# Sale Management System for a Direct Sale Business

By

Ms. Ploy Suksrisomboon

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

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

July, 1998

# Sale Management System for a Direct Sale Business

by
Ms. Ploy Suksrisomboon

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

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

July 1998

| | |
|---|---|
| Project Title | Sale Management System for a Direct Sale Business |
| Name | Ms. Ploy Suksrisomboon |
| Project Advisor | Prof. Dr. Srisakdi Charmonman |
| Academic Year | 1998 |

---

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

Approved Committee:

| | |
|---|---|
| (Prof.Dr. Srisakdi Charmonman) | (Air Marshal Dr. Chulit Meesajjee) |
| Chairman and Advisor | Dean and Co-Advisor |
| (Dr.Sudhiporn Patumtaewapibal) | (Assoc.Prof.Somchai Thayarnyong) |
| Member | MUA Representative |

July 1998

# ABSTRACT

The Sale Management System is the proposed computerized system particularly designed to develop existing functions for sale and marketing department, purchasing department and inventory department of Grand Total Co.,Ltd. The existing system are subjected to many problems resulting from manual operations such as process delay, data out of date, etc. Presently, all functions related to the system such as collecting data, checking and processing the data more timely, and providing accurate information for the management are done manually.

The proposed system was designed to help the information collection function. The tools of structured analysis such as context diagram, dataflow diagram, and data dictionary are used for the analysis phase. The detailed design is carried out through file design, software design, screen design and report design. The main components of this design include the process of checking quantity of merchandise for sales, stock card of merchandise report for inventory control and etc.

Finally, the major outcome of this project is a computerized system that can help staff to fetch information in their jobs more quickly and efficiently. Sale and marketing department, inventory department and purchasing department could get the reports to help them in controlling, planning, forecasting and decision making.

# ACKNOWLEDGEMENTS

The author would like to take this opportunity to express her earnest appreciation to Prof. Dr.Srisakdi Charmonman, the advisor of this project for spending his valuable time in counseling the author on this project.

The author wishes to thank other committee members, Assoc.Prof. Somchai Thayarnyong, Air Marshal Dr. Chulit Meesajjee and Dr. Sudhiporn Patumtaewapibal for offering her an opportunity to accomplish this project.

Finally, the author is greatly thankful to all lecturers in the MS CIS program of Assumption University for their constant encouragement and constructive suggestion during the entire course of study.

# TABLE OF CONTENTS

# LIST OF FIGURES

| Figure | | Page |
|---|---|---|

**Figure** **Page**

| Figure | | Page |
|---|---|---|

| Figure | | Page |
|---|---|---|

# LIST OF TABLES

# I. INTRODUCTION

## 1.1 Background of the Project.

The Grand Total Co,Ltd. is a company in the group of company of Grand Group. It's the direct sale business to distribute the work 6 departments :

1. Sale and marketing department.
2. Inventory department.
3. Purchasing department.
4. Accounting and finance department.
5. Personal department.
6. Transportation department.

They buy product for the group of company of Grand Group and agent and distribute around every part in Thailand.

## 1.2 Objectives of the Project.

The objective of the project on Sale Management System or the sale administrator can be summarized as follows :

1. To analyse the present system and identify all exsiting problems.
2. To obtain the management's requirements pertaining to the new systems.
3. To support all level management.
4. To develop Sale Management System for the business of Grand Total Co., Ltd.

## 1.3 Scope of the Project.

Scope of the project will cover the important business of the direct sale and are summarized as follows :

1. Inventory management is controlling merchandise quantity to be adequate and correct for sale and service.
2. Purchase order managenent is controlling, checking the pending order of customer and inventory.
3. Sale management is generating various kind of summary reports.

## 1.4 Deliverables

The deliverables for the project on Sale Management System of Grand Total Co.,Ltd. are as follows :

1. Screen layouts for user-interface.
2. Various hard copy reports :

**1. Master File System**

1. Type of Merchandise
2. Group of Merchandise
3. Structure of Merchandise
4. Status of Merchandise
5. Merchandise Detail
6. Customer History
7. List of Bank Name
8. Vendor History
9. Condition Payment

**2. Inventory System**

1. Merchandise remain
2. Stock card of Merchandise
3. Merchandise unmovement
4. Merchandise under limited

**3. Purchase Order System**

1. Purchasing Order
2. Merchandise Receive
3. Merchandise Return
4. Pending Order
5. Price List

**4. Sale System**

1. Sale Request
2. Tax Invoice

## 1.5 Project Plan

Table 1.1. The Project Plan Represented in Gantt Chart

| ACTIVITIES | JAN'98 | | | | FEB'98 | | | | MAR'98 | | | | APR'98 | | | | MAY'98 | | | | JUN'98 | | | | JUL'98 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| **SYSTEM ANALYSIS :** | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Problem definition and solution identification | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Develop physical work flow of existing system | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Identify the contents of the existing data store | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Develop the logical DFD of the existing system | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **DETAIL ANALYSIS & DESIGN :** | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Develop the context diagram | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Develop the logical DFD of the new system | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Identify the data dictionary | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Develop the database design | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Develop the structure chart | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Develop the physical DFD of the new system | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Design the screen & report | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **IMPLEMENTATION :** | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Programming | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Data conversion | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Testing | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| - Documentation | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

# II. ANALYSIS OF THE EXISTING SYSTEM

## 2.1 Background of The Organization.

The Grand Total Co.,Ltd. distributes product by Direct Sale. The products are imported from foreign. There are many kinds of products for examples : vacuum cleaner, kitchen utensis, medicine and so on.

The company grows up fast and has many problems resulting from manual operations such as process delay. The Sale Management System is the proposed computerized system particularly designd to develop existing organization.

The Sale Management System is composed of master system, inventory system, purchase order system and sale system to develop the existing system.

## 2.2 Existing Business Function.

The function areas are within sale and marketing department, inventory department, and purchasing department. All of the customer orders procedures are manually operated.

2.2.1 Sale and Marketing Department

a. To keep the details of sale.

b. To generate reports of the sale.

2.2.2 Inventory Department

a. To keep the record paying and receiving of stock card.

b. To keep sufficient merchandise, check and report to purchasing department.

2.2.3 Purchasing Department

a. To order merchandise at the right time with the right volumn.

b. To follow the pending purchasing order.

## 2.3 Problem Definition.

2.3.1 Lack of information for planning and manual stock card is late.

2.3.2 Reports can not support for controlling, forecasting and decision making.

2.3.3 Lost customer and oppotunity sale.

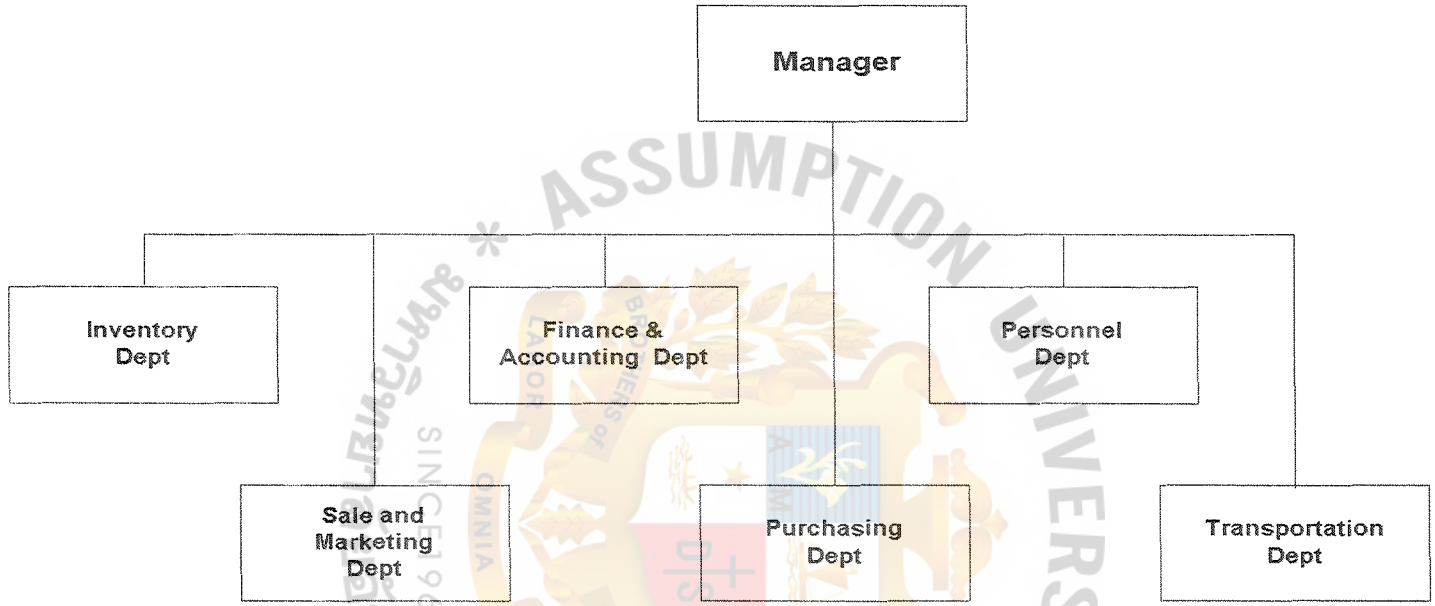


Figure 2.1. Oragniaztion Chart of Grand Total Co., Ltd.

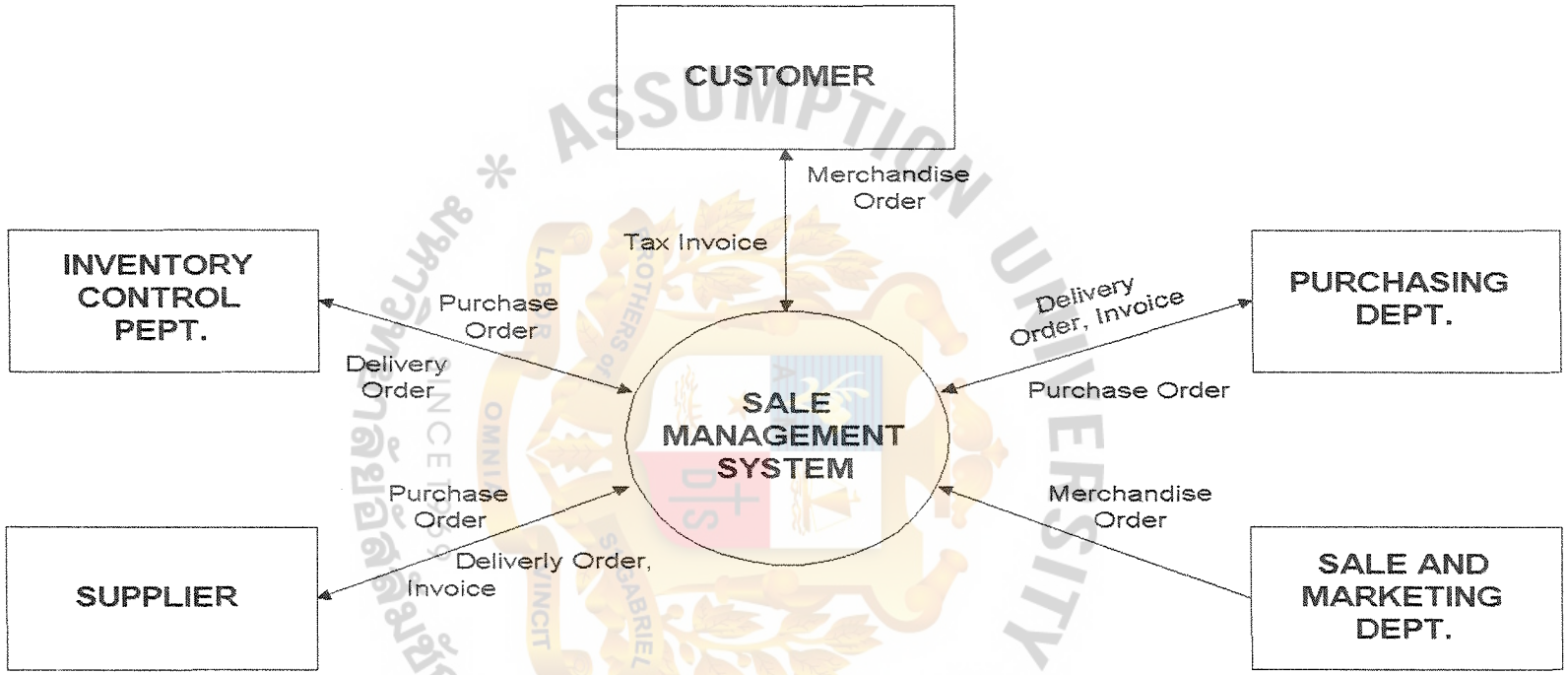


**Figure 2.2. Context Diagram of Existing System**

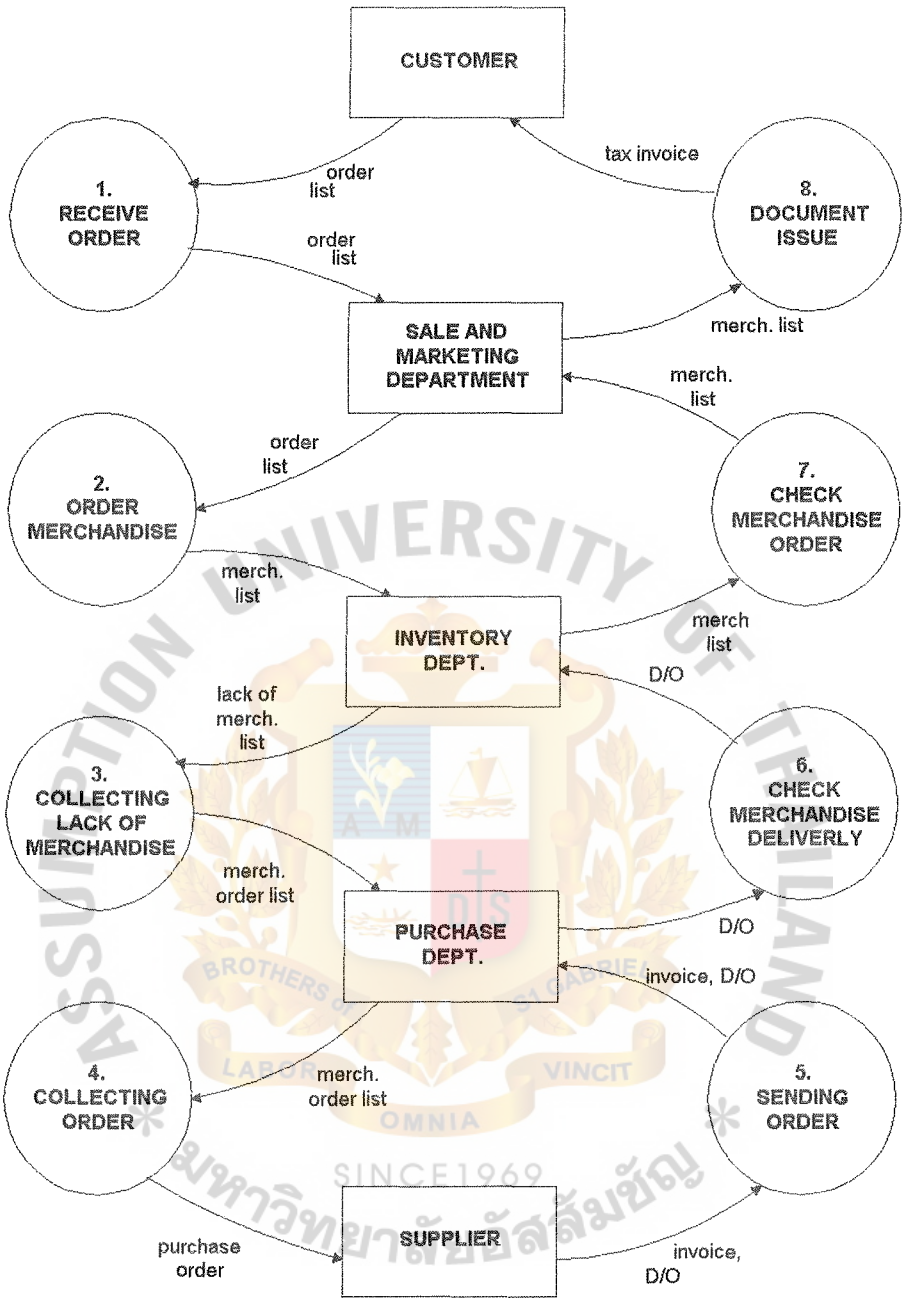


**Figure 2.3. Data Flow Diagram ( level 0 ) of Existing System**

# III. PROPOSED SYSTEM

## 3.1 User Requirement.

To meet the objectives of the system, user must be defined as follows : -

3.1.1 Input Requirement :

The inputs are needed as database to generate the output requirements.

3.1.2 Output Requirement :

The outputs are the reports that can be devided into two types. One is daily report, and the other is monthly reports

**Table 3.1. Reports ' Frequence**

| REPORT | FREQUENEY |
|---|---|
| **MASTER FILE SYSTEM** | |
| 1. Type of Merchandise | On Demand |
| 2. Group of Merchandise | On Demand |
| 3. Structure of Merchandise | On Demand |
| 4. Status of Merchandise | On Demand |
| 5. Merchandise Detail | On Demand |
| 6. Customer History | On Demand |
| 7. List of Bank Name | On Demand |
| 8. Vendor History | On Demand |
| 9. Condition Payment | On Demand |
| **INVENTORY SYSTEM** | |
| 1. Merchandise remain | Monthly |
| 2. Stock card of merchandise | On Demand |
| 3. Merchandise unmovement | Monthly |
| 4. Merchandise under limited | Monthly |
| **PURCHASE ORDER SYSTEM** | |
| 1. Purchasing Order | Daily |
| 2. Merchandise Receive | Daily |
| 3. Merchandise Return | Daily |
| 4. Pending Order | Daily |
| 5. Price List | Daily |
| **SALE SYSTEM** | |
| 1. Sale request | Daily |
| 2. Tax Invoice | Daily |

## 3.2 SYSTEM DESIGN

The followings describe aspects of system design :

1. Software Design : Context Diagram & Data Flow Diagram
2. Screen Design : Input Screen
3. Report Design : Output Screen

### 3.2.1 Context Diagram

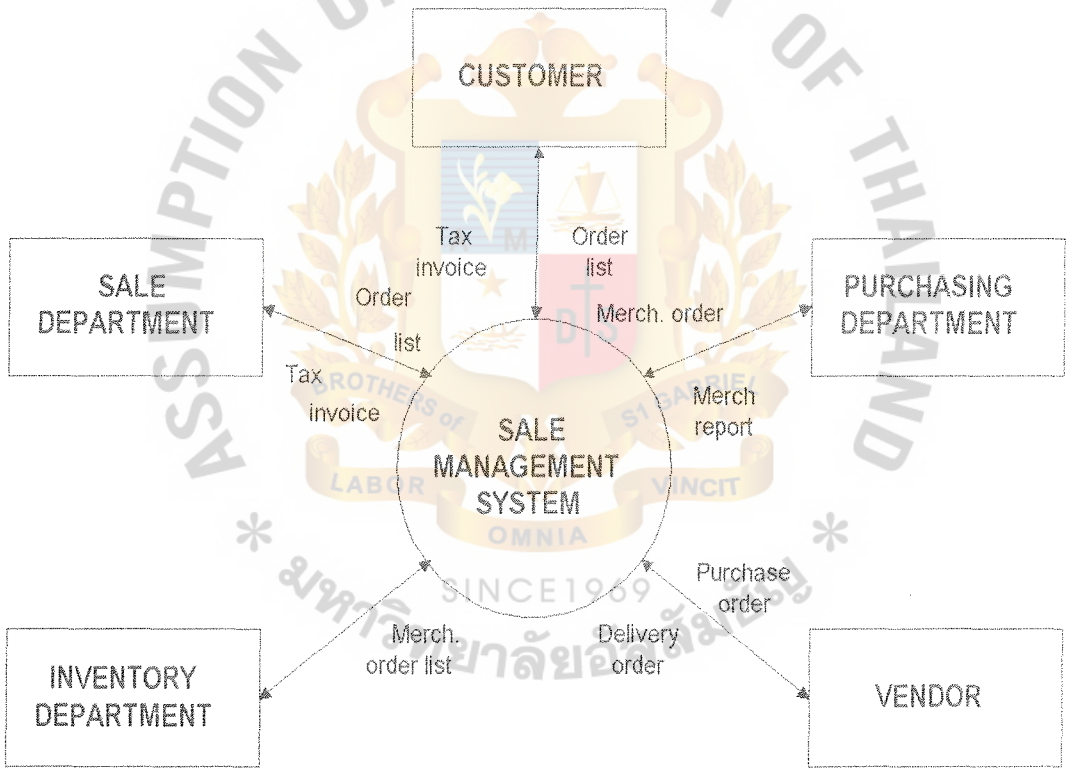


**Figure 3.1. Context Diagram of Proposed System**

### 3.2.2 Data Flow Diagram

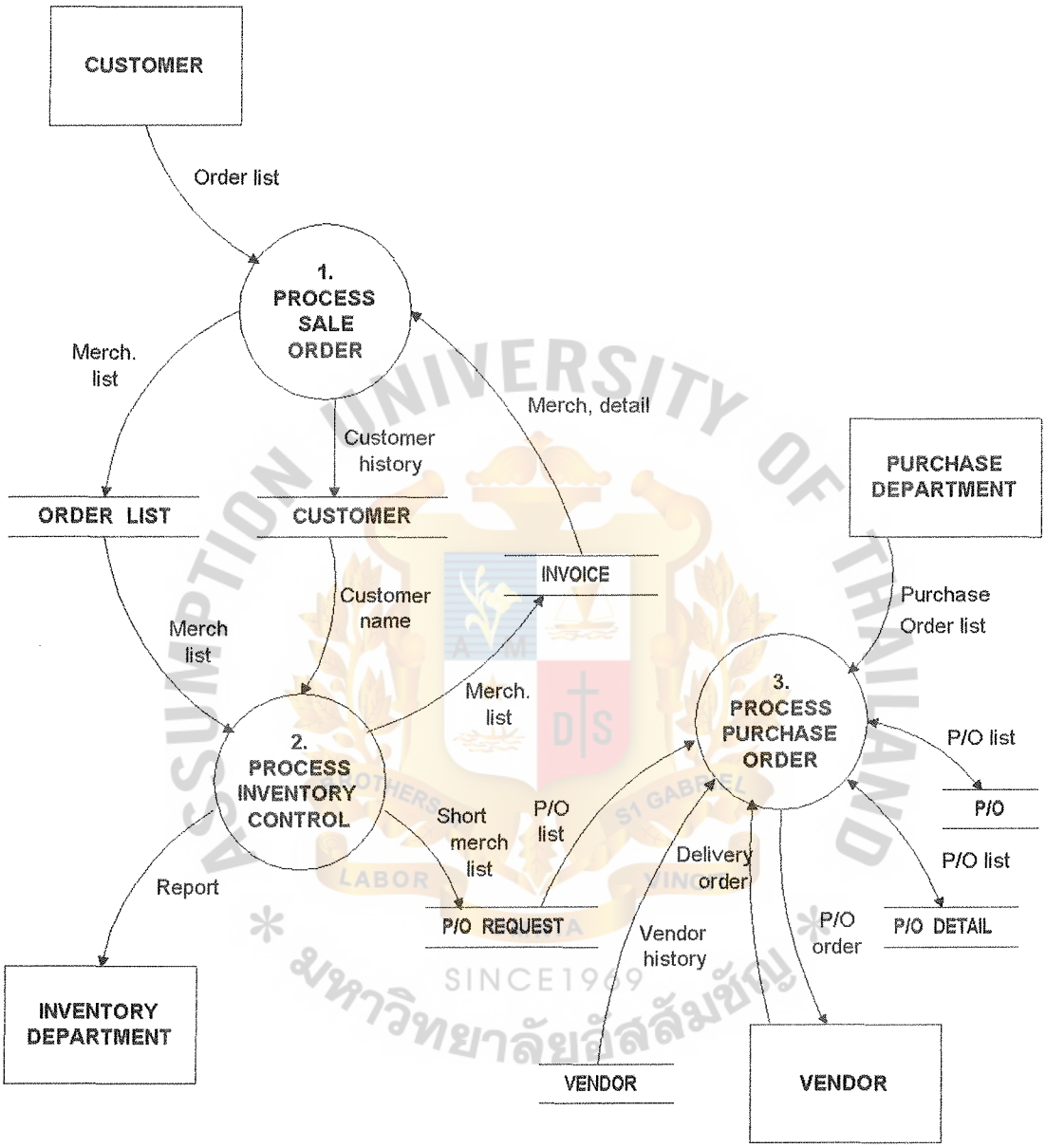


Figure 3.2. Data Flow Diagram ( Level 0 ) - Proposed System

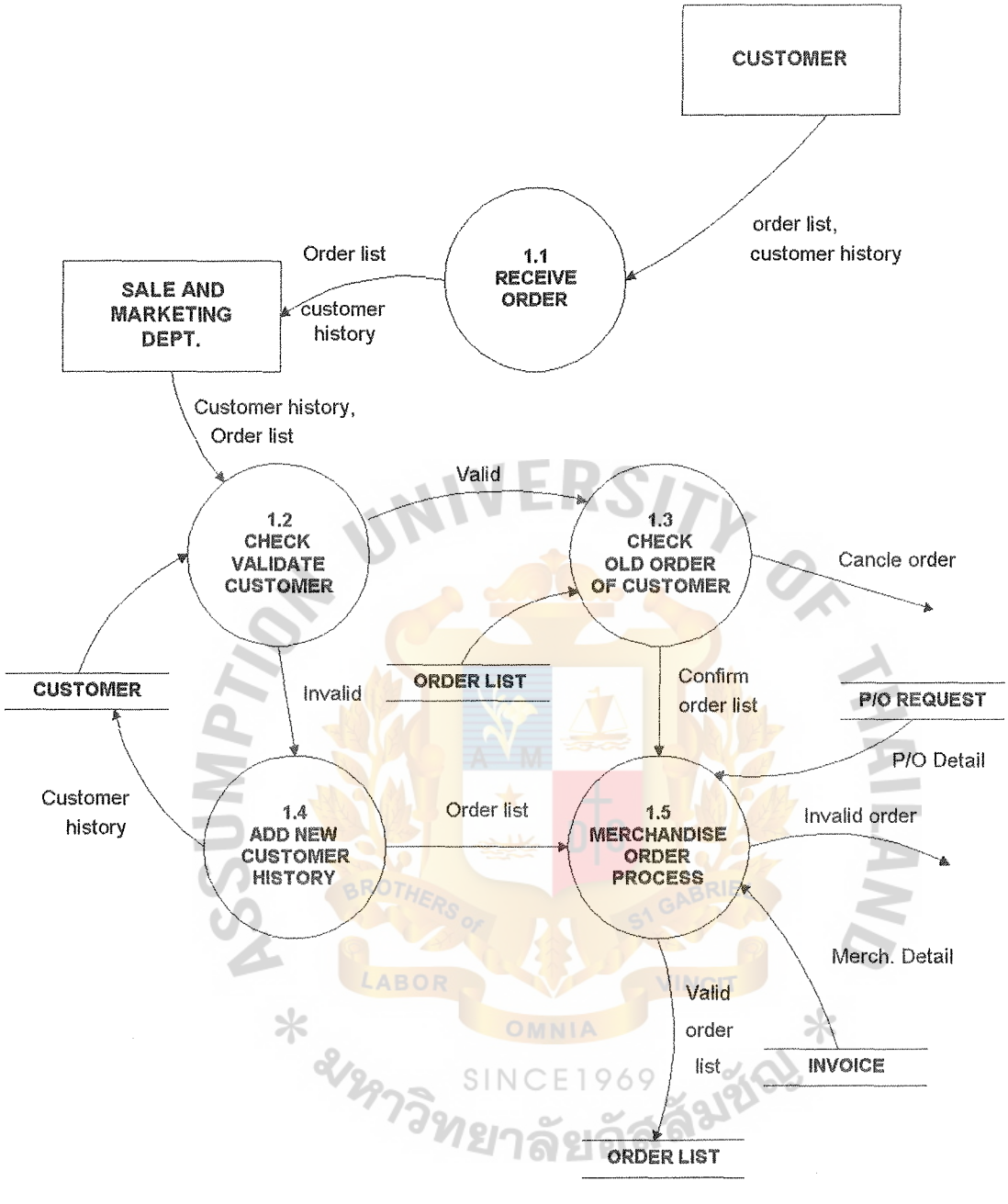


Figure 3.3. Process 1 Process Sale Order

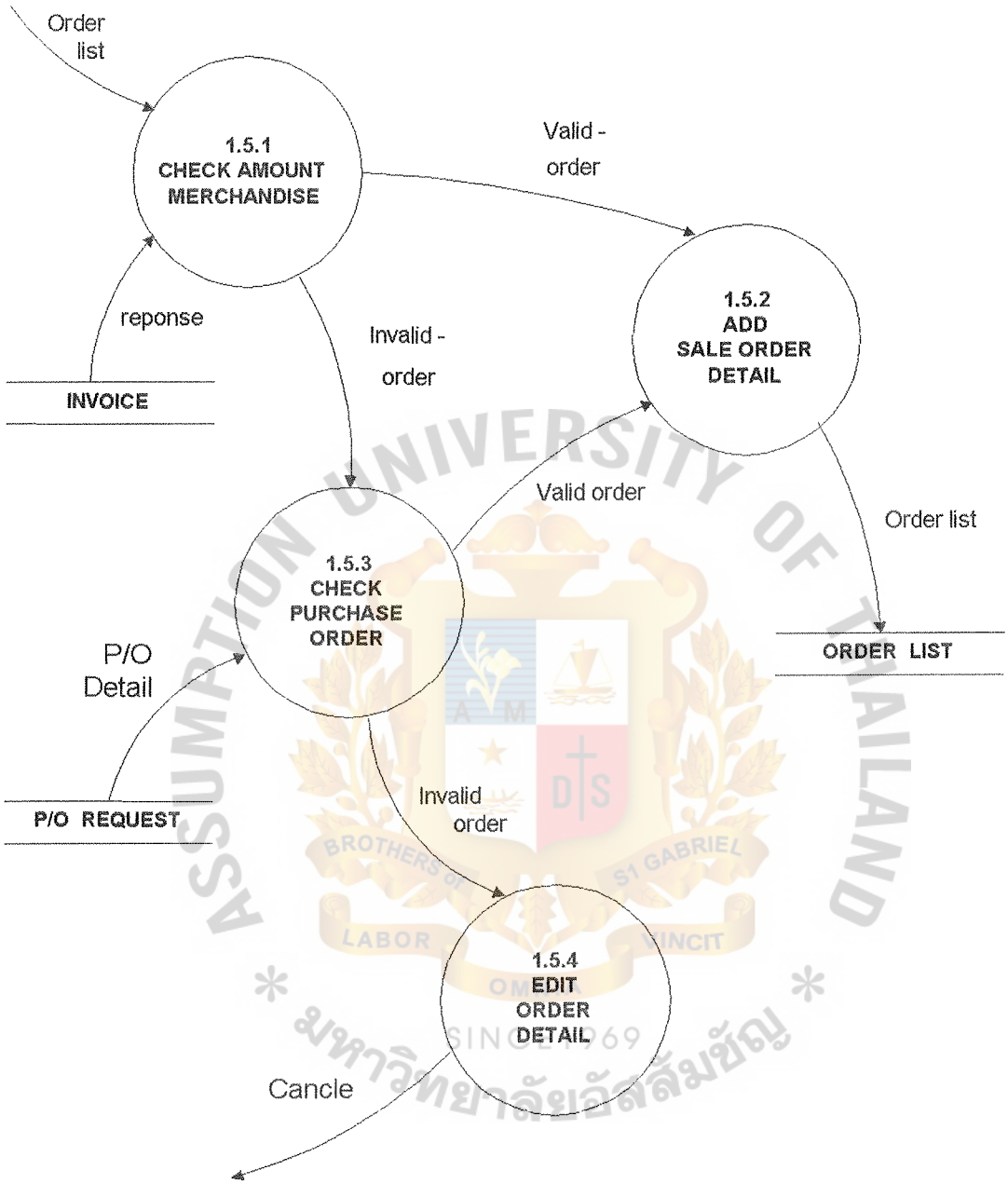


**Figure 3.4. Process 1.5 Merchandise Order Process**

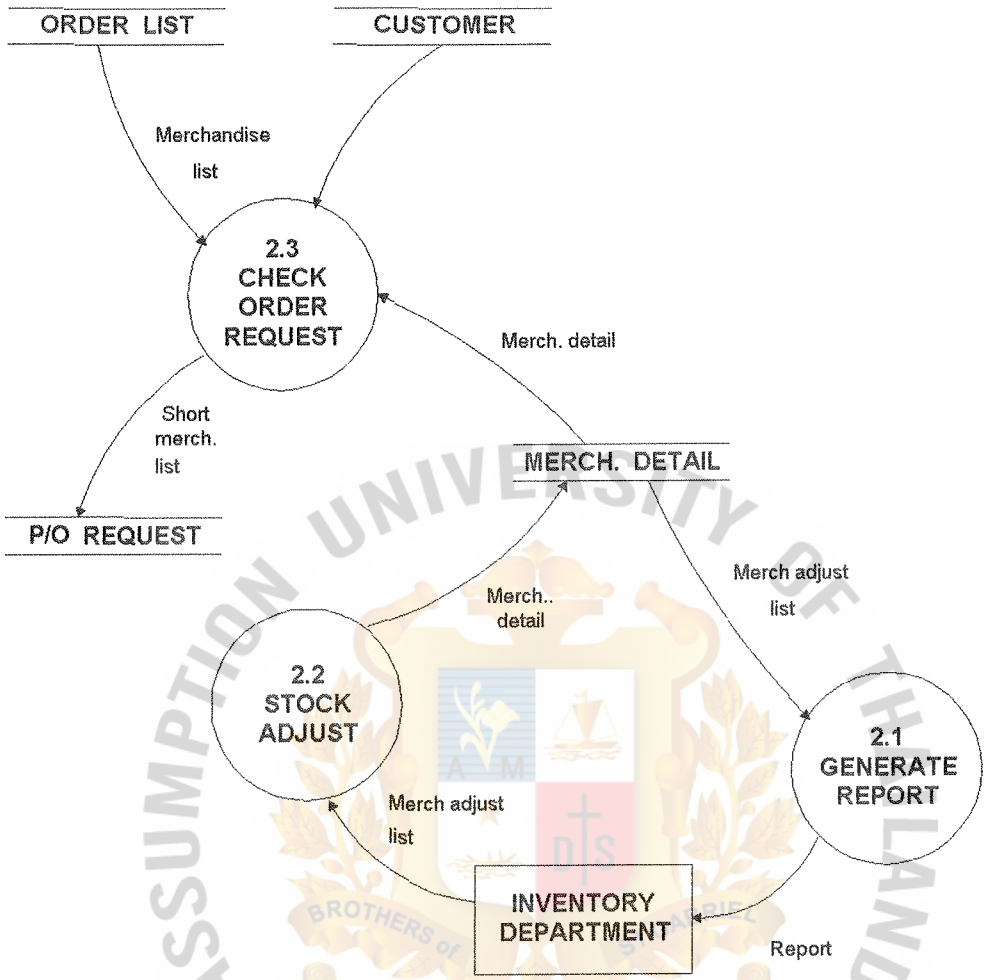


**Figure 3.5. Process 2 Process Inventory Control**

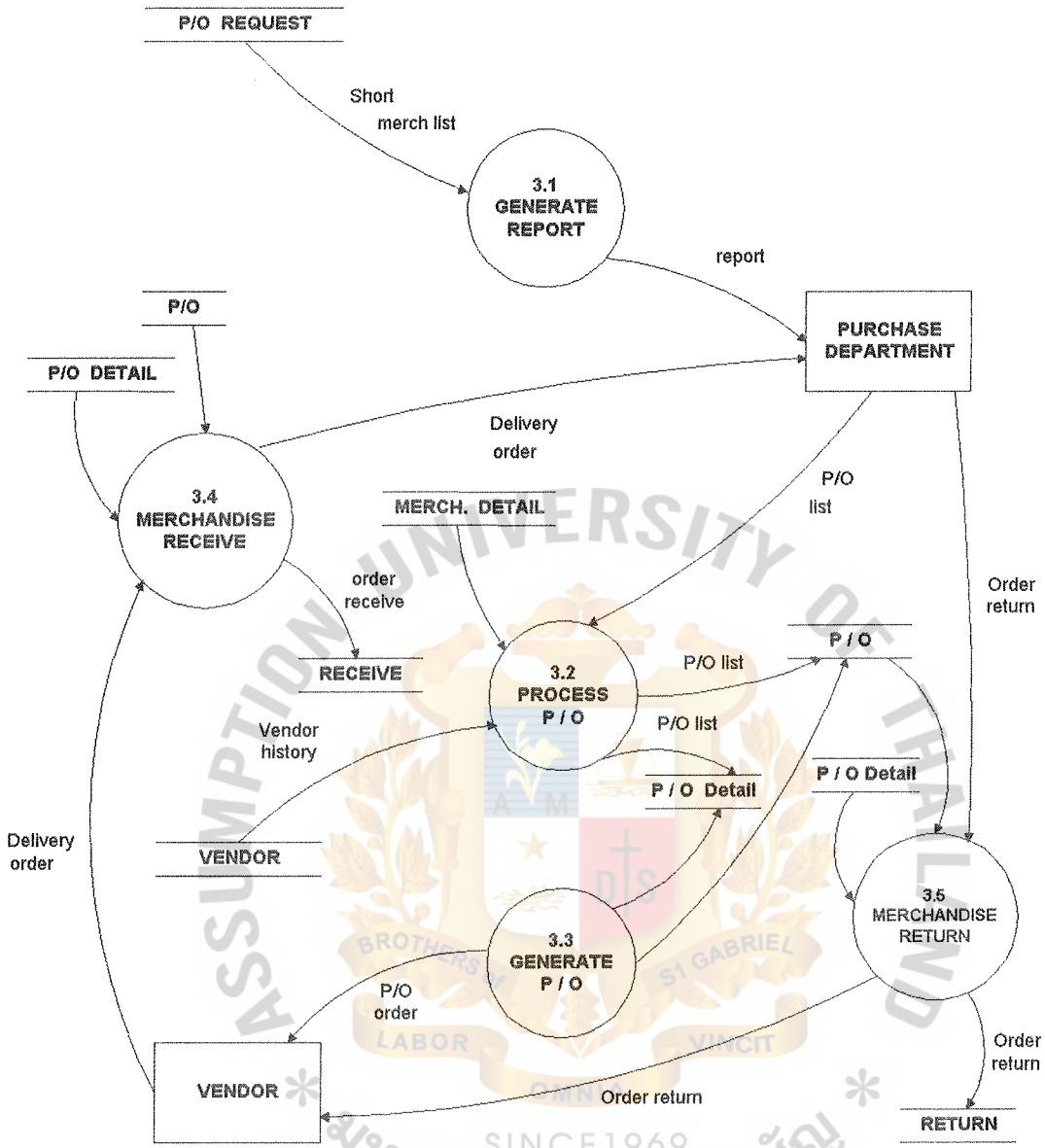


Figure 3.6. Process 3 Purchase Order Process

### 3.2.3 Structure Chart

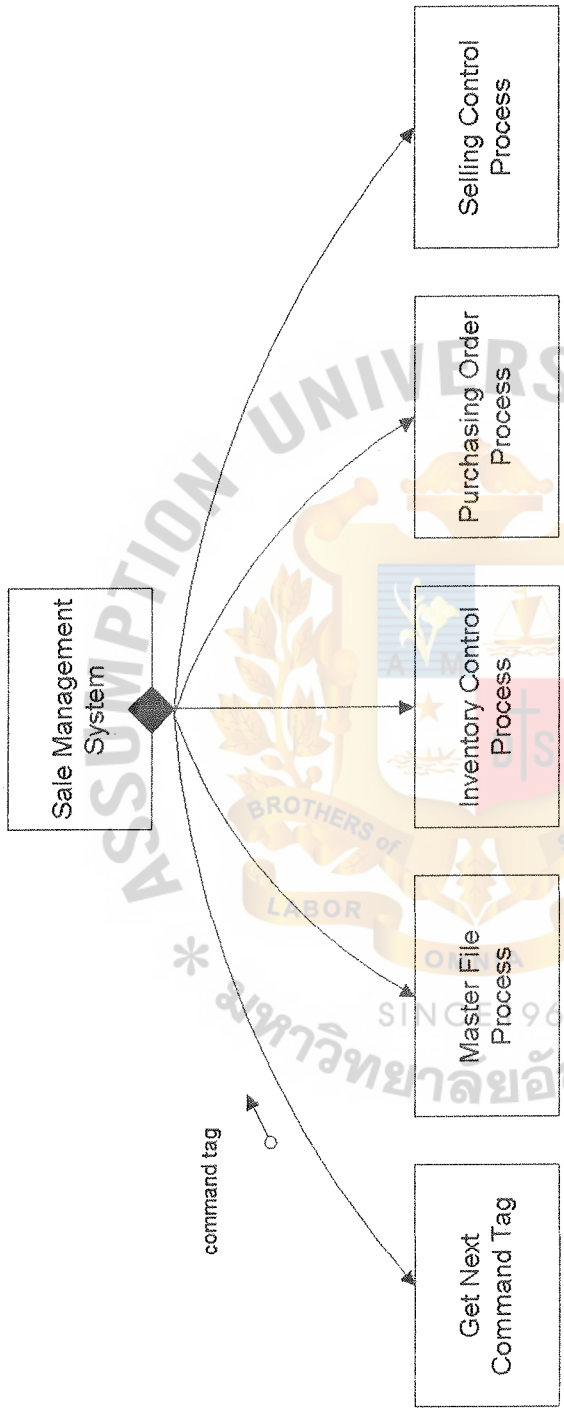


Figure 3.7. Structure Chart of Sale Management System

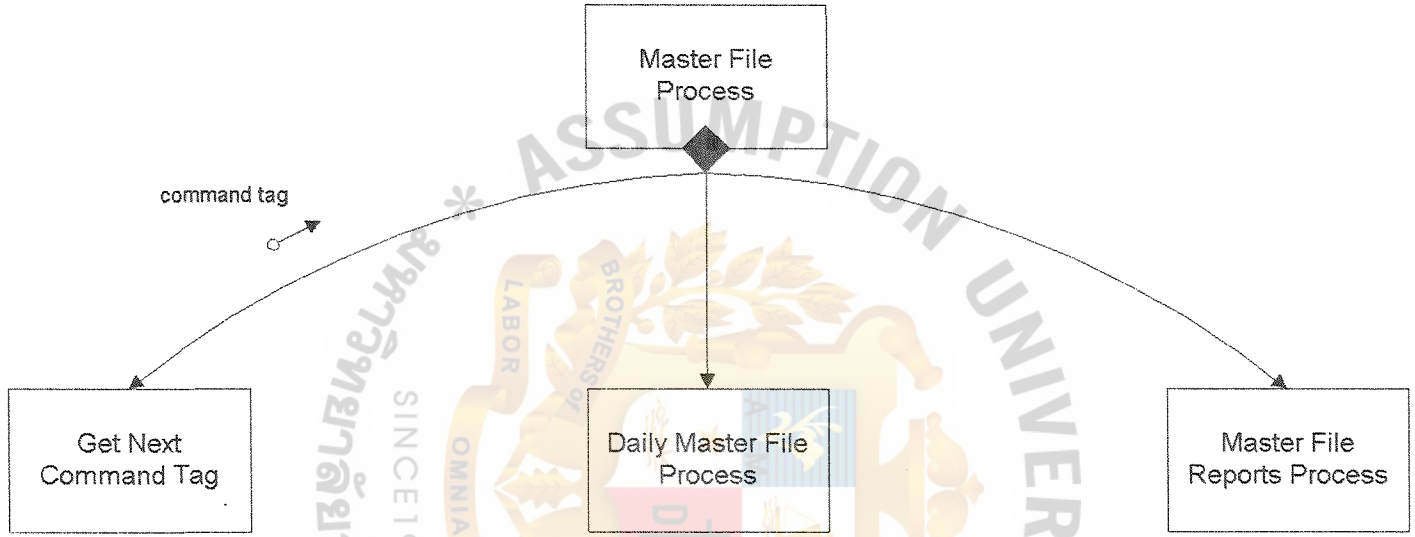


**Figure 3.8. Structure Chart of Master File Process**

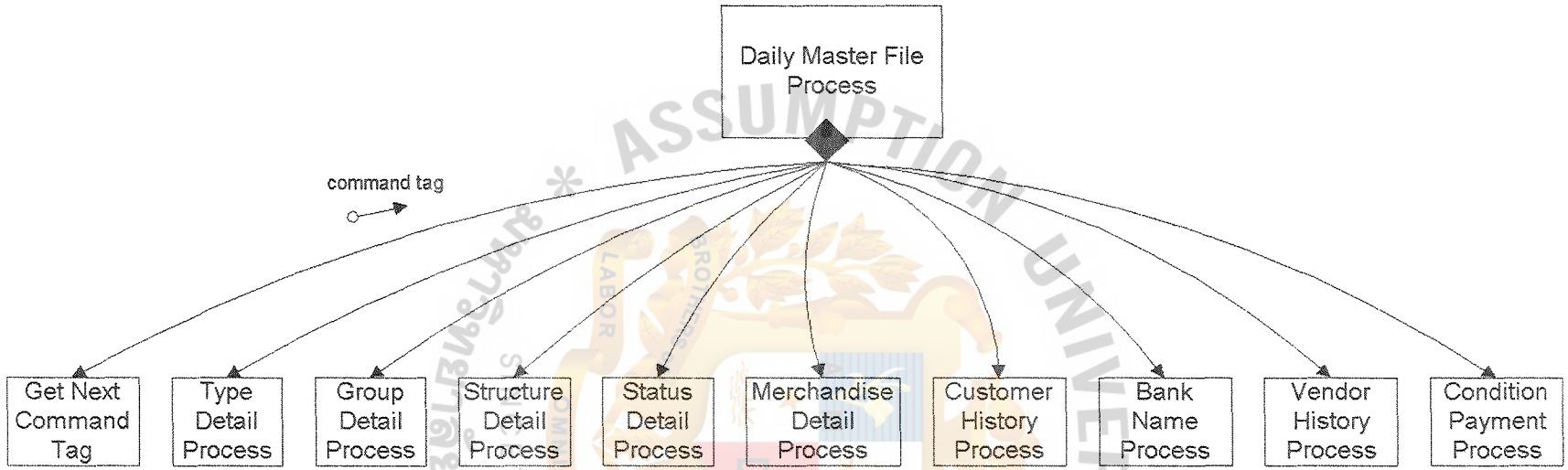


**Figure 3.9. Structure Chart of Daily Master File Process**

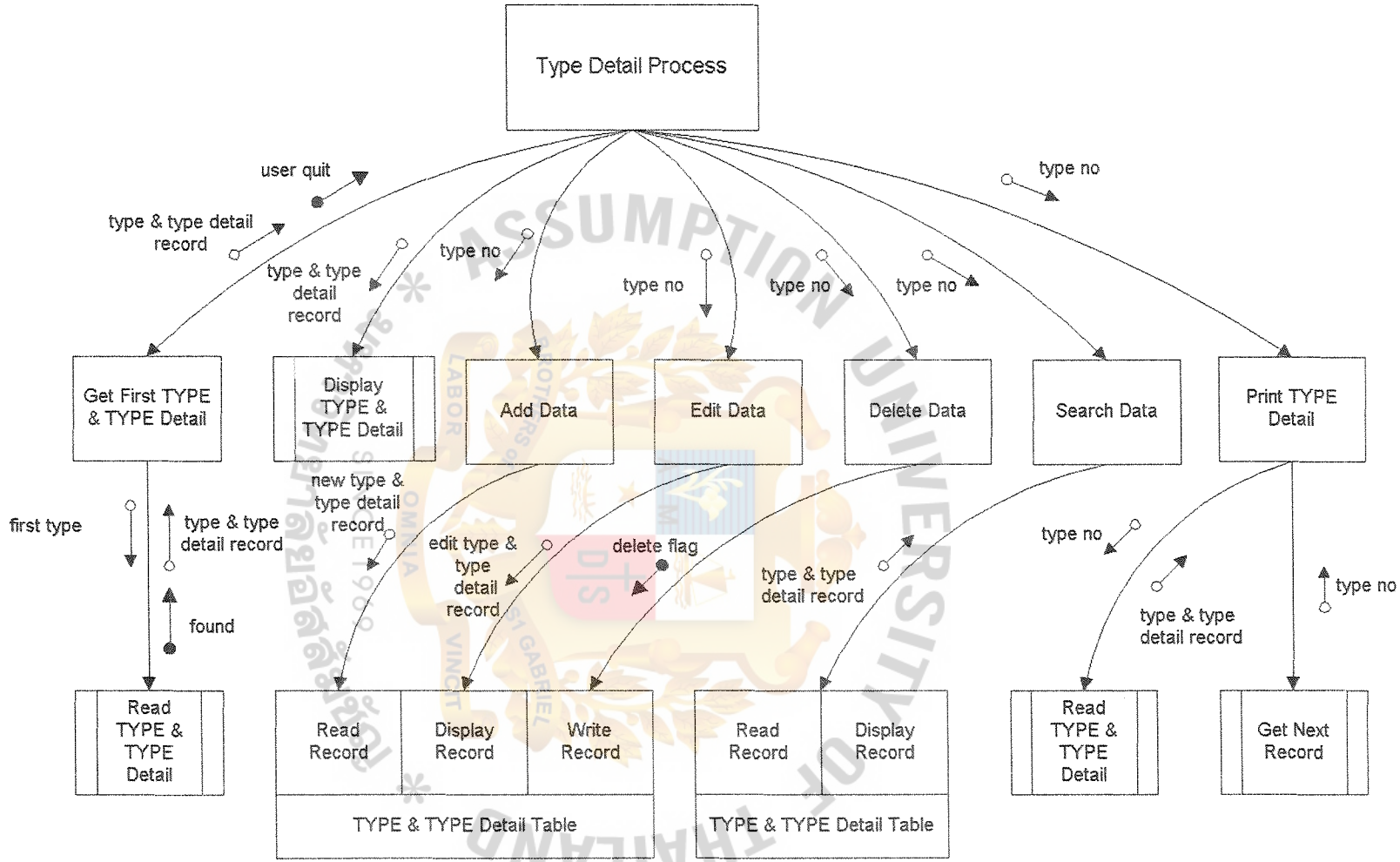


**Figure 3.10. Structure Chart of Type Detail Process**

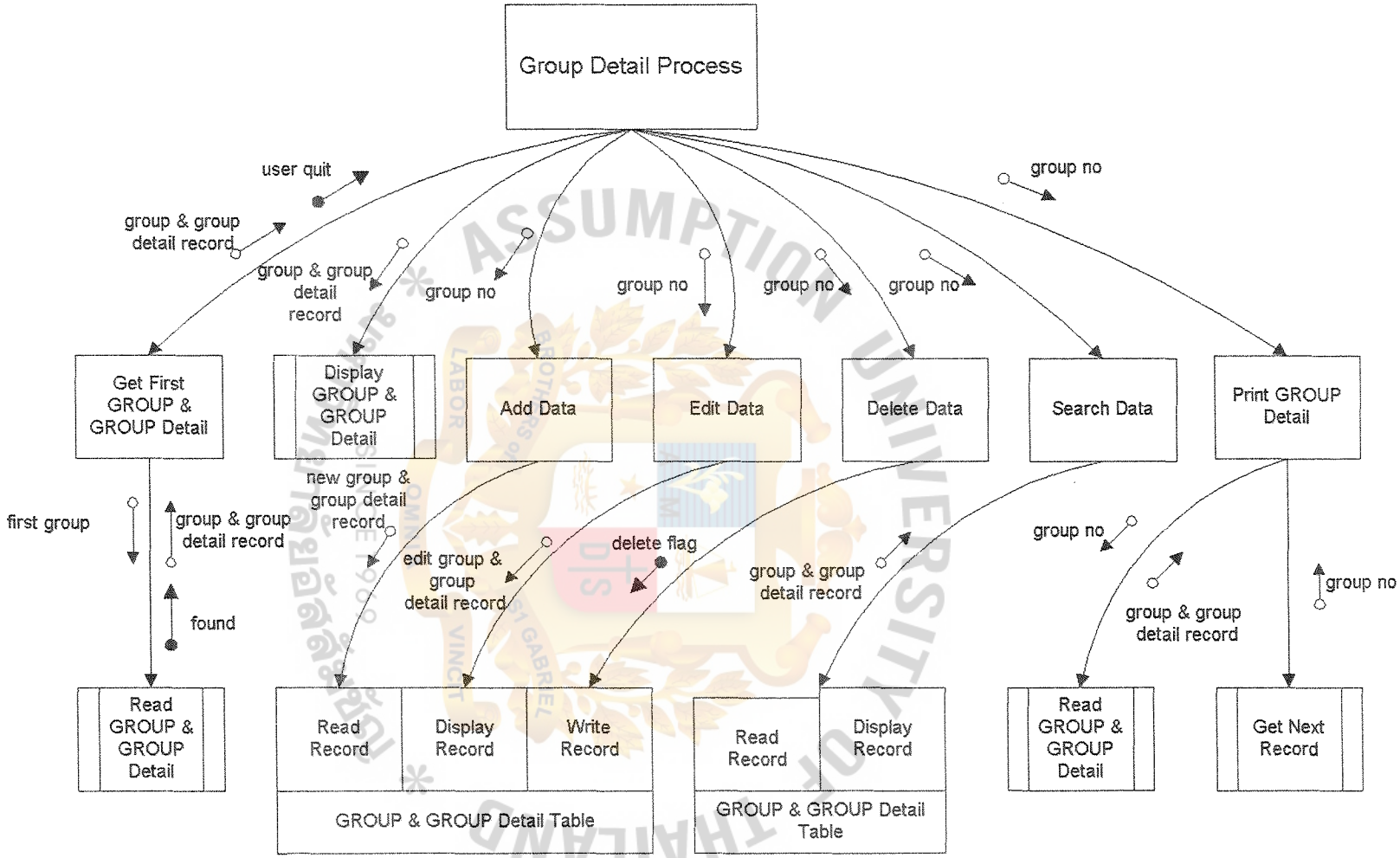


**Figure 3.11. Structure Chart of Group Detail Process**

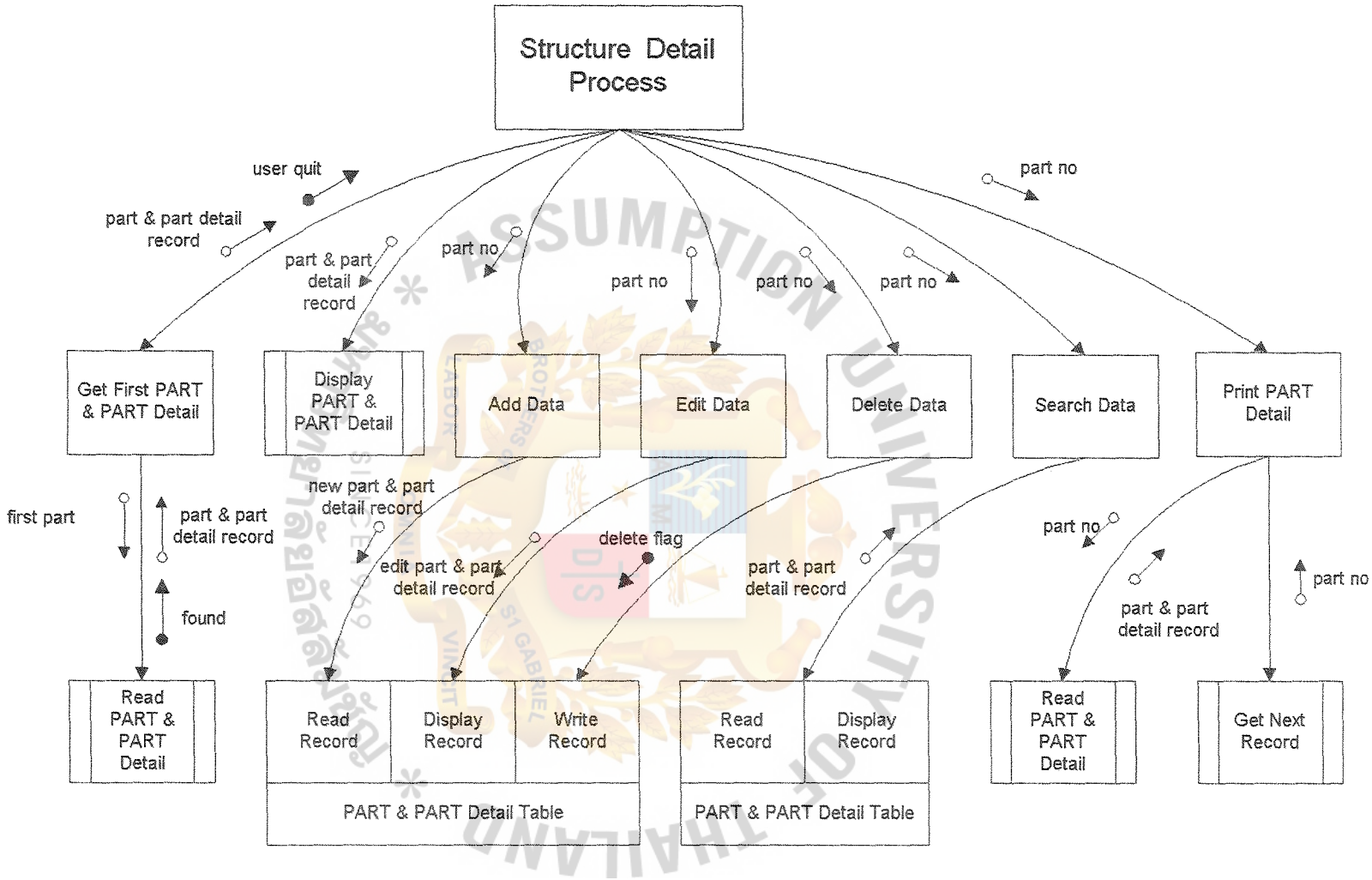


Figure 3.12. Structure Chart of Part Detail Process

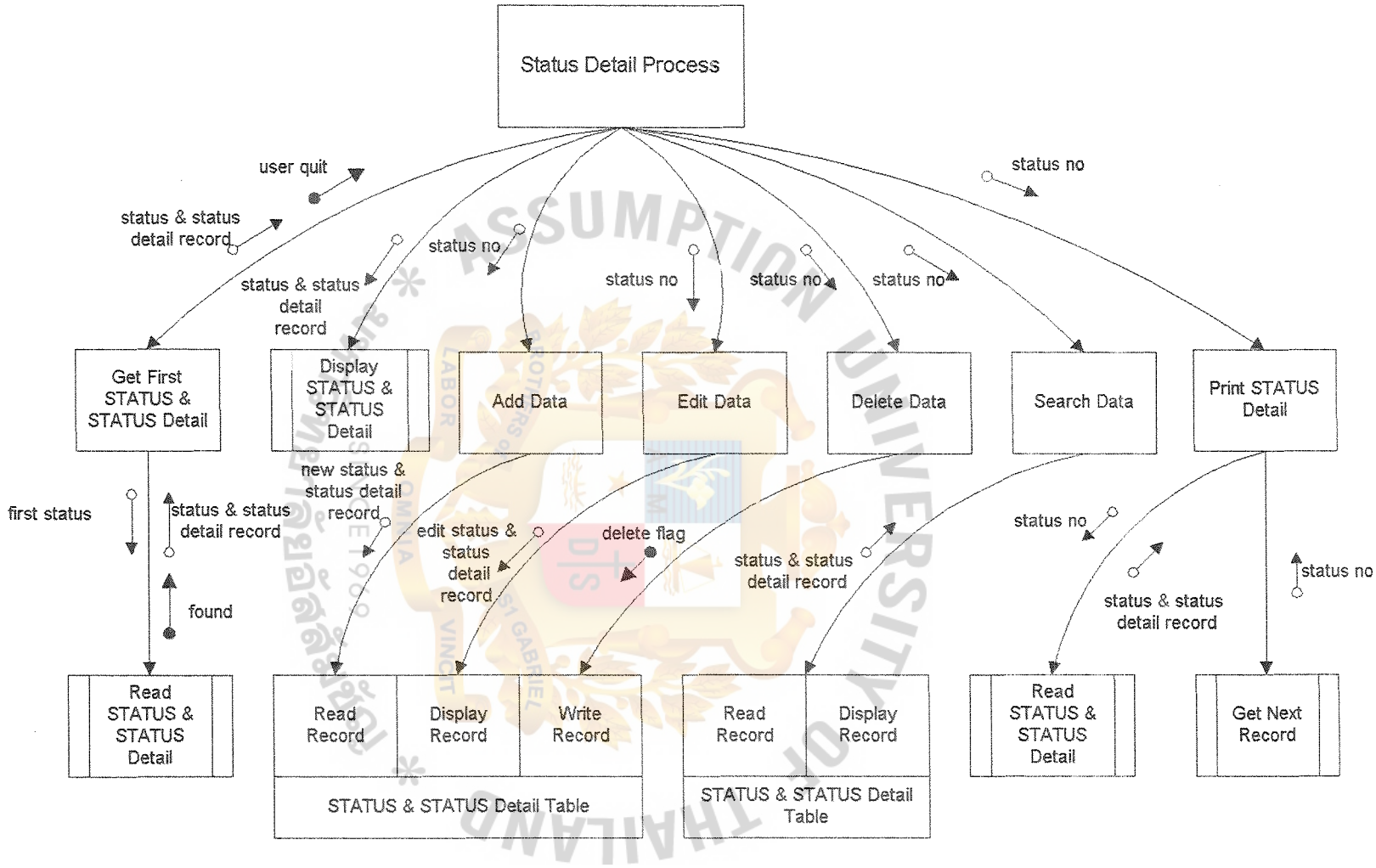


**Figure 3.13. Structure Chart of Status Detail Process**

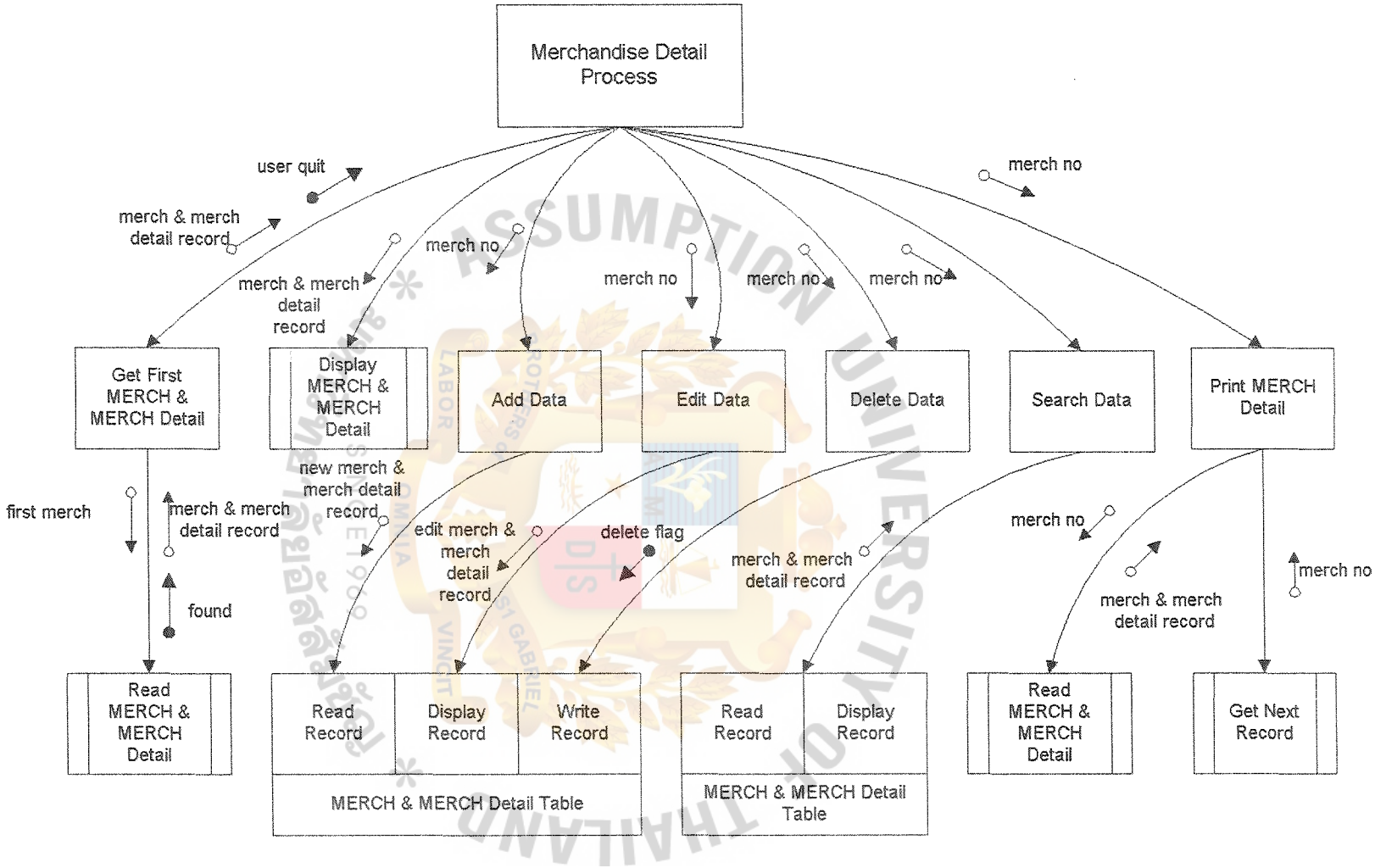


**Figure 3.14. Structure Chart of Merchandisc Detail Process**

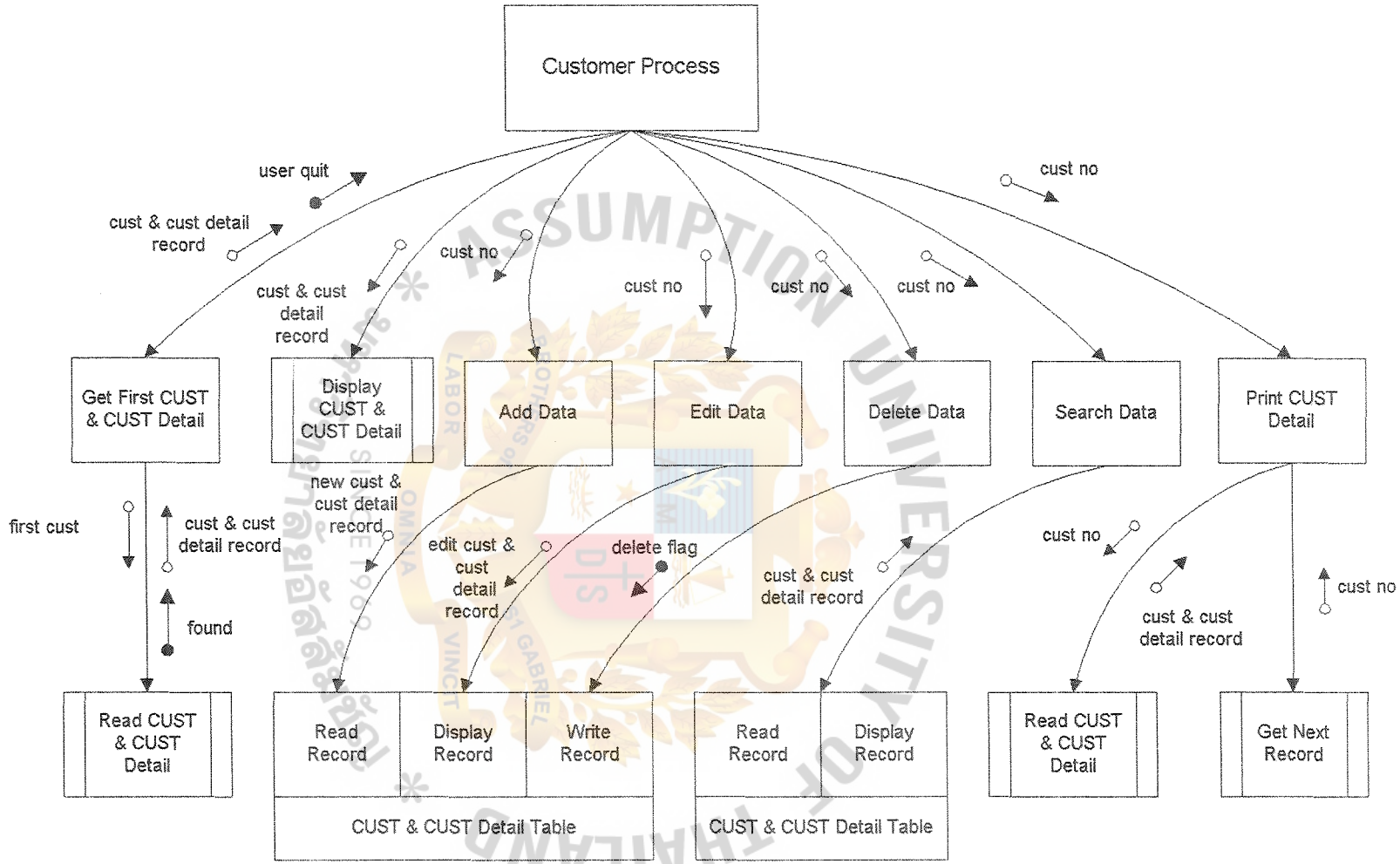


**Figure 3.15. Structure Chart of Customer Process**

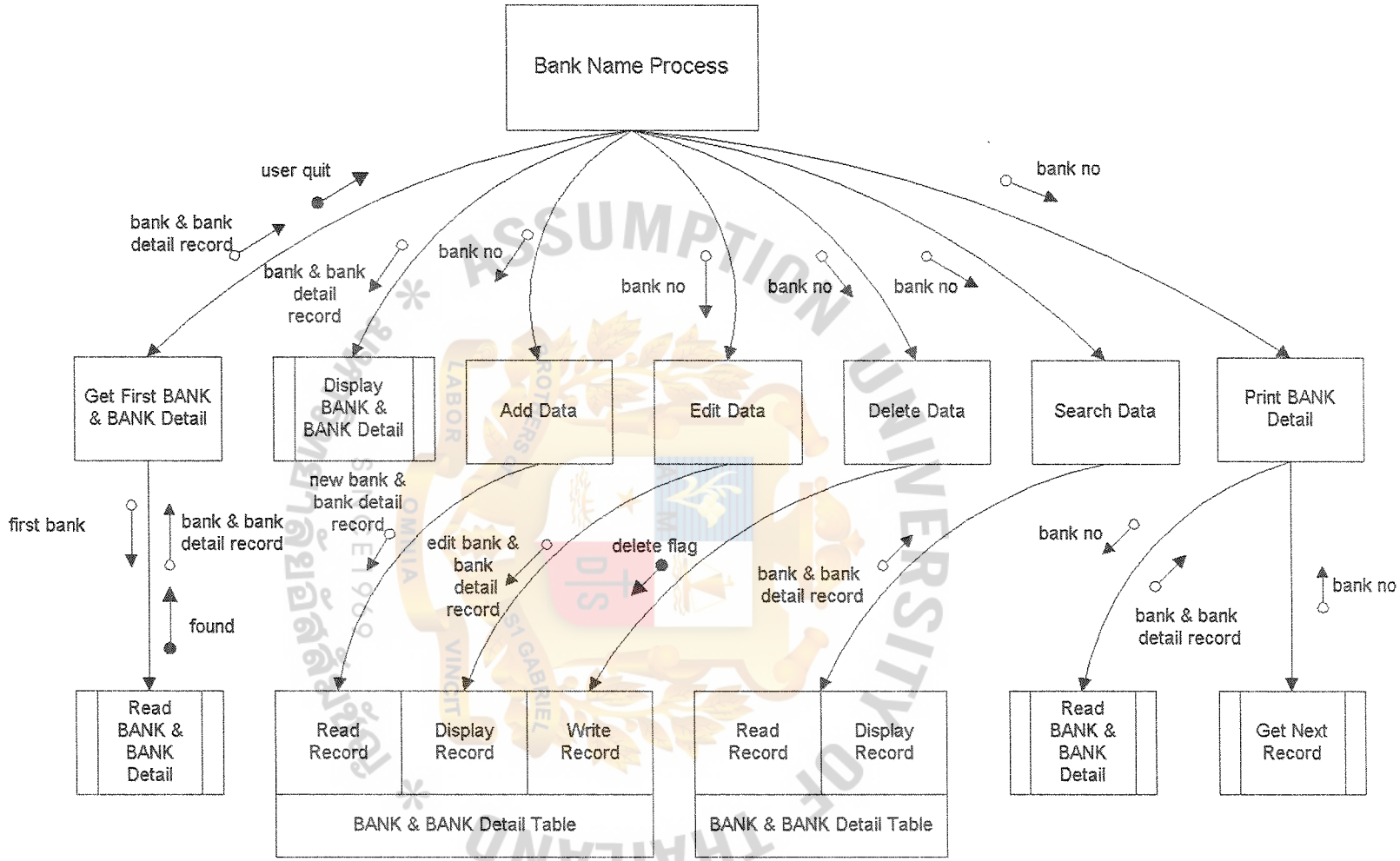


Figure 3.16. Structure Chart of Bank Name Process

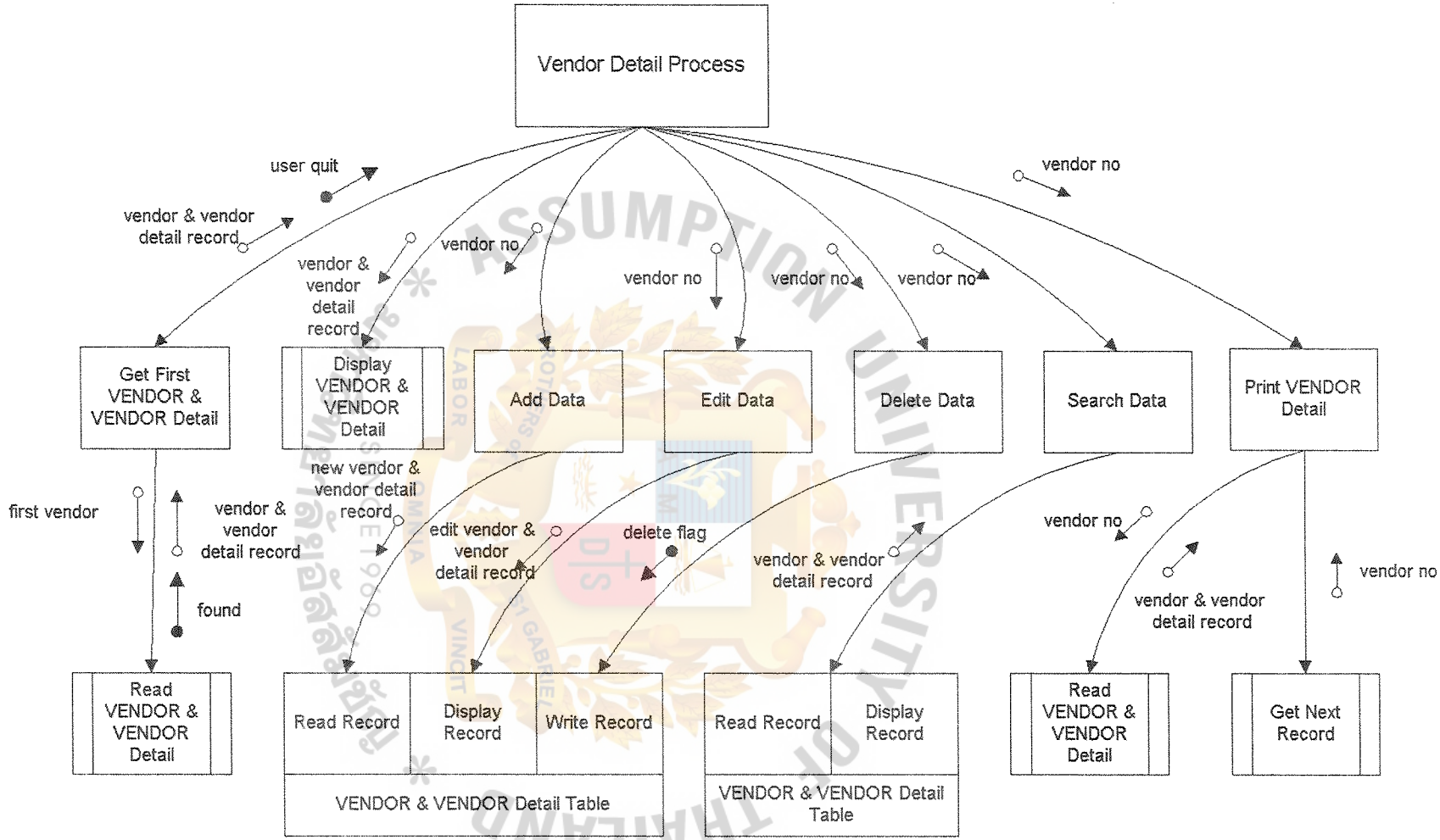


**Figure 3.17. Structure Chart of Vendor Detail Process**

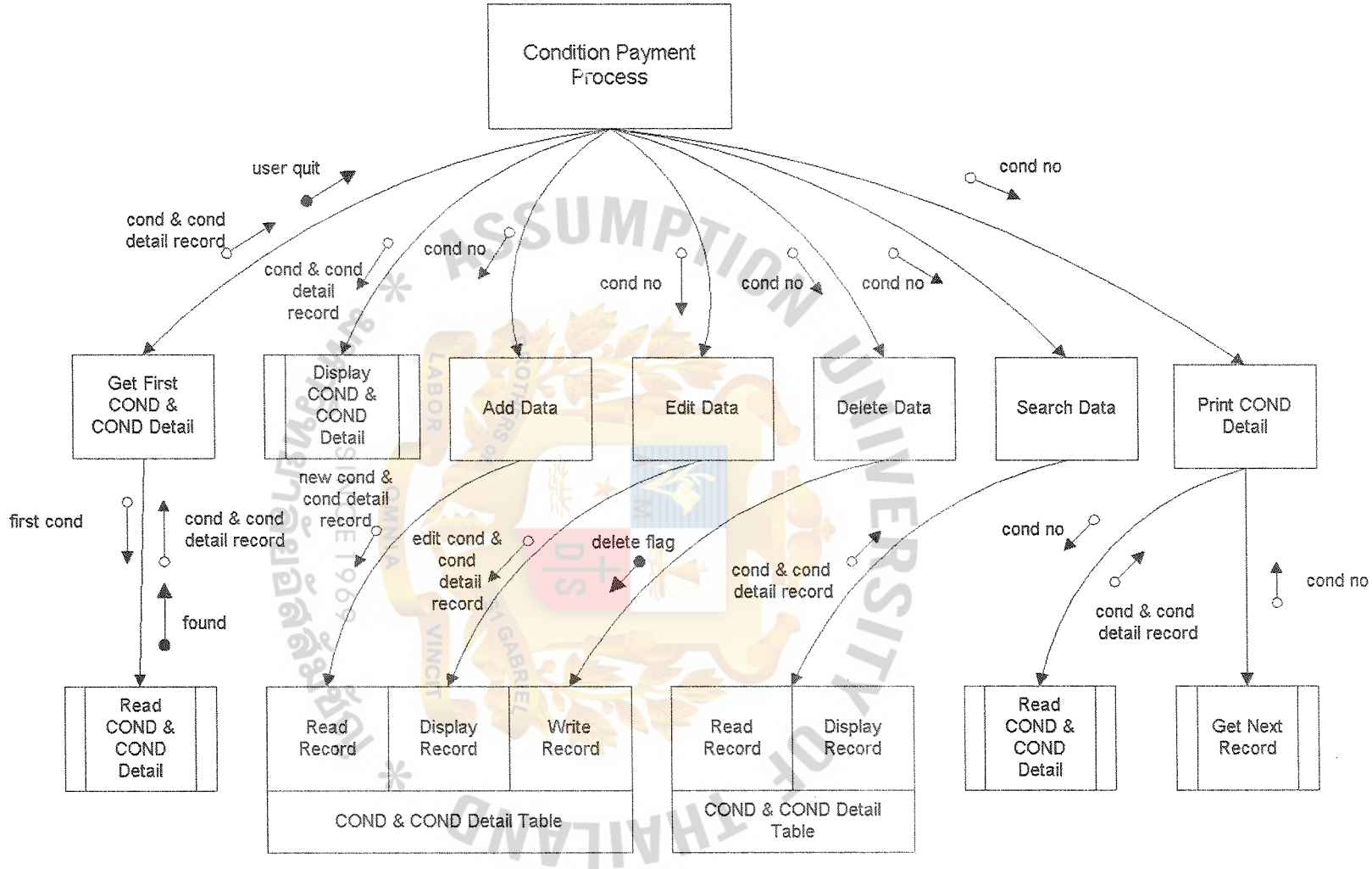


**Figure 3.18. Structure Chart of Condition Payment Process**

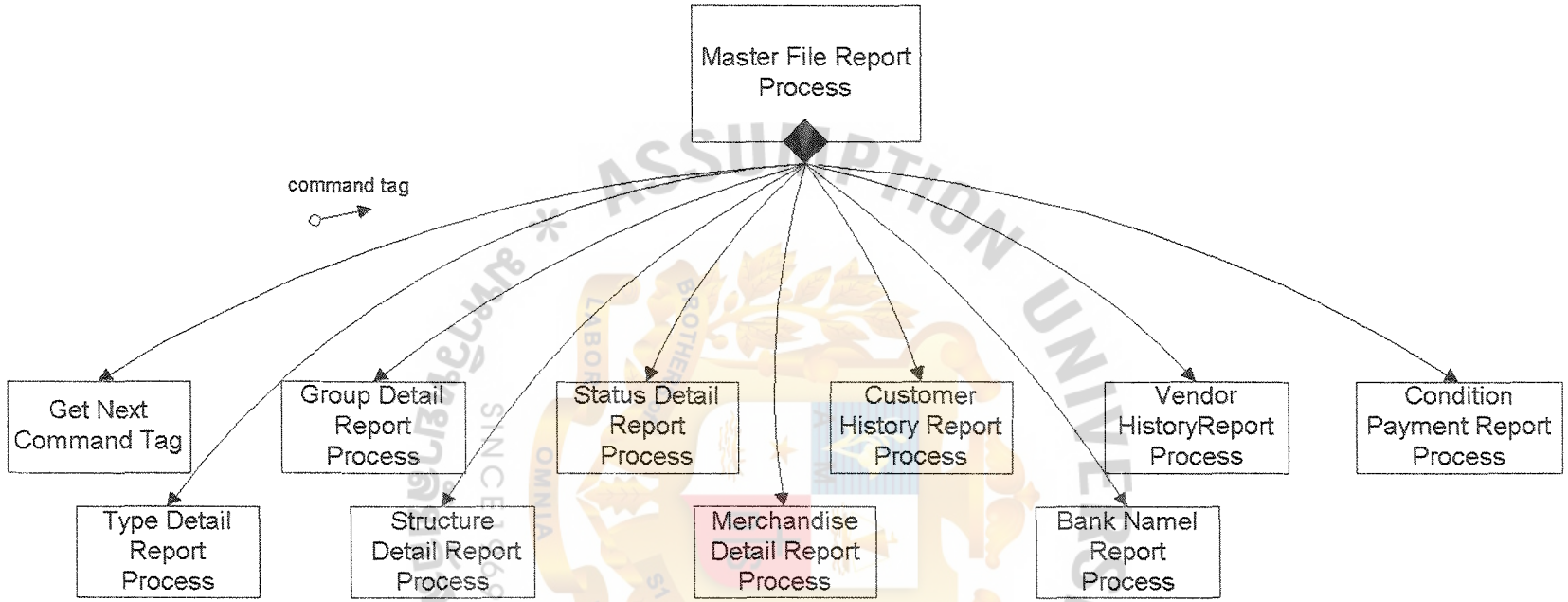


**Figure 3.19. Structure Chart of Master File Report Process**

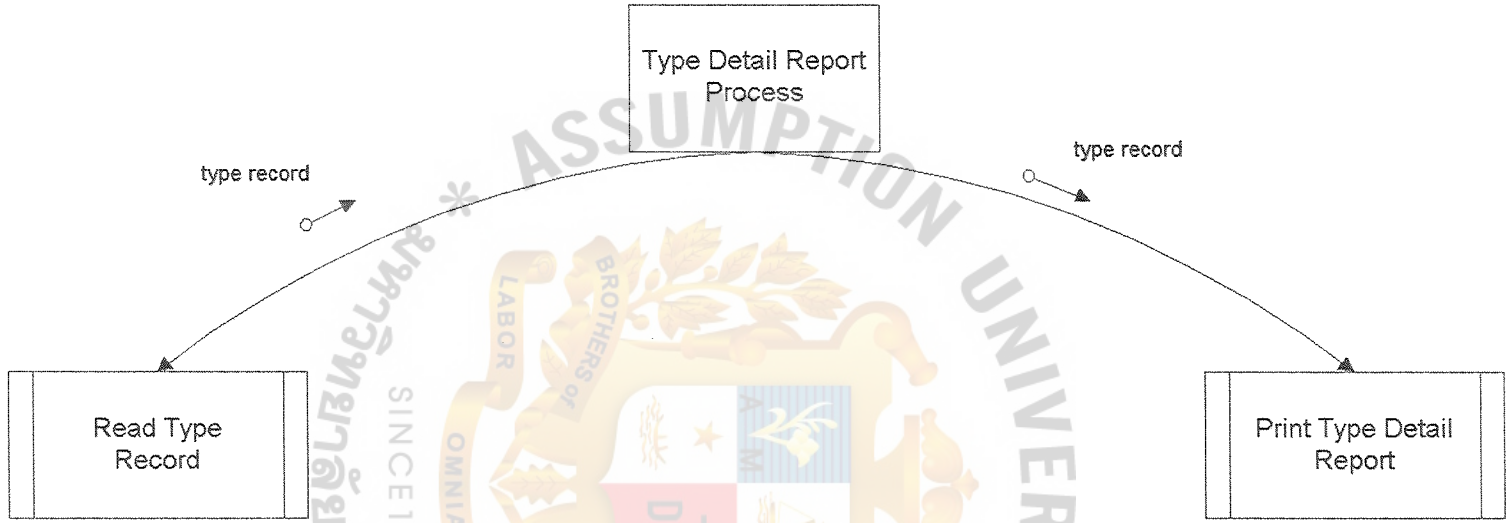


**Figure 3.20. Structure Chart of Type Detail Report Process**

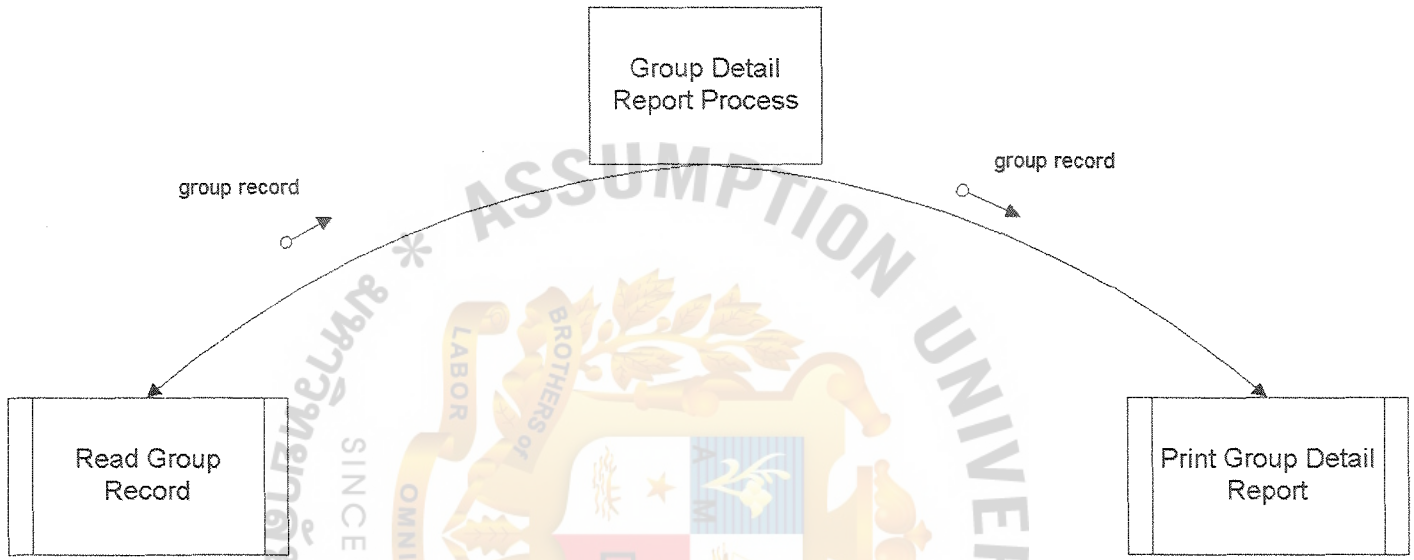


**Figure 3.21. Structure Chart of Group Detail Report Process**


**Figure 3.22. Structure Chart of Structure Detail Report Process**

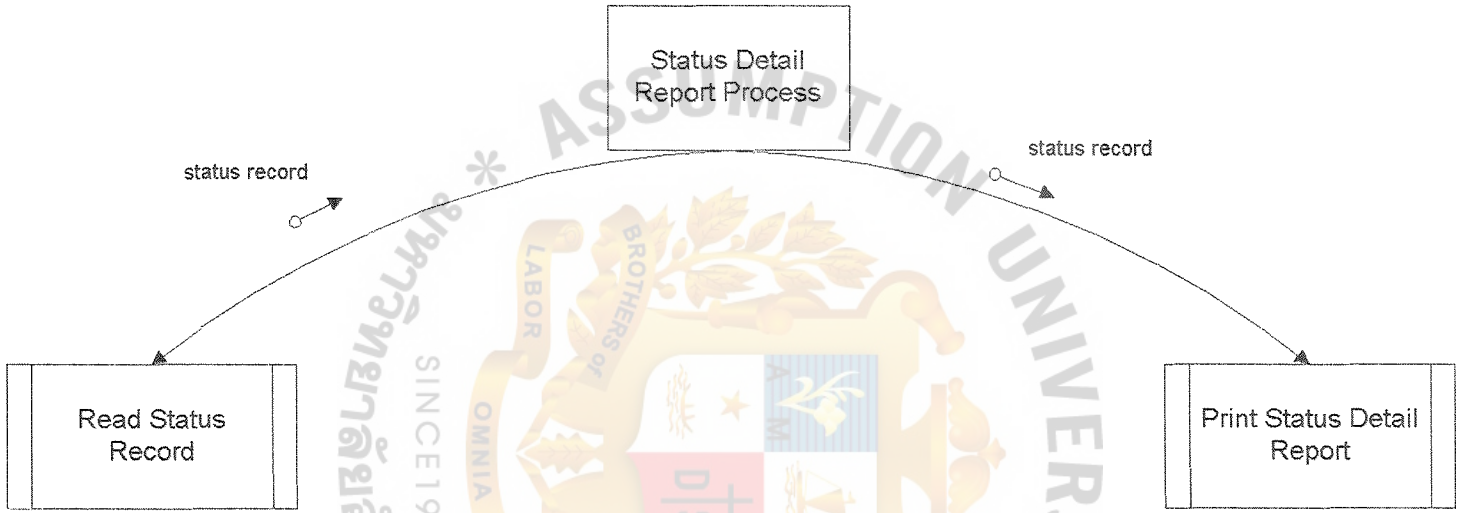


**Figure 3.23. Structure Chart of Status Detail Report Process**


**Figure 3.24. Structure Chart of Merchandise Detail Report Process**


**Figure 3.25. Structure Chart of Customer History Report Process**


**Figure 3.26. Structure Chart of Bank Name Report Process**


**Figure 3.27. Structure Chart of Vendor History Report Process**

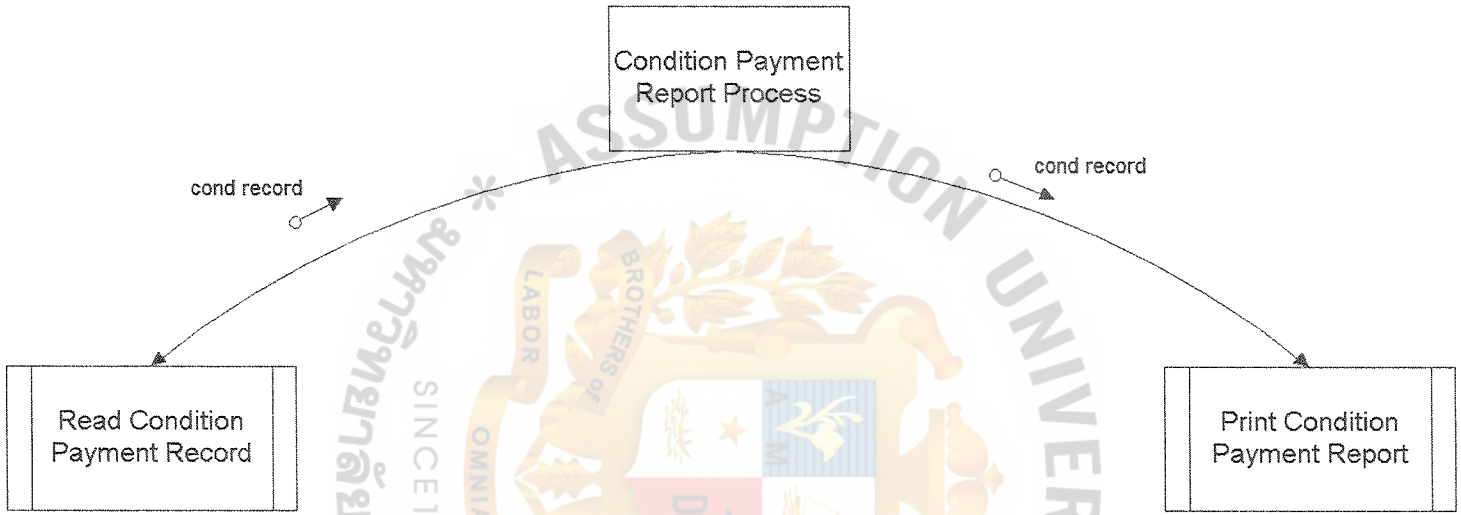


**Figure 3.28. Structure Chart of Condition Payment Report Process**

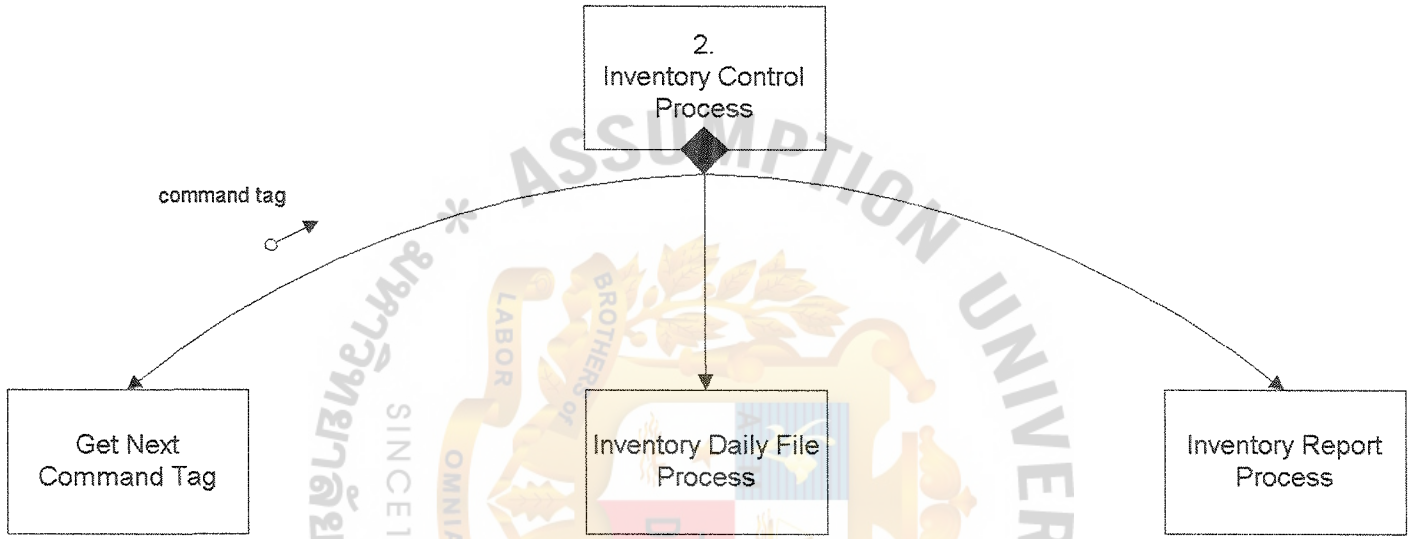


**Figure 3.29. Structure Chart of Inventory Control Process**

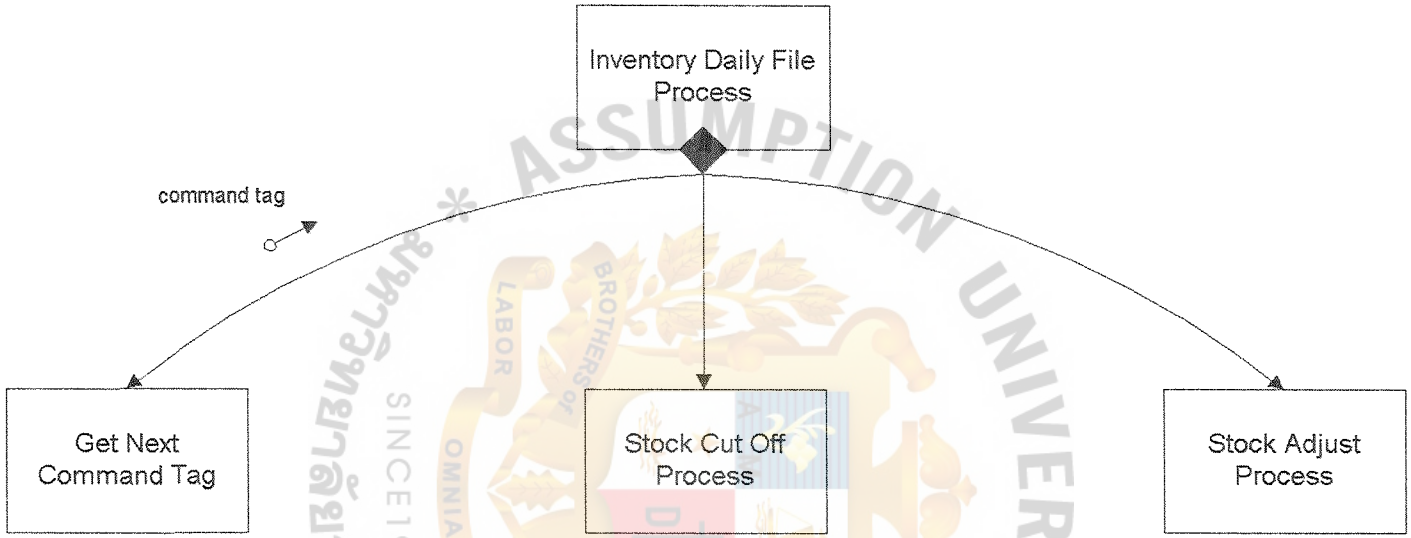


**Figure 3.30. Structure Chart of Inventory Daily File Process**

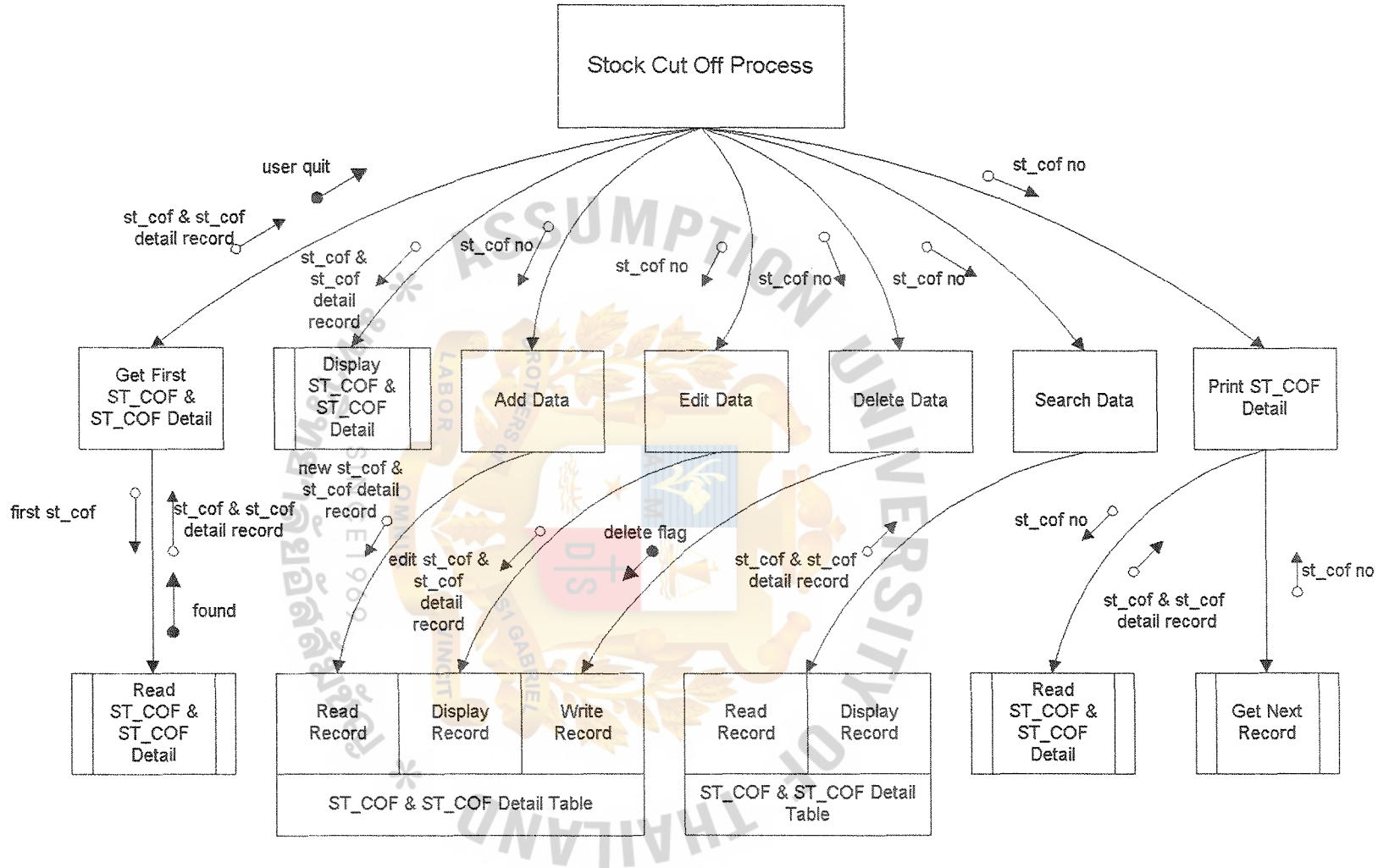


**Figure 3.31. Structure Chart of Stock Cut Off Process**

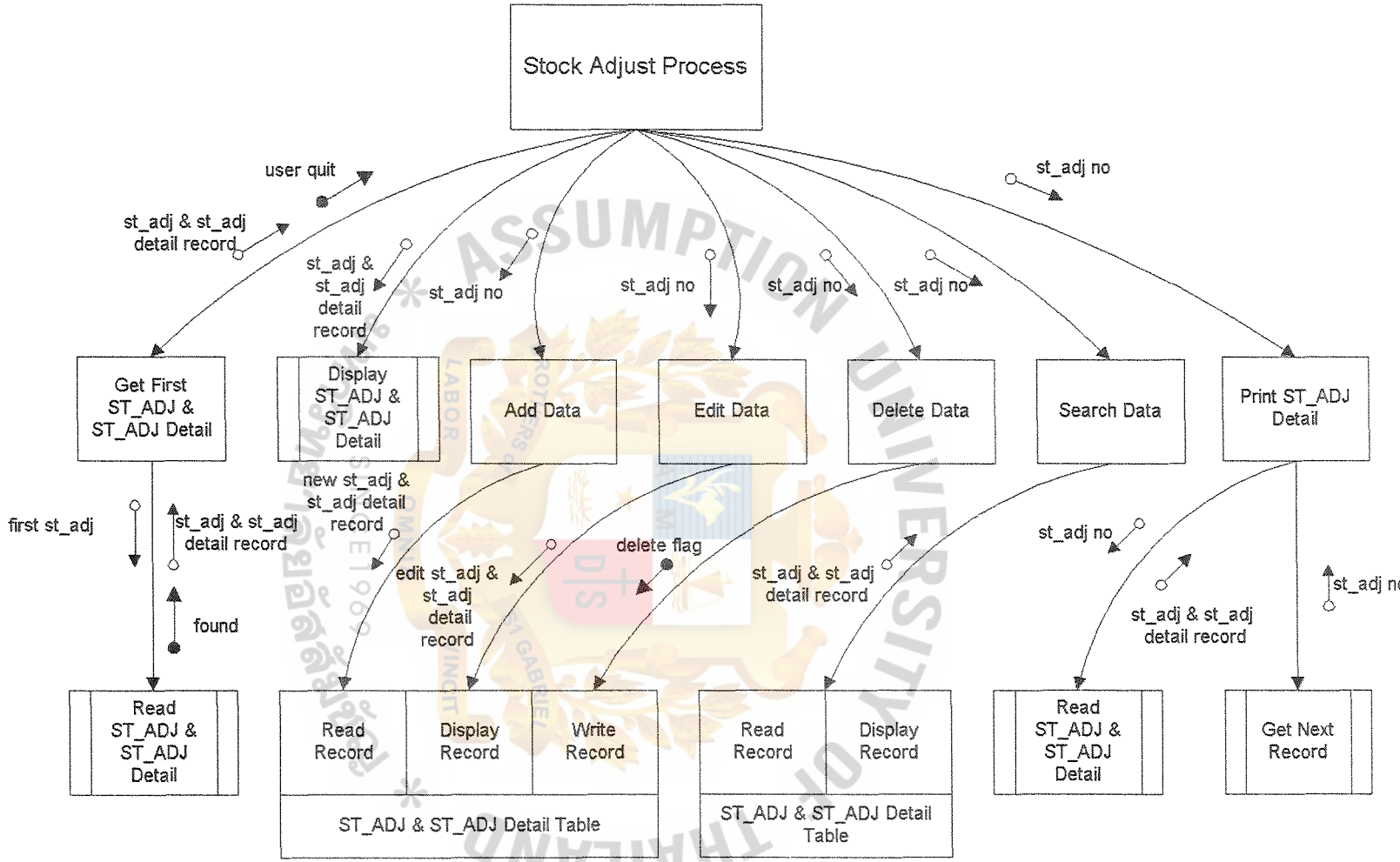


**Figure 3.32. Structure Chart of Stock Adjust Process**

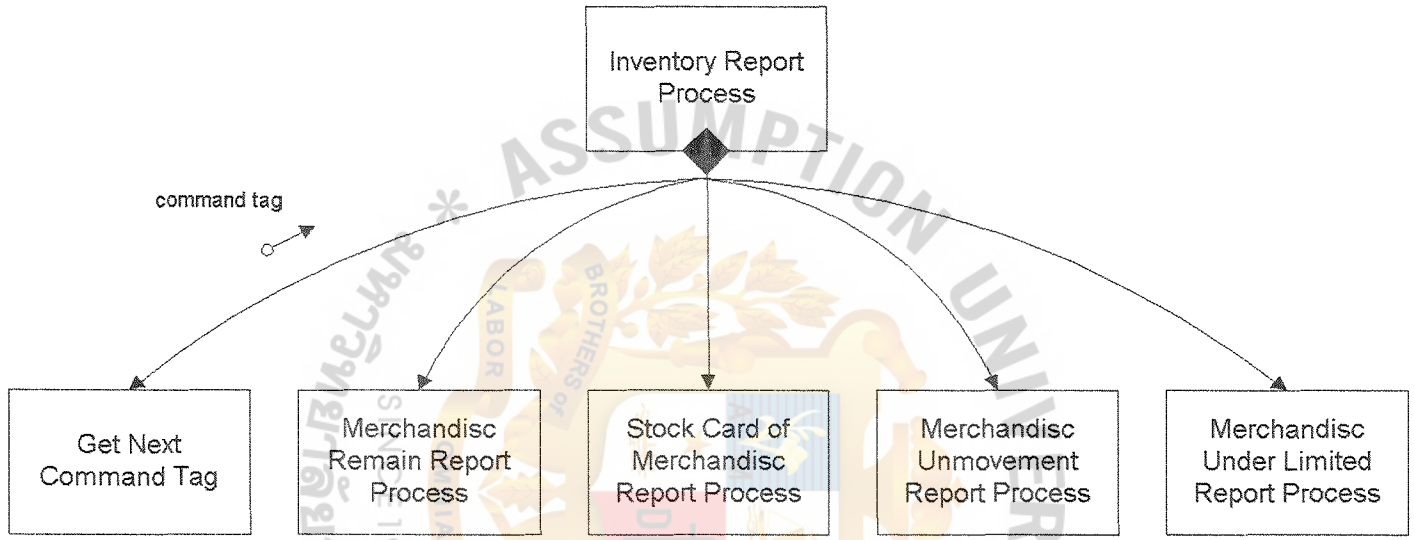


**Figure 3.33. Structure Chart of Inventory Report Process**


**Figure 3.34. Structure Chart of Merch Remain Report Process**

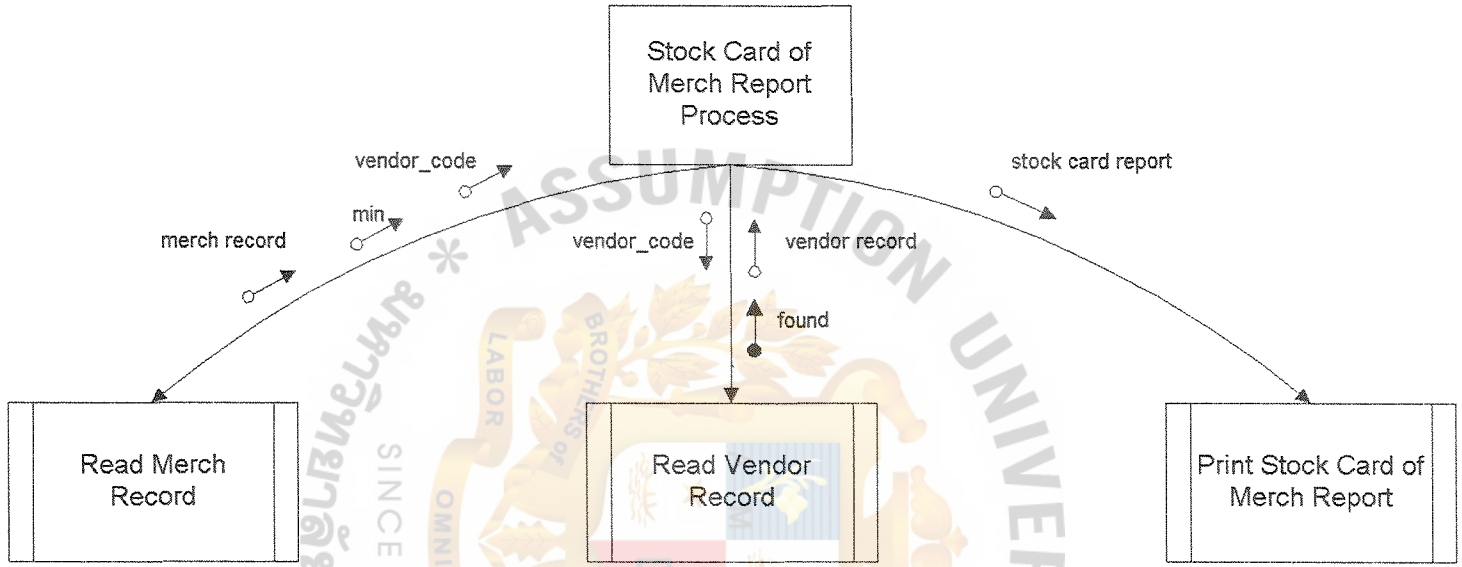


**Figure 3.35. Structure Chart of Stock Card of Merch Report Process**

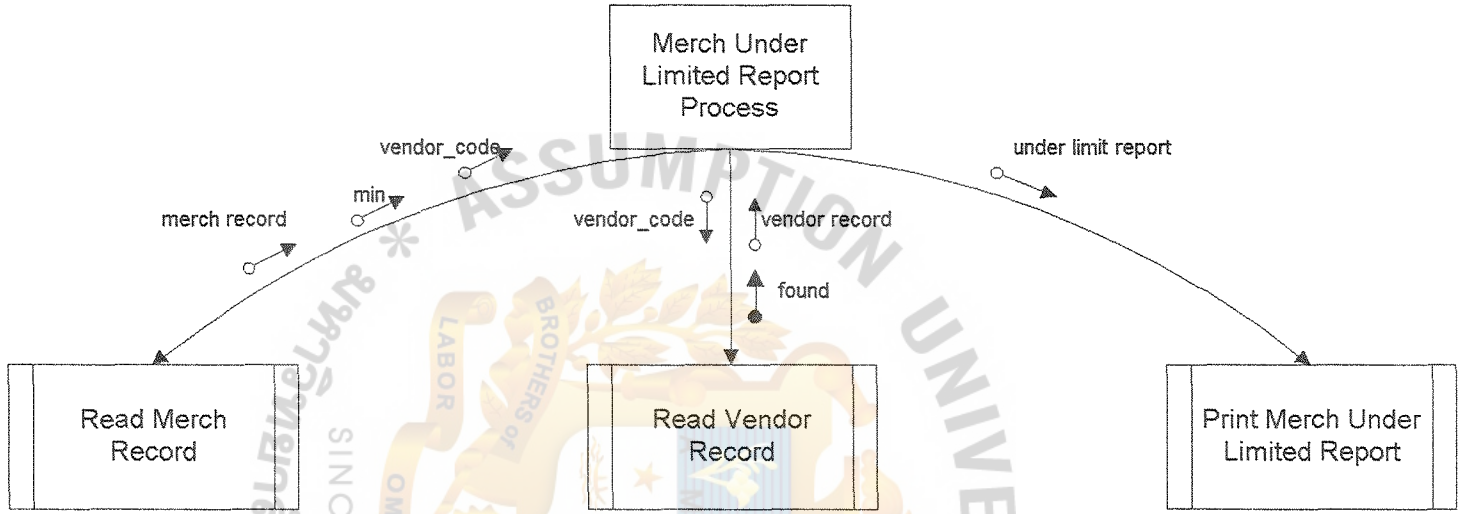


**Figure 3.36. Structure Chart of Merch Under Limitted Report Process**

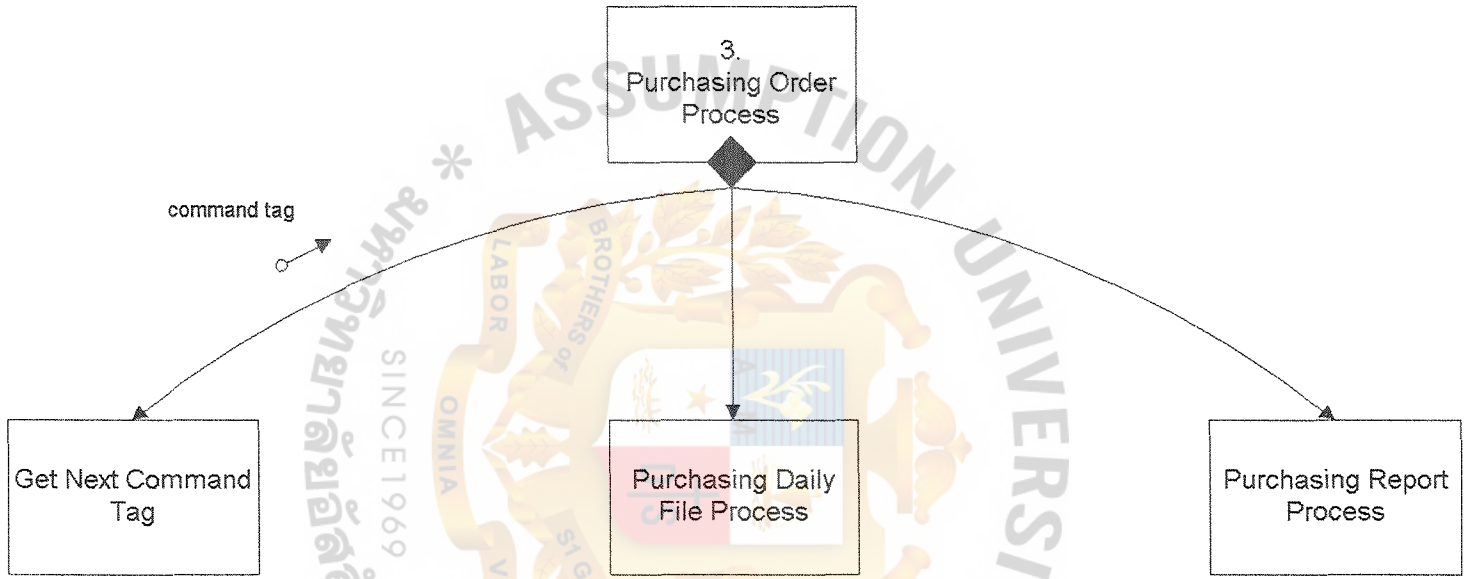


**Figure 3.37. Structure Chart of Purchasing Order Process**

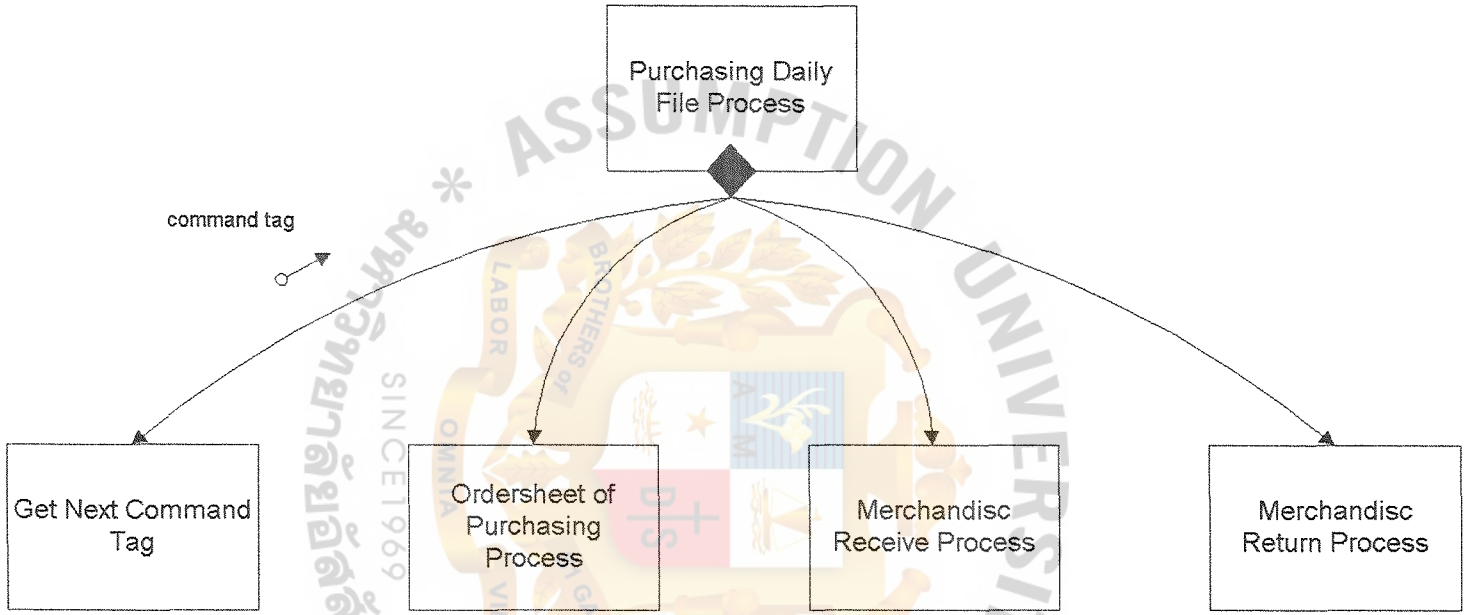


**Figure 3.38. Structure Chart of Purchasing Daily File Process**

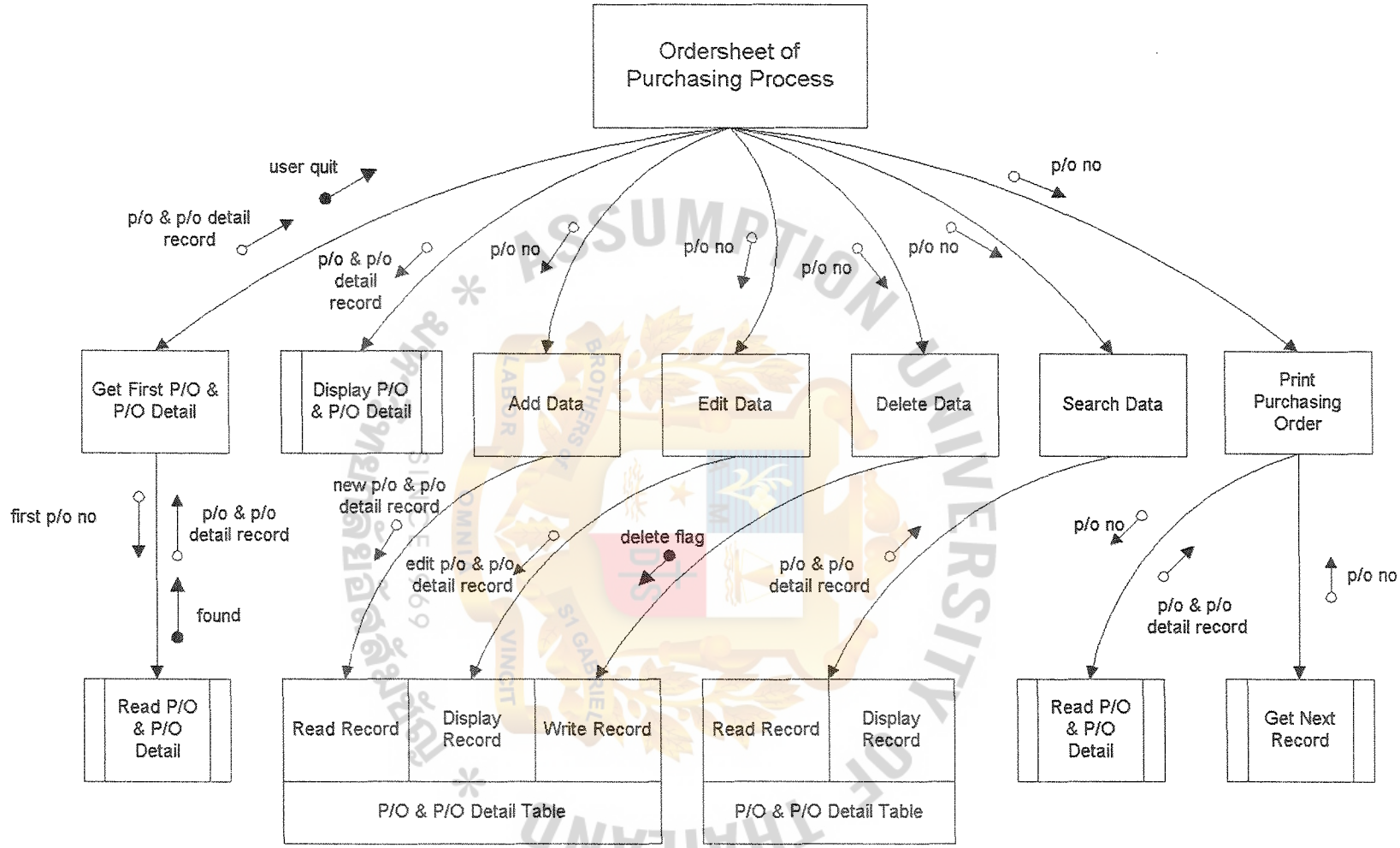


Figure 3.39. Structure Chart of Ordersheet of Purchasing Process

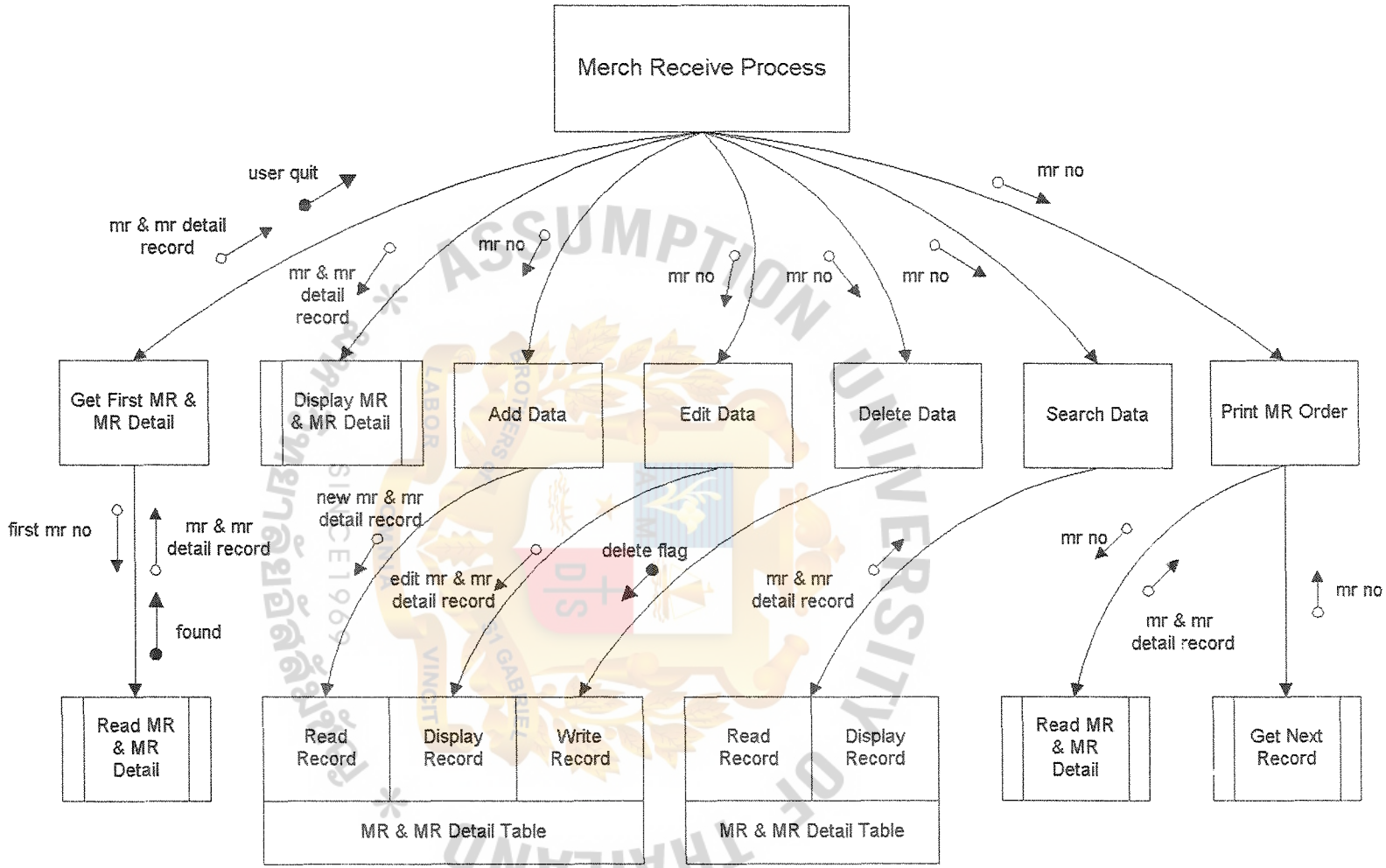


**Figure 3.40. Structure Chart of Merch Receive Process**

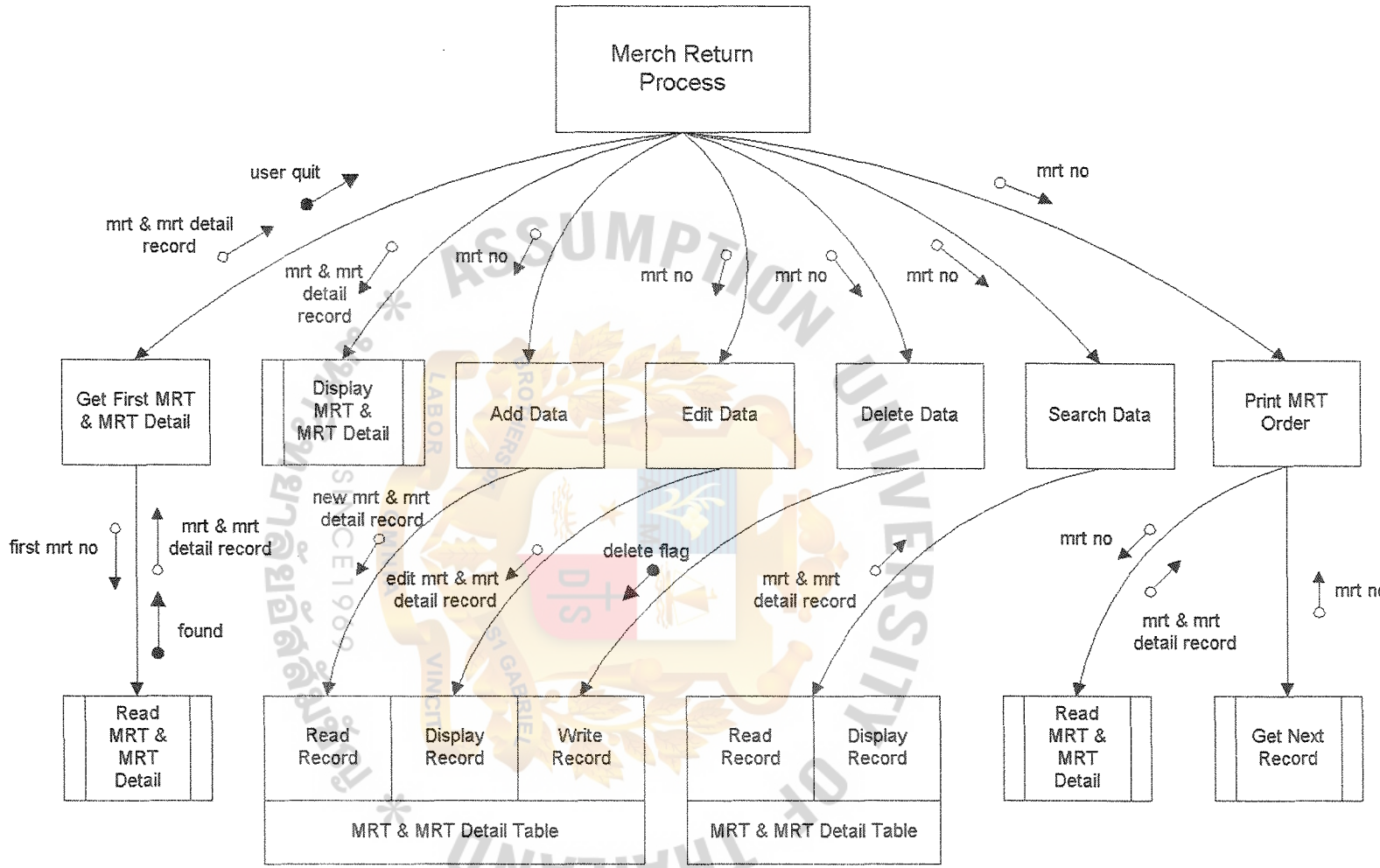


Figure 3.41. Structure Chart of Merch Return Process

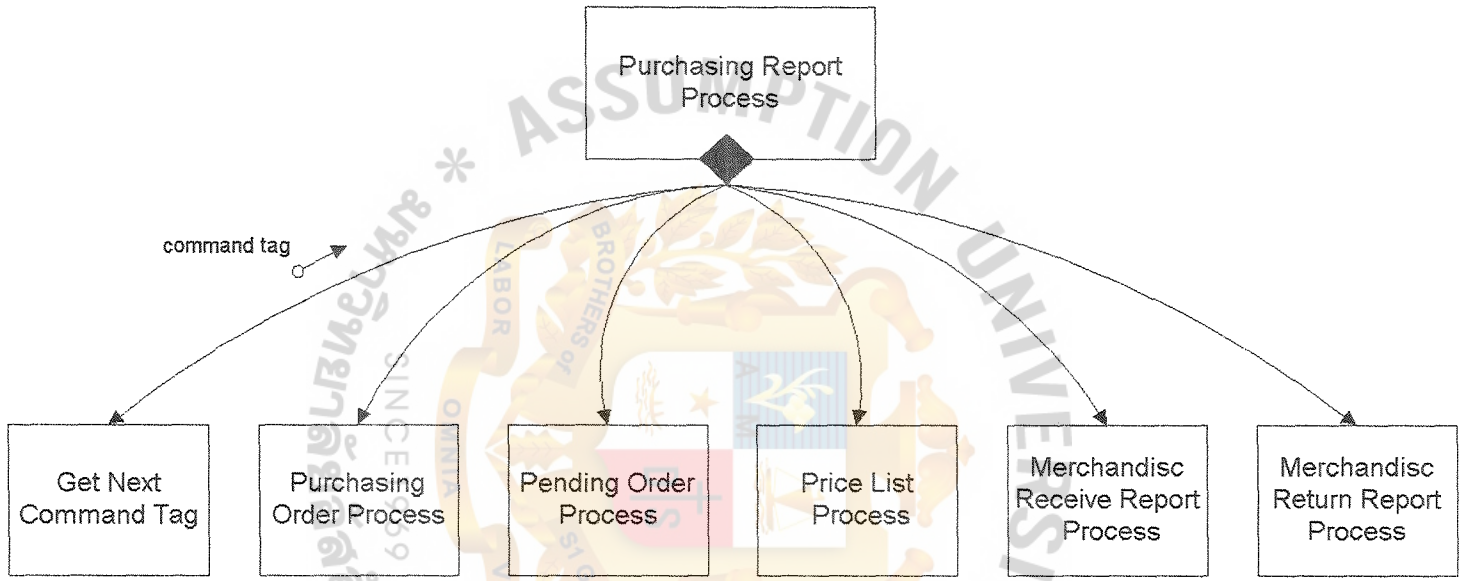


**Figure 3.42. Structure Chart of Purchasing Report Process**

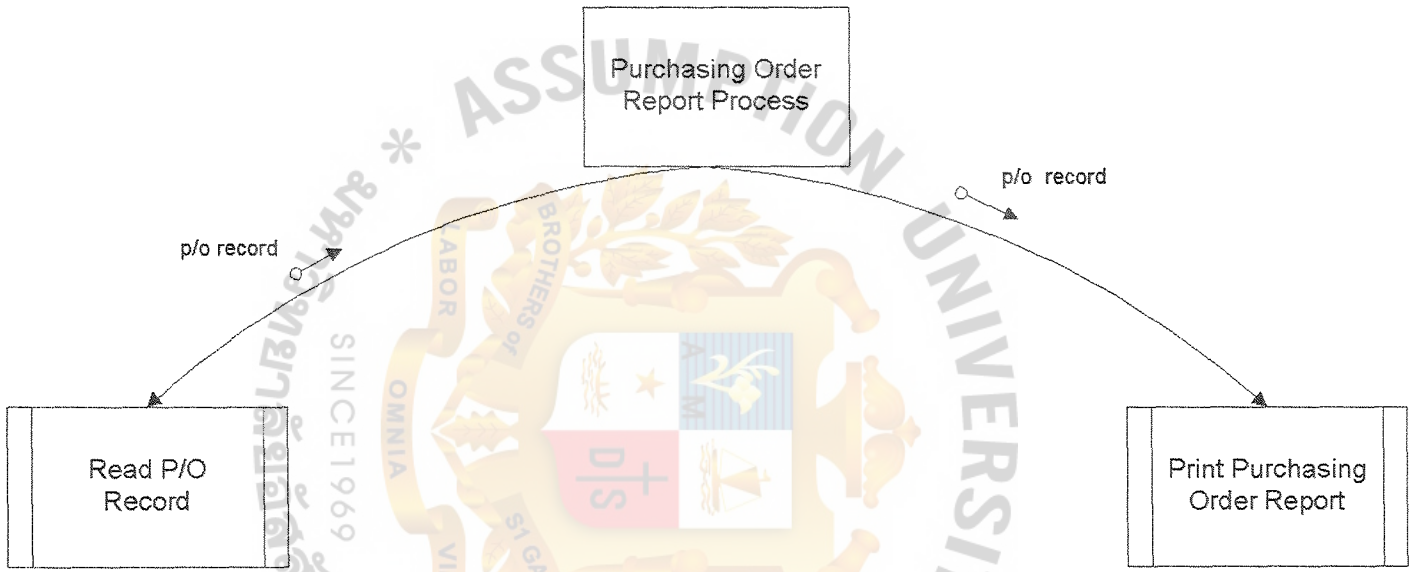


**Figure 3.43. Structure Chart of Purchasing Order Report Process**

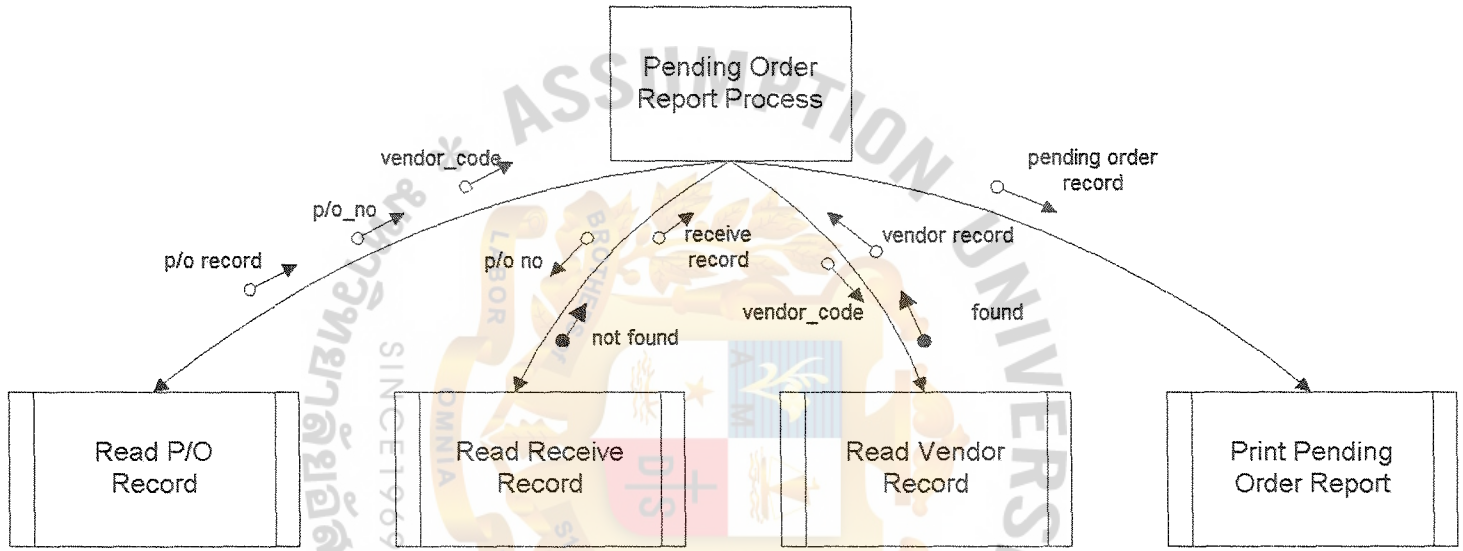


**Figure 3.44. Structure Chart of Pending Order Report Process**

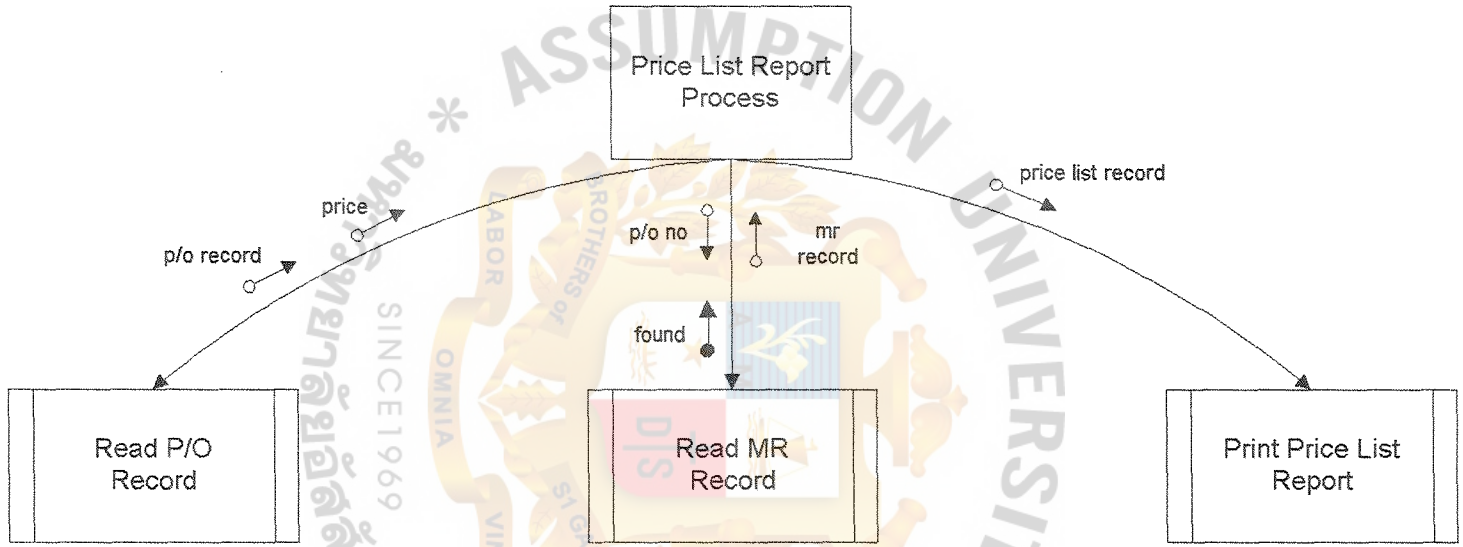


**Figure 3.45. Structure Chart of Price List Report Process**

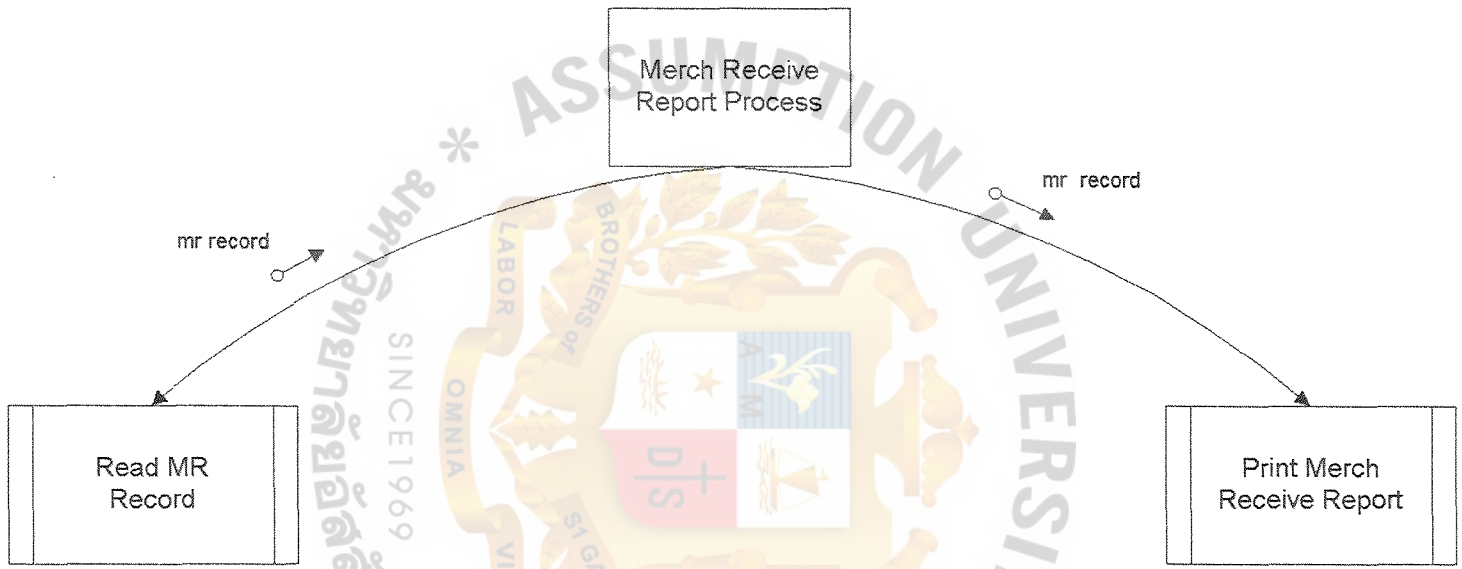


**Figure 3.46. Structure Chart of Merch Receive Report Process**

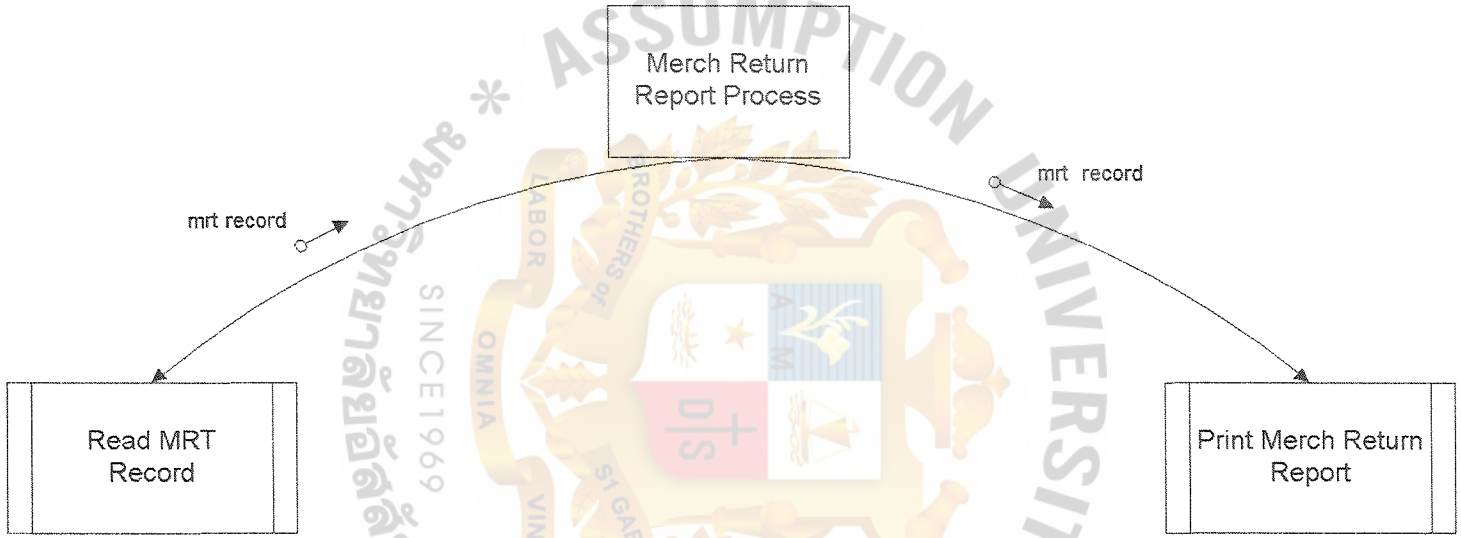


**Figure 3.47. Structure Chart of Merch Return Report Process**

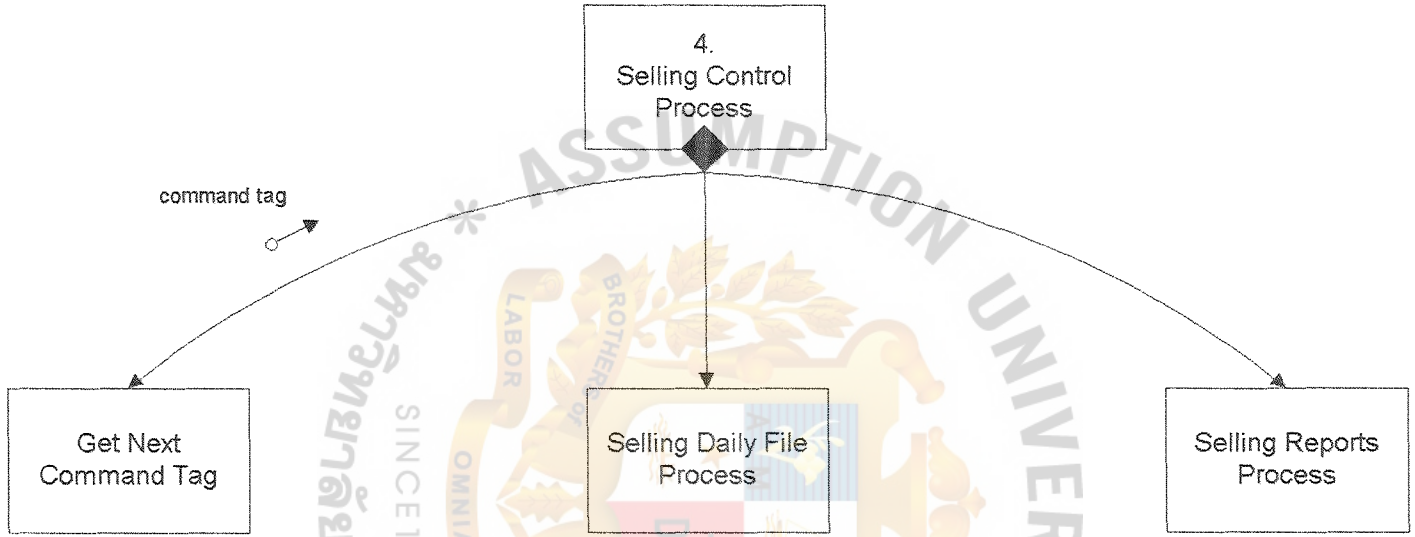


**Figure 3.48. Structure Chart of Selling Control Process**

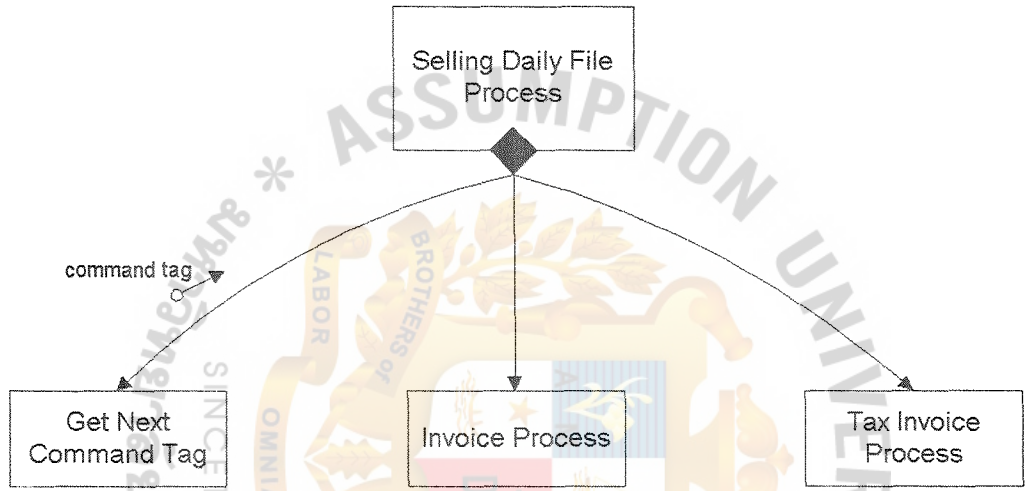


**Figure 3.49. Structure Chart of Selling Daily File Process**

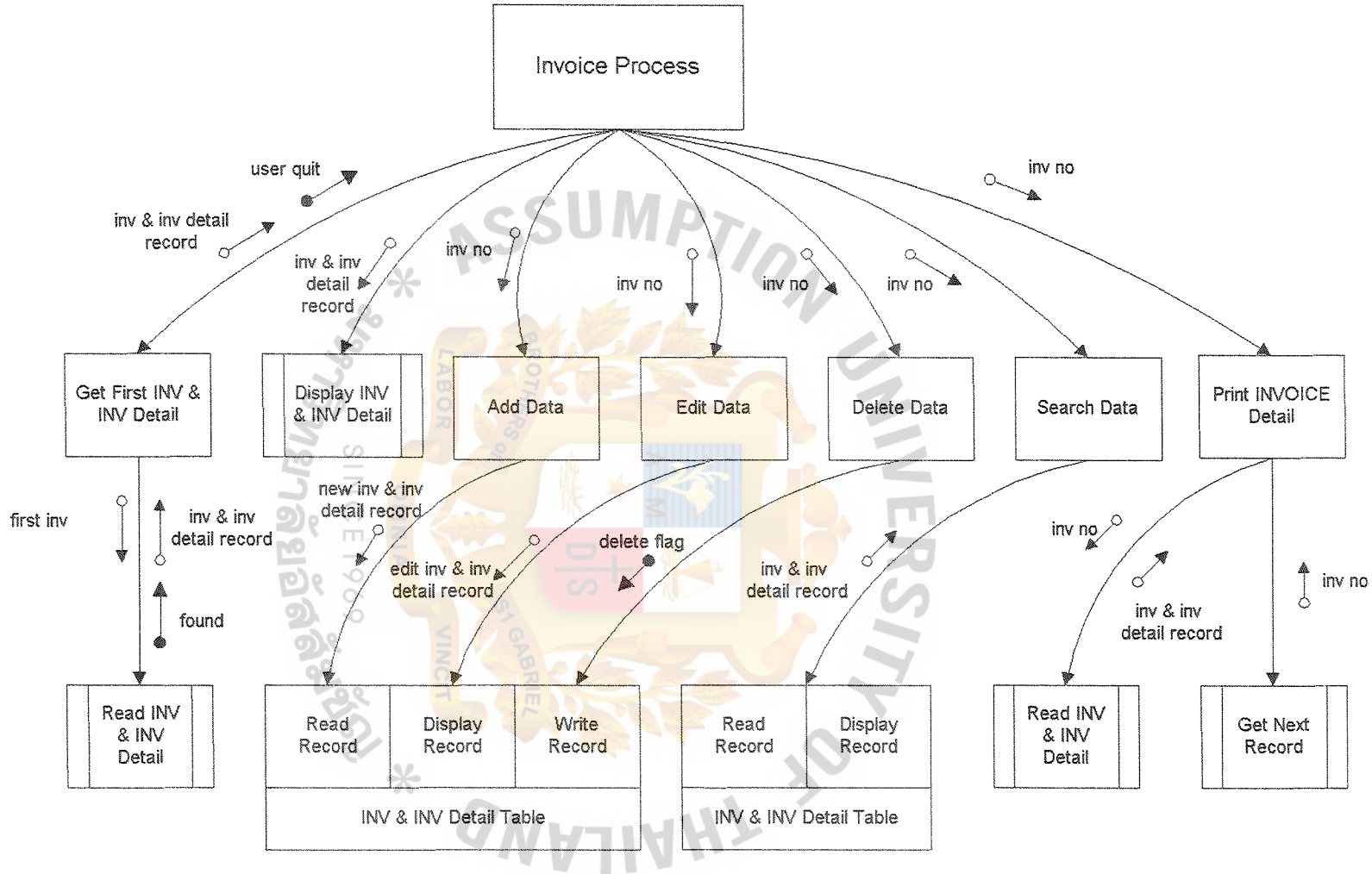


**Figure 3.50. Structure Chart of Invoice Process**

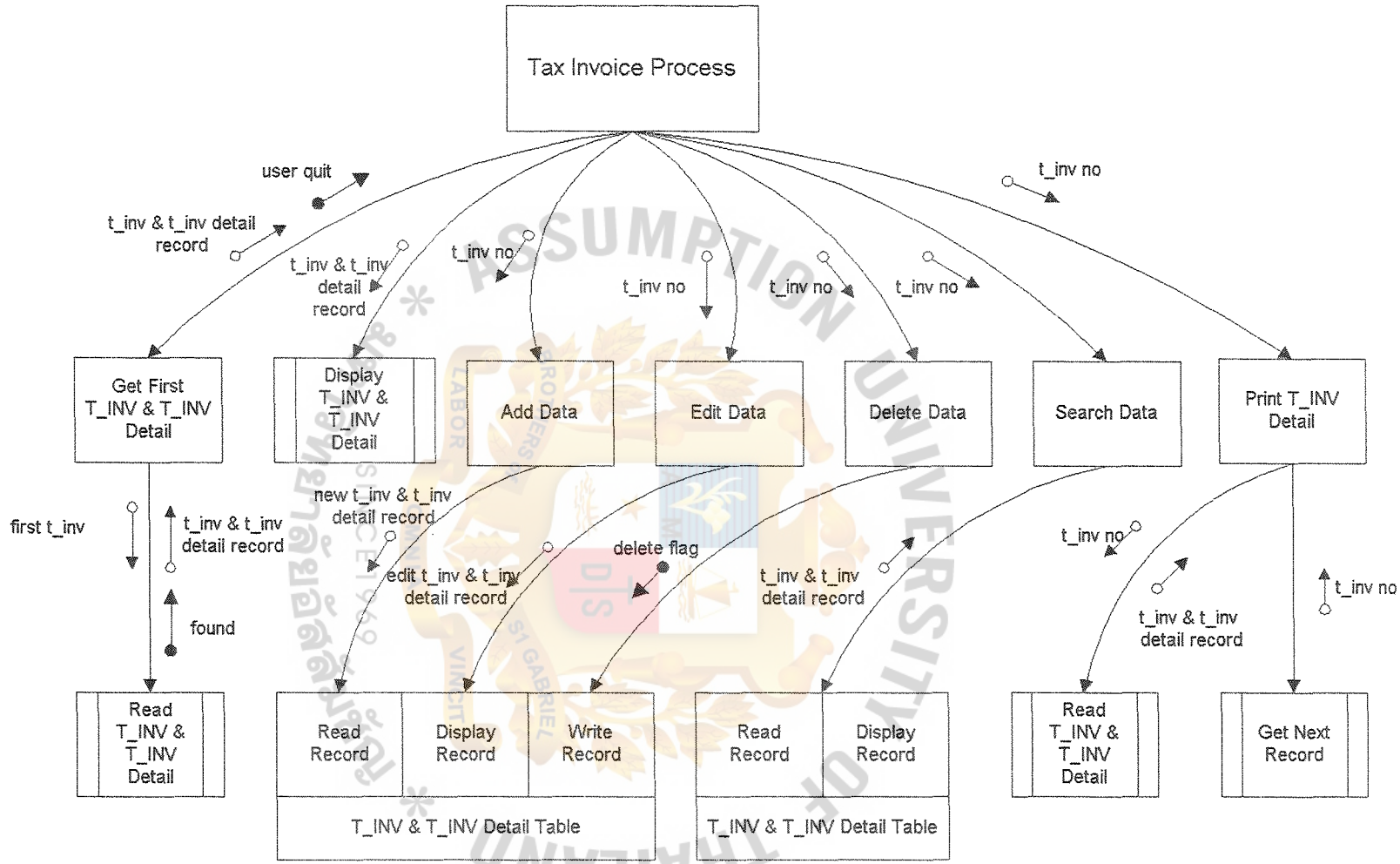


Figure 3.51. Structure Chart of Tax nvoice Process

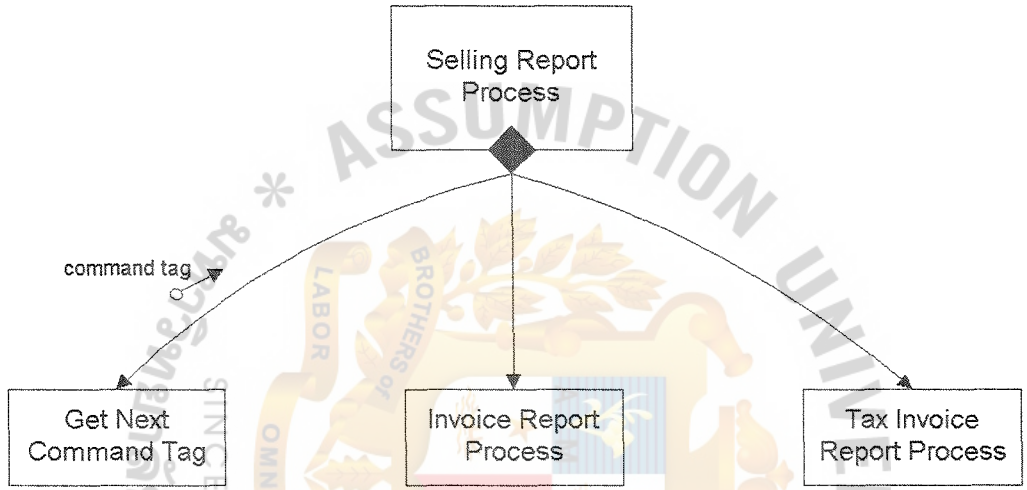


**Figure 3.52. Structure Chart of Selling Report Process**

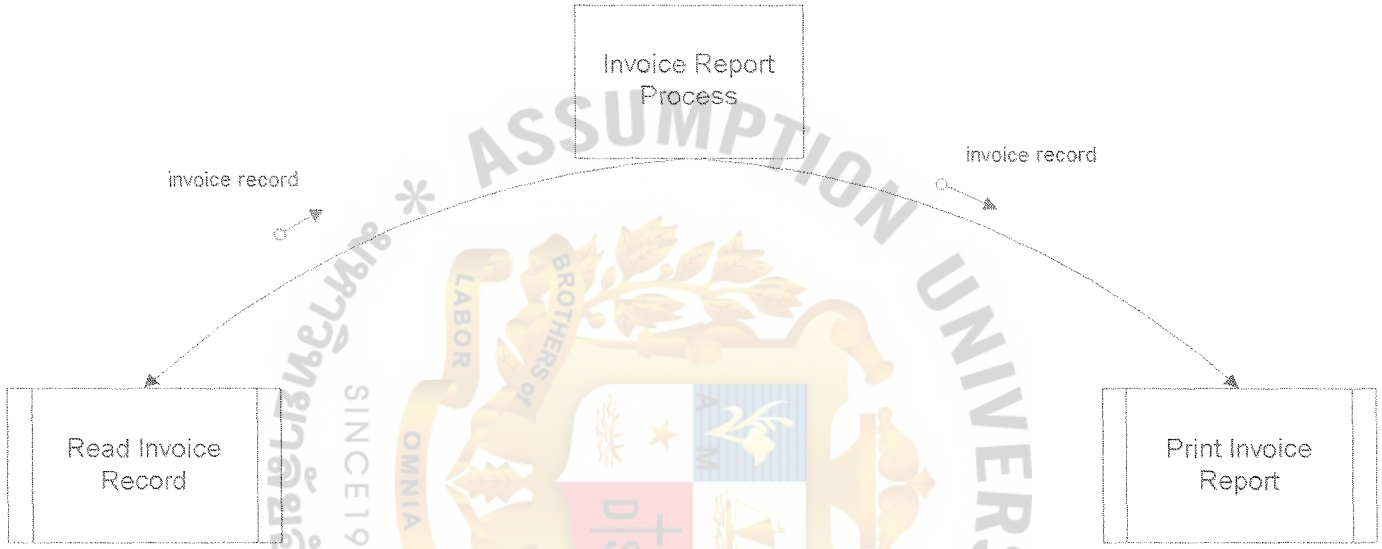


Figure 3.53. Structure Chart of Invoice Report Process

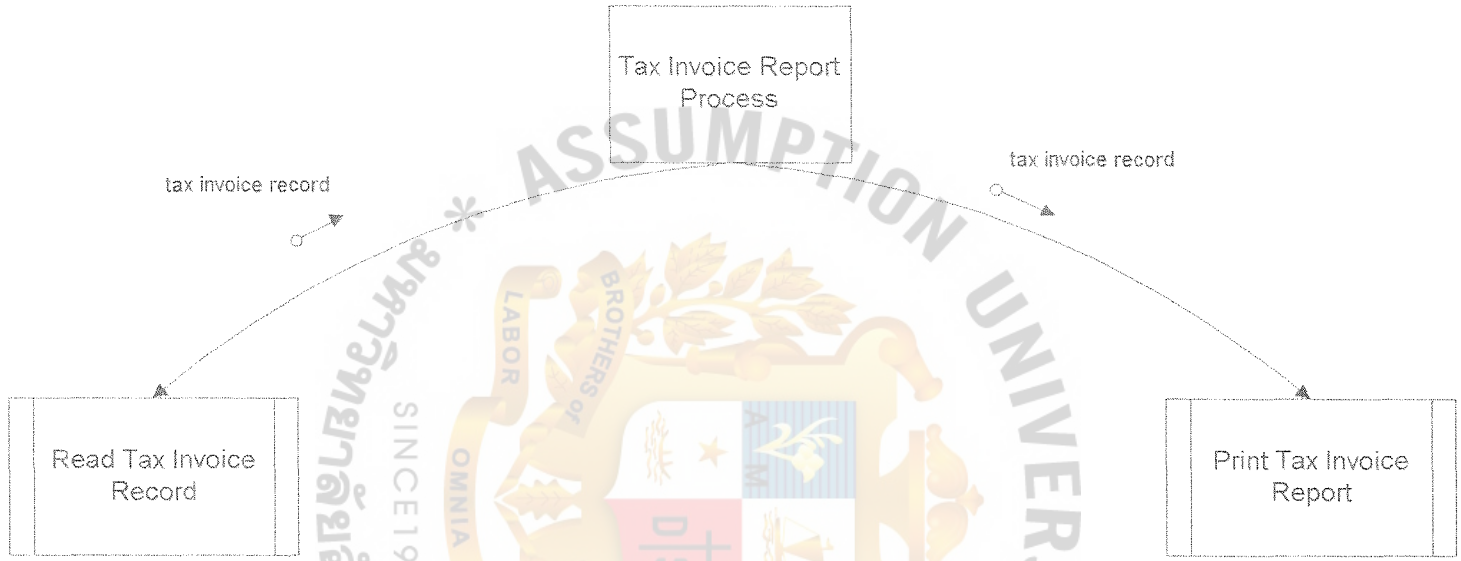


Figure 3.54. Structure Chart of Tax Invoice Report Process

### 3.2.4 Chart Of Screen Menu

The charts of the followings make user able to overview working system clear and easy to understand.

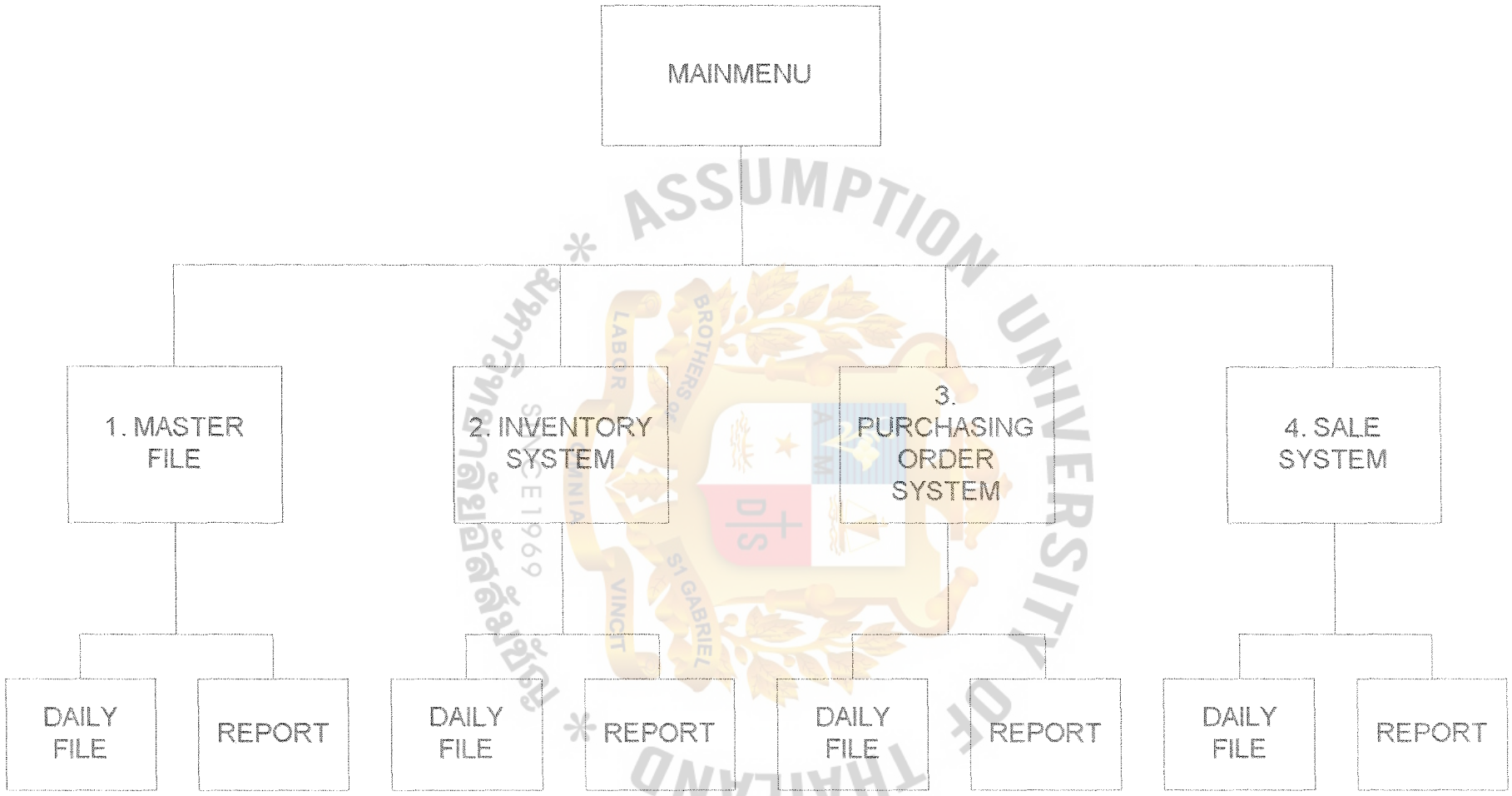


Figure 3.55. Chart of Screen Mainmenu

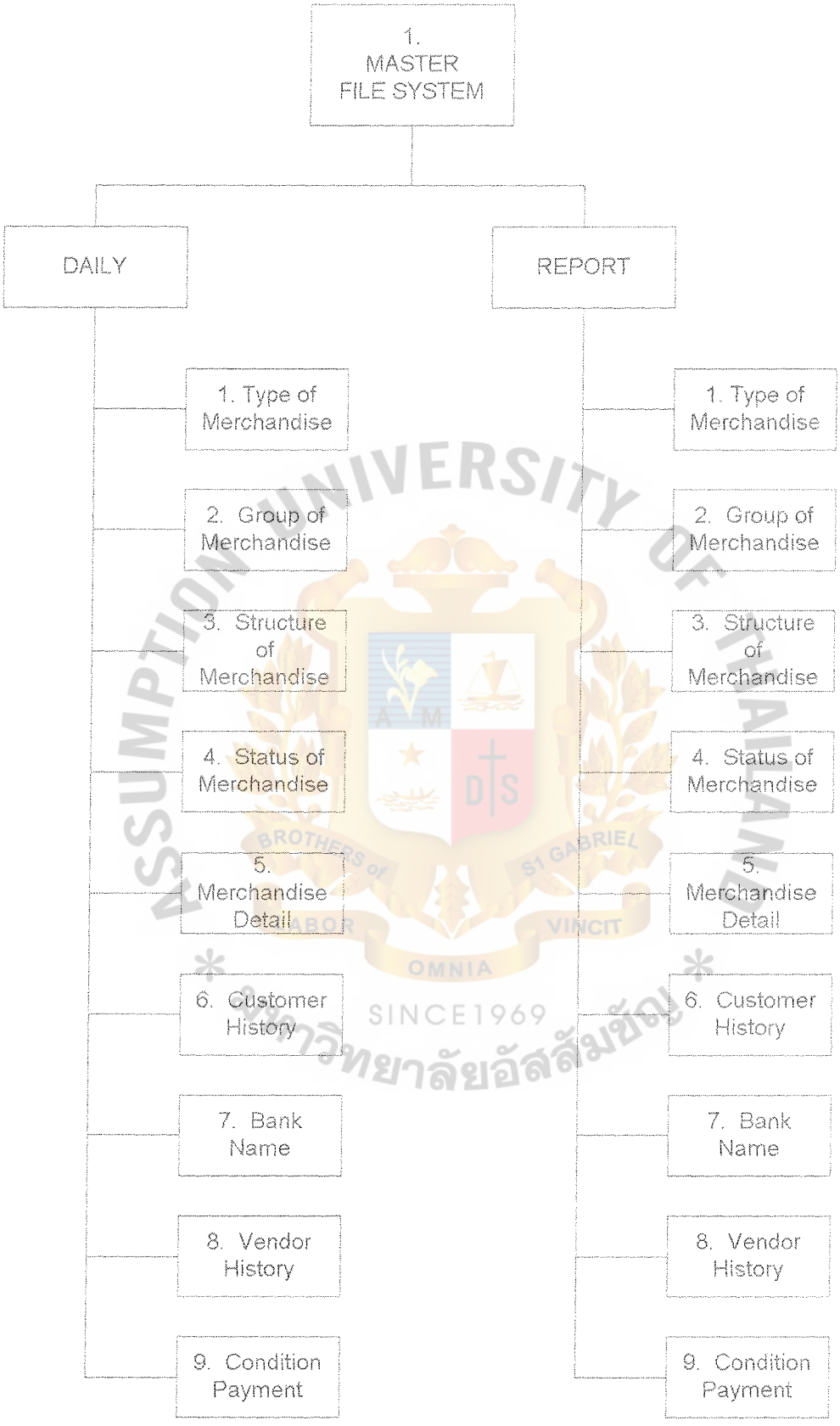


**Figure 3.56. Chart of Screen Menu of MASTER FILE SYSTEM**

```
1. MASTER FILE SYSTEM

[ 1. Type Of Merchandise ]

Type Code   : x

Type Name   : xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

F2=Add   F3=Edit   F4=Del   F5=Find   ESC=Exit
```

**Figure 3.57. Daily File Type of Merchandise**

```
[ 2. Group Of Merchandise ]

Group Code   : xx

Group Name   : xxxxxxxxxxxxxxxxxxxxxxxxxxxx

F2=Add   F3=Edit   F4=Del   F5=Find   ESC=Exit
```

**Figure 3.58. Daily File Group of Merchandise**

```
1. MASTER FILE SYSTEM

[ 3. Structure Of Merchandise ]

Type Code    : x

Type Name    : xxxxxxxxxxxxxxxx

F2=Add    F3=Edit    F4=Del    F5=Find    ESC=Exit
```

**Figure 3.59. Daily File Structure of Merchandise**

```
[ 4. Status Of Merchandise ]

Status Code    : x

Status Name    : xxxxxxxxxxxxxx

F2=Add    F3=Edit    F4=Del    F5=Find    ESC=Exit
```

**Figure 3.60. Daily File Status of Merchandise**

```
1. MASTER FILE SYSTEM

                    [ 5. Merchandise Detail ]

Type Code      : x                 Type Name      : xxxxxxxxxx
Group Code     : x                 Group Name     : xxxxxxxxxx
Structure Code : x                 Structure Name : xxxxxxxxxx
Merch. Code    : xxxx              Merch. Name    : xxxxxxxxxx
Status Code    : x                 Status Name    : xxxxxxxxxx
Unit           : xxxxxxx

Capital        :          9,999,999
Price          :          9,999,999
Discount       :             99,999
Vat            :        VAT  ( Y/N )
Minimum Quanlity :            9,999
Maximum Quanlity :           99,999
Balance Quanlity :           99,999
Pending Quanlity :           99,999
Location       :           xxxxxxx

F2=Add     F3=Edit     F4=Del     F5=Find     ESC=Exit
```

**Figure 3.61. Daily File Merchandise Detail**

1. MASTER FILE SYSTEM

[ 6. Customer History ]

CUSTOMER CODE : K-0020

CUSTOMER NAME : K - SUPPLY

TAX ID :

ADDRESS : 70 SOI SUKHUMVIT

ZIP CODE : 10260 TELEPHONE : 311-9999

EXECUTIVE NAME :

CUSTOMER LEVEL : A ( A = Very Good, B = Good, C = )

REMARK :

F2 = Add F3 = Edit F4 = Del F5 = Find ESC = EXIT

**Figure 3.62. Daily File Customer History**

```
1. MASTER FILE SYSTEM

[ 7. Bank Name ]

Bank Code   : FMB
Bank Name   : THAI FAMER BANK

F2=Add   F3=Edit   F4=Del   F5=Find   ESC=Exit
```

**Figure 3.63. Daily File Bank Name**

```
[ 8. Vendor History ]

Vendor Code :  x-xxxx  Vendor Name  :  xxxxxxx.........xx
Tax Id      :  xxxxxxxx
Address     :
Zip Code    :  xxxxx
Telephone   :
Executive Name  :

F2 = Add   F3 = Edit   F4 = Del   F5 = Find   ESC = EXIT
```

**Figure 3.64. Daily File Vendor History**

```
1. MASTER FILE SYSTEM

[ 9. Condition Payment ]

Payment Code    : 000  ( 0XX = cash,  1XX = credit )
Description     : CASH

F2=Add   F3=Edit   F4=Del   F5=Find   ESC=Exit
```

**Figure 3.65. Daily File Condition Payment**

1. MASTER FILE REPORT

1. TYPE OF MERCHANDISE REPORT PAGE :

| Type Code | Type Name |
|---|---|
| x | xxxxx.....xxxxxxxxxx |
| " | " |
| " | " |
| " | " |
| " | " |

PRINT DATE : 99/99/99 TIME : HH : MM.

**Figure 3.66. Type of Merchandise Report**

1. MASTER FILE REPORT

2. GROUP OF MERCHANDISE REPORT PAGE :

| Group Code | Group Name |
|---|---|
| xx | xxxxxxxxxxxxxxxxxxxxxx |
| " | " |
| " | " |
| " | " |
| " | " |

PRINT DATE : 99/99/99 TIME : HH : MM.

**Figure 3.67. Group of Merchandise Report**

1. MASTER FILE REPORT

3. STRUCTURE OF MERCHANDISE REPORT PAGE :

| Structure Code | Structure |
|---|---|
| x | xxxxxx.....xxxxx |
| " | " |
| " | " |
| " | " |

PRINT DATE : 99/99/99 TIME : HH : MM.

**Figure 3.68. Structure of Merchandise Report**

1. MASTER FILE REPORT

4. STATUS OF MERCHANDISE REPORT PAGE :

| Status Code | * Status Name |
|---|---|
| x | xxxxxx.....xxxxx |
| " | " |
| " | " |
| " | " |

PRINT DATE : 99/99/99 TIME : HH : MM.

**Figure 3.69. Status of Merchandise Report**

1. MASTER FILE REPORT

5. MERCHANDISE DETAIL REPORT PAGE :

| Order Code | Order Name | Unit | Balance Qty. | Pending Qty. | Max Qty. | Min Qty. | Capital | Price |
|---|---|---|---|---|---|---|---|---|
| xxxx | xxxxxxxxxx | xxxxxxx | 999,999 | 99,999 | 99,999 | 9,999 | 999,999 | 999,999 |
| " | " | " | " | " | " | " | " | " |
| " | " | " | " | " | " | " | " | " |
| " | " | " | " | " | " | " | " | " |

PRINT DATE : 99/99/99 TIME : HH : MM :

**Figure 3.70. Merchandise Detail Report**

1. MASTER FILE REPORT

6. CUSTOMER HISTORY REPORT PAGE :

Customer Code : xxxxx Customer Name : xxxxxxxxxxxxxxxxxxxxxxx

Tax - Id : x-xx-x-xxxxx-x

Address : xxxxxxxxxxxxxxx

:

Telephone : xxx - xxxx

PRINT DATE : 99/99/99 TIME : HH : MM.

**Figure 3.71. Customer History Report**

1. MASTER FILE REPORT

7. BANK NAME REPORT PAGE :

| Bank Code | Bank Name |
|---|---|
| xxx | xxxxxx.....xxxxx |
| " | " |
| " | " |
| " | " |
| " | " |

PRINT DATE : 99/99/99 TIME : HH : MM.

**Figure 3.72. Bank Name Report**

1. MASTER FILE REPORT

8. VENDOR HISTORY REPORT PAGE :

Vendor Code : xxxxx Vendor Name : xxxxxxxxxxxxxxxxxxxxxx

Administer : 1. xxxxxxxxxx 2. xxxxxxxxxx

TAX - ID : x-xx-x-xxxxx-x

Address : xxxxxxxxxxxxxxx

:

Telephone : xxx - xxxx

PRINT DATE : 99/99/99 TIME : HH : MM.

**Figure 3.73. Vendor History Report**

1. MASTER FILE REPORT

9. CONDITIONAL PAYMENT REPORT PAGE :

| Code | Conditional |
|---|---|
| xxx | xxxxxx.....xxxxx |
| " | " |
| " | " |
| " | " |

PRINT DATE : 99/99/99 TIME : HH : MM.

**Figure 3.74. Conditional Payment Report**

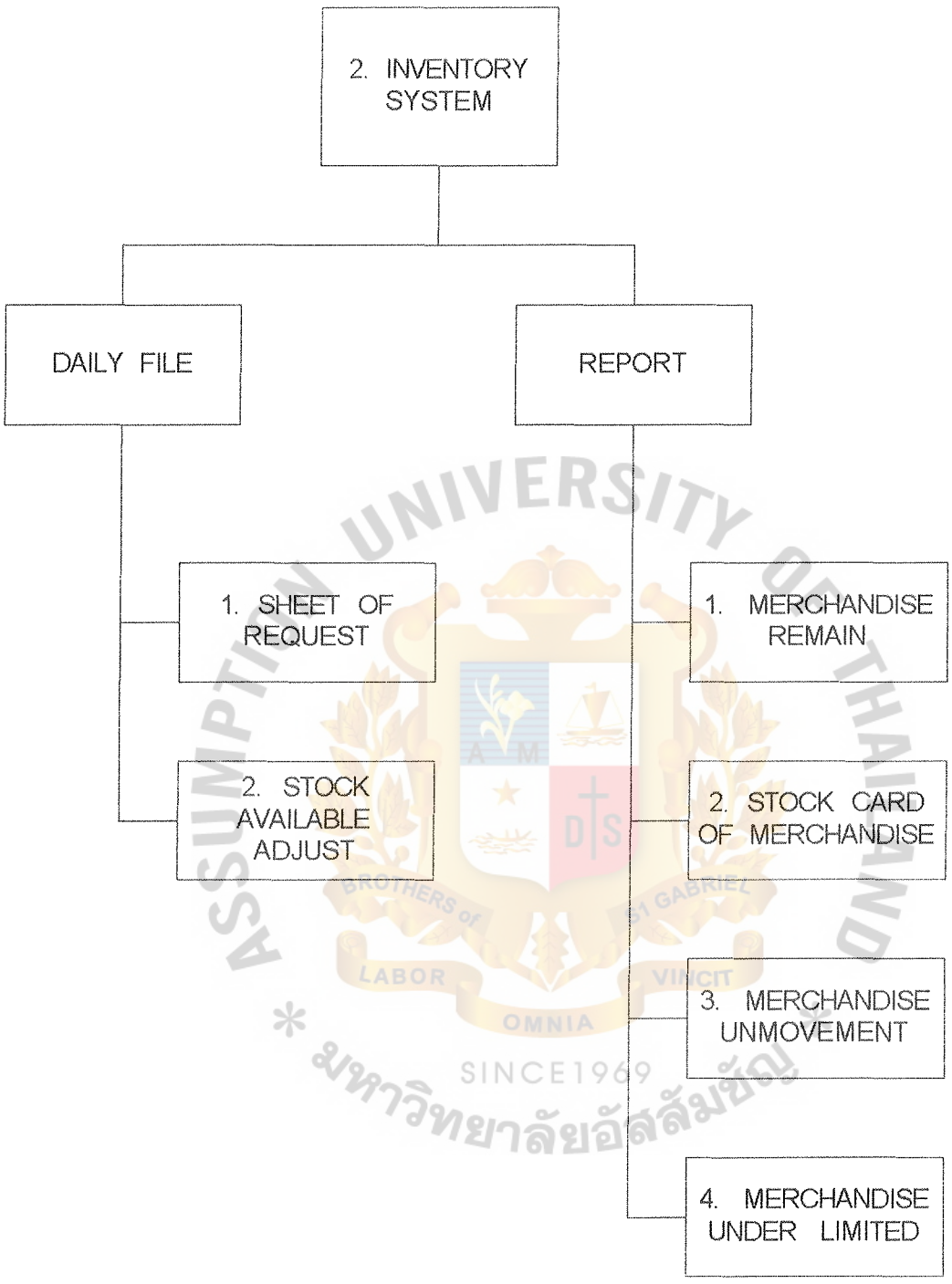


**Figure 3.75. Chart of Screen Menu of INVENTORY SYSTEM**

2. INVENTORY SYSTEM

[ 1. Sheet Of Request ]

Requist Number : 40INV000001 Order Date : 13/05/98

Customer Code : 99999 - General Customer

Conditional Of Payment : 007 - Cash 07 Days

Referrence Number : 10111111111

Note :

| Merch, Code | Merchandise name | Quantity | Unit name | Unit price | Total |
|---|---|---|---|---|---|
| 101C-V500-0 | Vacuum Cleaner | 2 | price | 7,900 | 15,800 |

F5 = Find ESC=Exit

**Figure 3.76. Daily File Sheet of Request**

2. INVENTORY SYSTEM

[ 2. Stock Available Adjust ]

| | | |
|---|---|---|
| Type Of Merchandise | : | 1 - Import |
| Merchandise Code | : | 101C-V100-0 |
| Unit Name | : | Price (s) |
| Date | : | 50.00 ( + / - ) |
| Note | : | |

F2 = Add F5 = Find ESC=Exit

**Figrue 3.77. Daily File Stock Available Adjust**

2. INVENTORY SYSTEM

1. MERCHANDISE REMAIN PAGE :

Type of Merchandise : 1 - Import

| Merch. Code | Merch. Name | Unit Name | Balance |
|---|---|---|---|
| Group Of Merch. : 01 - Electric | | | |
| 101C-V100-0 | Vacuum Cleaner A.B. 900 | Piece(s) | 40 |
| 101C-V200-0 | Vacuum Cleaner A.B. 850 | Piece(s) | 38 |
| Group Of Merch. : 02 - Kitchen Utensils | | | |
| 102C-K200-0 | Steam Boiler | Piece(s) | 20 |

PRINT DATE : 99/99/99 TIME : HH : MM :

**Figure 3.78. Merchandise Remain Report**

2. INVENTORY SYSTEM

2. STOCK CARD OF MERCHANDISE REPORT PAGE :

| Type Code | Merch Code | Order Date | Document Number | Merch Name | Merch Receive | Merch Pay | Note |
|---|---|---|---|---|---|---|---|
| 1 | 101C-A900-0 | 13/05/40 | 40INV000001 | Vacuum Cleaner A.B. 900 | | | |
| | 101C-A850-0 | 14/05/40 | 40INV000003 | Vacuum Cleaner A.B. 850 | | | |

PRINT DATE : 99/99/99 TIME : HH : MM :

**Figure 3.79. Stock Card of Merchandise Report**

2. INVENTORY SYSTEM

3. MERCHANDISE UNMOVEMENT REPORT PAGE :

| Merch Code | Merch Name | Merch Remain | Unit Name | Last Sale Date | Amount Of Days |
|---|---|---|---|---|---|
| 110S-x020-0 | xxx_____xx | 2 | Price | 01/02/97 | 410 Days |
| 110S-x040-0 | Water Morters | 2 | Price | 01/02/97 | 410 Days |

PRINT DATE : 99/99/99 TIME : HH : MM:

**Figure 3.80. Merchandise Unmovement Report**

2. INVENTORY SYSTEM

4. MERCHANDISE UNDER LIMIT REPORT

Type of Merchandise 1 - Import PAGE :

| Merchandise | | Balance | Minimum Limited | Unit Name |
|---|---|---|---|---|
| Goup of merch | 02 - Kitchen | | | |
| 102C-0020-0 | Steam Machine | 2 | 5 | Machine |

PRINT DATE : 99/99/99 TIME : HH : MM:

**Figure 3.81. Merchandise Under Limit Report**

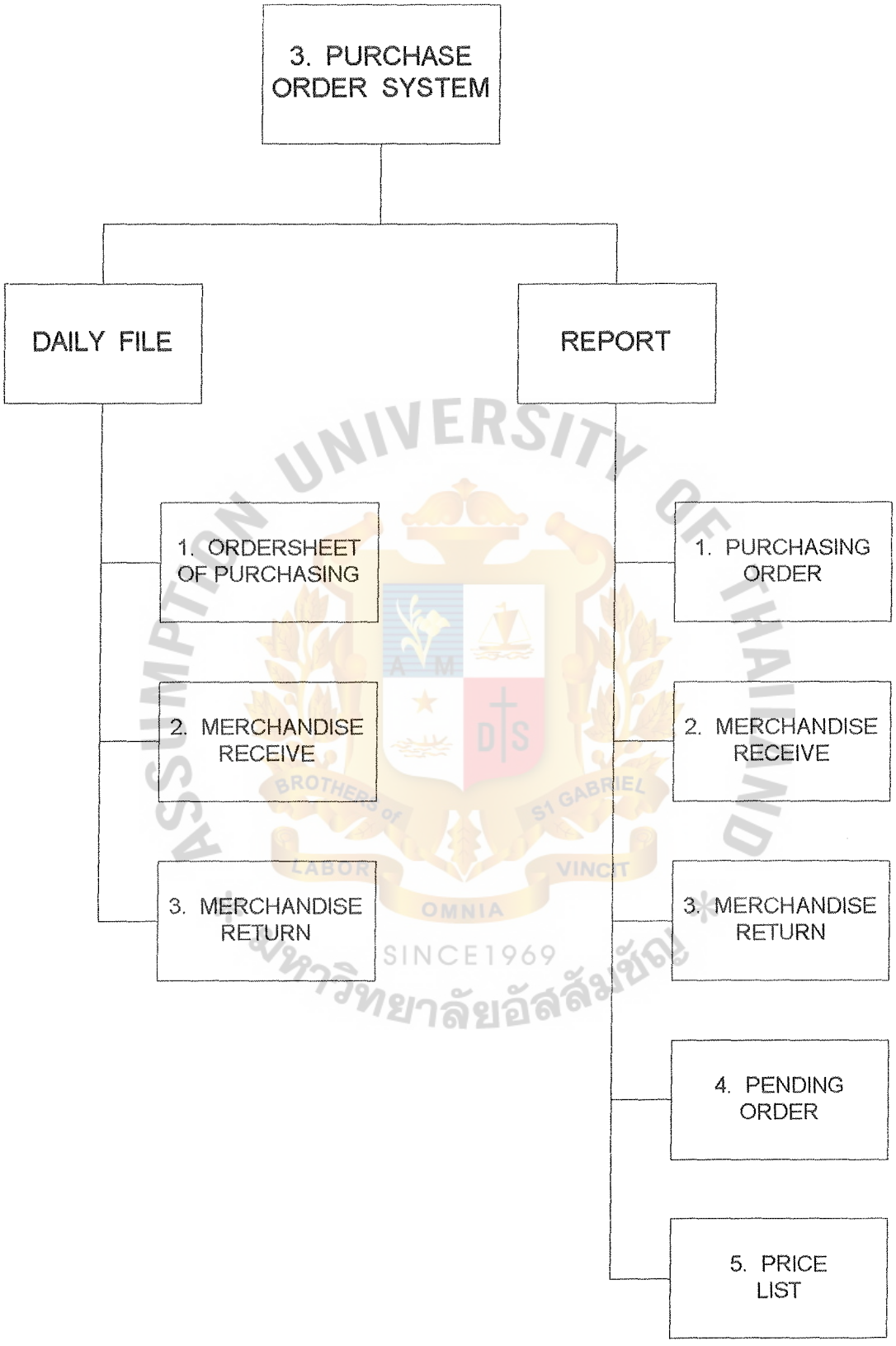


**Figure 3.82. Chart of Screen Menu of PURCHASE ORDER SYSTEM**

3. PURCHASING ORDER

[ 1. Ordersheet of Purchase ]

Vendor code : xxxxx - xxxxxxx.....xxx

Purchase Number : xxPOxxxxxxx  Order Date : 99/99/99

Condition Payment : ( ) 1. Cash  ( ) 2. Credit  Expire Date : 99/99/99

Note. :

| Product Code | Product Name | Quantity Order | Unit Name |
|---|---|---|---|
| xxxx-xxxx-x | xxxxxxxxxxx.....xxxx | 99,999 | xxxxxxx |
| " | " | " | " |
| " | " | " | " |

Total Item 99 Item Total Amount 9,999 xxxxxxx

F2 = Add F3 = Edit F4 = Del F5 = Find Esc=Exit

**Figure 3.83. Daily File Ordersheet of Purchase**

3. PURCHASING ORDER

[ 2. Merchandise Receive ]

Vendor code : xxxxx - xxxxxxx.....xxx

Invoice Number : xxINVxxxxxxx Order Date : 99/99/99

Purchase Number : xxPOxxxxxxx Expire Date : 99/99/99

Receive Date : 99/99/99

Note :

| Merch. Code | Merchandise Name | Receive Order | Unit |
|---|---|---|---|
| xxxx-xxxx-x | xxxxxxxxxxx.....xxxx | 99,999 | xxxxxxx |
| " | " | " | " |

Total Item 99 Item Total Amount 9,999 xxxxxxx

F2 = Add F3 = Edit F4 = Del F5 = Find Esc=Exit

**Figure 3.84. Daily File Merchandise Receive**

3. PURCHASING ORDER

[ 3. Merchandise Return ]

Vendor code : xxxxx - xxxxxxx.....xxx

Return Number : xxINVxxxxxxx  Order Date : 99/99/99

Purchase Number : xxPOxxxxxxx  Expire Date : 99/99/99

Receive Date : 99/99/99

Note :

| Merch. Code | Merchandise Name | # Return | Unit |
|---|---|---|---|
| xxxx-xxxx-x | xxxxxxxxxxx.....xxxx | 99 | xxxxxxx |
| " | " | " | " |

Total Item 99 Item Total Amount 9,999 xxxxxxx

F2 = Add F3 = Edit F4 = Del F5 = Find Esc=Exit

**Figure 3.85. Daily File Merchandise Return**

3. PURCHASING ORDER

1. PURCHASING ORDER REPORT

Vendor Code : xxxxx Vendor Name : xxxxxxxxxxxx.....xxxxxx PAGE :

| Merchandise Code | Merchandise Name | Order Date | Quantity Order | Unit Name |
|---|---|---|---|---|
| xxxx-xxxx-x | xxxxxxxxxx.....xxxxxxx | 99/99/99 | 9,999 | xxxxxxx |
| " | " | " | " | " |
| " | " | " | " | " |
| " | " | " | " | " |

PRINT DATE : 99/99/99 TIME : HH : MM:

**Figure 3.86. Purchasing Order Report**

3. PURCHASING ORDER

2. MERCHANDISE RECEIVE REPORT < SELECT PAYMENT : 1 = CASH > PAGE :

| Vendor | | Purchase Number | Order Date | Payment | |
|---|---|---|---|---|---|
| Report Number | Merchandise Code | Merchandise Name | | Amount of Merchandise | Report Name |
| xxxxx - xxxxxxx.....xxxxxxx | | xxPOxxxxxxx | 99/99/99 | Cash | |
| xxInvxxxxxxx | xxxx-xxxx-x | xxxxxx.....xxxxxxxx | | 9,999 | xxxxxxx |
| " | " | " | | " | " |
| " | " | " | | " | " |
| " | " | " | | " | " |

PRINT DATE : 99/99/99 TIME : HH : MM:

**Figure 3.87. Merchandise Receive Report**

3. PURCHASING ORDER

3. MERCHANDISE RETURN REPORT PAGE :

Vendor Code Vendor Name

Report Number Order Date

Merchandise Code Merchandise Name Amount of Receive/Return

xxxxx - xxxxxxx.....xxxxxxx

Purchanse Number : xxPOxxxxxxx

Receive Report xxPOxxxxxxx 99/99/99

xxxx-xxxx-x xxxx.....xxxxxx 9,999

Invoice Report xxInvxxxxxxx 99/99/99

xxxx-xxxx-x xxxx.....xxxxxx 9,999

PRINT DATE : 99/99/99 TIME : HH : MM:

**Figure 3.88. Merchandise Return Report**

3. PURCHASING ORDER

4. PENDING ORDER REPORT

Vendor Code : x-xxxx Vendor Name : xxxxx.....xxxxx PAGE :

| Order Number | Order Date | Merchandise Code | Merchandise Name | Quantity Order | Quantity Acceped | Quantity Pending | Quantity Balance |
|---|---|---|---|---|---|---|---|
| xxOR-xxxxx | 99/99/99 | xxxx-xxxx-x | xxxxx_____xxx | 9,999 | 9,999 | 9,999 | 9,999 |
| " | " | " | " | " | " | " | " |
| " | " | " | " | " | " | " | " |
| " | " | " | " | " | " | " | " |

PRINT DATE : 99/99/99 TIME : HH : MM :

**Figure 3.89. Pending Order Report**

3. PURCHASING ORDER

5. PRICE LIST REPORT

Vendor Code : x-xxxx Vendor Name : xxxxx.....xxxxx PAGE :

| Receive Number | Receive Date | Merchandise Code | Merchandise Name | Unit Name | New Price / Unit | Old Price / Unit |
|---|---|---|---|---|---|---|
| xxPR-xxxxx | 99/99/99 | xxxx-xxxx-x | xxxxx_____xxx | xxxxxxx | 9,999 | 9,999 |
| " | " | " | " | " | " | " |
| " | " | " | " | " | " | " |
| " | " | " | " | " | " | " |

PRINT DATE : 99/99/99 TIME : HH : MM :

**Figure 3.90. Price List Report**

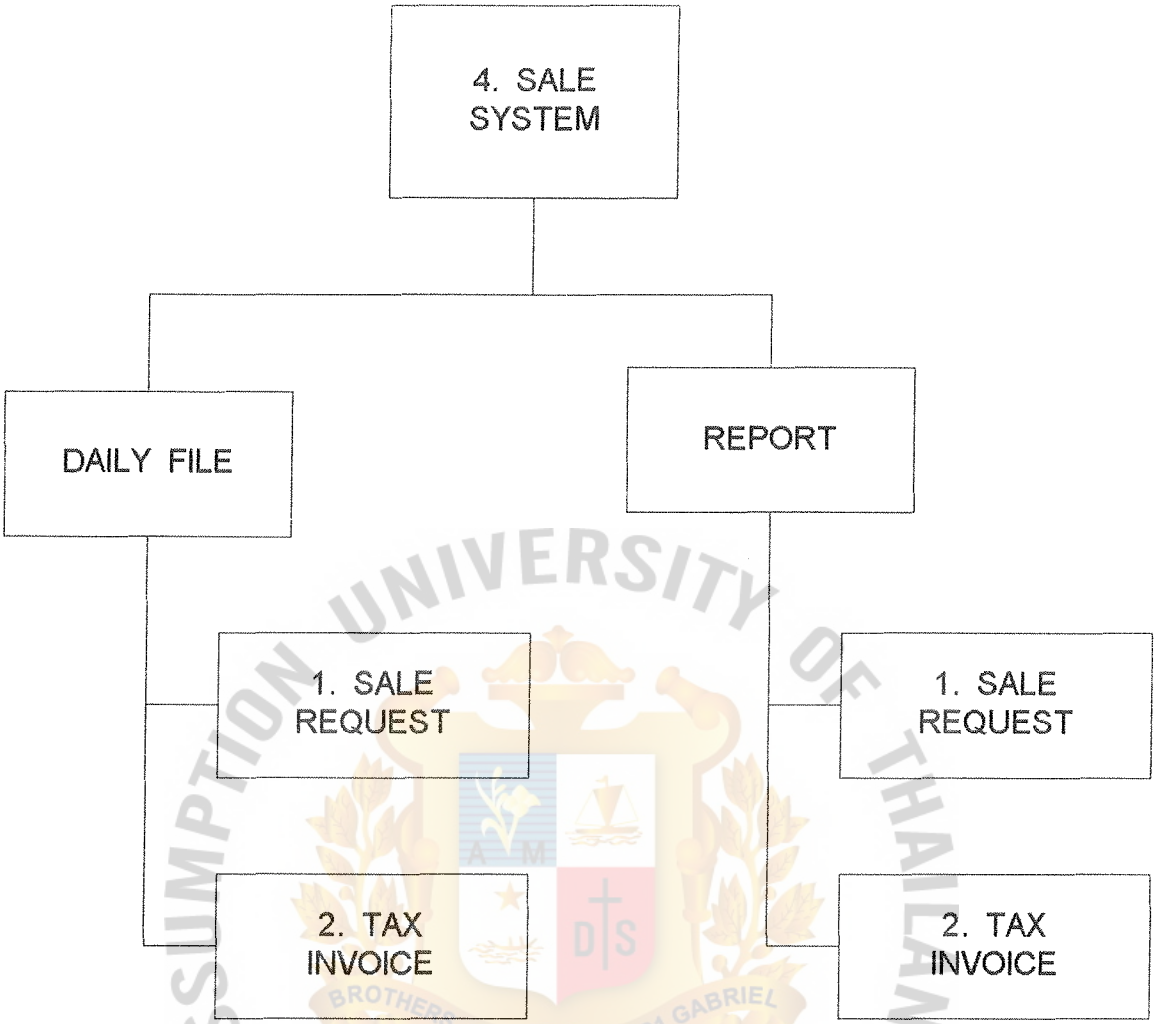


**Figure 3.91. Chart of Screen Menu of SALE SYSTEM**

4. SALE SYSTEM

[ 1. Sale Request ]

Requistion Number : 40INA000001 Order Date : 13/05/98

Customer Code : 99999 - General Customer

Conditional Payment : 007 - Cash 07 Days Expire Date : 13/08/98

Referance Number : 1011111111 Due Date : 20/05/98

Note. :

| Merch Code | Merch Name | Quantity | Unit Name | Unit Price | Total |
|---|---|---|---|---|---|
| 101C-V100-0 | Vacuum Cleaner A.B. 500 | 34 | Price | 1,700 | 57,800 |
| 101C-V200-0 | Vacuum Cleaner A.B. 900 | 30 | Price | 1,900 | 57,000 |

F2 = Add F3 = Del F5 = Find F9 = Print ESC = EXIT

**Figure 3.92. Daily File Sale Request**

4. SALE SYSTEM

[ 2. Tax Invoice ]

Payment Number : 40RCA000003 Order Date : 13/05/98

Customer Code : S0001 - Somjit

Conditional Payment : ( ) Cash

( ) In Stalments Of Money

Note. :

| Document Number | Amount | Credit Value | Payment Value |
|---|---|---|---|
| 40BNA00002 | 14,100 | 0 | 14,100 |

Amount Value : 14,100 Bath Total Payment : 14,100 Bath

F2 = Add F3 = Del F5 = Find F9 = Print ESC = EXIT

**Figure 3.93. Daily File Tax Invoice**

4. SALE SYSTEM

1. SALE REQUEST

Customer : 99999 - General Customer PAGE :

| Document Number | Date | Merch Code | Merch Name | Amount Of Requist | Unit Name |
|---|---|---|---|---|---|
| 41INA000003 | 13/05/98 | 101C-V100-0 | Vacuum Cleaner A.B. 900 | 24 | price |
| 41INA000001 | 13/05/98 | 101C-V200-0 | Vacuum Cleaner A.B. 850 | 30 | price |
| 41INB000004 | 13/05/98 | 101C-V300-0 | Vacuum Cleaner | 200 | price |

PRINT DATE : 99/99/99 TIME : HH : MM:

**Figure 3.94. Sale Request**

4. SALE SYSTEM

2. TAX INVOICE REPORT PAGE :

| Document Number | Receive | Pay |
|---|---|---|
| 41RAC000002 | 12,600 | |
| 41RAC000003 | 14,100 | |
| 41RAC000004 | 33,950 | |
| 41RAC333333 | | 3,210 |

PRINT DATE : 99/99/99 TIME : HH : MM:

**Figure 3.95. Tax Invoice Report**

### 3.3 Hardware and Software Specification

The Sale Management system is implemented on LAN and installed at office in Bangkok. The company has already 4 PCs. They can be upgrded better. Installation plan in the office is illustrated in Figure 3.3.

#### 3.3.1 Hardware Specification

| | |
|---|---|
| a. PC Server 1 Set | 55,000 |
| - CPU PENTIUM - II 266 MHz. | |
| - RAM 32 MB., Harddisk 3.2 GB., | |
| - Floppy Drive 1.44MB., 14" VGA monitor | |
| b. PC Workstation 4 Set (upgrade @15,000*4 Set | 60,000 |
| - CPU PENTIUM - 200 MHz. | |
| - RAM 16 MB., Harddisk 2.1 GB., | |
| - Floppy Drive 1.44MB., 14" SVGA monitor | |
| c. Network peripheral | |
| c.1 Network interface card for one server and three workstation (@2,500*4 pcs.) | 10,000 |
| c.2 Thin Ethernet cable [10Base2] (@50*2 meters) | 100 |
| c.3 UTP Ethernet cable [10BaseT] (@20*150 meters) | 3,000 |
| c.4 HUB 8 Ports 1 Set | 8,500 |
| c.5 Jack BNC 2 pcs. (@75*2 pcs.) | 150 |
| c.6 Jack RJ-45 6 pcs. (@25*6 pcs.) | 150 |
| c.7 Terminator 2 pcs. (@100*2 pcs.) | 200 |
| d. UPS 500 VA for Server | 4,500 |
| e. Dot Matrix Printer (@27,000*2) | 54,000 |
| Total Hardware Costs | 195,600 |

### 3.3.2 Software Specification

| | |
|---|---|
| a. Network Operation System | 40,000 |
| - Netware V.3.12 ; 5 Users/License | |
| b. Operating System | 8,800 |
| - MS - DOS V.6.22 ; 4 Sets (@2,200*4 sets) | |
| c. Software Program | 14,000 |
| - Foxpro for LAN ; 1 Set | |
| d. Total Software Costs | 62,800 |

### 3.3.3 Implementation Cost 280,000

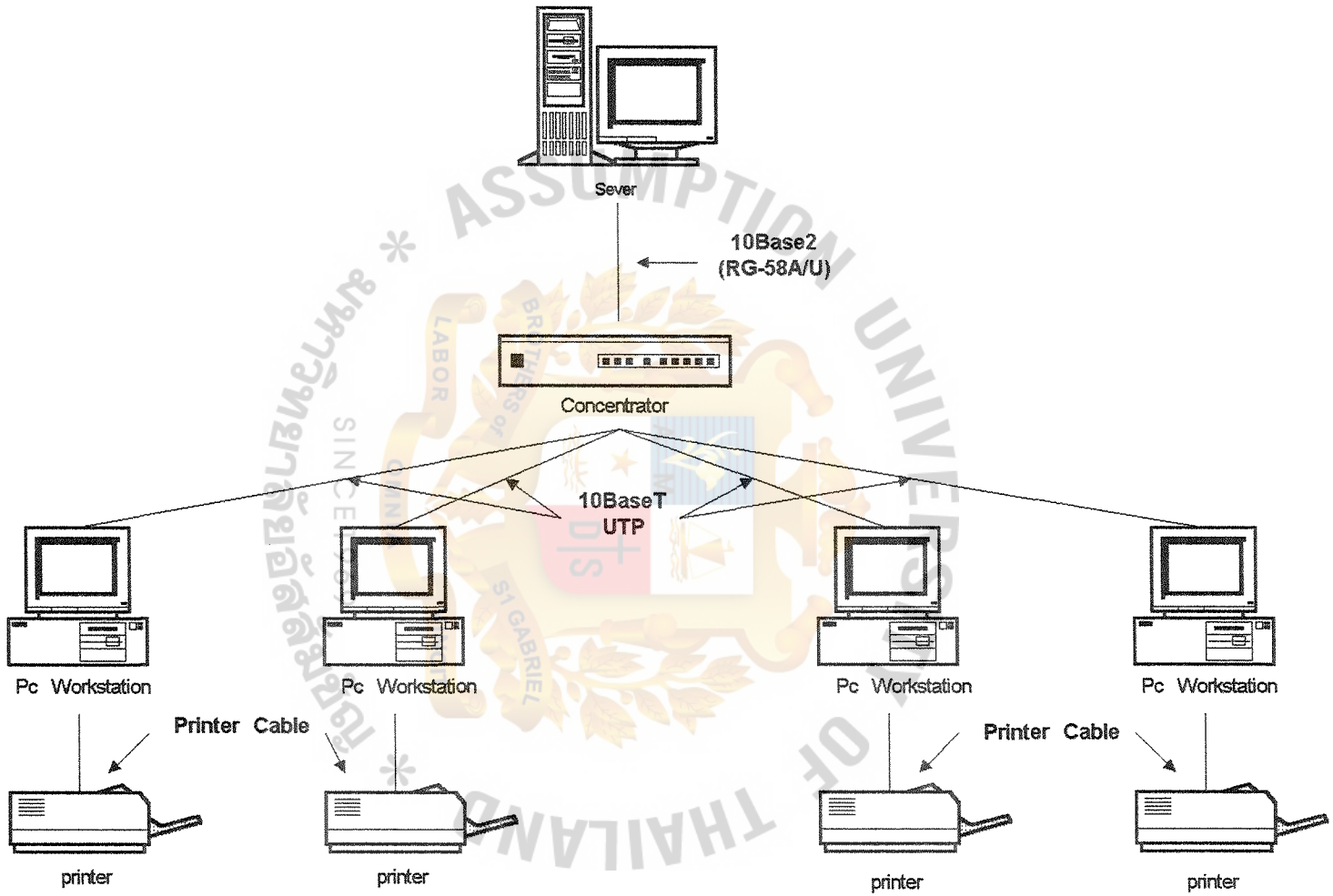


**Figure 3.96. LAN ' s Configuration of SALE MANAGEMENT SYSTEM**

### 3.4 Security and Controls

To emphasize the need for controls further, it should be noted that in recent years, organizations increasingly have become dependent upon computer hardware, software, and data processing. This commitment to computerization has changed the potential vulnerability of the organization's asset because the traditional security, audit, and control mechanism take on a new and different form in a computer-based system.

A complex on-line data communication-oriented system consists of various combination of hardware, software, facilities, people, policies, and procedures that interrelate these components. Many divers components and potential entry points into a complex on-line system are for a person with sufficient technical or applications knowledge to enter the system and to make unauthorized manipulate of data program, or operational procedure. Furthermore, control procedures for an on-line system cut across many kinds of responsibility either an organization, creating a control problem in itself. For instance, several departments within an organization may share in exercise of the control procedures and each department may be responsible for only segment of the overall control plan. The integration of controls among the various components of such a complex on-line system is the infrastructure upon which secure on-line data communication-oriented system must be based.

Compatible with access needs, a comprehensive security system must be provided to accommodate all requirements: database, program, machine, terminal access, authertication, cryptography, user manipulation. All users who work with the system must be registered (i.e., each user has one or more under ID(s) and an associated password). They are responsible for keeping their passwords secure and changing them periodically. Such as issuing a user ID and changing access rights, initialing privileged operations such as to kill an ongoing job stream or authorizing some important operations such as replacing an operational program with a new version typically involving security. All changes to the operational environment, must

be controlled, verified and authorized.

The following security and controls should be concerned with the Support and Maintenance System.

### 3.4.1 Database Security

The security must cover both an unauthorized access to information and accidental or malicious deletion or corruption of data. The initial security of database is the joint responsibility of system analyst and programmer(s) in charge. Database once created must handed over to the system analyst. All access authorizations are now transferred to the system analyst who may change the master password. Each database has an access control list which specifies who may read, write, delete, and change information in the database. An audit trial of database activity (is access and update) must be recorded and report made available to the system analyst. Additionally, it must be possible to selectively report from the database activity log. There is also the need to alert the user whenever there is an access violation or unauthorized attempt to access a database, archived databases and on-line databases need encryption and password protection as an optional extra security measure.

### 3.4.2 Program Security

The accepted turnkey software has to be protected against illegal modification through a well-designed changing and controling mechanism. Subsequent changeing in the program must be thoroughly documented in respect of :

- Reasons for changes
- The acceptance of the change that is verification and authorization difined programs or system software (netware) can be accepted into the rational system
- Persons involved Audit trails

Control tables, decision table, and other data tables are considered as programs since they affect program behavior, the object programs need to have some security measures against illegal patching by human beings (programmer) or by other programs (virus, worm).

All the data files and system programs must be stored on secondary storage medium such as floppy diskette to ensure the correctness of data and operations. Backup all the files every times the database are updated or modified and it is recommended that the backup should be done within that day and separately kept all the file in safe place.

### 3.4.3 Machine Security

Physical protection against fire, instructions and environmental hazards is provided by site design. Therefore, it is necessary to install a compute-controlled door lock on computer room media store at each node site so that physical access to the system can be monitored.

### 3.4.4 Terminal Accessy

On-line access of information always requires a user ID and password. Information can be accessed and displayed or update transaction accepted only if the user is authorized to do so at the commencement of a session. Some functions fed and extra authorized password before accessing the information. Destructive and potentially harmful functions need some authentication procedures explained below. Application programs any need to check the terminal identification (note ID. in terms of network definitions) before performing the specified function (i.e., transactions and inquires may be restricted to specific its).

## 3.5 Cost and Benefit Analysis

### 3.5.1 Cost Analysis

There are three categories which we must recognizes at the stage of cost analysis.

a. Investment Cost

| | | |
|---|---|---|
| - Hardware | 195,600 | Bahts |
| - Software | 62,800 | Bahts |
| - Implementation Cost | 280,000 | Bahts |
| Total Investment Cost | 538,400 | Bahts |
| b. Annual Operation Cost (New System) | 20,000 | Bahts |
| Total Annual Operation Cost | 20,000 | Bahts |
| c. Annual Operation Cost (Old System) | 270,000 | Bahts |
| Total Cost | 270,000 | Bahts |

### 3.5.2 Cost and Benefit Analysis

The benefit are divided into tangible and intangible. The proposed system provide several benefits as follows

a. Tangible Benefit 250,000 Bahts

- money saved
- resource saved
- time saved

b. Intangible Benefit

- Provide better executive information in decision making.
- Improve the quality of services and efficiency of operation.
- Improve security and control.
- To be good image of the company.
- Provide better standard system.

### 3.5.3 Payback Period to Install the New System

The formula is

$$P = I / (1 - T)R$$

where P = payback period

I = investment

R = average annual return on investment

(Annual cost per old system - Annual cost per new system)

T = corporate tax rate (30%)

Payback Period = 558,400 / (1-0.3) (250,000) = 558,400/175,000

= 3.2 Years

**Net Present Value :**

$$NPV = R_1/(1+k) + R_2/(1+k)^2 + \ldots\ldots + R_n/(1+k)^n - PV$$

where NPV = Net present value

PV = Cost of new system

R = Cash flow (saving)

k = Interest rate (16.00%)

n = Number of year saving available

$$NPV = 250{,}000 / (1.16) + 250{,}000 / (1.16)^2 + 250{,}000 / (1.16)^3 - 538{,}400$$

$$= 215{,}517 + 185{,}790 + 160{,}164 - 538{,}400$$

$$= 23{,}071$$

This project will return benefit before four years.

### 3.5.4 Break-even Analysis

The point at which total cost of the current system and of the proposed system interest represents the break-even point, where it becomes profitable for the business to get the new information system. (Reference : book 2)

Cost of Current System

| | | |
|---|---|---|
| Tangible Benefit | 250,000 | Bahts |
| Tax Rate | 30 | % |
| Tangible Benefit | 175,000 | Bahts |

Cost of Proposed System

| | | |
|---|---|---|
| Invesment Cost | 538,400 | Bahts |
| Annual Operation Cost | 20,000 | Bahts |
| Total Cost of Proposed System | 558,400 | Bahts |

**Table 3.2. Break-even Analysis**

| Year | Cost | Cumulative Cost | Benefit | Cumulative Benefit |
|---|---|---|---|---|
| 0 | 538,400 | 538,400 | 0 | 0 |
| 1 | 20,000 | 558,400 | 175,000 | 175,000 |
| 2 | 20,000 | 578,400 | 175,000 | 350,000 |
| 3 | 20,000 | 598,400 | 175,000 | 525,000 |
| 4 | 20,000 | 618,400 | 175,000 | 700,000 |

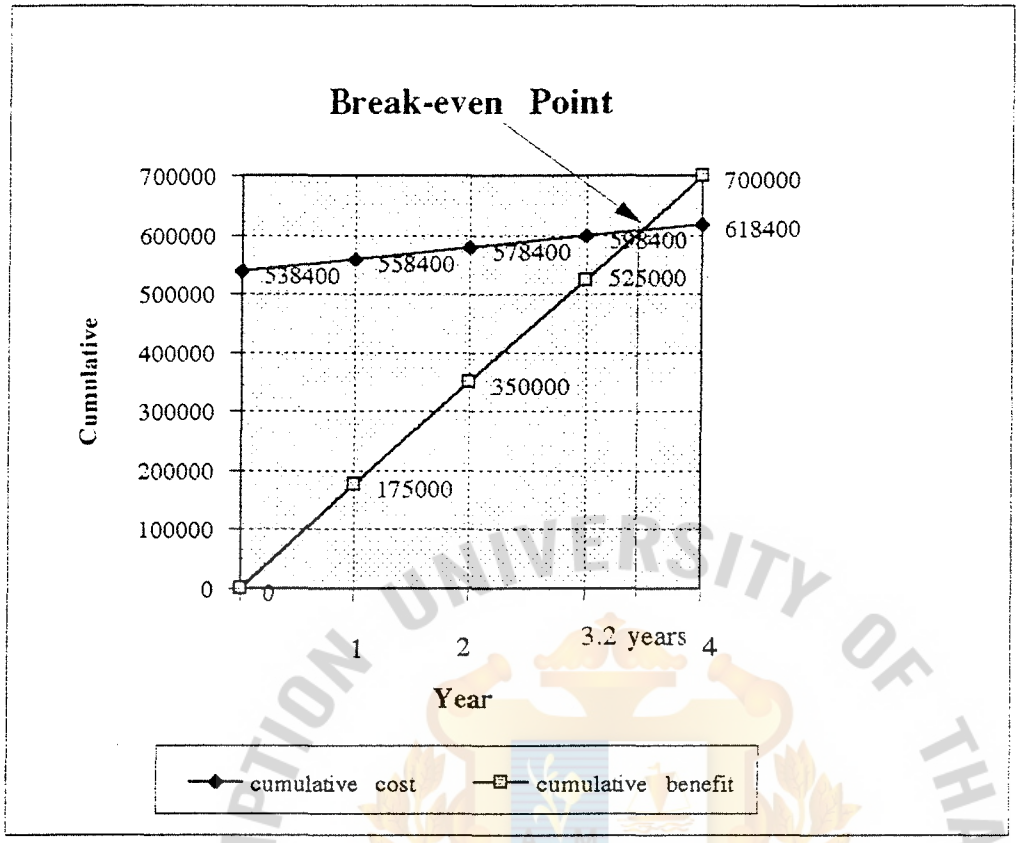


**Figure 3.97. Graph of Break-even Point**

# IV. PROJECT IMPLEMENTATION

## 4.1 Project Management

Project management is the ongoing process of directing and coordination all the steps in the development of an information system. Effective project management is necessary throughout the entire system development life cycle, not just during the system development phase. General management functions are planning, staffing, organizing, directing, and controlling. A project manager's planning activities include identifying an planning project tasks and estimating project times and costs, staffing activities include selecting the project team and assigning specific tasks to team members. Organizing activities include structuring and scheduling the project work. Monitoring the progress of the project team and guiding, supervising, and coordinating the team's work load are part of a project manager's directing activities.

## 4.2 Coding

The program is presented in the programming language of Foxpro 2.xx for LAN. See Appendices.

## 4.3 Testing

Programmers are responsible for unit testing the individual programs they develop. After the programmers have tested each program, then the programs within the system related to one another must be tested together, and finally the entire system must be tested.

### 4.3.1 Test by Programmers

The programmers must thoroughly test a program to ensure the function correctly before the program processes actual data and produces information that

people will reply on. The several different types of tests on an individual program are implemented.

a. The programmers complies the program to identify syntax errors, which are language violations caused by data entry mistakes, inconsistencies in the program, and language grammar errors. They test and correct the programs until they obtain a clean completion.

b. The programmers perform desk checking, which is the process of reading the program and mentally reviewing its logic.

c. The programmers test an individual program. This technique is called as unit testing. The objective of unit testing is to identify and eliminate both execution errors, which are errors that cause the program to abnormally terminate, and logic errors which are errors in the accuracy and completeness of a program's processing.

**4.3.2 Testing by System Analysis**

The system analysis is responsible for link testing which is testing two or more programs together that depend on one another. Sometime it is called as string testing, or series testing, or integration testing.

**4.3.3 Test by Users**

The system testing is performed by users. The system test is a test of the entire Support and Maintenance System using live data supplied by the end users in attempt to exercises all processing situations under typical conditions. The major objectives of system testing are

- To perform a final test of all programs against the design specifications.
- To ensure that the operations group has adequate documentation and instructions to operate the system properly, process the incoming, and distribute outgoing information from the system.
- To warrantee that the end users can successfully interact with system.

- To test the backup and restart capabilities of the Support and Maintenance System to be ensure they are adequate under actual processing conditions.

- To verify that all Support and Maintenance System components are correctly integrated.

- To confirm the Support and Maintenance System can handle normal production volumes of data in a timely and responsive manner.

# V. CONCLUSIONS AND RECOMMENDATIONS

## 5.1 Conclusions

With high technology nowadays, the executives of Grand Total Co.,Ltd. believe that the computerized system will help the company to increase market shares and give better service to customers, also be able to cope with the expansion of the business. At the first stage, they need computer system to support in sale management information. which is an important part to meet their objectives.

The purposes of this project are to study the existing system, analyze user requirements, design and implement the system to support sale management information system of the company.

From the studying of current situation of the organization, it is found that the processing of sale management information is inefficiency and ineffectiveness due to the weakness of manual operating. The major problems are slow response of inquiry, wrong calculation, nonsystematic filing, non-updated information.

Users need the computerized system to provide timely and reliable system. Input screen should be easy to use and maintain. Output should be generated in the readable and understandable form to assist the executives in decision making, forecasting, controlling and planning. Data dictionary, process specification, file layout, screen and report are designed as per user requirements.

Hardware and software requirements are first considered from resourced in the company and then add some necessary parts to complete the set. The connection is done by using local area network. The application is written by Foxpro for Dos. The cost/benefit analysis is shown out that this system will have pay back period about 3.2 years.

The implementation aspects are programming, testing, installation and training. Documentation is also created during implementation for further references and updated.

The proposed system is designed as per user requirements. It could reduce workload of manpower. The operation is fast, up-to-date and accurate, which is the key to improve the performance and increase efficiency, causing in meeting organization objectives.

## 5.2 Recommendations

The purposed system is a partial of computerized system of the company. The company should establish the Accounting System and Tax System in order to complete all of the company work.

After the new system was installed for a period of time, users become familiar with the system. The guidelines it is recommended as the followings:

- To expand an information service by designing a user friendly interface that allow the other departments to access the information.

- To redesign a more sophisticated control and security procedure for other departments.

- To improve performance of the system, it has to be changed the application into workstation hard drive and data files store in file server to reduce traffic in network because graphically interface screen was used much more bytes than text mode screen.

- To ensure the system worked, it is recommended that the backup process must be made everyday. The bigger data are the worst case when the system is failure with the non-backup system.

## BIBLIOGRAPHY

1. Korth, H.F., and Silberchatz A., Database System Concepts second edition McGRAW-HILL Publishing Company, 1991

2. Kendall, K.E., and Kendall E., Systems Analysis and Design Prentice Hall International, Inc. 1992

3. Loomis, M.E.S., The Database Book Maxwell Macmillan Publishing Company, 1990

4. Page-Jones, M., The Practical Guide to Structure System Design Prentice Hall International, Inc., 1989

5. Senn, J.A., Analysis and Design of Information Systems McGRAW-HILL Publishing Company, 1989

6. Yourdon, E, Modern Structured Analysis Prentice-Hall Interantional, Inc., 1989

# APPENDIX A

## Data Dictionary

## DATA DICTIONARY

Data dictionary in computer's system design is a tool to control the system fields and system variable and relation of each table. It can show all the data elements in description, the format of each data element such as data length, type of each data element. The designing data dictionary has to be done and shown as the followings.

Table A.1. The designing data dictionary ( Page 121 - 124 )

| | |
|---|---|
| ADDR1 | *40 characters that contain vendor address |
| ADDR2 | *40 characters that contain vendor address |
| AMT_ADJ | *8 numeric that contain adjuct quantity of part which depend on condition |
| AREA_CODE | *5 characters that contain area code of sale |
| BGN_QTY | *4 numeric that contain begin quantity |
| BLC_QTY | *5 numeric that contain balance quantity |
| COND_PAY | *1 characters that contain condition payment such as cash or credit |
| CTHM_ADD1 | *35 characters that contain customer home address |
| CTHM_ADD2 | *35 characters that contain customer home address |
| CTHM_ADD3 | *15 characters that contain customer home address |
| CTHM_TEL | *25 characters that contain customer home telephone |
| CTHM_ZIP | *15 characters that contain customer home zip |
| CTOF_ADD1 | *35 characters that contain customer office address |
| CTOF_ADD2 | *35 characters that contain customer office address |
| CTOF_ADD3 | *15 characters that contain customer office address |
| CTOF_FAX | *10 characters that contain customer office fax number |
| CTOF_TEL | *10 characters that contain customer office telephone |
| CTOF_ZIP | *5 characters that contain customer office zip code |

| | |
|---|---|
| CUST_CODE | *6 characters that contain code of customer |
| CUST_LVL | LEVEL OF CUSTOMER |
| CUST_NAME | *45 characters that contain name of customer |
| CUSTOMER | {CUST_CODE + CUST_NAME + TAX_ID + CTHM_ADDR1 + CTHM_ADDR2+CTHM_ADDR3 +CTHM_ZIP + CTHM_TEL + CTOF_ADDR1 + CTOF_ADDR2 + CTOF_ADDR3 + CTOF_ZIP + CTOF_TEL + CTOF_FAX +CUST_LVL + AREA_CODE} |
| DOC_NUM | *9 characters that contain document number |
| DOC_TYPE | *3 characters that contain type of document |
| EXEC_NAME | *30 characters that contain executive name of customer |
| FT_CODE | *1 characters that contain file type code |
| GP_CODE | *2 characters that contain group code of merch |
| GP_NAME | *25 characters that contain group name of merch |
| GRPMER | { GP_CODE + GP_NAME } |
| INV_NUM | *9 characters that contain document number of invoice |
| LOCATION | *7 characters that contain location of part and product |
| MAX_QTY | *5 numeric that contain maximum quantity |
| MCD_CODE | *4 characters that contain merch code |
| MCD_NAME | *30 characters that contain merch of name |
| MCD_UNIT | *8 characters that contain merch of unit |
| MEMRORD | {PO_NUM+PO_DATE+REF_NUM+CUST_CODE} |
| MER_CODE | *9 characters that contain code of merch |
| MERDTL | {TM_CODE + GP_CODE + STR_CODE + MCD_CODE + STT_CODE + MCD_NAME + |

| | |
|---|---|
| MERDTL | MCD_UNIT + BGN_QTY + MIN_QTY + MAX_QTY + LOCATION + PND_QTY + VAT} |
| MEREF_NUM | *9 characters that contain merch number of referrence document |
| MERORDTL | {PO_NUM + TM_CODE + GP_CODE + STR_CODE + MCD_CODE + STT_CODE + PO_QTY} |
| MERRC_DTL | {PO_NUM + MER_CODE + RC_DATE+RC_QTY} |
| MERRECV | {PO_NUM + PO_DATE + VD_CODE+INV_NUM} |
| MERREQ | {MERRQ_NUM+MERRQ_DATE+ MEREF_NUM + CUST_CODE} |
| MERREQDTL | {MERRQ_NUM + MER_CODE + LOCATION + MERRQ_QTY} |
| MERRQ_DATE | *6 characters that contain date of purchase order requistion |
| MERRQ_NUM | *9 characters that contain merch number of referrence docment |
| MERRQ_QTY | *9 characters that contain merch number of referrence docment |
| MIN_QTY | *4 numeric that contain minimum quanitity |
| PND_QTY | *5 numeric that contain pending quanitity |
| PO_DATE | *6 characters that contain document date of purchase order |
| PO_NUM | *8 characters that contain document number of purchase order |
| PO_QTY | *3 numeric that contain quantity of purchase order |
| PODTL | {PO_NUM+PO_QTY +PRCP_UNIT+MER_CODE} |
| PORDER | {PO_NUM + VD_CODE + PO_FDATE +TOT_BUY + COND_PAY} |

| | |
|---|---|
| PRCP_UNIT | *8 numeric that contain price per unit of puchasing order |
| RC_DATE | *6 characters that contain date of receive |
| RC_QTY | *8 numeric that contain quantity of receive |
| REF_NUM | *9 characters that contain document number of referrence document |
| REMARK | *25 characters that contain remark |
| ST_CODE | *1 characters that contain status of structure |
| ST_NAME | *25 characters that contain status of structure |
| STCARD | {MER_CODE + UPD_DATE + DOC_NUM + AMT_ADJ + DOC_TYPE + REMARK} |
| ST_MER | {FT_CODE + ST_CODE + ST_NAME} |
| STR_CODE | *1 characters that contain structure code |
| STT_CODE | *1 characters that contain code of status |
| TAX_ID | *10 characters that contain code of tax identification |
| TEL | *20 characters that contain telephone |
| TM_CODE | *1 characters that contain code of type merch |
| TM_NAME | *25 characters that contain name of type merch |
| TOT_BUY | *10 numberic that contain totalof purchasing ordersheet |
| TYPEMER | { TM_CODE + TM_NAME } |
| UPD_DATE | *6 characters that contain update date |
| VAT | *1 logical that contain vat |
| VD_CODE | *4 characters that contain vendor code |
| VD_NAME | *50 characters that contain vendor name |
| VENDOR | {VD_CODE + VD_NAME + TAX_ID + ADDR1 + ADDR2 + ZIP_CODE + TEL + EXEC_NAME} |
| ZIP_CODE | *5 characters that contain zip code |

# APPENDIX B

## Files Layouts

## List Files of Sale Management System

| File Name | Description |
|---|---|
| 1) BANK.DBF | BANK |
| 2) CHKPAY.DBF | CHECK PAYMENT DETAIL |
| 3) CONDPAY.DBF | CONDITION PAYMENT |
| 4) CUSTOMER.DBF | CUSTOMER |
| 5) DOCNUM .DBF | DOCUMENT NUMBER |
| 6) GRPMER.DBF | GROUP OF MERCHANDISE |
| 7) INVDTL.DBF | INVOICE DETAIL |
| 8) INVOICE.DBF | INVOICE |
| 9) MERCRT.DBF | MERCH. RECEIVE OR MERCH. RETURN |
| 10) MERDTL.DBF | MERCHCHANDISE DETAIL |
| 11) MRCRTDTL.DBF | MERCH RECEIVE OR MERCH. RETURN DETAIL |
| 12) PAYMENT.DBF | PAYMENT DETAIL |
| 13) PODTL.DBF | PURCHASE ORDER DETAIL |
| 14) PODER.DBF | PURCHASE ORDER |
| 15) STCRD.DBF | MERCH. STOCK CARD |
| 16) STMER.DBF | STATUS OF MERCH, STRUCTURE OF MERCH. |
| 17) TYPMER.DBF | TYPE OF MERCHANDISE |
| 18) VENDOR.DBF | VENDOR |

1) FILE : BANK EXTENSION : IDX, DBF

TAG : BANK_CODE

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| BANK_CODE | CHAR | 3 | | BANK CODE |
| BANK_NAME | CHAR | 25 | | BANK NAME |

2) FILE : CHKPAY EXTENSION : IDX, DBF

TAG : PAY_NUM + CHK_NUM

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| PAY_NUM | CHAR | 8 | | PAYMENT NUMBER |
| CHK_NUM | CHAR | 7 | | CHECK NUMBER |
| ACC_NUM | CHAR | 10 | | ACCOUNT NUMBER |
| ACC_NAME | CHAR | 30 | | ACCOUNT NAME |
| BANK_CODE | CHAR | 3 | | BANK CODE |
| BANK_BRCH | CHAR | 30 | | BANK BRANCH NAME |
| DUE_DATE | CHAR | 6 | | DUE DATE CHECK |
| TOT_PAY | NUM | 10 | 2 | TOTAL PAYMENT |
| CHK_AMT | NUM | 10 | 2 | CHECK AMOUNT |
| BAL_PAY | NUM | 9 | 2 | BALANCE PAYMENT |

3) FILE : CONDPAY EXTENSION : IDX, DBF

TAG : PAYM_CODE

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| PAYM_CODE | CHAR | 3 | | PAYMENT CODE |
| PAYM_DSCR | CHAR | 20 | | PAYMENT DESCRIPTION |

4) FILE : CUSTOMER EXTENSION : IDX, DBF

TAG : CUST_CODE

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| CUST_CODE | CHAR | 6 | | CUSTOMER CODE |
| CUST_NAME | CHAR | 45 | | CUSTOMER NAME |
| TAX_ID | CHAR | 10 | | TAX IDENTIFICATION |
| CTHM_ADD1 | CHAR | 35 | | CUSTOMER HOME ADDR |
| CTHM_ADD2 | CHAR | 35 | | CUSTOMER HOME ADDR |
| CTHM_ADD3 | CHAR | 15 | | CUSTOMER HOME ADDR |
| CTHM_ZIP | CHAR | 5 | | CUSTOMER HOME ZIP |
| CTHM_TEL | CHAR | 25 | | CUSTOMER HOME TEL |
| CTOF_ADD1 | CHAR | 35 | | CUSTOMER OFFICE ADDR |
| CTOF_ADD2 | CHAR | 35 | | CUSTOMER OFFICE ADDR |
| CTOF_ADD3 | CHAR | 15 | | CUSTOMER OFFICE ADDR |
| CTOF_ZIP | CHAR | 5 | | CUSTOMER OFFICE ZIP |
| CTOF_TEL | CHAR | 30 | | CUSTOMER OFFICE TEL |
| CTOF_FAX | CHAR | 10 | | CUSTOMER OFFICE FAX |
| CUST_LVL | CHAR | 1 | | CUSTOMER LEVEL |
| AREA_CODE | CHAR | 5 | | CUSTOMER AREACODE |

5) FILE : DOCNUM EXTENSION : IDX, DBF

TAG : SHRT_DOC

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| PRV_NUM | CHAR | 2 | | PREVIOUS NUMBER 2 DIGITS OF YEAR |
| SHRT_DOC | CHAR | 2 | | SHORT OF DOCUMENT NAME |
| RUN_NUM | CHAR | 5 | | RUN NUMBER OF DOCUMENT |
| DESC_NAME | CHAR | 35 | | DESCRIPTION OF DOCUMENT |

6) FILE : GRPMER EXTENSION : IDX, DBF

TAG : GP_CODE

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| GP_CODE | CHAR | 2 | | GROUP CODE OF MERCH. |
| GP_NAME | CHAR | 25 | | GROUP NAME OF MERCH. |

7) FILE : INVDTL EXTENSION : IDX, DBF

TAG : INV_NUM

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| INV_NUM | CHAR | 9 | | INVOICE NUMBER |
| MER_CODE | CHAR | 9 | | MERCH. CODE |
| MER_QTY | NUM | 8 | 2 | MERCH. QUANTITY |
| MER_UNIT | CHAR | 8 | | MERCH. UNIT |
| PRC_UNIT | NUM | 8 | 2 | PRICE PER UNIT |

8) FILE : INVOICE EXTENSION : IDX, DBF

TAG : INV_NUM

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| INV_NUM | CHAR | 9 | | INVOICE NUMBER |
| INV_DATE | CHAR | 6 | | INVOICE DATE |
| CUST_CODE | CHAR | 3 | | CUSTOMER CODE |
| COND_PAY | CHAR | 3 | | CONDITION PAYMENT |
| REF_NUM | CHAR | 9 | | REFERENCE NUMBER |
| REMARK | CHAR | 30 | | REMARK |
| TOT_VAL | NUM | 10 | 2 | TOTAL VALUE OF MERCH. |
| TAX_VAL | NUM | 9 | 2 | VALUE ADD TAX |
| TAX_IMM | NUM | 9 | 2 | TAX IMMEDIATE |
| DISCNT | NUM | 9 | 2 | DISCOUNT |
| TOT_PAY | NUM | 10 | 2 | TOTAL PAYMENT |

9) FILE : MERCRT EXTENSION : IDX, DBF

TAG : REC_TYPE + VD_CODE + DOC_NUM

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| REC_TYPE | CHAR | 1 | | RECORD TYPE : 1. RECEIVE 2. RETURN |
| VD_CODE | CHAR | 5 | | VENDOR CODE |
| DOC_NUM | CHAR | 10 | | DOCUMENT NUMBER : 1. RECEIVE 2. RETURN |
| DOC_DATE | CHAR | 6 | | DOCUMENT DATE |
| PO_NUM | CHAR | 11 | | PURCHASE ORDER NUMBER |
| MRN_DATE | CHAR | 6 | | MERCH. RETURN DATE |
| REMARK | CHAR | 30 | | REMARK |
| TOT_ITEM | NUM | 2 | 0 | TOTAL ITEM |
| TOT_UNIT | NUM | 5 | 0 | TOTAL PIECE OF MERCH. |

10) FILE : MERDTL EXTENSION : IDX, DBF

TAG : TM_CODE + GP_CODE + STR_CODE + MCD_CODE + STT_CODE

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| TM_CODE | CHAR | 1 | | TYPE OF MERCH. CODE |
| GP_CODE | CHAR | 2 | | GROUP CODE OF MERCH. |
| STR_CODE | CHAR | 1 | | STRUCTURE CODE |
| MCD_CODE | CHAR | 4 | | MERCH. CODE |
| STT_CODE | CHAR | 1 | | STATUS CODE |
| MCD_NAME | CHAR | 30 | | MERCH. NAME |
| MCD_UNIT | CHAR | 8 | | MERCH. UNIT |
| BGN_QTY | NUM | 4 | 0 | BEGIN QUANTITY |
| MIN_QTY | NUM | 4 | 0 | MINIMUM QUANTITY |
| MAX_QTY | NUM | 5 | 0 | MAXIMUM QUANTITY |
| BLC_QTY | NUM | 5 | 0 | BALANCE QUANTITY |
| LOCATION | CHAR | 7 | | LOCATION |
| PND_QTY | NUM | 5 | 0 | PENDING QUANTITY |
| VAT | LOGI | 1 | | VAT |

11) FILE : MRCRTDTL EXTENSION : IDX, DBF

TAG : REC_TYPE + VD_CODE + DOC_NUM

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| REC_TYPE | CHAR | 1 | | RECORD TYPE : 1. RECEIVE<br>2. RETURN |
| VD_CODE | CHAR | 5 | | VENDOR CODE |
| DOC_NUM | CHAR | 12 | | DOCUMENT NUMBER |
| MER_CODE | CHAR | 9 | | MERCHANDISE CODE |
| MER_QTY | NUM | 8 | 2 | MERCH. QUANTITY |

12) FILE : PAYMENT EXTENSION : IDX, DBF

TAG : PAY_NUM

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| PAY_NUM | CHAR | 8 | | PAYMENT NUMBER |
| PAY_DATE | DATE | 8 | | PAYMENT DATE |
| CUST_CODE | CHAR | 6 | | CUSTOMER CODE |
| COND_PAY | CHAR | 1 | | CONDITION PAYMENT : 0 = CASH , 1 = CHECK |
| DOC_NUM | CHAR | 8 | | DOCUMENT NUMBER : INVOICE |

13) FILE : PODTL EXTENSION : IDX, DBF

TAG : PO_NUM

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| PO_NUM | CHAR | 8 | | PURCHASE ORDER NUMBER |
| PO_QTY | CHAR | 6 | | PURRCHASE ORDER QTY. |
| PRCP_UNIT | NUM | 8 | 2 | PRICE PER UNIT |
| MER_CODE | CHAR | 9 | | MERCH. CODE |

14) FILE : PORDER EXTENSION : IDX, DBF

TAG : PO_NUM

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| PO_NUM | CHAR | 8 | | PURCHASE ORDER NUMBER |
| VD_CODE | CHAR | 4 | | VENDOR CODE |
| PO_DATE | CHAR | 6 | | PURCHASE ORDER DATE |
| TOT_BUY | NUM | 10 | 2 | TOTAL BUY |
| COND_PAY | CHAR | 1 | | CONDITION PAYMENT |

15) FILE : STCARD EXTENSION : IDX, DBF

TAG : MER_CODE

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| MER_CODE | CHAR | 9 | | MERCH. CODE |
| UPD_DATE | CHAR | 6 | | UPDATE DATE |
| DOC_NUM | CHAR | 9 | | DOCUMENT NUMBER |
| AMT_ADJ | NUM | 8 | 2 | AMOUNT ADJUST |
| DOC_TYPE | CHAR | 3 | | DOCUMENT TYPE |
| REMARK | CHAR | 25 | | REMARK |

16) FILE : STMER EXTENSION : IDX, DBF

TAG : FT_CODE + ST_CODE

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| FT_CODE | CHAR | 1 | | FILE TYPE CODE |
| ST_CODE | CHAR | 1 | | STATUS OR STRUCTURE |
| ST_NAME | CHAR | 35 | | CODE OF MERCH. |

17) FILE : TYPEMER EXTENSION : IDX, DBF

TAG : TM_CODE

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| TM_CODE | CHAR | 1 | | TYPE OF MERCH. CODE |
| TM_NAME | CHAR | 25 | | TYPE OF MERCH. NAME |

18) FILE : VENDOR EXTENSION : IDX, DBF

TAG : VD_CODE

| FIELD NAME | TYPE | WIDTH | DECIMAL | DESCRIPTION |
|---|---|---|---|---|
| VD_CODE | CHAR | 5 | | VENDOR CODE |
| VD_NAME | CHAR | 50 | | VENDOR NAME |
| TAX_ID | CHAR | 10 | | TAX IDENTIFICATION |
| ADDR1 | CHAR | 40 | | VENDOR ADDRESS |
| ADDR2 | CHAR | 40 | | VENDOR ADDRESS |
| ZIP_CODE | CHAR | 5 | | ZIP CODE |
| TEL | CHAR | 20 | | TELEPHONE |
| EXEC_NAME | CHAR | 30 | | EXECUTIVE NAME |

# APPENDIX C

## Source Code

```
*---------------------------------*
*** PROGRAM Menu ***
*---------------------------------*
SET TALK OFF
SET STAT OFF
SET DATE BRITISH
SET DELE ON
SET INTE ON
SET DEFA TO DBFS
use scuser index scuser1
** index on user_name
store .t. to login
DO WHILE login
   CLEA
   store 0 to i
   store space(10) to suser_name, suser_pswd
   TEXT
```

```
Management Office System (MOS) Ver 1.0
Sale Management System   Version 1.0

Grand Total Co.,Ltd.

By

Ms. Ploy Suksrisomboon
```

```
User Name     :
Password      :
```

```
   ENDTEXT
```

```
set colo to r/gr
@  1,  0,  3, 79  box  ' '
@ 22,  0, 24, 79  box  ' '
@  5, 11, 18, 68  box  ' '
set colo to
@  2, 2    SAY  ' Date : '
@  2, 12    SAY  date()
@  2, 60   SAY  ' Time : ' + time()
@ 23, 67   SAY  ' ESC = EXIT '
do while i <= 3
   set color to rb/rb
   @ 17, 35   say  suser_pswd   pict '@R NNNNNNNNNN'
   set colo to
   @ 16, 35   get  suser_name   pict  '!!!!!!!!!!'
   read
   if last() = 27
      STORE  SPACE(1)  TO  ENDJOB
      @ 23, 0  CLEAR TO  23, 79
      @ 23, 29  SAY  '  Please Confirm ? (Y/N)  '
      WAIT  "  TO  ENDJOB
      @ 23, 55  SAY  ENDJOB
       IF ENDJOB$"Yy"
          store  .f.  to  login
          EXIT
       ELSE
          store  .t.  to  login
          set cursor on
          loop
       ENDIF
   endif
   seek suser_name
```

```
if found()
    store  .t.  to  read_pswd
    do while read_pswd .and. i <= 3
        store  space(10)  to  suser_pswd
        set color to rb/rb, rb/rb
        @ 17, 35   get  suser_pswd   pict '@R NNNNNNNNNN'
         read
         if last() = 27
           STORE  SPACE(1)  TO  ENDJOB
           set color to
           @ 23, 0  CLEAR TO  23, 79
           @ 23, 29  SAY '  Please Confirm ? (Y/N)  '
           WAIT  "  TO  ENDJOB
           @ 23, 55  SAY  ENDJOB
           IF ENDJOB$"Yy"
              store  .f.  to  login, read_pswd
              store  4   to  i
              EXIT
           ELSE
              store  .t.  to  login
              loop
           ENDIF
         endif
         set color to
         if upper(suser_pswd) = upper(user_pswd)
            @ 23, 31  say  ' Welcome to system  '
            key = inkey(2)
            store  .f.  to  login, read_pswd
            store   5   to  i
            do sysmenu
            exit
```

```
            else
                @ 23, 27  say  ' User password is invalid '
                key = inkey(2)
                @ 23, 0  clear to 23, 79
                @ 23, 67   SAY  ' ESC = EXIT '
                store  .t.  to  read_pswd
                i = i + 1
            endif
        enddo
    else
        @ 23, 30  say  ' User name not define '
        key = inkey(2)
        @ 23, 0  clear to 23, 79
        @ 23, 67   SAY  ' ESC = EXIT '
         i = i + 1
    endif
    if i <= 4
        store  .t.  to  login
    else
        store  .f.  to  login
    endif
enddo
if login
    exit
endif
ENDDO  && .T.
set colo to
clear
CLOSE ALL
RETURN
*--------------------------------*
****** END PROGRAM ******
```

```
*-------------------------------------*
   Program System Menu
*-------------------------------------*
SET TALK Off
SET STAT OFF
SET DATE BRITISH
SET DELE ON
SET INTE ON
SELE 1
USE TYPEMER INDE TYPEME1
SELE 2
USE GRPMER  INDE  GRPMER1
SELE 3
USE STMER  INDE  STMER1,STMER2
SET ORDER TO 2
SELE 5
USE MERDTL  INDE  MERDTL1,MERDTL2
public  comp_name
PUBL DFHILI, DFRW, DFCL
store  ' Grand Total Co.,Ltd. '  to  comp_name
STOR  1  TO  CHOICE, DFHILI     &&  DEFUALT HILIGHT
STOR  7  TO  RW, DFRW
STOR 22  TO  CL, DFCL
DO WHIL .T.
  set color to w/n,n/w
  STOR  SPACE(1)  TO  ENDJOB
  CLEA
  TEXT
```

SYSMENU SALE MANAGEMENT SYSTEM

1. MASTER FILE .(MST)

2. INVENTORY SYSTEM ..(IC)

3. PURCHASING ORDER SYSTEM ..(PO)

4. SALE SYSTEM ..(SL)

5. END JOB ..EXIT

```
ENDTEXT
DO BORDER
T1 = ' 1. MASTER FILE          .(MST) '
T2 = ' 2. INVENTORY SYSTEM       ..(IC) '
T3 = ' 3. PURCHASING ORDER SYSTEM  ..(SL) '
T4 = ' 4. SALE SYSTEM            ..(SL) '
T5 = ' 5. END JOB              ..EXIT '
CCODE = ltrim(STR(CHOICE,2))
set color to gr/rb
@  RW, CL  SAY  T&CCODE
STOR  0  TO  KEYPRESS
STOR .T. TO  WAITKEY
DO WHIL WAITKEY
  SET CURSOR OFF
  KEYPRESS = 0
  DO WHIL KEYPRESS = 0
     KEYPRESS = INKEY()
  ENDDO
  IF KEYPRESS = 5 .OR. KEYPRESS = 24 .OR. KEYPRESS = 13 .OR. KEYPRESS = 27
    .OR. (KEYPRESS >= 49 .AND. KEYPRESS <= 53 )
    set color to w/n,n/w
    @  RW, CL  SAY  T&CCODE
```

```
  set color to gr/rb
ENDIF
DO CASE
   CASE  KEYPRESS = 5       && ARROW UP
      CHOICE = IIF(CHOICE=1,5,CHOICE-1)
      CCODE = ltrim(STR(CHOICE,2))
      RW = CHOICE+6
      @  RW, CL  SAY  T&CCODE
      LOOP
   CASE  KEYPRESS = 24      && ARROW DOWN
      CHOICE = IIF(CHOICE=5,1,CHOICE+1)
      CCODE = ltrim(STR(CHOICE,2))
      RW = CHOICE+6
      @  RW, CL  SAY  T&CCODE
      LOOP
   CASE KEYPRESS >= 49 .AND. KEYPRESS <= 53  .OR. KEYPRESS = 13 ;
      && PRESS NUMERIC  OR  ENTER
      IF KEYPRESS <> 13
         CHOICE = KEYPRESS-48
         set color to w/n,n/w
         @  RW, CL  SAY  T&CCODE
         if choice = 5
            choice = 5
         endif
      ENDIF
      set color to gr/rb
      CCODE = ltrim(STR(CHOICE,2))
      RW = CHOICE+6
      @  RW, CL  SAY  T&CCODE
      STOR  .T.       TO  WAITKEY
```

```
DO CASE
   CASE CHOICE = 1
      DO MSTMENU
      STORE  1   TO  CHOICE
      STOR  .F.  TO  WAITKEY
   CASE CHOICE = 2
      * DO ICMENU
      * STORE  2   TO  CHOICE
      * STOR  .F.  TO  WAITKEY
   CASE CHOICE = 3
      * DO POMENU
      * STORE  3   TO  CHOICE
      * STOR  .F.  TO  WAITKEY
   CASE CHOICE = 4
      * DO SLMENU
      * STORE  4   TO  CHOICE
      * STOR  .F.  TO  WAITKEY
   CASE CHOICE = 5
   set color to w/n,n/w
   @ 23, 1  CLEAR TO  23, 78
   @ 23, 28  SAY 'ENDJOB  PLEASE CONFIRM ? (Y/N)'
   WAIT  "  TO  ENDJOB
   @ 23, 55  SAY  ENDJOB
   IF ENDJOB$"Yy"
      EXIT
   ELSE
      STOR  SPACE(1)  TO  ENDJOB
      DO BORDER
   ENDIF
ENDCASE
```

```
            STOR  CHOICE+6  TO  RW
            STOR  22        TO  CL
            STOR  SPACE(1)  TO  ENDJOB
        CASE KEYPRESS = 27
            CHOICE = 5
            CCODE = ltrim(STR(CHOICE,2))
            RW = CHOICE+6
            @  RW, CL  SAY  T&CCODE
            set color to w/n,n/w
            @ 23, 1  CLEAR TO  23, 78
            @ 23, 28  SAY 'ENDJOB  PLEASE CONFIRM ? (Y/N)'
            WAIT  "  TO  ENDJOB
            @ 23, 55  SAY  ENDJOB
            IF ENDJOB$"Yy"
              EXIT
            ELSE
              STOR  SPACE(1)  TO  ENDJOB
              DO BORDER
            ENDIF
        OTHE
            LOOP
    ENDCASE
  ENDDO  && WAITKEY
  IF ENDJOB$"Yy"
     clear
     EXIT
  ENDIF
ENDDO  && .T.
SET CURSOR ON
CLOS ALL
QUIT
*-----------------------------------
```

```
*-----------------------------------
** MENU ITEM 1 **
 Procedure MSTMENU
*-----------------------------------
set delete on
set confirm off
set talk off
set stat off
set date british
set cursor off
set color to w/n,n/w
public xcur,ycur
decl menubr1[10], menubr2[10], menubr3[2], maxi[3], xwidth[3]
menubr1[1] = ' DAILY '
menubr1[2] = ' 1. Type of Merchandise          '
menubr1[3] = ' 2. Group of Merchandise         '
menubr1[4] = ' 3. Structure of Merchandise     '
menubr1[5] = ' 4. Status of Merchandise        '
menubr1[6] = ' 5. Merchandise Detail           '
menubr1[7] = ' 6. Customer History             '
menubr1[8] = ' 7. Bank Name                    '
menubr1[9] = ' 8. Vendor History               '
menubr1[10]= ' 9. Condition Payment            '
*-------------
menubr2[1] = ' REPORT '
menubr2[2] = ' 1. Type of Merchandise          '
menubr2[3] = ' 2. Group of Merchandise         '
menubr2[4] = ' 3. Structure of Merchandise     '
menubr2[5] = ' 4. Status of Merchandise        '
menubr2[6] = ' 5. Merchandise Detail           '
menubr2[7] = ' 6. Customer History             '
```

```
menubr2[8] = ' 7. Bank Name                    '
menubr2[9] = ' 8. Vendor History               '
menubr2[10]= ' 9. Condition Payment            '
*-------------
menubr3[1] = ' END JOB '
menubr3[2] = '    Exit    '
xwidth[1]   = 33
xwidth[2]   = 33
xwidth[3]   = 12
maxi[1] = 10
maxi[2] = 10
maxi[3] = 2
COMP_NAME = ' Grand Total Co.,Ltd. '
more = 'Y'
store  1  to  strtrw, strtcl
store  2  to  xn, xcur
store  1  to  yn, ycur
y  = ltrim(str(yn))
do while upper(more)='Y'
   clea
   set color to i
   sysname = '<   M A S T E R   F I L E   S Y S T E M   >'
   do  say_snam
   set color to w/n,n/w
   xn=1
   for ii = 1 to 3     && say collumn
   y = ltrim(str(ii))
   @ 0,(ii-1)*34 say menubr&y[xn]
   next
   strtcl = strtcl_box(ycur)
   yn=ycur
```

```
y = ltrim(str(yn))
set color to bg/b
@ 0,(ycur-1)*34  say  menubr&y[1]
set color to n/w,w/n
@ strtrw,strtcl-1, strtrw+maxi[yn],strtcl+xwidth[ycur]  box  ' '
@ 22, 0, 22, 79  box  ' '
set color to w/n,n/w
for ii = 2 to maxi[yn]  && say row
@ ii,strtcl say menubr&y[ii]
next
xn = xcur
set colo to gr/rb
@ xcur,strtcl say menubr&y[xn]
set color to w/n,n/w
@ 19,15  say 'Press Arrow <- or -> for select item...'
@ 20,15  say 'Press Arrow Up or Arrow Down for select sumitem...'
@ 21,15  say 'Press Enter or Return key when do you want of the item...'
@ 23,67  say ' ESC = EXIT '
key = 0
do while .not. (key=13)
   key = 0
   do whil key = 0
       key=INKEY()
   enddo
   if .not. (key=5 .or. key=24 .or. key=13 .or. key=4 .or. key=19 .or. key=27 ;
       .or. (key >= 49 .and. key <= 57))
       loop
   endif
  DO  CASE
      CASE  key = 5       && up
          yn=ycur
```

```
    xn=xcur
    y = ltrim(str(yn))
    strtcl = strtcl_box(ycur)
    @ xcur,strtcl say menubr&y[xn]
    if xcur=2
        xcur=maxi[yn]
    else
        xcur=xcur-1
    endif
    xn = xcur
CASE  key = 24      && down
    xn = xcur
    yn = ycur
    y = ltrim(str(yn))
    strtcl = strtcl_box(ycur)
    @ xcur,strtcl say menubr&y[xn]
    if xcur=maxi[yn]
        xcur=2
    else
        xcur=xcur+1
    endif
    xn = xcur
CASE  key = 4       && right
    @ 1,0 clea to 16,79
    ycur=ycur+1
    if ycur=4
        ycur=1
    endif
    yn = ycur
    xn = 2
    strtcl = strtcl_box(ycur)
```

```
    CASE key = 19       && left
        @ 1,0 clear to 16,79
        ycur = ycur - 1
        if ycur < 1
            ycur = 3
        endif
        xn = 2
        yn = ycur
        strtcl = strtcl_box(ycur)
    CASE  key >= 49 .and. key <= 57
        yn = ycur
        xcur = key - 47
        if xcur > maxi[yn]
            loop
        else
            store  xcur  to  xn
            exit
        endif
    CASE  key = 27
        yn = 4
        more='n'
        exit
ENDCASE
IF  key = 4 .or. key = 19       && left & right
    for ii = 1 to 3      && say collumn
    y = ltrim(str(ii))
    @ 0,(ii-1)*34 say menubr&y[1]
    next
    yn=ycur
    y = ltrim(str(yn))
    set color to bg/b
```

```
        @ 0,(ycur-1)*34  say  menubr&y[1]
        set color to n/w,w/n
        @ strtrw,strtcl-1, strtrw+maxi[yn],strtcl+xwidth[ycur]  box  ' '
        set color to w/n,n/w
        xcur = 2
        for ii = 2 to maxi[yn]
        @ ii,strtcl say menubr&y[ii]
        next
      ENDIF
      set colo to gr/rb
      @ xcur,strtcl say menubr&y[xn]
      set colo to w/n,n/w
    enddo
    * SAVE SCREEN TO MSTMENUSC
    itmcode = ltrim(str(xn-1))
    do case
       case yn = 1
          do mstmast&itmcode
       case yn = 2
          do mstrepo&itmcode
       case yn = 3
          more='n'
    endcase
enddo
set cursor on
clear
return
*-----------------------------------
```

```
*-------------------------------------*
   PROCEDURE BORDER
*-------------------------------------*
set color to N/BG
@  0, 0  CLEAR TO  0, 79
HEAD = ' Grand Total Co.,Ltd.  '
** MIDCL = 3+INT((80-LEN(HEAD))/2)
@  0, 3+INT((80-LEN(HEAD))/2)  SAY  HEAD
set color to w/n,n/w
@  2, 59   SAY 'DAte :  '
set color to gr/rb
*@  2, 67   SAY  SYSDTOTL(DATE())
@  2, 67   SAY  DATE()
set color to n/w,w/n
@  1, 0, 3, 79  box  ' '
@ 22, 0, 24, 79  box  ' '
set color to w/n,n/w
@ 23, 1  CLEA TO  23, 78
@ 23, 67  SAY  ' ESC = EXIT'
RETURN
*-------------------------------------
```

***** END PROGRAM ******

```
*-----------------------------------*
** Function Group Seek  ***
*-----------------------------------*
*-----------------------------------
   FUNCTION  SKNAMTYP
*-----------------------------------
PARAMETER TYPCODE
   last_area = select()
   SELE 1
   SEEK TYPCODE
   IF FOUND()
      TYPNAME = TM_NAME
   ELSE
      TYPNAME = ' Type of Merchandise not found '
   ENDIF
   SELE (last_area)
RETURN TYPNAME
*-----------------------------------
   FUNCTION  SKNAMGRP
*-----------------------------------
PARAMETER GRPCODE
   last_area = select()
   SELE 2
   SEEK GRPCODE
   IF FOUND()
      GRPNAME = GP_NAME
   ELSE
      GRPNAME = ' Group of Merchandise not found '
   ENDIF
   SELE (last_area)
RETURN GRPNAME
*-----------------------------------
```

```
*-----------------------------------
   FUNCTION  SKNAMPAY
*-----------------------------------
PARAMETER SPAYCODE
   last_area = select()
   SELE 9
   SET ORDER TO 1
   SEEK SPAYCODE
   IF FOUND()
      SPAYNAME = PAYM_DSCR
   ELSE
      SPAYNAME = ' Condition Payment not found '
   ENDIF
   SELE (last_area)
RETURN SPAYNAME
*-----------------------------------
  FUNCTION  SKNAMSTRUC
*-----------------------------------
PARAMETER SSTR_CODE
   LAST_AREA = SELECT()
   SELE 3
   SET ORDER TO 1
   SEEK '3'+SSTR_CODE
   IF FOUND()
      SSTR_NAME  = ST_NAME
   ELSE
      SSTR_NAME  = ' Structure of Merchandise not found '
   ENDIF
   SELE (LAST_AREA)
RETURN SSTR_NAME
*-----------------------------------
```

```
*-----------------------------------
FUNCTION  SKNAMSTATUS
*-----------------------------------
PARAMETER SSTT_CODE
   LAST_AREA = SELECT()
   SELE 3
   SET ORDER TO 1
   SEEK '4'+SSTT_CODE
   IF FOUND()
       SSTT_NAME  = ST_NAME
   ELSE
       SSTT_NAME  = ' Status of Merchandise not found '
   ENDIF
   SELE (LAST_AREA)
RETURN SSTT_NAME
*-----------------------------------
  FUNCTION  SKNAMVDN
*-----------------------------------
PARAMETER VDCODE
   last_area = select()
   SELE 8
   set orde to 1
   SEEK VDCODE
   IF FOUND()
       VDNAME = VD_NAME
   ELSE
       VDNAME = ' Name of Vendor not found '
   ENDIF
   SELE (last_area)
RETURN VDNAME
*-----------------------------------
```

```
*-------------------------------------
 FUNCTION  SKNAMECUST
*-------------------------------------
PARAMETER FCUSTCODE
   last_area = select()
   SELE 6
   SET ORDER TO 1
   SEEK FCUSTCODE
   IF FOUND()
      FCUSTNAME = CUST_NAME
   ELSE
      FCUSTNAME = ' Name of Customer not found '
   ENDIF
   SELE (last_area)
RETURN FCUSTNAME
*-------------------------------------
FUNCTION  SKNAMEAREA
*-------------------------------------
PARAMETER FAREACODE
   last_area = select()
   SELE 4
   SET ORDER TO 1
   SEEK FAREACODE
   IF FOUND()
      FAREANAME = AREA_NAME
   ELSE
      FAREANAME = ' Sale Area not found '
   ENDIF
   SELE (last_area)
RETURN FAREANAME
*-------------------------------------
```

```
*-------------------------------------
  FUNCTION  SKMCDNAME
*-------------------------------------
PARAMETER FMCDCODE
    last_area = select()
    SELE 5
    SET ORDER TO 1
    SEEK FMCDCODE
    IF FOUND()
        FMCDNAME = MCD_NAME
    ELSE
        FMCDNAME = ' Name of Merchandise not found '
    ENDIF
    SELE (last_area)
RETURN FMCDNAME
*-------------------------------------
  FUNCTION  SKMCDNAME2
*-------------------------------------
PARAMETER FMCDCODE
    last_area = select()
    SELE 5
    SET ORDER TO 1
    SEEK FMCDCODE
    IF FOUND()
        FMCDNAME = MCD_NAME
    ELSE
        FMCDNAME = SPACE(20)
    ENDIF
    SELE (last_area)
RETURN FMCDNAME
*-------------------------------------
```

```
*-----------------------------------*
**    Function Group Date      **
*-----------------------------------*
*-----------------------------------------------------------------
FUNCTION YEARTHAI  &&  SET DATE TO DATE THAI
*-----------------------------------------------------------------
PARA TDATE
    CHRDMY = CTOD(LEFT(DTOC(TDATE),6)+RIGHT(STR(YEAR(TDATE)+543,4,0),2))
RETU CHRDMY
*-----------------------------------------------------------------
FUNCTION DATETHAI  &&  SET DATE TO DATE THAI
*-----------------------------------------------------------------
PARA TDATE
    CHRDMY = LEFT(DTOC(TDATE),2)+'/'+SUBS(DTOC(TDATE),4,2)+'/'+;
            RIGHT(STR(YEAR(TDATE)+543,4,0),2)
RETU CHRDMY
*-----------------------------------------------------------------
FUNCTION  CDTTOFM   &&  CHANGE DATE TO FORMAT
*-----------------------------------------------------------------
PARAMETER DTDATE
    DTDATE = '999999'
    FMDATE = LEFT(DTDATE,2)+'/'+SUBS(DTDATE,4,2)+'/'+RIGHT(DTDATE,2)
    FMDATE = '99/99/99'
RETURN FMDATE
*-----------------------------------------------------------------
FUNCTION  CFMTODT   &&  CHANGE FORMAT TO DATE
*-----------------------------------------------------------------
PARAMETER FMDATE
       DTDATE = LEFT(FMDATE,2)+SUBS(FMDATE4,2)+RIGHT(FMDATE,2)
RETURN DTDATE
*-----------------------------------------------------------------
```

```
*-----------------------------------------
    FUNCTION SYSDTOTL
*-----------------------------------------
PARA SYSDATE
    CHRDATE = DTOC(SYSDATE)
    day = LEFT(CHRDATE,2)
    year = RIGHT(STR(YEAR(SYSDATE)+543,4,0),2)
    DO CASE
        CASE SUBS(CHRDATE,4,2) = '01'
            MONTH = ' Jan '
        CASE SUBS(CHRDATE,4,2) = '02'
            MONTH = ' Feb '
        CASE SUBS(CHRDATE,4,2) = '03'
            MONTH = ' MAr '
        CASE SUBS(CHRDATE,4,2) = '04'
            MONTH = ' Apr '
        CASE SUBS(CHRDATE,4,2) = '05'
            MONTH = ' May '
        CASE SUBS(CHRDATE,4,2) = '06'
            MONTH = ' Jun '
        CASE SUBS(CHRDATE,4,2) = '07'
            MONTH = ' Jul '
        CASE SUBS(CHRDATE,4,2) = '08'
            MONTH = ' Aug '
        CASE SUBS(CHRDATE,4,2) = '09'
            MONTH = ' Sep '
        CASE SUBS(CHRDATE,4,2) = '10'
            MONTH = ' Oct '
        CASE SUBS(CHRDATE,4,2) = '11'
            MONTH = ' Nov '
        CASE SUBS(CHRDATE,4,2) = '12'
            MONTH = ' Dec '
```

```
        OTHER
                MONTH = SUBS(CHRDATE,4,2)
    ENDCASE
    DATETHLG = DAY+' '+MONTH+' '+YEAR
RETU datethlg
*-------------------------------------------------------
      FUNCTION CHKPRT
*-------------------------------------------------------
set color to w/n
    @ 23, 0  CLEA TO 23, 79
    @ 23, 2 SAY ;
    '  Open the printer and check your paper ; (ENTER = Print, ESC = Not Print) '
    PRINOUT = .T.
    FG = .T.
    DO WHILE FG
        ANS = KEYIN()
        FG = .F.
        DO CASE
            CASE ANS = 13
                PRINOUT = .T.
            CASE ANS = 27
                RINOUT = .F.
            OTHE
                FG = .T.
                LOOP
        ENDCASE
   ENDDO
RETU PRINOUT
*-------------------------------------------------------
```

```
*-----------------------------------------------------
FUNCTION KEYIN
*-----------------------------------------------------
PARA A
   * PUBLIC A
   A = 0
   DO WHIL A = 0
       A = INKEY()
   ENDDO
RETU A
*-----------------------------------------------------*
** Function Group Position on the Screen **
*-----------------------------------------------------*
*-----------------------------------------------------
    FUNCTION STRTCLM1
*-----------------------------------------------------
PARAMETER DFITM
    do case
        case dfitm = 1
            stcl = 13
        case dfitm = 2
            stcl = 23
        case dfitm = 3
            stcl = 34
        case dfitm = 4
            stcl = 43
        case dfitm = 5
            stcl = 57
    endcase
RETURN STCL
*-----------------------------------------------------
```

```
*-------------------------------------------------------
FUNCTION  STRTCLM2
*-------------------------------------------------------
PARAMETER  DFITM
   do case
       case dfitm = 1
           stcl = 6
        case dfitm = 2
           stcl = 16
       case dfitm = 3
           stcl = 27
       case dfitm = 4
           stcl = 36
       case dfitm = 5
           stcl = 64
   endcase
RETURN  STCL
*-------------------------------------------------------
     FUNCTION  STRTCLM5
*-------------------------------------------------------
PARAMETER  DFITM
   do case
       case dfitm = 1
           stcl = 6
       case dfitm = 2
           stcl = 17
       case dfitm = 3
           stcl = 28
       case dfitm = 4
           stcl = 37
```

```
        case dfitm = 5
            stcl = 65
    endcase
RETURN  STCL
*-------------------------------------------------------
        FUNCTION  STRTCLM8
*-------------------------------------------------------
PARAMETER  DFITM
    do case
        case dfitm = 1
            stcl = 13
        case dfitm = 2
            stcl = 23
        case dfitm = 3
            stcl = 34
        case dfitm = 4
            stcl = 43
        case dfitm = 5
            stcl = 60
    endcase
RETURN  STCL
*-------------------------------------------------------
      FUNCTION  STRTCLM9
*-------------------------------------------------------
PARAMETER  DFITM
    do case
        case dfitm = 1
            stcl = 6
        case dfitm = 2
            stcl = 17
```

```
        case dfitm = 3
            stcl = 28
        case dfitm = 4
            stcl = 37
        case dfitm = 5
            stcl = 64
    endcase
RETURN  STCL
*-------------------------------------------------------
      FUNCTION  STRREP1
*-------------------------------------------------------
PARAMETER  DFITM
    do case
        case dfitm = 1
            stcl = 6
        case dfitm = 2
            stcl = 18
        case dfitm = 3
            stcl = 30
        case dfitm = 4
            stcl = 42
    endcase
RETURN  STCL
*-------------------------------------------------------
      FUNCTION  STCLRP21
*-------------------------------------------------------
PARAMETER  DFITM
    do case
        case dfitm = 1
            stclr21 = 30
```

```
        case dfitm = 2
            stclr21 = 59
    endcase
RETURN  STCLR21
*-----------------------------------------------------
    FUNCTION  STCLRP22
*-----------------------------------------------------
PARAMETER  DFITM
    do case
        case dfitm = 1
            stclr22 = 4
        case dfitm = 2
            stclr22 = 24
        case dfitm = 3
            stclr22 = 58
        case dfitm = 4
            stclr22 = 70
    endcase
RETURN  STCLR22
*-----------------------------------------------------
    FUNCTION  STCLRP23
*-----------------------------------------------------
PARAMETER  DFITM
    do case
        case dfitm = 1
            stclr23 =  4
        case dfitm = 2
            stclr23 = 24
        case dfitm = 3
            stclr23 = 41
```

```
            case dfitm = 4
                stclr23 = 58
            case dfitm = 5
                stclr23 = 70
        endcase
 RETURN  STCLR23
*-------------------------------------------------------
        FUNCTION  strtcl_box
*-------------------------------------------------------
parameter  item_main
    do case
        case item_main = 1
            strtcl_b = 1
        case item_main = 2
            strtcl_b  = 32
    case item_main = 3
            strtcl_b = 65
  endcase
return  strtcl_b
*-------------------------------------------------------
        FUNCTION  strtcl_bx2
*-------------------------------------------------------
parameter  item_main
  do case
      case item_main = 1
          strtcl_b = 1
      case item_main = 2
          strtcl_b = 15
      case item_main = 3
          strtcl_b = 23
```

```
        case item_main = 4
            strtcl_b = 60
  endcase
return  strtcl_b
*-------------------------------------------------------
          PROCEDURE   say_snam
*-------------------------------------------------------
set colo to i
MIDLINE = INT((80-LEN(SYSNAME))/2)
@ 24, 0   CLEAR  TO  24, 79
@ 24, MIDLINE  SAY  SYSNAME
set colo to w/n,n/w
RETURN
*-------------------------------------------------------
 Procedure  DSPMSG   &&  DISPLAY MASSAGE
*-------------------------------------------------------
PARAMETER  BGNROW, BGNCOL, ENDCOL, LENLIN
    MIDLIN = INT((LENLIN-LEN(MASSAGE))/2) + BGNCOL
    @ BGNROW, BGNCOL  CLEAR  TO  BGNROW, ENDCOL
    @ BGNROW, MIDLIN  SAY  MASSAGE
    @ 23, 0  SAY  "
    WAIT "
    @ BGNROW, BGNCOL  CLEAR  TO  BGNROW, ENDCOL
 RETURN
*-------------------------------------------------------
Procedure  DSPMSGM1
*-------------------------------------------------------
MIDLINE = INT((52-LEN(MASSAGE))/2) + 13
    @ 11, 13  clear  to  11, 65
    @ 11, MIDLINE  SAY  MASSAGE
    WAIT ' '
```

```
   @ 11, 13  clear  to  11, 65
 RETURN
*---------------------------------------
    Function  Getnamcus
*---------------------------------------
para  r1,c1,r2,c2,maxrec
    windows(r1-1,c1-1,r2+1,c2+1)
    set color to w/n,n/w
    do whil .t.
        a = achoice(r1,c1,r2,c2,custcode)
        if a > 0 .and. a < maxrec+1
            exit
        else
            if a = 0
                a = maxrec + 1
                stor  ' '  to  custcode[a]
                exit
            else
                ? chr(7)
            endif
        endif
    enddo
retu custcode[a]
*---------------------------------------
    Function  Getnampay
*---------------------------------------
para  r1,c1,r2,c2,maxrec
    windows(r1-1,c1-1,r2+1,c2+1)
    set color to w/n,n/w
    do whil .t.
       a = achoice(r1,c1,r2,c2,condcode)
```

```
        if a > 0 .and. a < maxrec+1
           exit
        else
           if a = 0
              a = maxrec + 1
              stor  ' '  to  condcode[a]
              exit
           else
              ? chr(7)
           endif
        endif
     enddo
retu condcode[a]
*-----------------------------------
```

*********  END  PROGRAM  ***************

```
*--------------------------------
 Procedure  MSTMAST1
*--------------------------------
set defa to dbfs
set cursor off
set delete on
m3bar1 = ' F2=Add '  && F2=-1
m3bar2 = ' F3=Edit '  && F3=-2
m3bar3 = ' F4=Del '     && F4=-3
m3bar4 = ' F5=Find '  && F5=-4
m3bar5 = ' ESC=Exit '
SELE 1
USE TYPEMER INDE TYPEME1
go bottom
botrec = recno()
GO TOP
stor   1   to  fcitm
stor   2   to  fcl
STOR  12  TO  STRTROW
STOR  13  TO  STRTCOL
STOR  66  TO  ENDCOLU
STOR  57  TO  LENGLINE
STOR  .T.  TO PROCMAS3
DO WHILE PROCMAS3
    set colo to r/gr+
    STOR TM_CODE  TO  ST_CODE
    STOR TM_NAME  TO  STNAME
    set colo to gr+/b,b/gr+
    @  7, 11 clear to 11, 65
    set colo to b/rb
    set colo to gr+/b,b/gr+
```

```
@  7, 11  SAY  '┌──────────────── [ 1. Type of Merchandise ]────────────────┐'
@  8, 11  SAY  '│                                                           '
@  9, 11  SAY  '│   Type Code   :                                           '
@ 10, 11  SAY  '│   Type Name   :                                           '
@ 11, 11  SAY  '├───────────────────────────────────────────────────────────'
@ 12, 11  SAY  '│                                                           '
@ 13, 11  SAY  '└───────────────────────────────────────────────────────────'
set colo to i
sysname = '<   M A S T E R   F I L E   S Y S T E M   >'
do say_snam
set colo to r/gr+
@  9, 31  say  ST_CODE    PICT  'X'
@ 10, 31  say  STNAME
fcl = strtclm1(fcitm)
set colo to w/n,n/w
@ 12, 13 CLEAR TO 12,65
@ 12, 13  SAY  ' F2=Add    F3=Edit    F4=Del    F5=Find '
@ 12, 57  SAY  ' ESC Exit '
itm = ltrim(str(fcitm))
set colo to n/w,w/n
@ 12,fcl say m3bar&itm
set colo to w/n,n/w
store  .t.  to  selact
DO WHILE selact
    set cursor off
    key = 0
    do whil key = 0
        key=INKEY()
    enddo
    if key=-1 .or. key=-2 .or. key=-3 .or. key=-4 .or. key=27 .or. key=13
        store .f.  to  selact
    endif
```

```
DO CASE
    CASE  key = 4        && right
        @ 12,fcl say m3bar&itm
        fcitm = fcitm + 1
        if fcitm = 6
            fcitm = 1
        endif
    CASE  key = 19       && left
        @ 12,fcl say m3bar&itm
        fcitm = fcitm - 1
        if fcitm < 1
            fcitm = 5
        endif
    CASE  key = 24       && down
        CURREC = RECNO()
        GO BOTTOM
        BOTREC = RECNO()
        GO CURREC
        IF botrec <> RECNO() .AND. .NOT. EOF()
            set colo to r/gr+
            SKIP
            @  9, 31  SAY  TM_CODE   PICT  'X'
            @ 10, 31  SAY  TM_NAME
        ENDIF
    CASE key = 5        && up
        IF .NOT. BOF()
            SKIP -1
        ENDIF
        set colo to r/gr+
        @  9, 31  SAY  TM_CODE   PICT  'X'
        @ 10, 31  SAY  TM_NAME
```

```
CASE key = 1       && HOME
    GO TOP
    set colo to r/gr+
    @  9, 31  SAY  TM_CODE   PICT  'X'
    @ 10, 31  SAY  TM_NAME
CASE key = 6       && END
    GO BOTTOM
    botrec = recno()
    set colo to r/gr+
    @  9, 31  SAY  TM_CODE   PICT  'X'
    @ 10, 31  SAY  TM_NAME
 CASE  KEY = 27  .OR.  fcitm = 5   &&  EXIT
    STOR  .F.  TO PROCMAS3
    EXIT
CASE  KEY = -1   .OR. (KEY = 13 .AND. fcitm = 1)   &&  F2=ADD DATA
    set cursor on
    set colo to r/gr+
    fcitm = 1
    STORE  SPACE(1)  TO  ST_CODE
    STORE  SPACE(25) TO  STNAME
    @  9, 31  GET  ST_CODE   PICT  'X'
    @ 10, 31  GET  STNAME
    READ
    IF LAST() = 27
        EXIT
    ENDIF
    IF ST_CODE = ' '
        MASSAGE = ' Please key Type Code '
        DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, LENGLINE
    ELSE
        SEEK ST_CODE
```

```
        IF  FOUND()
            MASSAGE = ' This Code Use Duplicate '
            DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, ;
             LENGLINE
        ELSE
             SELE 1
             APPEND  BLANK
             REPL  TM_CODE  WITH  ST_CODE,  TM_NAME  WITH    ;
                   STNAME
             ? CHR(7)
        ENDIF
    ENDIF
    currec = recno()
    go bottom
    botrec = recno()
    go currec
    fcl = strtclm1(fcitm)
CASE  KEY = -2   .OR.  (KEY = 13  .AND. fcitm = 2)   &&  F3=EDIT DATA
    set cursor on
    set colo to r/gr+
    fcitm = 2
    currec = recno()
    STORE  TM_CODE  TO  ST_CODE, SOT_CODE
    STNAME = TM_NAME
    @  9, 31  GET  ST_CODE  PICT  'X'
    @ 10, 31  GET  STNAME
    READ
    IF LAST() = 27
        EXIT
    ENDIF
```

```
IF ST_CODE = ' '
    MASSAGE = ' Enter Code Again Please '
    DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, LENGLINE
ELSE
     SEEK SOT_CODE
     IF .NOT. FOUND()
          MASSAGE = ' Information can not Update '
          DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, ;
          LENGLINE
     ELSE
          SELE 1
          REPL  TM_CODE  WITH  ST_CODE,  TM_NAME  WITH  ;
                    STNAME
          ? CHR(7)
     ENDIF
ENDIF
go bottom
botrec = recno()
go currec
fcl = strtclm1(fcitm)
CASE  KEY = -3   .OR.  (KEY = 13 .AND. fcitm = 3)   &&  F4=DELETE DATA
     set cursor on
     set colo to w/n,n/n
     fcitm = 3
     currec = recno()
     STOR  SPACE(1)  TO  CONFDEL
     @ 12,13 CLEAR TO 12,65
     @ 12,19  SAY ' Delete Reccord Confirm Please ? (Y/N) '
     SET COLO TO N/W,N/W
     @ 12,60  GET  CONFDEL PICT 'X'
     READ
```

```
set colo to
IF CONFDEL $ 'Yy'
    GO CURREC
    DELETE
    ? chr(7)
    IF .NOT. BOF()
        SKIP -1
    ENDIF
ENDIF
fcl = strtclm1(fcitm)
CASE  KEY = -4  .OR.  (KEY = 13 .AND. fcitm = 4)  &&  F5=FIND DATA
SAVE SCREEN TO MSTMAST1
fcitm = 4
currec = recno()
set colo to gr+/rb,rb/gr+
@ 8 ,20 SAY '  ------- Select Topic...  ----------------'
@ 9 ,20 SAY '                                           '
@ 10,20 SAY '                                           '
set colo to N/GR,GR/N
@ 10,21 SAY '  (*) Type Code                            '
set colo to gr+/rb,rb/gr+
@ 10,58 SAY '                                           '
@ 11,20 SAY '                                           '
@ 12,20 SAY '  ---------------------------------------  '
@ 13,20 SAY '   Data :                                  '
@ 14,20 SAY '  ---------------------------------------  '
@ 15,20 SAY '                                           '
@ 16,20 SAY '  ---------------------------------------  '
@ 17,20 SAY '       < O.K. >          < Cancle >        '
@ 18,20 SAY '                                           '
set colo to r+/rb,rg/r+
```

```
@ 17,26 SAY '< O.K. >'
@ 17,45 SAY '< Cancle >'
A = 0
DO WHILE A = 0
    A = INKEY()
ENDDO
IF LAST() = 27
    RESTORE SCREEN FROM MSTMAST1
    EXIT
ENDIF
set colo to gr+/rb,rb/gr+
@ 10,21 SAY  '(*) Type Code                    '
STORE  SPACE(1)  TO  ST_CODE
STORE .T. TO FLAG
STORE 1 TO OK
SET COLO TO W/N,N/W
@ 17,26 SAY '< O.K. >'
@ 17,45 SAY '< Cancle >'
set cursor on
set colo to N/W+,W+/N
@  13, 32  GET  ST_CODE  PICT  'X'
read
IF LAST() = 27
    RESTORE SCREEN FROM MSTMAST1
    EXIT
ENDIF
set cursor off
DO WHILE FLAG
    STORE 0 TO KEY
    DO WHILE KEY = 0
        KEY = INKEY()
    ENDDO
```

```
DO CASE
    CASE  key = 4       && right
        OK = OK + 1
        if OK = 3
            OK = 1
        endif
    CASE  key = 19      && left
        OK = OK - 1
        if OK < 1
            OK = 2
        endif
    CASE  KEY = 13
        IF OK = 1
            STORE .F. TO FLAG
        ELSE
            STORE .T. TO FLAG
            STORE 1 TO OK
            STORE  SPACE(1)  TO  ST_CODE
            set colo to r+/rb,rb/r+
            @ 17,26 SAY '< O.K. >'
            @ 17,45 SAY '< Cancle >'
            IF OK = 1
                SET COLO TO W/N,N/W
                @ 17,26 SAY '< O.K. >'
            ELSE
                SET COLO TO W/N,N/W
                @ 17,45 SAY '< Cancle >'
            ENDIF
            set cursor on
            set colo to N/W+,W+/N
            @  13, 32  GET  ST_CODE  PICT  'X'
```

```
                read
                IF LAST() = 27
                    RESTORE SCREEN FROM MSTMAST1
                    EXIT
                ENDIF
                set cursor off
            ENDIF
    ENDCASE
    set colo to r+/rb,rb/r+
    @ 17,26 SAY '< O.K. >'
    @ 17,45 SAY '< Cancle >'
    IF OK = 1
        SET COLO TO W/N,N/W
        @ 17,26 SAY '< O.K. >'
    ELSE
        SET COLO TO W/N,N/W
        @ 17,45 SAY '< Cancle >'
    ENDIF
ENDDO
RESTORE SCREEN FROM MSTMAST1
set colo to N/W
IF LAST() = 27
    EXIT
ENDIF
SEEK ST_CODE
IF .NOT. FOUND()
    MASSAGE = ' Record Not Found '
    DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, LENGLINE
    go currec
ENDIF
fcl = strtclm1(fcitm)
```

```
            OTHER
                LOOP
        ENDCASE
        fcl = strtclm1(fcitm)
        itm = ltrim(str(fcitm))
        set colo to n/w,w/n
        @ 12,fcl say m3bar&itm
        set colo to w/n,n/w
    ENDDO
ENDDO
RETURN
*---------------------------------
```

```
*----------------------------------
 Procedure  MSTMAST2
*----------------------------------
set defa to dbfs
set cursor off
set delete on
m3bar1 = ' F2=Add '  && F2=-1
m3bar2 = ' F3=Edit '  && F3=-2
m3bar3 = ' F4=Del '     && F4=-3
m3bar4 = ' F5=Find '  && F5=-4
m3bar5 = ' ESC=Exit '
SELE 2
USE GRPMER  INDE  GRPMER1
go bottom
botrec = recno()
go top
stor   1  to  fcitm
stor   2  to  fcl
STOR  14  TO  STRTROW
STOR  13  TO  STRTCOL
STOR  66  TO  ENDCOLU
STOR  60  TO  LENGLINE
STOR  .T.  TO PROCMAS1
DO WHILE PROCMAS1
    STOR  GP_CODE  TO  SG_CODE
    STOR  GP_NAME  TO  SG_NAME
    set colo to gr+/b,b/gr+
    @ 8, 11 clear to 15, 65
    set colo to gr+/b,b/gr+
```

```
@  8, 11  SAY  ' ------------ [ 2. Group of Merchandise ] ----------- '
@  9, 11  SAY  '                                                       '
@ 10, 11  SAY  '   Group Code  :                                       '
@ 11, 11  SAY  '   Group Name  :                                       '
@ 12, 11  SAY  '                                                       '
@ 13, 11  SAY  '                                                       '
@ 14, 11  SAY  '                                                       '
@ 15, 11  SAY  '                                                       '
set colo to i
sysname = '<   M A S T E R   F I L E   S Y S T E M   >'
do say_snam
do sayfmas2
fcl = strtclm1(fcitm)
set colo to w/n,n/w
@ 14, 13 CLEAR TO 14,65
@ 14, 13  SAY  ' F2=Add    F3=Edit    F4=Del   F5=Find '
@ 14, 57  SAY  ' ESC=Exit '
itm = ltrim(str(fcitm))
set colo to n/w,w/n
@ 14,fcl say m3bar&itm
set colo to w/n,n/w
store  .t.  to  selact
DO WHILE selact
    set cursor off
    key = 0
    do whil key = 0
        key=INKEY()
    enddo
    CURREC = RECNO()
    GO BOTTOM
    BOTREC = RECNO()
```

```
GO CURREC
if key=-1 .or. key=-2 .or. key=-3 .or. key=-4 .or. key=27 .or. key=13
   store .f.  to  selact
endif
DO CASE
     CASE  key = 4        && right
          @ 14,fcl say m3bar&itm
          fcitm = fcitm + 1
          if fcitm = 6
               fcitm = 1
          endif
     CASE  key = 19       && left
          @ 14,fcl say m3bar&itm
          fcitm = fcitm - 1
          if fcitm < 1
               fcitm = 5
          endif
     CASE  key = 24       && down
          IF botrec <> RECNO() .AND. .NOT. EOF()
               SKIP
               STORE GP_CODE TO SG_CODE
               STORE GP_NAME TO SG_NAME
               do sayfmas2
          ENDIF
     CASE  key = 5        && up
          IF .NOT. BOF()
               SKIP -1
          ENDIF
          STORE GP_CODE TO SG_CODE
          STORE GP_NAME TO SG_NAME
          do sayfmas2
```

```
CASE  key = 1       && HOME
      GO TOP
      STORE GP_CODE TO SG_CODE
      STORE GP_NAME TO SG_NAME
      do sayfmas2
CASE  key = 6       && END
      GO BOTTOM
      botrec = recno()
      STORE GP_CODE TO SG_CODE
      STORE GP_NAME TO SG_NAME
      do sayfmas2
CASE  KEY = 27  .OR.  fcitm = 5   &&  EXIT
      STOR  .F.  TO PROCMAS1
      EXIT
CASE  KEY = -1   .OR. (KEY = 13 .AND. fcitm = 1)   &&  F2=ADD DATA
      fcitm = 1
      currec = recno()
      set colo to gr+/b,b/gr+
      @ 10, 29 CLEAR TO 10,65
      @ 11, 29 CLEAR TO 11,65
      STORE  SPACE(1)  TO  ST_CODE
      STORE  SPACE(2)  TO  SG_CODE
      STORE  SPACE(35) TO  SG_NAME
      SET COLO TO N/W,N/W
      @ 10, 29 GET SG_CODE
      @ 11, 29 GET SG_NAME
      CLEAR GETS
      set cursor on
      set colo to r/gr+
      DO WHILE .T.
          @ 10, 29  get  SG_CODE   PICT  'XX'
```

```
@ 11, 29  get  SG_NAME
READ
IF LAST() = 27
     EXIT
ENDIF
IF  LEN(LTRIM(SG_CODE)) < 2
    MASSAGE = ' Enter Code Again Please '
    DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, ;
    LENGLINE
ELSE
    SEEK SG_CODE
    IF  FOUND()
        MASSAGE = ' This Code Use Dupplicate '
        DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, ;
        LENGLINE
        go currec
        do sayfmas2
        exit
    ELSE
        SELE 2
        APPEND  BLANK
        REPLACE  GP_CODE  WITH  SG_CODE, ;
                 GP_NAME  WITH  SG_NAME
        currec = recno()
        go bottom
        botrec = recno()
        ? CHR(7)
        EXIT
    ENDIF
ENDIF
set colo to w/n,n/w
```

```
        @ 14, 13 CLEAR TO 14,65
        @ 14, 13  SAY  ' F2=Add    F3=Edit    F4=Del   F5=Find '
        @ 14, 57  SAY  ' ESC=Exit '
        itm = ltrim(str(fcitm))
        set colo to n/w,w/n
        @ 14,fcl say m1bar&itm
        set colo to w/n,n/w
    ENDDO
    currec = recno()
    go bottom
    botrec = recno()
    go currec
CASE  KEY = -2   .OR.  (KEY = 13 .AND. fcitm = 2)   &&  F3=EDIT DATA
    fcitm = 2
    currec = recno()
    set colo to r/gr+
    SELE 2
    STORE  RIGHT(GP_CODE,2)  TO  SG_CODE, SOG_CODE
    store  GP_name  to  SG_NAME
    set cursor on
    set colo to r/gr+
    @ 10, 29  get  SG_CODE  PICT  'XX'
    @ 11, 29  get  SG_NAME
    READ
    IF LAST() = 27
        EXIT
    ENDIF
    SELE 2
    IF SG_CODE = SPACE(2) .OR. LEN(LTRIM(SG_CODE)) < 2
        MASSAGE = ' Enter Code Again Please '
        DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, ;
```

```
    LENGLINE
ELSE
    IF SG_CODE = SOG_CODE
        SEEK SOG_CODE
        IF FOUND()
            REPLACE  GP_NAME  WITH  SG_NAME
            ? CHR(7)
        ELSE
            MASSAGE = ' Information can not Update  '
            DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, ;
            LENGLINE
        ENDIF
    ELSE
        SEEK SG_CODE
        IF .NOT. FOUND()
            GO CURREC
            REPLACE  GP_CODE  WITH  SG_CODE, ;
                     GP_NAME  WITH  SG_NAME
            ? CHR(7)
        ELSE
            MASSAGE = ' Duplicate Record, record can not add '
            DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, ;
            LENGLINE
        ENDIF
    ENDIF
ENDIF
go bottom
botrec = recno()
go currec
fcl = strtclm1(fcitm)
```

```
CASE  KEY = -3  .OR. (KEY = 13 .AND. fcitm = 3)   &&  F4=DELETE DATA
     fcitm = 3
     currec = recno()
     STOR  SPACE(1)  TO  CONFDEL
    @ 14,13  clear  to  14, 65
    @ 14,19  SAY  ' Delete Record Confirm Please ? (Y/N) '
    SET COLO TO N/W,N/W
    @ 14,60  GET  CONFDEL PICT 'X'
    READ
    IF CONFDEL $ 'Yy'
        GO CURREC
        DELETE
        ? chr(7)
        IF .NOT. BOF()
            SKIP -1
        ENDIF
    ENDIF
    fcl = strtclm1(fcitm)
CASE  KEY = -4  .OR.  (KEY = 13 .AND. fcitm = 4)  &&  F5=FIND DATA
     SAVE SCREEN TO MSTMAST2
     fcitm = 4
     currec = recno()
     set colo to gr+/rb,rb/gr+
```

```
@ 8 ,20 SAY '  ------- Select Topic...  ---------------'
@ 9 ,20 SAY '                                          '
@ 10,20 SAY '                                          '
set colo to N/GR,GR/N
@ 10,21 SAY '(*) Group Code                            '
set colo to gr+/rb,rb/gr+
@ 10,58 SAY '                                          '
@ 11,20 SAY '                                          '
@ 12,20 SAY '  ------------------------------------    '
@ 13,20 SAY '  Data :                                 '
@ 14,20 SAY '  ------------------------------------    '
@ 15,20 SAY '                                          '
@ 16,20 SAY '  ------------------------------------    '
@ 17,20 SAY '     < O.K. >          < Cancle >         '
@ 18,20 SAY '                                          '
set colo to r+/rb,rg/r+
@ 17,26 SAY '< O.K. >'
@ 17,45 SAY '< Cancle >'
A = 0
DO WHILE A = 0
    A = INKEY()
ENDDO
IF LAST() = 27
    RESTORE SCREEN FROM MSTMAST2
    EXIT
ENDIF
set colo to gr+/rb,rb/gr+
@ 10,21 SAY '(*) Group Code                            '
STORE SPACE(2) TO SG_CODE
STORE SPACE(1) TO ST_CODE
STORE .T. TO FLAG
```

```
STORE 1 TO OK
SET COLO TO W/N,N/W
@ 17,26 SAY '< O.K. >'
set cursor on
set colo to N/W+,W+/N
@ 13, 31  get  SG_CODE   PICT  'XXX'
read
IF LAST() = 27
    RESTORE SCREEN FROM MSTMAST2
    EXIT
ENDIF
set cursor off
DO WHILE FLAG
    STORE 0 TO KEY
    DO WHILE KEY = 0
        KEY = INKEY()
    ENDDO
    DO CASE
        CASE  key = 4       && right
              OK = OK + 1
              if OK = 3
                  OK = 1
              endif
        CASE  key = 19      && left
              OK = OK - 1
              if OK < 1
                  OK = 2
              endif
        CASE  KEY = 13
              IF OK = 1
                  STORE .F. TO FLAG
```

```
        ELSE
            STORE  1 TO OK
            STORE .T. TO FLAG
            STORE  SPACE(2)  TO  SG_CODE
            set colo to r+/rb,rb/r+
            @ 17,26 SAY '< O.K. >'
            @ 17,45 SAY '< Cancle >'
            IF OK = 1
                SET COLO TO W/N,N/W
                @ 17,26 SAY '< O.K. >'
            ELSE
                SET COLO TO W/N,N/W
                @ 17,45 SAY '< Cancle >'
            ENDIF
            set cursor on
            set colo to N/W+,W+/N
            @ 13, 31  get  SG_CODE   PICT  'XXX'
            read
            IF LAST() = 27
                RESTORE SCREEN FROM MSTMAST2
                EXIT
            ENDIF
            set cursor off
        ENDIF
ENDCASE
set colo to r+/rb,rb/r+
@ 17,26 SAY '< O.K. >'
@ 17,45 SAY '< Cancle >'
IF OK = 1
    SET COLO TO W/N,N/W
    @ 17,26 SAY '< O.K. >'
```

```
                ELSE
                    SET COLO TO W/N,N/W
                    @ 17,45 SAY '< Cancle >'
                ENDIF
            ENDDO
            RESTORE SCREEN FROM MSTMAST2
            set colo to N/W+
            IF LAST() = 27
                EXIT
            ENDIF
            SEEK SG_CODE
            IF .NOT. FOUND()
                MASSAGE = ' Group of Merchase not found '
                DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, LENGLINE
                go currec
            ELSE
                @ 11, 29  say  ALLTRIM(GP_NAME)
            ENDIF
            fcl = strtclm1(fcitm)
        OTHER
            LOOP
    ENDCASE
    fcl = strtclm1(fcitm)
    itm = ltrim(str(fcitm))
    set colo to n/w,w/n
    @ 14,fcl say m3bar&itm
    set colo to w/n,n/w
  ENDDO
ENDDO
RETURN
*-------------------------------
```

```
*-------------------------------
 Procedure  SAYFMAS2
*-------------------------------
   DO CLRSCREEN
   set colo to r/gr+
   @ 10, 29  say  SG_CODE   PICT  'XX'
   @ 11, 29  say  ALLTRIM(SKNAMGRP(SG_CODE))
RETURN
*--------------------------------
 Procedure  CLRSCREEN
*--------------------------------
    set colo to W/B
    @ 10, 29  say  '  '
    @ 11, 29  say  '                    '
RETURN
*--------------------------------
```

```
*-------------------------------
 Procedure  MSTMAST3
*-------------------------------
set defa to dbfs
set cursor off
set dele on
m3bar1 = ' F2=Add '  && F2=-1
m3bar2 = ' F3=Edit '  && F3=-2
m3bar3 = ' F4=Del '     && F4=-3
m3bar4 = ' F5=Find '  && F5=-4
m3bar5 = ' ESC=Exit '
SELE 3
USE STMER INDEX STMER1, STMER2
SET ORDER TO 1
go bottom
botrec = recno()
to top
stor  1  to  fcitm
stor  2  to  fcl
STOR  12  TO  STRTROW
STOR  13  TO  STRTCOL
STOR  66  TO  ENDCOLU
STOR  52  TO  LENGLINE
STOR  .T.  TO PROCMAS3
DO WHILE PROCMAS3
    set colo to r/gr+
    STOR ST_CODE  TO  SFT_CODE
    STOR ST_NAME  TO  SSTNAME
    set colo to gr+/b,b/gr+
    @  7, 11 clear to 11, 65
    set colo to b/rb
```

```
set colo to gr+/b,b/gr+
@  7, 11  SAY  '  --------- [ 3. Structure of Merchandise ] ----------  '
@  8, 11  SAY  '  |                                                    |  '
@  9, 11  SAY  '  |  Structure Code  :                                 |  '
@ 10, 11  SAY  '  |  Structure Name  :                                 |  '
@ 11, 11  SAY  '  |----------------------------------------------------|  '
@ 12, 11  SAY  '  |                                                    |  '
@ 13, 11  SAY  '  |____________________________________________________|  '
set colo to i
sysname = '<   M A S T E R   F I L E   S Y S T E M   >'
do say_snam
set colo to r/gr+
DO WHILE .T.
   IF FT_CODE = '3'
      @  9, 32  say  ST_CODE    PICT 'X'
      @ 10, 32  say  ST_NAME
      EXIT
   ENDIF
   SKIP
ENDDO
fcl = strtclm1(fcitm)
set colo to w/n,n/w
@ 12, 13 CLEAR TO 12,65
@ 12, 13  SAY  ' F2=Add    F3=Edit    F4=Del   F5=Find '
@ 12, 57  SAY  ' ESC=Exit '
itm = ltrim(str(fcitm))
set colo to n/w,w/n
@ 12,fcl say m3bar&itm
set colo to w/n,n/w
store  .t.  to  selact
```

```
DO WHILE selact
    set cursor off
    key = 0
    do whil key = 0
        key=INKEY()
    enddo
    CURREC = RECNO()
    GO BOTTOM
    BOTREC = RECNO()
    GO CURREC
    if key=-1 .or. key=-2 .or. key=-3 .or. key=-4 .or. key=27 .or. key=13
        store .f. to selact
    endif
    DO CASE
        CASE key = 4        && right
            @ 12,fcl say m3bar&itm
            fcitm = fcitm + 1
            if fcitm = 6
                fcitm = 1
            endif
        CASE key = 19       && left
            @ 12,fcl say m3bar&itm
            fcitm = fcitm - 1
            if fcitm < 1
                fcitm = 5
            endif
        CASE key = 24       && down
            IF .NOT. EOF()
                DO WHILE .T. .AND. .NOT. EOF()
                    SKIP
```

```
            IF FT_CODE = '3'
                EXIT
            ENDIF
        ENDDO
    ENDIF
    IF EOF()
        GO CURREC
        ? CHR(7)
    ENDIF
    set colo to r/gr+
    @  9, 32  SAY  ST_CODE   PICT  'X'
    @ 10, 32  SAY  ST_NAME
CASE  key = 5       && up
    IF .NOT. EOF()
        DO WHILE .T. .AND. .NOT. EOF()
            SKIP -1
            IF FT_CODE = '3'
                EXIT
            ENDIF
        ENDDO
    ENDIF
    IF BOTREC = RECNO() .OR. EOF()
        DO WHILE .T.
            SKIP
            IF FT_CODE = '3'
                EXIT
            ENDIF
        ENDDO
    ENDIF
    set colo to r/gr+
    @  9, 32  SAY  ST_CODE   PICT  'X'
    @ 10, 32  SAY  ST_NAME
```

```
CASE key = 1        && HOME
    GO TOP
    DO WHILE .T.
        IF FT_CODE = '3'
            EXIT
        ENDIF
        SKIP
    ENDDO
    set colo to r/gr+
    @  9, 32  SAY  ST_CODE  PICT  'X'
    @ 10, 32  SAY  ST_NAME
CASE  key = 6       && END
    GO BOTTOM
    botrec = recno()
    DO WHILE .T.
        SKIP -1
        IF FT_CODE = '3'
            EXIT
        ENDIF
    ENDDO
    set colo to r/gr+
    @  9, 32  SAY  ST_CODE  PICT  'X'
    @ 10, 32  SAY  ST_NAME
CASE  KEY = 27  .OR.  fcitm = 5   &&  EXIT
    STOR  .F.  TO PROCMAS3
    EXIT
CASE  KEY = -1   .OR. (KEY = 13 .AND. fcitm = 1)   &&  F2=ADD DATA
    set cursor on
    set colo to r/gr+
    fcitm = 1
    STORE  SPACE(1)  TO  SFT_CODE
```

```
STORE  SPACE(35) TO  SSTNAME
@  9, 32  GET  SFT_CODE   PICT  '!'
@ 10, 32  GET  SSTNAME
READ
IF LAST() = 27
   EXIT
ENDIF
IF SFT_CODE = ' '
    MASSAGE = ' Enter Code Again Please '
    DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, LENGLINE
ELSE
    SET ORDER TO 2
    SEEK SFT_CODE
    IF  FOUND()
        MASSAGE = ' This Code Use Dupplicate '
        DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, ;
         LENGLINE
    ELSE
        SELE 3
        APPEND  BLANK
        REPLACE   FT_CODE  WITH  '3'
        REPLACE   ST_CODE  WITH  SFT_CODE,  ST_NAME  WITH  ;
                  SSTNAME
        ? CHR(7)
    ENDIF
ENDIF
currec = recno()
go bottom
botrec = recno()
go currec
fcl = strtclm1(fcitm)
```

```
CASE  KEY = -2   .OR.  (KEY = 13 .AND. fcitm = 2)   &&  F3=EDIT DATA
    set cursor on
    set colo to r/gr+
    fcitm = 2
    currec = recno()
    STORE ST_CODE  TO  SFT_CODE, SOT_CODE
    STORE ST_NAME   TO  SSTNAME
    @  9, 32  GET  SFT_CODE  PICT  '!'
    @ 10, 32  GET  SSTNAME
    READ
    IF LAST() = 27
        EXIT
    ENDIF
    IF SFT_CODE = ' '
        MASSAGE = ' Enter Code Again Please '
        DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, LENGLINE
    ELSE
        SET ORDER TO 2
        SEEK SOT_CODE
        IF .NOT. FOUND()
            MASSAGE = ' Information can not Update '
            DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, ;
             LENGLINE
        ELSE
            SELE 3
            SET ORDER TO 2
            REPLACE  FT_CODE  WITH '3'
            REPLACE  ST_CODE  WITH  SFT_CODE,  ST_NAME  WITH  ;
                     SSTNAME
            ? CHR(7)
        ENDIF
    ENDIF
```

```
    go bottom
    botrec = recno()
    go currec
    fcl = strtclm1(fcitm)
CASE  key = -3 .or. (key = 13 .and. fcitm = 3) &&  F4=DELETE    DATA
    set cursor on
    set colo to w/n,n/n
    fcitm = 3
    currec = recno()
    STOR  SPACE(1)  TO  CONFDEL
    @ 12,13 CLEAR TO 12,65
    @ 12,19  SAY ' Delete Recore Confirm Please ? (Y/N) '
    set colo to n/W,N/W
    @ 12,60  GET  CONFDEL PICT 'X'
    READ
    set colo to
    IF  CONFDEL $ 'Yy'
        GO CURREC
        DELETE
       ? chr(7)
        IF .NOT. BOF()
            SKIP -1
        ENDIF
    ENDIF
    fcl = strtclm1(fcitm)
CASE  KEY = -4   .OR.  (KEY = 13 .AND. fcitm = 4)  &&  F5=FIND DATA
    SAVE SCREEN TO MSTMAST3
    fcitm = 4
    currec = recno()
    set colo to gr+/rb,rb/gr+
```

```
@ 8 ,20 SAY '  ------ Select Topic...  ---------------'
@ 9 ,20 SAY '  |                                      |'
@ 10,20 SAY '  |                                      |'
set colo to N/GR,GR/N
@ 10,21 SAY '(*) Group Code                           '
set colo to gr+/rb,rb/gr+
@ 10,58 SAY '  |                                      |'
@ 11,20 SAY '  |                                      |'
@ 12,20 SAY '  |--------------------------------------|'
@ 13,20 SAY '  | Data :                               |'
@ 14,20 SAY '  |--------------------------------------|'
@ 15,20 SAY '  |                                      |'
@ 16,20 SAY '  |--------------------------------------|'
@ 17,20 SAY '  |     < O.K. >        < Cancle >       |'
@ 18,20 SAY '  |__________________________________________|'
set colo to r+/rb,rg/r+
@ 17,26 SAY '< O.K. >'
@ 17,45 SAY '< Cancle >'
A = 0
DO WHILE A = 0
   A = INKEY()
ENDDO
IF LAST() = 27
    RESTORE SCREEN FROM MSTMAST3
    EXIT
ENDIF
set colo to gr+/rb,rb/gr+
@ 10,21 SAY '(*) Structure Code                       '
STORE SPACE(1) TO SFT_CODE
STORE .T. TO FLAG
STORE 1 TO OK
```

```
ELSE
    STORE .T. TO FLAG
    STORE  1  TO OK
    STORE  SPACE(1)  TO  SFT_CODE
    set colo to r+/rb,rb/r+
    @ 17,26 SAY '< O.K. >'
    @ 17,45 SAY '< Cancle >'
    IF OK = 1
        SET COLO TO W/N,N/W
        @ 17,26 SAY '< O.K. >'
    ELSE
        SET COLO TO W/N,N/W
        @ 17,45 SAY '< Cancle >'
    ENDIF
    set cursor on
    set colo to N/W+,W+/N
    @  13, 32  GET  SFT_CODE  PICT  '!'
    read
    IF LAST() = 27
        RESTORE SCREEN FROM MSTMAST3
        EXIT
    ENDIF
    set cursor off
ENDIF
ENDCASE
set colo to r+/rb,rb/r+
@ 17,26 SAY '< O.K. >'
@ 17,45 SAY '< Cancle >'
IF OK = 1
    SET COLO TO W/N,N/W
    @ 17,26 SAY '< O.K. >'
```

```
                ELSE
                    SET COLO TO W/N,N/W
                    @ 17,45 SAY '< Cancle >'
                ENDIF
            ENDDO
            RESTORE SCREEN FROM MSTMAST3
            set colo to N/W+
            IF LAST() = 27
               EXIT
            ENDIF
            SET ORDER TO 1
            SEEK '3'+SFT_CODE
            IF .NOT. FOUND()
               MASSAGE = ' Record NOt Found'
               DO DSPMSG WITH STRTROW, STRTCOL, ENDCOLU, LENGLINE
               go currec
            ENDIF
            SET ORDER TO 1
            fcl = strtclm1(fcitm)
        OTHER
            LOOP
    ENDCASE
    SET ORDER TO 1
    fcl = strtclm1(fcitm)
    itm = ltrim(str(fcitm))
    set colo to n/w,w/n
    @ 12,fcl say m3bar&itm
    set colo to w/n,n/w
  ENDDO
ENDDO
RETURN
*----------------------------------
```

```
*-----------------------------------------
    Procedure  MSTREPO1
*-----------------------------------------
set talk off
set cursor off
set defa to dbfs
SELE 1
USE TYPEMER INDE TYPEME1
set orde to 1
go top
repfunc1 = ' F10  Print '  && F10=-9
repfunc2 = ' ESC  Exit '
store   1  to  fcitm
store   0  to  I      && i = row of record on display screen
store   0  TO  J      && j = number of record on display screen
store   1  to  k      && k = page of display screen
set colo to W/B,W/N
@ 0, 0  clear  to  21, 79
@ 1, 1  SAY  'Grand Total Co., Ltd.  '
@ 2, 1  SAY  '1. Type of Merchandise Report '
@ 3, 0  SAY  '-----------------------------------------------------------------------------'
@ 5, 10  SAY  '       Type Code                    Type Name '
do dpftos1
set colo to w/n,n/w
@ 23, 0  clear to  23, 79
@ 23, 30  SAY  'F10  Print                    ESC  Exit '
fcl = stclrp21(fcitm)
itm = ltrim(str(fcitm))
set colo to n/w,w/n
@ 22, 0 clear to 22, 79
@ 23,fcl say repfunc&itm
```

```
set colo to w/n,n/w
store .t. to selact
do while selact
  A = 0
  DO WHIL A = 0
    A = INKEY()
  ENDDO
  if a=-9 .or. a=27 .or. a=4 .or. a=19 .or. a=18
    set colo to w/n,n/w
    @ 23,fcl say repfunc&itm
  endif
  DO CASE
    CASE A = -9  .OR.  (A = 13 .AND. fcitm = 1)    && PRINT
      stor 1 to fcitm
      fcl = stclrp21(fcitm)
      itm = ltrim(str(fcitm))
      set colo to n/w,w/n
      @ 23,fcl say repfunc&itm
      set colo to w/n,n/w
      DO MSTPRNR1
    CASE A = 27  .OR.  (A = 13 .AND. fcitm = 2)   && EXIT
      store .f. to selact
      EXIT
    CASE A = 4        && right
      set colo to w/n,n/w
      @ 23,fcl say repfunc&itm
      fcitm = fcitm + 1
      if fcitm = 3
        fcitm = 1
      endif
    CASE A = 19       && left
      set colo to w/n,n/w
```

```
      @ 23,fcl say repfunc&itm
      fcitm = fcitm - 1
      if fcitm < 1
        fcitm = 2
        @ 23,fcl say repfunc&itm
      endif
    OTHE
      A = 0
      * LOOP
  ENDCASE
  fcl = stclrp21(fcitm)
  itm = ltrim(str(fcitm))
  set colo to n/w,w/n
  @ 23,fcl say repfunc&itm
  set colo to w/n,n/w
enddo
RETURN
*-----------------------------
   Procedure  DPFTOS1
*-----------------------------
  sele 1
  set colo to W/B,W/N
  store  7   to  row
  do while .not. eof()
    @ row, 21  say  TM_CODE
    @ row, 35  say  TM_NAME
    row = row + 1
    skip
  enddo
RETURN
*-----------------------------------
```

```
*-----------------------------------
 PROCEDURE MSTPRNR1
*-----------------------------------
 PRTOUT = CHKPRT()
 IF PRTOUT
   SET DEVI TO PRINT
   SELE 1
   go top
   stor 30  to  l_page
   stor  0  to  page_Num
   do while .not. eof()
     @ 0,  0  SAY  CHR(27)+CHR(80)          &&  Letter 10 cpi
     @ 0,  0  say  CHR(27)+CHR(120)+CHR(1)     && LQ
     @ 0,  0  SAY  CHR(27)+CHR(51)+CHR(45)     && line feed spacing 1/4"
     @ 1,  1  SAY  ' Grand Total Co., Ltd.  '
     @ 2,  1  SAY  ' Type of Merchandise Report  '
     PAGE_NUM = PAGE_NUM + 1
     @ 2, 65  SAY  'Page  : '
     @ 2, 76  SAY  PAGE_NUM      PICT  '99'
     @ 3,  1  SAY  REPL('-',79)
     @ 5, 10  SAY  '      Type Code          Type Name '
     stor  7  to  row
     do whil row-7 < l_page .and. .not. eof()
       @ row, 21  say  TM_CODE  PICT  'X'
       @ row, 35  say  TM_NAME  PICT  '@X25'
       row = row + 1
       skip
     enddo
     @ row,     1  SAY  REPL('-',79)
     @ row+1,  2  SAY  'Print Date : '
     @ row+1, 17  SAY  SysDTOTL(DATE())
```

```
    @ row+1,  2  SAY  'Print Date : '
    @ row+1, 65  SAY  'Time : '
    @ row+1, 65  SAY  'Time : '
    @ row+1, 72  SAY  Time()
    EJECT
  enddo &&
  stor  2  to  fcitm
 ENDIF
 SET DEVI TO SCREEN
 set colo to w/n,n/w
 @ 23, 0  clear  to  23, 79
 @ 23, 30  SAY  'F10  Print                    ESC  Exit '
RETURN
*----------------------------------------
```

```
*----------------------------------------
    Procedure  MSTREPO2
*----------------------------------------
set talk off
set cursor off
set defa to dbfs
SELE 2
USE GRPMER  INDE  GRPMER1
set orde to 1
go top
repfunc1 = ' F10  Print '  && F10=-9
repfunc2 = ' ESC  Exit'
store    1  to  fcitm
store    0  to  I      && i = row of record on display screen
store    0  TO  J      && j = number of record on display screen
store    1  to  k      && k = page of display screen
set colo to W/B,W/N
@ 0, 0  clear  to  21, 79
@ 1, 1  SAY  'Grand Total Co., Ltd.  '
@ 2, 1  SAY  '2. Group of Merchandise Report  '
@ 3, 0  SAY  '-------------------------------------------------------------------------------'
@ 5, 10  SAY  '        Group Code      Group Name  '
do dpftos2
set colo to w/n,n/w
@ 23, 0  clear to  23, 79
@ 23, 30  SAY  'F10  Print                    ESC  Exit '
fcl = stclrp21(fcitm)
itm = ltrim(str(fcitm))
set colo to n/w,w/n
@ 22, 0 clear to 22, 79
@ 23,fcl say repfunc&itm
```

```
set colo to w/n,n/w
store .t. to selact
do while selact
  A = 0
  DO WHIL A = 0
    A = INKEY()
  ENDDO
  if a=-9 .or. a=27 .or. a=4 .or. a=19 .or. a=18
    set colo to w/n,n/w
    @ 23,fcl say repfunc&itm
  endif
  DO CASE
    CASE A = -9  .OR.  (A = 13 .AND. fcitm = 1)    && PRINT
      stor 1 to fcitm
      fcl = stclrp21(fcitm)
      itm = ltrim(str(fcitm))
      set colo to n/w,w/n
      @ 23,fcl say repfunc&itm
      set colo to w/n,n/w
      DO MSTPRNR2
    CASE A = 27  .OR.  (A = 13 .AND. fcitm = 2)  && EXIT
      store .f. to selact
      EXIT
    CASE A = 4       && right
      set colo to w/n,n/w
      @ 23,fcl say repfunc&itm
      fcitm = fcitm + 1
      if fcitm = 3
        fcitm = 1
      endif
    CASE A = 19      && left
      set colo to w/n,n/w
```

```
      @ 23,fcl say repfunc&itm
      fcitm = fcitm - 1
      if fcitm < 1
        fcitm = 2
        @ 23,fcl say repfunc&itm
      endif
    OTHE
      A = 0
      * LOOP
  ENDCASE
  fcl = stclrp21(fcitm)
  itm = ltrim(str(fcitm))
  set colo to n/w,w/n
  @ 23,fcl say repfunc&itm
  set colo to w/n,n/w
enddo
RETURN
*-------------------------------
    Procedure  DPFTOS2
*-----------------------------------
  sele 2
  set colo to W/B,W/N
  store  7   to  row
  do while .not. eof()
    @ row, 21  say  GP_CODE  pict  'XX'
    @ row, 32  say  GP_NAME  pict  '@X25'
    row = row + 1
    skip
  enddo
RETURN
*----------------------------------
```

```
*---------------------------------
 PROCEDURE MSTPRNR2
*---------------------------------
 PRTOUT = CHKPRT()
 IF PRTOUT
   SET DEVI TO PRINT
   SELE 2
   go top
   stor 30  to  l_page
   stor  0  to  page_Num
   do while .not. eof()
     @ 0, 0  SAY  CHR(27)+CHR(79)          && Letter 10 cpi
     @ 0, 0  say  CHR(27)+CHR(120)+CHR(1)    && LQ
     @ 0, 0  SAY  CHR(27)+CHR(51)+CHR(45)    && line feed spacing 1/4"
     @ 1, 1  SAY  ' Grand Total Co., Ltd.  '
     @ 2, 1  SAY  ' Group of Merchandise Report  '
     PAGE_NUM = PAGE_NUM + 1
     @ 2, 65  SAY  'Page  : '
     @ 2, 76  SAY  PAGE_NUM     PICT  '99'
     @ 3,  1  SAY  REPL('-',79)
     @ 5, 10  SAY  '      Group Code      Group Name  '
     stor  7  to  row
     do whil row-7 < l_page .and. .not. eof()
       @ row, 21  say  GP_CODE  PICT  'XX'
       @ row, 32  say  GP_NAME  PICT  '@X25'
       row = row + 1
       skip
     enddo
     @ row,    1  SAY  REPL('-',79)
     @ row+1,  2  SAY  'Print Date : '
     @ row+1, 17  SAY  SysDTOTL(DATE())
```

```
    @ row+1, 65  SAY  'Time : '
    @ row+1, 72  SAY  Time()
    EJECT
   enddo &&
   SET DEVI TO SCREEN
   stor  2  to  fcitm
  ENDIF
  set colo to w/n,n/w
  @ 23, 0  clear  to  23, 79
  @ 23, 30  SAY  'F10  Print                    ESC  Exit '
RETURN
*-----------------------------------------
```

```
*----------------------------------------
      Procedure  MSTREPO3
*----------------------------------------
set talk off
set cursor off
set defa to dbfs
SELE 3
USE STMER INDEX STMER1
go bottom
botrec = recno()
go top
repfunc1 = ' F10  Print '  && F10=-9
repfunc2 = ' ESC  Exit'
store    1  to  fcitm
store    0  to  I      && i = row of record on display screen
store    0  TO  J      && j = number of record on display screen
store    1  to  k      && k = page of display screen
set colo to W/B,W/N
@ 0, 0  clear  to  21, 79
@ 1, 1  SAY  'Grand Total Co., Ltd.  '
@ 2, 1  SAY  '3. Structure of Merchandise Report  '
@ 3, 0  SAY  '-------------------------------------------------------------------------------- '
@ 5, 10 SAY  '       Structure Code     Structure  '
do dpftos3
set colo to w/n,n/w
@ 23, 0  clear to  23, 79
@ 23, 30  SAY  'F10  Print                    ESC  Exit '
fcl = stclrp21(fcitm)
itm = ltrim(str(fcitm))
set colo to n/w,w/n
@ 22, 0 clear to 22, 79
```

```
@ 23,fcl say repfunc&itm
set colo to w/n,n/w
store .t. to selact
do while selact
  A = 0
  DO WHIL A = 0
    A = INKEY()
  ENDDO
  if a=-9 .or. a=27 .or. a=4 .or. a=19 .or. a=18
    set colo to w/n,n/w
    @ 23,fcl say repfunc&itm
  endif
  DO CASE
    CASE A = -9   .OR. (A = 13 .AND. fcitm = 1)    && PRINT
      stor 1 to fcitm
      fcl = stclrp21(fcitm)
      itm = ltrim(str(fcitm))
      set colo to n/w,w/n
      @ 23,fcl say repfunc&itm
      set colo to w/n,n/w
      DO MSTPRNR3
    CASE A = 27 .OR. (A = 13 .AND. fcitm = 2)   && EXIT
      store .f. to selact
      EXIT
    CASE A = 4        && right
      set colo to w/n,n/w
      @ 23,fcl say repfunc&itm
      fcitm = fcitm + 1
      if fcitm = 3
        fcitm = 1
      endif
```

```
    CASE A = 19       && left
      set colo to w/n,n/w
      @ 23,fcl say repfunc&itm
      fcitm = fcitm - 1
      if fcitm < 1
        fcitm = 2
        @ 23,fcl say repfunc&itm
      endif
    OTHE
      A = 0
      * LOOP
  ENDCASE
  fcl = stclrp21(fcitm)
  itm = ltrim(str(fcitm))
  set colo to n/w,w/n
  @ 23,fcl say repfunc&itm
  set colo to w/n,n/w
enddo
RETURN
*-------------------------------
   Procedure  DPFTOS3
*-------------------------------
  sele 3
  set colo to W/B,W/N
  store  7   to  row
  do while .not. eof()
    if ft_code = '3'
      @ row, 23  say  ST_CODE  pict  'X'
      @ row, 35  say  ST_NAME  pict  '@X35'
    endif
    row = row + 1
```

```
    skip
  enddo
RETURN
*----------------------------------
    PROCEDURE MSTPRNR3
*----------------------------------
  PRTOUT = CHKPRT()
  IF PRTOUT
    SET DEVI TO PRINT
    SELE 3
    go top
    stor 30  to  l_page
    stor  0  to  page_Num
    do while .not. eof()
      @ 0, 0  SAY  CHR(27)+CHR(79)          &&  Letter 10 cpi
      @ 0, 0  say  CHR(27)+CHR(120)+CHR(1)    && LQ
      @ 0, 0  SAY  CHR(27)+CHR(51)+CHR(45)   && line feed spacing 1/4"
      @ 1, 1  SAY  ' Grand Total Co., Ltd.  '
      @ 2, 1  SAY  ' Structure of Merchandise Report  '
      PAGE_NUM = PAGE_NUM + 1
      @ 2, 65  SAY  'Page  : '
      @ 2, 76  SAY  PAGE_NUM     PICT  '99'
      @ 3,  1  SAY  REPL('-',79)
      @ 5, 10  SAY  '       Structure Code     Structure  '
      stor  7  to  row
      do whil row-7 < l_page .and. .not. eof()
        if ft_code = '3'
          @ row, 23  say  ST_CODE  PICT  'X'
          @ row, 35  say  ST_NAME  PICT  '@X35'
        endif
        row = row + 1
```

```
      skip
    enddo
    @ row,    1  SAY  REPL('-',79)
    @ row+1,  2  SAY  'Print Date : '
    @ row+1, 17  SAY  SysDTOTL(DATE())
    @ row+1, 65  SAY  'Time : '
    @ row+1, 72  SAY  Time()
    EJECT
  enddo &&
  SET DEVI TO SCREEN
  stor  2  to  fcitm
 ENDIF
 set colo to w/n,n/w
 @ 23, 0  clear  to  23, 79
 @ 23, 30  SAY  'F10  Print                    ESC  Exit '
RETURN
*-----------------------------------------------
```