



NEAR EAST UNIVERSITY

FACULTY OF ECONOMICS AND  
ADMINISTRATIVE SCIENCES

DEPARTMENT OF COMPUTER  
INFORMATION SYSTEMS

2005/2006 FALL TERM

CIS 400

(Graduation project)

COMPUTER STORE AUTOMATION SYSTEM

Submitted To : Dept.Of Computer Information Systems

Submitted by : ALI FUAT TATLIMAVUK  
20020389

Lefkosa,

2005





## I. ACKNOWLEDGEMENT

First and foremost I would like to thank very much Assist.Prof. Yalçın AKÇALI who is chairman of Computer Information Systems department and Miss Nadire Çavuş who never left his support and always encouraged me during my study.

I would also like to thanks all instructors of department of Computer Information System for their support.

And finally, I would like to thank very much to my family for providing me with financial support during all my educational life and for their psychological support in all parts of my life by providing me comfort.

## II. ABSTRACT

The Computer Store Automation System is going to prepare to solve problem that company processes which is made by hand. This system is going to make company processes fast, easily and more reliable.

The program provides manage and take hold of business transactions' record, personnel records, training records, stock records, customer records, at a small or medium computer store. A scheduled user manual prepared for helping the users to select an suitable action.

I select this project because, if a firm sold technological parts especially computer parts that company need a software to use on their jobs, when they work manually works take a lot of time and needed more workers. Therefore I decide to make a program to satisfy their software needs.

Computer store automation system has different alternative user groups, these groups provide three choices of access availabilities, admins have more access availabilities that is whole program, accountants have restrictions on personnel area , users can access only products related menus.

Computer store automation system aims to help company manager, company personnel and customers, the system provides easy, quick and more reliable process on company works.

### III. INTRODUCTION

The companies were doing their processes manually, such as hold record of customers and suppliers, stock controlling, accounting, and other business transactions. But recently IT (Information Technology) started to help companies or firms. Then it has been very popular because it is faster, cheaper than manually and so easy work with IT.

It is necessary to companies work with computer programs in their work to more valuable present and more efficiently working. Managers, workers and customers feel their self they are really in a technological company when they work or being in a program like this.

At this point Computer Store Automation System will provide easiness and quickness of company processes that are customer records, purchased and sold products, stock controlling, accounting records, personnel records. Also system has a authorization steps to determine and control the access levels, to use this feature every user have different authorization levels that given by managers with username and password.

I hope this system will be useful to all personnel that in the company and all customers.



This program prepared by using Borland DELPHI programming language and Database was structured by using PARADOX that connected with program.

This system designed for using at Windows Operating System and Turkish date format, also have a installment package to easily setup.

The Computer store automation system have 6 (six) main process;

## **ENTRANCE**

This process to operate the program, get user profile that means now who using this program, purpose of this process is determine the authorization level if Admin level operate the program all processes are available for using, if Accountant level operate User Accounts Settings, Personnel Records and Personnel Attendance processes are disable, other processes are available for using, if User level operate the program all Account Processes(Customer Accounts, Supplier Accounts, Personnel Accounts), Personnel Records and Personnel Attendance processes are disable , other processes are available for using.

### **1- CUSTOMER**

At this process customer information such as name, surname, telephone will be taken from the customer and save to Database.

This process has sub processes;

- **Add customer** : in this sub process users can entry new customers to the database

- **Search / update** : in this sub process users can search an old-recorded customer and if necessary update their information
- **Delete customer** : in this process users can delete recorded customers if there is no need to information about the selected customers

## 2- STOCK PROCESS

All processes about products and suppliers will be made in this processes

This process has sub processes;

- **Supplier process** : this process provide recording supplier information, update them when necessary, and may delete records if all contacts are give out.
- **Purchase process** : all products that are purchased, recorded with suppliers information and total amount of invoices computed then saved in this process.
- **Selling process** : when a customer arrive the company and ask some products this process is using to perform it, process shows the available products in stock, give a recommended price for selling that is entering system on purchasing and give a chance to discount on recommended price, when selling to be completed if there is an installment request compute installment amounts and determine the payment plan and finally print all invoices and payment plans.

### 3- ACCOUNT PROCESS

Accounting jobs are done by using this process

This process has sub processes ;

- **Customer Accounts** : this process provide the information from DB who have debt to company and customer payments recorded at this process.
- **Supplier Accounts** : firstly this process using for check supplier statements if there will be an error in statements then recorded the payments that made to suppliers.
- **Personnel Accounts** : at this process compute the personnel salary on based of working days, and recorded advance payments & salary payments.

### 4- PERSONNEL PROCESS

All records about personnel take at this process

This process has sub processes

- **Personnel records** : here identity information about personnel are taken such as Ssn number, name, surname, photo.
- **Personnel Attendance** : At this process daily personnel attends are recorded and it used at salary evaluation, also permits are given from this process to personnel.

## 5- REPORTS PROCESS

This process provides various reports to managers and other users

This process has sub processes;

- **Customer Reports** : This section involves reports about customers, give satisfy information to managers.

Reports are ; All customers, customers who have debt, customers who buying lots of products, customers who delay debts, general customer profile of company.

- **Supplier Reports** : This section involves reports about suppliers, give necessary information to workers and managers.

Reports are ; supplier list, which supplier supply which products , which suppliers are where.

- **Stock Reports** : This section involves stock reports, give detail information about products in stock and general stock situation to managers and workers.

Reports are ; available products, under critical point products, near critical point products, most wanted products, never sold products , more profitable products.

- **Personnel Reports** : This section involves general reports about personnel.

Reports are ; personnel list, personnel profile, attendance of each personnel.

- **Account Reports :** This section involves general accounting reports , give critical and desired information to managers who make decision in company.

Reports are ; sales between two date, purchases between two date, daily and recent time supplier payments, general and daily receivables information, salary of personnel, expense – revenue of months and days.

## **6- SETTINGS PROCESS**

Products descriptions and other specific settings included in this process

This process has sub process ;

- **Product description :** products name and brand entering by using this process, so its names and brands appear as a normalized form during make stock records also during selling products, when created a report there wont be a data loss because of standard form of names and brands.
- **Change Password :** this process give a chance to user to change their user name and password.

**User Settings :** this process prepared for Admin s it will provide the change user names, passwords and authorization levels of users.



## **IV. DESIGN OF THE SYSTEM**

### **EXPLANATION**

**The minimum requirements of the system are as follows ;**

- Intel Pentium III processor 800 Mhz
- 256 MB, DDR of RAM
- 20 GB Harddisk
- 14” Monitor (800 x 600 resolution )
- 32x CD-ROM
- Printer
- Microsoft Windows 2000

**The recommended requirements of the system are as follows ;**

- Intel Pentium IV processor 800 Mhz
- 512 MB, DDR of RAM
- 40 GB Harddisk
- 15” Monitor (1024 x 768 resolution )
- 52x CD-ROM
- USP Printer
- Microsoft Windows XP Professional

# **1. PROJECT IDENTIFICATION AND SELECTION**

## **THE AIM OF THE PROJECT**

The Computer Store Automation System aims to help managers and employees who work in a company that sold computer parts, this system provides to employees easy, quick and reliable work area to record daily transactions, also system provides to managers easy, quick and more reliable work area at their jobs like more available records to control and check necessary information, quickly take a report throughout program, more control above their personnel, easily backup company information because all information in digital area.

At existing system products made warranty with a text document that given the customer and it's a big weakness because of customer unconsciousness but at this system all products that sold the company in warranty extension with product Serial number.

# **1. PROJECT IDENTIFICATION AND SELECTION**

## **THE AIM OF THE PROJECT**

The Computer Store Automation System aims to help managers and employees who work in a company that sold computer parts, this system provides to employees easy, quick and reliable work area to record daily transactions, also system provides to managers easy, quick and more reliable work area at their jobs like more available records to control and check necessary information, quickly take a report throughout program, more control above their personnel, easily backup company information because all information in digital area.

At existing system products made warranty with a text document that given the customer and it's a big weakness because of customer unconsciousness but at this system all products that sold the company in warranty extension with product Serial number.

## **THE PROJECT BOUNDARIES :**

1. The existing records that hold at papers are in confused situation so it may takes long time for transfer to new system.
2. The products that sold before build this system are not exactly known so first one or two years there will be an uncertainty at warranty process.
3. The current products in stock are not exactly known , there are no personnel records at paper or any where so before using system needed a time period to hold this records.

## **RESOURCE LIMITS**

This system has a limited budget because it wanted like a trial version of big system, so all resources were not available. This system shows the benefits and weakness to managers if it will be successful the system need a improving with some modules.

## **2. PROJECT INITIATION AND PLANNING**

### **TECHNICAL FEASIBILITY**

The hardware that necessary for system are also to be found in company so it can be easily build and operate the system

### **OPERATIONAL FEASIBILITY**

At operation part, system has a strength and big chance because almost of personnel have computer using knowledge at good level however this system has own logic and algorithm so in a short training period user can learn the system, there will be a training program for user.

### **ECONOMICAL FEASIBILITY**

Actually company afford the project costs and they give a limit to all projects, so all research and development steps were in a limited time and cost.



### **3.ANALYSIS**

#### **THE REQUIREMENT DETERMINATION**

The requirements made by managers are as follows;

- Exact records about personnel attend, evaluation at salary payments
- To provide easy and quick data access
- To provide normalized data form about stock
- To prevent data loss
- To restrict data access build an authorization level to provide system security
- To provide an attention mechanism on unexpected situation

The requirements made by users are as follows;

- To increase speed of operations
- To provide facility on make records
- Easy and quick access on necessary information about supplier and customer
- To provide some automatic process during operation

## **EXISTING SYSTEM CAPABILITIES**

- The computer store works manually, all records hold by hand.
- There is no control system on personnel attends
- Warranty process made by hand and certificate given to customer
- During the selling products availability search by hand in stock
- After a product sold bill was prepared by hand
- Reporting process was very long and tiring because of need to control of all documents
- There is no a standard record about customers
- Receivables and payments recorded a ledger manually

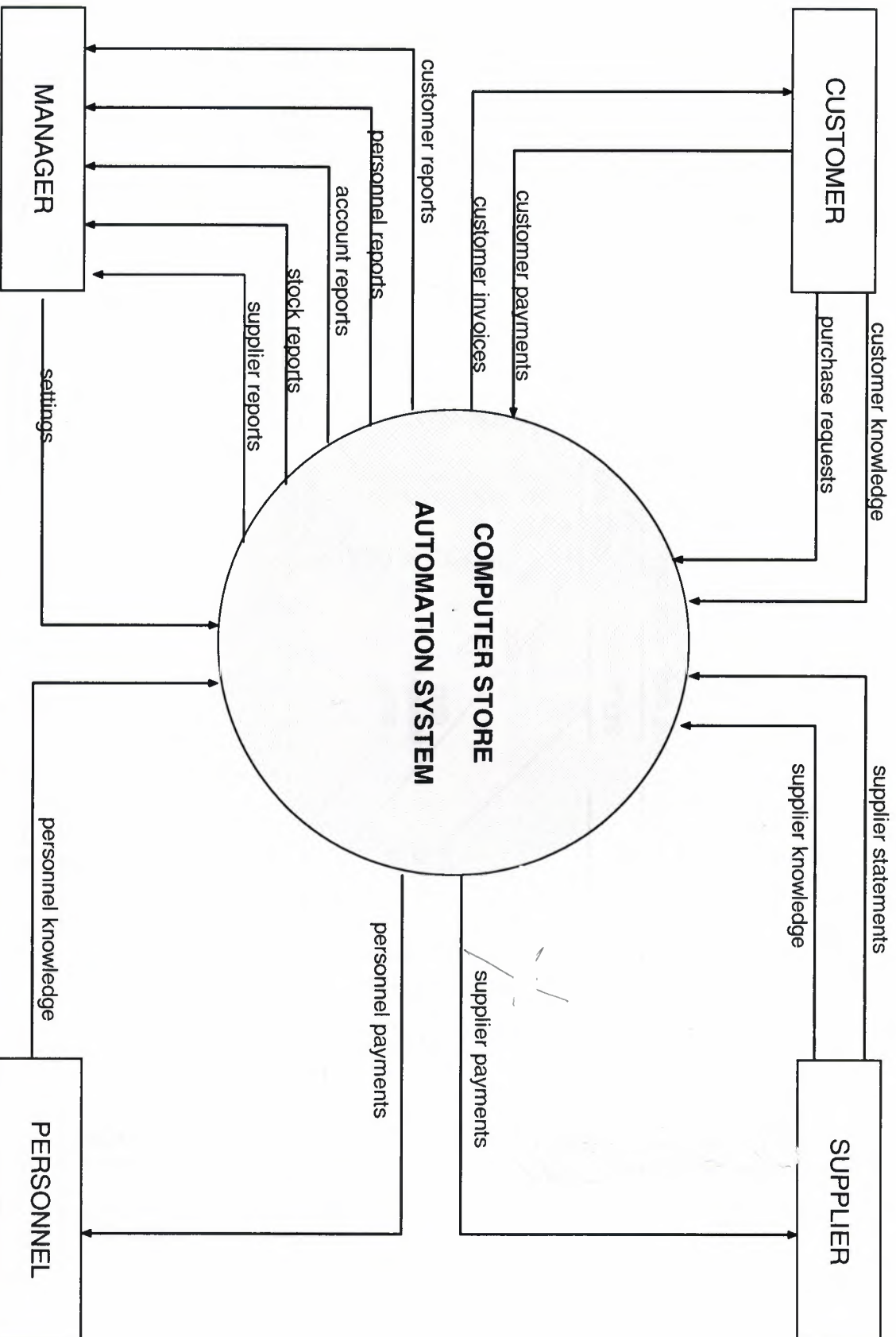
## **NEW SYSTEM'S PROVIDENCES**

- All records hold on the computer
- Personnel attendance recorded daily and giving report when its want, evaluate it at salary payments
- Serial number of product presents the product warranty and detail information
- A normalized form on product names and brands
- Search products which is available or not just by using system
- After selling bill and other acquitting documents prepared by system and printed
- Reports are will taken easily

- All customers & suppliers and their information keep in the database ready to use every time
- Accounting records such as payment dates, receivable dates, personnel salaries keep in database
- Reports can be prepare like a document with using printing process
- There are three user authorizations, they are ADMIN who access every process, ACCOUNTANT and USER s have limited access authorizations.

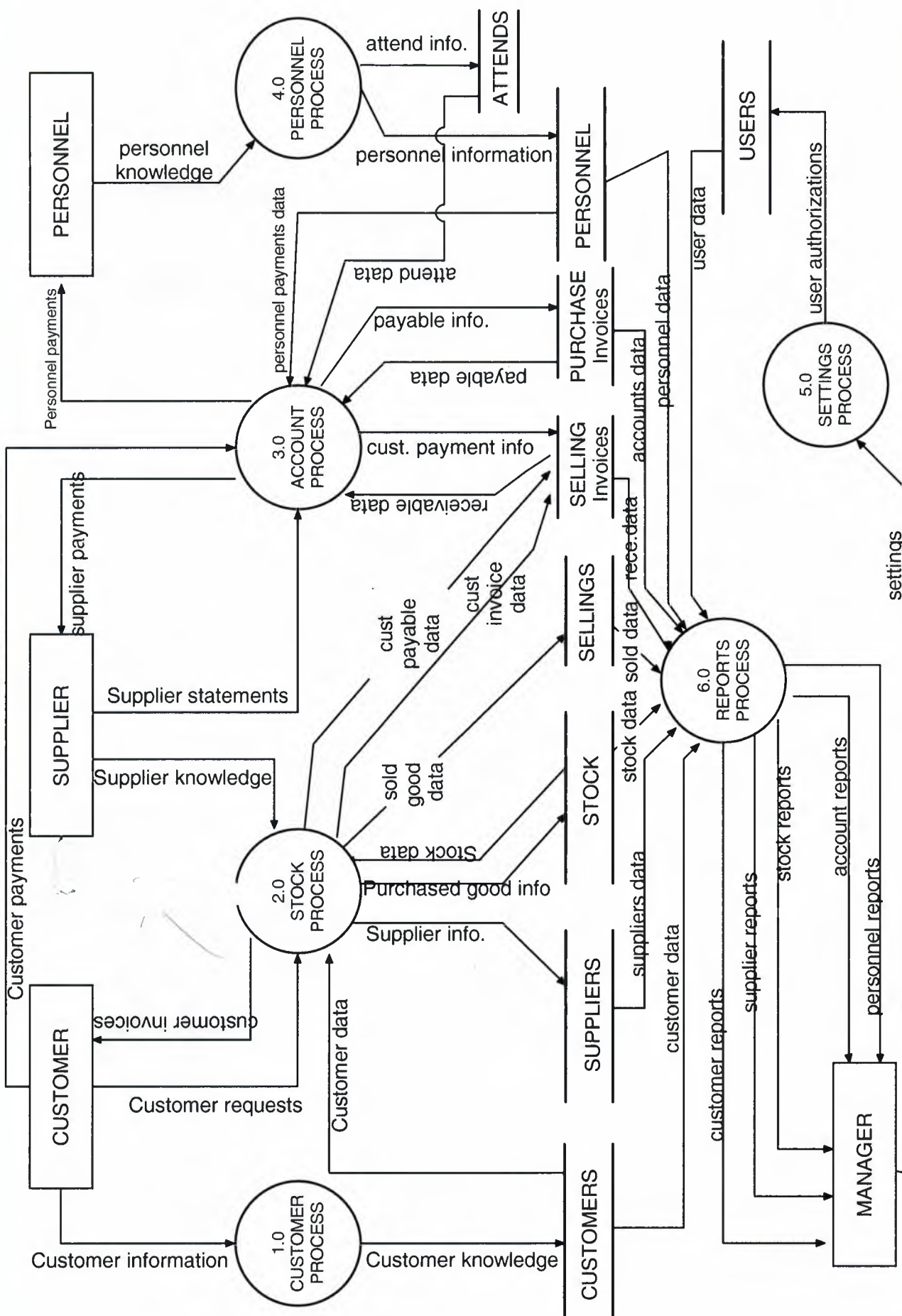
## **DATA FLOW DIAGRAMS**

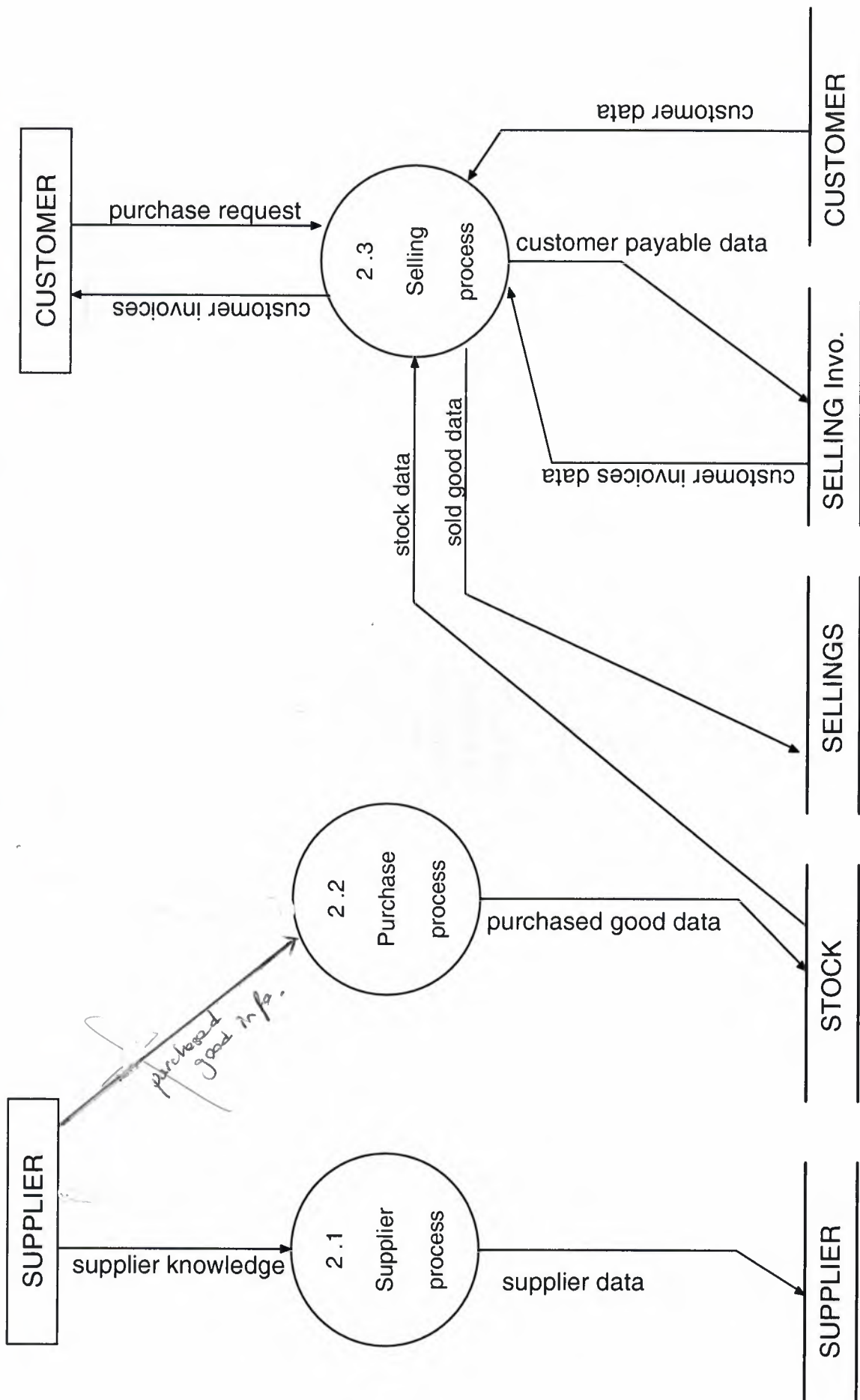
# CONTEXT DATA FLOW DIAGRAM FOR COMPUTER STORE AUTOMATION SYSTEM

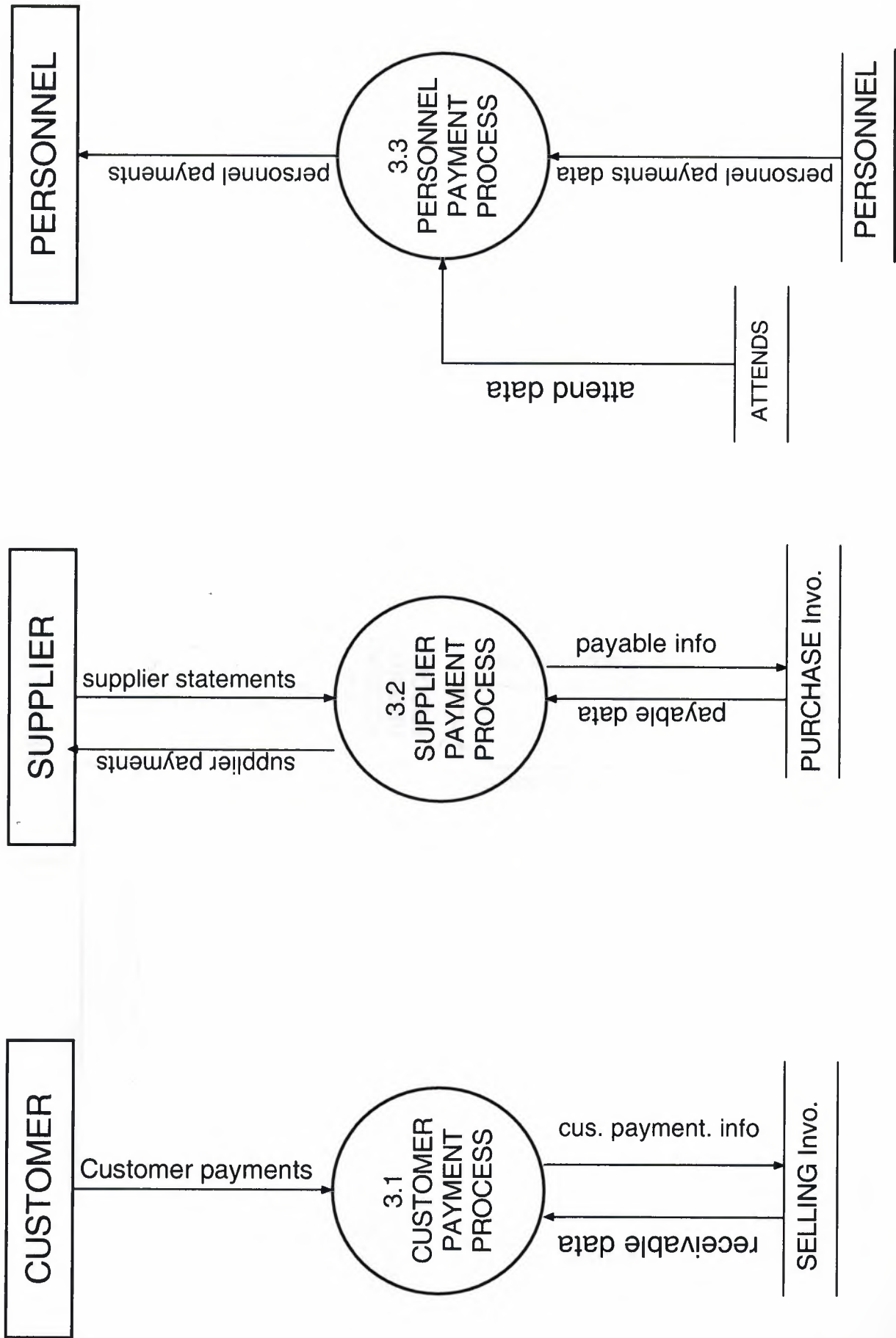


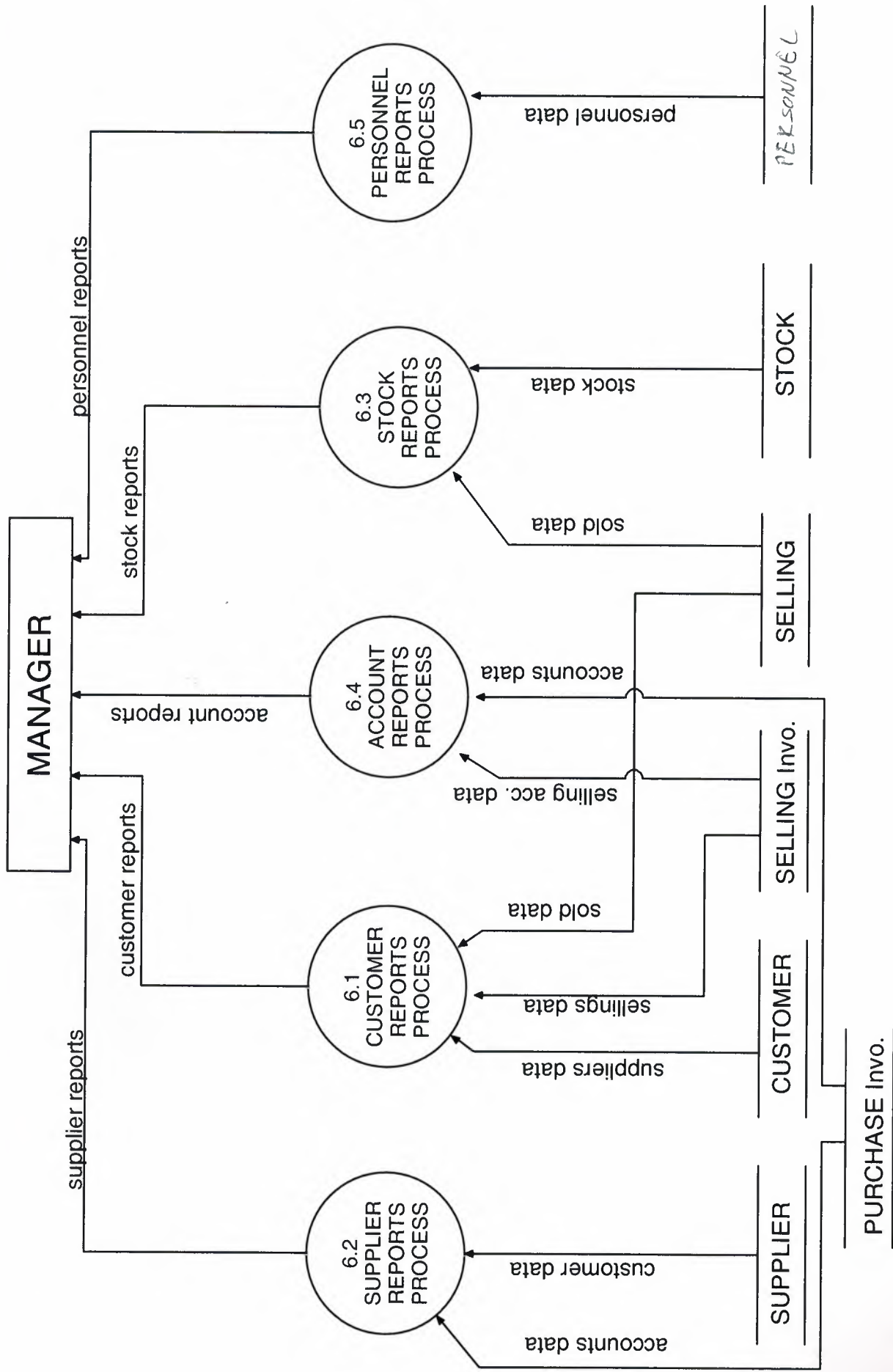


# FOR COMPUTER STORE AUTOMATION SYSTEM







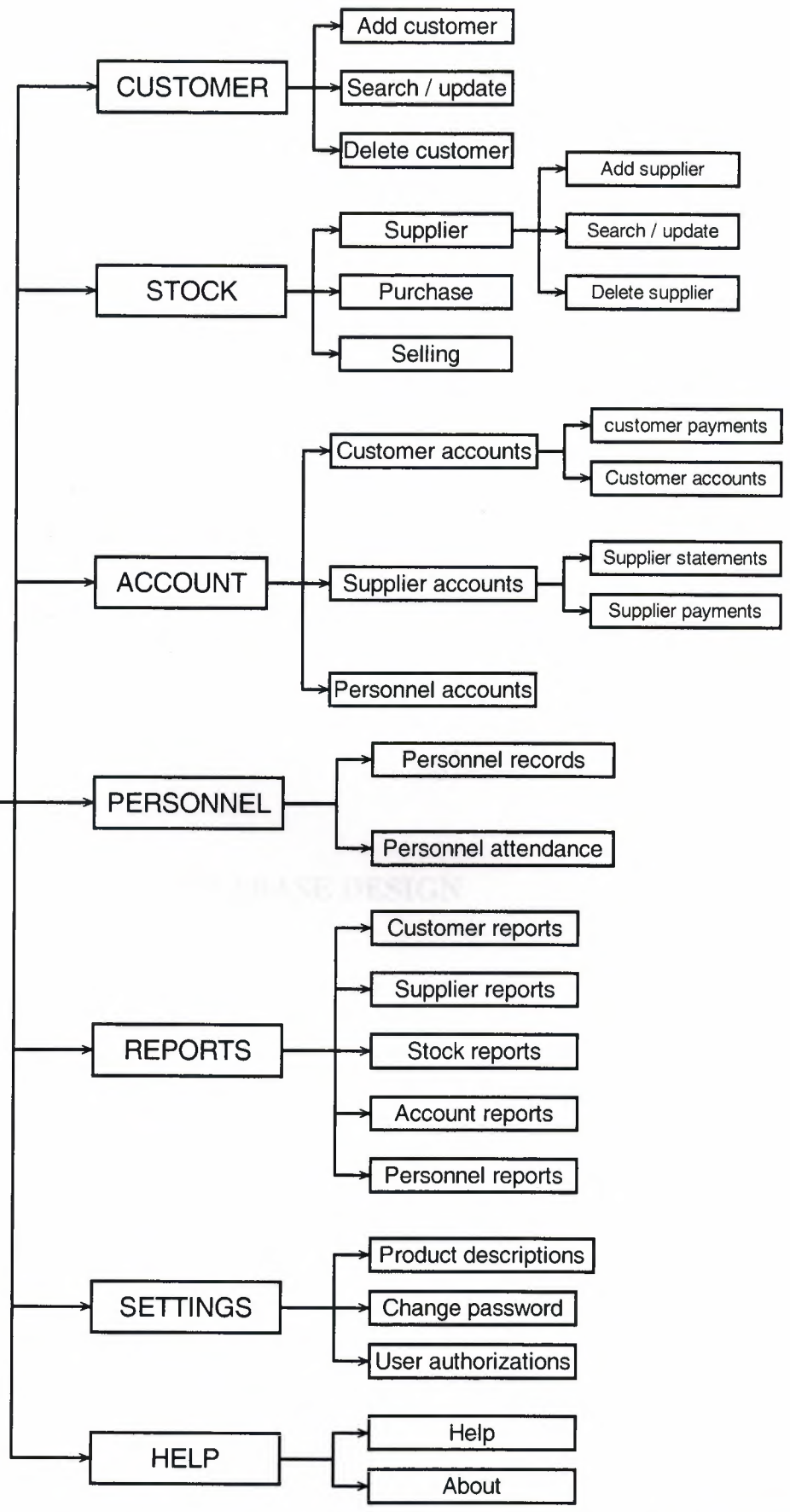


## SYSTEM BLOCK DIAGRAM





COMPUTER STORE AUTOMATION SYSTEM



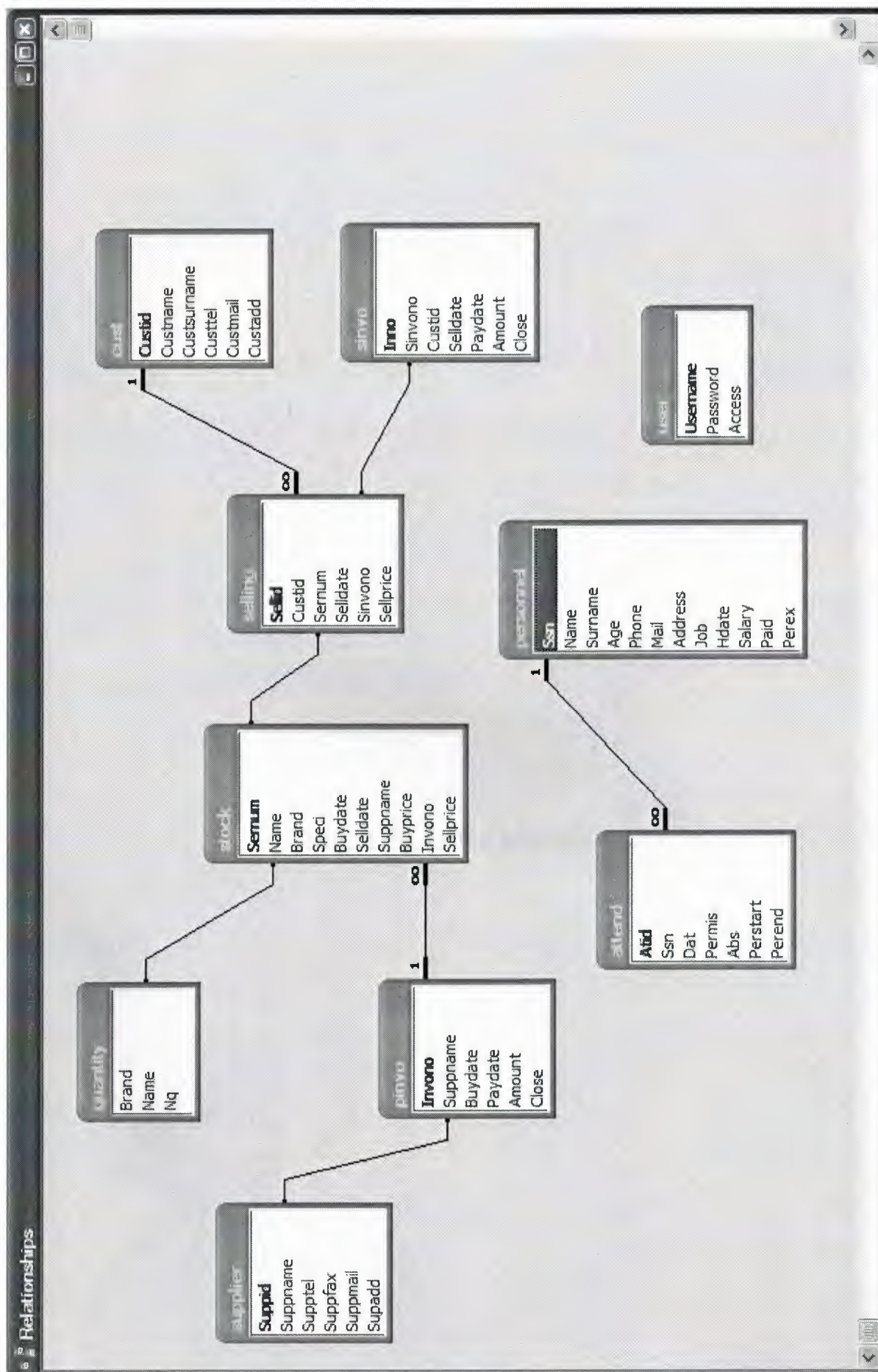
## **DATABASE DESIGN**

## **DATABASE TABLES**

There are 10(ten) tables in system Database, they are ;

- Users
- Personnel
- Attendance
- Customer
- Supplier
- Stock
- Pro\_form
- P\_invoices
- Selling
- S\_invoices

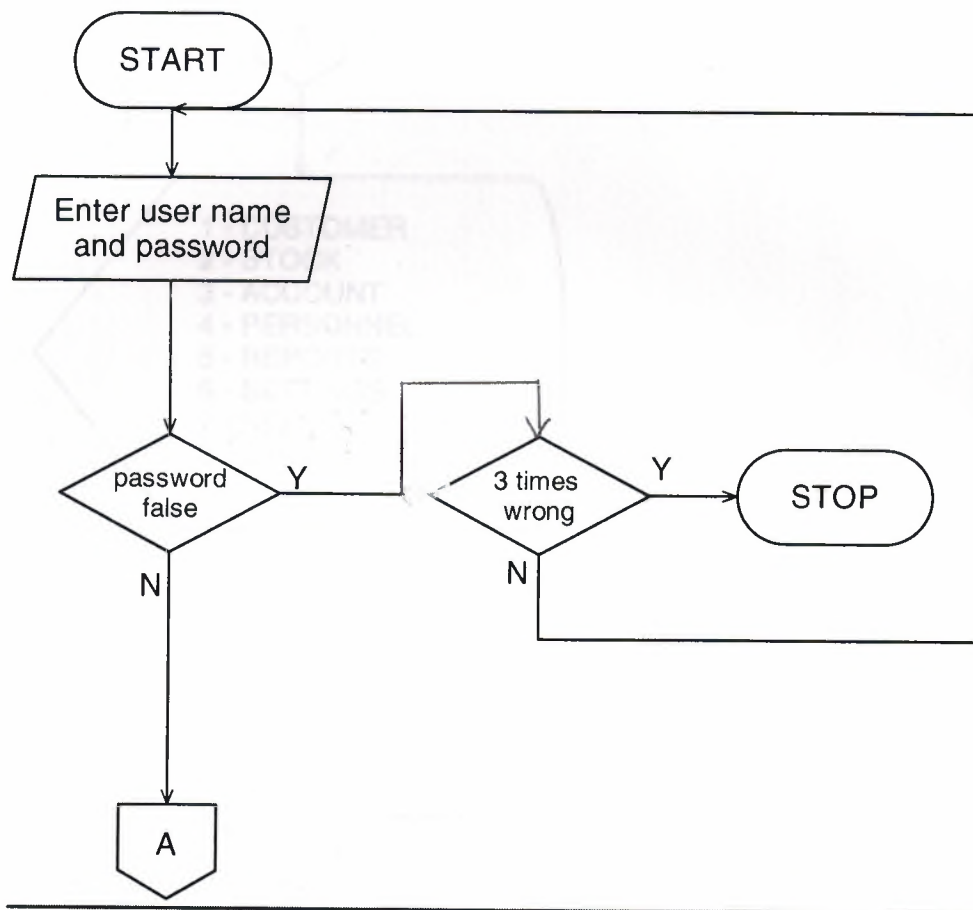




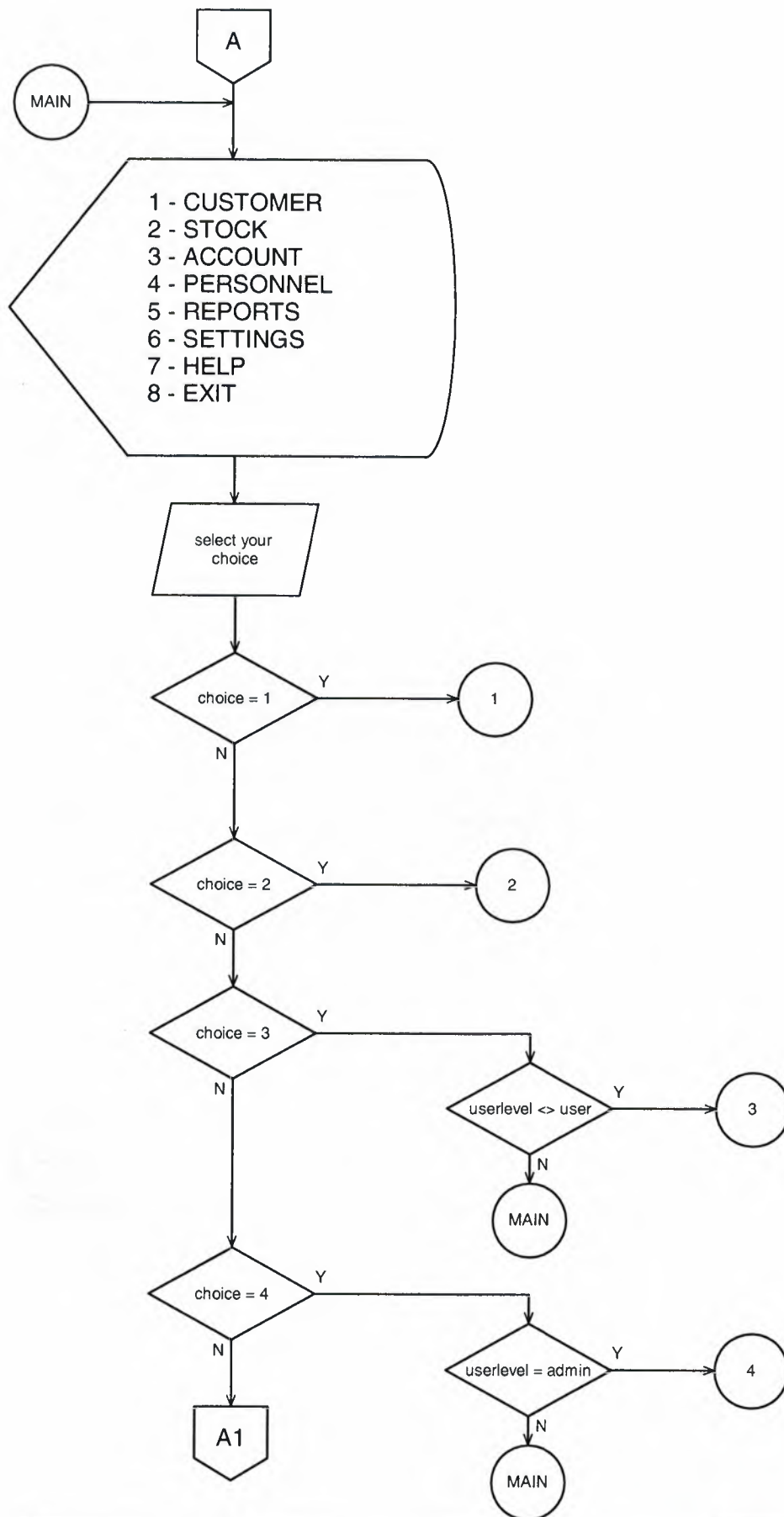
## **USER FLOW CHARTS**

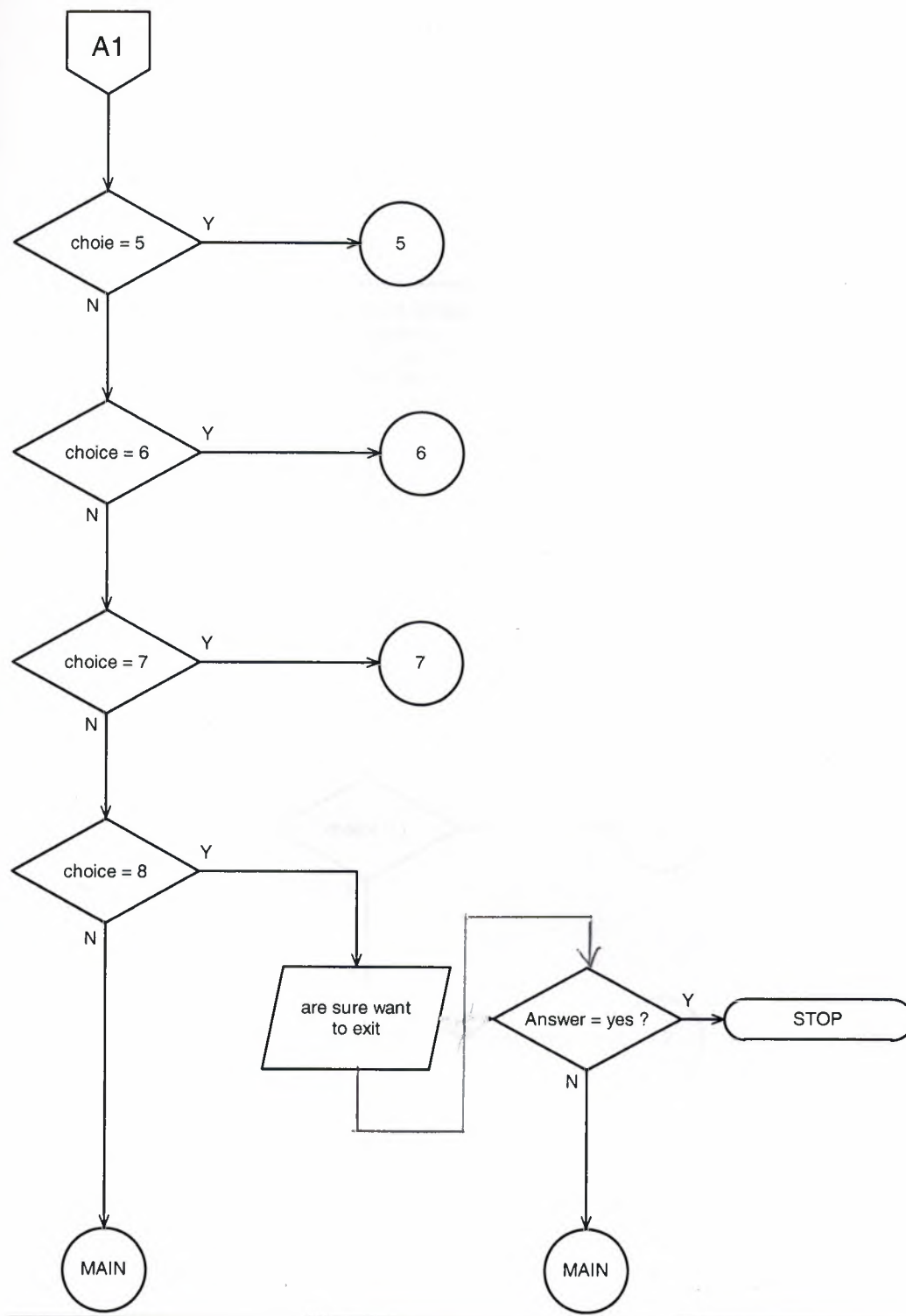


## PASSWORD ENTRY

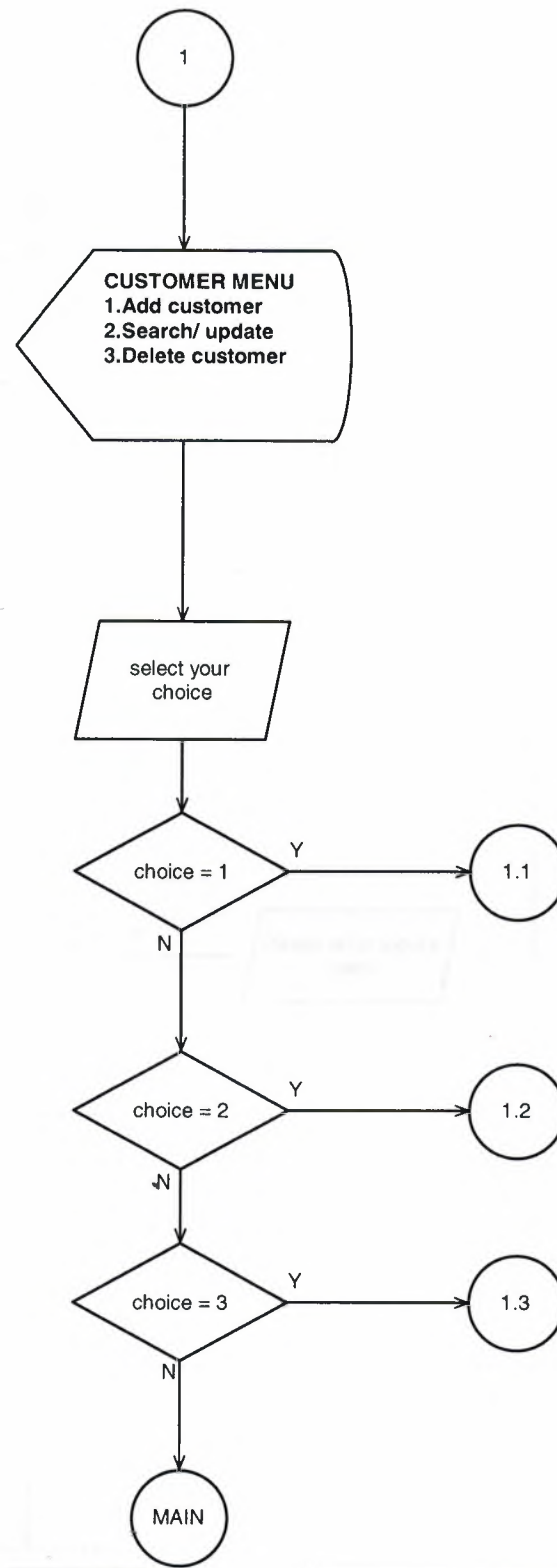


## MAIN MENU

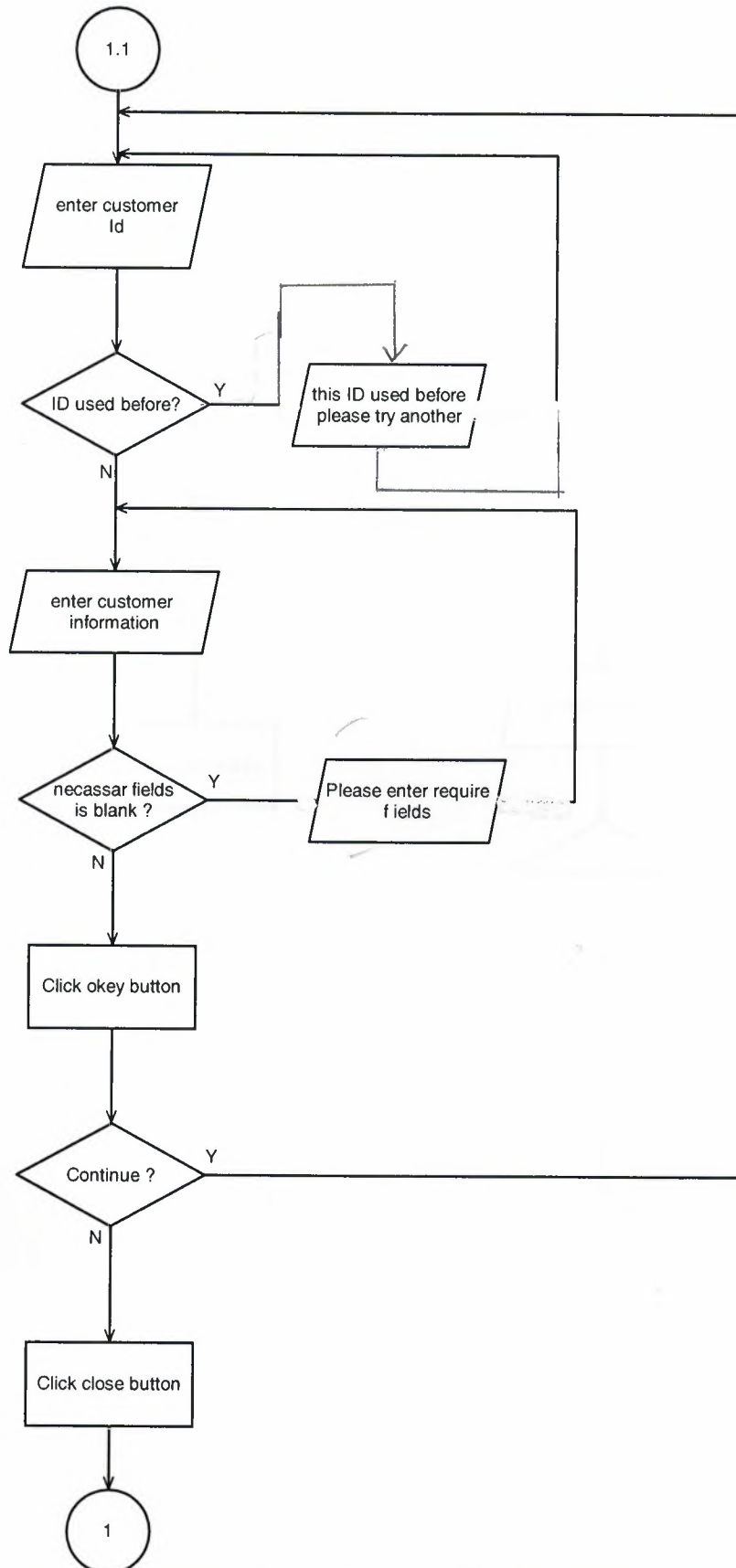




## CUSTOMER

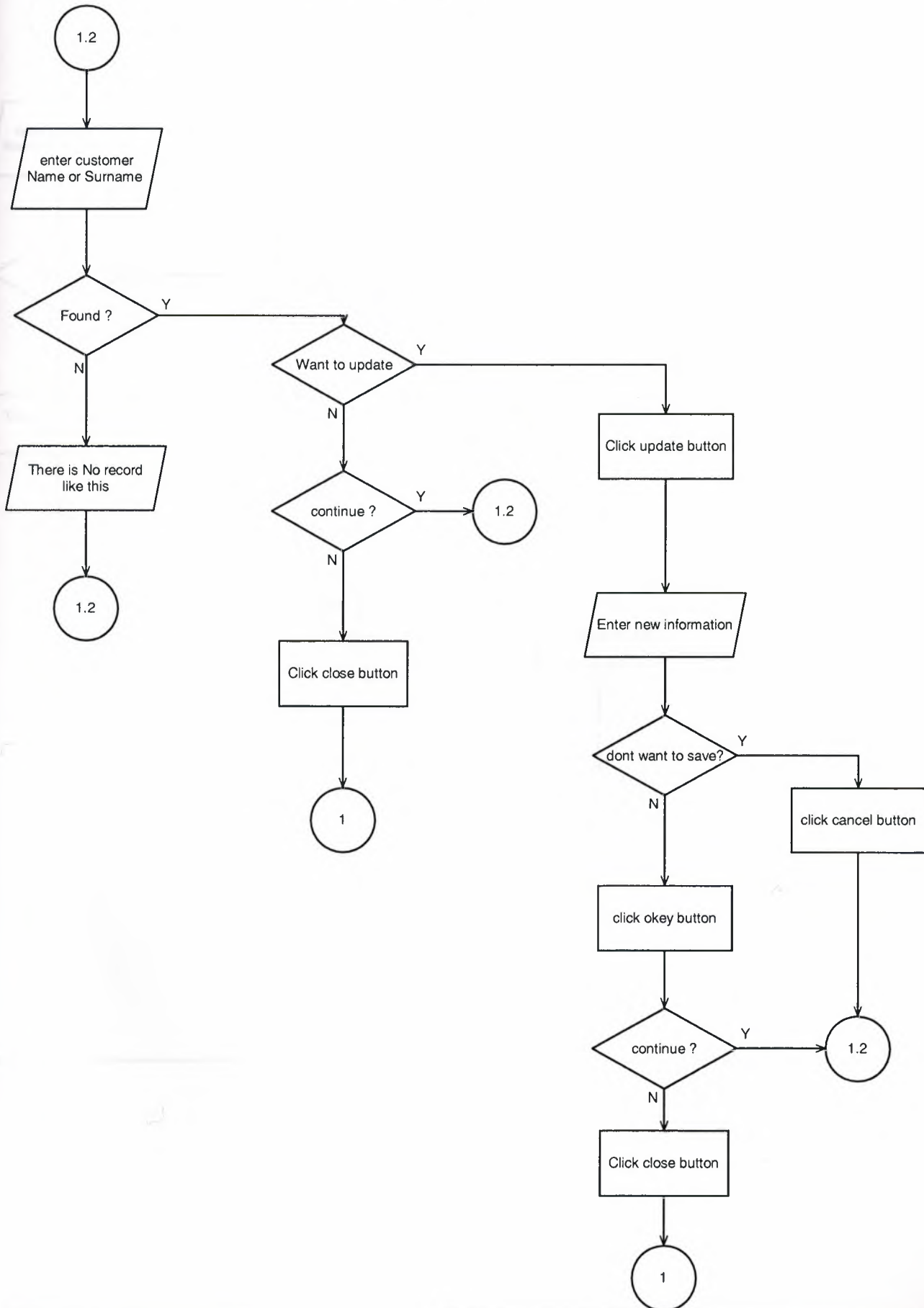


## ADD CUSTOMER

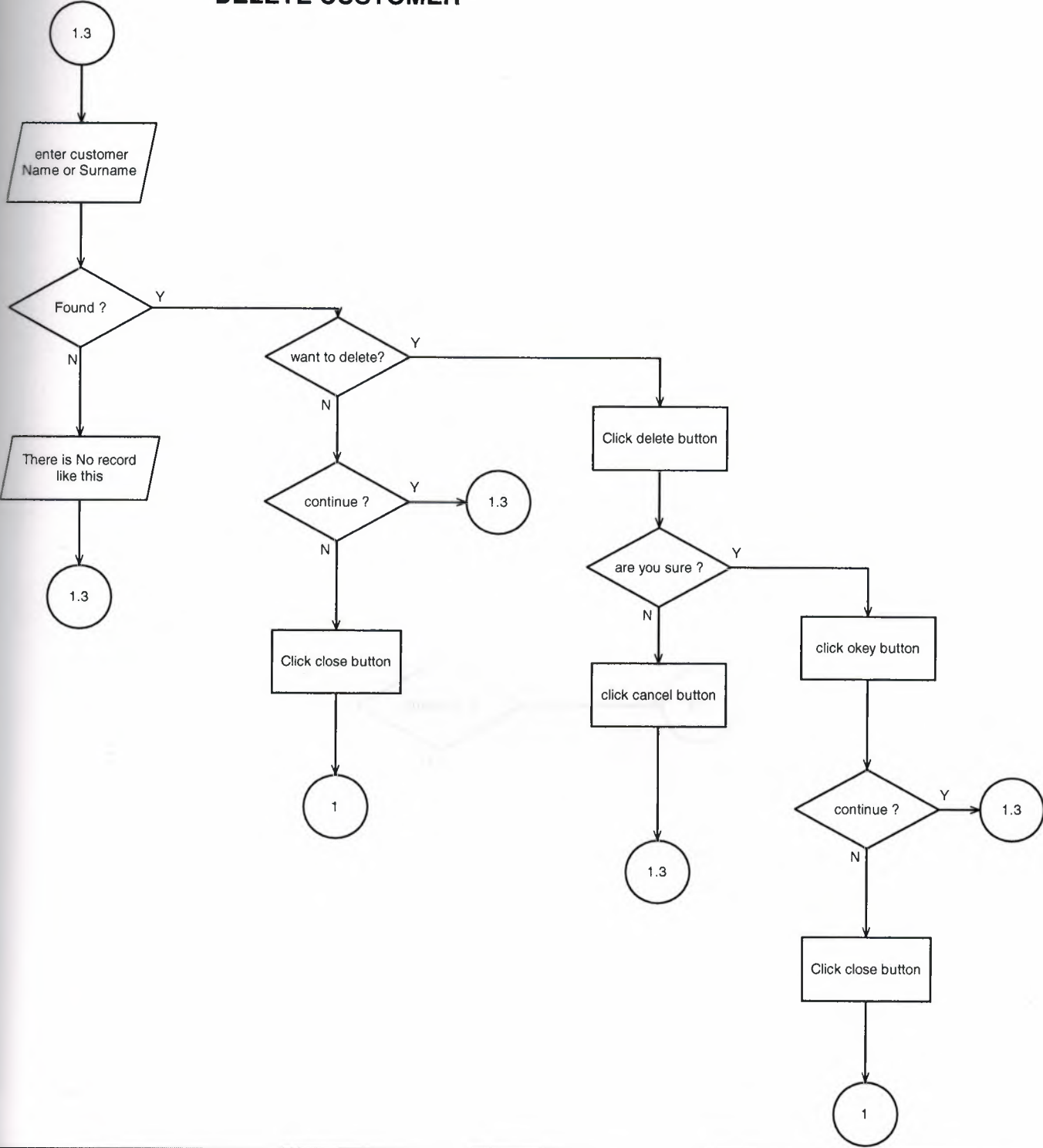




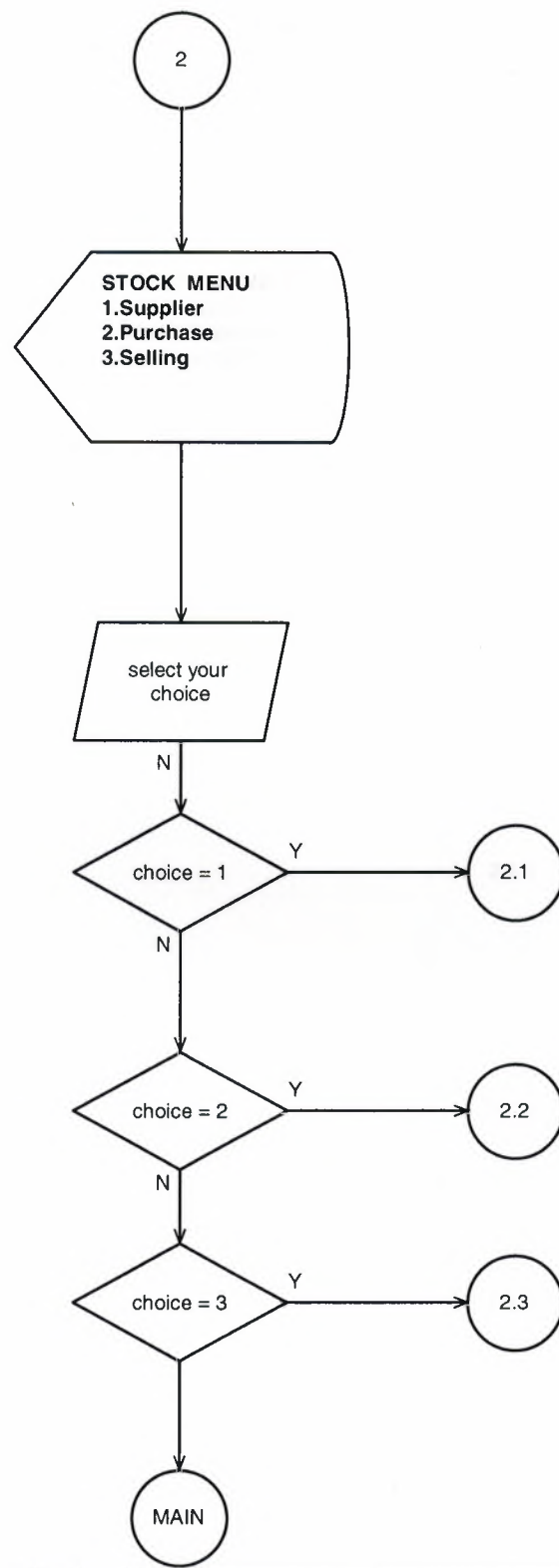
## SEARCH / UPDATE



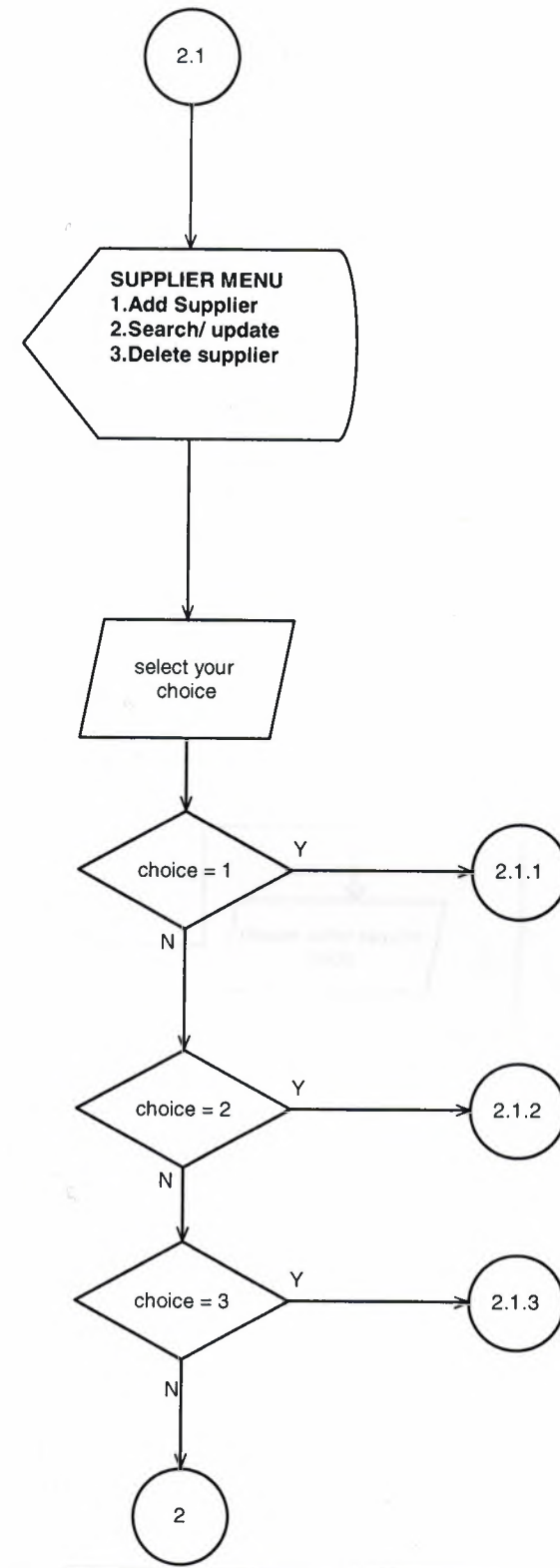
DELETE CUSTOMER



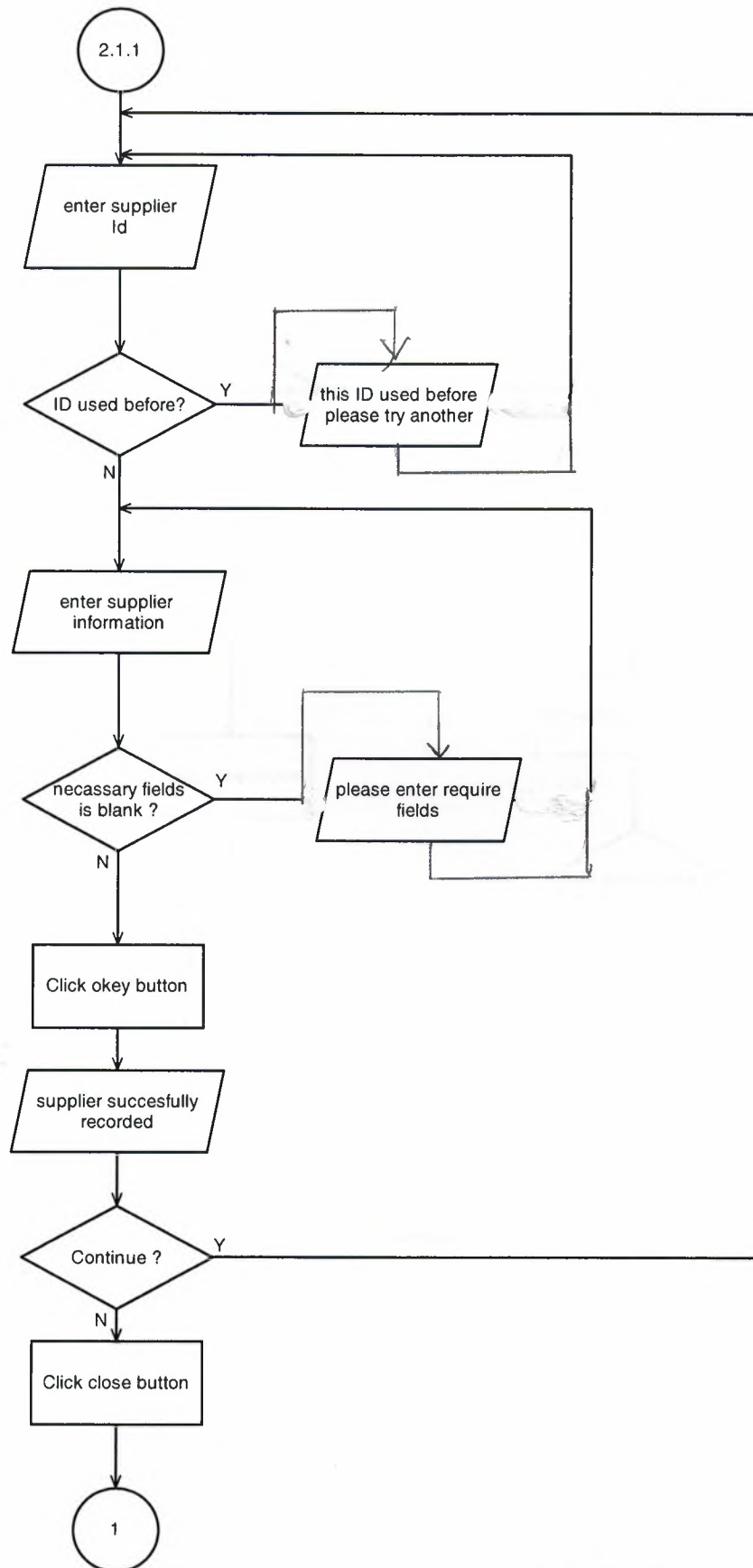
## STOCK



## SUPPLIER

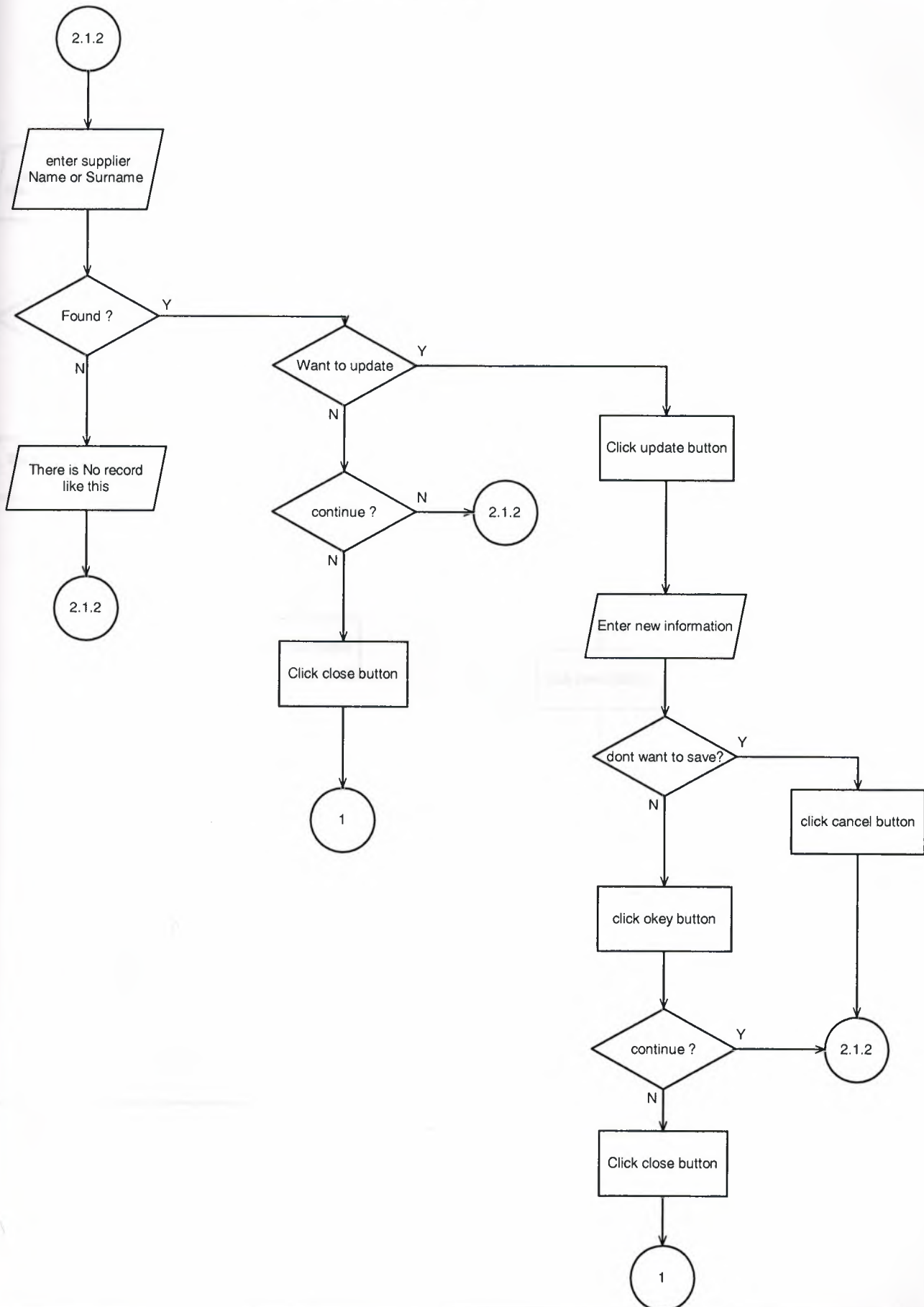


## ADD SUPLIER

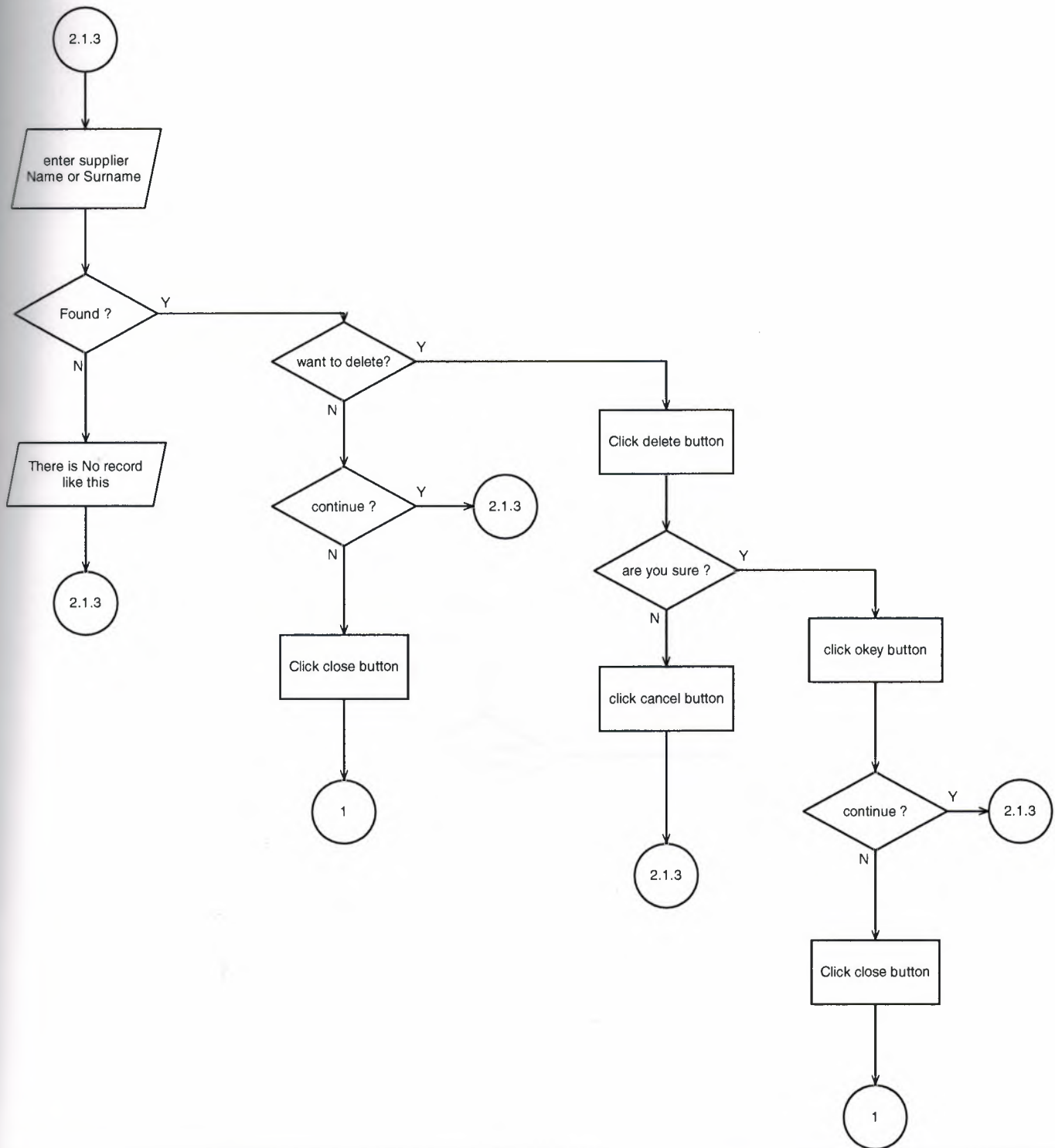




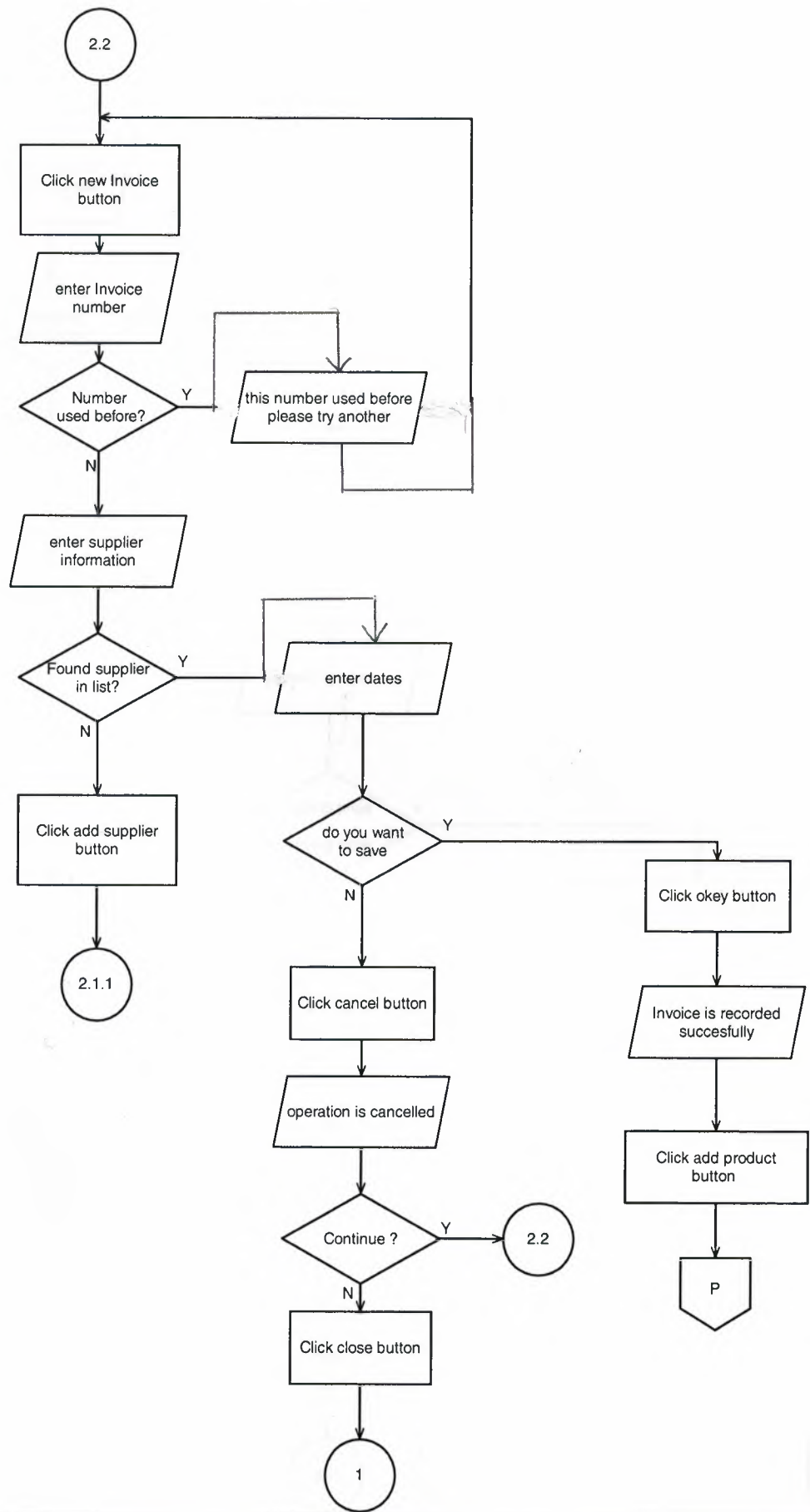
## SEARCH / UPDATE



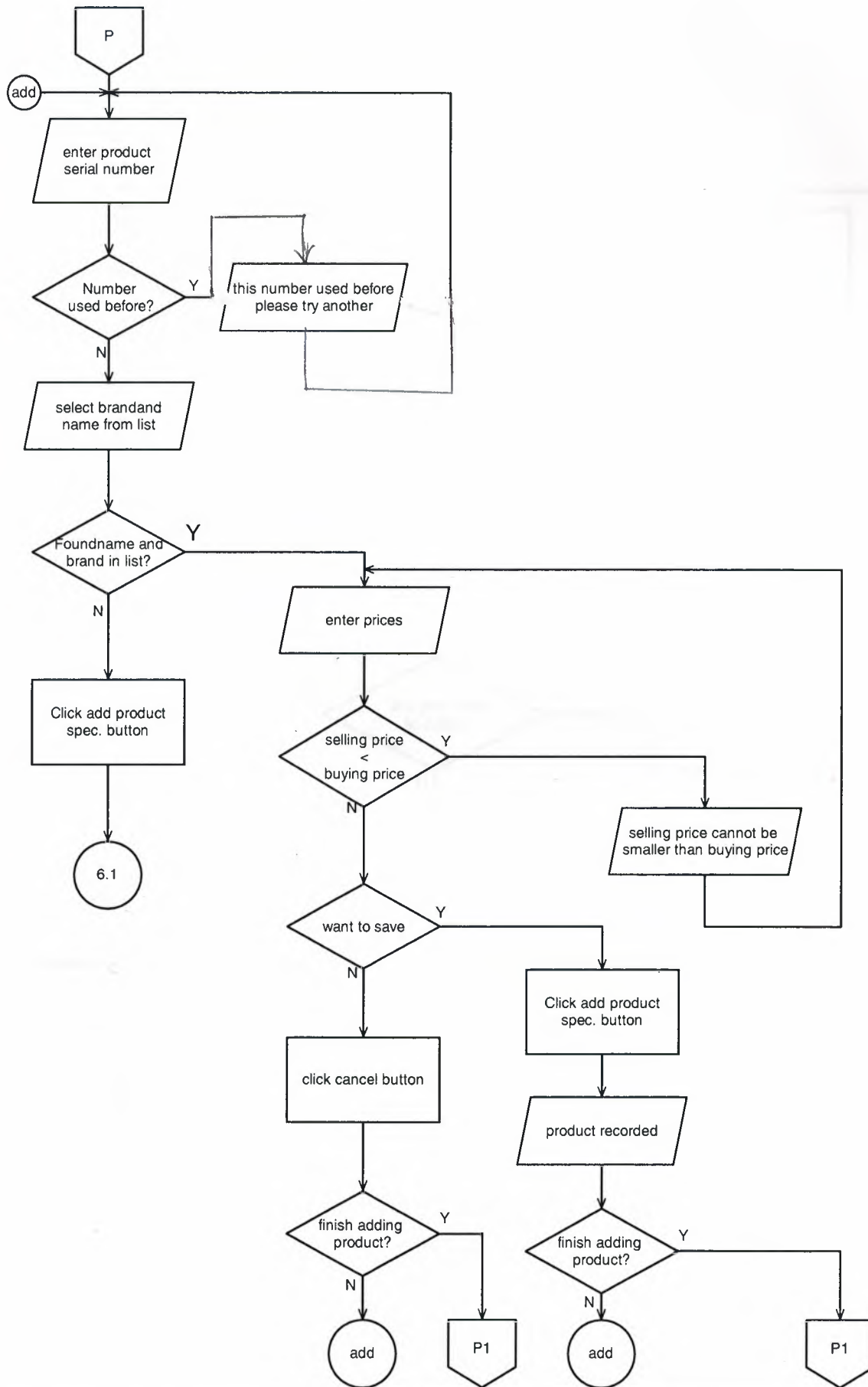
## DELETE SUPPLIER



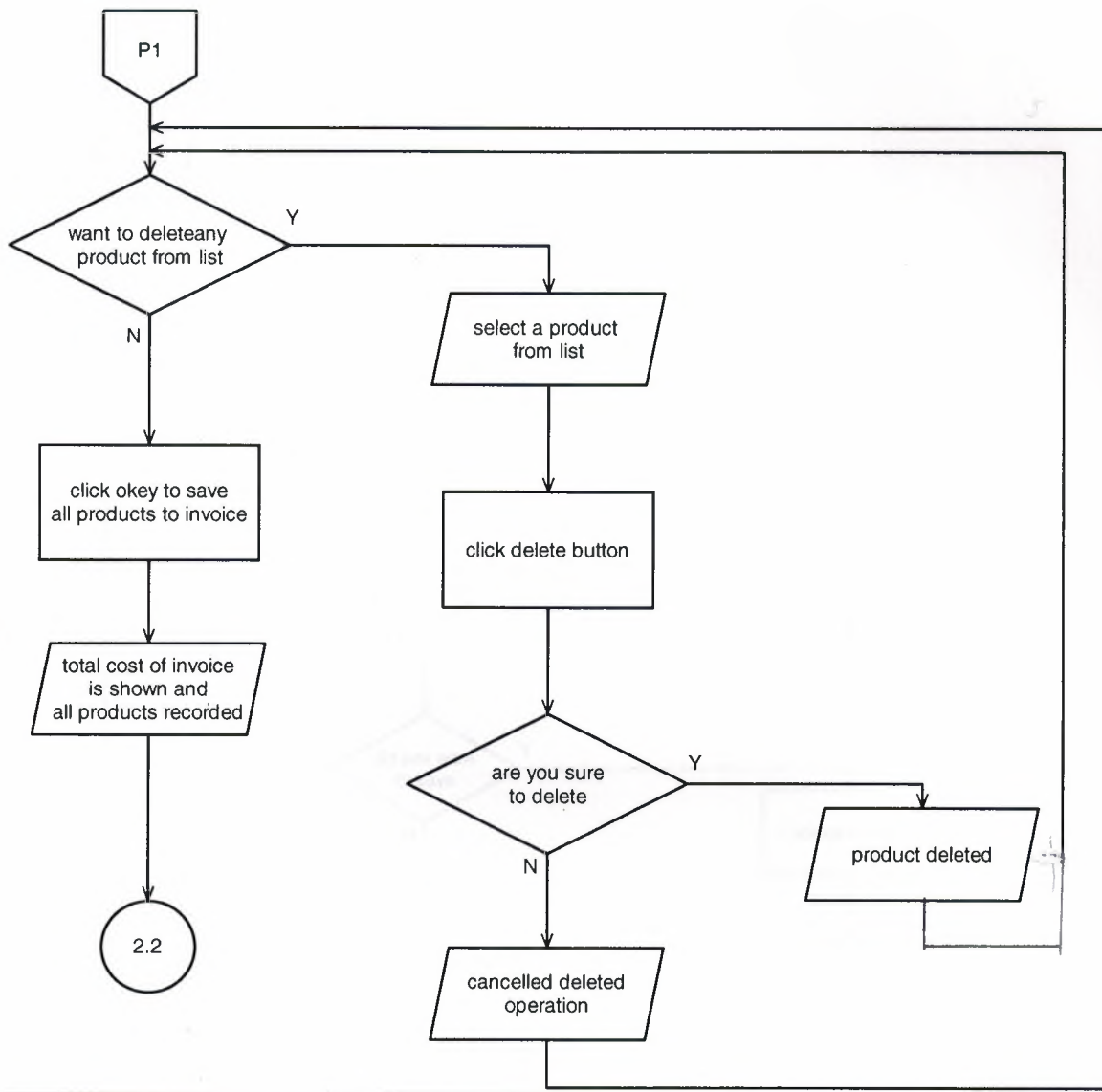
PURCHASE



## PURCHASE

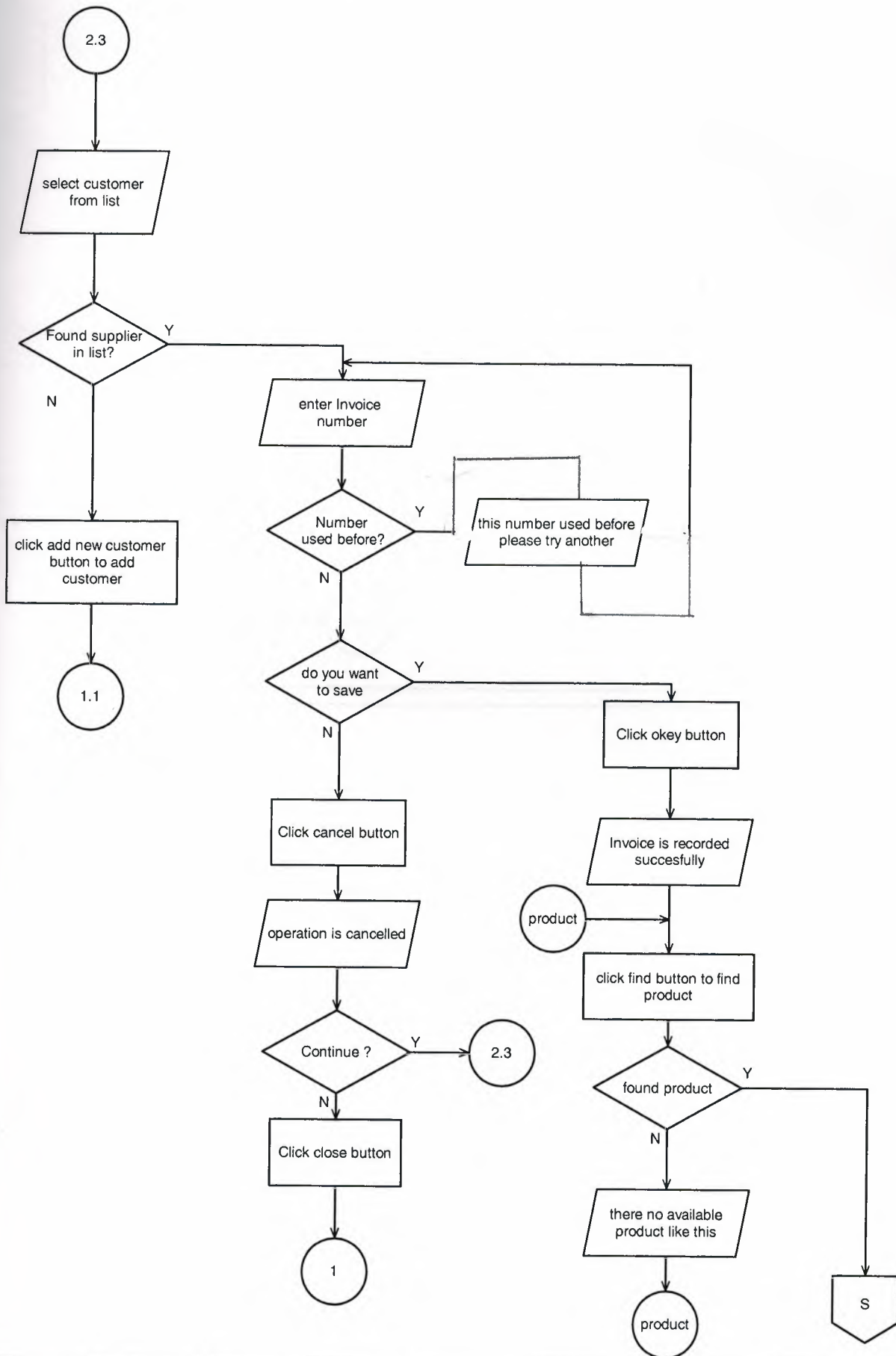


## PURCHASE

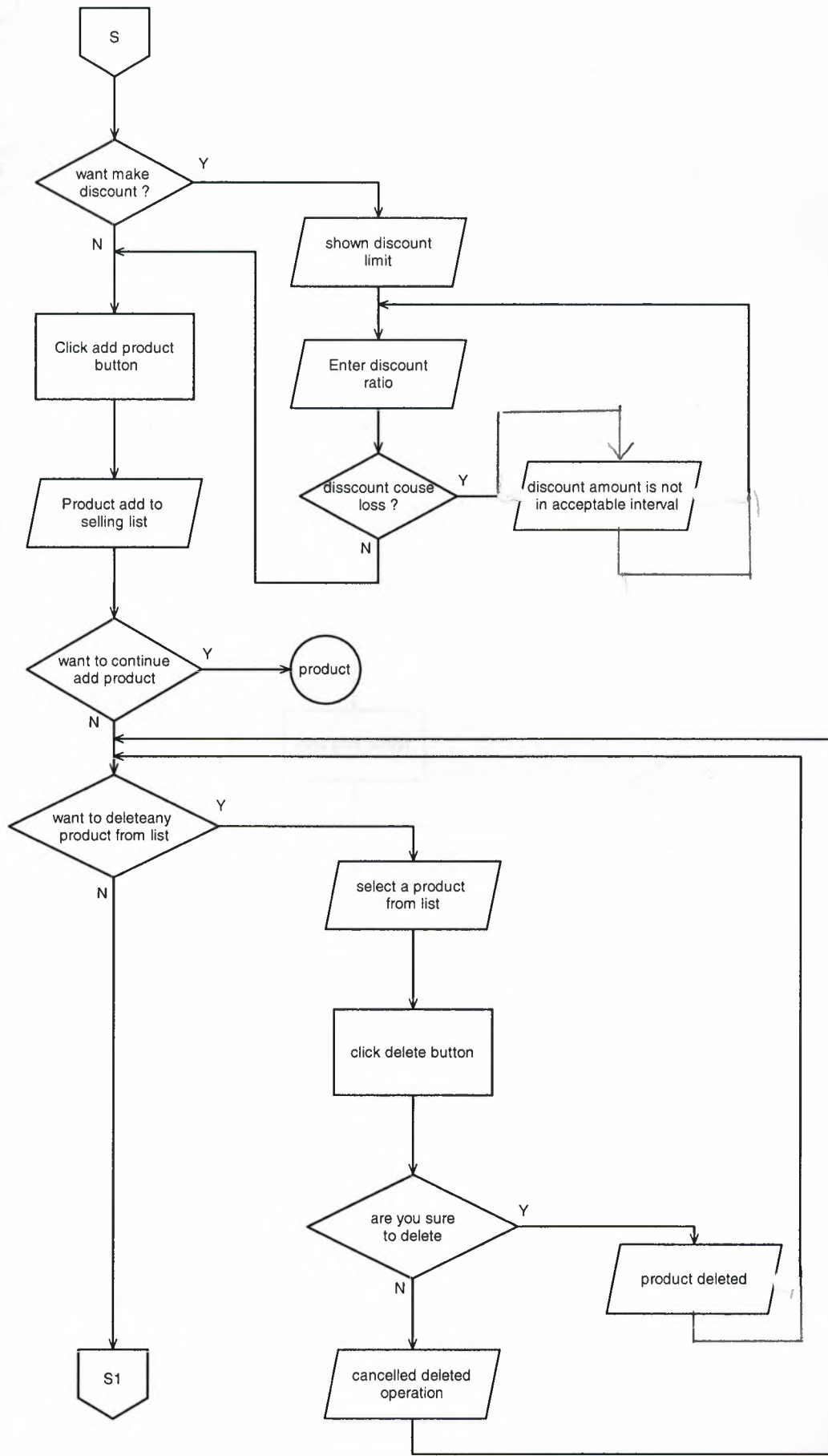




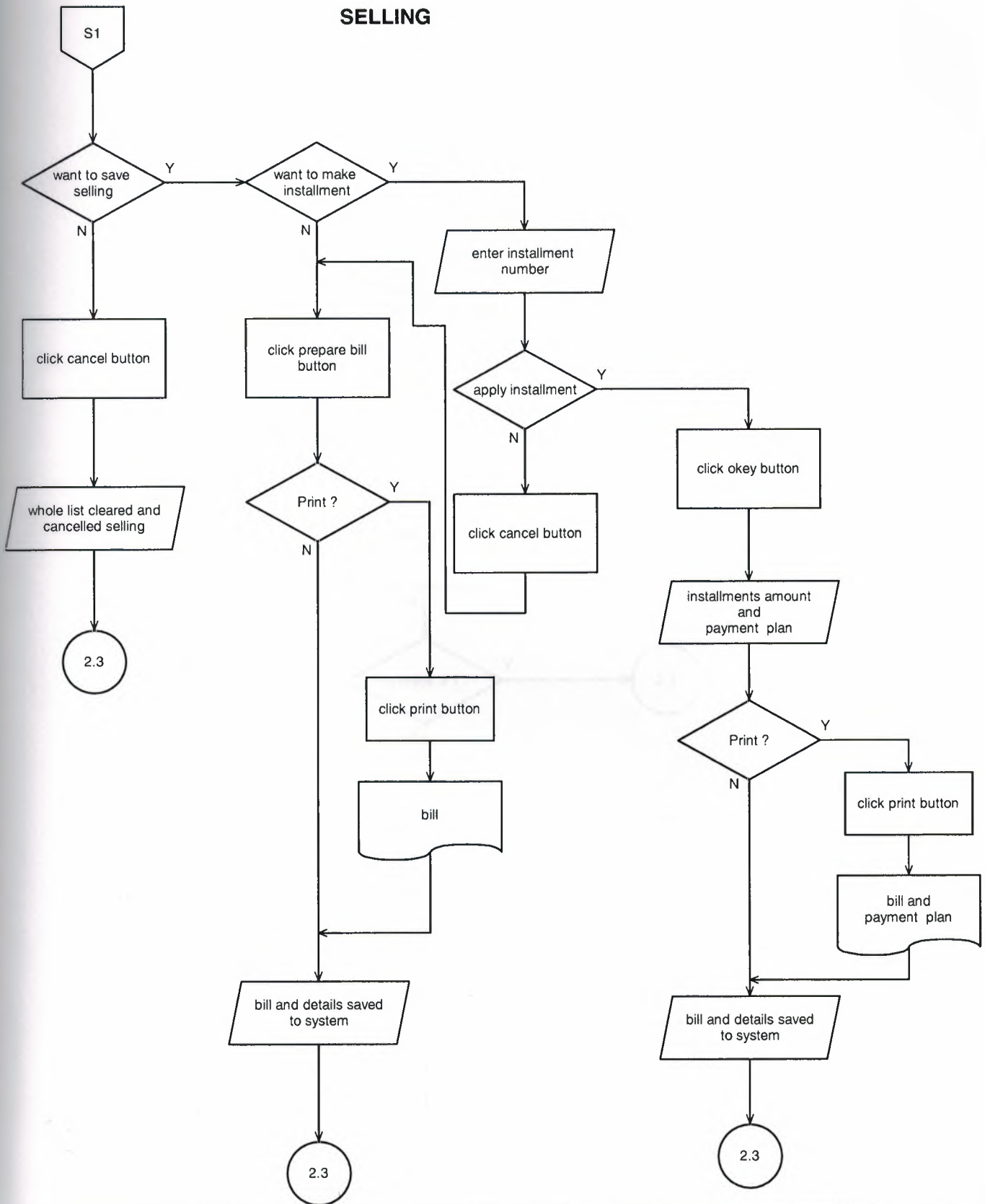
## SELLING



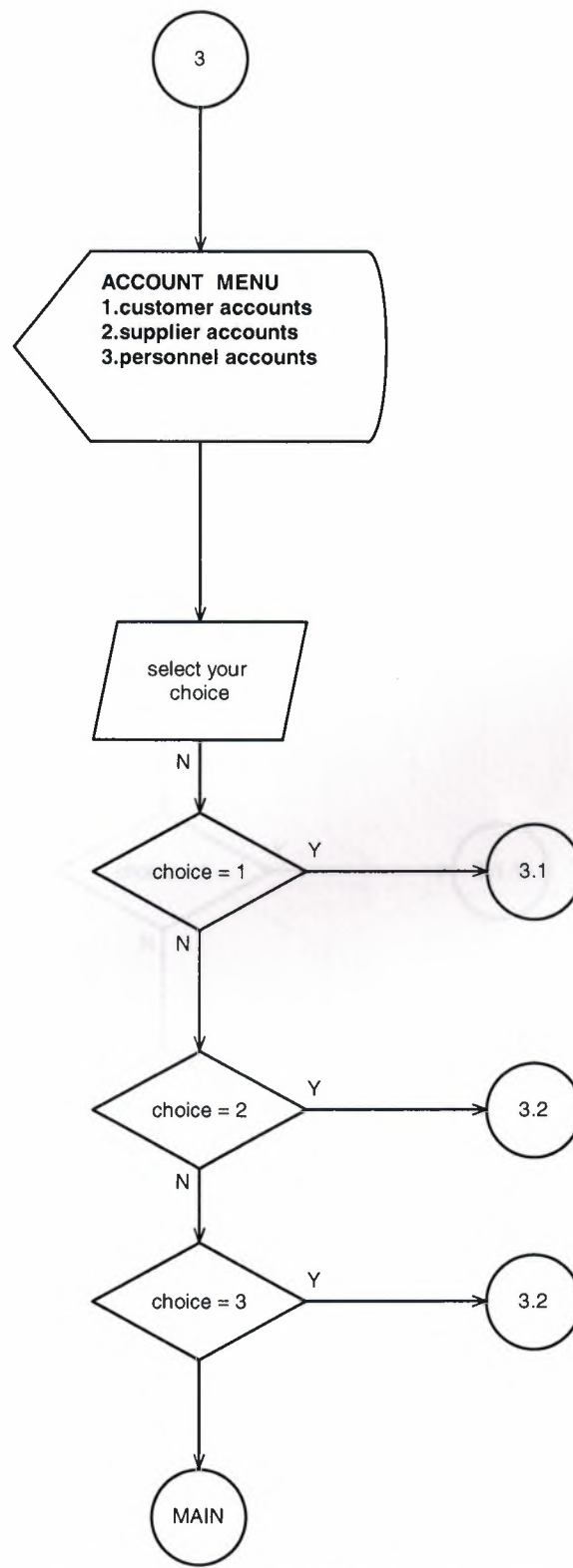
## SELLING



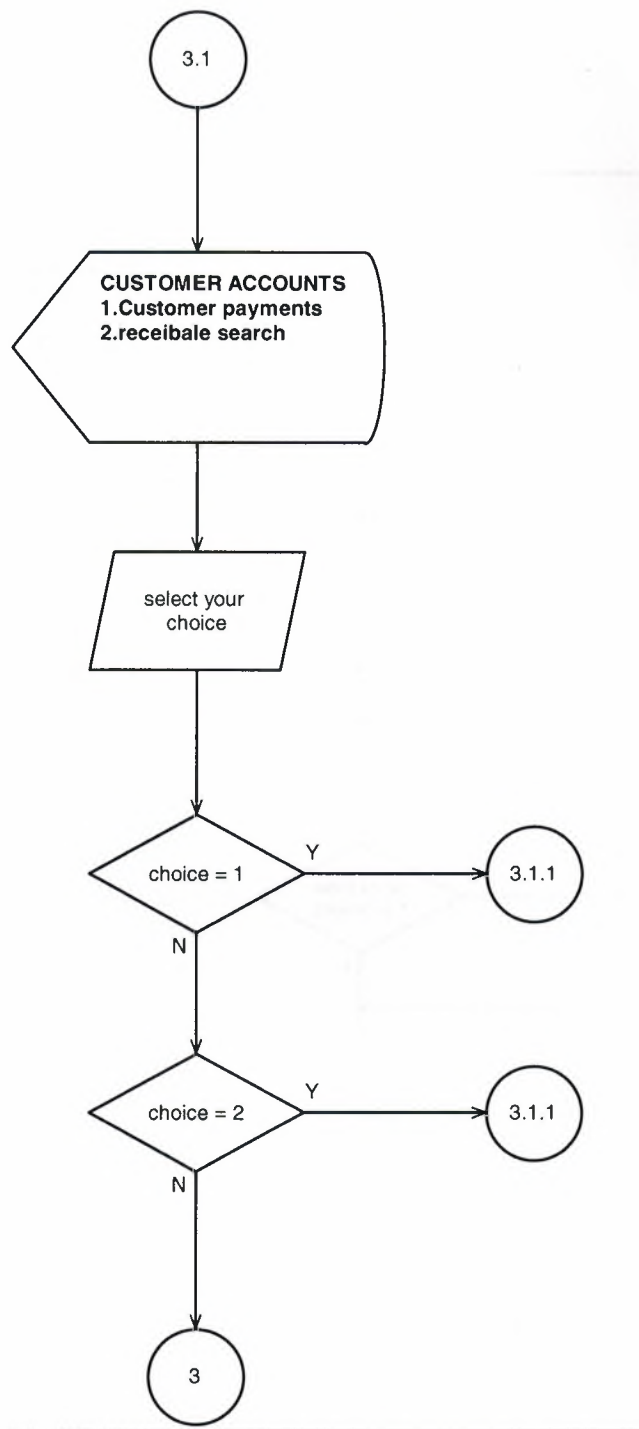
## SELLING



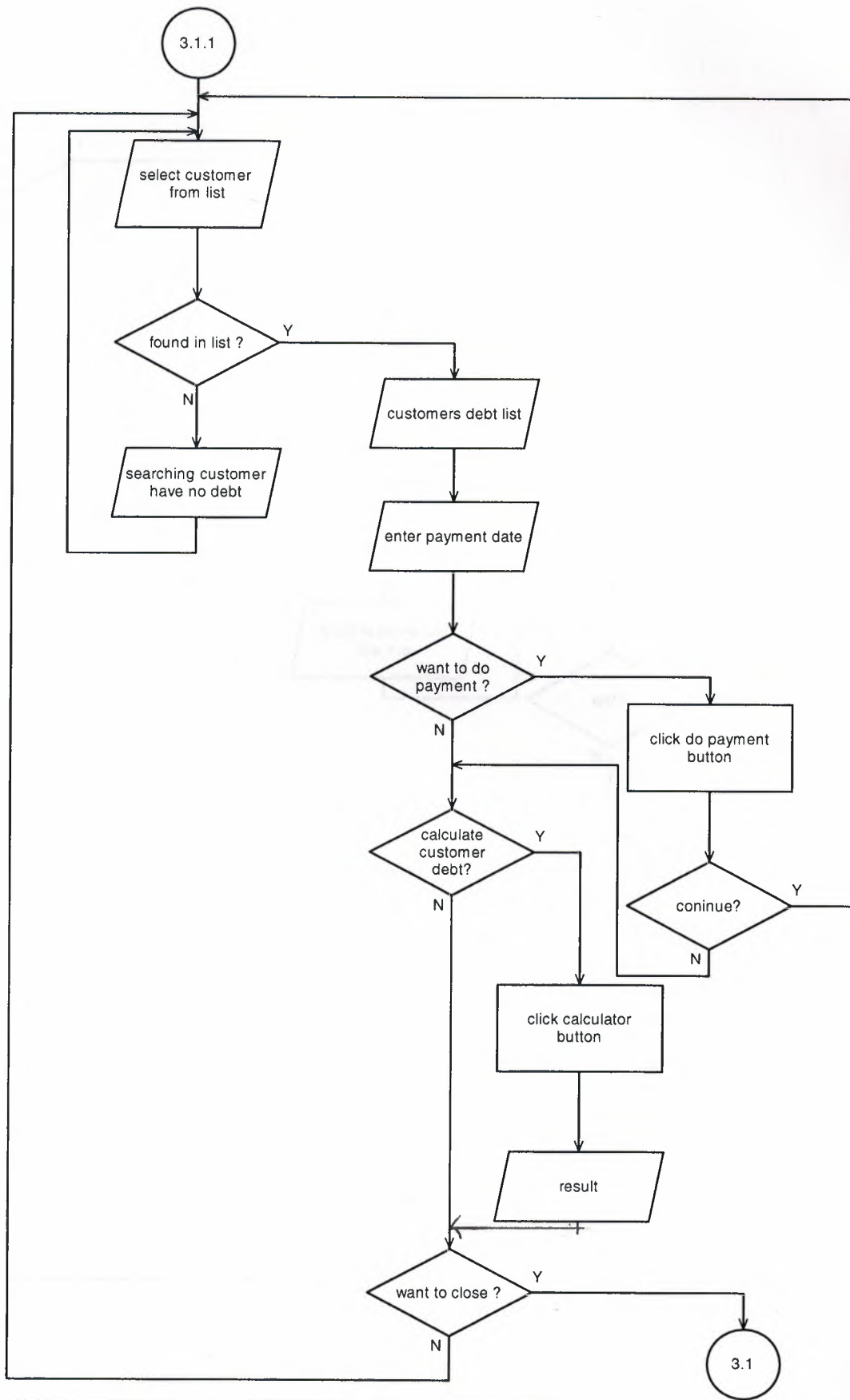
## ACCOUNT



## CUSTOMER ACCOUNTS

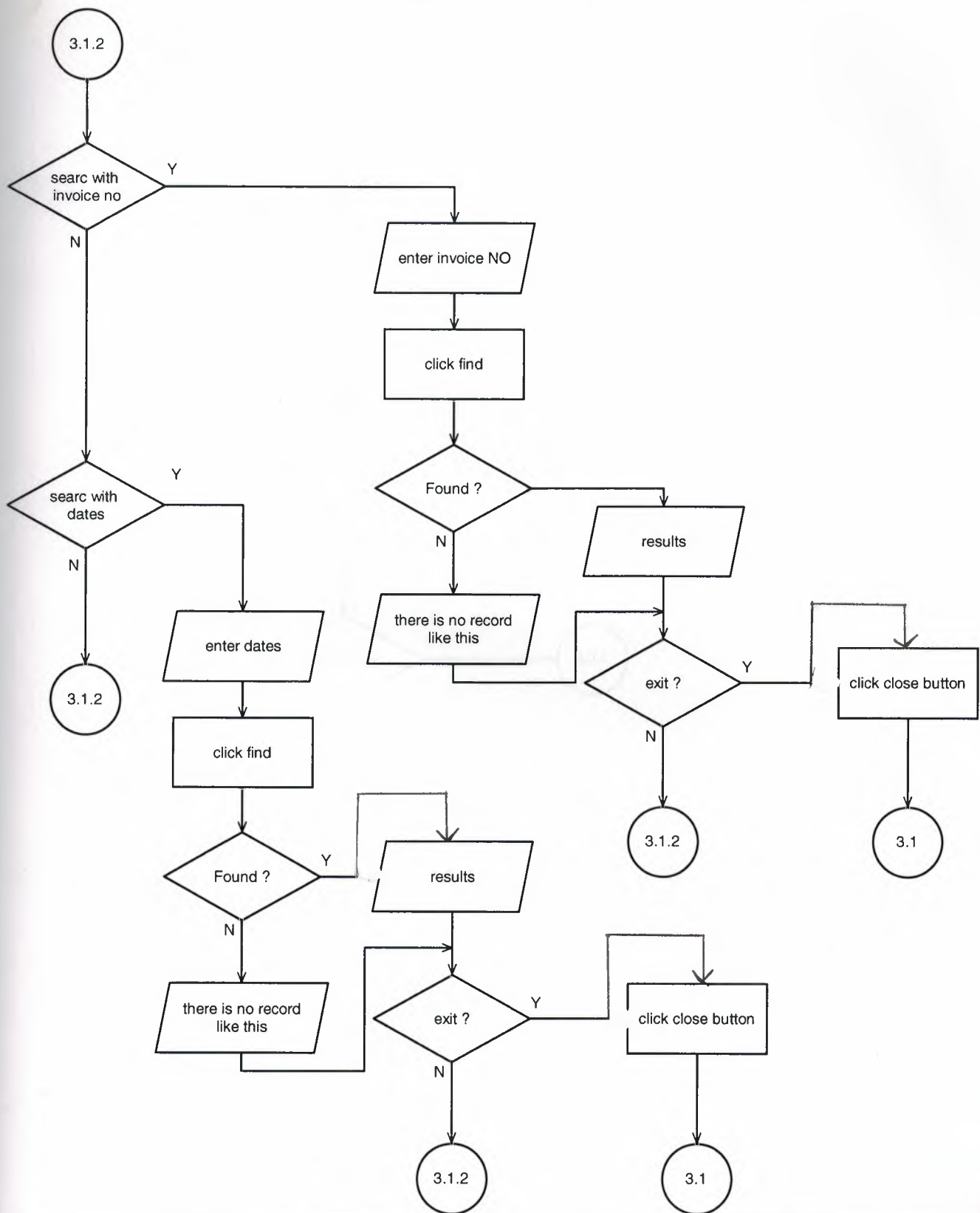


## CUSTOMER PAYMENTS

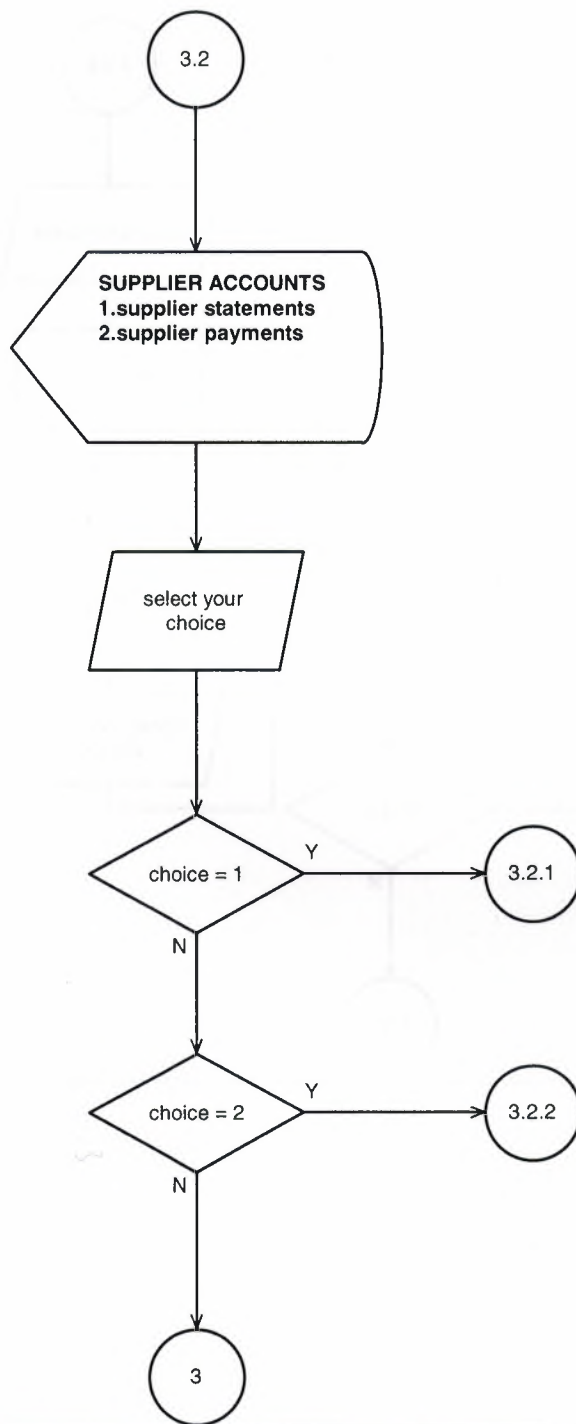




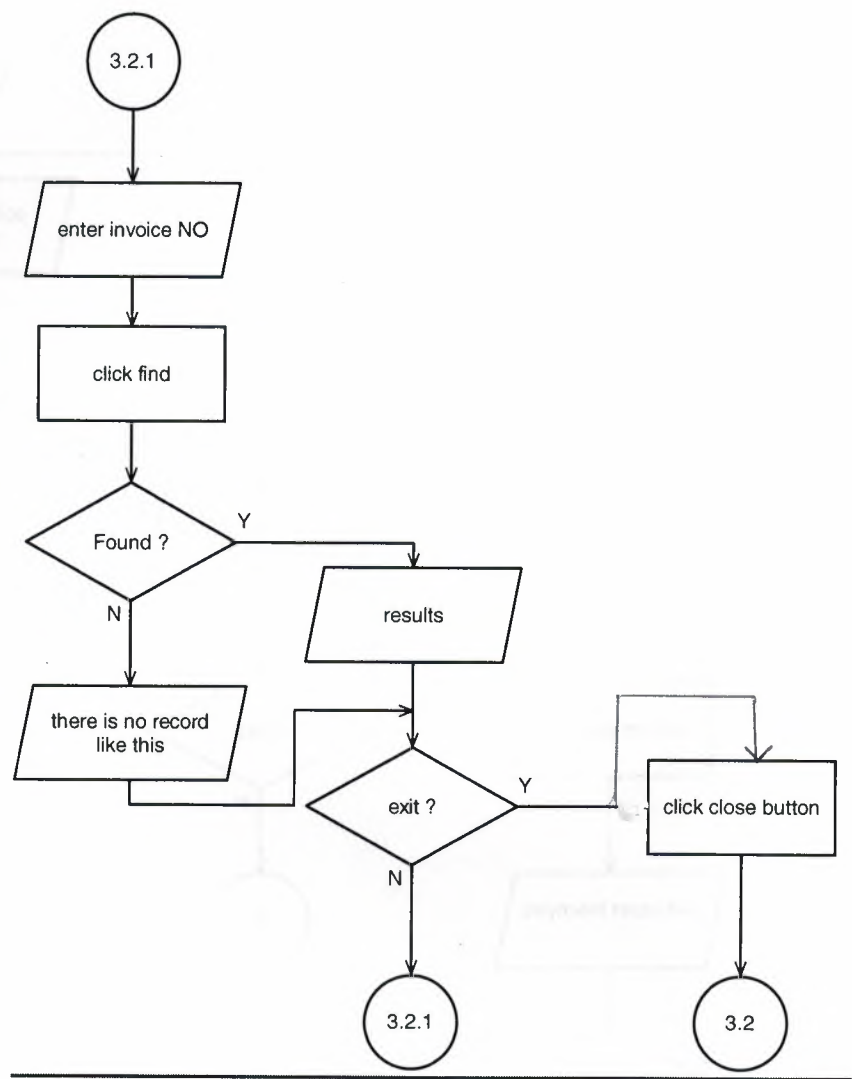
## SEARCH RECEIVABLES



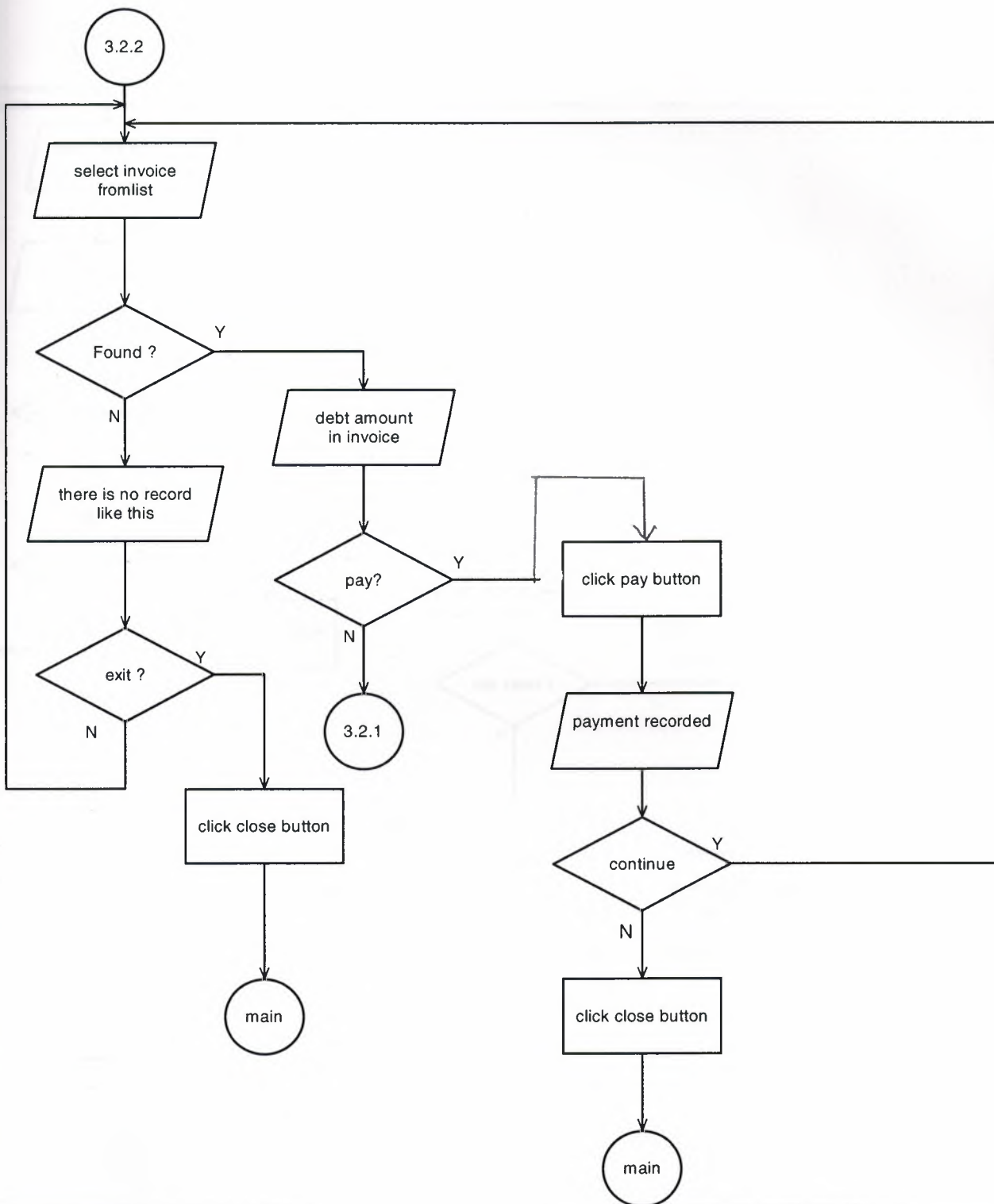
## SUPPLIER ACCOUNTS



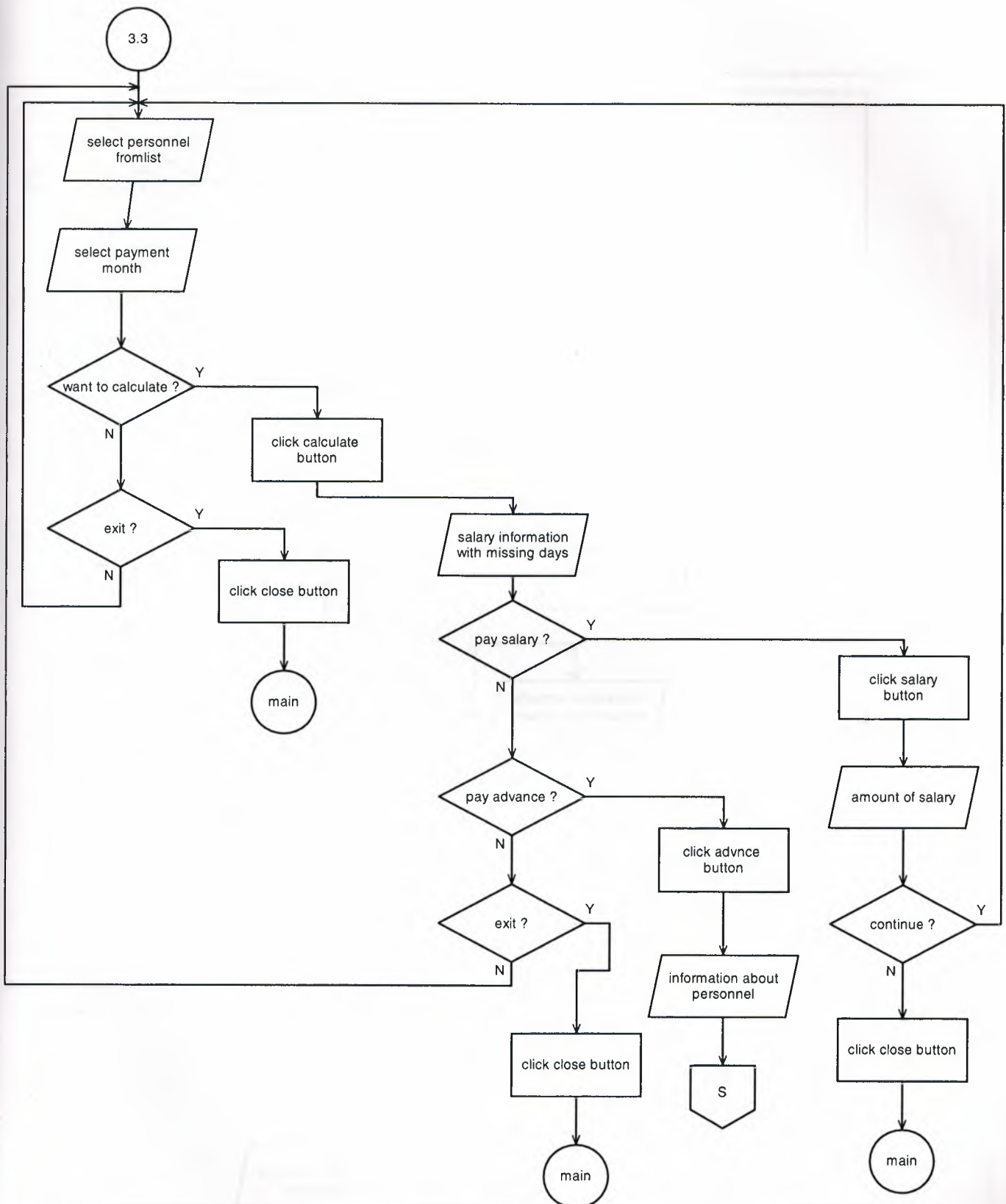
SUPPLIER STATEMENTS



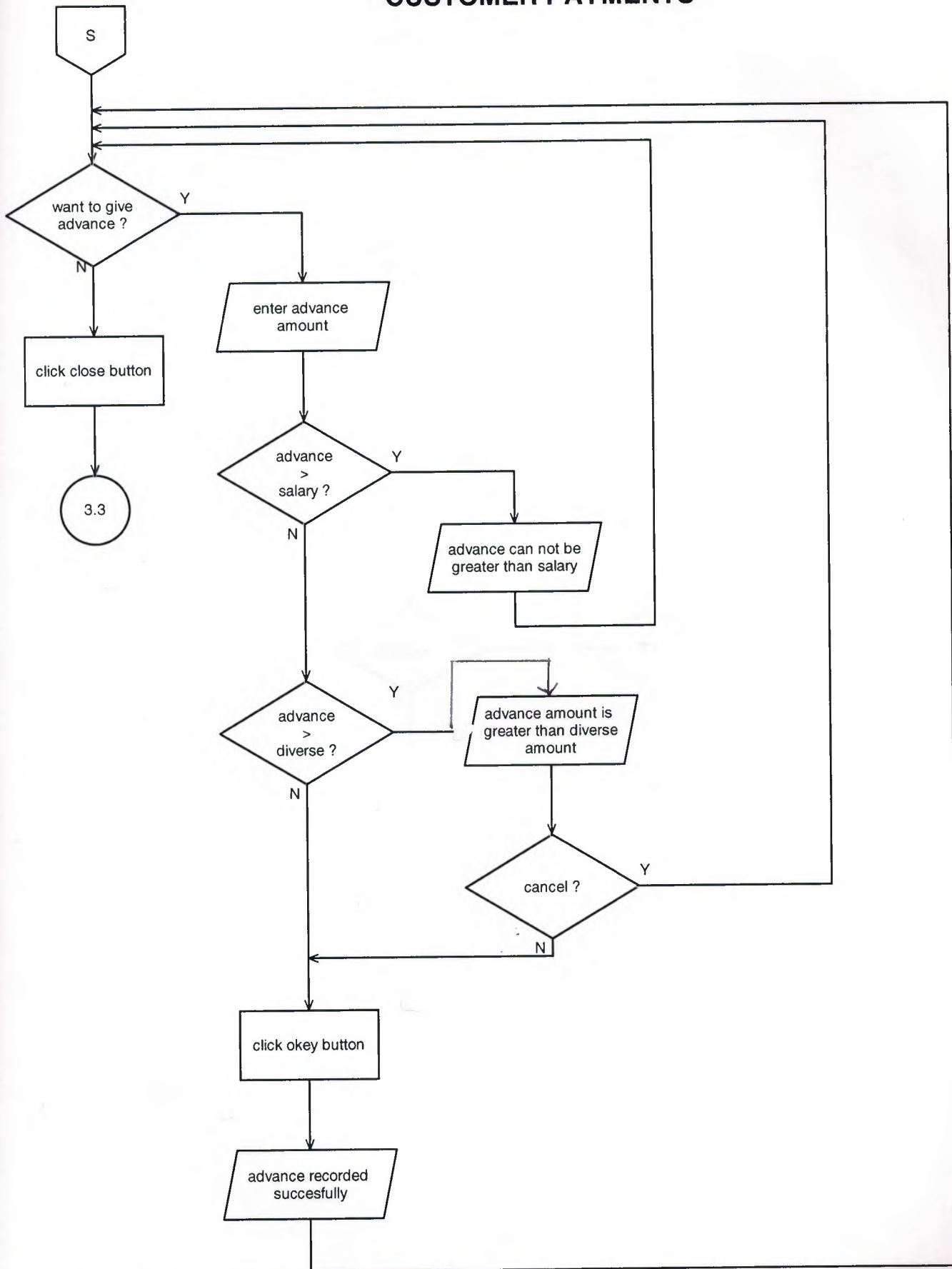
## SUPPLIER PAYMENTS



## PERSONNEL PAYMENTS

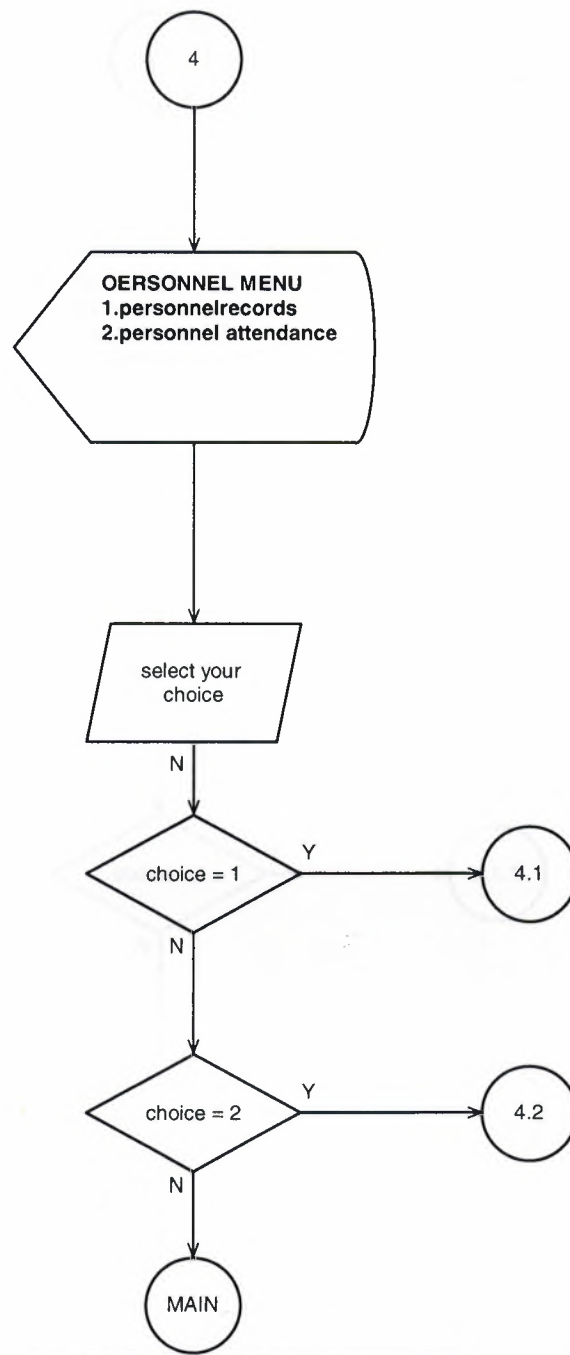


## CUSTOMER PAYMENTS

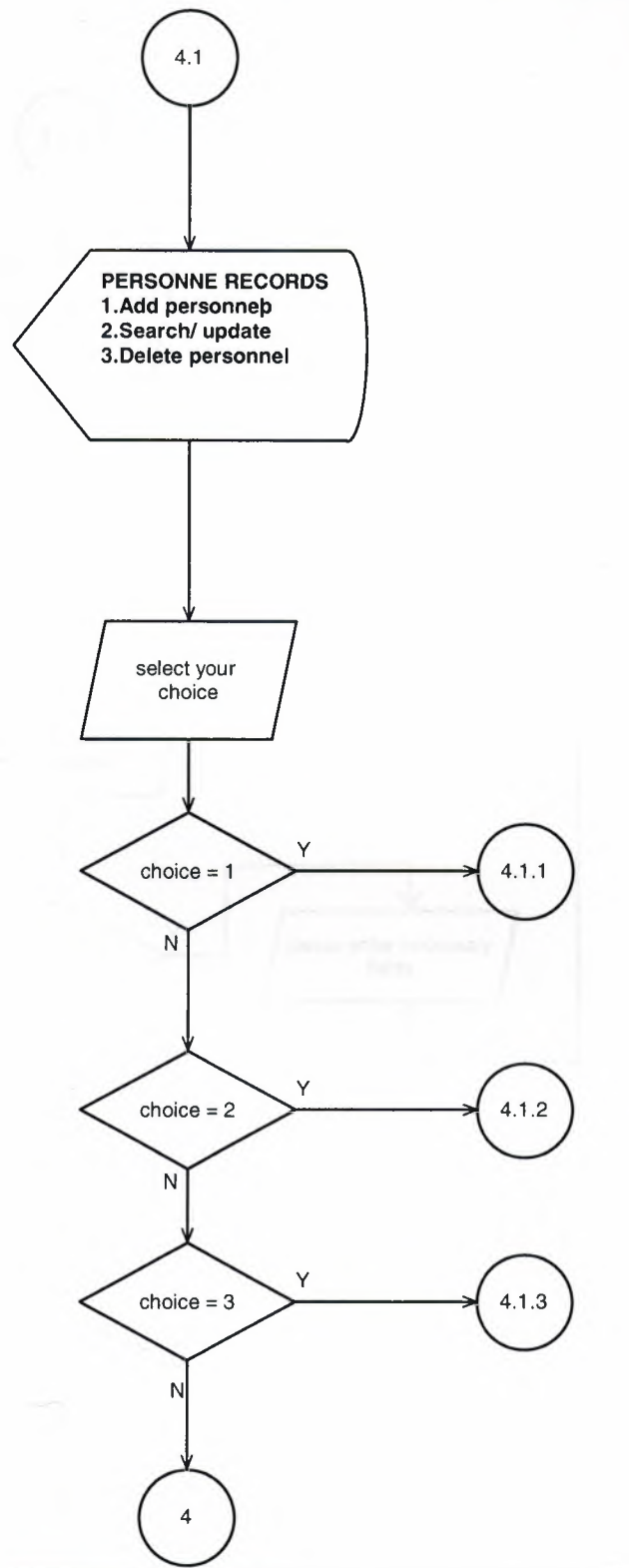




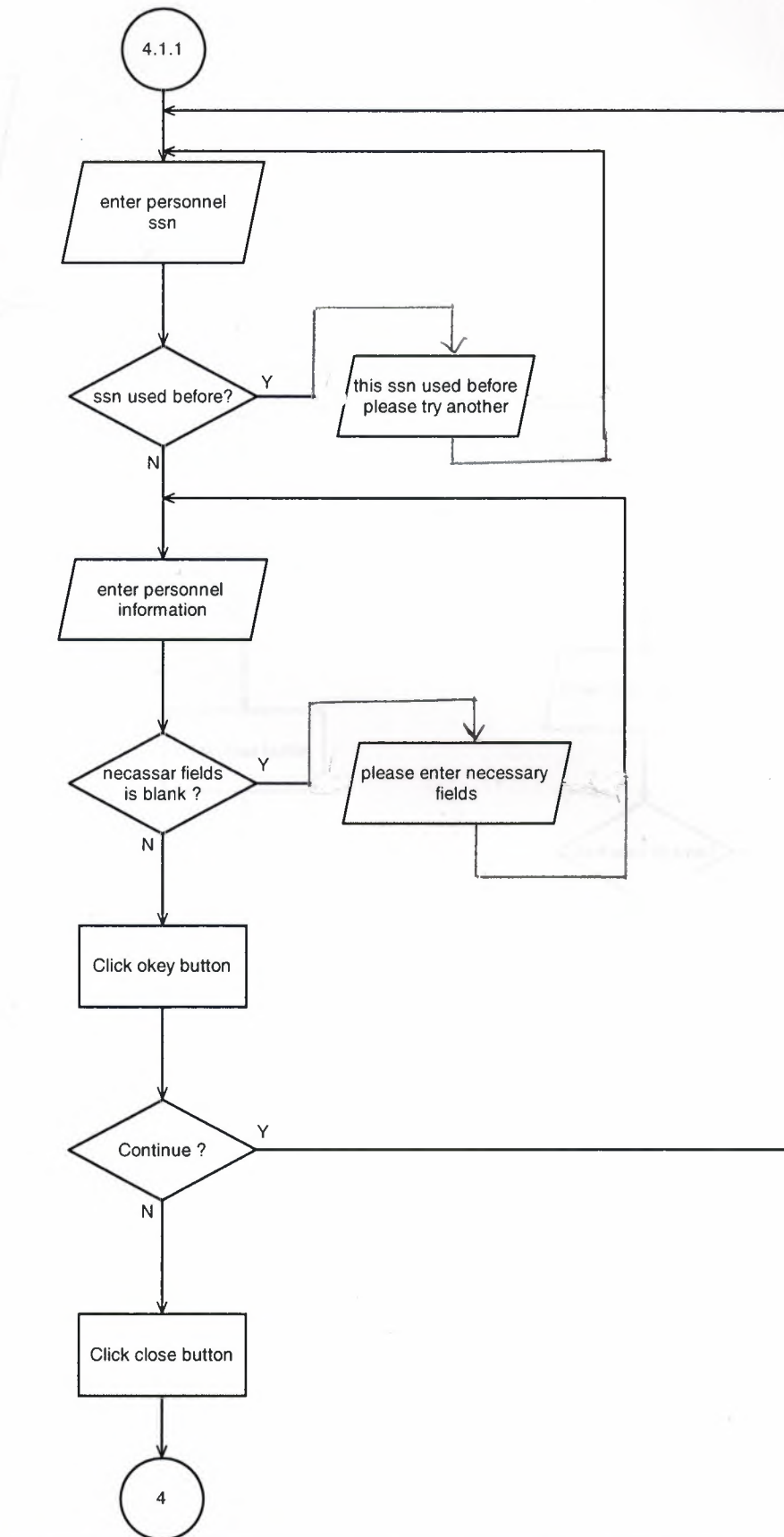
## PERSONNEL RECORDS



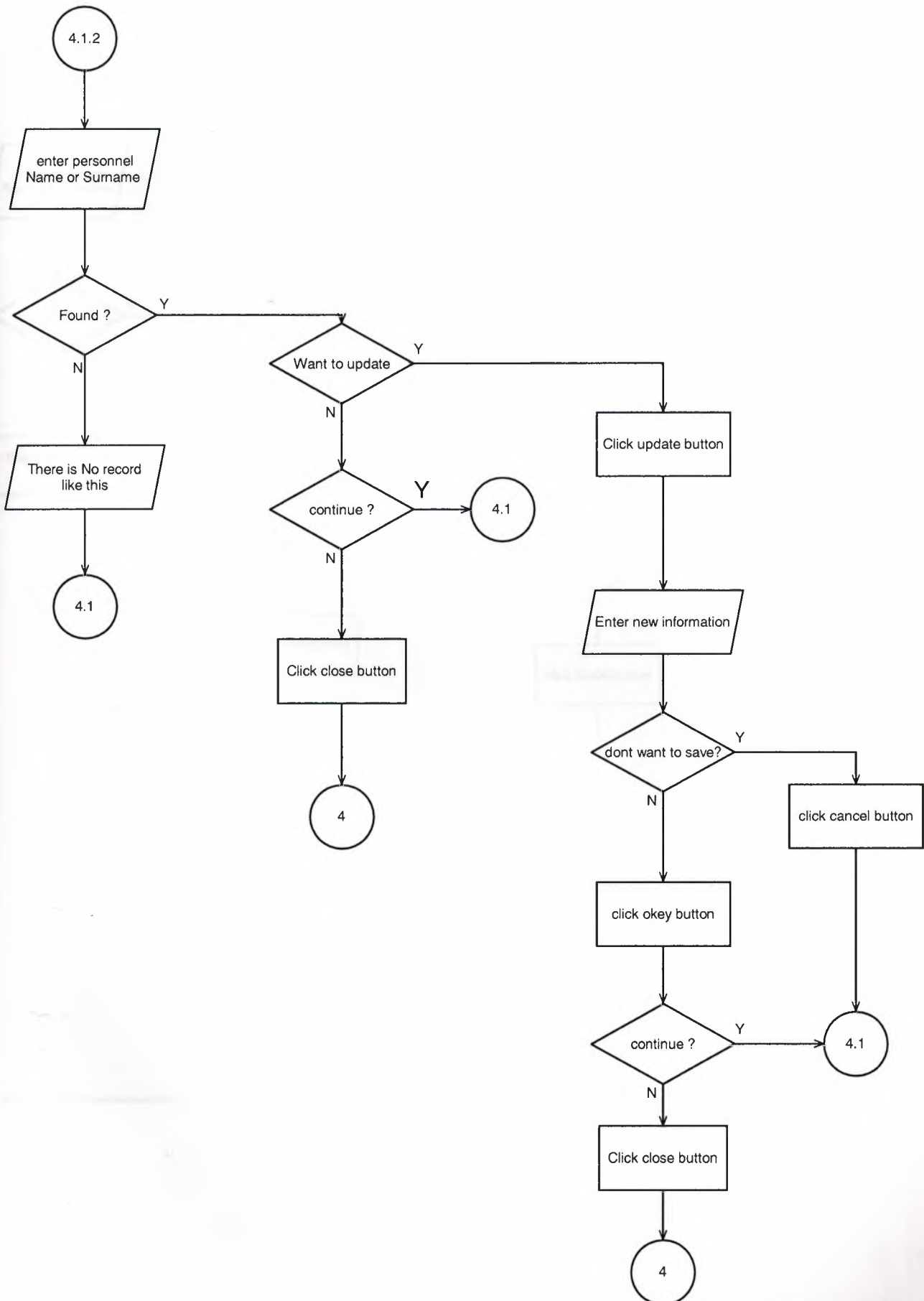
## PERSONNEL RECORDS



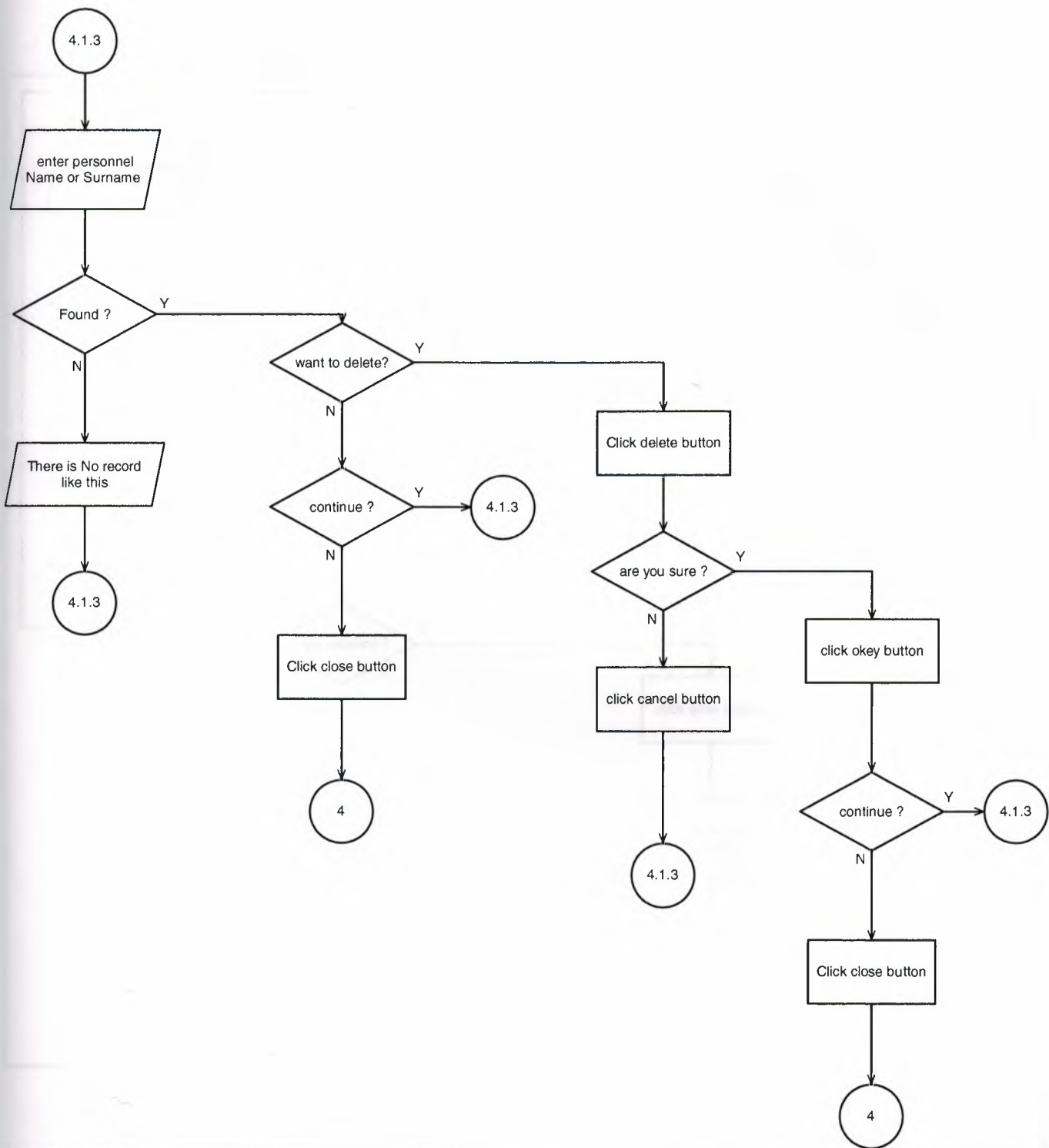
## ADD PERSONNEL



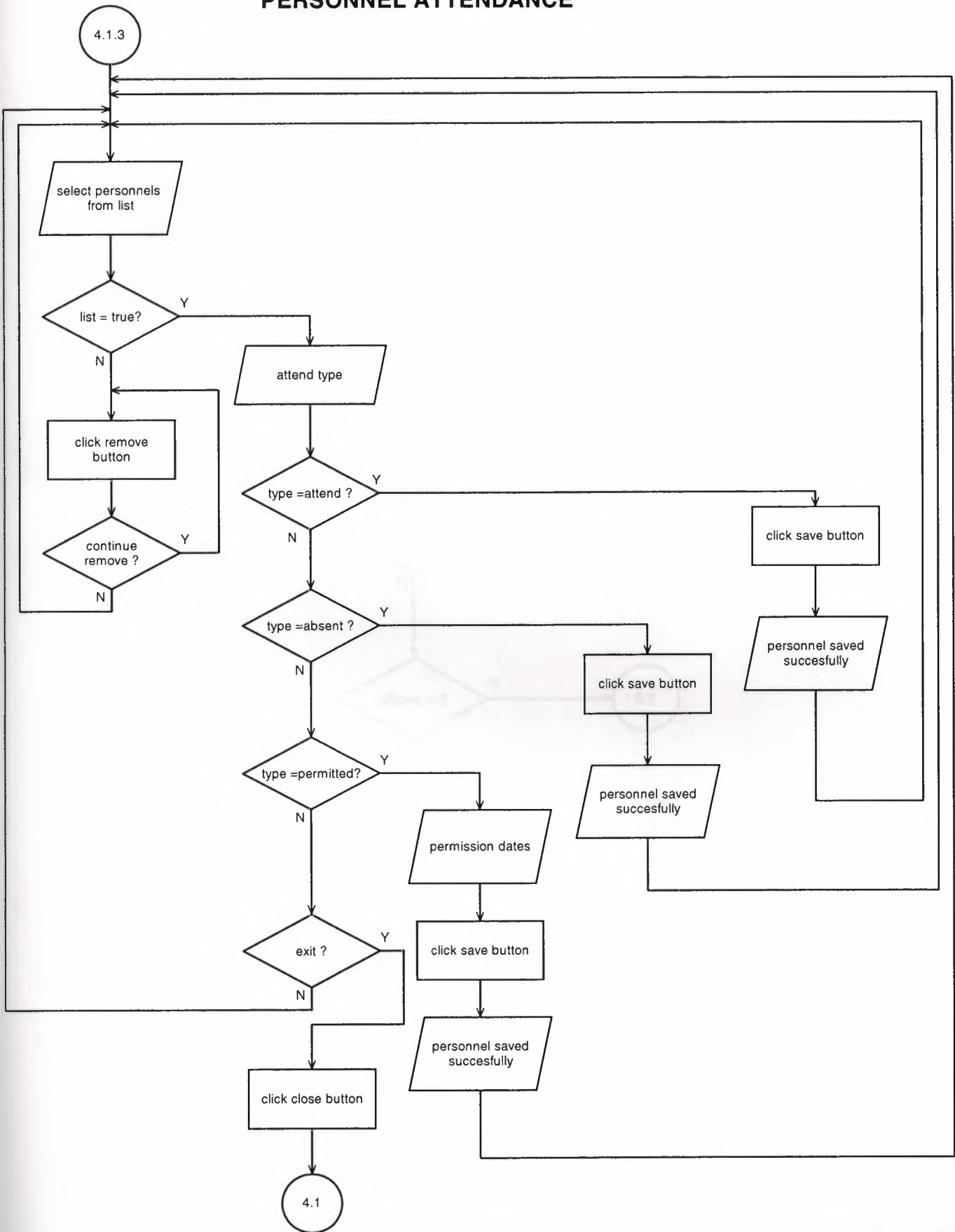
## SEARCH / UPDATE



## DELETE PERSONNEL

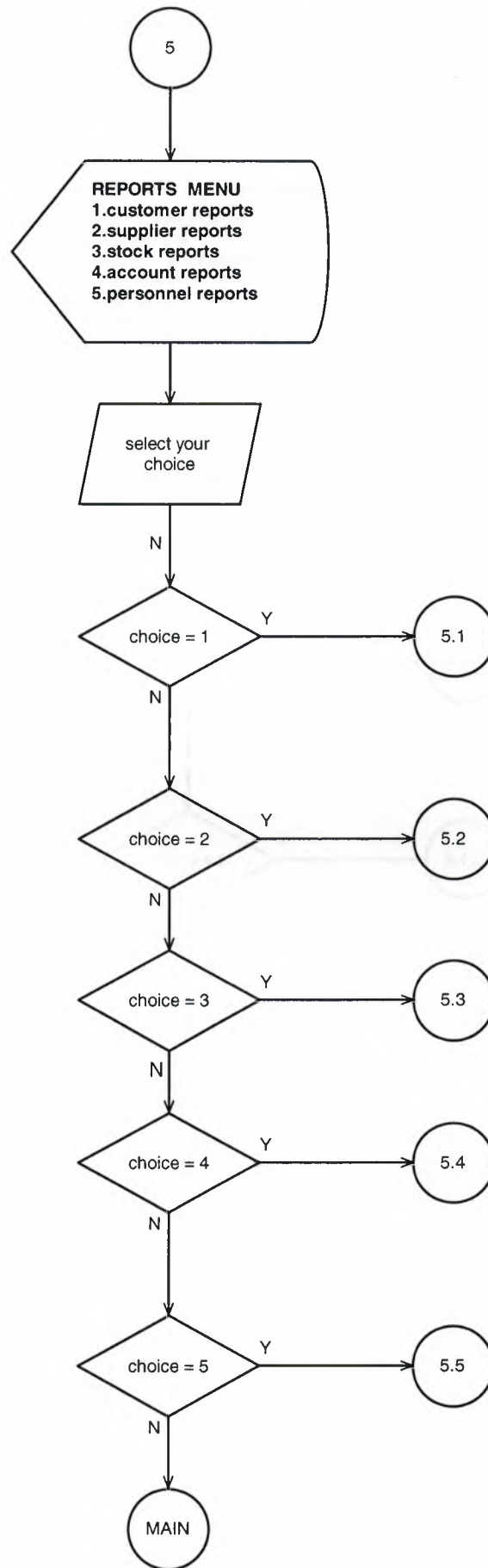


## PERSONNEL ATTENDANCE

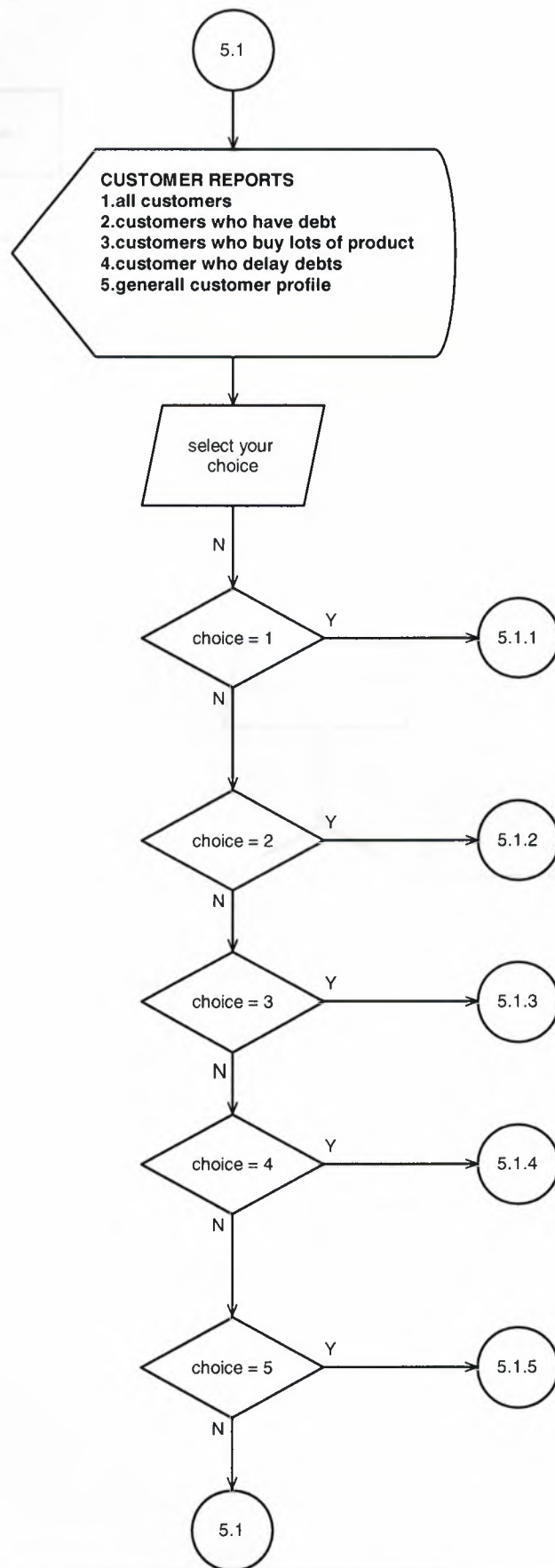


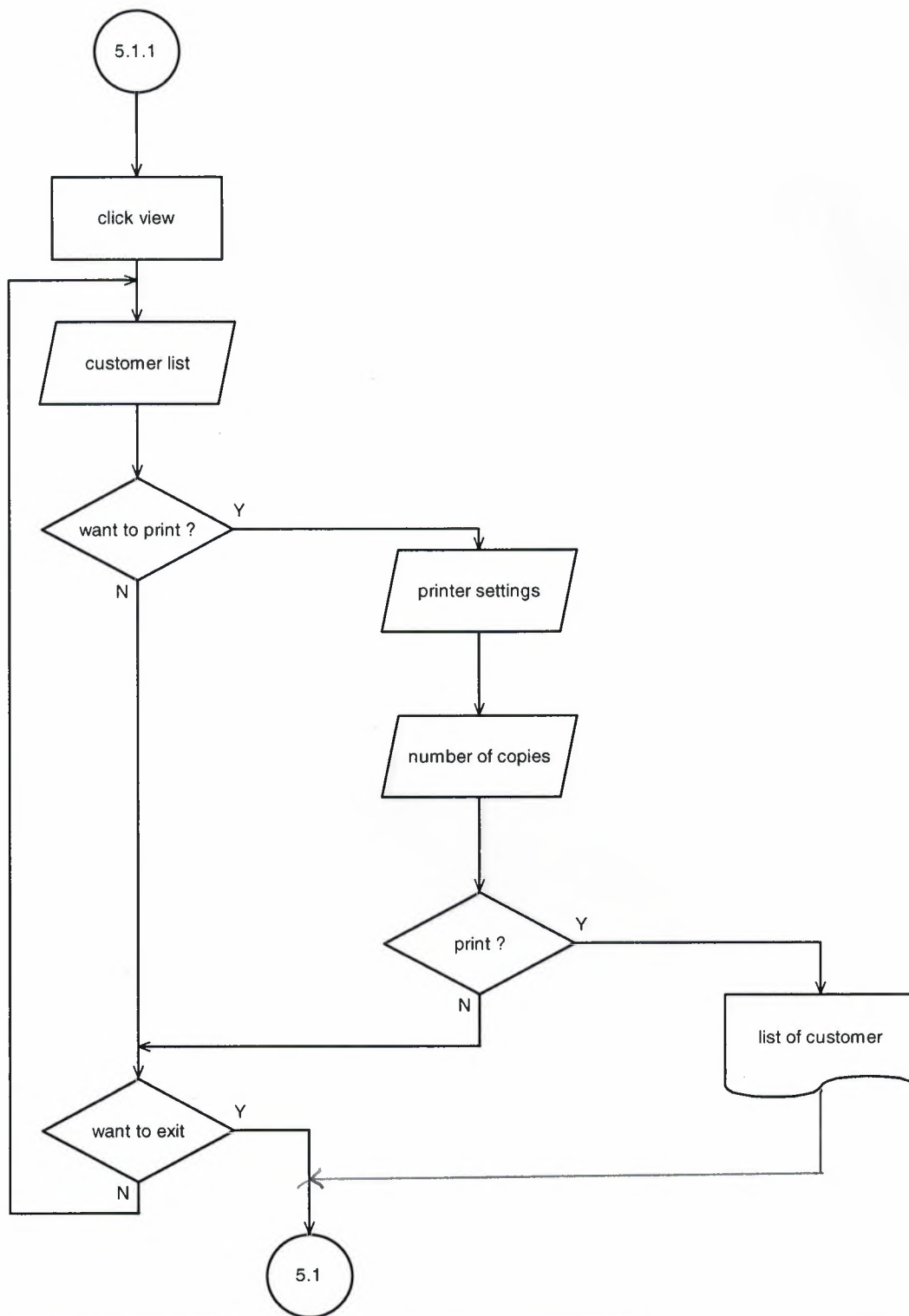


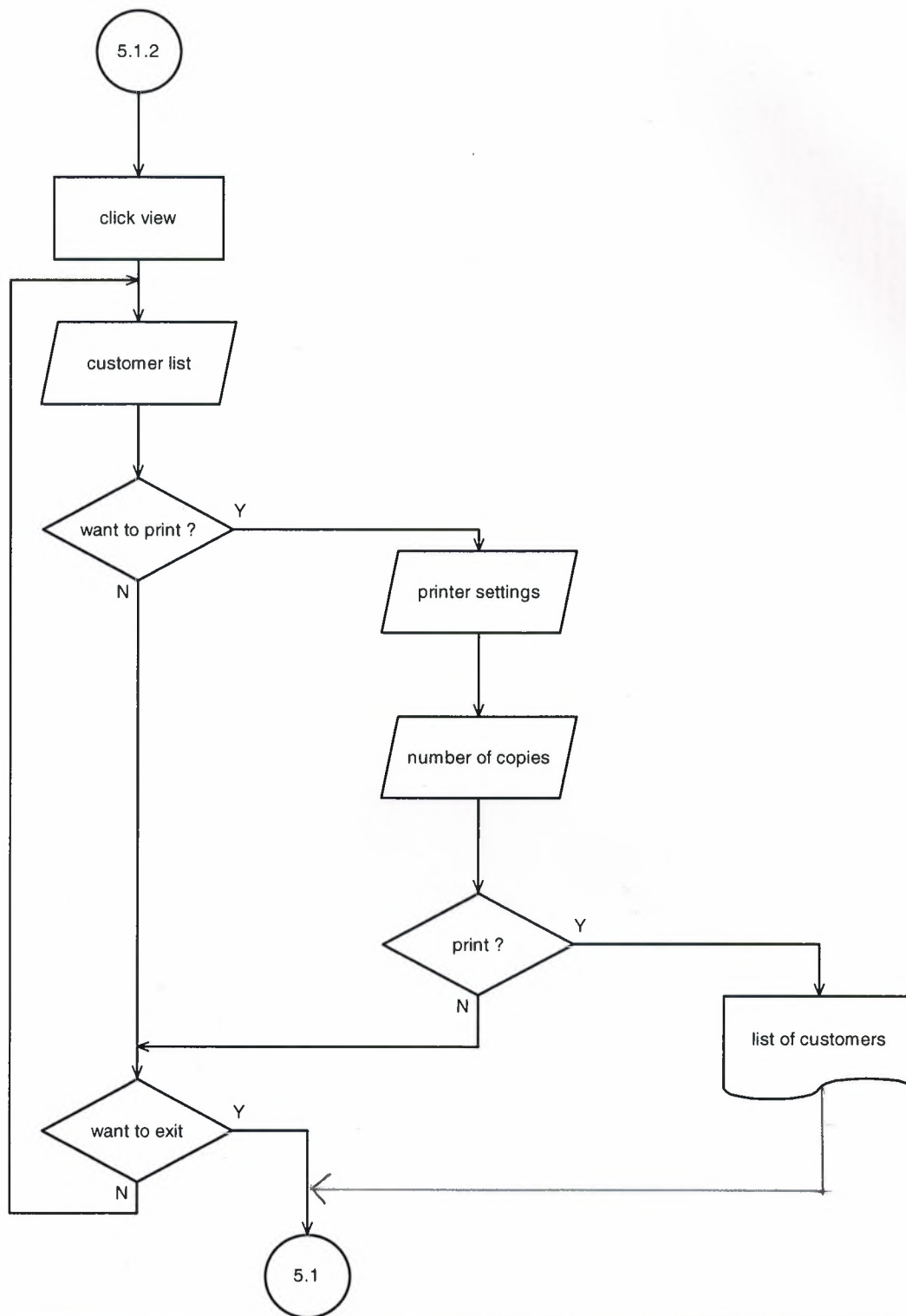
## REPORTS

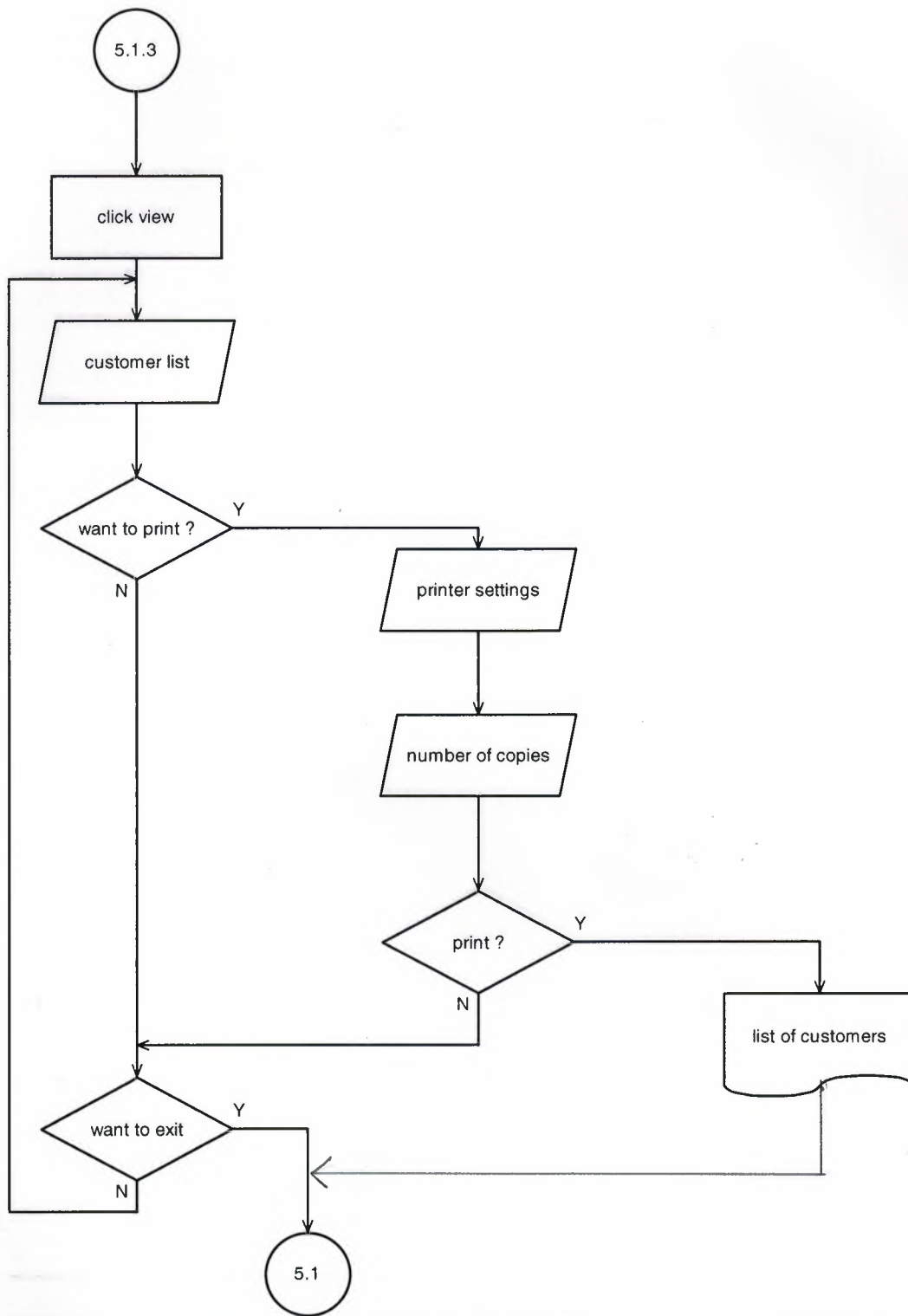


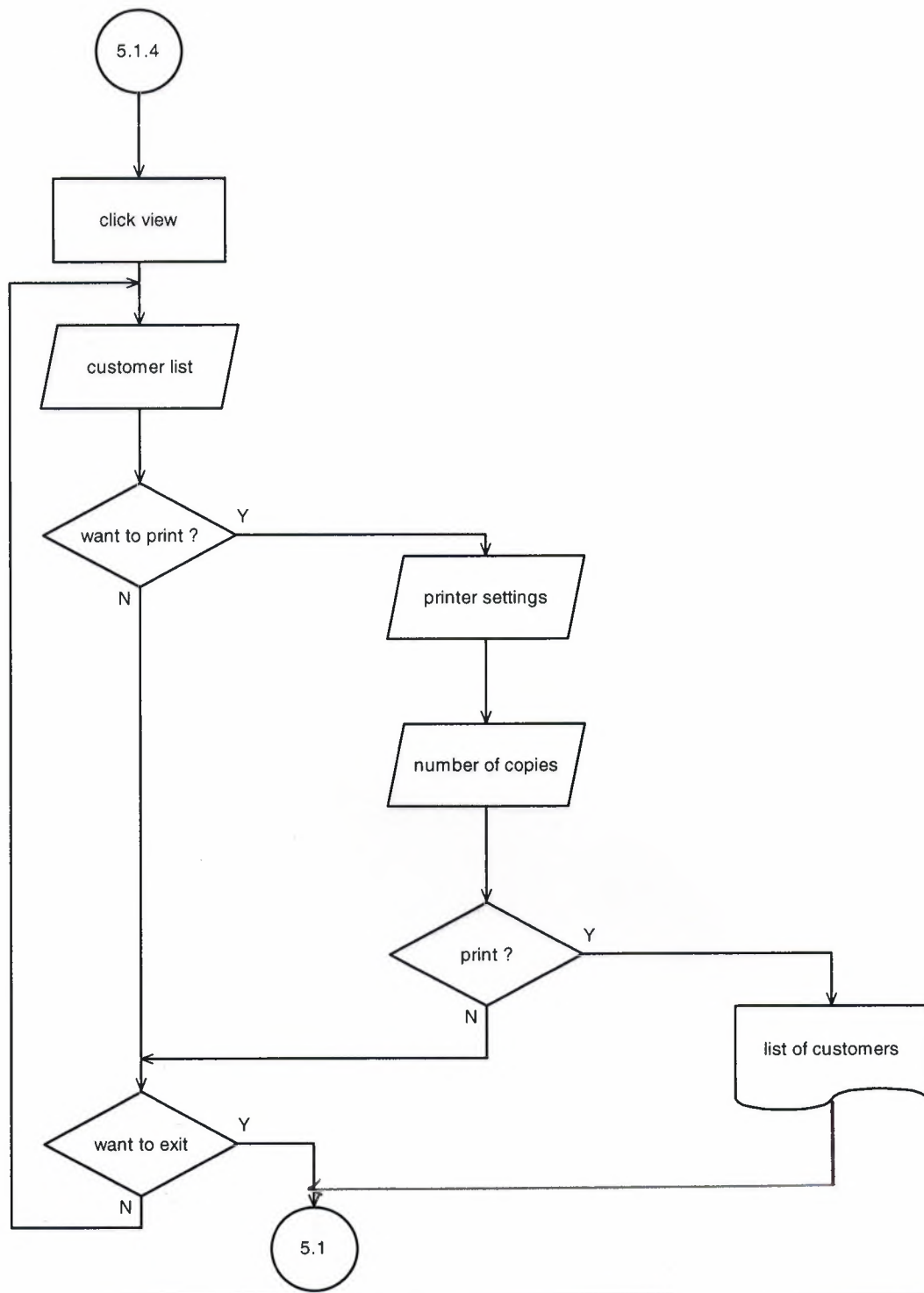
## CUSTOMER REPORTS



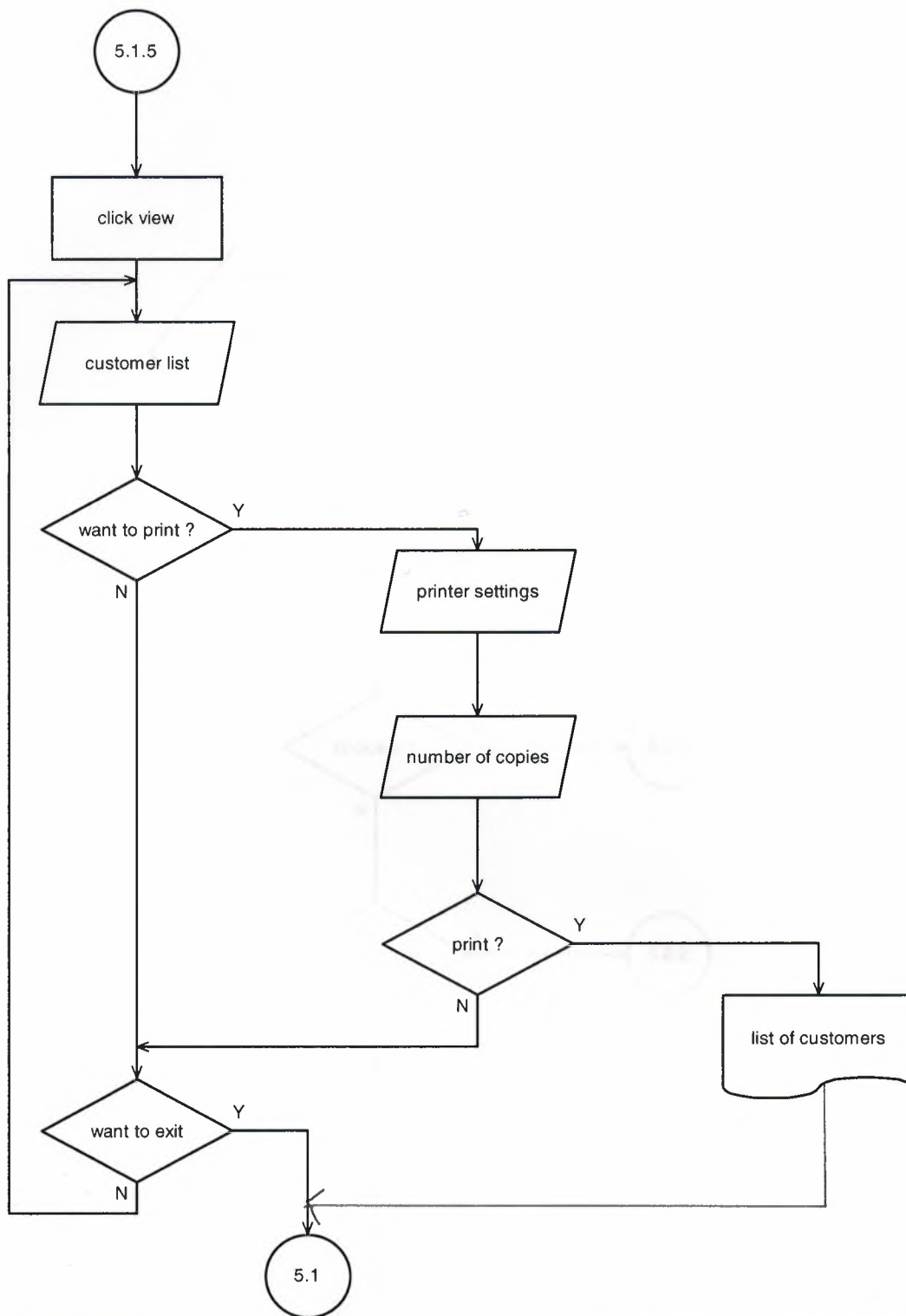




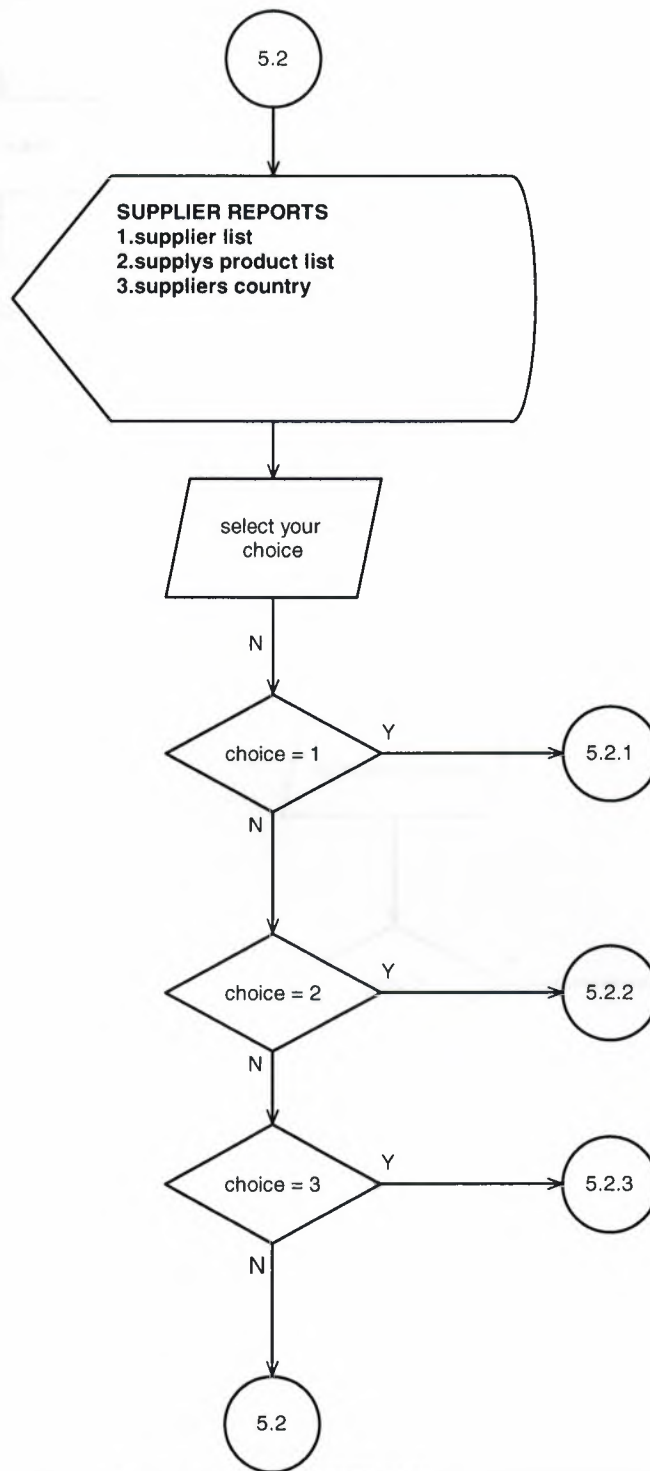


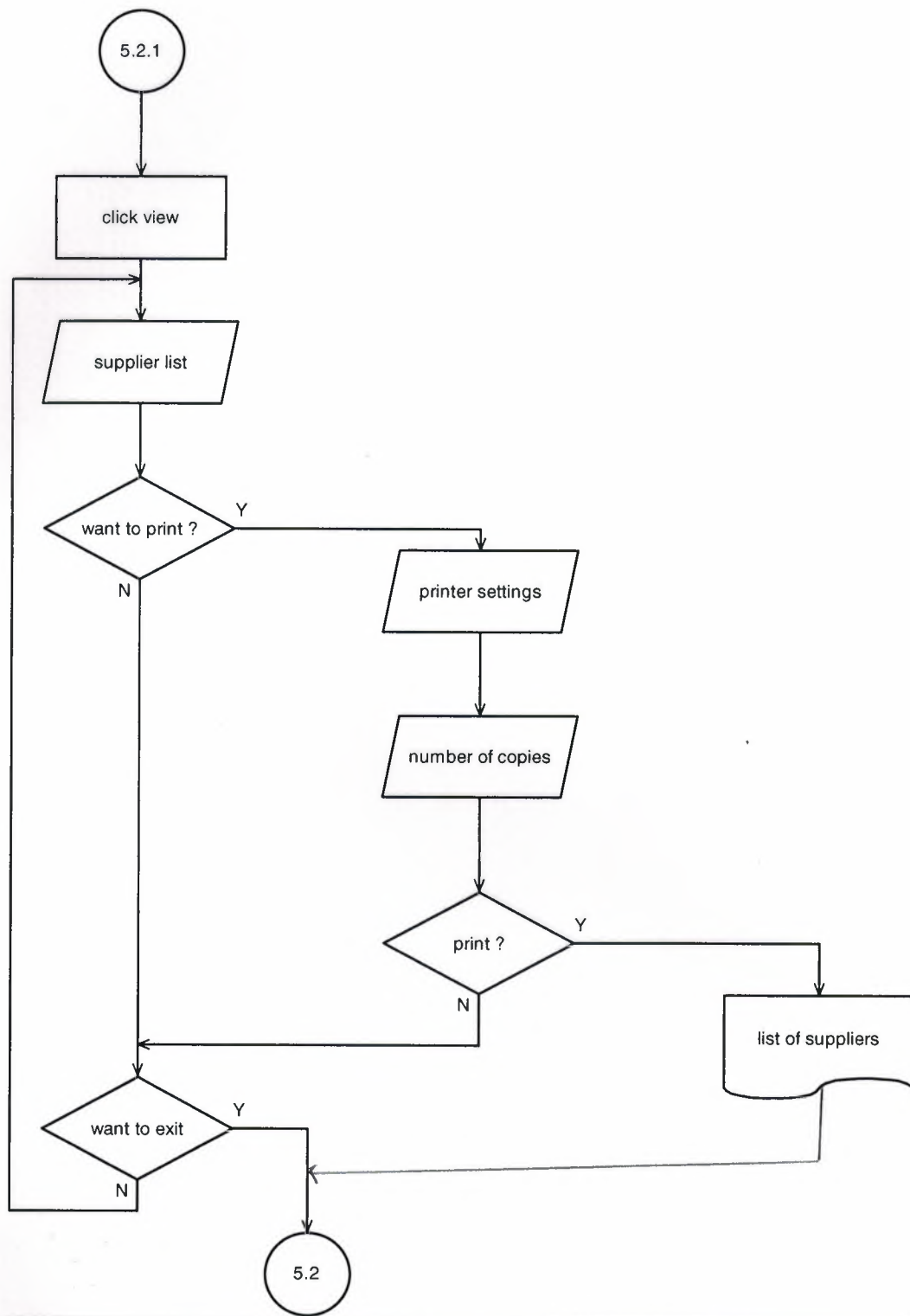


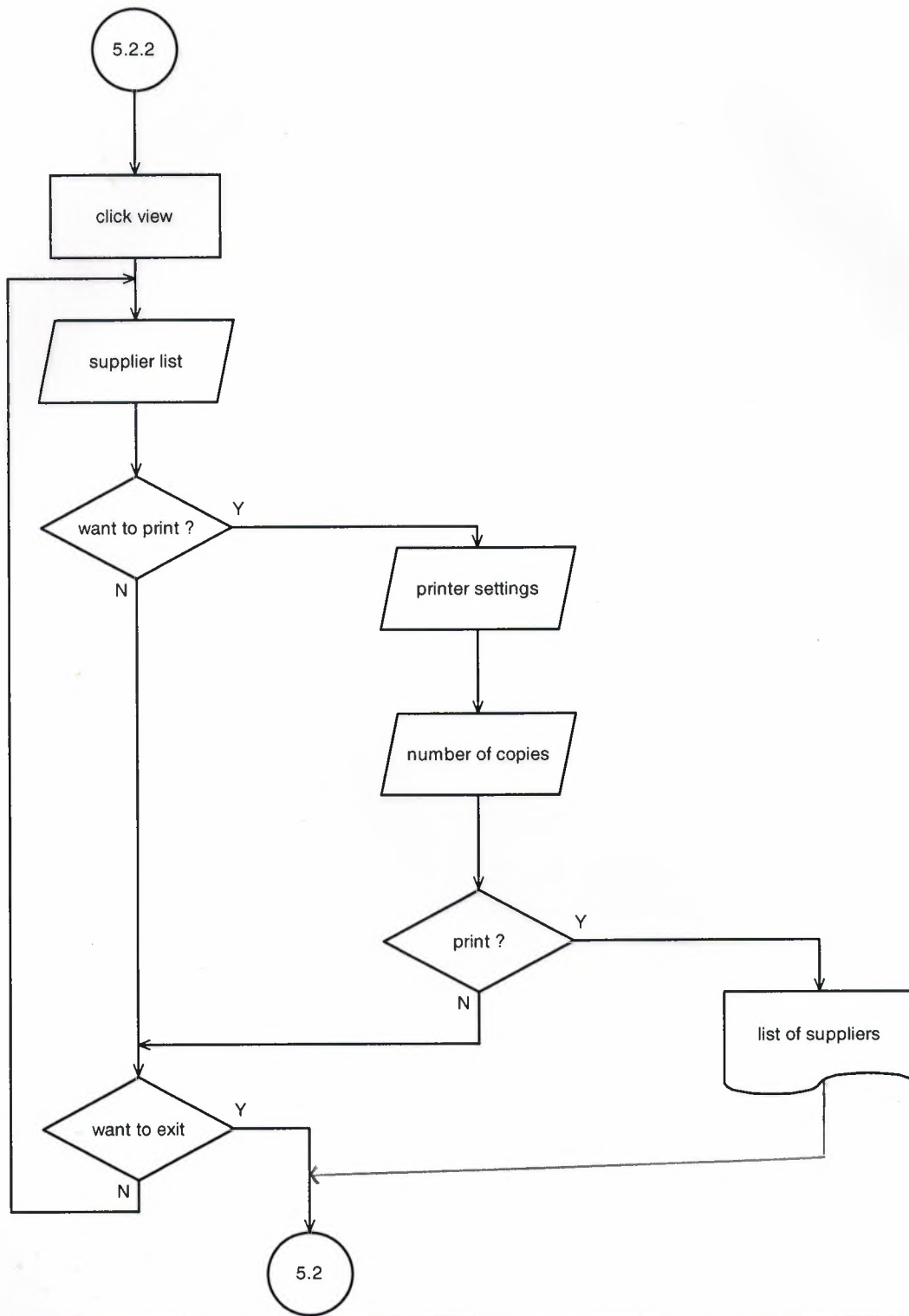


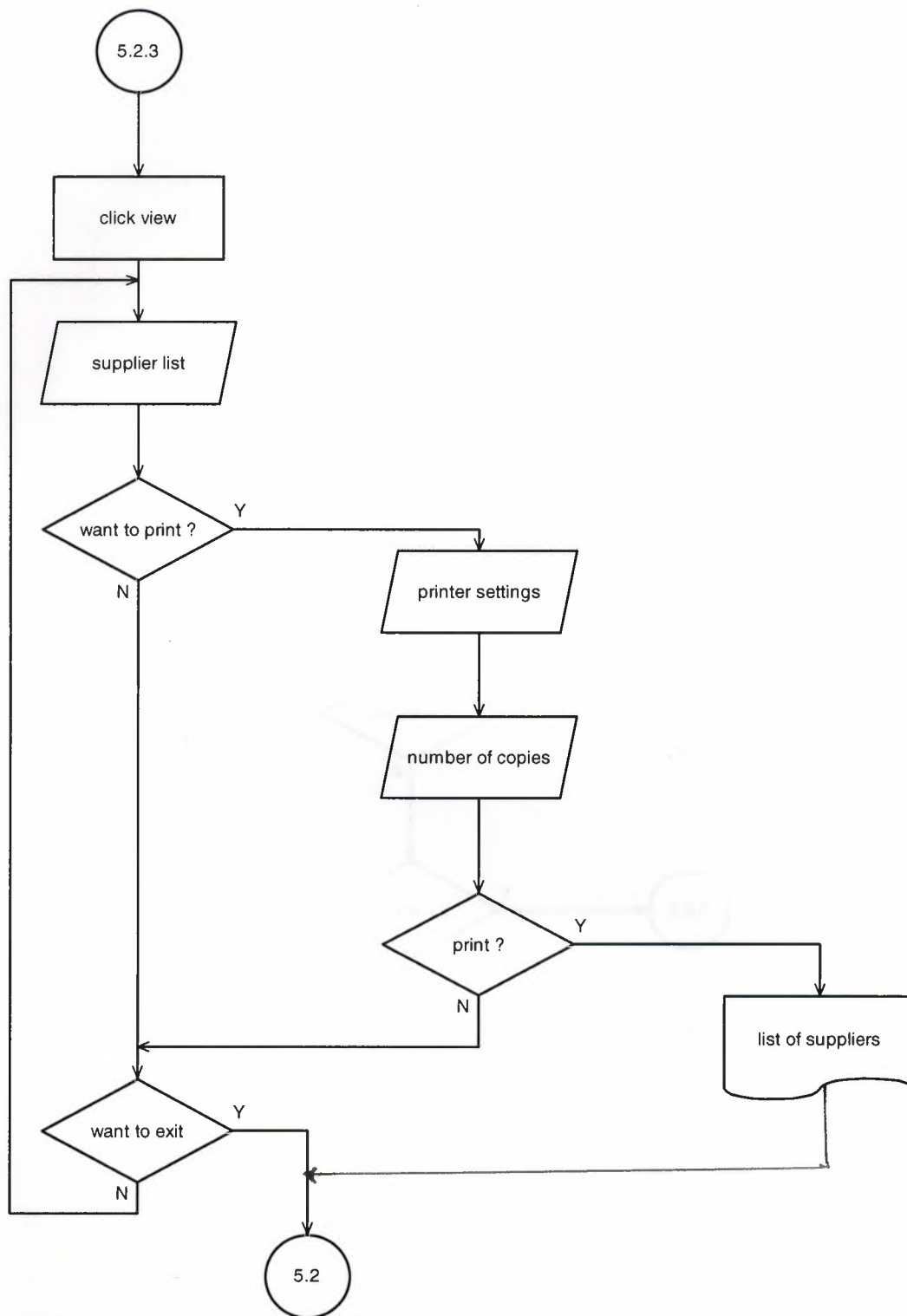


## SUPPLIER REPORTS

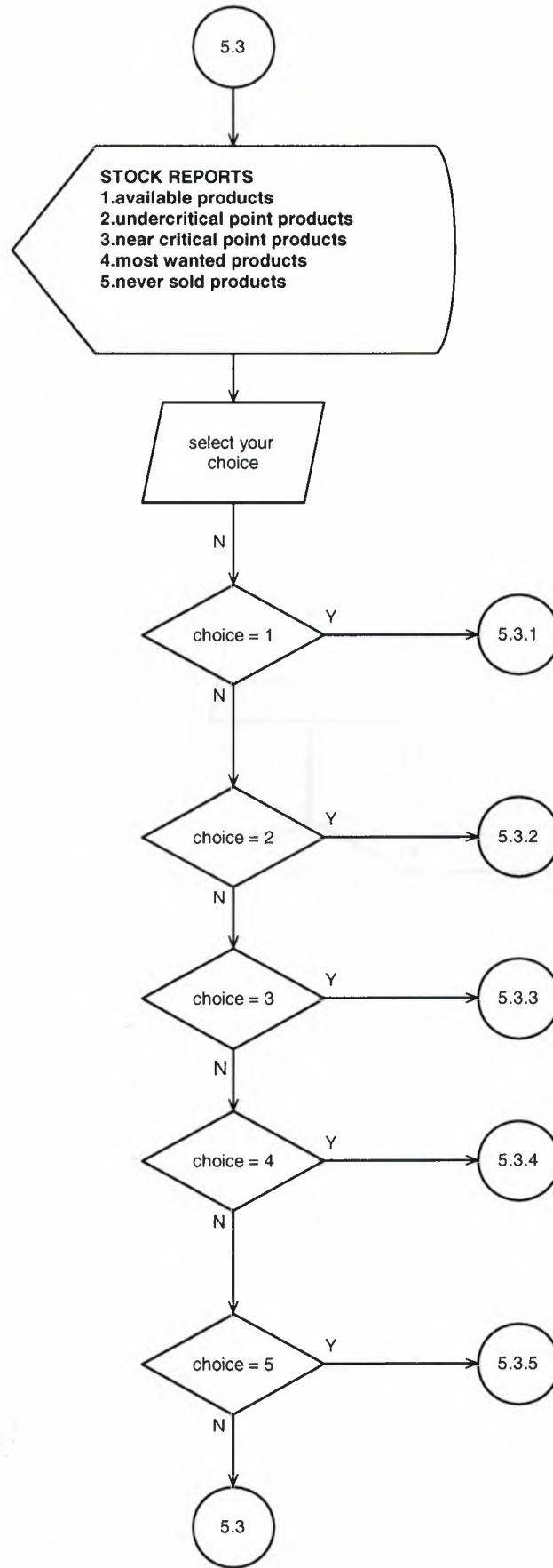




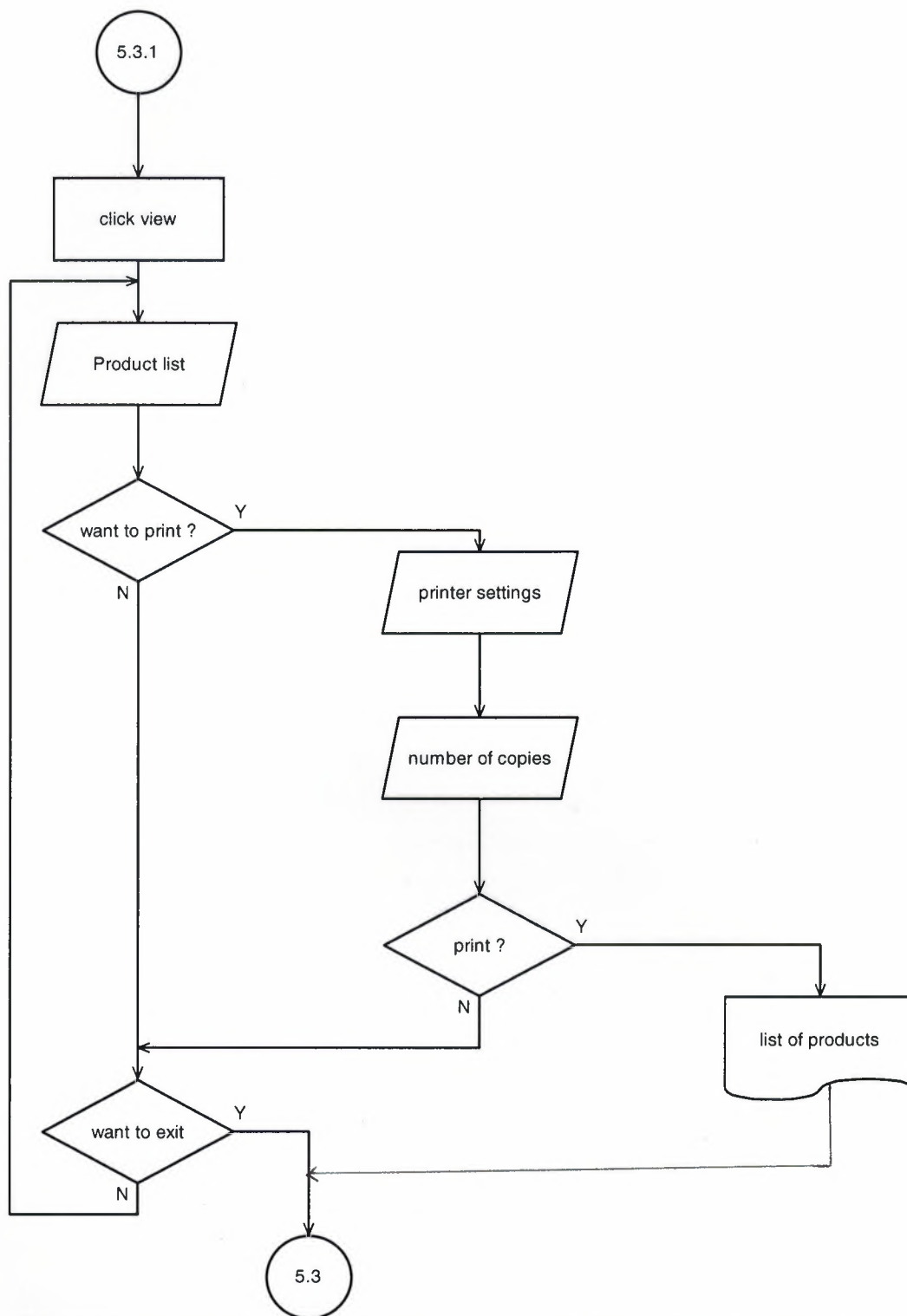


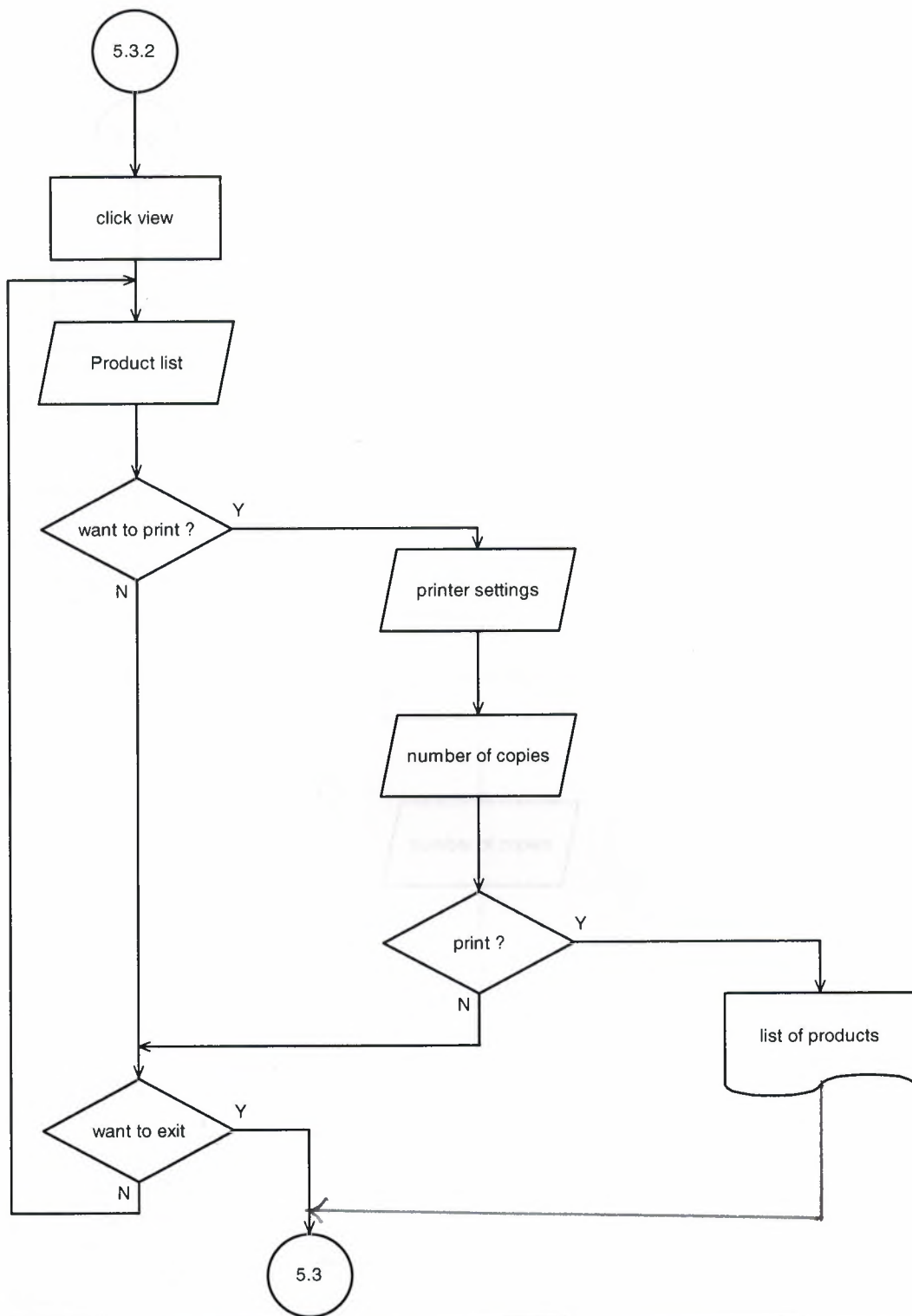


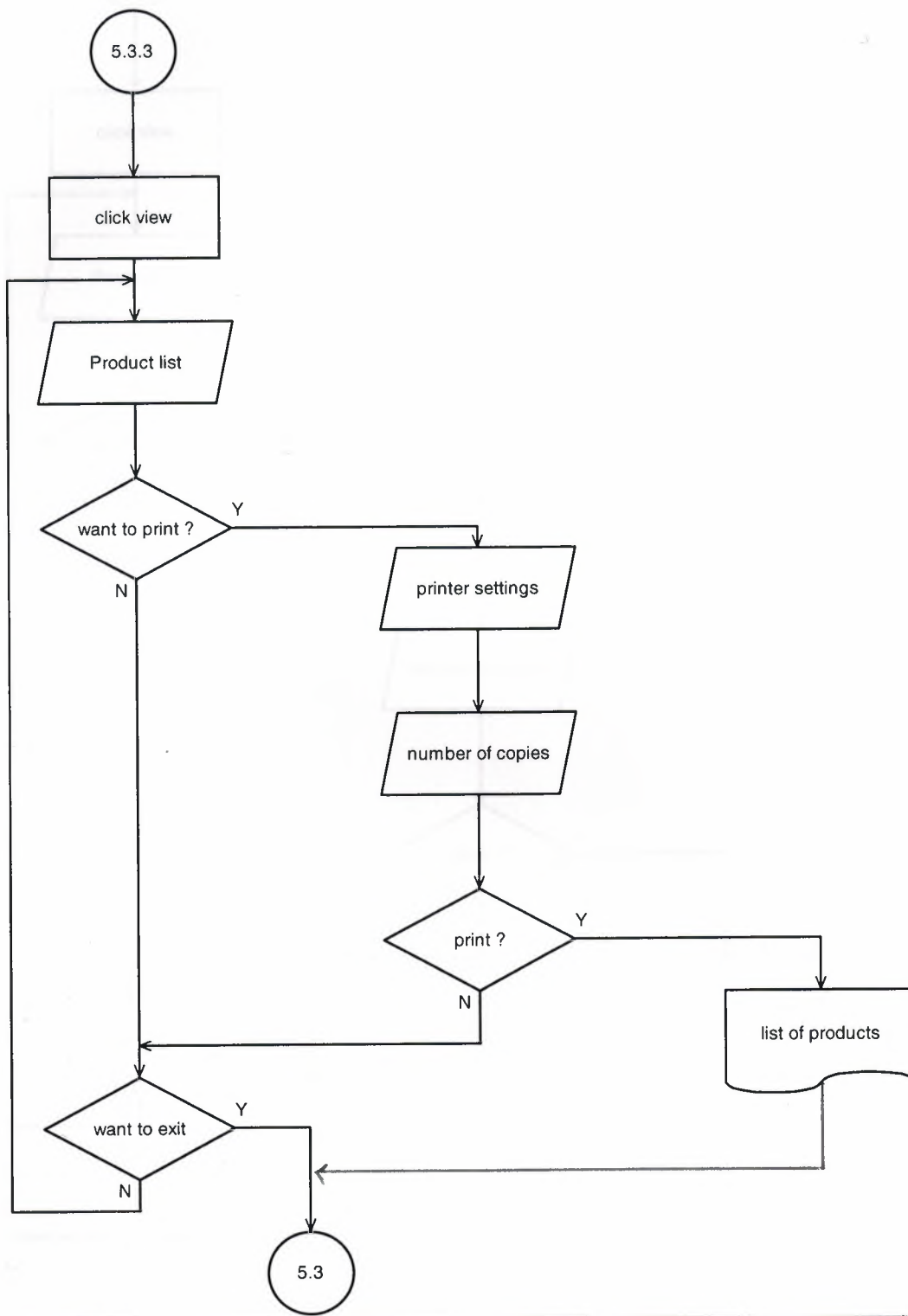
## STOCK REPORTS

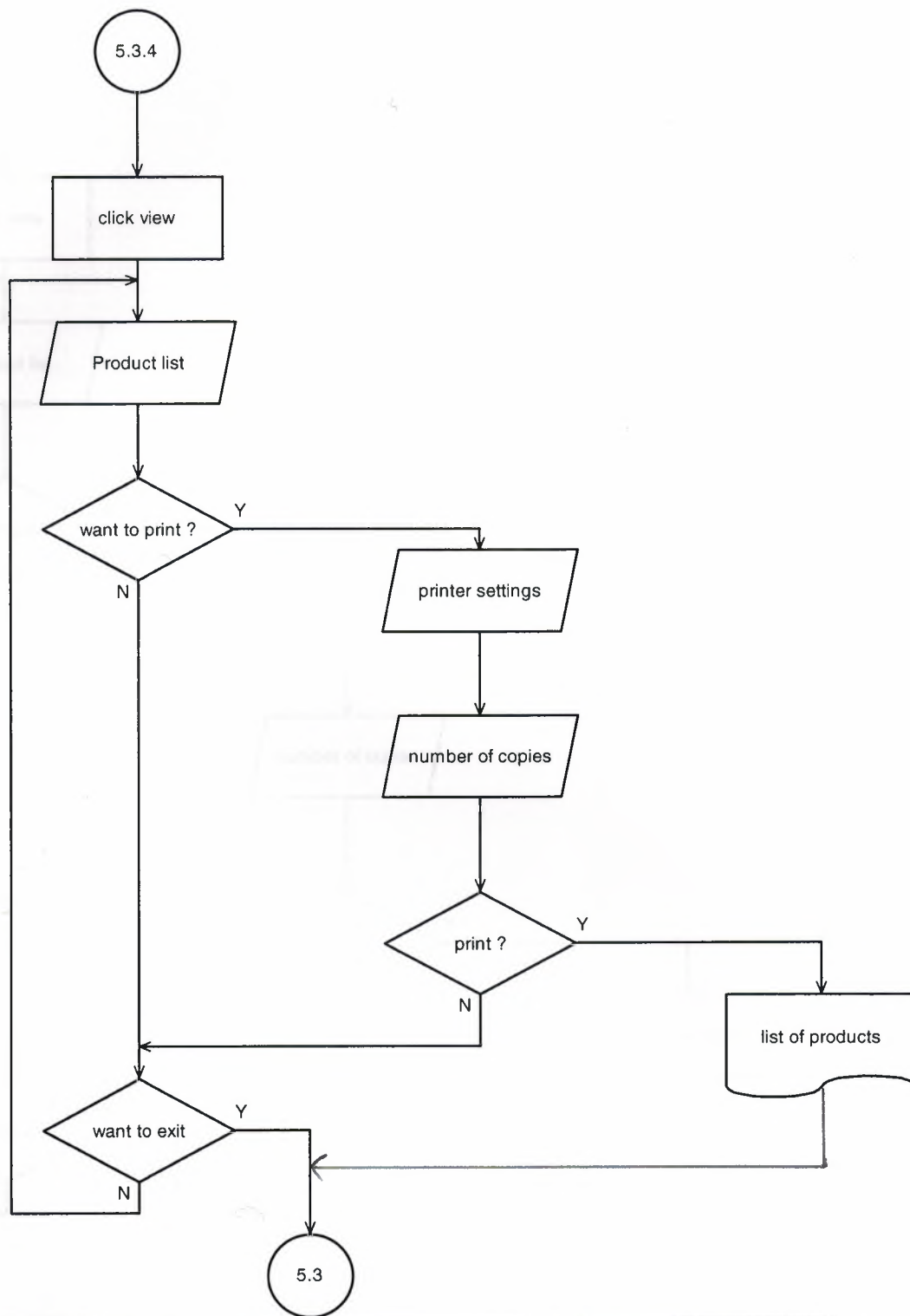


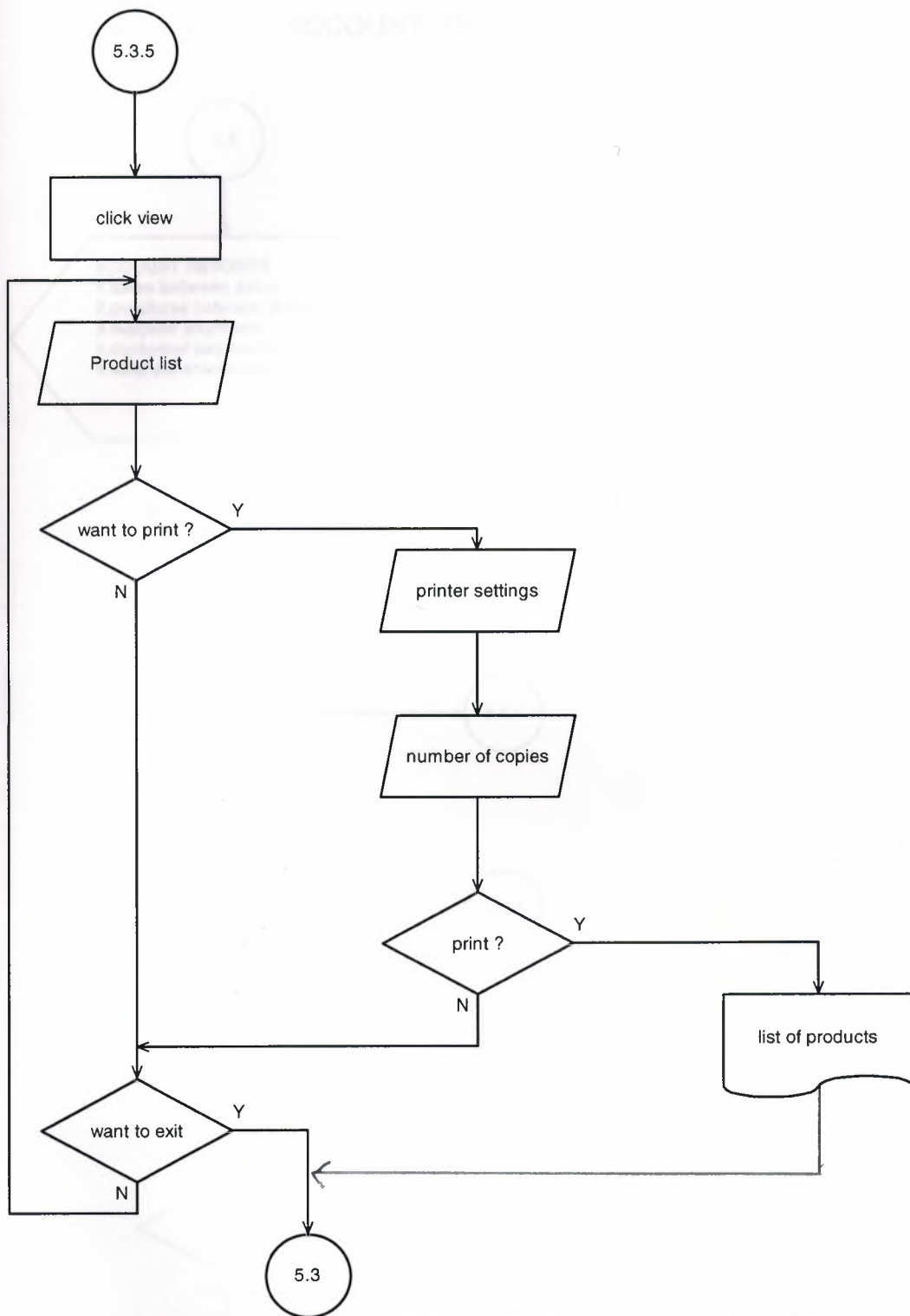




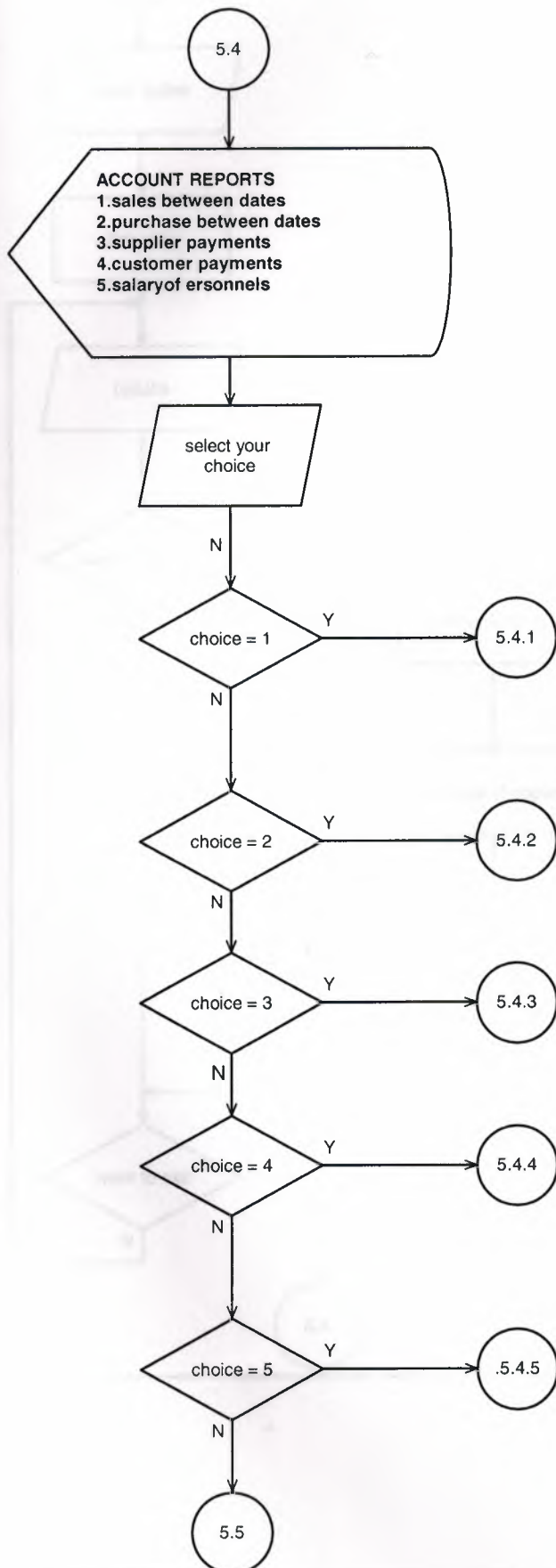




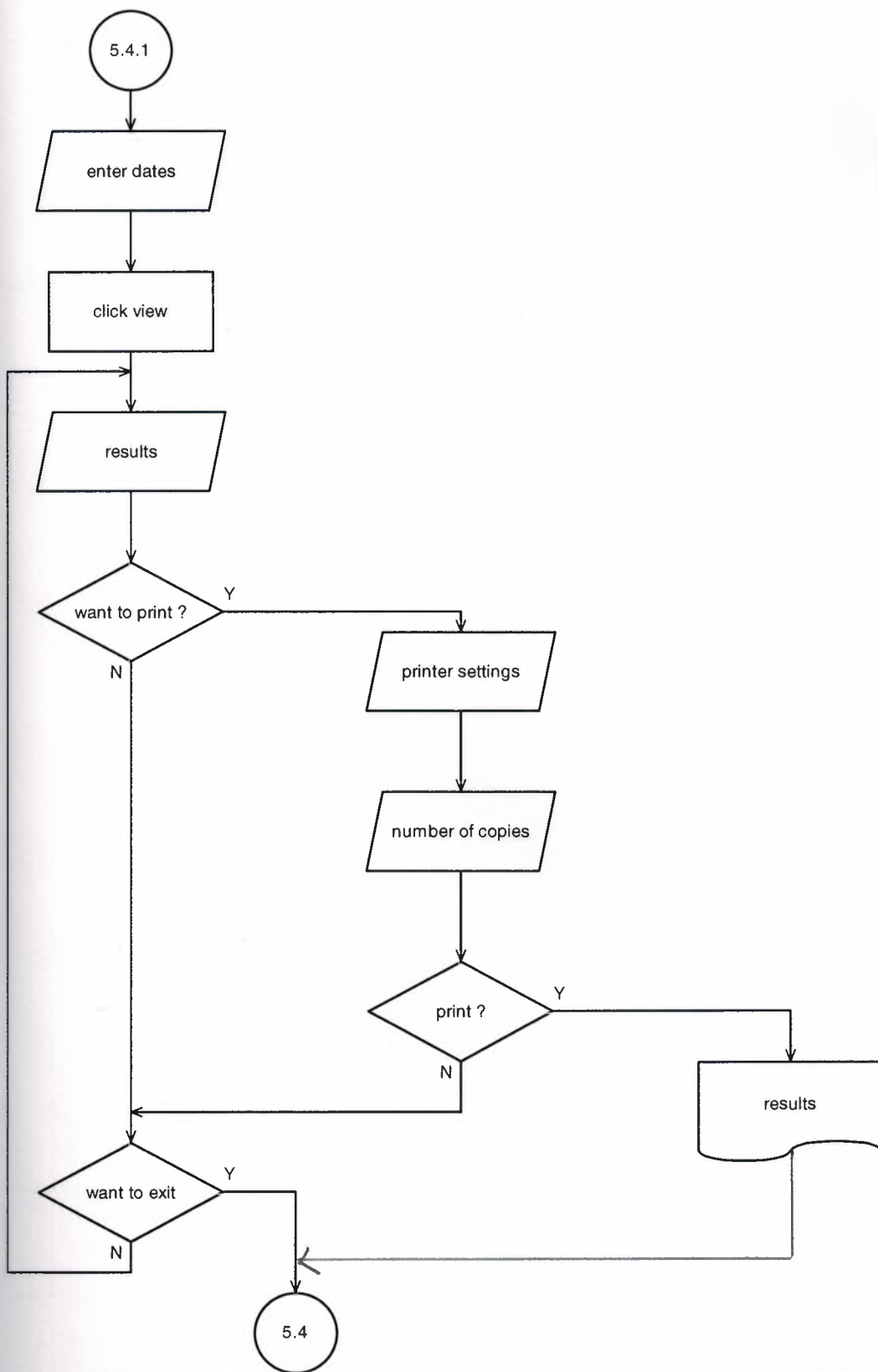


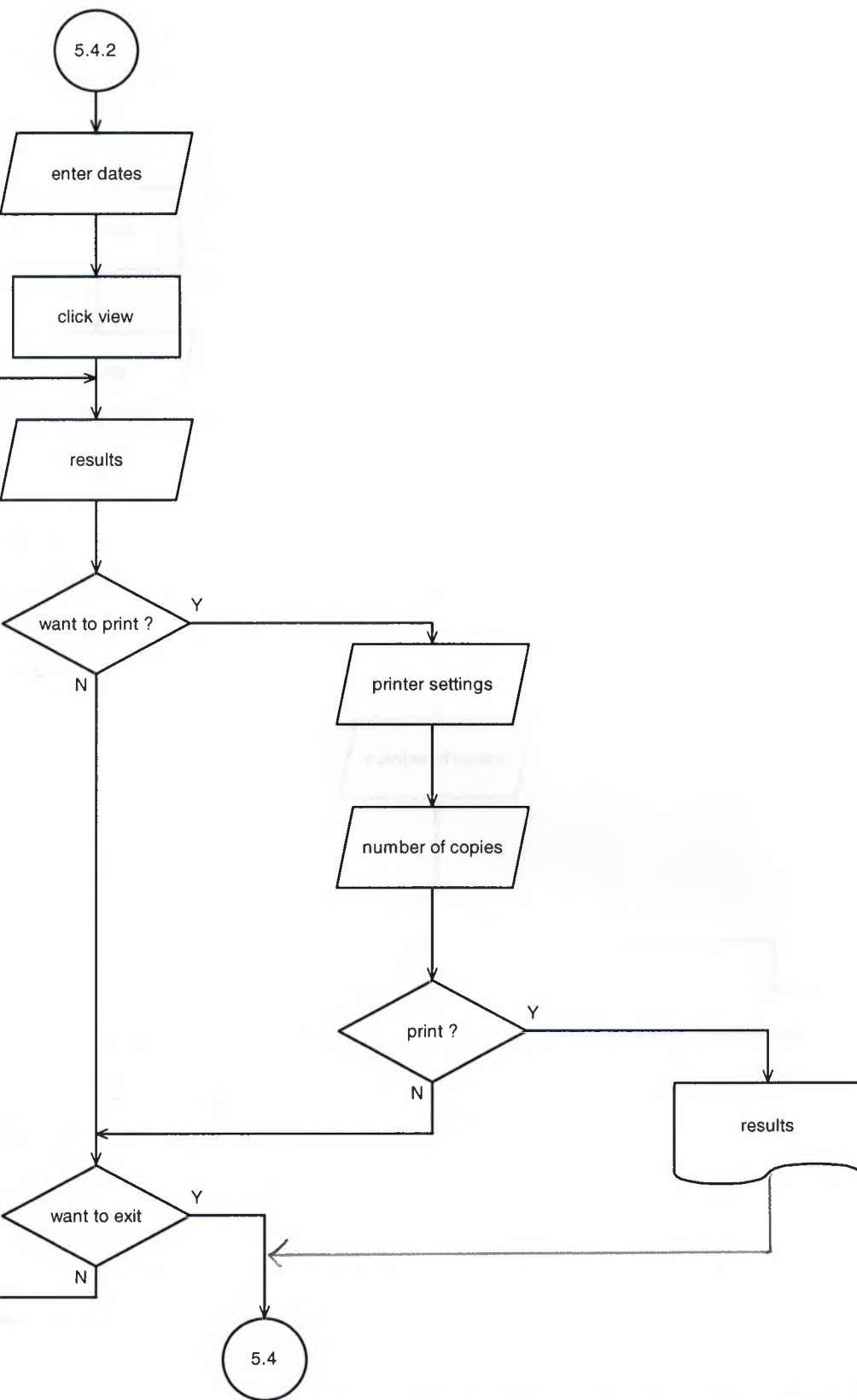


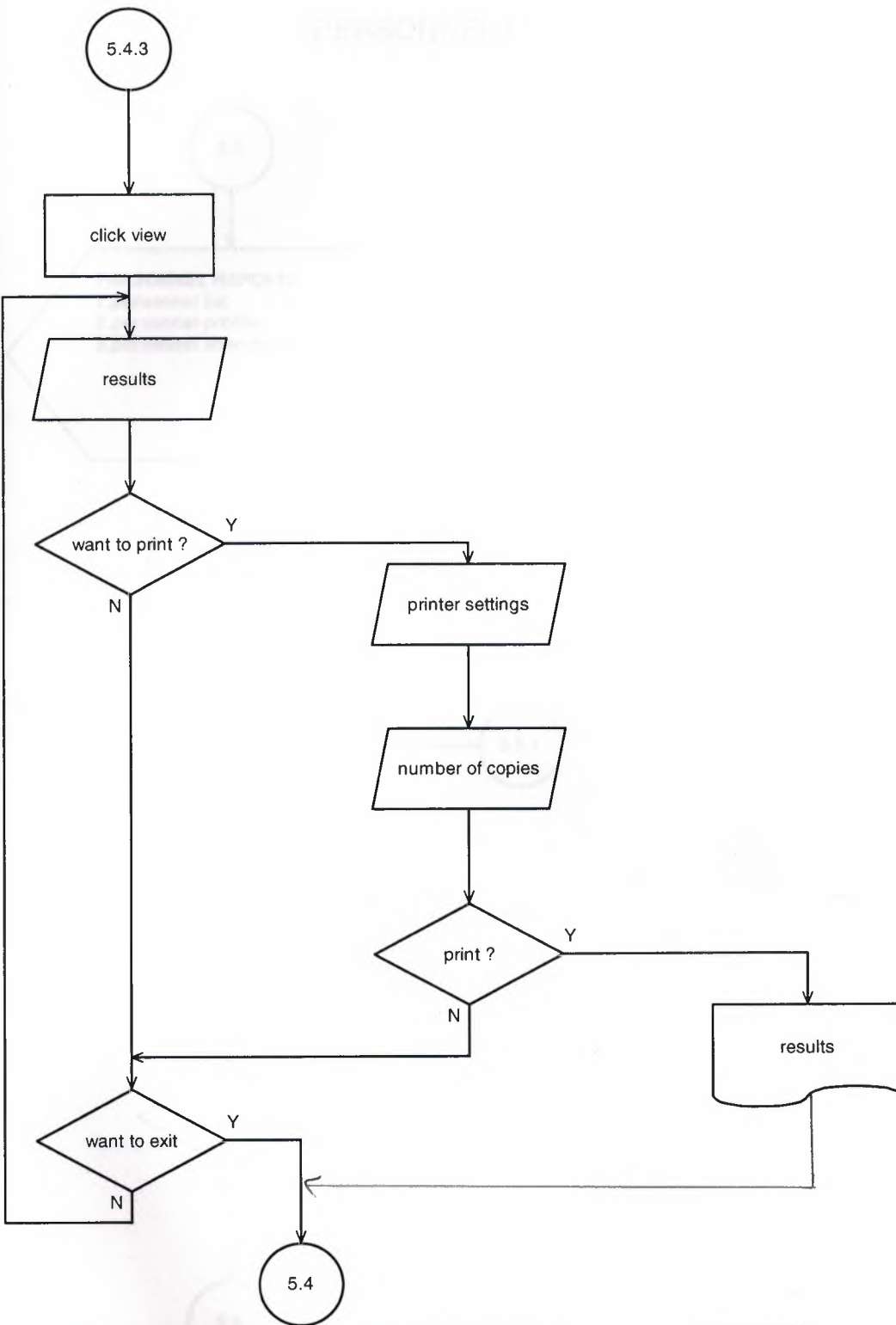
## ACCOUNT REPORTS



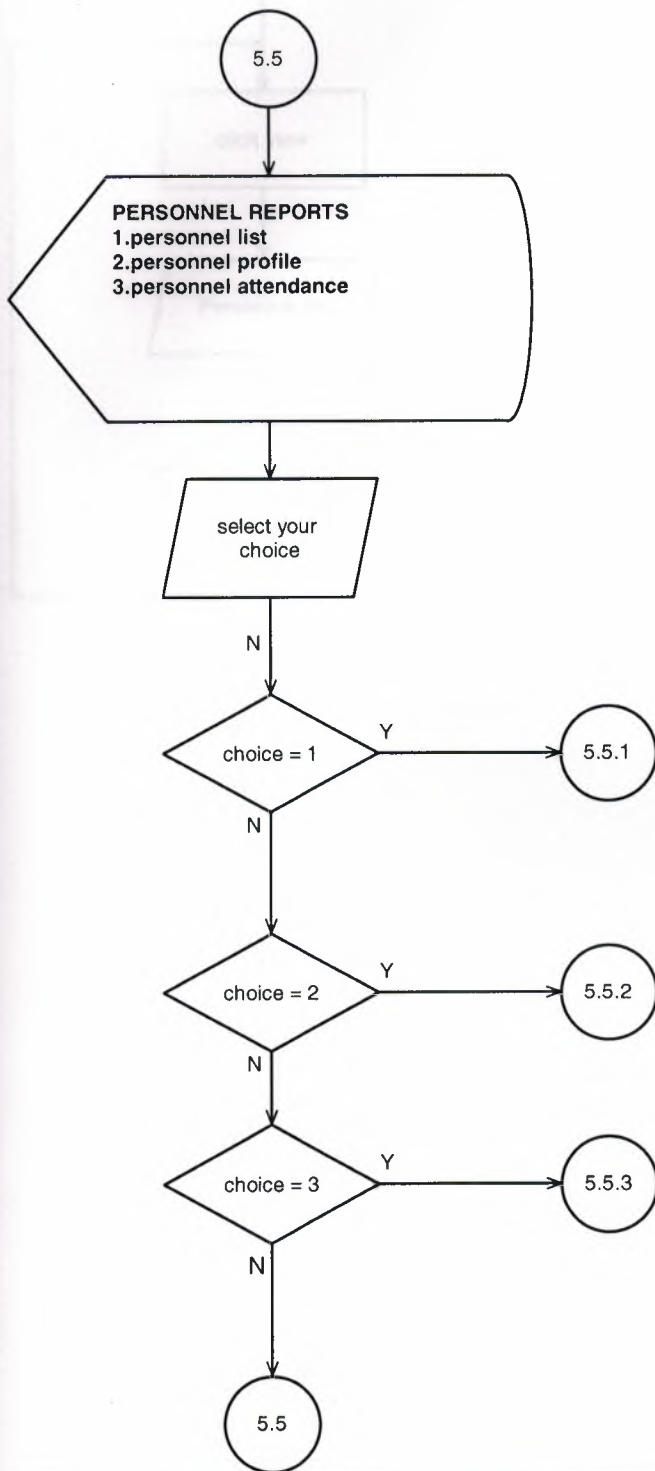


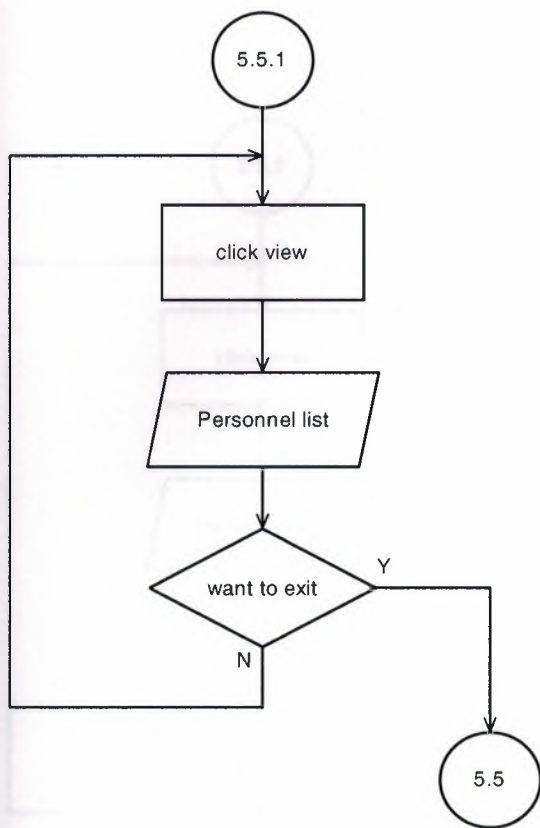


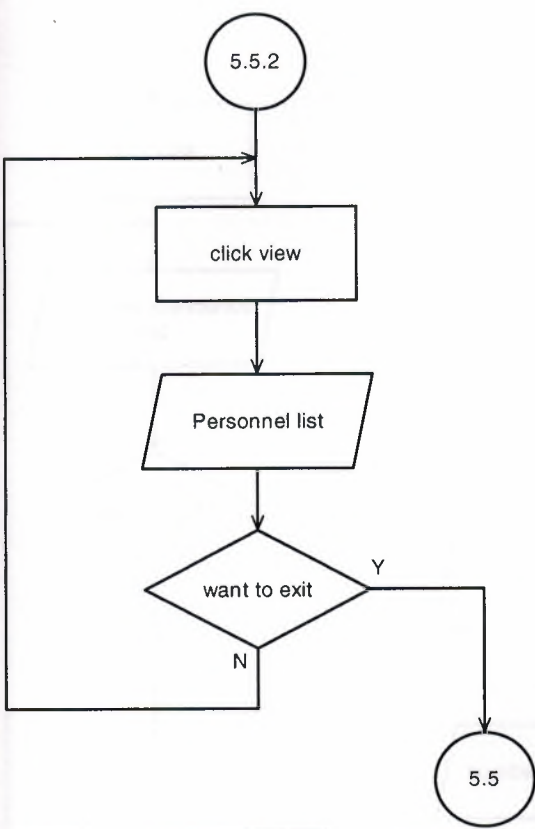




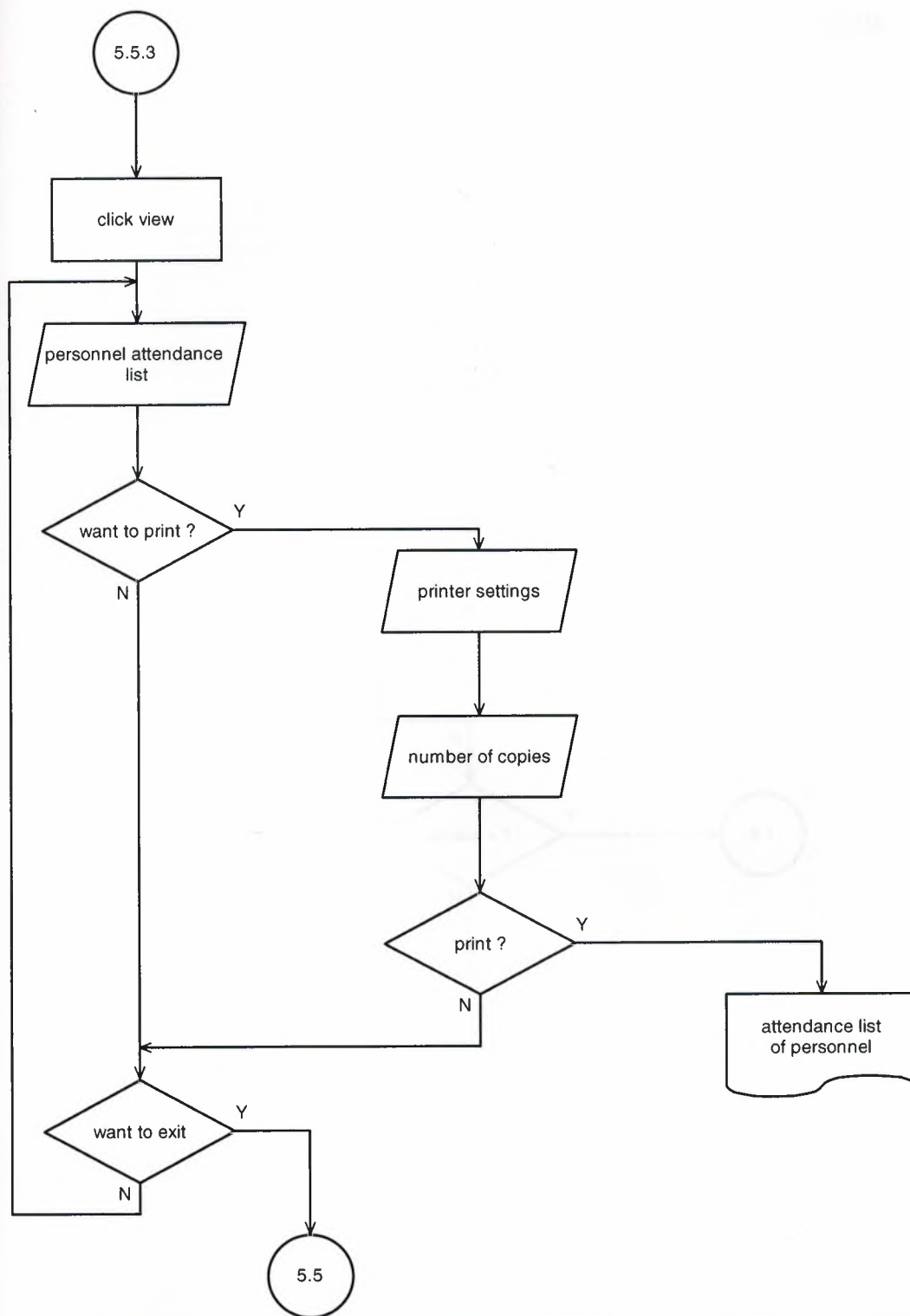
## PERSONNEL REPORTS



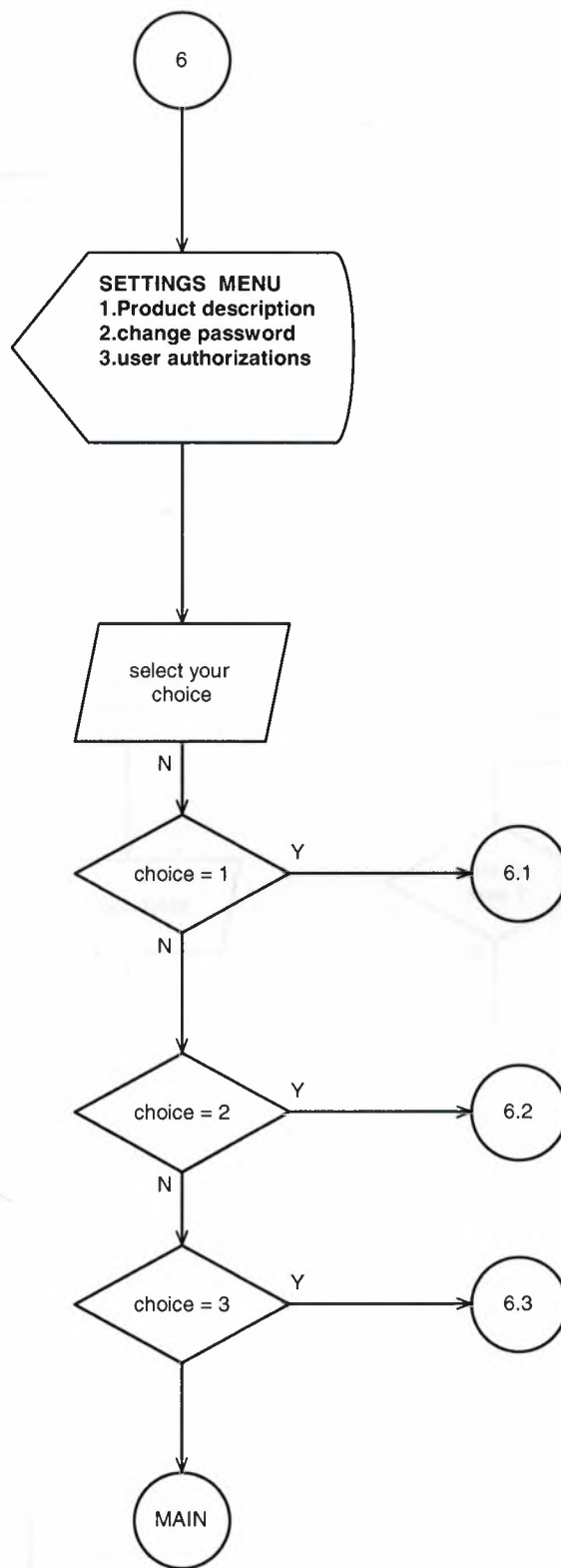




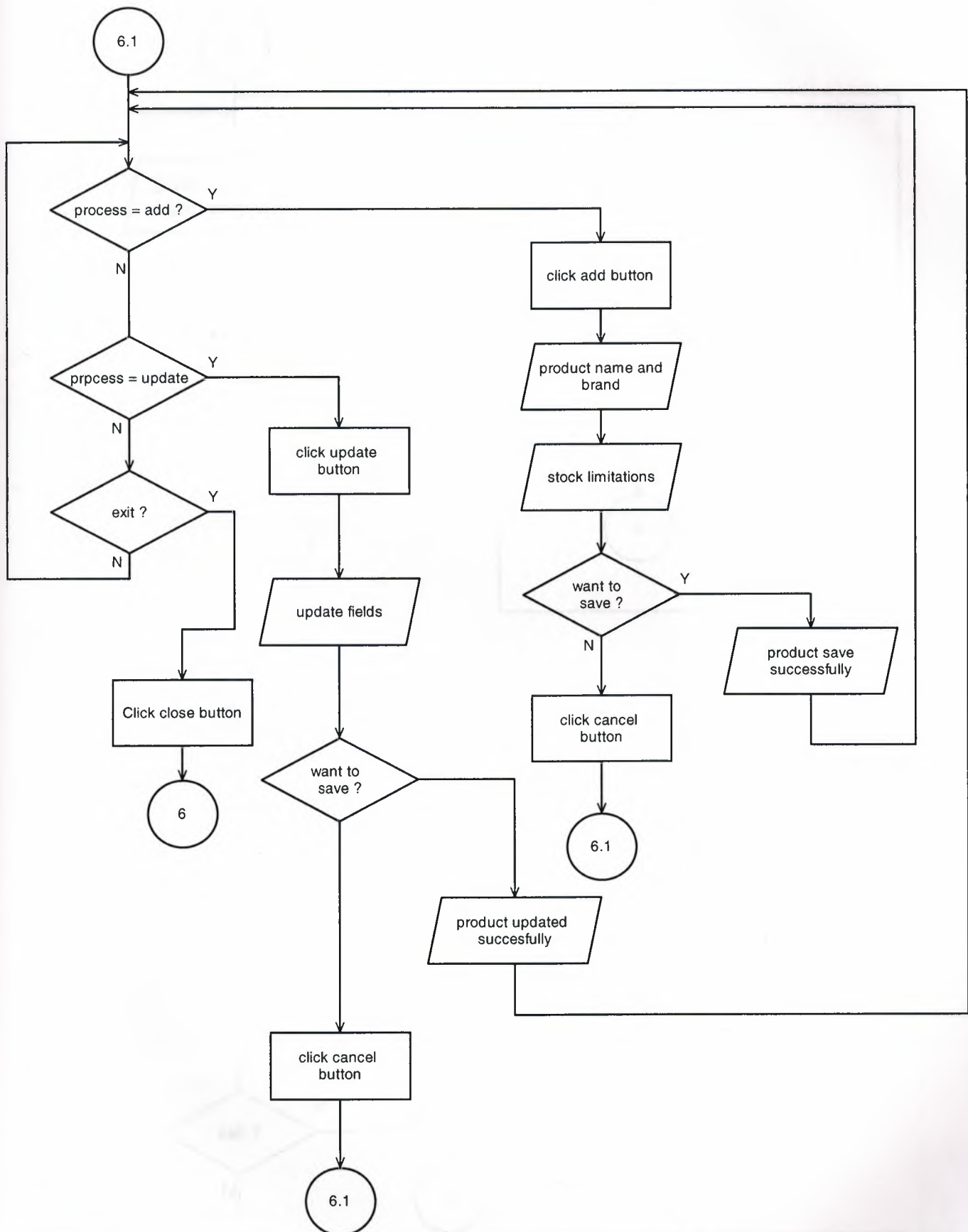




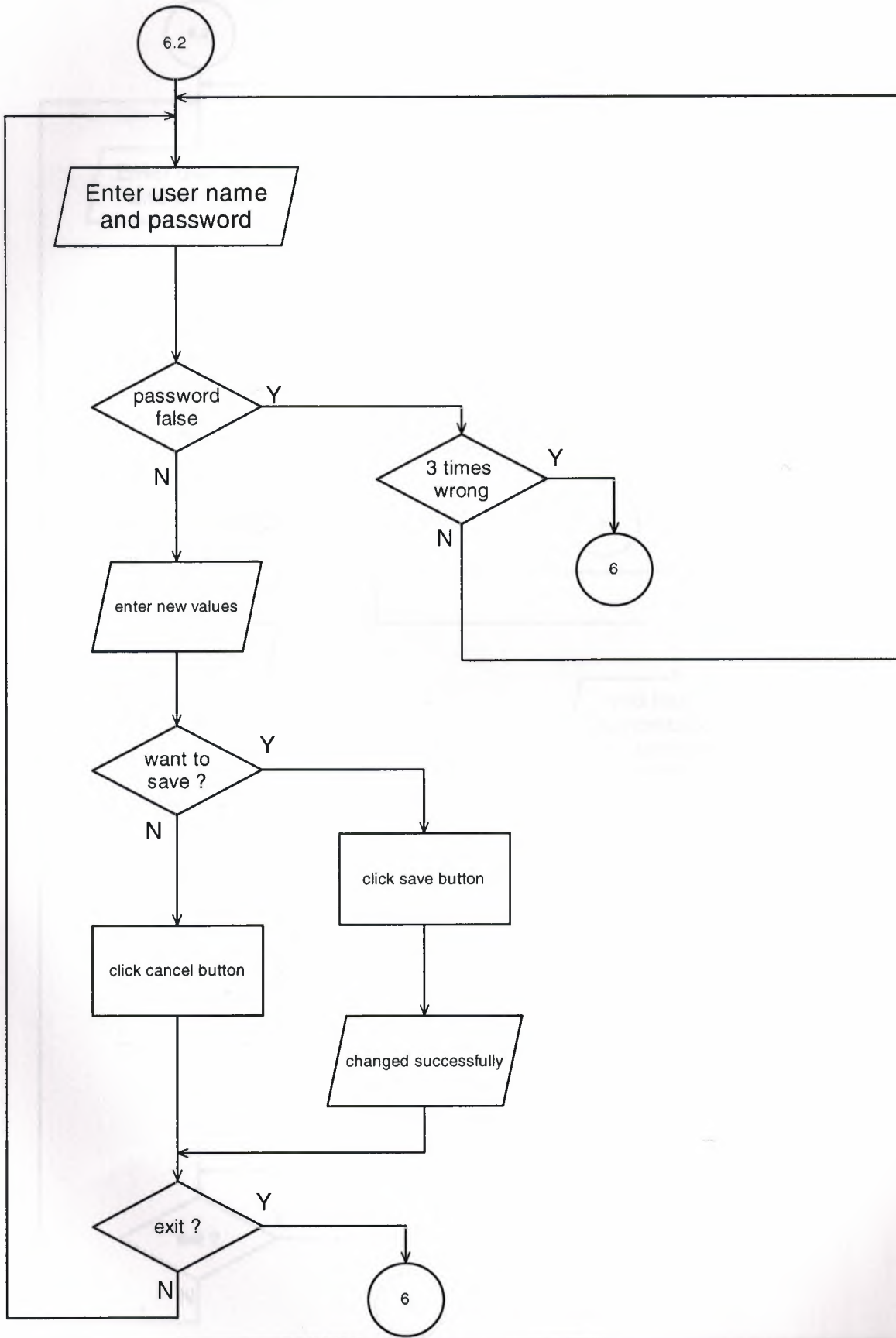
## SETTINGS



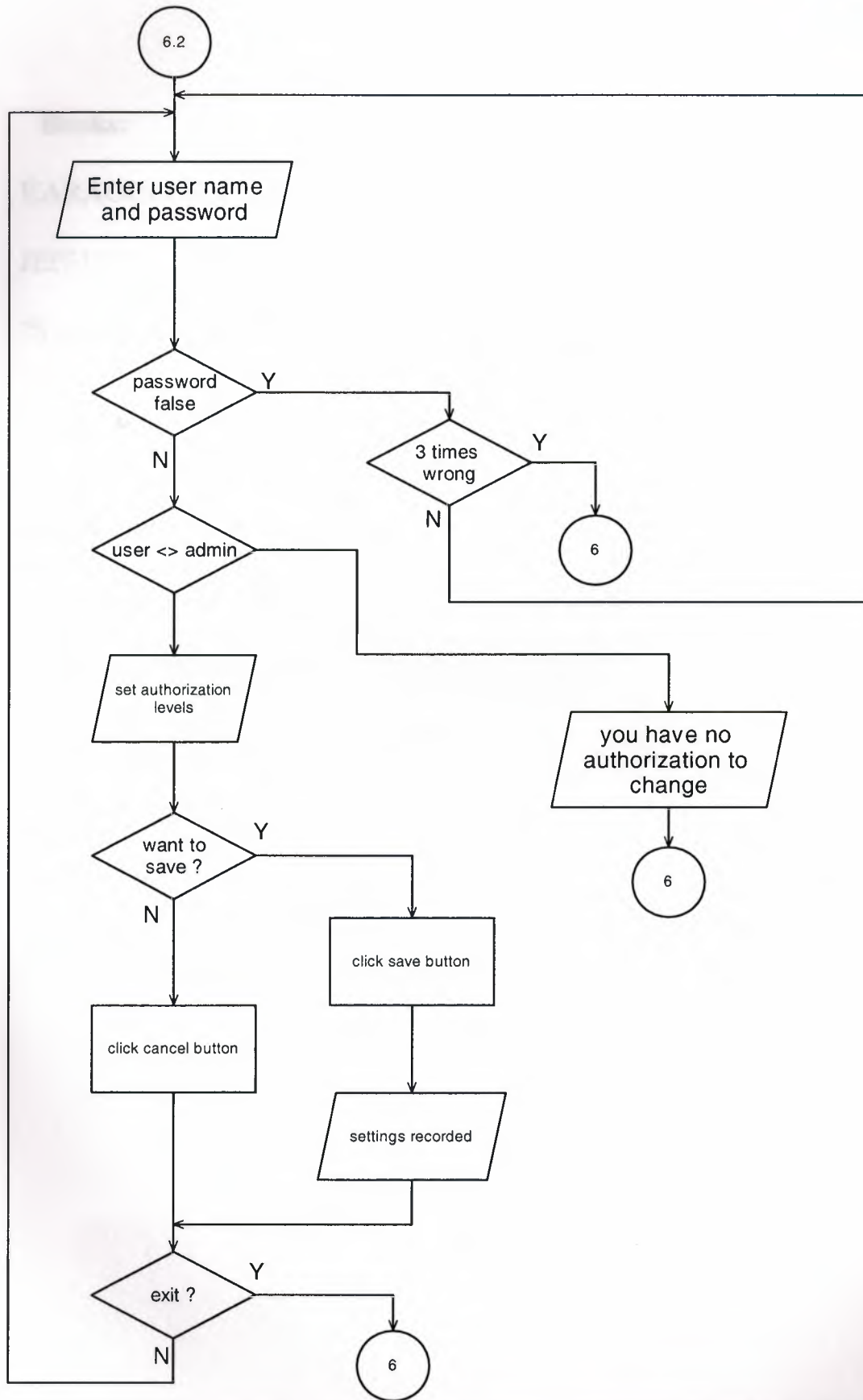
## PRODUCT DESCRIPTIONS



# CHANGE PASSWORD



## USER AUTHORIZATIONS



## REFERENCES

### Books:

KARAGULLE Ihsan, "Borland Delphi 7" Türkmen Kitabevi, 200

JEFFERY L. Whitten, LENNIE D.Bently ,KEVIN C. Dittmen ,

"System Analysis and Design Methods" Mc Grawhill, sixth edition.

### Web Sites:

[www.borland.com](http://www.borland.com)

[www.delphiturk.com](http://www.delphiturk.com)

[www.programmersheaven.com](http://www.programmersheaven.com)

[www.programlama.com](http://www.programlama.com)



## USER GUIDELINES

"Find"

update , add , delete

"okkey" button

button

Hint: "Find", "okkey"

in Figure 2.

## 1- CUSTOMER

At this process customer information such as name, surname, telephone will be taken from the customer and save to Database.

- **Add customer** : in this process you can entry new customers to the database. Firstly you should click “add customer” button as you seen in Figure 1 , then enter the customer information to related fields ,after entering necessary information you should click “okey” button to save this customer , if you don’t want to save please click “cancel” to cancel this operation , if you want to see all customers please click the “show all customers” button

Hint: “add customer”, “okey “, “cancel” and “show all customers” buttons are ,seen in Figure 1

- **Search / update** : in this process you can search an old-recorded customer and if necessary update their information. To find a customer you should write name or surname of the customer in fields then click the “find” button. When you find the customer that you want to show or update please enter the new information to necessary fields then click the “okey” button, if you don’t want to save changes please click the “cancel” button .

Hint: “find”, “okey “, “cancel” and “show all customers” buttons are , seen in Figure 2.

- **Delete customer :** in this process you can delete a recorded customers if there is no need to information about the selected customers, if you want to delete a customer record please select a customer from customer list or find via using “find” button when you select a customer that you want to delete please click the “ delete” button you will see an confirm message to delete, check again the customer if are you sure please okey button on confirm window ,if you click cancel button on confirm window the selected customer wont be deleted.

Hint: “find” , “ delete” and “cancel” buttons are seen in Figure 3.

## 2- STOCK PROCESS

All processes about products and suppliers will be made in this processes

- **Supplier process :** this process provide recording supplier information, update them when necessary, and may delete records if all contacts are give out.

If you want to add a new supplier please use the “add new supplier” page in supplier form, to add new supplier click “add supplier button” then enter the supplier information to related fields, after entering necessary information you should click “okey” button to save this supplier , if you don’t want to save please click “cancel” to cancel this operation , if you want to see all supplier please click the “show current supplier” button

Hint: “add customer”, “okey “, “cancel” and “show current supplier” buttons are ,seen in Figure 4

In update page you can search an old-recorded supplier and if necessary update their information. To find a supplier you should write name or surname of the supplier in fields then click the “find” button. When you find the customer that you want to show or update please enter the new information to necessary fields then click the “okey” button, if you don’t want to save changes please click the “cancel” button .

Hint: “find”, “okey “, “cancel” and “show all customers” buttons are , seen in Figure 5.

In delete page you can delete a recorded suppliers if there is no need to information a bout the selected suppliers, if you want to delete a supplier record please select a supplier from supplier list or find via using “find” button, when you select a supplier that you want to delete please click the “delete” button you will see an confirm message to delete, check again the supplier if are you sure please okey button on confirm window ,if you click cancel button on confirm window the selected supplier wont be deleted.

Hint: “find” , “ delete” and “cancel” buttons are seen in Figure 6.



- **Purchase process** : all products that are purchased, recorded with suppliers information and total amount of invoices computed then saved in this process.

To make a purchasing process firstly you should add new invoice or update a recorded invoice, select your process type (update or add new invoice) then fill related fields in invoice when you finished filling information to invoice please click “okey” button to add or delete a product , after click second part is enabled for your using, you can add product via clicking “add product” button , after click “add product” button enter the product information( don’t forget selling price can not be smaller than buying price if you enter a value smaller than buying price the system give a warning to you if you see this warning please read carefully), after entering information please click the “save” button in second part if you continue add product please repeat above actions, when you finished add product to calculate total cost and saving products please enter the “okey” button.

If you want to delete a product from invoice before saving you should select product that you want to delete and click delete button in appear second part .

Hint : all buttons are seen in Figure 7.

- **Selling process** : when a customer arrive the company and ask some products this process is using to perform it, process shows the available products in stock, give a recommended price for selling that is entering system on purchasing and give a chance to discount on recommended price, when selling to be completed if there is an instalment request compute instalment amounts and determine the payment plan and finally print all invoices and payment plans.

Hint : selling process seen in Figure 8

### 3- ACCOUNT PROCESS

Accounting jobs are done by using this process

**Customer Accounts** : this process provide the information from DB who have debt to company and customer payments recorded at this process.

Firstly select a customer from customer list then you will see the customers debt , if you want you can calculate all debts of this customer using calculator, if customer pay a debt please click “do payment” button.

Hint : calculator, do payment ,find and cancel buttons are seen in Figure 9.



- **Supplier Accounts** : firstly this process using for check supplier statements if there will be an error in statements then recorded the payments that made to suppliers.

Hint: check supplier statements is seen in Figure 12,

Do payment and cancelled is seen in Figure 11.

- **Personnel Accounts** : at this process compute the personnel salary on based of working days, and recorded advance payments & salary payments.

Firstly select a customer and select a months of year then click the calculate button information will be appear as Figure 12, you can do salary payments or advance payments, if you do advance payments the Diverse amount will be appear as Figure 13.

#### 4- PERSONNEL PROCESS

All records about personnel take at this process

This process has sub processes

- **Personnel records** : here identity information about personnel are taken such as Ssn number, name, surname, photo.

This process provide recording personnel information, update them when necessary, and may delete records if all contacts are give out.

If you want to add a new personnel please use the “add new personnel” page in personnel form, to add new personnel click “add personnel

button” then enter the personnel information to related fields, after entering necessary information you should click “okey” button to save this personnel, if you don’t want to save please click “cancel” to cancel this operation , if you want to see all personel please click the “show all personnel” button

At update you can search an old-recorded personnel and if necessary update their information. To find a personnel you should write name or surname of the personnel in fields then click the “find” button. When you find the personnel that you want to show or update please enter the new information to necessary fields then click the “okey” button, if you don’t want to save changes please click the “cancel” button .

At delete you can delete a recorded personnel if there is no need to information a bout the selected personnel, if you want to delete a personnel record please select a personnel from personnel list or find via using “find” button, when you select a personnel that you want to delete please click the “delete” button you will see an confirm message to delete, check again the personnel if are you sure please okey button on confirm window ,if you click cancel button on confirm window the selected personnel wont be deleted.

Hint: “find” , “add new”, “date” ,“ delete” and “cancel” buttons are seen in Figure 15.

- **Personnel Attendance :** At this process daily personnel attends are recorded and it used at salary evaluation, also permits are given from this process to personnel.

Hint :figure16, figure 17, figure 18.

## 5- REPORTS PROCESS

This process provides various reports to managers and other users

If there will a date criteria in report you will see the “first date” and “second date” enter this fields related to your wants , if you want to daily reports two fields should be same date.

If there will appear a “print” button it means that you will print this report if you want, when you click print button a window will ask to you your printer settings, you set your printer and please print button.

- **Customer Reports :** This section involves reports about customers, give satisfy information to managers.

Reports are ; All customers (figure 19), customers who have debt (figure 20), customers who buying lots of products (figure 21), customers who delay debts (figure 22).

- **Supplier Reports :** This section involves reports about suppliers, give necessary information to workers and managers.

Reports are ; supplier list (figure 23), which supplier supply which products (figure 24).

- **Stock Reports** : This section involves stock reports, give detail information about products in stock and general stock situation to managers and workers.

Reports are ; available products (figure 25) , under critical point products (figure 26), near critical point products (figure 26), most wanted products, never sold products.

- **Personnel Reports** : This section involves general reports about personnel.

Reports are ; personnel list (figure 27), attendance of each personnel (figure 28).

- **Account Reports** : This section involves general accounting reports , give critical and desired information to managers who make decision in company.

Reports are ; sales between two date (figure 29), purchases between two date (figure 30), daily and recent time supplier payments (figure 31), general and daily receivables information (figure 10), expense – revenue of months and days (figure 33).



## 6- SETTINGS PROCESS

Products descriptions and other specific settings included in this process

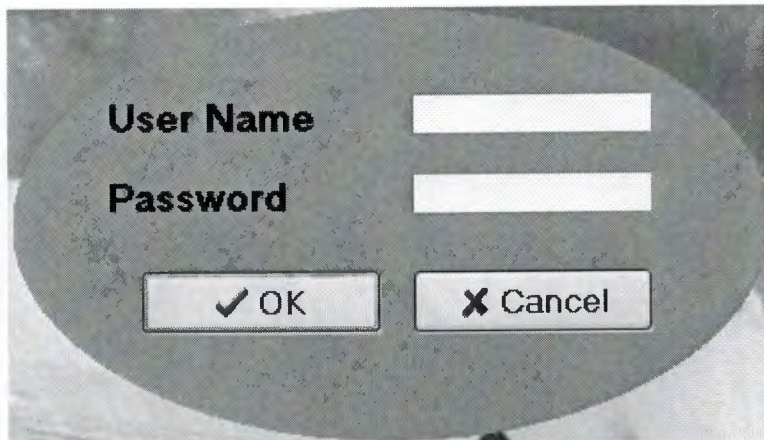
- **Product description** : products name and brand entering by using this process, so its names and brands appear as a normalized form during make stock records also during selling products, when created a report there wont be a data loss because of standard form of names and brands (figure 34).
- **Change Password** : this process give a chance to user to change their user name and password(figure 35).

**User Settings** : this process prepared for Admin s it will provide the change user names, passwords and authorization levels of users ( Figure 36).

## **APPENDIX I**

## SCREEN OUTPUTS

### PASSWORD



**User Name**

**Password**







FIGURE 1

CUSTOMER RECORDS

ADD CUSTOMER

UPDATE RECORDS

DELETE RECORD

ID Number

11

Add Customer

Name

Kemal

✓ OKEY

SurName

Safir

✗ Cancel

Telephone

0 392 223 4562

E - Mail

Kemal@yulhoo.com

Address

Lefkuşa

✕ CLOSE

Show Current Customers

FIGURE 2

CUSTOMER RECORDS

ADD CUSTOMER

UPDATE RECORDS

DELETE RECORD

Name

Surname

ID Number

11

Name

Kemal

Surname

Safir

Telephone

0 392 223 4562

E - Mail

kemal@yahoo.com

Address

Leiköşe

OK

CANCEL

BitBim1

11

Kemal

Safir

12

Seda

Aksel

13

Ahmet

Sevin

14

Teyfik

Poçan

Refresh Table

CLOSE



FIGURE 3

CUSTOMER RECORDS

ADD CUSTOMER

UPDATE RECORDS

DELETE RECORD

Name

Surname

ID Number

11

Name

Kemal

SurName

Safir

Telephone

0 392 223 4562

E - Mail

kemal@yahoo.com

Address

Leftkoşa

DELETE

CANCEL

Find

Refresh Table

Custname	Custsurname
Kemal	Safir
Seda	Aksel
Ahmet	Serin
Teyfik	Poçan

CLOSE



FIGURE 4

7 SUPPLIER

ADD SUPPLIER

UPDATE RECORDS

DELETE RECORD

ID Number

23

ADD Customer

CLOSE

Company Name

Erden Ltd.

OKEY

Telephone

0 332 255 3467

CANCEL

Fax

0 332 255 3467

Mail

erdenltd@hotmail.com

BitBin5

Address

Istanbul/Turkiye

Show Current Supplier



FIGURE 5

ADD SUPPLIER

UPDATE RECORDS

DELETE RECORD

ID Number

Name

23

Erden Ltd.

0 332 255 3467

0 332 255 3467

erdenltd@hotmail.com

Istanbul/Turkiye

OK

CANCEL

Refresh Table

ID Number	Company Name	Telephone
23	Erden Ltd.	0 332 255 3467
24	Gaılp tic.	0 212 364 6757
26	Sahil tic.	0 212 356 8978

CLOSE

110

ADD SUPPLIER
UPDATE RECORDS
DELETE RECORD

ID Number

Name

ID Number

Company Name

Telephone

Fax

Mail

Address

**FIND**

**CLOSE**

Refresh Table
↺

Suppid	Suppname	Supptel
23	Erdem Ltd.	0 332 255 3467
24	Gallp tic.	0 212 364 6757
26	Sahil tic.	0 212 356 8978

**DELETE**

**CANCEL**



FIG 7

Invo	Supplier Name	Buydate	Paydate
13567	Galipl itc.	20.02.2006	28.02.2006
15432	Erdem Ltd.	15.02.2006	25.02.2006

Invoice NO	Supplier Name	Buying Date	Payment Date	Total Amount
13567	Galipl itc.	20.02.2006	28.02.2006	190.00 YTL

**New Invoice**

**Update Invoice**

**OKEY** **X CANCEL**

Serial Number	Name	Brand	Specification	Buying Price	Selling Price	TOTAL COST
1235344	Monitor	Vestel	Flat	180.00 YTL	220.00 YTL	.....

**Add Product** **Save** **Delete**

**OKEY** **X CANCEL**



Customer 14 Teyfik Pogan

Invoice NO

1231223

CLOSE

Add New Customer

Selling Date

24.02.2006

Save

Total Amount 190

Cancel

## SELLING

Make Discount ☐ Apply

&gt;&gt;&gt;Add Product

☐ Installment Payment

Prepare Bill

BitBm13

Selling Price

Name  
Brand

Refresh Table

Serial Number	Product Name	Product Brand	Product Specification

Delete Product From Invoice

Sold Product List In This Invoice

Customer	Serial Number	Selling Date	Invoice Number	Selling Price
14	6783452	24.02.2006	1231223	95,00 YTL



FIGURE 9

Customer Accounts

Payment

Receivables Search

Customer

11 Kemal Safir

▼

Name

FIND

Payment Date

▼ Today

Surname

⊘ CLOSE

Customer ID

11

Name

Kemal

Surname

Safir

Invoices NO

1222333

Selling Date

17.02.2006

Payment Date

23.02.2006

Amount

73.33 YTL

✓ Do Payment

✕ Cancel

Calculator

All Debts Of This Customer

Customer	Invoice NO	Amount	Payment Plan	Selling Date
11	1222333	73.33 YTL	23.02.2006	17.02.2006
11	1222333	73.33 YTL	23.03.2006	17.02.2006
11	1222333	73.33 YTL	23.04.2006	17.02.2006

FIGURE 10

Customer Accounts

Payment

Receivables Search

With Invoice Number

Invoice List

Invoice Number

Search

Daily Or Between Two Dates

First Date

Second Date

Search

Close

All Receivables

Customer ID	Customer Name	Customer Surname	Payment Plan	Amount	Selling Date	Invoices NO
11	Kemal	Safir	23.02.2006	73.33 YTL	17.02.2006	1222333
11	Kemal	Safir	23.03.2006	73.33 YTL	17.02.2006	1222333
11	Kemal	Safir	23.04.2006	73.33 YTL	17.02.2006	1222333

CALCULATOR

TOTAL RECEIVABLE

INSTALLMENT NUMBER

Total Receivable

Retry ( C )

ADD

Subtract

Close

115



FIGURE 11

**T4 Supplier Accounts**

Check Supplier Statements    Payments

Invoice NO

Payment Amount... **540 YTL**   

Invoices NO	Supplier Name	Payment Plan	Amount	Buying Date
13567	Galp tic.	28.02.2006	190,00 YTL	20.02.2006
15432	Eidem Ltd.	25.02.2006	540,00 YTL	15.02.2006

FIGURE 12

Supplier Accounts

Check Supplier Statements

Payments

Invoice NO

Search

ALL INVOICES

Invoice NO	Supplier Name	Amount	Buydate	Paydate
13567	Galip tic.	190,00 YTL	20.02.2006	28.02.2006
15432	Erdem Ltd.	540,00 YTL	15.02.2006	25.02.2006

Products In Invoice

Serial Number	Product Name	Product Brand
6734281	Ram	Kingston
6783452	Ram	Kingston

CLOSE




FIGURE 13


Personal Accounts

Monthly Personnel Payments

NameAhmet

Surname

 FIND

 CLOSE

S.S.N99352726

NameAhmet

Salary1.200,00 YTL

SurnameOzcan

February

2006

Calculate

Gratis Days0 DAY(S)

Earning Amount1200 YTL

Paided Amount0 YTL

Balance1200 YTL

Do Salary Payments

Do Advance Payment

Ssn	Name	Surname	Salary
99352726	Ahmet	Ozcan	1.200,00 YTL



FIGURE 14

Personnel Accounts

Monthly Personnel Payments

Name

Ahmet

Surname

FIND

S.S.N

993352726

Name

Ahmet

Salary

1.200,00 YTL

Surname

Ozcan

February

▼

2006

▼

Calculate

Gratis Days

0 DAYS

Earning Amount

1200 YTL

Paided Amount

0 YTL

Balance

1200 YTL

Do Salary Payments

Do Advance Payment

Partical Payment

Working Days..... 0

Deverse Amount... 0

Balance..... 0

Advance Amount...

✓ Okey

✗ Cancel

✗ CLOSE

✗ CLOSE



FIGURE 15

PERSONNEL RECORDS

Personnel Records

Name

Surname

Social Security NO

99352726

Name

Ahmet

Surname

Ozcan

Age

32

Phone

0 533 561 2354

E - Mail

ahmet@wvalla.com

Address

Leftkoşa

Job

Worker

Hire Date

21.12.2005

Salary


1.200,00 YTL

Permission Extend

10

Find

Load



ALL PERSONNEL

Ssn	Name	Surname
99352726	Ahmet	Ozcan

CLOSE

ADD Personnel

Update

Delete

Okey

Cancel

FIGURE 16

**PERSONNEL ATTENDANCE**

Attendance | Check Records | Permissions

Date: 23.02.2006

Personnel List

<input checked="" type="checkbox"/>	9988776	Ayşe Yildiz
<input type="checkbox"/>	98735343	Onur Sabah
<input type="checkbox"/>	99352726	Ahmet Ozcan

RadioGroup1

☒ Attend Personnels

☐ Absent Personnels

✓ SAVE

<< Remove <<

⊘ CANCEL

Selected Personnels

9988776 Ayşe Yildiz

⊘ CLOSE



FIGURE 17

PERSONNEL ATTENDANCE

AttendanceCheck RecordsPermissions

Date23.02.2006

CLOSE

Attend PersonnelsPermitted PersonnelsAbsent Personnels

name	surname	phone	ssn
Ayşe	Yıldız	0 542 888 7656	9988776
Onur	Sabah	0 533 678 5435	98735343
Ahmet	Ozcan	0 533 861 2354	99352726

Delete a Record To Reform

FIGURE 18

PERSONNEL ATTENDANCE
Attendance
Check Records
Permissions
CLOSE

Ssn	Name	Surname	Perex
9988776	Ayşe	Yıldız	1:
98735343	Onur	Sabah	1:
99352726	Ahmet	Ozcan	11

98735343 Onur Sabah
Permission Extend 15
Used Permissions 0
Permission Type Salaried
Starting Day 15.02.2006
Ending Day 15.02.2006
Permitted Days -
Okey
Cancel



FIGURE 19

7 CUSTOMER REPORTS

All customers

Indebted Customers

Most shopping Customers

Customers who delay debts

Print

Custid	Custname	Custsurname	Custtel	Custmail	Custadd
11	Kemal	Safir	0 392 223 4562	kemal@yahoo.com	Lefkoşa
12	Seda	Aksel	0 392 225 6784	seda@hotmail.com	Girne
13	Ahmet	Sevin	0 392 227 6549	ahmet@gmail.com	Güzelyurt
14	Teyfik	Poçan	0 392 9874	teyfik@yahoo.com	Lefkoşa



FIGURE 20

CUSTOMER REPORTS									
All customers		Indebted Customers		Most shopping Customers		Customers who delay debts			
custid	custname	custsurname	Phone	sinvono	selldate	amount	Payment Plan		
11	Kenal	Safir	0 392 223 4562	1222333	17.02.2006	73.33 YTL	23.02.2006		
11	Kenal	Safir	0 392 223 4562	1222333	17.02.2006	73.33 YTL	23.03.2006		
11	Kenal	Safir	0 392 223 4562	1222333	17.02.2006	73.33 YTL	23.04.2006		
12	Seda	Aksel	0 392 225 6784	2132345	23.02.2006	135.00 YTL	23.02.2006		
12	Seda	Aksel	0 392 225 6784	4325345	16.02.2006	235.00 YTL	16.02.2006		
12	Seda	Aksel	0 392 225 6784	4343243	23.02.2006	200.00 YTL	23.02.2006		

FIGURE 21

CUSTOMER REPORTS					
All customers		Indebted Customers		Most shopping Customers	
				Customers who delay debts	
Customer ID	Name	Surname	Number Of shopping	Total Amount	
11	Kemal	Safir	1	220.00 YTL	
12	Seda	Aksel	3	570.00 YTL	



FIGURE 22

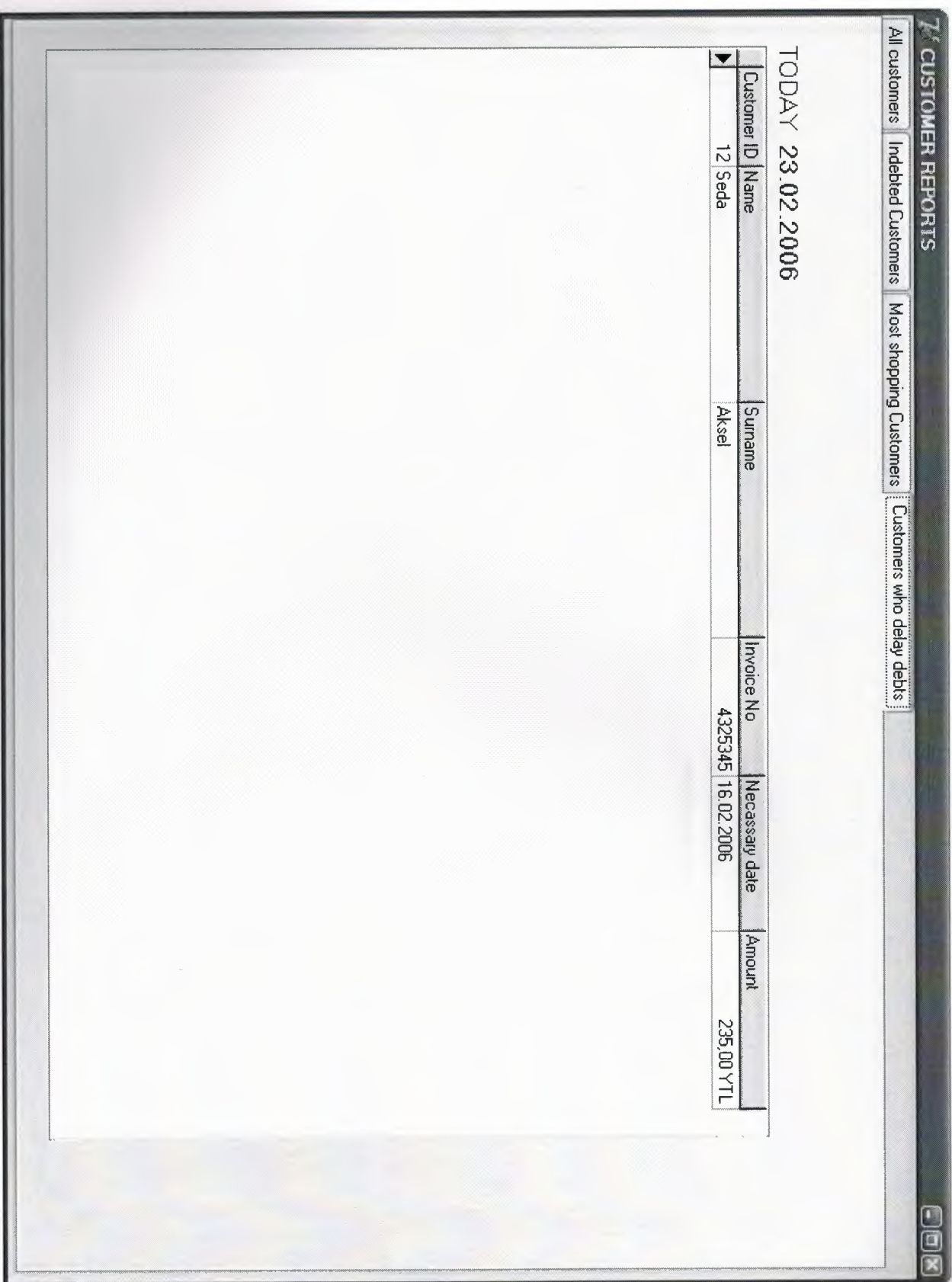


FIGURE 23

7<sup>+</sup> SUPPLIER REPORTS

Supplier List

Which Product Supplied where

Suppid	Suppname	Supptel	Suppfax	Suppmail	Suppadd
23	Erdem Ltd.	0 332 255 3467	0 332 255 3467	erdemltd@hotmail.com	Istanbul/Turkiye
24	Galip tic.	0 212 364 6757	0 212 364 9999	galip@walla.com	Ankara / Turkey
26	Sahil tic.	0 212 356 8978	0 212 3568743	sahil@yahoo.com	Istanbul / Turkey



FIGURE 24

**7. SUPPLIER REPORTS**

Supplier List    Which Product Supplied where

☐ Search With Product Brand    Kingston

☒ Search With Product Name    Hdd

Show

suppname	supptel	suppfax	suppmail	suppadd
Erdem Ltd.	0 332 255 3467	0 332 255 3467	erdemltd@hotmail.com	Istanbul/Turkiye

FIGURE 25

STOCK REPORTS						
Available Products in Stock		Stock Levels	Never Sold Products	Most W/anted Products		
Serial Number	Brand	Name	Specifications	Buying date	Buyprice	Suppname
6783452	Kingston	Ram	256 Mb	20.02.2006	80,00 YTL	Galip itc.



FIGURE 26

STOCK REPORTS

Available Products In Stock

Stock Levels

Never Sold Products

Most Wanted Products

PRODUCTS THAT UNDER CRITICAL POINT

PRODUCT NAME

QUANTITY IN STOCK

NECESSARY QUANTITY

Rain

1

3

PRODUCTS THAT NEAR CRITICAL POINT

PRODUCT NAME

QUANTITY IN STOCK

NECESSARY QUANTITY

Show

FIGURE 27

PERSONNEL REPORTS					
Personnel List		Attendance			
Ssn	Name	Surname	Age	Phone	Mail
9988776	Ayşe	Yıldız	30	0 542 888 7656	ayse@gmail.com
98735343	Onur	Sabah	28	0 533 678 5435	onusabah@hotmail.com
99352726	Ahmet	Ozcan	32	0 533 861 2354	ahmet@walla.com



FIGURE 28

PERSONNEL REPORTS

Personnel List

Attendance

Personnel SSN

9988776

First Date

23.02.2006

Second Date

23.02.2006

Show

ATTEND DAYS

ssn	dat
9988776	23.02.2006

PERMITTED DAYS

ssn	perstat	perend	permis
-----	---------	--------	--------

ABSENT DAYS

ssn	dat
-----	-----

FIGURE 29

ACCOUNT REPORTS

Sales Between To Dates

Purchase Between Two Dates

Supplier Payments

Transactions

First Date

09.02.2006

▼

Second Date

23.02.2006

▼

Total Income 790 YTL

Show

sernum	brand	name	sellprice	selldate	sinvono
1235344	Vestel	Monitor	220.00 YTL	17.02.2006	1222333
6734281	Kingston	Ram	135.00 YTL	23.02.2006	2132345
2382338	Seegate	Hdd	235.00 YTL	16.02.2006	4325345
6753423	Vestel	Monitor	200.00 YTL	23.02.2006	4343243



FIGURE 30

ACCOUNT REPORTS

Sales Between Two Dates

Purchase Between Two Dates

Supplier Payments

Transactions

First Date

17.02.2006

Second Date

23.02.2006

Show

Total Cost

190 YTL

num	brand	name	buypice	buydate	invono
6734281	Kingston	Ram	110,00 YTL	20.02.2006	13567
6783452	Kingston	Ram	80,00 YTL	20.02.2006	13567



FIGURE 31

**T6 ACCOUNT REPORTS**

First Date: 01.02.2006  
 Second Date: 28.02.2006

**Total Debt 730 YTL**

Invono	Buydate	Payment Plan	Amount	Suppname
13567	20.02.2006	28.02.2006	190,00 YTL	Galip tic.
15432	15.02.2006	25.02.2006	540,00 YTL	Erdem Ltd

FIGURE 33

**ACCOUNT REPORTS**

First Date

Second Date

Total Receives **208 YTL**

sinvono	cusid	paydate	Amount
1222333	11	23.02.2006	73.33 YTL
2132345	12	23.02.2006	135.00 YTL

Total Payments **190 YTL**

invono	suppname	paydate	Amount
13567	Galip tic.	20.02.2006	190.00 YTL





FIGURE 34

**SPECIFICATION**

Product Name

Product Brand

Necessary Unit In Stock

 Add Product  Update Product

Name	Brand
▶ Monitor	Philips
Ram	Kingston
Hdd	Seagate
Monitor	Yestel
Ram	Samsung
Hdd	Samsung

FIGURE 35

**PASSWORDS**

Change Password User Accounts

Current User Name

Current Password  **Change**

Label3

Label4

**✓ Okey** **✗ Cancel**

**ⓧ CLOSE**



FIGURE 36

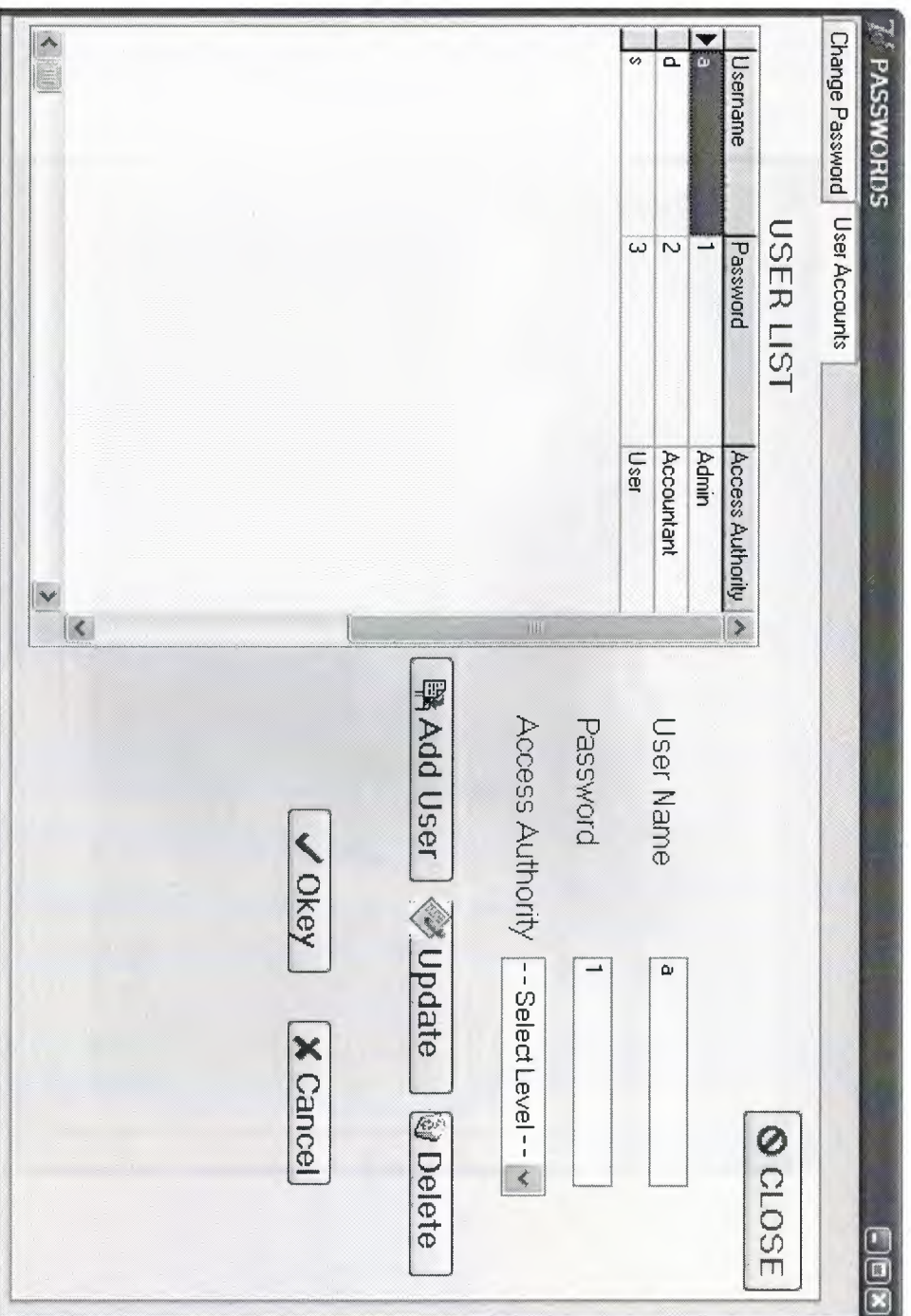
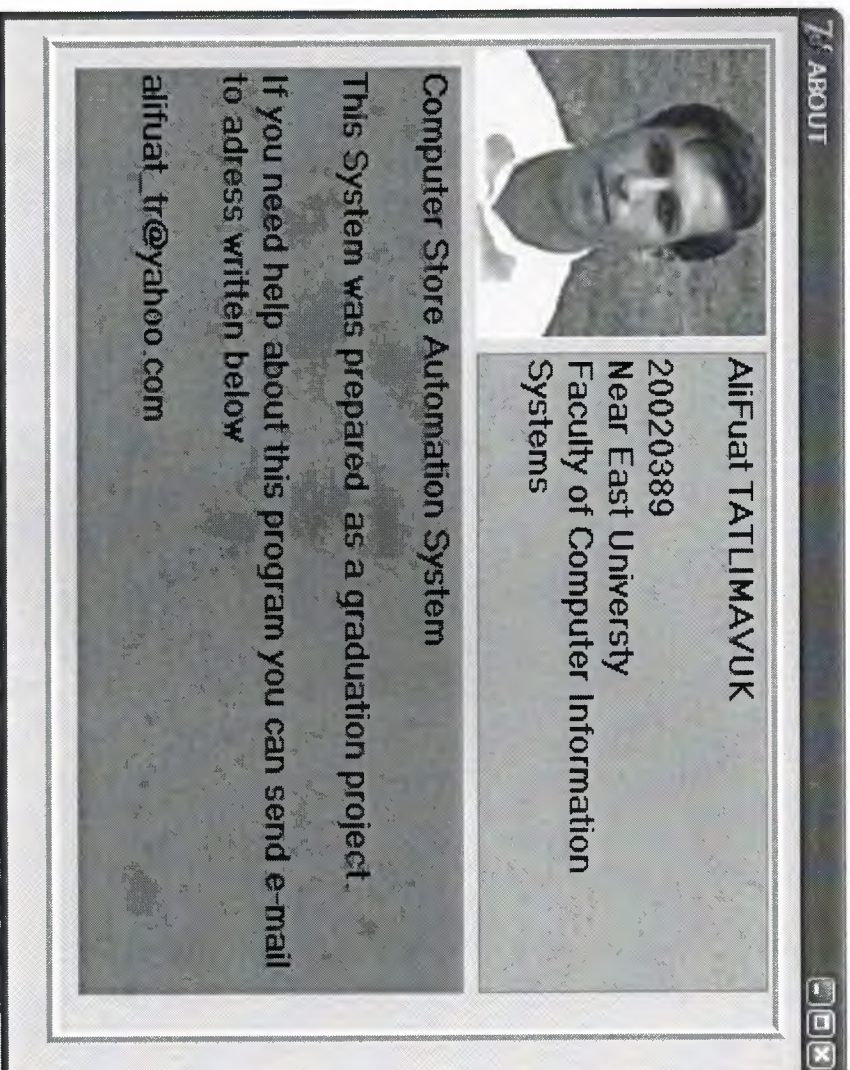




FIGURE 37



## APPENDIX II

## SOURCE CODE

PROJECT :

```
program Project1;
```

```
uses
```

```
  Forms,
```

```
  Unit1 in 'Unit1.pas' {password},
```

```
  Unit2 in 'Unit2.pas' {customer},
```

```
  Unit3 in 'Unit3.pas' {supplier},
```

```
  Unit4 in 'Unit4.pas' {purchase},
```

```
  Unit5 in 'Unit5.pas' {quantity},
```

```
  Unit6 in 'Unit6.pas' {main},
```

```
  Unit7 in 'Unit7.pas' {selling},
```

```
  Unit8 in 'Unit8.pas' {custacc},
```

```
  Unit9 in 'Unit9.pas' {suppacc},
```

```
  Unit10 in 'Unit10.pas' {personnel},
```

```
  Unit11 in 'Unit11.pas' {peracc},
```

```
  Unit12 in 'Unit12.pas' {peratt},
```

```
  Unit13 in 'Unit13.pas' {setting},
```

```
  Unit14 in 'Unit14.pas' {repcust},
```

```
  Unit15 in 'Unit15.pas' {custreport},
```

Unit16 in 'Unit16.pas' {repcustprint},

Unit17 in 'Unit17.pas' {repsupp},

Unit18 in 'Unit18.pas' {repsuppprint},

Unit19 in 'Unit19.pas' {repstock},

Unit20 in 'Unit20.pas' {repacc},

Unit21 in 'Unit21.pas' {repperson},

Unit22 in 'Unit22.pas' {invprt},

Unit23 in 'Unit23.pas' {Form23};

{ \$R \*.res }

begin

Application.Initialize;

Application.CreateForm(Tpassword, password);

Application.CreateForm(Tmain, main);

Application.CreateForm(Trepstock, repstock);

Application.CreateForm(Trepsupp, repsupp);

Application.CreateForm(Trepcust, repcust);

Application.CreateForm(Tcustacc, custacc);

Application.CreateForm(Tselling, selling);

Application.CreateForm(Tsupplier, supplier);

Application.CreateForm(Tperacc, peracc);



```
Application.CreateForm(Tperatt, peratt);  
Application.CreateForm(Tpersonnel, personnel);  
Application.CreateForm(Tsuppacc, suppacc);  
Application.CreateForm(Tpurchase, purchase);  
Application.CreateForm(Tcustomer, customer);  
Application.CreateForm(Tquantity, quantity);  
Application.CreateForm(Tsetting, setting);  
Application.CreateForm(Tcustreport, custreport);  
Application.CreateForm(Trepcustprint, repcustprint);  
Application.CreateForm(Trepsuppprint, repsuppprint);  
Application.CreateForm(Trepacc, repacc);  
Application.CreateForm(Trepperson, repperson);  
Application.CreateForm(Tinvprt, invprt);  
Application.CreateForm(TForm23, Form23);  
Application.Run;  
end.
```

## PASSWORD FORM:

```
unit Unit1;

interface

uses

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, XPMan, StdCtrls, Buttons, DB, DBTables;

type

    Tpassword = class(TForm)

        XPManifest1: TXPManifest;

        Table1: TTable;

        Label1: TLabel;

        Label2: TLabel;

        Edit1: TEdit;

        Edit2: TEdit;

        BitBtn1: TBitBtn;

        BitBtn2: TBitBtn;

        procedure BitBtn1Click(Sender: TObject);

        procedure BitBtn2Click(Sender: TObject);

        procedure FormCreate(Sender: TObject);

    private

        { Private declarations }

    public
```

```

    { Public declarations }

end;

begin

var
    password: Tpassword;
    x:real;

    sekilhandle: HRGN;

implementation

uses Unit2, Unit6;

{$R *.dfm}

procedure Tpassword.BitBtn1Click(Sender: TObject);

begin

    if table1.findkey([edit1.text])then

```

```

begin
  if (table1.Fields.FieldByName('password').Value<>edit2.text) then
    begin
      x:=x+1;
      showmessage('Wrong Password');
      edit2.SetFocus ;
    end
  else
    begin
      main.show;
      main.Edit1.Text :=password.Edit1.Text;
      password.hide;
    end;
  end
else
  if (table1.Fields.FieldByName('username').Value<>edit1.text) then
    begin
      x:=x+1;
      showmessage('Wrong User Name');
      edit1.SetFocus ;
    end;
  end;
end;

```



if x=3 then

begin showmessage('No More Try');

application.Terminate ;

end;

end;

procedure Tpassword.BitBtn2Click(Sender: TObject);

begin

application.Terminate ;

end;

procedure Tpassword.FormCreate(Sender: TObject);

begin

x:=0;

sekilhandle:=createellipticrgn(-1,0,password.Width,password.Height) ;

setwindowrgn(password.Handle,sekilhandle,TRUE);

end;

end.

MAIN MENU:

unit Unit6;

interface UNTI: TMenu

uses pplicrAcc;

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Menus, StdCtrls, DB, DBTables, ExtCtrls;

type

Tmain = class(TForm)

MainMenu1: TMainMenu;

cust: TMenuItem;

add: TMenuItem;

update: TMenuItem;

delete: TMenuItem;

stock: TMenuItem;

supp: TMenuItem;

adds: TMenuItem;

updates: TMenuItem;

deletes: TMenuItem;

N1: TMenuItem;

purc: TMenuItem;

N2: TMenuItem;  
Sell: TMenuItem;  
ACCOUNT1: TMenuItem;  
custa: TMenuItem;  
SupplierAccounts1: TMenuItem;  
suppa: TMenuItem;  
N3: TMenuItem;  
PersonnelAccounts1: TMenuItem;  
custaccpay: TMenuItem;  
custaccrec: TMenuItem;  
suppacsta: TMenuItem;  
suppaccpay: TMenuItem;  
N4: TMenuItem;  
N5: TMenuItem;  
PERSONNEL1: TMenuItem;  
PersonnelRecords1: TMenuItem;  
pera: TMenuItem;  
EXIT1: TMenuItem;  
Edit2: TEdit;  
edit1: TEdit;  
Query1: TQuery;  
Query1Username: TStringField;

Query1Password: TStringField;  
Query1Access: TStringField;  
Button1: TButton;  
REPORTS1: TMenuItem;  
SETTINGS1: TMenuItem;  
ProductNormalized1: TMenuItem;  
N6: TMenuItem;  
ABOUT1: TMenuItem;  
HELP1: TMenuItem;  
ChangeProfile1: TMenuItem;  
N7: TMenuItem;  
UserAccounts1: TMenuItem;  
CustomerReports1: TMenuItem;  
N8: TMenuItem;  
SupplierReports1: TMenuItem;  
N9: TMenuItem;  
StockReports1: TMenuItem;  
N10: TMenuItem;  
AccountReports1: TMenuItem;  
N11: TMenuItem;  
PersonnelReports1: TMenuItem;  
Image1: TImage;



```

procedure addClick(Sender: TObject);

procedure updateClick(Sender: TObject);

procedure deleteClick(Sender: TObject);

procedure addsClick(Sender: TObject);

procedure updatesClick(Sender: TObject);

procedure deletesClick(Sender: TObject);

procedure purcClick(Sender: TObject);

procedure SellClick(Sender: TObject);

procedure custaccpayClick(Sender: TObject);

procedure custaccrecClick(Sender: TObject);

procedure suppaccstaClick(Sender: TObject);

procedure suppaccpayClick(Sender: TObject);

procedure PersonnelAccounts1Click(Sender: TObject);

procedure PersonnelRecords1Click(Sender: TObject);

procedure peraClick(Sender: TObject);

procedure EXIT1Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure FormClick(Sender: TObject);

procedure MainMenu1Change(Sender: TObject; Source: TMenuItem;
Rebuild: Boolean);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

```

```

procedure ProductNormalized1Click(Sender: TObject);

procedure ChangeProfile1Click(Sender: TObject);

procedure UserAccounts1Click(Sender: TObject);

procedure IndebtedCustomers1Click(Sender: TObject);

procedure CustomerReports1Click(Sender: TObject);

procedure SupplierReports1Click(Sender: TObject);

procedure StockReports1Click(Sender: TObject);

procedure AccountReports1Click(Sender: TObject);

procedure PersonnelReports1Click(Sender: TObject);

procedure ABOUT1Click(Sender: TObject);

procedure Image1Click(Sender: TObject);

private

  { Private declarations }

public

  { Public declarations }

end;

var

  main: Tmain;

implementation

```

```
uses Unit2, Unit3, Unit4, Unit7, Unit8, Unit9, Unit11, Unit10, Unit12,  
Unit5, Unit13, Unit14, Unit17, Unit19, Unit20, Unit21, Unit23;
```

```
{ $R *.dfm }
```

```
procedure Tmain.addClick(Sender: TObject);
```

```
begin
```

```
customer.show;
```

```
customer.TabSheet1.Show ;
```

```
end;
```

```
procedure Tmain.updateClick(Sender: TObject);
```

```
begin
```

```
customer.show;
```

```
customer.TabSheet2.Show ;
```

```
end;
```

```
procedure Tmain.deleteClick(Sender: TObject);
```

```
begin
```

```
customer.show;
```

```
customer.TabSheet3.Show ;
```

end;

procedure Tmain.addsClick(Sender: TObject);

begin

supplier.show;

supplier.TabSheet1.Show;

end;

procedure Tmain.updatesClick(Sender: TObject);

begin

supplier.show;

supplier.TabSheet2.Show;

end;

procedure Tmain.deletesClick(Sender: TObject);

begin

supplier.show;

supplier.TabSheet3.Show;

end;

procedure Tmain.purcClick(Sender: TObject);

begin



purchase.show;

end;

procedure Tmain.SellClick(Sender: TObject);

begin

selling.show;

end;

procedure Tmain.custaccpayClick(Sender: TObject);

begin

custacc.show;

custacc.TabSheet1.Show ;

end;

procedure Tmain.custaccrecClick(Sender: TObject);

begin

custacc.show;

custacc.TabSheet2.Show;

end;

procedure Tmain.suppaccstaClick(Sender: TObject);

begin

```
suppacc.Show ;
```

```
suppacc.TabSheet1.Show;
```

```
end;
```

```
procedure Tmain.suppaccpayClick(Sender: TObject);
```

```
begin
```

```
suppacc.Show;
```

```
suppacc.TabSheet2.Show;
```

```
end;
```

```
procedure Tmain.PersonnelAccounts1Click(Sender: TObject);
```

```
begin
```

```
peracc.show;
```

```
end;
```

```
procedure Tmain.PersonnelRecords1Click(Sender: TObject);
```

```
begin
```

```
personnel.show;
```

```
end;
```

```
procedure Tmain.peraClick(Sender: TObject);
```

```
begin  
peratt.show;  
end;
```

```
procedure Tmain.EXIT1Click(Sender: TObject);  
  
begin  
application.Terminate ;  
  
end;
```

```
procedure Tmain.Button1Click(Sender: TObject);  
  
begin  
query1.Close;  
query1.SQL.Clear ;  
query1.SQL.Add('select * from db\user where username="'+edit1.text+'"');  
query1.ExecSQL ;  
query1.Open;
```

```
edit2.Text :=query1 Access.AsString ;
```

```
if edit2.Text='Accountant' then  
  
begin
```

```
//custaccpay.Enabled :=false;
```

```
useraccounts1.Enabled :=false;
```

```
pera.Enabled :=false;
```

```
end;
```

```
if edit2.Text='User' then
```

```
begin
```

```
    custaccpay.Enabled :=false;
```

```
    custaccrec.Enabled :=false;
```

```
    //suppaccsta.Enabled :=false;
```

```
    suppaccpay.Enabled :=false; Change(Sender, TObject, Self)
```

```
    personnelaccounts1.Enabled :=false;
```

```
    useraccounts1.Enabled :=false;
```

```
    pera.Enabled :=false;
```

```
end;
```

```
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
```



```
procedure Tmain.FormShow(Sender: TObject);
```

```
begin
```

```
button1.SetFocus ;
```

```
button1.Click ;
```

```
end;
```

```
procedure Tmain.FormClick(Sender: TObject);
```

```
begin
```

```
button1.Click ;
```

```
end;
```

```
procedure Tmain.MainMenu1Change(Sender: TObject; Source: TMenuItem;
```

```
Rebuild: Boolean);
```

```
begin
```

```
button1.Click ;
```

```
end;
```

```
procedure Tmain.FormClose(Sender: TObject; var Action: TCloseAction);
```

```
begin
```

```
application.Terminate ;
```

```
end;
```

```
procedure Tmain.ProductNormalized1Click(Sender: TObject);
```

```

begin
quantity.show;
end;

procedure Tmain.ChangeProfile1Click(Sender: TObject);
begin
setting.show;
setting.TabSheet1.Show;
end;

procedure Tmain.UserAccounts1Click(Sender: TObject);
begin
setting.Show;
setting.TabSheet2.Show;
end;

procedure Tmain.IndebtedCustomers1Click(Sender: TObject);
begin
repcust.show;
end;

procedure Tmain.CustomerReports1Click(Sender: TObject);
begin
repcust.Show;
end;

```

```
procedure Tmain.SupplierReports1Click(Sender: TObject);
```

```
begin
```

```
  repsupp.show;
```

```
end;
```

```
procedure Tmain.StockReports1Click(Sender: TObject);
```

```
begin
```

```
  repstock.show;
```

```
end;
```

```
procedure Tmain.AccountReports1Click(Sender: TObject);
```

```
begin
```

```
  repacc.show;
```

```
end;
```

```
procedure Tmain.PersonnelReports1Click(Sender: TObject);
```

```
begin
```

```
  repperson.show;
```

```
end;
```

```
procedure Tmain.ABOUT1Click(Sender: TObject);
```

```
begin
```

```
  form23.show;
```

```
end;
```

```
end.
```

CUSTOMER FORM:

unit Unit2;

interface

interface

uses

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Buttons, DBTables, Grids, DBGrids, ExtCtrls, DBCtrls,  
DB, Mask, ComCtrls, bde;

type

Tcustomer = class(TForm)

PageControl1: TPageControl;

TabSheet1: TTabSheet;

TabSheet2: TTabSheet;

TabSheet3: TTabSheet;

DBEdit1: TDBEdit;

DBEdit2: TDBEdit;

DBEdit3: TDBEdit;

DBEdit4: TDBEdit;

DBEdit5: TDBEdit;

DBEdit6: TDBEdit;

Table1: TTable;



DataSource1: TDataSource;  
DBNavigator1: TDBNavigator;  
DBGrid1: TDBGrid;  
Query1: TQuery;  
DataSource2: TDataSource;  
BitBtn1: TBitBtn;  
Label1: TLabel;  
Label2: TLabel;  
Label3: TLabel;  
Label4: TLabel;  
Label5: TLabel;  
Label6: TLabel;  
BitBtn2: TBitBtn;  
BitBtn3: TBitBtn;  
BitBtn4: TBitBtn;  
DBGrid2: TDBGrid;  
BitBtn5: TBitBtn;  
Edit1: TEdit;  
Edit2: TEdit;  
Label7: TLabel;  
Label8: TLabel;  
BitBtn6: TBitBtn;

DBEdit7: TDBEdit;  
DBEdit8: TDBEdit;  
DBEdit9: TDBEdit;  
DBEdit10: TDBEdit;  
DBEdit11: TDBEdit;  
DBEdit12: TDBEdit;  
Label9: TLabel;  
Label10: TLabel;  
Label11: TLabel;  
Label12: TLabel;  
Label13: TLabel;  
Label14: TLabel;  
Query1Custid: TFloatField;  
Query1Custname: TStringField;  
Query1Custsurname: TStringField;  
Query1Custtel: TStringField;  
Query1Custmail: TStringField;  
Query1Custadd: TStringField;  
DBNavigator2: TDBNavigator;  
BitBtn7: TBitBtn;  
BitBtn8: TBitBtn;  
Label15: TLabel;

Label16: TLabel;  
Label17: TLabel;  
Label18: TLabel;  
Label19: TLabel;  
Label20: TLabel;  
Label21: TLabel;  
Label22: TLabel;  
Edit4: TEdit;  
BitBtn9: TBitBtn;  
DBEdit13: TDBEdit;  
DBEdit14: TDBEdit;  
DBEdit15: TDBEdit;  
DBEdit16: TDBEdit;  
DBEdit17: TDBEdit;  
DBEdit18: TDBEdit;  
DBGrid3: TDBGrid;  
BitBtn10: TBitBtn;  
BitBtn11: TBitBtn;  
BitBtn12: TBitBtn;  
BitBtn13: TBitBtn;  
edit3: TEdit;  
BitBtn14: TBitBtn;

```

BitBtn15: TBitBtn;

BitBtn16: TBitBtn;

procedure closem;

procedure open;

procedure BitBtn1Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure PageControl1Change(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

procedure BitBtn3Click(Sender: TObject);

procedure BitBtn4Click(Sender: TObject);

procedure BitBtn5Click(Sender: TObject);

procedure BitBtn6Click(Sender: TObject);

procedure BitBtn7Click(Sender: TObject);

procedure BitBtn8Click(Sender: TObject);

procedure BitBtn12Click(Sender: TObject);

procedure BitBtn13Click(Sender: TObject);

procedure BitBtn9Click(Sender: TObject);

procedure BitBtn10Click(Sender: TObject);

procedure BitBtn11Click(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Edit1KeyPress(Sender: TObject; var Key: Char);

procedure Edit2KeyPress(Sender: TObject; var Key: Char);

```



```

procedure edit3KeyPress(Sender: TObject; var Key: Char);

procedure Edit4KeyPress(Sender: TObject; var Key: Char);

procedure BitBtn14Click(Sender: TObject);

procedure BitBtn15Click(Sender: TObject);

procedure BitBtn16Click(Sender: TObject);

private

    { Private declarations }

public

    { Public declarations }

end;

var

    customer: Tcustomer;

    e:real;

implementation

{$R *.dfm}

procedure Tcustomer.closem ;

begin

dbedit1.Enabled :=false;

dbedit2.Enabled :=false;

```

```
dbedit3.Enabled :=false;  
dbedit4.Enabled :=false;  
dbedit5.Enabled :=false;  
dbedit6.Enabled :=false;  
dbgrid2.Visible :=false;  
end;
```

```
procedure Tcustomer.open ;  
begin  
dbedit1.Enabled :=true;  
dbedit2.Enabled :=true;  
dbedit3.Enabled :=true;  
dbedit4.Enabled :=true;  
dbedit5.Enabled :=true;  
dbedit6.Enabled :=true;  
end;
```

```
procedure Tcustomer.BitBtn1Click(Sender: TObject);  
begin
```

application.Terminate ;

end;

procedure Tcustomer.FormCreate(Sender: TObject);

begin

closem;

end;

procedure Tcustomer.PageControl1Change(Sender: TObject);

begin

closem;

end;

procedure Tcustomer.BitBtn2Click(Sender: TObject);

begin

open;

dbnavigator1.BtnClick(nbinsert);

e:=1;

dbedit1.SetFocus ;

end;

procedure Tcustomer.BitBtn3Click(Sender: TObject);

```
begin
if e=1 then
begin
if dbedit1.Text = " then
begin
showmessage('Please Enter The Customer ID number');
dbedit1.SetFocus ;
end
else if dbedit2.Text = " then
begin
showmessage('Please Enter The Customer Name');
dbedit2.SetFocus ;
end
else if dbedit3.Text = " then
begin
showmessage('Please Enter The Customer Surname');
dbedit3.SetFocus ;
end else
begin
try
dbnavigator1.BtnClick(nbpost);
showmessage('Customer have been Recorded succesfully');
```



```

closem;

e:=0;

except

on EDBEngineError do

begin

application.MessageBox(' This Customer ID is Used Before'+#13+

        ' Please Enter Another ID... ', 'Error', mb_OK+mb_IconStop);


dbedit1.SetFocus ;

dbedit1.Clear ;


end;

end ;

end;

end

else

showmessage('Please Firstly Add a Customer')

end;


procedure Tcustomer.BitBtn4Click(Sender: TObject);

begin

```

```
dbnavigator1.BtnClick(nbcancel);
```

```
closem;
```

```
e:=0;
```

```
end;
```

```
procedure Tcustomer.BitBtn5Click(Sender: TObject);
```

```
begin
```

```
dbgrid2.Visible :=true;
```

```
end;
```

```
procedure Tcustomer.BitBtn6Click(Sender: TObject);
```

```
begin
```

```
if (edit1.text<>") or (edit2.text<>") then
```

```
begin
```

```
    query1.Close;
```

```
    query1.SQL.Clear;
```

```
    query1.SQL.Add('select * from db\cust where');
```

```
if edit1.text<>" then
```

```
begin
```

```

    query1.SQL.Add(' custname = '"+edit1.text+"'");

end;

if edit2.text<>" then

begin

query1.SQL.Add(' custsurname = '"+edit2.Text+"' ')

end;


query1.Open;

if query1.RecordCount <> 0 then

begin

dbnavigator2.BtnClick(nbedit);

dbedit8.SetFocus;

end

else

begin

showmessage('There Is NO Record Like This');

edit1.Clear ;

edit1.SetFocus ;

end;

end

else

begin

```

```
showmessage('Please Enter A Criteria For Search');
```

```
edit1.SetFocus ;
```

```
end;
```

```
end;
```

```
procedure Tcustomer.BitBtn7Click(Sender: TObject);
```

```
begin
```

```
dbnavigator2.BtnClick(nbpost);
```

```
end;
```

```
procedure Tcustomer.BitBtn8Click(Sender: TObject);
```

```
begin
```

```
showmessage('Update operation Cancelled');
```

```
dbnavigator1.BtnClick(nbcancel);
```

```
end;
```

```
procedure Tcustomer.BitBtn12Click(Sender: TObject);
```

```
begin
```

```
query1.Close;
```

```
    query1.SQL.Clear;
```

```
    query1.SQL.Add('select * from db\cust ');
```

```
    query1.Open;
```



end;

procedure Tcustomer.BitBtn13Click(Sender: TObject);

begin

query1.Close;

query1.SQL.Clear;

query1.SQL.Add('select \* from db\cust ');

query1.Open;

end;

procedure Tcustomer.BitBtn9Click(Sender: TObject);

begin

if (edit3.text<>") or (edit4.text<>") then

begin

query1.Close;

query1.SQL.Clear;

query1.SQL.Add('select \* from db\cust where');

if edit3.text<>" then

begin

query1.SQL.Add('where custname = "'+edit3.text+'");

end;

```

if edit4.text<>" then
    begin
        query1.SQL.Add(' custsurname = '"+edit4.Text+"' ')
        end;

        query1.Open;
        if query1.RecordCount <> 0 then
            begin

            end
        else
            begin
                showmessage('There Is NO Record Like This');
                edit3.Clear ;
                edit3.SetFocus ;
                end;

            end
        else
            begin
                showmessage('Please Enter A Criteria For Search');
                edit3.SetFocus ;

```

end;

end;

procedure Tcustomer.BitBtn10Click(Sender: TObject);

begin

dbnavigator2.BtnClick( nbdelete);

end;

procedure Tcustomer.BitBtn11Click(Sender: TObject);

begin

dbnavigator2.BtnClick(nbcancel);

edit3.Clear ;

edit4.Clear ;

end;

procedure Tcustomer.Button1Click(Sender: TObject);

begin

tabsheet1.Show;

end;

```

procedure Tcustomer.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
if (key in ['0'..'9']) then
begin
key:=#0;

messagedlg('Please Enter a alphabetical Character ',mtwarning,[mbok],0);

end;

if key=#13 then
begin

Bitbtn6.Click ;

end;

end;

```

```

procedure Tcustomer.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
if (key in ['0'..'9']) then
begin
key:=#0;

messagedlg('Please Enter a alphabetical Character ',mtwarning,[mbok],0);

end;

if key=#13 then

```



```

begin
    Bitbtn6.Click ;
end;

end;

procedure Tcustomer.edit3KeyPress(Sender: TObject; var Key: Char);
begin
    if (key in ['0'..'9']) then
        begin
            key:=#0;
            messagedlg('Please Enter a alphabetical Character ',mtwarning,[mbok],0);
        end;
        if key=#13 then
            begin
                Bitbtn9.Click ;
            end;
        end;
end;

procedure Tcustomer.Edit4KeyPress(Sender: TObject; var Key: Char);
begin
    if (key in ['0'..'9']) then
        begin
            key:=#0;

```

```

    messagedlg('Please Enter a alphabetical Character ',mtwarning,[mbok],0);

end;

if key=#13 then

begin

    Bitbtn9.Click ;

    end;

end;


procedure Tcustomer.BitBtn14Click(Sender: TObject);

var y:word;

begin

if e=1 then

begin

y:=application.MessageBox('Last customer information was not saved'+#13+

        'Do you Want to save this information ?'+#13+

        ', 'Warning',MB_YESNO + MB_IconExclamation);

    if (y=IDYES) then

        begin

            bitbtn3.Click ;

            end;

        if (y=IDNO) then

```

```

begin
    dbnavigator1.BtnClick(nbcancel)

    e:=0;

    close ;

    end;

end

else

close;

end;

procedure Tcustomer.BitBtn15Click(Sender: TObject);

begin

close;

end;

procedure Tcustomer.BitBtn16Click(Sender: TObject);

begin

close;

end;

end.

```

SUPPLIER FORM:

unit Unit3; *Label*

*Label3: TLabel;*

interface *TForm*

*Label3: TLabel;*

uses *Label3*

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, ExtCtrls, DBCtrls, StdCtrls, Mask, Buttons, Grids, DBGrids,  
DBTables, ComCtrls;

type

Tsupplier = class(TForm)

PageControl1: TPageControl;

TabSheet1: TTabSheet;

TabSheet2: TTabSheet;

TabSheet3: TTabSheet;

Table1: TTable;

Query1: TQuery;

DataSource1: TDataSource;

DataSource2: TDataSource;

DBGrid1: TDBGrid;

BitBtn1: TBitBtn;

Label1: TLabel;



Label2: TLabel;  
Label3: TLabel;  
Label4: TLabel;  
Label5: TLabel;  
Label6: TLabel;  
DBEdit1: TDBEdit;  
DBEdit2: TDBEdit;  
DBEdit3: TDBEdit;  
DBEdit4: TDBEdit;  
DBEdit5: TDBEdit;  
DBEdit6: TDBEdit;  
DBNavigator1: TDBNavigator;  
Query1Suppid: TFloatField;  
Query1Suppname: TStringField;  
Query1Supptel: TStringField;  
Query1Suppfax: TStringField;  
Query1Suppmail: TStringField;  
Query1Supadd: TStringField;  
BitBtn2: TBitBtn;  
BitBtn3: TBitBtn;  
BitBtn4: TBitBtn;  
Label7: TLabel;

Label8: TLabel;  
Label9: TLabel;  
Label10: TLabel;  
Label11: TLabel;  
Label12: TLabel;  
Label13: TLabel;  
Label14: TLabel;  
Edit2: TEdit;  
BitBtn6: TBitBtn;  
DBEdit7: TDBEdit;  
DBEdit8: TDBEdit;  
DBEdit9: TDBEdit;  
DBEdit10: TDBEdit;  
DBEdit11: TDBEdit;  
DBEdit12: TDBEdit;  
DBGrid2: TDBGrid;  
BitBtn7: TBitBtn;  
DBNavigator2: TDBNavigator;  
BitBtn8: TBitBtn;  
BitBtn9: TBitBtn;  
Label15: TLabel;  
Label16: TLabel;

Label17: TLabel;

Label18: TLabel;

Label19: TLabel;

Label20: TLabel;

Label21: TLabel;

Label22: TLabel;

Edit3: TEdit;

Edit4: TEdit;

DBGrid3: TDBGrid;

DBEdit13: TDBEdit;

DBEdit14: TDBEdit;

DBEdit15: TDBEdit;

DBEdit16: TDBEdit;

DBEdit17: TDBEdit;

DBEdit18: TDBEdit;

BitBtn10: TBitBtn;

BitBtn11: TBitBtn;

BitBtn12: TBitBtn;

Edit1: TEdit;

BitBtn13: TBitBtn;

BitBtn14: TBitBtn;

BitBtn15: TBitBtn;

```

BitBtn16: TBitBtn;

procedure BitBtn2Click(Sender: TObject);

procedure open;

procedure closem;

procedure BitBtn3Click(Sender: TObject);

procedure BitBtn5Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure PageControl1Change(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

procedure BitBtn6Click(Sender: TObject);

procedure BitBtn8Click(Sender: TObject);

procedure BitBtn9Click(Sender: TObject);

procedure BitBtn7Click(Sender: TObject);

procedure BitBtn10Click(Sender: TObject);

procedure BitBtn11Click(Sender: TObject);

procedure BitBtn12Click(Sender: TObject);


procedure Edit3KeyPress(Sender: TObject; var Key: Char);

procedure Edit1KeyPress(Sender: TObject; var Key: Char);

// procedure DataSource2DataChange(Sender: TObject; Field: TField);

procedure BitBtn14Click(Sender: TObject);

procedure BitBtn4Click(Sender: TObject);

```



```

procedure BitBtn15Click(Sender: TObject);

procedure BitBtn16Click(Sender: TObject);

private
    { Private declarations }

public
    { Public declarations }

end;

var
    supplier: Tsupplier;
    e: real;

implementation

{$R *.dfm}

procedure Tsupplier.closem ;
begin
    dbedit1.Enabled :=false;
    dbedit2.Enabled :=false;
    dbedit3.Enabled :=false;
    dbedit4.Enabled :=false;

```

```
dbedit5.Enabled :=false;
dbedit6.Enabled :=false;
dbgrid1.Visible :=false;
end;
```

```
procedure Tsupplier.open ;
begin
dbedit1.Enabled :=true;
dbedit2.Enabled :=true;
dbedit3.Enabled :=true;
dbedit4.Enabled :=true;
dbedit5.Enabled :=true;
dbedit6.Enabled :=true;
end;
```

```
procedure Tsupplier.BitBtn2Click(Sender: TObject);
begin
open;
dbnavigator1.BtnClick(nbinsert);
e:=1;
dbedit1.SetFocus ;
```

end;

procedure Tsupplier.BitBtn3Click(Sender: TObject);

begin

if e=1 then

begin

try

dbnavigator1.BtnClick(nbpost);

showmessage('Supplier have been Recorded succesfully');

closem;

e:=0;

except

on EDBEngineError do

begin

application.MessageBox(' This Supplier ID is Used Before'+#13+

    ' Please Enter Another ID... ', 'Error', mb\_OK+mb\_IconStop);

dbedit1.SetFocus ;

dbedit1.Clear ;

end;

end ;

end

else

showmessage('Please Firstly Add a Supplier')

end;

procedure Tsupplier.BitBtn5Click(Sender: TObject);

begin

application.Terminate ;

end;

procedure Tsupplier.FormCreate(Sender: TObject);

begin

closem;

end;

procedure Tsupplier.PageControl1Change(Sender: TObject);

begin

table1.Refresh ;

closem;



end;

```
procedure Tsupplier.BitBtn1Click(Sender: TObject);
```

```
begin
```

```
dbgrid1.Visible :=true;
```

```
end;
```

```
procedure Tsupplier.BitBtn6Click(Sender: TObject);
```

```
begin
```

```
if (edit1.text<>") or (edit2.text<>") then
```

```
begin
```

```
query1.Close;
```

```
    query1.SQL.Clear;
```

```
    query1.SQL.Add('select * from db\supplier where ');
```

```
    if edit1.text<>" then
```

```
        begin
```

```
            query1.SQL.Add(' suppid ='+(edit1.text));
```

```
        end;
```

```
    if edit2.text<>" then
```

```
        begin
```

```

query1.SQL.Add(' suppname = '"+edit2.Text+"' ');

end;

query1.Open;

if query1.RecordCount <> 0 then

begin

dbnavigator2.BtnClick(nbedit);

dbedit8.SetFocus;

end

else

begin

showmessage('There Is NO Record Like This');

edit1.Clear ;

edit1.SetFocus ;

end;

end

else

begin

showmessage('Please Enter A Criteria For Search');

end;

```

end;

procedure Tsupplier.BitBtn8Click(Sender: TObject);

begin

dbnavigator2.BtnClick(nbpost);

end;

procedure Tsupplier.BitBtn9Click(Sender: TObject);

begin

dbnavigator1.BtnClick(nbcancel);

end;

procedure Tsupplier.BitBtn7Click(Sender: TObject);

begin

query1.Close;

query1.SQL.Clear;

query1.SQL.Add('select \* from db\supplier');

query1.Open;

end;

procedure Tsupplier.BitBtn10Click(Sender: TObject);

begin

```
dbnavigator2.BtnClick(nbdelete);
```

```
end;
```

```
procedure Tsupplier.BitBtn11Click(Sender: TObject);
```

```
begin
```

```
dbnavigator2.BtnClick(nbcancel);
```

```
edit3.Clear ;
```

```
edit4.Clear ;
```

```
end;
```

```
procedure Tsupplier.BitBtn12Click(Sender: TObject);
```

```
begin
```

```
if (edit3.text<>") or (edit4.text<>") then
```

```
begin
```

```
    query1.Close;
```

```
    query1.SQL.Clear;
```

```
    query1.SQL.Add('select * from db\supplier where');
```

```
    if edit3.text<>" then
```

```
        begin
```

```
            query1.SQL.Add('where suppid= '+(edit3.text));
```

```
        end;
```

```
    if edit4.text<>" then
```

```
        begin
```



```

query1.SQL.Add(' suppname = '"+edit4.Text+"' ')
end;

query1.Open;

if query1.RecordCount <> 0 then
begin
end
else
begin
showmessage('There Is NO Record Like This');
edit3.Clear ;
edit3.SetFocus ;
end;
end

else
showmessage('Please Enter A Criteria For Search');
end;

procedure Tsupplier.Edit3KeyPress(Sender: TObject; var Key: Char);
begin
if not((key in ['0'..'9'])or(key=#8) or (key=#13)) then
begin
key:=#0;

messagedlg('Please Enter a Numeric Character ',mtwarning,[mbok],0);

```

```

end;

if key=#13 then

begin

    bitbtn12.Click ;

    end;

end;

application Name:

procedure Tsupplier.Edit1KeyPress(Sender: TObject; var Key: Char);

begin

if not((key in ['0'..'9'])or(key=#8) or (key=#13)) then

begin

    key:=#0;

    messagedlg('Please Enter a Numeric Character ',mtwarning,[mbok],0);

end;

if key=#13 then

begin

    Bitbtn6.Click ;

    end;

end;

end;

```

```
procedure Tsupplier.BitBtn14Click(Sender: TObject);
```

```
begin
```

```
var y:word;
```

```
begin
```

```
if e=1 then
```

```
begin
```

```
y:=application.MessageBox('Last customer information was not saved'+#13+
```

```
    'Do you Want to save this information ?'+#13+
```

```
    ','Warning',MB_YESNO + MB_IconExclamation);
```

```
if (y=IDYES) then
```

```
begin
```

```
    bitbtn3.Click ;
```

```
end;
```

```
if (y=IDNO) then
```

```
begin
```

```
    dbnavigator1.BtnClick(nbcancel);
```

```
end;
```

```
e:=0;
```

```
close ;
```

```
end;
```

end

else

close;

end;

procedure Tsupplier.BitBtn4Click(Sender: TObject);

begin

dbnavigator1.BtnClick(nbcancel);

closem;

e:=0;

end;

procedure Tsupplier.BitBtn15Click(Sender: TObject);

begin

close;

end;

procedure Tsupplier.BitBtn16Click(Sender: TObject);

begin

close;

end;

end.

PURCHASE FORM:



unit Unit4;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, ExtCtrls, Grids, DBGrids, StdCtrls, Buttons, DBCtrls, Mask, DB,  
DBTables, ComCtrls, strutils;

type

Tpurchase = class(TForm)

Panel1: TPanel;

Table1: TTable;

Table2: TTable;

DataSource1: TDataSource;

DataSource2: TDataSource;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

DBEdit1: TDBEdit;

DBNavigator1: TDBNavigator;

BitBtn1: TBitBtn;  
BitBtn2: TBitBtn;  
BitBtn3: TBitBtn;  
BitBtn4: TBitBtn;  
DBGrid1: TDBGrid;  
DBLookupComboBox1: TDBLookupComboBox;  
Table3: TTable;  
DataSource3: TDataSource;  
DateTimePicker1: TDateTimePicker;  
DateTimePicker2: TDateTimePicker;  
DBEdit2: TDBEdit;  
DBEdit3: TDBEdit;  
DBEdit4: TDBEdit;  
BitBtn5: TBitBtn;  
BitBtn6: TBitBtn;  
DBNavigator2: TDBNavigator;  
Label5: TLabel;  
Label6: TLabel;  
Label7: TLabel;  
Label8: TLabel;  
Label9: TLabel;  
DBEdit5: TDBEdit;

DBEdit6: TDBEdit;  
DBEdit7: TDBEdit;  
DBEdit10: TDBEdit;  
DBEdit11: TDBEdit;  
BitBtn7: TBitBtn;  
DBLookupComboBox2: TDBLookupComboBox;  
DBLookupComboBox3: TDBLookupComboBox;  
Query1: TQuery;  
Query2: TQuery;  
DataSource4: TDataSource;  
DataSource5: TDataSource;  
BitBtn8: TBitBtn;  
BitBtn9: TBitBtn;  
BitBtn10: TBitBtn;  
BitBtn11: TBitBtn;  
Label10: TLabel;  
Label11: TLabel;  
DBGrid2: TDBGrid;  
Query3: TQuery;  
DataSource6: TDataSource;  
Query3Buyprice: TCurrencyField;  
Query3Sernum: TFloatField;

```
Query3Name: TStringField;

DBEdit8: TDBEdit;

Label12: TLabel;

panel2: TPanel;

BitBtn12: TBitBtn;

DBEdit9: TDBEdit;

Label13: TLabel;

Query3Sellprice: TCurrencyField;

Panel3: TPanel;

BitBtn13: TBitBtn;

procedure BitBtn1Click(Sender: TObject);

procedure closem;

procedure open;

procedure FormCreate(Sender: TObject);

procedure BitBtn3Click(Sender: TObject);

procedure BitBtn4Click(Sender: TObject);

procedure DateTimePicker1Change(Sender: TObject);

procedure DateTimePicker2Change(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

procedure BitBtn5Click(Sender: TObject);

procedure BitBtn6Click(Sender: TObject);

procedure BitBtn7Click(Sender: TObject);
```



```

procedure BitBtn8Click(Sender: TObject);

procedure DBLookupComboBox2Click(Sender: TObject);

procedure DBLookupComboBox3Click(Sender: TObject);

procedure DBLookupComboBox2CloseUp(Sender: TObject);

procedure BitBtn10Click(Sender: TObject);

procedure BitBtn9Click(Sender: TObject);

procedure BitBtn11Click(Sender: TObject);

procedure cost;

procedure open2;

procedure close2;

procedure BitBtn12Click(Sender: TObject);

procedure DBEdit11KeyPress(Sender: TObject; var Key: Char);

procedure DBEdit9KeyPress(Sender: TObject; var Key: Char);

procedure BitBtn13Click(Sender: TObject);

private

    { Private declarations }

public

    { Public declarations }

end;

var

    purchase: Tpurchase;

```

```

f:real;
i:real;
bp,sp:real;
implementation
uses Unit3, Unit5;

{$R *.dfm}

procedure Tpurchase.cost ;
var a: integer;
begin
    i:=0;
    query3.First ;
    for a := 1 to query3.RecordCount do
    begin
        i:=i+ query3.buyprice.AsFloat;
        query3.Next ;
    end;
    label11.caption:=floattostr(i);
end;

procedure Tpurchase.closem;
begin

```

```

dbedit1.Enabled := False;

dbedit2.Enabled := False;

dbedit3.Enabled := False;

dbedit8.Enabled := False;

i:=0;

dblookupcombobox1.Enabled := False;

datetimepicker1.Enabled := False;

datetimepicker2.Enabled := False;

end;

procedure Tpurchase.close2;

begin

dbedit5.Enabled := False;

dbedit10.Enabled := False;

dbedit11.Enabled := False;

dbedit9.Enabled :=false;


dblookupcombobox2.Enabled := False;

dblookupcombobox3.Enabled := False;

end;

procedure Tpurchase.open2;

```

begin

dbedit5.Enabled := True;

dbedit10.Enabled := True;

dbedit11.Enabled := True;

dbedit9.Enabled := true;

dblookupcombobox2.Enabled := True;

dblookupcombobox3.Enabled := True;

end;

procedure Tpurchase.open;

begin

dbedit1.Enabled := True;

dbedit2.Enabled := True;

dbedit3.Enabled := True;

//dbedit8.Enabled := True;

dblookupcombobox1.Enabled := True;

datetimepicker1.Enabled := True;

datetimepicker2.Enabled := True;

end;



```
procedure Tpurchase.BitBtn1Click(Sender: TObject);
```

```
begin
```

```
f:=1;
```

```
open;
```

```
dbnavigator1.BtnClick(nbinsert);
```

```
dbedit1.SetFocus ;
```

```
end;
```

```
procedure Tpurchase.FormCreate(Sender: TObject);
```

```
begin
```

```
close2;
```

```
panel2.Enabled := false;
```

```
f:=0;
```

```
bp:=0;
```

```
sp:=0;
```

```
closem;
```

```
end;
```

```
procedure Tpurchase.BitBtn3Click(Sender: TObject);
```

```
begin
```

```
if dbedit1.Text= " then
```

```

begin
showmessage('Please Enter Invoice No');
dbedit1.SetFocus ;
end

    else if dbedit2.Text= " then
        begin
            showmessage('Please Enter Buying Date');
        end

        else if dbedit3.Text= " then
            begin
                showmessage('Please Enter Payment Date');
            end

else if f=1 then
begin

try
dbnavigator1.BtnClick(nbpost);
panel2.Enabled :=true;
bitbtn7.Click ;
dbedit4.Text:=dbedit1.Text ;

```

```

dbedit6.Text:=dbedit2.Text ;

dbedit7.Text:=dblookupcombobox1.Text ;

closem;

f:=0;

//bitbtn7.Click ;

except

on EDBEngineError do

begin

application.MessageBox(' This Invoices NO is Used Before'+#13+

        ' Please Enter Another NO... ', 'Error', mb_OK+mb_IconStop);


dbedit1.SetFocus ;

dbedit1.Clear ;


end;

end ;

Begin

dateofyear;

end

else

showmessage('For new invoice please click new invoice button'+#13+

        'For update invoice please click update invoice button');

```

end;

procedure Tpurchase.BitBtn4Click(Sender: TObject);

begin

f:=0;

dbnavigator1.BtnClick(nbcancel);

closem;

end;

procedure Tpurchase.DateTimePicker1Change(Sender: TObject);

begin

datetimepicker1.Checked:=true;

dbedit2.Text:=datetostr(Datetimepicker1.Date);

end;

procedure Tpurchase.DateTimePicker2Change(Sender: TObject);

begin

datetimepicker2.Checked:=true;

dbedit3.Text:=datetostr(Datetimepicker2.Date);

end;

procedure Tpurchase.BitBtn2Click(Sender: TObject);

```
begin  
f:=1;  
open;  
dbnavigator1.BtnClick(nbedit);  
end;
```

```
procedure Tpurchase.BitBtn5Click(Sender: TObject);  
begin  
application.Terminate ;  
end;
```

```
procedure Tpurchase.BitBtn6Click(Sender: TObject);  
begin  
supplier.Show;  
supplier.TabSheet1.Show;  
end;
```

```
procedure Tpurchase.BitBtn7Click(Sender: TObject);  
begin  
open2;  
dbnavigator2.BtnClick(nbinsert);  
dbedit4.Text:=dbedit1.Text ;
```



dbedit6.Text:=dbedit2.Text ;

dbedit7.Text:=dblookupcombobox1.Text ;

bitbtn9.Enabled :=true;

dbedit5.SetFocus ;

end;

procedure Tpurchase.BitBtn8Click(Sender: TObject);

begin

quantity.show;

end;

procedure Tpurchase.DBLookupComboBox2Click(Sender: TObject);

begin

query1.Close;

query1.Open;

query2.Close;

query2.Open;

end;

procedure Tpurchase.DBLookupComboBox3Click(Sender: TObject);

begin

```
query1.Refresh;
```

```
query2.Refresh ;
```

```
end;
```

```
procedure Tpurchase.DBLookupComboBox2CloseUp(Sender: TObject);
```

```
begin
```

```
query2.Close;
```

```
    query2.SQL.Clear;
```

```
    query2.SQL.Add('select distinct(brand) from db\quantity where name =
```

```
''+dblookupcombobox2.text+'');
```

```
    query2.Open;
```

```
end;
```

```
procedure Tpurchase.BitBtn10Click(Sender: TObject);
```

```
begin
```

```
bitbtn9.Enabled :=false;
```

```
dbnavigator2.BtnClick(nbcancel);
```

```
close2;
```

```
end;
```

```
procedure Tpurchase.BitBtn9Click(Sender: TObject);
```

```
var p:integer;
```

```

b,s:string;

begin

if dbedit5.Text =" then

begin

showmessage('Please Enter The Serial Number');

dbedit5.SetFocus;

end

else  if (dbedit11.Text <>" ) and (dbedit9.Text <>" ) then

begin

b:=ansireplacetext(dbedit11.text,',','');

s:=ansireplacetext(dbedit9.text,',','');

b:=ansireplacetext(b,',00','');

s:=ansireplacetext(s,',00','');

p:= pos('Y',b );

bp:=strtofloat(leftstr(b,p-1)) ;

p:= pos('Y',s );

sp:=strtofloat(leftstr(s,p-1)) ;


if sp >bp then

begin

try

dbnavigator2.BtnClick(nbpost);

```

```

query3.Close;

query3.SQL.Clear;

query3.SQL.Add('select sernum,name,brand,buyprice,speci,sellprice from
db\stock where invono = '"+dbedit4.text+"'");

query3.ExecSQL ;

query3.Open;


cost;

close2;

dbgrid2.Visible :=true;

bitbtn9.Enabled :=false;

    except

on EDBEngineError do

begin

    application.MessageBox(' This Serial Number is Used Before'+#13+
        ' Please Enter Another ID... ','Error',mb_OK+mb_IconStop);


    dbedit5.SetFocus ;

end;

end;

end

```

```

else

begin

showmessage('Selling Price Should Be Greater Than Buying Price');

dbedit9.SetFocus ;

end ;

end

else

begin

if dbedit11.Text ='' then

begin

showmessage('please Enter Buying Price');

dbedit11.SetFocus ;

end

else

begin

showmessage('please Enter Selling Price');

dbedit9.SetFocus ;

end;

end;

end;

```



end;

procedure Tpurchase.BitBtn11Click(Sender: TObject);

begin

dbnavigator1.BtnClick(nbedit);

dbedit8.Enabled :=true;

dbedit8.Text := label11.Caption ;

dbedit8.Enabled := False;

dbnavigator1.BtnClick(nbpost);

close2;

panel2.Enabled :=false;

bitbtn9.Enabled :=false;

end;

procedure Tpurchase.BitBtn12Click(Sender: TObject);

begin

```

dbnavigator2.BtnClick(nbdelete);

query3.Refresh;

end;

procedure Tpurchase.DBEdit11KeyPress(Sender: TObject; var Key: Char);
begin
    if not((key in ['0'..'9'])or(key=#8) or (key=#13)) then
        begin
            key:=#0;

            messagedlg('Please Enter a Numeric Character ',mtwarning,[mbok],0);

            end ;
        end;

    procedure Tpurchase.DBEdit9KeyPress(Sender: TObject; var Key: Char);
    begin
        if not((key in ['0'..'9'])or(key=#8) or (key=#13)) then
            begin
                key:=#0;

                messagedlg('Please Enter a Numeric Character ',mtwarning,[mbok],0);

                end;
            end;

        procedure Tpurchase.BitBtn13Click(Sender: TObject);

```

```

var y:word;

begin

if f=1 then

begin

y:=application.MessageBox('Last invoice information was not saved'+#13+

' Do you Want to exit without saving ?'+#13+

",'Warning',MB_YESNO + MB_IconExclamation);

if (y=IDYES) then

begin

dbnavigator1.BtnClick(nbcancel);

close;

end;

if (y=IDNO) then

begin

end;

end

end

else

close;

end;

end.

SELLING FORM:

```

unit Unit4; TDBNavigator

BtnEdit2: TDBEdit;

interface TDBNavigator

BtnEdit4: TDBEdit;

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, ExtCtrls, Grids, DBGrids, StdCtrls, Buttons, DBCtrls, Mask, DB,  
DBTables, ComCtrls, strutils;

type

Tpurchase = class(TForm)

Panel1: TPanel;

Table1: TTable;

Table2: TTable;

DataSource1: TDataSource;

DataSource2: TDataSource;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

DBEdit1: TDBEdit;

DBNavigator1: TDBNavigator;

BitBtn1: TBitBtn;  
BitBtn2: TBitBtn;  
BitBtn3: TBitBtn;  
BitBtn4: TBitBtn;  
DBGrid1: TDBGrid;  
DBLookupComboBox1: TDBLookupComboBox;  
Table3: TTable;  
DataSource3: TDataSource;  
DateTimePicker1: TDateTimePicker;  
DateTimePicker2: TDateTimePicker;  
DBEdit2: TDBEdit;  
DBEdit3: TDBEdit;  
DBEdit4: TDBEdit;  
BitBtn5: TBitBtn;  
BitBtn6: TBitBtn;  
DBNavigator2: TDBNavigator;  
Label5: TLabel;  
Label6: TLabel;  
Label7: TLabel;  
Label8: TLabel;  
Label9: TLabel;  
DBEdit5: TDBEdit;



DBEdit6: TDBEdit;  
DBEdit7: TDBEdit;  
DBEdit10: TDBEdit;  
DBEdit11: TDBEdit;  
BitBtn7: TBitBtn;  
DBLookupComboBox2: TDBLookupComboBox;  
DBLookupComboBox3: TDBLookupComboBox;  
Query1: TQuery;  
Query2: TQuery;  
DataSource4: TDataSource;  
DataSource5: TDataSource;  
BitBtn8: TBitBtn;  
BitBtn9: TBitBtn;  
BitBtn10: TBitBtn;  
BitBtn11: TBitBtn;  
Label10: TLabel;  
Label11: TLabel;  
DBGrid2: TDBGrid;  
Query3: TQuery;  
DataSource6: TDataSource;  
Query3Buyprice: TCurrencyField;  
Query3Sernum: TFloatField;

```
Query3Name: TStringField;

DBEdit8: TDBEdit;

Label12: TLabel;

panel2: TPanel;

BitBtn12: TBitBtn;

DBEdit9: TDBEdit;

Label13: TLabel;

Query3Sellprice: TCurrencyField;

Panel3: TPanel;

BitBtn13: TBitBtn;

procedure BitBtn1Click(Sender: TObject);

procedure closem;

procedure open;

procedure FormCreate(Sender: TObject);

procedure BitBtn3Click(Sender: TObject);

procedure BitBtn4Click(Sender: TObject);

procedure DateTimePicker1Change(Sender: TObject);

procedure DateTimePicker2Change(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

procedure BitBtn5Click(Sender: TObject);

procedure BitBtn6Click(Sender: TObject);

procedure BitBtn7Click(Sender: TObject);
```

```

procedure BitBtn8Click(Sender: TObject);

procedure DBLookupComboBox2Click(Sender: TObject);

procedure DBLookupComboBox3Click(Sender: TObject);

procedure DBLookupComboBox2CloseUp(Sender: TObject);

procedure BitBtn10Click(Sender: TObject);

procedure BitBtn9Click(Sender: TObject);

procedure BitBtn11Click(Sender: TObject);

procedure cost;

procedure open2;

procedure close2;

procedure BitBtn12Click(Sender: TObject);

procedure DBEdit11KeyPress(Sender: TObject; var Key: Char);

procedure DBEdit9KeyPress(Sender: TObject; var Key: Char);

procedure BitBtn13Click(Sender: TObject);

private
    { Private declarations }

public
    { Public declarations }

end;

var
    purchase: Tpurchase;

```

f:real;

i:real;

bp,sp:real;

implementation

uses Unit3, Unit5;

{ \$R \*.dfm }

procedure Tpurchase.cost ;

var a: integer;

begin

i:=0;

query3.First ;

for a := 1 to query3.RecordCount do

begin

i:=i+ query3.buyprice.AsFloat;

query3.Next ;

end;

label11.caption:=floattostr(i);

end;

procedure Tpurchase.closem;

begin

```

dbedit1.Enabled := False;

dbedit2.Enabled := False;

dbedit3.Enabled := False;

dbedit8.Enabled := False;

i:=0;

dblookupcombobox1.Enabled := False;

datetimepicker1.Enabled := False;

datetimepicker2.Enabled := False;

end;

procedure Tpurchase.close2;

begin

dbedit5.Enabled := False;

dbedit10.Enabled := False;

dbedit11.Enabled := False;

dbedit9.Enabled :=false;

dblookupcombobox2.Enabled := False;

dblookupcombobox3.Enabled := False;

end;

procedure Tpurchase.open2;

```



begin

dbedit5.Enabled := True;

dbedit10.Enabled := True;

dbedit11.Enabled := True;

dbedit9.Enabled :=true;

dblookupcombobox2.Enabled := True;

dblookupcombobox3.Enabled := True;

end;

procedure Tpurchase.open;

begin

dbedit1.Enabled := True;

dbedit2.Enabled := True;

dbedit3.Enabled := True;

//dbedit8.Enabled := True;

dblookupcombobox1.Enabled := True;

datetimepicker1.Enabled := True;

datetimepicker2.Enabled := True;

end;

```

procedure Tpurchase.BitBtn1Click(Sender: TObject);
begin
    f:=1;
    open;
    dbnavigator1.BtnClick(nbinsert);
    dbedit1.SetFocus ;
end;

```

```

procedure Tpurchase.FormCreate(Sender: TObject);
begin
    close2;
    panel2.Enabled := false;
    f:=0;
    bp:=0;
    sp:=0;
    closem;
end;

```

```

procedure Tpurchase.BitBtn3Click(Sender: TObject);
begin
    if dbedit1.Text= " then

```

```

begin
    showmessage('Please Enter Invoice No');
    dbedit1.SetFocus ;
end

else if dbedit2.Text= " " then
    begin
        showmessage('Please Enter Buying Date');
    end

    else if dbedit3.Text= " " then
        begin
            showmessage('Please Enter Payment Date');
        end

    else if f=1 then
        begin

        try

            dbnavigator1.BtnClick(nbpost);
            panel2.Enabled :=true;
            bitbtn7.Click ;

            dbedit4.Text:=dbedit1.Text ;
            dbedit6.Text:=dbedit2.Text ;

```

```

dbedit7.Text:=dblookupcombobox1.Text ;

closem;

f:=0;

//bitbtn7.Click ;

except

on EDBEngineError do

begin

application.MessageBox(' This Invoices NO is Used Before'+#13+

        ' Please Enter Another NO... ','Error',mb_OK+mb_IconStop);

        dbedit1.SetFocus ;

        dbedit1.Clear ;

        end;

        end ;

end

else

showmessage('For new invoice please click new invoice button'+#13+

        'For update invoice please click update invoice button');

end;

procedure Tpurchase.BitBtn4Click(Sender: TObject);

begin

```

f:=0;

dbnavigator1.BtnClick(nbcancel);

closem;

end;

procedure Tpurchase.DateTimePicker1Change(Sender: TObject);

begin

datetimepicker1.Checked:=true;

dbedit2.Text:=datetostr(Datetimepicker1.Date);

end;

procedure Tpurchase.DateTimePicker2Change(Sender: TObject);

begin

datetimepicker2.Checked:=true;

dbedit3.Text:=datetostr(Datetimepicker2.Date);

end;

procedure Tpurchase.BitBtn2Click(Sender: TObject);

begin

f:=1;

open;

dbnavigator1.BtnClick(nbedit);



end;

procedure Tpurchase.BitBtn5Click(Sender: TObject);

begin

application.Terminate ;

end;

procedure Tpurchase.BitBtn6Click(Sender: TObject);

begin

supplier.Show;

supplier.TabSheet1.Show;

end;

procedure Tpurchase.BitBtn7Click(Sender: TObject);

begin

open2;

dbnavigator2.BtnClick(nbinsert);

dbedit4.Text:=dbedit1.Text ;

dbedit6.Text:=dbedit2.Text ;

dbedit7.Text:=dblookupcombobox1.Text ;

bitbtn9.Enabled :=true;

dbedit5.SetFocus ;

end;

procedure Tpurchase.BitBtn8Click(Sender: TObject);

begin

quantity.show;

end;

procedure Tpurchase.DBLookupComboBox2Click(Sender: TObject);

begin

query1.Close;

query1.Open;

query2.Close;

query2.Open;

end;

procedure Tpurchase.DBLookupComboBox3Click(Sender: TObject);

begin

query1.Refresh;

query2.Refresh ;

end;

begin

```

procedure Tpurchase.DBLookupComboBox2CloseUp(Sender: TObject);
begin
    query2.Close;

    query2.SQL.Clear;

    query2.SQL.Add('select distinct(brand) from db\quantity where name =
    '"+dblookupcombobox2.text+"'");

    query2.Open;
end;

```

```

procedure Tpurchase.BitBtn10Click(Sender: TObject);
begin
    bitbtn9.Enabled :=false;

    dbnavigator2.BtnClick(nbcancel);

    close2;

end;

```

```

procedure Tpurchase.BitBtn9Click(Sender: TObject);
var p:integer;
b,s:string;
begin
    if dbedit5.Text =" then
begin

```

```

showmessage('Please Enter The Serial Number');

dbedit5.SetFocus;

end

else if (dbedit11.Text <> "") and (dbedit9.Text <> "") then

    begin

        b:=ansireplacetext(dbedit11.text, '.', '');

        s:=ansireplacetext(dbedit9.text, '.', '');

        b:=ansireplacetext(b, ',00', '');

        s:=ansireplacetext(s, ',00', '');

        p:= pos('Y',b );

        bp:=strtofloat(leftstr(b,p-1)) ;

        p:= pos('Y',s );

        sp:=strtofloat(leftstr(s,p-1)) ;

        if sp > bp then

            begin

                try

                    dbnavigator2.BtnClick(nbpost);

                query3.Close;

                query3.SQL.Clear;

```

```

query3.SQL.Add('select sernum,name,brand,buyprice,speci,sellprice from
db\stock where invono = '"+dbedit4.text+"'");

query3.ExecSQL ;

query3.Open;

cost;

close2;

dbgrid2.Visible :=true;

bitbtn9.Enabled :=false;

    except

on EDBEngineError do

begin

    application.MessageBox(' This Serial Number is Used Before'+#13+

        ' Please Enter Another ID... ','Error',mb_OK+mb_IconStop);

    dbedit5.SetFocus ;

end;

end;

end

else

begin

showmessage('Selling Price Should Be Greater Than Buying Price');

```



```

        dbedit9.SetFocus ;

    end ;

end

else

begin

    if dbedit11.Text =" then

        begin

            showmessage('please Enter Buying Price');

            dbedit11.SetFocus ;

        end

    else

        begin

            showmessage('please Enter Selling Price');

            dbedit9.SetFocus ;

        end;

    end;

end;

end;

```

```
procedure Tpurchase.BitBtn11Click(Sender: TObject);
```

```
begin
```

```
dbnavigator1.BtnClick(nbedit);
```

```
dbedit8.Enabled :=true;
```

```
dbedit8.Text := label11.Caption ;
```

```
dbedit8.Enabled := False;
```

```
dbnavigator1.BtnClick(nbpost);
```

```
close2;
```

```
panel2.Enabled :=false;
```

```
bitbtn9.Enabled :=false;
```

```
end;
```

```
procedure Tpurchase.BitBtn12Click(Sender: TObject);
```

```
begin
```

```
dbnavigator2.BtnClick(nbdelete);
```

```
query3.Refresh;
```

```
end;
```

```
procedure Tpurchase.DBEdit11KeyPress(Sender: TObject; var Key: Char);
```

```
begin
```

```
if not((key in ['0'..'9'])or(key=#8) or (key=#13)) then
```

```
begin
```

```
key:=#0;
```

```
messagedlg('Please Enter a Numeric Character ',mtwarning,[mbok],0);
```

```
end ;
```

```
end;
```

```
procedure Tpurchase.DBEdit9KeyPress(Sender: TObject; var Key: Char);
```

```
begin
```

```
if not((key in ['0'..'9'])or(key=#8) or (key=#13)) then
```

```
begin
```

```
key:=#0;
```

```
messagedlg('Please Enter a Numeric Character ',mtwarning,[mbok],0);
```

```
end;
```

```
end;
```

```
procedure Tpurchase.BitBtn13Click(Sender: TObject);
```

```

var y:word;
begin
if f=1 then
begin
y:=application.MessageBox('Last invoice information was not saved'+#13+
' Do you Want to exit without saving ?'+#13+
', 'Warning', MB_YESNO + MB_IconExclamation);
if (y=IDYES) then
begin
dbnavigator1.BtnClick(nbcancel);
close;
end;
if (y=IDNO) then
begin
end;
end;
end;
end.

```

## CUSTOMER ACCOUNTS FORM:

unit Unit8;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, ComCtrls, StdCtrls, DB, DBTables, Mask, DBCtrls, strutils,  
Grids, DBGrids, Buttons, ExtCtrls;

type

Tcustacc = class(TForm)

PageControl1: TPageControl;

TabSheet1: TTabSheet;

Query1: TQuery;

DataSource1: TDataSource;

ComboBox1: TComboBox;

Label1: TLabel;

Query2: TQuery;

DataSource2: TDataSource;

Label2: TLabel;

Label3: TLabel;



Label4: TLabel;  
Label5: TLabel;  
Label6: TLabel;  
Label7: TLabel;  
Label8: TLabel;  
DBEdit1: TDBEdit;  
DBEdit4: TDBEdit;  
DBEdit5: TDBEdit;  
DBEdit6: TDBEdit;  
DBEdit7: TDBEdit;  
DBEdit2: TDBEdit;  
DBEdit3: TDBEdit;  
Edit1: TEdit;  
Label9: TLabel;  
DBGrid1: TDBGrid;  
Edit2: TEdit;  
Edit3: TEdit;  
Label10: TLabel;  
Label11: TLabel;  
BitBtn1: TBitBtn;  
Query1custid: TFloatField;  
Query1custname: TStringField;

Query1custsurname: TStringField;  
BitBtn2: TBitBtn;  
BitBtn3: TBitBtn;  
Query2sinvono: TFloatField;  
Query2custid: TFloatField;  
Query2selldate: TDateField;  
Query2paydate: TDateField;  
Query2amount: TCurrencyField;  
Query2close: TStringField;  
Query2custname: TStringField;  
Query2custsurname: TStringField;  
Edit4: TEdit;  
Label12: TLabel;  
DateTimePicker1: TDateTimePicker;  
BitBtn4: TBitBtn;  
Panel1: TPanel;  
BitBtn5: TBitBtn;  
BitBtn6: TBitBtn;  
Label14: TLabel;  
Label15: TLabel;  
BitBtn7: TBitBtn;  
BitBtn8: TBitBtn;

Panel2: TPanel;  
Label13: TLabel;  
BitBtn9: TBitBtn;  
BitBtn10: TBitBtn;  
Label16: TLabel;  
Label17: TLabel;  
TabSheet2: TTabSheet;  
DBGrid2: TDBGrid;  
BitBtn11: TBitBtn;  
Label18: TLabel;  
Edit5: TEdit;  
Query3: TQuery;  
DataSource3: TDataSource;  
DBLookupComboBox1: TDBLookupComboBox;  
Label19: TLabel;  
Edit6: TEdit;  
Edit7: TEdit;  
BitBtn12: TBitBtn;  
BitBtn13: TBitBtn;  
DateTimePicker2: TDateTimePicker;  
DateTimePicker3: TDateTimePicker;  
Label20: TLabel;

Label21: TLabel;  
Label22: TLabel;  
Label23: TLabel;  
Panel3: TPanel;  
Panel4: TPanel;  
Label24: TLabel;  
Label25: TLabel;  
Label26: TLabel;  
BitBtn14: TBitBtn;  
Label27: TLabel;  
Label28: TLabel;  
BitBtn15: TBitBtn;  
BitBtn16: TBitBtn;  
BitBtn17: TBitBtn;  
BitBtn18: TBitBtn;  
BitBtn19: TBitBtn;  
BitBtn20: TBitBtn;  
BitBtn21: TBitBtn;  
  
procedure FormCreate(Sender: TObject);  
  
procedure ComboBox1Change(Sender: TObject);  
  
procedure open;  
  
procedure closem;

```
procedure BitBtn1Click(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

procedure DateTimePicker1Change(Sender: TObject);

procedure BitBtn3Click(Sender: TObject);

procedure BitBtn4Click(Sender: TObject);

procedure BitBtn7Click(Sender: TObject);

procedure BitBtn5Click(Sender: TObject);

procedure BitBtn8Click(Sender: TObject);

procedure BitBtn6Click(Sender: TObject);

procedure BitBtn9Click(Sender: TObject);

procedure BitBtn10Click(Sender: TObject);

procedure PageControl1Change(Sender: TObject);

procedure BitBtn11Click(Sender: TObject);

procedure DBLookupComboBox1CloseUp(Sender: TObject);

procedure BitBtn13Click(Sender: TObject);

procedure DateTimePicker2Change(Sender: TObject);

procedure DateTimePicker3Change(Sender: TObject);

procedure BitBtn12Click(Sender: TObject);

procedure BitBtn15Click(Sender: TObject);

procedure BitBtn16Click(Sender: TObject);

procedure BitBtn17Click(Sender: TObject);

procedure BitBtn18Click(Sender: TObject);
```



```
procedure BitBtn19Click(Sender: TObject);
```

```
procedure BitBtn14Click(Sender: TObject);
```

```
procedure BitBtn20Click(Sender: TObject);
```

```
procedure BitBtn21Click(Sender: TObject);
```

```
private
```

```
{ Private declarations }
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
var
```

```
custacc: Tcustacc;
```

```
c:string;
```

```
ta:real;
```

```
wa:real;
```

```
ins:integer;
```

```
implementation
```

```
{ $R *.dfm }
```

```
procedure Tcustacc.open ;
```

begin

dbedit1.Visible :=true;

dbedit2.Visible :=true;

dbedit3.Visible :=true;

dbedit4.Visible :=true;

dbedit5.Visible :=true;

dbedit6.Visible :=true;

dbedit7.Visible :=true;

dbgrid1.Visible :=true;

edit4.Clear ;

end;

procedure Tcustacc.closem ;

begin

dbedit1.Visible :=false;

dbedit2.Visible :=false;

dbedit3.Visible :=false;

dbedit4.Visible :=false;

dbedit5.Visible :=false;

dbedit6.Visible :=false;

dbedit7.Visible :=false;

dbgrid1.Visible :=false;

```

edit4.Clear ;

end;

procedure Tcustacc.FormCreate(Sender: TObject);

var

a : double;

b : string;

begin

closem;

combobox1.Items.Clear ;

DataSource1.DataSet.first;

while not(DataSource1.DataSet.eof) do

begin

a := DataSource1.DataSet.FieldValues['custid'];

b := floattostr(a);

combobox1.Items.add(b+'

'+DataSource1.DataSet.FieldValues['custname']+'

'+DataSource1.DataSet.FieldValues['custsurname']));

DataSource1.DataSet.next;

end ;

```

end;

procedure Tcustacc.ComboBox1Change(Sender: TObject);

var

p:integer;

n:string;

begin

n:='NO';

p:= pos(' ',combobox1.Text );

c:=leftstr(combobox1.text,p-1);

query2.Close;

query2.SQL.Clear;

query2.SQL.Add('select

s.sinvono,s.custid,s.selldate,s.paydate,s.amount,s.close,c.custname,c.custsurnam

e from db\sinvo s , db\cust c where custid='+c+' and s.custid=c.custid and

s.close='' +edit1.text+'' ');

query2.Open;

if query2.RecordCount < 1 then

begin

showmessage('This Customer Have NO Debt');

closem;

end

else

begin

query2.Edit;

open;

end;

end;

procedure Tcustacc.BitBtn1Click(Sender: TObject);

begin

if (edit2.text<>") or (edit3.text<>") then

begin

query1.Close;

query1.SQL.Clear;

query1.SQL.Add('select \* from db\cust where');

if edit2.text<>" then

begin

query1.SQL.Add(' custname = "'+edit2.text+'"');

end;



```

if edit3.text<>" then

begin
    query1.SQL.Add(' custsurname = '"+edit3.Text+"' ')

end;

query1.Open;

if query1.RecordCount <> 0 then

begin

    c:=query1.custid.AsString ;

    query2.Close;

    query2.SQL.Clear;

    query2.SQL.Add('select

s.sinvono,s.custid,s.selldate,s.paydate,s.amount,s.close,c.custname,c.custsurnam

e from db\sinvo s , db\cust c where custid='+c+' and s.custid=c.custid and

s.close='"+edit1.text+"' ');

    query2.Open;

    if query2.RecordCount < 1 then

begin

    showmessage('This Customer Have NO Debt');

    closem;

end

else

```

```

begin
    query2.Edit;
    open;
    end;
end
else
begin
    showmessage('There Is NO Record Like This');
    edit2.Clear ;
    edit3.Clear ;
    edit2.SetFocus ;
    closem;
    end;
end
else
begin
    showmessage('Please Enter A Criteria For Search');
    edit2.SetFocus ;
    closem;
    end;
end;
begin

```

```

procedure Tcustacc.BitBtn2Click(Sender: TObject);

begin
    if query2.RecordCount < 1 then
    begin
        showmessage('Please Select A Customer');
        edit2.SetFocus ;
    end
    else if edit4.Text = '' then
    begin
        showmessage('Please Enter Payment Date');
        edit4.setfocus;
    end
    else
    begin
        query2.Edit ;
        query2close.AsString:='YES';
        query2.Post ;
    end;
end;

procedure Tcustacc.DateTimePicker1Change(Sender: TObject);

begin

```

```
datetimepicker1.Checked :=true;
```

```
edit4.Text:=datetostr(datetimepicker1.Date );
```

```
end;
```

```
procedure Tcustacc.BitBtn3Click(Sender: TObject);
```

```
begin
```

```
closem;
```

```
end;
```

```
procedure Tcustacc.BitBtn4Click(Sender: TObject);
```

```
begin
```

```
edit4.Text:=datetostr(now);
```

```
end;
```

```
procedure Tcustacc.BitBtn7Click(Sender: TObject);
```

```
begin
```

```
ins:=0;
```

```
ta:=0;
```

```
query2.First ;
```

```
while not(DataSource2.DataSet.eof) do
```

```
begin
```

```

ta :=ta+strtofloat( DataSource2.DataSet.FieldValues['amount']);

DataSource2.DataSet.next;

ins:=ins+1;

end ;

label15.Caption :=floattostr(ta)+' YTL';

label17.Caption :=inttostr(ins);

end;

```

```

procedure Tcustacc.BitBtn5Click(Sender: TObject);

begin

ta :=ta+strtofloat( DataSource2.DataSet.FieldValues['amount']);

ins:=ins+1;

label15.Caption:=floattostr(ta)+' YTL';

label17.Caption :=inttostr(ins);

end;

```

```

procedure Tcustacc.BitBtn8Click(Sender: TObject);

begin

ins:=0;

ta:=0;

label15.Caption:='0';

label17.Caption :='0';

```



end;

procedure Tcustacc.BitBtn6Click(Sender: TObject);

begin

ta :=ta-strtofloat( DataSource2.DataSet.FieldValues['amount']);

ins:=ins-1;

label17.Caption :=inttostr(ins);

label15.Caption:=floattostr(ta)+' YTL';

end;

procedure Tcustacc.BitBtn9Click(Sender: TObject);

begin

label15.Caption :='-----';

label17.Caption :='--';

panel1.Visible :=false;

ta:=0;

ins:=0;

end;

procedure Tcustacc.BitBtn10Click(Sender: TObject);

```

begin
panel1.Visible :=true;
end;

procedure Tcustacc.PageControl1Change(Sender: TObject);
begin
query2.close ;
query2.SQL.Clear ;
query2.SQL.Add('select
s.sinvono,s.custid,s.selldate,s.paydate,s.amount,s.close,c.custname,c.custsurnam
e from db\sinvo s , db\cust c where s.custid=c.custid');
query2.ExecSQL ;
query2.Open;

procedure Tcustacc.BitBtn1Click(Sender: TObject);
begin
if edit5.Text <>"" then
begin

```

```

query2.Close;

query2.SQL.Clear;

query2.SQL.Add('select
s.sinvono,s.custid,s.selldate,s.paydate,s.amount,s.close,c.custname,c.custsurnam
e from db\sinvo s , db\cust c where s.custid=c.custid and
sinvono="'+edit5.Text+'" ');

query2.Open

end

else

begin

showmessage('Please Enter An InVoice Number')

end;

end;

end;

procedure Tcustacc.DBLookupComboBox1CloseUp(Sender: TObject);

begin

edit5.Text :=dblookupcombobox1.Text ;

end;

end;

procedure Tcustacc.BitBtn13Click(Sender: TObject);

begin

query2.Close;

```

```
query2.SQL.Clear;  
query2.SQL.Add('select  
s.sinvono,s.custid,s.selldate,s.paydate,s.amount,s.close,c.custname,c.custsurnam  
e');
```

```
query2.SQL.Add('from db\sinvo s , db\cust c ');  
query2.SQL.Add('where s.custid=c.custid');  
query2.Open;  
end;
```

```
procedure Tcustacc.DateTimePicker2Change(Sender: TObject);  
begin  
edit6.Text:=datetostr(datetimepicker2.Date);  
end;
```

```
procedure Tcustacc.DateTimePicker3Change(Sender: TObject);  
begin  
edit7.Text:=datetostr(datetimepicker3.Date);  
end;
```

```
procedure Tcustacc.BitBtn12Click(Sender: TObject);  
begin
```

```

if edit6.Text <> "" then
begin
    query2.Close;

    query2.SQL.Clear;

    query2.SQL.Add('select
s.sinvono,s.custid,s.selldate,s.paydate,s.amount,s.close,c.custname,c.custsumnam
e from db\sinvo s , db\cust c where s.custid=c.custid and
paydate="'+edit6.Text+'" ');

    query2.Open ;

    if edit7.Text <> "" then
begin
        query2.Close;

        query2.SQL.Clear;

        query2.SQL.Add('select
s.sinvono,s.custid,s.selldate,s.paydate,s.amount,s.close,c.custname,c.custsumnam
e from db\sinvo s , db\cust c where s.custid=c.custid and
paydate>="'+edit6.Text+'" and paydate <= '"+edit7.text+'" ');

        query2.Open

    end;

end;

```



end;

procedure Tcustacc.BitBtn15Click(Sender: TObject);

begin

ins:=0;

ta:=0;

query2.First ;

while not(DataSource2.DataSet.eof) do

begin

ta :=ta+strtofloat( DataSource2.DataSet.FieldValues['amount']);

DataSource2.DataSet.next;

ins:=ins+1;

end ;

label27.Caption :=floattostr(ta)+' YTL';

label28.Caption :=inttostr(ins);

end;

procedure Tcustacc.BitBtn16Click(Sender: TObject);

begin

ins:=0;

ta:=0;

```
label27.Caption:=' 0';
```

```
label28.Caption :=' 0';
```

```
end;
```

```
procedure Tcustacc.BitBtn17Click(Sender: TObject);
```

```
begin
```

```
ta :=ta+strtofloat( DataSource2.DataSet.FieldValues['amount']);
```

```
ins:=ins+1;
```

```
label27.Caption:=floattostr(ta)+' YTL';
```

```
label28.Caption :=inttostr(ins);
```

```
end;
```

```
procedure Tcustacc.BitBtn18Click(Sender: TObject);
```

```
begin
```

```
ta :=ta-strtofloat( DataSource2.DataSet.FieldValues['amount']);
```

```
ins:=ins-1;
```

```
label28.Caption :=inttostr(ins);
```

```
label27.Caption:=floattostr(ta)+' YTL';
```

```
end;
```

```
procedure Tcustacc.BitBtn19Click(Sender: TObject);
```

```
begin
```

label27.Caption := '-----';

label28.Caption := '--';

panel3.Visible := false;

ta:=0;

ins:=0;

end;

procedure Tcustacc.BitBtn14Click(Sender: TObject);

begin

panel3.Visible := true;

end;

procedure Tcustacc.BitBtn20Click(Sender: TObject);

begin

close;

end;

procedure Tcustacc.BitBtn21Click(Sender: TObject);

begin

close;

end;

end.

## SUPPLIER ACCOUNTS FORM:

```
unit Unit9;

interface

uses

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ComCtrls, DB, DBTables, Grids, DBGrids, StdCtrls, Buttons,
    DBCtrls, ExtCtrls;

type

    Tsuppacc = class(TForm)
        PageControl1: TPageControl;
        TabSheet1: TTabSheet;
        Query1: TQuery;
        DataSource1: TDataSource;
        DataSource2: TDataSource;
        DBGrid1: TDBGrid;
        DBGrid2: TDBGrid;
        Query2: TQuery;
        BitBtn1: TBitBtn;
        Query1Invono: TFloatField;
```

Query1Suppname: TStringField;  
Query1Buydate: TDateField;  
Query1Paydate: TDateField;  
Query1Amount: TCurrencyField;  
Query1Close: TStringField;  
Edit1: TEdit;  
BitBtn2: TBitBtn;  
Edit2: TEdit;  
Label1: TLabel;  
TabSheet2: TTabSheet;  
DBGrid3: TDBGrid;  
DBLookupComboBox1: TDBLookupComboBox;  
Label2: TLabel;  
Label3: TLabel;  
Label4: TLabel;  
BitBtn3: TBitBtn;  
Label5: TLabel;  
Label6: TLabel;  
panell1: TPanel;  
Label7: TLabel;  
BitBtn4: TBitBtn;  
BitBtn5: TBitBtn;



```

procedure BitBtn1Click(Sender: TObject);

procedure DBGrid1CellClick(Column: TColumn);

procedure BitBtn2Click(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure DBLookupComboBox1CloseUp(Sender: TObject);

procedure BitBtn4Click(Sender: TObject);

procedure BitBtn3Click(Sender: TObject);

procedure TabSheet1Show(Sender: TObject);

procedure BitBtn5Click(Sender: TObject);

private
{ Private declarations }

public
{ Public declarations }

end;

var
    suppacc: Tsuppacc;

implementation

uses Unit6;

```

```

{$R *.dfm}

end;

procedure Tsuppacc.BitBtn1Click(Sender: TObject);

begin
if edit1.Text <>"" then
begin
query1.Close;

query1.SQL.clear;

query1.SQL.Add('select * from db\pinvo where invono="'+edit1.Text+'" ');

query1.ExecSQL ;

query1.Open;

edit1.Text:=query1.invono.AsString ;

query2.Close;

query2.SQL.Clear ;

query2.SQL.Add('select * from db\stock where invono="'+edit1.text+'"');

query2.ExecSQL ;

query2.Open;

end
else
begin

showmessage('Please Enter Invoice NO ');

```

edit1.SetFocus ;

end;

end;

procedure Tsuppacc.DBGrid1CellClick(Column: TColumn);

begin

edit1.Text:=query1invono.AsString ;

query2.Close;

query2.SQL.Clear ;

query2.SQL.Add('select \* from db\stock where invono="'+edit1.text+'"');

query2.ExecSQL ;

query2.Open;

edit1.Clear ;

end;

procedure Tsuppacc.BitBtn2Click(Sender: TObject);

begin

query1.Close;

query1.SQL.clear;

query1.SQL.Add('select \* from db\pinvo where close="'+edit2.Text+'" ');

query1.ExecSQL ;

```

query1.Open;

end;

procedure Tsuppacc.FormCreate(Sender: TObject);
begin
edit1.Text:=query1invono.AsString ;

query2.Close;

query2.SQL.Clear ;

query2.SQL.Add('select * from db\stock where invono="'+edit1.text+'"');

query2.ExecSQL ;

query2.Open;

edit1.Clear ;

end;

procedure Tsuppacc.DBLookupComboBox1CloseUp(Sender: TObject);
begin

label4.Caption :=query1amount.AsString+' YTL' ;

end;

procedure Tsuppacc.BitBtn4Click(Sender: TObject);
begin

//bitbtn2.Click ;

close;

end;

```

```

procedure Tsuppacc.BitBtn3Click(Sender: TObject);
begin
    query1.Edit ;
    query1.close.AsString := 'YES';
    query1.Post;
end;

procedure Tsuppacc.TabSheet1Show(Sender: TObject);
begin
    if main.edit2.Text = 'User' then
    begin
        label5.Caption := 'Dear ' + main.edit1.Text;
        panel1.Visible := true;
    end;
end;

procedure Tsuppacc.BitBtn5Click(Sender: TObject);
begin
    close;
end;
end.

```



## PERSONNEL ACCOUNTS FORM:

```
unit Unit11;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, ComCtrls, StdCtrls, Buttons, Grids, DBGrids, DB, DBTables, Mask,  
DBCtrls, ExtCtrls;
```

```
type
```

```
Tperacc = class(TForm)
```

```
PageControl1: TPageControl;
```

```
TabSheet1: TTabSheet;
```

```
Query1: TQuery;
```

```
DataSource1: TDataSource;
```

```
DBGrid1: TDBGrid;
```

```
Label1: TLabel;
```

```
Label2: TLabel;
```

```
Edit1: TEdit;
```

```
Edit2: TEdit;
```

```
BitBtn1: TBitBtn;
```

Label3: TLabel;  
Label4: TLabel;  
Label5: TLabel;  
Label6: TLabel;  
Label7: TLabel;  
Label8: TLabel;  
DBEdit1: TDBEdit;  
Label9: TLabel;  
Label10: TLabel;  
DBEdit2: TDBEdit;  
DBEdit3: TDBEdit;  
DBEdit4: TDBEdit;  
Query1Ssn: TFloatField;  
Query1Name: TStringField;  
Query1Surname: TStringField;  
Query1Age: TFloatField;  
Query1Phone: TStringField;  
Query1Mail: TStringField;  
Query1Address: TStringField;  
Query1Job: TStringField;  
Query1Hdate: TDateField;  
Query1Salary: TCurrencyField;

Query1Photo: TGraphicField;  
Query1Paid: TCurrencyField;  
BitBtn2: TBitBtn;  
Edit3: TEdit;  
Edit4: TEdit;  
Edit5: TEdit;  
Edit6: TEdit;  
Panel1: TPanel;  
ComboBox1: TComboBox;  
ComboBox2: TComboBox;  
Query2: TQuery;  
DataSource2: TDataSource;  
BitBtn3: TBitBtn;  
BitBtn4: TBitBtn;  
Panel2: TPanel;  
Label11: TLabel;  
Label12: TLabel;  
Label13: TLabel;  
Label14: TLabel;  
Label15: TLabel;  
Edit7: TEdit;  
BitBtn5: TBitBtn;

```

Label17: TLabel;

Label18: TLabel;

BitBtn6: TBitBtn;

BitBtn7: TBitBtn;

Panel3: TPanel;

Label16: TLabel;

Query2COUNTOFssn: TIntegerField;

Query3: TQuery;

DataSource3: TDataSource;

BitBtn8: TBitBtn;

procedure BitBtn1Click(Sender: TObject);

procedure ComboBox1CloseUp(Sender: TObject);

procedure ComboBox2CloseUp(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

procedure BitBtn3Click(Sender: TObject);

procedure BitBtn4Click(Sender: TObject);

procedure BitBtn5Click(Sender: TObject);

procedure BitBtn6Click(Sender: TObject);

procedure BitBtn7Click(Sender: TObject);

procedure BitBtn8Click(Sender: TObject);

private

{ Private declarations }

```

```
public
{ Public declarations }

end;
```

```
var

peracc: Tperacc;

a,b:string;

s,g,e,p,ba,c,r:real;
```

```
implementation
```

```
{ $R *.dfm }
```

```
procedure Tperacc.BitBtn1Click(Sender: TObject);
```

```
begin
```

```
    bitbtn3.Enabled :=false;
```

```
    bitbtn4.Enabled :=false;
```

```
    if (edit1.text<>") or (edit2.Text <>") then
```

```
    begin
```

```
        query1.Close;
```



```
query1.SQL.Clear ;
```

```
query1.SQL.Add('select * from db\personnel where ');
```

```
if edit1.Text <> "" then
```

```
begin
```

```
query1.SQL.Add('name="'+edit1.Text+'"');
```

```
end;
```

```
if edit2.Text <> "" then
```

```
begin
```

```
query1.SQL.Add('surname="'+edit2.Text+'"');
```

```
end;
```

```
query1.Open;
```

```
if query1.RecordCount < 1 then
```

```
begin
```

```
showmessage('There Is NO Record Like This');
```

```
edit1.SetFocus ;
```

```
end;
```

```
end
```

```
else
```

```
begin
```

```
showmessage('Please Enter A Criteria For Search');
```

```
edit1.SetFocus ;
```

```
end;
```

```
end;
```

```
procedure Tperacc.ComboBox1CloseUp(Sender: TObject);
```

```
begin
```

```
if combobox1.ItemIndex = 0 then
```

```
begin
```

```
    a:='01.01';
```

```
    b:='31.01';
```

```
end
```

```
    else if combobox1.ItemIndex = 1 then
```

```
begin
```

```
    a:='01.02';
```

```
    b:='28.02';
```

```
end
```

```
else if combobox1.ItemIndex = 2 then
```

```
begin
```

```
    a:='01.03';
```

```
    b:='31.03';
```

```
end
```

```
else if combobox1.ItemIndex = 3 then
begin
    a:='01.04';
    b:='30.04';
end
else if combobox1.ItemIndex = 4 then
begin
    a:='01.05';
    b:='31.05';
end
else if combobox1.ItemIndex = 5 then
begin
    a:='01.06';
    b:='30.06';
end
else if combobox1.ItemIndex = 6 then
begin
    a:='01.07';
    b:='31.07';
end
else if combobox1.ItemIndex = 7 then
begin
```

```
a:='01.08';  
b:='31.08';  
  
end  
  
else if combobox1.ItemIndex = 8 then  
  
begin  
    a:='01.09';  
    b:='30.09';  
  
end  
  
else if combobox1.ItemIndex = 9 then  
  
begin  
    a:='01.10';  
    b:='31.10';  
  
end  
  
else if combobox1.ItemIndex = 10 then  
  
begin  
    a:='01.11';  
    b:='30.11';  
  
end  
  
else if combobox1.ItemIndex = 11 then  
  
begin  
    a:='01.12';  
    b:='31.12';
```

end ;

end;

procedure Tperacc.ComboBox2CloseUp(Sender: TObject);

begin

if combobox2.ItemIndex = 0 then

begin

a:=a+'.2005';

b:=b+'.2005';

end

else if combobox2.ItemIndex = 1 then

begin

a:=a+'.2006';

b:=b+'.2006';

end



```
else if combobox2.ItemIndex = 2 then
begin
    a:=a+'.2007';
    b:=b+'.2007';
end
else if combobox2.ItemIndex = 3 then
begin
    a:=a+'.2008';
    b:=b+'.2008';
end
else if combobox2.ItemIndex = 4 then
begin
    a:=a+'.2009';
    b:=b+'.2009';
end
else if combobox2.ItemIndex = 5 then
begin
    a:=a+'.2010';
    b:=b+'.2010';
end
else if combobox2.ItemIndex = 6 then
begin
```

```

a:=a+'.2011';
b:=b+'.2011';

end

else if combobox2.ItemIndex = 7 then

begin
a:=a+'.2012';
b:=b+'.2012';

end

else if combobox2.ItemIndex = 8 then

begin
a:=a+'.2013';
b:=b+'.2013';

end

else if combobox2.ItemIndex = 9 then

begin
a:=a+'.2014';
b:=b+'.2014';

end

else if combobox2.ItemIndex = 10 then

begin
a:=a+'.2015';
b:=b+'.2015';

```

```

end

else if combobox2.ItemIndex = 11 then

begin

    a:=a+'.2016';

    b:=b+'.2016';

end ;

end;

procedure Tperacc.BitBtn2Click(Sender: TObject);

var x,y:real;

begin

x:=0;

y:=0;

if length(a)<9 then

begin

showmessage('Please Select Month AND Year');

end

else

begin

query2.Close;

query2.SQL.Clear ;

```

```

query2.SQL.Add('select count(ssn) from db\attend where
ssn="'+query1.ssn.AsString+'" and dat >= "'+a+'" and dat <="'+b+'" and
abs="'+True+' ');

query2.Open;

g:=strtofloat(query2.countofssn.AsString );

p:=strtofloat(query1.paid.AsString );

eod5.Text=query1.paid.AsString+'%';

query3.Close;

query3.SQL.Clear ;

query3.SQL.Add('select perstart,perend from db\attend where
ssn='+query1.ssn.AsString+' and perend>="'+a+'" and perstart<="'+b+'");

query3.ExecSQL ;

query3.Open ;

```

```

DataSource3.DataSet.first;

while not(DataSource3.DataSet.eof) do
begin
x := (DataSource3.DataSet.FieldValues['perend'])-
(DataSource3.DataSet.FieldValues['perstart']);

y:=y+x+1;

DataSource3.DataSet.next;

end ;

```

```

query2.SQL.Add('select count(ssn) from db\attend where
ssn="'+query1.ssn.AsString+'" and dat >= "'+a+'" and dat <="'+b+'" and
abs="'+True+' ');
query2.Open;
g:=strtofloat(query2.countof.ssn.AsString );
p:=strtofloat(query1.paid.AsString );
edit3.Text=query1.paid.AsString + ' ' & g;

query3.Close;
query3.SQL.Clear ;
query3.SQL.Add('select perstart,perend from db\attend where
ssn='+query1.ssn.AsString +' and perend>="'+a+'" and perstart<="'+b+'");
query3.ExecSQL ;
query3.Open ;

DataSource3.DataSet.first;

while not(DataSource3.DataSet.eof) do
begin
x := (DataSource3.DataSet.FieldValues['perend'])-
(DataSource3.DataSet.FieldValues['perstart']);
y:=y+x+1;
DataSource3.DataSet.next;
end ;

```



```

g:=g+y;

edit3.Text := floattostr(g)+' DAY(S)';

s:=strtofloat(query1.salary.AsString );

e:=s-((s/30)*g);

edit4.Text:=floattostr(e)+' YTL';

p:=strtofloat(query1.paid.AsString );

edit5.Text:=query1.paid.AsString+' YTL';

ba:=e-p;

edit6.Text :=floattostr(ba)+' YTL';

bitbtn3.Enabled :=true;

bitbtn4.Enabled :=true;

end;

procedure Tperacc.BitBtn3Click(Sender: TObject);
begin
query1.Edit ;

```

```
query1paid.AsString := '0';
```

```
query1.Post;
```

```
end;
```

```
procedure Tperacc.BitBtn4Click(Sender: TObject);
```

```
begin
```

```
query2.Close;
```

```
query2.SQL.Clear ;
```

```
query2.SQL.Add('select count(ssn) from db\attend where
```

```
ssn="" + query1ssn.AsString + "" and dat >= "" + a + "" and dat <= "" + b + "" and
```

```
abs="" + 'false' + "" or permis="" + 'salaried' + ""');
```

```
query2.Open;
```

```
panel2.Visible := true;
```

```
c := strtoint(query2countofssn.AsString );
```

```
label13.Caption := floattostr(c);
```

```
r := ((s/30)*c);
```

```
label14.Caption := floattostr(r);
```

```
label18.Caption := floattostr((r-p));
```

```
end;
```

```

procedure Tperacc.BitBtn5Click(Sender: TObject);

var y:word;

begin
if edit7.Text <>"" then
begin
if strtofloat(edit7.text)<=s then
begin
if strtofloat(edit7.text)>strtofloat(label18.Caption) then
begin
y:=application.MessageBox('The Amount That You Will Give'+#13+
' Greater Then Deverse Amount'+#13+
'Are You Sure Give This Amount ?','Warning',MB_YESNO +
MB_IconExclamation);
if (y=IDYES) then
begin
query1.Edit;
query1paid.AsString :=edit7.Text;
query1.Post;
end;
if (y=IDNO) then
begin

```

```

edit7.Clear ;

edit7.SetFocus ;

end;

end

else

begin

query1.Edit;

query1paid.AsString :=edit7.Text;

query1.Post;

end;

end

else

begin

showmessage('The Amount Can not Be Greater Than Salary');

end;

end;

end

else

begin

Showmessage('Please Enter Advance Amount');

edit7.SetFocus ;

```

end;

end;

interface

procedure Tperacc.BitBtn6Click(Sender: TObject);

begin

edit7.Clear ; Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

bitbtn7.SetFocus ; Buttons, DB, DBTables, Data-

end; Dlg, Grids, DBGrids,

procedure Tperacc.BitBtn7Click(Sender: TObject);

begin

edit7.Clear ;

panel2.Visible :=false;

bitbtn2.Click ;

end;

procedure Tperacc.BitBtn8Click(Sender: TObject);

begin

close;

end;

end.



## PERSONNEL FORM:

unit Unit10;

interface TLabel;

uses TLabel;

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Buttons, DB, DBTables, ComCtrls, Mask, DBCtrls,  
ExtDlgs, Grids, DBGrids;

type

Tpersonnel = class(TForm)

PageControl1: TPageControl;

TabSheet1: TTabSheet;

Label1: TLabel;

Label2: TLabel;

Edit1: TEdit;

Edit2: TEdit;

BitBtn1: TBitBtn;

Query1: TQuery;

DataSource1: TDataSource;

Label3: TLabel;

Label4: TLabel;  
Label5: TLabel;  
Label6: TLabel;  
Label7: TLabel;  
Label8: TLabel;  
Label9: TLabel;  
Label10: TLabel;  
Label11: TLabel;  
DBImage1: TDBImage;  
DBEdit1: TDBEdit;  
DBEdit2: TDBEdit;  
DBEdit3: TDBEdit;  
DBEdit4: TDBEdit;  
DBEdit5: TDBEdit;  
DBEdit6: TDBEdit;  
DBEdit7: TDBEdit;  
DBEdit8: TDBEdit;  
DBEdit9: TDBEdit;  
Label12: TLabel;  
DBEdit10: TDBEdit;  
opd: TOpenPictureDialog;  
BitBtn2: TBitBtn;

BitBtn3: TBitBtn;

BitBtn4: TBitBtn;

BitBtn5: TBitBtn;

BitBtn6: TBitBtn;

BitBtn7: TBitBtn;

DBGrid1: TDBGrid;

BitBtn8: TBitBtn;

Label13: TLabel;

DBEdit11: TDBEdit;

BitBtn9: TBitBtn;

procedure BitBtn2Click(Sender: TObject);

procedure open;

procedure closem;

procedure FormCreate(Sender: TObject);

procedure BitBtn3Click(Sender: TObject);

procedure BitBtn7Click(Sender: TObject);

procedure BitBtn4Click(Sender: TObject);

procedure BitBtn8Click(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

procedure BitBtn5Click(Sender: TObject);

procedure BitBtn6Click(Sender: TObject);

procedure BitBtn9Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

personnel: Tpersonnel;

procedure

begin

implementation

{ \$R \*.dfm }

procedure Tpersonnel.open ;

begin

dbedit1.Enabled := True;

dbedit2.Enabled := True;

dbedit3.Enabled := True;

dbedit4.Enabled := True;

dbedit5.Enabled := True;

dbedit6.Enabled := True;

```
dbedit7.Enabled := True;  
dbedit8.Enabled := True;  
dbedit9.Enabled := True;  
dbedit10.Enabled := True;  
bitbtn2.Enabled := True;  
dbimage1.Enabled := True;  
end;
```

```
procedure Tpersonnel.closem ;
```

```
begin
```

```
dbedit1.Enabled := False;  
dbedit2.Enabled := False;  
dbedit3.Enabled := False;  
dbedit4.Enabled := False;  
dbedit5.Enabled := False;  
dbedit6.Enabled := False;  
dbedit7.Enabled := False;  
dbedit8.Enabled := False;  
dbedit9.Enabled := False;  
dbedit10.Enabled := False;  
bitbtn2.Enabled := False;  
dbimage1.Enabled := False;
```



end;

procedure Tpersonnel.BitBtn2Click(Sender: TObject);

begin

if opd.Execute then DBImage1.Picture.LoadFromFile(opd.filename);

end;

procedure Tpersonnel.FormCreate(Sender: TObject);

begin

closem;

end;

procedure Tpersonnel.BitBtn3Click(Sender: TObject);

begin

open;

query1.Insert;

dbedit1.SetFocus ;

end;

procedure Tpersonnel.BitBtn7Click(Sender: TObject);

begin

```

try

query1.Post;

showmessage('Personnel have been Recorded succesfully');

closem;


except

on EDatabaseError do

begin

application.MessageBox(' Please Select A Process Type'+#13+
        ' Then Click OKEY Button ','Error',mb_OK+mb_IconStop);


end;

end ;


end;

```

```
procedure Tpersonnel.BitBtn4Click(Sender: TObject);
```

```
begin
```

```
open;
```

```
query1.Edit;
```

```
end;
```

```
procedure Tpersonnel.BitBtn8Click(Sender: TObject);
```

```
begin
```

```
query1.Close;
```

```
query1.SQL.Clear ;
```

```
query1.SQL.Add('select * from db\personnel');
```

```
query1.Open;
```

```
end;
```

```
procedure Tpersonnel.BitBtn1Click(Sender: TObject);
```

```
begin
```

```
if (edit1.text<>") or (edit2.Text <>") then
```

```
begin
```

```
    query1.Close;
```

```
    query1.SQL.Clear ;
```

```
    query1.SQL.Add('select * from db\personnel where ');
```

```

if edit1.Text <> "" then
begin
query1.SQL.Add('name="'+edit1.Text+'" ');
end;

if edit2.Text <> "" then
begin
query1.SQL.Add('surname="'+edit2.Text+'" ');
end;
query1.Open;
if query1.RecordCount < 1 then
begin
showmessage('There Is NO Record Like This');
edit1.SetFocus ;
end;
end
else
begin
showmessage('Please Enter A Criteria For Search');
edit1.SetFocus ;
end;

```

end;

procedure Tpersonnel.BitBtn5Click(Sender: TObject);

begin

query1.Delete;

close;

end;

procedure Tpersonnel.BitBtn6Click(Sender: TObject);

begin

query1.Cancel ;

close;

end;

procedure Tpersonnel.BitBtn9Click(Sender: TObject);

begin

close;

end;

end.



## PERSONNEL ATTENDANCE FORM:

unit Unit12;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, DBTables, ComCtrls, StdCtrls, CheckLst, ExtCtrls, Buttons ,  
strutils,  
Grids, DBGrids;

type

Tperatt = class(TForm)

PageControl1: TPageControl;

TabSheet1: TTabSheet;

Query1: TQuery;

DataSource1: TDataSource;

Query2: TQuery;

DataSource2: TDataSource;

Label1: TLabel;

CheckListBox1: TCheckListBox;

ListBox1: TListBox;

RadioGroup1: TRadioGroup;  
RadioButton1: TRadioButton;  
RadioButton3: TRadioButton;  
Label2: TLabel;  
Label3: TLabel;  
BitBtn1: TBitBtn;  
BitBtn2: TBitBtn;  
BitBtn3: TBitBtn;  
TabSheet2: TTabSheet;  
DateTimePicker2: TDateTimePicker;  
Label5: TLabel;  
BitBtn4: TBitBtn;  
BitBtn5: TBitBtn;  
BitBtn6: TBitBtn;  
DBGrid1: TDBGrid;  
Query3: TQuery;  
DataSource3: TDataSource;  
BitBtn7: TBitBtn;  
Query3name: TStringField;  
Query3surname: TStringField;  
Query3phone: TStringField;  
Query3ssn: TFloatField;

Edit2: TEdit;  
TabSheet3: TTabSheet;  
Label4: TLabel;  
Label6: TLabel;  
Label7: TLabel;  
Edit1: TEdit;  
ComboBox1: TComboBox;  
DateTimePicker1: TDateTimePicker;  
DateTimePicker3: TDateTimePicker;  
DBGrid2: TDBGrid;  
Query4: TQuery;  
DataSource4: TDataSource;  
Label8: TLabel;  
Label9: TLabel;  
Label10: TLabel;  
Label11: TLabel;  
Label12: TLabel;  
Label13: TLabel;  
BitBtn8: TBitBtn;  
DBGrid3: TDBGrid;  
Query1Ssn: TFloatField;  
Query1Name: TStringField;

```

Query1Surname: TStringField;

Query1Perex: TFloatField;

BitBtn9: TBitBtn;

BitBtn10: TBitBtn;

BitBtn11: TBitBtn;

BitBtn12: TBitBtn;

procedure FormCreate(Sender: TObject);

procedure CheckListBox1ClickCheck(Sender: TObject);

procedure RadioButton1Click(Sender: TObject);

procedure RadioButton3Click(Sender: TObject);

procedure RadioButton2Click(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

procedure BitBtn3Click(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

procedure BitBtn4Click(Sender: TObject);

procedure BitBtn5Click(Sender: TObject);

procedure BitBtn6Click(Sender: TObject);

procedure PageControl1Change(Sender: TObject);

procedure BitBtn7Click(Sender: TObject);

procedure DateTimePicker3CloseUp(Sender: TObject);

procedure Edit1Change(Sender: TObject);

procedure DBGrid2CellClick(Column: TColumn);

```

```

procedure BitBtn8Click(Sender: TObject);

procedure TabSheet3Show(Sender: TObject);

procedure TabSheet1Show(Sender: TObject);

procedure BitBtn9Click(Sender: TObject);

procedure BitBtn10Click(Sender: TObject);

procedure BitBtn11Click(Sender: TObject);

procedure BitBtn12Click(Sender: TObject);

private
    { Private declarations }

public
    { Public declarations }

end;

var

peratt: Tperatt;

implementation

{$R *.dfm}

procedure Tperatt.FormCreate(Sender: TObject);

```



begin

edit2.Text:=datetostr(now);

end;

procedure Tperatt.CheckListBox1ClickCheck(Sender: TObject);

var

ind: integer;

begin

if (radiobutton1.Checked) or (radiobutton3.Checked) then

begin

ind:=checkboxlistbox1.ItemIndex ;

listbox1.Items.Add(' '+checkboxlistbox1.Items[ind] );

end

else

showmessage('Please Select An Attendance Type');

end;

```
procedure Tperatt.RadioButton1Click(Sender: TObject);  
begin
```

```
listbox1.Items.Clear ;
```

```
listbox1.Visible :=true;
```

```
end;
```

```
procedure Tperatt.RadioButton3Click(Sender: TObject);
```

```
begin
```

```
listbox1.Items.Clear ;
```

```
listbox1.Visible :=true;
```

```
end;
```

```
procedure Tperatt.RadioButton2Click(Sender: TObject);
```

```
begin
```

```
listbox1.Items.Clear ;
```

```
listbox1.Visible :=false;
```

```
edit1.Visible :=true;
```

```
label4.Visible := true;
```

```
combobox1.Visible :=true;
```

```
edit1.Clear ;
```

```
end; while (i <= 10)
```

```
begin
```

```
procedure Tperatt.BitBtn2Click(Sender: TObject);
```

```
begin
```

```
listbox1.Items.Delete(listbox1.ItemIndex);
```

```
end;
```

```
procedure Tperatt.BitBtn3Click(Sender: TObject);
```

```
begin
```

```
listbox1.Clear ;
```

```
end;
```

```
procedure Tperatt.BitBtn1Click(Sender: TObject);
```

```
var ic,i,p: integer;
```

```
s : double;
```

```
begin
```

```
if (radiobutton1.Checked) or (radiobutton3.Checked) then
```

```
begin
```

```
if (radiobutton1.Checked ) then
```

```

begin
i:=0;

while (i <=(listbox1.count-1)) do
begin
Datasource2.Edit;

datasource2.DataSet.Append;

datasource2.DataSet.FieldValues['dat']:=edit2.text;

p:= pos(' ',listbox1.Items.ValueFromIndex[i]);

s:=strtofloat(leftstr(listbox1.Items.ValueFromIndex[i],p-1)) ;

datasource2.dataset.fieldvalues['ssn']:=s;

datasource2.DataSet.FieldValues['abs']:=False';

datasource2.dataset.post;

i:=i+1;

end;

showmessage('ATTEND Personels succesfully recorded');

{
for ic:=0 to (checkboxlistbox1.Items.Count-1) do
begin
if checkboxlistbox1.state[ic]=cbchecked then

checkboxlistbox1.Items.Delete(checkboxlistbox1.ItemIndex) ;

end;}

listbox1.Clear ;

```

```

    tabsheet2.Show ;

    tabsheet1.show;

end;

if (radiobutton3.Checked ) then
begin
i:=0;

while (i <=(listbox1.count-1)) do
begin
    Datasource2.Edit;

    datasource2.DataSet.Append;

    datasource2.DataSet.FieldValues['dat']:=edit2.Text ;

    p:= pos(' ',listbox1.Items.ValueFromIndex[i]);

    s:=strtofloat(leftstr(listbox1.Items.ValueFromIndex[i],p-1)) ;

    datasource2.dataset.fieldvalues['ssn']:=s;

    datasource2.DataSet.FieldValues['abs']:= 'True';

    datasource2.dataset.post;

    i:=i+1;

```



```

end;

showmessage('ABSENT Personels succesfully recorded');

listbox1.Clear ;

tabsheet2.Show ;

tabsheet1.show;

end;

else

showmessage('please Select Personnels And Attend Kind');

end;

procedure Tperatt.BitBtn4Click(Sender: TObject);

begin

query3.Close;

query3.SQL.Clear ;

query3.SQL.Add('select a.ssn,p.name,p.surname,p.phone from db\attend a ,

db\personnel p where a.ssn=p.ssn and

a.dat="'+datetostr(datetimerpicker2.Date)+'" and a.permis is null and a.abs

='+false+' ');

```

```

query3.Open;

dbgrid1.Visible :=true;

bitbtn7.Visible :=true;

end;

query3.ExecSQL;

procedure Tperatt.BitBtn5Click(Sender: TObject);
begin
query3.Close;

query3.SQL.Clear ;

query3.SQL.Add('select a.ssn,p.name,p.surname,p.phone,a.permis from
db\attend a , db\personnel p where a.ssn=p.ssn and
a.perend>="'+datetostr(datetimepicker2.Date)+'" ');

query3.Open;

dbgrid1.Visible :=true;

bitbtn7.Visible :=true;

end;

procedure Tperatt.BitBtn6Click(Sender: TObject);
begin
query3.Close;

query3.SQL.Clear ;

```

```

query3.SQL.Add('select a.ssn,p.name,p.surname,p.phone from db\attend a ,
db\personnel p where a.ssn=p.ssn and
a.dat="'+datetostr(datetimepicker2.Date)+'" and a.permiss is null and a.abs
='+ 'True'+' ');
query3.ExecSQL ;
query3.Open;
dbgrid1.Visible :=true;
bitbtn7.Visible :=true;
end;

```

```

procedure Tperatt.PageControl1Change(Sender: TObject);
begin

dbgrid1.Visible :=false;
bitbtn7.Visible :=false;
end;

```

```

procedure Tperatt.BitBtn7Click(Sender: TObject);
begin
query3.Edit;
query3ssn.Clear ;
query3.Post;

```

end;

query4.SQL;

procedure Tperatt.DateTimePicker3CloseUp(Sender: TObject);

var p:real;

begin

label9.Caption :=floattostr(((datetimepicker3.Date) -(datetimepicker1.Date))+1);

p:=strtofloat(label9.caption);

if p < 1 then

begin

showmessage('Ending day Should not be before Starting Day');

label9.Caption:='-';

end;

end;

procedure Tperatt.Edit1Change(Sender: TObject);

var a,b:real;

begin

a:=0;

b:=0;

label12.Caption :=query1perex.AsString ;

```

query4.Close;

query4.SQL.Clear ;

query4.SQL.Add('select perstart,perend from db\attend where permis is not null
and ssn='+query1ssn.AsString +");

query4.ExecSQL ;

query4.Open ;

```

begin

```

DataSource4.DataSet.first;

while not(DataSource4.DataSet.eof) do

begin

a := (DataSource4.DataSet.FieldValues['perend'])-
(DataSource4.DataSet.FieldValues['perstart']);

b:=b+a+1;

DataSource4.DataSet.next;

end ;

label13.Caption :=floattostr(b);

end;

```

```

procedure Tperatt.DBGrid2CellClick(Column: TColumn);

begin

edit1.Text :=query1ssn.AsString +' '+query1name.AsString +'
'+query1surname.AsString ;

```



end;

procedure Tperatt.BitBtn8Click(Sender: TObject);

var i,p: integer;

s : double;

begin

if label9.Caption = '-' then

begin

showmessage('Please Enter The Starting And Ending Dates');

end

else if strtofloat(label9.Caption) > (strtofloat(label12.Caption) -

strtofloat(label13.caption)) then

begin

application.MessageBox('This Permission Days Out Of Personels Limit'+#13+

' Please Check Permission Extends'+#13+

' !!! ', 'Warning', MB\_OK +

MB\_IconExclamation);

end

else

```

begin
    Datasource2.Edit;

    datasource2.DataSet.Append;

    datasource2.DataSet.FieldValues['dat']:=edit2.Text ;

    p:= pos(' ',edit1.Text );
    s:=strtofloat(leftstr(edit1.Text ,p-1)) ;

    datasource2.dataset.fieldvalues['ssn']:=s;

    datasource2.DataSet.FieldValues['abs']:=False';

    datasource2.DataSet.FieldValues['permis']:=combobox1.Text ;

    datasource2.DataSet.FieldValues['perstart']:=datetostr(datettimepicker1.Date)

;

    datasource2.DataSet.FieldValues['perend']:=datetostr(datettimepicker3.Date)

;

    datasource2.dataset.post;

    showmessage('PERMITTED Personel succesfully recorded');

begin
query1.close;
query1.SQL.Clear;
//query1.SQL.Add('select p.id,p.name,p.surname,p.pers,a.ssn,a.perstart,a.perend from person p join person a on p.id=a.id and a.perend<'''+edit1.Text+'''');
end;

```

end;

procedure Tperatt.TabSheet3Show(Sender: TObject);

begin

query1.Close;

query1.SQL.Clear ;

query1.sql.Add('select \* from db\personnel');

query1.ExecSQL ;

query1.Open;

end;

procedure Tperatt.TabSheet1Show(Sender: TObject);

var

a : double;

b : string;

begin

query1.close;

query1.SQL.Clear ;

//query1.SQL.Add('select p.ssn,p.name,p.surname,p.perex,a.ssn,a.perend from

db\personnel p,db\attend a where a.ssn=p.ssn and a.perend<""+edit2.Text+"" ');

```
query1.SQL.Add('select * from db\personnel where ssn NOT IN (select ssn  
from db\attend where perend>="'+edit2.Text+'" or dat="'+edit2.Text+'")) ;
```

```
query1.ExecSQL ;
```

```
query1.Open ;
```

```
checklistbox1.Items.Clear ;
```

```
DataSource1.DataSet.first;
```

```
while not(DataSource1.DataSet.eof) do
```

```
begin
```

```
  a := DataSource1.DataSet.FieldValues['ssn'];
```

```
  b := floattostr(a);
```

```
  checklistbox1.Font.Color:=clblack;
```

```
  checklistbox1.Items.Add(b+''
```

```
  '+DataSource1.DataSet.FieldValues['name']+''
```

```
  '+DataSource1.DataSet.FieldValues['surname']);
```

```
  DataSource1.DataSet.next;
```

```
end ;
```

```
end;
```

```
procedure Tperatt.BitBtn9Click(Sender: TObject);
```

```
begin
```

close;

end;

procedure Tperatt.BitBtn10Click(Sender: TObject);

begin

close;

end;

procedure Tperatt.BitBtn11Click(Sender: TObject);

begin

close; exit = click;

end;

procedure Tperatt.BitBtn12Click(Sender: TObject);

begin DataSource1 := nil;

label9.Caption := '-'; end;

showmessage('Operation Cancelled');

end;

end.

DBGrid2: TDBGrid;

Query2sinvoice: TFloatField;



## CUSTOMER REPORTS FORM:

```
unit Unit14;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, ComCtrls, StdCtrls, Buttons, Grids, DBGrids, DB, DBTables;
```

```
type
```

```
Trepcust = class(TForm)
```

```
PageControl1: TPageControl;
```

```
TabSheet1: TTabSheet;
```

```
Query1: TQuery;
```

```
DataSource1: TDataSource;
```

```
DBGrid1: TDBGrid;
```

```
TabSheet2: TTabSheet;
```

```
BitBtn2: TBitBtn;
```

```
Query2: TQuery;
```

```
DataSource2: TDataSource;
```

```
DBGrid2: TDBGrid;
```

```
Query2sinvono: TFloatField;
```

```
Query2custid: TFloatField;

Query2custname: TStringField;

Query2custsurname: TStringField;

Query2selldate: TDateField;

Query2paydate: TDateField;

Query2amount: TCurrencyField;

Query2close: TStringField;

TabSheet3: TTabSheet;

Query3: TQuery;

DataSource3: TDataSource;

DBGrid3: TDBGrid;

TabSheet4: TTabSheet;

Query4: TQuery;

DBGrid4: TDBGrid;

DataSource4: TDataSource;

Label1: TLabel;

Label2: TLabel;

Query2custtel: TStringField;

procedure BitBtn1Click(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

procedure BitBtn3Click(Sender: TObject);

procedure TabSheet4Show(Sender: TObject);
```

```

procedure TabSheet1Show(Sender: TObject);

procedure TabSheet2Show(Sender: TObject);

procedure TabSheet3Show(Sender: TObject);

procedure TabSheet1ContextPopup(Sender: TObject; MousePos: TPoint;
    var Handled: Boolean);

private
    { Private declarations }

public
    { Public declarations }

end;

var
    repcust: Trepcust;

implementation

uses Unit15, Unit16;

{$R *.dfm}

procedure Trepcust.BitBtn1Click(Sender: TObject);

begin

```

```
query1.Close;
```

```
query1.SQL.Clear ;
```

```
query1.SQL.Add('select custid,custname,custsurname,custadd,custtel,custmail  
from db\cust where custname="'+a+'" ');
```

```
query1.ExecSQL ;
```

```
query1.Open;
```

```
procedure Trepcust.
```

```
begin
```

```
label1.Cu
```

```
end;
```

```
qu
```

```
procedure Trepcust.BitBtn2Click(Sender: TObject);
```

```
begin
```

```
repcustprint.QuickRep1.DataSet := repcustprint.table1;
```

```
repcustprint.QuickRep1.Preview ;
```

```
end;
```

```
procedure Trepcust.BitBtn3Click(Sender: TObject);
```

```
begin
```

```

query2.Close;

query2.SQL.Clear ;

query2.SQL.Add('select

s.sinvono,s.custid,c.custname,c.custsurname,s.selldate,s.paydate,s.amount,s.clos

e from db\sinvo s , db\cust c where s.custid=c.custid ');

query2.Open;

end;

```

```

procedure Trepcust.TabSheet4Show(Sender: TObject);

begin

label1.Caption :=datetostr(now);

query4.Close;

query4.SQL.Clear ;

query4.SQL.Add('select

c.custid,c.custname,c.custsurname,s.sinvono,s.paydate,s.amount from db\cust c ,

db\sinvo s where c.custid=s.custid and s.paydate < '"+label1.Caption+"' and

s.close='"+'NO'+"'");

query4.Open;

end;

```

```

procedure Trepcust.TabSheet1Show(Sender: TObject);

```



```
begin
```

```
query1.Close;
```

```
query1.Open;
```

```
end;
```

```
procedure Trepcust.TabSheet2Show(Sender: TObject);
```

```
begin
```

```
query2.Close;
```

```
query2.Open;
```

```
end;
```

```
procedure Trepcust.TabSheet3Show(Sender: TObject);
```

```
begin
```

```
query3.Close;
```

```
query3.Open;
```

```
end;
```

```
end.
```

## SUPPLIER REPORTS FORM:

unit Unit17;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Buttons, DB, Grids, DBGrids, DBTables, ComCtrls,  
DBCtrls, ExtCtrls;

type

Trepsupp = class(TForm)

PageControl1: TPageControl;

TabSheet1: TTabSheet;

Query1: TQuery;

DBGrid1: TDBGrid;

DataSource1: TDataSource;

BitBtn1: TBitBtn;

TabSheet2: TTabSheet;

Query2: TQuery;

DataSource2: TDataSource;

DBGrid2: TDBGrid;

```

Query3: TQuery;

DataSource3: TDataSource;

Query3brand: TStringField;

Query4: TQuery;

DataSource4: TDataSource;

BitBtn2: TBitBtn;

RadioGroup1: TRadioGroup;

RadioButton1: TRadioButton;

RadioButton2: TRadioButton;

DBLookupComboBox1: TDBLookupComboBox;

DBLookupComboBox2: TDBLookupComboBox;

procedure BitBtn1Click(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

procedure TabSheet1Show(Sender: TObject);

private

    { Private declarations }

public

    { Public declarations }

end;

var

    repsupp: Trepsupp;

```

implementation

```
uses Unit18;
```

```
{ $R *.dfm }
```

```
procedure Trepssup.BitBtn1Click(Sender: TObject);
```

```
begin
```

```
//repsuppprint.show;
```

```
repsuppprint.QuickRep1.Preview ;
```

```
end;
```

```
procedure Trepssup.BitBtn2Click(Sender: TObject);
```

```
begin
```

```
if radiobutton1.Checked = true then
```

```
begin
```

```
if dblookupcombobox1.Text = "" then
```

```
begin
```

```
showmessage('Please Select A Brand For Search');
```

```
dblookupcombobox1.SetFocus ;
```

```
end
```

```

else

begin

query2.Close;

query2.SQL.Clear ;

query2.SQL.Add('select suppname,supptel,suppfax,suppmail,supadd from

db\supplier where suppname in (select suppname from db\stock where brand

="'+dblookupcombobox1.Text +'")' );

query2.Open;

end;

end;

if radiobutton2.Checked = true then

begin

if dblookupcombobox2.Text =" then

begin

showmessage('Please Select A Name For Search');

dblookupcombobox2.SetFocus ;

end

else

begin

query2.Close;

query2.SQL.Clear ;

```



```
query2.SQL.Add('select suppname,supptel,suppfax,suppmail,supadd from  
db\supplier where suppname in (select suppname from db\stock where name  
="'+dblookupcombobox2.Text +'")' );
```

```
query2.Open;
```

```
end;
```

```
end;
```

```
if (radiobutton1.Checked =false ) and (radiobutton2.Checked =false ) then
```

```
begin
```

```
showmessage('Please Select A Criteria for search');
```

```
end;
```

```
end;
```

```
procedure Trepsupp.TabSheet1Show(Sender: TObject);
```

```
begin
```

```
query1.Close;
```

```
query1.Open;
```

```
end;
```

```
end.
```

## STOCK REPORTS FORM:

```
unit Unit19;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Grids, DBGrids, DB, DBTables, ComCtrls, StdCtrls, Buttons;
```

```
type
```

```
Trepstock = class(TForm)
```

```
PageControl1: TPageControl;
```

```
TabSheet1: TTabSheet;
```

```
Query1: TQuery;
```

```
DataSource1: TDataSource;
```

```
DBGrid1: TDBGrid;
```

```
TabSheet2: TTabSheet;
```

```
Query2: TQuery;
```

```
DataSource2: TDataSource;
```

```
Query2name: TStringField;
```

```
BitBtn1: TBitBtn;
```

```
Query3: TQuery;
```

```

DBGrid3: TDBGrid;

DataSource3: TDataSource;

Query3Brand: TStringField;

Query3Name: TStringField;

Query3Nq: TFloatField;

Query2quantity: TIntegerField;

ListBox1: TListBox;

ListBox2: TListBox;

TabSheet3: TTabSheet;

DBGrid2: TDBGrid;

Query4: TQuery;

DataSource4: TDataSource;

DBGrid4: TDBGrid;

TabSheet4: TTabSheet;

Query5: TQuery;

DataSource5: TDataSource;

DBGrid5: TDBGrid;

Query4name: TStringField;

Query4COUNTOfname: TIntegerField;

procedure BitBtn1Click(Sender: TObject);

procedure TabSheet1Show(Sender: TObject);

procedure TabSheet4Show(Sender: TObject);

```

```

procedure TabSheet2Show(Sender: TObject);

procedure TabSheet3Show(Sender: TObject);

private

    { Private declarations }

public

    { Public declarations }

end;

var

    repstock: Trepstock;

implementation

{$R *.dfm}

procedure Trepstock.BitBtn1Click(Sender: TObject);

var na:string;

b,q:real;

begin

datasource2.DataSet.First ;

while not(DataSource2.DataSet.eof) do

    begin

```

```

na:=datasource2.DataSet.FieldValues['name'];

b:=datasource2.DataSet.FieldValues['quantity'];

datasource3.DataSet.First ;

while not(DataSource3.DataSet.eof) do

begin
    if datasource3.DataSet.FieldValues['name']=na then

begin
    q:=datasource3.DataSet.FieldValues['nq'];

end;

datasource3.DataSet.Next ;

end;

if b<q then

begin

listbox1.Items.Add(' '+na+' '+floattostr(b)+'

'+floattostr(q));

end;

if (b>=q) and (b<=(q+3)) then

begin

listbox2.Items.Add(' '+na+' '+floattostr(b)+'

'+floattostr(q));

end;

```



```

        datasource2.DataSet.Next ;

    end;

end;

procedure Trepstock.TabSheet1Show(Sender: TObject);
begin
    query1.Refresh ;
    query1.Close;
    query1.Open;
end;

procedure Trepstock.TabSheet4Show(Sender: TObject);
begin
    query5.Close;
    query5.Open;

end;

procedure Trepstock.TabSheet2Show(Sender: TObject);
begin
    query2.Close;
    query2.Open;

```

```
query3.Close;
```

```
query3.Open;
```

```
end;
```

```
inter
```

```
procedure Trepstock.TabSheet3Show(Sender: TObject);
```

```
begin
```

```
query4.Close;
```

```
query4.Open;
```

```
end;
```

```
type
```

```
end.   
page2 = cl
```

```
PageControl:
```

```
TabSheet1: TTabSheet;
```

```
Query1: TQuery;
```

```
DBGrid1: TDBGrid;
```

```
DataSource1: TDataSource;
```

```
DateTimePicker1: TDateTimePicker;
```

```
DateTimePicker2: TDateTimePicker;
```

```
TabSheet2: TTabSheet;
```

```
Label1: TLabel;
```

## ACCOUNT REPORTS FORM :

```
unit Unit20;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, ComCtrls, DB, Grids, DBGrids, DBTables, Buttons, math;
```

```
type
```

```
Trepacc = class(TForm)
```

```
PageControl1: TPageControl;
```

```
TabSheet1: TTabSheet;
```

```
Query1: TQuery;
```

```
DBGrid1: TDBGrid;
```

```
DataSource1: TDataSource;
```

```
DateTimePicker1: TDateTimePicker;
```

```
DateTimePicker2: TDateTimePicker;
```

```
TabSheet2: TTabSheet;
```

```
Label1: TLabel;
```

```
Label2: TLabel;
```

```
BitBtn1: TBitBtn;
```

Label3: TLabel;  
Label4: TLabel;  
Query2: TQuery;  
DBGrid2: TDBGrid;  
DataSource2: TDataSource;  
DateTimePicker3: TDateTimePicker;  
DateTimePicker4: TDateTimePicker;  
Label5: TLabel;  
Label6: TLabel;  
BitBtn2: TBitBtn;  
Label7: TLabel;  
Label8: TLabel;  
TabSheet3: TTabSheet;  
Query3: TQuery;  
DataSource3: TDataSource;  
DBGrid3: TDBGrid;  
Label9: TLabel;  
Label10: TLabel;  
DateTimePicker5: TDateTimePicker;  
DateTimePicker6: TDateTimePicker;  
BitBtn3: TBitBtn;  
Label11: TLabel;

Label12: TLabel;  
Query3Invono: TFloatField;  
Query3Suppname: TStringField;  
Query3Buydate: TDateField;  
Query3Paydate: TDateField;  
Query3Amount: TCurrencyField;  
Query3Close: TStringField;  
TabSheet4: TTabSheet;  
Query4: TQuery;  
DataSource4: TDataSource;  
DBGrid4: TDBGrid;  
DBGrid5: TDBGrid;  
Query5: TQuery;  
DataSource5: TDataSource;  
BitBtn4: TBitBtn;  
Label13: TLabel;  
Label14: TLabel;  
Label15: TLabel;  
Label16: TLabel;  
Label17: TLabel;  
Label18: TLabel;  
DateTimePicker7: TDateTimePicker;



```

DateTimePicker8: TDateTimePicker;

procedure TabSheet1Show(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

procedure TabSheet2Show(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

procedure TabSheet3Show(Sender: TObject);

procedure BitBtn3Click(Sender: TObject);

procedure TabSheet4Show(Sender: TObject);

procedure BitBtn4Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

repacc: Trepacc;

implementation

{$R *.dfm}

```

```
procedure Trepacc.TabSheet1Show(Sender: TObject);
```

```
begin
```

```
datetimepicker1.Date :=(now);
```

```
datetimepicker2.Date :=(now);
```

```
end;
```

```
procedure Trepacc.BitBtn1Click(Sender: TObject);
```

```
var t:real;
```

```
begin
```

```
t:=0;
```

```
query1.Close;
```

```
query1.SQL.Clear;
```

```
query1.SQL.Add('select
```

```
p.sernum,p.brand,p.name,s.sellprice,s.selldate,s.sinvono from db\stock
```

```
p,db\selling s where s.sernum=p.sernum and s.selldate
```

```
>='"+datetostr(datetimepicker1.Date)+'"' and s.selldate
```

```
<='"+datetostr(datetimepicker2.Date)+'"');
```

```
query1.Open ;
```

```
query1.Open ;
```

```
datasource1.DataSet.First ;
```

```
while not(datasource1.DataSet.Eof ) do
```

begin

t:=t+ strtofloat(datasource1.DataSet.FieldValues['sellprice']);

datasource1.DataSet.Next;

label4.Caption :=floattostr(floor(t))+ ' YTL';

end;

end;

procedure Trepacc.TabSheet2Show(Sender: TObject);

begin

datetimepicker3.Date :=(now);

datetimepicker4.Date :=(now);

end;

procedure Trepacc.BitBtn2Click(Sender: TObject);

var t:real;

begin

t:=0;

query2.Close;

query2.SQL.Clear;

```
query2.SQL.Add('select sernum,brand,name,buyprice,buydate,invono from  
db\stock where buydate >="'+datetostr(datetimestr3.Date)+'" and buydate  
<="'+datetostr(datetimestr4.Date) +"'");
```

```
query2.Open ;
```

```
query2.Open ;
```

```
datasource2.DataSet.First ;
```

```
while not(datasource2.DataSet.EOF ) do
```

```
begin
```

```
t:=t+ strtoint(datasource2.DataSet.FieldValues['buyprice']);
```

```
datasource2.DataSet.Next;
```

```
label7.Caption :=floattostr(floor(t))+ ' YTL';
```

```
end;
```

```
end;
```

```
procedure Trepacc.TabSheet3Show(Sender: TObject);
```

```
begin
```

```
datetimestr5.Date :=(now);
```

```
datetimestr6.Date :=(now);
```

```
end;
```

```

procedure Trepacc.BitBtn3Click(Sender: TObject);

var t:real;

begin
t:=0;

query3.Close;

query3.SQL.Clear;

query3.SQL.Add('select * from db\pinvo  where paydate
>="'+datetostr(datetimepicker5.Date)+'" and paydate
<="'+datetostr(datetimepicker6.Date) +'" and close="'+NO+'");

query3.Open ;

query3.Open ;

datasource3.DataSet.First ;

while not(datasource3.DataSet.Eof ) do
begin
t:=t+ strtofloat(datasource3.DataSet.FieldValues['Amount']);

datasource3.DataSet.Next;

label12.Caption :=floattostr(floor(t))+ ' YTL';

end;

datasource4.DataSet.First;

end; while not(datasource4.DataSet.Eof ) do

```



```

procedure Trepacc.TabSheet4Show(Sender: TObject);
begin
    datettimepicker5.Date :=(now);
    datettimepicker6.Date :=(now);
end;

procedure Trepacc.BitBtn4Click(Sender: TObject);
var t,p:real;
begin
    t:=0;
    p:=0;

    query4.Close;

    query4.SQL.Clear;

    query4.SQL.Add('select sinvono,custid,paydate,Amount from db\sinvo
where paydate >="'+datetostr(datettimepicker7.Date)+'" and paydate
<="'+datetostr(datettimepicker8.Date) +' " and close="'+YES+' "');

    query4.Open ;

    query4.Open ;

    datasource4.DataSet.First ;

    while not(datasource4.DataSet.Eof ) do
    begin

```

## PERSONNEL REPORTS FORM :

unit Unit21;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, Grids, DBGrids, DBTables, ComCtrls, DBCtrls, StdCtrls,  
Buttons;

type

Trepperson = class(TForm)

PageControl1: TPageControl;

TabSheet1: TTabSheet;

Query1: TQuery;

DBGrid1: TDBGrid;

DataSource1: TDataSource;

TabSheet2: TTabSheet;

DBLookupComboBox1: TDBLookupComboBox;

Query2: TQuery;

*DataSource2: TDataSource;*

DBGrid2: TDBGrid;

```

Query3: TQuery;

Query4: TQuery;

DataSource3: TDataSource;

DataSource4: TDataSource;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

DateTimePicker1: TDateTimePicker;

DateTimePicker2: TDateTimePicker;

Label5: TLabel;

Label6: TLabel;

Label4: TLabel;

DBGrid3: TDBGrid;

DBGrid4: TDBGrid;

BitBtn1: TBitBtn;

procedure TabSheet1Show(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

private

    { Private declarations }

public

    { Public declarations }

end;

```

var

repperson: Trepperson;

implementation

{ \$R \*.dfm }

procedure Clear ;

procedure Trepperson.TabSheet1Show(Sender: TObject);

begin

query1.Close;

query1.Open;

end;

procedure Trepperson.BitBtn1Click(Sender: TObject);

begin

if dblookupcombobox1.Text = " then

begin

showmessage('Please Select A Personnel To Show');

dblookupcombobox1.SetFocus ;

end

else

```

begin

query2.Close;

query2.SQL.Clear ;

query2.SQL.Add('select ssn,dat from db\attend where

ssn="'+dblookupcombobox1.Text+'" and abs="'+False+'" and permis is null ');

query2.Open;

query3.Close;

query3.SQL.Clear ;

query3.SQL.Add('select ssn,perstart,perend,permis from db\attend where

ssn="'+dblookupcombobox1.Text+'" and abs="'+False+'" and permis is not

null ');

query3.Open;

query4.Close;

query4.SQL.Clear ;

query4.SQL.Add('select ssn,dat from db\attend where

ssn="'+dblookupcombobox1.Text+'" and abs="'+True+'" and permis is null ');

query4.Open;

end;

end;

end.

```



## PRODUCT DESCRIPTIONS FORM :

```
unit Unit5;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Buttons, Mask, DBCtrls, Grids, DBGrids, ExtCtrls, DB,  
DBTables;
```

```
type
```

```
Tquantity = class(TForm)
```

```
Table1: TTable;
```

```
DataSource1: TDataSource;
```

```
DBNavigator1: TDBNavigator;
```

```
DBGrid1: TDBGrid;
```

```
DBEdit1: TDBEdit;
```

```
DBEdit2: TDBEdit;
```

```
DBEdit3: TDBEdit;
```

```
BitBtn1: TBitBtn;
```

```
BitBtn2: TBitBtn;
```

```
Label1: TLabel;
```

```

Label2: TLabel;

Label3: TLabel;

BitBtn3: TBitBtn;

BitBtn4: TBitBtn;

procedure BitBtn1Click(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

procedure BitBtn3Click(Sender: TObject);

procedure BitBtn4Click(Sender: TObject);

private
{ Private declarations }

public
{ Public declarations }

end;

var
quantity: Tquantity;

implementation

{$R *.dfm}

procedure Tquantity.BitBtn1Click(Sender: TObject);

```

begin

dbnavigator1.BtnClick(nbinsert);

end;

procedure Tquantity.BitBtn2Click(Sender: TObject);

begin

dbnavigator1.BtnClick(nbedit);

end;

procedure Tquantity.BitBtn3Click(Sender: TObject);

begin

dbnavigator1.BtnClick(nbpost);

end;

procedure Tquantity.BitBtn4Click(Sender: TObject);

begin

dbnavigator1.BtnClick(nbcancel);

end;

end.

## SETTINGS FORM:

```
unit Unit13;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Mask, DBCtrls, DB, DBTables, ComCtrls, Buttons, Grids,  
DBGrids, ExtCtrls;
```

```
type
```

```
Tsetting = class(TForm)
```

```
PageControl1: TPageControl;
```

```
TabSheet1: TTabSheet;
```

```
Query1: TQuery;
```

```
DataSource1: TDataSource;
```

```
DBEdit1: TDBEdit;
```

```
DBEdit2: TDBEdit;
```

```
Label1: TLabel;
```

```
Label2: TLabel;
```

```
Edit1: TEdit;
```

```
Edit2: TEdit;
```

BitBtn1: TBitBtn;  
Query1Username: TStringField;  
Query1Password: TStringField;  
Query1Access: TStringField;  
Label3: TLabel;  
Label4: TLabel;  
BitBtn2: TBitBtn;  
BitBtn3: TBitBtn;  
TabSheet2: TTabSheet;  
Panel1: TPanel;  
DBGrid1: TDBGrid;  
Label5: TLabel;  
Label6: TLabel;  
Query2: TQuery;  
DataSource2: TDataSource;  
Query2Username: TStringField;  
Query2Password: TStringField;  
Query2Access: TStringField;  
Label7: TLabel;  
Label8: TLabel;  
Label9: TLabel;  
DBEdit3: TDBEdit;



```

DBEdit4: TDBEdit;

ComboBox1: TComboBox;

BitBtn5: TBitBtn;

BitBtn6: TBitBtn;

BitBtn7: TBitBtn;

BitBtn8: TBitBtn;

BitBtn9: TBitBtn;

BitBtn4: TBitBtn;

BitBtn10: TBitBtn;

Label10: TLabel;

procedure TabSheet1Show(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

procedure BitBtn3Click(Sender: TObject);

procedure BitBtn4Click(Sender: TObject);

procedure BitBtn6Click(Sender: TObject);

procedure BitBtn5Click(Sender: TObject);

procedure BitBtn7Click(Sender: TObject);

procedure BitBtn8Click(Sender: TObject);

procedure BitBtn9Click(Sender: TObject);

procedure BitBtn10Click(Sender: TObject);

private

```

```
{ Private declarations }
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
var
```

```
  setting: Tsetting;
```

```
  i:real;
```

```
implementation
```

```
uses Unit6;
```

```
{ $R *.dfm }
```

```
procedure Tsetting.TabSheet1Show(Sender: TObject);
```

```
begin
```

```
  i:=0;
```

```
  edit1.Enabled :=True;
```

```
  edit1.Text :=main.edit1.Text ;
```

```
  edit1.Enabled :=false;
```

```
  edit2.SetFocus ;
```

```

query1.Close;

query1.SQL.Clear ;

query1.SQL.Add('select * from db\user where username="'+edit1.Text+'" ');

query1.ExecSQL ;

query1.Open;

LABEL6.Caption := 'Dear ' + edit1.Text ;

if query1access.AsString <> 'Admin' then
begin
    setting.TabSheet2.Visible := false;
    panel1.Visible := false;
end;

end;

procedure Tsetting.BitBtn1Click(Sender: TObject);
begin
    if i < 3 then
    begin
        if edit2.text = query1password.AsString then
        begin
            label3.Visible := true;

```

```

label4.Visible :=true;

dbedit1.Visible :=true;

dbedit2.Visible :=true;

bitbtn2.Visible :=true;

bitbtn3.Visible :=true;

query1.edit ;

dbedit1.SetFocus ;

begin

end

else

begin

showmessage(' WRONG PASSWORD ');

edit2.Clear ;

edit2.SetFocus ;

i:=i+1;

end;

end

else

begin

showmessage('NO MORE TRY'+#13+

'      !!! ');

```

```

main.ChangeProfile1.Enabled :=false;

setting.Close ;

end;

end;

procedure Tsetting.BitBtn2Click(Sender: TObject);
begin
if (dbedit1.Text <> "") and (dbedit2.Text <> "") then
begin
try
query1.Post;

label3.Visible :=false;

label4.Visible :=false;

dbedit1.Visible :=false;

dbedit2.Visible :=false;

bitbtn2.Visible :=false;

bitbtn3.Visible :=false;

showmessage('Succesfully Changed Your Profile');

except

on edbengineerror do

begin

showmessage('This USERNAME Used By Another User'+#13+

'      Please Try Another USERNAME');

```



```

dbedit1.SetFocus ;

end;

end;

end

else

begin
showmessage('Please Enter The USERNAME and PASSWORD Properly');
dbedit1.SetFocus ;

end;

end;

procedure Tsetting.BitBtn3Click(Sender: TObject);
begin
query1.cancel;

label3.Visible :=false;

label4.Visible :=false;

dbedit1.Visible :=false;

dbedit2.Visible :=false;

bitbtn2.Visible :=false;

bitbtn3.Visible :=false;

showmessage('Your Current Profile Did not Changed');

end;

```

```

procedure Tsetting.BitBtn4Click(Sender: TObject);

begin

close;

end;

procedure Tsetting.BitBtn6Click(Sender: TObject);

begin

query2.Insert; Please S

dbedit1.SetFocus ;

end;

procedure Tsetting.BitBtn5Click(Sender: TObject);

begin

query2.Edit; el;

dbedit1.SetFocus ;

end;

procedure Tsetting.BitBtn7Click(Sender: TObject);

begin

query2.Delete ;

end;

procedure Tsetting.BitBtn8Click(Sender: TObject);

begin

```

```

if combobox1.Text <>' - Select Authority - ' then
begin
query2access.AsString :=combobox1.Text ;
query2.Post;
end
else
begin
showmessage('Please Set The Access Authority Of User');
combobox1.SetFocus ;
end;
end;

procedure Tsetting.BitBtn9Click(Sender: TObject);
begin
query2.Cancel ;
end;

procedure Tsetting.BitBtn10Click(Sender: TObject);
begin
close;
end;

end.

```

## PRINT FORMS:

unit Unit16;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, QuickRpt, QRCtrls, ExtCtrls, DB, DBTables;

type

Trepcustprint = class(TForm)

QuickRep1: TQuickRep;

ColumnHeaderBand1: TQRBand;

QRLabel1: TQRLabel;

QRLabel2: TQRLabel;

QRLabel3: TQRLabel;

QRLabel4: TQRLabel;

QRLabel5: TQRLabel;

QRLabel6: TQRLabel;

Table1: TTable;

DetailBand1: TQRBand;

QRDBText1: TQRDBText;

```

QRDBText2: TQRDBText;

QRDBText3: TQRDBText;

QRDBText4: TQRDBText;

QRDBText5: TQRDBText;

QRDBText6: TQRDBText;

QRLabel7: TQRLabel;

QRExpr1: TQRExpr;

QRExpr2: TQRExpr;

procedure Button1Click(Sender: TObject);

private

    { Private declarations }

public

    { Public declarations }

end;

var

    repcustprint: Trepcustprint;

implementation

uses Unit14;

```



```
{ $R *.dfm }
```

```
procedure Trepcustprint.Button1Click(Sender: TObject);
```

```
begin
```

```
repcust.show;
```

```
repcust.DataSource1.Enabled :=true;
```

```
end;
```

```
end.
```

```
unit Unit15;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, DBTables, QRCtrls, QuickRpt, ExtCtrls;
```

```
type
```

```
Tcustreport = class(TForm)
```

```
custrep1: TQuickRep;
```

```
TitleBand1: TQRBand;
```

```

DetailBand1: TQRBand;

QRSysData1: TQRSysData;

QRLabel1: TQRLabel;

QRDBText1: TQRDBText;

q: TQuery;

QRLabel2: TQRLabel;

QRDBText2: TQRDBText;

private
{ Private declarations }

public
{ Public declarations }

end;

var
    custreport: Tcustreport;

implementation

{$R *.dfm}

end.

```

unit Unit22;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, QRCtrls, QuickRpt, ExtCtrls;

type

Tinvprt = class(TForm)

QuickRep1: TQuickRep;

ColumnHeaderBand1: TQRBand;

QRLabel1: TQRLabel;

QRLabel2: TQRLabel;

QRExpr1: TQRExpr;

QRLabel4: TQRLabel;

QRLabel5: TQRLabel;

QRLabel7: TQRLabel;

QRLabel8: TQRLabel;

DetailBand1: TQRBand;

QRDBText1: TQRDBText;

QRDBText2: TQRDBText;

PageFooterBand1: TQRBand;

QRLabel9: TQRLabel;

QRDBText3: TQRDBText;

QRDBText4: TQRDBText;

QRDBText5: TQRDBText;

private

{ Private declarations }

public

{ Public declarations }

end;

var

invprt: Tinvprt;

implementation

uses Unit7;

{ \$R \*.dfm }

end.