1988

NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

HOTEL RESERVATION SYSTEM

Graduation Project
COM 400

Prepared By: Selim Türkyılmaz (991882)

Supervisor: Mr.Okan Donangil

Lefkoşe - 2000

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I wish to tank my supervisor **Mr.Okan DONANGİL**, for intellectual support, encouragement, enthusiasm which made this thesis possible, and for his patience in correcting both my stylistic and scientific errors.

I tank to my wife, **Demet**, for her support and understanding.

I also wish to thank those who helped me to handle various computer problems, especially **Osman TEKİN** and **O.Taha CANANER**.

ABSTRACT

In the project, Hotel Reservation system and how it can be handled by computers is discussed. Basic idea of Data Base Management Systems is used and the implementation of this Hotel Reservation System is done by DELPHİ 5.0 which is an object oriented language. DELPHİ 5.0 is a powerfull database language and very easy to use than other languages.

The software written can be used to make reservation or registration for a customer to the hotel. It also calculates the bill of that a customer should pay when he leaves.

1. INTRODUCTION

Hotels are places where people can accommodate for a period of time. As a major component of world tourism, it drives the economic engines of developing countries and accounts for a great deal of money.

Due to the development of computer and communication technologies be automation become an important system in controlling the operations, (such as front office,) of Hotels.

The ability of computers, to store, process and manipulate information has greatly improved the efficiency of automation in hotels. By doing so, the guality of services that hotels offer to their customers have been greatly improved.

Automation of such services in hotels, saves the staff from paper work, and it can enhance the opportunities for staff utilization in the hospitality, and service.

2.WHAT IS AUTOMATION?

The development and integration of computers and communication technologies has made the the computers to be widely used in industry and business in our daily lives.

The ability of computers to process and manipulate information very fast and much more accurately have brought the automation into our agenda.

Automation is a system in which any organization performs their routine work by computers. Automation saves as from doing a lot of paper work and from time which means money in business especially. Also, by performing the operation of an organization using automation, helps to store a lot of information in a less space and gives us the advertage to access this information very fast.

Appendices.

80-100

15

3.HOTEL RESERVATION SYSTEM

Reservation process in a Hotel is carried out by the reception in the front-office. When a customer asks for a reservation in a Hotel, the receptionist asks for the following information;

*** **Reservation Date,**

*** **Name / Surname,**

*** **Check-in Date,**

*** **Check-out Date,**

*** **Incoming With,**

*** **Country,**

*** **City,**

*** **Reservation Done by who,**

*** **Room Number,**

*** **Note.**

(For the flowchart of Reservation system see **figure 1**.)

(For the screen layout of Reservation see **figure 2**.)

(For the flowchart of Reservation of Program see **figure 3**.)

(For the screen layout of Customer Reservation List see **figure 4.)**

(For the screen layout of Customer Reservation Report **figure 5.)**

(For the screen layout of Customer Reservation Report List **figure 6.)**

3.1Flowchart of System of Reservation

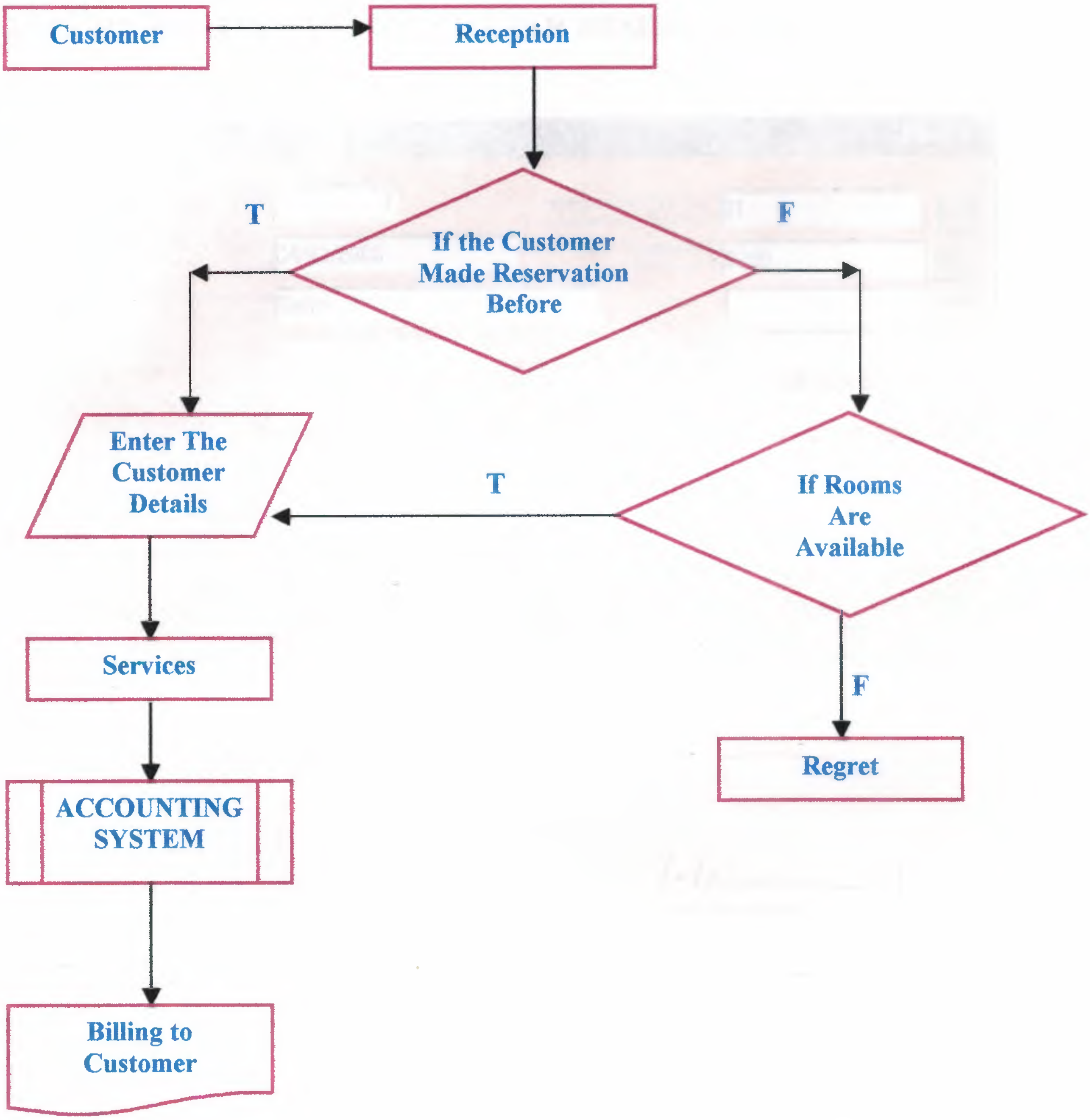
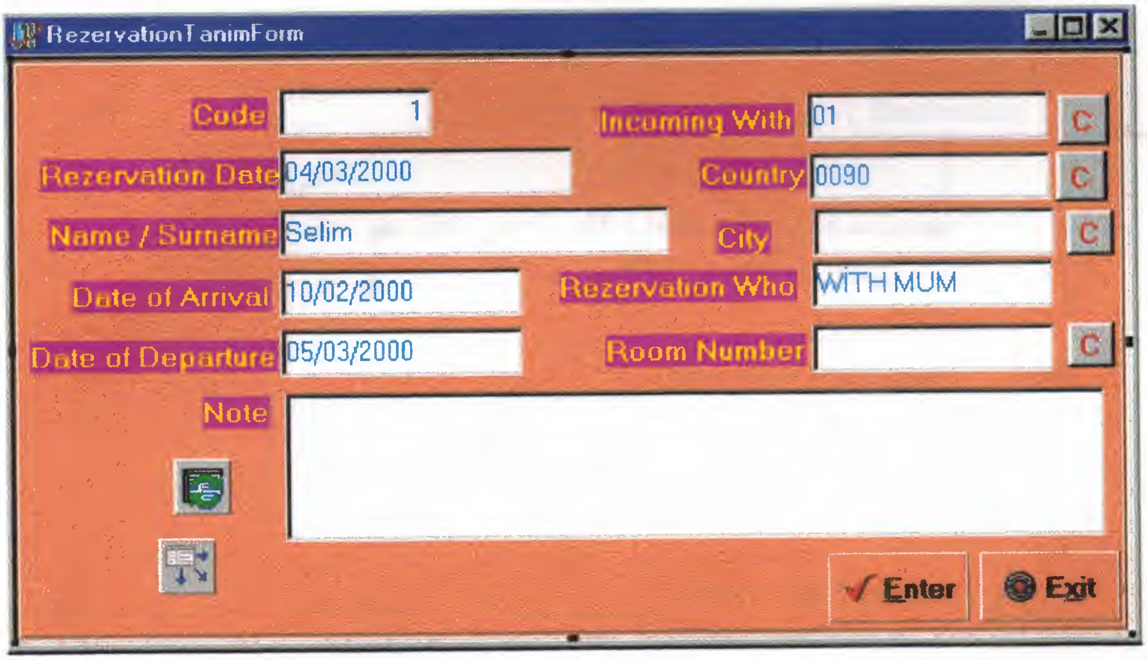


Fig 1.1 (⟶)

Figure 1.

3.2 SCREEN LAYOUT OF CUSTOMER RESERVATION;



RezervationTanimForm

Code	1	Incoming With	01
Rezervation Date	04/03/2000	Country	0090
Name / Surname	Selim	City	
Date of Arrival	10/02/2000	Rezervation Who	WITH MUM
Date of Departure	05/03/2000	Room Number	
Note			

Figure 2.

Chapter 1.

1.1. (- . . .)

fig 1.1. -

1. ___

1.1 (___)

1.1.1 ___

1.1.2 . ___

1.2 ___

Chapter 2 (--).

2.

2.1.

3.3 Flowchart of Customer Reservation;

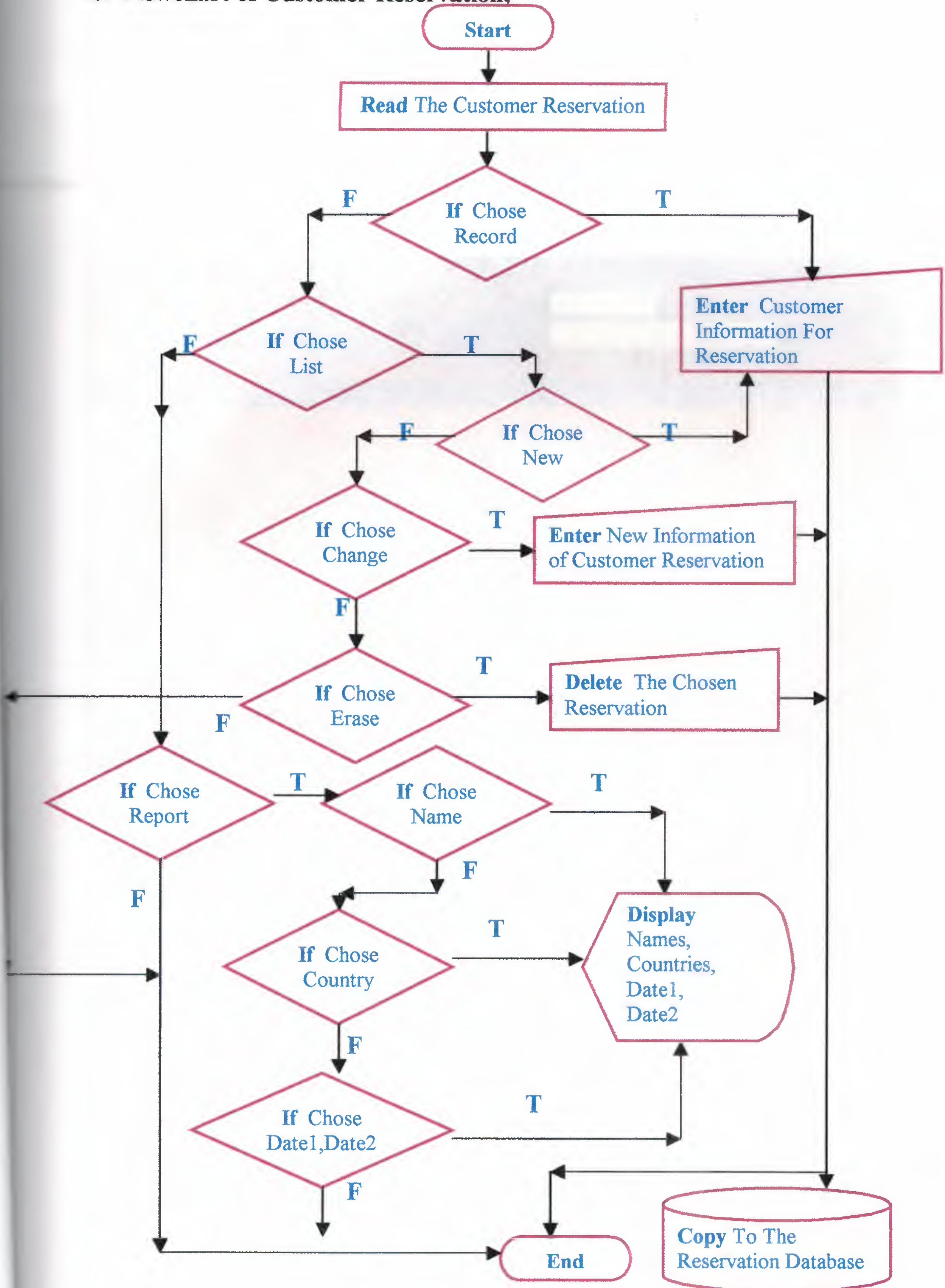
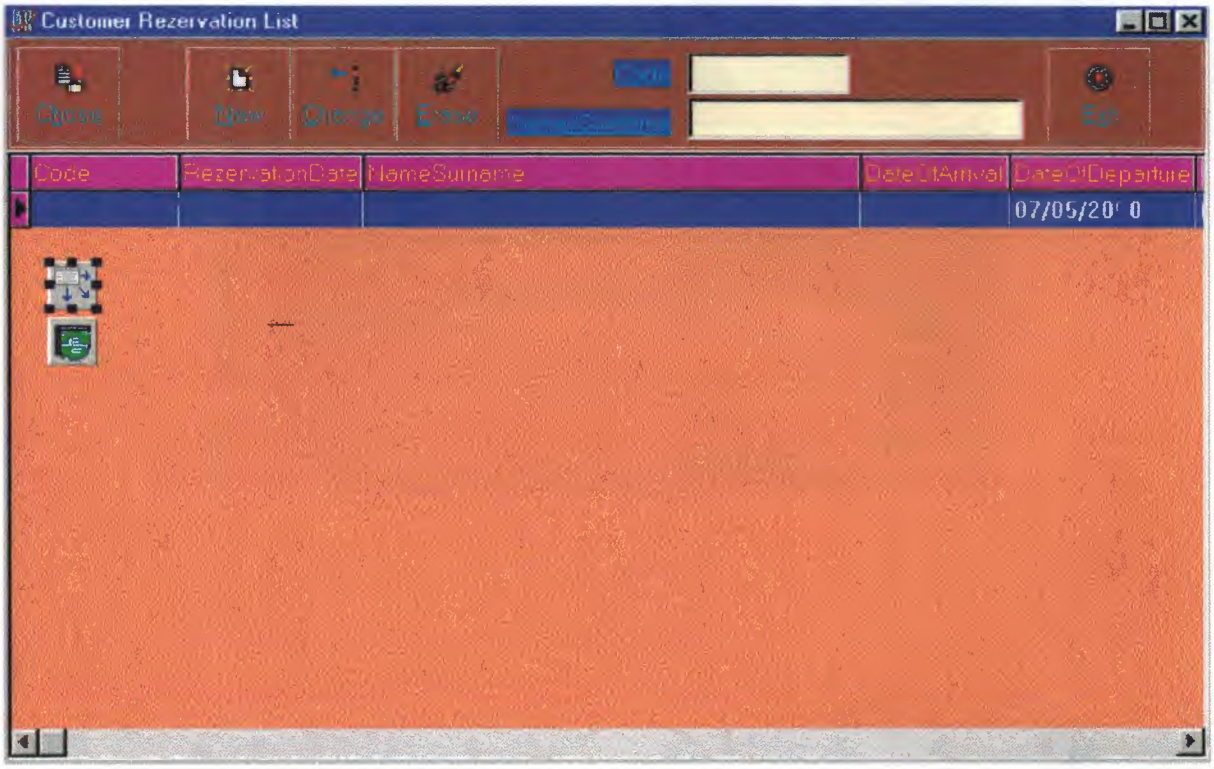

Start
Read The Customer Reservation
If Chose Record
F
T
Enter Customer Information For Reservation
If Chose List
F
T
If Chose New
F
T
If Chose Change
T
Enter New Information of Customer Reservation
F
If Chose Erase
T
Delete The Chosen Reservation
F
If Chose Report
T
If Chose Name
T
F
F
If Chose Country
T
Display Names, Countries, Date1, Date2
F
If Chose Date1,Date2
T
F
End
Copy To The Reservation Database

Figure 3.

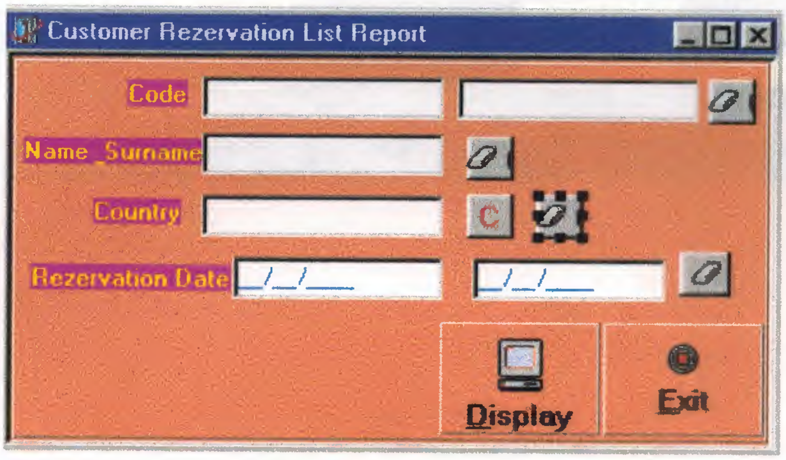
3.4 SCREEN LAYOUT OF CUSTOMER RESERVATION LIST;



Code	RezervationDate	NameSurname	DateOfArrival	DateOfDeparture
				07/05/20' 0

Figure 4.

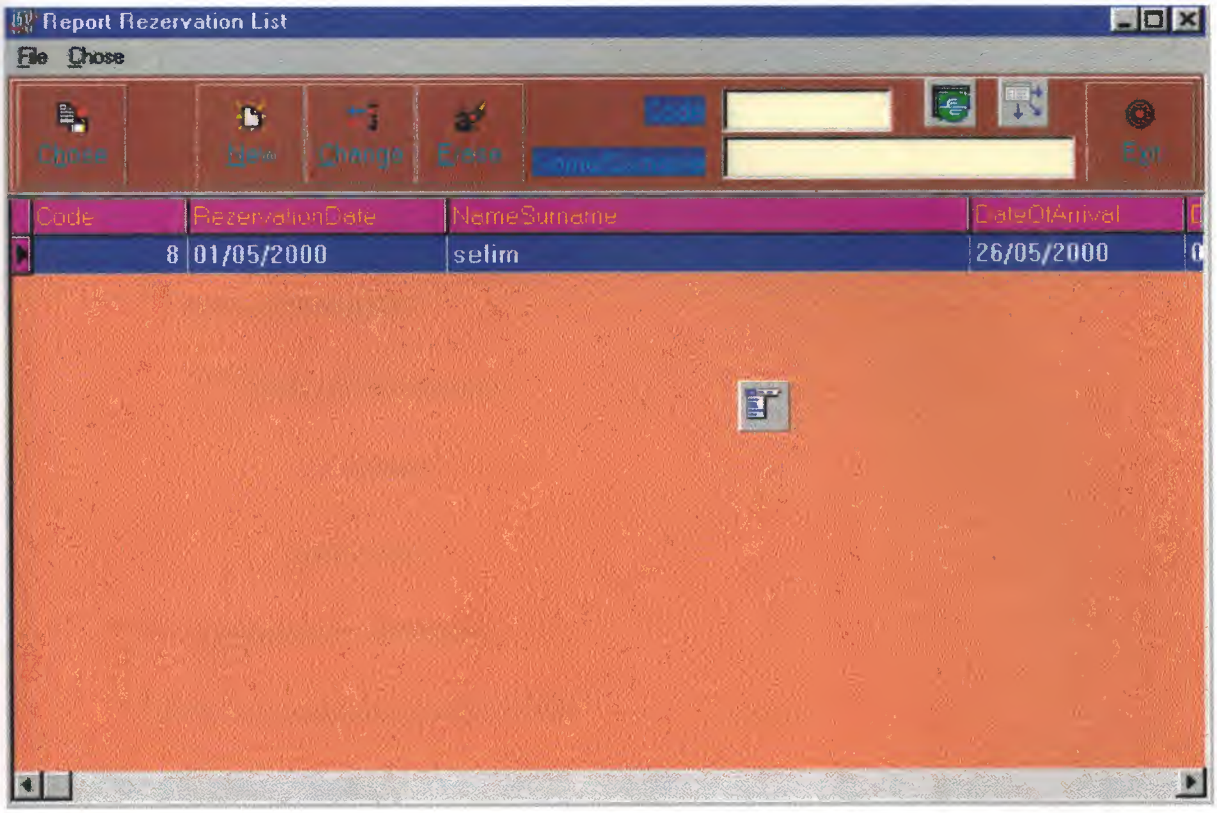
3.5 SCREEN LAYOUT OF DEFINITION OF RESERVATION REPORT;



Customer Rezervation List Report

Figure 5.

3.6 SCREEN LAYOUT OF CUSTOMER RESERVATION REPORT LIST;



Code	RezervationDate	NameSurname	DateOfArrival
8	01/05/2000	selim	26/05/2000

Figure 6.

3.7 OVERHEAD FOR THE RECEPTION

*** Reservation with the details,

*** List the customers on the reservation,

*** List of rooms and free rooms and with their specialist,

*** List of country and code of country,

*** List of city and code of city,

*** Types of services;

*** Bed and Breakfast:

- Halfboard,
- Fullboard.

*** Registration with details,

*** List the customers in the Hotel,

*** Customer Extras

- Bar
- Restaurant
- Swimming Pool,

*** Total Bill of Customer.

4. Customer Registration Process

After checking the reservation or the available rooms, the customer will be accepted to the Hotel.

During this properties, the personel information of a customer will be saved. Tis information will be as follws:

***** Name / Surname,**

***** Passport Number,**

***** Check – in Date,**

***** Check – out Date,**

***** Pension,**

***** Room Number,**

***** Price,**

***** Incoming With,**

***** Age,**

***** Country,**

***** City,**

***** Accompanied By,**

***** Address,**

***** Note.**

(For the flowchart of Customer Registration see **figure 7.**)

(For the screen layout of Customer Registration see **figure 8.**)

(For the screen layout of Customer Registration List see **figure 9.**)

4.1 Flowchart of Customer Registration;

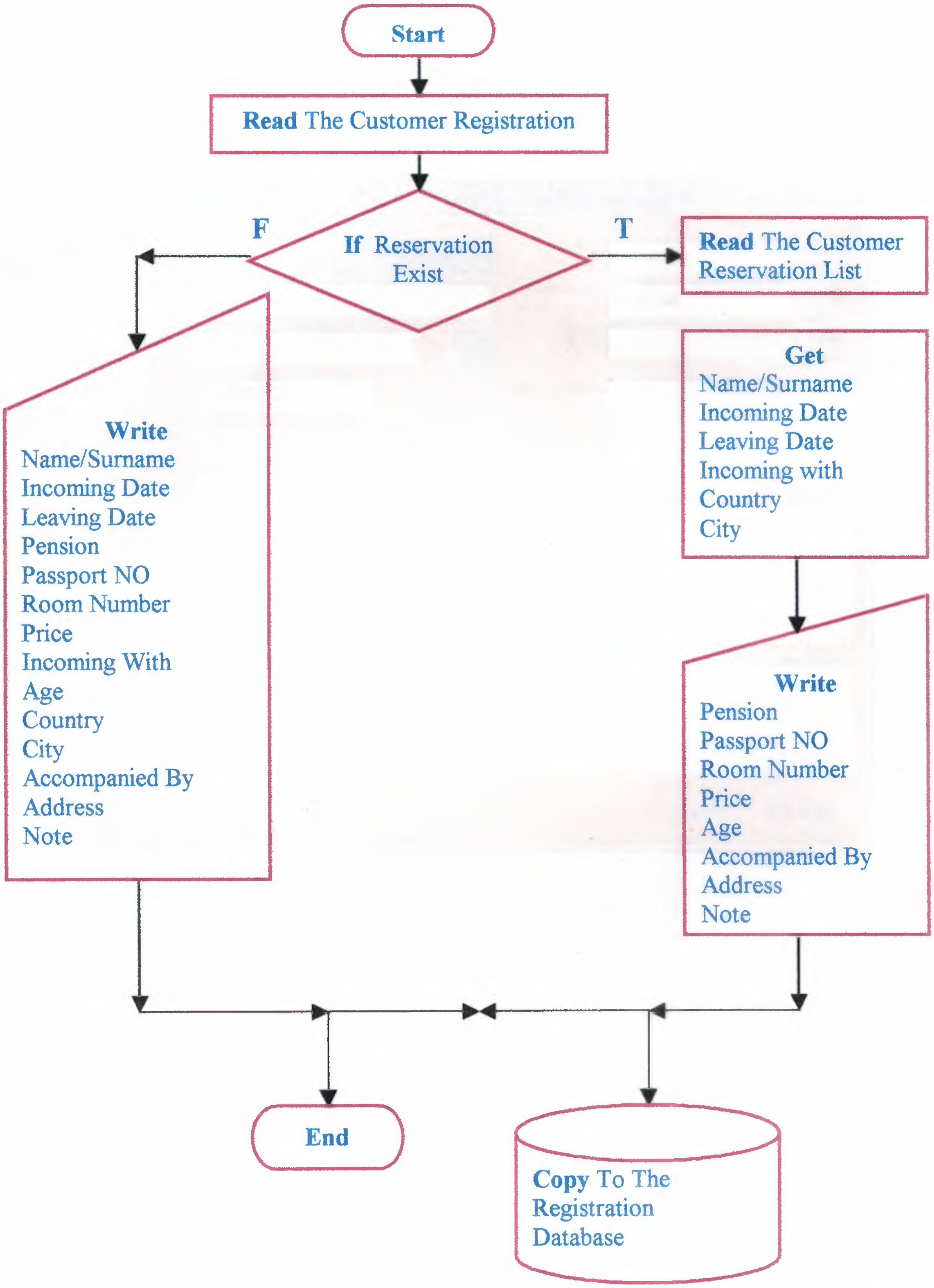
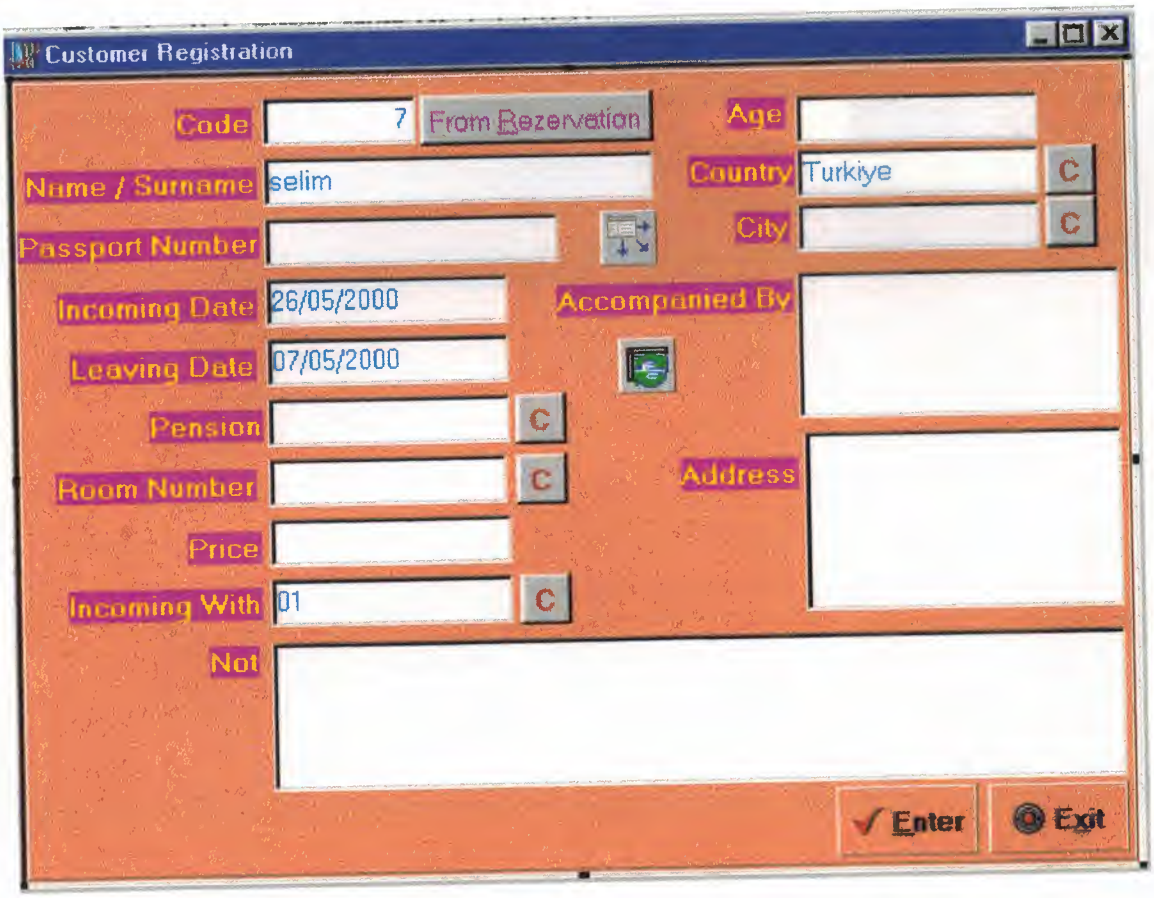


Figure 7.

4.2 SCREEN LAYOUT OF CUSTOMER REGISTRATION;



Customer Registration

Code 7

Name / Surname selim

Passport Number

Incoming Date 26/05/2000

Leaving Date 07/05/2000

Pension

Room Number

Price

Incoming With 01

Not

Age

Country Turkiye

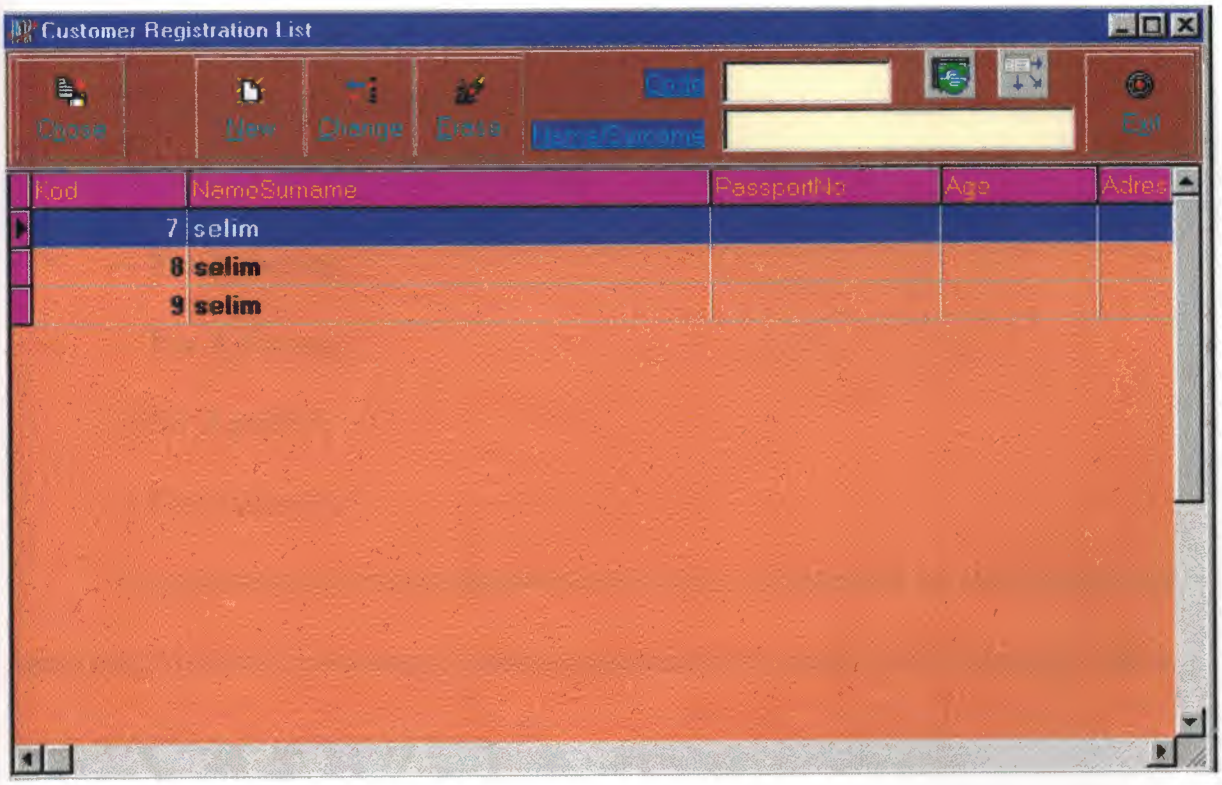
City

Accompanied By

Address

Figure 8.

4.3 SCREEN LAYOUT OF CUSTOMER REGISTRATION LIST;



Kod	NameSurname	PassportNo	Age	Adres
7	selim			
8	selim			
9	selim			

Figure 9.

4.4 DOCUMENTS OF ROOMS:

CLASSIFICATION OF ROOMS:

*** **According to the Quality**

--- First Quality (Suit),

---Second Quality(Double),

---Thirty Quality(Single).

*** **According to the Person's Number**

--- For 1 person,

--- For 2 person,

---For 3 person,

--- For 4 person.

Properties and number of person of any room is described by the receptionist according to Hotel's properties. And the receptionist can search if he want to look a room according to number of room.

(For the flowchart see **figure 1.**)

(For screen layout see **figure 2.**)

(For the screen layout of Room List **figure 3.**)

4.4.1 Flowchart of Rooms;

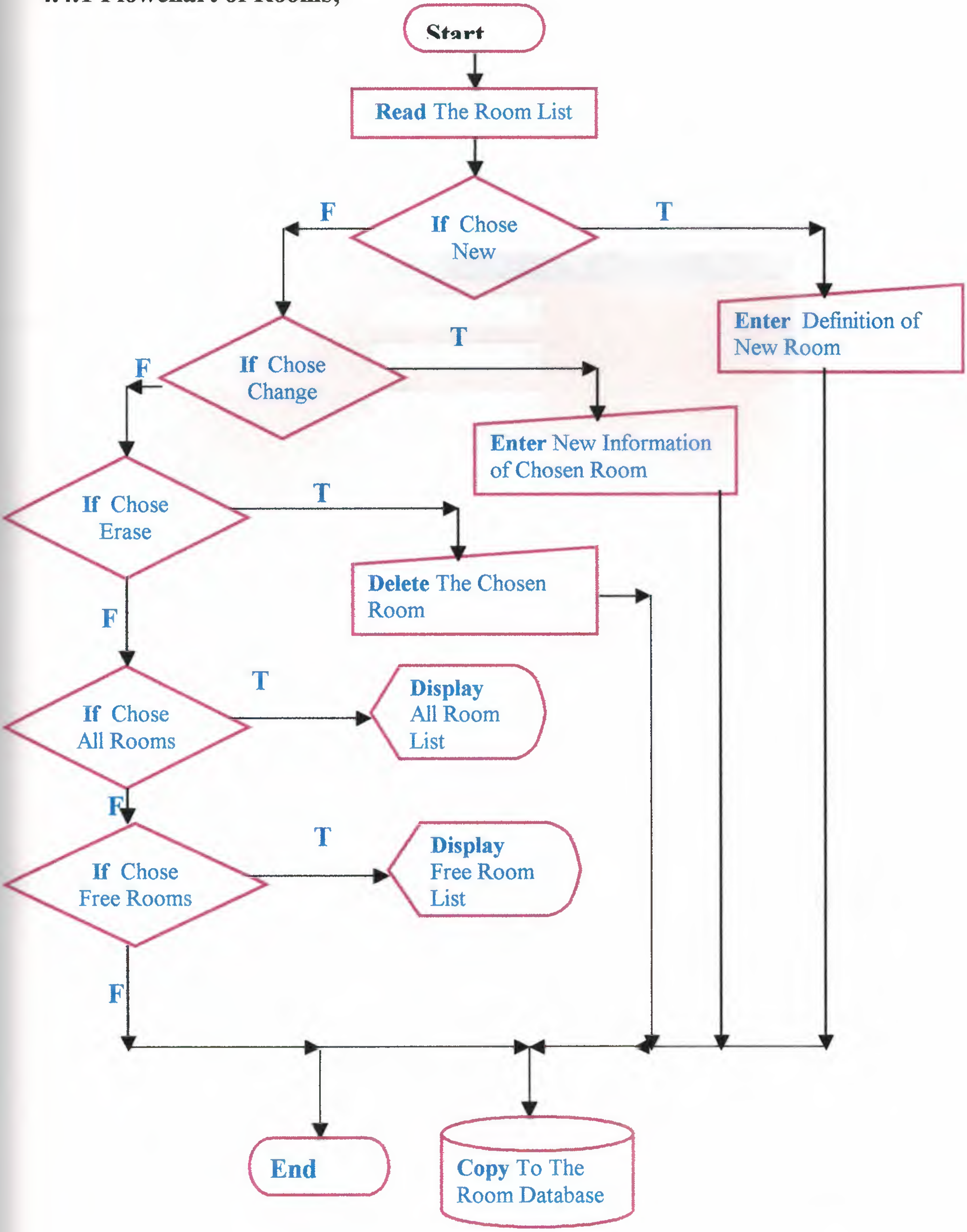
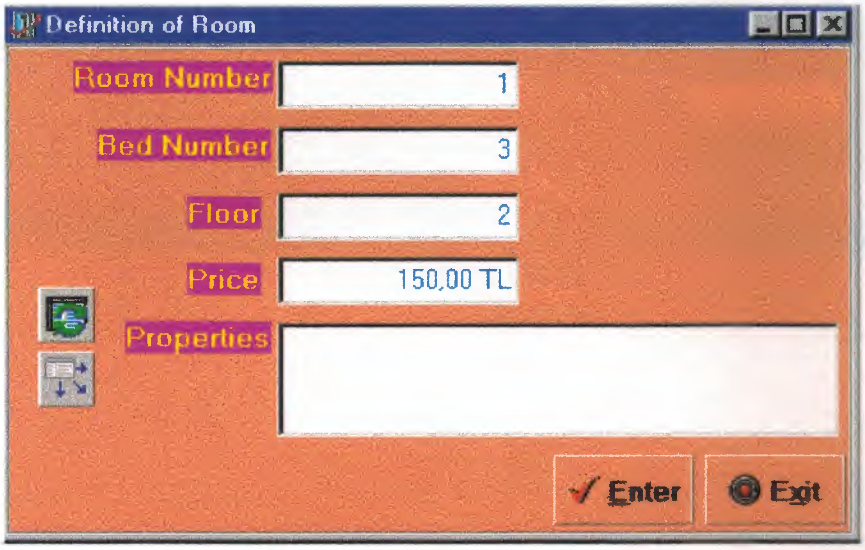


Figure 10.

4.4.2 SCREEN LAYOUT OF DEFINITION OF ROOM;



Room Number 1

Bed Number 3

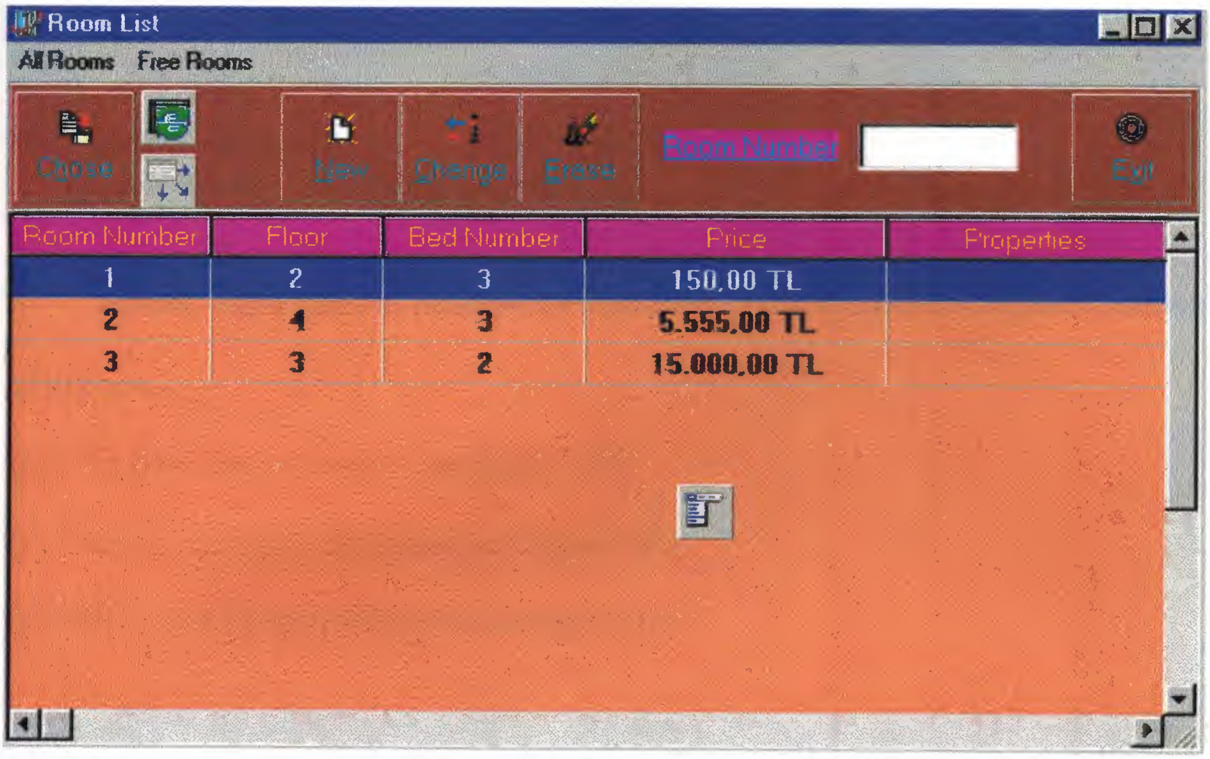
Floor 2

Price 150,00 TL

Properties

Figure 11.

4.4.3 SCREEN LAYOUT OF ROOM LIST



Room Number	Floor	Bed Number	Price	Properties
1	2	3	150,00 TL	
2	4	3	5.555,00 TL	
3	3	2	15.000,00 TL	

Figure 12.

4.5 DOCUMENTS OF PENSION:

***** Bed and Breakfast: (BB)**
The Price of the Brakfast times each day.

***** Half Board: (HB)**
Bed and Breakfast + Price of dinner times each day.

***** Full Board: (FB)**
Half Board + Price of noon eating.

Before, these types of pension are described by the receptionist at the definition of pension.

(For the flowchart of pension see **figure 13.**)

(For the screen layout of pension see **figure 14.**)

(For the screen layout of Pension List **figure 15.**)

4.5.1Flowchart of Pension;

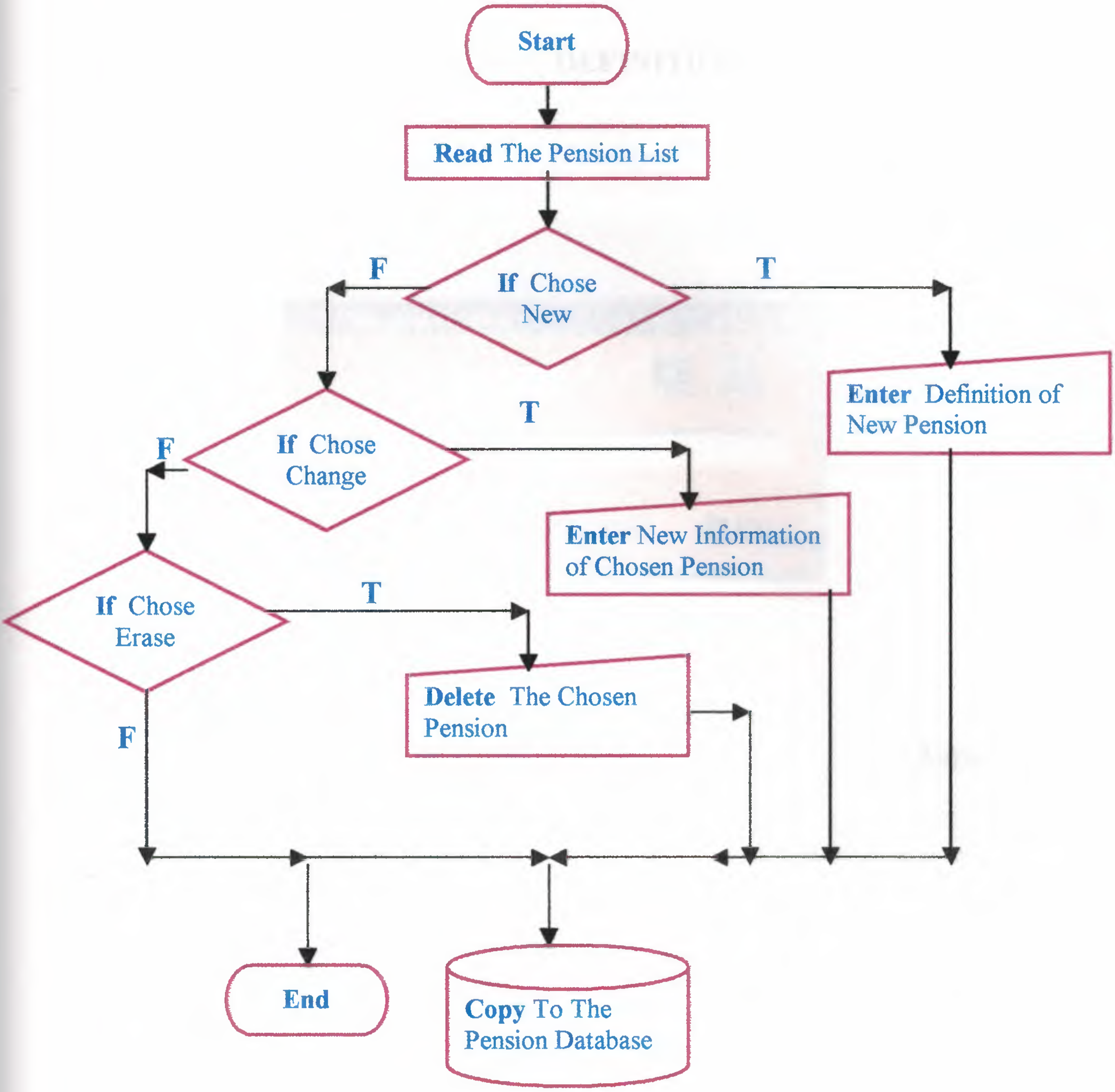
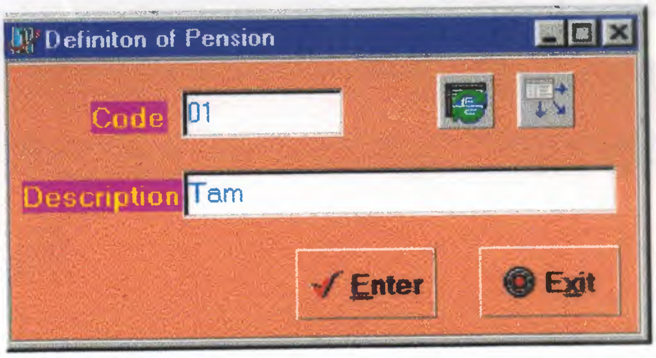

Start
Read The Pension List
If Chose New
F
T
Enter Definition of New Pension
If Chose Change
F
T
Enter New Information of Chosen Pension
If Chose Erase
T
Delete The Chosen Pension
F
End
Copy To The Pension Database

Figure 13

4.5.2 SCREEN LAYOUT OF DEFINITION OF PENSION;

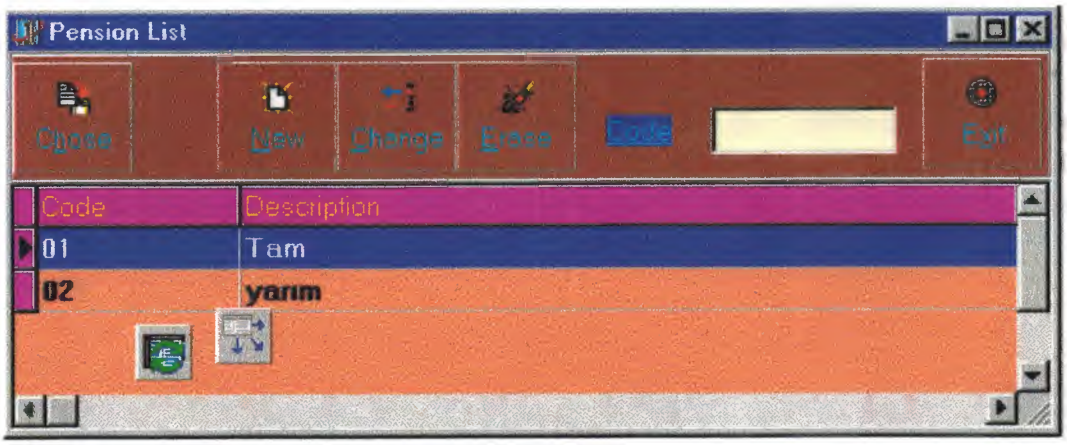


Code 01

Description Tam

Figure 14.

4.5.3 SCREEN LAYOUT OF PENSION LIST;



Code	Description
01	Tam
02	yarım

Figure 15.

4.6 DOCUMENT OF COUNTRY:

***** Code,**
***** Name of Country.**

Code is a type and parameter. Code is described by the receptionist. It can be postal zip etc... The receptionist does not need to write Code and Name of Country again and again for reservation and regisration. Because this information is written by the receptionist before. But the receptionist needs to write a new Country , he can write esaily.

(For flowchart of Country see **figure 16.**)

(For the screen layout of Definition of Country see **figure 17.**)

(For the screen layout of Country List see **figure 18.**)

4.6.1 Flowchart of Country;

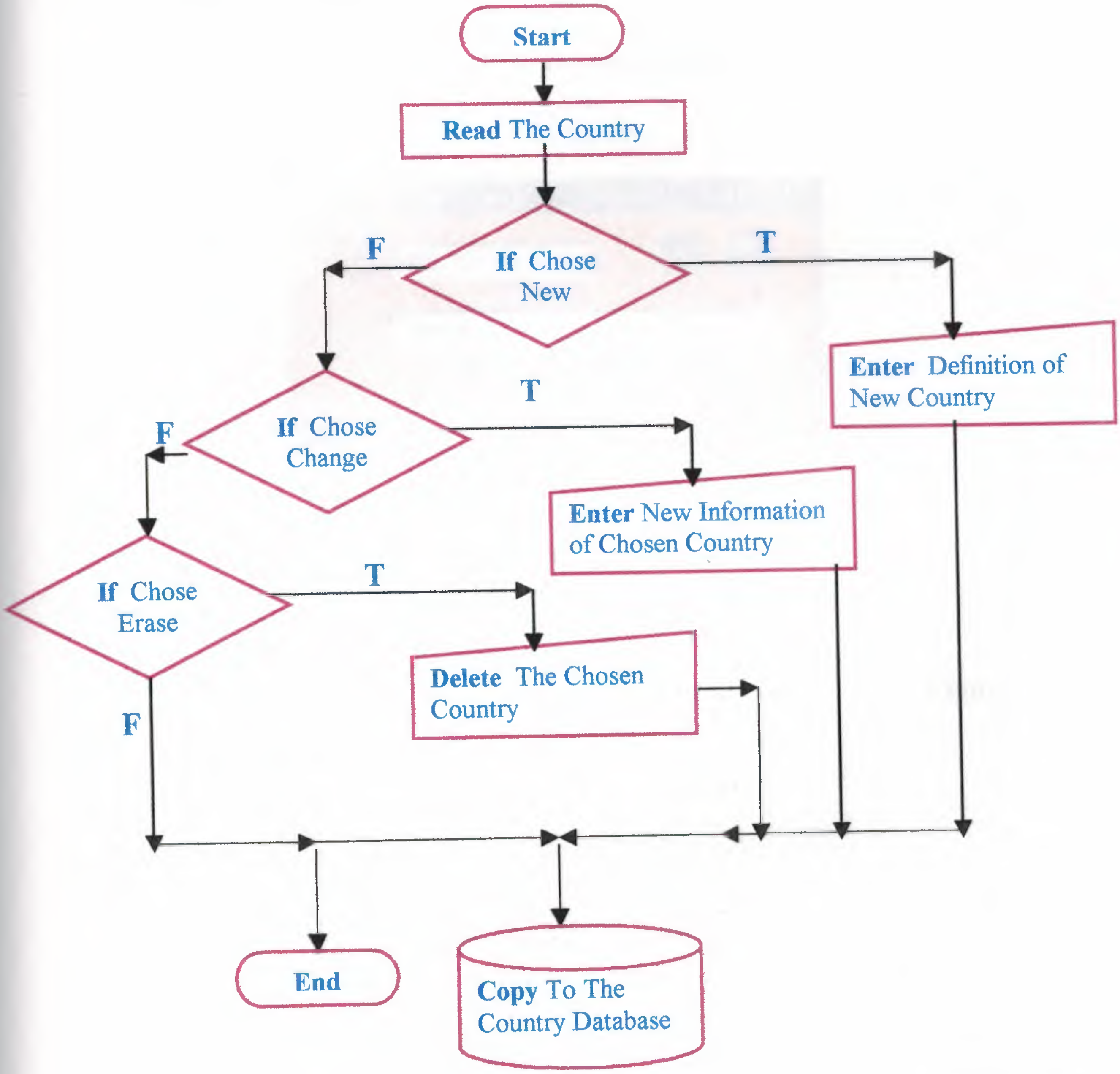
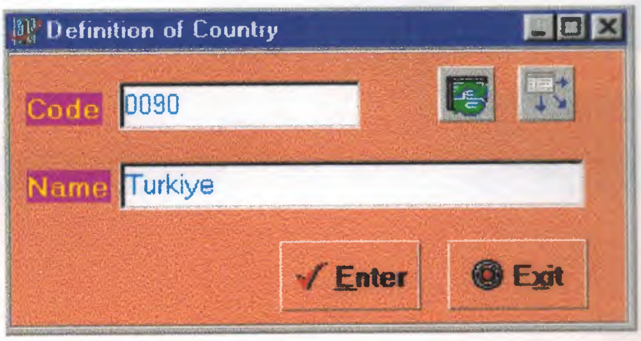

Start
Read The Country
If Chose New
F
T
Enter Definition of New Country
If Chose Change
F
T
Enter New Information of Chosen Country
If Chose Erase
T
F
Delete The Chosen Country
End
Copy To The Country Database

Figure 16.

4.6.2 SCREEN LAYOUT OF DEFINITION OF COUNTRY;

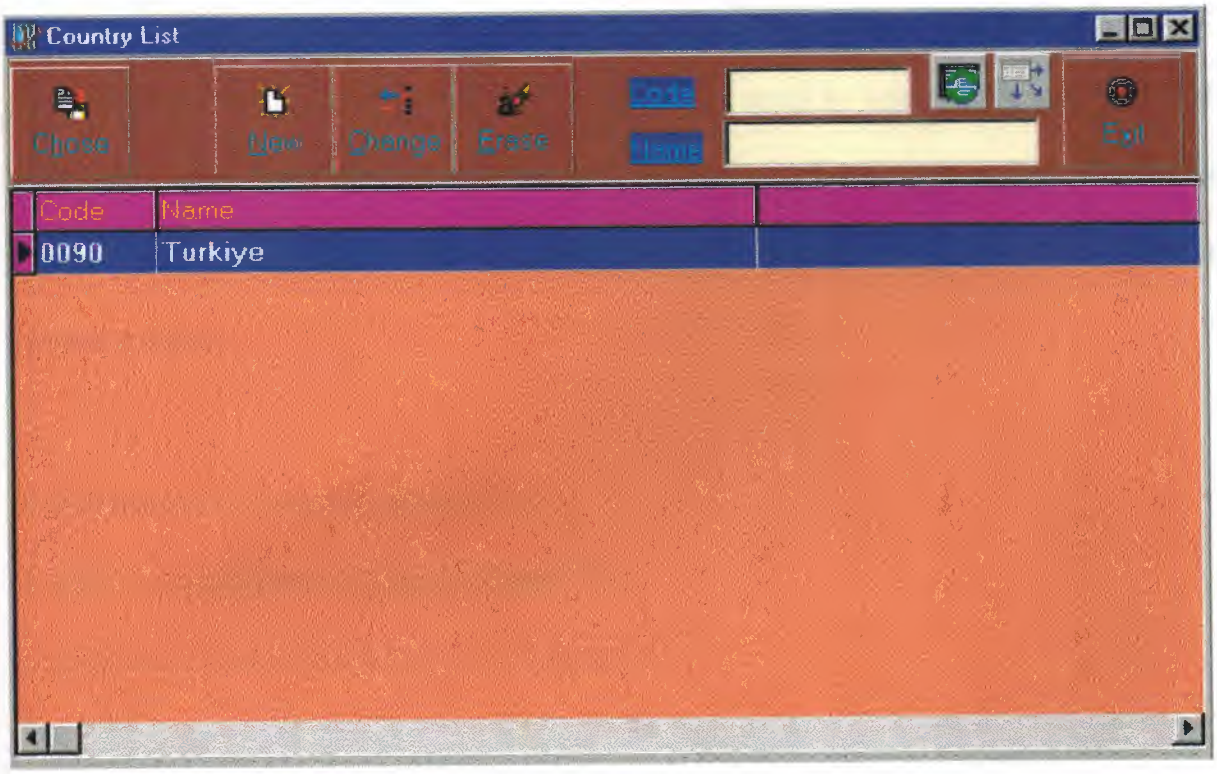


Code: 0090

Name: Turkiye

Figure 17.

4.6.3 SCREEN LAYOUT OF COUNTRY LIST;



Code	Name
0090	Turkiye

Figure 18.

4.7 DOCUMENTS OF CITY:

***** Code,**
***** Name of City.**

Code is a type and parameter. Code is described by the receptionist. It can be plate number and etc... The receptionist does not need to write Code and Name of City again and again for reservation and regisration. Because this information is written by the receptionist before. But the receptionist needs to write a new City , he can write esaily.

(For flowchart of City see **figure 19**.)

(For the screen layout of City see **figure 20.**)

(For the screen layout of City List see **figure 21.)**

4.7.1 Flowchart of City;

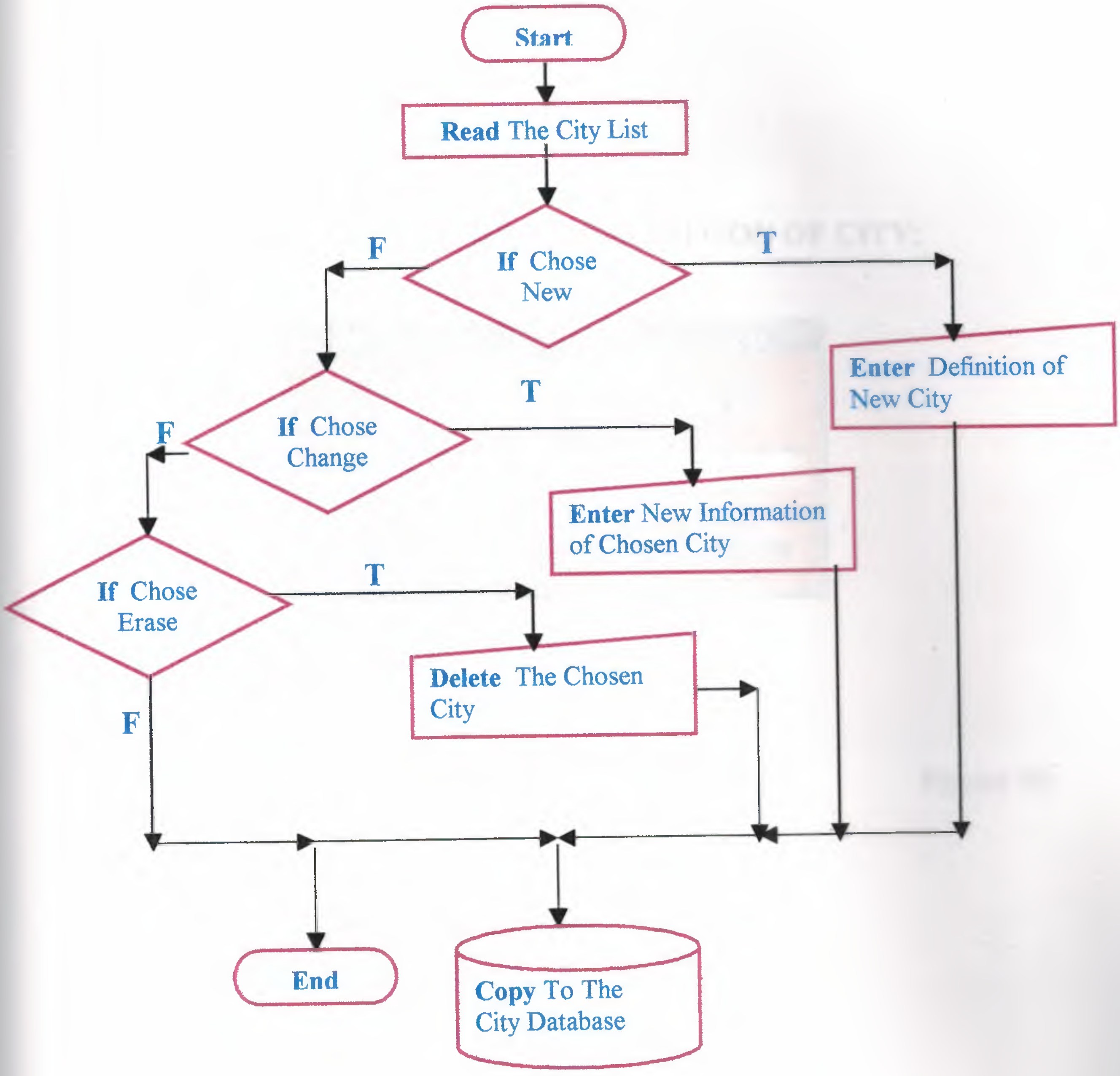
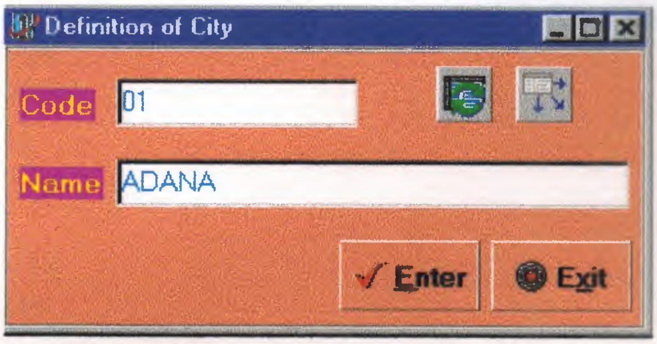


Figure 19

4.7.2 SCREEN LAYOUT OF DEFINITION OF CITY;



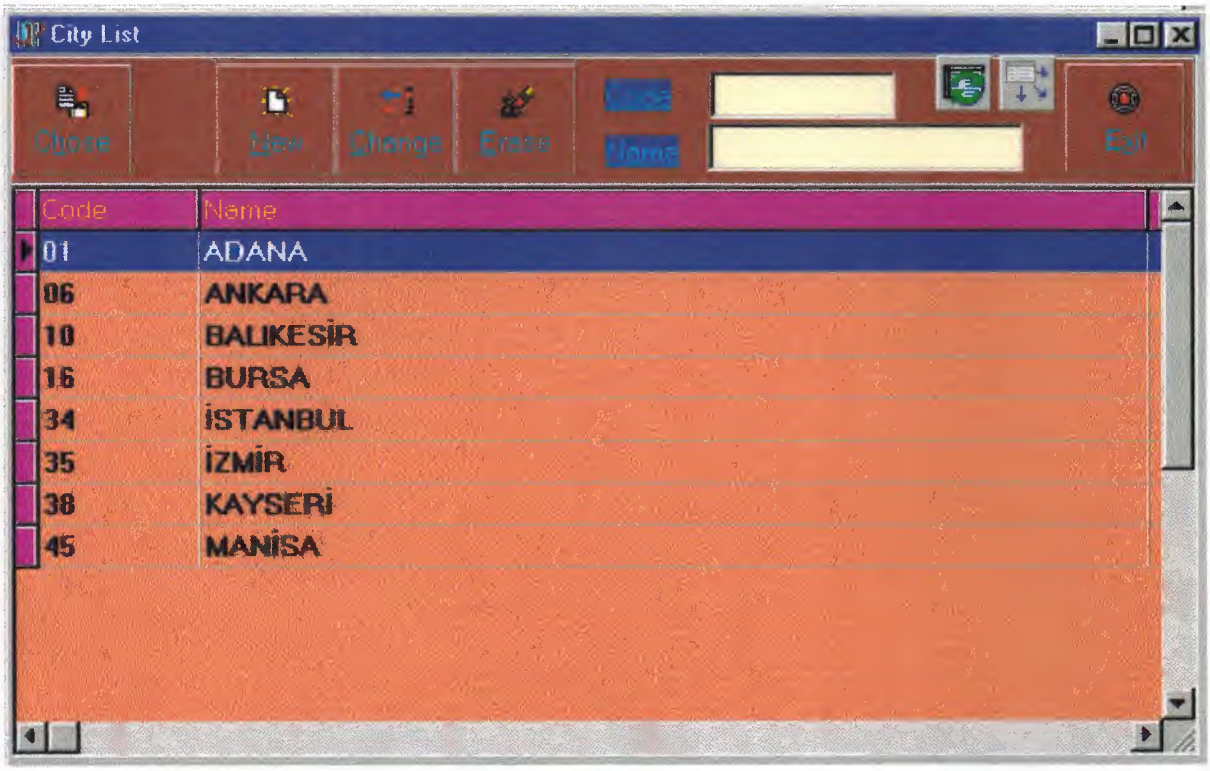
Definition of City

Code: 01

Name: ADANA

Figure 20.

4.7.3 SCREEN LAYOUT OF CITY LIST;



Code	Name
01	ADANA
06	ANKARA
10	BALIKESİR
16	BURSA
34	İSTANBUL
35	İZMİR
38	KAYSERİ
45	MANİSA

Figure 21.

4.8.1 Flowchart of Incoming With;

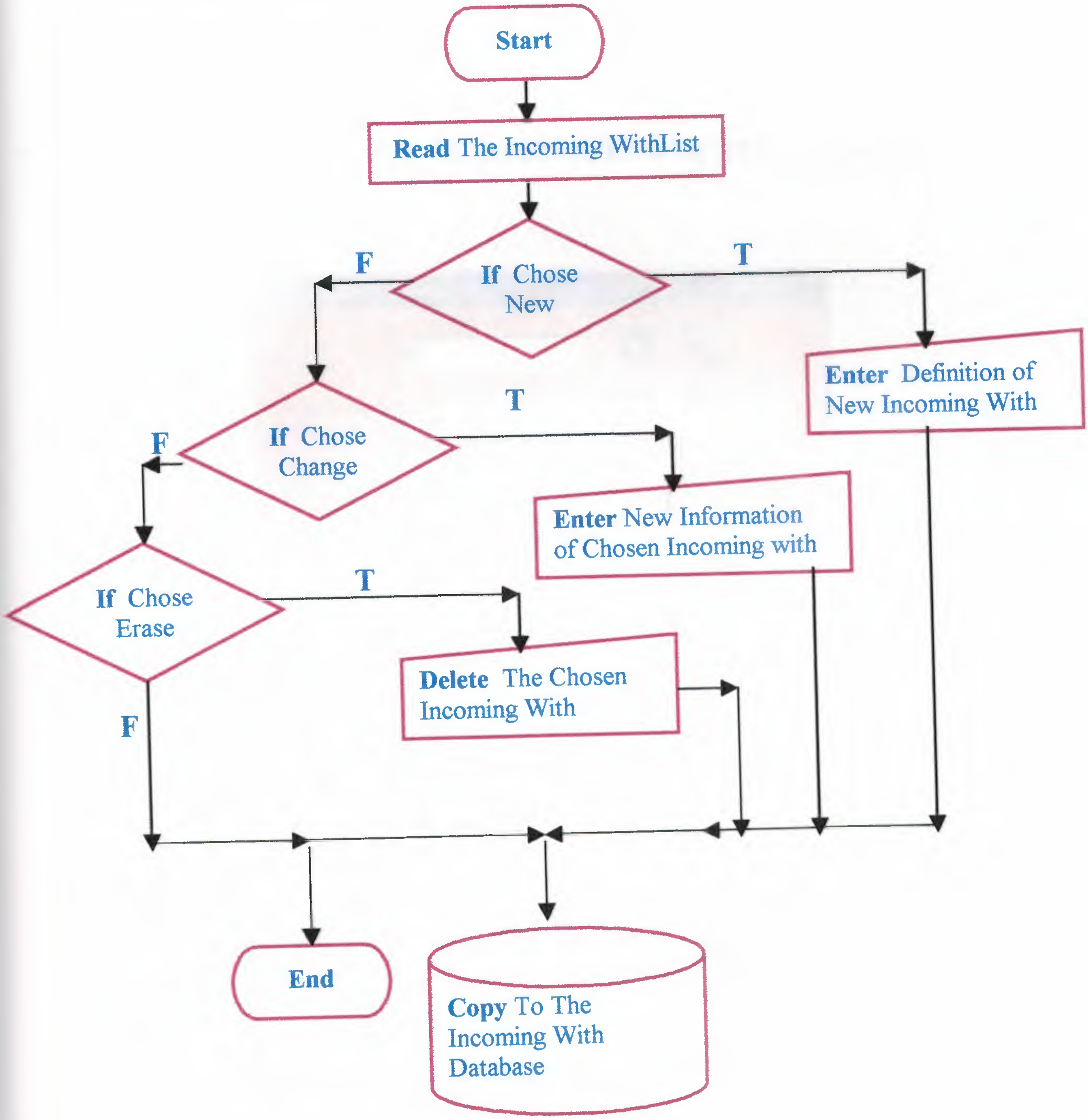
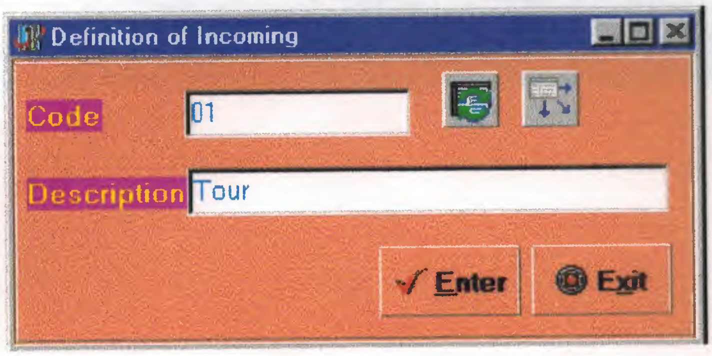

Start
Read The Incoming WithList
If Chose New
F
T
Enter Definition of New Incoming With
If Chose Change
F
T
Enter New Information of Chosen Incoming with
If Chose Erase
T
F
Delete The Chosen Incoming With
End
Copy To The Incoming With Database

Figure 22.

4.8.2 SCREEN LAYOUT OF DEFINITION OF INCOMING WITH;

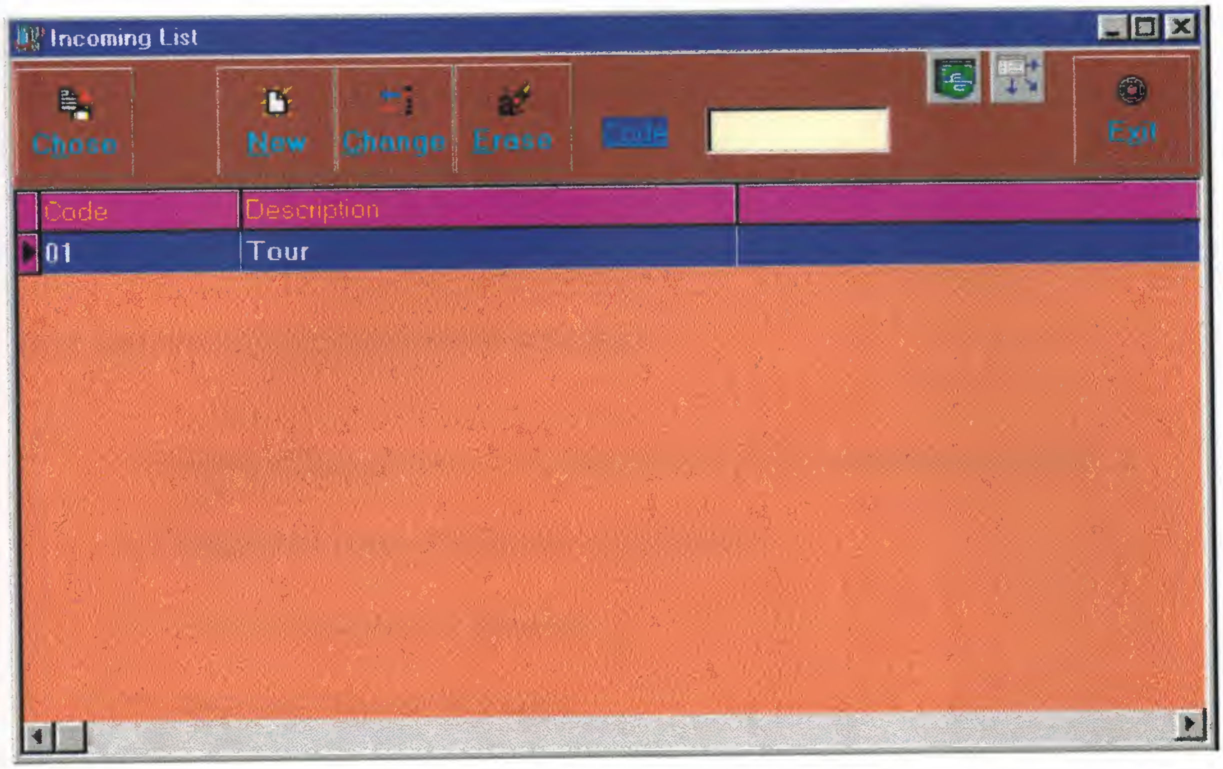


Code 01

Description Tour

Figure 23.

4.8.3 SCREEN LAYOUT OF INCOMING WITH LIST;



Code	Description
01	Tour

Figure 24.

5. Accounting Process

When a customer is going to check out from the Hotel his bill will be prepared by the reception. In order to prepare the bill, the receptionist will choose the record of the required customer and by entering the check – out date, his bill will be prepared together with his extras.

5.1 Customer Extras:

DOCUMENTS OF CUSTOMER EXTRA:

***** Name of Department (e.g. Restaurant, Bar, Swimming Pool etc...),**

***** Signature, Name / Surname of Cashier,**

***** Name Surname of Customer,**

***** Date and Amount of Extra,**

***** Detail of Account,**

***** Bill of Account.**

(For the flowchart of Accounting see **figure 25.**)

(For the screen layout of Accounting see **figure 26.**)

(For the screen layout of Accounting List see **figure 27.**)

(For the flowchart of Detail of Extra Expenditure of Customer see **figure 28.**)

5.2 Flowchart of Customer Account;

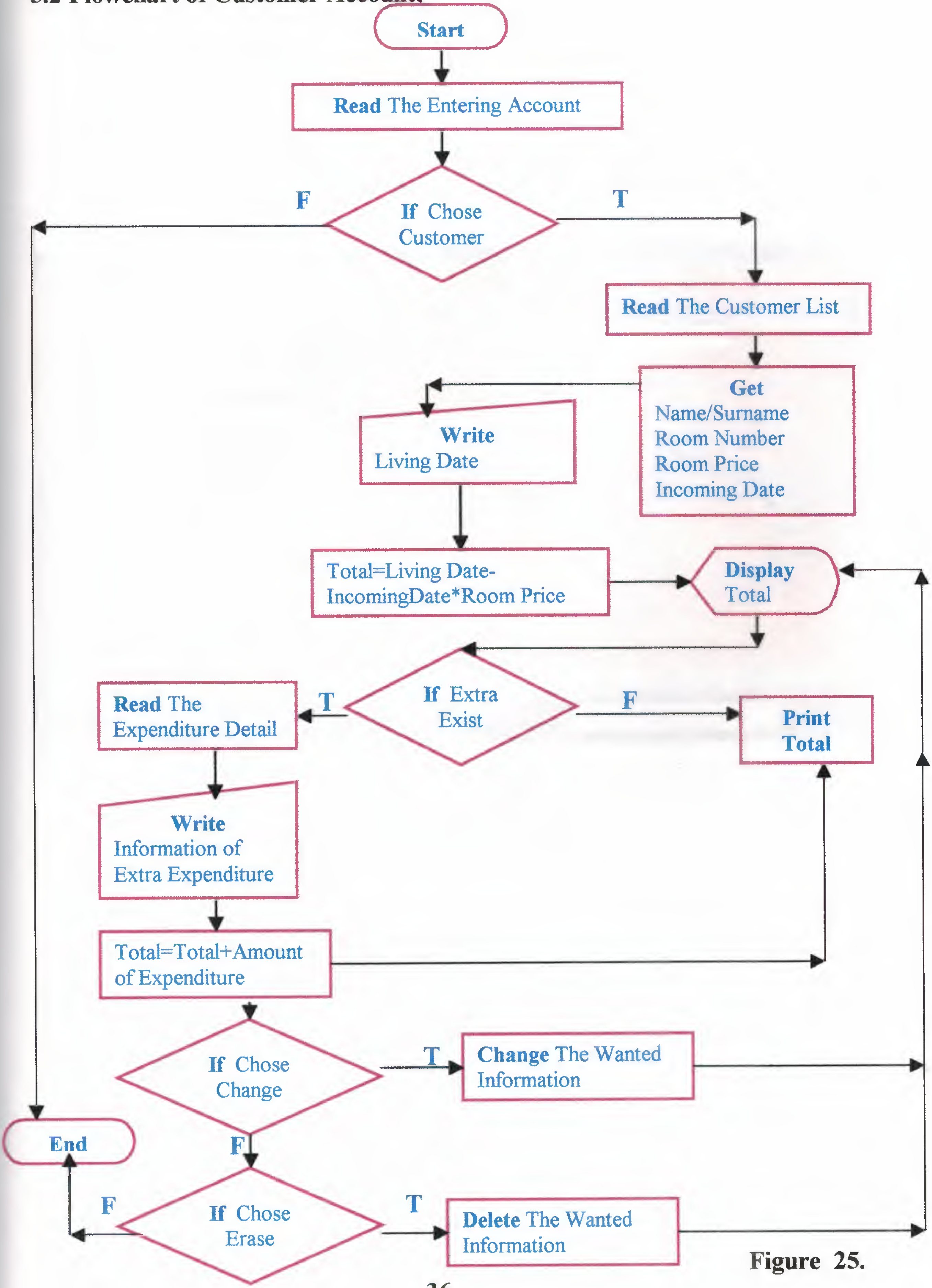
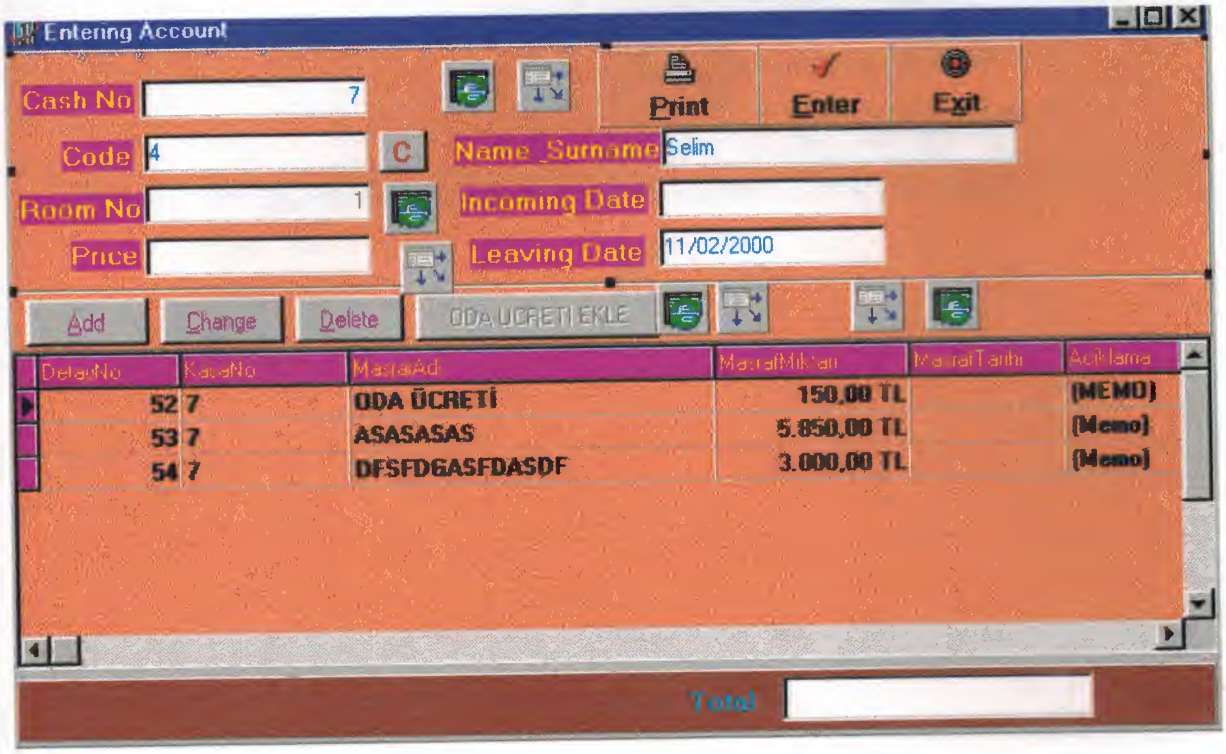

Start
Read The Entering Account
If Chose Customer
F
T
Read The Customer List
Get
Name/Surname
Room Number
Room Price
Incoming Date
Write
Living Date
Total=Living Date-IncomingDate*Room Price
Display Total
If Extra Exist
T
F
Read The Expenditure Detail
Print Total
Write
Information of Extra Expenditure
Total=Total+Amount of Expenditure
If Chose Change
T
Change The Wanted Information
End
F
If Chose Erase
F
T
Delete The Wanted Information

Figure 25.

5.3 SCREEN LAYOUT OF ENTERING ACCOUNT;



Cash No: 7

Code: 4

Room No: 1

Price:

Name_Surname: Selim

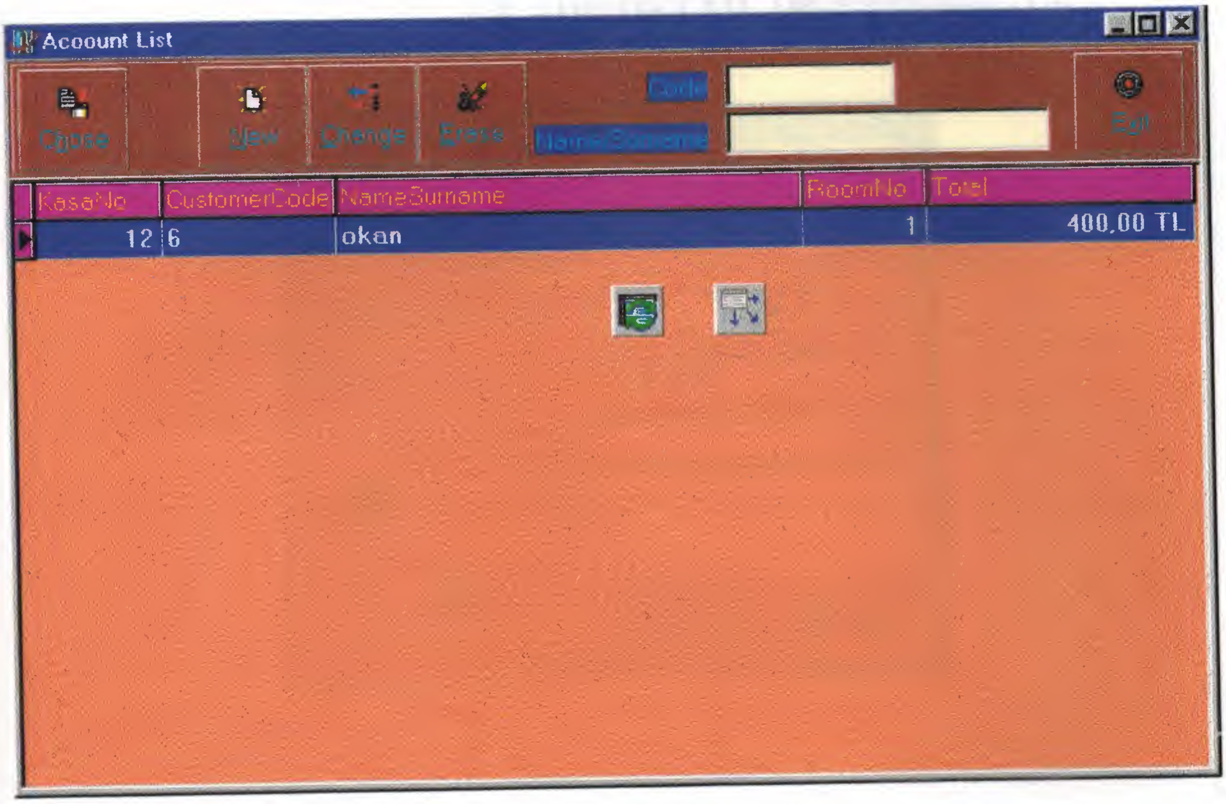
Incoming Date:

Leaving Date: 11/02/2000

DetayNo	KasaNo	MasrafAdi	MasrafMiktari	MasrafTarihi	Aciklama
52	7	ODA ÜCRETİ	150,00 TL		(MEMO)
53	7	ASASASAS	5.850,00 TL		(Memo)
54	7	DFSFDGASFDASDF	3.000,00 TL		(Memo)

Figure 26.

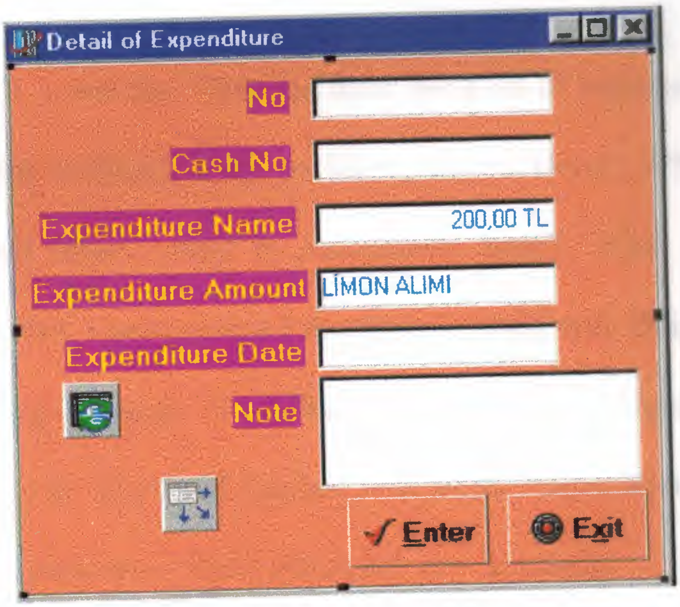
5.4 SCREEN LAYOUT OF ACCOUNT LIST;



KasaNo	CustomerCode	NameSurname	RoomNo	Total
12	6	okan	1	400,00 TL

Figure 27.

5.5 SCREEN LAYOUT OF ADD DETAIL OF EXPENDITURE ;



No:

Cash No:

Expenditure Name: 200,00 TL

Expenditure Amount: LİMON ALIMI

Expenditure Date:

Note:

Figure 28.

CONCLUSION

For the result, I said that I learned there are a lot of important things from this project such as; What is Hotel?, What are there jobs in a Hotel?, What is Automation?, How is Automation done?, Why is Automation done?, What are techniques of Automation?. Whit this, I believe that I developed myself with Delphi which is object oriented programming language. And I learned importance of Hotel which is heart of Tourism very well. In shortly, I learned rservation, registration, account, extras, properties of rooms, pension of Hotel which are very important.

REFERENCES:

İhsan Karagülle & Zeydin Pala (1999, August). Delphi 4.0. Türkmen Kitabevi.

Memik Yanık (1998). Delphi 3.0. Beta Kitabevi.

Appendix A:

CODE OF DEFINITION OF ROOM IN PROGRAM

```
unit OdaTanimUnit;

interface

uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
        Buttons, ExtCtrls, StdCtrls, DBCtrls, Mask, Db, DBTables, Refresh;

type
        TOdaTanimForm = class(TForm)
         Panel1: TPanel;
         SpeedButton2: TSpeedButton;
         SpeedButton1: TSpeedButton;
         Label1: TLabel;
         Label2: TLabel;
         OdaSource: TDataSource;
         Label3: TLabel;
         Label4: TLabel;
         Label5: TLabel;
         DBEdit1: TDBEdit;
         DBEdit2: TDBEdit;
         DBEdit3: TDBEdit;
         Price: TDBEdit;
         DBMemo1: TDBMemo;
         OdaTable: TSelfRefreshTable;
         procedure SpeedButton1Click(Sender: TObject);
         procedure SpeedButton2Click(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
        OdaTanimForm: TOdaTanimForm;

implementation

{$R *.DFM}

procedure TOdaTanimForm.SpeedButton1Click(Sender: TObject);
begin
        OdaTable.FieldByName('Check').AsString:='True';
        OdaTable.Post;
        Close;
end;
```

```
procedure TOdaTanimForm.SpeedButton2Click(Sender: TObject);
begin
        Close;
end;

end.
```

Appendix B:

CODE OF ROOM LIST IN PROGRAM

```
unit OdaListeUnit;

interface

uses
          Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
          Db, DBTables, Refresh, Grids, DBGrids, StdCtrls, Buttons, ExtCtrls, Menus;

type
          TOdaListeForm = class(TForm)
            Panel1: TPanel;
            DegistirButon: TSpeedButton;
            YeniButon: TSpeedButton;
            SilButon: TSpeedButton;
            CikisButon: TSpeedButton;
            Label1: TLabel;
            SecButon: TSpeedButton;
            RoomNumber: TEdit;
            DBGrid1: TDBGrid;
            OdaTable: TSelfRefreshTable;
            OdaSource: TDataSource;
            OdaTableRoomNumber: TIntegerField;
            OdaTableFloor: TIntegerField;
            OdaTableBedNumber: TIntegerField;
            OdaTablePrice: TCurrencyField;
            OdaTableProperties: TStringField;
            OdaTableCheck: TBooleanField;
            MainMenu1: TMainMenu;
            TmOdalar1: TMenuItem;
            BoOdalar1: TMenuItem;
            procedure RoomNumberChange(Sender: TObject);
            procedure YeniButonClick(Sender: TObject);
            procedure DegistirButonClick(Sender: TObject);
            procedure CikisButonClick(Sender: TObject);
            procedure DBGrid1DblClick(Sender: TObject);
            procedure TmOdalar1Click(Sender: TObject);
            procedure BoOdalar1Click(Sender: TObject);
            procedure SilButonClick(Sender: TObject);
            procedure SecButonClick(Sender: TObject);
 private
   { Private declarations }
 public
   { Public declarations }
 end;
```

```
var
        OdaListeForm: TOdaListeForm;

implementation

uses OdaTanimUnit;

{$R *.DFM}

procedure TOdaListeForm.RoomNumberChange(Sender: TObject);
begin
        if (RoomNumber.Text='') then
          begin exit; end
        else
        OdaTable.Locate('RoomNumber',RoomNumber.text,[loCaseInsensitive,
        loPartialKey]);
end;

procedure TOdaListeForm.YeniButonClick(Sender: TObject);
begin
        OdaTanimForm.OdaTable.Append;
        OdaTanimForm.ShowModal;
end;

procedure TOdaListeForm.DegistirButonClick(Sender: TObject);
begin
        OdaTanimForm.OdaTable.GotoCurrent(OdaListeForm.OdaTable);
        OdaTanimForm.OdaTable.Edit;
        OdaTanimForm.ShowModal;
end;

procedure TOdaListeForm.CikisButonClick(Sender: TObject);
begin
        ModalResult:=MrCancel;
end;

procedure TOdaListeForm.DBGrid1DblClick(Sender: TObject);
begin
         Close;
end;

procedure TOdaListeForm.TmOdalar1Click(Sender: TObject);
begin
         OdaTable.Filter:='';
         OdaTable.Filtered:=True;
         OdaTable.Active:=fALSE;
         OdaTable.Active:=tRUE;
end;
```

```
procedure TOdaListeForm.BoOdalar1Click(Sender: TObject);
begin
          OdaTable.Filter:='';
          OdaTable.Filtered:=True;
          OdaTable.Filter:='Check='+chr(39)+'True'+chr(39);
end;

procedure TOdaListeForm.SilButonClick(Sender: TObject);
begin
        if messageDlg('Are You Sure You Want to Delete This Record?',
          mtconfirmation, [mbYes,MbNo],0)=idYes then
          OdaTable.Delete;
end;

procedure TOdaListeForm.SecButonClick(Sender: TObject);
begin
        ModalResult:=Mrok ;
end;

end.
```

Appendix C:

CODE OF DEFINITION OF PENSION IN PROGRAM

```
unit PansiyonTanimUnit;

interface

uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
        Db, DBTables, Refresh, StdCtrls, Mask, DBCtrls, Buttons, ExtCtrls;

type
        TPansiyonTanimForm = class(TForm)
          Panel1: TPanel;
          Label1: TLabel;
          Label2: TLabel;
          SpeedButton1: TSpeedButton;
          SpeedButton2: TSpeedButton;
          DBEdit1: TDBEdit;
          DBEdit2: TDBEdit;
          PansiyonTable: TSelfRefreshTable;
          PansiyonSource: TDataSource;
          procedure SpeedButton1Click(Sender: TObject);
          procedure SpeedButton2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
        PansiyonTanimForm: TPansiyonTanimForm;

implementation

{$R *.DFM}

procedure TPansiyonTanimForm.SpeedButton1Click(Sender: TObject);
begin
        PansiyonTable.Post;
        Close;
end;

procedure TPansiyonTanimForm.SpeedButton2Click(Sender: TObject);
begin
       Close;
end;
end.
```

Appendix D:

CODE OF PENSION LIST IN PROGRAM

```
unit PansiyonListeUnit;

interface

uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
        Grids, DBGrids, StdCtrls, Buttons, ExtCtrls, Db, DBTables, Refresh;

type
        TPansiyonListeForm = class(TForm)
          PansiyonTable: TSelfRefreshTable;
          PansiyonSource: TDataSource;
          Panel1: TPanel;
          DegistirButon: TSpeedButton;
          YeniButon: TSpeedButton;
          SilButon: TSpeedButton;
          CikisButon: TSpeedButton;
          Label1: TLabel;
          SecButon: TSpeedButton;
          KodEdit: TEdit;
          DBGrid1: TDBGrid;
          PansiyonTableCode: TStringField;
          PansiyonTableDescription: TStringField;
          procedure KodEditChange(Sender: TObject);
          procedure CikisButonClick(Sender: TObject);
          procedure DBGrid1DblClick(Sender: TObject);
          procedure YeniButonClick(Sender: TObject);
          procedure DegistirButonClick(Sender: TObject);
          procedure SilButonClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

 var
         PansiyonListeForm: TPansiyonListeForm;

 implementation

        uses PansiyonTanimUnit;

 {$R *.DFM}
```

```
procedure TPansiyonListeForm.KodEditChange(Sender: TObject);
begin
        if (KodEdit.Text='') then
          begin exit; end
        else
        PansiyonTable.Locate('Code',KodEdit.text,[loCaseInsensitive, loPartialKey]);
end;

procedure TPansiyonListeForm.CikisButonClick(Sender: TObject);
begin
        Close;
end;

procedure TPansiyonListeForm.DBGrid1DblClick(Sender: TObject);
begin
        Close;
end;

procedure TPansiyonListeForm.YeniButonClick(Sender: TObject);
begin
        PansiyonTanimForm.PansiyonTable.Append;
        PansiyonTanimForm.ShowModal;
 end;

procedure TPansiyonListeForm.DegistirButonClick(Sender: TObject);
begin

        PansiyonTanimForm.PansiyonTable.GotoCurrent(PansiyonListeForm.PansiyonTable);
          PansiyonTanimForm.PansiyonTable.Edit;
          PansiyonTanimForm.ShowModal;
end;

procedure TPansiyonListeForm.SilButonClick(Sender: TObject);
begin
        if messageDlg('Are You Sure You Want To Delete This Record?',
          mtInformation, [mbYes,MbNo],0)=idYes then
          PansiyonTable.Delete;
end;

end.
```

Appendix E:

CODE OF DEFINITION OF INCOMING WITH IN PROGRAM

```
unit    IncomingTanimUnit;

interface

uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
        Db, DBTables, Refresh, StdCtrls, Mask, DBCtrls, Buttons, ExtCtrls;

type
        TIncomingTanimForm = class(TForm)
         Panel1: TPanel;
         Label1: TLabel;
         Label2: TLabel;
         SpeedButton1: TSpeedButton;
         SpeedButton2: TSpeedButton;
         DBEdit1: TDBEdit;
         DBEdit2: TDBEdit;
         IncomingTable: TSelfRefreshTable;
         IncomingSource: TDataSource;
         procedure SpeedButton1Click(Sender: TObject);
         procedure SpeedButton2Click(Sender: TObject);
  private
   { Private declarations }
  public
   { Public declarations }
  end;

var
        IncomingTanimForm: TIncomingTanimForm;

implementation
{$R *.DFM}

procedure TIncomingTanimForm.SpeedButton1Click(Sender: TObject);
begin
        IncomingTable.Post;
        Close;
end;

procedure TIncomingTanimForm.SpeedButton2Click(Sender: TObject);
begin
        Close;
end;

end.
```

Appendix F:

CODE OF INCOMING WITH LIST IN PROGRAM

```
unit IncomingListeUnit;

interface

uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
        Grids, DBGrids, Db, DBTables, Refresh, StdCtrls, Buttons, ExtCtrls;

type
        TIncomingListeForm = class(TForm)
         Panel1: TPanel;
         DegistirButon: TSpeedButton;
         YeniButon: TSpeedButton;
         SilButon: TSpeedButton;
         CikisButon: TSpeedButton;
         Label1: TLabel;
         SecButon: TSpeedButton;
         KodEdit: TEdit;
         IncomingTable: TSelfRefreshTable;
         PansiyonSource: TDataSource;
         DBGrid1: TDBGrid;
         procedure KodEditChange(Sender: TObject);
         procedure CikisButonClick(Sender: TObject);
         procedure DBGrid1DblClick(Sender: TObject);
         procedure YeniButonClick(Sender: TObject);
         procedure DegistirButonClick(Sender: TObject);
         procedure SilButonClick(Sender: TObject);
 private
   { Private declarations }
 public
   { Public declarations }
 end;

var
        IncomingListeForm: TIncomingListeForm;

implementation

       uses IncomingTanimUnit;

{$R *.DFM}
```

```
procedure TIncomingListeForm.KodEditChange(Sender: TObject);
begin
          if (KodEdit.Text='') then
            begin exit; end
          else
          IncomingTable.Locate('Code',KodEdit.text,[loCaseInsensitive, loPartialKey]);
end;

procedure TIncomingListeForm.CikisButonClick(Sender: TObject);
begin
          Close;
end;

procedure TIncomingListeForm.DBGrid1DblClick(Sender: TObject);
begin
          Close;
end;

procedure TIncomingListeForm.YeniButonClick(Sender: TObject);
begin
           IncomingTanimForm.IncomingTable.Append;
           IncomingTanimForm.ShowModal;
end;

procedure TIncomingListeForm.DegistirButonClick(Sender: TObject);
begin
          IncomingTanimForm.IncomingTable.GotoCurrent(IncomingListeForm.IncomingTable);
          IncomingTanimForm.IncomingTable.Edit;
          IncomingTanimForm.ShowModal;
end;

procedure TIncomingListeForm.SilButonClick(Sender: TObject);
begin
          if messageDlg('Are You Sure You Want To Delete This Record?',
            mtInformation, [mbYes,MbNo],0)=idYes then
            IncomingTable.Delete;

end;

end.
```

Appendix G:

CODE OF DEFINITION OF COUNTRY IN PROGRAM

```
unit UlkeTanimUnit;

interface

uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
        Db, DBTables, Refresh, StdCtrls, Mask, DBCtrls, Buttons, ExtCtrls;

type
        TUlkeTanimForm = class(TForm)
         Panel1: TPanel;
         Label1: TLabel;
         Label2: TLabel;
         SpeedButton1: TSpeedButton;
         SpeedButton2: TSpeedButton;
         DBEdit1: TDBEdit;
         DBEdit2: TDBEdit;
         UlkeTable: TSelfRefreshTable;
         UlkeSource: TDataSource;
         procedure SpeedButton1Click(Sender: TObject);
         procedure SpeedButton2Click(Sender: TObject);
  private
   { Private declarations }
  public
   { Public declarations }
  end;
var
        UlkeTanimForm: TUlkeTanimForm;

implementation
{$R *.DFM}

procedure TUlkeTanimForm.SpeedButton1Click(Sender: TObject);
begin
        UlkeTable.Post;
        Close;
end;

procedure TUlkeTanimForm.SpeedButton2Click(Sender: TObject);
begin
        Close;
end;

end.
```

Appendix H:

CODE OF COUNTRY LIST IN PROGRAM

```
unit UlkeListeUnit;
interface
uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
        Db, DBTables, Refresh, Grids, DBGrids, StdCtrls, Buttons, ExtCtrls;
type
        TUlkeListeForm = class(TForm)
         Panel1: TPanel;
         DegistirButon: TSpeedButton;
         YeniButon: TSpeedButton;
         SilButon: TSpeedButton;
         CikisButon: TSpeedButton;
         Label1: TLabel;
         Label2: TLabel;
         SecButon: TSpeedButton;
         AdiEdit: TEdit;
         KodEdit: TEdit;
         DBGrid1: TDBGrid;
         UlkeTable: TSelfRefreshTable;
         UlkeSource: TDataSource;
         UlkeTableKod: TStringField;
         UlkeTableName: TStringField;
         procedure SecButonClick(Sender: TObject);
         procedure CikisButonClick(Sender: TObject);
         procedure DBGrid1DblClick(Sender: TObject);
         procedure KodEditChange(Sender: TObject);
         procedure AdiEditChange(Sender: TObject);
         procedure YeniButonClick(Sender: TObject);
         procedure DegistirButonClick(Sender: TObject);
         procedure SilButonClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

 var
        UlkeListeForm: TUlkeListeForm;

 implementation

        uses UlkeTanimUnit;

 {$R *.DFM}
```

```
procedure TUlkeListeForm.SecButonClick(Sender: TObject);
begin
        Close;
end;

procedure TUlkeListeForm.CikisButonClick(Sender: TObject);
begin
        Close;
end;

procedure TUlkeListeForm.DBGrid1DblClick(Sender: TObject);
begin
        Close;
end;

procedure TUlkeListeForm.KodEditChange(Sender: TObject);
begin
        if (KodEdit.Text='') then
          begin exit; end
        else
        UlkeTable.Locate('Kod',KodEdit.text,[loCaseInsensitive, loPartialKey]);
end;

procedure TUlkeListeForm.AdiEditChange(Sender: TObject);
begin
        if (AdiEdit.Text='') then
          begin exit; end
        else
         UlkeTable.Locate('Name',AdiEdit.text,[loCaseInsensitive, loPartialKey]);
end;

procedure TUlkeListeForm.YeniButonClick(Sender: TObject);
begin
        UlkeTanimForm.UlkeTable.Append;
        UlkeTanimForm.ShowModal;
end;

procedure TUlkeListeForm.DegistirButonClick(Sender: TObject);
begin
        UlkeTanimForm.UlkeTable.GotoCurrent(UlkeListeForm.UlkeTable);
        UlkeTanimForm.UlkeTable.Edit;
        UlkeTanimForm.ShowModal;
end;
procedure TUlkeListeForm.SilButonClick(Sender: TObject);
begin
         if messageDlg('Are You Sure You Want To Delete This Record?',
        mtInformation, [mbYes,MbNo],0)=idYes then
        UlkeTable.Delete;
end;    end.
```

Appendix I:

CODE OF DEFINITION OF CITY IN PROGRAM

```
unit SehirTanimUnit;

interface
uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
        Buttons, StdCtrls, Mask, DBCtrls, Db, DBTables, Refresh, ExtCtrls;

type
        TSehirTanimForm = class(TForm)
         Panel1: TPanel;
         Label1: TLabel;
         Label2: TLabel;
         SehirSource: TDataSource;
         SehirTable: TSelfRefreshTable;
         DBEdit1: TDBEdit;
         DBEdit2: TDBEdit;
         SpeedButton1: TSpeedButton;
         SpeedButton2: TSpeedButton;
         procedure SpeedButton1Click(Sender: TObject);
         procedure SpeedButton2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
        SehirTanimForm: TSehirTanimForm;

implementation

       uses SehirListeUnit;

{$R *.DFM}

procedure TSehirTanimForm.SpeedButton1Click(Sender: TObject);
begin
        SehirTable.Post;
         Close;
end;

procedure TSehirTanimForm.SpeedButton2Click(Sender: TObject);
begin
        Close;
end;

end.
```

Appendix J:

CODE OF CITY LIST IN PROGRAM

```
unit SehirListeUnit;

interface

uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
        StdCtrls, Grids, DBGrids, Db, DBTables, Refresh, Buttons, ExtCtrls, Menus;

type
        TSehirListeForm = class(TForm)
         Panel1: TPanel;
         DegistirButon: TSpeedButton;
         YeniButon: TSpeedButton;
         SilButon: TSpeedButton;
         CikisButon: TSpeedButton;
         AdiEdit: TEdit;
         DBGrid1: TDBGrid;
         KodEdit: TEdit;
         Label1: TLabel;
         Label2: TLabel;
         SehirTable: TSelfRefreshTable;
         SehirSource: TDataSource;
         SecButon: TSpeedButton;
         SehirTableAdi: TStringField;
         SehirTableKod: TStringField;
         procedure YeniButonClick(Sender: TObject);
         procedure DegistirButonClick(Sender: TObject);
         procedure SilButonClick(Sender: TObject);
         procedure KodEditChange(Sender: TObject);
         procedure AdiEditChange(Sender: TObject);
         procedure CikisButonClick(Sender: TObject);
         procedure DBGrid1DblClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
        SehirListeForm: TSehirListeForm;

implementation
        uses SehirTanimUnit;
{$R *.DFM}
```

```
procedure TSehirListeForm.YeniButonClick(Sender: TObject);
begin
            SehirTanimForm.SehirTable.Append;
            SehirTanimForm.ShowModal;
end;

procedure TSehirListeForm.DegistirButonClick(Sender: TObject);
begin
          SehirTanimForm.SehirTable.GotoCurrent(SehirListeForm.SehirTable);
          SehirTanimForm.SehirTable.Edit;
          SehirTanimForm.ShowModal;
end;

procedure TSehirListeForm.SilButonClick(Sender: TObject);
begin
           if messageDlg('Are You Sure You Want To Delete This Record?',
          mtInformation, [mbYes,MbNo],0)=idYes then
          SehirTable.Delete;
end;

procedure TSehirListeForm.KodEditChange(Sender: TObject);
begin
          if (KodEdit.Text='') then
            begin exit; end
          else
          SehirTable.Locate('Kod',KodEdit.text,[loCaseInsensitive, loPartialKey]);
end;

procedure TSehirListeForm.AdiEditChange(Sender: TObject);
begin
          if (AdiEdit.Text='') then
            begin exit; end
          else
           SehirTable.Locate('Adi',AdiEdit.text,[loCaseInsensitive, loPartialKey]);
end;

procedure TSehirListeForm.CikisButonClick(Sender: TObject);
begin
          Close;
end;

procedure TSehirListeForm.DBGrid1DblClick(Sender: TObject);
begin
          Close;
end;

end.
```

Appendix K:

CODE OF MAIN MENU IN PROGRAM

```
unit AnaFormUnit;

interface

uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
        Menus;

type
        TAnaForm = class(TForm)
          MainMenu1: TMainMenu;
          File1: TMenuItem;
          Rezer1: TMenuItem;
          Definition1: TMenuItem;
          Registration1: TMenuItem;
          City1: TMenuItem;
          Country1: TMenuItem;
          Room1: TMenuItem;
          Record1: TMenuItem;
          List1: TMenuItem;
          Report1: TMenuItem;
          Exit1: TMenuItem;
          Record2: TMenuItem;
          List2: TMenuItem;
          Report2: TMenuItem;
          Pension1: TMenuItem;
          IncomingWith1: TMenuItem;
          deme1: TMenuItem;
          Giri1: TMenuItem;
          Liste1: TMenuItem;
          procedure City1Click(Sender: TObject);
          procedure Room1Click(Sender: TObject);
          procedure Pension1Click(Sender: TObject);
          procedure IncomingWith1Click(Sender: TObject);
          procedure Record2Click(Sender: TObject);
          procedure List2Click(Sender: TObject);
          procedure Record1Click(Sender: TObject);
          procedure List1Click(Sender: TObject);
          procedure Giri1Click(Sender: TObject);
          procedure Liste1Click(Sender: TObject);
          procedure Exit1Click(Sender: TObject);
          procedure Country1Click(Sender: TObject);
          procedure Report1Click(Sender: TObject);
```

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

var
             AnaForm: TAnaForm;

implementation

           uses SehirListeUnit, OdaListeUnit, PansiyonListeUnit, IncomingListeUnit,
             MusteriKayitUnit, MusteriListeUnit, RezervasyonTanimUnit,
             RezervationListeUnit, KasaGirisUnit, KasaListeUnit, UlkeListeUnit, Unit1;

{$R *.DFM}

procedure TAnaForm.City1Click(Sender: TObject);
begin
            SehirListeForm.ShowModal;
end;

procedure TAnaForm.Room1Click(Sender: TObject);
begin
            OdaListeForm.ShowModal;
end;

procedure TAnaForm.Pension1Click(Sender: TObject);
begin
            PansiyonListeForm.ShowModal;
end;

procedure TAnaForm.IncomingWith1Click(Sender: TObject);
begin
            IncomingListeForm.ShowModal;
end;

procedure TAnaForm.Record2Click(Sender: TObject);
begin
            MusteriTanimForm.KayitTable.Append;
            MusteriTanimForm.ShowModal;
end;

procedure TAnaForm.List2Click(Sender: TObject);
begin
            MusteriKayitListesi.ShowModal;
end;
```

```
procedure TAnaForm.Record1Click(Sender: TObject);
begin
            RezervationTanimForm.RezervationTable.Append;
            RezervationTanimForm.ShowModal;
end;

procedure TAnaForm.List1Click(Sender: TObject);
begin
         Form1.Show;
end;

procedure TAnaForm.Giri1Click(Sender: TObject);
var
          x:integer;
begin
          KasaGirisForm.KasaTable.insert;
          KasaGirisForm.ShowModal;
end;

procedure TAnaForm.Liste1Click(Sender: TObject);
begin
         KasaListeForm.ShowModal;
end;

procedure TAnaForm.Exit1Click(Sender: TObject);
begin
         Close;
end;

procedure TAnaForm.Country1Click(Sender: TObject);
begin
         UlkeListeForm.ShowModal;
end;

procedure TAnaForm.Report1Click(Sender: TObject);
begin
         RezervationListeForm.ShowModal;
end;

end.
```

Appendix L:

CODE OF CUSTOMER RESERVATION IN PROGRAM

```
unit RezervasyonTanimUnit;

interface

uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
        StdCtrls, DBCtrls, Mask, Buttons, ExtCtrls, Db, DBTables, Refresh;

type
        TRezervationTanimForm = class(TForm)
         Panel1: TPanel;
         Label1: TLabel;
         Label2: TLabel;
         Label3: TLabel;
         SpeedButton1: TSpeedButton;
         SpeedButton2: TSpeedButton;
         Label6: TLabel;
         Label7: TLabel;
         Label8: TLabel;
         Label10: TLabel;
         Label12: TLabel;
         Label13: TLabel;
         SpeedButton4: TSpeedButton;
         SpeedButton6: TSpeedButton;
         DBEdit1: TDBEdit;
         DBEdit2: TDBEdit;
         DBEdit3: TDBEdit;
         DBEdit4: TDBEdit;
         DBEdit5: TDBEdit;
         IncomingWith: TDBEdit;
         Incoming: TDBEdit;
         Ulke: TDBEdit;
         DBMemo2: TDBMemo;
         RezervationTable: TSelfRefreshTable;
         RezervasyonSource: TDataSource;
         RezervationTableCode: TAutoIncField;
         RezervationTableRezervationDate: TDateField;
         RezervationTableNameSurname: TStringField;
         RezervationTableDateOfArrival: TDateField;
         RezervationTableDateOfDeparture: TDateField;
         RezervationTablePension: TStringField;
         RezervationTableIncomingWith: TStringField;
         RezervationTableCountry: TStringField;
         RezervationTableRezervationWho: TStringField;
```

```
            RezervationTableNot: TStringField;
            RezervationTableCity: TStringField;
            Label4: TLabel;
            SpeedButton3: TSpeedButton;
            City: TDBEdit;
            Label5: TLabel;
            DBEdit6: TDBEdit;
            SpeedButton5: TSpeedButton;
            RezervationTableRoomNumber: TIntegerField;
            procedure SpeedButton1Click(Sender: TObject);
            procedure SpeedButton2Click(Sender: TObject);
            procedure SpeedButton4Click(Sender: TObject);
            procedure SpeedButton6Click(Sender: TObject);
            procedure SpeedButton3Click(Sender: TObject);
            procedure SpeedButton5Click(Sender: TObject);
            procedure DBEdit1KeyPress(Sender: TObject; var Key: Char);
            procedure DBEdit2KeyPress(Sender: TObject; var Key: Char);
            procedure DBEdit3KeyPress(Sender: TObject; var Key: Char);
            procedure DBEdit4KeyPress(Sender: TObject; var Key: Char);
            procedure DBEdit5KeyPress(Sender: TObject; var Key: Char);
            procedure IncomingWithKeyPress(Sender: TObject; var Key: Char);
            procedure UlkeKeyPress(Sender: TObject; var Key: Char);
            procedure CityKeyPress(Sender: TObject; var Key: Char);
            procedure IncomingKeyPress(Sender: TObject; var Key: Char);
            procedure DBEdit6KeyPress(Sender: TObject; var Key: Char);
            procedure DBMemo2KeyPress(Sender: TObject; var Key: Char);

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

var
          RezervationTanimForm: TRezervationTanimForm;

implementation
uses    PansiyonListeUnit, UlkeListeUnit, IncomingListeUnit;

{$R *.DFM}

procedure TRezervationTanimForm.SpeedButton1Click(Sender: TObject);
begin
        RezervationTable.Post;
        Close;
end;
```

```
procedure TRezervationTanimForm.SpeedButton2Click(Sender: TObject);
begin
        Close;
end;



procedure TRezervationTanimForm.SpeedButton5Click(Sender: TObject);
begin
        PansiyonListeForm.ShowModal;
        Pansiyon.Text:=PansiyonListeForm.PansiyonTable.FieldValues['Code'];
end;

procedure TRezervationTanimForm.SpeedButton4Click(Sender: TObject);
begin
        UlkeListeForm.ShowModal;
        Ulke.Text:=UlkeListeForm.UlkeTable.FieldValues['Kod'];
end;

procedure TRezervationTanimForm.SpeedButton6Click(Sender: TObject);
begin
        INcomingListeForm.ShowModal;
        IncomingWith.Text:=INcomingListeForm.IncomingTable.FieldValues['Code'];
end;

procedure TRezervationTanimForm.DBEdit1KeyPress(Sender: TObject;
  var Key: Char);
begin
       If key=#13 then DBEdit2.SetFocus;
end;

procedure TRezervationTanimForm.DBEdit2KeyPress(Sender: TObject;
  var Key: Char);
begin
       If key=#13 then DBEdit3.SetFocus;
end;
procedure TRezervationTanimForm.DBEdit3KeyPress(Sender: TObject;
  var Key: Char);
begin
       If key=#13 then DBEdit4.SetFocus;
end;

procedure TRezervationTanimForm.DBEdit4KeyPress(Sender: TObject;
  var Key: Char);
begin
       If key=#13 then DBEdit5.SetFocus;
end;
```

```
procedure TRezervationTanimForm.DBEdit5KeyPress(Sender: TObject;
  var Key: Char);
begin
        If key=#13 then IncomingWith.SetFocus;
end;

procedure TRezervationTanimForm.IncomingWithKeyPress(Sender: TObject;
  var Key: Char);
begin
        If key=#13 then Ulke.SetFocus;
end;

procedure TRezervationTanimForm.UlkeKeyPress(Sender: TObject;
  var Key: Char);
begin
        If key=#13 then City.SetFocus;
end;

procedure TRezervationTanimForm.CityKeyPress(Sender: TObject;
  var Key: Char);
begin
        If key=#13 then Incoming.SetFocus;
end;

procedure TRezervationTanimForm.IncomingKeyPress(Sender: TObject;
  var Key: Char);
begin
        If key=#13 then DBEdit6.SetFocus;
end;

procedure TRezervationTanimForm.DBEdit6KeyPress(Sender: TObject;
  var Key: Char);
begin
        If key=#13 then DBMemo2.SetFocus;
end;

procedure TRezervationTanimForm.DBMemo2KeyPress(Sender: TObject;
  var Key: Char);
begin
        If key=#13 then DBEdit1.SetFocus;
end;

end.
```

Appendix M:

CODE OF CUSTOMER RESERVATION LIST IN PROGRAM

```
unit Unit1;

interface

uses
          Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
          Db, DBTables, Refresh, Grids, DBGrids, StdCtrls, Buttons, ExtCtrls;

type
          TForm1 = class(TForm)
            Panel1: TPanel;
            DegistirButon: TSpeedButton;
            YeniButon: TSpeedButton;
            SilButon: TSpeedButton;
            CikisButon: TSpeedButton;
            Label1: TLabel;
            Label2: TLabel;
            SecButon: TSpeedButton;
            AdiEdit: TEdit;
            KodEdit: TEdit;
            DBGrid1: TDBGrid;
            RezervationTable: TSelfRefreshTable;
            RezervationSource: TDataSource;
            RezervationTableCode: TAutoIncField;
            RezervationTableRezervationDate: TDateField;
            RezervationTableNameSurname: TStringField;
            RezervationTableDateOfArrival: TDateField;
            RezervationTableDateOfDeparture: TDateField;
            RezervationTableIncomingWith: TStringField;
            RezervationTableCountry: TStringField;
            RezervationTableRezervationWho: TStringField;
            RezervationTableNot: TStringField;
            procedure YeniButonClick(Sender: TObject);
            procedure DegistirButonClick(Sender: TObject);
            procedure SecButonClick(Sender: TObject);
            procedure SilButonClick(Sender: TObject);
            procedure KodEditChange(Sender: TObject);
            procedure AdiEditChange(Sender: TObject);
            procedure CikisButonClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```
var
            Form1: TForm1;

implementation
uses RezervasyonTanimUnit, RezervationListeUnit;
{$R *.DFM}

procedure TForm1.YeniButonClick(Sender: TObject);
begin
             RezervationTanimForm.RezervationTable.Append;
             RezervationTanimForm.ShowModal;
end;

procedure TForm1.DegistirButonClick(Sender: TObject);
begin
          RezervationTanimForm.RezervationTable.GotoCurrent(RezervationListeForm.Rezerv
          ationTable);
           RezervationTanimForm.RezervationTable.Edit;
           RezervationTanimForm.ShowModal;
end;

procedure TForm1.SecButonClick(Sender: TObject);
begin
          ModalResult:=MrOk;
end;

procedure TForm1.SilButonClick(Sender: TObject);
begin
          if messageDlg('Are You Sure You Want to Delete This Record?',
           mtconfirmation, [mbYes,MbNo],0)=idYes then
           RezervationTable.Delete;
end;

procedure TForm1.KodEditChange(Sender: TObject);
begin
          if (kodEdit.Text='') then
             begin exit; end
           else
                RezervationTable.Locate('Code',KodEdit.text,[loCaseInsensitive, loPartialKey]);
end;

procedure TForm1.AdiEditChange(Sender: TObject);
begin
          if (AdiEdit.Text='') then
             begin exit; end
           else
           RezervationTable.Locate('NameSurname',AdiEdit.text,[loCaseInsensitive,
          loPartialKey]);
end;
```

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

end.
```

Appendix N:

CODE OF REPORT OF RESERVATION IN PROGRAM

```
unit RezervationReportUnit;

interface

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

type
        TRezervasyonRaporForm = class(TForm)
          Panel1: TPanel;
          Goster: TSpeedButton;
          Cikis: TSpeedButton;
          Label26: TLabel;
          Label2: TLabel;
          Label3: TLabel;
          Sec: TSpeedButton;
          Label4: TLabel;
          SpeedButton1: TSpeedButton;
          SpeedButton2: TSpeedButton;
          SpeedButton6: TSpeedButton;
          SpeedButton7: TSpeedButton;
          Country: TEdit;
          NS: TEdit;
          Code1: TEdit;
          Code2: TEdit;
          Date1: TMaskEdit;
          Date2: TMaskEdit;
          procedure GosterClick(Sender: TObject);
          procedure CikisClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
        RezervasyonRaporForm: TRezervasyonRaporForm;

implementation

        uses RezervationListeUnit;

{$R *.DFM}
```

```
procedure TRezervasyonRaporForm.GosterClick(Sender: TObject);
begin
        RezervationListeForm.RezervationTable.Filter:=";
        RezervationListeForm.RezervationTable.Filtered:=True;
        if (Code1.Text<>") AND (Code2.Text=") then
        RezervationListeForm.RezervationTable.Filter:='Code='+chr(39)+Code1.text+chr(39)
        ;

        if (Code1.Text<>") and  (Code2.Text<>") and
        (RezervationListeForm.RezervationTable.Filter<>") then
        RezervationListeForm.RezervationTable.Filter:=RezervationListeForm.RezervationTa
        ble.Filter+'and  Code>='+chr(39)+Code1.Text+chr(39)+' and
        Code<='+chr(39)+Code2.Text+chr(39)
        else if (Code1.Text<>") and  (Code2.Text<>") and
        (RezervationListeForm.RezervationTable.Filter=") then
        RezervationListeForm.RezervationTable.Filter:='Code>='+chr(39)+Code1.Text+chr(3
        9)+' and Code<='+chr(39)+Code2.Text+chr(39);

        if (NS.Text<>") and (RezervationListeForm.RezervationTable.Filter<>") then

        RezervationListeForm.RezervationTable.Filter:=RezervationListeForm.RezervationTa
        ble.Filter+  'and NameSurname='+chr(39)+NS.text+chr(39)
        else if (NS.Text<>") and (RezervationListeForm.RezervationTable.Filter=") then
        RezervationListeForm.RezervationTable.Filter:='NameSurname='+chr(39)+NS.text+c
        hr(39);

        if (Country.Text<>") and (RezervationListeForm.RezervationTable.Filter<>") then

        RezervationListeForm.RezervationTable.Filter:=RezervationListeForm.RezervationTa
        ble.Filter+  'and Country='+chr(39)+Country.text+chr(39)
        else if (Country.Text<>") and (RezervationListeForm.RezervationTable.Filter=") then
        RezervationListeForm.RezervationTable.Filter:='Country='+chr(39)+Country.text+chr
        (39);

        if (Date1.Text<>'  .  .    ') and  (DAte2.Text<>'  .  .    ') and
        (RezervationListeForm.RezervationTable.Filter<>") then

        RezervationListeForm.RezervationTable.Filter:=RezervationListeForm.RezervationTa
        ble.Filter+'and  RezervationDate>='+chr(39)+DAte1.Text+chr(39)+' and
        RezervationDate<='+chr(39)+Date2.Text+chr(39)
        else if (Date1.Text<>'  .  .    ') and  (Date1.Text<>'  .  .    ') and
        (RezervationListeForm.RezervationTable.Filter=") then
        RezervationListeForm.RezervationTable.Filter:='RezervationDate>='+chr(39)+DAte1.
        Text+chr(39)+' and RezervationDate<='+chr(39)+Date2.Text+chr(39);

 end;
```

```
procedure TRezervasyonRaporForm.CikisClick(Sender: TObject);
begin
        Close;
end;

end.
```

Appendix O:

CODE OF CUSTOMER REGISTRATION IN PROGRAM

```
unit MusteriKayitUnit;

interface

uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
        Db, DBTables, Refresh, Buttons, StdCtrls, ExtCtrls, DBCtrls, Mask;

type
        TMusteriTanimForm = class(TForm)
         Panel1: TPanel;
         Label1: TLabel;
         Label2: TLabel;
         Label3: TLabel;
         Label4: TLabel;
         Label5: TLabel;
         SpeedButton1: TSpeedButton;
         SpeedButton2: TSpeedButton;
         KayitTable: TSelfRefreshTable;
         KayitSource: TDataSource;
         Label6: TLabel;
         Label7: TLabel;
         Label8: TLabel;
         Label9: TLabel;
         Label10: TLabel;
         Label11: TLabel;
         Label12: TLabel;
         Label13: TLabel;
         Label14: TLabel;
         Code: TDBEdit;
         NameSurname: TDBEdit;
         PN: TDBEdit;
         IncDate: TDBEdit;
         LvDate: TDBEdit;
         Pansiyon: TDBEdit;
         RN: TDBEdit;
         Price: TDBEdit;
         Incoming: TDBEdit;
         Age: TDBEdit;
         City: TDBEdit;
         Ulke: TDBEdit;
         Adres: TDBMemo;
         DBMemo2: TDBMemo;
        SpeedButton3: TSpeedButton;
        SpeedButton4: TSpeedButton;
        SpeedButton5: TSpeedButton;
        SpeedButton6: TSpeedButton;
        SpeedButton7: TSpeedButton;
        KayitTableKod: TAutoIncField;
        KayitTableNameSurname: TStringField;
        KayitTablePassportNo: TStringField;
        KayitTableAge: TIntegerField;
        KayitTableOccupation: TStringField;
        KayitTableAdress: TStringField;
        KayitTableCity: TStringField;
        KayitTableCountry: TStringField;
        KayitTableIncomingDate: TDateField;
        KayitTableLeavingDate: TDateField;
        KayitTablePension: TStringField;
        KayitTableRoomNumber: TIntegerField;
        KayitTablePrice: TCurrencyField;
        KayitTableIncomingWith: TStringField;
        KayitTableNot: TStringField;
        SpeedButton8: TSpeedButton;
        Label15: TLabel;
        DBMemo1: TDBMemo;
        KayitTableAccompaniedBy: TStringField;
        procedure SpeedButton5Click(Sender: TObject);
        procedure SpeedButton6Click(Sender: TObject);
        procedure SpeedButton3Click(Sender: TObject);
        procedure SpeedButton4Click(Sender: TObject);
        procedure SpeedButton1Click(Sender: TObject);
        procedure SpeedButton2Click(Sender: TObject);
        procedure SpeedButton7Click(Sender: TObject);
        procedure SpeedButton8Click(Sender: TObject);
        procedure CodeKeyPress(Sender: TObject; var Key: Char);
        procedure NameSurnameKeyPress(Sender: TObject; var Key: Char);
        procedure PNKeyPress(Sender: TObject; var Key: Char);
        procedure IncDateKeyPress(Sender: TObject; var Key: Char);
        procedure LvDateKeyPress(Sender: TObject; var Key: Char);
        procedure PansiyonKeyPress(Sender: TObject; var Key: Char);
        procedure RNKeyPress(Sender: TObject; var Key: Char);
        procedure PriceKeyPress(Sender: TObject; var Key: Char);
        procedure IncomingKeyPress(Sender: TObject; var Key: Char);
        procedure AgeKeyPress(Sender: TObject; var Key: Char);
        procedure UlkeKeyPress(Sender: TObject; var Key: Char);
        procedure CityKeyPress(Sender: TObject; var Key: Char);
        procedure DBMemo1KeyPress(Sender: TObject; var Key: Char);
        procedure AdresKeyPress(Sender: TObject; var Key: Char);
        procedure DBMemo2KeyPress(Sender: TObject; var Key: Char);
private
        { Private declarations }
```

```
public
        { Public declarations }
  end;


var
      MusteriTanimForm: TMusteriTanimForm;

implementation

     uses PansiyonListeUnit, IncomingListeUnit, OdaListeUnit, UlkeListeUnit,
      SehirListeUnit, RezervationListeUnit, Unit1;

{$R *.DFM}

procedure TMusteriTanimForm.SpeedButton5Click(Sender: TObject);
begin
     PansiyonListeForm.ShowModal;
     Pansiyon.Text:=PansiyonListeForm.PansiyonTable.FieldValues['Code'];
end;

procedure TMusteriTanimForm.SpeedButton6Click(Sender: TObject);
begin
     IncomingListeForm.ShowModal;
     Incoming.Text:=IncomingListeForm.IncomingTable.FieldValues['Code'];
end;

procedure TMusteriTanimForm.SpeedButton3Click(Sender: TObject);
begin
     SehirListeForm.ShowModal;
     City.Text:=SehirListeForm.SehirTable.FieldValues['Adi'];

end;

procedure TMusteriTanimForm.SpeedButton4Click(Sender: TObject);
begin
     UlkeListeForm.ShowModal;
     Ulke.Text:=UlkeListeForm.UlkeTable.FieldValues['Name'];
end;

procedure TMusteriTanimForm.SpeedButton1Click(Sender: TObject);
begin
     KayitTable.Post;
     OdaListeForm.OdaTable.Edit;
     OdaListeForm.OdaTable.FieldByName('Check').AsString:='False';
     OdaListeForm.OdaTable.Post;
     Close;
end;
```

```
procedure TMusteriTanimForm.SpeedButton2Click(Sender: TObject);
begin
        Close;
end;

procedure TMusteriTanimForm.SpeedButton7Click(Sender: TObject);
begin
 RezervationListeForm.ShowModal;

NameSurname.Text:=RezervationListeForm.RezervationTable.FieldByname('NameSurname').
AsString;

IncDate.Text:=RezervationListeForm.RezervationTable.FieldByname('DAteOfArrival').AsStri
ng;

LvDate.Text:=RezervationListeForm.RezervationTable.FieldByname('DAteOfDeparture').AsS
tring;

Incoming.Text:=RezervationListeForm.RezervationTable.FieldByname('IncomingWith').AsStr
ing;
 Ulke.Text:=RezervationListeForm.RezervationTable.FieldByname('Country').AsString;
end;

procedure TMusteriTanimForm.SpeedButton8Click(Sender: TObject);
begin
         OdaListeForm.OdaTable.Filter:='';
         OdaListeForm.OdaTable.Filtered:=True;
         OdaListeForm.OdaTable.Filter:='Check='+chr(39)+'True'+chr(39);
         OdaListeForm.ShowModal;
         if odalisteForm.ModalResult=Mrok then
          begin
           KayitTable.FieldByName('Room
        Number').AsString:=OdaListeForm.OdaTable.FieldByName('RoomNumber').AsString;
           KayitTable.FieldByName('Price').AsString:=OdaListeForm.OdaTAble.FieldByName
        ('Price').AsString;
  end;
end;

procedure TMusteriTanimForm.CodeKeyPress(Sender: TObject; var Key: Char);
begin
        If key=#13 then NameSurname.SetFocus;
end;

procedure TMusteriTanimForm.NameSurnameKeyPress(Sender: TObject;
  var Key: Char);
begin
        If key=#13 then PN.SetFocus;
end;
```

```
procedure TMusteriTanimForm.PNKeyPress(Sender: TObject; var Key: Char);
begin
        If key=#13 then IncDate.SetFocus;
end;


procedure TMusteriTanimForm.IncDateKeyPress(Sender: TObject;
  var Key: Char);
begin
        If key=#13 then LvDate.SetFocus;
end;

procedure TMusteriTanimForm.LvDateKeyPress(Sender: TObject; var Key: Char);
begin
        If key=#13 then Pansiyon.SetFocus;
end;

procedure TMusteriTanimForm.PansiyonKeyPress(Sender: TObject;
  var Key: Char);
begin
        If key=#13 then RN.SetFocus;
end;

procedure TMusteriTanimForm.RNKeyPress(Sender: TObject; var Key: Char);
begin
        If key=#13 then Price.SetFocus;
end;

procedure TMusteriTanimForm.PriceKeyPress(Sender: TObject; var Key: Char);
begin
        If key=#13 then Incoming.SetFocus;
end;

procedure TMusteriTanimForm.IncomingKeyPress(Sender: TObject;
  var Key: Char);
begin
        If key=#13 then Age.SetFocus;
end;


procedure TMusteriTanimForm.AgeKeyPress(Sender: TObject; var Key: Char);
begin
        If key=#13 then Ulke.SetFocus;
end;

procedure TMusteriTanimForm.UlkeKeyPress(Sender: TObject; var Key: Char);
begin
        If key=#13 then City.SetFocus;
end;
```

```
procedure TMusteriTanimForm.CityKeyPress(Sender: TObject; var Key: Char);
begin
        If key=#13 then DBMemo1.SetFocus;
end;


procedure TMusteriTanimForm.DBMemo1KeyPress(Sender: TObject;
  var Key: Char);
begin
        If key=#13 then Adres.SetFocus;
end;

procedure TMusteriTanimForm.AdresKeyPress(Sender: TObject; var Key: Char);
begin
        If key=#13 then DBMemo2.SetFocus;
end;

procedure TMusteriTanimForm.DBMemo2KeyPress(Sender: TObject;
  var Key: Char);
begin
        If key=#13 then Code.SetFocus;
end;

end.
```

Appendix P:

CODE OF CUSTOMER REGISTRATION LIST IN PROGRAM

```
unit MusteriListeUnit;

interface

uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
        Grids, DBGrids, StdCtrls, Buttons, ExtCtrls, Db, DBTables, Refresh, Menus;

type
        TMusteriKayitListesi = class(TForm)
         KayitTable: TSelfRefreshTable;
         KayitSource: TDataSource;
         Panel1: TPanel;
         DegistirButon: TSpeedButton;
         YeniButon: TSpeedButton;
         SilButon: TSpeedButton;
         CikisButon: TSpeedButton;
         Label1: TLabel;
         Label2: TLabel;
         SecButon: TSpeedButton;
         AdiEdit: TEdit;
         KodEdit: TEdit;
         DBGrid1: TDBGrid;
         MainMenu1: TMainMenu;
         BoOdaListesi1: TMenuItem;
         procedure YeniButonClick(Sender: TObject);
         procedure DegistirButonClick(Sender: TObject);
         procedure SilButonClick(Sender: TObject);
         procedure CikisButonClick(Sender: TObject);
         procedure SecButonClick(Sender: TObject);
         procedure DBGrid1DblClick(Sender: TObject);
         procedure KodEditChange(Sender: TObject);
         procedure AdiEditChange(Sender: TObject);
         procedure BoOdaListesi1Click(Sender: TObject);
  private
   { Private declarations }
  public
   { Public declarations }
  end;

var  MusteriKayitListesi: TMusteriKayitListesi;

implementation
```

```
uses MusteriKayitUnit, OdaListeUnit;


{$R *.DFM}

procedure TMusteriKayitListesi.YeniButonClick(Sender: TObject);
begin
          MusteriTanimForm.KayitTable.Append;
          MusteriTanimForm.ShowModal;
end;

procedure TMusteriKayitListesi.DegistirButonClick(Sender: TObject);
begin
          MusteriTanimForm.KayitTable.GotoCurrent(MusteriKayitListesi.KayitTable);
          MusteriTanimForm.KayitTable.Edit;
          MusteriTanimForm.ShowModal;
end;

procedure TMusteriKayitListesi.SilButonClick(Sender: TObject);
begin
         if messageDlg('Are You Sure You Want to Delete This Record?',
          mtconfirmation, [mbYes,MbNo],0)=idYes then
          KayitTable.Delete;
end;

procedure TMusteriKayitListesi.CikisButonClick(Sender: TObject);
begin
          ModalResult:=MrCancel;
end;

procedure TMusteriKayitListesi.SecButonClick(Sender: TObject);
begin
          ModalResult:=MrOk;
end;

procedure TMusteriKayitListesi.DBGrid1DblClick(Sender: TObject);
begin
          ModalResult:=MrOk;
end;

procedure TMusteriKayitListesi.KodEditChange(Sender: TObject);
begin
         if (kodEdit.Text='') then
           begin exit; end
          else
          KayitTable.Locate('Kod',KodEdit.text,[loCaseInsensitive, loPartialKey]);
end;
```

```
procedure TMusteriKayitListesi.AdiEditChange(Sender: TObject);
begin
        if (AdiEdit.Text='') then
          begin exit; end
         else
         KayitTable.Locate('NameSurname',AdiEdit.text,[loCaseInsensitive, loPartialKey]);
end;

procedure TMusteriKayitListesi.BoOdaListesi1Click(Sender: TObject);
begin
          OdaListeForm.OdaTable.Filter:='';
          OdaListeForm.OdaTable.Filtered:=True;
          OdaListeForm.OdaTable.Filter:='Check='+chr(39)+'True'+chr(39);
          OdaListeForm.ShowModal;
end;

end.
```

Appendix Q:

CODE OF ENTERING ACCOUNT IN PROGRAM

```
unit KasaGirisUnit;

interface

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

type
      TKasaGirisForm = class(TForm)
        Panel1: TPanel;
        Panel2: TPanel;
        DBGrid1: TDBGrid;
        Cd: TDBEdit;
        NS: TDBEdit;
        Add: TSpeedButton;
        Change: TSpeedButton;
        Delete: TSpeedButton;
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        SpeedButton4: TSpeedButton;
        SpeedButton5: TSpeedButton;
        SpeedButton6: TSpeedButton;
        KasaTable: TSelfRefreshTable;
        KasaSource: TDataSource;
        Panel3: TPanel;
        Label4: TLabel;
        KasaDetayTable: TSelfRefreshTable;
        KasaDetaySource: TDataSource;
        Label5: TLabel;
        RoomNo: TDBEdit;
        KasaNo: TDBEdit;
        SpeedButton1: TSpeedButton;
        Label7: TLabel;
        UcretEkle: TSpeedButton;
        KayitTable: TSelfRefreshTable;
        KayitSource: TDataSource;
        IncomingDate: TDBEdit;
        KayitTableKod: TAutoIncField;
        KayitTableNameSurname: TStringField;
        KayitTablePassportNo: TStringField;
        KayitTableAge: TIntegerField;
        KayitTableOccupation: TStringField;
        KayitTableAdress: TStringField;
        KayitTableCity: TStringField;
        KayitTableCountry: TStringField;
        KayitTableIncomingDate: TDateField;
        KayitTableLeavingDate: TDateField;
        KayitTablePension: TStringField;
        KayitTableRoomNumber: TIntegerField;
        KayitTablePrice: TCurrencyField;
        KayitTableIncomingWith: TStringField;
        KayitTableNot: TStringField;
        Label6: TLabel;
        LeaveDate: TDBEdit;
        Label8: TLabel;
        Price: TDBEdit;
        total: TEdit;
        KasaTableKasaNo: TAutoIncField;
        KasaTableDate: TDateField;
        KasaTableCustomerCode: TStringField;
        KasaTableNameSurname: TStringField;
        KasaTableRoomNo: TIntegerField;
        KasaTableTotal: TCurrencyField;
        KasaDetayTableDetayNo: TAutoIncField;
        KasaDetayTableKasaNo: TStringField;
        KasaDetayTableMasrafAdi: TStringField;
        KasaDetayTableMasrafMiktari: TCurrencyField;
        KasaDetayTableAciklama: TMemoField;
        procedure SpeedButton6Click(Sender: TObject);
        procedure SpeedButton5Click(Sender: TObject);
        procedure SpeedButton4Click(Sender: TObject);
        procedure AddClick(Sender: TObject);
        procedure DeleteClick(Sender: TObject);
        procedure ChangeClick(Sender: TObject);
        procedure SpeedButton1Click(Sender: TObject);
        procedure UcretEkleClick(Sender: TObject);
        procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
      KasaGirisForm: TKasaGirisForm;

implementation

     uses MusteriListeUnit, KasaDetayUnit, FaturaBasimiRapor;
```

```
{$R *.DFM}


procedure TKasaGirisForm.SpeedButton6Click(Sender: TObject);
begin
          MusteriKayitListesi.ShowModal;
          Cd.Text:=MusteriKayitListesi.KayitTable.FieldByName('Kod').AsString;
          NS.Text:=MusteriKayitListesi.KayitTable.FieldByName('NameSurname').AsString;
          RoomNo.Text:=MusteriKayitListesi.KayitTable.FieldByName('Room
        Number').AsString;
          IncomingDate.Text:=MusteriKayitListesi.KayitTable.FieldByName('Incoming
        Date').AsString;
          Price.Text:=MusteriKayitListesi.KayitTable.FieldByName('Price').AsString;
end;

procedure TKasaGirisForm.SpeedButton5Click(Sender: TObject);
begin
        Close;
end;

procedure TKasaGirisForm.SpeedButton4Click(Sender: TObject);
begin
        KasaTable.Post;
        UcretEkle.Enabled:=True;
end;

procedure TKasaGirisForm.AddClick(Sender: TObject);
var
        x:integer;
begin
        KasaDetayForm.KasaDetayTable.Last;
        KasaDetayForm.KasaDetayTable.Append;
        KasaDetayForm.KasaNo.Text:=KasaNo.Text;
        KasaDetayForm.ShowModal;
end;
procedure TKasaGirisForm.DeleteClick(Sender: TObject);
begin
        if messageDlg('Are You Sure You Want To Delete This Record?',
        mtInformation, [mbYes,MbNo],0)=idYes then
        KasaDetayTable.Delete;
end;

procedure TKasaGirisForm.ChangeClick(Sender: TObject);
begin
        KasaDetayForm.KasaDetayTable.GotoCurrent(KasaGirisForm.KasadETAYTable);
        KasaDetayForm.KasaDetayTable.Edit;
        KasaDetayForm.ShowModal;
end;
```

```
procedure TKasaGirisForm.SpeedButton1Click(Sender: TObject);
begin
        FaturaBasimi.Toplam.Caption:=Total.Text;
        FaturaBasimi.Preview;   end;
procedure TKasaGirisForm.UcretEkleClick(Sender: TObject);
var
        fark,x,y,t,ayfark,s,k:integer; ay,gun,yil,ay2,gun2,yil2:word;   tarih1, tarih2:TDate;
        z,count,i,toplam:integer;
begin
        if (leaveDate.text='') then
 begin
         Showmessage('Please Check The Leaving Date Field');
         LeaveDate.SetFocus;
        end
        else
        tarih1:=StrToDate(IncomingDate.Text);
        DecodeDate(tarih1,yil,ay,gun);
        tarih2:=StrToDate(LeaveDate.Text);
        DecodeDate(tarih2,yil2,ay2,gun2);
        if (gun2>gun) and  (ay2=ay) then
begin
         x:=gun2-gun;
        end
        else if (ay2>ay) and (gun>gun2) then
 begin
         ayfark:=ay2-ay;
         x:=(Ayfark*30)+(gun2-gun);
        end
        else if (ay2>ay) and (gun<gun2) then
 begin
         ayfark:=ay2-ay;
         x:=(Ayfark*30)+(gun-gun2);
        end
        else if (ay2>ay) and (gun=gun2) then
 begin
         ayfark:=ay2-ay;
         x:=(Ayfark*30);
end;
 begin
        y:=KayitTable.FieldValues['Price'];
        fark:=y*x;
        showmessage(IntToStr(x));
        KasaDetayTable.Append;
        KasaDetayTable.FieldByName('KasaNo').AsString:=KasaNo.Text;
        KasaDetayTable.FieldByName('MasrafAdi').AsString:='ODA ÜCRETİ';
        KasaDetayTable.FieldByName('MasrafMiktari').AsString:=IntToStr(fark);
        KasaDetayTable.FieldByName('Aciklama').AsString:='ODA ÜCRETİ';
```

```
           KasaDetayTable.Post;
           KasaTable.Edit;
          KasaTable.FieldByName('Total').AsInteger:=KasaDetayForm.KasaDetayTable.FieldBy
          Name('MasrafMiktari').AsInteger +
          KasaDetayForm.KasaDetayTable.FieldByName('MasrafMiktari').AsInteger;


begin
           count:=KasaDetayTable.RecordCount;
           toplam:=0;
           for i:=1 to count do
  begin
             z:=KasaDetayTable.FieldByName('MasrafMiktari').AsInteger;
             toplam:=toplam+z;
             total.Text:=IntToStr(toplam);
end;
end;
end;

procedure TKasaGirisForm.FormShow(Sender: TObject);
var
           y,x,i,toplam:integer;
begin
           x:=KasaDetayTable.RecordCount;
           toplam:=0;
           KasaDetayTable.First;
           for i:=1 to x do
  begin
             y:=KasaDetayTable.FieldByName('MasrafMiktari').AsInteger;
             toplam:=toplam+y;
             KasaDetayTable.Next;
             ShowMessage(IntToStr(Toplam));
  end;
            total.Text:=IntToStr(toplam);
end;

end.
```

Appendix R:

CODE OF DETAIL OF ACCOUNT IN PROGRAM

```
unit KasaDetayUnit;

interface

uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
        Db, DBTables, Refresh, StdCtrls, DBCtrls, Mask, Buttons, ExtCtrls;

type
        TKasaDetayForm = class(TForm)
          Panel1: TPanel;
          SpeedButton4: TSpeedButton;
          SpeedButton5: TSpeedButton;
          Label1: TLabel;
          Label2: TLabel;
          Label3: TLabel;
          Label4: TLabel;
          Label5: TLabel;
          MM: TDBEdit;
          MA: TDBEdit;
          Aciklama: TDBMemo;
          KasaDetayTable: TSelfRefreshTable;
          KasaDetaySource: TDataSource;
          No: TEdit;
          KasaNo: TEdit;
          KasaDetayTableDetayNo: TAutoIncField;
          KasaDetayTableKasaNo: TStringField;
          KasaDetayTableMasrafAdi: TStringField;
          KasaDetayTableMasrafMiktari: TCurrencyField;
          KasaDetayTableAciklama: TMemoField;
          procedure SpeedButton5Click(Sender: TObject);
          procedure SpeedButton4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

 var
  KasaDetayForm: TKasaDetayForm;

 implementation

        uses KasaGirisUnit;
```

```
{$R *.DFM}


procedure TKasaDetayForm.SpeedButton5Click(Sender: TObject);
begin
          Close;
end;

procedure TKasaDetayForm.SpeedButton4Click(Sender: TObject);
var
         y,x,i,toplam:integer;
begin
          KasaDetayTable.FieldByName('KasaNo').AsString:=KasaNo.Text;
          KasaDetayTable.Post;
          Close;
          x:=KasaGirisForm.KasaDetayTable.RecordCount;
          toplam:=0;
          KasaGirisForm.KasaDetayTable.First;
          for i:=1 to x do
           begin
            y:=KasaGirisForm.KasaDetayTable.FieldByName('MasrafMiktari').AsInteger;
            toplam:=toplam+y;
            KasaGirisForm.KasaDetayTable.Next;
            KasaGirisForm.total.Text:=IntToStr(toplam);
 end;
end;

end.
```

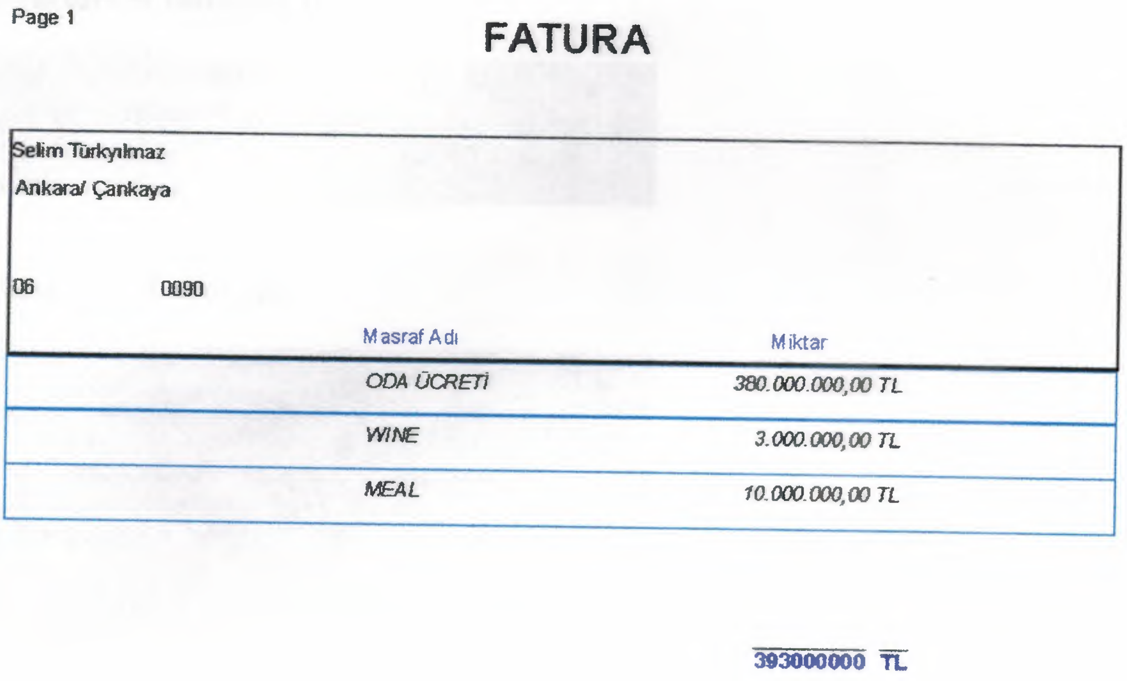
Appendix S:

CODE OF ACCOUNT LIST IN PROGRAM

```
unit KasaListeUnit;

interface

uses
        Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
        Grids, DBGrids, StdCtrls, Buttons, ExtCtrls, Db, DBTables, Refresh;

type
        TKasaListeForm = class(TForm)
          Panel1: TPanel;
          DegistirButon: TSpeedButton;
          YeniButon: TSpeedButton;
          SilButon: TSpeedButton;
          CikisButon: TSpeedButton;
          Label1: TLabel;
          Label2: TLabel;
          SecButon: TSpeedButton;
          AdiEdit: TEdit;
          KodEdit: TEdit;
          DBGrid1: TDBGrid;
          KasaTable: TSelfRefreshTable;
          KasaSource: TDataSource;
          KasaTableKasaNo: TAutoIncField;
          KasaTableDate: TDateField;
          KasaTableCustomerCode: TStringField;
          KasaTableNameSurname: TStringField;
          KasaTableRoomNo: TIntegerField;
          KasaTableTotal: TCurrencyField;
          procedure YeniButonClick(Sender: TObject);
          procedure DegistirButonClick(Sender: TObject);
          procedure CikisButonClick(Sender: TObject);
          procedure SilButonClick(Sender: TObject);
          procedure KodEditChange(Sender: TObject);
          procedure AdiEditChange(Sender: TObject);
 private
   { Private declarations }
 public
   { Public declarations }
 end;
var
       KasaListeForm: TKasaListeForm;

implementation
```

```
        uses KasaGirisUnit;

{$R *.DFM}

procedure TKasaListeForm.YeniButonClick(Sender: TObject);
var
        x:integer;
begin
        KasaGirisForm.KasaTable.insert;
        KasaGirisForm.ShowModal;
end;

procedure TKasaListeForm.DegistirButonClick(Sender: TObject);
begin
        KasaGirisForm.KasaTable.GotoCurrent(KasaListeForm.KasaTable);
        KasaGirisForm.KasaTable.Edit;
        KasaGirisForm.ShowModal;
end;

procedure TKasaListeForm.CikisButonClick(Sender: TObject);
begin
        Close;
end;

procedure TKasaListeForm.SilButonClick(Sender: TObject);
begin
        if messageDlg('Are You Sure You Want To Delete This Record?',
        mtInformation, [mbYes,MbNo],0)=idYes then
        KasaTable.Delete;
end;

procedure TKasaListeForm.KodEditChange(Sender: TObject);
begin
        if (kodEdit.Text='') then
          begin exit; end
        else
        KasaTable.Locate('CustomerCode',KodEdit.text,[loCaseInsensitive,loPartialKey]);
end;

procedure TKasaListeForm.AdiEditChange(Sender: TObject);
begin
        if (AdiEdit.Text='') then
          begin exit; end
        else
        KasaTable.Locate('NameSurname',AdiEdit.text,[loCaseInsensitive, loPartialKey]);
end;
end.
```

Appendix T:



Page 1

FATURA

Selim Türkyılmaz
Ankara/ Çankaya

06 0090

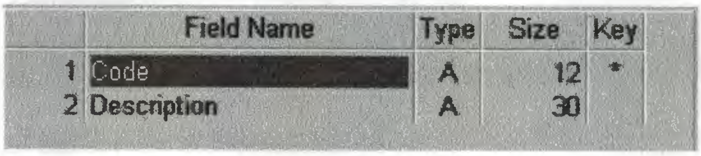
Masraf Adı	Miktar
ODA ÜCRETİ	380.000.000,00 TL
WINE	3.000.000,00 TL
MEAL	10.000.000,00 TL

393000000 TL

Appendix U:

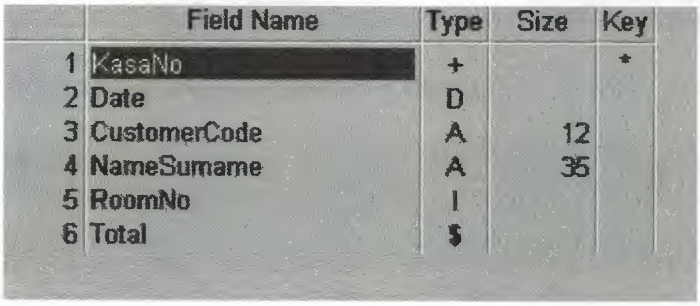
DATA STRUCTURES

1-Structure of Incoming With;



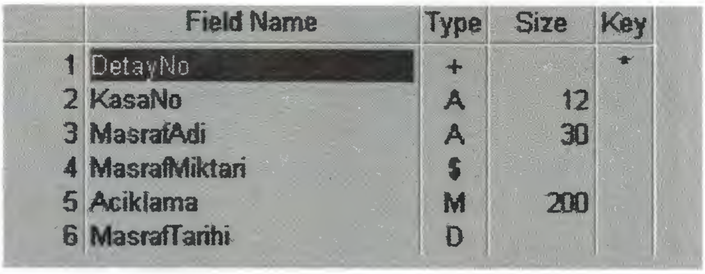
	Field Name	Type	Size	Key
1	Code	A	12	*
2	Description	A	30	

2- Structure of Account;



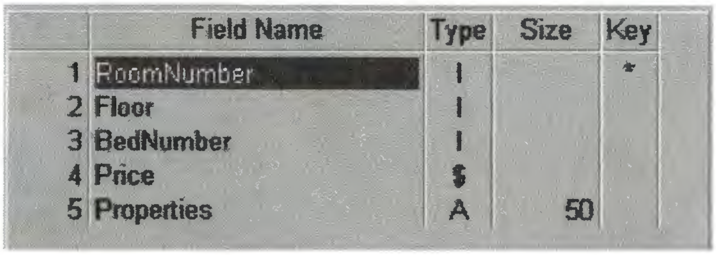
	Field Name	Type	Size	Key
1	KasaNo	+		*
2	Date	D		
3	CustomerCode	A	12	
4	NameSurname	A	35	
5	RoomNo	I		
6	Total	$		

3- Structure of Detail of Expenditure;



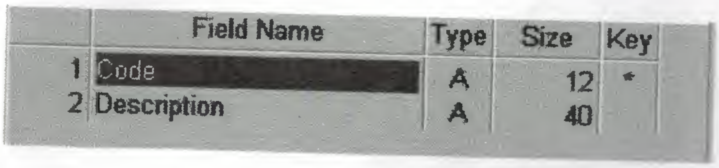
	Field Name	Type	Size	Key
1	DetayNo	+		*
2	KasaNo	A	12	
3	MasrafAdi	A	30	
4	MasrafMiktari	$		
5	Aciklama	M	200	
6	MasrafTarihi	D		

4- Structure of Definition of Room;



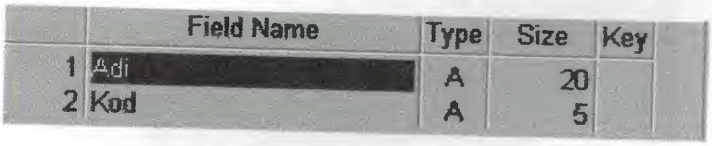
	Field Name	Type	Size	Key
1	RoomNumber	I		*
2	Floor	I		
3	BedNumber	I		
4	Price	$		
5	Properties	A	50	

5- Structure of Definition of Pension;



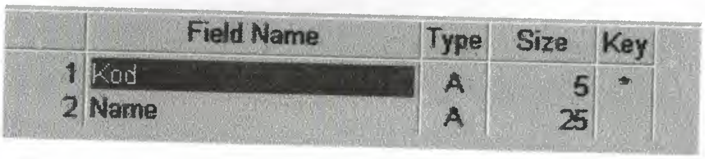
	Field Name	Type	Size	Key
1	Code	A	12	*
2	Description	A	40	

6- Structure of Definition of City;



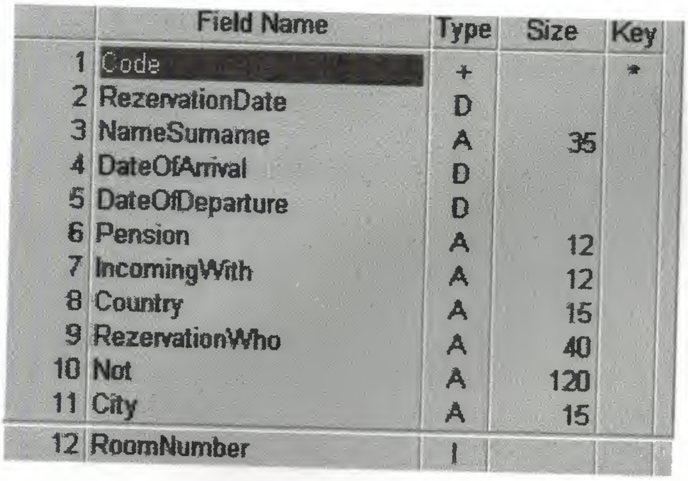
	Field Name	Type	Size	Key
1	Adi	A	20	
2	Kod	A	5	

7-Structure of Definition of Country;



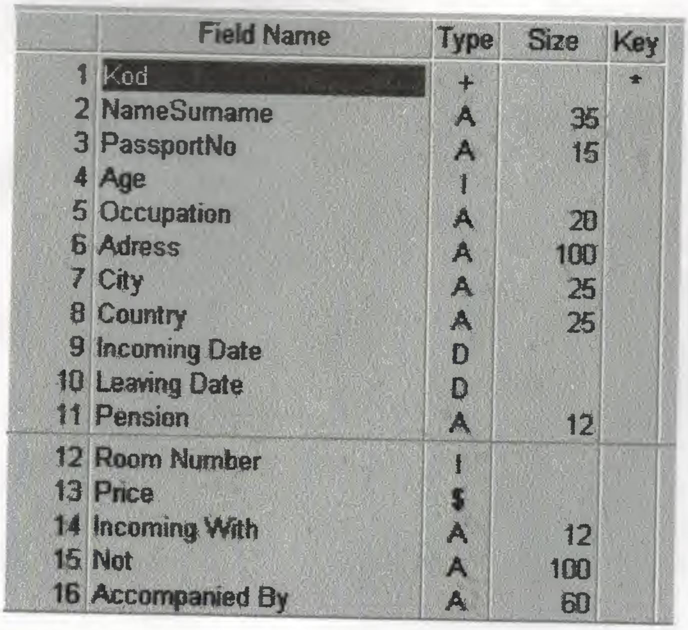
	Field Name	Type	Size	Key
1	Kod	A	5	*
2	Name	A	25	

8- Structure of Customer Reservation;



	Field Name	Type	Size	Key
1	Code	+		*
2	RezervationDate	D		
3	NameSurname	A	35	
4	DateOfArrival	D		
5	DateOfDeparture	D		
6	Pension	A	12	
7	IncomingWith	A	12	
8	Country	A	15	
9	RezervationWho	A	40	
10	Not	A	120	
11	City	A	15	
12	RoomNumber	I		

9- Structure of Customer Registration;



	Field Name	Type	Size	Key
1	Kod	+		*
2	NameSurname	A	35	
3	PassportNo	A	15	
4	Age	I		
5	Occupation	A	20	
6	Adress	A	100	
7	City	A	25	
8	Country	A	25	
9	Incoming Date	D		
10	Leaving Date	D		
11	Pension	A	12	
12	Room Number	I		
13	Price	$		
14	Incoming With	A	12	
15	Not	A	100	
16	Accompanied By	A	60	

DEFINITION OF USED DATA TYPES

I = Integer,
A = Alpha,
$ = Money
D = Date
+ = Autoincrement,
M = Memo.