

NEAR EAST UNIVERSITY



FACUTLY OF ECONOMIC AND  
ADMINISTRATION SCIENCE

DEPARTMENT OF COMPUTER  
INFORMATION SYSTEM

2002/2003 SPRING TERM

CLS 400

(Graduation Project )

Subject

PROCESS OF SHOPPING CENTER INFORMATION SYSTEM

Submitted To : Dr. Yalçın Akçalı  
: Miss. Nadire Çavuş

Submitted By : Mohammed Abdo (980710)

Lefkoşa  
2003



62 44 13 99 11

NEU

## TABLE OF CONTENTS



I. ACKNOELEDGMENTS.....	1
II. ABSTRACT.....	3
III. INTRODUCTION.....	5
IV. EXPLANATION OF THE SYSTEM.....	12
-Data Flow Diagram.....	13
-System Block Diagram.....	19
-Data Base Design.....	21
V. USER MENSUAL.....	25
VI. LIST OF SUBPROGRAMS.....	27
-Program Flow Chart.....	28
-Source Program.....	64
-Screen Output.....	129
VII. REFERANCES.....	150

## ACKNOWLEDGMENTS

First I would like to thank all teachers for supporting me during my study. Were this part of the project represent the hardest parts because the words can't describe my feeling while I am writing this section of the project, and the happiness of starting a new page of my life.

What ever I say how much I am grateful for them and for what they have teach me and supporting me during the learning process.

So I am

very

## I- ACKNOLEDGMENTS

## ACKNOWLEDGMENTS

First I would like to thanks all teachers for supporting me during my study . Were this part of the project represent the hardest parts because the words cant describe my feeling while I m writing this sections of the project , and the happiness of starting a new stage of my life .

What ever I say how much I m grateful for them and for what they have teach me and supporting me during the passing years .

So I say to them with the button on my heart thank you all for everything and my greeting is for all the teachers and special thanks for the teachers of computer information system department.

And I don't want to forget here my friend who help me during this project were he didn't denied any help I have asked it from him .

Chapter 1

The first thing we should understand is what  
the problem is. We are looking at the process of  
the body and the way it works. The process is the  
way the body works. The body works in a way that  
we can understand and we can sell it to the customer.

The basic things in this kind of product is the process which they  
represent the body for any thing in the world and they  
are in the following way.

# WE ABSTRACT



## ABSTRACT

This project describing the process of selling and purchasing which is happening in the shopping centers and how they business run and from which sections contains and which of these process is the essential from the shopping centers , in other words how we buy the stocks from the suppliers and how we sell it to the customers .

The basic things in this kind of project is four process which they represent the body for any shopping center a round the world and they are in the following lines.

First process which is the stock , which we buy and sell , the second process is the suppliers , where we buy it from him , third process is the customers who buy this providing stock , and the last one is the account to know how our business is going .

## III- INTRODUCTION

## INTRODUCTION

This project like I have said in the abstract is about and shopping center process and how these processes working with each others like one frame .

If we have looked to the main menu we will see that this project include four main process which they represent the whole process

For the shopping centers, now let's talk about them with some details .

First process in the main menu which is the stock , the stock have also a sub menu where it explain the stock process and what we need to know about it, the stock menu is contains the following operations

1-add stock

2-search stock

3-delete stock

4-update stock

5-list stock

now by getting more deeply in these operations to know the purpose from include it in the stock menu and what is the reasons beyond it

for add stock , we know that we have more than a one stock and the market change every day and new stocks is been produced

for this reason in the adding stock you can store the stock name, quantity ,quality, stock-code , to keep everything is in control and to know everything you want about the stocks files .



for searching the stock there is two ways for the searching on the wanted stock first methods is search by the name of the stock , and the second method is searching by the number of the wanted stock

for deleting the stocks the purpose here as we know there is some stocks become old or outdated it for that they become worthless so we delete it from our records .

some times we make a mistakes during recording the stocks for that we use the update so we can insert the correct data or information about the wanted it stock .

the listing is to see the stocks records in better way and give a along list of the stocks that you have it in your records.

The second main process is the customer process , the customer were represent the most important value for any business and the main reason beyond establishing this business

As we saw in the stock menu that we have five sub menus were they are also in the customer menu .

1-add customer

2-delete customer

3-search customer

4-update customer

5-list customer

as we know that we can gain a new customers everyday for this reason we need to record them and there name ,surname, and customer number,

as the way in adding and gaining customer we can also lose them or they could like to buy from other shopping center in that time we wont need to keep there records so we delete it ,

for searching here we can search on the customer by inserting his name or if we want we can search on him by inserting his number and then we have the wanted information a bout him .

the updating is occurs when we record the customer information in wrongly so we need to accurate them .

in the way of listing , here we can list the customer information and that will include about the available information a bout them like name surname, customer number ,and customer phone no.

The third main process is the supplier process ,and they are the sellers of the stocks where we buy the stocks from them and they are playing an important role for the business because they know a lot of information about the stocks and how to provide it and in which price for this we need to make a better control about them and their data. Also in the supplier we include it a five main process which they are.

1-add supplier

2-search supplier

3-delete supplier

4-list supplier

5-update supplier

adding the supplier is important why , because when the stock become missing or the demand on this kind of this stock is high , we can ask from the different suppliers to provide it for us also to provide the shopping centers with a different type of product with a different quality .

*in case that we need a supplier for requesting a stock we need a* information about him then we search on the wanted it supplier by entering his name or his number.

Some suppliers they went to bankrupt or insolvency , in this time they become worthless for our business so we delete them.



Last thing is updating the supplier , in case he change some information that we have it and after he inform us a bout this change so we record them.

The last main process in this project is talking a bout the accounting

And how this process is happing in the business ,and how they know where they are standing ,the accounting is very important for any business because he show the road and tell you if you are gain or lose in sales .

Now list know from what the account contains , its contains from .

1-debit account

2-credit account

3-balance account

the debit is what we own for the others and what we have to pay it for them

for example in this project we buy the stock from the suppliers so it's a purchasing price and its debit on us we have to pay it back to them for this reason we have a list of the stock -code which is the primary key also in the credit account and we have beside the stock-code the purchasing price .

the other process is the credit , which is represent the selling price of the stock for the customers here also like the debit account we have a list of the stock-code and the selling price .

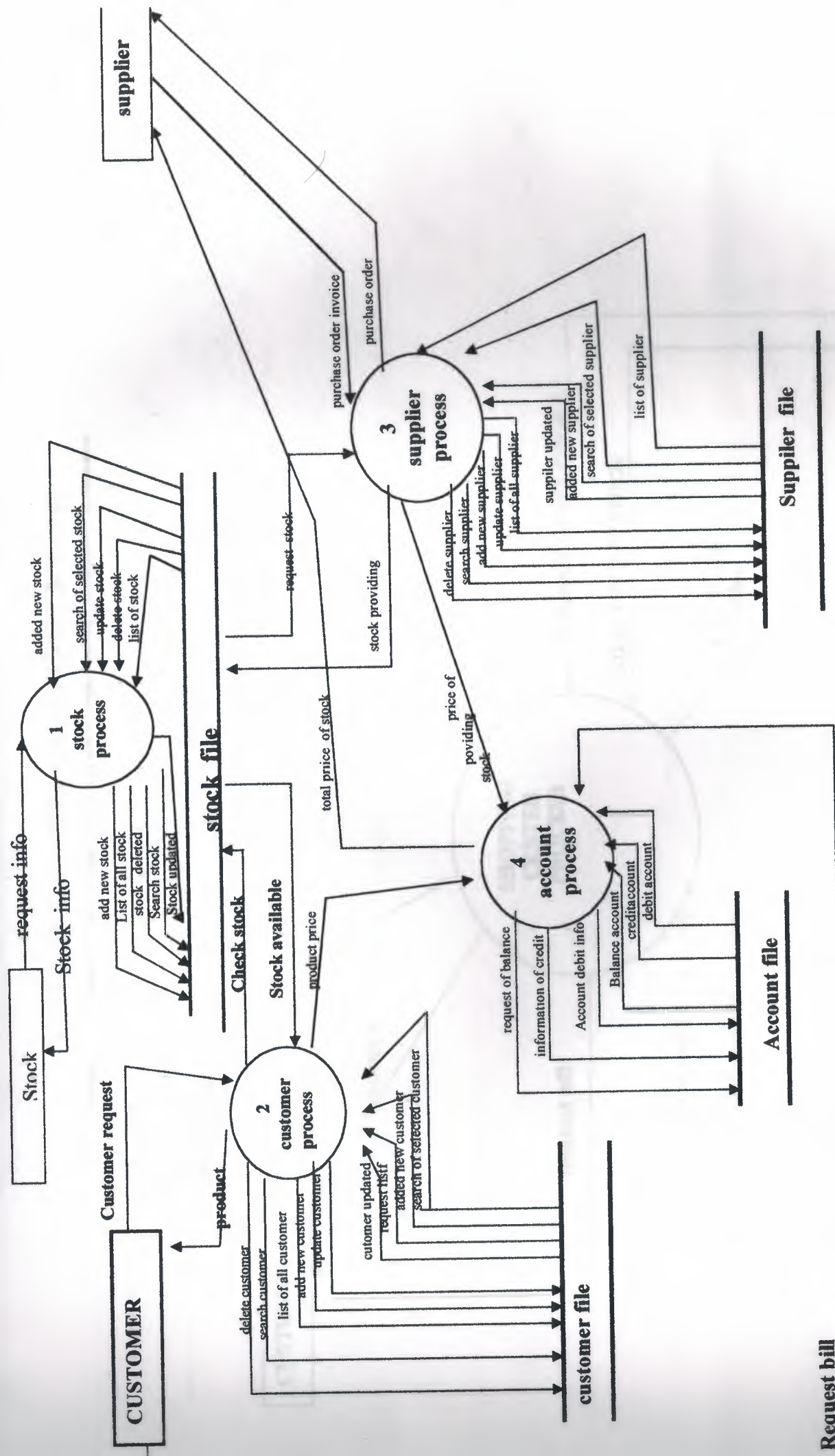
last process of the account is the balance account the reason is to know if we have gain on sale , or loss on sale

by entering the stock -- code the balance account display on the screen the purchasing price and the selling price and give you the net price where it s gain or loss in sale .



## **IV- EXPLANATION OF THE SYSTEM**

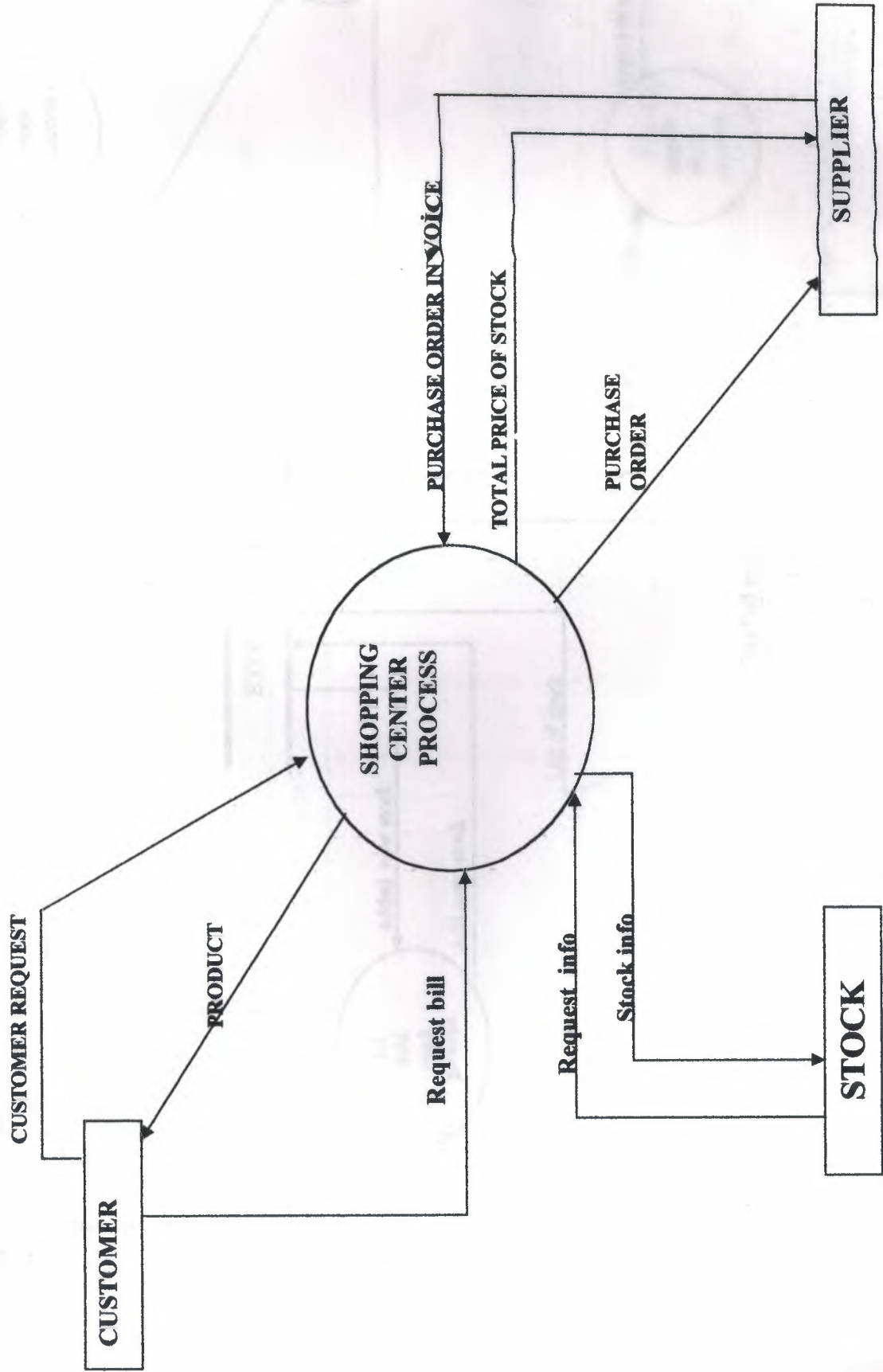
# **DATA FLOW DIAGRAM**



TOP LEVEL DED FOR SHOPPING CENTER

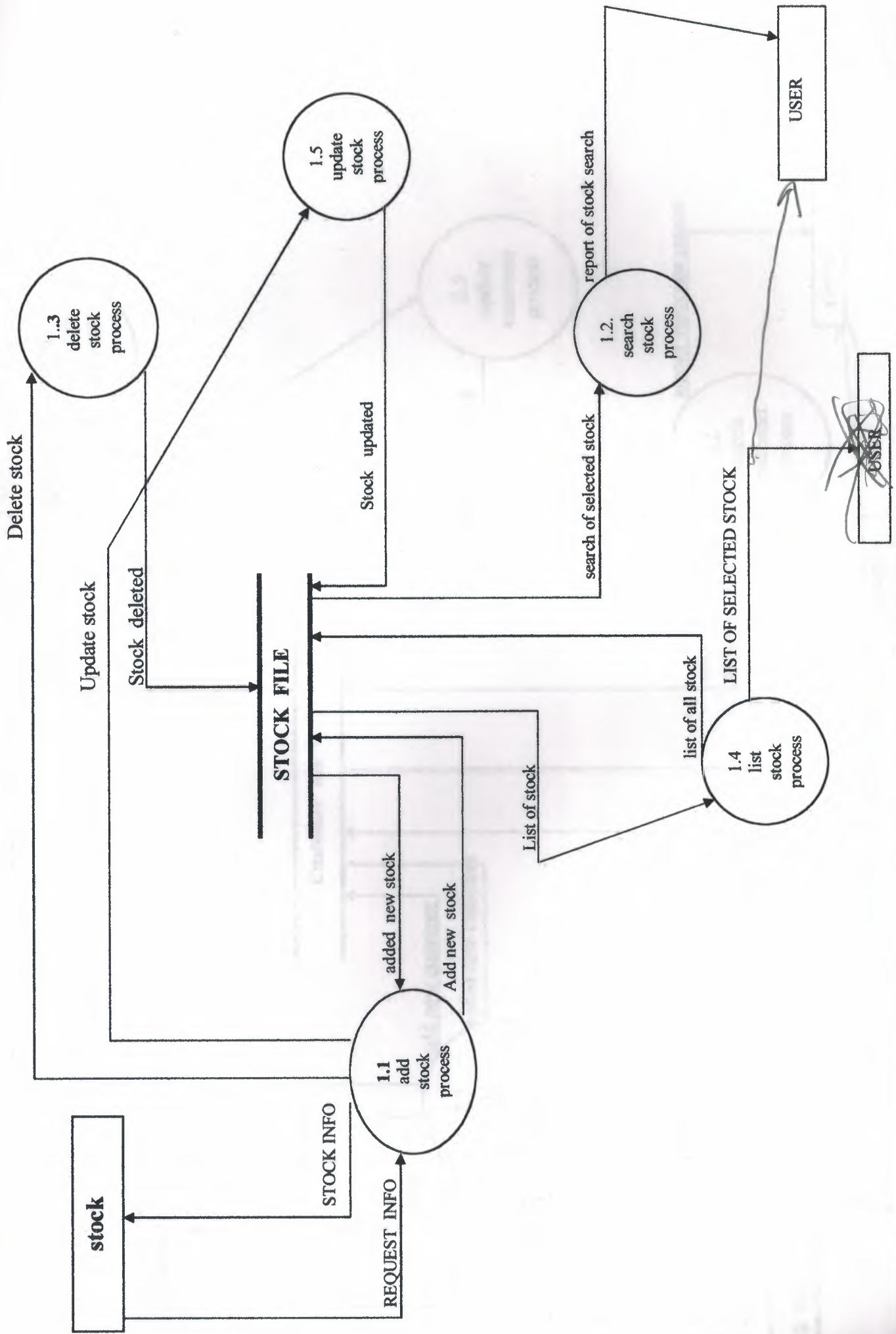
Request bill

## CONTEXT DFD FOR SHOPPING CENTER



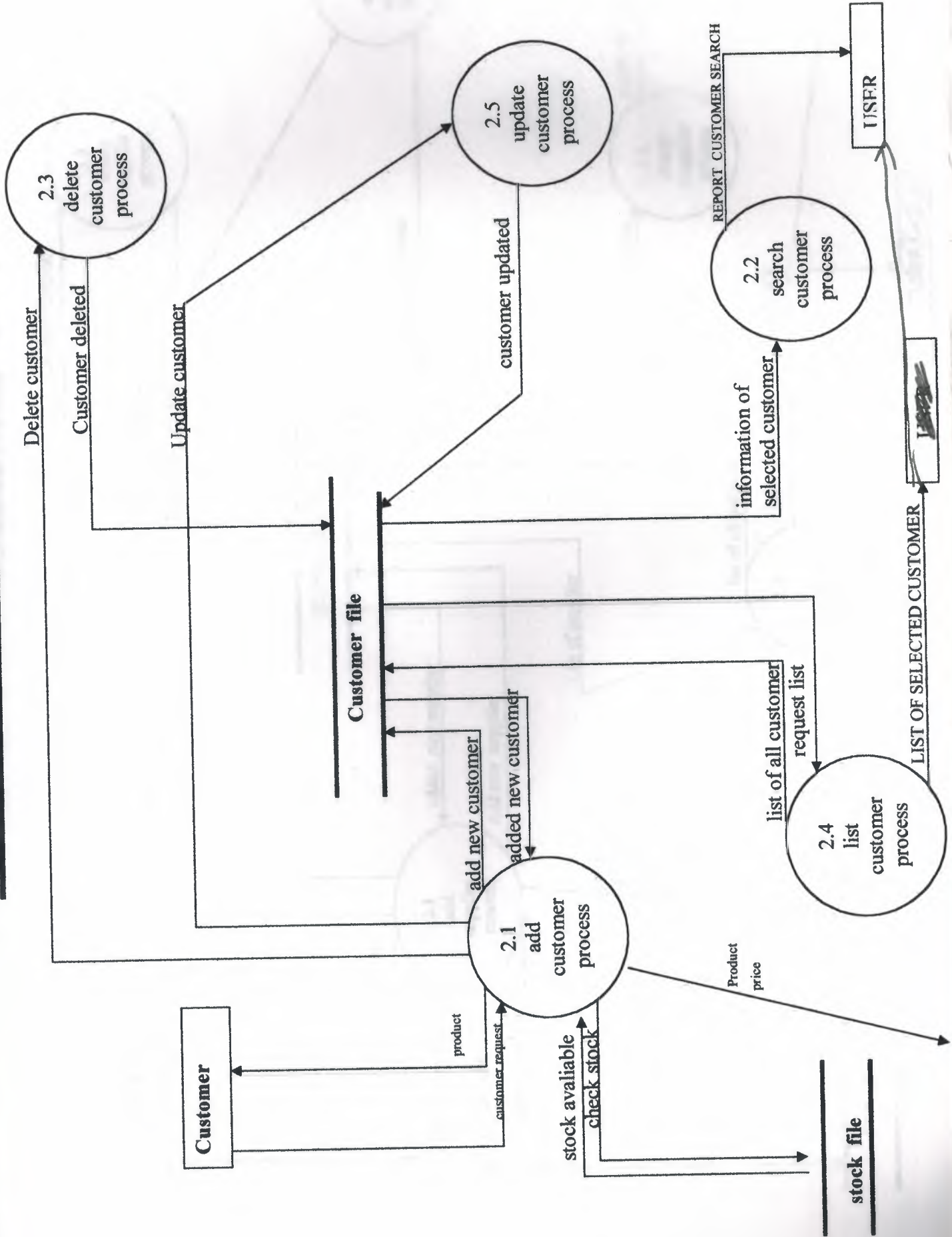


## DETAILED DFD FOR STOCK PROCESS(1)

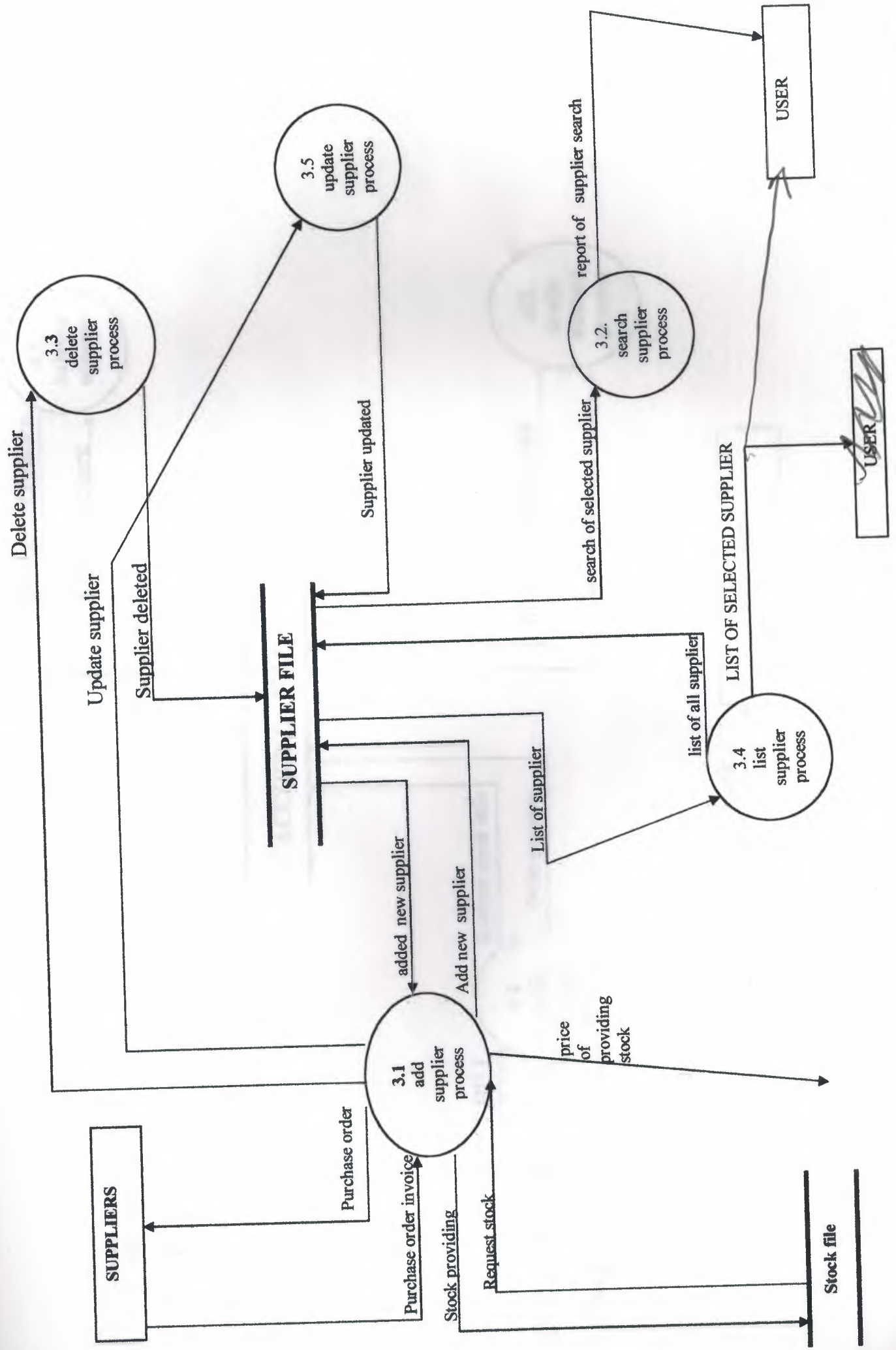




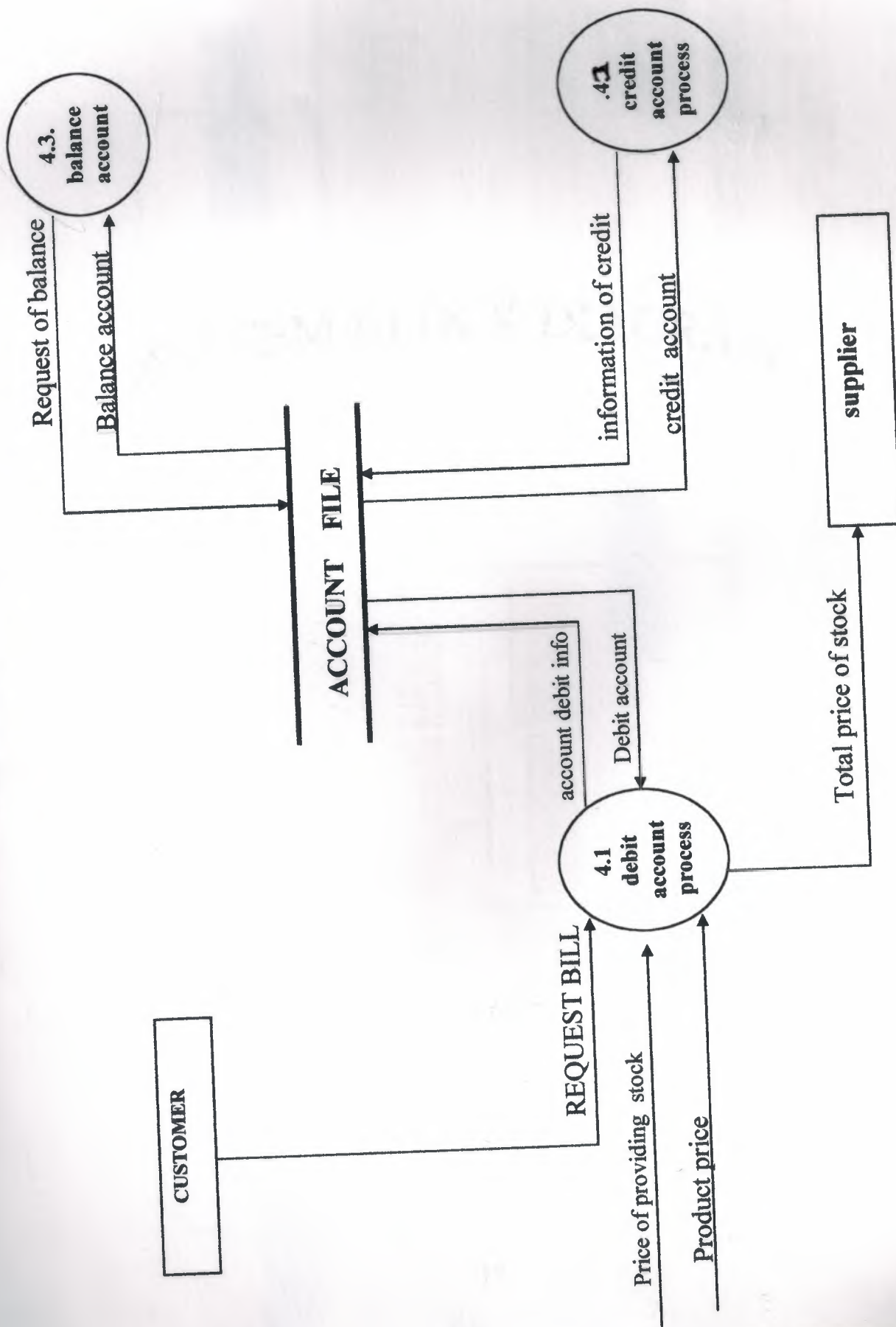
## DETAILED DFD FOR CUSTOMER PROCESS (2)



# DETAILED DFD FOR SUPPLIER PROCESS(3)



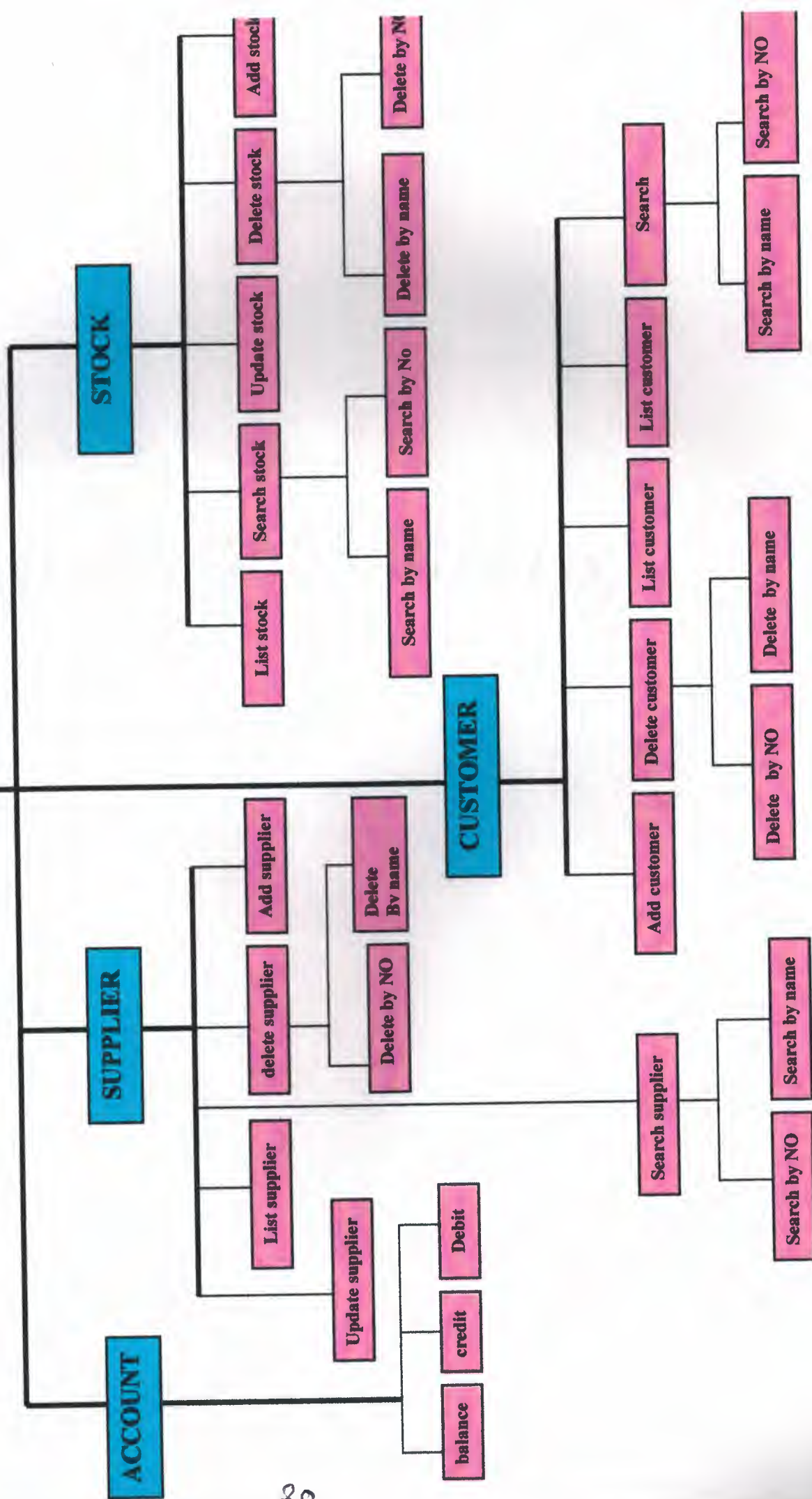
# DETAILED DED FOR ACCOUNT PROCESS



# SYSTEM BLOCK DIAGRAM



# SHOPPING CENTER





## DATA BASE DESIGN

### 1-STOCK DBF.

FILEDS	TYPE	DESCRIPTION
Stock name	Character	30
Quantity	Numeric	25
Quality	Character	30
Stock code	Numeric	15

## DATA BASE DESIGN

FILEDS	TYPE	DESCRIPTION
Stock name	Character	30
Quantity	Numeric	25
Quality	Character	30
Stock code	Numeric	15
Date		
Numeric		

## **DATA BASE DESGIN**

---

### **1- STOCK DBF.**

<b>FILEDS</b>	<b>TYPE</b>	<b>DESCRIPTION</b>
<b>Stock name</b>	<b>Character</b>	<b>30</b>
<b>Quantity</b>	<b>Numeric</b>	<b>25</b>
<b>Quality</b>	<b>Character</b>	<b>30</b>
<b>Stock code</b>	<b>Numeric</b>	<b>15</b>

### **2-CUSTOMER DBF.**

<b>FILEDS</b>	<b>TYPE</b>	<b>DESCRIPTION</b>
<b>Customer name</b>	<b>Character</b>	<b>30</b>
<b>Customer surname</b>	<b>Character</b>	<b>25</b>
<b>Customer address</b>	<b>Character</b>	<b>30</b>
<b>Customer NO</b>	<b>Numeric</b>	<b>15</b>
<b>Date</b>	<b>Date</b>	<b>-</b>
<b>Customer Tel NO</b>	<b>Numeric</b>	<b>25</b>

## **DATA BASE DESGIN**

---

### **3- SUPPLIER DBF.**

<b>FILEDS</b>	<b>TYPE</b>	<b>DESCRIPTION</b>
Supplier name	Character	30
Supplier surname	Character	25
Supplier address	Character	30
Supplier nationality	Numeric	15
Supplier company	Numeric	15
Supplier Tel NO	Numeric	25

### **4-DEBIT DBF.**

<b>FILEDS</b>	<b>TYPE</b>	<b>DESCRIPTION</b>
STOCK CODE	Numeric	30
STOCK PRICE	Numeric	25

## DATA BASE DESGIN

### 5- CREDIT DBF.

FILEDS	TYPE	DESCRIPTION
STOCK CODE	NUMERIC	30
STOCK PRICE	NUMARIC	25

### 6- BALANCE DBF.

FILEDS	TYPE	DESCRIPTION
CODE	Numeric	30
PURCHASING PRICE	Numeric	25
SELLING PRICE	Numeric	30
NET TOTAL	Numeric	15





# V.USER MANUAL



## USER MANUAL

Place the cd in the cd-rom, open project folder, select the main menu .exe application click on it.

It will show to you the main menu process screen, see (figure .1), this screen is the main screen of the process shopping information system, as its clear from the screen its include four main process which they represent the body of the whole process.

choose the process which you want to operate by a double click on the mouse, for example if you clicked on the stock process, you have the stock menu . see (figure 2) which is contain four processes in this process you can add a new stock and store it in the stock adding file you enter the name of the stock and the quantity ,and the quality and ,the essential point is the data validation . which mean that you enter the correct data type from the keyboard a simple example that the quantity of the stock is required only a numeric data type .

In this file also you can use the buttons which they will help you to make the work done easier ,there is ten button in the file and each one of them have a specific mission

top means it's the top of the record or it's the first record available, prev means the previous record , for next is for the following record ,end is the end of the file locate is show you the location of the record ,add for adding and register the new stocks ,edit which can be represent as updating for the records and if you want to make any changing in there information ,delete to remove the want it record , print to give the advice for the printer and print the record you want , and close to closing the current file and back to stock menu.

The second process is a bout searching on the stock (figure4) were you get the choice to determining the way of search by using the stock name or by using the stock code .

Here by entering the stock name the query of the stock will a pear on the screen and show you the stock that you been looking for see (figure 5) after that you close it by clicking on the (ESC) button .

Now for deleting menu which is the (figure 6) here also you can choice deleting the stock by two methods one of them by entering the stock name and the other methods by entering the stock code , and the process is done like the following , if you have entered the stock code and its available then it will be removed and show you a mas-age that its been removed , if there is no existence for this record its show you a massage tell you the record is not found press and key to continue .

For the last process in the stock menu is the listing process, by clicking on the stock list you will have the list of the available records that you have .

The second main process is the customer process .in fact if you follow the same procedure that we adapted in the explaining the stock process you wont have any trouble during running the program. The customer menu is shown in (figure.7) where there is the same process that I been explained the stock menu process .

The third main process is the supplier process which is the (figure.12) In this process its also the same like the rest of the other processes and you can use it in the same way.

The last process its the most important process for many reason , and the vital things here the using of this process the account process is show in (figure17) this process is contain from three sub process which they are the debit process (figure18) and, the credit process (figure 18),and the last one is the balance process figure 19.

The using of the account process required the attention ,why ,the reason is to have correct calculation .

If you have clicked on the debit file its will show you a file were the user can enter the stock code and the stock price , here the stock code is the primary key which mean it's the relation ship between the debit and credit and the balance .

If you clicked on the credit file you will enter the same stock code ,if u asking why the answer will become by talking how the account calculate and works.

The last process in the account is the balance account , here you just have to enter the stock code then the balance account will show to you in the account file the price of purchasing I mean the price that we purchased the stock from the supplier ,and show to you the price of the selling price , or it's the price of the stock selling to the customer, and also show to you the balance .

The balance mean if there is gain or loss in the sale

Last notice that the data type for the account should be only numeric or the program will prevent you from entering any other data type.

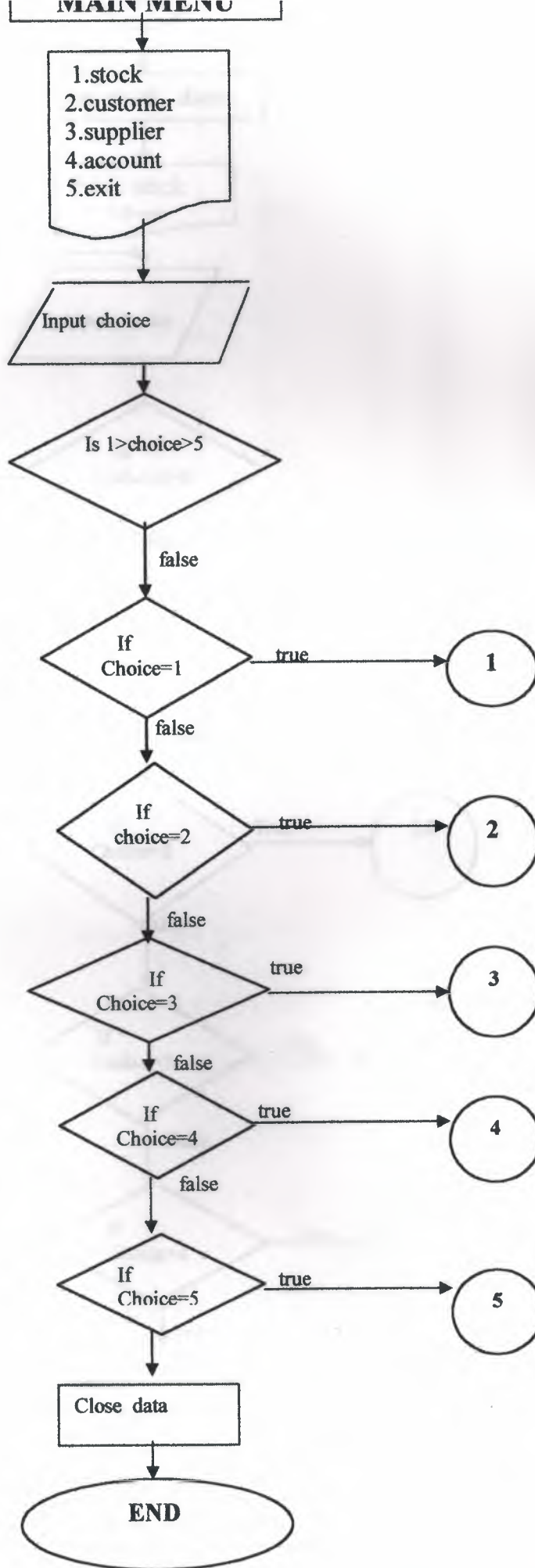
This program don't required a specific computer to been run in it, its work on any computer type with any speed available .

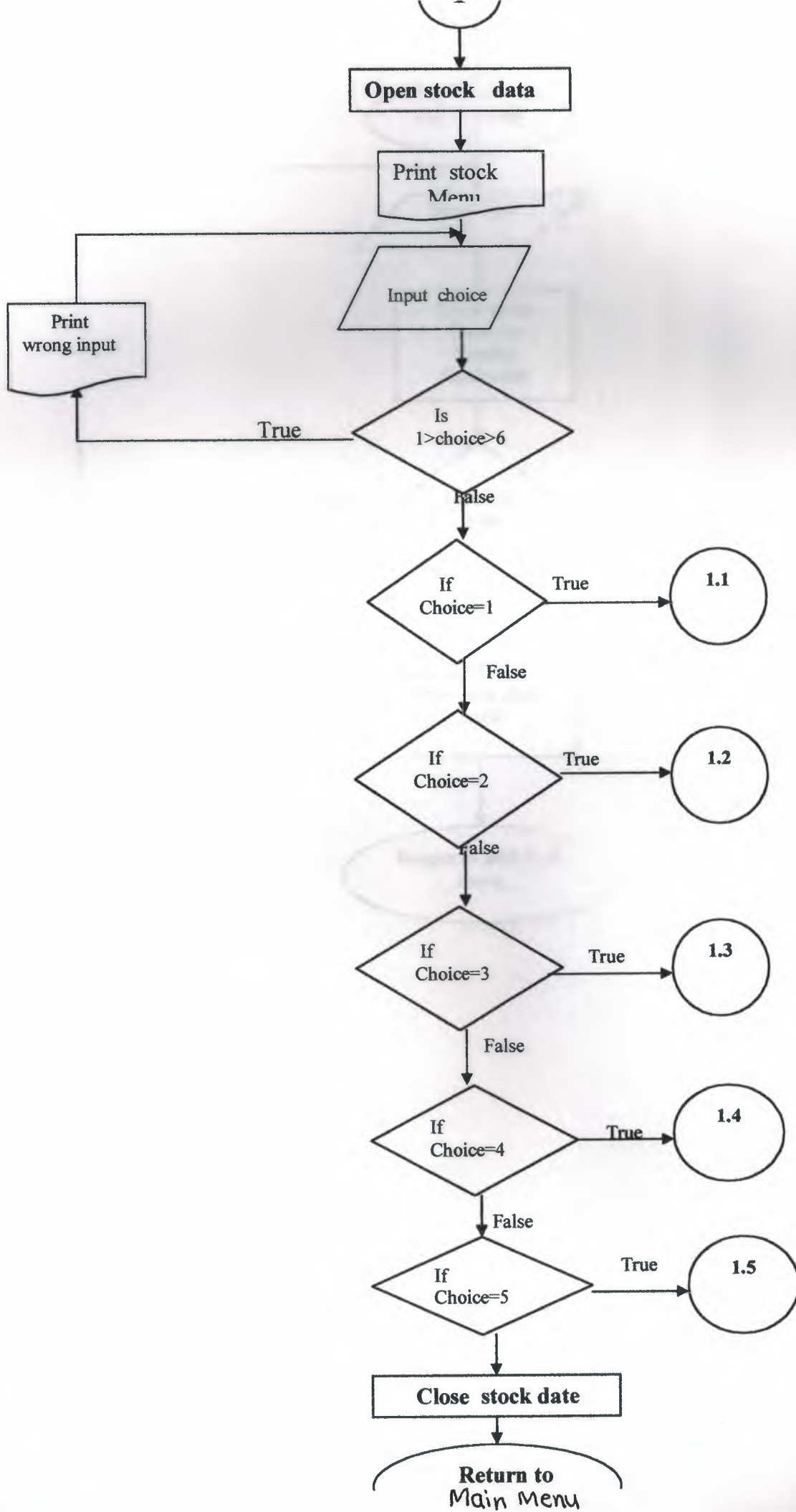
## **VI. LIST OF SUBPROGRAMES**

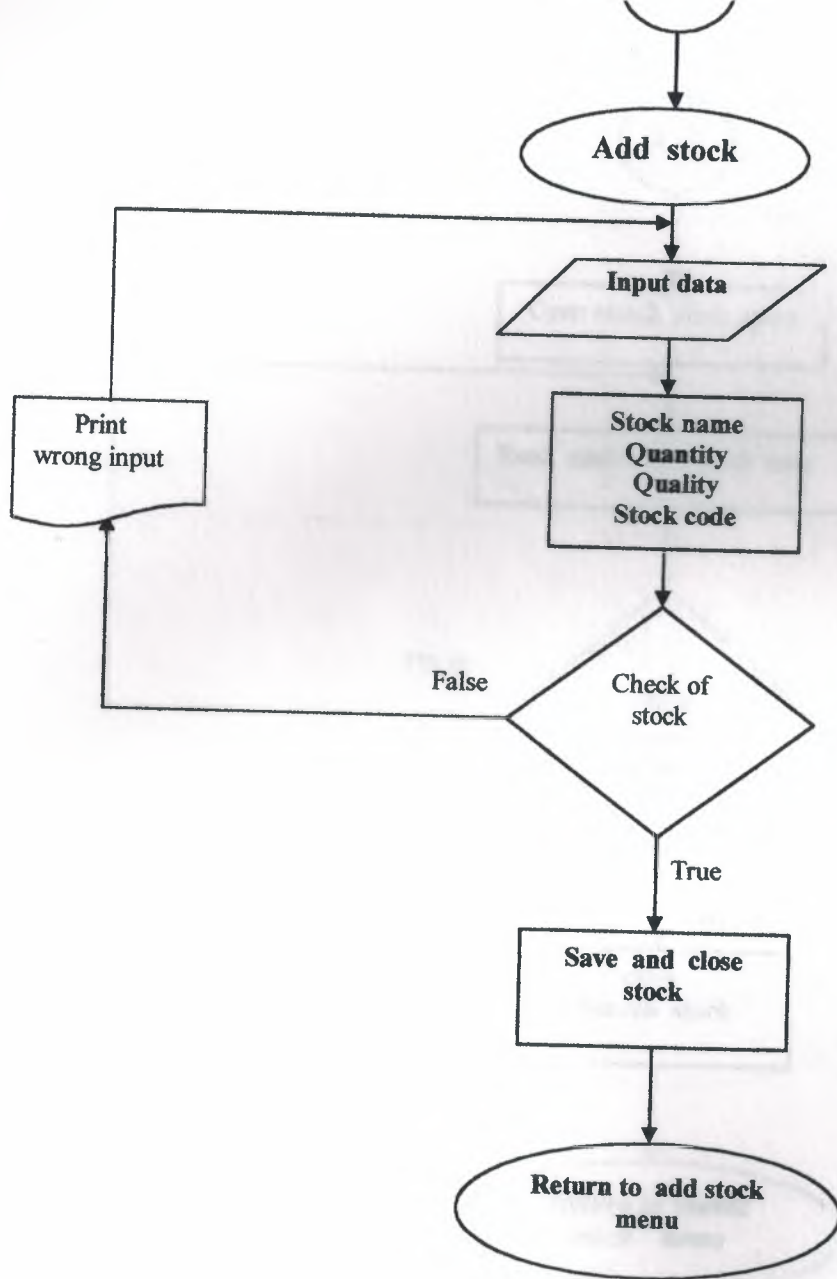


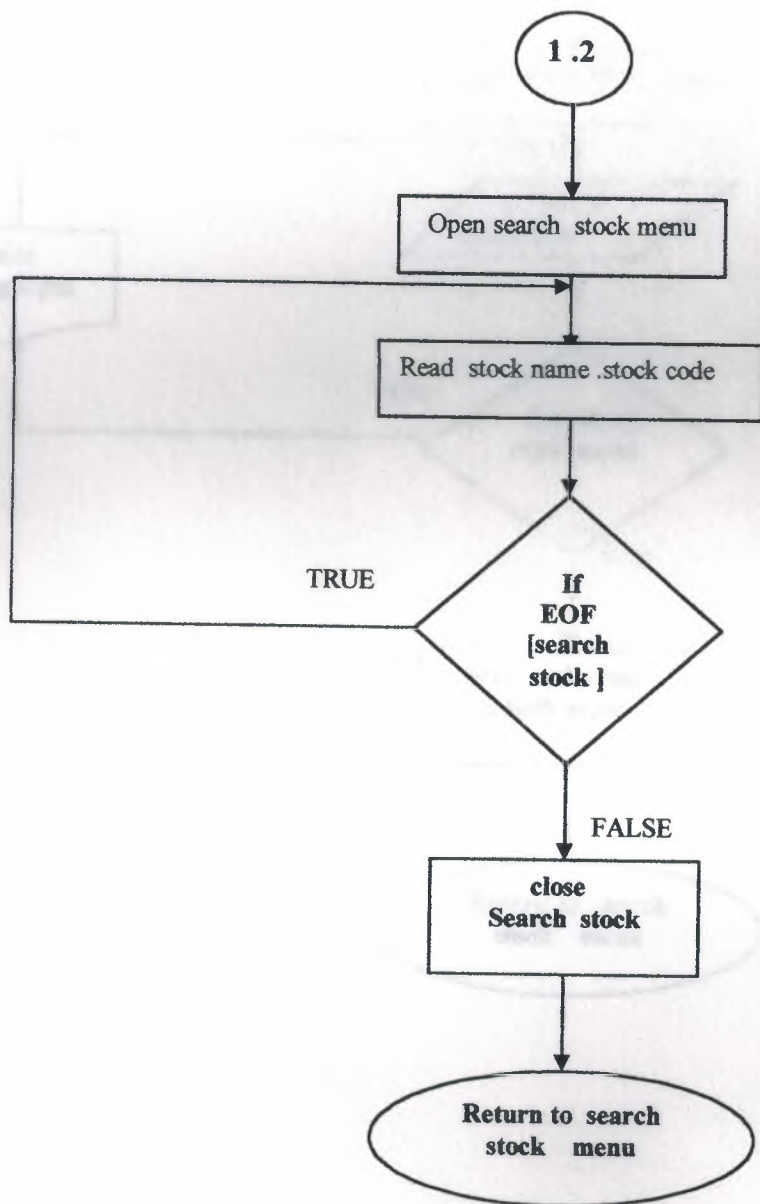
# FLOW CHARTS



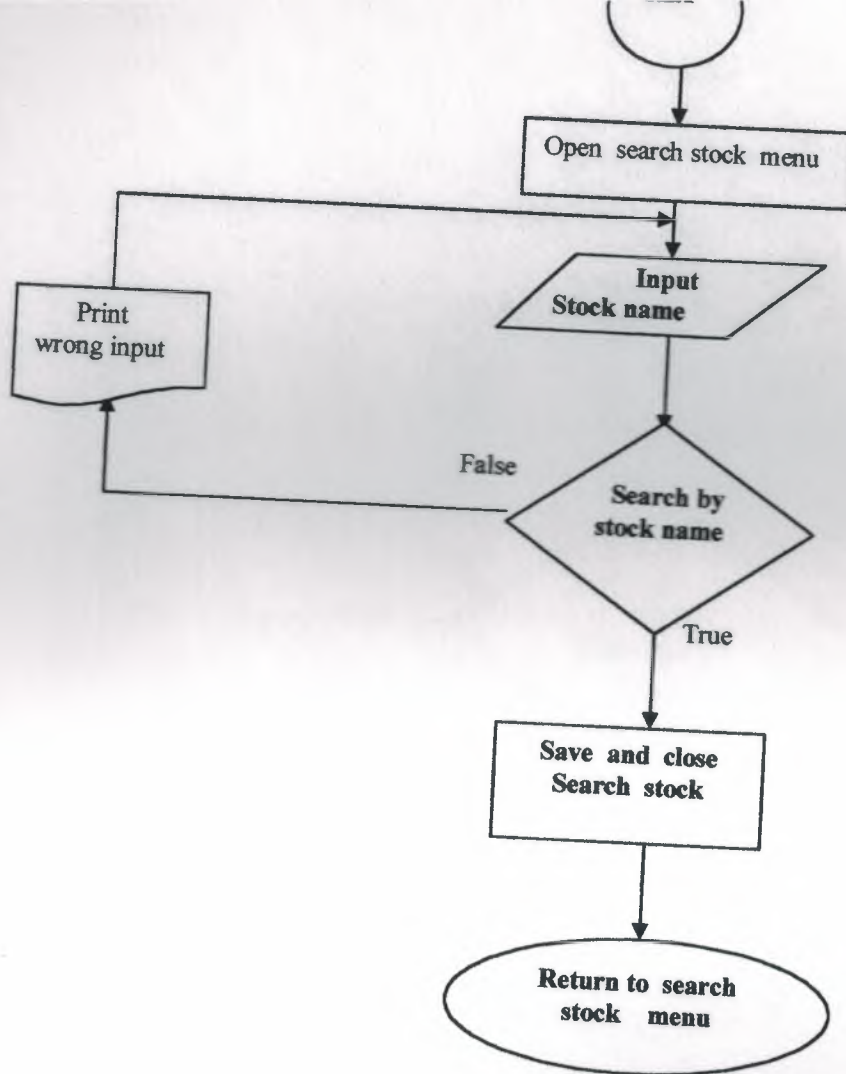


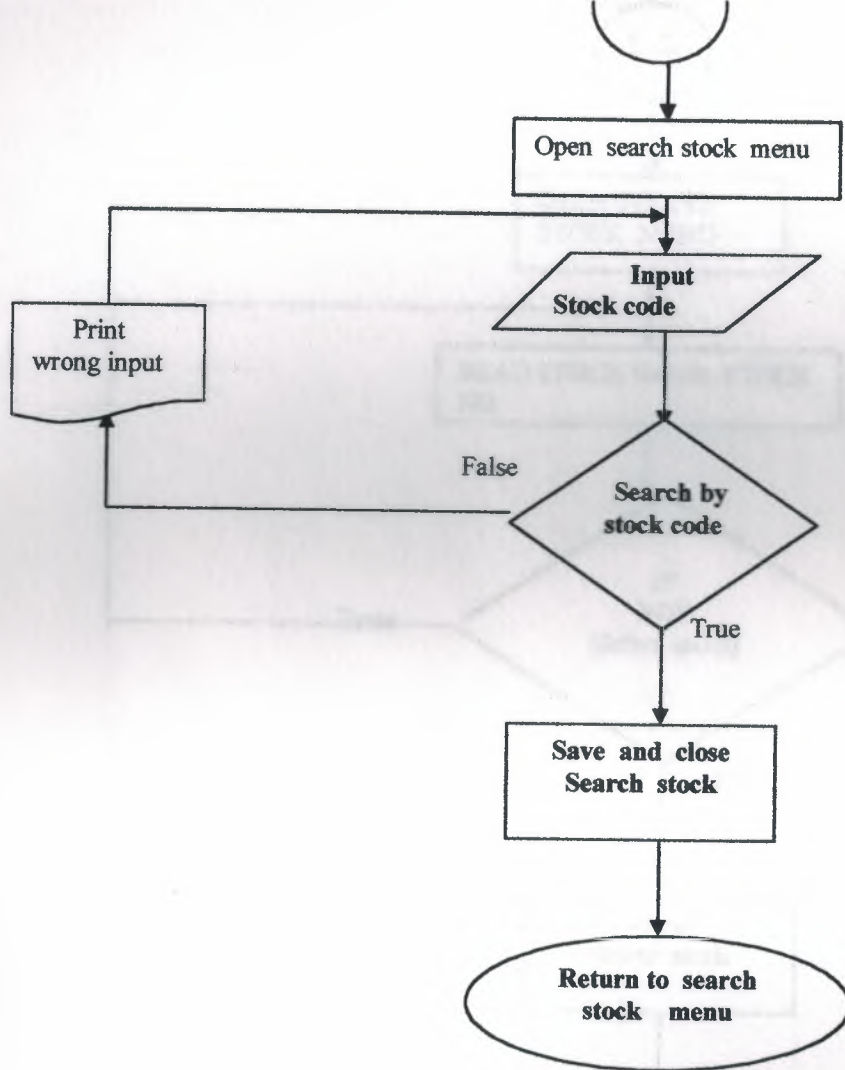












1.3

OPEN DELETE  
STOCK MENU

READ STOCK NAME, STOCK  
NO

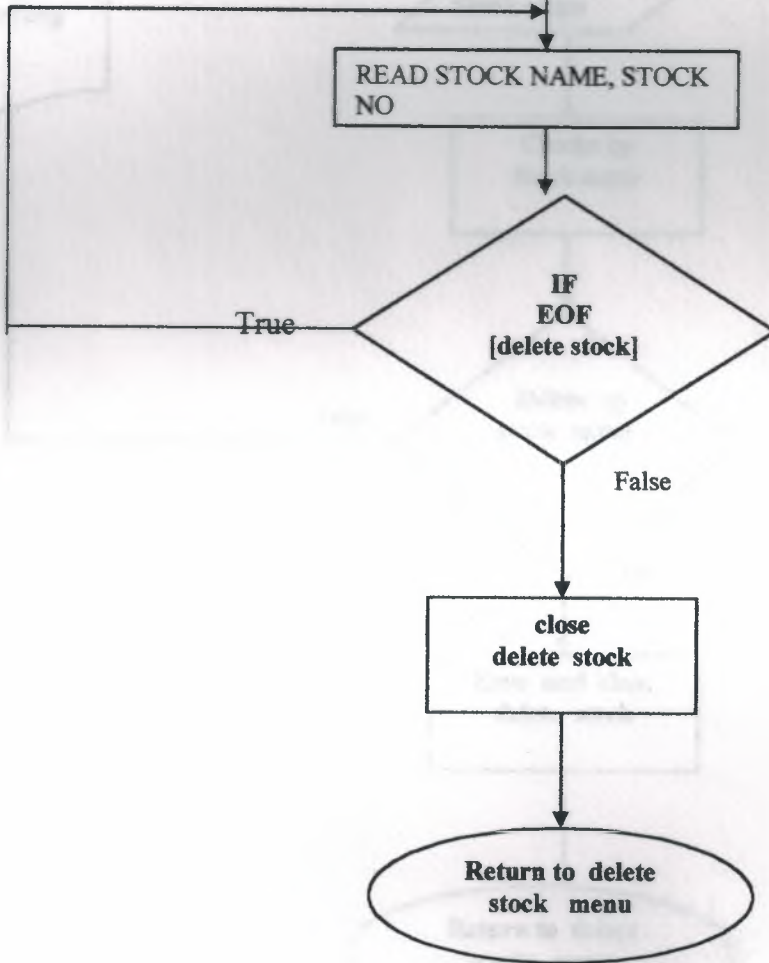
IF  
EOF  
[delete stock]

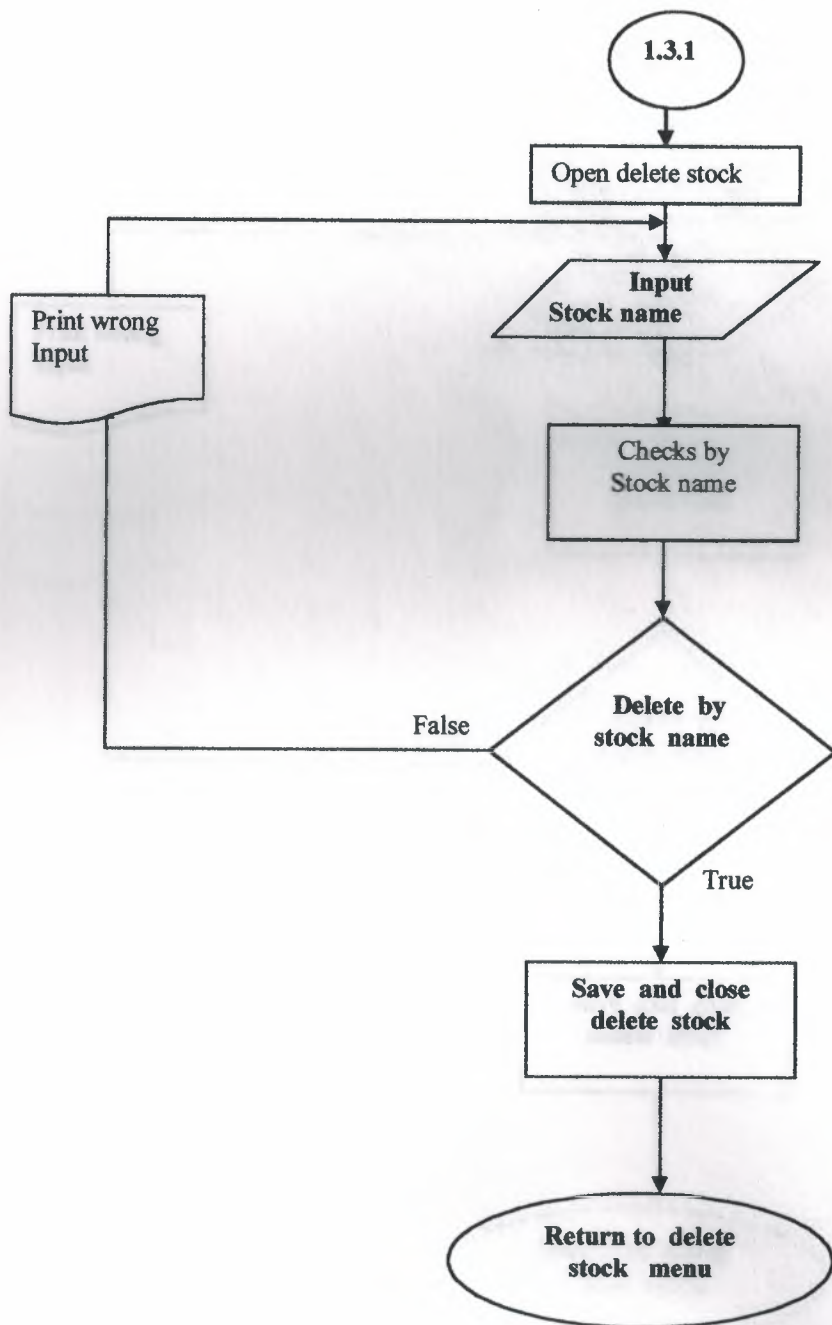
True

False

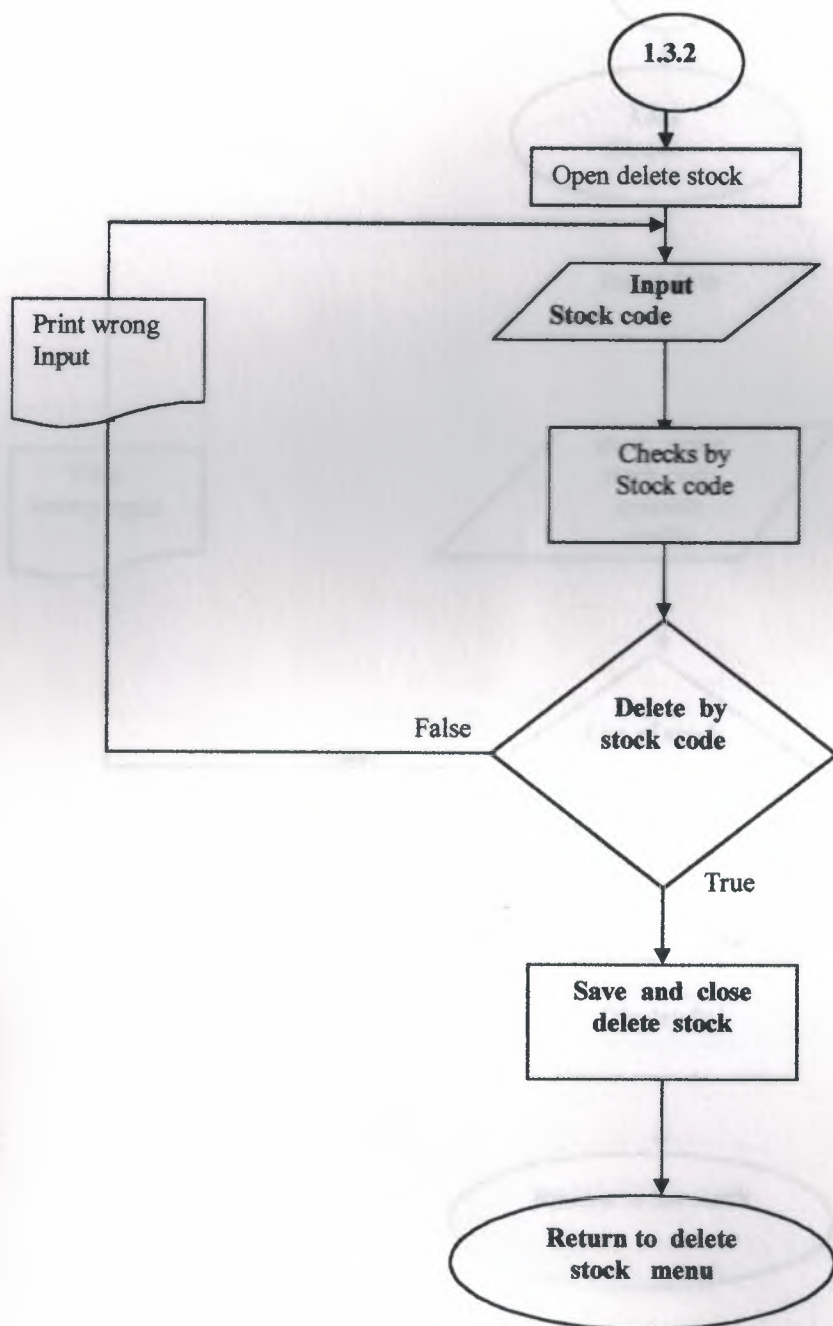
close  
delete stock

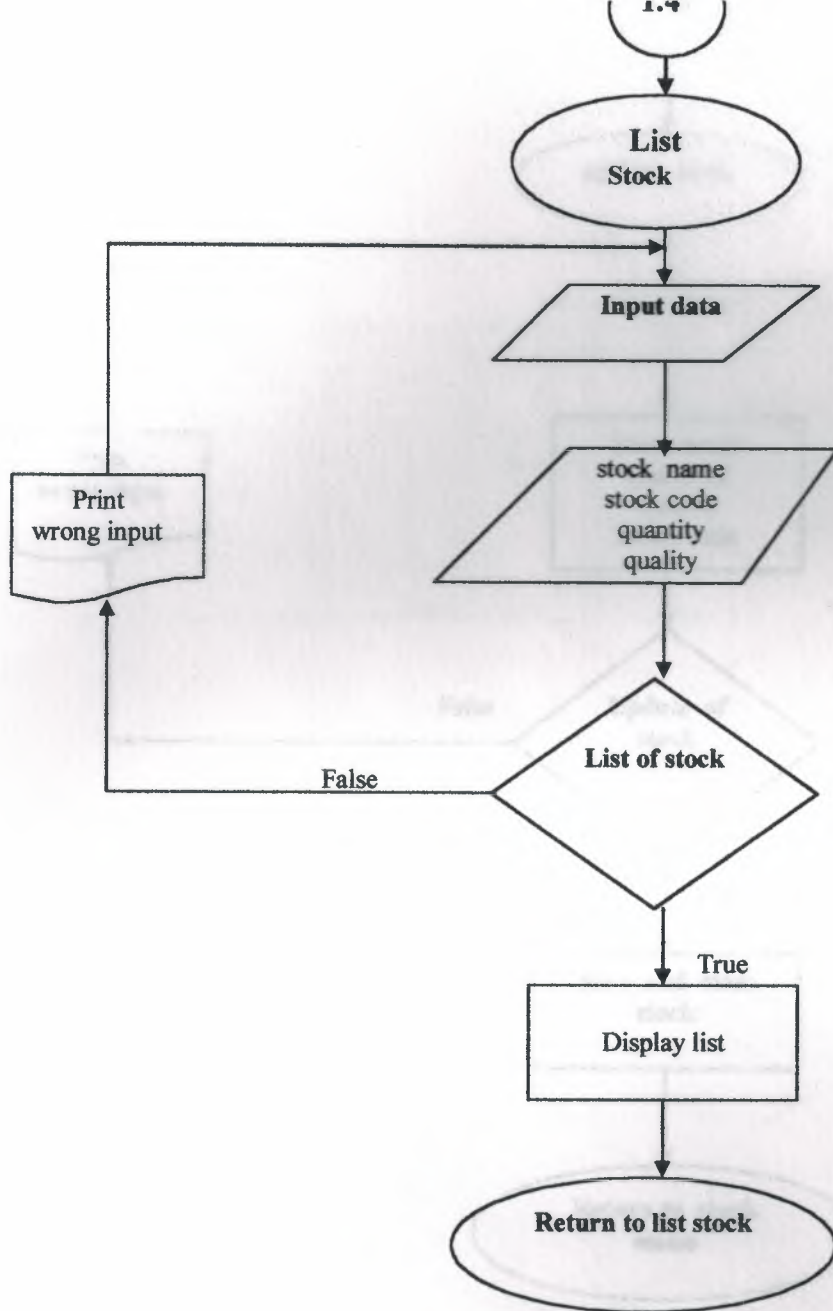
Return to delete  
stock menu

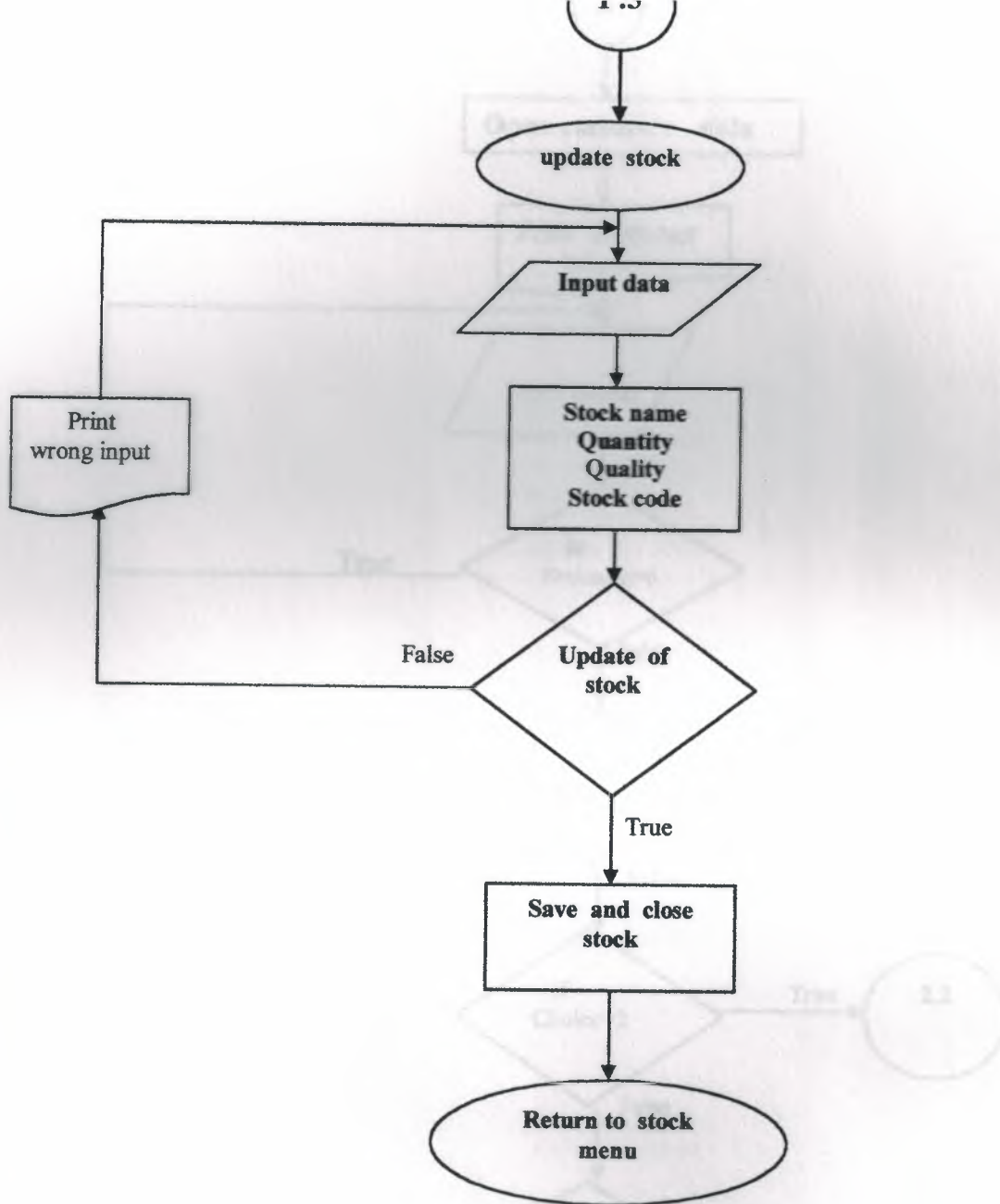


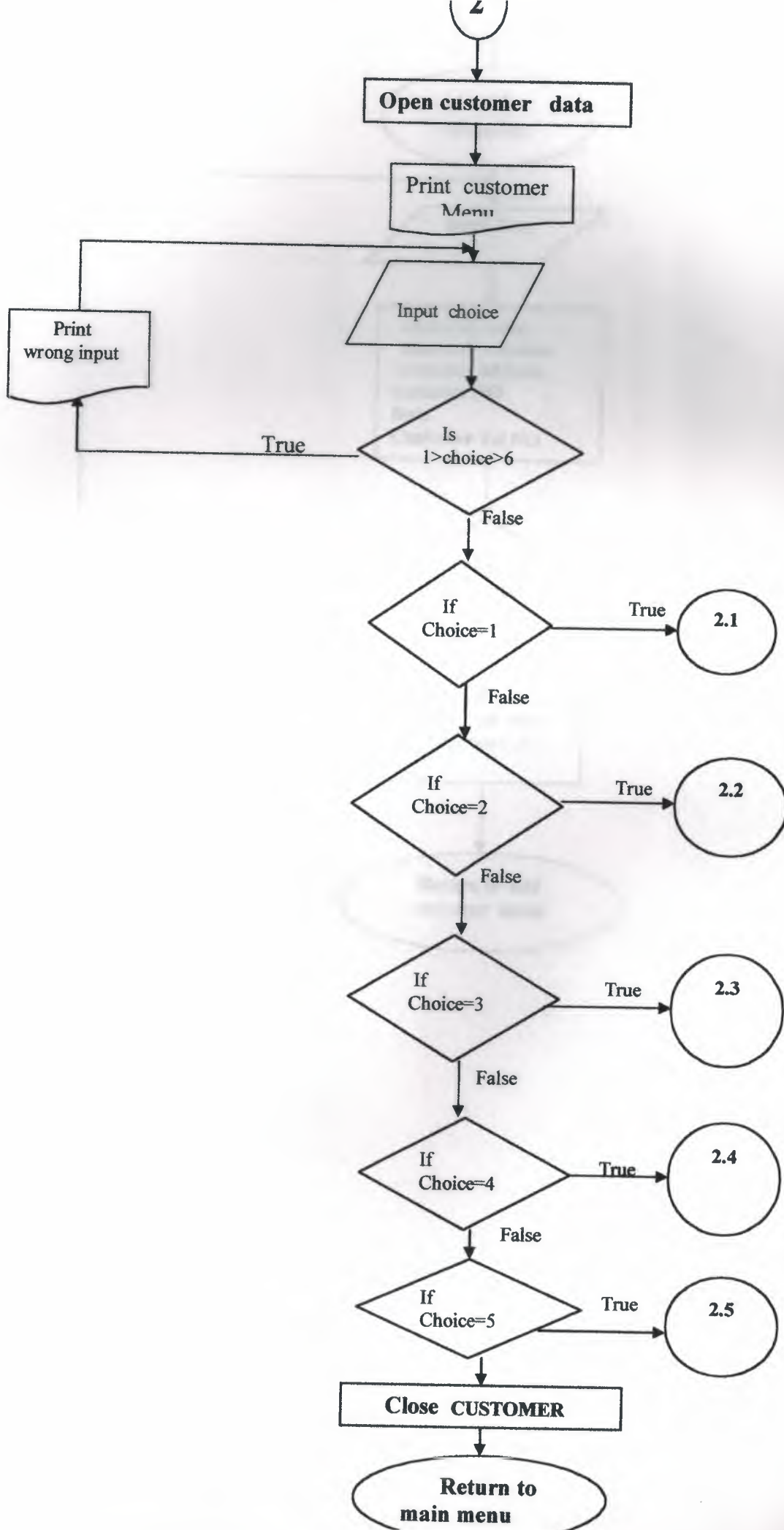




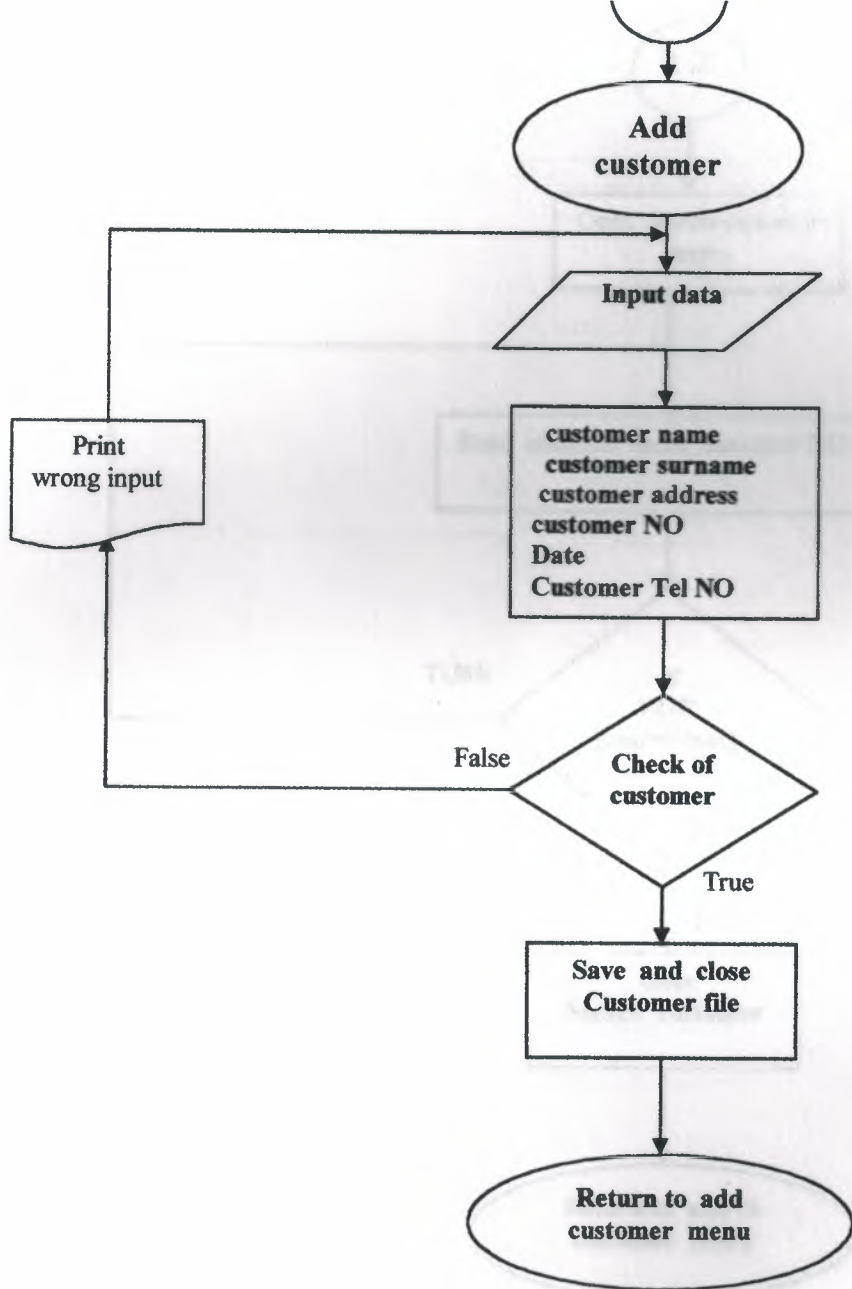












2.2

Open search customer  
menu

Read customer name ,customer NO

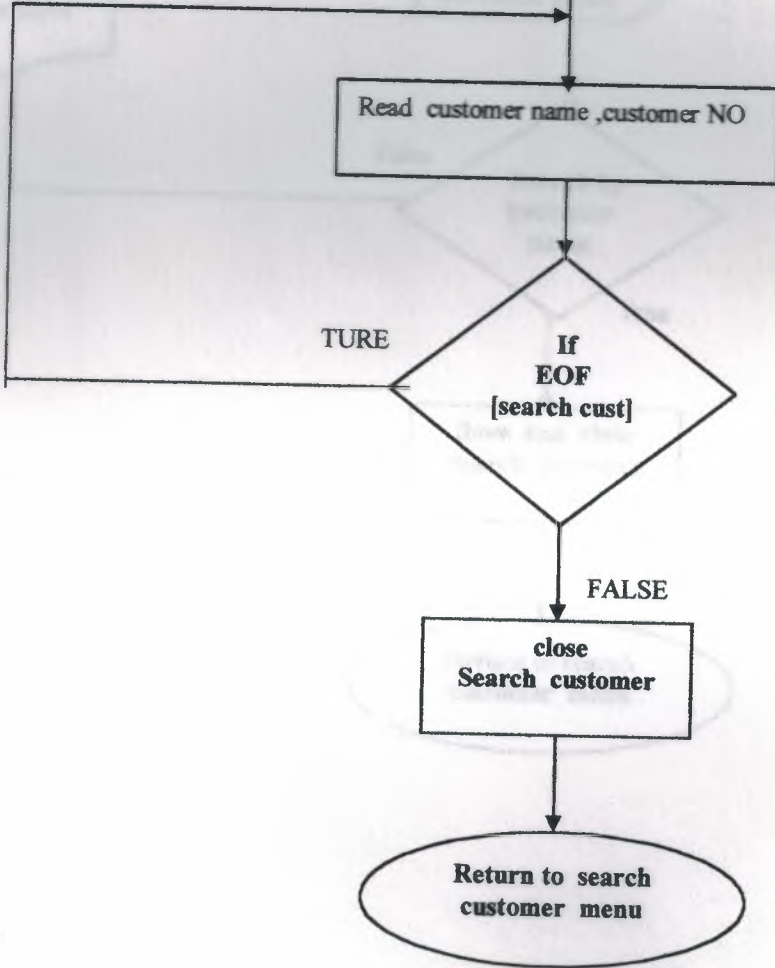
TURE

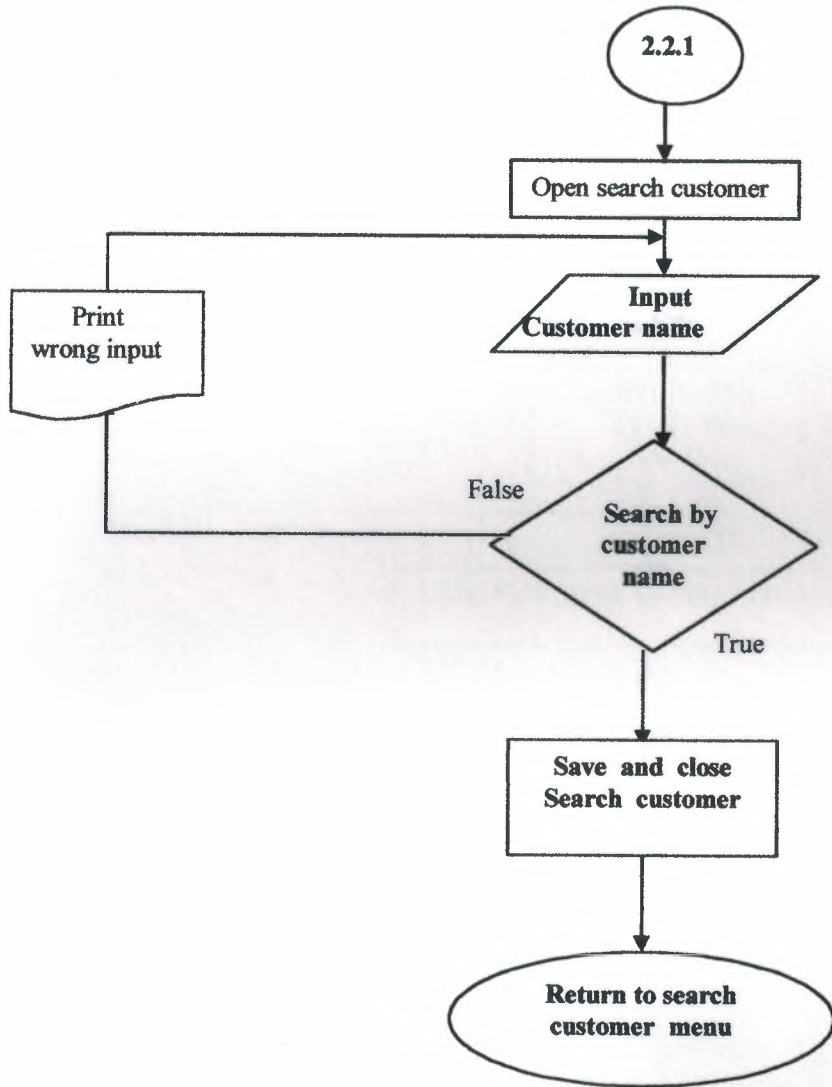
If  
EOF  
[search cust]

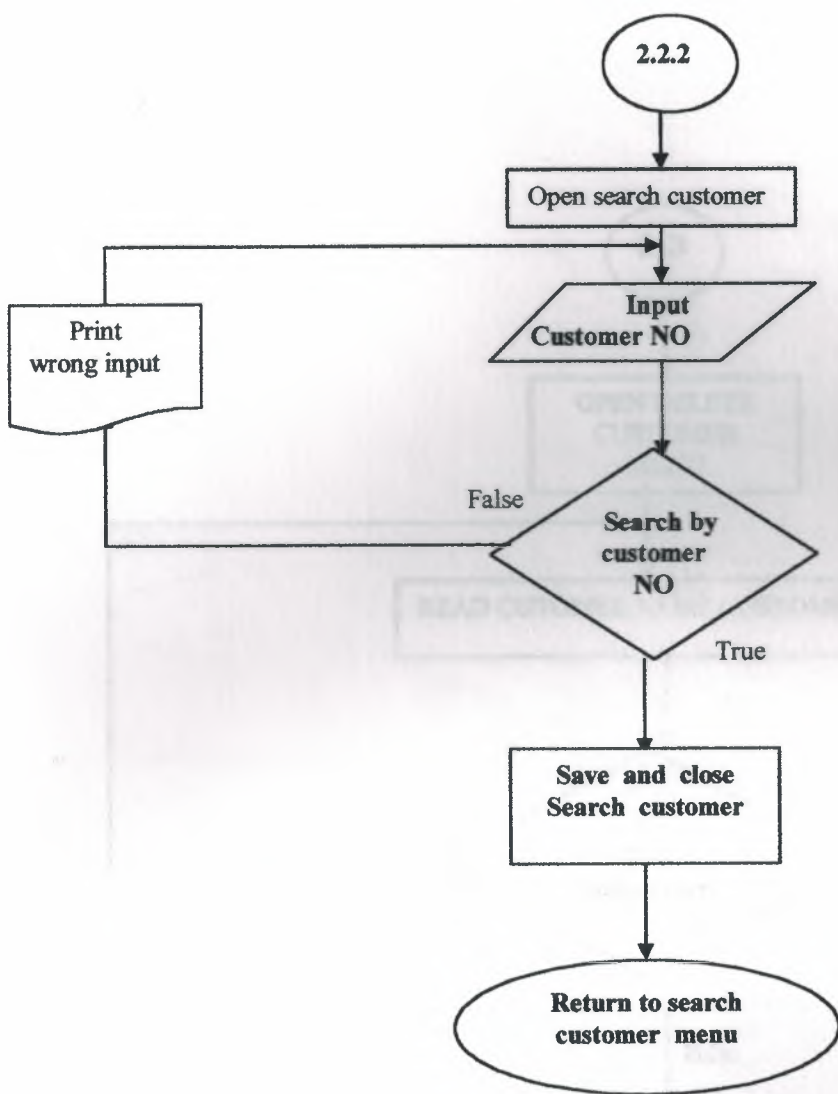
FALSE

close  
Search customer

Return to search  
customer menu









2.3

OPEN DELETE  
CUSTOMER  
MENU

READ CUTOMER NAME,CUSTOMER NO

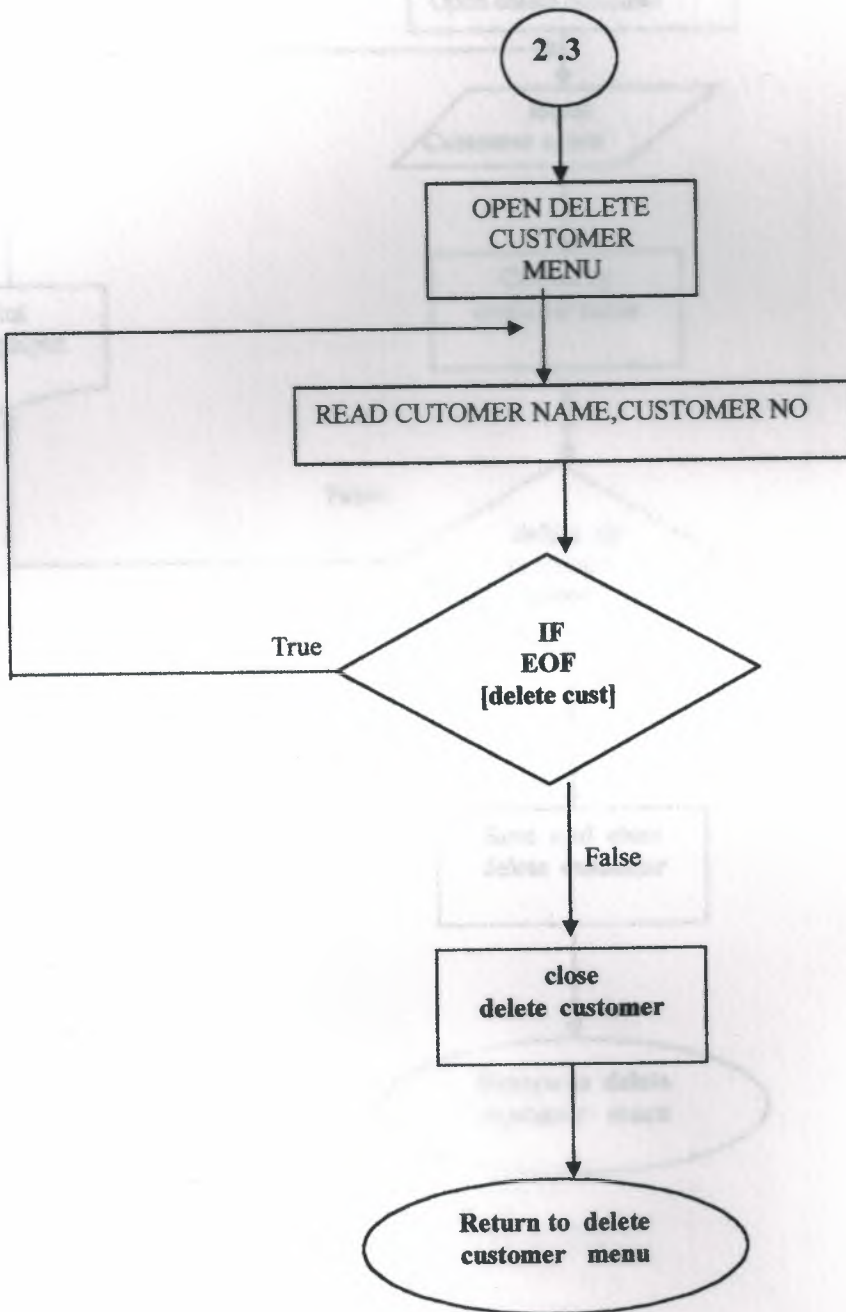
IF  
EOF  
[delete cust]

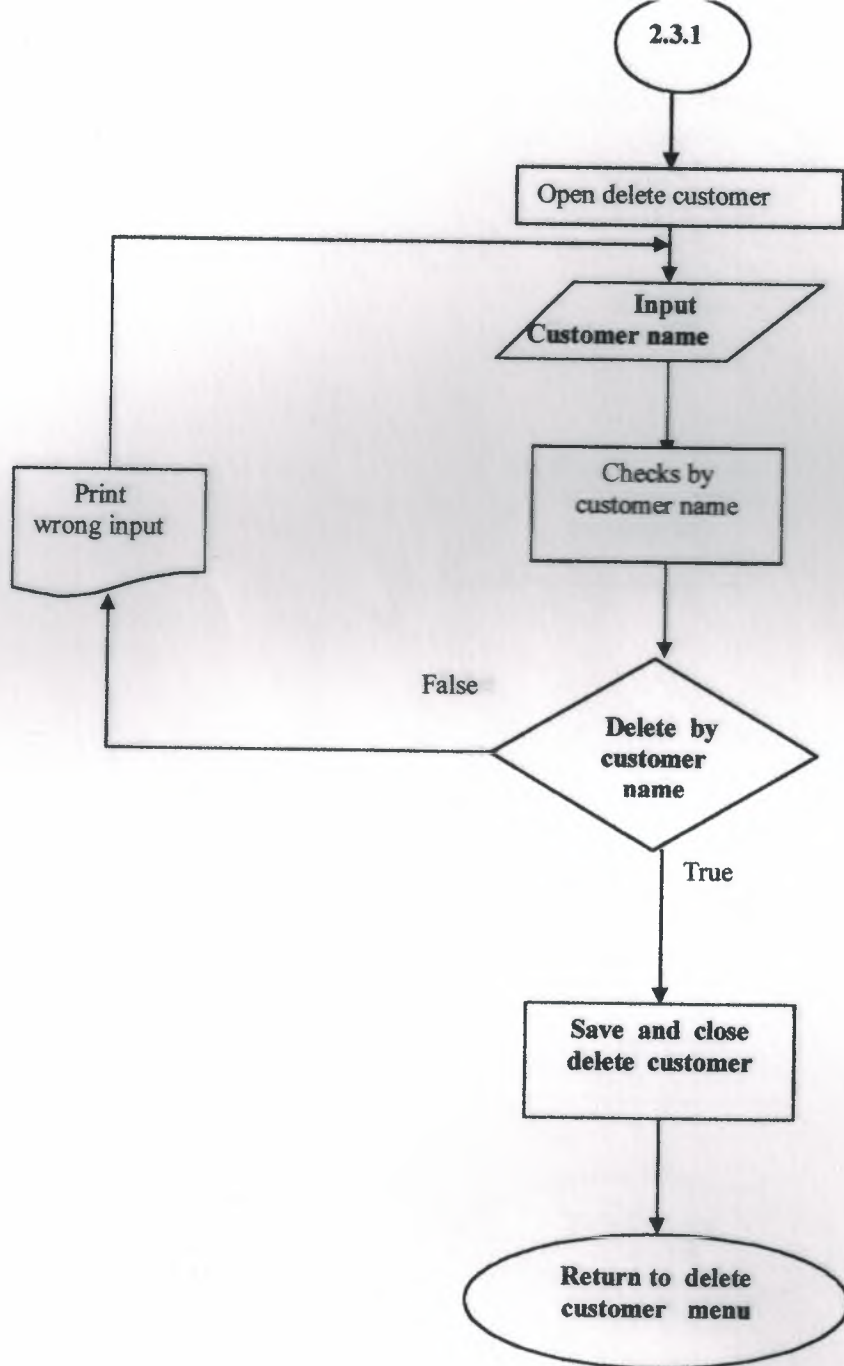
True

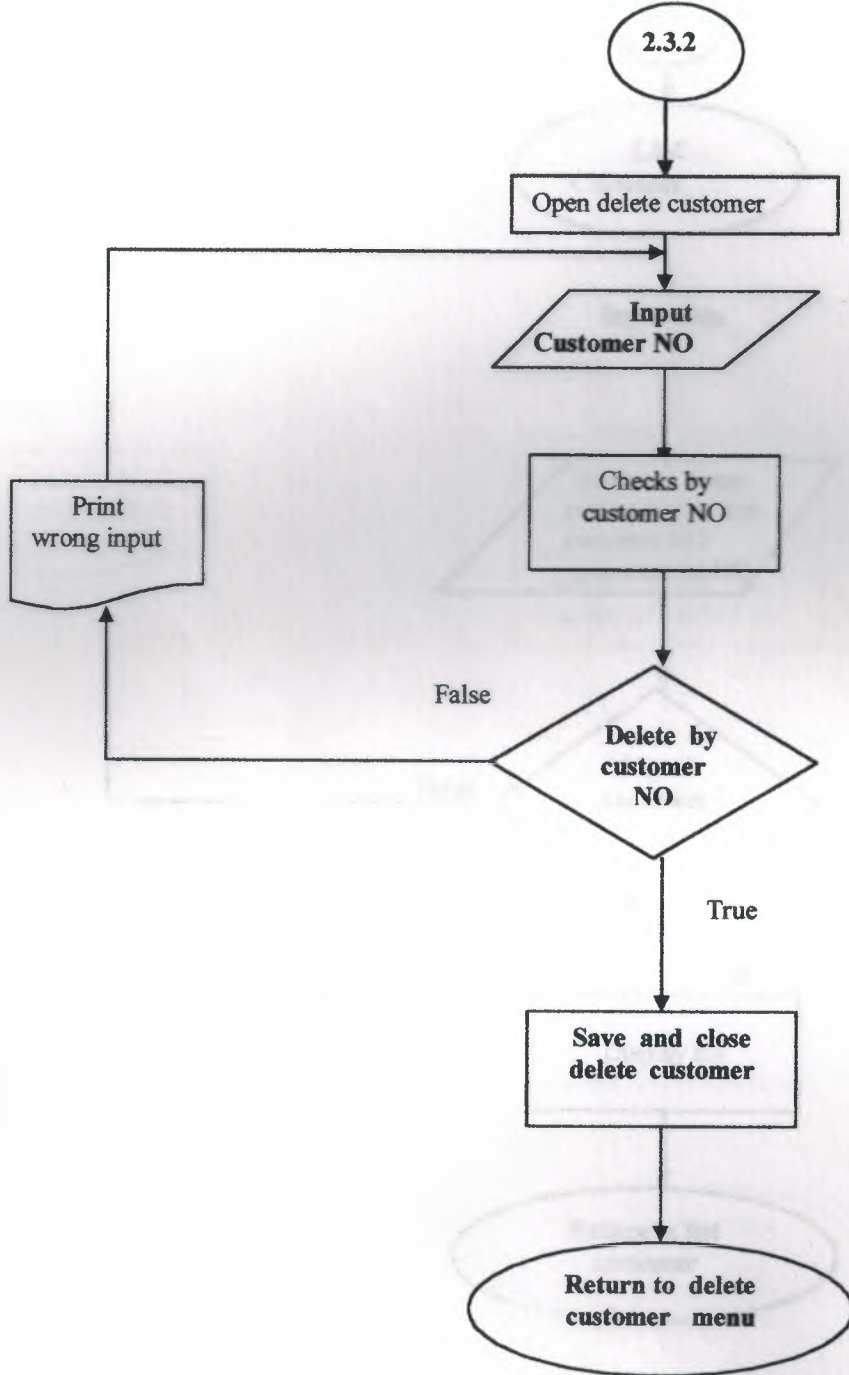
False

close  
delete customer

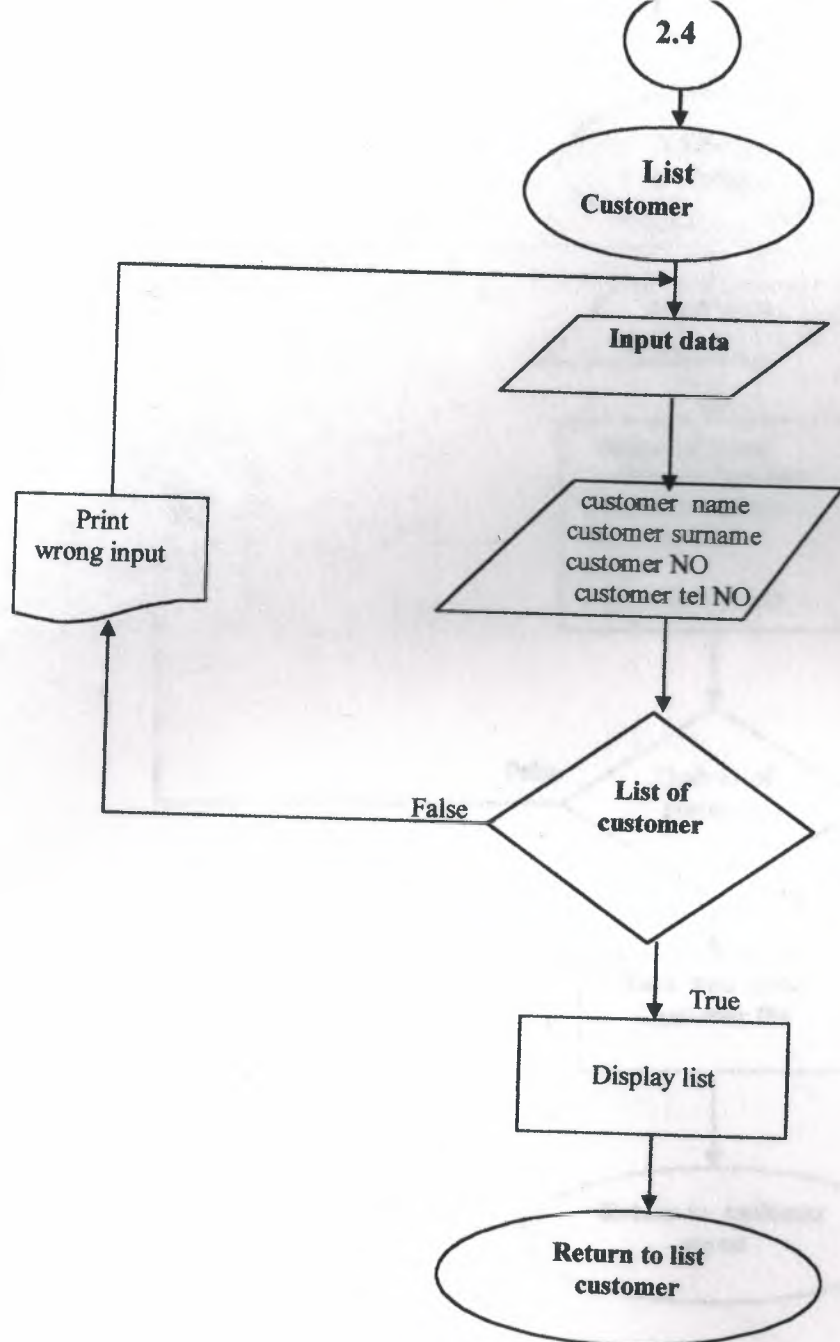
Return to delete  
customer menu





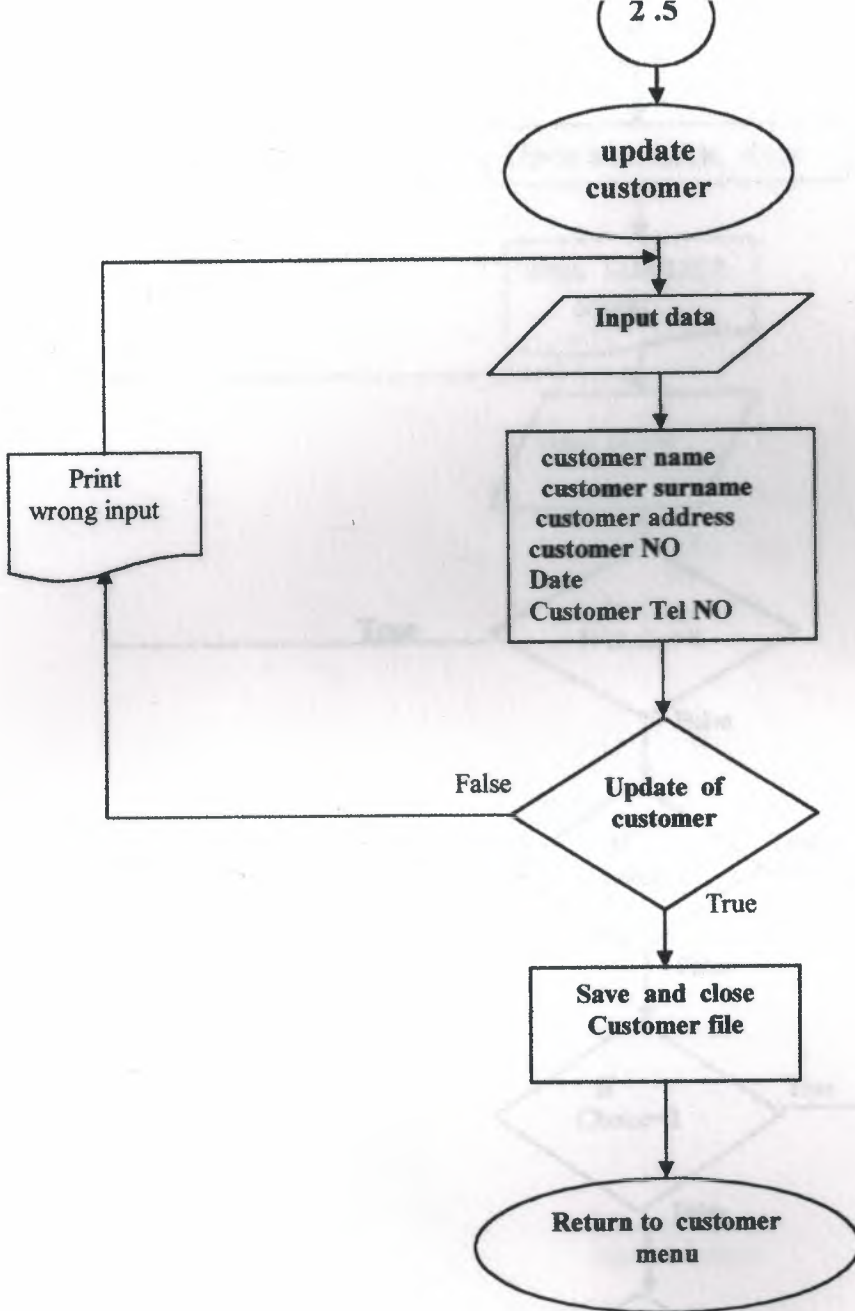


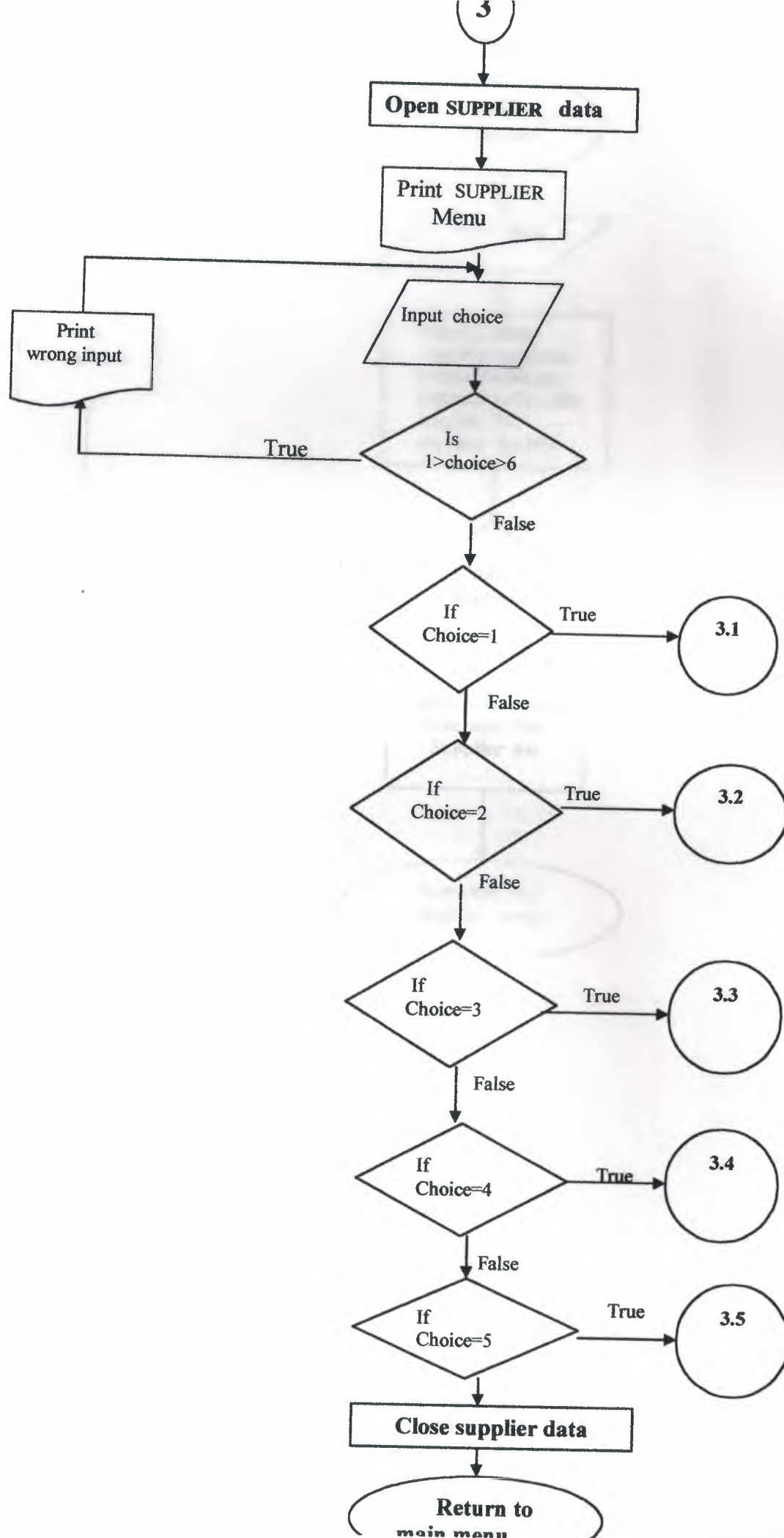
2.4



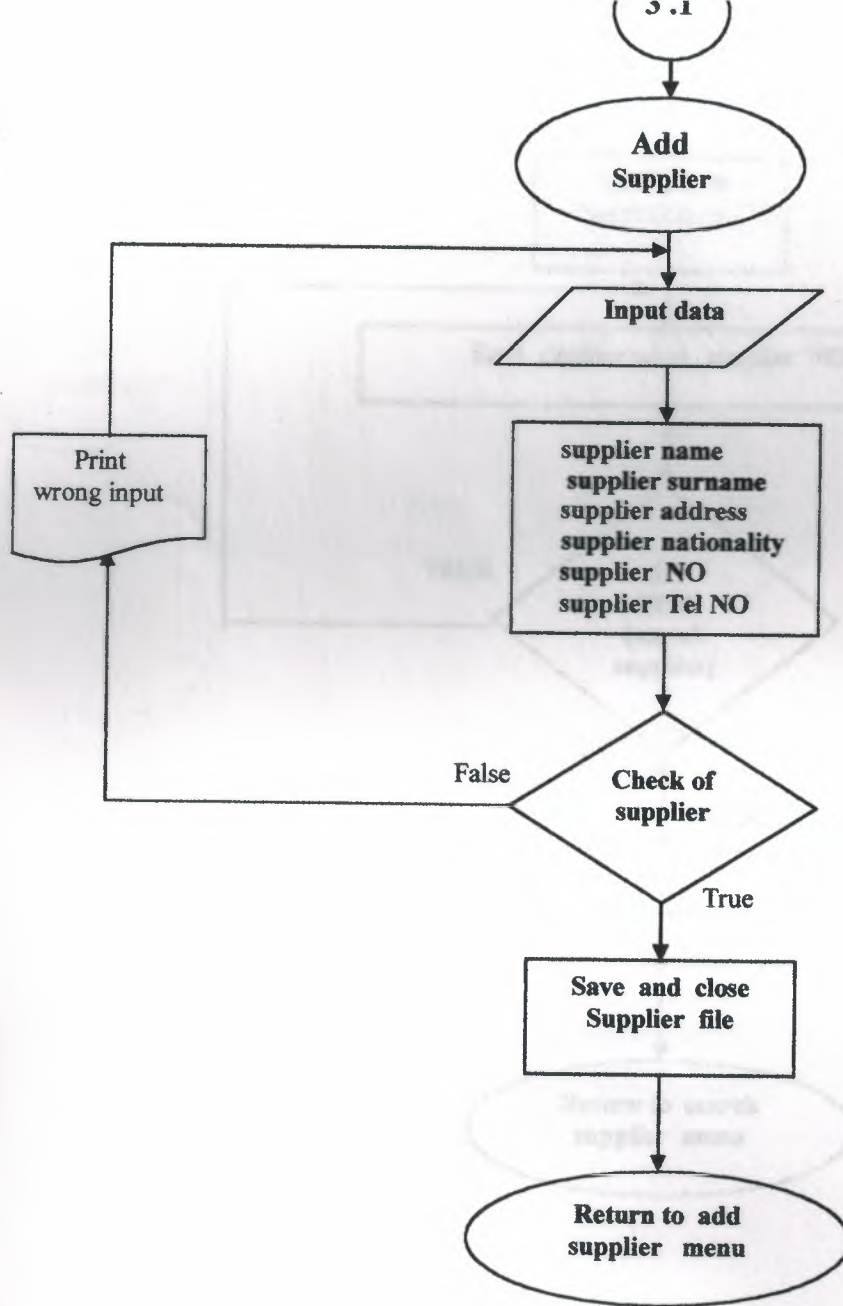


2.5





3.1



3.2

Open search  
Supplier menu

Read supplier name , supplier NO

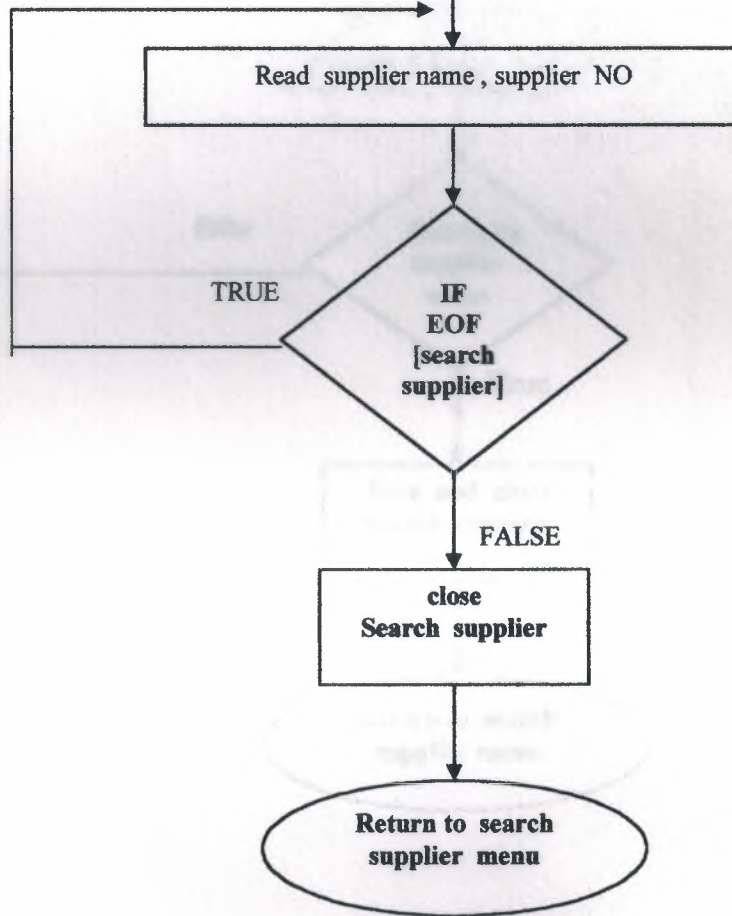
TRUE

IF  
EOF  
[search  
supplier]

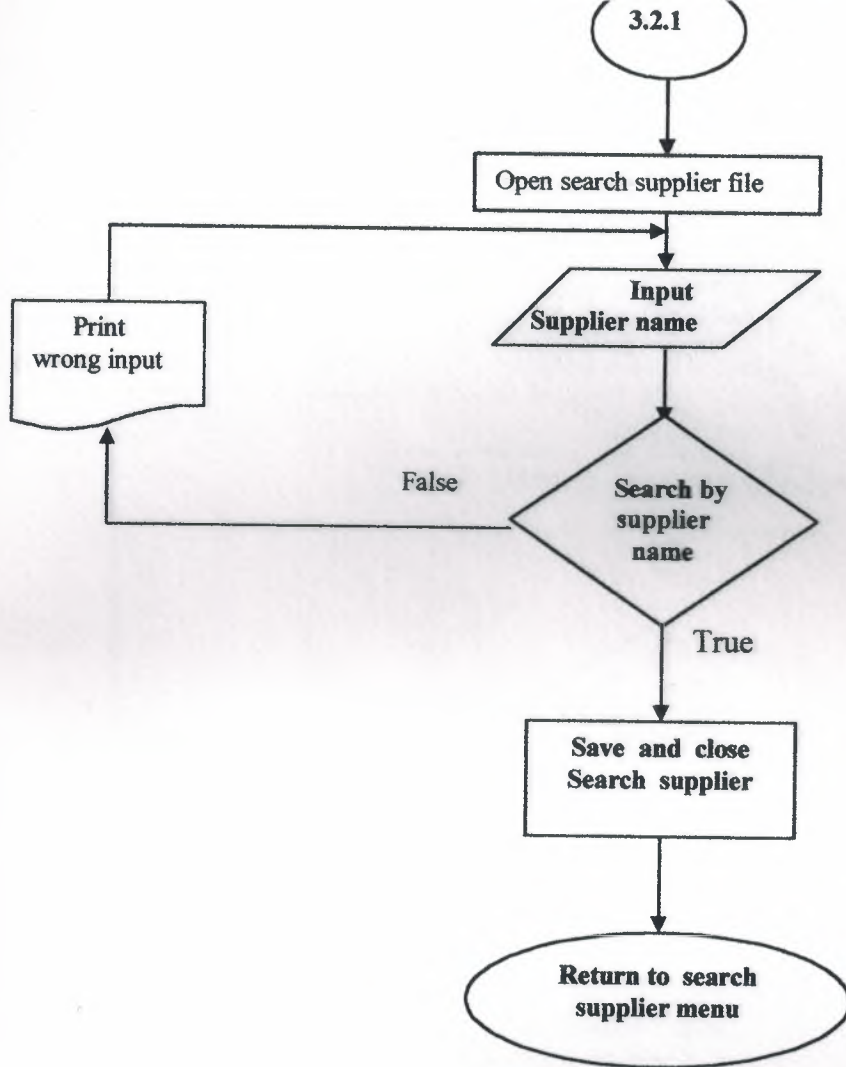
FALSE

close  
Search supplier

Return to search  
supplier menu







3.2.2

Open search supplier file

Input  
Supplier NO

Search by  
supplier  
NO

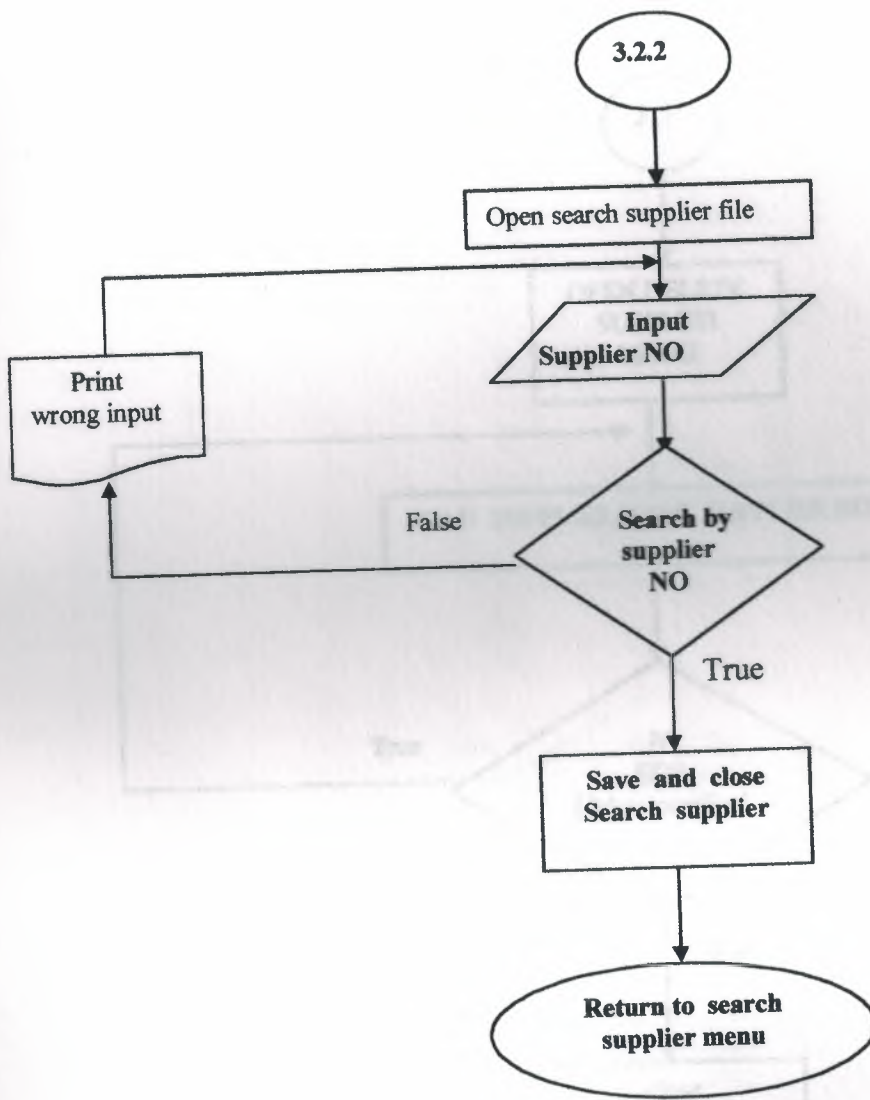
False

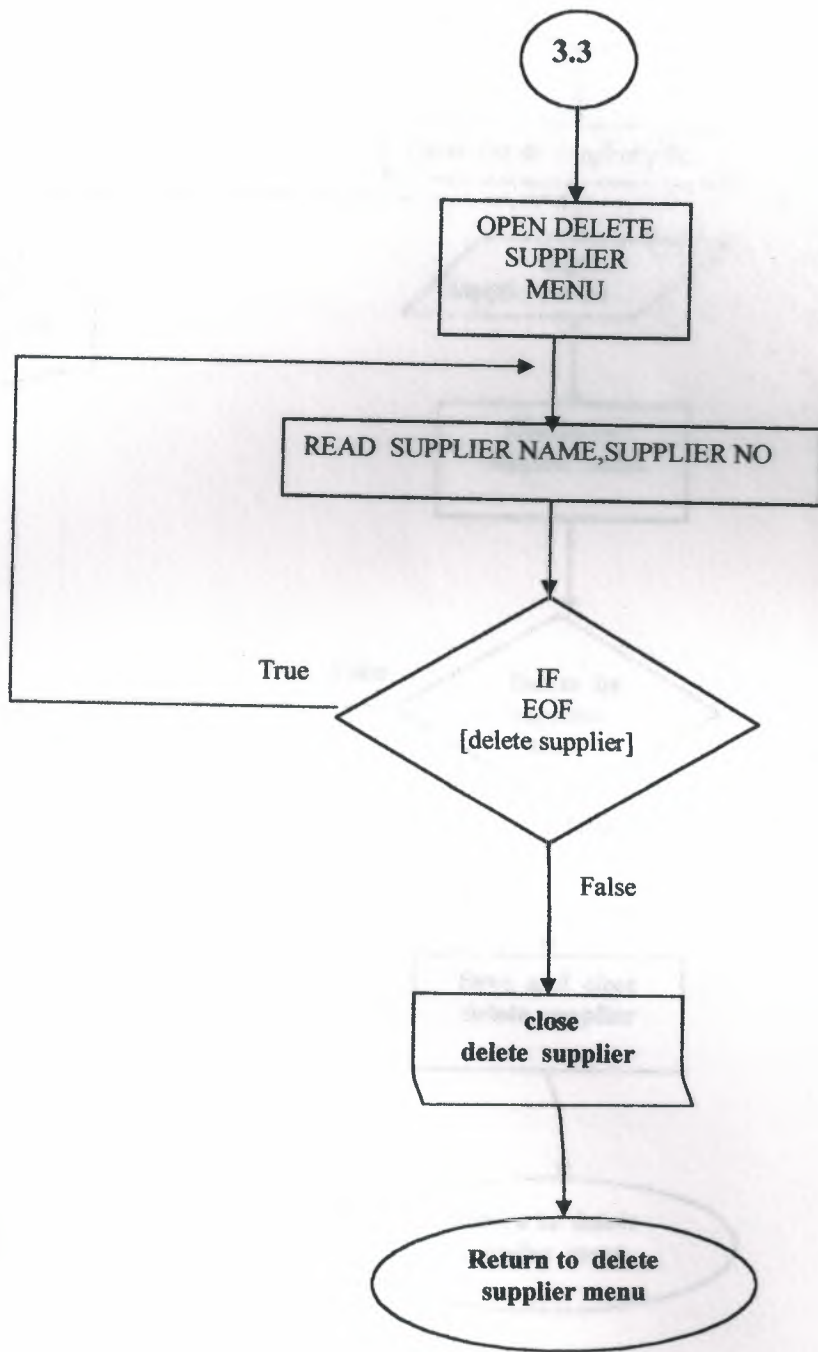
True

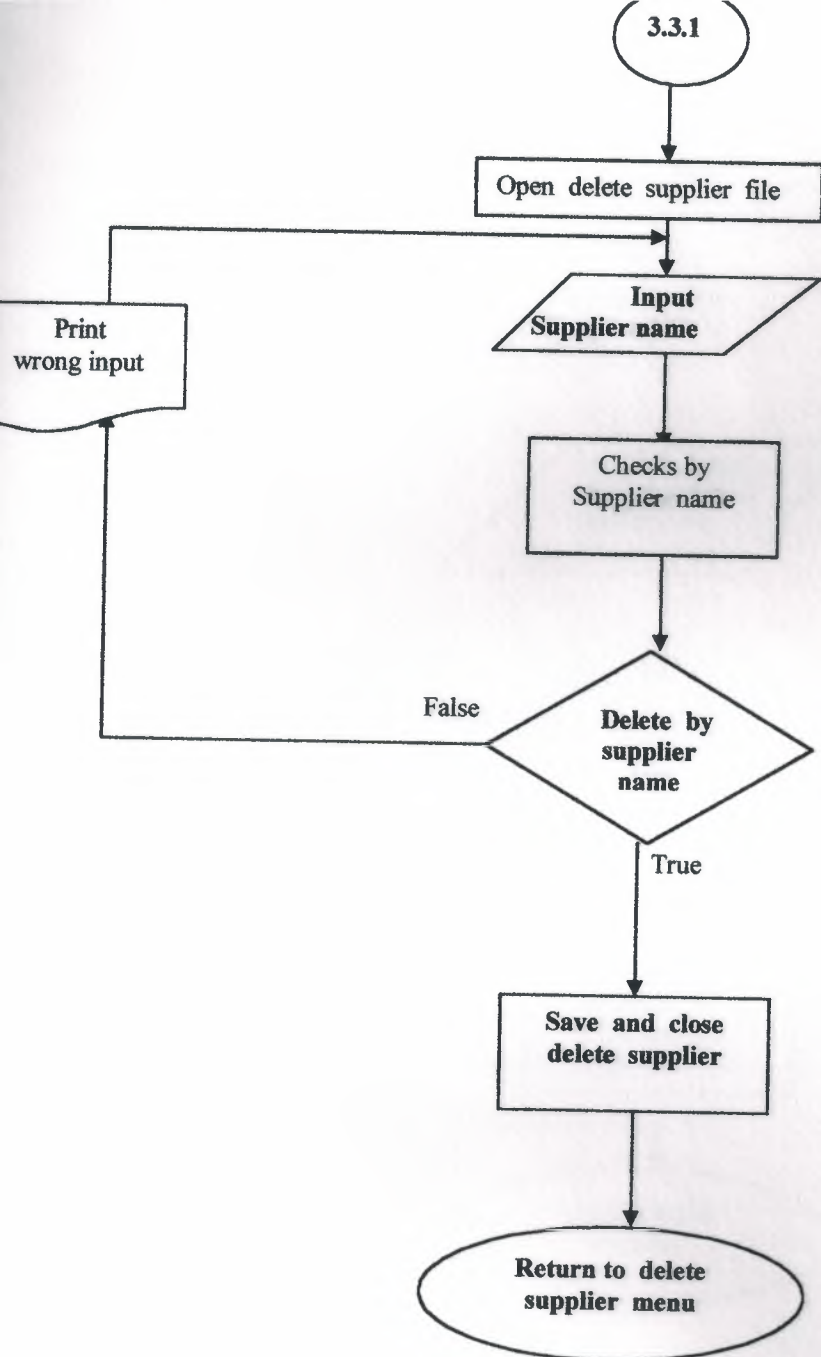
Save and close  
Search supplier

Return to search  
supplier menu

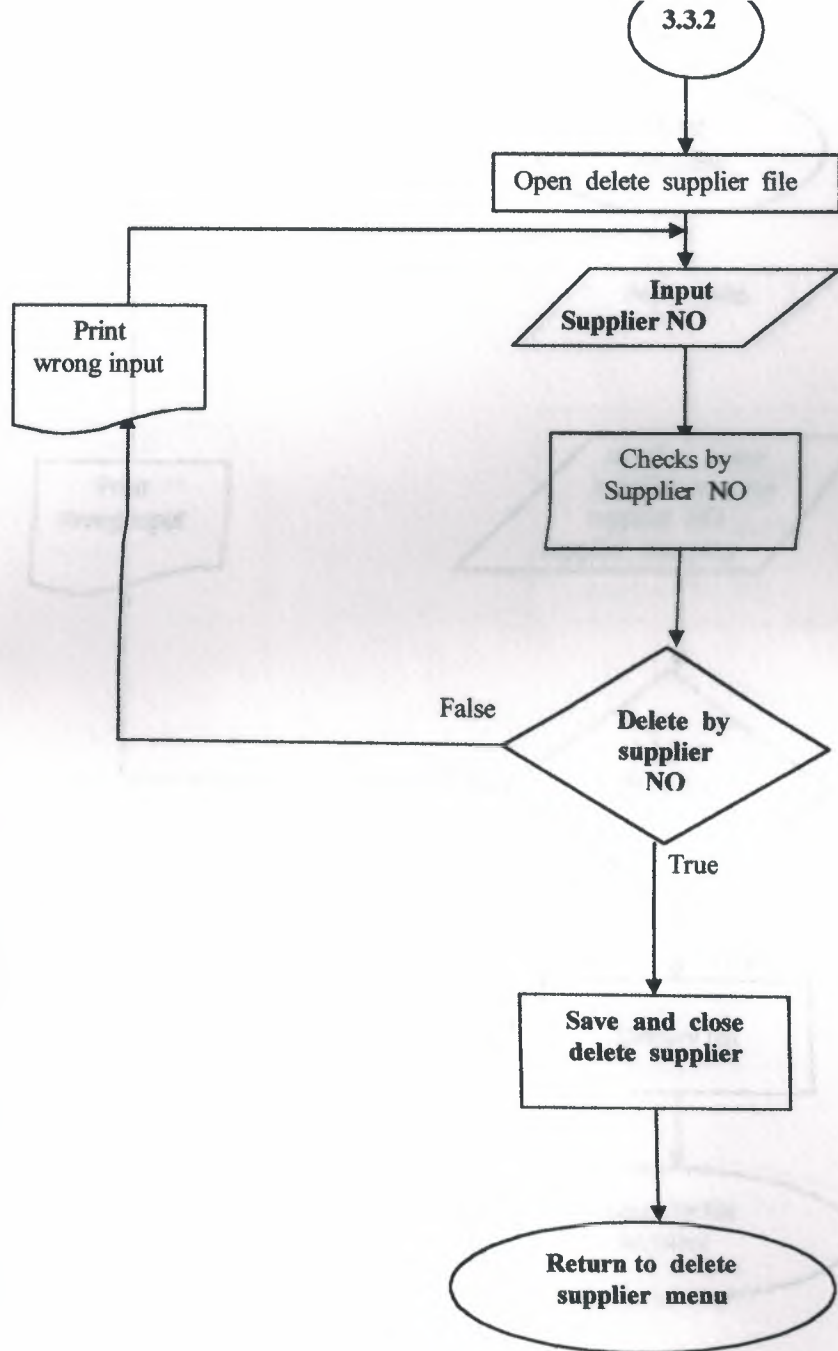
Print  
wrong input

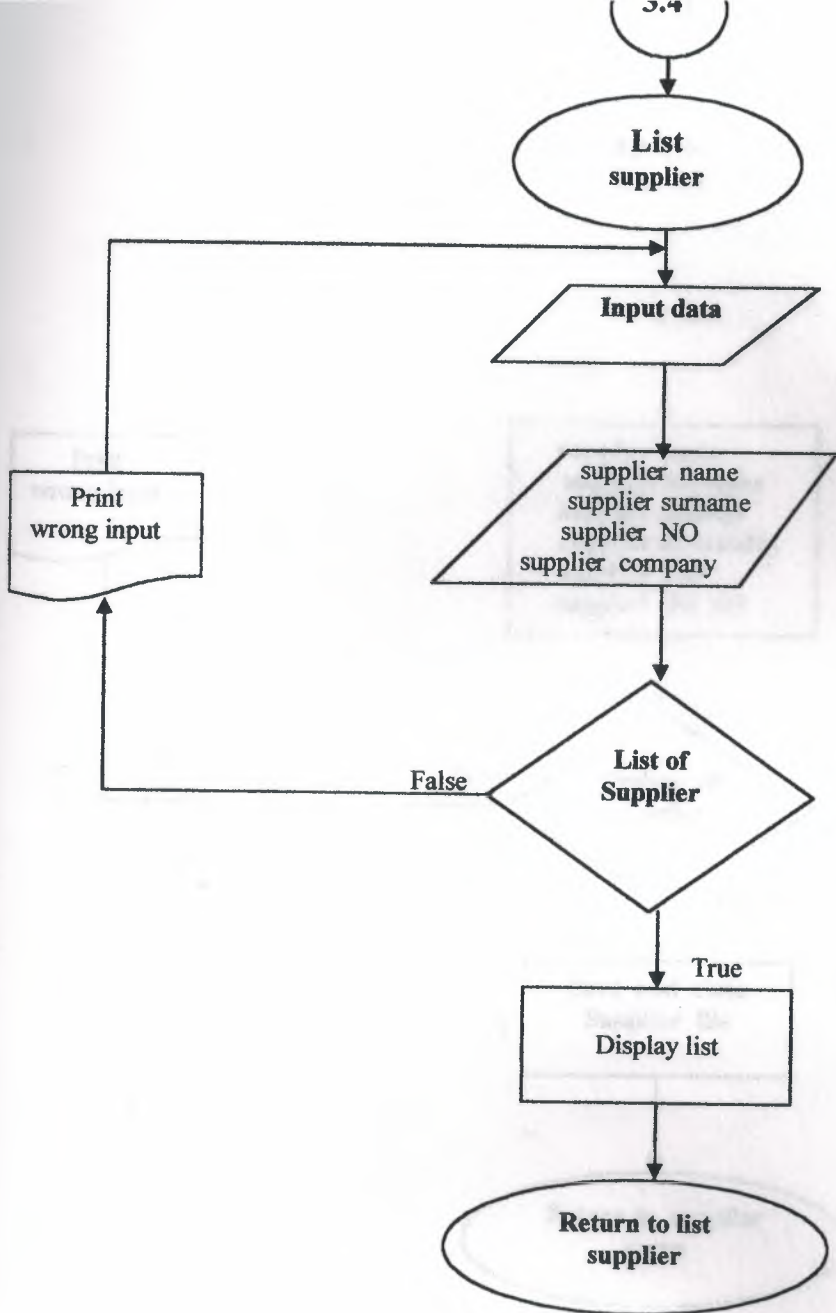


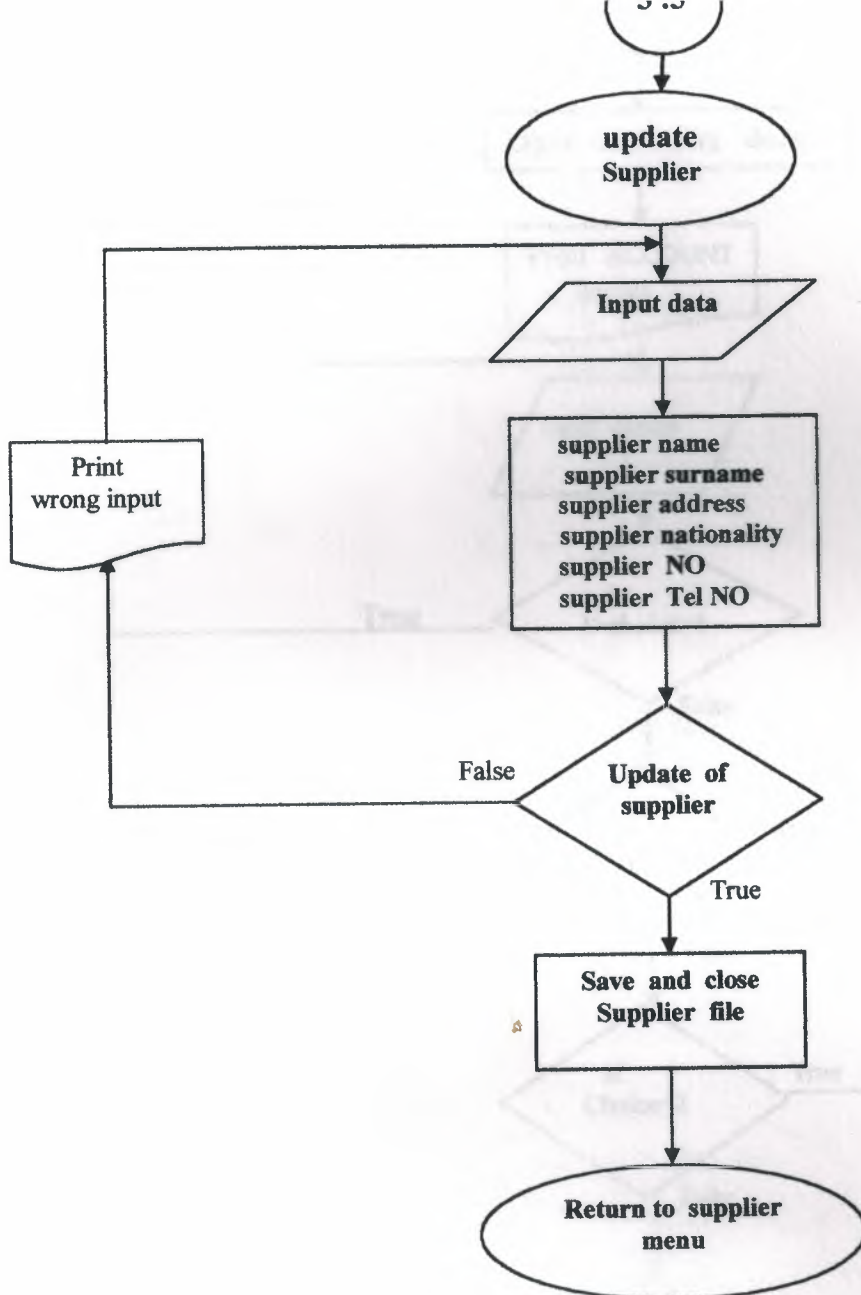


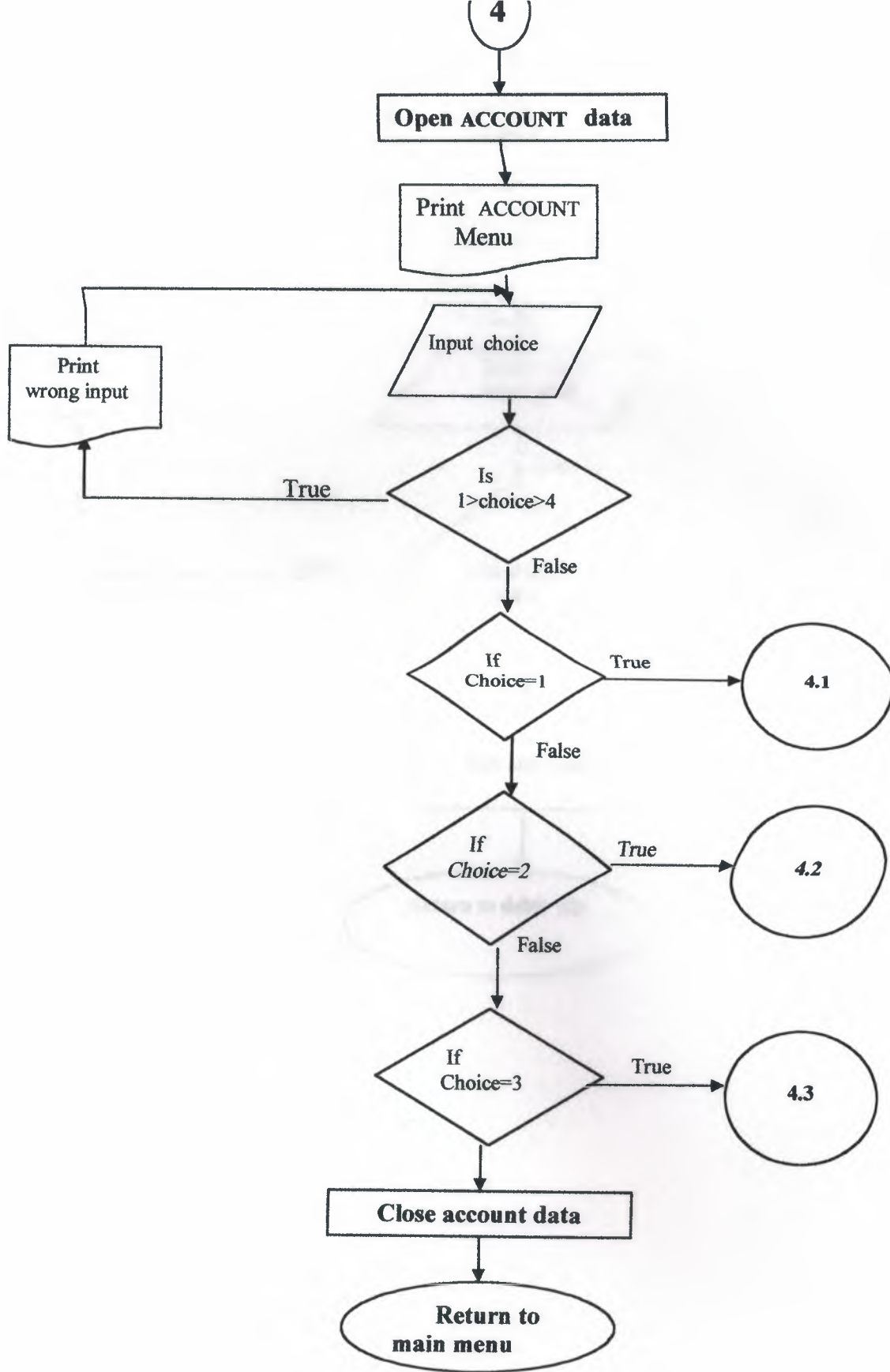




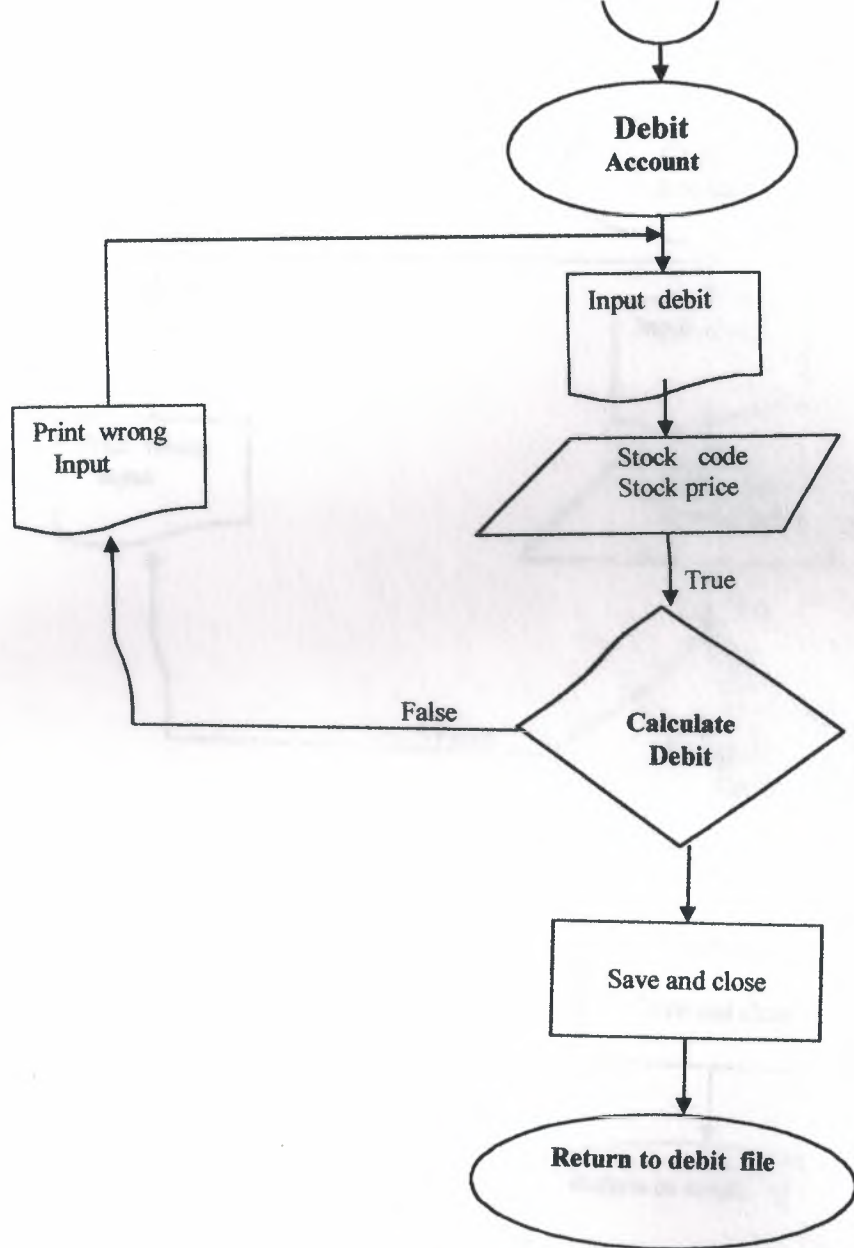


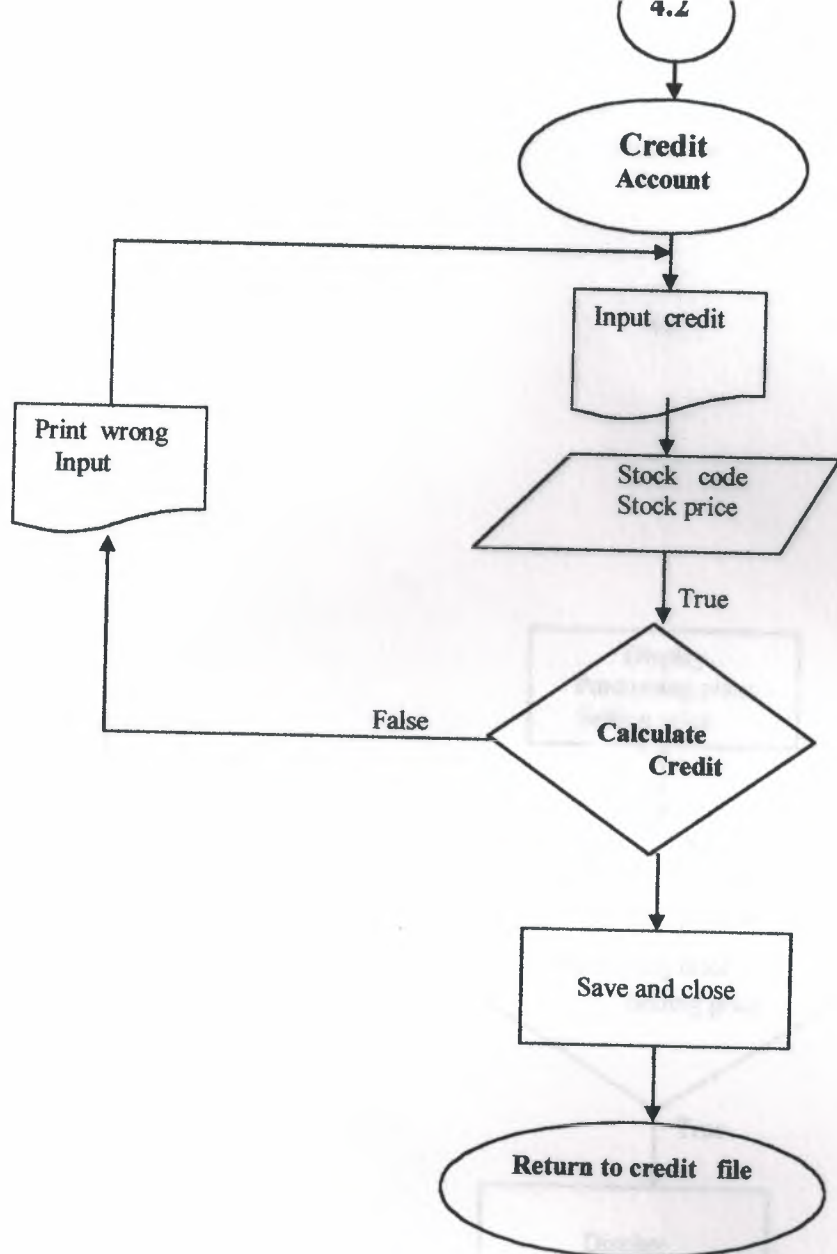


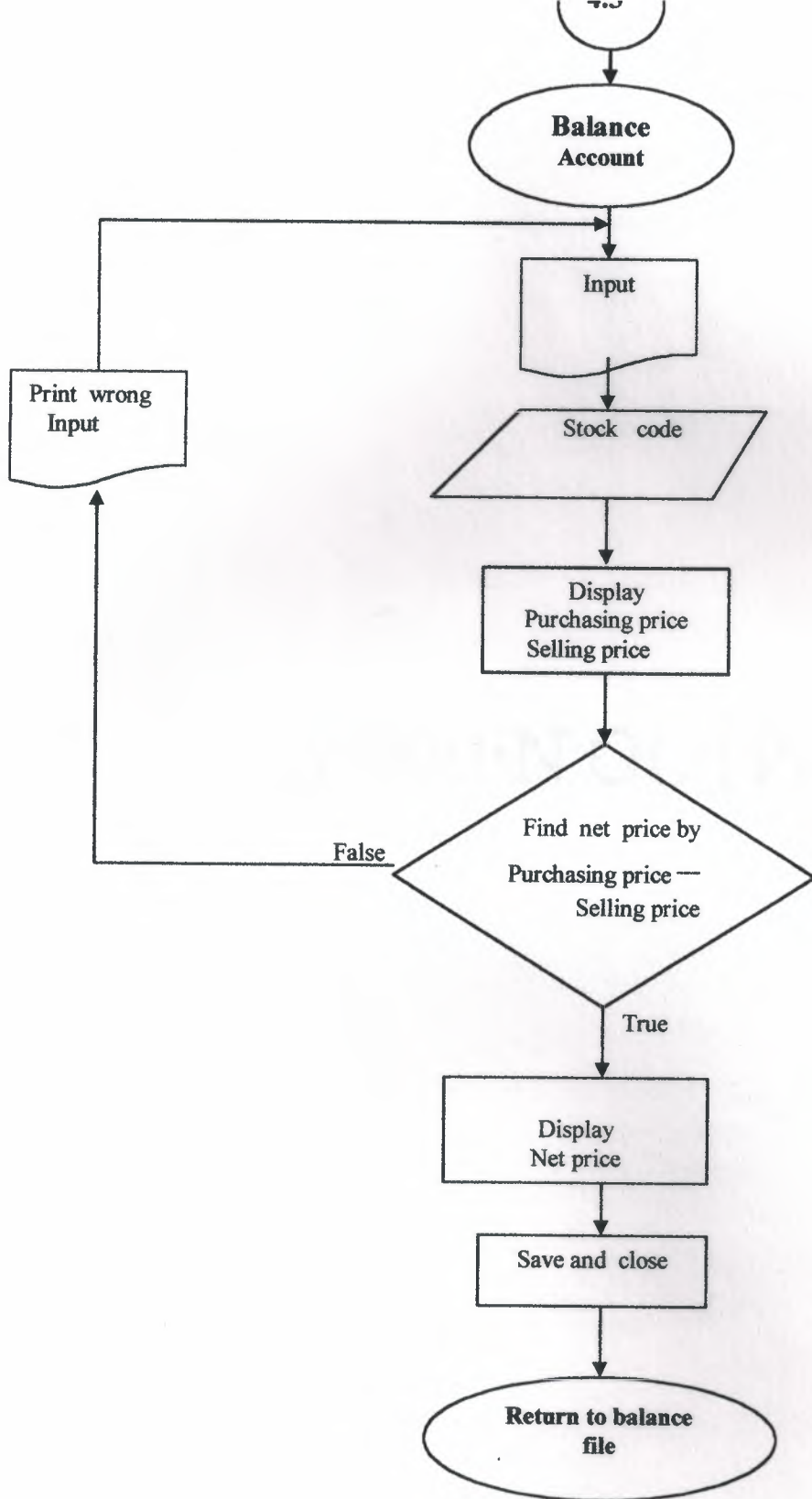












## **- SCREEN OUTPUT**



## **VII. REFERENCES**

## **SOURCE PROGRAM**

64

```

CLEAR ALL
CLOSE ALL
SET SYSMENU OFF
SET SAFETY OFF
SET TALK OFF
SET STATUS BAR OFF
SET SYSMENU ON

```

```

IF NOT WEXIST("_0z01g7q5j")
    DEFINE WINDOW _0z01g7q5j ;
        AT 0.000, 0.000 ;
        SIZE 37.769,103.600 ;
        FONT "MS Sans Serif", 8 ;
        TITLE "SEMEAL MARKET" ;
        FLOAT ;
        NOCLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,,0,0,255)
        MOVE WINDOW _0z01g7q5j CENTER
ENDIF

```

```

* *****
*
*                               MAIN/Windows Screen Layout
*
* *****
*

```

```

#REGION 1
IF WVISIBLE("_0z01g7q5j")
    ACTIVATE WINDOW _0z01g7q5j SAME
ELSE
    ACTIVATE WINDOW _0z01g7q5j NOSHOW
ENDIF

```

```

IF NOT WVISIBLE("_0z01g7q5j")
    ACTIVATE WINDOW _0z01g7q5j
ENDIF

```

```

READ CYCLE

```

```

PROCEDURE BT
    PARAMETER m.btnname
    DO CASE
CASE m.btnname='stock'
    Do c:\project\stock.prg
CASE m.btnname='customer'
    DO c:\project\customer.prg
CASE m.btnname='supplier'

```

```
DO c:\project\supplier.prg
CASE m.btnname='account'
DO c:\project\account.prg
CASE m.btnname='EXIT'
set sysm to defa
DEACTIVATE WINDOW _0z01g7q5j
RELEASE WINDOW _0z01g7q5j
ENDCASE
RETURN
```



```

IF NOT WEXIST("_0yx0uv09t")
    DEFINE WINDOW _0yx0uv09t ;
        AT 0.000, 0.000 ;
        SIZE 23.538,54.600 ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,192,192,192)
        MOVE WINDOW _0yx0uv09t CENTER
ENDIF

*****
*
*          STOCK/Windows Screen Layout
*
*****

#REGION 1
IF WVISIBLE("_0yx0uv09t")
    ACTIVATE WINDOW _0yx0uv09t SAME
ELSE
    ACTIVATE WINDOW _0yx0uv09t NOSHOW
ENDIF

IF NOT WVISIBLE("_0yx0uv09t")
    ACTIVATE WINDOW _0yx0uv09t
ENDIF

READ CYCLE

PROCEDURE BT
    PARAMETER m.btnname
    DO CASE
CASE m.btnname='add'
    Do c:\project\st.prg
CASE m.btnname='search'
    DO c:\project\stsearch.prg
CASE m.btnname='Delet'
    DO c:\project\stdelet.prg
CASE m.btnname='LIST'
    DO c:\project\stock.qpr
CASE m.btnname='EXIT'
    set sysm to defa
DEACTIVATE WINDOW _0yx0uv09t
RELEASE WINDOW _0yx0uv09t
ENDCASE

```

```

IF NOT WEXIST("_Oyx12kk71")
    DEFINE WINDOW _Oyx12kk71 ;
        AT 0.000, 0.000 ;
        SIZE 22.077,52.800 ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,,0,128,128)
        MOVE WINDOW _Oyx12kk71 CENTER
ENDIF

*
* *****
*
* STSEARCH/Windows Screen Layout
*
* *****
*

#REGION 1
IF WVISIBLE("_Oyx12kk71")
    ACTIVATE WINDOW _Oyx12kk71 SAME
ELSE
    ACTIVATE WINDOW _Oyx12kk71 NOSHOWN
ENDIF

IF NOT WVISIBLE("_Oyx12kk71")
    ACTIVATE WINDOW _Oyx12kk71
ENDIF

READ CYCLE
PROCEDURE BT
    PARAMETER m.btnname
do case
CASE m.btnname='EXIT'
set sysm to defa
DEACTIVATE WINDOW _Oyx12kk71
RELEASE WINDOW _Oyx12kk71
ENDCASE
RETURN

*****
procedure brosedata
DEFINE WINDOW aal ;
    AT 0.000, 0.000 ;
    SIZE 37.075,120.333 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;

```

```

        FLOAT ;
        CLOSE ;
MINIMIZE
        MOVE WINDOW aa1 CENTER
        activate window aa1
BROWSE WINDOW AA1
        deactivate window aa1
        return
*****PROCEDURE WHEN THE CHOICE FOUND*****
PROCEDURE FIND
IF NOT WEXIST("_03910y9u2")
        DEFINE WINDOW _03910y9u2 ;
                AT 0.000, 0.000 ;
                SIZE 9.231,55.000 ;
                FONT "MS Sans Serif", 8 ;
                FLOAT ;
                CLOSE ;
                MINIMIZE ;
                SYSTEM ;
                COLOR RGB(,,,255,255,255)
                MOVE WINDOW _03910y9u2 CENTER
ENDIF

IF WVISIBLE("_03910y9u2")
        ACTIVATE WINDOW _03910y9u2 SAME
ELSE
        ACTIVATE WINDOW _03910y9u2 NOSHOW
ENDIF
@ 1.769,14.400 SAY "Searching It's Ok" ;
        FONT "MS Sans Serif", 10 ;
        STYLE "BT"
@ 5.231,2.200 SAY "Press The Enter Key To Continue !" ;
        FONT "MS Sans Serif", 10 ;
        STYLE "BT"
        IF NOT WVISIBLE("_03910y9u2")
ACTIVATE WINDOW _03910y9u2
        ENDIF
read
release window _03910y9u2
do brosedata
return
*****PROCEDURE WHEN THE CHOICE NOT FOUND*****
PROCEDURE NOT
IF NOT WEXIST("_03910y9u2")
        DEFINE WINDOW _03910y9u2 ;
                AT 0.000, 0.000 ;
                SIZE 9.231,55.000 ;
                FONT "MS Sans Serif", 8 ;
                FLOAT ;
                CLOSE ;
                MINIMIZE ;
                SYSTEM ;

```

```

                                COLOR RGB(,,,255,255,255)
                                MOVE WINDOW _03910y9u2 CENTER
ENDIF

IF WVISIBLE("_03910y9u2")
    ACTIVATE WINDOW _03910y9u2 SAME
ELSE
    ACTIVATE WINDOW _03910y9u2 NOSHOW
ENDIF
@ 1.769,14.400 SAY "It's Not Found" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
@ 5.231,2.200 SAY "Press The Enter Key To Continue !" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
    IF NOT WVISIBLE("_03910y9u2")
        ACTIVATE WINDOW _03910y9u2
    ENDIF
read
release window _03910y9u2.

RETURN

*****      PROCEDURE SEARCHING BY NAME      *****
procedure NAME
    CLOSE DATABASE
    IF NOT WEXIST("_0390zsz3a")
        DEFINE WINDOW _0390zsz3a ;
            AT 0.000, 0.000 ;
            SIZE 6.538,70.200 ;
            FONT "MS Sans Serif", 8 ;
            FLOAT ;
            NOCLOSE ;
            MINIMIZE ;
            SYSTEM ;
            COLOR RGB(,,,255,255,255)
            MOVE WINDOW _0390zsz3a CENTER
    ENDIF
    IF WVISIBLE("_0390zsz3a")
        ACTIVATE WINDOW _0390zsz3a SAME
    ELSE
        ACTIVATE WINDOW _0390zsz3a NOSHOW
    ENDIF
    @ 2.846,7.800 SAY "Enter Stock Name:" ;
        FONT "MS Sans Serif", 10 ;
        STYLE "BT"
    IF NOT WVISIBLE("_0390zsz3a")
        ACTIVATE WINDOW _0390zsz3a
    ENDIF
    your_ch=space(10)
    @2.999,34.800 get your_ch
    read

```



```

release window _0390zsz3a
use c:\project\stock.dbf
LOCATE ALL FOR UPPER(ALLTRIM(Stname))= UPPER(ALLTRIM(your_ch))
IF FOUND() THEN
    do find
ELSE
    DO not
ENDIF
return
***** PROCEDURE SEARCHING BY NO*****
procedure CODE
    CLOSE DATABASE
    IF NOT WEXIST("_0390zsz3a")
    DEFINE WINDOW _0390zsz3a ;
        AT 0.000, 0.000 ;
        SIZE 6.538,70.200 ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        NOCLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,255,255,255)
        MOVE WINDOW _0390zsz3a CENTER
    ENDIF
    IF WVISIBLE("_0390zsz3a")
        ACTIVATE WINDOW _0390zsz3a SAME
    ELSE
        ACTIVATE WINDOW _0390zsz3a NOSHOWN
    ENDIF
    @ 2.846,7.800 SAY "Enter Stock Code: " ;
        FONT "MS Sans Serif", 10 ;
        STYLE "BT"
    IF NOT WVISIBLE("_0390zsz3a")
    ACTIVATE WINDOW _0390zsz3a
    ENDIF
    your_ch=space(10)
    @2.999,34.800 get your_ch
    read
    release window _0390zsz3a
    use c:\project\stock.dbf
    LOCATE ALL FOR UPPER(ALLTRIM(Stcode))= UPPER(ALLTRIM(your_ch))
    IF FOUND() THEN
        DO find
    ELSE
        DO not
    ENDIF

```

```

IF NOT WEXIST("_0yy0wt928")
    DEFINE WINDOW _0yy0wt928 ;
        AT 0.000, 0.000 ;
        SIZE 22.077,52.800 ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        NOCLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,,0,0,255)
        MOVE WINDOW _0yy0wt928 CENTER
ENDIF

```

```

* *****
*
* STDELET/Windows Screen Layout
*
* *****
*

```

```

#REGION 1
IF WVISIBLE("_0yy0wt928")
    ACTIVATE WINDOW _0yy0wt928 SAME
ELSE
    ACTIVATE WINDOW _0yy0wt928 NOSHOW
ENDIF

```

```

IF NOT WVISIBLE("_0yy0wt928")
    ACTIVATE WINDOW _0yy0wt928
ENDIF

```

```

READ CYCLE
PROCEDURE BT
    PARAMETER m.btnname
do case
CASE m.btnname='EXIT'
set sysm to defa
DEACTIVATE WINDOW _0yy0wt928
RELEASE WINDOW _0yy0wt928
ENDCASE
RETURN

```

```

*****PROCEDURE WHEN THE CHOICE FOUND*****
PROCEDURE delet
IF NOT WEXIST("_03910y9u2")
    DEFINE WINDOW _03910y9u2 ;
        AT 0.000, 0.000 ;
        SIZE 9.231,55.000 ;

```

```

        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,,255,255,255)
        MOVE WINDOW _03910y9u2 CENTER
ENDIF

IF WVISIBLE("_03910y9u2")
    ACTIVATE WINDOW _03910y9u2 SAME
ELSE
    ACTIVATE WINDOW _03910y9u2 NOSHOW
ENDIF
@ 1.769,14.400 SAY "Deleting It's Ok" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
@ 5.231,2.200 SAY "Press The Enter Key To Continue !" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
    IF NOT WVISIBLE("_03910y9u2")
        ACTIVATE WINDOW _03910y9u2
    ENDIF
read
release window _03910y9u2
PACK
return
*****PROCEDURE WHEN THE CHOICE NOT FOUND*****
PROCEDURE NOT
IF NOT WEXIST("_03910y9u2")
    DEFINE WINDOW _03910y9u2 ;
        AT 0.000, 0.000 ;
        SIZE 9.231,55.000 ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,,255,255,255)
        MOVE WINDOW _03910y9u2 CENTER
ENDIF

IF WVISIBLE("_03910y9u2")
    ACTIVATE WINDOW _03910y9u2 SAME
ELSE
    ACTIVATE WINDOW _03910y9u2 NOSHOW
ENDIF
@ 1.769,14.400 SAY "It's Not Found" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
@ 5.231,2.200 SAY "Press The Enter Key To Continue !" ;
    FONT "MS Sans Serif", 10 ;

```



```

        STYLE "BT"
        IF NOT WVISIBLE("_03910y9u2")
ACTIVATE WINDOW _03910y9u2
        ENDIF
read
release window _03910y9u2.
RETURN

*****      PROCEDURE SEARCHING BY NAME      *****
procedure NAME
        CLOSE DATABASE
        IF NOT WEXIST("_0390zsz3a")
            DEFINE WINDOW _0390zsz3a ;
                AT 0.000, 0.000 ;
                SIZE 6.538,70.200 ;
                FONT "MS Sans Serif", 8 ;
                FLOAT ;
                NOCLOSE ;
                MINIMIZE ;
                SYSTEM ;
                COLOR RGB(,,,255,255,255)
                MOVE WINDOW _0390zsz3a CENTER
        ENDIF
        IF WVISIBLE("_0390zsz3a")
            ACTIVATE WINDOW _0390zsz3a SAME
        ELSE
            ACTIVATE WINDOW _0390zsz3a NOSHOW
        ENDIF
        @ 2.846,7.800 SAY "Enter Stock Name:" ;
            FONT "MS Sans Serif", 10 ;
            STYLE "BT"
        IF NOT WVISIBLE("_0390zsz3a")
ACTIVATE WINDOW _0390zsz3a
        ENDIF
        your_ch=space(10)
        @2.999,34.800 get your_ch
        read
        release window _0390zsz3a
            use c:\project\stock.dbf
            LOCATE ALL FOR UPPER(ALLTRIM(Stname))= UPPER(ALLTRIM(your_ch))
        IF FOUND() THEN
            DELET
                do delet
            ELSE
                DO not
        ENDIF
return
*****      PROCEDURE SEARCHING BY NO*****
procedure CODE
        CLOSE DATABASE
        IF NOT WEXIST("_0390zsz3a")
            DEFINE WINDOW _0390zsz3a ;

```



```

        AT 0.000, 0.000 ;
        SIZE 6.538,70.200 ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        NOCLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,,255,255,255)
        MOVE WINDOW _0390zsz3a CENTER
ENDIF
IF WVISIBLE("_0390zsz3a")
    ACTIVATE WINDOW _0390zsz3a SAME
ELSE
    ACTIVATE WINDOW _0390zsz3a NOSHOW
ENDIF
@ 2.846,7.800 SAY "Enter Stock Code: " ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
IF NOT WVISIBLE("_0390zsz3a")
ACTIVATE WINDOW _0390zsz3a
ENDIF
    your_ch=space(10)
    @2.999,34.800 get your_ch
    read
    release window _0390zsz3a
    use c:\project\stock.dbf
    LOCATE ALL FOR UPPER(ALLTRIM(Stcode))= UPPER(ALLTRIM(your_ch))
    IF FOUND() THEN
        delet
        do delet
    ELSE
        DO not
    ENDIF

```

```

IF NOT WEXIST("_Oyx0vilpn")
    DEFINE WINDOW _Oyx0vilpn ;
        AT 0.000, 0.000 ;
        SIZE 23.538,54.600 ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,192,192,192)
        MOVE WINDOW _Oyx0vilpn CENTER

```

```

ENDIF

```

```

* *****
*
*          CUSTOMER/Windows Screen Layout
*
* *****

```

```

#REGION 1
IF WVISIBLE("_Oyx0vilpn")
    ACTIVATE WINDOW _Oyx0vilpn SAME
ELSE
    ACTIVATE WINDOW _Oyx0vilpn NOSHOW
ENDIF

```

```

IF NOT WVISIBLE("_Oyx0vilpn")
    ACTIVATE WINDOW _Oyx0vilpn
ENDIF

```

```

READ CYCLE

```

```

PROCEDURE BT
    PARAMETER m.btnname
    DO CASE
CASE m.btnname='add'
    Do c:\project\cust.prg
CASE m.btnname='SEARCH'
    DO c:\project\custsear.prg
CASE m.btnname='Delet'
    DO c:\project\custdele.prg
CASE m.btnname='LIST'
    DO c:\project\customer.qpr
CASE m.btnname='EXIT'
set sysm to defa
DEACTIVATE WINDOW _Oyx0vilpn
RELEASE WINDOW _Oyx0vilpn
ENDCASE

```

```

IF NOT WEXIST("_0yx15md6p")
    DEFINE WINDOW _0yx15md6p ;
        AT 0.000, 0.000 ;
        SIZE 22.077,52.800 ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        NOCLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,,0,128,128)
        MOVE WINDOW _0yx15md6p CENTER
ENDIF

```

```

*      *****
*      *
*      *      CUSTSEAR/Windows Screen Layout
*      *
*      *****
*

```

```

#REGION 1
IF WVISIBLE("_0yx15md6p")
    ACTIVATE WINDOW _0yx15md6p SAME
ELSE
    ACTIVATE WINDOW _0yx15md6p NOSHOW
ENDIF

```

```

IF NOT WVISIBLE("_0yx15md6p")
    ACTIVATE WINDOW _0yx15md6p
ENDIF

```

READ CYCLE

```

PROCEDURE BT
    PARAMETER m.btnname
do case
CASE m.btnname='EXIT'
set sysm to defa
DEACTIVATE WINDOW _0yx15md6p
RELEASE WINDOW _0yx15md6p
ENDCASE
RETURN

```

```

*****
procedure brosedata
DEFINE WINDOW aal ;
    AT 0.000, 0.000 ;
    SIZE 37.075,120.333 ;
    FONT "MS Sans Serif", 8 ;

```

```

        STYLE "B" ;
        FLOAT ;
        CLOSE ;
    MINIMIZE
        MOVE WINDOW aa1 CENTER
        activate window aa1
    BROWSE WINDOW AA1
    deactivate window aa1
    return
*****PROCEDURE WHEN THE CHOICE FOUND*****
PROCEDURE FIND
IF NOT WEXIST("_03910y9u2")
    DEFINE WINDOW _03910y9u2 ;
        AT 0.000, 0.000 ;
        SIZE 9.231,55.000 ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,,255,255,255)
        MOVE WINDOW _03910y9u2 CENTER
ENDIF

IF WVISIBLE("_03910y9u2")
    ACTIVATE WINDOW _03910y9u2 SAME
ELSE
    ACTIVATE WINDOW 03910y9u2 NOSHOW
ENDIF
@ 1.769,14.400 SAY "Searching It's Ok" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
@ 5.231,2.200 SAY "Press The Enter Key To Continue !" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
    IF NOT WVISIBLE("_03910y9u2")
ACTIVATE WINDOW _03910y9u2
    ENDIF
read
release window _03910y9u2
do brosedata
return
*****PROCEDURE WHEN THE CHOICE NOT FOUND*****
PROCEDURE NOT
IF NOT WEXIST("_03910y9u2")
    DEFINE WINDOW _03910y9u2 ;
        AT 0.000, 0.000 ;
        SIZE 9.231,55.000 ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;

```



```

        SYSTEM ;
        COLOR RGB(,,,255,255,255)
        MOVE WINDOW _03910y9u2 CENTER
ENDIF

IF WVISIBLE("_03910y9u2")
    ACTIVATE WINDOW _03910y9u2 SAME
ELSE
    ACTIVATE WINDOW _03910y9u2 NOSHOW
ENDIF
@ 1.769,14.400 SAY "It's Not Found" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
@ 5.231,2.200 SAY "Press The Enter Key To Continue !" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
    IF NOT WVISIBLE("_03910y9u2")
        ACTIVATE WINDOW _03910y9u2
    ENDIF
read
release window _03910y9u2.

RETURN

*****      PROCEDURE SEARCHING BY NAME      *****
procedure NAME
    CLOSE DATABASE

    IF NOT WEXIST("_0390zsz3a")
        DEFINE WINDOW _0390zsz3a ;
            AT 0.000, 0.000 ;
            SIZE 6.538,70.200 ;
            FONT "MS Sans Serif", 8 ;
            FLOAT ;
            NOCLOSE ;
            MINIMIZE ;
            SYSTEM ;
            COLOR RGB(,,,255,255,255)
            MOVE WINDOW _0390zsz3a CENTER
    ENDIF
    IF WVISIBLE("_0390zsz3a")
        ACTIVATE WINDOW _0390zsz3a SAME
    ELSE
        ACTIVATE WINDOW _0390zsz3a NOSHOW
    ENDIF
@ 2.846,7.800 SAY "Enter Customer Name:" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
    IF NOT WVISIBLE("_0390zsz3a")
        ACTIVATE WINDOW _0390zsz3a
    ENDIF
    your_ch=space(10)

```

```

@2.999,40.800 get your_ch
read
  release window _0390zsz3a
  use c:\project\customer.dbf
  LOCATE ALL FOR UPPER(ALLTRIM(Cusname))= UPPER(ALLTRIM(your_ch))
)
IF FOUND() THEN
  do find
ELSE
  DO not
ENDIF
return
***** PROCEDURE SEARCHING BY NO*****
procedure NO
  CLOSE DATABASE
  IF NOT WEXIST("_0390zsz3a")
  DEFINE WINDOW _0390zsz3a ;
    AT 0.000, 0.000 ;
    SIZE 6.538,70.200 ;
    FONT "MS Sans Serif", 8 ;
    FLOAT ;
    NOCLOSE ;
    MINIMIZE ;
    SYSTEM ;
    COLOR RGB(,,,255,255,255)
    MOVE WINDOW _0390zsz3a CENTER
  ENDIF
  IF WVISIBLE("_0390zsz3a")
    ACTIVATE WINDOW _0390zsz3a SAME
  ELSE
    ACTIVATE WINDOW _0390zsz3a NOSHOW
  ENDIF
  @ 2.846,7.800 SAY "Enter customer No: " ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
  IF NOT WVISIBLE("_0390zsz3a")
  ACTIVATE WINDOW _0390zsz3a
  ENDIF
  your_ch=space(10)
  @2.999,34.800 get your_ch
  read
  release window _0390zsz3a
    use c:\project\customer.dbf
  LOCATE ALL FOR UPPER(ALLTRIM(cusno))= UPPER(ALLTRIM(your_ch))
  IF FOUND() THEN
    DO find
  ELSE
    DO not
  ENDIF

```

```

IF NOT WEXIST("_0yy0yk2tf")
    DEFINE WINDOW _0yy0yk2tf ;
        AT 0.000, 0.000 ;
        SIZE 22.077,52.800 ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        NOCLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,,0,0,255)
        MOVE WINDOW _0yy0yk2tf CENTER
ENDIF

```

```

* *****
*
*
*          CUSTDELE/Windows Screen Layout
*
* *****
*

```

```

#REGION 1
IF WVISIBLE("_0yy0yk2tf")
    ACTIVATE WINDOW _0yy0yk2tf SAME
ELSE
    ACTIVATE WINDOW _0yy0yk2tf NOSHOWN
ENDIF

```

```

IF NOT WVISIBLE("_0yy0yk2tf")
    ACTIVATE WINDOW _0yy0yk2tf
ENDIF

```

```

READ CYCLE
PROCEDURE BT
    PARAMETER m.btnname
do case
CASE m.btnname='EXIT'
set sysm to defa
DEACTIVATE WINDOW _0yy0yk2tf
RELEASE WINDOW _0yy0yk2tf
ENDCASE
RETURN

```

```

*****PROCEDURE WHEN THE CHOICE FOUND*****
PROCEDURE DELET
IF NOT WEXIST("_03910y9u2")
    DEFINE WINDOW _03910y9u2 ;
        AT 0.000, 0.000 ;
        SIZE 9.231,55.000 ;

```



```

        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,,255,255,255)
        MOVE WINDOW _03910y9u2 CENTER
ENDIF

IF WVISIBLE("_03910y9u2")
        ACTIVATE WINDOW _03910y9u2 SAME
ELSE
        ACTIVATE WINDOW _03910y9u2 NOSHOWN
ENDIF
@ 1.769,14.400 SAY "Deleting It's Ok" ;
        FONT "MS Sans Serif", 10 ;
        STYLE "BT"
@ 5.231,2.200 SAY "Press The Enter Key To Continue !" ;
        FONT "MS Sans Serif", 10 ;
        STYLE "BT"
        IF NOT WVISIBLE("_03910y9u2")
ACTIVATE WINDOW _03910y9u2
        ENDIF
read
release window _03910y9u2
PACK
return
*****PROCEDURE WHEN THE CHOICE NOT FOUND*****
PROCEDURE NOT
IF NOT WEXIST("_03910y9u2")
        DEFINE WINDOW _03910y9u2 ;
                AT 0.000, 0.000 ;
                SIZE 9.231,55.000 ;
                FONT "MS Sans Serif", 8 ;
                FLOAT ;
                CLOSE ;
                MINIMIZE ;
                SYSTEM ;
                COLOR RGB(,,,255,255,255)
                MOVE WINDOW _03910y9u2 CENTER
ENDIF
IF WVISIBLE("_03910y9u2")
        ACTIVATE WINDOW _03910y9u2 SAME
ELSE
        ACTIVATE WINDOW _03910y9u2 NOSHOWN
ENDIF
@ 1.769,14.400 SAY "It's Not Found" ;
        FONT "MS Sans Serif", 10 ;
        STYLE "BT"
@ 5.231,2.200 SAY "Press The Enter Key To Continue !" ;
        FONT "MS Sans Serif", 10 ;

```



```

        STYLE "BT"
        IF NOT WVISIBLE("_03910y9u2")
ACTIVATE WINDOW _03910y9u2
        ENDIF
read
release window _03910y9u2.

```

```

RETURN

```

```

*****      PROCEDURE SEARCHING BY NAME      *****

```

```

procedure NAME

```

```

        CLOSE DATABASE
        IF NOT WEXIST("_0390zsz3a")
            DEFINE WINDOW _0390zsz3a ;
                AT 0.000, 0.000 ;
                SIZE 6.538,70.200 ;
                FONT "MS Sans Serif", 8 ;
                FLOAT ;
                NOCLOSE ;
                MINIMIZE ;
                SYSTEM ;
                COLOR RGB(,,,255,255,255)
                MOVE WINDOW _0390zsz3a CENTER

```

```

ENDIF

```

```

IF WVISIBLE("_0390zsz3a")
    ACTIVATE WINDOW _0390zsz3a SAME

```

```

ELSE
    ACTIVATE WINDOW _0390zsz3a NOSHOW

```

```

ENDIF

```

```

@ 2.846,7.800 SAY "Enter Customer Name:" ;

```

```

    FONT "MS Sans Serif", 10 ;

```

```

    STYLE "BT"

```

```

IF NOT WVISIBLE("_0390zsz3a")

```

```

ACTIVATE WINDOW _0390zsz3a

```

```

ENDIF

```

```

your_ch=space(10)

```

```

@2.999,40.800 get your_ch

```

```

read

```

```

release window _0390zsz3a

```

```

    use c:\project\customer.dbf

```

```

    LOCATE ALL FOR UPPER(ALLTRIM(Cusname))= UPPER(ALLTRIM(your_ch)

```

```

)

```

```

    IF FOUND() THEN

```

```

        DELET

```

```

        do DELET

```

```

    ELSE

```

```

        DO not

```

```

    ENDIF

```

```

return

```

```

*****      PROCEDURE SEARCHING BY NO      *****

```

```

procedure NO

```

```

    CLOSE DATABASE

```

```

IF NOT WEXIST("_0390zsz3a")
DEFINE WINDOW _0390zsz3a ;
    AT 0.000, 0.000 ;
    SIZE 6.538,70.200 ;
    FONT "MS Sans Serif", 8 ;
    FLOAT ;
    NOCLOSE ;
    MINIMIZE ;
    SYSTEM ;
    COLOR RGB(,,,255,255,255)
    MOVE WINDOW _0390zsz3a CENTER
ENDIF
IF WVISIBLE("_0390zsz3a")
    ACTIVATE WINDOW _0390zsz3a SAME
ELSE
    ACTIVATE WINDOW _0390zsz3a NOSHOW
ENDIF
@ 2.846,7.800 SAY "Enter customer No: " ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
IF NOT WVISIBLE("_0390zsz3a")
ACTIVATE WINDOW _0390zsz3a
ENDIF
your_ch=space(10)
@2.999,34.800 get your_ch
read
release window _0390zsz3a
    use c:\project\customer.dbf
    LOCATE ALL FOR UPPER(ALLTRIM(cusno))= UPPER(ALLTRIM(your_ch))
IF FOUND() THEN
    DELET
    DO DELET
ELSE
    DO not
ENDIF

```



```

NOT WEXIST("_0yx167a86")
  DEFINE WINDOW _0yx167a86 ;
    AT 0.000, 0.000 ;
    SIZE 22.077,52.800 ;
    FONT "MS Sans Serif", 8 ;
    FLOAT ;
    NOCLOSE ;
    MINIMIZE ;
    SYSTEM ;
    COLOR RGB(,,,0,128,128)
    MOVE WINDOW _0yx167a86 CENTER

```

F

```

*****
*
*                SUPSEARC/Windows Screen Layout
*
*****

```

ION 1

```

VISIBLE("_0yx167a86")
  ACTIVATE WINDOW _0yx167a86 SAME

  ACTIVATE WINDOW _0yx167a86 NOSHOW

```

F

```

NOT WVISIBLE("_0yx167a86")
  ACTIVATE WINDOW _0yx167a86

```

IF

D CYCLE

EDURE BT

```

  PARAMETER m.btnname
case
  E m.btnname='EXIT'
    sysm to defa
  CTIVATE WINDOW _0yx167a86
  EASE WINDOW _0yx167a86
  ENDCASE

```

URN

```

*****
cedure brosedata
INE WINDOW aal ;
  AT 0.000, 0.000 ;

```



```

        SIZE 37.075,120.333 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B" ;
        FLOAT ;
        CLOSE ;
MINIMIZE
        MOVE WINDOW aal CENTER
        activate window aal
BROWSE WINDOW AA1
        deactivate window aal
        return
*****PROCEDURE WHEN THE CHOICE FOUND*****
PROCEDURE FIND
IF NOT WEXIST("_03910y9u2")
        DEFINE WINDOW _03910y9u2 ;
                AT 0.000, 0.000 ;
                SIZE 9.231,55.000 ;
                FONT "MS Sans Serif", 8 ;
                FLOAT ;
                CLOSE ;
                MINIMIZE ;
                SYSTEM ;
                COLOR RGB(,,,255,255,255)
                MOVE WINDOW _03910y9u2 CENTER
ENDIF

IF WVISIBLE("_03910y9u2")
        ACTIVATE WINDOW _03910y9u2 SAME
ELSE
        ACTIVATE WINDOW _03910y9u2 NOSHOW
ENDIF
@ 1.769,14.400 SAY "Searching It's Ok" ;
        FONT "MS Sans Serif", 10 ;
        STYLE "BT"
@ 5.231,2.200 SAY "Press The Enter Key To Continue !" ;
        FONT "MS Sans Serif", 10 ;
        STYLE "BT"
        IF NOT WVISIBLE("_03910y9u2")
ACTIVATE WINDOW _03910y9u2
        ENDIF
read
release window _03910y9u2
do brosedata
return
*****PROCEDURE WHEN THE CHOICE NOT FOUND*****
PROCEDURE NOT
IF NOT WEXIST("_03910y9u2")
        DEFINE WINDOW _03910y9u2 ;
                AT 0.000, 0.000 ;
                SIZE 9.231,55.000 ;
                FONT "MS Sans Serif", 8 ;
                FLOAT ;

```

```

CLOSE ;
MINIMIZE ;
SYSTEM ;
COLOR RGB(,,255,255,255)
MOVE WINDOW _03910y9u2 CENTER
ENDIF

```

```

IF WVISIBLE("_03910y9u2")
    ACTIVATE WINDOW _03910y9u2 SAME
ELSE
    ACTIVATE WINDOW _03910y9u2 NOSHOW
ENDIF

```

```

@ 1.769,14.400 SAY "It's Not Found" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
@ 5.231,2.200 SAY "Press The Enter Key To Continue !" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
    IF NOT WVISIBLE("_03910y9u2")
        ACTIVATE WINDOW _03910y9u2
    ENDIF
read
release window _03910y9u2.

```

RETURN

\*\*\*\*\* PROCEDURE SEARCHING BY NAME \*\*\*\*\*

procedure NAME

```

    CLOSE DATABASE
    IF NOT WEXIST("_0390zsz3a")
        DEFINE WINDOW _0390zsz3a ;
            AT 0.000, 0.000 ;
            SIZE 6.538,70.200 ;
            FONT "MS Sans Serif", 8 ;
            FLOAT ;
            NOCLOSE ;
            MINIMIZE ;
            SYSTEM ;
            COLOR RGB(,,255,255,255)
            MOVE WINDOW _0390zsz3a CENTER
    ENDIF

```

```

    IF WVISIBLE("_0390zsz3a")
        ACTIVATE WINDOW _0390zsz3a SAME
    ELSE
        ACTIVATE WINDOW _0390zsz3a NOSHOW
    ENDIF
    @ 2.846,7.800 SAY "Enter Supplier Name:" ;
        FONT "MS Sans Serif", 10 ;
        STYLE "BT"
    IF NOT WVISIBLE("_0390zsz3a")
        ACTIVATE WINDOW _0390zsz3a
    ENDIF

```

```

your_ch=space(10)
@2.999,40.800 get your_ch
read
release window _0390zsz3a
use c:\project\supplier.dbf
LOCATE ALL FOR UPPER(ALLTRIM(supname))= UPPER(ALLTRIM(your_ch))

)

IF FOUND() THEN
do find
ELSE
DO not
ENDIF

return
***** PROCEDURE SEARCHING BY NO*****
procedure NO
CLOSE DATABASE
IF NOT WEXIST("_0390zsz3a")
DEFINE WINDOW _0390zsz3a ;
    AT 0.000, 0.000 ;
    SIZE 6.538,70.200 ;
    FONT "MS Sans Serif", 8 ;
    FLOAT ;
    NOCLOSE ;
    MINIMIZE ;
    SYSTEM ;
    COLOR RGB(,,255,255,255)
    MOVE WINDOW _0390zsz3a CENTER
ENDIF
IF WVISIBLE("_0390zsz3a")
    ACTIVATE WINDOW _0390zsz3a SAME
ELSE
    ACTIVATE WINDOW _0390zsz3a NOSHOW
ENDIF
@ 2.846,7.800 SAY "Enter Supplier No: " ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
IF NOT WVISIBLE("_0390zsz3a")
ACTIVATE WINDOW _0390zsz3a
ENDIF
your_ch=space(10)
@2.999,34.800 get your_ch
read
release window _0390zsz3a
use c:\project\supplier.dbf
LOCATE ALL FOR UPPER(ALLTRIM(supno))= UPPER(ALLTRIM(your_ch))
IF FOUND() THEN
DO find
ELSE
DO not
ENDIF

```



```

IF NOT WEXIST("_0yy0y9r4i")
    DEFINE WINDOW _0yy0y9r4i ;
        AT 0.000, 0.000 ;
        SIZE 22.077,52.800 ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        NOCLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,,0,0,255)
        MOVE WINDOW _0yy0y9r4i CENTER
ENDIF

```

```

* *****
*
* SUPDELET/Windows Screen Layout
*
* *****
*

```

```

#REGION 1
IF WVISIBLE("_0yy0y9r4i")
    ACTIVATE WINDOW _0yy0y9r4i SAME
ELSE
    ACTIVATE WINDOW _0yy0y9r4i NOSHOWN
ENDIF

```

```

IF NOT WVISIBLE("_0yy0y9r4i")
    ACTIVATE WINDOW _0yy0y9r4i
ENDIF

```

READ CYCLE

```

PROCEDURE BT
    PARAMETER m.btnname
do case
CASE m.btnname='EXIT'
set sysm to defa
DEACTIVATE WINDOW _0yy0y9r4i
RELEASE WINDOW _0yy0y9r4i
ENDCASE
RETURN

```

```

*****PROCEDURE WHEN THE CHOICE FOUND*****
PROCEDURE DELET
IF NOT WEXIST("_03910y9u2")
    DEFINE WINDOW _03910y9u2 ;
        AT 0.000, 0.000 ;
        SIZE 9.231,55.000 ;

```



```

        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,,255,255,255)
        MOVE WINDOW _03910y9u2 CENTER
ENDIF

IF WVISIBLE("_03910y9u2")
    ACTIVATE WINDOW _03910y9u2 SAME
ELSE
    ACTIVATE WINDOW _03910y9u2 NOSHOWN
ENDIF
@ 1.769,14.400 SAY "Deteting It's Ok" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
@ 5.231,2.200 SAY "Press The Enter Key To Continue !" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
    IF NOT WVISIBLE("_03910y9u2")
        ACTIVATE WINDOW _03910y9u2
    ENDIF
read
release window _03910y9u2
BACK
return
*****PROCEDURE WHEN THE CHOICE NOT FOUND*****
PROCEDURE NOT
IF NOT WEXIST("_03910y9u2")
    DEFINE WINDOW _03910y9u2 ;
        AT 0.000, 0.000 ;
        SIZE 9.231,55.000 ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,,255,255,255)
        MOVE WINDOW _03910y9u2 CENTER
    ENDIF
IF WVISIBLE("_03910y9u2")
    ACTIVATE WINDOW _03910y9u2 SAME
ELSE
    ACTIVATE WINDOW _03910y9u2 NOSHOWN
ENDIF
@ 1.769,14.400 SAY "It's Not Found" ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
@ 5.231,2.200 SAY "Press The Enter Key To Continue !" ;
    FONT "MS Sans Serif", 10 ;

```

```

        STYLE "BT"
        IF NOT WVISIBLE("_03910y9u2")
ACTIVATE WINDOW _03910y9u2
        ENDIF
read
release window _03910y9u2.

```

```

RETURN

```

```

*****      PROCEDURE SEARCHING BY NAME      *****

```

```

procedure NAME
        CLOSE DATABASE
        IF NOT WEXIST("_0390zsz3a")
            DEFINE WINDOW _0390zsz3a ;
                AT 0.000, 0.000 ;
                SIZE 6.538,70.200 ;
                FONT "MS Sans Serif", 8 ;
                FLOAT ;
                NOCLOSE ;
                MINIMIZE ;
                SYSTEM ;
                COLOR RGB(,,255,255,255)
                MOVE WINDOW _0390zsz3a CENTER
        ENDIF
        IF WVISIBLE("_0390zsz3a")
            ACTIVATE WINDOW _0390zsz3a SAME
        ELSE
            ACTIVATE WINDOW _0390zsz3a NOSHOW
        ENDIF
@ 2.846,7.800 SAY "Enter Supplier Name:" ;
        FONT "MS Sans Serif", 10 ;
        STYLE "BT"
        IF NOT WVISIBLE("_0390zsz3a")
ACTIVATE WINDOW _0390zsz3a
        ENDIF

```

```

        your_ch=space(10)
        @2.999,40.800 get your_ch
        read
        release window _0390zsz3a
        use c:\project\supplier.dbf
        LOCATE ALL FOR UPPER(ALLTRIM(supname))= UPPER(ALLTRIM(your_ch))

```

```

        IF FOUND() THEN
            DELET
            do DELET
        ELSE
            DO not
        ENDIF

```

```

return

```

```

*****      PROCEDURE SEARCHING BY NO*****

```

```

procedure NO
        CLOSE DATABASE

```

```

IF NOT WEXIST("_0390zsz3a")
DEFINE WINDOW _0390zsz3a ;
    AT 0.000, 0.000 ;
    SIZE 6.538,70.200 ;
    FONT "MS Sans Serif", 8 ;
    FLOAT ;
    NOCLOSE ;
    MINIMIZE ;
    SYSTEM ;
    COLOR RGB(,,,255,255,255)
    MOVE WINDOW _0390zsz3a CENTER
ENDIF
IF WVISIBLE("_0390zsz3a")
    ACTIVATE WINDOW _0390zsz3a SAME
ELSE
    ACTIVATE WINDOW _0390zsz3a NOSHOWN
ENDIF
@ 2.846,7.800 SAY "Enter Supplier No: " ;
    FONT "MS Sans Serif", 10 ;
    STYLE "BT"
IF NOT WVISIBLE("_0390zsz3a")
ACTIVATE WINDOW _0390zsz3a
ENDIF
your_ch=space(10)
@2.999,34.800 get your_ch
read
release window _0390zsz3a
use c:\project\supplier.dbf
LOCATE ALL FOR UPPER(ALLTRIM(supno))= UPPER(ALLTRIM(your_ch))
IF FOUND() THEN
    DELET
    DO DELET
ELSE
    DO not
ENDIF

```



```
IF NOT WEXIST("_0yx0vwsx8")
    DEFINE WINDOW _0yx0vwsx8 ;
        AT 0.000, 0.000 ;
        SIZE 23.538,54.600 ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,192,192,192)
        MOVE WINDOW _0yx0vwsx8 CENTER
```

```
*****  
*  
*  
*          ACCOUNT/Windows Screen Layout  
*  
*****
```

```
#REGION 1
IF WVISIBLE("_0yx0vwsx8")
    ACTIVATE WINDOW _0yx0vwsx8 SAME
ELSE
    ACTIVATE WINDOW _0yx0vwsx8 NOSHOW
ENDIF
```

```
IF NOT WVISIBLE("_0yx0vwsx8")
    ACTIVATE WINDOW _0yx0vwsx8
ENDIF
```

## READ CYCLE

```

PROCEDURE BT
    PARAMETER m.btnname
    DO CASE
CASE m.btnname='debit'
    Do c:\project\deb.prg
CASE m.btnname='credit'
    DO c:\project\credit.prg
CASE m.btnname='balance'
    DO c:\project\balance.prg
CASE m.btnname='EXIT'
set sysm to defa
DEACTIVATE WINDOW _0yx0vwsx8
RELEASE WINDOW _0yx0vwsx8
    ENDCASE
RETURN

```



```

* *****
*
* 08/06/03 CREDIT.PRG 00:53:10
*
* *****
*
* Author's Name
*
* Copyright (c) 2003 Company Name
* Address
* City, Zip
*
* Description:
* This program was automatically generated by GENSCRN.
*
* *****

```

```

* *****
*
* CREDIT/Windows Setup Code - SECTION 1
*
* *****

```

```

#REGION 1
PRIVATE wzfields,wztalk
IF SET("TALK") = "ON"
    SET TALK OFF
    m.wztalk = "ON"
ELSE
    m.wztalk = "OFF"
ENDIF
m.wzfields=SET('FIELDS')
SET FIELDS OFF
IF m.wztalk = "ON"
    SET TALK ON
ENDIF

```

```

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat
IF SET("TALK") = "ON"
    SET TALK OFF
    m.talkstat = "ON"
ELSE
    m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

```
m.rborder = SET("READBORDER")
SET READBORDER ON
```

```
m.currarea = SELECT()
```

```
* *****
*
*      S9591120/Windows Databases, Indexes, Relations
*
* *****
*
```

```
IF USED("credit")
    SELECT credit
    SET ORDER TO TAG "_0z01h9cku"
ELSE
    SELECT 0
    USE (LOCFILE("c:\project\credit.dbf","DBF","Where is credit?"));
        AGAIN ALIAS credit ;
        ORDER TAG "_0z01h9cku"
ENDIF
```

```
* *****
*
*      Windows Window definitions
*
* *****
*
```

```
IF NOT WEXIST("_0z01hc8yb")
    DEFINE WINDOW _0z01hc8yb ;
        AT 0.000, 0.000 ;
        SIZE 13.692,83.667 ;
        TITLE "Credit" ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B" ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        COLOR RGB(,,,255,0,255)
    MOVE WINDOW _0z01hc8yb CENTER
ENDIF
```

```
* *****
*
*      CREDIT/Windows Setup Code - SECTION 2
*
*
```

```

*          *****
*
#REGION 1

#DEFINE C_DBFEMPTY      'Database is empty, add a record?'
#DEFINE C_EDITS          'Please finish your edits.'
#DEFINE C_TOPFILE        'Top of file.'
#DEFINE C_ENDFILE        'End of file.'
#DEFINE C_BRTITLE        'Locate Record'
#DEFINE C_NOLOCK         'Sorry, could not lock record -- t
ry again later.'
#DEFINE C_ECANCEL        'Edits Canceled.'
#DEFINE C_DELREC         'Delete selected record?'
#DEFINE C_NOFEAT         'Feature not available yet.'
#DEFINE C_NOWIZ          'Wizard application is not availab
le.'
#DEFINE C_MAKEREPO       'Creating report with Report Wizar
d.'
#DEFINE C_NOREPO         'Could not create report.'
#DEFINE C_DELNOTE        'Deleting records...'
#DEFINE C_READONLY       'Table is read-only. No editing al
lowed.'
#DEFINE C_NOTABLE        'No table selected. Open table or
run query.'
#DEFINE C_BADEXPR        'Invalid expression.'
#DEFINE C_LOCWIZ         'Locate WIZARD.APP:'
#DEFINE C_MULTITABLE     'You have multiple related tables. Adding
records in not allowed.'

MOVE WINDOW '_0z01hc8yb' CENTER
PRIVATE isediting,isadding,wztblarr
PRIVATE wzolddelete,wzolderror,wzoldesc
PRIVATE wzalias, tempcurs,wzlastrec
PRIVATE isreadonly,find_drop,is2table

IF EMPTY(ALIAS())
    WAIT WINDOW C_NOTABLE
    RETURN
ENDIF

wztblarr= ''
wzalias=SELECT()
isediting=.F.
isadding=.F.
is2table = .F.
wzolddelete=SET('DELETE')
SET DELETED ON
tempcurs=SYS(2015)  &&used if General field
wzlastrec = 1
wzolderror=ON('error')
ON ERROR DO wizerrorhandler

```



```
wzoldesc=ON('KEY','ESCAPE')
ON KEY LABEL ESCAPE
m.find_drop = IIF(_DOS,0,2)
```

```
m.isreadonly=IIF(ISREAD(),.T.,.F.)
IF m.isreadonly
    WAIT WINDOW C_READONLY TIMEOUT 1
ENDIF
```

```
IF RECCOUNT()=0 AND !m.isreadonly AND fox_alert(C_DBFEMPTY)
    APPEND BLANK
ENDIF
```

```
GOTO TOP
SCATTER MEMVAR MEMO
```

```
* *****
*
*          CREDIT/Windows Screen Layout
*
* *****
*
```

```
#REGION 1
IF WVISIBLE("_0z01hc8yb")
    ACTIVATE WINDOW _0z01hc8yb SAME
ELSE
    ACTIVATE WINDOW _0z01hc8yb NOSHOW
ENDIF
```

```
IF NOT WVISIBLE("_0z01hc8yb")
    ACTIVATE WINDOW _0z01hc8yb
ENDIF
```

```
* *****
*
*          WindowsREAD contains clauses from SCREEN s9591120
*
* *****
*
```

```
READ CYCLE ;
    ACTIVATE READACT() ;
    DEACTIVATE READDEAC() ;
    NOLOCK
```

```
RELEASE WINDOW _0z01hc8yb
```

```
* *****
```



Windows Closing Databases

```
*
*
*
*
* *****
*
*
IF USED("credit")
    SELECT credit
    USE
ENDIF

SELECT (m.currarea)

#REGION 0

SET READBORDER &rborder

IF m.talkstat = "ON"
    SET TALK ON
ENDIF
IF m.compstat = "ON"
    SET COMPATIBLE ON
ENDIF

*
* *****
*
*
* CREDIT/Windows Cleanup Code
*
* *****
*

#REGION 1
SET DELETED &wzolddelete
SET FIELDS &wzfields
ON ERROR &wzolderror
ON KEY LABEL ESCAPE &wzoldesc
DO CASE
CASE _DOS AND SET('DISPLAY')='VGA25'
    @24,0 CLEAR TO 24,79
CASE _DOS AND SET('DISPLAY')='VGA50'
    @49,0 CLEAR TO 49,79
CASE _DOS
    @24,0 CLEAR TO 24,79
ENDCASE

****Procedures****

*
* *****
*
```

```

*          *   CREDIT/Windows Supporting Procedures and Functions
*          *
*          *****
*
#REGION 1
PROCEDURE readdeac
  IF isediting
    ACTIVATE WINDOW '_Oz01hc8yb'
    WAIT WINDOW C_EDIT$ NOWAIT
  ENDIF
  IF !WVISIBLE(WOUTPUT())
    CLEAR READ
    RETURN .T.
  ENDIF
RETURN .F.

PROCEDURE readact
  IF !isediting
    SELECT (m.wzalias)
    SHOW GETS
  ENDIF
  DO REFRESH
RETURN

PROCEDURE wizerrorhandler
  * This very simple error handler is primarily intended
  * to trap for General field OLE errors which may occur
  * during editing from the MODIFY GENERAL window.
  WAIT WINDOW message()
RETURN

PROCEDURE printrec
  PRIVATE soldError,wizfname,saverec,savearea,tmpcurs,tmps
tr
  PRIVATE prnt_btn,p_recs,p_output,pr_out,pr_record
  STORE 1 TO p_recs,p_output
  STORE 0 TO prnt_btn
  STORE RECNO() TO saverec
  m.soldError=ON('error')
  DO pdialog
  IF m.prnt_btn = 2
    RETURN
  ENDIF
  IF !FILE(ALIAS()+'.FRX')
    m.wizfname=SYS(2004)+'WIZARDS\'+ 'WIZARD.APP'
    IF !FILE(m.wizfname)
      ON ERROR *
      m.wizfname=LOCFILE('WIZARD.APP','APP',C_LO
CWIZ)
      ON ERROR &soldError

```

```

        IF !'WIZARD.APP'$UPPER(m.wizfname)
        WAIT WINDOW C_NOWIZ
        RETURN
        ENDIF
    ENDIF
    WAIT WINDOW C_MAKEREPO NOWAIT
    m.savearea=SELECT()
    m.tmpcurs='_'+LEFT(SYS(3),7)
    CREATE CURSOR (m.tmpcurs) (comment m)
    m.tmpstr = '* LAYOUT = COLUMNAR'+CHR(13)+CHR(10)
    INSERT INTO (m.tmpcurs) VALUES(m.tmpstr)
    SELECT (m.savearea)
    DO (m.wizfname) WITH '', 'WZ QREPO', 'NOSCRN/CREATE'
    ,ALIAS(),m.tmpcurs
    USE IN (m.tmpcurs)
    WAIT CLEAR
    IF !FILE(ALIAS()+'.FRX') &&wizard could not creat
e report
        WAIT WINDOW C_NOREPO
        RETURN
        ENDIF
    ENDIF

    m.pr_out=IIF(m.p_output=1,'TO PRINT NOCONSOLE','PREVIEW'
)
    m.pr_record=IIF(m.p_recs=1,'NEXT 1','ALL')
    REPORT FORM (ALIAS()) &pr_out &pr_record
    GO m.saverec
RETURN

```

#### PROCEDURE BTN\_VAL

```

    PARAMETER m.btnname
    DO CASE
    CASE m.btnname='TOP'
        GO TOP
        WAIT WINDOW C_TOPFILE NOWAIT
    CASE m.btnname='PREV'
        IF !BOF()
            SKIP -1
        ENDIF
        IF BOF()
            WAIT WINDOW C_TOPFILE NOWAIT
            GO TOP
        ENDIF
    CASE m.btnname='NEXT'
        IF !EOF()
            SKIP 1
        ENDIF
        IF EOF()
            WAIT WINDOW C_ENDFILE NOWAIT
            GO BOTTOM
        ENDIF
    END CASE

```



```

ENDIF
CASE m.btnname='END'
    GO BOTTOM
    WAIT WINDOW C_ENDFILE NOWAIT
CASE m.btnname='LOCATE'
    DO loc_dlog
CASE m.btnname='ADD' AND !isediting &&add record
    isediting=.T.
    isadding=.T.
    =edithand('ADD')
    _curobj=1
    DO refresh
    SHOW GETS
    RETURN
CASE m.btnname='EDIT' AND !isediting &&edit record
    IF EOF() OR BOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        RETURN
    ENDIF
    IF RLOCK()
        isediting=.T.
        _curobj=1
        DO refresh
        RETURN
    ELSE
        WAIT WINDOW C_NOLOCK
    ENDIF
CASE m.btnname='EDIT' AND isediting &&save record
    IF isadding
        =edithand('SAVE')
    ELSE
        GATHER MEMVAR MEMO
    ENDIF
    UNLOCK
    isediting=.F.
    isadding=.F.
    DO refresh
CASE m.btnname='DELETE' AND isediting &&cancel record
    IF isadding
        =edithand('CANCEL')
    ENDIF
    isediting=.F.
    isadding=.F.
    UNLOCK
    WAIT WINDOW C_ECANCEL NOWAIT
    DO refresh
CASE m.btnname='DELETE'
    IF EOF() OR BOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        RETURN
    ENDIF
    IF fox_alert(C_DELREC)

```



```

        DELETE
        IF !EOF() AND DELETED()
            SKIP 1
        ENDIF
        IF EOF()
            WAIT WINDOW C__ENDFILE NOWAIT
            GO BOTTOM
        ENDIF
    ENDIF
    CASE m.btnname='PRINT'
        DO printrec
        RETURN
    CASE m.btnname='EXIT'
        m.bailout=.T.    &&this is needed if used with FoxA
        CLEAR READ
        RETURN
    ENDCASE
    SCATTER MEMVAR MEMO
    SHOW GETS
RETURN
pp

PROCEDURE REFRESH
DO CASE
CASE m.isreadonly AND RECCOUNT()=0
    SHOW GETS DISABLE
    SHOW GET exit_btn ENABLE
CASE m.isreadonly
    SHOW GET add_btn DISABLE
    SHOW GET del_btn DISABLE
    SHOW GET edit_btn DISABLE
CASE (RECCOUNT()=0 OR EOF()) AND !m.isediting
    SHOW GETS DISABLE
    SHOW GET add_btn ENABLE
    SHOW GET exit_btn ENABLE
CASE m.isediting
    SHOW GET find_drop DISABLE
    SHOW GET top_btn DISABLE
    SHOW GET prev_btn DISABLE
    SHOW GET loc_btn DISABLE
    SHOW GET next_btn DISABLE
    SHOW GET end_btn DISABLE
    SHOW GET add_btn DISABLE
    SHOW GET prnt_btn DISABLE
    SHOW GET exit_btn DISABLE
    SHOW GET edit_btn,1 PROMPT "\<Save"
    SHOW GET del_btn,1 PROMPT "\<Cancel"
    ON KEY LABEL ESCAPE DO BTN_VAL WITH 'DELETE'
    RETURN
OTHERWISE
    SHOW GET edit_btn,1 PROMPT "Ed\<it"

```

```

SHOW GET del_btn,1 PROMPT "\<Delete"
SHOW GETS ENABLE
ENDCASE
F m.is2table
SHOW GET add_btn DISABLE
ENDIF
N KEY LABEL ESCAPE
URN

CEDURE edithand
PARAMETER m.paction
* procedure handles edits
DO CASE
CASE m.paction = 'ADD'
SCATTER MEMVAR MEMO BLANK
CASE m.paction = 'SAVE'
INSERT INTO (ALIAS()) FROM MEMVAR
CASE m.paction = 'CANCEL'
* nothing here
ENDCASE
URN

CEDURE fox_alert
PARAMETER wzalrtmess
PRIVATE alrtbtn
m.alrtbtn=2
DEFINE WINDOW _qeclij2t7 AT 0,0 SIZE 8,50 ;
FONT "MS Sans Serif",10 STYLE 'B' ;
FLOAT NOCLOSE NOMINIMIZE DOUBLE TITLE WTITLE()
MOVE WINDOW _qeclij2t7 CENTER
ACTIVATE WINDOW _qeclij2t7 NOSHOW
@ 2,(50-txtwidth(wzalrtmess))/2 SAY wzalrtmess;
FONT "MS Sans Serif", 10 STYLE "B"
@ 6,18 GET m.alrtbtn ;
PICTURE "@*HT \<OK; \? \! \<Cancel" ;
SIZE 1.769,8.667,1.333 ;
FONT "MS Sans Serif", 8 STYLE "B"
ACTIVATE WINDOW _qeclij2t7
READ CYCLE MODAL
RELEASE WINDOW _qeclij2t7
TURN m.alrtbtn=1

CEDURE pdialog
DEFINE WINDOW _qjn12zbvh ;
AT 0.000, 0.000 ;
SIZE 13.231,54.800 ;
TITLE "Microsoft FoxPro" ;
FONT "MS Sans Serif", 8 ;
FLOAT NOCLOSE MINIMIZE SYSTEM
MOVE WINDOW _qjn12zbvh CENTER

```

ACTIVATE WINDOW \_qjn12zbvh NOSHOW

ACTIVATE WINDOW \_qjn12zbvh

READ CYCLE MODAL

RELEASE WINDOW \_qjn12zbvh

RETURN

PROCEDURE loc\_dlog

PRIVATE gfields,i

DEFINE WINDOW wzlocate FROM 1,1 TO 20,40;

SYSTEM GROW CLOSE ZOOM FLOAT FONT "MS Sans Serif",

MOVE WINDOW wzlocate CENTER

m.gfields=SET('FIELDS',2)

IF !EMPTY(RELATION(1))

SET FIELDS ON

IF m.gfields # 'GLOBAL'

SET FIELDS GLOBAL

ENDIF

IF EMPTY(FLDLIST())

m.i=1

DO WHILE !EMPTY(OBJVAR(m.i))

IF ATC('M.',OBJVAR(m.i))=0

SET FIELDS TO (OBJVAR(m.i))

ENDIF

m.i = m.i + 1

ENDDO

ENDIF

ENDIF

BROWSE WINDOW wzlocate NOEDIT NODELETE ;

NOMENU TITLE C\_BRTITLE

SET FIELDS &gfields

SET FIELDS OFF

RELEASE WINDOW wzlocate

RETURN

```

*****
*
* 09/06/03                      DEB.PRG                      13:07:02
*
*****
*
* Author's Name
*
* Copyright (c) 2003 Company Name
* Address
* City,      Zip
*
* Description:
* This program was automatically generated by GENSCRN.
*
*****

*****
*
*                      DEB/Windows Setup Code - SECTION 1
*
*****

```

```

#REGION 1
PRIVATE wzfields,wztalk
IF SET("TALK") = "ON"
    SET TALK OFF
    m.wztalk = "ON"
ELSE
    m.wztalk = "OFF"
ENDIF
m.wzfields=SET('FIELDS')
SET FIELDS OFF
IF m.wztalk = "ON"
    SET TALK ON
ENDIF

```

```

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
    SET TALK OFF
    m.talkstat = "ON"
ELSE
    m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```



```
border = SET("READBORDER")
SET READBORDER ON
```

```
currarea = SELECT()
```

```
*****
*
*       S7222913/Windows Databases, Indexes, Relations
*
*****
```

```
IF USED("deb")
    SELECT deb
    SET ORDER TO TAG "_0z10s16r7"

ELSE
    SELECT 0
    USE (LOCFILE("c:\project\deb.dbf","DBF","Where is deb?"));
    AGAIN ALIAS deb ;
    ORDER TAG "_0z10s16r7"

ENDIF
```

```
*****
*
*       Windows Window definitions
*
*****
```

```
IF NOT WEXIST("_0z10s45fn")
    DEFINE WINDOW _0z10s45fn ;
        AT 0.000, 0.000 ;
        SIZE 11.308,83.333 ;
        TITLE "Deb" ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B" ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        COLOR RGB(,,,255,0,255)
    MOVE WINDOW _0z10s45fn CENTER

ENDIF
```

```
*****
*
*       DEB/Windows Setup Code - SECTION 2
*
*****
```

\*

#REGION 1

```
#DEFINE C_DBFEMPTY      'Database is empty, add a record?'
#DEFINE C_EDITS          'Please finish your edits.'
#DEFINE C_TOPFILE        'Top of file.'
#DEFINE C_ENDFILE        'End of file.'
#DEFINE C_BRTITLE        'Locate Record'
#DEFINE C_NOLOCK         'Sorry, could not lock record -- t
ry again later.'
#DEFINE C_ECANCEL        'Edits Canceled.'
#DEFINE C_DELREC         'Delete selected record?'
#DEFINE C_NOFEAT         'Feature not available yet.'
#DEFINE C_NOWIZ          'Wizard application is not availab
le.'
#DEFINE C_MAKEREPO       'Creating report with Report Wizar
d.'
#DEFINE C_NOREPO         'Could not create report.'
#DEFINE C_DELNOTE        'Deleting records...'
#DEFINE C_READONLY       'Table is read-only. No editing al
lowed.'
#DEFINE C_NOTABLE        'No table selected. Open table or
run query.'
#DEFINE C_BADEXP         'Invalid expression.'
#DEFINE C_LOCWIZ         'Locate WIZARD.APP:'
#DEFINE C_MULTITABLE     'You have multiple related tables. Adding
records in not allowed.'
```

```
MOVE WINDOW '_0z10s45fn' CENTER
PRIVATE isediting,isadding,wztblarr
PRIVATE wzolddelete,wzolderror,wzoldesc
PRIVATE wzalias, tempcurs,wzlastrec
PRIVATE isreadonly,find_drop,is2table
```

```
IF EMPTY(ALIAS())
    WAIT WINDOW C_NOTABLE
    RETURN
ENDIF
```

```
m.wztblarr= ''
m.wzalias=SELECT()
m.isediting=.F.
m.isadding=.F.
m.is2table = .F.
m.wzolddelete=SET('DELETE')
SET DELETED ON
m.tempcurs=SYS(2015)  &&used if General field
m.wzlastrec = 1
m.wzolderror=ON('error')
ON ERROR DO wizerrorhandler
wzoldesc=ON('KEY','ESCAPE')
```

```
ON KEY LABEL ESCAPE
m.find_drop = IIF(_DOS,0,2)
```

```
m.find_drop = IIF(_DOS, 0, 2)
```

```
m.isreadonly=IIF(ISREAD(),.T.,.F.)
```

IF m.isreadonly

WAIT WINDOW C\_READONLY TIMEOUT 1

ENDIF

```
IF RECCOUNT()=0 AND !m.isreadonly AND fox_alert(C_DBFEMPTY)
    APPEND BLANK
```

APPEND BLANK

ENDIF

GOTO TOP

SCATTER MEMVAR MEMO

```
*****  
*  
*  
*      DEB/Windows Screen Layout  
*  
*****
```

#REGION 1

```
IF WVISIBLE (" 0z10s45fn")
```

ACTIVATE WINDOW 0z10s45fn SAME

ELSE

ACTIVATE WINDOW \_0z10s45fn NOSHOWN

ENDIF

```
IF NOT WVISIBLE(" 0z10s45fn")
```

ACTIVATE WINDOW 0z10s45fn

ENDIF

\*\*\*\*\*  
\*  
\*  
\* WindowsREAD contains clauses from SCREEN s7222913  
\*  
\*\*\*\*\*

READ CYCLE ;

ACTIVATE READACT ( ) ;

```
DEACTIVATE READDEAC() ;
```

NOLOCK

RELEASE WINDOW 0z10s45fn

\*\*\*\*\*



```

*                               Windows Closing Databases
*
*****

IF USED("deb")
    SELECT deb
    USE
ENDIF

SELECT (m.currarea)

#REGION 0

SET READBORDER &rborder

IF m.talkstat = "ON"
    SET TALK ON
ENDIF
IF m.compstat = "ON"
    SET COMPATIBLE ON
ENDIF

*****
*
*                               DEB/Windows Cleanup Code
*
*****

#REGION 1
SET DELETED &wzolddelete
SET FIELDS &wzfields
ON ERROR &wzolderror
ON KEY LABEL ESCAPE &wzoldesc
DO CASE
CASE _DOS AND SET('DISPLAY')='VGA25'
    @24,0 CLEAR TO 24,79
CASE _DOS AND SET('DISPLAY')='VGA50'
    @49,0 CLEAR TO 49,79
CASE _DOS
    @24,0 CLEAR TO 24,79
ENDCASE

****Procedures****

*****
*
*                               DEB/Windows Supporting Procedures and Functions

```



```

*
* *****
*

```

```

#REGION 1
PROCEDURE readdeac
  IF isediting
    ACTIVATE WINDOW '_Oz10s45fn'
    WAIT WINDOW C_EDITS NOWAIT
  ENDIF
  IF !WVISIBLE(WOUTPUT())
    CLEAR READ
    RETURN .T.
  ENDIF
RETURN .F.

PROCEDURE readact
  IF !isediting
    SELECT (m.wzalias)
    SHOW GETS
  ENDIF
  DO REFRESH
RETURN

PROCEDURE wizerrorhandler
  * This very simple error handler is primarily intended
  * to trap for General field OLE errors which may occur
  * during editing from the MODIFY GENERAL window.
  WAIT WINDOW message()
RETURN

PROCEDURE printrec
  PRIVATE sOldError,wizfname,saverec,savearea,tmpcurs,tmps
tr
  PRIVATE prnt_btn,p_recs,p_output,pr_out,pr_record
  STORE 1 TO p_recs,p_output
  STORE 0 TO prnt_btn
  STORE RECNO() TO saverec
  m.sOldError=ON('error')
  DO pdialog
  IF m.prnt_btn = 2
    RETURN
  ENDIF
  IF !FILE(ALIAS()+'.FRX')
    m.wizfname=SYS(2004)+'WIZARDS\'+ 'WIZARD.APP'
    IF !FILE(m.wizfname)
      ON ERROR *
      m.wizfname=LOCFILE('WIZARD.APP','APP',C_LO
CWIZ)

      ON ERROR &sOldError
      IF !'WIZARD.APP'$UPPER(m.wizfname)

```

```

        WAIT WINDOW C_NOWIZ
        RETURN
    ENDIF
ENDIF
WAIT WINDOW C_MAKEREPO NOWAIT
    m.savearea=SELECT()
    m.tmpcurs=' '+LEFT(SYS(3),7)
    CREATE CURSOR (m.tmpcurs) (comment m)
    m.tmpstr = '* LAYOUT = COLUMNAR'+CHR(13)+CHR(10)
    INSERT INTO (m.tmpcurs) VALUES(m.tmpstr)
    SELECT (m.savearea)
    DO (m.wizfname) WITH '', 'WZ_QREPO', 'NOSCRN/CREATE'
,ALIAS(),m.tmpcurs
    USE IN (m.tmpcurs)
WAIT CLEAR
    IF !FILE(ALIAS()+'.FRX') &&wizard could not creat
e report
        WAIT WINDOW C_NOREPO
        RETURN
    ENDIF
ENDIF

    m.pr_out=IIF(m.p_output=1,'TO PRINT NOCONSOLE','PREVIEW'
)

    m.pr_record=IIF(m.p_recs=1,'NEXT 1','ALL')
    REPORT FORM (ALIAS()) &pr_out &pr_record
    GO m.saverec

RETURN

```

```

PROCEDURE BTN_VAL
    PARAMETER m.btnname
    DO CASE
    CASE m.btnname='TOP'
        GO TOP
        WAIT WINDOW C_TOPFILE NOWAIT
    CASE m.btnname='PREV'
        IF !BOF()
            SKIP -1
        ENDIF
        IF BOF()
            WAIT WINDOW C_TOPFILE NOWAIT
            GO TOP
        ENDIF
    CASE m.btnname='NEXT'
        IF !EOF()
            SKIP 1
        ENDIF
        IF EOF()
            WAIT WINDOW C_ENDFILE NOWAIT
            GO BOTTOM
        ENDIF
    ENDIF

```

```

CASE m.btnname='END'
    GO BOTTOM
    WAIT WINDOW C_ENDFILE NOWAIT
CASE m.btnname='LOCATE'
    DO loc_dlog
CASE m.btnname='ADD' AND !isediting &&add record
    isediting=.T.
    isadding=.T.
    =edithand('ADD')
    _curobj=1
    DO refresh
    SHOW GETS
    RETURN
CASE m.btnname='EDIT' AND !isediting &&edit record
    IF EOF() OR BOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        RETURN
    ENDIF
    IF RLOCK()
        isediting=.T.
        _curobj=1
        DO refresh
        RETURN
    ELSE
        WAIT WINDOW C_NOLOCK
    ENDIF
CASE m.btnname='EDIT' AND isediting &&save record
    IF isadding
        =edithand('SAVE')
    ELSE
        GATHER MEMVAR MEMO
    ENDIF
    UNLOCK
    isediting=.F.
    isadding=.F.
    DO refresh
CASE m.btnname='DELETE' AND isediting &&cancel record
    IF isadding
        =edithand('CANCEL')
    ENDIF
    isediting=.F.
    isadding=.F.
    UNLOCK
    WAIT WINDOW C_ECANCEL NOWAIT
    DO refresh
CASE m.btnname='DELETE'
    IF EOF() OR BOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        RETURN
    ENDIF
    IF fox_alert(C_DELREC)
        DELETE

```



```

        IF !EOF() AND DELETED()
            SKIP 1
        ENDIF
        IF EOF()
            WAIT WINDOW C__ENDFILE NOWAIT
            GO BOTTOM
        ENDIF
    ENDIF
CASE m.btnname='PRINT'
    DO printrec
    RETURN
CASE m.btnname='EXIT'
    m.bailout=.T.    &&this is needed if used with FoxA

    CLEAR READ
    RETURN
ENDCASE
SCATTER MEMVAR MEMO
SHOW GETS
TURN

PROCEDURE REFRESH
DO CASE
CASE m.isreadonly AND RECCOUNT()=0
    SHOW GETS DISABLE
    SHOW GET exit_btn ENABLE
CASE m.isreadonly
    SHOW GET add_btn DISABLE
    SHOW GET del_btn DISABLE
    SHOW GET edit_btn DISABLE
CASE (RECCOUNT()=0 OR EOF()) AND !m.isediting
    SHOW GETS DISABLE
    SHOW GET add_btn ENABLE
    SHOW GET exit_btn ENABLE
CASE m.isediting
    SHOW GET find_drop DISABLE
    SHOW GET top_btn DISABLE
    SHOW GET prev_btn DISABLE
    SHOW GET loc_btn DISABLE
    SHOW GET next_btn DISABLE
    SHOW GET end_btn DISABLE
    SHOW GET add_btn DISABLE
    SHOW GET prnt_btn DISABLE
    SHOW GET exit_btn DISABLE
    SHOW GET edit_btn,1 PROMPT "\<Save"
    SHOW GET del_btn,1 PROMPT "\<Cancel"
    ON KEY LABEL ESCAPE DO BTN_VAL WITH 'DELETE'
    RETURN
OTHERWISE
    SHOW GET edit_btn,1 PROMPT "Ed\<it"
    SHOW GET del_btn,1 PROMPT "\<Delete"

```



```

        SHOW GETS ENABLE
    ENDCASE
    IF m.is2table
        SHOW GET add_btn DISABLE
    ENDIF
    ON KEY LABEL ESCAPE
RETURN

```

```

PROCEDURE edithand
    PARAMETER m.paction
    * procedure handles edits
    DO CASE
    CASE m.paction = 'ADD'
        SCATTER MEMVAR MEMO BLANK
    CASE m.paction = 'SAVE'
        INSERT INTO (ALIAS()) FROM MEMVAR
    CASE m.paction = 'CANCEL'
        * nothing here
    ENDCASE
RETURN

```

```

PROCEDURE fox_alert
    PARAMETER wzalrtmess
    PRIVATE alrtbtn
    m.alrtbtn=2
    DEFINE WINDOW _qeclij2t7 AT 0,0 SIZE 8,50 ;
        FONT "MS Sans Serif",10 STYLE 'B' ;
        FLOAT NOCLOSE NOMINIMIZE DOUBLE TITLE WTITLE()
    MOVE WINDOW _qeclij2t7 CENTER
    ACTIVATE WINDOW _qeclij2t7 NOSHOW
    @ 2,(50-txtwidth(wzalrtmess))/2 SAY wzalrtmess;
        FONT "MS Sans Serif", 10 STYLE "B"
    @ 6,18 GET m.alrtbtn ;
        PICTURE "@*HT \<OK;\?\\<Cancel" ;
        SIZE 1.769,8.667,1.333 ;
        FONT "MS Sans Serif", 8 STYLE "B"
    ACTIVATE WINDOW _qeclij2t7
    READ CYCLE MODAL
    RELEASE WINDOW _qeclij2t7
RETURN m.alrtbtn=1

```

```

PROCEDURE pdialog
    DEFINE WINDOW _qjn12zbvh ;
        AT 0.000, 0.000 ;
        SIZE 13.231,54.800 ;
        TITLE "Microsoft FoxPro" ;
        FONT "MS Sans Serif", 8 ;
        FLOAT NOCLOSE MINIMIZE SYSTEM
    MOVE WINDOW _qjn12zbvh CENTER
    ACTIVATE WINDOW _qjn12zbvh NOSHOW

```

```
ACTIVATE WINDOW _qjn12zbvh
READ CYCLE MODAL
RELEASE WINDOW _qjn12zbvh
```

```
RETURN
```

```
PROCEDURE loc_dlog
```

```
PRIVATE gfields,i
```

```
DEFINE WINDOW wzlocate FROM 1,1 TO 20,40;
```

```
SYSTEM GROW CLOSE ZOOM FLOAT FONT "MS Sans Serif",
```

```
MOVE WINDOW wzlocate CENTER
```

```
m.gfields=SET('FIELDS',2)
```

```
IF !EMPTY(RELATION(1))
```

```
SET FIELDS ON
```

```
IF m.gfields # 'GLOBAL'
```

```
SET FIELDS GLOBAL
```

```
ENDIF
```

```
IF EMPTY(FLDLIST())
```

```
m.i=1
```

```
DO WHILE !EMPTY(OBJVAR(m.i))
```

```
IF ATC('M.',OBJVAR(m.i))=0
```

```
SET FIELDS TO (OBJVAR(m.i)
```

```
ENDIF
```

```
m.i = m.i + 1
```

```
ENDDO
```

```
ENDIF
```

```
ENDIF
```

```
BROWSE WINDOW wzlocate NOEDIT NODELETE ;
```

```
NOMENU TITLE C_BRTITLE
```

```
SET FIELDS &gfields
```

```
SET FIELDS OFF
```

```
RELEASE WINDOW wzlocate
```

```
RETURN
```

```

*****
*
*          BALANCE/Windows Setup Code - SECTION 1
*
*****

```

```

#REGION 1
PRIVATE wzfields,wztalk
IF SET("TALK") = "ON"
    SET TALK OFF
    m.wztalk = "ON"
ELSE
    m.wztalk = "OFF"
ENDIF
m.wzfields=SET('FIELDS')
SET FIELDS OFF
IF m.wztalk = "ON"
    SET TALK ON
ENDIF

```

```

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
    SET TALK OFF
    m.talkstat = "ON"
ELSE
    m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.rborder = SET("READBORDER")
SET READBORDER ON

m.currarea = SELECT()

```

```

*****
*
*          S3011840/Windows Databases, Indexes, Relations
*
*****

IF USED("balance")
    SELECT balance

```



```

* *****
*
*          BALANCE/Windows Setup Code - SECTION 1
*
* *****

```

```

#REGION 1
PRIVATE wzfields,wztalk
IF SET("TALK") = "ON"
    SET TALK OFF
    m.wztalk = "ON"
ELSE
    m.wztalk = "OFF"
ENDIF
m.wzfields=SET('FIELDS')
SET FIELDS OFF
IF m.wztalk = "ON"
    SET TALK ON
ENDIF

```

```

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat
IF SET("TALK") = "ON"
    SET TALK OFF
    m.talkstat = "ON"
ELSE
    m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

m.rborder = SET("READBORDER")
SET READBORDER ON

m.currarea = SELECT()

```

```

* *****
*
*          S3011840/Windows Databases, Indexes, Relations
*
* *****

```

```

IF USED("balance")
    SELECT balance

```



```

SET ORDER TO TAG "_0z0092zvu"
ELSE
    SELECT 0
    USE (LOCFILE("c:\project\balance.dbf","DBF","Where is bala
nce?"));
        AGAIN ALIAS balance ;
        ORDER TAG "_0z0092zvu"
ENDIF

```

```

* *****
*
* Windows Window definitions
*
* *****
*

```

```

IF NOT WEXIST("_0z10plw5e")
    DEFINE WINDOW _0z10plw5e ;
        AT 0.000, 0.000 ;
        SIZE 15.231,83.667 ;
        TITLE "Balance" ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B" ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        COLOR RGB(,,,255,0,255)
    MOVE WINDOW _0z10plw5e CENTER
ENDIF

```

```

* *****
*
* BALANCE/Windows Setup Code - SECTION 2
*
* *****
*

```

#REGION 1

#DEFINE C_DBFEMPTY	'Database is empty, add a record?'
#DEFINE C_EDITS	'Please finish your edits.'
#DEFINE C_TOPFILE	'Top of file.'
#DEFINE C_ENDFILE	'End of file.'
#DEFINE C_BRTITLE	'Locate Record'
#DEFINE C_NOLOCK	'Sorry, could not lock record -- t
ry again later.'	
#DEFINE C_ECANCEL	'Edits Canceled.'
#DEFINE C_DELREC	'Delete selected record?'
#DEFINE C_NOFEAT	'Feature not available yet.'
#DEFINE C_NOWIZ	'Wizard application is not availab

```

DEFINE C_MAKEREPO 'Creating report with Report Wizar
DEFINE C_NOREPO 'Could not create report.'
DEFINE C_DELNOTE 'Deleting records...'
DEFINE C_READONLY 'Table is read-only. No editing al
lowed.'
DEFINE C_NOTABLE 'No table selected. Open table or
run query.'
DEFINE C_BADEXP 'Invalid expression.'
DEFINE C_LOCWIZ 'Locate WIZARD.APP:'
DEFINE C_MULTITABLE 'You have multiple related tables. Adding
records in not allowed.'

MOVE WINDOW '_0z10plw5e' CENTER
PRIVATE isediting, isadding, wztblarr
PRIVATE wzolddelete, wzolderror, wzoldesc
PRIVATE wzalias, tempcurs, wzlastrec
PRIVATE isreadonly, find_drop, is2table

IF EMPTY(ALIAS())
    WAIT WINDOW C_NOTABLE
    RETURN
ENDIF

m.wztblarr= ''
m.wzalias=SELECT()
m.isediting=.F.
m.isadding=.F.
m.is2table = .F.
m.wzolddelete=SET('DELETE')
SET DELETED ON
m.tempcurs=SYS(2015) &&used if General field
m.wzlastrec = 1
m.wzolderror=ON('error')
ON ERROR DO wizerrorhandler
wzoldesc=ON('KEY','ESCAPE')
ON KEY LABEL ESCAPE
m.find_drop = IIF(_DOS,0,2)

m.isreadonly=IIF(ISREAD(),.T.,.F.)
IF m.isreadonly
    WAIT WINDOW C_READONLY TIMEOUT 1
ENDIF

IF RECCOUNT()=0 AND !m.isreadonly AND fox_alert(C_DBFEMPTY)
    APPEND BLANK
ENDIF

GOTO TOP
SCATTER MEMVAR MEMO

```

```

* *****
*
*                               BALANCE/Windows Screen Layout
*
* *****

```

```

#REGION 1
IF WVISIBLE("_0z10plw5e")
    ACTIVATE WINDOW _0z10plw5e SAME
ELSE
    ACTIVATE WINDOW _0z10plw5e NOSHOW
ENDIF

```

```

IF NOT WVISIBLE("_0z10plw5e")
    ACTIVATE WINDOW _0z10plw5e
ENDIF

```

```

* *****
*
*                               WindowsREAD contains clauses from SCREEN s3011840
*
* *****

```

```

READ CYCLE ;
    ACTIVATE READACT() ;
    DEACTIVATE READDEAC() ;
    NOLOCK

```

```

RELEASE WINDOW _0z10plw5e

```

```

* *****
*
*                               Windows Closing Databases
*
* *****

```

```

IF USED("balance")
    SELECT balance
    USE
ENDIF

```

```

SELECT (m.currarea)

```

```

#REGION 0

```



EADBORDER &rborder

talkstat = "ON"

SET TALK ON

compstat = "ON"

SET COMPATIBLE ON

```
*****
*
*      BALANCE/Windows Cleanup Code
*
*****
```

ION 1

DELETED &wzolddelete

FIELDS &wzfields

ERROR &wzolderror

KEY LABEL ESCAPE &wzoldesc

ASE

DOS AND SET('DISPLAY')='VGA25'

@24,0 CLEAR TO 24,79

DOS AND SET('DISPLAY')='VGA50'

@49,0 CLEAR TO 49,79

DOS

@24,0 CLEAR TO 24,79

ASE

Procedures\*\*\*\*

```
*****
*
*      BALANCE/Windows Supporting Procedures and Functions
*
*****
```

ION 1

EDURE readdeac

isediting

ACTIVATE WINDOW '\_0z10plw5e'

WAIT WINDOW C\_EDIT5 NOWAIT

NDIF

!WVISIBLE(WOUTPUT())

CLEAR READ

RETURN .T.

NDIF

URN .F.



```

PROCEDURE readact
  IF !isediting
    SELECT (m.wzalias)
    SHOW GETS
  ENDIF
  DO REFRESH
RETURN

PROCEDURE wizerrorhandler
  * This very simple error handler is primarily intended
  * to trap for General field OLE errors which may occur
  * during editing from the MODIFY GENERAL window.
  WAIT WINDOW message()
RETURN

PROCEDURE printrec
  PRIVATE sOldError,wizfname,saverec,savearea,tmpcurs,tmps
tr
  PRIVATE prnt_btn,p_recs,p_output,pr_out,pr_record
  STORE 1 TO p_recs,p_output
  STORE 0 TO prnt_btn
  STORE RECNO() TO saverec
  m.sOldError=ON('error')
  DO pdialog
  IF m.prnt_btn = 2
    RETURN
  ENDIF
  IF !FILE(ALIAS()+'.FRX')
    m.wizfname=SYS(2004)+'WIZARDS\'+ 'WIZARD.APP'
    IF !FILE(m.wizfname)
      ON ERROR *
      m.wizfname=LOCFILE('WIZARD.APP','APP',C_LO
CWIZ)
      ON ERROR &sOldError
      IF !'WIZARD.APP'$UPPER(m.wizfname)
      WAIT WINDOW C_NOWIZ
      RETURN
    ENDIF
  ENDIF
  WAIT WINDOW C_MAKEREPO NOWAIT
  m.savearea=SELECT()
  m.tmpcurs='_'+LEFT(SYS(3),7)
  CREATE CURSOR (m.tmpcurs) (comment m)
  m.tmpstr = '* LAYOUT = COLUMNAR'+CHR(13)+CHR(10)
  INSERT INTO (m.tmpcurs) VALUES(m.tmpstr)
  SELECT (m.savearea)
  DO (m.wizfname) WITH '', 'WZ_QREPO','NOSCRN/CREATE'
,ALIAS(),m.tmpcurs
  USE IN (m.tmpcurs)
  WAIT CLEAR

```

```

e report
        IF !FILE(ALIAS()+'.FRX') &&wizard could not creat
        WAIT WINDOW C_NOREPO
        RETURN
        ENDIF
    ENDIF

    m.pr_out=IIF(m.p_output=1,'TO PRINT NOCONSOLE','PREVIEW'
)
    m.pr_record=IIF(m.p_recs=1,'NEXT 1','ALL')
    REPORT FORM (ALIAS()) &pr_out &pr_record
    GO m.saverec
RETURN

```

```

PROCEDURE BTN_VAL
    PARAMETER m.btnname
    DO CASE
    CASE m.btnname='TOP'
        GO TOP
        WAIT WINDOW C_TOPFILE NOWAIT
    CASE m.btnname='PREV'
        IF !BOF()
            SKIP -1
        ENDIF
        IF BOF()
            WAIT WINDOW C_TOPFILE NOWAIT
            GO TOP
        ENDIF
    CASE m.btnname='NEXT'
        IF !EOF()
            SKIP 1
        ENDIF
        IF EOF()
            WAIT WINDOW C_ENDFILE NOWAIT
            GO BOTTOM
        ENDIF
    CASE m.btnname='END'
        GO BOTTOM
        WAIT WINDOW C_ENDFILE NOWAIT
    CASE m.btnname='LOCATE'
        DO loc_dlog
    CASE m.btnname='ADD' AND !isediting &&add record
        isediting=.T.
        isadding=.T.
        =edithand('ADD')
        _curobj=1
        DO refresh
        SHOW GETS
        RETURN
    CASE m.btnname='EDIT' AND !isediting &&edit record
        IF EOF() OR BOF()

```

```

        WAIT WINDOW C_ENDFILE NOWAIT
        RETURN
    ENDIF
    IF RLOCK()
        isediting=.T.
        _curobj=1
        DO refresh
        RETURN
    ELSE
        WAIT WINDOW C_NOLOCK
    ENDIF
CASE m.btnname='EDIT' AND isediting &&save record
    IF isadding
        =edithand('SAVE')
    ELSE
        GATHER MEMVAR MEMO
    ENDIF
    UNLOCK
    isediting=.F.
    isadding=.F.
    DO refresh
CASE m.btnname='DELETE' AND isediting &&cancel record
    IF isadding
        =edithand('CANCEL')
    ENDIF
    isediting=.F.
    isadding=.F.
    UNLOCK
    WAIT WINDOW C_ECANCEL NOWAIT
    DO refresh
CASE m.btnname='DELETE'
    IF EOF() OR BOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        RETURN
    ENDIF
    IF fox_alert(C_DELREC)
        DELETE
        IF !EOF() AND DELETED()
            SKIP 1
        ENDIF
        IF EOF()
            WAIT WINDOW C_ENDFILE NOWAIT
            GO BOTTOM
        ENDIF
    ENDIF
CASE m.btnname='PRINT'
    DO printrec
    RETURN
CASE m.btnname='EXIT'
    m.bailout=.T.    &&this is needed if used with FoxA

CLEAR READ

```



```

        RETURN
    ENDCASE
    SCATTER MEMVAR MEMO
    SHOW GETS
RETURN

PROCEDURE REFRESH
DO CASE
CASE m.isreadonly AND RECCOUNT()=0
    SHOW GETS DISABLE
    SHOW GET exit_btn ENABLE
CASE m.isreadonly
    SHOW GET add_btn DISABLE
    SHOW GET del_btn DISABLE
    SHOW GET edit_btn DISABLE
CASE (RECCOUNT()=0 OR EOF()) AND !m.isediting
    SHOW GETS DISABLE
    SHOW GET add_btn ENABLE
    SHOW GET exit_btn ENABLE
CASE m.isediting
    SHOW GET find_drop DISABLE
    SHOW GET top_btn DISABLE
    SHOW GET prev_btn DISABLE
    SHOW GET loc_btn DISABLE
    SHOW GET next_btn DISABLE
    SHOW GET end_btn DISABLE
    SHOW GET add_btn DISABLE
    SHOW GET prnt_btn DISABLE
    SHOW GET exit_btn DISABLE
    SHOW GET edit_btn,1 PROMPT "\<Save"
    SHOW GET del_btn,1 PROMPT "\<Delete"
    ON KEY LABEL ESCAPE DO BTN_VAL WITH 'DELETE'
    RETURN
OTHERWISE
    SHOW GET edit_btn,1 PROMPT "Ed\<it"
    SHOW GET del_btn,1 PROMPT "\<Delete"
    SHOW GETS ENABLE
ENDCASE
IF m.is2table
    SHOW GET add_btn DISABLE
ENDIF
ON KEY LABEL ESCAPE
RETURN

PROCEDURE edithand
    PARAMETER m.paction
    * procedure handles edits
    DO CASE
    CASE m.paction = 'ADD'
        SCATTER MEMVAR MEMO BLANK

```



```

CASE m.paction = 'SAVE'
    INSERT INTO (ALIAS()) FROM MEMVAR
CASE m.paction = 'CANCEL'
    * nothing here
ENDCASE

```

```

RETURN

```

```

PROCEDURE fox_alert
    PARAMETER wzalrtmess
    PRIVATE alrtbtn
    m.alrtbtn=2
    DEFINE WINDOW _qeclij2t7 AT 0,0 SIZE 8,50 ;
        FONT "MS Sans Serif",10 STYLE 'B' ;
        FLOAT NOCLOSE NOMINIMIZE DOUBLE TITLE WTITLE()
    MOVE WINDOW _qeclij2t7 CENTER
    ACTIVATE WINDOW _qeclij2t7 NOSHOW
    @ 2,(50-txtwidth(wzalrtmess))/2 SAY wzalrtmess;
        FONT "MS Sans Serif", 10 STYLE "B"
    @ 6,18 GET m.alrtbtn ;
        PICTURE "@*HT \<OK;\?\!\<Cancel" ;
        SIZE 1.769,8.667,1.333 ;
        FONT "MS Sans Serif", 8 STYLE "B"
    ACTIVATE WINDOW _qeclij2t7
    READ CYCLE MODAL
    RELEASE WINDOW _qeclij2t7
RETURN m.alrtbtn=1

```

```

PROCEDURE pdialog
    DEFINE WINDOW _qjnl2zbvh ;
        AT 0.000, 0.000 ;
        SIZE 13.231,54.800 ;
        TITLE "Microsoft FoxPro" ;
        FONT "MS Sans Serif", 8 ;
        FLOAT NOCLOSE MINIMIZE SYSTEM
    MOVE WINDOW _qjnl2zbvh CENTER
    ACTIVATE WINDOW _qjnl2zbvh NOSHOW
    @ 2.846,33.600 SAY "Output:" ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
    @ 2.846,4.800 SAY "Print:" ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
    @ 4.692,7.200 GET m.p_recs ;
        PICTURE "@*RVN \<Current Record;\<All Records" ;
        SIZE 1.308,18.500,0.308 ;
        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
    @ 4.692,36.000 GET m.p_output ;
        PICTURE "@*RVN \<Printer;Pre\<view" ;
        SIZE 1.308,12.000,0.308 ;

```

```

        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
@ 10.154,16.600 GET m.prnt_btn ;
        PICTURE "@*HT P\<r\int;Ca\<ncel" ;
        SIZE 1.769,8.667,0.667 ;
        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B"
        ACTIVATE WINDOW _qjn12zbvh
        READ CYCLE MODAL
        RELEASE WINDOW _qjn12zbvh
RETURN

```

```

PROCEDURE loc_dlog
PRIVATE gfields,i
DEFINE WINDOW wzlocate FROM 1,1 TO 20,40;
        SYSTEM GROW CLOSE ZOOM FLOAT FONT "MS Sans Serif",
8
        MOVE WINDOW wzlocate CENTER
        m.gfields=SET('FIELDS',2)
        IF !EMPTY(RELATION(1))
                SET FIELDS ON
                IF m.gfields # 'GLOBAL'
                        SET FIELDS GLOBAL
                ENDIF
                IF EMPTY(FLDLIST())
                        m.i=1
                        DO WHILE !EMPTY(OBJVAR(m.i))
                                IF ATC('M.',OBJVAR(m.i))=0
                                        SET FIELDS TO (OBJVAR(m.i))
                                ENDIF
                                m.i = m.i + 1
                        ENDDO
                ENDIF
        ENDIF
        BROWSE WINDOW wzlocate NOEDIT NODELETE ;
        NOMENU TITLE C_BRTITLE
        SET FIELDS &gfields
        SET FIELDS OFF
        RELEASE WINDOW wzlocate
RETURN

```

\*\*\*\*\*

procedure PURCH

```

        close database
        use c:\project\credit.dbf
        LOCATE ALL FOR UPPER(ALLTRIM(code))= UPPER(ALLTRIM(stcode))
        IF FOUND() THEN

```

```

    m.purch=stprice
USE c:\project\balance.dbf
ENDIF
RETURN
*****
procedure SEL

    close database
    use c:\project\deb.dbf
    LOCATE ALL FOR UPPER(ALLTRIM(code))= UPPER(ALLTRIM(stcode))
IF FOUND() THEN
    m.selling=price
    use c:\project\balance.dbf
ENDIF
RETURN
*****
procedure SUM

m.total = m.selling - m.purch

RETURN

```



## MAIN MENU



(Figure.1)



## STOCK MENU



(Figure 2)

## STOCK FILE

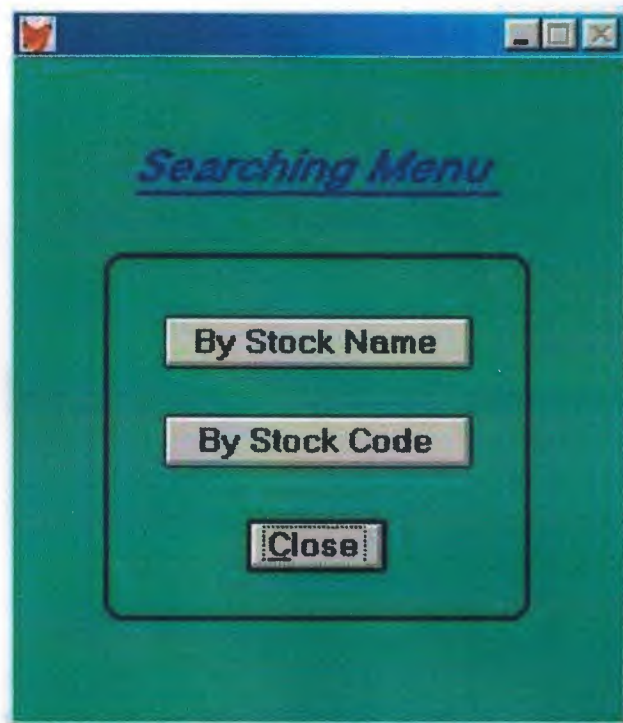
The image shows a software window titled 'Stock File'. It contains four input fields: 'Stock Name' with the value 'MOBILES', 'Quantity' with the value '24', 'Quality' with the value 'SWEDISH', and 'Stock Code' with the value 'STK01'. Below these fields is a row of nine buttons: 'Top', 'Prev', 'Next', 'End', 'Locate', 'Add', 'Edit', 'Delete', and 'Close'.

Stock Name:	MOBILES
Quantity:	24
Quality:	SWEDISH
Stock Code:	STK01

Top Prev Next End Locate Add Edit Delete Close

(Figure.3)

## SEARCHING MENU



(Figure. 4)

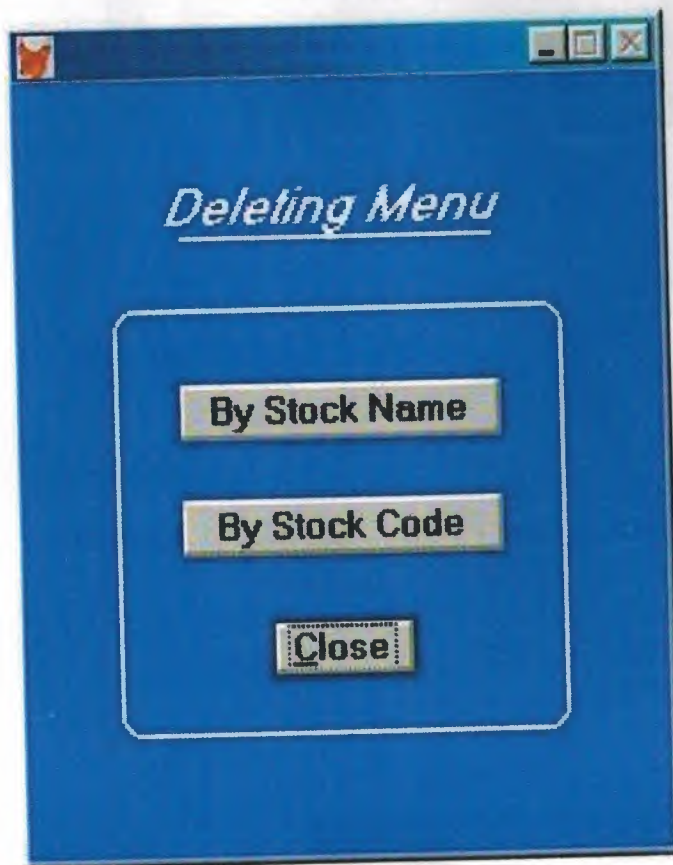
## STOCK QUERY

[illegible]

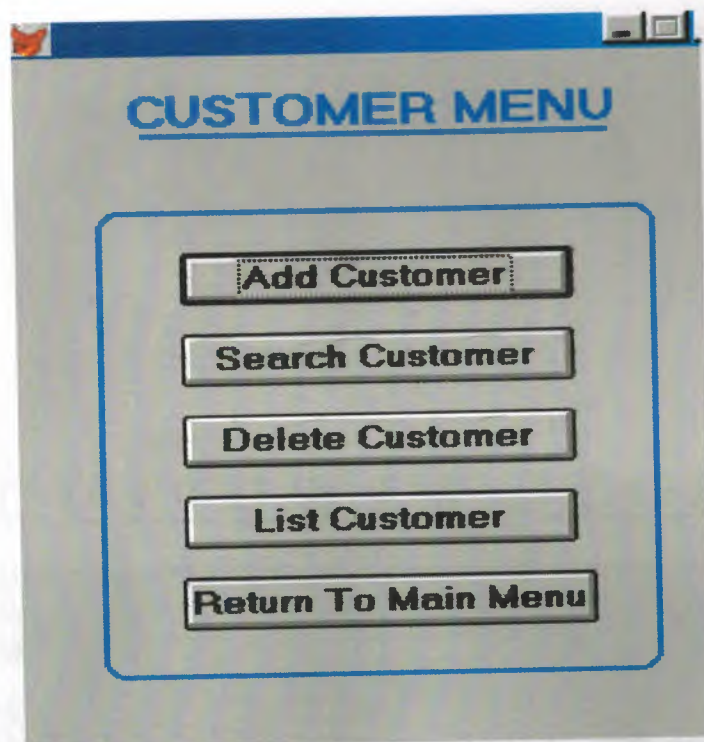
(Figure.5)



## DELETING MENU



(Figure . 6)



(Figure.7)

SEARCHING MENU

## CUSTOMER FILE

The screenshot shows a window titled 'Customer' with a sub-header 'Customer File'. It contains several text input fields with the following data: Customer Name: JAMES, Customer Surname: KALSS, Customer address: CYPRUS, Customer No: 101, Date: 10/03/1999, and Customer Tel No: 90542875421. At the bottom, there is a row of buttons: Top, Prev, Next, End, Locate, Add, Edit, Delete, Print, and Close.

Customer Name	JAMES
Customer Surname	KALSS
Customer address	CYPRUS
Customer No	101
Date	10/03/1999
Customer Tel No	90542875421

Top Prev Next End Locate Add Edit Delete Print Close

(Figure.8)

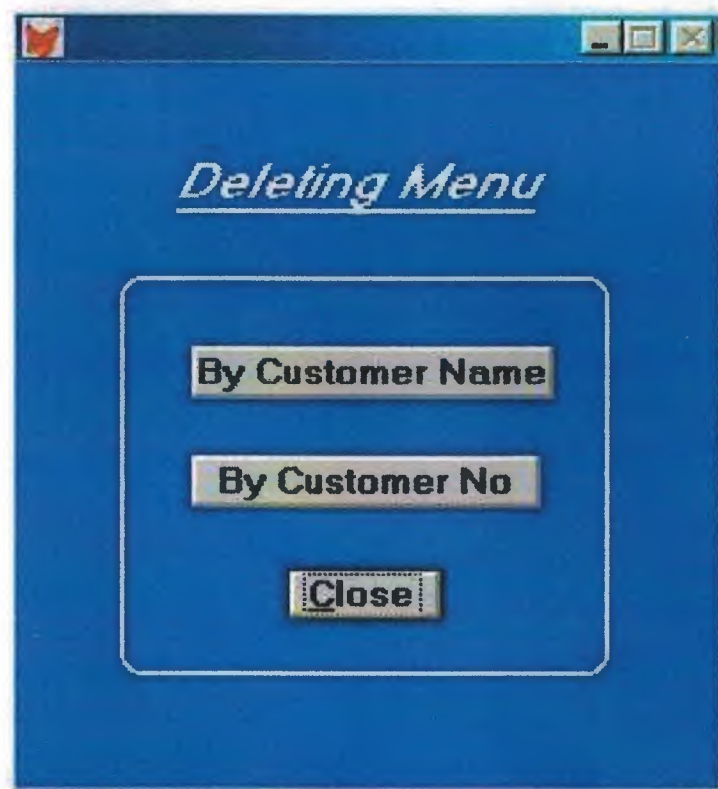
## SEARCHING MENU



(Figure.9)

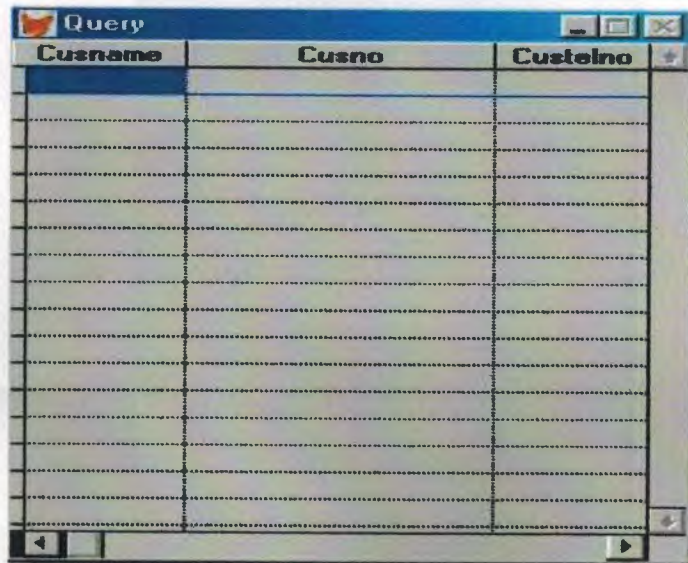


## DELETING MENU



(Figure.10)

## CUSTOMER QUERY



A screenshot of a software window titled "Query". The window displays a table with three columns: "Cusname", "Cusno", and "Custelno". The table is currently empty, showing only the header row. The window has a standard Windows-style title bar with minimize, maximize, and close buttons. At the bottom of the table, there are navigation arrows (back and forward) and a small status area.

Cusname	Cusno	Custelno
---------	-------	----------

(Figure.11)

## Supplier menu



(Figure.12)

## S U P P L I E R F I L E

Supplier

### Supplier File

Supplier Name:

Supplier Surname:

Supplier Address:

Supplier Nationality:

Supplier company:

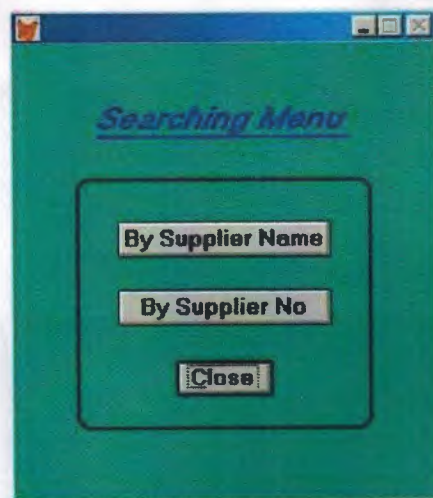
Supplier No:

Supplier Tel No:  0

(Figure.13)

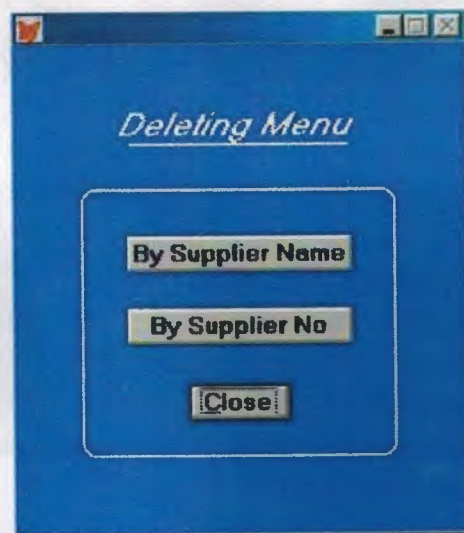


## SEARCHING MENU




(Figure.1u)

## DELETING MENU



(Figure.15)

## S U P P L I E R Q U E R Y



A screenshot of a software window titled "Query". The window displays a table with three columns: "Supname", "Supno", and "Suptelno". The table has a header row and approximately 20 empty data rows. The window includes standard Windows-style window controls (minimize, maximize, close) in the top right corner and scrollbars on the right and bottom.

Supname	Supno	Suptelno

(Figure.16)

## ACCOUNT MENU



(Figure.17)



# DEBIT FILE

CREDIT FILE

The image shows a software window titled "Debit File". At the top left of the window is a small icon and the text "Deb". The main title "Debit File" is centered at the top. Below the title, there are two input fields: "Stock Code:" followed by an empty text box, and "stock Price:" followed by a text box containing the number "0". At the bottom of the window, there is a horizontal row of nine buttons: "Top", "Prev", "Next", "End", "Locate", "Add", "Edit", "Delete", and "Close". Each button has a small icon to its left.

(Figure 18)

## CREDIT FILE

The screenshot shows a window titled 'Credit' with a sub-header 'Credit File'. It contains two input fields: 'Stock Code' with the value '001' and 'Stock Price' with the value '12'. Below these fields is a row of ten buttons: 'Top', 'Prev', 'Next', 'End', 'Locate', 'Add', 'Edit', 'Delete', 'Print', and 'Close'.

Stock Code	Stock Price
001	12

Top Prev Next End Locate Add Edit Delete Print Close

(Figure 119)

## B A L A N C E F I L E

The screenshot shows a window titled "Balance" with a sub-header "Balance File". Below the header are four input fields with labels: "Order", "Purchasing Price", "Selling Price", and "Net Total". The "Purchasing Price", "Selling Price", and "Net Total" fields contain the value "0". At the bottom of the window is a row of ten buttons: "Top", "Prev", "Next", "End", "Locate", "Add", "Edit", "Delete", "Print", and "Close".

Order

Purchasing Price
0

Selling Price
0

Net Total
0

Top	Prev	Next	End	Locate	Add	Edit	Delete	Print	Close
-----	------	------	-----	--------	-----	------	--------	-------	-------

(Figure 20)



[1] <http://www.guegue.net/~emmysoft/vfp.html>

[2] <http://www.computing.net/windows31/wwwboard/forum/9804.html>

[3] <http://forums.belution.com/en/foxpro>

[4] [emmysoft.ni/free/www.guegue.com](http://emmysoft.ni.free/www.guegue.com)<http://>