Faculty of Computer Science & Information Technology
University Of Malaya
50603 Kuala Lumpur
Malaysia

Project Title : Web-Based Business Planning System (Analysis and Forecast)

Supervisor : Prof. Madya Dr. P.Sellappan
Moderator : Cik Nazean Jomhari
Name : How Men Mee
Matrix No : WEK 990043

Dissertation submitted by in fulfillment of the requirements for the Degree of Bachelor of Computer Science
Submission Date 25 January 2002

TABLE OF CONTENTS

CHAPTER 4: SYSTEM DESIGN

CHAPTER 6: SYSTEM EVALUATION

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

CHAPTER 2: LITERATURE REVIEW

CHAPTER 4: SYSTEM DESIGN

CHAPTER 5: SYSTEM DEVELOPMENT AND TESTING

LIST OF TABLES

CHAPTER 2: LITERATURE REVIEW

CHAPTER 3: METHODOLOGY AND SYSTEM ANALYSIS

CHAPTER 4: SYSTEM DESIGN

ABSTRACT

Planning is an important activity. It requires making estimates about future needs and developments, which are difficult to predict and control. So, **Web-Based Business Planning System (BPS)** explains, more in detail, the fundamental of planning and how it is used in decision making, capital project feasibility studies, and planning the future of a business.

Web-Based Business Planning System (BPS) is a centralized channel on the web to allow user (manager or sales person) to retrieve or upload information through this system anytime and anyway as long as they connect to the Internet. This system allow user to view or update their sales data, do analysis or forecast and view the analysis and forecast report. **Web-Based Business Planning System (BPS)** is capable of improving the quality and outcome of analysis and forecast by ensuring that accurate, fast and timely information is made accessible to appropriate personnel at all level. In addition, **Web-Based Business Planning System (BPS)** inform the user about the report via PivotTable and PivotChart.

Methodology that I was selected to develop this **Web-Based Business Planning System (BPS)** is Waterfall model with prototyping. The flexible of this methodology made me easy to development the system and reduces the risk involved. The development tool that I go to use is Ms Visual InterDev, whereas the web server was Internet Information Server 5.0 with back-end database Ms SQL server 2000 and OLAP. Both servers were deployed on Windows 2000 Server. The system design chapter will illustrate how the entire system works.

Many problems were encountered during the process of development this system. Many approaches were used to solve the entire problem to perform a good **Web-Based Business Planning System (BPS).**

Besides, also discuss about the strength, limitation and the future enhancement of the system.

ACKNOWLEDGEMENT

Through the project development, many people had kindly given me a hand directly or indirectly. So, I would to take this opportunity to point out the invaluable advice, assistance and help of some particular individuals who had impacted my own work the most.

First and foremost, I would like to express my highest gratitude to my respected supervisor, Professor Madya Dr. P. Sellappan for his invaluable advice, generous guidance, encouragement, and support along the project.

Secondly, I would like to express my warmest appreciation to Cik Nazean Jomhari for sparing her precious time to be my project moderator. For special mention, I would like to express my sincere gratitude to my course mates, Lam Wei Li, Lim Lee Peng, Chan Chee Yong, Tey Ek Wei, Wong Kok Seng and others whom I have not mentioned here.

Last but not least, I wish to thank my beloved family for their unfaltering support to me thought the duration of my project development. I am especially indebted and grateful to all of them.

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Historically, the bulk of investment in corporate computing has been in systems that generate or capture data such as accounting, order processing, manufacturing and customer information. Besides this, many organizations are trying to invest suitable applications or technologies that can deliver additional value from this collected data.

This project is called **Web Based Business Planning System: Analysis and Forecasting (BPS)**. It is basically a tool that is very crucial in running a business. This planning system is a roadmap for a business, explaining where they are, where they want to go, and how they expect to get there. Good planning is key to a successful business.

This system allows manager and executives to analyze and diagnose the health of a business. A good **BPS** should give clear indication of business indexes in a fast and efficient manner.

1.2 PROJECT OVERVIEW

BPS is a system that helps an organization to plan for their future, decide the route of the business and organize the structure of the company. The areas embraced by **BPS** will functionally include analysis system, forecasting system and reporting system.

Due to the fact that this database provides a broad range of services, the information stored in the **BPS** database can be divided into two main groupings, one is data related to sales data and another one is activity-based data. The sales data include sales data by geographical, time, sales personnel, customer and product while the activity-based data include call data and non-call data.

This project is a web-based information system with a relational database, which designed based on the client-server architecture. After that, data inside the database will transform to OLAPS cubes. OLAPS cubes are contain data and will be used for calculate purpose. With OLAP applications can delivery fast analysis of multidimensional information to the user.

This system can give access to all type of analysis and forecast record. All the report is generate in PivotTable and PivotChart format. Hence, it would be created with compatible systems to support multiple users and help them to achieve effective information. The users for this system can be manager, executive and sales person of an organization.

Anyway, only manager and executives can access to the database component. In other words, only they have the authority to edit or update data in the database that is under their scope. . In other words, validation of the specific user's login id is needed to maintain the relevant data.

Furthermore this system is also realizing the concept of paperless office. The paperless office virtually eliminates spaces and folder for keeping the data with manual efforts.

1.3 PROBLEM DEFINITION

Now a day, all organizations have multidimensional data. Even thought a small organization would like to track sales by product, salesperson, geography, customer and time. These organizations have long sought tools to access, navigate and analyze multidimensional data in an easy, fast and natural way.

Sometimes we hear a manager say that he cannot do a good job of planning because his company or industry is to volatile or there are too many uncertainties or he doesn't have time. Frequently, because of such arguments, no planning at all is done. Postponing or ignoring the need for analysis and planning, including financial planning can substantially increase vulnerability and lead to disaster.

Problems that always faced by the organization are:

- Management reports inflexible
- Too difficult to get new information from other divisional office
- Spend too much time reporting what happened in the past
- Not enough time finding ways to improve what will happen in the future

1.4 AIMS AND OBJECTIVES

For many businesses, it has always been a problem converting operational data into information that is useful for management. They faced many problems while presenting these data to the user. They cannot present these data in an effective, portable and user-friendly manner.

The objectives of this **BPS** are:

a) **High speed and efficiency**

Design a system that can present data in a high speed and efficiency way for analyze purpose when data amounts to million of records. The data is matched against parameters that are relevant to the business needs. To perform this function, the system has to convert operational data (in RDBMS format) into a data warehouse.

b) Fast analysis

Using OLAPS application to delivery fast analyze of multidimensional information to user. Their role is to aid in corporate reporting and decision support.

c) Web interface to view data

Create a web interface to view data. This system able user to disseminate information on a 'anytime', 'anywhere' basis because many analysts need analyze the data while disconnected from the corporate network.

1.5 PROJECT SCOPE

This project can be broken into three main sections (see Figure 1-1):

a) Data collection and transformation

b) Parameter setting

c) Reporting and decision support

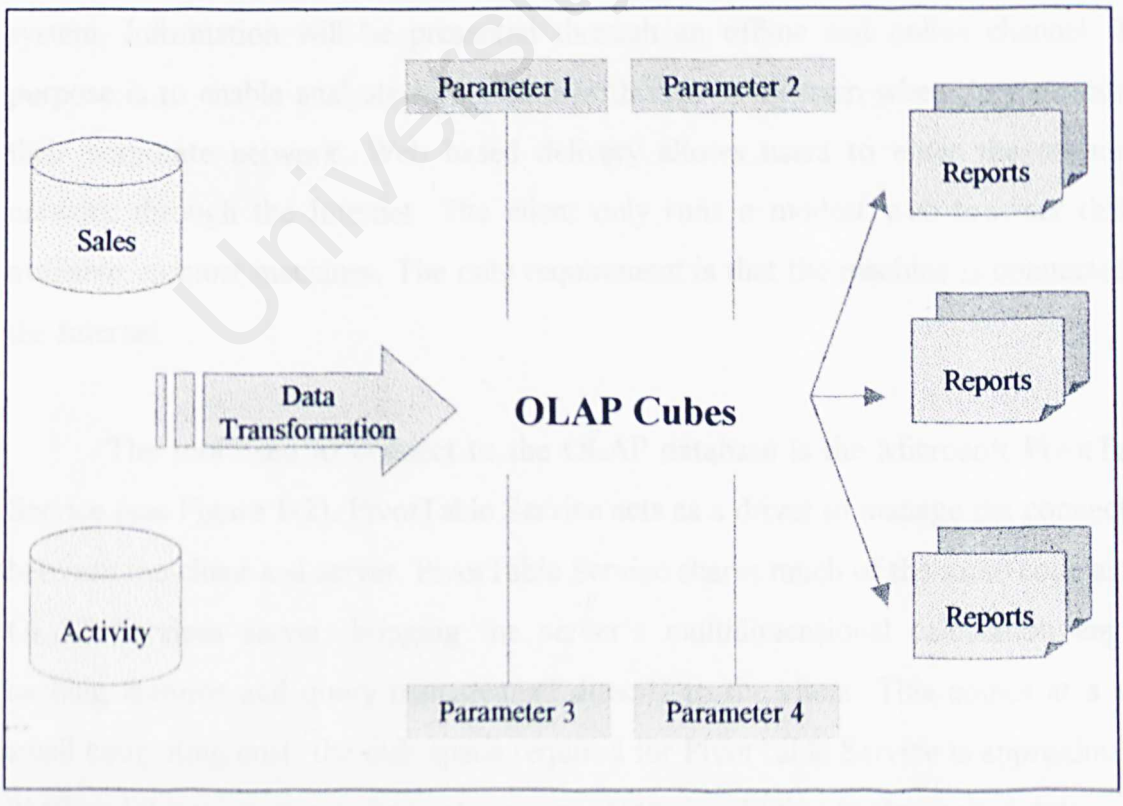


Figure 1-1: Project scope

DATA COLLECTION AND TRANSFORMATION

Operational data is assumed to reside in relational database format. The type of data can be broken up into 2 types: sales and activity. The activity-based data includes Call and Non-Call data. Data is to be transformed to the OLAP cubes.

The Cubes have to be designed to take information from the RDBMS. The fields and aggregations are to be designed to fulfill business needs. The design of the cubes is yet to be finalized.

PARAMETER SETTING

Users can set parameters so that data can be compared with these parameters for indices of growth and performance. The parameters can be zone targets, product targets, and time targets. They can also include parameters like call quota. These parameters affect the figures that are used in analysis later on.

REPORTING AND DECISION SUPPORT

The main portion of the system lies in the reporting and decision support system. Information will be presented through an offline and online channel. The purpose is to enable analysts to continue with their work even when they are out of their corporate network. Web based delivery allows users to enter the corporate network through the Internet. The client only runs a modest web browser that is available on most machines. The only requirement is that the machine is connected to the Internet.

The tool used to connect to the OLAP database is the Microsoft PivotTable Service (see Figure 1-2). PivotTable Service acts as a driver to manage the connection between the client and server. PivotTable Service shares much of the same code as the OLAP Services server, bringing the server's multidimensional calculation engine, caching features and query management directly to the client. This comes at a very small computing cost: the disk space required for PivotTable Service is approximately 2MB and the memory requirements are only 500K in addition to the cached data.

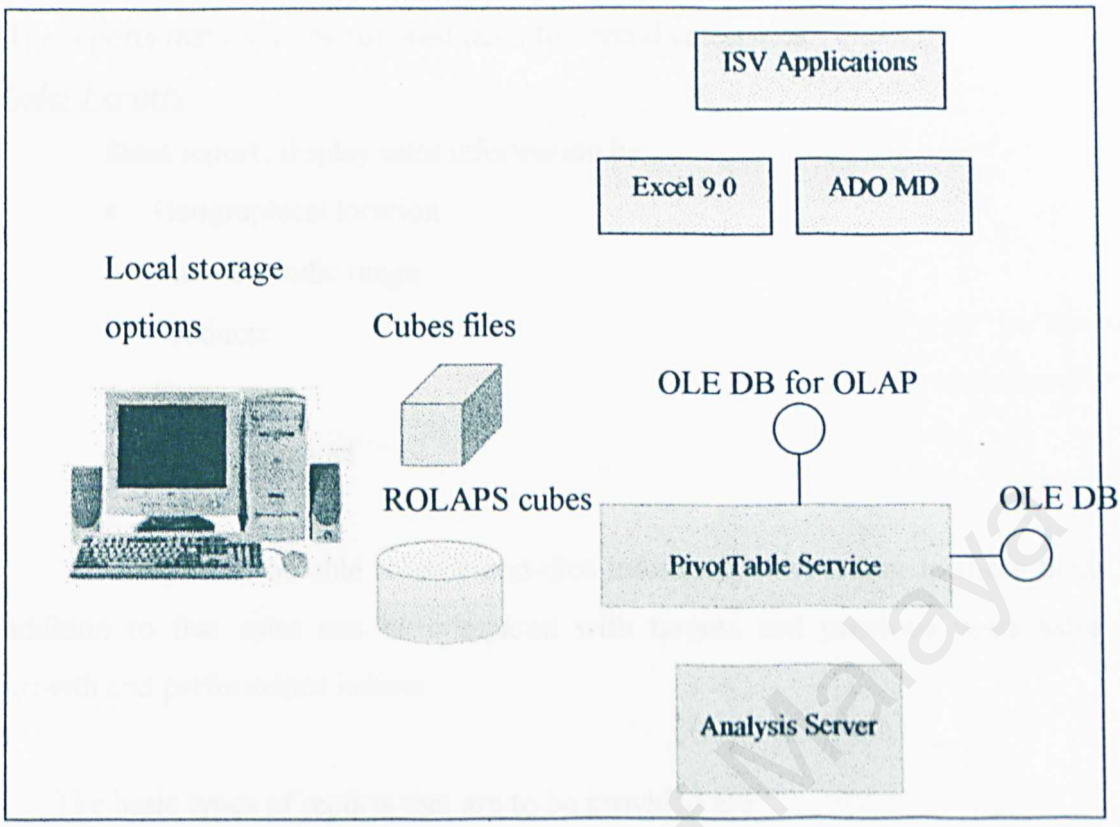


Figure 1-2: PivotTable Service

PivotTable Service also provides the mechanism for disconnected usage. Portions of cubes defined and accessed from a server can be saved on the client for later access when disconnected from the network. This functionality can fulfill the offline requirement of the system. PivotTable Service provides the connectivity for Wed-based applications. The ActiveX Data Objects (ADO) extension provides multidimensional data access. ADO/MD is the tool for utilizing the full functionality of OLAP Service.

The wed interface will be supported by ActiveX controls called Office Wed components. They will provide basic OLAP browsing and charting capabilities. The controls available in this set of components are:

- PivotTable control
- Spreadsheet control
- PivotChart control

The reports that are to be supplied fall into several categories:

Sales Reports

Sales reports display sales information by

- Geographical location
- Time-Periodic range
- Products
- Customers
- Sales personnel

Users must be able to slice-and-dice information according to these fields. In addition to that sales can be compared with targets and previous years sales for growth and performance indices.

The basic types of reports that are to be provided are

a) PivotTable of Sales
b) PivotTable of Sales versus targets and historical data
c) Chart of Sales versus Time
d) Chart of Sales by product, customer, location or personnel
e) Charts of Sales Comparison with Time

Activity based reports

In addition to sales data, operational data includes data from Call reports, non-Call Reports and Opportunity reports. These data can be used to correlate with sales data for indications of effectiveness and performance of sales force.

The basic type of calculation and reports that are to be provided are

a) Cost and revenue of customer visit by personnel and period
b) Effective revenue of visit
c) Success rate of visits
d) Summary Chart of revenue/ visit versus time

e) Chart of revenue/ visit versus time by personnel

f) Effective revenue per working hour

g) Average revenue per personnel

Predictive reports

Analysts need to do forecasts based on current sales or activity so that they can plan and budget for the future. The system should come up with some predictive algorithms that can help them in this area.

The based types of reports that are to be provided are:

a) Curve-fitting for sales with time

b) Sales prediction for increased call rate

c) Sales prediction for increased non-call rate

1.6 IMPORTANT OR SIGNIFICANCE OF A PROJECT

This project is to create a system that can analyze sales and operational data that is relevant to the business. Data will include sales data, call report and non-call report data. The focus of this project will be initially on a Sales Force Automation System.

The significances of the **BPS** system are:

a) Data is in Data mart or Data warehouse for fast retrieval

All the operational data will keep track in the Data mart or Data warehouse no matter how large is the data. Data will store systematically to reduce the problems like data redundancy and inconsistency. So that data is available anytime.

b) Web interface to view data

As mentioned early, web interface allows disconnected users to enter corporate network for continue their work thought the Internet. For example, a sales manager is viewing a revenue summary for a particular region while visiting a regional office.

c) Analyses data into chart or graph format

After analysts analyze all the data, they will present the information into chart or graph format. This will make users more understanding the analyses result.

d) Screen to allow user to add in planning data

At this part, users have to fill up a simple form. Users have to insert all the relate information such as region, parameters and others. Information stated above will be used to make comparison purpose.

1.7 EXPECTED OUTCOME

A business planning system should help you not only to prepare for the future but also learn from the past by answering the following questions:

- **How do we develop consistent and actionable business plans for all field force levels?**

  By communicating the company's goals both quantitatively and qualitatively, field managers at every level will be able to better align their efforts with company's overall objectives.

  Open communication amongst representatives and managers will help to identify and share best practices throughout your organization.

- **How do we expand field force knowledge of market potential and develop strategies for reaching that potential?**
  Managers should have the information & training to analyze their performance relative to their market at both a local and national level.

- **How do we track planned and actual sales achievement in order to continually assess and refine strategies?**
  Allow managers to understand the impact of their strategies as they are implemented in the field.

  Quarterly tracking allows all members of the organization to know how well they are doing relative to their individual objectives.

1.8 PROJECT LIMITATION

This **BPS** has certain limitations as below:

- Firstly, this system is not functioning without the accessibility to the Internet.
- Second, maintenance for the database component is quite difficult and costly. So, the recovery works are difficult to implement in this system.

1.9 SCHEDULE

A project schedule that consists of the whole development's activities is carefully planned out to achieve a systematic progress and ensure on-time delivery of the product. It is important to have a project schedule as it acts as a time management and control to the developer making sure he is in route of the direction of the project.

The project schedule is as shown in Figure 1-3.

ID	Task Name	Start	Finish	Duration
1	Resource searching & reading	6/14/2001	7/4/2001	15d
2	Literature review	6/14/2001	8/1/2001	35d
3	System analysis	8/2/2001	8/29/2001	20d
4	Proposal writing & finalization	8/30/2001	9/18/2001	14d
5	System Design	8/30/2001	11/21/2001	60d
6	Module implementation	11/22/2001	12/26/2001	25d
7	System integration & testing	12/27/2001	1/23/2002	20d
8	Reporting writing	1/24/2002	1/29/2002	4d

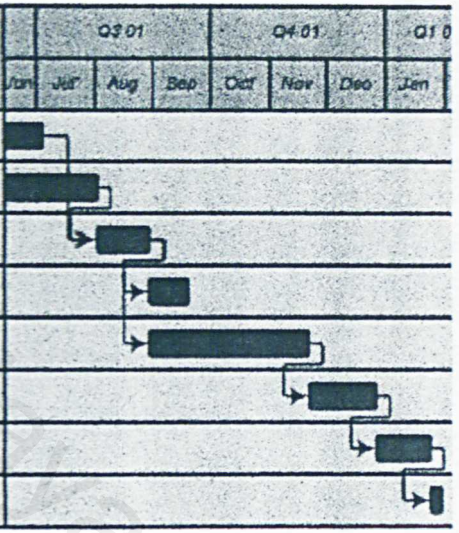


Figure 1-3: Project Schedule

1.10 DEVELOPMENT STRATEGY

Development strategy is an idea of developing an initial implementation; exposing this to user command and refining this through many versions until an adequate system has been develop. In developing this project, a sequence of steps in followed to accomplish the success of the project. These steps can be divided into five sequential phases, although in reality the phase are interrelated and often are accomplished simultaneous.

The five phases are:

- Requirements
- System design
- Coding
- Testing
- Operations [1]

These processes are important because they impose consistency and structure on the whole project development. These are many process models available in the software engineering. There are that has been chosen for this project is the waterfall model with prototyping, where the stages depicted as cascading from one to another. One development stage has to be complete before next begins.

Figure 1-4 shows the stage of the waterfall model with prototyping. [2]

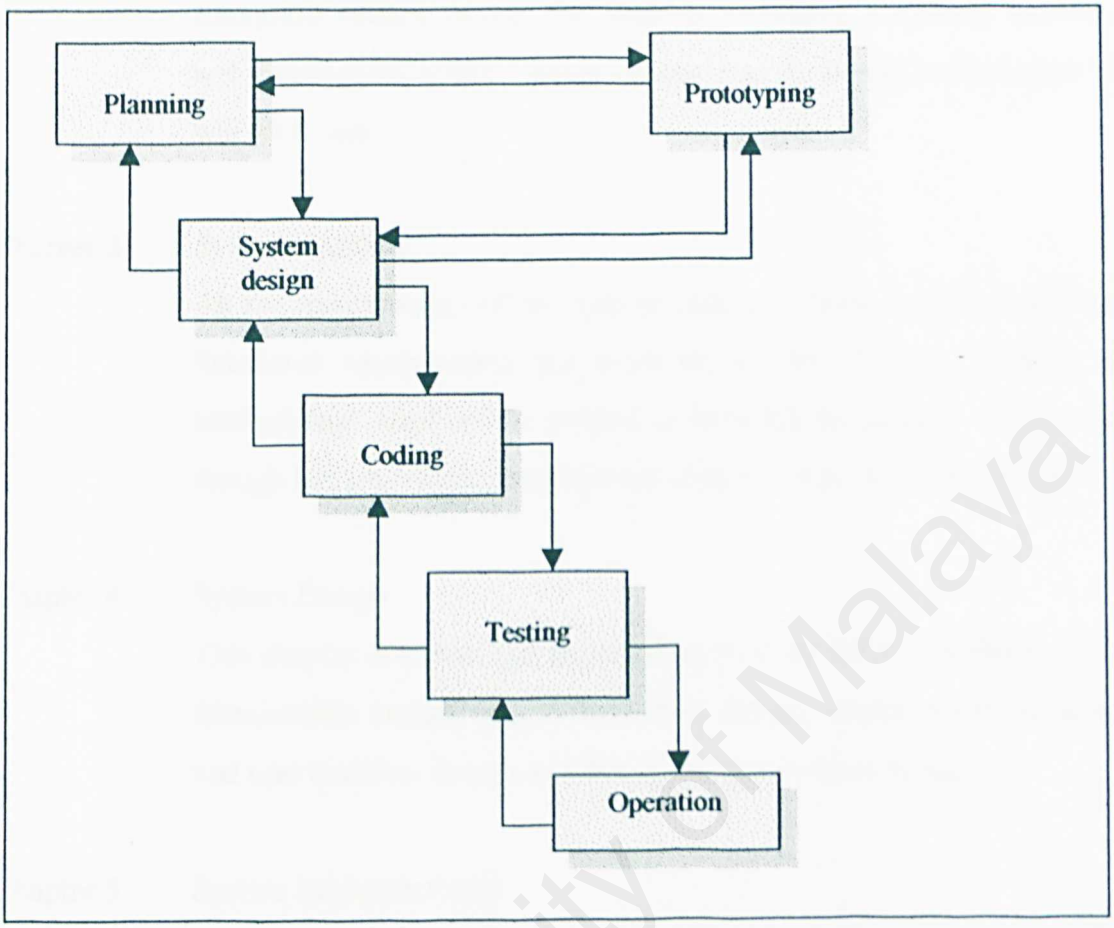


Figure 1-4: Waterfall model with prototyping

1.11 REPORT REVIEW

The purpose of this report is to document all the researches and phases of developing the **BPS**. In general, this report can be subdivided into 6 chapters.

Chapter 1: Introduction

The report begins with a simple project introduction. It contains an overview of the project, the aims and objectives, project scope, important or significance, development strategy and schedule.

Chapter 2: Literature Review

Literature review covers the findings regarding statistical technique, web application, client / server computing, tools and technologies that will go to use.

Chapter 3: System Analysis

All the requirements of the system including both functional and non-functional requirements are explored in this chapter. Besides, the methodology used in the project is included in chapter. It also steps though features of the development tools to be used for this project.

Chapter 4: System Design

This chapter describes the logical design of the **BPS**. It includes system functionality design, system data flow design, system modules design, and user interface design and data-entry procedures design.

Chapter 5: System Implementation

This chapter illustrates the coding and testing process of the **BPS**. It describes tools to use, way of implementing system, codes writing and way of carrying out each implementing task. Besides, ways used to test the system are also included. This includes unit testing, module testing and system testing.

Chapter 6: System Evaluation

This chapter summarized the entire development process of **BPS**. Problems arose during the implementation stage and the relevant resolutions are discussed. Besides, system's strengths and limitations are also included in the chapter.

CHAPTER SUMMARY

In this chapter, I generally introduce the system that I will go to develop. I will describe what is **Business Planning System**, objective, significance, project scope, schedule and others.

Besides, I also compare existing system with my system. The main different between my system and existing system are my system will use OLAP for analytical technology and presenting information in PivotTable and PivotChart. During the comparison, I encounter some of the problem that happens in the existing system. I try to solve this problem and make sure will not happen in my system.

The expected outcome for my system is creating a business planning system in user-friendly manner. That will help user easy to maintain the system.

LIST OF REFERENCE

[1] Kenneth E. Kendall and Julie E. Kendall, "System Analysis And Design", 2nd Edition, Prentice Hall: 1992.

[2] Shari Lawrence Pfleeger, "Software Engineering, Theory and Practice", Prentice Hall Inc: 1998.

CHAPTER 2

LITERATURE REVIEW

2.1 WHAT IS LITERATURE REVIEW

A literature review is to sufficiently equip the developer with some knowledge of the strengths and limitations of several development tools. If helps the developer to know some of the existing features offered by a similar system and to get the better understanding on the development tools and techniques that can be used in the development process. This is a real challenge before a final decision can be reached to start developing the system. So that, the developer will know which are right tools should be choose to develop the system.

In the process of developing **BPS,** this part of the research has been done to understand various new concepts, which especially focus on the information and information system. A research also has been carried out to compare the current existing system with the **BPS**, which will be developed. New features are added in **BPS**.

Review of literature is a background study about the knowledge and information gained to develop this project. It enables the developer to get more knowledge on the development methodologies used in the project and have an overview of how to improve the weakness of the current system and fulfill the requirement needed.

2.2 COMPARISON WITH EXISTING SYSTEM

In the market now, there has many systems that are similar with **BPS** such as Cartesis Performance Analyzer, One Page Planning and Performance, Comshare and others. All these systems are help organization to generate financial or budget report into an effective written document.

It is quite difficult to know what are the tools and technologies that a system using. All these systems are just advertise how they will help an organization to do business plan and never mention what the tools and technologies are going to use.

Anyway generally, I know that all of them are using the different tools and technologies. Some are using Oracle for database maintenance, some are using Open Database Connectivity for database connection and some are using Common Gateway Interface or HyperText Transfer Protocol for web application programming. The main different between my system and existing system are my system will use OLAP for analytical technology and presenting data in PivotTable and PivotChart format.

2.3 PROJECT BACKGROUND STUDY

2.31 WHAT IS BUSINESS PLANNING SYSTEM

A business plan is a road map for your business, explaining where you are, where you want to go, and how you expect to get there. Good planning is key to a successful business.

To meet the challenges of a changing marketplace, managers at each level of the sales organization must think more strategically, while delegating the tactical decisions down to the lowest practical level. Field sales managers must develop concise plans on how they will achieve the goals established by their company. Where multiple sales groups exist, these plans must also address how to coordinate the efforts of representatives who share responsibility for the same geography and target audiences.

When planning is an integral part of the sales organization, the business planning system can be used to facilitate sharing of best practices, coaching of field sales managers, and assessing strategies and tactics.

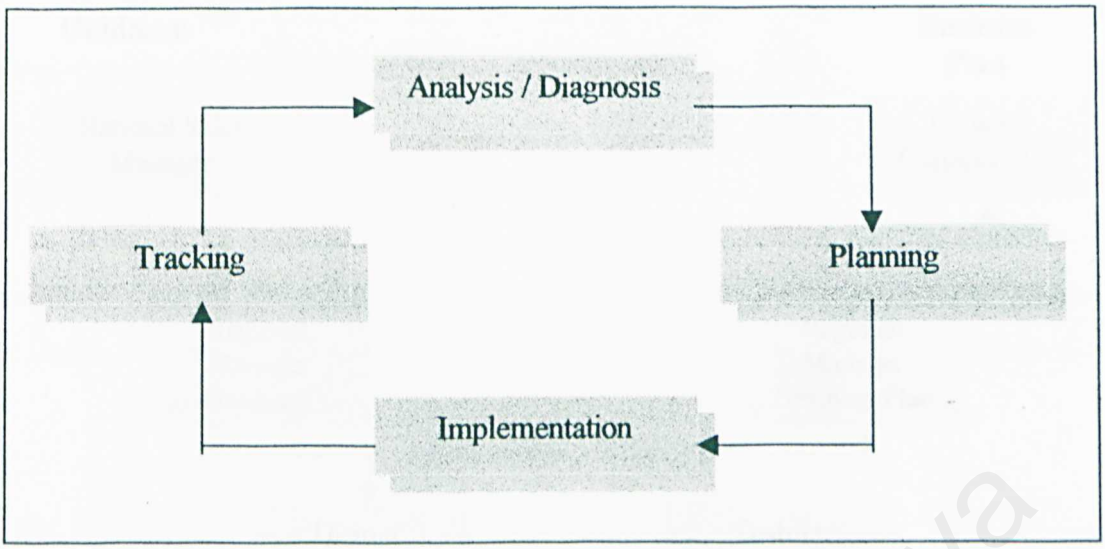


Figure 2-1: Planning cycle

The business planning system provides a structured format (see Figure 2-1) for fulfilling this planning cycle at all levels of the organization. By supporting organization in the analysis / diagnosis and planning stages of this cycle, the business planning system helps organization meet the challenges of, and understand the opportunities in, their market environment. This unique system also provides tracking information, allowing continuous assessment and improvement of strategies.

The business planning system facilitates two-way communication (see Figure 2-2) through a software-supported process that enables easy and effective transfer of information. Goals and objectives are communicated to field managers from the highest levels of the organization prior to the development of field business plans. This focuses the field on preparing plans that meet national objectives. The bottom up approach to creating business plans allows managers to take field insights into account prior to planning. Simultaneously, this process empowers representatives and helps them to assess the true impact of their sales efforts. [1]

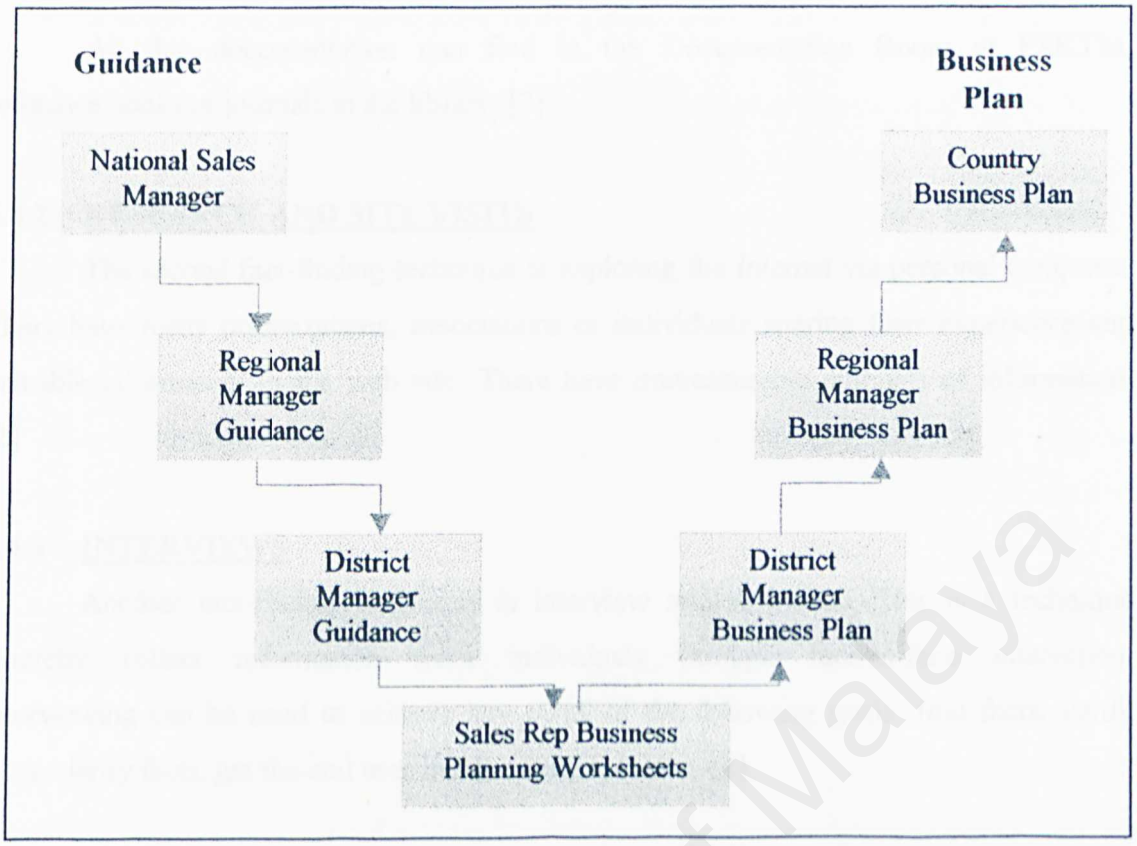


Figure 2-2: Two-way communication

2.4 TYPES OF ITEMS REVIEWED

2.4.1 EXISTING DOCUMENTATION

The first fact-finding technique is by studying the existing documentation. This will give more information and understandable about the system that will going to design. First, is to collect and review document that is describe the problem and function of business planning system. These include:

- Information systems projects requests – past and present
- Samples of manual and computerized databases and reports
- The company's mission statement and strategic plan.

All this documentation can find in the Documentation Room at FSKTM, reference books or journals in the library. [2]

2.4.2 RESEARCH AND SITE VISITS

The second fact-finding technique is exploring the Internet via personal computer. There have many organizations, associations or individuals sharing their experience and valuable information in the web site. There have immeasurable amounts of information. [2]

2.4.3 INTERVIEWS

Another fact-finding technique is interview related person. This is a technique whereby collect information from individuals through face-to-face interaction. Interviewing can be used to achieve any of all of the following goals: find facts, verify facts, clarify facts, get the end user involves and opinions. [2]

2.5 STATISTICAL TECHNIQUE

2.5.1 QUALITATIVE TECHNIQUE

The qualitative techniques provide a framework with which quantitative techniques (including forms of quantitative analyses, such as decision trees and linear programming) are brought to bear on a particular problem. The objective of the quantitative techniques is to bring together in a logical, unbiased and systematic way, all the information and judgment and rating schemed to turn qualitative information into quantitative estimates. Quantitative techniques are most commonly used in forecasting something about which amount, type and quality of historical data are limited.

The commonly use quantitative techniques include Delphi method, Panel of consensus and historical analogies. [3]

2.5.1.1 Delphi method

The point of Delphi Method is to obtain the consensus of a panel of experts about a given problem or issue. The method attempts to avoid the possible negative aspect associated with group dynamics (e.g. suppression of minority opinions, domination by strong individuals who may be incorrect, unwillingness to change public positions and bandwagon effects). Therefore, instead of bringing these experts together in a debating forum, the Delphi Method relies on the distribution of questionnaires to the experts with an admonishment not to discuss the problem among them. They may not know who are the other members of the panel are and they are not provided with individual opinions or estimates.

The initial questionnaires may be used to state the problem and to obtain preliminary estimates and reason or assumptions behind them. The responses are then summarized and feedback to the panel members with widely differing estimates are asked to review the responses and have appropriate, revise their estimates. Through several iterations it may be possible to refine the differences between experts to accept the majority opinion. If an expert feels strongly about another position and can articulate it persuasively, the method will provide a range of opinion, which may be desible in conditions of high uncertainty. [3]

2.5.1.2 Panel of Consensus

A panel of consensus can be as simple as having teachers sit around a conference table and decide collectively on the forecast for a type of question. Bringing teachers from various schools together increase the amount of relevant information available to the decision maker. A further advantage of the approach is the speed with which forecast can be obtained particularly in the absence of complete historical data. This advantage may be offset by the lack of accountability for the forecast.

The typical problems of group dynamics will also become apparent here and will be compounded by the relative rank of the teachers. Unfortunately, the person with the best insight may not carry sufficient weight to sway the whole group decision. [3]

2.5.1.3 Historical Analogies

History may be a reasonable guide in situations such as the introduction of a new syllabus. This concept is that, the expert analyses the similar situation to the newly introduction syllabus and makes a forecasting based on it.

2.5.2 QUANTITATIVE TECHNIQUES

If appropriate and sufficient data are available, then quantitative projection technique can be employed such quantitative technique can be further classified into categories:

- **Probabilistic (stochastic) techniques**
  These techniques focus entirely on patterns, pattern changes and disturbances caused by random influences. These techniques include frequency distribution, moving-average model, exponential smoothing model, time series decomposition, regression models and Box-Jenkins models.

- **Deterministic (causal) techniques**
  These techniques incorporate the identification and explicit determination of relationship between the factor to be forecast and other influencing factors. There include economic models and leading indications.

Within the listing of probabilistic projected technique, there are essentially two approaches. The first approach is the best illustrated by the time series decomposition technique. The primary assumption on which this methodology is based is that the data can be decomposed into several in observable components such as trend, seasonally, cycle and irregularity and that the components can be analyzed and projected into the future on an individual basis. The forecast is then merely the combination of the projections of the components.

A second approach is associated with the Box-Jenkins and econometric time series modeling methodologies. Their theoretical foundations ate grounded primarily in statistical concepts and do not assume that the data are represented by the superposition of separate components. Rather, the data have an overall representation in which the components are not separately identifiable or specified.

There is often a further distinction made between tine series and econometric methods: although they have strong similarities in their mathematical representation, they differ vaster in their estimation methodologies. [3]

The application of the statistical forecasting methods included:

a) Joint Forecast of U.S Marital Fertility, Nuptiality, Births and Marriage Using Time Series Model [4]

b) Methods for National Population Forecasts: A Review [5]

c) The Use of Leading Indicators for Forecasting and Analysis: International Interdependence of Business Cycle in Manufacturing Industry [6]

d) Sales Forecasting Using Multi-equation Transfer Models [7]

The quantitative techniques (both probabilistic and deterministic technique) are discussed in the following section:

2.5.2.1 Moving-average model

To implement the moving averages method, first choose a span, the number of terms I each moving average. The role of the span is important. If the span is large – say, 12 months – then many observations go into each average, and extreme values have relatively little effect on the forecasts. The resulting series of forecasts will be much smoother than the original series. In contrast, if the span is small – say, 3 months – then extreme observations have a larger effect on the forecasts, and the forecast series will be much less smooth. If the span is 1, there is no smoothing effect at all.

What span should we use? This requires some judgment. If we believe the ups and downs in the series are random noise, then we don't want future forecasts to react too quickly to these ups and downs, and we should use a relatively large span. But if we want to track every little zigzag – under the belief that each up or down is predictable – then we should use a smaller span. [8]

2.5.2.2 Exponential Smoothing Model

Exponential smoothing is a method that bases its forecasts on a weighted average of past observations, with more weight put on the more recent observations, and it requires very little data storage. This is particularly true for companies that routinely make forecasts of hundreds or even thousands of items. In addition, it is not difficult for most business people to understand, at least conceptually. Therefore, this method finds widespread use in the business world, particularly when frequent and automatic forecasts of many items are required.

There are many versions of exponential smoothing:

a) Simple exponential smoothing

Simple exponential smoothing has a single smoothing constant denoted by α and the level L_t of the series at time t. The method is defined by the following two equations, where F_{t+k} is the forecast of Y_{t+k} made at time t:

$$L_t = \alpha Y_t + (1- \alpha) L_{t-1} \qquad (2\text{-}1)$$

$$F_{t+k} = L_t \qquad (2\text{-}2)$$

Equation (2-1) shows how to update the estimate of the level. It is a weighted average of the current observation, Y_t, and the previous level, L_{t-1}, with respective weights α and 1- α. Equation (2-2) shows how forecasts are made. It says that the k-period-ahead forecast, F_{t+k}, made of Y_{t+k} in period t is the most recently estimated level, L_t. This is the same for any value of k>1. [8]

b) Holt's Model for Trend

In addition to the level of the series, L_t, Holt's method includes a trend term, T_t, and a corresponding smoothing constant β. The interpretation of L_t is exactly as before. The interpretation of T_t is that it represents an estimate of the change in the series from one period to the next. The equations for Holt's model are as follows:

$$L_t = \alpha Y_t + (1- \alpha)(L_{t-1} + T_{t-1}) \quad (2\text{-}3)$$

$$Tt = \beta(L_t - L_{t-1}) + (1 - \beta)\, T_{t-1} \quad (2\text{-}4)$$

$$F_{t+k} = L_t + kT_t \quad (2\text{-}5)$$

Equation (2-3) says that the updated level is a weighted average of the current observation and the previous level plus the estimated change. Equation (2-4) says that the updated trend term is a weighted average of the between two consecutive levels and the previous trend term. Equation (2-5) says that the k-period-ahead forecast made in period t is the estimated level plus k times the estimated change per period. [8]

2.6 WEB APPLICATION

Open Internet standards have changed the architecture of distributed computing forever. The basic language of the web, HTML, has become a popular language for representing the elements of the user interface cross-platform support for scripting languages and Java applets makes if possible to combine dynamic elements with static text for a more interactive user experience. Web technologies are used by organization to:

- Create an environment for component- based development
- Enable distribution of application throughout an enterprise
- Create and customize new application quickly
- Update databases remotely by using an ordinary web browser
- Add transaction processing to web applications
- Manage resources and enable remote system administration
- Provide business to business information sharing

2.7 CLIENT / SERVER COMPUTING

Before exploring into the details of building a Web-based application, it might be helpful to review the architecture model of the web, and the roles of the browser and server in that model.

Briefly, client / server is a computational architecture that involves client processes request service from server processes. In fact, client / server computing is the logical extension of modular programming. Modular programming has as its fundamental assumption that separation of a large piece of software into its constituent parts ("modules") creates the possibility for easier development and better maintainability. Client / server computing takes this a step farther by recognizing that those modules need not all be executed within the same memory space.

With this architecture, the calling module becomes the "Client" (that which requests a services) and the called module becomes the "server" (that which provides the services). The logical extension of this is to have clients and servers running on the appropriate hardware and software platforms for their functions. [9]

2.7.1 CHARACTERISTICS OF CLIENT / SERVER ARCHITECTURE

The basic characteristics of Client / Server architecture are:

a) Combination of a client or front-end portion that interacts with the user, and a server or back-end portion that interacts with the **shared resource**. The client process contains solution specific logic and provides the interface between the user and the rest of the application system. The server process acts as a software engine that manages shared resources such as databases, printers, modems, or high-powered processors.

b) The front-end task and back-end task have fundamentally different requirement for computing resources such as processor speeds, memory, disk speeds and capacities and input / output devices.

c) The environment is typically heterogeneous and **multivendor**. The hardware platform and operating system of client and server are not usually the same. Client and server processes communicate though a well-defined set of standard application program interfaces (API's) and (RPC's).

d) An important characteristic of Client / Server system id scalability. They can be scaled horizontally or vertically. Horizontal scaling means adding or removing client workstations with only a slight performance impact. Vertical scaling means migrating to a larger and faster server machine or multiserver. [10]

2.7.2 TWO-TIER ARCHITECTURE

A two-tier architecture (see Figure 2-3) is where a client talks directly to a server with no intervening server. It is typically used in small environments (less than 50 users).

Early two-tier (Client / Server) application was developed to access larger databases and manipulate the data with the user interface into the client application. The server's task was simply to process as many requests for data storage and retrieve as possible.

Two-tier application perform many of the functions of stand-alone system: They present a user interface, gather and process user input, perform the request processing and report the status of the request. This sequence of commands can be repeated as many times as necessary. Because servers provide only access to the data, the client uses its local resources to perform most of the processing.

One major advantage of this model was that by allowing multiple users simultaneously access the same application data, updates from one computer were instantly made available to all computers that had access to the server. However, this approach will usually result in an ineffective system, as the server becomes overwhelmed as the number of clients increased. To properly scale to hundred or thousands of users, it is usually necessary to move to a three-tier architecture.

A two-tier Client / Server modes has several critical limitations:

a) **Not scalable**

The inability of a two-tier approach to grow beyond the physical boundaries of a client machine and a server machine prevents this model from being scalable.

b) **Unmanageable**

Because we cannot encapsulate business rules and deploy them centrally, sharing common process and reusing our work is different at best.

c) **Poor performance**

The binding of the graphical interface to the data source consumes major resources on the client machine, which results in poor performance and unfortunately, unhappy clients. [11]

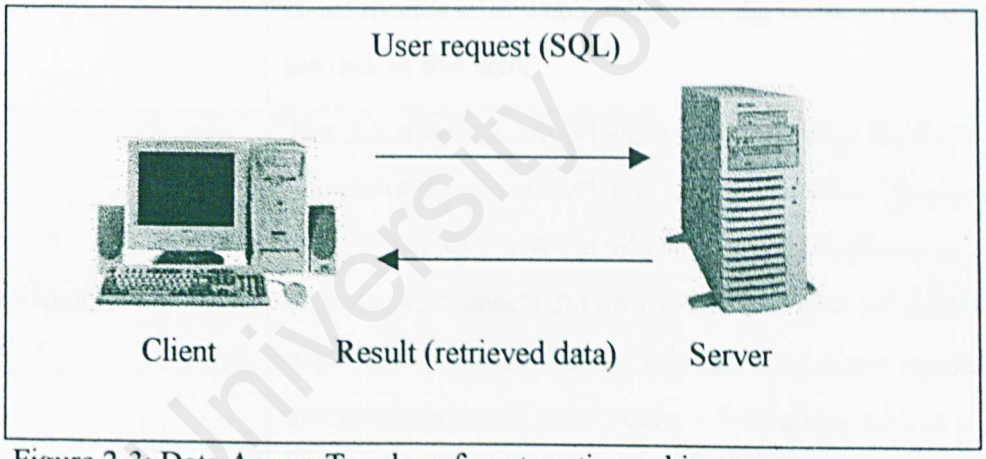


Figure 2-3: Data Access Topology for a two-tier architecture

2.7.3 THREE-TIER ARCHITECTURE

A three-tier architecture is a multitier application also known as n-tier architecture. It introduces a server (or an 'agent') between the client and the server, the business logic. The role of the agent is manifold.

It can provide translation services (as in adapting a legacy application on a mainframe to a Client / Server environment), metering services (as in acting as a transaction monitor to limit the number of simultaneous request to a given server), or intelligent agent services (as in mapping a request to a number of different servers, gather the results, and returning a single response to the client).

Most system will perform the following three main tasks, which correspond to three or layer of the n-tier model (see Table 2-1):

TASK	DESCRIPTION
User interface and navigation	Labeled Tier 1 in the following graphic, this layer comprises the entire user experience. Not only does this layer provide a graphical interface so that users can interact with the application, input data and view the result once the client receive it. In web application, the browser performs the task of this layer.
Business logic	Tier 2, between the interface and data services layers, is the domain of the distributed application developer. Business logic, which captures the rules that govern application processing, connects the user at one end with the data at the other. The functions that the rules govern closely mimic everybody business tasks, and can be a single task or a series of tasks.
Data service	Shown as Tier 3 in the following graphic, data services are provided by a structured (SQL, Oracle database) or unstructured (Microsoft Exchange, Microsoft Message Queuing) data store, which manages and provides access to the application data. A single application may enlist the services of one or more data stores.

Table 2-1: Three-tier architecture tasks

This model requires much more analysis and design up front, but greatly reduces maintenance costs and increases functional flexibility in the long run. The Figure 2-4 depicts the Microsoft technologies that service the various tiers in the new system design. [12]

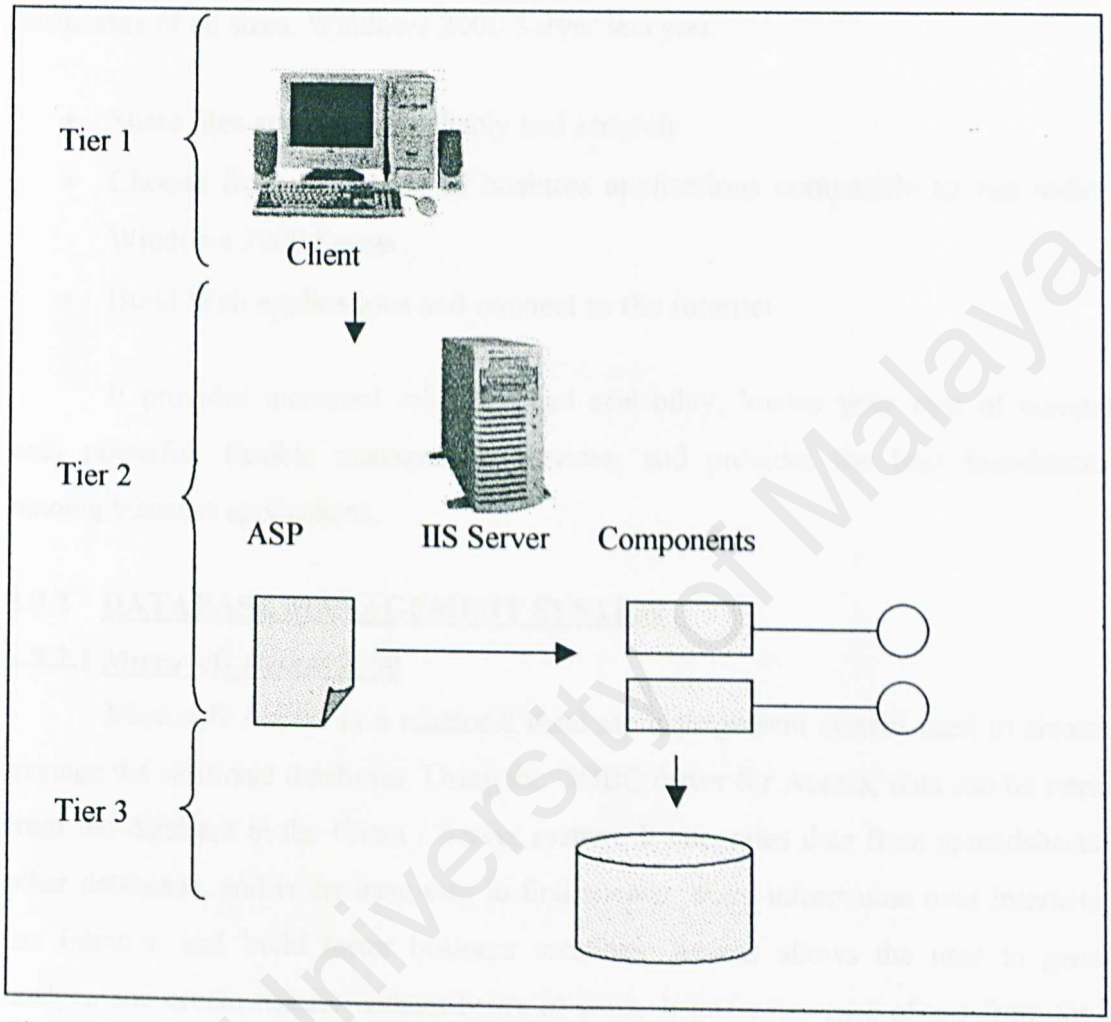


Figure 2-4: Three-tier architecture diagram

2.8 TOOLS AND TECHNOLOGIES CONSIDERATION

2.8.1 DEVELOPMENT PLATFORM

2.8.1.1 Window 2000 Server

Windows 2000 Server is the multipurpose network operating system for businesses of all sizes. Windows 2000 Server lets you:

- Share files and printers reliably and securely.
- Choose from thousands of business applications compatible to run today on Windows 2000 Server.
- Build Web applications and connect to the Internet.

It provides increased reliability and scalability, lowers your cost of computing with powerful, flexible management services, and provides the best foundation for running business applications.

2.8.2 DATABASE MANAGEMENT SYSTEM

2.8.2.1 Microsoft Access 2000

Microsoft Access in a relational database management system used to create and manage the relational databases. Using the ODBC driver for Access, data can be retrieved from the database in the Client / Server system. It integrates data from spreadsheets and other databases, and is the easy way to find answer, share information over Internets and the Intranet, and build faster business solutions. Access allows the user to generate, analyze and create reports without hours of work. It integrates ease of use from the data entry point to printing in HTML.

Access offers the following features to make designing and using a database easier and make the databases more powerful.

- Database Wizard

  The Database Wizard creates a wide variety of complete databases for tasks ranging from household inventory to event management. Users can then add their own data.

- Database splitter Wizard

  The front-end / back-end application is an application consisting of two database files. The back-end database file contains the tables. The front-end database file contains all other database objects (queries, forms, reports, macros and modules) and links to the tables in the back-end database. Typically the back-end database is located on a network server and copies of front-end database are installed on individual users' computer. The Database splitter Wizard creates a front-end / back-end application, splitting a database into a back-end containing the data (tables) and a front-end containing all other objects and links to the table in the back-end. The original database remains unaltered.

- Run Database utilities on an Open Database

  User can use the Compact Database and Repair Database Subcommands on the current database

- Performance Analyzer

  Users can optimize the performance of some or all the objects in database. The Performance analyzer will analyzes the database, suggests the best way to optimize its speed and performance and then automatically makes the necessary changes as requested.

- Database Properties

  From the Database Properties command, users can view, change and defines database properties, such as title, subject, author and creation data, for use in locating and identifying the database. [13]

2.8.2.2 Microsoft SQL Server 2000

Microsoft SQL Server 2000 is a family of products that meet the data storage and analysis requirements of the largest data processing systems and commercial Web sites. The same products can provide easy-to-use data storage and analysis services to an individual or small business. SQL Server 2000 offers:

- A modern relational database engine that can scale from running on an individual desktop to running the largest Web sites. SQL Server 2000 is integrated with Microsoft Windows 2000 fail over clusters to provide exceptionally reliable data servers, and integrated with Windows 2000 authentication and encryption to implement secure systems.

- Integration into the Microsoft data access environment. SQL Server 2000 provides native support for ADO, OLE DB, and ODBC. SQL Server also introduces integrated support for Web-based application development, supporting HTTP access using URLs, and returning data as XML documents.

- An integrated set of Analysis Services tools for performing complex data analysis and data mining of data warehouses.

- Replication services, which allow sites to place copies of data on multiple computers to improve overall system performance while keeping the data synchronized.

- English Query, which applications can use to answer ad-hoc user questions. When given a string containing a question about the data in a database or data warehouse, English Query returns an SQL or MDX statement that can be run to get the answer.

- Full-Text Search, which extends the pattern matching capabilities of SQL Server 2000 beyond the simple pattern matching available in the SQL language, including searches in files stored outside of SQL Server databases.

- Meta Data Services, which provide facilities for storing, viewing, and retrieving descriptions of the objects in your applications and system.

- Data Transformation Services (DTS) that make it easier to build OLAP data warehouses. DTS provides powerful services that allow records of individual transactions to be transformed into summary information stored in a data warehouse. [14]

2.8.3 MICROSOFT DATA ACCESS TECHNOLOGY

2.8.3.1 Open Database Connectivity (ODBC)

ODBC is a standard for accessing data. It was designed to allow the programmers to use a common set of routines to access the data stored in database, regardless of the type of database in which the data is stored. They can manipulate the data without worrying the exactly where the data was stored, or which type of database was storing it. It provided interface transparency – the programmer can access an Oracle database in the same way they accessed a SQL Server database. (see Figure 2-5)

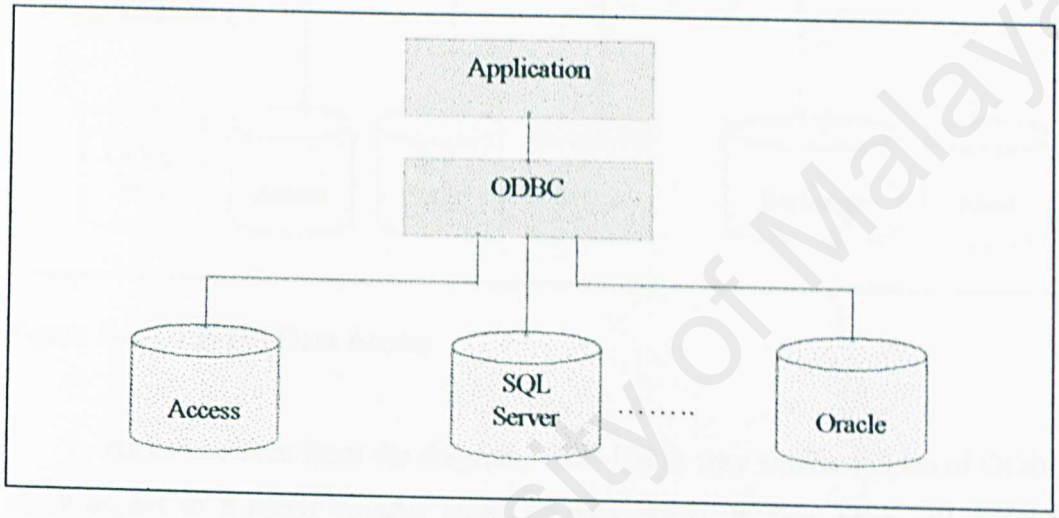


Figure 2-5: ODBC Data Access

All of the methods for accessing data, the Open Database Connectivity (ODBC) is the most efficient in terms of execution speed. In terms of programming, it requires the most time and the most caution. [15]

2.8.3.2 OLE-DB

OLE-DB is faster and easier to use than ODBC. The following Figure 2-6 is a picture of data access using OLE-DB.

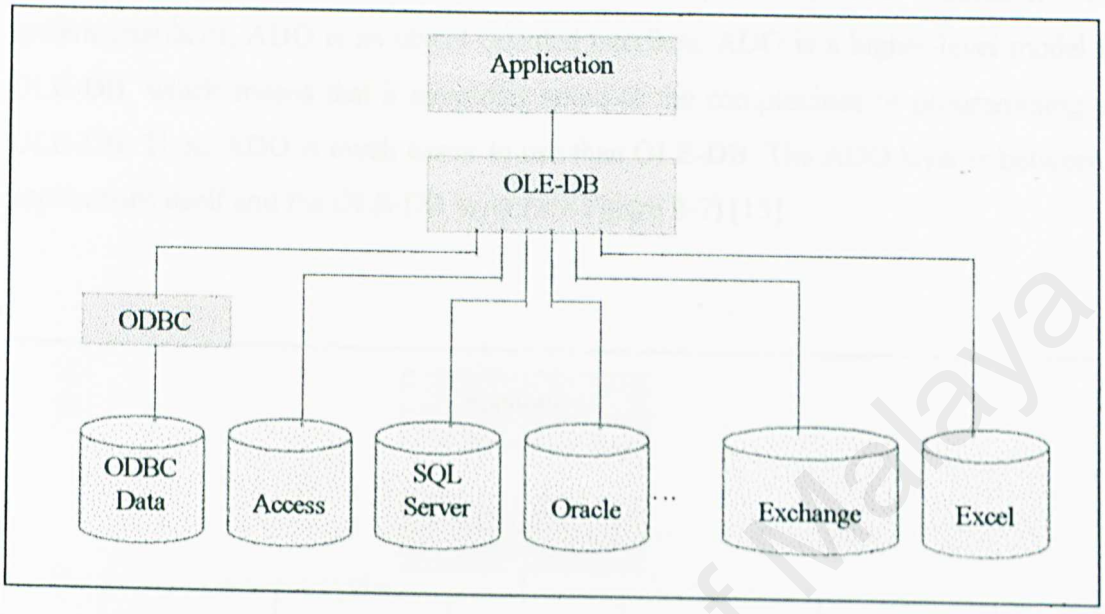


Figure 2-6: OLE-DB Data Access

As can be seen from the diagram, OLE-DB is very similar to idea of ODBC, but if allow access to a much broader range of data stores. In fact, OLE-DB even supports database connections through ODBC, so that effectively the generic OLE-DB layer allows the connection to legacy database through existing ODBC connections.

OLE-DB introduces data providers and data consumers. A data provider is something that provides data for data consumers, such as an application written in language like Visual Basic or Visual C++. [15]

2.8.3.3 ActiveX Data Objects (ADO)

ADO is an Application Program Interface (API) from Microsoft that lets programmers writing windows application get access to relational and non-relational database from both Microsoft and other database providers. Like Microsoft's other system interfaces, ADO is an object-oriented interface. ADO is a higher-level model than OLE-DB, which means that it simplifies some of the complexities of programming with OLE-DB. Thus, ADO is much easier to use than OLE-DB. The ADO layer is between the applications itself and the OLE-DB layer.(see Figure 2-7) [15]

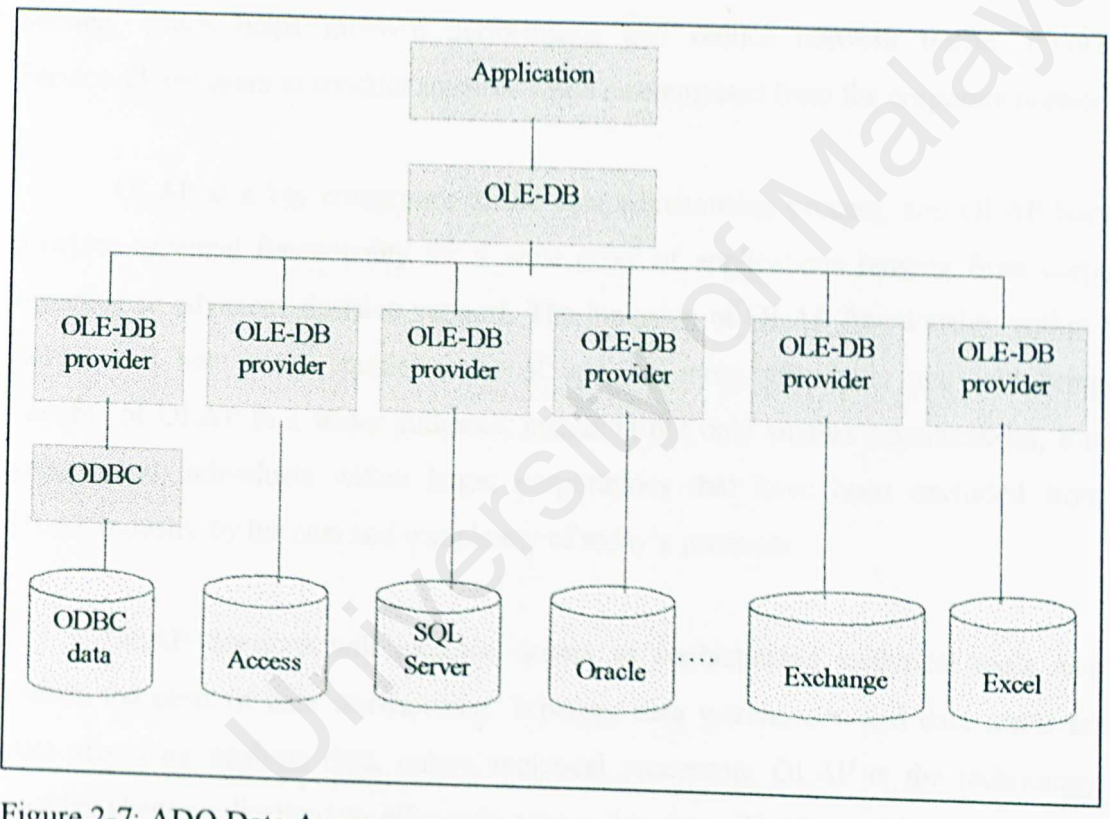


Figure 2-7: ADO Data Access

2.8.4 ANALYTICAL TECHNOLOGY

2.8.4.1 Online Analytical Processing (OLAP)

OLAP is an increasingly popular technology that can dramatically improve business analysis and makes multidimensional analysis accessible to a broader audience and potentially at a significantly lower cost of ownership.

OLAP Services includes a middle-tier server that allows users to perform sophisticated analyses on large volumes of data with exceptional performance. A second feature of OLAP Services is a client cache and calculation engine called PivotTable Service, which helps improve performance and reduce network traffic. PivotTable Service allows users to conduct analyses while disconnected from the corporate network.

OLAP is a key component in the data warehousing process, and OLAP Services provides essential functionality for a wide array of applications ranging from corporate reporting to advances decision support. The inclusion of OLAP functionality within SQL Server will help make multidimensional analysis more affordable and will bring the benefits of OLAP to a wider audience, including not only smaller organizations, but also groups and individuals within larger corporations that have been excluded from the OLAP industry by the cost and complexity of today's products.

OLAP Services will increase access to sophisticated analytical tools and can reduce the costs of data warehousing. Whereas data warehouses and data marts are the data stores for analysis data, online analytical processing OLAP is the technology that enables client applications to efficiently access this data. OLAP provides many benefits to analytical users.

For example:

- An intuitive multidimensional data model makes it easy to select, navigate, and explore the data.
- An analytical query language provides power to explore complex business data relationships.
- Pre-calculation of frequently queried data enables very fast response time to ad hoc queries [16]

Introduction to cubes

Cubes (see Figure 4-1) are the main objects in online analytic processing (OLAP). A cube is a set of data that is usually constructed from a subset of a data warehouse and is organized and summarized into a multidimensional structure defined by a set of dimensions and measures.

A cube provides an easy-to-use mechanism for querying data with quick and uniform response times. Pre-calculated summary data called aggregations provides the mechanism for rapid and uniform response times to queries. Aggregations are created for a cube before end users access it.

Every cube has a star (see Figure 4-2) or snowflake schemas, which is the set of joined tables in the data warehouse from which the cube draws its source data. The central table in the schema is the fact table, the source of the cube's measures. The other tables are dimension tables, the sources of the cube's dimensions.

Each cube dimension can contain a hierarchy of levels to specify the categorical breakdown available to end-users. Dimension levels are a powerful data modeling tool because they allow end users to ask questions at a high level and then expand a dimension hierarchy to reveal more detail. [16]

Introduction to PivotTable Service

PivotTable Service is used to build client applications that interact with multidimensional data. PivotTable Service also provides methods for online and offline data mining analysis of multidimensional data and relational data. PivotTable Service is included as part of Analysis Services.

PivotTable Service is the primary method for interacting with Analysis Services in order to accomplish such tasks as connecting to a cube or data mining model, querying a cube or data-mining model, and retrieving schema information. [16]

2.8.5 WEB DEVELOPMENT TECHNOLOGY

2.8.5.1 Common Gateway Interface (CGI)

The CGI is a mechanism for creating scripts on the server, which can then be used to create dynamic web application. CGI is the standard way for a Web Server to pass a web user's request to an application program and to receive data back to forward to the user. When the user requests a web page (for example, by clicking on a highlighted word or entering a web site address), the server sends back the requested page. However, when a user fills out a form on a web page and sends it in, it usually needs to be processed by an application program. The Web Server typically passes the form information to a small application program that processes the data and may send back a confirmation message. This method or convention for passing data back and forth between the server and the application is called the Common Gateway Interface (CGI). It is part of the web's HTTP protocol.

However, CGI has some severe shortcomings. The major one is that it adds an extra level to our browser-server model of interaction: namely, it is necessary to run a CGI program to create the dynamic page, before the page is processed on the server.

Also the format in which CGI receives and transmit data means that this data is most easily manipulated by many programming languages, hence programming languages that has good facilities for manipulating text and communicating with other software has to be used. The most able programming languages that can work on any operating system for doing this are C, C++ and Perl. Visual Basic does not offer sufficiently adequate text-handling facilities and is therefore rarely user with CGI. [15]

2.8.5.2 Microsoft Active Server Page (ASP)

ASP is Microsoft's most recent Web Server application development technology, designed to make it easier for web application developers to create sophisticated web application. Because ASP supports VBScript, JavaScript, Perl and other scripting languages, web developers do not need to learn an entirely new language. ASP has been designed to make use of the existing knowledge of web application developers.

ASP is an ISAPI extension, which builds on top of the ISAPI infrastructure to provide a server-side application framework, making it even easier to build dynamic web application. An ASP document contains both HTML syntax and server-side script logic. When the Web Server receive an HTTP request for the ASP document, a 'virtual' output HTML file is generated (in memory) for the response using a combination of both the HTML static information plus any HTML that is generated by scripting.

Benefit of using ASP

There are many benefits in using ASP that make it one the most powerful tools available for developing sophisticated web application.

Here are some of the benefits:

- ASP complements client-side scripting
- ASP development is easy to learn
- With the ASP development environment, you can easily leverage existing investment and skills.
- ASP development is compile- free
- The ASP environment is extensible
- ASP protects proprietary business algorithms and information [15]

ASP objects

Although we can use script in an ASP, this itself exerts constrains on what we can actually achieve, as the scripting languages we are using have very limited functionality. However they do have useful ability, namely they can act upon objects that are part of ASP core engine, by calling their methods and setting their properties. ASP provides a distinctive set of objects that we can manipulate using scripting languages. There are six build in objects provided by the ASP core engine, which we cam use in our server programs. Unlike other object models, such as the browser Object Model, the ASP build in objects do not form a hierarchy. They on relate to each other logically.

The Object Model provides us with a Server Object, which has method and properties that offer general utility functions that we can use through scripts. The server object represents the environment in which our pages run, and the remaining 5 objects are used to make up an Active Server application. These objects are Server, Application, ObjectContext, request, Response and Session. (see Table 2-2)

NAME OF OBJECT	PURPOSE
Server Object	Manage server objects, HTML streams and URLs.
Application Object	Share information between users of an ASP application.
ObjectContext Object	Commit or abort a transaction that is managed by Microsoft Transaction Server.
Request Object	Used to retrieve information passed to the web server by the web browser.
Response Object	Sends text output to the web browser.
Session Object	Store temporary information used during the course of a user session.

Table 2-2: The six build in ASP object.

2.8.5.3 Comparison between ASP and CGI

ASP provides all of the functionality of CGI application in an easier to use and more robust environment. ASP is an easier way for your server to access information in a form not readable by the client (such as an SQL database) and then act as a gateway between the two to provide information that the client can view and use.

With CGI, the server creates as many processes as the number of client requests received. The more concurrent request there are, the more concurrent processes created by the server. However creating a process for every request is time-consuming and requires large amounts of server RAM. In addition, this can restrict the resources available for sharing from the server application itself, slowing down performance and increasing wait times on the web.

ASP runs in the same process as Web Server, handling client requests faster and more efficiently. It is much easier to develop dynamic content and web application with ASP. [17]

2.8.6 WEB APPPLICATION TOOLS

2.8.6.1 Microsoft Visual InterDev 6

Visual InterDev is a comprehensive, Web based application development tool. It provides an integrated environment that brings together various technologies to work toward a common goal of building robust and dynamic application for the web. It achieves this integrated development environment through the use of its Microsoft Development Environment, also employed by Visual J++. User can open and work on Visual J++ project while simultaneously creating Visual InterDev project. This feature greatly enhances productivity, especially when building components using the Microsoft Component Object Model (COM) and incorporating these objects into the Visual InterDev application. It can also open multiple Visual InterDev projects within the same workspace.

Visual InterDev enables the developers to build application that are dynamic and interactive. Visual InterDev enables the developer to build dynamic web pages through the use of Client and Server side script. By default, Visual InterDev supports the use of VBScript and Jscript.

Database integration vital to an application Visual InterDev provides a rich and robust set if visual database tools to immediately enhance your productivity. Visual InterDev supports the major ODBC – compliant databases, both on the desktop and the server.

Visual InterDev supports the major object based technologies that exist for developing Web-based application, including ActiveX Controls and Java applets. Transformation of script code into 'Object-based' functions such as scrip lets and using them within the Visual InterDev project can be done. Visual InterDev supports the use of third party ActiveX Controls and enables the use to integrate custom ActiveX control.

Furthermore, Visual InterDev enables single developer to work and test parts of a project against local web server without interfering with the team then synchronize and deploy changes to the shared Master Web Server.

It adds database connections to a web site without any programming, including visually setting connection properties such as cursor drivers, query time out, etc. it also drags and drop from the Data Environment to quickly create sophisticated data driven HTML forms.

Query Builder allows developers to visually construct complex SQL statements against any ODBC database. Test any query in the live test pane before using within a web page.

The Visual Database Tools in Visual InterDev now provide enhanced support for Oracle databases, as will as Microsoft SQL Server. [18]

2.8.6.2 Microsoft Front Page

The Microsoft FrontPage 2000 Web site creation and management tool gives everything need to create and manage exactly the site whether creating a personal Web page or a corporate Internet or intranet site. And because it works like the rest of Microsoft Office Premium, FrontPage 2000 is easy to learn and use. Create, edit, manage, and update site—all from one easy-to-use application. The multiple views in FrontPage 2000 allow to see all the files in Web, run reports to find slow pages and older files, set up site's navigational structure, and keep track of Web tasks.

For example, can create and edit Web pages in Page View, use Folders View to see all the content on Web, and then set up how these pages link to one another in Navigation View. [19]

2.8.6.3 Comparison between Microsoft Front Page and Microsoft Visual Interdev

Visual InterDev is a web development tool designed for programmers, while Microsoft FrontPage is a web-authoring tool designed for non-programmers. Microsoft FrontPage is a member of the Microsoft offline family and looks and works like other Microsoft Visual Developments tool, including Microsoft Visual C++, Microsoft Visual J++, and Microsoft Visual FoxPro.

Because most web sites are created by terms of people, including both programmers and non-programmers, Visual InterDev and Microsoft FrontPage interoperate so that teams of people with different sets of skill can work together on the same web site. Web masters, technical developers, database administrators and MIS personnel would require Visual InterDev. On the other hand, content publishers – such as sales, marketing, graphic arts and finance- would need FrontPage. [20]

2.8.7 SCRIPTING LANGUAGE

2.8.7.1 VISUAL BASIC SCRIPTING (VBScript)

VBScript is powerful. It was designed to provide client-side scripting capabilities within Microsoft's Internet Explorer web browser. Various capabilities of VBScript can be used to develop richly interactive web pages that respond to user input in an intelligent manner.

VBScript is easier to use because it is based on the easy-to-learn BASIC (Beginner's all Purpose Symbolic Instruction Code) language. In fact, VBScript is a free available scripting language.

VBScript code is lightweight, fast and optimized to be transmitted via the Internet. Because VBScript code is lightweight, it can be quickly transmitted to users browsing a Website.

The only learning curve in VBScript is to figure out how does the language integrates with the environment in which it is implemented or used. Such environment can be Windows Operating System.

2.8.7.1 JAVA SCRIPTING (JavaScript)

JavaScript is a relatively new scripting language, developed by Netscape Communication and Sun Microsystems, which have rapidly gained popularity among Internet developers. JavaScript preceded the European Computer Manufacturers Association (EMCA) standard and it was the first web scripting language for dynamic interaction and content.

JavaScript is easy-to-use and designed for creating live online applications. It is analogous to VBScript. Like VBScript, JavaScript is based on a programming language.

Basic online applications and functions can be added to web pages with JavaScript, but the number and complexity of available API functions are less than what is available with Java. JavaScript code, which can be included in a web page along with the HTML code, is generally considered easier to write than the Java language itself.

CHAPTER SUMMARY

In this chapter 2, I will review all the documentation related with my system. I compare several tools and technologies that are useful for my system. By reviewing these things, I will get more idea and information for my system. I compare all these tools and technologies to make comparison. From here, I will know whether feature for these tools and technologies are suitable to my system or not.

LIST OF REFERENCE

[1] www.zsassociates.com/expertise/issue/busplan.html

[2] Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman. "System Analysis and Design Methods." 5th Edition. New York, McGraw-Hill: 2000.

[3] Levenbach, Hans and Cleary, James P. "The Modern Forecaster." New York, Vas Nostramd Reintiold Company: 1984.

[4] Carter, Lawrence R.and Lee, Ronald. "Joint Forecast of U.S Marital Fertility, Nuptiality, Births and Marriage Using Time Series Model." Journal of the American Statistical Association 81, no. 396 (1986): 902-911.

[5] Land, Kenneth C. "Methods for National Population Forecasts: A Review." Journal of the American Statistical Association 81, no. 396 (1986): 888-901.

[6] Berk, J.M. and Bikker, J.A. "The Use of Leading Indicators for Forecasting and Analysis: International Interdependence of Business Cycle in Manufacturing Industry." Journal of Forecasting, no. 14 (1995): 1-23.

[7] Liu, Lom-Mu. "Sales Forecasting Using Multi-equation Transfer Models." Journal of Forecasting, no. 6 (1987): 223-238.

[8] S. Christian Albright, Wayne L. Winston and Christopher J. Zappe. "Managerial Statistics." U.S.A.: Duxbury Thomson Learning.

[9] http://msdn.microsoft.com/library/psdk/lisref/eadg4cbp.htm

[10] www.faqs.org/faqs/client_server_faq/section_4_html

[11] www.vbhow.to/books/vb6cs/ch01/ch01.asp

[12] http://msdn.microsoft.com/library/psdk/lisref/eadg6v2.htm

[13] Rusell A. Stultz. "Learn Microsoft

[14] http://msdn.microsoft.com/library/default.asp

[15] Davis Buser, John Kauffman, Juan T. Llibre, Brian Francis, David Sussman, Chris Ullman, JonDuckett. "Beginning Active Server Pages 3.0." Wrox Press Ltd: 2000

[16] http://www.microsoft.com/sql

[17] http://msdn.microsoft.com/aspfaq.htm

[18] www.microsoft.com/catalog/display.asp?site=663&subid=22&pg=1

[19] http://www.galathea.net/frontpage.html

[20] http://msdn.microsoft.com/vinterdev/prodinfo/qa.htm

CHAPTER 3

METHODOLOGY AND SYSTEM ANALYSIS

3.1 INTRODUCTION

After the literature survey, the next step is to perform a detailed analysis. System analysis is an attempt to understands how the system can help to solve the problem. A complete understanding of software requirements is essential to the success of a software development effort. The overall emphasis is to gather data and requirements for the new system and to consider alternative solutions to the problem within these constraints and the feasibility of these solutions.

3.2 METHODOLOGY

As mentioned either in the chapter 1 introduction, waterfall modes (see Figure 1-4) has been introduced in a developing the whole project. This methodology is very important in order to make sure that the project has been well planned from the beginning stage until the end of this project. To guarantee the success of this project, research has been on the related fields and system planning based on the approaches provided. The steps are explained in the following subsections. [1]

3.2.1 PLANNING

The first phase is to identity the problems and objectives of this project. This step is very important because addressing the wrong objectives would waste subsequent time and lead to the failure at last. The objective of this project is to build a wed based application that allows users to store and retrieve the multidimensional data. It brings several advantages to the users. It provides the service for the users to analysis data in a faster way. Users would have easy access to the **BPS** as long as they get hook up on the Internet.

In identifying the problems, there are a few steps that can be followed to accomplish the process. They are explained in the section 3.2.

3.2.2 SYSTEM ANALYSIS

The next phase comes to analyzing the system needs. In this phase, the system is analyses by looking the functional requirement and non-functional requirements of the system. They are elaborated in section 3.3.

3.2.3 SYSTEM DESIGN

In this phase of the systems development life cycle, the information collected is needed to accomplish the logical design of the system. Accurate data flow procedures are planned so that the processes within the system are correct. [2]

3.2.4 TESTING

System testing is very important to assure the quality of the system. The goal of testing is to find the faults in the codes.

3.2.5 OPERATION

The process of charging a system after it has been delivered and is in use is called software maintenance. The charges may involve simple charges to correct existing errors, more extensive to correct design error or significant enhancements to correct specification. [1]

3.2.6 PROTOTYPING

This phase is to create a prototype according specification that are specify in the system design phase. With a prototype model, users will more understanding how the system works.

3.3 ANALYSIS PROCEDURES

The Process of analysis involves the following procedures

- Problem identification
- Evaluation and synthesis
- Modeling
- Specification

3.3.1 PROBLEM IDENTIFICATION

Before a new system can be built, we must identify the problem that needs to be solved in order to ensure the success of this project. For **BPS**, the problem is how to store, retrieve and manage a large amount of multidimensional data in a database system in an effective and efficient way, how to use the information to do analysis, calculation reporting. In additional, another problem would be to determine how to improve the performance of the system in transmitting large data.

3.3.2 EVALUATION AND SYNTHESIS

In this stage, analysis of the problems needs to be done by dividing the problems into small parts (invoking the "divide and conquer" technique) so that the problem will be easier to be understood and solved. The following problems are system requirements that must be considered:

- What kind of database is used to store the images?
- Using the relational DBMS or object oriented DBMS?
- How are the multidimensional data stored DBMS?
- How to index the data?
- Who manages the data?
- Who will be able to summit images to the system?
- How to do analysis?
- What method is used to do analysis?

3.3.3 MODELLING

We create models to gain a better understanding of the actual entity to be built. The model focuses on what the system must do, usually a graphical notation (such as data flow diagram) is used to depict information, processing, system behavior and other characteristics.

3.3.4 SPECIFICATION

The requirement specification is a complete listing that defines what the system should do. It will be used the system design and system testing. [1]

3.4 ANALYSIS

Requirement analyses cover the area of functional and non-functional requirements of **BPS**. The functional requirements can be categorized to the reporting section, the database management section and the analysis and forecasting section.

3.4.1 FUNCTIONAL REQUIREMENTS

3.4.1.1 Reporting Module

Output Module

This is the major service provided to the general users (all the staff in a organization) to look for report as well as information related with the organization. The system provides 3 type of display:

- Full record – display of the full information of a particular item selected
- Reporting analysis – display the results after analysis by using the selected data.
- Forecast report – display the results after forecasting

Data entry Module

The system provides several options to enable users to get the information and report they need. The options provided by the system include:

- Select report – user are allow to choose the report, they want to view
- Select display format – user can either choose report to be display in PivotTable or PivotChart format
- Select field – user can add in some of the field to the report to get more information

3.4.1.2 Database Management Module

This module only can access by user like managers or executives. This module allow managers or executives to:

- Edit data or information – they have the authority to edit the information such as user profile, operational data in the database.
- Update data or information – they have key in new operational data or user profile after collect all the new data from other divisional office.

3.4.1.3 Analysis and Forecasting Module

This module only allows managers or executives to do analysis of multidimensional data. They use all the operational data compare with the target and generate report to the general user. They also use the previous year sales data to do forecasting.

3.4.2 NON FUNCTIONAL REQUIREMENTS

Non-functional requirements are as important as functional requirements. It is defined as constraints under which the system must operate and the standards, which must be met by the delivered system.

3.4.2.1 User Interface

The system is required to have an interface that is specifically intended to allow the user to access the internal components of that system in a relatively easy fashion and without having to know specifically how everything is put together or how it works together. Common interface must be created in order to reduce the learning curves of the users. For example, saving and retrieving in are windows program is the same basic process as in all other window-based application.

3.4.2.2 Reliability

This system should be reliability, which means that it does not produce dangerous or costly failures when it is used in a reasonable manner. That is in a manner that a typical user experts is normal. This definition recognizes that a system may not always use in the ways that the designer experts.

3.4.2.3 Efficiency

Efficiency means a process or a procedure that can be called or accessed in an unlimited number of times to produce similar outcomes or outputs at a creditable pace or speed.

3.4.2.4 Simplicity

Simplicity refers to keeping forms and screen properly uncluttered in a manner that focuses the user attention.

3.4.2.5 Maintenance

A product is maintainable if the programs are easily modified and tested in the case of updating a process to meet a new requirement, correcting errors, or move to a different computer system. [1]

3.4.2.6 Understandability

Understandability in terms of the coding method used, allows other programmers to understand the logic of program flows, thus changes can be made easily upon the necessary program segments without modifying other users can use this system with out difficultly.

3.5 DATA FLOW DIAGRAM (DFD)

A data flow diagram (DFD) is a graphical technique that enables analysts to depict information flow in an information system. It allows how the data flow into the system, how they are transformed and how they leave the system. DFD provides a mechanism for functional modeling as well as information flow modeling. The components of the DFD consist of the following items shown in the Table 3.1.

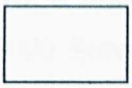
SYMBOL	COMPONENT
	Entity
	Process
	Data store
	Data flow

Table 3.1: Components of DFD

3.6 DEVELOPMENT AND PROGRAMMING TECHNOLOGY

This section will identify the suitable programming languages, development tools, server, database system and implementation platform that are used to develop the system. An analysis has been done in making the decision. The ideal solution for this project is easy to develop and deploy and also easy integration with the latest emerging technologies.

3.6.1 WINDOW 2000 SERVER

The Window 2000 Server was chosen to form the implementation platform. It was chosen primarily caused by some reason. Firstly, Window 2000 Server workstations allow users to control over access to files and other resources on their machines. The user can take advantage of various features of the NT logon process to control who can access the computers and set file permissions so that valuable data is protected from intruders even if the hard driver is physically removed. It is important because security is high risk for a company. Secondly, the Graphical User Interface for Window 2000 Server was user-friendly and ease to use. This means the user can shorten the learning time and save the cost to follow the course.

3.6.2 INTERNET INFORMATION SERVER (IIS) 5.0

This is the web server chosen for this project. IIS is chosen because users and groups from IIS were integrated with NT users and groups. It is important because Window 2000 Server was chosen as the network operating system for the new system.

Further more, IIS is easy to develop and a powerful application that can be created through ISAPI and Active Server Page (ASP). ASP is supported by various languages such as VBScript, PERL, and JavaScript. IIS includes great sample application and full access to library, and sample code that can be easily cut and paste. [3]

3.6.3 MIRCOSOFT ACTIVE SERVER PAGES (ASP) 3.0

ASP was chosen as the programming language technology because it is possible to use all the power of a real computer language right in the Web page. The user doesn't need to write complicated C code to interface with the Web Server and then compile it into an executable or DLL file. All they need to do is creates HTML page and then put programming code along side the HTML for then to work together.

ASP solves all the problems associated with CGI and server APIs. ASP is simple to learn and easy to use. ASP makes it much quicker and easier to create highly interactive Web sites. It also makes the pages easier to maintain and update in the future. [4]

3.6.4 HYPERTEXT MARKUP LANGUAGE

This is the basic language required to display text, images and so forth on the web page. It is a simple system for defining the appearance and functionality of hypertext document published via the World Wide Web. [5]

3.6.5 MICROSOFT SQL SERVER 2000

Microsoft SQL Server is chosen as the database server it is the best database solution to be used with Window 2000 Server. SQL Server is able to handle a large amount of data.

SQL Server allows the developer to define up to 32767 databases. A total of 2 million tables can be defined with each of 32676 databases. A table is allowed to define up to 250 columns. This means that SQL Server can manage to store a vast amount of data, which is important especially for sales data.

3.6.6 ONLINE ANALYTICAL PROCESSING (OLAP)

After consider, OLAP application was chosen because

- Fast

  Can delivery fast analysis of multidimensional data. Most queries can be delivered to the user in five seconds or less

- Analysis

  Performs basic numerical and statistical analysis of the data, predefined by an application developer or defined ad hoc by the user.

- Information

  Accesses all the data and information necessary and relevant for the application, wherever it may reside and nor limited by volume. [6]

3.6.7 VBSCRIPT

After consider several languages that are available in the market, VBScript was chosen as the main scripting language. Id is because VBScript is the default scripting language of Active Server Pages. The approach is chosen because it is simple to implement and no additional software is required besides Window 2000 Server and Internet Information Server 5.0. [7]

3.6.8 MICROSOFT VISUAL INTERDEV 6

After surveying the development tools, it is decided that Microsoft Visual InterDev will be used to develop the Web application as it provides a friendly development environment to develop Web application. Visual InterDev is used to create and edit content, and the content is automatically uploaded to the Web Server. When the system includes a database connection, Visual InterDev allows viewing and interacting with the data on the database server. The Microsoft Web Browser is integrated with Visual InterDev to view Web pages during development. Beside that, Microsoft Visual Basic 5 will used to develop Active Server Pages component. [8]

3.7 SYSTEM REQUIREMENTS

3.7.1 SERVER SYSTEM REQUIREMENTS

- A server with at lest Pentium 133 MHz processor or above
- At least 32MB of RAM
- At least 400MB of free space in hard disk
- Network connection with recommended bandwidth at 10Mbps or more
- Other standard computer peripherals

3.7.2 CLIENT SYSTEM REQUIREMENTS

- A computer with at least Pentium 100MHz processor or above
- At least 16MB of RAM
- Network connection through existing network configuration or modem (recommended at least 14.4 Kbps)
- Other standard computer peripherals

CHAPTER SUMMARY

After review very tools, techniques and technologies, I had made comparison between these tools, techniques and technologies. I had made decision to choose all the tools, techniques, and technologies that are suitable for my system to perform a good system. Tools, techniques and technologies I will go to use are:

Operating system	Window 2000 Server
Web server	IIS
Browser	Internet Explorer
Database	Microsoft SQL Server 2000
Programming	Microsoft VB, ASP
Reporting tools	PivotTable and PivotChart
Analytical technology	OLAP
Forecasting method	Moving-average method

Table 3-2: Development tools

LIST OF REFERENCE

[1] Kenneth E. Kendall and Julie E. Kendall. "System Analysis and Design", 3rd Edition, Prentice Hall International, Inc: 1996.

[2] Shari Lawrence Pfleeger, "Software Engineering, Theory and Practice", Prentice Hall Inc, 1998

[3] http://msdn. Microsoft.com/library/

[4] Homer, Enfield, Gross, Jakab, Hartwell, Gill, Francis, Harrison. "Professional Active Server Pages", Wrox Press Ltd.: 1997.

[5] Todd Stanffer, "HTML By Example" Qus, Macmilian Computer Publishing: 1996

[6] http://www.microsoft.com/sql/

[7] Jerry Honeycutt, "VBScript By Example" Que Publishing.

[8] Mike Marrison. "Special Edition Using Microsoft InterDev" Qus, Macmillian Computer Publishing: 1998.

CHAPTER 4

SYSTEM DESIGN

4.1 BPS'S DATABASE DESIGN

BPS database is a relational database model. The relational database model is implemented through a very sophisticated Relational Database Management System (RSBMS). The most important advantage of the RDBMS is the ability to implement the Relational Model, which lets the user / designer operate in a human logical environment RDBMS takes care of all of the complex physical details. Thus, the relational database is perceived by user to be a collection of tables in which data are stored.

Each of the table is a matrix consisting of series of row or column intersection. Tables, also known as relations, are related to each other by sharing a common entity characteristic. For example, Rec_Ty is one of the fields in Job_Rpt_Day_Tran_Current table. This common attribute between these tables makes it easy to connect data from these tables, which are completely independent from one another. [1]

Transfer data to a single MS SQL Server 2000 database, and build a customized MS Excel 2000 front end. This provides data-entry screens for both forecast sales and financial data as well as an element of fixed reporting, such as profit-and-loss accounts.

4.1.1 OLAP CUBES

In addition, MS OLAP Services creates a cube (see Figure 4-1), which is then used by OLAP Work, via Excel, for all flexible reporting requirements. The single database covers several issues; different units of measurement, weekly versus monthly forecasts and sales versus product forecasts.

The system replaces the previous system without requiring rewrites to three existing data feeds (two from external sales person providing weekly and monthly actual sales information, and an internal data extract from a separate system for products distributed internally).

This information, together with the forecast data entered via the Excel front-end, is processed into a single cube used by OLAP Work. In addition, financial information such as prices and costs, is entered via the Excel front end and stored in the same SQL Server 2000 database, but not used by the cube. This information is used to generate fixed financial reports.

The cube is processed immediately following any new data loads or hierarchy changes. This prevents the need for specific weekly cube builds that the old system required. The front end also gives additional capabilities such as hierarchy maintenance.

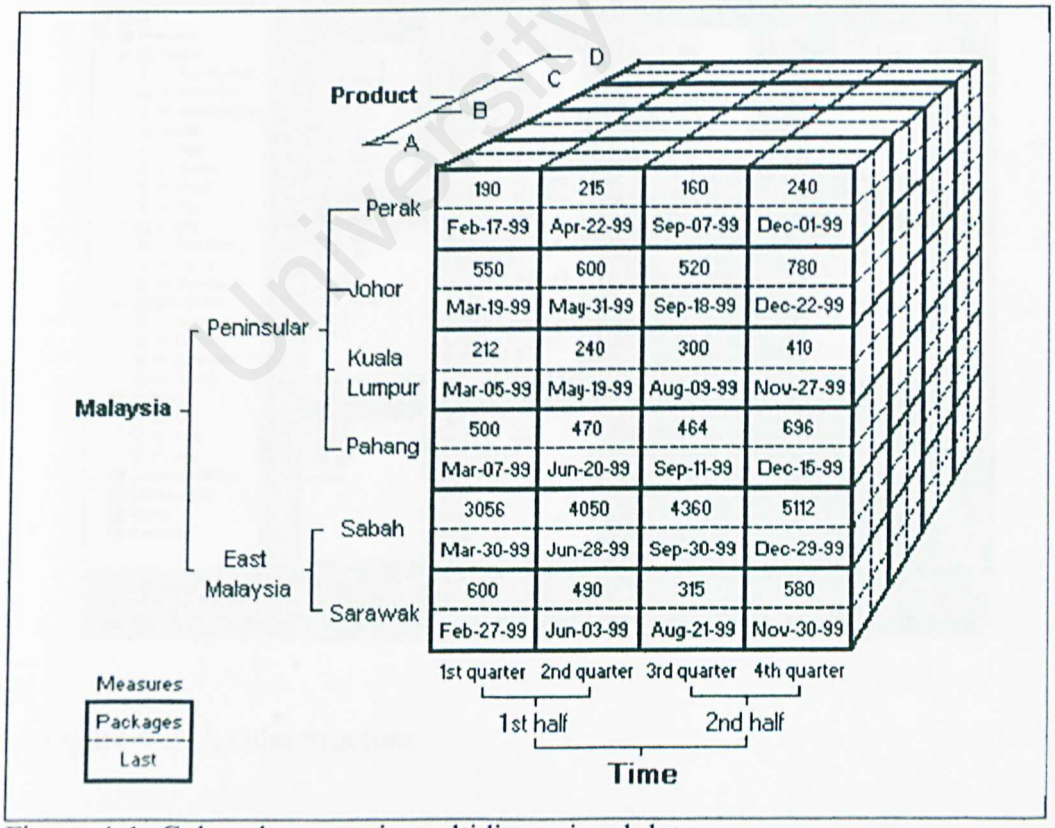


Figure 4-1: Cubes that contain multidimensional data

4.1.2 BUILDING THE OLAP CUBES

First, is mapping the initial database schema to the multidimensional model. In most OLAP implementation, it is assumed that the data has been prepared for analysis through data warehousing, whereby information has been extracted from operational systems, validated, and summarized prior to incorporation into an OLAP application. This is a vital step in the process, which ensures that the data being viewed by the OLAP user is correct, consistent and matches organizational definition for data.

To define a cube, can either choose a star or snowflake schemas. After that select a fact table and identify measures within the fact table. Then select or create dimensions, each composed of one or more columns from another table. A star schema is chosen for the **BPS**. The following figure (see Figure 4-2) shows a star schema that had build for **BPS**.

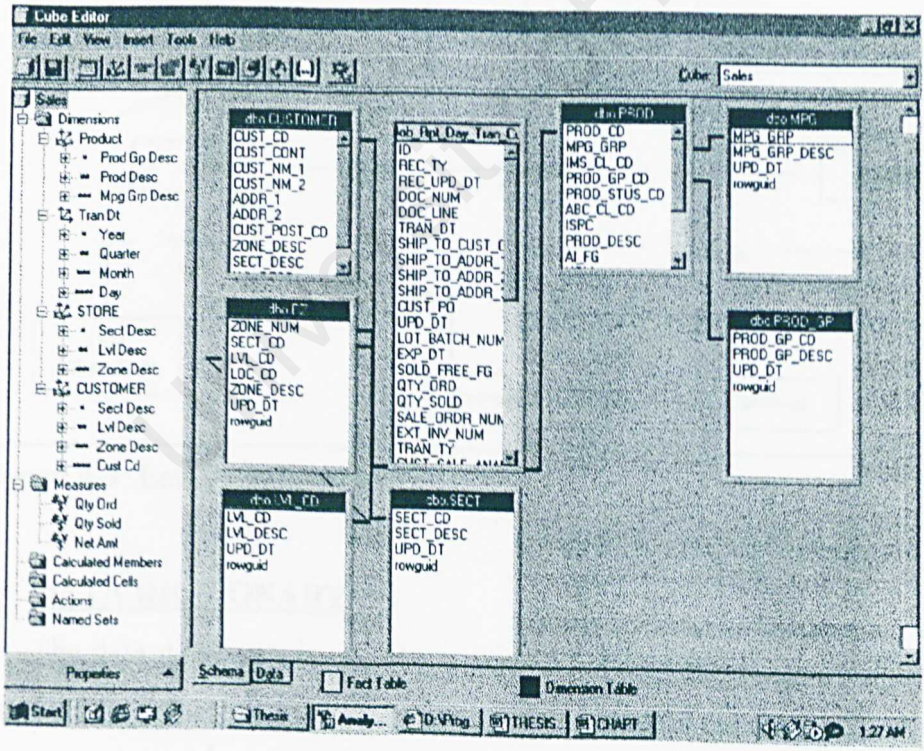


Figure 4-2: A cube structure

After define a new cube, can design its aggregations. Designing the aggregations it to specifies the summarization strategy. After you design the aggregations of a new cube, a simple analysis can be done based on the aggregations. A cube can be change or its source data can be changes to update the data.

4.1.3 ENTITY RELATIONSHIP (E-R) MODEL

An E-R model provides a simplified picture of the relationship among entities. It is a tool that is commonly used to:

- Translate different views of data among managers, and users to fit into a common framework (see Figure 4-3).
- Define data processing and constraint requirements to help us meet the different news.
- Help implement the database. [2]

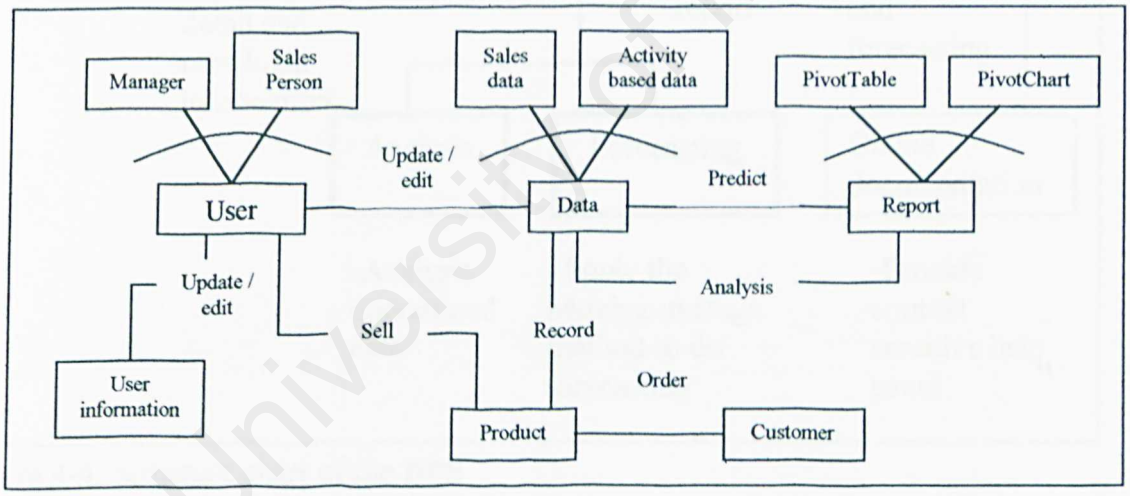


Figure 4-3: Entity-relationship for **BPS**

4.1.4 DATA DICTIONARY

The data dictionary is used to provide a detailed accounting of all tables found within the user / designer created database. Thus the data dictionary contains (at least) all the attribute manes and characteristic for each table in the system. In short, the data dictionary contains metadata (data about data). Data dictionary for **BPS** database is shown at the appendix.

4.2 SYSTEM DECOMPOSITION DESCRIPTION

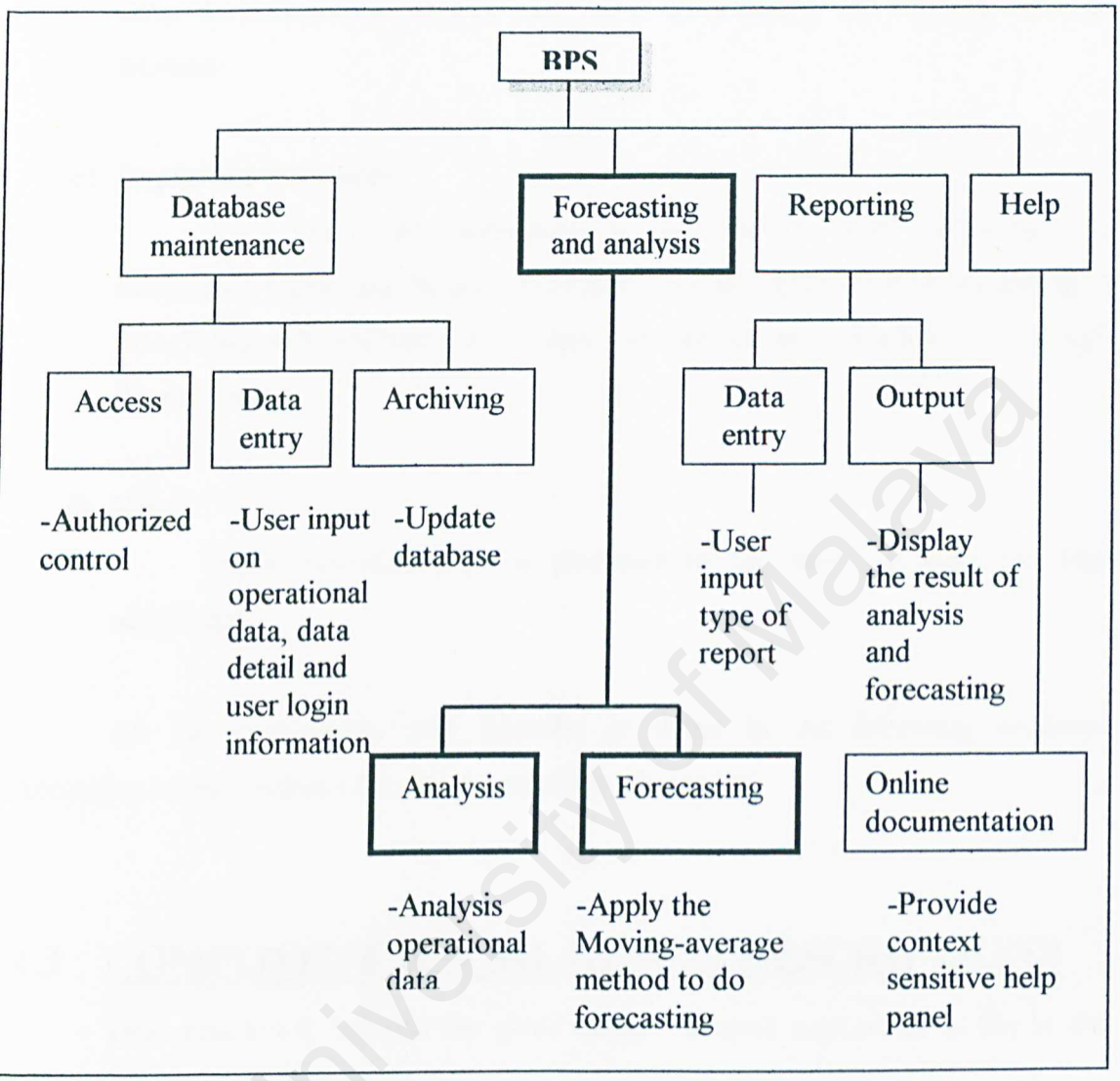


Figure 4-4: Structure chart of the **BPS**

The structure of the system in shown is shown Figure 4-4. This system architecture makes function allocation among of four main components:

a) Database maintenance component

This component consists of functions associated with collecting data entry from the authorized users, arching and the access control for the database.

b) Analysis and forecasting component

This component is associated with the functions analyzing the sales data, do forecasting on past year sales by applying the Moving Average Method.

c) Reporting component

This component generates several report types including the forecasting report and the analyzed report. All the reports will be presenting in PivotTable or PivotChart format. Any user can access to this function through the main menu.

d) Help

Online documentation is provided to the users through the help component.

All this components will describe in detail in the following sections. According to the functions that are provided by this system.

4.3 COMPONENT DETAIL DESIGN DESCRIPTIONS

This section will describe the detail design for each component in the in the following format (see Table 4-1):

Format	Description
Function	Specific function allocated to the component with additional detail as required, including requirements derived by stepwise refinements of functions allocated to the component. [3]
Sequential Control Features	Control details are covered in the section, depicting the order of processing, including the data flow diagram for each module. [3]

Application Data	Data required to deploy the system in the application are including all the files, records and fields needed for the application. [3]
User Interface	The description about how the users can interact with the module and function.

Table 4-1: Format to describe each component

4.3.1 COMPONENT DATABASE MAINTAINANCE

4.3.1.1 Function

As shown in Figure 4-4, this component shall provide the following functions (see Table 4-2) to the users:

Function	Description
Access	The function that verify and validate the authorized users to manipulate the database.
Data entry	This function will require users to key in the type, detail of operational data, input new data and the log in information.
Archiving	This function will update the operational data and the user log in information in the database.

Table 4-2: Functions that provided in the component Database Maintenance

The authorized users such as manager or executive can make any changes in the collections of the operational data: including update records, add new records and deleting. For the user such as sales person can access to this component but cannot make any changes. Besides those listed function, this component also should generate an error message on every mistake of the user.

4.3.1.2 Sequential Control Features

The order of the processing in this component is represent by the data flow diagram (DFD) that is shown in Figure 4-5 and Figure 4-6.

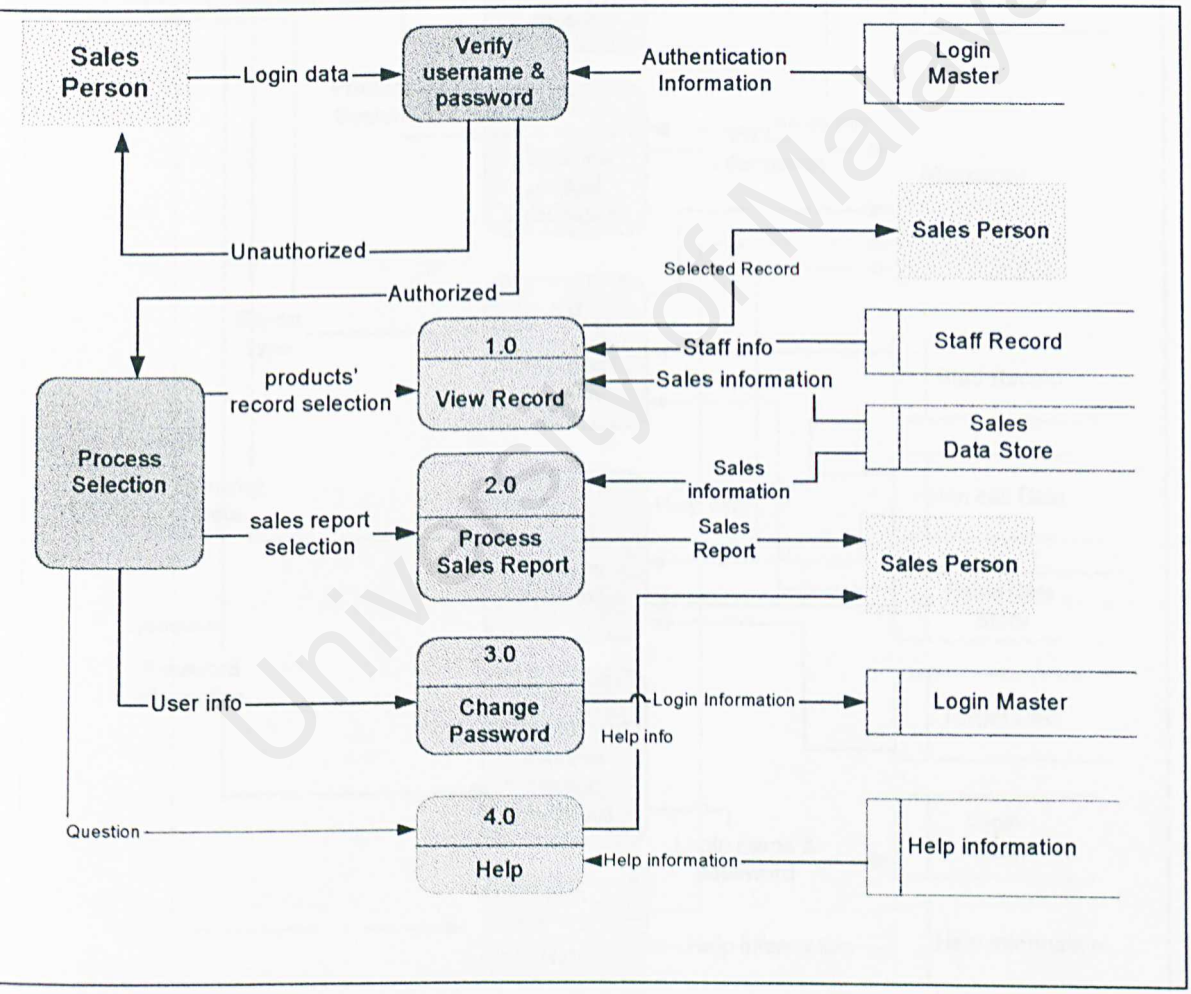


Figure 4-5: DFD for the Database Maintenance Component (Sales Person)

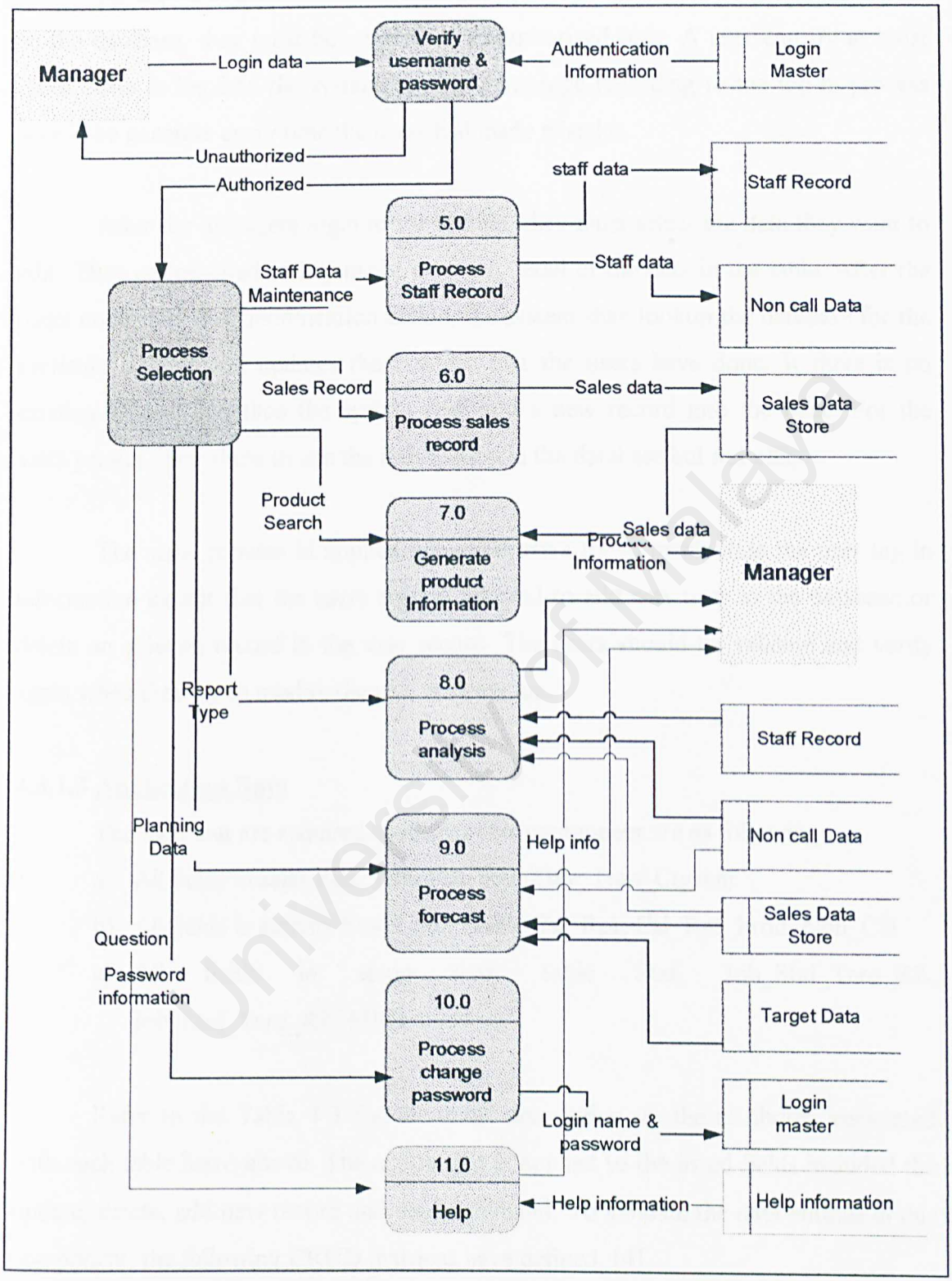


Figure 4-6: DFD for the Database Maintenance Component (Manager)

As shown in the DFD in the Figure 4-4, every time before the users can access to the database, they must be verified as an authorized user. A user can try at most three times to log into the system. An error message regarding to the log in process should be generate every time the users had made mistake.

After the managers login to the system, they must select the data they want to edit. They are required to key in the type and detail of the data in the table. After the users confirmed their modification action, the system shall lookup the database for the particular record and updates the changes that the users have done. It there is no existing record, and then the system shall add a new record into the table. For the sales person, they allow to use the information in the database but not edit.

The same process id applied when the users try to manipulate the user log in information except that the users are not allowed to add new user to the database or delete an existing record in the user record. The users should be validate and verify again when they try to modify the user information.

4.3.1.3 Application Data

The data that are required to display this component are as following:

a) All fields in sales data: table Job_Rpt_Day_Tran_Current
b) All fields in activity based data: table Cal_Rpt, Cal_Rpt_Prod, Non_Cal
c) All fields in setup data: table Staf, Job_Staf_Targ_RZ, Job_Staf_Targ_RZ_MPG

Refer to the Table 4-3 for the detail description on the attributes associated with each table listed above. The action that is applied to the listed fields included the update, delete, add new record and read command. To validate the data entities in this component, the following CRUD matrices have defined. [4]

	Sales data				Activity based data				Setup data			
	C	R	U	D	C	R	U	D	C	R	U	D
Verify user is valid										X		
Lookup the records		X				X						
Update records	X	X	X	X	X	X	X	X				
Lookup the user information										X		
Update user records									X	X	X	X
Lookup the idioms												

Table 4-3: Matrices of Data Entities against Events in Component Database Maintenance.

The meaning of C, R, U and D is followed

- **C**reated (same as SQL INSERT command)
- **R**ead (same as SQL SELECT command)
- **U**pdated (same as SQL UPDATE command)
- **D**elete (same as SQL DELETE command)

The 'X' shows that there is an input from or output to the table to the associated event. The row that has no entities means that it does not refer or affect table in the database.

4.3.2 COMPONENT ANALYSIS AND FORECASTING

4.3.2.1 Function

As shown in Figure 4-4, this component shall provide the following functions (see Table 4-4) to the users:

Function	Description
Analysis	This function shall analysis the data that base on the user inputs from the Data Entry function.
Apply Moving Average Method	This is the function that shall apply the Moving Average Method (see Section 2.5.2.1) to forecast the sales in the coming year.

Table 4-4: Function in component Analysis and Forecasting

This component can access by user such as manager or executives. The range of year that the users can forecast is depends on the records that are currently kept in the database.

4.3.2.2 Sequential Control Features

The order of processing for generate analysis function is shown in Figure 4-7 and the forecasting function is shown in Figure 4-8.

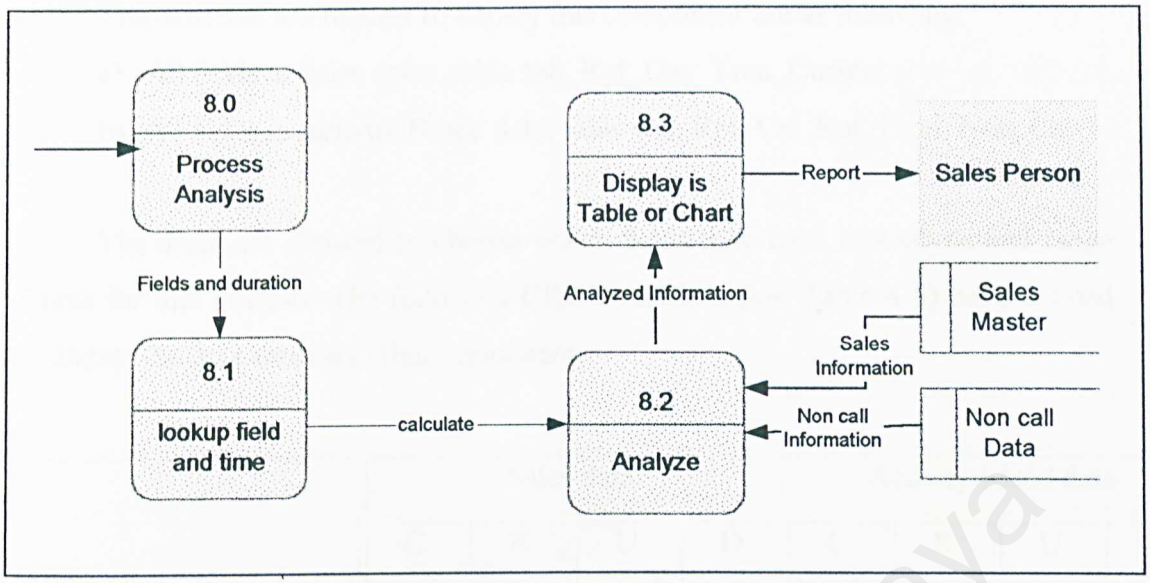


Figure 4-7: DFD for analysis function

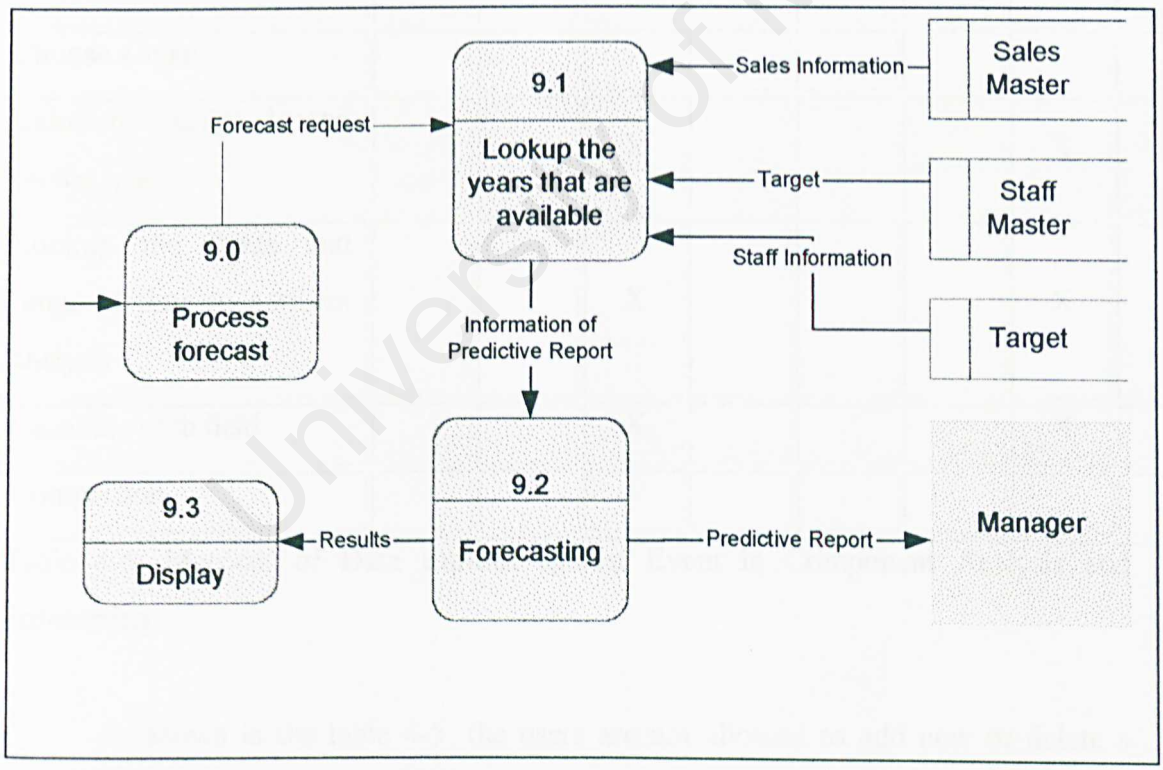


Figure 4-8: DFD for forecast function

4.3.2.3 Application data

The data that are request to deploy this component are as following:

a) All fields in Sales data: table Job_Rpt_Day_Tran_Current

b) All fields in Activity Based data: table Cal_Rpt, Cal_Rpt_Prod, Non_Cal

The users are allowed to choose which fields they want to analysis and range of time for that analysis. The following CRUD matrices (see Table 4-5) have defined to validate the data entities in this component.

	Sales data				Activity-based data			
	C	R	U	D	C	R	U	D
Lookup the year that are available to forecast			X				X	
Calculate total sales			X				X	
Choose a span								
Calculate average depend on the span			X				X	
Lookup the fields and range of time to perform analysis			X				X	
Calculate each field			X				X	
Comparison								

Table 4-5: Matrices of Data Entities against Event in Component Analysis and Forecasting

As shown in the table 4-5, the users are not allowed to add new or delete a record from the database. Also, users are not allowed to edit the cube, this function shall be provided only through the Database Maintenance component in the previous section.

4.3.3 REPORTING COMPONENT

4.3.3.1 Function

As shown in Figure 4-4, this component shall provide the following function (see Table 4-6) to the users:

Function	Description
Data entry	This function require users to input the time or the type of report they want to browse
Output	This function shall display the forecast and analysis report to the user in PivotTable or PivotChart.

Table 4-6: Function for component Reporting

4.3.3.2 Sequential Control Features

Figure 4-8 describes the data flow diagram for reporting component.

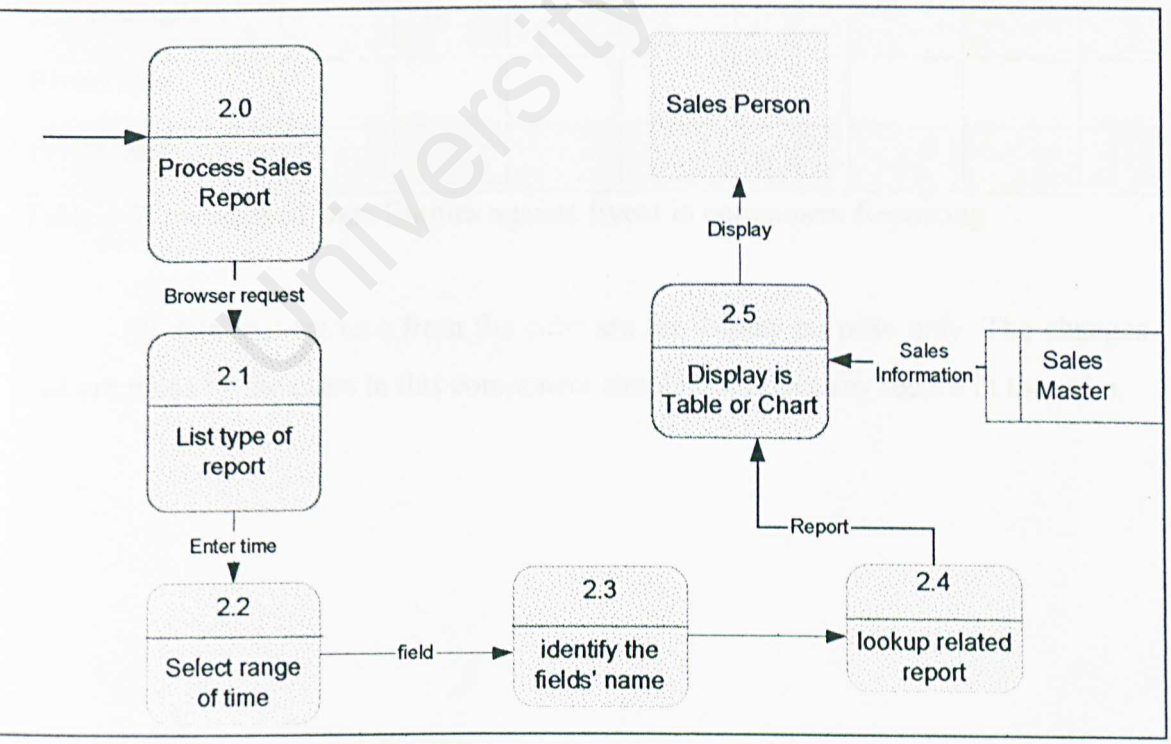


Figure 4-9: DFD for the component Reporting

4.3.3.3 Application data

The data that are required to deploy this component are as following (see Table 4-7):

a) All the fields in sales data: table Job_Rpt_Day_Tran_Current

b) All the fields in activity-based data: table Cal_Rpt, Cal_Rpt_Prod, Non_Cal

	Sales data				Activity-based data			
	C	R	U	D	C	R	U	D
List all type of report								
User select range of time								
Identify the fields' name								
Lookup the report details			X				X	
PivotTable								
PivotChart								

Table 4-7: matrices of Data Entities against Event in component Reporting

All data that retrieve from the cube are for display purpose only. The changes that are made by the users in this component should not affect any record in the cube.

4.3.4 HELP

4.3.4.1 Function

As shown in Figure 4-4, this component shall provide the online documentation to the users. This component shall provide the following topics in the online documentation.

a) Browse report - description on how to manipulate the browse report function.

b) Forecasting – description on how to manipulate the forecast function.

c) Analysis – description on how to manipulate the analysis function.

d) Database maintenance – description on how to manipulate the database maintenance function

The documentation on those topics should in both text and graphical format. All the users are allow entering to this component.

4.3.4.2 Sequential Control Features

The users may activate this function in this component from any other component in this system. They can read a topic at a time. They also can navigate back and forward in this component.

4.3.4.3 Application data

This component does not require any data from the database. Thus, there are no application data for this component.

4.4 WEB PAGES DESIGN

User interface plays a very important role in determining the quality of an information system. User interface is the component of the system that communicates with the users. Therefore, the input data collected from the users and output data generated for the users depend on a well-designed user interface. The interface design should meet the objectives for effectiveness, accuracy, ease of use, consistency, simplicity and attractiveness.

4.4.1 GENERAL PRINCIPLES

The general principles that should be applied in user interface design are mentioned in the following sections.

a) Consistency

A consistent format for command input, menu selection, data output and other interface function should be maintained. An inconsistent user interface will only lead to confusion.

b) Meaningful feedback

Feedback including proper and error message are displayed for a particular case to ensure that users do not panic when errors occur.

c) Robustness

System should be able to protect itself from errors and handle certain mistakes caused by the user input. This can be achieved by validating and verifying the users' input at the users' side.

d) Request for confirmation

For the critical functions such as delete, the system should ask for confirmation.

e) **Help**

Help is an important module in any system to provide the guideline to the users using the system.

4.4.2 INFORMATION DISPLAY

Information display is an important issue. The misplacing of information may confuse the users and lead to misconception on the results.

- Display information that is relevant to the current context
- Use consistent label and colour
- Provide meaningful error messages
- Use the presentation format that promotes ease of understanding the information.

4.4.3 DATA INPUT

Guidelines on data input:

- Minimize the number of input action required
- Use mouse to select predefined sets of input
- Allow user to control the interactive flow

4.4.4 USER INTERFACE DESIGN OF BPS

During the user interface design stage, the following issues have been taken into consideration:

- A consistent format for command input and menu selection
- Error messages for the invalid input value
- Use consistent label and color

The interface of the system can refer to appendix b.

CHAPTER SUMMARY

This chapter is discussing the system design. In this chapter I discuss on three main component of my system that are database maintenance component, analysis and forecasting component and reporting component. For each this component, I discuss about their function, sequential control feature and application data. From the data flow diagram show how this function works.

LIST OF REFERENCE

[1] Peter Rob and Carlos Coronel 1997. Database System: Designs, Implementation, and Management, 3rd Edition, Course Technology, Cambridge.

[2] http://www.soroban.com/whats_soroban_eng.html

[3] Ali, Benforooz, and Hudson, Frederick J. "Software Engineering Fundamentals." New York, Oxford, Oxford University Press: 1996.

[4] Ow, Siew Hock and Mashkuri, Hj, Yaaclb. 1998. Manual on the Writing of Specification and Review Checklists for Client / Server System. 1st edition. Damansara Utama, Petaling Jaya, Sejana Publishing.

CHAPTER 5

SYSTEM DEVELOPMENT & TESTING

5.1 SYSTEM DEVELOPMENT

5.1.1 STARTING OFF

After preparing all the resources, servers and development tools installations are the very first step before development work begins. The significant for using Microsoft's products is because it shows all the sequence of products installations. This is to ensure smooth execution without system errors.

The first process is to install the Window 2000 Server and made all necessary configuration of the Window 2000 Server. Window 2000 Server was chosen because of it flexible management services, and provides the best foundation for running business applications. So, have to make sure the entire configuration is correct.

After that is to install all needed network component. Besides that, also install development tools such as Visual InterDev 6, Microsoft Active Server Pages, Microsoft Front Page and Microsoft SQL Server. Other graphical software like Macromedia Flash and Dream weaver also install.

5.1.2 SETTING UP THE ENVIROMENT

BPS is a system that using a large amount of data to do analysis and forecasting. So, it is needed to create a database to maintain the data. All the data are keep track inside the database. So has to establish a connection to the database. There is variety of method to establish a connection including ODBC data source names (DSN), OLE DB data links, and so-called DSN-less connection.

5.1.2.1 Create Data Source Name (DSN)

The first step to establish a connection to data is to create a data source name (DSN) using ODBC; an ODBC data source name (DSN) is used. DSN is a method of standardizing database connectivity. There are three types of DSN: System DSN, User DSN and File DSN.

A system DSN is available to all users on the current machine, including NT services, making it ideal for use with ASP/ADO. It's also the fastest type of DSN because the connection information is stored in Windows Registry. A System DSN tends to be more convenient but less secure than a User DSN. [1]

To create a System/User DSN

1. On the **Start** menu, click **Settings**, and then click **Control Panel**.
2. Double-click ODBC to open the driver manager's user interface, the ODBC Data Source Administrator.
3. Click the **System DSN/User DSN** tab.
4. Click **Add**.

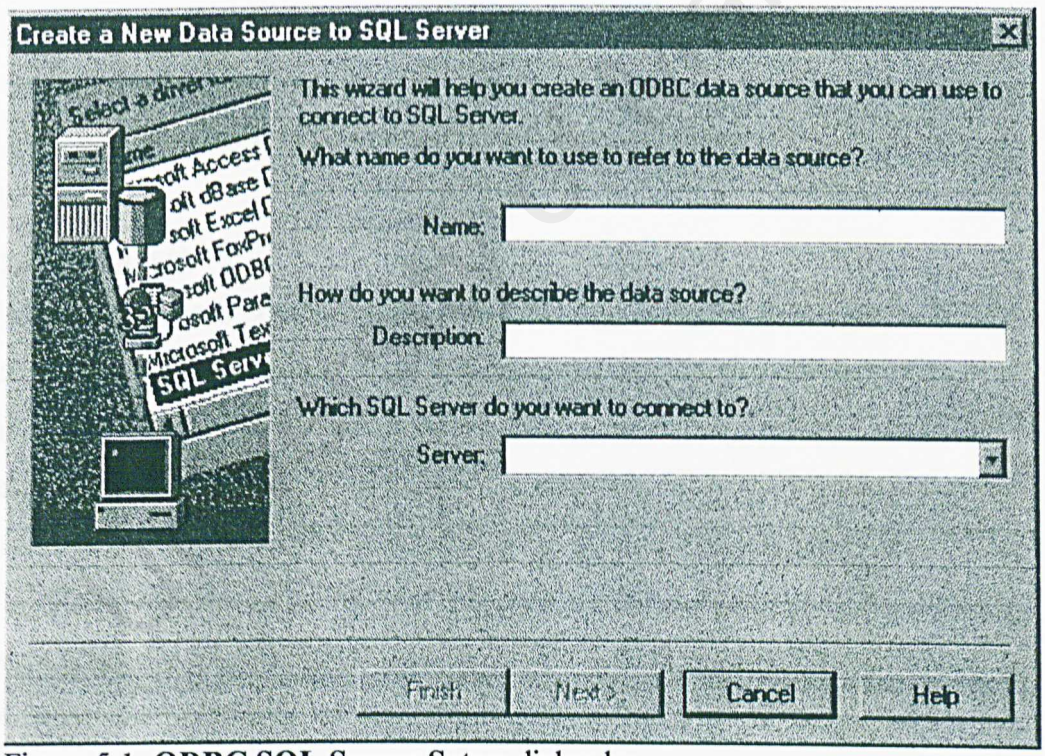


Figure 5.1: **ODBC SQL Server Setup** dialog box

5. From the installed ODBC drivers list, select **SQL Server** (depend on where the data or table store) and click **OK** to bring up the **ODBC SQL Server Setup** dialog box as shown in Figure 5.1.

6. Give your data source a name, such as **BPS**. This can be either the name of the server to which you are connecting, or the name of the database to which you are connecting or some other name that is meaningful to you, such as the purpose of the connection. You are likely to accumulate several data sources, so the name should be descriptive.
7. The Description field is optional.
8. Enter the name of the SQL Server in the **Server** box. The SQL Server should have the same name as the Windows 2000-based computer on which it is running.
9. Click the **Options** button and provide the name of the database on SQL Server to which you want to connect. Remember that several databases with different names can exist on one SQL Server.
10. Finally, click the **Finish** button and you are given a chance to test DSN. It is important to do this before creating the DSN, because it's a quick and easy way to make sure that we supply the right information.

5.1.2.2 Create Database

After create a data source name, the following thing is to create a **BPS** database using SQL Server. At the Enterprise Manager, import all necessary data or table. During the process of importing data, we can set or fix the data type of all the data. A relationship between all the data has to establish. The relationship is very important and useful.

5.1.2.3 Create Cube

The most important part of this program is to build a cube. A cube is a multidimensional structure of data. Cubes are defined by a set of dimensions and measures (Please refer back to chapter 4 for the detail of a cube.) The process of building the cube, happen in the Analysis Manager.

Before design a cube, the important step is to set up a database structure. A database is a structure that holds cubes, roles, data sources, shared dimensions, and mining models together. Then need to connect to the data source that set up earlier in the ODBC Data Source Administrator.

To Create A Cube:

1. In the Analysis Manager tree pane, under the **BPS** database, right-click the Cubes folder, click to **New Cube**, and then click **Wizard**.
2. In the **Welcome** step of the Cube Wizard, click **Next**.
3. In the **Select a fact table from a data source** step, expand the **BPS** data source, and then click **Job_trans_current_2001**.
4. You can view the data in the **Job_trans_current_2001** table by clicking **Browse data**. After you finish browsing data, close the **Browse data** window, and then click **Next**.
5. To define the measures for your cube, under **Fact table numeric columns**, double-click **qty_ord**. Repeat this procedure for the **qty_sold** and **net_amt** columns, and then click **Next**.
6. Click **New Dimension**.
7. In the **Welcome** step, click **Next**.
8. In the **Choose how you want to create the dimension** step, select the type of the dimension (example: **Star Schema: A single dimension table)**, and then click **Next**.
9. In the **Select the dimension table** step (see Figure 5.2), click **Customer**, and then click **Next**.

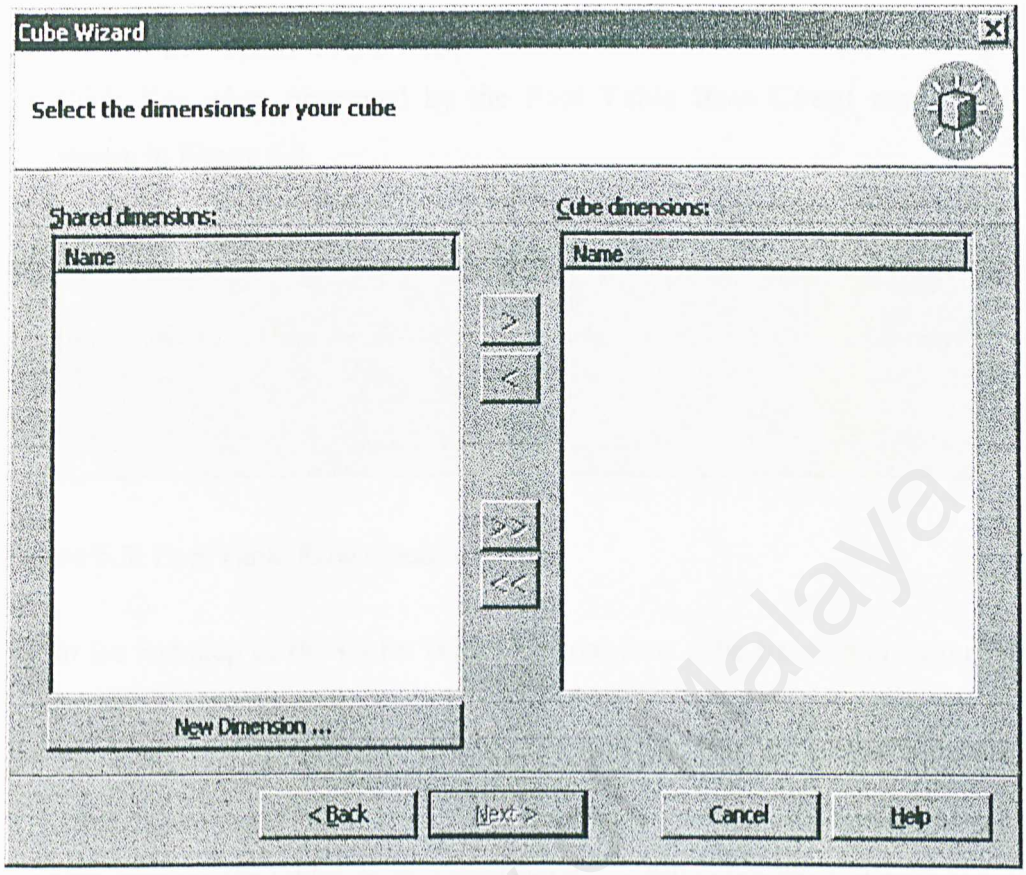


Figure 5.2: Select the dimension table

10. In the **Select the dimension type** step, click **Next**.
11. To define the levels for your dimension, under **Available columns**, double-click the **Country**, **State_Province**, **City**, and **lname** columns, in that order. After you double-click each column, its name appears under **Dimension levels**. After you have selected all four columns, click **Next**.
12. In the **Specify the member key columns** step, click **Next**.
13. In the **Select advanced options** step, click **Next**.
14. In the last step of the wizard, enter **Customer** in the **Dimension name** box, and leave the **Share this dimension with other cubes** box selected. Click **Finish**.
15. In the Cube Wizard, you should see the Customer dimension in the **Cube dimensions** list.
16. Repeat step 6-15 to create the other dimension.

17. In the **Cube Wizard**, click **Next**.
18. Click **Yes** when prompted by the **Fact Table Row Count** message as shown in Figure 5.3.

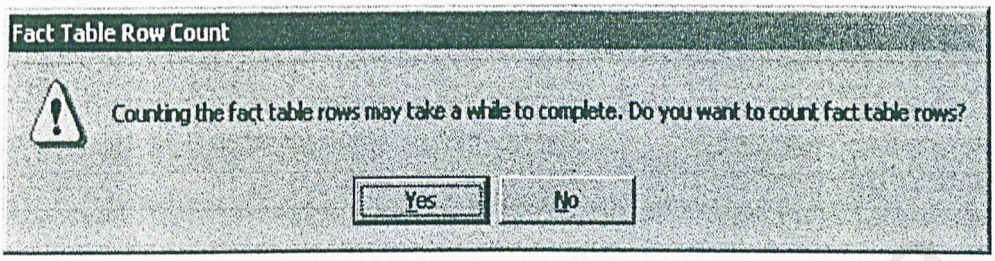


Figure 5.3: Fact Table Row Count message

19. In the last step of the **Cube Wizard**, name your cube **Sales**, and then click **Finish**.
20. The wizard closes and then launches **Cube Editor** (see Figure 5.4), which contains the cube you just created. By clicking on the blue or yellow title bars, arrange the tables so that they match the following illustration.

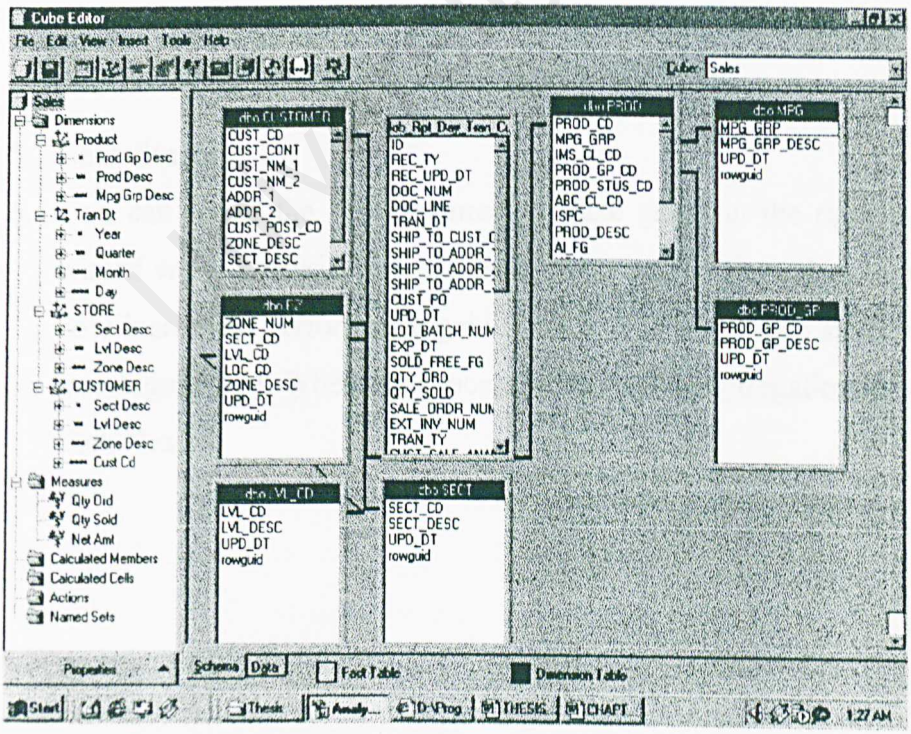


Figure 5.4: A complete Sales cube

Follow the steps above to continue create the target cube, human resource cube and activity based cube.

To Design Storage And Process The Cube:

1. In the Analysis Manager tree pane, expand the Cubes folder, right-click the Sales cube, and then click **Design Storage**.
2. In the **Welcome** step, click **Next**.
3. Select MOLAP as your data storage type, and then click **Next**.
4. Under **Set Aggregation Options**, click **Performance gain reaches**. In the box, enter **40** to indicate the percentage.

   You are instructing Analysis Services to give a performance boost of up to 40 percent, regardless of how much disk space this requires. Administrators can use this tuning ability to balance the need for query performance against the disk space required to store aggregation data.

5. Click **Start**.
6. You can watch the **Performance vs. Size** graph in the right side of the wizard while Analysis Services designs the aggregations. Here you can see how increasing performance gain requires additional disk space utilization (see Figure 5.5). When the process of designing aggregations is complete, click **Next**.

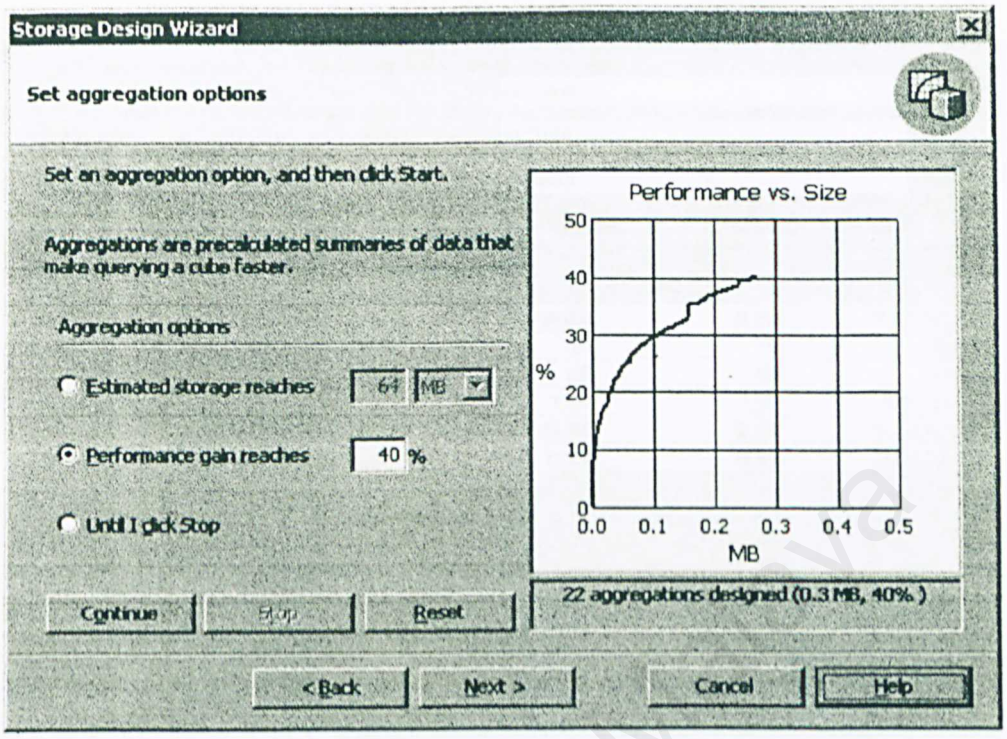


Figure 5.5: Storage Design Wizard

7. Under **What do you want to do?**, select **Process now**, and then click **Finish**.

   Note: Processing the aggregations may take some time.
8. In the window that appears, you can watch your cube while it is being processed. When processing is complete, a message appears confirming that the processing was completed successfully.
9. Click **Close** to return to the Analysis Manager tree pane.

To Browse Cube Data:

1. In the Analysis Manager tree pane, right-click the Sales cube, and then click **Browse Data**.
2. Cube Browser appears (see Figure 5.6), displaying a grid made up of one dimension and the measures of your cube. The additional three dimensions appear at the top of the browser.

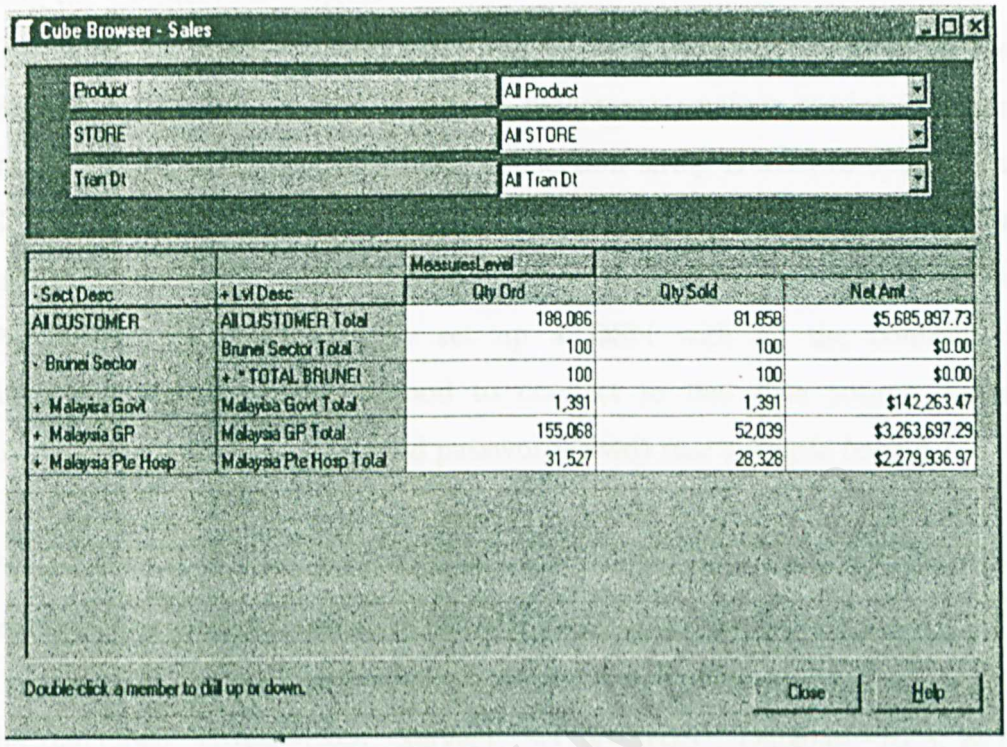


Figure 5.6: Cube Browser

3. To replace one dimension in the grid with another, drag the dimension from the top box and drop it directly on top of the column you want to exchange it with. Make sure the pointer appears with a double-ended arrow during this process.
4. Using this drag and drop technique, select the **Product** dimension button and drag it to the grid, dropping it directly on top of **Measures**. The Product and Measures dimensions will switch positions in Cube Browser.

5.1.2.4 Accessing Database Programmatically

ADO is a set of objects that allow programmers to program their data access logic from languages like Visual Basic as well as scripting languages. The base objects, such as Connection, Recordset, and Command, can be created independently of one another and are related in a hierarchical sense. [2]

Using the Connection Object:

The Connection object is what ADO uses to store our information about the data store connection. In fact, it actually represents a unique session with the data store. One of its key properties, connection string is used to specify the DSN, username and password required and other information necessary for establish a connection to the database and authentication. This is used in the **BPS** project frequently. Simply set up a DSN with all the connection information and use Open method to connect to that data source, while providing the login name (Uid) and password (Pwd) *(see example below).*

```
<%
Set objConn = Server.CreateObject("ADODB.Connection")

objConn.Open "Driver={SQL Server}; Server=SHIS2; Database=BPS;"&
_    "Address=136.79.1.18;    Network=DBMSSOCN;    Uid=sa;    "&_
"Pwd=password;"
objConn.Close
%>
```

Using the Recordset Object:

The Recordset object represent the data that we're working with. We were trying to get data from the data store into a Recordset object. The Recordset object uses objConn for its data connection and then executes the Open method, passing the plain text SQL query as the only parameter. *(see example below)*

```
<%
Set objConn = Server.CreateObject("ADODB.Connection")
objConn.ConnectionString = "DSN=smarthealth; Uid=BPS; Pwd=password"
objConn.Open

Set objRs = Server.CreateObject("ADODB.Recordset")
strSQL = "SELECT * FROM UserProfile WHERE UserID = '" & strUserID
& "';"
objRS.Open strSQL, objConn
……
objRs.Close
objConn.Close
%>
```

5.1.2.5 Using New Mail Object Component

This object is used to send a mail to the targeted email directly, but this object is only can be created on PC with mail-server. This is not a problem for users who used Windows 2000 Server or Client and Window NT Server because both of them have SMTP mail-server. The following is a newmail object.

```
Set sendmail = Server.CreateObject("CDONTS.NewMail")
sendmail.From = "webmaster@bps.com"
sendmail.To = email
sendmail.Subject = "The Login Information You Requested"
sendmail.BodyFormat = 0
sendmail.Body = strbody
sendmail.Importance = 2
sendmail.Send
```

5.1.2.6 Using Built In Server Object

ASP provides a compendium of objects and components, which manage interaction between the web server and the browser. They are called Request, Response, Server, Application and Session, which each deal with a specific aspect of interactivity. Their usage is explained in the following.

Request Object:

This object is used to deal with a request that a user makes – that is, when they ask the browser to see a particular web page or web application. The 'request' might be made in the form of input from an HTML form, or just by typing in a URL.

Response Object:

This object is used to deal with the server's response back to the browser.

Server Object:

This object is used represent the web server itself. Thus, it provides several commonly used functions relating to things that the web server might do.

Application Object:

This object is used to represent the web application. Thus, we can use it to manage things like the contents of the application.

Session Object:

This object is used to represent the user's session, and to store information about that session. We can use this object to manage things like the maximum time that the web server will wait between user 'requests', before terminating the session. [2]

5.1.2.7 Create Active Server Page (ASP) Script

VBScript is the default scripting language on the IIS web server. But it is still perfectly legal to mix languages, as long as they are properly specified for each section of code in the application's page (web page).

There are three different methods to write different bits of the page's content:

- Pure HTML

  for example, Line 1: First HTML line

- ASP code

  for example, <%Response.Write "Line 2: First ASP line
"%>

- Server-side script in a <SCRIPT> tag

  for example,

  ```
  <SCRIPT LANGUAGE=VBSCRIPT RUNAT=SERVER>
     Response.Write "Line 4: First server-side VBScript line<BR>"
  </SCRIPT>
  ```

5.1.2.8 Using ActiveX Control

We will use the ActiveX Control such as Pivot Table (see Figure 5.7) and Pivot Chart (see Figure 5.8) to display result of analysis.

Sales

Drop Filter Fields Here

				Year ▾ Quarter Month				
				⊟ 1997				
				⊞ Q1	⊞ Q2	⊞ Q3	⊞ Q4	Total
Country ▾	State Province	City	Name	Profit	Profit	Profit	Profit	Profit
⊟ USA	⊟ CA	⊞ Altadena		705.1027	834.5261	803.8206	1002.4275	3345.8769
		⊞ Arcadia		663.7447	712.1568	827.8155	887.1444	3090.8614
		⊞ Bellflower		873.1641	970.9348	1027.436	1123.9641	3995.499
		⊞ Berkeley		61.2092	43.003	31.997	54.6711	190.8803
		⊞ Beverly Hills		917.4103	753.2403	825.672	1218.5855	3714.9081
		⊞ Burbank		866.4991	964.3247	957.0228	1166.4396	3954.2862
		⊞ Burlingame		43.4173	132.7417	37.926	32.8757	246.9607
		⊞ Chula Vista		871.741	799.6435	1167.791	952.2141	3791.3896
		⊞ Colma		15.2186	75.2209	33.7144	49.4496	173.6035
		⊞ Concord		18.6814	19.1212	49.0582	46.4281	133.2889
		⊞ Coronado		571.2643	901.3712	749.5822	797.7606	3019.9783
		⊞ Daly City		69.9666	18.3327	41.16	29.6788	159.1381

Figure 5.7: Example of Pivot Table

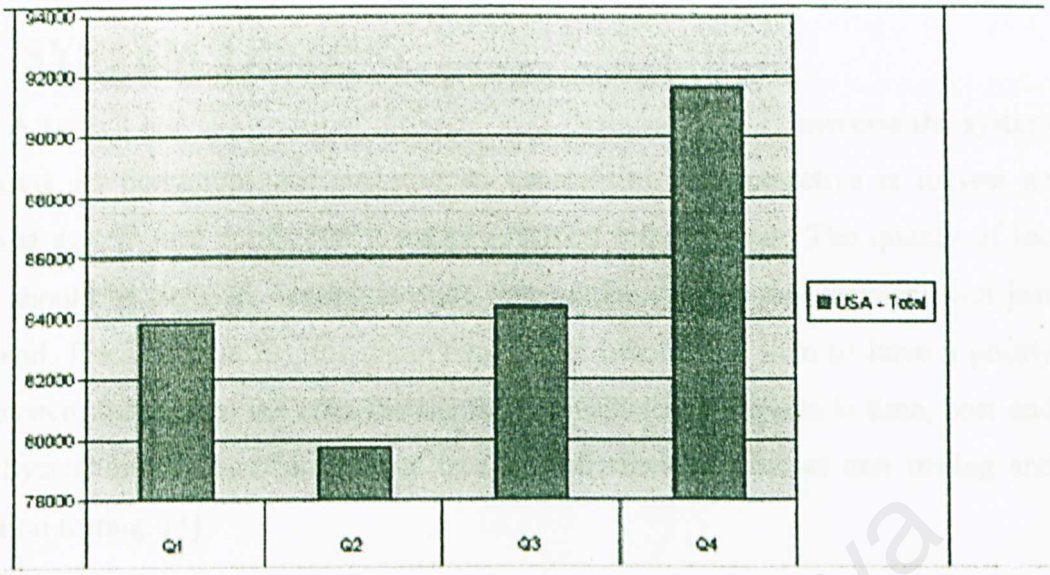


Figure 5.8: Example of Pivot Chart

5.1.2.9 Security Management

All the data and information such as sales transaction, pay roll, financial and customer profiles are private and confidential. Essentially, we just need to allow our users to access data they need, while protecting them from accidentally destroying or damaging it.

So, all the users have to login into the system by using their own registered User ID and Password. ODBC has ability to provide user information on each data source, which can be either NT authentication or SQL authentication. In this case, the latter that requires a User ID and Password for each connection to the data source is used.

5.2 SYSTEM TESTING

A system test is a series of different tests designed to fully exercise the system to uncover its limitations and measure its capabilities. The objective is to test an integrated system and verify that it meets specified requirements. The quality of the system should be defined. Testing is done throughout systems development, not just as the end. Because it is far less disruptive to test beforehand than to have a poorly tested system fail system fail after installation that will result a waste in time, cost and effort. System testing can be separate into several sections such as unit testing and integration testing. [3]

5.2.1 UNIT TESTING

The approach of unit testing is performing a set of basic tests at the module level. None of them are relates to the behavior of the system. Each is related to whether the functions work – especially when they have been executed in each of their potential states. These modules might be performing tasks such as send email, checking data, input data and etc.

Registration module, Login module, Change Password module, Analysis module and Forecasting module consists of several units for testing in this phase. Those units that were independently unit-tested are: open and close connection to database, insert record into database, write back value into database, retrieve data from database, edit data stored in the database, search around database for a particular record, form posting, display returned query result, validate data before submission, error statement and display individual record requested by user in detail format.

Forgot Password module involved testing on SMTP mail server to make sure that password retrieval mail from **Forgot Password module** is sent successfully.

During testing, all programs are checked to verify whether the routine works as it is written. A predetermined set of data to the component being tested and observe what output actions and data are produced. If cannot get the expected output, then we will examine the code by reading through it, trying to spot algorithm, data and syntax faults. Two type of unit testing are applied in this project, there are **white box testing** and **black box testing**.

5.2.1.1 White Box Testing

When using white box testing, it is involves directly with the structure of the code within a module or code segment. White-box test design allows one to peek inside the "box", and it focuses specifically on using internal knowledge of the software to guide the selection of test data (see Figure 5.9). Synonyms for white-box include: structural, glass-box and clear-box. [4]

White box testing also trying to find out any possible class usages that will make the class crash. [5]

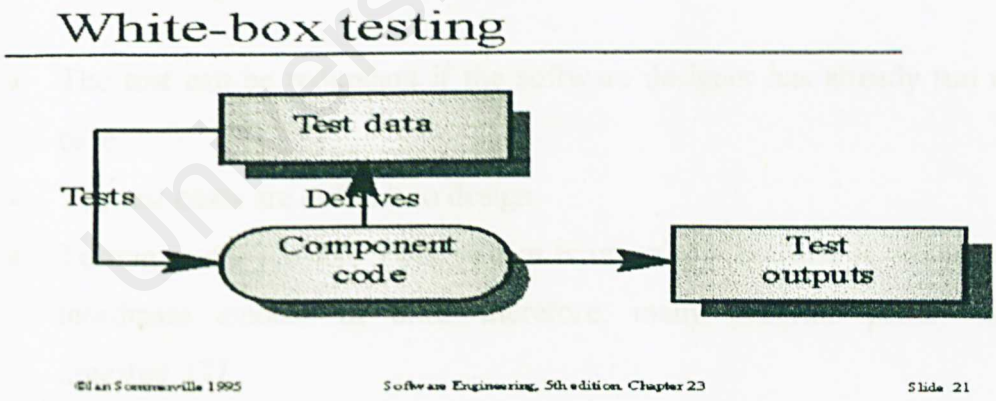


Figure 5.9: Diagram of White Box Testing

5.2.1.2 Black Box Testing

In addition to white box testing there is also black box testing. Mainly, this consists of assuming that the logic structure of the code is unknown – that it is a "black box". This is the point at which the function of a module is tested. Black box testing should not be performed by the author of the program who knows too much about the program internals. black box tests do not necessarily involve the participation of users. [6]

The advantages of this type of testing include:

- The test is unbiased because the designer and the tester are independent of each other.
- The tester does not need knowledge of any specific programming languages.
- The test is done from the point of view of the user, not the designer.
- Test cases can be designed as soon as the specifications are complete.

The disadvantages of this type of testing include:

- The test can be redundant if the software designer has already run a test case.
- The test cases are difficult to design.
- Testing every possible input stream is unrealistic because it would take a inordinate amount of time; therefore, many program paths will go untested. [7]

5.2.2 INTEGRATION TESTING

One of the most frustrating parts of software development and testing is software integration. Normally, the purpose of software integration is to adequately test whether or not the software actually runs as one program. Testing performed at this level is usually black box testing, although it is possible to link processing or logic paths between two modules.

Typically, software integration is viewed from the following nine perspectives:

- Incremental versus nonincremental
- Big bang
- Top down
- Bottom up
- Risk based
- Threaded
- Outside-out
- Inside-out
- Little bang

We will using the bottom up integration to test this system. Bottom up integration start with the lowest-level modules and build up from there. Often, this approach requires a substantial amount of rewriting or reprogramming as we move up the calling chain. [3]

5.2.3 SYSTEM TESTING

This is the final testing stage of a system. It is performed to uncover its limitations, measure it capabilities and make certain that the whole system works according to users' specifications. Developers will join the users to perform this stage of testing where the system is checked against the users' requirements description.

If there is a need for change, system modification will then be carried out if the users' requirements were not met as described in the specifications. If the users are satisfied with the system's characteristics, the system is ready to be delivery for use. The testing result will show whether or not the entire system specifications and objectives are achieved.

5.2.3.1 System Testing Considerations

Because system-testing takes place at a higher level, the testing focuses on behavior rather than function or functional structure. In system testing, not only does the behavior of the individual functions need to be tested; further functional tests must also involve:

a) **The Event List:** All the possible triggers must be exercised and the expected results compared with the actual results. This often requires an 'oracle' to generate expected results to be compared with the actual results.

b) **Specific Scenarios:** The entire set of possible scenarios, or user profiles should be specified for a given application.

c) **Transaction Tracking:** A list of possible transactions, either extracted from the scenarios or from the event list are tracked through the software system to ascertain that they function correctly from 'input' to 'output'.

d) **Screen Mapping:** A map through all the screens through all conditions, which can lead to other menus, is created. Each time a screen or a pull down menu is reached which requires input or generates input, the appropriate functions are processed to see that the functions operate correctly. When they lead to a subsystem or the main system for processing, their input is captured and passed to the subsystem as input—or is retrieved from it as output.

e) **Error Message Testing:** Every error message, which can be generated by the system is extracted from the code and placed in a table to test for appropriateness and understandability. Error message is also checked from a "national language" point of view to see that they have enough room to be translated both in the table and on the screen where they are displayed.

f) **Documentation Testing:** All examples used in the user's manual must be tested for correctness and for whether or not the manual gives the exact answers users will obtain when they run the examples.

5.2.3.2 Fundament Test (Product Verification Testing)

In addition to the functional tests based on functional behavior, as defined above, there are other tests fundamental to all software. Four of these fundamental tests are described next.

a) **Usability:** How easy is it for the user to figure out what to do without resource to a training course or to a user's manual? This usability should be based on building user interfaces that have patterns already familiar to the typical user.

b) **Reliability:** Reliability tests are conducted, according to mathematical models of system reliability, to ensure that the system can be used/operated for periods of time before a fault is detected. Reliability is the probability of some function of the system failing within a specified time. Reliability testing is monitoring the mean time between failures. Reliability and consistency testing go hand in hand where the system behavior (inputs, outputs, response tine) is measured for consistency.

c) **Performance:** Performance tests are conducted to ensure that the system response time meets user expectations and do not exceed the specified performance criteria under heavy stress or volume. During these tests, response time and the transaction rate are measured. The purpose of performance teats is to test-run the performance of various functions of the software within a specified hardware configuration. The performance tests can couple this test with stress testing.

d) **Serviceability:** when the user's software system crashes is it possible for a support person manning the phone line to acquire enough information to enable the maintenance organization to assess the error and fix it without additional information or the need to send a programmer on site? [3]

CHAPTER SUMMARY

In this chapter is describing how the system starts develop. First, will discuss about how to create a database and set the data source name. After that is start develop the main function, which is analysis component. Besides that, also talk about how I used the programming technique, ActiveX Control and security management to develop my system.

After develop the system, is the turn for system test take part. Unit testing, integration testing and system testing is used to make sure that the system perform correctly.

LIST OF REFERENCE

[1] Help in the Microsoft SQL Server—Analysis Manager

[2] David Buser, John Kauffman, Juan T. Llibre, Brian Francis, David Sussman, Chris Ullman,Jon Duckett. "Beginner Active Server Pages 3.0." U.K., Wrox Press Ltd.1999.

[3] Alka Jarvis, Vern Grandall. "Inroads to Software Quality." U.S.A., Prentice Hall: 1997

[4] http://www.comp.lanes.ac.uk/computin/users/tam/CS231/Slides_23.htm

[5] http://www.parasoft.com

[6] http://www.issco.unige.ch/ewg95/node82.html

[7] http://www.webopedia.com/TERM/B/Black_Box_Testing.html

CHAPTER 6

SYSTEM EVALUATION

6.1 SYSTEM EVALUATION

6.1.1 PROBLEM ENCONTERED AND RECOMMENDED SOLUTIONS

As usual, problems always occur during system development and it has been no exception in this project. Throughout this project, many problems have kept unfolding one after another as development work progressed due to many reasons.

6.1.1.1 During Analysis Phase

Since this is the first time to develop a web based system, so it was no experience and limited knowledge in web programming technologies on determine the scope of the system so that it can be completed within the given time frame. However, this problem was overcome after done the literature review and analysis on some of the development tools such as Visual InterDev, Visual Basic, Microsoft SQL Server and ASP technology.

6.1.1.2 During Design Phase

During design phase, the problem that encountered was not enough time to study and detail on all the programming languages and development tools. So that was having problem to produce a good design and solution of the system. Mainly, this was due to inexperience and insufficient knowledge of designing a system. The best way to learn is to read as many approaches used in previous year students' report documentation and suft the information at related web site.

6.1.1.3 During Implementation Phase

a) Problem on installation

There was a lot of problem happen while start installing and configuring the Window 2000 Server, IIS and other development tools. The whole system cannot function efficiently. These problems were solve after a few time on formatting and reinstallation. From experience, it is essential to know the sequence of products installations and the setting of the software. This is to ensure smooth execution without system errors.

b) No prior experience in the chosen programming languages

ASP scripting was written in VBScript. Due to prior inexperience in VBScript and VB, there was a learning curve in understanding how the ASP objects work. Scripting in a new environment such as ASP requires some knowledge of what the ASP objects do and how to use the objects to build the required functionality of the web application. These objects are the server components made available by IIS. The best way of learning ASP scripting during this project was going through some of the ASP examples available on the ASP reference books, Internet and also Microsoft's web site. Those examples are simple and easy to understand how ASP works within IIS.

c) No prior experience in Online Analytical Processing

Cubes are the main objects in online analytical processing (OLAP), a technology that provides fast access to data in a database. It is a multidimensional structure that contains dimensions and measures. Dimensions define the structure of the cube, while measures provide the numerical values of interest to the end user. I faced problem on how to using the analysis services to create a cube. I was no idea on how to related the data and use them to create cube that can use to analysis data. This problem was solve after refer to the tutorial about create a cube.

6.1.1.4 During Testing Phase

a) Unanticipated appearance of web pages in difference web browsers

The appearance of web pages is different on Internet Explorer 5 and Netscape Navigator and Communicator during the testing phase, such as different positioning of graphics, text, and tables on these web browsers. One may appear perfectly in one browser but may not in the other.

b) Unanticipated script error

Runtime error occurs when a command attempts to perform an action that is invalid. For example, the error message that appeared during my development phase is as below:

> *"A Runtime Error has occurred. Line 593*
> *Error:ADO could not find the object in the collection corresponding to the name or ordinal reference requested by the application."*

In fact, the conditions that result in runtime error depends on the language we are scripting with. A condition might cause a runtime error in VBScript but might not cause an error in Jscript.

c) Setting error

When testing on the client side, some of the page cannot be display due to some of the path was not set correctly. This is because when insert a image or a plug in object into the Microsoft Front Page to create a page, this program will automatically set the path as below:

> *File:///D:/Inetpub/wwwroot/folder/file-name.asp*

This problem can be solve by change all the path to:

> *Http://server-name/folder/file-name.asp*

6.1.2 SYSTEM STRENGTH

This system is not as powerful as the existing one in the market. But still has some strength when compare with them.

a) Fast analysis

When user wants to analysis the data in the database, the system will give a fast response time to display the result. This is because my system is using the OLAP application to delivery fast analyzing of multidimensional information to user.

b) Web interface to view data

Create a web interface to view data. This system able user to disseminate information on a 'anytime', 'anywhere' basis because many analysts need analyze the data while disconnected from the corporate network.

c) Analyses data into chart or graph format

After analysts analyze all the data, they will present the information into chart or graph format. This will make users more understanding the analyses result. Using the Pivot Table to display result of analyze is more flexible because user can just drag in what the fields they need to analyze without referring to the whole report. User can also view different type of chart regarding to the Pivot Table. The Chart will change automatically while the data is change.

d) Simple user interface

User interface is simple and user-friendly for users to get accustomed to

e) Transparent

System is transparent, as users do not need to know where the database resides, how the system is structured, etc. For example, users do not need to know how to retrieve from and insert records into the database. All they need to do are submit keywords and then view query results.

f) Error messaging

In **BPS**, the error messages are immediately displayed just a short while user key in the wrong data. This allows the user to identify their error effectively and advice will be given for them to continue using the system confidently.

g) Online add in planning data

User can add in the new data such as new transaction, new product and target.

6.1.3 SYSTEM LIMITATION

Despite some of the system strengths mentioned previously, there are limitations, which cannot be researched and developed due to time constraint. Those limitations are:

a) The forecasting method that is using is not too accurate.

b) Due to the reason this system is using the cube structure, so this system is requires only Window 2000 and above for execution.

6.1.4 FUTURE ENCHANCEMENT

Future enhancement is very essential in the development of web-based system. Here are some future enhancement's suggestions for the BPS system:

a) **Forecasting method**

Use a more accurate method to forecasting. Beside this, the system will automatically generate a report and a chart based on the result of the forecasting.

b) **Printing out the result of analysis and forecasting**

After user get the result of the analysis and forecasting, user can print out the result in a documentation format.

c) **Edit feature**

For the data enter part, user can easily edit the data if they had made any mistake or error. So, can make sure that all the data is accurate and correct.

d) **Encryption and decryption of password**

Password should be encrypted before store in database and decrypted during password retrieval process.

e) **Advanced analysis**

Create more advanced analysis based on the data such as budgeting and other financial applications.

f) **Add role to the cube**

Define which users or user groups can access and query the data in a cube. Each cube role applies to a single cube and indicates the objects in the cube they can access and the kind of access they have to those objects. This can define the security at different levels of granularity in the cube

CHAPTER SUMMARY

This chapter is discussing about the system evaluation. I will list out those problems that I encountered during development of the system included with the solution for those problems. The strength and limitation of the system is listed here. Besides, also included the enhancement that can be do at future.

CONCLUSION

Finally, this project has achieved to deliver the system in time and fulfilled most of the objectives and requirements as determined during system analysis phase. This system is successfully give fast analysis on a large amount of data that is retrieve from the database and also displays the result to the user by using Pivot Table and Pivot Chart. Besides this, also can do forecasting based on the current data. The user is allow to enter new transaction into the database. Users also can update the data and their profile information.

There was a lot of valuable knowledge and experience gained during the development of this system. As the project progressed, so did the number of clearer views on how Internet technologies work, ASP concepts and maintaining and configuring IIS. Learning how to use HTML, ASP, VBScript and so on proved to be a valuable experience and knowledge. And learning the OLAP technology let me know more about using the cube structure to do analysis. Besides, skills in using software such as Visual Interdev 6, graphical software such as Macromedia Dreamweaver, Microsoft FrontPage 2000 and Adobe Photoshop have been acquired.

While programming skills are essential, good practice on software engineering techniques must also be applied efficiently. This project has provided the golden chance to experience using the techniques, paradigms, and approaches learned from System Analysis & Design and Software Engineering courses in the second year and third year study respectively.

Lastly, there is still need improvement in this system. As mentioned before, the system scope was defined through waterfall model with prototype, this makes the system most expandable in terms of functionality. Enhancement could still made to the system with more features added.

APPENDIX A

DATA DICTIONARY

Table structure

Sales data

Table name: Jod_Rpt Day_ Tran_ Current

Primary key: ID

Description: Sales transaction table for current year

Field name	Type	Length	Description
ID	Numeric	9	Identity ID
Rec_tTy	varchar	2	Record type
Rec_Upd_Dt	Datetime	8	Record update date
Tran_Dt	Datetime	8	Transaction date
Sold_Free_fg	varchar	1	Free / sold
Upd_Dt	Datetime	8	Update date
Qty_Ord	Int	4	Quantity ordered
Qty_Sold	Int	4	Quantity delivery
Sale_Ordr_Num	varchar	8	Sale order number
Cust_Cd	varchar	15	Customer code
Cust_Cont	varchar	30	Customer contact
Cust_Nm	varchar	50	Customer name
Addr	varchar	50	Customer address
Cust_Post_cd	varchar	10	Customer post code
St_Cd	varchar	3	Sector code
St_Desc	varchar	25	Sector description
Zone_Num	varchar	3	Zone number
Zone_Cd	varchar	24	Zone description
Lvl_Cd	varchar	5	Level code
Lvl_Desc	varchar	24	Level description
Prod_Team_Cd	varchar	3	Product team code
Prod_Team_Desc	varchar	50	Product team description

Staf_Id	varchar	10	Staff ID
Staf_Nm	varchar	50	Staff Name
Prod_Cd	varchar	23	Product code
Prod_Desc	varchar	50	Product description
Mpg_Grp	varchar	3	MPG group
Mpg_Grp_Desc	varchar	50	MPG group description
Prod_Gp_Cd	varchar	6	Product group code
Prod_Gp_Desc	varchar	24	Product group description
Net_Amt	Money	8	Net sales
Rowguid	Uniqueidentifier	16	Unique ID for replication

Table name: Jod_Rpt Day_ Tran_ 2000

Primary key: ID

Description: Table is similar to Job_Rpt_Day_Tran

Table name: Jod_Rpt Day_ Tran_ 1999

Primary key: ID

Description: Table is similar to Job_Rpt_Day_Tran

Activity data

Table name: Cal_Rpt

Primary key: Cal_Rpt_Id

Index keys: Staf_Id, User_ID, Zone_Num, Contact_ID, Period, Cr_Date

Description: Call report header

Field Name	Type	Length	Description
Cal_Rpt_ID	nvarchar	16	Uniqueid for each call report
Staf_Id	nvarchar	10	Staf ID of sales person
User_ID	nvarchar	10	UserID of the login user
Zone_Num	nvarchar	5	Zone of the contact
Contact_Id	nvarchar	16	Unique id for each contact
Period	nvarchar	2	Period of the year
Call_Obj	ntext	16	Call objecrive
Car	ntext	16	Call activity report

Car_Flag	bit	1	'1'if there is car and '' if none
Sales_YN	char	1	'1' if sales>0 and '0' if no sales
Net_Amt	numeric	9	Number representing sales from call
Cr_Time	smalldatetime	4	Creation time
Cr_By	nvarchar	5	Creation by
Mod_Date	smalldatetime	4	Last modified date
Mod_Time	numeric	5	Last modified time
Mod_By	nvarchar	10	Last modified by
Rowguid	uniqueidentifier	16	Unique ID for replication

Table name: Cal_Rpt_Prod

Primary key: Cal_Rpt_Id+Prod_Cd

Index keys: CallType

Description: Call report products

Field Name	Type	Length	Description
Cal_Rpt_Id	Nvarchar	16	Unique report ID
Prod_Cd	Nvarchar	25	Product code ID
Prod_Det	Nvarchar	50	(unused)
CallType	Int	4	Values are 1,2,3,4
Remark	Ntext	16	Remarks for this product detail
Cr_Date	Smalldatetime	4	Creation data
Cr_Time	Numeric	5	Creation time
Cr_By	Nvarchar	10	Creation by
Mod_Date	smalldatetime	4	Last modified date
Mod_Time	Numeric	5	Last modified time
Mod_By	Nvarchar	10	Last modified by
Rowguid	Uniqueidentifier	16	Unique ID for replication

Table name: Non-Cal

Primary key: Non_Cal_Id

Description: Non-Call Report

Field Name	Type	Length	Dscription
NonCal_Id	nvarcahr	16	Unique Non-Cal user
Staf_Id	Nvarcahr	10	UserID of login user
Date_Eff_From	smalldatetime	4	Date effective from
Date_Eff_To	Smalldatetime	4	Date effective to
Time_Frm	Numeric	5	Time effective from
Time_To	Numeric	5	Time effective to
Non_Cal_Ty	Nvarcahr	1	Non call type
Remarks	Ntext	16	Remarks
Cr_Date	Smalldatetime	4	Creation date
Cr_Time	Numeric	5	Creation time
Cr_By	Nvarcahr	10	Created by
Mod_Date	Smalldatetime	4	Last modified date
Mod_Time	Numeric	5	Last modified time
Mod_By	Nvarcahr	10	Last modified by
Rowguid	Uiqueidentifier	16	Unique ID for replication

Setup data

Table name: Staf

Primary key: Staf_ID

Description: staff information table

Field name	Type	Length	Description
Staf_Id	Varchar	10	Staf ID of sales person
Hd_Staf_Ty	Varchar	10	Head stf ID
Staf_Ty	Varchar	3	StfType
Prod_Team_Cd	Varchar	3	Product team
Loc_Cd	Varchar	3	Location code
Staf_Nm	Varchar	20	Name
Staf_Init	Varchar	50	Initials

Staf_Oth_Nm	Varchar	20	Other name
Staf_Addr_1	Varchar	50	Address 1
Staf_addr_2	Varchar	50	Address 2
Staf_Addr_3	Varchar	50	Address 3
Staf_Post_Cd	Varchar	10	Postcode
Staf_Tel	Varchar	30	Telephone number
Plac_In_chrg	Varchar	50	-
Staf_Fax	Varchar	30	Fax number
Salary	Numeric	9	Salary

Table name: Job_Staf_Targ_RZ

Primary key: Staf_ID+Zone_Num+ Targ_Dt

Description: staff , Zone Target table

Field name	Type	Length	Description
Staf_Id	Varchar	10	Staf ID of sales person
Zone_Num	Smallint	2	Zone
Targ_Dt	Smallint	2	Target Month
Targ_Amt	money	8	Target Amount

Table name: Job_Staf_Targ_RZ_MPG

Primary key: Staf_ID+MPG_GRP+Zone_Num+ Targ_Dt

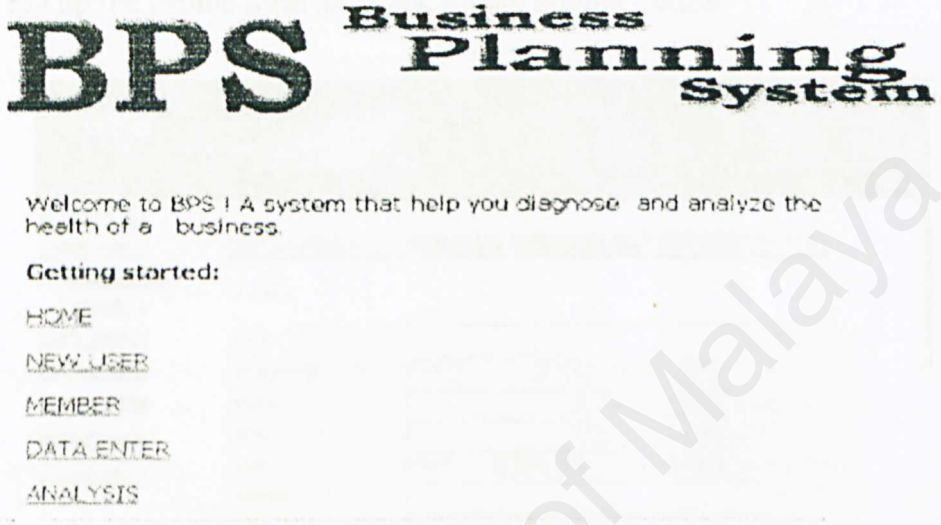
Description: staff , Zone Target table

Field name	Type	Length	Description
Staf_Id	Varchar	10	Staf ID of sales person
Zonr_Num	Smallint	2	Zone
MPG_Grp	Smallint	2	MPG Group
Targ_Dt	Smallint	2	Target month
Targ_Amt	money	8	Target amount

APPENDIX B

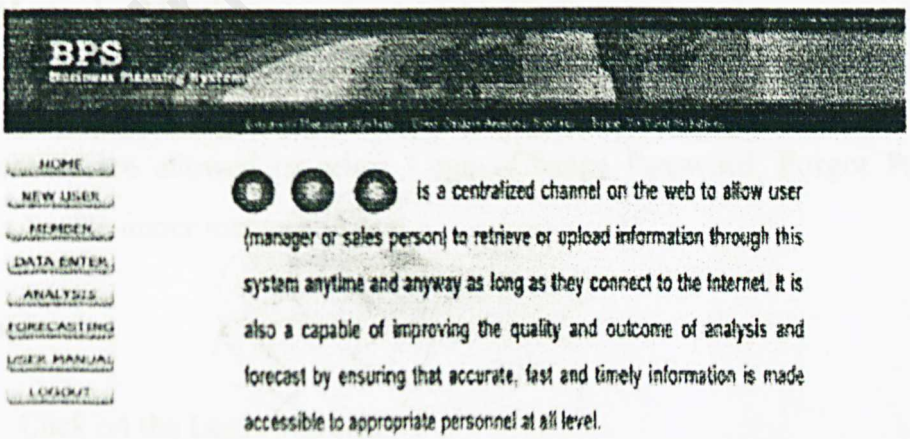
USER MANUAL

When the user click on the User Manual Button, a new window about the user manual will display as shown below. So user can refer to this user manual to continue their work.



User Manual – Home

Home is main page of Business Planning system. All users can access it without login. On this page will briefly describe the definition about business planning system. Other than that will also display the latest notice and announcement about the company to the member of business planning system.

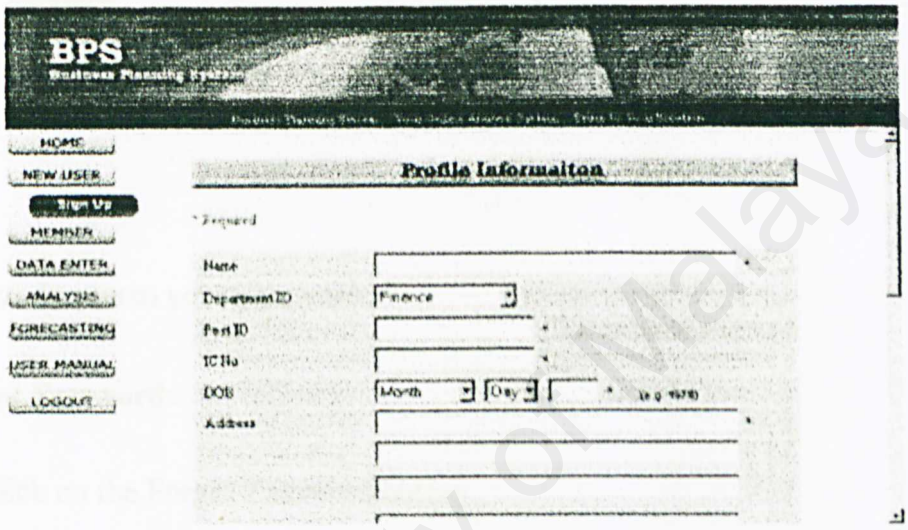


User Manual – New User

All the users have to sign up as a member to access into this system.

Sign Up:

1. Click on the Sign Up button.
2. Fill up the Profile form and click on the Submit button.



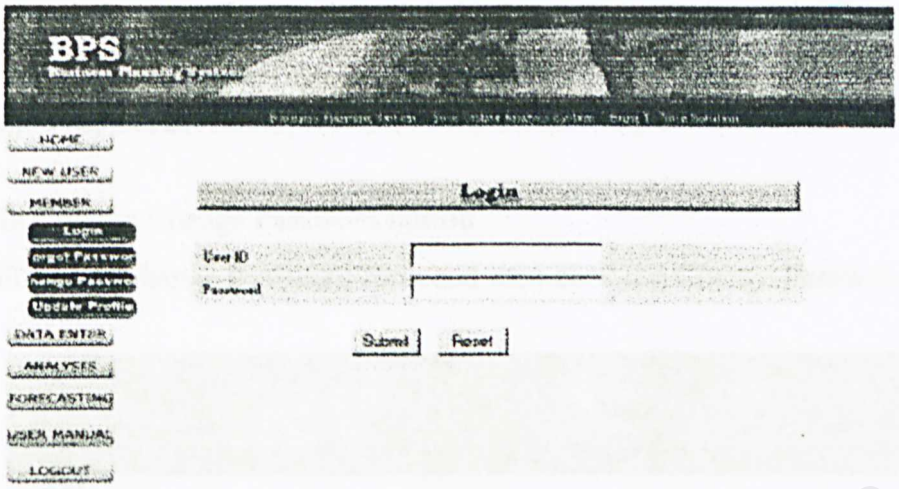
3. After sign up as a member of business planning system, you can access into this system using the User ID and Password that you have registered before.

User Manual – Member

All the member have to login before enter into the system. All registered users or members are allowed to select Login, Change Password, Forgot Password and Update Profile under member option.

Login:

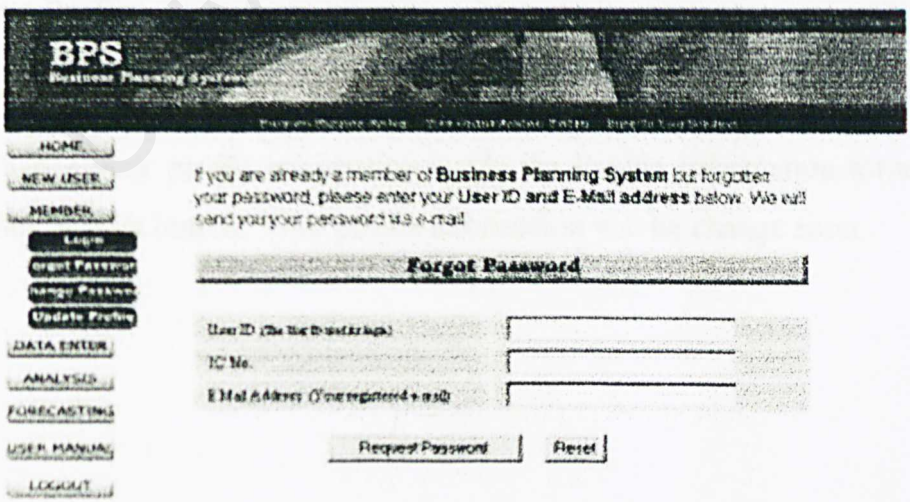
1. Click on the Login button.
2. Fill in your User ID and Password and then click the Submit button. After login, you have the authorization to access any page that you want.



If you have forgotten your password,

Forgot Password:

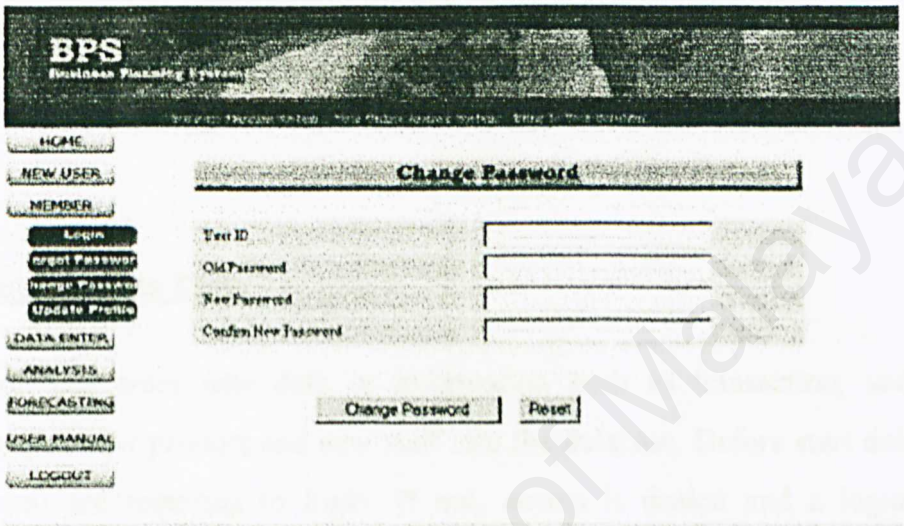
1. Click on the Forgot Password button.
2. Fill in your User ID, identity card number (IC No), and email address and click the Request Password button.
3. Your password will be send to your email account.



If you want to change your password,

Change Password:

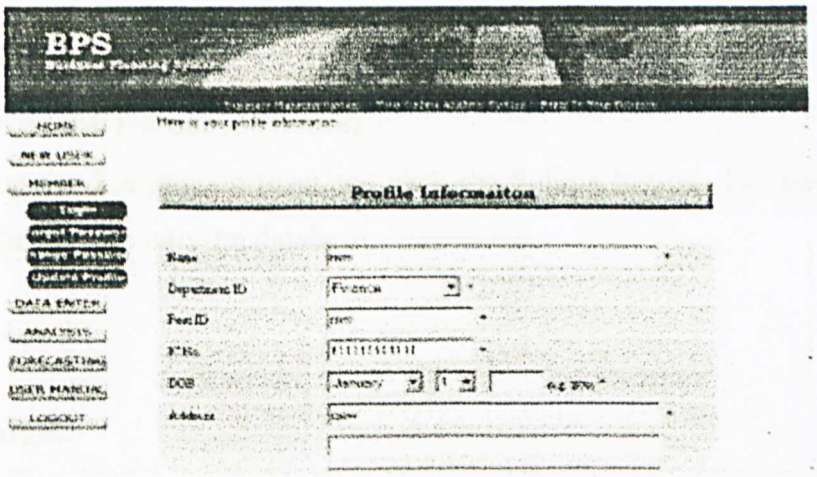
1. Click on the Change Password button.
2. Fill up the Change Password form and then click the Change Password button.



If you want to update your profile,

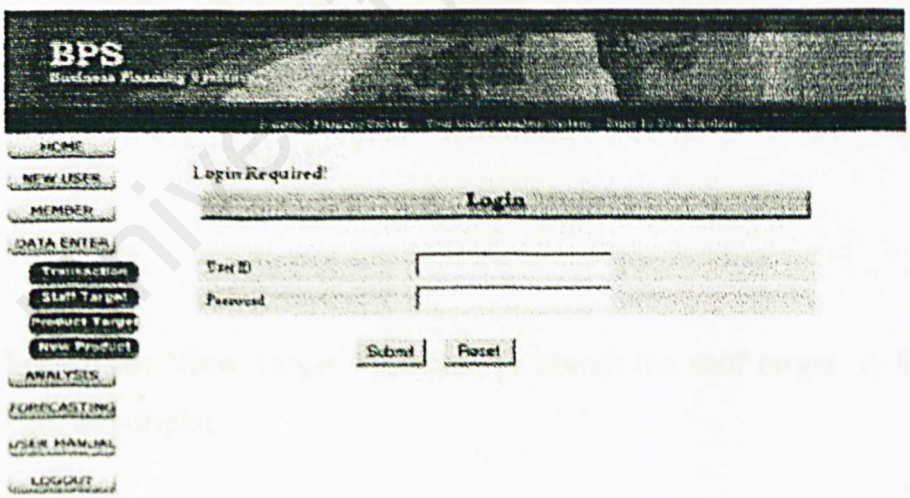
Update Profile:

1. Click on the Update Profile button.
2. Change your profile information inside the Profile Information form and then click Submit button. Your profile information will be change soon.



User Manual – Data Enter

You can enter new data or information such as transaction, staff target, product target, new product and new staff into the database. Before start doing all the process, you are requiring to login. If not, access is denied and a login requires window will display.

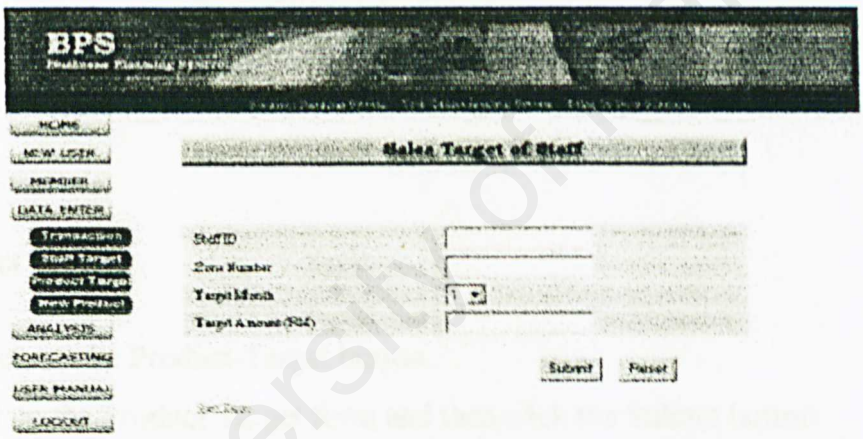


Transaction:

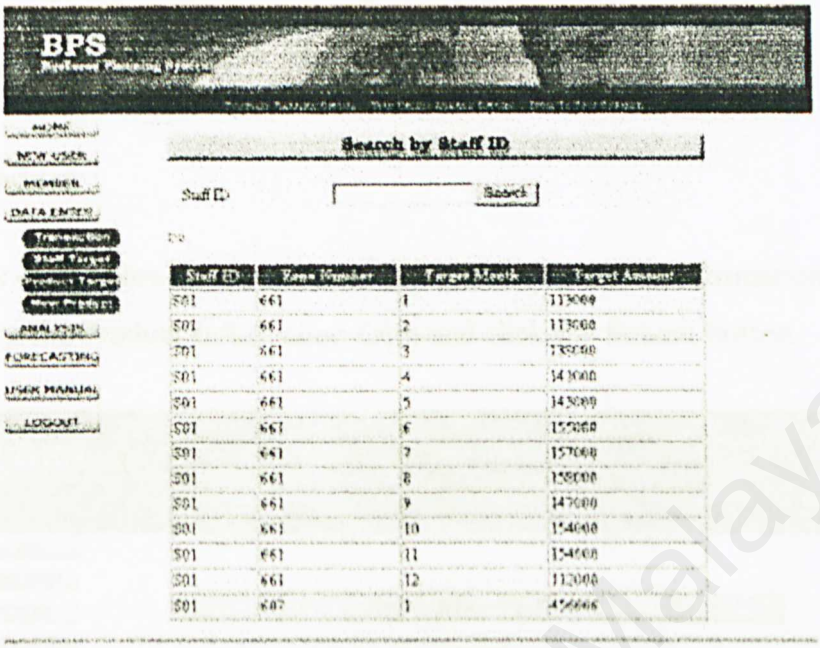
1. Click on the Transaction button.
2. Fill up the Transaction form and click the Submit button. The new transaction will be up date into the database.

Staff Target:

1. Click on the Staff Target button to input new target of staff.
2. Fill up the Staff Target form and click the Submit button.

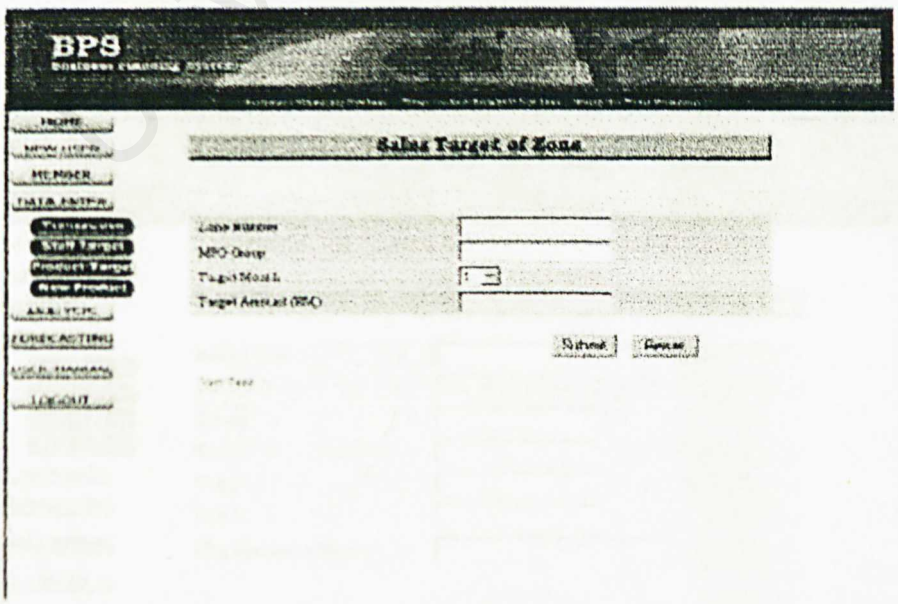


3. Click on the View Target hyperlink, to search the staff target. A list of staff target will display.



Product Target:

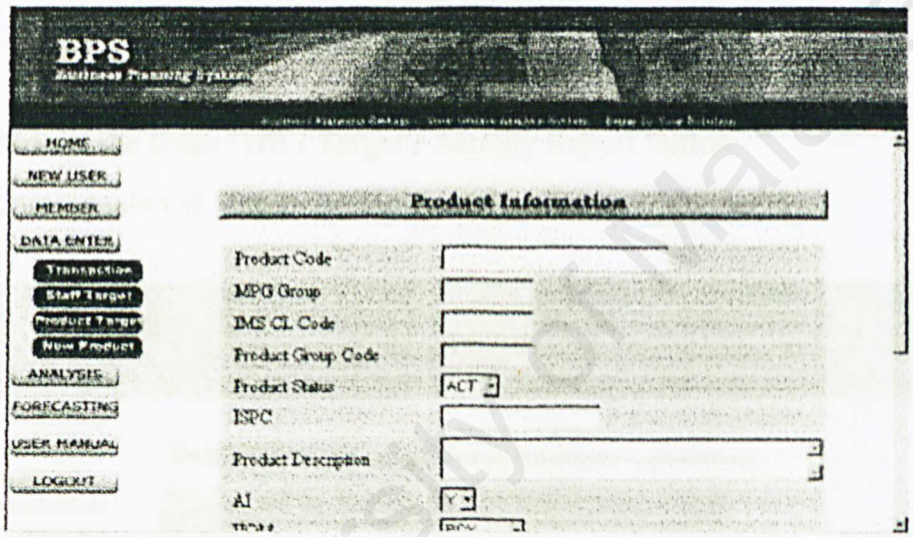
1. Click on the Product Target button.
2. Fill up the Product Target form and then click the Submit button.



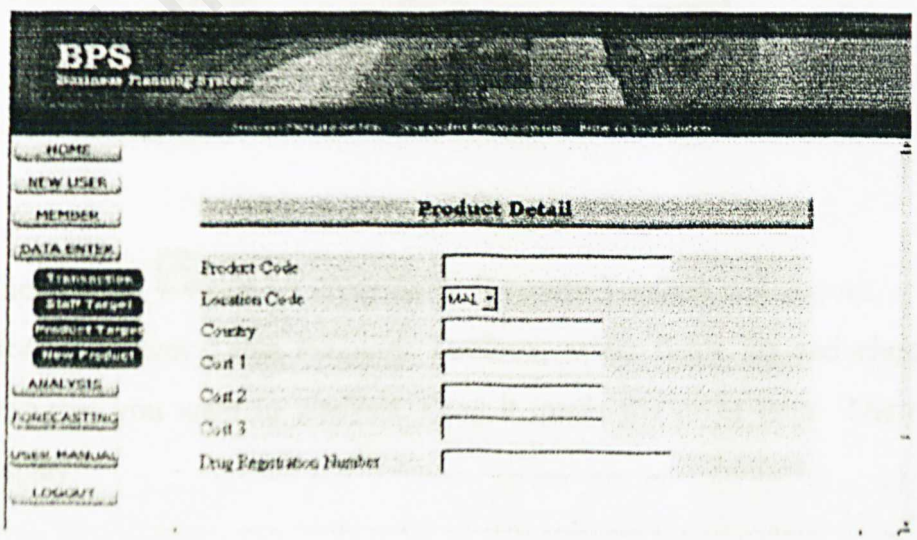
3. Click on the View Target hyperlink, to search the zone target. A list of zone target will display according to the ID search.

New Product:

1. Click on the New Product button to input new product information.
2. Fill up the Product Information form and click the Submit button.



3. After that, continue to fill up the Product Detail form and click the Submit button according to the Zone No search.



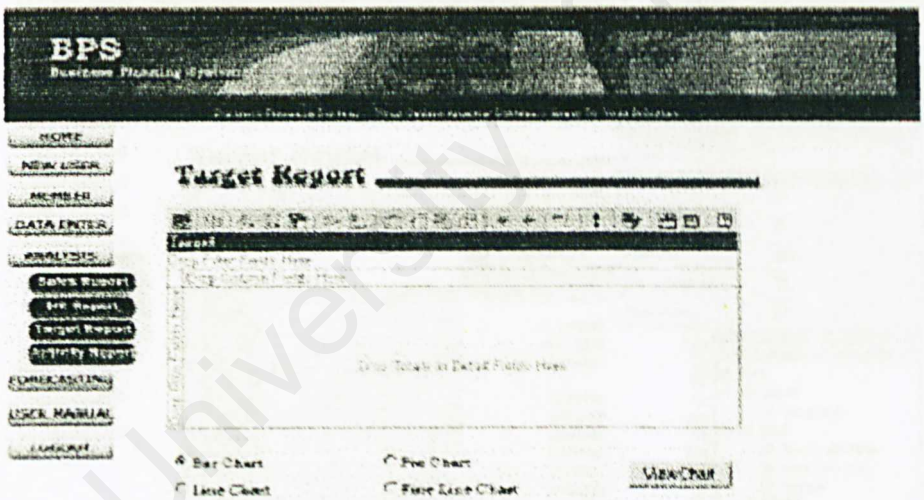
4. After fill up the Product Detail form, continue to fill up the MPG Description form and click the Submit button. All the new information about the new product will keep into the database.

User Manual – Analysis

You can view report such as Sales report, Human Resources report, Target report and Activity report at here.

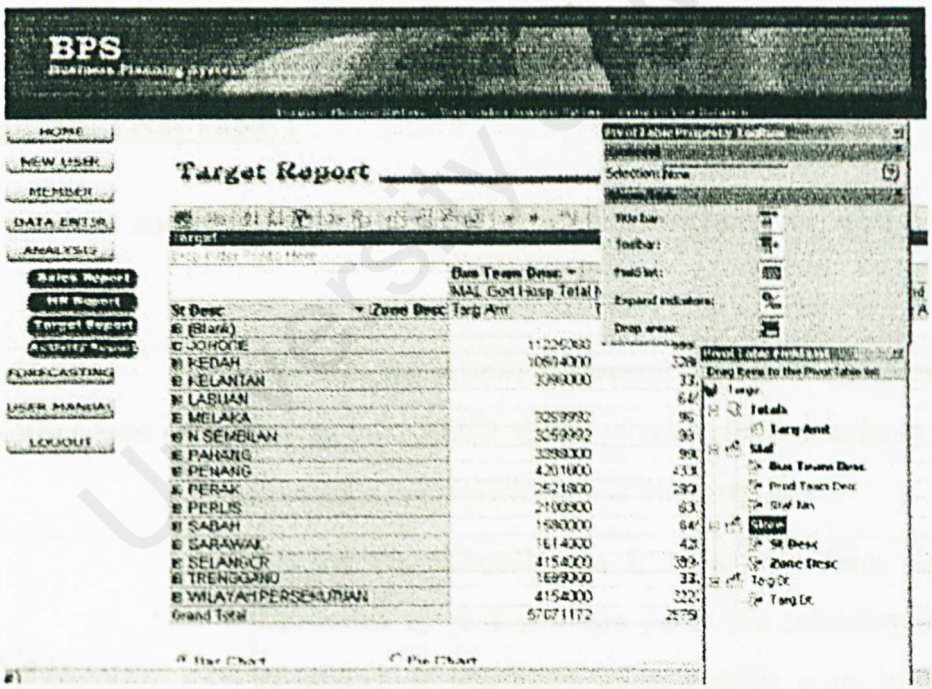
Sales / HR / Target / Activity Report:

1. Click on the Sales / HR / Target / Activity Report button.
2. A pivot table will display on the screen.

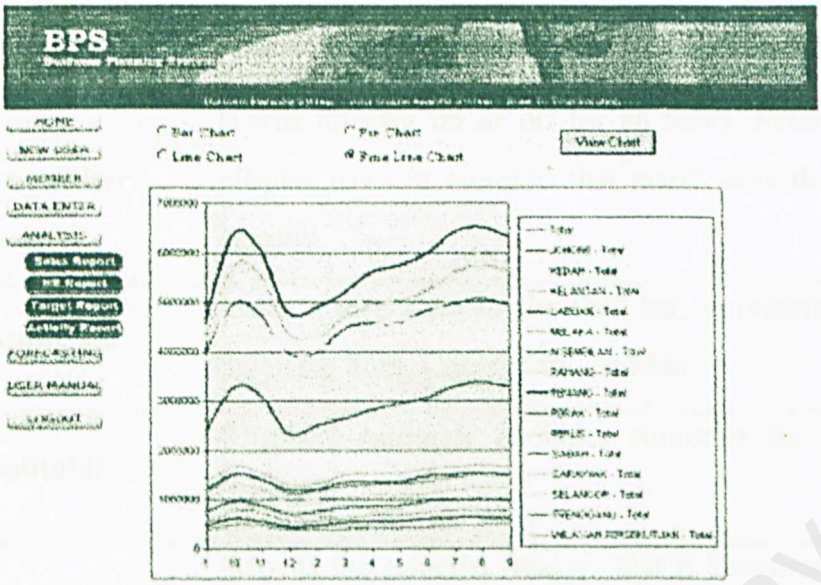


3. Click on the icon. A Pivot Table Property Toolbox will pop up.
4. From the Pivot Table Property Toolbox, select fields list and choose which field that you want to analysis. Drag it inside the pivot table. The result will display.

- From the group of field buttons on the right, drag the fields that you want onto the ROW and COLUMN areas in the diagram. In PivotTable reports, row field and item labels appear on the left. Column field and item labels appear across the top. In PivotChart reports, row field items are plotted on the category axis, and column field items are plotted as data series. For OLAP source data, only fields identified by  at the beginning of the row on the **PivotTable** toolbar can be used as row, column, or page fields.
- Drag the fields that contain the data that you want to summarize onto the DATA area. In PivotChart reports, data fields determine the values for the value axis. For OLAP source data, only fields identified by  at the beginning of the row on the toolbar can be used for data fields. These fields appear at the bottom of the group of field buttons.



5. To view the report in pivot chart form, choose the format of the chart from the radio button and click the View Chart button. A chart will display and show the result.



6. You can change the format of the chart by click another radio button.

User Manual – Pivot Table 1

Toolbar buttons and commands

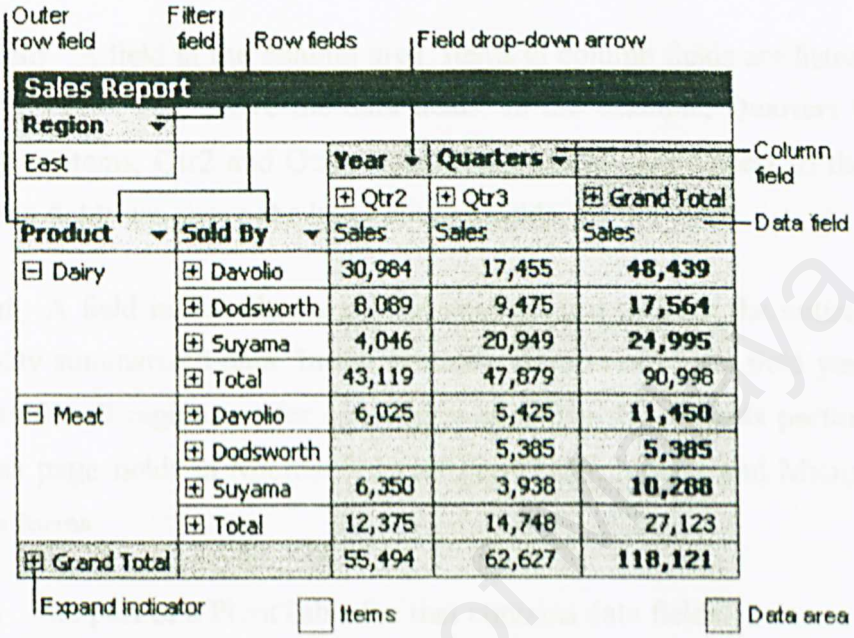
	Office logo cube	Displays the **About Microsoft Office Web Components** dialog box, which shows which version you have installed. The cube spins while data is being retrieved.
	Copy	Copies the selected data from a PivotTable list to the Clipboard so that you can paste the selection into other programs. If you need an interactive copy in Microsoft Excel, use **Export to Excel** instead.
	Sort Ascending	Sorts data in the selected field in A-Z 0-9 order.

	**Sort Descending**	Sorts data in the selected field in Z-A 9-0 order.
	**AutoFilter**	Turns filtering on or off for all fields. Filtering lets you display rows or columns that match only the values you specify.
	**AutoCalc**	Adds a total field to the field list, summarizing the data by using Sum, Count, Min, or Max.
	**Subtotal**	Displays subtotals or hides subtotals for the selected field.
	**Move to Row Area**	Moves the selected field so that it becomes a row field. Row fields summarize data by displaying the items of data in a field down the rows of a PivotTable list.
	**Move to Column Area**	Moves the selected field so that it becomes a column field. Column fields summarize data by displaying the items of data in a field across the columns of a PivotTable list.
	**Move to Filter Area**	Moves the selected field so that it becomes a filter field. Filter fields filter the entire PivotTable list so that only data for the selected item in the field is displayed.
	**Move to Detail**	Moves the selected field to the detail area so that all data in the field appears in unsummarized form.
	**Promote**	Moves the selected inner row or column field to the next outer level. When a PivotTable list has more than one row field or column field, the outer fields are the ones that are further from the data, to the left or top.
	**Demote**	Moves the selected outer row or column field to the next inner level. When a PivotTable list has more than one row field or column field, the inner fields are the ones that are closer to the data, to the right or bottom.

	**Expand or Collapse**	Hides all detail or shows all detail for the selected field, item of data, or cell in the data area.
	**Refresh**	Updates the PivotTable list from the source file or database, displaying any new data and adding any new fields to the field list.
	**Export to Excel**	Runs or switches to Excel and copies the PivotTable list into a new workbook as an Excel PivotTable report, the interactive Excel equivalent of a PivotTable list.
	**Property Toolbox**	Displays the **PivotTable Property Toolbox** so that you can carry out commands and set properties.
	**Field List**	Displays the list of fields available from the source data for the PivotTable list so that you can add more data to the PivotTable list.
	**Help**	Displays Help topics about how to use a PivotTable list.

User Manual – Pivot Table 2

Elements of a PivotTable List



Sales Report				
Region				
East		Year	Quarters	
		Qtr2	Qtr3	Grand Total
Product	Sold By	Sales	Sales	Sales
Dairy	Davolio	30,984	17,455	**48,439**
	Dodsworth	8,089	9,475	**17,564**
	Suyama	4,046	20,949	**24,995**
	Total	43,119	47,879	90,998
Meat	Davolio	6,025	5,425	**11,450**
	Dodsworth		5,385	**5,385**
	Suyama	6,350	3,938	**10,288**
	Total	12,375	14,748	27,123
Grand Total		55,494	62,627	**118,121**

Fields Fields correspond to a column of data and can summarize multiple rows of information. Types of fields include row fields, column fields, data fields, and filter fields.

Item A unique element of data within a field. Items in row and column fields are listed down columns and across rows. The cell where a row and column intersect displays summarized data for the items listed at the top of the column and the left side of the row.

When additional detail data for an item is available for display, the expand indicator appears beside the item.

Row field A field in the row area. In the previous example, Product and Sold By are row fields. Items in row fields are listed down the left side of a PivotTable list. Inner row fields, such as Sold By, are closest to the data area; outer row fields, such as Product, are to the left of the inner row fields.

Column field A field in the column area. Items in column fields are listed across the top of a PivotTable list, above the data fields. In the example, Quarters is a column field with two items, Qtr2 and Qtr3. Inner column fields are closest to the data area; outer column fields are above the inner column fields.

Filter field A field in the filter area that you can use to filter the entire PivotTable list to display summarized data. In the example, Region is a filter field you can use to display data for all regions or for one region at a time. Filter fields perform the same functions as page fields in Microsoft Excel PivotTable reports and Microsoft Access PivotTable forms.

Data area The part of a PivotTable list that contains data fields.

Data field A field that contains summary values from the source data. In the example, Sales is a data field that contains summarized sales amounts — each value in the data field is the sum of the entries from the Sales field or column in the source data.

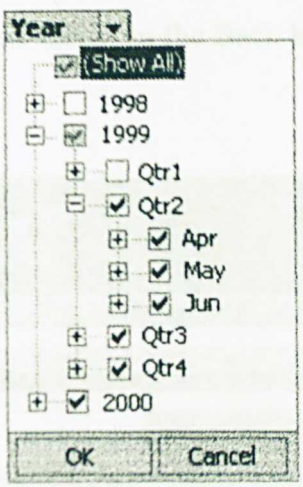
A data field usually summarizes numeric data, such as statistics or sales amounts, with the Sum summary function. In the example, the cell containing 30,984 displays Davolio's total sales for dairy products in the second quarter. This cell combines the sales amounts from every record in the source data that has Dairy for the product, Davolio for the salesperson, and Qtr2 for the quarter.

The underlying data can also be text, summarized by default with the Count summary function, which yields the number of items — for example, a data field that contains the type of the product sold might be used to count the number of dairy sales Davolio made in the second quarter. A PivotTable list can also summarize data using Min and Max summary functions, which yield the smallest and largest values, respectively.

Detail area The part of a PivotTable list (not shown in the example) that can display all of the rows, or records, from the source data. A PivotTable list can display detail rows and summarized data simultaneously. Fields in the detail area are called detail fields.

Field drop-down arrow The arrow at the right side of each field. Click this arrow to drop down a list from which you can select the items to display and hide.

In PivotTable lists that are based on source data from OLAP databases, you can select items at different levels in a dimension field. Date and time fields provide different levels in all types of source data. For example, clicking the drop-down arrow for a Year field might display the following:



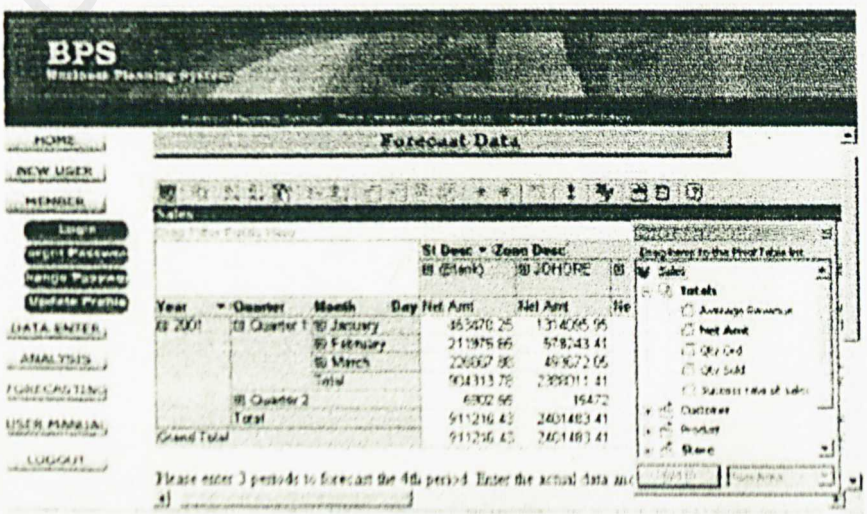
Example of a drop-down list for an OLAP dimension field

The gray check for the year 1999 indicates that it is currently selected for display but some of the items under it are not selected for display. The year 1998 is hidden. Within 1999, Qtr1 is hidden. Clicking **Show All** selects all items in the field at all levels for display.

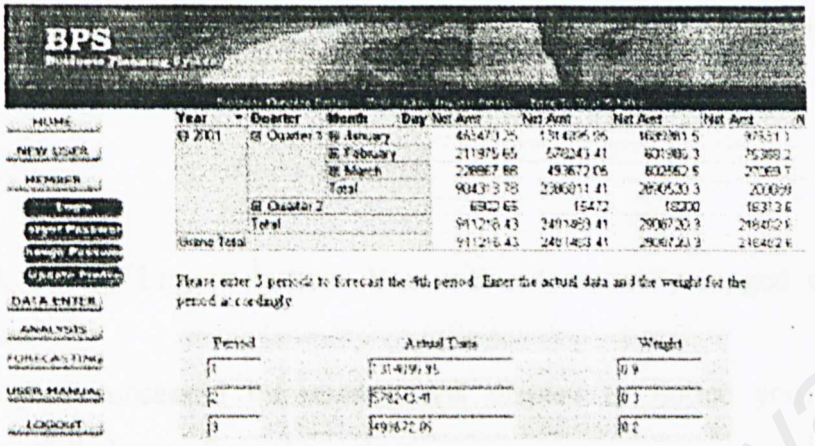
User Manual – Forecasting

This forecast feature allows users to forecast sales amount or sales quantity based on location, sales person, product and so on. The forecast method using here is a short-range forecast named **Weighted Moving Average** where user are required to input the weight of all the data entered according to the period choose before. The weight is the probability of the will repeated or higher weight is more relevant for the forecast. Please note that the total of weight is always equal to 1.

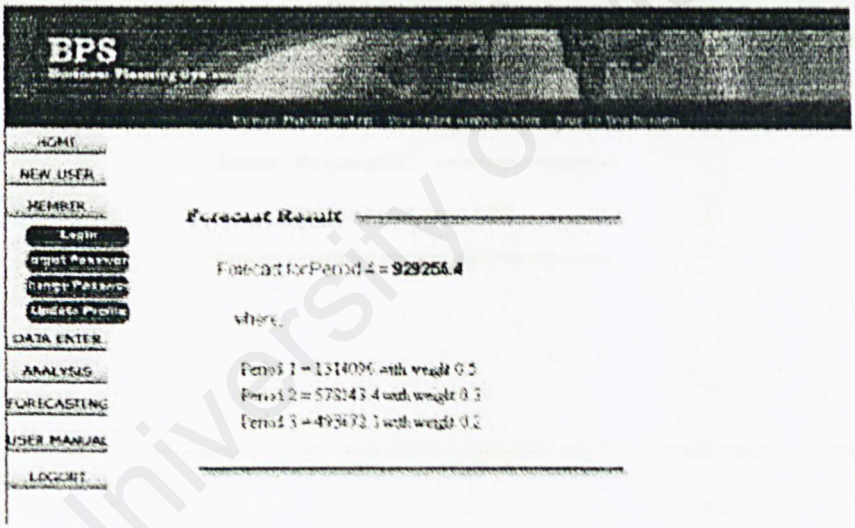
First, retrieve the historical data from the PivotTable based on what to forecast. And then, select the period and the actual data from the PivotTable that to be used in the forecasting. Fill in the form provided and clicks submit. The forecast result will prompt based on the data entered. Click on the Back button to see how display data in the PivotTable.



2. Input the forecast data.



3. The forecast result will be displayed as below:



User Manual – Logout

After finish using this system, you are advice to logout from this system for security purpose.

Logout:

1. Click on the Logout button. You will automatically logged out from the system.
2. A logout successful information will display to notice you that u have successfully logout from the system.