL Server running
SQL Enterprise Manager in Microsoft SQL Server	View and edit tables created in database and create jobs.

Table 5-3: Database development tool and their purpose

- Image/Graphic creation tool

Table state image/graphic creation tool used and their purpose.

Tool	Purpose
Adobe Photoshop 7.0	Create graphic

Table 5-4: Image/Graphic tool and their purpose

5.1.3 Report Writing

All the problems encountered, together with solutions found throughout the processes (from system implementation until system evaluation) were recorded as well as result from system testing and system integration.

5.2 System Coding – Coding Approach, Style and Scripting Language

5.2.1 Database Implementation

For SIS, the database is stored in a distributed server in which any data creation, updates or data retrieval will be connected directly to the database server through ADO.NET.

The database includes tables to keep users' details including users' authentications information. SIS is an online application in which the users can create, edit and delete any records directly into the SIS database.

After the SIS is completed and tested successfully, all the raw data were flush from the database. All the unnecessary tables were eliminated from SIS database to avoid data overlapping and to reduce workload of the entire system when deployment.

5.2.2 Application Server Configuration

Internet Information Server (IIS) is a Microsoft's offering a Web publishing and web server that allow users of windows NT/2000 to serve web page on the Internet. IIS is available in both Professional and Server version of Windows 2000.

Essentially, all Web page files should be place into the default directory of \InetPub\wwwroot\ and naming the home page Default.htm or Default.aspx. Administrator can choose to create a virtual directory instead of place all the web page in the default root folder.

5.3 Program Implementation

5.3.1 Coding Approach

The methodology used in this development of the SIS is the incremental prototyping methodology. This phase will begin with module design, followed by the implementation of preliminary prototype. On completion of the preliminary prototype, additional functions are added into prototype. This phase is interactive and may require trace backs to previous stages within the incremental prototyping phase if error were found. It ends with the complete implementation of module.

ASP.NET with VB.NET is used to develop the entire SIS, which code behind method id used. ASP.NET forms are divided into two sections: the user interface and the logic (or code). The user interface comprises HTML markup and ASP.NET web controls, whereas the logic is the programmatic code that interacts with the user interface. This make the page will look more simple and easier to manage.

To increase the coding readability and to help in future enhancements, a page is formed by small pieces of files through the use of "INCLUDE". This is very important as it reduces workload of system developers especially when they make changes on the layout of interfaces. Besides, it also enables system to be developed in shortest time as it allows few developers to work on separate modules at the same time.

5.3.2 Coding for stored procedures

1. Stored procedures were created with Ms Query in Microsoft SQL Server.
2. Each stored procedure was developed using Transact SQL.
3. All stored procedures were store in SIS database.

5.3.3 Coding for application

1. Application was developed with the standard exe module of Microsoft ASP.NET. Each unit was an ASP.NET Form.
2. All forms were stored in ASP.NET project.
3. This project was later compiled to obtain an EXE file.
4. Packet and delepment wizard was later used to create setup files so that this application can be installed in client.

CHAPTER 6 : Testing

6.1 Introduction

Testing phase is carried out right after finished the design and implementation phase of the software development life cycle (SDLC) and cannot be missed out. This technique is applied to this system in order to avoid unwanted errors during the running of the system.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. Once source code has been generated, system must be tested as many errors as possible before delivery to the targeted customer.

Glen Myers states a number of rules that can serve well as testing objectives:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an as-yet-discovered error.
- A successful test is one that uncovers an as-yet-discovered error.

These objectives imply a dramatic change in viewpoint. They move counter to the commonly held view that a successful test is one in which no errors found. The main objective is to design tests that systematically uncover different classes of errors and t do so with a minimum amount of time and effort.

If testing is conducted successfully according to the objectives previously, it will uncover errors in the system. In addition, testing demonstrates that software functions appear to be working according to specification, that behavioral and performance requirements appears to have been met.

Software testability is simply how easily a computer program can be tested. There are certainly metrics that could be used to measure testability in most of its aspects. Sometimes, testability is used to mean how adequately a particular set of tests will cover the product. The set of characteristics that lead to testable software are Operability, Observability, Controllability, Decomposability, Simplicity and Stability. A good test has a high probability of finding error, is not redundant, should be "best of breed" and also should be either too simple or too complex.

The system can be tested in one of two ways:

1. Knowing the specified function that a product has been designed to perform, tests can be conducted that demonstrate each function is fully operational while at the same time searching for errors in each function.
2. Knowing the internal workings of a product, tests can be conducted to ensure that "all gears mesh," that is, internal operations are performed according to specifications and all internal components have been adequately exercised. The first test approach is called black box- testing and the second, white-box testing.

6.1.1 Black-box testing

Black-box testing alludes to tests that are conducted at the software interface. Although they are designed to uncover errors, black-box testing also can be used to demonstrate that software functions are operational, that input is properly accepted and output is correctly produced, and that the integrity of external information is maintained. A black-box testing examines some fundamental aspect of a system with little regard for the internal logical structure of the system.

Black-box testing also called behavioral testing, focuses on the functional requirements of the system. That is, black-box testing enables the software engineer to derive sets of input conditions that will fully exercise all function requirements for a program. Black-box testing is not an alternative to white-box testing. It is a complementary approach that is likely to uncover a different class of errors that white-box testing. Black-box testing attempts to find errors in the following categories:

1. Incorrect or missing functions
2. Interface errors
3. Errors in data structures or external database access
4. Behavior or performance errors
5. Initialization and termination errors.

6.1.2 Equivalence partitioning

Equivalence partitioning is a black-box testing method that divides the input domain of a program into classes of data from which test cases can be derived. An ideal test case single-handedly uncovers a class of errors that might otherwise require many cases to be executed before the general error is observed. Equivalence partitioning strives to define a test case that uncovers classes of errors, thereby reducing the total number of test cases that must be developed.

Test case design for equivalence partitioning is based on an evaluation of equivalence classes for an input condition. Equivalence class is set of valid or invalid states of input conditions.

6.2 Testing Process

In general, the testing process of Supermarket Inventory System can be shown in the following figure. All the details will be further explained in subsequent sub-sections.

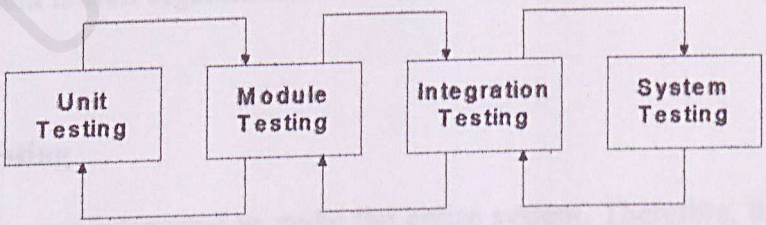


Figure 6-1: Testing Process

6.2.1 Types of Testing

1) Unit Testing

Unit test is the process to test the individual component to ensure that they function properly. Each component is tested independently without the interference from other system components. Unit test is performed concurrently with the development process.

2) Module Testing

Module testing is performed without other system modules. A module consists of a collection of dependent components 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.

3) Integration Testing

Integration test is needed when all modules are integrated. The main focus in integration test is to navigate the interfaces repeatedly to detect any interface mismatch problem. Several important aspects are checked to ensure that the flow of the data in Supermarket Inventory System is well organized and are user friendly to all the system users.

4) System Testing

The sub-systems are integrated to make the entire system. Therefore, the main purpose in system testing is to find errors that result from unanticipated interactions between sub-

systems. Besides, it is used to validate whether the system meets its functional and non-functional requirement.

Finally, a performance test is performed to compare the integrated modules with the non-functional system requirements. These requirements include security, interoperability, flexibility and reliability.

CHAPTER 7 : System Evaluation

7.1 Introduction

Evaluation is the ultimate phase of developing a system and an important phase before delivery the system to the end users. Evaluation was related to user environment, attitudes, information priorities and several other concerns that are to be considered carefully before effectiveness can be concluded. At all phases of the system approaches, evaluation is a process that occurs continuously, drawing on a variety of sources and information.

7.2 Problems Encountered and its Solutions

- Lack of knowledge in application programming

As there is no prior knowledge in application programming in a real world business environment, a lot of studies have to be done to familiarize with the concept of application programming. Programming languages and various application development tools need to be learnt with a short time span. Choosing ASP.NET is a wise decision due to its short learning curve. Discussion with course-mates, seeking advice from the internet and self studies gave a lot of knowledge.

- Wide area of studies

In order to successfully develop and implement system, researches have to be done. Insufficient knowledge about the implementation was the major problem faced during the initial stage. Extensive studies have to be done in order to understand its

requirements. Furthermore, various technologies and tools had to be explored in order to choose the right tools.

The internet was a great help in searching for necessary information. Besides, knowledge was also gained by visiting supermarket, interviewing staffs and discussion with my supervisor and fellow course-mates.

- Difficulties in defining the system scope

At the beginning stage, I was having doubts on the features and functions that I should include into the system because of limited information gathered. But this is clarified after I had visited a few web sites. Moreover, the research session and reading on various books have helped me in determining the features that I should include in my system. Due to the time frame given, it was impossible to incorporate too many features into system. From time to time, I have to consult my supervisor for more advice.

7.3 System Strengths

- Security feature

Every user has a login id and password. First time login, users must change their login id and password. This will prevent unauthorized users from accessing the system. An invalid login message will be prompted if a user tries to use this system without logging in or logging in wrongly.

- User friendliness and easy-to-use interface

SIS is developed based on GUI. Some useful buttons are provided which gives users faster access. Besides, it has a user-environment. This user-friendly interface can shorten the learning curve and reduce training costs which include money and time.

- Generate report (Annual Sales report)

The system will automatically generate a graph from the data for clearer view of supermarket's performance.

- Provides data access

For data to be useful, it is organized and stored in the form of database. Furthermore, it is real time database information and any changes made to the records can be updated instantly to the SIS database.

- Provides an easy to use tool

The command and the layout of SIS are simple and wall organized. Therefore, it is easy to use, simple to learn up and understandable. Normal users with some computer knowledge will find SIS easy to handle.

- Information message

SIS provides error messages when a user attempts to perform illegal actions. It also provides message after a certain task has been completed. These messages allow the users to understand what is going on and keep users informed of what has been done.

7.4 Limitation

- Lack of functional modules

Currently, the available functional modules in SIS are limited. This is because lack of information gathered during the data collection and time constraint.

- Database not encrypted

Inventory information stored in server database is not secured enough because it is stored in a plain test format rather than in encrypted format.

- Insufficient functionality

Due to time constraints, SIS that created does not reflect a real SIS for supermarket. This system is only a miniature of a complete SIS for supermarket.

7.5 System Constraints and Future Enhancements

As mentioned before, SIS is still not fine enough to work at its full efficiency. Some refining work needs to be done to the system to increase its usability and reliability. The aspects to be refine and some suggestions to upgrade the system are as below:

- Provide discount scheduler

There should be a module that enable user to schedule the discount period for certain inventory. The sale price should be able to change to discount price according to the scheduler and change back to normal price automatically.

- Alert to the shortage of item

SIS can inform the user of the system about the level of the item when it is in shortage. This function will enable the system's user alert to the shortage of the item and will take action to order the item as soon as possible before it is out of stock.

- Database encrypted

Inventory information stored in server database is more secured because it is stored in an encrypted format.

CHAPTER 8 : Conclusion

Supermarket Inventory System is developed using ASP.NET (Application programming Language), Microsoft Window 2000 Server (Operating System) and SQL Server 2000 Database (Database Management System)

This project has offers me the opportunity to enhance the knowledge and skills learnt from lecture. Initially, I had approach the manager, supervisor, clerk and cashier from the Gedung Harian Supermarket and KP Minimarket during the requirements acquisition. From the interview, I had collected information that is needed to develop the system. The modeling process requires repeatedly review and rectification to produce the right models that are essential to the well design of system. This is the chance for me to revise and enhance the theoretical knowledge that I had learnt which are analysis modeling, data design, interface design and functional design. In addition, my programming knowledge and skills had been enhanced through application coding process.

Upon the completion of this system, I had gained the experience, knowledge and confidant from this system development. Hopefully, with these experience and knowledge, I will be capable to accomplish future software development tasks in a more efficient and effective manner.

References

- Comparing JavaServer Pages™ and Microsoft® Active Server Pages™ Technologies. Available at: http://java.sun.com/products/jsp/jsp-asp.html

- Deitel & Deitel, T.R.Nieto. (1999). Visual Basic 6 How To Program. California: Prentice-Hall, International, Inc.

- Difference between ASP and JSP. Available at: http://java.sun.com/products/jsp/jsp-asp.html

- http://www.oracle.com

- http://www.zdnet.co.uk/pcmag/labs/1999/04/visual

- http://www.iplanet.com

- Joe Salemi. (1995). Guide to Client/Server Databases. Second edition: Ziff-Davis Press

- Kendall, Kenneth E. and Kendall, Julie E. (1996). System Analysis and Design. 4th edition. California: Prentice-Hall, International, Inc.

- Mandel, T. (1997) The Elements of User Interface Design, Wiley.

- Oxford Advanced Learner's - English-Chinese Dictionary Third Edition (1987). Oxford University Press.

- Pressman, Roger S. (2001) Software Engineering: a practitioner's approach – 5th edition. McGraw-Hill.

- P.Sellappan. (2001). Visual Basic 6 & Internet. 1st edition. Malaysia.

- Sommerwille, I. (1995). Software Engineering. 5th edition. Reading: Addison-Wesley Ltd.

- Wynkoop, Stephen. (1997). Using Microsoft SQL Server 6.5. 2nd edition. New York: Que Corporation. 15-112.

Appendix A: User Manual

Supermarket Inventory System (SIS) is a client server system with the objective to computerize and enhance the inventory control system for the supermarket. This manual is a guide to help user to using SIS effectively to achieve the goal.

Figure A- 1: Login Page

Function:	Log in interface ▪ User key in Login Id and Password. ▪ If the information is valid, the user is allowed to log in the system and different options screen will be display based on the type of user.

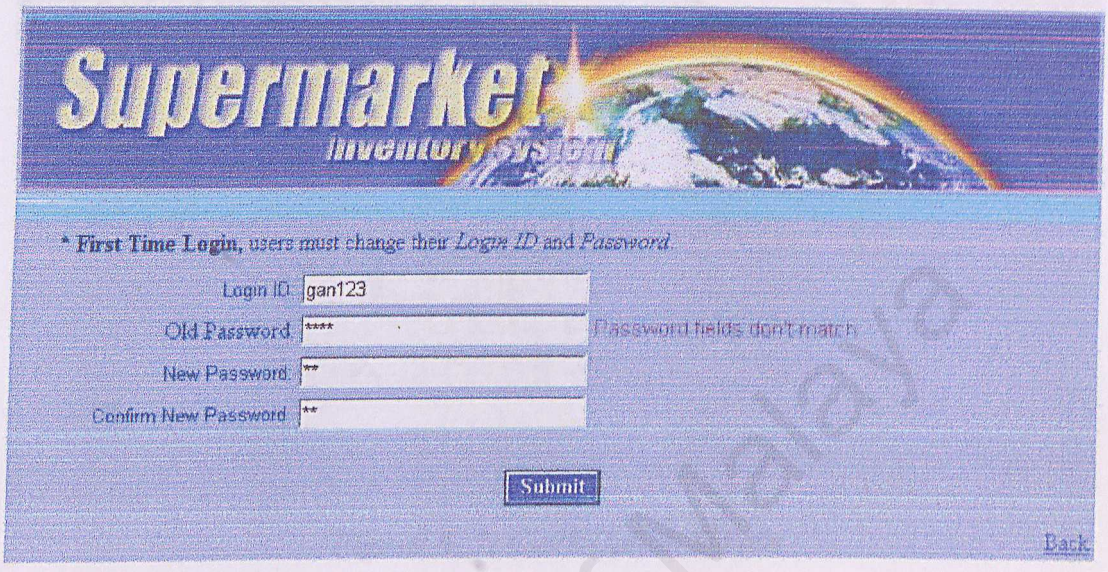


Figure A- 2: First Time Login Page

Function:	First time login interface ▪ This page will display for the new users after login. ▪ First time login, users must change their login id and password. ▪ When the Submit button is clicked, the login id and new password of the users will be updated into the database. ▪ The Back hyperlink will enable the users to back to the Login interface.

Figure A- 3: Clerk Option Screen

Function:	Options Screen interface ▪ This options screen will be display when the valid user is Clerk. ▪ Different options screen will be display when the user is Supervisor or Manager. ▪ Clerk can selects functions from the screen by just clicking the button. ▪ After clicking the button, screen of the chosen function will be displayed. ▪ The Logout button will enable the clerk to back to the Login interface.

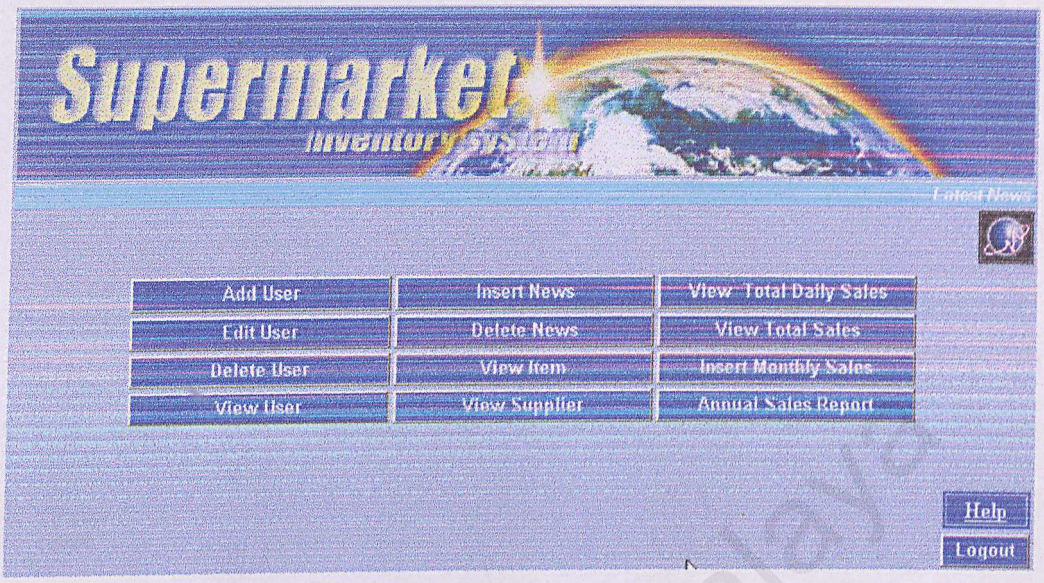


Figure A- 4: Manager Option Screen

Figure A- 5: Supervisor Option Screen

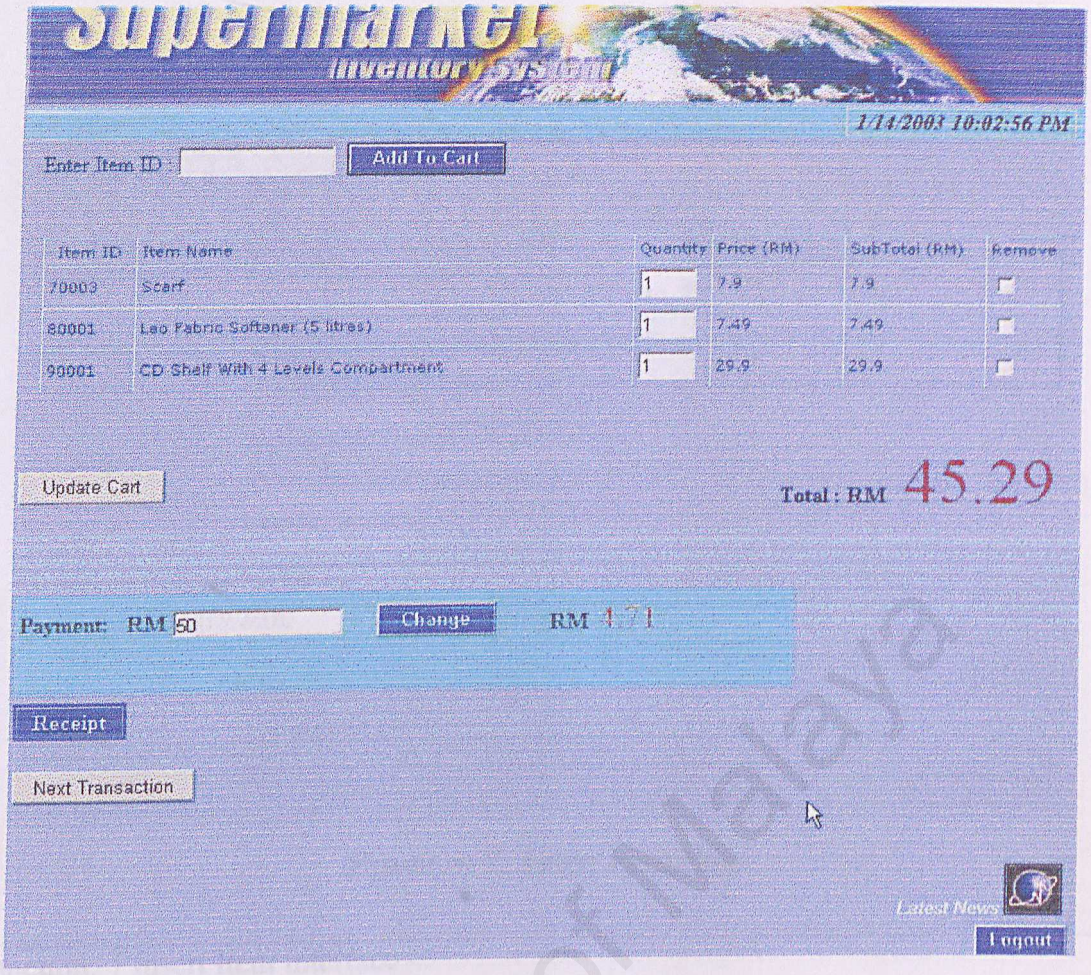
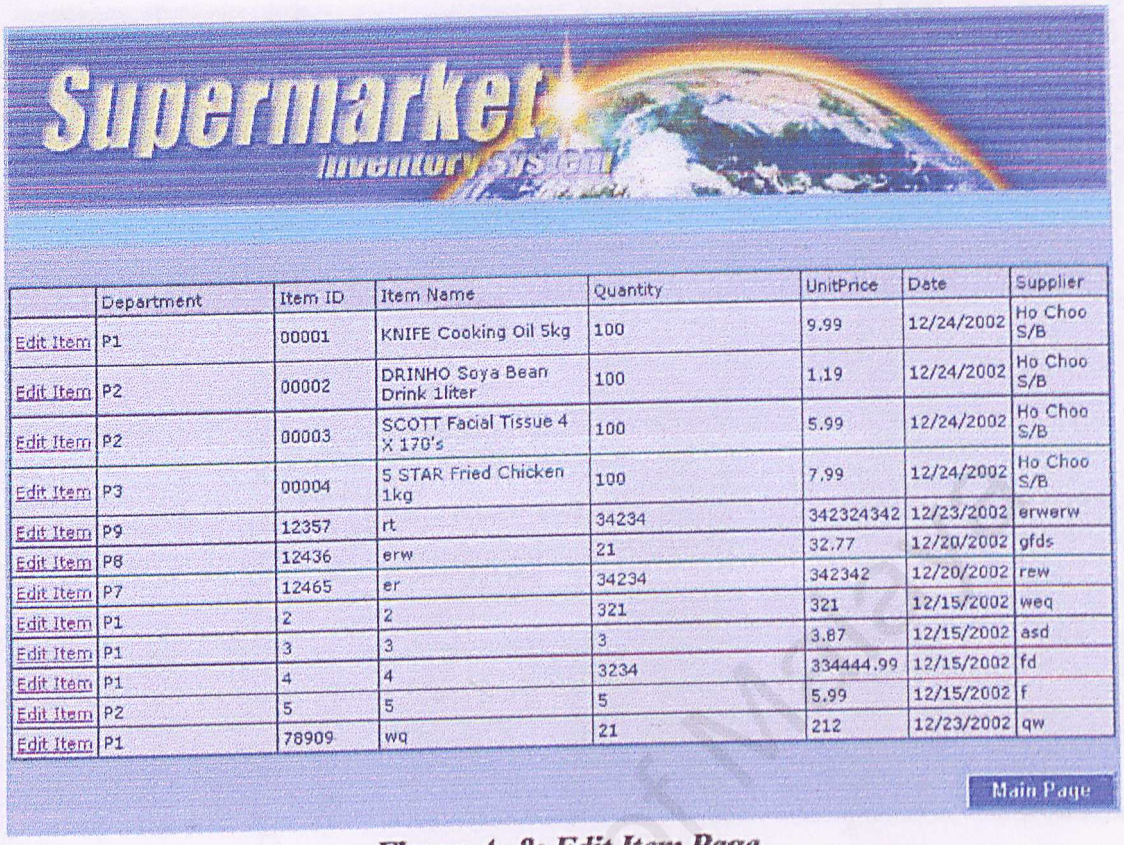


Figure A- 6: Sold Out Page

Function:	Sold Out interface ▪ This interface will be displayed when the valid user is Cashier. ▪ Cashier keys in Item ID in the text box and click Add To Cart button, Item name and item price will be retrieve automatically corresponding to the item id. Subtotal and Total will be calculated automatically. ▪ Cashier can key in Quantity or remove item(click check box) by just clicking Update Cart button ▪ When cashier keys in the value of the cash given by the customer, change will be computed. ▪ In order to print the receipt, cashier just needs to click Receipt button. ▪ In order to end the transaction, the cashier just needs to click the Next Transaction button. ▪ The Logout button will enable the cashier back to the Login interface.

Figure A- 7: Insert Item Page

Function:	Insert Item interface ▪ This screen is display when the Clerk clicks the option of Add Item. ▪ Clerk just needs to select item department from the drop down list, keys in the item id, item name, quantity, price, date and supplier based on the invoice that they had received from the supplier. ▪ When the Add Item button is clicked, the information of the item will be updated into the database. ▪ When the Main Page button is clicked, the options menu screen will be displayed.



	Department	Item ID	Item Name	Quantity	UnitPrice	Date	Supplier
Edit Item	P1	00001	KNIFE Cooking Oil 5kg	100	9.99	12/24/2002	Ho Choo S/B
Edit Item	P2	00002	DRINHO Soya Bean Drink 1liter	100	1.19	12/24/2002	Ho Choo S/B
Edit Item	P2	00003	SCOTT Facial Tissue 4 X 170's	100	5.99	12/24/2002	Ho Choo S/B
Edit Item	P3	00004	5 STAR Fried Chicken 1kg	100	7.99	12/24/2002	Ho Choo S/B
Edit Item	P9	12357	rt	34234	342324342	12/23/2002	erwerw
Edit Item	P8	12436	erw	21	32.77	12/20/2002	gfds
Edit Item	P7	12465	er	34234	342342	12/20/2002	rew
Edit Item	P1	2	2	321	321	12/15/2002	weq
Edit Item	P1	3	3	3	3.87	12/15/2002	asd
Edit Item	P1	4	4	3234	334444.99	12/15/2002	fd
Edit Item	P2	5	5	5	5.99	12/15/2002	f
Edit Item	P1	78909	wq	21	212	12/23/2002	qw

Figure A- 8: Edit Item Page

Function:	Edit Item interface ▪ This screen is display when the Clerk clicks the option of Edit Item. ▪ Clerk just needs to click Edit Item hyperlink, and then any changes to that particular item (except item ID) can be made. These changes can be made when Clerk received Debit Note. ▪ When the Update hyperlink is click, the database will be updated. ▪ When the Main Page button is clicked, the options menu screen will be displayed.

Figure A- 9: Delete Item Page

Function:	Delete Item interface ▪ This screen is display when the Clerk clicks the option of Delete Item. ▪ When Clerk key in Item ID, Information about the Item in the database will be displayed. ▪ When the Delete Item button is clicked, the information about that certain particular item will permanently deleted from the database and the database will be updated. ▪ When the Main Page button is clicked, the options menu screen will be displayed.

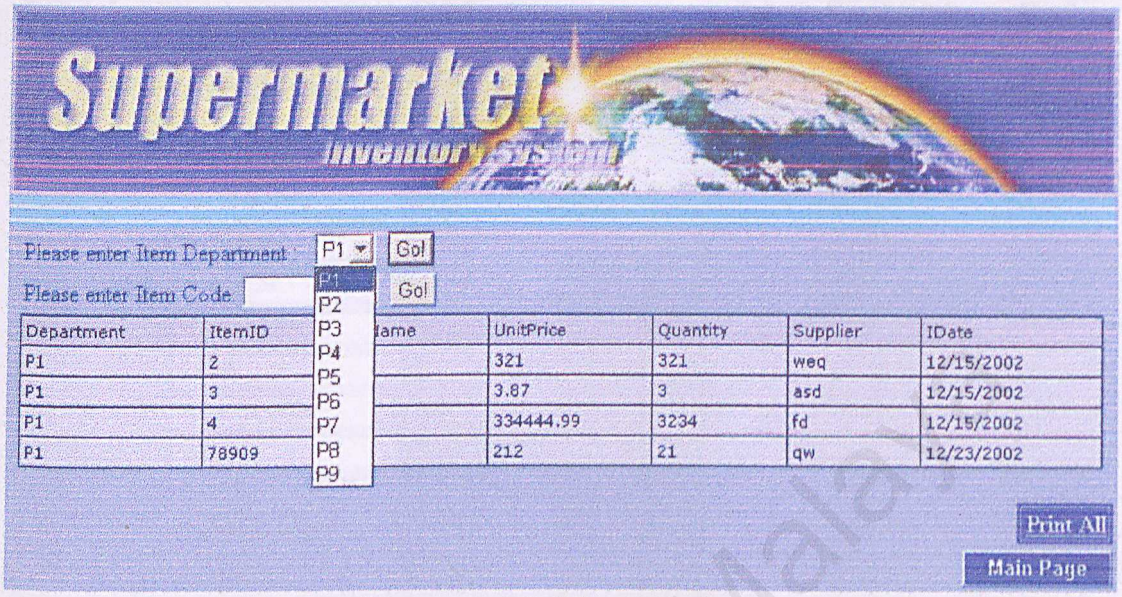
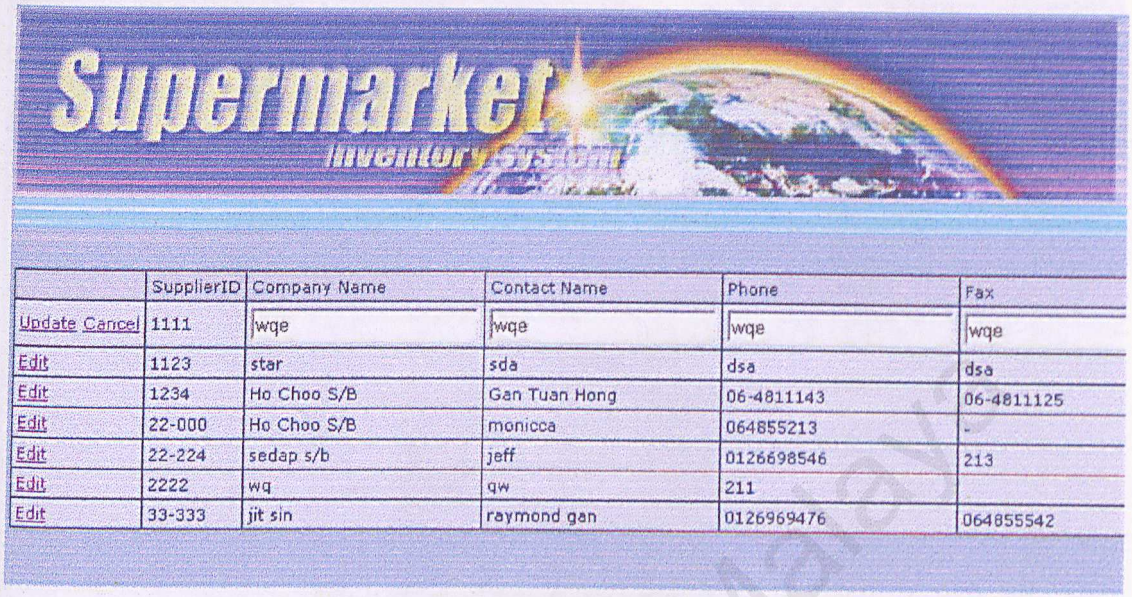


Figure A- 10: View Item Page

Function:	View Item interface ▪ This screen is display when the Clerk, Supervisor or Manager clicks the option of View Item. ▪ Clerk, Supervisor or Manager can view the lastest information of the items in the database by just select item department or key in Item ID. The information diplayed cannot be modified ▪ When the Main Page button is clicked, the options menu screen will be displayed.

Figure A- 11: Insert Supplier Page

Function:	Insert Supplier interface ▪ This screen is display when the Clerk clicks the option of Add Supplier. ▪ Clerk just needs to keys in supplier id, company name, contact name, address, postcode, state, phone number and fax number based on the information that they had received from the supplier. ▪ When the Add Supplier button is clicked, the information of the supplier will be updated into the database. ▪ When the Main Page button is clicked, the options menu screen will be displayed.



	SupplierID	Company Name	Contact Name	Phone	Fax
Update Cancel	1111	wqe	wqe	wqe	wqe
Edit	1123	star	sda	dsa	dsa
Edit	1234	Ho Choo S/B	Gan Tuan Hong	06-4811143	06-4811125
Edit	22-000	Ho Choo S/B	monicca	064855213	-
Edit	22-224	sedap s/b	jeff	0126698546	213
Edit	2222	wq	qw	211	
Edit	33-333	jit sin	raymond gan	0126969476	064855542

Figure A- 12: Edit Supplier page

Function:	Edit Supplier interface ▪ This screen is display when the Clerk clicks the option of Edit Supplier. ▪ Clerk just needs to click Edit hyperlink, and then any changes to that particular Supplier (except Supplier ID) can be made. These changes can be made when Clerk received latest information of that certain supplier. ▪ When the Update hyperlink is click, the database will be updated. ▪ When the Main Page button is clicked, the options menu screen will be displayed.

Figure A- 13: Delete Supplier Page

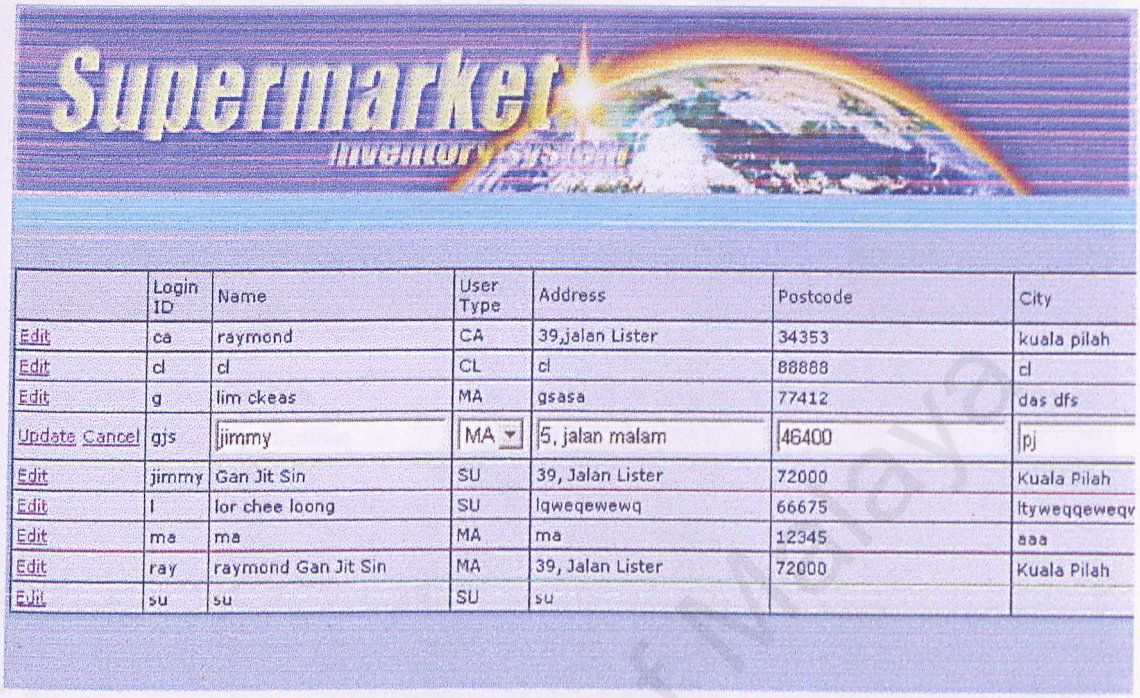
Function:	Delete Supplier interface ▪ This screen is display when the Clerk clicks the option of Delete Supplier. ▪ When Clerk key in Supplier ID or select supplier name, information about the supplier in the database will be displayed. ▪ When the Delete Supplier button is clicked, the information about that certain particular item will permanently deleted from the database and the database will be updated. ▪ When the Main Page button is clicked, the options menu screen will be displayed.

Figure A- 14: View Supplier Page

Function:	View Supplier interface ▪ This screen is display when the Clerk, Supervisor or Manager clicks the option of View Supplier. ▪ Clerk, Supervisor or Manager can view the lastest information of the items in the database by just select item department or key in Item ID. The information diplayed cannot be modified ▪ When the Main Page button is clicked, the options menu screen will be displayed.

Figure A- 15: Add New User Page

Function:	Insert New User interface ▪ This screen is display when the Manager clicks the option of Add User. ▪ Manager needs to create new login id and password for the new user and key in user type and user's personal information. ▪ When the Submit button is clicked, the information of the user will be updated into the database. ▪ When the Main Page button is clicked, the options menu screen will be displayed.



	Login ID	Name	User Type	Address	Postcode	City
Edit	ca	raymond	CA	39,jalan Lister	34353	kuala pilah
Edit	cl	cl	CL	cl	88888	cl
Edit	g	lim ckeas	MA	gsasa	77412	das dfs
Update Cancel	gjs	jimmy	MA	5, jalan malam	46400	pj
Edit	jimmy	Gan Jit Sin	SU	39, Jalan Lister	72000	Kuala Pilah
Edit	l	lor chee loong	SU	lqweqewewq	66675	ltyweqqeweqv
Edit	ma	ma	MA	ma	12345	aaa
Edit	ray	raymond Gan Jit Sin	MA	39, Jalan Lister	72000	Kuala Pilah
Edit	su	su	SU	su		

Figure A- 16: Edit User Page

Function:	Edit User interface ▪ This screen is display when the Manager clicks the option of Edit User. ▪ Manager just needs to click Edit hyperlink, and then any changes to that particular User (except Supplier ID) can be made. ▪ When the Update hyperlink is clicked, the database will be updated. ▪ When the Main Page button is clicked, the options menu screen will be displayed.

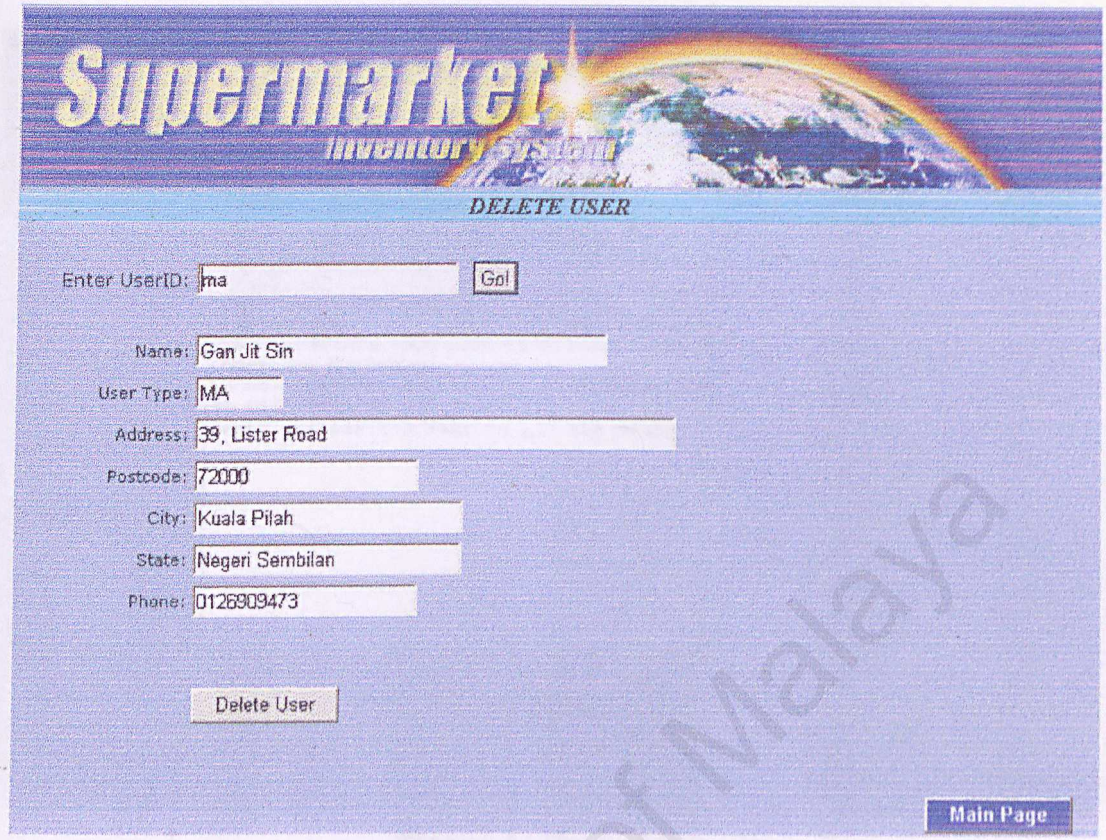


Figure A- 17: Delete User Page

Function:	Delete User interface ▪ This screen is display when the Manager clicks the option of Delete User. ▪ When Manager key in User ID, Information about the user in the database will be displayed. ▪ When the Delete User button is clicked, the information about that certain particular user will permanently deleted from the database and the database will be updated. ▪ When the Main Page button is clicked, the options menu screen will be displayed.

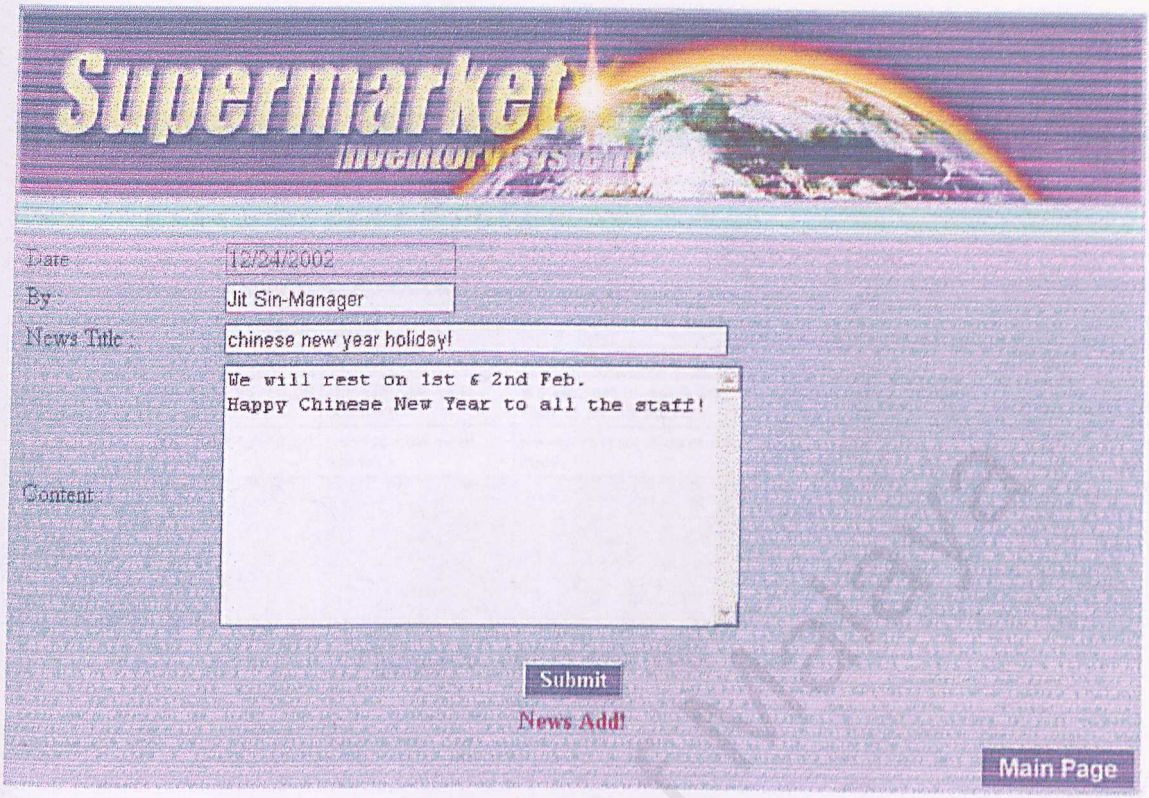


Figure A- 18: Add News Page

Function:	Insert News interface ▪ This screen is display when the Manager clicks the option of Add News. ▪ Manager just needs to key in name, news title and content (latest or important news wants to inform cashier, supervisor and clerk). ▪ When the Submit button is clicked, the information of the news will be updated into the database. ▪ When the Main Page button is clicked, the options menu screen will be displayed.

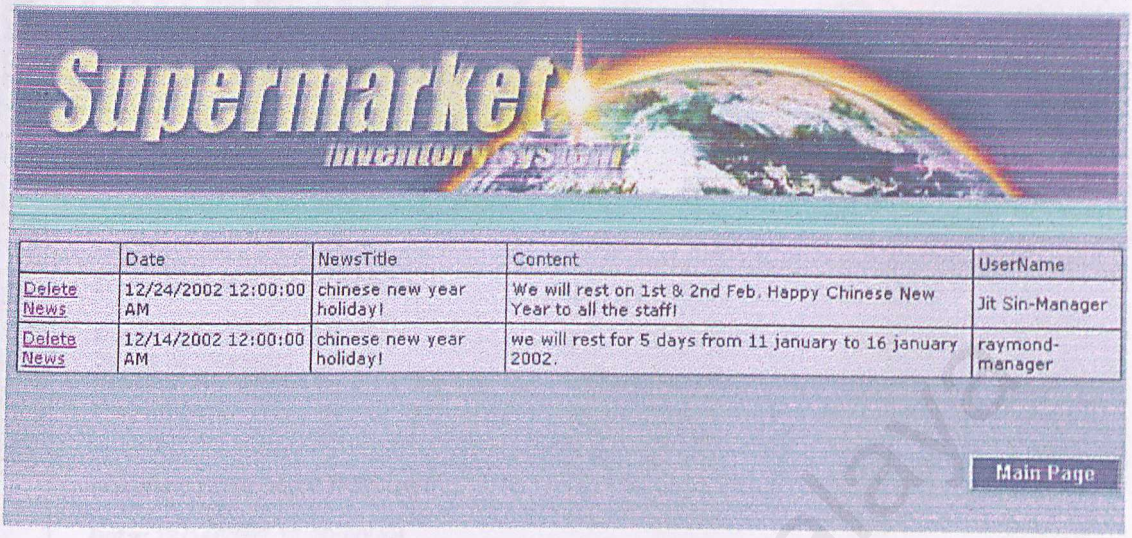


Figure A- 19: Delete News Page

Function:	Delete News interface ▪ This screen is display when the Manager clicks the option of Delete News. ▪ When the Delete News hyperlink is clicked, the information about that certain particular news will permanently deleted from the database and the database will be updated. ▪ When the Main Page button is clicked, the options menu screen will be displayed.

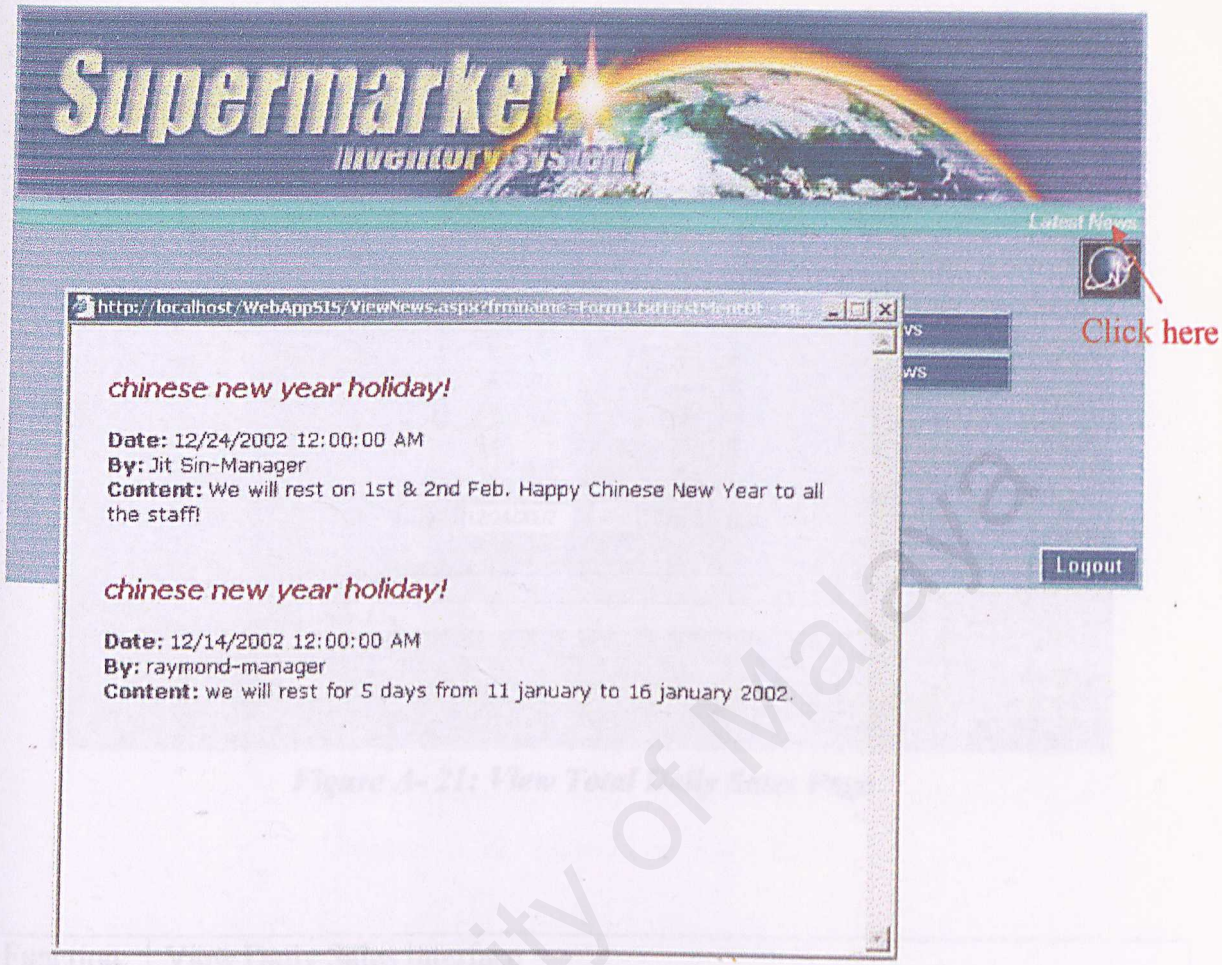


Figure A- 20: View News Page

Function:	View News interface ▪ This pop-up screen is display when the Cashier, Clerk, Supervisor or Manager clicks the Latest News button on the main page.

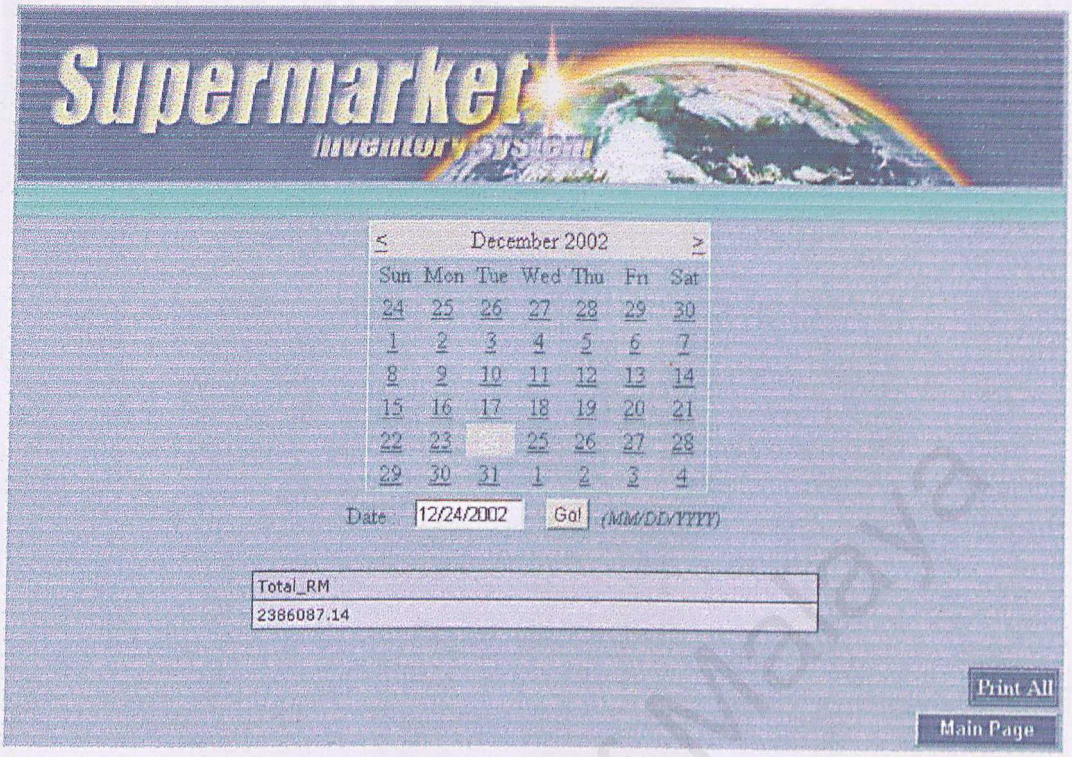


Figure A- 21: View Total Daily Sales Page

Function:	View Daily Sales interface ▪ This screen is displayed when the Manager clicks the option of View Daily Sales. ▪ This screen is only accessible to the Manager. ▪ The Manager just needs to click the date on the calendar provided, and then click the Go button. Then the total sales for that selected date will be displayed. ▪ When the Main Page button is clicked, the options menu screen will be displayed.

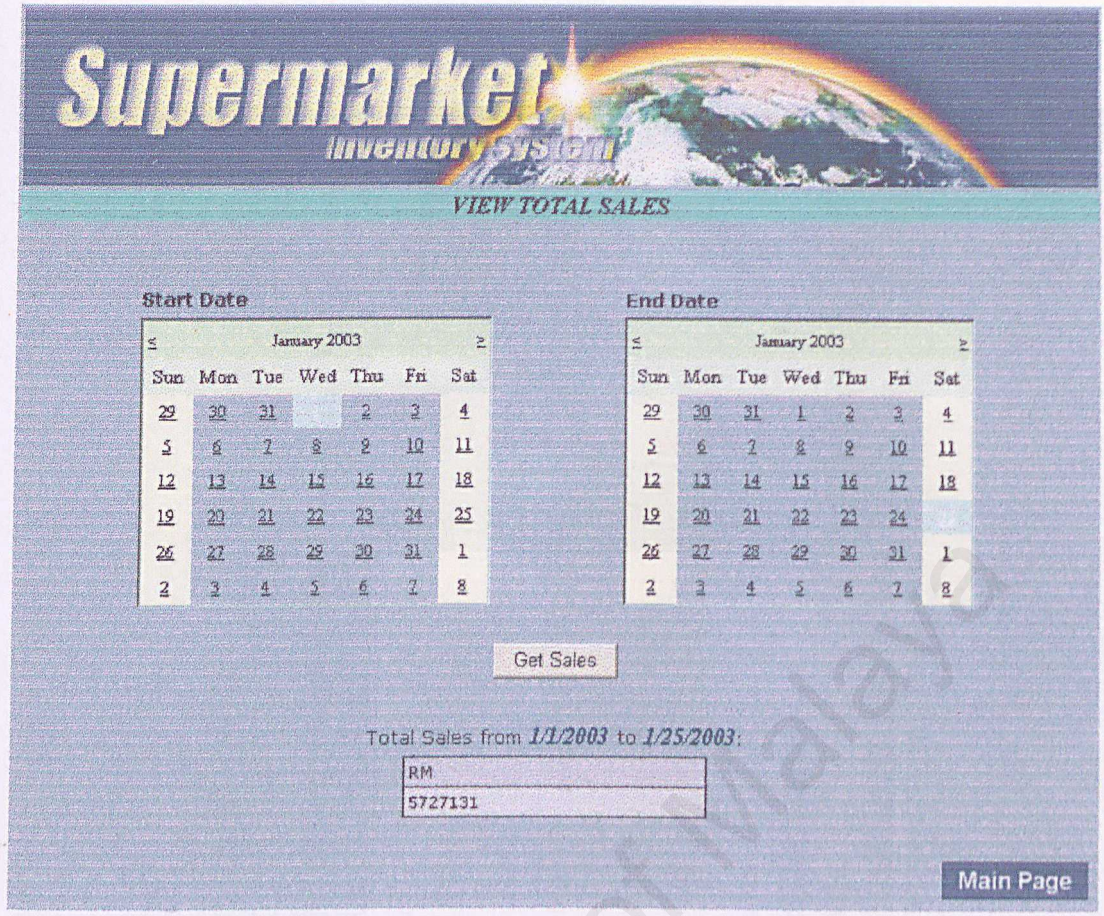


Figure A- 22: View Total Sales Page

Function:	View Total Sales interface ▪ This screen is displayed when the Manager clicks the option of View Total Sales. ▪ This screen is only accessible to the Manager. ▪ The Manager just needs to click the date (start date and end date) on the calendar provided, and then click the Get Sales button. Then the total transaction for that selected date will be displayed. ▪ When the Main Page button is clicked, the options menu screen will be displayed.

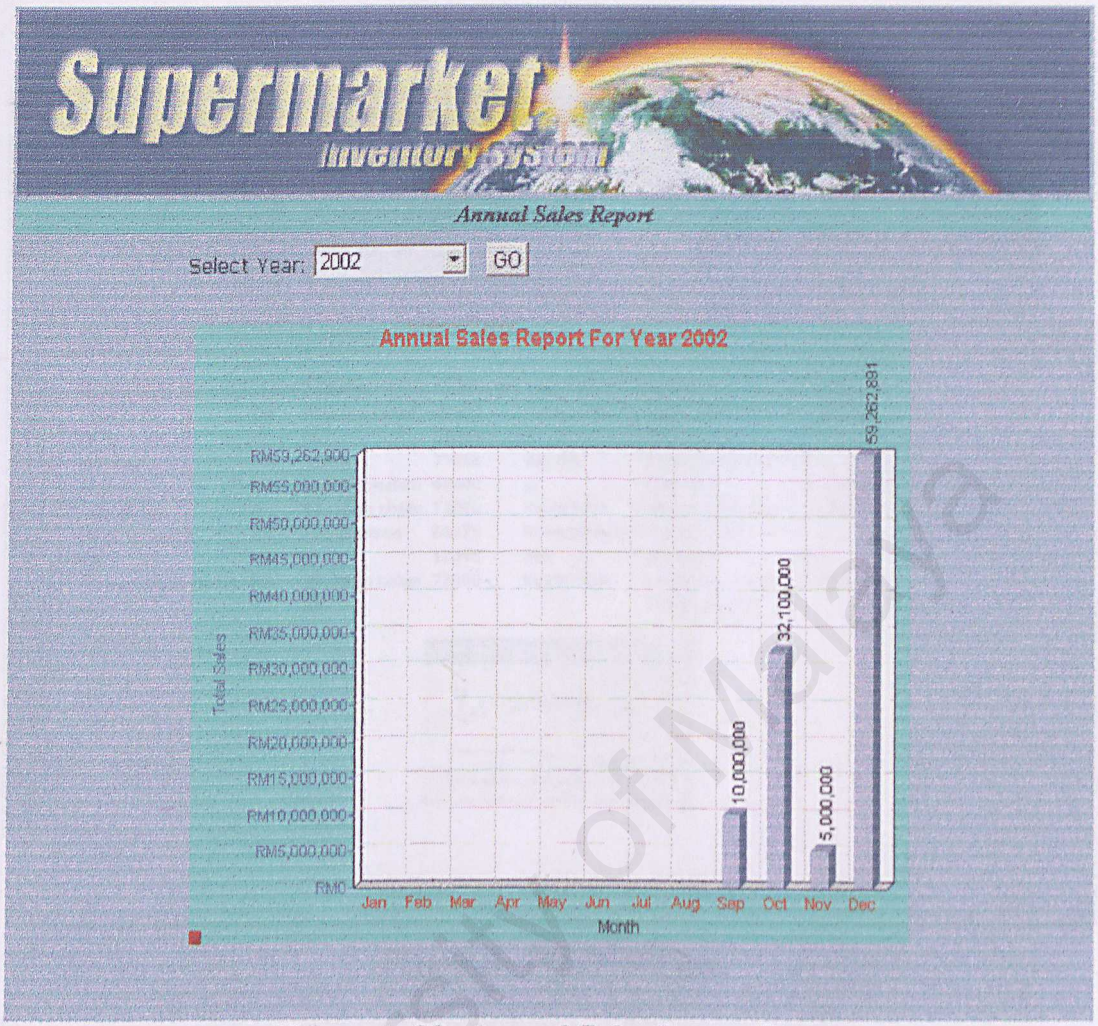


Figure A- 23: Annual Sales Report Page

Function:	Annual Sales Report interface ▪ This screen is displayed when the Manager clicks the option of Annual Sales Report. ▪ This screen is only accessible to the Manager. ▪ The Manager just needs to select the year, and then click the Go button. The system will automatically generate a graph from the data for clearer view of supermarket's performance.

Figure A- 24: Print Page

Function:	Print Information interface ▪ This pop-up screen is display when the Clerk, Supervisor or Manager clicks the Print All button. They can print the user, item or supplier's details.

SUPERMARKET No: 25

Tel: 06-4811143 Date: 12/29/2002 4:49:20 PM

Item Name	Quantity	Price (RM)	Subtotal (RM)
kacang	1	3.09	3.09
KNIFE Cooking Oil 5kg	12	9.99	119.88
DRINHO Soya Bean Drink 1liter	1	1.19	1.19
SCOTT Facial Tissue 4 X 170's	3	5.99	17.97
5 STAR Fried Chicken 1kg	1	7.99	7.99

Total: RM 150.12

***Remarks:**

Please retain receipt in case of refunds.

Refund is within 8 days of purchase.

THANK YOU, SEE YOU AGAIN!

Figure A- 25: Receipt

Appendix B: Installation Guide

- Deploy the Supermarket Inventory System

  1. Explore the SIS folder
  2. Open the debug folder and run setup.exe.
  3. Click Next along the installation and do not make any changes, setup will be done automatically.

- Create User and Table in the Database (Microsoft SQL 2000)

  1. Run the SQL Server Enterprise Manager.
  2. Expand the Microsoft SQL Servers icon.
  3. Right click on the (LOCAL)(Windows NT) and select Properties.
  4. Select the Security tab and check the radio button on SQL Server and Windows. Press Ok.
  5. Expand (LOCAL)(Windows NT) and select Databases.
  6. Right click on the Databases and select New Database.
  7. Key in the Name as SIS and press Ok.
  8. Expand the SIS.
  9. Right click on Tables and select New Table to create tables as below.

	Column Name	Data Type	Length	Allow Nulls
🔑	CRecordNo	int	4	
	CartID	char	10	✓
	ItemID	char	6	✓
	Quantity	int	4	✓

Table Name: Carts

Column Name	Data Type	Length	Allow Nulls
Type	char	10	✓
Counter	int	4	✓

Table Name: Counters

Column Name	Data Type	Length	Allow Nulls
Department	char	3	✓
ItemID	char	6	
ItemName	varchar	50	✓
UnitPrice	money	8	✓
Quantity	int	4	✓
Supplier	varchar	50	✓
IDate	varchar	11	✓

Table Name: Items

Column Name	Data Type	Length	Allow Nulls
[Date]	datetime	8	✓
NewsTitle	varchar	100	
Content	varchar	1000	✓
UserName	varchar	50	✓

Table Name: TblNews

Column Name	Data Type	Length	Allow Nulls
SiriNo	int	4	✓

Table Name: TblReceipt

Column Name	Data Type	Length	Allow Nulls
TranDate	datetime	8	
Total_Daily_Transaction	money	8	✓

Table Name: TblTransaction

Column Name	Data Type	Length	Allow Nulls
LoginId	varchar	20	
Password	varchar	20	
Type	char	3	
Name	varchar	50	
Address	varchar	70	✓
Postcode	varchar	5	✓
City	varchar	50	✓
State	varchar	50	✓
Phone	varchar	50	✓
status	int	4	✓
Email	varchar	30	✓

Table Name: TblUserLog

Column Name	Data Type	Length	Allow Nulls
SupplierId	char	6	
CompanyName	varchar	50	
ContactName	varchar	50	✓
Address	varchar	100	✓
Postcode	char	5	✓
City	varchar	20	✓
State	varchar	20	✓
Phone	varchar	50	✓
Fax	varchar	12	✓

Table Name: TblSupplier

10. Right click on Stored Procedures and select New Stored Procedures to create Stored Procedures as below.

```
CREATE Procedure UpdateQuantity
(
@ItemID      char(6),
@Quantity    int
)
AS

UPDATE Items
SET Quantity=@Quantity
WHERE ItemID=@ItemID
```

Stored Procedure Name: UpdateQuantity

```
CREATE Procedure CartItemCount
(
   @CartID   char(10),
   @ItemCount int OUTPUT
)
AS

SELECT
   @ItemCount = COUNT(ItemID)

FROM
   Carts

WHERE
   CartID = @CartID
GO
```

Stored Procedure Name: CartItemCount

```
CREATE Procedure CartList
(
   @CartID char(10)
)
AS

SELECT
   Carts.ItemID,
   Items.ItemName,
   Carts.Quantity,
   Items.UnitPrice,
   Cast((Items.UnitPrice * Carts.Quantity) as money) as SubTotal

FROM
   Items,
   Carts

WHERE
   Items.ItemID = Carts.ItemID
 AND
   Carts.CartID = @CartID

ORDER BY
   Items.ItemID
GO
```

Stored Procedure Name: CartList

```
CREATE Procedure CartRemoveItem
(
   @CartID char(10),
   @ItemID char(6)
)
AS

DELETE FROM Carts

WHERE
   CartID = @CartID
  AND
   ItemID = @ItemID
GO
```

Stored Procedure Name: CartRemoveItem

```
CREATE Procedure CartTotal
(
   @CartID    char(10),
   @Total money OUTPUT
)
AS

SELECT
   @Total = SUM(Items.UnitPrice * Carts.Quantity)

FROM
   Carts,
   Items

WHERE
   Carts.CartID = @CartID
  AND
   Items.ItemID = Carts.ItemID
GO
```

Stored Procedure Name: CartTotal

```
CREATE Procedure CartUpdate
(
   @CartID   char(10),
   @ItemID   char(6),
   @Quantity  int
)
AS

UPDATE Carts

SET
   Quantity = @Quantity

WHERE
   CartID = @CartID
 AND
   ItemID = @ItemID
GO
```

Stored Procedure Name: CartUpdate

```
CREATE Procedure GetItemQuantity
(
@ItemID      char(6),
@Quantity     int OUTPUT
)
AS

SELECT @Quantity=Quantity
FROM Items
WHERE ItemID=@ItemID
GO
```

Stored Procedure Name: GetItemQuantity

11. Right click on User and select New Database User.

12. In the General tab, select <new> from the Login name.

13. Enter Name as ray.

14. Select SQL Server Authentication. Enter Password as 55315575 and Database as SIS.

15. Select the Database Access tab on the top, select SIS and press Ok. Close the Database User Properties.
16. Select User icon and right click the ray user to select Properties.
17. Click the Permissions button.
18. Checked all check box in (Select, Insert, Update, Delete and DRI) for the row named Items, TblSupplier, TblTransaction, TblUserLog, TblNews, Carts, Counters, TblSales and TblReceipt.
19. Checked all check boxes in (Exec) for the row named UpdateQuantity, GetItemQuantity, CartUpdate, CartTotal, CartRemoveItem, CartList and CartItemCount.

- Navigate the AP Online Forum System

1. Run Internet Explore.
2. Navigate the URL http://localhost\WebAppSIS \Supermarket.html

Appendix C: Program Source Codes

C.1 Sold Out Page

Page Source: SalesList.aspx, SalesList.aspx.vb

Component Source: CartDB

Stored Procedures: CartList, CartUpdate, CartRemoveItem, CartItemCount, CartTotal, UpdateQuantity, GetItemQuantity

C.1.1 SalesList.aspx.vb

```
Imports System.Data
Imports System.Data.SqlClient

Public Class SalesList
    Inherits System.Web.UI.Page
    Protected WithEvents Image1 As System.Web.UI.WebControls.Image
    Protected WithEvents date1 As System.Web.UI.WebControls.Label
    Protected WithEvents txtAddItem As System.Web.UI.WebControls.TextBox
    Protected WithEvents btnAddItem As System.Web.UI.WebControls.Button
    Protected WithEvents conStatus As System.Web.UI.WebControls.Label
    Protected WithEvents MyDataGrid As System.Web.UI.WebControls.DataGrid
    Protected WithEvents lblTotal As System.Web.UI.WebControls.Label
    Protected WithEvents TextBoxCash As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label9 As System.Web.UI.WebControls.Label
    Protected WithEvents ButtonChange As System.Web.UI.WebControls.Button
    Protected WithEvents LabelChange As System.Web.UI.WebControls.Label
    Protected WithEvents CompareValidator1 As System.Web.UI.WebControls.CompareValidator
    Protected WithEvents SalesDetails As System.Web.UI.WebControls.Panel
```

```
    Protected WithEvents HyplFirstMentDt As System.Web.UI.WebControls.HyperLink
    Protected WithEvents Button1 As System.Web.UI.WebControls.Button
    Protected WithEvents MyDataGrid9 As System.Web.UI.WebControls.DataGrid
    Protected WithEvents UpdateCart As System.Web.UI.HtmlControls.HtmlInputButton
    Protected WithEvents HyperLink1 As System.Web.UI.WebControls.HyperLink
    Protected WithEvents nexttransaction As System.Web.UI.HtmlControls.HtmlInputButton

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

    End Sub

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Dim MyConnection As SqlConnection
    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'get today date
        Dim MyDate1 As Date
        MyDate1 = Date.Now
        date1.Text = MyDate1

        MyConnection = New SqlConnection("Server=localhost;uid=ray;pwd=55315575;Database=SIS")

        If Not (IsPostBack) Then
            BindGrid()
            If Session("CartID") = "" Then

                Dim cart As SIS.CartDB = New SIS.CartDB()
                Dim cartNo As Integer = cart.GetCartNo()
```

```
        Dim cartID As String = "CID" + CStr(cartNo)
        Session("CartID") = cartID
        cart.IncreaseCartNo(cartNo)
      End If
      PopulateCartList()
    End If
End Sub


Private Sub btnAddItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAddItem.Click
    Dim checkItemID As SIS.CartDB = New SIS.CartDB()
    Dim checkResult As Boolean = checkItemID.CheckItemIDExist(Server.HtmlEncode(Trim(txtAddItem.Text)))

    If checkResult = True Then
      Response.Redirect("AddToCart.aspx?ItemID=" + Trim(txtAddItem.Text))
    Else
      conStatus.Text = "No record found!"
    End If
End Sub

Private Sub UpdateCart_ServerClick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles UpdateCart.ServerClick
    conStatus.Text = ""
    UpdateCartDB()
    PopulateCartList()
End Sub

Sub PopulateCartList()

    Dim cart As SIS.CartDB = New SIS.CartDB()
    If Session("CartID") <> "" Then
      If cart.GetItemCount(Session("CartID")) = 0 Then
        SalesDetails.Visible = False
        conStatus.Text = ""
        TextBoxCash.Text = ""

      Else
        MyDataGrid.DataSource = cart.GetCartDetails(Session("CartID"))
```

```
        MyDataGrid.DataBind()

        lblTotal.Text = cart.GetTotal(Session("CartID"))
      End If
    Else
      SalesDetails.Visible = False
      conStatus.Text = "Missing Cart ID!"
    End If

End Sub

Sub UpdateCartDB()

    Dim cart As SIS.CartDB = New SIS.CartDB()

    Dim i As Integer
    For i = 0 To MyDataGrid.Items.Count - 1
      Dim QuantityTxt As HtmlInputText = CType(MyDataGrid.Items(i).FindControl("txtQty"), HtmlInputText)
      Dim btnRemove As CheckBox = CType(MyDataGrid.Items(i).FindControl("Remove"), CheckBox)

      Dim Quantity As Integer
      Try
        Quantity = CInt(QuantityTxt.Value)

        If Quantity <> CInt(MyDataGrid.DataKeys(i)) Or btnRemove.Checked = True Then
          Dim lblItemID As Label = CType(MyDataGrid.Items(i).FindControl("ItemID"), Label)

          If Quantity = 0 Or btnRemove.Checked = True Then
            cart.CartRemoveItem(Session("CartID"), CStr(Trim(lblItemID.Text)))
          Else
            cart.UpdateCart(Session("CartID"), CStr(Trim(lblItemID.Text)), Quantity)
          End If
        End If
      Catch
        conStatus.Text = "There has been a problem with one or more of your inputs."
      End Try
    Next
```

```
End Sub

Sub UpdateCartDB1()

    Dim cart As SIS.CartDB = New SIS.CartDB()

    Dim i As Integer
    For i = 0 To MyDataGrid.Items.Count - 1
        Dim QuantityTxt As HtmlInputText = CType(MyDataGrid.Items(i).FindControl("txtQty"), HtmlInputText)
        Dim btnRemove As CheckBox = CType(MyDataGrid.Items(i).FindControl("Remove"), CheckBox)

        Dim Quantity As Integer
        Try
            Quantity = CInt(QuantityTxt.Value)

            If Quantity <> CInt(MyDataGrid.DataKeys(i)) Or btnRemove.Checked = False Then
                Dim lblItemID As Label = CType(MyDataGrid.Items(i).FindControl("ItemID"), Label)

                If Quantity = 0 Or btnRemove.Checked = False Then
                    cart.CartRemoveItem(Session("CartID"), CStr(Trim(lblItemID.Text)))
                Else
                    cart.UpdateCart(Session("CartID"), CStr(Trim(lblItemID.Text)), Quantity)
                End If
            End If
        Catch
            conStatus.Text = "There has been a problem with one or more of your inputs."
        End Try
    Next

End Sub

Private Sub nexttransaction_ServerClick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles nexttransaction.ServerClick
    UpdateCartDB1()
    PopulateCartList()
    TextBoxCash.Text = ""
    LabelChange.Text = ""
```

```
    txtAddItem.Text = ""

    If (Page.IsValid) Then

        Dim MyDate As Date
        MyDate = Today
        Dim DS As DataSet
        Dim MyCommand As SqlCommand

        Dim InsertCmd As String = "insert into TblTransaction (TranDate,Total_Daily_Transaction) values (@TranDate,@Total_Daily_Transaction)"

        MyCommand = New SqlCommand(InsertCmd, MyConnection)


        MyCommand.Parameters.Add(New SqlParameter("@Total_Daily_Transaction", SqlDbType.Money, 8))
        MyCommand.Parameters("@Total_Daily_Transaction").Value = lblTotal.Text
        MyCommand.Parameters.Add(New SqlParameter("@TranDate", SqlDbType.DateTime, 8))
        MyCommand.Parameters("@TranDate").Value = MyDate

        MyCommand.Connection.Open()
        MyCommand.ExecuteNonQuery()
        MyCommand.Connection.Close()

    End If

    BindGrid()
End Sub

Sub BindGrid()

    Dim DS As DataSet
    Dim MyCommand As SqlDataAdapter
    MyCommand = New SqlDataAdapter("select * from TblTransaction", MyConnection)

    DS = New DataSet()
    MyCommand.Fill(DS, "TblTransaction")
```

```
    MyDataGrid9.DataSource = DS.Tables("TblTransaction").DefaultView
    MyDataGrid9.DataBind()
End Sub

Private Sub ButtonChange_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButtonChange.Click
    If TextBoxCash.Text = "" Then
        Label9.Text = "*"
    Else
        If lblTotal.Text = "" Then
            LabelChange.Text = TextBoxCash.Text
        Else
            Label9.Text = ""
            LabelChange.Text = TextBoxCash.Text - lblTotal.Text
        End If
    End If

    Dim updateQty As SIS.CartDB = New SIS.CartDB()
    Dim NewQty As Integer

    Dim i As Integer
    For i = 0 To MyDataGrid.Items.Count - 1


        Dim lblItemID As Label = CType(MyDataGrid.Items(i).FindControl("ItemID"), Label)
        Dim QuantityTxt As HtmlInputText = CType(MyDataGrid.Items(i).FindControl("txtQty"), HtmlInputText)

        Dim ItemID As String = CStr(Trim(lblItemID.Text))
        Dim Quantity As Integer = CInt(Trim(QuantityTxt.Value))


        Dim OldItemQty As Integer = updateQty.GetItemQuantity(ItemID)
        If OldItemQty < Quantity Then
            conStatus.Text = "Item " + ItemID + "  not enough stock!"
            Return
        Else
            NewQty = OldItemQty - Quantity
```

```
            End If

            updateQty.UpdateQuantity(ItemID, NewQty)

        Next
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Session("LoginId") = ""
        Response.Redirect("WebForm1.aspx")
    End Sub


    'Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    ' UpdateCartDB()

    ' Dim cart As SIS.CartDB = New SIS.CartDB()
    ' If cart.GetItemCount(Session("CartID")) <> 0 Then
    '  Response.Redirect("resit.aspx")

    ' End If
    ' End Sub
End Class
```

C.1.2 SalesList.aspx

```
Imports System
Imports System.Configuration
Imports System.Data
Imports System.Data.SqlClient

Namespace SIS

  Public Class CartDetails

    Public ItemID As String
    Public ItemName As String
    Public Quantity As Integer
    Public UnitPrice As Decimal

  End Class

  Public Class CartDB

    Dim strConnection As String = ConfigurationSettings.AppSettings("constring")
    Dim myConnection As SqlConnection = New SqlConnection(strConnection)

    Public Function GetCartNo() As Integer
      Dim CartNo As String = "SELECT @Counter=Counter FROM Counters WHERE Type='Cart'"
      Dim MyCommand As SqlCommand = New SqlCommand(CartNo, myConnection)

      Dim parameterCounter As SqlParameter = New SqlParameter("@Counter", SqlDbType.Int, 4)
      parameterCounter.Direction = ParameterDirection.Output
      MyCommand.Parameters.Add(parameterCounter)

      MyCommand.Connection.Open()
      MyCommand.ExecuteNonQuery()
      MyCommand.Connection.Close()

      Return parameterCounter.Value
```

```
End Function

Public Function GetCRecordNo() As Integer
    Dim CRecordNo As String = "SELECT @Counter=Counter FROM Counters WHERE Type='CRecordID'"
    Dim MyCommand As SqlCommand = New SqlCommand(CRecordNo, myConnection)

    Dim parameterCounter As SqlParameter = New SqlParameter("@Counter", SqlDbType.Int, 4)
    parameterCounter.Direction = ParameterDirection.Output
    MyCommand.Parameters.Add(parameterCounter)

    MyCommand.Connection.Open()
    MyCommand.ExecuteNonQuery()
    MyCommand.Connection.Close()

    Return parameterCounter.Value

End Function

Public Sub IncreaseCartNo(ByVal CartNo As String)

    Dim IncreaseCartNo As String = "UPDATE Counters SET Counter=@Counter WHERE Type='Cart'"
    Dim MyCommand As SqlCommand = New SqlCommand(IncreaseCartNo, myConnection)

    MyCommand.Parameters.Add(New SqlParameter("@Counter", SqlDbType.Int, 4))
    MyCommand.Parameters("@Counter").Value = CartNo + 1

    MyCommand.Connection.Open()
    MyCommand.ExecuteNonQuery()
    MyCommand.Connection.Close()

End Sub

Public Sub IncreaseCRecordNo(ByVal CRecordNo As Integer)

    Dim IncreaseCRecordNo As String = "UPDATE Counters SET Counter=@Counter WHERE Type='CRecordID'"
    Dim MyCommand As SqlCommand = New SqlCommand(IncreaseCRecordNo, myConnection)
```

```
    MyCommand.Parameters.Add(New SqlParameter("@Counter", SqlDbType.Int, 4))
    MyCommand.Parameters("@Counter").Value = CRecordNo + 1

    MyCommand.Connection.Open()
    MyCommand.ExecuteNonQuery()
    MyCommand.Connection.Close()

End Sub

Public Function CheckItemIDExist(ByVal ItemID As String) As Boolean

    Dim result As Boolean

    Dim strSQL As String = "SELECT * FROM Items WHERE ItemID= '" & ItemID & "'"
    Dim checkItem As SqlCommand = New SqlCommand(strSQL, myConnection)
    Dim myDataReader As SqlDataReader

    myConnection.Open()

    myDataReader = checkItem.ExecuteReader()
    If myDataReader.Read() = True Then
      result = True
    Else
      result = False
    End If

    Return result

End Function

Public Sub AddItem(ByVal CRecordNo As Integer, ByVal cartID As String, ByVal ItemID As String, ByVal Qty As Integer)

    Dim Add As String = "INSERT INTO Carts (CRecordNo,cartID,ItemID,Quantity) VALUES (@CRecordNo,@cartID,@ItemID,@Quantity)"
    Dim MyCommand As SqlCommand = New SqlCommand(Add, myConnection)

    MyCommand.Parameters.Add(New SqlParameter("@CRecordNo", SqlDbType.Int, 4))
    MyCommand.Parameters("@CRecordNo").Value = CRecordNo
```

```
    MyCommand.Parameters.Add(New SqlParameter("@cartID", SqlDbType.Char, 10))
    MyCommand.Parameters("@cartID").Value = cartID

    MyCommand.Parameters.Add(New SqlParameter("@ItemID", SqlDbType.Char, 6))
    MyCommand.Parameters("@ItemID").Value = ItemID

    MyCommand.Parameters.Add(New SqlParameter("@Quantity", SqlDbType.Int, 4))
    MyCommand.Parameters("@Quantity").Value = Qty

    MyCommand.Connection.Open()
    MyCommand.ExecuteNonQuery()
    MyCommand.Connection.Close()

End Sub

Public Function GetItemCount(ByVal cartID As String) As Integer

    Dim mycommand As SqlCommand = New SqlCommand("CartItemCount", myConnection)
    mycommand.CommandType = CommandType.StoredProcedure

    Dim parameterCartID As SqlParameter = New SqlParameter("@CartID", SqlDbType.Char, 10)
    parameterCartID.Value = cartID
    mycommand.Parameters.Add(parameterCartID)

    Dim parameterItemCount As SqlParameter = New SqlParameter("@ItemCount", SqlDbType.Int, 4)
    parameterItemCount.Direction = ParameterDirection.Output
    mycommand.Parameters.Add(parameterItemCount)

    mycommand.Connection.Open()
    mycommand.ExecuteNonQuery()
    mycommand.Connection.Close()

    Return CInt(parameterItemCount.Value)

End Function

Public Function GetTotal(ByVal CartID As String) As Decimal
```

```
    Dim mycommand As SqlCommand = New SqlCommand("CartTotal", myConnection)
    mycommand.CommandType = CommandType.StoredProcedure

    Dim parameterCartID As SqlParameter = New SqlParameter("@CartID", SqlDbType.Char, 10)
    parameterCartID.Value = CartID
    mycommand.Parameters.Add(parameterCartID)

    Dim parameterTotal As SqlParameter = New SqlParameter("@Total", SqlDbType.Money, 8)
    parameterTotal.Direction = ParameterDirection.Output
    mycommand.Parameters.Add(parameterTotal)

    mycommand.Connection.Open()
    mycommand.ExecuteNonQuery()
    mycommand.Connection.Close()

    If CStr(parameterTotal.Value) <> "" Then
      Return CType(parameterTotal.Value, Decimal)
    Else
      Return 0
    End If

End Function

Public Function GetTotal1(ByVal CartID As String) As Decimal

    Dim mycommand As SqlCommand = New SqlCommand("CartTotal", myConnection)
    mycommand.CommandType = CommandType.StoredProcedure

    Dim parameterCartID As SqlParameter = New SqlParameter("@CartID", SqlDbType.Char, 10)
    parameterCartID.Value = CartID
    mycommand.Parameters.Add(parameterCartID)

    Dim parameterTotal As SqlParameter = New SqlParameter("@Total", SqlDbType.Money, 8)
    parameterTotal.Direction = ParameterDirection.Output
    mycommand.Parameters.Add(parameterTotal)

    mycommand.Connection.Open()
```

```
    mycommand.ExecuteNonQuery()
    mycommand.Connection.Close()

    If (parameterTotal.Value) <> "" Then
      Return (parameterTotal.Value)
    Else
      Return 0
    End If

  End Function


  Public Function GetCartDetails(ByVal cartID As String) As SqlDataReader

    Dim mycommand As SqlCommand = New SqlCommand("CartList", myConnection)
    mycommand.CommandType = CommandType.StoredProcedure

    Dim parameterCartID As SqlParameter = New SqlParameter("@CartID", SqlDbType.Char, 10)
    parameterCartID.Value = cartID
    mycommand.Parameters.Add(parameterCartID)

    myConnection.Open()
    Dim result As SqlDataReader = mycommand.ExecuteReader(CommandBehavior.CloseConnection)
    Return result

  End Function

  Public Sub UpdateCart(ByVal CartID As String, ByVal ItemID As String, ByVal Quantity As Integer)

    If Quantity < 0 Then
      Throw New Exception("Quantity cannot be a negative number")
    Else

      Dim mycommand As SqlCommand = New SqlCommand("CartUpdate", myConnection)
      mycommand.CommandType = CommandType.StoredProcedure
```

```
        Dim parameterCartID As SqlParameter = New SqlParameter("@CartID", SqlDbType.Char, 10)
        parameterCartID.Value = CartID
        mycommand.Parameters.Add(parameterCartID)

        Dim parameterItemID As SqlParameter = New SqlParameter("@ItemID", SqlDbType.Char, 6)
        parameterItemID.Value = ItemID
        mycommand.Parameters.Add(parameterItemID)

        Dim parameterQuantity As SqlParameter = New SqlParameter("@Quantity", SqlDbType.Int, 4)
        parameterQuantity.Value = Quantity
        mycommand.Parameters.Add(parameterQuantity)

        mycommand.Connection.Open()
        mycommand.ExecuteNonQuery()
        mycommand.Connection.Close()
    End If

End Sub

Public Sub CartRemoveItem(ByVal CartID As String, ByVal ItemID As String)

    Dim mycommand As SqlCommand = New SqlCommand("CartRemoveItem", myConnection)
    mycommand.CommandType = CommandType.StoredProcedure

    Dim parameterCartID As SqlParameter = New SqlParameter("@CartID", SqlDbType.Char, 10)
    parameterCartID.Value = CartID
    mycommand.Parameters.Add(parameterCartID)

    Dim parameterItemID As SqlParameter = New SqlParameter("@ItemID", SqlDbType.Char, 6)
    parameterItemID.Value = ItemID
    mycommand.Parameters.Add(parameterItemID)

    mycommand.Connection.Open()
    mycommand.ExecuteNonQuery()
    mycommand.Connection.Close()

End Sub
```

```
Public Sub UpdateQuantity(ByVal ItemID As String, ByVal Quantity As Integer)

    Dim mycommand As SqlCommand = New SqlCommand("UpdateQuantity", myConnection)
    mycommand.CommandType = CommandType.StoredProcedure

    mycommand.Parameters.Add(New SqlParameter("@ItemID", SqlDbType.Char, 6))
    mycommand.Parameters("@ItemID").Value = ItemID

    mycommand.Parameters.Add(New SqlParameter("@Quantity", SqlDbType.Int, 4))
    mycommand.Parameters("@Quantity").Value = Quantity

    mycommand.Connection.Open()
    mycommand.ExecuteNonQuery()
    mycommand.Connection.Close()

End Sub

Public Function GetItemQuantity(ByVal ItemID As String) As Integer

    Dim mycommand As SqlCommand = New SqlCommand("GetItemQuantity", myConnection)
    mycommand.CommandType = CommandType.StoredProcedure

    mycommand.Parameters.Add(New SqlParameter("@ItemID", SqlDbType.Char, 6))
    mycommand.Parameters("@ItemID").Value = ItemID

    Dim parameterQuantity As SqlParameter = New SqlParameter("@Quantity", SqlDbType.Int, 4)
    parameterQuantity.Direction = ParameterDirection.Output
    mycommand.Parameters.Add(parameterQuantity)

    mycommand.Connection.Open()
    mycommand.ExecuteNonQuery()
    mycommand.Connection.Close()

    Return parameterQuantity.Value

End Function
```

```
    Public Function GetItems(ByVal cartID As String) As SqlDataReader

        ' Create Instance of Connection and Command Object
        Dim myConnection As SqlConnection = New SqlConnection(ConfigurationSettings.AppSettings("constring"))
        Dim myCommand As SqlCommand = New SqlCommand("CartList", myConnection)

        ' Mark the Command as a SPROC
        myCommand.CommandType = CommandType.StoredProcedure

        ' Add Parameters to SPROC
        Dim parameterCartID As SqlParameter = New SqlParameter("@CartID", SqlDbType.NVarChar, 50)
        parameterCartID.Value = cartID
        myCommand.Parameters.Add(parameterCartID)

        ' Execute the command
        myConnection.Open()
        Dim result As SqlDataReader = myCommand.ExecuteReader(CommandBehavior.CloseConnection)

        ' Return the datareader result
        Return result

    End Function


End Class

End Namespace
```