An Electronic Administrative System for Imageintellect

by

Mr. Kerati Charoenwattanachai

A Final Report of the Three-Credit Course
CE 6998 Project

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer and Engineering Management
Assumption University

November 2003

An Electronic Administrative System for Imageintellect

by
Mr. Kerati Charoenwattanachai

A Final Report of the Three-Credit Course
CE 6998 Project

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer and Engineering Management
Assumption University

November 2003

Project Title	An Electronic Administrative System for Imageintellect
Name	Mr. Kerati Charoenwattanachai
Project Advisor	Rear Admiral Prasart Sribhadung
Academic Year	November 2003

The Graduate School of Assumption University has approved this final report of the three-credit course, CE 6998 PROJECT, submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer and Engineering Management.

Approval Committee:

[illegible]
(Rear Admiral Prasart Srib dung)
Advisor

[illegible]
(Prof.Dr. Srisakdi Charmonman)
Chairman

[illegible]
(Dr. Chamnong Jtifigt apanich)
Dean and Co-advisor

(Assoc.Prof. Somchai Thayarnyong)
CHE Representative

November 2003

ABSTRACT

This project is to analyze, design, develop and implement a web application that provides an improvement of administration for ImageIntellect Company.

This report introduces company background, product and service. It demonstrates the analysis of current problems and areas of improvement. Besides, it analyzes the existing hardware and software that can used as a tool for development in the proposed system.

Moreover, this report is designed for the proposed system. It presents by using data flow diagram to make it clear and ease of understanding. It includes database design which explains how data or information is kept. Screen and report design is included to represent interface of the web application.

Besides the analysis of project cost and benefit is included and presented in a form of graph. It shows the payback period of this project also.

The proposed system offers the methodology for implementation by providing system testing, security and control, training and conversion from old to new system.

ACKNOWLEDGEMENTS

This project cannot be possible if lacks anyone of the following listed below. And, no word can be said more than this "Thank you."

First thanks for the writer's family. He would like to say thank you to his father and mother, sister and brother. They encourage him in not only education, but to include other aspects in his life. If he does not have all of them, he guarantees that this project cannot be done.

Next is his advisor, He is Rear Admiral Prasart Sribhadung who provides guidance, consultation and recommendations. He teaches the writer how to make this project successful step by step.

Last but not least are the writer's friends. He would like to say thanks to both undergraduate and graduate school. All of them teach him to know the word "Friend". They may not be the main group for helping this project but they encouraged him on other aspects that cannot be covered in lessons.

Last the writer has no item to all of them to say thank them. The best way that the writer can do for all of them is saying this "Love and Thank you all of you."

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

I. INTRODUCTION

1.1 Background of the Project

Nowadays, there is high competition in every business industry. The way to survive or win in business war, a company must have a competitive advantage such as lower price, high quality, product differentiation, etc. One point that can make company have a competitive advantage is to have a lot of data because that data can be used for sales prediction, product promotion and so on. Raw data comes from internal or external organizations. For internal, it can be employee profile, sales information, production information, etc. External information can be information of competitor, supplier, government or economy.

Although the company has more information, it cannot use it effectively immediately. Therefore, the company should have information management from both internal and external raw data to create competitive advantage in business.

Company can manage that data to be effective by merging with information technology (IT) because IT will enhance the effectiveness of work such as speed, quality, etc for production, quality control, and management.

For example; company presents it by using the Internet technology for management and allocation of its resources together within the organization. Other is using the Internet for promoting itself to public or doing all business right.

So we can see that internet has impact with individual and organization. Trend of using internet has continuously increased respectively. The internet reduces the course of doing business; expand the scope of communication between each other to globalization.

The development of web site or web application has a lot of tools for helping. It can be in the form of graphic interface mode such as Dreamweaver or FrontPage. Another tool is in the form of text editor mode such as Notepad or EditPlus. Moreover, programming language that can be used to develop web application are a lot also such as PER, PHP, ASP, etc. in which each one has the capability or difference from others.

Hence, this project tries to develop web application for helping in organization management, particularly in scope. The programming language that is used to develop is Active Sever Page (ASP) because an organization has already used this programming language to develop web application.

1.2 Objectives of the Project

(1) To design, develop, test and produce a prototype of Electronic Administrative System for Imagelntellect Co., Ltd.

(2) To analyze hardware and software specification appropriate to this system

(3) To design database system to prevent duplicate and create consistency of data.

1.3 Scope of the project

Create and develop a web application with two main features:

(1) Provide information to the public as the front office feature, and

(2) Provide management information for the company's internal staff as the back office feature.

1.4 Deliverables

The completion of Electronic Administrative System for Imagelntellect Co., Ltd. project will be delivered in two formats:

(1) The first format is the document of project including introduction, literature review, system analysis, system design, system development and

implementation, system cost and benefit analysis, conclusion and recommendation.

(2) The second format is prototype of web site, including web site design.

II. LITERATURE REVIEW

2.1 What Is a Web Application?

A Web Application is an application based on the public standards used on the World Wide Web. The Internet and the World Wide Web in particular, has been such a tremendous success that it makes sense to emulate the principles used there, even if your application will never be accessible to the world via the Internet.

All multiuser applications today are client/server applications in one form or another-each user works at a client workstation, and some central server stores the common data. Both the "old-fashioned" client/server applications developed with tools like Oracle Forms and the new Web applications are really client/server applications.

To distinguish between traditional client/server applications and Web applications, we will use another measure: how many layers (or tiers) the whole architecture has?

In traditional client/server application, a client computer handled the user interface, and a database server stored the data. The actual functionality of the application could reside on the client and/or in the databases. This is two-tier architecture.

Client
Database

Figure 2.1. The Traditional Client/Server Application.

Back in 1990, when the World Wide Web was originally conceived at CERN, the European Laboratory for Particle Physics in Geneva, Switzerland, it was intended to serve static files like research papers. The Web thus also originally had two tiers: a browser that needed only to display the file on the screen and a file server (or Web server) that delivered the file on request.

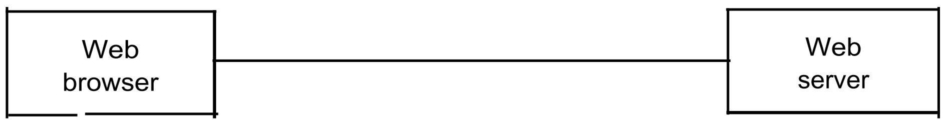


Figure 2.2. Two Tiers Web Application.

This was great; but after a while, Web developers wanted more. They wanted the ability to create dynamic content by running a program that created a customized page on the fly. The Web server was therefore extended with the possibility to call programs through the Common Gateway Interface (CGI).

This was great, but it wasn't long before Web developers wanted to let these programs access databases. This was achieved in many different ways from different programming languages; but because the databases normally resided on other servers than the Web server, there was now a Web browser on the client side, and two or more layers of servers on the server side. This leads to the terms of three-tier and multitier applications.

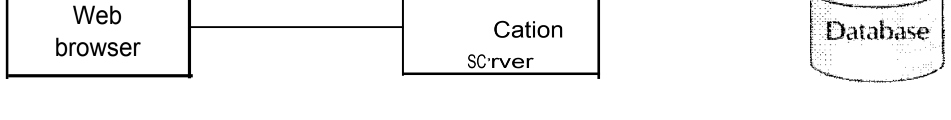


Figure 2.3. Three Tiers or Multitier Web Application.

A Web application is thus defined as a multitier application based on the Internet standards, using a Web browser as client.

<u>**The Internet Standards**</u>

Web applications overcome many of the problems that plague the client/server world, because the communication and presentation of Web pages are based on a number of public standards.

Table 2.1. Public Standards Used on the Internet.

Standard	**Full Name**	**Applies to**
TCP/IP	Transmission Control Protocol/ Internet Protocol	Network communication between client and server hardware
HTTP	Hypertext Transport Protocol	Communication between Web server and Web browser
HTML	Hypertext Markup Language	Page definition
XML	Extensible Markup Language	User-specific definition of content
XFITML	Extensible Hypertext Markup Language	Page definition (latest version of HTML)

TCP/IP

The most basic standard is the TCP/IP protocol. This protocol defines the rules for establishing connects between computers, that is, between the client and server hardware.

IP Numbers

Every computer on a TCP/IP network must have a unique IP number, written as four groups of numbers where each number lies in the range from 0 to 255, like this: 192.168.27.43.

These numbers can be fixed (that is, defined in the computer configuration), or they can be dynamically assigned. In the case of dynamically assigned numbers, one computer is given a pool of numbers to give out to other computers that request one. The protocol for requesting and receiving a number is called Dynamic Host Configuration Protocol (DHCP), and the server that gives out the numbers is called a DHCP server.

Server Names

The IP number allows computer to find one another and to communicate; but we humans prefer names, not just numbers. Therefore, every computer can also have a name. This name is written as three or more words, separated by periods, as shown in Figure 2.4.

The rightmost ("com," in Figure 2.4.) is the top-level domain (TLD), and the second word from right is the second-level domain (SLD).

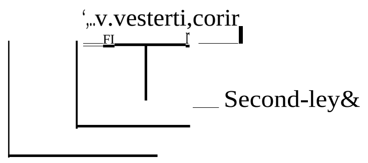


Figure 2.4. Elements of a Server Name.

Top-level domains are administered by the Internet Corporation for Assigned Names and Numbers (ICANN) and exist in several forms, as listed in Table 2.2. In addition to these, a number of new top-level domains are under consideration at ICANN.

In the generic top-level domains, companies and individuals can register their own second-level domain with a registrar accredited by the ICANN. Your Internet Service Provider can do this for you.

To register a second-level domain in one of the special top-level domains, your organization must fit in one of the categories, and you must contact the relevant authority.

Table 2.2. Top-Level Domains.

Type	Domains	Intended For
Generic TLDs	.corn	Commercial organizations
	.net	Networking Organizations
	.org	Other organizations, including nonprofit
Special TLDs	.edu	Higher education
	.gov	U.S. government
	.int	International organizations
	.mil	U. S . Department of Defense
Country TLDs	.uk	United Kingdom
	jp	Japan
	.de	Germany
	.dk	Denmark

From Name to Number

In order for the computer to be able to communicate with a server, it needs to know that Server's IP number. In a small, closed network, you can create a file (called a hosts file) that contains a list of names and corresponding numbers. However, on the Internet, this is, of course, impossible. Instead, the computer that needs to know the number of another computer will contact a special information server — a Domain Name System (DNS) server — that knows about these names and numbers. A DNS server has information about some servers that it is responsible for, and it knows which other servers to contact for information about other servers.

HTTP

While the TCP/IP protocol is the standard for low-level communication between two computers, the HTTP protocol is a higher-level protocol that controls the communication between a Web browser and a Web server. You recognize it as the prefix in full Web addresses

When you enter a Web address in a browser, the browser sends an HTTP message to the server, requesting a Web page. If the page can be found, the Web server responds with another HTTP message containing information about the data type it is returning and, of course, the page itself If the page cannot be found or another error occurs, the Web server returns an HTTP error message.

HTML

The actual Web page returned by the Web server must conform to the Hypertext Markup Language (HTML) standard. This standard defines the allowed format for Web pages.

HTML was based on older work on Structured General Markup Language (SGML), which is a standard for indicating the structure of documents. Therefore, HTML was originally intended to model only the structure of a document, not the layout. That's why HTML initially only contained information like "this is a level 2 subheading" and not "this is Times Roman 24 point bold".

In the Web world, the layout is not strictly defined pixel by pixel. Instead, the page is defined in HTML, and the browser determines how to display the page on the screen. The advantage of this is that the browser will automatically try to compensate for smaller or bigger screens. The disadvantage is that the developer has to give up the exact control over the screen layout.

XML

HTML is very useful for creating Web pages, but it suffers from one serous drawback: The HTML format does not provide any information about the content. If a Web page contains the word Shakespeare, you don't know if the page contains a text by Shakespeare or about Shakespeare.

The way to solve this is to expand the library of tags using Extensible Markup Language (XML). This means that the page will not just contain the word Shakespeare with some optional formatting instruction; instead, it contains the data and a label indicating what the data is.

Table 2.3. Difference between HTML and XML Code.

HTML Code	**XML Code**	**Displays As**
<b>Shakespeare</b>	<author>Shakespeare<author>	Shakespeare

To control how this data displays on the screen, HTML provides basic formatting information like "use bold font." In the XML world, this is handled with Extensible

Stylesheet Language (XSL), which allows you to define how to present the data in an XML file to a browser. This allows a clean separation of data values from their presentation.

The XML standard defines the rules for defining new sets of tags; you can consider XML to be a language for describing languages. It is rapidly becoming the standard for all kinds of automatic data exchange between computers-many organizations are using XML to define standard formats for exchanging invoices, orders, and other types of information. Even the latest version of the HTML standard (called XHTML) is defined in XML.

Intranets, Extranets, and the Internet

The Internet is the network of publicly accessible computers running the Internet standard protocol TCP/IP. Many of these computers run Web servers, so you can connect to them using Web browsers. However, many computers do something else.

An intranet is a private network of computers that also uses the TCP/IP protocol. The difference is that an organization can have its own intranet shielded from the Internet through the use of a firewall. Figure 2.5. shows an intranet for the company with the domain company.com.

The firewall controls all traffic to and from the intranet. It can be configured for example, to allow all clients on the inside to connect to servers on the Internet, maybe barring a few; and it might specify that clients outside on the Internet can access only the server www.company.com, but no others.

An extranet is a network of computers connecting an organization with selected partners, suppliers, and/or customers. In Figure 2.5., for instance, the firewall might allow clients belonging to specific partners to access both the server partner.company.com and the public server www.company.com, but not the internal server intranet.company.com.

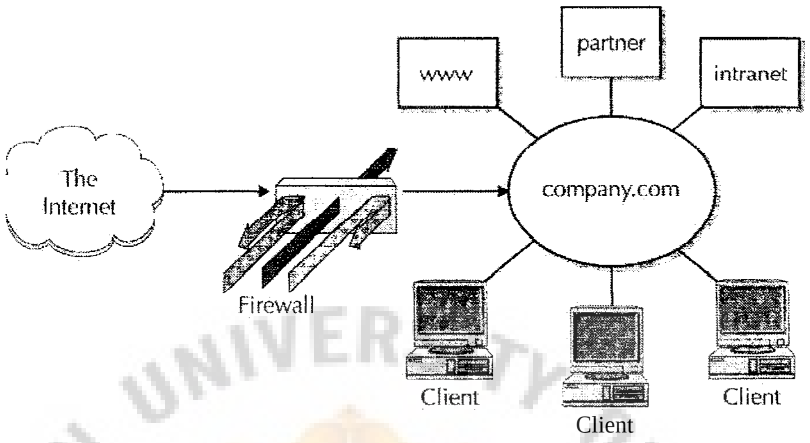


Figure 2.5. Intranets, Extranets, and the Internet.

<u>HTML-Based applications</u>

If you want applications that do not need extensive download and initialization before they start, you must limit yourself to applications that use only HTML.

<u>Ultra-Light Clients</u>

In the most extreme case, your Web application will have to cater to extremely lightweight clients like PDAs or smart phones. In this case, you must use especially restricted HTML-like languages such as Wireless Markup Language (WML).

<u>Pure HTML Clients</u>

In classic HTML-based Web applications, the server simply send HTML pages to the browser and the browser paints the screen. The HTML language has gone through several versions of the HTML language. In addition, some enhancements to HTML were invented by Netscape and some by Microsoft; some of these enhancements later found their way into a version of the HTML standard.

Therefore, some care is needed to ensure that your pages look as intended on both types of browsers and all the versions you decide to support. And if you want to use the later features of HTML, you must accept that some older browsers will not render your HTML page in the manner you intend.

Browser Plug-Ins

The data returned by the Web server to the browser will normally be in HTML format, and the Web browser will display it on the screen. However, you might sometimes need richer data formats than plain HTML. In this case, you can let the server deliver a file in a different format. This is indicated by the Multipurpose Internet Mail Extension (MIME) type information in the HTTP header that precedes the data.

If the browser does not know how to display the data type it receives, it will look for a plug-in program. This is a program created especially for the purpose of handling one or more data types, and normally made freely available for download. Examples of browser plug-ins are the Adobe Acrobat Reader and the Macromedia Flash Player.

Client-Side Scripting

With the help of plug-ins, it is possible to create documents and other types of presentations with a precision not supported by HTML. But for data validation and responding to user events, you need a simple programming language that will execute on the client

Fortunately, browser manufacturers realized this, and the solution is JavaScript. This is a scripting language that allows you to perform simple tasks such as field-level data validation. JavaScript code is embedded in the HTML code and can read and change the individual page elements.

The JavaScript language has gone through the same rapid evolution as HTML and exists in several different versions. Even more than HTML, JavaScript implementation differs between Microsoft and Netscape browsers. So if you want to use JavaScript extensively in your application, you must carefully test your application in as many different browser versions as possible.

Benefits of Web Applications

Now that you know what a Web application is, let's have a look at some of the benefits they offer.

Free Infrastructure

A major benefit is the fact that the whole infrastructure is already in place. This actual software showing the user interface and communication users the standard protocols already configured.

Free Upgrades

In addition, when the application resides on the server, new versions will be immediately and simultaneously available to every user. There is no need to distribute updated application files to every user, and everybody always has the latest version.

Interchangeable Components

Because Web applications are based on published standards, it is, in principle, possible to exchange either the server or the browser without breaking the application.

Focus on Usability

Because Web applications can be made available to everyone with a Web browser, it is no longer piratical (or even possible) to gather every user for a training class. The application must be truly self-explanatory.

This, by necessity, leads to a greatly improved focus on the usability of the application. Not that usability testing is new-commercial software developers have been using this for a long while. But for most in-house applications, it was accepted that the user interface might be somewhat clunky-after all, "the users will get used to it." This was not caused by ill will on the side of application developers-they simply did not have the very different skills needed to design an attractive and intuitive user interface.

This approach normally means more work for the development team in the form of a series of prototypes tested with the users. In effect, because Web applications can have many more users, it pays off to invest some extra development time in order to create a truly useful application.

2.2 Active Server Pages

Active Server Pages (ASP) was developed by Microsoft to run alongside its web server, Internet Information Server (IIS). Both IIS and ASP are designed to tightly integrate into the Windows operating system. II/S is so easy to install and configure that almost anyone can have a Web server up and running in minutes. Whilst the same is true of servers such as Apache, IIS enjoys the benefits of the Microsoft band name. People who would shy away from difficult tasks such as configuring a Web server are reassured and willing to make the effort if the software comes from Microsoft.

If you plan to run a Web server on a specific operating system then making as much use as possible of the facilities of the system is a good idea. ASP lets you do just that. Scripts are run through dynamic link libraries (DLL). Each DLL loads into memory and can service requests repeatedly until unloaded. This makes ASPs very efficient at run-time when compared to traditional CGI scripting. Even more useful, though, is the ability of ASP scripts to access any DCOM, ActiveX, object on the system. Therefore ASP can easily include in organization-sized distributed systems which involve numerous components and which may move far beyond the HTTP/CGI model of computing.

That's all very well but you are probably asking yourselves exactly what ASP is. Put simply, ASP extends the HTML pages by embedding serve-side scripting into the HTML. These scripts are processed by a suitable Web server and the processed page sent to the browser. The Web browser never gets to see the scripts even though they started off inside the page.

Clearly the ASP model has a lot to recommend it. If you are handling static pages which include dynamic elements then you can greatly reduce the processing requirements by first building those static elements. Using CGI the whole page must be built each time that it is required, which may lead to excessive effort when only a few data items are changing.

Because ASP is a Microsoft technology you might expect to have to use one of its languages for the scripting parts of each page. In fact the technology supports any scripting language although most ASP developers will use either Jscript or VBScript. Most texts that you read about ASP use VBScript for their examples but that is purely an illustrative choice. You can write ASP in any scripting language which provides a suitable ActiveX scripting engine to link the Peri interpreter to the IIS Web server. Fortunately the ActiveState distribution of Perl includes just such an engine so we can use Perl to create ASP scripts. That means that not only can you continue to use the same server-side language as for your CGI scripts, but you can compare the technologies and make informed decisions about the merits of each for your projects.

<u>The Request Object</u>

ASP uses its built-in functions to take a lot of the hard work out of handling form data. In the traditional CGI model of Web development each programmer must handle requests from users in their own way. This has given us useful code libraries such as CGI .pm, created by experienced programmers who have distilled the knowledge of he community and created something useful.

Microsoft has done the same with ASP. Like all Web servers, IIS has its own functionality for extracting data from client request whether set with the POST or GET method. Whilst in CGI scripting the data is passed to the application for processing directly from the server, in ASP scripting the scripts receives pre-processed data.

Data sent by a user is packaged as a request object which Perlscript calls $Request. The request object may contain a number of different data items but the most important ones are data about cookies, data sent via POST, and data sent via GET.

The Response Object

In the ASP scripts I have shown so far, you'll have seen a Perl variable call $Response. After reading the description of request objects you might be able to guess what this particular variable is.

The response object controls the transmission of data from the server to the Web browser. Any type of data can be sent back, although obviously you'll normally be sending HTML. You can also configure the HTTP header by handing things such as cookies to it directly from your ASP scripts.

You'll mostly be using the $Response->Write property of this object to dynamically create HTML but there are a few others which you should know about. The response object has a cookie. When setting a cookie for a page you must write it before sending any other output to the page otherwise the browser will simply ignore it.

To make sure that everything happens in the correct order, the response object has a $Response-> Buffer property which you can set to on to cache the page before sending it. Caching is important when you are generating lots of data and want the user to see the whole page rather than just part of it.

HTML pages can have an expiry time. This can be set from the ASP script and is measured in minutes from the time that the browser receives the pages. The expiry time is set through the $Response->Expires property. Finally, browser can be automatically redirected to another site by using $Response->Redirect ($URL).

The Server Object

The server object holds information about the web server itself and lets you use some of its functionality. For instance you can set the maximum that a script will execute before generating an error. This is especially useful if you are accessing databases which may not be present on the system or which may be very busy. It avoids the situation where a single script hogs the server for an inordinate, or even indefinite, length of time.

A method called $Server->HTMLEncode will convert characters such as < into their HTML equivalent such as &lt; automatically — obviously useful when you are processing data from databases for inclusion in Web pages.

The Session Object

HTTP is stateless and has no concept of a session. I've already shown you two ways of artificially creating Web sessions by adding hidden fields to forms and by using cookies. ASP provides a third option called session. This uses cookies but they are set by the server itself, not by the programmer and are valid only for a limited time. Using $Session the following sequence happens:

(1) The user requests a page. Any cookies which are valid for the server are returned along with the page.

(2) If no cookies are sent a new session is created and a cookie is set on the browser.

(3) If cookies were sent they are checked to see if one is an ASP session ID. If a session ID is returned then it is checked to see if it has timed out yet.

(4) If the cookie has timed out a new session created and a new cookie set.

(5) If the cookie is valid and not timed out the existing session is restored from memory.

Obviously there are a great many applications where this sort of background session control is very useful. As with all of these ASP ideas, if you want to know more read the documentation which comes with your Web server and with ActiveState Perl.

The Application Object

Finally we have the application object. This is used to share data within a Web application. The idea of an application is rather than nebulous but is basically developer-assigned set of pages within a site, or set of sites, on one server. In the vast majority of cases you won't ever need to use this object so I don't plan to discuss it any further.

2.3 Database Management System

Introducing the Database

Organizations usually prosper when their managers act on the basis of relevant information. To generate relevant information efficiently, you need quick access to the data (raw facts) from which the required information is produced. Data management, which focuses on data collection, storage, and retrieval, thus constitutes a core activity for any organization.

Typically, efficient data management requires the use of computer database. We define a database as a shared and integrated computer structure that houses a collection of

(1) End-user data, that is, raw facts of interest to the end user.

(2) Metadata, or "data about data", through which the data are integrated.

The metadata provide a description of the data characteristics and the set of relationships that link the data found within the database. In a sense, a database resembles a very well-organized electronic filing cabinet in which powerful software, known as a database management system (DBMS), helps manage the cabinet's contents.

A DBMS is a collection of programs that manages the database structure and controls access to the data stored in the database. The DBMS makes it possible to share the data in the database among multiple applications or users. Figure 2.6. illustrates that the DBMS stands between the database and the user(s). In effect, the DBMS serves as the intermediary between the user and the database by translating user requests into the complex code required to fulfill those requests. The DBMS hides much as of the database's internal complexity from the applications programs that uses the database. The applications program may be written by a programmer, using a programming language such as COBAL, or it may be created through the use of DBMS utility programs.

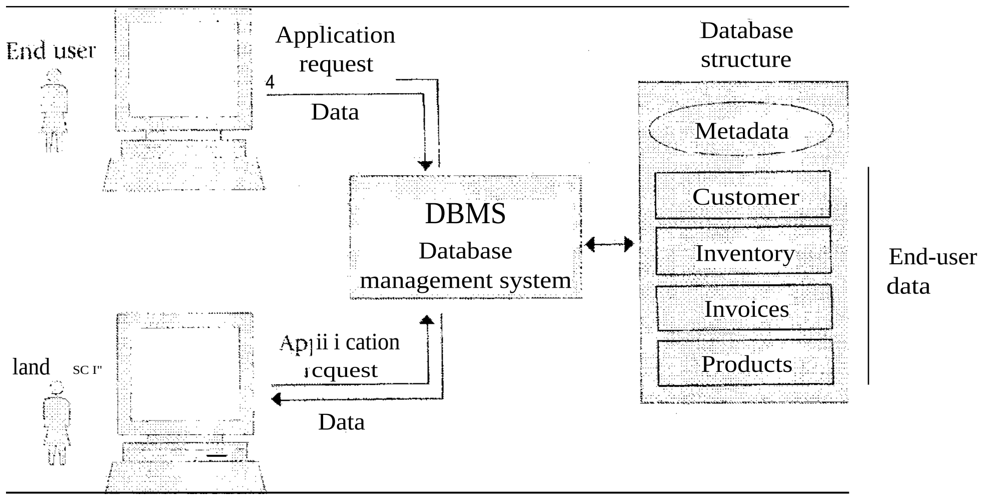


Figure 2.6. The DBMS Interaction between the End User and the Database.

Keep in mind that the DBMS is the database software you buy commercially; you do not have the option to make design changes in the DBMS. Therefore, when we speak of database design, we mean the design of the database structure that will be used to store and manage data rather than the design of the DBMS. Once the database design is completed, the DBMS handles all the complicated activities required to translate the designer's view of the structures into structures that are useable to the computer.

Why Database Design Is Important

A good database does not just happen; the structure of its contents must be designed carefully. In fact, database design is such a crucial aspect of working with databases that most of this book is dedicated to the development of good database design techniques. Even a good DBMS will perform poorly with a badly designed database.

A well-designed database facilitates data management and becomes a valuable information generator. A poorly designed database is likely to become a breeding ground for redundant data, that is, unnecessarily duplicated data. Redundant data are often the source of difficult-to-trance information errors.

A database contains redundant data when the same data are kept in different locations for the same entity. (An entity is a person, place, or thing for which data are to be collected and stored.) For example, data redundancy exists when a customer's telephone number exists in a customer file, a sales agent file, and an invoice file. If the customer's phone number changed, the correction may not be made in all the locations where it occurs. Therefore, the existence of redundant data makes it possible to have uncorrected data entries, and you probably won't know which value is correct. Reports will tend to yield different results, depending on which version of the data was used. In short, uncontrolled data redundancies are typical of a poorly designed database.

A poorly designed database tends to generate errors that are likely to lead to bad decisions. A bad database design is often self-correcting: Organizations using poorly designed databases often fail because their managers do not have access to timely (or even correct) information, thereby eliminating the bad database design!

Because the database plays such an important role, its design is the subject of detailed study in the modern data environment. Database design is simply too important to be left to luck. That's why college students study database design, why organizations of all types and sizes send personnel to database design seminars, and why database design consultants often make an excellent living.

DBMS and Its Features

Database management systems (DBMS) are introduced to alleviate the data-dependency problem and also to remove unnecessary burdens from the application programmer. An application programmer should be able to concentrate on application programming and let utilities take care of data modeling and data-language support.

This is how DBMS came into being. We can define DBMS as a software system embodying

(1) A data model.

(2) A data language for definition and manipulation of data.

(3) Means to enforce and implement security, integrity, and concurrency.

The data-definition language (DDL) and data-manipulation language (DML) are the means of communication, in addition to logical data-structuring (data modeling), the user uses in dealing with database. The term schema is used to convey both a data structure and the facility used to represent it.

As can be seen from the figure, the internal schema converts logical structure and DML operations to physical data structures and file operations to manipulation the physical data. It is in this way that data independence can be achieved. Unlike file systems, in DBMS, physical data are not duplicated among/between users. They are a common pool of sharable data. They can be shared because the internal schema provides a general interface to all the users. In such an environment, we need a caretaker and an administrator, the database administrator (DBA), to take care of the problems of the user community and to manage the database to best fit their requirements. In that respect, the responsibilities of a DBA include database creation and maintenance, database security and integrity, system monitoring, and performance tuning. To achieve these, the DBA is responsible for the design, implementation, and control of database environment, and for the enforcement of database security and integrity.

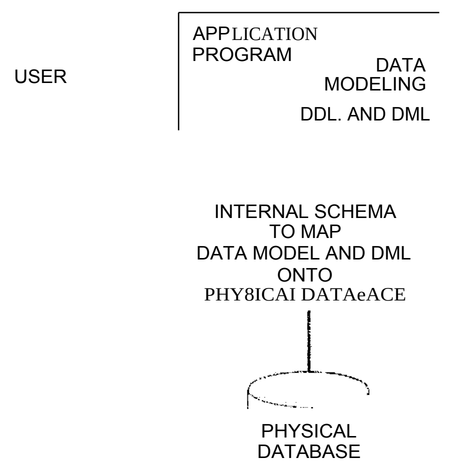


Figure 2.7. DBMS Architecture.

A DBMS must meet certain standards set forth by the community of its implementers and users. It must preserve data independence and data consistency. Because a DBMS can be shared, users can perform concurrent updates on the shared data that may nullify others' updates and/or present inconsistent data to others. If concurrency control in DBMS is not enforced, for example, agents operating on remote terminals could sell more seats than there are available on a flight, a depleted inventory item could be sold and a bill generated for it, etc.

The data languages supported by a DBMS must be universal in the sense of meeting all semantic and practical needs with the power of the mathematical operations and utility features in them. These languages must be usable both as standalone query languages and as data sublanguages (DSL). A DSL is a DDL/DML that can be embedded in a general-purpose programming language such as COBOL, PL/1, Pascal, C, etc. The general purpose programming language is then referred to as the host. An embedded system is useful in application programming, whereas a query language is used mostly to handle ad hoc requests from a database.

As can be seen in the figure, the database-related code segments (shown as a DSL statement) are put in the form of calls to a DBMS that returns both the answer and status to the data buffer of a host program.

Database Design Lifecycle

Database design covers the "lifecycle" of a system. It is not only the process of determining the initial database structures, but also the process of programming user applications and the maintenance and evolution of the database. The process is iterative.

The six phases of the database design lifecycle are:

(1) Requirements analysis

Strategic planning, business model, implementation plan, data flow modeling.

(2) Design of conceptual structure

Entity modeling, functions specifications, access graphs, entity-relationship modeling, view integration.

(3) Design of logical schema

Reducing denormalized conceptual objects to relations, converting entity-relationship diagram to relational schema, refining relational schema by normalization.

(4) Design of physical schema

Access plans, placement strategies, indexes, external storage estimates, controlling access path selection.

(5) **Programming user applications**

Choosing programmatic interface, programming in embedded SQL, application generation.

(6) **Testing and maintenance of database system**

Structured walkthroughs and inspections, housekeeping, monitoring, auditing, perfective maintenance.

2.4 Entity-Relationship Model

One technique which does allow the design of data structures on the basis of processes (business functions) is known as entity-relationship (E-R) modeling. In this approach, information being modeled is grouped into entities or relationships among implementation issues. Though derived to satisfy the user requirements (processes), the E-R design does not show processes in any explicit way, instead, it offers a diagrammatic conceptual database structure. The structure integrates and views on data and needs to be further converted to logical and physical schema corresponding to the chosen DBMS.

Entity

An entity is an object which can be distinctly identified and is important to the information system of the organization. The entity represents a business fact or rule and is usually expressed as the subject or object of a sentence that defines that fact or rule. It is common for the entity name to be a noun. Examples of entities are COURSE, RESEARCH REPORT, COMMISSION, EMPLOYEE, INVOICE and ITEM.

It is important to distinguish, at all times, between an entity set (type) and entity instance (occurrence). For example, COURSE is an entity set and CSCI-335 (Databases) is an instance from the entity set COURSE. Clearly, entity instances must be uniquely identifiable. When the meaning is obvious from the context, we simply use the word entity for both instances and sets. In the diagrammatic representation of a conceptual design, entity set are visualized by means of rectangular boxes. Entity instances, though not represented graphically, give semantics and interpretation to the diagram.

How to organize information into sets and instances is a difficult design decision. An abstraction process called classification/instantiation is involved here. Classification groups entities sharing common characteristics and holding uniform conditions into a class (which we call entity set or type). The reverse of classification is called instantiation when entity instances (occurrences) are obtained.

Relationship

A relationship is an association between entities from different or identical sets. Entities are associated for business reasons. Relationships can usually be denoted by a verb. For example, the entity COURSE may be associated with the entity TEACHER via the relationship TEACHES. An entity EMPLOYEE can be involved in a relationship IS_MANAGER_OF to indicate a multiple-level managerial structure of employees.

As in the case of entities, a distinction is made between a relationship set (type) and a relationship instance (occurrence). For instance, TEACHES is a relationship set and leszek teaches CSCI-335' is an instance of the relationship set TEACHES. Relationship instances must be uniquely distinguishable. There are several different methods of graphically representing relationship sets. In the most restricted representation, relationships are indicated by means of lines connecting entities (with or without relationship names written along the lines). However, a frequent method of representation is to use diamonds or oval boxes to connect entities.

Attribute

Entities and relationships have properties which can be described as attribute-value pairs. For example, are_of _expertise might be an attribute of a TEACHER and knowledge bases could be the value of this attribute. Another attribute of the TEACHER may be years of teaching experience, for which the value may be 15. For each entity or relationship instance, each attribute will be assigned a value. If all values of a certain attribute (or set of attributes) are different, then they can serve to uniquely identify entity or relationship instances such as attribute is known as an entity (or relationship) identifier (key). An entity or relationship can be many unique identifiers (e.g. course_number and course name can be two equivalent identifiers of the entity COURSE).

Attributes can be simple or group. A simple attribute can have only atomic values and does not have an internal structure. Sometimes, the internal structure is a matter of semantic interpretation and processing requirements, for example, ADDRESS can be a group attribute in one design and a simple attribute in another. If only the full address is required in processing, then ADDRESS can be designated a simple attribute. However, if a part of the address is explicitly required for processing (in queries like, List all students from a given city), then ADDRESS should be made a group attribute (with city name being a simple attribute within that group).

2.5 Data Flow Diagrams

Data flow diagrams (DFD) are a hallmark of structured analysis and design. They enable a graphical, concise and partitioned representation of any analyzed or designed information system. DFDs describe the system at the logical level. The physical characteristics of the system, including the computerized processing environment, are not visualized in the DFDs. As a result, the DFDs are complemented by other methods and techniques, such as document flowcharts, conventional system and program flowcharts, decision tables, structure charts or Nassi-Shneiderman diagrams.

Symbols in Data Flow Diagrams

DFD consists of five basic elements:

(1) Data flows

(2) Processes (data transformations)

(3) Data stores (files)

(4) External entities:

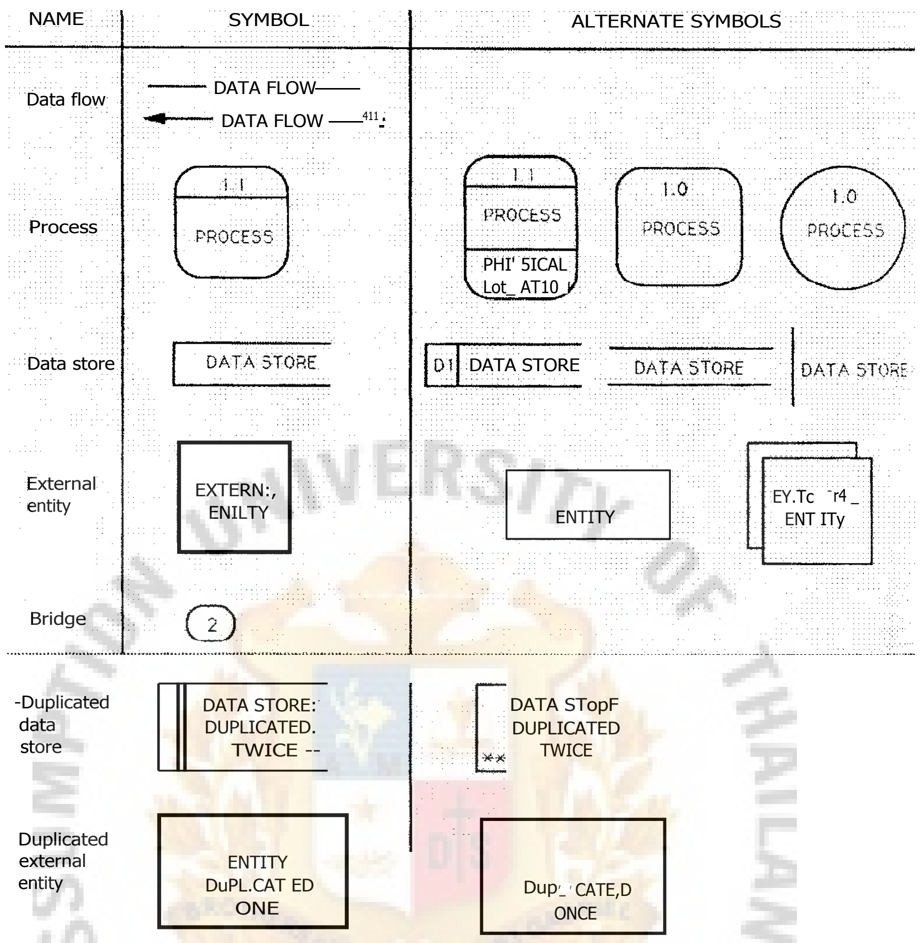
(a) Data Sources (originators)

(b) Sinks (terminators, receivers)

(5) **Bridges** (connectors)

Typical graphical representations for DFD elements are shown in Figure 2.8. To enhance the diagram layout, data stores and external entities can appear more than once in the same diagram. Simple graphical extensions to DFDs allow easy recognition of the duplicated objects. One common extension uses extra lines inside the data store or external entity-n lines for n repetitions (not including the first). Another solution is to include asterisks (*) in the repetitious graphical symbols-n asterisks for n repetitions.

Table 2.4. Graphical Representation of DFD Elements.



Data Flows

A data flow is a packet of data which moves from one element of the DFD to another. The packet can be a document, an arbitrary grouping of data or an elementary data (i.e. not further decomposable).

Processes

A process transforms incoming data flow(s) into outgoing data flow(s). It is represented graphically as an oval or circle containing an identifying number and a label. Optionally, the process can also identify a physical location (or person or program) where it is performed.

Data Stores

A data store holds data. It accepts and releases data flows as required by processes. It can be represented graphically in a number of ways, for example by a rectangle with one short side open. Optionally, the data store can have an identification symbol in a box on the left-hand side of the rectangle.

External Entities

An external entity can be a data source or a data sink (or both). A source is an external system which supplies some data flows to be processed in the system (as determined by the scope of the DFD). A sink is an external system which receives some data flows from the DFD. An external entity is represented graphically by a labeled square or rectangle, possibly stylized.

Bridges

A bridge establishes a special connection from a lower level child diagram to a higher level parent diagram. A special aspect of this connection is that the bridge identifies an uncle process (or aunt, if you prefer).

III. SYSTEM ANALYSIS

3.1 Organization Background

ImageIntellect Co., Ltd. is an in-house web development company. The company is a Small and Medium Enterprise (SME). ImageIntellect offers expertise in developing web sites, databases and image production. With offices in the UK and Thailand we are uniquely positioned to offer a range of expert technical skills at cost effective rates. ImageIntellect is a British owned company, which can offer you the benefit of price advantages without neglecting personal service or quality.

This company deals with IT and computer technology directly. Therefore, major activity is creating and developing web application that used in business solution. Web application that is developed will respond to customers' needs. Each web application will differ up to customers' requirement.

Target market of company is any business company that would like to have web application to use in their business solution. Customers of this company are foreigners. It emphasizes on European nations, especially United Kingdom.

The communication within the company uses intranet via Local Area Network (LAN). Staff must use computer as a base for running business and data communication from one to another computer, or share resources. For communication outside company with customers or abroad staff, company uses Internet by leasing lease line from private company. It can be used to upload files from test server to live server, sending and receiving e-mail, or exchange data.

3.2 Organization Chart

The organization chart shows the chain of command and responsibility from top to bottom of the organization. The structure of the organization is a form of centralization because capability to decision making comes from top level management only.

This report emphasizes on the organization structure Thailand section only because this system supports only Thailand branch. Although headquarters in England can review information but they do not have authorization to make anything with the system.

This figure below presents the organization structure of Imagelntellect control:

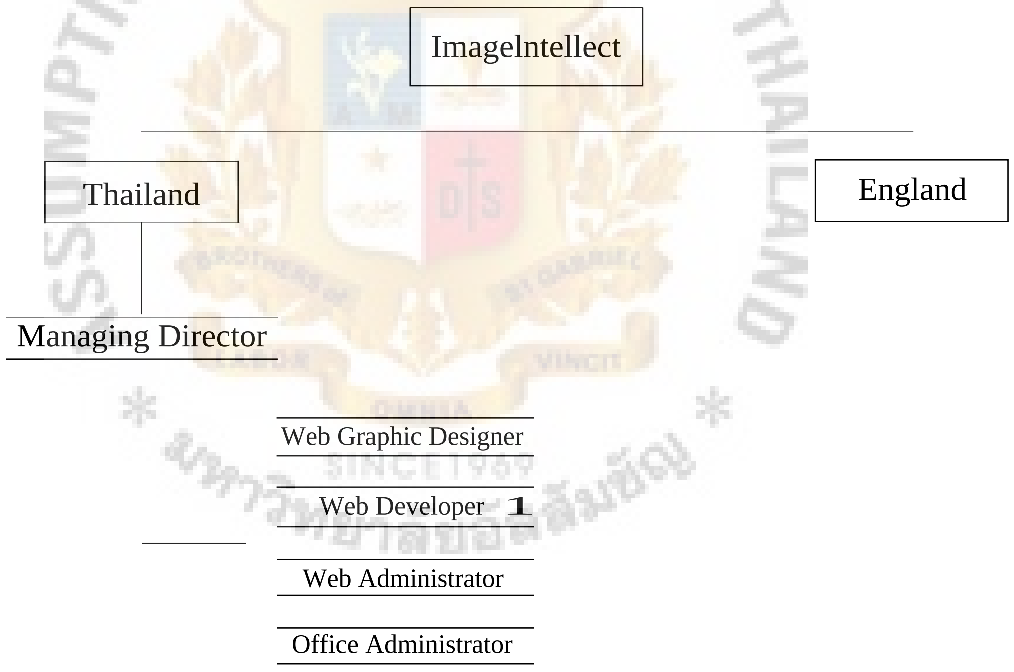


Figure 3.1. Organization Chart of Imagelntellect.

3.3 Working Process

ImageIntellect provides service for creating and developing customer web application project. Customer will send the requirement. Company will make a quotation and send it back. If customer agrees to the quotation, company will make a contract for customer to sign. Within the contract, it presents any condition between customer and company such as budget, job warranty, working time, etc.

In the initial stages of the project, web graphic designs web interface will be sent to customer for approval. When customer is satisfied with web interface, next step is creating and developing website. Customer can access to view progress of website via web browser. Customer can adjust and comment on any detail if they have, until the whole website accomplished.

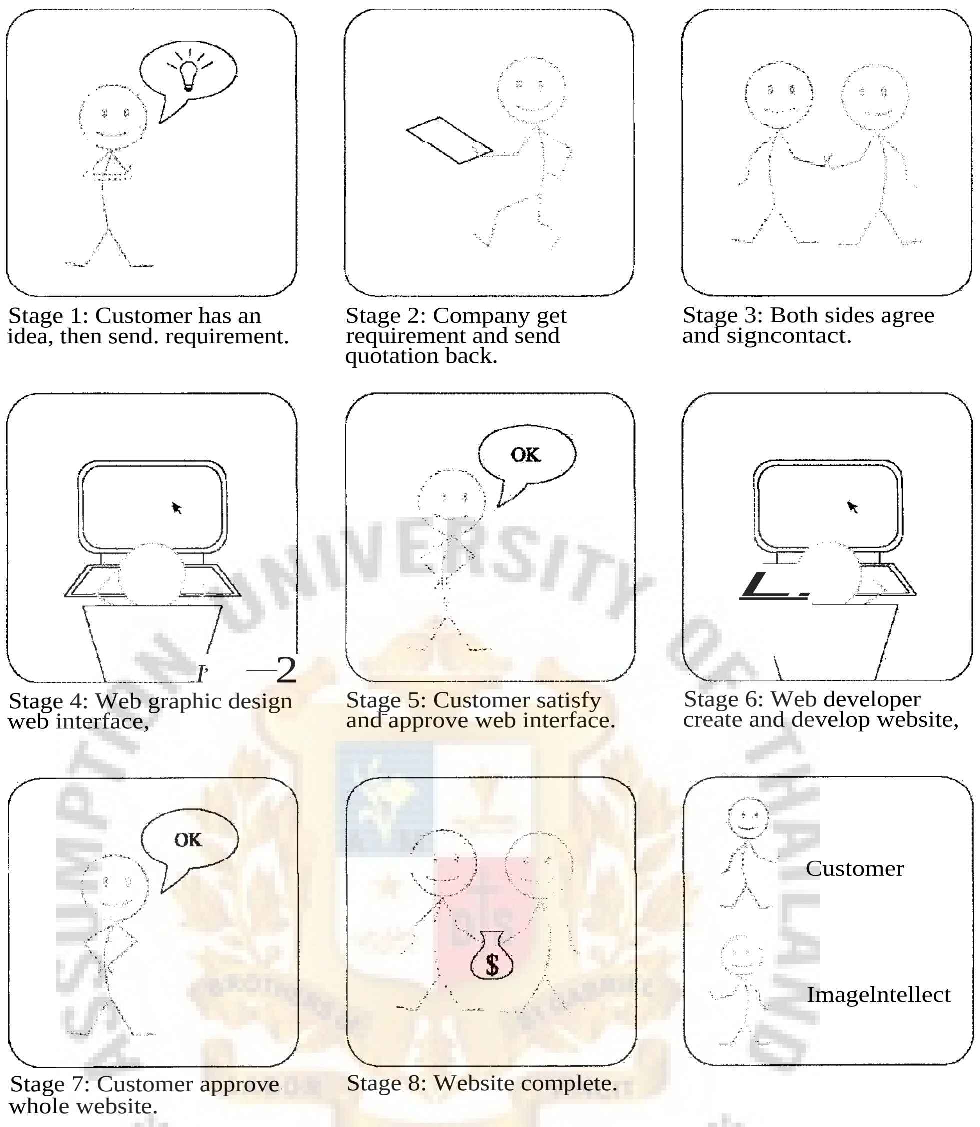

Stage 1: Customer has an idea, then send. requirement.
Stage 2: Company get requirement and send quotation back.
Stage 3: Both sides agree and signcontact.
Stage 4: Web graphic design web interface,
OK
Stage 5: Customer satisfy and approve web interface.
Stage 6: Web developer create and develop website,
OK
Stage 7: Customer approve whole website.
$
Stage 8: Website complete.
Customer
Imagelntellect

Figure 3.2. Story Board of Working Process.

3.4 Current Problem and Area of Improvement

Current problem and area of improvement in the existing system has the following:

Organization Management

(1) Employee or staff must write a daily report that concludes what job is done. It usually takes more time to think of what job they do and waste working time.

(2) Manager and board has problems to track work in progress and job done for each employee because it is in document or paper form.

(3) Organization must buy many office equipments such as paper, pen, etc. for employees to make a report. Moreover, documents are kept in files and binders. It increases cost of company.

Cash Management

(1) Most documents are paper documents kept in file and binders. Cost of office equipment increases.

(2) When manager or auditor, would like to find source of cash inflow or outflow, it is very difficult to retrieve it from cabinet or file keeper.

(3) Company has problem to analyze and review cash flow within organization because all data are kept in paper form. Moreover, it is difficult to see a clear picture for estimation and planning in the future.

3.5 Hardware and Software

Hardware specification is divided into 3 groups. First group is computer that is used for creating development testing of web application. This can be called "Client Computer". Second group is computer that runs or operates as a server. This is called "Test Server". Last group is other hardware or peripheral device such as printer, scanner, facsimile, etc.

For software, there are 2 main groups of software. First is "Operating System" (OS) which divided into 2 parts. Part one is OS that used in running for server side such as "Window2000 Server". Part two is software specification that is used for running on client side such as "WindowXP Professional".

Second software specification is software that is used to develop and fulfill supplement to make application carry out, the example is "MS Interdev", "SQL"

Table 3.1. Existing Hardware.

Server - CPU P IV 2000 MHz - RANI 256 MB - Disk drive 80 GB - Monitor 19 Inch	1	A test and database server
Client - CPU P IV 1200 MHz - RANI 128 MB - Disk drive 40 GB - Monitor 17 Inch	3	Web development PC and Office PC
Printer All-in-One	1	Network printer
Hub 8 ports	1	LAN equipment
ADSL system	1	Internet connection

Table 3.2. Existing Software.

Software	Quantity	Function
Microsoft Windows 2000 Server	1	Network operating system
Microsoft Windows XP Professional	3	Operating system
Microsoft Office XP Professional	4	Office tool
Microsoft Interdev 6.0	3	Web development tool
Microsoft SQL Server 2000	3	Database tool

Table 3.2. Existing Software (Continued).

Adobe Photoshop 7.0	3	Image tool
Norton Antivirus 2003	4	Virus scan tool

IV. SYSTEM DESIGN

4.1 Data Flow Diagram

The proposed system has been designed in order to collect all related information for the system. The information is processed and used as a tool.

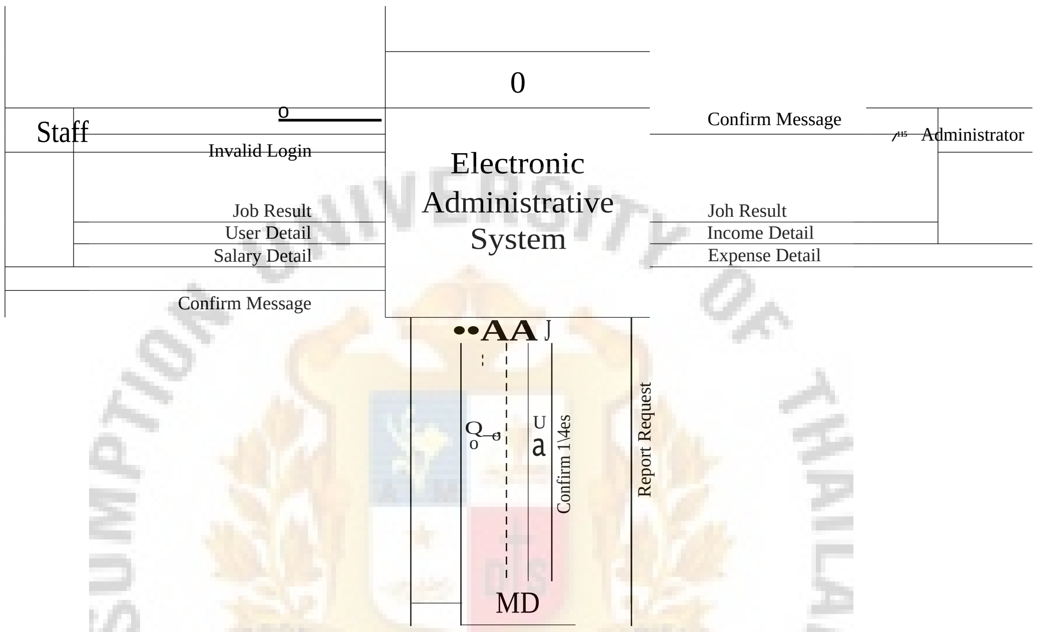


Figure **4.1.** Context Diagram.

From Figure 4.1., it is seen that external entities in the system are staff, administrator, and managing director (MD). All entities are people who work in Imagelntellect Thailand. Information that staff gives to the system is login, user detail, salary detail and job result. For administrator, information that provides is job result and details of cash are income, expense. Lastly, MD can manage job and cash flow. Moreover, MD can request for report for job and cash.

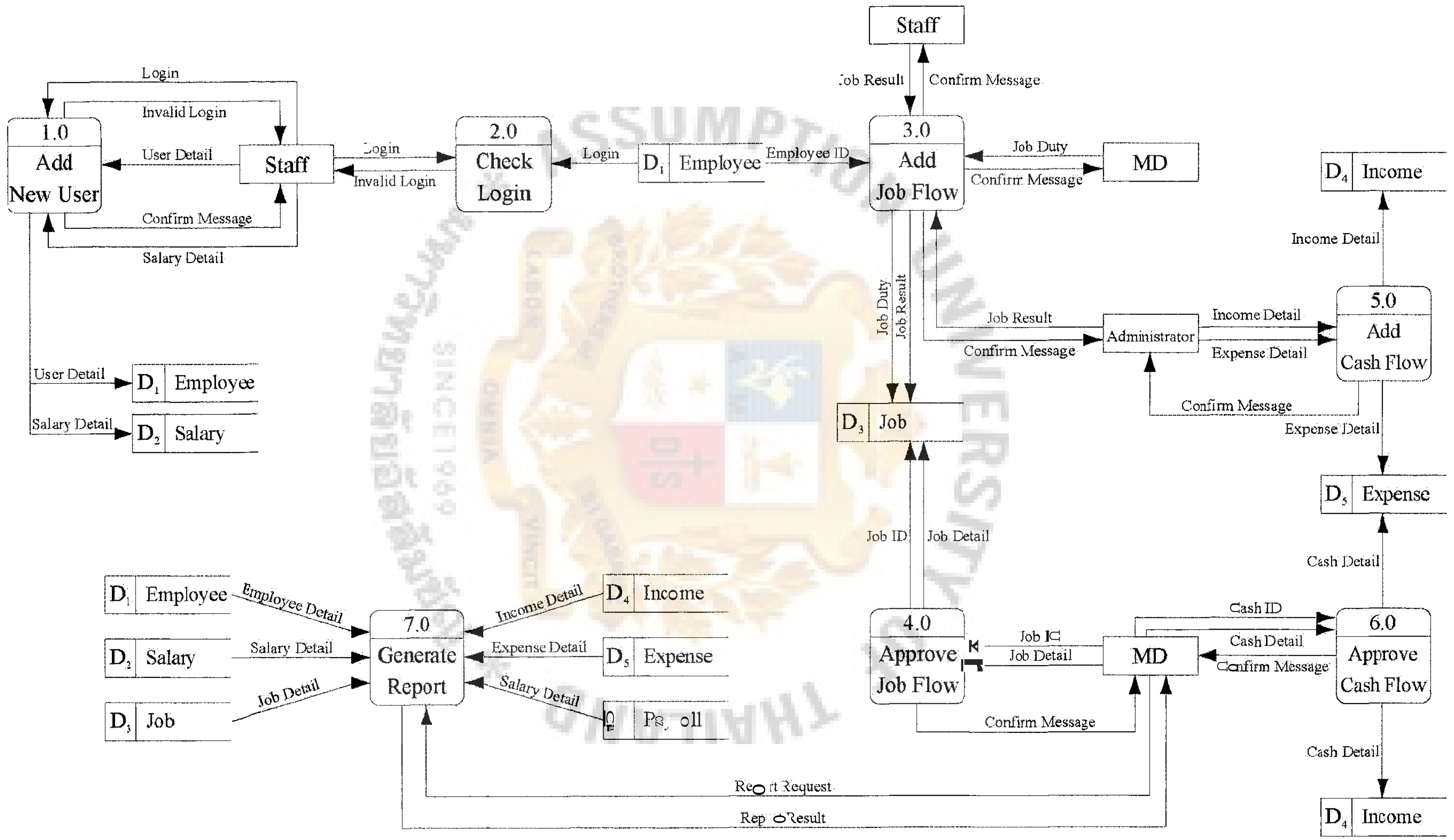


Figure 4.2 DFD Level 0.

Data Flow Diagram level 0 illustrates the relationship along the system. Initial stage starts from adding new user who can use login for access to the system. When they are already accessed, they can only add job result. On the other hand, MD can assign job duty for each employee. Moreover, MD can update, delete, or approve each job. This can make MD verify and track each job effectively.

For cash flow, the person that has the responsibility to insert is administrator. Information that is put is detail about income and expense. MD has duty to edit, delete, or approve each transaction.

Lastly, whenever MD would like to get report for planning, performance appraisal, or controlling, MD can request for report that he wants to get, system will display summary of require report.

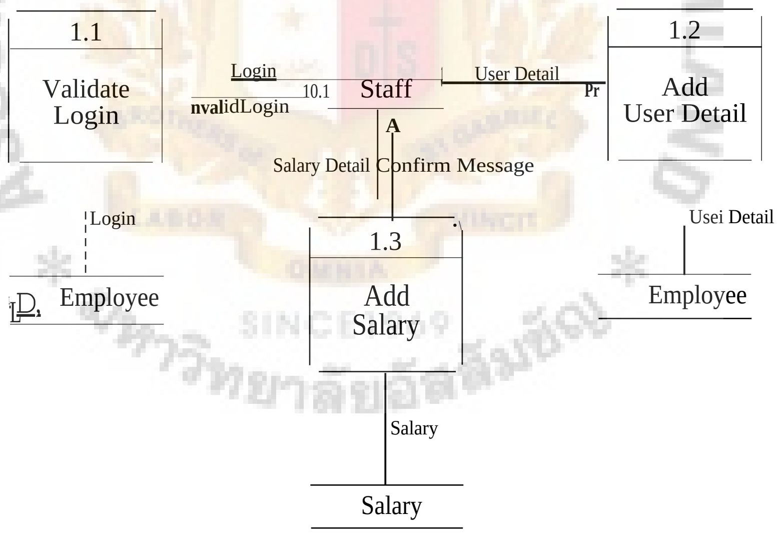


Figure 4.3. DFD Level 1 Process 1.

Data Flow Diagram level 1 of process 1 shows step of adding new user for this system. First step is verification for login. This can validate new login that has already been reserved or not, if it is invalid, staff must choose new login until it is available. After that staff input user and salary detail into database via web application.

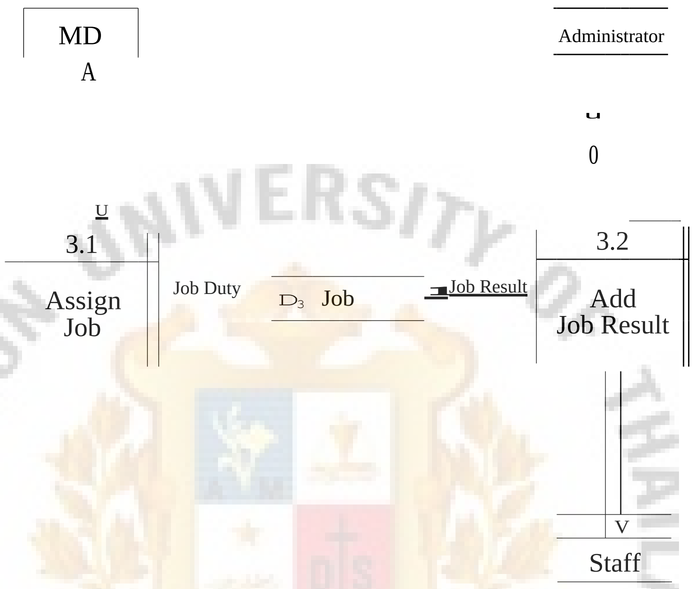


Figure 4.4. DFD Level 1 Process 3.

Data Flow Diagram level 1 of process 3 shows MD can assign job duty for each employee beyond. Both staff and administrator can access to the system for viewing an assigned job. Both of them can add job result when job finished.

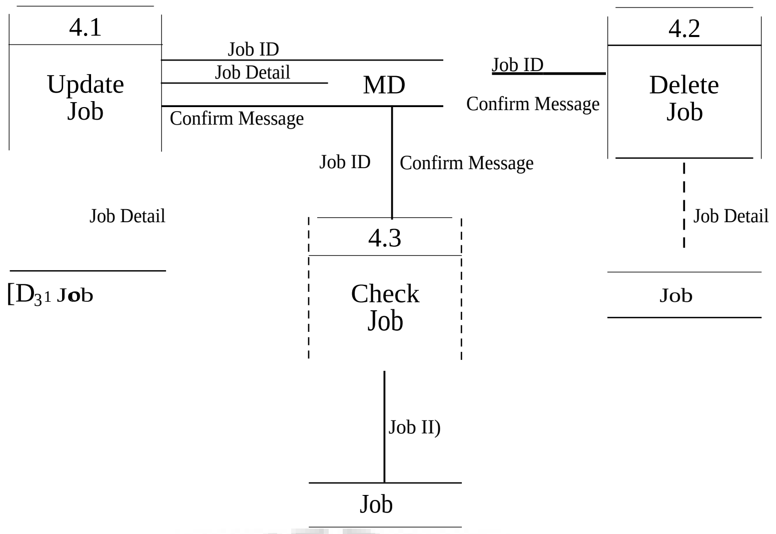


Figure 4.5. DFD Level 1 Process 4.

Data Flow Diagram level 1 of process 4 shows management of job for MD. MD can update job, if there is any mistake or more information for each existing job. If MD would like to delete job from database, it can be done easily by input job id. However, MD can approve job for verification of a completed job.

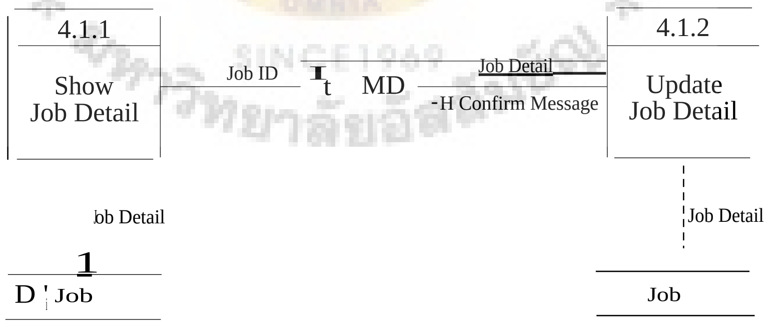


Figure 4.6. DFD Level 2 Process 4.1.

Data Flow Diagram level 1 of process 4.1 presents step for updating job. First, MID put job id and get job detail back from system. After existing job is presented, MD can change job detail and update it via web application.

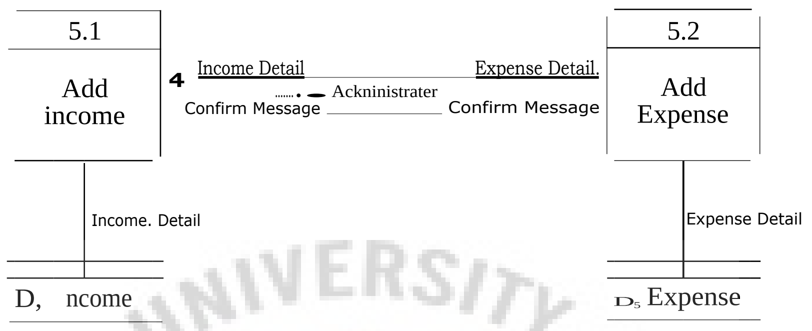


Figure 4.7. DFD Level 1 Process 5.

Data Flow Diagram level 1 of process 5 show insertion of cash information. This process is done by administrator. Administrator has duty for adding cash detail that includes income and expense.

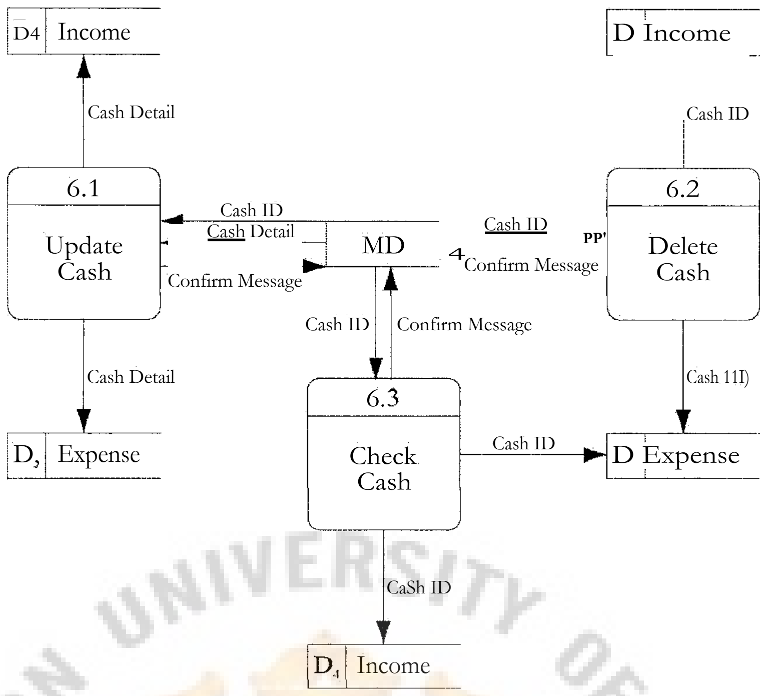


Figure 4.8. DFD Level 1 Process 6.

Data Flow Diagram level 1 of process 6 shows management cash information of MD. MD can update, delete, and approve cash to verify cash flow that moves in and out.

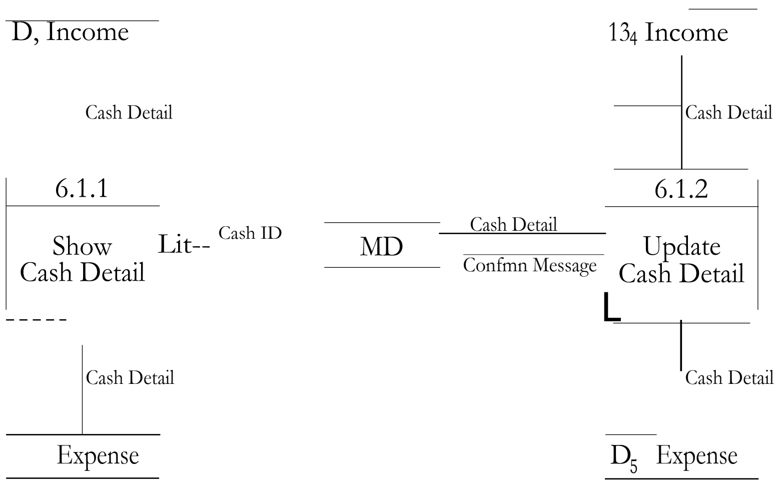


Figure 4.9. DFD Level 2 Process 6.1.

Data Flow Diagram level 1 of process 6.2 shows step of updating cash detail. First, MD put cash ID to retrieve cash detail. After cash detail is displayed, MD can update cash detail and update into database.

4.2 Database Design

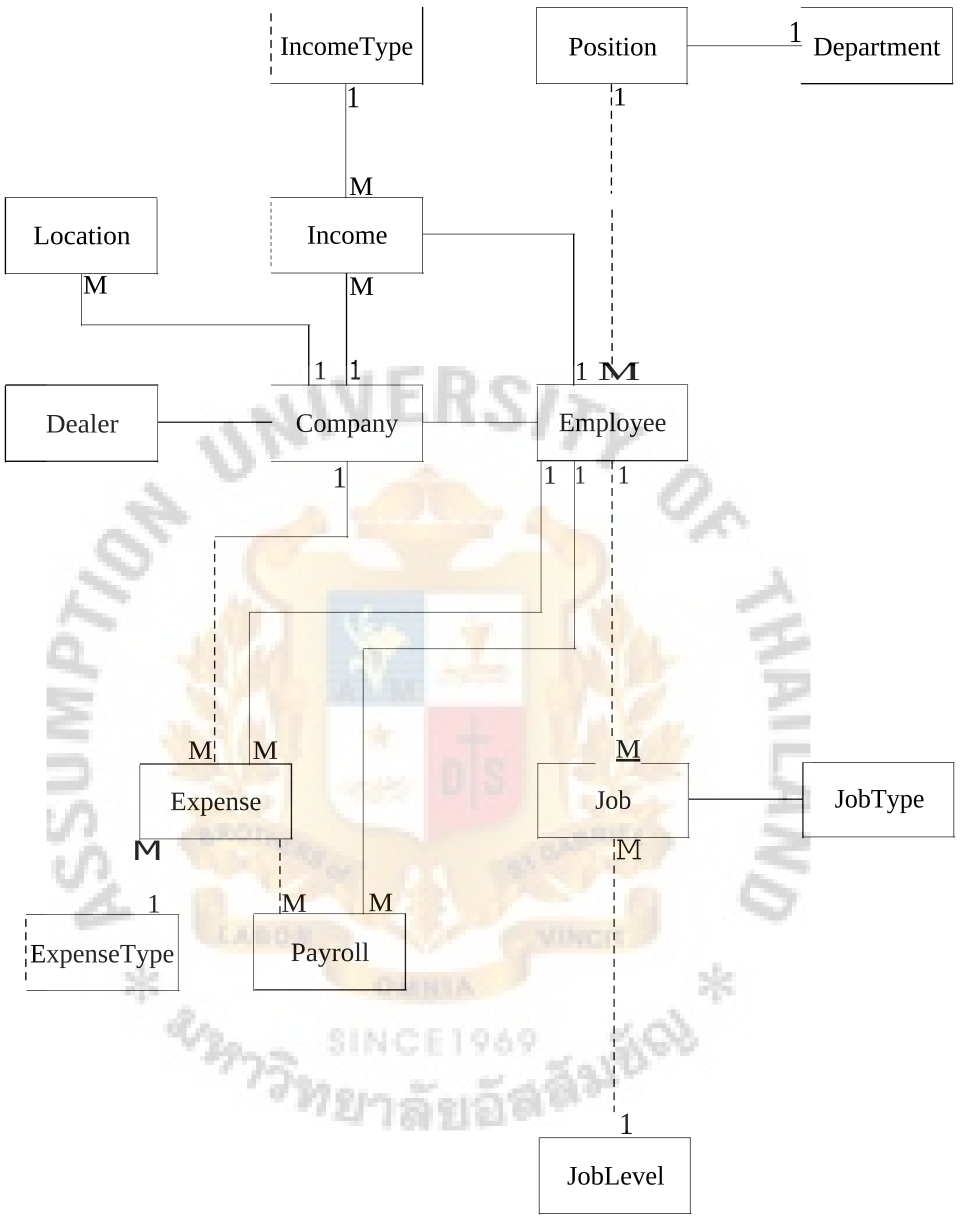


Figure 4.10. E-R Diagram.

Company : It keeps data about company information.

Dealer : This table contains information about company that deals or relates with Imagelntellect Company. It can be supplier, business partner, etc.

Department	: To store master table for department for classifying the position.
Employee	: This table records employee profile.
Expense	: All expenditures are kept by this one.
ExpenseType	: To classify type of expenditure, it uses this one for separation.
Income	: To store income information of company.
IncomeType	: This table used to classify income into groups.
Job	: List of each job or activity done is stored here.
JobLevel	: This table used to classify importance or priority for each job.
JobType	: To separate type of jobs, it uses this one.
Location	: To store records of location or place that company rented.
Payroll	: This table records list of salaries paid each month.
Position	: The duty, responsibility, position and salary are stored.

4.3 Screen Design

User interface of proposed system is designed for user friendly interface. It means developers use concept of nice and clear to create web application. Content provided clear explanation that eases use and understanding. Moreover, this web application uses color theme to get along with the same color of organization.

Please see Appendix C for viewing input, output and report design. Appendix C includes page mock-up and layout grid of website.

4.4 Report Design

Report design is important because it can help Managing Director (MD) to analyze making any decision correctly and precisely. Report for this system has 3 main types stated below:

Job Report

(1) Job level report. This report shows job that is classified by using level of job. This shows manager about job priority from the highest to lowest.

(2) Job type report. It presents job that is determined by using type of job.

(3) Job staff report. Report presents information about which job is done by whom. Besides, manager can view detail of each job such as period of doing time, start and finish date, etc.

Cash Report

(1) Income report. It shows information about income such as source, value of money, type of income, etc.

(2) Expense report. This report is opposite of the above. This present expense information such as source, value of money, type of income, etc.

(3) Payroll report. This shows detail about salary paid each month. Moreover, it calculates social security and personal tax also.

(4) Income statement report. It is the same with income statement on paper. This system can calculate the difference between income and expense in a period of time automatically.

Other Report

(1) Staff report. This report shows personal detail or profile for each staff.

(2) Client report. It presents information for each supplier with contact person.

4.5 Module Design

Designs of system and database will be studied by programs and then be broken into pieces of program modules. The system consists of a hierarchy of modules and they are designed following these principles.

(1) Modularity and Partitioning

Lower-level modules are generally smaller in scope and size compared to higher-level modules and serve to partition processes into separate functions.

(2) Coupling

Modules should have little dependence on other modules in a system.

(3) Cohesion

Modules should carry out a single processing function.

(4) Span of Control

Modules should interact with and manage the functions of a limited number of lower-level modules.

(5) Size

The number of instructions contained in a module should be limited so that module size is generally small.

(6) Shared Use

Functions should not be duplicated in separate modules, but established in a single module that can be invoked by any other module when needed.

Top-down method is used throughout the design process. Starting at the general levels to gain an understanding of the system and gradually moving down to levels of greater detail. During the discussion of input and menu design, a top-down approach is emphasized. The main menu contains several choices. Making one choice produces another menu in which more detailed options are presented to the user. This capability provides users with an easy-to-understand method for using the system and selecting options.

The top-down method is also widely used in software design. Each function the system will perform is first identified and then developed in greater detail. The procedures and processes are developed a step at a time, from general to specific.

V. FINANCIAL STATEMENT

5.1 Breakeven Analysis

Benefits from the purposed system occurred by monthly cost reduction. The tangible benefit is performance improvement and better resources utilization and control. It is expected that the company can save by keeping a number of employees at the current level for the next two years. It reduces cost of office supply such as paper, binder clips, staples, printer ink, etc, besides, it includes cost reduction of working or operating time.

Table 5.1. Project Benefit.

Benefit [illegible]	Amount (Baht)
Office Supply	1,500/month
Saving for Operating Cost	20,000/month
Total Benefit	21,500/month

Cost of the proposed system has initial and fixed cost for developing this system. For initial cost, the development cost is cost of developing the proposed system. Another cost is monthly fixed cost of hosting expansion. This proposed system requires 50 MB more for store data. The maintenance cost is included because the new system needs web developer for debugging and creating more functions. The maintenance tasks do not need require full-time support.

Table 5.2. Project Cost.

Cost	Amount (Baht)
Development Cost for Developer Salary	100,000
Initial Cost	100,000
Hosting Expansion Cost	300/month
Maintenance Cost	10,000/month
Total Fixed Cost	10,300/month

Table 5.3 shows the purpose system can achieve breakeven point on the fifth month. The calculation below shows the way to compute payback period:

Payback period = 8+((11,200-11,100)/11,200)

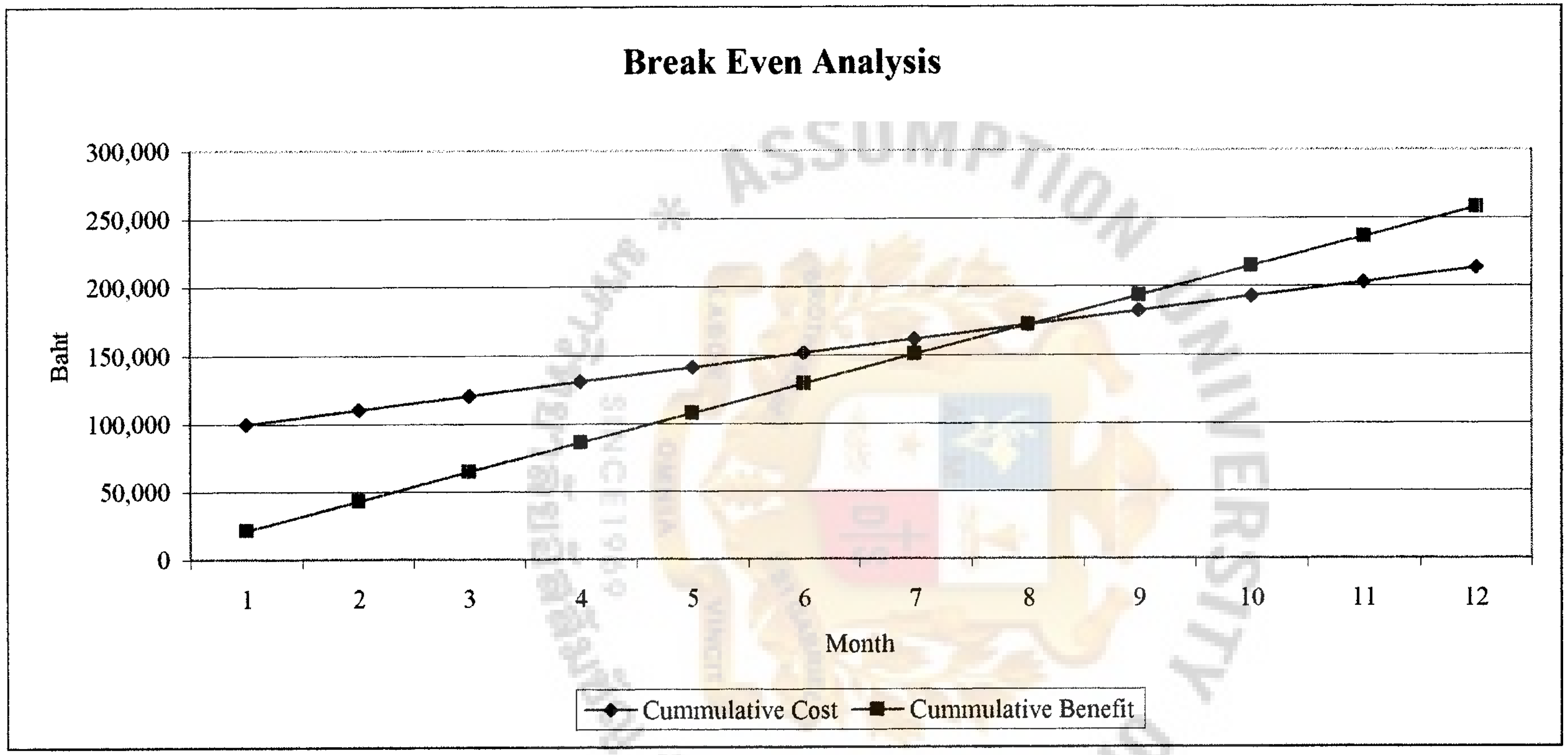
Payback period = 8.008 month

Table 5.2. Project Breakeven Analysis.

1	100,000	100,000	21,500	21,500	-78,500	-78,500
2	10,300	110,300	21,500	43,000	11,200	-67,300
3	10,300	120,600	21,500	64,500	11,200	-56,100
4	10,300	130,900	21,500	86,000	11,200	-44,900
5	10,300	141,200	21,500	107,500	11,200	-33,700
6	10,300	151,500	21,500	129,000	11,200	-22,500
7	10,300	161,800	21,500	150,500	11,200	-11,300
8	10,300	172,100	21,500	172,000	11,200	-100
9	10,300	182,400	21,500	193,500	11,200	11,100
10	10,300	192,700	21,500	215,000	11,200	22,300

Table 5.2. Project Breakeven Analysis (Continued).

11	10,300	203,000	21,500	236,500	11,200	33,500
12	10,300	213,300	21,500	258,000	11,200	44,700



Breakeven Analysis Graph.

VI. SYSTEM IMPLEMENTATION

6.1 System Testing

Philosophy of testing is to find error. The way to test system is putting data into system to determine whether the system will process them correctly. Testing is divided into 3 main strategies:

(1) **Unit testing** means each module is tested alone in an attempt to discover any error in its code. This testing focuses or finds errors in each program module. This enables the tester to detect errors in coding and logic that are contained within that module alone.

(2) **Integration testing** is the process of bringing together all modules that a program comprises for testing process. Errors can occur when each module is combined together for operating individual function. Therefore, this testing can check and validate any error within the same function.

(3) **System testing** is bringing together all the programs that a system comprises for testing purposes. This is a combination of integration testing to test the whole system. The testing of a whole can make system running smoothly.

All of these testing can assure that website will process correctly and provide a system can be available effectively all the time.

6.2 System Security and Control

The following security policies are established and enforced in order to ensure that the system is safe from exposures at appropriate level:

(1) Users are separated into two groups, database administrator and end user. Only database administrator can access the structure of data tables directly. For end users can create, read, and write data via the application software.

(2) Each user has a unique logon name and password.

(3) Each user has a different level for access into the system. The authorization to manage system depends on the access control level.

(4) End users cannot delete master data and protect transactions in order to maintain data integrity. However database administrator can do it.

(5) When users do any activity or transaction, the system will record who and when data is interacting automatically. So, administrator can track who interacts with system.

(6) Data is backed up automatically once a week.

(7) Virus scanner is installed in the server and every workstation. A new virus data definition or engine files are set to be automatically downloaded and updated weekly from the Internet.

(8) Input data is validated by the program and DBMS.

Moreover, company sets up policy, procedure, standard and regulation to control staff. This can be done in short, intermediate and long terms.

6.3 System Conversion

Direct Cutover method is used for converting from the old system to the new one. This converting method is suitable with a small system because it changes the system from old to new immediately. Although it makes high risks to occur any mistake it will not be critical with company.

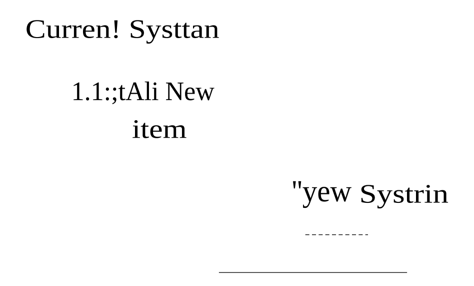


Figure 6.1. Direct Cutover Conversion.

6.4 System Training

Training is a methodology to give education and knowledge about the system. Training can help new system to perform or operate smoothly error free. If users understand how to use the system, they can use system effectively to create benefits for company. Besides, training can prevent misunderstanding in using new system.

Moreover, this system creates a manual guide that teaches or exhibits how to use the new system step by step. Manual guide can help end users learn how to solve problems by themselves.

VII. CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

ImageIntellect Company is a company that operates an in-house web development business. Company has 2 offices located in England and Thailand. Company suffers a lot of problems such as tracking and evaluation job flow that is very difficult, management cash inflow and outflow that are difficult to do.

An electronic administrative system is developed to create effective management within company. This system is deigned to collect all information related to cash and job flow process.

This system helps the manager to track both job result and cash flow. Besides all information is presented in a form of summary report that is easy to understand.

Initial cost to setup the proposed system is 100,000 baht and monthly fixed cost is 10,300 baht. On the other hand, benefit from this system is 21,500 baht per month. The proposed system achieves payback period within 8 months and continuous benefit's growth gradually.

The proposed system has a testing and evaluation before using. Besides, manual and training is implemented to fulfill using new system effectively. Conversion from old to new system uses the idea of direct cutover because it suits small systems.

7.2 Recommendations

ImageIntellect would like to make further features to cover beyond aspect below:

(1) To expand new features by setting up electronic organizer.

This feature can help ImageIntellect Company in collaboration between headquarter and branch. Both sides can manage an online schedule. It can make an effective relationship in job flow for each other. It creates benefit for cooperation simultaneously.

(2) To create customer order feature.

This function can be very useful because it will create customer relationship management. Customer can fill in the requirement form that provides general function for web application and includes special requirements.

(3) To create grading job result function.

Managing Director can give a mark for each finished job. Moreover, this function can summarize all grading marks and represent it in a form of bar chart for ease to understand and use for planning in the future.

APPENDIX A

DATA DICTIONARY

Cash Detail	= Income Detail + Expense Detail
Cash ID	= Unique number for identify cash number
Confirm Message	= Message for confirmation
Employee Detail	= Detail of employee
Employee ID	= Unique number for identify employee number
Expense Detail	= Detail of expense
Income Detail	= Detail of income
Job Detail	= Detail of job
Job Duty	= Duty of each job that is assigned by MD
Job ID	= Unique number for identify job number
Job Result	= Result of job that is done by each staff
Login	= Login and password for access to system
Report Result	= Result of required report
Report Request	= Request information for required report
Salary Detail	= Detail of employee salary
User Detail	= Detail of user

APPENDIX B

DATABASE

Table B.1. Company.

Field Name	Data Type	Description
CompanyID	Int	Primary Key Not null
Name	Varchar(50)	
RegistrationNumber	Varchar(50)	
Capital	Int	Default = 0

Table 112. Dealer.

DealerliD	Int	Primary Key Not null
Firstname	Varchar(50)	
Lastname	Varchar(50)	
AddressNumber	Varchar(25)	
Street	Varchar(50)	
SubDistrict	Varchar(50)	
District	Varchar(50)	
Province	Varchar(50)	
Zip	Varchar(10)	
Telephone	Varchar(25)	
Mobile	Varchar(25)	
Fax	Varchar(25)	
Email	Varchar(100)	
Business	Varchar(50)	
Contact 1	Varchar(100)	
Contact2	Varchar(100)	
CompanylD	Int	Foreign Key Not null

Table B.3. Department.

Field Name	Data Type	Description
DepartmentID	Int	Primary Key Not null
Name	Varchar(50)	

Table B.4. Employee.

i-,1,-.		
EmployeelD	Int	Primary Key Not null
Firstname	Varchar(50)	
Lastname	Varchar(50)	
Gender	Bit	0 = Male 1 = Female Default = 0
BirthDate	DateTime	
Age	Int	Default = 0
AddressNumber	Varchar(25)	
Street	Varchar(50)	
SubDistrict	Varchar(50)	
District	Varchar(50)	
Province	Varchar(50)	
Zip	Varchar(10)	
Telephone	Varchar(25)	
Mobile	Varchar(25)	
Fax	Varchar(25)	
Email	Varchar(100)	
EmergencyContact	Varchar(100)	
Status	Bit	0 = Full-time 1 = Part-time Default = 0
Username	Varchar(25)	
Password	Varchar(25)	
TaxNumber	Varchar(25)	

Table B.4. Employee (Continued).

Field Name	Data Type	Description
S SNumber	Varchar(25)	
StartDate	DateTime	
FinishDate	DateTime	
CompanyID	Int	Foreign Key Not null
PositionID	Int	Foreign Key Not null

Table B.S. Expense.

ExpenselD	Int	Primary Key Not null
Total	Int	Default = 0
HaveTax	Bit	0 = Not have 1 = Have Default = 0
Tax	Int	Default = Total*0.07
Payee	Varchar(50)	
PayeeReference	Varchar(50)	
Cash	Bit	0 = No 1 = Yes Default = 0
Bank	Bit	0 = No 1 = Yes Default = 0
Cheque	Bit	0 = No 1 =Yes Default = 0
ChequeNumber	Varchar(25)	
AddedBy	Int	Foreign Key Not null
AddedDate	DateTime	getDate()
ApproveDate	DateTime	

Table B.5. Expense (Continued).

ExpenseTypeID	Int	Foreign Key Not null
CompanylD	Int	Foreign Key Not null

Table B.6. ExpenseType.

ExpenseTypelD	Int	Primary Key Not null
Name	Varchar(50)	

Table B.7. Income.

IncomeID	Int	Primary Key Not null
Total	Int	Default = 0
HaveTax	Bit	0 = Not have 1 = Have Default = 0
Tax	Int	Default = Total*0.07
Payer	Varchar(50)	
PayerReference	Varchar(50)	
Cash	Bit	0 = No 1 = Yes Default =
Bank	Bit	0 = No 1 = Yes Default =
Cheque	Bit	0 = No 1 = Yes Default = 0

Table B.7. Income (Continued).

ChequeNumber	Varchar(25)	
AddedBy	Int	Foreign Key Not null
AddedDate	DateTime	getDate()
ApproveDate	DateTime	
IncomeTypelD	Int	Foreign Key Not null
CompanyLD	Int	Foreign Key Not null

Table B.8. IncomeType.

IncomeTypelD	Int	Primary Key Not null
Name	Varchar(50)	

Table B.9. Job.

JobID	Int	Primary Key Not null
Title	Varchar(50)	
Description	Text	
ResultFlag	Bit	0 = Not finish 1 = Finish Default = 0
Comment	Text	
AddedDate	DateTime	getDate()
StartDate	DateTime	
DueDate	DateTime	
ApproveDate	DateTime	
JobLevellD	Int	Foreign Key

Table B.9. Job (Continued).

JobLevelID	Int	Foreign Key Not null
JobTypeID	Int	Foreign Key Not null
EmployeelD	Int	Foreign Key Not null

Table B.10. JobLevel.

JobLevelID	Int	Primary Key Not null
Name	Varchar(50)	
Description	Text	

Table B.11. JobType.

JobTypeID	Int	Primary Key Not null
Name	Varchar(50)	
Description	Text	

Table B.12. Location.

Location1D	Int	Primary Key Not null
AddressNumber	Varchar(25)	
Street	Varchar(50)	
SubDistrict	Varchar(50)	
District	Varchar(50)	

Table B.12. Location (Continued).

Field Name	Data Type	Description
Province	Varchar(50)	
Zip	Varchar(10)	
Telephone	Varchar(25)	
Mobile	Varchar(25)	
Fax	Varchar(25)	
CompanylD	Int	Foreign Key Not null

Table B.13. Payroll.

[illegible]	[illegible]	[illegible]
PayrollID	Int	Primary Key Not null
Month	Int	
Year	Int	
Total	Int	Default = 0
TaxDeduction	Int	Default = 0
S SDeduction	Int	Default = 0
Bonus	Int	Default = 0
Commission	Int	Default = 0
OT	Int	Default = 0
Cash	Bit	0 = No 1 = Yes Default = 0
Bank	Bit	0 = No 1 = Yes Default = 0
Cheque	Bit	0 = No 1 = Yes Default = 0
ChequeNumb er	Varchar(25)	
ExpenselD	Int	Foreign Key Not null

Table B.13. Payroll (Continued).

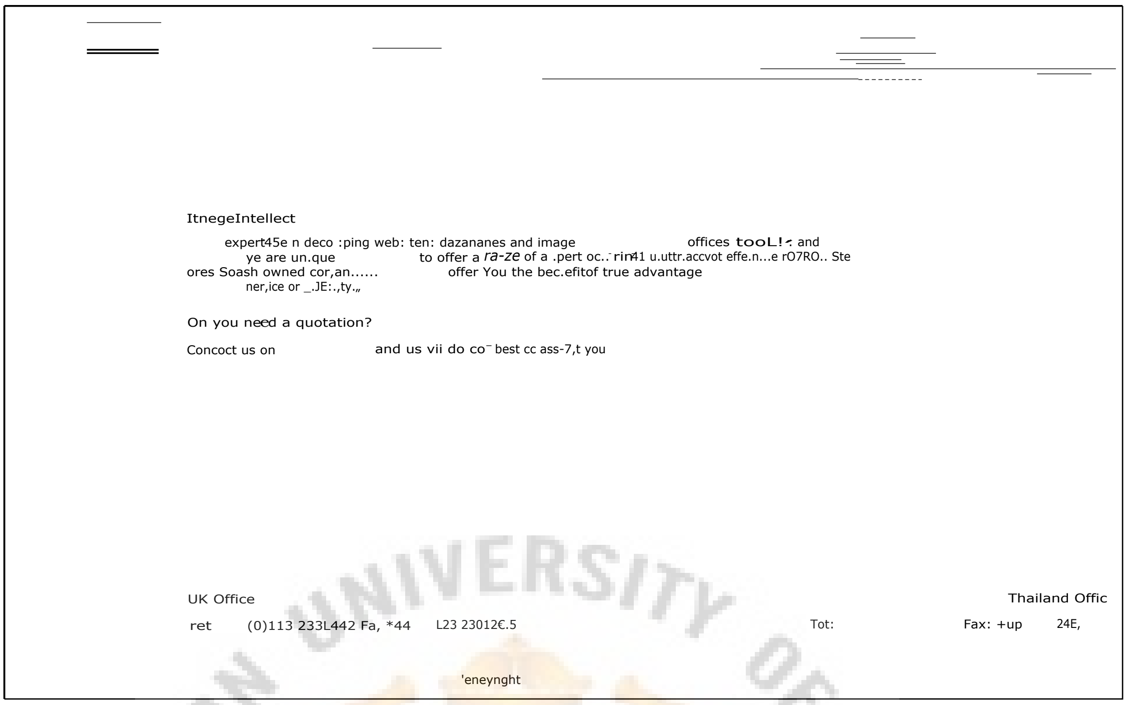
EmployeeID	int	Primary Key Not null

Table B.14. Position.

[illegible]	[illegible]	[illegible]
PositionlD	Int	Primary Key Not null
Name	Varchar(50)	
Duty	Text	
SalaryAmount	Int	Default = 0
DepartmentlD	Int	Foreign Key Not null

APPENDIX C

WEBSITE INTERFACE



ItnegeIntellect

expert45e n deco :ping web: ten: dazananes and image offices tooL!˂ and
ye are un.que to offer a *ra-ze* of a .pert oc..˙rin41 u.uttr.accvot effe.n...e rO7RO.. Ste
ores Soash owned cor,an...... offer You the bec.efitof true advantage
ner,ice or _.JE:.,ty.„

On you need a quotation?

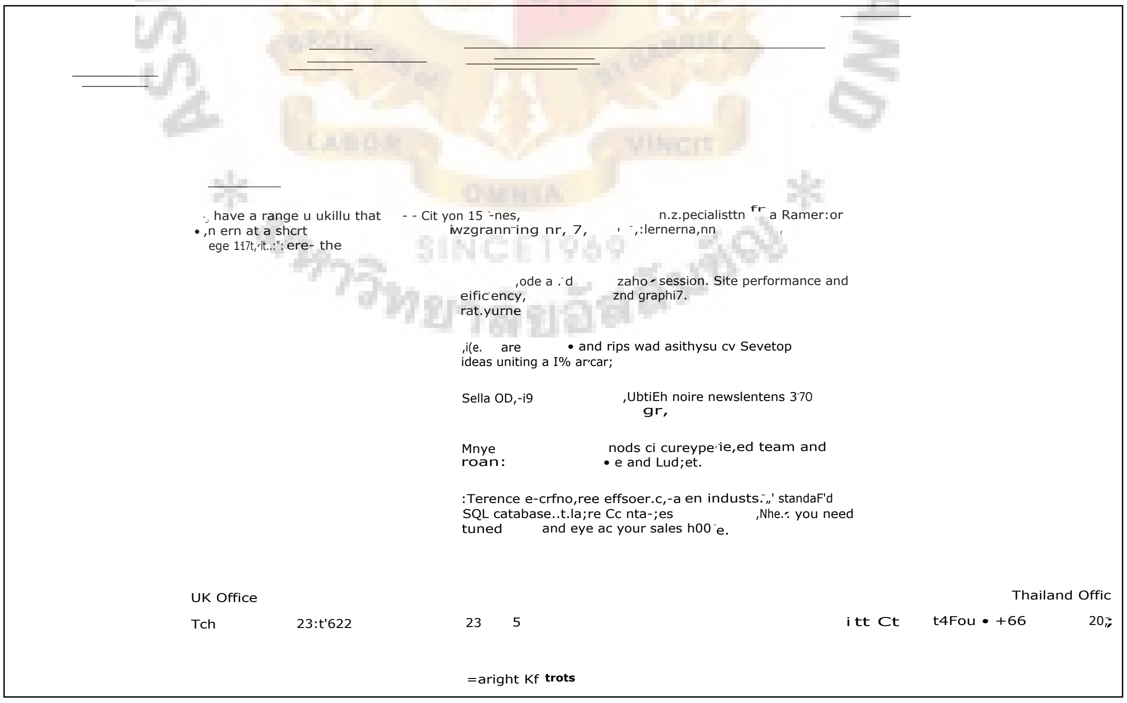
Concoct us on and us vii do co⁻ best cc ass-7,t you

UK Office Thailand Offic

ret (0)113 233L442 Fa, *44 L23 23012C.5 Tot: Fax: +up 24E,

'eneynght

Figure C.1. Default Page of Imageintellect.



·̜ have a range u ukillu that - - Cit yon 15 ⁻nes, n.z.pecialisttn ᶠʳ a Ramer:or
• ,n ern at a shcrt iwzgrann⁻ing nr, 7, ' ˉ,:lernerna,nn ,
ege 1‡7t,ʼit..:': ere- the

,ode a .˙d zaho˂ session. Site performance and
eific ency, znd graphi7.
rat.yurne

,i(e. are • and rips wad asithysu cv Sevetop
ideas uniting a I% ar·car;

Sella OD,-i9 ,UbtiEh noire newslentens 3˙70
gr,

Mnye nods ci cureype˙ie,ed team and
roan: • e and Lud;et.

:Terence e-crfno,ree effsoer.c,-a en industs.˙„' standaF'd
SQL catabase..t.la;re Cc nta-;es ,Nhe.˂ you need
tuned and eye ac your sales h00˙e.

UK Office Thailand Offic

Tch 23:t'622 23 5 i tt Ct t4Fou • +66 20;

=aright Kf trots

Figure C.2. Service Information.

We deliver wets and Er4Sedleit as See fiiizrsi a hii&ifi,is zilled team ct sind understand butt to make c... critical req: eamersts,

frr.ar d ecu busrtess sAtLut tittair•di

- Drarnetica'r retuCe
- ,5-Kase the rich ryckiediin daiveicsrnent-
- Gain access tc ,tcniciistacc'softicarei engiiticierng prejects.with it-ithetwisa.cannct Es Sc techricai ar staff iirnitaticns.
- Taie a fresh arc new. serspest we sea stalled p-sjesit

51K Office

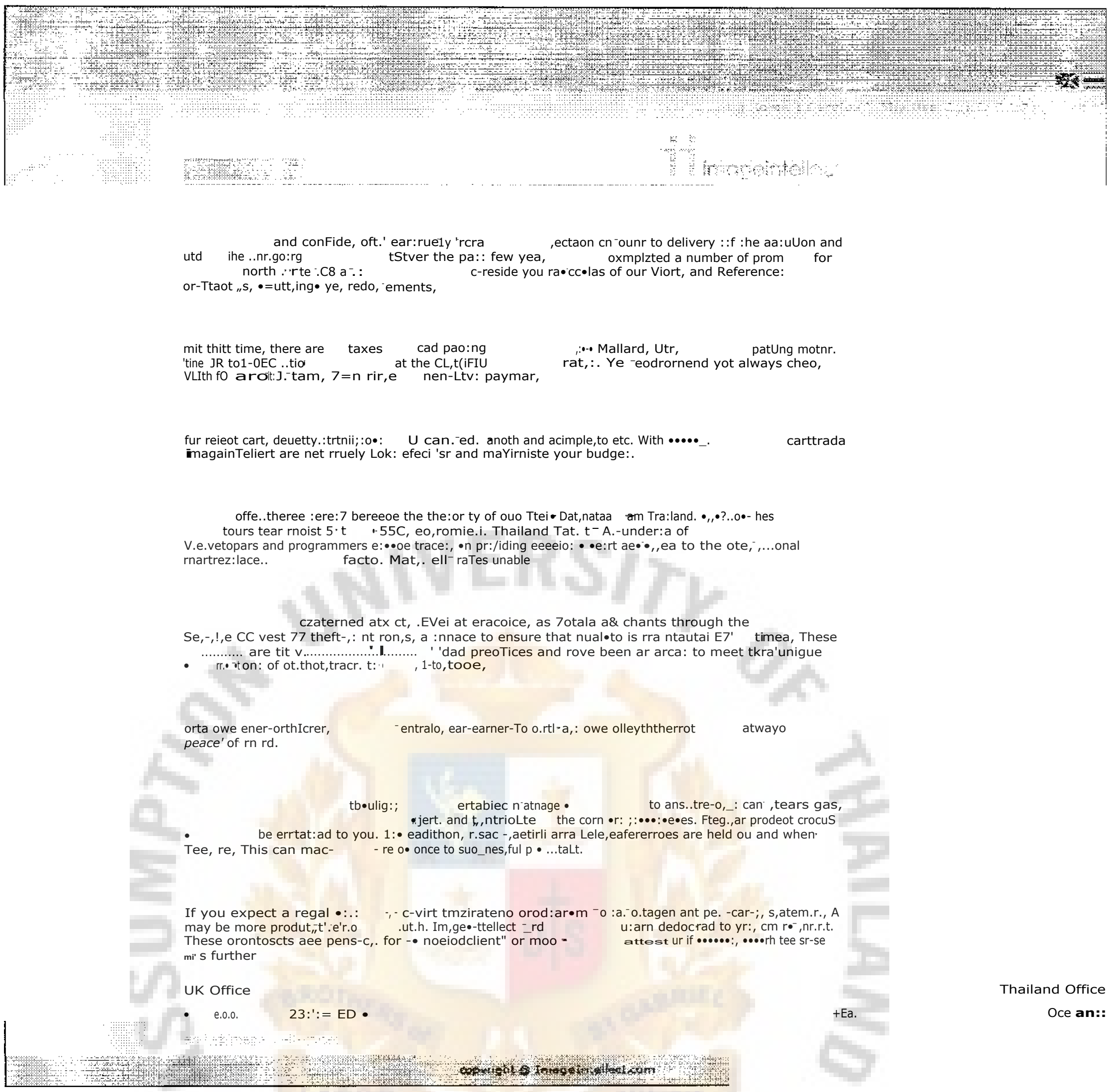
Tel: iii3iit:5; 235C622 Fa:: 444iii1._ -3Ct2n5

Thailand Office

TES: ttii 2032 Fei:7 +66 (0)2 246 2033

copyright @ T.tarri

Figure C.3. Why Use Us Information.



Imagaintellect

and conFide, oft.' ear:ruely 'rcra ,ectaon cn-ounr to delivery ::f :he aa:uUon and utd ihe ..nr.go:rg tStver the pa:: few yea, oxmplzted a number of prom for north .·r te .C8 a-. : c-reside you ra•cc•las of our Viort, and Reference: or-Ttaot „s, •=utt,ing• ye, redo, ements,

mit thitt time, there are taxes cad pao:ng ,:•• Mallard, Utr, patUng motnr. 'tine JR to1-0EC ..tio at the CL,t(iFIU rat,:. Ye -eodrornend yot always cheo, VLIth f0 aro:J.-tam, 7=n rir,e nen-Ltv: paymar,

fur reieot cart, deuetty.:trtnii;:o•: U can.-ed. anoth and acimple,to etc. With •••••_. carttrada imagainTeliert are net rruely Lok: efeci 'sr and maYirniste your budge:.

offe..theree :ere:7 bereeoe the the:or ty of ouo Ttei• Dat,nataa am Tra:land. •,,•?..o•- hes tours tear rnoist 5·t •55C, eo,romie.i. Thailand Tat. t- A.-under:a of V.e.vetopars and programmers e:••oe trace:, •n pr:/iding eeeeio: • •e:rt ae••,,ea to the ote,-,...onal rnartrez:lace.. facto. Mat,. ell- raTes unable

czaterned atx ct, .EVei at eracoice, as 7otala a& chants through the Se,-,!,e CC vest 77 theft-,: nt ron,s, a :nnace to ensure that nual•to is rra ntautai E7' timea, These .... are tit v...................I........ ' 'dad preoTices and rove been ar arca: to meet tkra'unigue • m.•ton: of ot.thot,tracr. t: , 1-to,tooe,

orta owe ener-orthIcrer, -entralo, ear-earner-To o.rtl-a,: owe olleyththerrot atwayo *peace'* of rn rd.

tb•ulig:; ertabiec n-atnage • to ans..tre-o,_: can ,tears gas, •jert. and t,,ntrioLte the corn •r: ;:•••:•e•es. Fteg.,ar prodeot crocuS • be errtat:ad to you. 1:• eadithon, r.sac -,aetirli arra Lele,eafererroes are held ou and when Tee, re, This can mac- - re o• once to suo_nes,ful p • ...taLt.

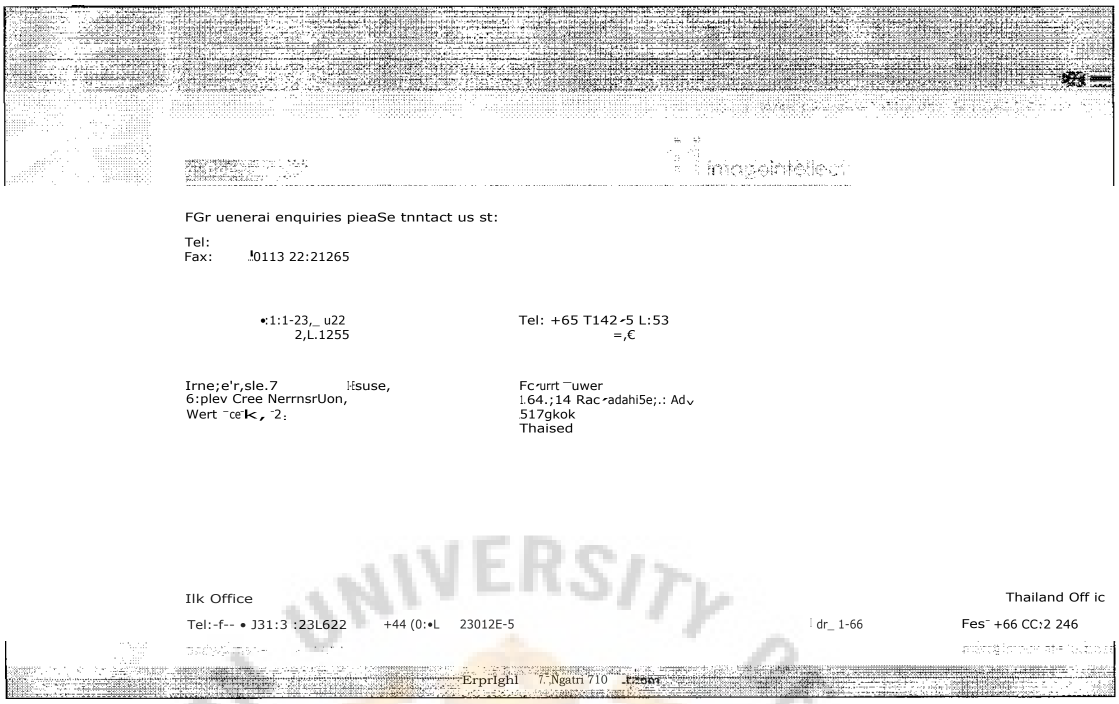
If you expect a regal •:.: -,- c-virt tmzirateno orod:ar•m -o :a.-o.tagen ant pe. -car-;, s,atem.r., A may be more produt„t'.e'r.o .ut.h. Im,ge•-ttellect -_rd u:arn dedocrad to yr:, cm r•-,nr.r.t. These orontoscts aee pens-c,. for -• noeiodclient" or moo - attest ur if ••••••:, ••••rh tee sr-se mi's further

UK Office Thailand Office

• e.o.o. 23:':= ED • +Ea. Oce an::

copyright © Imagaintellect.com

Figure C.4. FAQ Information.



imageintellect

FGr uenerai enquiries pieaSe tnntact us st:

Tel:
Fax: !0113 22:21265

•:1:1-23,_ u22 Tel: +65 T142-5 L:53
2,L.1255 =,€

Irne;e'r,sle.7 Hsuse, Fc-urrt ¯uwer
6:plev Cree NerrnsrUon, 1.64.;14 Rac-adahi5e;.: Adv
Wert ¯ce¯k, ¯2: 517gkok
Thaised

Ilk Office Thailand Off ic

Tel:-f-- • J31:3 :23L622 +44 (0:•L 23012E-5 dr_ 1-66 Fes¯ +66 CC:2 246

ErprIghl 7. Ngatri 710 .t.com

Figure C.5. Contact Us Information.

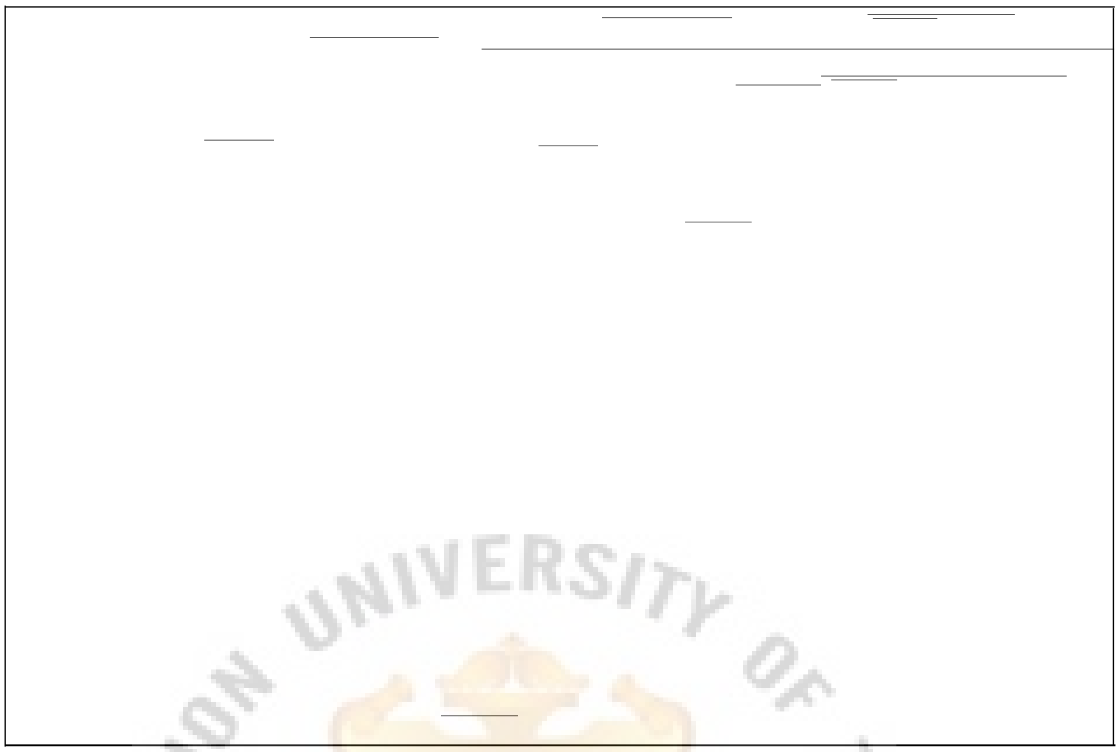


Figure C.6. Add Employee.

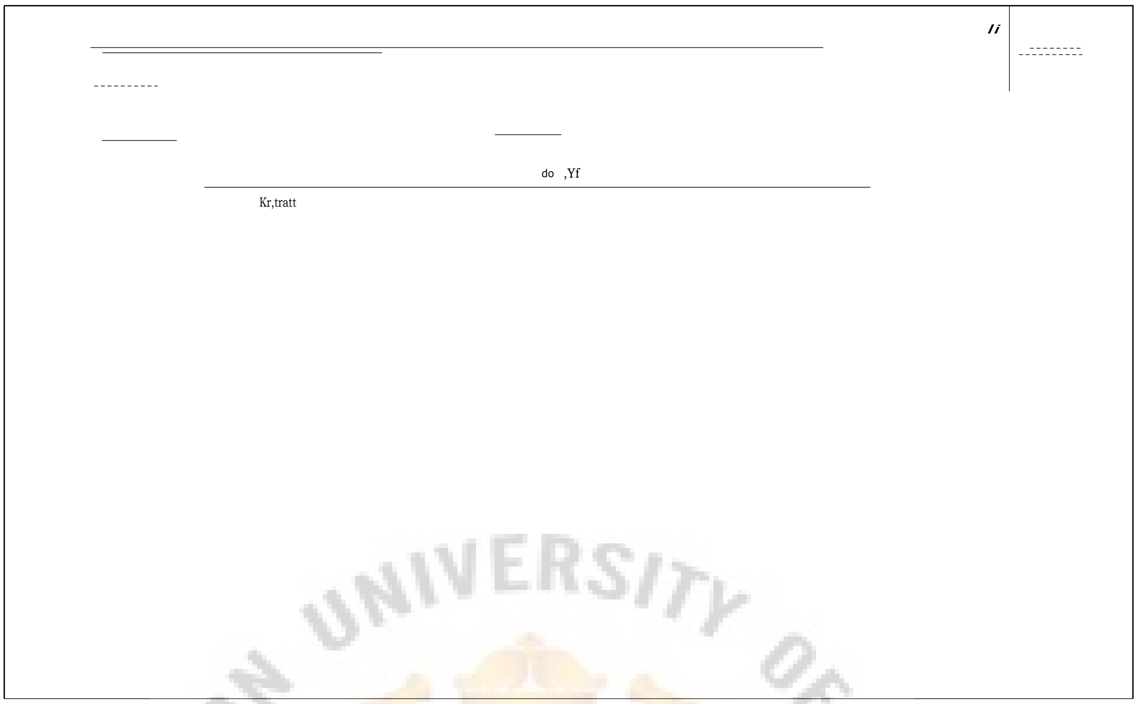


Figure C.7. Employee List.

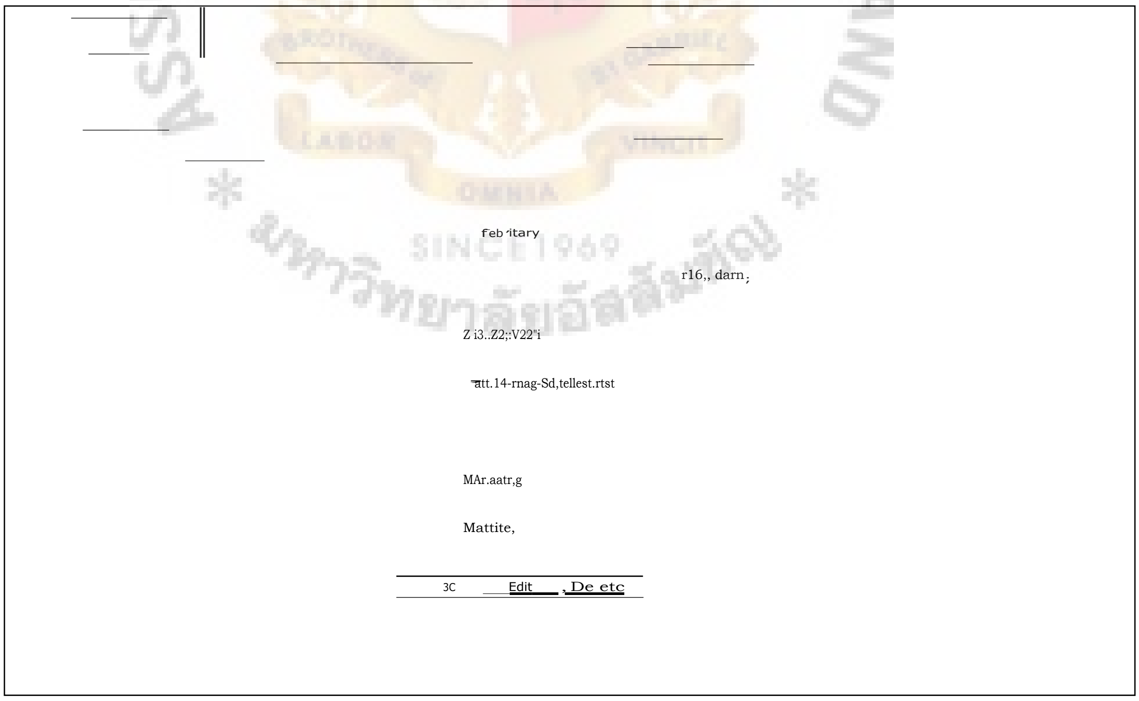


Figure C.S. Employee Detail.

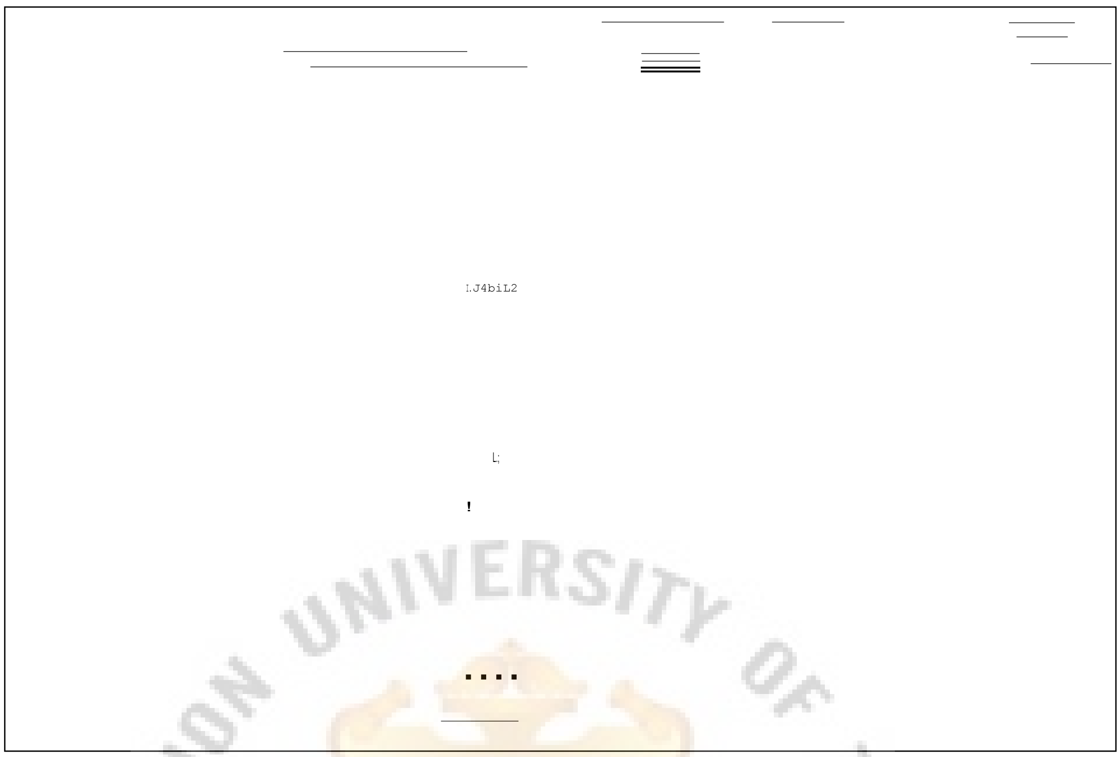


Figure C.9. Employee Edit.

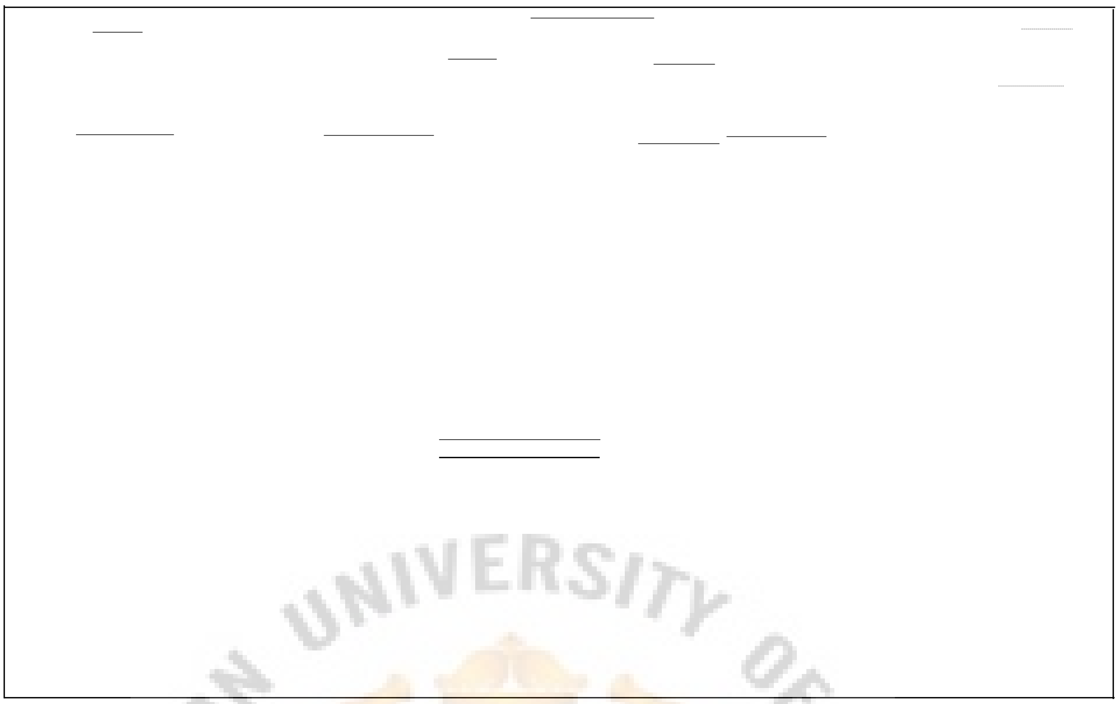
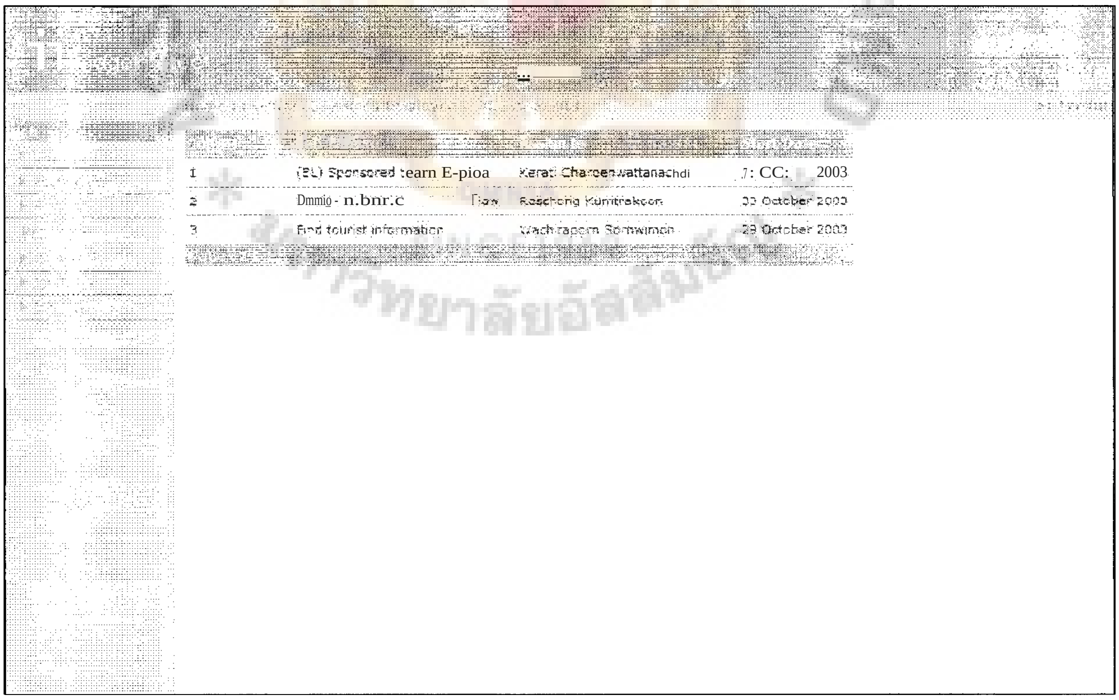
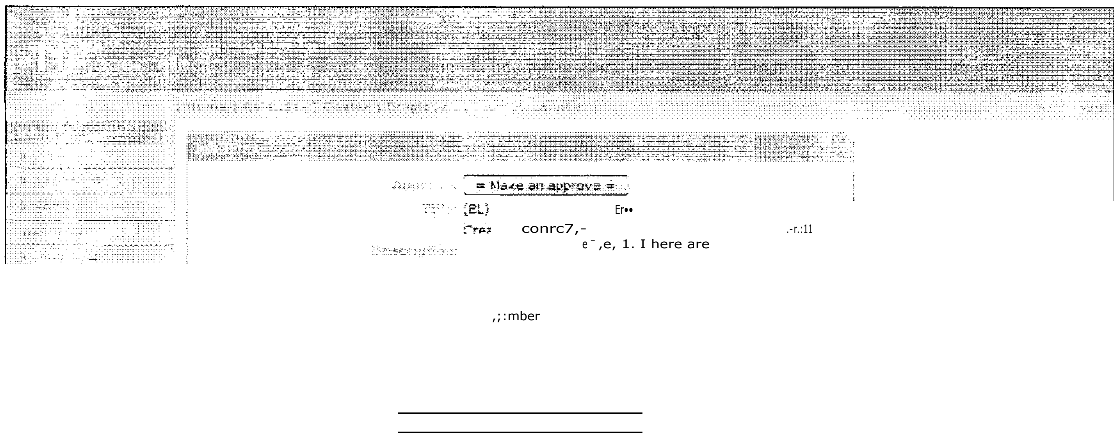


Figure C.10. Assign Job.



1	(EL) Sponsored team E-pioa	Kerati Charoenwattanachdi	7: CC: 2003
2	Dmmio - n.bnr.c Flow	Rosshong Kumitrakoon	22 October 2003
3	Find tourist information	Wachiraporn Somwimon	29 October 2003

Figure C.11. Job List.



= Make an approve =

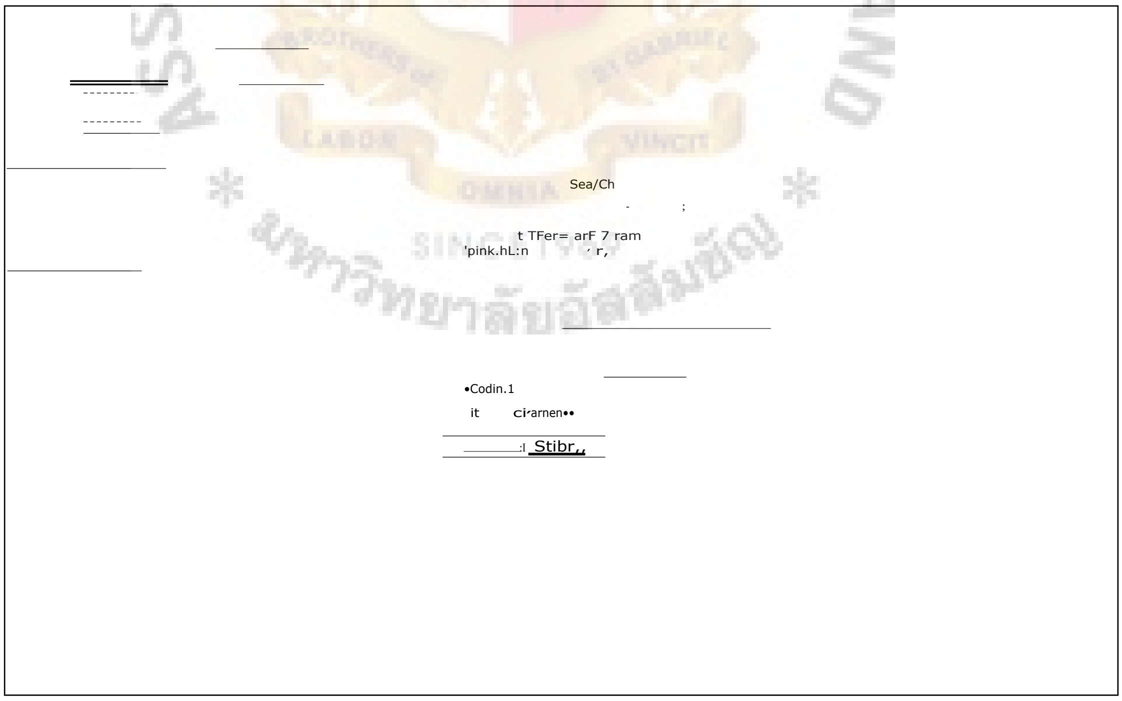
(BL) Er••

Crea conrc7,- -r.:11

e ,e, 1. I here are

,;:mber

Figure C.12. Add Job Approve.



Sea/Ch

t TFer= arF 7 ram

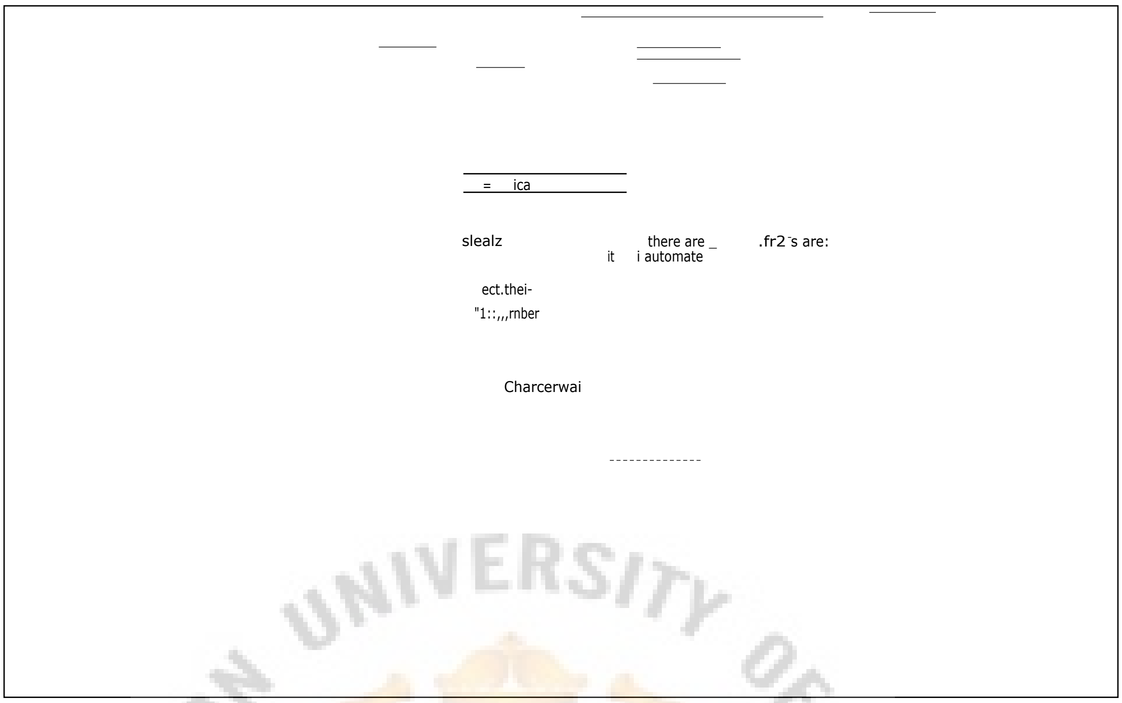
'pink.hL:n r,

•Codin.1

it ci-arnen••

:I Stibr,,

Figure C.13. Job Edit.


= ica

slealz there are _ .fr2 s are:
it i automate

ect.thei-
"1::,,,rnber

Charcerwai

Figure C.14. Add Job Result.

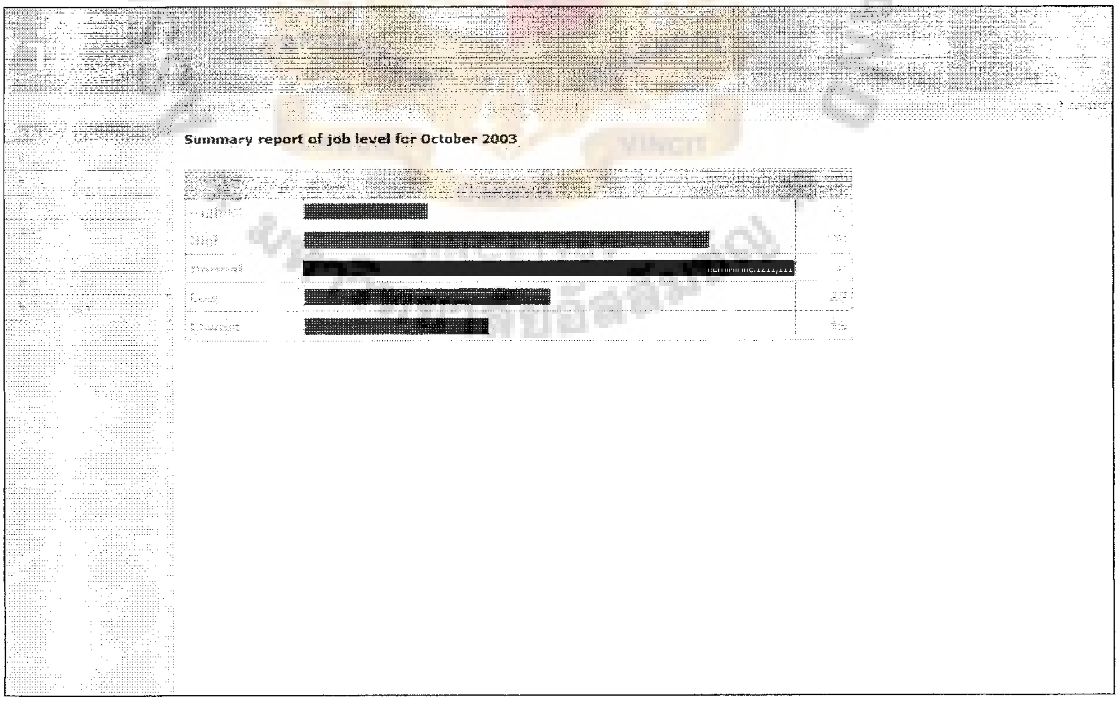


Figure C.15. Job Level Bar Chart.

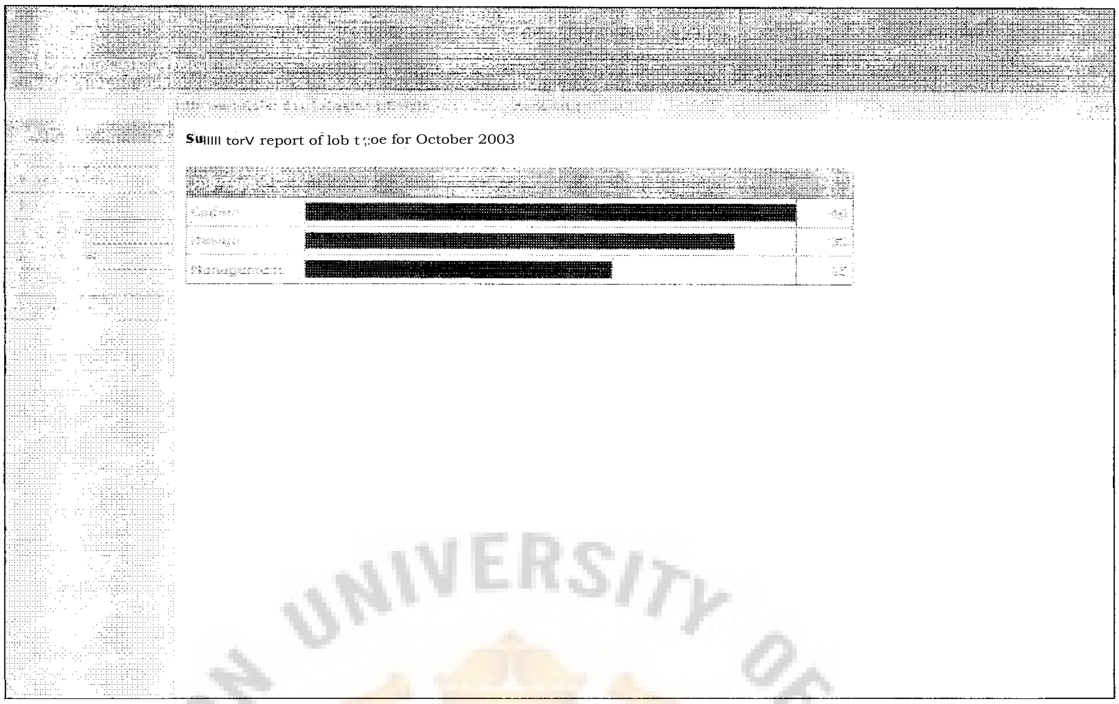

Suııııı torV report of lob t ;:oe for October 2003

Figure C.16. Job Type Bar Chart.

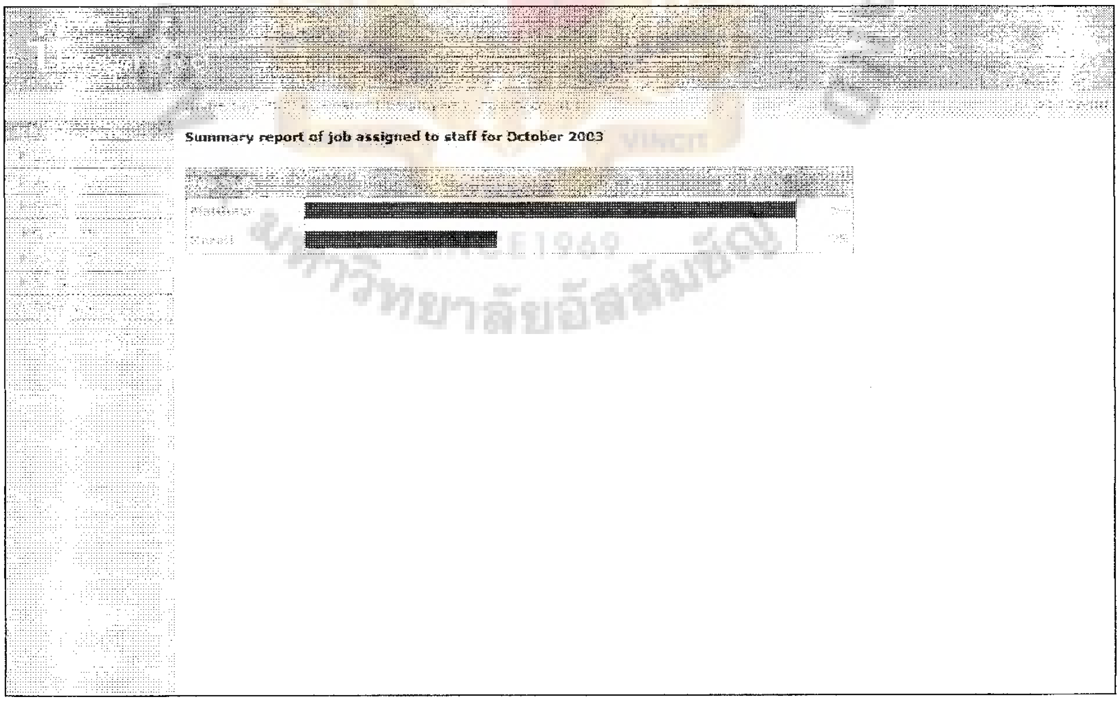
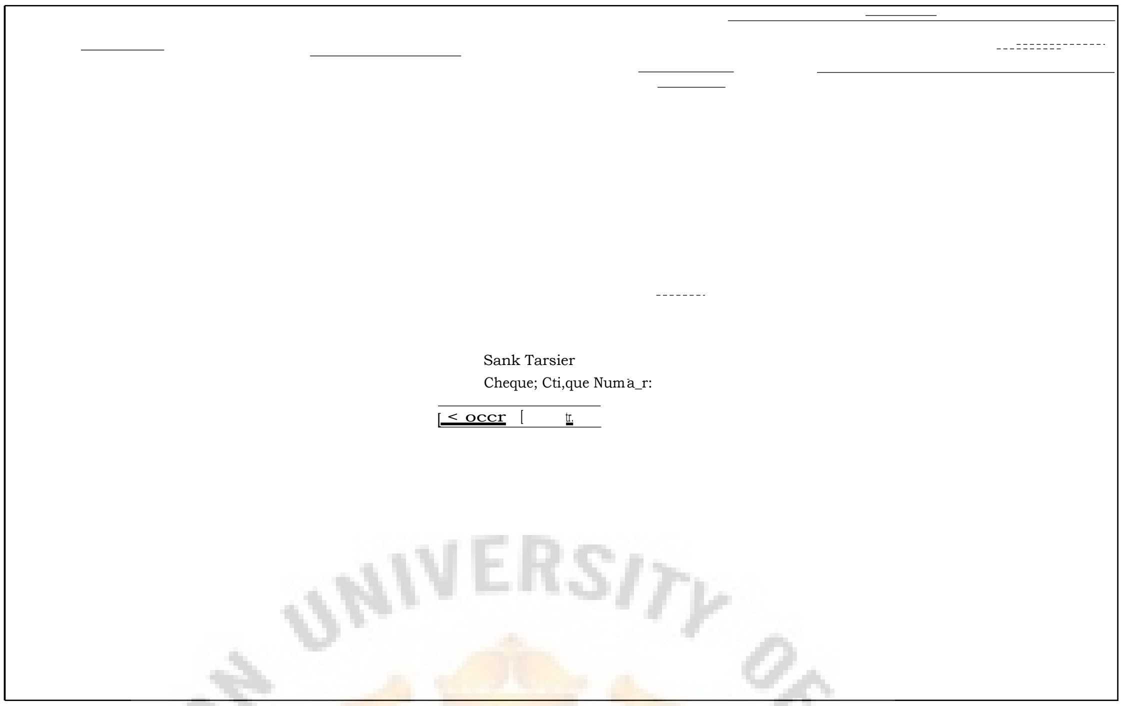

Summary report of job assigned to staff for October 2003

Figure C.17. Job Staff Bar Chart.


Sank Tarsier

Cheque; Cti,que Numa_r:

< occr

Figure C.18. Add Income.

Figure C.19. Income List.

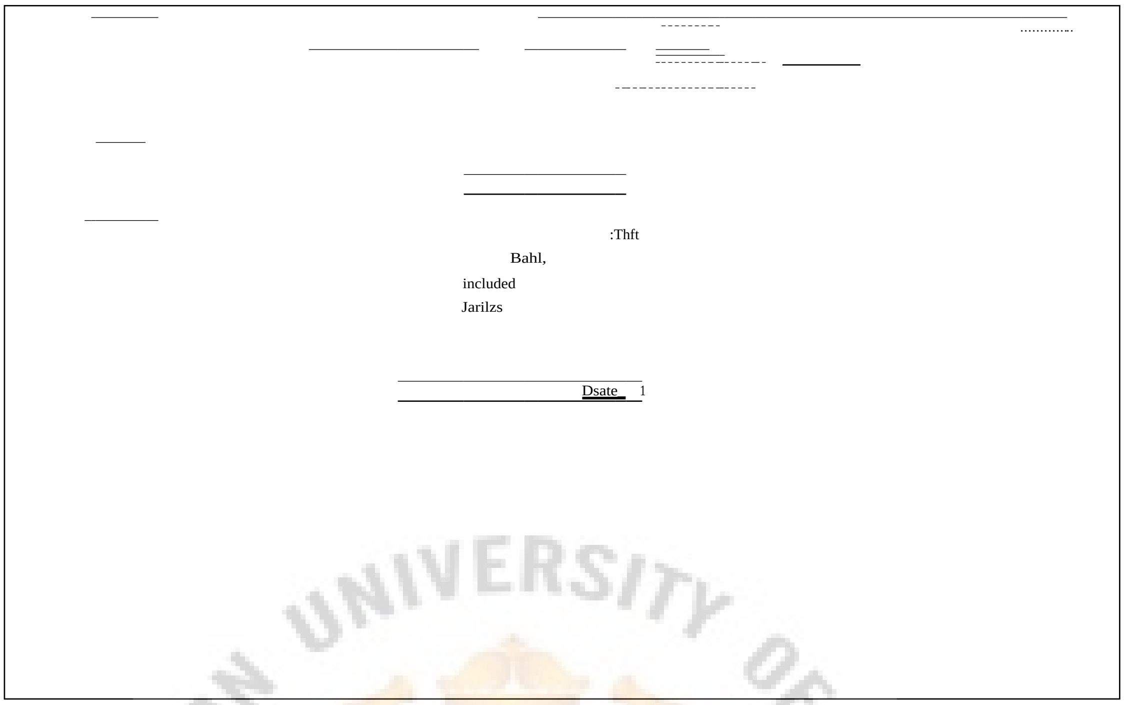


Figure C.20. Add Income Approve.

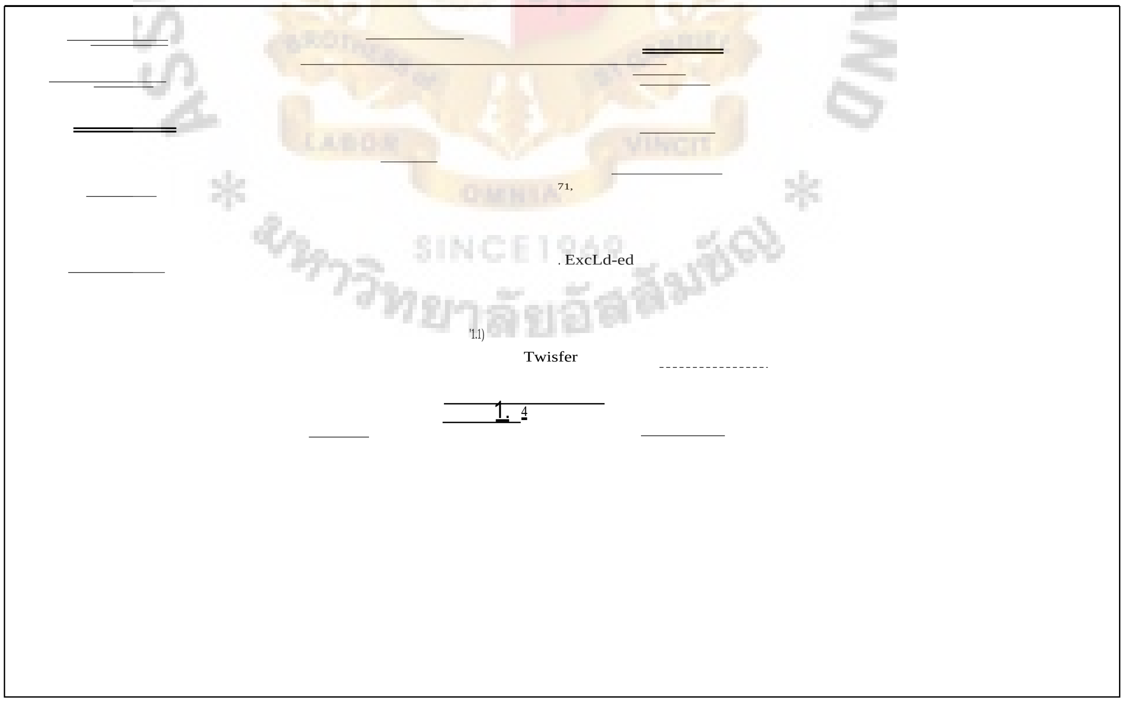
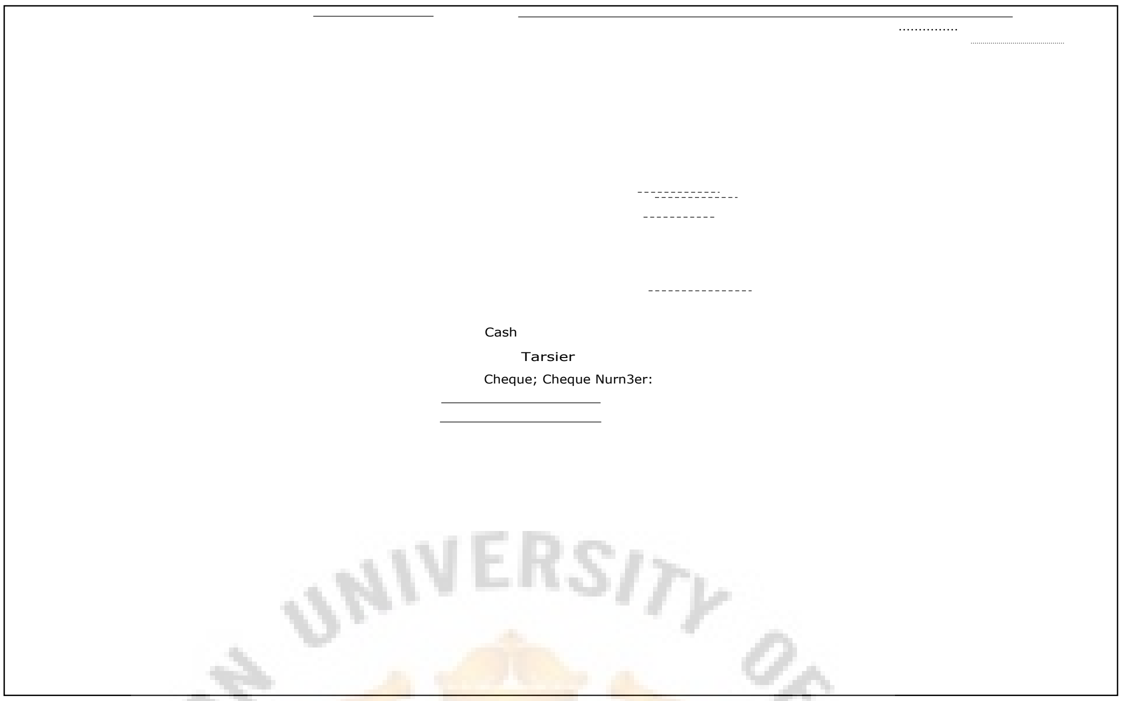


Figure C.21. Income Edit.

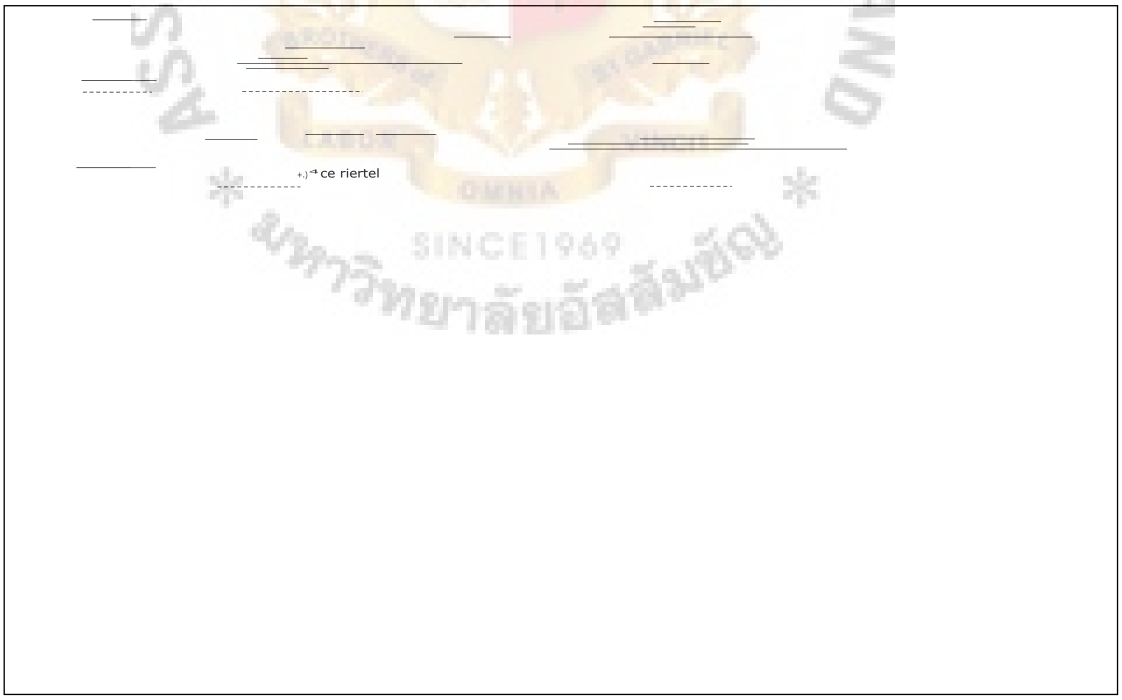


Cash

Tarsier

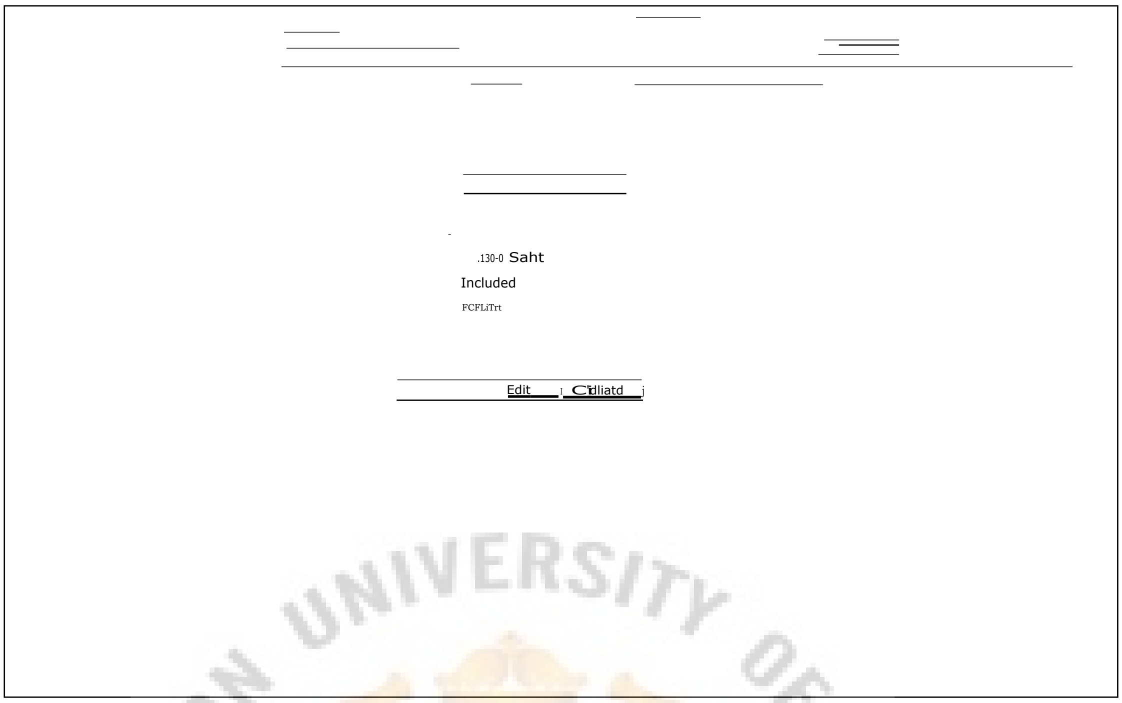
Cheque; Cheque Nurn3er:

Figure C.22. Add Expense.



+.)⁻⁴ ce riertel

Figure C.23. Expense List.

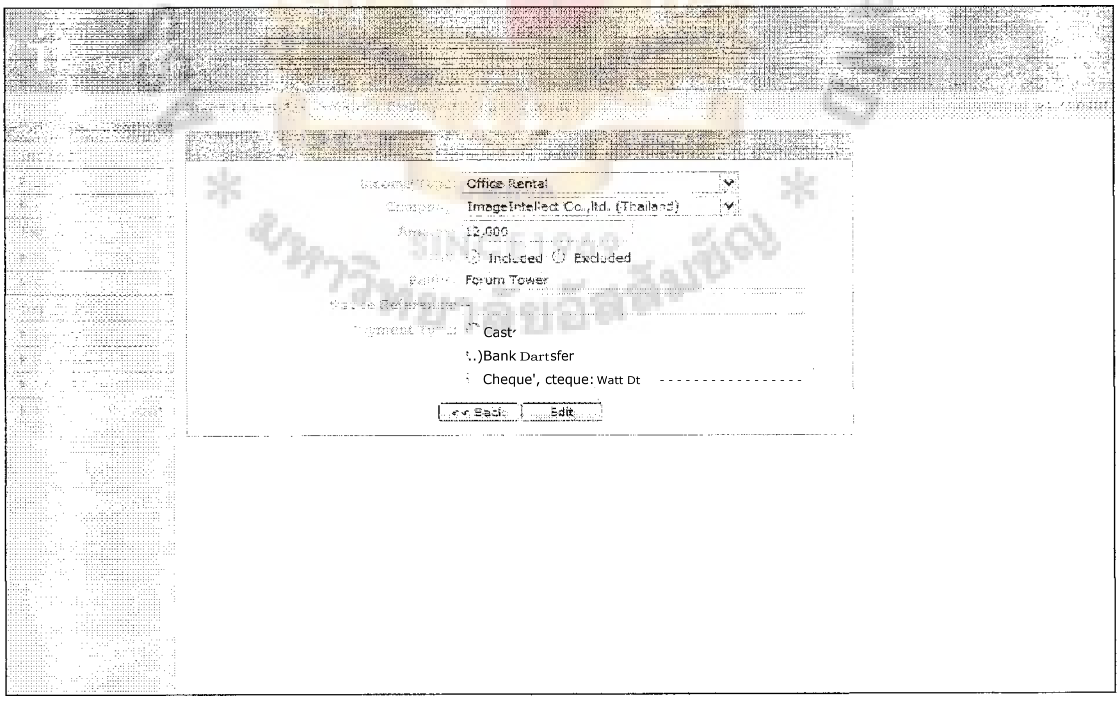


.130-0 Saht

Included

FCFLiTrt

Figure C.24. Add Expense Approve.



Office Rental

ImageIntellect Co.,ltd. (Thailand)

12,000

Included Excluded

Forum Tower

Cast'

..)Bank Dartsfer

Cheque', cteque: Watt Dt

Figure C.25. Expense Edit.

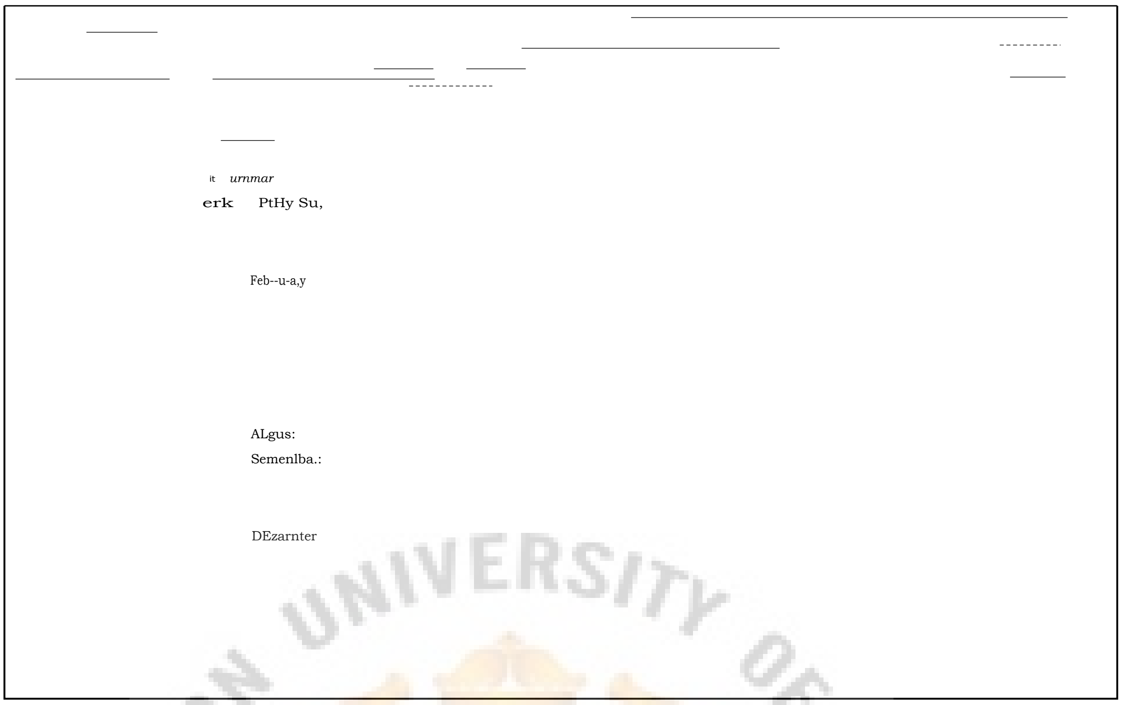

it urnmar
erk PtHy Su,
Feb--u-a,y
ALgus:
Semenlba.:
DEzarnter

Figure C.26. Income Overview Report.

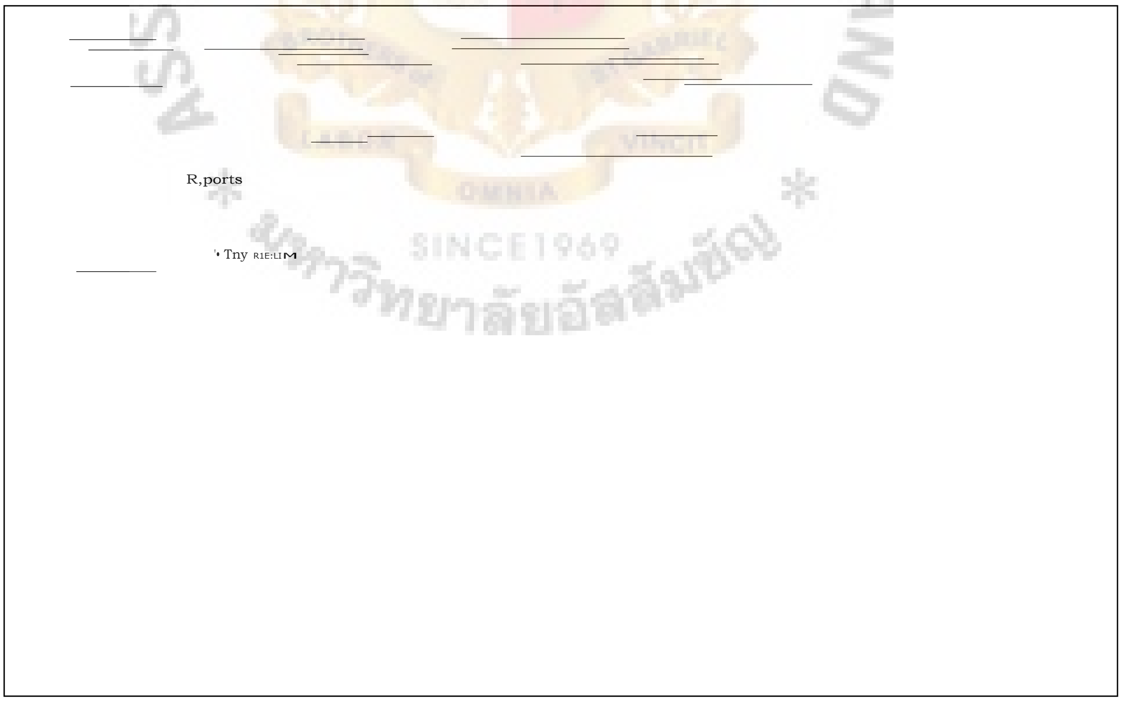
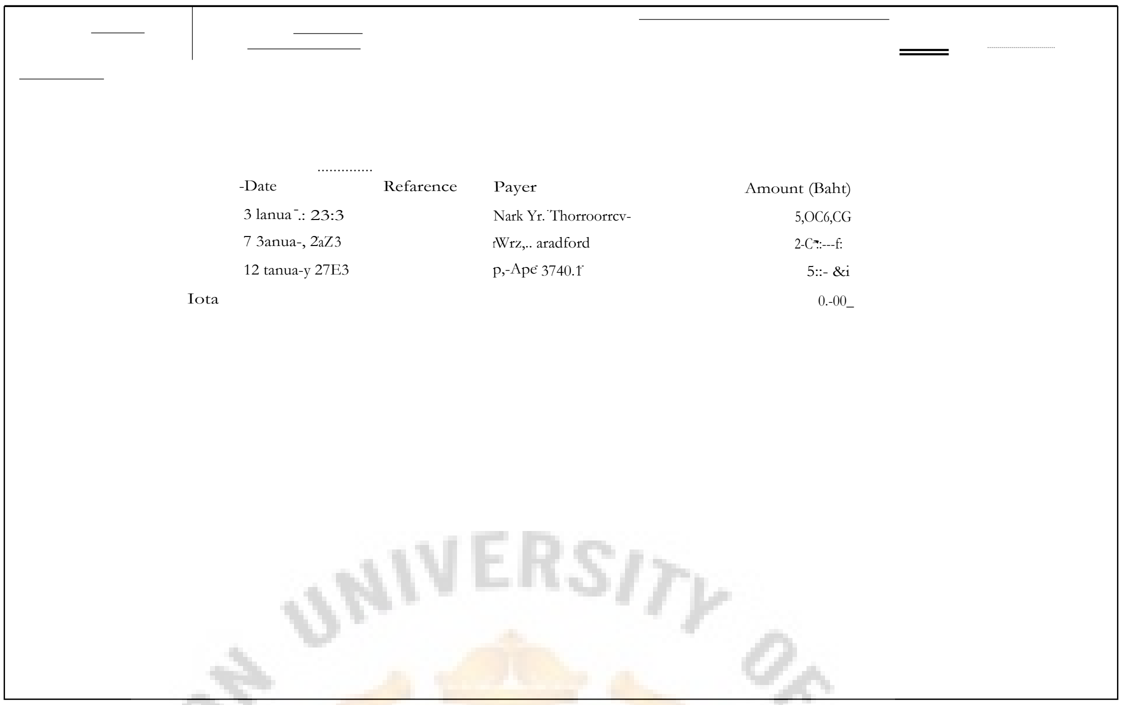
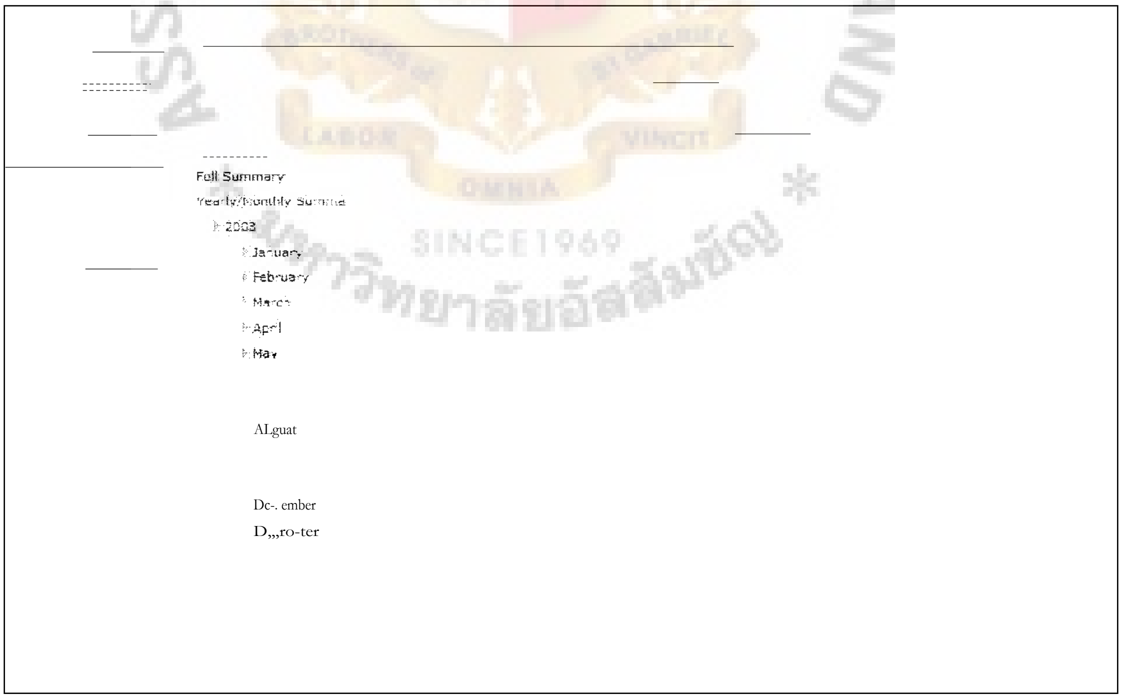

R,ports
'• Tny RIE:LIM

Figure C.27. Income Type Report.



-Date	Refarence	Payer	Amount (Baht)
3 lanua‾.: 23:3		Nark Yr. Thorroorrcv-	5,OC6,CG
7 3anua-, 2aZ3		rWrz,.. aradford	2-C‾::---f:
12 tanua-y 27E3		p,-Ape 3740.1	5::- &i
Iota			0.-00_

Figure C.28. Income List Report.



Full Summary
Yearly/Monthly Summa
2003
January
February
March
April
May
ALguat
Dc-. ember
D,,,ro-ter

Figure C.29. Expense Overview Report.

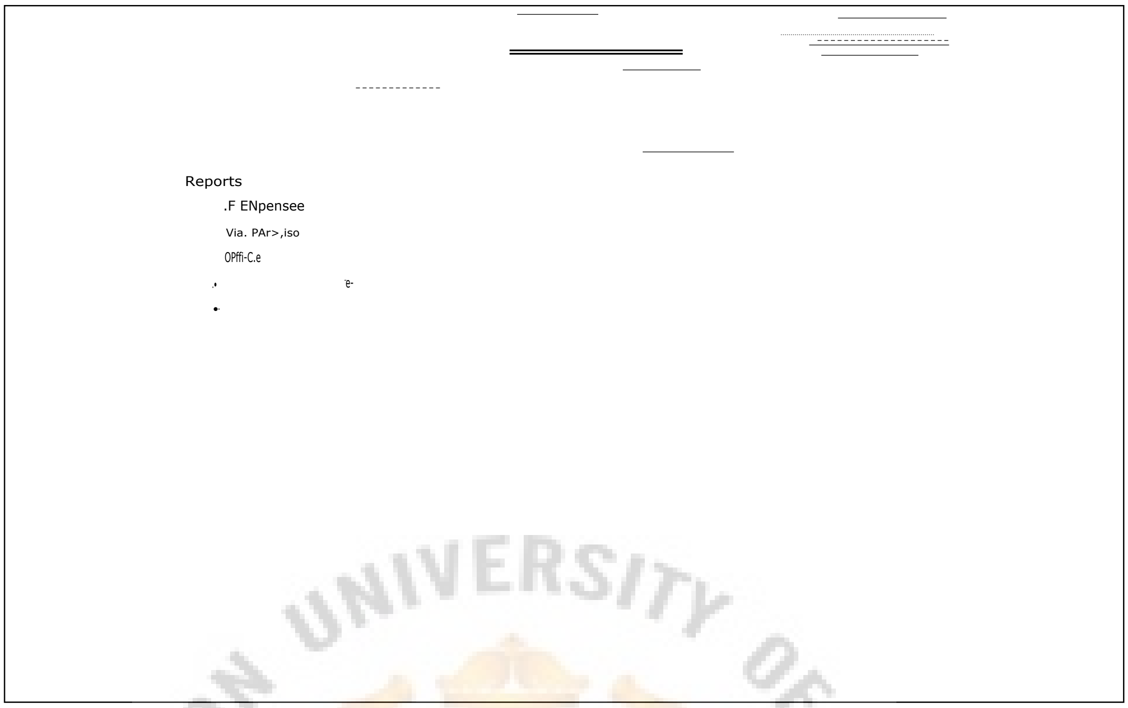
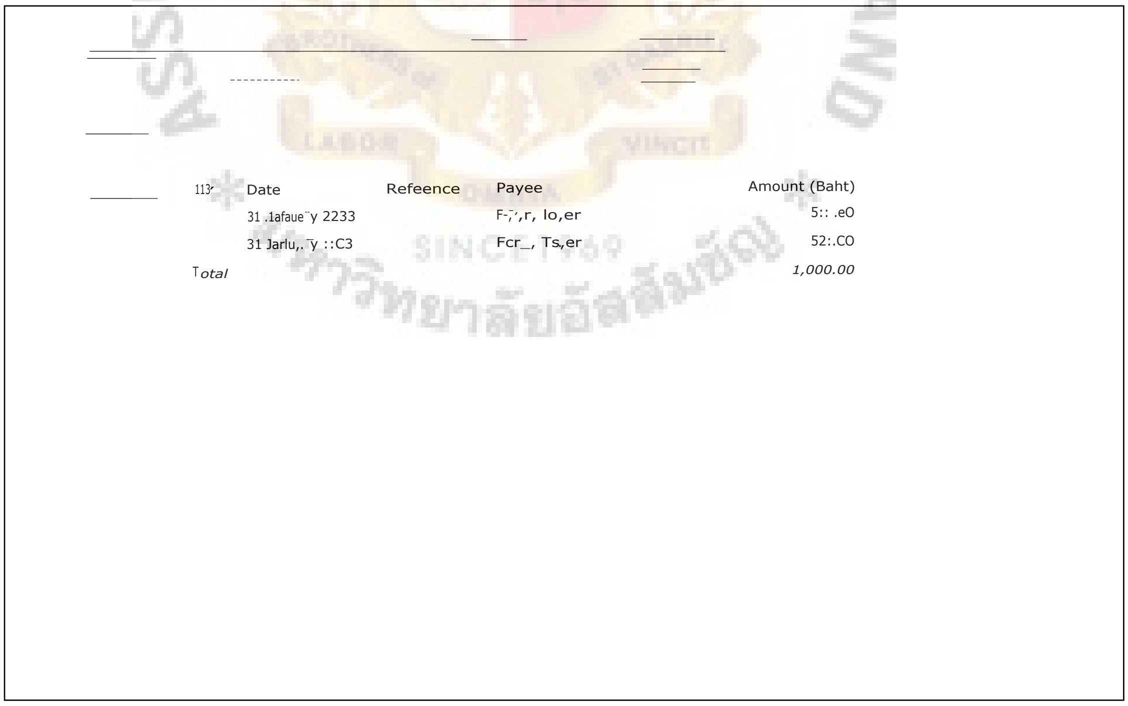
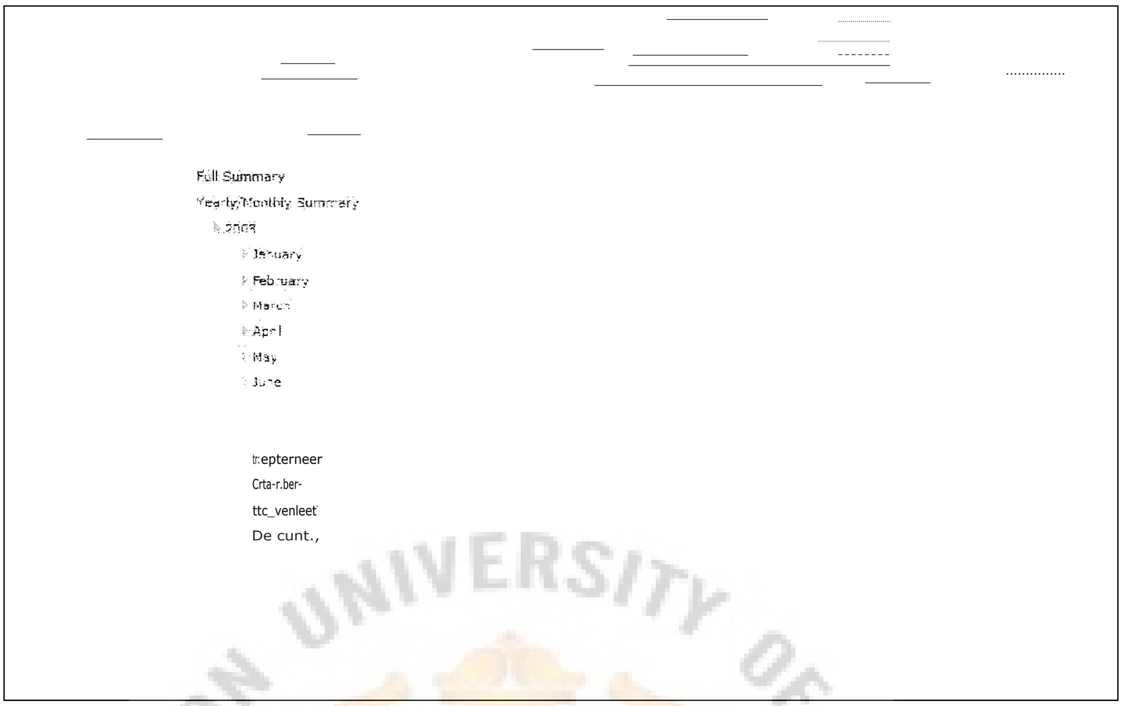


Figure C.30. Expense Type Report.



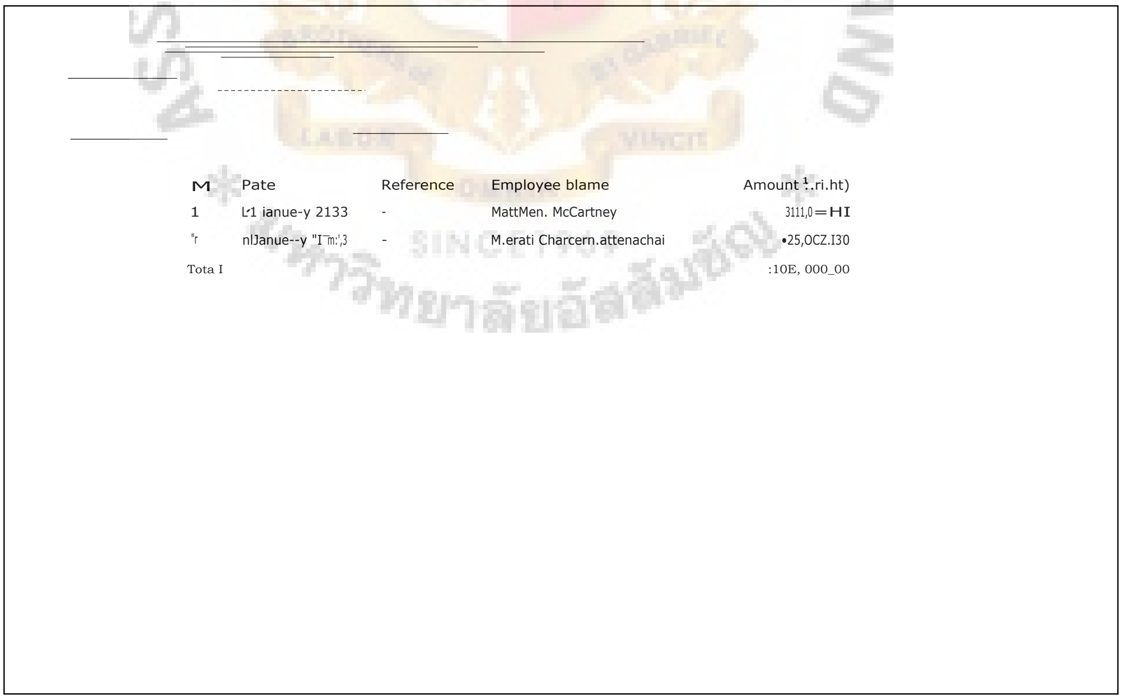
113'	Date	Refeence	Payee	Amount (Baht)
	31 .1afaue¨y 2233		F-,',r, lo,er	5:: .eO
	31 Jarlu,.¯y ::C3		Fcr_, Ts,er	52:.CO
Total				*1,000.00*

Figure C.31. Expense List Report.



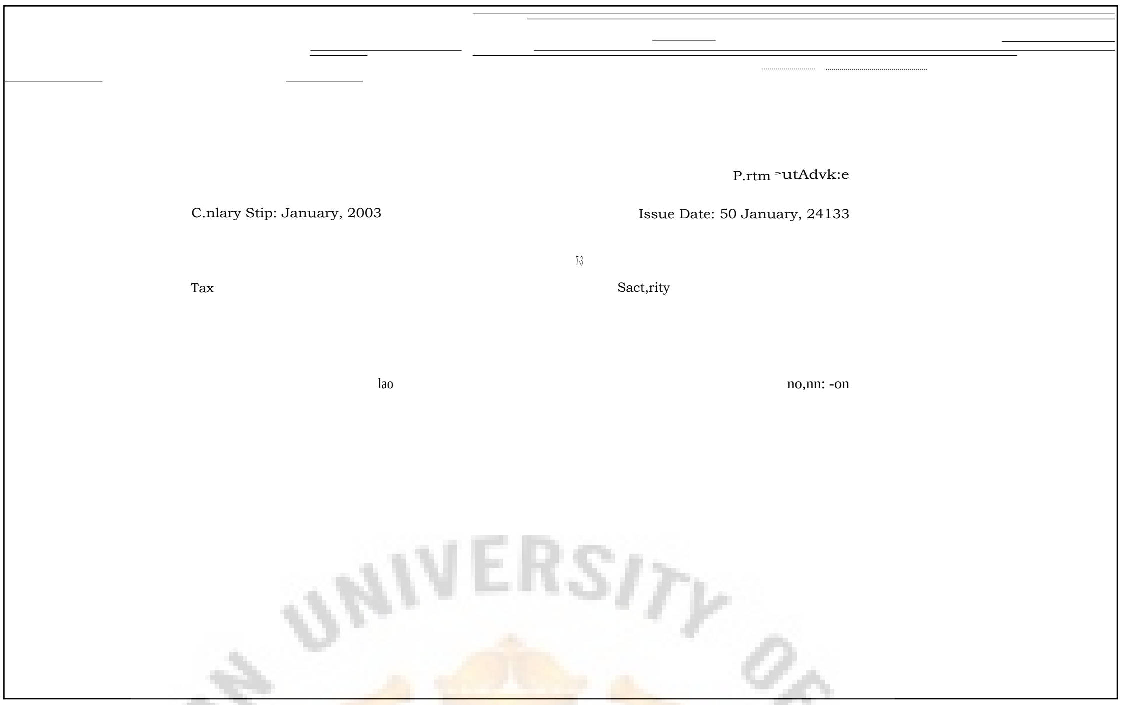
Full Summary
Yearly/Monthly Summary
2008
January
February
March
April
May
June
tr:epterneer
Crta-r.ber-
ttc_venleet
De cunt.,

Figure C.32. Payroll Overview Report.



M	Pate	Reference	Employee blame	Amount 1.ri.ht)
1	L1 ianue-y 2133	-	MattMen. McCartney	3111,0=HI
"r	nlJanue--y "I m:',3	-	M.erati Charcern.attenachai	•25,OCZ.I30
Tota I				:10E, 000_00

Figure C.33. Payroll List Report.



P.rtm ~utAdvk:e

C.nlary Stip: January, 2003

Issue Date: 50 January, 24133

Tax

Sact,rity

lao

no,nn: -on

Figure C.34. Payroll Detail.

APPENDIX D

PROGRAM STRUCTURAL DESIGN

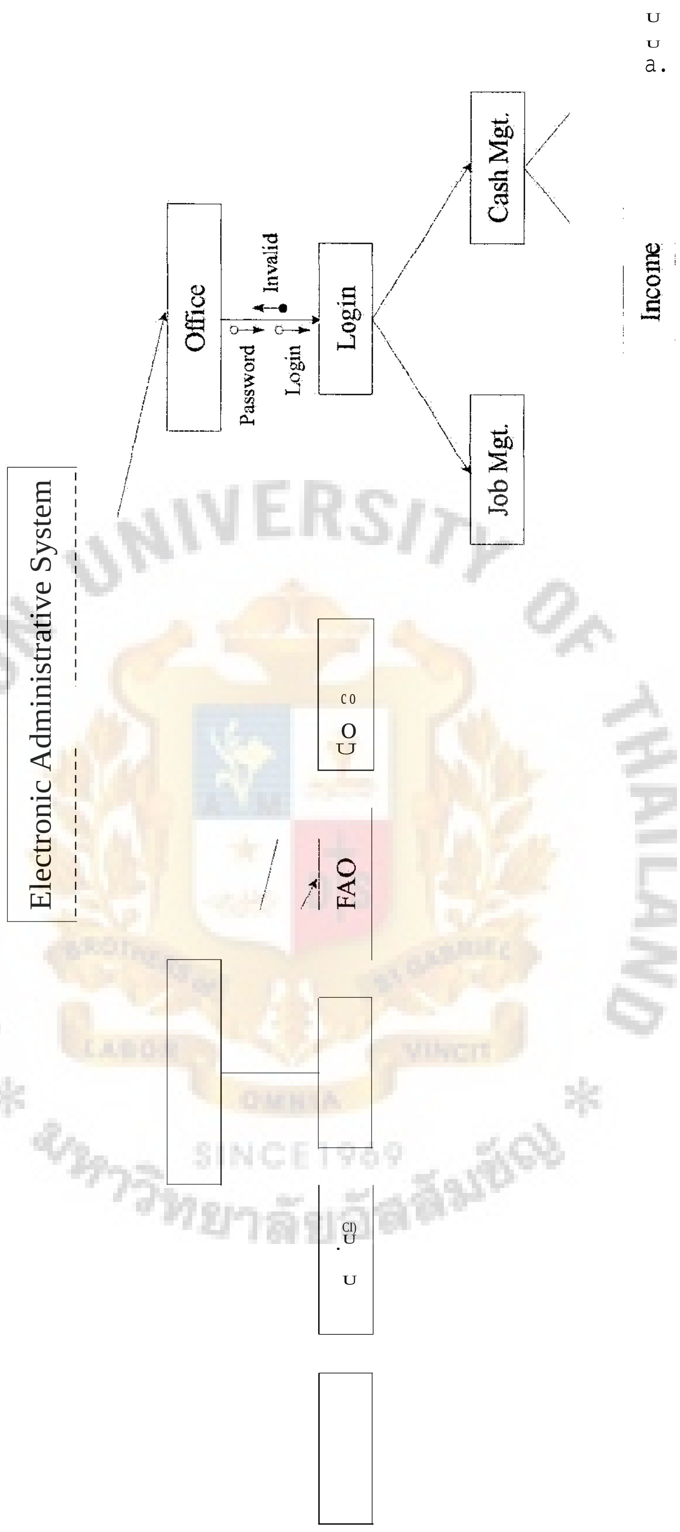


Figure D.1. Overall Process.

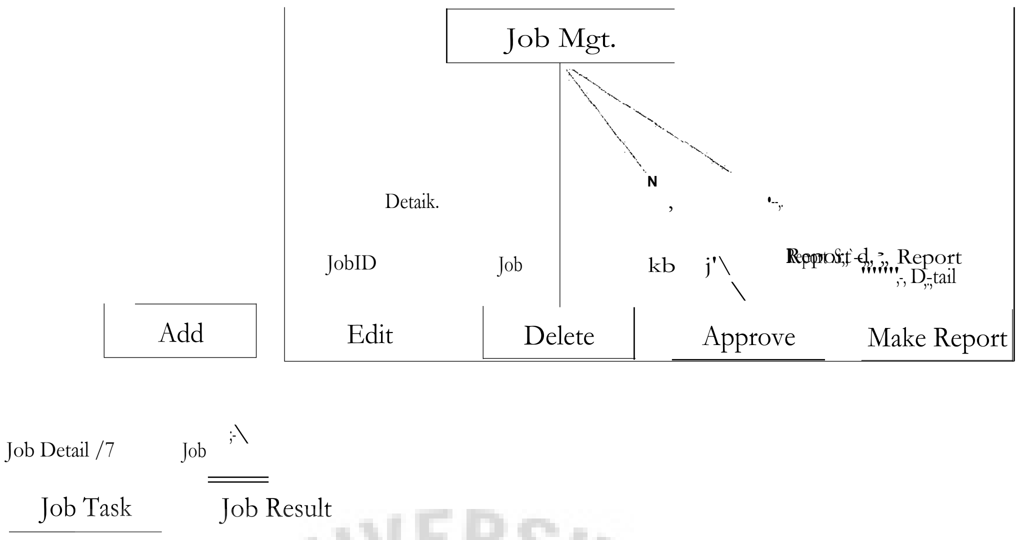


Figure D.2. Job Process.

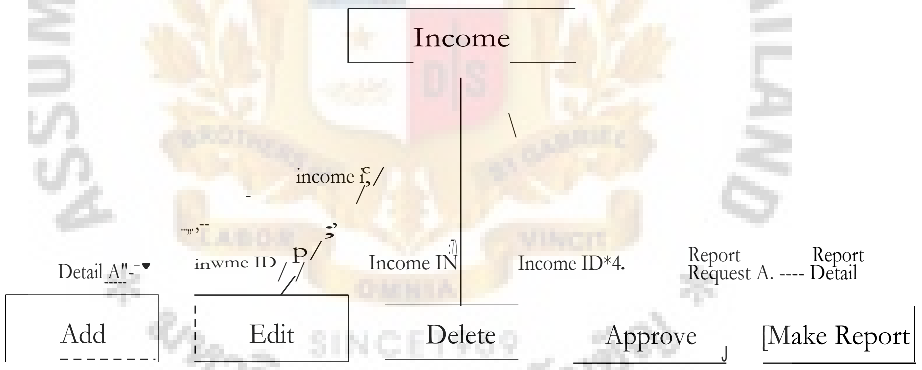


Figure D.3. Income Process.

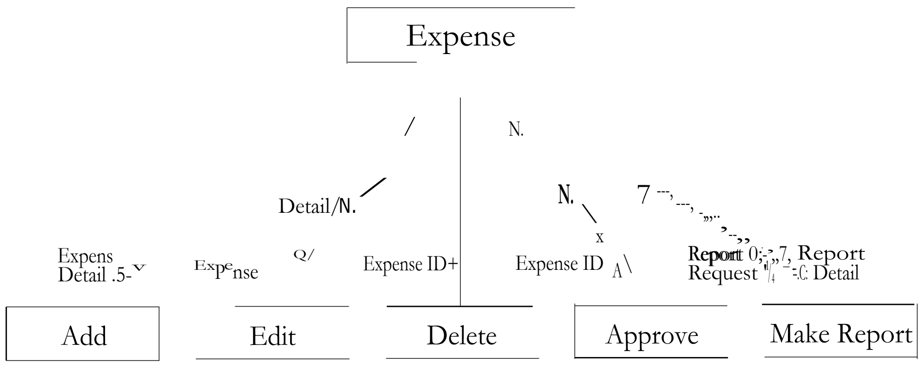


Figure D.4. Expense Process.

BIBLIOGRAPHY

English References

1. Chris Bates Web Programming. West Sussex: John Wiley & Sons Ltd., 2002.

2. Esen Ozkarahan "Database Management," Concepts, Design, and Practice. New Jersey: Prentice-Hall, Inc., 1990.

3. Matt J. Crouch Web Programming with ASP and COM. Massachusetts: Addison Wesley Longman, Inc., 2000.

Peter Rob, Carlos Coronel "Database Systems" Design, Implementation, and Management. Massachusetts: Boyd & Fraser Publishing Company, 1995.

5. L.A. Maciaszek: Database Design and Implementation. West Sussex: Prentice Hall of Australia Pty Ltd., 1990.

6. Sten E. Vesterli: Oracle Web Applications 101. California: Osborne/McGraw-Hill, 2001.

Thai References

1. V59117EI e1'Sq1̍11/1. V1JJ515 Web Design firla@DflU1111611bIllhillillaOli1̍1/1. flIVV11111111111M: 11M171-1, 2544.

2. 55919f€J dirprol. twm̍lcirrit h 111511714