



NEAR EAST UNIVERSITY

**FACULTY OF ECONOMICS AND
ADMINISTRATIVE SCIENCE**

**DEPARTMENT OF COMPUTER
INFORMATION SYSTEMS**

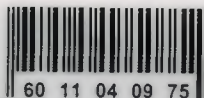
2003/2004 FALL TERM
CIS 400
(Graduation Project)

GLASSES SELLING SYSTEM

SUBMITTED TO : **Dr. Yalçın Akçalı**
Miss. Nadire Çavuş

SUBMITTED BY : **Naziyet Başkut(20011494)**

LEFKOŞA
2004



NEU

CONTENTS

PAGE

I.ACKNOWLEDGEMENTS.....	1
II. ABSTRACT.....	2
III. INTRODUCTION.....	3
IV. EXPLANATION OF THE SYSTEM.....	4
-Data Flow Diagram.....	4
-System Block Diagram.....	10
-Database Design.....	11
V. USERMANUAL.....	16
VI.LIST OF SUB PROGRAMS.....	17
-Program Flow Chart.....	17
-SourcProgram.....	32
-Screen Output.....	94
VII. REFERENCES.....	120

I. ACKNOWLEDGEMENTS

I want to thank Dr. Yalçın Akçalı , Miss Nadire Çavuş,Mr.Mustafa Menekay,Mr.İlham Hüseyinov,and my friend Vedat. Who are my teachers , and they helped my during making this projects.

I also would like to thank the people who create internet source of “delphiogreniyorum.com”.This site helped my project.

II. ABSTRACT

This project is for recording the sale and inventory of glasses. This project requires a lot of inputs through a user who passes these inputs through a process to ultimately retrieve or get an output according to his/her wish. The inputs are as follows: product code, product name, product size, product piece, buying place, buying price, selling price, buying date, customer name, customer list. These inputs are passed through various processes. These processes are as follows: Recording of entered customer in to the database, the reduction of the sold glasses from the inventory, the recording of the customer to whom the product has been sold (The process of entering the name and other information of the customer). The procedure and process regarding whether the customer's purchase has been cash or by installments. In the accounting procedures and process, on the other hand, the revenues and expenses of the company are calculated following the cost calculation. In the reporting phase, though we can see the main inventory report, cash sale report and installment report.

III. INTRODUCTION

As I am introducing this project for glasses inventory system , I would like to explain the necceary procedures followed in the programme .

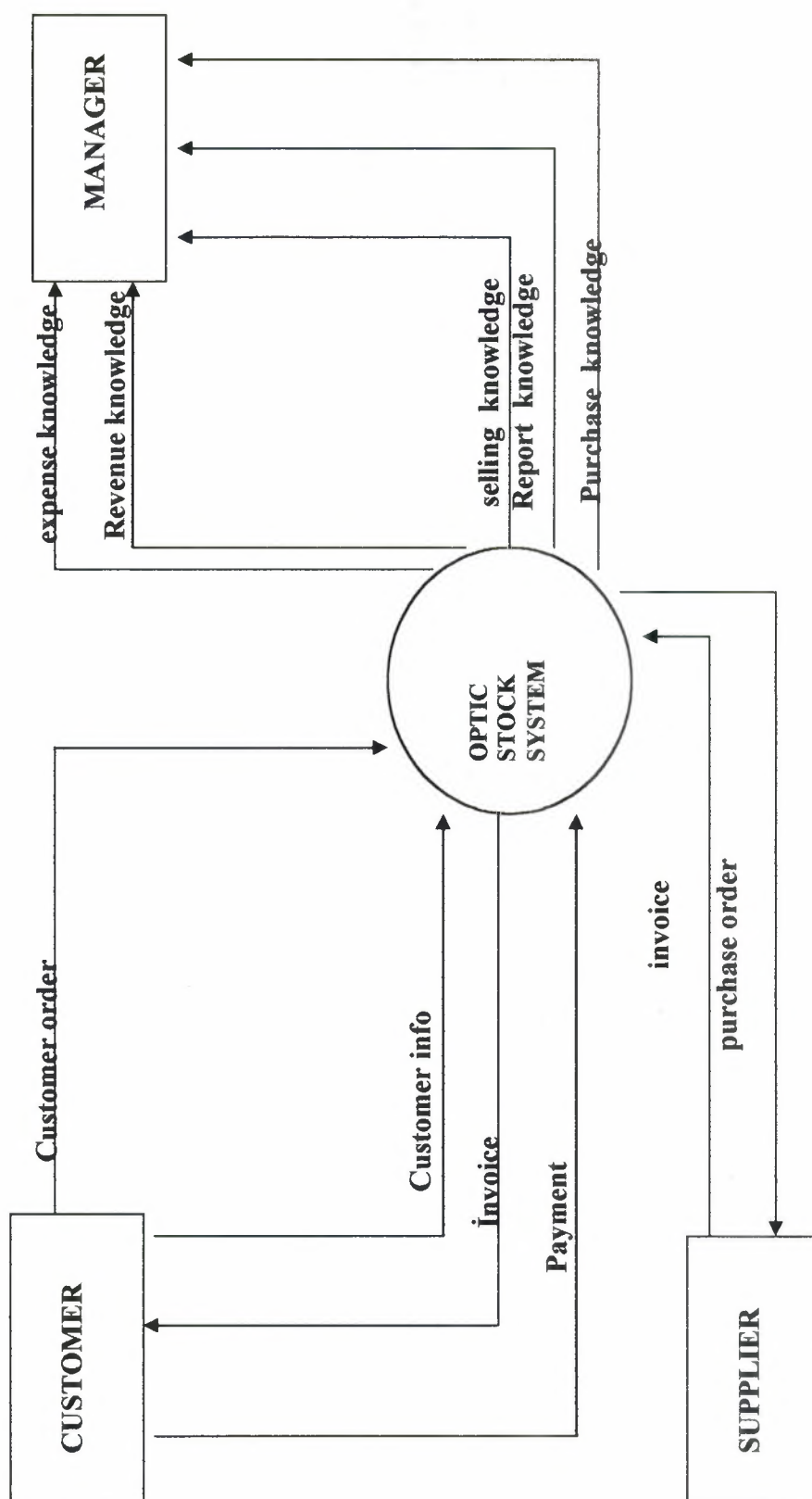
First we entered the product purchase for , then following the entry of the user input , the recording procedures of the purchase product in to the database is completed .

Depending upon the request of the user , the searching of the record product deleteing the existing record of any product can also be done . We ca complete these procedures though entering the product code variable . Then , we enter the sale form . Here , there exist two kinds of form being the cash and installment sale form . If the sale is in cash , then , we load on teh information related to the product and customer in to the database . If the sale is by installment , We load on the information of the product customers detail , again , but besides we also load on the information regarding the number of installment and amount of the installment. If I wish I can enter the list of the cash sale customer. If the sale is by installments , then I enter the counting number of the installment in to another received payment form .

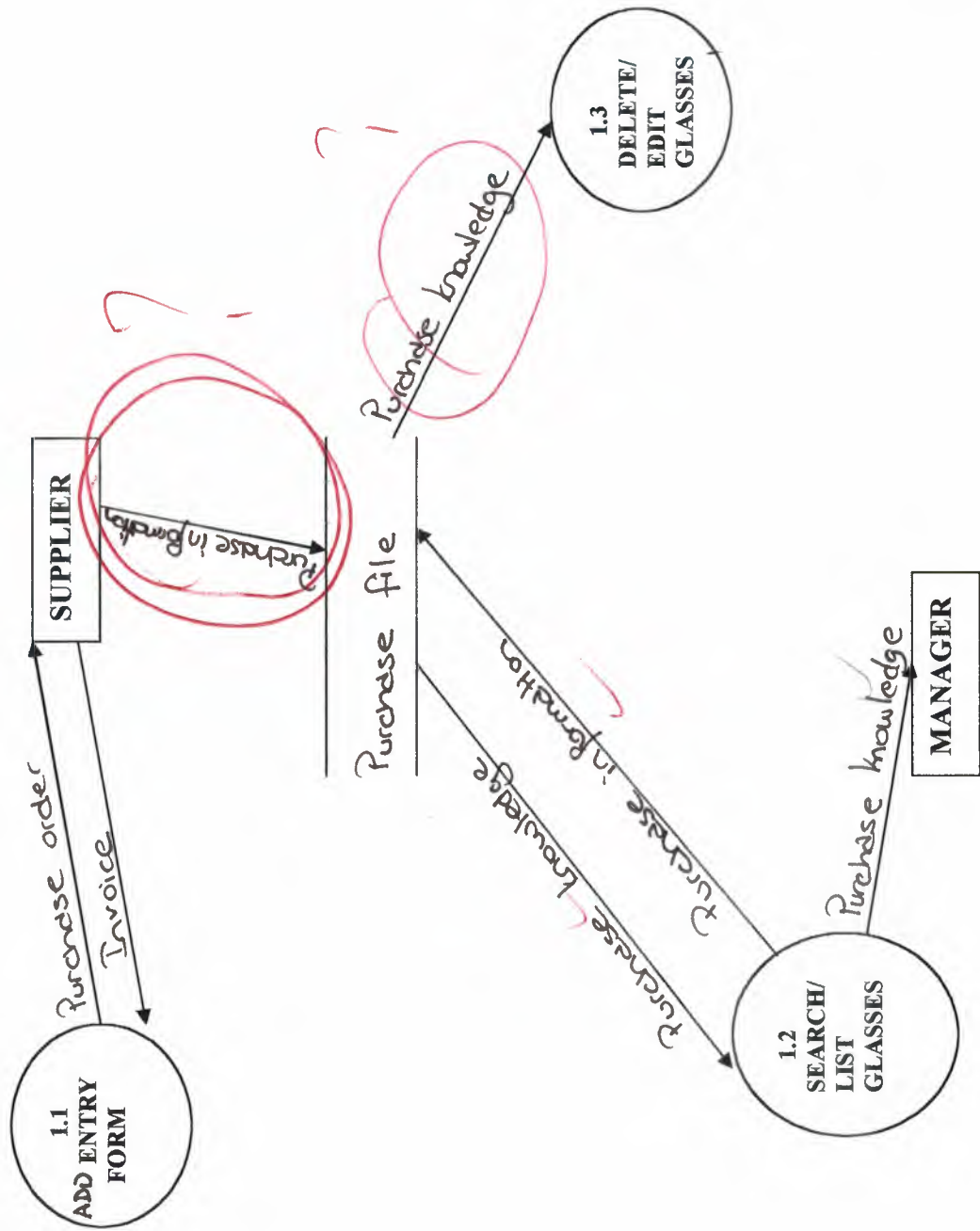
In the accounting procedures , on the other hand , I enter the cost percentage with which the purchaseing cost of the glasses is replace by that percentage . From here on , the expences and revenues are calculated and the payroll of teh company is duly set up . Through the reports procedure though , I can see in the main inventory report , the code number and the name of glasses currently existing in the inventory . We can see the names of customer , their installments amounts , in a report form , through the installment sale report .

In the cash sale report , on the other hand , we can see the list serial number , name of the product , name of the customer , the amount of the product sold, its total price and date.

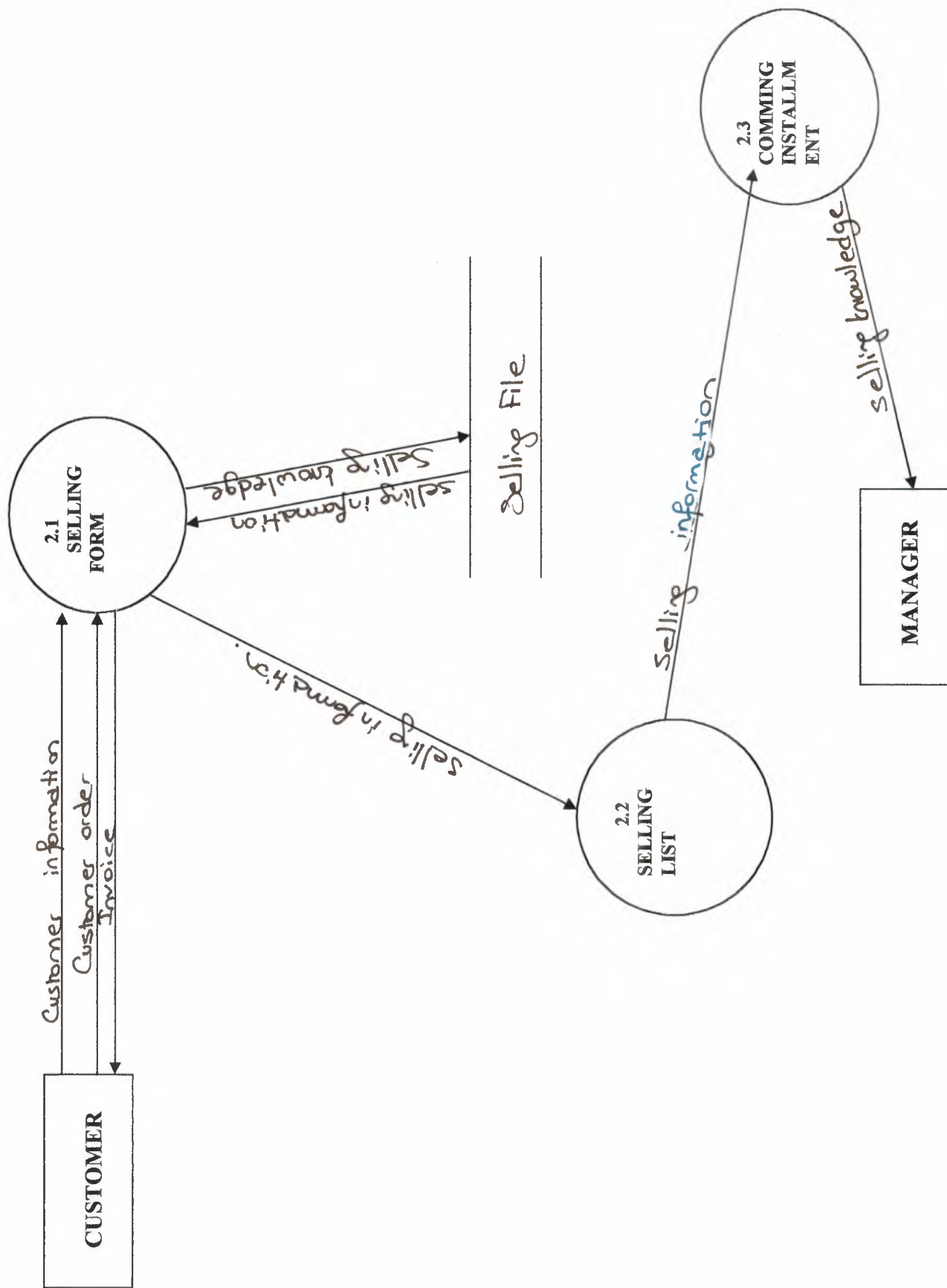
IV. EXPLANATION OF THE SYSTEM
DATA FLOW DIAGRAM
CONTEXT DIAGRAM DFD FOR THE OPTIC STOCK SYSTEM



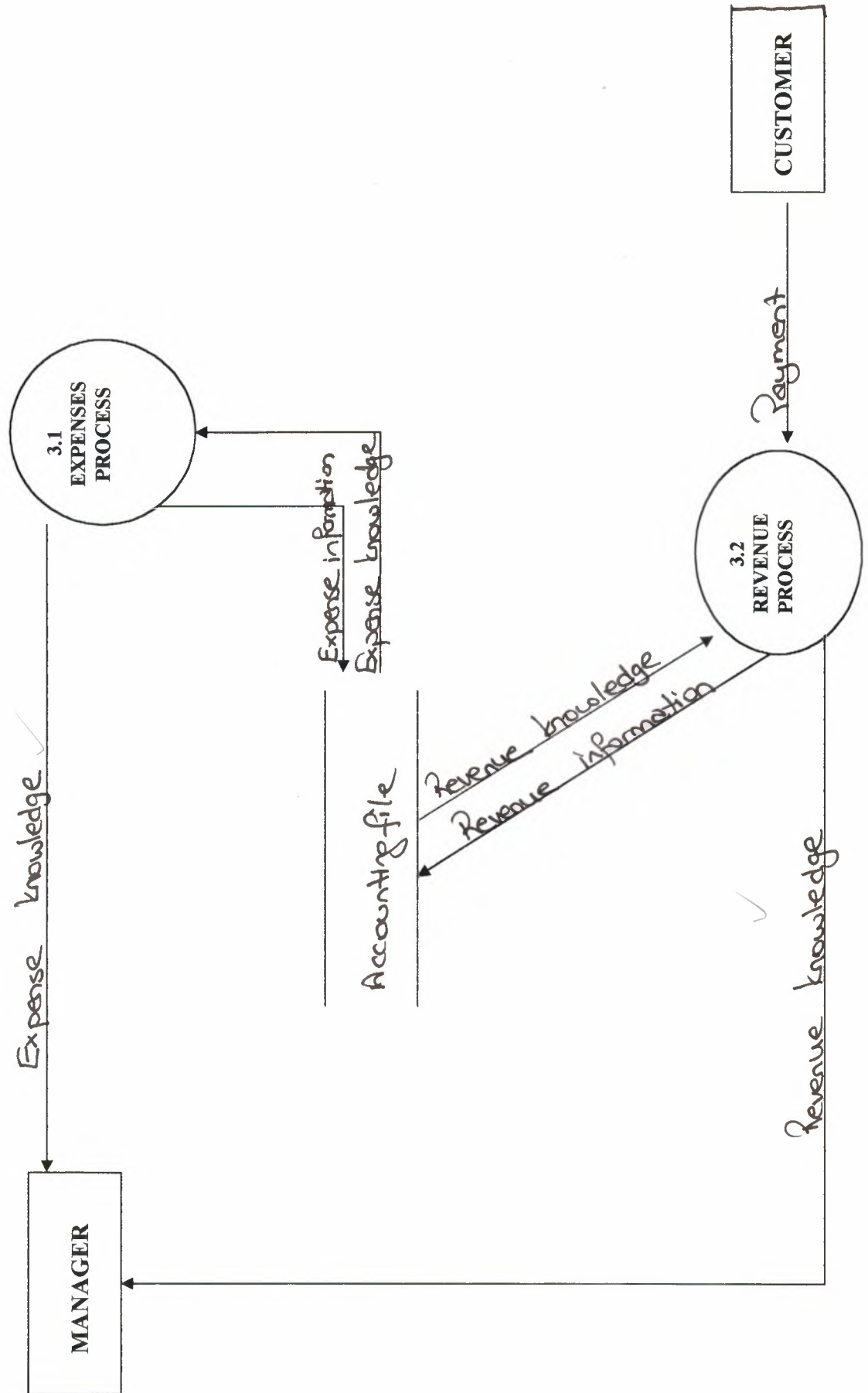
DETAILED DFD FOR PURCHASE PROCESS(1)



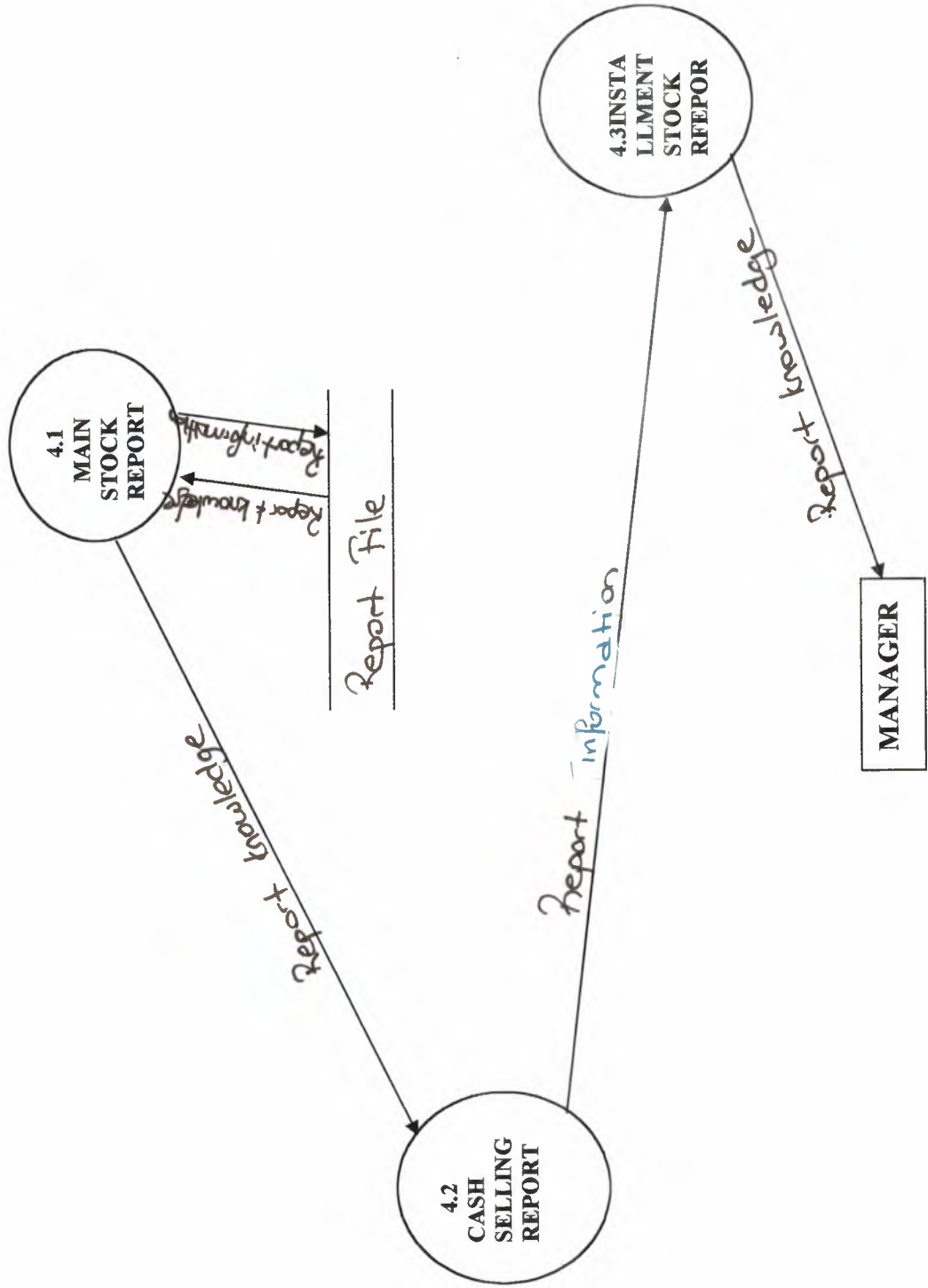
DETAIL DFD FOR SELLING PROCESS(2)



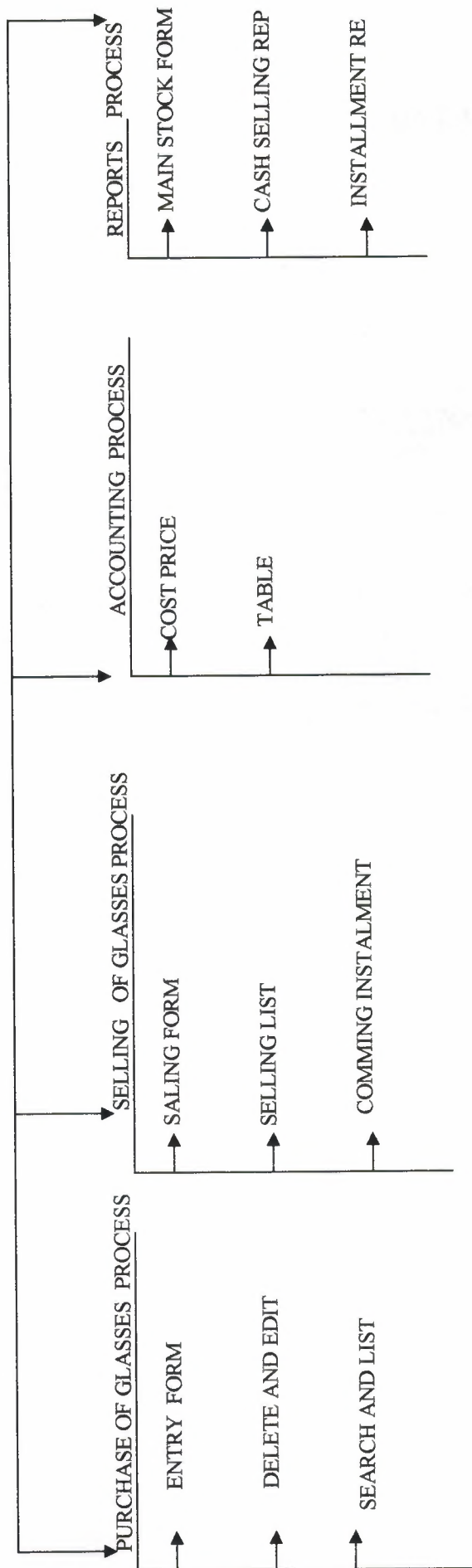
DETAILED DFD OF ACCOUNT
PROCESS(3)



DETAILED DFD FOR REPORT PROCESS(4)



SYSTEM BLOCK DIAGRAM



DATABASE DESING

FRAME STOCK.DB

	Field Name	Type	Size	Key
1	LIST NO	I		
2	PRODUCT CODE	A	9	
3	PRODUCT NAME	A	21	
4	PRODUCT PEACE	I		
5	PRODUCT SIZE	A	12	
6	CONTROL UNIT	I		
7	BUYING PRICE	I		
8	BUYING PLACE	A	21	
9	CELLING PRICE	I		
10	BUYING DATE	D		
11	SPECIAL NOTES	A	21	

ACCOUNTING.DB

	Field Name	Type	Size	Key
1	LIST NO	I		
2	PRODUCT NAME	A	21	
3	PRODUCT CODE	A	9	
4	BUYING PRICE	A	15	
5	COST PRICE	A	15	
6	CASH SAILING	A	15	
7	INSTALLMENT SAILING	A	15	
8	PROFIT LIMIT	A	15	
9	SAILING CHARGER	A	16	
10	SAILING DATE	D		

CASH SELLING.DB

	Field Name	Type	Size	Key
1	LIST NO	I		
2	CUSTOMER NAME	A	21	
3	PRODUCT CODE	A	9	
4	PRODUCT NAME	A	21	
5	PIECE OF SALING	I		
6	BUYING PRICE	I		
7	SALING PRICE	I		
8	TOTAL PRICE	I		
9	COMPANY NAME	A	15	
10	SALING CHARGER	A	15	
11	SALING DATE	D		

GLASSES STOCK.DB

	Field Name	Type	Size	Key
1	LIST NO	I		
2	PRODUCT CODE	A	9	
3	PRODUCT NAME	A	21	
4	PRODUCT PEACE	I		
5	PRODUCT SIZE	A	12	
6	CONTROL UNIT	I		
7	CELLING PRICE	I		
8	BUYING PRICE	I		
9	BUYING PLACE	A	16	
10	BUYING DATE	D		
11	SPECIAL NOTES	A	21	

INSTALLMENT .DB

	Field Name	Type	Size	Key
1	LIST NUMBER	I		
2	CUSTOMER CODE	A	12	
3	CUSTOMER NAME	A	21	
4	PRODUCT CODE	A	12	
5	PRODUCT NAME	A	21	
6	COMPANY NAME	A	21	
7	SAILING CHARGER	A	15	
8	SAILING DATE	D		
9	PIECE OF SAILING	I		
10	SAILING PRICE	I		
11	TOTAL PRICE	I		

V. USER MANUAL

How to install this program:

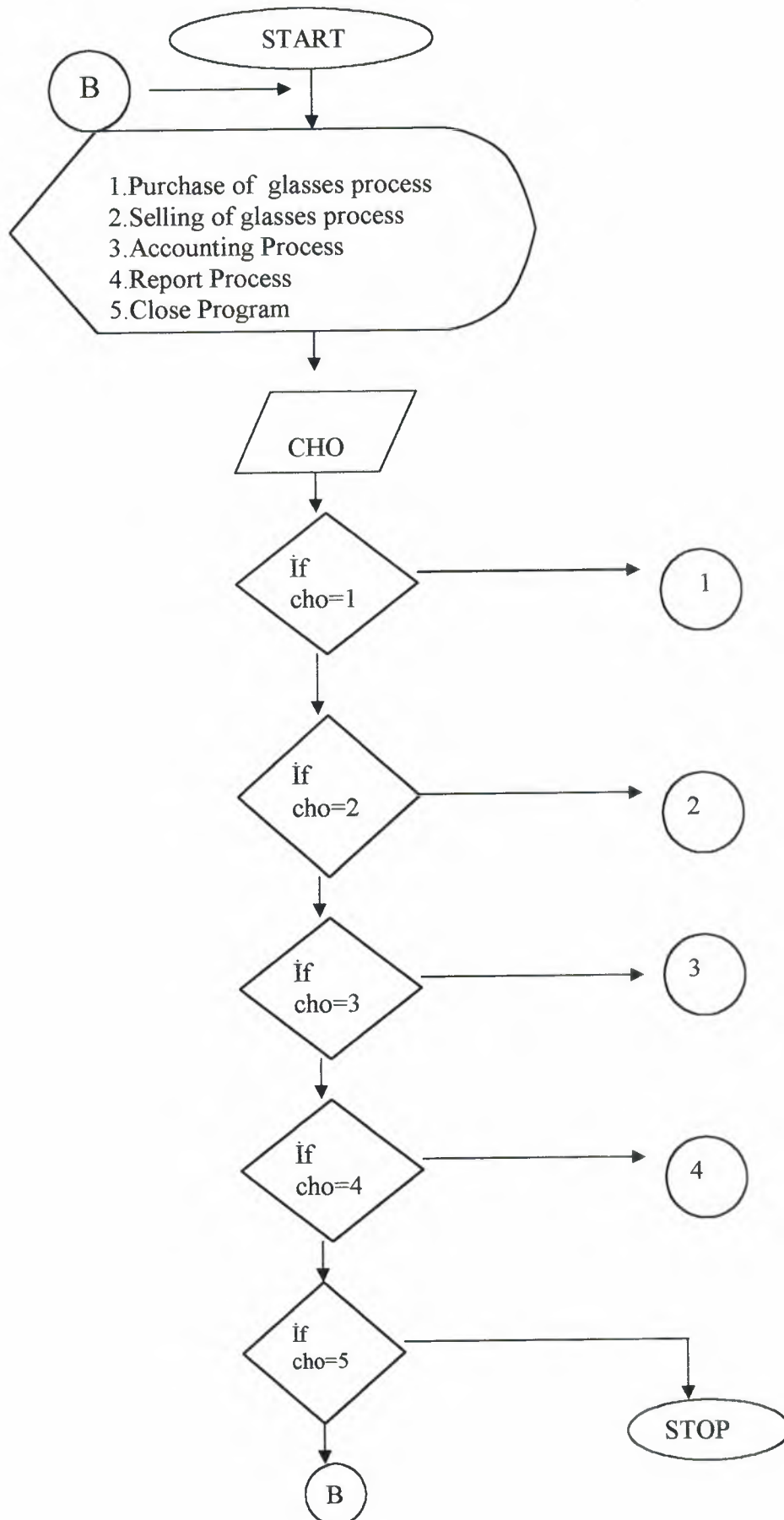
1. Put the CD into the CD-ROM. Then click the my computer icon.
2. Click CD-ROM icon to open CD. Then click the project icon.
3. Now click the set up icon
4. This set up page, click this icon to start installation.
5. The set up will guide you to install program successfully. But you may see the some messages title this. Click OK to update some system files.
6. They you will see message again ask you restart your computer. Click OK again. If everything okey the set up will start run the program.

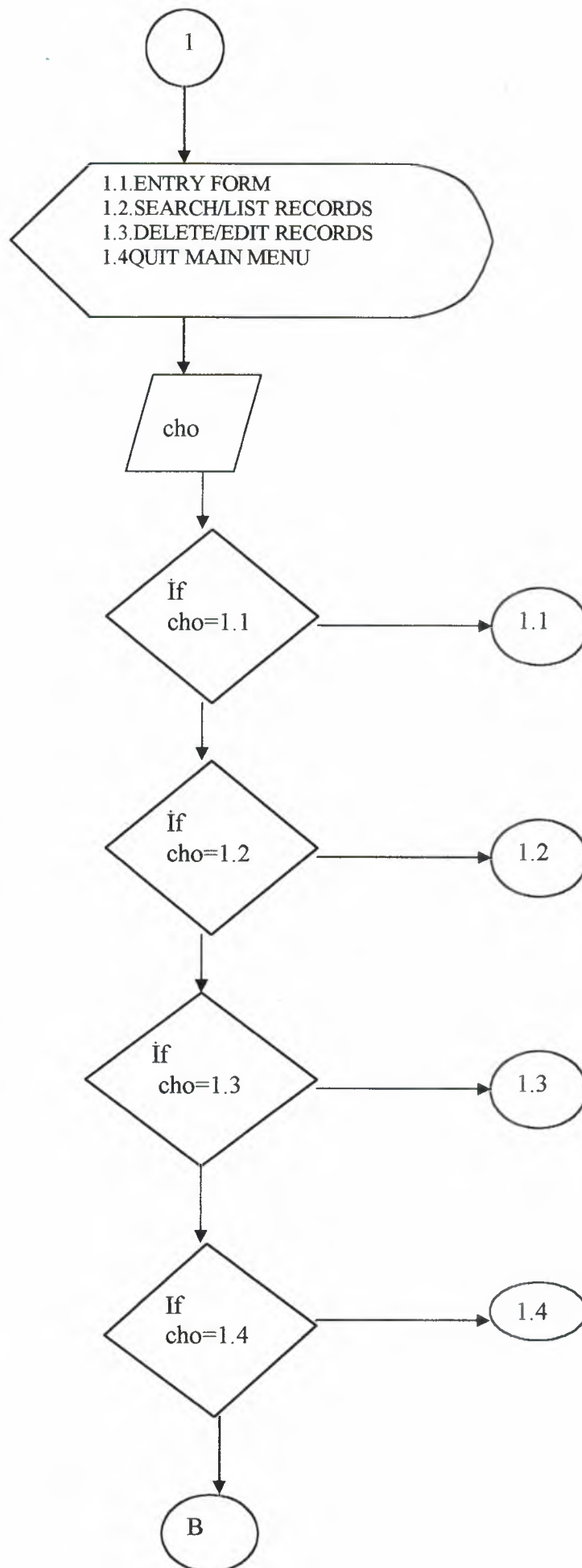
NOTE: Delphi 6.0 must be install in your computer.

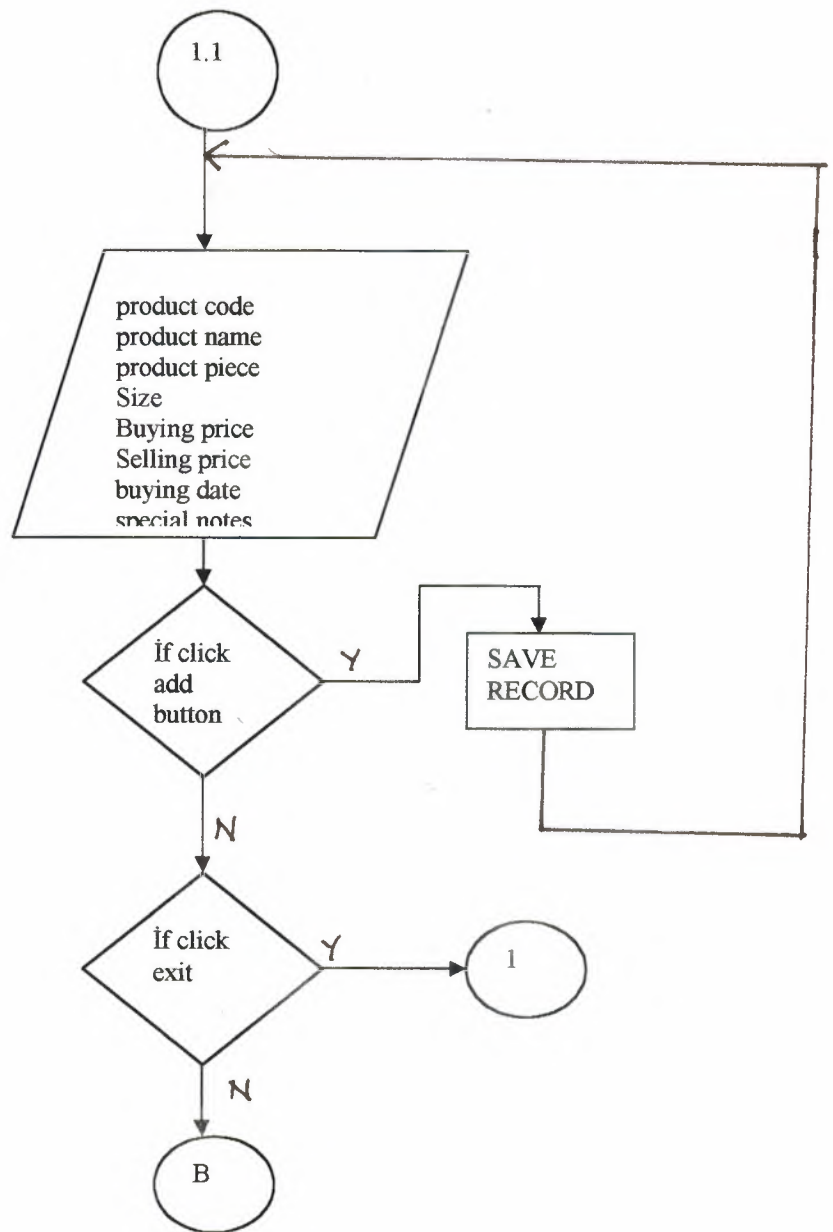
If you click project icon ,you can see the main menu :figure 1
If you add the product in database , you click add entry form : figure2
Search the product in database ,you click search /list button:figure 3
Delete and edit product in database , you click delete/edit button:figure4
If you control the product minimum stock level , you click stck control button:figure 5
If You want to quite menu, you click quite menu button :figure 6
If you want to selling product cash or installment ,click selling form:figure 7
The product is selling cash click cash selling button:figure 8
The product is selling installment click installment selling button:figure 9
If you want to se cash selling all list ,click cash selling all list button: figure 10
If you want to see list by customer ,click list by customer b button: figure 11
If you want to see list by date ,click list by date button: figure 12
If you want to quit menu ,click quit menu button:figure13
If you want to see installment selling all list ,click installment selling all list button: figure 14
If you want to see installment list by customer ,click list by customer b button: figure 15
If you want to see list by date ,click list by date button: figure 16
If you want to quit menu ,click quit menu button:figure17
If you want to calculate percentage of cost price ,click cost limit button:figure 18
If you want to see expense and revenue of this company figure 19
If you want to quit menu, click quit menu button:figure20
If you collect installment payment of customer ,click coming installment button figure 21
If you want to see all report in program,you click report button figure22
If you want to see main stock report ,clickmain stock report button figure 23
If you want to see cash selling report ,click cash selling report button figure 24
If you want to see installment selling report ,click installment selling report button figure25

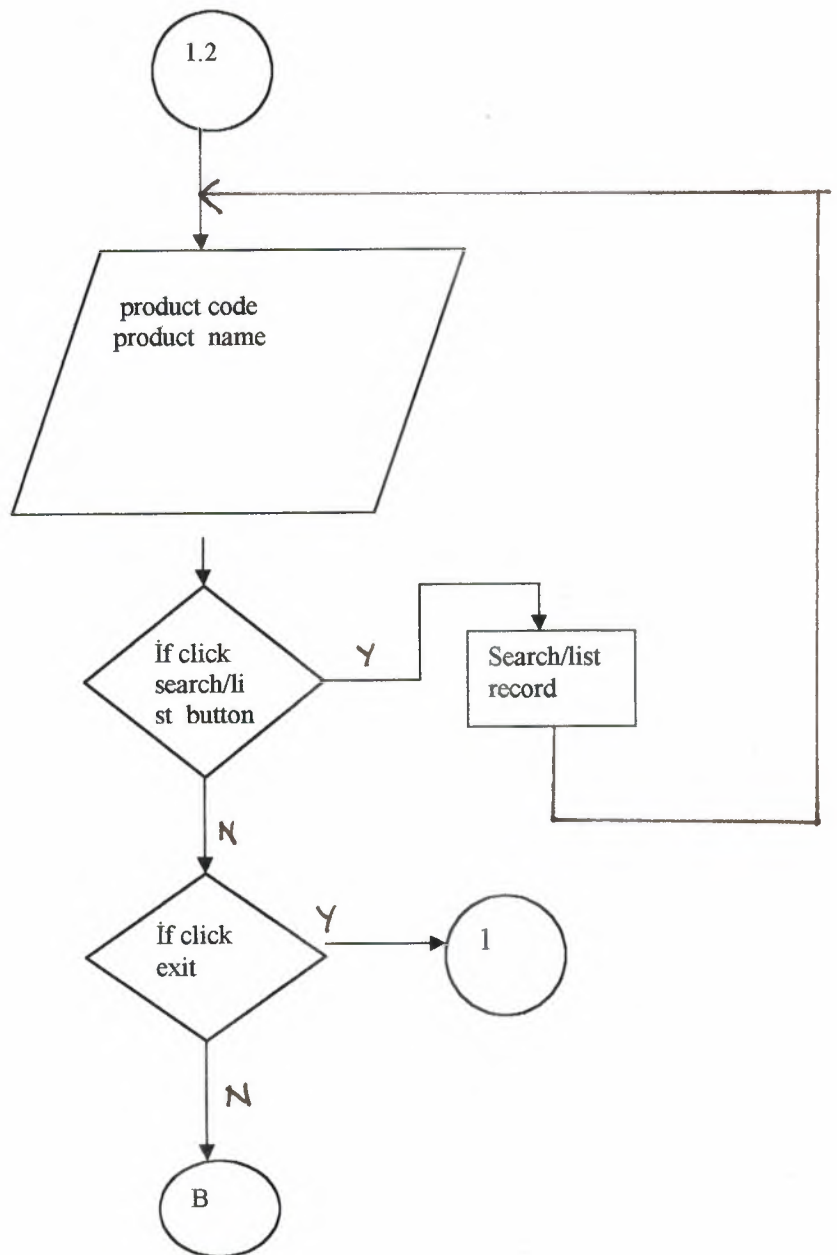
VI. LIST OF SUBPROGRAM

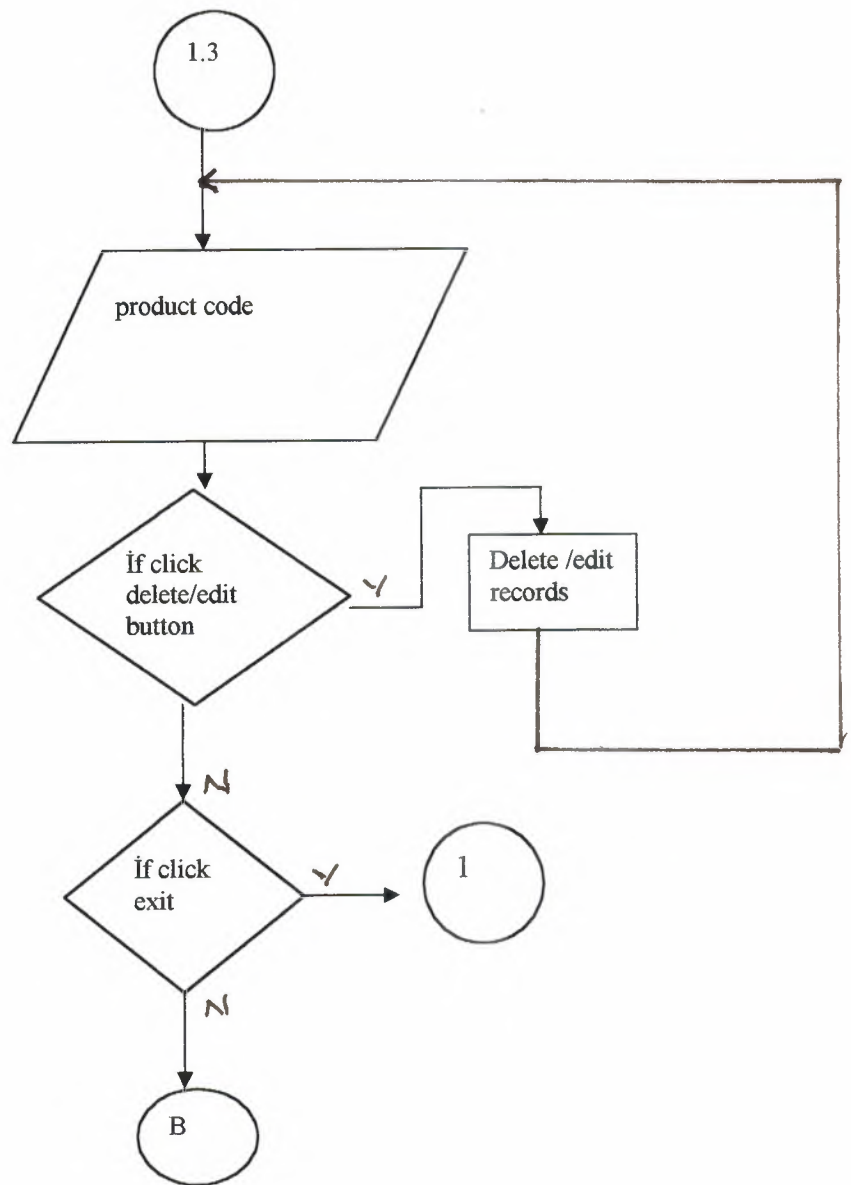
USER FLOWCHART

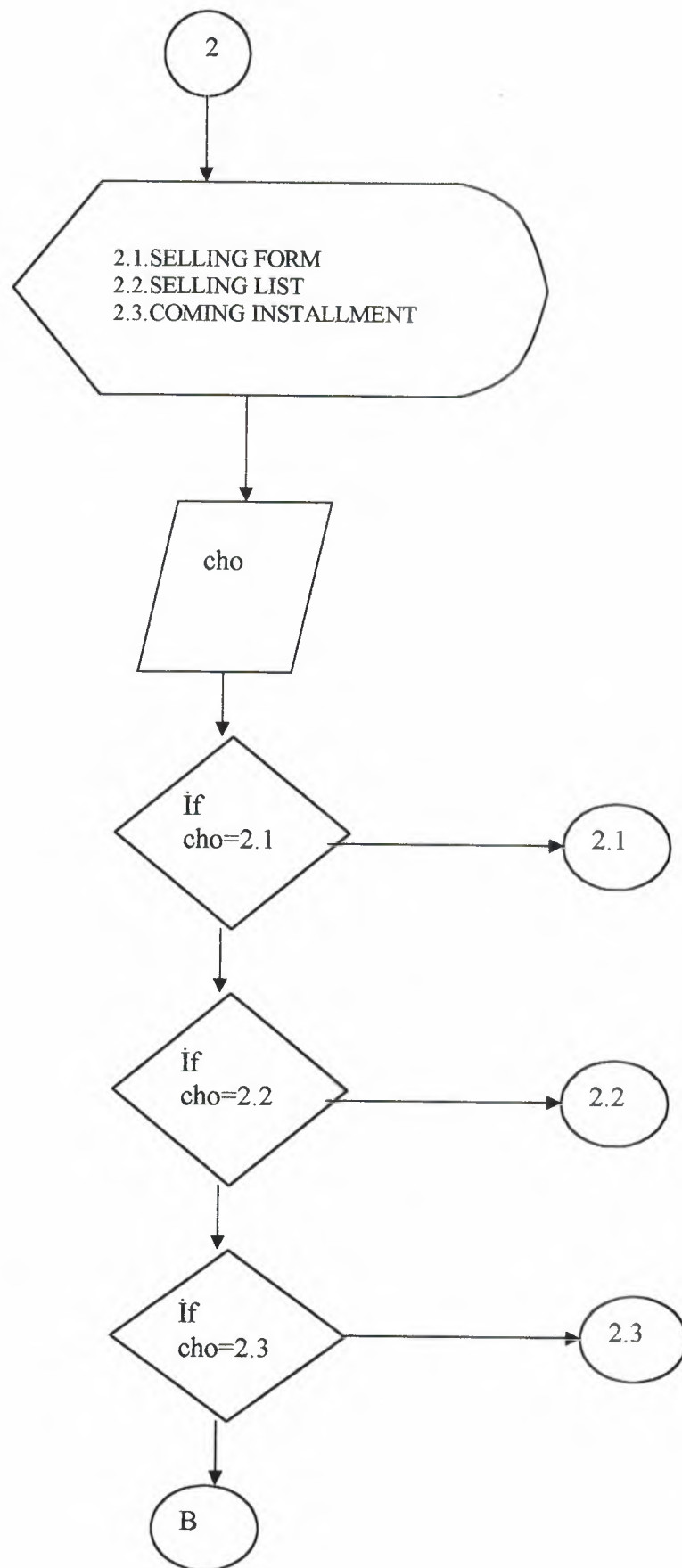


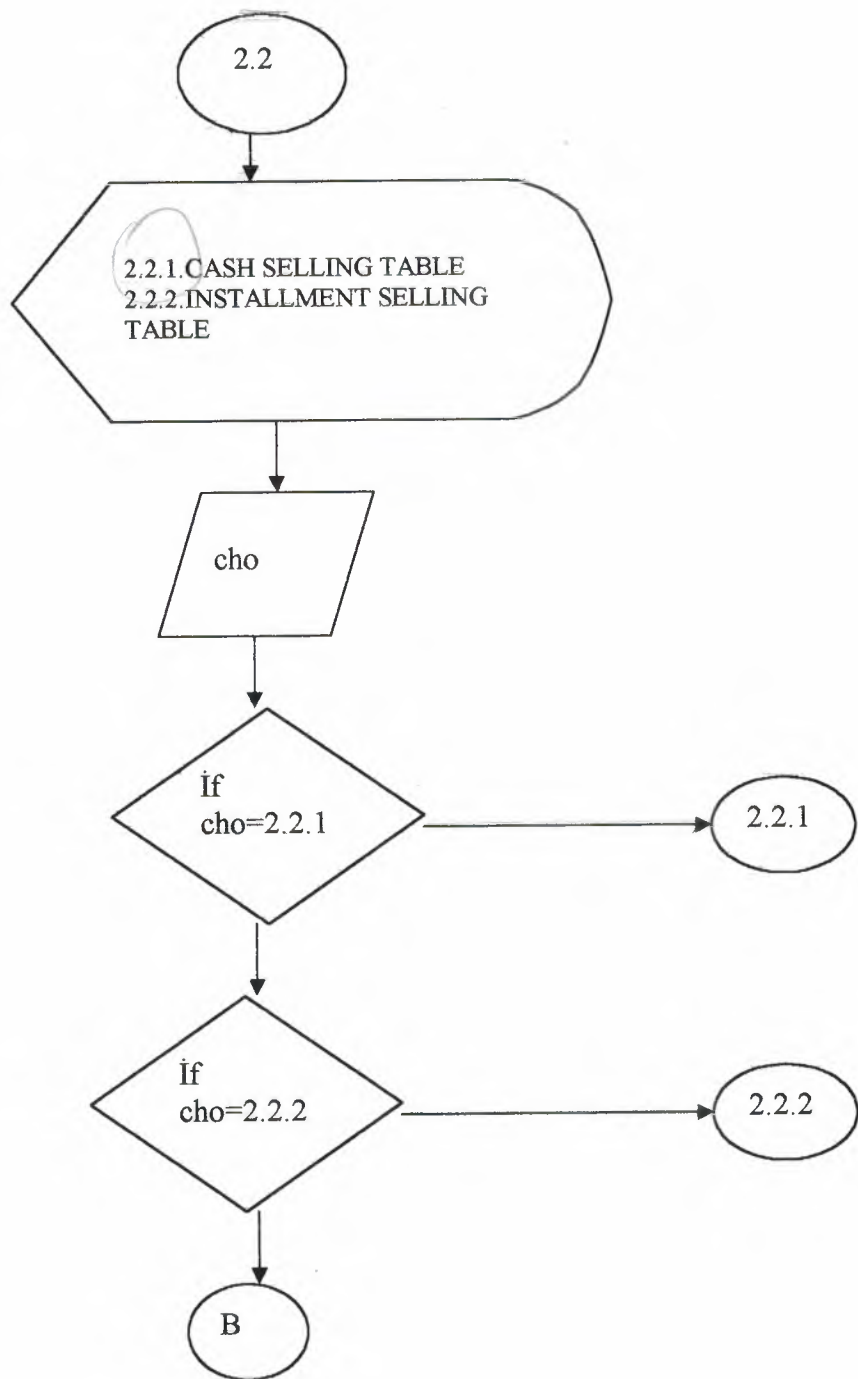


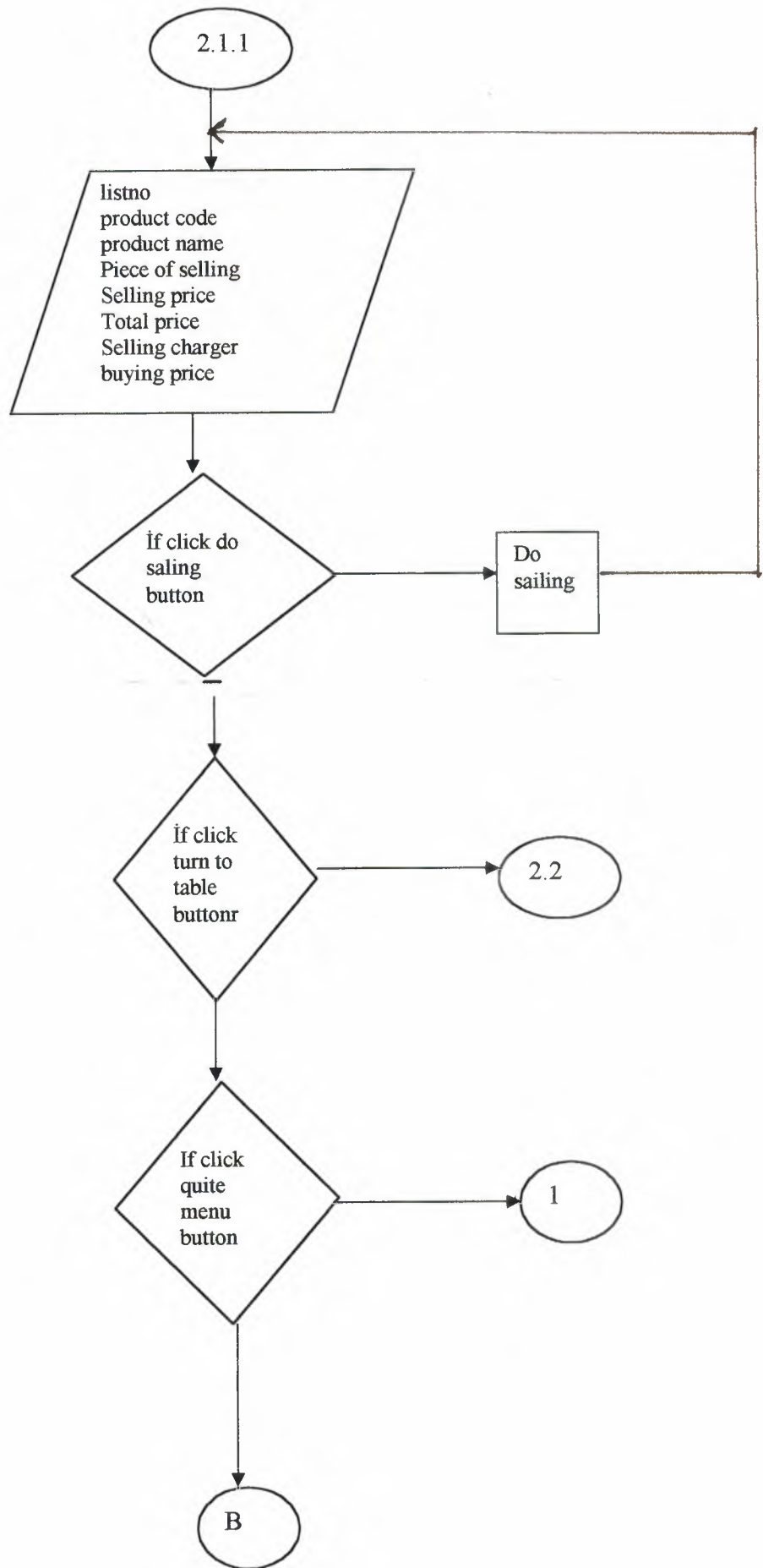


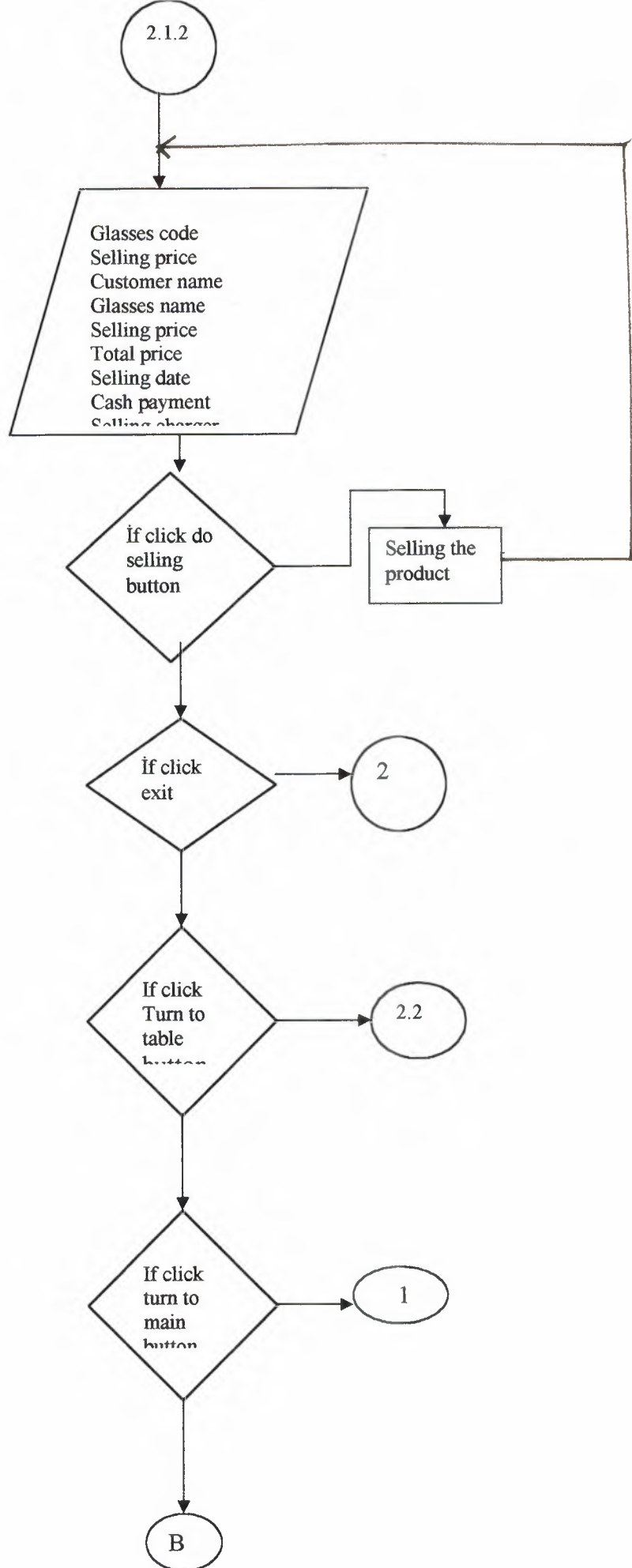


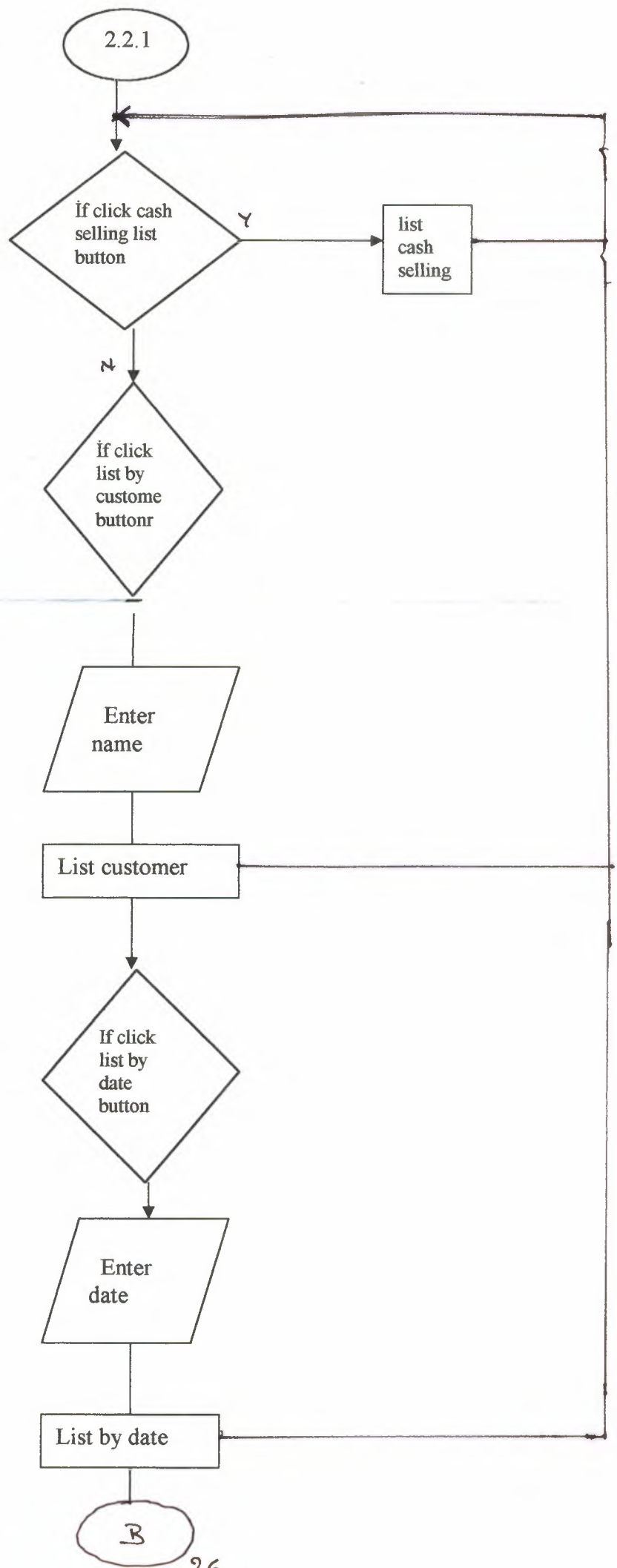


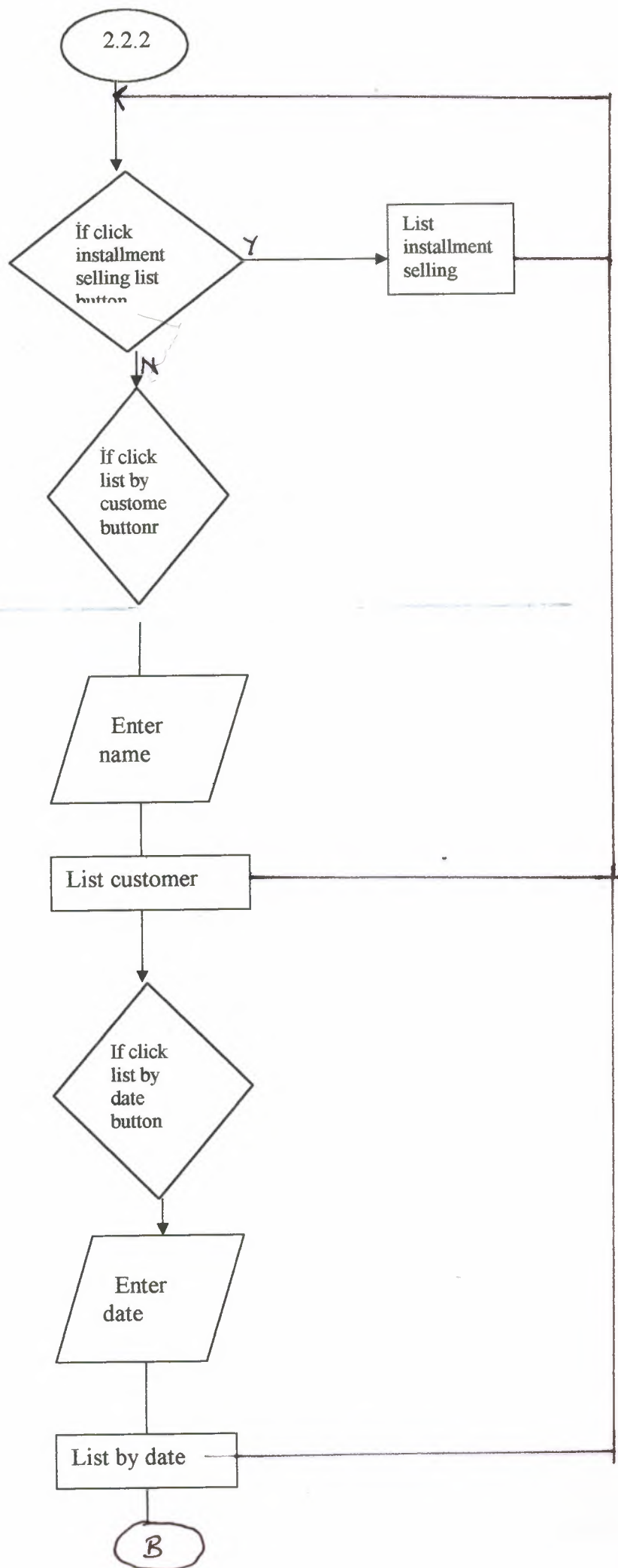


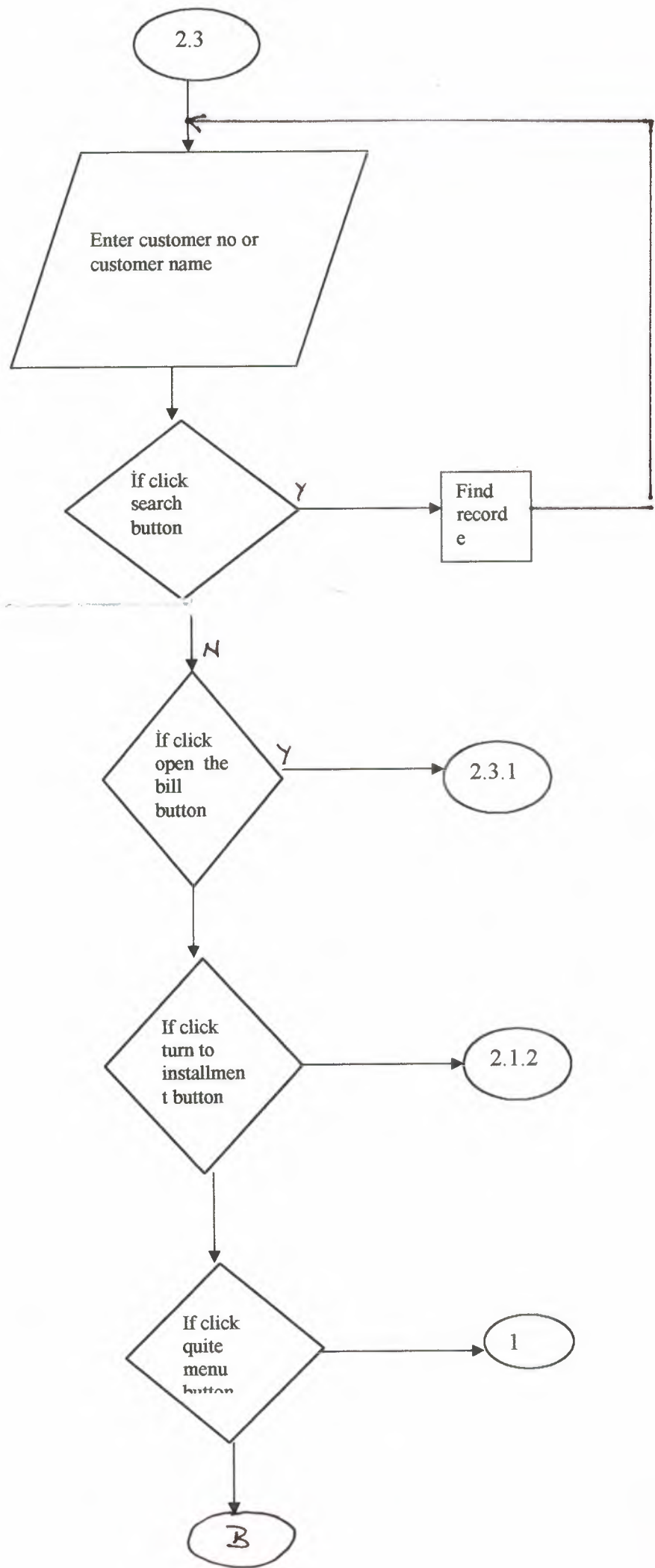


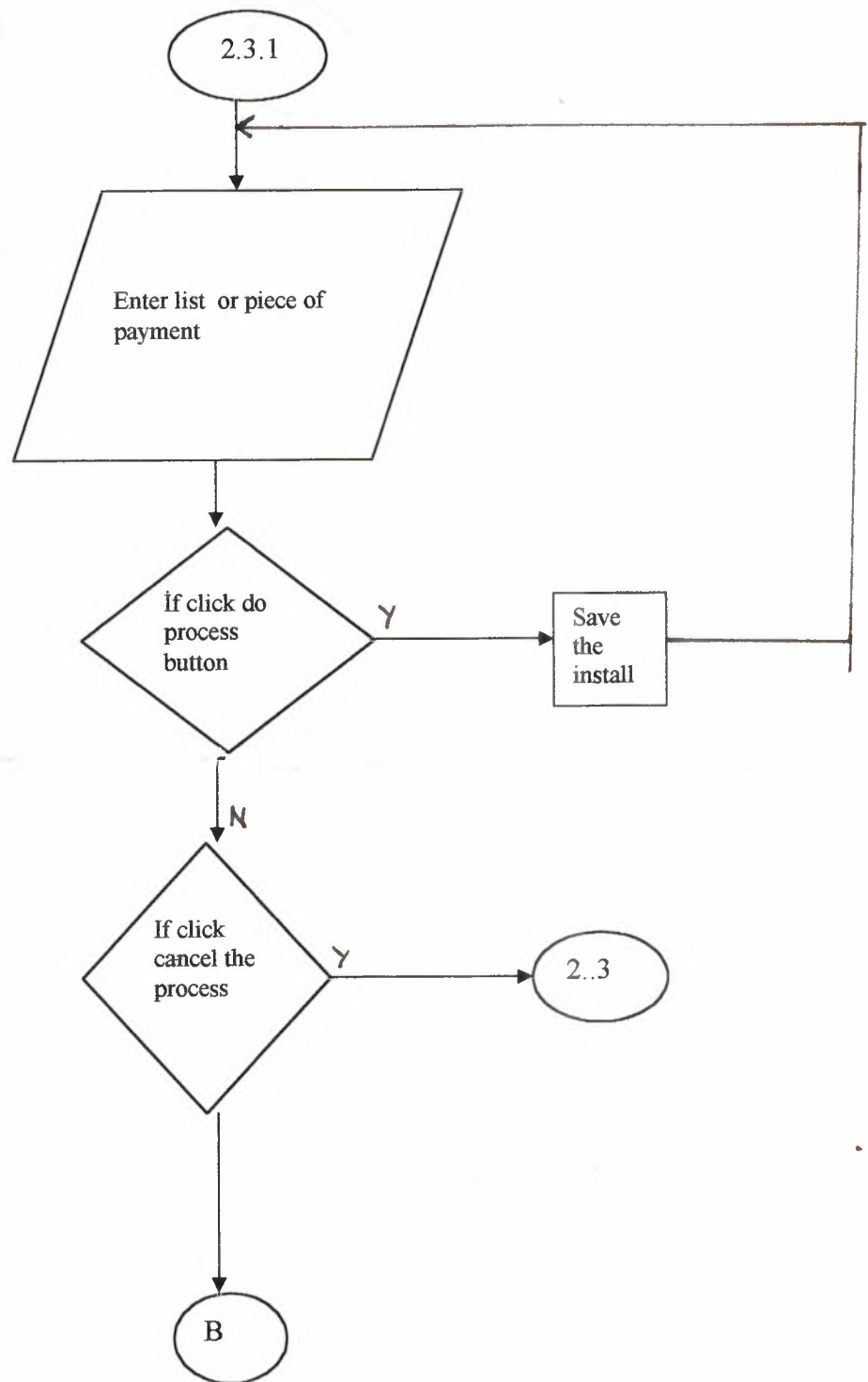


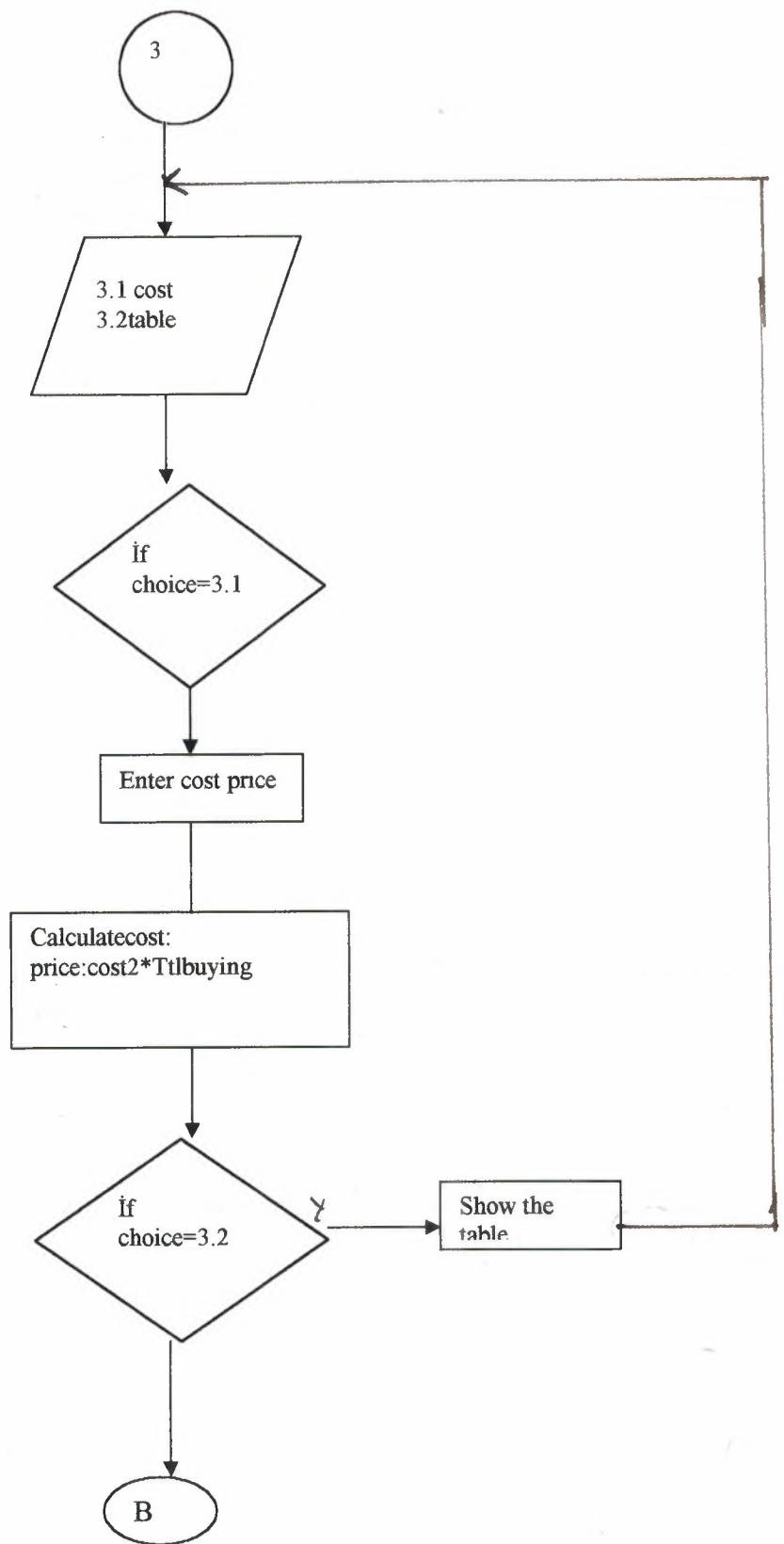


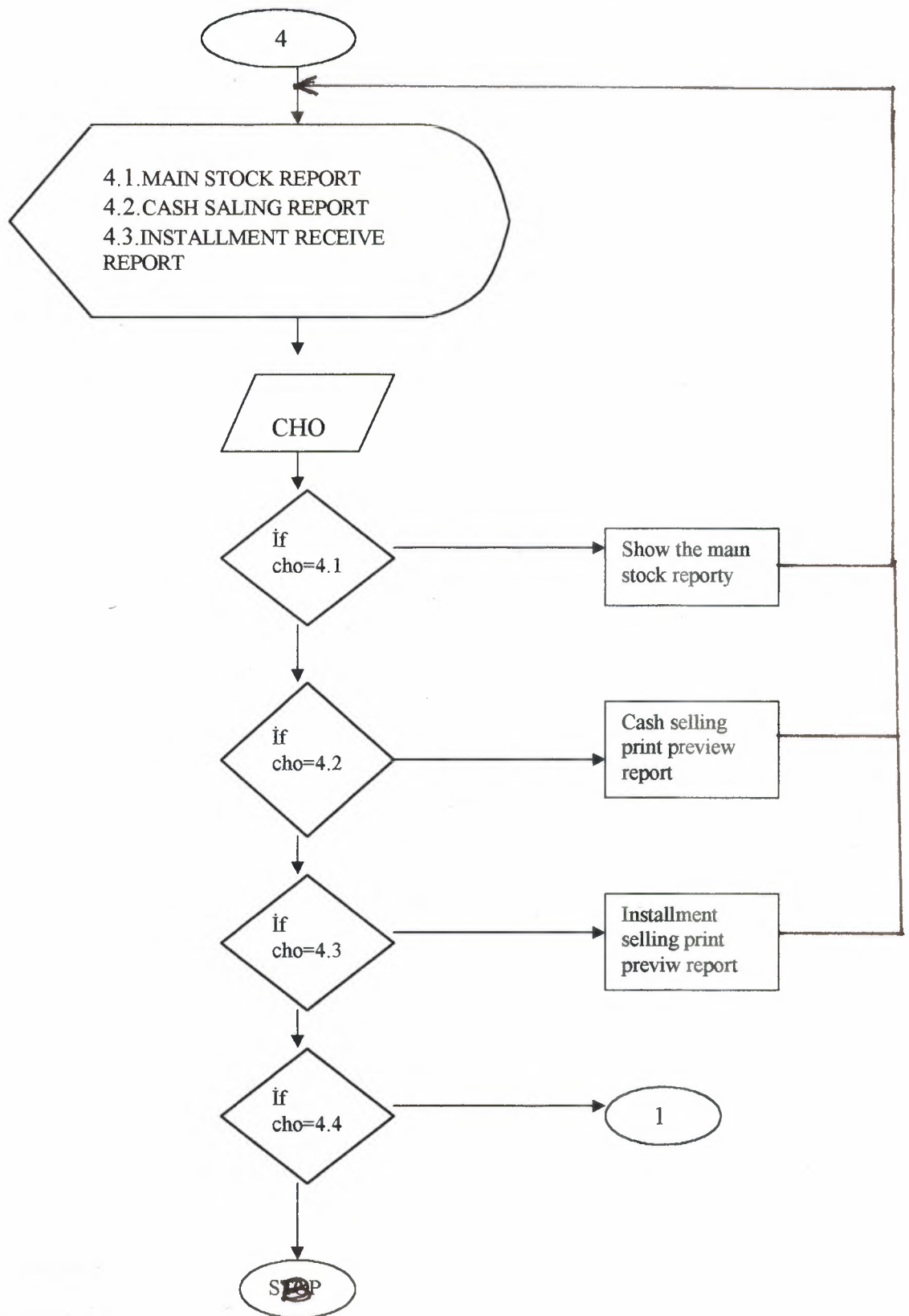












SOURCE PROGRAM

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, StdCtrls, Buttons;

type

TForm1 = class(TForm)

GroupBox1: TGroupBox;

StaticText1: TStaticText;

StaticText2: TStaticText;

BitBtn1: TBitBtn;

BitBtn4: TBitBtn;

BitBtn5: TBitBtn;

BitBtn6: TBitBtn;

BitBtn7: TBitBtn;

BitBtn8: TBitBtn;

BitBtn9: TBitBtn;

BitBtn10: TBitBtn;

BitBtn11: TBitBtn;

BitBtn12: TBitBtn;

Panel3: TPanel;

BitBtn13: TBitBtn;

BitBtn14: TBitBtn;

BitBtn15: TBitBtn;

BitBtn16: TBitBtn;

BitBtn17: TBitBtn;

BitBtn18: TBitBtn;

Panel4: TPanel;

procedure BitBtn11Click(Sender: TObject);

procedure BitBtn18Click(Sender: TObject);

procedure BitBtn14Click(Sender: TObject);

procedure BitBtn17Click(Sender: TObject);

procedure BitBtn7Click(Sender: TObject);

procedure BitBtn8Click(Sender: TObject);

procedure BitBtn9Click(Sender: TObject);

procedure BitBtn12Click(Sender: TObject);

procedure BitBtn10Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

uses Unit2, Unit3, Unit5, Unit6, Unit8, Unit9, Unit10, Unit14;

{ \$R *.dfm }

```
procedure TForm1.BitBtn11Click(Sender: TObject);
begin
  HALT;
end;
```

```
procedure TForm1.BitBtn18Click(Sender: TObject);
begin
  Form2.Show;
  Form1.Visible := False;
end;
```

```
procedure TForm1.BitBtn14Click(Sender: TObject);
begin
  Form2.Show;
  Form1.Visible := False;
end;
```

```
procedure TForm1.BitBtn17Click(Sender: TObject);
begin
  Form2.Show;
  Form1.Visible := False;
end;
```

```
procedure TForm1.BitBtn7Click(Sender: TObject);
begin
  Form5.Show;
  Form1.Visible := False;
end;
```

```
procedure TForm1.BitBtn8Click(Sender: TObject);
begin
  Form8.Show;
  Form1.Visible := False;
end;
```

```
procedure TForm1.BitBtn9Click(Sender: TObject);
begin
  Form9.Show;
  Form1.Visible := False;
end;
```

```
procedure TForm1.BitBtn12Click(Sender: TObject);
begin
```

```
Form10.Show;  
Form1.Visible := False;  
end;
```

```
procedure TForm1.BitBtn10Click(Sender: TObject);  
begin  
Form14.Show;  
Form1.Visible := False;  
end;  
  
end.
```

unit Unit2;

interface

uses

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

type

TForm2 = class(TForm)
 PageControl1: TPageControl;
 TabSheet1: TTabSheet;
 TabSheet2: TTabSheet;
 TabSheet3: TTabSheet;
 Edit1: TEdit;
 Edit2: TEdit;
 Edit3: TEdit;
 Edit4: TEdit;
 Edit5: TEdit;
 Edit7: TEdit;
 Edit8: TEdit;
 Label1: TLabel;
 Label2: TLabel;
 Label3: TLabel;
 Label4: TLabel;
 Label5: TLabel;
 Label6: TLabel;
 Label7: TLabel;
 Label8: TLabel;
 BitBtn1: TBitBtn;
 BitBtn2: TBitBtn;
 DBGrid1: TDBGrid;
 DataSource1: TDataSource;
 GroupBox1: TGroupBox;
 Label9: TLabel;
 Edit9: TEdit;
 BitBtn3: TBitBtn;
 TabSheet4: TTabSheet;
 BitBtn4: TBitBtn;
 BitBtn5: TBitBtn;
 BitBtn6: TBitBtn;
 BitBtn7: TBitBtn;
 BitBtn8: TBitBtn;
 BitBtn9: TBitBtn;
 BitBtn10: TBitBtn;
 BitBtn11: TBitBtn;
 BitBtn12: TBitBtn;
 BitBtn13: TBitBtn;
 BitBtn14: TBitBtn;

```

begin
  If (Table1PRODUCT_CODE.Text = Edit1.Text) Then
    begin
      Aranan := True;
      ShowMessage('BY SEARCHING ON TABLE IT SHOWED FROM,
'+Table1PRODUCT_CODE.Text+'      THIS PRODUCT CODE AND
'+Table1PRODUCT_NAME.Text+'      NAMED PRODUCT
'+Table1PRODUCTPEACE.Text+'      PEACE IS FOUNDED!');
    end;
    Table1.Next;
  end;

```

```

  If (Aranan = False ) Then
    begin
      ShowMessage('THE PRODUCT IS NOT FOUNDED!');
    end;

```

```

end;

```

```

procedure TForm2.BitBtn23Click(Sender: TObject);
begin
  Edit1.Text := ''; Edit5.Text := '';
  Edit2.Text := ''; Edit7.Text := '';
  Edit3.Text := ''; Edit7.Text := '';
  Edit4.Text := ''; Edit8.Text := '';
  Edit17.Text := ''; Edit18.Text := '';
end;

```

```

procedure TForm2.BitBtn18Click(Sender: TObject);
begin
  Form10.Show;
  Form2.Visible := False;
end;

```

```

procedure TForm2.BitBtn24Click(Sender: TObject);
begin
  Form11.QuickRep1.Preview;
end;

end.

```



```

Table1.Edit;
Table1PRODUCT_CODE.Text := Edit13.Text;
Table1PRODUCT_NAME.Text := Edit12.Text;
Table1PRODUCTSIZE.Text := Edit10.Text;
Table1BUYINGPLACE.Text := Edit14.Text;
Table1SPECIALNOTES.Text := Edit20.Text;
Table1LISTNO.Value := StrToInt(Edit19.Text);
Table1PRODUCTPEACE.Value := StrToInt(Edit11.Text);
Table1BUYINGPRICE.Value := StrToInt(Edit15.Text);
Table1CELLINGPRICE.Value := StrToInt(Edit16.Text);
Table1CONTROLUNIT.Value := StrToInt(Edit17.Text);
Table1BUYINGDATE.Value := DateTimePicker1.Date;
Table1.UpdateRecord;
Table1.Refresh;
Edit19.Text := ""; Edit15.Text := "";
Edit12.Text := ""; Edit16.Text := "";
Edit13.Text := ""; Edit11.Text := "";
Edit14.Text := ""; Edit10.Text := "";
Edit17.Text := ""; Edit18.Text := "";
ShowMessage('RERECORD PROCESS IS COMPLETED!');
end;
Table1.Next;
end;
end;

end;

```

```

procedure TForm2.BitBtn13Click(Sender: TObject);
begin
Form5.Show;
Form2.Visible := False;
end;

```

```

procedure TForm2.Edit21Change(Sender: TObject);
begin
if (Edit21.Text='') then
begin exit;end
else
Form2.Table1.Locate('PRODUCT_NAME',Edit21.Text,[Locaseinsensitive,Lopartialkey]);
end;

```

```

procedure TForm2.BitBtn1Click(Sender: TObject);
Var
Aranan : Boolean;
begin
Aranan := False;

```

```

Table1.First;
While not Table1.Eof Do

```

```

begin
Aranan := False;

If MessageDlg('ARE YOU SURE THAT YOU WANT TO DELETE THE RECORD FROM
THE TABLE!..',
mtconfirmation, [mbYES, mbNO], 0) = mrYES Then
begin
Table1.First;
While not Table1.Eof Do
begin
If (Edit9.Text = Table1PRODUCT_CODE.Text) Then
begin
Aranan := True;
Table1.Delete;
Table1.Refresh;
Edit19.Text := "; Edit15.Text := ";
Edit12.Text := "; Edit16.Text := ";
Edit13.Text := "; Edit11.Text := ";
Edit14.Text := "; Edit10.Text := ";
Edit17.Text := "; Edit18.Text := ";
ShowMessage('THE RECORD IS DELETED!');
end;
Table1.Next;
end;
end;

If (Aranan = False ) Then
begin
ShowMessage('THE PRODUCT IS NOT FOUNDED!');
end;

end;

```

```

procedure TForm2.BitBtn19Click(Sender: TObject);
Var
Aranan : Boolean;
begin
Aranan := False;

If MessageDlg('ARE YOU SURE THAT YOU WANT TO CHANGE THE RECORD
FROM THE TABLE!..',
mtconfirmation, [mbYES, mbNO], 0) = mrYES Then
begin
Table1.First;
While not Table1.Eof Do
begin
If (Edit9.Text = Table1PRODUCT_CODE.Text) Then
begin
Table1.Refresh;

```

```

begin
if (Edit6.Text='') then
begin exit;end
else
Form2.Table1.Locate('PRODUCT_CODE',Edit6.Text,[Locaseinsensitive,Lopartialkey]);

end;

```

```

procedure TForm2.BitBtn3Click(Sender: TObject);
Var
Aranan : Boolean;
begin
Aranan := False;

```

```

Table1.First;
While not Table1.Eof Do
begin
If (Edit9.Text = Table1PRODUCT_CODE.Text) Then
begin
Aranan := True;
Edit13.Text := Table1PRODUCT_CODE.Text;
Edit12.Text := Table1PRODUCT_NAME.Text;
Edit10.Text := Table1PRODUCTSIZE.Text;
Edit14.Text := Table1BUYINGPLACE.Text;
Edit20.Text := Table1SPECIALNOTES.Text;
Edit19.Text := InttoStr(Table1LISTNO.Value);
Edit11.Text := InttoStr(Table1PRODUCTPEACE.Value);
Edit15.Text := InttoStr(Table1BUYINGPRICE.Value);
Edit16.Text := InttoStr(Table1CELLINGPRICE.Value);
Edit17.Text := InttoStr(Table1CONTROLUNIT.Value);
Edit1.Text := InttoStr(Table1LISTNO.Value);
DateTimePicker1.Date := Table1BUYINGDATE.Value;
ShowMessage('THE PRODUCT IS FOUNDED!');
end;
Table1.Next;
end;

```

```

If (Aranan = False ) Then
begin
ShowMessage('THE PRODUCT IS NOT FOUNDED!');
end;

end;

```

```

procedure TForm2.BitBtn20Click(Sender: TObject);
Var
Aranan : Boolean;

```

```

While not Table1.Eof Do
begin
If (Edit1.Text = Table1PRODUCT_CODE.Text) Then
begin
AyniKod := True;
ShowMessage('SAME PRODUCT CODE IS ENTERED, PLEASE CHANGE IT!');
end;
Table1.Next;
end;

```

```

If (AyniKod = False) Then
begin
Table1.Refresh;
Table1.First;
While not Table1.Eof Do
begin
Table1.Next;
end;

```

```

If (Table1.Eof) Then
begin
ListNo := (Table1LISTNO.Value + 1);
end;

```

```

Table1.Append;
Table1LISTNO.Value := ListNo;
Table1PRODUCT_CODE.Text := Edit1.Text;
Table1PRODUCT_NAME.Text := Edit2.Text;
Table1PRODUCTSIZE.Text := Edit4.Text;
Table1BUYINGPLACE.Text := Edit5.Text;
Table1SPECIALNOTES.Text := Edit18.Text;
Table1PRODUCTPEACE.Value := StrToInt(Edit3.Text);
Table1BUYINGPRICE.Value := StrToInt(Edit7.Text);
Table1CELLINGPRICE.Value := StrToInt(Edit8.Text);
Table1CONTROLUNIT.Value := StrToInt(Edit17.Text);
Table1BUYINGDATE.Value := DateTimePicker1.Date;
Table1.Post;

```

```

Edit1.Text := ''; Edit5.Text := '';
Edit2.Text := ''; Edit7.Text := '';
Edit3.Text := ''; Edit7.Text := '';
Edit4.Text := ''; Edit8.Text := '';
Edit17.Text := ''; Edit18.Text := '';
ShowMessage('RECORD PROCESS IS COMPLETED!');
end;

```

```

end;

```

```

procedure TForm2.Edit6Change(Sender: TObject);

```

```

Table1PRODUCT_CODE: TStringField;
Table1PRODUCT_NAME: TStringField;
Table1PRODUCTPEACE: TIntegerField;
Table1PRODUCTSIZE: TStringField;
Table1CONTROLUNIT: TIntegerField;
Table1BUYINGPRICE: TIntegerField;
Table1BUYINGPLACE: TStringField;
Table1CELLINGPRICE: TIntegerField;
Table1BUYINGDATE: TDateField;
Table1SPECIALNOTES: TStringField;
procedure BitBtn8Click(Sender: TObject);
procedure BitBtn22Click(Sender: TObject);
procedure Edit6Change(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn20Click(Sender: TObject);
procedure BitBtn19Click(Sender: TObject);
procedure BitBtn13Click(Sender: TObject);
procedure Edit21Change(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn23Click(Sender: TObject);
procedure BitBtn18Click(Sender: TObject);
procedure BitBtn24Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form2: TForm2;

implementation

uses Unit1, Unit5, Unit10, Unit11, Unit12;

{$R *.dfm}

procedure TForm2.BitBtn8Click(Sender: TObject);
begin
  Form1.Show;
  Form2.Visible := False;
end;

procedure TForm2.BitBtn22Click(Sender: TObject);
Var
  AyniKod : Boolean; ListNo : Integer;
begin
  AyniKod := False; ListNo := 0;

  Table1.First;

```


BitBtn15: TBitBtn;
BitBtn16: TBitBtn;
BitBtn17: TBitBtn;
BitBtn18: TBitBtn;
Edit10: TEdit;
Edit11: TEdit;
Edit12: TEdit;
Edit13: TEdit;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Label16: TLabel;
Label17: TLabel;
Edit14: TEdit;
Edit15: TEdit;
Edit16: TEdit;
DateTimePicker1: TDateTimePicker;
DateTimePicker2: TDateTimePicker;
BitBtn19: TBitBtn;
BitBtn20: TBitBtn;
GroupBox2: TGroupBox;
Label18: TLabel;
Label20: TLabel;
Edit17: TEdit;
Label19: TLabel;
BitBtn22: TBitBtn;
BitBtn23: TBitBtn;
BitBtn24: TBitBtn;
GroupBox3: TGroupBox;
GroupBox4: TGroupBox;
GroupBox5: TGroupBox;
GroupBox6: TGroupBox;
GroupBox7: TGroupBox;
Label21: TLabel;
Edit18: TEdit;
GroupBox8: TGroupBox;
GroupBox9: TGroupBox;
Edit6: TEdit;
Edit19: TEdit;
Label22: TLabel;
Label23: TLabel;
Edit20: TEdit;
DBNavigator1: TDBNavigator;
Label24: TLabel;
Edit21: TEdit;
Table1: TTable;
Table1LISTNO: TIntegerField;

unit Unit3;

interface

uses

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

type

TForm3 = class(TForm)
 BitBtn1: TBitBtn;
 Label1: TLabel;
 Edit1: TEdit;
 Label2: TLabel;
 Edit2: TEdit;
 Label5: TLabel;
 Edit5: TEdit;
 Label8: TLabel;
 GroupBox1: TGroupBox;
 Panel1: TPanel;
 Label6: TLabel;
 Edit6: TEdit;
 Label7: TLabel;
 Edit7: TEdit;
 BitBtn2: TBitBtn;
 DateTimePicker1: TDateTimePicker;
 BitBtn3: TBitBtn;
 BitBtn4: TBitBtn;
 Label3: TLabel;
 Edit3: TEdit;
 Label4: TLabel;
 Edit4: TEdit;
 BitBtn5: TBitBtn;
 BitBtn6: TBitBtn;
 Label9: TLabel;
 Edit8: TEdit;
 Table2: TTable;
 Table2LISTNO: TIntegerField;
 Table2CUSTOMERNAME: TStringField;
 Table2PRODUCTCODE: TStringField;
 Table2PRODUCTNAME: TStringField;
 Table2PIECEOFSALE: TIntegerField;
 Table2SALINGPRICE: TIntegerField;
 Table2TOTALPRICE: TIntegerField;
 Table2COMPANYNAME: TStringField;
 Table2SALINGCHARGER: TStringField;
 Table2SALINGDATE: TDateField;
 Table1: TTable;
 Table1LISTNO: TIntegerField;
 Table1PRODUCT_CODE: TStringField;

```
begin  
Form6.Show;  
Form3.Visible := False;  
end;  
  
end.
```

```

procedure TForm3.Edit6Change(Sender: TObject);
begin
  Table1.First;
  While not Table1.Eof Do
  begin
    If (Edit6.Text = Table1PRODUCT_CODE.Text) Then
    begin
      Edit3.Text := Table1PRODUCT_NAME.Text;
      Edit5.Text := InttoStr(Table1CELLINGPRICE.Value);
    end;
    Table1.Next;
  end;
end;

```

```

procedure TForm3.Edit3Change(Sender: TObject);
begin
  Table1.First;
  While not Table1.Eof Do
  begin
    If (Table1PRODUCT_NAME.Text = Edit3.Text) Then
    begin
      Edit6.Text := Table1PRODUCT_CODE.Text;
      Edit5.Text := InttoStr(Table1CELLINGPRICE.Value);
    end;
    Table1.Next;
  end;
end;

```

```

procedure TForm3.Edit7Change(Sender: TObject);
Var
  Price : Integer;
begin
  If (Edit7.Text <> "") Then
  begin
    Price := ((StrToInt(Edit7.Text)) * (StrToInt(Edit5.Text)));
    Edit4.Text := InttoStr(Price);
  end;
end;

```

```

procedure TForm3.BitBtn4Click(Sender: TObject);
begin
  Form1.Show;
  Form3.Visible := False;
end;

```

```

procedure TForm3.BitBtn6Click(Sender: TObject);

```

```

While not Table1.Eof Do
begin
  If (Edit6.Text = Table1PRODUCT_CODE.Text) Then
  begin
    Table1.Refresh;
    Table1.Edit;
    Table1PRODUCTPEACE.Value := ((Table1PRODUCTPEACE.Value) -
(StrtoInt(Edit7.Text)));
    Table1.UpdateRecord;
    Table1.Refresh;
  end;
  Table1.Next;
end;

Table1.Refresh;
Table1.First;
While not Table1.Eof Do
begin
  Table1.Next;
end;

If (Table1.Eof) Then
begin
  ListNo := (Table1LISTNO.Value + 1);
end;

Table2.Append;
Table2LISTNO.Value := ListNo;
Table2PRODUCTCODE.Text := Edit6.Text;
Table2PRODUCTNAME.Text := Edit3.Text;
Table2CUSTOMERNAME.Text := Edit2.Text;
Table2PIECEOFSALE.Text := Edit7.Text;
Table2COMPANYNAME.Value := Edit1.Text;
Table2SALINGCHARGER.Text := Edit8.Text;
Table2SALINGPRICE.Value := StrtoInt(Edit5.Text);
Table2TOTALPRICE.Value := StrtoInt(Edit4.Text);
Table2SALINGDATE.Value := DateTimePicker1.Date;
Table2.Post;
Table1.Refresh;
Edit1.Text := "; Edit5.Text := ";
Edit2.Text := "; Edit6.Text := ";
Edit3.Text := "; Edit8.Text := ";
Edit4.Text := "; Edit8.Text := ";
ShowMessage('PRODUCT CASH SELLING PROCESS IS COMPLETED!');
end;

end;

```

```

Table1PRODUCT_NAME: TStringField;
Table1PRODUCTPEACE: TIntegerField;
Table1PRODUCTSIZE: TStringField;
Table1CONTROLUNIT: TIntegerField;
Table1BUYINGPRICE: TIntegerField;
Table1BUYINGPLACE: TStringField;
Table1CELLINGPRICE: TIntegerField;
Table1BUYINGDATE: TDateField;
Table1SPECIALNOTES: TStringField;
Table2BUYINGPRICE: TIntegerField;
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure Edit6Change(Sender: TObject);
procedure Edit3Change(Sender: TObject);
procedure Edit7Change(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form3: TForm3;

implementation

uses Unit1, Unit6;

{$R *.dfm}

procedure TForm3.BitBtn1Click(Sender: TObject);
begin
  Form1.Show;
  Form3.Visible := False;
end;

procedure TForm3.BitBtn2Click(Sender: TObject);
Var
  Aranan : Boolean;
  ListNo : Integer;
begin
  Aranan := False;

  If MessageDlg('ARE YOU SURE THAT YOU WANT TO DO THIS CELLING
PROCESS!..',
mtconfirmation, [mbYES, mbNO], 0) = mrYES Then
begin
  Table1.First;

```



unit Unit4;

interface

uses

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

type

```
TForm4 = class(TForm)
  Label1: TLabel;
  Label2: TLabel;
  Label5: TLabel;
  Label8: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  BitBtn1: TBitBtn;
  Edit1: TEdit;
  Edit2: TEdit;
  Edit5: TEdit;
  GroupBox1: TGroupBox;
  Panel1: TPanel;
  Edit6: TEdit;
  Edit7: TEdit;
  BitBtn2: TBitBtn;
  DateTimePicker1: TDateTimePicker;
  BitBtn3: TBitBtn;
  BitBtn4: TBitBtn;
  Edit3: TEdit;
  Edit4: TEdit;
  Label9: TLabel;
  Edit8: TEdit;
  Edit9: TEdit;
  Label10: TLabel;
  BitBtn5: TBitBtn;
  BitBtn6: TBitBtn;
  Label11: TLabel;
  Edit10: TEdit;
  Label12: TLabel;
  Edit11: TEdit;
  Table1: TTable;
  Table1LISTNO: TIntegerField;
  Table1PRODUCT_CODE: TStringField;
  Table1PRODUCT_NAME: TStringField;
  Table1PRODUCTPEACE: TIntegerField;
  Table1PRODUCTSIZE: TStringField;
  Table1CONTROLUNIT: TIntegerField;
  Table1BUYINGPRICE: TIntegerField;
```



```
procedure TForm4.Edit8Change(Sender: TObject);
Var
Rest : Integer;
begin
If (Edit8.Text <> "") Then
begin
Rest := ((StrToInt(Edit4.Text)) - (StrToInt(Edit8.Text)));
Edit9.Text := InttoStr(Rest);
end;
end;
```

```
procedure TForm4.BitBtn6Click(Sender: TObject);
begin
Form7.Show;
Form4.Visible := False;
end;

end.
```

```

Table2.Refresh;
Table2.First;
While not Table2.Eof Do
begin
    Table2.Next;
end;

If (Table2.Eof) Then
begin
    ListNo := (Table2LISTNUMBER.Value + 1);
end;

Table2.Append;
Table2LISTNUMBER.Value := ListNo;
Table2BUYINGPRICE.Value := BPrice;
Table2COMPANYNAME.Text := Edit1.Text;
Table2CUSTOMERNAME.Text := Edit2.Text;
Table2PRODUCTNAME.Text := Edit3.Text;
Table2PRODUCTCODE.Text := Edit6.Text;
Table2SAILINGCHARGER.Text := Edit10.Text;
Table2CUSTOMERCODE.Text := Edit11.Text;
Table2CASHPAYMENT.Value := StrToInt(Edit8.Text);
Table2TOTALPAYMENT.Value := StrToInt(Edit8.Text);
Table2TOTALPRICE.Value := StrToInt(Edit4.Text);
Table2SAILINGPRICE.Value := StrToInt(Edit5.Text);
Table2PIECEOFSAILING.Value := StrToInt(Edit7.Text);
Table2REMAININGPAYMENT.Value := StrToInt(Edit9.Text);
Table2SAILINGDATE.Value := DateTimePicker1.Date;
Table2Result.Text := 'OPEN';
Table2.Post;
Table2.Refresh;

Edit1.Text := ''; Edit6.Text := '';
Edit2.Text := ''; Edit7.Text := '';
Edit3.Text := ''; Edit8.Text := '';
Edit4.Text := ''; Edit9.Text := '';
Edit5.Text := ''; Edit10.Text := ''; Edit11.Text := '';
ShowMessage('PRODUCT INSTALLMENT SAILING PROCESS IS COMPLETED!');
end;
end;

end;

procedure TForm4.BitBtn4Click(Sender: TObject);
begin
    Form1.Show;
    Form4.Visible := False;
end;

```

```

    Edit6.Text := Table1PRODUCT_CODE.Text;
    Edit5.Text := IntToStr(Table1CELLINGPRICE.Value);
end;
Table1.Next;
end;

end;

procedure TForm4.BitBtn2Click(Sender: TObject);
Var
ListNo, BPrice : Integer;
Aranan, AyniKod : Boolean;
begin
Aranan := False; AyniKod := False; ListNo := 0;

If MessageDlg('ARE YOU SURE THAT YOU WANT TO DO THIS CELLING
PROCESS!..',
mtconfirmation, [mbYES, mbNO], 0) = mrYES Then
begin

Table2.First;
While not Table2.Eof Do
begin
If (Edit11.Text = Table2CUSTOMERCODE.Text) Then
begin
AyniKod := True;
ShowMessage('SAME CUSTOMER CODE IS ENTERED, PLEASE CHANGE IT!');
end;
Table2.Next;
end;

If (AyniKod = False) Then
begin

Table1.Refresh;
Table1.First;
While not Table1.Eof Do
begin
If (Edit6.Text = Table1PRODUCT_CODE.Text) Then
begin
Table1.Refresh;
Table1.Edit;
BPrice := Table1BUYINGPRICE.Value;
Table1PRODUCTPEACE.Value := ((Table1PRODUCTPEACE.Value) -
(StrToInt(Edit7.Text)));
Table1.Post;
Table1.Refresh;
end;
Table1.Next;
end;
end;

```

```

    procedure Edit8Change(Sender: TObject);
    procedure BitBtn6Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form4: TForm4;

implementation

uses Unit1, Unit7;

{$R *.dfm}

procedure TForm4.Edit7Change(Sender: TObject);
Var
Price : Integer;
begin
    If (Edit7.Text <> "") Then
        begin
            Price := ((StrToInt(Edit7.Text)) * (StrToInt(Edit5.Text)));
            Edit4.Text := IntToStr(Price);
        end;
    end;

procedure TForm4.Edit6Change(Sender: TObject);
begin
Table1.First;
    While not Table1.Eof Do
        begin
            If (Table1PRODUCT_CODE.Text = Edit6.Text) Then
                begin
                    Edit3.Text := Table1PRODUCT_NAME.Text;
                    Edit5.Text := IntToStr(Table1CELLINGPRICE.Value);
                end;
            Table1.Next;
        end;

end;

procedure TForm4.Edit3Change(Sender: TObject);
begin
Table1.First;
    While not Table1.Eof Do
        begin
            If (Table1PRODUCT_NAME.Text = Edit3.Text) Then
                begin

```

Table1BUYINGPLACE: TStringField;
 Table1CELLINGPRICE: TIntegerField;
 Table1BUYINGDATE: TDateField;
 Table1SPECIALNOTES: TStringField;
 Table2: TTable;
 Table2LISTNUMBER: TIntegerField;
 Table2CUSTOMERCODE: TStringField;
 Table2CUSTOMERNAME: TStringField;
 Table2PRODUCTCODE: TStringField;
 Table2PRODUCTNAME: TStringField;
 Table2COMPANYNAME: TStringField;
 Table2SAILINGCHARGER: TStringField;
 Table2SAILINGDATE: TDateField;
 Table2PIECEOFSAILING: TIntegerField;
 Table2SAILINGPRICE: TIntegerField;
 Table2TOTALPRICE: TIntegerField;
 Table2TOTALPAYMENT: TIntegerField;
 Table2REMAININGPAYMENT: TIntegerField;
 Table2CASHPAYMENT: TIntegerField;
 Table2PAYMENT1: TIntegerField;
 Table2DATE1: TDateField;
 Table2PAYMENT2: TIntegerField;
 Table2DATE2: TDateField;
 Table2PAYMENT3: TIntegerField;
 Table2DATE3: TDateField;
 Table2PAYMENT4: TIntegerField;
 Table2DATE4: TDateField;
 Table2PAYMENT5: TIntegerField;
 Table2DATE5: TDateField;
 Table2PAYMENT6: TIntegerField;
 Table2DATE6: TDateField;
 Table2PAYMENT7: TIntegerField;
 Table2DATE7: TDateField;
 Table2PAYMENT8: TIntegerField;
 Table2DATE8: TDateField;
 Table2PAYMENT9: TIntegerField;
 Table2DATE9: TDateField;
 Table2PAYMENT10: TIntegerField;
 Table2DATE10: TDateField;
 Table2PAYMENT11: TIntegerField;
 Table2DATE11: TDateField;
 Table2PAYMENT12: TIntegerField;
 Table2DATE12: TDateField;
 Table2RESULT: TStringField;
 Table2BUYINGPRICE: TIntegerField;
 procedure Edit7Change(Sender: TObject);
 procedure Edit6Change(Sender: TObject);
 procedure Edit3Change(Sender: TObject);
 procedure BitBtn2Click(Sender: TObject);
 procedure BitBtn4Click(Sender: TObject);

nit Unit5;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Buttons, ExtCtrls;

type

TForm5 = class(TForm)
 GroupBox1: TGroupBox;
 GroupBox2: TGroupBox;
 Panel1: TPanel;
 Panel2: TPanel;
 BitBtn1: TBitBtn;
 BitBtn2: TBitBtn;
 procedure BitBtn1Click(Sender: TObject);
 procedure BitBtn2Click(Sender: TObject);
private
 { Private declarations }
public
 { Public declarations }
end;

var

Form5: TForm5;

implementation

uses Unit3, Unit4;

{ \$R *.dfm }

procedure TForm5.BitBtn1Click(Sender: TObject);
begin
 Form3.Show;
 Form5.Visible := False;
end;

procedure TForm5.BitBtn2Click(Sender: TObject);
begin
 Form4.Show;
 Form5.Visible := False;
end;

end.

unit Unit7;

interface

uses

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

type

TForm7 = class(TForm)
 PageControl1: TPageControl;
 TabSheet1: TTabSheet;
 TabSheet2: TTabSheet;
 TabSheet3: TTabSheet;
 TabSheet4: TTabSheet;
 DataSource1: TDataSource;
 DBGrid1: TDBGrid;
 GroupBox1: TGroupBox;
 Label1: TLabel;
 Edit1: TEdit;
 BitBtn1: TBitBtn;
 BitBtn9: TBitBtn;
 DBGrid2: TDBGrid;
 DBGrid3: TDBGrid;
 GroupBox2: TGroupBox;
 Label2: TLabel;
 Label3: TLabel;
 Label4: TLabel;
 DateTimePicker1: TDateTimePicker;
 DateTimePicker2: TDateTimePicker;
 BitBtn2: TBitBtn;
 BitBtn5: TBitBtn;
 BitBtn3: TBitBtn;
 BitBtn7: TBitBtn;
 BitBtn4: TBitBtn;
 DataSource2: TDataSource;
 DBNavigator2: TDBNavigator;
 DBNavigator1: TDBNavigator;
 DBNavigator3: TDBNavigator;
 BitBtn6: TBitBtn;
 Table1: TTable;
 Table2: TTable;
 Table1LISTNUMBER: TIntegerField;
 Table1CUSTOMERCODE: TStringField;
 Table1CUSTOMERNAME: TStringField;
 Table1PRODUCTCODE: TStringField;
 Table1PRODUCTNAME: TStringField;
 Table1COMPANYNAME: TStringField;
 Table1SAILINGCHARGER: TStringField;


```

Table2PAYMENT12.Value := Table1PAYMENT12.Value;
Table2DATE12.Value := Table1DATE12.Value;
Table2RESULT.Text := Table1RESULT.Value;
Table2REMAININGPAYMENT.Value := Table1REMAININGPAYMENT.Value;
Table2.Post;
Table2.Refresh;
end;
Table1.Next;
end;

```

```

If (Aranan = False ) Then
begin
  ShowMessage('THE RECORD IS NOT FOUNDED!');
end;

```

```

end;

```

```

procedure TForm7.BitBtn6Click(Sender: TObject);
begin
  Form4.Show;
  Form7.Visible := False;
end;

```

```

procedure TForm7.BitBtn10Click(Sender: TObject);
begin
  Form13.QuickRep1.Preview;
end;

```

```

end.

```

```

Table2.Refresh;
Table2.First;
While not (Table2.IsEmpty) Do
begin
Table2.Delete;
Table2.Next;
end;

```

```

Table2.Refresh;
Table1.First;
While not Table1.Eof Do
begin
If ((StrToDate(Table1.SAILINGDATE.Text) > DateTimePicker1.Date) AND
((StrToDate(Table1.SAILINGDATE.Text)) < DateTimePicker2.Date)) Then
begin
Aranan := True;
Table2.Append;
Table2.LISTNUMBER.Value := Table1.LISTNUMBER.Value;
Table2.PRODUCTCODE.Text := Table1.PRODUCTCODE.Text;
Table2.PRODUCTNAME.Text := Table1.PRODUCTNAME.Text;
Table2.CUSTOMERNAME.Text := Table1.CUSTOMERNAME.Text;
Table2.PIECEOFSAILING.Text := Table1.PIECEOFSAILING.Text;
Table2.COMPANYNAME.Value := Table1.COMPANYNAME.Value;
Table2.SAILINGCHARGER.Value := Table1.SAILINGCHARGER.Value;
Table2.SAILINGPRICE.Value := Table1.SAILINGPRICE.Value;
Table2.TOTALPRICE.Value := Table1.TOTALPRICE.Value;
Table2.TOTALPAYMENT.Value := Table1.TOTALPAYMENT.Value;
Table2.SAILINGDATE.Value := Table1.SAILINGDATE.Value;
Table2.PAYMENT1.Value := Table1.PAYMENT1.Value;
Table2.DATE1.Value := Table1.DATE1.Value;
Table2.PAYMENT2.Value := Table1.PAYMENT2.Value;
Table2.DATE2.Value := Table1.DATE2.Value;
Table2.PAYMENT3.Value := Table1.PAYMENT3.Value;
Table2.DATE3.Value := Table1.DATE3.Value;
Table2.PAYMENT4.Value := Table1.PAYMENT4.Value;
Table2.DATE4.Value := Table1.DATE4.Value;
Table2.PAYMENT5.Value := Table1.PAYMENT5.Value;
Table2.DATE5.Value := Table1.DATE5.Value;
Table2.PAYMENT6.Value := Table1.PAYMENT6.Value;
Table2.DATE6.Value := Table1.DATE6.Value;
Table2.PAYMENT7.Value := Table1.PAYMENT7.Value;
Table2.DATE7.Value := Table1.DATE7.Value;
Table2.PAYMENT8.Value := Table1.PAYMENT8.Value;
Table2.DATE8.Value := Table1.DATE8.Value;
Table2.PAYMENT9.Value := Table1.PAYMENT9.Value;
Table2.DATE9.Value := Table1.DATE9.Value;
Table2.PAYMENT10.Value := Table1.PAYMENT10.Value;
Table2.DATE10.Value := Table1.DATE10.Value;
Table2.PAYMENT11.Value := Table1.PAYMENT11.Value;
Table2.DATE11.Value := Table1.DATE11.Value;

```

```

Table2TOTALPRICE.Value := Table1TOTALPRICE.Value;
Table2TOTALPAYMENT.Value := Table1TOTALPAYMENT.Value;
Table2SAILINGDATE.Value := Table1SAILINGDATE.Value;
Table2PAYMENT1.Value := Table1PAYMENT1.Value;
Table2DATE1.Value := Table1DATE1.Value;
Table2PAYMENT2.Value := Table1PAYMENT2.Value;
Table2DATE2.Value := Table1DATE2.Value;
Table2PAYMENT3.Value := Table1PAYMENT3.Value;
Table2DATE3.Value := Table1DATE3.Value;
Table2PAYMENT4.Value := Table1PAYMENT4.Value;
Table2DATE4.Value := Table1DATE4.Value;
Table2PAYMENT5.Value := Table1PAYMENT5.Value;
Table2DATE5.Value := Table1DATE5.Value;
Table2PAYMENT6.Value := Table1PAYMENT6.Value;
Table2DATE6.Value := Table1DATE6.Value;
Table2PAYMENT7.Value := Table1PAYMENT7.Value;
Table2DATE7.Value := Table1DATE7.Value;
Table2PAYMENT8.Value := Table1PAYMENT8.Value;
Table2DATE8.Value := Table1DATE8.Value;
Table2PAYMENT9.Value := Table1PAYMENT9.Value;
Table2DATE9.Value := Table1DATE9.Value;
Table2PAYMENT10.Value := Table1PAYMENT10.Value;
Table2DATE10.Value := Table1DATE10.Value;
Table2PAYMENT11.Value := Table1PAYMENT11.Value;
Table2DATE11.Value := Table1DATE11.Value;
Table2PAYMENT12.Value := Table1PAYMENT12.Value;
Table2DATE12.Value := Table1DATE12.Value;
Table2RESULT.Text := Table1RESULT.Value;
Table2REMAININGPAYMENT.Value := Table1REMAININGPAYMENT.Value;

```

```

Table2.Post;
Table2.Refresh;
end;
Table1.Next;
end;

```

```

If (Aranan = False ) Then
begin
  ShowMessage('THE RECORD IS NOT FOUNDED!');
end;

```

```
end;
```

```

procedure TForm7.BitBtn2Click(Sender: TObject);
Var
Aranan : Boolean;
begin
Aranan := False;

```

```
Form7.Visible := False;  
end;
```

```
procedure TForm7.BitBtn3Click(Sender: TObject);  
begin  
Form2.Show;  
Form7.Visible := False;  
end;
```

```
procedure TForm7.BitBtn7Click(Sender: TObject);  
begin  
Form3.Show;  
Form7.Visible := False;  
end;
```

```
procedure TForm7.BitBtn8Click(Sender: TObject);  
begin  
Form4.Show;  
Form7.Visible := False;  
end;
```

```
procedure TForm7.BitBtn1Click(Sender: TObject);  
Var  
Aranan : Boolean;  
begin  
Aranan := False;  
Table2.Refresh;  
Table2.First;  
While not (Table2.IsEmpty) Do  
begin  
Table2.Delete;  
Table2.Next;  
end;
```

```
Table2.Refresh;  
Table1.First;  
While not Table1.Eof Do  
begin  
If (Table1.CUSTOMERNAME.Text = Edit1.Text) Then  
begin  
Aranan := True;  
Table2.Append;  
Table2.LISTNUMBER.Value := Table1.LISTNUMBER.Value;  
Table2.PRODUCTCODE.Text := Table1.PRODUCTCODE.Text;  
Table2.PRODUCTNAME.Text := Table1.PRODUCTNAME.Text;  
Table2.CUSTOMERNAME.Text := Table1.CUSTOMERNAME.Text;  
Table2.PIECEOFSAILING.Text := Table1.PIECEOFSAILING.Text;  
Table2.COMPANYNAME.Value := Table1.COMPANYNAME.Value;  
Table2.SAILINGCHARGER.Value := Table1.SAILINGCHARGER.Value;  
Table2.SAILINGPRICE.Value := Table1.SAILINGPRICE.Value;
```



```

Table2PAYMENT3: TIntegerField;
Table2DATE3: TDateField;
Table2PAYMENT4: TIntegerField;
Table2DATE4: TDateField;
Table2PAYMENT5: TIntegerField;
Table2DATE5: TDateField;
Table2PAYMENT6: TIntegerField;
Table2DATE6: TDateField;
Table2PAYMENT7: TIntegerField;
Table2DATE7: TDateField;
Table2PAYMENT8: TIntegerField;
Table2DATE8: TDateField;
Table2PAYMENT9: TIntegerField;
Table2DATE9: TDateField;
Table2PAYMENT10: TIntegerField;
Table2DATE10: TDateField;
Table2PAYMENT11: TIntegerField;
Table2DATE11: TDateField;
Table2PAYMENT12: TIntegerField;
Table2DATE12: TDateField;
Table2RESULT: TStringField;
Table1BUYINGPRICE: TIntegerField;
Table2BUYINGPRICE: TIntegerField;
BitBtn10: TBitBtn;
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn7Click(Sender: TObject);
procedure BitBtn8Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
procedure BitBtn10Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form7: TForm7;

implementation

uses Unit1, Unit2, Unit3, Unit4, Unit13;

{$R *.dfm}

procedure TForm7.BitBtn4Click(Sender: TObject);
begin
  Form1.Show;

```

Table1SAILINGDATE: TDateField;
 Table1PIECEOFSAILING: TIntegerField;
 Table1SAILINGPRICE: TIntegerField;
 Table1TOTALPRICE: TIntegerField;
 Table1TOTALPAYMENT: TIntegerField;
 Table1REMAININGPAYMENT: TIntegerField;
 Table1CASHPAYMENT: TIntegerField;
 Table1PAYMENT1: TIntegerField;
 Table1DATE1: TDateField;
 Table1PAYMENT2: TIntegerField;
 Table1DATE2: TDateField;
 Table1PAYMENT3: TIntegerField;
 Table1DATE3: TDateField;
 Table1PAYMENT4: TIntegerField;
 Table1DATE4: TDateField;
 Table1PAYMENT5: TIntegerField;
 Table1DATE5: TDateField;
 Table1PAYMENT6: TIntegerField;
 Table1DATE6: TDateField;
 Table1PAYMENT7: TIntegerField;
 Table1DATE7: TDateField;
 Table1PAYMENT8: TIntegerField;
 Table1DATE8: TDateField;
 Table1PAYMENT9: TIntegerField;
 Table1DATE9: TDateField;
 Table1PAYMENT10: TIntegerField;
 Table1DATE10: TDateField;
 Table1PAYMENT11: TIntegerField;
 Table1DATE11: TDateField;
 Table1PAYMENT12: TIntegerField;
 Table1DATE12: TDateField;
 Table1RESULT: TStringField;
 Table2LISTNUMBER: TIntegerField;
 Table2CUSTOMERCODE: TStringField;
 Table2CUSTOMERNAME: TStringField;
 Table2PRODUCTCODE: TStringField;
 Table2PRODUCTNAME: TStringField;
 Table2COMPANYNAME: TStringField;
 Table2SAILINGCHARGER: TStringField;
 Table2SAILINGDATE: TDateField;
 Table2PIECEOFSAILING: TIntegerField;
 Table2SAILINGPRICE: TIntegerField;
 Table2TOTALPRICE: TIntegerField;
 Table2TOTALPAYMENT: TIntegerField;
 Table2REMAININGPAYMENT: TIntegerField;
 Table2CASHPAYMENT: TIntegerField;
 Table2PAYMENT1: TIntegerField;
 Table2DATE1: TDateField;
 Table2PAYMENT2: TIntegerField;
 Table2DATE2: TDateField;

```

unit Unit8;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ExtCtrls;

type
  TForm8 = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    Panel1: TPanel;
    BitBtn1: TBitBtn;
    Panel2: TPanel;
    BitBtn2: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form8: TForm8;

implementation

uses Unit6, Unit7;

{$R *.dfm}

procedure TForm8.BitBtn1Click(Sender: TObject);
begin
  Form6.Show;
  Form8.Visible := False;
end;

procedure TForm8.BitBtn2Click(Sender: TObject);
begin
  Form7.Show;
  Form8.Visible := False;
end;

end.

```


unit Unit9;

interface

uses

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

type

```
TForm9 = class(TForm)
  DataSource1: TDataSource;
  PageControl1: TPageControl;
  TabSheet1: TTabSheet;
  TabSheet2: TTabSheet;
  TabSheet3: TTabSheet;
  DBGrid1: TDBGrid;
  BitBtn6: TBitBtn;
  BitBtn4: TBitBtn;
  BitBtn3: TBitBtn;
  BitBtn1: TBitBtn;
  Edit1: TEdit;
  Label1: TLabel;
  Label2: TLabel;
  BitBtn2: TBitBtn;
  BitBtn5: TBitBtn;
  Label3: TLabel;
  Table2: TTable;
  Table3: TTable;
  Table2LISTNO: TIntegerField;
  Table2CUSTOMERNAME: TStringField;
  Table2PRODUCTCODE: TStringField;
  Table2PRODUCTNAME: TStringField;
  Table2PIECEOFSALE: TIntegerField;
  Table2SALEPRICE: TIntegerField;
  Table2TOTALPRICE: TIntegerField;
  Table2COMPANYNAME: TStringField;
  Table2SALECHARGER: TStringField;
  Table2SALEDATE: TDateField;
  Table3LISTNUMBER: TIntegerField;
  Table3CUSTOMERCODE: TStringField;
  Table3CUSTOMERNAME: TStringField;
  Table3PRODUCTCODE: TStringField;
  Table3PRODUCTNAME: TStringField;
  Table3COMPANYNAME: TStringField;
  Table3SALECHARGER: TStringField;
  Table3SALEDATE: TDateField;
  Table3PIECEOFSALE: TIntegerField;
  Table3SALEPRICE: TIntegerField;
  Table3TOTALPRICE: TIntegerField;
  Table3TOTALPAYMENT: TIntegerField;
```

Table1INSTALLMENTSAILING.Value := InttoStr(TtlInstallment);
Table1.Post;

end;

end.

```

While not Table1.Eof Do
begin
  TtlBuying      := (StrToInt64(Table1BUYINGPRICE.Text) + TtlBuying);
  Cost2         := (StrToInt64(Table1BUYINGPRICE.Text) +
(Trunc(StrToInt64(Table1BUYINGPRICE.Text) / 100) * (StrToInt(Edit1.Text))));
  TtlCost       := (Cost2 + TtlBuying);
  Table1.Edit;
  Table1COSTPRICE.Text := InttoStr(Cost2);
  Table1.Post;
  Table1.Next;
end;

```

```

Table1.First;
While not (Table1.Eof) Do
begin
  If (Table1CASHSAILING.Text <> "") Then
  begin
    TtlCash      := (StrToInt64(Table1CASHSAILING.Text) + TtlCash);
    Table1.Edit;
    Table1PROFITLIMIT.Text := InttoStr(StrToInt64(Table1CASHSAILING.Text) -
StrToInt64(Table1COSTPRICE.Text));
    TtlProfit     := (StrToInt64(Table1PROFITLIMIT.Value) + (TtlProfit));
    Table1.Post;
  end;

```

```

  If (Table1INSTALLMENTSAILING.Text <> "") Then
  begin
    Table1.Edit;
    TtlInstallment := (StrToInt64(Table1INSTALLMENTSAILING.Text) +
TtlInstallment);
    Table1PROFITLIMIT.Text :=
InttoStr(StrToInt64(Table1INSTALLMENTSAILING.Text) -
StrToInt64(Table1COSTPRICE.Text));
    TtlProfit      := (StrToInt64(Table1PROFITLIMIT.Text) + (TtlProfit));
    Table1.Post;
  end;
  Table1.Next;
end;

```

```

Table1.Append;
Table1.Post;
Table1.Append;
Table1.Post;

```

```

Table1.Append;
Table1PRODUCTCODE.Text      := 'TOTALS';
Table1BUYINGPRICE.Text      := InttoStr(TtlBuying);
Table1COSTPRICE.Text        := InttoStr(TtlCost);
Table1PROFITLIMIT.Text      := InttoStr(TtlProfit);
Table1CASHSAILING.Value     := InttoStr(TtlCash);

```

```

begin
Cost := 0; TtlCash := 0; TtlInstallment := 0; Profit := 0; TtlProfit := 0; TtlBuying := 0;
TtlCost := 0; ListNo := 1;

```

```

Table1.Refresh;
Table1.First;
While not (Table1.IsEmpty) Do
begin
Table1.Delete;
Table1.Next;
end;

```

```

Table2.First;
While not Table2.Eof Do
begin
Table1.Append;
Table1LISTNO.Value := ListNo;
ListNo := Table1LISTNO.Value + 1;
Table1PRODUCTNAME.Text := Table2PRODUCTNAME.Text;
Table1PRODUCTCODE.Text := Table2PRODUCTCODE.Text;
Table1CASHSAILING.Text := Table2TOTALPRICE.Text;
Table1BUYINGPRICE.Text := Table2BUYINGPRICE.Text;
Table1SAILINGDATE.Value := Table2SALINGDATE.Value;
Table1SAILINGCHARGER.Value := Table2SALINGCHARGER.Value;
Table1.Post;
Table2.Next;
end;
Table1.Refresh;

```

```

Table3.First;
While not Table3.Eof Do
begin
Table1.Append;
Table1LISTNO.Value := ListNo;
ListNo := Table1LISTNO.Value + 1;
Table1PRODUCTNAME.Text := Table3PRODUCTNAME.Text;
Table1PRODUCTCODE.Text := Table3PRODUCTCODE.Text;
Table1BUYINGPRICE.Text := Table3BUYINGPRICE.Text;
Table1INSTALLMENTSAILING.Text := Table3TOTALPRICE.Text;
Table1SAILINGDATE.Value := Table3SAILINGDATE.Value;
Table1SAILINGCHARGER.Value := Table3SAILINGCHARGER.Value;
Table1.Post;
Table3.Next;
end;
Table1.Refresh;

```

```

Table1.First;

```

```

Table1SAILINGCHARGER: TStringField;
Table1SAILINGDATE: TDateField;
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form9: TForm9;

implementation

uses Unit1, Unit2, Unit3, Unit4;

{$R *.dfm}

procedure TForm9.BitBtn4Click(Sender: TObject);
begin
  Form1.Show;
  Form9.Visible := False;
end;

procedure TForm9.BitBtn3Click(Sender: TObject);
begin
  Form2.Show;
  Form9.Visible := False;
end;

procedure TForm9.BitBtn1Click(Sender: TObject);
begin
  Form3.Show;
  Form9.Visible := False;
end;

procedure TForm9.BitBtn6Click(Sender: TObject);
begin
  Form4.Show;
  Form9.Visible := False;
end;

procedure TForm9.BitBtn5Click(Sender: TObject);
Var
  Cost, TtlCash, TtlInstallment, Profit, TtlProfit, ListNo, ListNo2, Cost2, TtlCost, TtlBuying :
  Int64;

```


Table3REMAININGPAYMENT: TIntegerField;
 Table3CASHPAYMENT: TIntegerField;
 Table3PAYMENT1: TIntegerField;
 Table3DATE1: TDateField;
 Table3PAYMENT2: TIntegerField;
 Table3DATE2: TDateField;
 Table3PAYMENT3: TIntegerField;
 Table3DATE3: TDateField;
 Table3PAYMENT4: TIntegerField;
 Table3DATE4: TDateField;
 Table3PAYMENT5: TIntegerField;
 Table3DATE5: TDateField;
 Table3PAYMENT6: TIntegerField;
 Table3DATE6: TDateField;
 Table3PAYMENT7: TIntegerField;
 Table3DATE7: TDateField;
 Table3PAYMENT8: TIntegerField;
 Table3DATE8: TDateField;
 Table3PAYMENT9: TIntegerField;
 Table3DATE9: TDateField;
 Table3PAYMENT10: TIntegerField;
 Table3DATE10: TDateField;
 Table3PAYMENT11: TIntegerField;
 Table3DATE11: TDateField;
 Table3PAYMENT12: TIntegerField;
 Table3DATE12: TDateField;
 Table3RESULT: TStringField;
 Table4: TTable;
 Table4LISTNO: TIntegerField;
 Table4PRODUCT_CODE: TStringField;
 Table4PRODUCT_NAME: TStringField;
 Table4PRODUCTPEACE: TIntegerField;
 Table4PRODUCTSIZE: TStringField;
 Table4CONTROLUNIT: TIntegerField;
 Table4BUYINGPRICE: TIntegerField;
 Table4BUYINGPLACE: TStringField;
 Table4CELLINGPRICE: TIntegerField;
 Table4BUYINGDATE: TDateField;
 Table4SPECIALNOTES: TStringField;
 Table2BUYINGPRICE: TIntegerField;
 Table3BUYINGPRICE: TIntegerField;
 Table1: TTable;
 Table1LISTNO: TIntegerField;
 Table1PRODUCTNAME: TStringField;
 Table1PRODUCTCODE: TStringField;
 Table1BUYINGPRICE: TStringField;
 Table1COSTPRICE: TStringField;
 Table1CASHSAILING: TStringField;
 Table1INSTALLMENTSAILING: TStringField;
 Table1PROFITLIMIT: TStringField;

nit Unit10;

interface

uses

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

type

TForm10 = class(TForm)
 Panel1: TPanel;
 Edit1: TEdit;
 Edit2: TEdit;
 Label1: TLabel;
 Label2: TLabel;
 Label3: TLabel;
 Label4: TLabel;
 BitBtn1: TBitBtn;
 DataSource1: TDataSource;
 DBGrid1: TDBGrid;
 DBNavigator1: TDBNavigator;
 BitBtn2: TBitBtn;
 BitBtn3: TBitBtn;
 BitBtn4: TBitBtn;
 GroupBox1: TGroupBox;
 Edit3: TEdit;
 Label5: TLabel;
 BitBtn5: TBitBtn;
 BitBtn6: TBitBtn;
 Label6: TLabel;
 Label7: TLabel;
 GroupBox2: TGroupBox;
 Label8: TLabel;
 Edit4: TEdit;
 GroupBox3: TGroupBox;
 CheckBox1: TCheckBox;
 CheckBox2: TCheckBox;
 CheckBox3: TCheckBox;
 CheckBox4: TCheckBox;
 CheckBox5: TCheckBox;
 CheckBox6: TCheckBox;
 CheckBox7: TCheckBox;
 CheckBox8: TCheckBox;
 CheckBox9: TCheckBox;
 CheckBox10: TCheckBox;
 CheckBox11: TCheckBox;
 CheckBox12: TCheckBox;
 Label9: TLabel;
 DateTimePicker1: TDateTimePicker;

```
begin
  Table1PAYMENT3.Value := StrToInt(Edit4.Text);
  Table1DATE3.Text := DatetoStr(DateTimePicker1.Date);
end;
```

```
If (CheckBox4.Checked = True) Then
begin
  Table1PAYMENT4.Value := StrToInt(Edit4.Text);
  Table1DATE4.Text := DatetoStr(DateTimePicker1.Date);
end;
```

```
If (CheckBox5.Checked = True) Then
begin
  Table1PAYMENT5.Value := StrToInt(Edit4.Text);
  Table1DATE5.Text := DatetoStr(DateTimePicker1.Date);
end;
```

```
If (CheckBox6.Checked = True) Then
begin
  Table1PAYMENT6.Value := StrToInt(Edit4.Text);
  Table1DATE6.Text := DatetoStr(DateTimePicker1.Date);
end;
```

```
If (CheckBox7.Checked = True) Then
begin
  Table1PAYMENT7.Value := StrToInt(Edit4.Text);
  Table1DATE7.Text := DatetoStr(DateTimePicker1.Date);
end;
```

```
If (CheckBox8.Checked = True) Then
begin
  Table1PAYMENT8.Value := StrToInt(Edit4.Text);
  Table1DATE8.Text := DatetoStr(DateTimePicker1.Date);
end;
```

```
If (CheckBox9.Checked = True) Then
begin
  Table1PAYMENT9.Value := StrToInt(Edit4.Text);
  Table1DATE9.Text := DatetoStr(DateTimePicker1.Date);
end;
```

```
If (CheckBox10.Checked = True) Then
begin
  Table1PAYMENT10.Value := StrToInt(Edit4.Text);
  Table1DATE10.Text := DatetoStr(DateTimePicker1.Date);
end;
```

```
If (CheckBox11.Checked = True) Then
begin
  Table1PAYMENT11.Value := StrToInt(Edit4.Text);
```

end;

```
If (Aranan = False ) Then
begin
  ShowMessage('THE RECORD IS NOT FOUNDED!');
end;
```

end;

```
procedure TForm10.BitBtn5Click(Sender: TObject);
Var
ListNo : Boolean;
begin
ListNo := False;
```

```
If MessageDlg('ARE YOU SURE THAT YOU WANT TO MAKE PAYMENT EVENT
FROM THE TABLE!..',
mtconfirmation, [mbYES, mbNO], 0) = mrYES Then
begin
  Table1.First;
  While not Table1.Eof Do
  begin
    If (Table1LISTNUMBER.Value = StrToInt(Edit3.Text)) Then
    begin
      ListNo := True;
      Table1.Refresh;
      Table1.Edit;
      Table1REMAININGPAYMENT.Value := ((Table1REMAININGPAYMENT.Value) -
(StrToInt(Edit4.Text)));
```

```
    If (Table1REMAININGPAYMENT.Value <= 0) Then
    begin
      Table1RESULT.Text := 'CLOSED'
    end;
```

```
  If (CheckBox1.Checked = True) Then
  begin
    Table1PAYMENT1.Value := StrToInt(Edit4.Text);
    Table1DATE1.Text := DatetoStr(DateTimePicker1.Date);
  end;
```

```
  If (CheckBox2.Checked = True) Then
  begin
    Table1PAYMENT2.Value := StrToInt(Edit4.Text);
    Table1DATE2.Text := DatetoStr(DateTimePicker1.Date);
  end;
```

```
  If (CheckBox3.Checked = True) Then
```

```

While not Table1.Eof Do
begin
  If (Table1.CUSTOMERNAME.Text = Edit2.Text) Then
  begin
    Aranan := True;
    Table2.Append;
    Table2.LISTNUMBER.Value := Table1.LISTNUMBER.Value;
    Table2.PRODUCTCODE.Text := Table1.PRODUCTCODE.Text;
    Table2.PRODUCTNAME.Text := Table1.PRODUCTNAME.Text;
    Table2.CUSTOMERCODE.Text := Table1.CUSTOMERCODE.Text;
    Table2.CUSTOMERNAME.Text := Table1.CUSTOMERNAME.Text;
    Table2.PIECEOFSAILING.Text := Table1.PIECEOFSAILING.Text;
    Table2.COMPANYNAME.Value := Table1.COMPANYNAME.Value;
    Table2.SAILINGCHARGER.Value := Table1.SAILINGCHARGER.Value;
    Table2.SAILINGPRICE.Value := Table1.SAILINGPRICE.Value;
    Table2.TOTALPRICE.Value := Table1.TOTALPRICE.Value;
    Table2.TOTALPAYMENT.Value := Table1.TOTALPAYMENT.Value;
    Table2.SAILINGDATE.Value := Table1.SAILINGDATE.Value;
    Table2.CASHPAYMENT.Value := Table1.CASHPAYMENT.Value;
    Table2.PAYMENT1.Value := Table1.PAYMENT1.Value;
    Table2.DATE1.Text := Table1.DATE1.Text;
    Table2.PAYMENT2.Value := Table1.PAYMENT2.Value;
    Table2.DATE2.Text := Table1.DATE2.Text;
    Table2.PAYMENT3.Value := Table1.PAYMENT3.Value;
    Table2.DATE3.Text := Table1.DATE3.Text;
    Table2.PAYMENT4.Value := Table1.PAYMENT4.Value;
    Table2.DATE4.Text := Table1.DATE4.Text;
    Table2.PAYMENT5.Value := Table1.PAYMENT5.Value;
    Table2.DATE5.Text := Table1.DATE5.Text;
    Table2.PAYMENT6.Value := Table1.PAYMENT6.Value;
    Table2.DATE6.Text := Table1.DATE6.Text;
    Table2.PAYMENT7.Value := Table1.PAYMENT7.Value;
    Table2.DATE7.Text := Table1.DATE7.Text;
    Table2.PAYMENT8.Value := Table1.PAYMENT8.Value;
    Table2.DATE8.Text := Table1.DATE8.Text;
    Table2.PAYMENT9.Value := Table1.PAYMENT9.Value;
    Table2.DATE9.Text := Table1.DATE9.Text;
    Table2.PAYMENT10.Value := Table1.PAYMENT10.Value;
    Table2.DATE10.Text := Table1.DATE10.Text;
    Table2.PAYMENT11.Value := Table1.PAYMENT11.Value;
    Table2.DATE11.Text := Table1.DATE11.Text;
    Table2.PAYMENT12.Value := Table1.PAYMENT12.Value;
    Table2.DATE12.Text := Table1.DATE12.Text;
    Table2.RESULT.Text := Table1.RESULT.Value;
    Table2.REMAININGPAYMENT.Value := Table1.REMAININGPAYMENT.Value;
    Table2.Post;
    Table2.Refresh;
  end;
  Table1.Next;
end;

```



```

Table2LISTNUMBER.Value := Table1LISTNUMBER.Value;
Table2PRODUCTCODE.Text := Table1PRODUCTCODE.Text;
Table2PRODUCTNAME.Text := Table1PRODUCTNAME.Text;
Table2CUSTOMERCODE.Text := Table1CUSTOMERCODE.Text;
Table2CUSTOMERNAME.Text := Table1CUSTOMERNAME.Text;
Table2PIECEOFSAILING.Text := Table1PIECEOFSAILING.Text;
Table2COMPANYNAME.Value := Table1COMPANYNAME.Value;
Table2SAILINGCHARGER.Value := Table1SAILINGCHARGER.Value;
Table2SAILINGPRICE.Value := Table1SAILINGPRICE.Value;
Table2TOTALPRICE.Value := Table1TOTALPRICE.Value;
Table2TOTALPAYMENT.Value := Table1TOTALPAYMENT.Value;
Table2SAILINGDATE.Value := Table1SAILINGDATE.Value;
Table2CASHPAYMENT.Value := Table1CASHPAYMENT.Value;
Table2PAYMENT1.Value := Table1PAYMENT1.Value;
Table2DATE1.Value := Table1DATE1.Value;
Table2PAYMENT2.Value := Table1PAYMENT2.Value;
Table2DATE2.Value := Table1DATE2.Value;
Table2PAYMENT3.Value := Table1PAYMENT3.Value;
Table2DATE3.Value := Table1DATE3.Value;
Table2PAYMENT4.Value := Table1PAYMENT4.Value;
Table2DATE4.Value := Table1DATE4.Value;
Table2PAYMENT5.Value := Table1PAYMENT5.Value;
Table2DATE5.Value := Table1DATE5.Value;
Table2PAYMENT6.Value := Table1PAYMENT6.Value;
Table2DATE6.Value := Table1DATE6.Value;
Table2PAYMENT7.Value := Table1PAYMENT7.Value;
Table2DATE7.Value := Table1DATE7.Value;
Table2PAYMENT8.Value := Table1PAYMENT8.Value;
Table2DATE8.Value := Table1DATE8.Value;
Table2PAYMENT9.Value := Table1PAYMENT9.Value;
Table2DATE9.Value := Table1DATE9.Value;
Table2PAYMENT10.Value := Table1PAYMENT10.Value;
Table2DATE10.Value := Table1DATE10.Value;
Table2PAYMENT11.Value := Table1PAYMENT11.Value;
Table2DATE11.Value := Table1DATE11.Value;
Table2PAYMENT12.Value := Table1PAYMENT12.Value;
Table2DATE12.Value := Table1DATE12.Value;
Table2RESULT.Text := Table1RESULT.Value;
Table2REMAININGPAYMENT.Value := Table1REMAININGPAYMENT.Value;
Table2.Post;
Table2.Refresh;
end;
Table1.Next;
end;
end;

```

```

If (Edit2.Text <> "") Then
begin
Table2.Refresh;
Table1.First;

```

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

```
procedure TForm10.BitBtn4Click(Sender: TObject);  
begin  
Form1.Show;  
Form10.Visible := False;  
end;
```

```
procedure TForm10.BitBtn2Click(Sender: TObject);  
begin  
GroupBox1.Visible := True;  
GroupBox2.Visible := False;  
BitBtn5.Enabled := False;  
end;
```

```
procedure TForm10.Edit3Change(Sender: TObject);  
begin  
BitBtn5.Enabled := True;  
end;
```

```
procedure TForm10.BitBtn6Click(Sender: TObject);  
begin  
GroupBox1.Visible := False;  
GroupBox2.Visible := True;  
BitBtn5.Enabled := False;  
end;
```

```
procedure TForm10.BitBtn1Click(Sender: TObject);  
Var  
Aranan : Boolean;  
begin  
Aranan := False;  
Table2.Refresh;  
Table2.First;  
While not (Table2.IsEmpty) Do  
begin  
Table2.Delete;  
Table2.Next;  
end;
```

```
If (Edit1.Text <> "") Then  
begin  
Table2.Refresh;  
Table1.First;  
While not Table1.Eof Do  
begin  
If (Table1.CUSTOMERCODE.Text = Edit1.Text) Then  
begin  
Aranan := True;  
Table2.Append;
```



```

Table1SAILINGPRICE: TIntegerField;
Table1TOTALPRICE: TIntegerField;
Table1TOTALPAYMENT: TIntegerField;
Table1REMAININGPAYMENT: TIntegerField;
Table1CASHPAYMENT: TIntegerField;
Table1PAYMENT1: TIntegerField;
Table1DATE1: TDateField;
Table1PAYMENT2: TIntegerField;
Table1DATE2: TDateField;
Table1PAYMENT3: TIntegerField;
Table1DATE3: TDateField;
Table1PAYMENT4: TIntegerField;
Table1DATE4: TDateField;
Table1PAYMENT5: TIntegerField;
Table1DATE5: TDateField;
Table1PAYMENT6: TIntegerField;
Table1DATE6: TDateField;
Table1PAYMENT7: TIntegerField;
Table1DATE7: TDateField;
Table1PAYMENT8: TIntegerField;
Table1DATE8: TDateField;
Table1PAYMENT9: TIntegerField;
Table1DATE9: TDateField;
Table1PAYMENT10: TIntegerField;
Table1DATE10: TDateField;
Table1PAYMENT11: TIntegerField;
Table1DATE11: TDateField;
Table1PAYMENT12: TIntegerField;
Table1DATE12: TDateField;
Table1RESULT: TStringField;
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure Edit3Change(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form10: TForm10;

implementation

uses Unit1, Unit4;

```

Table1: TTable;
 Table2: TTable;
 Table2LISTNUMBER: TIntegerField;
 Table2CUSTOMERCODE: TStringField;
 Table2CUSTOMERNAME: TStringField;
 Table2PRODUCTCODE: TStringField;
 Table2PRODUCTNAME: TStringField;
 Table2COMPANYNAME: TStringField;
 Table2SAILINGCHARGER: TStringField;
 Table2SAILINGDATE: TDateField;
 Table2PIECEOFSAILING: TIntegerField;
 Table2SAILINGPRICE: TIntegerField;
 Table2TOTALPRICE: TIntegerField;
 Table2TOTALPAYMENT: TIntegerField;
 Table2REMAININGPAYMENT: TIntegerField;
 Table2CASHPAYMENT: TIntegerField;
 Table2PAYMENT1: TIntegerField;
 Table2DATE1: TDateField;
 Table2PAYMENT2: TIntegerField;
 Table2DATE2: TDateField;
 Table2PAYMENT3: TIntegerField;
 Table2DATE3: TDateField;
 Table2PAYMENT4: TIntegerField;
 Table2DATE4: TDateField;
 Table2PAYMENT5: TIntegerField;
 Table2DATE5: TDateField;
 Table2PAYMENT6: TIntegerField;
 Table2DATE6: TDateField;
 Table2PAYMENT7: TIntegerField;
 Table2DATE7: TDateField;
 Table2PAYMENT8: TIntegerField;
 Table2DATE8: TDateField;
 Table2PAYMENT9: TIntegerField;
 Table2DATE9: TDateField;
 Table2PAYMENT10: TIntegerField;
 Table2DATE10: TDateField;
 Table2PAYMENT11: TIntegerField;
 Table2DATE11: TDateField;
 Table2PAYMENT12: TIntegerField;
 Table2DATE12: TDateField;
 Table2RESULT: TStringField;
 Table1LISTNUMBER: TIntegerField;
 Table1CUSTOMERCODE: TStringField;
 Table1CUSTOMERNAME: TStringField;
 Table1PRODUCTCODE: TStringField;
 Table1PRODUCTNAME: TStringField;
 Table1COMPANYNAME: TStringField;
 Table1SAILINGCHARGER: TStringField;
 Table1SAILINGDATE: TDateField;
 Table1PIECEOFSAILING: TIntegerField;

nit Unit10;

interface

uses

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

type

TForm10 = class(TForm)
 Panel1: TPanel;
 Edit1: TEdit;
 Edit2: TEdit;
 Label1: TLabel;
 Label2: TLabel;
 Label3: TLabel;
 Label4: TLabel;
 BitBtn1: TBitBtn;
 DataSource1: TDataSource;
 DBGrid1: TDBGrid;
 DBNavigator1: TDBNavigator;
 BitBtn2: TBitBtn;
 BitBtn3: TBitBtn;
 BitBtn4: TBitBtn;
 GroupBox1: TGroupBox;
 Edit3: TEdit;
 Label5: TLabel;
 BitBtn5: TBitBtn;
 BitBtn6: TBitBtn;
 Label6: TLabel;
 Label7: TLabel;
 GroupBox2: TGroupBox;
 Label8: TLabel;
 Edit4: TEdit;
 GroupBox3: TGroupBox;
 CheckBox1: TCheckBox;
 CheckBox2: TCheckBox;
 CheckBox3: TCheckBox;
 CheckBox4: TCheckBox;
 CheckBox5: TCheckBox;
 CheckBox6: TCheckBox;
 CheckBox7: TCheckBox;
 CheckBox8: TCheckBox;
 CheckBox9: TCheckBox;
 CheckBox10: TCheckBox;
 CheckBox11: TCheckBox;
 CheckBox12: TCheckBox;
 Label9: TLabel;
 DateTimePicker1: TDateTimePicker;

```
Table1DATE11.Text := DatetoStr(DateTimePicker1.Date);  
end;
```

```
If (CheckBox12.Checked = True) Then  
begin  
Table1PAYMENT12.Value := StrToInt(Edit4.Text);  
Table1DATE12.Text := DatetoStr(DateTimePicker1.Date);  
end;
```

```
Table1TOTALPAYMENT.Value := ((Table1CASHPAYMENT.Value) +  
(Table1PAYMENT1.Value) + (Table1PAYMENT2.Value) + (Table1PAYMENT3.Value)  
+ (Table1PAYMENT4.Value) + (Table1PAYMENT5.Value) +  
(Table1PAYMENT6.Value) + (Table1PAYMENT7.Value)  
+ (Table1PAYMENT8.Value) + (Table1PAYMENT9.Value) +  
(Table1PAYMENT10.Value) + (Table1PAYMENT11.Value) +  
(Table1PAYMENT12.Value));
```

```
Table1.Post;  
ShowMessage('THE PAYMENT INSTALLMENT PROCESS IS COMPLETED FOR  
LISTED NUMBER BY ' + Edit3.Text + ' CUSTOMER ON THE LIST');  
Edit3.Text := ""; Edit4.Text := "";  
BitBtn5.Enabled := False;  
end;  
Table1.Next;  
end;
```

```
If (ListNo = False) Then  
begin  
ShowMessage('THE LIST NUMBER ' + Edit3.Text + ' IS CAN NOT FIND, PLEASE  
CHECK IS THAT RIGHT?');  
end;  
end;  
  
end;
```

```
procedure TForm10.BitBtn3Click(Sender: TObject);  
begin  
Form4.Show;  
Form10.Visible := False;  
end;  
  
end.
```



```

begin
  Table1PAYMENT3.Value := StrToInt(Edit4.Text);
  Table1DATE3.Text := DatetoStr(DateTimePicker1.Date);
end;

If (CheckBox4.Checked = True) Then
begin
  Table1PAYMENT4.Value := StrToInt(Edit4.Text);
  Table1DATE4.Text := DatetoStr(DateTimePicker1.Date);
end;

If (CheckBox5.Checked = True) Then
begin
  Table1PAYMENT5.Value := StrToInt(Edit4.Text);
  Table1DATE5.Text := DatetoStr(DateTimePicker1.Date);
end;

If (CheckBox6.Checked = True) Then
begin
  Table1PAYMENT6.Value := StrToInt(Edit4.Text);
  Table1DATE6.Text := DatetoStr(DateTimePicker1.Date);
end;

If (CheckBox7.Checked = True) Then
begin
  Table1PAYMENT7.Value := StrToInt(Edit4.Text);
  Table1DATE7.Text := DatetoStr(DateTimePicker1.Date);
end;

If (CheckBox8.Checked = True) Then
begin
  Table1PAYMENT8.Value := StrToInt(Edit4.Text);
  Table1DATE8.Text := DatetoStr(DateTimePicker1.Date);
end;

If (CheckBox9.Checked = True) Then
begin
  Table1PAYMENT9.Value := StrToInt(Edit4.Text);
  Table1DATE9.Text := DatetoStr(DateTimePicker1.Date);
end;

If (CheckBox10.Checked = True) Then
begin
  Table1PAYMENT10.Value := StrToInt(Edit4.Text);
  Table1DATE10.Text := DatetoStr(DateTimePicker1.Date);
end;

If (CheckBox11.Checked = True) Then
begin
  Table1PAYMENT11.Value := StrToInt(Edit4.Text);

```

end;

```
If (Aranan = False ) Then
begin
  ShowMessage('THE RECORD IS NOT FOUNDED!');
end;
```

end;

```
procedure TForm10.BitBtn5Click(Sender: TObject);
Var
ListNo : Boolean;
begin
ListNo := False;
```

```
If MessageDlg('ARE YOU SURE THAT YOU WANT TO MAKE PAYMENT EVENT
FROM THE TABLE!..',
mtconfirmation, [mbYES, mbNO], 0) = mrYES Then
begin
  Table1.First;
  While not Table1.Eof Do
  begin
    If (Table1.LISTNUMBER.Value = StrToInt(Edit3.Text)) Then
    begin
      ListNo := True;
      Table1.Refresh;
      Table1.Edit;
      Table1.REMAININGPAYMENT.Value := ((Table1.REMAININGPAYMENT.Value) -
(StrToInt(Edit4.Text)));
```

```
    If (Table1.REMAININGPAYMENT.Value <= 0) Then
    begin
      Table1.RESULT.Text := 'CLOSED'
    end;
```

```
    If (CheckBox1.Checked = True) Then
    begin
      Table1.PAYMENT1.Value := StrToInt(Edit4.Text);
      Table1.DATE1.Text := DatetoStr(DateTimePicker1.Date);
    end;
```

```
    If (CheckBox2.Checked = True) Then
    begin
      Table1.PAYMENT2.Value := StrToInt(Edit4.Text);
      Table1.DATE2.Text := DatetoStr(DateTimePicker1.Date);
    end;
```

```
    If (CheckBox3.Checked = True) Then
```



```

While not Table1.Eof Do
begin
If (Table1.CUSTOMERNAME.Text = Edit2.Text) Then
begin
Aranan := True;
Table2.Append;
Table2.LISTNUMBER.Value := Table1.LISTNUMBER.Value;
Table2.PRODUCTCODE.Text := Table1.PRODUCTCODE.Text;
Table2.PRODUCTNAME.Text := Table1.PRODUCTNAME.Text;
Table2.CUSTOMERCODE.Text := Table1.CUSTOMERCODE.Text;
Table2.CUSTOMERNAME.Text := Table1.CUSTOMERNAME.Text;
Table2.PIECEOFSAILING.Text := Table1.PIECEOFSAILING.Text;
Table2.COMPANYNAME.Value := Table1.COMPANYNAME.Value;
Table2.SAILINGCHARGER.Value := Table1.SAILINGCHARGER.Value;
Table2.SAILINGPRICE.Value := Table1.SAILINGPRICE.Value;
Table2.TOTALPRICE.Value := Table1.TOTALPRICE.Value;
Table2.TOTALPAYMENT.Value := Table1.TOTALPAYMENT.Value;
Table2.SAILINGDATE.Value := Table1.SAILINGDATE.Value;
Table2.CASHPAYMENT.Value := Table1.CASHPAYMENT.Value;
Table2.PAYMENT1.Value := Table1.PAYMENT1.Value;
Table2.DATE1.Text := Table1.DATE1.Text;
Table2.PAYMENT2.Value := Table1.PAYMENT2.Value;
Table2.DATE2.Text := Table1.DATE2.Text;
Table2.PAYMENT3.Value := Table1.PAYMENT3.Value;
Table2.DATE3.Text := Table1.DATE3.Text;
Table2.PAYMENT4.Value := Table1.PAYMENT4.Value;
Table2.DATE4.Text := Table1.DATE4.Text;
Table2.PAYMENT5.Value := Table1.PAYMENT5.Value;
Table2.DATE5.Text := Table1.DATE5.Text;
Table2.PAYMENT6.Value := Table1.PAYMENT6.Value;
Table2.DATE6.Text := Table1.DATE6.Text;
Table2.PAYMENT7.Value := Table1.PAYMENT7.Value;
Table2.DATE7.Text := Table1.DATE7.Text;
Table2.PAYMENT8.Value := Table1.PAYMENT8.Value;
Table2.DATE8.Text := Table1.DATE8.Text;
Table2.PAYMENT9.Value := Table1.PAYMENT9.Value;
Table2.DATE9.Text := Table1.DATE9.Text;
Table2.PAYMENT10.Value := Table1.PAYMENT10.Value;
Table2.DATE10.Text := Table1.DATE10.Text;
Table2.PAYMENT11.Value := Table1.PAYMENT11.Value;
Table2.DATE11.Text := Table1.DATE11.Text;
Table2.PAYMENT12.Value := Table1.PAYMENT12.Value;
Table2.DATE12.Text := Table1.DATE12.Text;
Table2.RESULT.Text := Table1.RESULT.Value;
Table2.REMAININGPAYMENT.Value := Table1.REMAININGPAYMENT.Value;
Table2.Post;
Table2.Refresh;
end;
Table1.Next;
end;

```

```

Table2LISTNUMBER.Value := Table1LISTNUMBER.Value;
Table2PRODUCTCODE.Text := Table1PRODUCTCODE.Text;
Table2PRODUCTNAME.Text := Table1PRODUCTNAME.Text;
Table2CUSTOMERCODE.Text := Table1CUSTOMERCODE.Text;
Table2CUSTOMERNAME.Text := Table1CUSTOMERNAME.Text;
Table2PIECEOFSAILING.Text := Table1PIECEOFSAILING.Text;
Table2COMPANYNAME.Value := Table1COMPANYNAME.Value;
Table2SAILINGCHARGER.Value := Table1SAILINGCHARGER.Value;
Table2SAILINGPRICE.Value := Table1SAILINGPRICE.Value;
Table2TOTALPRICE.Value := Table1TOTALPRICE.Value;
Table2TOTALPAYMENT.Value := Table1TOTALPAYMENT.Value;
Table2SAILINGDATE.Value := Table1SAILINGDATE.Value;
Table2CASHPAYMENT.Value := Table1CASHPAYMENT.Value;
Table2PAYMENT1.Value := Table1PAYMENT1.Value;
Table2DATE1.Value := Table1DATE1.Value;
Table2PAYMENT2.Value := Table1PAYMENT2.Value;
Table2DATE2.Value := Table1DATE2.Value;
Table2PAYMENT3.Value := Table1PAYMENT3.Value;
Table2DATE3.Value := Table1DATE3.Value;
Table2PAYMENT4.Value := Table1PAYMENT4.Value;
Table2DATE4.Value := Table1DATE4.Value;
Table2PAYMENT5.Value := Table1PAYMENT5.Value;
Table2DATE5.Value := Table1DATE5.Value;
Table2PAYMENT6.Value := Table1PAYMENT6.Value;
Table2DATE6.Value := Table1DATE6.Value;
Table2PAYMENT7.Value := Table1PAYMENT7.Value;
Table2DATE7.Value := Table1DATE7.Value;
Table2PAYMENT8.Value := Table1PAYMENT8.Value;
Table2DATE8.Value := Table1DATE8.Value;
Table2PAYMENT9.Value := Table1PAYMENT9.Value;
Table2DATE9.Value := Table1DATE9.Value;
Table2PAYMENT10.Value := Table1PAYMENT10.Value;
Table2DATE10.Value := Table1DATE10.Value;
Table2PAYMENT11.Value := Table1PAYMENT11.Value;
Table2DATE11.Value := Table1DATE11.Value;
Table2PAYMENT12.Value := Table1PAYMENT12.Value;
Table2DATE12.Value := Table1DATE12.Value;
Table2RESULT.Text := Table1RESULT.Value;
Table2REMAININGPAYMENT.Value := Table1REMAININGPAYMENT.Value;
Table2.Post;
Table2.Refresh;
end;
Table1.Next;
end;
end;

```

```

If (Edit2.Text <> "") Then
begin
Table2.Refresh;
Table1.First;

```

{ \$R *.dfm }

```
procedure TForm10.BitBtn4Click(Sender: TObject);
begin
  Form1.Show;
  Form10.Visible := False;
end;
```

```
procedure TForm10.BitBtn2Click(Sender: TObject);
begin
  GroupBox1.Visible := True;
  GroupBox2.Visible := False;
  BitBtn5.Enabled := False;
end;
```

```
procedure TForm10.Edit3Change(Sender: TObject);
begin
  BitBtn5.Enabled := True;
end;
```

```
procedure TForm10.BitBtn6Click(Sender: TObject);
begin
  GroupBox1.Visible := False;
  GroupBox2.Visible := True;
  BitBtn5.Enabled := False;
end;
```

```
procedure TForm10.BitBtn1Click(Sender: TObject);
Var
  Aranan : Boolean;
begin
  Aranan := False;
  Table2.Refresh;
  Table2.First;
  While not (Table2.IsEmpty) Do
  begin
    Table2.Delete;
    Table2.Next;
  end;
```

```
If (Edit1.Text <> "") Then
begin
  Table2.Refresh;
  Table1.First;
  While not Table1.Eof Do
  begin
    If (Table1.CUSTOMERCODE.Text = Edit1.Text) Then
    begin
      Aranan := True;
      Table2.Append;
```



```

Table1SAILINGPRICE: TIntegerField;
Table1TOTALPRICE: TIntegerField;
Table1TOTALPAYMENT: TIntegerField;
Table1REMAININGPAYMENT: TIntegerField;
Table1CASHPAYMENT: TIntegerField;
Table1PAYMENT1: TIntegerField;
Table1DATE1: TDateField;
Table1PAYMENT2: TIntegerField;
Table1DATE2: TDateField;
Table1PAYMENT3: TIntegerField;
Table1DATE3: TDateField;
Table1PAYMENT4: TIntegerField;
Table1DATE4: TDateField;
Table1PAYMENT5: TIntegerField;
Table1DATE5: TDateField;
Table1PAYMENT6: TIntegerField;
Table1DATE6: TDateField;
Table1PAYMENT7: TIntegerField;
Table1DATE7: TDateField;
Table1PAYMENT8: TIntegerField;
Table1DATE8: TDateField;
Table1PAYMENT9: TIntegerField;
Table1DATE9: TDateField;
Table1PAYMENT10: TIntegerField;
Table1DATE10: TDateField;
Table1PAYMENT11: TIntegerField;
Table1DATE11: TDateField;
Table1PAYMENT12: TIntegerField;
Table1DATE12: TDateField;
Table1RESULT: TStringField;
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure Edit3Change(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form10: TForm10;

implementation

uses Unit1, Unit4;

```

Table1: TTable;
 Table2: TTable;
 Table2LISTNUMBER: TIntegerField;
 Table2CUSTOMERCODE: TStringField;
 Table2CUSTOMERNAME: TStringField;
 Table2PRODUCTCODE: TStringField;
 Table2PRODUCTNAME: TStringField;
 Table2COMPANYNAME: TStringField;
 Table2SAILINGCHARGER: TStringField;
 Table2SAILINGDATE: TDateField;
 Table2PIECEOFSAILING: TIntegerField;
 Table2SAILINGPRICE: TIntegerField;
 Table2TOTALPRICE: TIntegerField;
 Table2TOTALPAYMENT: TIntegerField;
 Table2REMAININGPAYMENT: TIntegerField;
 Table2CASHPAYMENT: TIntegerField;
 Table2PAYMENT1: TIntegerField;
 Table2DATE1: TDateField;
 Table2PAYMENT2: TIntegerField;
 Table2DATE2: TDateField;
 Table2PAYMENT3: TIntegerField;
 Table2DATE3: TDateField;
 Table2PAYMENT4: TIntegerField;
 Table2DATE4: TDateField;
 Table2PAYMENT5: TIntegerField;
 Table2DATE5: TDateField;
 Table2PAYMENT6: TIntegerField;
 Table2DATE6: TDateField;
 Table2PAYMENT7: TIntegerField;
 Table2DATE7: TDateField;
 Table2PAYMENT8: TIntegerField;
 Table2DATE8: TDateField;
 Table2PAYMENT9: TIntegerField;
 Table2DATE9: TDateField;
 Table2PAYMENT10: TIntegerField;
 Table2DATE10: TDateField;
 Table2PAYMENT11: TIntegerField;
 Table2DATE11: TDateField;
 Table2PAYMENT12: TIntegerField;
 Table2DATE12: TDateField;
 Table2RESULT: TStringField;
 Table1LISTNUMBER: TIntegerField;
 Table1CUSTOMERCODE: TStringField;
 Table1CUSTOMERNAME: TStringField;
 Table1PRODUCTCODE: TStringField;
 Table1PRODUCTNAME: TStringField;
 Table1COMPANYNAME: TStringField;
 Table1SAILINGCHARGER: TStringField;
 Table1SAILINGDATE: TDateField;
 Table1PIECEOFSAILING: TIntegerField;

```

unit Unit11;

interface

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

type
  TForm11 = class(TForm)
    QuickRep1: TQuickRep;
    TitleBand1: TQRBand;
    PageHeaderBand1: TQRBand;
    PageFooterBand1: TQRBand;
    Table1: TTable;
    QRLabel1: TQRLabel;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    DetailBand1: TQRBand;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
    QRDBText5: TQRDBText;
    QRDBText6: TQRDBText;
    QRDBText7: TQRDBText;
    QRDBText8: TQRDBText;
    QRLabel7: TQRLabel;
    QRLabel8: TQRLabel;
    QRLabel9: TQRLabel;
    QRSysData1: TQRSysData;
    QRSysData2: TQRSysData;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form11: TForm11;

implementation

{$R *.dfm}

end.

```



```

unit Unit12;

interface

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

type
  TForm12 = class(TForm)
    QuickRep1: TQuickRep;
    DetailBand1: TQRBand;
    PageFooterBand1: TQRBand;
    PageHeaderBand1: TQRBand;
    Table1: TTable;
    QRLabel1: TQRLabel;
    TitleBand1: TQRBand;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRLabel7: TQRLabel;
    QRLabel8: TQRLabel;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
    QRDBText5: TQRDBText;
    QRDBText6: TQRDBText;
    QRDBText7: TQRDBText;
    QRSysData1: TQRSysData;
    QRSysData2: TQRSysData;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form12: TForm12;

implementation

uses Unit11;

{$R *.dfm}

procedure TForm12.BitBtn1Click(Sender: TObject);

```

```
begin
Form11.QuickRep1.Preview;
Form12.Visible := False;
end;

end.
```

```

unit Unit12;

interface

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

type
  TForm12 = class(TForm)
    QuickRep1: TQuickRep;
    DetailBand1: TQRBand;
    PageFooterBand1: TQRBand;
    PageHeaderBand1: TQRBand;
    Table1: TTable;
    QRLabel1: TQRLabel;
    TitleBand1: TQRBand;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRLabel7: TQRLabel;
    QRLabel8: TQRLabel;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
    QRDBText5: TQRDBText;
    QRDBText6: TQRDBText;
    QRDBText7: TQRDBText;
    QRSysData1: TQRSysData;
    QRSysData2: TQRSysData;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form12: TForm12;

implementation

uses Unit11;

{$R *.dfm}

procedure TForm12.BitBtn1Click(Sender: TObject);

```


unit Unit13;

interface

uses

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

type

TForm13 = class(TForm)
 QuickRep1: TQuickRep;
 DetailBand1: TQRBand;
 QRDBText2: TQRDBText;
 QRDBText3: TQRDBText;
 QRDBText4: TQRDBText;
 QRDBText5: TQRDBText;
 QRDBText6: TQRDBText;
 QRDBText7: TQRDBText;
 PageFooterBand1: TQRBand;
 QRSysData1: TQRSysData;
 QRSysData2: TQRSysData;
 PageHeaderBand1: TQRBand;
 QRLabel1: TQRLabel;
 TitleBand1: TQRBand;
 QRLabel3: TQRLabel;
 QRLabel4: TQRLabel;
 QRLabel5: TQRLabel;
 QRLabel6: TQRLabel;
 QRLabel7: TQRLabel;
 QRLabel8: TQRLabel;
 Table1: TTable;
 QRDBText1: TQRDBText;
 QRLabel2: TQRLabel;

private

 { Private declarations }

public

 { Public declarations }

end;

var

 Form13: TForm13;

implementation

 {\$R *.dfm}

end.

unit Unit14;

interface

uses

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

type

TForm14 = class(TForm)

BitBtn1: TBitBtn;

BitBtn2: TBitBtn;

BitBtn3: TBitBtn;

BitBtn4: TBitBtn;

procedure BitBtn3Click(Sender: TObject);

procedure BitBtn4Click(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form14: TForm14;

implementation

uses Unit13, Unit1, Unit11, Unit12;

{ \$R *.dfm }

procedure TForm14.BitBtn3Click(Sender: TObject);

begin

Form13.QuickRep1.Preview;

end;

procedure TForm14.BitBtn4Click(Sender: TObject);

begin

Form1.Show;

Form14.Visible := False;

end;

procedure TForm14.BitBtn1Click(Sender: TObject);

begin

Form11.QuickRep1.Preview;

end;

procedure TForm14.BitBtn2Click(Sender: TObject);


```
begin  
Form12.QuickRep1.Preview;  
end;  
  
end.
```

SCREEN OUTPUT

Form1

GLASSES STOCK SYSTEM

MAIN MENU

PURCHASE OF GLASSES

✓ ENTRY FORM

✗ DELETE/EDIT FORM

? SEARCH/LIST FORM

SALING OF GLASSES

✓ SAILING FORMS

✓ SAILING LISTS

✓ COMING INSTALLMENT

✓ ACCOUNT

✓ REPORTS

✗ CLOSE PROGRAM

figure 1

Form2

ADD ENTRY FORM | SEARCH / LIST OF RECORD | DELETE / EDIT RECORD | QUIT ME

PRODUCT CODE

P. NAME

P. PEACE

SIZE

BUYING PLACE

BUYING PRICE

SAILING PRICE

BUYING DATE

SPECIAL NOTES FOR THE PRODUCT

NUMBER OF MINIMUM LIMIT FOR STOCK CONTROL UNIT

✓ ADD RECORD

? SEARCH THE RECORD

✗ CANCEL RECORD PROCESS

SHOW PRINT OUT

Figure 2

Form2

ADD ENTRY FORM SEARCH / LIST OF RECORD DELETE / EDIT RECORD QUIT ME

Please enter the 'PRODUCT CODE' or 'PRODUCT NAME' that you want to SEARCH

PRODUCT CODE

PRODUCT NAME

LIST NO	PRODUCT CODE	PRODUCT NAME	PRODUCT PEACE	PRODUCT SIZE	CONTROL U
1	005	POLICE	1	12x12	
2	001	POLICE	6	10x10	
3	002	RAIWAN	18	20x20	
4	003	SUN	1	15x15	
5	004	ROWER	11	12x12	
6	006	RIVER	-1	20x20	
7	007	METRO	28	10x10	
8	008	POLICE	20	20x20	
9	009	FLY	15	12x12	
10	010	RAIBAN	15	18x18	
11	011	METRO	50	9x9	
12	012	POLICE	15	15x15	
13	013	RIWER	12	20x20	
14	014	POLICE	30	15x15	
15	0015	POLICE	17	10x10	

Figure 3

Form2

ADD ENTRY FORM | SEARCH / LIST OF RECORD | DELETE / EDIT RECORD | QUIT ME

Please enter the code number of product that you want to edit and click to the "FIND" button!

PRODUCT CODE

P. CODE	<input type="text"/>	BUYING PLACE	<input type="text"/>
P. NAME	<input type="text"/>	BUYING PRICE	<input type="text"/>
P. PEACE	<input type="text"/>	CELLING PRICE	<input type="text"/>
SIZE	<input type="text"/>	BUYING DATE	<input type="text" value="30.10.2003"/>
LIST NO	<input type="text"/>	SPECIAL NOTES	<input type="text"/>

figure-4

Form2

ADD ENTRY FORM SEARCH / LIST OF RECORD DELETE / EDIT RECORD STOK CONTROL QUIT MENUS

To check stock limits click to "CHECK STOCK LIMITS"

PageControl1: TPageControl
 Origin: 8, 8; Size: 705 x 441
 Tab Stop: True; Order: 0

CHECK STOCK LIMITS

LIST NO	PRODUCT_CODE	PRODUCT_NAME	PRODUCT PEACE	PRODUCT SIZE	CONTROL UNIT
1	005	POLICE	0	12X12	3
2	005	POLICE	5	10x10	3

figure 5

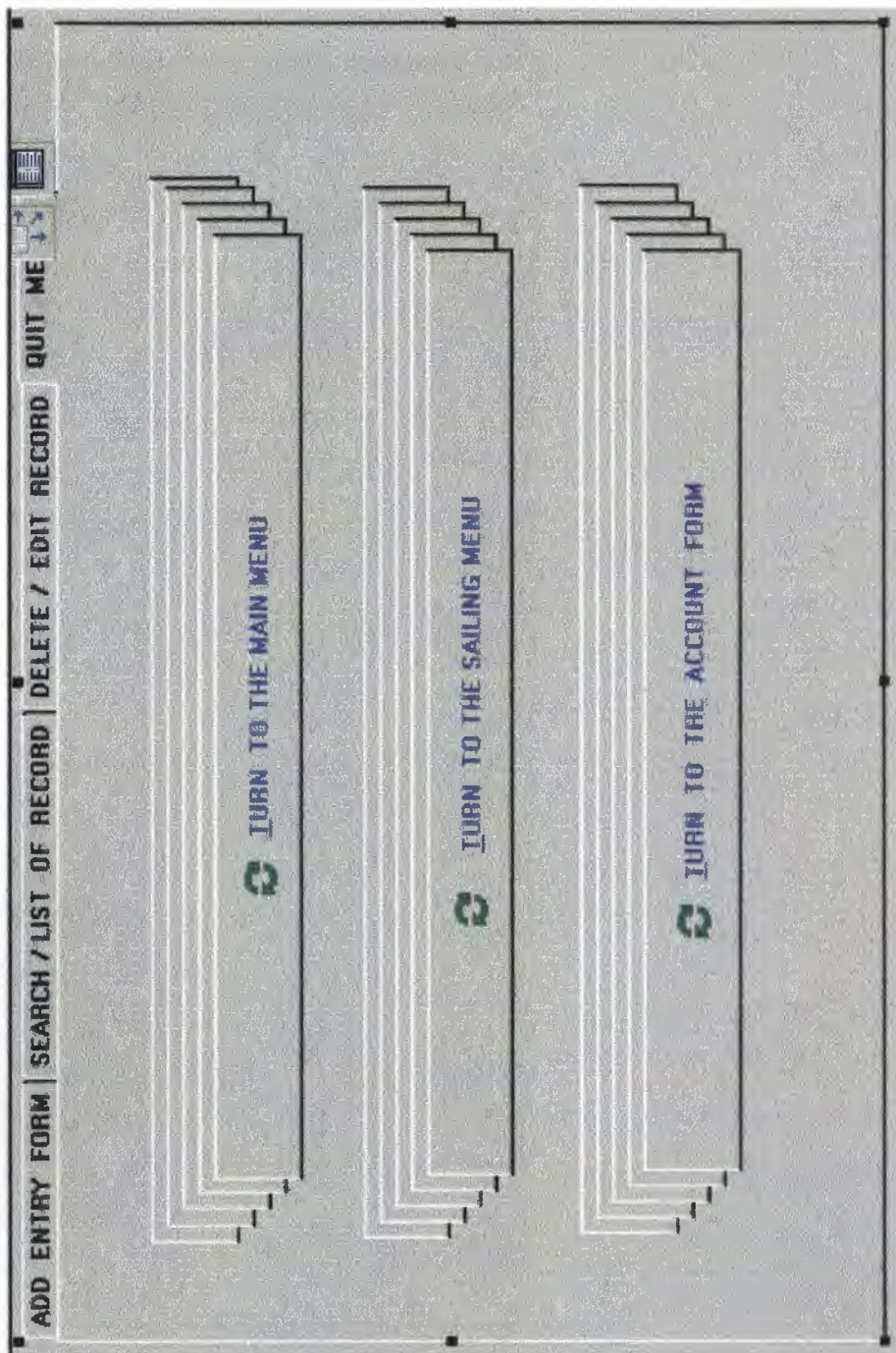


figure-6

Form5

Please make your type of saling...

CASH SALING FORM

INSTALLMENT SALING FORM

figure 7

Form3

CASH SALING FORM

PRODUCT CODE		PRODUCT NAME	
SALING PIECE		SALING PRICE	
CUSTOMER NAME		TOTAL PRICE	
COMPANY NAME		SALING DATE	04.12.2003
		SALING CHARGER	

 DO CELLING

 TURN TO MAIN MENU

 TURN TO TABLES FORM

Figure 8

Form4

INSTALLMENT Saling FORM

P. CODE		PRODUCT NAME	
SALING PIECE		SALING PRICE	
CUSTOMER CODE		TOTAL PRICE	
CUSTOMER NAME		CASH PAYMENT	
COMPANY NAME		REST OF PAYMENT	
SALING DATE	04.12.2003	SALING CHARGER	

✓ DO CELLING

TURN TO TABLES FORM

TURN TO MAIN MENU

Figure 9

Form6

CASH SALING ALL LIST | LIST BY CUSTOMER | LIST BY DATE | QUIT MENU

SHOW PRINT PREVIEW

LIST NO	CUSTOMER NAME	PRODUCT CODE	PRODUCT NAME	PIECE OF SALING	SALING PRICE	TOTAL
1	MEHMET YÜREKLİ	002	RAIWAN	1	50000000	50
2	CEMALİYE RENKLİ	006	RIVER	2	80000000	160
3	BAHRI ERDEMLİ	004	ROWER	1	30000000	30
4	BEYAZ GÜLEÇ	003	SUN	3	150000000	450
5	FIRAT KARA	001	POLICE	1	10000000	10
6	YEŞİM GÜL	005	POLICE	1	1250000000	1250
7	ALİ YÜCE	006	RIVER	2	25000000	50
8	MEHMET AK	006	RIVER	1	150000000	150
9	FATMA FIDAN	003	SUN	2	40000000	80

Figure 10

Form6

CASH SALING ALL LIST | LIST BY CUSTOMER | LIST BY DATE | QUIT MENU

ENTER THE LIMIT DATES

FIRST DATE: 06.12.2003

FIRST DATE: 06.12.2003

PRINT THE TABLE

SEARCH AND LIST

LIST NO	CUSTOMER NAME	PRODUCT CODE	PRODUCT NAME	PIECE OF SALING	SALING PRICE	TOTAL
16 CDF		002	RAIWAN	1	30000000	
16 CD		006	RIVER	2	60000000	
16 BGH		004	ROWER	1	30000000	
BG		003	SUN	3	90000000	

Figure 12

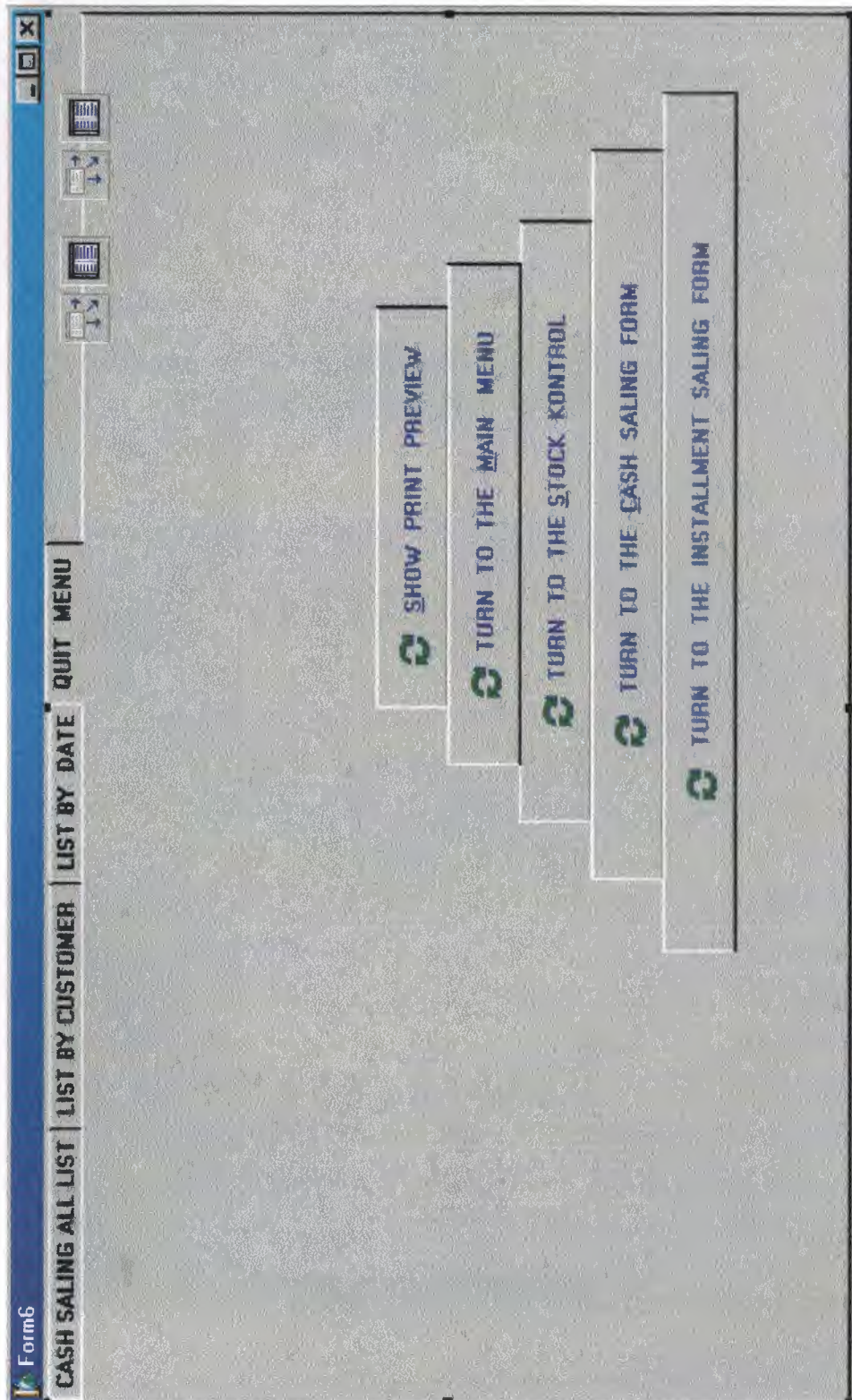


Figure 13

Form7

INSTALLMENT SAILING ALL LIST | LIST BY CUSTOMER | LIST BY DATE | QUIT MENU

SHOW REVIEW

LIST NUMBER	CUSTOMER CODE	CUSTOMER NAME	PRODUCT CODE	PRODUCT NAME	COMPANY NAME
1	ART	AHMET GÜNEŞ	005	POLICE	K.T.H.Y
2	002	VASİYE ÇETİN	003	ROWER	CB
3	009	ALİ YEŞİL	004	ROWER	E.B.S.
4	008	HAKAN VELİ	005	POLICE	
5	001	ALİ AY	005	POLICE	POLİS GNL. MD.
6	007	A.B.C. LTD. ŞT.	006	RIVER	A.B.C. LTD. ŞT.
7	003	GÜL KIRMIZI	007	METRO	L.ŞA TÜRK. BELE
8	006	HAKAN ALİ	0015	METRO	BEYAZ MİMARLIK
9	004	MELTEM BEYAZ	004	ROWER	ROCKS KOTEL
10	005	CİHAN BEY	006	RIVER	DENİZKIZI HOTEL
11	0010	VEDAT YILDIRIM	010	RAIBAN	TURKCELL

Figure 14

Form7

INSTALLMENT SAILING ALL LIST LIST BY CUSTOMER LIST BY DATE QUIT MENU

ENTER THE NAME

SEARCH AND LIST

LIST NUMBER	CUSTOMER CODE	CUSTOMER NAME	PRODUCT CODE	PRODUCT NAME	COMPANY NAME
4		HAKAN VELI	005	POLICE	
5		ALI AY	005	POLICE	POLIS GNL. MD.
6		A.B.C. LTD. ŞT.	006	RIVER	A.B.C. LTD. ŞT.
7		GÜL KIRMIZI	007	METRO	L.ŞA TÜRK. BEL
8		HAKAN ALI	0015	METRO	BEYAZ MİMARLI
9		MELTEM BEYAZ	004	ROWER	ROCKS KOTEL

Figure 15

Form7

INSTALLMENT SAILING ALL LIST | LIST BY CUSTOMER | LIST BY DATE | QUIT MENU

ENTER THE DATE LIMITS

FIRST DATE: 06.12.2003 | LAST DATE: 06.12.2003

SEARCH AND LIST

LIST NUMBER	CUSTOMER CODE	CUSTOMER NAME	PRODUCT CODE	PRODUCT NAME	COMPANY NAME
4		HAKAN VELİ	005	POLICE	POLİS GNL. MD.
5		ALİ AY	005	POLICE	A.B.C. LTD. ŞT.
6		A.B.C. LTD. ŞT.	006	RIVER	L.ŞA TÜRK. BELE
7		GÜL KIRMIZI	007	METRO	BEYAZ MİMARLIK
8		HAKAN ALİ	0015	METRO	ROCKS KOTEL
9		MELTEM BEYAZ	004	ROWER	

Figure 16

Form7

INSTALLMENT SAILING ALL LIST | LIST BY CUSTOMER | LIST BY DATE | QUIT MENU

TURN TO THE INSTALLMENT SAILING FORM

TURN TO THE MAIN MENU

TURN TO THE STOCK CONTROL

TURN TO THE CASH SAILING FORM

Figure 17

Form8

Please make your type of saling table form...

CASH SALINGS TABLE FORM

INSTALLMENTS SALINGS TABLE FORM

Form9

COST LIMIT | TABLE | QUIT MENU

ON THIS PAGE YOU HAVE TO ENTER THE PERCENTAGE OF COST
AND THEN BY CLICKING "DO CALCULATIONS" BUTTON THE PROGRAM CAN MAKE THE CALCULATIONS

PLEASE ENTER THE PERCENTAGE OF COST PRICE %

DO CALCULATIONS

Figure 18

COST LIMIT		TABLE		QUIT MENU			
LIST NO	PRODUCT NAME	PRODUCT CODE	BUYING PRICE	COST PRICE	CASH SAILING	INSTALLMENT SAILING	PROFIT
1	RAIWAN	002	30000000	12000000	50000000		-700000
2	RIVER	006	50000000	200000000	160000000		-400000
3	ROWER	004	25000000	100000000	30000000		-700000
4	SUN	003	10000000	40000000	450000000		410000
5	POLICE	001	45000000	180000000	10000000		-170000
6	POLICE	005	7000000	28000000	1250000000		122200
7	RIVER	006	15000000	60000000	5000000		-550000
8	RIVER	006	90000000	360000000	150000000		-210000
9	SUN	003	25000000	100000000	80000000		-200000
10	POLICE	005	10000000	40000000		150000000	110000
11	ROWER	003	10000000	40000000		90000000	500000
12	ROWER	004	10000000	40000000		135000000	950000
13	POLICE	005	20000000	80000000		1000000000	920000
14	POLICE	005	20000000	80000000		500000000	420000
15	RIVER	006	1000000	4000000		15000000	110000
16	METRO	007	5000000	20000000		50000000	300000
17	METRO	0015	10000000	40000000		90000000	500000
18	ROWER	004	10000000	40000000		135000000	950000
19	RIVER	006	1000000	4000000		3000000	-100000
20	RAIBAN	010	40000000	160000000		600000000	440000

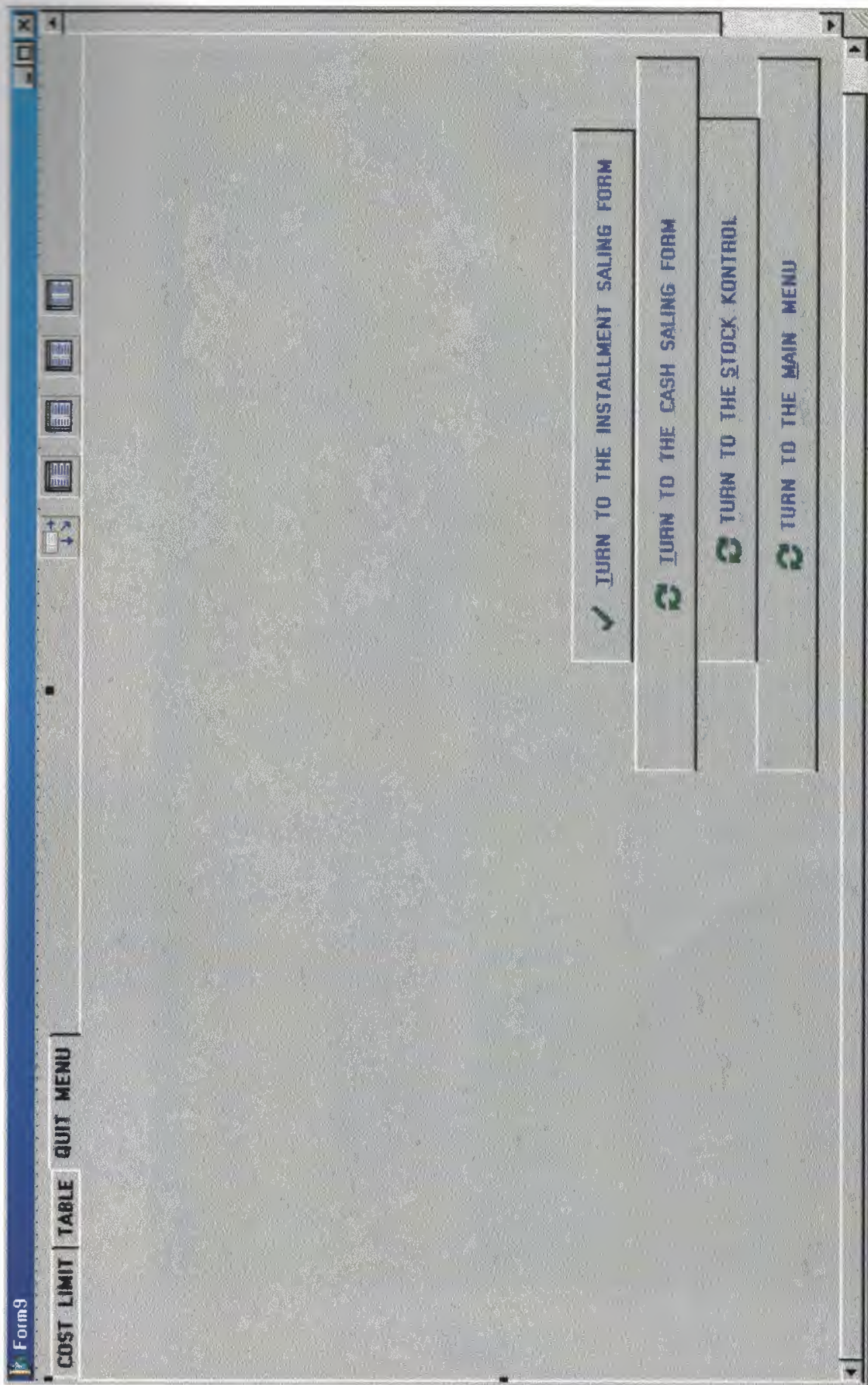


Figure 20

Form10

RECEIVED PAYMENT PROCESS PAGE

Enter the "LIST NUMBER" of INSTALLMENT EVENT and then click to the DO PROCESS or if you want you can click to the CANCEL button to cancel the PAY BILL.

ENTER THE LIST NUMBER	PEACE OF PAYMENT	DATE OF PAYMENT
		28.12.2003

Please sign the number of payment

<input type="checkbox"/> PAYMENT 1	<input type="checkbox"/> PAYMENT 4	<input type="checkbox"/> PAYMENT 7	<input type="checkbox"/> PAYMENT 10
<input type="checkbox"/> PAYMENT 2	<input type="checkbox"/> PAYMENT 5	<input type="checkbox"/> PAYMENT 8	<input type="checkbox"/> PAYMENT 11
<input type="checkbox"/> PAYMENT 3	<input type="checkbox"/> PAYMENT 6	<input type="checkbox"/> PAYMENT 9	<input type="checkbox"/> PAYMENT 12

DO PROCESS

X CANCEL THE PROCESS

Form14

TURN TO MAIN MENU

MAIN STOK REPORT PREVIEW

CASH SAILING STOK REPORT PREVIEW

INSTALLMENT STOK REPORT PREVIEW

Figure 22

MAIN STOK REPORT

<u>L. NO</u>	<u>PRODUCT NAME</u>	<u>P. CODE</u>	<u>P. SIZE</u>	<u>PIECE</u>	<u>B. PRICE</u>	<u>S. PRICE</u>	<u>B. PLACE</u>
1	POLICE	005	12x12	1	20000000	500000000	ISTANBUL
2	POLICE	001	10x10	6	10000000	25000000	ISTANBUL
3	RAIWAN	002	20x20	18	25000000	50000000	ISTANBUL
4	SUN	003	15x15	1	20000000	40000000	ISTANBUL
5	ROWER	004	12x12	11	10000000	45000000	ISTANBUL
6	RIVER	006	20x20	-1	1000000	3000000	ANKARA
7	METRO	007	10x10	28	5000000	25000000	ANKARA
8	POLICE	008	20x20	20	50000000	100000000	IZMIR
9	FLY	009	12x12	15	35000000	75000000	ISTANBUL
10	RAIBAN	010	18x18	15	40000000	120000000	ISTANBUL
11	METRO	011	9x9	50	5000000	25000000	IZMIR
12	POLICE	012	15x15	15	20000000	50000000	IZMIR
13	RIWER	013	20x20	12	25000000	70000000	ANKARA
14	POLICE	014	15x15	30	20000000	60000000	ISTANBUL
15	POLICE	0015	10x10	17	10000000	30000000	ISTANBUL

Figure 23

CASH SAILINGS PRINT PREVIEW

<u>L. NO</u>	<u>PRODUCT NAME</u>	<u>CUSTOMER NAME</u>	<u>PIECE</u>	<u>S. PRICE</u>	<u>T. PRICE</u>	<u>SAILING DATE</u>
1	RAIWAN	MEHMET YÜREKLI	1	50000000	50000000	10.04.2003
2	RIVER	CEMALIYE RENKLI	2	80000000	160000000	11.04.2003
3	ROWER	BAHRI ERDEMLI	1	30000000	30000000	12.04.2003
4	SUN	BEYAZ GÜLEÇ	3	150000000	450000000	13.04.2003
5	POLICE	FIRAT KARA	1	10000000	10000000	04.12.2003
6	POLICE	YEŞİM GÜL	1	1250000000	1250000000	04.12.2003
7	RIVER	ALI YÜCE	2	25000000	5000000	04.12.2003
8	RIVER	MEHMET AK	1	150000000	150000000	06.12.2003
9	SUN	FATMA FIDAN	2	40000000	80000000	09.12.2003

Figure 24

INSTALLMENT SAILINGS PRINT PREVIEW

L. NO	PRODUCT NAME	CUSTOMER NAME	PIECE	S. PRICE	T. PRICE	SAILING DATE
1	POLICE	AHMET GÜNEŞ	5	30000000	150000000	02.12.2003
2	ROWER	VASFIYE ÇETİN	2	45000000	90000000	04.12.2003
3	ROWER	ALI YEŞİL	3	45000000	135000000	06.12.2003
4	POLICE	HAKAN VELİ	2	500000000	1000000000	07.12.2003
5	POLICE	ALİ AY	1	500000000	500000000	08.12.2003
6	RIVER	A.B.C. LTD. ŞT.	5	3000000	15000000	10.12.2003
7	METRO	GÜL KIRMIZI	2	25000000	50000000	10.12.2003
8	METRO	HAKAN ALİ	3	30000000	90000000	13.12.2003
9	ROWER	MELTEM BEYAZ	3	45000000	135000000	15.12.2003
10	RIVER	CIHAN BEY	1	3000000	3000000	21.12.2003
11	RAIBAN	VEDAT YILDIRIM	5	120000000	600000000	21.12.2003

Figure 25

VII.REFERENCES

My referances are www.delphiturk.com and delphi book wr'ter by Memik Yanik. Introduction to flowchart K.Inman. Computer algorithm Introduction to desing and analysis 3nd Edition Sara Baafe, Allein Van Gelder. Modern system analysis and desing 3nd Edition by Jefferey A.Heoffer.