Management Information System for Yunnan Airlines of China

by

Mr. Yong Li

A Final Report of the Six-Credit Course
CS 6998 - CS 6999 System Development Project

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

November 2003

St. Gabriel's Library, Au

Management Information System for Yunnan Airlines of China

by
Mr. Yong Li

A Final Report of the Six-Credit Course
CS 6998-CS 6999 System Development Project

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

November 2003

Project Title	Management Information System for Yunnan Airlines of China
Name	Mr. Yong Li
Project Advisor	Air Marshal Dr. Chulit Meesajjee
Academic Year	November 2003

The Graduate School of Assumption University has approved this final report of the three-credit course, CS 6998 – CS 6999 System Development Project, submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Information Systems.

Approval Committee:

AM chulit Meesajjee.

(Air Marshal Dr. Chulit Meesajjee)
Dean and Advisor

(Prof.Dr. Srisakdi Charmonman)
Chairman

Vichit Avatch

(Asst.Prof.Dr. Vichit Avatchanakorn)
Member

(Assoc.Prof. Somchai Thayarnyong)
CHE Representative

November 2003

ABSTRACT

Nowadays, computer based information systems have a very important role is all kinds of businesses. This technology improves the competitive advantage of the company, also, it can help reduce costs, and improve the organization's performance.

This project presents the analysis, design, and implementation of the Management Information system of Yunnan Airlines of China. The old system is a manual system where staff uses computers and handwork to process jobs. In fact, the old system is a semi-computerized system.

For saving time and reducing efficiency problems, the proposed system is developed a network structure to capture data and information. Eventually, the tickets sub-division may share data from the cabin and (or) airlines sub-division. The speed of transfering and processing data will be increased; the efficiency will also be boosted. At the same time, the operations manager can make good decisions in time, based on sharing the data and timely reports.

The DFD, ERD, and Feasibility Analysis approaches have been used in this project. Microsoft Visual Basic 6.0 is used to program in this project.

This project will help Yunnan Airlines boost its operations. Especially, it will improve the efficiency in cabin managment, airlines managing, scheduled flights and ticket management.

ACKNOWLEDGMENTS

There are many people who have made contributions to this project. The writer would like to acknowledge their efforts and thank them for their contributions.

First, He would hereby like to express Air Marshal Dr. Chulit Meesajjee, his advisor for his assistance, guidance, and valuable suggestions in preparing and developing the project. He would like to thank all his lecturers for the knowledge used in developing the project. He would also like to thank his classmates for their kind assistance and encouragement.

Finally He would like to give a special thanks to his parents, his wife and his daughter for their love, encouragement, and valuable advice.

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

St. Gabriel's Library, Au

TABLE OF CONTENTS

Chapter		Page
ABSTRACT		i
ACKNOWLEDGMENTS		ii
LIST OF FIGURES		v
LIST OF TABLES		vii
I.	INTRODUCTION	1
1.1	Background of Project	1
1.2	Objectives of Project	1
1.3	Scope of Project	2
1.4	Deliverables	3
1.5	Project Plan (Include Gantt Chart)	3
II.	THE EXISTING SYSTEM	5
2.1	Background of the Company	5
2.2	Existing Organization Function	5
2.3	Existing Computer System	10
2.4	Current Problems of Existing System	10
III.	THE PROPOSED SYSTEM	11
3.1	System Specification	11
3.2	Proposed Function	11
3.3	Hardware, Software and People ware Specification	15
3.4	Security and Control	17
3.5	Network Configuration	19
3.6	Cost and Benefit Analysis	21

ASSUMPTION UNIVERSITY OF THAILAND
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Chapter			Page
IV.	SYSTEM IMPLEMENTATION		30
	4.1	Overview of Project Implementation	30
	4.2	Schedule of the Project	30
	4.3	Conversion	32
V.	CONCLUSIONS AND RECOMMENDATIONS		33
	5.1	Conclusions	33
	5.2	Recommendations	34
APPENDIX	A	FEASIBILITY ANALYSIS	35
APPENDIX	B	DATA FLOW DIAGRAM	42
APPENDIX	C	STRUCTURE CHART	45
APPENDIX	D	ENTITY RELATIONSHIP DIAGRAM	49
APPENDIX	E	TABLE PROPERTIES	52
APPENDIX	F	DATABASE IMPLEMENT	57
APPENDIX	G	DATA DICTIONARY	60
APPENDIX	H	PROCESS SPECIFICATION	63
APPENDIX	I	USER INTERFACE	68
APPENDIX	J	OUTPUT REPORTS	78
APPENDIX	K	SOURCE CODE	82
BIBLIOGRAPHY			153

ASSUMPTION UNIVERSITY OF THAILAND
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

LIST OF FIGURES

Figure		Page
1.1	Project Plan for MIS of Yunnan Airlines of China.	4
2.1	Organization Chart of the Yunnan Airlines of China.	6
2.2	The Context Diagram of Yunnan Airlines of China.	8
2.3	The Data Flow Diagram of Yunnan Airlines of China.	9
3.1	Network Configuration of Proposed System.	20
3.2	Break-even Chart for the System.	25
3.3	Payback Chart for Project.	28
A.1	Payback Period for Candidate 1.	39
A.2	Payback Period for Candidate 2.	40
A.3	Payback Period for Candidate 3.	41
B.1	Data Flow Diagram Level 0 for MIS of Yunnan Airlines of China.	42
B.2	Data Flow Diagram Level1 Create Report.	43
B.3	Data Flow Diagram Level1 Passenger Register.	44
C.1	Structure Chart of Login.	45
C.2	Structure Chart of Creating Scheduled Flight.	46
C.3	Structure Chart of Booking.	47
C.4	Structure Chart of Generation Report.	48
D.1	Entity Relationship Diagram of Proposed System.	49
D.2	Key Base Entity Relationship Diagram of Proposed System.	50
D.3	Fully Attribute Entity Relationship Diagram of Proposed System.	51
I.1	Login Interface.	68
I.2	Main Interface.	68
I.3	Managing Airliner Interface.	69

Figure		Page
I.4	Input the Airliner Information.	69
I.5	Inquiry the Airliners Information.	70
I.6	Managing the Cabin Interface.	70
I.7	Input the Cabin Information.	71
I.8	Managing Scheduled Flight Interface.	71
I.9	Input the Scheduled Flight Information.	72
I.10	Inquiry Scheduled Flight Information.	72
I.11	Managing Passenger Type Interface.	73
I.12	Input the Passenger Type Information.	73
I.13	Managing Passenger Information Interface.	74
I.14	Input Passenger Information.	74
I.15	Managing Ticket Information Interface.	75
I.16	Booking the Ticket.	75
I.17	Inquiry the Ticket Information.	76
I.18	The Dialog of Printing.	76
I.19	Add a New Operator.	77
I.20	Change the Password of Operator.	77
J.1	The Ticket.	78
J.2	The Report of Airliner.	79
J.3	The Report of Scheduled Flight.	80
J.4	The Report of Ticket.	81

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

LIST OF TABLES

Table		Page
3.1	The Completed Feasibility Analysis Matrix.	12
3.1	Database Server Specification.	16
3.2	Client Specification.	16
3.3	Server Software Specification.	17
3.4	Client Software Specification.	17
3.5	Cost of Existing System, RMB¥.	21
3.6	Five Years Accumulated Existing System Cost, RMB¥.	22
3.7	Estimated Cost of Proposed System, RMB¥.	22
3.8	Five Years Accumulated Proposed System Cost, RMB¥.	24
3.9	The Comparison of the System Cost, RMB¥.	25
3.10	Development Cost, RMB¥.	26
3.11	Annual Operating Cost for Project, RMB¥.	27
3.11	Payback Analysis, RMB¥.	27
3.12	Net Present Value Analysis, RMB¥.	28
5.1	Performance Comparison.	33
A.1	The Cost of Candidate 1, RMB¥.	35
A.2	The Cost of Candidate 2, RMB¥.	36
A.3	The Cost of Candidate 3, RMB¥.	37
A.4	Payback Period for Candidate 1, RMB¥.	38
A.5	Payback Period for Candidate 2, RMB¥.	38
A.6	Payback Period for Candidate 3, RMB¥.	38
E.1	Airliner Information File Layout.	52
E.2	Scheduled Flight Information File Layout.	53

ASSUMPTION UNIVERSITY OF THAILAND SINCE1969 มหาวิทยาลัยอัสสัมชัญ

Table		Page
E.3	Passenger Information File Layout.	54
E.4	Passenger Type Information File Layout.	54
E.5	Service Information File Layout.	55
E.6	Ticket Information File Layout.	56
E.7	User Information File Layout.	56
H.1	Process Specification of Login.	63
H.2	Process Specification of Generate the Scheduled Flight.	64
H.3	Process Specification of Passenger Register.	65
H.4	Process Specification of Booking.	66
H.5	Process Specification of Creating Report.	67

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

I. INTRODUCTION

1.1 Background of Project

The existing system uses a manual system for inter-division processing systems, which are not efficient and cause many problems. Thus, I decided to take a system development project to improve the efficiency and solve the existing problems. This project is initiated to analyse and study the existing system, identify the requirements and design a new system to support these requirements.

There is great economic growth in China today; the competition could be compared to a "dog-eating-dog" one. In order to survive, Yunnan Airlines needs to cope with this kind of situation. The company tries to increase its competitive advantage by improving the speed of business transactions, reducing the response time, reducing mistakes in processing and reducing costs. The company must try to do this by improving inter-divisional processing systems.

The inter-division processing system is very important for the Yunnan Airlines. The existing system has an obvious problem, where there is variance between cabin and ticket selling, as there is no network between them, at same time, it also increases the managerial costs. Implementation of a new system can solve those problems, thus the organization will receive many benefits, and can improve customer satisfaction, make information more accurate with less redundancy, increase the speed of transferring information, and reduce costs for Yunnan Airlines.

1.2 Objectives of Project

The objectives of project management information system for Yunnan airlines are as follows:

(1) To study the existing system and identify the problems of existing system.

(2) To analyze the problems and determine the information requirements.

(3) To design a new system for connecting the cabin and ticketing. This new system will be suitable for the operation environment and will create outputs that support business operations management.

(4) To speed up the process of working when people do the same tasks as before.

(5) To provide the capacity to process greater amount of activities.

(6) To provide faster information retrieval facilities.

(7) To use computing to process data at lower costs than the existing manual system.

(8) To provide greater accuracy and better security.

(9) To provide the better and faster reports to management for decision-making.

1.3 Scope of Project

The project covers major parts of cabin and ticket system that includes:

(1) Developing the Cabin and Ticket System by reviewing company documents and conducting interviews to meet user requirements.

(2) Studying and analysing the existing system from data collection

(3) Designing the proposed system including:

- (a) Data Flow Diagrams
- (b) Input/output Screens
- (c) Structure Chart
- (d) Process Specifications
- (e) Data Dictionary

(4) Implementation stage to insure the proposed system against any errors, for a smooth transition process.

(5) Recommendation for further advanced technology to increase the performance and efficiency of the proposed system.

1.4 Deliverables

The deliverables for the project can classify as following:

(a) Hardware and software requirement

(b) Cost/Benefit analysis

(c) Essential diagram such as ERD and DFD

(d) Database design

(e) Data dictionary

(f) Interface design

(g) Process specification

1.5 Project Plan (Include Gantt Chart)

The project plan is shown in Figure 1.1.

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR VINCIT
OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

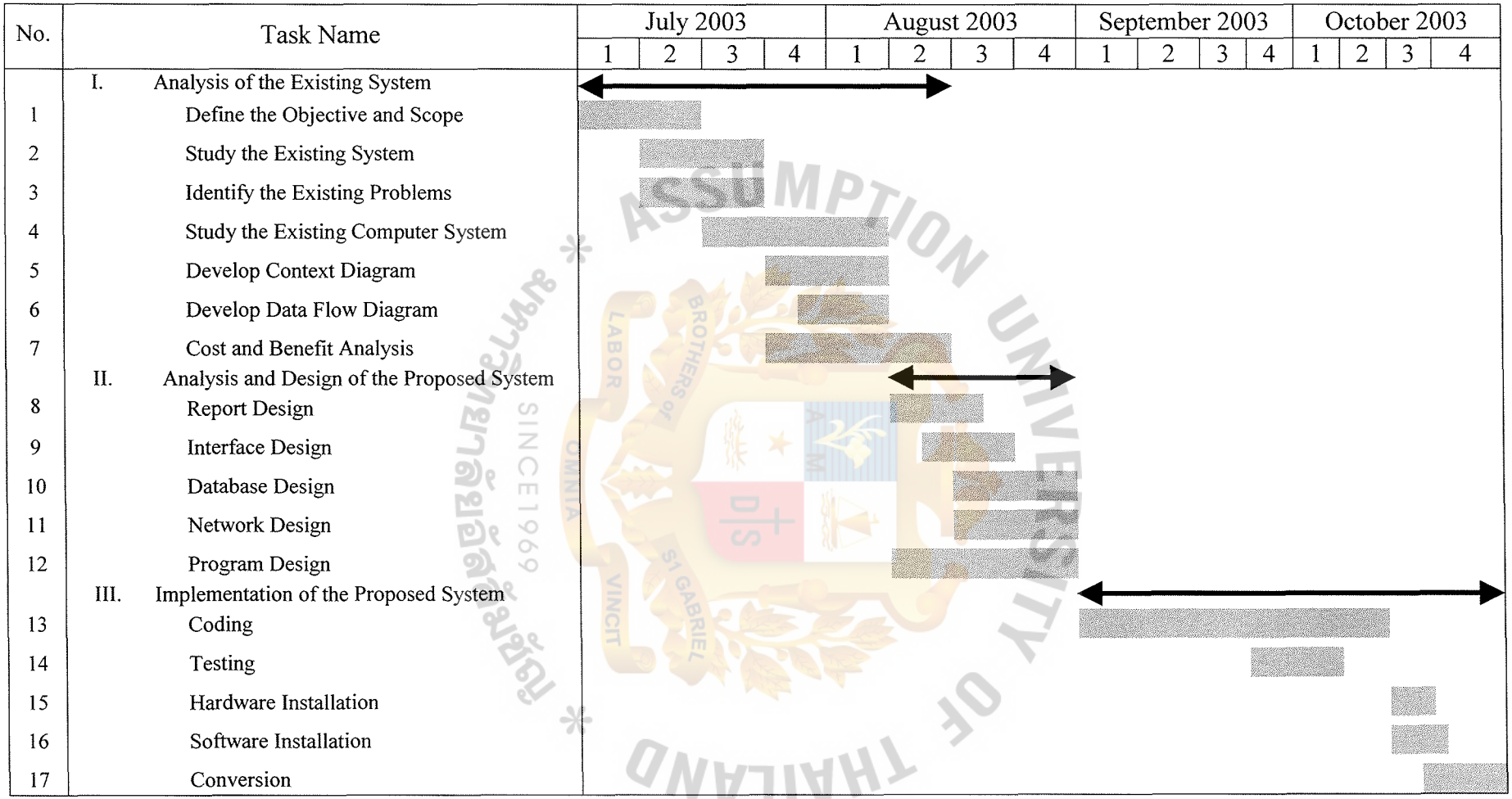

Figure 1.1. Project Plan for MIS of Yunnan Airlines of China.

II. THE EXISTING SYSTEM

2.1 Background of the Company

The headquarters of Yunnan Airlines is situated at Wujiaba Airport in Kunming, Yunnan Province, at the Southwest of China.

China Yunnan Airlines is an aviation transportation enterprise directly under the Civil Aviation Authority of China, symbolized by the Golden Peacock, is well known by domestic and international passengers. It is not only one of the largest airlines in China but also one of the most profitable, and it will continue to maintain the growth of the business and to take advantage of the expected Asian industry expansion. China Yunnan Airlines is fast becoming a major airline of the region.

Yunnan Airlines fleet is one of the most modern and efficient airlines in China. They have 20 Boeing 737 and 5 Boeing 767 aircrafts in operation. More new aircrafts will join Yunnan Airlines fleet shortly. Advanced aircrafts have been added to their airline making their fleet newer and more reliable.

In addition to Kunming Airport, Yunnan Airlines manages nine other airports in Yunnan Province: Mangshi Airport, Simao Airport, Baoshan Airport, Zhaotong Airport, Lijiang Airport, Dali Airport, Zhongdian Airport and Jinghong Airpor,. Lincang Airport. All of these airports are secondary airports in China.

2.2 Existing Organization Function

The organization structure of the Yunnan Airlines consists of three levels of Top-Down hierarchy. The company consists of six divisions. Those are Telecommunication, Finance, Technique, Logistics, Human Resource, and Operation. The organization chart of the Yunnan Airlines is shown in Figure 2.1.

(a) Top level at this level of hierarchy is the Managing Director of the Yunan Airlines.

(b) Middle level of hierarchy consists of all managers of all divisions, which are Telecommunication, Finance, Technique, Logistics, Human Resource, and Operation.

(c) The operations division consists of three sub-divisions in the same level. There are scheduled flight, cabin of aircraft, and ticket. This division is focus that this project is concerned.

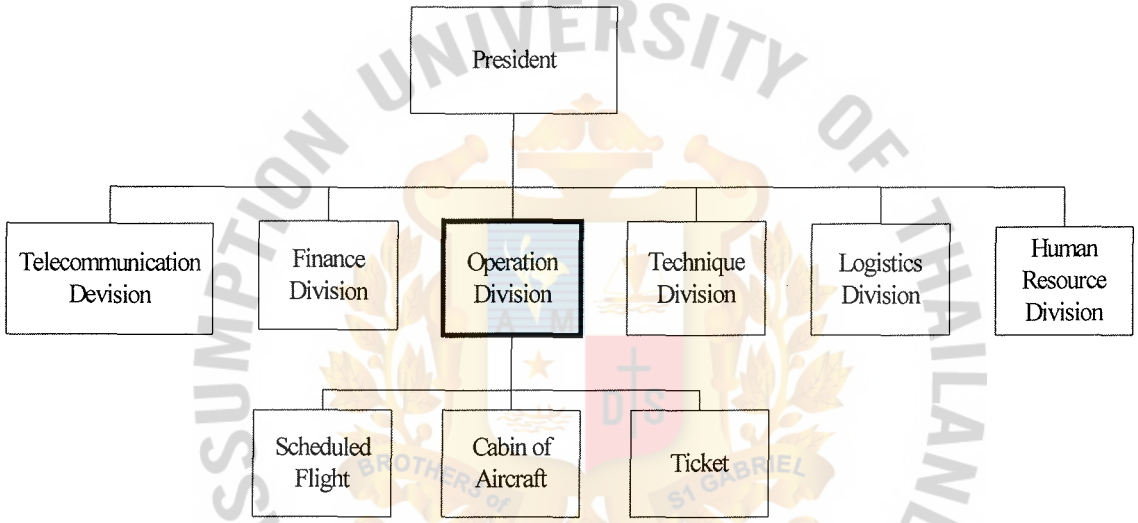

Figure 2.1. Organization Chart of the Yunnan Airlines of China.

The cabin, flight and ticketing of the existing system are a form of conventional files. In the file environment, data storage is built around the applications that will use the files. The conventional files are relatively easy to design and implement because they are normally designed for use with a single application or information system.

The conventional files of existing system also have numerous disadvantages. Duplication of data items in multiple files is normally cited as the principal disadvantage of file-based systems. In the existing system the files tend to be built

around single applications without regard to other (future) applications. All reports and documents are in paper form, and distributed through other divisions.

The existing system produces much paperwork, and need many staffs to operate. It makes redundant processes, inefficient operation, inflexibility, and so on.

As show in Figure 2.1 and Figure 2.2. The inter-division processing system of Yunan Airlines has the following functions:

(1) Aircraft system collects aircraft information and produces flight scheduled information. This information will be sent to the manager of operation division as form of report.

(2) Cabin system makes cabin information based on aircraft information, which is transfered from the manager of operations division. Cabin information will also be sent to the manager of the operations division.

(3) Ticket system produces ticket information based on flight scheduled information and cabin information, which transfer from manager of operation division.

(4) Makes passenger information based on booking, transfers passenger information to the manager of operations division, transfers sales information to the finance division.

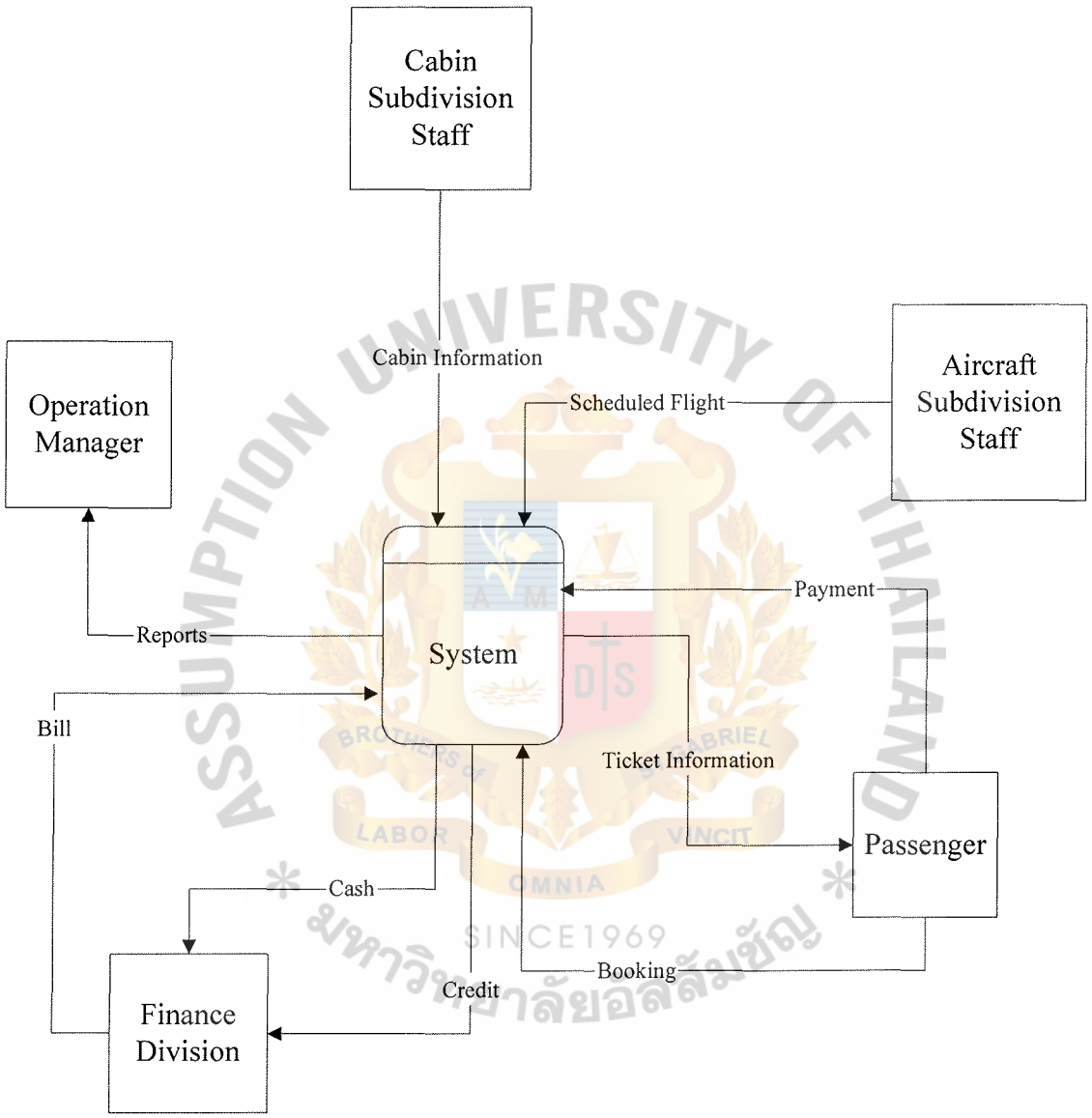

Figure 2.2. The Context Diagram of Yunnan Airlines of China.

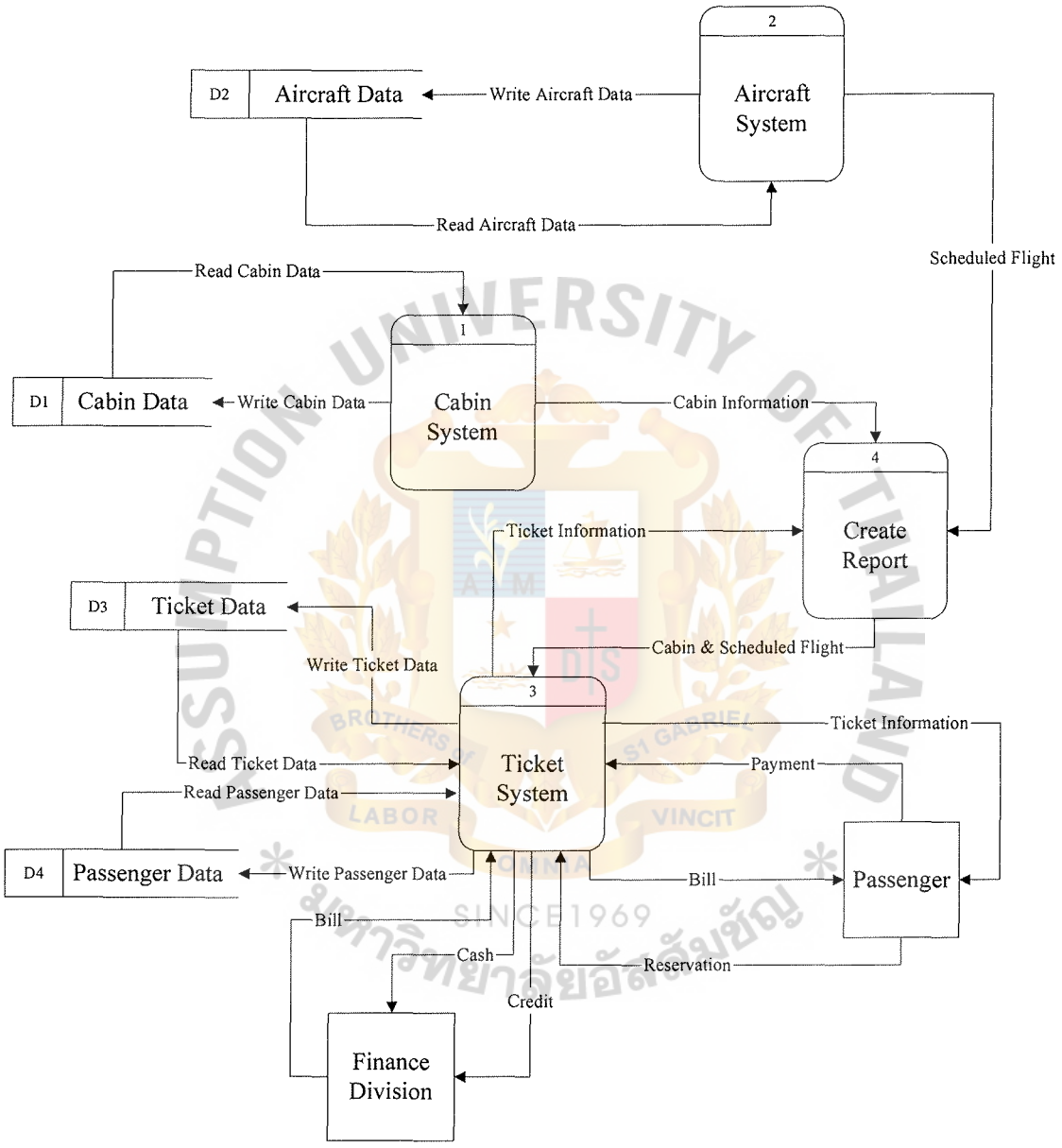

Figure 2.3. The Data Flow Diagram of Yunnan Airlines of China.

2.3 Existing Computer System

Currently Yunnan Airlines uses a manual system and staffs manage the cabin, the fight and ticket. There are no network among divisions of the cabin, the flight and ticket.

2.4 Current Problems of Existing System

The existing system is a manual system, and staffs manage severally the aircrafts, cabins and tickets. The problems of the existing system can be summarized as follows:

(1) It creates an obvious problem, where there is variance between cabin and ticket selling, as there is not network between them, at the same time, it also increases managerial costs.

(2) All information processing is done manually which causes delay in working and decision-making.

(3) The required reports cannot be produced in time for the management team.

(4) It takes more time and costs to gather information to make reports.

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

St. Gabriel's Library, Au
2569

III. THE PROPOSED SYSTEM

3.1 System Specification

The system specification that the user requires is the system which has figures as follows:

(1) The system is easy to manage and control.

(2) The system is used easily.

(3) The system is more powerful and more reliable than the manual system.

(4) The system will be able to assist management to manage and monitor the various projects.

(5) Providing accurate results

(6) Sharing the database that consists of information on aircrafts, cabins, passengers and tickets.

(7) Preparing security systems.

(8) Provide standard input and output reports.

(9) The hardware configuration is Local Area Network (LAN) based.

3.2 Proposed Function

3.2.1 Feasibility Analysis

Feasibility analysis can occur as each candidate is identified or after candidates have been identified. Table 3.1 shows a complete feasibility analysis matrix for each candidate. The full details will be shown in Appendix A. it consists of estimated costs, payback period analysis.

Table 3.1. The Completed Feasibility Analysis Matrix.

Feasibility Criteria	Wt.	Candidate 1	Candidate 2	Candidate 3
Operational Feasibility	30%	Fully supports the user requirements in term of functionality. Score: 70	Fully supports whole functionality, and easy to design and implement. Score: 90	Fully supports whole functionality. Fully support multi users. Score: 100
Technical Feasibility	30%	Use C++ to program main structure, and use MS Access to establish the database. Establish private network by itself. Score: 60	Use VB to program main structure, and use the MS SQL to Establish Database. Establish private network by itself. Score: 80	Use VB to program main structure, use the MS SQL to Establish Database, and link each client via lease line. Score: 70
Economic Feasibility Cost to develop: Payback period: Net present value: Detailed calculation:	30%	Approximately ¥1,019,900 Approximately 4 years 11 months Approximately ¥40904.62 See Appendix A Score: 50	Approximately ¥974,800.00 Approximately 4 year 10 months Approximately ¥86004.62 See Appendix A Score: 50	Approximately ¥454,800 Approximately 2 year 8 months Approximately ¥730497.56 See Appendix A Score: 90
Schedule Feasibility	10%	Around 9 months Score: 40	6 – 7 months Score: 55	Around 4 months Score: 85
Ranking		58	71.5	86.5

ASSUMPTION UNIVERSITY OF THAILAND

3.2.2 Context Diagram and Data Flow Diagram

Context diagram is a top-level diagram that contains a single process, but plays a very important role in studying the management information system. It determines the boundaries of the system. It means that anything that is not inside the process, and identified in the context diagram will not be part of the system.

Data Flow Diagram shows the flow of data into the system between processes and stores. In preparing the model. We emphasize what occurs, not how it is accomplished. Thus, the focus is on logical, rather than physical aspects of the system.

The Context Diagram and Data Flow Diagram are shown in Appendix B.

3.2.3 Database

The database system is designed by using an Entity-Relationship Diagram. It is a graphic modelling tool for showing relationships, which is a popular, has high-level of conceptual data.

(1) Structure Chart of the system will be shown in Appendix C.

(2) Entity relationship diagram (ER diagram)

ER diagram has a high-level of conceptual data that is the group of concepts that help us specify the structure of a database and a set of associated operations for specifying retrievals and updates on the database. The ER diagram is shown in Appendix D.

(3) Table properties

Table properties will show the format of each table or file. It is shown in Appendix E.

(4) Database implement

Based on the ER diagram and table property analysis, the logical framework is obtained. We use SQL language to create these tables. The source code is shown in Appendix F.

3.2.4 Data Dictionary

The data dictionary stores descriptions of data items and structures, as well as system process intend to be used by analysts, who retrieve the details and descriptions in stores, and during system design, when information such as data length, alternate names and data use in specific processes must be available. The data dictionary also stores validation information to guide the analysts in specifying controls for the system's acceptance of data.

The data dictionary will be shown in Appendix G.

3.2.5 Process Specification

The process specifications are created for primitive processes on a data flow diagram as well as for some higher-level processes that explode to a child diagram. These specifications explain the decision-making logic and formulas that will transform processing of inputing data and converting it to output. Each derived element or other previously created of derived elements that are inputed into the primitive process.

The process specification will be shown in Appendix H.

3.2.6 User Interface Design

The user interface is a common boundary between users and the computer system applications: the point where computer and the individuals interact. Users interface may cause the frequency of mistakes and errors when users of the system enter data or instruction. Its purpose is: avoiding user errors; tell the system and users what behaviour is to be performed.

St. Gabriel's Library, Au

It will be shown in Appendix I.

3.2.7 Output Reports

The output is applied to any information produced by an information system, whether printed or displayed. When we design a computer output., we create documents, reports, or other formats that contain information produced by the system.

The reports that can be produced by the system are listed as follows:

(1) Airtickets

(2) Scheduled Flight reports

(3) Cabin reports

(4) Passenger reports

The samples of output will be shown in Appendix J.

3.2.8 Source codes

It will be shown on Appendix K.

3.3 Hardware, Software and People ware Specification

(1) Hardware

Referring to the existing system, the user's PCs stand along, which are not linked to any servers. To implement the proposed system, it requires new hardware for a database server whose primary function is to offer the computing services or manage system resources for client PCs requested for those services. The users do not generally sit at a server and use it as a workstation. The hardware requirements are show in Tables 3.1 and 3.2 respectively.

Table 3.1. Database Server Specification.

Device	Specification
CPU	Intel Pentium 4, processor 3.0 GHz or higher
Cache memory	L2, 512 KB
Primary Memory	512 MB DDR-RAM or higher
Hard drive capacity	80 GB HDD (7200rpm)
Floppy drive	3.5" 1.44MB
DVD/CD-RW	Combo drive 48 × 24 × 48 × 16
Display	Acer CRT flat monitor 17"
LAN	LAN 10/100
UPS	SYNDOME SD 200 1KVA

Table 3.2. Client Specification.

Device	Specification
CPU	Intel Pentium 4, processor 3.0 GHz or higher
Cache memory	L2, 512 KB
Primary Memory	128 MB DDR-RAM or higher
Hard drive capacity	20 GB or Higher
Floppy drive	3.5" 1.44MB
CD-ROM	48 ×
Display	Acer LCD 15"
Modem	56 K, Internal
LAN	LAN 10/100

ASSUMPTION UNIVERSITY OF THAILAND

(2) Software

The followings are the software requirement of this project.

Table 3.3. Server Software Specification.

Software	Specification
Operating system	Microsoft Window 2000
Database server software	Microsoft SQL Server™ 2000
Application server	Microsoft Visual Basic 6.0

Table 3.4. Client Software Specification.

Software	Specification
Operating system	Microsoft Window 98
Application software	Microsoft office 2000
	Microsoft Visual Basic 6.0

(3) People ware

We need not add more people. This project will be operated by the current staff who have been trained and are experienced in managing, controlling, maintaining the hardware and software in Yunnan Airlines.

3.4 Security and Control

Security refers to the protection of computer-based resources such as hardware, software, data, procedures and people against alteration, destruction, or unauthorized use.

There are two main kinds of security, operations security and user security. Operations security has two major protection considerations hardware security and software security. User security will identify authorized users.

The access control functions are intended to limit system access to authorized users and limit system use to only those processes to which the authorized users are entitled.

(1) **User identification** is to identify the staff who are authorized to the information resources providing the password. The users use ID's and passwords to access the system. The different users will hold the different privileges at the same system. The staff who work in airliner subdivision can only view, update, delete, and add the airliner's information. The staff who work in cabin subdivision can only view, update, delete, and add cabin information and service information. The staff who work in the ticket subdivision can only view, update, delete, and book ticket information, also can input passenger information. Only a few users, such as manager can input the discounted information and view all information. The system administrator can only add and delete the staff who can use what privileges, rather than other operation privileges.

(2) **User authentication** is the authorities to gain access and use the information resources that can be verified by re-entering login and password information again. In order to protect the system, the system will be done if the ID or (and) password is (are) wrong three times. The character of a good password is the specification of the minimum range of password (about 6-8 digits). The display must not be displayed.

(3) **User authorization**, with the identification and authentication checks passed, a person can then be authorized certain levels or degrees of use. The access control files take the form of user and field directories. A user directory lists those persons authorized to have access to the system, and a field directory that authorizes users to have access to and perhaps modify certain fields, or data elements, in the database.

(4) **Physical security,** the main failure is due to power supply, and causes interruption to the computers or telecommunications. So UPS need to be used. A safe and secure place must be selected. Make a set of regulations for managing the data information centre and clients.

(5) **Backup and recovery,** in order to ensure the system and data is normal, all data should be backed up in tape or backup devices at the end of the day, kept in a secure place for the purpose of defending data against the risks, such as theft, fire, water, human error, etc.

3.5 Network Configuration

The objective of a network connection is sharing resources such as databases, printers, multiple-user can access to the system at the same time, controlling or using the system at different positions in organization. The connection between database server and clients can be established via LAN. For this project, an intranet is designed for the network of the proposed system. With the intranet, the users will use the same interface to access application and information. It has been leased lines to setup a network based TCP/IP protocols. It is accessible only by the organization members, employees, or people with authorization.

The proposed system is designed to use star topology that uses Hub as a centre of the connecting workstation, shown on Figure 3.1.

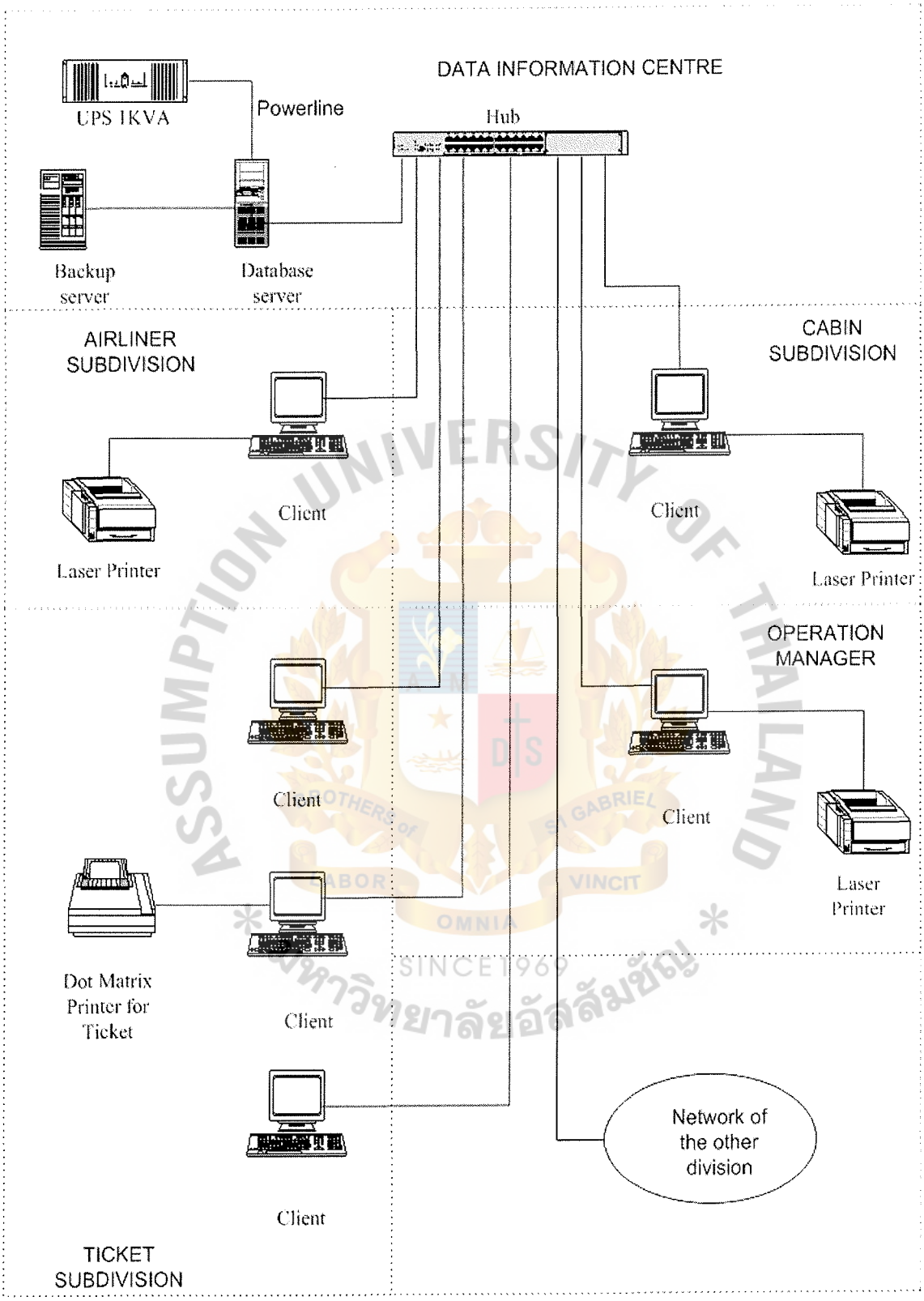

Figure 3.1. Network Configuration of Proposed System.

St. Gabriel's Library, Au

3.6 Cost and Benefit Analysis

3.6.1 Cost Analysis

Table 3.5. Cost of Existing System, RMB¥.

Cost Items		Years				
		1	2	3	4	5
Fixed Cost						
Hardware Cost:						
Personal Computer	15 units @ 6000	18000	18000	18000	18000	18000
Laser Printer	3 units @ 4000	2400	2400	2400	2400	2400
Dot Matrix Printer	10units @ 1500	3000	3000	3000	3000	3000
Total Hardware Cost		23400	23400	23400	23400	23400
Software Cost:						
Microsoft Window 98,	15 units @ 2200	6600	6600	6600	6600	6600
Microsoft Office 2000,	15 units@ 3000	9000	9000	9000	9000	9000
Total Software Cost:		15600	15600	15600	15600	15600
Total Fixed Cost:		39000	39000	39000	39000	39000
Operating Cost						
Salary:						
Operation Manager	1 person @ 3000	3000	3150	3308	3473	3647
Airliner Supervisor	1 person @ 2200	2200	2310	2426	2547	2674
Cabin Supervisor	1 person @ 2200	2200	2310	2426	2547	2674
Ticket Supervisor	1 person @ 2200	2200	2310	2426	2547	2674
Staffs	70 persons @ 1500	105000	110250	115763	121551	127628
Total monthly salary cost:		114600	120330	126347	132664	139297
Total annual salary cost:		1375200	1443960	1516158	1591966	1671564
Office Supplies & Miscellaneous:						
Stationary	30000 Per Annum	30000	31500	33075	34729	36465
Paper	20000 Per Annum	20000	21000	22050	23153	24310
Utility	35000 Per Annum	35000	36750	38588	40517	42543
Miscellaneous	25000 Per Annum	25000	26250	27563	28941	30388
Total annual office Supplies & Miscellaneous Cost:		110000	115500	121275	127339	133706
Total annual Operating Cost:						
Total Manual System Cost:		1485200	1559460	1637433	1719305	1805270

Note: The RMB¥ is an abbreviation for the unit of Chinese currency, the full word is Ren Ming Bi Yuan.

Table 3.6. Five Years Accumulated Existing System Cost, RMB¥.

Year	Total Existing System Cost	Accumulated Cost
1	1,524,200	1,524,200
2	1,598,460	3,122,660
3	1,676,433	4,799,093
4	1,758,305	6,557,398
5	1,844,270	8,401,668
Total	8,401,668	-

Table 3.7. Estimated Cost of Proposed System, RMB¥.

Cost Items	Years				
	1	2	3	4	5
Fixed Cost					
Hardware Cost:					
Database Server 1 unit @ 18000	3,600	3,600	3,600	3,600	3,600
Backup Server 1 unit @ 18000	3,600	3,600	3,600	3,600	3,600
Client (Personal Computer) 15 units @6000	18,000	18,000	18,000	18,000	18,000
Laser Printer 3 units @4000	2,400	2,400	2,400	2,400	2,400
Dot Matrix Printer 10 units @1000	2,000	2,000	2,000	2,000	2,000
UPS 1 KVA 1 unit @ 1800	360	360	360	360	360
Total Hardware Cost	29,960	29,960	29,960	29,960	29,960
Software Cost:					
Microsoft Window 2000, 1 units @ 2000	400	400	400	400	400
Microsoft SQL Server 2000, 1 units @ 2600	520	520	520	520	520

ASSUMPTION UNIVERSITY OF THAILAND
SINCE1969

Table 3.7. Estimated Cost of Proposed System, RMB¥ (Continued).

Cost Items	Years				
	1	2	3	4	5
Microsoft Visual Basic 6.0, 1 units @ 1400	280	280	280	280	280
Microsoft Window 98, 15 units @ 2000	6,000	6,000	6,000	6,000	6,000
Microsoft Office 2000, 15 units @ 2000	6,000	6,000	6,000	6,000	6,000
Total Software Cost:	13,640	13,640	13,640	13,640	13,640
People Ware Cost:					
System Analysts 6 months @ 3500	21,000	-	-	-	-
Programmers 2 months @ 4000	8,000	-	-	-	-
Network Specialist 1 month @ 3800	3,800	-	-	-	-
Database Specialist 1 month @ 4000	4,000	-	-	-	-
Total People Ware Cost:	36,800	-	-	-	-
Maintenance Cost:					
Hardware Maintenance Cost	-	1,500	1,500	1,500	1,500
Software Maintenance Cost	-	1,000	1,000	1,000	1,000
Total Maintenance Cost:	-	2,500	2,500	2,500	2,500
Implementation Cost:					
Training Cost	30,000	-	-	-	-
Installation Cost	15,000	-	-	-	-
Testing Cost	20,000				
Total Implementation Cost:	45,000	-	-	-	-
Total Fixed Cost:	124,960	45,660	45,660	45,660	45,660
Operating Cost					
Salary:					
Operation Manager 1 person @3000	3,000	3,150	3,308	3,473	3,647
Airliner Supervisor 1 person @2200	2,200	2,310	2,426	2,547	2,674

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF LABOR ST GABRIEL VINCIT OMNIA SINCE 1969
มหาวิทยาลัยอัสสัมชัญ

St. Gabriel's Library, Au

Table 3.7. Estimated Cost of Proposed System, RMB¥ (Continued).

Cost Items	Years				
	1	2	3	4	5
Cabin Supervisor 1 person @2200	2,200	2,310	2,426	2,547	2,674
Ticket Supervisor 1 person @2200	2,200	2,310	2,426	2,547	2,674
Staffs 50 persons @ 1800	90,000	94,500	99,225	104,186	109,396
System and Network Administrator 2 persons @3600	7,200	7,560	7,938	8,335	8,752
Total monthly salary cost:	106,800	112,140	117,747	123,634	129,816
Total annual salary cost:	1,281,600	1,345,680	1,412,964	1,483,612	1,557,793
Office Supplies & Miscellaneous:					
Stationary 12000 Per Annum	12,000	12,600	13,230	13,892	14,586
Paper 10000 Per Annum	10,000	10,500	11,025	11,576	12,155
Utility 25000 Per Annum	25,000	26,250	27,563	28,941	30,388
Miscellaneous 18000 Per Annum	18,000	18,900	19,845	20,837	21,879
Lease Line Cost	120,000	114,000	102,600	92,340	83,106
Total annual office Supplies & Miscellaneous Cost:	185,000	182,250	174,263	167,586	162,114
Total annual Operating Cost:	1,466,600	1,527,930	1,587,227	1,651,198	1,719,907
Total Proposed System Cost:	1,591,560	1,573,590	1,632,887	1,696,858	1,765,567

Table 3.8. Five Years Accumulated Proposed System Cost, RMB¥.

Year	Total Proposed System Cost	Accumulated Cost
1	1,591,560	1,591,560
2	1,573,590	3,165,150
3	1,632,887	4,798,037
4	1,696,858	6,494,895
5	1,765,567	8,260,462
Total	8,260,462	-

Table 3.9. The Comparison of the System Cost, RMB¥.

Year	Accumulated Existing System Cost	Accumulated Proposed System Cost
1	1,524,200	1,592,000
2	3,122,660	3,166,030
3	4,799,093	4,799,357
4	6,557,398	6,496,655
5	8,401,668	8,262,662

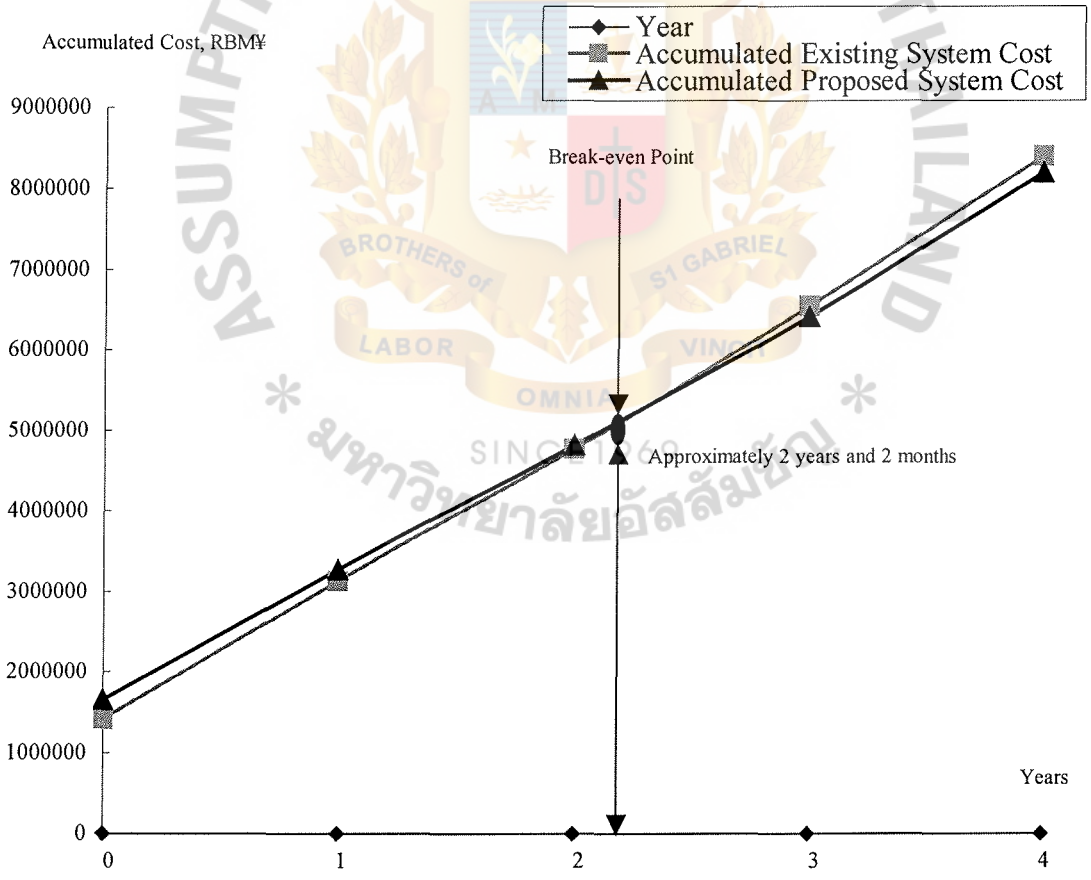

Figure 3.2. Break-even Chart for the System.

3.6.2 Benefit Analysis

(1) Payback Analysis

Table 3.10. Development Cost, RMB¥.

Items	Costs
Personnel:	
System Analysts 6 months @ 3500	21,000.00
Programmers 2 months @ 4000	8,000.00
Network Specialist 1 month @ 3800	3,800.00
Database Specialist 1 month @ 4000	4,000.00
	36,800.00
Hardware & Software:	
Database Server 1 unit @ 18000	18,000.00
Backup Server 1 unit @ 18000	18,000.00
Client (Personal Computer) 15 units @6000	90,000.00
Laser Printer 3 units @4000	12,000.00
Dot Matrix Printer 10 units @1220	12,200.00
UPS 1 KVA 1 unit @ 1800	1,800.00
Microsoft Window 2000, 1 unit @ 2000	2,000.00
Microsoft SQL Server 2000, 1 unit @ 2600	2,600.00
Microsoft Visual Basic 6.0, 1 unit @ 1400	1,400.00
Microsoft Window 98, 15 units @ 2000	30,000.00
Microsoft Office 2000, 15 units @ 2000	30,000.00
Training Cost	30,000.00
Installation Cost	15,000.00
Testing Cost	20,000.00
	218,000.00
Utilities:	200,000.00
Total development costs	454,800.00

Table 3.11. Annual Operating Cost for Project, RMB¥.

Items	Cost
Operating Cost:	
Salary:	
Operation Manager 1 person @3000	3,000
Airliner Supervisor 1 person @2200	2,200
Cabin Supervisor 1 person @2200	2,200
Ticket Supervisor 1 person @2200	2,200
Staffs 50 persons @ 1800	90,000
System and Network Administrator 2 persons @3600	7,200
Total monthly salary cost:	106,800
Total annual salary cost:	1,281,600
Office Supplies & Miscellaneous:	
Stationary 12000 Per Annum	12,000
Paper 10000 Per Annum	10,000
Utility 25000 Per Annum	25,000
Miscellaneous 18000 Per Annum	18,000
Lease Line Cost	120,000
Total annual office Supplies & Miscellaneous Cost:	185,000
Total annual Operating Cost:	1,466,600

Table 3.11. Payback Analysis, RMB¥.

Cash flow description	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Development Cost:	-454800					
Operation & Maintenance Costs:		-1466600	-1539930	-1616926.5	-1697772.83	-1782661
Discount factors for 10%	1	0.909	0.826	0.751	0.683	0.621
Time-adjust costs (Adjusted to present value):	-454800	-1333139.4	-1271982.18	-1214311.8	-1159578.84	-1107032
Cumulative time-adjusted costs over lifetime:	-454800	-1787939.4	-3059921.58	-4274233.38	-5433812.22	-6540844
Benefits derived from operation of new system:	0	1600000	1760000	1936000	2129600	2342560
Discount factors for 10%	1	0.909	0.826	0.751	0.683	0.621
Time-adjust benefits (Current of present value):	0	1454400	1453760	1453936	1454516.8	1454729
Cumulative time-adjusted benefits over lifetime:	0	1454400	2908160	4362096	5816612.8	7271342
	0	1	2	3	4	5
Cumulative time-adjusted Cost + Benefits	-454800	-333539	-151762	87863	382801	730498

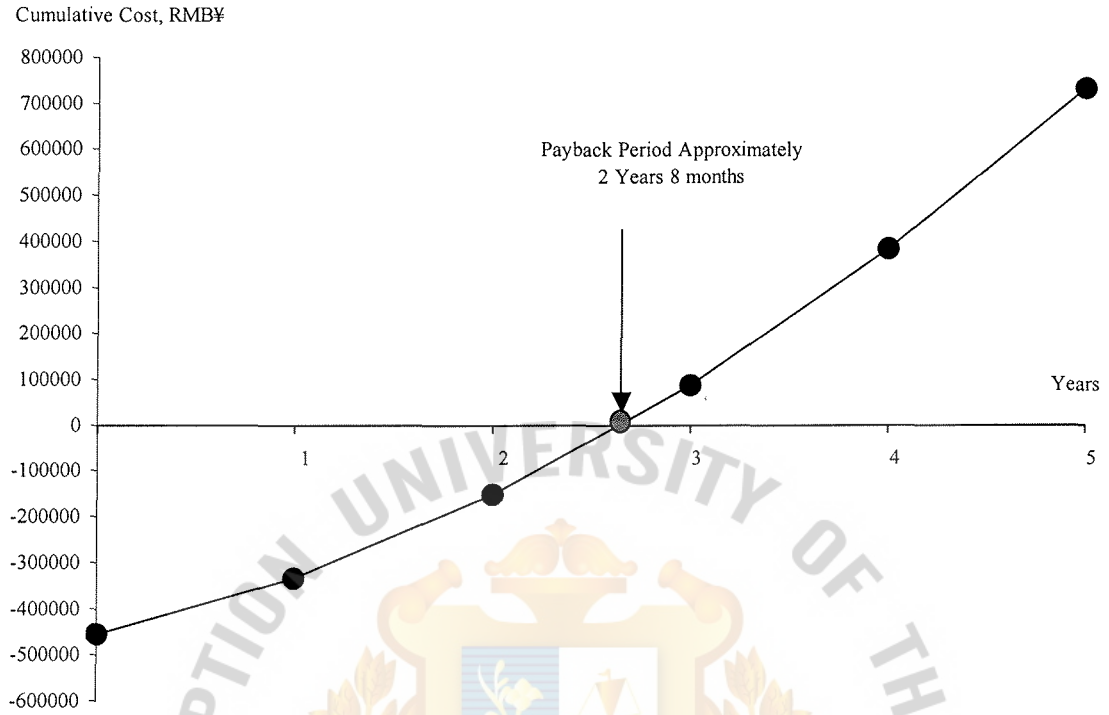

Figure 3.3. Payback Chart for Project.

Table 3.12. Net Present Value Analysis, RMB¥.

Cash flow description	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5	Total
Development Cost:	-454800						
Operation & Maintenance Costs:		-1466600	-1539930	-1616926.5	-1697772.83	-1782661.47	
Discount factors for 10%	1	0.909	0.826	0.751	0.683	0.621	
Time-adjust costs (Adjusted to present value):	-454800	-1333139	-1271982	-1214311	-1159579	-1107032	
Cumulative time-adjusted costs over lifetime:							-6540845
Benefits derived from operation of new system:	0	1600000	1760000	1936000	2129600	2342560	
Discount factors for 10%	1	0.909	0.826	0.751	0.683	0.621	
Time-adjust benefits (Current of present value):	0	1454400	1453760	1453936	1454516	1454729	
Cumulative time-adjusted benefits over lifetime:							7271343
Net Present Value of This Alternative							730498

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR VINCIT
OMNIA
SINCE1969

St. Gabriel's Library, Au

(2) Benefit

Based on Tables 3.5, 3.6, 3.7, 3.8, 3.9 and Figure 3.2, the proposed system will reduce the cost, and the cost of proposed system will be lower than existing system after 3 years approximately.

Based on Tables 3.10, 3.11, 3.12 and Figure 3.3, the proposed system will return the investments after 2 years and 10 months. And after 5 years, the proposed system will gain benefits of RMB¥ 730498.

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

IV. SYSTEM IMPLEMENTATION

4.1 Overview of Project Implementation

The implementation of the proposed system will start after the top management has agreed to implement the proposed system. There are five steps in the implementation.

First, the proposed system should be analyzed, and designed based on the background of the company & user requirements.

Second, programming in the proposed system, includes modular or partitions of some functions. Thus we use computer languages to program it, such as Visual Basic 6.0.

Third, the programmed system has to be tested, and ensured that it is correct, reliable, also fulfills the user's requirements. All bugs and errors should be debugged and modified for the purpose of running without any errors.

Fourth, after testing, the new system and network should be installed based on designing requirements.

Finally, editing the user manual, perform the training program in order to ensure that user knows how to use the new system effectively and correctly.

4.2 Schedule of the Project

4.2.1 System Analysis

The first step of implementation is to analyze the system, based on the user's requirements, Data Flow Diagrams will be drawn out. The process of system will also be pre-designed. The database structure is also pre-designed. Network construction will be pre-designed.

4.2.2 System Design

The program must be divided into some modules and (or) functions, and then, we write the runnable source codes or programs based on these modules and (or) functions.

In this project, we will use Microsoft Visual Basic 6.0 to code programs. Microsoft Visual Basic will automatically check syntax errors and run time errors and warn us to correct these errors. SQL is used for creating the database of the Management Information System of Airlines. All software used is with standard programs and has many useful functions for developers. It can help developers to save time and program efficiently.

Otherwise, after programing, the network structure should be pre-designed.

4.2.3 Testing

This stage concerns the executing of the program to find any errors before system is used. It consists of several steps as follow:

(1) Module testing

Each program should be tested individually, we can say it correct after testing the each module and (or) function.

(2) Link testing

This testing is implemented to ensure that the programs are linked and work together as required. For purpose of testing, a set of data contains both valid and invalid data to test all possible situations that may occur in the future.

(3) Integrated testing

The whole system (or all functions) will be tested to ensure the programming is correct, that it satisfies the user's requirements, and it is reliable and safe.

(4) Backup testing

After finishing testing and debugging, if the system is correct and faultless, the full system will be tested along with the backup, that is, to ensure that the system is backed up in case there are any disaster, and (or) they are destroyed.

(5) Network testing

For this project, the network is most important, so the network has to be tested after finishing the testing program. This testing ensures that telecommunication is correct and unobstructed.

4.2.4 System Installation

After installing system and linking the network, the whole system can be implemented. It consists of several steps:

(1) Obtaining the hardware

(2) Obtaining the software

(3) Preparing the database

(4) Installing tested programs

4.2.5 Providing Manual and Training

Before using the system, we should write a manual for the purpose of operating the system correctly. On the other side, the training courses must be offered for staff using a new system.

4.3 Conversion

Due to the existing system being manual system (can also say semi-computerized system); it is not difficult to convert. The existing system uses in each division. We need to just set up the database servers and backup servers, install the programs for the client, and set up a network among them.

V. CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In the current manual system, we find that the processing of Management Information System is delayed and ineffective because of the weakness as of the manual system, the filing is weak, transmission of information is very slow, it produces higher operation costs, and the procedure of jobs is also ineffective. So it is a waste of time as well as increase in costs.

The Management Information System for Yunnan Airlines of China is a computerized & network based system. The ticket subdivision can know cabin information and scheduled flight information directly and quickly. At the same time, other subdivisions can also share this data and information, thus, the management can know the operating conditions and information directly and quickly, and make good decisions.

Due to sharing the data and information, the ticket sales will be correct and timely, cabin subdivision can also know the passenger's information in time.

The benefits of the new system are very obvious, because, the processing time of each step is reduced obviously, it is changed to 5 minutes, as shown as Table 5.1.

Table 5.1. Performance Comparison.

Report Name	Estimated Time Needed	
	Manual	Proposed
Passenger List	8hr. –36 hr.	5 min.
Cabin Information	4hr. – 28 hr.	5 min.
Airliner Information	6hr. – 48 hr	5 min.
Ticket Information	24 hr. – 48 hr.	5 min.
Operation Report	10hr. – 24hr.	5 min.
Scheduled Flight Information	5hr. – 24 hr.	5 min.

Based on cost-benefit analysis, the proposed system has higher costs than exiting system, but, after 3 years, the costs are lower than existing system. Thus, based on the payback analysis, the investment will be retuned after 2 years and 10 months.

5.2 Recommendations

This project is the first management information system that Yunnan Airlines of China has developed to implement in an organization. The objective is to improve the effectiveness and efficiency of the old system, they want to solve the other problems existing.

Because of other problems, still existing in other divisions. We recommend that Yunnan Airlines of China should develop a Management Information System for other divisions, and set up network among them for the purpose of sharing data and information. Thus, it can raise the effectiveness and efficiency, and reduce the operation costs.

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

St. Gabriel's Library, Au

APPENDIX A

FEASIBILITY ANALYSIS

Table A.1. The Cost of Candidate 1, RMB¥.

Items	Costs
Personnel:	
System Analysts 6 months @ 3500	21,000.00
Programmers 5 months @ 6000	30,000.00
Network Specialist 3 month @ 3800	11,400.00
Database Specialist 1 month @ 4000	4,000.00
	66,400.00
Hardware & Software:	
Database Server 1 unit @ 18000	18,000.00
Backup Server 1 unit @ 18000	18,000.00
Client (Personal Computer) 15 units @6000	90,000.00
Laser Printer 3 units @4000	12,000.00
Dot Matrix Printer 10 units @1220	12,200.00
UPS 1 KVA 1 unit @ 1800	1,800.00
Coaxial-cable 90km @ 8000	720,000.00
Microsoft Window 2000, 1 units @ 2000	2,000.00
Microsoft Access, 1 units @ 1500	1,500.00
Microsoft C++, 1 units @ 8000	8,000.00
Microsoft Window 98, 15 units @ 2000	30,000.00
Microsoft Office 2000, 15 units @ 2000	30,000.00
Training Cost	30,000.00
Installation Cost	40,000.00
Testing Cost	40,000.00
	913,500.00
Utilities	200,000.00
Total development costs	1,019,900.00
Operating Cost	
Salary:	
Operation Manager 1 person @3000	3,000
Airliner Supervisor 1 person @2200	2,200
Cabin Supervisor 1 person @2200	2,200
Ticket Supervisor 1 person @2200	2,200
Staffs 50 persons @ 1800	90,000
System and Network Administrator 2 persons @3600	7,200
Total monthly salary cost	106,800
Total annual salary cost	1,281,600
Office Supplies & Miscellaneous:	
Stationary 12000 Per Annum	12,000
Paper 10000 Per Annum	10,000
Utility 25000 Per Annum	25,000
Miscellaneous 18000 Per Annum	18,000
Maintenance Line Cost	150,000
Total annual office Supplies & Miscellaneous Cost	215,000
Total annual Operating Cost	1,496,600
Total Computerized System Cost	2,516,500.00

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Table A.2. The Cost of Candidate 2, RMB¥.

Items	Costs
Personnel:	
System Analysts 6 months @ 3500	21,000.00
Programmers 2 months @ 4000	8,000.00
Network Specialist 1 month @ 3800	3,800.00
Database Specialist 1 month @ 4000	4,000.00
	36,800.00
Hardware & Software:	
Database Server 1 unit @ 18000	18,000.00
Backup Server 1 unit @ 18000	18,000.00
Client (Personal Computer) 15 units @6000	90,000.00
Laser Printer 3 units @4000	12,000.00
Dot Matrix Printer 10 units @1220	12,200.00
UPS 1 KVA 1 unit @ 1800	1,800.00
Coaxial-cable 90km @ 8000	720,000.00
Microsoft Window 2000, 1 units @ 2000	2,000.00
Microsoft SQL Server 2000, 1 units @ 2600	2,600.00
Microsoft Visual Basic 6.0, 1 units @ 1400	1,400.00
Microsoft Window 98, 15 units @ 2000	30,000.00
Microsoft Office 2000, 15 units @ 2000	30,000.00
Training Cost	40,000.00
Installation Cost	20,000.00
Testing Cost	30,000.00
	908,000.00
Utilities	200,000.00
Total development costs	974,800.00
Operating Cost	
Salary:	
Operation Manager 1 person @3000	3,000
Airliner Supervisor 1 person @2200	2,200
Cabin Supervisor 1 person @2200	2,200
Ticket Supervisor 1 person @2200	2,200
Staffs 50 persons @ 1800	90,000
System and Network Administrator 2 persons @3600	7,200
Total monthly salary cost	106,800
Total annual salary cost	1,281,600
Office Supplies & Miscellaneous:	
Stationary 12000 Per Annum	12,000
Paper 10000 Per Annum	10,000
Utility 25000 Per Annum	25,000
Miscellaneous 18000 Per Annum	18,000
Maintenance Line Cost	150,000
Total annual office Supplies & Miscellaneous Cost	215,000
Total annual Operating Cost	1,496,600
Total Computerized System Cost	2,471,400.00

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Table A.3. The Cost of Candidate 3, RMB¥.

Items	Costs
Personnel:	
System Analysts 6 months @ 3500	21,000.00
Programmers 2 months @ 4000	8,000.00
Network Specialist 1 month @ 3800	3,800.00
Database Specialist 1 month @ 4000	4,000.00
	36,800.00
Hardware & Software:	
Database Server 1 unit @ 18000	18,000.00
Backup Server 1 unit @ 18000	18,000.00
Client (Personal Computer) 15 units @6000	90,000.00
Laser Printer 3 units @4000	12,000.00
Dot Matrix Printer 10 units @1220	12,200.00
UPS 1 KVA 1 unit @ 1800	1,800.00
Microsoft Window 2000, 1 units @ 2000	2,000.00
Microsoft SQL Server 2000, 1 units @ 2600	2,600.00
Microsoft Visual Basic 6.0, 1 units @ 1400	1,400.00
Microsoft Window 98, 15 units @ 2000	30,000.00
Microsoft Office 2000, 15 units @ 2000	30,000.00
Training Cost	30,000.00
Installation Cost	15,000.00
Testing Cost	20,000.00
	218,000.00
Utilities	200,000.00
Total development costs	454,800.00
Operating Cost	
Salary:	
Operation Manager 1 person @3000	3,000
Airliner Supervisor 1 person @2200	2,200
Cabin Supervisor 1 person @2200	2,200
Ticket Supervisor 1 person @2200	2,200
Staffs 50 persons @ 1800	90,000
System and Network Administrator 2 persons @3600	7,200
Total monthly salary cost	106,800
Total annual salary cost	1,281,600
Office Supplies & Miscellaneous:	
Stationary 12000 Per Annum	12,000
Paper 10000 Per Annum	10,000
Utility 25000 Per Annum	25,000
Miscellaneous 18000 Per Annum	18,000
Lease Line Cost	120,000
Total annual office Supplies & Miscellaneous Cost	185,000
Total annual Operating Cost	1,466,600
Total Computerized System Cost	1,921,400.00

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
VINCIT OMNIA
SINCE1969

Table A.4. Payback Period for Candidate 1, RMB¥.

Cash flow description	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Development Cost:	-1019900					
Operation & Maintenance Costs:		-1496600	-1571430	-1650002	-1732502	-1819127
Discount factors for 10%	1	0.909	0.826	0.751	0.683	0.621
Time-adjust costs (Adjusted to present value):	-1019900	-1360409	-1298001	-1239151	-1183299	-1129678
Cumulative time-adjusted costs over lifetime:	-1019900	-2380309	-3678311	-4917462	-6100760	-7230438
Benefits derived from operation of new system:	0	1600000	1760000	1936000	2129600	2342560
Discount factors for 10%	1	0.909	0.826	0.751	0.683	0.621
Time-adjust benefits (Current of present value):	0	1454400	1453760	1453936	1454517	1454730
Cumulative time-adjusted benefits over lifetime:	0	1454400	2908160	4362096	5816613	7271343
Cumulative time-adjusted Cost + Benefits	-1019900	-925909	-770151	-555366	-284147	40905

Table A.5. Payback Period for Candidate 2, RMB¥.

Cash flow description	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Development Cost:	-974800					
Operation & Maintenance Costs:		-1496600	-1571430	-1650002	-1732502	-1819127
Discount factors for 10%	1	0.909	0.826	0.751	0.683	0.621
Time-adjust costs (Adjusted to present value):	-974800	-1360409	-1298001	-1239151	-1183299	-1129678
Cumulative time-adjusted costs over lifetime:	-974800	-2335209	-3633211	-4872362	-6055661	-7185338
Benefits derived from operation of new system:	0	1600000	1760000	1936000	2129600	2342560
Discount factors for 10%	1	0.909	0.826	0.751	0.683	0.621
Time-adjust benefits (Current of present value):	0	1454400	1453760	1453936	1454517	1454730
Cumulative time-adjusted benefits over lifetime:	0	1454400	2908160	4362096	5816613	7271343
Cumulative time-adjusted Cost + Benefits	-974800	-880809	-725051	-510266	-239048	86005

Table A.6. Payback Period for Candidate 3, RMB¥.

Cash flow description	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
Development Cost:	-454800					
Operation & Maintenance Costs:		-1466600	-1539930	-1616927	-1697773	-1782662
Discount factors for 10%	1	0.909	0.826	0.751	0.683	0.621
Time-adjust costs (Adjusted to present value):	-454800	-1333139	-1271982	-1214311.8	-1159579	-1107033
Cumulative time-adjusted costs over lifetime:	-454800	-1787939	-3059922	-4274233	-5433812	-6540845
Benefits derived from operation of new system:	0	1600000	1760000	1936000	2129600	2342560
Discount factors for 10%	1	0.909	0.826	0.751	0.683	0.621
Time-adjust benefits (Current of present value):	0	1454400	1453760	1453936	1454517	1454730
Cumulative time-adjusted benefits over lifetime:	0	1454400	2908160	4362096	5816613	7271343
Cumulative time-adjusted Cost + Benefits	-454800	-333539	-151762	87863	382801	730498

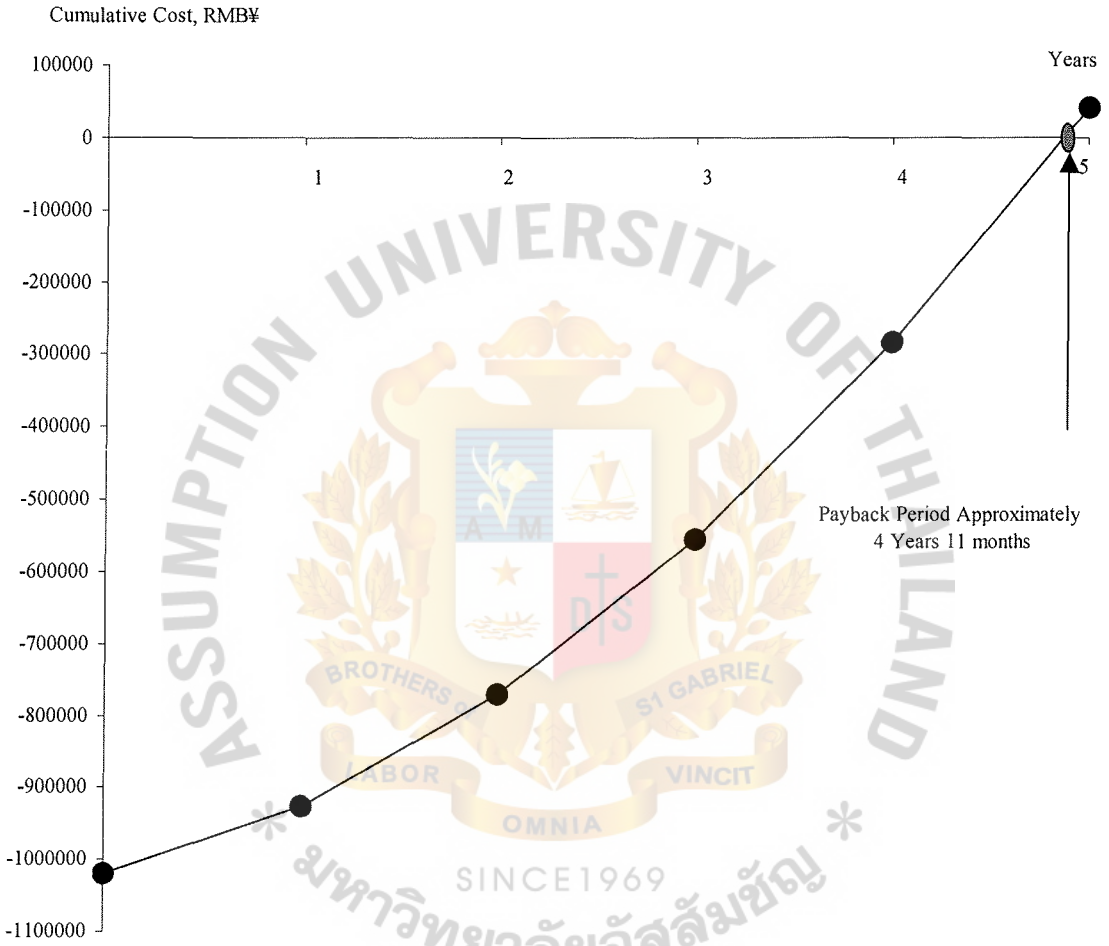

Figure A.1. Payback Period for Candidate 1.

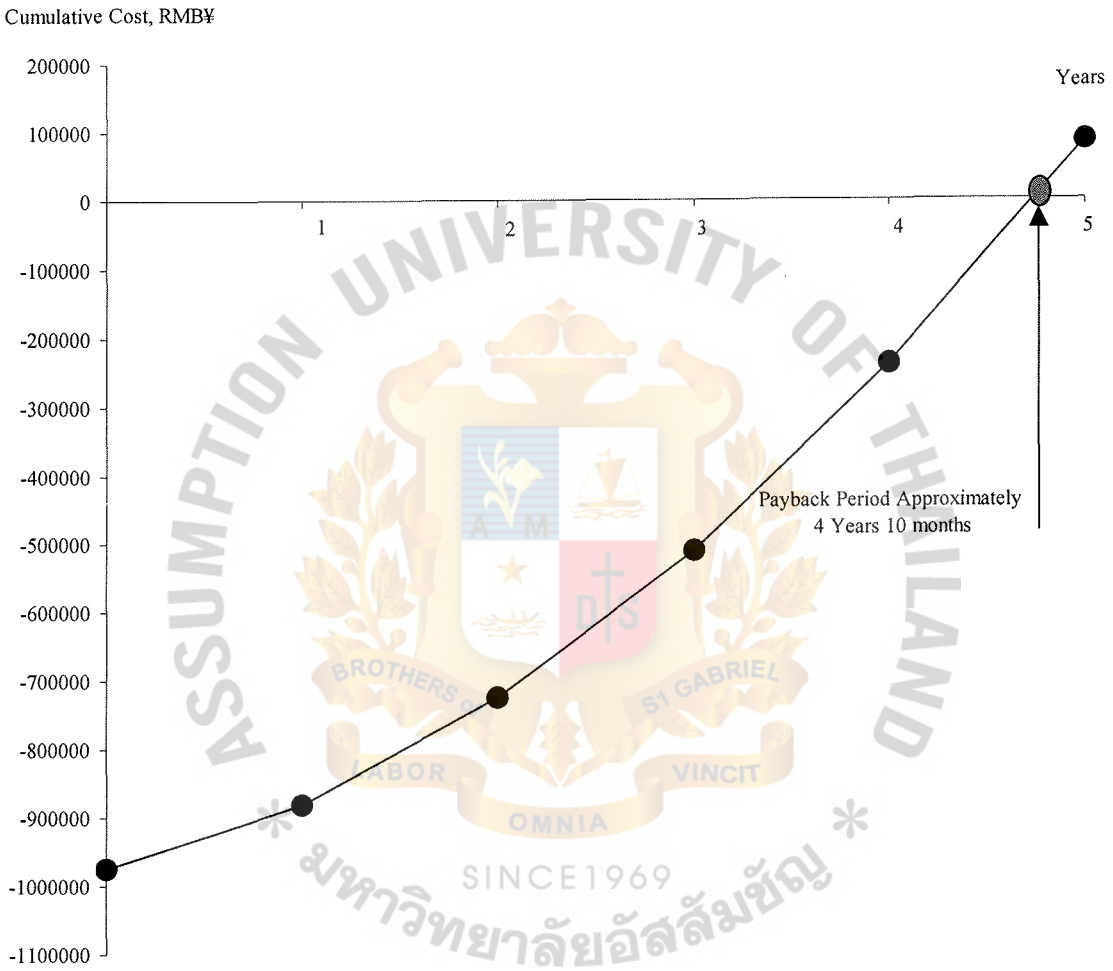

Figure A.2. Payback Period for Candidate 2.

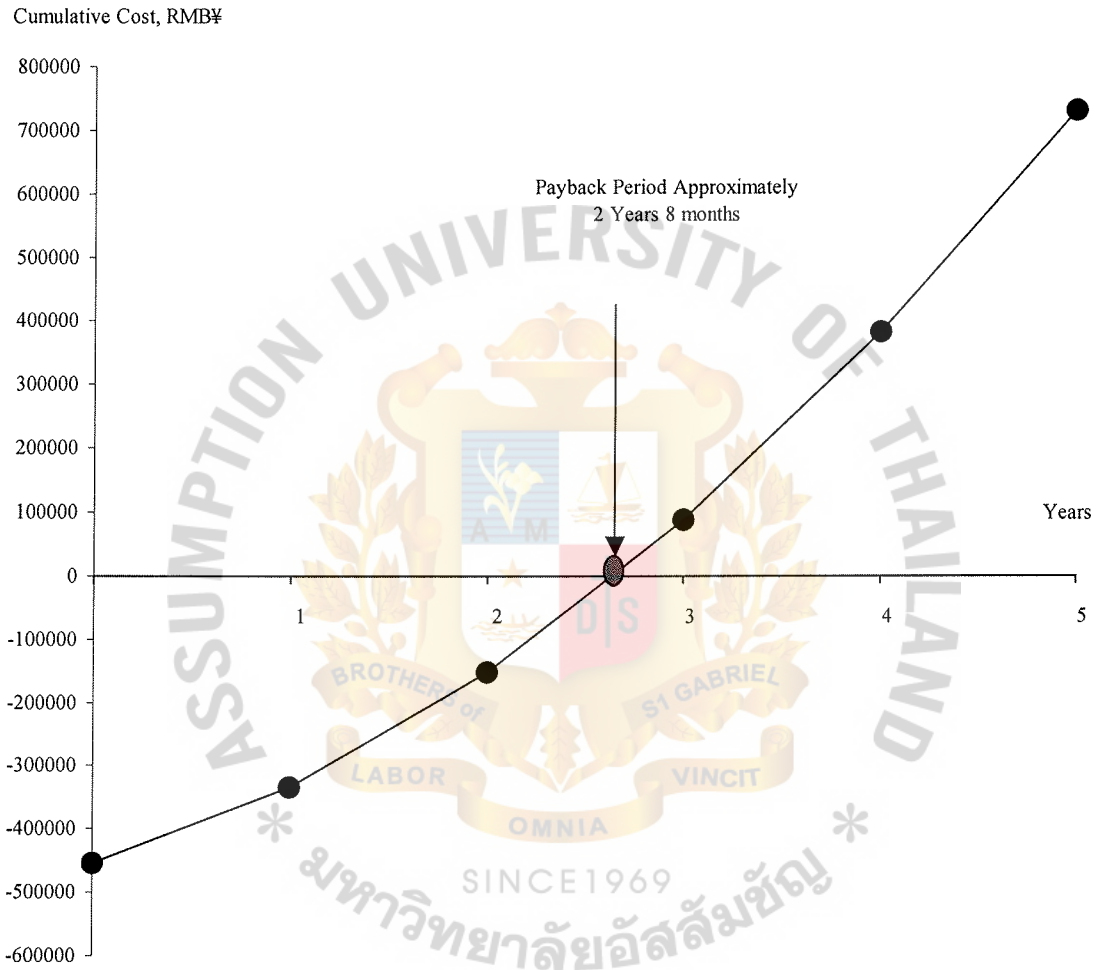

Figure A.3. Payback Period for Candidate 3.

St. Gabriel's Library, Au

APPENDIX B

DATA FLOW DIAGRAM

ASSUMPTION UNIVERSITY OF THAILAND
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

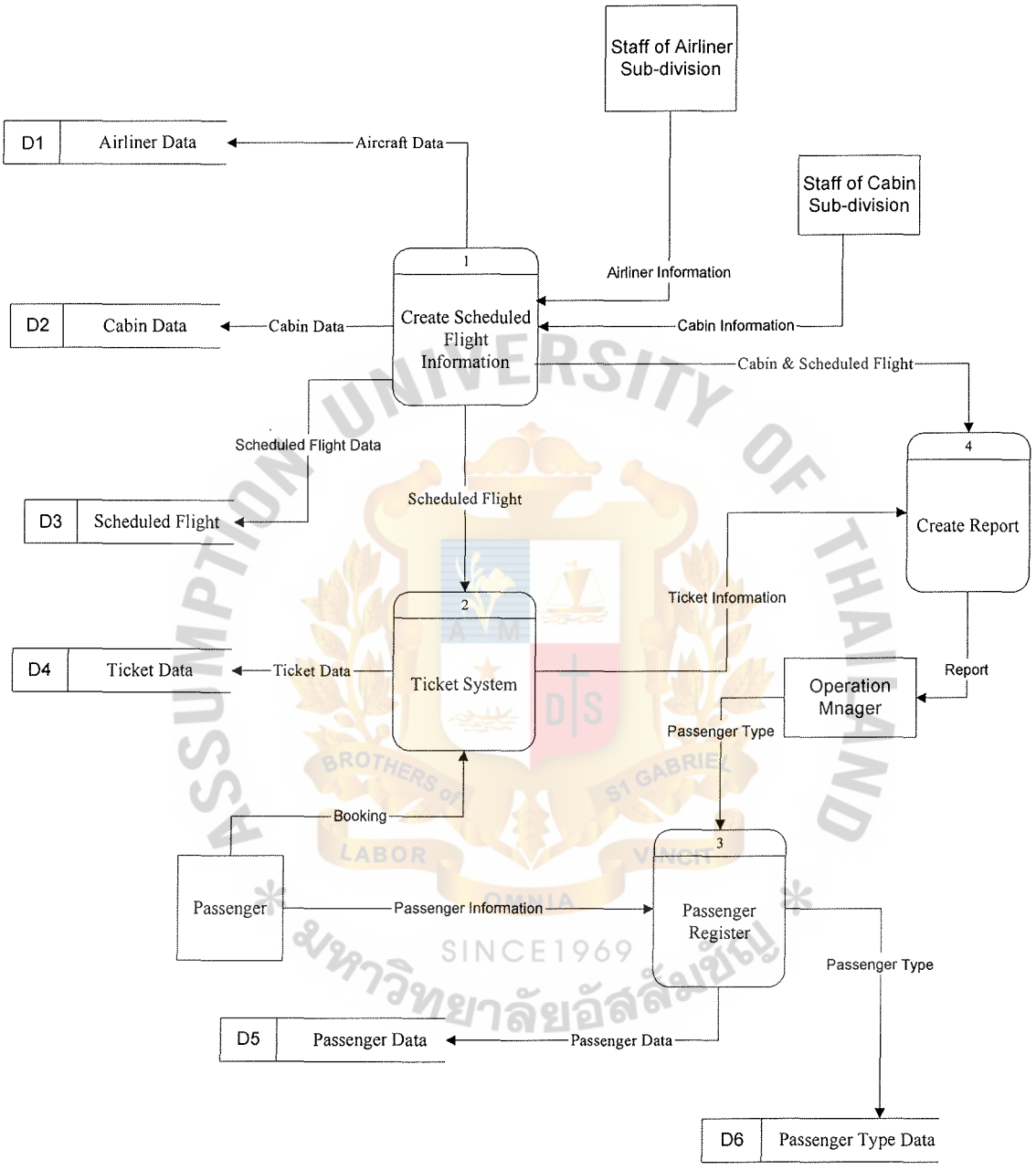

Figure B.1. Data Flow Diagram Level 0 for MIS of Yunnan Airlines of China.

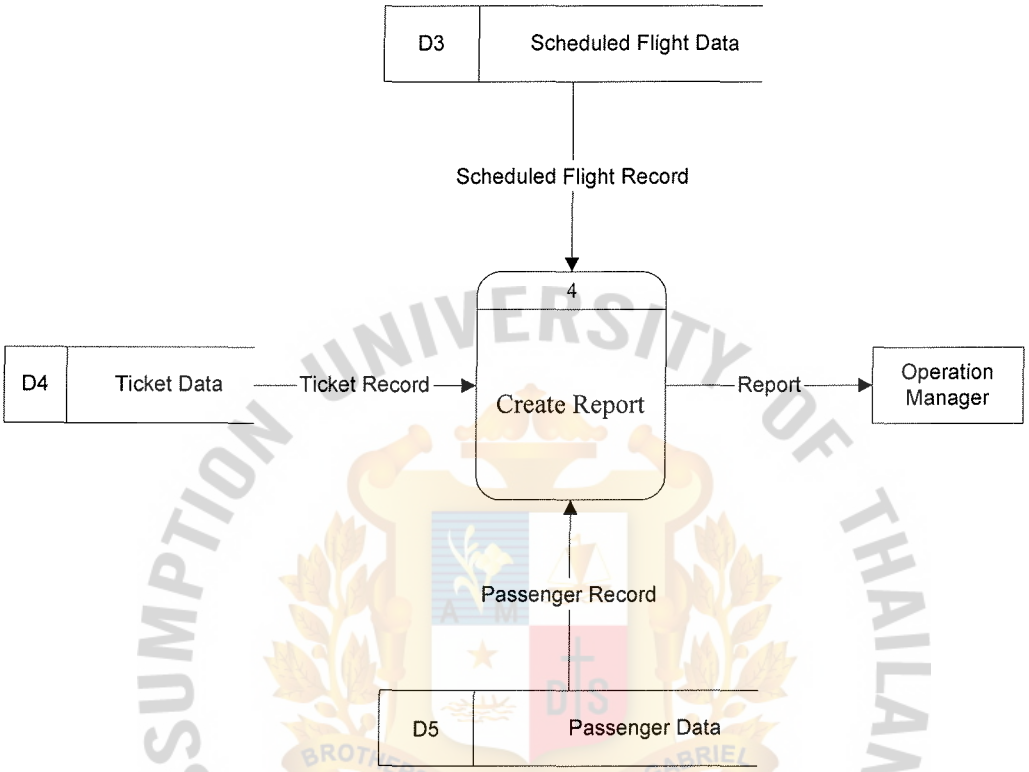

Figure B.2. Data Flow Diagram Level1 Create Report.

ASSUMPTION UNIVERSITY OF THAILAND
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

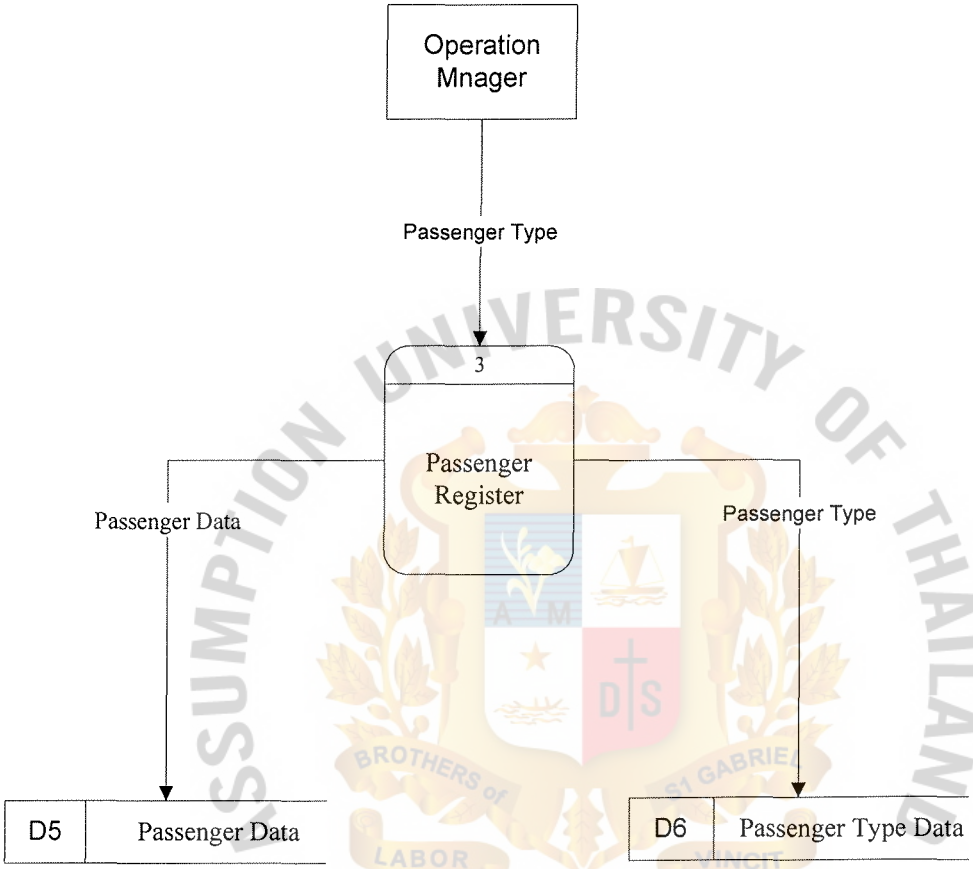

Figure B.3. Data Flow Diagram Level1 Passenger Register.

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

APPENDIX C

STRUCTURE CHART

ASSUMPTION UNIVERSITY OF THAILAND
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

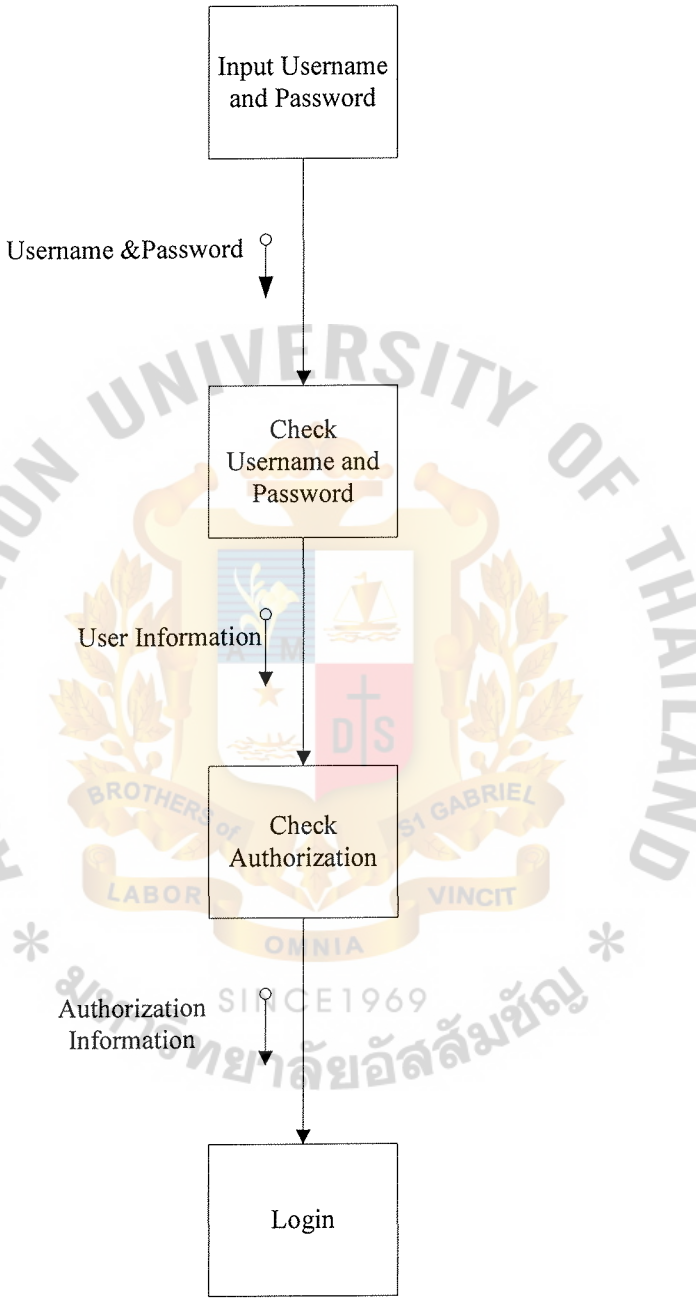

Figure C.1. Structure Chart of Login.

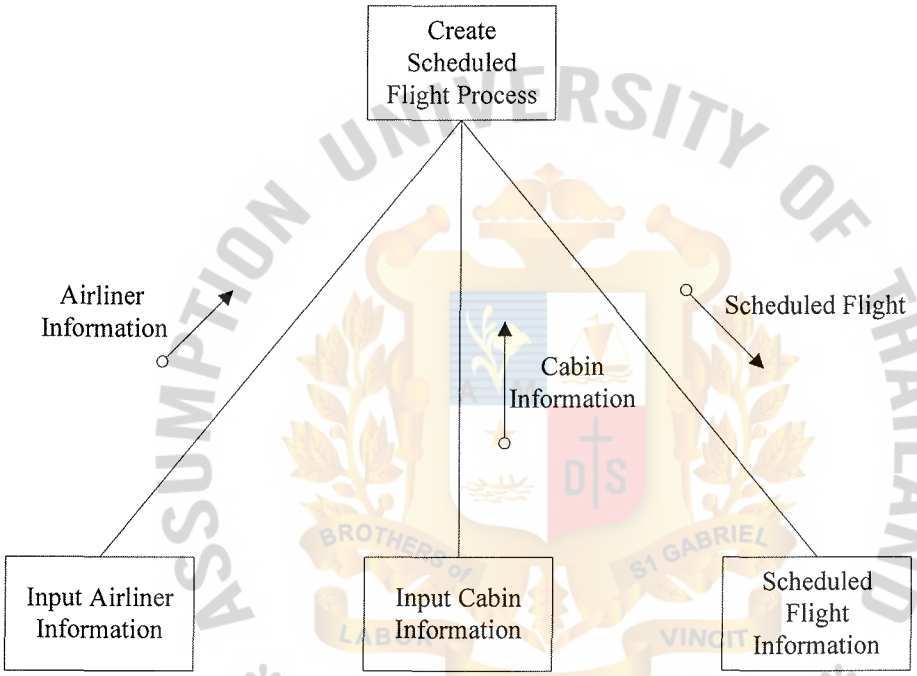

Figure C.2. Structure Chart of Creating Scheduled Flight.

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABORE VINCIT OMNIA
SINCE1969

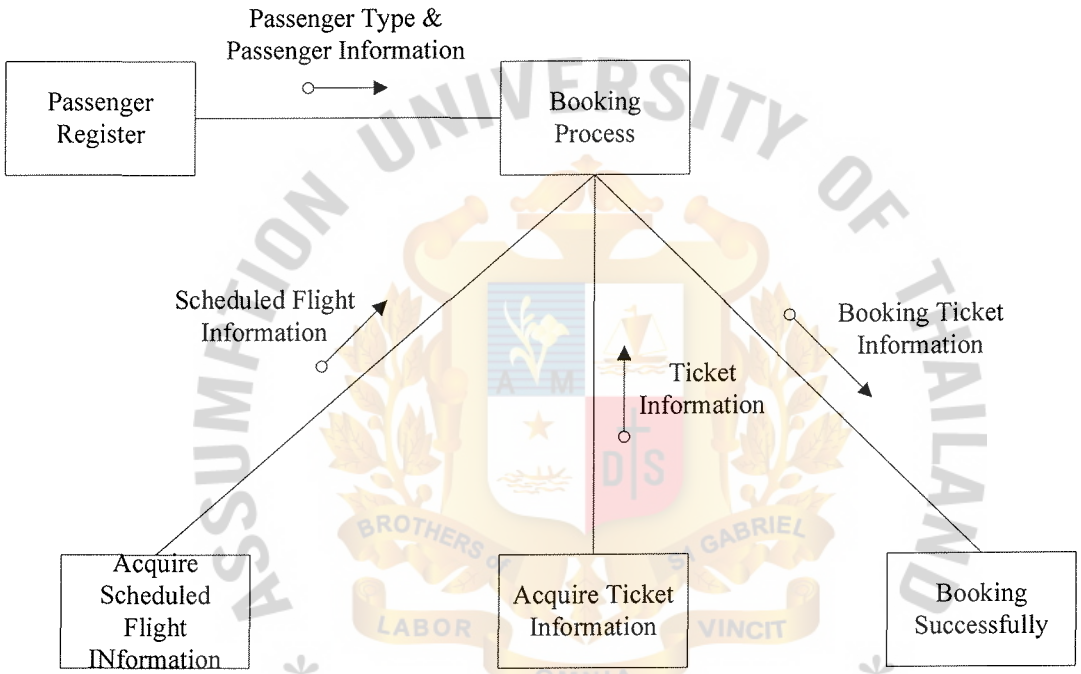

Figure C.3. Structure Chart of Booking.

ASSUMPTION UNIVERSITY OF THAILAND
SINCE1969

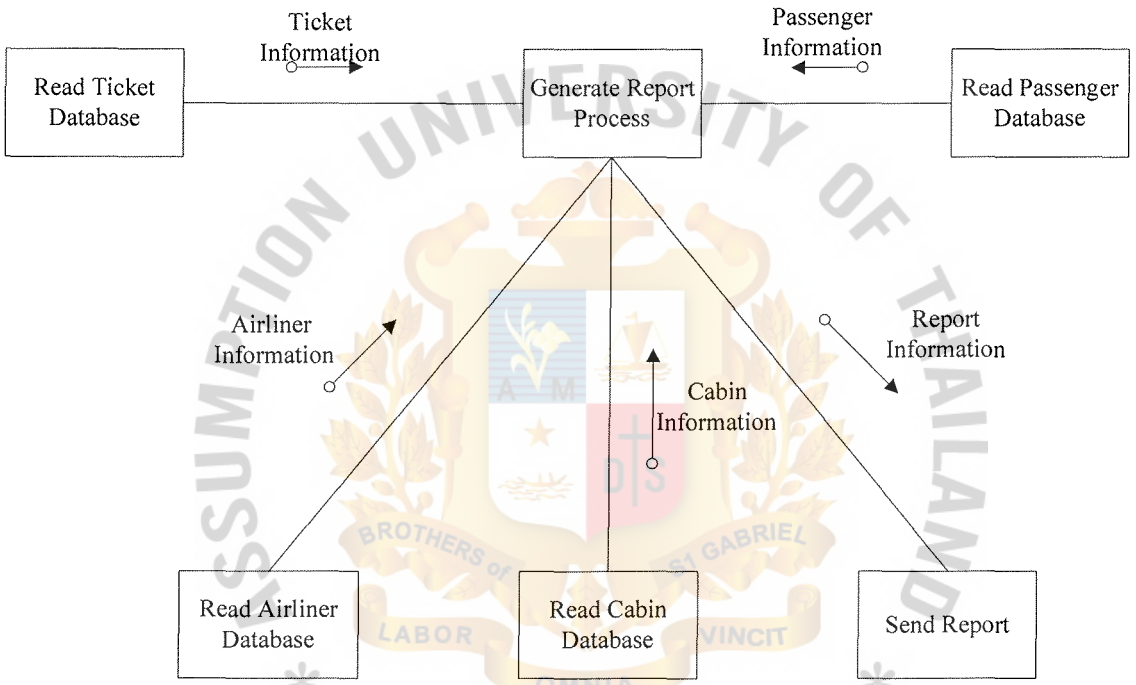

Figure C.4. Structure Chart of Generation Report.

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

APPENDIX D

ENTITY RELATIONSHIP DIAGRAM

ASSUMPTION UNIVERSITY OF THAILAND
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

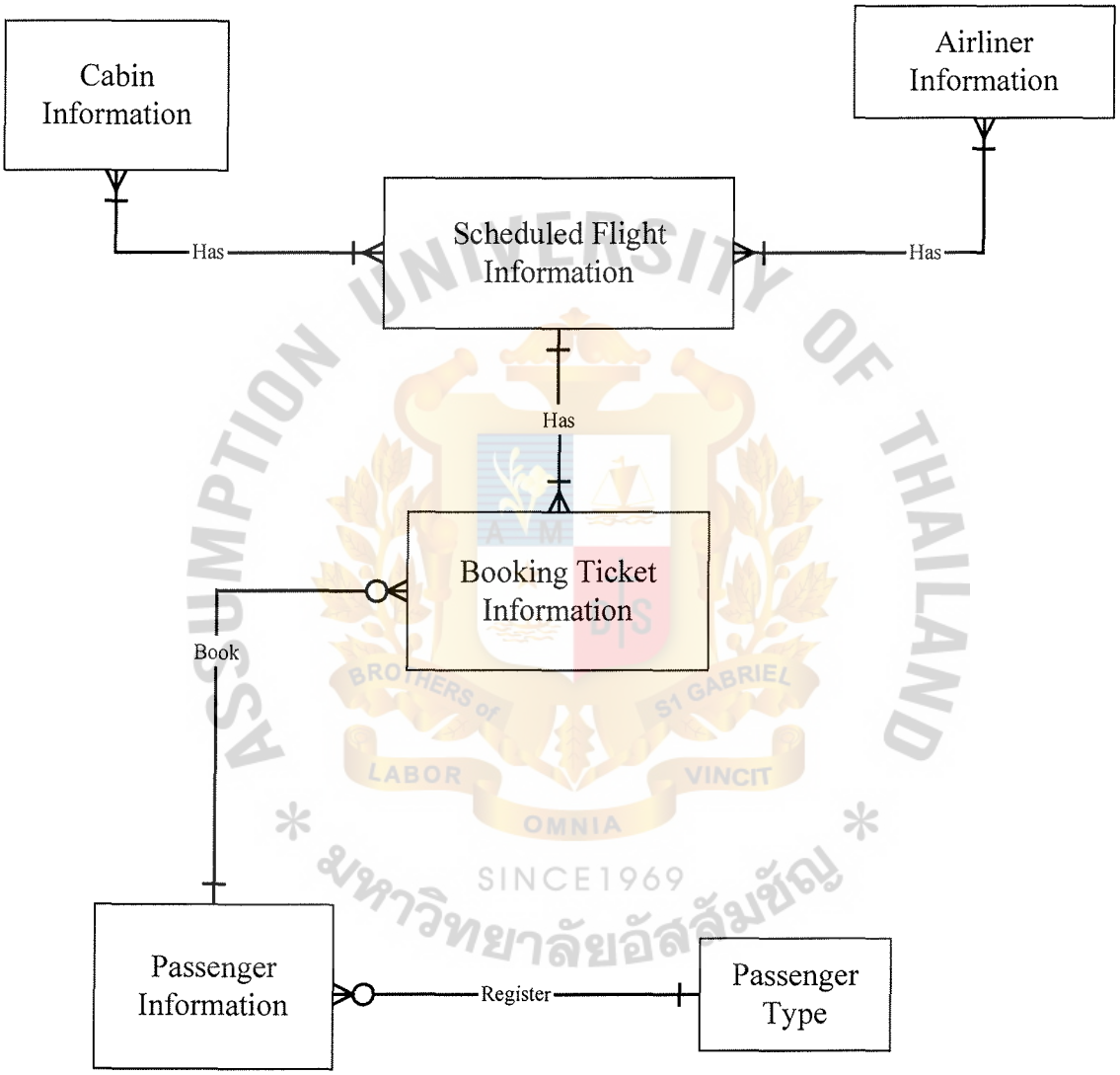

Figure D.1. Entity Relationship Diagram of Proposed System.

St. Gabriel's Library, Au

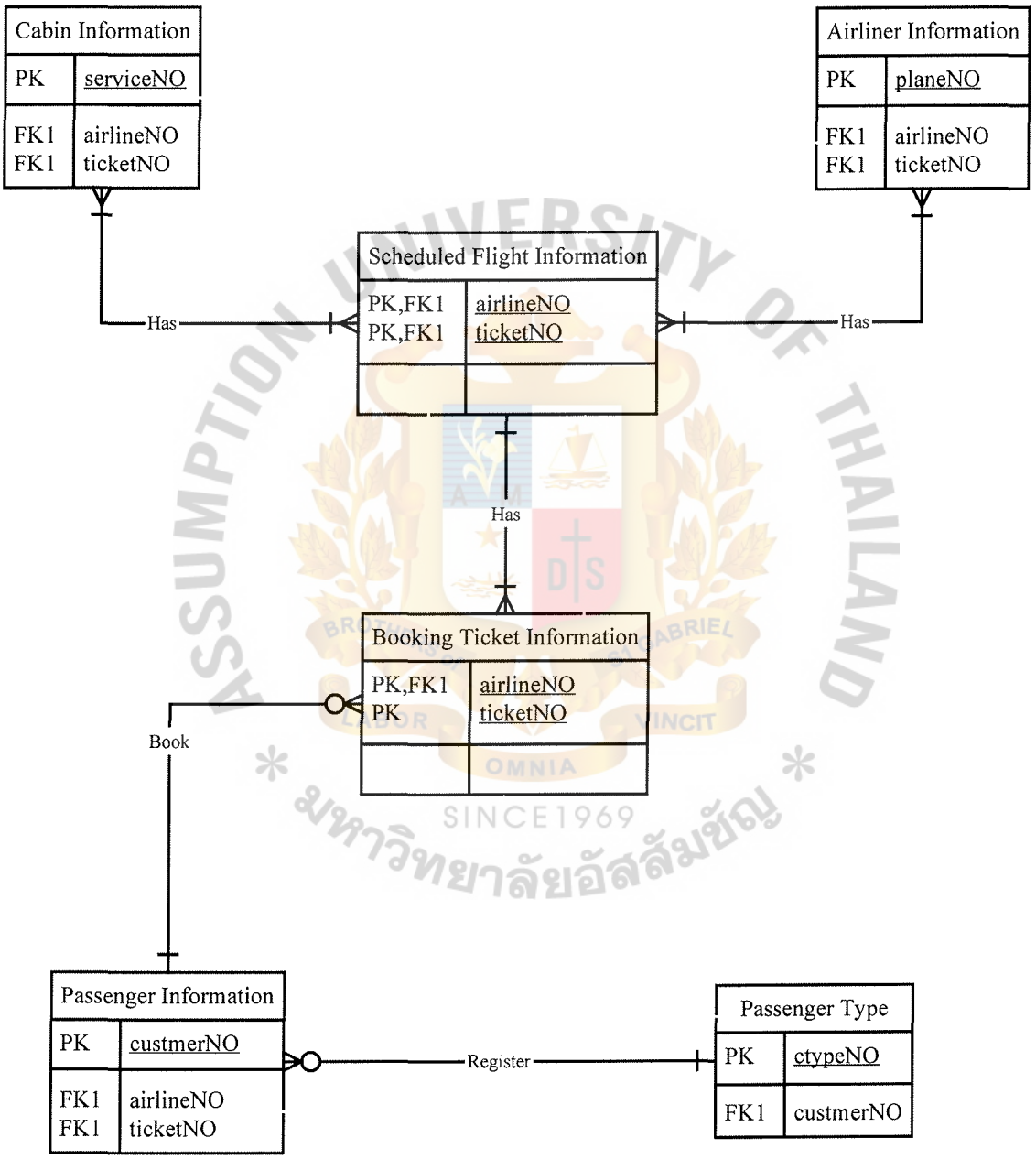

Figure D.2. Key Base Entity Relationship Diagram of Proposed System.

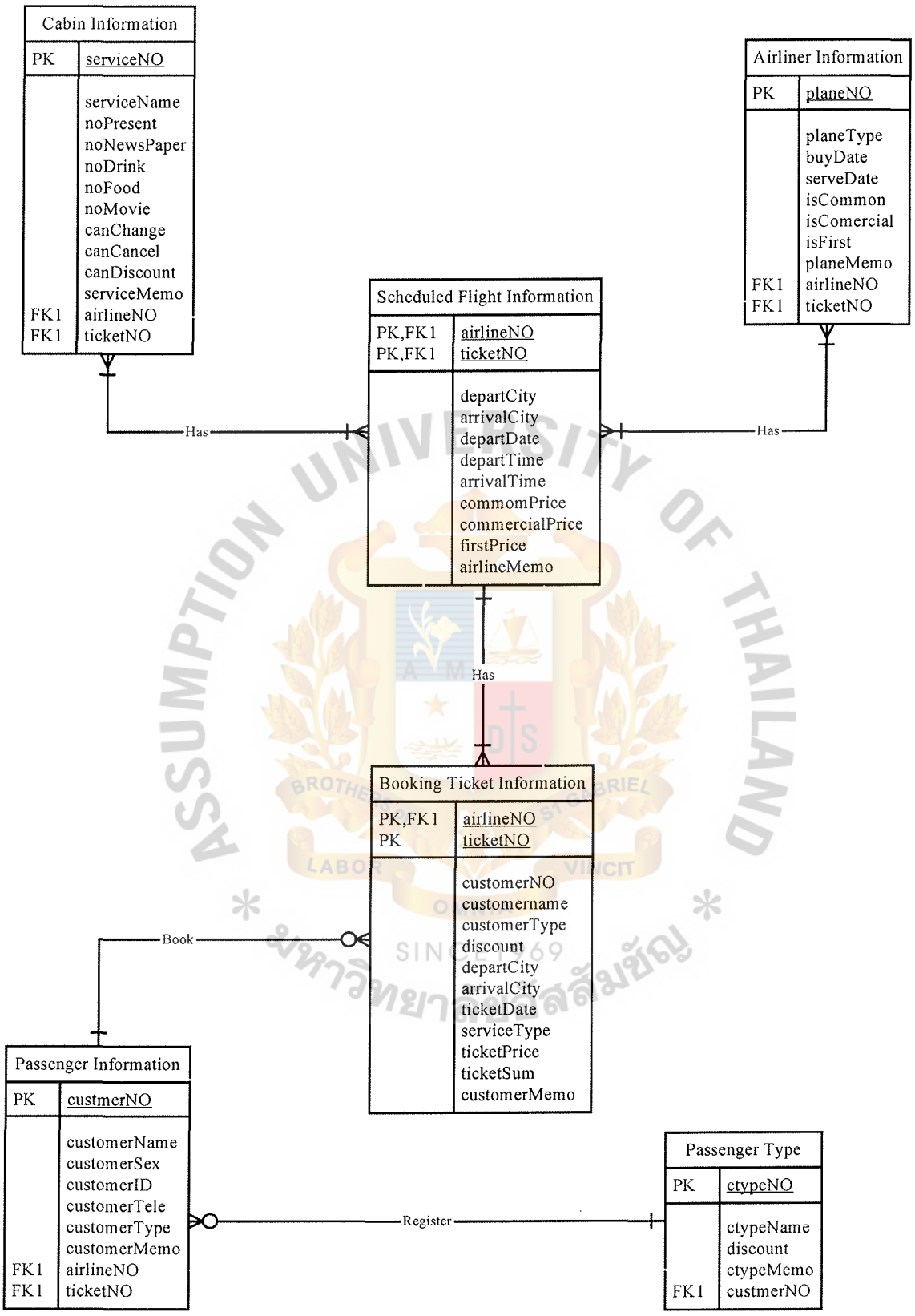

Figure D.3. Fully Attribute Entity Relationship Diagram of Proposed System.

APPENDIX E

TABLE PROPERTIES

ASSUMPTION UNIVERSITY OF THAILAND
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Table E.1. Airliner Information File Layout.

FILE NAME:		PlaneInfo			
SEQ	FIELD	TYPE	WIDTH	DEC	WHETHER NULL
1	planeNO	char	18	0	Not null
2	planeType	char	20	0	Null
3	buyDate	datetime	20	0	Null
4	serveDate	datretime	20	0	Null
5	isCommon	nvarchar	20	0	Null
6	isCommercial	nvarchar	20	0	Null
7	isFirst	nvarchar	20	0	Null
8	planeMemo	text			Null

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Table E.2. Scheduled Flight Information File Layout.

FILE NAME		airlineInfo			
SEQ	FIELD	TYPE	WIDTH	DEC	WHETHER NULL
1	airlineNO	char	14	0	Not Null
2	departCity	nvarchar	50	0	Null
3	arrivalCity	nvarchar	50	0	Null
4	departDate	char	10	0	Null
5	departTime	char	10	0	Null
6	arrivalTime	char	10	0	Null
7	planeNO	char	18	0	Null
8	commonPrice	numeric	18	2	Null
9	commercialPrice	numeric	18	2	Null
10	firstPrice	numeric	18	2	Null
11	airlineMemo	text			Null

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Table E.3. Passenger Information File Layout.

FILE NAME		customerInfo			
SEQ	FIELD	TYPE	WIDTH	DEC	WHETHER NULL
1	customerNO	char	14	0	Not Null
2	customerName	char	50	0	Null
3	customerSex	char	4	0	Null
4	customerID	char	18	0	Null
5	customerTele	char	20	0	Null
6	customerType	char	14	0	Null
7	customerMemo	text			Null

Table E.4. Passenger Type Information File Layout.

FILE NAME		customerType			
SEQ	FIELD	TYPE	WIDTH	DEC	WHETHER NULL
1	ctypeNO	char	14	0	Not Null
2	ctypeName	char	20	0	Null
3	discount	numeric	2	0	Null
4	ctypeMemo	text			Null

ASSUMPTION UNIVERSITY OF THAILAND
SINCE1969

St. Gabriel's Library, Au

Table E.5. Service Information File Layout.

FILE NAME		serviceInfo			
SEQ	FIELD	TYPE	WIDTH	DEC	WHETHER NULL
1	serviceNO	char	4	0	Not Null
2	serviceName	varchar	20	0	Null
3	noPresent	char	4	0	Null
4	noNewsPaper	char	4	0	Null
5	noDrink	char	4	0	Null
6	NoFood	char	4	0	Null
7	noMovie	char	4	0	Null
8	canChange	char	4	0	Null
9	canCancel	char	4	0	Null
10	canDiscount	char	4	0	Null
11	serviceMemo	text			Null

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Table E.6. Ticket Information File Layout.

FILE NAME:		ticketInfo			
SEQ	FIELD	TYPE	WIDTH	DEC	WHETHER NULL
1	ticketNO	char	14	0	Not Null
2	customerNO	char	14	0	Null
3	customerName	char	50	0	Null
4	customerType	char	14	0	Null
5	discount	numeric	18	0	Null
6	airlineNO	char	14	0	Null
7	departCity	nvarchar	50	0	Null
8	arrivalCity	nvarchar	50	0	Null
9	ticketDate	datetime			Null
10	serviceType	char	20	0	Null
11	ticketPrice	numeric	18	2	Null
12	ticketSum	numeric	18	2	Null
13	ticketMemo	text			Null

Table E.7. User Information File Layout.

FILE NAME:		user_Info			
SEQ	FIELD	TYPE	WIDTH	DEC	WHETHER NULL
1	user_ID	char	10	0	Not null
2	user_PWD	char	10	0	Null
3	user_Des	char	20	0	Null

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

ASSUMPTION UNIVERSITY OF THAILAND
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

APPENDIX F

DATABASE IMPLEMENT

```
use master
go

if exists (select * from dbo.sysdatabases where name = 'Ticket')
drop database Ticket
GO

create database Ticket
go
use Ticket
go

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[user_Info]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[user_Info]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[airlineInfo]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[airlineInfo]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[customerInfo]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[customerInfo]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[customerType]')
and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[customerType]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[planeInfo]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[planeInfo]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[serviceInfo]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[serviceInfo]
GO

if exists (select * from dbo.sysobjects where id = object_id(N'[dbo].[ticketInfo]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[ticketInfo]
GO

CREATE TABLE [dbo].[user_Info] (
        [user_ID] [char] (10) COLLATE Chinese_PRC_CI_AS NOT NULL ,
        [user_PWD] [char] (10) COLLATE Chinese_PRC_CI_AS NULL ,
```

```
	[user_Des] [char] (20) COLLATE Chinese_PRC_CI_AS NULL ,
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[airlineInfo] (
	[airlineNO] [char] (14) COLLATE Chinese_PRC_CI_AS NOT NULL ,
	[departCity] [nvarchar] (50) COLLATE Chinese_PRC_CI_AS NULL ,
	[arrivalCity] [nvarchar] (50) COLLATE Chinese_PRC_CI_AS NULL ,
	[departDate] [char] (10) COLLATE Chinese_PRC_CI_AS NULL ,
	[departTime] [char] (10) COLLATE Chinese_PRC_CI_AS NULL ,
	[arrivalTime] [char] (10) COLLATE Chinese_PRC_CI_AS NULL ,
	[planeNO] [char] (18) COLLATE Chinese_PRC_CI_AS NULL ,
	[commonPrice] [numeric](18, 2) NULL ,
	[commercialPrice] [numeric](18, 2) NULL ,
	[firstPrice] [numeric](18, 2) NULL ,
	[airlineMemo] [text] COLLATE Chinese_PRC_CI_AS NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[customerInfo] (
	[customerNO] [char] (14) COLLATE Chinese_PRC_CI_AS NOT NULL ,
	[customerName] [char] (50) COLLATE Chinese_PRC_CI_AS NULL ,
	[customerSex] [char] (4) COLLATE Chinese_PRC_CI_AS NULL ,
	[customerID] [char] (18) COLLATE Chinese_PRC_CI_AS NULL ,
	[customerTele] [char] (20) COLLATE Chinese_PRC_CI_AS NULL ,
	[customerType] [char] (14) COLLATE Chinese_PRC_CI_AS NULL ,
	[customerMemo] [text] COLLATE Chinese_PRC_CI_AS NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[customerType] (
	[ctypeNO] [char] (14) COLLATE Chinese_PRC_CI_AS NOT NULL ,
	[ctypeName] [char] (20) COLLATE Chinese_PRC_CI_AS NULL ,
	[discount] [numeric](2, 0) NULL ,
	[ctypeMemo] [text] COLLATE Chinese_PRC_CI_AS NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[planeInfo] (
	[planeNO] [char] (18) COLLATE Chinese_PRC_CI_AS NOT NULL ,
	[planeType] [char] (20) COLLATE Chinese_PRC_CI_AS NULL ,
	[buyDate] [datetime] NULL ,
	[serveDate] [datetime] NULL ,
	[isCommon] [nvarchar] (20) COLLATE Chinese_PRC_CI_AS NULL ,
	[isCommercial] [nvarchar] (20) COLLATE Chinese_PRC_CI_AS NULL ,
	[isFirst] [nvarchar] (20) COLLATE Chinese_PRC_CI_AS NULL ,
	[planeMemo] [text] COLLATE Chinese_PRC_CI_AS NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[serviceInfo] (
	[serviceNO] [char] (4) COLLATE Chinese_PRC_CI_AS NOT NULL ,
	[serviceName] [nvarchar] (20) COLLATE Chinese_PRC_CI_AS NULL ,
	[noPresent] [char] (4) COLLATE Chinese_PRC_CI_AS NULL ,
	[noNewsPaper] [char] (4) COLLATE Chinese_PRC_CI_AS NULL ,
	[noDrink] [char] (4) COLLATE Chinese_PRC_CI_AS NULL ,
	[noFood] [char] (4) COLLATE Chinese_PRC_CI_AS NULL ,
	[noMovie] [char] (4) COLLATE Chinese_PRC_CI_AS NULL ,
	[canChange] [char] (4) COLLATE Chinese_PRC_CI_AS NULL ,
	[canCancel] [char] (4) COLLATE Chinese_PRC_CI_AS NULL ,
	[canDiscount] [char] (4) COLLATE Chinese_PRC_CI_AS NULL ,
	[serviceMemo] [text] COLLATE Chinese_PRC_CI_AS NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[ticketInfo] (
	[ticketNO] [char] (14) COLLATE Chinese_PRC_CI_AS NOT NULL ,
	[customerNO] [char] (14) COLLATE Chinese_PRC_CI_AS NULL ,
	[customerName] [char] (50) COLLATE Chinese_PRC_CI_AS NULL ,
	[customerType] [char] (14) COLLATE Chinese_PRC_CI_AS NULL ,
	[discount] [numeric](18, 0) NULL ,
	[airlineNO] [char] (14) COLLATE Chinese_PRC_CI_AS NULL ,
	[departCity] [nvarchar] (50) COLLATE Chinese_PRC_CI_AS NULL ,
	[arrivalCity] [nvarchar] (50) COLLATE Chinese_PRC_CI_AS NULL ,
	[ticketDate] [datetime] NULL ,
	[serviceType] [nvarchar] (20) COLLATE Chinese_PRC_CI_AS NULL ,
	[ticketPrice] [numeric](18, 2) NULL ,
	[ticketSum] [numeric](18, 2) NULL ,
	[ticketMemo] [text] COLLATE Chinese_PRC_CI_AS NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

SINCE1969
มหาวิทยาลัยอัสสัมชัญ

APPENDIX G

DATA DICTIONARY

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS of ST GABRIEL
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

A

airlineMemo	The Remark of Scheduled Flight
airlineNO	The Number of Scheduled Flight
arrivalCity	The Arrival City
arrivalTime	The Time of Arrival

B

buyDate	The Time of Buying

C

canCancel	Whether Can Withdraw The Ticket
canChange	Whether Can Change the Ticket
canDiscount	Whether Offer Discount
commercialPrice	The price of Business Class Cabin
commonPrice	The Price of Economy Class Cabin
ctypeMemo	The Remark of Passenger Type
ctypeName	The Name of Type
ctypeNO	The Number of Passenger Type
customerID	The ID of Passenger
customerMemo	The Remark of Passenger
customerName	The Name of Passenger
customerNO	The Passenger Number
customerSex	The Gender of Passenger
customerTele	The Telephone of Passenger
customerType	The Type of Passenger

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST. GABRIEL
VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

D

departCity	The Departure City
departDate	The Date of Departure
departTime	The Time of Departure
discount	The Discount

F

firstPrice	The Price of First Class Cabin

I

isCommercial	The Amount of The Business Class Cabin
isCommon	The Amount of The Economy Class Cabin
isFirst	The Amount of The First Class Cabin

N

noDrink	Whether Offer Drink
noFood	Whether Offer Food
noMovie	Whether Offer Movie
noNewsPaper	Whether Offer Newspaper
noPresent	Whether Offer Present

P

planeMemo	The Remark of Airliner
planeNO	The number of Airliner
planeType	The Type of Airliner

S

serveDate	The Time Starting to Service
serviceMemo	The Remark of the Service

ASSUMPTION UNIVERSITY OF THAILAND
มหาวิทยาลัยอัสสัมชัญ
SINCE1969

serviceName	The Name of Service
serviceNO	The Number of Service
serviceType	The Type of the Service
T	
ticketDate	The Date of Issuing Ticket
ticketMemo	The Remark of Ticket
ticketNO	The Number of The Ticket
ticketPrice	The Airfare (The Price of the Ticket)
ticketSum	The Total Amount of the Ticket
U	
user_Des	The Description of User (Authority of the Operation Right)
user_ID	The User Name (Use for Access System)
user_PWD	The Password of User (Use for access System)

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

APPENDIX H

PROCESS SPECIFICATION

ASSUMPTION UNIVERSITY OF THAILAND
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Table H.1. Process Specification of Login.

Items	Description
Process Name	User Login
Data In	Username by User Input Password by User Input Username From Database Password From Database
Data Out	Acceptance or Rejection
Process	1) Receive Username & Password From User 2) Read Username & password From Database 3) Compare Whether It Is Correct 4) Send the Acceptance or Rejection Information to System
Attachment	1) User 2) User Database

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Table H.2. Process Specification of Generate the Scheduled Flight.

Items	Description
Process Name	Generate the Scheduled Flight
Data In	Airliner Information Cabin Information Service Information
Data Out	Scheduled Flight
Process	1) Receive Airliner Information from the Airliner Sub-division 2) Receive Cabin Information From the Cabin Sub-division 3) Receive Service Information From the Cabin Sub-division 4) Generate the Scheduled Flight 5) Send the Scheduled Flight Information to Ticket Subdivision
Attachment	1) Airliner Database 2) Cabin Database 3) Scheduled Flight Database 4) Airliner Sub-division 5) Cabin Sub-division 6) Ticket Sub-division

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

St. Gabriel's Library, Au

Table H.3. Process Specification of Passenger Register.

Items	Description
Process Name	Passenger Register
Data In	Passenger Information Passenger Type
Data Out	Passenger Information
Process	1) Receive Passenger Information From Passenger 2) Read Passenger Type from Passenger Database 3) Send Passenger Information to Passenger Database
Attachment	1) Passenger 2) Passenger Database 3) Passenger Type Database 4) Ticket Sub-division

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Table H.4. Process Specification of Booking.

Items	Description
Process Name	Booking
Data In	Scheduled Flight Information Passenger Information
Data Out	Ticket Information
Process	1) Receive Booking From Passenger 2) Read Scheduled Flight Information From Scheduled Flight Database 3) Read Passenger Information From Passenger Information Database 4) Generate Ticket Information 5) Print the Ticket Information to Passenger 6) Send Ticket Information to Ticket Database
Attachment	1) Passenger 2) Ticket Sub-division 3) Printer 4) Passenger Database 5) Schedule Flight Database 6) Ticket Database

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

Table H.5. Process Specification of Creating Report.

Items	Description
Process Name	Creating Report
Data In	Ticket Information Airliner Information Cabin Information Service Information Scheduled Flight Information Passenger Information
Data Out	Report Detail Information
Process	1) Read the Airliner Information 2) Read the Cabin Information 3) Read the Service Information 4) Read Scheduled Flight Information 5) Read Passenger Information 6) Read Ticket Information 7) Generate the Report 8) Send the Report to Operation Manager
Attachment	1) Ticket Database 2) Airliner Database 3) Cabin Database 4) Service Database 5) Scheduled Flight Database 6) Passenger Database 7) Operation Manager

ASSUMPTION UNIVERSITY OF THAILAND
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

APPENDIX I

USER INTERFACE

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS of ST GABRIEL
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

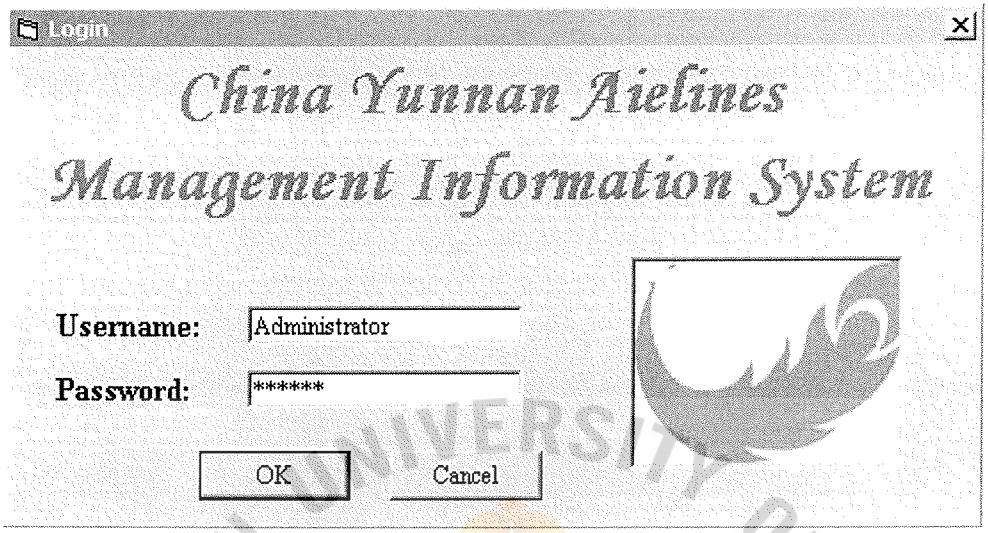

Figure I.1. Login Interface.

Figure I.2. Main Interface.

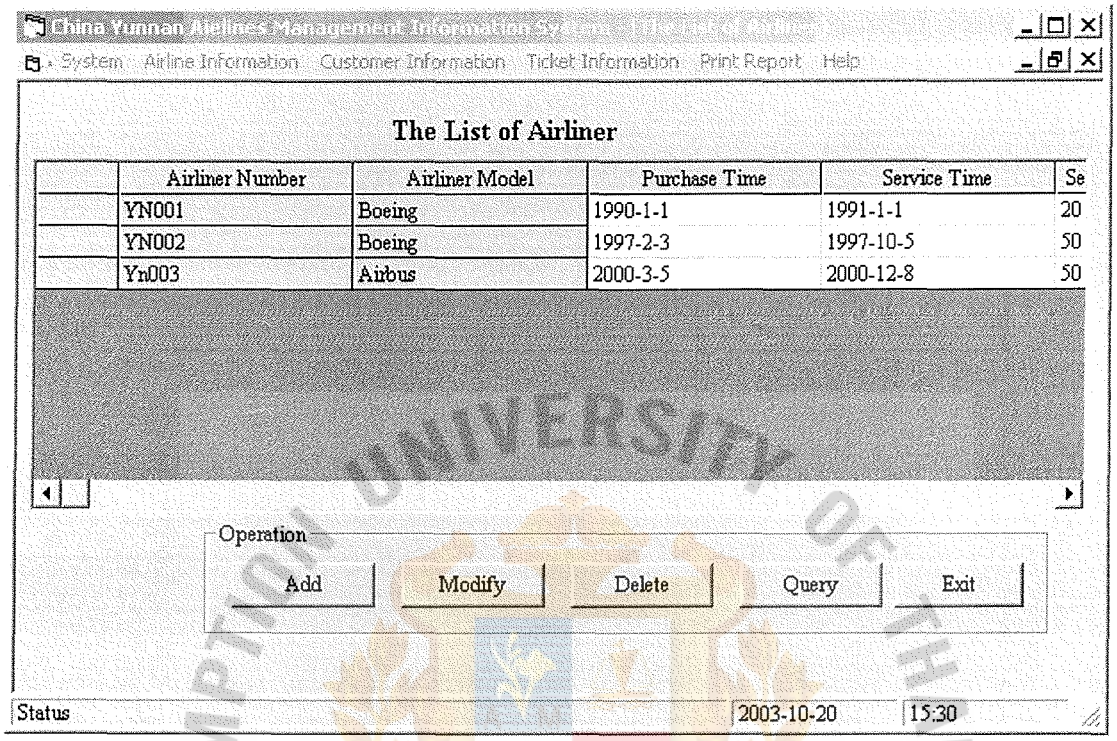

Figure I.3. Managing Airliner Interface.

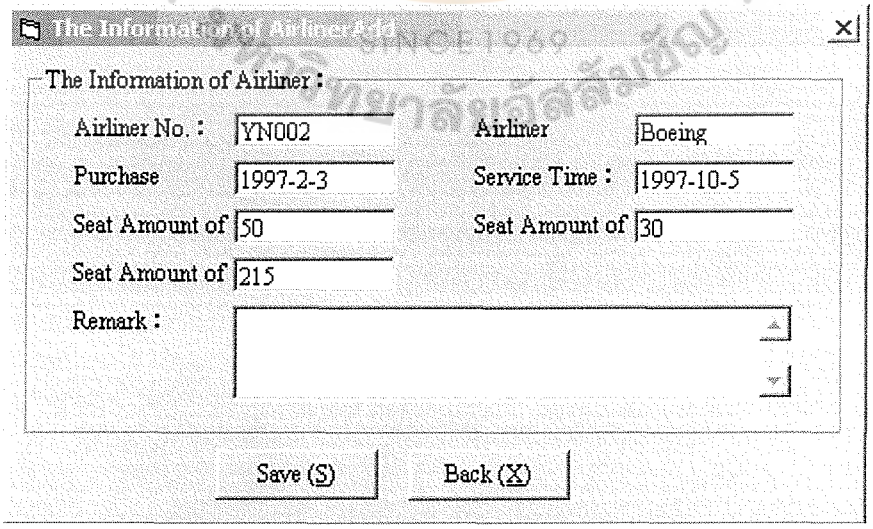

Figure I.4. Input the Airliner Information.

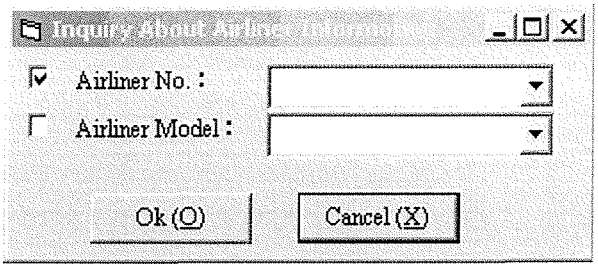

Figure I.5. Inquiry the Airliners Information.

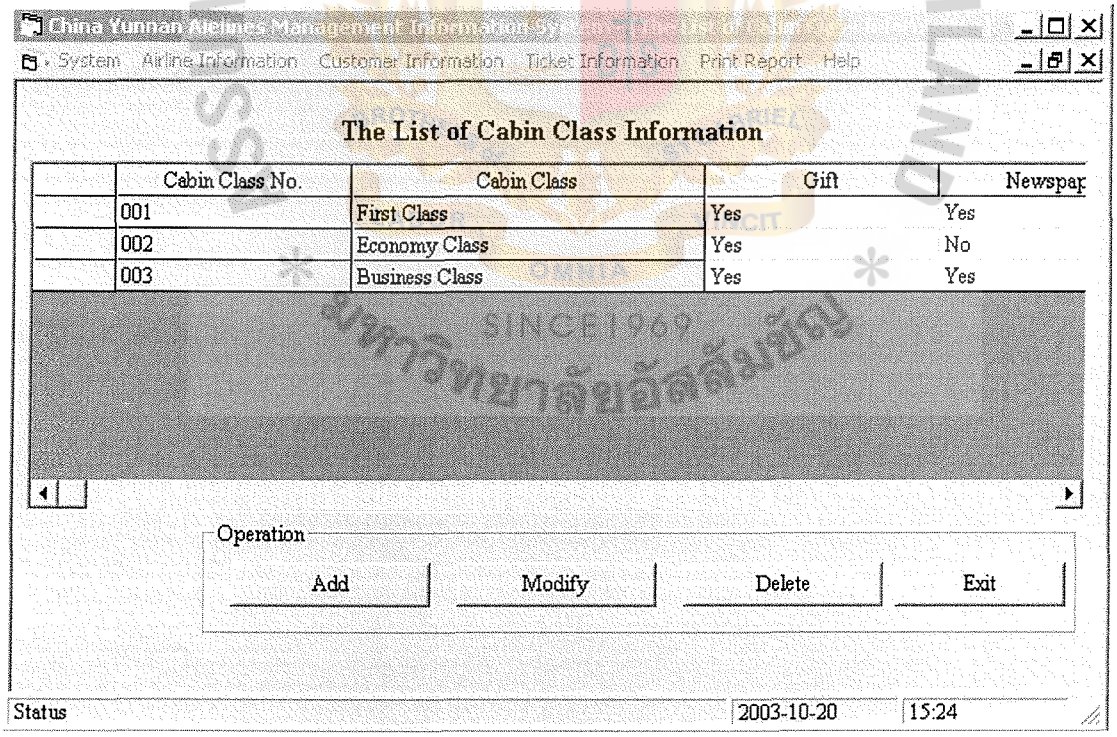

Figure I.6. Managing the Cabin Interface.

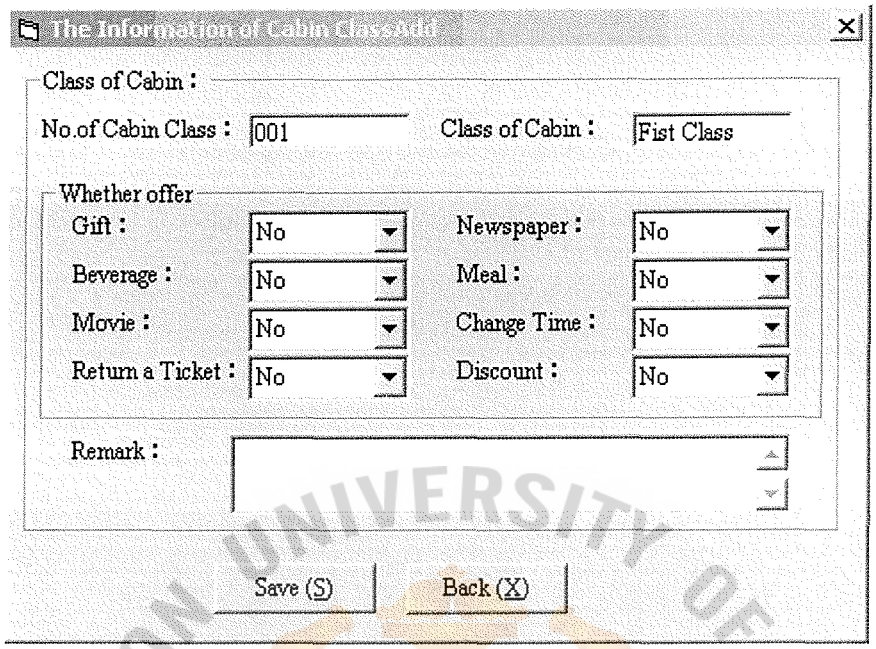

Figure I.7. Input the Cabin Information.

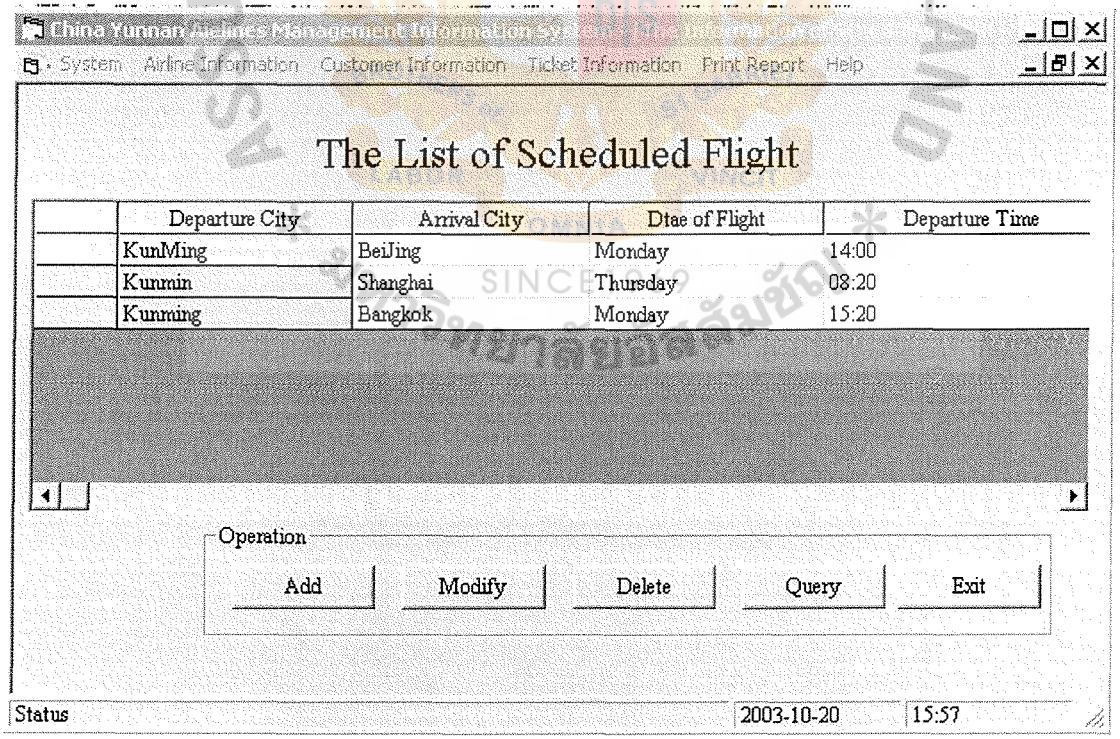

Figure I.8. Managing Scheduled Flight Interface.

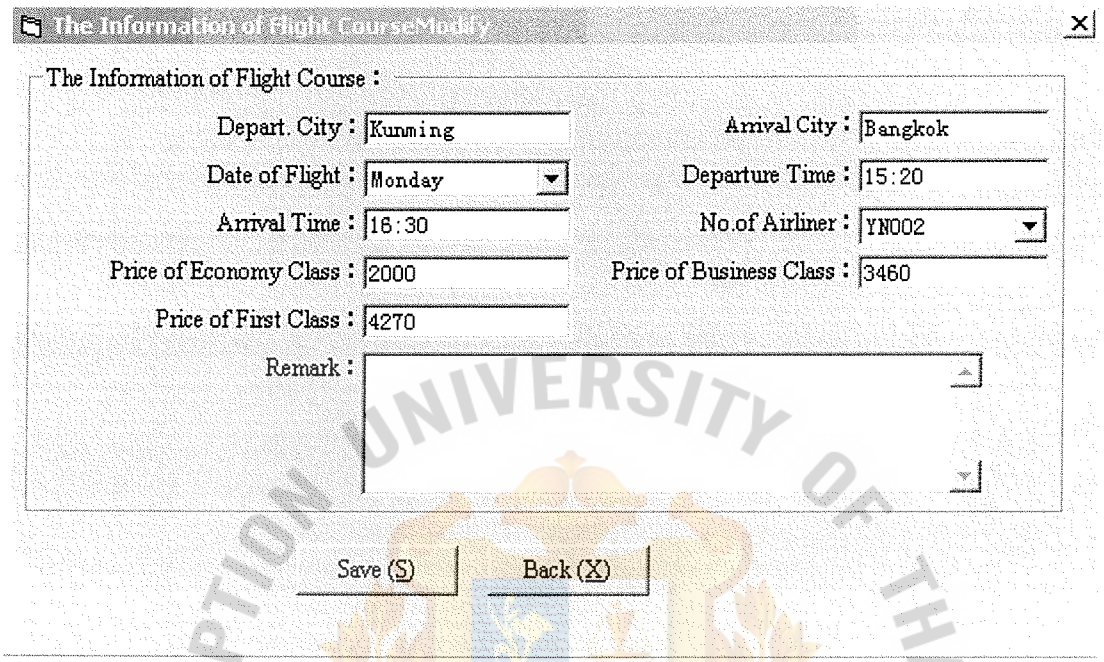

Figure I.9. Input the Scheduled Flight Information.

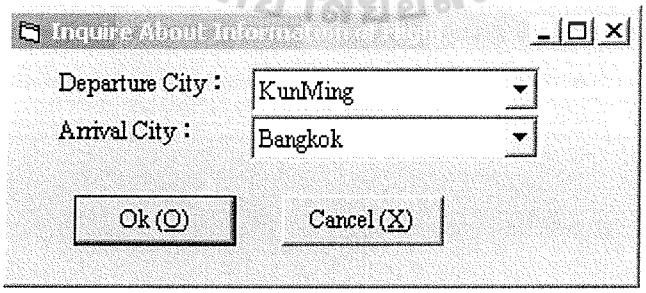

Figure I.10. Inquiry Scheduled Flight Information.

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

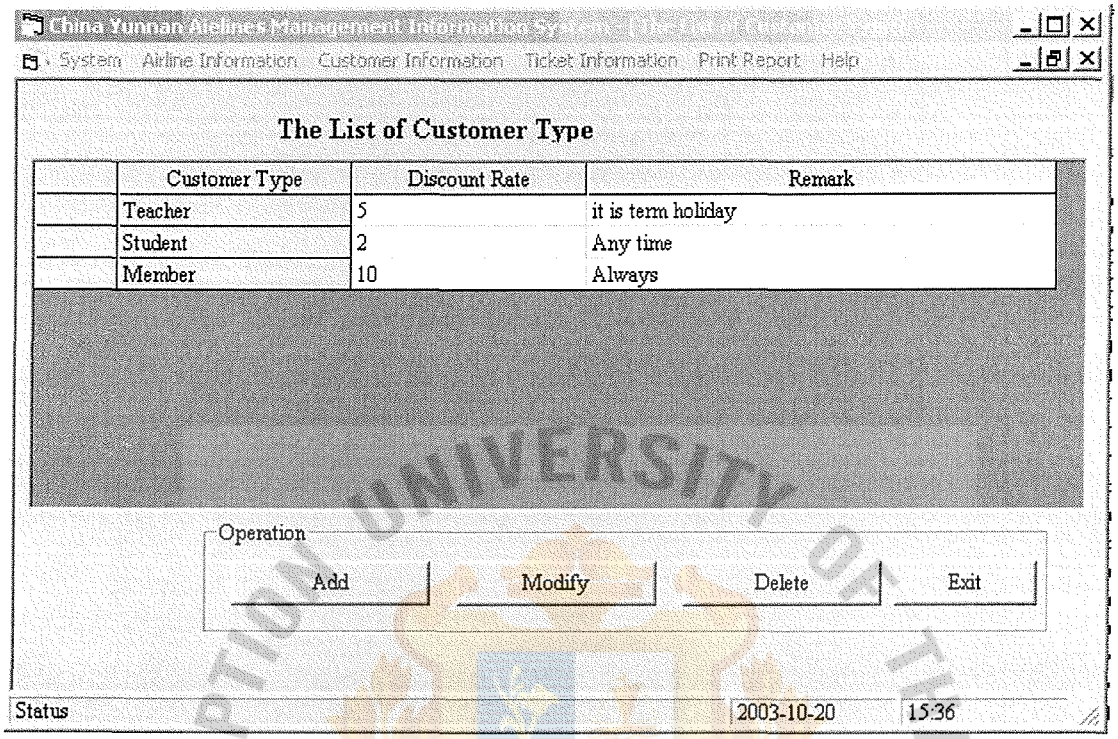

Figure I.11. Managing Passenger Type Interface.

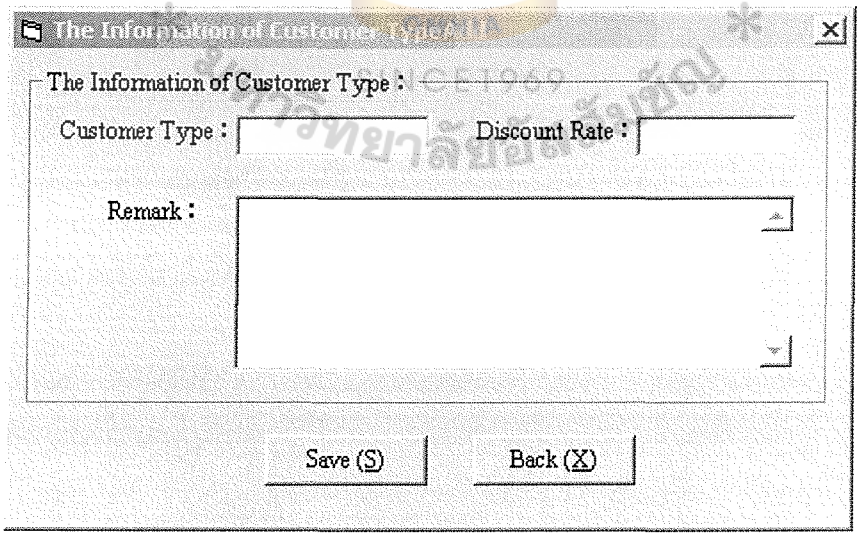

Figure I.12. Input the Passenger Type Information.

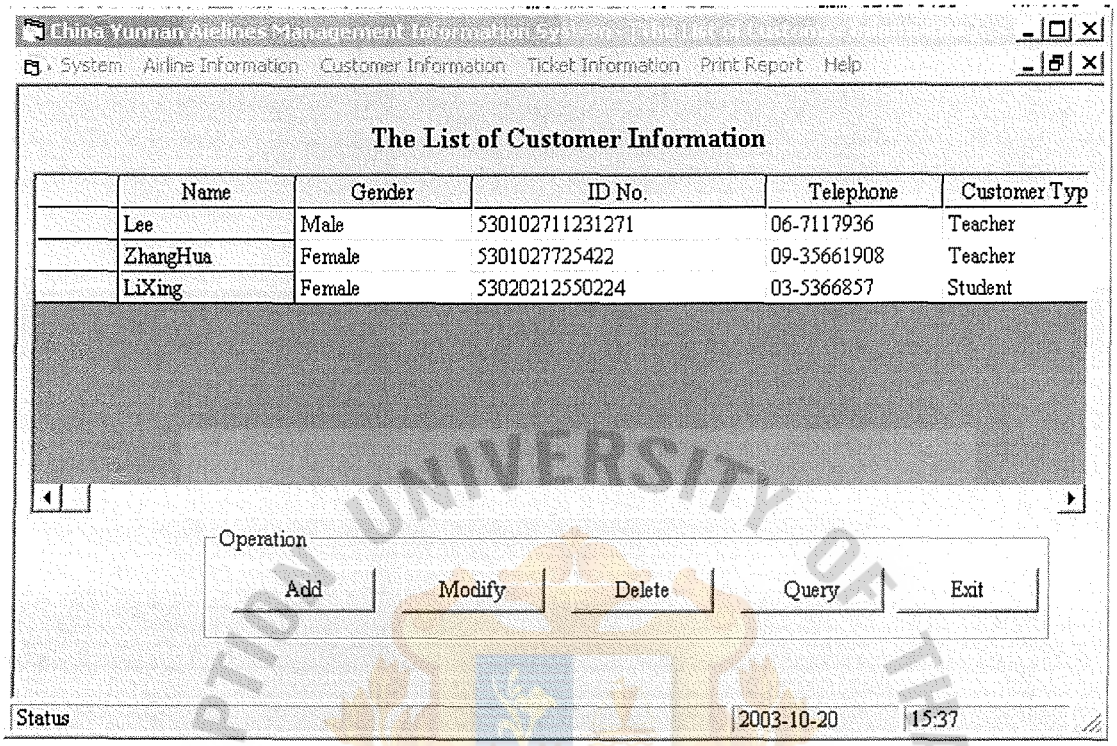

Figure I.13. Managing Passenger Information Interface.

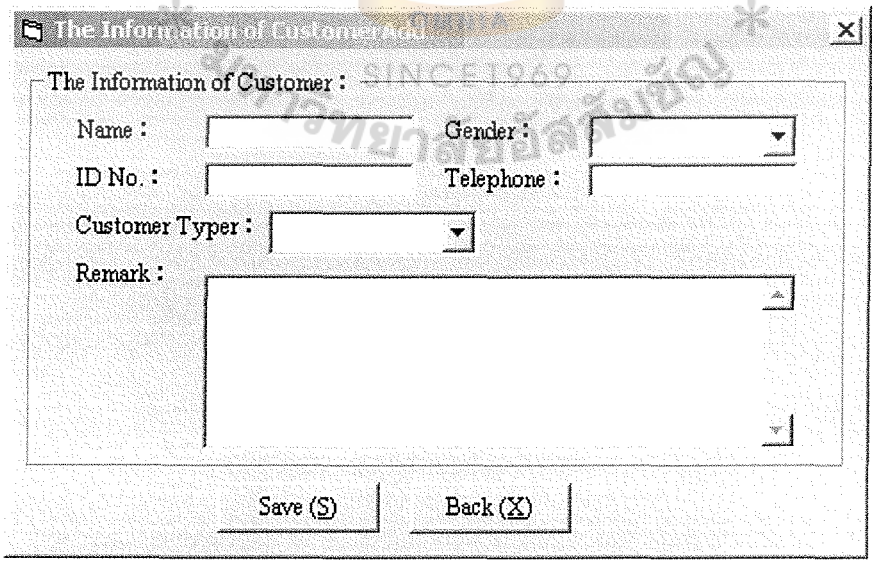

Figure I.14. Input Passenger Information.

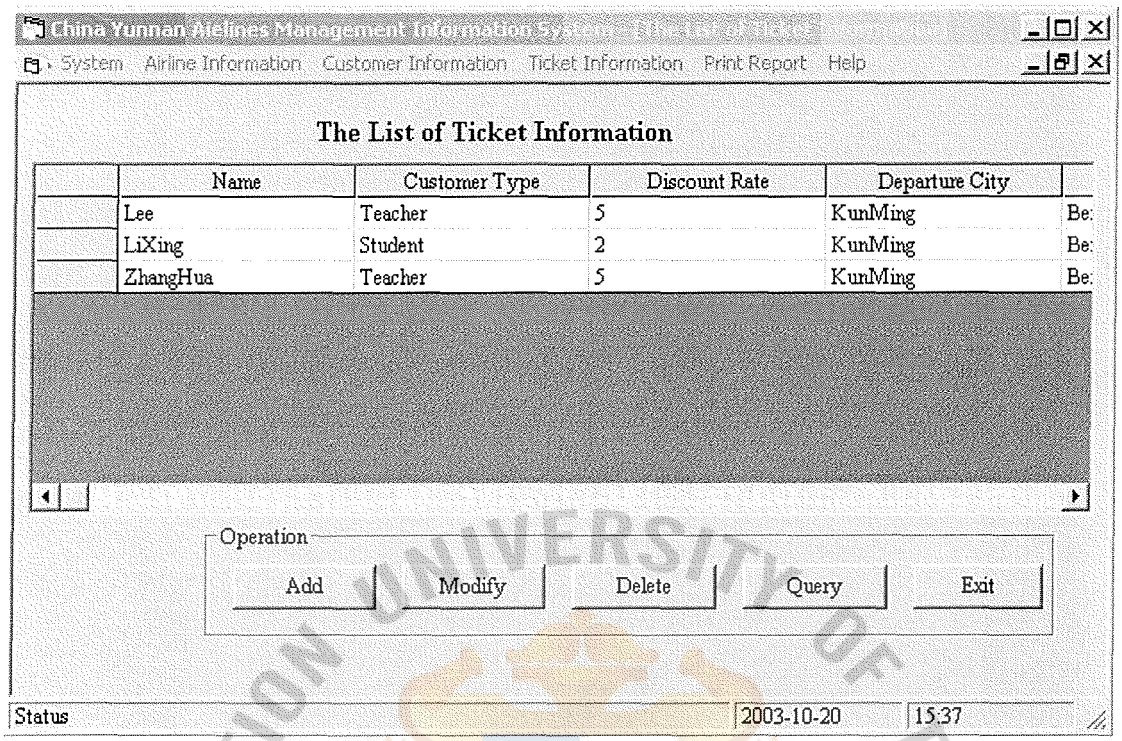

Figure I.15. Managing Ticket Information Interface.

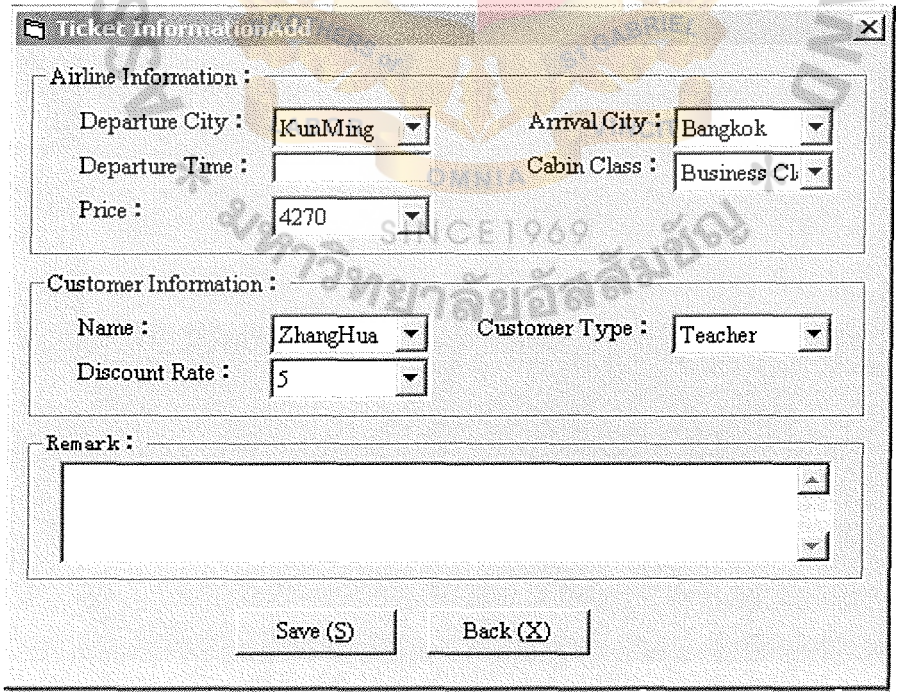

Figure I.16. Booking the Ticket.

St. Gabriel's Library, Au

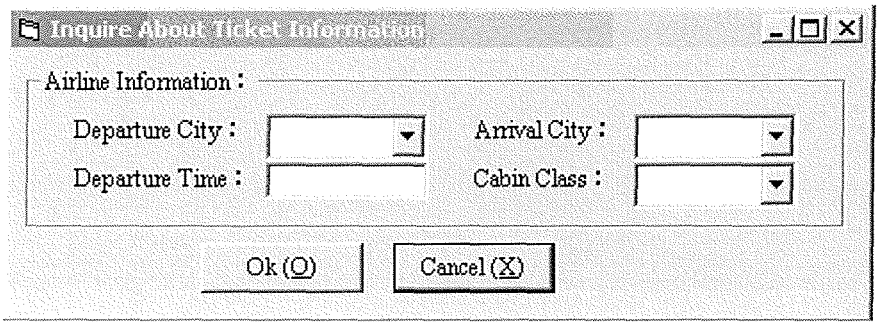

Figure I.17. Inquiry the Ticket Information.

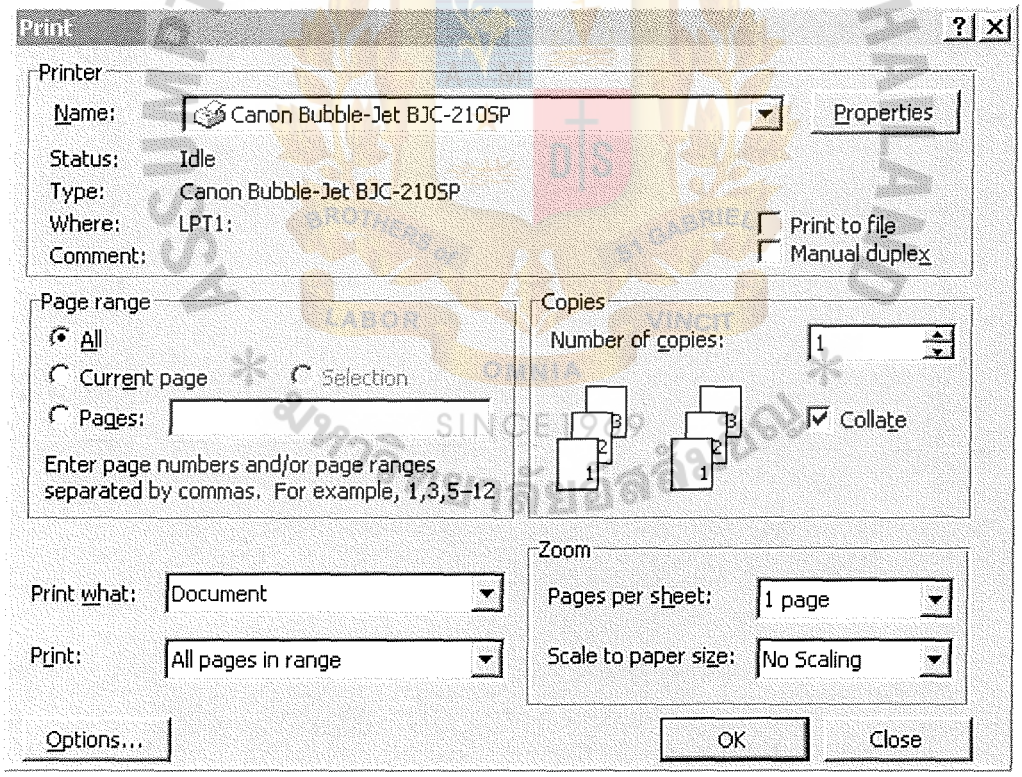

Figure I.18. The Dialog of Printing.

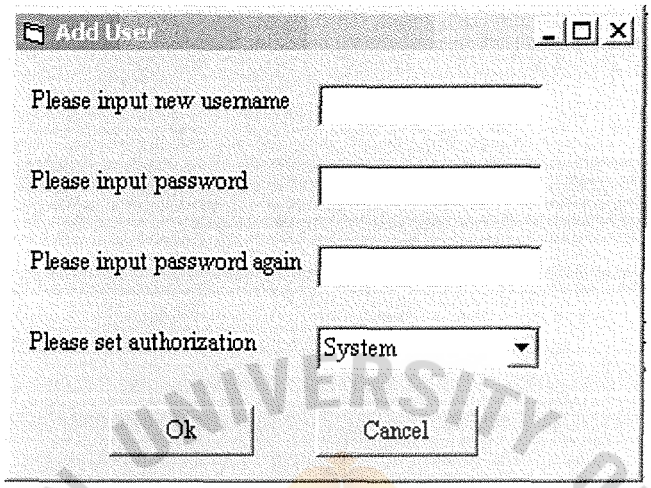

Figure I.19. Add a New Operator.

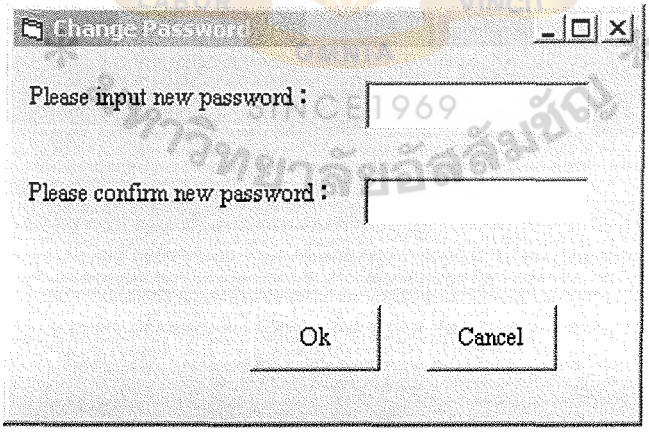

Figure I.20. Change the Password of Operator.

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR VINCIT
SINCE 1969

APPENDIX J

OUTPUT REPORTS

ASSUMPTION UNIVERSITY OF THAILAND
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

ISSUED BY

CONJUCTION TICKET

ORIGIN/DESTINATION

ENDORSMENTS/RESTRICTIONS

DATE OF ISSUE

AIRLINE DATA

NAME OF PASSENGER (NOT TRANSFERABLE)

ISSUED IN EXCHANGE FOR

中国云南航空公司
CHINA YUNNAN AIRLINES

X/O	NOT GOOD FOR PASSENGER FROM	CARRIER	FLIGHT	CLASS	DATE	STATUS	FARE BASIS	NOT VALID BEFORE	NOT VALID AFTER	ALLOW
	TO									
	TO									
	TO									
	TO									
		BAGGAGE CHECKED UNCHECKED →	PCS	WT	UNCHECKED	PCS	WT	UNCHECKED	PCS	WT

FARE

EQUIV. FARE PAID

TAX/FEE/CHARGE

TAX/FEE/CHARGE

TAX/FEE/CHARGE

FORM OF PAYMENT	APP.CODE	TOUR CODE

TOTAL

A/L AGT INFO

CONTROL NO.

CPN	AIRLINE CODE	FORM AND SERIAL NUMBER	CK	ORIGIONAL ISSUE

ASSUMPTION UNIVERSITY OF THAILAND

Figure J.1. The Ticket.

Yunnan Airlines Of China

中国云南航空公司
CHINA YUNNAN AIRLINES

Airliner Report

SEQ	Airliner No.	Airliner Type	Buy Date	Serve Date	Amount of Economy Class	Amount of Business Class	Amount of First Class	Remark
Total:								

Figure J.2. The Report of Airliner.

Yunnan Airlines of China

Scheduled Flight Report

SEQ	Scheduled Flight No.	Departure City	Arrival City	Departure Date	Departure Time	Arrival Time	Airliner No.	Price of Economy Class	Price of Business Class	Price of First Class	Remark
Total:											

Figure J.3. The Report of Scheduled Flight.

Yunnan Airlines of China

Ticket Report

SEQ	Ticket No.	Passenger No.	Passenger Name	Passenger Type	Discount	Airline No.	Depart City	Arrival City	Ticket Date	Service Type	Ticket Price	Ticket Sum	Remark
Total:													

Figure J.4. The Report of Ticket.

ASSUMPTION UNIVERSITY OF THAILAND
SINCE 1969

APPENDIX K

SOURCE CODE

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

<Public Fuction>

```
Public fMainForm As frmMain
Public UserType As String
Public UserName As String
Public gintSmode As Integer 'for service
Public gintPmode As Integer 'for plane
Public gintAmode As Integer 'for airline
Public gintTmode As Integer 'for customertype
Public gintCmode As Integer 'for customer
Public gintKmode As Integer 'for ticket

Sub Main()
   Dim fLogin As New frmLogin
   fLogin.Show vbModal
   If Not fLogin.OK Then
      'Login Failed so exit app
      End
   End If
   Unload fLogin

   Set fMainForm = New frmMain
   fMainForm.Show
End Sub

Public Function ConnectString() _
  As String
'returns a DB ConnectString
  ConnectString = "FileDSN=ticket.dsn;UID=sa;PWD="
End Function
Public Function ExecuteSQL(ByVal SQL _
  As String, MsgString As String) _
  As ADODB.Recordset
'executes SQL and returns Recordset
```

```
    Dim cnn As ADODB.Connection
    Dim rst As ADODB.Recordset
    Dim sTokens() As String

    On Error GoTo ExecuteSQL_Error

    sTokens = Split(SQL)
    Set cnn = New ADODB.Connection
    cnn.Open ConnectString
    If InStr("INSERT,DELETE,UPDATE", _
      UCase$(sTokens(0))) Then
      cnn.Execute SQL
      MsgString = sTokens(0) & _
        " query successful"
    Else
      Set rst = New ADODB.Recordset
      rst.Open Trim$(SQL), cnn, _
        adOpenKeyset, _
        adLockOptimistic
      'rst.MoveLast     'get RecordCount
      Set ExecuteSQL = rst
      MsgString = "Found" & rst.RecordCount & _
        " records "
    End If
ExecuteSQL_Exit:
    Set rst = Nothing
    Set cnn = Nothing
    Exit Function

ExecuteSQL_Error:
    MsgString = "Querying error: " & _
      Err.Description
    Resume ExecuteSQL_Exit
```

```
End Function
Public Sub EnterToTab(Keyasc As Integer)
   If Keyasc = 13 Then
      SendKeys "{TAB}"
   End If
End Sub
Public Function GetRkno() As String
   GetRkno = Format(Now, "yymmddhhmmss")
   Randomize
   GetRkno = GetRkno & Int((99 - 10 + 1) * Rnd + 10)
End Function
```

<Loging>

```
Private Declare Function GetUserName Lib "advapi32.dll" Alias "GetUserNameA"
(ByVal lpbuffer As String, nSize As Long) As Long

Public OK As Boolean
'Record the certain times of input
Dim miCount As Integer
Private Sub Form_Load()
   Dim sBuffer As String
   Dim lSize As Long

   sBuffer = Space$(255)
   lSize = Len(sBuffer)
   Call GetUserName(sBuffer, lSize)
   If lSize > 0 Then
      txtUserName.Text = ""
   Else
      txtUserName.Text = vbNullString
   End If

   OK = False
   miCount = 0
End Sub

Private Sub cmdCancel_Click()
   OK = False
   Me.Hide
End Sub
```

```
Private Sub cmdOK_Click()
    Dim txtSQL As String
    Dim mrc As ADODB.Recordset
    Dim MsgText As String

    'ToDo: create test for correct password
    'check for correct password

    UserName = ""
    UserType = ""
    If Trim(txtUserName.Text = "") Then
        MsgBox "This user is not esixted，Please input username again！ ", vbOKOnly +
vbExclamation, "Warning"
        txtUserName.SetFocus
    Else
        txtSQL = "select * from user_Info where user_ID = '" & txtUserName.Text & "'"
        Set mrc = ExecuteSQL(txtSQL, MsgText)
        If mrc.EOF = True Then
            MsgBox "This user is not esixted，Please input username again！ ", vbOKOnly
+ vbExclamation, "Warning"
            txtUserName.SetFocus
        Else
            If Trim(mrc.Fields(1)) = Trim(txtPassword.Text) Then
                OK = True
                UserType = Trim(mrc.Fields(2))
                mrc.Close
                Me.Hide
                UserName = Trim(txtUserName.Text)

            Else
                MsgBox "The password is not correct，Please input again！ ", vbOKOnly +
vbExclamation, "Warning"
                txtPassword.SetFocus
                txtPassword.Text = ""
            End If
        End If
    End If

    miCount = miCount + 1
    If miCount = 3 Then
        Me.Hide
    End If
    Exit Sub
End Sub
```

<frmMain>

```
Private Sub MDIForm_Load()
```

```
Dim MsgText As String
Dim UType(6) As String
Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)
Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)
UType(0) = "System"
UType(1) = "Airliner"
UType(2) = "Flight"
UType(3) = "Ticket"
UType(4) = "Manager"
UType(5) = "Sale"
UType(6) = "SuperAdministrator"
      If UserType = UType(0) Then
         menuModifypwd.Enabled = True
         menuAdduser.Enabled = True
         menuExit.Enabled = True
         menuCarbin.Enabled = False
         menuPlane.Enabled = False
         menuAirline.Enabled = False
         menuCtype.Enabled = False
         menuCustomer.Enabled = False
         menuBookticket.Enabled = False
         menuPrTicket.Enabled = False
         menuPrCustomer.Enabled = False
         menuPrAirliner.Enabled = False
         menuAbout.Enabled = True
      ElseIf UserType = UType(1) Then
         menuModifypwd.Enabled = True
         menuAdduser.Enabled = False
         menuExit.Enabled = True
         menuCarbin.Enabled = False
         menuPlane.Enabled = True
         menuAirline.Enabled = False
         menuCtype.Enabled = False
         menuCustomer.Enabled = False
         menuBookticket.Enabled = False
         menuPrTicket.Enabled = False
         menuPrCustomer.Enabled = False
         menuPrAirliner.Enabled = True
         menuAbout.Enabled = True
      ElseIf UserType = UType(2) Then
         menuModifypwd.Enabled = True
         menuAdduser.Enabled = False
         menuExit.Enabled = True
         menuCarbin.Enabled = True
         menuPlane.Enabled = False
         menuAirline.Enabled = True
         menuCtype.Enabled = False
         menuCustomer.Enabled = False
```

```
    menuBookticket.Enabled = False
    menuPrTicket.Enabled = False
    menuPrCustomer.Enabled = False
    menuPrAirliner.Enabled = False
    menuAbout.Enabled = True
ElseIf UserType = UType(3) Then
    menuModifypwd.Enabled = True
    menuAdduser.Enabled = False
    menuExit.Enabled = True
    menuCarbin.Enabled = False
    menuPlane.Enabled = False
    menuAirline.Enabled = False
    menuCtype.Enabled = False
    menuCustomer.Enabled = False
    menuBookticket.Enabled = True
    menuPrTicket.Enabled = True
    menuPrCustomer.Enabled = False
    menuPrAirliner.Enabled = False
    menuAbout.Enabled = True
ElseIf UserType = UType(4) Then
    menuModifypwd.Enabled = True
    menuAdduser.Enabled = False
    menuExit.Enabled = True
    menuCarbin.Enabled = False
    menuPlane.Enabled = False
    menuAirline.Enabled = False
    menuCtype.Enabled = True
    menuCustomer.Enabled = False
    menuBookticket.Enabled = False
    menuPrTicket.Enabled = True
    menuPrCustomer.Enabled = True
    menuPrAirliner.Enabled = True
    menuAbout.Enabled = True
ElseIf UserType = UType(5) Then
    menuModifypwd.Enabled = True
    menuAdduser.Enabled = False
    menuExit.Enabled = True
    menuCarbin.Enabled = False
    menuPlane.Enabled = False
    menuAirline.Enabled = False
    menuCtype.Enabled = False
    menuCustomer.Enabled = True
    menuBookticket.Enabled = False
    menuPrTicket.Enabled = False
    menuPrCustomer.Enabled = True
    menuPrAirliner.Enabled = False
    menuAbout.Enabled = True
ElseIf UserType = UType(6) Then
    menuModifypwd.Enabled = True
    menuAdduser.Enabled = True
```

```
            menuExit.Enabled = True
            menuCarbin.Enabled = True
            menuPlane.Enabled = True
            menuAirline.Enabled = True
            menuCtype.Enabled = True
            menuCustomer.Enabled = True
            menuBookticket.Enabled = True
            menuPrTicket.Enabled = True
            menuPrCustomer.Enabled = True
            menuPrAirliner.Enabled = True
            menuAbout.Enabled = True
        End If

End Sub

Private Sub MDIForm_Unload(Cancel As Integer)
    If Me.WindowState <> vbMinimized Then
        SaveSetting App.Title, "Settings", "MainLeft", Me.Left
        SaveSetting App.Title, "Settings", "MainTop", Me.Top
        SaveSetting App.Title, "Settings", "MainWidth", Me.Width
        SaveSetting App.Title, "Settings", "MainHeight", Me.Height
    End If
End Sub

Private Sub menuAbout_Click()
    frmAbout.Show 0
End Sub

Private Sub menuAdduser_Click()
    frmAdduser.Show 0
End Sub

Private Sub menuAirline_Click()
    frmAirline.txtSQL = "select * from airlineInfo"
    frmAirline.Show 0
End Sub

Private Sub menuBookticket_Click()
    frmTicket.txtSQL = "select * from ticketInfo"
    frmTicket.Show 0
End Sub

Private Sub menuCarbin_Click()
    frmService.txtSQL = "select * from serviceInfo"
    frmService.Show 0
End Sub

Private Sub menuCtype_Click()
    frmcType.txtSQL = "select * from customerType"
    frmcType.Show 0
```

```
End Sub

Private Sub menuCustomer_Click()
   frmCustomer.txtSQL = "select * from customerInfo"
   frmCustomer.Show 0
End Sub

Private Sub menuExit_Click()
   Unload Me
End Sub

Private Sub menuModifypwd_Click()
   frmModifyuserinfo.Show 0
End Sub

Private Sub menuPlane_Click()
   frmPlane.txtSQL = "select * from planeInfo"
   frmPlane.Show 0
End Sub

Private Sub menuPrAirliner_Click()
   rpAirliner.Show
   On Error Resume Next
   With CommonDialog1
      .Flags = cd1CCFullOpen
      .ShowPrinter
   End With
End Sub

Private Sub menuPrCustomer_Click()
   rpCustomer.Show 0
   On Error Resume Next
   With CommonDialog1
      .Flags = cd1CCFullOpen
      .ShowPrinter
   End With
End Sub

Private Sub menuPrTicket_Click()
   rpTicket.Show
   On Error Resume Next
   With CommonDialog1
      .Flags = cd1CCFullOpen
      .ShowPrinter
   End With
End Sub
```

<Add User>

```
Private Sub cmdCancel_Click()
    Unload Me
End Sub
Private Sub Form_Load()

    'Set value
    Combo1(0).AddItem "System"
    Combo1(0).AddItem "Airliner"
    Combo1(0).AddItem "Flight"
    Combo1(0).AddItem "Ticket"
    Combo1(0).AddItem "Sale"
    Combo1(0).AddItem "Manager"
    Combo1(0).AddItem "SuperAdministrator"
    Combo1(0).ListIndex = 0

End Sub

Private Sub cmdOK_Click()
    Dim txtSQL As String
    Dim mrc As ADODB.Recordset
    Dim MsgText As String

    'Judge whether is null
    If Trim(Text1(0).Text) = "" Then
        MsgBox "Please input username! ", vbOKOnly + vbExclamation, "Warning"
        Exit Sub
        Text1(0).SetFocus
    Else
        txtSQL = "select * from user_Info"
        Set mrc = ExecuteSQL(txtSQL, MsgText)
        'Judge whether recordset is null
        While (mrc.EOF = False)
            'Judge whether exist repeatable record
            If Trim(mrc.Fields(0)) = Trim(Text1(0)) Then
                MsgBox "The user already existed, Please input a new one!", vbOKOnly +
vbExclamation, "Warning"
                Text1(0).SetFocus
                Text1(0).Text = ""
                Text1(1).Text = ""
                Text1(2).Text = ""
                Exit Sub
            Else
                'Move to next record
                mrc.MoveNext
            End If
        Wend
    End If
    'Judge whether the password inputed two times is same
    If Trim(Text1(1).Text) <> Trim(Text1(2).Text) Then
```

```
        MsgBox "The password inputed two times is not same, Please check it!",
vbOKOnly + vbExclamation, "Warning"
        Text1(1).SetFocus
        Text1(1).Text = ""
        Text1(2).Text = ""
        Exit Sub
    Else
    'Judge whether the password is null
        If Text1(1).Text = "" Then
            MsgBox "The password cannot be null!", vbOKOnly + vbExclamation,
"Warning"
            Text1(1).SetFocus
            Text1(1).Text = ""
            Text1(2).Text = ""
        Else
            'Add a new user
            mrc.AddNew
            mrc.Fields(0) = Trim(Text1(0).Text)
            mrc.Fields(1) = Trim(Text1(1).Text)
            mrc.Fields(2) = Trim(Combo1(0).Text)
            'Update database
            mrc.Update
            'Close recordsets
            mrc.Close
            Me.Hide
            MsgBox "Added a new user successfully!", vbOKOnly + vbExclamation,
"Adding User"
        End If
    End If
End Sub
```

<Change Password>

```
Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    Dim txtSQL, MsgText As String
    Dim mrc As ADODB.Recordset
    'Judge whether is null
    If Trim(Text1(0).Text) <> Trim(Text1(1).Text) Then
        MsgBox "The password is not correct ! ", vbOKOnly + vbExclamation,
"WARNING"
        Text1(0).SetFocus
        Text1(0).Text = ""
    Else
        txtSQL = "select * from user_Info where user_ID='" & UserName & "'"
```

```
        Set mrc = ExecuteSQL(txtSQL, MsgText)
        mrc.Fields(1) = Trim(Text1(0).Text)
        mrc.Update
        mrc.Close
        MsgBox "The password changed successfully! ", vbOKOnly + vbExclamation,
"WARNING"
        Me.Hide
    End If
End Sub
```

< Manage Airliner Information>

```
Option Explicit
Public txtSQL As String
Dim mrc As ADODB.Recordset

Private Sub cmdAdd_Click()
    gintPmode = 1
    frmPlane1.Show 1
End Sub

Private Sub cmdDelete_Click()
    Dim txtSQL As String
    Dim intCount As Integer
    Dim mrc As ADODB.Recordset
    Dim MsgText As String

    If msgList.Rows > 1 Then
        If MsgBox(" Are you sure you want to delete a record that airliner Number is " &
Trim(msgList.TextMatrix(msgList.Row, 1)) & " and model is " &
Trim(msgList.TextMatrix(msgList.Row, 2)) & "? ", vbOKCancel + vbExclamation,
"Warning") = vbOK Then
            intCount = msgList.Row
            txtSQL = "delete from planeInfo where planeNO ='" &
Trim(msgList.TextMatrix(intCount, 1)) & "'"
            Set mrc = ExecuteSQL(txtSQL, MsgText)

            Unload frmPlane
            frmPlane.txtSQL = "select * from planeInfo"
            frmPlane.Show
        End If
    End If

End Sub

Private Sub cmdExit_Click()
    Unload Me
End Sub
```

```
Private Sub cmdModify_Click()
   Dim intCount As Integer

   If frmPlane.msgList.Rows > 1 Then
      gintPmode = 2
      intCount = msgList.Row
      If intCount > 0 Then
        frmPlane1.txtSQL = "select * from planeInfo where planeNO ='" &
Trim(msgList.TextMatrix(intCount, 1)) & "'"
        frmPlane1.Show 1
      Else
        MsgBox "Warning", vbOKOnly + vbExclamation, " Please select the record you
want to change ! "
      End If
   End If
End Sub



Private Sub Command3_Click()
   frmPlane2.Show 1
End Sub

Private Sub Form_Load()
   ShowTitle
   ShowData
End Sub

Private Sub Form_Resize()
   If Me.WindowState <> vbMinimized And fMainForm.WindowState <>
vbMinimized Then
      'Boundary processing
      If Me.ScaleHeight < 10 * lblTitle.Height Then

         Exit Sub
      End If
      If Me.ScaleWidth < lblTitle.Width + lblTitle.Width / 2 Then

         Exit Sub
      End If
      'The position of widget controlling

      lblTitle.Top = lblTitle.Height
      lblTitle.Left = (Me.Width - lblTitle.Width) / 2

      msgList.Top = lblTitle.Top + lblTitle.Height + lblTitle.Height / 2
      msgList.Width = Me.ScaleWidth - 200
      msgList.Left = Me.ScaleLeft + 100
      msgList.Height = Me.ScaleHeight - msgList.Top - 1500
```

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

```
        Frame2.Top = msgList.Top + msgList.Height + 50
        Frame2.Left = Me.ScaleWidth / 2 - 3000
    End If
End Sub

Public Sub FormClose()
    Unload Me
End Sub
Private Sub ShowData()

    Dim j As Integer
    Dim i As Integer
    Dim MsgText As String

    Set mrc = ExecuteSQL(txtSQL, MsgText)
        With msgList
        .Rows = 1

        Do While Not mrc.EOF
            .Rows = .Rows + 1
            For i = 1 To mrc.Fields.Count
                If Not IsNull(Trim(mrc.Fields(i - 1))) Then
                Select Case mrc.Fields(i - 1).Type
                    Case adDBDate
                    .TextMatrix(.Rows - 1, i) = Format(mrc.Fields(i - 1) & "", "yyyy-mm-dd")
                    Case Else
                        .TextMatrix(.Rows - 1, i) = mrc.Fields(i - 1) & ""
                End Select
                End If
            Next i
            mrc.MoveNext
        Loop

    End With
    mrc.Close

End Sub

'Display list head of Grid
Private Sub ShowTitle()
    Dim i As Integer

    With msgList
        .Cols = 9
        .TextMatrix(0, 1) = "Airliner Number"
        .TextMatrix(0, 2) = "Airliner Model"
        .TextMatrix(0, 3) = "Purchase Time"
        .TextMatrix(0, 4) = "Service Time"
        .TextMatrix(0, 5) = "Seat Amount of Economy Class"
        .TextMatrix(0, 6) = "Seat Amount of Business Class"
```

```
        .TextMatrix(0, 7) = "Seat Amount of First Class"
        .TextMatrix(0, 8) = "Remark"

        'Fix list head
        .FixedRows = 1

        'Set alignment of column
        For i = 0 To 8
            .ColAlignment(i) = 0
        Next i

        'Align list head center
        .FillStyle = flexFillRepeat
        .Col = 0
        .Row = 0
        .RowSel = 1
        .ColSel = .Cols - 1
        .CellAlignment = 4

        'Set size of cell
        .ColWidth(0) = 700
        .ColWidth(1) = 2000
        .ColWidth(2) = 2000
        .ColWidth(3) = 2000
        .ColWidth(4) = 2000
        .ColWidth(5) = 2500
        .ColWidth(6) = 2500
        .ColWidth(7) = 2500
        .ColWidth(8) = 4500
        .Row = 1

    End With
End Sub

Private Sub msgList_MouseUp(Button As Integer, Shift As Integer, x As Single, y As
Single)
    'Popup right key
    If Button = 2 And Shift = 0 Then

    End If

End Sub
```

<Input Airliner Information>

```
Option Explicit
Public txtSQL As String
Dim mrc As ADODB.Recordset
```

```
Private Sub cmdAdd_Click()
   gintPmode = 1
   frmPlane1.Show 1

End Sub

Private Sub cmdDelete_Click()
   Dim txtSQL As String
   Dim intCount As Integer
   Dim mrc As ADODB.Recordset
   Dim MsgText As String

   If msgList.Rows > 1 Then
      If MsgBox(" Are you sure you want to delete a record that airliner Number is " &
Trim(msgList.TextMatrix(msgList.Row, 1)) & " and model is " &
Trim(msgList.TextMatrix(msgList.Row, 2)) & "? ", vbOKCancel + vbExclamation,
"Warning") = vbOK Then
         intCount = msgList.Row
         txtSQL = "delete from planeInfo where planeNO ='" &
Trim(msgList.TextMatrix(intCount, 1)) & "'"
         Set mrc = ExecuteSQL(txtSQL, MsgText)

         Unload frmPlane
         frmPlane.txtSQL = "select * from planeInfo"
         frmPlane.Show
      End If
   End If
End Sub

Private Sub cmdExit_Click()
   Unload Me
End Sub

Private Sub cmdModify_Click()
   Dim intCount As Integer

   If frmPlane.msgList.Rows > 1 Then
      gintPmode = 2
      intCount = msgList.Row
      If intCount > 0 Then
        frmPlane1.txtSQL = "select * from planeInfo where planeNO ='" &
Trim(msgList.TextMatrix(intCount, 1)) & "'"
        frmPlane1.Show 1
      Else
        MsgBox "Warning", vbOKOnly + vbExclamation, " Please select the record you
want to change! "
      End If
   End If
End Sub
```

```
Private Sub Command3_Click()
   frmPlane2.Show 1
End Sub

Private Sub Form_Load()
   ShowTitle
   ShowData

End Sub

Private Sub Form_Resize()
   If Me.WindowState <> vbMinimized And fMainForm.WindowState <>
vbMinimized Then
      'Boundary processing
      If Me.ScaleHeight < 10 * lblTitle.Height Then

         Exit Sub
      End If
      If Me.ScaleWidth < lblTitle.Width + lblTitle.Width / 2 Then

         Exit Sub
      End If
      'The position of widget controlling

      lblTitle.Top = lblTitle.Height
      lblTitle.Left = (Me.Width - lblTitle.Width) / 2

      msgList.Top = lblTitle.Top + lblTitle.Height + lblTitle.Height / 2
      msgList.Width = Me.ScaleWidth - 200
      msgList.Left = Me.ScaleLeft + 100
      msgList.Height = Me.ScaleHeight - msgList.Top - 1500

      Frame2.Top = msgList.Top + msgList.Height + 50
      Frame2.Left = Me.ScaleWidth / 2 - 3000
   End If
End Sub

Public Sub FormClose()
   Unload Me
End Sub
Private Sub ShowData()

   Dim j As Integer
   Dim i As Integer
   Dim MsgText As String

   Set mrc = ExecuteSQL(txtSQL, MsgText)
      With msgList
      .Rows = 1
```

St. Gabriel's Library, Au

```
        Do While Not mrc.EOF
            .Rows = .Rows + 1
            For i = 1 To mrc.Fields.Count
                If Not IsNull(Trim(mrc.Fields(i - 1))) Then
                Select Case mrc.Fields(i - 1).Type
                    Case adDBDate
                    .TextMatrix(.Rows - 1, i) = Format(mrc.Fields(i - 1) & "", "yyyy-mm-dd")
                    Case Else
                        .TextMatrix(.Rows - 1, i) = mrc.Fields(i - 1) & ""
                End Select
                End If
            Next i
            mrc.MoveNext
        Loop


    End With
    mrc.Close
End Sub

'Display list head of Grid
Private Sub ShowTitle()
    Dim i As Integer

    With msgList
        .Cols = 9
        .TextMatrix(0, 1) = "Airliner Number"
        .TextMatrix(0, 2) = "Airliner Model"
        .TextMatrix(0, 3) = "Purchase Time"
        .TextMatrix(0, 4) = "Service Time"
        .TextMatrix(0, 5) = "Seat Amount of Economy Class"
        .TextMatrix(0, 6) = "Seat Amount of Business Class"
        .TextMatrix(0, 7) = "Seat Amount of First Class"
        .TextMatrix(0, 8) = "Remark"

        'Fix list head
        .FixedRows = 1

        'Set alignment of column
        For i = 0 To 8
            .ColAlignment(i) = 0
        Next i

        'Align list head center
        .FillStyle = flexFillRepeat
        .Col = 0
        .Row = 0
        .RowSel = 1
        .ColSel = .Cols - 1
        .CellAlignment = 4
```

```
        'Set size of cell
        .ColWidth(0) = 700
        .ColWidth(1) = 2000
        .ColWidth(2) = 2000
        .ColWidth(3) = 2000
        .ColWidth(4) = 2000
        .ColWidth(5) = 2500
        .ColWidth(6) = 2500
        .ColWidth(7) = 2500
        .ColWidth(8) = 4500
        .Row = 1

    End With
End Sub

Private Sub msgList_MouseUp(Button As Integer, Shift As Integer, x As Single, y As
Single)
    'Popup right key
    If Button = 2 And Shift = 0 Then

    End If

End Sub
```

<Inquiry Airliner Information>

```
Option Explicit
Private Sub chkItem_Click(Index As Integer)
    If Index = 0 Then
        Combo1(0).SetFocus
    Else
        Combo1(1).SetFocus
    End If
End Sub

Private Sub cmdExit_Click()
    Me.Hide
End Sub

Private Sub cmdOK_Click()
    Dim sQSql As String

    If chkItem(0).Value = vbChecked Then
        sQSql = " planeNO = '" & Trim(Combo1(0) & " ") & "'"
    End If

    If chkItem(1).Value = vbChecked Then
        If Trim(sQSql & " ") = "" Then
            sQSql = " planeType ='" & Trim(Combo1(1) & " ") & "'"
```

```
        Else
            sQSql = sQSql & " and planeType ='" & Trim(Combo1(1) & " ") & "'"
        End If
    End If

    If Trim(sQSql) = "" Then
        MsgBox "Please set condition of query ! ", vbOKOnly + vbExclamation,
"Warning"
        Me.Hide
        Exit Sub
    Else
        frmPlane.txtSQL = "select * from planeInfo where" & sQSql
        Me.Hide
        Unload frmPlane
        frmPlane.Show
    End If

End Sub
Private Sub Form_Load()

    Dim i As Integer
    Dim sSql As String
    Dim txtSQL As String
    Dim MsgText As String
    Dim mrc As ADODB.Recordset

    For i = 0 To 1
        Combo1(i).Clear
    Next i

    txtSQL = "select DISTINCT planeNO from planeInfo"
    Set mrc = ExecuteSQL(txtSQL, MsgText)

    If Not mrc.EOF Then

        Do While Not mrc.EOF
            Combo1(0).AddItem Trim(mrc.Fields(0))
            mrc.MoveNext
        Loop
    Else
        MsgBox " Please set airliner information first! ", vbOKOnly + vbExclamation,
"Warning"
        Exit Sub
    End If
    mrc.Close

    txtSQL = "select DISTINCT planeType from planeInfo"
    Set mrc = ExecuteSQL(txtSQL, MsgText)

    If Not mrc.EOF Then
```

```
        Do While Not mrc.EOF
           Combo1(1).AddItem Trim(mrc.Fields(0))
           mrc.MoveNext
        Loop
    Else
        MsgBox " Please set airliner information first! ", vbOKOnly + vbExclamation,
"Warning"
        Exit Sub
    End If
    mrc.Close

End Sub

Private Sub lblitem_Click(Index As Integer)
    chkItem(Index).Value = vbChecked

End Sub

Private Sub txtItem_GotFocus(Index As Integer)
    txtItem(Index).SelStart = 0
    txtItem(Index).SelLength = Len(txtItem(Index))
End Sub
```

<List Scheduled Flight Information>

```
Option Explicit
Public txtSQL As String
Dim mrc As ADODB.Recordset

Private Sub cmdAdd_Click()
    gintAmode = 1
    frmAirline1.Show 1

End Sub

Private Sub cmdDelete_Click()
    Dim txtSQL As String
    Dim intCount As Integer
    Dim mrc As ADODB.Recordset
    Dim MsgText As String

    If msgList.Rows > 1 Then
        If MsgBox("Are you sure you want to delete " &
Trim(msgList.TextMatrix(msgList.Row, 4)) & " record of flight course from " &
Trim(msgList.TextMatrix(msgList.Row, 2)) & " to " &
Trim(msgList.TextMatrix(msgList.Row, 3)) & "? ", vbOKCancel + vbExclamation,
"Warning") = vbOK Then
            intCount = msgList.Row
```

```
        txtSQL = "delete from airlineInfo where airlineNO ='" &
Trim(msgList.TextMatrix(intCount, 1)) & "'"
        Set mrc = ExecuteSQL(txtSQL, MsgText)

        Unload frmAirline
        frmAirline.txtSQL = "select * from airlineInfo"
        frmAirline.Show
      End If
   End If

End Sub

Private Sub cmdExit_Click()
   Unload Me
End Sub

Private Sub cmdModify_Click()
   Dim intCount As Integer

   If frmAirline.msgList.Rows > 1 Then
      gintAmode = 2
      intCount = msgList.Row
      If intCount > 0 Then
        frmAirline1.txtSQL = "select * from airlineInfo where airlineNO ='" &
Trim(msgList.TextMatrix(intCount, 1)) & "'"
        frmAirline1.Show 1
      Else
        MsgBox "Warning", vbOKOnly + vbExclamation, " Please select record you
want to modify first! "
      End If
   End If

End Sub

Private Sub Command3_Click()
   frmAirline2.Show 1
End Sub

Private Sub Form_Load()
   ShowTitle
   ShowData

End Sub

Private Sub Form_Resize()
   If Me.WindowState <> vbMinimized And fMainForm.WindowState <>
vbMinimized Then
      'Boundary Processing
      If Me.ScaleHeight < 10 * lblTitle.Height Then
```

ASSUMPTION UNIVERSITY OF THAILAND
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

```
        Exit Sub
      End If
      If Me.ScaleWidth < lblTitle.Width + lblTitle.Width / 2 Then

        Exit Sub
      End If
      'The Position of Widget Contoling

      lblTitle.Top = lblTitle.Height
      lblTitle.Left = (Me.Width - lblTitle.Width) / 2

      msgList.Top = lblTitle.Top + lblTitle.Height + lblTitle.Height / 2
      msgList.Width = Me.ScaleWidth - 200
      msgList.Left = Me.ScaleLeft + 100
      msgList.Height = Me.ScaleHeight - msgList.Top - 1500

      Frame2.Top = msgList.Top + msgList.Height + 50
      Frame2.Left = Me.ScaleWidth / 2 - 3000
   End If
End Sub

Public Sub FormClose()
   Unload Me
End Sub
Private Sub ShowData()

   Dim j As Integer
   Dim i As Integer
   Dim MsgText As String


   Set mrc = ExecuteSQL(txtSQL, MsgText)
      With msgList
      .Rows = 1

      Do While Not mrc.EOF
         .Rows = .Rows + 1
         For i = 1 To mrc.Fields.Count
            If Not IsNull(Trim(mrc.Fields(i - 1))) Then
            Select Case mrc.Fields(i - 1).Type
               Case adDBDate
               .TextMatrix(.Rows - 1, i) = Format(mrc.Fields(i - 1) & "", "yyyy-mm-dd")
               Case Else
                  .TextMatrix(.Rows - 1, i) = mrc.Fields(i - 1) & ""
            End Select
            End If
         Next i
         mrc.MoveNext
      Loop
```

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE 1969
มหาวิทยาลัยอัสสัมชัญ

```
    End With
    mrc.Close

End Sub


'Display List Head of Grid
Private Sub ShowTitle()
    Dim i As Integer

    With msgList
        .Cols = 12
        .TextMatrix(0, 1) = ""
        .TextMatrix(0, 2) = "Departure City"
        .TextMatrix(0, 3) = "Arrival City"
        .TextMatrix(0, 4) = "Dtae of Flight"
        .TextMatrix(0, 5) = "Departure Time"
        .TextMatrix(0, 6) = "Arrival Time"
        .TextMatrix(0, 7) = "No. of Airliner"
        .TextMatrix(0, 8) = "Price of Economy Class"
        .TextMatrix(0, 9) = "Price of Business Class"
        .TextMatrix(0, 10) = "Price of First Class"
        .TextMatrix(0, 11) = "Remark"

        'Fixed List Head
        .FixedRows = 1

        'Set Alignment of Column
        For i = 0 To 11
            .ColAlignment(i) = 0
        Next i


        'Align List Head Center
        .FillStyle = flexFillRepeat
        .Col = 0
        .Row = 0
        .RowSel = 1
        .ColSel = .Cols - 1
        .CellAlignment = 4

        'Set Size of Cell
        .ColWidth(0) = 700
        .ColWidth(1) = 0
        .ColWidth(2) = 2000
        .ColWidth(3) = 2000
        .ColWidth(4) = 2000
        .ColWidth(5) = 2500
```

ASSUMPTION UNIVERSITY OF THAILAND
มหาวิทยาลัยอัสสัมชัญ
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969

```
        .ColWidth(6) = 2500
        .ColWidth(7) = 2500
        .ColWidth(8) = 2500
        .ColWidth(9) = 2500
        .ColWidth(10) = 2500
        .ColWidth(11) = 4500
        .Row = 1

    End With
End Sub

Private Sub msgList_MouseUp(Button As Integer, Shift As Integer, x As Single, y As
Single)
    'Popup Right Key
    If Button = 2 And Shift = 0 Then

    End If

End Sub
```

<Input Scheduled Flight Information>

```
Option Explicit
'Whether Changed Record，ture is Changed
Dim mblChange As Boolean
Dim mrc As ADODB.Recordset
Public txtSQL As String

Private Sub cmdExit_Click()
    If mblChange And cmdSave.Enabled Then
        If MsgBox(" Do you want to save changes? ", vbOKCancel + vbExclamation,
"Warning") = vbOK Then
            'Save
            Call cmdSave_Click
        End If
    End If
    Unload Me
End Sub

Private Sub cmdSave_Click()
    Dim intCount As Integer
    Dim sMeg As String
    Dim MsgText As String


    For intCount = 0 To 6
        If Trim(txtItem(intCount) & " ") = "" Then
            Select Case intCount
                Case 0
```

```
            sMeg = "Departure City"
         Case 1
            sMeg = "Arrival City"
         Case 2
            sMeg = "Departure Time"
         Case 3
            sMeg = "Arrival Time"
         Case 4
            sMeg = "Price of Economy Class"
         Case 5
            sMeg = "Price of Business Class"
         Case 6
            sMeg = "Price of First Class"
      End Select
      sMeg = sMeg & " cannot be null!  "
      MsgBox sMeg, vbOKOnly + vbExclamation, "Warning"
      txtItem(intCount).SetFocus
      Exit Sub
   End If
Next intCount

For intCount = 0 To 1
   If Trim(Combo1(intCount) & " ") = "" Then
      Select Case intCount
         Case 0
            sMeg = "Date of Flight"
         Case 1
            sMeg = "No.of Airliner"
      End Select
      sMeg = sMeg & " cannot be null!  "
      MsgBox sMeg, vbOKOnly + vbExclamation, "Warning"
      Combo1(intCount).SetFocus
      Exit Sub
   End If
Next intCount

For intCount = 4 To 6
   If Not IsNumeric(Trim(txtItem(intCount))) Then
      sMeg = "Price of Ticket,"
      sMeg = sMeg & " Please input number!  "
      MsgBox sMeg, vbOKOnly + vbExclamation, "Warning"
      txtItem(intCount).SetFocus
   End If
Next intCount


For intCount = 2 To 3
   If IsDate(txtItem(intCount)) Then
      txtItem(intCount) = Format(txtItem(intCount), "hh:mm")
   Else
```

```
        MsgBox " The format of time is:（hh:mm） ! ", vbOKOnly + vbExclamation,
"Warning"
        txtItem(intCount).SetFocus
        Exit Sub
      End If
    Next intCount

    'Judge whether there are same records
    txtSQL = "select * from airlineInfo where airlineNO <> '" & Trim(txtNo) & "' and
departCity = '" & Trim(txtItem(0)) & "' and arrivalCity = '" & Trim(txtItem(1)) & "' and
departDate = '" & Trim(Combo1(0)) & " ' and departTime = '" & Trim(txtItem(2)) & "'
and  planeNO = '" & Trim(Combo1(1)) & """
    Set mrc = ExecuteSQL(txtSQL, MsgText)
    If mrc.EOF = False Then
      MsgBox "Exists a record of same flight ! ", vbOKOnly + vbExclamation,
"Warning"
      txtItem(0).SetFocus
      Exit Sub
    End If

    'Deleted the foregone records first
    txtSQL = "delete from airlineInfo where airlineNO='" & Trim(txtNo) & """
    Set mrc = ExecuteSQL(txtSQL, MsgText)


    'Then add new record
    txtSQL = "select * from airlineInfo"
    Set mrc = ExecuteSQL(txtSQL, MsgText)
    mrc.AddNew

    mrc.Fields(0) = txtNo

    For intCount = 0 To 1
      mrc.Fields(intCount + 1) = Trim(txtItem(intCount))
    Next intCount

    mrc.Fields(3) = Trim(Combo1(0))

    For intCount = 2 To 3
      mrc.Fields(intCount + 2) = Trim(txtItem(intCount))
    Next intCount

    mrc.Fields(6) = Trim(Combo1(1))

    For intCount = 4 To 7
      mrc.Fields(intCount + 3) = Trim(txtItem(intCount))
    Next intCount

    mrc.Update
    mrc.Close
```

```
    If gintAmode = 1 Then
        MsgBox " Added Record Successfully!  ", vbOKOnly + vbExclamation, "Adding
Record"
        For intCount = 0 To 7
            txtItem(intCount) = ""
        Next intCount

        mblChange = False
        Unload frmAirline
        frmAirline.txtSQL = "select * from airlineInfo"
        frmAirline.Show

    ElseIf gintAmode = 2 Then
        Unload Me
        Unload frmAirline

        frmAirline.txtSQL = "select * from airlineInfo"
        frmAirline.Show
    End If

End Sub

Private Sub Form_Load()
    Dim intCount As Integer
    Dim MsgText As String
    Dim i As Integer
    Dim mrcc As ADODB.Recordset

    If gintAmode = 1 Then
        Me.Caption = Me.Caption & "Add"
        txtNo = GetRkno()

    ElseIf gintAmode = 2 Then
        Set mrc = ExecuteSQL(txtSQL, MsgText)

        If mrc.EOF = False Then
            With mrc

                txtNo = .Fields(0)

                For intCount = 0 To 1
                    txtItem(intCount) = .Fields(intCount + 1)
                Next intCount

                For intCount = 2 To 3
                    txtItem(intCount) = .Fields(intCount + 2)
                Next intCount

                For intCount = 4 To 7
```

```
            txtItem(intCount) = .Fields(intCount + 3)
        Next intCount
      End With

    End If

    Me.Caption = Me.Caption & "Modify"
  End If

  Combo1(0).Clear

    With Combo1(0)
      .AddItem "Monday"
      .AddItem "Tuesday"
      .AddItem "Wednesday"
      .AddItem "Thursday"
      .AddItem "Friday"
      .AddItem "Saturday"
      .AddItem "Sunday"
    End With

    Combo1(1).Clear

    txtSQL = "select DISTINCT planeNO from planeInfo"
    Set mrcc = ExecuteSQL(txtSQL, MsgText)

    If mrcc.EOF = False Then
      Do While Not mrcc.EOF
        Combo1(1).AddItem mrcc.Fields(0)
        mrcc.MoveNext
      Loop
    Else
      MsgBox " Please set information of airliner first ! ", vbOKOnly +
vbExclamation, "Warning"
      Exit Sub
    End If

    mrcc.Close

  mblChange = False
End Sub

Private Sub Form_Unload(Cancel As Integer)
  gintAmode = 0
End Sub

Private Sub txtItem_Change(Index As Integer)
  'There is change then set gblchange
  mblChange = True
End Sub
```

St. Gabriel's Library, Au

```
Private Sub txtItem_GotFocus(Index As Integer)
   txtItem(Index).SelStart = 0
   txtItem(Index).SelLength = Len(txtItem(Index))
End Sub

Private Sub txtItem_KeyDown(Index As Integer, KeyCode As Integer, Shift As Integer)
   EnterToTab KeyCode
End Sub
```

<Inquiry Scheduled Flight Information>

```
Option Explicit
Private Sub cmdExit_Click()
   Me.Hide
End Sub

Private Sub cmdOK_Click()
   Dim sQSql As String

   If (Trim(Combo1(0)) = "") Or (Trim(Combo1(1)) = "") Then
      MsgBox " Please set a condition of query ! ", vbOKOnly + vbExclamation,
"Warning"
      Exit Sub
   Else
      sQSql = "select * from airlineInfo where departCity = '" & Trim(Combo1(0)) & "'
and arrivalCity = '" & Trim(Combo1(1)) & "'"
   End If
   Unload frmAirline
   frmAirline.txtSQL = sQSql
   frmAirline.Show

End Sub
Private Sub Form_Load()

   Dim i As Integer
   Dim sSql As String
   Dim txtSQL As String
   Dim MsgText As String
   Dim mrc As ADODB.Recordset

   For i = 0 To 1
      Combo1(i).Clear
   Next i

   txtSQL = "select DISTINCT departCity from airlineInfo"
   Set mrc = ExecuteSQL(txtSQL, MsgText)
```

```
    If Not mrc.EOF Then

        Do While Not mrc.EOF
            Combo1(0).AddItem Trim(mrc.Fields(0))
            mrc.MoveNext
        Loop
    Else
        MsgBox " Please set an information of flight first!  ", vbOKOnly + vbExclamation,
"Warning"
        Exit Sub
    End If
    mrc.Close

    txtSQL = "select DISTINCT arrivalCity from airlineInfo"
    Set mrc = ExecuteSQL(txtSQL, MsgText)

    If Not mrc.EOF Then

        Do While Not mrc.EOF
            Combo1(1).AddItem Trim(mrc.Fields(0))
            mrc.MoveNext
        Loop
    Else
        MsgBox " Please set an information of flight first!  ", vbOKOnly + vbExclamation,
"Warning"
        Exit Sub
    End If
    mrc.Close

End Sub

Private Sub txtItem_GotFocus(Index As Integer)
    txtItem(Index).SelStart = 0
    txtItem(Index).SelLength = Len(txtItem(Index))
End Sub
```

<List Passenger Type>

```
Option Explicit
Public txtSQL As String
Dim mrc As ADODB.Recordset

Private Sub cmdAdd_Click()
    gintTmode = 1
    frmcType1.Show 1

End Sub

Private Sub cmdDelete_Click()
    Dim txtSQL As String
```

```
    Dim intCount As Integer
    Dim mrc As ADODB.Recordset
    Dim MsgText As String


    If msgList.Rows > 1 Then
       If MsgBox(" Are you sure you want to delete a record of " &
Trim(msgList.TextMatrix(msgList.Row, 2)) & " of customer type? ", vbOKCancel +
vbExclamation, "Warning") = vbOK Then
          intCount = msgList.Row
          txtSQL = "delete from customerType where ctypeNO ='" &
Trim(msgList.TextMatrix(intCount, 1)) & "'"
          Set mrc = ExecuteSQL(txtSQL, MsgText)

          Unload frmcType
          frmcType.txtSQL = "select * from customerType"
          frmcType.Show
       End If
    End If

End Sub

Private Sub cmdExit_Click()
    Unload Me
End Sub

Private Sub cmdModify_Click()
    Dim intCount As Integer

    If frmcType.msgList.Rows > 1 Then
       gintTmode = 2
       intCount = msgList.Row
       If intCount > 0 Then
          frmcType1.txtSQL = "select * from customerType where ctypeNO ='" &
Trim(msgList.TextMatrix(intCount, 1)) & "'"
          frmcType1.Show 1
       Else
          MsgBox "Warning", vbOKOnly + vbExclamation, " Please select record you
want to modify ! "
       End If
    End If

End Sub

Private Sub Form_Load()
    ShowTitle
    ShowData

End Sub
```

```
Private Sub Form_Resize()
  If Me.WindowState <> vbMinimized And fMainForm.WindowState <>
vbMinimized Then
      'Boundary Process
      If Me.ScaleHeight < 10 * lblTitle.Height Then

         Exit Sub
      End If
      If Me.ScaleWidth < lblTitle.Width + lblTitle.Width / 2 Then

         Exit Sub
      End If
      'The Position of Widget Controlling

      lblTitle.Top = lblTitle.Height
      lblTitle.Left = (Me.Width - lblTitle.Width) / 2

      msgList.Top = lblTitle.Top + lblTitle.Height + lblTitle.Height / 2
      msgList.Width = Me.ScaleWidth - 200
      msgList.Left = Me.ScaleLeft + 100
      msgList.Height = Me.ScaleHeight - msgList.Top - 1500

      Frame2.Top = msgList.Top + msgList.Height + 50
      Frame2.Left = Me.ScaleWidth / 2 - 3000
  End If
End Sub

Public Sub FormClose()
  Unload Me
End Sub
Private Sub ShowData()

  Dim j As Integer
  Dim i As Integer
  Dim MsgText As String


  Set mrc = ExecuteSQL(txtSQL, MsgText)
    With msgList
    .Rows = 1

    Do While Not mrc.EOF
      .Rows = .Rows + 1
      For i = 1 To mrc.Fields.Count
        If Not IsNull(Trim(mrc.Fields(i - 1))) Then
        Select Case mrc.Fields(i - 1).Type
          Case adDBDate
          .TextMatrix(.Rows - 1, i) = Format(mrc.Fields(i - 1) & "", "yyyy-mm-dd")
          Case Else
            .TextMatrix(.Rows - 1, i) = mrc.Fields(i - 1) & ""
```

```
            End Select
            End If
        Next i
        mrc.MoveNext
    Loop


  End With
  mrc.Close
End Sub

'Display List Head of Grid
Private Sub ShowTitle()
  Dim i As Integer

  With msgList
    .Cols = 5
    .TextMatrix(0, 1) = ""
    .TextMatrix(0, 2) = "Customer Type"
    .TextMatrix(0, 3) = "Discount Rate"
    .TextMatrix(0, 4) = "Remark"


    'Fix List Head
    .FixedRows = 1

    'Set Alignment of Column
    For i = 0 To 4
      .ColAlignment(i) = 0
    Next i


    'Align List Head Center
    .FillStyle = flexFillRepeat
    .Col = 0
    .Row = 0
    .RowSel = 1
    .ColSel = .Cols - 1
    .CellAlignment = 4

    'Set Size of Cell
    .ColWidth(0) = 700
    .ColWidth(1) = 0
    .ColWidth(2) = 2000
    .ColWidth(3) = 2000
    .ColWidth(4) = 4000
    .Row = 1

  End With
End Sub
```

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR OMNIA VINCIT
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

```
Private Sub msgList_MouseUp(Button As Integer, Shift As Integer, x As Single, y As
Single)
   'Popup Right Key
   If Button = 2 And Shift = 0 Then

   End If

End Sub
```

<Input Passenger Type>

```
Option Explicit
'Whether Changed Records，ture is changed
Dim mblChange As Boolean
Dim mrc As ADODB.Recordset
Public txtSQL As String

Private Sub cmdExit_Click()
   If mblChange And cmdSave.Enabled Then
      If MsgBox(" Do you want to save record changed ? ", vbOKCancel +
vbExclamation, "Warning") = vbOK Then
         'Save
         Call cmdSave_Click
      End If
   End If
   Unload Me
End Sub

Private Sub cmdSave_Click()
   Dim intCount As Integer
   Dim sMeg As String
   Dim MsgText As String


   For intCount = 0 To 1
      If Trim(txtItem(intCount) & " ") = "" Then
         Select Case intCount
            Case 0
               sMeg = "Customer Type"
            Case 1
               sMeg = "Discount Rate"
         End Select
         sMeg = sMeg & " cannot be null！ "
         MsgBox sMeg, vbOKOnly + vbExclamation, "Warning"
         txtItem(intCount).SetFocus
         Exit Sub
      End If
   Next intCount
```

```
    If Not IsNumeric(Trim(txtItem(1))) Then
        sMeg = "Discount Rate,"
        sMeg = sMeg & " Please input number!  "
        MsgBox sMeg, vbOKOnly + vbExclamation, "Warning"
        txtItem(1).SetFocus
        Exit Sub
    End If

    'Judge whether exists same record
    txtSQL = "select * from customerType where ctypeNO<>'" & Trim(txtNo) & "' and
ctypeName='" & Trim(txtItem(0)) & "'"
    Set mrc = ExecuteSQL(txtSQL, MsgText)
    If mrc.EOF = False Then
        MsgBox " Already existed a record with same customer type! ", vbOKOnly +
vbExclamation, "Warning"
        txtItem(0).SetFocus
        Exit Sub
    End If

    'Delete foregone record first
    txtSQL = "delete from customerType where ctypeNO='" & Trim(txtNo) & "'"
    Set mrc = ExecuteSQL(txtSQL, MsgText)

    'Then add new record
    txtSQL = "select * from customerType"
    Set mrc = ExecuteSQL(txtSQL, MsgText)
    mrc.AddNew
    mrc.Fields(0) = txtNo
    For intCount = 0 To 2
        mrc.Fields(intCount + 1) = Trim(txtItem(intCount))
    Next intCount
    mrc.Update
    mrc.Close
    If gintTmode = 1 Then
        MsgBox " Added record successfully! ", vbOKOnly + vbExclamation, "Adding
Record"
        For intCount = 0 To 2
            txtItem(intCount) = ""
        Next intCount

        mblChange = False
        Unload frmcType
        frmcType.txtSQL = "select * from customerType"
        frmcType.Show

    ElseIf gintTmode = 2 Then
        Unload Me
        Unload frmcType
```

```
        frmcType.txtSQL = "select * from customerType"
        frmcType.Show
    End If

End Sub

Private Sub Form_Load()
    Dim intCount As Integer
    Dim MsgText As String
    Dim i As Integer

    If gintTmode = 1 Then
        Me.Caption = Me.Caption & "Add"
        txtNo = GetRkno()
    ElseIf gintTmode = 2 Then
        Set mrc = ExecuteSQL(txtSQL, MsgText)

        If mrc.EOF = False Then
            With mrc
                txtNo = .Fields(0)

                For intCount = 0 To 2
                    txtItem(intCount) = .Fields(intCount + 1)
                Next intCount
            End With
        End If
        Me.Caption = Me.Caption & "Modify"
    End If
    mblChange = False
End Sub
Private Sub Form_Unload(Cancel As Integer)
    gintTmode = 0
End Sub

Private Sub txtItem_Change(Index As Integer)
    'Exists change then set gblchange
    mblChange = True
End Sub

Private Sub txtItem_GotFocus(Index As Integer)
    txtItem(Index).SelStart = 0
    txtItem(Index).SelLength = Len(txtItem(Index))
End Sub

Private Sub txtItem_KeyDown(Index As Integer, KeyCode As Integer, Shift As Integer)
    EnterToTab KeyCode
End Sub
```

<Manage Passenger Information>

St. Gabriel's Library, Au

```
Option Explicit
Public txtSQL As String
Dim mrc As ADODB.Recordset
Private Sub cmdAdd_Click()
   gintCmode = 1
   frmCustomer1.Show 1
End Sub
Private Sub cmdDelete_Click()
   Dim txtSQL As String
   Dim intCount As Integer
   Dim mrc As ADODB.Recordset
   Dim MsgText As String

   If msgList.Rows > 1 Then
      If MsgBox(" Are you sure you want to delete record of customer who name is " &
Trim(msgList.TextMatrix(msgList.Row, 2)) & "? ", vbOKCancel + vbExclamation,
"Warning") = vbOK Then
         intCount = msgList.Row
         txtSQL = "delete from customerInfo where customerNO ='" &
Trim(msgList.TextMatrix(intCount, 1)) & "'"
         Set mrc = ExecuteSQL(txtSQL, MsgText)

         Unload frmCustomer
         frmCustomer.txtSQL = "select * from customerInfo"
         frmCustomer.Show
      End If
   End If
End Sub

Private Sub cmdExit_Click()
   Unload Me
End Sub

Private Sub cmdModify_Click()
   Dim intCount As Integer

   If frmCustomer.msgList.Rows > 1 Then
      gintCmode = 2
      intCount = msgList.Row
      If intCount > 0 Then
        frmCustomer1.txtSQL = "select * from customerInfo where customerNO ='" &
Trim(msgList.TextMatrix(intCount, 1)) & "'"
        frmCustomer1.Show 1
      Else
        MsgBox "Warning", vbOKOnly + vbExclamation, " Please select record you
want to modify first ! "
      End If
   End If
```

```
End Sub

Private Sub Command3_Click()
   frmCustomer2.Show 1
End Sub

Private Sub Form_Load()
   ShowTitle
   ShowData
End Sub

Private Sub Form_Resize()
   If Me.WindowState <> vbMinimized And fMainForm.WindowState <>
vbMinimized Then
      'Boundary processing
      If Me.ScaleHeight < 10 * lblTitle.Height Then
         Exit Sub
      End If
      If Me.ScaleWidth < lblTitle.Width + lblTitle.Width / 2 Then
         Exit Sub
      End If
      'The position of widget controlling

      lblTitle.Top = lblTitle.Height
      lblTitle.Left = (Me.Width - lblTitle.Width) / 2

      msgList.Top = lblTitle.Top + lblTitle.Height + lblTitle.Height / 2
      msgList.Width = Me.ScaleWidth - 200
      msgList.Left = Me.ScaleLeft + 100
      msgList.Height = Me.ScaleHeight - msgList.Top - 1500

      Frame2.Top = msgList.Top + msgList.Height + 50
      Frame2.Left = Me.ScaleWidth / 2 - 3000
   End If
End Sub

Public Sub FormClose()
   Unload Me
End Sub
Private Sub ShowData()
   Dim j As Integer
   Dim i As Integer
   Dim MsgText As String

   Set mrc = ExecuteSQL(txtSQL, MsgText)
      With msgList
      .Rows = 1

      Do While Not mrc.EOF
         .Rows = .Rows + 1
```

```
            For i = 1 To mrc.Fields.Count
               If Not IsNull(Trim(mrc.Fields(i - 1))) Then
               Select Case mrc.Fields(i - 1).Type
                  Case adDBDate
                  .TextMatrix(.Rows - 1, i) = Format(mrc.Fields(i - 1) & "", "yyyy-mm-dd")
                  Case Else
                     .TextMatrix(.Rows - 1, i) = mrc.Fields(i - 1) & ""
               End Select
               End If
            Next i
            mrc.MoveNext
         Loop

      End With
      mrc.Close
   End Sub

   'Display List Head of Grid
   Private Sub ShowTitle()
      Dim i As Integer

      With msgList
         .Cols = 8
         .TextMatrix(0, 1) = ""
         .TextMatrix(0, 2) = "Name"
         .TextMatrix(0, 3) = "Gender"
         .TextMatrix(0, 4) = "ID No."
         .TextMatrix(0, 5) = "Telephone"
         .TextMatrix(0, 6) = "Customer Type"
         .TextMatrix(0, 7) = "Remark"

         'Fix List Head
         .FixedRows = 1

         'Set alignment of column
         For i = 0 To 7
            .ColAlignment(i) = 0
         Next i

         'Align list head center
         .FillStyle = flexFillRepeat
         .Col = 0
         .Row = 0
         .RowSel = 1
         .ColSel = .Cols - 1
         .CellAlignment = 4

         'Set size of cell
         .ColWidth(0) = 700
         .ColWidth(1) = 0
```

HUMANITAS UNIVERSITY OF THAILAND
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

```
        .ColWidth(2) = 1500
        .ColWidth(3) = 1500
        .ColWidth(4) = 2500
        .ColWidth(5) = 1500
        .ColWidth(6) = 1500
        .ColWidth(7) = 4500
        .Row = 1
    End With
End Sub

Private Sub msgList_MouseUp(Button As Integer, Shift As Integer, x As Single, y As
Single)
    'Popup right key
    If Button = 2 And Shift = 0 Then
    End If
End Sub
```

<Input Passenger Information>

```
Option Explicit
'Whether changed record，ture is changed
Dim mblChange As Boolean
Dim mrc As ADODB.Recordset
Public txtSQL As String

Private Sub cmdExit_Click()
    If mblChange And cmdSave.Enabled Then
        If MsgBox(" Are you sure you want to save current record changed ? ",
vbOKCancel + vbExclamation, "Warning") = vbOK Then
            'Save
            Call cmdSave_Click
        End If
    End If
    Unload Me
End Sub

Private Sub cmdSave_Click()
    Dim intCount As Integer
    Dim sMeg As String
    Dim MsgText As String


    For intCount = 0 To 2
        If Trim(txtItem(intCount) & " ") = "" Then
            Select Case intCount
                Case 0
                    sMeg = "Name"
                Case 1
                    sMeg = "ID No."
```

```
            Case 2
               sMeg = "Telephone"
         End Select
         sMeg = sMeg & " cannot be null !  "
         MsgBox sMeg, vbOKOnly + vbExclamation, "Warning"
         txtItem(intCount).SetFocus
         Exit Sub
      End If
   Next intCount

   For intCount = 0 To 1
      If Trim(Combo1(intCount) & " ") = "" Then
         Select Case intCount
            Case 0
               sMeg = "Gender"
            Case 1
               sMeg = "Customer Type"
         End Select
         sMeg = sMeg & " cannot be null !  "
         MsgBox sMeg, vbOKOnly + vbExclamation, "Warning"
         Combo1(intCount).SetFocus
         Exit Sub
      End If
   Next intCount

   'Judge whether exists same record
   txtSQL = "select * from customerInfo where customerNO<>'" & Trim(txtNo) & "'
and customerName='" & Trim(txtItem(0)) & "'"
   Set mrc = ExecuteSQL(txtSQL, MsgText)
   If mrc.EOF = False Then
      MsgBox " Existed a record of same customer information ! ", vbOKOnly +
vbExclamation, "Warning"
      txtItem(0).SetFocus
      Exit Sub
   End If

   'Delete a foregone record
   txtSQL = "delete from customerInfo where customerNO='" & Trim(txtNo) & "'"
   Set mrc = ExecuteSQL(txtSQL, MsgText)

   'Then add an new record
   txtSQL = "select * from customerInfo"
   Set mrc = ExecuteSQL(txtSQL, MsgText)
   mrc.AddNew

   mrc.Fields(0) = Trim(txtNo)
   mrc.Fields(1) = Trim(txtItem(0))
   mrc.Fields(2) = Trim(Combo1(0))
   For intCount = 1 To 2
      mrc.Fields(intCount + 2) = Trim(txtItem(intCount))
```

```
    Next intCount

    mrc.Fields(5) = Trim(Combo1(1))
    mrc.Fields(6) = Trim(txtItem(3))
    mrc.Update
    mrc.Close

    If gintCmode = 1 Then
       MsgBox " Added a record successfully!  ", vbOKOnly + vbExclamation, "Adding
record"
       For intCount = 0 To 3
          txtItem(intCount) = ""
       Next intCount

       mblChange = False
       Unload frmCustomer
       frmCustomer.txtSQL = "select * from customerInfo"
       frmCustomer.Show

    ElseIf gintCmode = 2 Then
       Unload Me
       Unload frmCustomer

       frmCustomer.txtSQL = "select * from customerInfo"
       frmCustomer.Show
    End If

End Sub

Private Sub Form_Load()
    Dim intCount As Integer
    Dim MsgText As String
    Dim i As Integer
    Dim mrcc As ADODB.Recordset

    If gintCmode = 1 Then
       Me.Caption = Me.Caption & "Add"
       txtNo = GetRkno()

    ElseIf gintCmode = 2 Then
       Set mrc = ExecuteSQL(txtSQL, MsgText)

       If mrc.EOF = False Then
          With mrc
             txtNo = .Fields(0)
             txtItem(0) = .Fields(1)

             For intCount = 1 To 2
                txtItem(intCount) = .Fields(intCount + 2)
             Next intCount
```

```
            txtItem(3) = .Fields(6)
          End With
        End If
        Me.Caption = Me.Caption & "Modify"
    End If
    For intCount = 0 To 1
        Combo1(intCount).Clear
    Next intCount

    Combo1(0).AddItem "Male"
    Combo1(0).AddItem "Female"

    txtSQL = "select DISTINCT ctypeName from customerType"
    Set mrcc = ExecuteSQL(txtSQL, MsgText)
    If Not mrcc.EOF Then
        Do While Not mrcc.EOF
            Combo1(1).AddItem mrcc.Fields(0)
            mrcc.MoveNext
        Loop
    End If
    mrcc.Close
    mblChange = False
End Sub

Private Sub Form_Unload(Cancel As Integer)
    gintCmode = 0
End Sub

Private Sub txtItem_Change(Index As Integer)
    'Exists change then set gblchange
    mblChange = True
End Sub

Private Sub txtItem_GotFocus(Index As Integer)
    txtItem(Index).SelStart = 0
    txtItem(Index).SelLength = Len(txtItem(Index))
End Sub

Private Sub txtItem_KeyDown(Index As Integer, KeyCode As Integer, Shift As Integer)
    EnterToTab KeyCode
End Sub
```

<Inquiry Passenger Information>

```
Option Explicit
Private Sub chkItem_Click(Index As Integer)
    If Index = 0 Then
        Combo1(0).SetFocus
    Else
```

```
        Combo1(1).SetFocus
    End If
End Sub

Private Sub cmdExit_Click()
    Me.Hide
End Sub

Private Sub cmdOK_Click()
    Dim sQSql As String


    If chkItem(0).Value = vbChecked Then
        sQSql = " customerName = '" & Trim(Combo1(0) & " ") & "'"
    End If

    If chkItem(1).Value = vbChecked Then
        If Trim(sQSql & " ") = "" Then
            sQSql = " customerType ='" & Trim(Combo1(1) & " ") & "'"
        Else
            sQSql = sQSql & " and customerType ='" & Trim(Combo1(1) & " ") & "'"
        End If
    End If

    If Trim(sQSql) = "" Then
        MsgBox " Please set condition of query ! ", vbOKOnly + vbExclamation,
"Warning"
        Me.Hide
        Exit Sub
    Else
        frmCustomer.txtSQL = "select * from customerInfo where" & sQSql
        Me.Hide
        Unload frmCustomer
        frmCustomer.Show
    End If

End Sub
Private Sub Form_Load()
    Dim i As Integer
    Dim sSql As String
    Dim txtSQL As String
    Dim MsgText As String
    Dim mrc As ADODB.Recordset

    For i = 0 To 1
        Combo1(i).Clear
    Next i

    txtSQL = "select DISTINCT customerName from customerInfo"
    Set mrc = ExecuteSQL(txtSQL, MsgText)
```

```
    If Not mrc.EOF Then

       Do While Not mrc.EOF
          Combo1(0).AddItem Trim(mrc.Fields(0))
          mrc.MoveNext
       Loop
    Else
       MsgBox " Please set customer information first! ", vbOKOnly + vbExclamation,
"Warning"
       Exit Sub
    End If
    mrc.Close

    txtSQL = "select DISTINCT customerType from customerInfo"
    Set mrc = ExecuteSQL(txtSQL, MsgText)

    If Not mrc.EOF Then

       Do While Not mrc.EOF
          Combo1(1).AddItem Trim(mrc.Fields(0))
          mrc.MoveNext
       Loop
    Else
       MsgBox " Please set customer information first! ", vbOKOnly + vbExclamation,
"Warning"
       Exit Sub
    End If
    mrc.Close


End Sub

Private Sub lblitem_Click(Index As Integer)
    chkItem(Index).Value = vbChecked

End Sub

Private Sub txtItem_GotFocus(Index As Integer)
    txtItem(Index).SelStart = 0
    txtItem(Index).SelLength = Len(txtItem(Index))
End Sub
```

<Manage Cabin Information>

```
Option Explicit
Public txtSQL As String
Dim mrc As ADODB.Recordset
```

```
Private Sub cmdAdd_Click()
   gintSmode = 1
   frmService1.Show 1

End Sub

Private Sub cmdDelete_Click()
   Dim txtSQL As String
   Dim intCount As Integer
   Dim mrc As ADODB.Recordset
   Dim MsgText As String


   If msgList.Rows > 1 Then
      If MsgBox(" Are you sure you want to delete record that cabin class is " &
Trim(msgList.TextMatrix(msgList.Row, 2)) & "? ", vbOKCancel + vbExclamation,
"Warning") = vbOK Then
         intCount = msgList.Row
         txtSQL = "delete from serviceInfo where serviceNO ='" &
Trim(msgList.TextMatrix(intCount, 1)) & "'"
         Set mrc = ExecuteSQL(txtSQL, MsgText)

         Unload frmService
         frmService.txtSQL = "select * from serviceInfo"
         frmService.Show
      End If
   End If


End Sub

Private Sub cmdExit_Click()
   Unload Me
End Sub

Private Sub cmdModify_Click()
   Dim intCount As Integer

   If frmService.msgList.Rows > 1 Then
      gintSmode = 2
      intCount = msgList.Row
      If intCount > 0 Then
        frmService1.txtSQL = "select * from serviceInfo where serviceNO ='" &
Trim(msgList.TextMatrix(intCount, 1)) & "'"
        frmService1.Show 1
      Else
        MsgBox "Warning", vbOKOnly + vbExclamation, " Please select record you
want to modify first! "
      End If
   End If
```

```
End Sub

Private Sub Form_Load()
   ShowTitle
   ShowData

End Sub

Private Sub Form_Resize()
   If Me.WindowState <> vbMinimized And fMainForm.WindowState <>
vbMinimized Then
      'Boundary processing
      If Me.ScaleHeight < 10 * lblTitle.Height Then

         Exit Sub
      End If
      If Me.ScaleWidth < lblTitle.Width + lblTitle.Width / 2 Then

         Exit Sub
      End If
      'The position of widget controlling

      lblTitle.Top = lblTitle.Height
      lblTitle.Left = (Me.Width - lblTitle.Width) / 2

      msgList.Top = lblTitle.Top + lblTitle.Height + lblTitle.Height / 2
      msgList.Width = Me.ScaleWidth - 200
      msgList.Left = Me.ScaleLeft + 100
      msgList.Height = Me.ScaleHeight - msgList.Top - 1500

      Frame2.Top = msgList.Top + msgList.Height + 50
      Frame2.Left = Me.ScaleWidth / 2 - 3000
   End If
End Sub

Public Sub FormClose()
   Unload Me
End Sub
Private Sub ShowData()

   Dim j As Integer
   Dim i As Integer
   Dim MsgText As String

   Set mrc = ExecuteSQL(txtSQL, MsgText)
      With msgList
      .Rows = 1

      Do While Not mrc.EOF
```

```
            .Rows = .Rows + 1
            For i = 1 To mrc.Fields.Count
               If Not IsNull(Trim(mrc.Fields(i - 1))) Then
               Select Case mrc.Fields(i - 1).Type
                  Case adDBDate
                  .TextMatrix(.Rows - 1, i) = Format(mrc.Fields(i - 1) & "", "yyyy-mm-dd")
                  Case Else
                     .TextMatrix(.Rows - 1, i) = mrc.Fields(i - 1) & ""
               End Select
               End If
            Next i
            mrc.MoveNext
         Loop

      End With
      mrc.Close

End Sub

'Display list head of Grid
Private Sub ShowTitle()
   Dim i As Integer

   With msgList
      .Cols = 12
      .TextMatrix(0, 1) = "Cabin Class No."
      .TextMatrix(0, 2) = "Cabin Class"
      .TextMatrix(0, 3) = "Gift"
      .TextMatrix(0, 4) = "Newspaper"
      .TextMatrix(0, 5) = "Beverage"
      .TextMatrix(0, 6) = "Meal"
      .TextMatrix(0, 7) = "Movie"
      .TextMatrix(0, 8) = "Change time"
      .TextMatrix(0, 9) = "Return a ticket"
      .TextMatrix(0, 10) = "Discount"
      .TextMatrix(0, 11) = "Remark"

      'Fix list head
      .FixedRows = 1

      'set alignment of column
      For i = 0 To 11
         .ColAlignment(i) = 0
      Next i

      'Align list head center
      .FillStyle = flexFillRepeat
      .Col = 0
      .Row = 0
      .RowSel = 1
```

ASSUMPTION UNIVERSITY OF THAILAND
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

```
        .ColSel = .Cols - 1
        .CellAlignment = 4

        'Set size of cell
        .ColWidth(0) = 700
        .ColWidth(1) = 2000
        .ColWidth(2) = 3000
        .ColWidth(3) = 2000
        .ColWidth(4) = 2000
        .ColWidth(5) = 2000
        .ColWidth(6) = 2000
        .ColWidth(7) = 2000
        .ColWidth(8) = 2000
        .ColWidth(9) = 2000
        .ColWidth(10) = 2000
        .ColWidth(11) = 4000
        .Row = 1

    End With
End Sub

Private Sub msgList_MouseUp(Button As Integer, Shift As Integer, x As Single, y As
Single)
    'Popup right key
    If Button = 2 And Shift = 0 Then
    End If
End Sub
```

<Input Cabin Information>

```
Option Explicit
'Whether chaned record，ture is changed
Dim mblChange As Boolean
Dim mrc As ADODB.Recordset
Public txtSQL As String

Private Sub cmdExit_Click()
    If mblChange And cmdSave.Enabled Then
        If MsgBox(" Are you sure you want to save a changed record？ ", vbOKCancel +
vbExclamation, "Warning") = vbOK Then
            'Save
            Call cmdSave_Click
        End If
    End If
    Unload Me
End Sub

Private Sub cmdSave_Click()
    Dim intCount As Integer
```

St. Gabriel's Library, Au

```
    Dim sMeg As String
    Dim MsgText As String


    For intCount = 0 To 1
       If Trim(txtItem(intCount) & " ") = "" Then
          Select Case intCount
             Case 0
                sMeg = "Cabin Class No."
             Case 1
                sMeg = "Cabin Class"
          End Select
          sMeg = sMeg & " cannot be null!  "
          MsgBox sMeg, vbOKOnly + vbExclamation, "Warning"
          txtItem(intCount).SetFocus
          Exit Sub
       End If
    Next intCount

    If Not IsNumeric(Trim(txtItem(0))) Then
       sMeg = "Cabin Class No.,"
       sMeg = sMeg & " Please input number!  "
       MsgBox sMeg, vbOKOnly + vbExclamation, "Warning"
       txtItem(0).SetFocus
    End If

    'Judge whether exist record with same ID
    If gintSmode = 1 Then
       txtSQL = "select * from serviceInfo where serviceNO='" & Trim(txtItem(0)) & "'"
       Set mrc = ExecuteSQL(txtSQL, MsgText)
       If mrc.EOF = False Then
          MsgBox " Existed a record with this ID! ", vbOKOnly + vbExclamation,
"Warning"
          txtItem(0).SetFocus
          Exit Sub
       End If
       mrc.Close
    End If

    'Judge whether exist same record
    txtSQL = "select * from serviceInfo where serviceNO<>'" & Trim(txtItem(0)) & "'
and serviceName='" & Trim(txtItem(1)) & "'"
    Set mrc = ExecuteSQL(txtSQL, MsgText)
    If mrc.EOF = False Then
       MsgBox " Already existed a record with same cabin class! ", vbOKOnly +
vbExclamation, "Warning"
       txtItem(1).SetFocus
       Exit Sub
    End If
```

```
    'Delete a foregone record first
    txtSQL = "delete from serviceInfo where serviceNO='" & Trim(txtItem(0)) & "'"
    Set mrc = ExecuteSQL(txtSQL, MsgText)


    'Then add a new record
    txtSQL = "select * from serviceInfo"
    Set mrc = ExecuteSQL(txtSQL, MsgText)
    mrc.AddNew


    For intCount = 0 To 1
       mrc.Fields(intCount) = Trim(txtItem(intCount))
    Next intCount

    For intCount = 0 To 7
       mrc.Fields(intCount + 2) = Trim(Combo1(intCount))
    Next intCount

    mrc.Fields(10) = Trim(txtItem(2))


    mrc.Update
    mrc.Close

    If gintSmode = 1 Then
       MsgBox " Added a record successfully!  ", vbOKOnly + vbExclamation, "Adding
record"
       For intCount = 0 To 1
          txtItem(intCount) = ""
       Next intCount

       For intCount = 0 To 3
          Combo1(intCount).ListIndex = 0
       Next intCount

       txtItem(2) = ""
       mblChange = False
       Unload frmService
       frmService.txtSQL = "select * from serviceInfo"
       frmService.Show

    ElseIf gintSmode = 2 Then
       Unload Me
       Unload frmService

       frmService.txtSQL = "select * from serviceInfo"
       frmService.Show
    End If
```

ASSUMPTION UNIVERSITY OF THAILAND
มหาวิทยาลัยอัสสัมชัญ
SINCE1969
LABOR OMNIA VINCIT

```
End Sub


Private Sub Form_Load()
   Dim intCount As Integer
   Dim MsgText As String
   Dim i As Integer

   If gintSmode = 1 Then
      Me.Caption = Me.Caption & "Add"
      For i = 0 To 7
         Combo1(i).AddItem "No"
         Combo1(i).AddItem "Yes"
         Combo1(i).ListIndex = 0
      Next i
   ElseIf gintSmode = 2 Then
      Set mrc = ExecuteSQL(txtSQL, MsgText)

      If mrc.EOF = False Then
         With mrc
            For intCount = 0 To 1
               txtItem(intCount) = .Fields(intCount)
            Next intCount

            txtItem(2) = .Fields(10)
            For i = 0 To 7
               Combo1(i).AddItem "No"
               Combo1(i).AddItem "Yes"
               Combo1(i).ListIndex = 0
            Next i

         End With
         txtItem(0).Enabled = False
      End If

      Me.Caption = Me.Caption & "Modify"
   End If

   mblChange = False
End Sub

Private Sub Form_Unload(Cancel As Integer)
   gintSmode = 0
End Sub

Private Sub txtItem_Change(Index As Integer)
   'Exist change then set gblchange
   mblChange = True
End Sub
```

ASSUMPTION UNIVERSITY OF THAILAND
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

```
Private Sub txtItem_GotFocus(Index As Integer)
   txtItem(Index).SelStart = 0
   txtItem(Index).SelLength = Len(txtItem(Index))
End Sub

Private Sub txtItem_KeyDown(Index As Integer, KeyCode As Integer, Shift As Integer)
   EnterToTab KeyCode
End Sub
```

<Manage Ticket Information>

```
Option Explicit
Public txtSQL As String
Dim mrc As ADODB.Recordset

Private Sub cmdAdd_Click()
   gintKmode = 1
   frmTicket1.Show 1

End Sub

Private Sub cmdDelete_Click()
   Dim txtSQL As String
   Dim intCount As Integer
   Dim mrc As ADODB.Recordset
   Dim MsgText As String


   If msgList.Rows > 1 Then
      If MsgBox(" Are you sure you want to delete a customer " &
Trim(msgList.TextMatrix(msgList.Row, 3)) & Trim(msgList.TextMatrix(msgList.Row,
9)) & " who book a ticket from " & Trim(msgList.TextMatrix(msgList.Row, 7)) & " to
" & Trim(msgList.TextMatrix(msgList.Row, 8)) & "? ", vbOKCancel + vbExclamation,
"Warning") = vbOK Then
         intCount = msgList.Row
         txtSQL = "delete from ticketInfo where ticketNO ='" &
Trim(msgList.TextMatrix(intCount, 1)) & "'"
         Set mrc = ExecuteSQL(txtSQL, MsgText)

         Unload frmTicket
         frmTicket.txtSQL = "select * from ticketInfo"
         frmTicket.Show
      End If
   End If


End Sub
```

```
Private Sub cmdExit_Click()
   Unload Me
End Sub

Private Sub cmdModify_Click()
   Dim intCount As Integer

   If frmTicket.msgList.Rows > 1 Then
      gintKmode = 2
      intCount = msgList.Row
      If intCount > 0 Then
        frmTicket1.txtSQL = "select * from ticketInfo where ticketNO ='" &
Trim(msgList.TextMatrix(intCount, 1)) & "'"
        frmTicket1.Show 1
      Else
        MsgBox "Warning", vbOKOnly + vbExclamation, "Please select a record you
want to modify! "
      End If
   End If

End Sub



Private Sub Command3_Click()
   frmTicket2.Show 1
End Sub

Private Sub Form_Load()
   ShowTitle
   ShowData

End Sub

Private Sub Form_Resize()
  If Me.WindowState <> vbMinimized And fMainForm.WindowState <>
vbMinimized Then
     'Boundary processing
     If Me.ScaleHeight < 10 * lblTitle.Height Then

        Exit Sub
     End If
     If Me.ScaleWidth < lblTitle.Width + lblTitle.Width / 2 Then

        Exit Sub
     End If
     'The position of widget controlling

     lblTitle.Top = lblTitle.Height
```

```
        lblTitle.Left = (Me.Width - lblTitle.Width) / 2

        msgList.Top = lblTitle.Top + lblTitle.Height + lblTitle.Height / 2
        msgList.Width = Me.ScaleWidth - 200
        msgList.Left = Me.ScaleLeft + 100
        msgList.Height = Me.ScaleHeight - msgList.Top - 1500

        Frame2.Top = msgList.Top + msgList.Height + 50
        Frame2.Left = Me.ScaleWidth / 2 - 3000
    End If
End Sub



Public Sub FormClose()
    Unload Me
End Sub
Private Sub ShowData()

    Dim j As Integer
    Dim i As Integer
    Dim MsgText As String


    Set mrc = ExecuteSQL(txtSQL, MsgText)
        With msgList
        .Rows = 1

        Do While Not mrc.EOF
            .Rows = .Rows + 1
            For i = 1 To mrc.Fields.Count
                If Not IsNull(Trim(mrc.Fields(i - 1))) Then
                Select Case mrc.Fields(i - 1).Type
                    Case adDBDate
                    .TextMatrix(.Rows - 1, i) = Format(mrc.Fields(i - 1) & "", "yyyy-mm-dd")
                    Case Else
                        .TextMatrix(.Rows - 1, i) = mrc.Fields(i - 1) & ""
                End Select
                End If
            Next i
            mrc.MoveNext
        Loop


    End With
    mrc.Close


End Sub
```

```
'Display list head of Grid
Private Sub ShowTitle()
    Dim i As Integer

    With msgList
        .Cols = 14
        .TextMatrix(0, 1) = ""
        .TextMatrix(0, 2) = ""
        .TextMatrix(0, 3) = "Name"
        .TextMatrix(0, 4) = "Customer Type"
        .TextMatrix(0, 5) = "Discount Rate"
        .TextMatrix(0, 6) = ""
        .TextMatrix(0, 7) = "Departure City"
        .TextMatrix(0, 8) = "Arrival City"
        .TextMatrix(0, 9) = "Departure Time"
        .TextMatrix(0, 10) = "Cabin Class"
        .TextMatrix(0, 11) = "Price"
        .TextMatrix(0, 12) = "Final Price"
        .TextMatrix(0, 13) = "Remark"

        'Fix list head
        .FixedRows = 1

        'Set alignement of column
        For i = 0 To 13
            .ColAlignment(i) = 0
        Next i


        'Aligne list head center
        .FillStyle = flexFillRepeat
        .Col = 0
        .Row = 0
        .RowSel = 1
        .ColSel = .Cols - 1
        .CellAlignment = 4

        'Set size of cell
        .ColWidth(0) = 700
        .ColWidth(1) = 0
        .ColWidth(2) = 0
        .ColWidth(3) = 2000
        .ColWidth(4) = 2000
        .ColWidth(5) = 2000
        .ColWidth(6) = 0
        .ColWidth(7) = 2000
        .ColWidth(8) = 2000
        .ColWidth(9) = 2000
        .ColWidth(10) = 2000
```

ASSUMPTION UNIVERSITY OF THAILAND
BROTHERS OF ST GABRIEL
LABOR VINCIT OMNIA
SINCE1969
มหาวิทยาลัยอัสสัมชัญ

```
        .ColWidth(11) = 2000
        .ColWidth(12) = 2000
        .ColWidth(13) = 4500
        .Row = 1

    End With
End Sub



Private Sub msgList_MouseUp(Button As Integer, Shift As Integer, x As Single, y As
Single)
    'Popup right key
    If Button = 2 And Shift = 0 Then

    End If

End Sub
```

<Booking Ticket>

```
Option Explicit
'Whether changed record,ture is changed
Dim mblChange As Boolean
Dim mrc As ADODB.Recordset
Public txtSQL As String

Private Sub cmdExit_Click()
    If mblChange And cmdSave.Enabled Then
        If MsgBox(" Are you sure you want to save a changed record? ", vbOKCancel +
vbExclamation, "Warning") = vbOK Then
            'Save
            Call cmdSave_Click
        End If
    End If
    Unload Me
End Sub

Private Sub cmdSave_Click()
    Dim intCount As Integer
    Dim sMeg As String
    Dim MsgText As String


    For intCount = 0 To 6
        If Trim(Combo1(intCount) & " ") = "" Then
            Select Case intCount
                Case 0
```

```
                sMeg = "Departure City"
            Case 1
                sMeg = "Arrival City"
            Case 2
                sMeg = "Cabin Class"
            Case 3
                sMeg = ""
            Case 4
                sMeg = "Name"
            Case 5
                sMeg = ""
            Case 6
                sMeg = ""
        End Select
        sMeg = sMeg & " cannot be null ! "
        MsgBox sMeg, vbOKOnly + vbExclamation, "Warning"
        Combo1(intCount).SetFocus
        Exit Sub
    End If
  Next intCount

  If Trim(txtItem(0)) = "" Then
    sMeg = " Departure Time cannot be null ! "
    MsgBox sMeg, vbOKOnly + vbExclamation, "Warning"
    txtItem(0).SetFocus
    Exit Sub
  End If

  If IsDate(txtItem(0)) Then
    txtItem(0) = Format(txtItem(0), "yyyy-mm-dd")
  Else
    MsgBox " The format of time is ( yyyy-mm-dd ) ! ", vbOKOnly +
vbExclamation, "Warning"
    txtItem(0).SetFocus
    Exit Sub
  End If

  'Judge whether exist same record
  txtSQL = "select * from ticketInfo where ticketNO<>'" & Trim(txtticket) & "' and
customerNO ='" & Trim(txtcustomer) & "' and ticketDate = '" & Trim(txtItem(0)) & "'"
  Set mrc = ExecuteSQL(txtSQL, MsgText)
  If mrc.EOF = False Then
    MsgBox " Already existed a booking record with same customer ! ", vbOKOnly +
vbExclamation, "Warning"
    txtItem(1).SetFocus
    Exit Sub
  End If

  'Delete a foregone record first
  txtSQL = "delete from ticketInfo where ticketNO='" & Trim(txtticket) & "'"
```

```
    Set mrc = ExecuteSQL(txtSQL, MsgText)

    'Judge whether full seat in a flight
    txtSQL = "select * from ticketInfo where airlineNO = '" & Trim(txtairline) & " ' and
ticketDate = '" & Trim(txtItem(0)) & "' and serviceType = '" & Trim(Combo1(2)) & "'"
    Set mrc = ExecuteSQL(txtSQL, MsgText)

    If Not mrc.EOF Then
       If Combo1(2) = "Economy Class" Then
          If mrc.RecordCount > (ticketCsum - 1) Then
             MsgBox "Sorry，The seats of ecinomy class is full in this flight", vbOKOnly
+ vbExclamation, "Warning"
             Exit Sub
          End If
       ElseIf Combo1(2) = "Business Class" Then
          If mrc.RecordCount > (ticketMsum - 1) Then
             MsgBox "Sorry，The seats of business class is full in this flight", vbOKOnly
+ vbExclamation, "Warning"
             Exit Sub
          End If
       Else
          If mrc.RecordCount > (ticketFsum - 1) Then
             MsgBox "Sorry，The seats of first class is full in this flight", vbOKOnly +
vbExclamation, "Warning"
             Exit Sub
          End If
       End If
    End If

    mrc.Close


    'Add a new record again
    txtSQL = "select * from ticketInfo"
    Set mrc = ExecuteSQL(txtSQL, MsgText)
    mrc.AddNew

    mrc.Fields(0) = txtticket
    mrc.Fields(1) = txtcustomer


    For intCount = 2 To 4
       mrc.Fields(intCount) = Trim(Combo1(intCount + 2))
    Next intCount

    mrc.Fields(5) = txtairline

    For intCount = 6 To 7
       mrc.Fields(intCount) = Trim(Combo1(intCount - 6))
    Next intCount
```

```
    mrc.Fields(8) = Trim(txtItem(0))

    For intCount = 9 To 10
       mrc.Fields(intCount) = Trim(Combo1(intCount - 7))
    Next intCount

    mrc.Fields(11) = Trim(Combo1(3) * Combo1(6) / 100)
    mrc.Fields(12) = txtItem(1)



    mrc.Update
    mrc.Close

    If gintKmode = 1 Then
       MsgBox " Book successfully! The flight is " & Trim(txtItem(0)) & " from " &
Trim(Combo1(0)) & " to " & Trim(Combo1(1)) & ", it is tocket of " &
Trim(Combo1(2)) & ", the price is RMB￥" & Trim(Combo1(3) * Combo1(6) / 100)
& "Yuan! ", vbOKOnly + vbExclamation, "Booking record"
       For intCount = 0 To 1
          txtItem(intCount) = ""
       Next intCount

       mblChange = False
       Unload frmTicket
       frmTicket.txtSQL = "select * from ticketInfo"
       frmTicket.Show

    ElseIf gintKmode = 2 Then
       MsgBox " Modify successfully ! The flight after modification is " &
Trim(txtItem(0)) & " from " & Trim(Combo1(0)) & " to " & Trim(Combo1(1)) & ", it is
ticket of " & Trim(Combo1(2)) & ", the price is RMB￥" & Trim(Combo1(3) *
Combo1(6) / 100) & "Yuan! ", vbOKOnly + vbExclamation, "Booking record"
       Unload Me
       Unload frmTicket

       frmTicket.txtSQL = "select * from ticketInfo"
       frmTicket.Show
    End If

    Dialog.Show

End Sub


Private Sub Combo1_Click(Index As Integer)
    Dim mrcc As ADODB.Recordset
    Dim mrcd As ADODB.Recordset
```

```
    Dim MsgText As String

    If Index = 2 Then
       If Not (Trim(Combo1(0)) = "" Or Trim(Combo1(1)) = "") Then
          txtSQL = "select *  from airlineInfo where departCity = '" & Trim(Combo1(0))
 & "' and arrivalCity = '" & Trim(Combo1(1)) & "'"
          Set mrcc = ExecuteSQL(txtSQL, MsgText)

          If Not mrcc.EOF Then
             txtairline = mrcc.Fields(0)
             ticketCsum = mrcc!planeNO
             If Trim(Combo1(2)) = "Economy Class" Then
                Combo1(3).Clear
                Combo1(3).AddItem mrcc.Fields(7)
                Combo1(3).ListIndex = 0
             ElseIf Trim(Combo1(2)) = "Buiness Class" Then
                Combo1(3).Clear
                Combo1(3).AddItem mrcc.Fields(8)
                Combo1(3).ListIndex = 0
             Else
                Combo1(3).Clear
                Combo1(3).AddItem mrcc.Fields(9)
                Combo1(3).ListIndex = 0
             End If
          End If

          mrcc.Close

       txtSQL = "select * from planeInfo where planeNO = '" & Trim(ticketCsum) & "'"
          Set mrcc = ExecuteSQL(txtSQL, MsgText)

          If Not mrcc.EOF Then
             ticketCsum = mrcc.Fields(4)
             ticketMsum = mrcc.Fields(5)
             ticketFsum = mrcc.Fields(6)
          End If

          mrcc.Close
       End If
    ElseIf Index = 4 Then
       txtSQL = "select customerNO,customerType from customerInfo where
customerName = '" & Trim(Combo1(4)) & "'"
       Set mrcc = ExecuteSQL(txtSQL, MsgText)

       If Not mrcc.EOF Then
          txtcustomer = mrcc.Fields(0)
          Combo1(5).Clear
          Combo1(5).AddItem mrcc.Fields(1)
          Combo1(5).ListIndex = 0
```

```
        txtSQL = "select distinct discount from customerType where ctypeName = '" &
Trim(Combo1(5)) & "'"
        Set mrcd = ExecuteSQL(txtSQL, MsgText)

        If Not mrcd.EOF Then
           Combo1(6).Clear
           Combo1(6).AddItem mrcd.Fields(0)
           Combo1(6).ListIndex = 0
        End If

        mrcd.Close
     End If

     mrcc.Close

  End If

End Sub

Private Sub Form_Load()
  Dim intCount As Integer
  Dim MsgText As String
  Dim i As Integer
  Dim mrcc As ADODB.Recordset

  If gintKmode = 1 Then
     Me.Caption = Me.Caption & "Add"

     txtticket = GetRkno()

  ElseIf gintKmode = 2 Then
     Set mrc = ExecuteSQL(txtSQL, MsgText)

     If mrc.EOF = False Then
        With mrc
           txtticket = .Fields(0)
           txtItem(0) = .Fields(7)
           txtItem(1) = .Fields(12)
        End With
     End If

     Me.Caption = Me.Caption & "Modify"
  End If

     For i = 0 To 6
        Combo1(i).Clear
     Next i

     txtSQL = "select distinct departCity from airlineInfo "
     Set mrcc = ExecuteSQL(txtSQL, MsgText)
```

```
    If Not mrcc.EOF Then
       Do While Not mrcc.EOF
          Combo1(0).AddItem mrcc.Fields(0)
          mrcc.MoveNext
       Loop
    End If

    mrcc.Close

    txtSQL = "select distinct arrivalCity from airlineInfo"
    Set mrcc = ExecuteSQL(txtSQL, MsgText)

    If Not mrcc.EOF Then
       Do While Not mrcc.EOF
          Combo1(1).AddItem mrcc.Fields(0)
          mrcc.MoveNext
       Loop
    End If
    mrcc.Close

    txtSQL = "select distinct serviceName from serviceInfo"
    Set mrcc = ExecuteSQL(txtSQL, MsgText)

    If Not mrcc.EOF Then
       Do While Not mrcc.EOF
          Combo1(2).AddItem mrcc.Fields(0)
          mrcc.MoveNext
       Loop
    End If
    mrcc.Close

    txtSQL = "select distinct customerName from customerInfo"
    Set mrcc = ExecuteSQL(txtSQL, MsgText)

    If Not mrcc.EOF Then
       Do While Not mrcc.EOF
          Combo1(4).AddItem mrcc.Fields(0)
          mrcc.MoveNext
       Loop
    End If
    mrcc.Close

  mblChange = False
End Sub

Private Sub Form_Unload(Cancel As Integer)
  gintKmode = 0
End Sub
```

St. Gabriel's Library, Au

```
Private Sub txtItem_Change(Index As Integer)
   'exist change then set gblchange
   mblChange = True
End Sub

Private Sub txtItem_GotFocus(Index As Integer)
   txtItem(Index).SelStart = 0
   txtItem(Index).SelLength = Len(txtItem(Index))
End Sub

Private Sub txtItem_KeyDown(Index As Integer, KeyCode As Integer, Shift As Integer)
   EnterToTab KeyCode
End Sub
```

<Inquiry Ticket Information>

```
Option Explicit
Private Sub cmdExit_Click()
   Me.Hide
End Sub

Private Sub cmdOK_Click()
   Dim txtSQL As String
   Dim MsgText As String
   Dim mrc As ADODB.Recordset
   Dim intCount As Integer
   Dim sMeg As String
   Dim i, j, k


   For intCount = 0 To 1
      If Trim(Combo1(intCount) & " ") = "" Then
         Select Case intCount
            Case 0
               sMeg = "Departure City"
            Case 1
               sMeg = "Arrival City"
         End Select
         sMeg = sMeg & " cannot be null!  "
         MsgBox sMeg, vbOKOnly + vbExclamation, "Warning"
         Combo1(intCount).SetFocus
         Exit Sub
      End If
   Next intCount

   If Trim(txtItem(0)) = "" Then
      sMeg = " The departure time cannot be null!  "
      MsgBox sMeg, vbOKOnly + vbExclamation, "Warning"
```

```
        txtItem(0).SetFocus
        Exit Sub
    End If

    If IsDate(txtItem(0)) Then
        txtItem(0) = Format(txtItem(0), "yyyy-mm-dd")
    Else
        MsgBox " The format of time is （yyyy-mm-dd）！", vbOKOnly + vbExclamation, "Warning"
        txtItem(0).SetFocus
        Exit Sub
    End If

    If Not (Trim(Combo1(2)) = "") Then
        txtSQL = "select * from ticketInfo where ticketDate = '" & Trim(txtItem(0)) & "' and departCity = '" & Trim(Combo1(0)) & "' and arrivalCity = '" & Trim(Combo1(1)) & "' and serviceType = '" & Trim(Combo1(2)) & "'"
        Set mrc = ExecuteSQL(txtSQL, MsgText)

        If Trim(Combo1(2)) = "Economy Class" Then
            intCount = ticketCsum - mrc.RecordCount
        ElseIf Trim(Combo1(2)) = "Business Class" Then
            intCount = ticketMsum - mrc.RecordCount
        Else
            intCount = ticketFsum - mrc.RecordCount
        End If

        If intCount > 0 Then
            MsgBox Trim(txtItem(0)) & "The tickets from " & Trim(Combo1(0)) & " to " & Trim(Combo1(1)) & Trim(Combo1(2)) & " left " & intCount & " Sheet!", vbOKOnly + vbExclamation, "Ticket Information"
        Else
            MsgBox " Sorry," & Trim(txtItem(0)) & " The flight from " & Trim(Combo1(0)) & " to " & Trim(Combo1(1)) & Trim(Combo1(2)) & " is full，Please book other flight！", vbOKOnly + vbExclamation, "Ticket Information"
        End If
    Else
        txtSQL = "select * from ticketInfo where ticketDate = '" & Trim(txtItem(0)) & "' and departCity = '" & Trim(Combo1(0)) & "' and arrivalCity = '" & Trim(Combo1(1)) & "' and serviceType = 'Economy Class'"
        Set mrc = ExecuteSQL(txtSQL, MsgText)
        i = mrc.RecordCount
        mrc.Close

        txtSQL = "select * from ticketInfo where ticketDate = '" & Trim(txtItem(0)) & "' and departCity = '" & Trim(Combo1(0)) & "' and arrivalCity = '" & Trim(Combo1(1)) & "' and serviceType = 'Business Class'"
        Set mrc = ExecuteSQL(txtSQL, MsgText)
        j = mrc.RecordCount
        mrc.Close
```

```
        txtSQL = "select * from ticketInfo where ticketDate = '" & Trim(txtItem(0)) & "'
 and departCity = '" & Trim(Combo1(0)) & "' and arrivalCity = '" & Trim(Combo1(1))
 & "' and serviceType = 'First Class'"
        Set mrc = ExecuteSQL(txtSQL, MsgText)

        k = mrc.RecordCount

        mrc.Close

        MsgBox Trim(txtItem(0)) & " The economy class tickets from " &
 Trim(Combo1(0)) & " to " & Trim(Combo1(1)) & " left " & (ticketCsum - i) & "
 sheet,the buiness class tickets left " & (ticketMsum - j) & " sheet，the first clas tickets
 left " & (ticketFsum - k) & " sheet ! ", vbOKOnly + vbExclamation, "Ticket
 Information"
    End If
    Unload Me
End Sub

Private Sub Combo1_Click(Index As Integer)
    Dim mrcc As ADODB.Recordset
    Dim MsgText As String
    Dim txtSQL As String

    If Index = 1 Then
       If Not (Trim(Combo1(0)) = "") Then
          txtSQL = "select *  from airlineInfo where departCity = '" & Trim(Combo1(0))
& "' and arrivalCity = '" & Trim(Combo1(1)) & "'"
          Set mrcc = ExecuteSQL(txtSQL, MsgText)

          If Not mrcc.EOF Then
             ticketCsum = mrcc!planeNO
          End If
          mrcc.Close

      txtSQL = "select * from planeInfo where planeNO = '" & Trim(ticketCsum) & "'"
          Set mrcc = ExecuteSQL(txtSQL, MsgText)

          If Not mrcc.EOF Then
             ticketCsum = mrcc.Fields(4)
             ticketMsum = mrcc.Fields(5)
             ticketFsum = mrcc.Fields(6)
          End If

          mrcc.Close
       End If
    End If
End Sub

Private Sub Form_Load()
    Dim i As Integer
```

```
    Dim sSql As String
    Dim txtSQL As String
    Dim MsgText As String
    Dim mrc As ADODB.Recordset

    For i = 0 To 2
       Combo1(i).Clear
    Next i

    txtSQL = "select distinct departCity from airlineInfo "
    Set mrc = ExecuteSQL(txtSQL, MsgText)

    If Not mrc.EOF Then
       Do While Not mrc.EOF
          Combo1(0).AddItem mrc.Fields(0)
          mrc.MoveNext
       Loop
    End If

    mrc.Close

    txtSQL = "select distinct arrivalCity from airlineInfo"
    Set mrc = ExecuteSQL(txtSQL, MsgText)

    If Not mrc.EOF Then
       Do While Not mrc.EOF
          Combo1(1).AddItem mrc.Fields(0)
          mrc.MoveNext
       Loop
    End If
    mrc.Close

    txtSQL = "select distinct serviceName from serviceInfo"
    Set mrc = ExecuteSQL(txtSQL, MsgText)

    If Not mrc.EOF Then
       Do While Not mrc.EOF
          Combo1(2).AddItem mrc.Fields(0)
          mrc.MoveNext
       Loop
    End If
    mrc.Close

End Sub

Private Sub txtItem_GotFocus(Index As Integer)
   txtItem(Index).SelStart = 0
   txtItem(Index).SelLength = Len(txtItem(Index))
End Sub
```

ASSUMPTION UNIVERSITY OF THAILAND
SINCE1969

<About>

```
' Reg Key Security Options...
Const KEY_ALL_ACCESS = &H2003F

' Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1                    ' Unicode nul terminated string
Const REG_DWORD = 4                 ' 32-bit number
Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA"
(ByVal hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal
samDesired As Long, ByRef phkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias
"RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String, ByVal
lpReserved As Long, ByRef lpType As Long, ByVal lpData As String, ByRef lpcbData
As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As
Long

Private Sub Form_Load()
  lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." &
App.Revision
  lblTitle.Caption = App.Title
End Sub

Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

Public Sub StartSysInfo()
  On Error GoTo SysInfoErr

    Dim rc As Long
    Dim SysInfoPath As String


    ' Try To Get System Info Program Path\Name From Registry...
    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,
gREGVALSYSINFO, SysInfoPath) Then
```

```
        ' Try To Get System Info Program Path Only From Registry...
        ElseIf    GetKeyValue(HKEY_LOCAL_MACHINE,    gREGKEYSYSINFOLOC,
gREGVALSYSINFOLOC, SysInfoPath) Then
            ' Validate Existance Of Known 32 Bit File Version
            If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
                SysInfoPath = SysInfoPath & "\MSINFO32.EXE"


            ' Error - File Can Not Be Found...
            Else
                GoTo SysInfoErr
            End If
        ' Error - Registry Entry Can Not Be Found...
        Else
            GoTo SysInfoErr
        End If

        Call Shell(SysInfoPath, vbNormalFocus)

        Exit Sub
SysInfoErr:
        MsgBox "System Information Is Unavailable At This Time", vbOKOnly
End Sub


Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef As
String, ByRef KeyVal As String) As Boolean
        Dim i As Long                           ' Loop Counter
        Dim rc As Long                          ' Return Code
        Dim hKey As Long                        ' Handle To An Open Registry Key
        Dim hDepth As Long                      '
        Dim KeyValType As Long                  ' Data Type Of A Registry Key
        Dim tmpVal As String                    ' Tempory Storage For A Registry
Key Value
        Dim KeyValSize As Long                  ' Size Of Registry Key Variable
        '------------------------------------------------------------
        ' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}
        '------------------------------------------------------------
        rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' Open
Registry Key


        If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError          ' Handle Error...


        tmpVal = String$(1024, 0)               ' Allocate Variable Space
        KeyValSize = 1024                       ' Mark Variable Size

        '------------------------------------------------------------
        ' Retrieve Registry Key Value...
```

```
        '------------------------------------------------------------
        rc = RegQueryValueEx(hKey, SubKeyRef, 0, KeyValType, tmpVal, KeyValSize)
' Get/Create Key Value


        If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError          ' Handle Errors
        tmpVal = VBA.Left(tmpVal, InStr(tmpVal, VBA.Chr(0)) - 1)
        '------------------------------------------------------------
        ' Determine Key Value Type For Conversion...
        '------------------------------------------------------------
        Select Case KeyValType                              ' Search Data Types...
        Case REG_SZ                                     ' String Registry Key Data Type
            KeyVal = tmpVal                             ' Copy String Value
        Case REG_DWORD                                  ' Double Word Registry Key Data
Type
            For i = Len(tmpVal) To 1 Step -1                ' Convert Each Bit
                KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1)))   ' Build Value Char.
By Char.
            Next
            KeyVal = Format$("&h" + KeyVal)                 ' Convert Double Word To
String
        End Select
        GetKeyValue = True                              ' Return Success
        rc = RegCloseKey(hKey)                          ' Close Registry Key
        Exit Function                                   ' Exit
GetKeyError:    ' Cleanup After An Error Has Occured...
        KeyVal = ""                                     ' Set Return Val To Empty String
        GetKeyValue = False                             ' Return Failure
        rc = RegCloseKey(hKey)                          ' Close Registry Key
End Function
```

<Print Dialog>

```
Option Explicit
Private Sub OKButton_Click()
      On Error Resume Next
   With dlgCommonDialog
      .DialogTitle = "Print"
      .CancelError = True
      .Flags = 1
      Printer.FontSize = dlgCommonDialog.FontSize
      .ShowPrinter
      If Err <> MSComDlg.cdlCancel Then
        Printer.FontTransparent = False
        SetPrinterScale Myform
        PrintAnywhere Printer

Printer.NewPage
PrintAnywhere Printer
```

```
Printer.NewPage
            Printer.EndDoc
            Printer.KillDoc
        End If
    End With
End Sub
```

BIBLIOGRAPHY

1. Betley, Lonnie D., Kevin C. Dttman, and Jeffery L. Whitten. System Analysis and Design Methods, 4th Edition. Boston: Irwin McGraw Hill International, 1998.

2. Case, Thomas and Charles Parker. Management Information Systems Strategy and Action, 2nd Edition. Singapore: McGraw-Hill, 1993.

3. Charles, L. Phillips and H. Troy Nagle. Digital Control System Analysis and Design, 3rd Edition. NY: Prentice Hall, 1994.

4. Date, C. J. An Introduction to Database System, 6th Edition. MA: Addison Wesley Publishing Company, 1995.

5. Forouzan, Behrouz A. Introduction to Data Communication and Networking, 2nd Edition. Singapore: McGraw-Hill International, 1998.

6. Goldman, J. E. Client Server Information System: A Business Oriented Approach. NY: John Willey & Sons, Inc., 1999.

7. Jeff, Paulson. Production and Inventory Management Journal, 2nd Quarter, 1993: 77-81.

8. Kendall, Kenneth E. and Julie E. Kendall. System Analysis and Design, 5th Edition. New Jersey: Prentice-Hall, 2002.

9. Korth, F. Henry and Abraham Slberschatz. Database System Concepts. NY: McGraw-Hill International, 1991.

10. Kosiur, David. Understanding Electronic Commerce. Washington: Microsoft Press, 1997.

11. Laudon, Kenneth and Jane P. Landon. Management Information System. NJ: Prentice Hall Internation Company, 2000.

12. Page-Jones, Meilir. The Practical Guide to Structured System Design. NJ: Prentice-Hall International Company, 1998.

13. Pfleeger, Charles P. Security in Computing. United States of America: Prentice Hall, Inc., 1998.

14. Whitten, Jeffrey L., Lonnie D. Bentley, and Kevin C. Dittman. System Analysis and Design Methods, 5th Edition. Singapore: McGraw-Hill Higher Education, 2000.

15. Yourdon, Edwards. Modern Structured Analysis. NJ: Prentice Hall, 1989.

St. Gabriel's Library, Au