



A COMPUTERIZED REVENUE CYCLE SYSTEM FOR A TOOLING CORPORATION

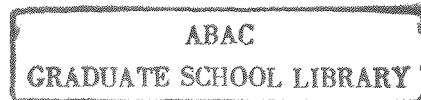
by

MR. KRITSADA DAMRONGPHOL

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

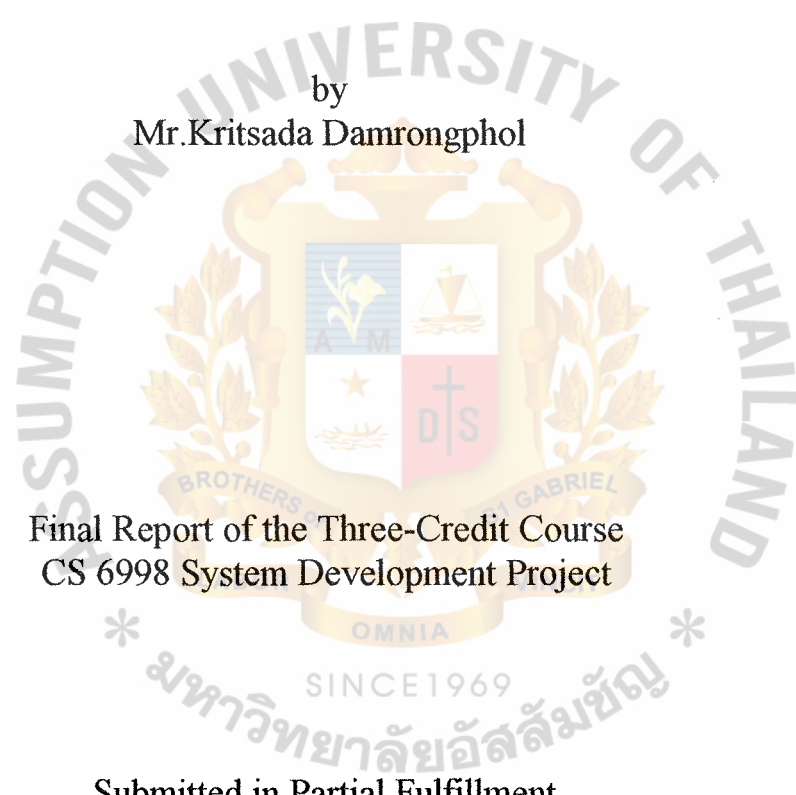
Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer Information Systems
Assumption University

November, 1996



A Computerized Revenue Cycle System
For a Tooling Corporation

by
Mr. Kritsada Damrongphol

A large, faint watermark of the Assumption University of Thailand seal is centered on the page. The seal is circular with a shield in the center. The shield is divided into four quadrants: top-left (blue with a white lily), top-right (white with a blue sailboat), bottom-left (white with a blue star), and bottom-right (red with a white cross and the letters "D S"). The shield is flanked by two golden lions. Below the shield is a banner with the text "BROTHERS OF THE HOLY GABRIEL". The outer ring of the seal contains the text "ASSUMPTION UNIVERSITY OF THAILAND" at the top and "มหาวิทยาลัยอัสสัมชัญ" at the bottom, with "SINCE 1969" and "OMNIA" in the center of the bottom arc.

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

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

November, 1996

Project Title : A Computerized Revenue Cycle System
For a Tooling Corporation

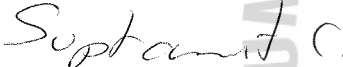
Name: Mr. Kritsada Damrongphol

Project Advisor: Dr. Suphamit Chittayasothorn

Academic Year: 1996

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

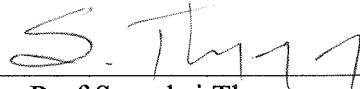
Approval Committee:


(Dr. Suphamit Chittayasothorn)
Advisor


(Prof. Dr. Srisakdi Charmonman)
Member


(Dr. Sudhiporn Patumtaewapibal)
Member


(Air Marshal Dr. Chulit Meesajjee)
Member


(Assoc. Prof. Somchai Thayarnyong)
Member

November, 1996

ABSTRACT

I. T. W.(Thailand) Co., Ltd, a retailing tool-work company, has been implemented a computerized revenue cycle system, which is mainly consisted of intergrated function of Sale, Inventory, Account Receivable and Collection System.

As a consequence of the rapid growth in construction operation, the company has realized the problems associated with currently manual operation of which are mainly due to:-

- The increasing routine transaction process done by Manual system.
- The cash flow deficit due to the company could not receive money from customer on time while, at the other hand, has to pay the creditor on exactly due date.
- The Number of days for collection of Account Receivable has been extended due to no updated report for Management Action.
- The inventory list was currently updated by stock card on the manual basis thus the balance show was not in a timely response basis.
- The redundancy of database has occurred since each department has its own database record, thus required multiple access and update.

With the implementing of Computerized system, it could get rid of above problems, of which it will result in efficiency and effectiveness in Revenue Cycle System.

In term of monetary analysis, the benefit that achieves is far ahead than the cost that spends. It reflects positive in Net Present Value as well as the Pay Back Period having been achieved in less than five years.

ACKNOWLEDGMENT

The project is intended to construct the computerized system into the integrated business company function. The result, as shown, could be seen in the efficiency, effectiveness as well cost reduction of the operation.

Without the full support and contribution by many persons during my preparation, this project is hardly finished in good shape.

I would like to convey my sincere gratitude to my advisor, Dr. Suphamit Chittayasothorn, who provides many guidances toward the successful completion of this project., our M.S. (C.I.S.) Lecturers, ABAC Librarians, Andersen co-workers and my beloved mother, who kindly support many invaluable suggestions and encouragements.

Any comments or suggestions are being advisable, for the outstanding delevrance of this project.

Kritsada Damrongphol

CONTENTS

ABSTRACT	I
ACKNOWLEDGMENTS	II
1. INTRODUCTION	1
1.1 Background of Project	1
1.2 Project Objective	3
1.3 Scope of Project	4
1.4 Deliverables	5
1.5 Project Plan	6
2. EXISTING SYSTEM	
2.1 Background of the Organization	7
2.2 Existing Business Functions	9
2.3 Current Problems and Areas for Improvement	11
2.4 Existing Computer System	14
3. THE PROPOSED SYSTEM	
3.1 Users Requirements	15
3.2 System Design	17
3.3 Hardware Requirement	26
3.4 Software Requirement	27
3.5 Security and Control	28
3.6 Cost / Benefits Analysis	29
3.7 Network Design	33

4. PROJECT IMPLEMENTATION	
4.1 Overview	35
4.2 Test Plan and Result	35
5. CONCLUSION AND RECOMMENDATION	
5.1 Conclusion	37
5.2 Recommendation	38
REFERENCES	39
APPENDIX A: Data Flow Diagram	40
APPENDIX B: Structure Chart	48
APPENDIX C: Data Dictionary	55
APPENDIX D: Process Specification	61
APPENDIX E: Module Specification	75
APPENDIX F: Database Management Design	86
APPENDIX G: Screen Design	90
APPENDIX H: Report / Hard Copy Design	105
APPENDIX I: Program Coding	120

LIST OF FIGURES

Figure 2.1: Organization Chart	8
Figure 2.2: Existing System: Context Diagram	12
Figure 2.3: Existing System: Data Flow Diagram Level 0	13
Figure 3.1: Context Diagram	20
Figure 3.2: Data Flow Diagram Level 0	21
Figure 3.3: Top Level of Structure Chart	23
Figure 3.4: ER Model Mapped to Relational Data Model (5NF)	25
Figure 3.5: Hardware and Software Estimation Cost	30
Figure 3.6: Benefit Versus Cost and its Cumulative Benefits	31
Figure 3.7: Network Layout	34
Figure A-1: Data Flow Diagram Level 1: Response to Customer	41
Figure A-2: Data Flow Diagram Level 1: Generate Cash Sale	42
Figure A-3: Data Flow Diagram Level 1: Generate Credit Sale	43
Figure A-4: Data Flow Diagram Level 1: Update Sale	44
Figure A-5: Data Flow Diagram Level 1: Update Inventory	45
Figure A-6: Data Flow Diagram Level 1: Update Account Receivable	46
Figure A-7: Data Flow Diagram Level 1: Collect Money	47
Figure B-1: Structure Chart: Cash Sale Process	49
Figure B-2: Structure Chart: Generate Credit Sale	50
Figure B-3: Structure Chart: Credit Sale Collection Process	51
Figure B-4: Structure Chart: Response to Customer Request	52
Figure B-5: Structure Chart: Update Inventory Process	53
Figure B-6: Structure Chart: Generate Report	54
Figure F-1: Entity Relationship Diagram	86
Figure F-2: ER Model Mapped to Relational Data Model	87
Figure F-3: Functional Dependencies	88

Figure G-1: Generate Cash Sale - Main Menu Screen	91
Figure G-1-1: Generate Cash Sale - Customer Menu Screen	92
Figure G-1-2: Generate Cash Sale - Cash Invoice Menu Screen	93
Figure G-1-3: Generate Cash Sale - Delivery Order Menu Screen	94
Figure G-1-4: Generate Cash Sale - Official Receipt Menu Screen	95
Figure G-2: Generate Credit Sale - Main Menu Screen	96
Figure G-2-1: Generate Credit Sale - Delivery Order Menu Screen	97
Figure G-2-2: Generate Credit Sale - Invoice Menu Screen	98
Figure G-2-3: Generate Credit Sale - Delivery Order Menu Screen	99
Figure G-3: Generate Credit Sale-Collection Menu Screen	100
Figure G-3-1: Credit Sale-Collection-Official Receipt Menu Screen	101
Figure G-4: Display Inventory Menu Screen	102
Figure G-5: Receipt-In Inventory Menu Screen	103
Figure G-6: Generate Report Menu Screen	104
Figure H-1: Sale Report	105
Figure H-2: Customer Report	106
Figure H-3: Inventory Report	107
Figure H-4: Receipt Report	108
Figure H-5: Account Receivable Outstanding Balance Report	109
Figure H-6: Account Receivable Aging Balance Report	110
Figure H-7: Sale By Product Report	111
Figure H-8: Sale By Customer Report	112
Figure H-9: Customer Division Report	113
Figure H-10: Customer Segment Report	114
Figure H-11: Gross Profit Analysis Product Report	115
Figure H-12: Invoice	116
Figure H-13: Delivery Order	117
Figure H-14: Cash Sale Invoice	118
Figure H-15: Official Receipt	119

1. INTRODUCTION

1.1 BACKGROUND OF THE PROJECT

Illinois Tools Works (I.T.W.) Thailand is the subsidiary of the renowned worldwide I.T.W in U.S.A. Starting from scratch, the company engaged only an agent company which required only a few transactions. However, due to the rapid growth in the construction business in Thailand, the company has to expand to act as a branch in order to serve said growth.

The company operates mainly by importing tools work from U.S. such as Electrical drill, Coating machine, Air pump as well as necessity material required by the customer.

As a consequence of the construction growth, the company enjoyed more sales, however the management has realized that the current system especially the revenue cycle starting from the initiate of sales, receivable balance, the cut-off inventory till the collection of money could not satisfy due to the following mainly reason:-

- No Sales report showing the current customer, sales amount.
Existing report has to summarize from accounting department which was done on request.

- The company extended number of days for collection since no updated report for the showing of status.
- Lack of proper documents to collect the money.
- The inventory list was currently updated by stock card on the manual basis thus the balance show was untimely basis and mismatch the type of issue and receipt of stock.
- The time of receipt money was slow while the company has to pay the balance due on time.
- The Redundancy of database has occurred since each department has its own database record, thus require multiple access and updated.

The Management has come to pay a major role in revolution the manual system into computerized system. With the implement system, each department could share the centralized database through Local Area Network (LAN). The real time access has been achieved thus refected in up-to-date information as well as timely response by Management action.

1.2 OBJECTIVE

Illinouise Tools Work (I. T. W.) revenue cycle system can be derived as follows

1. To analyze the existing Revenue cycle System (Sale System, Account Receiable System, Inventory System as well as Collection System.)
2. To clarify the problem of existing System
3. To design the integrated database and implement computerized system on all revenue cycle system.
4. To design the computerized system based on the basis needs of the user as well as met the standard required by Illinouis Tools Work -head office
5. To determine the cost versus benefit that would arise if the new system has been implemented.

1.3 SCOPE

The scope of I. T. W. Revenue Cycle can be indicated as follows:-

Sales Function

1. Retrieve and update Customer Information
2. Retrieve and update Inventory Balance
3. Determine Cash or Credit Sale
4. Generate Invoice and Delivery Order
5. Generate Total Sale report

Account Receivable function

1. Set-up Account Receivable when credit sale has been initiated.
2. Update Account Receivable of particular customer when receiving collection.
3. Generate Account Receivable Outstanding Report and Account Receivable-Aging Report on timely basis.

Collection function

1. Create Official Receipt to Customer.
2. Generate Collection Report on timely basis.

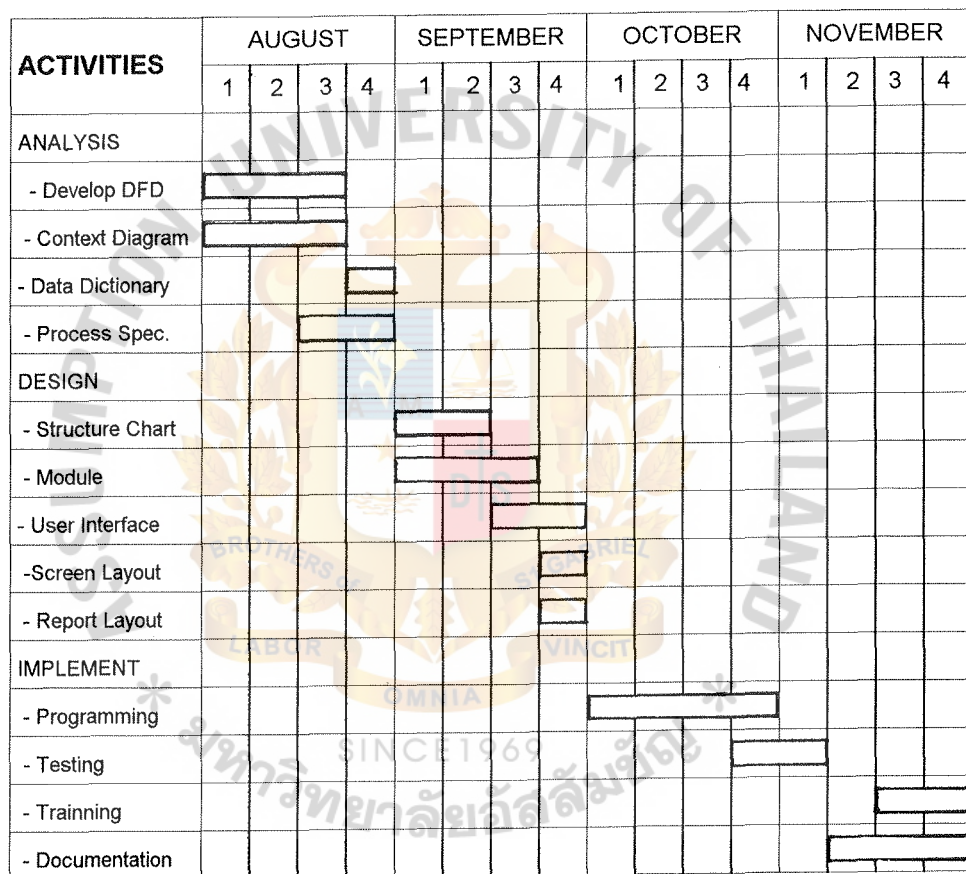
1.4 DELIVERABLES:-

The deliverables of I. T. W. Revenue Cycle System could be as follows:-

1. Screen Layout for user interface
 - Main Menu
 - Cash Sale
 - Credit Sale
 - Credit Sale Collection
 - Inventory
 - Customer
 - Generate Report
2. Application program
3. Hard Copy format
 - Sales report
 - Customer report
 - Inventory report
 - Receipt Report
 - Account Receivable-Outstanding balance report
 - Account Receivable-Aging balance report
 - Sale By Product Report
 - Sale By Customer Report
 - Customer- Division Report
 - Customer-Segment Report
 - Gross Profit Analysis by Product Report

1.5 PROJECT PLAN

THE TIME OF PROJECT
COULD BE
DELIVERABLE
AS FOLLOWS:-



PROJECT PLAN :
GANTT CHART

2. EXISTING SYSTEM

2.1 BACKGROUND OF THE COMPANY

I.T.W (THAILAND) , a subsidiary of the renowned I. T. W. group in U.S., performs a retailing business related construction tools throughout the kingdom. The company mainly imported the tool works from U.S and distributed its product to its customer. The product mainly covers Electrical drill and screw. Presently, company structure has been implemented from the day when it starts acting as an agent that could not response to the huge amount of transactions.

The company has only one office together with the warehouse which was rental at the Port. Each Salesperson will be responsible for each sale till the days of collection. The sale may be generated through cash sale and credit sale, however in normal case the sale is on the 30 days credit.

The normal procedure of I. T. W. revenue cycle is that

1. The customer initiated its sale through salesperson at the at the office or by phone
2. The Stock Department issue the release document to warehouse which later consequently issue the Delivery Order to customer.
3. The credit Department has to record its credit sale transaction and follow-up the collection on the due date.
4. Accounting Department records the transactions and get a summary report in Sale , Account Receivable and its collection.

I. T. W. Organization Chart is illustrated as shown in figure 2.1

ILLINOIS TOOLS WORK
(I.T.W.THAILAND)

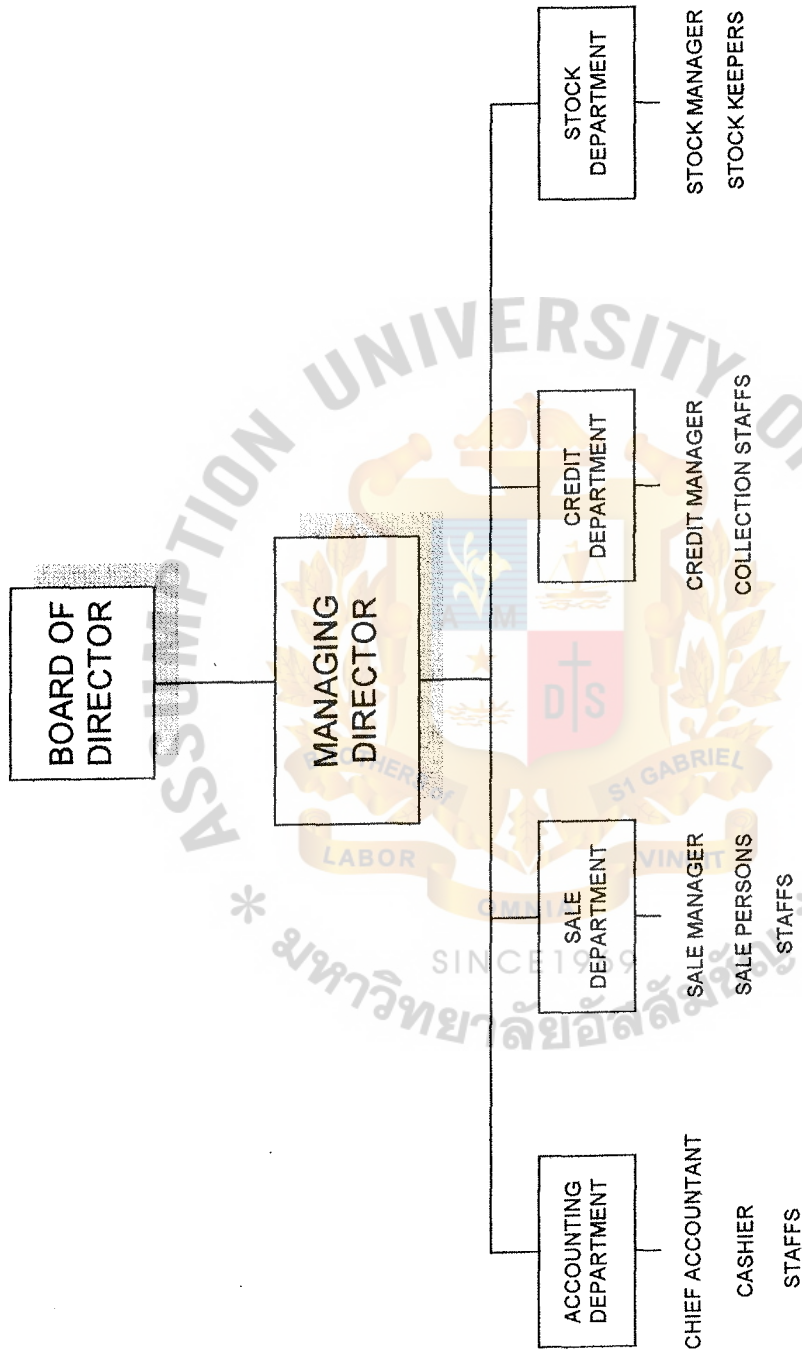


FIGURE 2.1 :
ORGANIZATION
CHART

2.2 EXISTING BUSINESS FUNCTION

The company comprises of 4 departments which consisted of:-

2.2.1 Sale Department :

- is responsible for sale function as well as promotion of the company's products.
- determine the credit status of the customer.
- check the availability of the product sold from the stock card summary prepared by stock department.
- create invoice as well as delivery order to the customer.
- update the monthly sale report.

2.2.2 Credit Department

- update the company's credit sale on account receivable report from issued invoice.
- monitor the status of account receivable report
- pursue the collection from customer.
- update the account receivable report when collection has been made.
- retrieve the account receivable status when requested.

2.2.3. Stock Department

- receiving and issuing the stock to/from warehouse.
- prepare the delivery order from stock release note.
- update the stock inventory report at the end of the day
- reconcile the amount of stock as well as verify the availability balance.

2.2.4 Accounting Department

- record the company's transaction entries.
- monitor the collection process.
- reconcile the discrepancy balance among various departments.
- prepare the monthly sale report, account receivable report, collection report as well as inventory report on accounting purpose
- prepare the financial statements for top-level of management and shareholders.



2.3 CURRENT PROBLEMS AND AREAS FOR IMPROVEMENT

- 2.3.1 The inventory report is not updated, thus no current status of inventory to ensure the availability of balance
- 2.3.2 Lack of proper documents to create the internal control among departments.
- 2.3.3 Lack of proper report for Account Receivable thus
- the collection of money has to delay, not on the due date, while the company has to pay the debt on time.
 - some collection has been received but the report is not updated.
- 2.3.4 Huge amount of transactions thus causing workload among company's staffs as well as required report has not been updated on time.
- 2.3.5 The differentiate of balance among each departments which have their own departments e.g. the balance of account receivable of credit department reflects the uncollectible amount while the accounting department report already reflects actual collection.
- 2.3.6 The delay of required report submit to the headquarter in U.S.
- 2.3.7 The cost of recruiting new employee is quite high and workload fluctuate from time to time especially at month ended.
- 2.3.8 Management could not monitor the company's current status and operation on time.
- 2.3.9 The redundancy of Database since each department has its own database record, thus requires multiple access and updates.

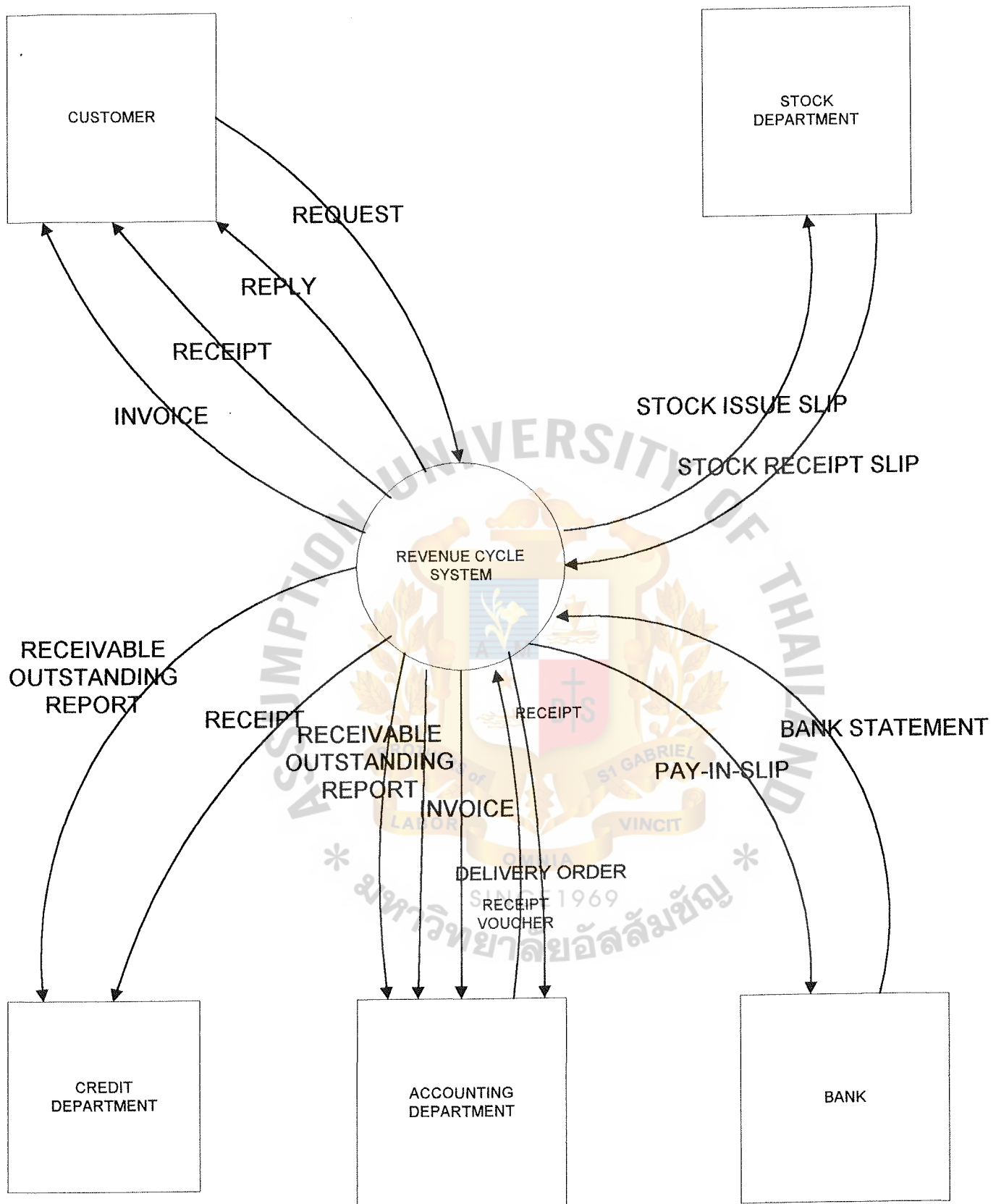


figure: 2.2
existing system:
context diagram

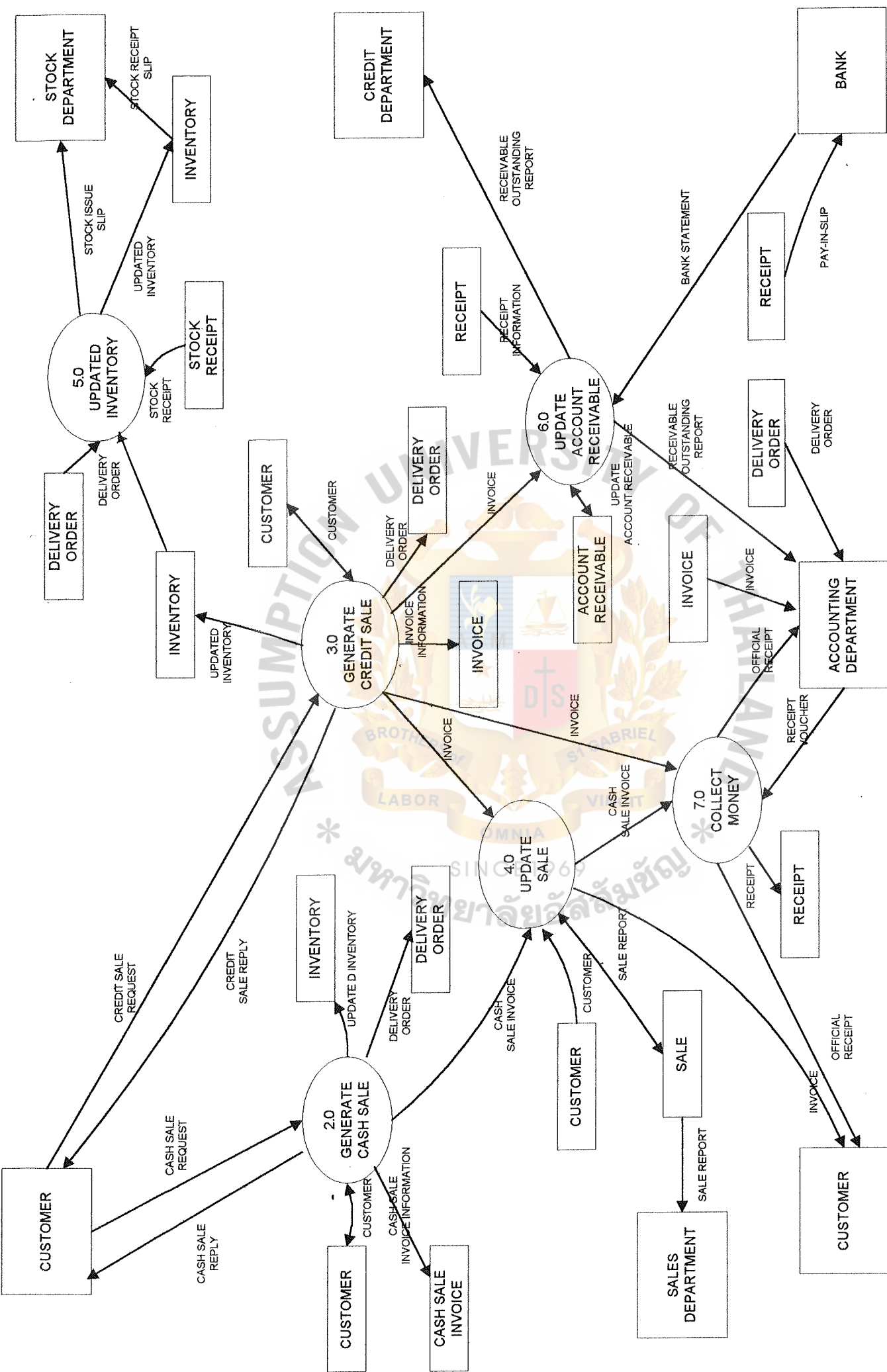


FIGURE 2.3: EXISTING DATA FLOW DIAGRAM LEVEL 0

2.4 EXISTING COMPUTER SYSTEM

<u>2.4.1 Sale Department</u>	- 2 stand-alone P. Cs. of which mainly product sale report as well as create invoice, delivery order and updates customer record.
<u>2.4.2 Credit Department</u>	- 2 stand-alone P. Cs. which prepare account receivable report as well as update account receivable balance.
<u>2.4.3. Stock Department</u>	- 1 stand-alone P. Cs. for producing inventory report.
<u>2.4.4. Accounting Department</u>	- 2 stand-alone P.C.s using Accounting Software for generate the accounting transaction entries to general ledger system.

3. PROPOSED SYSTEM

3.1 USER REQUIREMENTS

The proposed system has four parts, but it combines as an integrated system. It could be divided as following part:-

Features of Sale Portion

1. interactive inquiry capability to respond to customer request.
2. Customer verification
3. Product selected verification
4. Type of sale selection (credit sales/ cash sales)
5. Automatic cut-off stock
6. Calculation of Sale amount
7. Computer preparation of documents: invoice, delivery order
8. Computer Preparation of reports: Sales Summary, Customer report.

Features of Account Receivable Portion

1. Automatic adding all charges for sale amount to the proper customer account
2. Interactive inquiry capability to repond to customer account status information
3. Prepare aging of customer balances
4. Computer preparation of account receivable outstanding balance, showing all charges, payments, and adjustments

4. Computer preparation of account receivable outstanding balance, showing all charges, payments, and adjustments process as well as aging balance (current, 30days , 60 days or over)
5. Automatic preparation of management reports to show account balances together with sales and payments activities.

Features of Inventory Portion

1. Interactive inquiry capability to response to inventory status information
2. Automatic cut-off and up-dated inventory information
3. Computer preparation of inventory report showing the outstanding balance as well as the amount.

Features of Collection Portion

1. Automatic preparation of Receipt
2. Automatic invoice Matching
3. Automatic updated to relevent Account Receivable portion
4. Automatic preparation of collection summary

3.2 SYSTEM DESIGN

3.2.1. Data Flow Diagrams

The proposed system can better be represented by dividing the whole system into 6 processes as follows:

Process 1.0: Response to customer request

Responding to customer request of the availability of inventory, the salesperson can retrieve from the updated inventory database at a real time access.

Process 2.0: Generate Cash Sale

With the entering inventory required that is available in the inventory list and the customer would like to pay the amount immediately, then the cash sale process is selected.

if customer is new, customer is requested to enter the update customer mode by filling the customer information. If that customer is existed customer, only customer number is required. Inventory is automatically cut-off from relevant inventory database and Sale amount is calculated. The printing output is Delivery Order which is submitted to Stock Department.

Process 3.0: Generate credit sale

The alternative system design if those customer would like to enter credit sale which I.T.W. offers 30 days of credit. With the inventory selected by customer. The update of customer data is requested if he is a new customer. Then the system will automatically cut-off stock from relevant inventory database .

Sale is calculated as well as the amount will automatically updated to Account Receivable database. The printing output is Invoice which is submitted to customer as well as Delivery Order which in turn submitted to Stock Department.

Process 4.0: Update Sale

With the sale process from cash sale and credit sale, each transaction will be automatically updated to Sale Summary. The report will print out as request by Sale Department as well as Management level showing the sale status at a particular time.

Process 5.0: Update Inventory

When the Delivery Order has been initiated. It will automatically cut-off the inventory. In turn, if the stock has been received it will automatically updated the stock in particular stock number.

The System will have a particular real time to response to the salespersons or customers . Then if request by Stock Department or Management Level the report will be printed showing the inventory description, amount as well as product availability.

Process 6.0: Update Account Receivable

When the Credit Sale has been initiated, it will automatically update the Account Receivable database. This process also concern the collection of money which will match the amount of collection with the invoice then it will also automatically update the Account Receivable balance. The print output is Aging report and Account Receivable outstanding balance.

Process 7.0: Money Collection

The Money Collection process starting when Cashier receive money from customer then the Receipt Voucher will create for the internal control, then with the Receipt Voucher the system will automatically create the Receipt which will submitted to the customer.

The above Data Flow Diagrams are illustrated in Appendix : A

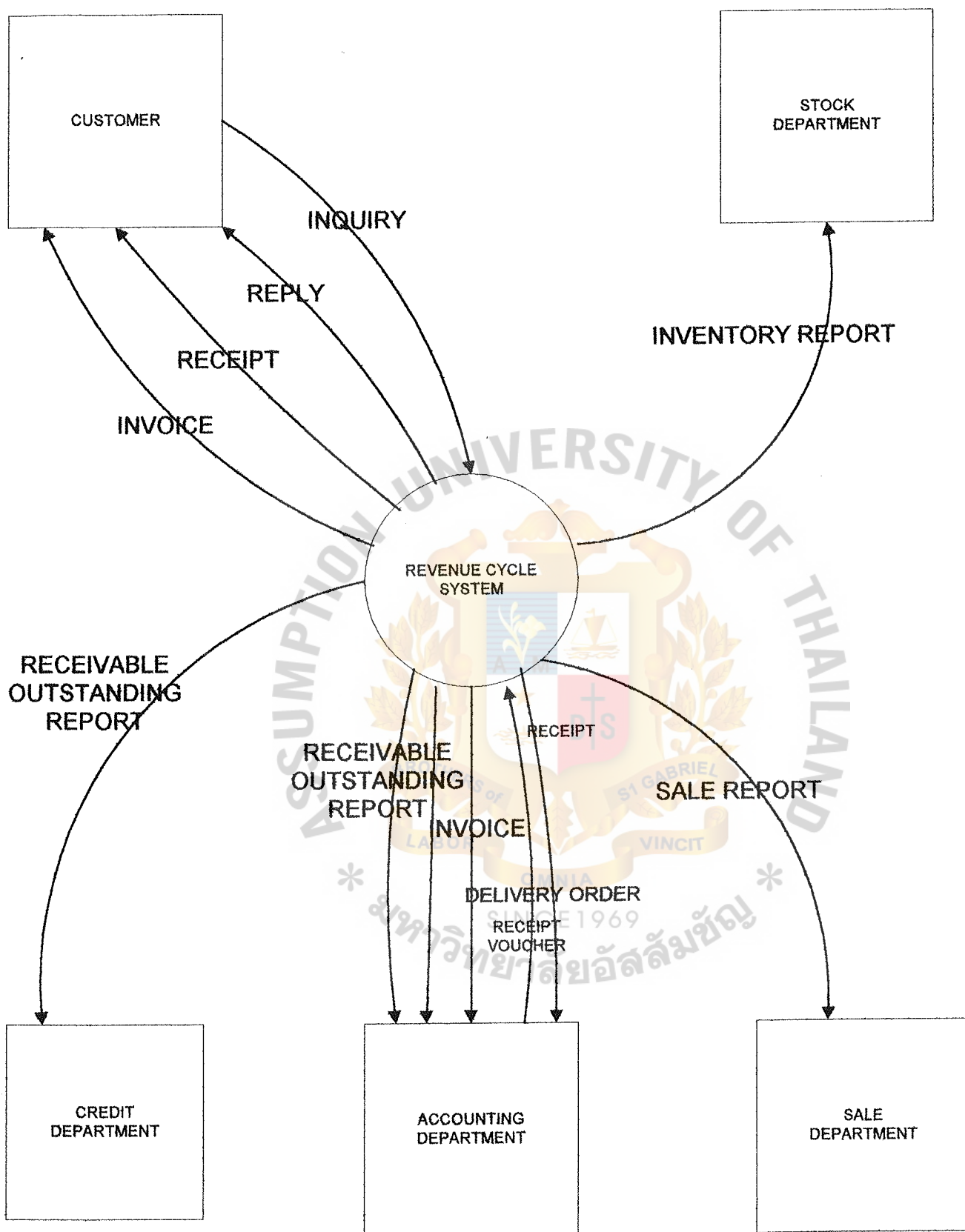


FIGURE 3.1: CONTEXT DIAGRAM

3.2.2 Structure Chart of I. T. W can be divided into following categories:-

- Generate Cash Sale
- Generate Credit Sale
- Generate Credit Sale Collection
- Response Customer's request
- Update inventory
- Generate Report

All the above structure charts are illustrated in (Appendix: B)

3.2.3 Data Dictionary (Appendix C)

3.2.4 Process Specification (Appendix D)

3.2.5 Module Specification (Appendix E)

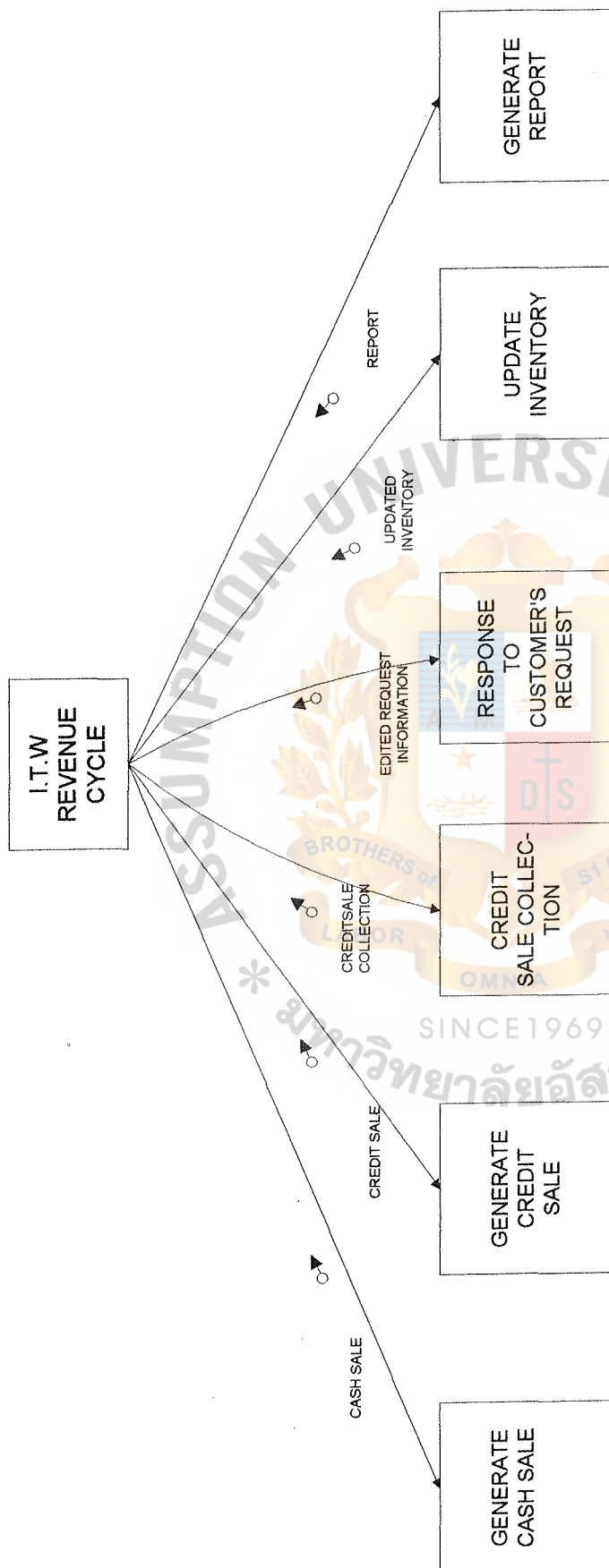


FIGURE 3.3 : TOP LEVEL OF STRUCTURE CHART

3.2.6 Database Management Design.

In order to extract the redundancy, the 5NF: fifth normal form has been designed.

- All data items contain atomic value only
- No non-key attribute
- No transitive Functional Dependencies between primary key and non-key attributes.
- No multiple candidate key where those candidate keys are composit and candidate keys are overlapped.
- No Multivalue dependence.

The illustration of Entity Relationship Model (ER) and its mapping to relational data model are illustrated in (Appendix F)

3.2.7 - Screen Design Listings (Appendix G)

3.2.8 - Report Design Listings (Appendix H)

3.2.9 - Program Coding (Appendix I)

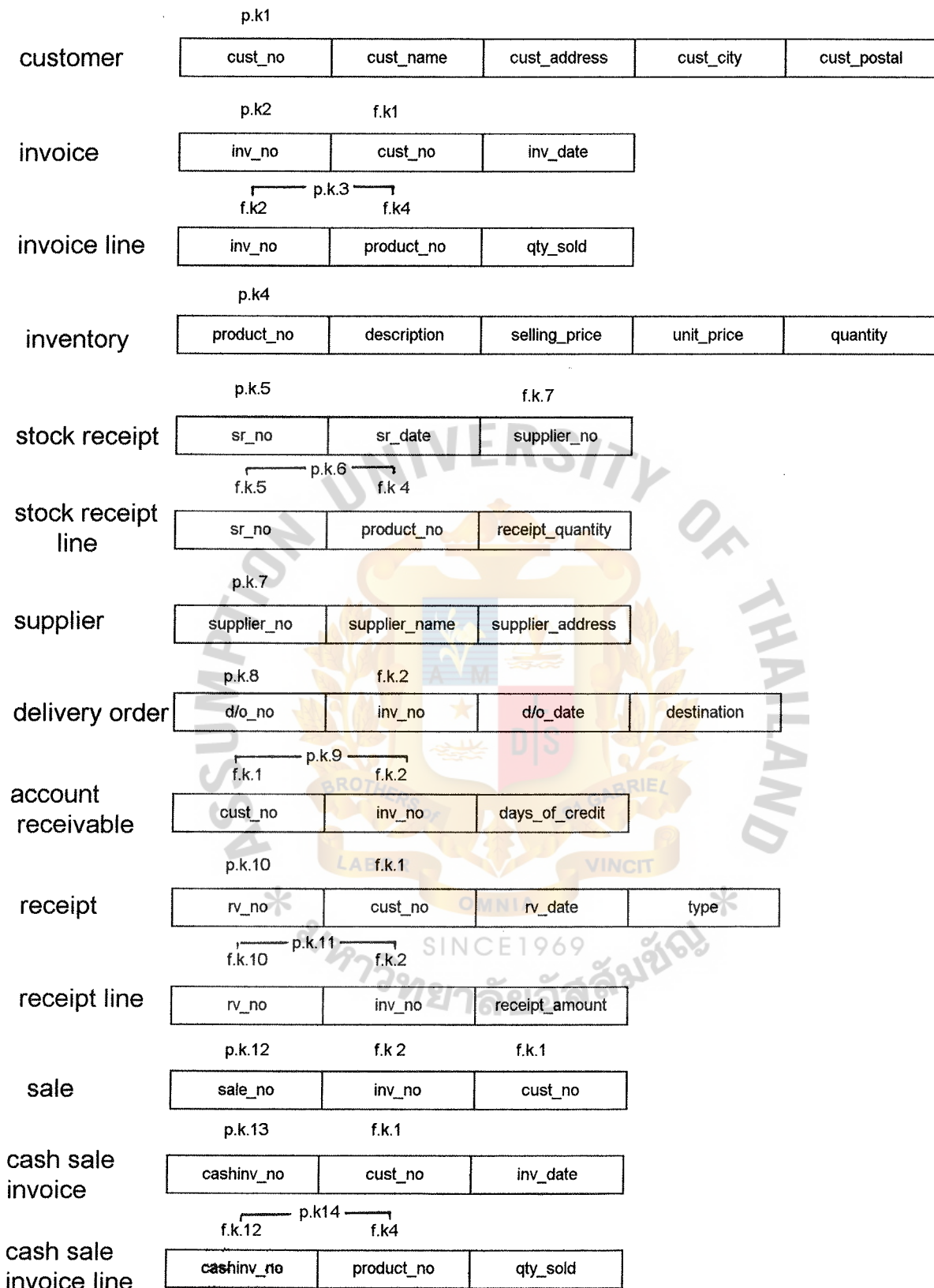


figure 3.4:
ER Model Mapped to Relational Data Model (5 NF)

3.3 HARDWARE REQUIREMENT

SERVER - COMPAQ PROLIANT 1500R

- Pentium 5/100
- Board and Chip upgradable
- 256-KB secondary write-back cache standard
- FlexSMP System Architecture, combining EISA and PCI buses
- 16MB of RAM expandable to 208 Mbtes
- ECC (Error checking and Correcting) memory
- Preinstalled NetFlex-2 ENET-TR Controller
- Intergrated 32 Bit Fast-Wide SCSI-2 Controller
- Integrated 1024 x 768 video graphics
- Preinstalled CD-ROM Drive
- Ships standard with SmartStart and Compaq Insight Manager
- Standard 19-inch rack-mountable form factor Communications platform
- Eight total storage device baysne
- Nine total slots: one processor, five EISA, one shared EISA/PCI
- Automatic Server Recovery-2 (ASR-2) and health logs
- Network OS Support: Netware, UNIX, Windows NT, OS/2, Lan Manager, VINES

WORKSTATION: PENTIUM MERCURY- 560 M

- Pentium chip 100 Mhz
- RAM 8 MB (EXP. to 192 MB)
- HardDisk 540 MB
- SVGA 14"

3.4 SOFTWARE REQUIREMENT

- Windows NT
- DOS 6.2
- Microsoft Windows for Workgroup
- Microsoft Office for Workgroup
 - Microsoft Word
 - Microsoft Excel
 - Microsoft Access
- Visual Basic Version 4



3.5 SECURITY AND CONTROLS

The system is enforcing the security not to be intercepted by any intruder by setting up the controls as follows:

HARDWARE

- LAN server will be kept under computer room which is access by authorized officer.
- All diskett will be stored in a locked container.

DATA SECURITY

Input Control

- Password will be required for each user to login.
- Password will automatically changed in each month.
- User are not allowed to log-in the menu that do not get involved.
- the review of log-in history will be performed daily.
- The ex-staff will be informed immediately and password will be discarded.

Processing Control

- Electric shock prevention will be implemented.
- Automatically Sorting number will be implemented.
- Exception report will be produced when error occurs.

Output Control

- Any discarded printout will be destroyed as a regulation.
- All files will be backed up.
- Log-in history will be printed out and reviewed on a timely basis.

3.6 COST AND BENEFIT ANALYSIS

SYSTEMS START-UP COSTS

Systems analysis (240 hr)	Baht	120,000
Systems Design (180 hr)		90,000
Programming		45,000
Development and implementation		35,000
Out-of - pocket expense		10,000
Hardware and Software cost		520,000*
Training		90,000
Site Preparation		<u>12,000</u>

Total Start-up cost 922,000

**please see details on page 30 -*

ANNUAL SYSTEMS OPERATING COSTS

Supplies	240,000
Operating Cost	150,000
Maintainance Cost	<u>60,000</u>
Total Annual Systems Operating costs	<u>450,000</u>
	<u>1,372,000</u>

SYSTEMS BENEFIT

Savings on additional personnel not needed (7 persons : 120,000 each)	<u>840,000</u>
--	----------------

ANNUAL OPERATING SAVINGS

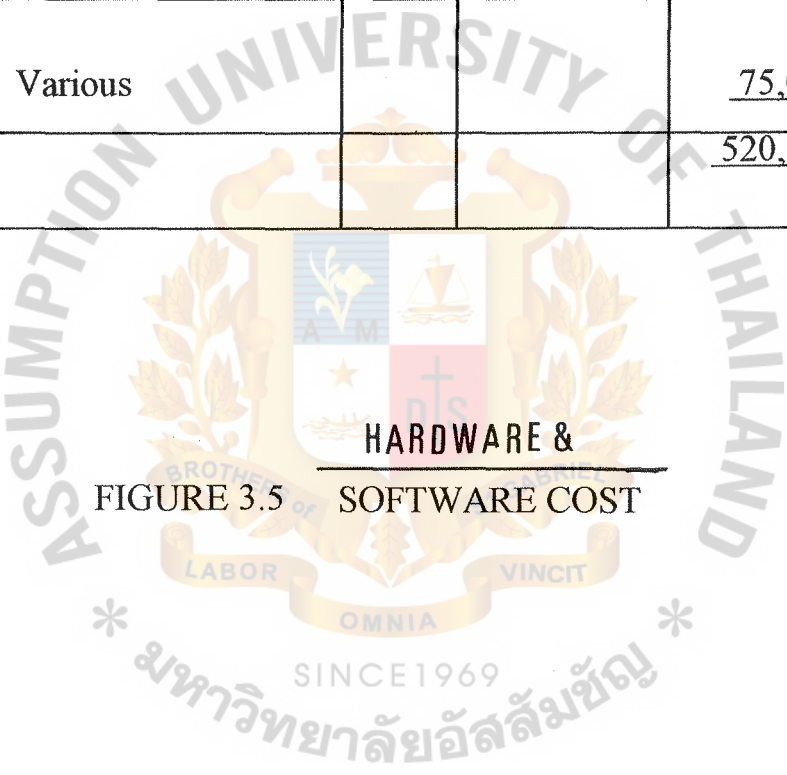
- Elimination of Manual Operation	600,000
- Elimination of stationery and documents	<u>150,000</u>
	<u>750,000</u>

Total Tangible system benefit 1,590,000

Plus: intangible benefit (faster collection, reduce redundancy of database, better customer relation and increase in sale effort)

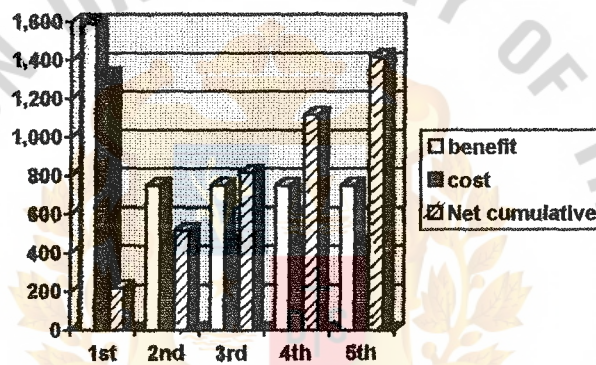
HARDWARE AND SOFTWARE COST ESTIMATION

Type	Description	item	unit cost	Total
File server	COMPAQ 1500R	1	210,000	210,000
Terminal	Pentium 100	4	25,000	100,000
Printer	HP-laser 4	1	25,000	25,000
	OKI 11433	4	12,500	50,000
Software	Window NT	1	50,000	50,000
	Microsoft Workgroup	1	10,000	10,000
Accessories and others	Various			<u>75,000</u>
TOTAL				<u>520,000</u>



 HARDWARE & SOFTWARE COST
 FIGURE 3.5

year	benefit	cost	Net Cumulative
1	1,590,000	1,372,000	218,000
2	750,000	450,000	518,000
3	750,000	450,000	818,000
4	750,000	450,000	1,118,000
5	750,000	450,000	1,418,000



AMOUNT: SHOWN IN THOUSAND BAHT

FIGURE 3.6 : Benefit Versus Cost and its Cumulative Benefits

PAYBACK PERIOD TO INSTALL THE PROPOSED SYSTEM

$$\text{Payback Period} = \frac{I}{(1 - T) * R}$$

where I = Investment or Capital Expenditure

T = Corporate Tax rate in percent

R = Average annual return on investment

$$\begin{aligned} \text{Payback Period} &= \frac{922,000}{(1 - .30) * 300,000} \\ &= 4.3 \text{ years} \end{aligned}$$

PRESENT VALUE OF THE PROPOSED SYSTEM

$$\text{NPV} = \frac{R_1}{(1+K)} + \frac{R_2}{(1+K)^2} + \frac{R_3}{(1+K)^3} + \frac{R_4}{(1+K)^4} + \dots + \frac{R_n}{(1+K)^n} - \text{PV}$$

where NPV = Net Present Value

PV = Cost of the Proposed System

R = Cash flow (Saving of new System)

K = Cost of money (Interest Rate)

n = Number of years

$$\frac{218,000}{(1.07)} + \frac{300,000}{(1.07)^2} + \frac{300,000}{(1.07)^3} + \frac{300,000}{(1.07)^4} + \frac{300,000}{(1.07)^5} - 1,222,000$$

$$(203,738.32 + 262,031.62 + 244,889.36 + 228,869.61 + 213,869.12) - 922,000$$

$$\text{NPV} = 231,398.03$$

Since it represents positive balance, so it is worth for its investment
in proposed system

3.7 NETWORK DESIGN

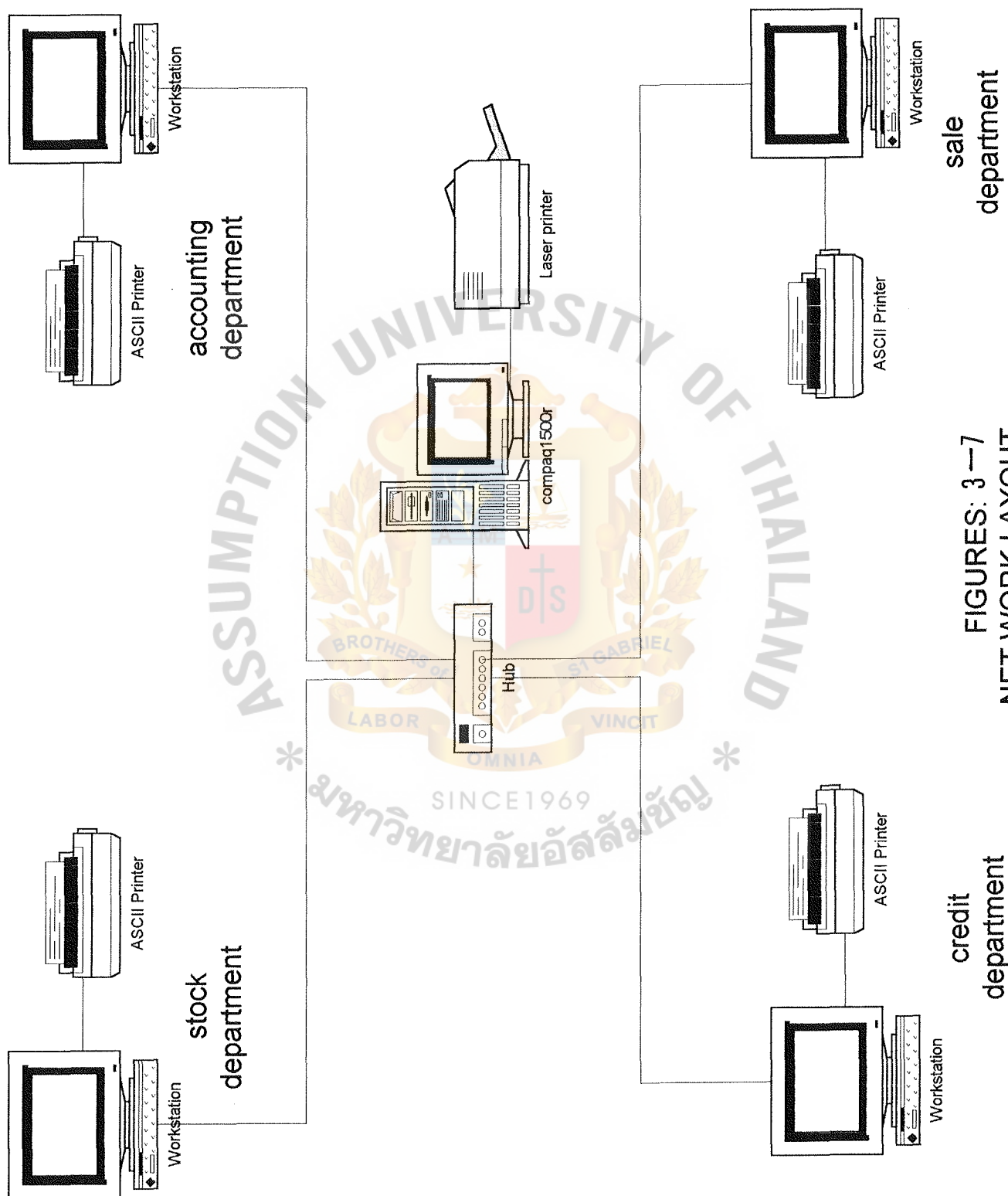
Local Area Network is implementing for I.T.W. Revenue Cycle of which the Central Workstation containing the main database is shared among the attached Terminal.

Star topology is required since a point to point link between each device and the central node.

The benefit is that

- Maintenance and Service is quite easier and regular in the market.
- It is very common in implementation . Existing wiring can be used.
- The System can easily be expanded , just patching the additional cables into the network at the wiring closet.
- Cost of implementation is inexpensive and the cost saving will be shifted to increase the speed of processing and System Control.

The network layout is represented as follows:-



FIGURES: 3-7
NET WORK LAYOUT

4.PROJECT IMPLEMENTATION

4.1 OVERVIEW

All control will be emphasized on the Project implementation,the project implementation can be divided into two phase.

- Work Performance Phase: The progress will be closely monitored and divided into sub-unit. Eash sub-unit will be acknowledged by customer for any progress.
- Time and Cost Phase: Punctuality is major concerned for implementation. Any variance of time and cost will be pre-mature discussed.
- The deliverable of our products is illustrated on GANTT chart Table which shown on Page

4.2 TEST PLAN AND RESULT

Testing will be done on phase by phase, the trial would be done by Hypothesis Data, any variance from users desires will be priority be corrected. The result of Testing will be represented in documentation with acknowledgement by responsible users. The major procedure can be described as follows:-

- Ascertainning that the new work flow actually run smoothly
- Determining that the output is correct , agreeing to the understanding of user.

4.3 CONVERSION

The New System will be converted into company by parallel run in order to ensure that no interruption will occur. The error detected (if any) will be firstly rectified, thus with the parallel run together with the old system, it will be provided most security.

4.4 TRAINING

To ensure the upmost efficient and effective system, the training will be conducted during the initial stage of trial test.

The Training will mainly focus as follows:-

Training Objective	- To provide user with clear understanding of I.T.W Revenue Cycle.
Training Method	- Each Department Staff will be divided into Sub-group, Lecture and presentation will be conducted together with the self-test by user.
Training Material	- All staff will be given a easy-to-read handbook comprising a reason why it is time to change and step-by-step of how to use.

5. CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

I. T. W. Revenue Cycle System can replace the current system by mean of :-

- Most applicable to the user's current system
- Fully Computerized System with accurate and real time response.
- Provide each department with user's requirement as well as a must report for Management critical judgement.
- Saving time in generating required procedure than current system.
- User's friendly environments.
- Apply the same standard format required by I. T. W. Head Office.
- Integrated database which is extract redundancy and multiple update and access.

With the help of I. T. W. Revenue Cycle, the company could remain its competitive edge in Selling System. Fast, reliable , secured and critical necessary process will surely provide the I. T. W. , sustainable growth.

5.2 RECOMMENDATION

The I. T. W. Revenue Cycle is open and flexible system which can better be *integrated* further in the following section:

- I. T. W. Expenditure Cycle
- I. T. W. Accounting System
- I. T. W. Internal Control Software provided by Internal Auditor.
- Any System required by I. T. W. Head Office.

With the rapid growth in information technology, this system can be modified well with the Upgrade Workstation and Electronic Data Intechange that will link with the major Supplier and Head Office.

REFERENCE

1. Kendall, Kenneth E, Kendall E. Julie, *System Analysis And Design*: New Jersey, Prentice Hall, Inc, 1992
2. Page-Jones, Meilir, *The Practical Guide to Structured System Design*, London; Yordan Press, 1988
3. Stephen A. Moscovice, Mark G. Simkin, Nancy A. Bagranoff, *Accounting Information Systems*; Wiley Press
4. Senn, James A, *Analysis & Design of information System*, New York; McGraw-Hill Publishing Company, 1989
5. Yourdan, Edward, *Modern structured Analysis*, New York: Prentice Hall Inc., 1989



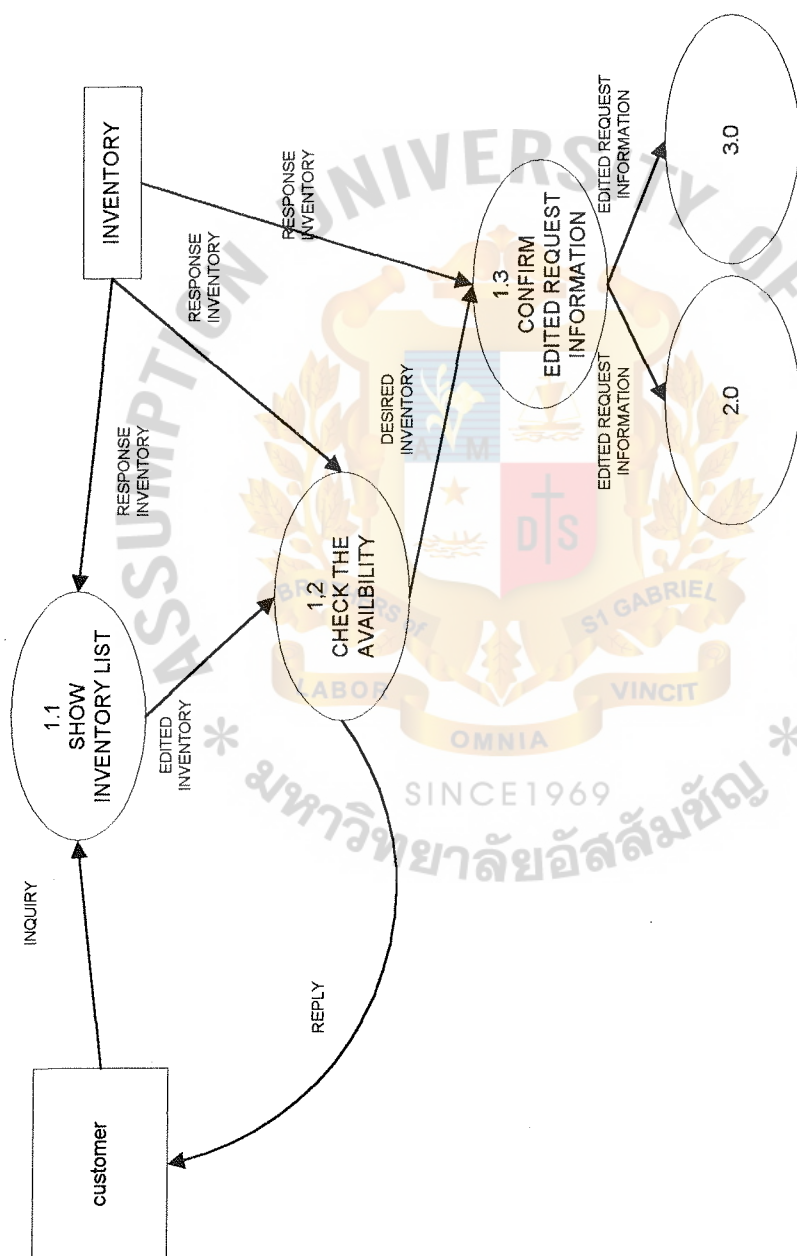


FIGURE A-1: DATA FLOW DIAGRAM LEVEL 1: RESPONSE TO CUSTOMER

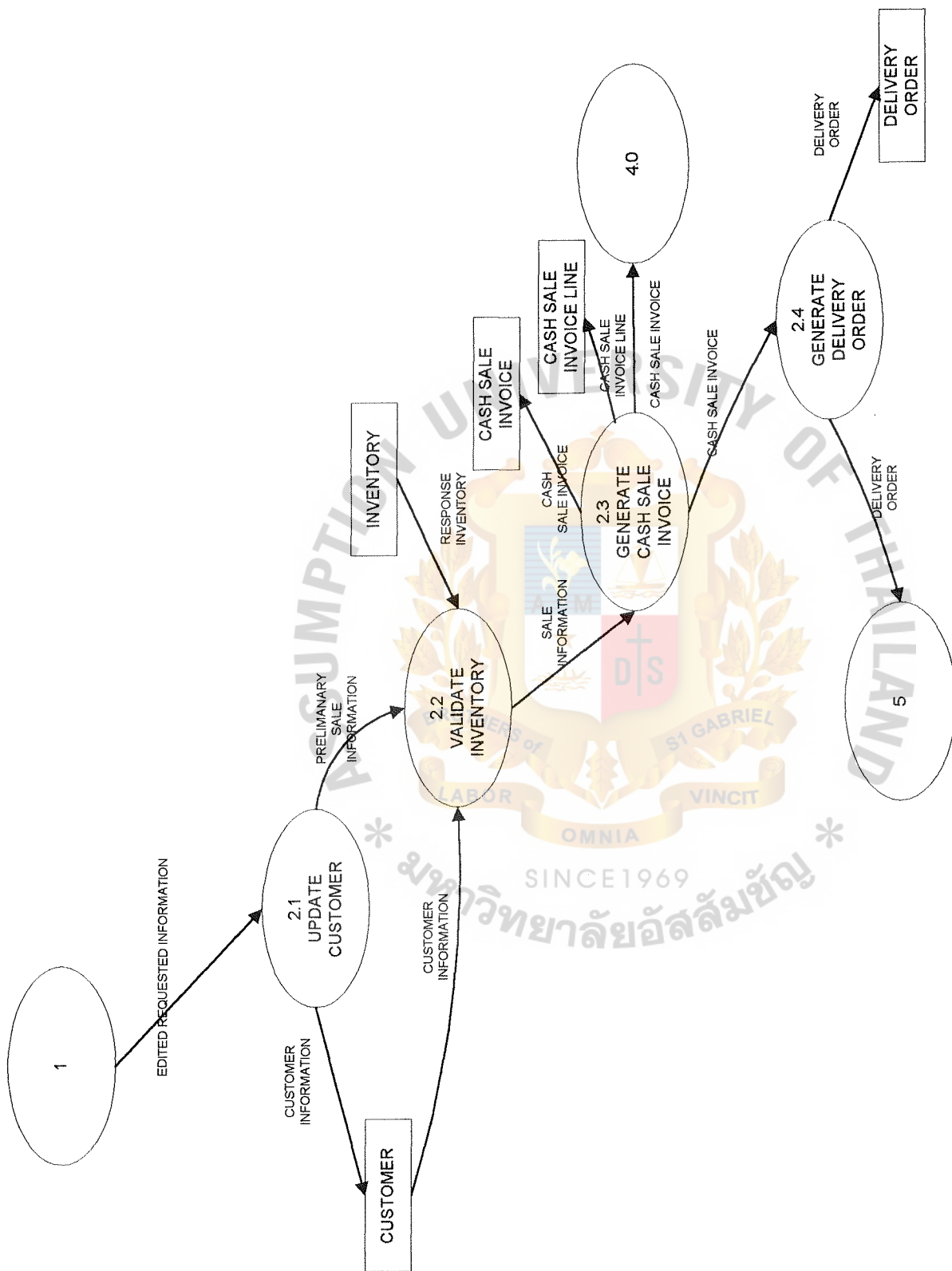


FIGURE A-2: DATA FLOW DIAGRAM LEVEL1: GENERATE CASH SALE

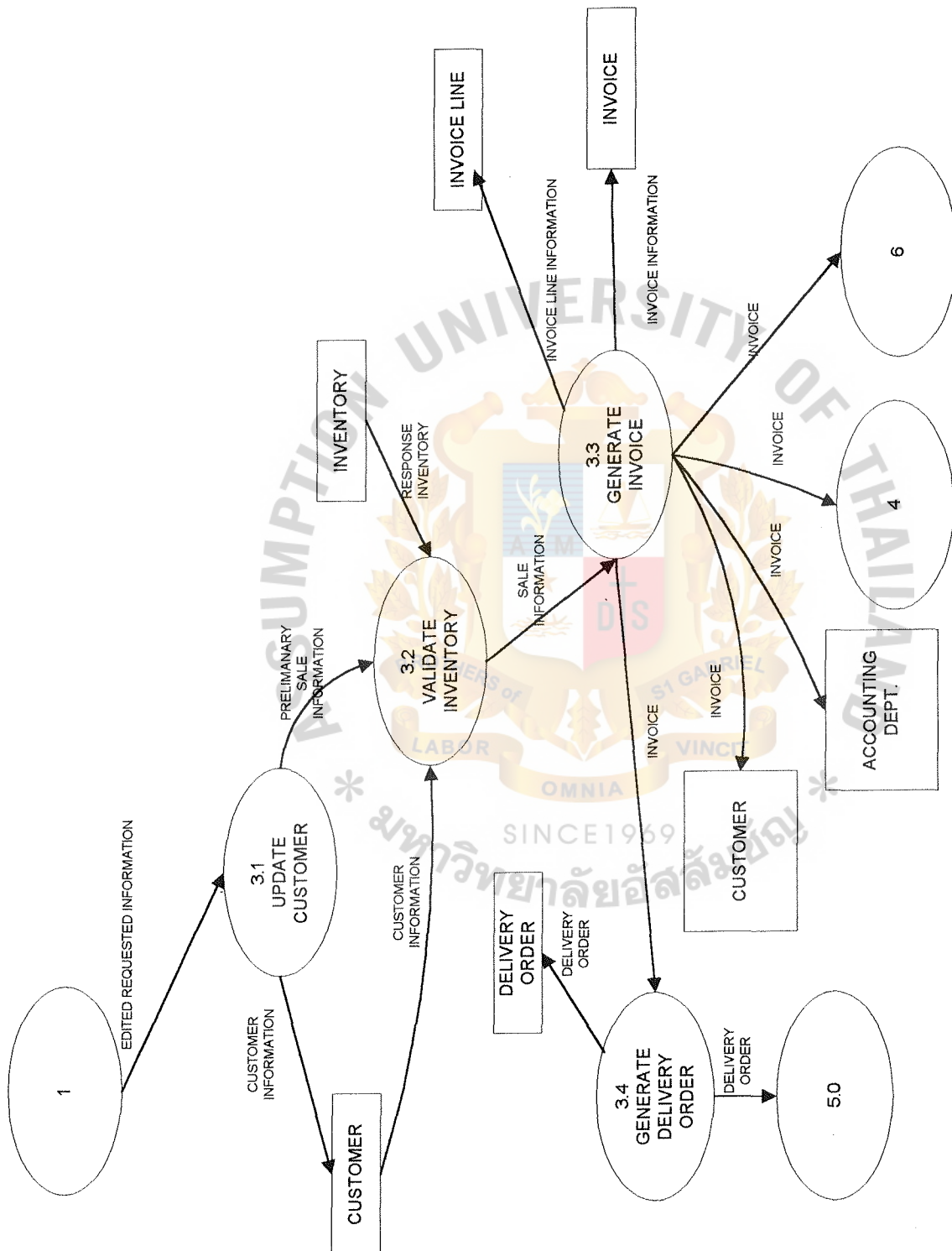


FIGURE A-3: DATA FLOW DIAGRAM LEVEL 1: GENERATE CREDIT SALE

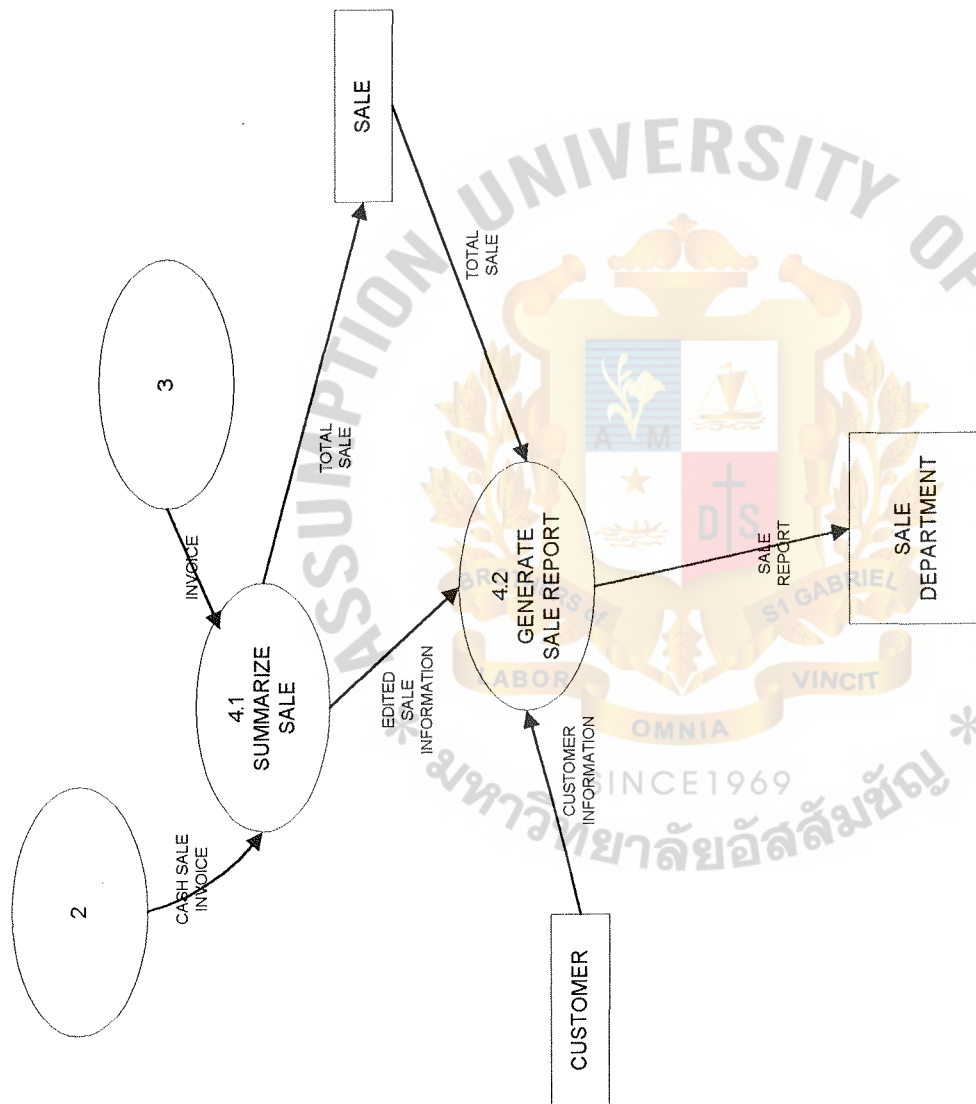


FIGURE A-4: DATA FLOW DIAGRAM LEVEL 1: UPDATE SALE

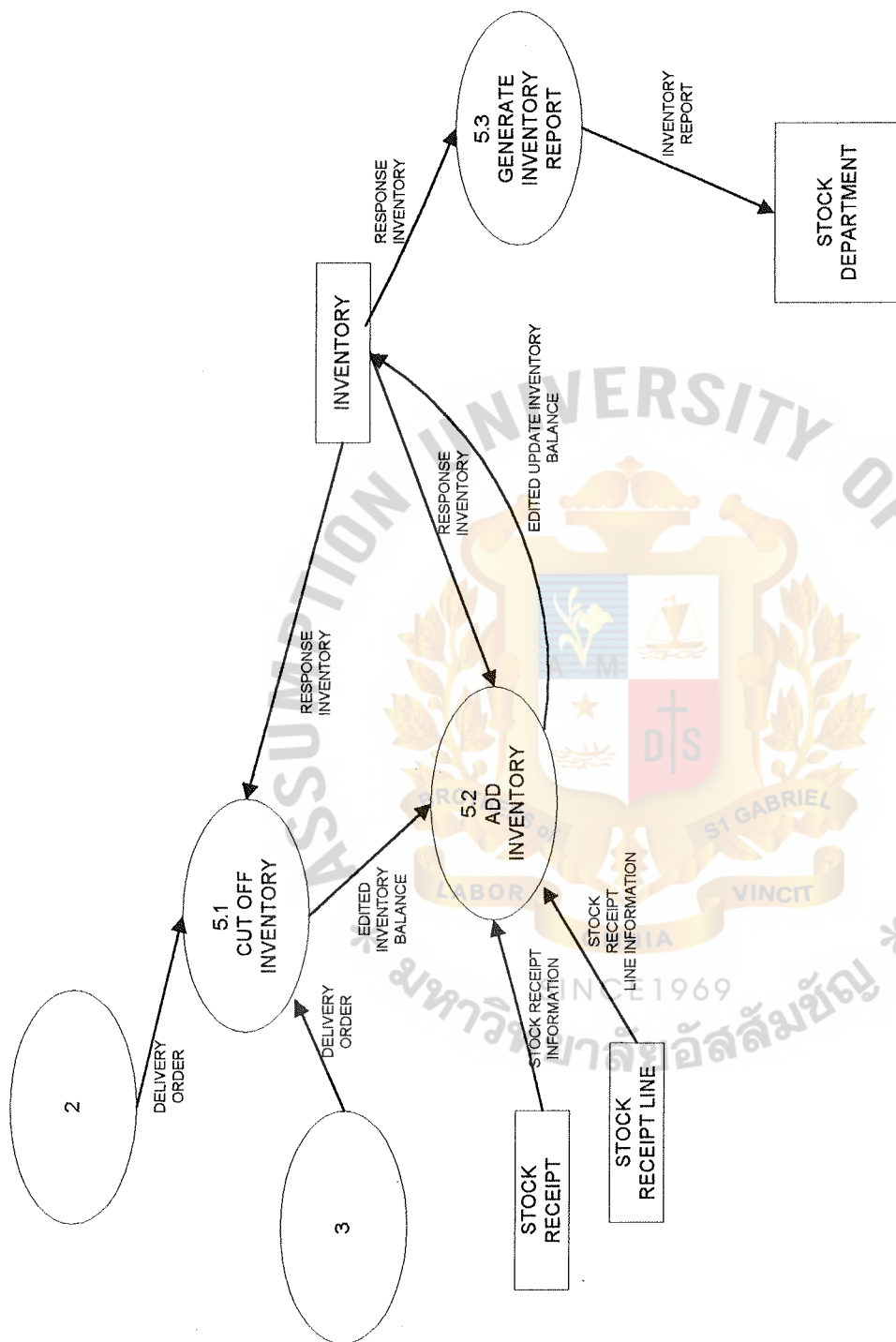


FIGURE A-5: DATA FLOW DIAGRAM LEVEL 1: UPDATE INVENTORY

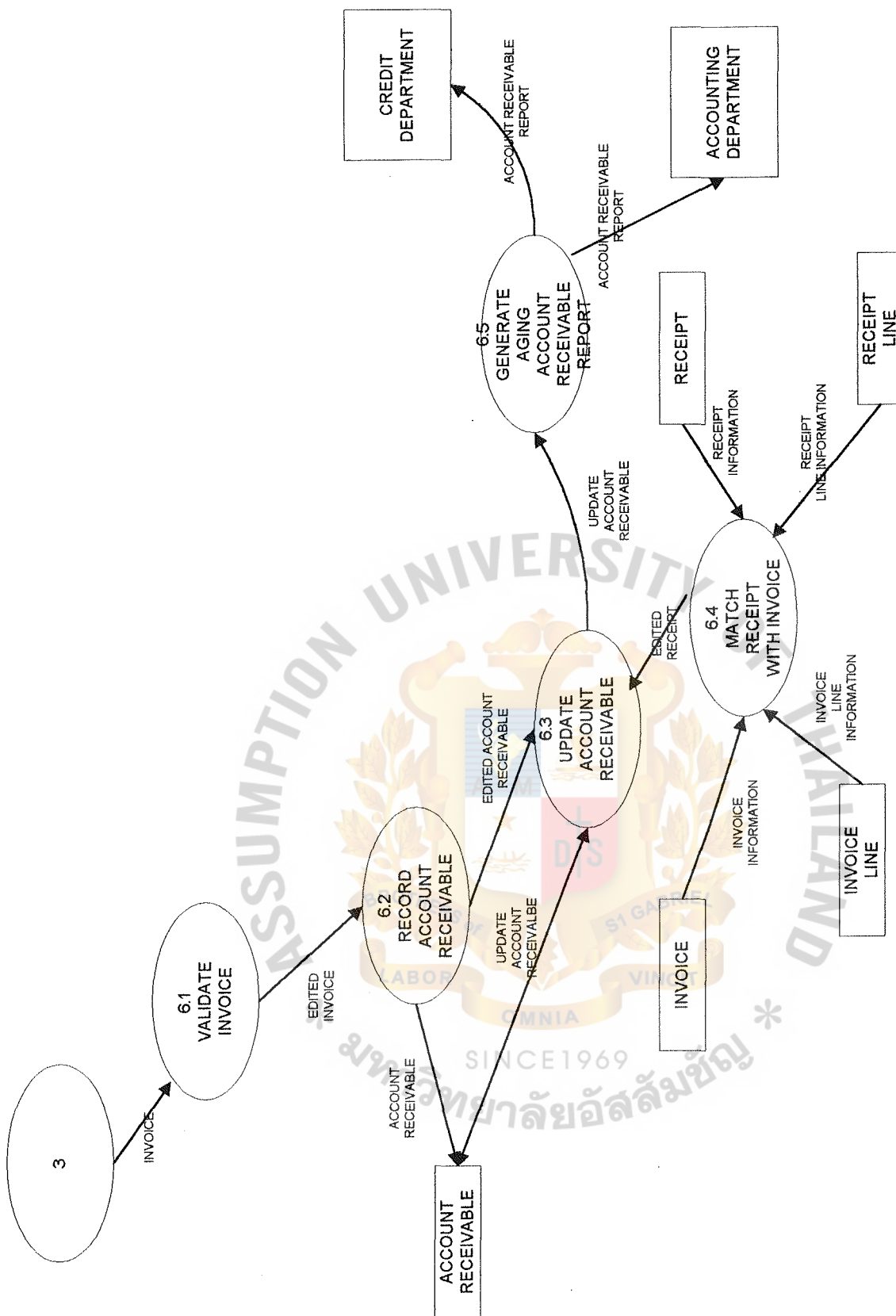


FIGURE A-6: DATA FLOW DIAGRAM LEVEL 1: UPDATE ACCOUNT RECEIVABLE

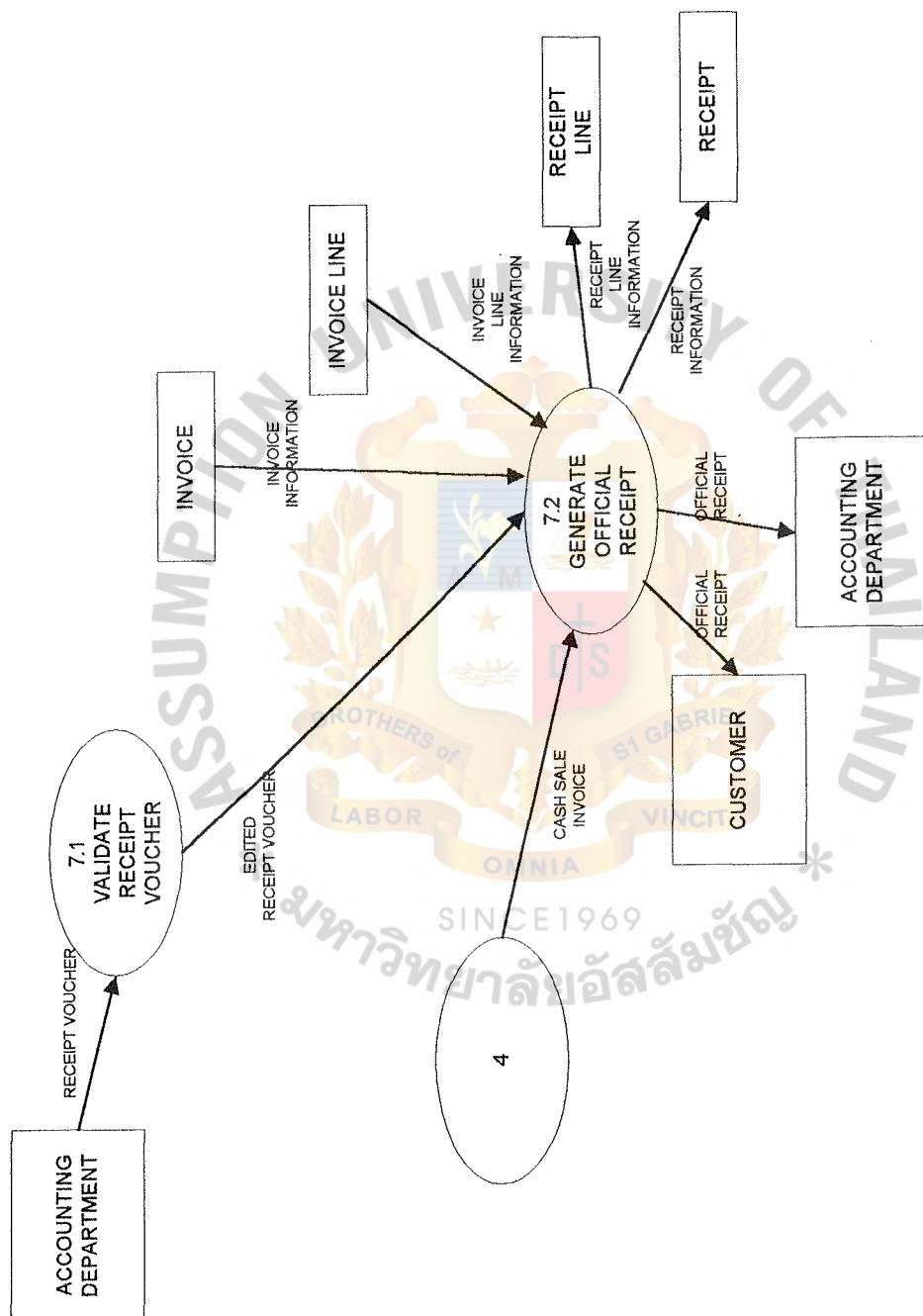
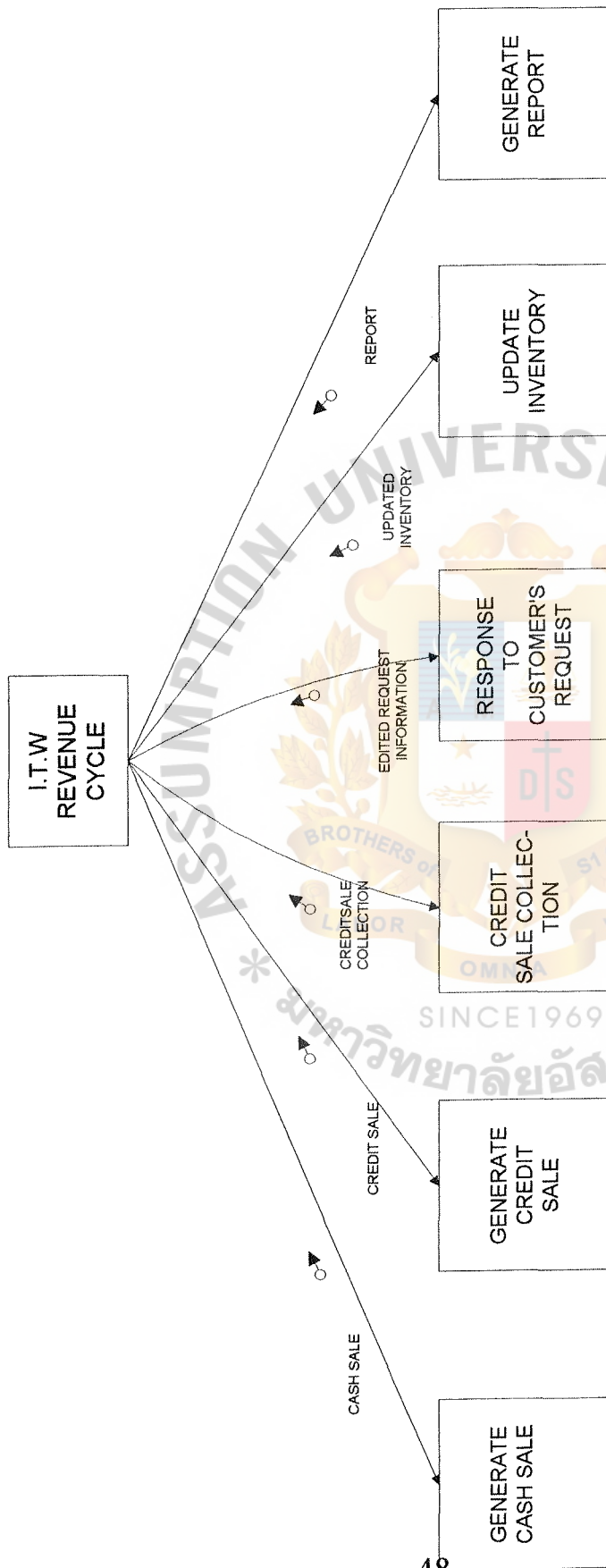


FIGURE A-7: DATA FLOW DIAGRAM LEVEL 1: COLLECT MONEY



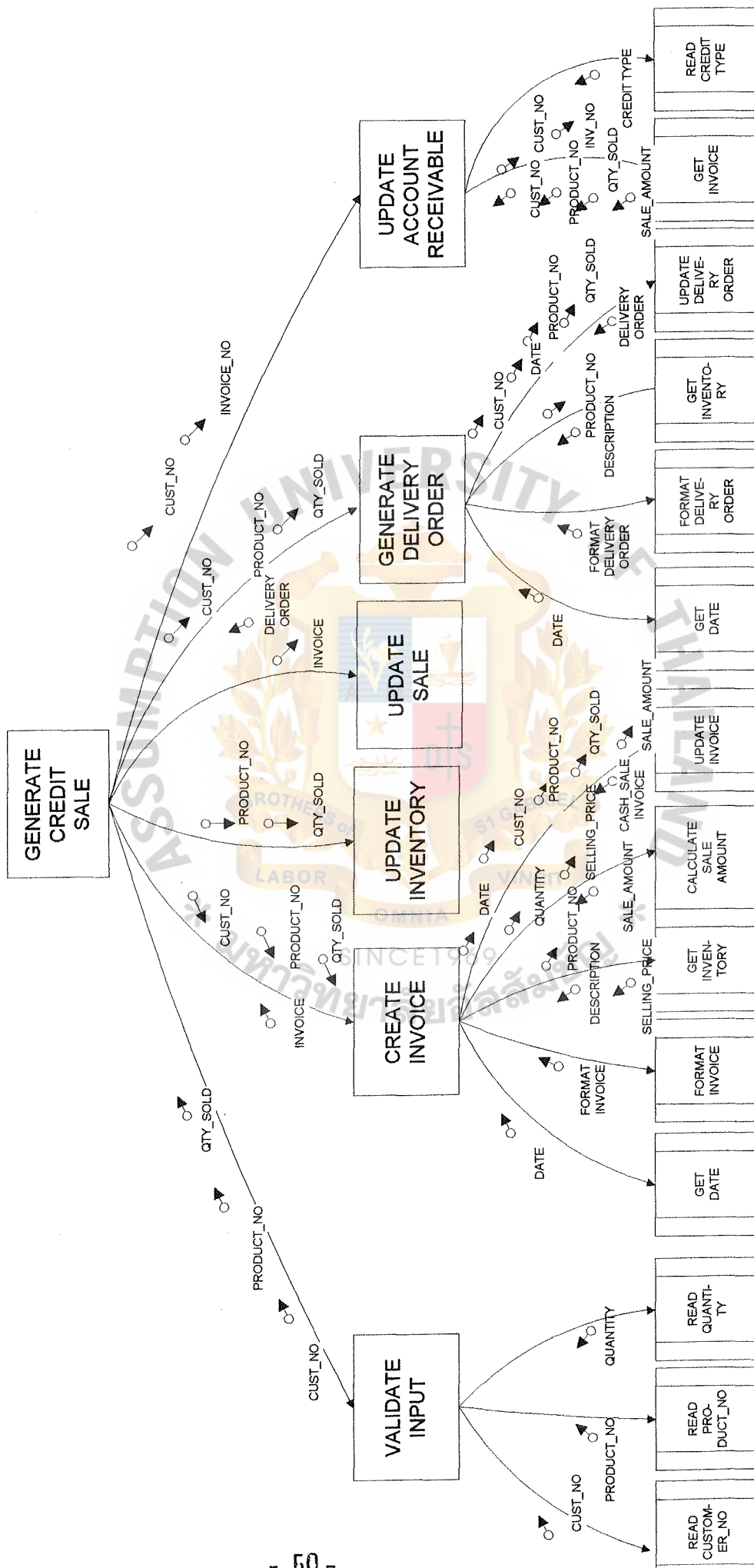


FIGURE B-2: GENERATE CREDIT SALE

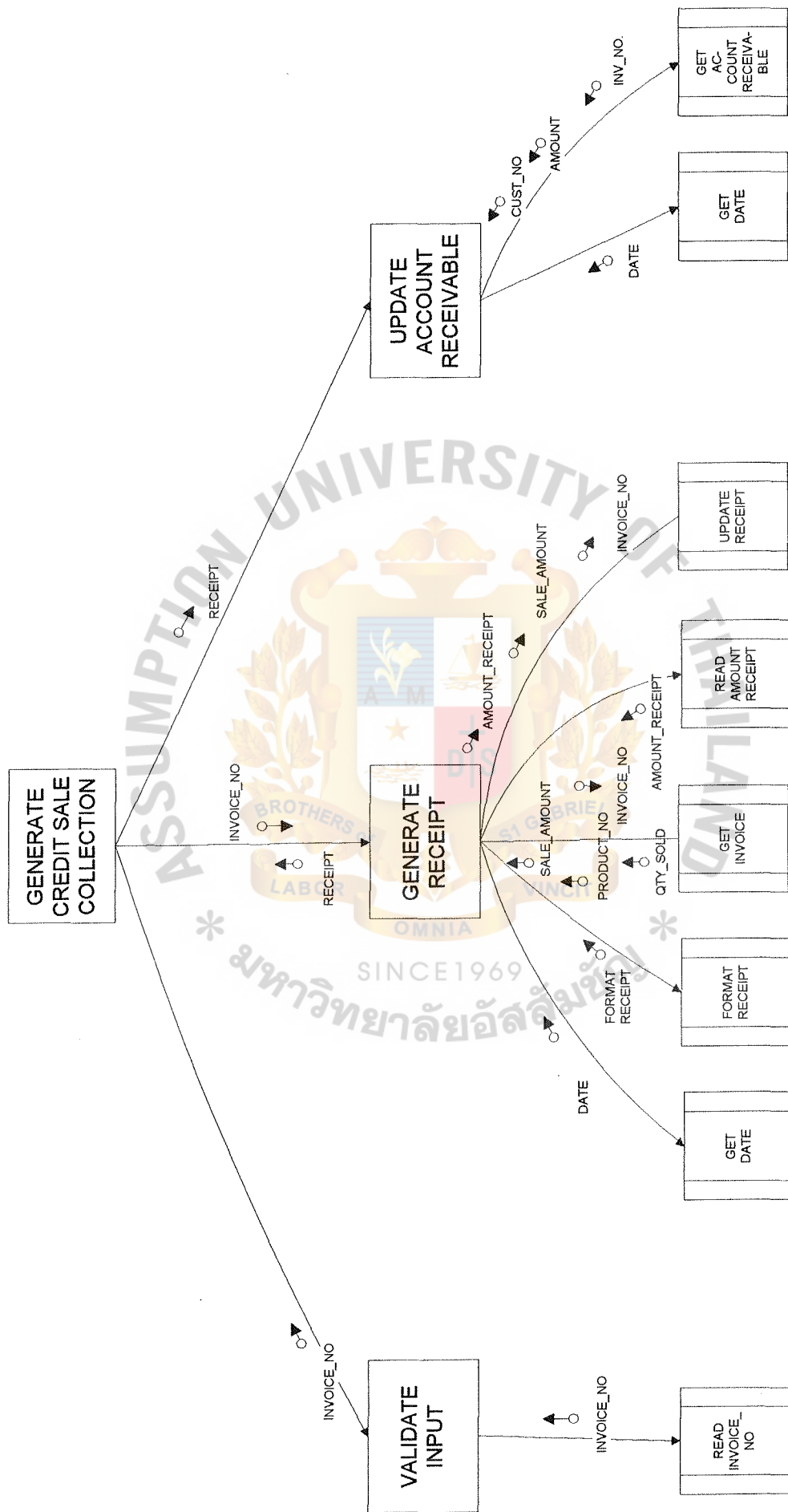


FIGURE B-3: STRUCTURE CHART: CREDIT SALE COLLECTION PROCESS

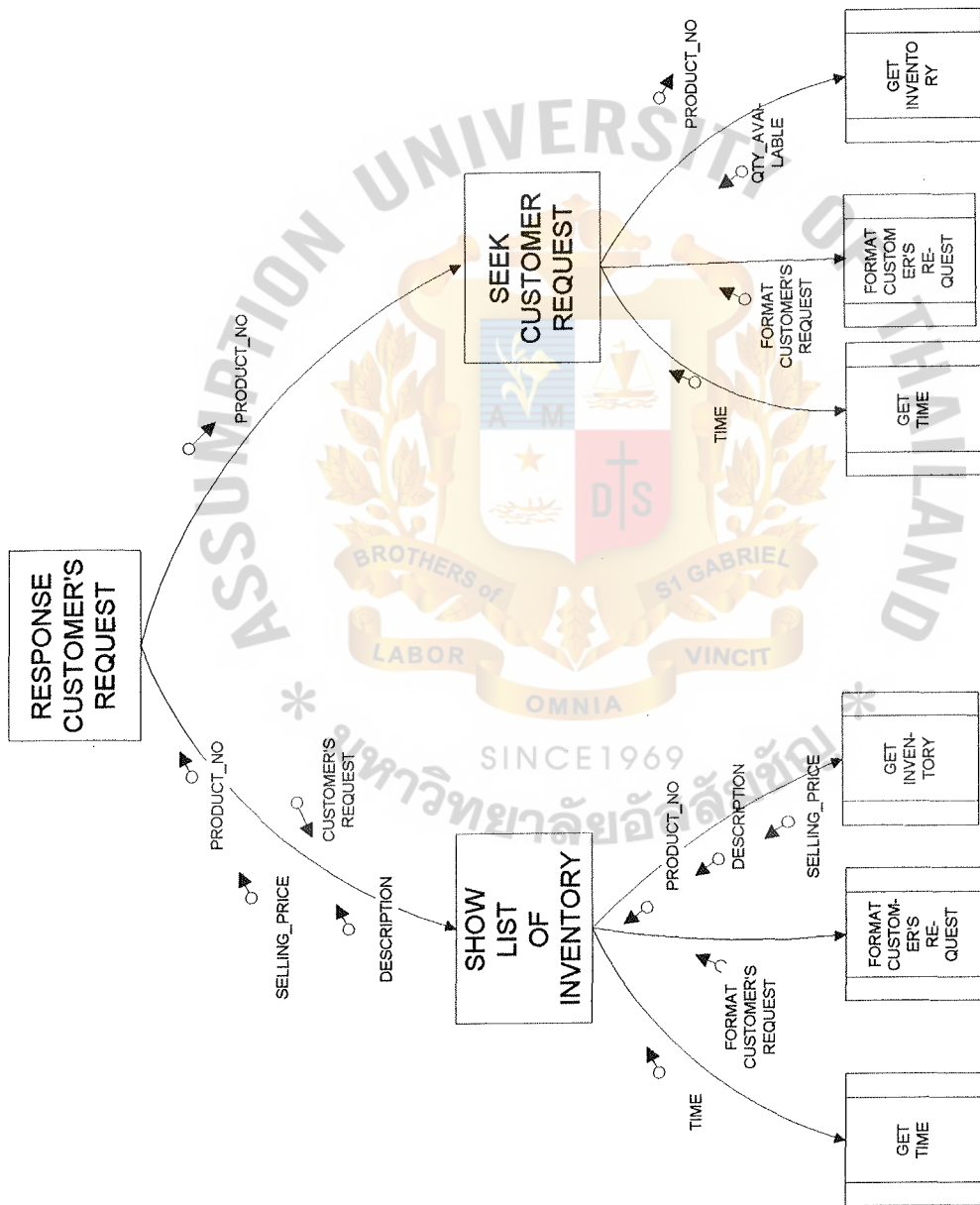


FIGURE B-4: STRUCTURE CHART : RESPONSE TO CUSTOMER'S REQUEST

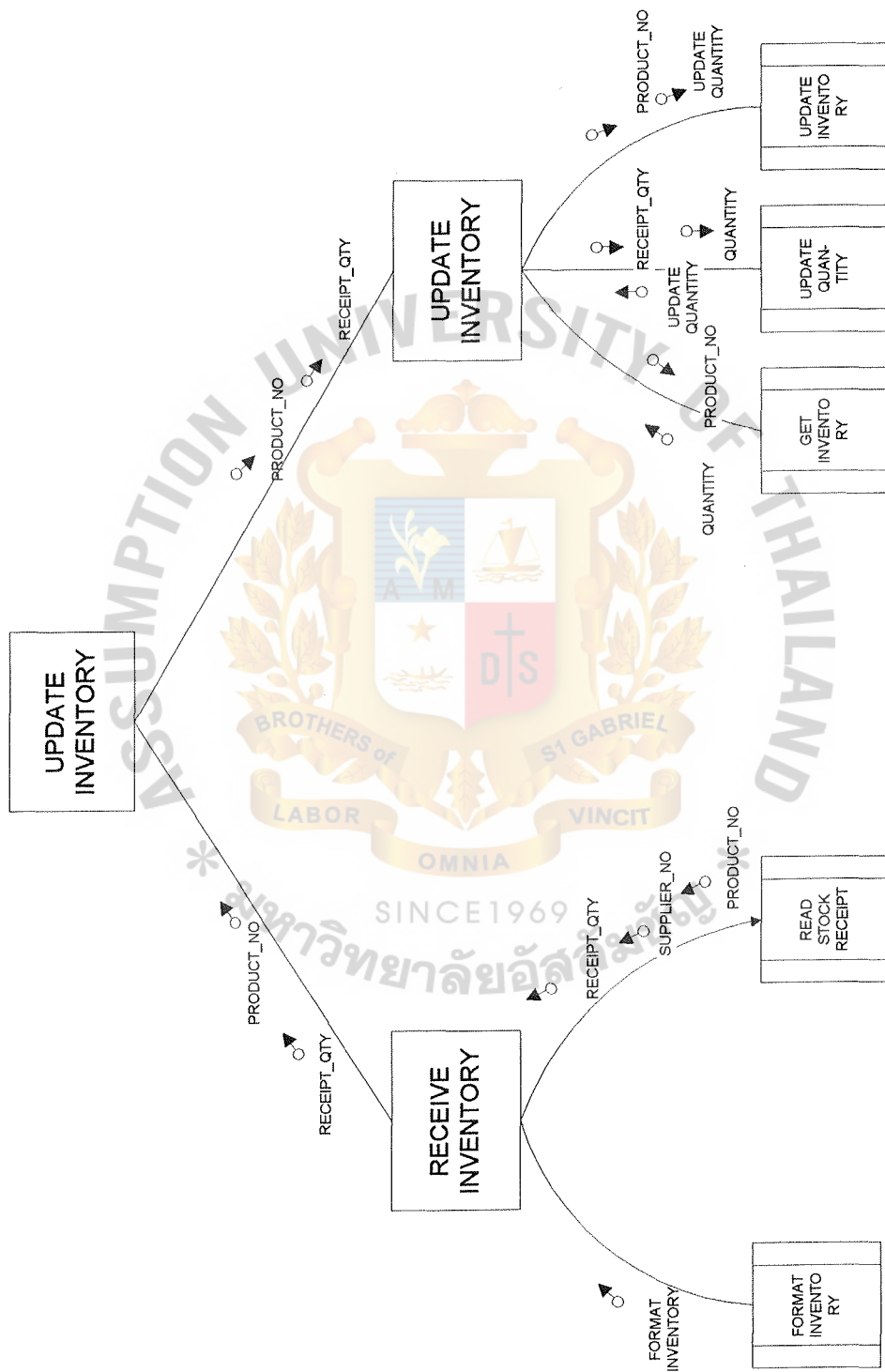


FIGURE B-5: STRUCTURE CHART: UPDATE INVENTORY PROCESS

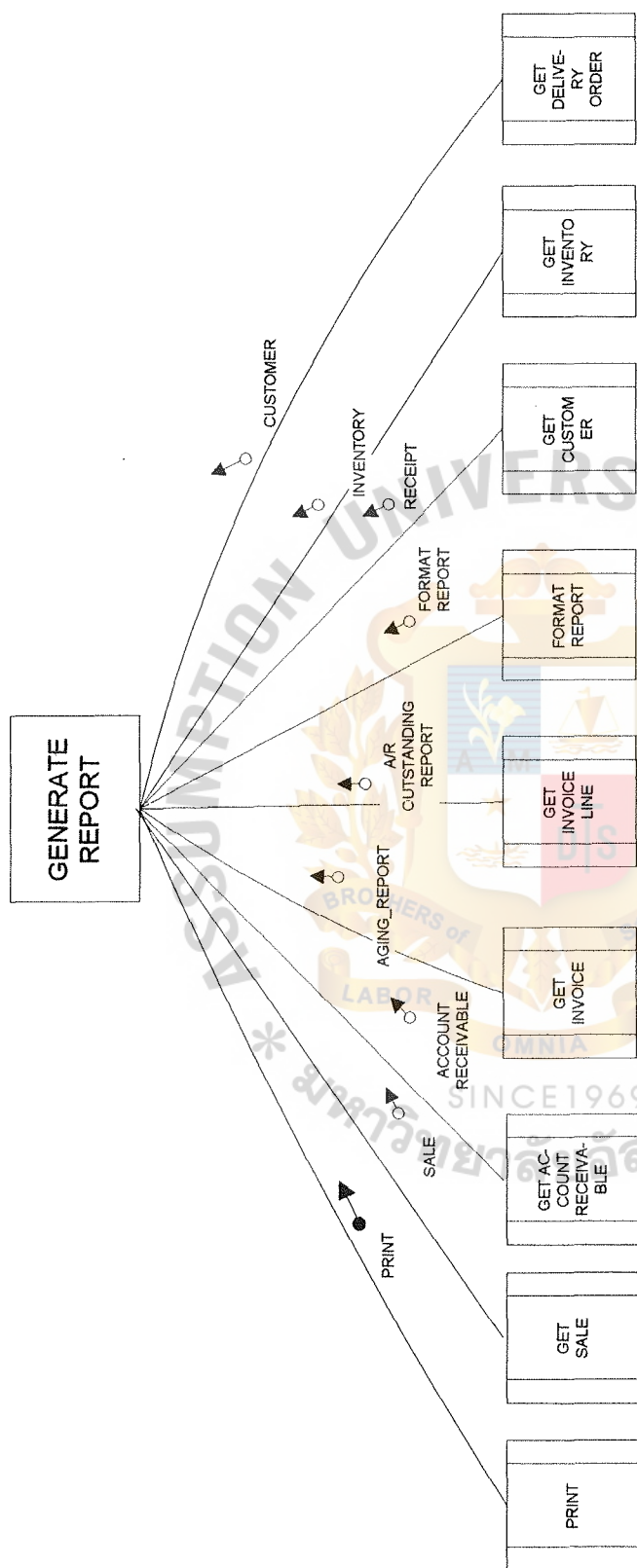


FIGURE B-6: STRUCTURE CHART: GENERATE REPORT

DATA DICTIONARY

ACCOUNT

RECEIVABLE = @CUST_NO + INV_NO + DAYS_OF_CRE-
DIT

CASH INVOICE = @CASH_INV_NO + CUST_NO +
CASH_INV_DATE

CASH INVOICE

LINE = @CASH_INV_NO + PRODUCT_NO +
QTY_SOLD

CUSTOMER = @CUST_NO + CUST_NAME +
CUST_ADDRESS + CUST_CITY +
CUST_POSTAL + CUST_TEL

DELIVERY

ORDER = @D/O_NO, INV_NO, CUST_NO, D/O_DATE,
DESTINATION

INVOICE = @INV_NO + CUST_NO + INV_DATE

INVOICE LINE = @INV_NO + PRODUCT_NO + QTY_SOLD

INVENTORY = @PRODUCT_NO + DESCRIPTION + SELLING_
PRICE + UNIT_PRICE + QUANTITY

SALE = @SALE_NO + INV_NO + CUST_NO

STOCK RECEIPT = @SR_NO + SR_DATE + SUPPLIER_NO

STOCK RECEIPT LINE = @SR_NO + PRODUCT_NO +
RECEIPT_QTY

SUPPLIER = SUPPLIER_NO+ SUPPLIER_NAME+ SUPPLIER
ADDRESS

RECEIPT = @RV_NO + CUST_NO + RV_DATE + TYPE

RECEIPT LINE = @RV_NO + INV_NO + RECEIPT_AMOUNT

ACCOUNT

RECEIVABLE = THE KEEPING RECORD OF THE CREDIT
SALE THAT IS AVAILABLE FOR
COLLECTION OF MONEY

CASH SALE INVOICE = THE DOCUMENT THAT RECORD THE
CASH SALE TRANSACTION AND KEPT
AS AN EVIDENCE IN SALE
TRANSACTION

CASH_INV_NO = *CASH SALE INVOICE NUMBER*
{NUMERIC_DIGIT}

CASH_INV_DATE= *DATE OF CASH SALE INVOICE*
{NUMERIC_DIGIT}

CUSTOMER = RECORD OF EXISTING CUSTOMER
INFORMATION

CUST_NO = *THE ASSIGNED NUMBER TO EACH
PARTICULAR CUSTOMER FOR
IDENTIFICATION*
{ NUMERIC_DIGIT}

CUST_NAME = *CUSTOMER NAME*
{LEGAL_CHARACTER}

CUST_ADDRESS = *PRESENT CUSTOMER ADDRESS*
{LEGAL_CHARACTER}

CUST_CITY = *PRESENT CUSTOMER CITY*
{LEGAL_CHARACTER}

CUST_POSTAL = *PRESENT CUSTOMER POSTAL CODE*
{NUMERIC_DIGIT}

CUST_TEL = *CUSTOMER TELEPHONE*
{NUMERIC_DIGIT}

DAYS_OF_CREDIT=* THE NUMBER OF DAYS THAT
GRANDS FOR THE CREDIT*
{ NUMERIC_DIGIT}

DELIVERY

ORDER = A DOCUMENTS DESCRIBED THE PRODUCT
NAME AND QUANTITY SOLD THAT PARTI-
CULARLY SENT TO STOCK DEPARTMENT
FOR DELIVERING THE PRODUCT

D/O_NO = *DELIVERY ORDER NUMBER* (COUNTER)
{NUMERIC_DIGIT}

D/O DATE = *DATE OF DELIVERY ORDER TO SHIP THE
PRODUCT*

DESTINATION = PLACE OF DELIVERY
{LEGAL_CHARACTER}

INQUIRY = THE REQUEST FROM CUSTOMER ABOUT
INVENTORY (DESCRIPTION, SELLING PRICE
UNIT ON HAND)

INVENTORY

REPORT = REPORT THAT SHOW CURRENTLY AT
PARTICULAR TIME FOR PRODUCT
NUMBER, DESCRIPTION, QUANTITY, UNIT
COST AND TOTAL AMOUNT

INV_NO = *INVOICE NUMBER SPECIFY TO THE EVENT
IDENTIFICATION OF CREDIT SALE*
{NUMERIC_DIGIT}

INV_DATE = *DATE OF MAKING CREDIT SALE TRANSAC-
TION*
{NUMERIC_DIGIT}

OFFICIAL RECEIPT = THE LEGAL DOCUMENTS INDICATED THE
COLLECTION OF MONEY OF WHICH IS
ISSUED TO CUSTOMER

PRODUCT_NO = *THE NUMBER IDENTIFICATION OF EACH
PRODUCT IDENTIFICATION*
{NUMERIC_DIGIT}

QTY_SOLD = *AMOUNT OF QUANTITY THAT WAS SOLD*
{NUMERIC_DIGIT}

QUANTITY = THE NUMBER OF QUANTITY ON HAND
IN INVENTORY
{NUMERIC_DIGIT}

RECEIPT = THE RECORD FOR THE RECEIVING
OF COLLECTION FROM BOTH CREDIT
SALE AND CASH SALE

RECEIPT_AMOUNT = *AMOUNT OF RECEIPT COLLECTION*
{NUMERIC_DIGIT}

RECEIPT_QTY = *THE NUMBER OF QUANTITY RECEIPT*
{NUMERIC_DIGIT}

RECEIVABLE

OUTSTANDING REPORT = REPORT THAT SHOW THE LIST OF
UNCOLLECTED CREDIT
SALEAMOUNT ARRANGED
ACCORDING TO EACH CUSTOMER
AND INVOICE NUMBER

REPLY = THE RESPONSE TO CUSTOMER INQUIRY

RESPONSE INVENTORY = {PRODUCT_NO, DESCRIPTION,
SELLING_PRICE, QUANTITY}

RV_DATE = *DATE OF RECIVING COLLECTION*
{NUMERIC_DIGIT}

RV_NO = *RECIPT VOUCHER NUMBER*
{NUMERIC_DIGIT}

SALE_NO = *SALE NUMBER (COUNTER NUMBER)*
{NUMERIC_DIGIT}

SALE REPORT = REPORT THAT SHOWS THE LIST OF SALE
TRANSACTION BOTH IN CREDIT AND CASH
SALE AND TOTAL SALE AT A PARTICULAR
TIME

SR_DATE = *STOCK RECEIPT DATE*
{NUMERIC_DIGIT}

SR_NO = *STOCK RECEIPT NO RUN IN COUNTER*
{NUMERIC_DIGIT}

STOCK RECEIPT = THE INFORMATION CONTAINING THE
RECEIPT PRODUCT NUMBER, RECEIPT
AMOUNT AND SUPPLIER NUMBER

SUPPLIER_AD-

DRESS = *ADDRESS OF SUPPLIER*
{LEGAL_CHARACTER}

SUPPLIER_NAME= *NAME OF SUPPLIER WHO PROVIDE THE
INVENTORY*
{LEGAL_CHARACTER}

SUPPLIER_NO = *SUPPLIER IDENTIFICATION NUMBER*
{NUMERIC_DIGIT}

TYPE = TYPE OF SALE
{LEGAL_CHARACTER}

UNIT_PRICE = COST PRICE PER UNIT OF INVENTORY

PROCESS SPECIFICATION

PROCESS 1.1 *SHOW INVENTORY LIST*

PRE-CONDITION: CUSTOMER'S INQUIRY

POST-CONDITION: EDITED INVENTORY:

BEGIN

DO WHILE NOT END OF FILE IN INVENTORY

WRITE PRODUCT NUMBER, PRODUCT DESCRIPTION

END DO

REPEAT UNTILL NO EDITED PRODUCT NUMBER

READ EDITED PRODUCT NUMBER

END REPEAT

END

PROCESS 1.2 *CHECK THE AVAILABILITY OF INVENTORY*

PRE-CONDITION: EDITED INVENTORY

POST-CONDITION: DESIRED INVENTORY

BEGIN

REPEAT UNTIL NO MORE EDITED PRODUCT NUMBER

GET EDITED PRODUCT NUMBER

WRITE PRODUCT NUMBER, DESCRIPTION,

AVAILABLE BALANCE

END REPEAT

END

PROCESS 1.3 CONFIRM EDITED REQUESTED INFORMATION

PRE-CONDITION: DESIRED INVENTORY

POST-CONDITION: EDITED REQUEST INFORMATION

BEGIN

REPEAT UNTIL NO MORE EDITED PRODUCT NUMBER

WRITE PRODUCT NUMBER, DESCRIPTION,

AVAILABLE BALANCE

WRITE "CONFIRM EDITED PRODUCT NUMBER"

IF CONFIRM = NO THEN DELETE EDITED PRODUCT
NUMBER

END REPEAT

END

PROCESS 2.1 UPDATE CUSTOMER

PRE-CONDITION: EDITED REQUEST INFORMATION

POST-CONDITION: PRELIMINARY SALE INFORMATION

BEGIN

READ CUSTOMER NAME

IF CUSTOMER NAME= CUST_NAME THEN WRITE

CUST_NO, CUST_NAME, CUST_ADDRESS, CUST_TEL

ELSE WRITE" NO RECORD EDIT NEW CUSTOMER"

END IF

READ CUSTOMER NAME, ADDRESS, CITY, POSTAL, CITY

TELEPHONE

RECORD CUSTOMER INFORMATION IN CUSTOMER FILE

END

PROCESS 2.2 *VALIDATE INVENTORY*

PRE-CONDITION: PRELIMINARY SALE INFORMATION

POST-CONDITION: SALE INFORMATION

BEGIN

REPEAT UNTILL NO MORE EDITED PRODUCT NUMBER

GET EDITED PRODUCT NUMBER

END REPEAT

END

PROCESS 2.3 *GENERATE CASH SALE INVOICE*

PRE-CONDITION: SALE INFORMATION

POST-CONDITION: CASH SALE INVOICE

BEGIN

GET CASH INVOICE NUMBER

WRITE CUSTOMER NUMBER, NAME, ADDRESS, CITY, TELEPHONE

REPEAT UNTILL NO MORE EDITED PRODUCT NUMBER

WRITE EDITED PRODUCT NUMBER, DESCRIPTION,
QUANTITY, SELLING PRICE

$\text{SALE AMOUNT} = \text{SELLING PRICE} * \text{QUANTITY}$

$\text{TOTAL SALE} = \text{TOTAL SALE} + \text{SALE AMOUNT}$

END REPEAT

WRITE TOTAL SALE

$\text{VAT} = \text{TOTAL SALE} * .07$

$\text{TOTAL AMOUNT} = \text{TOTAL SALE} + \text{VAT}$

WRITE TOTAL AMOUNT

RECORD CASH NVOICE NUMBER, CUSTOMER
NUMBER,CASH SALE INVOICE DATE IN CASH SALE
INVOICE
RECORD CASH SALE INVOICE NUMBER, PRODUCT NUMBER,
QUANTITY SOLD IN CASH SALE INVOICE LINE

END

PROCESS 2.4 *GENERATE DELIVERY ORDER*

PRE-CONDITION: CASH SALE INVOICE

POST-CONDITION: DELIVERY ORDER

BEGIN

GET DELIVERY ORDER NUMBER
READ DESTINATION PLACE
WRITE CUSTOMER NUMBER, NAME, ADDRESS, CITY, TELE-
PHONE, DESTINATION PLACE
REPEAT UNTILL NO MORE EDITED PRODUCT NUMBER
 WRITE EDITED PRODUCT NUMBER, DESCRIPTION,
 QUANTITY
END REPEAT
RECORD DELIVER ORDER NUBER, INVOICE NUBER,
CUSTOMER NUMBER, DELIVERY ORDER DATE,
DESTINATION IN DELIVERY ORDER

END

PROCESS 3.1 *UPDATE CUSTOMER*

PRE-CONDITION: EDITED REQUEST INFORMATION

POST-CONDITION: PRELIMANARY SALE INFORMATION

BEGIN

READ CUSTOMER NAME

IF CUSTOMER NAME= CUST_NAME THEN WRITE

CUST_NO, CUST_NAME, CUST_ADDRESS, CUST_TEL

ELSE WRITE" NO RECORD EDIT NEW CUSTOMER"

END IF

READ CUSTOMER NAME, ADDRESS, CITY, POSTAL, CITY

TELEPHONE

RECORD CUSTOMER INFORMATION IN CUSTOMER FILE

END

PROCESS 3.2 *VALIDATE INVENTORY*

PRE-CONDITION: PRELIMINARY SALE INFORMATION

POST-CONDITION: SALE INFORMATION

BEGIN

REPEAT UNTILL NO MORE EDITED PRODUCT NUMBER

GET EDITED PRODUCT NUMBER

END REPEAT

END

PROCESS 3.3 *GENERATE SALE INVOICE*

PRE-CONDITION: SALE INFORMATION

POST-CONDITION: INVOICE

BEGIN

GET INVOICE NUMBER

WRITE CUSTOMER NUMBER, NAME, ADDRESS, CITY, TELE-
PHONE

REPEAT UNTILL NO MORE EDITED PRODUCT NUMBER

WRITE EDITED PRODUCT NUMBER, DESCRIPTION,
QUANTITY, SELLING PRICE

$\text{SALE AMOUNT} = \text{SELLING PRICE} * \text{QUANTITY}$

$\text{TOTAL SALE} = \text{TOTAL SALE} + \text{SALE AMOUNT}$

END REPEAT

WRITE TOTAL SALE

$\text{VAT} = \text{TOTAL SALE} * .07$

$\text{TOTAL AMOUNT} = \text{TOTAL SALE} + \text{VAT}$

WRITE TOTAL AMOUNT

RECORD INVOICE NUMBER, CUSTOMER NUMBER, INVOICE
DATE IN INVOICE

RECORD INVOICE NUMBER, PRODUCT NUMBER,
QUANTITY SOLD IN INVENTORY LINE

END

PROCESS 3.4 *GENERATE DELIVERY ORDER*

PRE-CONDITION: INVOICE

POST-CONDITION: DELIVERY ORDER

BEGIN

GET DELIVERY ORDER NUMBER

READ DESTINATION PLACE

WRITE CUSTOMER NUMBER, NAME, ADDRESS, CITY, TELE-
PHONE, DESTINATION PLACE

REPEAT UNTILL NO MORE EDITED PRODUCT NUMBER

WRITE EDITED PRODUCT NUMBER, DESCRIPTION,
QUANTITY

END REPEAT

RECORD DELIVERY ORDER NUMBER, INVOICE NUMBER,
CUSTOMER NUMBER, DELIVERY ORDER DATE,
DESTINATION IN DELIVERY ORDER

END

PROCESS 4.1 UPDATE SALE

PRE-CONDITION: CASH SALE INVOICE, INVOICE

POST-CONDITION: EDITED SALE INFORMATION

BEGIN

GET SALE NUMBER (COUNTER)

DO WHILE THERE IS NO END OF FILE IN CASH SALE
INVOICE

WRITE CASH INVOICE NUMBER, DATE, CUSTOMER
NAME, TOTAL SALE

END DO

DO WHILE THERE IS NO END OF FILE IN SALE INVOICE

WRITE INVOICE NUMBER, DATE, CUSTOMER NAME,
TOTAL SALE
END DO
UPDATE TOTAL SALE = SUM@TOTAL SALE
END

PROCESS 4.2 GENERATE SALE REPORT

PRE-CONDITION: EDITED SALE INFORMATION

POST-CONDITION: SALE REPORT

BEGIN
DO WHILE THERE IS NO END OF FILE IN CASH SALE
 INVOICE
 RECEIVE PRINT COMMAND
 PRINT SALE NO, CASH SALE INVOICE NUMBER, DATE
 CUSTOMER NAME , TOTAL SALE
END DO
DO WHILE THERE IS NO END OF FILE IN INVOICE
 RECEIVE PRINT COMMAND
 PRINT SALE NO, INVOICE NUMBER , DATE
 CUSTOMER NAME, TOTAL SALE
END DO
PRINT UPDATE TOTAL SALE
END

PROCESS 5.1 CUT-OFF INVENTORY

PRE-CONDITION: DELIVERY ORDER

POST-CONDITION: EDITED INVENTORY BALANCE

BEGIN

DO WHILE THERE IS NO MORE END OF FILE IN DELIVERY

ORDER

GET PRODUCT NUMBER, QUANTITY SOLD

QUANTITY = QUANTITY - QUANTITY_SOLD

END DO

END

PROCESS 5.2 ADD INVENTORY

PRE-CONDITION: STOCK RECEIPT, EDITED INVENTORY BALANCE

POST-CONDITION: EDITED UPDATE INVENTORY BALANCE

BEGIN

DO WHILE THERE IS NO MORE END OF FILE IN STOCK

RECEIPT

GET PRODUCT NUMBER, QUANTITY RECEIPT

QUANTITY = QUANTITY + QUANTITY_RECEIPT

END DO

UPDATE QUANTITY

END

PROCESS 5.3 GENERATE THE INVENTORY REPORT

PRE-CONDITION: RESPONSE INVENTORY

POST-CONDITION: INVENTORY REPORT

BEGIN

DO WHILE THERE IS NO MORE END OF FILE IN INVENTORY

INVENTORY COST = QUANTITY * UNIT COST

PRINT PRODUCT NO, DESCRIPTION, QUANTITY,

UNIT COST, INVENTORY COST

END DO

TOTAL INVENTORY COST= @SUM INVENTORY COST

PRINT TOTAL INVENTORY COST

END

PROCESS 6.1 VALIDATE INVOICE

PRE-CONDITION: INVOICE

POST-CONDITION: EDITED INVOICE

BEGIN

DO WHILE THERE IS NO END OF FILE IN INVOICE

GET INVOICE NUMBER, INVOICE DATE, CUSTOMER NAME,

TOTAL AMOUNT

END DO

END

PROCESS 6.2 RECORD ACCOUNT RECEIVABLE

PRE-CONDITION: EDITED INVOICE

POST-CONDITION: EDITED ACCOUNT RECEIVABLE

BEGIN

DO WHILE THERE IS NO END OF FILE IN INVOICE

 SORT CUST_NO, INV-NO, TOTAL AMOUNT IN
 ACCOUNT RECEIVABLE

 DAY_OF_CREDIT = DAYS (TODAY DATE - INVOICE DATE)

 CASE

 DAY_OF_CREDIT > 30 DAYS BUT LESS THAN OR

 EQUAL 60 DAYS RECORD IN "30 DAYS - 60 DAYS

 DAY_OF_CREDIT > 60 DAYS RECORD IN "OVER 60
 DAYS"

 OTHERWISE RECORD AS "CURRENT"

 END CASE

 TOTAL OWNED BALANCE = TOTAL OWNED BALANCE +
 TOTAL AMOUNT

END DO

END

PROCESS 6.3 UPDATE ACCOUNT RECEIVABLE

PRE-CONDITION: EDITED ACCOUNT RECEIVABLE, EDITED RECEIPT

POST-CONDITON: UPDATE ACCOUNT RECEIVABLE

BEGIN

DO WHILE THERE IS NO END OF FILE IN ACCOUNT
RECEIVABLE

GET CUSTOMER NUMBER, INVOICE NUMBER, TOTAL
OWNED BALANCE IN ACCOUNT RECEIVABLE

SEEK CUSTOMER NUMBER IN RECEIPT

TOTAL OWNED BALANCE = TOTAL OWNED BALANCE -
AMOUNT RECEIPT

DELETE PAID INVOICE NUMBER IN ACCOUNT
RECEIVABLE

END DO

END

PROCESS 6.4 MATCH RECEIPT WITH INVOICE

PRE-CONDITION: RECEIPT

POST-CONDITION: EDITED RECEIPT

BEGIN

DO WHILE THERE IS NO END OF FILE IN RECEIPT

GET INVOICE NUMBER, AMOUNT RECEIPT IN RECEIPT

MATCH INVOICE NUMBER IN RECEIPT WITH INVOICE
NUMBER IN INVOICE

LIST INVOICE TO DELETE IN ACCOUNT RECEIVABLE

END DO

END

PROCESS 6.5 GENERATE AGING ACCOUNT RECEIVABLE REPORT

PRE-CONDITION: UPDATE ACCOUNT RECEIVABLE

POST-CONDITION: AGING ACCOUNT RECEIVABLE REPORT

BEGIN

DO WHILE THERE IS NO END OF FILE IN ACCOUNT
RECEIVABLE

PRINT CUSTOMER NUMBER , INVOICE NUMBER,
DAYS_OF_CREDIT, TOTAL AMOUNT OWNED,

CURRENT, 30 -60 DAYS, OVER 60 DAYS

TOTAL A/R BALANCE = TOTAL A/R BALANCE

+ TOTAL AMONT OWNED

END DO

END

PROCESS 7.1 VALIDATE RECEIPT VOUCHER

PRE-CONDITION: RECEIPT VOUCHER

POST-CONDITION: EDITED RECEIPT VOUCHER

BEGIN

READ CUSTOMER NUMBER, TYPE OF SALE

IF TYPE OF SALE = CREDIT ;

DO WHILE NO END OF FILE IN ACCOUNT RECEIVABLE

LIST INVOICE NUMBER, INVOICE DATE, AMOUNT, TOTAL

AMOUNT

END DO

ELSE

DO WHILE NO END OF FILE IN CASH SALE INVOICE

LIST CASH SALE INVOICE, CASH SALE INVOICE DATE,

AMOUNT

END DO

END IF

READ RECEIPT VOUCHER NUMBER, DATE, RECEIPT

AMOUNT, CASH SALE INVOICE NUMBER, INVOICE

NUMBER

RECORD IN RECEIPT FILE

END

PROCESS 7.2 GENERATE OFFICIAL RECEIPT

PRE-CONDITION: EDITED RECEIPT VOUCHER

POST-CONDITION: OFFICIAL RECEIPT

BEGIN

RUN OFFICIAL RECEIPT NUMBER

OPEN RECEIPT FILE

PRINT CUSTOMER NUMBER, CUSTOMER NAME, ADDRESS,
CITY, TELEPHONE

IF TYPE OF SALE = CREDIT SALE THEN

DO WHILE NOT END OF FILE IN INVOICE

GET INVOICE NUMBER

PRINT PRODUCT NO, DESCRIPTION, SELLING PRICE,
AMOUNT

END DO

PRINT TOTAL AMOUNT

PRINT AMOUNT RECEIPT

ELSE

DO WHILE NOT END OF FILE IN INVOICE IN CASH
SALE INVOICE

GET CASH SALE INVOICE NUMBER

PRINT PRODUCT NO, DESCRIPTION, SELLING
PRICE, AMOUNT

END DO

PRINT TOTAL AMOUNT

PRINT AMOUNT RECEIPT

END IF

END

MODULE SPECIFICATION

MODULE: GENERATE CASH SALE

PURPOSE: To validate input, create cash sale invoice, update inventory, update sale, generate delivery order as well as generate receipt

USES : -

RETURN : -

Call VALIDATE INPUT

Call CREATE CASH SALE INVOICE

Call UPDATE INVENTORY

Call GENERATE DELIVERY ORDER

Call GENERATE RECEIPT

MODULE: VALIDATE INPUT

PURPOSE: To obtain customer number , product number, and quantity desired

USE: -

RETURN : Customer number, product number, sold quantity

Begin

Read Cust_no, Product_no, qty_sold

End

MODULE: CREATE CASH SALE INVOICE

PURPOSE: To create cash sale invoice

USE: Cust_no, Product_no, Qty_sold

RETURN Cash Sale Invoice

Begin:

get DATE

get CASH SALE INVOICE FORMAT

get INVENTORY (product description, selling price)

compute SALE_ AMOUNT (qty_sold * selling price)

Update CASH SALE INVOICE

print CASHINV_NO, CASHINV_DATE, CUST_NAME,
CUST_ADDRESS, PRODUCT_NO, PRODUCT_DESCRIPTION,
QTY_SOLD, SELLING_PRICE, SALE_AMOUNT,
TOTAL_SALE_AMOUNT)

End

MODULE: UPDATE INVENTORY

PURPOSE: to update the quantity in INVENTORY

USES : Product_no, Qty_sold

RETURN: -

Begin

Get INVENTORY (Quantity)

Quantity = Quantity - Qty_sold

End

MODULE: UPDATE SALE

PURPOSE: To update amount of sale

USES: CASH SALE INVOICE

RETURN: -

Begin

Get Sale

store CASH SALE INVOICE

End

MODULE: GENERATE DELIVERY ORDER

PURPOSE: To generate Delivery Order

USES: Cust_no, Product_no, Qty_sold

RETURN: Delivery Order

Begin

get DATE

get FORMAT DELIVERY ORDER

get INVENTORY (product description)

update DELIVERY ORDER

print DO_NO, DO_DATE, CUST_NO, CUST_NAME, CUST_ADDRESS,

PRODUCT_NO, PRODUCT_DESCRIPTION, QTY_SOLD

END

MODULE: GENERATE RECEIPT

PURPOSE: to issue receipt to customer

USES: cashinv_no

RETURN: Official Receipt

Begin

get CASH SALE INVOICE (cust_no, product_no, qty_sold, sale amount)

Read AMOUT RECEIPT

print CUST_NO, CUST_NAME, CUST_ADDRESS, PRODUCT_NO,
PRODUCT_DESCRIPTION, QTY_SOLD, SALE AMOUNT

End

MODULE: GENERATE CREDIT SALE

PURPOSE: To validate input, create invoice, update inventory, update sale,
generate delivery order as well as update Account Receivable

USES : -

RETURN : -

Call VALIDATE INPUT

Call CREATE INVOICE

Call UPDATE INVENTORY

Call UPDATE SALE

Call GENERATE DELIVERY ORDER

Call UPDATE ACCOUNT RECEIVABLE

MODULE: VALIDATE INPUT

PURPOSE: To obtain customer number , product number, and quantity desired

USE: -

RETURN : Customer number, product number, sold quantity

Begin

Read Cust_no, Product_no, qty_sold

End

MODULE: CREATE INVOICE

PURPOSE: To create invoice

USE: Cust_no, Product_no, Qty_sold

RETURN Invoice

Begin:

get DATE

get INVOICE FORMAT

get INVENTORY (product description, selling price)

compute SALE_AMOUNT (qty_sold * selling price)

Update INVOICE

print INV_NO, INV_DATE, CUST_NAME,

CUST_ADDRESS, PRODUCT_NO, PRODUCT_DESCRIPTION,

QTY_SOLD, SELLING_PRICE, SALE_AMOUNT,

TOTAL_SALE_AMOUNT)

End

MODULE: UPDATE INVENTORY

PURPOSE: to update the quantity in INVENTORY

USES : Product_no, Qty_sold

RETURN: -

Begin

Get INVENTORY (Quantity)

Quantity = Quantity - Qty_sold

End

MODULE: UPDATE SALE

PURPOSE: To update amount of sale

USES: INVOICE

RETURN: -

Begin

Get SALE

store SALE INVOICE in SALE

End

MODULE: GENERATE DELIVERY ORDER

PURPOSE: To generate Delivery Order

USES: Cust_no, Product_no, Qty_sold

RETURN: Delivery Order

Begin

get DATE

get FORMAT DELIVERY ORDER

get INVENTORY (product description)

update DELIVERY ORDER

print DO_NO, DO_DATE, CUST_NO, CUST_NAME, CUST_ADDRESS,

PRODUCT_NO, PRODUCT_DESCRIPTION, QTY_SOLD

END

MODULE: UPDATE ACCOUNT RECEIVABLE

PURPOSE: to record the credit sale into ACCOUNT RECEIVABLE

USES: Cust_no, Inv_no

RETURN: -

Begin

get INVOICE (cust_no, inv_no, sale_amount, total_sale_amount)

days_of_credit = TODAY DATE - INV_DATE

store INVOICE in ACCOUNT RECEIVABLE

where amount_owed = amount_owed + total_sale_amount

End

MODULE: RESPONSE TO CUSTOMER'S REQUEST

PURPOSE: to show the inventory list and validate the quantity availability

USES: -

RETURN: -

Begin

call SHOW LIST OF INVENTORY

call SEEK CUSTOMER'S REQUEST

End

MODULE: SHOW LIST OF INVENTORY

PURPOSE: to show the inventory list

USES: Customer's request

RETURN: list of inventory

Begin

get TIME

get FORMAT CUSTOMER'S REQUEST

get INVENTORY (Product_no, Product_description, selling price)

printscreen PRODUCT_NO, PRODUCT_DESCRIPTION, SELLING PRICE

End

MODULE: SEEK CUSTOMER'S REQUEST

PURPOSE: To seek the customer's available balance of desired inventory

USES: product_no

RETURN: quantity

Begin

get TIME

get FORMAT CUSTOMER'S REQUEST

get INVENTORY (quantity)

printscreen PRODUCT_NO, PRODUCT_DESCRIPTION, SELLING PRICE,
QUANTITY)

End

MODULE: GENERATE CREDIT SALE COLLECTION

PURPOSE: to collect the credit sale transaction

USES: -

RETURN: -

Begin

call VALIDATE INPUT

call GENERATE RECEIPT

call UPDATE ACCOUNT RECEIVABLE

End

MODULE: VALIDATE INPUT

PURPOSE: to read the invoice number of receipt

USES: -

RETURN: invoice no.

Begin

read INVOICE_NO

End

MODULE: GENERATE RECEIPT

PURPOSE: to issue receipt to customer

USES: inv_no

RETURN: Official Receipt

Begin

get INVOICE (cust_no, product_no, qty_sold, sale amount)

Read AMOUT RECEIPT

print CUST_NO, CUST_NAME, CUST_ADDRESS, PRODUCT_NO,
PRODUCT_DESCRIPTION, QTY_SOLD, SALE AMOUNT

End

MODULE: UPDATE ACCOUNT RECEIVABLE

PURPOSE: To update account receivable when receiving the collection

USES: Receipt

RETURN: -

Begin

get DATE

get RECEIPT (rv_no, rv_date, cust_no, cust_name, product_no, qty_
sold, amount_receipt)

match AMOUNT_RECEIPT with INV_NO in ACCOUNT

RECEIVABLE where amount_owed = amount_owed - amount_receipt

End

MODULE: UPDATE INVENTORY

PURPOSE: to update the inventory through receipt of stock

USES: -

RETURN: -

Begin

call GENERATE AVAILABILITY BALANCE

call UPDATE INVENTORY

End

MODULE: receive inventory

PURPOSE: to receive the inventory to stock

USES: -

RETURN: receipt quantity

Begin

READ STOCK RECEIPT (product_description, quantity)

get format INVENTORY

End

MODULE: UPDATE INVENTORY

PURPOST: to update inventory when receipt of stock

USES: product_no, receipt quantity

RETURN: -

Begin

get INVENTORY (quantity)

get FORMAT INVENTORY

QUANTITY = QUANTITY + RECEIPT_QTY

store QUANTITY in INVENTORY

End

MODULE: GENERATE REPORT

PURPOST: to generate the report as required by the related parties

USES : -

RETURN: report

Begin

get DATE

repeat until EOF

read FILE_NAME

get PRINTOUT, FILE_NAME

end repeat

End

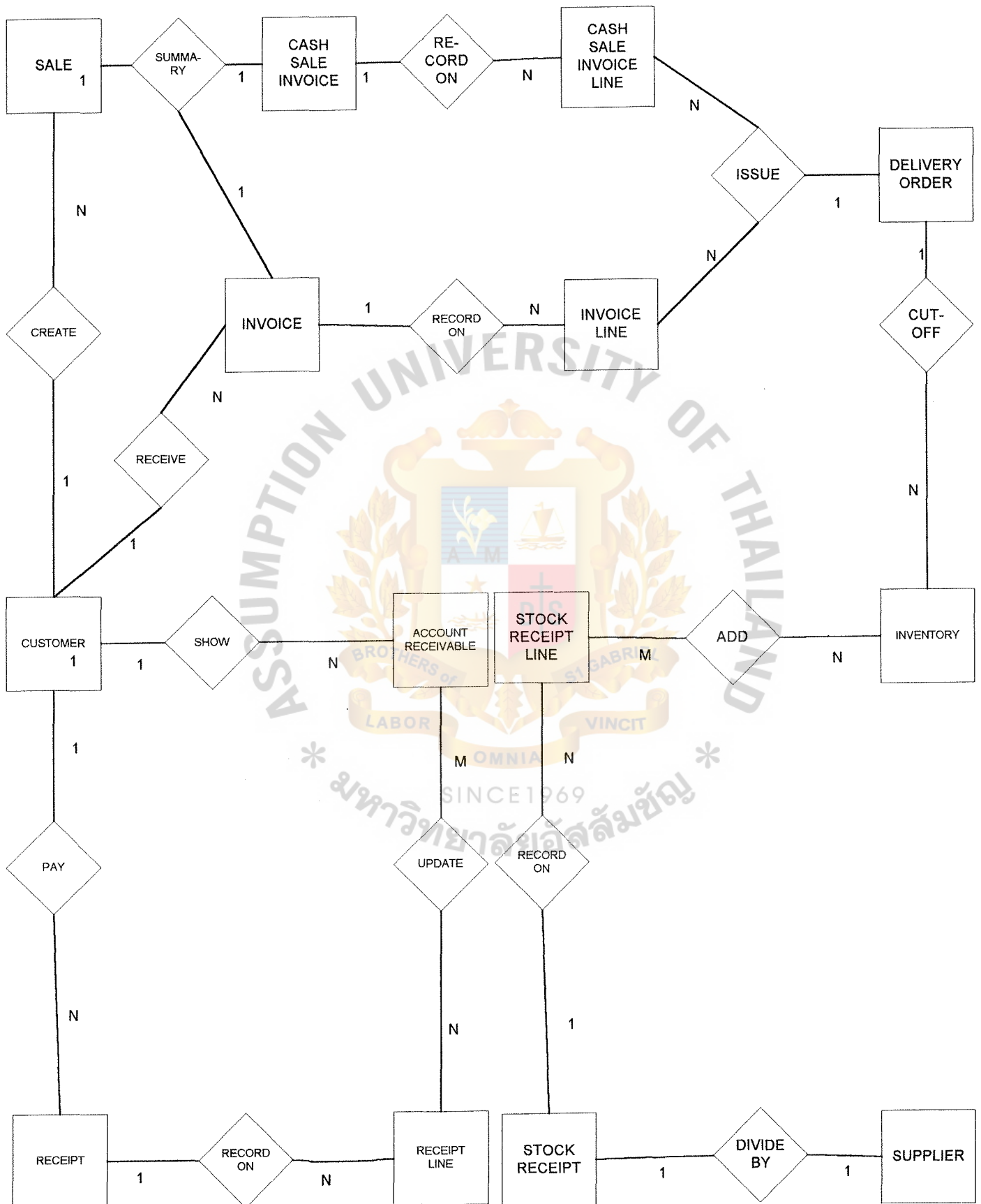


FIGURE F-1
ENTITY RELATIONSHIP DIAGRAM

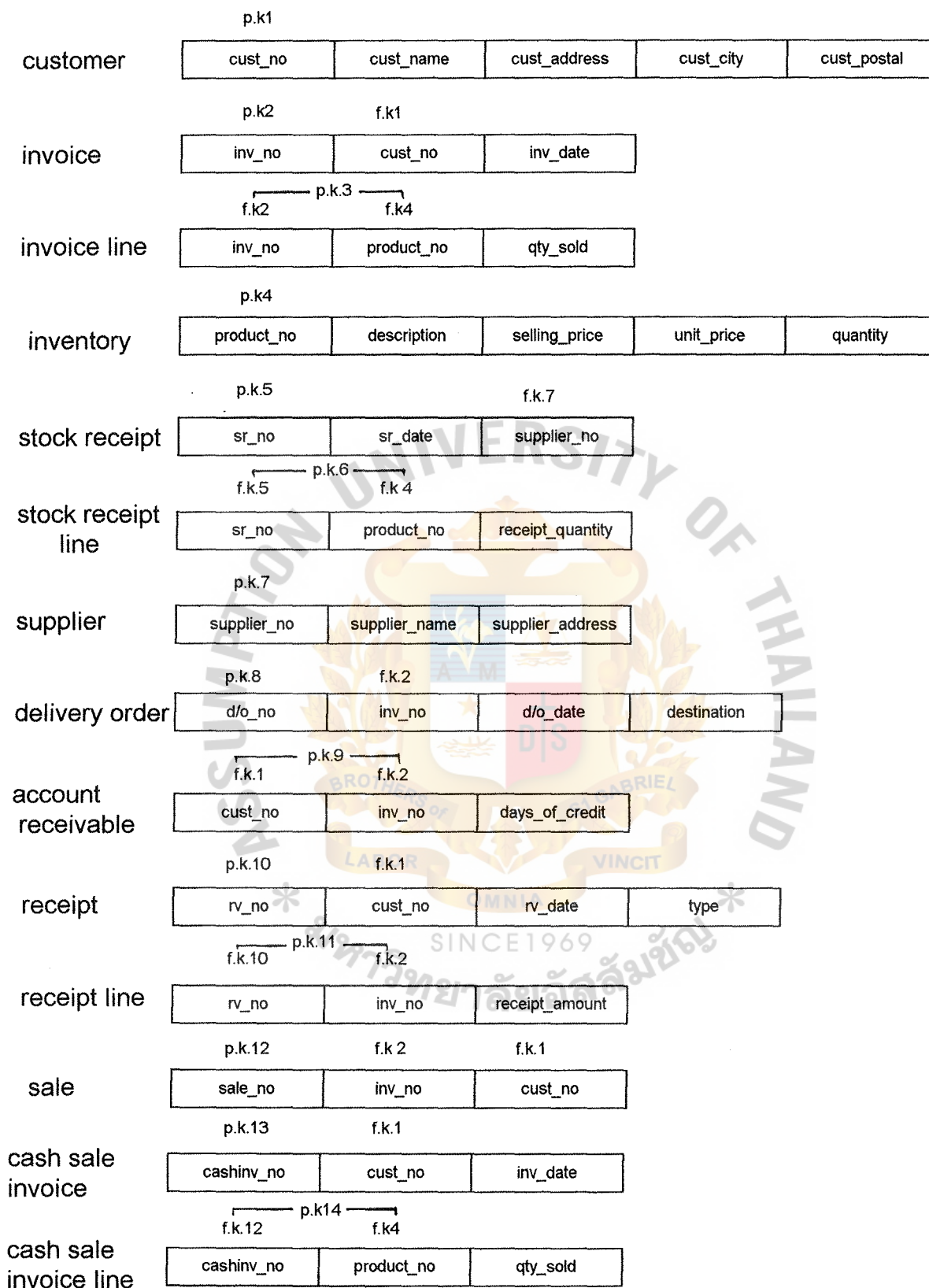


FIGURE F-2
ER Model Mapped to Relational Data Model (5 NF)

FUNCTIONAL DEPENDENCIES

CUSTOMER

CUST_NO	->	CUST_NAME
CUST_NO	->	CUST_ADDRESS
CUST_NO	->	CUST_CITY
CUST_NO	->	CUST_POSTAL

CASH SALE INVOICE

CASHINV_NO	->	CUST_NO
CASHINV_NO	->	CASHINV_DATE

CASH SALE INVOICE LINE

CASHINV_NO, PRODUCT_NO	->	QTY_SOLD
------------------------	----	----------

INVOICE

INV_NO	->	CUST_NO
INV_NO	->	INV_DATE

INVOICE LINE

INV_NO, PRODUCT_NO	->	QTY_SOLD
--------------------	----	----------

INVENTORY

PRODUCT_NO	->	DESCRIPTION
PRODUCT_NO	->	SELLING_PRICE
PRODUCT_NO	->	UNIT_PRICE
PRODUCT_NO	->	QUANTITY

STOCK RECEIPT

SR_NO	->	SR_DATE
SR_NO	->	SUPPLIER_NO

STOCK RECEIPT LINE

SR_NO, PRODUCT_NO -> RECEIPT_QTY

SUPPLIER

SUPPLIER_NO -> SUPPLIER_NAME
SUPPLIER_NO -> SUPPLIER_ADDRESS

DELIVERY ORDER

D/O_NO -> INV_NO
D/O_NO -> CASHINV_NO
D/O_NO -> CUST_NO
D/O_NO -> D/O_DATE
D/O_NO -> DESTINATION

ACCOUNT RECEIVABLE

CUST_NO, INV_NO -> DAYS_OF CREDIT

RECEIPT

RV_NO -> CUST_NO
RV_NO -> INV_NO
RV_NO -> RV_DATE
RV_NO -> TYPE

RECEIPT LINE

RV_NO, INV_NO -> RECEIPT_AMOUNT

SALE

SALE_NO -> INV_NO
SALE_NO -> CUST_NO

Microsoft Access - [customer]

File Edit View Records Window Help

NEW REVENUE CYCLE

WELCOME: PLEASE SELECT OPTION

SALE PROCESS	CASH SALE	
	CREDIT SALE	
	COLLECTION	
INVENTORY	DISPLAY	OCT. 15.96
	RECEIPT-IN	18.44
	REPORT	EXT

Record: 1 of 3

Form View

NUM

FIGURE G: MAIN MENU SCREEN

Microsoft Access - [customer]

File Edit View Records Window Help

17.4 REVENUE CYCLE

GENERATE CASH SALE

SALE PROCESS

PLS ENTER PWD

UPDATE CUSTOMER

CASH INVOICE

DELEVERT ORDER

OFFICIAL RECEIPT

SEP.12.96

18:44

EXIT

Record: 2 of 3

Form View

CAPS NUM

FIGURE G-1-: GENERATE CASH SALE - MAIN MENU SCREEN

ACCPRO Windows

CUSTOMER

Customer Description			
Cust No.	96002	Cust Name	Pochara Suthvong
Contact Name			
Addr. No.	425	Soi	Mitraparb Road Srinakarin
Tambon	Ornuch Amphur	Prakanong	Province Bangkok
Zip Code	10300	Telephone	3213442
FAX	3226446	TAX ID	0 Credit Term 30
Balance Information		Sale/Payment Information	
Credit Amt.	100000.00	Latest Buy Date	12/10/96 Latest Buy amt. 13900.00
Debt Amt.		Latest Paid Date	12/10/96 Latest Paid amt. 0.00
		Cust Level	1 Salesman No.
		123	

FIGURE G-1-1: GENERATE CASH SALE - CUSTOMER MENU SCREEN

ACCPRO Windows

cancel Save Print Look ADD EDIT DEL EXIT Del Row

Cust. No. 96002 Pochana Suthvong I/V No. CASH-96001 Date 13/10/39

Cust. 425 Sot Mitraparb SO. No. Date 13/10/39

Addr. RD. Snnakarn T. Onnuach Cond. 0 30 days Due Date 12/11/39

A. Prakanong Bangkok 10300

Tel. 3213442 Fax 3226446 Sales No. SA-101 Saleman 1

Goods Name	Unit	Loc.	Unit 1	Unit 2	Price	Discount	Amount
Panasonic Electric	11		2		6950.00		13900.00

Discount Rate		0.00
Total - Discount		12990.65
Tax Rate	7.00	909.35
Total		13900.00

☒ Inclusive VAT

Condition 0-Cash 1-Credit

FIGURE G-1-2: GENERATE CASH SALE - CASH INVOICE MENU SCREEN

ACCPRO for Windows

cancel Save Print Look ADD EDIT DEL EXIT Del. Row

Cust. No. 96002 Pochara Suthivong I/V No. INV-96002 Date 13/10/39
Cust. 425 So. Matraparb SO. No. DR-96001 Date 13/10/39
Add. RD. Sinakarn T. Onnuch Cond. 0 30 days Due Date 12/11/39
A. Prakanong Bangkok 10300 Sales No. SA-101 Salesman 1
Tel. 3213442 Fax 3226446

Goods Name	Unit	Loc.	Unit 1	Unit 2	Price	Discount	Amount
Panasonic Electric			2		6950.00		13900.00

Discount Rate 0.00
Total - Discount 12990.65
Tax Rate 7.00 909.35
Total 13900.00

☒ Inclusive VAT

Lookup Data in DATABASE

FIGURE G-1-4: GENERATE CASH SALE - OFFICIAL RECEIPT MENU
SCREEN E1969

Microsoft Access - [customer]

File Edit View Records Window Help

11.W REVENUE CYCLE

GENERATE CREDIT SALE

SALE PROCESS

UPDATE CUSTOMER

INVOICE

DELIVERY ORDER

SEP 12 96

18:44

PLS ENTER PWD

EXIT

Record: 3 of 3

Form View

CAPS NUM

FIGURE G-2: GENERATE CREDIT SALE - MAIN MENU SCREEN

ACCPRO for Windows

CUSTOMER

Customer Description

Cust No.	96001	Cust Name	Kritsada Damrongphol	
Contact Name				
Addr. No.	116	Soc.	Atsara 2	Road Baromrachachonee
Tambon	Arunamarn	Amphur	Bangkoknoi	Province Bangkok
Zip Code	10700	Telephone	4245591	
FAX	4245592	TAX ID	0144765335	Credit Term 30

Balance Information	Sales/Payment Information
Credit Amt. 100000.00	Latest Buy Date 12/10/39 Latest Buy amt 6500.00
Debt Amt. 0.00	Latest Paid Date 12/10/39 Latest Paid amt 6500.00
	Cust Level 1 Salesman No.

PRINT CUSTOMER

FIGURE G-2-1: GENERATE CREDIT SALE - CUSTOMER MENU SCREEN

ACCPRO for Windows - [DELIVE]

cancel Save Print Look ADD EDIT DEL EXIT Del Row

Cust. No. 96001 Kirtsada Damwongphol DO. No. DO-96001 Date 13/10/39
 Curt. 116 Sol Atswin 2 I/V No. INV-96001 Date 13/10/39
 Addr. RD. Boromratchachonee T. Arunamorn
 A. Bangkoknoi Bangkok 10700
 Tel. 4245591 Fax 4245592

Goods Name	Unit	Loc	Unit 1	Unit 2	Price	Discount	Amount	
Illinois Electric Drill			2		7500.00		15000.00	

Discount Rate		0.00
Total Discount		14018.69
Tax Rate	7.00	981.31
Total		15000.00

☒ Inclusive VAT

Lookup Data in DATABASE

FIGURE G-2-3: GENERATE CREDIT SALE -DELIVERY ORDER MENU
SCREEN E1969

Microsoft Access - [customer]

File Edit View Records Window Help

LOW REVENUE CYCLE

CREDIT SALE COLLECTION

CREDIT SALE COLLECTION OFFICIAL RECEIPT

SEP 12 86

PLS ENTER PWD

18 44

EXIT

Record: 1 of 3

Form View

CAPS NUM

FIGURE G-3: GENERATE CREDIT SALE-COLLECTION MENU SCREEN

ACCPRO for Windows

cancel Save Print Look ADD EDIT DEL EXIT Del Row

Cust. No.	96001	Kritsada Damrongphol	I/V No.	INV-96001	Date	13/10/39
Cust. Addr.	116	Sot Atsarin 2	SO. No.	OR-96002	Date	13/10/39
		RD. Boromratchachopee T. Arunamann	Cond.	0	30 days	Due Date 12/11/39
		A. Bangkoknoi Bangkok 10700	Sales No.			
Tel.	4245531		Fax	4245532		

Goods Name	Unit	Loc.	Unit 1	Unit 2	Price	Discount	Amount
Iluminous Electric Drill			2		7500.00		15000.00

☒ Inclusive VAT	Discount Rate	0.00
	Total - Discount	14018.69
	Tax Rate	7.00
	Total	15000.00

FIGURE G-3-1: CREDIT SALE-COLLECTION - OFFICIAL RECEIPT MENU SCREEN

GOODS									
cancel save print look ADD Edit DEL EXIT									
DESCRIPTION									
Goods No.	1-001						TYPE ☒ Goods ☐ Service		
Goods Name	Illinois Electric Drill								
Group	101	Unit No.		10					
Package	1	Std. Price	7500.00		Remark				
Buying Detail					Selling Detail				
BUY LEVEL	0.00	Latest Date	/ /		Latest Date	12/10/39	Price 1	7500.00	
Min	0.00	Latest Price			Latest Price	6000.00	2	7400.00	
Max	0.00	Not Receive	0.00				3	7300.00	
PRINT									

FIGURE G-4: DISPLAY INVENTORY MENU SCREEN

Microsoft Access - [customer]

File Edit View Records Window Help

I.V.W. REVENUE CYCLE

GENERATE REPORT

REPORT PROCESS

PLS ENTER PWD

SALE

CUSTOMER

INVENTORY

RECEIPT

A/R BALANCE

A/R AGING

SEP 12 96

18:44

EXIT

Record: 1 of 3

Form View

CAPS NUM

FIGURE G-6: GENERATE REPORT MENU SCREEN

ITW (THAILAND
SALE REPORT

AS OF 99/99/99 TIME 99:99:99
report # 1001

page 1/1

cust #	name	inv #	date	amount
9999	XXXXXXXXXXXXXXXXXXXX	99-9999	99/99/99	999,999,999.99
9999	XXXXXXXXXXXXXXXXXXXX	99-9999	99/99/99	999,999,999.99
9999	XXXXXXXXXXXXXXXXXXXX	99-9999	99/99/99	999,999,999.99
9999	XXXXXXXXXXXXXXXXXXXX	99-9999	99/99/99	999,999,999.99
9999	XXXXXXXXXXXXXXXXXXXX	99-9999	99/99/99	999,999,999.99
TOTAL				999,999,999.99

FIGURE H-1: SALE REPORT

ITW (THAILAND
CUSTOMER REPORT
AS OF 99/99/99 TIME 99/99/99
report# 1002

cust #	name /address	inv #	last purchase date	up to date total sale
9999	XXXXXXXXXXXXXXXXXXXX	99-999	99/99/99	999,999,999.99
9999	XXXXXXXXXXXXXXXXXXXX	99-999	99/99/99	999,999,999.99
9999	XXXXXXXXXXXXXXXXXXXX	99-999	99/99/99	999,999,999.99
			TOTAL	999,999,999.99

FIGURE H- 2: CUSTOMER REPORT

ITW (THAILAND)
INVENTORY REPORT
AS OF 99/99/99 TIME 99:99:99
report# 1003

page 1/1

product #	product name	date	in	out	qty	price	amount
999	XXXXXXXXXXXXXXXXXXXXX	B/F			99		
		99/99/99	99	99	99		
		99/99/99	99	99	99		
	total		99	99	99	999,999.99	999,999,999.99
999	XXXXXXXXXXXXXXXXXXXXX	B/F			99		
		99/99/99	99	99	99		
		99/99/99	99	99	99		
	total		99	99	99	999,999.99	999,999,999.99
999	XXXXXXXXXXXXXXXXXXXXX	B/F			99		
		99/99/99	99	99	99		
		99/99/99	99	99	99		
	total		99	99	99	999,999.99	999,999,999.99
999	XXXXXXXXXXXXXXXXXXXXX	B/F			99		
		99/99/99	99	99	99		
		99/99/99	99	99	99		
	total		99	99	99	999,999.99	999,999,999.99
999	XXXXXXXXXXXXXXXXXXXXX	B/F			99		
		99/99/99	99	99	99		
		99/99/99	99	99	99		
	total		99	99	99	999,999.99	999,999,999.99
						total	999,999,999.99

FIGURE H-3: INVENTORY REPORT

ITW (THAILAND
 RECEIPT REPORT
 AS OF 99/99/99 TIME 99/99/99
 report # 1005

rv #	rv date	name	inv #	date	amount	total
999	99/99/99	XXXXXXXXXXXXXXXXXXXX	99-999	99/99/99	999,999,999.99	
			99-999	99/99/99	999,999,999.99	999,999,999.99
999	99/99/99	XXXXXXXXXXXXXXXXXXXX	99-999	99/99/99	999,999,999.99	
			99-999	99/99/99	999,999,999.99	
			99-999	99/99/99	999,999,999.99	999,999,999.99
999	99/99/99	XXXXXXXXXXXXXXXXXXXX	99-999	99/99/99	999,999,999.99	
			99-999	99/99/99	999,999,999.99	
			99-999	99/99/99	999,999,999.99	999,999,999.99
999	99/99/99	XXXXXXXXXXXXXXXXXXXX	99-999	99/99/99	999,999,999.99	
			99-999	99/99/99	999,999,999.99	
			99-999	99/99/99	999,999,999.99	999,999,999.99
					-----	999,999,999.99

FIGURE H-4: RECEIPT REPORT

ITW (THAILAND
ACCOUNT RECEIVABLE REPORT
AS OF 99/99/99 TIME 99:99:99
report # 1005

page 1/1

cus.no	name	inv #	date	amount	total
99-999	XXXXXXXXXXXXXXXXXXXXX	99-999	99/99/99	999,999,999.99	
		99-999	99/99/99	999,999,999.99	
		99-999	99/99/99	999,999,999.99	999,999,999.99
99-999	XXXXXXXXXXXXXXXXXXXXX	99-999	99/99/99	999,999,999.99	
		99-999	99/99/99	999,999,999.99	
		99-999	99/99/99	999,999,999.99	999,999,999.99
99-999	XXXXXXXXXXXXXXXXXXXXX	99-999	99/99/99	999,999,999.99	
		99-999	99/99/99	999,999,999.99	
		99-999	99/99/99	999,999,999.99	999,999,999.99
99-999	XXXXXXXXXXXXXXXXXXXXX	99-999	99/99/99	999,999,999.99	
		99-999	99/99/99	999,999,999.99	
		99-999	99/99/99	999,999,999.99	999,999,999.99
99-999	XXXXXXXXXXXXXXXXXXXXX	99-999	99/99/99	999,999,999.99	
		99-999	99/99/99	999,999,999.99	
		99-999	99/99/99	999,999,999.99	999,999,999.99
				total	999,999,999.99

FIGURE H-5: ACCOUNT RECEIVABLE-OUTSTANDING BALANCE REPORT

ITW (THAILAND
ACCOUNT RECEIVABLE AGING REPORT
AS OF 99/99/99 TIME 99:99:99
report# 1001-1

page 1/1

cus.no	name	inv#	balance	current	30-60 days	over 60days
99-999	xxxxxxxxxxxxxxxxxxxxxxxx			999,999,999.99	999,999,999.99	999,999,999.99
		99-999		999,999,999.99	999,999,999.99	999,999,999.99
		99-999		999,999,999.99	999,999,999.99	999,999,999.99
		TOTAL	999,999,999.99	999,999,999.99	999,999,999.99	999,999,999.99
99-999	xxxxxxxxxxxxxxxxxxxxxxxx			999,999,999.99	999,999,999.99	999,999,999.99
		99-999		999,999,999.99	999,999,999.99	999,999,999.99
		99-999		999,999,999.99	999,999,999.99	999,999,999.99
		TOTAL	999,999,999.99	999,999,999.99	999,999,999.99	999,999,999.99
99-999	xxxxxxxxxxxxxxxxxxxxxxxx			999,999,999.99	999,999,999.99	999,999,999.99
		99-999		999,999,999.99	999,999,999.99	999,999,999.99
		99-999		999,999,999.99	999,999,999.99	999,999,999.99
		TOTAL	999,999,999.99	999,999,999.99	999,999,999.99	999,999,999.99
		GRAND TOTAL	999,999,999.99	999,999,999.99	999,999,999.99	999,999,999.99

FIGURE H-6: ACCOUNT RECEIVABLE-AGING BALANCE REPORT

I.T.W (THAILAND) CO., LTD.
 SALE BY PRODUCT REPORT
 AS OF 99/99/99 TIME 99:99:99

NO	JAN	FEB	MAR.	APRIL	MAY	JUNE	JULY	AUG	SEP	OCT.	NOV.	DEC.	TOTAL
AIR PUMP													
9999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.00	999,999.99	999,999.99	999,999.99
9999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.00	999,999.99	999,999.99	999,999.99
9999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.00	999,999.99	999,999.99	999,999.99
9999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.00	999,999.99	999,999.99	999,999.99
ELECTRIC DRILL													
9999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.00	999,999.99	999,999.99	999,999.99
9999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.00	999,999.99	999,999.99	999,999.99
9999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.00	999,999.99	999,999.99	999,999.99
9999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.00	999,999.99	999,999.99	999,999.99
SCREW													
9999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.00	999,999.99	999,999.99	999,999.99
9999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.00	999,999.99	999,999.99	999,999.99
9999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.00	999,999.99	999,999.99	999,999.99
9999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99	999,999.00	999,999.99	999,999.99	999,999.99

FIGURE H-7: SALE BY PRODUCT REPORT

[illegible]

FIGURE H-8: SALE BY CUSTOMER REPORT

I.T.W. (THAILAND) co.ltd.
 CUSTOMER - DIVISION REPORT
 AS OF 99/99/99 TIME 99:99:99

CUST#	NAME	ADDRESS	LAST PURCHASE	PRODUCT#
BANGKOK				
9999	XXXXXXXXXX	XXXXXXXXXX	99/99/99	9999
9999	XXXXXXXXXX	XXXXXXXXXX	99/99/99	9999
NORTHERN				
9999	XXXXXXXXXX	XXXXXXXXXX	99/99/99	9999
9999	XXXXXXXXXX	XXXXXXXXXX	99/99/99	9999
CENTRAL				
9999	XXXXXXXXXX	XXXXXXXXXX	99/99/99	9999
9999	XXXXXXXXXX	XXXXXXXXXX	99/99/99	9999
EASTERN				
9999	XXXXXXXXXX	XXXXXXXXXX	99/99/99	9999
9999	XXXXXXXXXX	XXXXXXXXXX	99/99/99	9999
WESTERN				
9999	XXXXXXXXXX	XXXXXXXXXX	99/99/99	9999
9999	XXXXXXXXXX	XXXXXXXXXX	99/99/99	9999
SOUTHERN				
9999	XXXXXXXXXX	XXXXXXXXXX	99/99/99	9999
9999	XXXXXXXXXX	XXXXXXXXXX	99/99/99	9999

FIGURE H-9: CUSTOMER DIVISION REPORT

I.T.W.(THAILAND)CO.,LTD.
 CUSTOMER- SEGMENT REPORT
 AS OF 99/99/99 TIME 99:99:99

CUST#	NAME	ADDRESS	BIRTHDATE	LAST PURCHASE
AGE BELOW 20				
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99
AGE 20 -35				
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99
AGE 35 -50				
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99
AGE OVER 50				
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99
9999	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXX	99/99/99	99/99/99

FIGURE H-10: CUSTOMER SEGMENT REPORT

I.T.W (THAILAND) CO., LTD.
GROSS PROFIT ANALYSIS BY PRODUCT
AS OF 99/99/99 TIME 99:99:99

ITEM NO.	NAME	UNIT SOLD	SALE PRICE	TOTAL SALES	COST PRICE	TOTAL COST	GROSS PROFIT
9999	XXXXXXXXXXXXXXXXXXXX	99999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99
9999	XXXXXXXXXXXXXXXXXXXX	99999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99
9999	XXXXXXXXXXXXXXXXXXXX	99999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99
9999	XXXXXXXXXXXXXXXXXXXX	99999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99
9999	XXXXXXXXXXXXXXXXXXXX	99999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99
9999	XXXXXXXXXXXXXXXXXXXX	99999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99
9999	XXXXXXXXXXXXXXXXXXXX	99999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99
9999	XXXXXXXXXXXXXXXXXXXX	99999	999,999.99	999,999.99	999,999.99	999,999.99	999,999.99

FIGURE H-11: GROSS PROFIT ANALYSIS BY PRODUCT REPORT

บริษัท ไอทีดับบลิว (ประเทศไทย) จำกัด
141 ST. LOUIS LANE II
SATHORN RD., BANGKOK
TEL: 222-1444 FAX: 222-1445

ใบกำกับสินค้า

TO _____

INVOICE NO. _____

DATE _____

[illegible]

CREDIT TERM	DAYS
1/10	10
1/15	15
1/20	20
1/30	30
2/10	10
2/15	15
2/20	20
2/30	30
3/10	10
3/15	15
3/20	20
3/30	30
4/10	10
4/15	15
4/20	20
4/30	30
5/10	10
5/15	15
5/20	20
5/30	30
6/10	10
6/15	15
6/20	20
6/30	30
7/10	10
7/15	15
7/20	20
7/30	30
8/10	10
8/15	15
8/20	20
8/30	30
9/10	10
9/15	15
9/20	20
9/30	30
10/10	10
10/15	15
10/20	20
10/30	30
11/10	10
11/15	15
11/20	20
11/30	30
12/10	10
12/15	15
12/20	20
12/30	30

-PLEASE NOTIFY US IMMEDIATELY THE INCORRECTNESS OF ACCOUNT

GROSS

VAT

TOTAL

CHECKED _____ APPROVED _____ CREDIT DEPT.. _____

FIGURE : H- 12 INVOICE

บริษัท ไอทีดับบลิว (ประเทศไทย) จำกัด
141 ST. LOUIS LANE II
SATHORN RD., BANGKOK
TEL: 222-1444 FAX: 222-1445

ใบส่งสินค้า

TO _____

DELIVERY ORDER NO. _____

INVOICE NO. _____

CASH INVOICE NO. _____

DATE _____

[illegible]

VAT

TOTAL

-PLEASE NOTIFY US IMMEDIATELY THE INCORRECTNESS OF ACCOUNT

CERTIFY AMOUNT RECEIVED TRUE AND CORRECT.

CHECKED

RECEIVED

STOCK DEPT..

APPROVAL

FIGURE : H-13: DELIVERY ORDER

บริษัท โอทีดับบลิว (ประเทศไทย) จำกัด
141 ST. LOUIS LANE II
SATHORN RD., BANGKOK
TEL: 222-1444 FAX: 222-1445

ใบกำกับสินค้า/ขายเงินสด

CASH INVOICE NO.

DATE _____

[illegible]

GROSS
VAT
TOTAL

-PLEASE NOTIFY US IMMEDIATELY THE INCORRECTNESS OF ACCOUNT

CHECKED _____ APPROVED _____ SALE DEPT. _____

FIGURE : H-14 CASH SALE INVOICE

บริษัท โอทีดับบลิว (ประเทศไทย) จำกัด
141 ST. LOUIS LANE II
SATHORN RD., BANGKOK
TEL: 222-1444 FAX: 222-1445

ใบเสร็จรับเงิน

CASH INVOICE NO.

RECEIPT VOUCHER NO

RECEIPT DATE

[illegible]

VAT

TOTAL _____

RECEIVED

RECEIVED BY ☐ CASH ☐ CHECK NUMBER ☐ TRANSFER OF ACCOUNT BANK NO

CASHIER ACCOUNTING DEPT. APPROVED.

FIGURE : H-15: OFFICIAL RECEIPT

PROGRAM CODING

```
Const xINV_DATE = 1
Const xSO_NO = 2
Const xSO_DATE = 3
Const xCUST_NO = 4
Const xCUST_NAME = 5
Const xAddr1 = 6
Const xAddr2 = 7
Const xAddr3 = 14
Const xTEL = 8
Const xFAX = 9
Const xCOND = 10
Const xCRTERM = 11
Const xDUE_DATE = 12
Const xSALE_NO = 13
```

```
Private Sub AddRec()
    Dim Msg As String ' Declare variables.
    Criteria = "INV_NO=" + TB_FIELD(xINV_NO).Text + ""
    BICARSA.Data1(BLHINV).Recordset.FindFirst Criteria ' Make record current.
    If Not BICARSA.Data1(BLHINV).Recordset.NoMatch Then ' Found
        Msg = "duplicated data"
        Msg = Msg & "press any key"
        Call ErrBox(Msg)
        ErrorFlag = True
    End If
End Sub
```

```
Private Function AddSpace(zStr As String, zLen As Integer) As String
    Dim i, zStrLen As Integer
    zStrLen = Len(zStr)
    For i = zStrLen + 1 To zLen
        zStr = zStr + " "
    Next i
    AddSpace = zStr
End Function
```

```
Private Sub BltStore()
    Dim i As Integer ' Declare variables.
    Dim zDoub As Double
    Dim zNum As Double
    Dim zDiscRate As Double
    Dim zPart_no As String
    Dim zStr As String
    Criteria = "INV_NO=" + TB_FIELD(xINV_NO).Text + "" ' Create the criteria.
    BICARSA.Data1(BLDINV).Recordset.FindFirst Criteria ' Make record current.
    i = 1
    If Not BICARSA.Data1(BLDINV).Recordset.NoMatch Then ' Found
        Do While StoreField(BLDINV, "INV_NO") = TB_FIELD(xINV_NO).Text
            Grid1.Row = i
            Grid1.Col = 1
            Grid1.Text = StoreField(BLDINV, "PART_NO")
            zPart_no = RTrim(Grid1.Text)
            DispProduct (Grid1.Text)
            DispProduct (zPart_no)
        Loop
    End If
End Sub
```



```

BICARSA.Data1(BLDINV).Recordset.FindNext Criteria
    If BICARSA.Data1(BLDINV).Recordset.NoMatch Then 'Not Found
        Exit Do
    End If
    i = i + 1
Loop
Grid2.Col = 1
Call DispTotal
End If
End Sub

Private Sub BT3D_ADD_Click()
    Screen.MousePointer = 11
    SwitchACD (False)
    BT3D_ADD.BevelWidth = 0
    SwitchGet (True)
    ACDFLAG = Add
    Screen.MousePointer = 0
End Sub

Private Sub BT3D_ADD_GotFocus()
    BICARSA.MessBar.Caption = "Add Mode"
End Sub

Private Sub BT3D_ADD_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    BICARSA.MessBar.Caption = "Add Mode"
End Sub

Private Sub BT3D_CANCEL_Click()
    'Msg = "confirm cancell"
    DelFrom.Caption = "Cancel"
    DelFrom.Label1.Caption = "Confirm Cancel!"
    DelFrom.Show 1
    If MsgBox(Msg, 48 + 4 + 256, "Cancel") = IDYES Then ' Get
user response.
        Screen.MousePointer = 11
        Call ClearVar
        Call SwitchGet(False)
        Call SwitchACD(True)
        Screen.MousePointer = 0
    End If
End Sub

Private Sub BT3D_CANCEL_GotFocus()
    BICARSA.MessBar.Caption = "Cancel!"
End Sub

Private Sub BT3D_CANCEL_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    BICARSA.MessBar.Caption = "Cancel !"
End Sub

Private Sub BT3D_DEL_Click()
    Screen.MousePointer = 11
    SwitchACD (False)
    BT3D_DEL.BevelWidth = 0

```

```

If Not ErrorFlag Then
    Grid1.Col = 0
    Grid1.Text = StoreField(ICMPROD, "PROD_NAME")
Else
    ErrorFlag = False
End If Const xINV_NO = 0
Grid1.Col = 2
If StoreField(BLDINV, "UNIT_NO") <> "" Then
    Grid1.Text = StoreField(BLDINV, "UNIT_NO")
Else
    Grid1.Text = MinorUnit(zPart_no)
End If
zStr = Grid1.Text
zRate# = Val(SeekString(ICMUNIT, "UNIT_NO", "RATE", zStr))
Grid1.Col = 3
If Not IsNull(StoreField(BLDINV, "LOC_NO")) Then
    Grid1.Text = StoreField(BLDINV, "LOC_NO")
End If
Grid1.Col = 4
zDoub = BICARSA.Data1(BLDINV).Recordset.Fields("QTY")
If zDoub <> 0 Then
    zMajor# = Fix(zDoub / zRate#)
    zMinor# = zDoub - (zMajor# * zRate#)
    zStr = Trim(Str(zMajor#)) & "." & Trim(Str(zMinor#))
    Grid1.Text = Format(zMajor#, "##0.00") & Format(zStr, "##0.00")
    Grid1.Col = 5
    Grid1.Text = Format(zMinor#, "##0.00")
    zNum = zDoub
End If
Grid1.Col = 6
zDoub = BICARSA.Data1(BLDINV).Recordset.Fields("PRICE")
If zDoub <> 0 Then
    Grid1.Text = Format(zDoub, "##0.00")
    zNum = zNum * zDoub
End If
Grid1.Col = 7
If Not IsNull(BICARSA.Data1(BLDINV).Recordset.Fields("DISC")) Then
    zDoub = BICARSA.Data1(BLDINV).Recordset.Fields("DISC")
End If
If zDoub <> 0 Then
    zDiscRate = zDoub * 100 / zNum
    Grid1.Text = Format(zDiscRate, "##0.00")
End If
If Not IsNull(BICARSA.Data1(BLDINV).Recordset.Fields("DiscCond"))
Then
    Grid1.Text = BICARSA.Data1(BLDINV).Recordset.Fields("DiscCond")
End If
Grid1.Col = 8
zDoub = BICARSA.Data1(BLDINV).Recordset.Fields("AMT")
If zDoub <> 0 Then
    Grid1.Text = Format(zDoub, "##0.00")
End If
BICARSA.Data1(BLDINV).Recordset.MoveNext
If BICARSA.Data1(BLDINV).Recordset.EOF Then
    Exit Do
End If

```

```
SwitchGet (True)
  BT3D_SAVE.Enabled = False
  ACDFLAG = Del
  Screen.MousePointer = 0
End Sub
```

```
Private Sub BT3D_DEL_GotFocus()
  BICARSA.MessBar.Caption = "Delete Mode"
End Sub
```

```
Private Sub BT3D_DEL_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
  BICARSA.MessBar.Caption = "Delete Mode"
End Sub
```

```
Private Sub BT3D_DELROW_Click()
  Screen.MousePointer = 11
  Dim i, j, CurRow, CurCol As Integer
  Dim zStr As String
  CurRow = Grid1.Row
  CurCol = Grid1.Col
  For i = Grid1.Row To Grid1.Rows - 2
    Grid1.Col = Grid1.LeftCol
    If Grid1.Text <> "" Then
      For j = Grid1.LeftCol To Grid1.Cols - 1
        Grid1.Row = i + 1
        Grid1.Col = j
        zStr = Grid1.Text
        Grid1.Row = i
        Grid1.Text = zStr
      Next j
    End If
  Next i
  DispTotal
  Grid1.Row = CurRow
  Grid1.Col = CurCol
  Grid1.SetFocus
  Screen.MousePointer = 0
End Sub
```

```
Private Sub BT3D_DELROW_GotFocus()
  BICARSA.MessBar.Caption = "Delete Row in Table"
End Sub
```

```
Private Sub BT3D_EDIT_Click()
  Screen.MousePointer = 11
  SwitchACD (False)
  BT3D_EDIT.BevelWidth = 0
  SwitchGet (True)
  ACDFLAG = Change
  Screen.MousePointer = 0
End Sub
```

```
Private Sub BT3D_EDIT_GotFocus()
  BICARSA.MessBar.Caption = "Edit Mode"
End Sub
```

```
Private Sub BT3D_EDIT_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    BICARSA.MessBar.Caption = "Edit Mode"
```

```
End Sub
```

```
Private Sub BT3D_EXIT_Click()
```

```
    'Msg = "confirm exit"
```

```
    DelFrom.Caption = "Exit Program"
```

```
    DelFrom.Label1.Caption = "Please Confirm!"
```

```
    DelFrom.Show 1
```

```
    If MsgReturn Then 'MsgBox(Msg, 48 + 4 + 256, "exit from program") = IDYES
```

```
Then ' Get user response.
```

```
    BICARSA.MessBar.Caption = " "
```

```
    Unload FORM_SATINV
```

```
End If
```

```
End Sub
```

```
Private Sub BT3D_EXIT_GotFocus()
```

```
    BICARSA.MessBar.Caption = "Exit Program"
```

```
End Sub
```

```
Private Sub BT3D_EXIT_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    BICARSA.MessBar.Caption = "Exit Program"
```

```
End Sub
```

```
Private Sub BT3D_Help_Click()
```

```
' Msg = "Help Module!"
```

```
' Msg = Msg & " To Be Supply"
```

```
' MsgBox Msg, 48, "Help" ' Get user response.
```

```
'FORM_PRNINV.Show
```

```
Screen.MousePointer = 11
```

```
Temp% = WinHelp(hWnd, HelpFile, HELP_CONTEXT, CLng(6100))
```

```
Screen.MousePointer = 0
```

```
End Sub
```

```
Private Sub BT3D_HELP_GotFocus()
```

```
    BICARSA.MessBar.Caption = "Help"
```

```
End Sub
```

```
Private Sub BT3D_HELP_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    BICARSA.MessBar.Caption = "Help"
```

```
End Sub
```

```
Private Sub BT3D_LOOK_Click()
```

```
Screen.MousePointer = 11
```

```
If LookField < MaxField Then
```

```
    Select Case LookField
```

```
        Case xINV_NO
```

```
            Screen.MousePointer = 11
```

```
            BT3D_LOOK.SetFocus
```

```
            Lookup.Caption = "INVOICE"
```

```
            Lookup.Title1 = "Inv. No."
```

```
            Lookup.Title2 = "Inv. Date"
```

```
            Lookup.DataIndex = BLHINV
```

```
            Lookup.Field1 = "INV_NO"
```

```
            Lookup.Field2 = "INV_DATE"
```



```

FORM_LOOKUP.Show
    FORM_LOOKUP.Show
    Case xCUST_NO
        Screen.MousePointer = 11
        BT3D_LOOK.SetFocus
        Lookup_Caption = "CUSTOMER"
        Lookup_Title1 = "Cust. No."
        Lookup_Title2 = "Cust. Name"
        Lookup_DataIndex = ARMCUST
        Lookup_Field1 = "CUST_NO"
        Lookup_Field2 = "CUST_NAME"
        FORM_LOOKUP.Show
    End Select
Else
    Select Case Grid1.Col
        Case 1
            Screen.MousePointer = 11
            BT3D_LOOK.SetFocus
            Lookup_Caption = "GOODS"
            Lookup_Title1 = "Goods No."
            Lookup_Title2 = "Goods Name"
            Lookup_DataIndex = ICMPROD
            Lookup_Field1 = "PART_NO"
            Lookup_Field2 = "PROD_NAME"
            FORM_LOOKUP.Show
        Case 2
            Screen.MousePointer = 11
            BT3D_LOOK.SetFocus
            Lookup_Caption = "UNIT"
            Lookup_Title1 = "No."
            Lookup_Title2 = "Unit 1"
            Lookup_Title3 = "Unit 2"
            Lookup_Title4 = "Rate"
            Lookup_DataIndex = ICMUNIT
            Lookup_Field1 = "UNIT_NO"
            Lookup_Field2 = "UNIT1"
            Lookup_Field3 = "UNIT2"
            Lookup_Field4 = "RATE"
            Lookup_unit = True
            FORM_LOOKUP3.Show
        End If
    Screen.MousePointer = 0
End Sub

Private Sub BT3D_LOOK_GotFocus()
    BICARSA.MessBar.Caption = "Lookup Date in DATABASE"
    If Lookup Then
        Lookup = False
        If LookField < MaxField Then
            If Len(Lookup_RetVal) > 0 Then
                TB_FIELD(LookField).Text = Lookup_RetVal
                TB_FIELD(LookField).SetFocus
                Call TB_FIELD_KeyPress(LookField, 13)
            Else
                TB_FIELD(LookField).SetFocus
            End If
        Else
            TB_FIELD(LookField).SetFocus
        End If
    Else
        TB_FIELD(LookField).SetFocus
    End If
End Sub

```

```

If Len(Lookup_RetVal) > 0 Then
    Text1.Text = " "
    Text1.MaxLength = 10 'Space(Len(Lookup_RetVal))
    Text1.Text = Lookup_RetVal
    Text1.Visible = True
    Text1_KeyPress (13)
Else
    Grid1.SetFocus
End If
End If
End If
End Sub

Private Sub BT3D_LOOK_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    BICARSA.MessBar.Caption = "Lookup Data in DATABASE"
End Sub

Private Sub BT3D_PRINT_Click()
    Screen.MousePointer = 11
    Dim zStr As String
    zStr = Format(SeekString(COMPANY, "COMP_NO", "INV_TYPE", ""), "#")
    If zStr = "1" Then
        'FORM_PRNINV.Show
    Else
        BICARSA.Report1.Destination = 1
        zStr = TB_FIELD(xINV_NO).Text
        a = SeekString(BLHINV, "INV_NO", "INV_NO", zStr)
        If Not ErrorFlag Then
            FmlaText$ = "{BLHINV.INV_NO} = " + zStr + ""
            BICARSA.Report1.ReportFileName = ReportDir + "SARTINV.RPT"
            On Error Resume Next
            BICARSA.Report1.SelectionFormula = FmlaText$
            BICARSA.Report1.Action = 1
            If Err <> 0 Then
                Screen.MousePointer = 0
                If Err = 20514 Then
                    Call ErrBox("Error : 20514 ,Text file already exist!")
                Else
                    Call ErrBox("Error : " + Str(Err))
                End If
                Screen.MousePointer = 11
            End If
        End If
    End If
    Screen.MousePointer = 0
End Sub

Private Sub BT3D_PRINT_GotFocus()
    BICARSA.MessBar.Caption = "PRINT INVOICE"
End Sub

Private Sub BT3D_PRINT_MouseMove(Button As Integer, Shift As Integer, X As
Single, Y As Single)
    BICARSA.MessBar.Caption = "PRINT INVOICE"
End Sub

```

```

Private Sub BT3D_SAVE_Click()
    Screen.MousePointer = 11
    If DataCorrect() Then
        Screen.MousePointer = 11 'err
        Call Updatesatinv
        Call OnlineProcess
        Screen.MousePointer = 0
        MsgReturn = MsgBox("PRINT INVOICE", 1 + 32 + 256, "Confirm !")
        Screen.MousePointer = 11
        If MsgReturn = IDOK Then
            Call BT3D_PRINT_Click
        Else
            Call ClearVar
        End If
    End If
    Screen.MousePointer = 0
End Sub

```

```

Private Sub BT3D_SAVE_GotFocus()
    BICARSA.MessBar.Caption = "SAVE DATA"
End Sub

```

```

Private Sub BT3D_SAVE_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    BICARSA.MessBar.Caption = "SAVE DATA"
End Sub

```

```

Private Sub Check3D1_Click(Value As Integer)
    If Value Then
        VatType = 3
    Else
        VatType = 0
    End If
End Sub

```

```

Private Sub CheckVAT_Click(Value As Integer)
    If Value Then
        VatType = 3
    Else
        VatType = 0
    End If
    Call DispTotal
End Sub

```

```

Private Sub ClearDescription(zIndex As Integer)

```

```

    Select Case zIndex
        Case 5 'Clear Sale Description
            LB_DISP(zIndex).Caption = ""
        Case 8
            TB_FIELD(xCUST_NAME).Text = ""
            TB_FIELD(xAddr1).Text = ""
            TB_FIELD(xAddr2).Text = ""
            TB_FIELD(xAddr3).Text = "" 'bbb
            TB_FIELD(xTEL).Text = ""
            TB_FIELD(xFAX).Text = ""
    End Select
End Sub

```

```

Private Sub ClearVar()
    Dim i, j As Integer
    If TB_FIELD(xINV_NO).Enabled Then
        For i = 0 To MaxField - 1
            If TB_FIELD(i).Mask = "###/###/###" Then
                TB_FIELD(i).Text = Today()
            Else
                TB_FIELD(i).Text = ""
            End If
        Next i
        TB_FIELD(xINV_NO).SetFocus
        For i = 1 To Grid1.Rows - 1
            Grid1.Row = i
            For j = 0 To Grid1.Cols - 1
                Grid1.Col = j
                Grid1.Text = " "
            Next j
        Next i
        For i = 0 To Grid2.Rows - 1
            Grid2.Row = i
            For j = 1 To Grid2.Cols - 1
                Grid2.Col = j
                Grid2.Text = " "
            Next j
        Next i
        Grid2.Row = 2
        Grid2.Col = 1
        Grid2.Text = Format(SeekString(COMPANY, "COMP_NO", "VAT_RATE",
""), "##0.00")
        Grid1.SelStartRow = 1
        Grid1.SelEndRow = 1
        Grid1.SelStartCol = 1
        Grid1.SelEndCol = 1
        Grid2.Col = 1
        Grid2.Row = 0
        Text1.Visible = False
        Text2.Visible = False
        TB_FIELD(xCOND).Text = 0
    End If
    Screen.MousePointer = 0
End Sub

```

```

Private Function DataCorrect() As Integer
    Dim i As Integer
    Dim zStr As String
    Grid1.Col = 1
    ErrorFlag = False
    For i = 1 To Grid1.Rows - 1
        Grid1.Row = i
        If Trim(Grid1.Text) <> "" Then
            DispProduct (Grid1.Text)
            If ErrorFlag Then
                DataCorrect = False
                Exit Function
            End If
        End If
    End If
    Next i

```



```

For i = 0 To MaxField - 1
    If TB_FIELD(i).Mask = "#####" Then
        zStr = TB_FIELD(i).Text
        If Not ValidDate(zStr) Then
            TB_FIELD(i).Text = Today()
            Msg = "Invalid Date"
            Msg = Msg & " Press any key to continue... "
            Call ErrBox(Msg)
            DataCorrect = False
            Exit Function
        End If
        If Not PeriodOpen(zStr) Then
            Msg = "Invalid Period"
            Msg = Msg & " Press any key to continue... "
            Call ErrBox(Msg)
            DataCorrect = False
            Exit Function
        End If
    End If
Next i
DataCorrect = True
End Function

Private Sub DelRec()
    Call EditRec
    If Not ErrorFlag Then
        MsgReturn = MsgBox("Delete", 1 + 48 + 256, "Confirm Delete")
        If MsgReturn Then
            Criteria = "INV_NO=" + TB_FIELD(xINV_NO).Text + "" 'Create the
criteria.
            BICARSA.Data1(BLHINV).Recordset.FindFirst Criteria ' Make record
current.
            If Not BICARSA.Data1(BLHINV).Recordset.NoMatch Then ' Found
                On Error Resume Next
                Call DelOldTran ' Delete old transaction
                BICARSA.Data1(BLHINV).Recordset.Delete
                If Err > 0 Then
                    Call ErrBox("Error : " + Str(Err))
                End If
            End If
        End If
        'Call ClearVar
    End If
    Call ClearVar
End If
End Sub

Private Sub DispMessage(FieldNo As Integer)
    Select Case FieldNo
        Case xINV_NO
            BICARSA.MessBar.Caption = "Invoice No."
        Case xINV_DATE
            BICARSA.MessBar.Caption = "Invoice Date"
        Case xSO_NO
            BICARSA.MessBar.Caption = "Sales Order No."
        Case xSO_DATE
            BICARSA.MessBar.Caption = "Sales Order Date"
        Case xCOND
            BICARSA.MessBar.Caption = "Condition 0=Cash 1=Credit"
    End Select

```

```

        Case xCRTERM
            BICARSA.MessBar.Caption = "Due"
        Case xDUE_DATE
            BICARSA.MessBar.Caption = "Due Date"
        Case xSALE_NO
            BICARSA.MessBar.Caption = "Salesman No."
        Case xCUST_NO
            BICARSA.MessBar.Caption = "Customer No."
        Case xCUST_NAME
            BICARSA.MessBar.Caption = "Customer Name"
        Case xAddr1, xAddr2, xAddr3 'bbb
            BICARSA.MessBar.Caption = "Customer Address"
        Case xTEL
            BICARSA.MessBar.Caption = "Telephone"
        Case xFAX
            BICARSA.MessBar.Caption = "Fax"
        Case 14
            BICARSA.MessBar.Caption = "Goods No."
        Case 15
            BICARSA.MessBar.Caption = "Unit No."
        Case 16
            BICARSA.MessBar.Caption = "Location No."
        Case 17
            BICARSA.MessBar.Caption = "Unit"
        Case 18
            BICARSA.MessBar.Caption = "Price"
        Case 19
            BICARSA.MessBar.Caption = "Discount"
        Case 20
            BICARSA.MessBar.Caption = "Amount"
    End Select

    BT3D_DELROW.Enabled = IIf(FieldNo < MaxField, False, True)
End Sub

Private Sub DispProduct(zKey As String)
    Criteria = "PART_NO = " + zKey + ""
    BICARSA.Data1(ICMPROD).Refresh
    BICARSA.Data1(ICMPROD).Recordset.FindFirst Criteria ' Make record current.
    If BICARSA.Data1(ICMPROD).Recordset.NoMatch Then ' Found
        Msg = "Not Found"
        'Msg = Msg & " Pre"
        Call ErrBox(Msg)
        ErrorFlag = True
    End If
End Sub

Private Sub DispTotal()
    Dim i, OldRow, OldCol, OldRow2, OldCol2 As Integer
    Dim zDiscount, zTotal1, zVat, zTotal2, zDispRate, zVatRate As Double
    Dim Total As Double
    Dim zStr As String
    Total = 0
    OldRow = Grid1.Row
    OldCol = Grid1.Col
    OldRow2 = Grid2.Row
    OldCol2 = Grid2.Col

```

```

For i = 1 To Grid1.Rows - 1
    Grid1.Row = i
    Grid1.Col = 8
    Total = Total + Val(Grid1.Text)
Next i
Grid2.Col = 1
Grid2.Row = 2
zVatRate = Val(Grid2.Text)          'Read Discount From Grid2
Grid2.Col = OldCol2

If Grid2.Col < 2 Then
    Grid2.Col = 1
    Grid2.Row = 0
    'zDispRate = Val(Grid2.Text)      'Read Discount From Grid2
    zStr = Grid2.Text
    Grid2.Col = 2
    Grid2.Row = 0
    'Grid2.Text = Format(Total * zDispRate / 100, "##0.00")
    Grid2.Text = Format(MultiDiscount(Total, zStr), "##0.00")
    zDiscount = Val(Grid2.Text)      'Read Discount From Grid2
    VatType = IIf(CheckVAT, 3, 0)
    If VatType = 3 Then              'Inclusive
        Grid2.Row = 3              'Total Amount
    Else
        Grid2.Row = 1              'Amount of Product & Service
    End If
    zTotal1 = Total - zDiscount
    Grid2.Text = Format(zTotal1, "##0.00")
    Grid2.Row = 2
    If VatType = 3 Then              'Inclusive
        Grid2.Text = Format(zTotal1 * zVatRate / (100 + zVatRate), "##0.00")
        zVat = Val(Grid2.Text)      'Read Discount From Grid2
        Grid2.Row = 1              'Write Total After Discount
        zTotal2 = zTotal1 - zVat
        Grid2.Text = Format(zTotal2, "##0.00")
    Else
        Grid2.Text = Format(zTotal1 * zVatRate / 100, "##0.00")
        zVat = Val(Grid2.Text)      'Read Discount From Grid2
        Grid2.Row = 3              'Write Total After Discount
        zTotal2 = zTotal1 + zVat
        Grid2.Text = Format(zTotal2, "##0.00")
    End If
Else
    Grid2.Col = 2
    Grid2.Row = 0
    zDiscount = Val(Grid2.Text)      'Read Discount From Grid2
    If Total <> 0 Then
        zDispRate = zDiscount * 100 / Total  'Read Discount From Grid2
    Else
        zDispRate = 0
    End If
    Grid2.Col = 1
    'Grid2.Text = Format(zDispRate, "##0.00")
    Grid2.Col = 2
    If VatType = 3 Then              'Inclusive
        Grid2.Row = 3              'Total Amount
    Else
        Grid2.Row = 1              'Amount of Product & Service
    End If

```

```

zTotal1 = Total - zDiscount
Grid2.Text = Format(zTotal1, "##0.00")

Grid2.Row = 2
zVat = Val(Grid2.Text)
If VatType = 3 Then 'Inclusive
    Grid2.Text = Format(zTotal1 * zVatRate / (100 + zVatRate), "##0.00")
    zVat = Val(Grid2.Text) 'Read Discount From Grid2
    Grid2.Row = 1 'Write Total After Discount
    zTotal2 = zTotal1 - zVat
    Grid2.Text = Format(zTotal2, "##0.00")
Else
    Grid2.Text = Format(zTotal1 * zVatRate / 100, "##0.00")
    zVat = Val(Grid2.Text) 'Read Discount From Grid2
    Grid2.Row = 3 'Write Total After Discount
    zTotal2 = zTotal1 + zVat
    Grid2.Text = Format(zTotal2, "##0.00")
End If
End If
Grid2.Row = OldRow2
Grid2.Col = OldCol2
Grid1.Row = OldRow
Grid1.Col = OldCol
End Sub

Private Sub EditRec()
    Dim Msg As String 'Declare variables.
    Dim zStr As String
    Dim zLong As Long, zDouble1, zDouble2
    Dim zTotal As Double
    a = TB_FIELD(0).Text
    Call ClearVar
    TB_FIELD(0).Text = a
    Criteria = "INV_NO=" + TB_FIELD(xINV_NO).Text + "" 'a
    BICARSA.Data1(BLHINV).Recordset.FindFirst Criteria 'Make record current.
    If BICARSA.Data1(BLHINV).Recordset.NoMatch Then 'Not Found
        Msg = "could not find defined data"
        Msg = Msg & " press any key"
        Call ErrBox(Msg)
        ErrorFlag = True
        Call ClearVar
        Exit Sub
    Else
        zStr = TB_FIELD(xINV_NO).Text
        zDouble1 = SeekString(ARTSA, "DOC_NO", "AMT", zStr)
        zStr = TB_FIELD(xINV_NO).Text
        zDouble2 = SeekString(ARTSA, "DOC_NO", "BAL", zStr)
        If zDouble1 > zDouble2 Then
            Msg = "Can not edit! It's already PAID..."
            Msg = Msg & " Press any key"
            Call ErrBox(Msg)
            ErrorFlag = True
        End If
        zLong = If(IsNull(BICARSA.Data1(BLHINV).Recordset.Fields("
INV_DATE")), Date, BICARSA.Data1(BLHINV).Recordset.Fields("INV_DATE"))
        TB_FIELD(xINV_DATE).Text = DateToStr(zLong)
        TB_FIELD(xSO_NO).Text = StoreField(BLHINV, "SO_NO")
        zLong = If(IsNull(BICARSA.Data1(BLHINV).Recordset.Fields("SO_DATE")),
Date, BICARSA.Data1(BLHINV).Recordset.Fields("SO_DATE"))
    End If

```



```

TB_FIELD(xSO_DATE).Text = DateToStr(zLong)
TB_FIELD(xCOND).Text = StoreField(BLHINV, "COND")
If xCOND = 0 Then
    TB_FIELD(xCRTERM).Text = " "
    TB_FIELD(xCRTERM).Enabled = False
Else
    TB_FIELD(xCRTERM).Text = StoreField(BLHINV, "CR_DAY")
End If
zLong = IIf(IsNull(BICARSA.Data1(BLHINV).Recordset.Fields("
DUE_DATE")), Date, BICARSA.Data1(BLHINV).Recordset.Fields("DUE_DATE"))
TB_FIELD(xDUE_DATE).Text = DateToStr(zLong)
TB_FIELD(xCUST_NO).Text = StoreField(BLHINV, "CUST_NO")
TB_FIELD(xCUST_NAME).Text = StoreField(BLHINV, "CUST_NAME")
TB_FIELD(xAddr1).Text = StoreField(BLHINV, "ADDR1")
TB_FIELD(xAddr2).Text = StoreField(BLHINV, "ADDR2")
TB_FIELD(xAddr3).Text = StoreField(BLHINV, "ADDR3") 'bbb
TB_FIELD(xTEL).Text = StoreField(BLHINV, "TEL")
TB_FIELD(xFAX).Text = StoreField(BLHINV, "FAX")
TB_FIELD(xSALE_NO).Text = StoreField(BLHINV, "SALE_NO")
zStr = TB_FIELD(xSALE_NO).Text
End If
Call BltStore
Grid2.Row = 0
Grid2.Col = 1
Grid2.Text = StoreField(BLHINV, "DISC_RATE")
Grid2.Text = StoreField(BLHINV, "DiscCond")
Grid2.Col = 2
Grid2.Text = StoreField(BLHINV, "DISC_AMT")
Grid2.Row = 2
Grid2.Col = 1
Grid2.Text = StoreField(BLHINV, "VAT_RATE")
Grid2.Col = 2
Grid2.Text = StoreField(BLHINV, "VAT_AMT")
Grid2.Row = 3
Grid2.Col = 2
Grid2.Text = StoreField(BLHINV, "AMT")
End Sub

Private Sub Form_Activate()
    Dim i As Integer
    Me.WindowState = 2
    MaxField = 15 ' Qty get filed of Heading part
    VatType = Val(SeekString(COMPANY, "COMP_NO", "VAT_TYPE", ""))
    Grid1.Col = 0
    For i = 1 To Grid1.Rows - 1
        ' Grid1.Row = i
        ' Grid1.Text = i
    Next i
    For i = 1 To Grid1.Cols - 1
        Grid1.FixedAlignment(i) = 2
    Next i
    Grid1.ColWidth(0) = 1900
    Grid1.FixedAlignment(0) = 0

    Grid1.ColAlignment(4) = 1 'Right justify
    Grid1.ColAlignment(5) = 1
    Grid1.ColAlignment(6) = 1
    Grid1.ColAlignment(7) = 1
    Grid1.ColAlignment(8) = 1

```

Grid2.ColAlignment(1) = 1
Grid2.ColAlignment(2) = 1

Grid1.Col = 0
Grid1.Row = 0
Grid1.Text = "Goods Name" 'DESC

Grid1.ColWidth(1) = 1100 'set Property each column

Grid1.Col = 1

Grid1.Row = 0

Grid1.Text = "Goods No." 'Part_no

Grid1.ColWidth(2) = 500

Grid1.Col = 2

Grid1.Row = 0

Grid1.Text = "Unit No." 'UNIT NO

Grid1.ColWidth(3) = 500

Grid1.Col = 3

Grid1.Row = 0

Grid1.Text = "Loc. No." 'Location

Grid1.ColWidth(4) = 1000

Grid1.Col = 4

Grid1.Row = 0

Grid1.Text = "Unit 1" 'Qty

Grid1.ColWidth(5) = 1000

Grid1.Col = 5

Grid1.Row = 0

Grid1.Text = "Unit 2" 'Qty

Grid1.ColWidth(6) = 1500

Grid1.Col = 6

Grid1.Row = 0

Grid1.Text = "Price" 'Price

Grid1.ColWidth(7) = 1000

Grid1.Col = 7

Grid1.Row = 0

Grid1.Text = "Discount" 'Discount

Grid1.ColWidth(8) = 1500

Grid1.Col = 8

Grid1.Row = 0

Grid1.Text = "Amount" 'Amount

Grid2.ColWidth(0) = 1500

Grid2.Col = 0

Grid2.Row = 0

Grid2.Text = "Discount Rate" 'Discount

Grid2.ColWidth(1) = 1000

Grid2.Col = 0

Grid2.Row = 1

Grid2.Text = "Total - Discount" 'total after discount

Grid2.ColWidth(2) = 1500

Grid2.Col = 0

Grid2.Row = 2

Grid2.Text = "Tax Rate" 'Vat

Grid2.Col = 0

Grid2.Row = 3

Grid2.Text = "Total" 'Total After VAT

BT3D_CANCEL.Enabled = False

BT3D_SAVE.Enabled = False

BT3D_LOOK.Enabled = False

BT3D_HELP.Enabled = True

```

BT3D_PRINT.Enabled = True
BT3D_ADD.Enabled = True
BT3D_EDIT.Enabled = True
BT3D_DEL.Enabled = True
BT3D_EXIT.Enabled = True
TB_FIELD(xINV_NO).MaxLength = 10
TB_FIELD(xSO_NO).MaxLength = 10
TB_FIELD(xCOND).MaxLength = 1
TB_FIELD(xCRTERM).MaxLength = 3
TB_FIELD(xSALE_NO).MaxLength = 10
TB_FIELD(xCUST_NO).MaxLength = 10 'customer code
TB_FIELD(xCUST_NAME).MaxLength = 60 'bbb 50 -> 60
'Customer Need
TB_FIELD(xAddr1).MaxLength = 64
TB_FIELD(xAddr2).MaxLength = 64
TB_FIELD(xAddr3).MaxLength = 64 'bbb
TB_FIELD(xTEL).MaxLength = 30
TB_FIELD(xFAX).MaxLength = 15
End Sub

Private Sub Form_Load()
    MaxField = 14 ' Qty get filed of Heading part
    BT3D_DELROW.Enabled = False
    Call SwitchGet(False)
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    BICARSA.MessBar.Caption = " "
End Sub

Private Sub Form_Unload(Cancel As Integer)
    If from_menu = 1 Then BICARSA.Picture1.Visible = True 'Billy
End Sub

Private Sub FuncCust()
    Dim Msg As String ' Declare variables.
    Criteria = "CUST_NO = '" + TB_FIELD(xCUST_NO).Text + "'" ' Create the
criteria.
    BICARSA.Data1(ARMCUST).Recordset.FindFirst Criteria ' Make record
current.
    If BICARSA.Data1(ARMCUST).Recordset.NoMatch Then ' Found
        Msg = "Not Found"
        'Msg = Msg & " press any key"
        Call ErrBox(Msg)
        ErrorFlag = True
    End If
End Sub

Private Sub FuncFlag()
    Dim Msg As Integer
    Select Case ACDFLAG
        Case Add
            Call AddRec
        Case Change
            Call EditRec
        Case Del
            Call DelRec
    End Select
End Sub

```

```

Private Sub Grid1_Click()
' If Grid1.Col <> 2 Then
    Text1.Text = Grid1.Text
    ShowTextBox
' End If
End Sub

Private Sub Grid1_GotFocus()
    BT3D_DELROW.Enabled = True
    LookField = MaxField - 1 + Grid1.Col ' set position can look up
    Grid2.Col = 1
    DispMessage (MaxField - 1 + Grid1.Col)
End Sub

Private Sub OEtStore()
    Dim i As Integer ' Declare variables.
    Dim zStr As Double
    Dim zQtyOrder, zQtyDeli, zQty, zPrice, zDISC As Double
    Criteria = "SO_NO=" + TB_FIELD(xSO_NO).Text + "" ' Create the criteria.
    BICARSA.Data1(OEDSO).Recordset.FindFirst Criteria ' Make record current.
    i = 1
    If Not BICARSA.Data1(OEDSO).Recordset.NoMatch Then ' Found
        Do While StoreField(OEDSO, "SO_NO") = TB_FIELD(xSO_NO).Text
            Grid1.Row = i
            Grid1.Col = 1
            Grid1.Text = StoreField(OEDSO, "PART_NO")
            DispProduct (Grid1.Text)
            If Not ErrorFlag Then
                Grid1.Col = 0
                Grid1.Text = StoreField(ICMPROD, "PROD_NAME")
            Else
                ErrorFlag = False 'err
            End If
            Grid1.Col = 2
            If Not IsNull(StoreField(OEDSO, "UNIT_NO")) Then
                Grid1.Text = StoreField(OEDSO, "UNIT_NO")
                'zStr1$ = Grid1.Text
                zStr1$ = StoreField(ICMPROD, "UNIT_NO")
                zRate# = Val(SeekString(ICMUNIT, "UNIT_NO", "RATE", zStr1$))
            End If
            Grid1.Col = 3
            If Not IsNull(StoreField(OEDSO, "LOC_NO")) Then
                Grid1.Text = StoreField(OEDSO, "LOC_NO")
            End If
            Grid1.Col = 4
            If IsNull(BICARSA.Data1(OEDSO).Recordset.Fields("QTY_ORDER"))
Then
                zQtyOrder = 0
            Else
                zQtyOrder = BICARSA.Data1(OEDSO).Recordset.Fields("
QTY_ORDER")
            End If
            If IsNull(BICARSA.Data1(OEDSO).Recordset.Fields("QTY_DELI")) Then
                zQtyDeli = 0
            Else
                zQtyDeli = BICARSA.Data1(OEDSO).Recordset.Fields("QTY_Deli")
            End If
        End While
    End If

```



```

zQty = zQtyOrder - zQtyDeli
zMajor# = Fix(zQty / zRate#)
zMinor# = zQty - (zMajor# * zRate#)
zStr = Trim(Str(zMajor#)) & "." & Trim(Str(zMinor#))
Grid1.Text = Format(zMajor#, "##0.00")      'Format(zStr, "##0.00")
Grid1.Col = 5
Grid1.Text = Format(zMinor#, "##0.00")
zNum# = zQty

```

```

'If zQty <> 0 Then
'  Grid1.Text = Format(zQty, "##0.00")
'End If
Grid1.Col = 6
If IsNull(BICARSA.Data1(OEDSO).Recordset.Fields("PRICE")) Then
  zPrice = 0
Else
  zPrice = BICARSA.Data1(OEDSO).Recordset.Fields("PRICE")
  zNum# = zNum# * zPrice
End If
If zPrice <> 0 Then
  Grid1.Text = Format(zPrice, "##0.00")
End If
Grid1.Col = 7
If IsNull(BICARSA.Data1(OEDSO).Recordset.Fields("DISC")) Then
  zDISC = 0
Else
  zDISC = BICARSA.Data1(OEDSO).Recordset.Fields("DISC")
End If
If zDISC <> 0 Then
  Grid1.Text = Format(zDISC, "##0.00")
End If
Grid1.Col = 8
'zStr = (zQty * zPrice) - zDISC
zStr = (zMajor# * zPrice) + (zMinor# * zPrice / zRate#)
If zStr <> 0 Then
  Grid1.Text = Format(zStr, "##0.00")
End If
'BICARSA.Data1(OEDSO).Recordset.MoveNext
'If BICARSA.Data1(OEDSO).Recordset.EOF Then
'  Exit Do
'End If
BICARSA.Data1(OEDSO).Recordset.FindNext Criteria
If BICARSA.Data1(OEDSO).Recordset.NoMatch Then 'Not Found
  Exit Do
End If

```

```

i = i + 1
Loop
Grid2.Col = 1
Call DispTotal
End If
End Sub

```

```

Private Sub Online_AR(zDOC_NO As String, zCDATE As Long, zDUE_DATE As
Long, zCUST_NO As String, zSALE_NO As String, zAMT As Double)
  Dim Criteria As String, zDouble As Double
  Dim zDOC_DATE As String
  Dim zMONTH As String
  Dim zVat As Double

```

```

Criteria = "DOC_NO=" + TB_FIELD(xINV_NO).Text + "" ' Create the criteria.
'Update ARTSA Overdue Customer record
On Error Resume Next
BICARSA.Data1(ARTSA).Recordset.FindFirst Criteria ' Make record current.
If Not BICARSA.Data1(ARTSA).Recordset.NoMatch Then ' Found
    BICARSA.Data1(ARTSA).Recordset.Edit
Else
    BICARSA.Data1(ARTSA).Recordset.AddNew
    BICARSA.Data1(ARTSA).Recordset.Fields("BAL") = zAMT
End If
BICARSA.Data1(ARTSA).Recordset.Fields("DOC_NO") = zDOC_NO
BICARSA.Data1(ARTSA).Recordset.Fields("CDATE") = zCDATE
BICARSA.Data1(ARTSA).Recordset.Fields("DUE_DATE") = zDUE_DATE
BICARSA.Data1(ARTSA).Recordset.Fields("CUST_NO") = zCUST_NO
BICARSA.Data1(ARTSA).Recordset.Fields("SALE_NO") = zSALE_NO
BICARSA.Data1(ARTSA).Recordset.Fields("AMT") = zAMT
'If IsNull(BICARSA.Data1(ARTSA).Recordset.Fields("BAL")) Then
'    BICARSA.Data1(ARTSA).Recordset.Fields("BAL") = zAMT
'End If
BICARSA.Data1(ARTSA).Recordset.Update
If Err > 0 Then
    Msg = "Error " & Error
    Call ErrBox(Msg)
End If
'Update ARTMONTH individual account receivable file
On Error Resume Next
BICARSA.Data1(ARTMONTH).Recordset.FindFirst Criteria ' Make record
current.
If Not BICARSA.Data1(ARTMONTH).Recordset.NoMatch Then ' Found
    BICARSA.Data1(ARTMONTH).Recordset.Edit
Else
    BICARSA.Data1(ARTMONTH).Recordset.AddNew
End If
BICARSA.Data1(ARTMONTH).Recordset.Fields("DOC_NO") = zDOC_NO
BICARSA.Data1(ARTMONTH).Recordset.Fields("CDATE") = zCDATE
BICARSA.Data1(ARTMONTH).Recordset.Fields("CUST_NO") = zCUST_NO
BICARSA.Data1(ARTMONTH).Recordset.Fields("AMT") = zAMT
BICARSA.Data1(ARTMONTH).Recordset.Fields("REFNO_F") = "1"
BICARSA.Data1(ARTMONTH).Recordset.Update
If Err > 0 Then
    Msg = "Error " & Error
    Call ErrBox(Msg)
End If
Criteria = "CUST_NO=" + zCUST_NO + "" ' Create the criteria.
'Update ARMCUST Account Receivable Credit File
BICARSA.Data1(ARMCUST).Recordset.FindFirst Criteria ' Make record
current.
If Not BICARSA.Data1(ARMCUST).Recordset.NoMatch Then ' Found
    BICARSA.Data1(ARMCUST).Recordset.Edit
    BICARSA.Data1(ARMCUST).Recordset.Fields("LSALEDATE") = zCDATE
    BICARSA.Data1(ARMCUST).Recordset.Fields("LSALEAMT") = zAMT
    zDouble = BICARSA.Data1(ARMCUST).Recordset.Fields("LIABIL")
    BICARSA.Data1(ARMCUST).Recordset.Fields("LIABIL") = zDouble + zAMT
    BICARSA.Data1(ARMCUST).Recordset.Update
If Err > 0 Then
    Msg = "Error " & Error
    Call ErrBox(Msg)
End If
End If
End If

```

End Sub

```
Private Sub Online_OE(zSO_NO As String, zPart_no As String, zQty As Double)
    Dim zDouble As Double
    Criteria = "SO_NO=" + zSO_NO + " and PART_NO=" + zPart_no + "" ' Create
the criteria.
    BICARSA.Data1(OEDSO).Recordset.FindFirst Criteria ' Make record current.
    If Not BICARSA.Data1(OEDSO).Recordset.NoMatch Then ' Found
        BICARSA.Data1(OEDSO).Recordset.Edit
        zDouble = IIf(IsNull(BICARSA.Data1(OEDSO).Recordset.Fields("
QTY_DELI")), 0, BICARSA.Data1(OEDSO).Recordset.Fields("QTY_DELI"))
        BICARSA.Data1(OEDSO).Recordset.Fields("QTY_DELI") = zDouble + zQty
        BICARSA.Data1(OEDSO).Recordset.Update
    If Err > 0 Then
        Msg = "Error " & Error
        Call ErrBox(Msg)
    End If
End If
End Sub
```

```
Private Function SumDetailAmt(zINV_NO As String) As Double
    Dim TempDB As Database
    Dim TempSN As Snapshot
    Dim zSum As Double
    zSum = 0
    Set TempDB = OpenDatabase(DataDir & "BICARSA.MDB")
    Set TempSN = TempDB.CreateSnapshot("BLDINV")
    TempSN.Filter = "INV_NO=" & zINV_NO & ""
    Set TempSN = TempSN.CreateSnapshot()
    TempSN.MoveFirst
    Do While Not TempSN.EOF
        zSum = zSum + TempSN!AMT
        TempSN.MoveNext
    Loop
    SumDetailAmt = zSum
    TempSN.Close
    TempDB.Close
End Function
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
    Dim OldCol As Integer
    Dim zStr As String
    Dim zPart_no As String, zCUST_NO As String
    Dim zNum As Double
    Dim zDiscount As Double
    Select Case KeyAscii
        Case 13
            Grid1.Text = Text1.Text
            Select Case Grid1.Col
                Case 1 'PART NO
                    Grid1.Text = Text1.Text
                    zPart_no = Text1.Text
                    DispProduct (Grid1.Text)
                    If Not ErrorFlag Then
                        zCUST_NO = TB_FIELD(xCUST_NO).Text
                        zPrice = PriceLevel(zPart_no, zCUST_NO)
                        Grid1.Col = 6
                        Grid1.Text = Format(zPrice, "##0.00")
                        Grid1.Col = 0
                    End If
                End Select
            End Select
    End Select
```

```

Grid1.Text = StoreField(ICMPROD, "PROD_NAME")
Grid1.Col = 1
Grid1.Col = 2 'UNIT_NO 06/01/39
Grid1.Text = StoreField(ICMPROD, "UNIT_NO") '06/01/39
End If
Case 2
' Check Error don't want return value
' zstr use for syntax
zStr = Grid1.Text
zRate# = Val(SeekString(ICMUNIT, "UNIT_NO", "RATE", zStr))
Case 3
' Check Error don't want return value
' zstr use for syntax
zStr = Grid1.Text
zStr = SeekString(ICMLOC, "LOC_NO", "LOC_NAME", zStr)
Case 4, 5, 6, 7
OldCol = Grid1.Col
Grid1.Col = 1 'Part_no
zPartno$ = Grid1.Text
Grid1.Col = 2
zStr = Grid1.Text
zRate# = Val(SeekString(ICMUNIT, "UNIT_NO", "RATE", zStr))
Grid1.Col = 3 'From Location
zLocNo$ = Grid1.Text
Grid1.Col = 4 'QTY
zMajor# = Val(Grid1.Text)
Grid1.Col = 5
zMinor# = Val(Grid1.Text)
'zMinor# = zNum - zMajor#
zQty# = zMajor# * zRate# + zMinor#
'Un-recorded stock for cut-off
If OldCol = 4 Then
If Not CheckBal(zPartno$, zLocNo$, zQty#) Then
Grid1.SetFocus
Exit Sub
End If
End If
'Grid1.Col = 4 'QTY
'zNum = Val(Grid1.Text)
Grid1.Col = 6 'Price
xPrice# = Val(Grid1.Text)
'zAMT# = zNum * xPrice#
Grid1.Col = 7 'Discount
zAMT# = zQty# * (xPrice# / zRate#)
zStr = Grid1.Text
zDiscount = MultiDiscount(zAMT#, zStr) 'zNum * Val(Grid1.Text) / 100
'Term ájéµéÍ§jÖÃãËéà»ç¹ %
'zNum = zNum - Val(Grid1.Text)
'zNum = zNum - zDiscount
If zMinor# <> 0 And zRate# <> 0 Then
zAMT# = (zMajor# * xPrice#) + (xPrice# / zRate# * zMinor#) - zDiscount
Else
zAMT# = (zMajor# * xPrice#) - zDiscount
End If

Grid1.Col = 8
Grid1.Text = Format(zAMT#, "##0.00")
Call DispTotal
Grid1.Col = OldCol

```



```

(xAddr1).Text, 50)
    BICARSA.Data1(BLHINV).Recordset.Fields("ADDR2") = Left(TB_FIELD
(xAddr2).Text, 50)
    BICARSA.Data1(BLHINV).Recordset.Fields("ADDR3") = Left(TB_FIELD
(xAddr3).Text, 50) 'bbb
    BICARSA.Data1(BLHINV).Recordset.Fields("TEL") = TB_FIELD(xTEL).Text
    BICARSA.Data1(BLHINV).Recordset.Fields("FAX") = TB_FIELD(xFAX).Text
    Grid2.Row = 0
    Grid2.Col = 1
    'BICARSA.Data1(BLHINV).Recordset.Fields("DISC_RATE") = Val(Grid2.Text)
    BICARSA.Data1(BLHINV).Recordset.Fields("DiscCond") = Grid2.Text
    Grid2.Col = 2
    BICARSA.Data1(BLHINV).Recordset.Fields("DISC_AMT") = Val(Grid2.Text)
    Grid2.Row = 2
    Grid2.Col = 1
    BICARSA.Data1(BLHINV).Recordset.Fields("VAT_RATE") = Val(Grid2.Text)
    Grid2.Col = 2
    BICARSA.Data1(BLHINV).Recordset.Fields("VAT_AMT") = Val(Grid2.Text)
    Grid2.Row = 3 'err
    Grid2.Col = 2
    BICARSA.Data1(BLHINV).Recordset.Fields("AMT") = Val(Grid2.Text)
    BICARSA.Data1(BLHINV).Recordset.Update
    If Err <> 0 Then
        Msg = "Error " & Error
        Call ErrBox(Msg)
    End If
    'Call DelOldTran ' Delete old transaction
    i = 1
    Do While i < Grid1.Rows
        Grid1.Row = i
        Grid1.Col = 1
        zStr = Grid1.Text
        If Len(Trim(zStr)) > 0 Then
            On Error Resume Next 'err
            BICARSA.Data1(BLDINV).Recordset.AddNew
            BICARSA.Data1(BLDINV).Recordset.Fields("INV_NO") = TB_FIELD
(xINV_NO).Text
            BICARSA.Data1(BLDINV).Recordset.Fields("PART_NO") = Grid1.Text
            Grid1.Col = 2
            BICARSA.Data1(BLDINV).Recordset.Fields("UNIT_NO") = Grid1.Text
            zStr = Grid1.Text
            zRate# = Val(SeekString(ICMUNIT, "UNIT_NO", "RATE", zStr))
            Grid1.Col = 3
            BICARSA.Data1(BLDINV).Recordset.Fields("LOC_NO") = Grid1.Text
            Grid1.Col = 4
            zMajor# = Val(Grid1.Text)
            Grid1.Col = 5
            zMinor# = Val(Grid1.Text)
            'zMajor# = Fix(zNUM)
            'zMinor# = zNum - zMajor#
            zQty# = zMajor# * zRate# + zMinor#
            BICARSA.Data1(BLDINV).Recordset.Fields("QTY") = zQty#
            *****
            Grid1.Col = 6
            BICARSA.Data1(BLDINV).Recordset.Fields("PRICE") = Val(Grid1.Text)
            zNum = zQty# * Val(Grid1.Text) / zRate#
            Grid1.Col = 7
            zStr = Grid1.Text

```

```

BICARSA.Data1(BLDINV).Recordset.Fields("DiscCond") = zStr
    BICARSA.Data1(BLDINV).Recordset.Fields("DISC") = MultiDiscount
(zNum, zStr)
    Grid1.Col = 8
    BICARSA.Data1(BLDINV).Recordset.Fields("AMT") = Val(Grid1.Text)
    BICARSA.Data1(BLDINV).Recordset.Update
If Err <> 0 Then
    Msg = "Error " & Error
    Call ErrBox(Msg)
End If
End If
    i = i + 1
Loop
End Sub

```

