NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

TELEPHONE EXCHANGE BILLING SYSTEM IN DELPHI

Graduation Project
COM-400

Student: Okay Gezici (950293)

Supervisor: Mr. Ümit İlhan

Nicosia- 2002

ACKNOWLEDGEMENT

All my thanks to those who endlessly encouranged me through out my undergraduate studies and provided me with all kind of support financially or moralling.Who stands beside me at every moment of my life,My Parent? Specially to my brother in law MEHMET ÇELİK. Feeling proud to decicate this project for them together with my respectful brother Cemal, Murat, Serdar, Yousaf ,Bülent and Erhan.

I would like to express my faithful thanks to my University with all its educational staff and my instructors who have been a good guide for me,specially my supervisor **Mr. Ümit İLHAN** who provided me with Valuable advises and help to achieve my graduation project besides being an example of the responsible teacher.

Thanks to all my friends and school mates who joined me and shared my work sportingly.

ABSTRACT

As the 'Information age' has affected every aspect of our life, the sake for computerizing many information systems became an important need.Consequently,one of the major branches that are contributed efficiently by information revoulation is the 'database management systems'.

This project is concerned of using computer program in Telphone Exchange Billing System. It composed and written using Delphi 5 Language, which is considered as one of, the best strongest Visual Languages.

This project is a complete Telephone Exchange Billing program,which covers all the services necessary for the Telphone system, such as, costumers related information, unit, billing arrangement and classifying.

TABLE OF CONTENTS

INTRODUCTION

Now a day's the computer science both hardware and software is being developed over the previous years, programming is always providing the scients by a systematic development, in my project I did construct special programmed related to Telephone Exchange Billing System. By using a computer every things became easier then before and if developed the humans in all it's range, now you can do what every you want in a short period. I'm advising everybody to increase his knowledge about the computer.

The steps that I have followed in achieving these goals are summarized below:

First of all, I went to Telsim and I met the Manager of Information Process and I get form him Information about progress. Such as,how to add new customer and how we could find information about them.By obtaining the answers for these questions I managed to built up an idea about how the operation runs over there,as a result I translated the operaion into a software programming.

DELPHI

TmainMenu

Use TMainMenu to provide the main menu for a form. To begin designing a menu, add a main menu to a form, and double-click the component.

TMainMenu introduces properties and methods for merging the drop-down menus of the main menu with the main menu of another form, and for assisting in the menu negotiation process of an OLE container.

TPopupMenu

Use TPopupMenu to define the pop-up menu that appears when the user clicks on a control with the right mouse button. To make a pop-up menu available, assign the TPopupMenu object to the control's PopupMenu property.

If the popup menu's ParentBiDiMode is True, the popup menu's BiDiMode is set to the BiDiMode of the control that activates it. If a control cannot be found, the BiDiMode is set to Application.BiDiMode.

The popup menu's BiDiMode affects all of its menu items.

Ttimer

Is used to simplify calling the Windows API timer functions SetTimer and KillTimer, and to simplify processing the WM_TIMER messages. Use one timer component for each timer in the application.

Timer properties and methods affect the functionality of the timer by providing information for the timer event. This information includes the timer interval, which corresponds to the parameter for the Windows API SetTimer function. The actual execution of the timer occurs through its OnTimer event.

Tlabel

TLabel is not a descendant of TWinControl, so it does not have its own window and can't receive direct input from the keyboard. To add an object to a form that can respond to keyboard input (other than setting focus to another object when an accelerator key is typed) in addition to displaying text, use TStaticText.

To add an object to a form that displays text that a user can scroll or edit, use

TcomboBox

A TComboBox component is an edit box with a scrollable drop-down list attached to it. Users can select an item from the list or type directly into the edit box.

Tedit

Use a TEdit object to put a standard Windows edit control on a form. Edit controls are used to retrieve text that users type. Edit controls can also display text to the user.

When only displaying text to the user, choose an edit control to allow users to select text and copy it to the Clipboard. Choose a label object if the selection capabilities of an edit control are not needed.

TEdit implements the generic behavior introduced in TCustomEdit. TEdit publishes many of the properties inherited from TCustomEdit, but does not introduce any new behavior. For specialized edit controls, use other descendant classes of TCustomEdit or derive from it.

TbitBtn

Bitmap buttons exhibit the same behavior as button controls. Use them to initiate actions from forms and dialog boxes.

Bitmap buttons have additional properties that are used to specify the bitmap images, and their appearance and placement on the button. You can choose from predefined bitmap buttons styles or use your own, customized bitmap for the button. Although the button can be associated with only one bitmap, the bitmap (glyph property) can be subdivided into four equal-sized portions, which are displayed based on the state of the button: up, down, and disabled.

Note: Speed buttons can use the fourth part of the glyph with their fourth state called "stay down".

The Kind property of TBitBtn provides commonly used buttons, such as OK, Cancel, Help, and so on. These predefined button types have appropriate graphical images and default behaviors, so you can easily add them to your application with little or no coding necessary.

TgroupBox

The TGroupBox component represents a standard Windows group box, used to group related controls on a form. When another control component is placed within a group box, the group box becomes the parent of that component.

Ttable

Use TTable to access data in a single database table using the Borland Database Engine (BDE). TTable provides direct access to every record and field in an underlying database table, whether it is from Paradox, dBASE, Access, FoxPro, an ODBC-compliant database, or an SQL database on a remote server, such as InterBase, Oracle, Sybase, MS-SQL Server, Informix, or DB2. A table component can also work with a subset of records within a database table using ranges and filters.

At design time, create, delete, update, or rename the database table connected to a TTable by right-clicking on the TTable and using the pop-up menu.

TdataSource

Use TDataSource to provide a conduit between a dataset and data-aware controls on a form that enable display, navigation, and editing of the data underlying the dataset.

All datasets must be associated with a data source component if their data is to be displayed and manipulated in data-aware controls. Similarly, each data-aware control needs to be associated with a data source component in order for the control to receive and manipulate data.

Data source components also link datasets in master-detail relationships.

TDBNavigator

Use the database navigator on forms that contain data-aware controls, such as TDBGrid or TDBEdit. TDBNavigator provides the user with control over the dataset when editing or viewing the data.

When the user chooses one of the navigator buttons, the appropriate action occurs on the dataset the navigator is linked to. For example, if the user clicks the Insert button, a blank record is inserted in the dataset.

TDBNavigator can show any or all of the following buttons:

Button Purpose

First Sets the current record to the first record in the dataset, disables the First and Prior buttons, and enables the Next and last buttons if they are disabled.

Prior Sets the current record to the previous record and enables the Last and Next buttons if they are disabled.

Next Sets the current record to the next record and enables the First and Prior buttons if they are disabled.

Last Sets the current record to the last record in the dataset, disables the Last and Next buttons, and enables the First and Prior buttons if they are disabled.

Insert Inserts a new record before the current record, and sets the dataset into Insert and Edit states.

Delete Deletes the current record and makes the next record the current record.

Edit Puts the dataset into Edit state so that the current record can be modified.

Post Writes changes in the current record to the database.

Cancel Cancels edits to the current record, restores the record display to its condition prior to editing, turns off Insert and Edit states if they are active.

Refresh Refreshes the buffered data in the associated dataset. For TQuery components, this button is disabled unless the RequestLive property is True.

TPanel

Use TPanel to put an empty panel on a form. Panels provide properties for providing a bevelled border around the control, as well as methods to help manage the placement of child controls embedded in the panel.

TPanel implements the generic behavior introduced in TCustomPanel. TPanel publishes many of the properties inherited from TCustomPanel, but does not introduce any new behavior.

Tquery

Use TQuery to access one or more tables in a database using SQL statements. Query components can be used with remote database servers (such as Sybase, SQL Server, Oracle, Informix, DB2, and InterBase), with local tables (Paradox, InterBase, dBASE, Access, and FoxPro), and with ODBC-compliant databases.

Query components are useful because they can

Access more than one table at a time (called a "join" in SQL).

Automatically access a subset of rows and columns in its underlying table(s), rather than always returning all rows and columns.

Note: TQuery is of particular importance to the development of scalable database applications. If there is any chance that an application built to run against local databases will be scaled to a remote SQL database server in the future, use TQuery components from the start to ensure easier scaling later.

TDBGrid

Put a TDBGrid object on a form to display and edit the records from a database table or query. Applications can use the data grid to insert, delete, or edit data in the database, or simply to display it.

At runtime, users can use the database navigator (TDBNavigator) to move through data in the grid, and to insert, delete, and edit the data. Edits that are made in the data grid are not posted to the underlying dataset until the user moves to a different record or closes the application.

TDBGrid implements the generic behavior introduced in TCustomDBGrid. TDBGrid publishes many of the properties inherited from TCustomDBGrid, but does not introduce any new behavior.

TQRLabel component

Use TQRLabel is much like a regular TLabel, only for reports. Use it to print static text on a report by entering text in the Caption property.

You an also change the text to be printed in the OnPrint event, making it easy to print calculations or other information.

TQRLabel inherits properties, methods and events from TQRCustomLabel

TQRShape component

TQRShape is used to draw simple shapes like rectangles, circles and lines on a report.

Set the type of shape you want in the Shape property.

Set the pen color and style you want in the Pen property.

Set the fill style and color you want in the Brush property.

DATABASES

Money.DBF

FIELD NAME	TYPE	SIZE
Gsmno	A	12
Mounth	A	2
Akontur	L	---
Tax	L	---
LMounth	L	---
MSum	L	---
MSum Letter	A	255
Year	A	4

Custoumer.DBF

FIELD NAME	TYPE	SIZE
Name	A	20
Name2	A	20
Surname	A	20
Sex	A	6
BDate	L	---
BPlace	A	20
Jop	A	20
Nations	A	20
IDNo	A	20
FName	A	20
MName	A	20
Msurname	A	20
CType	A	10
Sim No	A	16
GSMNo	A	12
Hadres1	A	50
Hadres2	A	50

Hcity	A	20
HPcode	A	5
Jadres1	A	50
Jadres2	A	50
Jcity	A	20
Jpcode	A	6

TEL.DBF

FIELD NAME	TYPE	SIZE
receivedno	A	12
dialedno	A	12
date	A	10
kontur	L	---
mounth	A	2
year	A	4

Totalmoney.DB

FIELD NAME	TYPE	SIZE
Name	A	40
surname	A	20
gsmno	A	12
mounth	A	2
mprice	L	---
year	A	4

MAIN FORM

We can control all operation by this form. We have 3 main menu option:

- Customer
- Report
- Help

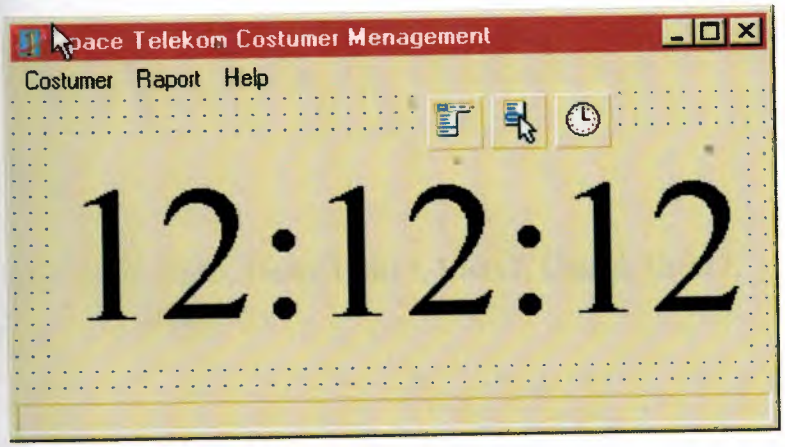
Customer Sub menu

- Add Customer
- Edit customer
- Delete Customer

Report sub menu

- Customer Report
  - * All Customer
    - * List
- Selected Customer
  - * Data
  - *Fatura
- Bill
- Help
- About

We can see all time a much on the screen



```
unit Unit8;
interface
uses
 Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
 ComCtrls, StdCtrls, ExtCtrls, Menus;
type
 Tmain = class(TForm)
  procedure Timer1Timer(Sender: TObject);
  procedure About1Click(Sender: TObject);
  procedure AddNewCostumer1Click(Sender: TObject);
  procedure EditCostumer1Click(Sender: TObject);
  procedure DeleteCostumer1Click(Sender: TObject);
  procedure SearchCostumer1Click(Sender: TObject);
  procedure ExitProgram1Click(Sender: TObject);
  procedure Fatura1Click(Sender: TObject);
  procedure Data1Click(Sender: TObject);
  procedure List1Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;
var
 main: Tmain;
implementation
uses Unit2, Unit1, Unit4, Unit5, Unit6, Unit13, Unit12, Unit16, Unit17,
 Unit9, Unit20;
{$R *.DFM}

procedure Tmain.Timer1Timer(Sender: TObject);
begin
 Label1.Caption:=timetostr(Time);
```

```
end;
procedure Tmain.About1Click(Sender: TObject);
begin
form2.show;
end;

procedure Tmain.AddNewCostumer1Click(Sender: TObject);
begin
Form1.show;
end;

procedure Tmain.EditCostumer1Click(Sender: TObject);
begin
form12.show;
end;

procedure Tmain.DeleteCostumer1Click(Sender: TObject);
begin
form13.show;
end;

procedure Tmain.SearchCostumer1Click(Sender: TObject);
begin
 Form6.Show;
end;

procedure Tmain.ExitProgram1Click(Sender: TObject);
Var
Mesaj :  Integer;
begin
Mesaj := Application.MessageBox('Do You Want To Really Quit ?',  'Q U I T', 36);
 If (Mesaj = IdYES) Then
  begin
  Main.Close;
```

```
end;
procedure Tmain.Fatura1Click(Sender: TObject);
begin
Form16.show;
end;

procedure Tmain.Data1Click(Sender: TObject);
begin
Form20.Show;
end;

procedure Tmain.List1Click(Sender: TObject);
begin
 Form17.Show;
end;
end.
```

Add New Customer

We add new customer by this procedure. We take very important data from customer ID.

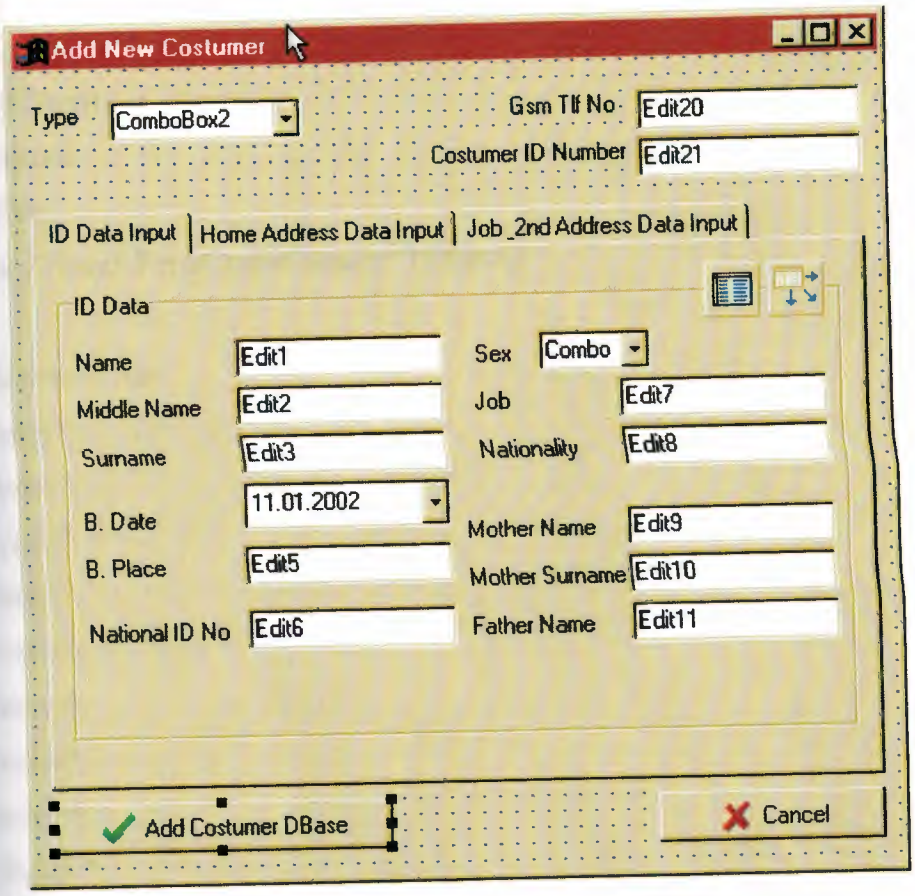
Customer ID number is GSM simcard number.

Customer two types

- Standard we must print a bill
- Ready-card we must know personal information an no print bill.

We check customer by mother surname option. When two same customer name we search by mother surname.

Main field is GSM no.




```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ComCtrls, Buttons, Db, DBTables;
type
  TForm1 = class(TForm)
    procedure FormCreate(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```
var
 Form1: TForm1;
implementation
{$R *.DFM}
procedure TForm1.FormCreate(Sender: TObject);
begin
Table1.Active:=True;
Edit1.Text:='';
Edit2.Text:='';
Edit3.Text:='';
Edit5.Text:='';
Edit6.Text:='';
Edit7.Text:='';
Edit8.Text:='';
Edit9.Text:='';
Edit10.Text:='';
Edit11.Text:='';
Edit12.Text:='';
Edit13.Text:='';
Edit14.Text:='';
Edit15.Text:='';
Edit16.Text:='';
Edit17.Text:='';
Edit18.Text:='';
Edit19.Text:='';
Edit20.Text:='';
Edit21.Text:='';
ComboBox1.Text:='';
ComboBox2.Text:='';
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
var
Mes:Integer;
```

```
begin
Mes:=Application.MessageBox('Are You Sure To Save This Data To DataBase ?
'SAVE DataBase',36);
If (Mes=IdYES) then
begin
Table1.Append;

    Table1Name.Text:=Edit1.Text;
    Table1Name2.Text:=Edit2.Text;
    Table1Surname.Text:=Edit3.Text;
    Table1BDate.Value:=DateTimePicker1.Date;
    Table1BPlace.Text:=Edit5.Text;
    Table1IDNo.text:=Edit6.Text;
    Table1Job.text:=Edit7.Text;
    Table1Nations.Text:=Edit8.Text;
    Table1MName.Text:=Edit9.Text;
    Table1MSurname.Text:=Edit10.Text;
    Table1FName.Text:=Edit11.Text;
    Table1HAdres1.Text:=Edit12.Text;
    Table1HAdres2.Text:=Edit13.Text;
    Table1HCity.Text:=Edit14.Text;
    Table1HPCode.Text:=Edit15.Text;
    Table1JAdres1.Text:=Edit16.Text;
    Table1JAdres2.Text:=Edit17.Text;
    Table1JCity.Text:=Edit18.Text;
    Table1JPCode.Text:=Edit19.Text;
    Table1GsmNo.Text:=Edit20.Text;
    Table1SimNo.Text:=Edit21.Text;
    Table1Sex.Text:=ComboBox1.Text;
    Table1CType.Text:=ComboBox2.Text;

Table1.Post;
Table1.Refresh;
```

```
    Edit1.Text:='';
    Edit2.Text:='';
    Edit3.Text:='';
    Edit5.Text:='';
    Edit6.Text:='';
    Edit7.Text:='';
    Edit8.Text:='';
    Edit9.Text:='';
    Edit10.Text:='';
    Edit11.Text:='';
    Edit12.Text:='';
    Edit13.Text:='';
    Edit14.Text:='';
    Edit15.Text:='';
    Edit16.Text:='';
    Edit17.Text:='';
    Edit18.Text:='';
    Edit19.Text:='';
    Edit20.Text:='';
    Edit21.Text:='';
    ComboBox1.Text:='';
    ComboBox2.Text:='';
end;
end;

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
 Form1.Close;
end;
end.
```

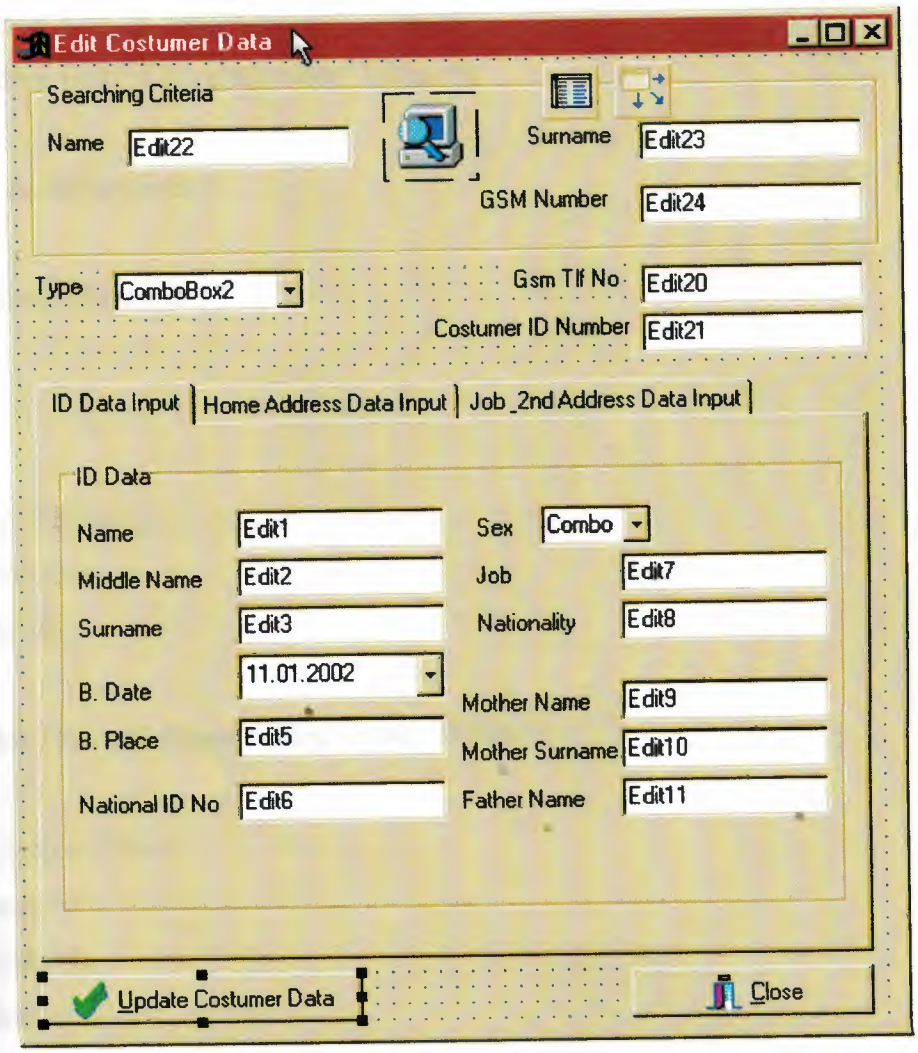
Edit Customer Data

We edit customer data by two procedure. If we wantt to edit a customer data we have 3 choice.

- Search by name
- Search by surname
- Search by GSM number

After this all data will it its area.

We can change them and when we push to "update customer data" all data will be changed.



```
unit Unit12;
interface
uses
  Windows, Messages, Classes, SysUtils, Graphics, Controls, StdCtrls, Forms,
  Dialogs, DBCtrls, DB, DBTables, ExtCtrls, ComCtrls, Buttons;
type
  TForm12 = class(TForm)
    procedure FormCreate(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure Edit22Change(Sender: TObject);
    procedure Edit23Change(Sender: TObject);
    procedure Edit24Change(Sender: TObject);
  private
    { private declarations }
  public
    { public declarations }
  end;

var
  Form12: TForm12;
implementation
{$R *.DFM}

procedure TForm12.FormCreate(Sender: TObject);
begin
Table1.Active:=True;
Edit1.Text:='';
Edit2.Text:='';
Edit3.Text:='';
Edit5.Text:='';
Edit6.Text:='';
Edit7.Text:='';
Edit8.Text:='';
Edit9.Text:='';
```

```
Edit10.Text:='';
Edit11.Text:='';
Edit12.Text:='';
Edit13.Text:='';
Edit14.Text:='';
Edit15.Text:='';
Edit16.Text:='';
Edit17.Text:='';
Edit18.Text:='';
Edit19.Text:='';
Edit20.Text:='';
Edit21.Text:='';
Edit22.Text:='';
Edit23.Text:='';
Edit24.Text:='';
ComboBox1.Text:='';
ComboBox2.Text:='';
end;

procedure TForm12.BitBtn1Click(Sender: TObject);
Var
Mesaj :  Integer;
begin
Mesaj := Application.MessageBox('Do You Want To Change This Record ?',  'Edit
Costumer Dara', 36);
 If (Mesaj = IdYES) Then
  begin
   Table1.First;
    While not Table1.Eof Do
     begin
      If (Table1Name.Text = Edit1.Text) Then
       begin
```

```
      Mesaj:= Application.MessageBox('Save Operation Sucsesfully Finished','Save
OK', 48);
      end;
     Table1.Next;
    end;
  end;
end;

procedure TForm12.Edit22Change(Sender: TObject);
begin
Table1.Open;
Animate1.Visible:=True;
Table1.First;
While not Table1.Eof Do
 Begin
 If (Table1Name.Text=Edit22.Text) then
  Begin
      Edit1.Text:=Table1Name.Text;
      Edit2.Text:=Table1Name2.Text;
      Edit3.Text:=Table1Surname.Text;
      DateTimePicker1.Date:=Table1BDate.Value;
      Edit5.Text:=Table1BPlace.Text;
      Edit6.Text:=Table1IDNo.text;
      Edit7.Text:=Table1Job.text;
      Edit8.Text:=Table1Nations.Text;
      Edit9.Text:=Table1MName.Text;
      Edit10.Text:=Table1MSurname.Text;
      Edit11.Text:=Table1FName.Text;
      Edit12.Text:=Table1HAdres1.Text;
      Edit13.Text:=Table1HAdres2.Text;
      Edit14.Text:=Table1HCity.Text;
      Edit15.Text:=Table1HPCode.Text;
      Edit6.Text:=Table1JAdres1.Text;
      Edit17.Text:=Table1JAdres2.Text;
```

```
      Edit18.Text:=Table1JCity.Text;
      Edit19.Text:=Table1JPCode.Text;
      Edit20.Text:=Table1GsmNo.Text;
      Edit21.Text:=Table1SimNo.Text;
      ComboBox1.Text:=Table1Sex.Text;
      ComboBox2.Text:=Table1CType.Text;
      Animate1.Visible:=False;
  end;
  Table1.Next;
 end;
end;

procedure TForm12.Edit23Change(Sender: TObject);
begin
Animate1.Visible:=True;
Table1.First;
While not Table1.Eof Do
 Begin
 If (Table1Surname.Text=Edit23.Text) then
  Begin
      Edit1.Text:=Table1Name.Text;
      Edit2.Text:=Table1Name2.Text;
      Edit3.Text:=Table1Surname.Text;
      DateTimePicker1.Date:=Table1BDate.Value;
      Edit5.Text:=Table1BPlace.Text;
      Edit6.Text:=Table1IDNo.text;
      Edit7.Text:=Table1Job.text;
      Edit8.Text:=Table1Nations.Text;
      Edit9.Text:=Table1MName.Text;
      Edit10.Text:=Table1MSurname.Text;
      Edit11.Text:=Table1FName.Text;
      Edit12.Text:=Table1HAdres1.Text;
      Edit13.Text:=Table1HAdres2.Text;
      Edit14.Text:=Table1HCity.Text;
```

```
      Edit15.Text:=Table1HPCode.Text;
      Edit6.Text:=Table1JAdres1.Text;
      Edit17.Text:=Table1JAdres2.Text;
      Edit18.Text:=Table1JCity.Text;
      Edit19.Text:=Table1JPCode.Text;
      Edit20.Text:=Table1GsmNo.Text;
      Edit21.Text:=Table1SimNo.Text;
      ComboBox1.Text:=Table1Sex.Text;
      ComboBox2.Text:=Table1CType.Text;
      Animate1.Visible:=False;
  end;
  Table1.Next;
 end;
end;

procedure TForm12.Edit24Change(Sender: TObject);
begin
Animate1.Visible:=True;
Table1.First;
While not Table1.Eof Do
 Begin
 If (Table1GsmNo.Text=Edit24.Text) then
  Begin
      Edit1.Text:=Table1Name.Text;
      Edit2.Text:=Table1Name2.Text;
      Edit3.Text:=Table1Surname.Text;
      DateTimePicker1.Date:=Table1BDate.Value;
      Edit5.Text:=Table1BPlace.Text;
      Edit6.Text:=Table1IDNo.text;
      Edit7.Text:=Table1Job.text;
      Edit8.Text:=Table1Nations.Text;
      Edit9.Text:=Table1MName.Text;
      Edit10.Text:=Table1MSurname.Text;
      Edit11.Text:=Table1FName.Text;
```

```
  procedure FormActivate(Sender: TObject);
 private
   { Private declarations }
 public
   { Public declarations }
 end;

var
 Form13: TForm13;
implementation
{$R *.DFM}
procedure TForm13.BitBtn1Click(Sender: TObject);
Var
Mesaj :  Integer;
begin
Mesaj:=Application.MessageBox('Do You Really Want To Delete This Costumer',
'DELETE COSTUMER', 36);
 If (Mesaj = IdYES) Then
  begin
   Table1.Delete;
   Table1.Refresh;
   Application.MessageBox('Costumer Deleted', 'D E L E T E D', 48);
  end;
end;

procedure TForm13.FormCreate(Sender: TObject);
begin
Table1.Open;
Edit1.Text:='';
Edit2.Text:='';
Edit3.Text:='';
Edit5.Text:='';
Edit6.Text:='';
Edit7.Text:='';
```

```
Edit8.Text:='';
Edit9.Text:='';
Edit10.Text:='';
Edit11.Text:='';
Edit12.Text:='';
Edit13.Text:='';
Edit14.Text:='';
Edit15.Text:='';
Edit16.Text:='';
Edit17.Text:='';
Edit18.Text:='';
Edit19.Text:='';
Edit20.Text:='';
Edit21.Text:='';
Edit22.Text:='';
Edit23.Text:='';
Edit24.Text:='';
ComboBox1.Text:='';
ComboBox2.Text:='';
end;

procedure TForm13.Edit22Change(Sender: TObject);
begin
Animate1.Visible:=True;
Table1.First;
While not Table1.Eof Do
 Begin
 If (Table1Name.Text=Edit22.Text) then
  Begin
     Edit1.Text:=Table1Name.Text;
     Edit2.Text:=Table1Name2.Text;
     Edit3.Text:=Table1Surname.Text;
     DateTimePicker1.Date:=Table1BDate.Value;
     Edit5.Text:=Table1BPlace.Text;
```

```
      Edit6.Text:=Table1IDNo.text;
      Edit7.Text:=Table1Job.text;
      Edit8.Text:=Table1Nations.Text;
      Edit9.Text:=Table1MName.Text;
      Edit10.Text:=Table1MSurname.Text;
      Edit11.Text:=Table1FName.Text;
      Edit12.Text:=Table1HAdres1.Text;
      Edit13.Text:=Table1HAdres2.Text;
      Edit14.Text:=Table1HCity.Text;
      Edit15.Text:=Table1HPCode.Text;
      Edit6.Text:=Table1JAdres1.Text;
      Edit17.Text:=Table1JAdres2.Text;
      Edit18.Text:=Table1JCity.Text;
      Edit19.Text:=Table1JPCode.Text;
      Edit20.Text:=Table1GsmNo.Text;
      Edit21.Text:=Table1SimNo.Text;
      ComboBox1.Text:=Table1Sex.Text;
      ComboBox2.Text:=Table1CType.Text;
      Animate1.Visible:=False;
  end;
  Table1.Next;
 end;
end;

procedure TForm13.Edit23Change(Sender: TObject);
begin
Animate1.Visible:=True;
Table1.First;
While not Table1.Eof Do
 Begin
 If (Table1Surname.Text=Edit23.Text) then
  Begin
      Edit1.Text:=Table1Name.Text;
      Edit2.Text:=Table1Name2.Text;
```

```
    Edit12.Text:=Table1HAdres1.Text;
    Edit13.Text:=Table1HAdres2.Text;
    Edit14.Text:=Table1HCity.Text;
    Edit15.Text:=Table1HPCode.Text;
    Edit6.Text:=Table1JAdres1.Text;
    Edit17.Text:=Table1JAdres2.Text;
    Edit18.Text:=Table1JCity.Text;
    Edit19.Text:=Table1JPCode.Text;
    Edit20.Text:=Table1GsmNo.Text;
    Edit21.Text:=Table1SimNo.Text;
    ComboBox1.Text:=Table1Sex.Text;
    ComboBox2.Text:=Table1CType.Text;
    Animate1.Visible:=False;
  end;
  Table1.Next;
 end;
end;
end.
```

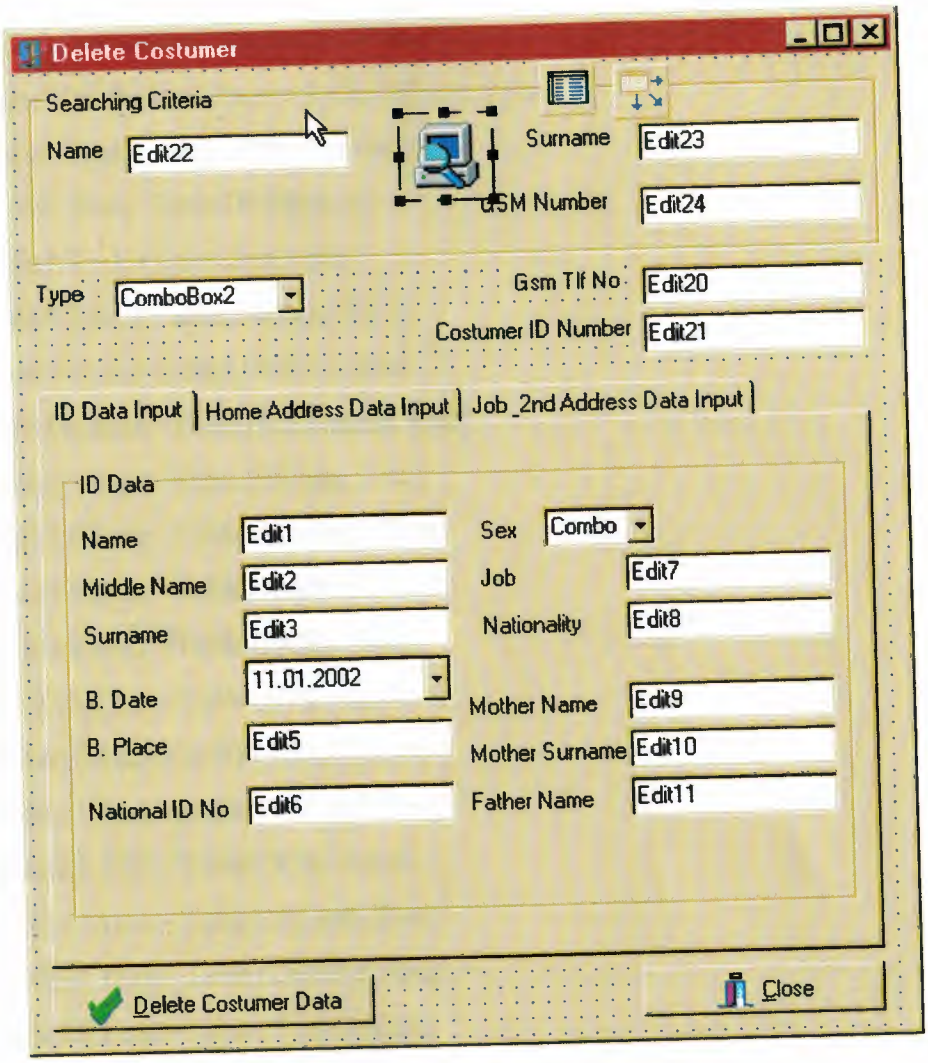
Delete Customer

When a customer cancelled GSM Telephone Number, we delete by this procedure.

At start we have three charce same edit procedure

- Search by name
- Search by surname
- Search by GSM number

After searching when we seen datas we may push to "Delete Customer data" buttons on form After confirmation, it will be deleted our database




```
unit Unit13;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ComCtrls, Db, DBTables;
type
  TForm13 = class(TForm)
    procedure BitBtn1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Edit22Change(Sender: TObject);
    procedure Edit23Change(Sender: TObject);
    procedure Edit24Change(Sender: TObject);
```

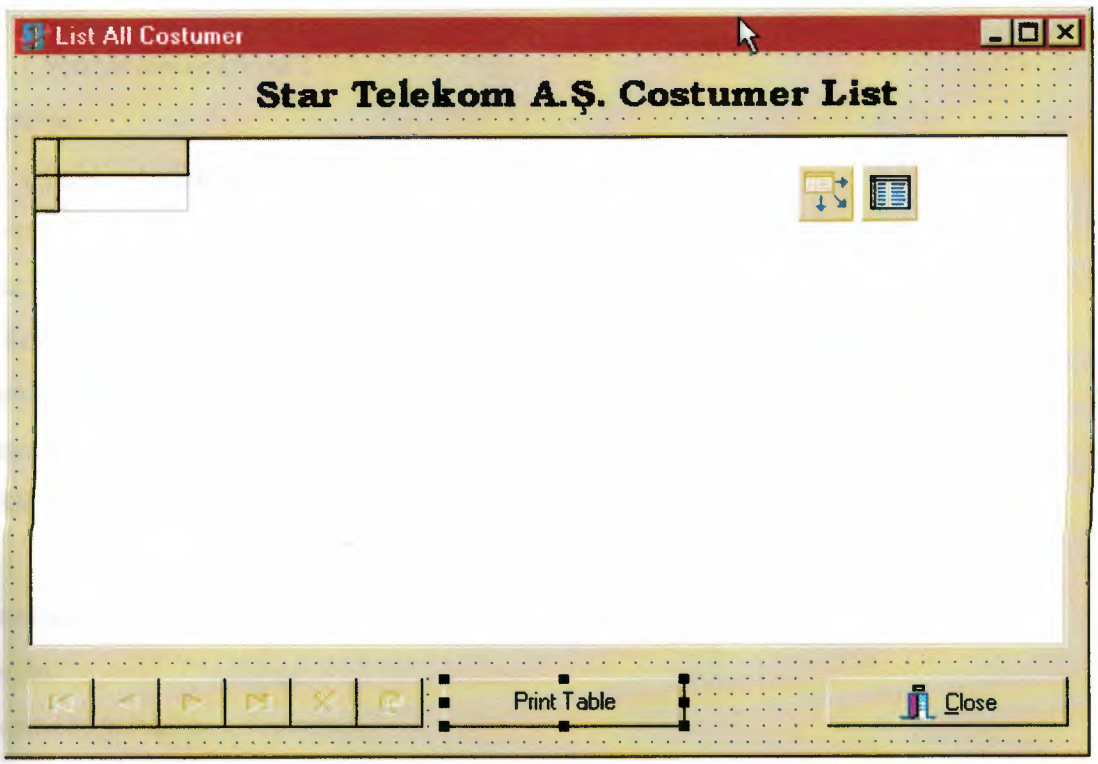
```
Edit3.Text:=Table1Surname.Text;
DateTimePicker1.Date:=Table1BDate.Value;
Edit5.Text:=Table1BPlace.Text;
Edit6.Text:=Table1IDNo.text;
Edit7.Text:=Table1Job.text;
Edit8.Text:=Table1Nations.Text;
Edit9.Text:=Table1MName.Text;
Edit10.Text:=Table1MSurname.Text;
Edit11.Text:=Table1FName.Text;
Edit12.Text:=Table1HAdres1.Text;
Edit13.Text:=Table1HAdres2.Text;
Edit14.Text:=Table1HCity.Text;
Edit15.Text:=Table1HPCode.Text;
Edit6.Text:=Table1JAdres1.Text;
Edit17.Text:=Table1JAdres2.Text;
Edit18.Text:=Table1JCity.Text;
Edit19.Text:=Table1JPCode.Text;
Edit20.Text:=Table1GsmNo.Text;
Edit21.Text:=Table1SimNo.Text;
ComboBox1.Text:=Table1Sex.Text;
ComboBox2.Text:=Table1CType.Text;
Animate1.Visible:=False;
end;
Table1.Next;
end;
end;

procedure TForm13.Edit24Change(Sender: TObject);
begin
Animate1.Visible:=True;
Table1.First;
While not Table1.Eof Do
Begin
If (Table1GsmNo.Text=Edit24.Text) then
```

```
Begin
    Edit1.Text:=Table1Name.Text;
    Edit2.Text:=Table1Name2.Text;
    Edit3.Text:=Table1Surname.Text;
    DateTimePicker1.Date:=Table1BDate.Value;
    Edit5.Text:=Table1BPlace.Text;
    Edit6.Text:=Table1IDNo.text;
    Edit7.Text:=Table1Job.text;
    Edit8.Text:=Table1Nations.Text;
    Edit9.Text:=Table1MName.Text;
    Edit10.Text:=Table1MSurname.Text;
    Edit11.Text:=Table1FName.Text;
    Edit12.Text:=Table1HAdres1.Text;
    Edit13.Text:=Table1HAdres2.Text;
    Edit14.Text:=Table1HCity.Text;
    Edit15.Text:=Table1HPCode.Text;
    Edit6.Text:=Table1JAdres1.Text;
    Edit17.Text:=Table1JAdres2.Text;
    Edit18.Text:=Table1JCity.Text;
    Edit19.Text:=Table1JPCode.Text;
    Edit20.Text:=Table1GsmNo.Text;
    Edit21.Text:=Table1SimNo.Text;
    ComboBox1.Text:=Table1Sex.Text;
    ComboBox2.Text:=Table1CType.Text;
    Animate1.Visible:=False;
  end;
  Table1.Next;
 end;
end;

procedure TForm13.FormActivate(Sender: TObject);
begin
Table1.Open;
Edit1.Text:='';
```

```
Edit2.Text:='';
Edit3.Text:='';
Edit5.Text:='';
Edit6.Text:='';
Edit7.Text:='';
Edit8.Text:='';
Edit9.Text:='';
Edit10.Text:='';
Edit11.Text:='';
Edit12.Text:='';
Edit13.Text:='';
Edit14.Text:='';
Edit15.Text:='';
Edit16.Text:='';
Edit17.Text:='';
Edit18.Text:='';
Edit19.Text:='';
Edit20.Text:='';
Edit21.Text:='';
Edit22.Text:='';
Edit23.Text:='';
Edit24.Text:='';
ComboBox1.Text:='';
ComboBox2.Text:='';
end;

end.
```

List All Customer

We take all customer list by two procedure we may take print in two form.




```
unit Unit17;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Db, DBTables, StdCtrls, Buttons, ExtCtrls, DBCtrls, Grids, DBGrids;
type
  TForm17 = class(TForm)
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    BitBtn1: TBitBtn;
    Table1: TTable;
    DataSource1: TDataSource;
```

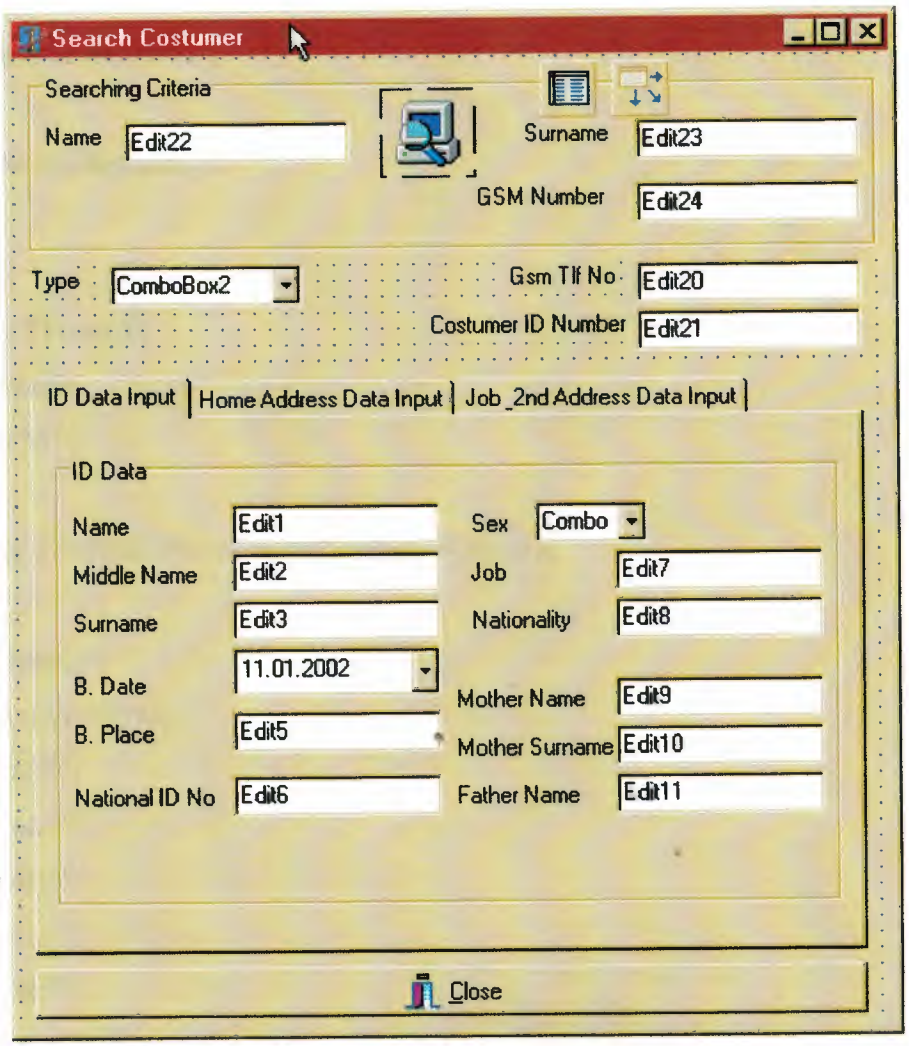
```
    Button1: TButton;
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form17: TForm17;
implementation
uses Unit18;
{$R *.DFM}

procedure TForm17.Button1Click(Sender: TObject);
begin
Form18.QuickRep1.Preview;
end;

procedure TForm17.FormCreate(Sender: TObject);
begin
Table1.Active:=True;
end;

end.
```

Search Customer

If we want to only search a one customer we must use this procedure

We have 3 charge same

- Search by name
- Search by surname
- Search by GSM number




```
unit Unit20;
interface
uses
```

```
 Windows, Messages, Classes, SysUtils, Graphics, Controls, StdCtrls, Forms,
 Dialogs, DBCtrls, DB, DBTables, ExtCtrls, ComCtrls, Buttons;
type
 TForm20 = class(TForm)
  procedure FormCreate(Sender: TObject);
  procedure Edit22Change(Sender: TObject);
  procedure Edit23Change(Sender: TObject);
  procedure Edit24Change(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);
 private
  { private declarations }
 public
  { public declarations }
 end;
var
 Form20: TForm20;
implementation
{$R *.DFM}

procedure TForm20.FormCreate(Sender: TObject);
begin
Table1.Open;
Table1.Active:=True;
Edit1.Text:='';
Edit2.Text:='';
Edit3.Text:='';
Edit5.Text:='';
Edit6.Text:='';
Edit7.Text:='';
Edit8.Text:='';
Edit9.Text:='';
Edit10.Text:='';
Edit11.Text:='';
Edit12.Text:='';
```

```
Edit13.Text:='';
Edit14.Text:='';
Edit15.Text:='';
Edit16.Text:='';
Edit17.Text:='';
Edit18.Text:='';
Edit19.Text:='';
Edit20.Text:='';
Edit21.Text:='';
Edit22.Text:='';
Edit23.Text:='';
Edit24.Text:='';
ComboBox1.Text:='';
ComboBox2.Text:='';
end;

procedure TForm20.Edit22Change(Sender: TObject);
begin
Table1.Open;
Animate1.Visible:=True;
Table1.First;
While not Table1.Eof Do
 Begin
 If (Table1Name.Text=Edit22.Text) then
  Begin
     Edit1.Text:=Table1Name.Text;
     Edit2.Text:=Table1Name2.Text;
     Edit3.Text:=Table1Surname.Text;
     DateTimePicker1.Date:=Table1BDate.Value;
     Edit5.Text:=Table1BPlace.Text;
     Edit6.Text:=Table1IDNo.text;
     Edit7.Text:=Table1Job.text;
     Edit8.Text:=Table1Nations.Text;
     Edit9.Text:=Table1MName.Text;
```

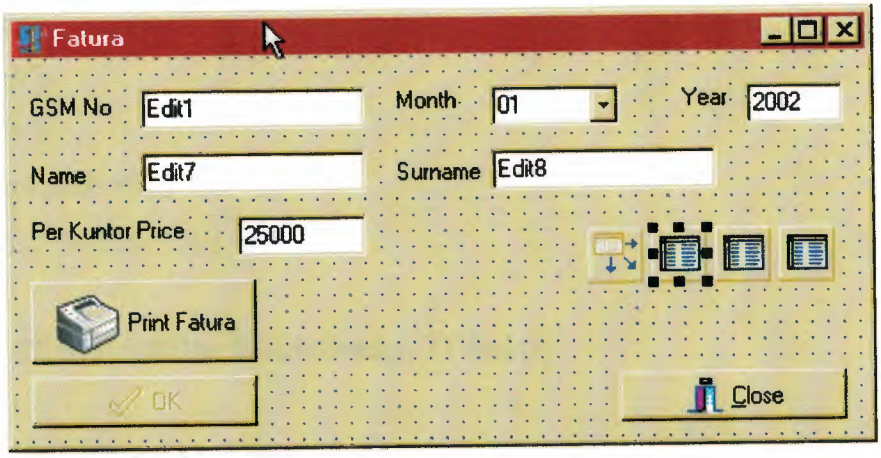
```
    Edit10.Text:=Table1MSurname.Text;
    Edit11.Text:=Table1FName.Text;
    Edit12.Text:=Table1HAdres1.Text;
    Edit13.Text:=Table1HAdres2.Text;
    Edit14.Text:=Table1HCity.Text;
    Edit15.Text:=Table1HPCode.Text;
    Edit6.Text:=Table1JAdres1.Text;
    Edit17.Text:=Table1JAdres2.Text;
    Edit18.Text:=Table1JCity.Text;
    Edit19.Text:=Table1JPCode.Text;
    Edit20.Text:=Table1GsmNo.Text;
    Edit21.Text:=Table1SimNo.Text;
    ComboBox1.Text:=Table1Sex.Text;
    ComboBox2.Text:=Table1CType.Text;
    Animate1.Visible:=False;
  end;
  Table1.Next;
 end;
end;

procedure TForm20.Edit23Change(Sender: TObject);
begin
Animate1.Visible:=True;
Table1.First;
While not Table1.Eof Do
 Begin
 If (Table1Surname.Text=Edit23.Text) then
  Begin
      Edit1.Text:=Table1Name.Text;
      Edit2.Text:=Table1Name2.Text;
      Edit3.Text:=Table1Surname.Text;
      DateTimePicker1.Date:=Table1BDate.Value;
      Edit5.Text:=Table1BPlace.Text;
      Edit6.Text:=Table1IDNo.text;
```

```
      Edit7.Text:=Table1Job.text;
      Edit8.Text:=Table1Nations.Text;
      Edit9.Text:=Table1MName.Text;
      Edit10.Text:=Table1MSurname.Text;
      Edit11.Text:=Table1FName.Text;
      Edit12.Text:=Table1HAdres1.Text;
      Edit13.Text:=Table1HAdres2.Text;
      Edit14.Text:=Table1HCity.Text;
      Edit15.Text:=Table1HPCode.Text;
      Edit6.Text:=Table1JAdres1.Text;
      Edit17.Text:=Table1JAdres2.Text;
      Edit18.Text:=Table1JCity.Text;
      Edit19.Text:=Table1JPCode.Text;
      Edit20.Text:=Table1GsmNo.Text;
      Edit21.Text:=Table1SimNo.Text;
      ComboBox1.Text:=Table1Sex.Text;
      ComboBox2.Text:=Table1CType.Text;
      Animate1.Visible:=False;
   end;
   Table1.Next;
  end;
end;

procedure TForm20.Edit24Change(Sender: TObject);
begin
Animate1.Visible:=True;
Table1.First;
While not Table1.Eof Do
 Begin
 If (Table1GsmNo.Text=Edit24.Text) then
  Begin
      Edit1.Text:=Table1Name.Text;
      Edit2.Text:=Table1Name2.Text;
      Edit3.Text:=Table1Surname.Text;
```

```
    DateTimePicker1.Date:=Table1BDate.Value;
    Edit5.Text:=Table1BPlace.Text;
    Edit6.Text:=Table1IDNo.text;
    Edit7.Text:=Table1Job.text;
    Edit8.Text:=Table1Nations.Text;
    Edit9.Text:=Table1MName.Text;
    Edit10.Text:=Table1MSurname.Text;
    Edit11.Text:=Table1FName.Text;
    Edit12.Text:=Table1HAdres1.Text;
    Edit13.Text:=Table1HAdres2.Text;
    Edit14.Text:=Table1HCity.Text;
    Edit15.Text:=Table1HPCode.Text;
    Edit6.Text:=Table1JAdres1.Text;
    Edit17.Text:=Table1JAdres2.Text;
    Edit18.Text:=Table1JCity.Text;
    Edit19.Text:=Table1JPCode.Text;
    Edit20.Text:=Table1GsmNo.Text;
    Edit21.Text:=Table1SimNo.Text;
    ComboBox1.Text:=Table1Sex.Text;
    ComboBox2.Text:=Table1CType.Text;
    Animate1.Visible:=False;
  end;
  Table1.Next;
 end;
end;
```

Design Bill

We want to print a one customer bill we must use this procedure. At start up we must enter GSM no Customer search this person in Data Base and list name and surname. After this we may push "print Fatura" button then we can see Fatura Form on screen



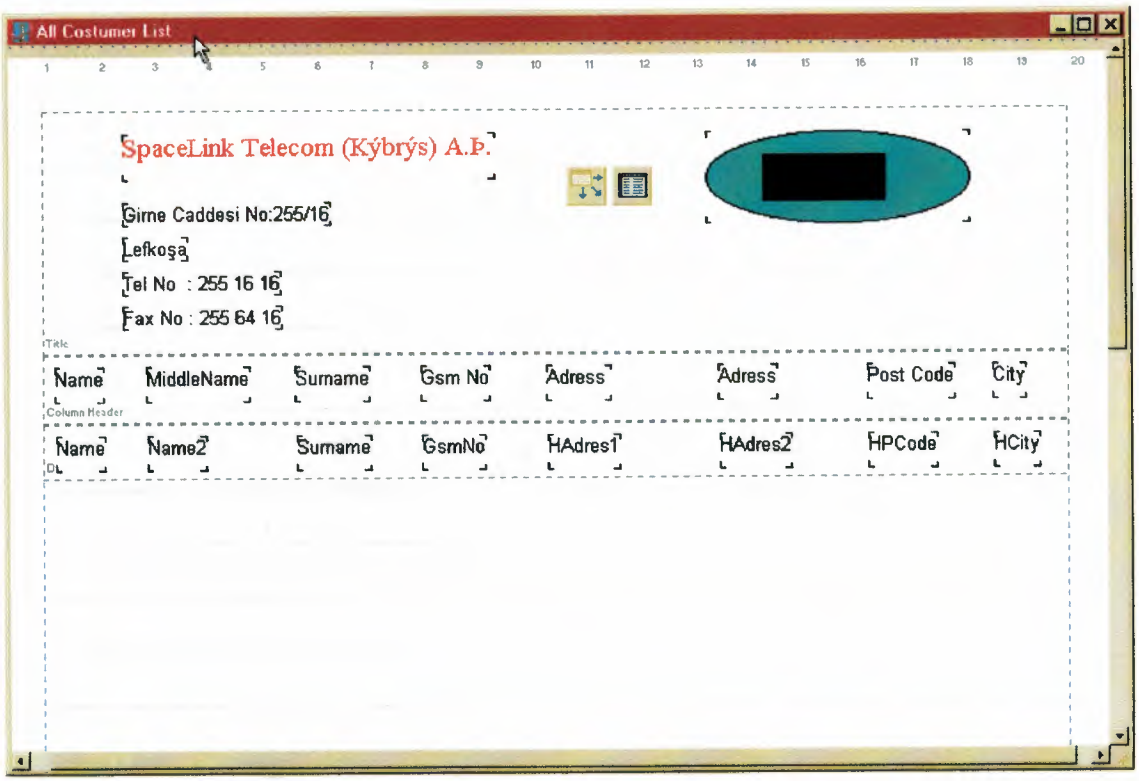

```
unit Unit16;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Db, DBTables, StdCtrls, Buttons;
type
  TForm16 = class(TForm)
    procedure FormCreate(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
 var
```

```
 Form16: TForm16;
implementation

uses Unit19;
{$R *.DFM}
procedure TForm16.FormCreate(Sender: TObject);
begin
Edit1.text:='';
Edit7.text:='';
Edit8.text:='';
ComboBox1.text:='01';
end;

procedure TForm16.BitBtn3Click(Sender: TObject);
begin
if (Edit1.text='5429876543') then
 begin
  Form19.QuickRep1.Preview;
 end;
end;

procedure TForm16.BitBtn1Click(Sender: TObject);
var
Kuntor,MPrice,TPriceYaz,Tax,LMonth,Penalty,TMonth,GTotal:Longint;
begin
Table1.Open;Table1.Active:=True;
Table2.Open;Table2.Active:=True;
Table3.Open;Table3.Active:=True;

Table1.First;
Table2.First;

While not Table2.Eof do
 Begin
```

```
If (Table2ArayanTlf.Text=Edit1.Text) then
 if (Table2Month.Text=ComboBox1.Text) then
   if (Table2Year.Text=Edit10.Text) then
 Begin
     Kuntor:=Kuntor+strtoint(Table2Kuntor.Text);
 end
else
 Begin
  Table2.Next;
 end;
Begin
if (strtoint(Table3LMonth.text)>0) then
 begin
  LMonth:=strtoint(Table3LMonth.Text);
  Penalty:=LMonth*10;Penalty:=(Penalty div 100);
 end
else
  begin
  LMonth:=0;
  Penalty:=0;
  end;
Begin
 MPrice:=Kuntor*strtoint(Edit7.text);
 Tax:=MPrice*13;Tax:=(Tax div 100);
 TMonth:=MPrice+Tax;
 GTotal:=TMonth+LMonth+Penalty;
 {TPriceYaz:=SYaz1.Sayi(GTotal);}

 Table3.Append;
  Table3GsmNo.Text:=Edit1.Text;
  Table3Month.Text:=ComboBox1.Text;
  Table3AKuntor.Value:=Kuntor;
  Table3Tax.Value:=Tax;
  Table3MSum.Value:=TMonth;
```

```
  Table3LMonth.Value:=LMonth;

  Table3.Post;
  Table3.Refresh;
 end;
end;
 end;
 end;

Procedure TForm16.Edit1Change(Sender: TObject);
begin
Table1.First;
While not Table1.Eof Do
 Begin
 If (Table1GsmNo.Text=Edit1.Text) then
  Begin
      Edit7.Text:=Table1Name.Text;
      Edit8.Text:=Table1Surname.Text;
   end;
   Table1.Next;
  end;
 end;
 end.
```

Customers Report

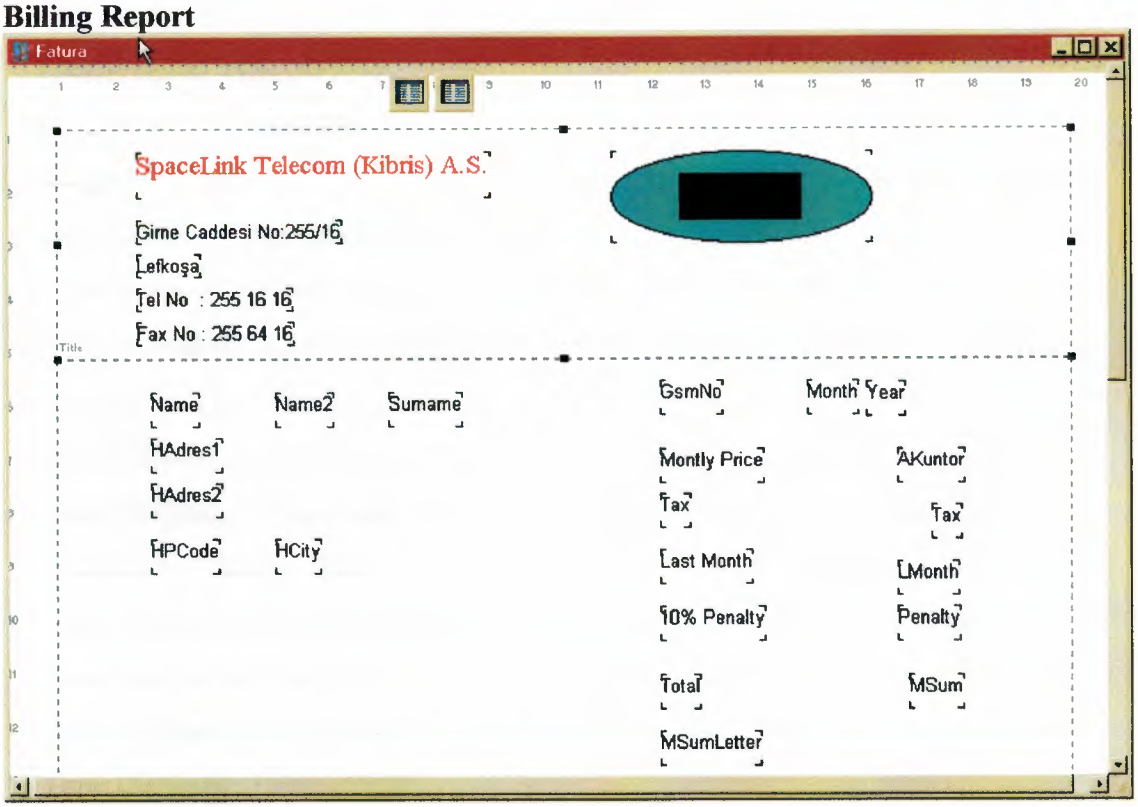



```
unit Unit18;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Db, DBTables, Qrctrls, QuickRpt, ExtCtrls;
type
  TForm18 = class(TForm)
    QuickRep1: TQuickRep;
    QRBand1: TQRBand;
    QRLabel1: TQRLabel;
    QRLabel2: TQRLabel;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRShape1: TQRShape;
    QRLabel3: TQRLabel;
```

```
QRBand2: TQRBand;
QRLabel7: TQRLabel;
QRLabel8: TQRLabel;
QRLabel9: TQRLabel;
QRLabel10: TQRLabel;
QRLabel11: TQRLabel;
QRLabel12: TQRLabel;
QRLabel13: TQRLabel;
QRLabel14: TQRLabel;
Table1: TTable;
Table1Name: TStringField;
Table1Name2: TStringField;
Table1Surname: TStringField;
Table1Sex: TStringField;
Table1BDate: TDateField;
Table1BPlace: TStringField;
Table1Job: TStringField;
Table1Nations: TStringField;
Table1IDNo: TStringField;
Table1FName: TStringField;
Table1MName: TStringField;
Table1MSurname: TStringField;
Table1CType: TStringField;
Table1SimNo: TStringField;
Table1GsmNo: TStringField;
Table1HAdres1: TStringField;
Table1HAdres2: TStringField;
Table1HCity: TStringField;
Table1HPCode: TStringField;
Table1JAdres1: TStringField;
Table1JAdres2: TStringField;
Table1JCity: TStringField;
Table1JPCode: TStringField;
DataSource1: TDataSource;
```

```
    QRBand3: TQRBand;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
    QRDBText5: TQRDBText;
    QRDBText6: TQRDBText;
    QRDBText7: TQRDBText;
    QRDBText8: TQRDBText;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form18: TForm18;

implementation

{$R *.DFM}

procedure TForm18.FormCreate(Sender: TObject);
begin
Table1.Refresh;
Table1.Open;
end;

end.
```

Billing Report




```
unit Unit19;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Db, DBTables, Qrctrls, QuickRpt, ExtCtrls;
type
  TForm19 = class(TForm)
    QuickRep1: TQuickRep;
    QRBand1: TQRBand;
    QRLabel1: TQRLabel;
    QRLabel2: TQRLabel;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRLabel3: TQRLabel;
    QRShape1: TQRShape;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
```

```
QRDBText3: TQRDBText;
QRDBText4: TQRDBText;
QRDBText5: TQRDBText;
Table1: TTable;
Table1Name: TStringField;
Table1Name2: TStringField;
Table1Surname: TStringField;
Table1Sex: TStringField;
Table1BDate: TDateField;
Table1BPlace: TStringField;
Table1Job: TStringField;
Table1Nations: TStringField;
Table1IDNo: TStringField;
Table1FName: TStringField;
Table1MName: TStringField;
Table1MSurname: TStringField;
Table1CType: TStringField;
Table1SimNo: TStringField;
Table1GsmNo: TStringField;
Table1HAdres1: TStringField;
Table1HAdres2: TStringField;
Table1HCity: TStringField;
Table1HPCode: TStringField;
Table1JAdres1: TStringField;
Table1JAdres2: TStringField;
Table1JCity: TStringField;
Table1JPCode: TStringField;
QRDBText6: TQRDBText;
QRDBText7: TQRDBText;
Table2: TTable;
QRDBText8: TQRDBText;
QRDBText9: TQRDBText;
QRDBText10: TQRDBText;
QRDBText11: TQRDBText;
```

```
    QRLabel7: TQRLabel;
    QRLabel8: TQRLabel;
    QRLabel9: TQRLabel;
    QRLabel10: TQRLabel;
    QRDBText12: TQRDBText;
    QRLabel11: TQRLabel;
    QRDBText13: TQRDBText;
    QRDBText14: TQRDBText;
    QRDBText15: TQRDBText;
    Table2GsmNo: TStringField;
    Table2Month: TStringField;
    Table2AKuntor: TIntegerField;
    Table2Tax: TIntegerField;
    Table2LMonth: TIntegerField;
    Table2MSum: TIntegerField;
    Table2Ok: TStringField;
    Table2MSumLetter: TStringField;
    Table2Year: TStringField;
    QRBand2: TQRBand;
    QRDBText16: TQRDBText;
    Table2Penalty: TIntegerField;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form19: TForm19;

implementation
{$R *.DFM}
end.
```

CONCLUSION

In this program I goal to Telephone Exchange Billing System. For this reason choosen the Delphi programming . Here I provided the recording and removing new custumer, found information about customer and billing. It is also more practical and useful together which is easiy friendly use.

When I prepared this project I understood Delphi Programming is very useful, there are more advantages than other programs. As a conclusion I can that with the help of this program which written by Delphi 5 programming.

REFERENCES

[1] Memik Yanık, Delphi 5, Beta, İstanbul, 2001

[2] Marco Cantu Delphi 4, Alfa, Bursa, 1999

[3] Marco Cantu Delphi 5, Alfa, Bursa, 2000

[4] Memik Yanık, Borland Delphi 5, İstanbul, 1999