NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

PHARMACY AUTOMATION PROGRAM

Graduation Project
COM-400

Pelin Özdemir

Prof.Dr. Rahib Abiyev

Nicosia-2008

ACKNOWLEDGEMENTS

First of all I would like to thank my supervisor Assoc. Prof .Dr . Rahib Abiyev for his continuous support and assistance thoughout the course of the project.

Second, I would like to express my gratitude to Near East University for the scholarship that made the work possible.

Finally, Overcome the difficulties that I faced during various stages of the preparation of this project .I would like to thank my family,H.Barış Çiçek, Haldun Özyol and Ural Yanal who helped me.Their love ,guidance and their never ending belief in me and their encouragement has played a crucial me in my success.

ABSTRACT

As the information age has effected every aspect of our life,the need for computerizing many information systems has raised.

Programming is always providing the scientists a continuous systematic development in their studies and research.With this project is a special program related to Pharmacy Automation.The pharmaceutical industry should not be regarded as an isolated and unrelated field from the other industries but it is within this framework that the history of pharmaceutical developments should be examined new concepts in pharmaceutical design have been eveloped more recently in an effort to meet the changing preferences and new characteristics.

The pharmacy Automation Program consist of many departments like selling, customers,purchasing.Program that has been given in this project.

Before arriving at this point,this project has gone through important phases:

-First one was the requirements definition for which I had to investigate the systematic working principles at a pharmacy.

-Second step was designing the system and software intended to serve for the purpose of pharmaceutical automation.

-Final step was the computer-based implementation of the design using Delphi language.

Within the scope of this project I solved following problems;

-medicine registration for saved data and stock control,

-warehouse registration for purchasing medicine.

-Foundation registration for selling medicine.

-Invoices search within prescriptions.

TABLE OF CONTENTS

LIST OF TABLE

LIST OF FIGURES

INTRODUCTION

As a Pharmacy program is necessary for all pharmacies, in the project it was aimed to write a program considering the problems that we were faced till today in pharmacies.The main structure of the program was designed to apply to the medicine stock control and sales control.The program is user friendly and very simply adapted to different stock programs with simple changes.Using the enormous advantages of Delphi program gives the change to update this code in future due to pharmacy needs.In the following chapters the main structures and menus of the program are explained and finally the source code of the program is presented.

In chapter one, I summarize to development environment of delphi shortly.

In chapter two I brief to database design with paradox.How to create database and how to work?

In chapter three, I would like pripare to follow charts of pharmacy automation program.

In chapter four, I summarize to development of program modules of pharmacy manager system.

CHAPTER ONE

Integrated Development Environment of DELPHI

1.1.What Is Delphi?

By now you know that Delphi is Borland's best-selling rapid application development(RAD) product for writing Windows Applications,With Delphi ,you can write Windows programs more quickly an more easily than was ever possible before.You can create Win 32 console applications or Win32 graphical user interface(GUI) programs.When creating Win32 GUI applications with delphi,you have all the power of true compiled programming language (Object Pascal) wrapped up in a RAD environment.What this means is that you can create the user interface to a program (the user interface means the menus,dialog boxes,main window,and so on)using drag-and-drop techniques for the true rapid application development.You can also drop ActiveX controls on forms to create specialized programs such as Web browsers in a metter of minutes.Delphi gives you all this,and virtually no cost:you don't sacrifice program execution speed because Delphi generates fast compiled code.

1.2.Look at the Delphi IDE

This section contains a Delphi integrated development environment(IDE).You will get the IDE a once-over noew and examine it in more detail on day,''The Delphi IDE Explored''.Because you are trackling Windows Programming ,I'll assume you are advanced enough to have figured out how to start Delphi.Whwn you first start the program,you are presented with both a blank form and the IDE ,as shown in figure 1.1.

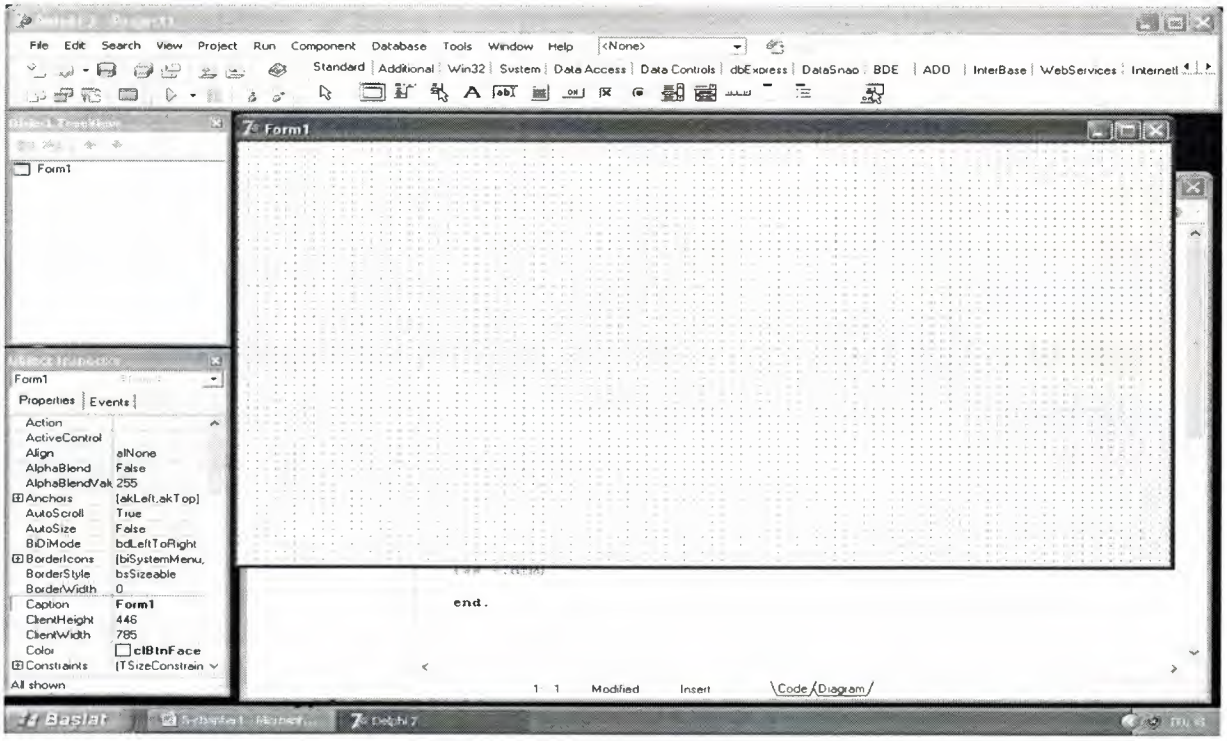

Figure 1.1 The IDE and the initial blank form

Example easy a program ''HELLO'' in figure 1.2.

Figure 1.2 example of application in delphi

Change caption and bold in object inspector with 'HELLO'.

1.3.VCL Components of Delphi

VCL means visual component library.You'll see used various component in my project.

1.3.1.BitBtn Component

The Bitbtn component is a perfect example of how a component can be extended to provide additional functionally.In this case ,the standart button component is extended to enable a bitmap to be displayed on face of button.

1.3.2.Label Component

The label component is used to display text on a form.Sometimes the label text is determined at designe time and never change.In order case is the label is dynamic and is change at runtime as the program dictatese.Use label's caption property to set the label at runtime.The labelk componenent ha sno specialized methods or event beyond what is available with other components.

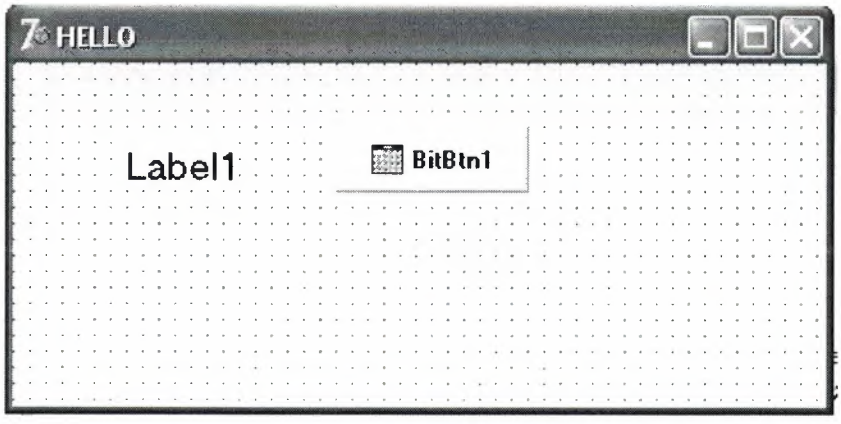

Figure 1.3 Label and BitBtn components

1.3.3.Edit Component

The edit component encapsulates the basic single line edit control.This component has no aligne or aligment property.It has no aligment propert because the text in a single line edit control can only be left justified.The edit component has no align property because it can not (or more accurately,should not) be expanded to fill the client are of window.

1.3.4.Image Component

The image control can be used to display a graphical image-icon(ICO)-Bitmap(BMP),Metafile(WMF),GIF,JPEG,etc.The picture property specifies the image that appears on the image control.There are many ways to assign for the Tımage component;a Tpicture's method load from file can be used to read graphics from disk or assign method can be used to get the image from clipboard,for example.

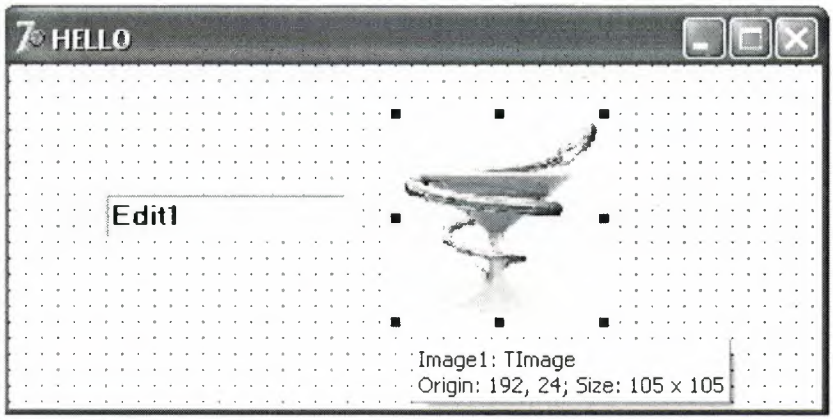

Figure 1.4 Edit and Image components

1.3.5 Date Time Picker Component

The DateTimePicker control allows users to display and select a single date ,using a minimum amount of space.It displays a full month calendar only when the user clicks the control.

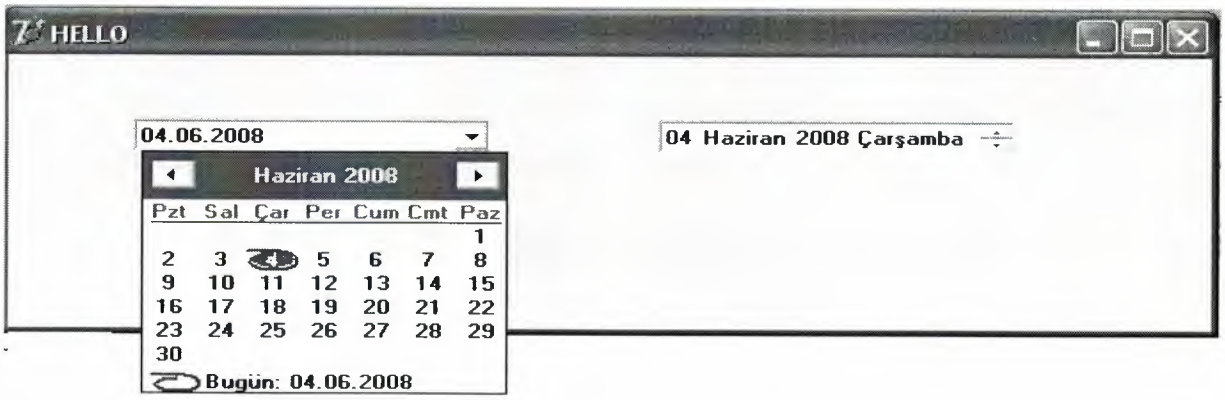

Figure 1.5 date modes of DateTimePicker

1.3.6.Database Component

The VCL database components fall into two categories:nonvisual data Access components and visual data-aware components.Simply put,the nonvisual data Access components provide the mechanism that enables you go get at the data ,and the visual data aware components enable you to view and edit the data.The data access components are derived from the Tdataset class and include Ttable,Tquery and Tstoreproc.The visual data aware components include TDBEdit,TDBListbox,TDBGrid and TDBNavigator, and more.These components work much like the standarts edit ,listbox and grid components expect that they are tied to particular table or field in a table.By edit in gone of the data-aware components,you are actually editing the underlying database as well.

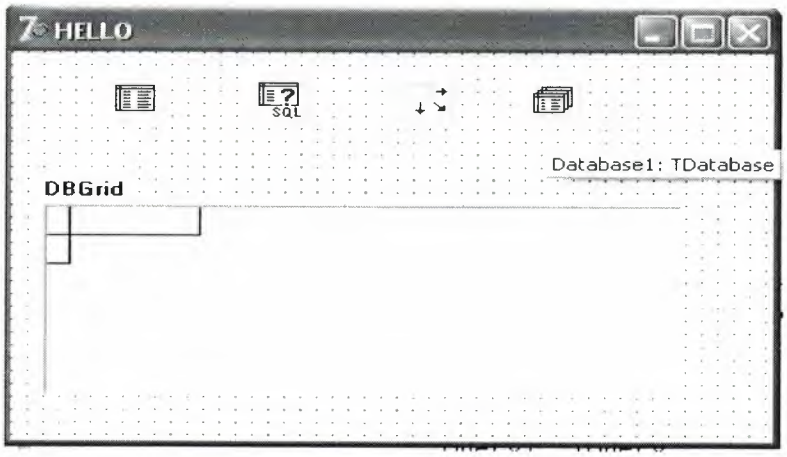

Figure 1.6 Database Component

1.3.7.Query Component

The BDE API enables the client to use SQL or Query by example (QBE) to access Dbase,FoxPro,Access and Paradox tables(standard databases)as well as server-based SQL tables.

A group of BDE query interface functions is provided for passing either SQL Queries or QBE queries to both server-based and PC-based sources.

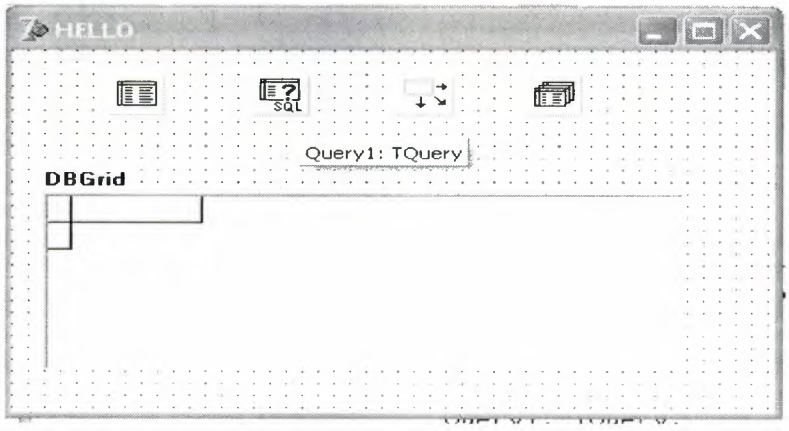

Figure 1.7 Query component

1.3.8.DataSource Component

The Datasource component provides a mechanism to hook dataset components(table,query or storeproc.) to the visual components that the display the data (Dbgrid,Dbedit,Dlistbox and so on).The primary purpose of Datasource is to enable making changes to your application easier.All the data components on a form are hooked up to the Datasource ,which is then hooked up to the dataset.

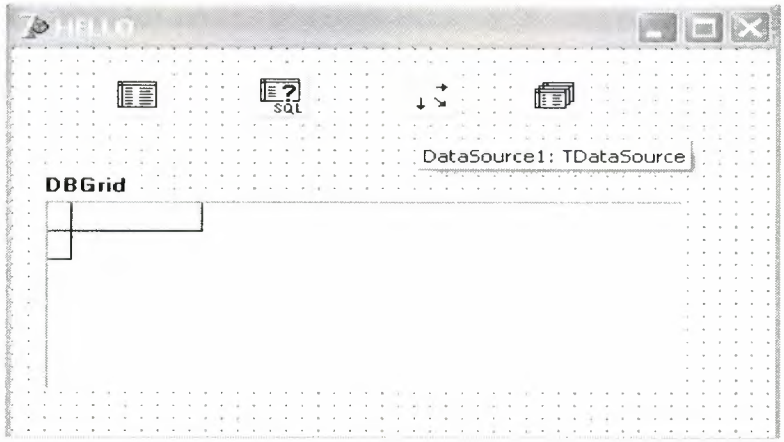

Figure 1.8 Data Source component

1.3.9.Dbgrid Component

The Dbgrid component displays a dataset in tabular,or spreadsheet,format.One of the most important properties of the Dbgrid is columns property.This property enables you to change the number and order of the columns that appear in the grid.You can add ,remove and order columns using the columns editor.

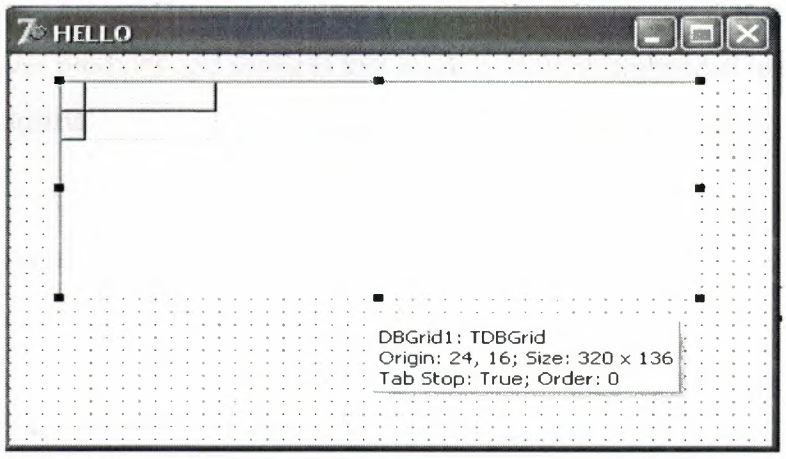

Figure 1.9 DBGrid component

CHAPTER TWO

DATABASE DESIGN USING BY PARADOX

2.1.Why is computer necessary in our life

Computer software has become a driving foce;it is a powerful that set Desicion-making and serves as a basis for moderninvestigation and problem solving.Computer have become a key factor that gives products and services that modern look,its embedded in systems of all kinds;medical,industrial,military,entertainment,even offive-based products.

A computer system in a service management record can promise better speed and efficiency with almost no change of effors.

2.2.How to develop a database application

The steps involved in database application development any relational data base application there are always the same basics steps to follow.Paradox is relational data base management system ,all data is stored in paradox database in the form of simple tables.Another name for a table as relation.

The steps of Paradox database design like this:

-Database design

-Tables design

-Forms design

-Query design

2.3.Relational Database

DBMS (DataBase Management System)has established themselves as one of the primary means for data storage for information based systems raging from large business applications to simple pc based programs.However a relational database management system (RDBMS) is the system used to work with data management operations more than 15 years ,and still improving,providing more sophisticated storage ,retrival systems.Relational database management systems provides organisations with ability to handle huge amount of data and changing it into meaningful information.

2.4.Accessing Database From Delphi

Create the Database and tables within it ,ensure BDE is installed on system and registry settings point to its location.

For local file-based databases using native links:

-Paradox,dbase tables are accessed via the STANDARD driver.

-Ascii tables are accessed via the ASCII driver.

-Access files are accessed via access driver which then accesses the JET engine which needs to be present too.

2.5.Creation of Database with Paradox

Paradox databases: a database is just a Window's directory,and a BDE alias is pointer to that directory.

Other databases :cannot create databases within Delphi&thuss usually require the specific server's utility to create databases.

2.6.The Facilities of Paradox

Paradox is relational DBMS(DataBase Management System) with all the features neccessary to develop and used a database application.The facilities it offers can be found on most modern relational DBMS and Paradox.

-Tables are where all the data is stored.

-Queries are the way you extract data from the database.

-Forms are the method used for input and display of database data.

-Reports are used to display nicely formated data on paper.

2.7.BDE(The Borland Database Engine)

The Borland Database Engine (BDE) includes an API for directly using its functionality.The API consists of a set of functions that can be called from any programming language capable of loading windows DLLs and using functions contained in them.BDE functions are optimized for calling from C or C++;however,DELPHI Pascal syntax is also provided in the function reference.

Over the years ,two different types of database systems have developed that traditionally supported different data access approaches:

-PC-based database systems (such as Paradox,dBASE,and B-Trieve)have supported the indexed sequential access method (ISAM)type of data access.However ,these systems have supported different kinds of APIs .

-Server-based database systems(such as interbase,sybase,oracle and DB2)have supported the ANSI standard SQL language.However ,an industry standard for an API is just emerging:X/open SQL call level interface(CLI).This standard addresses only SQL-based database needs ,and does not fully address ISAM type data search requirements.

2.7.1.DataBase Drivers

Each driver is implicitly loaded by the system when an application first requests a service from that driver.At that time ,any configurable setting found in the Windows Registry or the Borland Database Engine (BDE) configuration file (IDAPLCFG) related to this driver are used to initialize it.Examples of configurable settings are the default table level and the language driver to be used when the table is created.

Drivers are owned by the client or the system; once a driver is loaded, all other clients registered with BDE have access to it.

The application developer can also inquire about driver capabilities, such as whether or not the driver support transactions.

Dbase,Pradox,Access,FoxPro,and text drivers

The standart drivers for Paradox,Dbase, Access, FoxPro, and text database are shipped with BDE.

SQL drivers

For server-based SQL database system such as Informix,DB2, InterBase, Oracle, and Sybase separate native BDE SQL drivers are avaliable.

ODBC drivers

Any ODBC driver can be used with BDE,because BDE has an ODBC connectivity socket.The rich features of BDE, such as navigational access to data, bi-directional cursors, and cross-database operations, are also automatically enabled even when an ODBC driver is in use.Enhanced ODBC connectivity.BDE functions like DbiAddAlias and DbiOpenDatabase automatically add ODBC drivers and data sources as BDE aliases to the active session when they aren't currently stored in the configuration file.The BDE also support ODBC 3 drivers.

2.7.2.DAO(data access objects)

The DAO approach to database programming often requires more code, but like SQL compared to the Qury Design View, offers greater control to the progremmer over what's going on his/her application.

Data Access Object are things like databases, recordsets, table and query definitions, and fields, Rather than tying a recor set to a data control when we use DAO we shall allow our programs to create and manipulate recordsets

2.7.3.ADO(activeX data objects)

The ADO programming is in principle very similar to DAO programming but cointains some new commands.ADO is Microsoft's new approach to database programming which aims to give the programmer a more consistent way of conncting to a broad range of different types of data source.

2.8.The Application of PARADOX

Paradox provides database power to give you the information you need to make better decision and manage your business.

With my programming experience and the knowledge you have of your business operation ,you are guaranteed an extremely powerful and user friendly application.

Dbclick Borland Delphi 7→Data Base Desktop.

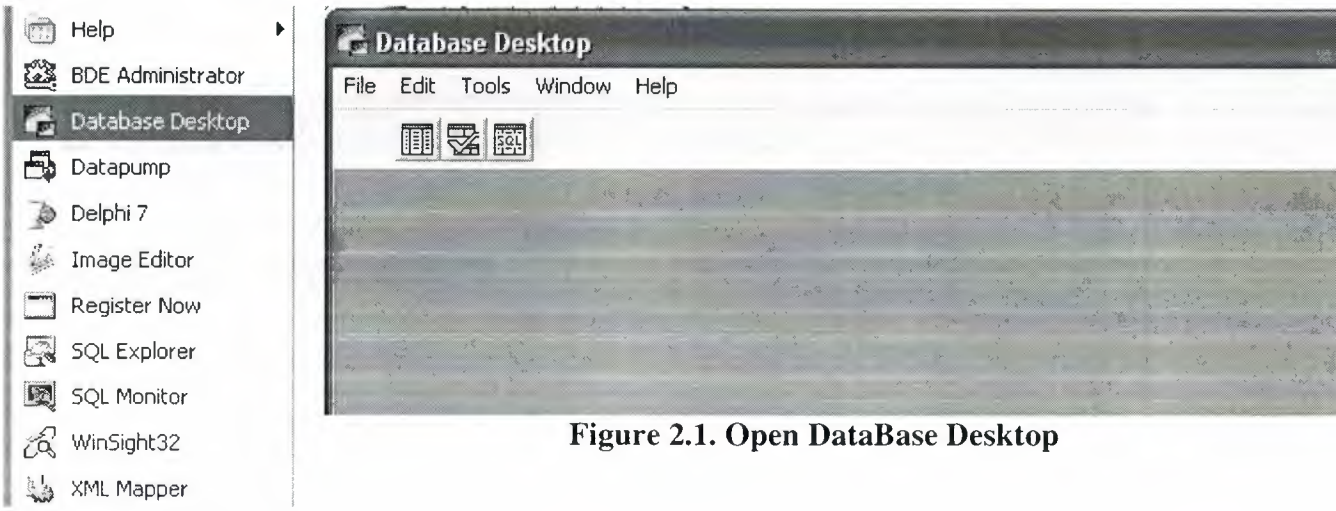

Figure 2.1. Open DataBase Desktop

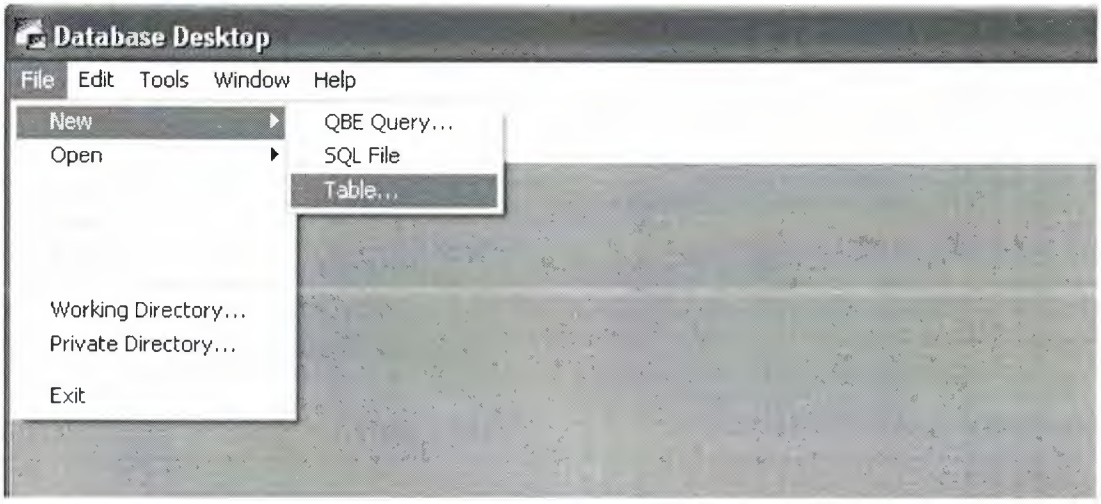

Figure 2.2. Window of File menu

Open file then choose new table.It's for creating new database table.

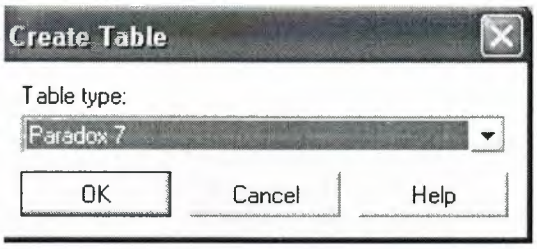

Figure 2.3. Table Type

Choosing table type 'PARADOX 7' for our new table.

And also,I choose 'dBASE for Windows 'type for some of my tables.

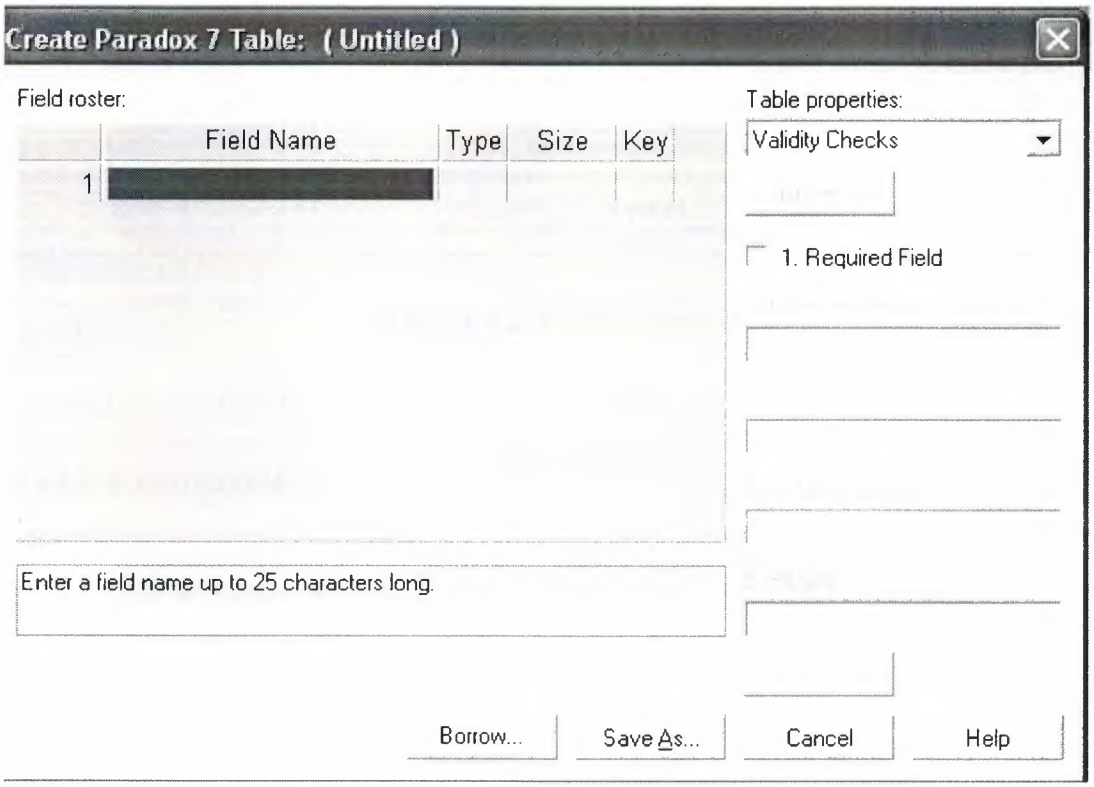

Figure 2.4.Create Paradox 7 table

Create paradox 7 table,then enter values:

-field name ;name of columns for our table

-type;types of data,such as Alpha,Number,Short,Date,etc.

-size; sizes of data

-Key;key fields(must be top of fields in the field roaster)

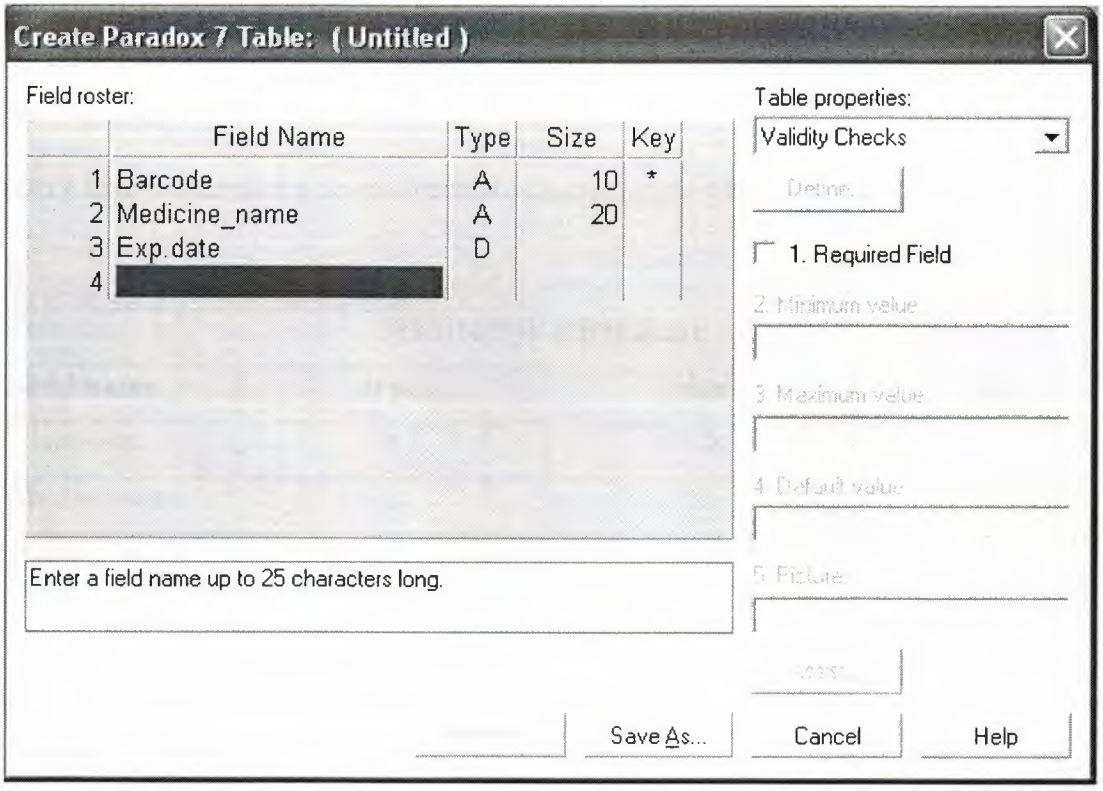

Figure 2.5.Enter values

Enter values,then click 'save As'.Then gives it file name.

Finally we can see our table;

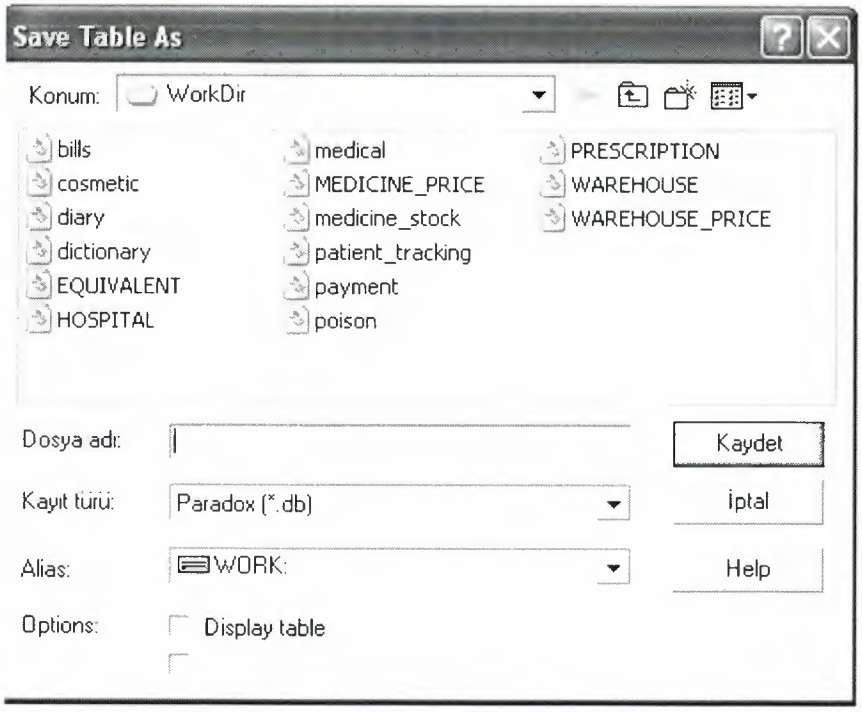

Figure 2.6.All Tables

2.9.Database Structure

Program's database includes seventeen tables.Some tables given below.

MEDICINE-PRICE.DB			
Field name	**type**	**size**	**key**
barcode	A	10	*
Medicine_name	A	20	
İncluding_VAT	N		
Excluding_VAT	N		
profit	N		
Profit_rate	S		
İncluding_VAT-selling	N		

Table 2.1. Medicine Prices table

MEDICINE_STOCK.DB			
Field name	**type**	**size**	**key**
barcode	A	10	*
Medicine_name	A	20	
Stock_on_hand	S		
Shelf_quantity	S		
Start_date	D		
Exp_date	D		

Table 2.2. Stocks of Medicine table

PATIENT_TRACKING.DB			
Field name	**Type**	**Size**	**Key**
Name_surname	A	20	*
Phone_number	A	15	
Medicine_name	A	20	
Treatment_start_date	D		
Treatment_end_date	D		
condition	A	5	

Table 2.3. Patient Tracking Table

EQUIVALENT.DB			
Field name	**Type**	**Size**	**Key**
Barcode	A	10	*
Medicine_name	A	20	
Active-substance	A	20	
Equivalent_medicine	A	20	
Ingredients	A	240	
Usage	A	50	
Side_effect	A	240	

Table 2.4. Equivalent of Medicines table

WAREHOUSE.DB			
Field name	**Type**	**Size**	**Key**
No	S		*
Warehouse_name	A	20	
Contact_person	A	20	
Phone	A	15	
Fax	A	15	
Address	A	25	
Web	A	20	

Table 2.5.Warehouse Table

BILLS.DB			
Field name	**type**	**size**	**key**
Prescription_no	S		*
Name_surname	A	20	
Hospital	A	25	
Medicine_name	A	20	
Unit	S		
Price	N		
Date	D		

Table 2.6. Bills table

MEDICAL.DB			
Field name	**Type**	**Size**	**Key**
Barcode	A	10	*
Medical_material	A	20	
Start_date	D		
Exp_date	D		
Stock_on_hand	S		
Shelf_quantity	S		

Table 2.7.Stocks of Medical Materials Table

HOSPITAL.DB			
Field name	**type**	**size**	**Key**
Hospital_name	A	25	*
Address	A	20	
Phone	A	15	
Web	A	20	

Table 2.8.Hospital table

POISON.DB			
Field name	**Type**	**Size**	**Key**
Name	**A**	**20**	
Explanation	**A**	**25**	
Medical	**A**	**25**	

Table 2.9.Poison Table

2.10.Working With SQL

SQL standart of structered Query language.SQL is used to communicate with a database.According to ANSI (American National Standard Institute),it is the standard language for relational database management systems.SQL statements are used to perform tasks such as update data on a database ,or retrieve data from a database.Some common relational database management systems that use SQL are :Oracle ,Sybase,Microsoft SQL Server,Access,Ingress,etc.Altought most database systems use SQL ,most of them also have their own additional proprietary extension that are usually only used on their system.However,the standard SQL commands such as 'Select' ,'Insert','Delete','Create',and 'Drop' can be use to accomplish all most everything that one needs to the with a database.

2.10.1.Table Basics

A relational database system contains one or more objects called tables.The data or information for the database are stored in these tables.Tables are uniquely identified by their names and are comprised of colums and rows.Colums contain the column name,data type,and an other attributes for the column.Rows contain the records or data for the columns.Here is a sample table called 'weather'.

City, state, high and low are the columns.The rows contain the data for this tables;

Weather

City state high low

Phoenix Arizona 105 90

Tucson Arizona 101 92

Flagstaff Arizona 88 69

San Diego California 77 60

Albuquerque New Mexico 80 72

2.10.2.Selecting Data

The select statement is used to query the database and retrive selected data that match the criteria that you spacify.Here is the format of a simple select statement:

Select 'column1'[,'column2.etc] from 'tablename'[where condition'];

[]=optional

The column names that follow the select keyword determine which columns will be returned in the results.You can select as many column names that you'd like ,or you can use a '*'to select all columns.

The table name that follows the keyword from specifies the table that will be queried to retrieve the desired results.

The where clause (optinal) spesifies which data values or rows will be returned or displayed,based on the criteria described after the keyword where.

2.10.3.Like

The like pattern matching operator can also be used in the conditional selection of the where clause.Like is very powerful operator that allows you to select only rows that are 'like'what you specify.The percent sign '%' can be used as a wild card to match any possible character that might appear before or after the characters specified.For example:

Select first, last, city from empinfo where first LIKE 'Er%';

This SQL statement will match any first names that start with 'Er'.Strings must be in single quotes or you can specify,

Select first, last from empinfo where last LIKE '%s';

This statement will match any last names that end in a 's'.

This * form empinfo where first='Eric';

This will only select rows where the first name equals 'Eric' exactly.

2.10.4.Updating Records

The update statement is used to update or change records that match a specified criteria.This is accomplished by carefully constructing a where clause.

Update 'tablename' set 'columnname'='newvalue'[,'nextcolumn'='newvalue2'...]where 'columnname'OPERATOR 'value'[and/or'column'OPERATOR'value'];

[]=optinal

Example: update phone _book set area _code=623 where prefix =979;

2.10.5.Deleting Records

The delete statement is used to delete records or rows from the table.

Delete from 'tablename' where 'columnname' OPERATOR 'value'[and/or'column'

[]=optional

To delete an entire record/row from a table , enter 'delete from' followed by the table name,followed by the where clause which contains the conditions to delete.If you leave off the where clause, all records will be deleted.

2.10.6.Drop a Table

Drop a table command is used to delete a table and all rows in the table.The delete an entire table including all of its rows, issue the drop table command followed by table name.Drop table is different from deleting all of the records in the table.Deleting all of the records in the table leaves the table including column and constraint information.Dropping the table removes the table defination as well as all of its rows.

Drop table 'tablename';

Example:Drop table employee;

CHAPTER THREE

PHARMACY AUTOMATION PROGRAM:

FLOW-CHARTS OF PROGRAM MODULES

3.1.Flow-Chart of Main Program

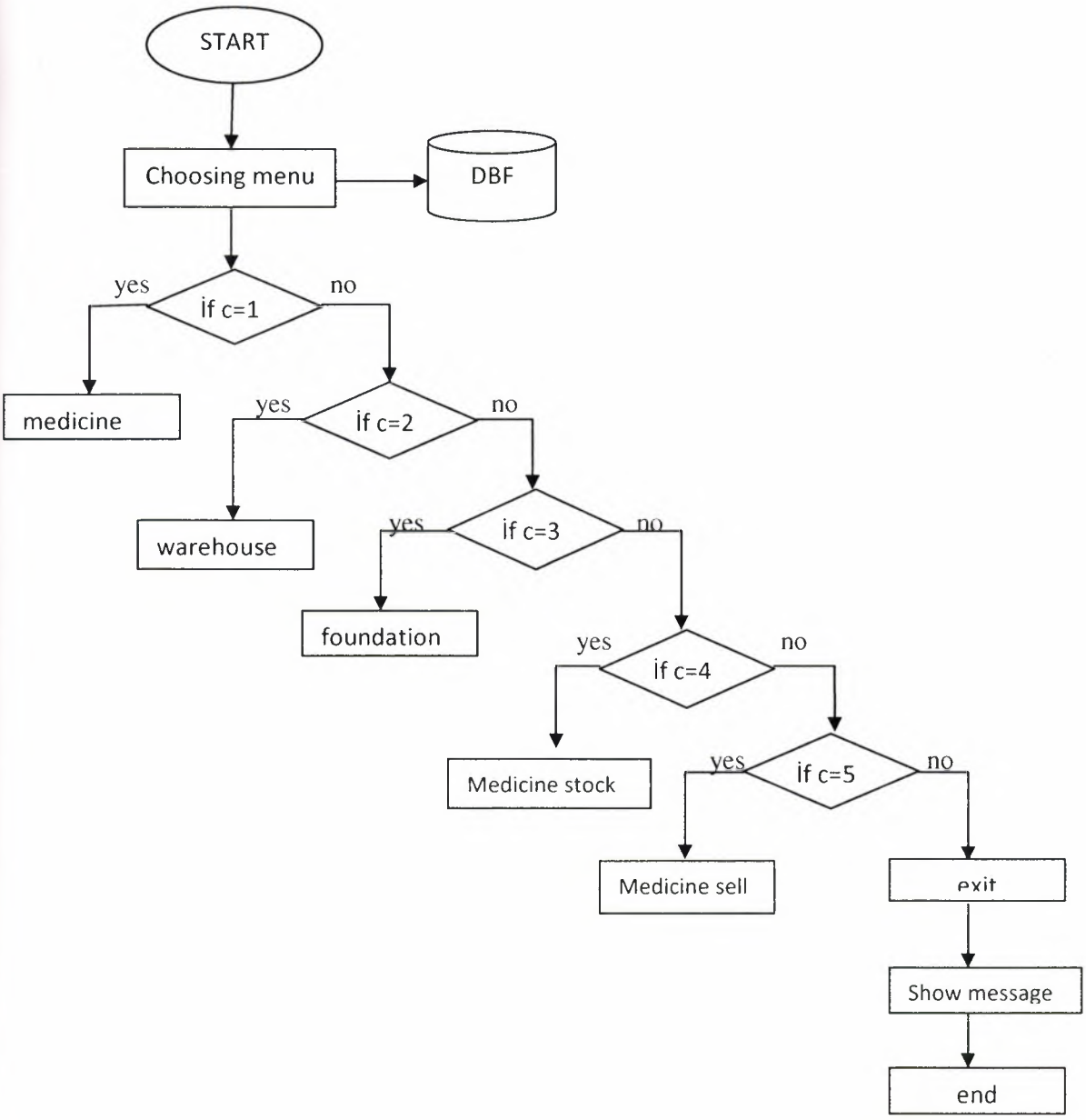

Figure 3.1. Main Menu Flow-Chart

3.2.Flow-Chart of Medicine Registration

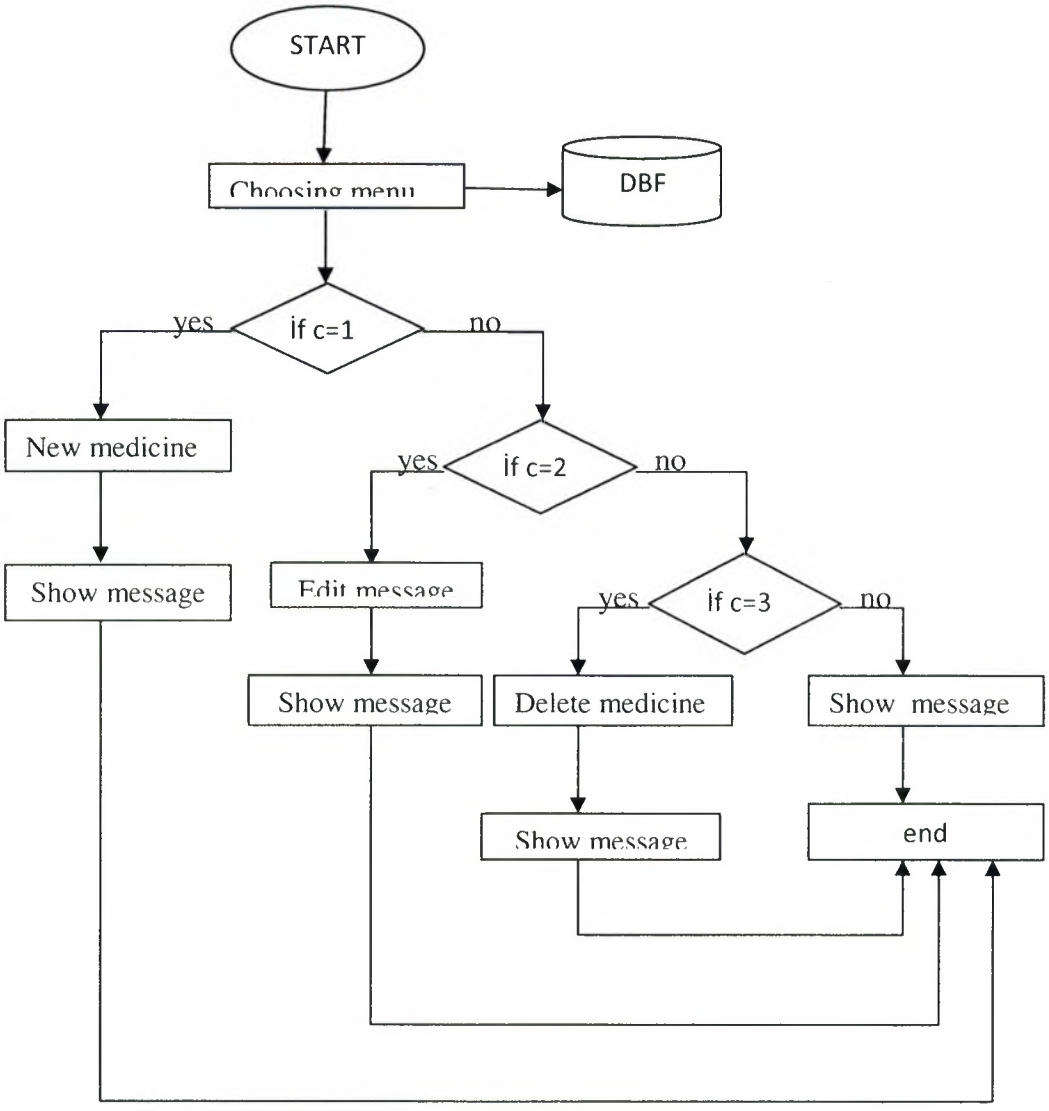

Figure 3.2. Medicine Menu Flow-Chart

3.3.Flow-Chart of Prescription Search

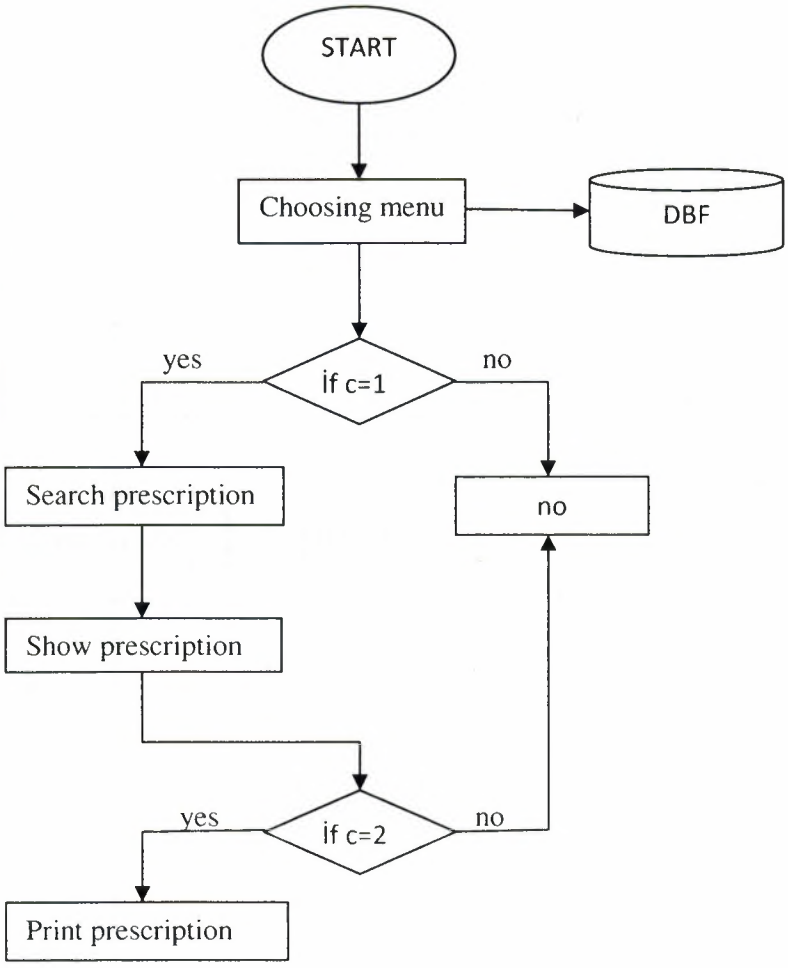

Figure 3.3. Prescription Search Flow-Chart

3.4.Flow –Chart of Warehouse Registration

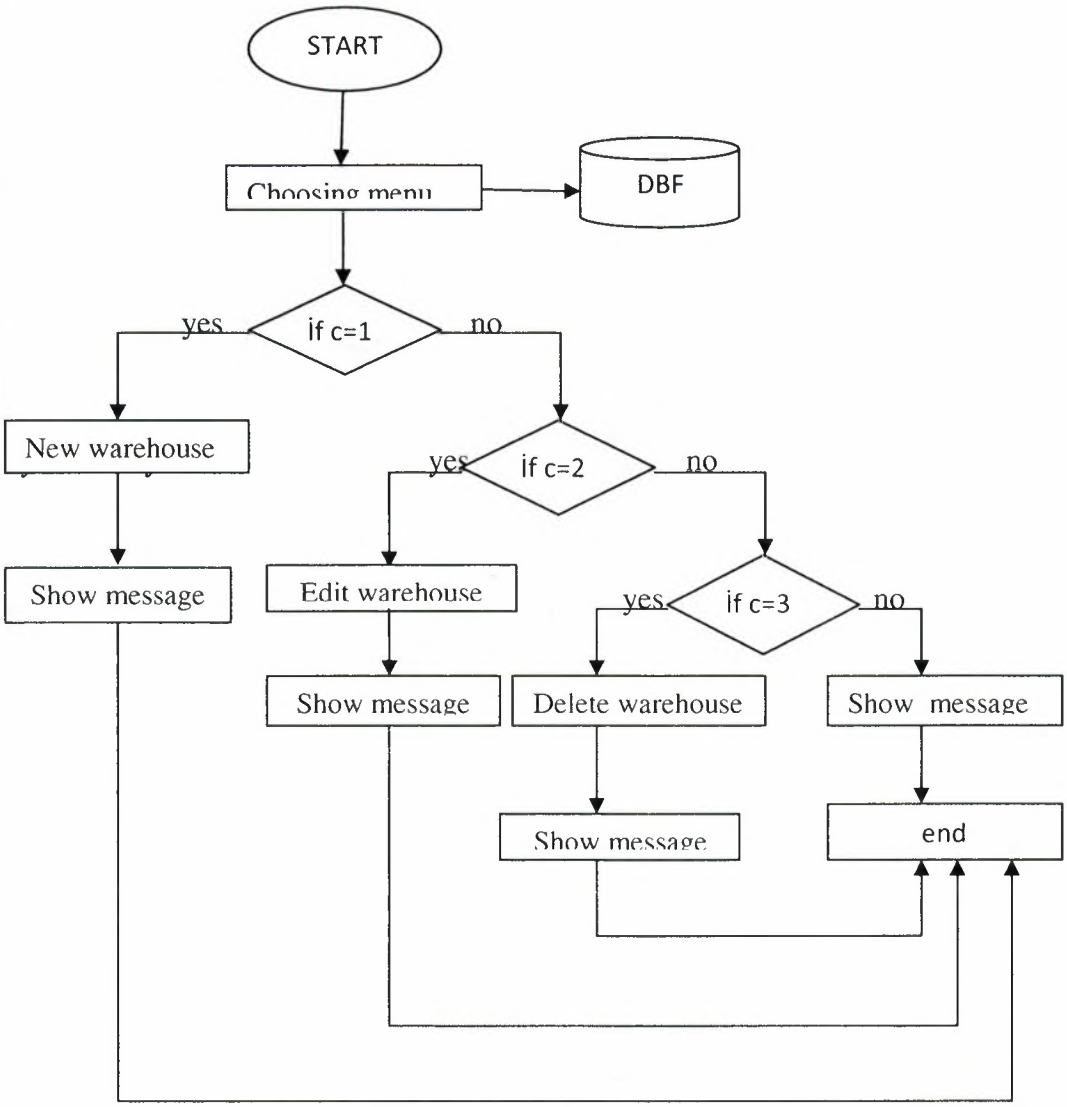

Figure 3.4. Warehouse Registration Flow-Chart

3.5.Flow-Chart of Medicine Selling

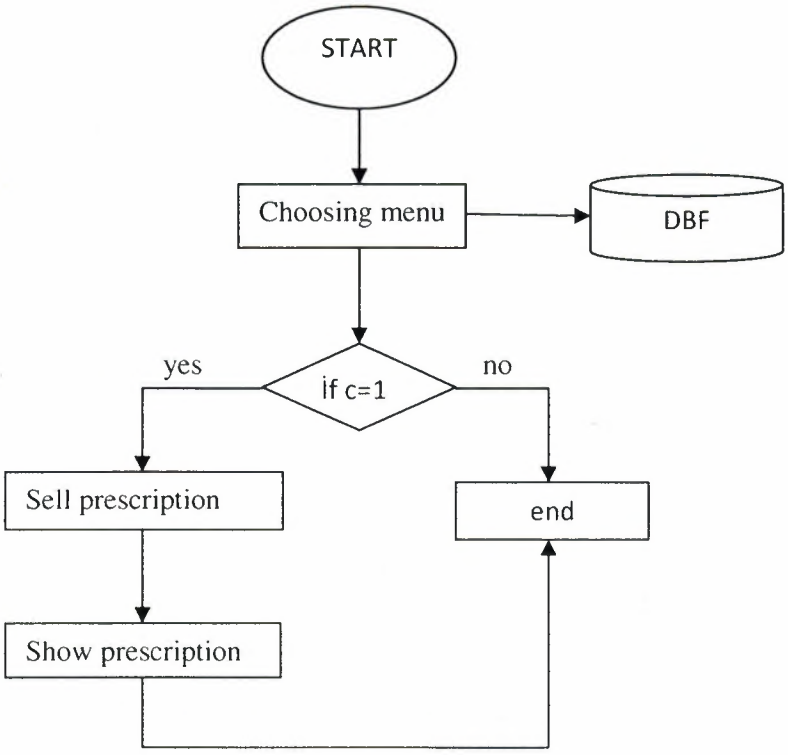

Figure 3.5. Medicine Selling Flow-Chart

CHAPTER FOUR

DEVELOPMENT OF PROGRAM MODULES OF PHARMACY AUTOMATION PROGRAM

4.1.Main Menu Screen

This is the main menu of the program.There is also some sub menus on the top and under of the main menu.From the main menu we can go sub programs by using this sub menus.There are also some buttons.They are used to go to the sub programs.They are providing facilities for users of the program.We can see all sub programs on the main menu.

Medicines button is used to go record part of the program.In the part we enter medicine record information.

Warehouse button is used to record of medicines warehouses.

Foundation button is used for medicine percentage about to foundation.

Stock button is used for enter medicine,medical material and cosmetics stock.

Prescription button used for search prescription.

Hospital button is used for hospitals information.

Customer button is used for customer information and patient tracking.

Bills button is used for date to date searching

Report button is used for financial inventory.

Dictionary and Poison buttons are medical dictionary.

Emergency phones button is used for emergency numbers.

Personal button is used for personal enters such as diary or notes.

Figure 4.1. opening Screen

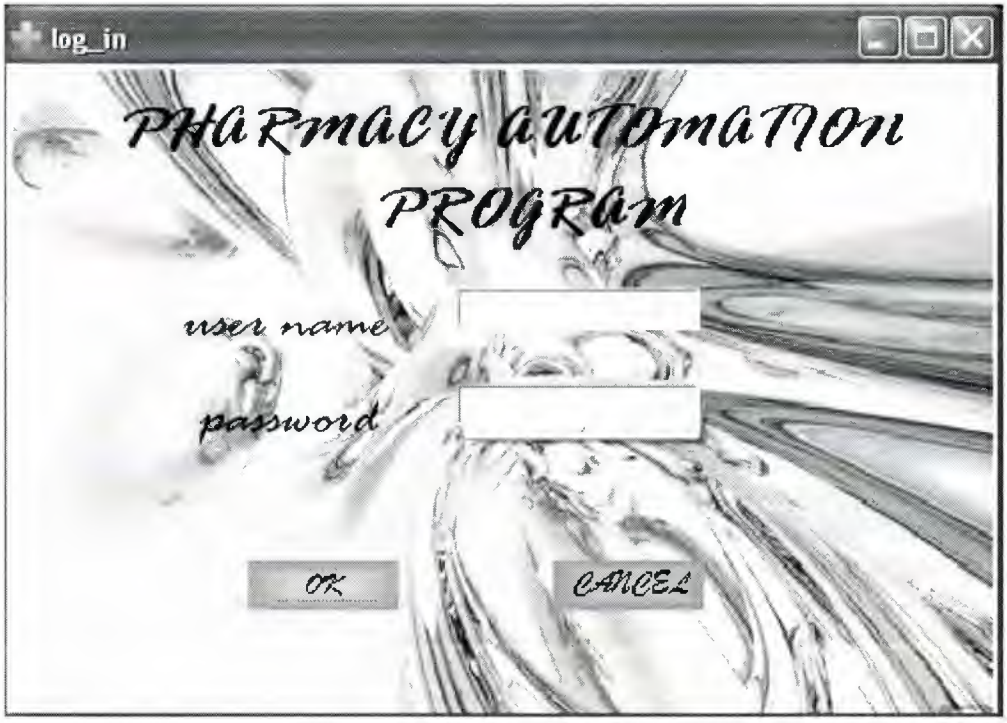

Figure 4.2.log_in screen

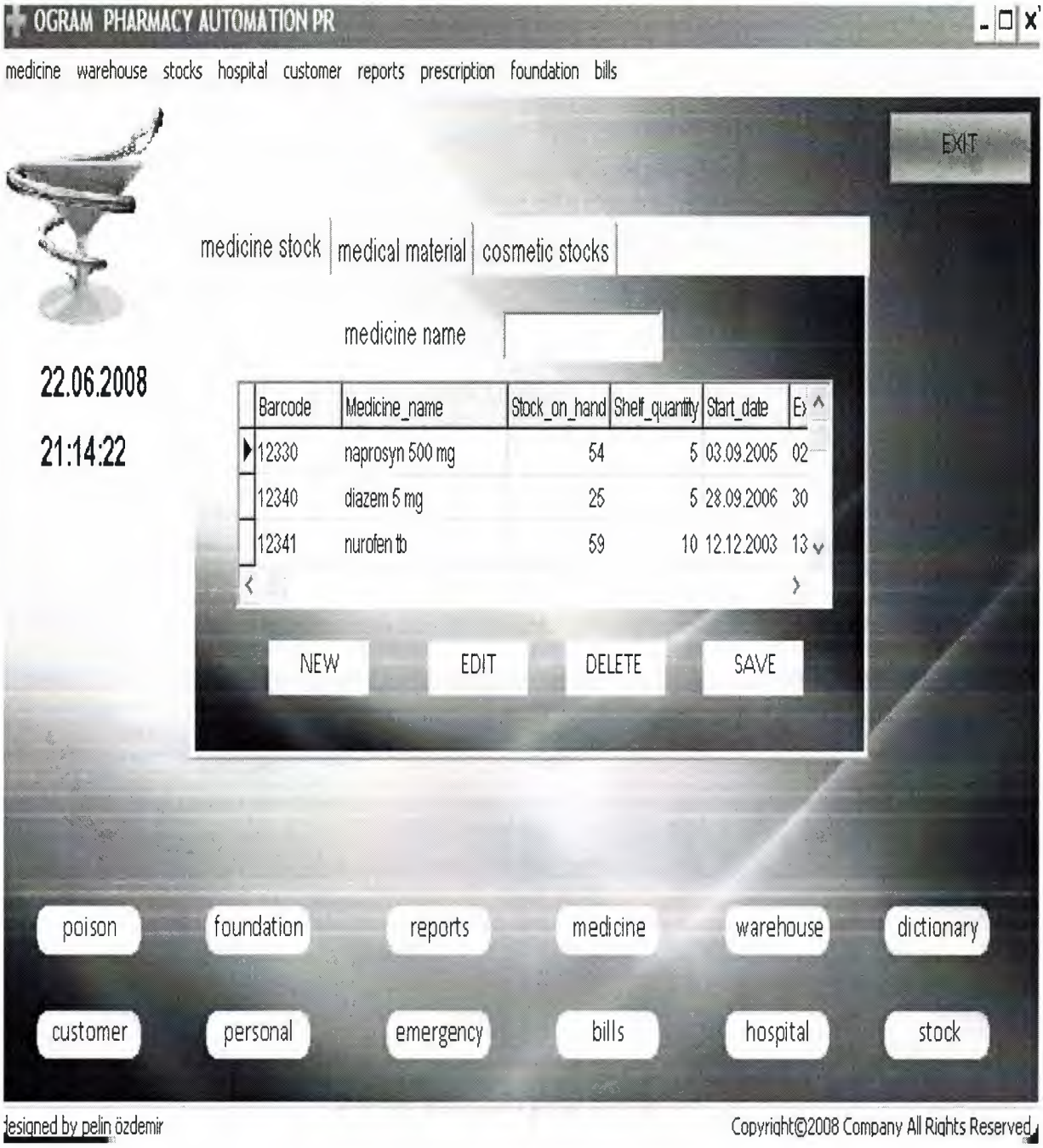

Figure 4.3.main menu

4.2.Medicine Screen

Section of showing everything about medicine,such as prospectus,stocks, equivalent medicine and medicine price.

The type of record are searching,deleting,editing,adding,finding and saving with screen.

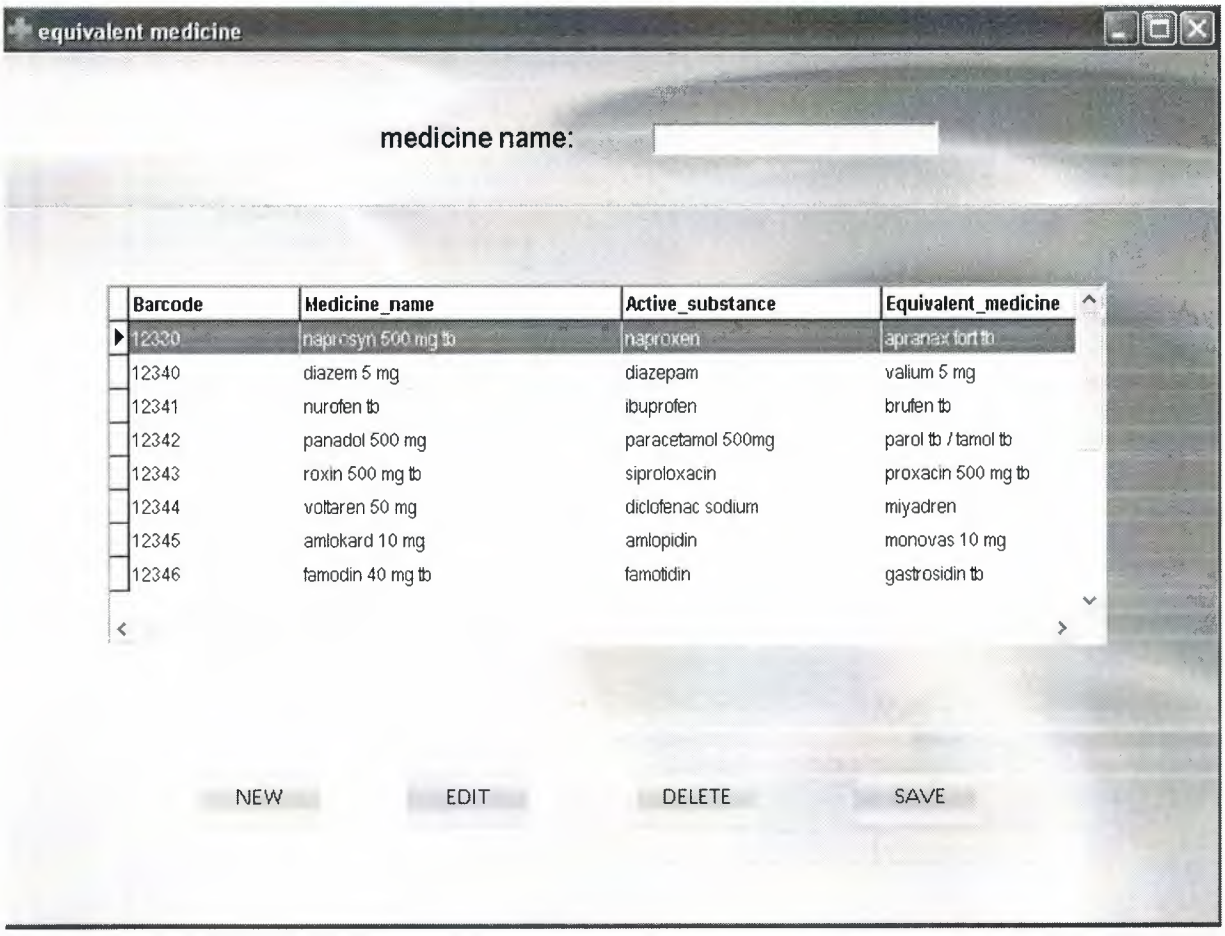

Figure 4.4.equivalent medicine

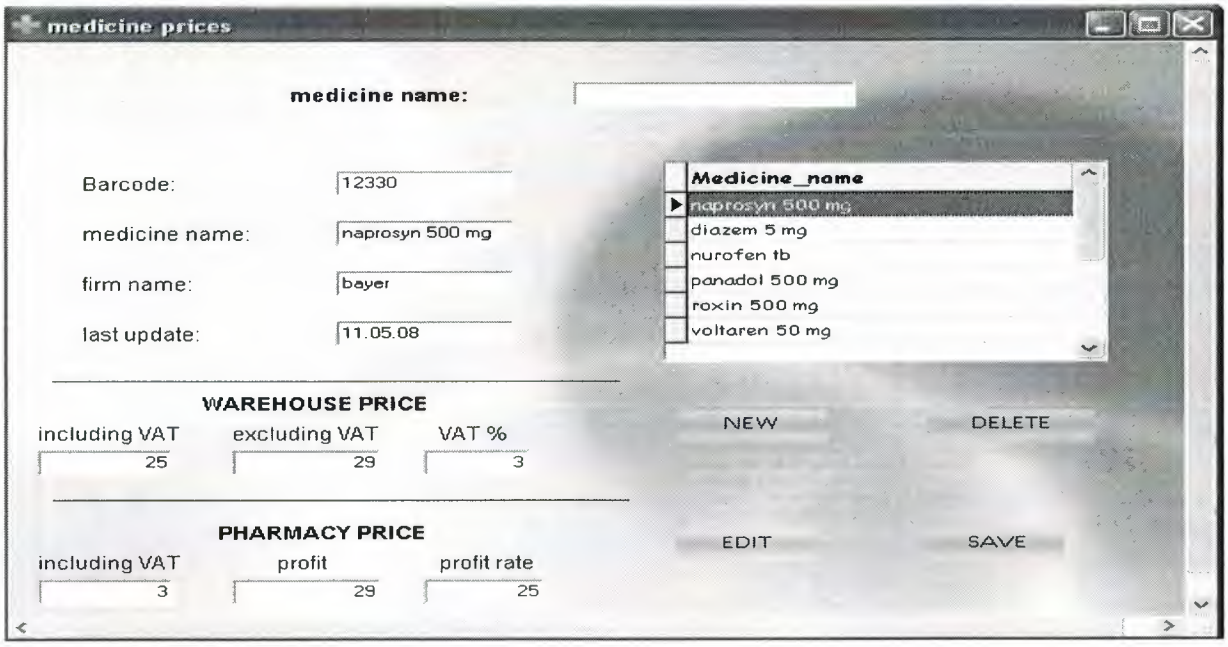

Figure 4.5.medicine prices

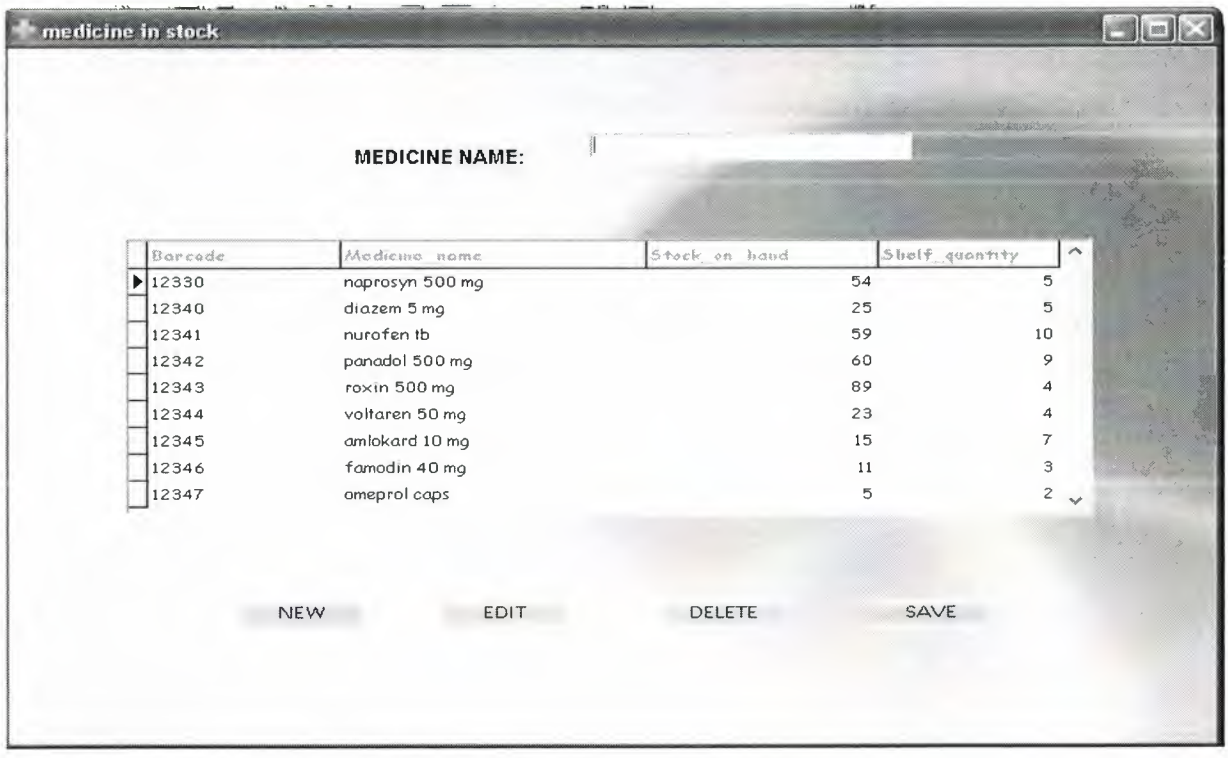

Figure 4.6.medicine in stock

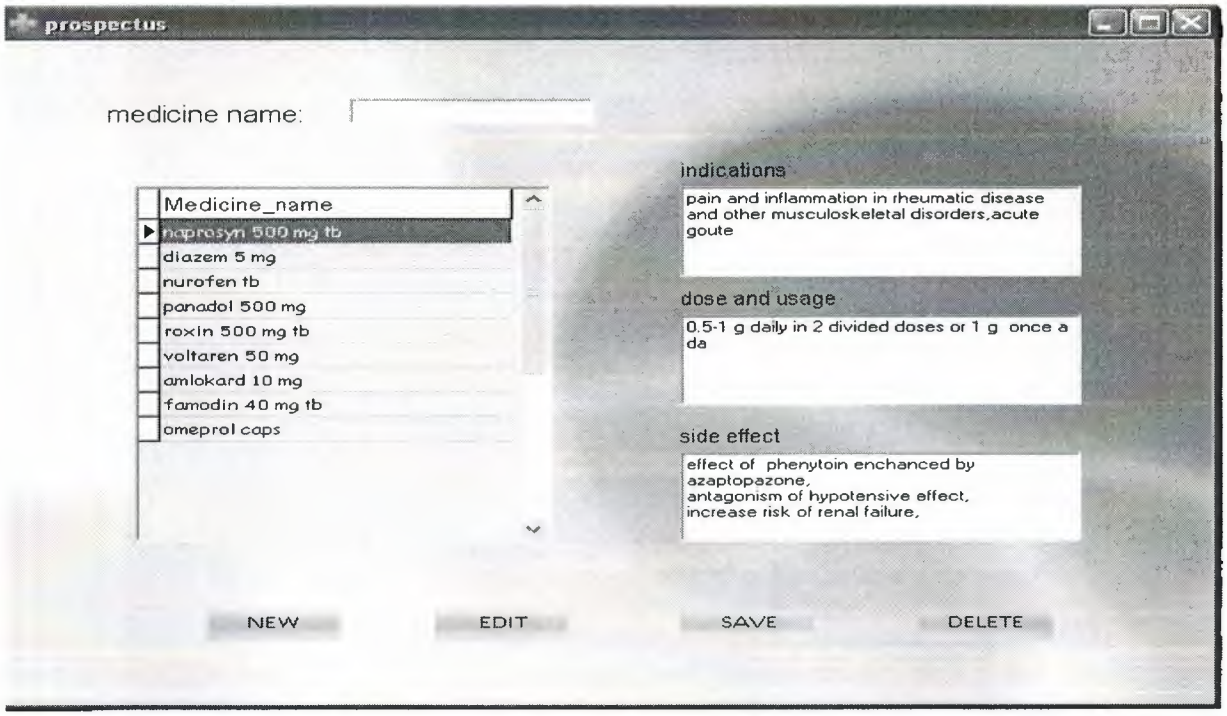

Figure 4.7.prospectus

4.3.Warehouse information

With this form,we can save,edit,add and deleting to the rehouse record.When we need to buy any medicine,we should call to the medicine warehouses.When we but medicine we should save warehouse name for our dept.

Figure 4.8.warehouse information

Figure 4.9.warehouse prices

4.4.Foundation screen

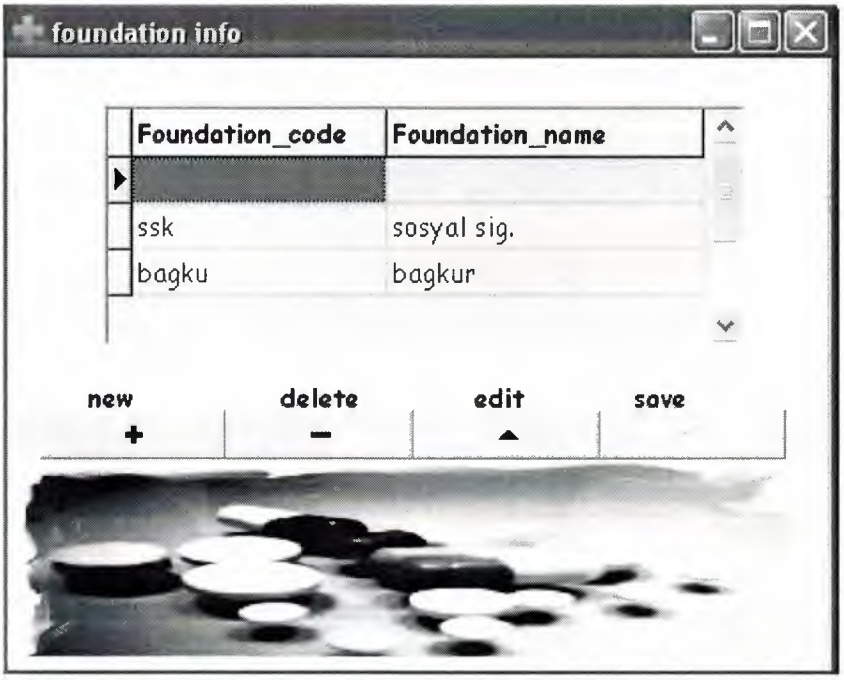

Figure 4.10.foundation information

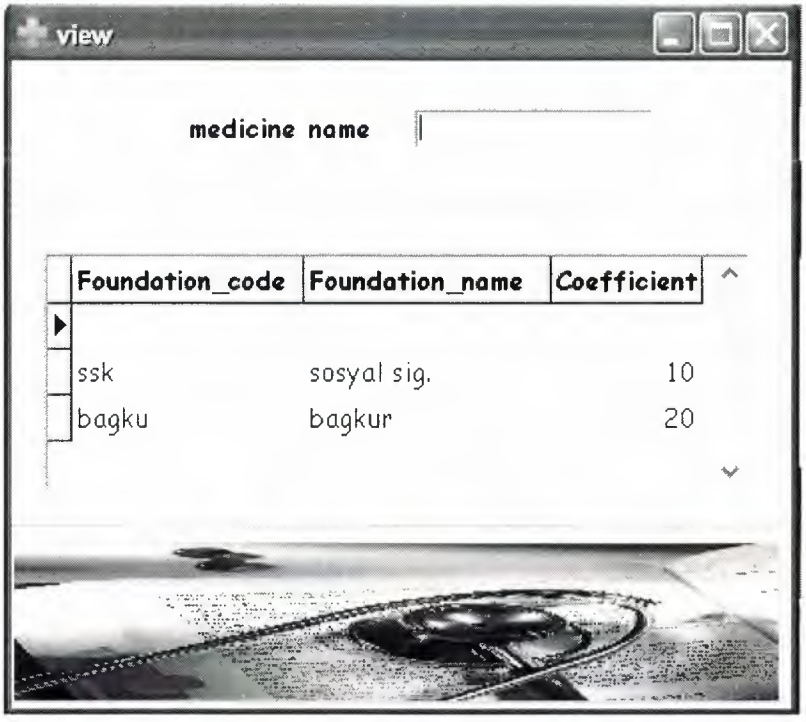

Figure 4.11.view

4.5.Bills screen

The report of all sold medicine by prescription.also we can print prescription which we want from these screen.

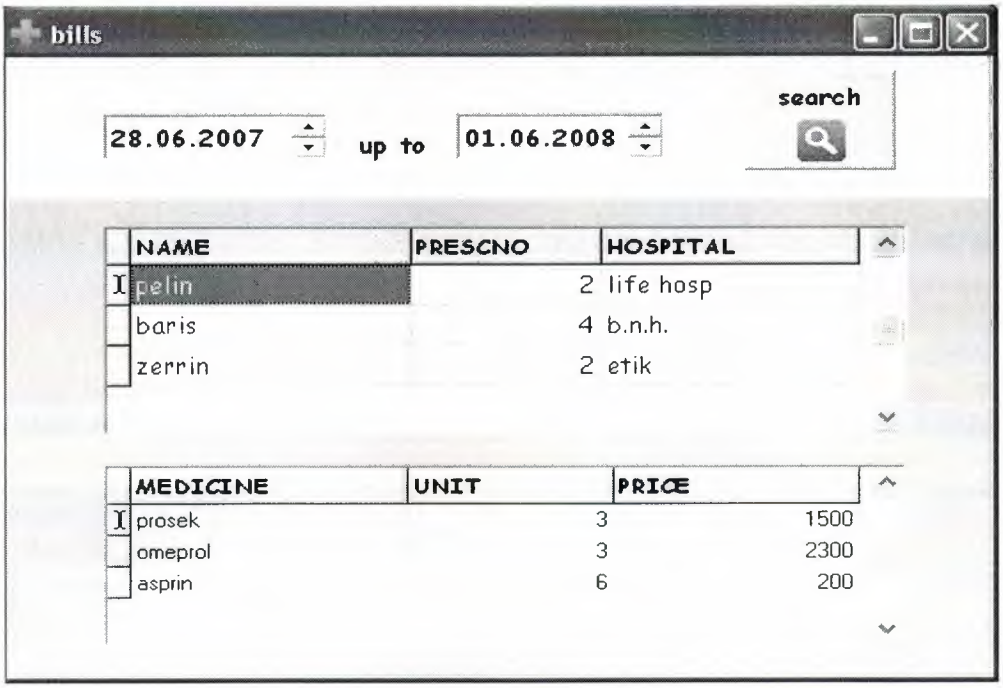

Figure 4.12.bills screen

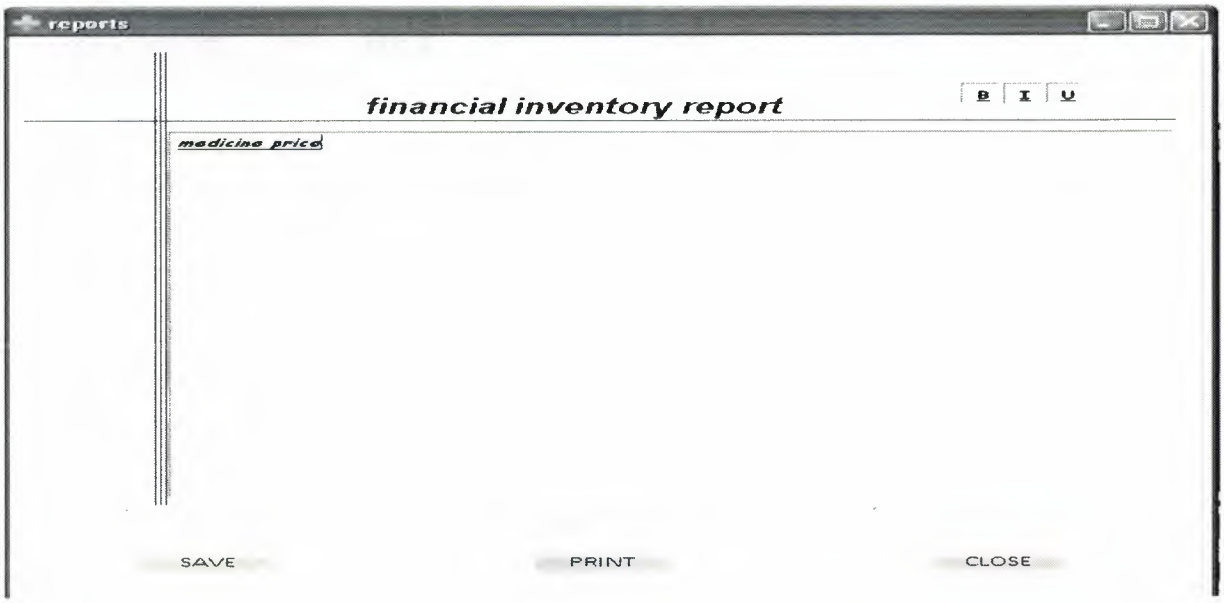

Figure 4.13.report

4.6.Stocks screen

For buying any medicine firstly should we save its record to our database from medicine form.after that we can choose medicine ,medicine stock border,buyed unit,buyed place ,buyed date and buying and selling price.Also we can see medicine how many we have.And also medical material and cosmetics stock.

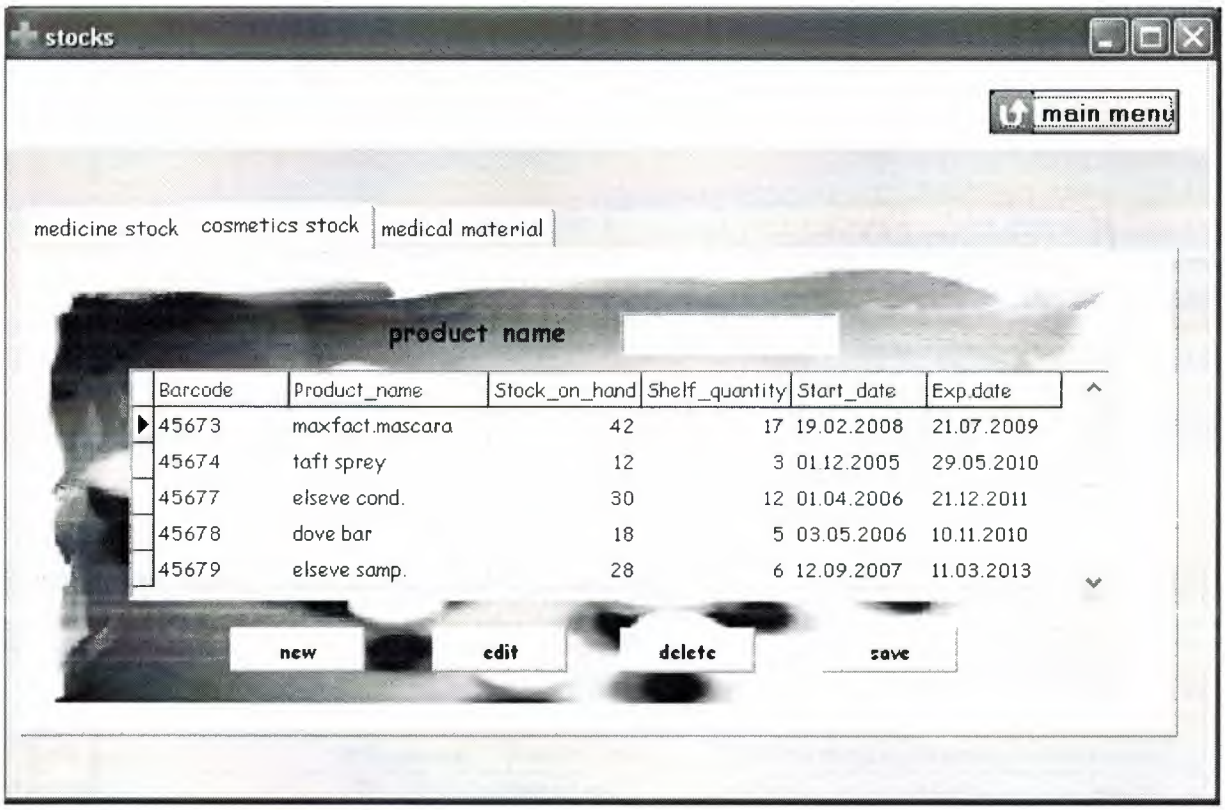

Figure 4.14.stocks screen

4.7.Customer screen

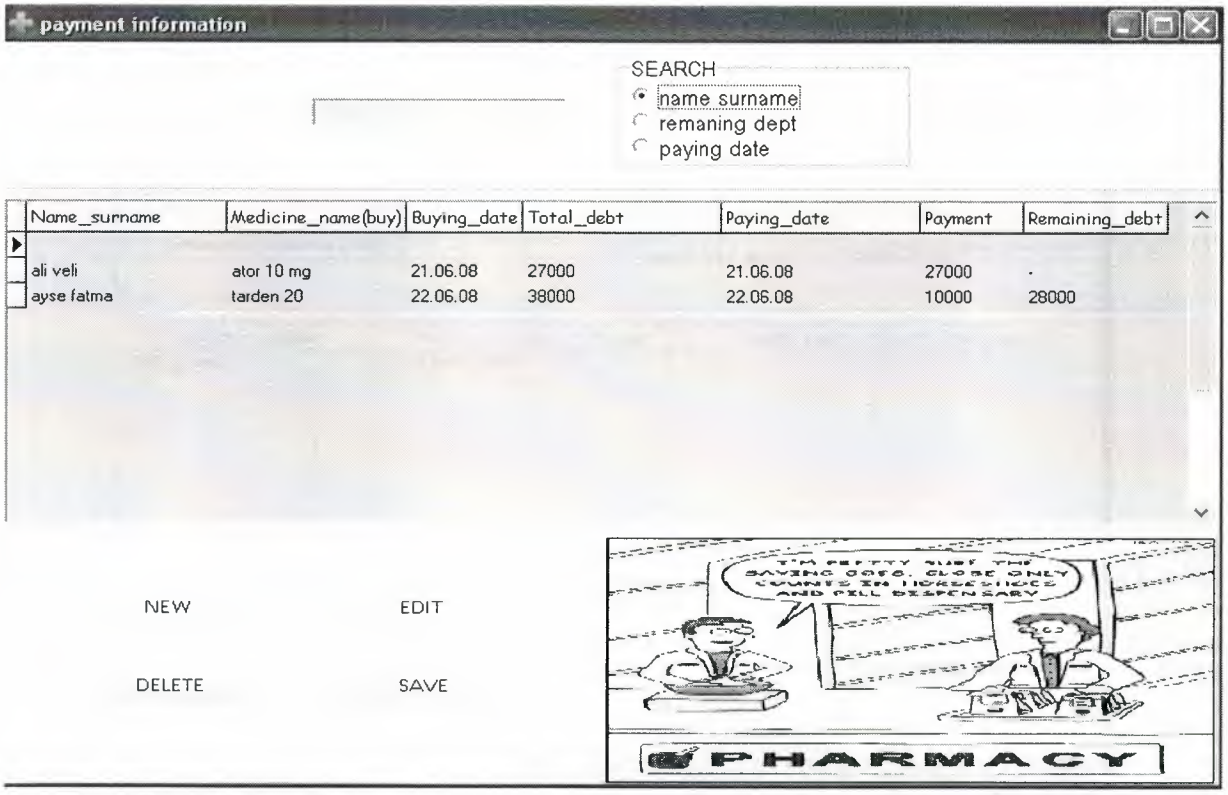

Figure 4.15.paymant information screen

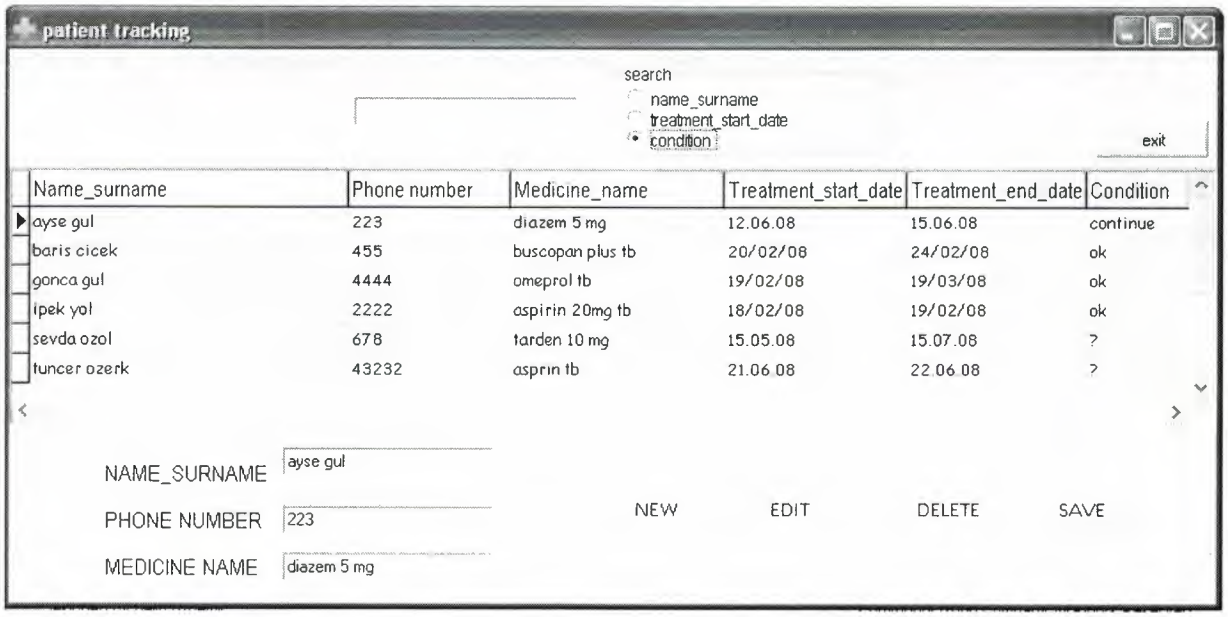

Figure 4.16.patient tracking screen

4.8.Prescription screen

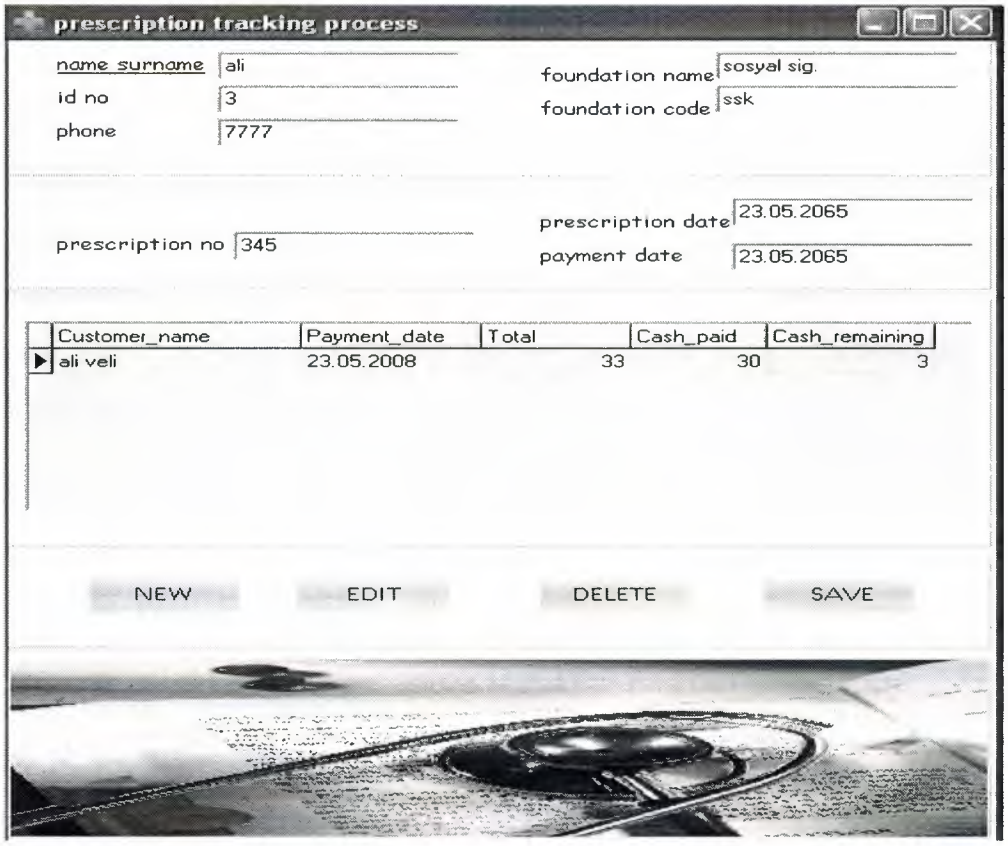

Figure 4.17.prescription tracking process screen

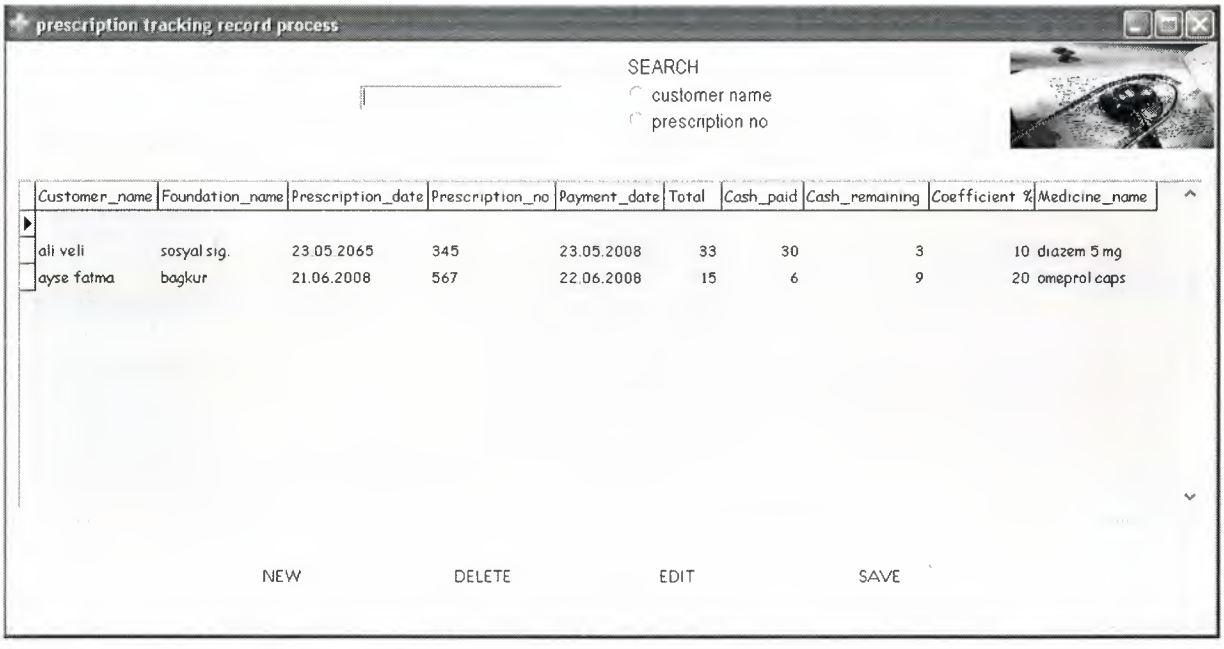

Figure 4.18.prescription tracking record process screen

CONCLUSION

Delphi is an easy program to grasp.Because of this reason this program is decided to used by operators.

Delphi is a Microsoft Windows programming Language .Delphi is a distinctly different language providing powerfull features such as graphical user interfaces ,even handling , access to the Win32 API , object–oriented features , error handling ,structured programming, and much more.

In this project medicine database was built by programmers.. It is easy to use and It can be used by most kind of drugstore.Delphi was used for writing this programme.Paradox 7 was used for keeping all my database.

In this study our main aim to put across is that this program can be operated by someone who has never used it before.

In this program there is also menus to make your writting much simpler,It containing windows menus and also a facility to prepare reports.

REFERENCES

1- http://www.delphiturk.com/

2- http://www.programlama.com/

3- http://www.delphiturkiye.com/

4- Özyol Pharmacy/Nicosia-KKTC

APPENDIX

```
unit Unit14;

interface

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

type
  TForm14 = class(TForm)
    Image1: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form14: TForm14;

implementation

{$R *.dfm}

end.
unit Unit1;
```

```
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, OleServer, WordXP, ExtCtrls, StdCtrls, ComCtrls, Buttons,
  jpeg, TabNotBk, DB, DBTables, Grids, DBGrids;

type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    medicine1: TMenuItem;
    warehouse1: TMenuItem;
    stocks1: TMenuItem;
    hospital1: TMenuItem;
    adddeleteedit1: TMenuItem;
    medicineprices1: TMenuItem;
    medicineinstock1: TMenuItem;
    warehouseinformation1: TMenuItem;
    warehouseprices1: TMenuItem;
    prospekts1: TMenuItem;
    customer1: TMenuItem;
    customerinformation1: TMenuItem;
    patienttracking1: TMenuItem;
    reports1: TMenuItem;
    prescription1: TMenuItem;
    prescriptiontracking1: TMenuItem;
    recordprocess1: TMenuItem;
    Timer1: TTimer;
    Panel2: TPanel;
    StatusBar1: TStatusBar;
    bills1: TMenuItem;
```

foundation1: TMenuItem;

foundationinfo1: TMenuItem;

view1: TMenuItem;

Image1: TImage;

Label1: TLabel;

Label2: TLabel;

Image2: TImage;

BitBtn7: TBitBtn;

BitBtn9: TBitBtn;

BitBtn10: TBitBtn;

BitBtn11: TBitBtn;

BitBtn12: TBitBtn;

BitBtn13: TBitBtn;

BitBtn14: TBitBtn;

TabbedNotebook1: TTabbedNotebook;

BitBtn2: TBitBtn;

Image3: TImage;

Label3: TLabel;

BitBtn3: TBitBtn;

BitBtn6: TBitBtn;

BitBtn15: TBitBtn;

DBGrid1: TDBGrid;

DataSource1: TDataSource;

Query1: TQuery;

Image4: TImage;

Image5: TImage;

BitBtn16: TBitBtn;

BitBtn17: TBitBtn;

BitBtn18: TBitBtn;

BitBtn4: TBitBtn;

BitBtn8: TBitBtn;

```
Edit1: TEdit;
DBGrid2: TDBGrid;
DataSource2: TDataSource;
Query2: TQuery;
BitBtn1: TBitBtn;
BitBtn19: TBitBtn;
BitBtn21: TBitBtn;
BitBtn22: TBitBtn;
BitBtn23: TBitBtn;
BitBtn24: TBitBtn;
BitBtn25: TBitBtn;
BitBtn5: TBitBtn;
BitBtn20: TBitBtn;
Query3: TQuery;
DataSource3: TDataSource;
DBGrid3: TDBGrid;
Edit2: TEdit;
Label4: TLabel;
Edit3: TEdit;
Label5: TLabel;
procedure adddeleteedit1Click(Sender: TObject);
procedure medicineprices1Click(Sender: TObject);
procedure medicineinstock1Click(Sender: TObject);
procedure prospekts1Click(Sender: TObject);
procedure customerinformation1Click(Sender: TObject);
procedure patienttracking1Click(Sender: TObject);
procedure warehouseinformation1Click(Sender: TObject);
procedure warehouseprices1Click(Sender: TObject);
procedure reports1Click(Sender: TObject);
procedure prescriptiontracking1Click(Sender: TObject);
procedure recordprocess1Click(Sender: TObject);
```

```
procedure Timer1Timer(Sender: TObject);
procedure hospital1Click(Sender: TObject);
procedure stocks1Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure bills1Click(Sender: TObject);
procedure foundationinfo1Click(Sender: TObject);
procedure view1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure BitBtn7Click(Sender: TObject);
procedure BitBtn8Click(Sender: TObject);
procedure BitBtn11Click(Sender: TObject);
procedure BitBtn14Click(Sender: TObject);
procedure BitBtn13Click(Sender: TObject);
procedure BitBtn12Click(Sender: TObject);
procedure BitBtn10Click(Sender: TObject);
procedure BitBtn9Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn16Click(Sender: TObject);
procedure BitBtn17Click(Sender: TObject);
procedure BitBtn18Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
procedure BitBtn15Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn20Click(Sender: TObject);
procedure BitBtn19Click(Sender: TObject);
procedure BitBtn21Click(Sender: TObject);
procedure BitBtn22Click(Sender: TObject);
```

```
  procedure BitBtn23Click(Sender: TObject);
  procedure BitBtn24Click(Sender: TObject);
  procedure BitBtn25Click(Sender: TObject);

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

var
 Form1: TForm1;

implementation

uses Unit2, Unit4, Unit5, Unit3, Unit6, Unit7, Unit8, Unit9, Unit10,
 Unit11, Unit12, Unit15, Unit16, Unit17, Unit18, Unit19, Unit20, Unit21,
 Unit22, Unit23, Unit24, Unit25, Unit26, Unit27, Unit13;

{$R *.dfm}

procedure TForm1.adddeleteedit1Click(Sender: TObject);
begin
form2.show;
end;

procedure TForm1.medicinepricesl Click(Sender: TObject);
begin
form4.show;
end;
```

```
procedure TForm1.medicineinstock1Click(Sender: TObject);
begin
form3.show;
end;

procedure TForm1.prospekts1Click(Sender: TObject);
begin
form5.Show;
end;

procedure TForm1.customerinformation1Click(Sender: TObject);
begin
form6.show;
end;

procedure TForm1.patienttracking1Click(Sender: TObject);
begin
form7.show;
end;

procedure TForm1.warehouseinformation1Click(Sender: TObject);
begin
form8.show;
end;

procedure TForm1.warehouseprices1Click(Sender: TObject);
begin
form9.show;
end;

procedure TForm1.reports1Click(Sender: TObject);
```

```
begin
form10.show;
end;

procedure TForm1.prescriptiontracking1Click(Sender: TObject);
begin
form11.show;
end;

procedure TForm1.recordprocess1Click(Sender: TObject);
begin
form12.show;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
caption:=copy(caption,2,length(caption)-1)+caption[1];
label1.caption:=datetostr(date);
label2.caption:=timetostr(time);
end;

procedure TForm1.hospital1Click(Sender: TObject);
begin
form15.show;
end;

procedure TForm1.stocks1Click(Sender: TObject);
begin
form16.show;
end;
```

```
procedure TForm1.BitBtn4Click(Sender: TObject);
begin
form16.show;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
form16.Show;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin

form21.show;
end;

procedure TForm1.bills1Click(Sender: TObject);
begin
form24.Show;
end;

procedure TForm1.foundationinfo1Click(Sender: TObject);
begin
form25.show;
end;

procedure TForm1.view1Click(Sender: TObject);
begin
form26.show;
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
begin
SetWindowRgn(form1.handle,
    CreateRoundRectRgn(5,5,form1.Width-5,form1.Height-5,15,15),
    True);

    SetWindowRgn(bitbtn10.handle,
    CreateRoundRectRgn(5,5,bitbtn10.Width-5,bitbtn10.Height-5,15,15),
    True);
    SetWindowRgn(bitbtn12.handle,
    CreateRoundRectRgn(5,5,bitbtn12.Width-5,bitbtn12.Height-5,15,15),
    True);
    SetWindowRgn(bitbtn13.handle,
    CreateRoundRectRgn(5,5,bitbtn13.Width-5,bitbtn13.Height-5,15,15),
    True);
    SetWindowRgn(bitbtn14.handle,
    CreateRoundRectRgn(5,5,bitbtn14.Width-5,bitbtn14.Height-5,15,15),
    True);
    SetWindowRgn(bitbtn17.handle,
    CreateRoundRectRgn(5,5,bitbtn17.Width-5,bitbtn17.Height-5,15,15),
    True);
    SetWindowRgn(bitbtn18.handle,
    CreateRoundRectRgn(5,5,bitbtn18.Width-5,bitbtn18.Height-5,15,15),
    True);
    SetWindowRgn(bitbtn7.handle,
    CreateRoundRectRgn(5,5,bitbtn7.Width-5,bitbtn7.Height-5,15,15),
    True);
    SetWindowRgn(bitbtn9.handle,
    CreateRoundRectRgn(5,5,bitbtn9.Width-5,bitbtn9.Height-5,15,15),
    True);
    SetWindowRgn(bitbtn16.handle,
```

```
    CreateRoundRectRgn(5,5,bitbtn16.Width-5,bitbtn16.Height-5,15,15),
    True);
    SetWindowRgn(bitbtn2.handle,
    CreateRoundRectRgn(5,5,bitbtn2.Width-5,bitbtn2.Height-5,15,15),
    True);
    SetWindowRgn(bitbtn8.handle,
    CreateRoundRectRgn(5,5,bitbtn8.Width-5,bitbtn8.Height-5,15,15),
    True);
    SetWindowRgn(bitbtn4.handle,
    CreateRoundRectRgn(5,5,bitbtn4.Width-5,bitbtn4.Height-5,15,15),
    True);

end;

procedure TForm1.BitBtn7Click(Sender: TObject);
begin
form21.Show;
end;

procedure TForm1.BitBtn8Click(Sender: TObject);
begin
form17.show;
end;

procedure TForm1.BitBtn11Click(Sender: TObject);
begin
form13.close;
end;

procedure TForm1.BitBtn14Click(Sender: TObject);
begin
```

```
form19.show;
end;

procedure TForm1.BitBtn13Click(Sender: TObject);
begin
form20.show;
end;

procedure TForm1.BitBtn12Click(Sender: TObject);
begin
form27.show;
end;

procedure TForm1.BitBtn10Click(Sender: TObject);
begin
form23.show;
end;

procedure TForm1.BitBtn9Click(Sender: TObject);
begin
form24.show;
end;

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
form15.Show;
end;

procedure TForm1.BitBtn16Click(Sender: TObject);
begin
form22.show;
```

```
end;

procedure TForm1.BitBtn17Click(Sender: TObject);
begin
form10.Show;
end;

procedure TForm1.BitBtn18Click(Sender: TObject);
begin
form18.show;
end;

procedure TForm1.BitBtn3Click(Sender: TObject);
begin
edit1.text:='';
edit1.SetFocus;
query1.insert;
end;

procedure TForm1.BitBtn5Click(Sender: TObject);
begin
query1.Edit;
edit1.SetFocus;
end;

procedure TForm1.BitBtn6Click(Sender: TObject);
var
a:word;
begin
a:=application.MessageBox('are u sure?','warning',36);
if (a=idyes )then
```

```
begin
query1.delete;
end;
end;

procedure TForm1.BitBtn15Click(Sender: TObject);
begin
query1.Post;
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
edit2.text:='';
edit2.SetFocus;
query2.insert;
end;

procedure TForm1.BitBtn20Click(Sender: TObject);
begin
query2.Edit;
edit2.SetFocus;
end;

procedure TForm1.BitBtn19Click(Sender: TObject);

var
a:word;
begin
a:=application.MessageBox('are u sure?','warning',36);
if (a=idyes )then
begin
```

```
query2.delete;
end;
end;

procedure TForm1.BitBtn21Click(Sender: TObject);
begin
query2.Post;
end;

procedure TForm1.BitBtn22Click(Sender: TObject);
begin
edit3.text:='';
edit3.SetFocus;
query3.insert;
end;

procedure TForm1.BitBtn23Click(Sender: TObject);

var
a:word;
begin
a:=application.MessageBox('are u sure?','warning',36);
if (a=idyes )then
begin
query3.delete;
end;
end;

procedure TForm1.BitBtn24Click(Sender: TObject);
begin
query3.Post;
```

```
end;

procedure TForm1.BitBtn25Click(Sender: TObject);
begin
query3.Edit;
edit3.SetFocus;
end;

end.
unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, Grids, DBGrids, StdCtrls, Mask, DBCtrls, ExtCtrls,
  FMTBcd, SqlExpr, Buttons, jpeg;

type
  TForm2 = class(TForm)
    DataSource1: TDataSource;
    Query1: TQuery;
    Panel1: TPanel;
    Panel5: TPanel;
    Image1: TImage;
    DBGrid1: TDBGrid;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    Image2: TImage;
```

```
  Label1: TLabel;
  Edit1: TEdit;
  procedure Button1Click(Sender: TObject);
  procedure Edit1Change(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn2Click(Sender: TObject);
  procedure BitBtn3Click(Sender: TObject);
  procedure BitBtn4Click(Sender: TObject);



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

var
 Form2: TForm2;

implementation

{$R *.dfm}



procedure TForm2.Button1Click(Sender: TObject);
begin
query1.Close;
query1.Sql.Clear;
query1.SQL.Add('select barcode,medicine_name,active_substance,equivalent_medicine');
```

```
query1.SQL.Add('from equivalent.db');
query1.Open;
end;

procedure TForm2.Edit1Change(Sender: TObject);
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from equivalent where medicine_name like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;

procedure TForm2.BitBtn1Click(Sender: TObject);
begin
edit1.text:='';
edit1.SetFocus;
query1.insert;
end;

procedure TForm2.BitBtn2Click(Sender: TObject);
begin
 query1.Edit;
edit1.SetFocus;
end;

procedure TForm2.BitBtn3Click(Sender: TObject);
var
a:word;
begin
a:=application.messagebox('are you sure?','warning',36);
```

```
if (a=idyes)then
begin
query1.Delete;
end;
end;

procedure TForm2.BitBtn4Click(Sender: TObject);
begin
query1.Post;
end;

end.
unit Unit3;

interface

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

type
 TForm3 = class(TForm)
   DataSource1: TDataSource;
   Query1: TQuery;
   Panel5: TPanel;
   Image1: TImage;
   Label7: TLabel;
   Edit1: TEdit;
   DBGrid1: TDBGrid;
   BitBtn1: TBitBtn;
```

```
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    procedure Button1Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

{$R *.dfm}

procedure TForm3.Button1Click(Sender: TObject);
begin
query1.close;
query1.sql.clear;
query1.sql.add('select barcode,medicine_name,stock_on_hand,shelf_quantity');
query1.sql.add('from medicine_stock.db ');
query1.requestlive:=true;
query1.open;
```

```
end;

procedure TForm3.Edit1Change(Sender: TObject);
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from medicine_stock where medicine_name like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;

procedure TForm3.BitBtn1Click(Sender: TObject);
begin
edit1.text:='';
edit1.SetFocus;
query1.insert;
end;

procedure TForm3.BitBtn2Click(Sender: TObject);
begin
query1.Edit;
edit1.SetFocus;
end;

procedure TForm3.BitBtn3Click(Sender: TObject);
var
a:word;
begin
a:=application.messagebox('are you sure?','warning',36);
if (a=idyes)then
begin
```

```
query1.Delete;
end;
end;

procedure TForm3.BitBtn4Click(Sender: TObject);
begin
query1.Post;
end;

end.
unit Unit4;

interface

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

type
  TForm4 = class(TForm)
    DataSource1: TDataSource;
    Query1: TQuery;
    Panel8: TPanel;
    Image1: TImage;
    DBGrid1: TDBGrid;
    Edit1: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
```

```
    Label5: TLabel;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    DBEdit1: TDBEdit;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    DBEdit8: TDBEdit;
    DBEdit9: TDBEdit;
    DBEdit2: TDBEdit;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Label10: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    DBEdit5: TDBEdit;
    DBEdit6: TDBEdit;
    DBEdit7: TDBEdit;
    Label14: TLabel;
    Label15: TLabel;
    procedure Edit2Change(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure BitBtn5Click(Sender: TObject);
```

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

var
 Form4: TForm4;

implementation

uses Unit3;

{$R *.dfm}

procedure TForm4.Edit2Change(Sender: TObject);
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * from medicine_price where medicine_name
like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;
```

```
procedure TForm4.BitBtn1Click(Sender: TObject);
begin
edit1.text:='';
edit1.SetFocus;
query1.insert;
end;

procedure TForm4.BitBtn2Click(Sender: TObject);
begin
query1.Edit;
edit1.SetFocus;
end;

procedure TForm4.BitBtn3Click(Sender: TObject);
var
a:word;
begin
a:=application.messagebox('are you sure?','warning',36);
if (a=idyes)then
begin
query1.Delete;
end;

end;

procedure TForm4.BitBtn4Click(Sender: TObject);
begin
query1.Post;
end;
```

```
procedure TForm4.Edit1Change(Sender: TObject);
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from medicine_price where medicine_name like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;

procedure TForm4.BitBtn5Click(Sender: TObject);
begin
form3.show;
end;

end.
unit Unit5;

interface

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

type
  TForm5 = class(TForm)
    DataSource1: TDataSource;
    Query1: TQuery;
    Panel4: TPanel;
    Image1: TImage;
```

```
    DBGrid1: TDBGrid;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    Label4: TLabel;
    Edit1: TEdit;
    DBMemo1: TDBMemo;
    DBMemo2: TDBMemo;
    DBMemo3: TDBMemo;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    procedure Edit1Change(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form5: TForm5;

implementation

{$R *.dfm}
```

```
procedure TForm5.Edit1Change(Sender: TObject);
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from equivalent where medicine_name like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;

procedure TForm5.BitBtn1Click(Sender: TObject);
begin
edit1.text:='';
edit1.SetFocus;
query1.insert;
end;

procedure TForm5.BitBtn2Click(Sender: TObject);
begin
query1.Edit;
edit1.SetFocus;
end;

procedure TForm5.BitBtn3Click(Sender: TObject);
begin
query1.Post;
end;

procedure TForm5.BitBtn4Click(Sender: TObject);
var
a:word;
begin
```

```
a:=application.messagebox('are you sure?','warning',36);
if (a=idyes)then
begin
query1.Delete;
end;

end;

end.
unit Unit6;

interface

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

type
  TForm6 = class(TForm)
    Query1: TQuery;
    DBGrid1: TDBGrid;
    Panel1: TPanel;
    DataSource1: TDataSource;
    Edit1: TEdit;
    RadioGroup1: TRadioGroup;
    Panel2: TPanel;
    Panel3: TPanel;
    Image1: TImage;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
```

```
  BitBtn3: TBitBtn;
  BitBtn4: TBitBtn;
  procedure Edit1Change(Sender: TObject);
  procedure BitBtn2Click(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn3Click(Sender: TObject);
  procedure BitBtn4Click(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
 Form6: TForm6;

implementation

{$R *.dfm}

procedure TForm6.Edit1Change(Sender: TObject);
begin
if (radiogroup1.itemindex=0)then
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from payment where name_surname like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;
if (radiogroup1.itemindex=1)then
```

```
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from payment where remaining_debt like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;
if (radiogroup1.itemindex=2)then
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from payment where paying_date like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;
end;

procedure TForm6.BitBtn2Click(Sender: TObject);
var
a:word;
begin
a:=application.MessageBox('are u sure?','warning',36);
if (a=idyes )then
begin
query1.delete;
end;
end;

procedure TForm6.BitBtn1Click(Sender: TObject);
begin
edit1.text:='';
```

```
edit1.SetFocus;
query1.insert;
end;

procedure TForm6.BitBtn3Click(Sender: TObject);
begin
query1.Edit;
edit1.SetFocus;
end;

procedure TForm6.BitBtn4Click(Sender: TObject);
begin
query1.Post;
end;

end.
unit Unit7;

interface

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

type
  TForm7 = class(TForm)
    Panel1: TPanel;
    DBGrid1: TDBGrid;
    Query1: TQuery;
    DataSource1: TDataSource;
```

```
    Panel2: TPanel;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    BitBtn5: TBitBtn;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit3: TDBEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Edit1: TEdit;
    RadioGroup1: TRadioGroup;
    Label4: TLabel;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure BitBtn5Click(Sender: TObject);
    procedure DBEdit1KeyPress(Sender: TObject; var Key: Char);
    procedure DBEdit2KeyPress(Sender: TObject; var Key: Char);
    procedure DBEdit3KeyPress(Sender: TObject; var Key: Char);
    procedure Edit1Change(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
```

```
  Form7: TForm7;

implementation

uses Unit1;

{$R *.dfm}

procedure TForm7.BitBtn1Click(Sender: TObject);
begin
dbedit1.Text:='';
dbedit1.SetFocus;
query1.insert;
end;

procedure TForm7.BitBtn2Click(Sender: TObject);
begin
query1.Post;
end;

procedure TForm7.BitBtn3Click(Sender: TObject);
var
a:word;
begin
a:=application.MessageBox('are u sure?','warning',36);
if (a=idyes )then
begin
query1.delete;
end;

end;
```

```
procedure TForm7.BitBtn4Click(Sender: TObject);
begin
query1.Edit;
dbedit1.SetFocus;
end;

procedure TForm7.BitBtn5Click(Sender: TObject);
begin
form1.close;
end;

procedure TForm7.DBEdit1KeyPress(Sender: TObject; var Key: Char);
begin
if(key=#13)then dbedit1.setfocus;
end;

procedure TForm7.DBEdit2KeyPress(Sender: TObject; var Key: Char);
begin
if(key=#13)then dbedit2.setfocus;
end;

procedure TForm7.DBEdit3KeyPress(Sender: TObject; var Key: Char);
begin
if(key=#13)then dbedit1.setfocus;
end;

procedure TForm7.Edit1Change(Sender: TObject);
begin
if (radiogroup1.itemindex=0)then
begin
```

```
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from patient_tracking where name_surname like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;
if (radiogroup1.itemindex=1)then
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from patient_tracking where treatment_start_date like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;
if (radiogroup1.itemindex=2)then
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from patient_tracking where condition like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;
end;


end.
unit Unit8;


interface


uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
```

```
  Dialogs, StdCtrls, Buttons, DBCtrls, ExtCtrls, DB, DBTables, Mask, jpeg;

type
  TForm8 = class(TForm)
    DataSource1: TDataSource;
    Query1: TQuery;
    Image1: TImage;
    BitBtn1: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    DBEdit5: TDBEdit;
    DBEdit4: TDBEdit;
    Label6: TLabel;
    Label5: TLabel;
    DBEdit3: TDBEdit;
    Label4: TLabel;
    DBEdit2: TDBEdit;
    Label3: TLabel;
    DBEdit1: TDBEdit;
    Label2: TLabel;
    Label1: TLabel;
    Edit6: TEdit;
    BitBtn5: TBitBtn;
    procedure Edit6Change(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure BitBtn5Click(Sender: TObject);
```

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

var
 Form8: TForm8;

implementation

{$R *.dfm}
```

```
procedure TForm8.Edit6Change(Sender: TObject);
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from warehouse where warehouse_name like'+#39+(edit6.Text)+'%'+#39);
query1.Open;
```

```
end;

procedure TForm8.BitBtn1Click(Sender: TObject);
begin
edit6.text:='';
edit6.SetFocus;
query1.insert;
end;

procedure TForm8.BitBtn3Click(Sender: TObject);
var
a:word;
begin
a:=application.messagebox('are you sure?','warning',36);
if (a=idyes)then
begin
query1.Delete;
end;
end;

procedure TForm8.BitBtn4Click(Sender: TObject);
begin
query1.Post;
end;

procedure TForm8.BitBtn5Click(Sender: TObject);
begin
query1.Edit;
edit6.SetFocus;
end;
```

```
end.
unit Unit9;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, jpeg, ExtCtrls, StdCtrls, Grids, DBGrids, Mask, DBCtrls, DB,
  DBTables;

type
  TForm9 = class(TForm)
    Panel2: TPanel;
    DBGrid1: TDBGrid;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    DataSource1: TDataSource;
    Query1: TQuery;
    Image1: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Edit1: TEdit;
    Label6: TLabel;
    procedure Edit1Change(Sender: TObject);
  private
    { Private declarations }
```

```
 public
  { Public declarations }
 end;

var
 Form9: TForm9;

implementation

{$R *.dfm}

procedure TForm9.Edit1Change(Sender: TObject);
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from warehouse_price where warehousename like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;

end.

unit Unit10;

interface

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

```
type
 TForm10 = class(TForm)
  Panel1: TPanel;
  SaveDialog1: TSaveDialog;
  PrintDialog1: TPrintDialog;
  Image1: TImage;
  richReport: TMemo;
  Panel2: TPanel;
  BitBtn1: TBitBtn;
  BitBtn2: TBitBtn;
  BitBtn3: TBitBtn;
  Label1: TLabel;
  SpeedButton1: TSpeedButton;
  SpeedButton2: TSpeedButton;
  SpeedButton3: TSpeedButton;
  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn3Click(Sender: TObject);
  procedure SpeedButton1Click(Sender: TObject);
  procedure SpeedButton2Click(Sender: TObject);
  procedure SpeedButton3Click(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
```

```
 Form10: TForm10;

implementation

{$R *.dfm}

procedure TForm10.BitBtn1Click(Sender: TObject);
begin
if SaveDialog1.Execute then
     richReport.Lines.SaveToFile(SaveDialog1.FileName + '.txt');
end;

procedure TForm10.BitBtn3Click(Sender: TObject);
begin
close;
end;

procedure TForm10.SpeedButton1Click(Sender: TObject);
begin
if speedbutton1.Down then
richreport.Font.Style:=richreport.Font.Style+[fsbold]
else
richreport.Font.Style:=richreport.Font.Style-[fsbold];
end;

procedure TForm10.SpeedButton2Click(Sender: TObject);
begin
```

```
if speedbutton2.Down then
richreport.Font.Style:=richreport.Font.Style+[fsitalic]
else
richreport.Font.Style:=richreport.Font.Style-[fsitalic];
end;

procedure TForm10.SpeedButton3Click(Sender: TObject);
begin
if speedbutton1.Down then
richreport.Font.Style:=richreport.Font.Style+[fsunderline]
else
richreport.Font.Style:=richreport.Font.Style-[fsunderline];
end;

end.
unit Unit11;

interface

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

type
  TForm11 = class(TForm)
    Panel1: TPanel;
```

Panel2: TPanel;

Panel3: TPanel;

DBGrid1: TDBGrid;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Panel4: TPanel;

Label8: TLabel;

Query1: TQuery;

DataSource1: TDataSource;

DBEdit1: TDBEdit;

DBEdit2: TDBEdit;

Edit1: TEdit;

DBEdit3: TDBEdit;

DBEdit4: TDBEdit;

DBEdit5: TDBEdit;

DBEdit6: TDBEdit;

DBEdit7: TDBEdit;

BitBtn1: TBitBtn;

BitBtn2: TBitBtn;

BitBtn3: TBitBtn;

BitBtn4: TBitBtn;

Panel5: TPanel;

```
    Image1: TImage;
    procedure Edit1Change(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form11: TForm11;

implementation

{$R *.dfm}

procedure TForm11.Edit1Change(Sender: TObject);
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from prescription where customer_name like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;
```

```
procedure TForm11.BitBtn1Click(Sender: TObject);
begin
edit1.text:='';
edit1.SetFocus;
query1.insert;
end;

procedure TForm11.BitBtn2Click(Sender: TObject);
begin
query1.Edit;
edit1.SetFocus;
end;

procedure TForm11.BitBtn3Click(Sender: TObject);
var
a:word;
begin
a:=application.messagebox('are you sure?','warning',36);
if (a=idyes)then
begin
query1.Delete;
end;
end;

procedure TForm11.BitBtn4Click(Sender: TObject);
begin
```

```
query1.Post;
end;

end.
unit Unit12;

interface

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

type
  TForm12 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    DBGrid1: TDBGrid;
    Panel3: TPanel;
    Query1: TQuery;
    DataSource1: TDataSource;
    Edit1: TEdit;
    RadioGroup1: TRadioGroup;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    Image1: TImage;
```

```
  procedure Edit1Change(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn2Click(Sender: TObject);
  procedure BitBtn3Click(Sender: TObject);
  procedure BitBtn4Click(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
 Form12: TForm12;

implementation

{$R *.dfm}

procedure TForm12.Edit1Change(Sender: TObject);
begin
if (radiogroup1.itemindex=0)then
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from prescription where customer_name like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
```

```
end;
if (radiogroup1.itemindex=1)then
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from prescirption where prescription_no like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;



end;

procedure TForm12.BitBtn1Click(Sender: TObject);
begin
edit1.Text:='';
edit1.SetFocus;
query1.insert;
end;

procedure TForm12.BitBtn2Click(Sender: TObject);
var
a:word;
begin
a:=application.MessageBox('are u sure?','warning',36);
if (a=idyes )then
begin
```

```
query1.delete;
end;
end;

procedure TForm12.BitBtn3Click(Sender: TObject);
begin
query1.Edit;
edit1.SetFocus;
end;

procedure TForm12.BitBtn4Click(Sender: TObject);
begin
query1.Post;
end;

end.
unit Unit14;

interface

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

type
  TForm14 = class(TForm)
    Image1: TImage;
```

```
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
 Form14: TForm14;

implementation

{$R *.dfm}

end.
unit Unit15;

interface

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

type
```

```
TForm15 = class(TForm)
  Panel1: TPanel;
  Panel2: TPanel;
  Panel3: TPanel;
  Label1: TLabel;
  Edit1: TEdit;
  DBGrid1: TDBGrid;
  DBEdit1: TDBEdit;
  DBEdit2: TDBEdit;
  DBEdit3: TDBEdit;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  DataSource1: TDataSource;
  Query1: TQuery;
  BitBtn1: TBitBtn;
  BitBtn2: TBitBtn;
  BitBtn3: TBitBtn;
  BitBtn4: TBitBtn;
  procedure Edit1Change(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn2Click(Sender: TObject);
  procedure BitBtn4Click(Sender: TObject);
  procedure BitBtn3Click(Sender: TObject);
private
  { Private declarations }
public
```

```
  { Public declarations }
 end;

var
 Form15: TForm15;

implementation

{$R *.dfm}

procedure TForm15.Edit1Change(Sender: TObject);
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from hospital where hospitalname like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;

procedure TForm15.BitBtn1Click(Sender: TObject);
begin
edit1.text:='';
edit1.SetFocus;
query1.insert;
end;

procedure TForm15.BitBtn2Click(Sender: TObject);
```

```
begin
query1.Edit;
edit1.SetFocus;
end;

procedure TForm15.BitBtn4Click(Sender: TObject);
begin
query1.Post;
end;

procedure TForm15.BitBtn3Click(Sender: TObject);
var
a:word;
begin
a:=application.messagebox('are you sure?','warning',36);
if (a=idyes)then
begin
query1.Delete;
end;
end;

end.
unit Unit16;

interface

uses
```

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

type
  TForm16 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    TabbedNotebook1: TTabbedNotebook;
    DataSource1: TDataSource;
    Query1: TQuery;
    DBGrid1: TDBGrid;
    Edit1: TEdit;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    DBGrid2: TDBGrid;
    DataSource2: TDataSource;
    Query2: TQuery;
    Edit2: TEdit;
    BitBtn5: TBitBtn;
    BitBtn6: TBitBtn;
    BitBtn7: TBitBtn;
    BitBtn8: TBitBtn;
    DataSource3: TDataSource;
    DBGrid3: TDBGrid;
```

```
Query3: TQuery;
Edit3: TEdit;
BitBtn9: TBitBtn;
BitBtn10: TBitBtn;
BitBtn11: TBitBtn;
BitBtn12: TBitBtn;
Image2: TImage;
Label1: TLabel;
Image1: TImage;
Image3: TImage;
Label2: TLabel;
Label3: TLabel;
BitBtn13: TBitBtn;
procedure Edit1Change(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
procedure BitBtn7Click(Sender: TObject);
procedure BitBtn8Click(Sender: TObject);
procedure Edit2Change(Sender: TObject);
procedure BitBtn9Click(Sender: TObject);
procedure BitBtn10Click(Sender: TObject);
procedure BitBtn11Click(Sender: TObject);
procedure BitBtn12Click(Sender: TObject);
```

```
  procedure Edit3Change(Sender: TObject);
  procedure BitBtn13Click(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
 Form16: TForm16;

implementation

uses Unit1;

{$R *.dfm}

procedure TForm16.Edit1Change(Sender: TObject);
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from medical where medical_material like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;

procedure TForm16.BitBtn1Click(Sender: TObject);
```

```
begin
edit1.text:='';
edit1.SetFocus;
query1.insert;
end;

procedure TForm16.BitBtn2Click(Sender: TObject);
begin
query1.Edit;
edit1.SetFocus;
end;

procedure TForm16.BitBtn3Click(Sender: TObject);
var
a:word;
begin
a:=application.MessageBox('are u sure?','warning',36);
if (a=idyes )then
begin
query1.delete;
end;
end;

procedure TForm16.BitBtn4Click(Sender: TObject);
begin
query1.Post;
end;
```

```
procedure TForm16.BitBtn5Click(Sender: TObject);
begin
edit2.text:='';
edit2.SetFocus;
query2.insert;
end;

procedure TForm16.BitBtn6Click(Sender: TObject);
begin
query2.Edit;
edit2.SetFocus;
end;

procedure TForm16.BitBtn7Click(Sender: TObject);
var
a:word;
begin
a:=application.MessageBox('are u sure?','warning',36);
if (a=idyes )then
begin
query2.delete;
end;
end;

procedure TForm16.BitBtn8Click(Sender: TObject);
begin
```

```
query2.Post;
end;

procedure TForm16.Edit2Change(Sender: TObject);
begin
query2.Close;
query2.SQL.Clear;
query2.SQL.Add('select * ');
query2.SQL.add('from medicine_stock where medicine_name
like'+#39+(edit2.Text)+'%'+#39);
query2.Open;
end;

procedure TForm16.BitBtn9Click(Sender: TObject);
begin
edit3.text:='';
edit3.SetFocus;
query3.insert;
end;

procedure TForm16.BitBtn10Click(Sender: TObject);
begin
query3.Edit;
edit3.SetFocus;
end;

procedure TForm16.BitBtn11Click(Sender: TObject);
```

```
var
a:word;
begin
a:=application.messagebox('are you sure?','warning',36);
if (a=idyes)then
begin
query3.Delete;
end;
end;

procedure TForm16.BitBtn12Click(Sender: TObject);
begin
query3.Post;
end;

procedure TForm16.Edit3Change(Sender: TObject);
begin
query3.Close;
query3.SQL.Clear;
query3.SQL.Add('select * ');
query3.SQL.add('from cosmetic where product_name like'+#39+(edit3.Text)+'%'+#39);
query3.Open;
end;

procedure TForm16.BitBtn13Click(Sender: TObject);
begin
form16.Close;
```

```
form1.show;
end;

end.
unit Unit17;

interface

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

type
  TForm17 = class(TForm)
    Panel1: TPanel;
    DBGrid1: TDBGrid;
    DataSource1: TDataSource;
    Table1: TTable;
    DBNavigator1: TDBNavigator;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Panel2: TPanel;
    Image1: TImage;
```

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

var
  Form17: TForm17;

implementation

{$R *.dfm}

end.
unit Unit18;

interface

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

type
  TForm18 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Label1: TLabel;
```

```
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  Label8: TLabel;
  Label9: TLabel;
  Label10: TLabel;
  Label12: TLabel;
  Label13: TLabel;
  Label14: TLabel;
  Label15: TLabel;
  Label16: TLabel;
  Label11: TLabel;
  Label17: TLabel;
  Label18: TLabel;
  Label19: TLabel;
  Label20: TLabel;
  Label21: TLabel;
  Label22: TLabel;
private
  { Private declarations }
public
  { Public declarations }
end;
```

```
var
  Form18: TForm18;

implementation

{$R *.dfm}

end.
unit Unit19;

interface

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

type
  TForm19 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    DBGrid1: TDBGrid;
    Query1: TQuery;
    DataSource1: TDataSource;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    Panel3: TPanel;
```

```
    Edit1: TEdit;
    MonthCalendar1: TMonthCalendar;
    Image1: TImage;
    Timer1: TTimer;
    Label1: TLabel;
    BitBtn3: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form19: TForm19;

implementation

uses Unit1;

{$R *.dfm}

procedure TForm19.BitBtn1Click(Sender: TObject);
begin
```

```
edit1.text:='';
edit1.SetFocus;
query1.insert;
end;

procedure TForm19.BitBtn2Click(Sender: TObject);
var
a:word;
begin
a:=application.messagebox('are you sure?','warning',36);
if (a=idyes)then
begin
query1.Delete;
end;
end;

procedure TForm19.Timer1Timer(Sender: TObject);
begin
caption:=copy(caption,2,length(caption)-1)+caption[1];
label1.caption:=datetostr(date);
label1.caption:=timetostr(time);


end;

procedure TForm19.BitBtn3Click(Sender: TObject);
begin
```

```
form19.Close;
form1.show;
end;

end.
unit Unit20;

interface

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

type
  TForm20 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    DataSource1: TDataSource;
    Table1: TTable;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
```

```
    Label5: TLabel;
    Label6: TLabel;
    Image1: TImage;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form20: TForm20;

implementation

uses Unit1;

{$R *.dfm}

procedure TForm20.BitBtn1Click(Sender: TObject);
begin
form20.Close;
form1.show;
end;

end.
```

```
unit Unit20;

interface

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

type
  TForm20 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    DataSource1: TDataSource;
    Table1: TTable;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Image1: TImage;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
```

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

var
  Form20: TForm20;

implementation

uses Unit1;

{$R *.dfm}

procedure TForm20.BitBtn1Click(Sender: TObject);
begin
form20.Close;
form1.show;
end;

end.
unit Unit21;

interface

uses
```

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

type
 TForm21 = class(TForm)
  Image1: TImage;
  BitBtn6: TBitBtn;
  BitBtn7: TBitBtn;
  BitBtn8: TBitBtn;
  BitBtn9: TBitBtn;
  BitBtn10: TBitBtn;
  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn2Click(Sender: TObject);
  procedure BitBtn3Click(Sender: TObject);
  procedure BitBtn4Click(Sender: TObject);
  procedure BitBtn5Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure BitBtn6Click(Sender: TObject);
  procedure BitBtn9Click(Sender: TObject);
  procedure BitBtn10Click(Sender: TObject);
  procedure BitBtn8Click(Sender: TObject);
  procedure BitBtn7Click(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;
```

```
var
  Form21: TForm21;

implementation

uses Unit2, Unit4, Unit3, Unit5, Unit1;

{$R *.dfm}

procedure TForm21.BitBtn1Click(Sender: TObject);
begin
form2.show;
end;

procedure TForm21.BitBtn2Click(Sender: TObject);
begin
form4.show;
end;

procedure TForm21.BitBtn3Click(Sender: TObject);
begin
form3.show;
end;

procedure TForm21.BitBtn4Click(Sender: TObject);
begin
```

```
form5.show;
end;

procedure TForm21.BitBtn5Click(Sender: TObject);
begin
form21.Close;
form1.show;
end;

procedure TForm21.FormCreate(Sender: TObject);
begin
setwindowlong(form21.Handle,GWL_STYLE,getwindowlong(handle,GWL_STYLE)and not
WS_CAPTION);
height:=clientheight;
end;

procedure TForm21.BitBtn6Click(Sender: TObject);
begin
form3.show;
end;

procedure TForm21.BitBtn9Click(Sender: TObject);
begin
form5.show;
end;

procedure TForm21.BitBtn10Click(Sender: TObject);
```

```
begin
form21.Close;
form1.show;
end;

procedure TForm21.BitBtn8Click(Sender: TObject);
begin
form4.show;
end;

procedure TForm21.BitBtn7Click(Sender: TObject);
begin
form2.show;
end;

end.
unit Unit22;

interface

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

type
  TForm22 = class(TForm)
    Image1: TImage;
```

```
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form22: TForm22;

implementation

uses Unit8, Unit9, Unit1;

{$R *.dfm}

procedure TForm22.BitBtn1Click(Sender: TObject);
begin
form8.show;
end;
```

```
procedure TForm22.BitBtn2Click(Sender: TObject);
begin
form9.show;
end;

procedure TForm22.BitBtn3Click(Sender: TObject);
begin
form22.Close;
form1.show;
end;

procedure TForm22.FormCreate(Sender: TObject);
begin
setwindowlong(form22.Handle,GWL_STYLE,getwindowlong(handle,GWL_STYLE)and not
WS_CAPTION);
height:=clientheight;
end;

end.
unit Unit23;

interface

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

```
type
  TForm23 = class(TForm)
    Panel1: TPanel;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    procedure FormCreate(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form23: TForm23;

implementation

uses Unit1, Unit6, Unit7;

{$R *.dfm}

procedure TForm23.FormCreate(Sender: TObject);
begin
```

```
setwindowlong(form23.Handle,GWL_STYLE,getwindowlong(handle,GWL_STYLE)and not
WS_CAPTION);

height:=clientheight;

end;


procedure TForm23.BitBtn3Click(Sender: TObject);

begin

form23.Close;

form1.show;

end;


procedure TForm23.BitBtn1Click(Sender: TObject);

begin

form6.show;


end;


procedure TForm23.BitBtn2Click(Sender: TObject);

begin

form7.show;

end;


end.

unit Unit24;


interface
```

```
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, Grids, DBGrids, StdCtrls, Buttons, ComCtrls,
  ExtCtrls, Mask, ADODB, jpeg;

type
  TForm24 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Label1: TLabel;
    BitBtn1: TBitBtn;
    DBGrid1: TDBGrid;
    DataSource1: TDataSource;
    DataSource2: TDataSource;
    DateTimePicker2: TDateTimePicker;
    Table1: TTable;
    DBGrid2: TDBGrid;
    DateTimePicker1: TDateTimePicker;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form24: TForm24;
```

```
implementation

{$R *.dfm}

procedure TForm24.BitBtn1Click(Sender: TObject);
var k:integer;
begin
table1.indexname:='date';
table1.SetRange([datetostr(datetimepicker1.date)],[datetostr(datetimepicker2.date)]);
table1.ApplyRange;
end;
end.
unit Unit25;

interface

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

type
  TForm25 = class(TForm)
    Panel2: TPanel;
    Panel3: TPanel;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
```

```
    Table1: TTable;
    Image1: TImage;
    DBNavigator1: TDBNavigator;
    Label5: TLabel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form25: TForm25;

implementation

{$R *.dfm}

end.
unit Unit26;

interface

uses
```

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

type
  TForm26 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    DataSource1: TDataSource;
    Query1: TQuery;
    DBGrid1: TDBGrid;
    Label1: TLabel;
    Edit1: TEdit;
    Panel3: TPanel;
    Image1: TImage;
    procedure Edit1Change(Sender: TObject);

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

var
  Form26: TForm26;

implementation
```

```
{$R *.dfm}


procedure TForm26.Edit1Change(Sender: TObject);
begin
query1.Close;
query1.SQL.Clear;
query1.SQL.Add('select * ');
query1.SQL.add('from prescription where medicine_name like'+#39+(edit1.Text)+'%'+#39);
query1.Open;
end;
end.
```