LEASING SYSTEM OF OLAN MOTOR

by

Mr. Srikieat Sriprawatkul

A Final Report of the Three - Credit Course
CS 6998 System Development Project

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer Information Systems
Assumption University

November, 2000

Leasing System of Olan Motor

by
Mr. Srikieat Sriprawatkul

A Final Report of the Three-Credit Course
CS 6998 System Development Project

Submitted in Partial Fulfillment
of the Requirements of the Degree of
Master of Science
in Computer Information Systems
Assumption University

November 2000

Project Title	Leasing System of Olan Motor.
Name	Mr. Srikieat Sriprawatkul
Project Advisor	Assoc.Prof.Dr. Suphamit Chittayasothorn
Academic Year	November 2000

The Graduate School of Assumption University has approved this final report of the three-credit course, CS 6998 System Development Project, submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Information Systems.

Approval Committee:

(Assoc.Prof.Dr. Suphamit Chittayasothorn)
Advisor

(Prof.Dr. Srisakdi Charmonman)
Chairman

(Air Marshal Dr. Chulit Meesajjee)
Dean and Co-advisor

(Asst.Prof.Dr. Vichit Avatchanakorn)
Member

(Assoc.Prof. Somchai Thayarnyong)
MUA Representative

November 2000

ABSTRACT

Today, information systems provide the communication and analytic power that firms need for conducting trade and managing businesses. The power of computer hardware and software are available to support management decision making which helps the firm gains a strategic planning.

The main objective of the development is to create a new system that allows the company to have more opportunity to growth and expansion by the computerized system. This project emphasizes on designing the new system that improves the existing operation in the leasing system of the company instead of using manual operation.

The proposed system is developed in accordance with the systems analysis and design techniques. It covers the user requirement, analysis, design, hardware and software requirement, security and control, cost and benefit analysis, and system implementation. The new system will serve computerized operations and produce input, process and generate output more efficiently.

ACKNOWLEDGEMENTS

The writer thank his advisor, Dr. Supamit from his heart and stand in deepest gratitude for the assistance provided by him. His invaluable guidance, advice, and support has molded this project into a presentable piece of work. He deem myself fortunate as he studied under his wings at the Assumption University. He felt secured enough because of his kind help and moral support during my entire stay at ABAC.

He wish to thank the entire faculty of Graduate School, especially Prof. Dr. Srisakdi Charmonman, Chairman CIS board and Air Marshal Dr. Chulit Meesajjee, Dean MS(CIS) for providing building blocks to this course and insights into the field of Computer information system, which is of utmost importance for our knowledge.

He would likes to acknowledge the assistance provided by all the personnel at Leasing of many company in carrying out the data collection despite their busy schedules

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

1.1 Background of the project

Globalisation of the world's industrial economies greatly enhances the value of information to the firm and offers new opportunities to business. Today, information systems provide the communication and analytic power that firms need for conducting trade and managing businesses. Technical change moves much faster than humans and organizations are changing. The power of computer hardware and software has grown much more rapidly than the ability of the organizations to apply and use this technology. To stay competitive, many organizations actually need to be redesigned. They will need to use information technology to simplify communication and coordination, and eliminate unnecessary work.

With the passage of time, every organization grows in term of size, complexity and specialization. The demand and needs of reliable, accurate, timely and economical information by people at all levels of the management level also grows, at a much faster rate. Especially in the strategic decision-making, the management faces a variety of problems, usually of non-recurring in nature and of great importance that are handled at times under conditions of total uncertainty. Only with the provision of accurate, timely and complete information can this variety and uncertainty be possibly reduced.

Thus in today's context of information needs, without computerization manual provision of information is inadequate. The use of computerized system is also increased day by day. Since the use is increasing, it is not very expensive to have a computerized system.

Nowadays, computer system plays an important role in the business. It introduces the new era of leasing system. Moreover, the concept of information system, database

system design and new technology give more opportunity to manage and control the system efficiency.

Advent of the computer system, the problems for loss of opportunity such as inability to search the correct data, inability to know the history record are declined. Moreover, it also provides the benefit in searching, sorting and evaluating the information faster. However, the success or failure of the new concept in computer system depends on the good database design, well-organized data and system support. Olan Motor Co., Ltd. still uses a manual system to manage and control the leasing system. The new leasing system can be used to solve current problems and make the operation run smoothly and efficiently.

1.2 Objectives of the Project

The main objective of this project is to understand the existing system, design the new system with the following objectives:

(1) To analyze the existing system.

(2) To study the requirements of users and design computer information system according to those user's requirements.

(3) To design the new system development for the leasing system. This new system will be suitable for this operation environment and create outputs that support business operation management.

(4) Develop the new system to eliminate errors due to the manual system

(5) Providing timely and reliable information in decision-making for management team.

(6) Using computing system to keep information instead of using human intervention for fast accessing information.

(7) Good relational database structure for efficiency use of data and future usage.

(8) Report will be delivered to the manager on time since all procedures can be performed on schedule

(9) Increasing the knowledge and perception of all staff with new technology.

(10) To have a reference table and master file and establish appropriate data maintenance procedure

(11) To improve the company's operation by using the computer-based information system

(12) To enable the employees to accept the new system.

(13) Developing and giving knowledge of the new era to the employees.

(14) To establish quality data, maintenance methods in the proper way.

1.3 Scope of the Project

The project will cover major parts of the leasing system for construction material business, which includes the following topics:

Leasing management system:

(a) Determine the code for all contract that happen in the organization.

(b) Provide and update the contract's detail of all customers

(c) Provide and update leasing transaction.

(d) Verify and measure the accuracy of the customer's information.

Customer Information

(a) Determine the code for all customers.

(b) Provide and update the detail of customer's information.

(c) Classify the detail of all customers.

Contract Information

(a) Determine the code for all contracts.

(b) Provide the necessary information in each contract..

Report for management team

(a) Monthly report of the new customers

(b) Monthly payment report of each customer

(c) Monthly loan report of the customer

(d) Company report

(e) Customer information report

(f) Occupy motorcycle report

(g) Bad debt customer report

(h) Arrear report

(i) Account closed report

1.4 Deliverables

The Project Plan of this project of Leasing System of Olan Motor Co., Ltd. are the following:

Project introduction

(a) Background of the project

(b) Objectives

(c) Scope

Description of the proposed new system

(a) Background of the Organization

(b) Existing system

(c) Current problems and Area for improvement

Description of the proposed new system

(a) System Specification (User requirements)

(1) Context Diagram

(2) Data flow Diagram

(b) System Design

(c) Hardware and Software Requirement

(d) Security and controls

Project Implementation

(a) Overview of project implementation

(b) Test plan and results

Conclusions and Recommendations

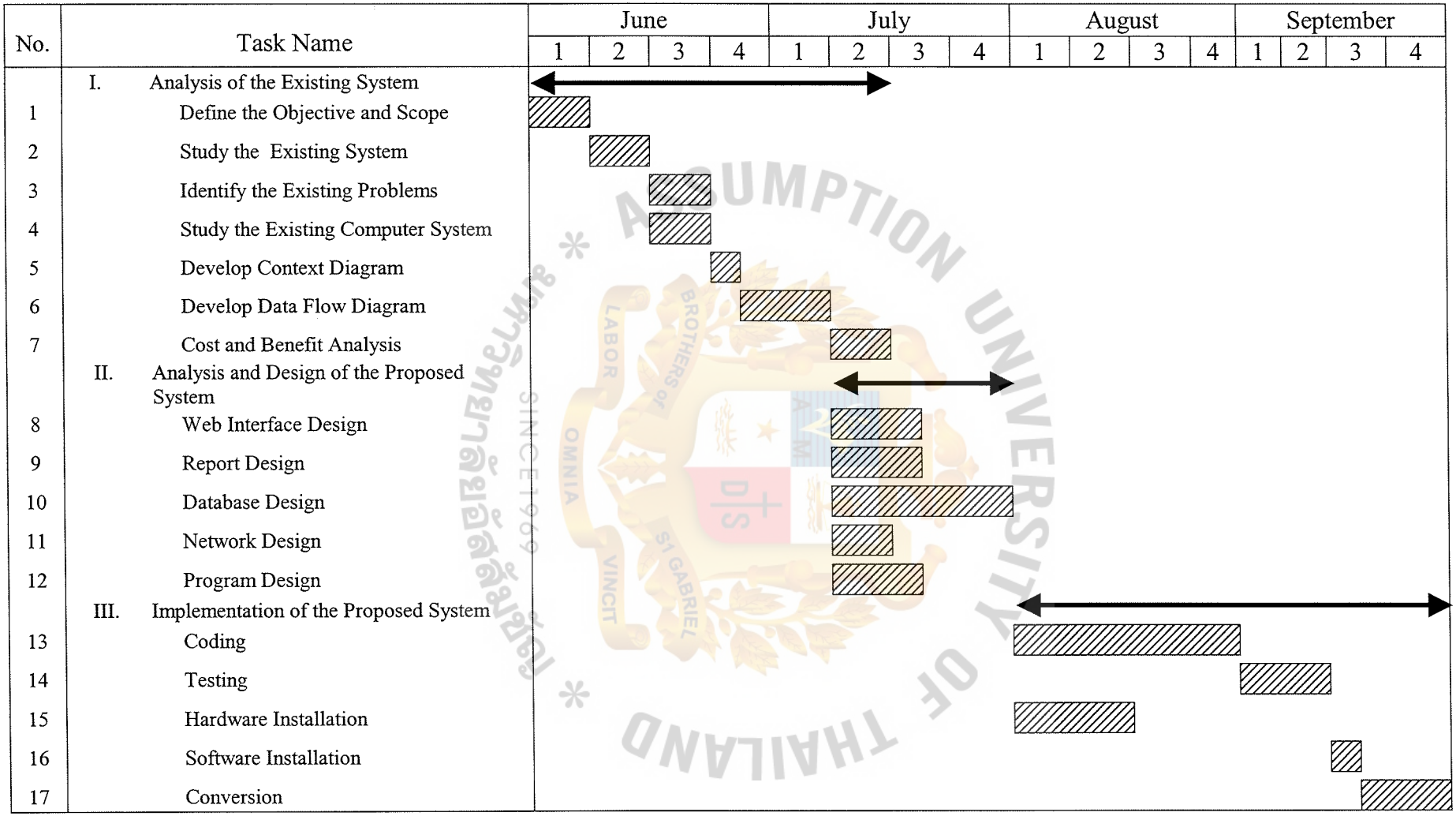

Figure 1.1. Project Plan.

II. EXISTING SYSTEM

2.1 Background of the Company

Olan Motor Co., Ltd is the small business, which has been established since 1970's. The company provides sale and service to the motorcycle business to generate sales and profits. This company is the agent of the four famous brand names from Japan. Those are HONDA, SUZUZI, YAMAHA and KAWASAKI. Not only sale and service all of these brand names, the company also has activities that involved in the motorcycle business such as exchange and trade the used motorcycle with the new ones, re-financing and pawn used motorcycles. The company also provides both cash and credit term in purchasing motorcycles. Moreover, Olan Motor joins with many finance companies in order to support the credit term to make the customer convenience.

With the long history of our company, it provides the company to the leader in motorcycle retailing business in Bangkok. The company has sold the motorcycle to both rural and urban area and also has a plan to sell our products abroad. Olan Motor also join venture with TOS Company that provides service in delivery of the product to the customers both domestic and international. It is very helpful to reduce the extra cost and make our product compete with other competitors.

Moreover, with the large sale volume of the company, it makes the complex information and stock system. The company needs to stock a lot of motorcycles, spare part and some important tools. It also needs to collect some important information. The company has many customers and suppliers. Moreover, the customers are subdivided into 2 parts: Wholesale customer and Retail customer.

However, most of the sale volume of the company come from retail customer (85% of total sale). Most of the retail customers purchase our product in credit term. It is very necessary to have the system that supports the leasing system.

In the existing system, the company has 2 methods to manage with the credit customers. Firstly, the company uses the leasing company to give the credit of the customer. However, It is very awkward and takes some time to complete the contract. It is also inconvenient and inflexible to do the business in the long term. Using the leasing company, there are a lot of documents and many processes to complete the contract. It wastes time to wait for the checker, who checks the customer's chronology and prepare the document to the leasing company.

Secondly, the company gives the credit to the customer. This method uses manual system to control, maintain and manage the system. This method is useful with the small number of customers. However, there are a lot of customers in the list of company. It is very hard to manage the system with the large number of customers

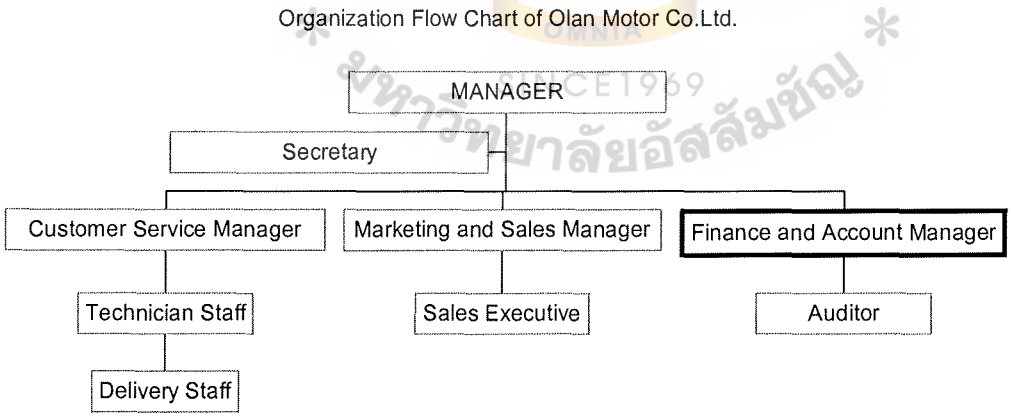

Figure 2.1. Organization Chart of Olan Motor Co., Ltd.

2.2 Existing Business Function

The company has 3 main departments. Each department's function is given below:

(1) Customer Service Department: The function of the Customer Service Department is concerned with the service to the customers. Most of the activities in this department are to take care and supervise the customer through the process of the system.

(2) Marketing Department: The main function of this department is involved in purchase inventory, control and manage the stock of the company. Moreover, it created the tactic and strategy for the organization in order to increase the sale volume. Some activities in this department are like sales department.

(3) Finance and Account Department: This department supervises the financial and account of the company. The main duties of this department are control, manage and evaluate the leasing system. Most activities of the new system are involved in this department.

2.3 Current Problems and Areas for Improvement

The current problems with the existing system are as follows:

Security: In the present system, all the information is kept in the record and unauthorized person can access this data. They can modify, delete and update the data easily so there is no security in the existing system.

Data Inconsistency: In the present system, many unrelated data are written in the record and some date are even written more than once in the same record in different formats. It is very dangerous to have the record that gives the incorrect data.

Waste storage space: Manual system is the paper work and requires some storage space to keep the data in the proper manner.

Work and time: Due to incorrect information and because of no security the workload becomes more since they have to again go and verify all the records and find out the correct data. Because the extra work take more time, the work of whole company may get some effect.

In order to solve the existing problem, the company decides to manage the leasing program by their own. Using the information and computerized system, it can manage and maintain the system easily.

Areas for Improvement

(1) The leasing system is designed for the financial and account department. It used to prepare the calculation of the payment of the customer automatically and notice the users when the customer has some problems occur.

(2) The new system helps the financial and account department to manage and control the financial status of the company.

(3) The organization can reduce human labor and human error by using the new system.

(4) The daily work is more systematic. The new system provides more reliable and correct information for decision-making and forecasting for management team.

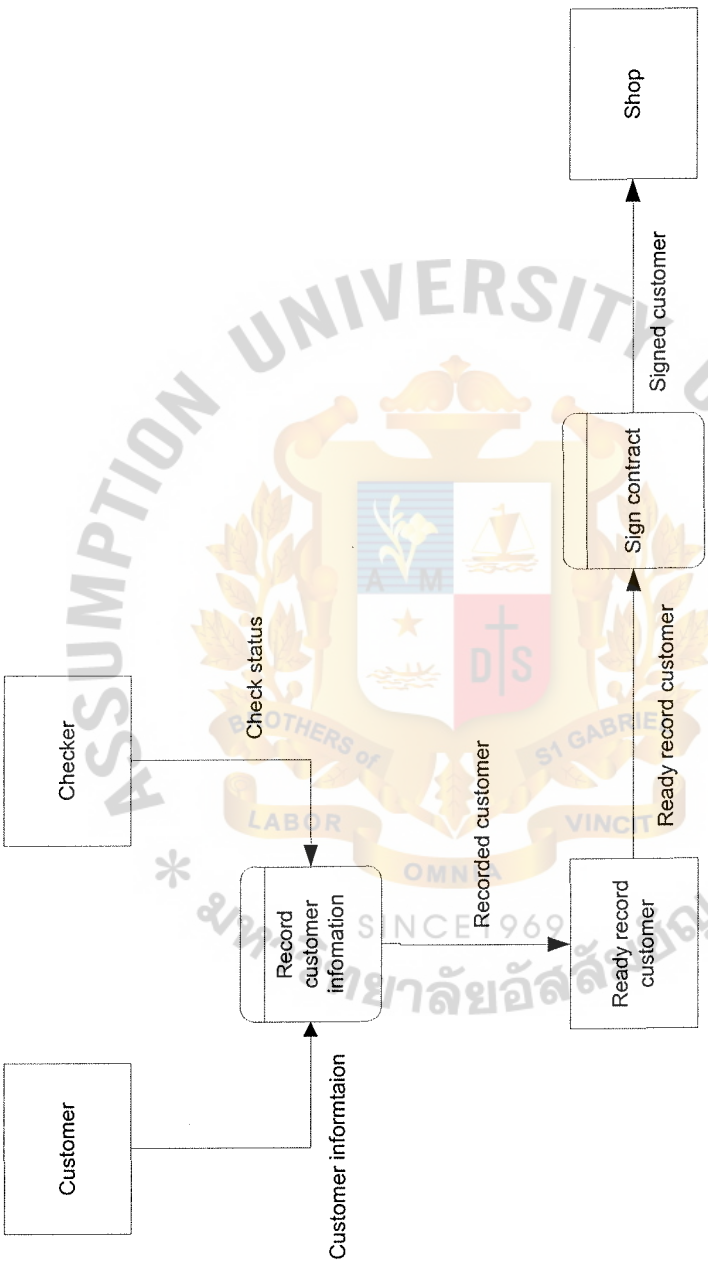

Figure 2.2. Making Contract of Existing System.

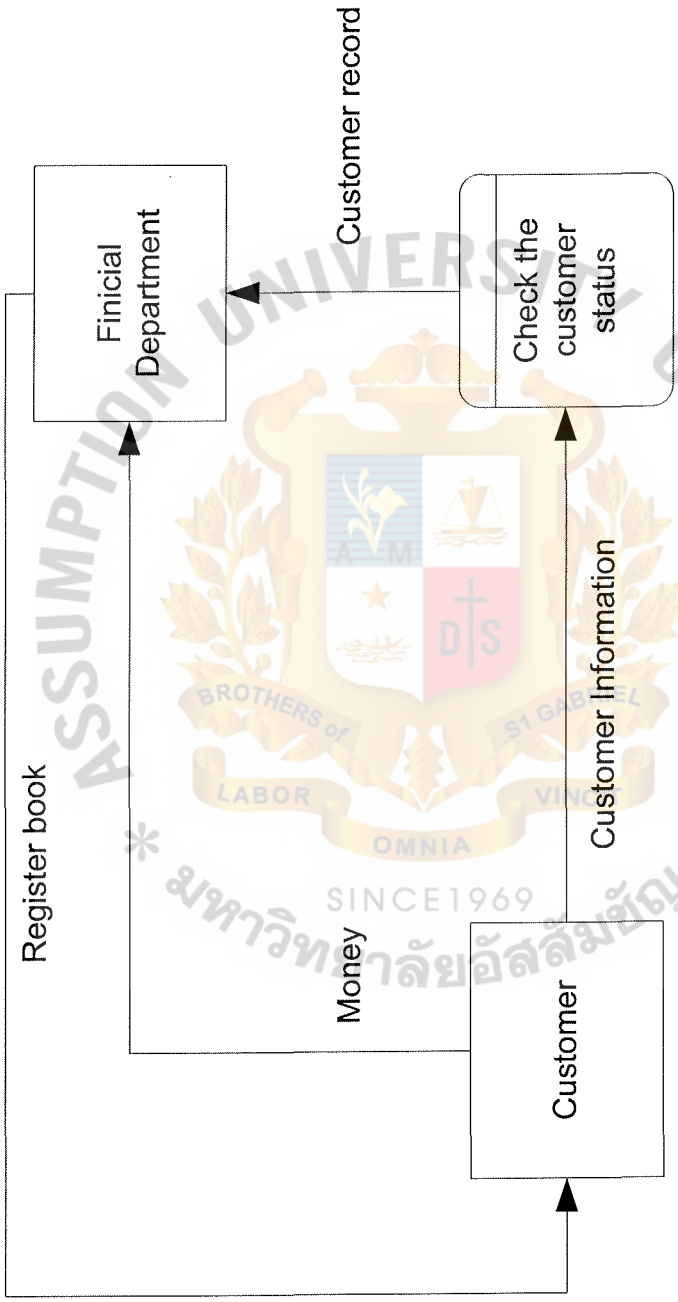

Figure 2.4. Closed Account of Existing System.

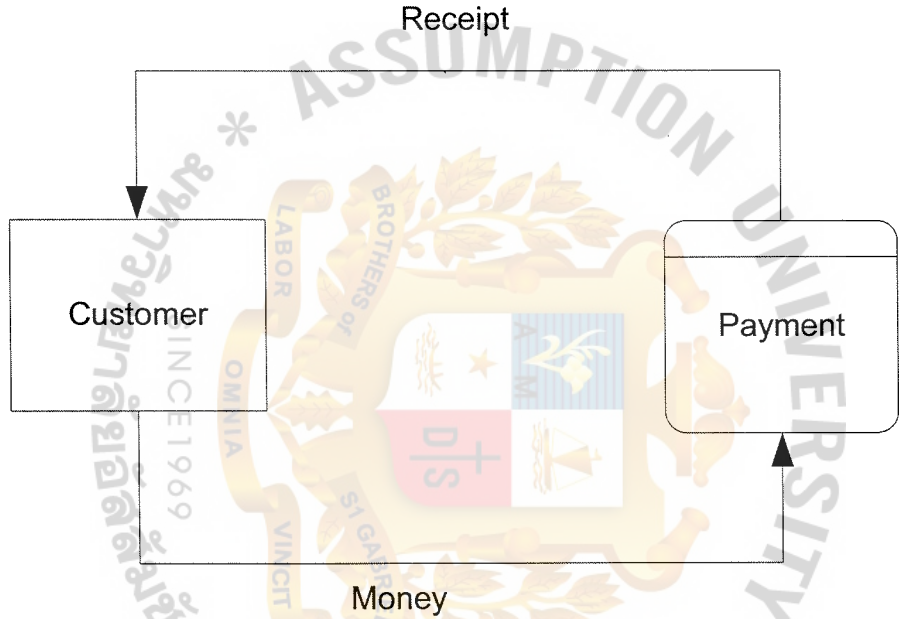

Figure 2.3. Payment Process of Existing System.

III. PROPOSED SYSTEM

3.1 User Requirements

The proposed system is required to solve the problems of the existing system, which are previously mentioned. The following are requirements of the user:

(1) The staffs are able to view the customer record from the computer at any time.

(2) The staffs take less time to obtain the required information such as customer status report, payment detail and financial status

(3) The proposed system can provide up-to-date and accurate information.

(4) The proposed system provides security by allowing only authorized person to access the data in the system.

(5) The proposed system has ability to exchange information easily with other department without any problems.

(6) The proposed system allows multiple users to access the database at the same time.

(7) The proposed system can display the historical and current of the customer status and payments period when demand needs.

(8) The proposed system is not difficult to be studied and used by users.

(9) The proposed system can reduce work and save time.

(10) The manger can retrieve data via the computer system

(11) The system should be able to reduce paper work.

3.2 System Design

In the proposed system the top-down approach is used and pseudocode are written as the modular approach for programming.

3.2.1 Top-down Approach

In designing the proposed system the top-down approach is used since it is easy to visualize the system with the help of a picture. The advantage of using this approach is it includes avoiding chaos of attempting to design a system "all a once." It has the ability to have separate system analysis team to be working in parallel on different but necessary subsystem.

3.2.2 Modular Development

Once the top-down design approach is taken, the modular approach is used in programming. This approach involves in breaking the programming into logical, manageable portions or module. While doing the modular programming the following points where kept in mind

(1) Keep each module to a manageable size (ideally including only one function).

(2) Pay particular attention to the critical interfaces (the data and control variables that are passed to other modules).

(3) Minimize the number of modules the user needs to modify when making changes.

(4) Maintain the hierarchical relationships set up in top-down phases.

3.3.3 Pseudocode

The uses of pseudocode are common and can be understood easily as they are written in simple English. It explains the logic of each module. The data flow diagrams are also used to write the pseudocode logic. In the proposed system we have the data flow diagram. The pseudocodes are shown in the process specification. The process specification has been put in Appendix C.

3.3 Hardware and Software Requirements

The proposed system is an integrated system. There is an on-line processing system running in a PC-LAN environment but the Financial and Account Department did not have a computerized system so the proposed system is made by keeping in mind that the system will attached to the other departments. By having an integrated system the processing of the whole company will become better. The requirement of the hardware and software are as follows:

3.3.1 Hardware

The new system requires the Local Area Network system to share the whole database with the other department.

(1) File Server PC Compatible Pentium 600 MHz

(a)	CPU	Intel Pentium III 600 MHz
(b)	Memory	SD RAM 128 MB Bus 133 MH
(c)	Hard Drive	13 GB Seagate
(d)	Graphic Video	SVGA 8 MB of Memory
(e)	Display Model	17" Super VGA Color-digital
(f)	Keyboard	104 keys
(g)	CD-Rom	48X Creative
(h)	Power Supply	250 watts line automatic line switching
(i)	Miscellaneous	Floppy Drive, Sound Blaster

(2) Workstation 3 sets

(a)	CPU	Intel Pentium II MHz
(b)	Memory	SD RAM 128 MB Bus 133 MHz
(c)	Hard Drive	13 GB Seagate

(d)	Graphic Video	SVGA 8 MB of Memory
(e)	Display Model	15" Super VGA Color-digital
(f)	Keyboard	104 keys
(g)	CD-Rom	48X Creative
(h)	Power Supply	250 watts line automatic line switching
(i)	Miscellaneous	Floppy Drive, Sound Blaster

(3) Network Equipment

(a) UTP

(b) LAN bit Multi Switching (Hub) 12 Port

(c) Ethernet LAN Card 10/100 Mbps

(4) Printer

(a) Dot-matrix Printer

(b) Laser Print HP 6P

(5) Miscellaneous

(a) UPS (Uninterruptible Power Supply) 1KVA 1 set

3.3.2 Software

To configure the software requirements, it is required to set the operation system as well as the database and the application program. The system works under a Local Area Network

(1) Network operation system

(a) Microsoft Window NT server 4.0

It has the capability to be extended to at least 26 users concurrently in case of the expansion of the system. It can be configured to both bus and star topology.

(2) Operation system

(a) Microsoft Window 98

This is an operation system that is needed to run the application working at the client side. Without it, none of the application can work so it is one of the most important things to run the application.

(3) System Development system

(a) Microsoft Office 97

This is used for many purposes, as there are different kinds of things in Microsoft office. It has Microsoft Word that is a word processor to be used in typing letters or making a memo etc. Then Microsoft Excel that is to be used as the spreadsheet. There are other things with it that is to install such as MS-DOS, PowerPoint, Project Manager and Microsoft Exchange.

(b) Microsoft Access 97

This is used as the database in our project. It is the file server but the system uses it as the database system in order to keep the information. It is used to drive the application and manage the database files. The database of this system can be shared by other system in the company.

(c) Internet Explorer

This application is used to run the application that we made. We use this browser to show the input and output screen to the users.

3.4 Network Specification

The objective of network connection is sharing resources that are database or information, application programs, and hardware. This network allows multiple users to access resources at the same period of time. This system is designed by using Bus

topology that uses Hub to be the center of the connecting workstations

The components of the network configuration are defined as follows:

(1) Network Topology : Bus Topology

(2) Interconnection : Hub 12 ports

(3) Wiring and cable : UTP 4 Pairs CAT-5

(4) Server : PCs File Server

(5) Workstation : PCs

(6) Network Operation : Microsoft Window NT

(7) Network interface card

Network Architectures

Our project uses distributed date and logic. Sometimes called three-tiered or n-tiered client / server computing, this approach distributes database and business logic to separate servers. The advantages of architecture is when the number of clients grows, the three-tiered system introduce an application server. Typically, the application server supplies two services. First, it provides a transaction monitor to manage transactions. Second, some or all of the business logic of the application can be moved from the client to the server. The Client will be installed at the front office to operate the routine job. Application server and DB server will be kept in the computer room in order to make sure that the data will be kept in the secure place.This architecture is installed in the main office.

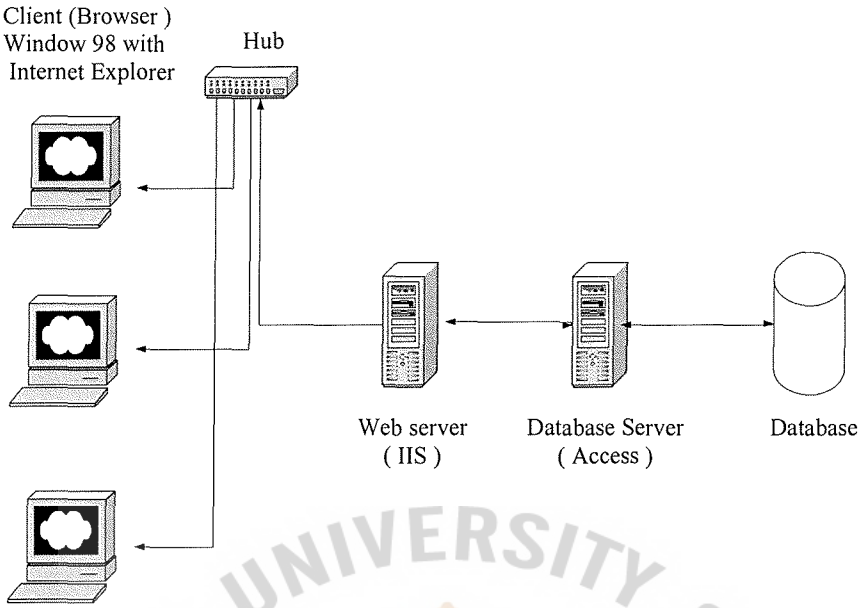

Figure 3.1. Network Architectures of the Proposed System.

3.5 Security and Controls

Security in computer is a vital issue. The major assets of computing system are hardware, software and data. An attack to the hardware renders data being processed as lost or unusable. There are risks of theft or malicious destruction of hardware. Software can also face the threat of copying of programs and can also be destroyed maliciously or it could be modified or deleted. Data attack is one of other serious problems as an unauthorized party might gain access to it and modify it or could lack the information. The proposed system also needs to provide security for the Local Area Network.

Considering the attacks that the computer system faces, the following security and control methods are proposed:

3.5.1 User-Oriented Access Control

(1) The user identifiers (ID) and passwords are assigned to authorized persons. When a user logs on, the system asks for both a user ID and a password.

(2) The system allows a user to logon only if that user's ID is known to the

system and if the user knows the password associated with the system with that ID.

(3) The user ID and password can determine the privileges accorded to the user. A few users (such as manager and assistant manager) have supervisory or poweruser status in window NT authorities that allow them to read files and to perform some functions.

3.5.2 Physical Security

(1) The failure of the main electricity supply causes interruption to the function of the computer facility or telecommunication network. UPS(Uniterruptible Power Supply) is used to supply power instead of the main electricity supply.

(2) Special detectors combined with removal or extractor fans and filters in the computer room and surrounding area are installed in order to protect against smoke and gas.

(3) The staffs working in the computer room are not allowed to eat, drink or smoke. These actions can cause damage to the computer.

(4) The computer hardware must be locked in the office at closing time, and the key should be entrusted to an authorized person.

3.5.3 Other Security

(1) Staffs have to back up the important information everyday in order to prevent the damage in the harddisk.

(2) Data correction must be made immediately after recovering errors on report.

(3) The historical and current data reports must be kept in categorized file for managerial planning.

(4) A virus-checking program will be installed for scanning virus before running

any program. Service information system will update the virus-checking program every 2 months.

(5) Staff should be provided with adequate training of how to use the system.

3.6 Cost and Benefit Analysis

3.6.1 Cost Analysis

To consider the finical aspects of the new system that is to be implemented, cost comparison should be made between the existing system and the new system. A comparison should be made on the investment cost, implementation costs, and annual operation costs.

Investment cost: These are non-recurring capital outlays to develop or acquire new equipment and technology such as hardware and software and other items which necessitate overall cost of the proposed system. Thus it should have the following consideration initialized for the proposed system:

(1) Future expansion policies: This means in future if the system is to expanded from workstation to more places it should have the capability to increases easily without having any problem.

(2) Computer life cycle: This life cycle of the computer should also be kept in mind as it also has a life period. With time the speed and other function of the computer starts finishing so if the new computers are put then the system should support them.

(3) Computer technology: Nowadays technology is changing at a rapid speed that is why we need to keep in touch with it so that if something new comes which will help a lot in the system we obtain that change easily.

(4) Space requirement: Since the existing system is manual it is managed by people in the company. They do not use the computer right now but with

time they start like the facilities of the computer system therefore the requirement of more space increases. Therefore the space requirement should be kept in mind.

(5) Speed necessity according to the nature of work: With time the speed of the system decreases. There are many reasons for that so to improve that it should have a facility.

The criteria of choosing the hardware have been concluded after discussing all the above details with the management by explaining the requirement of each of hardware and other items that are needed to have a good system which can be expanded easily in future. At the same time the requirement of the software were also discussed.

Hardware Requirement

(1)	File Server PC Compatible Pentium 600 MHz 1set		120,000	Baht
	(a) CPU	Intel Pentium III 600 MHz		
	(b) Memory	SD RAM 128 MB Bus 133 MHz		
	(c) Hard Drive	13 GB Seagate		
	(d) Graphic Video	SVGA 8 MB of Memory		
	(e) Display Model	17" Super VGA Color-digital		
	(f) Keyboard	104 keys		
	(g) CD-Rom	48X Creative		
	(h) Power Supply	250 watts line automatic line switching		
	(i) Miscellaneous	Floppy Drive, Sound Blaster		
(2)	Workstation 3 sets	70,000 * 3	210,000	Baht
	(a) CPU	Intel Pentium II MHz		
	(b) Memory	SD RAM 128 MB Bus 133 MHz		
	(c) Hard Drive	13 GB Seagate		

(d)	Graphic Video	SVGA 8 MB of Memory		
(e)	Display Model	15" Super VGA Color-digital		
(f)	Keyboard	104 keys		
(g)	CD-Rom	48X Creative		
(h)	Power Supply	250 watts line automatic line switching		
(i)	Miscellaneous	Floppy Drive, Sound Blaster		
(3)	Network Equipment		12,400	Baht
(a)	UTP			
(b)	LAN bit Multi Switching (Hub) 12 Port			
(c)	Ethernet LAN Card 10/100 Mbps			
(4)	Printer			
(a)	Dot-matrix Printer (Epson 217I, 24 pin)		7,000	Baht
(b)	Laser Print HP 6P		10,600	Baht
(5)	Miscellaneous			
(a)	UPS (Uninterruptible Power Supply) 1KVA 1 set		4,000	Baht
	Total hardware		364,000	Baht

Software Requirement

(1)	Network operation system			
(a)	Microsoft Window NT server 4.0		40,000	Baht
(2)	Operation system			
(a)	Microsoft Window 98		30,400	Baht
(3)	System Development system			
(a)	Microsoft Office 97		10,000	Baht
(b)	Microsoft Access 97			
(c)	Microsoft Internet Explorer			

Total Software	90,400	Baht

Implementation cost: These are basically the cost incurred to install the proposed system and are one-time cost outlays. So it includes the cost of development of system and also the implementation cost of new system. In developing of system, the cost of time taking is to be considered. After developing the implementation take place.

Implement Cost

Software development and training cost	40,000	Baht

Annual Operataion Cost: These are the recurring costs, which operate the system on a monthly or yearly basis depending on the nature of the business. It means the cost of software and hardware maintenance and consumable is taken in consideration.

Maintenance costs (per year)	10,000	Baht
Miscellaneous costs	5,000	Baht
Total annual operation cost	15,000	Baht

3.6.2 Cost Benefit Analysis

Tangible benefits: These are measured in baht, resources or time saved. The measurement of tangible benefits is not always easy. The example of tangible benefits are increase in speed of processing, access information on a more timely basis than was possible. After the implementation of the new system, we can accrue annual benefit from the following:

(1) Reduction of stationary and paper usage

(2) Reduction of human labor

(3) Increasing efficiency in processing

(4) Increasing the access to information on timely basis

(5) Advantage in decreasing the errors in updating the record

(6) Decrease the amount of time consumption

Intangible benefits: These are benefits that accrue in the organization due to the information system that are difficult to measure but are important nonetheless. Intangible benefits include improving the decision making process, enhancing accuracy, etc.

(1) Improving customer goodwill. The proposed system provides quick and efficient services for customers.The customers receive goods correctly and quickly. We can keep current customers and also may gain new customers.

(2) Provide timely, up-to-date, and accurate information/reports to support decision-making for management team.

(3) Reducing redundant process and data. Employees can work easily and increase speed of doing daily operation.

(4) Reducing human error from doing documentation.

(5) Improving employee morale. The new system operates some tasks instead of using human labor, so employees can save time to do other tasks and do not work overtime.

(6) Providing better communication between manger and staffs.

(7) Other departments have more chances to develop their own new system easily according to leasing system.

(8) Smoothing of the operation

Payback Analysis

System development costs are incurred long before benefits begin to accrue, so it will take some time for the benefits to overtake the costs. This period of time is called the "Payback period". Payback analysis determined how much time will lapse before accrued benefits overtake accrued and continuing costs.

From Figure 3.3, we conclude that our system has 2 years and 6 months Payback period

Net Present Value (NPV)

Net present value is a sophisticated capital budgeting technique, which is calculated by subtracting the project's initial investment from the present value of cash inflows discounted at a rate to the firm's cost of capital. The basic formula for NPV is as follows.

From Figure 3.4, the project has net present value 1,751,888 baht

Decision criteria

If NPV is positive, the investment is good. If negative, the investment is bad. So this project is in the good environment.

Table 3.1. Cost of Existing System, Baht.

Cost Items	Year 1	Year 2	Year 3	Year 4	Year 5
Personnel					
Financial Staff 6 positons	864,000	890,000	910,000	925,000	945,000
Receptionists 5 positions	600,000	625,000	655,000	689,000	698,000
Office Rental	50,000	50,000	50,000	50,000	50,000
Office Equipment	80,000	92,000	105,800	121,000	139,000
Utilities Cost	180,000	180,000	180,000	180,000	180,000
Other Cost	60,000	69,000	79,300	91,000	104,000
Total Cost	1,834,000	1,906,000	1,980,100	2,056,000	2,116,000
Cumulative Costs	1,042,000	2,948,000	4,928,100	6,984,100	9,100,100

Table 3.2. Cost of Proposed System, Baht.

Cost Items	Year 1	Year 2	Year 3	Year 4	Year 5	
Development Cost						
Personnel						
System Analyst / Design 1 position	144,000	0	0	0	0	
Programmer 2 position	288,000	0	0	0	0	
Hardware & Software						
Hardware	72,800	72,800	72,800	72,800	72,800	
Software	18,080	18,080	18,080	18,080	18,080	
Other Expenses						
Training	40,000	0	0	0	0	
Operation Cost						
Personnel						
Finicial staff 3 positions	432,000	450,000	480,000	510,000	535,000	2,407,000
Receptionists 2 positions	240,000	255,000	280,000	300,000	320,000	1,395,000
System Administrators 1 position	180,000	200,000	220,000	240,000	260,000	1,100,000
Utilities Cost	180,000	180,000	180,000	180,000	180,000	900,000
Expenses						
Computer Maintenance Cost	0	15,000	15,000	15,000	15,000	60,000
Other Expenses	5,000	5,000	5,000	5,000	5,000	25,000
Total Cost	1,599,880	1,195,880	1,270,880	1,340,880	1,405,880	5,887,000
Cumulative Costs	1,963,400	3,159,280	4,430,160	5,771,040	7,176,920	13,063,920

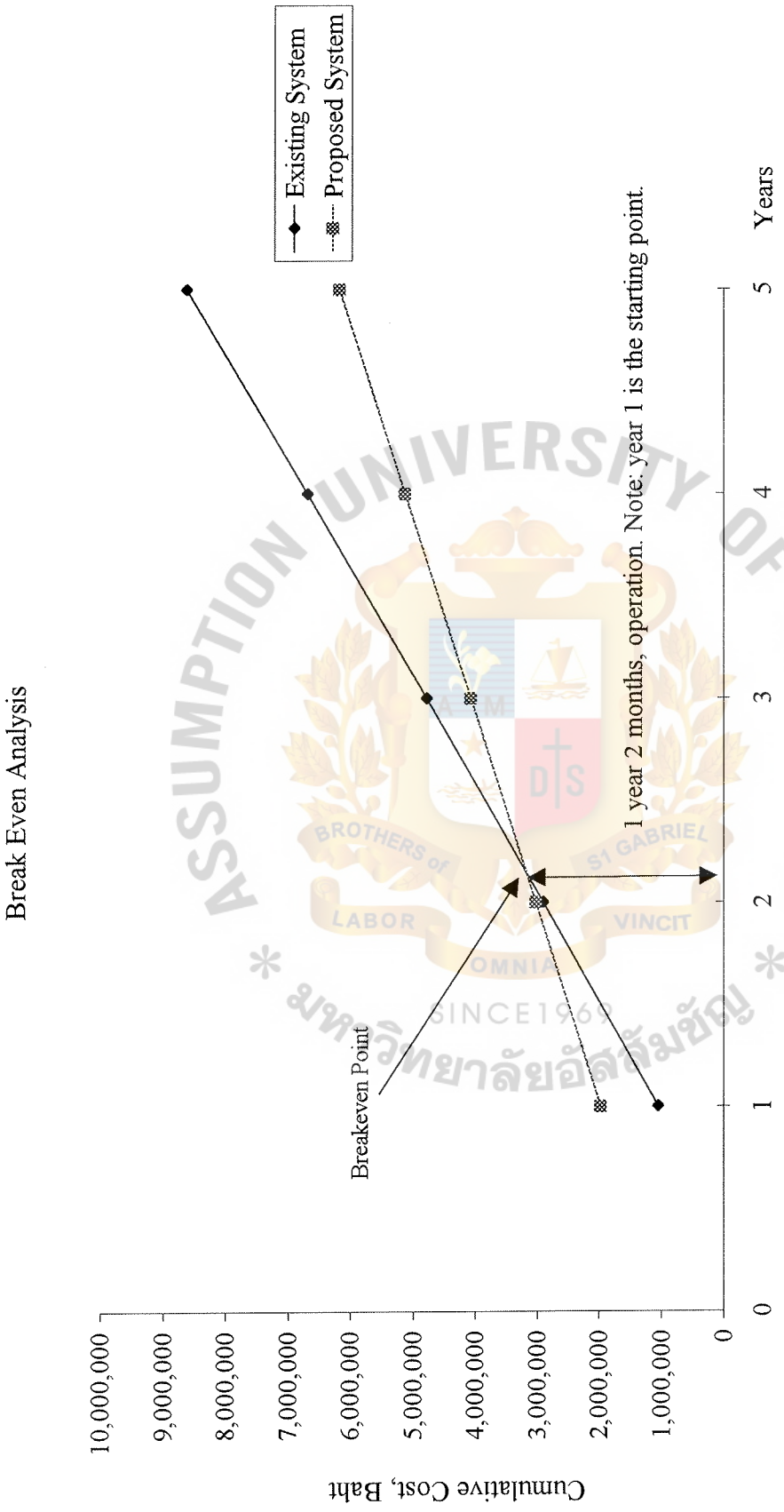

Figure 3.2. Break Even Analysis.

Table 3.3. Payback Analysis of the Proposed System, Baht.

Cost Items	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Development Cost:	926,400.00	0.00	0.00	0.00	0.00	0.00
Operation and Maintenance Cost:	1,037,000.00	1,052,000.00	1,052,000.00	1,052,000.00	1,052,000.00	1,052,000.00
Discount Factors for 12%:	1.000	0.893	0.797	0.712	0.636	0.507
Time-Adjusted Costs (Adjusted to Present Value)	1,963,400.00	939,436.00	838,444.00	749,024.00	669,072.00	533,364.00
Cumulative Time-Adjusted Costs Over Lifetime:	1,963,400.00	2,902,836.00	1,777,880.00	1,587,468.00	1,418,096.00	1,202,436.00

Benefits Derives From Opearation of New Syste	0.00	732,000.00	732,000.00	732,000.00	732,000.00	732,000.00
Discount Factors for 12%	1.00	0.87	0.80	0.71	0.64	0.51
Time-Adjusted Cost (Adjusted to Present Value):	0.00	636,108.00	583,404.00	521,184.00	465,552.00	371,124.00
Cumulative Time-Adjusted Benefits Over Liftime	0.00	636,108.00	1,219,512.00	1,740,696.00	2,206,248.00	2,577,372.00

Cumulative Time-Adjusted Costs & Benefits:	-1,963,400.00	-2,266,728.00	-558,368.00	153,228.00	788,152.00	1,374,936.00

Table 3.4. Net Present Value of the Proposed System, Baht.

Cost Items	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Development Cost:	926,400.00					
Operation and Maintenance Cost:	1,037,000.00	1,052,000.00	1,052,000.00	1,052,000.00	1,052,000.00	1,052,000.00
Discount Factors for 12%:	1.000	0.893	0.797	0.712	0.636	0.507
Present Value of Annual Costs:	1,963,400.00	939,436.00	838,444.00	749,024.00	669,072.00	533,364.00
Total Present Value of Lifetime Costs:						533,364.00

Benefits Derives From Opearation of New System	0.00	732,000.00	732,000.00	732,000.00	732,000.00	
Discount Factors for 12%	1.00	0.87	0.80	0.71	0.64	
Present Value of Annual Benefits	0.00	636,108.00	583,404.00	521,184.00	465,552.00	
Total Present Value of Lifetime Benefits:						2,206,248.00

Net Present Value of this Alternative:						1,672,884.00

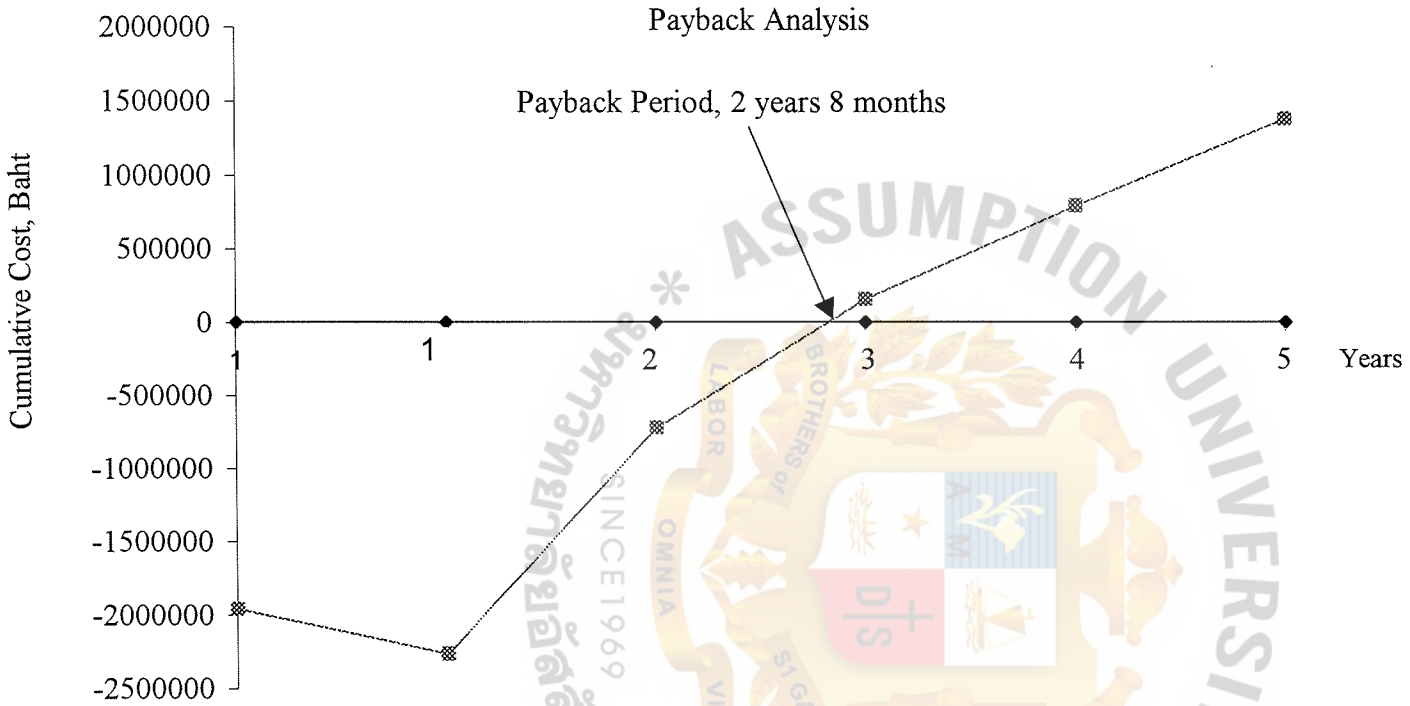

Figure 3.3. Payback Analysis.

IV. PROJECT IMPLEMENTATION

4.1 System Implementation

System implementation is the construction of the new system and the delivery of that system into production (day-to-day operation). The purpose of the system implementation is to build and test a functional system that fulfills business and design requirements and to smoothly convert from the old system to the new system.

The system implementation consists of the many activities and are defined as follows:

4.2 Testing

(a) Network testing

Testing the new computer networks after we install it according to the network design requirement.

This activity can be summarized as follows:

(1) Review the network design requirements outlined in the technical design statement developed during system design.

(2) Construct and then test new networks.

(3) Revise network specification for future reference.

(b) Database Testing

This task must immediately precede other programming activities because database is the resources shared by the computer program to be written.

This activity can be summarized as follows:

(1) Review the technical design statement for database design requirements.

(2) Locate production database that may contain representation data for testing database tables. Otherwise, generate test data for database tables.

(3) Build databases per design specifications.

(4) Load tables with the sample data.

(5) Revise database schema and store as necessary for the future reference.

(c) Program testing

Program testing should be defined after the entire program has been written. This activity can be summarized as follows:

(1) Review the design specifications.

(2) Formulate the project team and assign responsibilities.

(3) Write and document programs and perform unit testing.

(4) Review program document for quality standards.

(5) Programmer is supposed to test each module separately so that he can ensure that all modules are going to function.

(6) The next step is link one module to another module and test them. This testing is implemented to ensure that the modules linked are working as required. While doing this all types of data enter takes place to ensure there is no problem. An invalid data center is also made to check if the program can detect errors or not.

(7) After linking all the modules, the full system is tested. In this testing the users and the operators take part. So the user tests the whole system by putting testing data. The purpose of this test is to ensure the system can be easily operated and well documented. This testing also

covers the user's manual is clear and understood by the operators as well as by the users.

(8) Backup testing is the last testing for the proposed system. After compellation of full system testing and fixing all the bugs the full system is tested with the backup to ensure that the backup procedure is functioning.

(9) Along each step of test the system analysts and the programmers create the checklist of testing to make sure that every step is implemented and debugged before installing the proposed system for the users to use it.

(10) Place the new program and reusable components in the software library.

(d) Training

Converting to a new system necessitates that system users be trained and provided with documentation (user manuals) that guides them through using the new system. Group training (2-3 persons per group) is used because it is better to use our time and the first group of trainees (group of management team) can then train several other groups.

This activity can be summarized as follows:

(1) Collect documentation that may prove useful in developing user documentation and training guides.

(2) Write user documentation manuals that is easy to understand.

(3) Review the training needs of the system users.

(4) Schedule training sessions.

(5) Conduct training sessions and distribute user documentation.

4.3 Installing the System

Since the existing system is a manual system there are a few steps for installing the system. The installation is going to follow the parallel conversion method. In parallel conversion method, both old and new system are operated for the same time. This is done to ensure that all major problems in the new system have been solved before the old system is discarded. Parallel conversion minimizes the risk of major flows in the new system causing irreparable harm to the business. Parallel conversion is suitable for the change from the manual system to the computerized system, although it increases the cost of running two systems over some period and consumes more time with double workload of employees. When the employees can run the new system smoothly and all major problems can be solved, the double workloads will be reduced.

During the installation period, the system analyst and programmer have to be there to make sure that they follow the following steps:

(1) A list of all the file and document that are to be installed should be made which will be used during the installation.

(2) All necessary data must be prepared to ensure that all the records will be entered or keyed into the system. All these records should be grouped into at least of 50 records each so that they can be easy to check easily if while inputting the data some record is missed.

(3) The analyst is supposed to check the customer site to ensure that all the preparation is finished. This includes checking that all electrical lines are properly put, air conditioning is there and there is a control for controlling humidity. There should be enough space for the equipment as proposed in the proposed system.

(4) During the installing stage the analyst and programmer try not disturb the current system. And after the installation is finished with the proposed system the programmer performs the training task. The training is given in house.

V. CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

This project comprises on the benefits gained by providing a computerized system to replace existing manual system. Such a computerization would generate better data management for the leasing system. In the existing system, the user has to use the handwriting to make contact to other department and this requires filling of form and typing reports. This is a tedious and time-consuming process. Many times the problem occurred because of too much time spent in operation, or error from staff's operation. Moreover, other department needs to get some data from existing system and for them it is difficult to get the information on time since the details have been written down. When information is written then the need is to input the data in the computerized system.

The proposed system has great advantages over the existing system. All problems that were faced by the existing system are eliminated such as security of the information, management of the customer history and so on. There are other valuable advantages provided by the proposed system. The proposed system can assist the top management in making decision as they can get the information as soon as they need.

The proposed system can assist the top management in making decision as they can get the information as soon as they need. They can get any type of information they would like to get. All the benefits, which the proposed system provides to the users, promises it can improve the efficiency of data management.

The table below compares the benefit of proposed system over existing system.

Table 5.1 Comparison between Proposed System and Existing System.

Items	Existing System	Proposed System
1. Management	Slow management due to lack of well design process.	Better data management in term of time-consuming process.
2. Security	Data is kept in record and unauthorized person can access it easily.	Using password protection. Only authorized person can access the data.
3. Data inconsistency	Data is duplicate due to many persons write the same data.	Only responsible person can write the data to database.
4. Waste storage space	Existing system is the paper work. It needs space to keep the data.	Data is kept in the file format. No need for storage space.
5. Work and time	Due to all of these problems, it needs to recheck everything again. So more times required.	It improves speed and time to mange the data. Reduce double check work

The reason why speed to perform the proposed system is faster than existing system is shown in table 5.2

Table 5.2. Speed Comparison between Existing System and Proposed System.

Process	Existing System	Proposed System
Making Contract Process	Manual process and lack of good management	Computer-based technology with well-train staff.
Payment Process	Slow processing because spent more time to fine the customer record	Fast processing because using computer to search the required record which process faster than manual system.
Closed Account Process	Very slow because lack of good record management.	Fast because record kept in order and easy to search

5.2 Recommendations

The proposed system promised an efficient and effective way of carrying on the activities related to the organization. The recommendations to be made for the organization are as follows:

(1) Improve the skills of the staff by providing good training facilities regularly.

(2) Investment plan for the future in terms of hardware and software requirements

(3) Upgrade and expansion in future as per requirement

(4) Regular maintenance of the new system

(5) Frequent reviews to keep up to date with growing user requirements.

(6) Prepare required reports as often as possible.

(7) Consider computerizing the remaining department.

The system should be developed to cover other system of the company. System in other department of the company should be further developed to the computer-based system. Further system would help the company to reduce much of the operation costs in the long run. Anyway the leasing system we developed here is considered to be part

of the future system. The expansion of the development will function without double-checking some works in the leasing system.

System maintenance facilities should be provided to the concerned authorities, which will enable the setting of the password with a view of providing security to the system from lacking the information. The proposed system should be reviewed properly every six months. Backup should be kept of everyday work and of a week also. It is also recommended that the organization should immediately replace the existing system with the proposed system.

APPENDIX A

DATA FLOW DIAGRAM

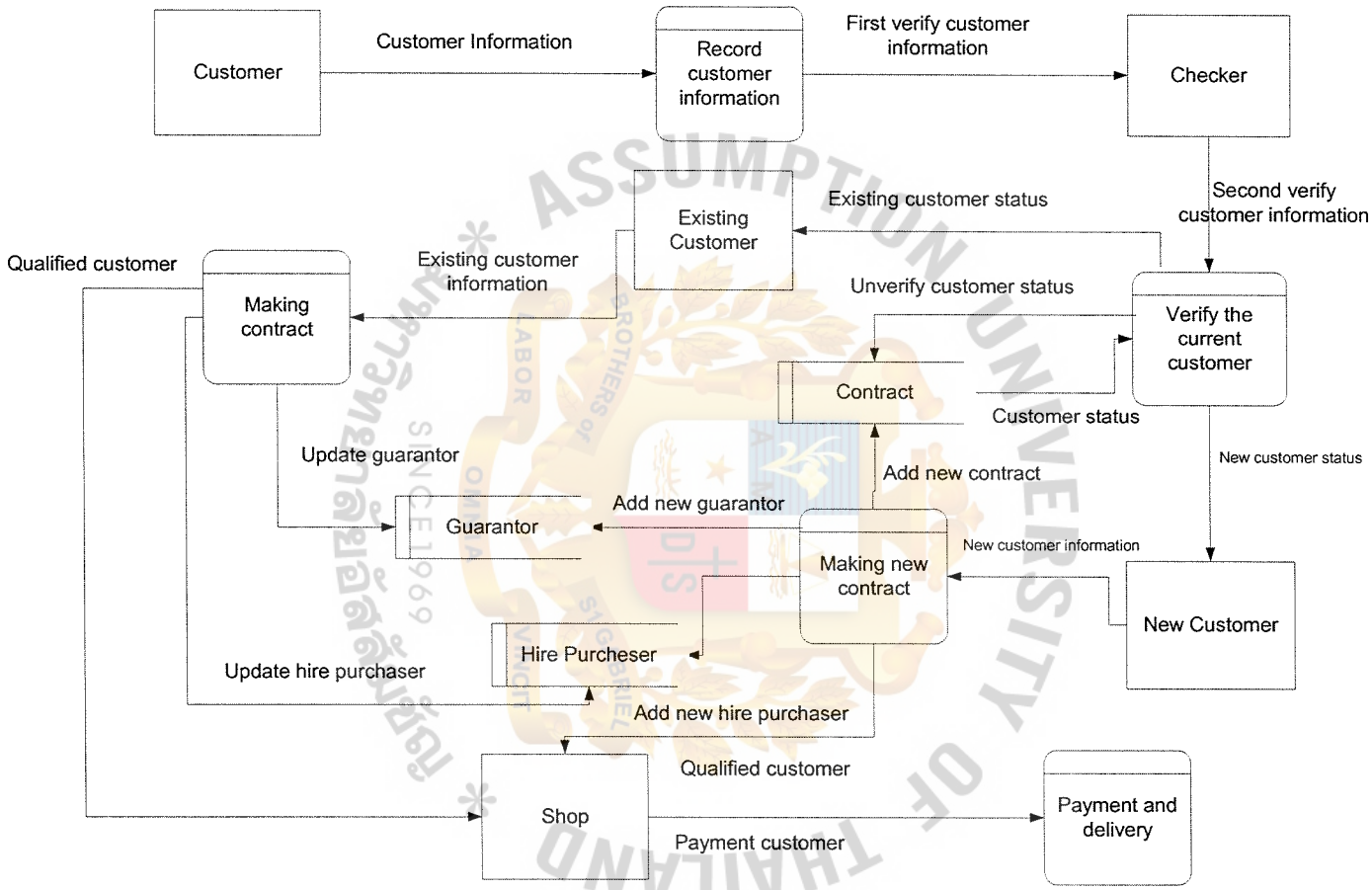

Figure A.1. DFD Making Contract Process Level 1.

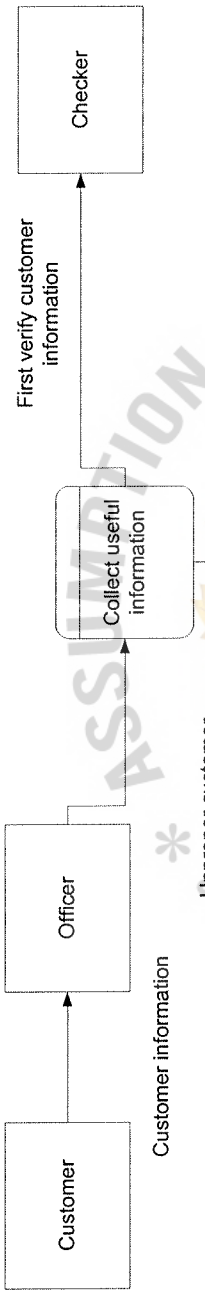

Figure A.2. DFD Making Contract Process Level 2. Record Customer Information.

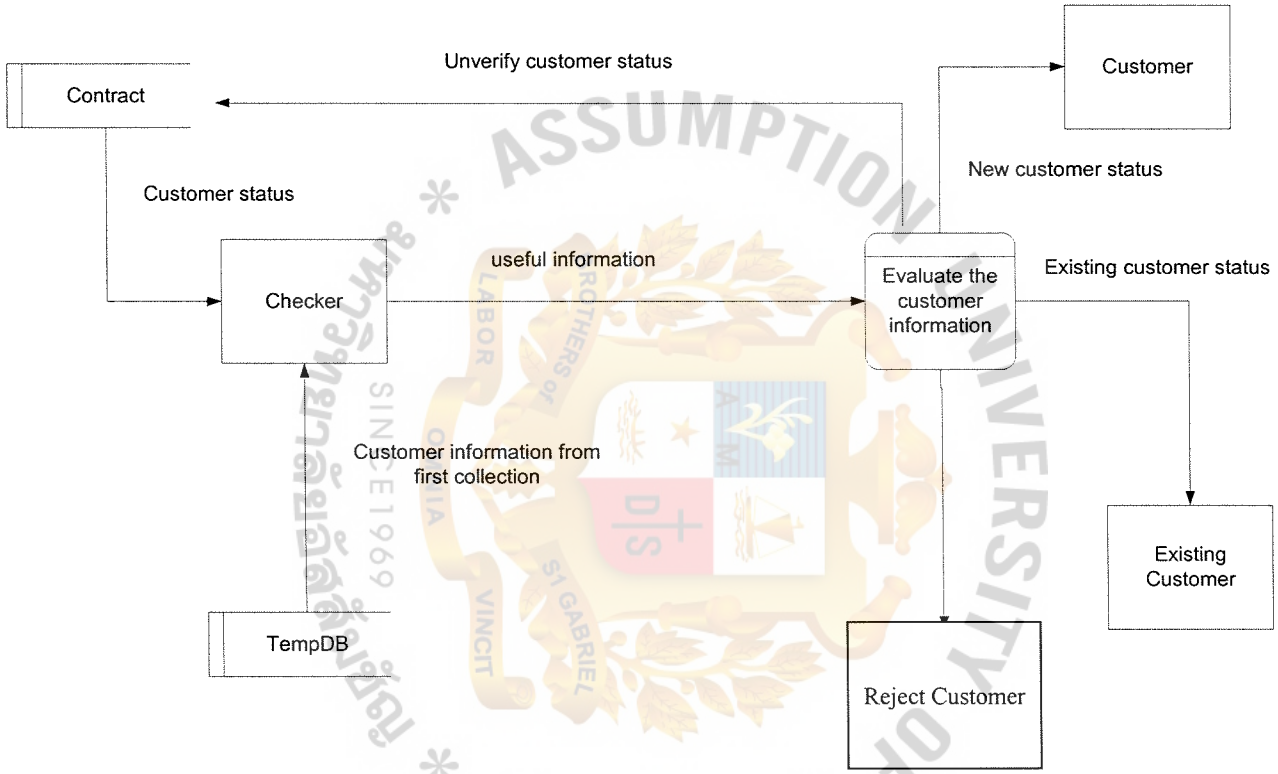

Figure A.3. DFD Making Contract Process Level 2.Verify the Current Customer.

Figure A.4. DFD Payment Process Level 1.

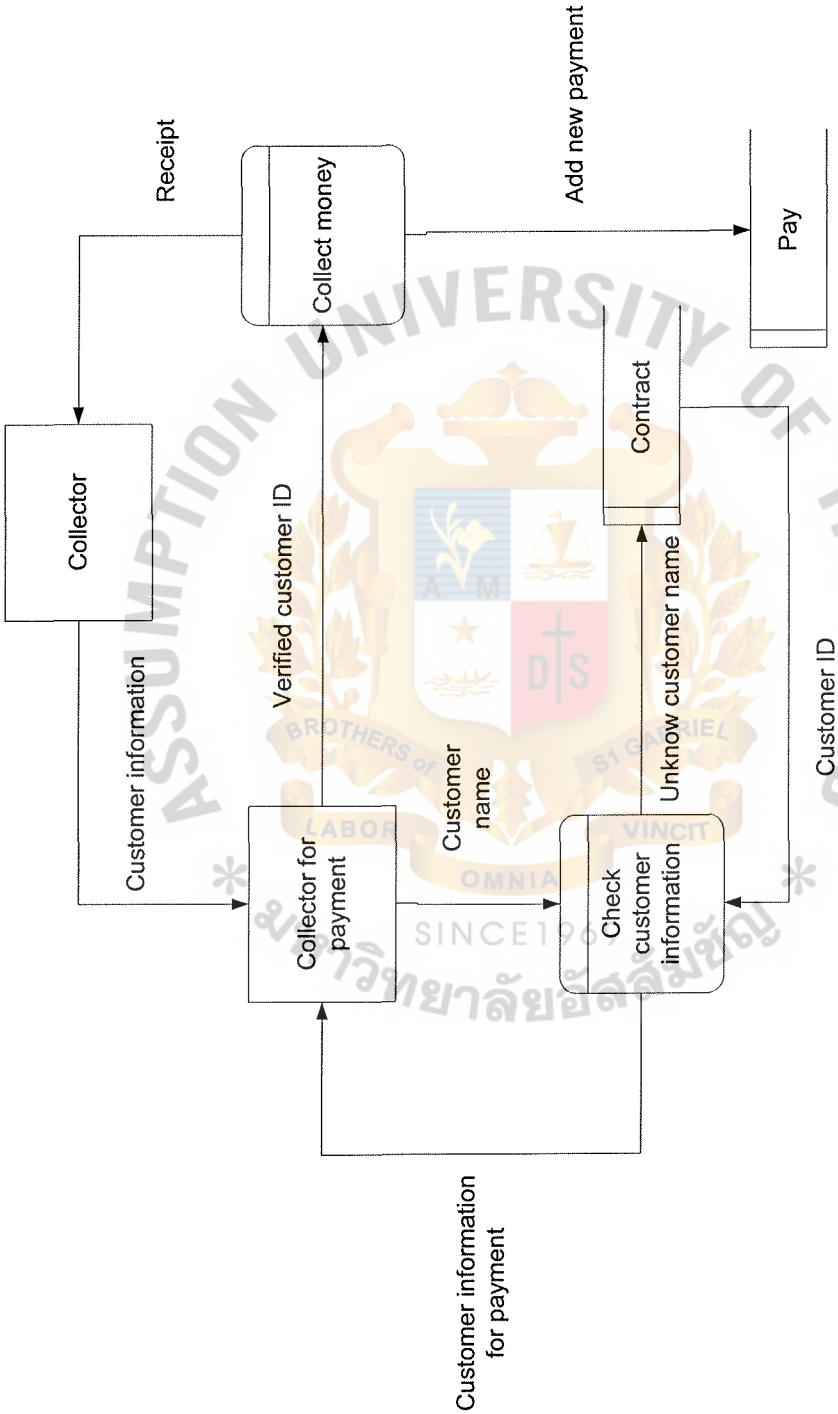

Figure A.5. DFD Payment Process Level 2. Pay the Money.

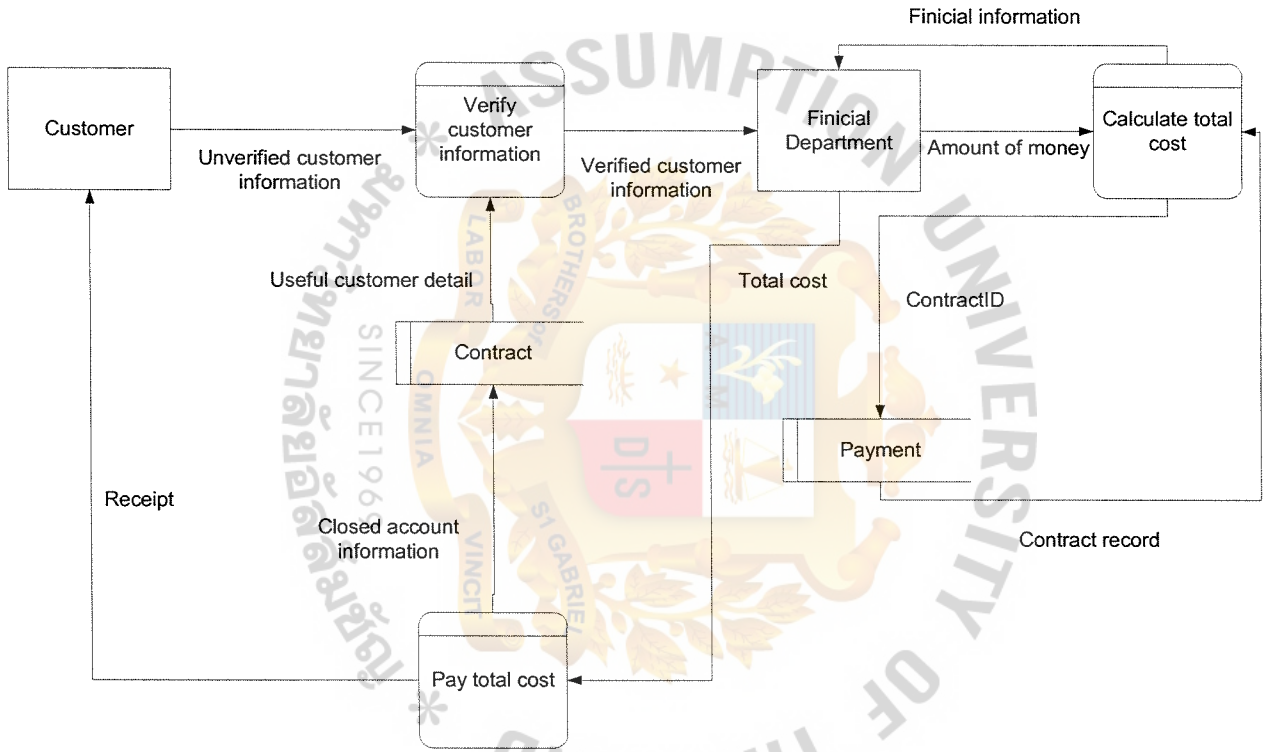

Figure A.6. DFD Closed Account Process Level 1.

APPENDIX B

DATA DICTIONARY

DATA DICTIONARY

Table B.1. Data Dictionary of Leasing System Data Flow Diagram.

Field Name	Meaning
Add new contract	The flow of the date in order to add new customer.
Add new guarantor	The flow of the data that use for add new guarantor.
Add new hire purchaser	The data flow to add hire purchaser.
Add payment	The data flow to add the payment.
Check customer information	The process that check and verify the customer information
Checker	The person who check the status of the customer.
Collect money	The process that keeps the money and stores the information about the payment.
Collect useful information	The process that collect the useful information for making the contract.
Collector for payment	The person who collect the money from the customer.
Customer information	The customer data flow.
Customer information for the next process	The flow of customer information to the next process.
Customer information for payment	The data flow of the customer information in order to payment.

Table B.1. Data Dictionary of Leasing System Data Flow Diagram.(Continue)

Field name	Meaning
Customer information from first collection	The first data flow in first collection.
Customer name	The flow of the customer name.
Customer status	The customer status data flow.
Evaluate the customer information	The process that evaluate and verify the status of the customer and permit customer to loan the money.
Existing customer	The person who has been the old customer before and want to lease motorcycle again.
Existing customer information	The flow of the existing customer information.
Existing customer status	The flow of the existing customer status.
First verify customer information	The data flow of the customer information to verify for the first time.
Guarantor	The database that collect the data about the guarantor.
Hire Purchaser	The data store that collect the information of the hire purchaser.
Making contract	The process that make the contract in order to lease the motorcycle.
Making new contract	The process that making the contract for the new customer.

Table B.1. Data Dictionary of Leasing System Data Flow Diagram.(Continue)

Field name	Meaning
New customer	The customer who never leased with our company before and want to lease the motorcycle.
New customer information	The new customer information data flow.
New customer status	The flow of the new customer status.
Officer	The person who is responsible for collecting the information of the customer.
Pay	The database that keep all payment of the contract.
Pay the money	The process that the customer pay the money and give the receipt back to the customer.
Payment customer	The data flow of the information about the payment of the customer.
Qualified customer	The flow of the qualified customer.
Receipt	The process to store the customer information into the data store.
Reject customer	The person who unqualified status because of any reasons.
Reject status	The reject status data flow.
Second verify customer information	The second data flow for the customer information in order to verify.
Shop	The place to sell and buy the motorcycle.
Unknown customer name	The flow of the unknown customer.

Table B.1. Data Dictionary of Leasing System Data Flow Diagram.(Continue)

Field name	Meaning
Improper customer name	The flow of the improper information.
Unverified customer status	The data flow of the unverified customer status.
Upgrade guarantor	The upgrade guarantor data flow.
Upgrade hire purchaser	The data flow for update hire purchaser.
Verify the current customer	The process that verify the current customer in order to reject or accept the customer.

APPENDIX C

PROCESS SPECIFICATION

Verify the Current Customer

(1) Module intention

(a) Function of the module

(1) Get the customer information

(2) Filter these information in order to verify the credit of its customer

(3) Verify the status of the current customer

(4) Check the history record of the current customer

(5) First step to verify the status of the current customer

(b) Input and output of the module

(1) Input

(a) The information about the customer.

(b) Useful information of the customer

(c) Customer Status

(2) Output

(a) New customer status

(b) Unverified customer status

(c) Existing customer status

(d) Reject customer status

(c) Constraint / limitation

(a) Some data are not correct

(b) Some data are missing

(c) Waste some time to collect all information

(2) Module implementation – Pseudocode

Begin

(a) Find the customer name

(b) Check the status of the customer by using their name or resident ID

(c) Return the customer status

(d) Send their status to the next process

End

(3) Test Plan

(a) Test criteria

(1) Check the data input

(2) Check the quality the customer information

(b) Test Data

(1) To check the status of the current customer by using their name or resident ID

(c) Expect Output

(1) The status of the current customer

Making New Contract

(1) Module intention

(a) Function of the module

(1) Get the correct information to make the new contract

(2) Verify all of these information

(3) Check the status of the customer

(4) Draw the address of the customer

(b) Input and Output of the module

(1) Input

(a) The customer information

(b) Date created

(c) New customer status

(d) Others

(2) Output

(a) Contract ID

(b) Customer ID

(c) Customer status

(d) The detail of the contract

(c) Constraint / Limitation

(1) No recursive occur

(2) Some information are missing or not correct

(3) Cannot check the correctness of the information

(2) Module Implementation – Pseudocode

Begin

(a) Get the customer information

(b) Create the contractID, customerID

(c) Verify the correctness of the customer information

(d) Check the customer information

(e) Input the data into the Database

(f) Print monthly report

End

(3) Test Plan

(a) Test Criteria

(1) Check the data input

(2) Unable to wait for some information

(3) Verify the correctness of the information

(b) Test Data

(1) Get the correct and useful information and put it to the database

(c) Expected Output

(1) The quality customer

(2) Contract report

(3) Customer report

(4) Contract detail

Making Contract

(1) Module intention

(a) Function of the module

(1) Get the correct information to make the contract

(2) Verify all of these old information

(3) Check the status of the customer

(4) Draw the address of the customer

(5) Update the information of the existing customer

(b) Input and Output of the module

(1) Input

(a) The customer information

(b) Date created

(c) Existing customer status

(d) Others

(2) Output

(a) Contract ID

(b) Customer status

(c) The detail of the contract

(c) Constraint / Limitation

(1) No recursive occur

(2) Some information are missing or not correct

(3) Cannot check the correctness of the information

(2) Module Implementation – Pseuducode

Begin

(a) Get the customer information

(b) Create the contractID, customerID

(c) Verify the correctness of the customer information

(d) Check the customer information

(e) Input the data into the Database

(f) Print monthly report

End

(3) Test Plan

(a) Test Criteria

(1) Check the data input

(2) Unable to wait for some information

(3) Verify the correctness of the information

(b) Test Data

(1) Get the correct and useful information and put it to the database

(c) Expected Output

(1) The quality customer

(2) Contract report

(3) Customer report

(4) Contract detail

Check Customer Information

(1) Module intention

(a) Function of the module

(1) Find the necessary information of the customer for payment

(b) Input and output of the module

(1) Input

(a) The customer name

(2) Output

(a) Customer ID

(b) Contract ID

(c) Constraint / Limitation

(1) Unable to find customer name

(2) Module implementation – Pseudocode

Begin

(a) Key the customer name

(b) Find the customer ID

(c) Find the contract ID

(d) Use these information to next process

End

(3) Test plan

(a) Test Criteria

(1) Check the customer name

(b) Test Data

(1) Able to get the correct customer ID and related information

(c) Expected Output

(1) Customer ID

(2) Contract ID

Collect Money

(1) Module intention

(a) Function of the module

(1) Collect the information about payment of each customer

(2) Verify the payment information of each customer

(b) Input and output of the module

(1) Input

(a) Customer name

(b) Customer ID

(c) Contract ID

(d) Amount of payment

(e) Payment Date

(2) Output

(a) Receipt

(c) Constraint / Limitation

(1) Slow process when don't know the customer name

(2) Module implementation – Pseuducode

Begin

(a) Key the customer name

(b) Find the customer ID

(c) Find the contract ID

(d) Find the contract and customer information

(e) Input the amount of payment

(f) Get the receipt from the system

End

(3) Test plan

(a) Test Criteria

(1) Check the customer name

(b) Test Data

(1) Able to get the correct customer ID and related information

(c) Expected Output

(1) Receipt

Calculate Total Cost

(1) Module intention

(a) Function of the module

(1) Calculate the cost of each customer in order to closed their account

(2) Verify the correctness of the history record of each payment of each customer

(b) Input and output of the module

(1) Input

(a) Customer name

(b) Customer ID

(c) Contract ID

(d) Amount of payment

(e) Receipt

(f) Closed Date

(2) Output

(a) Interested

(b) Cost

(c) Total payment

(c) Constraint / Limitation

(1) Some data are missing when it is not stored properly.

(2) Module implementation – Pseuducode

Begin

(a) Key the customer name

(b) Find the customer ID

(c) Find the contract ID

(d) Find the contract and customer information

(e) Input the amount of payment

(f) Collect the receipt number

(g) Look up the history record of customer

(h) Calculate the interest

(i) Calculate the total cost

(j) Payment of the customer

(k) Update the payment database

End

(3) Test plan

(a) Test Criteria

(1) Check the customer name

(2) Check the receipt

(3) Check the history record

(4) Check the contract information

(b) Test Data

(1) Able to get the correct customer ID and related information

(2) Able to calculate the correct amount of each payment

(c) Expected Output

(1) Total cost

Pay Total Cost

(1) Module intention

(a) Function of the module

(1) Collect the total amount of payment of each customer

(2) Verify the payment information of each customer

(3) Issue the receipt and closed account of the customer

(b) Input and output of the module

(1) Input

(a) Customer name

(b) Customer ID

(c) Contract ID

(d) Amount of payment

(e) Payment Date

(2) Output

(a) Receipt

(c) Constraint / Limitation

(1) Physical limitation (Customer does not have money to pay)

(2) Module implementation – Pseuducode

Begin

(a) Key the customer name

(b) Find the customer ID

(c) Find the contract ID

(d) Find the contract and customer information

(e) Put the Total amount of payment

(f) Update the contract database

(g) Closed the account of the customer

(h) Issue the receipt from the system

End

(3) Test plan

(a) Test Criteria

(1) Check the customer name

(b) Test Data

(1) Able to get the correct customer ID and related information

(c) Expected Output

(1) Receipt

APPENDIX D

ENTITY RELATIONSHIP

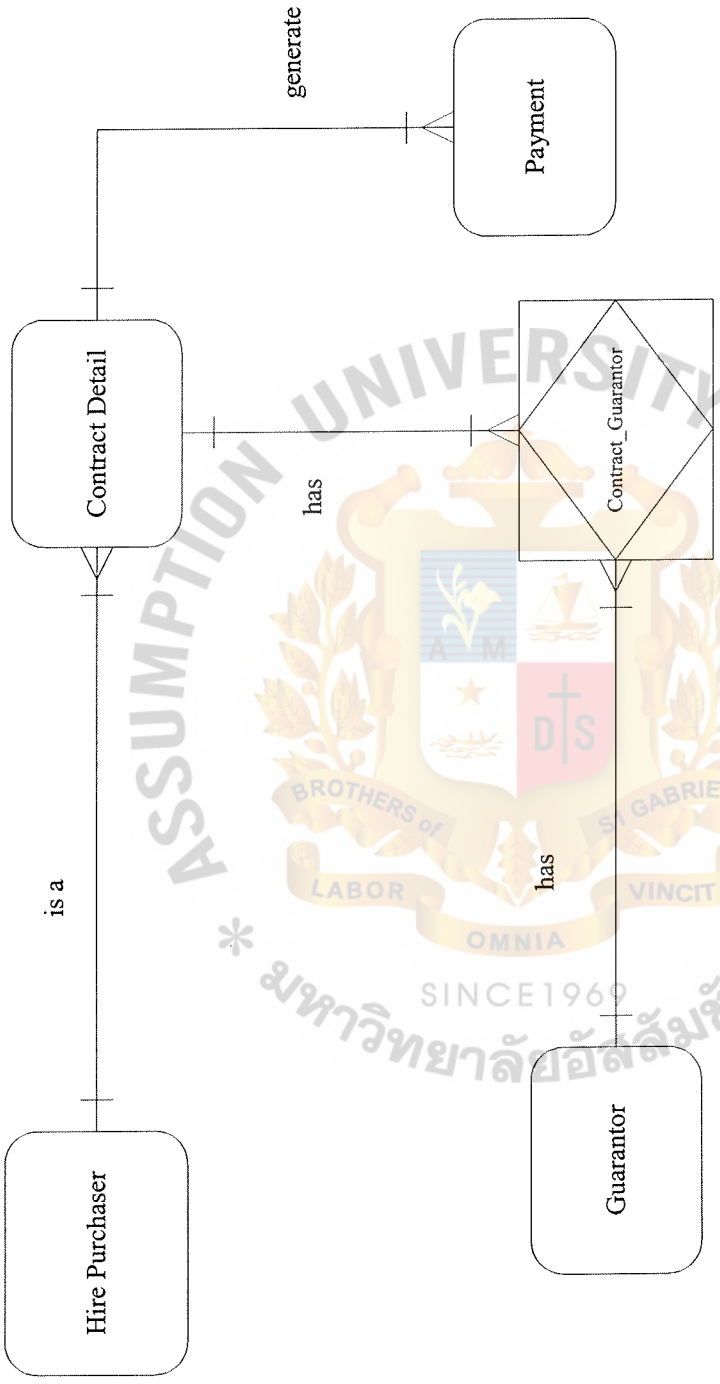

Figure D.1. Entity Relationship Leasing System Context Diagram.

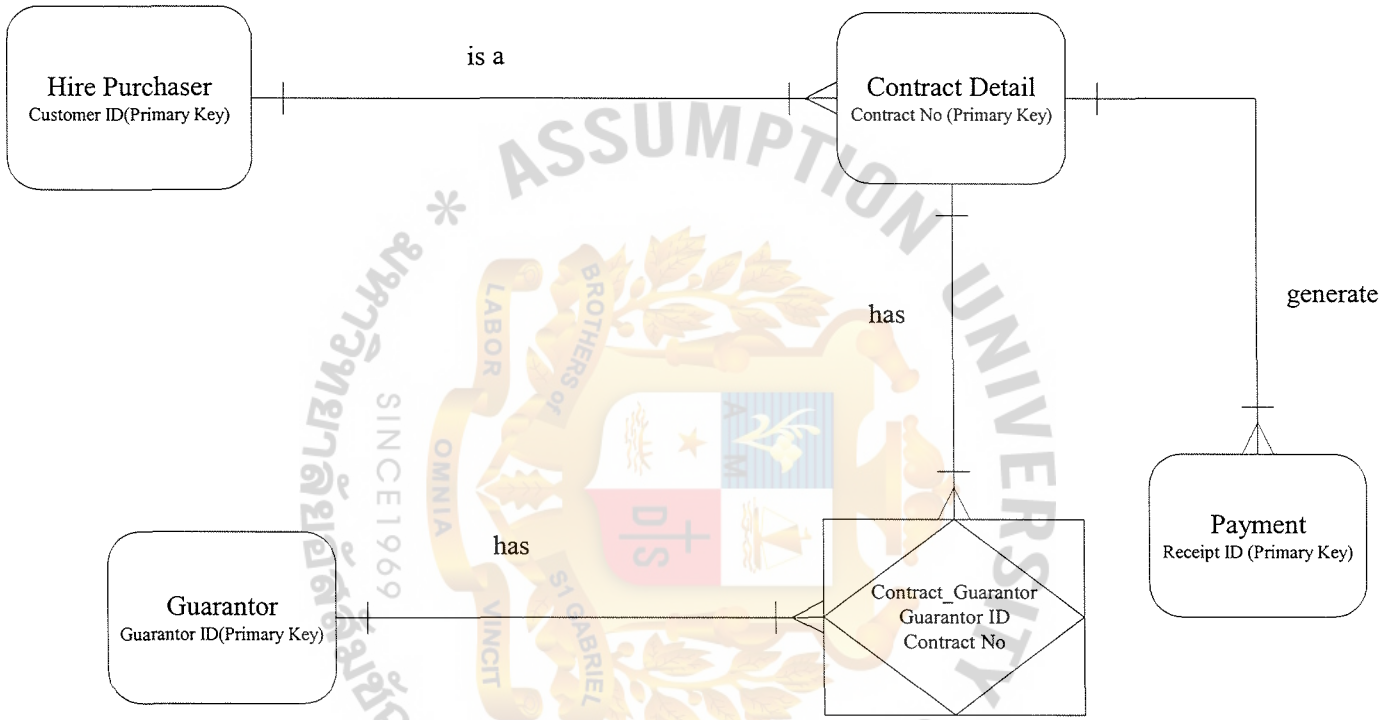

Figure D.2. Entity Relationship Leasing System Key-Based Data Model.

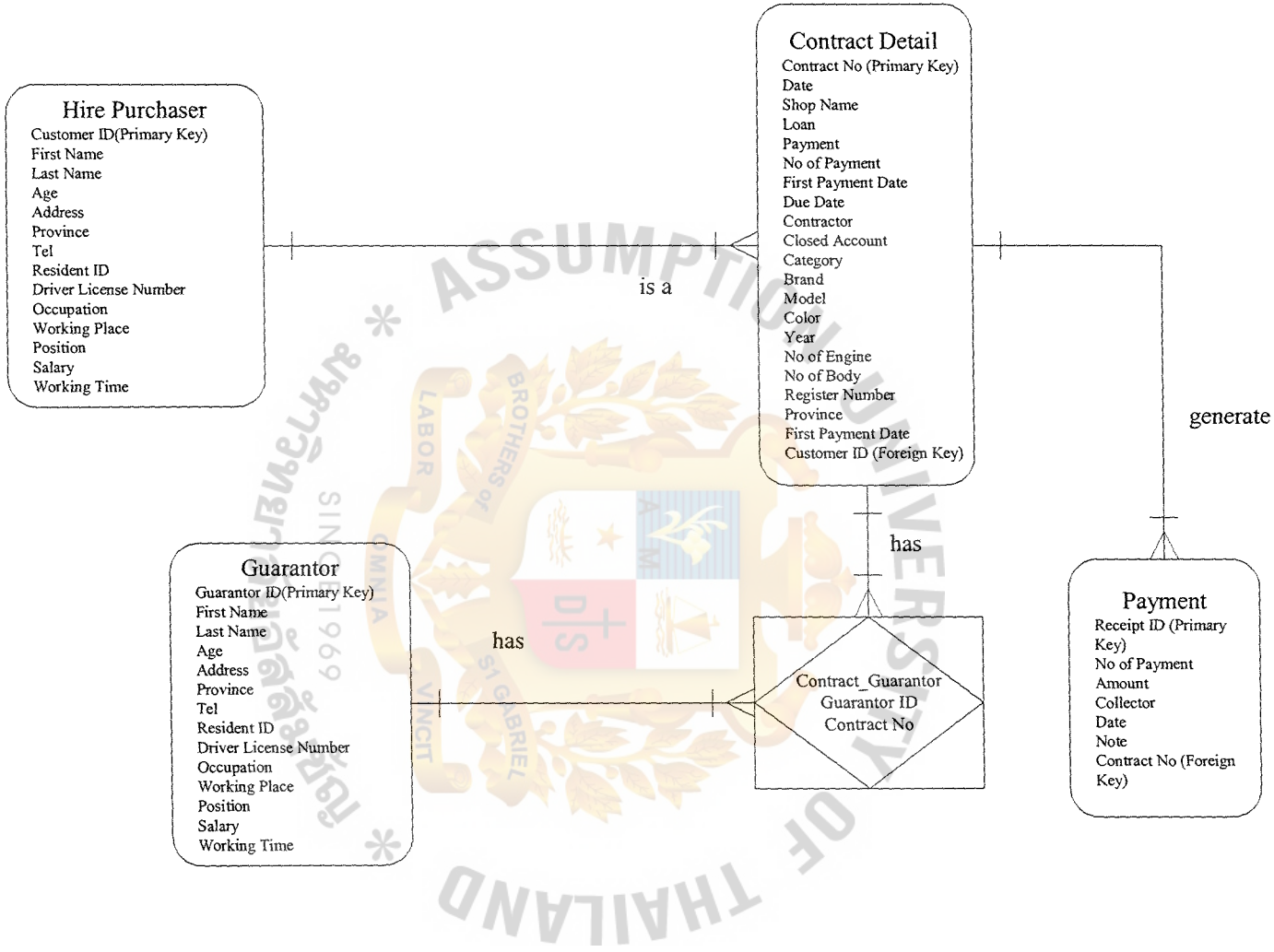

Figure D.3. Entity Relationship Leasing System Fully Attribute Data Model.

Table D.1. File Structure of Hire Purchase Table.

Sequence	Field Name	Data Type	Length	Decimal Place	Description	Key
1	Customer ID	Numeric	5	-	The unique number for customer	Primary Key
2	Firstname	Character	15	-	First name of hire purchaser	Attribute
3	Lastname	Character	15	-	Last name of hire purchaser	Attribute
4	Age	Numeric	5	-	Age of hire purchaser	Attribute
5	Address	Character	50	-	Address of hire purchaser	Attribute
6	Province	Character	15	-	Province of hire purchaser	Attribute
7	Tel	Numeric	10	-	Telephone of hire purchaser	Attribute
8	Resident ID	Numeric	15	-	Resident number of hire purchaser	Attribute
9	Driver License Number	Numeric	15	-	Driver license number	Attribute
10	Occupation	Character	15	-	Occupatio of hire purchaser	Attribute
11	Working Place	Character	50	-	Working place of hire purchaser	Attribute
12	Position	Character	15	-	Position of hire purchaser	Attribute
13	Salary	Numeric	8	2	Salary of hire purchaser	Attribute
14	Working Time	Date	8	-	Working time of hire purchaser	Attribute

Table D.2. File Structure of Guarantor Table.

Sequence	Field Name	Data Type	Length	Decimal Place	Description	Key
1	Customer ID	Numeric	5	-	The unique number guarantor	Primary Key
2	Firstname	Character	15	-	First name of guarantor	Attribute
3	Lastname	Character	15	-	Last name of guarantor	Attribute
4	Age	Numeric	5	-	Age of guarantor	Attribute
5	Address	Character	50	-	Address of guarantor	Attribute
6	Province	Character	15	-	Province of guarantor	Attribute
7	Tel	Numeric	10	-	Telephone of guarantor	Attribute
8	Resident ID	Numeric	15	-	Resident number of guarantor	Attribute
9	Driver License Number	Numeric	15	-	Driver license number of guarantor	Attribute
10	Occupation	Character	15	-	Occupatio of guarantor	Attribute
11	Working Place	Character	50	-	Working place of guarantor	Attribute
12	Position	Character	15	-	Position of guarantor	Attribute
13	Salary	Numeric	8	2	Salary of guarantor	Attribute
14	Working Time	Date	8	-	Working time of guarantor	Attribute

Table D.3. File Structure of Payment Table.

Sequence	Field Name	Data Type	Length	Decimal Place	Description	Key
1	ReceiptID	Numeric	8	-	The unique number to identify payment	Primary Key
2	No of Payment	Numeric	8	-	Number of time of each payment	Attribute
3	Amount	Numeric	8	-	The amount of money of each payment	Attribute
4	Collector	Character	15	-	The person who keeps the money	Attribute
5	Note	Character	100	-	The message to notify or remind.	Attribute
6	Date	Date	8	-	Receipt date	Attribute
7	Contract No	Numeric	8	-	The unique number to identify contract	Foreign Key

Table D.4. File Structure of Contract_Guarantor Table.

Sequence	Field Name	Data Type	Length	Decimal Place	Description	Key
1	Guarantor ID	Numeric	8	-	The unique number to identify guarantor	Foreign Key
2	Contract No	Numeric	8	-	The unique number to identify contract	Foreign Key

Table D.5. File Structure of Hire Purchase Table.

Sequence	Field Name	Data Type	Length	Decimal Place	Description	Key
1	Contract No	Numeric	8	-	The unique number to identify each contract.	Primary Key
2	Date	Date	8	-	Date to make the contract	Attribute
3	Shop Name	Character	15	-	Name of shop to make contract	Attribute
4	Loan	Numeric	8	-	Amount of loan in this contract	Attribute
5	Payment	Numeric	8	-	Amount of each payment in the contract	Attribute
6	No of Payment	Numeric	8	-	Number of payment in this loan	Attribute
7	First Payment Da	Date	8	-	First date to payment	Attribute
8	Due Date	Date	8	-	Date to payment each month	Attribute
9	Contractor	Character	15	-	The name of person who verify the status	Attribute
10	Closed Account	Character	15	-	Closed account status	Attribute
11	Brand	Character	15	-	Motorcycle brand	Attribute
12	Model	Character	15	-	Motorcycle model	Attribute
13	Color	Character	15	-	Motorcycle color	Attribute
14	Year	Character	15	-	Year to produce this motorcycle	Attribute
15	No of engine	Character	15	-	Engine number	Attribute
16	No of Body	Character	15	-	Body number	Attribute
17	Register number	Character	15	-	The register number of the motorcycle	Attribute
18	Province	Character	15	-	The province of the motorcycle	Attribute
19	Customer ID	Numeric	8	-	The unique number to identify customer	Foreign Key

APPENDIX E

INPUT AND OUTPUT DESIGN

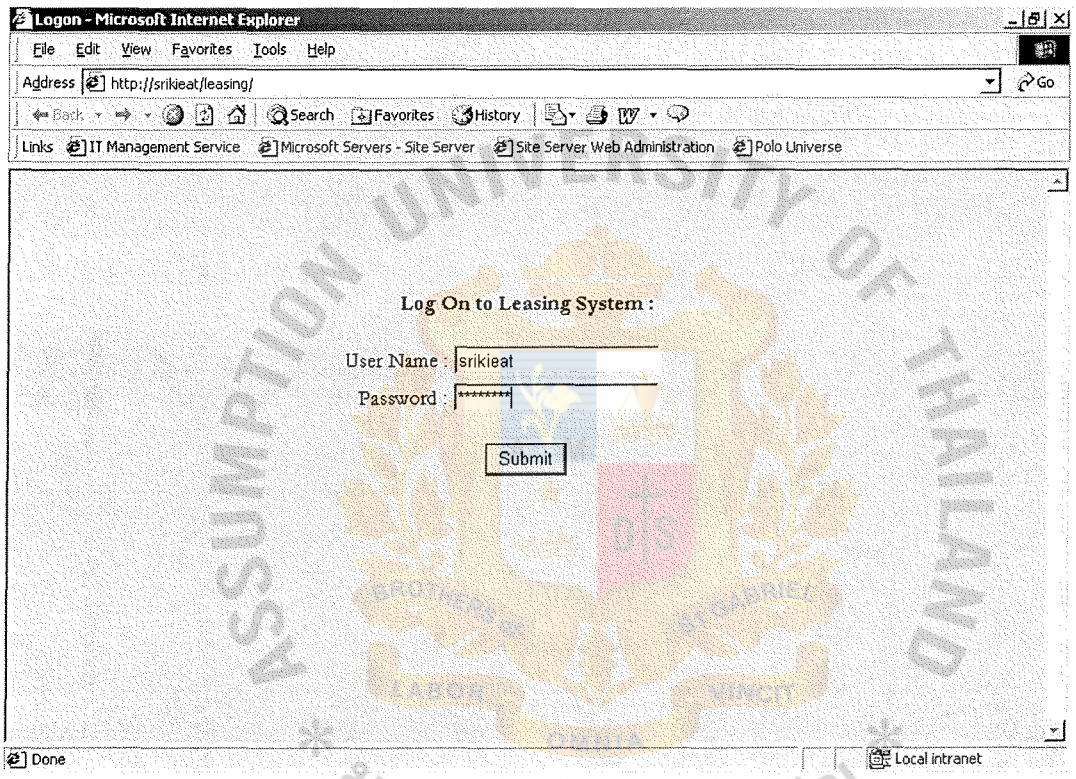

Figure E.1. Login.

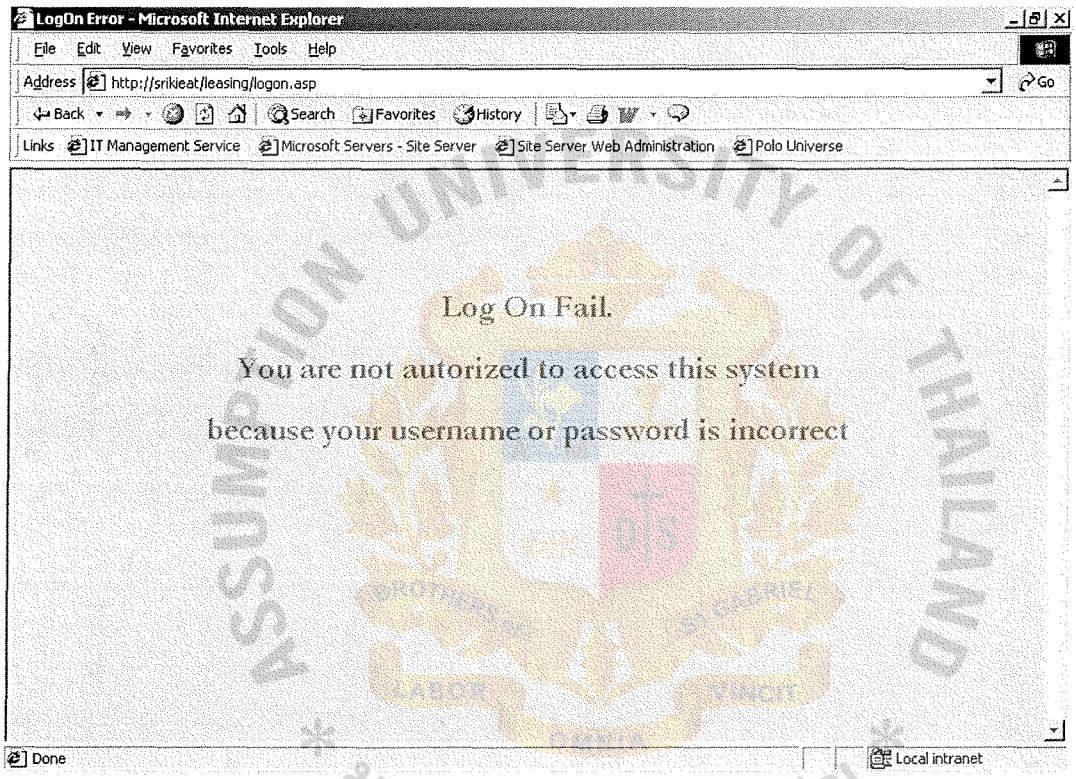

Figure E.2. Login Fail.

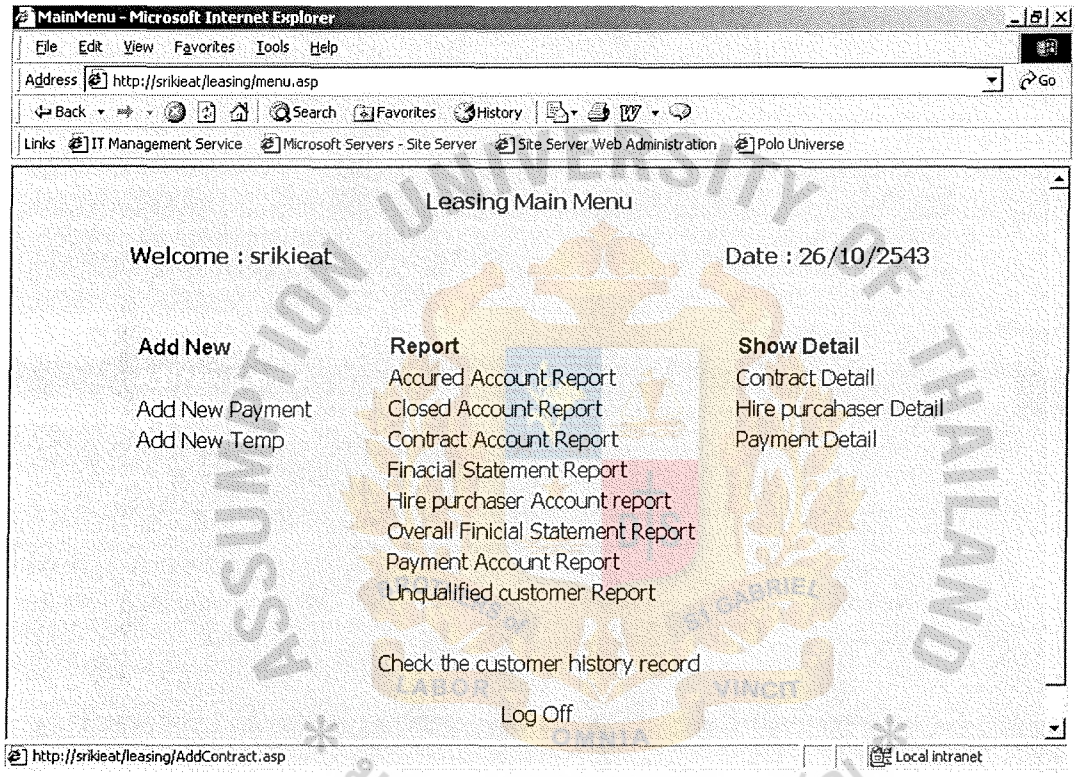

Figure E.3. Main Menu.

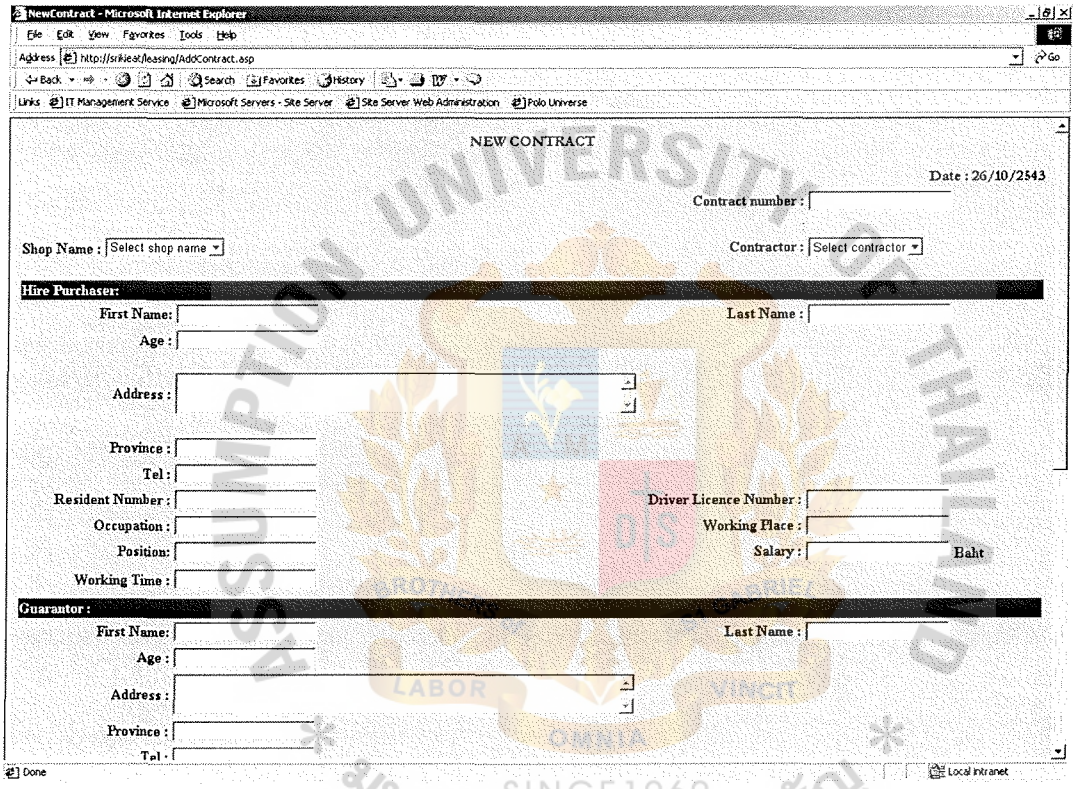

Figure E.4. Add New Contract 1.

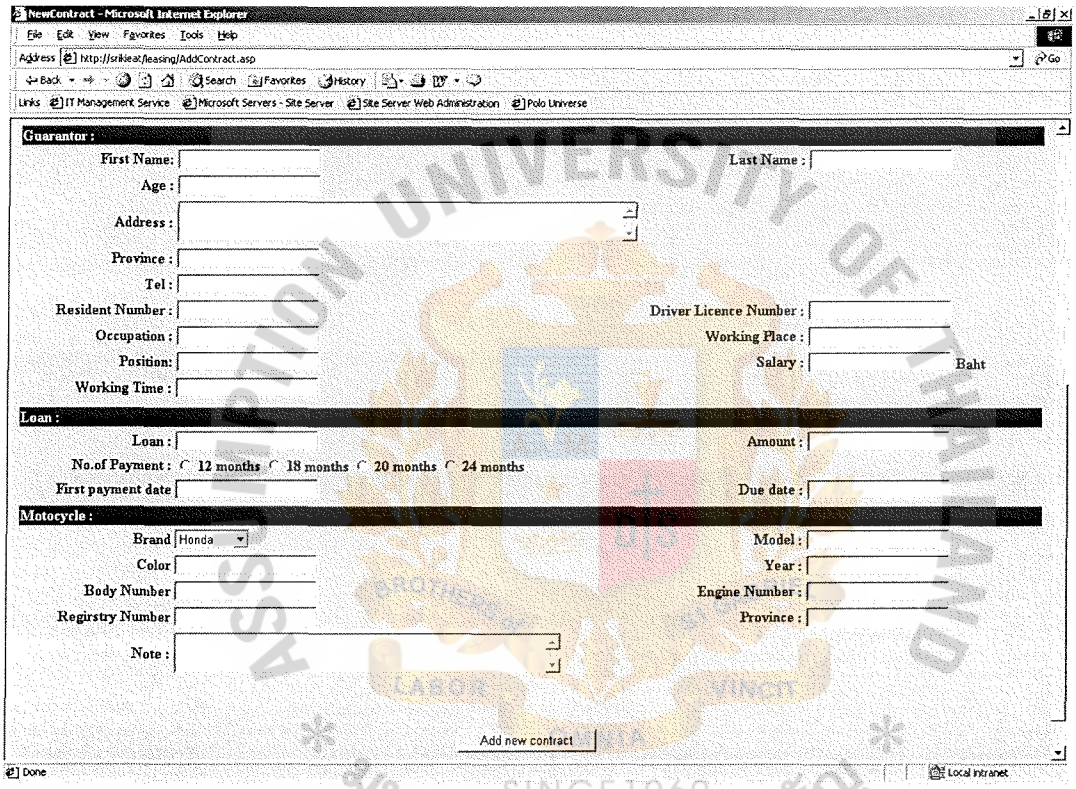

Figure E.5. Add New Contract 2.

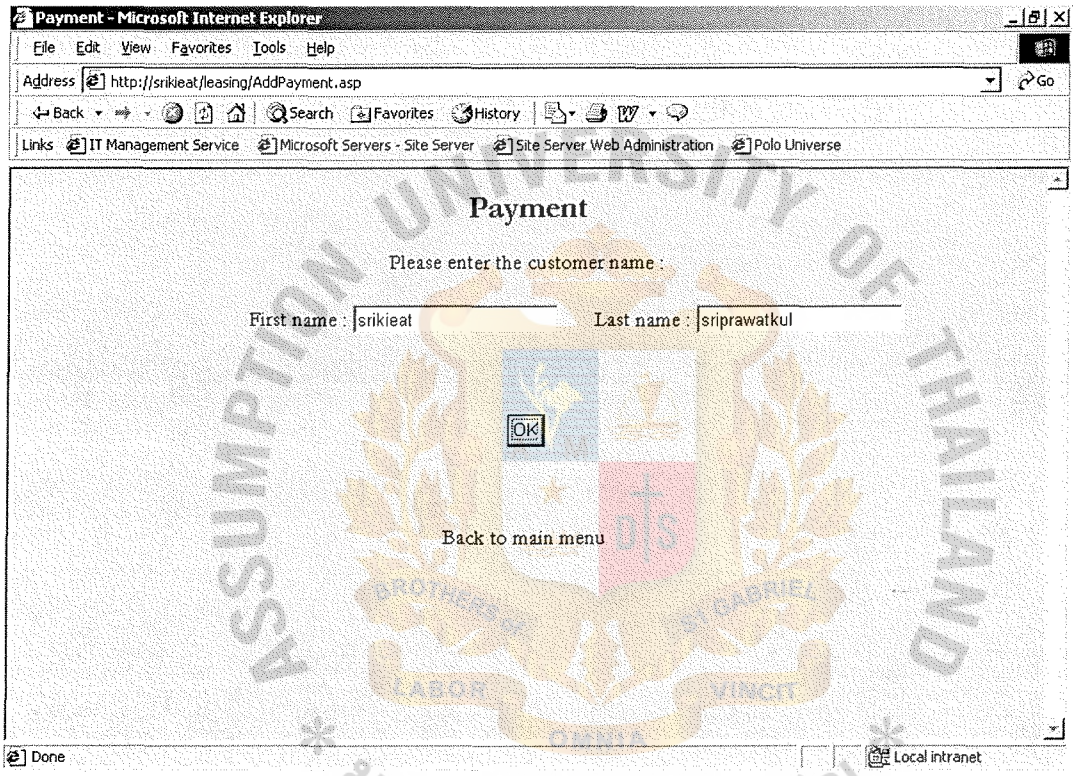

Figure E.6. Add New Payment Search.

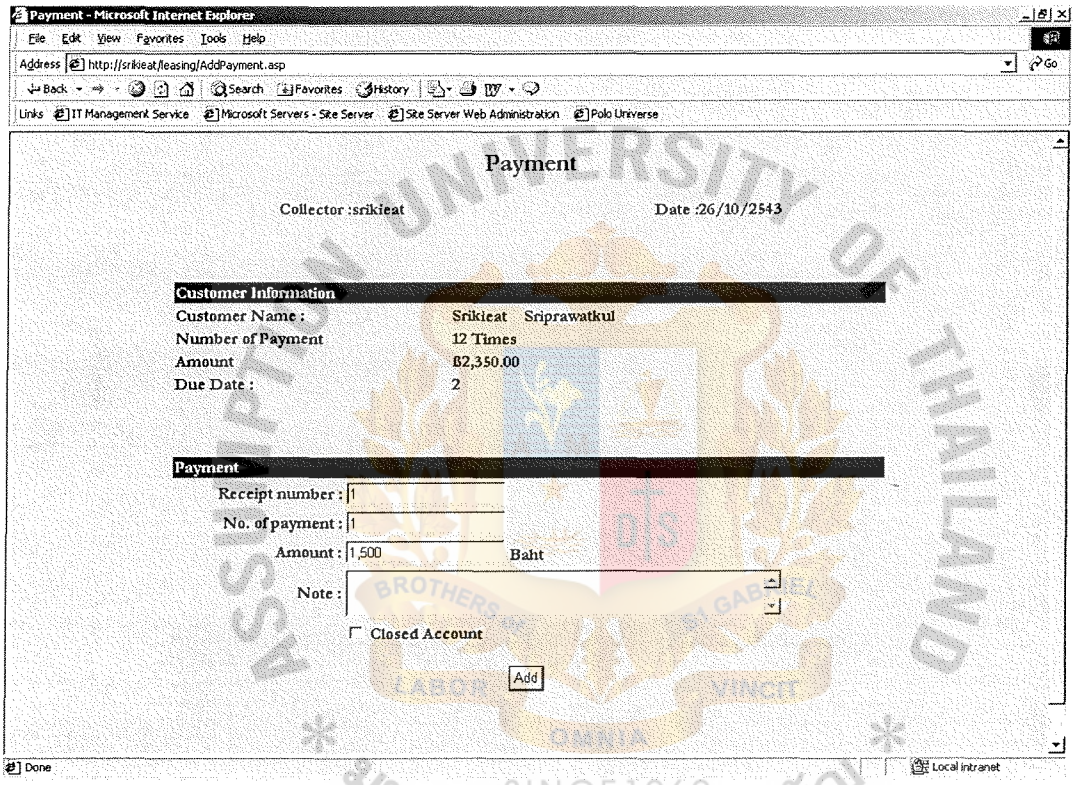

Figure E.7. Add New Payment Screen.

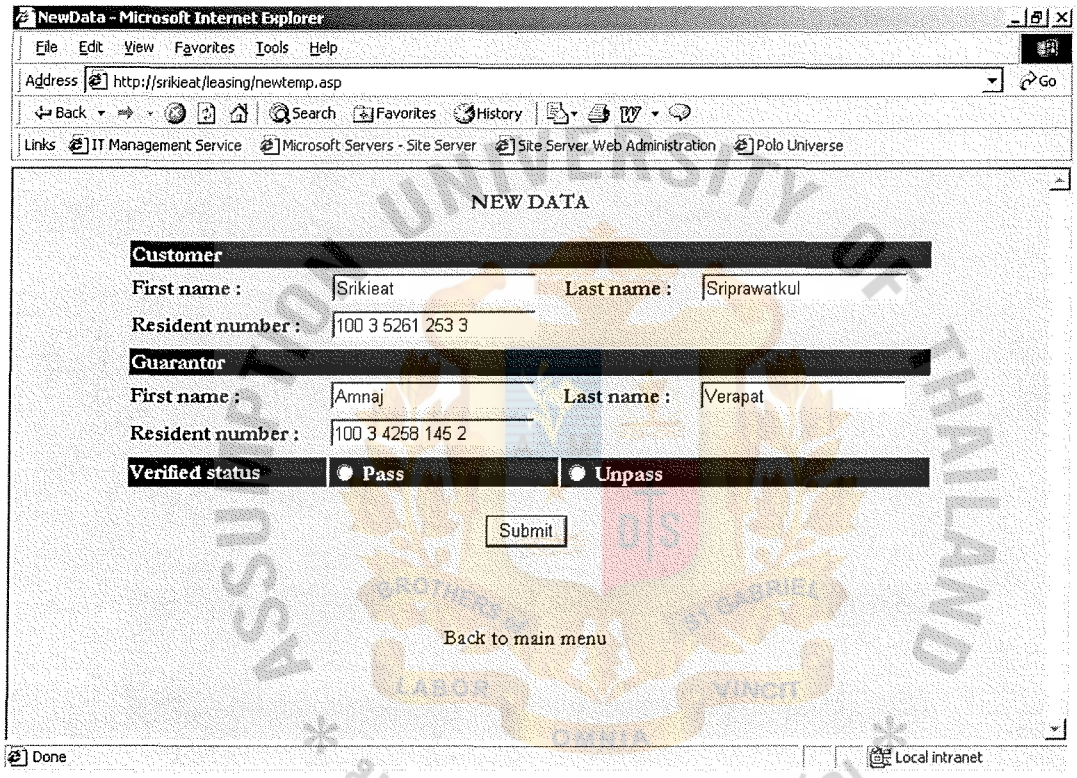

Figure E.8. Add New Data.

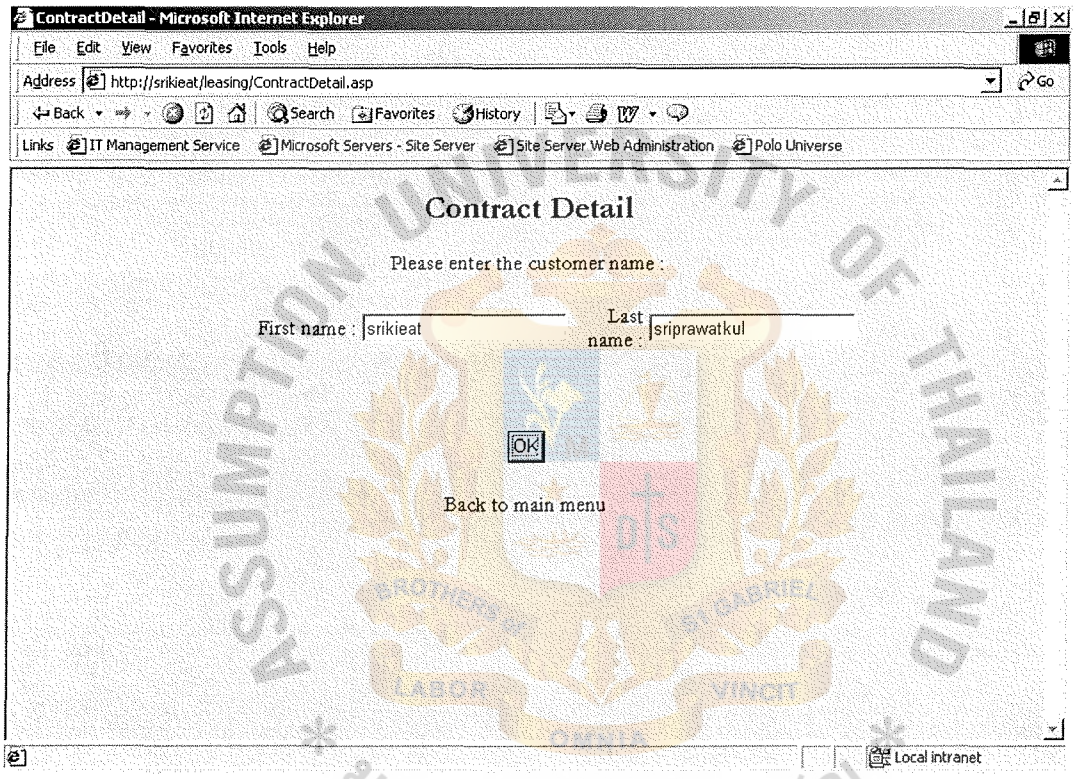

Figure E.9. Search for Contract Detail.

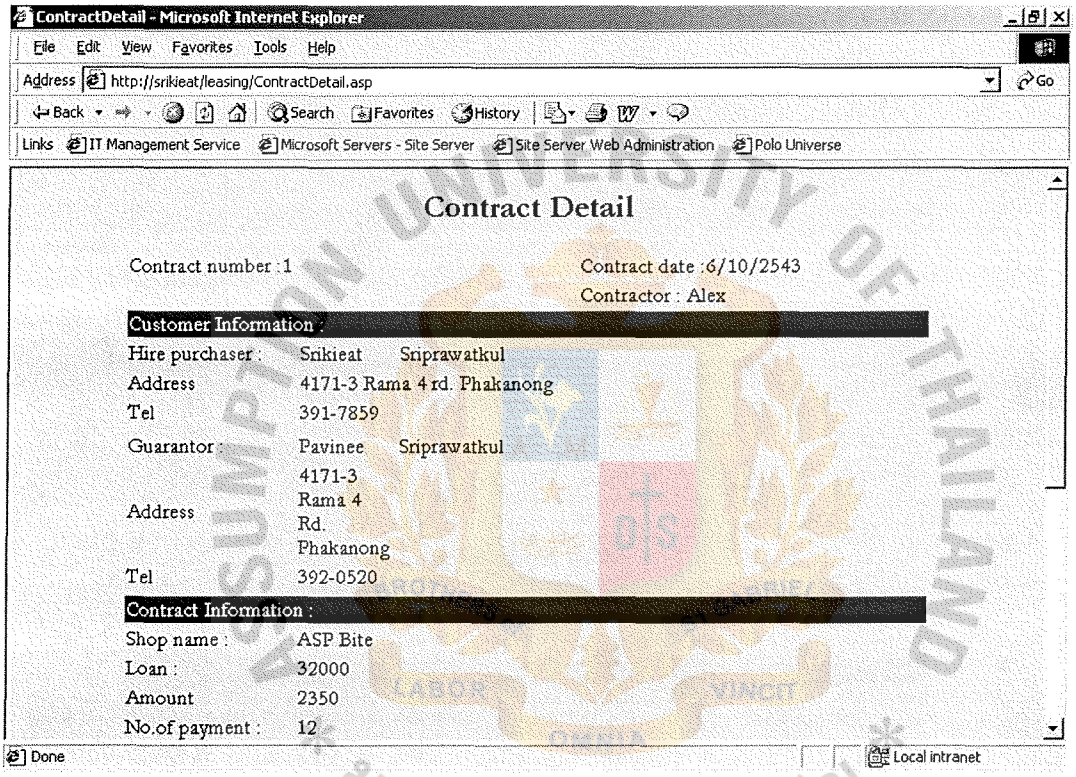

Figure E.10. Contract Detail 1.

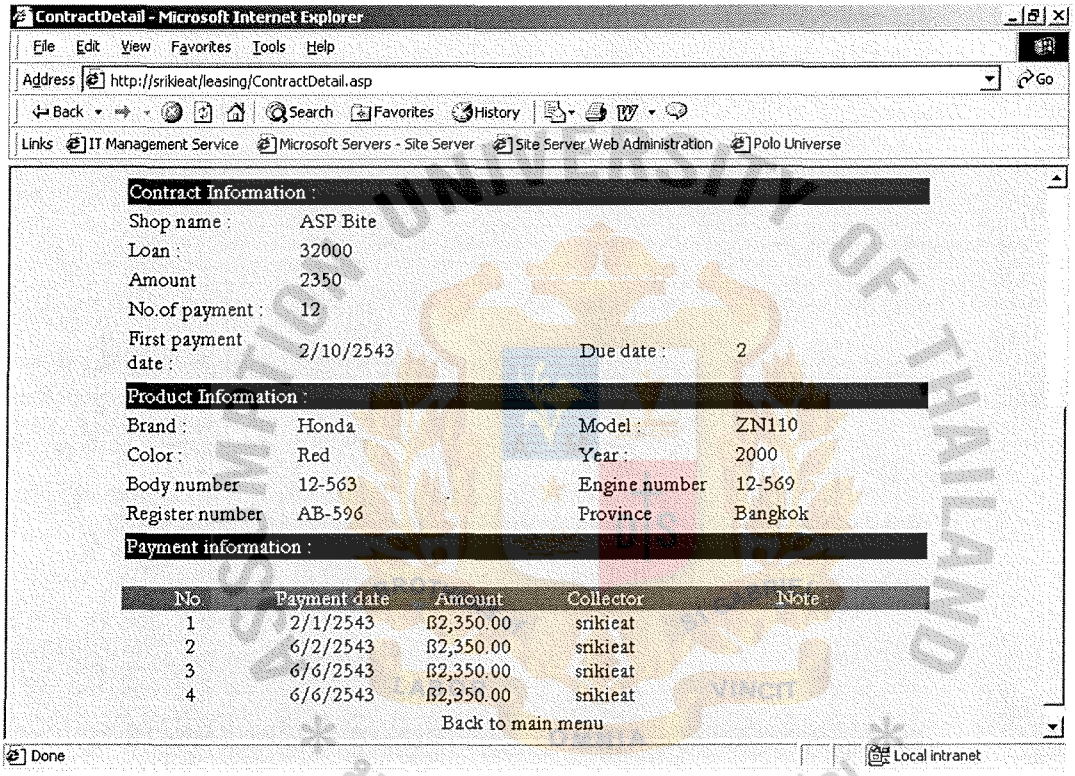

Figure E.11. Contract Detail 2.

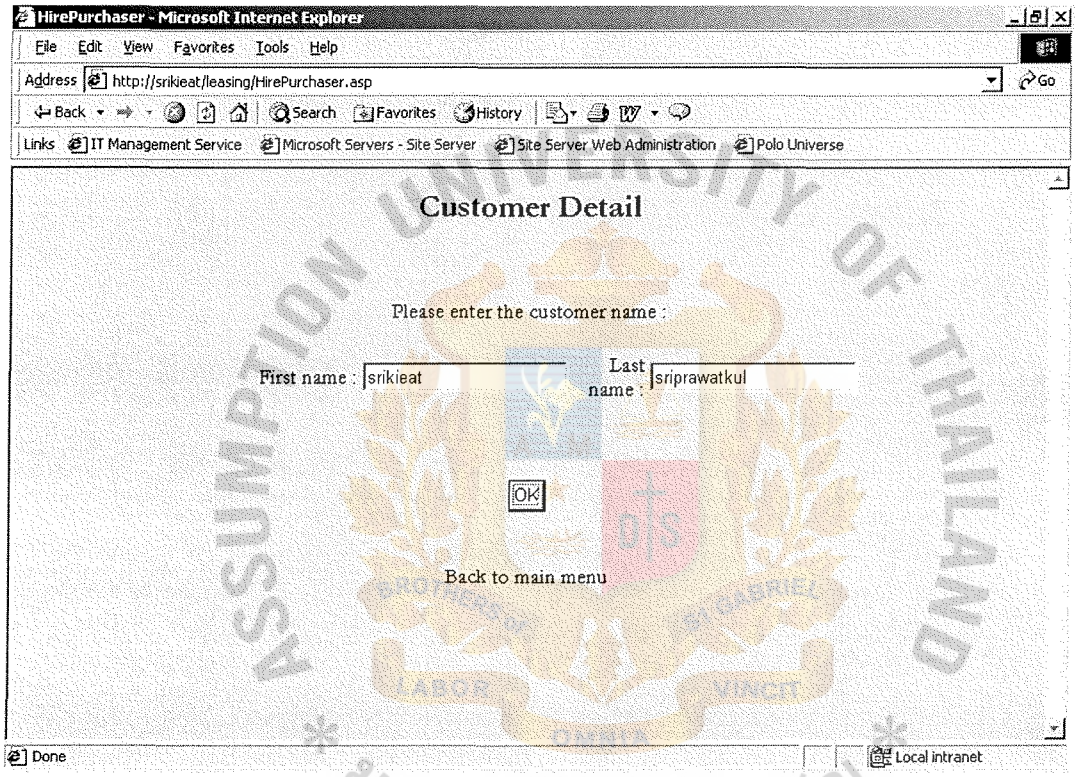

Figure E.12. Search for Customer Detail.

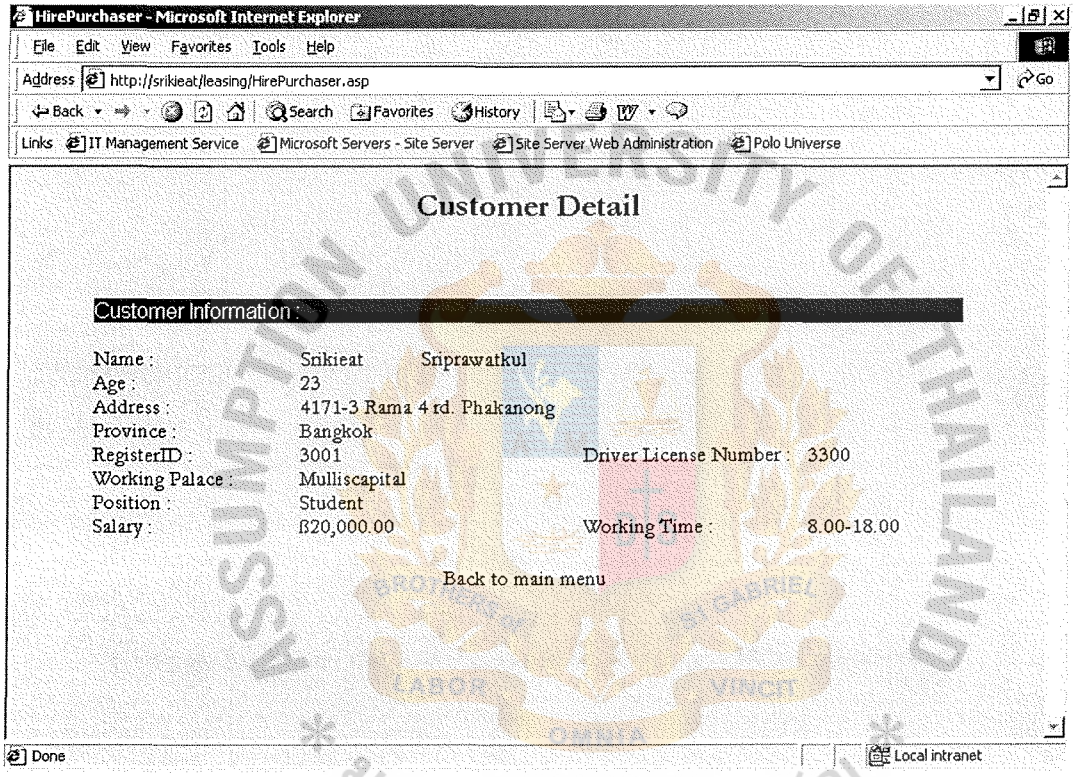

Figure E.13. Customer Detail.

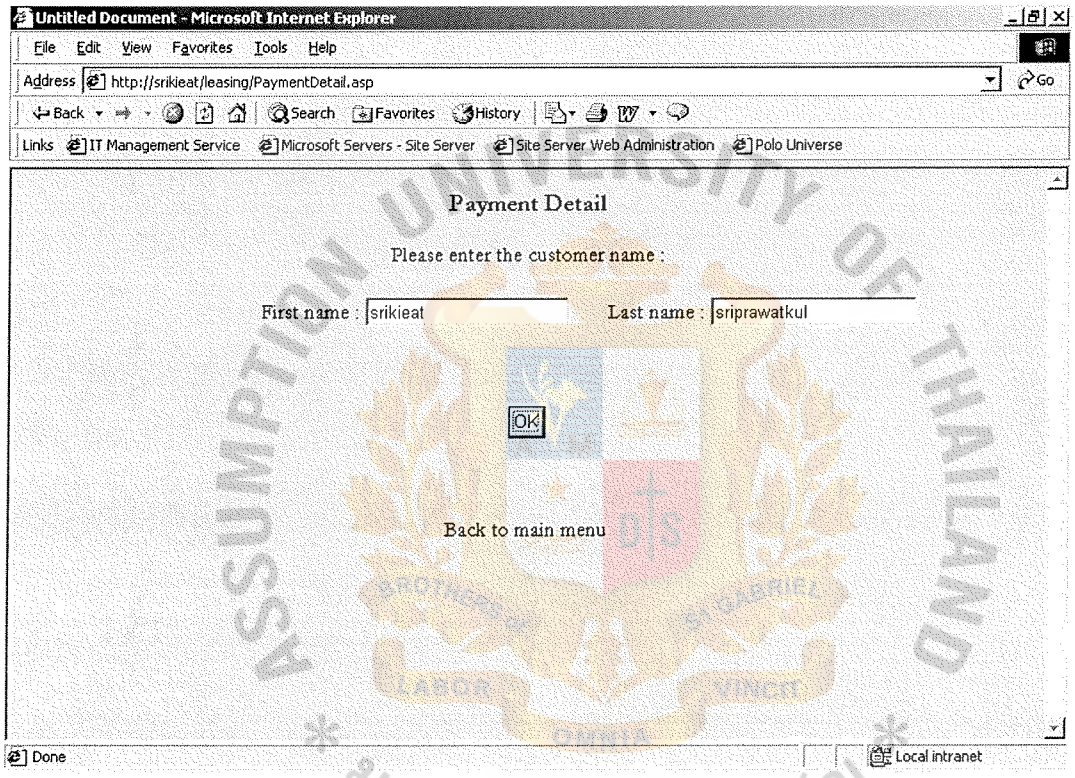

Figure E.14. Search for Payment Detail.

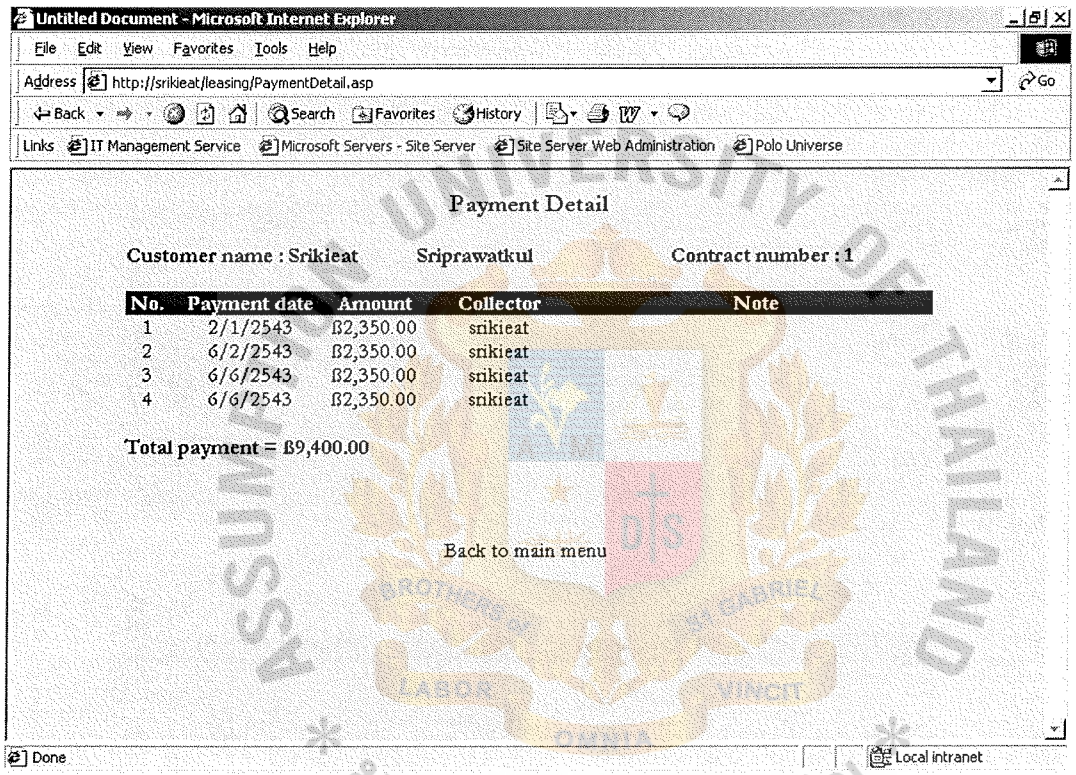

Figure E.15. Payment Detail.

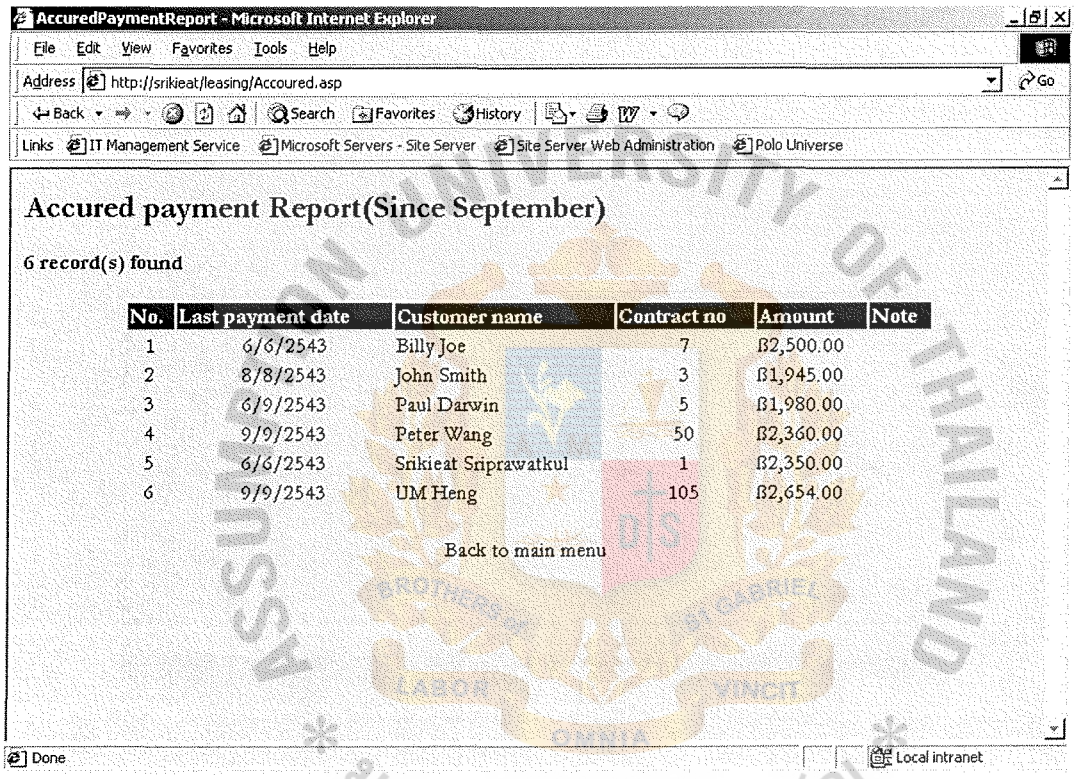

Figure E.16. Accrued Payment Report.

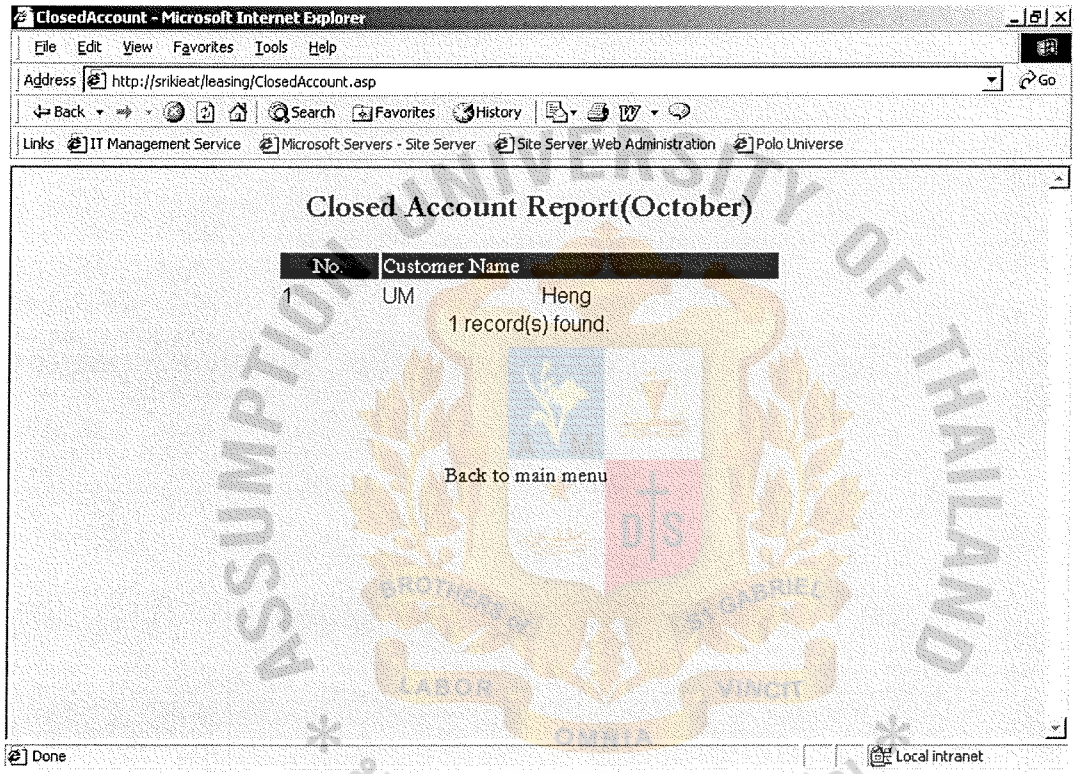

Figure E.17. Closed Account Report.

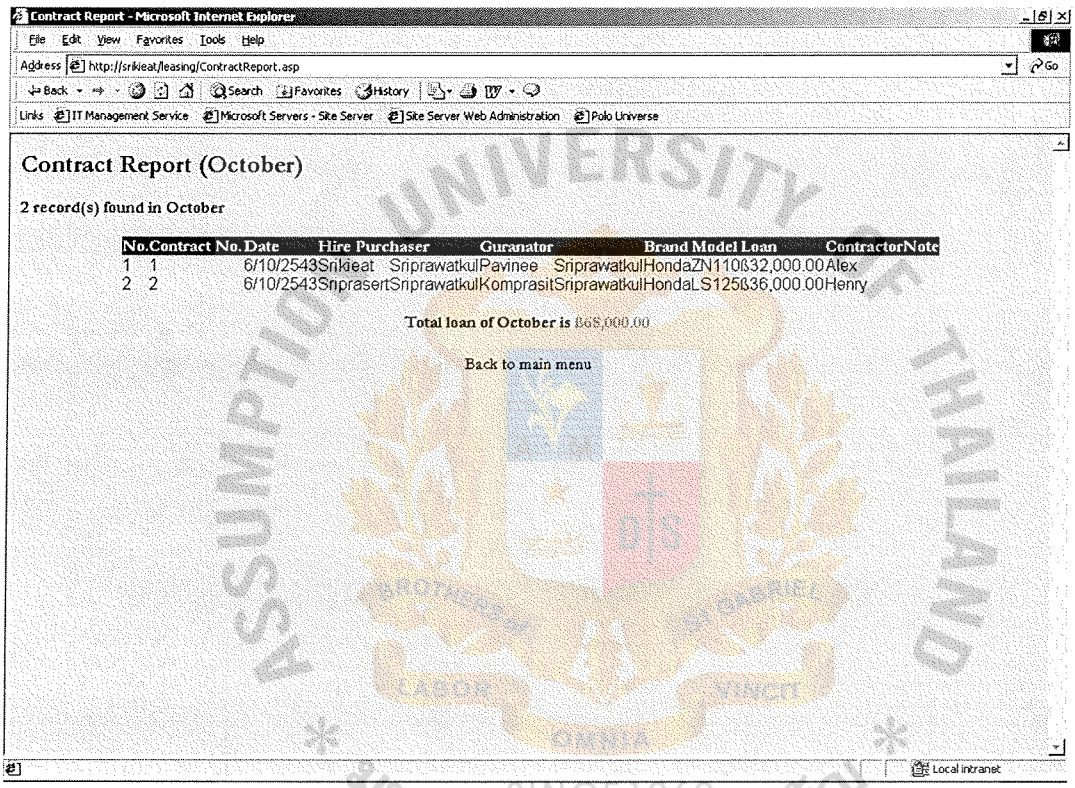

Figure E.18. Contract Report.

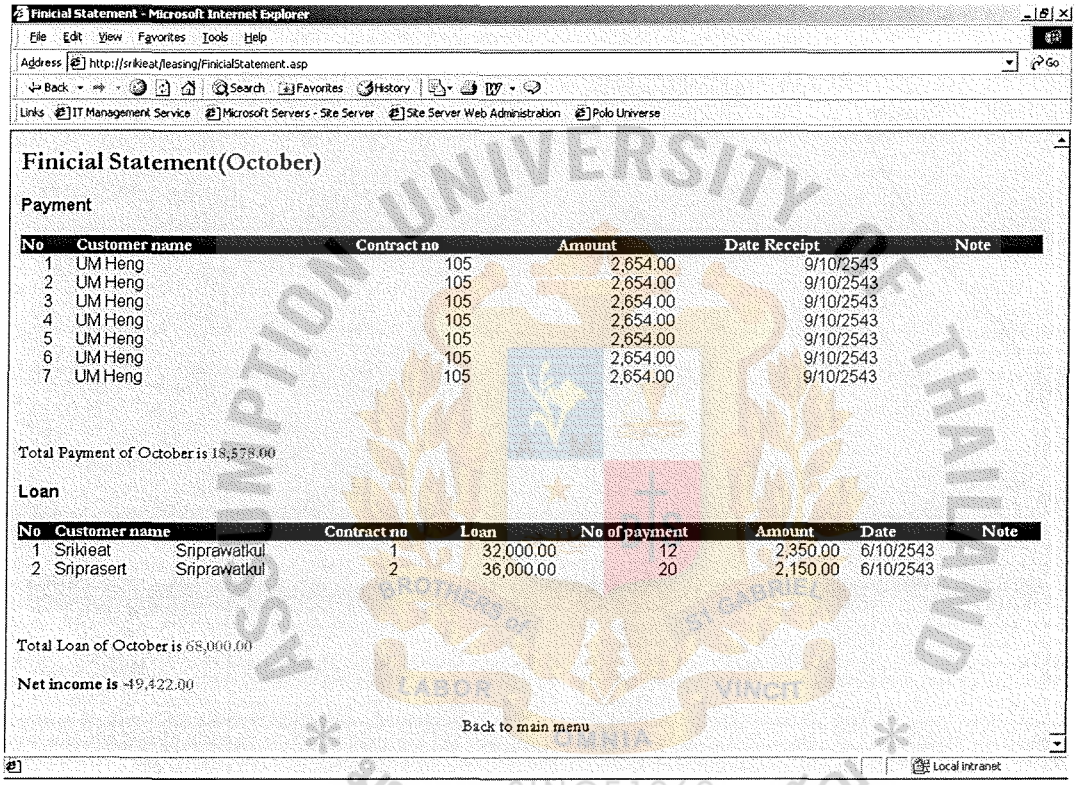

Figure E.19. Finicial Report.

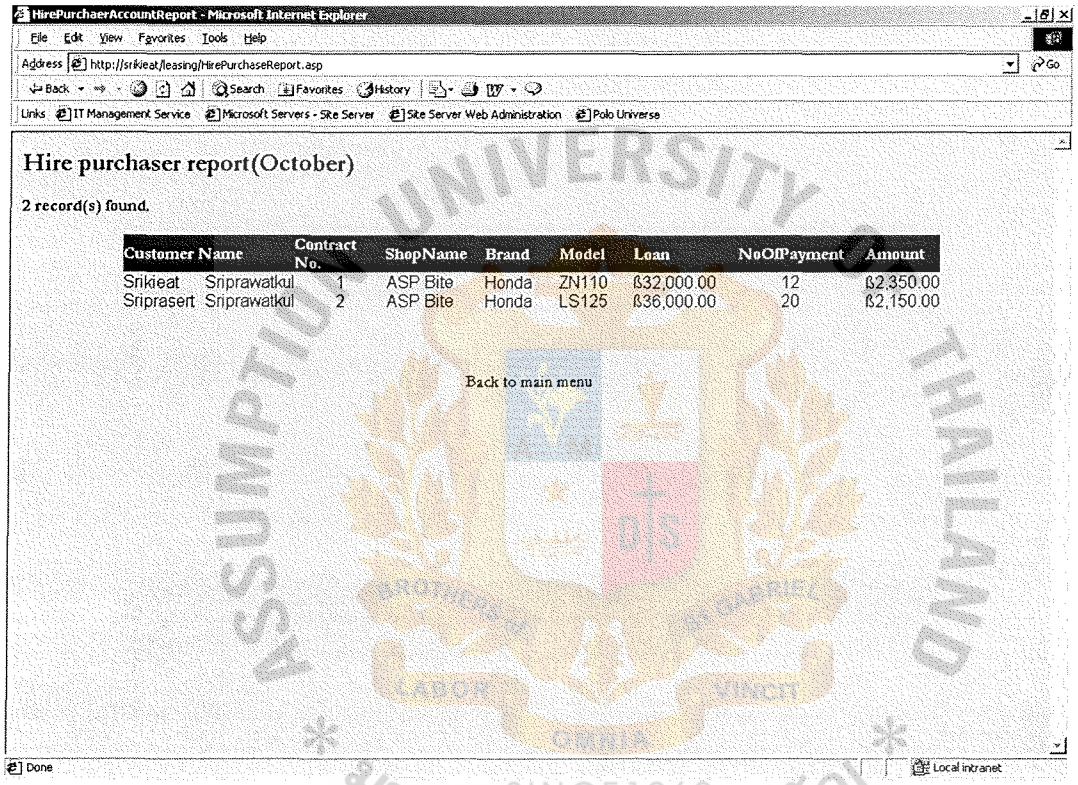

Figure E.20. Hire Purchaser Report.

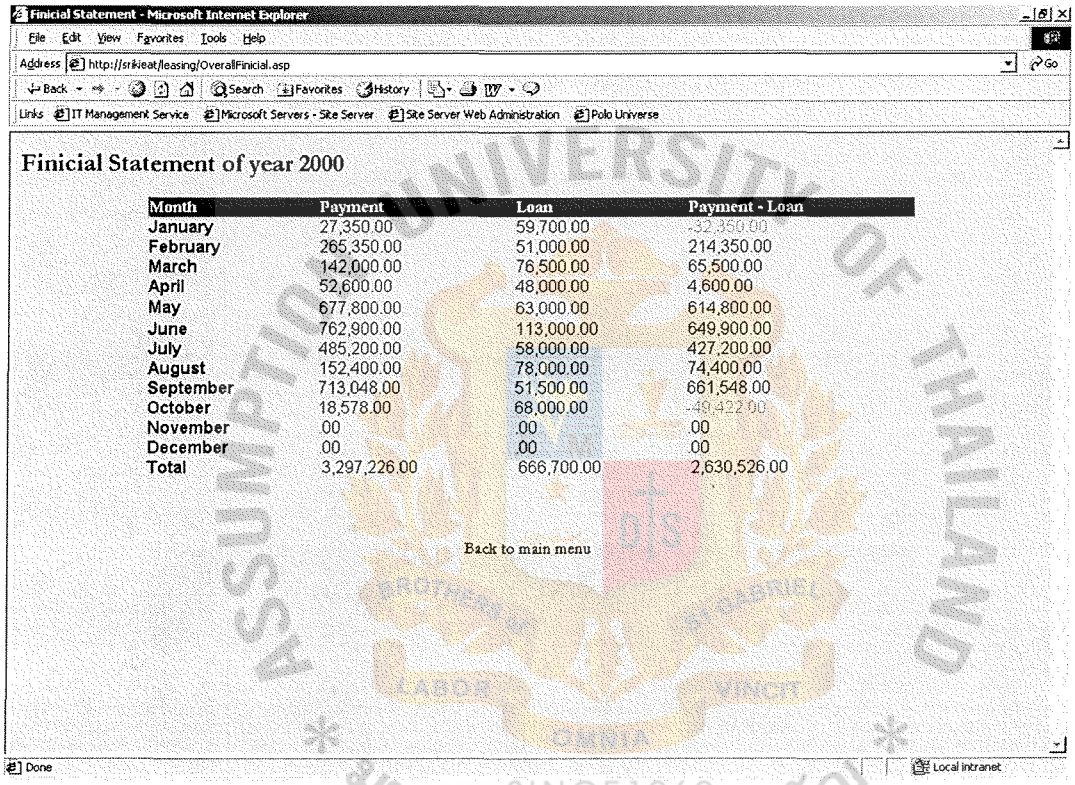

Figure E.21. Finicial Report 2.

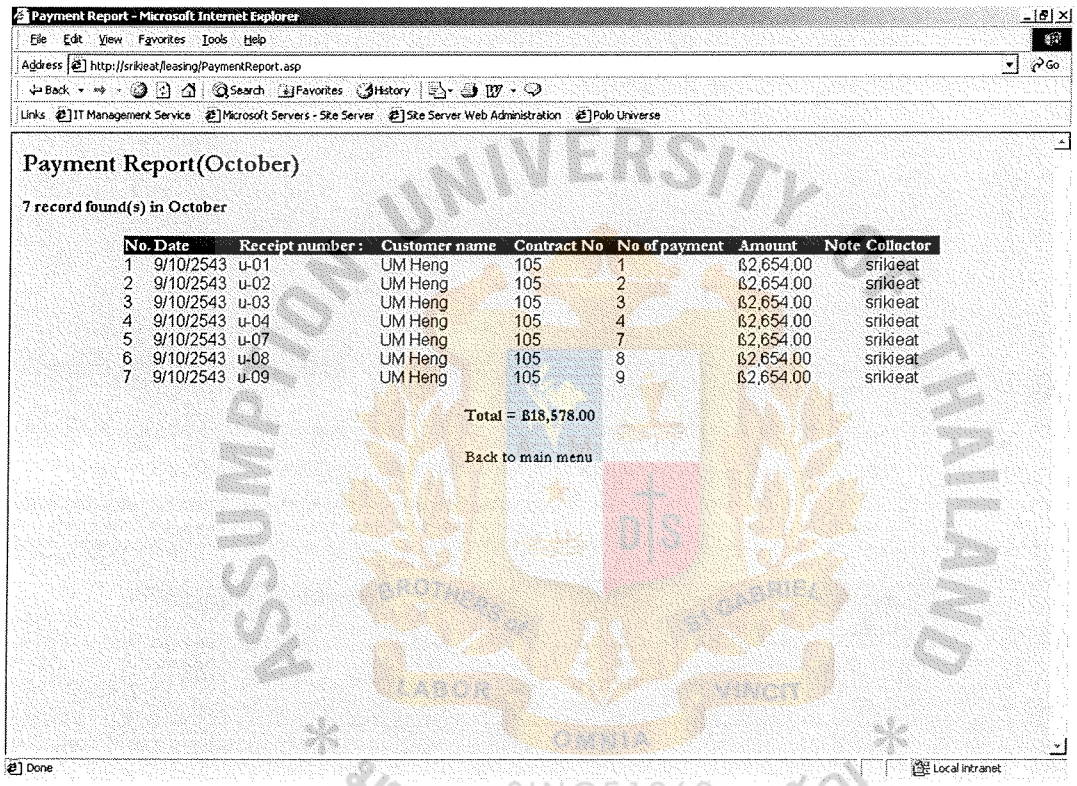

Figure E.22. Payment Report.

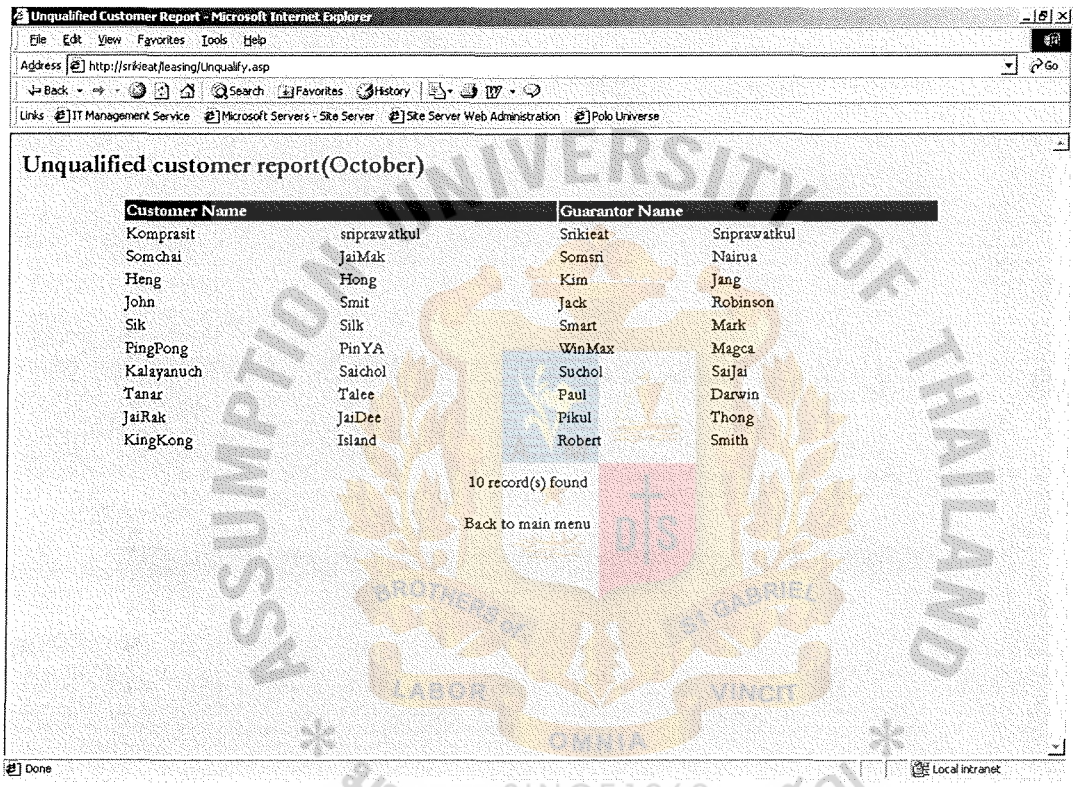

Figure E.23. Unqualified Customer Report

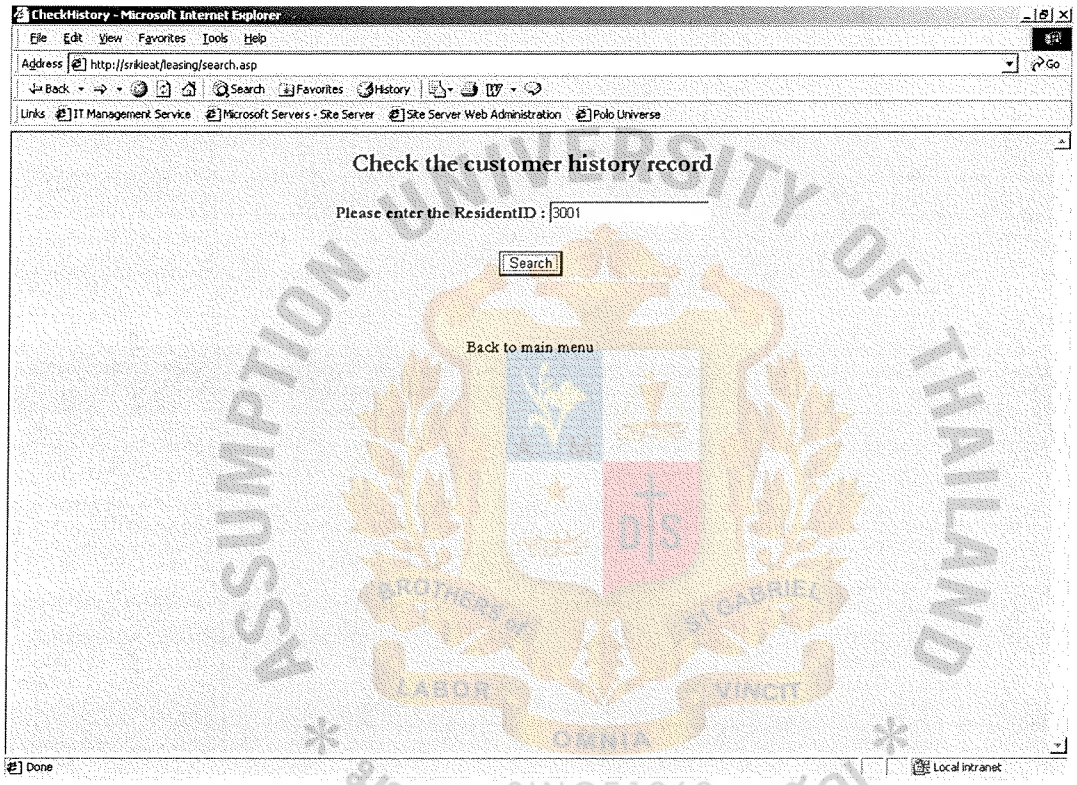

Figure E.25. Search Customer History Record.

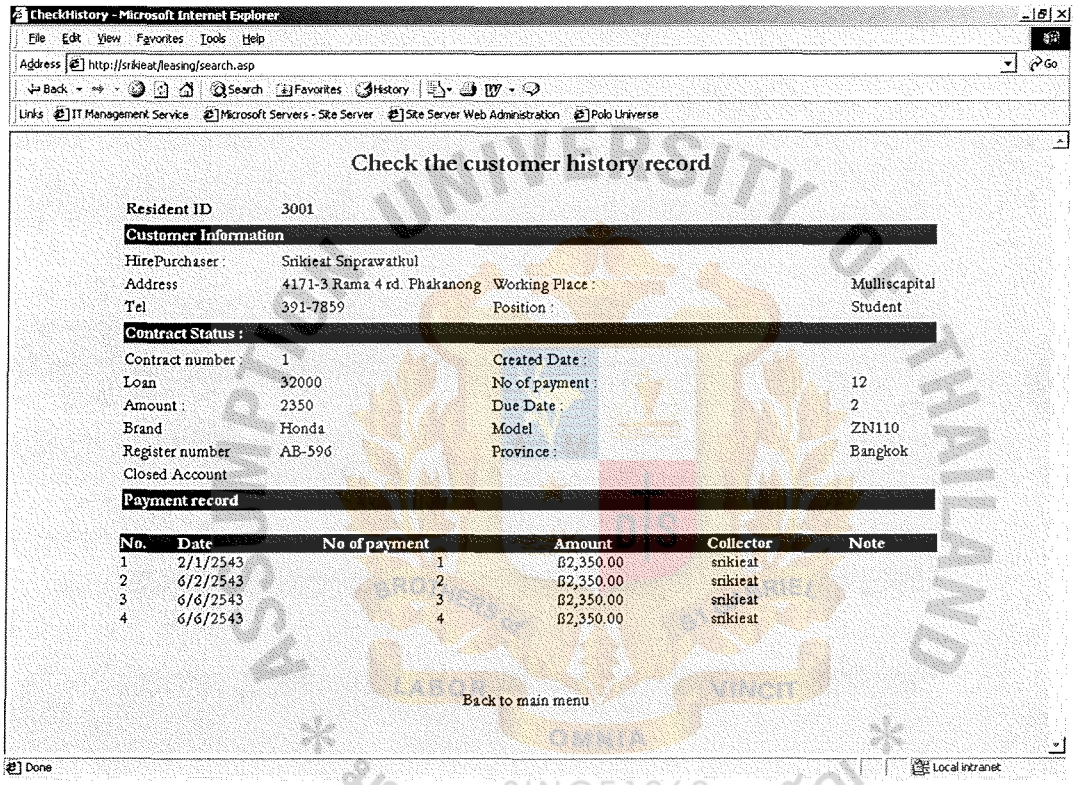

Figure E.25. Result of Search Customer History Record.

APPENDIX F

SOURCE CODE

File name: Default.asp

```
<html>
<head>
<title>Logon</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#CCCCCC">
<p align="center"> 
<p align="center"> 
<p align="center"><b><font face="Garamond">Log On to Leasing System
:</font></b>
<form method="post" action="logon.asp">
 <table width="47%" border="0" align="center">
  <tr>
   <td>
    <div align="right"><font face="Garamond">User Name :</font></div>
   </td>
   <td>
    <input type="text" name="UserName">
   </td>
  </tr>
  <tr>
   <td>
    <div align="right"><font face="Garamond">Password :</font></div>
   </td>
   <td>
    <input type="password" name="Password">
   </td>
  </tr>
 </table>
 <p align="center"> <font face="Garamond">
  <input type="submit" name="Submit" value="Submit">
  </font></p>
```

```
</form>
</body>
</html>
```

File name : Logon.asp

```
<html>
<head>
<title>LogOn Error</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#CCCCCC">
<%
        strUserName = request("username")
        strPassword = request("password")
        session("user") = strUserName
%>
<%
        set conn = server.createObject("ADODB.Connection")
        conn.open "DSN=Leasing"
        sql = "select * from login where name ='"&strUserName&"' and password =
'"&strPassword&"'"
        set rs = conn.execute(sql)
%>
<% if rs.eof then %>
<p align="center"><font color="#FF3333" face="Garamond"><font
size="5"><b><font color="#FF0000">Log
 On Fail.</font></b></font></font></p>
<p align="center"><font color="#FF0000"><b><font face="Garamond" size="5">You
 are not autorized to access this system </font></b></font></p>
<p align="center"><font color="#FF0000"><b><font face="Garamond"
size="5">because
 your username or password are incorrect</font><font face="Garamond"><font
size="4">
```

```
  </font></font><%else%></b></font><b> <% response.redirect "menu.asp"%>
<%end if%>
  <% set conn = nothing %> </b> </p>
</body>
</html>
```

File name: menu.asp

```
<html>
<head>
<title>MainMenu</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
<!--
a:hover {  color: #FFFF00; text-decoration: none}
a:link {  color: #000000; text-decoration: none}
a:active {  color: #CC0000; text-decoration: none}
a:visited {  text-decoration: none}
-->
</style>
</head>
<% strUserName = request("UserName") %>
<body bgcolor="#CCCCCC" vlink="#000000">
<div id="Layer1" style="position:absolute; left:186px; top:122px; width:184px;
height:131px; z-index:1">
 <table width="100%" border="0" align="left">
   <tr>
     <td><b>Add New</b></td>
   </tr>
   <tr>
     <td><font face="Tahoma"><a href="AddContract.asp">Add New
Contract</a></font></td>
   </tr>
   <tr>
```

```
      <td><font face="Tahoma"><a href="AddPayment.asp">Add New
 Payment</a></font></td>
    </tr>
    <tr>
      <td><a href="newtemp.asp"><font face="Tahoma">Add New
Temp</font></a></td>
    </tr>
  </table>
</div>
<div id="Layer2" style="position:absolute; left:636px; top:121px; width:182px;
height:79px; z-index:2">
  <table width="98%" border="0" align="center">
    <tr>
      <td width="47%">
        <div align="left"><b>Show Detail</b></div>
      </td>
    </tr>
    <tr>
      <td width="47%">
        <div align="left"><font face="Tahoma"><a href="ContractDetail.asp">Contract
          Detail</a></font></div>
      </td>
    </tr>
    <tr>
      <td width="47%">
        <div align="left"><font face="Tahoma"><a href="HirePurchaser.asp">Hire
          purcahaser Detail</a></font></div>
      </td>
    </tr>
    <tr>
      <td width="47%">
        <div align="left"><font face="Tahoma"><a href="PaymentDetail.asp"><font
face="Tahoma">Payment
```

```
        Detail</font></a></font></div>
      </td>
    </tr>
  </table>
</div>
<div id="Layer3" style="position:absolute; left:376px; top:121px; width:293px;
height:115px; z-index:3">
  <table width="84%" border="0">
    <tr>
      <td><b> Report</b></td>
    </tr>
    <tr>
      <td><font face="Tahoma"><a href="Accoured.asp">Accured Account
Report</a></font></td>
    </tr>
    <tr>
      <td><font face="Tahoma"><a href="ClosedAccount.asp">Closed Account
Report</a></font></td>
    </tr>
    <tr>
      <td><font face="Tahoma"><a href="ContractReport.asp">Contract Account
Report</a></font></td>
    </tr>
    <tr>
      <td><font face="Tahoma"><a href="FinicialStatement.asp">Finacial Statement
        Report</a></font></td>
    </tr>
    <tr>
      <td><font face="Tahoma"><a href="HirePurchaseReport.asp">Hire purchaser
        Account report</a></font></td>
    </tr>
    <tr>
```

```
      <td><font face="Tahoma"><a href="OverallFinicial.asp">Overall Finicial
 Statement
        Report</a></font></td>
    </tr>
    <tr>
      <td><font face="Tahoma"><a href="PaymentReport.asp">Payment Account
 Report</a></font></td>
    </tr>
    <tr>
      <td><a href="Unqualify.asp"><font face="Tahoma">Unqualified customer
 Report</font></a></td>
    </tr>
  </table>
</div>
<div id="Layer4" style="position:absolute; left:373px; top:445px; width:242px;
 height:57px; z-index:4">
  <p align="center"><font face="Tahoma"><a href="search.asp">Check the customer
    history record</a></font></p>
  <div align="center"><font face="Tahoma"><a href="default.asp">Log
 Off</a></font>
  </div>
  <p> </p>
</div>
<div align="center">
  <p><b><font face="Tahoma">Leasing Main Menu</font></b> </p>
  <table width="80%" border="0">
    <tr>
      <td height="10"><font face="Tahoma"><b>Welcome :
 <%=session("user")%></b></font></td>
      <td height="10">
        <div align="right"><font face="Tahoma"><b>Date :
 <%=formatDateTime(date())%></b></font></div>
      </td>
```

```
  </tr>
 </table>
 <p> </p>
 <p><b> </b></p>
 <p> </p>
 </div>
</body>
</html>
```

File name : AddContractData.asp

```
<%
'******* Get Hire Purchasing Data ***********
strHFirstName = request("HfirstName")
strHlastName = request("HlastName")
strHAge = request("HAge")
strHAddress = request("Haddress")
strHTel = request("Htel")
strHProvince = request("Hprovince")
strHResidentNo = request("HresidentNo")
strHLicence = request("Hlicence")
strHoccupation = request("Hoccupation")
strHPlace = request("Hplace")
strHPosition = request("Hposition")
strHSalary = request("Hsalary")
strHtime = request("Htime")
'*****************************************

'*********insert into hire purchaser database ********************
set conn = Server.CreateObject("ADODB.Connection")
conn.open "DSN=Leasing"
SQL = "insert into HirePurchaser "
SQL = SQL & "(FirstName,
LastName,Age,Address,Province,tel,residentID,DriverLicenseNumber,Occupation,Wor
kingPlace,Position,Salary,WorkingTime) "
```

```
SQL = SQL & "values ('" & strHfirstname & "','" & strHlastName & "','" & strHage &
"','" & strHaddress & "', '" & strHprovince & "', '" & strHtel & "', '" & strHResidentNo
& "','" & strHLicence & "','" & strHOccupation & "','" & strHPlace & "','" &
strHPosition & "','" & strHSalary & "','" & strHtime & "')"
set HirePurchaser = conn.execute(SQL)
'*****************************************************************
'******* Get Guarantor Data ************
strgFirstName = request("gfirstName")
strglastName = request("glastName")
strgAge = request("gAge")
strgAddress = request("gaddress")
strgTel = request("gtel")
strgProvince = request("gprovince")
strgResidentNo = request("gresidentNo")
strgLicence = request("glicence")
strgoccupation = request("goccupation")
strgPlace = request("gplace")
strgPosition = request("gposition")
strgSalary = request("gsalary")
strgtime = request("gtime")
'*****************************************

'*********insert into Guarantor database *********************
'set conn = Server.CreateObject("ADODB.Connection")
'conn.open "DSN=Leasing"
SQL = "insert into Guarantor "
SQL = SQL & "(FirstName,
LastName,Age,Address,Province,tel,residentID,DriverLicenseNumber,Occupation,Wor
kingPlace,Position,Salary,WorkingTime) "
SQL = SQL & "values ('" & strGfirstname & "','" & strGlastName & "','" & strGage &
"','" & strGaddress & "', '" & strGprovince & "', '" & strGtel & "', '" & strGResidentNo
```

```
& "','" & strGLicence & "','" & strGOccupation & "','" & strGPlace & "','" &
strGPosition & "','" & strGSalary & "','" & strGtime & "')"
set Guarnator = conn.execute(SQL)
'*****************************************************************

'************* Get contract data **********************
strContractNo = Cint(request("ContractNo"))
strShop = request("shop")
strContractor = request("contractor")
strLoan = request("loan")
strAmount = request("Amount")
strNoPayment = request("NoPayment")
strFirstDate = request("FirstDate")
strDueDate = request("DueDate")
strBrand = request("brand")
strModel = request("Model")
strColor = request("Color")
strYear = request("Year")
strBnumber = request("Bnumber")
strEnumber = request("Enumber")
strRnumber = request("Rnumber")
strprovince = request("Province")
strNote = request("note")
'*****************************************************

'***************Add data to Contract Database ******************
SQL = ""
SQL = "insert into Contract "
SQL = SQL &
"(ContractNo,ShopName,Contractor,Loan,Amount,NoOFPayment,FirstPAymentDate,D
ueDate,Brand,Model,Color,Year,BodyNumber,EngineNumber,RegisterNumber,Provinc
e,CreatedDate,CustomerID,GuarantorID) "
```

```
SQL = SQL & "values ('"& strContractNo &"','"& strShop &"','"& strContractor
&"','"& strLoan &"','"& strAmount &"','"& strNoPayment &"','"& strFirstDate &"','"&
strDueDate &"','"& strBrand &"','"& strModel &"','"& strColor &"','"& strYear &"','"&
strBnumber &"','"& strEnumber &"','"& strRnumber &"','"& strProvince &"',date(),'"&
strHresidentNo &"','"& strGresidentNo &"')"
Set Contract = conn.execute (SQL)
success = "true"
'*****************************************************************
%>
<%
If success = "true" then
response.redirect "menu.asp"
End if
%>
```

File name: AddPayment

```
<%
'********************** Get Data ******************************
        strPayment = request("Payment")
        strAmount = request("Amount")
        strNote = request("note")
        strClosedAccount = request("ClosedAccount")
        strCollector = session("user")
        strContractNo = request("ContractNo")
        strFirstName = request("firstname")
        strLastName = request("lastname")
        strReceipt = request("Receipt")
'*****************************************************************
'********** Make condition**********************
if strCloserdAccount = "" then
        strClosedAccount = "no"
else strClosedAccount = "yes"
end if
'************************************************
```

```
'********************* Connect to Database *********************
set conn = Server.CreateObject("ADODB.Connection")
conn.open = "DSN=Leasing"
SQL = "insert into payment "
SQL = SQL &
"(ReceiptNo,CustomerName,ContractNo,NoOfPayment,Amount,Collector,DateReceipt
) "
SQL = SQL & "values ('"& strReceipt &"'," & strFirstName &" "& strLastName & "',
'" & strContractNo & "', '" & strPayment & "', '" & strAmount & "', '" & strCollector &
"',date())"
set rs = conn.execute(SQL)
'*********************************************************
'********************* insert to contract DB *********************
SQL = ""
SQL = "update contract "
SQL = SQL & "set ClosedAccount = '" & strClosedAccount & "' "
SQL = SQL & "where ContractNo = " & strContractNo & ""
set contract = conn.execute(SQL)
'*********************************************************
response.redirect "menu.asp"
%>
```

File name : AddTempDB.asp

```
<%
'***************** get data *********************

strCfirstName = request("cfirstname")
strClastName = request("clastname")
strCID = request("CID")
strGfirstName = request("gFirstName")
strGlastName = request("gLastName")
strGID = request("GID")
strVerify = request("verify")
'*********************************************
```

```
'********** connect to DB and add data **********
set conn = Server.CreateObject("ADODB.Connection")
conn.open = "DSN=leasing"
set rs = conn.execute("insert into tempDB (CFirstName, ClastName,
CID,GFirstName,GlastName,GID,verify,CreatedDate) values
('"&strCfirstName&"','"&strCLastName&"','"&strCID&"' , "'& strGfirstName &"', '" &
strGlastName & "', "'&strGID&"','"&strVerify&"',date())")
response.redirect "menu.asp"
'*********************************************
%>
```

File name : Closed account.asp

```
<html>
<head>
<title>ClosedAccount</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<!--#include file = "function.asp"-->
<body bgcolor="#CCCCCC">
<% Mymonth = month(date()) %> <% call convertToMonth(Mymonth) %>
<p align="center"><font size="5"><b><font face="Garamond">Closed Account
Report(<% = mymonth %>)
  </font></b></font> <%
'***************** Connect to database **********************
Set conn = Server.CreateObject("ADODB.Connection")
conn.open "DSN=Leasing"
SQL = "select  distinct h.Firstname , h.Lastname from contract c,HirePurchaser h ,
payment p where p.contractNo = c.contractNo and c.customerID = h.residentID and
c.closedAccount = 'yes' and month(dateReceipt) = month(date())"
Set rs = conn.execute(SQL)
'*********************************************************
%>
<% if rs.eof then %> <% end if %> </p>
<table width="50%" border="0" align="center">
```

```
  <tr bgcolor="#333333">
    <td>
      <div align="center"><font color="#FFFFFF" face="Garamond">No.</font></div>
    </td>
    <td colspan="2"><font color="#FFFFFF" face="Garamond">Customer
Name</font></td>
  </tr>
          <% i = 0 %>
  <% do until rs.eof %>
  <tr>  <% i = i + 1 %>
    <td><% = i %></td>
    <td><%= rs("Firstname") %> </td>
    <td><% = rs("LastName") %></td>
  </tr>
  <% rs.movenext %>
 <% loop %>
</table>
<div align="center"><% = i %> record(s) found. </div>
<p>  </p>
<p> </p>
<!--#include file = "i_bottom.asp"-->
</body>
</html>
```

File name : Contract report

```
<html>
<head>
<title>Contract Report</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#CCCCCC">
<!--#include file = "function.asp"-->
<%
'************************ Connect to DB ************************
```

```
Set conn = server.CreateObject("ADODB.Connection")
conn.open "DSN=Leasing"
Set rs = server.CreateObject("ADODB.recordset")
rs.cursorlocation = 3
SQL = " select p.firstname as pfname, p.lastname as plname, g.firstname as gfname,
g.lastname as glname, createdDate,contractno,brand,model,loan,contractor,note from
contract c , hirepurchaser p , guarantor g where c.customerID = p.residentID and
c.guarantorId = g.residentId and month(CreatedDate) = month(date())"
rs.open sql ,conn
'*****************************************************************
%>
<% Mymonth = month(date()) %> <% call ConvertToMonth(Mymonth) %>
<p><b><font face="Garamond" size="5">Contract Report (<%=MyMonth%>)
</font></b></p>
<p><b><font face="Garamond"><% = rs.recordcount %> record(s) found in
<%=Mymonth%></font></b></p>
<table width="80%" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr bgcolor="#333333">
    <td><b><font color="#FFFFFF" face="Garamond">No.</font></b></td>
    <td><b><font color="#FFFFFF" face="Garamond">Contract No. </font></b></td>
    <td><b><font color="#FFFFFF" face="Garamond">Date</font></b></td>
    <td colspan="2"><b><font color="#FFFFFF" face="Garamond">Hire
Purchaser</font></b></td>
    <td colspan="2"><b><font color="#FFFFFF" face="Garamond">Guranator
</font></b></td>
    <td><b><font color="#FFFFFF" face="Garamond">Brand</font></b></td>
    <td><b><font color="#FFFFFF" face="Garamond">Model</font></b></td>
    <td><b><font color="#FFFFFF" face="Garamond">Loan</font></b></td>
    <td><b><font color="#FFFFFF" face="Garamond">Contractor</font></b></td>
    <td><b><font color="#FFFFFF" face="Garamond">Note</font></b></td>
  </tr>
  <tr> <% i = 1 %> <% count = 0 %>
<% do until rs.eof %>
```

```
    <td><% = i %></td>
    <td><%= rs("ContractNo")%></td>
    <td><% = rs("createdDate")%></td>
    <td><% = rs("pfname")%></td>
    <td><% = rs("plname")%></td>
    <td><% = rs("gfname")%></td>
    <td><% = rs("glname") %></td>
    <td><% = rs("Brand")%></td>
    <td><% = rs("model")%></td>
    <td><% = formatcurrency(rs("loan"))%></td>
    <td><% = rs("Contractor")%></td>
    <td><% = rs("note")%></td>
  </tr><% i = i+ 1 %> <% count = count + rs("loan") %>
<% rs.movenext %>
<% loop %>
</table>
<p align="center"><b><font face="Garamond">Total loan of <%=Mymonth %>
is<font color="#FF0000">
  <% = formatcurrency(count) %> </font> </font></b></p>
<!--#include file ="i_bottom.asp"-->
</body>
</html>
```

File name : Finance statement

```
<html>
<head>
<title>Finicial Statement</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<!--#include file = "function.asp"-->
<% Mymonth = month(date()) %>
<% call ConvertTomonth(Mymonth) %>
<%
'********************* Connect to database **********************
```

```
Set conn = Server.CreateObject("ADODB.Connection")
conn.open "DSN=Leasing"
Set rs = Server.CreateObject("ADODB.Recordset")
SQL = "select * from payment where month(dateReceipt) = month(date())"
rs.open SQL,Conn
'*****************************************************************
'********************* Connnect to contract *********************
Set rsContract = Server.CreateObject("ADODB.Recordset")
SQL = ""
SQL = "select * from contract c,HirePurchaser h where h.ResidentID = c.customerID
and month(createdDate) = month(date())"
rsContract.open SQL,conn
'*****************************************************************
%>
<body bgcolor="cccccc">
<p><font size="5"><b><font face="Garamond">Finicial
Statement(<%=Mymonth%>)</font></b></font>
</p>
<p><b>Payment</b></p>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr bgcolor="#333333">
    <td><font face="Garamond"><b><font
color="#FFFFFF">No</font></b></font></td>
    <td><font face="Garamond"><b><font color="#FFFFFF">Customer
name</font></b></font></td>
    <td><font face="Garamond"><b><font color="#FFFFFF">Contract no
</font></b></font></td>
    <td><font face="Garamond"><b><font
color="#FFFFFF">Amount</font></b></font></td>
    <td><font face="Garamond"><b><font color="#FFFFFF">Date
Receipt</font></b></font></td>
    <td><font face="Garamond"><b><font
color="#FFFFFF">Note</font></b></font></td>
```

```
  </tr>
  <% i = 1 %> <% countPayment = 0 %>
  <tr><% do until rs.eof %>
   <td>
    <div align="center"><% = i %> </div>
   </td>
   <td> <%= rs("CustomerName")%>
    <div align="left"></div>
   </td>
   <td>
    <div align="center"> <% = rs("ContractNo") %> </div>
   </td>
   <td>
    <div align="center"> <% = formatnumber(rs("Amount")) %></div>
   </td>
   <td>
    <div align="center"> <% =rs("DateReceipt") %></div>
   </td>
   <td>
    <div align="center"> <% = rs("Note")%></div>
   </td>
  </tr><% countPayment = countPayment + rs("Amount") %>
  <% rs.movenext %><% i = i +1 %> <% loop %>
</table>
<p> </p>
<p><font face="Garamond">Total Payment of <%=Mymonth %> is<font
color="#FF0000">
  <b><% = Formatnumber(CountPayment) %> </b></font></font> </p>
<p><b>Loan </b></p>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr bgcolor="#333333">
    <td><font color="#FFFFFF" face="Garamond"><b>No</b></font></td>
```

```
  <td colspan="2"><font color="#FFFFFF" face="Garamond"><b>Customer
name</b></font></td>
  <td><font color="#FFFFFF" face="Garamond"><b>Contract no </b></font></td>
  <td><font color="#FFFFFF" face="Garamond"><b>Loan</b></font></td>
  <td><font color="#FFFFFF" face="Garamond"><b>No of payment</b></font></td>
  <td><font color="#FFFFFF" face="Garamond"><b>Amount</b></font></td>
  <td><font color="#FFFFFF" face="Garamond"><b>Date</b></font></td>
  <td><font color="#FFFFFF" face="Garamond"><b>Note</b></font></td>
 </tr>
 <% i = 1 %> <% CountLoan = 0 %>
 <tr><% do until rsContract.eof %>
  <td>
   <div align="center"><% = i %> </div>
  </td>
  <td>
   <div align="left"><%= rsContract("FirstName")%> </div>
  </td>
  <td>
   <div align="left"> <% = rsContract("Lastname")%></div>
   </td>
  <td>
   <div align="center"> <% = rsContract("ContractNo") %> </div>
  </td>
  <td>
   <div align="center"><% = formatnumber(rsContract("Loan")) %></div>
  </td>
  <td>
   <div align="center"><% = rsContract("NoOfPayment")%></div>
  </td>
  <td>
   <div align="center"><% = formatnumber(rsContract("Amount")) %></div>
  </td>
  <td><% =rsContract("CreatedDate") %>
```

```
      <div align="center"></div>
    </td>
    <td>
      <div align="center"> <% = rsContract("Note")%></div>
    </td>
  </tr><% countLoan = countLoan + rsContract("Loan") %>
  <% rsContract.movenext %><% i = i +1 %> <% loop %>
</table>
<p> </p>
<p><font face="Garamond">Total Loan of <% = mymonth %> is<font
color="#CC0000">
  </font><b><font face="Garamond"><font color="#ff0000"><% =
Formatnumber(countLoan) %></font></font></b></font></p>
<font face="Garamond"><% countTotal = countPayment - countLoan %> </font>
<p><b><font face="Garamond">Net income is<font color="#FF0000"> <% =
Formatnumber(CountTotal) %></font></font></b></p>
<!--#include file = "i_bottom.asp"-->
</body>
</html>
```

File name : function.asp

```
<%
function ConvertToMonth(m)
select case m
case 1 :
        m1 = "January"
case 2 :
        m1 = "February"
case 3 :
        m1 = "March"
case 4 :
        m1 = "April"
case 5 :
        m1 = "May"
```

```
case 6 :
        m1 = "June"
case 7 :
        m1 = "July"
case 8 :
        m1 = "August"
case 9 :
        m1 = "September"
case 10 :
        m1 = "October"
case 11 :
        m1 = "November"
case 12 :
        m1 = "December"
end select
        m = m1
end function
function color(x)
if x < 0 then
        response.write "<font color = red>" & x &"</font>"
else
        response.write"<font color = black>" & x & "</font>"
end if
end function
%>
```

File name : Hire purchaser.asp

```
<html>
<head>
<title>HirePurchaser</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#CCCCCC">
<div align="center">
```

```
  <p><font size="5"><b><font face="Garamond">Customer Detail</font></b></font>
  </p>
  <p>  </p>
 </div>
 <!--#include file= "i_search.asp"-->
 <table width="80%" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr bgcolor="#333333">
   <td width="24%"><font color="#FFFFFF">Customer Information : </font></td>
   <td colspan="2"><font color="#FFFFFF"></font></td>
   <td width="26%"><font color="#FFFFFF"></font></td>
   <td width="15%"><font color="#FFFFFF"></font></td>
   <td width="0%"><font color="#FFFFFF"></font></td>
   <td width="2%"><font color="#FFFFFF"></font></td>
  </tr>
  <tr>
   <td width="24%"> </td>
   <td width="14%"> </td>
   <td width="19%"> </td>
   <td width="26%"> </td>
   <td width="15%"> </td>
   <td width="0%"> </td>
   <td width="2%"> </td>
  </tr>
  <tr>
   <td width="24%"><font face="Garamond">Name :</font></td>
   <td width="14%"><font face="Garamond"><% =rsCustomer("FirstName")
%></font></td>
   <td width="19%"><font face="Garamond"><% =rsCustomer("LastName")
%></font></td>
   <td width="26%"><font face="Garamond"> </font></td>
   <td width="15%"><font face="Garamond"></font></td>
   <td width="0%"><font face="Garamond"></font></td>
   <td width="2%"><font face="Garamond"></font></td>
```

```
  </tr>
  <tr>
    <td width="24%"><font face="Garamond">Age :</font></td>
    <td colspan="2"><font face="Garamond"><% =rsCustomer("Age") %></font></td>
    <td width="26%"><font face="Garamond"></font></td>
    <td width="15%"><font face="Garamond"></font></td>
    <td width="0%"><font face="Garamond"></font></td>
    <td width="2%"><font face="Garamond"></font></td>
  </tr>
  <tr>
    <td width="24%"><font face="Garamond">Address : </font></td>
    <td colspan="2"><font face="Garamond"><% =rsCustomer("Address")
%></font></td>
    <td width="26%"><font face="Garamond"></font></td>
    <td width="15%"><font face="Garamond"></font></td>
    <td width="0%"><font face="Garamond"></font></td>
    <td width="2%"><font face="Garamond"></font></td>
  </tr>
  <tr>
    <td width="24%"><font face="Garamond">Province : </font></td>
    <td colspan="2"><font face="Garamond"><% =rsCustomer("Province")
%></font></td>
    <td width="26%"><font face="Garamond"></font></td>
    <td width="15%"><font face="Garamond"></font></td>
    <td width="0%"><font face="Garamond"></font></td>
    <td width="2%"><font face="Garamond"></font></td>
  </tr>
  <tr>
    <td width="24%"><font face="Garamond">RegisterID : </font></td>
    <td colspan="2"><font face="Garamond"><% =rsCustomer("ResidentID")
%></font></td>
    <td width="26%"><font face="Garamond">Driver License Number : </font></td>
```

```
    <td width="15%"><font face="Garamond"><%
 =rsCustomer("DriverLicenseNumber") %></font></td>
    <td width="0%"><font face="Garamond"></font></td>
    <td width="2%"><font face="Garamond"></font></td>
  </tr>
  <tr>
    <td width="24%"><font face="Garamond">Working Palace :</font></td>
    <td colspan="2"><font face="Garamond"><% =rsCustomer("WorkingPlace")
 %></font></td>
    <td width="26%"><font face="Garamond"></font></td>
    <td width="15%"><font face="Garamond"></font></td>
    <td width="0%"><font face="Garamond"></font></td>
    <td width="2%"><font face="Garamond"></font></td>
  </tr>
  <tr>
    <td width="24%"><font face="Garamond">Position :</font></td>
    <td colspan="2"><font face="Garamond"><% =rsCustomer("Position")
 %></font></td>
    <td width="26%"><font face="Garamond"></font></td>
    <td width="15%"><font face="Garamond"></font></td>
    <td width="0%"><font face="Garamond"></font></td>
    <td width="2%"><font face="Garamond"></font></td>
  </tr>
  <tr>
    <td width="24%"><font face="Garamond">Salary : </font></td>
    <td colspan="2"><font face="Garamond"><%
 =formatcurrency(rsCustomer("Salary")) %></font></td>
    <td width="26%"><font face="Garamond">Working Time : </font></td>
    <td width="15%"><font face="Garamond"><% =rsCustomer("WorkingTime")
 %></font></td>
    <td width="0%"><font face="Garamond"></font></td>
    <td width="2%"><font face="Garamond"></font></td>
  </tr>
```

```
</table>
<% end if%>
<br>
<!--#include file = "i_bottom.asp"-->
</body>
</html>
```

File name: Hire purchaser report.asp

```
<MM:BeginLock translatorClass="MM_SSI" type="ssi_include"
depFiles="C:\LEASING\function.asp" orig="%3C!--#include file =
%22function.asp%22--%3E" ><%
function ConvertToMonth(m)
select case m
case 1 :
        m1 = "January"
case 2 :
        m1 = "February"
case 3 :
        m1 = "March"
case 4 :
        m1 = "April"
case 5 :
        m1 = "May"
case 6 :
        m1 = "June"
case 7 :
        m1 = "July"
case 8 :
        m1 = "August"
case 9 :
        m1 = "September"
case 10 :
        m1 = "October"
case 11 :
```

```
        m1 = "November"
case 12 :
        m1 = "December"
end select
        m = m1
end function
function color(x)
if x < 0 then
        response.write "<font color = red>" & x &"</font>"
else
        response.write"<font color = black>" & x & "</font>"
end if
end function
%>
<MM:EndLock> <% mymonth = month(date()) %> <% call
convertToMonth(mymonth) %>
<%
'************** Contect to DB and get the date ******************
set conn = Server.CreateObject("ADODB.Connection")
conn.open "DSN=Leasing"
SQL = "select * from HirePurchaser h , contract c where h.residentID = c.CustomerID
and month(createdDate) = month(date())"
set rs = Server.CreateObject("ADODB.Recordset")
rs.cursorlocation = 3
rs.open sql,conn
%>
<p><font face="Garamond" size="5"><b>Hire purchaser
report(<%=mymonth%>)</b></font></p>
<p><b><font face="Garamond"><%= rs.recordcount%> record(s)
found.</font></b></p>
<table width="80%" border="0" align="center" cellpadding="0" cellspacing="0">
 <tr bgcolor="#333333">
   <td colspan="2">
```

```
      <div align="left"><font face="Garamond"><b><font color="#FFFFFF">Customer
        Name</font></b></font></div>
    </td>
    <td><font color="#CCFFFF" face="Garamond"><b>Contract No.</b></font></td>
    <td width="13%">
      <div align="left"><font face="Garamond"><b><font
color="#FFFFFF">ShopName</font></b></font></div>
    </td>
    <td width="10%">
      <div align="left"><font face="Garamond"><b><font
color="#FFFFFF">Brand</font></b></font></div>
    </td>
    <td width="10%">
      <div align="left"><font face="Garamond"><b><font color="#FFFFFF">Model
</font></b></font></div>
    </td>
    <td width="8%">
      <div align="left"><font face="Garamond"><b><font
color="#FFFFFF">Loan</font></b></font></div>
    </td>
    <td width="19%">
      <div align="center"><font face="Garamond"><b><font
color="#FFFFFF">NoOfPayment</font></b></font></div>
    </td>
    <td width="16%">
      <div align="left"><font face="Garamond"><b><font
color="#FFFFFF">Amount</font></b></font></div>
    </td>
  </tr>
  <% do until rs.eof %>
  <tr bgcolor="#CCCCCC">
    <td width="11%" height="0"><% = rs("FirstName") %>
      <div align="left"></div>
```

```
    </td>
    <td width="11%" height="0"><% = rs("LastName") %></td>
    <td width="13%" height="0">
      <div align="center"><%= rs("ContractNo") %></div>
    </td>
    <td width="13%" height="0"><% = rs("shopName") %>
      <div align="left"></div>
    </td>
    <td width="10%" height="0"><%= rs("Brand")%>
      <div align="left"></div>
    </td>
    <td width="10%" height="0"><% = rs("Model") %>
      <div align="left"></div>
    </td>
    <td width="8%" height="0"><% = formatcurrency(rs("Loan"))%>
      <div align="left"></div>
    </td>
    <td width="19%">
      <div align="center"><% = rs("NoOfPayment")%></div>
    </td>
    <td width="16%" height="0"><% = formatcurrency(rs("Amount"))%>
      <div align="left"></div>
    </td>
  </tr>
  <% rs.movenext %> <% loop %>
</table>
<p> </p>
```

File name : Mfinancial.asp

```
<%
'**************** get data ******************************
strMonth = request("month")
'************************************************************
%>
```

```
<html>
<head>
<title>Finicial Statement</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<!--#include file = "function.asp"-->
<%
'********************* Connect to database **********************
Set conn = Server.CreateObject("ADODB.Connection")
conn.open "DSN=Leasing"
Set rs = Server.CreateObject("ADODB.Recordset")
SQL = "select * from payment where month(dateReceipt) = "& strMonth &""
rs.open SQL,Conn
'************************************************************
'********************* Connnect to contract *********************
Set rsContract = Server.CreateObject("ADODB.Recordset")
SQL = ""
SQL = "select * from contract c,HirePurchaser h where h.ResidentID = c.customerID
and month(createdDate) = "& strMonth &""
rsContract.open SQL,conn
'************************************************************
%>
<% call ConvertTomonth(StrMonth) %>
<body bgcolor="cccccc">
<p><font size="5"><b><font face="Garamond"><%=StrMonth%></font></b></font>
</p>
<p><b>Payment</b></p>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
 <tr bgcolor="#333333">
   <td><font face="Garamond"><b><font
color="#FFFFFF">No</font></b></font></td>
   <td><font face="Garamond"><b><font color="#FFFFFF">Customer
name</font></b></font></td>
```

```
    <td><font face="Garamond"><b><font color="#FFFFFF">Contract no
</font></b></font></td>
    <td><font face="Garamond"><b><font
color="#FFFFFF">Amount</font></b></font></td>
    <td><font face="Garamond"><b><font color="#FFFFFF">Date
Receipt</font></b></font></td>
    <td><font face="Garamond"><b><font
color="#FFFFFF">Note</font></b></font></td>
  </tr>
  <% i = 1 %> <% countPayment = 0 %>
  <tr><% do until rs.eof %>
    <td>
      <div align="center"><% = i %> </div>
    </td>
    <td> <%= rs("CustomerName")%>
      <div align="left"></div>
    </td>
    <td>
      <div align="center"> <% = rs("ContractNo") %> </div>
    </td>
    <td>
      <div align="center"> <% = formatnumber(rs("Amount")) %></div>
    </td>
    <td>
      <div align="center"> <% =rs("DateReceipt") %></div>
    </td>
    <td>
      <div align="center"> <% = rs("Note")%></div>
    </td>
  </tr><% countPayment = countPayment + rs("Amount") %>
  <% rs.movenext %><% i = i +1 %> <% loop %>
</table>
<p> </p>
```

```
<p><font face="Garamond">Total Payment of <%=Mymonth %> is<font
color="#FF0000">
 <b><% = Formatnumber(CountPayment) %> </b></font></font> </p>
<p><b>Loan </b></p>
<table width="100%" border="0" align="center" cellpadding="0" cellspacing="0">
 <tr bgcolor="#333333">
  <td><font color="#FFFFFF" face="Garamond"><b>No</b></font></td>
  <td colspan="2"><font color="#FFFFFF" face="Garamond"><b>Customer
name</b></font></td>
  <td><font color="#FFFFFF" face="Garamond"><b>Contract no </b></font></td>
  <td><font color="#FFFFFF" face="Garamond"><b>Loan</b></font></td>
  <td><font color="#FFFFFF" face="Garamond"><b>No of payment</b></font></td>
  <td><font color="#FFFFFF" face="Garamond"><b>Amount</b></font></td>
  <td><font color="#FFFFFF" face="Garamond"><b>Date</b></font></td>
  <td><font color="#FFFFFF" face="Garamond"><b>Note</b></font></td>
 </tr>
 <% i = 1 %> <% CountLoan = 0 %>
 <tr><% do until rsContract.eof %>
  <td>
   <div align="center"><% = i %> </div>
  </td>
  <td>
   <div align="left"><%= rsContract("FirstName")%> </div>
  </td>
  <td>
   <div align="left"> <% = rsContract("Lastname")%></div>
   </td>
  <td>
   <div align="center"> <% = rsContract("ContractNo") %> </div>
  </td>
  <td>
   <div align="center"><% = formatnumber(rsContract("Loan")) %></div>
  </td>
```

```
    <td>
      <div align="center"><% = rsContract("NoOfPayment")%></div>
    </td>
    <td>
      <div align="center"><% = formatnumber(rsContract("Amount")) %></div>
    </td>
    <td><% =rsContract("CreatedDate") %>
      <div align="center"></div>
    </td>
    <td>
      <div align="center"> <% = rsContract("Note")%></div>
    </td>
  </tr><% countLoan = countLoan + rsContract("Loan") %>
  <% rsContract.movenext %><% i = i +1 %> <% loop %>
</table>
<p> </p>
<p><font face="Garamond">Total Loan of <% = mymonth %> is<font
color="#CC0000">
  </font><b><font face="Garamond"><font color="#ff0000"><% =
Formatnumber(countLoan) %></font></font></b></font></p>
<font face="Garamond"><% countTotal = countPayment - countLoan %> </font>
<p><b><font face="Garamond">Net income is<font color="#FF0000"> <% =
Formatnumber(CountTotal) %></font></font></b></p>
<!--#include file = "i_bottom.asp"-->
</body>
</html>
```

File name : OverallFinancial.asp

```
<!--#include file ="function.asp"-->
<html>
<head>
<title>Finicial Statement</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
```

```
<!--
a:hover {  color: #FFFF33; text-decoration: none}
a:active {  color: #CC0000; text-decoration: none}
a:link {  color: #333333; text-decoration: none}
a:visited {  text-decoration: none}
-->
</style>
</head>
<% on error resume next %>
<body bgcolor="cccccc" vlink="#000000">
<p><font face="Garamond" size="5"><b>Finicial Statement of year <% = year(date())
%>  </b></font></p>
<%
'************************ Script ***********************************
Set Conn = server.CreateObject("ADODB.Connection")
Conn.open "DSN=Leasing"
'************* Jan ************************************************
Set rsJan = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(Amount)as SAmount from payment where month(DateReceipt) = 1
and Year(DateReceipt) = year(Date())"
rsJan.open SQL,conn
Set rsJanLoan = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(loan) as Sloan from contract where month(CreatedDate) = 1 and
Year(CreatedDate) = year(date())"
rsJanLoan.open SQL,conn
'************ Feb *************************************************
Set rsFeb = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(Amount)as SAmount from payment where month(DateReceipt) = 2
and Year(DateReceipt) = year(Date())"
rsFeb.open SQL,conn
Set rsFebLoan = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(loan) as Sloan from contract where month(CreatedDate) = 2 and
Year(CreatedDate) = year(date())"
```

```
rsFebLoan.open SQL,conn
'********** Mar ***********************************************
Set rsMar = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(Amount)as SAmount from payment where month(DateReceipt) = 3
and Year(DateReceipt) = year(Date())"
rsMar.open SQL,conn
Set rsMarLoan = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(loan) as Sloan from contract where month(CreatedDate) = 3 and
Year(CreatedDate) = year(date())"
rsMarLoan.open SQL,conn
'********** Apr ***********************************************
Set rsApr = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(Amount)as SAmount from payment where month(DateReceipt) = 4
and Year(DateReceipt) = year(Date())"
rsApr.open SQL,conn
Set rsAprLoan = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(loan) as Sloan from contract where month(CreatedDate) = 4 and
Year(CreatedDate) = year(date())"
rsAprLoan.open SQL,conn
'********** May ***********************************************
Set rsMay = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(Amount)as SAmount from payment where month(DateReceipt) = 5
and Year(DateReceipt) = year(Date())"
rsMay.open SQL,conn
Set rsMayLoan = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(loan) as Sloan from contract where month(CreatedDate) = 5 and
Year(CreatedDate) = year(date())"
rsMayLoan.open SQL,conn
'********** Jun ***********************************************
Set rsJun = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(Amount)as SAmount from payment where month(DateReceipt) = 6
and Year(DateReceipt) = year(Date())"
rsJun.open SQL,conn
```

```
Set rsJunLoan = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(loan) as Sloan from contract where month(CreatedDate) = 6 and
Year(CreatedDate) = year(date())"
rsJunLoan.open SQL,conn
'********** Jul ************************************************
Set rsJul = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(Amount)as SAmount from payment where month(DateReceipt) = 7
and Year(DateReceipt) = year(Date())"
rsJul.open SQL,conn
Set rsJulLoan = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(loan) as Sloan from contract where month(CreatedDate) = 7 and
Year(CreatedDate) = year(date())"
rsJulLoan.open SQL,conn
'***********Aug ************************************************
Set rsAug = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(Amount)as SAmount from payment where month(DateReceipt) = 8
and Year(DateReceipt) = year(Date())"
rsAug.open SQL,conn
Set rsAugLoan = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(loan) as Sloan from contract where month(CreatedDate) = 8 and
Year(CreatedDate) = year(date())"
rsAugLoan.open SQL,conn
'********* Sep ************************************************
Set rsSep = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(Amount)as SAmount from payment where month(DateReceipt) = 9
and Year(DateReceipt) = year(Date())"
rsSep.open SQL,conn
Set rsSepLoan = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(loan) as Sloan from contract where month(CreatedDate) = 9 and
Year(CreatedDate) = year(date())"
rsSepLoan.open SQL,conn
'********** Oct ************************************************
Set rsOct = Server.CreateObject("ADODB.Recordset")
```

```
SQL = "select sum(Amount)as SAmount from payment where month(DateReceipt) =
10 and Year(DateReceipt) = year(Date())"
rsOct.open SQL,conn
Set rsOctLoan = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(loan) as Sloan from contract where month(CreatedDate) = 10 and
Year(CreatedDate) = year(date())"
rsOctLoan.open SQL,conn
'********* Nov ************************************************
Set rsNov = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(Amount)as SAmount from payment where month(DateReceipt) =
11 and Year(DateReceipt) = year(Date())"
rsNov.open SQL,conn
Set rsNovLoan = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(loan) as Sloan from contract where month(CreatedDate) = 11 and
Year(CreatedDate) = year(date())"
rsNovLoan.open SQL,conn
'********* Dec ************************************************
Set rsDec = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(Amount)as SAmount from payment where month(DateReceipt) =
12 and Year(DateReceipt) = year(Date())"
rsDec.open SQL,conn
Set rsDecLoan = Server.CreateObject("ADODB.Recordset")
SQL = "select sum(loan) as Sloan from contract where month(CreatedDate) = 12 and
Year(CreatedDate) = year(date())"
rsDecLoan.open SQL,conn
%>
<% TotalPayment = 0 %>
<% TotalLoan = 0 %>
<table width="75%" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr bgcolor="#333333">
    <td height="19"><b><font face="Garamond"><font
color="#FFFFFF">Month</font></font></b></td>
```

```
    <td height="19"><font face="Garamond"><b><font
 color="#FFFFFF">Payment</font></b></font></td>
    <td height="19"><font face="Garamond"><b><font
 color="#FFFFFF">Loan</font></b></font></td>
    <td height="19"><font face="Garamond"><b><font color="#FFFFFF">Payment -
 Loan</font></b></font></td>
  </tr>
  <tr>
    <td><font color="#000099"><b><a
 href="MFinance.asp?month=1">January</a></b></font></td>
    <%  strSloan = rsJanLoan("Sloan")
          strSAmount = rsJan("SAMount")
          if IsEmpty(StrSloan) or IsNull(strSLoan) then
                StrSloan = 0
          end if
          if IsEmpty(StrSAmount) or IsNull(strSAmount) then
                StrSAmount = 0
          end if
 %> <% totalPayment = TotalPayment + strSAmount %> <% TotalLoan = TotalLoan +
 strSLoan %>
    <td><% =formatnumber(strSAmount) %></td>
    <td><% = formatnumber(StrSloan)%></td>
    <% StrNov = strSAmount-StrSloan %>
    <td><% = color(formatnumber(StrNov)) %></td>
  </tr>
  <tr>
    <td><font color="#000099"><b><a
 href="MFinance.asp?month=2">February</a></b></font></td>
    <%  strSloan = rsFebLoan("Sloan")
          strSAmount = rsFeb("SAMount")
          if IsEmpty(StrSloan) or IsNull(strSLoan) then
                StrSloan = 0
          end if
```

```
        if IsEmpty(StrSAmount) or IsNull(strSAmount) then
                StrSAmount = 0
        end if
%> <% totalPayment = TotalPayment + strSAmount %> <% TotalLoan = TotalLoan +
strSLoan %>
   <td><% =formatnumber(strSAmount) %></td>
   <td><% = formatnumber(StrSloan)%></td>
   <% StrNov = strSAmount-StrSloan %>
   <td><% = color(formatnumber(StrNov)) %></td>
 </tr>
 <tr>
   <td><font color="#000099"><b><a
href="MFinance.asp?month=3">March</a></b></font></td>
   <%  strSloan = rsMarLoan("Sloan")
        strSAmount = rsMar("SAMount")
        if IsEmpty(StrSloan) or IsNull(strSLoan) then
                StrSloan = 0
        end if
        if IsEmpty(StrSAmount) or IsNull(strSAmount) then
                StrSAmount = 0
        end if
%> <% totalPayment = TotalPayment + strSAmount %> <% TotalLoan = TotalLoan +
strSLoan %>
   <td><% =formatnumber(strSAmount) %></td>
   <td><% = formatnumber(StrSloan)%></td>
   <% StrNov = strSAmount-StrSloan %>
   <td><% = color(formatnumber(StrNov)) %></td>
 </tr>
 <tr>
   <td><font color="#000099"><b><a
href="MFinance.asp?month=4">April</a></b></font></td>
   <%  strSloan = rsAprLoan("Sloan")
        strSAmount = rsApr("SAMount")
```

```
        if IsEmpty(StrSloan) or IsNull(strSLoan) then
                StrSloan = 0
        end if
        if IsEmpty(StrSAmount) or IsNull(strSAmount) then
                StrSAmount = 0
        end if

%> <% totalPayment = TotalPayment + strSAmount %> <% TotalLoan = TotalLoan +
strSLoan %>
   <td><% =formatnumber(strSAmount) %></td>
   <td><% = formatnumber(StrSloan)%></td>
   <% StrNov = strSAmount-StrSloan %>
   <td><% = color(formatnumber(StrNov)) %></td>
 </tr>
 <tr>
   <td height="22"><font color="#000099"><b><a
href="MFinance.asp?month=5">May</a></b></font></td>
   <%  strSloan = rsMayLoan("Sloan")
        strSAmount = rsMay("SAMount")
        if IsEmpty(StrSloan) or IsNull(strSLoan) then
                StrSloan = 0
        end if
        if IsEmpty(StrSAmount) or IsNull(strSAmount) then
                StrSAmount = 0
        end if

%> <% totalPayment = TotalPayment + strSAmount %> <% TotalLoan = TotalLoan +
strSLoan %>
   <td><% =formatnumber(strSAmount) %></td>
   <td><% = formatnumber(StrSloan)%></td>
   <% StrNov = strSAmount-StrSloan %>
   <td><% = color(formatnumber(StrNov)) %></td>
 </tr>
```

```
  <tr>
   <td><font color="#000099"><b><a
href="MFinance.asp?month=6">June</a></b></font></td>
   <%  strSloan = rsJunLoan("Sloan")
       strSAmount = rsJun("SAMount")
       if IsEmpty(StrSloan) or IsNull(strSLoan) then
            StrSloan = 0
       end if
       if IsEmpty(StrSAmount) or IsNull(strSAmount) then
            StrSAmount = 0
       end if

%> <% totalPayment = TotalPayment + strSAmount %> <% TotalLoan = TotalLoan +
strSLoan %>
   <td><% =formatnumber(strSAmount) %></td>
   <td><% = formatnumber(StrSloan)%></td>
   <% StrNov = strSAmount-StrSloan %>
   <td><% = color(formatnumber(StrNov)) %></td>
 </tr>
 <tr>
   <td><font color="#000099"><b><a
href="MFinance.asp?month=7">July</a></b></font></td>
   <%  strSloan = rsJulLoan("Sloan")
       strSAmount = rsJul("SAMount")
       if IsEmpty(StrSloan) or IsNull(strSLoan) then
            StrSloan = 0
       end if
       if IsEmpty(StrSAmount) or IsNull(strSAmount) then
            StrSAmount = 0
       end if

%> <% totalPayment = TotalPayment + strSAmount %> <% TotalLoan = TotalLoan +
strSLoan %>
```

```
    <td><% =formatnumber(strSAmount) %></td>
    <td><% = formatnumber(StrSloan)%></td>
    <% StrNov = strSAmount-StrSloan %>
    <td><% = color(formatnumber(StrNov)) %></td>
  </tr>
  <tr>
    <td><font color="#000099"><b><a
href="MFinance.asp?month=8">August</a></b></font></td>
    <%  strSloan = rsAugLoan("Sloan")
        strSAmount = rsAug("SAMount")
        if IsEmpty(StrSloan) or IsNull(strSLoan) then
            StrSloan = 0
        end if
        if IsEmpty(StrSAmount) or IsNull(strSAmount) then
            StrSAmount = 0
        end if

%> <% totalPayment = TotalPayment + strSAmount %> <% TotalLoan = TotalLoan +
strSLoan %>
    <td><% =formatnumber(strSAmount) %></td>
    <td><% = formatnumber(StrSloan)%></td>
    <% StrNov = strSAmount-StrSloan %>
    <td><% = color(formatnumber(StrNov)) %></td>
  </tr>
  <tr>
    <td><font color="#000099"><b><a
href="MFinance.asp?month=9">September</a></b></font></td>
    <%  strSloan = rsSepLoan("Sloan")
        strSAmount = rsSep("SAMount")
        if IsEmpty(StrSloan) or IsNull(strSLoan) then
            StrSloan = 0
        end if
        if IsEmpty(StrSAmount) or IsNull(strSAmount) then
```

```
                StrSAmount = 0
            end if

%> <% totalPayment = TotalPayment + strSAmount %> <% TotalLoan = TotalLoan +
strSLoan %>
    <td><% =formatnumber(strSAmount) %></td>
    <td><% = formatnumber(StrSloan)%></td>
    <% StrNov = strSAmount-StrSloan %>
    <td><% = color(formatnumber(StrNov)) %></td>
  </tr>
  <tr>
    <td><font color="#000099"><b><a
href="MFinance.asp?month=10">October</a></b></font></td>
    <%  strSloan = rsOctLoan("Sloan")
          strSAmount = rsOct("SAMount")
          if IsEmpty(StrSloan) or IsNull(strSLoan) then
                StrSloan = 0
          end if
          if IsEmpty(StrSAmount) or IsNull(strSAmount) then
                StrSAmount = 0
          end if

%> <% totalPayment = TotalPayment + strSAmount %> <% TotalLoan = TotalLoan +
strSLoan %>
    <td><% =formatnumber(strSAmount) %></td>
    <td><% = formatnumber(StrSloan)%></td>
    <% StrNov = strSAmount-StrSloan %>
    <td><% = color(formatnumber(StrNov)) %></td>
  </tr>
  <tr>
    <td><font color="#000099"><b><a
href="MFinance.asp?month=11">November</a></b></font></td>
    <%  strSloan = rsNovLoan("Sloan")
```

```
        strSAmount = rsNov("SAMount")
        if IsEmpty(StrSloan) or IsNull(strSLoan) then
                StrSloan = 0
        end if
        if IsEmpty(StrSAmount) or IsNull(strSAmount) then
                StrSAmount = 0
        end if

%> <% totalPayment = TotalPayment + strSAmount %> <% TotalLoan = TotalLoan +
strSLoan %>
   <td><% =formatnumber(strSAmount) %></td>
   <td><% = formatnumber(StrSloan)%></td>
   <% StrNov = strSAmount-StrSloan %>
   <td><% = color(formatnumber(StrNov)) %></td>
  </tr>
  <tr>
   <td><font color="#000099"><b><a
href="MFinance.asp?month=12">December</a></b></font></td>
   <%  strSloan = rsDecLoan("Sloan")
        strSAmount = rsDec("SAMount")
        if IsEmpty(StrSloan) or IsNull(strSLoan) then
                StrSloan = 0
        end if
        if IsEmpty(StrSAmount) or IsNull(strSAmount) then
                StrSAmount = 0
        end if

%> <% totalPayment = TotalPayment + strSAmount %> <% TotalLoan = TotalLoan +
strSLoan %>
   <td><% =formatnumber(strSAmount) %></td>
   <td><% = formatnumber(StrSloan)%></td>
   <% StrNov = strSAmount-StrSloan
```

File name : Payment detail

```
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#CCCCCC">
<p align="center"><font size="4" face="Garamond"><b>Payment Detail
</b></font></p>
<!--#include file ="i_search.asp"-->
<%StrCNo = rsContract("ContractNo")%>
<%
'************* Connect DB **************************
set conn = Server.CreateObject("ADODB.Connection")
conn.open "DSN=Leasing"
SQL = "select * from contract where contractNo = "& strCno & ""
set rs = conn.execute(SQL)
'*************************************************
strCustomerID = rs("customerID")
strContractNo = rs("ContractNo")
'************* Connect to HirePurchaser ************
SQL = ""
SQL = SQL & "select firstname , lastname from hirepurchaser where residentID = " &
strCustomerID & ""
set rsCustomer = conn.execute(SQL)
'*************************************************
'************ Connect to Payment ******************
SQL = ""
SQL = SQL & " select * from payment where contractNo = "& strContractNo &""
set rsPayment = conn.execute(SQL)
'*************************************************
%>
<table width="80%" border="0" align="center" cellpadding="0" cellspacing="0">
```

```
  <tr>
    <td colspan="3"><b><font face="Garamond">Customer name : <% =
 rsCustomer("FirstName")%></font></b></td>
    <td colspan="2"><b><font face="Garamond"><%= rsCustomer("LastName")
 %></font></b><b></b></td>
    <td width="36%"><font face="Garamond"><b>Contract number :
 <%=rs("ContractNo")%></b></font></td>
  </tr>
  <tr bgcolor="#333333">
    <td width="7%" bgcolor="#CCCCCC"> </td>
    <td width="23%" bgcolor="#CCCCCC"> </td>
    <td width="6%" bgcolor="#CCCCCC"> </td>
    <td width="23%" bgcolor="#CCCCCC"> </td>
    <td colspan="2" bgcolor="#CCCCCC"> </td>
  </tr>
  <% if not rsPayment.eof then %>
  <tr bgcolor="#333333">
    <td width="7%">
      <div align="center"><font face="Garamond"><b><font
 color="#FFFFFF">No.</font></b></font></div>
    </td>
    <td width="23%">
      <div align="center"><font face="Garamond"><b><font
 color="#FFFFFF">Payment
        date</font></b></font></div>
    </td>
    <td width="6%">
      <div align="center"><font face="Garamond"><b><font
 color="#FFFFFF">Amount</font></b></font></div>
    </td>
    <td width="23%">
      <div align="center"><font face="Garamond"><b><font
 color="#FFFFFF">Collector</font></b></font></div>
```

```
    </td>
    <td colspan="2">
      <div align="center"><font face="Garamond"><b><font
color="#FFFFFF">Note</font></b></font></div>
    </td>
  </tr>
<% count = 0 %>
  <% do until rsPayment.eof %>
  <tr bgcolor="#CCCCCC">
    <td width="7%">
      <div align="center"><font face="Garamond"><%= rsPayment("NoOfPayment")
%></font></div>
    </td>
    <td width="23%">
      <div align="center"><font face="Garamond"><% = rsPayment("DateReceipt")
%></font></div>
    </td>
    <td width="6%">
      <div align="center"><font face="Garamond"><% =
formatcurrency(rsPayment("Amount")) %><b></b></font></div>
    </td>
    <td width="23%">
      <div align="center"><font face="Garamond"><% = rsPayment("Collector")
%></font></div>
    </td>
    <td colspan="2">
      <div align="center"><font face="Garamond"><% = rsPayment("Note")
%></font></div>
    </td>
  </tr>
          <% count = rsPayment("Amount") + count %>
  <% rsPayment.movenext %> <% loop %>
  <tr>
```

```
    <td colspan="5"> </td>
    <td width="36%"> </td>
  </tr>
  <tr>
    <td colspan="5"><font face="Garamond"><b>Total payment = <% =
formatcurrency(count) %></b></font></td>
    <td width="36%"> </td>
  </tr>
</table>
<% end if %>
<% end if%>
<p> </p>
<!--#include file = "i_bottom.asp"-->
</body>
</html>
```

File name : Payment report.asp

```
<html>
<head>
<title>Payment Report</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<!--#include file = "function.asp"-->
<%
'**************** Connect DB ************************************
set conn = Server.CreateObject("ADODB.Connection")
conn.open "DSN=Leasing"
set rs = Server.CreateObject("ADODB.Recordset")
SQL = " select * from payment"
rs.cursorlocation = 3
rs.open SQL,Conn
'****************************************************************
%>
<body bgcolor="#CCCCCC">
```

```
<% Mymonth = month(date()) %> <% call ConvertToMonth(Mymonth) %>
<p><font size="5"><b><font face="Garamond">Payment
Report(<%=Mymonth%>)</font></b></font>
</p>
<p><b><font face="Garamond"><% = rs.recordcount %> record found(s) in
<%=mymonth %></font></b></p>
<table width="80%" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr bgcolor="#333333">
    <td><b><font color="#FFFFFF" face="Garamond">No. </font></b></td>
    <td><b><font color="#FFFFFF" face="Garamond">Date</font></b></td>
    <td><b><font color="#FFFFFF" face="Garamond">Receipt number
:</font></b></td>
    <td><b><font color="#FFFFFF" face="Garamond">Customer
name</font></b></td>
    <td><b><font color="#FFFFFF" face="Garamond">Contract No</font></b></td>
    <td><b><font color="#FFFFFF" face="Garamond">No of payment</font></b></td>
    <td><b><font color="#FFFFFF" face="Garamond">Amount</font></b></td>
    <td><b><font color="#FFFFFF" face="Garamond">Note</font></b></td>
    <td><b><font color="#FFFFFF" face="Garamond">Colloctor</font></b></td>
  </tr>
  <% i = 1 %><% count = 0 %>
  <tr> <% do until rs.eof %>
    <td><% =i %></td>
    <td><% = rs("DateReceipt") %></td>
    <td><% = rs("Receiptno") %></td>
    <td><% =rs("CustomerName") %></td>
    <td><% = rs("ContractNo") %></td>
    <td><% = rs("noOfPayment") %></td>
    <td><% = formatcurrency(rs("AMount"))%></td>
    <td><% = rs("note")%></td>
    <td><% = rs("Collector")%></td>
  </tr><% i = i + 1 %> <% count = count + rs("amount") %>
  <% rs.movenext %> <%loop%>
```

```
</table>
<p align="center"><font face="Garamond"><b>Total = <% = formatcurrency(count)
%>

 </b> </font></p>
<!--#include file = "i_bottom.asp"-->
</body>
</html>
```

File name : Search.asp

```
<html>
<head>
<title>CheckHistory</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<% strResidentID = request("ResidentID")%>
<body bgcolor="#CCCCCC">
<div align="center"><b><font size="5" face="Garamond">Check the customer history
 record</font></b> </div>
<% if strResidentID = "" then %>
<form name = "f" method="post" action="">
 <table width="80%" border="0" align="center">
  <tr>
   <td width="52%">
    <div align="right"><b><font face="Garamond">Please enter the ResidentID
     : </font></b></div>
   </td>
   <td width="48%"> <b><font face="Garamond">
    <input type="text" name="ResidentID">
    </font></b></td>
  </tr>
  <tr>
   <td width="52%"> </td>
   <td width="48%"> </td>
```

```
    </tr>
  </table>
  <div align="center">
    <input type="button" name="btnSubmit" value="Search">
  </div>
</form>
<% end if%> <% if strResidentID <> "" then %> <%
'***************** Find Data ************************
set conn = server.createObject("ADODB.Connection")
conn.open "DSN=Leasing"
set rs = server.createObject("ADODB.recordset")
rs.cursorlocation = 3
SQL = "select * from HirePurchaser h, contract c where h.residentID = c.customerID
and h.residentID = "&strResidentID&""
rs.open SQL,conn
'************************************************
%>
<% if rs.eof then %>
<div align="center">
  <p> </p>
  <p> </p>
  <p> </p>
  <p><font color="#FF0000" face="Garamond" size="5"><b>No record found for this
    Resident number</b></font> <% end if %> <% if not rs.eof then%>
<%strContractNo = rs("ContractNo") %>
  </p>
</div>
<table width="80%" border="0" align="center">
  <tr>
    <td width="22%"><b><font face="Garamond">Resident ID </font></b></td>
    <td colspan="2"><b><font
face="Garamond"><%=rs("residentID")%></font></b></td>
    <td width="47%"><font face="Garamond"></font></td>
```

```
    <td width="7%"><font face="Garamond"></font></td>
  </tr>
  <tr>
    <td width="22%"><font face="Garamond"></font></td>
    <td colspan="2"><font face="Garamond"></font></td>
    <td width="47%"><font face="Garamond"></font></td>
    <td width="7%"><font face="Garamond"></font></td>
  </tr>
  <tr>
    <td colspan="5" bgcolor="#333333"><b><font face="Garamond"
color="#FFFFFF">Customer
      Information</font></b></td>
  </tr>
  <tr>
    <td width="22%"><font face="Garamond"></font></td>
    <td colspan="2"><font face="Garamond"></font></td>
    <td width="47%"><font face="Garamond"></font></td>
    <td width="7%"><font face="Garamond"></font></td>
  </tr>
  <tr>
    <td width="22%"><font face="Garamond">HirePurchaser :</font></td>
    <td width="2%"><font face="Garamond"><% = rs("Firstname")%> </font></td>
    <td width="22%"><font face="Garamond"><% = rs("Lastname") %></font></td>
    <td width="47%"><font face="Garamond"></font></td>
    <td width="7%"><font face="Garamond"></font></td>
  </tr>
  <tr>
    <td width="22%"><font face="Garamond">Address</font></td>
    <td colspan="2"><font face="Garamond"><% = rs("Address")%></font></td>
    <td width="47%"><font face="Garamond">Working Place :</font></td>
    <td width="7%"><font face="Garamond"><% =rs("WorkingPlace")%></font></td>
  </tr>
  <tr>
```

```
    <td width="22%"><font face="Garamond">Tel</font></td>
    <td colspan="2"><font face="Garamond"><%=rs("tel")%></font></td>
    <td width="47%"><font face="Garamond">Position :</font></td>
    <td width="7%"><font face="Garamond"><% = rs("position")%></font></td>
  </tr>
  <tr>
    <td width="22%"><font face="Garamond"></font></td>
    <td colspan="2"><font face="Garamond"></font></td>
    <td width="47%"><font face="Garamond"></font></td>
    <td width="7%"><font face="Garamond"></font></td>
  </tr>
  <tr>
    <td colspan="5" bgcolor="#333333"><b><font face="Garamond"
color="#FFFFFF">Contract
      Status :</font></b></td>
  </tr>
  <tr>
    <td width="22%"><font face="Garamond"></font></td>
    <td colspan="2"><font face="Garamond"></font></td>
    <td width="47%"><font face="Garamond"></font></td>
    <td width="7%"><font face="Garamond"></font></td>
  </tr>
  <tr>
    <td width="22%"><font face="Garamond">Contract number :</font></td>
    <td colspan="2"><font face="Garamond"><% = strContractno%></font></td>
    <td width="47%"><font face="Garamond">Created Date :</font></td>
    <td width="7%"><font face="Garamond"></font></td>
  </tr>
  <tr>
    <td width="22%"><font face="Garamond">Loan</font></td>
    <td colspan="2"><font face="Garamond"><% = rs("loan")%></font></td>
    <td width="47%"><font face="Garamond">No of payment :</font></td>
    <td width="7%"><font face="Garamond"><%= rs("noofpayment")%></font></td>
```

```
 </tr>
 <tr>
  <td width="22%"><font face="Garamond">Amount :</font></td>
  <td colspan="2"><font face="Garamond"><% = rs("Amount")%></font></td>
  <td width="47%"><font face="Garamond">Due Date :</font></td>
  <td width="7%"><font face="Garamond"><% = rs("DueDate")%></font></td>
 </tr>
 <tr>
  <td width="22%"><font face="Garamond">Brand </font></td>
  <td colspan="2"><font face="Garamond"><% = rs("Brand")%></font></td>
  <td width="47%"><font face="Garamond">Model</font></td>
  <td width="7%"><font face="Garamond"><% = rs("Model")%></font></td>
 </tr>
 <tr>
  <td width="22%"><font face="Garamond">Register number</font></td>
  <td colspan="2"><font face="Garamond"><% =
rs("Registernumber")%></font></td>
  <td width="47%"><font face="Garamond">Province :</font></td>
  <td width="7%"><font face="Garamond"><% = rs("province")%></font></td>
 </tr>
 <tr>
  <td width="22%"><font face="Garamond">Closed Account</font></td>
  <td colspan="2"><font face="Garamond"><%=ClosedAccount%></font></td>
  <td width="47%"><font face="Garamond"></font></td>
  <td width="7%"><font face="Garamond"></font></td>
 </tr>
 <tr>
  <td width="22%"><font face="Garamond"></font></td>
  <td colspan="2"><font face="Garamond"></font></td>
  <td width="47%"><font face="Garamond"></font></td>
  <td width="7%"><font face="Garamond"></font></td>
 </tr>
 <tr>
```

```
    <td colspan="5" bgcolor="#333333"><b><font face="Garamond"
color="#FFFFFF">Payment
      record</font></b></td>
  </tr>
  <tr>
    <td width="22%"><font face="Garamond"></font></td>
    <td colspan="2"><font face="Garamond"></font></td>
    <td width="47%"><font face="Garamond"></font></td>
    <td width="7%"><font face="Garamond"></font></td>
  </tr>
</table>

<div align="center"><font face="Garamond"><b><%
'********************* payment**************************
set rsPayment = Server.CreateObject("ADODB.Recordset")
SQL = "select * from payment where contractno = "& strcontractno &""
rsPayment.open SQL,conn
'*****************************************************
%> <% if rsPayment.eof then %> No payment Record</b> <% else %> </font> </div>
<table width="80%" border="0" cellpadding="0" cellspacing="0" align="center">
 <tr bgcolor="#CCCCCC">
   <td> </td>
   <td> </td>
   <td> </td>
   <td> </td>
   <td> </td>
   <td> </td>
   <td> </td>
 </tr>
 <tr bgcolor="#333333">
   <td><b><font face="Garamond" color="#FFFFFF">No.</font></b></td>
   <td><b><font face="Garamond" color="#FFFFFF">Date</font></b></td>
   <td><b><font face="Garamond" color="#FFFFFF">No of payment</font></b></td>
```

```
    <td><b><font face="Garamond" color="#FFFFFF">Amount</font></b></td>
    <td><b><font face="Garamond" color="#FFFFFF">Collector</font></b></td>
    <td><b><font face="Garamond" color="#FFFFFF">Note</font></b></td>
    <td><b><font face="Garamond" color="#FFFFFF"></font></b></td>
  </tr>
  <% i = 1 %> <% do until rsPayment.eof %>
  <tr>
    <td><font face="Garamond"><% = i %></font></td>
    <td><font face="Garamond"><% = rsPayment("DateReceipt")%></font></td>
    <td>
      <div align="center"><font face="Garamond"> <% =
rsPayment("NoOfPayment")%></font></div>
    </td>
    <td><font face="Garamond"> <% =
formatcurrency(rsPayment("Amount"))%></font></td>
    <td><font face="Garamond"> <% = rsPayment("collector")%></font></td>
    <td><font face="Garamond"> <% = rsPayment("note")%></font></td>
    <td><font face="Garamond"></font></td>
  </tr>
  <% i = i + 1 %> <% rsPayment.movenext %> <% loop %>
</table>
<% end if %>
<% end if %>
<% end if %>
<p> </p>
<!--#include file = "i_bottom.asp"-->
</body>
</html>
<script language="vbScript">

sub btnSubmit_OnClick()

if f.ResidentID.value = "" then
```

```
        msgbox "Please enter the residentID"
else
        f.submit()
end if
end sub
</script>
```

File name : Unqualify.asp

```
<html>
<head>
<title>Unqualified Customer Report</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<!--#include file ="function.asp"-->
<%
'****************Connect to Database ************

Set conn = Server.CreateObject("ADODB.Connection")
conn.open  "DSN=Leasing"
SQL = "select * from tempDB where verify = 'no' and month(createdDate) =
month(date())"
Set rs = conn.execute (SQL)

'************************************************
%>
<%mymonth = month(date())%>
<%  p = convertTOMonth(mymonth) %>
<body bgcolor="#CCCCCC">
<p><b><font face="Garamond" size="5">Unqualified customer
report(<%=mymonth%>)</font></b>
</p>
<table width="80%" border="0" align="center">
 <tr bgcolor="#999999">
```

```
   <td colspan="2" bgcolor="#333333"><b><font color="#FFFFFF"
face="Garamond">Customer
     Name</font></b></td>
   <td colspan="2" bgcolor="#333333"><b><font color="#FFFFFF"
face="Garamond">Guarantor
     Name</font></b></td>
  </tr>
  <% i = 0 %> <% do until rs.eof %>
  <tr>
   <td><font face="Garamond"><%=rs("CFirstName")%> </font></td>
   <td><font face="Garamond"><%=rs("CLastName")%></font></td>
   <td><font face="Garamond"><%=rs("GFirstName")%> </font></td>
   <td><font face="Garamond"><%=rs("GLastName")%></font></td>
  </tr>
  <% i = i + 1 %> <% rs.movenext %> <% loop %>
</table>
<p align="center"><font face="Garamond"><%=i%> record(s) found </font></p>
<!--#include file ="i_bottom.asp"-->
</body>
</html>
```

BIBLIOGRAPHY

1. Date, C. J. An Introduction to Database Systems. MA: Addison-Wesley, 1995.

2. Eliason, Alan L. System Development: Analysis, Design, and Implementation. Boston: Little, Brown, 1987.

3. FitzGerald, J. and Ardre F. FitzGerald. Fundamentals of System Analysis.New York: John Wiley & Sons, Inc., 1987.

4. Korth, F. Henry and Abraham Silberschats, Database System Concepts.New York: McGraw-Hill International, Inc., 1991.

5. Laudon, Kenneth C. and Jane Price Laudon. Management Information Systems: A Contemporary Perspective. New York: Macmillan, 1988.

6. Loomis, Mary E. S. Data Management and File Structures, Second Edition. London: Prentice-Hall International, 1989.

7. Page-Jones, Meilir. The Practical Guide to Structured System Design. London: Yourdon Press, 1989.

8. Senn, James A. Analysis & Design of Information Systems. New York: McGraw-Hill, 1989.

9. Whitten, Jeffery L. and Lonnie D. Bentley. System Analysis and Design Methods. Burr Ridge: McGraw Hill, 1998.

10. Yourdon, Edward. Modern Structure Analysis. New York: Prentice Hall, 1989.