



NEAR EAST UNIVERSITY

**FACULTY OF ECONOMICS AND
ADMINISTRATIVE SCIENCE**

**Computer Information System
Department**

**GRADUATION PROJECT
2002/2003 SUMMER TERM**

CLINICAL ADMINISTRATION SYSTEM

**SUBMITTED TO : DR. YALÇIN AKÇALI
MISS. NADIRE ÇAVUŞ**

**SUBMITTED BY : YOUSEF ALI ABOU KHURJ
STUDENT NO. : 992103**

NICOSIA



NEU

ACKNOWLEDGEMENTS



First of all I am thanking full to the most gracious "ALLAH" the almighty, who enable me to finalize my project.

Secondly, I would like to award my supervisor ^{Yalcin}~~Prof.~~ Dr Yalcin Akcali and Ms Nadire Cavus for being so operative averring supervise me in this Project, and for his overwhelming and limitless help they had done to me.

Thirdly, I will never over look the encourage I had resaved from all my family, specially my parents, my "father" the best father in this world, and the best sweetest woman in the word "mother", for there supporting me and caring for me, I am thankful for them and my family members from brother and sisters without any exceptions.

Also I would like to thank Raghda for her support to me during my study and I will never forget her longing and pray for me to finish my project. I am so thanking full for u my Love.

Finally I would like to thank the best friends I had met them in Cyprus and in my life like Hamada, Adnan, Hani, Muzher, Amir, Fadi, Ali, Ahamd, Nader, Shquier and of course never forget the efforts of Khalid Al Masri and Reyad Bader and all without any exceptions who I can't express my feeling in such word that for there suggestion and evolution through out completing my project, And also for all of my friends in Lebanon.

I will never forget them

ABSTRACT

Every body wants to be individualized his or herself. Hand he or she really wants to do something extra and different because this is the age of new and more in venture, and recoveries. So when assigned this project, at that time I felt a new scoped and found much more potentials to show my talent and professionalism. I personally thought that this report whose be prove as a foundation store towards the new horizon and would become beneficial to not only for the personals user but also it would help the student to get the information in a system way. Thought this program I tried my level best to minimize the errors rate as low as possible.

This system has been designed in a way that it would work speedier than the normal record keeping system.

Introduction

1.1 Project Overview

The objective of the clinical administration system is to introduce the new concept to manage the clinical side System. The clinical admin Registration project is actually registration management software for helping the User and Doctor staff, those who want register. Update, delete the records of registration management. This software is with related computer program. The patients are registered with our software in order to get benefit. When they are registered users and Doctors can take real benefit from this program and can insert , update , managing all processing, can issue some aspects of some diseases and how can they describe and deal with patients and this software provide easy way to get , registration patient , Medical Report, search of patients who consult anted at clinical

They can use every facility on program. This Software contains following basics

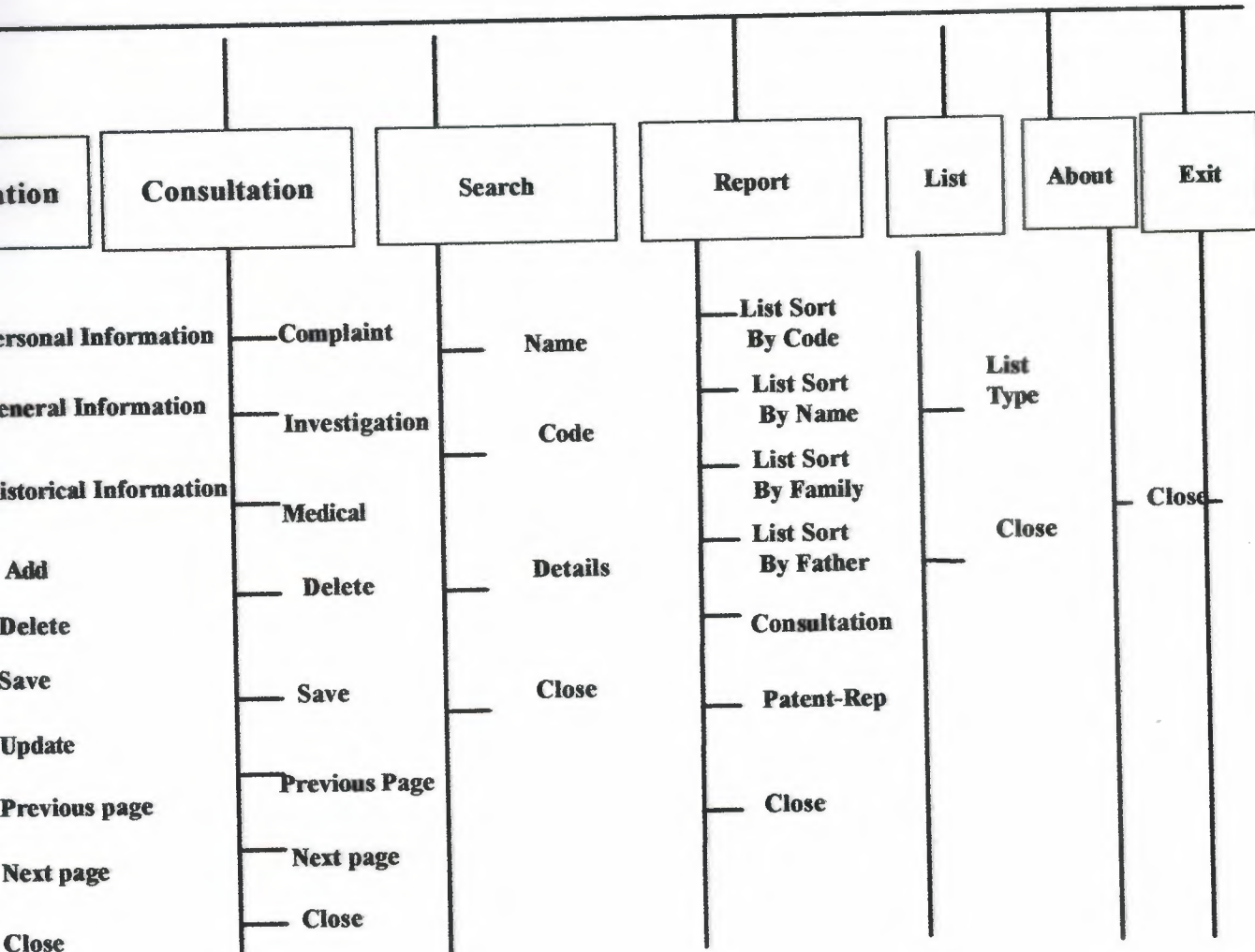
- Designing a database for the Registration and information about Patient
- Designing a database for the Consultation Process
- Designing a database for Complaint process with the description
- Designing a database for investigation process with the description
- Developing an interface for adding patient
- Developing a search mechanism to search with different way.
- Developing an update mechanism to update the information.
- Developing a List mechanism to list with different way.

EXPLANATION OF THE SYSTEM

SYSTEM BLOCK DIAGRAM

Block Diagram

NEU CLINIC



Patience

Field area

Code

Name

Family

Address

City

State

Area

Phone

Photo

Notes

Profession

Group

Physical

Exam

Prescription

Vaccination

Insurance

Referral

Notes

Signature

Date

Level of education

Level of education

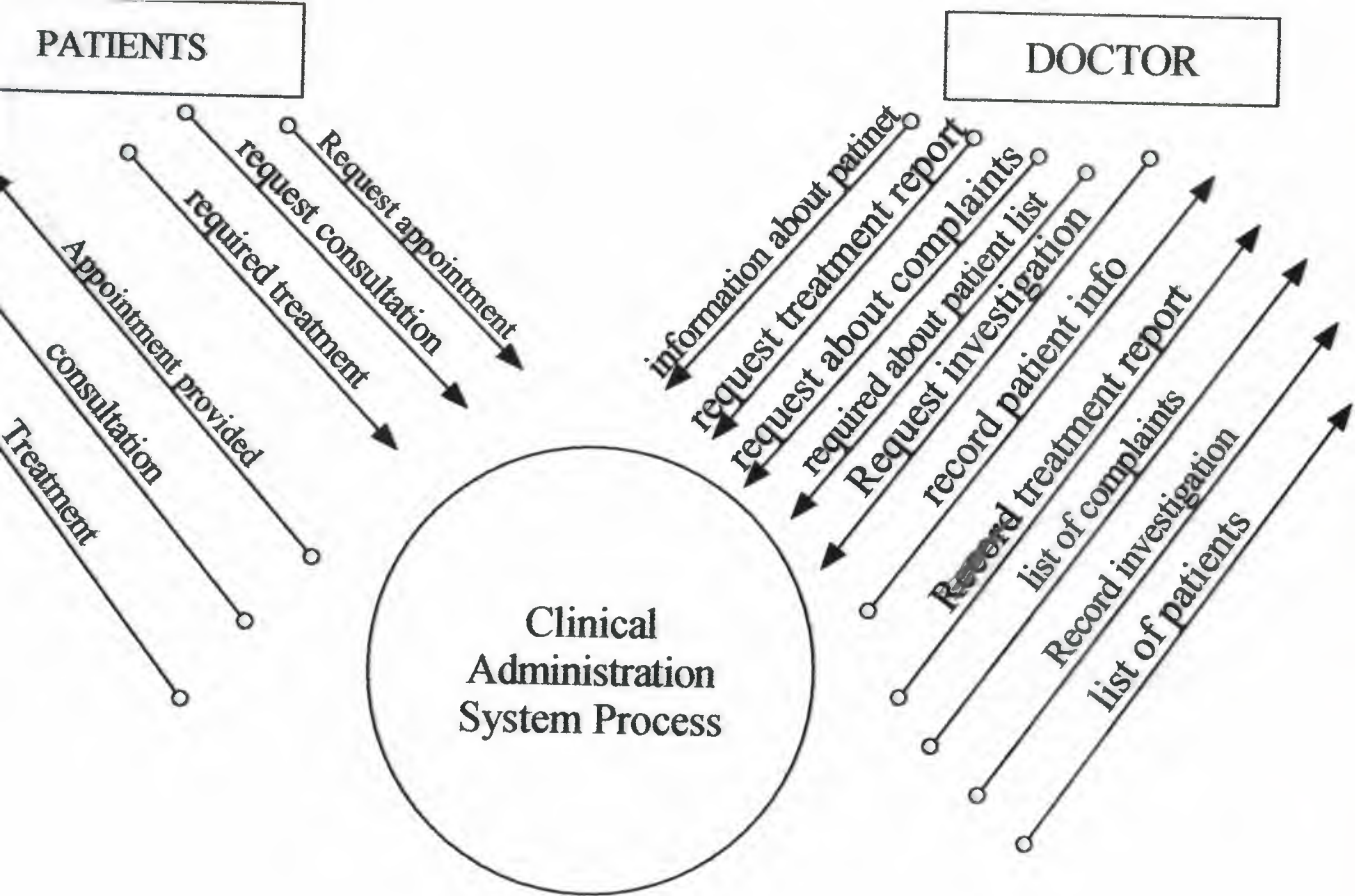
Level of education

DATABASE DESIGN

Patient Table.

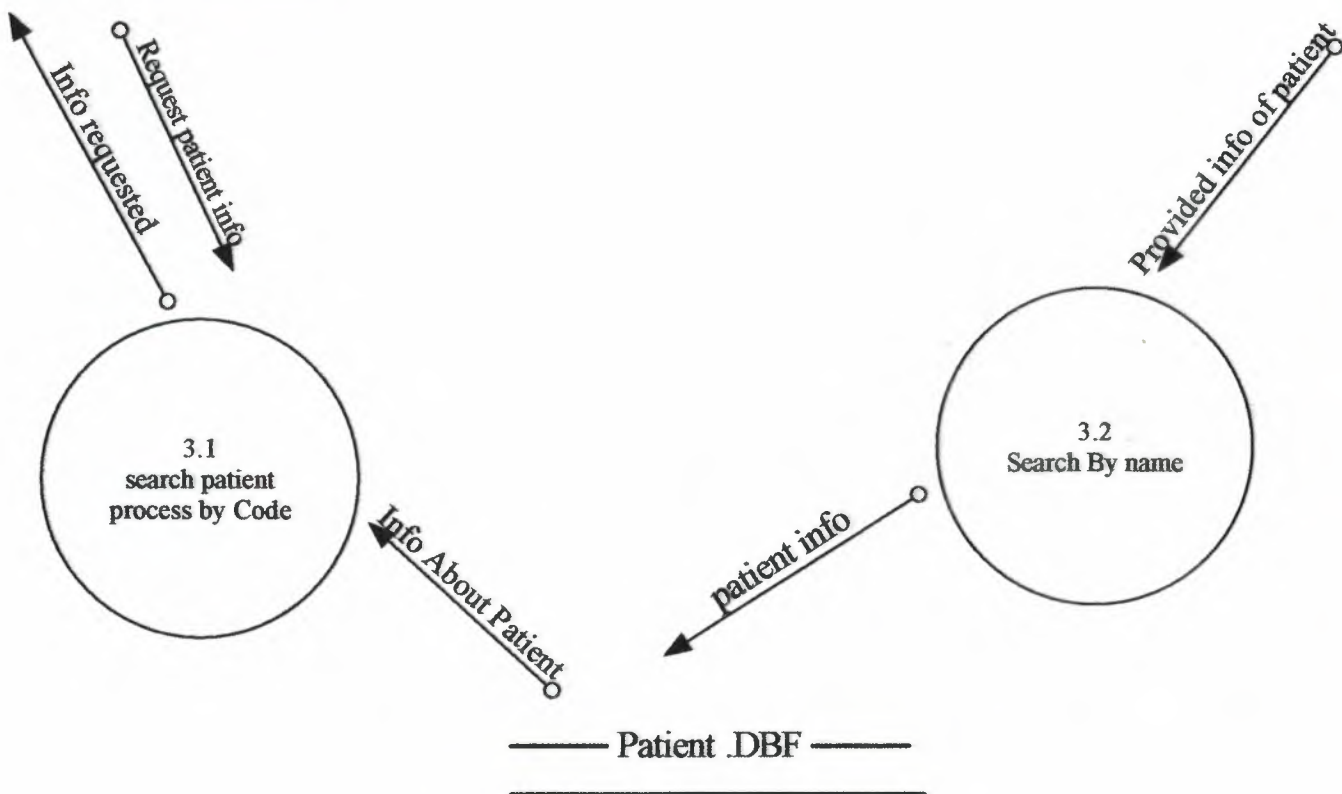
Field name	Type	Size	Key
Code	+		*
Name	A	30	
Family	A	30	
Father	A	30	
DOB	D		
Gender	A	7	
City	A	30	
Street	A	30	
Building	A	30	
Phone	N		
Email	A	30	
Profession	A	30	
Company	A	30	
Mobile	D		
Comments	A	60	
Private	A	15	
Weight	N		
Length	N		
Smoker	L		
Alcohol	L		
Cholesterol	L		
Diabetes	L		
Com classifaction	A	30	
Com description	A	60	
invest classifaction	A	30	
Ivest description	A	60	
Medcail description	A	30	

DATA FLOW DIAGARM

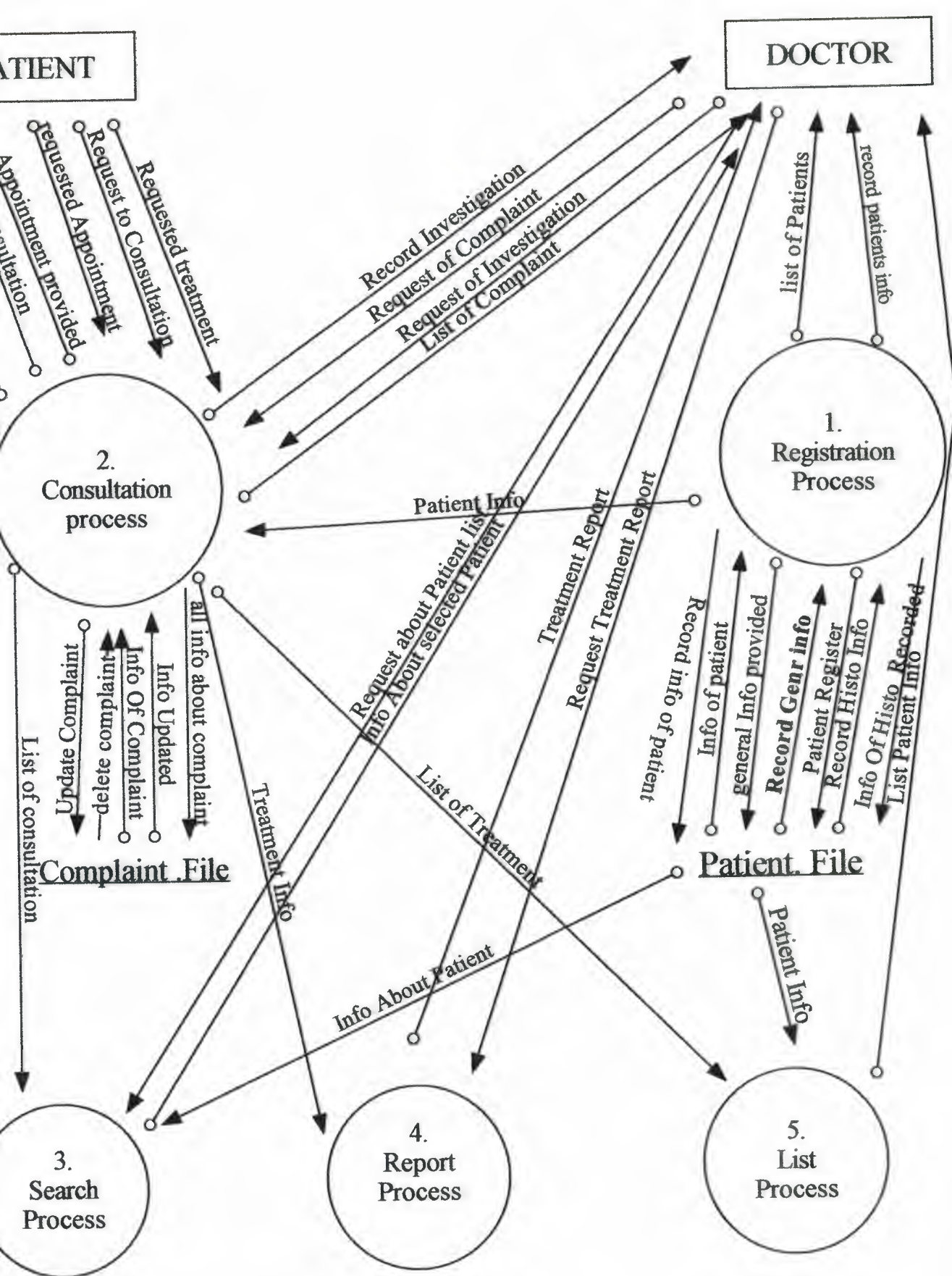


CONTEXT DFD FOR CLINICAL ADMINISTRATION SYSTEM

DOCTOR

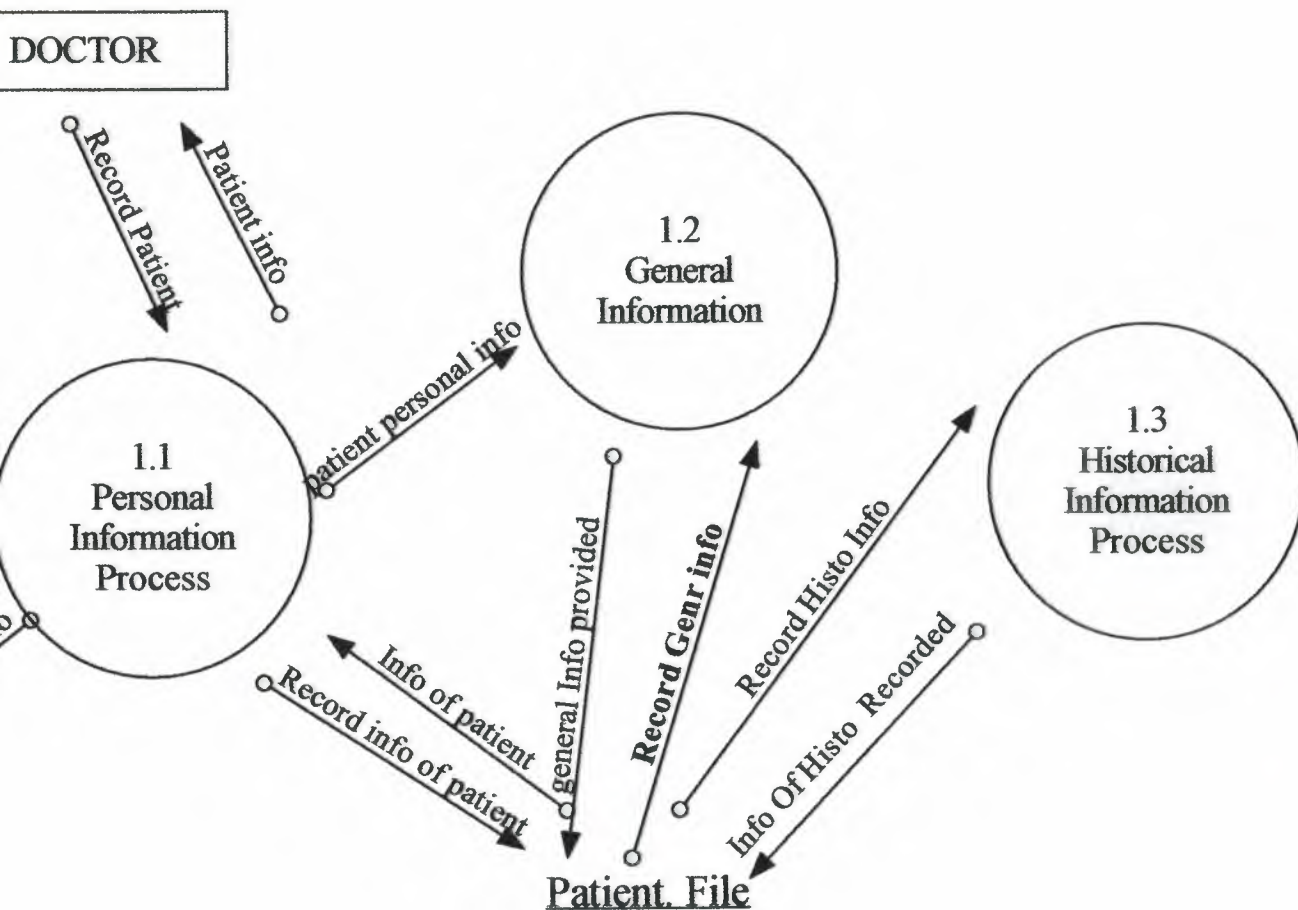


DETAILED DFD FOR SEARCH PROCESS (3)

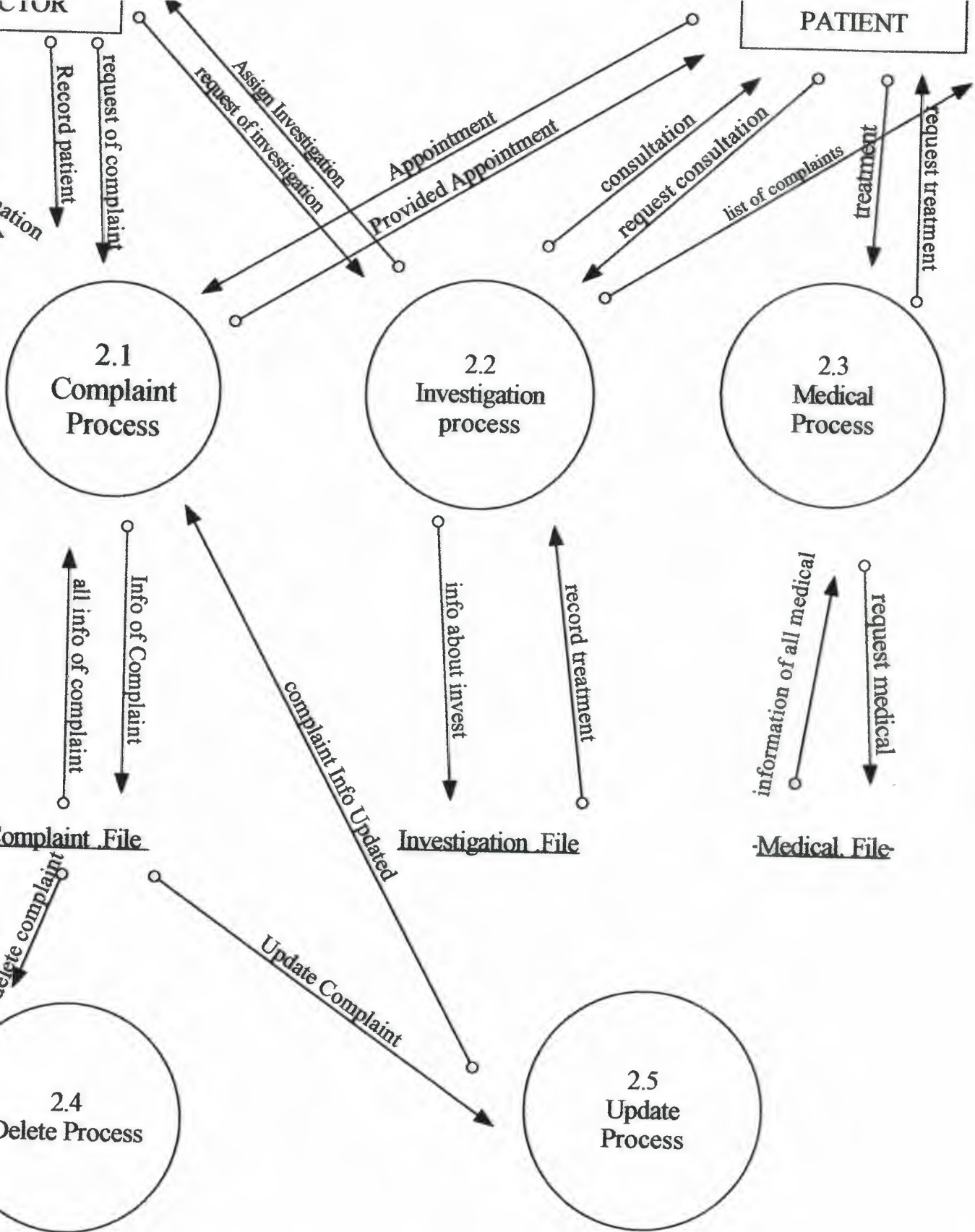


cal File

Top Level DFD For Clinical Administration System



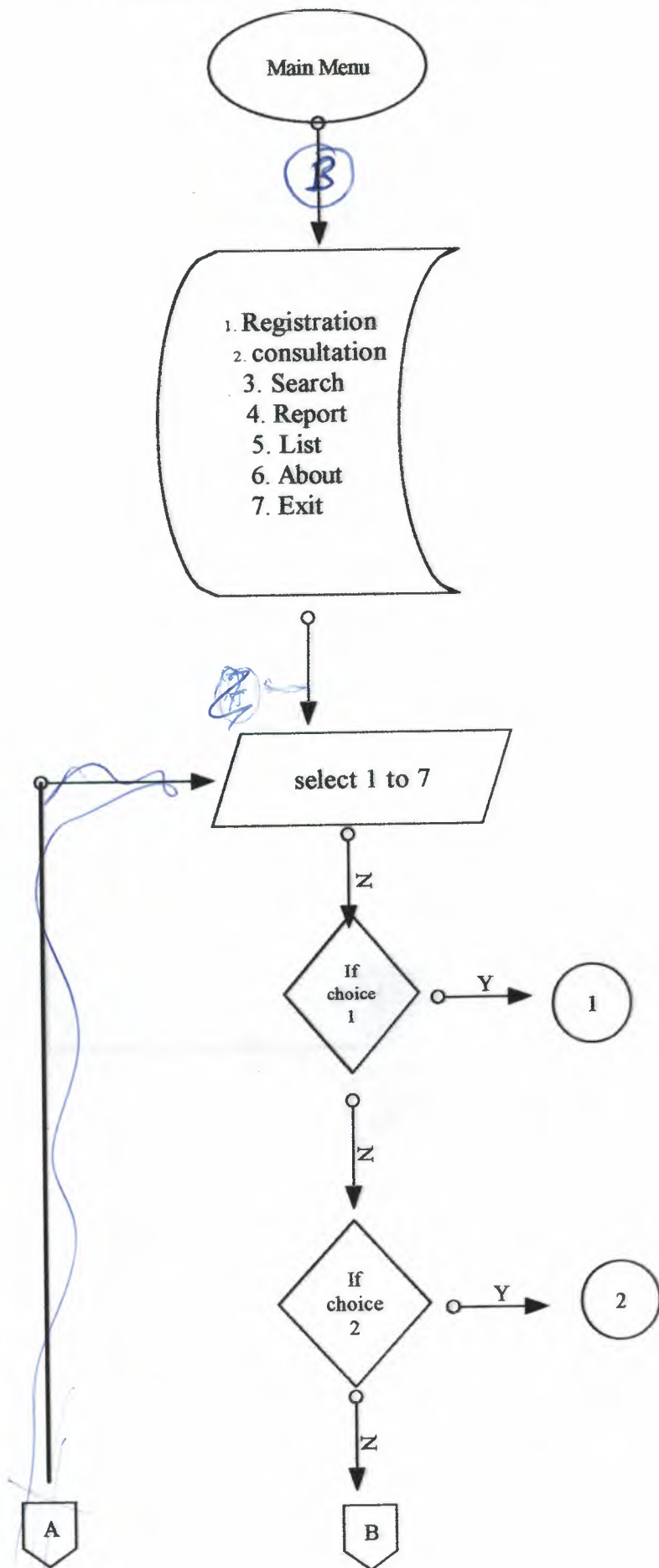
Detailed DFD For Registration Process (1)

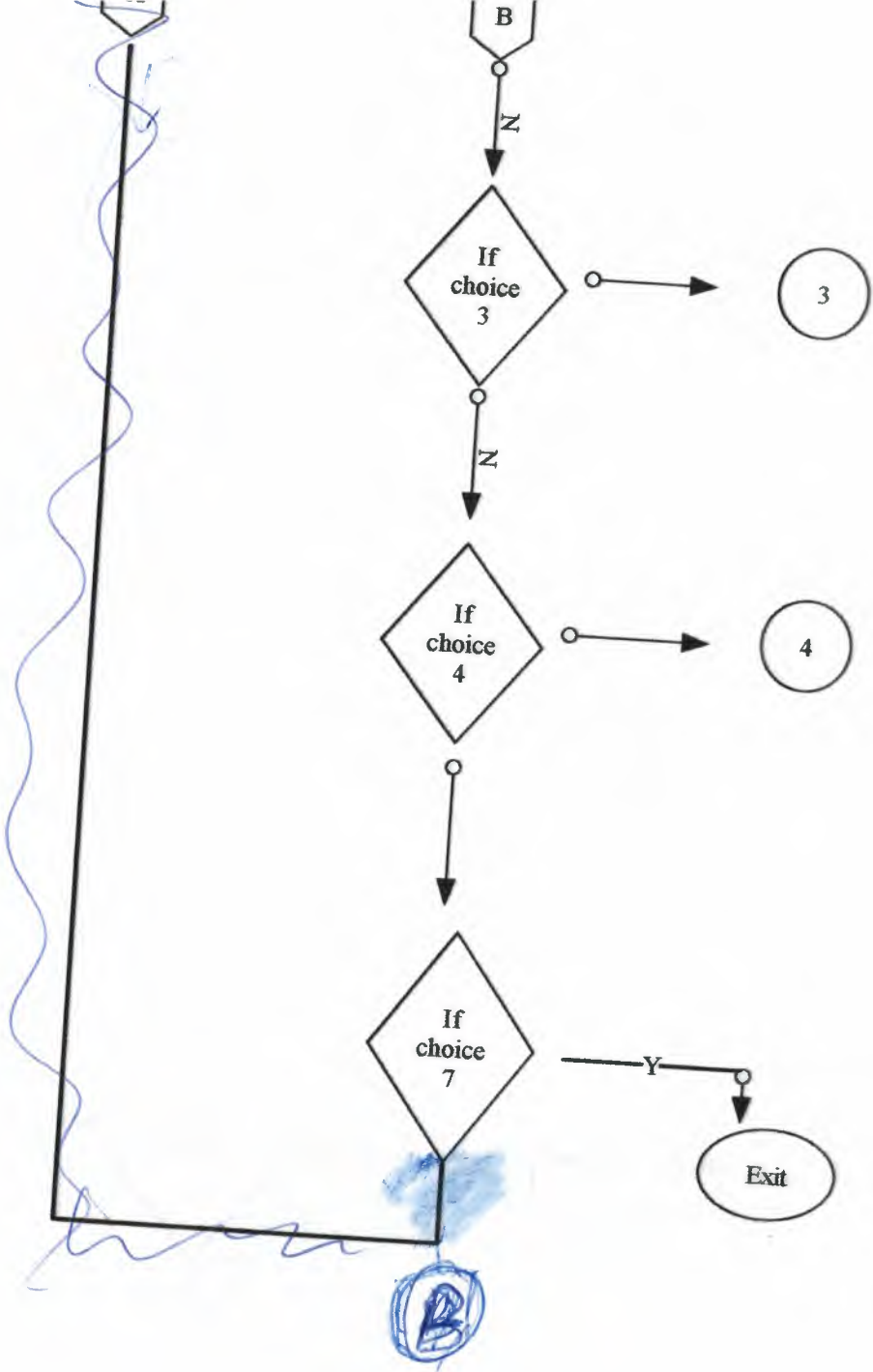


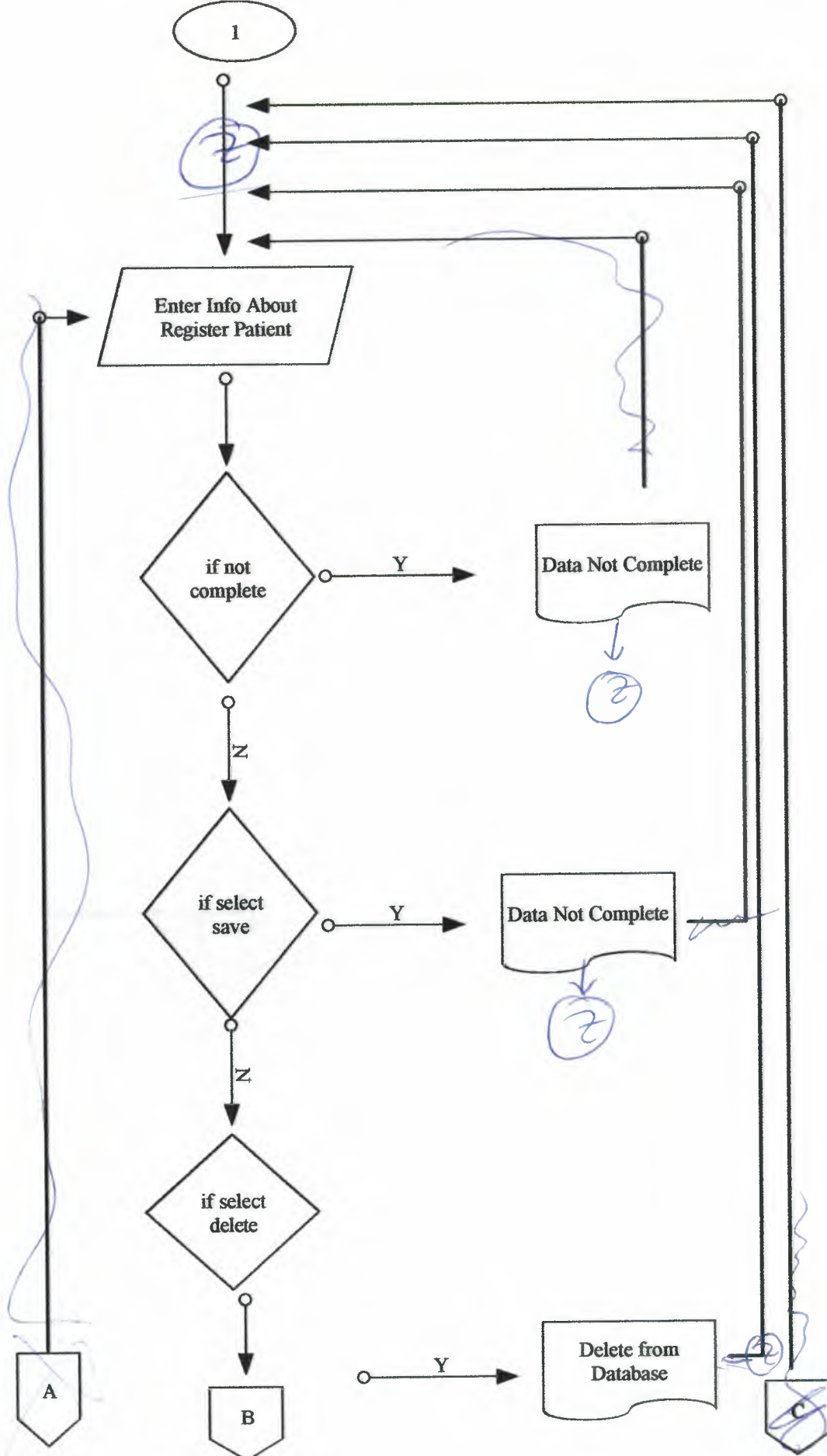
Detailed DFD For Consultation Process (2)

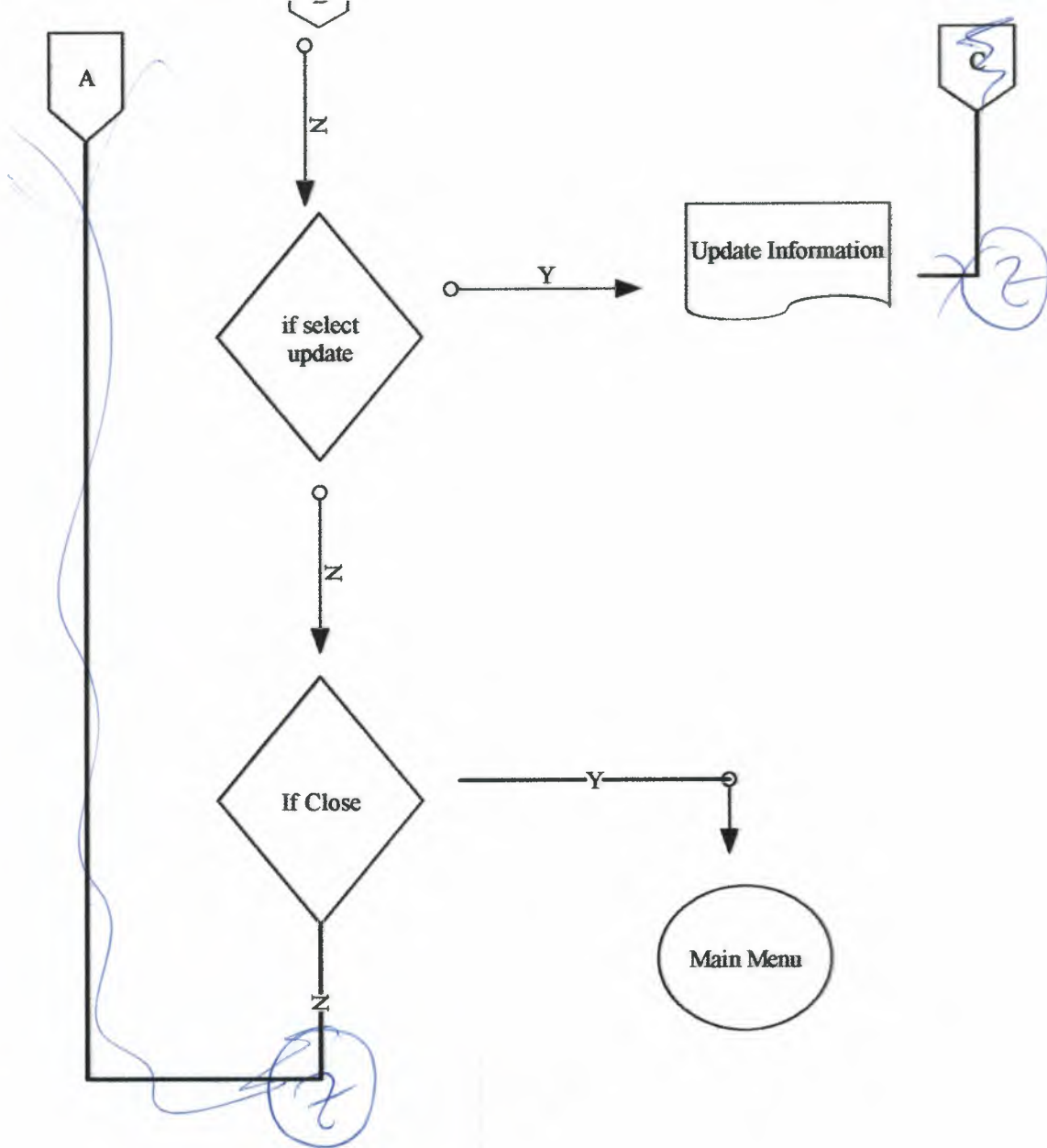
USER FLOW CHART

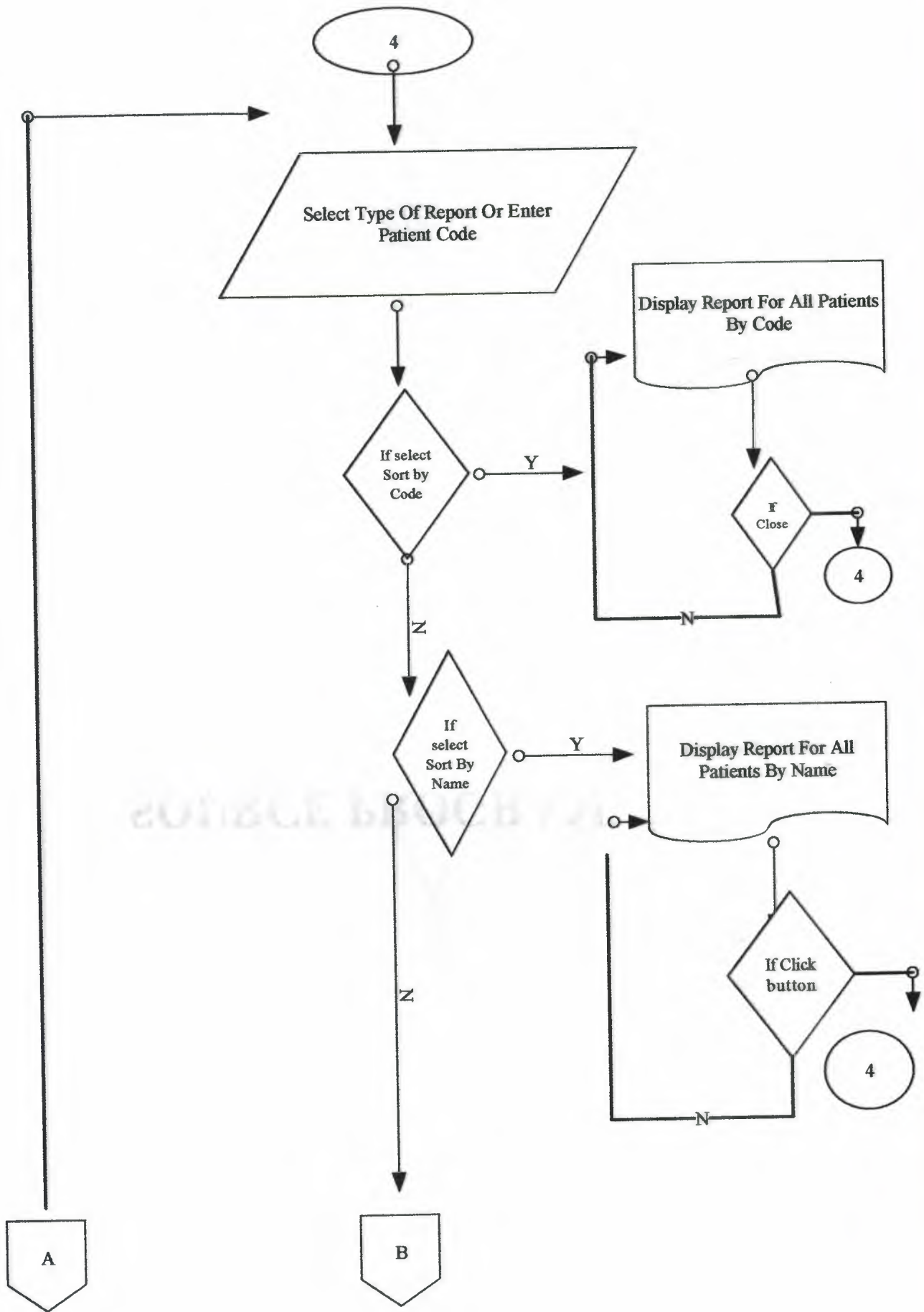
MAIN MENU FLOW CHART

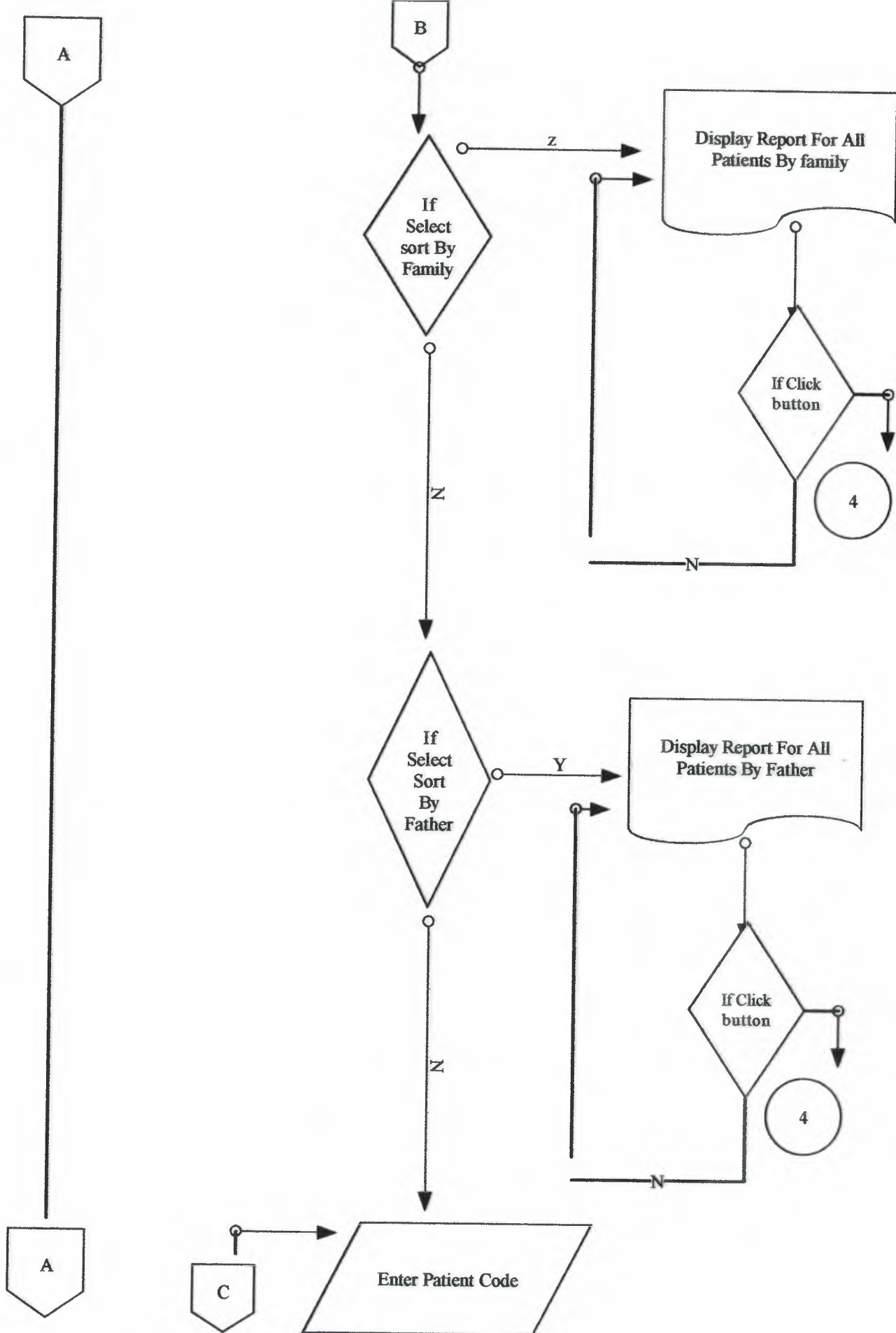


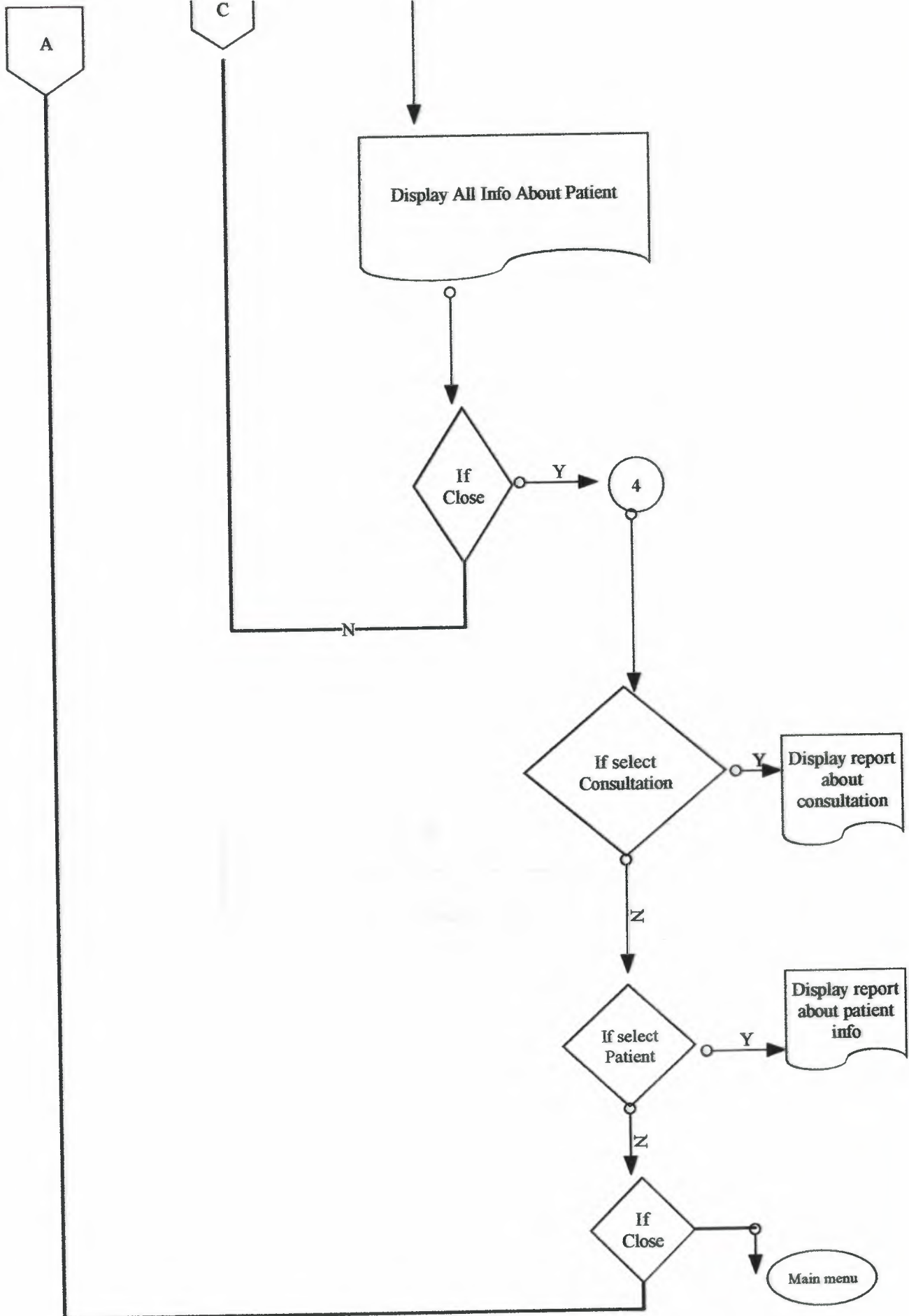




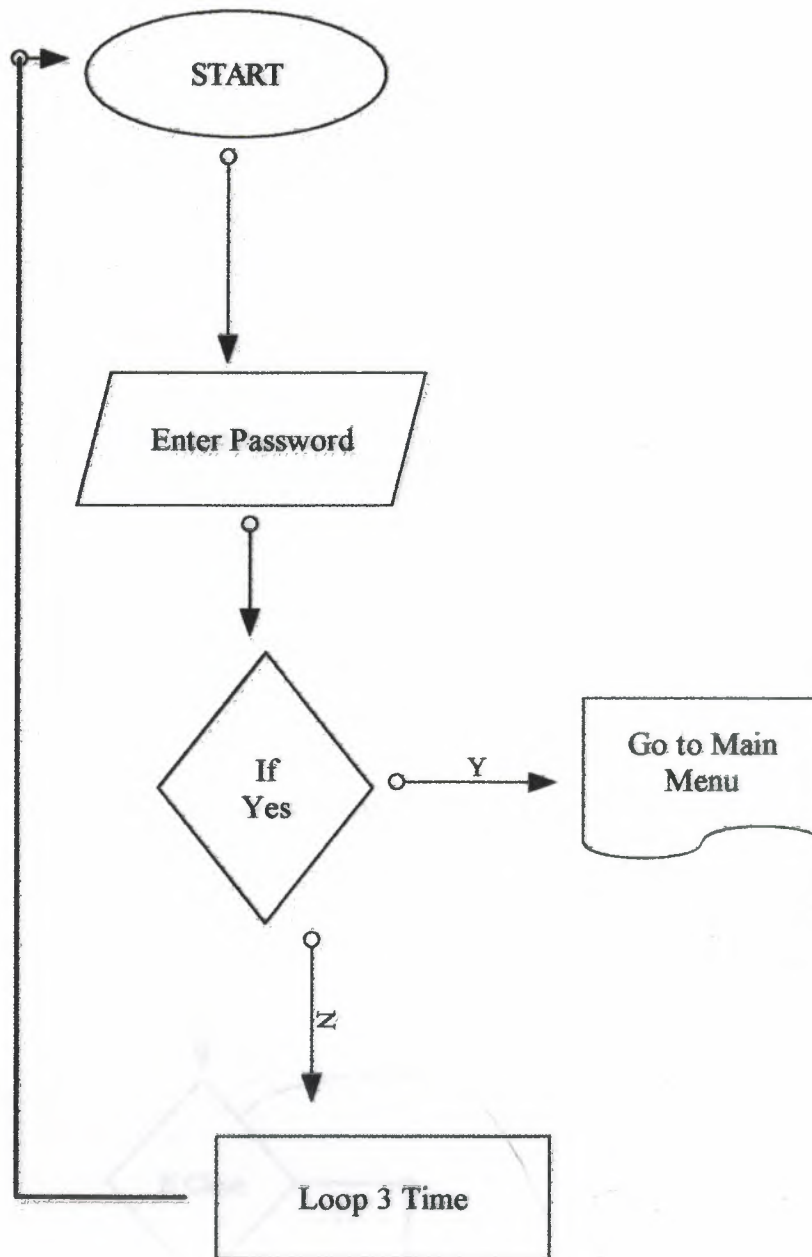




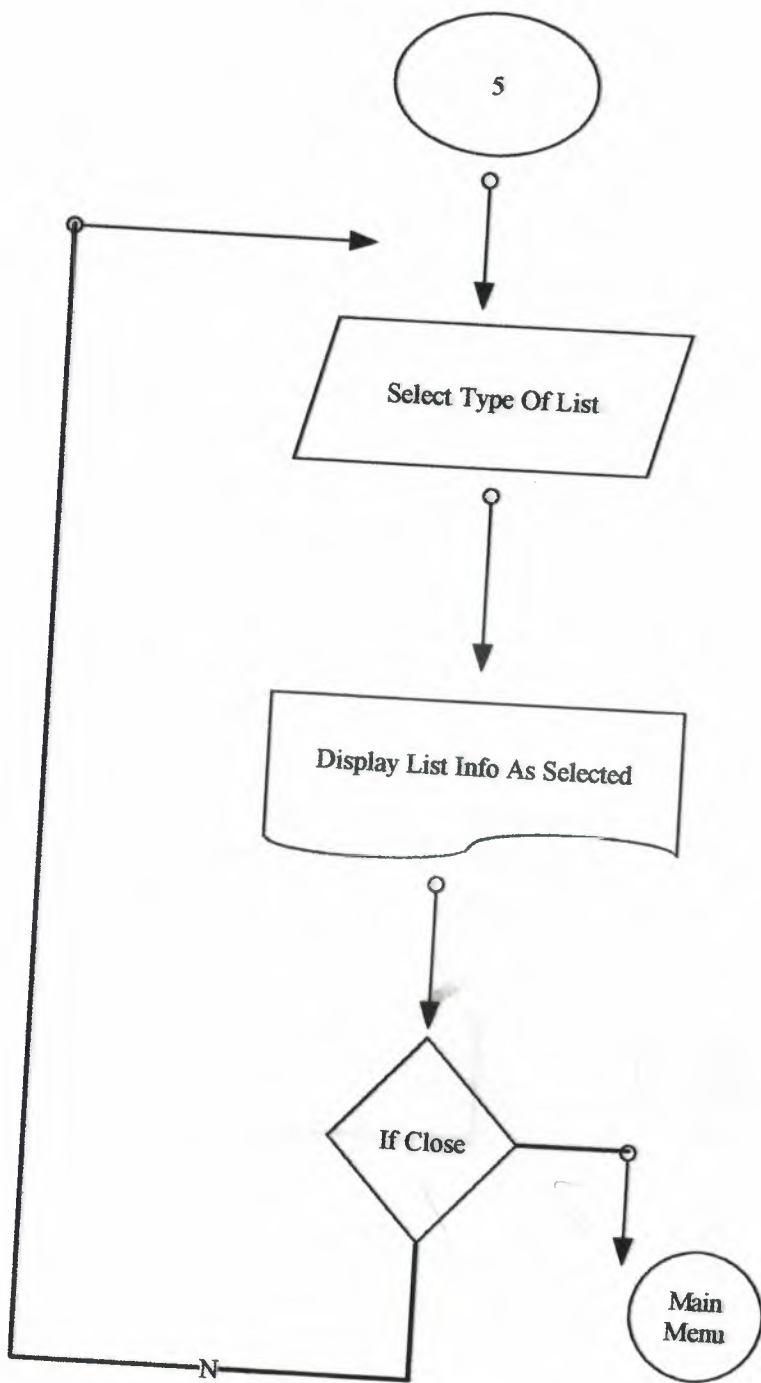




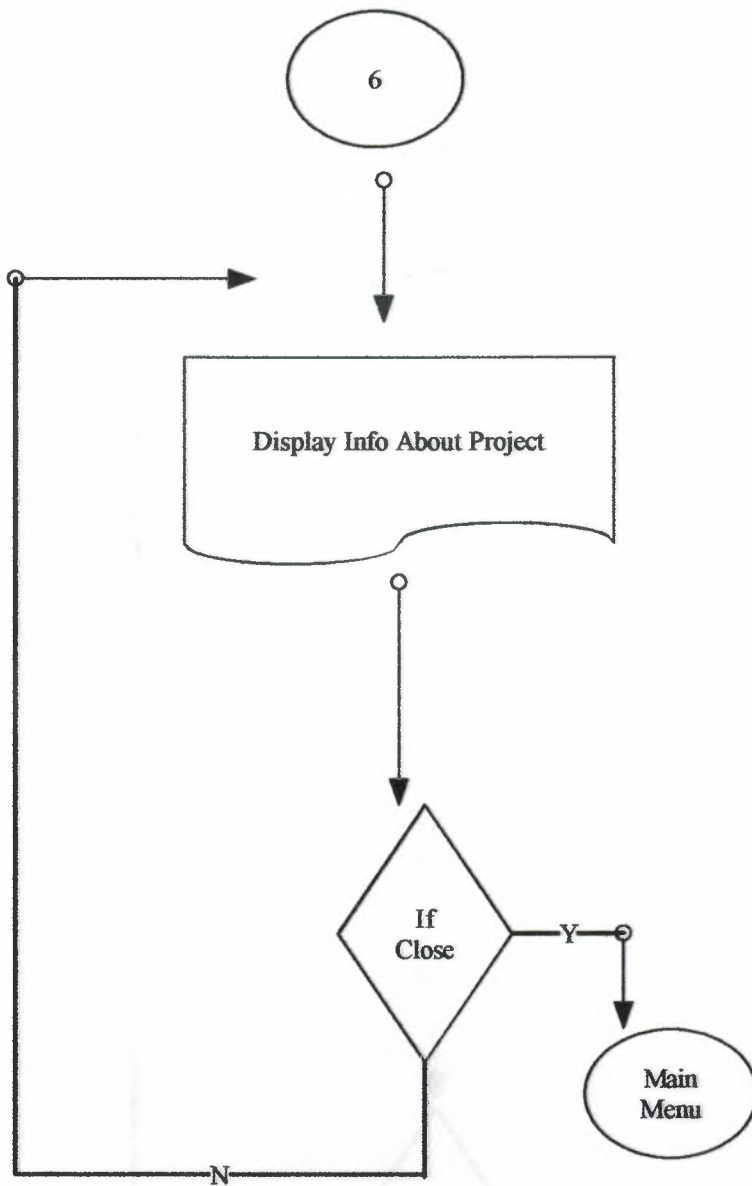
PASSWORD FLOW CHART

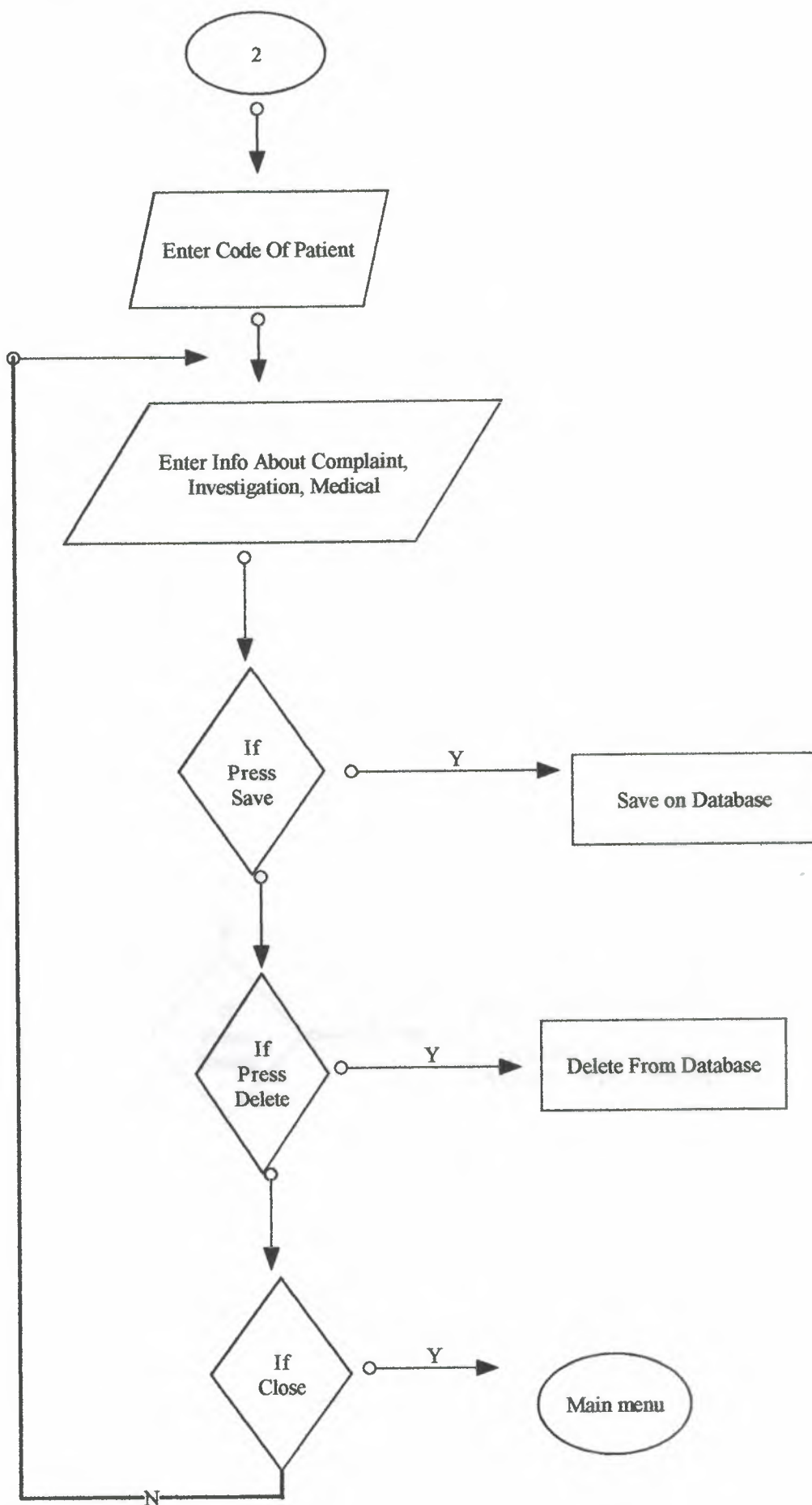


List Flow Chart

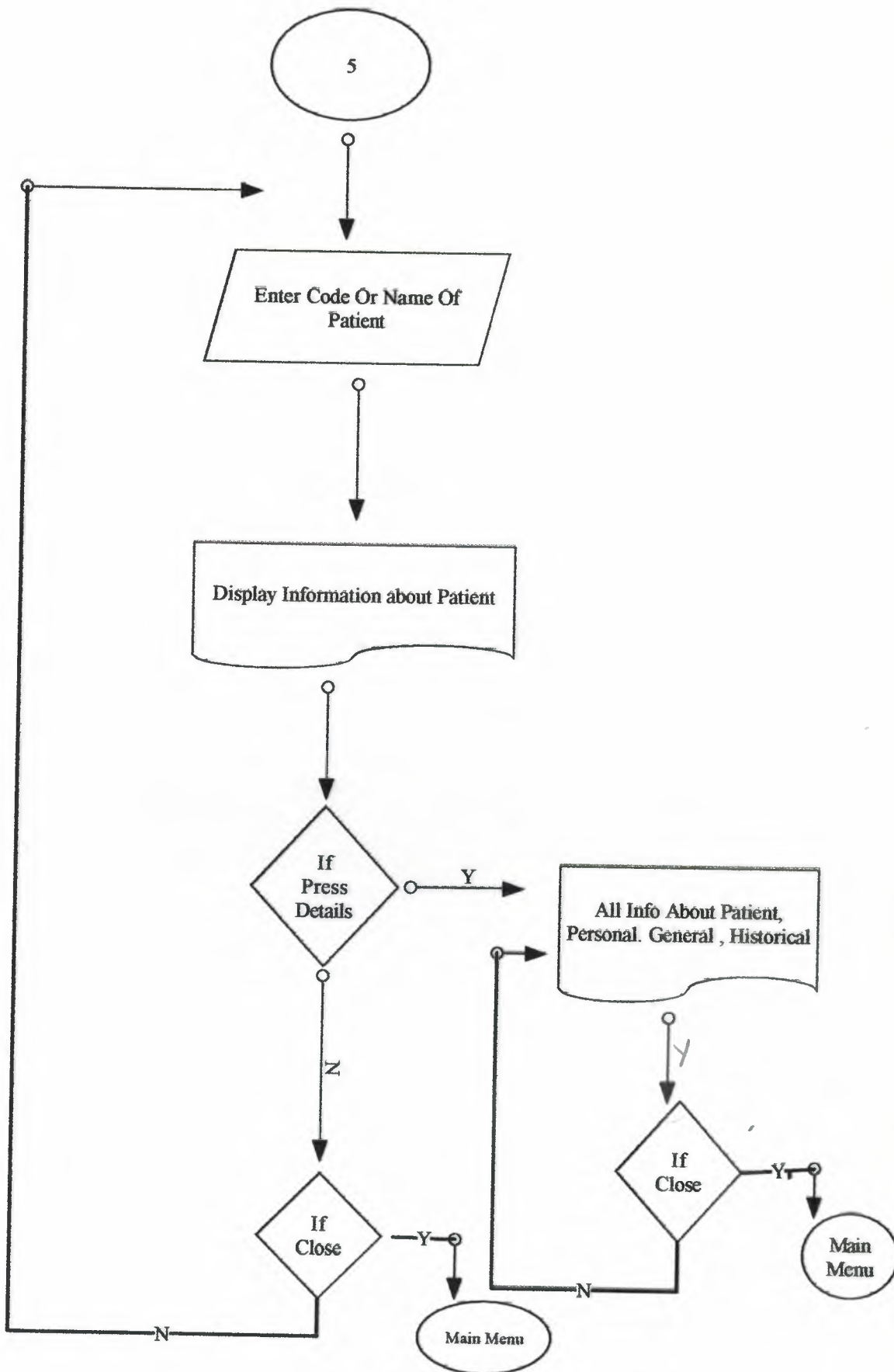


About Flow Chart





Search Flow Chart



SOURCE PROGRAM

1. The purpose of this program is to provide a means for the user to enter and store data in a file.

2. The program is written in Fortran and runs on a CDC 3600 computer.

3. The program is designed to be used by a person who is familiar with the use of a computer.

4. The program is written in a standard Fortran language and should be portable to other computers.

5. The program is written in a standard Fortran language and should be portable to other computers.

6. The program is written in a standard Fortran language and should be portable to other computers.

7. The program is written in a standard Fortran language and should be portable to other computers.

8. The program is written in a standard Fortran language and should be portable to other computers.

9. The program is written in a standard Fortran language and should be portable to other computers.

10. The program is written in a standard Fortran language and should be portable to other computers.

AUTHENTICATION SOURCE



Authentication_u;

face

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Logs, DB, DBTables, StdCtrls, Buttons, ExtCtrls, jpeg;

```
Authentication_frm = class(TForm)
edit1: TEdit;
edit2: TEdit;
name: TLabel;
label1: TLabel;
BitBtn1: TBitBtn;
table1: TTable;
label2: TLabel;
shape1: TShape;
shape2: TShape;
label3: TLabel;
label4: TLabel;
label5: TLabel;
label6: TLabel;
image1: TImage;
procedure BitBtn1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
    Private declarations
public
    Public declarations
end;
```

```
Authentication_frm: TAuthentication_frm;
Count: integer;
```

Implementation

Menu_u;

(.dfm)

```
procedure TAuthentication_frm.BitBtn1Click(Sender: TObject);
begin
    table1.Filter:='name='' + edit1.Text + '' and password='' + edit2.Text + ''';
    table1.Filtered:=true;
```

```
if table1.RecordCount>0 then
begin
    Menu_frm.Show;
```

```
Authentication_frm.Hide;
end
else if count=3 then
    application.Terminate;

inc(count);
```

```
;
cedure TAuthentication_frm.FormCreate(Sender: TObject);
in
    count:=1;
    label3.Caption:=datetostr(date);
    label4.Caption:=timetostr(time());
;

cedure TAuthentication_frm.FormClose(Sender: TObject;
var Action: TCloseAction);
in
    application.Terminate;
```

MAIN MENU

Menu_u;

Face

ows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
ogs, StdCtrls, Buttons, jpeg, ExtCtrls;

```
Menu_frm = class(TForm)
  Panel1: TPanel;
  BitBtn1: TBitBtn;
  BitBtn2: TBitBtn;
  BitBtn3: TBitBtn;
  BitBtn4: TBitBtn;
  BitBtn5: TBitBtn;
  BitBtn6: TBitBtn;
  BitBtn7: TBitBtn;
  Panel2: TPanel;
  procedure BitBtn2Click(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn3Click(Sender: TObject);
  procedure BitBtn6Click(Sender: TObject);
  procedure BitBtn5Click(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure BitBtn4Click(Sender: TObject);
  procedure BitBtn7Click(Sender: TObject);
private
  Private declarations }
public
  Public declarations }
;
```

Menu_frm: TMenu_frm;

mentation

Consultation_u, registration_u, Search_name_u, List_u, Starting_u,
ort_Menu_u, unit1;

.dfm)

```
procedure TMenu_frm.BitBtn2Click(Sender: TObject);
begin
  Consultation_frm.show;
  Menu_frm.Hide;
end;
```

```
procedure TMenu_frm.BitBtn1Click(Sender: TObject);
begin
  registration_frm.show;
  Menu_frm.Hide;
end;
```

```
nd;

procedure TMenu_frm.BitBtn3Click(Sender: TObject);
begin
    Search_name_frm.show;
    menu_frm.Hide;
end;

procedure TMenu_frm.BitBtn6Click(Sender: TObject);
begin
    ShowMessage('Thank you For using this Software');
    menu_frm.Close;
end;

procedure TMenu_frm.BitBtn5Click(Sender: TObject);
begin
    List_frm.Show;
    Menu_frm.Hide;
end;

procedure TMenu_frm.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    Starting_frm.Close;
end;

procedure TMenu_frm.BitBtn4Click(Sender: TObject);
begin
    Report_Menu.show;
    Menu_frm.Hide;
end;

procedure TMenu_frm.BitBtn7Click(Sender: TObject);
begin
    form1.show;
end;

nd.
```


REGISTRATION SOURCE

```
unit registration_u;
```

```
interface
```

```
uses
```

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

```
type
```

```
TRegistration_frm = class(TForm)  
  PageControl1: TPageControl;  
  TabSheet1: TTabSheet;  
  TabSheet2: TTabSheet;  
  Next_btn: TBitBtn;  
  TabSheet3: TTabSheet;  
  Table1: TTable;  
  Table1Code: TAutoIncField;  
  Table1Name: TStringField;  
  Table1Family: TStringField;  
  Table1Father: TStringField;  
  Table1DOB: TDateField;  
  Table1Gender: TStringField;  
  Table1City: TStringField;  
  Table1Street: TStringField;  
  Table1Bulding: TStringField;  
  Table1Phone: TFloatField;  
  Table1Email: TStringField;  
  Table1Proffrsson: TStringField;  
  Table1Company: TStringField;  
  Table1Mobile: TFloatField;  
  Table1Comments: TStringField;  
  Table1Private: TStringField;  
  d: TFloatField;  
  Table1Length: TFloatField;  
  Table1Smoker: TBooleanField;  
  Table1Alcohol: TBooleanField;  
  Table1Cholesterol: TBooleanField;  
  Table1Diabties: TBooleanField;  
  Table1Com_classification: TStringField;  
  Table1Com_description: TStringField;  
  Table1Invest_classification: TStringField;  
  Table1Invest_description: TStringField;  
  Table1Medical_classification: TStringField;  
  Table1Medical_description: TStringField;  
  DataSource1: TDataSource;  
  GroupBox1: TGroupBox;  
  DBCheckBox1: TDBCheckBox;  
  DBCheckBox2: TDBCheckBox;  
  DBCheckBox3: TDBCheckBox;  
  DBCheckBox4: TDBCheckBox;  
  GroupBox2: TGroupBox;  
  DBEdit17: TDBEdit;  
  DBEdit16: TDBEdit;
```

```

Label17: TLabel;
Label18: TLabel;
GroupBox3: TGroupBox;
DBMemo1: TDBMemo;
Label16: TLabel;
Label15: TLabel;
DBEdit15: TDBEdit;
DBEdit14: TDBEdit;
Label14: TLabel;
Label13: TLabel;
DBEdit13: TDBEdit;
GroupBox4: TGroupBox;
DBEdit12: TDBEdit;
Label12: TLabel;
DBEdit11: TDBEdit;
Label11: TLabel;
DBEdit10: TDBEdit;
Label10: TLabel;
DBEdit9: TDBEdit;
Label9: TLabel;
DBEdit8: TDBEdit;
Label8: TLabel;
GroupBox5: TGroupBox;
Label7: TLabel;
Label6: TLabel;
DBEdit5: TDBEdit;
Label5: TLabel;
DBEdit4: TDBEdit;
Label4: TLabel;
DBEdit3: TDBEdit;
Label3: TLabel;
DBEdit2: TDBEdit;
Label2: TLabel;
Label1: TLabel;
DBEdit1: TDBEdit;
ADD_btn: TBitBtn;
Delete_btn: TBitBtn;
Save_btn: TBitBtn;
Previous_btn: TBitBtn;
DBComboBox1: TDBComboBox;
DBComboBox2: TDBComboBox;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
Button1: TButton;
Table1Date_ofday: TDateField;
DBEdit6: TDBEdit;
Label19: TLabel;
procedure Next_btnClick(Sender: TObject);
procedure ADD_btnClick(Sender: TObject);
procedure Delete_btnClick(Sender: TObject);
procedure Save_btnClick(Sender: TObject);
procedure Previous_btnClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormCreate(Sender: TObject);
procedure TabSheet2Show(Sender: TObject);
procedure TabSheet3Show(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);

```

```

procedure BitBtn2Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure DBEdit2Change(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Registration_frm: TRegistration_frm;

implementation

uses Menu_u, Search_name_u;

{$R *.dfm}

procedure TRegistration_frm.Next_btnClick(Sender: TObject);
begin
  if PageControl1.ActivePageIndex<2 then
    PageControl1.SelectNextPage(true,true);
end;

procedure TRegistration_frm.ADD_btnClick(Sender: TObject);
begin
  table1.Insert;
end;

procedure TRegistration_frm.Delete_btnClick(Sender: TObject);
begin
  if MessageDlg('A Patient record is about to be deleted,'+#13+'Are you sure?'
    ,
    mtConfirmation, [mbYes, mbNo], 0) = mrYes then
    table1.Delete;
end;

procedure TRegistration_frm.Save_btnClick(Sender: TObject);
begin
  try
    table1.Edit;
    table1.Post;
    showmessage('Data have been saved');
  except
    showmessage('record already exists');
  end;
end;

procedure TRegistration_frm.Previous_btnClick(Sender: TObject);
begin
  if PageControl1.ActivePageIndex<>0 then
    PageControl1.SelectNextPage(false,true);
end;

procedure TRegistration_frm.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin

```

```

    Menu_frm.show;
end;

procedure TRegistration_frm.FormCreate(Sender: TObject);
begin
    table1.Insert;
end;

procedure TRegistration_frm.TabSheet2Show(Sender: TObject);
begin
    DBEdit8.SetFocus;
end;

procedure TRegistration_frm.TabSheet3Show(Sender: TObject);
begin
    DBEdit16.SetFocus;
end;

procedure TRegistration_frm.BitBtn1Click(Sender: TObject);
var
    NewString:string;
    ClickedOK:Boolean;
begin
    ClickedOK := InputQuery('Input Box', ' Code', NewString);
    if ClickedOK = true then
        begin
            try
                strtoint(NewString);
                table1.Filter:='code=' + NewString;
                table1.Filtered:=true;
            except
                showmessage('Code must be number');
            end ;
        end;
    //
end;

procedure TRegistration_frm.BitBtn2Click(Sender: TObject);
begin
    Registration_frm.Close;

end;

procedure TRegistration_frm.Button1Click(Sender: TObject);
begin
    Registration_frm.Close;
    Search_name_u.Search_name_frm.Show;
end;

procedure TRegistration_frm.DBEdit2Change(Sender: TObject);
begin
    if dbedit2.Text= '1' then
        begin
            showmessage('error');
            dbedit2.Clear;
        end;
    end;

```

CONSULTAION SOURCE

```
unit Consultation_rep_u;

interface

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

type
  TConsultation_rep = class(TForm)
    QuickRep1: TQuickRep;
    QRBand1: TQRBand;
    Table1: TTable;
    QRShape1: TQRShape;
    QRShape2: TQRShape;
    QRLabel23: TQRLabel;
    QRDBText1: TQRDBText;
    QRLabel8: TQRLabel;
    QRLabel9: TQRLabel;
    QRLabel10: TQRLabel;
    QRLabel11: TQRLabel;
    QRLabel12: TQRLabel;
    QRLabel13: TQRLabel;
    QRLabel14: TQRLabel;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
    QRDBText5: TQRDBText;
    QRDBText6: TQRDBText;
    QRDBText7: TQRDBText;
    QRDBText8: TQRDBText;
    QRLabel1: TQRLabel;
    QRDBText9: TQRDBText;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRDBText10: TQRDBText;
    QRDBText11: TQRDBText;
    QRDBText12: TQRDBText;
    QRDBText13: TQRDBText;
    QRShape3: TQRShape;
    QRShape4: TQRShape;
    QRShape5: TQRShape;
    QRLabel7: TQRLabel;
    QRLabel15: TQRLabel;
    QRLabel16: TQRLabel;
  private
    { Private declarations }
  public
```



```
    { Public declarations }
end;

var
    Consultation_rep: TConsultation_rep;

implementation

{$R *.dfm}

end.
```

```
procedure TConsultation_rep.Button1Click(Sender: TObject);
begin
    Search_btnClick(Sender: TObject);
end;

procedure TConsultation_rep.Button2Click(Sender: TObject);
begin
    ShowFormClick(Sender: TObject);
end;

procedure TConsultation_rep.Button3Click(Sender: TObject);
begin
    ShowFormClick(Sender: TObject);
end;
```

SEARCH SOURCE

```
unit Search_name_u;

interface

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

type
  TSearch_name_frm = class(TForm)
    DBGrid1: TDBGrid;
    DataSource1: TDataSource;
    Search_btn: TBitBtn;
    Table1: TTable;
    BitBtn1: TBitBtn;
    close_btn: TButton;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Label4: TLabel;
    procedure Edit1Change(Sender: TObject);
    procedure Search_btnClick(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure close_btnClick(Sender: TObject);
    procedure Edit4Change(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Search_name_frm: TSearch_name_frm;

implementation

uses registration_u, Menu_u, Report_Menu_u;

{$R *.dfm}

procedure TSearch_name_frm.Edit1Change(Sender: TObject);
begin
  // table1.Filter:='code='+edit1.Text;
end;
```

```

procedure TSearch_name_frm.Search_btnClick(Sender: TObject);
var
  str, str2:string;
  name, family, father:string;
  i:integer;
begin
  if edit1.Text<>'' then
  begin
    name:='name='' +edit1.Text+'*'';
    str:=name+' and';
  end;

  if edit2.Text<>'' then
  begin
    family:='family='' +edit2.Text+'*'';
    str:=str+' '+family+' and';
  end;

  if edit3.Text<>'' then
  begin
    father:='father='' +edit3.Text+'*'';
    str:=str+' '+father+' and';
  end;
  setlength(str2, length(str)-4);

  if Length(str)>4 then
  for i:=1 to (Length(str)-4) do
    str2[i]:=str[i];

  table1.Filter:=str2;
  table1.Filtered:=true;

end;

```

```

procedure TSearch_name_frm.BitBtn1Click(Sender: TObject);
begin
  Registration_frm.Table1.Filter:='code='+table1.IndexFields[0].AsString;
  Registration_frm.Table1.Filtered:=true;

  if Report_Menu.Visible =false then
  begin
    Registration_frm.show;
    Search_name_frm.Hide;
  end
  else
    Report_Menu.Edit1.Text:=table1.IndexFields[0].AsString;
end;

```

```

procedure TSearch_name_frm.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  if Report_Menu.Visible =false then
    Menu_frm.show;
end;

```

```
procedure TSearch_name_frm.close_btnClick(Sender: TObject);  
begin  
    close;  
end;
```

```
procedure TSearch_name_frm.Edit4Change(Sender: TObject);  
begin  
    if edit4.Text<>' ' then  
    begin  
        table1.Filter:='code='+edit4.Text;  
        table1.Filtered:=true;  
    end  
    else  
        table1.Filtered:=false;  
end;  
  
end.  
  
end.
```

REPORT SOURCE

```
unit Report_Menu_u;

interface

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

type
  TReport_Menu = class(TForm)
    GroupBox1: TGroupBox;
    BitBtn1: TBitBtn;
    BitBtn3: TBitBtn;
    GroupBox2: TGroupBox;
    BitBtn2: TBitBtn;
    BitBtn4: TBitBtn;
    BitBtn5: TBitBtn;
    BitBtn6: TBitBtn;
    Edit1: TEdit;
    BitBtn7: TBitBtn;
    BitBtn8: TBitBtn;
    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 FormClose(Sender: TObject; var Action: TCloseAction);
    procedure BitBtn8Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Report_Menu: TReport_Menu;

implementation

uses Consultation_rep_u, List_rep_u, patient_rep_u, Search_name_u, Menu_u;

{$R *.dfm}

procedure TReport_Menu.BitBtn1Click(Sender: TObject);
begin
  try
    strtoint(edit1.Text);
    Consultation_rep.Table1.Filter:='code='+Edit1.Text;
    Consultation_rep.Table1.Filtered:=true;
    Consultation_rep.QuickRep1.Preview;
  except
```



```

        showmessage('check the patient code value');
    end;
end;

procedure TReport_Menu.BitBtn2Click(Sender: TObject);
var
    str:string;
begin
    str:='select * from patient order by code';
    List_rep.Query1.Close;
    List_rep.Query1.SQL.Clear;
    List_rep.Query1.SQL.Add(str);
    List_rep.Query1.Open;

    List_rep.QuickRep1.Preview;
end;

procedure TReport_Menu.BitBtn3Click(Sender: TObject);
begin
    try
        strtoint(edit1.Text);
        patient_rep.Table1.Filter:='code='+Edit1.Text;
        patient_rep.Table1.Filtered:=true;
        patient_rep.QuickRep1.Preview;
    except
        showmessage('check the patient code value');
    end;
end;

procedure TReport_Menu.BitBtn4Click(Sender: TObject);
var
    str:string;
begin
    str:='select * from patient order by name';
    List_rep.Query1.Close;
    List_rep.Query1.SQL.Clear;
    List_rep.Query1.SQL.Add(str);
    List_rep.Query1.Open;

    List_rep.QuickRep1.Preview;
end;

procedure TReport_Menu.BitBtn5Click(Sender: TObject);
var
    str:string;
begin
    str:='select * from patient order by family';
    List_rep.Query1.Close;
    List_rep.Query1.SQL.Clear;
    List_rep.Query1.SQL.Add(str);
    List_rep.Query1.Open;

    List_rep.QuickRep1.Preview;
end;

procedure TReport_Menu.BitBtn6Click(Sender: TObject);
var

```

```
    str:string;
begin
    str:='select * from patient order by Father';
    List_rep.Query1.Close;
    List_rep.Query1.SQL.Clear;
    List_rep.Query1.SQL.Add(str);
    List_rep.Query1.Open;

    List_rep.QuickRep1.Preview;
end;

procedure TReport_Menu.BitBtn7Click(Sender: TObject);
begin
    Search_name_frm.show;
end;

procedure TReport_Menu.FormClose(Sender: TObject;
    var Action: TCloseAction);
begin
    Menu_frm.show;
end;

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

end.
```

LIST SOURCE

```
unit List_rep_u;

interface

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

type
  TList_rep = class(TForm)
    QuickRep1: TQuickRep;
    QRBand1: TQRBand;
    QRBand3: TQRBand;
    QRBand2: TQRBand;
    QRLabel1: TQRLabel;
    QRDBText1: TQRDBText;
    QRLabel2: TQRLabel;
    QRDBText2: TQRDBText;
    QRLabel3: TQRLabel;
    QRDBText3: TQRDBText;
    QRLabel4: TQRLabel;
    QRDBText4: TQRDBText;
    QRLabel5: TQRLabel;
    QRDBText5: TQRDBText;
    QRShapel: TQRShape;
    Query1: TQuery;
    QRShape2: TQRShape;
    QRShape3: TQRShape;
    QRLabel23: TQRLabel;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  List_rep: TList_rep;

implementation

{$R *.dfm}

end.
```

```

unit List_u;

interface

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

type
  TList_frm = class(TForm)
    Query1: TQuery;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    ComboBox1: TComboBox;
    Label1: TLabel;
    BitBtn1: TBitBtn;
    procedure ComboBox1Change(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure CloseClick(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  List_frm: TList_frm;

implementation

uses Menu_u;

{$R *.dfm}

procedure TList_frm.ComboBox1Change(Sender: TObject);
var
  str:string;
begin
  case ComboBox1.ItemIndex of
    0: str:='select * from patient order by code asc';
    1: str:='select * from patient order by name asc';
    2: str:='select * from patient order by family asc';
    3: str:='select * from patient order by father asc';
  end;
  query1.Close;
  query1.SQL.Clear;
  query1.SQL.Add(str);
  query1.Open;

end;

procedure TList_frm.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Menu_frm.show;
end;

```

```
procedure TList_frm.CloseClick(Sender: TObject);
begin
    close;
end;

procedure TList_frm.BitBtn1Click(Sender: TObject);
begin
    close
end;

end.
```


COM CALSIFICATION SOURCE

```
unit com_Clasification_u;

interface

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

type
  Tcom_Clasification_frm = class(TForm)
    DBGrid1: TDBGrid;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure BitBtn4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  com_Clasification_frm: Tcom_Clasification_frm;

implementation
uses
  Consultation_u;

{$R *.dfm}

procedure Tcom_Clasification_frm.BitBtn1Click(Sender: TObject);
begin
  Consultation_frm.table2.Insert;
end;

procedure Tcom_Clasification_frm.BitBtn2Click(Sender: TObject);
begin
  Consultation_frm.Table2.Post;
end;

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

procedure Tcom_Clasification_frm.FormClose(Sender: TObject;
  var Action: TCloseAction);
```

```
begin
    Consultation_frm.Table2.Refresh;
end;

procedure Tcom_Clasification_frm.BitBtn4Click(Sender: TObject);
begin
    Consultation_frm.Table2.Delete;
end;

end.
```

INVI CLASIFICATION SOURCE

```
unit Invi_Clasification_u;

interface

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

type
  TInvi_Clasification_frm = class(TForm)
    DBGrid1: TDBGrid;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Invi_Clasification_frm: TInvi_Clasification_frm;

implementation

uses
  Consultation_u;

{$R *.dfm}

procedure TInvi_Clasification_frm.BitBtn1Click(Sender: TObject);
begin
  Consultation_frm.table3.Insert;
end;

procedure TInvi_Clasification_frm.BitBtn2Click(Sender: TObject);
begin
  Consultation_frm.Table3.Post;
end;

procedure TInvi_Clasification_frm.BitBtn4Click(Sender: TObject);
begin
  Consultation_frm.Table3.Delete;
end;

procedure TInvi_Clasification_frm.BitBtn3Click(Sender: TObject);
```

```
begin
  close
end;

procedure TInvi_Clasification_frm.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  Consultation_frm.Table3.Refresh;
end;

end.
```

MED CLASIFICACION SOURCE

```
unit Med_Clasification_u;

interface

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

type
  TMed_Clasification_frm = class(TForm)
    DBGrid1: TDBGrid;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Med_Clasification_frm: TMed_Clasification_frm;

implementation

uses
  Consultation_u;

{$R *.dfm}

procedure TMed_Clasification_frm.BitBtn1Click(Sender: TObject);
begin
  Consultation_frm.table4.Insert;
end;

procedure TMed_Clasification_frm.BitBtn2Click(Sender: TObject);
begin
  Consultation_frm.Table4.Post;
end;

procedure TMed_Clasification_frm.BitBtn4Click(Sender: TObject);
begin
  Consultation_frm.Table4.Delete;
end;

procedure TMed_Clasification_frm.BitBtn3Click(Sender: TObject);
```



```
begin
    close;
end;
```

```
procedure TMed_Clasification_frm.FormClose(Sender: TObject;
    var Action: TCloseAction);
```

```
begin
    Consultation_frm.Table4.Refresh;
end;
```

```
end.
```

STARTING SOURCE

```
unit Starting_u;

interface

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

type
  TStarting_frm = class(TForm)
    Image1: TImage;
    Timer1: TTimer;
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Starting_frm: TStarting_frm;

implementation

uses Menu_u, Authentication_u;

{$R *.dfm}

procedure TStarting_frm.Timer1Timer(Sender: TObject);
begin
  Authentication_frm.Show;
  Starting_frm.Hide;

  timer1.Enabled:=false;
end;

end.
```

ABOUT SOURCE

```
unit Unit1;

interface

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

type
  TForm1 = class(TForm)
    Image1: TImage;
    Image2: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label14: TLabel;
    Label15: TLabel;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation
  uses Menu_u;
  {$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
  close;
end;

end.
```

PATIENT REP SOURCE

```
unit patient_rep_u;
```

```
interface
```

```
uses
```

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

```
type
```

```
  Tpatient_rep = class(TForm)
```

```
    QuickRep1: TQuickRep;
```

```
    QRBand1: TQRBand;
```

```
    Table1: TTable;
```

```
    QRDBText1: TQRDBText;
```

```
    QRLabel3: TQRLabel;
```

```
    QRLabel4: TQRLabel;
```

```
    QRLabel5: TQRLabel;
```

```
    QRLabel6: TQRLabel;
```

```
    QRLabel7: TQRLabel;
```

```
    QRLabel8: TQRLabel;
```

```
    QRLabel9: TQRLabel;
```

```
    QRLabel10: TQRLabel;
```

```
    QRLabel11: TQRLabel;
```

```
    QRLabel12: TQRLabel;
```

```
    QRLabel13: TQRLabel;
```

```
    QRLabel14: TQRLabel;
```

```
    QRDBText2: TQRDBText;
```

```
    QRDBText3: TQRDBText;
```

```
    QRDBText4: TQRDBText;
```

```
    QRDBText5: TQRDBText;
```

```
    QRDBText6: TQRDBText;
```

```
    QRDBText7: TQRDBText;
```

```
    QRDBText8: TQRDBText;
```

```
    QRDBText9: TQRDBText;
```

```
    QRDBText10: TQRDBText;
```

```
    QRDBText11: TQRDBText;
```

```
    QRLabel1: TQRLabel;
```

```
    QRLabel2: TQRLabel;
```

```
    QRLabel15: TQRLabel;
```

```
    QRLabel16: TQRLabel;
```

```
    QRDBText12: TQRDBText;
```

```
    QRDBText16: TQRDBText;
```

```
    QRDBText13: TQRDBText;
```

```
    QRDBText14: TQRDBText;
```

```
    QRDBText15: TQRDBText;
```

```
    QRLabel17: TQRLabel;
```

```
    QRLabel18: TQRLabel;
```

```
    QRDBText17: TQRDBText;
```

```
    QRDBText18: TQRDBText;
```

```
    QRLabel19: TQRLabel;
```

```
    QRLabel20: TQRLabel;
```

```
    QRLabel21: TQRLabel;
```

```

QRLabel22: TQRLabel;
QRDBText19: TQRDBText;
QRDBText20: TQRDBText;
QRDBText21: TQRDBText;
QRDBText22: TQRDBText;
QRShape1: TQRShape;
QRShape2: TQRShape;
QRLabel23: TQRLabel;
QRShape3: TQRShape;
QRLabel24: TQRLabel;
QRShape4: TQRShape;
QRLabel25: TQRLabel;
QRShape5: TQRShape;
QRShape6: TQRShape;
QRLabel26: TQRLabel;
QRLabel27: TQRLabel;
private
{ Private declarations }
public
{ Public declarations }
end;

var
  patient_rep: Tpatient_rep;

implementation

{$R *.dfm}

end.

```


LIST SOURCE

```
unit List_rep_u;

interface

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

type
  TList_rep = class(TForm)
    QuickRep1: TQuickRep;
    QRBand1: TQRBand;
    QRBand3: TQRBand;
    QRBand2: TQRBand;
    QRLabel1: TQRLabel;
    QRDBText1: TQRDBText;
    QRLabel2: TQRLabel;
    QRDBText2: TQRDBText;
    QRLabel3: TQRLabel;
    QRDBText3: TQRDBText;
    QRLabel4: TQRLabel;
    QRDBText4: TQRDBText;
    QRLabel5: TQRLabel;
    QRDBText5: TQRDBText;
    QRShape1: TQRShape;
    Query1: TQuery;
    QRShape2: TQRShape;
    QRShape3: TQRShape;
    QRLabel23: TQRLabel;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  List_rep: TList_rep;

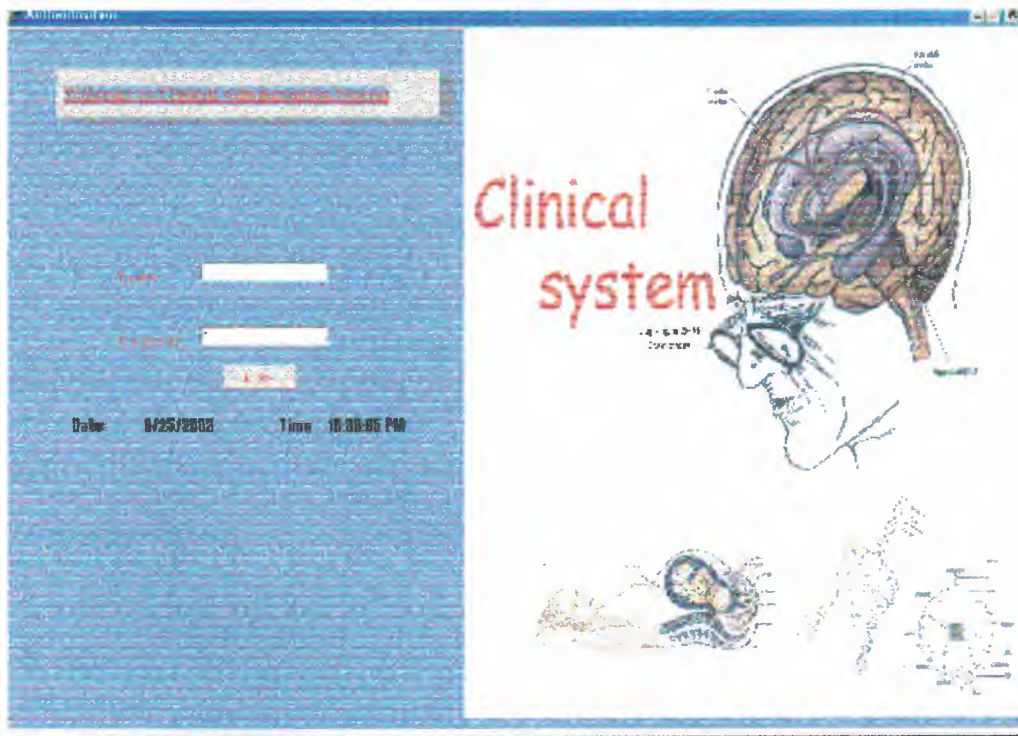
implementation

{$R *.dfm}

end.
```

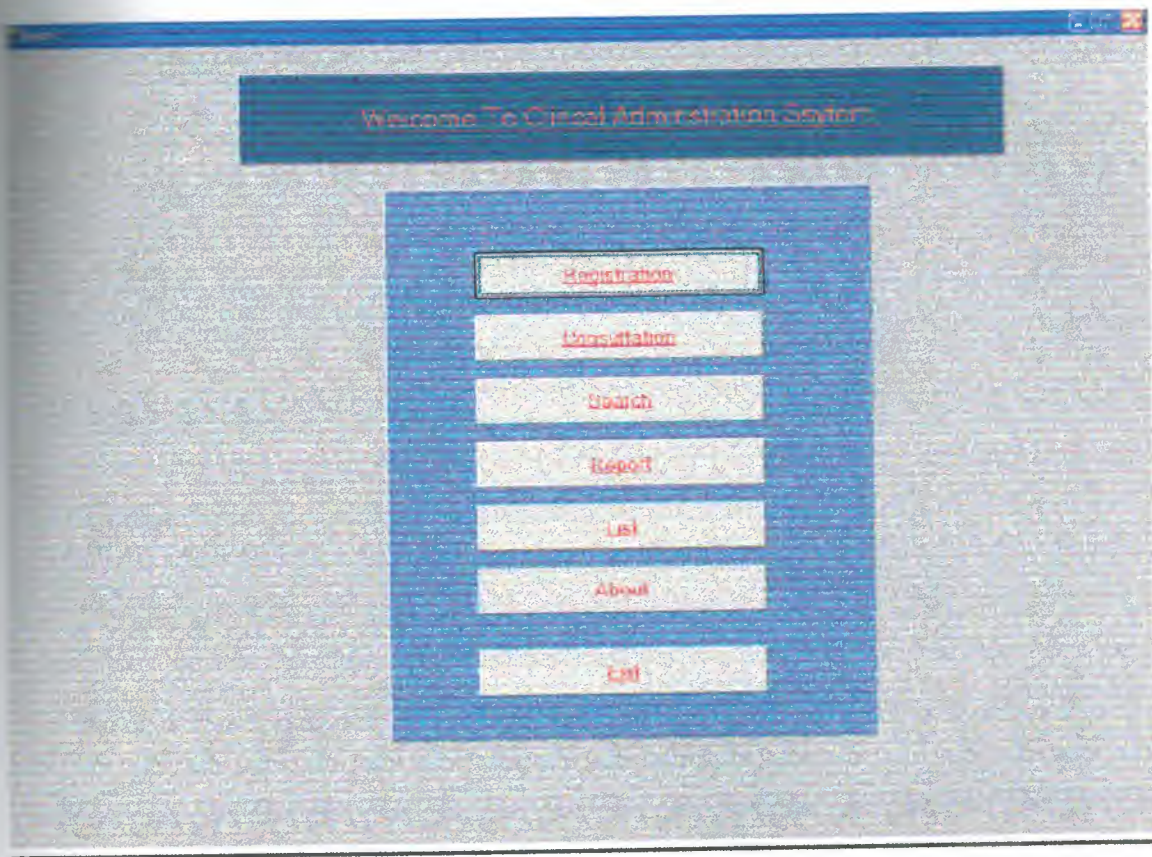
USER MANUAL

Password Menu



This is the form of password process that includes name of user and also the password that enables him/her to enter through program by pressing Enter Button. This system has a security option, Program can terminate if the user enter wrong password for three times.

Main Menu



This is the very basic function of the program. All executive start from here. In this function .it's going to enter the function by the selection of the user. It contains of 7 processes and they are as follows:

- Registration
- Consultation
- Search
- Report
- List
- About me
- Exit

Registration Menu .A

Registration

Personal Information | General Information | Historical Information

Personal Information

Code

Name

Family

Father

DOB

Gender

Private

Date

Close ADD Delete Save Update Previous Page Next Page

Back to search

This is registration form menu. A enables user to add personal information's about patient and allows him/her to save, delete, and update patient records. And backward to another form also here a button that can reach the user to search menu to check and make search for the recorded patients.

Registration Menu .B

The screenshot shows a software window titled "Registration" with a blue background. At the top, there are three tabs: "Personal Information", "General Information", and "Historical Information". The "General Information" tab is currently selected. Below the tabs, there are two main sections: "Address" on the left and "Occupation" on the right. The "Address" section contains five text input fields labeled "City", "Street", "Building", "Phone", and "Email". The "Occupation" section contains three text input fields labeled "Profession", "Company", and "Mobile", followed by a larger text area labeled "Comments". At the bottom of the window, there is a row of buttons: "Close", "ADD", "Delete", "Save", "Update", "Previous Page", and "Next Page". Below this row, there is a button labeled "Back to search".

This is another registration form menu. B enables user to add general information's about patient and allows him/her to save, delete, and update patient records. And backward to another form also here a button that can reach the user to search menu to check and make search for the recorded patients.

Registration Menu .C

The screenshot shows a Windows-style application window titled "Registration". It features three tabs: "Personal Information", "General Information", and "Historical Information". The "Personal Information" tab is active, displaying a "Physical" section with input fields for "Weight" and "Length", and a "Risk Factors" section with checkboxes for "Smoker", "Alcohol", "Cholesterol", and "Diabetes". At the bottom of the window, there is a row of buttons: "Close", "ADD", "Delete", "Save", "Update", "Previous Page", and "Next Page". A "Back to search" button is located at the bottom left.

This is another registration form menu. C enables user to add historical information's about patient and allows him/her to save, delete, and update patient records and then close the form and return back to the main menu to choose another process.

Consultation form .A

The screenshot shows a software window titled "Consultation". Inside, there's a "Personal Information" section with input fields for "Code", "Name", "Family", and "Father". Below this is a tabbed interface with three tabs: "Complaint", "Investigation", and "Medical". The "Complaint" tab is selected, displaying a "Classification" dropdown menu and a large "Description" text area. At the bottom of the window are five buttons: "Close", "Delete", "Save", "Previous Page", and "Next Page".

Here the consultation process contains of three main procedures and here it deals with patient by make consultation for him/ her through compliant, investigation, medical.

And enables user to add, delete and save patient information and this occurs when user inserts patient code above and automatically display patient record comes from Database

COMPLAINT FUCNTION

It contains some kind of complaints recorded in program and giving up by Doctor of what cases may the patient complain and also there is a classification of complaint and description as text box to give some notes for patient by Doctor .

Consultation form .B

Consultation

Personal Information

Code >>

Name Family Father

Complaint Investigation Medical

Classification Tumor Markers >>>

Description

Close Delete Save Previous Page Next Page

Here the consultation process contains of three main procedures and here it deals with patient by make consultation for him/ her through compliant, investigation, medical.

And enables user to add, delete and save patient information and this occurs when user inserts patient code above and automatically display patient record comes from Database

INVESTIGATION FUCNTION

Its something different that in a case of patient complain of something here we can notes up some kind of investigation and also provide them with description given up by Doctor as note.

Consultation form .C

The screenshot shows a software window titled 'Consultation'. It contains several input fields and buttons. At the top, there's a 'Personal Information' section with fields for 'Code', 'Name', 'Family', and 'Father'. Below this, there are three tabs: 'Complaint', 'Investigation', and 'Medical'. The 'Medical' tab is currently selected. Inside the 'Medical' tab, there's a 'Classification' dropdown menu showing 'ANTIVIRALIS' and a '>>>' button. Below that is a large empty text area labeled 'Description'. At the bottom of the window, there are five buttons: 'Close', 'Delete', 'Save', 'Previous Page', and 'Next Page'.

Here the consultation process contains of three main procedures and here it deals with patient by make consultation for him/ her through compliant, investigation, medical.

And enables user to add, delete and save patient information and this occurs when user inserts patient code above and automatically display patient record comes from Database

MEDCAIL FUCNTION

This is what we see and touched in our real life as giving the name and kind of medical to patient and also by describe for the patient some main procedures should follow during his/her medical.

Search Form. A

Code	Name	Family	Father	DOB	Gender	City	Street
3	alid alid	albadar	ahmed	1/1/2001	male		
5	reyad hasan	ali	sabas	3/3/2003	male	gaza	kljlk
8	a	1z	a				

Here the search form that helps user to interact with patients and make search about their information's by name, father, and family and directly when write name patient all information will display for him/her.

Also here a detail button that also reach user to view the information of patient recorded in registration form and can be able to change the date he/she wants.

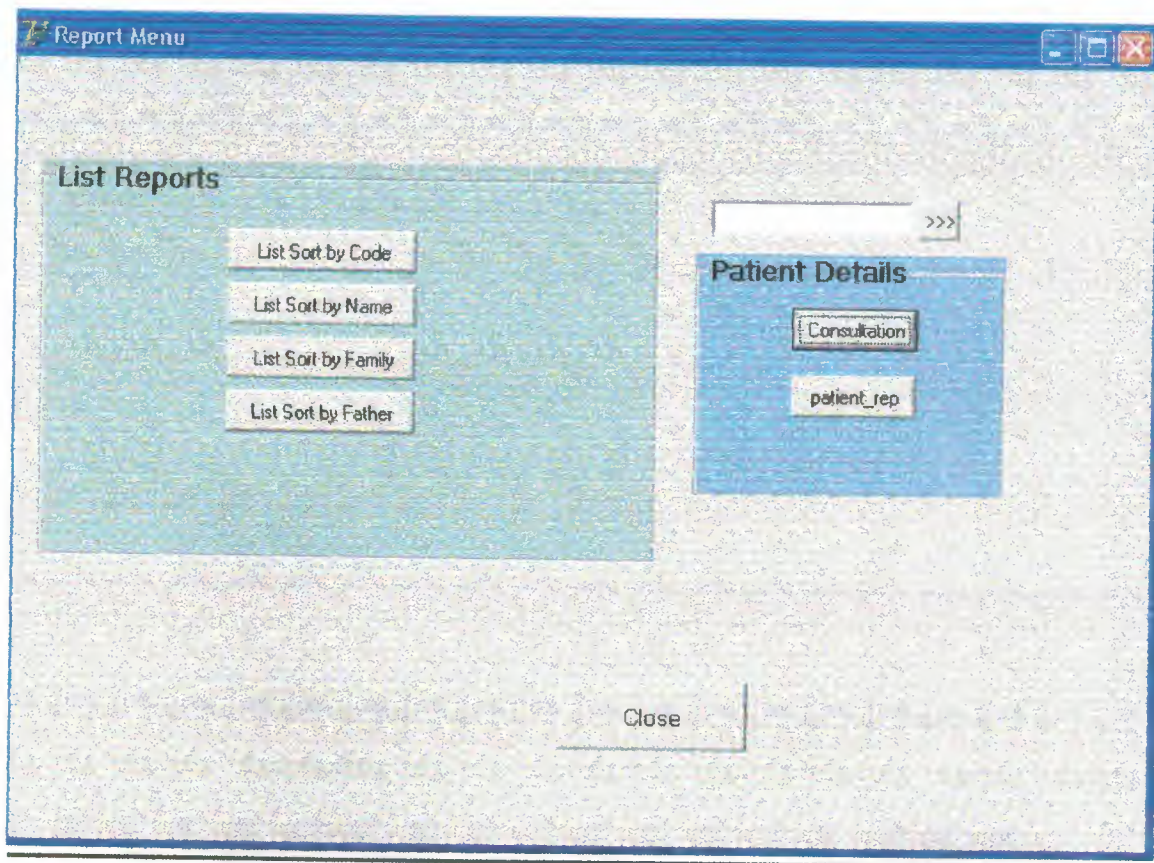
Search Form. B

Code	Name	Family	Father	DOB	Gender	City	Street
3	didstadi	albader	zhmed	1/1/2001	male		
5	reyad hasan	ali	sabas	3/3/2003	male	gaza	kljlk
8	a	1z	a				

Here the search form by Code that helps user to interact with patients and make search about their information's directly by inserting the code of patient he wants

Also here a detail button that also reach user to view the information of patient recorded in registration form and can be able to change the date he/she wants.

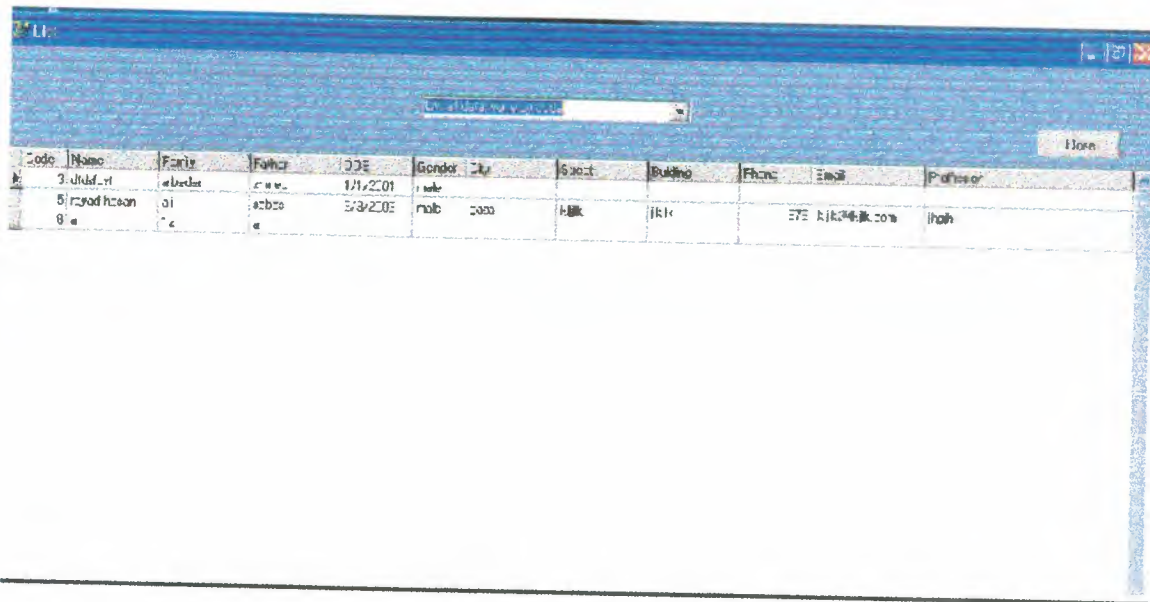
Report Menu



Here report menu which contain list of reports the user want to view and check and also enables him/her to view records saved by code, list sort by name, list sort by family, list code by father.

Here also very interactive option I decided to put in my program that make work on this program easier and comfortable by writing the code of specified patient directly user can see the consultation and patient reports that user passing through them and tracking by them and giving up as a report of his/her information and also the consultation process and close menu to return back to Main menu.

List Form



Code	Name	Family	Father	DOB	Gender	Age	School	Building	Phone	Email	Professor
3	Urdul H	albedat	albedat	1/1/2001	male						
5	Urdul H	albedat	albedat	1/1/2001	male						

Here list Menu which contain list of all patients save in the database and viewed up ascending to descending order by list data by code, list date by name, list date by father, list date by family and close button to return back to Main menu.

USER MANUAL

Software Installation

- You need Borland Delphi 6 or 7 in your computer.
- Create folder in drive D name is example Delphi.
- Copy the all file from folder project in your local drive in D:\example Delphi. Drive.

Compiling the File

- Double click to Run or F9

Run the File

- If there is no error file will be run.

REFERENCES

Books

- Murtata , Iaad , “ Delphi 6 example book “ , Alwafer, 2000, Edition I

Web sites

- www.kdtool.net