NEAR EAST UNIVERSITY

FaG(~'lty of Engineering

Departm~nfof Computer Engineering

TELEPHONE EXCHANGE BILLING PROGRAM

Graduation Project
COM -400

Prepared By: Erkan Hakverdi (950457)

Supervisor : Umit ilhan

Lefkoşa - 2000

CONTENTS

ACKNOWLEDGEMENTS

First of all I want to thanks my parents, becaıse of their finite support, helps and moral. And also to my brothers and sisters they helped rne and they accept all my wishes.

To my friends Salih Soysal, Ahmet Nalbant and Mahmut Beke because they give all possibility, all helps and even their heme to me, And Especially to my advisor Mr. Umit Ilhan beeaıse of his helps and support.

To an my friends that they helped and moralise me during this projects.

ABSTRACT

This is my graduation prejeeı. The preject is Telephone Exchaage Billing. Program ta.kes randomly some phone number. This numbers are used two places, First is dialing number. Second is dialed number. The dialing number caHsthe dialed number.

There are some features with this number. These are the dialed number never can call his number. And never called two or more time at the same time. And also the dialed number cannot be dialing number when the it as dialed number.

Fer printing, there is two print out seleetion one is normal bin that is contain only unit price, total duration and total price. The other is detailed bill that is contain date of calli.ng,dialed' number duration of calling.

These are some imortant netice and feanıre for this program. And the others are explained in the main body of program.

INTRODUCTION

The telephone exchange billing program is a program to take bill for a dialer. The bill is in known date intervals which he/she called, or it can be used for periodical time interval (e.g. monthly) to calculate and bill duration ofa dialer.

The program has two main purposes. üne is behavior as machine. This machine has some functions. When a dialer call somewhere the machine shows those; dialer phone, dialed phone, starting time and ending time. These functions are done by apart of program and called simulation.

On the other hand when simulation part working the program keeps these information on a daily database file. The file is a binary file creating by program, when the program runs first time.

The other purpose is calculating the duration in periodic or requested time interval for a dialer. The calculation is done by multiplying total minutes with a unit price, which is a constant value until the user changes it.

THE TELEPHONE EXCHANGE BILLING PROGRAM

The program is written in Delphi version 4.0. this programming language is a visual programming language and one of the best programming language.

A simple way of describing Delphi is a sophisticated Pascal compiler. Delphi' s roots lie in Turbo Pascal, introduced in the mid-1980s. This view of Delphi, however, doesn't capture the real power of Delphi. Object Pascal, the object-oriented extensions to Pascal, is the underlying language of Delphi. The Visual Component Library, or VCL, is a hierarchy of Object Pascal objects that allow you to design programs. A better way of describing Delphi is an Object Pascal-based visual development environrnent.

The VCL is intimately tied to the Delphi IDE, and is what gives you the ability to quickly develop applications. The Component palette and Object Inspector allow you to drop VCL components on forms and then manipulate the properties and events of those controls without having to write a single line of code.

Despite its name, the VCL is not entirely made up of visual components: In fact, of the over 600 objects in the VCL, most are not visual. The Delphi IDE allows you .to visually add some nonvisual components to your programs. For example, if you wanted to write a database application that connected to a table, you would drop a TDataSource component on your form. TDataSource is a nonvisual component, but is represented on the form by an icon (which doesn't show up at runtime), and you can manipulatethe properties and events of TDataSource in the Object Inspector just as you would a visual control.

All VCL objects, and in fact all objects in Object Pascal, are derived from Tübject. Tübject is unique in that it is an abstract object that has no properties or events, only methods that allow you to derive objects from this base class. Use Tübject as the immediate base class when writing simple objects that are not components. Components are objects that you can manipulate at design time. All components in the VCL are derived from the abstract component type TComponent. The VCL components you will likely use the most are the VCL's controls, such as TForm or TSpeedButton. Controls are visual components derived from the abstract component type TControl.

You can use Delphi to create Object Pascal objects without using the VCL, although by creating any objects in Object Pascal, both your objects and VCL objects will share a common ancestor in Tübject. However, by deriving new objects from VCL object, much of the work in designing applications is done for you by Delphi. For example, if you wanted to use a progress bar in your application but didn't like TProgressBar, Delphi's control that creates a progress bar, you could create a new object based on TProgressBar and override its properties, events, or methods.

Most of the objects you use in Delphi are components you can see at both design time and runtime, such as edit boxes and string grids. A few, such as common dialog boxes, are components you don't see at design time, but they appear at runtime. Still others, such as timers and data source components, never have any visual representation in your application at runtime, but they are there for your application to use.

Occasionally you might want to create your own nonvisual objects for your application. For example, you might want to create a TEmployee object that contains Name, Title, and HourlyPayRate fields. You could then add a CalculatePay method that uses the data in the HourlyPayRate field to compute a paycheck amount for a given period.

The TEmployee type declaration could look like this:

```
type
  TEmployee = class(TObject)
   Name: string[25],·
   Title: string[25],·
   HourlyRate: Double,·
  function CalculatePayAmount: Double;
  end,·
```

In this case, TEmployee is derived from Tübject. TEmployee contains three fields and one method. Because it is derived from Tobject, TEmployee also contains all methods of TObject (Tübject has no fields),

Place object type declarations you create without the help of Delphi in the type declaration part of your unit along with the form type declaration.

In the variable declaration part of the unit, you need to declare a variable of the newtype:

```
var
  Employee: TEmployee;
```

TEmployee is just an object type. An actual object doesn't exist in memory until the object is instantiated, or created, by a constructor call. A constructor is a method that allocates memory for the new object and points to the new object, which is called an instance of the object type. Calling the Create method, the constructor, assigns this instance to a variable.

If you want to declare an instance of the TEmployee type, your code must call Create before you can access any of the fields of the object:

```
Employee := TEmployee. Create;
```

In the TEmployee type declaration, there isn't a Create method, but because TEmployee is derived from TObject, and TObject has a Create method, Tlimployee, can call Create to create a TEmployee instance, which is assigned to the Employee variable.

Now you are ready to access the fields of an Employee object, just as you would any other Delphi object

VCL

Each component you put on a form .becomes a user-interface object, a database object, ora system function, such asa system timer. By placing components on forms, you build the interface of your application.

Components are objects in the true object-oriented programming (OOP) sense. Because they are objects, Delphi components

Encapsulate some set of data and data-access functions Inherit data and behavior from the objects they are derived from (called their ancestors)

Operate interchangeably with other objects derived from a common ancestor, throıgh a concept called polymorphism

Although each component-has its own unique aspects, all components share certain qualities they inherit from .a common ancestor object called TComponent. TComponent defines the minimum set of attributes necessary for a component to operate in the Delphi environment.

The important thing to understand about components at this point is that they contain three kinds of information:

State information. Information about the present condition of a component is called a property of the component. Properties are named attributes of a component that a user or application can read or set. Examples of component properties are Height and Color.

Action information. Components generally have certain actions they can perform on demand. These actions are defined by methods, which are procedures and functions you call in code to tell the component what to do. Examples of component methods are Hide and Refresh.

Feedback information. Components provide opportunities for application developers to attach code to certain occurrences, or events, making components fully interactive. By responding to events, you bring your application to life. Examples of component events are OnClick and OnKeyDown.

Delphi provides a large number of different components, many of which perform similar functions, but in specialized ways. This section is designed to help you choose among those groups of similar components. First it describes those properties common to all visual components. Then it discusses different functional groups of components, how to choose among them, and briefly describes the properties that make each component in a group unique.

A- ANALYSIS OF PROGRAM

Process of System:

The program is included to subsystem.

Simmulation of Dialer and Save the information produced by simmulation

File processes

Simmulation: is the main part of the program. In this part the program simmulate the following;

Some one calls some other one; this is the first thing that is done by the simmulation. The simmulation shows the dialer number and dialed number on the screen.

The duration of calliig. The simmulation takes the current time that is time the dialer start, in computer and shows on the screen. When dialer finished calling, the program take that time on computer but does not shows on the screen.

Totally simmulation including three colomns, dialer phone, dialed phone and time.

```
procedure TForml. Timer l Timer(Sender: TObject);
var
  i.j.k : Integer;
  randTel : Integer;
  Talking : Integer;
  wait : Longlnt;
begin
  if random(l0)=5 then begin
    randTel: =random(6) + 1;
    ifrandTel=l then begin
      Tel[MesCounter].Arayan:=Tels[l]+'      '+
        IntToStr(Random(990)+ 10)+' ';
      Tel[MesCounter ].Arayan: =Tel[Me~Counter ].Arayan+
        IntToStr(Random(90)+ 10)+' ';
      Tel[MesCounter ].Arayan: =Tel[MesCounter ].Arayan+
        IntToStr(Random(90)+ 1 0);
    end;
    if randTel=2 then begin
      Tel[MesCounter ].Arayan: =Tels [2] +' '+
        IntToStr(Random(990)+ 10)+' ';
      Tel[MesCounter ].Arayan: =Tel[MesCounter ].Arayan+
        IntToStr(Random(90)+10)+'    ';
      Tel[MesCounter ].Arayan: =Tel[MesCounter ].Arayan+
       IntToStr(Random(90)+ 1 0);
```

```
 end;
 if randTel=3  then  begin
   Tel[MesCounter ].Arayan:  =Tels[3J+' '+
    IntToStr(Random(990)   +1O)+' ';
   Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
     IntToStr(Random(90)   +1O)+' ';
   Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
    IntToStr(Random   (90)+10);
 end;
 if randTel=4  then  begin
   Tel[MesCounter].Arayan:=Tels[4]+'     '+
    IntToStr(Random(290)+200)+'    ';
   Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
     IntToStr(Random(990)+10)+'    ';
   Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
     IntToStr(Random(90)   +1O)+' ';
   Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
    IntToStr(Random(90)   +10);
 end;
 if randTel=5  then  begin
   Tel[MesCounter].Arayan:=Tels[5]+'     '+
    IntToStr(Random(90)+   1O)+'';
   Tel[MesCounter ].Arayan:  =Tel[Mes<pounter].Arayan+
   IntToStr(Random(90)   +10);
 end;
 ifrandTel=6  then  begin
   Tel[MesCounter].Arayan:=Tels[6]+'     '+IntToStr(Random(90)+   10)+' ';
   Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
    IntToStr(Random(90)+   10);
 end;
 if randTel=7  then  begin
   Tel[MesCounter].Arayan:=Tels[7]+'     '+IntToStr(Random(90)+   10)+' ';
   Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
    IntToStr(Random(90)+   10);
```

```
    end;

    randTel: =random(3) +5,·
    if randTel=5 then begin
     repeat
       Tel[MesCounter].Aranan:=Tels[5}+'     '+
        IntToStr(Random(90)   +1 O)+' ',·
       Tel[MesCounter ].Aranan: =Tel[MesCounter ].Aranan+
        IntToStr(Random(90)   +1 0),·
     until Tel[MesCounter }.Arayan<> Tel[MesCounter ].Aranan,·
    end;
    if randTel=6 then begin
     repeat
       Tel[MesCounter].Aranan:=Tels[6}+'     '+
       IntToStr(Random(90)   +1 O)+' ',·
       TelfMestlounter].Aranan: =Tel[MesCounter ].Aranan+
        IntToStr(Random(90)+   1 0);
     until Tel[MesCounter ].Arayan<> Tel[MesCounter ].Aranan,·
    end;
    if randTel=7 then begin
     repeat
       Tel[MesCounter].Aranan:=Tels[7}+'     '+IntToStr(Random(90)+   10)+' ',·
       Tel[MesCounter ].Aranan: =Tel[MesÇounter ].Aranan+
        IntToStr(Random(90)   +1 0),·
     until Tel[MesCounter }.Arayan<> Tel[MesCounter ].Aranan,·
    end;
    Tel[MesCounter ].Bassur: =Time;
    Tel[MesCounter]. Tarih: =Date;
    MesCounter: =MesCounter+ 1,·

    if MesCounter>TotShow    then inc(MesBase),·
   end;
  for wait: =0 to 10000000 do;
  refresh;
```

```
for k:=1 to TotShow-1 do begin
  canvas. TextOut(l O,k*20+50JntToStr(k));
 end;
 canvas.font. Color: =C!Maroon;
 canvas. Text0ut(50,50, 'Dialing');
 canvas. Text0ut(250,50, 'Dialed');
 canvas. Text0ut(450,50, 'Time');
j:=1; i:=MesBase; k:=0;
 canvas.Font. Color: =C!Navy;
 while (i<=MesBase+ TotShow)and(k< =MesCounter )and(j<TotShow+ 1) do
begin
  incik);
  if
(TimeToStr(Tel[k}.Bassur)<>'OO:OO:OO')and(TimeToStr(Tel[k].Bitsur)='00:00:00
')and(j<TotShow) then begin
    canvas. Text0ut(50,j*20+ 50, Tel[k}.Aranan);
    canvas. Text0ut(250,j*20+ 50, Tel[k].Arayan);
    canvas. Text0ut(450,j*20+ 50, TimeToStr(Tel[k].Bassur));
    inctj); incti);
  end;
 end;
 talking: =0;
for i:=l to MesCounter-1 do begin
  ifTimeToStr(Tel[i].BitSur)='00:00:00' then talking:=talking+ 1;
 end;
 if MesCounter>O then begin
  canvas.Font. Color: =C!Black;
  canvas. Text0ut(200, (TotShow+ 1) *20+50
    , 'Current Dialer .. : '+JntToStr(talking));
  canvas. Text0ut(450, (TotShow+l) *20+50,
   'Total Dialer .. : '+JntToStr(MesCounter-1));
end;
 if (MesCounter> l)and(random(5) =1) then begin
  randTel:=random(MesCounter-1)+ 1;
```

```
if TimeToStr(Tel[randTel].BitSur)  = '00:00:00' then begin
  Tel[randTelj.BitSur:=Now,·
  Tel[randTe l].Fark:= Tel[randTel]. BitSur-Tel [randTel]. BasSur,·
 end,·
end;
```

File Process: Creating of file is done when program runs first time. The files take data generated by the simmulation.

```
Re Write(FTel),·
far i:=1 to MesCounter do begin
  Seek(Ftel,FileSize(FTel));
  Write(FTel, Tel[i]);
end,·
La bel l. Caption: ='Time .. : '+time ToStr(now);
end;
```

System Definition:

The user to enter text or need to present text to the user. The type of control used to hold the information depends on the size and format of the information.

Edit	Edit a single line-of text.
Memo	Edit multiple lines of text.
Mask:Edit	Adhere to a particular format, such as a nine-digit postal code or a phone number.
RichEdit	Edit unlimited lines of text or use rich text format.

In the simmulation part nothing required to enter. When the program starts it is automatically generate <lata.

```
type
 TTel = record
  Arayan: String[40];
```

```
    Aranan: String[40];
    Tarih : TDate;
    Bas sur,
    BitSur,
    Fark, totFark : TTime;
  end;
  TForml = class(TForm)
    Timer 1: TTimer;
    Labell: TLabel;
    SpeedButtonl: TSpeedButton;
    SpeedButton2: TSpeedButton;
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction),·
    procedure Timer l Timer(Sender: TObject);
    procedure SpeedButtonl Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure FormActivate(Sender: TObject),·
  private
    { Private declarations}
  public
    { Public declarations}
    UnitPrice : Longint;
  end;
```

These are the main procedure that is already exists on **Delphi** and declared under variable parts.

B- FLOWCHART OF PROGRAM

SIMMULATION

BILLING

DETAILED BILLING

~DATE

~DIALED PHONE

~STARTTIME

~ EDNDING TIME

~URATION

NORMAL BILLING

PHONENO

UNITPRICE

TOTAL PRICE

C- PROCESS OF PROGRAM

Description Processes:

The program is consist of two form. When program starts the simulation form appears. The program was prepared in the 1024x768 screen resolution. Because of this when program run on other resolution it is seen on the right side on the screen.

All codes used is this program are written under the unitl. In this program each form represent by a unit. There is three button on the simulation form. Print, about and close button.

When you compile project in Delphi, an executable (.EXE) file is created that can then be run. The executable usually provides the basic functionality of program. You can extend the application by calling DLLs, packages, and other support files from the executable.

In designing a Windows EXE, there are two UI models for the implementation of the application:

Single document interface (SDI)

Multiple document interface (MDI)

In addition to the implementation model of applications, the design-time behavior of project and the run-time behavior of your application can be manipulated by setting project options in the Delphi IDE

The other form is called billing form. It has two interface one is normal billing interface In this, there are one Tedit foı:.~nteringphone number, two TmaskEdit for time interval, one command button to allow user to change the unit price. Some Tlabel s. there is special button to change the interface of billing form.

When click on it with mouse the interface changed as normal billing interface. When clicking second times it changes as detailed billing.

On detailed interface, it has TlistView. It shows the record that are generated by the simulation The titles are dialed phone, start time and end time. There is one Tlabel. It shows the total duration.

The Forms are used in ıırog:ram:

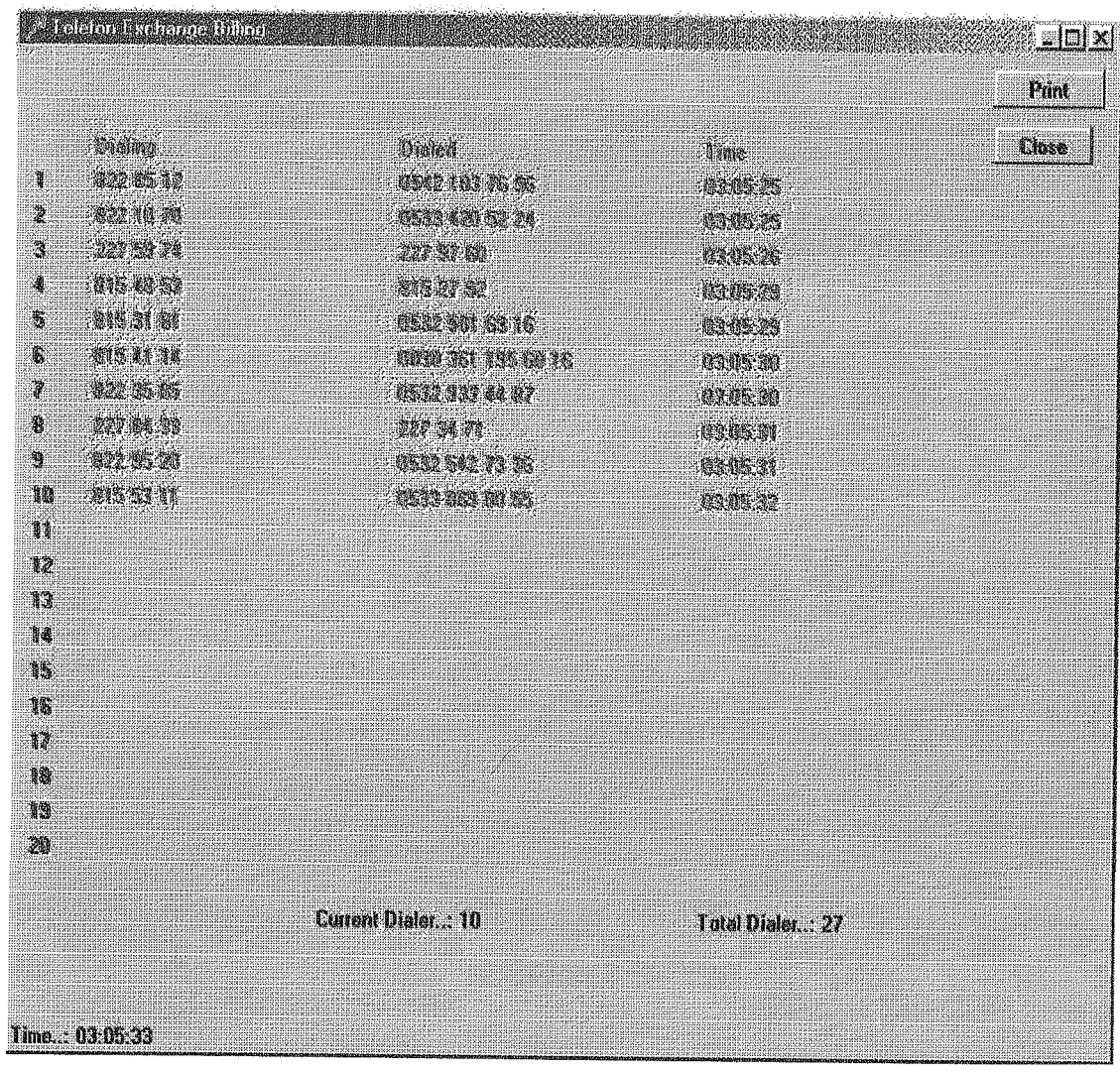


Figure 1

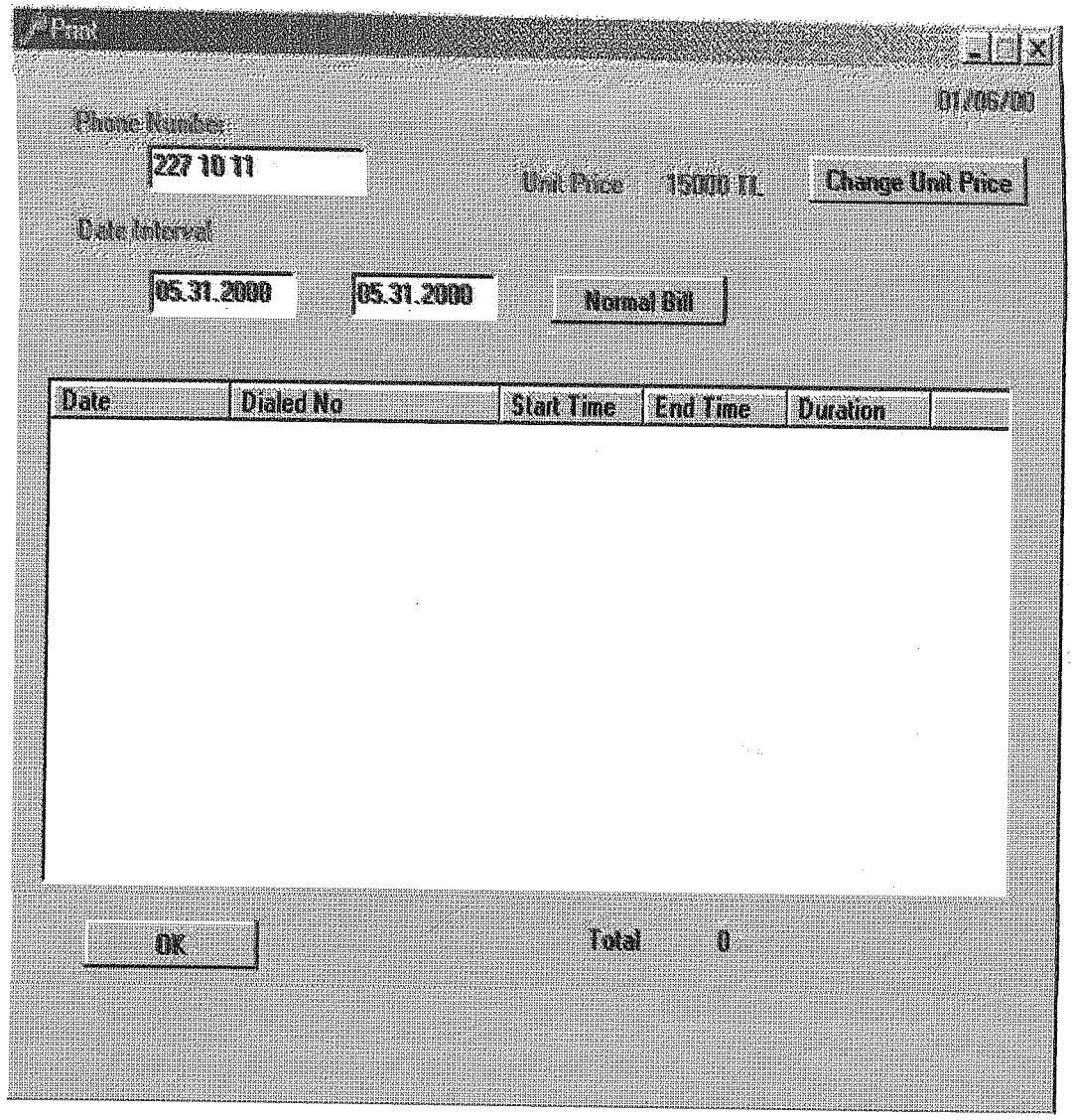

Print
01/06/00
Phone Number
227 10 11
Unit Price 15000 TL
Change Unit Price
Date Interval
05.31.2000
05.31.2000
Normal Bill
Date
Dialed No
Start Time
End Time
Duration
OK
Total 0

Figure 2.

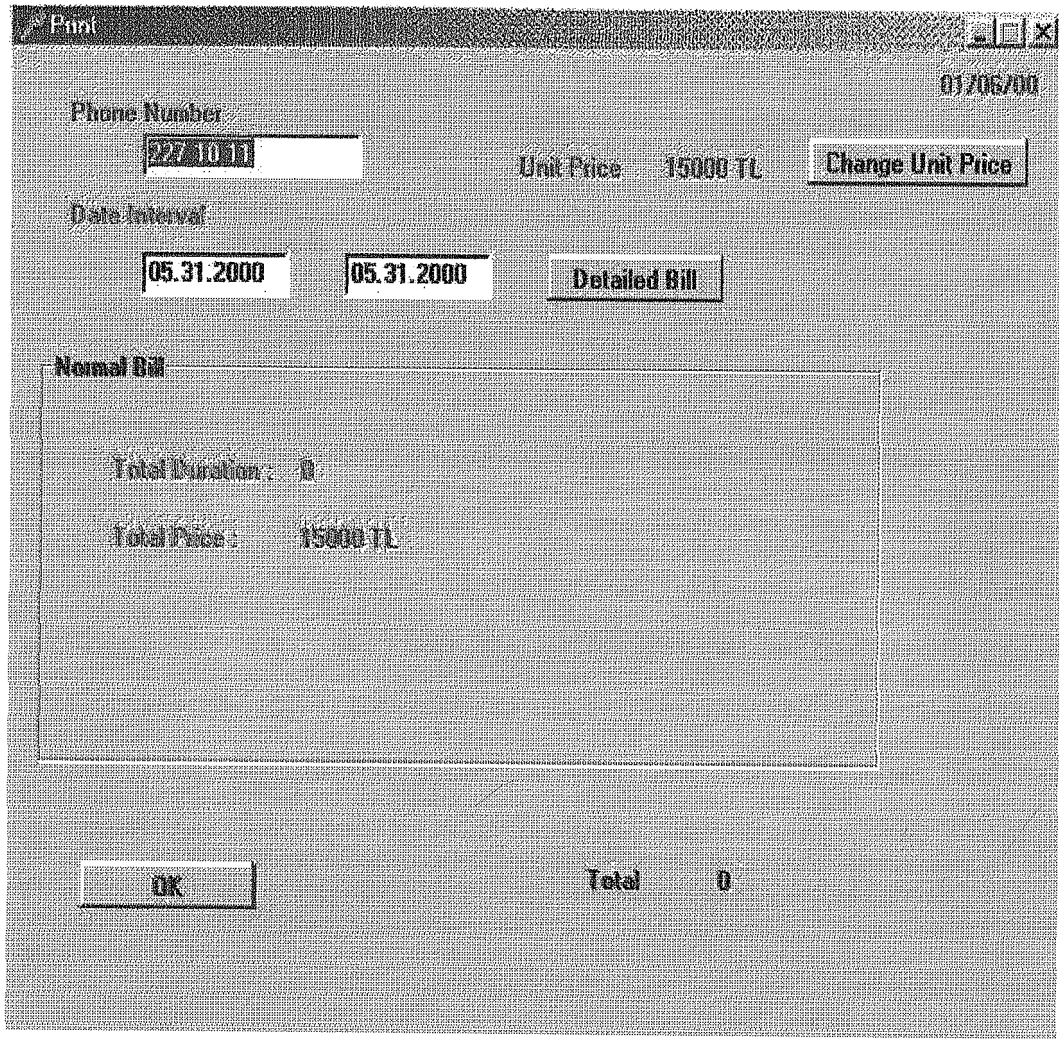

Print
01/06/00
Phone Number
227 10 11
Unit Price 15000 TL
Change Unit Price
Date Interval
05.31.2000
05.31.2000
Detailed Bill
Normal Bill
Total Duration : 0
Total Price : 15000 TL
OK
Total 0

Figure 3

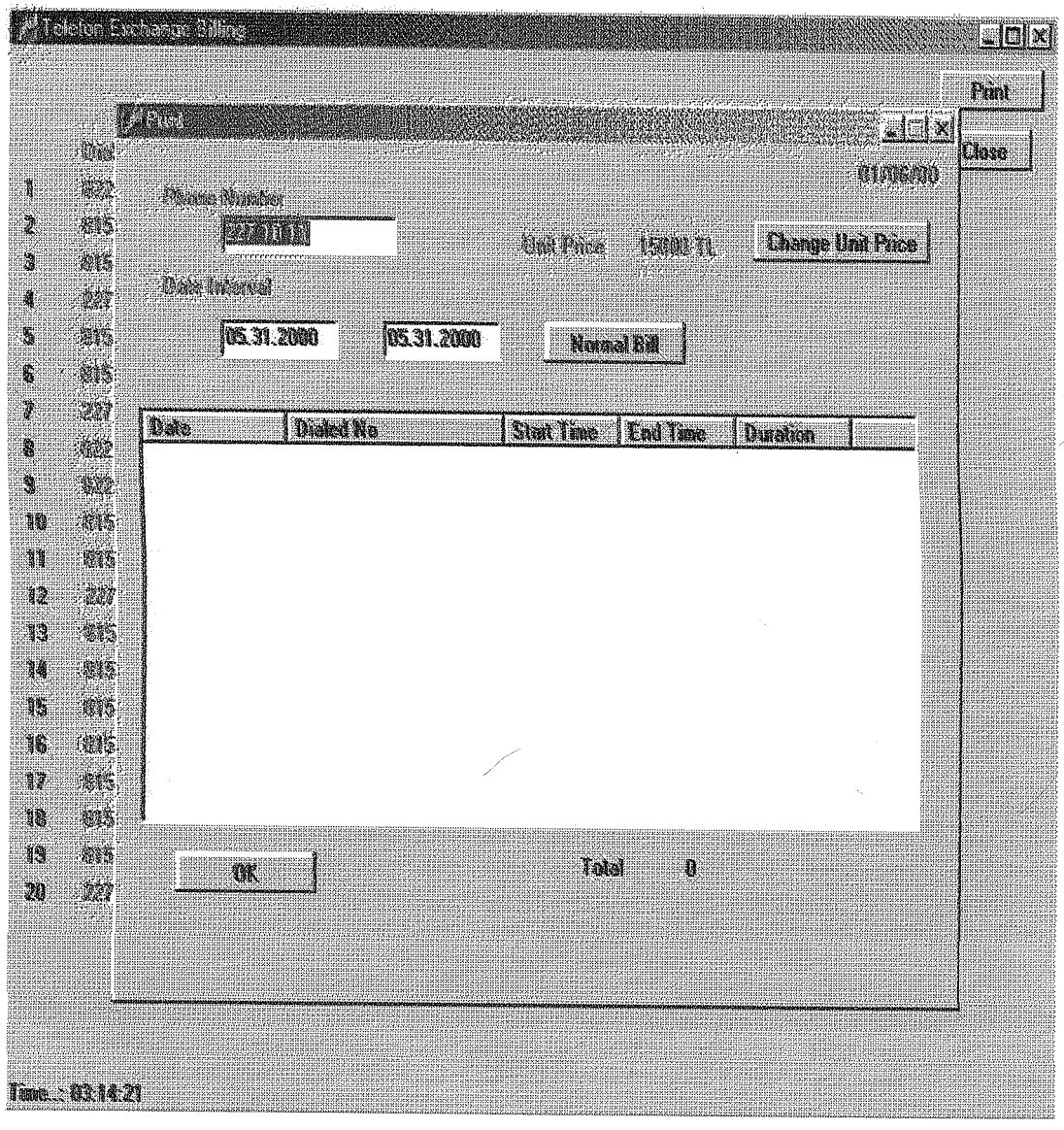

Telefon Exchange Billing
Print
Close
Print
01/06/00
Phone Number
Unit Price
Change Unit Price
Date Interval
05.31.2000
05.31.2000
Normal Bill
Date
Dialed No
Start Time
End Time
Duration
OK
Total
0
Time : 03:14:21

Figure 4

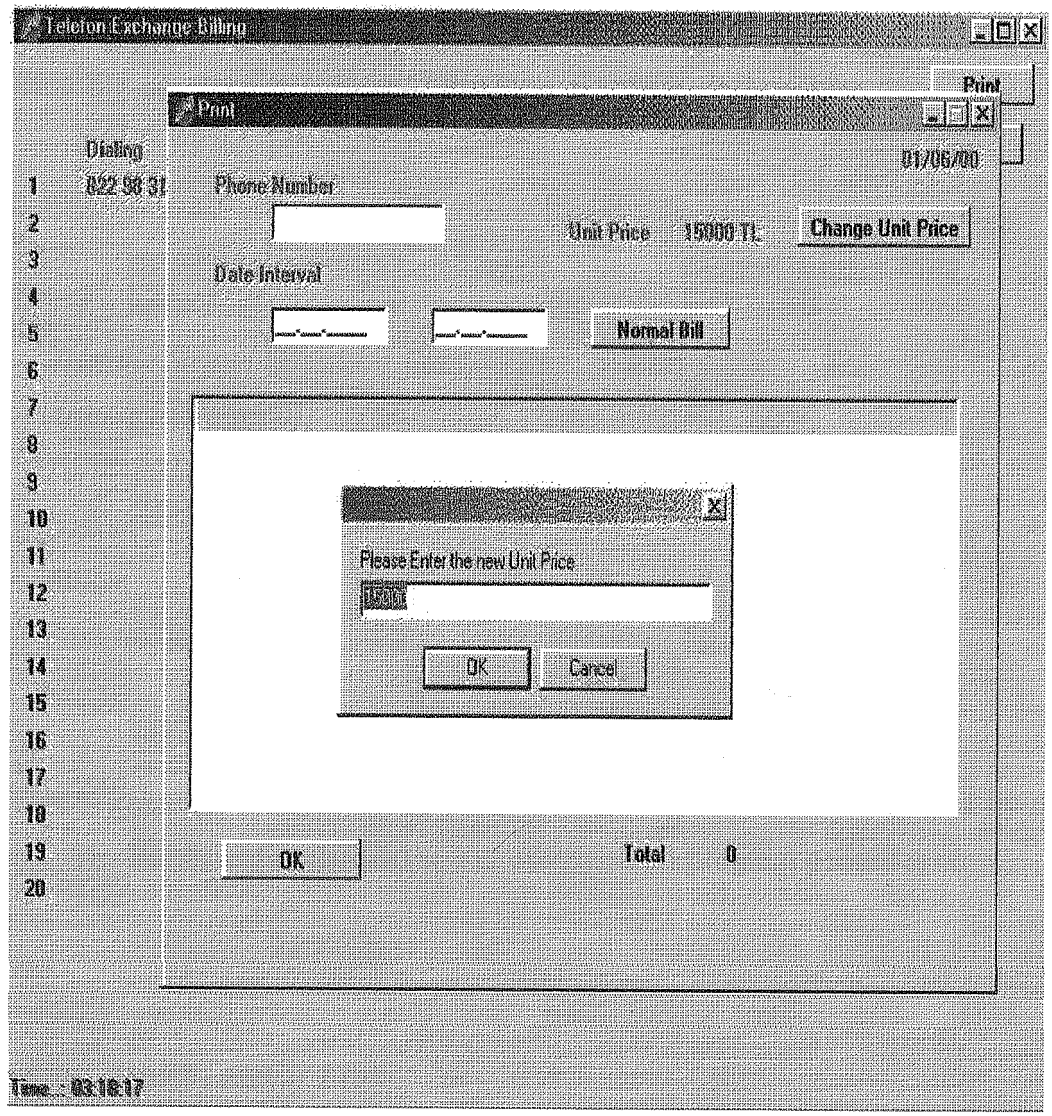

Telefon Exchange Billing
Print
Print
Dialing
822 98 31
Phone Number
01/06/00
Unit Price 15000 TL
Change Unit Price
Date Interval
Normal Bill
Please Enter the new Unit Price
OK
Cancel
OK
Total 0
Time : 03:16:17

Figure 5

D- FiLE ORGANISATION & PROCESS

Description of File:

The program uses two binary file. Binary file created by the program. The first one it is checked there exist or not. If there is no such file it is created file named "Tel.DAT" this is daily file in each day it stores the record to another file named "TelG.DAT". all ofrecords are in this file. All the program working with this file.

This file are created such this. Firstly a type was declared for these files like below.

```
Type
 TTel = record
  Arayan:  String[40];
  Aranan:  String[40];
  Tarih  : TDate;
  Bassur,
  BitSur,
  Fark,totFark:  TTime;
 end,·
```

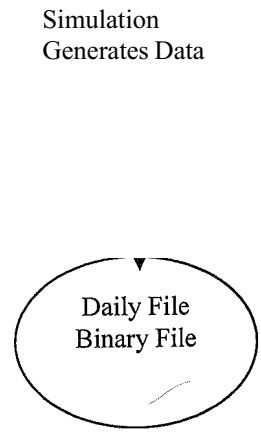
This is common for both file. TTel is a type of these file that is created by the programmer. The file should declare in the Var otherwise the program generates error.

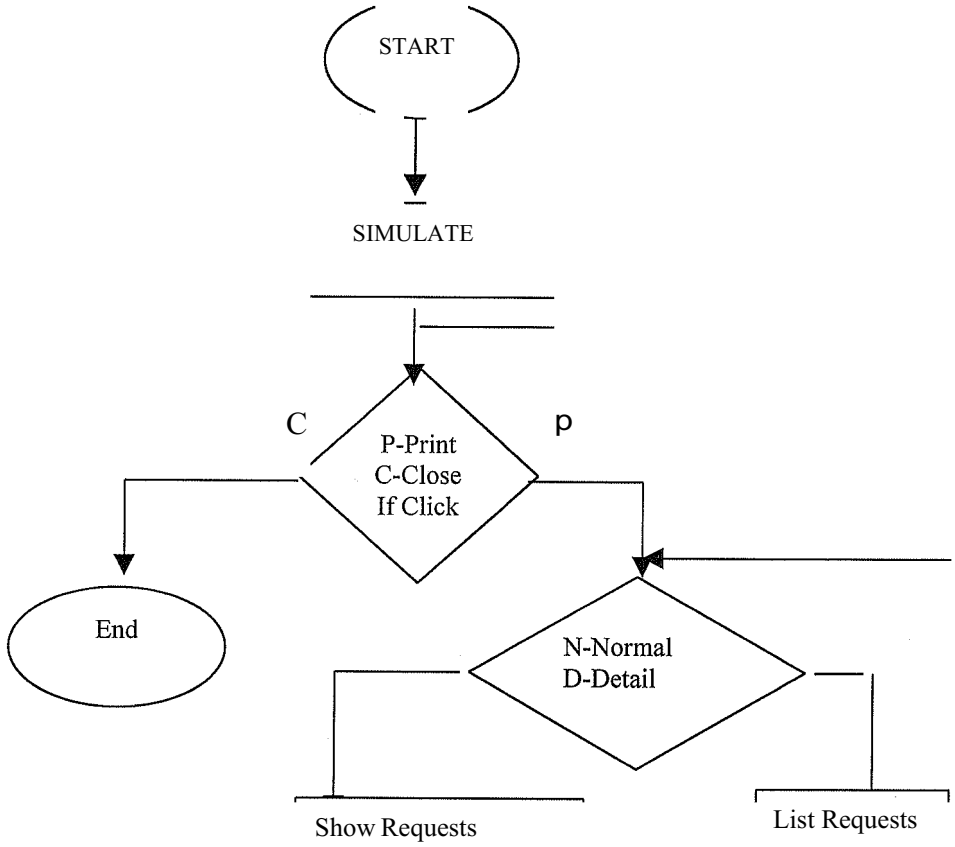
```
var
 Form]:  TForml,·
 FTel, FTelG : File of TTel;
 Tel : array[l.. TotMes] of TTel,·
 Mes  : array[l.. TotMes} of String;
 MesCounter,MesBase:  Jnteger,·
 Tels:  array[l..10}  of String;
```

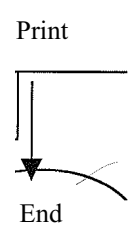
In the below, how the program check the file there exist or not. FTel and FTelG is var for these files.

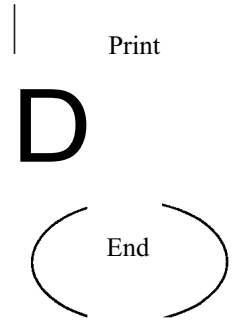
```
AssignFile(FTel,   'Tel.dat');
{$i-}Reset(FTel) ;{$i+}
if IOResult<>O  then ReWrite(FTel),·

AssignFile(FTelG,   'TelG.dat');
{$i-}Reset(FTelG);  {$i+}
if IOResult<>O  then ReWrite(FTelG);
```

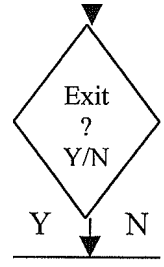
The File Process Flowchart:




START
SIMULATE
C
P-Print
C-Close
If Click
p
End
N-Normal
D-Detail
Show Requests
List Requests


Print
End


Print
D
End


Exit
?
Y/N
Y
N

File Access:

The file Ftel is a daily file. Every day it stores the records that generate by the simulation and stored on arrays. The following codes shows how the records saves in to file.

```
ReWrite(FTel);
 for i:=1 to MesCounter  do begin
   Seek(Ftel,FileSize(FTel));
    Write(FTel, Tel[i});
 end,·
```

The codes shows <lata transfer between Ftel and FtelG. It adds record after aday finish from the array to FTelG file.

```
var
  i: Integer,·
begin
 for i:=1 to MesCounter  do begin
   if (TimeToStr(Tel[i}.Bassur)<>   '00:00:00J  then begin
    if TimetoStr(Tel[i}.  Bitsur) = '00:00:O(f'then  Tel[i].BitSur:  =Now;
    Seek(FtelG,FileSize(FTelG)),·
     Write(FTelG,  Tel[i]),·
   end,·
 end,·
 CloseFile(FTelG);
 CloseFile(FTel)  ;
end;
```

Data Structure:

```
Arayan          : String[40];
Aranan          : String[40];
Tarih           : TDate;
Bassur          : TTime;
BitSur          : TTime;
Fark            : TTime;
TotFark         : TTime;

TForml          : class
Timerl          : TTimer;
Labell          : TLabel;
SpeedButtonl    : TSpeedButton;
SpeedButton2    : TSpeedButton;
UnitPrice       : Longint;

Forml           : TForml;
Ftel            : File ofTTel;
FTelG           : File of TTel;
Tel             : array[l..TotMes] ofTTel;
Mes             : array[l..TotMes] of String;
MesCounter      : Integer;
MesBase         : Integer;
Tels            : array[l..10] of String;
```

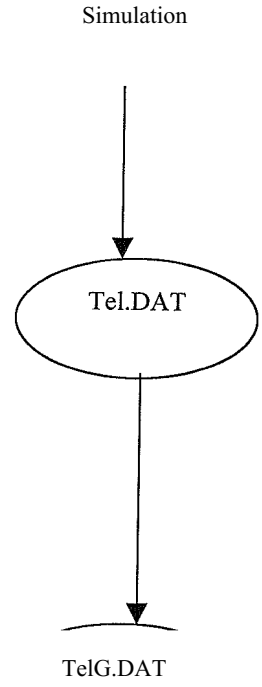
E- DATAFLOW ANALYSIS

The datas are stored in Tel. DAT files like below

```
ReWrite(FTel),·
 for  i:=1  to MesCounter   do  begin
   Seek(Ftel,FileSize(FTel));
    Write(FTel,  Tel[i));
  end;
```

when the day end or the user close the program the record in first Tel.DAT file are stored in second one called TelG.DAT.

```
begin
 for  i:=1  to MesCounter   do  begin
   if (TimeToStr(Tel[i}.Bassur)<>    '00:00:00')  then  begin
     ifTimetoStr(Tel[i].Bitsur)='00:00:00'       then  Tel[i}.BitSur:=Now;
     Seek(FtelG,FileSize(FTelG));
      Write(FTelG,  Tel[i)),·
   end;
  end;
  CloseFile(FTelG);
  CloseFile(FTel);
end;
```



F- THE PRiNTER MODULE AND OUTPUT

The printer module is a most important part of the program. Because the user can send record to the paper. The user can see the result on paper.

The codes for the printer written are below. There is one more thing when the large data are sand printer, it can not separate paper. Only one large paper required to tak:e output of such data.

```
var
  i : Integer;
 fx,jj;  : Integer;
begin
 .fx:=300; jj;:=300;
 Printer. Canvas. Font.Pitch:  =fpFixed;
```

```
Printer.BeginDoc;
Printer. Canvas.Font.Size: =18;
Printer. Canvas.Font.Size: =10;
Printer. Canvas. TextOut(fx+ 1500,fa,DateToStr(Now));
Printer.Canvas.Text0ut(fx,fa+70 , 'Phone Number: '+Editl.Text);
Printer. Canvas. TextOut(fx,fa+ 150, 'Date : '+MaskEditl. Text+' -
'+MaskEdit2. Text);
Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsbold];
if List View1. Visible then begin
 Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsbold];
 printer. Canvas. Text0ut(fx+600,fa-l 00, 'DETAILED BILL;;
 Printer. Canvas. Font. Style: =Printer. Canvas.Font.Style-[fsbold];

 Printer. Canvas. TextOut(fic ,fa+320, 'Date');
 Printer. Canvas. TextOut(fx+ 300, fa+320, 'Dialed No;;
 Printer. Canvas. Text0ut(fx+800, fa+ 320, 'Start time');
 Printer. Canvas. TextOut(fx+ 1100,fa+ 320, 'End time');
 Printer. Canvas. TextOut(fx+ 1400,fa+ 320, 'Duration');
 Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsbold];
 Printer.Canvas.Font.Size:=8;
 for i:=0 to Listi/iewl.Items.Count-l do begin
  Printer. Canvas. TextOut(fx ,fa+400+ (i*70),List View1. items [i]. Caption);
  Printer.Canvas.Text0ut(fx+300,fa+400+(i*70),
  ListView 1. items[i].Subltems[O});
  Printer. Canvas. Text0ut(fx+800 ,fa+400+(i*70),
  ListView 1. items[i}.Subltems[I });
  Printer. Canvas. TextOut(fx+ 1100,fa+400+(i*70),
 ListView 1. items[i].Subltems[2 });
  Printer. Canvas. TextOut(fic+1400,fa+400+(i*70),
 ListView 1. items[i].Subltems[3 ]);
 end;
end
else begin
 Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsbold];
```

```
  printer. Canvas. Text0ut(fx+600,fo-J  00, 'NORMAL BILL ');
  Printer. Canvas. Font.Style: =Printer. Canvas. Font.Style-[fsbold];

  Printer. Canvas. TextOut(fx,fo+ 320, 'Total Duration  : '+Label9. Caption  );
  Printer.Canvas.Text0ut(fx,fo+420,    'Unit Price     : '+Label5.Caption   );
  Printer. Canvas. TextOut(fx,fo+ 520, 'Total Price    : '+Labell 0. Caption  );
 end;
 Printer.EndDoc;
end;
```

These codes include both Normal Billing, and Detailed Billing paper order. Because in the normal bill there is only price and duration. But in the detailed billing there is date, phone number, starting time and ending time.

THE PROGRAM CODES

```
unit Unitl;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls, Buttons,·
const
  TotShow = 21;
  TotMes = 10000;
type
  TTel = record
    Arayan: String[40];
    Aranan: String[40];
    Tarih : TDate;
    Bas sur,
    BitSur,
    Fark, totFark: TTime;
  end;
  TForml = class(TForm)
    Timer 1: TTimer,·
    Labell: TLabel;
    SpeedButtonl: TSpeedButton;
    SpeedButton2: TSpeedButton;
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction),·
    procedure TimerlTimer(Sender: TObject);
    procedure SpeedButtonl Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
private
```

```
    { Private declarations}
  public
    { Public declarations}
    UnitPrice : Longint;
  end;

var
  Form]: TForml;
  FTel,FTelG : File of TTel;
  Tel: array[J .. TotMes] ofTTel;
  Mes : array[I..TotMes] o/String;
  MesCounter,MesBase : Integer;
  Tels: array[l .. 10] o/String;
implementation

uses Unit2;

{$R *.DFM}

procedure TForml.FormCreate(Sender: TObject);
begin
 AssignFile(FTel, 'Tel.dat');
 {$i-}Reset(FTel); {$i+}
 if IOResult<>Othen ReWrite(FTel);

 AssignFile(FTelG, 'Te!G.dat');
 {$i-}Reset(FTelG);{$i+}
 if IOResult<>Othen ReWrite(FTelG);

 MesCounter:=0;
 mesBase:=1;
 randomize;
 Canvas.Brush.Color:=Color;
 Tels[J]:='0542';
```

```
  Tels[2}:='0532';
  Tels[JJ:='0533';
  Tels[4}:='0090';
  Tels[5):='227';
  Tels[6}:='815';
  Tels[7]:='822';

  UnitPrice: =15000;
end;

procedure TForml.FormClose(Sender: TObject; var Action: TCloseAction),·
var
  i : Integer,·
begin
 far i:=1 to MesCounter do begin
   if (TimeToStr(Tel[i}.Bassur)<> '00:00:00~ then begin
     ifTimetoStr(Tel[i].Bitsur)= '00:00:00' then Tel[i}.BitSur: =Now;
     Seek(FtelG, FileSize (FTelG));
      Write(FTelG, Tel[i});
   end;
  end;
  CloseFile(FTelG);
  CloseFile (FTel);
end;

procedure TForml. Timer lTimer(Sender: TObject);
var
  i,j, k : Integer;
  randTel : Integer;
  Talking : Integer;
  wait : Longlnt;
begin
  ifrandom(10)=5 then begin
   randTel: =random(6) +1;
```

```
if randTel=l  then begin
  Tel[MesCounter].Arayan:=Tels[l}+'    '+IntToStr(Random(990)+10)+'   ';
  Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
    IntToStr(Random(90)+10)+'   ';
  Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
    IntToStr(Random(90)+  1O);
end,·
if randTel=2  then begin
  Tel[MesCounter].Arayan:=Tels[2}+'    '+IntToStr(Random(990)+10)+'   ';·
  Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
    IntToStr(Random(90)  +1O)+' ';·
  Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
    IntToStr(Random(90)  +1O);
end;
if randTel=3  then begin
  Tel[MesCounter].Arayan:=Tels[3}+'    '+IntToStr(Random(990)+  10)+' ';
  Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
    IntToStr(Random(90)+  10)+' ';
  Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
    IntToStr(Random(90)+  1O),·
end;
if randTel=4  then begin
  Tel[MesCounter].Arayan:=Tels[4}+'    '+IntToStr(Random(290)+200)+'   ';·
  Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
    IntToStr(Random(990)+  10)+' ';
  Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
   IntToStr(Random(90)  +1O)+' ';
  Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
    IntToStr(Random(90)+  1O);
end;
if randTel=5  then begin
  Tel[MesCounter].Arayan:=Tels[5}+'    '+IntToStr(Random(90)+  10)+' ';
  Tel[MesCounter ].Arayan:  =Tel[MesCounter }.Arayan+
    IntToStr(Random(90)+  1O);
```

```
end;
ifrandTe/=6 then begin
  Tel[MesCounter].Arayan:=Tels[6}+'     '+IntToStr(Random(90)+   10)+' ';
  Tel[MesCounter ].Arayan:  =Tel[MesCounter ].Arayan+
    IntToStr(Random(90)   +10);
end;
if randTe/=7  then begin
  Tel[MesCounter].Arayan:=Tels[7]+'     '+IntToStr(Random(90)+10)+'    ';
  Tel[MesCounter ].Arayan:  =Tel[MesCounter }.Arayan+
    IntToStr(Random(90)+   10);
end;

randTel:=random(J)+   5;
ifrandTel=5 then begin
 repeat
   Tel[MesCounterj.Aranan:=Tels[5]+'     '+IntToStr(Random(90)+   10)+' ';
   Tel[MesCounter }.Aranan:  =Tel[MesCounter }.Aranan+
      IntToStr(Random(90)+   10);
 until Tel[MesCounter }.Arayan<>  Tel[MesCounter }.Aranan;
end;
ifrandTel=6 then begin
 repeat
   Tel[MesCounter}.Aranan:=Tels[6}+11+IntT0Str(Random(90)+10)+'      ';
   Tel[MesCounter }.Aranan:  =Tel[MesCounter }.Aranan+
      IntToStr(Random(90)   +10);
 until Tel[MesCounter }.Arayan<>  Tel[MesCounter }.Aranan;
end;
if randTel=7  then begin
 repeat
   Tel[MesCounterj.Aranan:=Tels[7}+'     '+lntToStr(Random(90)+   1O)+'';
   Tel[MesCounter ].Aranan:  =Tel[MesCounter }.Aranan+
      IntToStr(Random(90)   +10);
 until Tel[MesCounter }.Arayan<>  Tel[MesCounter ].Aranan;
end;
```

```
  Tel[MesCounter ].Bassur: =Time;
  Tel[MesCounter].Tarih:=Date;
  MesCounter: =MesCounter+ 1;

  if MesCounter>TotShow  then inc(MesBase);
 end,·
far wait:=0 to 10000000 do;
 refresh;
far k:=1 to TotShow-1 do begin
  canvas. TextOut(l O,k*20+50,JntToStr(k));
 end;
 canvas.font. Color: =C!Maroon;
 canvas. Text0ut(5 0, 50, 'Dialing');
 canvas. Text0ut(250,50, 'Dialed;;
 canvas.Text0ut(450,50, 'Time');
ji=l; i:=MesBase; k:=0;
 canvas.Font. Color: =C!Navy;
 while (i<=MesBase+ TotShow)and(k< =MesCounter )and
      (j<TotShow+ 1) do begin
  inc(k);
  if (TimeToStr(Tel[k].Bassur)<> '00:00:00;and(TimeToStr
   (Tel[k].Bitsur)='00:00:00;and(j<TotShow) then begin
   canvas. Text0ut(50,j*20+ 50, Tel[k].Aranan);
   canvas. Text0ut(250,j*20+ 50, Tel[k].Arayan);
   canvas. Text0ut(450,j*20+ 50, TimeToStr(Tel[k].Bassur));
   inc(j); inc(i);
  end;
 end,·
 talking: =0;
far i:=1 to MesCounter-1 do begin
  if TimeToStr(Tel[i].BitSur) = '00:00:00' then talking: =talking+ 1;
 end;
 if MesCounter>O then begin
  canvas. Font. Color: =C!Black;
```

```
      canvas. Text0ut(200,  (TotShow+ 1) *20+50, 'Current Dialer .. :
          '+IntToStr(talking));
      canvas. Text0ut(450,  (TotShow+ 1) *20+50, 'Total Dialer .. :
          '+IntToStr(MesCounter-1));
    end;
    if (MesCounter> l)and(random(5)=1)   then begin
     randTel: =random(MesCounter-1)  + 1;
     if TimeToStr(Tel[randTel].BitSur)   = '00:00:00' then begin
       Tel[randTel].BitSur:   =Now;
       Tel[randTel].F ark:= Tel[randTel].BitSur-Tel  [randTel].BasSur;
     end;
    end;
    ReWrite(FTel);
   for i: =1 to MesCounter  do begin
     Seek(Ftel,FileSize(FTel));
     Write(FTel,Tel[i]);
    end;
    Label 1. Caption: ='Time .. : '+time ToStr(now);
  end;

 procedure   TForml.SpeedButtonlClick(Sender:       TObject);
  begin
   form2.Show,·
  end;

 procedure   TForml.SpeedButton2Click(Sender:       TObject);
  begin
    close;
  end;

 procedure   TForml.FormActivate(Sender:      TObject),·
  begin
    Timer 1. Interval: = 1;
 end,·
```

```
end.

unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ComCtrls, StdCtrls, Mask, Printers, Buttons;

type
  TTel = record
    Arayan: String[40];
    Aranan : String[40];
    Tarih : TDate;
    Bassur,
    BitSur,
    Fark,totFark: TTime;
  end;
  TForm2 = class(TForm)
    ListViewl: TListView;
    MaskEditl: TMaskEdit;
    MaskEdit2: TMaskEdit;
    Edit1: TEdit;
    Labell: TLabel;
    Labe/2: TLabel;
    Labe/3: TLabel;
    Labe/4: TLabel;
    SpeedButtonl: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    LabelS: TLabel;
    GroupBoxl: TGroupBox;
```

```
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Labell 0: TLabel;
    Labelll: TLabel,·
    procedure FormKeyPress(Sender: TObject; var Key: Char);
    procedure MaskEdit2Exit(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure SpeedButtonlClick(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
  private
    {Private declarations}
  publiC
    { Public declarations}
  end;

var
  Form2: TForm2;
  FTel,FTelG : File of TTel;
  Tel: TTel;
  Items: TListltem;
  Cols : TListColumn,·
implemehtation

uses Uniti;

{$R *.DFM}

procedure TForm2.FormKeyPress(Sender: TObject; var Key: Char);
begin
  if key=#13 then begin
```

```
    key:=#0;
    PostMessage(Handle,  WM_KEYDOWN,  09, 0),·
  end;
end;

procedure  TForm2.MaskEdit2Exit(Sender:    TObject);
var
  i : Integer;
  saat,dak,san,sal:    Word;
  Fark:  TTime;
begin
  List View1. Columns. Clear;
  ListViewl.Items.   Clear;
  Cols: =List View1. Columns.Add;
  Cols.Caption;  = 'Date',·
  Colsıwtdth: = 100;
  Cols: =List View1. Columns.Add;
  Cols. Caption:='Dialed  No',·
  Cols. Width: = 150;
  Cols: =ListView  1. Golumns.Add;
  Cols. Caption: = 'Start  Time';
  Cols.. Width: =80;
  Cols: =List View1. Columns.Add;
  Cols. Caption: = 'End  Time';
  Cols. Width: =80;
  Cols: =List View1. Columns.Add;
  CoisıOaption: = 'Durati on';
  Cols. Width: =80,·
 Labe/4. Caption: = '0';
for i:=O to FileSize(FTelG)-1   do  begin
  Seek(Fte!G,  i);
  Read(FtelG,  Tel),·
  if (editl.  Text=Tel.Aranan)and(StrToDate(MaskEditl.      Text)< =Tel. Tarih)
    and(StrToDate(MaskEdit2.    Text)>=Tel.  Tarih)  then  begin
```

```
        Items: =List View1.Items.Add;
        items. Caption: =DateToStr(Tel. Tarih);
        items. Sub!tems.Add(Tel.Arayan);
        items. Sub!tems.Add(TimetoStr(Tel.Bassur));
        items. Sub!tems.Add(TimetoStr(T   el. Bitsur));
        items.Sub!tems.Add(TimetoStr(Tel.Bitsur-Tel.Bassur));
        DecodeTime(Tel.Bitsur-Tel.Bassur,Saat,Dak,San,Sal);
        Fark: =Fark+ Tel.Bitsur-Tel.Bassur;
        {Labe/4. Caption: =IntToStr(StrTolnt(Labef   4. Caption) + (Dak*60) +San),')
      end;
    end;
    Labe/4. Caption: =TimeToStr(Fark),·
    {Label4.Caption:  =Labe/4. Caption+' sec'}
  end;

 procedure  TForm2.FormCreate(Sender:    TObject);
 begin
   AssignFile(FTel,  'Tel. dat');
   {$i-}Reset(FTel); {$i+}
   if IOResult<>O  then ReWrite(FTel);

   AssignFile(FTelG,  'Te!G.dat');
   {$i-}Reset(FTelG);{$i+}
   ifIOResult<>O  then ReWrite(FTelG);
 end;

procedure  TForm2.FormActivate(Sender:    TObject);
 begin
   Edit 1. Text: = '';
   MaskEditl.Text:='';
   MaskEdit2.  Text: = '';
   labe/4. Caption: = '0';
   Editl.SetFocus;
   List View1. Items. Clear;
```

```
  Form]. timer l.Interval: =1000;
  Labe/5. Caption: =IntToStr(Forml. UnitPrice) +'TL';
  Labe/8.Caption:=DateToStr(Now);
end;

procedure TForm2.SpeedButton1Click(Sender: TObject);
var
 i : Integer;
ft,fy : Integer;
begin
ft:=300;.fy:=300;
 Printer. Canvas.Font.Pitch: =fpliixed;
 Priruer.Beginôoe,
 Printer. Canvas.Font.Size: = 18;
 Printer. Canvas.Font.Size: =10;
 Printer.Canvas, Text0ut(/3c+ 1500,.fy,DateToStr(Now));
 Printer.Canvas.Text0ut(/3c,fy+70 , 'Phone Number: '+Editl.Text);
 Printer. Canvas. Text0ut(/3c,fy+ 150, 'Date : '+MaskEditl. Text+
  ' - '+MaskEdit2.Text);
Printer. Canvas.Font.Style: =Printer. Canvas.Font.Style+ [febold];
if ListViewl. Visible then begin
 Printer. Canvas.Font. Style: =Printer. Canvas.Font.Style+ [febold];
 printer. Canvas. Text0ut(/3c+600,fy-J 00, 'DETAILED BILL ');
 Printer. Canvas. Font. Style: =Printer. Canvas. Font. Style-ffsbold];

 frinter. Canvas. TextOut(fx ,fy+ 320, 'Date');
 ftititer.Çanvas. TextOut(fx+ 300, fy+ 320, 'Dialed No');
 Printer. Canvas. Text0ut(fx+800, fy+ 320, 'Start time');
 Printer. Canvas. TextOut(fx+ 1100,.fy+ 320, 'End time');
 Printer. Canvas. TextOut(fx+ 1400,.fy+ 320, 'Duration');
 Printer. Canvas.Font.Style: =Printer. Canvas.Font.Style-[febold];
 Printer. Canvas. Font. Size: =8;
for i:=O to ListViewl.Items.Count-1 do begin
  Printer.Canvas.TextOut(fx ,fy+400+(i*70),ListViewl.items[i].Caption);
```

```
        Printer. Canvas. TextOut(fx+ 300
          ,fy+400+(i*70),ListViewl.items[i].Subltems[OJ);
        Printer. Canvas. Text0ut(fx+800
         ,fy+400+(i*70),ListViewl.items[i].Subltems[l});
        Printer. Canvas. TextOut(fx+ 1100,fy+400+(i*70),
          List View 1. items [i}. Subltems [2J);
        Printer. Canvas. TextOut(fx+ l 400,fy+400+(i*70),
          List View 1. items [i]. Subltems [3J),·
      end;
    end
    else begin
     Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[febold];
     printer. Canvas. Text0ut(fx+600,fy-J  00, 'NORMAL BILL '),·
     Printer. Canvas. Fant.Style: =Printer. Canvas. Fant.Style-ffsbold];

     Printer. Canvas. TextOut(fx,fy+ 320, 'Total Duration : '+Label9. Caption );
     Printer.Canvas. Text0ut(fx,fy+420, 'Unit Price : '+Label5.Caption );
     Printer.Canvas.Text0ut(fx,fy+520, 'Total Price : '+Labe!JO.Caption );
    end,·
   Printer.EndDoc;
  end,·

 procedure TForm2.SpeedButton2Click(Sender: TObject);
  begin
   Form]. Uniti'rice=Str'Iolnttlnpıaboxt", 'Please Enter the new Unit Price'
   ,IntToStr(Forml. UnitPrice))),·
   Labe/5. Caption: =IntToStr(Forml. UnitPrice) +' TL';
 end,·

procedure TForm2.SpeedButton3Click(Sender: TObject);
 var
  saat,dak,san,sal: Word;
  Zaman : Integer;
 begin
```

```
  ListViewl.  Visible:=  not ListViewl.  Visible;
  GroupBoxl.  Visible:=not  GroupBoxl.  Visible;
  Decode  Time (StrTo Time (La be14. Caption), saat, dak, san, sal);
  Zaman:=(saat*60)+(dak);
  Label9. Caption: =La be14. Caption;
  LabelJO.Caption:=IntToStr((Forml.      UnitPrice*Zaman)+
   Form]. UnitPrice)+' TL';
  ifGroupBoxl. Visible then begin
   SpeedButton3. Caption: = 'Detailed Bili';
 end
 else begin
  SpeedButton3. Caption: = 'Normal Bili';
enci;
end;

end.
```

CONCLUSION

At the end we can divide the program into three part. Each part can be called as module.

The fırst module is a simulation module. This module randomly generate two different phone numbers. üne is called dialed, other is called dialing number. The dialing number dialing the dialed number. These two number never can be same as with each other.

The second module is file organization module. This module make data transfer between two binary file.

The last one is printer module that is the most important module. And also most efficient module in this program. It print two different type of bill, which type the dialer want.

Finally, the program allow the user to change the unit price. The calculatıon of price is that multiplying minutes by unit price gives us the total price. In real world there is different there are different price calculation. E.g. In TRNC from normal phone to GSM phone, the unit price is equal to 17.000 TL. But 1 minute equal to three the counter

APPENDIX

'f.be 'fwo Example Print Out

DETAILED BILL

03.06.2000

Phone Number : 227 10 10

Date : 05.05.2000 - 06.06.2000

Date	Dialed No	Start time	End time	Duration
30.05.2000	0533 471 22 46	22:06:03	22:06:41	00:00: 37
30.05.2000	0532 '880 40 99	22:07:04	22:07:05	00:00:00
30.05.2000.	0533 448'. 55 51	22:07:11	22:07:22	00:00:11
30.05.2000	0532 978 11 72	22:07:11	22:07:22	00:00:10
30.05.2000	0532 361 3S 59	22:07:12	22:07:22	00:00:10
30.05.2000	0090 462 103 72 22	22:07:12	22:07:16	00:00:04
30.05.2000	0532 397 20 67	22:07:19	22:07:22	00:00:02
30.05.2000	05.32 807 52 69	22:07:20	22:07:22	00:00:02
30.05.2000	0542 735 68 46	22:07:20	22:07:22	00:00:02
30.05.2000	815 79 75	22:07:21	22:07:22	00:00:01
30.05.2000	0542 953 20 23	22:07:22	22:07:22	00:00:00
30.05.2000	05.33 484 83 84	22 :07:47	22:07:54	00:00:06
30.05.2000	0532 398 13 65	22:07:49	22:.07:50	00:00:00
30.05.2000	815 *22* 15	22:07:50	22:07:54	00:00:04
30.05.2000	0090 202 138 81 23	22:07:51	22:07:51	00:00:00
30.05.2000	0090 359 557 43 24	22:07:51	22:07:54	00:00:03
30.05.2000	227 62 37	22:07:53	22:.07:54	00:00:01
30.05.2000	0542 985 11 92	22:08:14	22:08:15	00:00:01
30.05.2000	227 14 99	22:08:15	22:.08:15	00:00:00

NORMAL BILL

03.06.2000

Phone Number : 227 10 10

Date : 05.05.2000 - 06.06.2000

Total Duration : 00:01:42

Unit Price : 20000 TL

Total Price : 40000 TL