# NEAR EAST UNIVERSITY

## Faculty of Engineering

## Department of Computer Engineering

# HOSPITAL AUTOMATION PROGRAM

**Graduation Project**
**COM - 400**

**Prepared By: Salih Soysal (950124)**

**Supervisor : Okan Donangil**

**Lefkoşa - 2000**

# CONTENTS

## ACKNOWLEDGEMENT

First of all I would like to thanks my project supervisior Mr. Okan Donangil. During preparing my project he helped and supported me. And also I want to thanks my friends Erkan Hakverdi, Ahmet Nalband, Mahmut Boke and my cousin A. Bahadir Soysal because of they helps moralise. They give me their ideas and correct my errors and mismathc during program.

Espacially I would like to thanks my family. They support me during my whole education from primary school up to university. They support me as substantial and moral.

And finally I want to thanks my friends Osman Tekin and Onur Taha Cananer because of they helps and supports.

# ABSTRACT

The program consists of several databases for several jobs. The program divided into three parts or modules. Record module, Process module, and Database files.

The first module makes new records for patients. It saves information requested for patient in a database file. Some of these information are name, surname, id-no, if there exist social security no and etc. these module also allow user to change these information, update or delete.

The second module does many processes. It is the main part of program and the brain of the program. These processes are done for those the inner patient which can be laboratory results or others, for newer patient which can be that the patient rooms the patient illness and etc.

The database module is stores the data and when required take back them. This module is also very important. Because all of the information are stored in this files.

# INTRODUCTION

This project is a hospital automation program. The program can meet the need of a hospital. There are very complicated processes in hospitals. This program can meet all of these needs. This is the first purpose of this program.

The second is that to make easy the process of the hospital, and to take more efficiency in hospitals. How the program does these? It has a database file that stores all the information. The doctors can use this database to reach easily a patient information and can learn what is the case of patient and what he can do. Because the database including all the necessary information about patients. And also including X-Rays the laboratory results.

The program can give best statistical information. The user can take more efficient statistic like which illness are found more in a place, how many patient were get better using the database and so on.

And the program takes very useful archive for the patient. This is an important think for the real world. Because some times some old information can be necessary at the different times. This program can also meet these requests.

# DENMA HOSPITAL AUTOMATION

## A) DELPHI PROGRAMMING

In Delphi programming we have to declare some specific values in special places to make tables. This kind of things are same as in Access programming and you can not do nothing to its database but in Delphi programming language we have to declare what kind of database that we will use for that reason before we use tables or indexes or different places we have to first open a database and specify values that we will use in our program. After that, we can start to open our tables, forms, indexes and we may open the fields if we need. Delphi 4 expands the Object Pascal language to include dynamic arrays, method overloading, default parameters, and more.

Each month Delphi Informant is packed with technical articles that are of the utmost interest to the Delphi developer including:

Object-Oriented Programming.

Database Fundamentals.

Client/Server Development.

Programming in ObjectPascal.

Using the Borland Database Engine.

Information on New Delphi Add-In Products.

Product Reviews.

Book Reviews.

News from the Delphi Community.

Delphi User Group Information.

Delphi Informantion is the premiere technical publication for developers creating solutions with Borland Delphi. This award-winning monthly publication features leading-edge technical information on developing solutions with Delphi. Readers will learn advanced techniques for using Object Pascal, ActiveX controls, and other new technologies with Delphi and with third party software publishers. All of this information comes from industry experts including members of the Borland product teams and top third party developers and authors.

**Procedural Types in Expressions in Delphi**

Usually, using a procedural variable in a statement or an expression calls the procedure or function stored in the variable. There is one exception: When the compiler sees a procedural variable on the left side of an assignment statement, it knows that the right side has to represent a procedural value. For example, consider the following program:

```
type
 IntFunc = function: Integer;
var
 F: IntFunc;
 N: Integer;
function ReadInt: Integer; far;
var
 I: Integer;
begin
 Read(I);
 ReadInt := I;
end;
begin
 F := ReadInt; { Assign procedural value }
 N := ReadInt; { Assign function result }
```

The first statement in the main program assigns the procedural value (address of) ReadInt to the procedural variable F, where the second statement calls ReadInt and assigns the returned value to N. The distinction between getting the procedural value or calling the function is made by the type of the variable being assigned (F or N).

Unfortunately, there are situations where the compiler can't determine the desired action from the context. For example, in the following statement there is no obvious way the compiler can know if it should compare the procedural value in F to the procedural value of ReadInt to determine if F currently points to ReadInt, or if it should call F and ReadInt and then compare the returned values.

```
if F = ReadInt then
Edit1.Text := 'Equal';
```

Object Pascal syntax, however, specifies that the occurrence of a function identifier in an expression denotes a call to that function, so the effect of the preceding statement is to call F and ReadInt, and then compare the returned values. To compare the procedural value in F to the procedural value of ReadInt, the following construct must be used.

When applied to a procedural variable or a procedure or function identifier, the address (@) operator prevents the compiler from calling the procedure, and at the same time converts the argument into a pointer. @ F converts F into an untyped pointer variable that contains an address, and @ReadInt returns the address of ReadInt; the two pointer values can then be compared to determine if F currently refers to ReadInt.

The @ operator is often used when assigning an untyped pointer value to a procedural variable. For example, the GetProcAddress function defined by Windows (in the WinProcs unit) returns the address of an exported function in a DLL as an untyped pointer value. Using the @ operator, the result of a call to GetProcAddress can be assigned to a procedural variable.

```
type
 TStrComp = function(Str1, Str2: Pchar): Integer;
var
 StrComp: TStrComp;
 ...
begin
 ...
 @StrComp := GetProcAddress(KernelHandle, 'lstrcmpi');
 ...
end.
```

# B) DATABASES

## Introduction to Databases

A database is a collection of data. An address book or a phone directory are well known databases. Unlike one of these written or printed databases, with a computerized database you are not stuck with a single format. Not only can you search for a phone number, you can also search for an address, a first name, or whatever other data, or combination of data, is in the collection. Then you add the fact that a computer database is not restricted to certain data like name, address, and phone number. You can collect any data you specify, such as part numbers, order numbers, dates, or product prices.

With a computer database you can report exactly the information you want, in the format you want. You can report sales by city. You can report part numbers by customer, by route, or by category. You can report orders by salesman. You can report sales by dollar value. You can report whatever you want. For example: What if there were a route code assigned to each customer? Then deliveries could be sorted and made by route to save time. Such data can quickly make some businesses much more profitable.

Databases often contain sensitive information. Different databases provide security schemes for protecting that information. Some databases, such as Paradox and dBASE, only provide security at the table or field level. When users try to access protected tables, they are required to provide a password. Once users have been authenticated, they can see only those fields (columns) for which they have permission.

Most Structure Querty Language (SQL ) servers require a password and user name to use the database server at all. Once the user has logged in to the database, that username and password determine which tables can be used.

When designing database applications, you must consider what type of authentication is required by your database server. If you do not want your users to have to provide a password, you must either use a database that does not require one or you must provide the password and username to the server programmatically. When providing the password programmatically, care must be taken that security can't be breached by reading the password from the application.

If your application requires multiple passwords because you must log in to several protected systems or databases, you can have your users provide a single master

password which is used to access a table of passwords required by the protected systems. The application then supplies passwords program, without requiring the user to provide multiple passwords.

If you are requiring your user to supply a password, you must consider when the password is required. If you are using a local database but intend to scale up to a larger SQL server later, you may want to prompt for the password before you access the table, even though it is not required until then.

**Entering Structured Query Language (SQL) Statements**

This page is a tutorial of the SQL and is pioneering effort on the World Wide Web, as this is the first comprehensive SQL tutorial available on the Internet. SQL allows users to access data in relational database management systems, such as Oracle, Sybase, Informix, Microsoft SQL Server, Access, and others, by allowing users to describe the data the user wishes to see. SQL also allows users to define the data in a database, and manipulate that data. This page will describe how to use SQL, and give examples. The SQL used in this document is Abbreviation for American National Standadrs Institute (ANSI), or standard SQL, and no SQL features of specific database management systems will be discussed until the "Nonstandard SQL" section.

In some versions of Delphi, the SQL Explorer permits you to make SQL queries against an SQL database on a remote server. In other versions, the Database Explorer permits you to make queries against Paradox and dBASE tables.

**Data Stored**

Information is stored in a datasheet. A datasheet looks like a spreadsheet with it's many columns and rows. The rows are called records (for example, each customer is a different record). The columns are called fields. Each field contains some important attribute of each record (for example, a certain field would contain a name, another a telephone number, another a part number, or another maybe an order date.

**Relational Database**

The simplest form of computer database is the flat file database. Microsoft Excel can be used, and is used by many people, as a database. But this flat file database has serious limitations. One is its slow speed. Another is that a flat file duplicates a lot of

the data that could be entered just once. Duplicate data has to be input again and again. It also has to be sorted by the computer program. Sorting is the major cause of your mouse pointer turning into an hourglass.

On the other hand is the state of the art type of computer database, the "relational" database. Although the user doesn't know how it's storing information internally, it is stored in a very efficient manner and operates very quickly when properly designed. Microsoft Access is a relational database.

**Connecting to Database**

When a Delphi application connects to a database, that connection is encapsulated by a Database component. A database component encapsulates the connection to a single database in the context of a BDE session in an application. These topics describe database components and how to use them to manipulate database connections and control transactions.

**Managing Database Sessions**

Both standalone, full client database applications and database application servers communicate with databases through the BDE. An application's database connections, drivers, cursors, queries, and so on are maintained within the context of one or more BDE sessions. Sessions isolate a set of database access operations, such as database connections, without the need to start another instance of the application. In a Delphi application, you can manage BDE sessions using TSession and TSessionList components. Each TSession component in an application encapsulates a single BDE session. All sessions within an application are managed by a single TSessionList component.

All database applications also are automatically provided with a session list component, named Sessions, that enables the application to manage all of its session components. These topics describe the session and session list components and explains how to use them to control BDE sessions in your full client database applications and database application servers. Applications can declare, create, and manipulate additional session components as needed.

# C) HOSPITAL AUTOMATION

## C.1) A Sample Analysis:

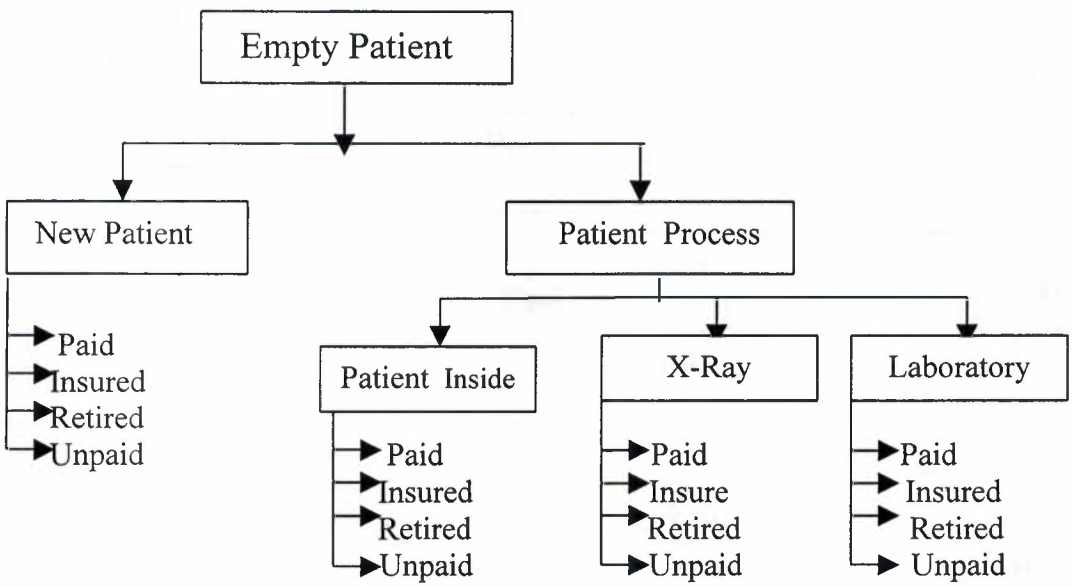
Form an hospital automation the following process should done.

- Patient Record Process
- Patient Process

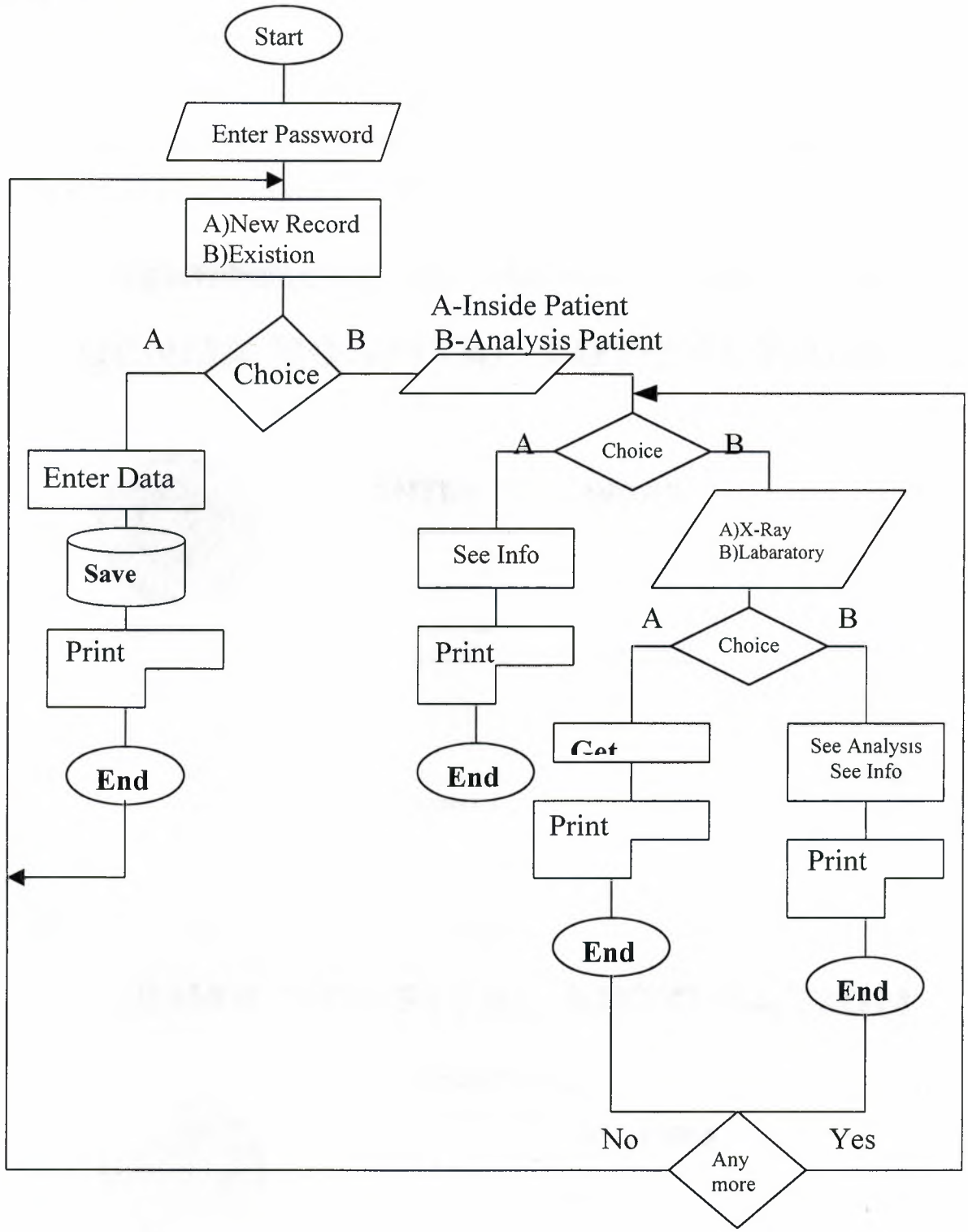
## C.2) System Definition:

The user check the records, as name, id no, end etc. If the patient these exist user will check for second process.

## C.3) Program Flowchart:



**C.4) System Flowchart:**



**C.5) Description Process:**

**C.5.1) Password Screen:**

First of all the start by password screen. The user must enter password(figure1.1). New menu will appear on screen. If the user don't enter correct password you couldn't enter the program(figure1.2).

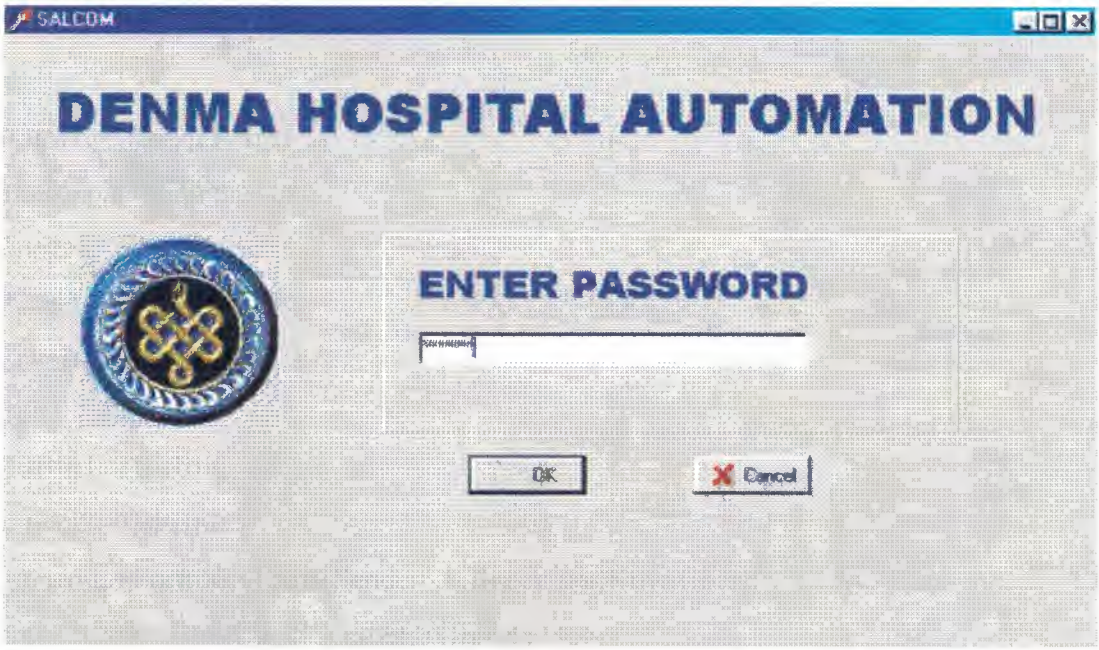



figure1.1

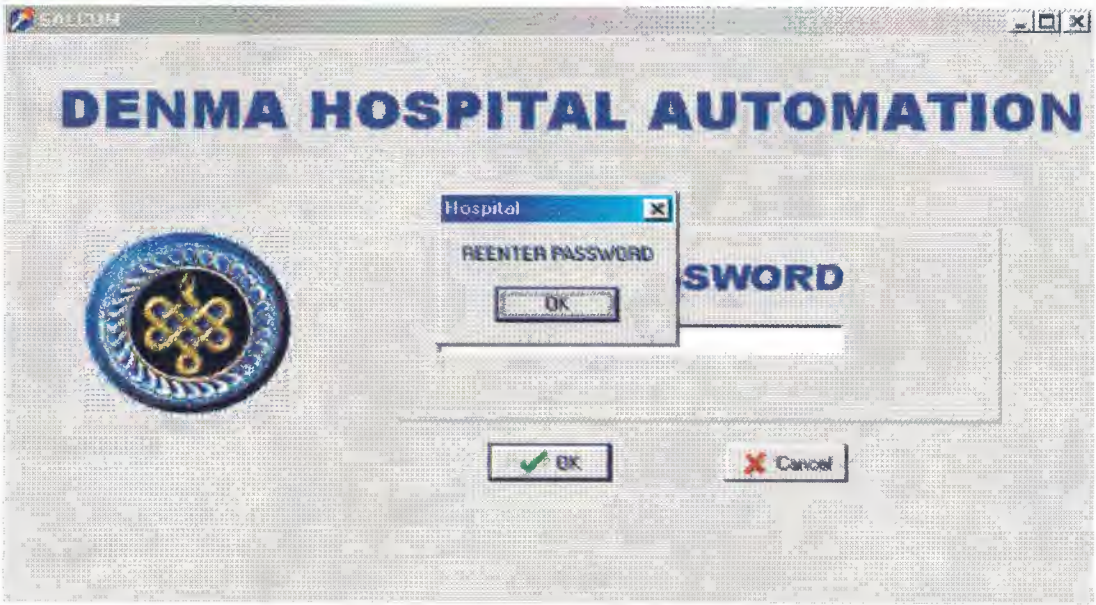



figure1.2

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, unit2, ExtCtrls, jpeg;

type
  TForm1 = class(TForm)
    StaticText1: TStaticText;
    Image1: TImage;
    Panel1: TPanel;
    Label1: TLabel;
    Edit1: TEdit;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    procedure FormKeyPress(Sender: TObject; var Key: Char);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}
```

```
procedure TForm1.FormKeyPress(Sender: TObject; var Key: Char);
begin
  if Key=#27 then Close;
  if Key=#13 then begin
    Key:=#0;
    PostMessage(Handle,WM_KEYDOWN,09,0);
  end;
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  if edit1.text='1' then begin
    form1.Hide;
    form2.Show;
  end
  else showmessage('REENTER PASSWORD');
end;

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

**C.5.2) Menu Screen:**

DENMA HOSPITAL AUTOMATION consist of two part. One of them Patient Record and the other is Patient Process.(figure2)

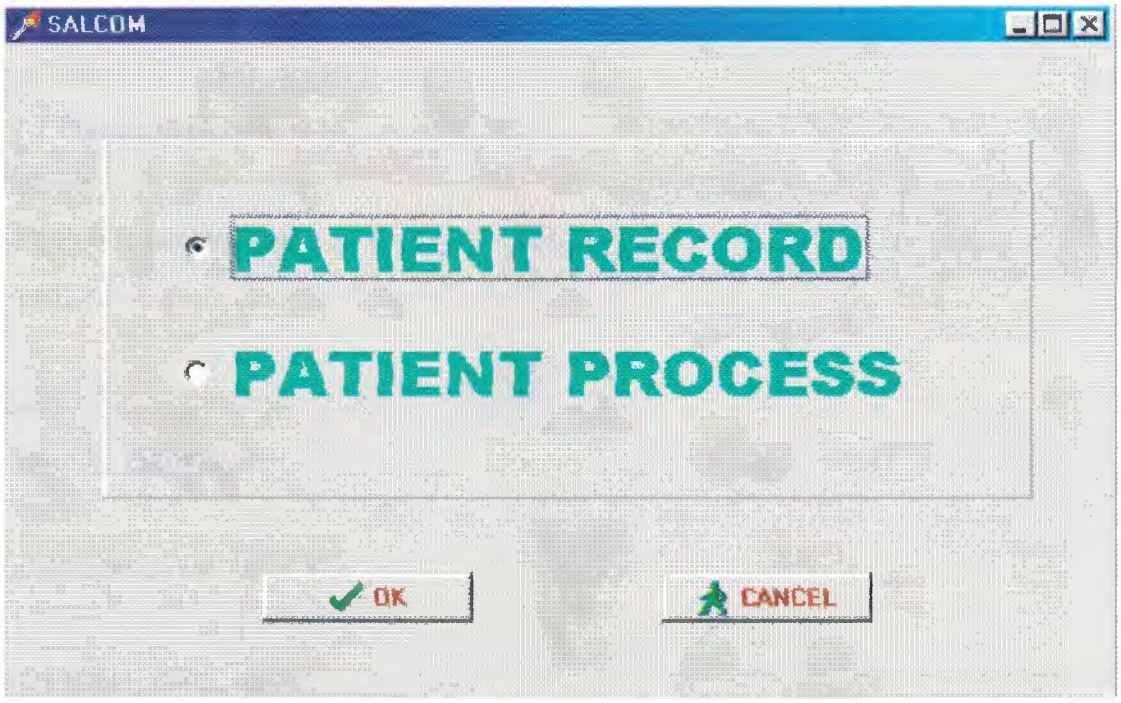



Figure2

```
unit Unit2;

interface

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

type
  TForm2 = class(TForm)
    Panel1: TPanel;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    BitBtn1: TBitBtn;
```

```
    BitBtn2: TBitBtn;
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormKeyPress(Sender: TObject; var Key: Char);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation

uses Unit1, unit3, unit4;

{$R *.DFM}

procedure TForm2.BitBtn2Click(Sender: TObject);
begin
  close;
end;

procedure TForm2.BitBtn1Click(Sender: TObject);
begin
    if radiobutton1.checked=true then
    form3.visible:=true
    else if radiobutton2.checked=true then
    form4.visible:=true
end;
```

```
procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  form1.Close;
end;

procedure TForm2.FormKeyPress(Sender: TObject; var Key: Char);
begin
  if Key=#13 then begin
    if RadioButton1.Checked then form3.Show;
    if RadioButton2.Checked then form4.Show;
  end;
  if Key=#27 then Close;
end;

procedure TForm2.FormCreate(Sender: TObject);
begin

end;
end.
```

**C.5.3) Patient Record Screen:**

If the user select the Patient Record menu user see only one menu. But user can select four part. They are insured, retired, payment, unpaid. If the patient has recorded before program is not need to store again. It will use the previous information about the patient. If the user want to printout click print button.

- **Unpaid:**

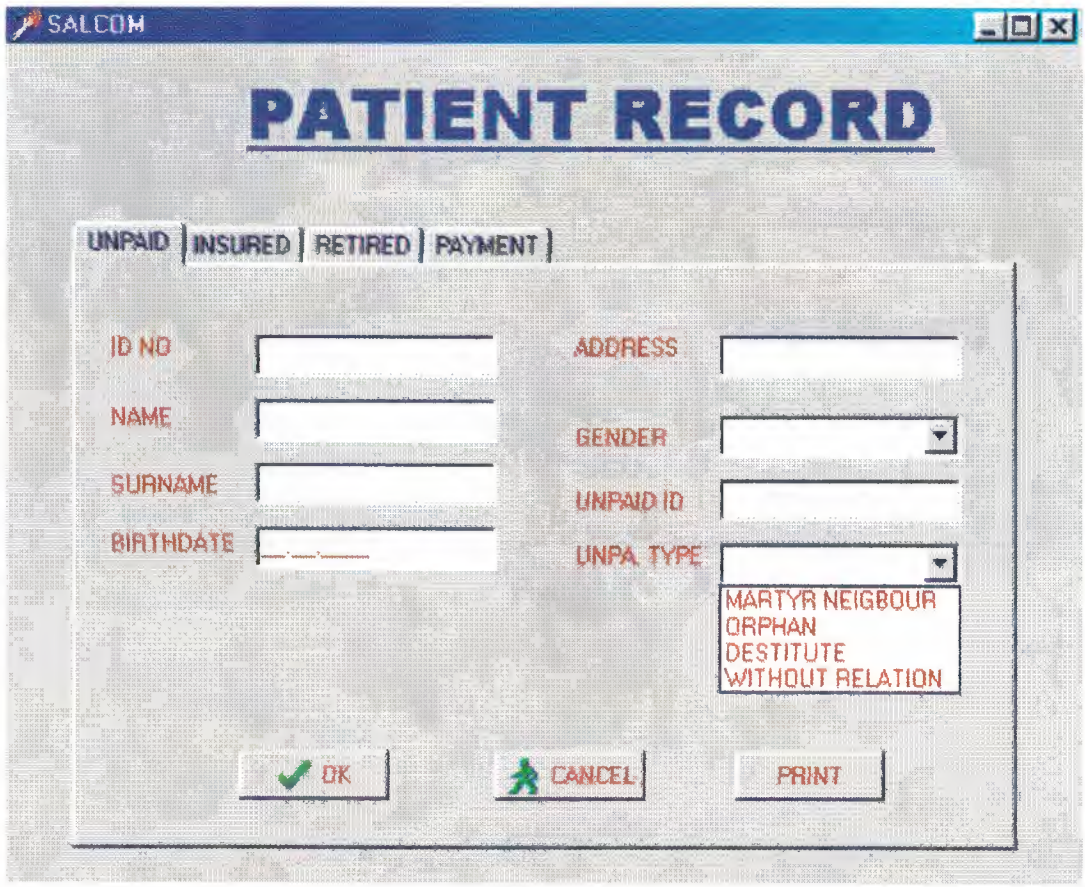



Figure3.1

- **Insured:**

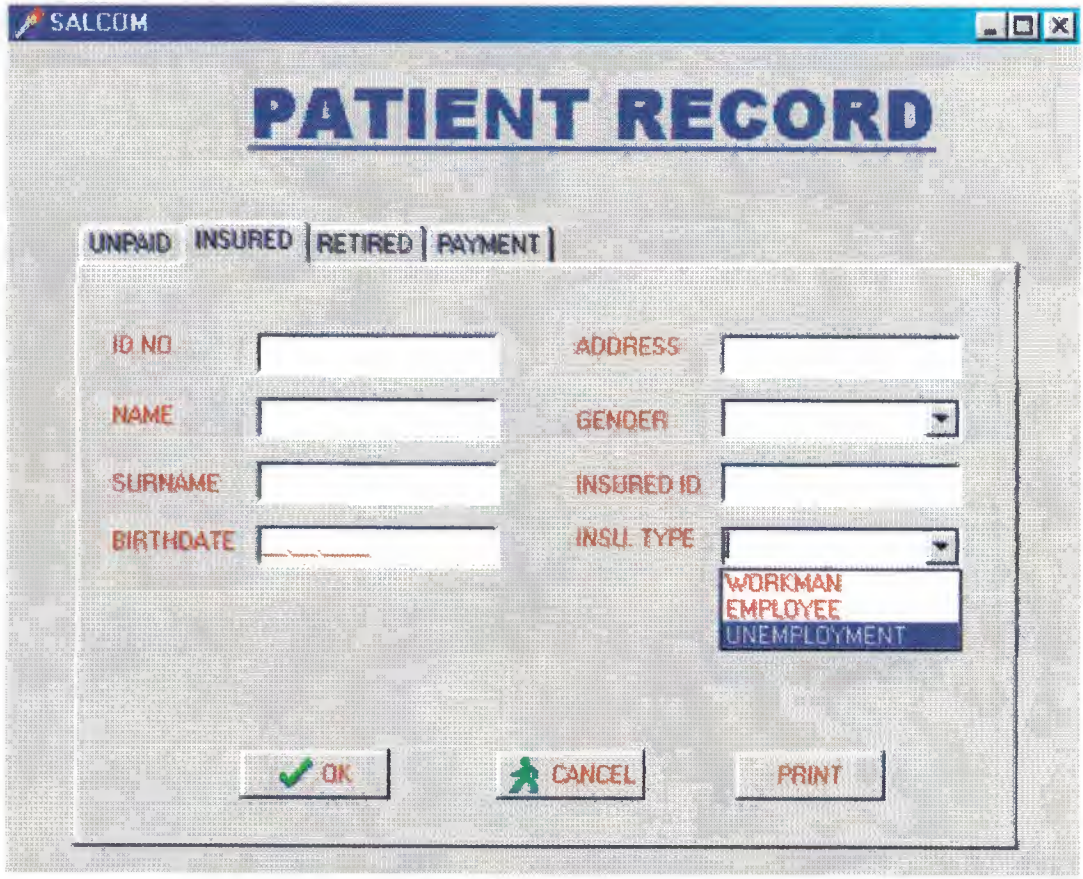



Figure3.2

- **Retired:**

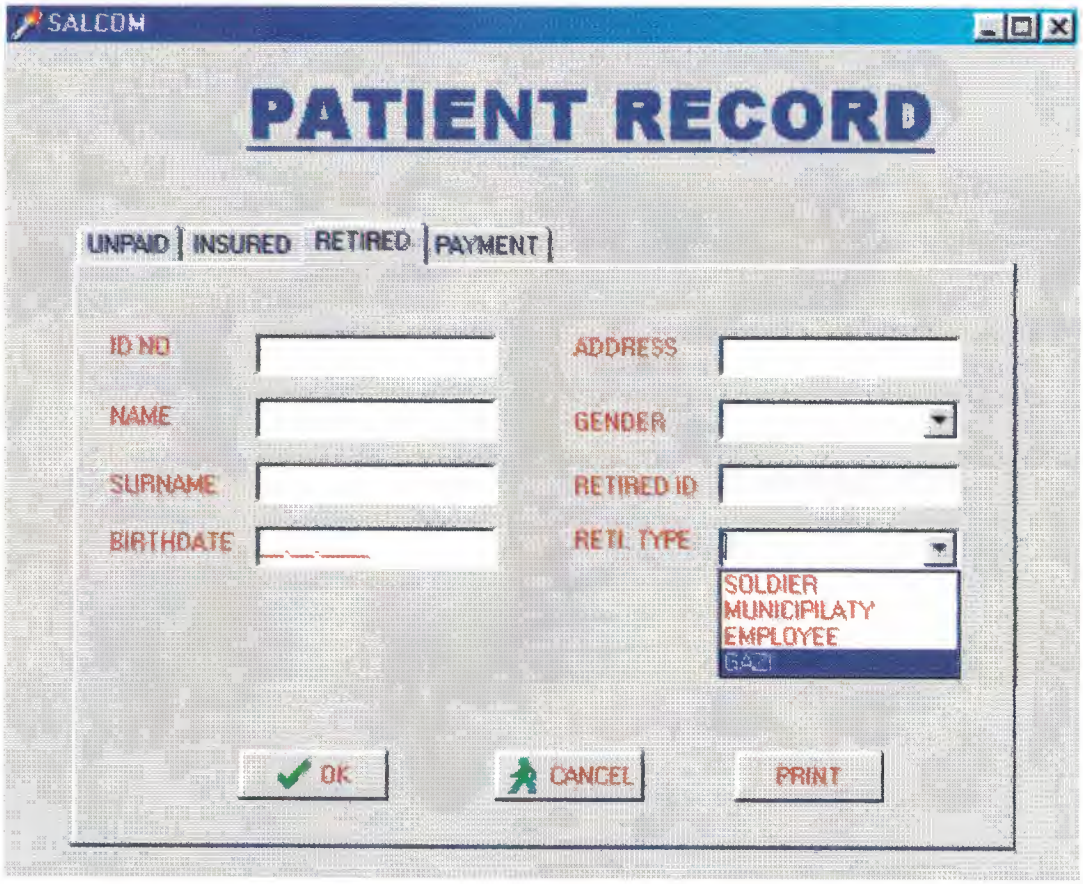



*Figure3.3*

- **Payment:**

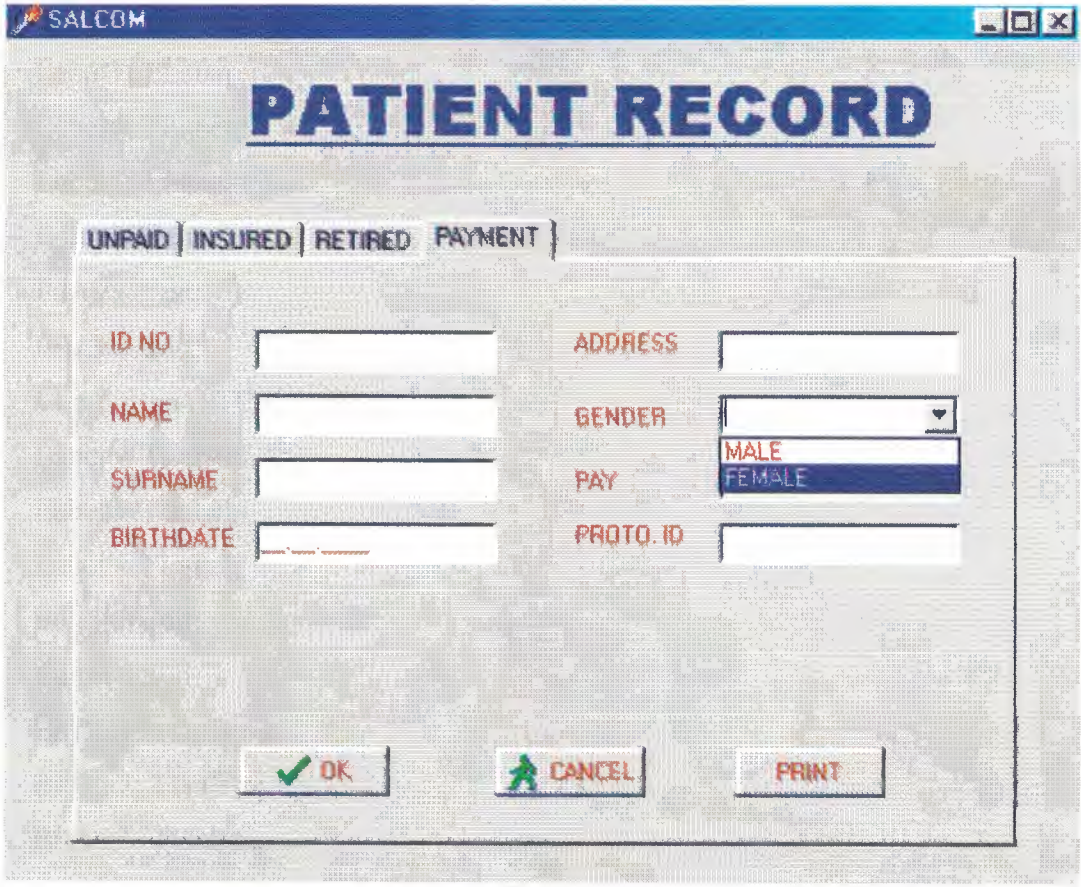



Figure3.4

```
unit Unit3;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ComCtrls, Mask, Db, DBTables,Printers;

type
  TForm3 = class(TForm)
    INSURED: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    TabSheet3: TTabSheet;
    TabSheet4: TTabSheet;
```

```
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Edit1: TEdit;
Edit2: TEdit;
Edit3: TEdit;
ComboBox1: TComboBox;
Label6: TLabel;
Label8: TLabel;
Label9: TLabel;
Edit5: TEdit;
Edit6: TEdit;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
BitBtn3: TBitBtn;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Edit8: TEdit;
Edit9: TEdit;
Edit10: TEdit;
ComboBox3: TComboBox;
Label15: TLabel;
Label17: TLabel;
Edit12: TEdit;
Edit13: TEdit;
BitBtn4: TBitBtn;
BitBtn5: TBitBtn;
BitBtn6: TBitBtn;
```

Label19: TLabel;

Label20: TLabel;

Label21: TLabel;

Label22: TLabel;

Label23: TLabel;

Edit15: TEdit;

Edit16: TEdit;

Edit17: TEdit;

ComboBox5: TComboBox;

Label24: TLabel;

Label26: TLabel;

Label27: TLabel;

Edit19: TEdit;

Edit20: TEdit;

Edit21: TEdit;

BitBtn7: TBitBtn;

BitBtn8: TBitBtn;

BitBtn9: TBitBtn;

Label28: TLabel;

Label29: TLabel;

Label30: TLabel;

Label31: TLabel;

Label32: TLabel;

Edit22: TEdit;

Edit23: TEdit;

Edit24: TEdit;

ComboBox7: TComboBox;

Label33: TLabel;

Label35: TLabel;

Label36: TLabel;

Edit26: TEdit;

Edit27: TEdit;

BitBtn10: TBitBtn;

```
  BitBtn11: TBitBtn;
  BitBtn12: TBitBtn;
  Label18: TLabel;
  ComboBox9: TComboBox;
  ComboBox10: TComboBox;
  ComboBox11: TComboBox;
  StaticText1: TStaticText;
  MaskEdit1: TMaskEdit;
  MaskEdit2: TMaskEdit;
  MaskEdit3: TMaskEdit;
  MaskEdit4: TMaskEdit;
  procedure BitBtn2Click(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn4Click(Sender: TObject);
  procedure BitBtn7Click(Sender: TObject);
  procedure BitBtn10Click(Sender: TObject);
  procedure FormKeyPress(Sender: TObject; var Key: Char);
  procedure FormActivate(Sender: TObject);
  procedure BitBtn9Click(Sender: TObject);
  procedure Edit15Exit(Sender: TObject);
  procedure Edit8Exit(Sender: TObject);
  procedure Edit1Exit(Sender: TObject);
  procedure Edit22Exit(Sender: TObject);
  procedure INSUREDChange(Sender: TObject);
  procedure BitBtn6Click(Sender: TObject);
  procedure BitBtn3Click(Sender: TObject);
  procedure BitBtn12Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

```
var
  Form3: TForm3;
  RStatus : String[20];
implementation

uses Unit1,unit8;

{$R *.DFM}

procedure AllInVisible;
begin
  with form3 do begin
    Edit1.Text:='';
    Edit2.Text:='';
    Edit3.Text:='';
    MaskEdit4.Text:='';
    ComboBox1.Text:='';
    Edit5.Text:='';

    Edit6.Text:='';
    ComboBox10.Text:='';
    Edit8.Text:='';
    Edit9.Text:='';
    Edit10.Text:='';
    MaskEdit3.Text:='';
    ComboBox3.Text:='';
    Edit12.Text:='';

    Edit13.Text:='';
    ComboBox9.Text:='';
    Edit15.Text:='';
    Edit16.Text:='';
    Edit17.Text:='';
```

```
    MaskEdit2.Text:='';
    ComboBox5.Text:='';
    Edit19.Text:='';

    Edit20.Text:='';
    Edit21.Text:='';
    Edit22.Text:='';
    Edit23.Text:='';
    Edit24.Text:='';
    MaskEdit1.Text:='';
    ComboBox7.Text:='';
    Edit26.Text:='';

    Edit27.Text:='';
    ComboBox11.Text:='';
  end;
end;

procedure TForm3.BitBtn2Click(Sender: TObject);
begin
   Close;
end;

procedure TForm3.BitBtn1Click(Sender: TObject);
begin
  with DataModule8 do begin
    if RStatus='Normal' then begin
      Table1.Append;
      Table1ID_NO.Text:=Edit1.Text;
      Table1NAME.Text:=Edit2.Text;
      Table1SURNAME.Text:=Edit3.Text;
      Table1BIRTHDATE.Text:=MaskEdit4.Text;
      Table1GENDER.Text:=ComboBox1.Text;
```

```
      Table1ADDRESS.Text:=Edit5.Text;
      Table1INSURED_ID.Text:=Edit6.Text;
      Table1INSURED_TYPE.Text:=ComboBox10.Text;
      Table1TYPE.Text:='Insured';
      Table1.Post;
    end;
  end;
  Edit1.SetFocus;
  AllInVisible;
end;

procedure TForm3.BitBtn4Click(Sender: TObject);
begin
  with DataModule8 do begin
    if RStatus='Normal' then begin
      Table1.Append;
      Table1ID_NO.Text:=Edit8.Text;
      Table1NAME.Text:=Edit9.Text;
      Table1SURNAME.Text:=Edit10.Text;
      Table1BIRTHDATE.Text:=MaskEdit3.Text;
      Table1GENDER.Text:=ComboBox3.Text;
      Table1ADDRESS.Text:=Edit12.Text;
      Table1RETIRED_ID.Text:=Edit13.Text;
      Table1RETIRED_TYPE.Text:=ComboBox9.Text;
      Table1TYPE.Text:='Retired';
      Table1.Post;
    end;
  end;
  Edit8.SetFocus;
  AllInVisible;
end;

procedure TForm3.BitBtn7Click(Sender: TObject);
```

```
begin
  with DataModule8 do begin
    if RStatus='Normal' then begin
      Table1.Append;
      Table1ID_NO.Text:=Edit15.Text;
      Table1NAME.Text:=Edit16.Text;
      Table1SURNAME.Text:=Edit17.Text;
      Table1BIRTHDATE.Text:=MaskEdit2.Text;
      Table1GENDER.Text:=ComboBox5.Text;
      Table1ADDRESS.Text:=Edit19.Text;
      Table1PAY.Text:=Edit20.Text;
      Table1PROTOCOL_ID.Text:=Edit21.Text;
      Table1TYPE.Text:='Payment';
      Table1.Post;
    end;
  end;
  Edit15.SetFocus;
  AllInVisible;
end;

procedure TForm3.BitBtn10Click(Sender: TObject);
begin
  with DataModule8 do begin
    if RStatus='Normal' then begin
      Table1.Append;
      Table1ID_NO.Text:=Edit22.Text;
      Table1NAME.Text:=Edit23.Text;
      Table1SURNAME.Text:=Edit24.Text;
      Table1BIRTHDATE.Text:=MaskEdit1.Text;
      Table1GENDER.Text:=ComboBox7.Text;
      Table1ADDRESS.Text:=Edit26.Text;
      Table1UNPAID_ID.Text:=Edit27.Text;
      Table1UNPAID_TYPE.Text:=ComboBox11.Text;
```

```
    Table1TYPE.Text:='Unpaid';
    Table1.Post;
   end;
  end;
  Edit22.SetFocus;
  AllInVisible;
end;

procedure TForm3.FormKeyPress(Sender: TObject; var Key: Char);
begin
  if Key=#13 then begin
    Key:=#0;
    PostMessage(Handle,WM_KEYDOWN,09,0);
  end;
  if Key=#27 then Close;
end;

procedure TForm3.FormActivate(Sender: TObject);
begin
  TabSheet4.PageIndex:=0;
end;

procedure TForm3.BitBtn9Click(Sender: TObject);
var
  xf,yf : Integer;
begin
  xf:=400; yf:=200;
  Printer.BeginDoc;
  Printer.Canvas.Font.Pitch:=fpFixed;
  Printer.Canvas.Font.Size:=16;
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+450,yf-50 ,'Payment Patient');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];
```

```
  Printer.Canvas.Font.Size:=12;
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf    ,yf+100,'Id No     : ');
  Printer.Canvas.TextOut(xf    ,yf+170,'Name      : ');
  Printer.Canvas.TextOut(xf    ,yf+240,'Surname   : ');
  Printer.Canvas.TextOut(xf    ,yf+310,'Birthdate : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+350,yf+100,Edit15.Text);
  Printer.Canvas.TextOut(xf+350,yf+170,Edit16.Text);
  Printer.Canvas.TextOut(xf+350,yf+240,Edit17.Text);
  Printer.Canvas.TextOut(xf+350,yf+310,MaskEdit2.Text);

  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+900,yf+100,'Address   : ');
  Printer.Canvas.TextOut(xf+900,yf+170,'Gender    : ');
  Printer.Canvas.TextOut(xf+900,yf+240,'Pay       : ');
  Printer.Canvas.TextOut(xf+900,yf+310,'Proto Id  : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+1250,yf+100,Edit19.Text);
  Printer.Canvas.TextOut(xf+1250,yf+170,ComboBox5.Text);
  Printer.Canvas.TextOut(xf+1250,yf+240,Edit20.Text);
  Printer.Canvas.TextOut(xf+1250,yf+310,Edit21.Text);
  Printer.EndDoc;
end;

procedure TForm3.Edit15Exit(Sender: TObject);
var
  founded : Boolean;
begin
  with DataModule8 do begin
    Table1.First; founded:=False;
```

```
  while (not Table1.Eof)and(not founded) do begin
   if CompareText(Edit15.Text,Table1ID_NO.Text)=0 then founded:=True;
   if not founded then Table1.Next;
  end;
  if founded then begin
   if Table1TYPE.Text='Payment' then begin
    RStatus:='Update';
    Edit15.Text:=Table1ID_NO.Text;
    Edit16.Text:=Table1NAME.Text;
    Edit17.Text:=Table1SURNAME.Text;
    MaskEdit2.Text:=Table1BIRTHDATE.Text;
    Edit19.Text:=Table1ADDRESS.Text;
    ComboBox5.Text:=Table1GENDER.Text;
    Edit20.Text:=Table1PAY.Text;
    Edit21.Text:=Table1PROTOCOL_ID.Text;
   end
   else begin
    ShowMessage('You can`t enter this Id No');
    AllInVisible;
    Edit15.SetFocus;
   end;
  end
  else begin
   RStatus:='Normal';
  end;
 end;
end;

procedure TForm3.Edit8Exit(Sender: TObject);
var
 founded : Boolean;
begin
 with DataModule8 do begin
```

```
  Table1.First; founded:=False;
  while (not Table1.Eof)and(not founded) do begin
   if CompareText(Edit8.Text,Table1ID_NO.Text)=0 then founded:=True;
   if not founded then Table1.Next;
  end;
  if founded then begin
   if Table1TYPE.Text='Retired' then begin
    RStatus:='Update';
    Edit8.Text:=Table1ID_NO.Text;
    Edit9.Text:=Table1NAME.Text;
    Edit10.Text:=Table1SURNAME.Text;
    MaskEdit3.Text:=Table1BIRTHDATE.Text;
    Edit12.Text:=Table1ADDRESS.Text;
    ComboBox3.Text:=Table1GENDER.Text;
    Edit13.Text:=Table1RETIRED_ID.Text;
    ComboBox9.Text:=Table1RETIRED_TYPE.Text;
   end
   else begin
    ShowMessage('You can`t enter this Id No');
    AllInVisible;
    Edit8.SetFocus;
   end;
  end
  else begin
   RStatus:='Normal';
  end;
 end;

end;

procedure TForm3.Edit1Exit(Sender: TObject);
var
 founded : Boolean;
```

```
begin
  with DataModule8 do begin
    Table1.First; founded:=False;
    while (not Table1.Eof)and(not founded) do begin
      if CompareText(Edit1.Text,Table1ID_NO.Text)=0 then founded:=True;
      if not founded then Table1.Next;
    end;
    if founded then begin
      if Table1TYPE.Text='Insured' then begin
        RStatus:='Update';
        Edit1.Text:=Table1ID_NO.Text;
        Edit2.Text:=Table1NAME.Text;
        Edit3.Text:=Table1SURNAME.Text;
        MaskEdit4.Text:=Table1BIRTHDATE.Text;
        Edit5.Text:=Table1ADDRESS.Text;
        ComboBox1.Text:=Table1GENDER.Text;
        Edit6.Text:=Table1INSURED_ID.Text;
        ComboBox10.Text:=Table1INSURED_TYPE.Text;
      end
      else begin
        ShowMessage('You can`t enter this Id No');
        AllInVisible;
        Edit1.SetFocus;
      end;
    end
    else begin
      RStatus:='Normal';
    end;
  end;
end;

procedure TForm3.Edit22Exit(Sender: TObject);
var
```

```
  founded : Boolean;
begin
  with DataModule8 do begin
    Table1.First; founded:=False;
    while (not Table1.Eof)and(not founded) do begin
      if CompareText(Edit22.Text,Table1ID_NO.Text)=0 then founded:=True;
      if not founded then Table1.Next;
    end;
    if founded then begin
      if Table1TYPE.Text='Unpaid' then begin
        RStatus:='Update';
        Edit22.Text:=Table1ID_NO.Text;
        Edit23.Text:=Table1NAME.Text;
        Edit24.Text:=Table1SURNAME.Text;
        MaskEdit1.Text:=Table1BIRTHDATE.Text;
        Edit26.Text:=Table1ADDRESS.Text;
        ComboBox7.Text:=Table1GENDER.Text;
        Edit27.Text:=Table1UNPAID_ID.Text;
        ComboBox11.Text:=Table1UNPAID_TYPE.Text;
      end
      else begin
        ShowMessage('You can`t enter this Id No');
        AllInVisible;
        Edit22.SetFocus;
      end;
    end
    else begin
      RStatus:='Normal';
    end;
  end;
end;

procedure TForm3.INSUREDChange(Sender: TObject);
```

```
begin
  AllInVisible;
end;

procedure TForm3.BitBtn6Click(Sender: TObject);
var
  xf,yf : Integer;
begin
  xf:=400; yf:=200;
  Printer.BeginDoc;
  Printer.Canvas.Font.Pitch:=fpFixed;
  Printer.Canvas.Font.Size:=16;
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+450,yf-50 ,'Retired Patient');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];
  Printer.Canvas.Font.Size:=12;
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf    ,yf+100,'Id No     : ');
  Printer.Canvas.TextOut(xf    ,yf+170,'Name      : ');
  Printer.Canvas.TextOut(xf    ,yf+240,'Surname   : ');
  Printer.Canvas.TextOut(xf    ,yf+310,'Birthdate : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+350,yf+100,Edit8.Text);
  Printer.Canvas.TextOut(xf+350,yf+170,Edit9.Text);
  Printer.Canvas.TextOut(xf+350,yf+240,Edit10.Text);
  Printer.Canvas.TextOut(xf+350,yf+310,MaskEdit3.Text);

  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+1000,yf+100,'Address      : ');
  Printer.Canvas.TextOut(xf+1000,yf+170,'Gender       : ');
  Printer.Canvas.TextOut(xf+1000,yf+240,'Retired      : ');
  Printer.Canvas.TextOut(xf+1000,yf+310,'Retired Type : ');
```

```
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+1450,yf+100,Edit12.Text);
  Printer.Canvas.TextOut(xf+1450,yf+170,ComboBox3.Text);
  Printer.Canvas.TextOut(xf+1450,yf+240,Edit13.Text);
  Printer.Canvas.TextOut(xf+1450,yf+310,ComboBox9.Text);
  Printer.EndDoc;
end;

procedure TForm3.BitBtn3Click(Sender: TObject);
var
  xf,yf : Integer;
begin
  xf:=400; yf:=200;
  Printer.BeginDoc;
  Printer.Canvas.Font.Pitch:=fpFixed;
  Printer.Canvas.Font.Size:=16;
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+450,yf-50 ,'Insured Patient');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];
  Printer.Canvas.Font.Size:=12;
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf    ,yf+100,'Id No     : ');
  Printer.Canvas.TextOut(xf    ,yf+170,'Name      : ');
  Printer.Canvas.TextOut(xf    ,yf+240,'Surname   : ');
  Printer.Canvas.TextOut(xf    ,yf+310,'Birthdate : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+350,yf+100,Edit1.Text);
  Printer.Canvas.TextOut(xf+350,yf+170,Edit2.Text);
  Printer.Canvas.TextOut(xf+350,yf+240,Edit3.Text);
  Printer.Canvas.TextOut(xf+350,yf+310,MaskEdit4.Text);
```

```
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+1000,yf+100,'Address     : ');
  Printer.Canvas.TextOut(xf+1000,yf+170,'Gender      : ');
  Printer.Canvas.TextOut(xf+1000,yf+240,'Insured Id  : ');
  Printer.Canvas.TextOut(xf+1000,yf+310,'Insured Type: ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+1450,yf+100,Edit5.Text);
  Printer.Canvas.TextOut(xf+1450,yf+170,ComboBox1.Text);
  Printer.Canvas.TextOut(xf+1450,yf+240,Edit6.Text);
  Printer.Canvas.TextOut(xf+1450,yf+310,ComboBox10.Text);
  Printer.EndDoc;
end;

procedure TForm3.BitBtn12Click(Sender: TObject);
var
  xf,yf : Integer;
begin
  xf:=400; yf:=200;
  Printer.BeginDoc;
  Printer.Canvas.Font.Pitch:=fpFixed;
  Printer.Canvas.Font.Size:=16;
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+450,yf-50 ,'Unpaid Patient');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];
  Printer.Canvas.Font.Size:=12;
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf    ,yf+100,'Id No     : ');
  Printer.Canvas.TextOut(xf    ,yf+170,'Name      : ');
  Printer.Canvas.TextOut(xf    ,yf+240,'Surname   : ');
  Printer.Canvas.TextOut(xf    ,yf+310,'Birthdate : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];
```

```
  Printer.Canvas.TextOut(xf+350,yf+100,Edit22.Text);
  Printer.Canvas.TextOut(xf+350,yf+170,Edit23.Text);
  Printer.Canvas.TextOut(xf+350,yf+240,Edit24.Text);
  Printer.Canvas.TextOut(xf+350,yf+310,MaskEdit1.Text);

  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+1000,yf+100,'Address     : ');
  Printer.Canvas.TextOut(xf+1000,yf+170,'Gender      : ');
  Printer.Canvas.TextOut(xf+1000,yf+240,'Unpaid Id   : ');
  Printer.Canvas.TextOut(xf+1000,yf+310,'Unpaid Type : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+1450,yf+100,Edit26.Text);
  Printer.Canvas.TextOut(xf+1450,yf+170,ComboBox7.Text);
  Printer.Canvas.TextOut(xf+1450,yf+240,Edit27.Text);
  Printer.Canvas.TextOut(xf+1450,yf+310,ComboBox11.Text);
  Printer.EndDoc;
end;

end.
```

**C.5.4) Patient Process Screen:**

Patient Process Menu consist of three part. They are Input Patient, X-Ray Form, Laboratory Form.(figure4)

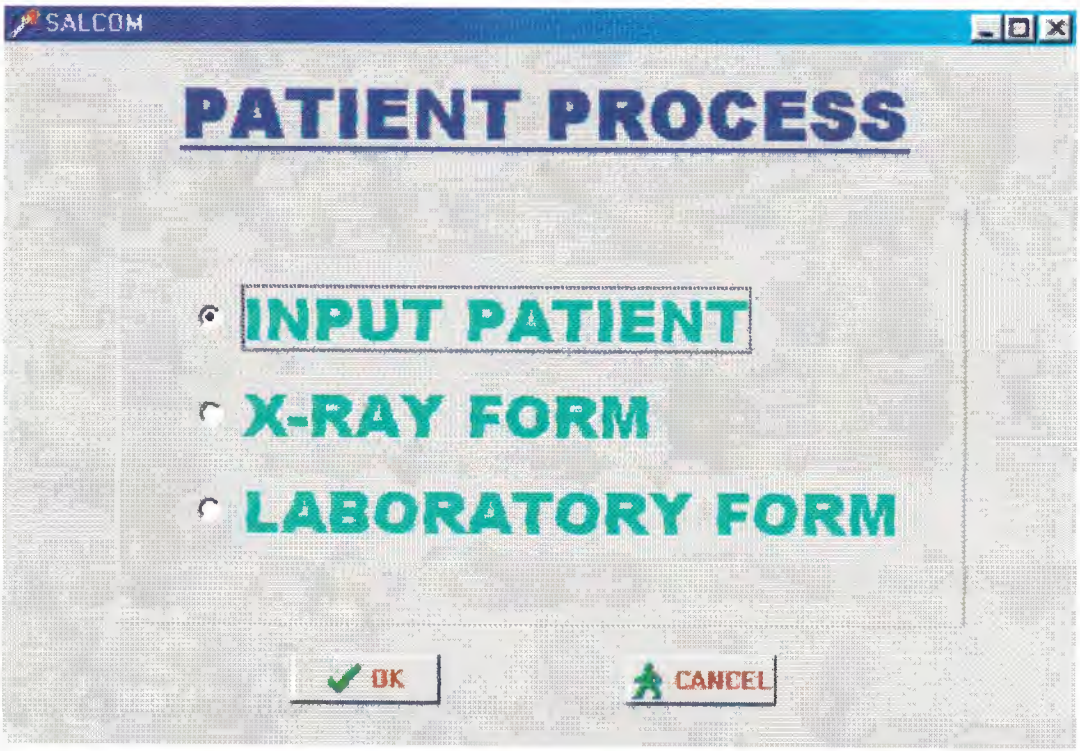



Figure4

```
unit Unit4;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  unit5, unit6, unit7, StdCtrls, Buttons, ExtCtrls;

type
  TForm4 = class(TForm)
    Panel1: TPanel;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    Label1: TLabel;
    RadioButton3: TRadioButton;
```

```
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure FormKeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form4: TForm4;

implementation

{$R *.DFM}

procedure TForm4.BitBtn2Click(Sender: TObject);
begin
   CLOSE;
end;

procedure TForm4.BitBtn1Click(Sender: TObject);
begin
     if radiobutton1.checked=true then
     form5.visible:=true
     else if radiobutton2.checked=true then
     form6.visible:=true
     else  if radiobutton3.checked=true then
          form7.visible:=true;

end;

procedure TForm4.FormKeyPress(Sender: TObject; var Key: Char);
```

```
begin
  if Key=#13 then begin
    if RadioButton1.Checked then form5.Show;
    if RadioButton2.Checked then form6.Show;
    if RadioButton3.Checked then form7.Show;
  end;
  if Key=#27 then Close;
end;
end.
```

**C.5.4.1) Input Patient Screen:**

Input patient menu consist of Id No, Name, Surname, Birthdate, Payment Type, Room No, Bed No, Service, Entry Date, Exit Date, Diagnosis, Medicine. If the patient has recorded before information will be displayed on screen.(figure4.1)

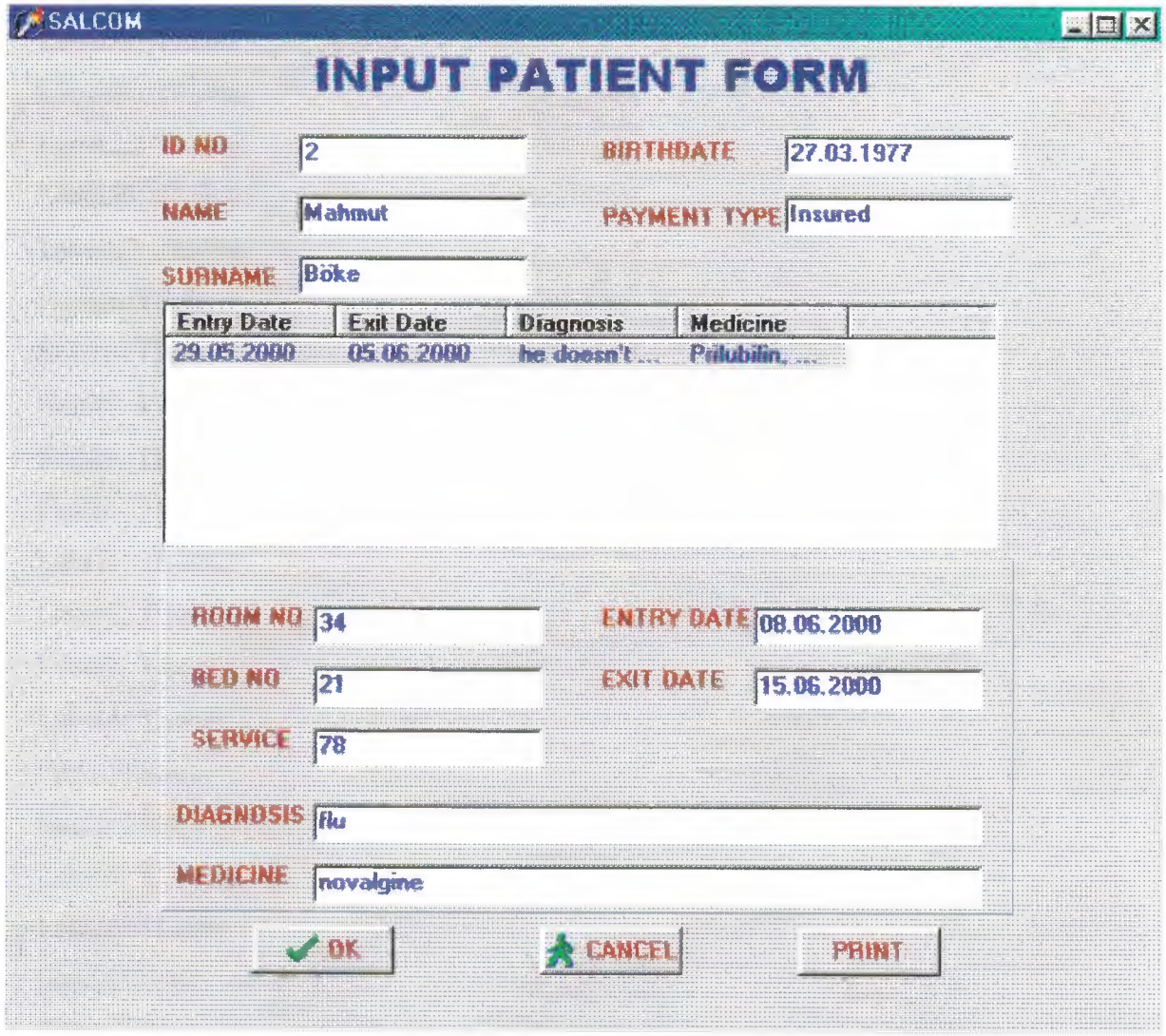



figure4.1

```
unit Unit5;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls, Buttons, Mask, Grids, DBGrids, ComCtrls, Printers;

type
```

```
TForm5 = class(TForm)
  BitBtn7: TBitBtn;
  BitBtn8: TBitBtn;
  BitBtn9: TBitBtn;
  Label19: TLabel;
  Label20: TLabel;
  Label21: TLabel;
  Label22: TLabel;
  Label26: TLabel;
  Label27: TLabel;
  Edit15: TEdit;
  Edit16: TEdit;
  Edit17: TEdit;
  Edit21: TEdit;
  Panel1: TPanel;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Edit1: TEdit;
  Edit2: TEdit;
  Label5: TLabel;
  Label6: TLabel;
  Edit5: TEdit;
  Edit6: TEdit;
  Label7: TLabel;
  MaskEdit1: TMaskEdit;
  MaskEdit2: TMaskEdit;
  Mask: TMaskEdit;
  Label8: TLabel;
  Edit3: TEdit;
  ListView1: TListView;
  procedure BitBtn8Click(Sender: TObject);
```

```
    procedure Edit15Exit(Sender: TObject);
    procedure BitBtn7Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure Edit21Exit(Sender: TObject);
    procedure Edit6Exit(Sender: TObject);
    procedure FormKeyPress(Sender: TObject; var Key: Char);
    procedure BitBtn9Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form5: TForm5;
  Items : TListItem;
  Cols  : TListColumn;
implementation

uses unit8;

{$R *.DFM}

procedure InitialValues;
begin
  with form5 do begin
    Edit15.Text:='';
    Edit16.Text:='';
    Edit17.Text:='';
    Mask.Text:='';
    Edit21.Text:='';
    ListView1.Items.Clear;
    Edit1.Text:='';
```

```
    Edit2.Text:='';
    Edit3.Text:='';
    MaskEdit1.Text:='';
    MaskEdit2.Text:='';
    Edit5.Text:='';
    Edit6.Text:='';
  end;
end;

procedure TForm5.BitBtn8Click(Sender: TObject);
begin
   Close;
end;

procedure TForm5.Edit15Exit(Sender: TObject);
var
  founded : Boolean;
begin
  if Edit15.Text<>'' then begin
    with DataModule8 do begin
      Table1.First; founded:=False;
      while (not table1.Eof)and(not founded) do begin
        if CompareText(Edit15.Text,Table1ID_NO.Text)=0 then begin
          founded:=True;
        end;
        if not founded then table1.Next;
      end;
      if not founded then begin
        ShowMessage('There is no patient which has that ID NO');
        Edit15.SetFocus;
      end
      else begin {if founded}
        InitialValues;
```

```
      Edit15.Text:=Table1ID_NO.Text;
      Edit16.Text:=Table1NAME.Text;
      Edit17.Text:=Table1SURNAME.Text;
      Mask.Text:=Table1BIRTHDATE.Text;
      Edit21.Text:=Table1TYPE.Text;
      table2.First; ListView1.Items.Clear; ListView1.Columns.Clear;
      Cols:=ListView1.Columns.Add;
      Cols.Caption:='Entry Date';
      Cols.Width:=90;
      Cols:=ListView1.Columns.Add;
      Cols.Caption:='Exit Date';
      Cols.Width:=90;
      Cols:=ListView1.Columns.Add;
      Cols.Caption:='Diagnosis';
      Cols.Width:=90;
      Cols:=ListView1.Columns.Add;
      Cols.Caption:='Medicine';
      Cols.Width:=90;
      while not Table2.Eof do begin
       if CompareText(Edit15.Text,Table2ID_NO.Text)=0 then begin
         Items:=ListView1.Items.Add;
         Items.Caption:=Table2ENTRY_DATE.Text;
         Items.SubItems.Add(Table2EXIT_DATE.Text);
         Items.SubItems.Add(Table2DIAGNOSIS.Text);
         Items.SubItems.Add(Table2MEDICINE.Text);
       end;
       Table2.Next;
      end;
      Edit1.SetFocus;
    end;
  end;
 end;
end;
```

```
procedure TForm5.BitBtn7Click(Sender: TObject);
begin
 with DataModule8 do begin
  Table2.Append;
  Table2ID_NO.Text:=Edit15.Text;
  Table2ROOM_NO.Text:=Edit1.Text;
  Table2BED_NO.Text:=Edit2.Text;
  Table2SERVICE.Text:=Edit3.Text;
  Table2ENTRY_DATE.Text:=MaskEdit1.Text;
  Table2EXIT_DATE.Text:=MaskEdit2.Text;
  Table2DIAGNOSIS.Text:=Edit5.Text;
  Table2MEDICINE.Text:=Edit6.Text;
  Table2.Post;
 end;
 InitialValues;
 Edit15.SetFocus;
end;

procedure TForm5.FormActivate(Sender: TObject);
begin
 InitialValues;
 Edit15.SetFocus;
end;

procedure TForm5.Edit21Exit(Sender: TObject);
begin
 Edit1.SetFocus;
end;

procedure TForm5.Edit6Exit(Sender: TObject);
begin
 BitBtn7.SetFocus;
```

```
end;

procedure TForm5.FormKeyPress(Sender: TObject; var Key: Char);
begin
 if Key=#13 then begin
  Key:=#0;
  PostMessage(Handle,WM_KEYDOWN,09,0);
 end;
 if Key=#27 then Close;
end;

procedure TForm5.BitBtn9Click(Sender: TObject);
var
 xf,yf : Integer;
begin
 xf:=400; yf:=200;
 Printer.BeginDoc;
 Printer.Canvas.Font.Pitch:=fpFixed;
 Printer.Canvas.Font.Size:=16;
 Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
 Printer.Canvas.TextOut(xf+450,yf-50 ,'Input Patient');
 Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];
 Printer.Canvas.Font.Size:=12;
 Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
 Printer.Canvas.TextOut(xf    ,yf+100,'Id No     : ');
 Printer.Canvas.TextOut(xf    ,yf+170,'Name      : ');
 Printer.Canvas.TextOut(xf    ,yf+240,'Surname   : ');

 Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

 Printer.Canvas.TextOut(xf+350,yf+100,Edit15.Text);
 Printer.Canvas.TextOut(xf+350,yf+170,Edit16.Text);
 Printer.Canvas.TextOut(xf+350,yf+240,Edit17.Text);
```

```
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+1000,yf+100,'Birthdate    : ');
  Printer.Canvas.TextOut(xf+1000,yf+170,'Payment Type : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+1450,yf+100,Mask.Text);
  Printer.Canvas.TextOut(xf+1450,yf+170,Edit21.Text);

  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf    ,yf+500,'Room No   : ');
  Printer.Canvas.TextOut(xf    ,yf+570,'Bed No    : ');
  Printer.Canvas.TextOut(xf    ,yf+640,'Service   : ');
  Printer.Canvas.TextOut(xf    ,yf+710,'Diagnosis : ');
  Printer.Canvas.TextOut(xf    ,yf+780,'Medicine  : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+350,yf+500,Edit1.Text);
  Printer.Canvas.TextOut(xf+350,yf+570,Edit2.Text);
  Printer.Canvas.TextOut(xf+350,yf+640,Edit3.Text);
  Printer.Canvas.TextOut(xf+350,yf+710,Edit5.Text);
  Printer.Canvas.TextOut(xf+350,yf+780,Edit6.Text);

  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+1000,yf+500,'Entry Date : ');
  Printer.Canvas.TextOut(xf+1000,yf+570,'Exit Date  : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+1450,yf+500,MaskEdit1.Text);
  Printer.Canvas.TextOut(xf+1450,yf+570,MaskEdit2.Text);

  Printer.EndDoc;
end; end.
```

**C.5.4.2) X-Ray Screen:**

X-Ray menu consist of Id No, Name, Surname, Birthdate, Payment Type, Type(head, brain, jaw, neck, lung, liver, thorax, hand, arm, foot), Broken Type(piece, unpiece, open, close), Number. If the patient has recorded before information will be displayed on screen.(figure4.2) If the user want to printout user must click the 'Print'.

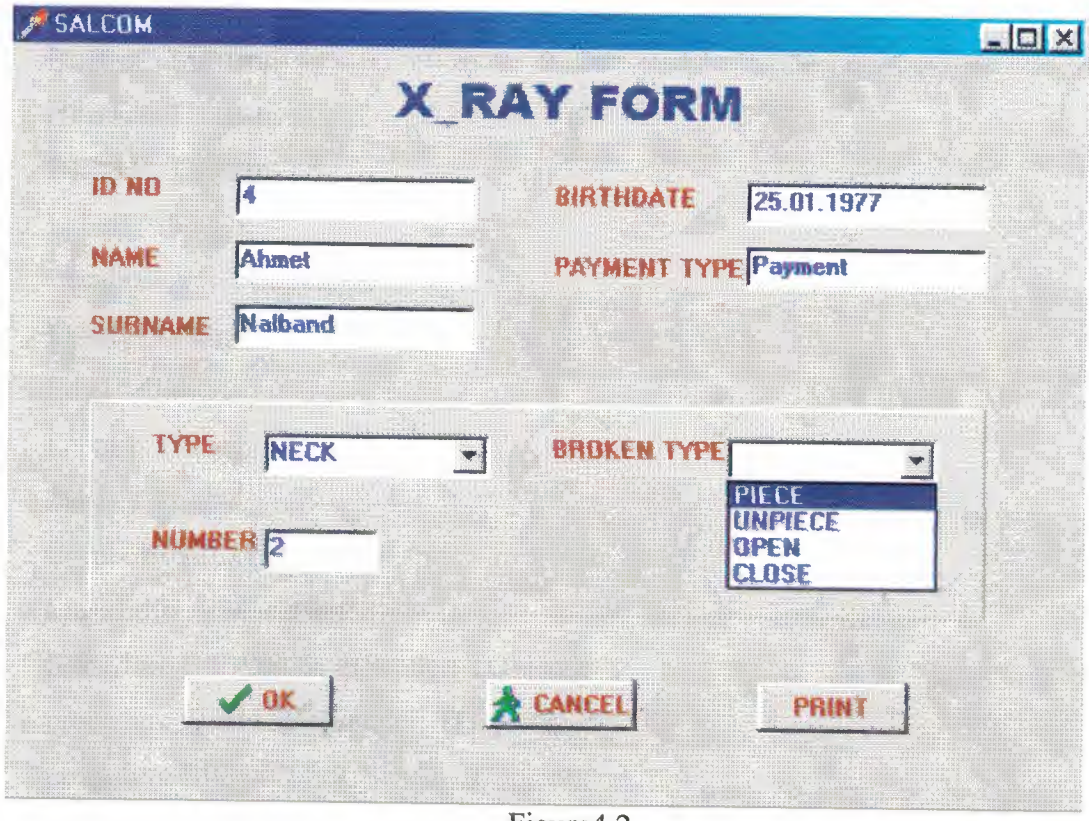



Figure4.2

```
unit Unit6;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ExtCtrls, Mask, Printers;

type
  TForm6 = class(TForm)
    Label19: TLabel;
    Label20: TLabel;
    Label21: TLabel;
```

```
    Label22: TLabel;
    Label26: TLabel;
    Edit15: TEdit;
    Edit16: TEdit;
    Edit17: TEdit;
    Edit21: TEdit;
    Panel1: TPanel;
    BitBtn7: TBitBtn;
    BitBtn8: TBitBtn;
    BitBtn9: TBitBtn;
    Label1: TLabel;
    ComboBox1: TComboBox;
    Label2: TLabel;
    Edit1: TEdit;
    Label4: TLabel;
    Label3: TLabel;
    ComboBox2: TComboBox;
    MaskEdit1: TMaskEdit;
    procedure BitBtn8Click(Sender: TObject);
    procedure Edit15Exit(Sender: TObject);
    procedure BitBtn7Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure FormKeyPress(Sender: TObject; var Key: Char);
    procedure BitBtn9Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form6: TForm6;
```

```
implementation

uses unit8;

{$R *.DFM}

procedure InitialValues;
begin
  with Form6 do begin
    Edit15.Text:='';
    Edit16.Text:='';
    Edit17.Text:='';
    MaskEdit1.Text:='';
    Edit21.Text:='';
    ComboBox1.Text:='';
    ComboBox2.Text:='';
    Edit1.Text:='';
  end;
end;

procedure TForm6.BitBtn8Click(Sender: TObject);
begin
   close;
end;

procedure TForm6.Edit15Exit(Sender: TObject);
var
  founded : Boolean;
begin
  with DataModule8 do begin
    if Edit15.Text<>'' then begin
      Table1.First; founded:=False;
      while (not Table1.Eof)and(not founded) do begin
```

```
      if CompareText(Edit15.Text,Table1ID_NO.Text)=0 then founded:=True;
      if not founded then table1.Next;
    end;
    if (not founded) then begin
      ShowMessage('There is no patient which has that ID NO');
      Edit15.Text;
    end
    else begin
      Edit15.Text:=Table1ID_NO.Text;
      Edit16.Text:=Table1NAME.Text;
      Edit17.Text:=Table1SURNAME.Text;
      MaskEdit1.Text:=Table1BIRTHDATE.Text;
      Edit21.Text:=Table1TYPE.Text;
      ComboBox1.SetFocus;
    end;
   end;
  end;
end;

procedure TForm6.BitBtn7Click(Sender: TObject);
begin
  with DataModule8 do begin
   Table3.Append;
   Table3ID_NO.Text:=Edit15.Text;
   Table3TYPE.Text:=ComboBox1.Text;
   Table3BROKEN_TYPE.Text:=ComboBox2.Text;
   Table3NUMBER.Text:=Edit1.Text;
   Table3.Post;
  end;
  InitialValues;
  Edit15.SetFocus;
end;
```

```
procedure TForm6.FormActivate(Sender: TObject);
begin
  InitialValues;
  Edit15.SetFocus;
end;

procedure TForm6.FormKeyPress(Sender: TObject; var Key: Char);
begin
  if Key=#27 then close;
end;

procedure TForm6.BitBtn9Click(Sender: TObject);
var
  xf,yf : Integer;
begin
  xf:=400; yf:=200;
  Printer.BeginDoc;
  Printer.Canvas.Font.Pitch:=fpFixed;
  Printer.Canvas.Font.Size:=16;
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+450,yf-50 ,'X - Ray');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];
  Printer.Canvas.Font.Size:=12;
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf    ,yf+100,'Id No     : ');
  Printer.Canvas.TextOut(xf    ,yf+170,'Name      : ');
  Printer.Canvas.TextOut(xf    ,yf+240,'Surname   : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+350,yf+100,Edit15.Text);
  Printer.Canvas.TextOut(xf+350,yf+170,Edit16.Text);
  Printer.Canvas.TextOut(xf+350,yf+240,Edit17.Text);
```

```
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+1000,yf+100,'Birthdate    : ');
  Printer.Canvas.TextOut(xf+1000,yf+170,'Payment Type : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+1450,yf+100,MaskEdit1.Text);
  Printer.Canvas.TextOut(xf+1450,yf+170,Edit21.Text);

  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf    ,yf+500,'Type   : ');
  Printer.Canvas.TextOut(xf    ,yf+570,'Number : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+350,yf+500,ComboBox1.Text);
  Printer.Canvas.TextOut(xf+350,yf+570,Edit1.Text);

  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+1000,yf+500,'Broken Type : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+1450,yf+500,ComboBox2.Text);

  Printer.EndDoc;
end;

end.
```

## C.5.4.3) Laboratory Screen:

There is only one form in Laboratory Screen. But this form consist of two part, Biochemistry Form(figure4.3.1) and Urine Form(figure4.3.2).

### i) Biochemistry Form:

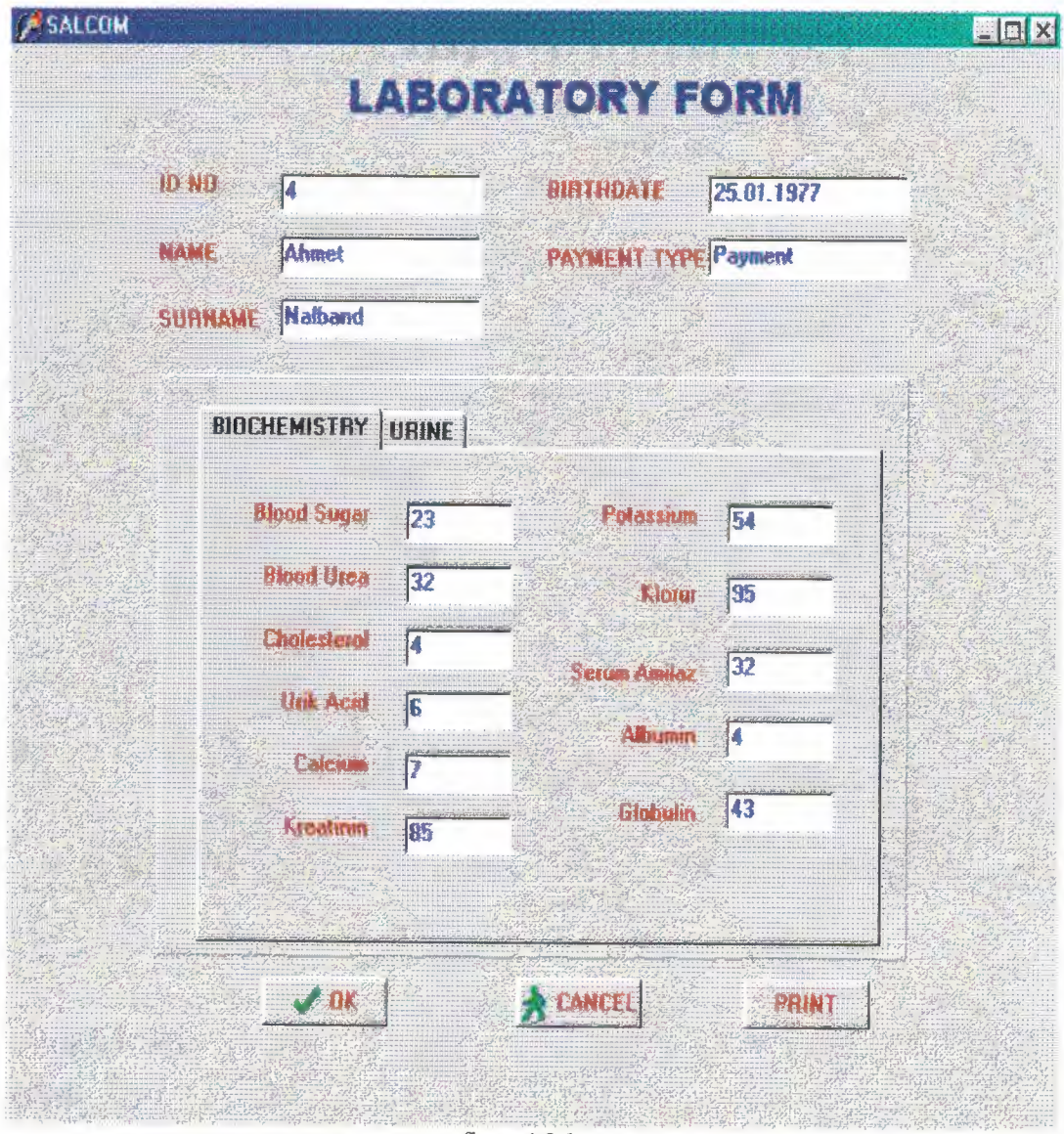



figure4.3.1

**ii) Urine Form:**

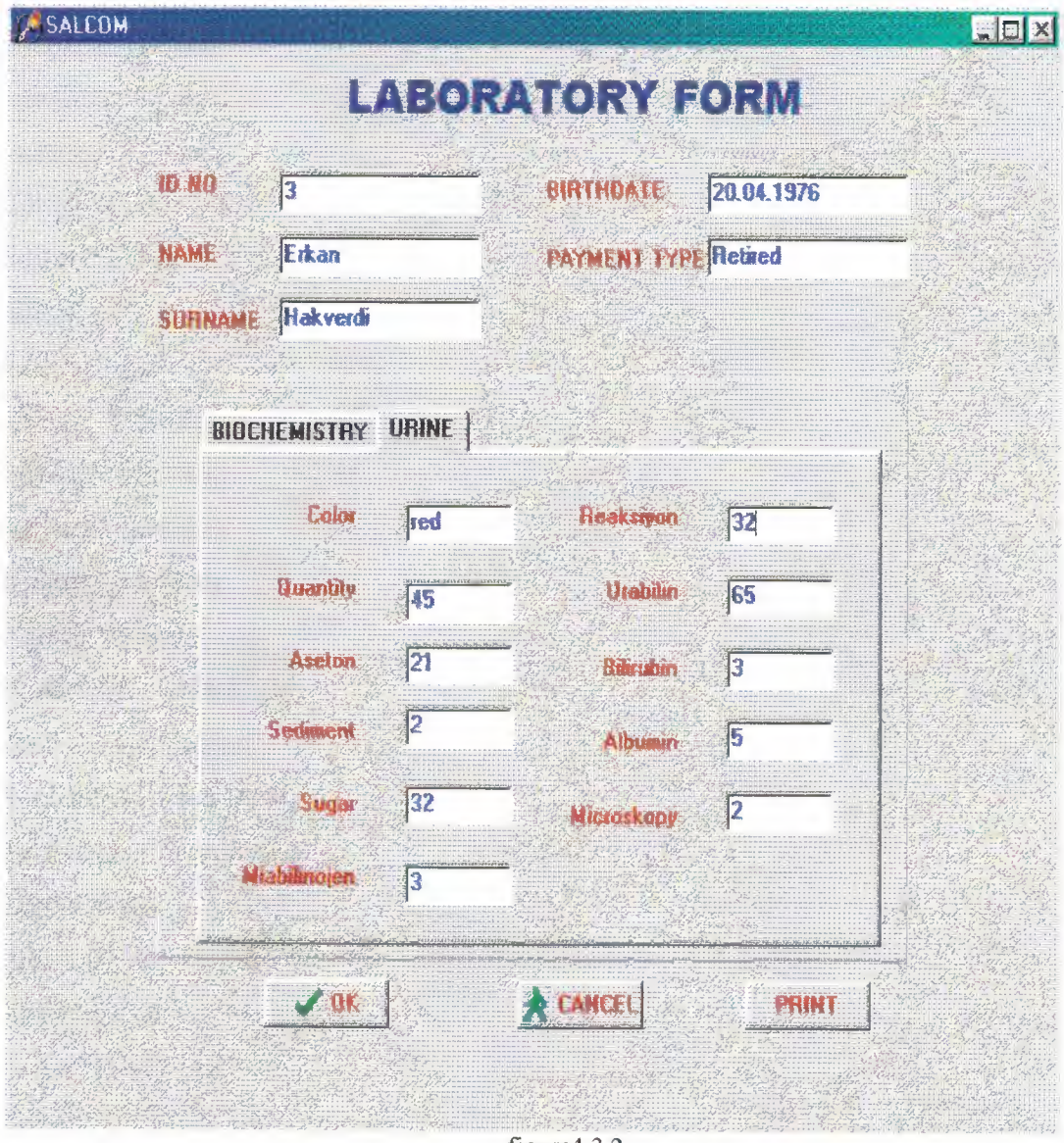



figure4.3.2

```
unit Unit7;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls, Buttons, Mask, ComCtrls, Printers;

type
  TForm7 = class(TForm)
```

Label19: TLabel;

Label20: TLabel;

Label21: TLabel;

Label22: TLabel;

Label26: TLabel;

Edit15: TEdit;

Edit16: TEdit;

Edit17: TEdit;

Edit21: TEdit;

Label1: TLabel;

BitBtn7: TBitBtn;

BitBtn8: TBitBtn;

BitBtn9: TBitBtn;

Panel1: TPanel;

MaskEdit1: TMaskEdit;

alala: TPageControl;

TabSheet1: TTabSheet;

TabSheet2: TTabSheet;

Edit1: TEdit;

Edit2: TEdit;

Edit3: TEdit;

Edit4: TEdit;

Edit5: TEdit;

Edit6: TEdit;

Edit7: TEdit;

Edit8: TEdit;

Edit9: TEdit;

Edit10: TEdit;

Edit11: TEdit;

e1: TEdit;

e2: TEdit;

e3: TEdit;

e4: TEdit;

```
e5: TEdit;
e6: TEdit;
e7: TEdit;
e8: TEdit;
e9: TEdit;
e10: TEdit;
e11: TEdit;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Label16: TLabel;
Label17: TLabel;
Label18: TLabel;
Label23: TLabel;
Label24: TLabel;
Label25: TLabel;
Label27: TLabel;
Label28: TLabel;
procedure BitBtn8Click(Sender: TObject);
procedure Edit15Exit(Sender: TObject);
procedure BitBtn7Click(Sender: TObject);
procedure FormActivate(Sender: TObject);
```

```
  procedure FormKeyPress(Sender: TObject; var Key: Char);
  procedure BitBtn9Click(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
 Form7: TForm7;

implementation

uses Unit8;

{$R *.DFM}

procedure InitialValues;
begin
 with Form7 do begin
  Edit15.Text:='';
  Edit16.Text:='';
  Edit17.Text:='';
  MaskEdit1.Text:='';
  Edit21.Text:='';
  Edit1.Text:='';
  Edit2.Text:='';
  Edit3.Text:='';
  Edit4.Text:='';
  Edit5.Text:='';
  Edit6.Text:='';
  Edit7.Text:='';
  Edit8.Text:='';
```

```
   Edit9.Text:='';
   Edit10.Text:='';
   Edit11.Text:='';
   e1.Text:='';
   e2.Text:='';
   e3.Text:='';
   e4.Text:='';
   e5.Text:='';
   e6.Text:='';
   e7.Text:='';
   e8.Text:='';
   e9.Text:='';
   e10.Text:='';
   e11.Text:='';
  end;
end;

procedure TForm7.BitBtn8Click(Sender: TObject);
begin
  close;
end;

procedure TForm7.Edit15Exit(Sender: TObject);
var
  founded : Boolean;
begin
  with DataModule8 do begin
   if Edit15.Text<>'' then begin
    Table1.First; founded:=False;
    while (not Table1.Eof)and(not founded) do begin
     if CompareText(Edit15.Text,Table1ID_NO.Text)=0 then founded:=True;
     if not founded then Table1.Next;
    end;
```

```
    if not founded then begin
      ShowMessage('There is no patient which has that ID NO');
      Edit15.Text;
    end
    else begin
      Edit15.Text:=Table1ID_NO.Text;
      Edit16.Text:=Table1NAME.Text;
      Edit17.Text:=Table1SURNAME.Text;
      MaskEdit1.Text:=Table1BIRTHDATE.Text;
      Edit21.Text:=Table1TYPE.Text;
    end;
   end;
  end;
end;

procedure TForm7.BitBtn7Click(Sender: TObject);
begin
  with DataModule8 do begin
   if alala.ActivePage=TabSheet1 then begin
    Table4.Append;
    Table4ID_NO.Text:=Edit15.Text;
    Table4BLOOD_SUGAR.Text:=Edit1.Text;
    Table4BLOOD_UREA.Text:=Edit2.Text;
    Table4CHOLESTEROL.Text:=Edit3.Text;
    Table4URIK_ACID.Text:=Edit4.Text;
    Table4CALCIUM.Text:=Edit5.Text;
    Table4KREATININ.Text:=Edit6.Text;
    Table4POTASSIUM.Text:=Edit7.Text;
    Table4KLORUR.Text:=Edit8.Text;
    Table4SERUM_AMILAZ.Text:=Edit9.Text;
    Table4ALBUMIN.Text:=Edit10.Text;
    Table4GLOBULIN.Text:=Edit11.Text;
    Table4.Post;
```

```
    end
    else if alala.ActivePage=TabSheet2 then begin
     Table5.Append;
     Table5ID_NO.Text:=Edit15.Text;
     Table5COLOR.Text:=e1.Text;
     Table5QUANTITY.Text:=e2.Text;
     Table5ASETON.Text:=e3.Text;
     Table5SEDIMENT.Text:=e4.Text;
     Table5SUGAR.Text:=e5.Text;
     Table5MIABILINOJEN.Text:=e6.Text;
     Table5REAKSIYON.Text:=e7.Text;
     Table5URABULIN.Text:=e8.Text;
     Table5BILIRUBIN.Text:=e9.Text;
     Table5ALBUMIN.Text:=e10.Text;
     Table5MICROSKOPY.Text:=e11.Text;
     Table5.Post;
    end;
  end;
  InitialValues;
  Edit15.SetFocus;
end;

procedure TForm7.FormActivate(Sender: TObject);
begin
  InitialValues;
  Edit15.SetFocus;
end;

procedure TForm7.FormKeyPress(Sender: TObject; var Key: Char);
begin
  if Key=#13 then begin
    Key:=#0;
    PostMessage(Handle,WM_KEYDOWN,09,0);
```

```
  end;
  if Key=#27 then close;
end;

procedure TForm7.BitBtn9Click(Sender: TObject);
var
  xf,yf : Integer;
begin
  xf:=400; yf:=200;
  Printer.BeginDoc;
  Printer.Canvas.Font.Pitch:=fpFixed;
  Printer.Canvas.Font.Size:=16;
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+450,yf-50 ,'Laboratory');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];
  Printer.Canvas.Font.Size:=12;
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf    ,yf+100,'Id No     : ');
  Printer.Canvas.TextOut(xf    ,yf+170,'Name      : ');
  Printer.Canvas.TextOut(xf    ,yf+240,'Surname   : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+350,yf+100,Edit15.Text);
  Printer.Canvas.TextOut(xf+350,yf+170,Edit16.Text);
  Printer.Canvas.TextOut(xf+350,yf+240,Edit17.Text);

  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+1000,yf+100,'Birthdate    : ');
  Printer.Canvas.TextOut(xf+1000,yf+170,'Payment Type : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+1450,yf+100,MaskEdit1.Text);
  Printer.Canvas.TextOut(xf+1450,yf+170,Edit21.Text);
```

```
if alala.ActivePage=TabSheet1 then begin
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf    ,yf+500,'Blood Sugar : ');
  Printer.Canvas.TextOut(xf    ,yf+570,'Blood Urea  : ');
  Printer.Canvas.TextOut(xf    ,yf+640,'Cholesterol : ');
  Printer.Canvas.TextOut(xf    ,yf+710,'Urik Acid   : ');
  Printer.Canvas.TextOut(xf    ,yf+780,'Calcium     : ');
  Printer.Canvas.TextOut(xf    ,yf+850,'Kreatinin   : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+450,yf+500,Edit1.Text);
  Printer.Canvas.TextOut(xf+450,yf+570,Edit2.Text);
  Printer.Canvas.TextOut(xf+450,yf+640,Edit3.Text);
  Printer.Canvas.TextOut(xf+450,yf+710,Edit4.Text);
  Printer.Canvas.TextOut(xf+450,yf+780,Edit5.Text);
  Printer.Canvas.TextOut(xf+450,yf+850,Edit6.Text);

  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+1000,yf+500,'Potassium    : ');
  Printer.Canvas.TextOut(xf+1000,yf+570,'Klorur       : ');
  Printer.Canvas.TextOut(xf+1000,yf+640,'Serum Amilaz : ');
  Printer.Canvas.TextOut(xf+1000,yf+710,'Albumin      : ');
  Printer.Canvas.TextOut(xf+1000,yf+780,'Globulin     : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+1450,yf+500,Edit7.Text);
  Printer.Canvas.TextOut(xf+1450,yf+570,Edit8.Text);
  Printer.Canvas.TextOut(xf+1450,yf+640,Edit9.Text);
  Printer.Canvas.TextOut(xf+1450,yf+710,Edit10.Text);
  Printer.Canvas.TextOut(xf+1450,yf+780,Edit11.Text);
end;
if alala.ActivePage=TabSheet2 then begin
```

```
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf   ,yf+500,'Color       : ');
  Printer.Canvas.TextOut(xf   ,yf+570,'Quantity    : ');
  Printer.Canvas.TextOut(xf   ,yf+640,'Aseton      : ');
  Printer.Canvas.TextOut(xf   ,yf+710,'Sediment    : ');
  Printer.Canvas.TextOut(xf   ,yf+780,'Sugar       : ');
  Printer.Canvas.TextOut(xf   ,yf+850,'Miabilinojen : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+450,yf+500,e1.Text);
  Printer.Canvas.TextOut(xf+450,yf+570,e2.Text);
  Printer.Canvas.TextOut(xf+450,yf+640,e3.Text);
  Printer.Canvas.TextOut(xf+450,yf+710,e4.Text);
  Printer.Canvas.TextOut(xf+450,yf+780,e5.Text);
  Printer.Canvas.TextOut(xf+450,yf+850,e6.Text);

  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style+[fsBold];
  Printer.Canvas.TextOut(xf+1000,yf+500,'Reaksiyon   : ');
  Printer.Canvas.TextOut(xf+1000,yf+570,'Urabulin    : ');
  Printer.Canvas.TextOut(xf+1000,yf+640,'Bilirubin   : ');
  Printer.Canvas.TextOut(xf+1000,yf+710,'Albumin     : ');
  Printer.Canvas.TextOut(xf+1000,yf+780,'Microskopy  : ');
  Printer.Canvas.Font.Style:=Printer.Canvas.Font.Style-[fsBold];

  Printer.Canvas.TextOut(xf+1450,yf+500,e7.Text);
  Printer.Canvas.TextOut(xf+1450,yf+570,e8.Text);
  Printer.Canvas.TextOut(xf+1450,yf+640,e9.Text);
  Printer.Canvas.TextOut(xf+1450,yf+710,e10.Text);
  Printer.Canvas.TextOut(xf+1450,yf+780,e11.Text);
 end;

 Printer.EndDoc;
end; end.
```

## D) DATA STRUCTURE

```
TForm1 = class(TForm)
StaticText1: TStaticText;
Image1: TImage;
Panel1: TPanel;
Label1: TLabel;
Edit1: TEdit;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
TForm2 = class(TForm)
Panel1: TPanel;
RadioButton1: TRadioButton;
RadioButton2: TRadioButton;
TForm3 = class(TForm)
INSURED: TPageControl;
TabSheet1: TTabSheet;
TabSheet2: TTabSheet;
TabSheet3: TTabSheet;
TabSheet4: TTabSheet;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Edit1: TEdit;
Edit2: TEdit;
Edit3: TEdit;
ComboBox1: TComboBox;
Label6: TLabel;
Label8: TLabel;
Label9: TLabel;
Edit5: TEdit;
Edit6: TEdit;
```

```
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
BitBtn3: TBitBtn;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Edit8: TEdit;
Edit9: TEdit;
Edit10: TEdit;
ComboBox3: TComboBox;
Label15: TLabel;
Label17: TLabel;
Edit12: TEdit;
Edit13: TEdit;
BitBtn4: TBitBtn;
BitBtn5: TBitBtn;
BitBtn6: TBitBtn;
Label19: TLabel;
Label20: TLabel;
Label21: TLabel;
Label22: TLabel;
Label23: TLabel;
Edit15: TEdit;
Edit16: TEdit;
Edit17: TEdit;
ComboBox5: TComboBox;
Label24: TLabel;
Label26: TLabel;
Label27: TLabel;
Edit19: TEdit;
Edit20: TEdit;
```

```
Edit21: TEdit;
BitBtn7: TBitBtn;
BitBtn8: TBitBtn;
BitBtn9: TBitBtn;
Label28: TLabel;
Label29: TLabel;
Label30: TLabel;
Label31: TLabel;
Label32: TLabel;
Edit22: TEdit;
Edit23: TEdit;
Edit24: TEdit;
ComboBox7: TComboBox;
Label33: TLabel;
Label35: TLabel;
Label36: TLabel;
Edit26: TEdit;
Edit27: TEdit;
BitBtn10: TBitBtn;
BitBtn11: TBitBtn;
BitBtn12: TBitBtn;
Label18: TLabel;
ComboBox9: TComboBox;
ComboBox10: TComboBox;
ComboBox11: TComboBox;
StaticText1: TStaticText;
MaskEdit1: TMaskEdit;
MaskEdit2: TMaskEdit;
MaskEdit3: TMaskEdit;
MaskEdit4: TMaskEdit;
TForm4 = class(TForm)
TForm7 = class(TForm)
Label19: TLabel;
```

```
    Label20: TLabel;
    Label21: TLabel;
    Label22: TLabel;
    Label26: TLabel;
    Edit15: TEdit;
    Edit16: TEdit;
    Edit17: TEdit;
    Edit21: TEdit;
    Label1: TLabel;
    BitBtn7: TBitBtn;
    BitBtn8: TBitBtn;
    BitBtn9: TBitBtn;
    Panel1: TPanel;
    MaskEdit1: TMaskEdit;
    alala: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
    Edit6: TEdit;
    Edit7: TEdit;
    Edit8: TEdit;
    Edit9: TEdit;
    Edit10: TEdit;
    Edit11: TEdit;
    e1: TEdit;
    e2: TEdit;
    e3: TEdit;
    e4: TEdit;
    e5: TEdit;
```

```
    e6: TEdit;
    e7: TEdit;
    e8: TEdit;
    e9: TEdit;
    e10: TEdit;
    e11: TEdit;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    Label15: TLabel;
    Label16: TLabel;
    Label17: TLabel;
    Label18: TLabel;
    Label23: TLabel;
    Label24: TLabel;
    Label25: TLabel;
    Label27: TLabel;
  TDataModule8 = class(TDataModule)
    Table1: TTable;
    Table1ID_NO: TFloatField;
    Table1NAME: TStringField;
    Table1SURNAME: TStringField;
    Table1BIRTHDATE: TDateField;
```

Table1GENDER: TStringField;
Table1ADDRESS: TStringField;
Table1SERVICE: TStringField;
Table1TYPE: TStringField;
Table1INSURED_ID: TFloatField;
Table1INSURED_TYPE: TStringField;
Table1RETIRED_ID: TFloatField;
Table1RETIRED_TYPE: TStringField;
Table1PAY: TFloatField;
Table1PROTOCOL_ID: TFloatField;
Table1UNPAID_ID: TFloatField;
Table1UNPAID_TYPE: TStringField;
Table2: TTable;
Table2ROOM_NO: TFloatField;
Table2BED_NO: TFloatField;
Table2ENTRY_DATE: TDateField;
Table2EXIT_DATE: TDateField;
Table2DIAGNOSIS: TStringField;
Table2MEDICINE: TStringField;
Table2ID_NO: TFloatField;
Table2SERVICE: TStringField;
Table3: TTable;
Table3ID_NO: TFloatField;
Table3TYPE: TStringField;
Table3BROKEN_TYPE: TStringField;
Table3NUMBER: TFloatField;
Table4: TTable;
Table4ID_NO: TFloatField;
Table4BLOOD_SUGAR: TFloatField;
Table4BLOOD_UREA: TFloatField;
Table4CHOLESTEROL: TFloatField;
Table4URIK_ACID: TFloatField;
Table4KREATININ: TFloatField;

Table4CALCIUM: TFloatField;

Table4POTASSIUM: TFloatField;

Table4KLORUR: TFloatField;

Table4SERUM_AMILAZ: TFloatField;

Table4ALBUMIN: TFloatField;

Table4GLOBULIN: TFloatField;

Table5: TTable;

Table5ID_NO: TFloatField;

Table5COLOR: TStringField;

Table5QUANTITY: TFloatField;

Table5ASETON: TFloatField;

Table5SEDIMENT: TFloatField;

Table5SUGAR: TFloatField;

Table5MIABILINOJEN: TFloatField;

Table5REAKSIYON: TFloatField;

Table5URABULIN: TFloatField;

Table5BILIRUBIN: TFloatField;

Table5ALBUMIN: TFloatField;

Table5MICROSKOPY: TFloatField;

# CONCLUSION

This project has concluded with more efficient processes. These are patient processes, database files to store data and printing on papers the requested information by the doctors or user.

The program can be used by the different small hospitals. It can meets the needs of those hospitals. But the importance of program is to allow the programmer to improve. This is upgradable of a program. And when improved can be used big hospital.

The printing module is also the most important part of program. Because the user can print his or her requested information on paper. And the also doctors can do.

Finally the interface of program are very easy and understandable. These shows the clear and useful of program.

# APPENDIX

# Unpaid Patient

| | | | |
|---|---|---|---|
| Id No | : 1 | Address | : YDU |
| Name | : Salih | Gender | : MALE |
| Surname | : Soysal | Unpaid Id | : 1 |
| Birthdate | : 21.05.1976 | Unpaid Type | : DESTITUTE |

# Insured Patient

| | | | |
|---|---|---|---|
| Id No | : 2 | Address | : Girne |
| Name | : Mahmut | Gender | : MALE |
| Surname | : Böke | Insured Id | : 5 |
| Birthdate | : 27.03.1977 | Insured Type | : WORKMAN |

# Retired Patient

| | | | |
|---|---|---|---|
| Id No | : 3 | Address | : Lapta |
| Name | : Erkan | Gender | : MALE |
| Surname | : Hakverdi | Retired | : 6 |
| Birthdate | : 20.04.1976 | Retired Type | : MUNICIPILA |

# Payment Patient

| | | | |
|---|---|---|---|
| Id No | : 4 | Address | : Girne |
| Name | : Ahmet | Gender | : MALE |
| Surname | : Nalband | Pay | : 1000000 |
| Birthdate | : 25.01.1977 | Proto Id | : 20 |

# Input Patient

| | | | |
|---|---|---|---|
| Id No | : 5 | Birthdate | : 10.02.1976 |
| Name | : Bahadır | Payment Type | : Insured |
| Surname | : Soysal | | |

| | | | |
|---|---|---|---|
| Room No | : 25 | Entry Date | : 21.05.2000 |
| Bed No | : 3 | Exit Date | : 25.05.2000 |
| Service | : eye | | |
| Diagnosis | : | | |
| Medicine | : novalgine | | |

# X - Ray

| | | | |
|---|---|---|---|
| Id No | : 6 | Birthdate | : 21.06.1978 |
| Name | : Ahmet | Payment Type | : Payment |
| Surname | : Sayal | | |

| | | | |
|---|---|---|---|
| Type | : ARM | Broken Type | : PIECE |
| Number | : 2 | | |

# Laboratory

| | | | |
|---|---|---|---|
| Id No | : 3 | Birthdate | : 20.04.1976 |
| Name | : Erkan | Payment Type | : Retired |
| Surname | : Hakverdi | | |

| | | | |
|---|---|---|---|
| Blood Sugar | : 12 | Potassium | : 6 |
| Blood Urea | : 3 | Klorur | : 55 |
| Cholesterol | : 2 | Serum Amilaz | : 4 |
| Urik Acid | : 34 | Albumin | : 4 |
| Calcium | : 45 | Globulin | : 78 |
| Kreatinin | : 56 | | |

# Laboratory

| | | | |
|---|---|---|---|
| Id No | : 5 | Birthdate | : 10.02.1976 |
| Name | : Bahadır | Payment Type | : Insured |
| Surname | : Soysal | | |

| | | | |
|---|---|---|---|
| Color | : yellow | Reaksiyon | : 65 |
| Quantity | : 32 | Urabulin | : 5 |
| Aseton | : 3 | Bilirubin | : 33 |
| Sediment | : 34 | Albumin | : 9 |
| Sugar | : 45 | Microskopy | : 6 |
| Miabilinojen | : 78 | | |