1988

Near East University

Faculty of Economics and Administrative Sciences

Computer Information Systems

CIS-(400)

---TRAVEL AGENCY---

Submitted by: Mehmet Sait YENİ
Student number:20021501
Submitted to: Department of CIS

LEFKOŞA/2007

CONTENTS

1.ACKNOWLEDGEMENTS

'' First, I would like to thank my supervisor Mr.Mustafa MENEKAY for his ivaluable advice and belief in my work and my self over the course of this Graduation Project..

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

Third, I thank my family for their constant encouragement and support during the preparition of this Project.

Finally, I would also like thank all my friends for their advice and support.''

2.ABSTRACT

Travel Agency Automation System is designed to use in a single computer by me. I coded this program to use in the second class local travel agencies. The mean of these travel agencies work in only one city and has at most two or three branches.

These organizations communicate each others by the phone. Because of, expenses of the internet connection and not to became widespread between the travel agencies of using internet.

This automation program has many useful abilities. These are; seat sale, seat reservation, sale of reserved seat, cancellation of reserved seat, cancellation of sold seat, record of the; branches, employees, vehicles and voyages. Also this program can produce some calculated reports; revenue of sale, voyage reports, bill payments, salary payments and the report of total balance.

The Project consist of introduction, three chapters and conclusion.

Chapter one describes the history of delphi, introducing to delphi, some examples about how to code basic processes in delphi.

Chapter two describes introducing to Microsoft Access, understanding of Microsoft Access and objects of Microsoft Access.

Chapter three includes the screenshots and user manual of Travel Agency Automation System. These screenshots are related to codes of the program which is explained in the appendix.

3.INTRODUCTION

CHAPTER ONE : DELPHI

1.1 A BRIEF HISTORY OF BORLAND'S DELPHI

Pascal

Delphi uses the language Pascal, a third generation structured language. It is what is called a highly typed language. This promotes a clean, consistent programming style, and, importantly, results in more reliable applications. Pascal has a considerable heritage:

Beginnings

Pascal appeared relatively late in the history of programming languages. It probably benefited from this, learning from Fortran, Cobol and IBM's PL/1 that appeared in the early 1960's. Niklaus Wirth is claimed to have started developing Pascal in 1968, with a first implementation appearing on a CDC 6000 series computer in 1970.

Curiously enough, the C language did not appear until 1972. C sought to serve quite different needs to Pascal. C was designed as a high level language that still provided the low level access that assembly languages gave. Pascal was designed for the development of structured, maintainable applications.

The 1970's

In 1975, Wirth teamed up with Jensen to produce the definitive Pascal reference book "Pascal User Manual and Report". Wirth moved on from Pascal in 1977 to work on Modula - the successor to Pascal.

The 1980's

In 1982 ISO Pascal appears. The big event is in November 1983, when Turbo Pascal is released in a blaze of publicity. Turbo Pascal reaches release 4 by 1987. Turbo

Pascal excelled on speed of compilation and execution, leaving the competition in its wake.

From Turbo Pascal to Delphi

Delphi, Borland's powerful Windows© and Linux© programming development tool first appeared in 1995. It derived from the Turbo Pascal© product line.

As the opposition took heed of Turbo Pascal, and caught up, Borland took a gamble on an Object Oriented version, mostly based on the Pascal object orientation extensions. The risk paid off, with a lot of the success due to the thought underlying the design of the IDE (Integrated Development Environment), and the retention of fast compilation and execution.

This first version of Delphi was somewhat limited when compared to today's heavyweights, but succeeded on the strength of what it did do. And speed was certainly a key factor. Delphi went through rapid changes through the 1990's.

Delphi for Microsoft .Net

From that first version, Delphi went through 7 further iterations before Borland decided to embrace the competition in the form of the Microsoft© .Net architecture with the stepping stone Delphi 8 and then fully with Delphi 2005 and 2006. Delphi however still remains, in the opinion of the author, the best development tool for stand alone Windows and Linux applications. Pascal is a cleaner and much more disciplined language than Basic, and adapted much better to Object Orientation than Basic.

Into the public domain

At the time of writing, Borland is looking to sell Delphi. An independent organisation is hoping to buy Delphi in order to make it an Open Source product, allowing all to contribute to its future.

1.2 WHAT IS BORLAND DELPHI ?

Borland Delphi is a sophisticated Windows programming environment, suitable for beginners and professional programmers alike. Using Delphi you can easily create self-contained, user friendly, highly efficient Windows applications in a very short time - with a minimum of manual coding.

Delphi provides all the tools you need to develop, test and deploy Windows applications, including a large number of so-called reusable components.

Borland Delphi, in it's latest version, provides a cross platform solution when used with Borland Kylix - Borland's RAD tool for the Linux platform.

Delphi's roots lie in Borland's Turbo Pascal, introduced in the mid-1980s. Object Pascal, the object-oriented extensions to Pascal, is the underlying language of Delphi. The Visual Component Library, or VCL, is a hierarchy of Object Pascal objects that allow you to design applications. A better way of describing Delphi is an Object Pascal-based visual development environment.

Delphi editions

The latest version of Delphi (in the moment of writing this chapter), version 6, comes in three different editions:

- Delphi Personal - makes learning to develop non-commercial Windows applications fast and fun. Delphi 6 Personal makes learning Windows development easy with drag-and-drop visual programming.
- Delphi Professional - adds the tools necessary to create applications with the latest Windows® ME/2000 look-and-feel. Dramatically enhance functionality with minimal code using the power and flexibility of SOAP and XML to easily integrate Web Services into client-side applications.
- Delphi Enterprise - includes additional tools, extensive options for Internet. Delphi 6 makes next-generation e-business development with Web Services a snap.

1.3 SOME MANIPULATIONS IN DELPHI

Changing the form size with a program

Open a new project in Delphi by selecting the menu sequence File | New Application (Delphi versions 4 to 7) or File | New | VCL Forms Application (Delphi 8). Place two Buttons on the Form. Do the static design by clicking once on each component in turn and then setting the appropriate property values in the Object Inspector as shown below.

```
procedure TForm1.Button2Click(Sender: TObject);
begin
Form1.Height := 200;
end;
```

Changing button dimensions

After opening a new project, set the following Form properties:

Now, using the Component | Tool palette, add two buttons to the form and set the following properties in the Object Inspector:

This completes the static design. To start with the run time design, double-click on Button1 (Small) and create the following event handler:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
Form1.Height := 200;
Form1.Width := 130;
Button1.Height := 30;
Button1.Width := 60;
Button2.Height := 30;
Button2.Width := 60;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
Form1.Height := 300;
```

```
Form1.Width := 400;
Button1.Height := 60;
Button1.Width := 120;
Button2.Height := 60;
Button2.Width := 120;
end;
```

Change a button's Caption in a program

This is not essential, and you can keep the original Form size and position the Buttons on it anywhere that is convenient.Double-click on Button1 in the Form Designer, and create the following event handler:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
Button2.Caption := 'Close';
end;
```

Enabling and disabling buttons

Since this program is so similar to the one in the previous problem, we can use that program as a starting point. Read in the program from the previous problem by using the menu sequence File | Open Project, changing to your working directory and then selecting C01p03p.dpr (Delphi 4 to 8) or C01p03p.bdsproj (Delphi 8). To create a copy you can use to modify for this problem without writing over the previous version, use File | Save As… to save the unit file (C01p04u.pas), and File | Save Project As… for the project file (C01p04p.dpr for Delphi 4 to 7 or C01p04p.bdsproj for Delphi 8). To change the program, we need to make sure that Button Second is disabled when the program starts up. We usually control the start-up behaviour of a simple program by setting the various properties at design time in the Object Inspector. In this case, the only property that differs from the previous problem is Button2's Enabled property.

To set this property, click once on Button2, and then go to the Object Inspector and find the Enabled property. You will see that its value is shown as True. Double-click on True to change it to False. This makes sure that when the program runs for the first time, Button2 is *not* enabled. We also have to change the two Buttons' Enabled properties while the program is running, and we do this by changing the Button1Click event handler as follows:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
Button1.Enabled := False;
Button2.Enabled := True;
Button2.Caption := 'Close';
end;
```

Changing the WindowState property

Open a new application and set the components and properties as show below:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
WindowState := wsMinimized;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
WindowState := wsNormal;
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
WindowState := wsMaximized;
end;
```

Reading a program

We repeat the program code here with line numbers to simplify the discussion:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
Form1.Width := 203;
Form1.Height := 87;
```

```
Button2.Top := 16;
Button2.Left := 104;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
Form1.Width := 120;
Form1.Height := 144;
Button2.Top := 73;
Button2.Left := 16;
end;
```

Multiple text displays

Start a new application. Create the interface shown and then set the static properties as follows:

```
procedure TfrmTextDemo.btnNameClick(Sender: TObject);
begin
lblName.Caption := edtText.Text;
edtText.Clear;
edtText.SetFocus;
end;
procedure TfrmTextDemo.btnDepartmentClick(Sender: TObject);
begin
lblDepartment.Caption := edtText.Text;
edtText.Clear;
edtText.SetFocus;
end;
procedure TfrmTextDemo.btnBirthDateClick(Sender: TObject);
begin
lblBirthDate.Caption := edtText.Text;
edtText.Clear;
edtText.SetFocus;
end;
procedure TfrmTextDemo.btnClearAllClick(Sender: TObject);
begin
lblBirthDate.Caption := '';
lblDepartment.Caption := '';
lblName.Caption := '';
end;
```

Displaying captions

The RadioButtons' OnClick event handlers are as follws:

```
procedure TfrmCaptions.radButton1Click(Sender: TObject);
begin
lblOutput.Caption := radButton1.Caption;
end;
procedure TfrmCaptions.radButton2Click(Sender: TObject);
begin
lblOutput.Caption := radButton2.Caption;
end;
procedure TfrmCaptions.radButton3Click(Sender: TObject);
begin
lblOutput.Caption := radButton3.Caption;
end;
procedure TfrmCaptions.radButton4Click(Sender: TObject);
begin
lblOutput.Caption := radButton4.Caption;
end;
```

Accessing 'Configuration Settings' files (.INI) with Delphi

How a simple text file can beat Registry in storing a few pieces of application specific configuration data.

The .INI files have a text-based file format for representing application configuration data in a format which is easily editable by humans and readable by a simple automatic parser.

Most people who use Windows have at least a passing acquaintance with .INI files and with the WIN.INI and SYSTEM.INI files in particular (check your Windows directory). Windows stored (and still does) such important information as configuration information about Windows in plain text files that are easily deleted, modified or viewed.

Even though, in 32-bit Windows Microsoft recommends using Registry to store application specific configuration data, in many cases you'll find that using INI files is much faster and safer.

One simple use of INI files, as a status saving mechanism, would be to save the size and location of a form if you want a form to reappear at it's previous position. In general everything you store in Registry you could store in an INI file.

.INI file format

Initialization or Configuration Settings file (.INI) is a text file with 64Kb limit divided into sections, each containing zero or more keys. Each key contains zero or more values. Example:

```
[SectionName]
keyname=value
;comment
keyname=value
```

Section names are enclosed in square brackets, and must begin at the beginning of a line. Section and key names are case-insensitive, and cannot contain spacing characters. The key name is followed by an equal sign ("="), optionally surrounded by spacing characters, which are ignored.
If the same section appears more than once in the same file, or if the same key appears more than once in the same section, then the last occurrence prevails.
A key can contain string, integer or boolean value.

Delphi uses INI file format in many cases. For example, the .DSK files (desktop settings) have a file format as described above.

TIniFile

Delphi provides the TIniFile class with methods designed to store and retrieve application-specific information and settings from INI files. The TIniFile class is declared in the inifiles.pas unit. Prior to working with the TIniFile control you need to create an instance of the class:

```
uses inifiles;

var
 IniFile : TIniFile;
begin
 IniFile := TIniFile.Create('myapp.ini');
```

This code creates an IniFile object and assigns 'myapp.ini' to the only property of the class - the FileName property - used to specify the name of the INI file you are to use.

The code as written above looks (stores) for the myapp.ini file in the \Windows directory. A better way to store application data is in the application's folder - just specify the full path name of the file for the Create method:

```
// place the INI in the application folder,
// let it have the application name
// and 'ini' for extension:
 IniFile := TIniFile.Create(ChangeFileExt(
         Application.ExeName,'.ini'));
```

Reading

The TIniFile class has several "read" methods. The ReadString reads a string value from a key, ReadInteger, ReadFloat and similiar are used to read a number from a key. All "read" methods have a default value that can be used if the entry does not exist. For example the ReadString is declared as: function ReadString(const Section, Ident, Default: String): String; override;

Writing

The TIniFile has a corresponding "write" method for each "read" method. In other words they are WriteString, WriteBool, WriteInteger, etc.

For example, if we want a program to remember the name of the last person who used it, when it was, and what were the main form coordinates, we might establish a section called "Users" and a keyword called "Last" and "Date" to track the information; and another section called "Placement" with keys "Top", "Left", "Width", "Height".

Note that the key named "Last" holds a string value, "Date" holds a TDateTime value, all keys in the "Placement" section hold an integer value.

The OnCreate event of the main form is the perfect place to store the code needed to access the values in the application's initialization file:

```
procedure TForm1.FormCreate(Sender: TObject);
var
 IniFile  : TIniFile;
 LastUser : string;
 LastDate : TDateTime;
begin
 IniFile := TIniFile.Create(
       ChangeFileExt(Application.ExeName,'.ini'));
 //if no last user return an empty string
 LastUser := IniFile.ReadString('User','Last','');
 //if no last date return todays date
```

```
  LastDate := IniFile.ReadDate('User', 'Date', Date);

  //show the message
  ShowMessage('This program was previously used by '
             + LastUser + ' on '
             + DateToStr(LastDate));

  Form1.Top := IniFile.ReadInteger
              ('Placement','Top', Form1.Top);
  Form1.Left := IniFile.ReadInteger
               ('Placement','Left', Form1.Left);
  Form1.Width := IniFile.ReadInteger
                ('Placement','Width', Form1.Width);
  Form1.Height := IniFile.ReadInteger
                 ('Placement','Height', Form1.Height);
  IniFile.Free;
end;
```

The OnClose event is ideal for the Save part of the project. Before the application terminates (OnClose of the main form) we save its current position and the info on the user:

```
procedure TForm1.FormClose
        (Sender: TObject; var Action: TCloseAction);
var
  IniFile  : TIniFile;
begin
  IniFile := TIniFile.Create(
       ChangeFileExt(Application.ExeName,'.ini'));
  IniFile.WriteString('User','Last','Zarko Gajic');
  IniFile.WriteDate('User', 'Date', Date);
```

```
 With IniFile, Form1 do
 begin
   WriteInteger('Placement','Top', Top);
   WriteInteger('Placement','Left', Left);
   WriteInteger('Placement','Width', Width);
   WriteInteger('Placement','Height', Height);
 end;
 IniFile.Free;
end;
```

Working with Sections

There are several methods designed to operate on entire sections of the INI file. For example the EraseSection erases an entire section of an INI file. The ReadSection and ReadSections fill a TStringList object with the names of all sections (and key names) in the INI file.

Note: another class in the registry unit, the TRegIniFile class was introduced for easy access to the system's Registry without having to know the structure of the Registry. TRegIniFile uses similar functional access to the Registry as TIniFile does to an INI file.

Limitations and downsides

The TIniFile class uses the Windows API which imposes a limit of 64KB on INI files. If you need to store more than 64KB of data, you should use the TMemIniFile instead, it does not have a limit of 64KB.

Another problem might rise if you have a section with more than 8K value. The ReadSection method uses the GetPrivateProfileString API function with the 8K buffer. One way to solve the problem is to write your own version of the ReadSection method.

The fact that anyone could access and change INI files means that the integrity of the information contained in the files could not be guaranteed. Yet, the Registry too can be edited with the tools provided by Microsoft.

CHAPTER TWO : MICROSOFT ACCESS

Microsoft Access (full name Microsoft Office Access) is a relational database management system from Microsoft, packaged with Microsoft Office Professional which combines the relational Microsoft Jet Database Engine with a graphical user interface. It can use data stored in Access/Jet, SQL Server, Oracle, or any ODBC-compliant data container. Skilled software developers and data architects use it to develop powerful, complex application software. Relatively unskilled programmers and non-programmer "power users" can use it to build simple applications without having to deal with features they don't understand. It supports substantial object-oriented (OO) techniques but falls short of being a fully OO development tool.

Microsoft Access was also the name of a communications program from Microsoft, meant to compete with ProComm and other programs. It proved a failure and was dropped. Years later they reused the name for their database software.

2.1 Uses

Access is widely used by small businesses, within departments of large corporations, and hobby programmers to create ad hoc customized systems for handling the creation and manipulation of data. Its ease of use and powerful design tools give the non-professional programmer a lot of power for little effort. However, this ease of use can be misleading. This sort of developer is often an office worker with little or no training in application or data design. Because Access makes it possible even for such developers to create usable systems, many are misled into thinking that the tool itself is limited to such applications.

Some professional application developers use Access for rapid application development, especially for the creation of prototypes and standalone applications that serve as tools for on-the-road salesmen. Access does not scale well if data access is via a network, so applications that are used by more than a handful of people tend to rely on a Client-Server based solution such as Oracle, DB2, Microsoft SQL Server, PostgreSQL, MySQL, MaxDB, or Filemaker. However, an Access "front end" (the forms, reports, queries and VB code) can be used against a host of database backends, including Access itself, SQL Server, Oracle, and any other ODBC-compliant product. This approach allows the developer to move a matured application's data to a more powerful server without sacrificing the development already in place.

Many developers who use Microsoft Access use the Leszynski naming convention, though this is not universal; it is a programming convention, not a DBMS-enforced rule.

2.2 Features

One of the benefits of Access from a programmer's perspective is its relative compatibility with SQL – queries may be viewed and edited as SQL statements, and SQL statements can be used directly in Macros and VBA Modules to manipulate Access tables. Users may mix and use both VBA and "Macros" for programming forms and logic and offers object-oriented possibilities.

The report writer in Access, while capable and up to the task of sophisticated report creation, is not as full-featured and powerful as another popular database report writer – Crystal Reports. MSDE (Microsoft SQL Server Desktop Engine) 2000, a mini-version of MS SQL Server 2000, is included with the developer edition of Office XP and may be used with Access as an alternative to the Jet Database Engine. (*Early versions of MSDE and Microsoft Exchange Server actually use the Jet engine to handle huge volumes of data and placed a "fake" application layer for those applications on top of it. Lack of knowledge about this fact has contributed to an undeserved disrespect for Access/Jet family of software products, particularly as regards "large" projects.)

Access' cut and paste functionality can make it a useful tool for connecting between other databases (for example, Oracle and Microsoft SQL Server during data or database conversions). Access comes with various import and export features that allow integration with Windows and other platform applications, several of which can be executed on demand from within applications or manually by the user. For example the very compact SNP format for sharing perfectly formatted reports with people who don't have the full Access software. It can also easily be upgraded to Microsoft SQL Server.

Unlike complete RDBMS's, it lacks database triggers and stored procedures. It does allow forms to contain code that is triggered as changes are made to the underlying table, and it is common to use pass-through queries and other techniques in Access to run stored procedures in RDBMSs that support these.

2.3 Development

The programming language available in Access is, as in other products of the Microsoft Office suite, Microsoft Visual Basic for Applications. Two database access libraries of COM components are provided: the legacy Data Access Objects (DAO), only available with Access, and the new ActiveX Data Objects (ADO).

Microsoft Access is easily applied to small projects but scales inefficiently to large projects if applications are designed poorly.

All database queries, forms, and reports are stored in the database, and in keeping with the ideals of the relational model, there is no possibility of making a physically structured hierarchy with them.

One design technique is to divide an Access application between data and programs. One database should contain only tables and relationships, while another would have all programs, forms, reports and queries, and links to the first database tables. Unfortunately, Access allows no relative paths when linking, so the development environment should have the same path as the production environment (Although you can write your own "dynamic-linker" routine in VBA that can search out a certain back-end file by searching through the directory tree, if it can't find it in the current path).

2.4 Relational Database

An organized database is composed of inter-related parts. Since you define these parts, you also organize them in a manner that helps some parts of your database to supply specific information to others. In one part, you would cover one category of data, such as people's personal information (name, date of birth, salary, hobbies, etc), in another you would cover what they buy in a store or other related transactions.

Microsoft Access is a relational database application used on desktop computers to manage information on different levels for different purposes.

Microsoft Access can be used for personal information management, in a small business to organize and manage all data, or in an enterprise to communicate with servers..

2.5 Database Management System (DBMS)

A database is an organized collection of data. Organization means method, it assumes discipline, it also anticipates efficient manner in using that information. Unless you are creating small applications for your personal use, you will usually need to share your data either with other people (users, database developers, etc) or other machines. To make your job easier, Microsoft Access provides in one package the database information and the tools you need to use your database. To be organized, you will divide your database in different related parts. The method of management you will use makes Microsoft Access a Database Management System (DBMS). When you create a database in Microsoft Access, you create a file that will include different parts of your database. These are referred to as tables queries, forms, reports, etc.

2.6 Microsoft Access as a software product

Microsoft Access is a classic computer application and it gets launched like the usual products you have probably been using. As such, to start this program, you could click Start -> (All) Programs -> Microsoft Access. As a regular member of the Microsoft Office suite of applications, if your installation created a sub-menu on the Start menu, you could click Start -> New Office Application and proceed from the New dialog box.

Although Microsoft Office 97 and Microsoft Office 2000 get installed in the C:\Program Files\Microsoft Office folder, they treat the shortcuts that launch them differently. The applications that are part of Microsoft Office 97 designate their shortcuts with full names and these are installed in the Microsoft Office folder. Microsoft Office 2000 (Premium) uses shortcut names to designate its shortcuts and they are installed in the Microsoft Office\Office folder. This means that you could launch an application from Windows Explorer or My Computer. Therefore, in order to launch Microsoft Access, locate its shortcut in Windows Explorer or My Computer and double-click it.

If you have a Microsoft Access database such as an E-Mail attachment, a file on a floppy disk, on the network, or in any other means, once you see its icon, you can double-click it. Not only will this action launch Microsoft Access, but also it will open the file.

You can also launch Microsoft Access from a shortcut. If you happen to use the software on a regular basis, you can create a shortcut on your desktop or on the Quick Launch area. Many users also take advantage of the Microsoft Office Shortcut Bar. Sometimes, the icon you need will not be there; in that case you should insert it manually.

When Microsoft Access starts, you are presented with a special dialog box that inquires about your intentions. From this dialog, you can do one of four things: create a database using one of the sample files, create a database from scratch, open an existing database, or open a "raw" program

2.7 Microsoft Access Interface

The title bar occupies the most top section of the Microsoft Access interface. It is made of three sections: the system icon, the main title bar area (this is referred to as The Title Bar), and the Windows system buttons.

The Title Bar

The button on the left holds the system menu. If you click it, it would display a menu that allows you to minimize, maximize, move, restore, or close the window. On the right side of the system icon is the name of the application, in this case Microsoft Access.

The main area of the title bar, on the right side of the name Microsoft Access, allows you to perform some operations on the window. If the application is not maximized, you can grab and drag this section of the title bar to move or resize the window. You can also double-click the title bar to maximize the window if it is not maximized or to restore it if it is maximized.

The Main Menu

Under the title bar is the Menu Bar made of words such as File, Edit, View, etc. In this book, "Menu Bar" and "Main Menu" are used interchangeably to designate the group of words under the title bar.

To use the menu, first locate a word that may represent the item you want and click. The items on the menu are organized in groups but the group does not follow strict rules. If you find the item you are looking for, you can click it.

The programmers who create such a menu follow some suggestions but do not abide by an operating rule. Therefore, sometimes the word on top simply represents a group.

In this book, when the arrow -> is used, this means you should click the word or group of words that follow the arrow. For example if you are asked to "Click File -> Save", this means "click File and then click Save".If you do not know where an item is located, you can just click one of the menu items to open the menu in general. After clicking an item such as File, Edit, or View, a window drops and displays another list of words. If you move the mouse, another item is highlighted. This allows you to move to another group or to select an item under a group.

2.8 Microsoft Access Database Objects

A Microsoft database is a file made of various internal objects: these are the tables, queries, forms, reports, etc. All these objects are managed from the Database Window. These objects kept in categories. To access an object, you click the button that corresponds to its category.

2.8.1 Database Objects: Tables

The table is the central point of your development, because all data is stored in tables. The functionality of your database relies on how you design your tables. For better organization, you will have various tables in your database, each for a different purpose.

A table is made of rows and columns. A row is considered a Record, it is a group of details about one specific item of the database. It could contain a customer's name, her phone number, her member ID, her work number, her marital status, etc.

A column is a field representing one particular category of information about the records in the table. For example, it would hold the names of all actors in a particular movie, another column would hold the titles of different movies in the video store.

Each table is recognized by its name. To open a particular table, you can double-click it. You can also right-click a table's name and click Open. If the desired table is already selected on the Database Window, you can click the Open button to open it.

2.8.2 Database Objects: Queries

A table can be large depending on the information it holds. To further organize your data, you should be able to retrieve necessary information for a specific purpose. The solution is to create a query (or queries) in order to limit part of the data in a table for a specific goal, for better management or search. That's the role of a query.

Just like tables, queries are managed from the Database Window in their own category. And you would open a query the way you open a table.

2.8.3 Database Objects: Forms

Tables are used to create the data in your database, but they are usually not good looking, as far as the users are concerned. Forms are windows objects used to view and/or enter data in your database.

A form can combine data that is part of one or more tables or queries. Forms are the window interfaces that you usually will ask your users to access when performing data entry in your database.

2.8.4 Database Objects: Reports

A report is the organized document that you will use to print your data. A report can include different parts or details about your database; it could include data from a table or a query, it could also get its data from various tables, queries, forms, or data that is calculated from other tables or forms. You could also create a completely independent report whose content is not related to any data on an object.

4. DESIGN OF THE SYSTEM

THE PROCESS

1.Customer process: This process shows customer information and it gives information about customer request (destination and time of travel, etc.)

2.Travel process: This process shows travel information about the time and the day of customers departure and arrival.

3.Reservation process: This process shows Passengers reservation or sales process and it gives information whether there is seat or not. If there is not seat we give negative information of reservation or sales however if there is seat we give positive information of reservation or sales.

3.1 Reenter Rezervation process: This process is used for reservation or sales so as to reach reservation or sales again.

3.2 Check Seat process: This process is used to check seat process.

3.3 Rezervation / Sales Ticket process: This process gives information either there is reservation or sales ticket process.

4. Accounting process: This process gives information about the money which is earnt from ticket sales and is given to the employees and it shows expenditures and invoices.

4.1 General Ledger process: This process gives information about the registration of accounting payable and accounting receivable.

4.2 A/R process: This process shows the money which is received.

4.3 A/P process: This process shows the money which is paid.

4.4 Income process: This process shows income process.

5.Employee process: This process shows information about employees and it gives information about employees payment.

5.1 Department process: It shows information about employees department.

6. Payroll process: Thit process gives information about employees payment and invoices.

7.Reports process: This process gives information about report and it provides information to the manager.

7.1 Income / Expense report process: This process shows income and expense report.

7.2 Expense report process: This process shows expense report.

7.3 Payroll report process: This process shows payroll report.

7.4 Sales income report process: This process shows sales income report.

7.5 travel report process: This process shows travel report.

THE EXTERNAL ENTITIES

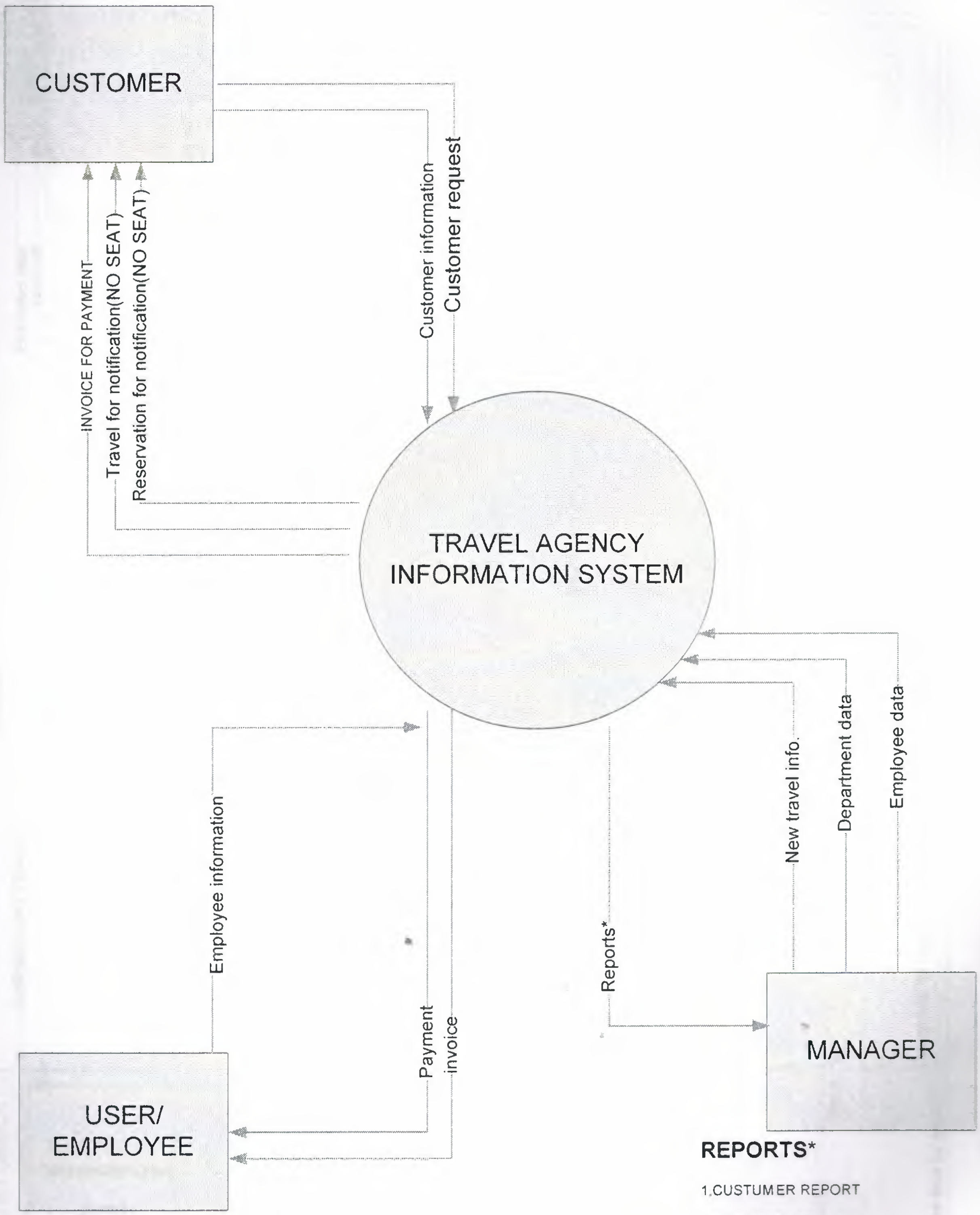
1. **Customer**
2. **User/employee**
3. **Manager**

THE FILES

1. **Sales file:** This file gives information about sales file.
2. **Reservation file:** This file gives information about reservation file.
3. **Reject file:** This file gives information about reject file.
4. **Travel file:** This file gives information about travel file.

5. **Account receivable file:** This file gives information about accounting receivable file.
6. **Account payable file:** This file gives information about accounting payable file.
7. **Payroll file:** This file gives information about payroll file.
8. **Employee file:** This file gives information about employee file.
9. **Customer file:** This file gives information about customer file.

CONTEXT DIAGRAM FOR TRAVEL AGENCY



REPORTS*

1.CUSTUMER REPORT

2.SALE REPORT

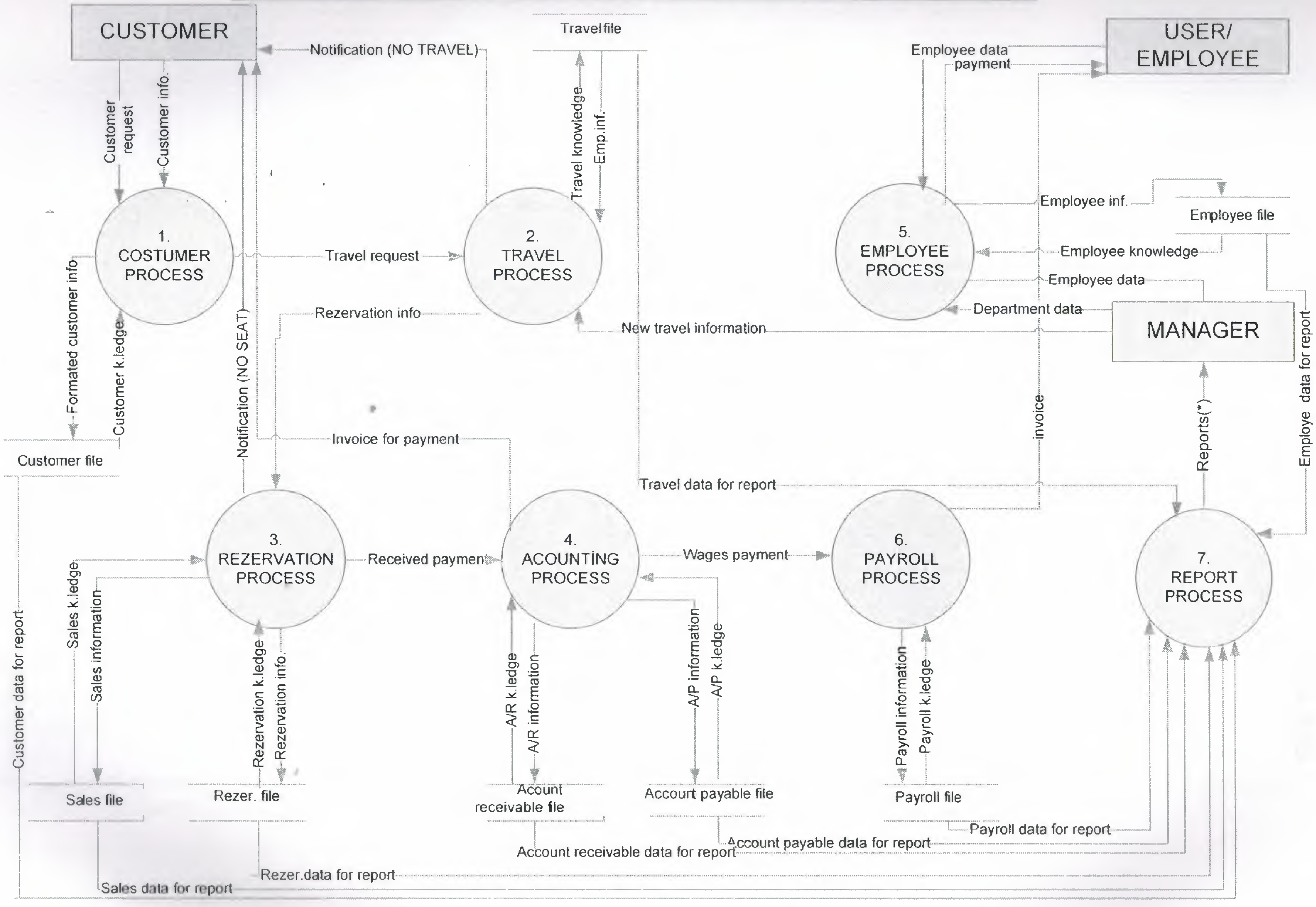
3.OCOUNTİNG REPORT

4.CANCEL REPORT

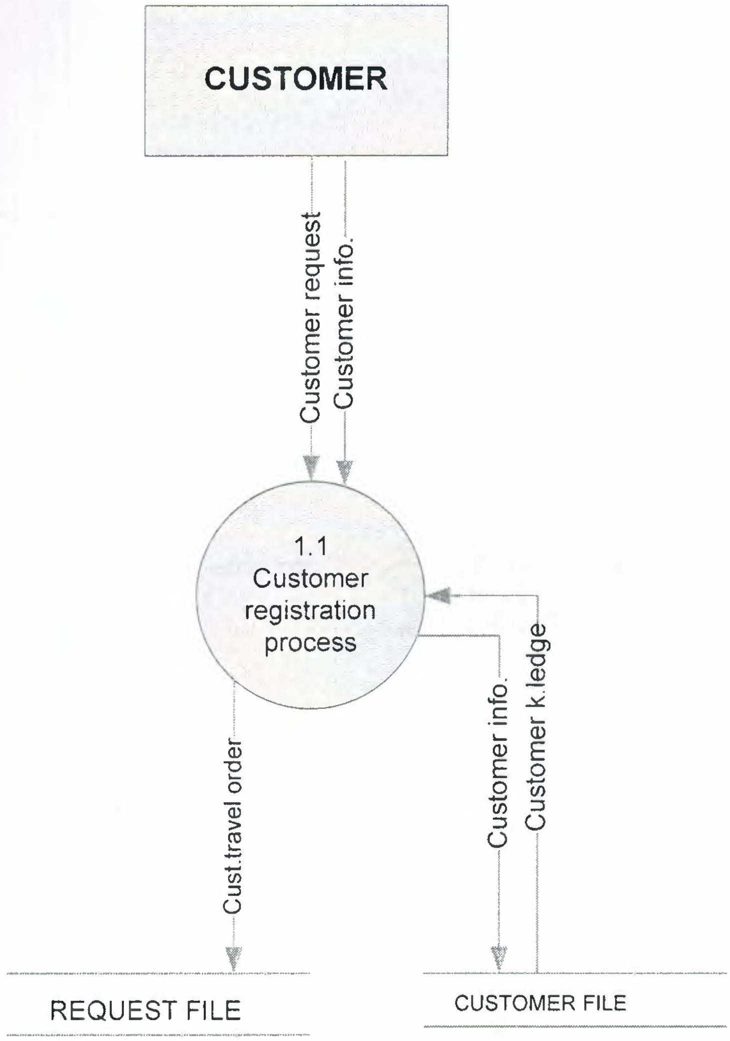
5.EMPLOYEE REPORT

6.REQUEST REPORT

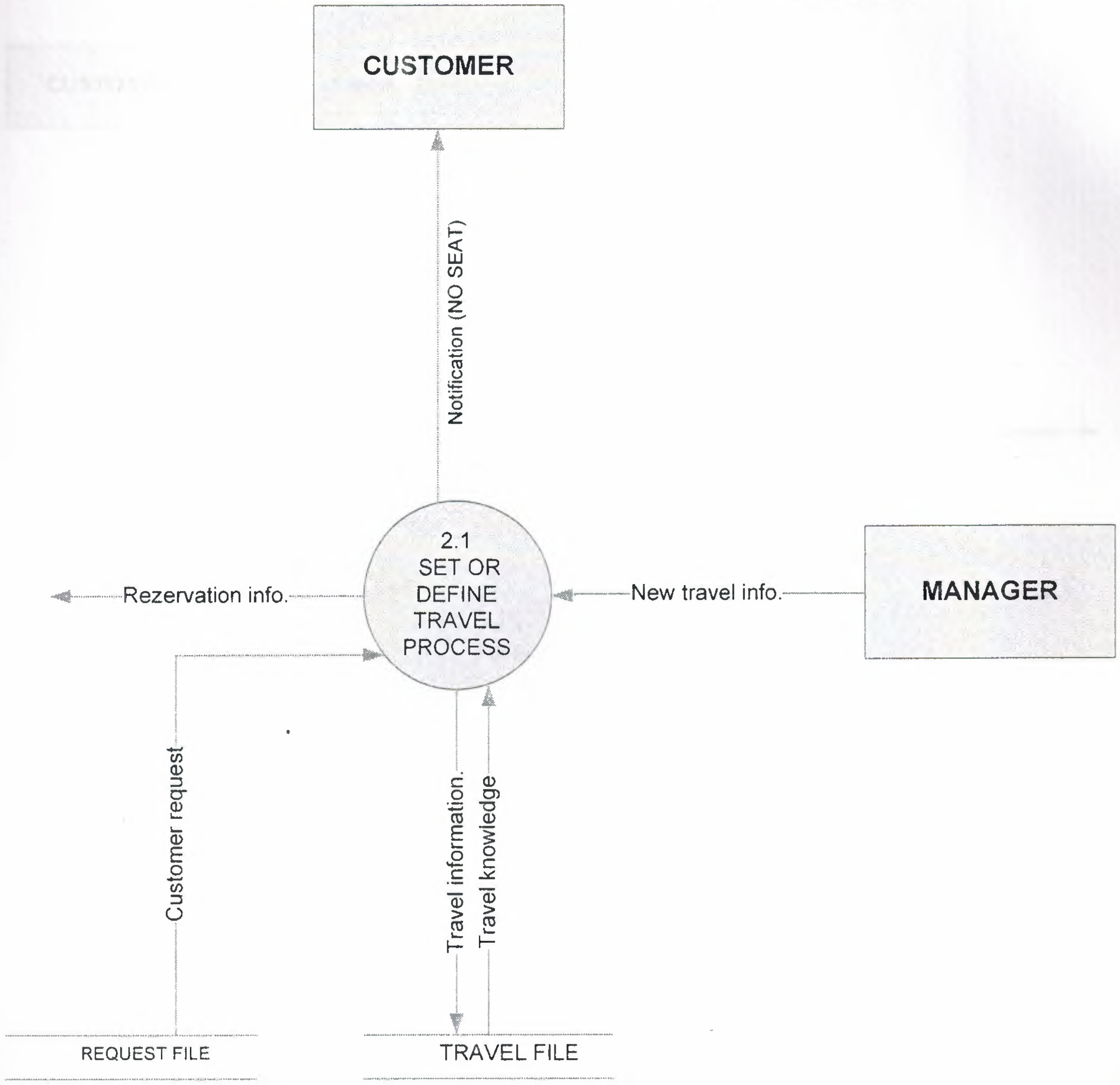
TOP LEVEL FOR TRAVEL AGENCY INFORMATION SYSTEM



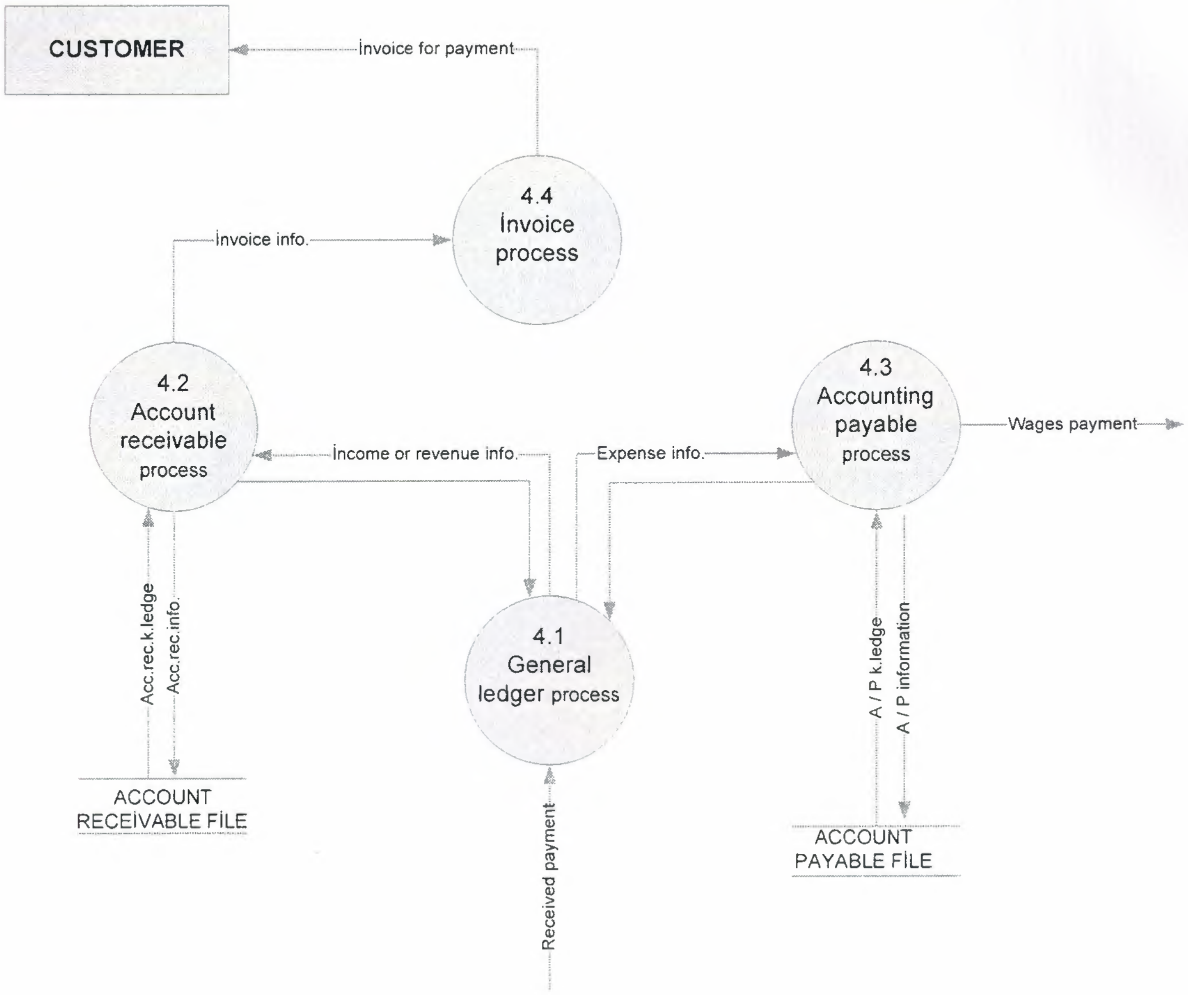
1.DETAIL DFD FOR CUSTOMER PROCESS



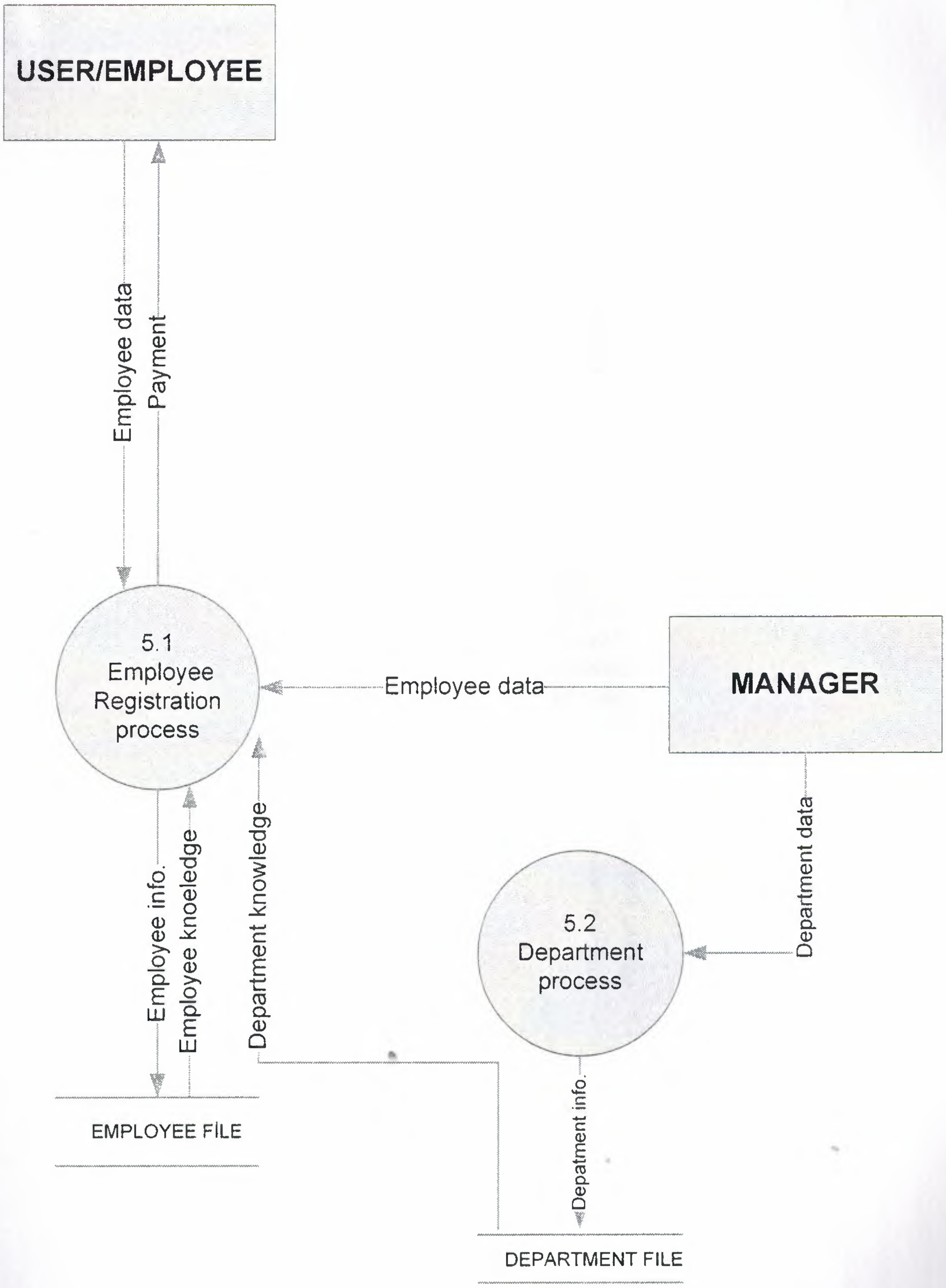
2. DETAIL DFD FOR TRAVEL PROCESS



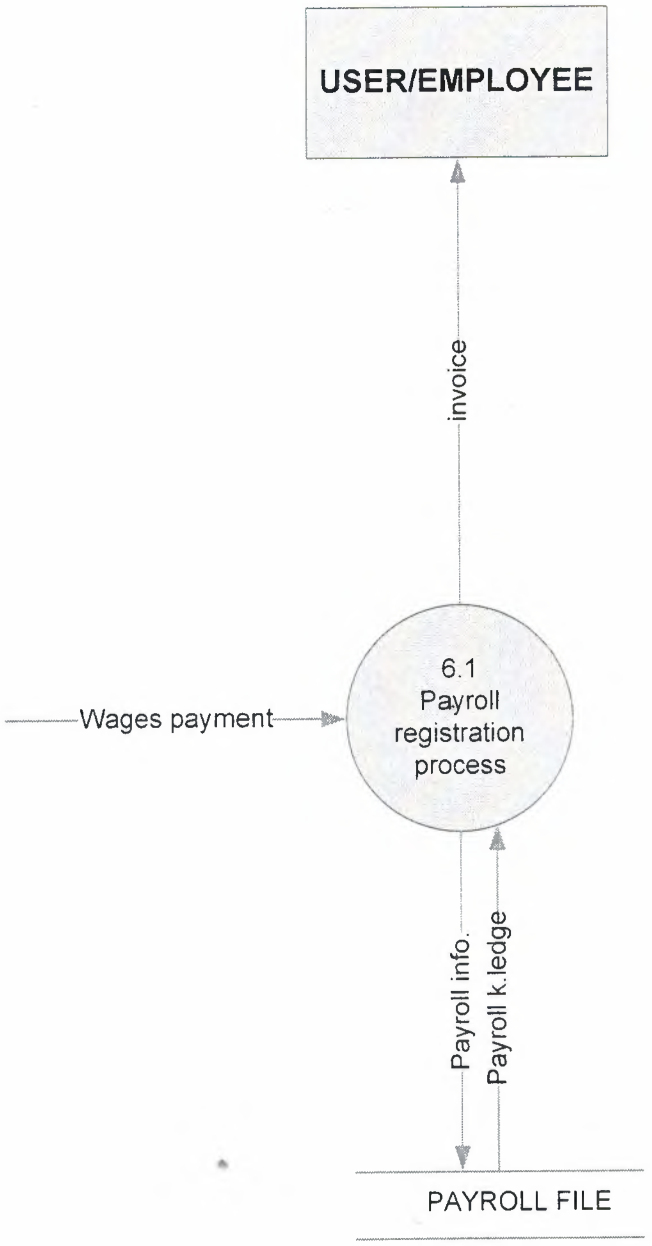
4. DETAIL DFD FOR ACCOUNTING PROCESS



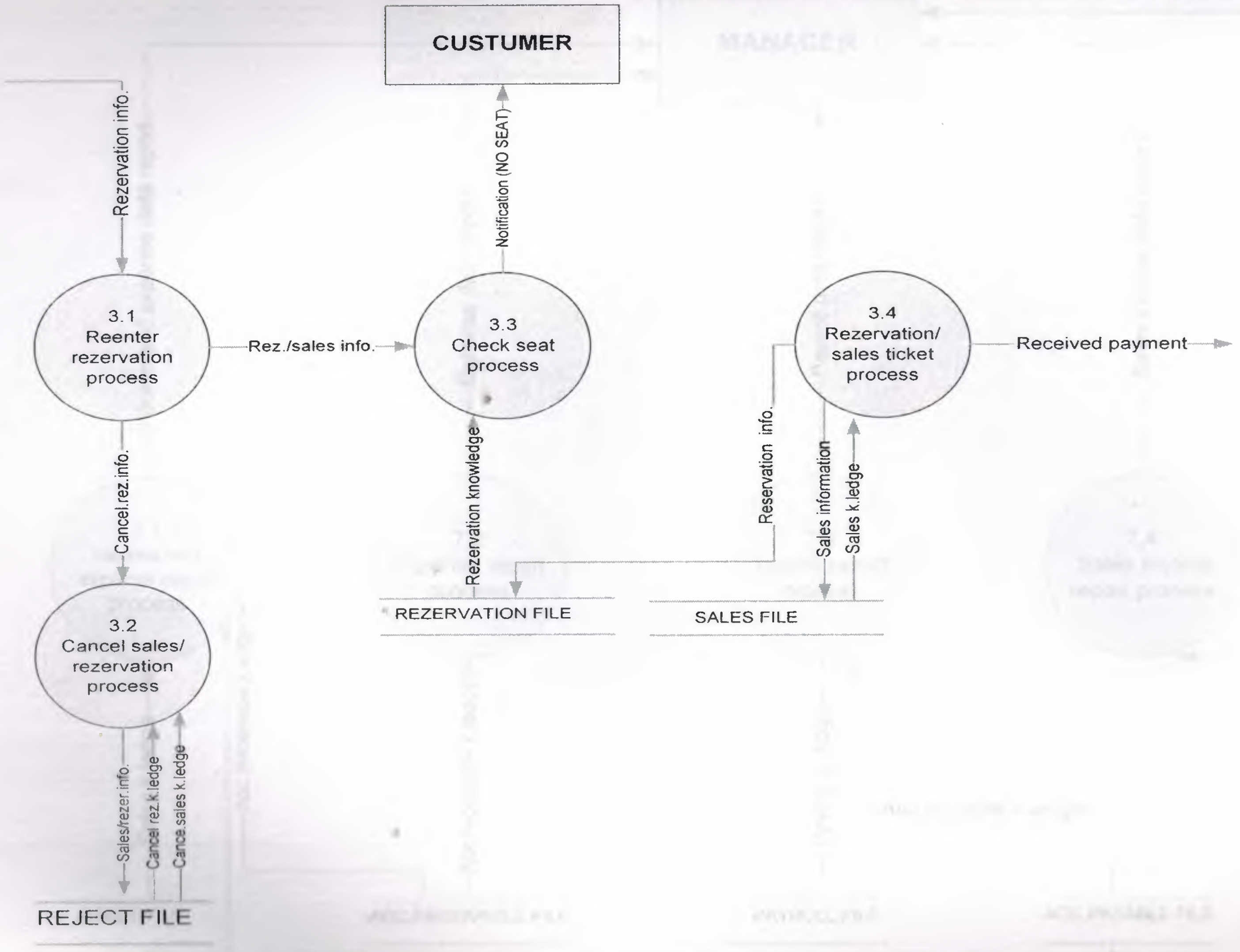
5. DETAIL DFD FOR EMPLOYEE PROCESS



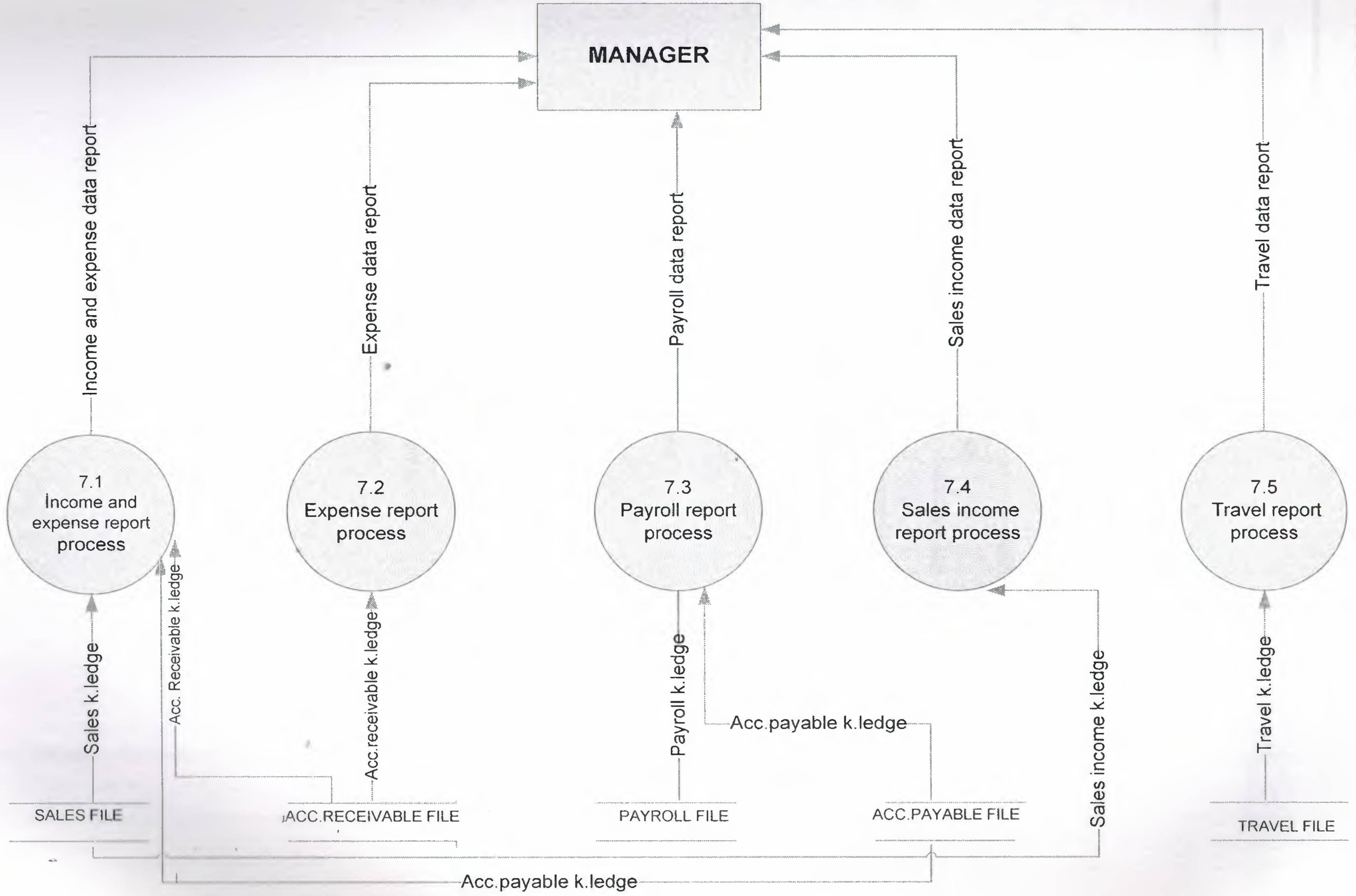
6. DETAIL DFD FOR PAYROLL PROCESS



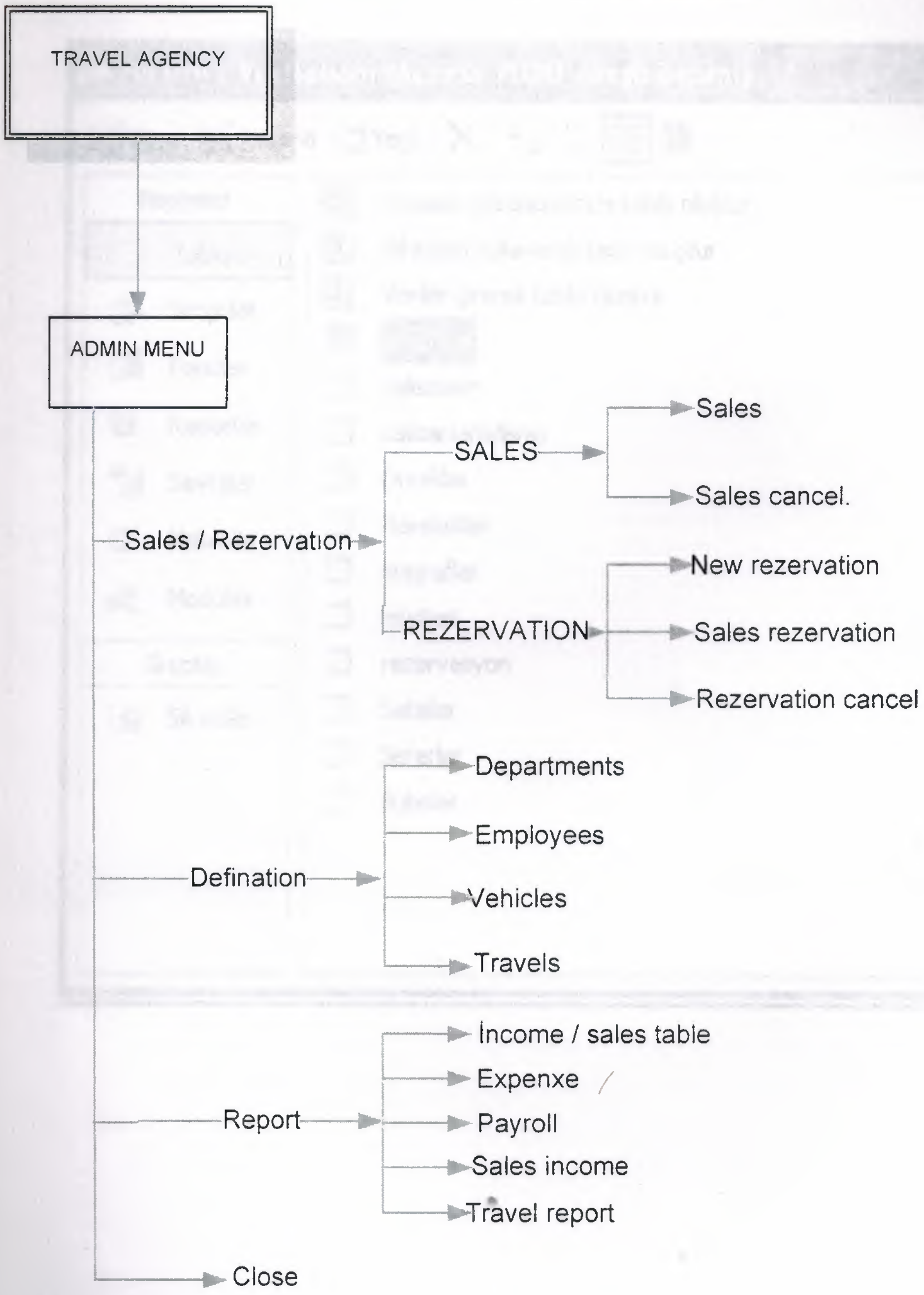
3. DETAIL DFD FOR REZERVATION/SALES PROCESS



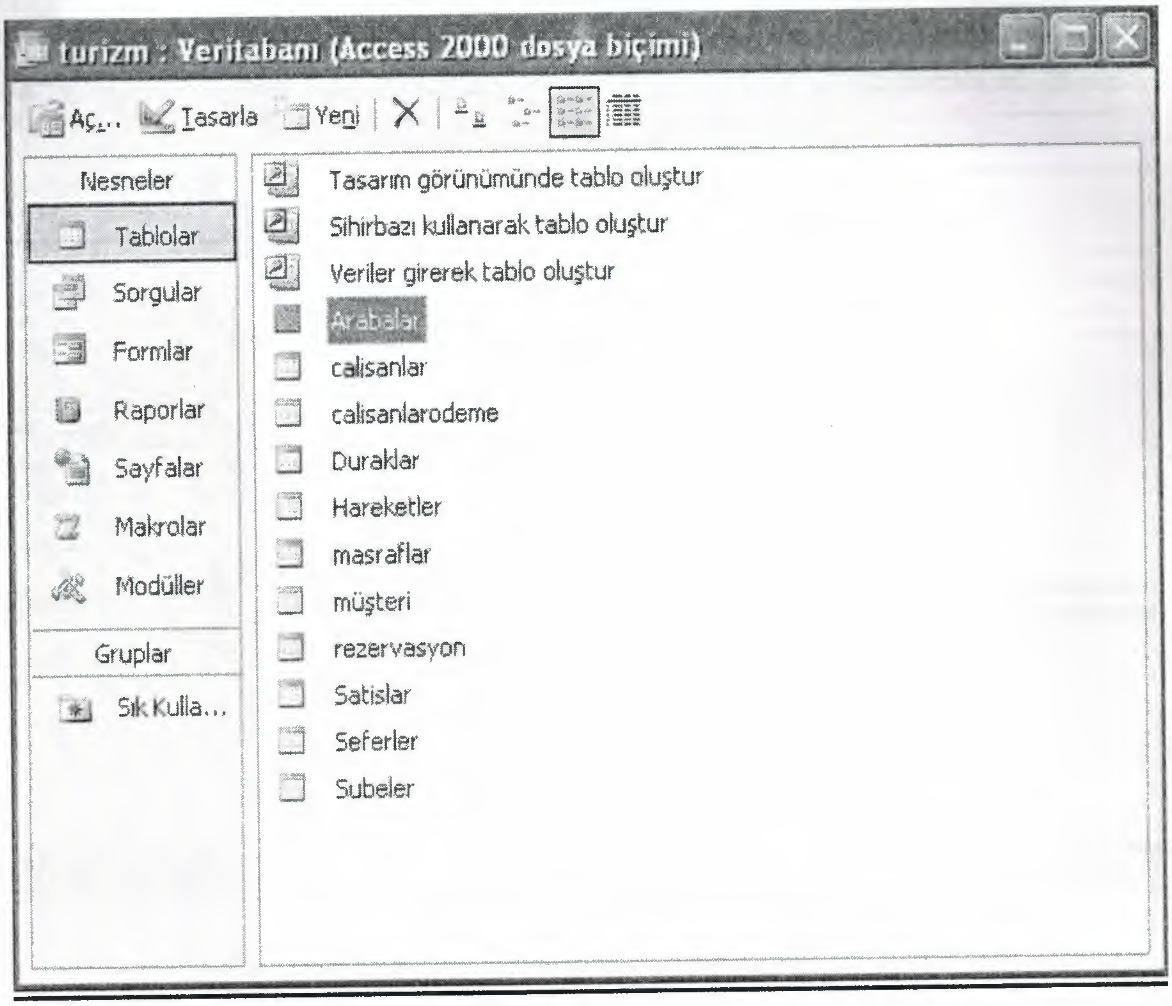
7. DETAIL DFD FOR REPORT PROCESS

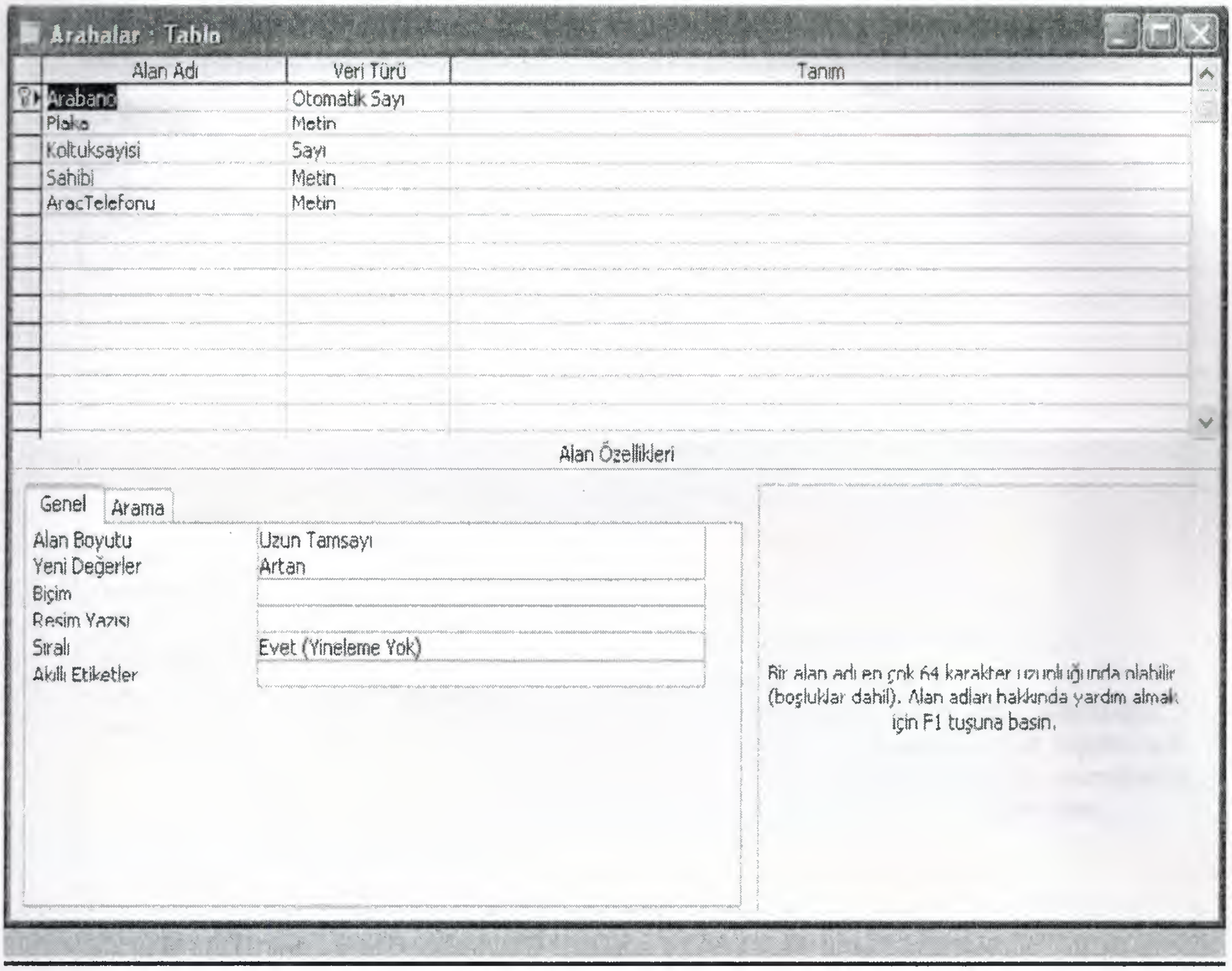


SYSTEM BLOCK DIAGRAM



DATABASE


turizm : Veritabanı (Access 2000 dosya biçimi)
Aç...
Tasarla
Yeni
Nesneler
Tablolar
Sorgular
Formlar
Raporlar
Sayfalar
Makrolar
Modüller
Gruplar
Sık Kulla...
Tasarım görünümünde tablo oluştur
Sihirbazı kullanarak tablo oluştur
Veriler girerek tablo oluştur
Arabalar
calisanlar
calisanlarodeme
Duraklar
Hareketler
masraflar
müşteri
rezervasyon
Satislar
Seferler
Subeler



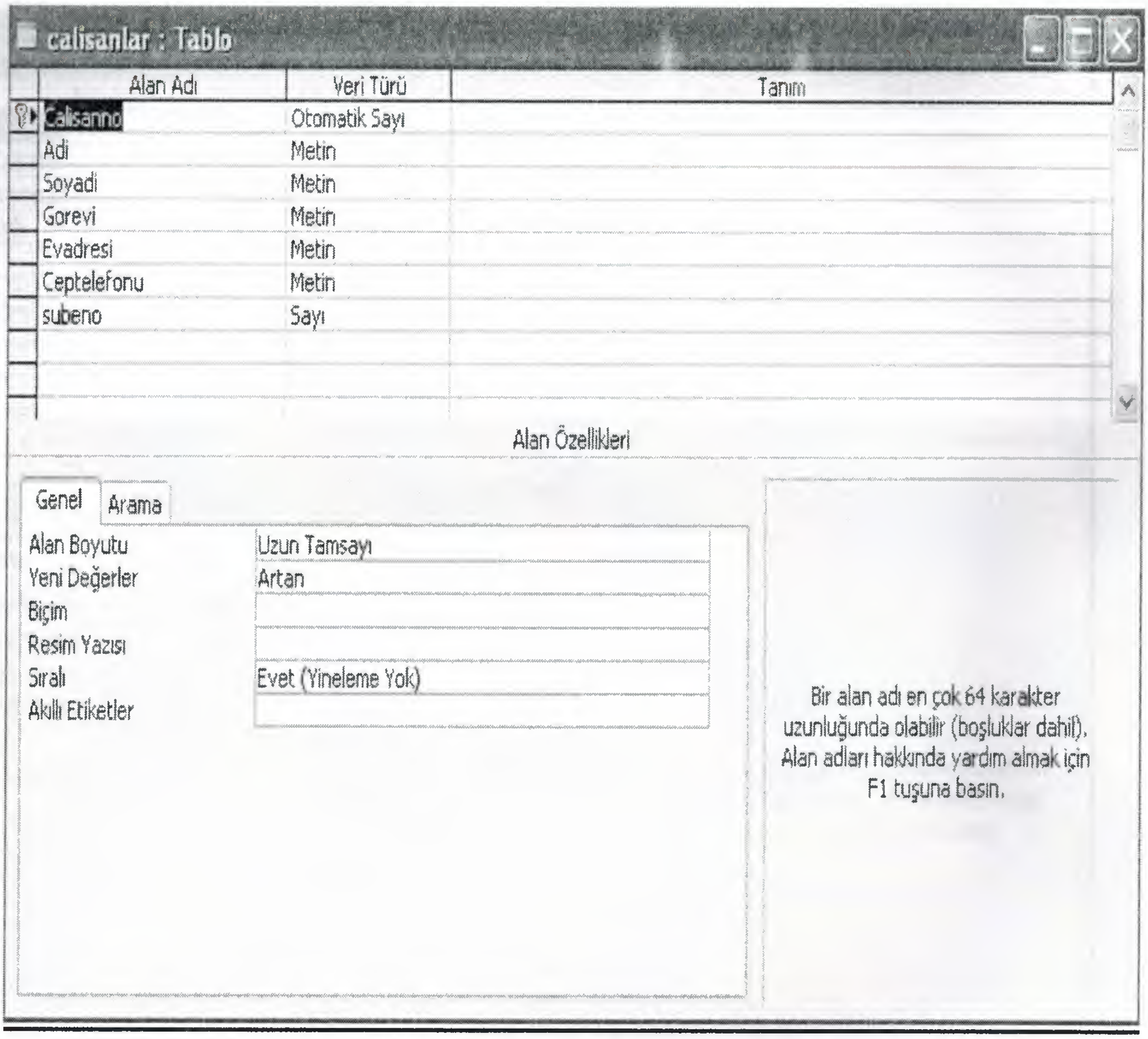
Arabalar : Tablo

Alan Adı	Veri Türü	Tanım
Arabano	Otomatik Sayı	
Plaka	Metin	
Koltuksayisi	Sayı	
Sahibi	Metin	
AracTelefonu	Metin	

Alan Özellikleri

Alan Boyutu	Uzun Tamsayı
Yeni Değerler	Artan
Biçim	
Resim Yazısı	
Sıralı	Evet (Yineleme Yok)
Akıllı Etiketler	

Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1 tuşuna basın.

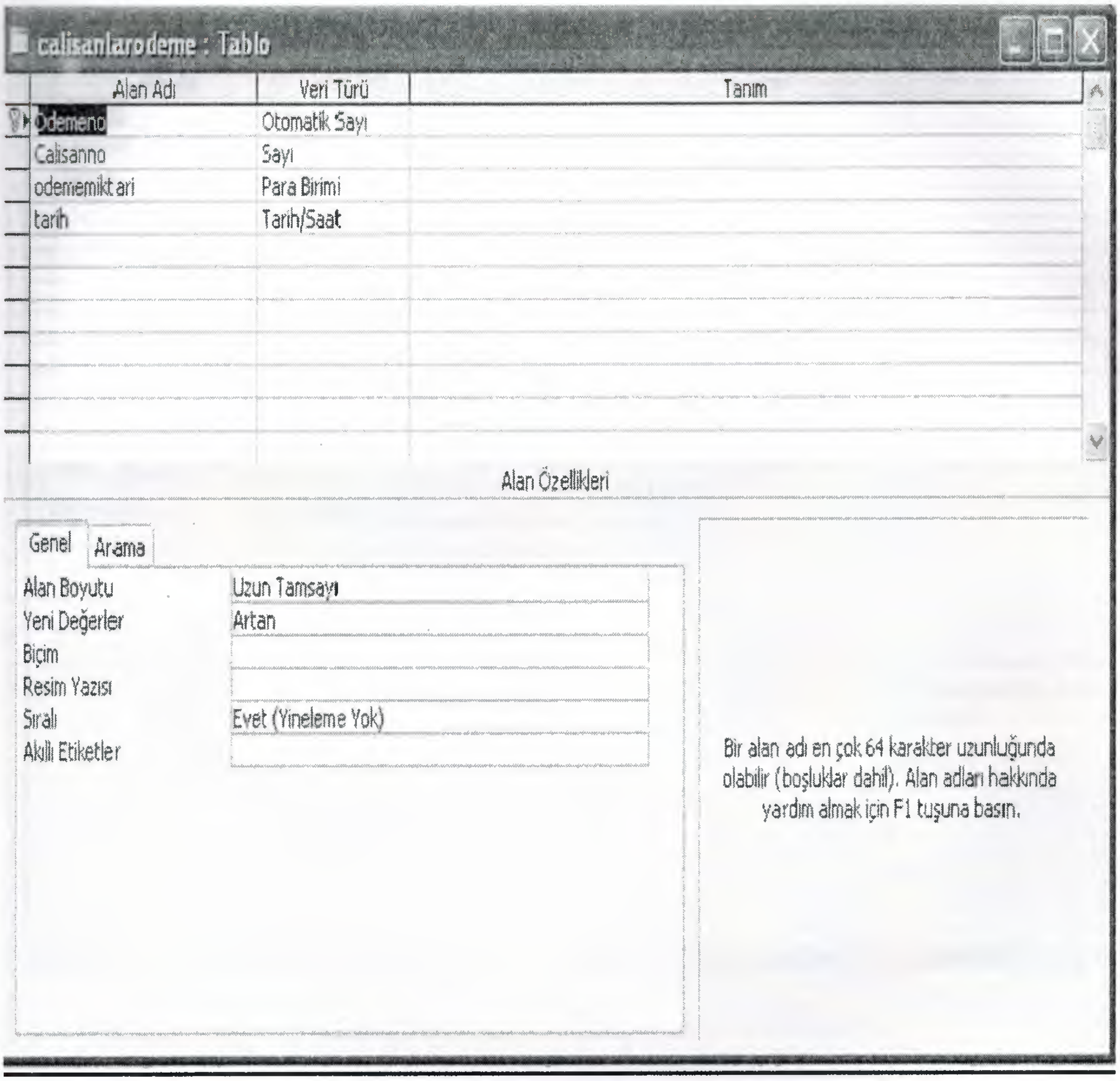


Alan Adı	Veri Türü	Tanım
Calisanno	Otomatik Sayı	
Adi	Metin	
Soyadi	Metin	
Gorevi	Metin	
Evadresi	Metin	
Ceptelefonu	Metin	
subeno	Sayı	

Alan Özellikleri

Alan Boyutu	Uzun Tamsayı
Yeni Değerler	Artan
Biçim	
Resim Yazısı	
Sıralı	Evet (Yineleme Yok)
Akıllı Etiketler	

Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1 tuşuna basın.

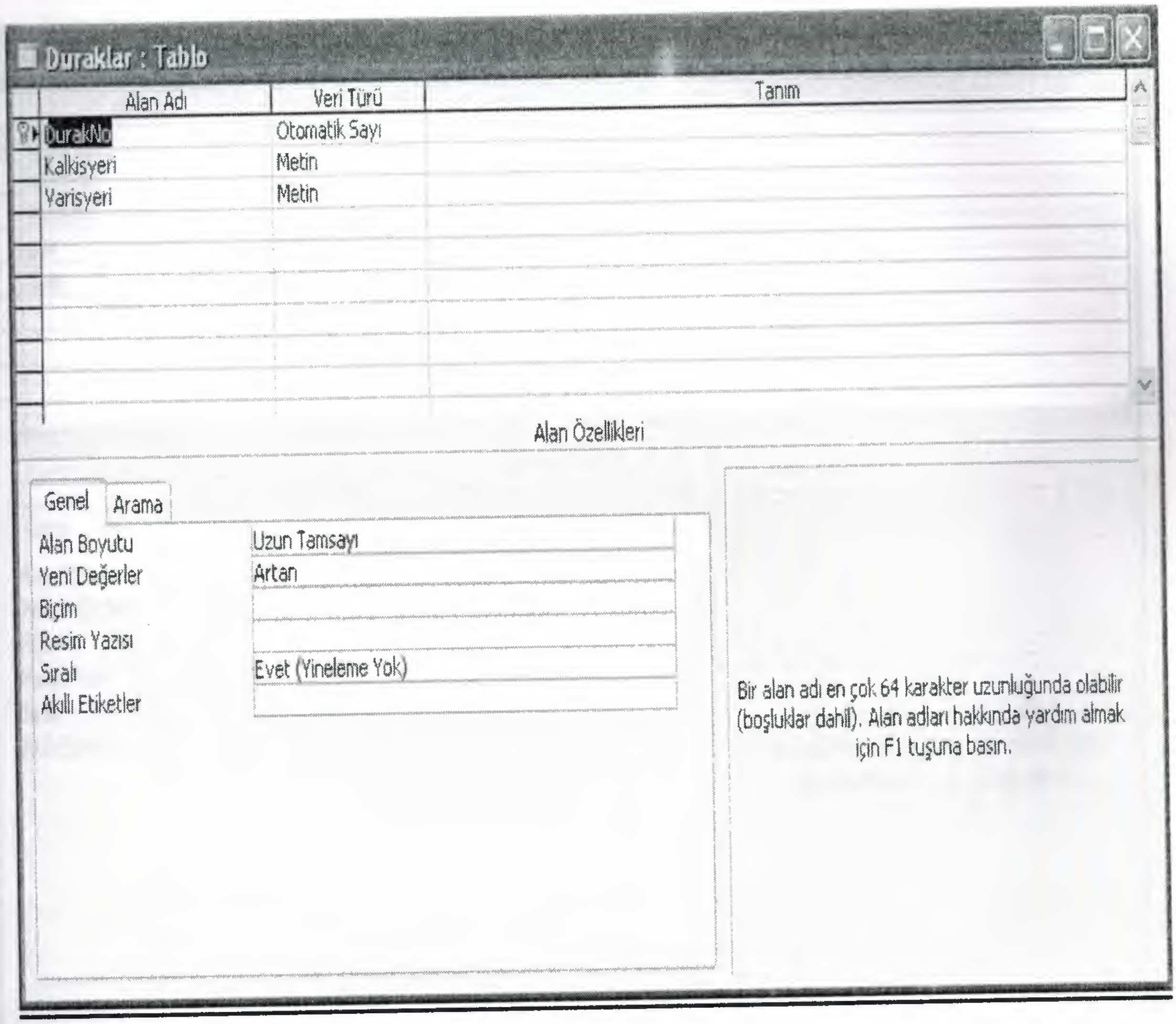


Alan Adı	Veri Türü	Tanım
Odemeno	Otomatik Sayı	
Calisanno	Sayı	
odememiktari	Para Birimi	
tarih	Tarih/Saat	

Alan Özellikleri

Alan Boyutu	Uzun Tamsayı
Yeni Değerler	Artan
Biçim	
Resim Yazısı	
Sıralı	Evet (Yineleme Yok)
Akıllı Etiketler	

Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1 tuşuna basın.

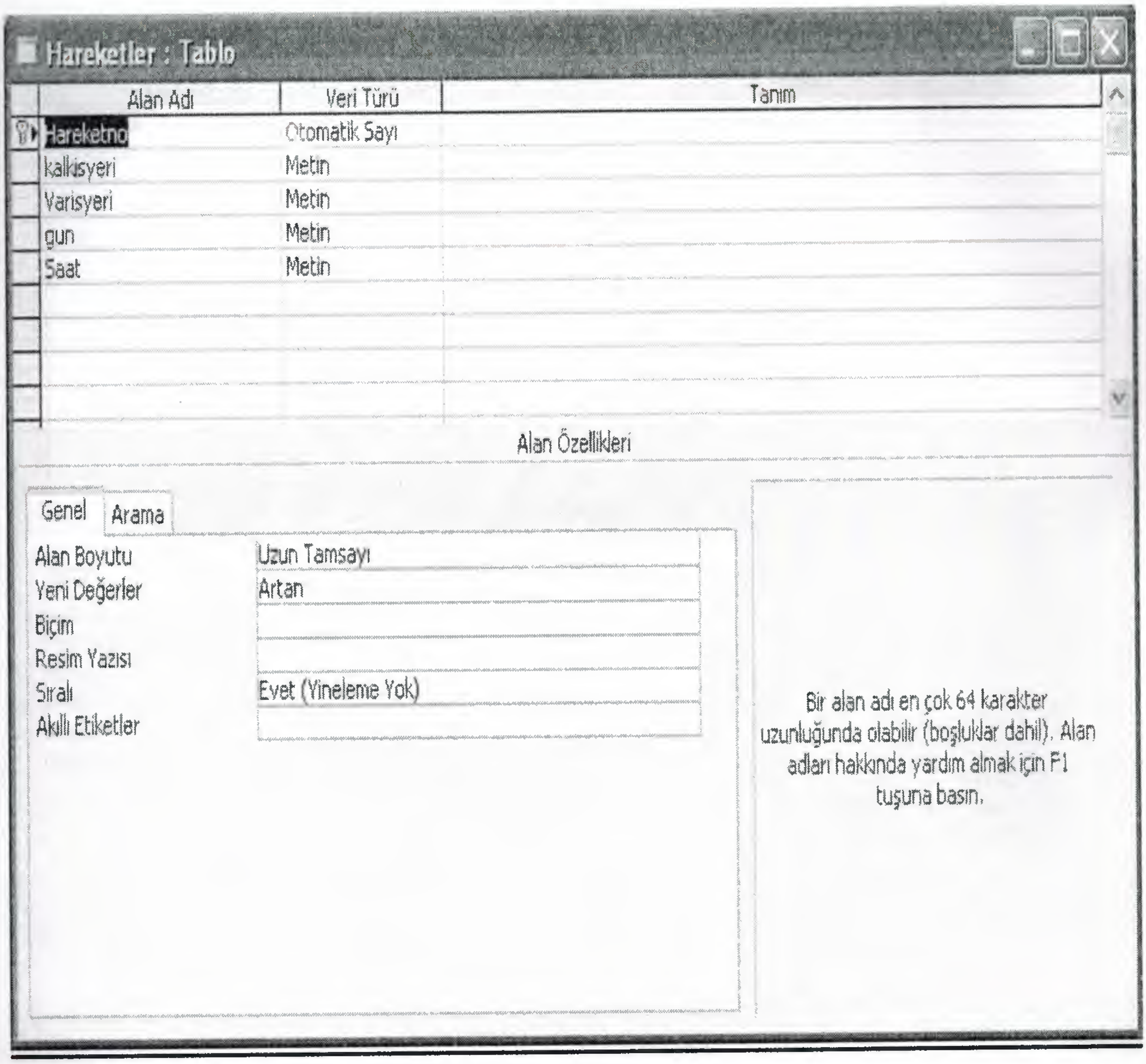


Alan Adı	Veri Türü	Tanım
DurakNo	Otomatik Sayı	
Kalkisyeri	Metin	
Varisyeri	Metin	

Alan Özellikleri

Alan Boyutu	Uzun Tamsayı
Yeni Değerler	Artan
Biçim	
Resim Yazısı	
Sıralı	Evet (Yineleme Yok)
Akıllı Etiketler	

Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1 tuşuna basın.

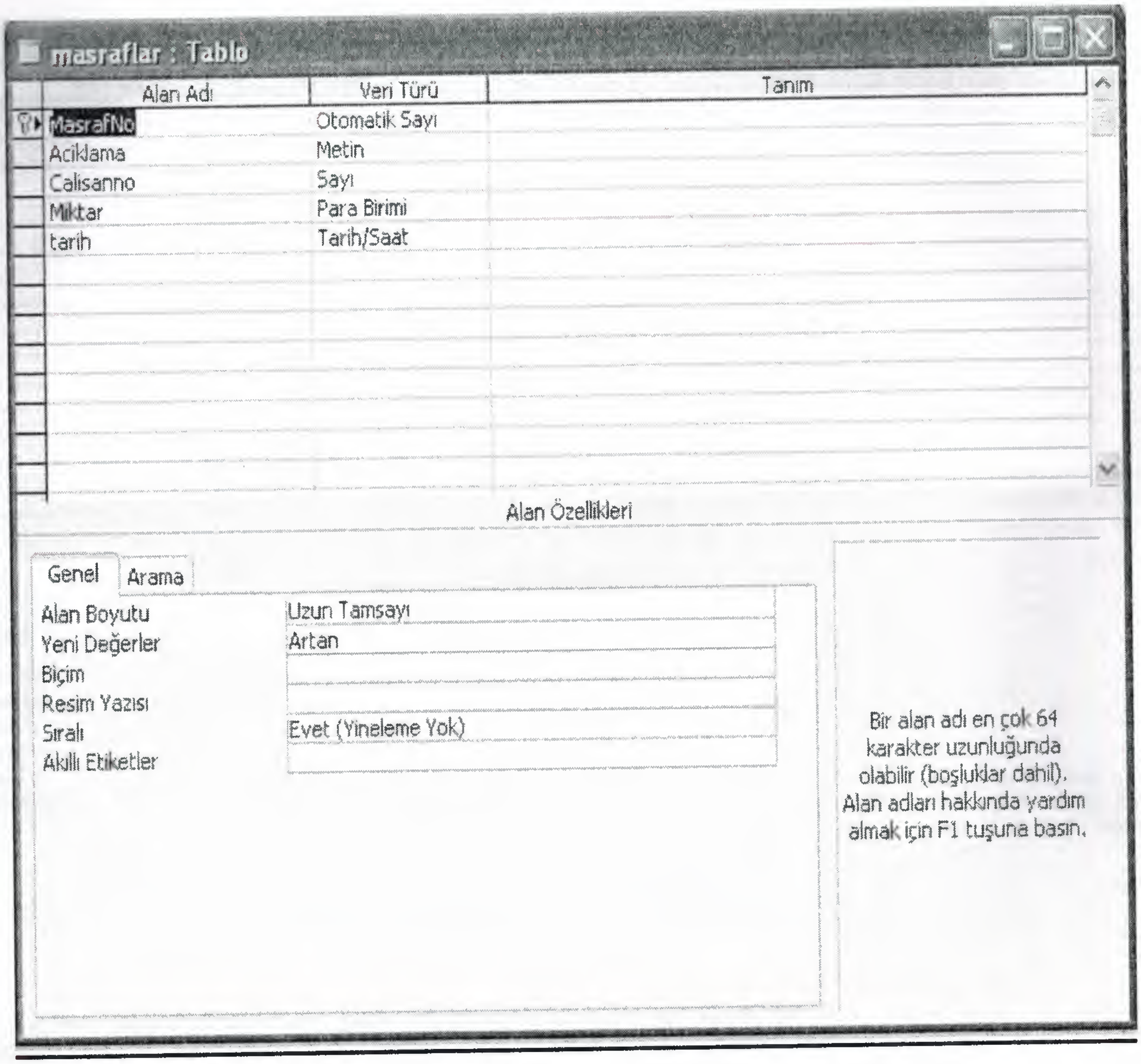


Alan Adı	Veri Türü	Tanım
Hareketno	Otomatik Sayı	
kalkisyeri	Metin	
Varisyeri	Metin	
gun	Metin	
Saat	Metin	

Alan Özellikleri

Alan Boyutu	Uzun Tamsayı
Yeni Değerler	Artan
Biçim	
Resim Yazısı	
Sıralı	Evet (Yineleme Yok)
Akıllı Etiketler	

Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1 tuşuna basın.



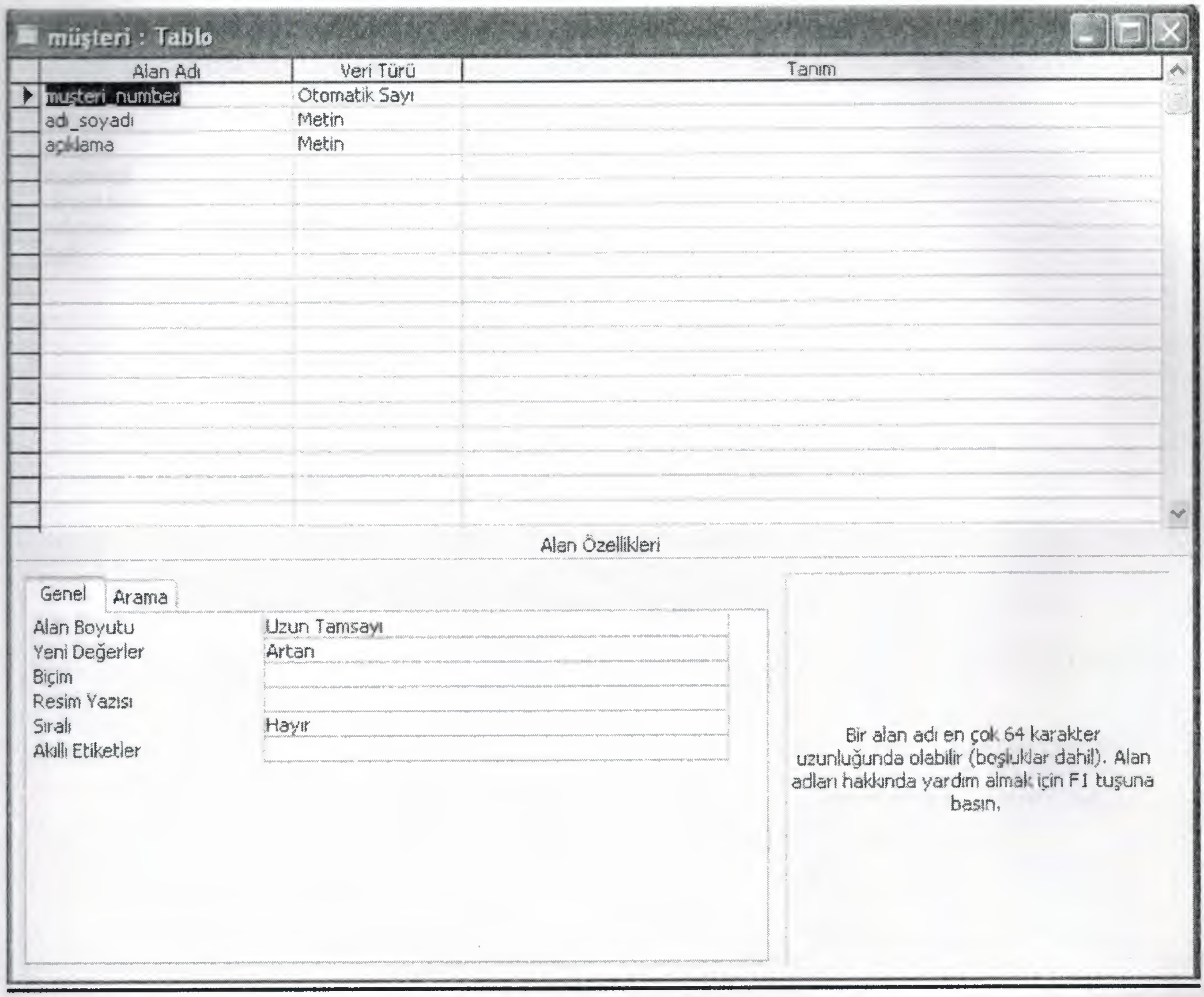
masraflar : Tablo

Alan Adı	Veri Türü	Tanım
MasrafNo	Otomatik Sayı	
Aciklama	Metin	
Calisanno	Sayı	
Miktar	Para Birimi	
tarih	Tarih/Saat	

Alan Özellikleri

Alan Boyutu	Uzun Tamsayı
Yeni Değerler	Artan
Biçim	
Resim Yazısı	
Sıralı	Evet (Yineleme Yok)
Akıllı Etiketler	

Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1 tuşuna basın.

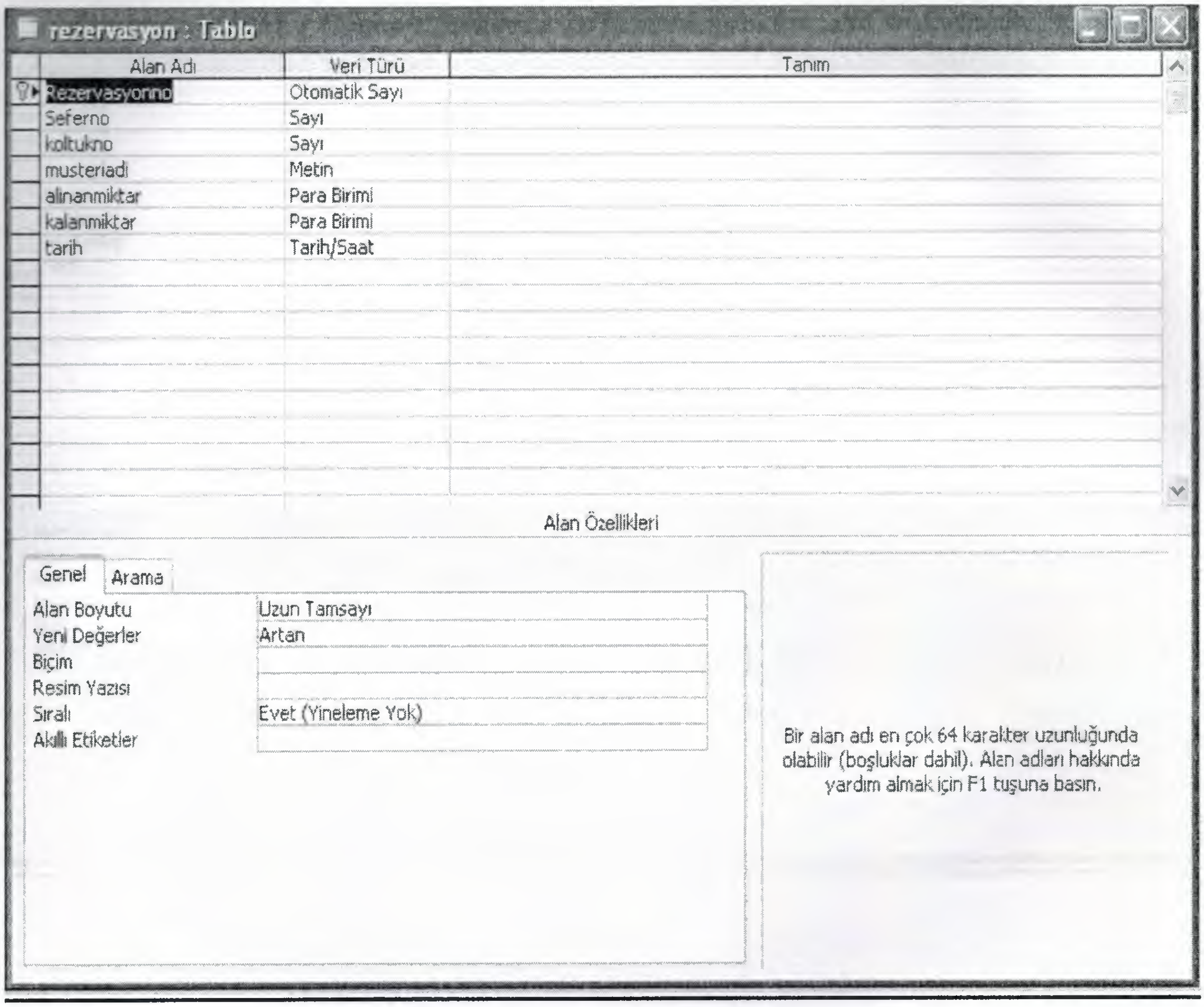


Alan Adı	Veri Türü	Tanım
muşteri_number	Otomatik Sayı	
adı_soyadı	Metin	
açıklama	Metin	

Alan Özellikleri

Alan Boyutu	Uzun Tamsayı
Yeni Değerler	Artan
Biçim	
Resim Yazısı	
Sıralı	Hayır
Akıllı Etiketler	

Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1 tuşuna basın.

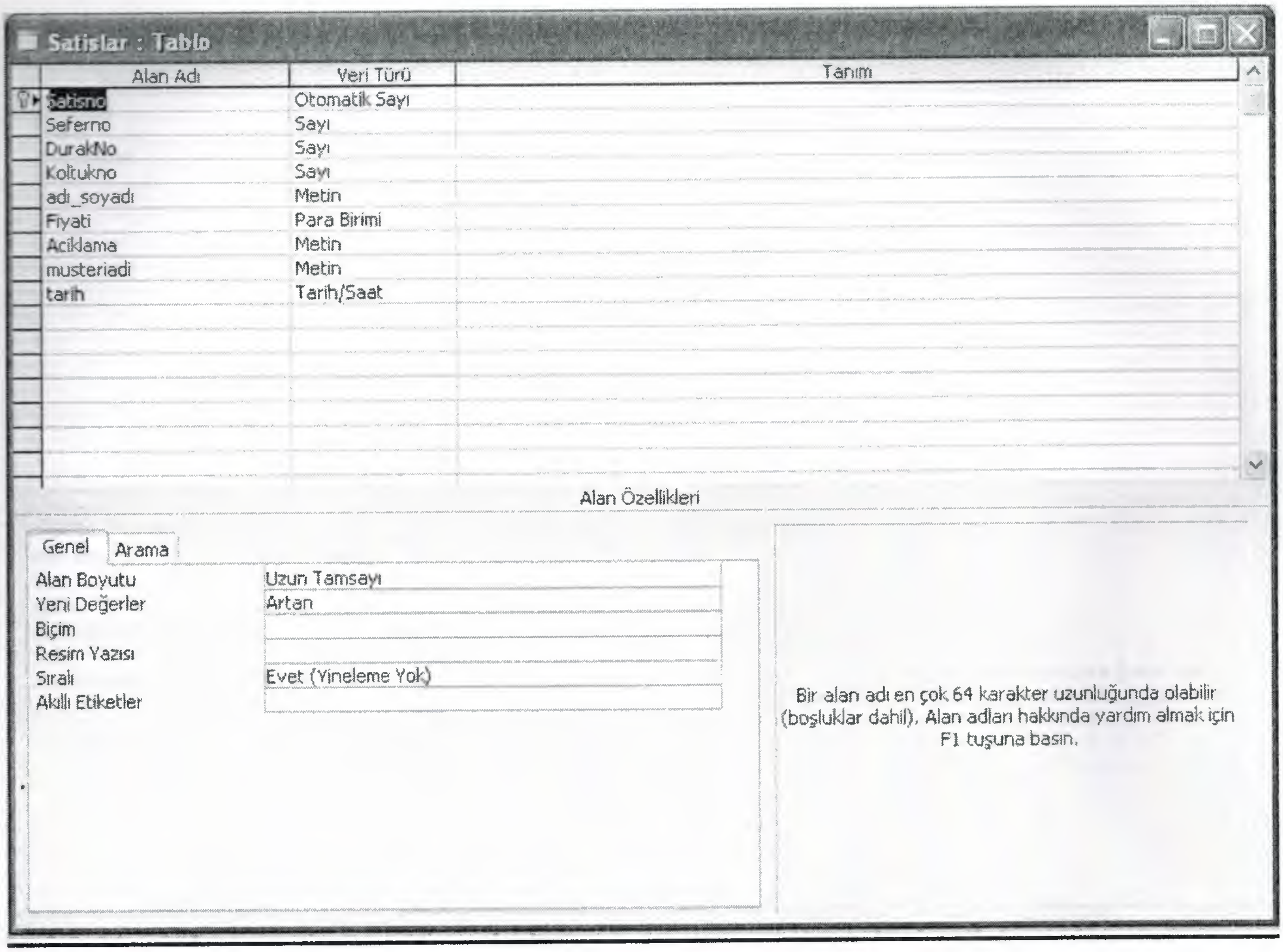


Alan Adı	Veri Türü	Tanım
Rezervasyonno	Otomatik Sayı	
Seferno	Sayı	
koltukno	Sayı	
musteriadi	Metin	
alinanmiktar	Para Birimi	
kalanmiktar	Para Birimi	
tarih	Tarih/Saat	

Alan Özellikleri

Alan Boyutu	Uzun Tamsayı
Yeni Değerler	Artan
Biçim	
Resim Yazısı	
Sıralı	Evet (Yineleme Yok)
Akıllı Etiketler	

Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1 tuşuna basın.



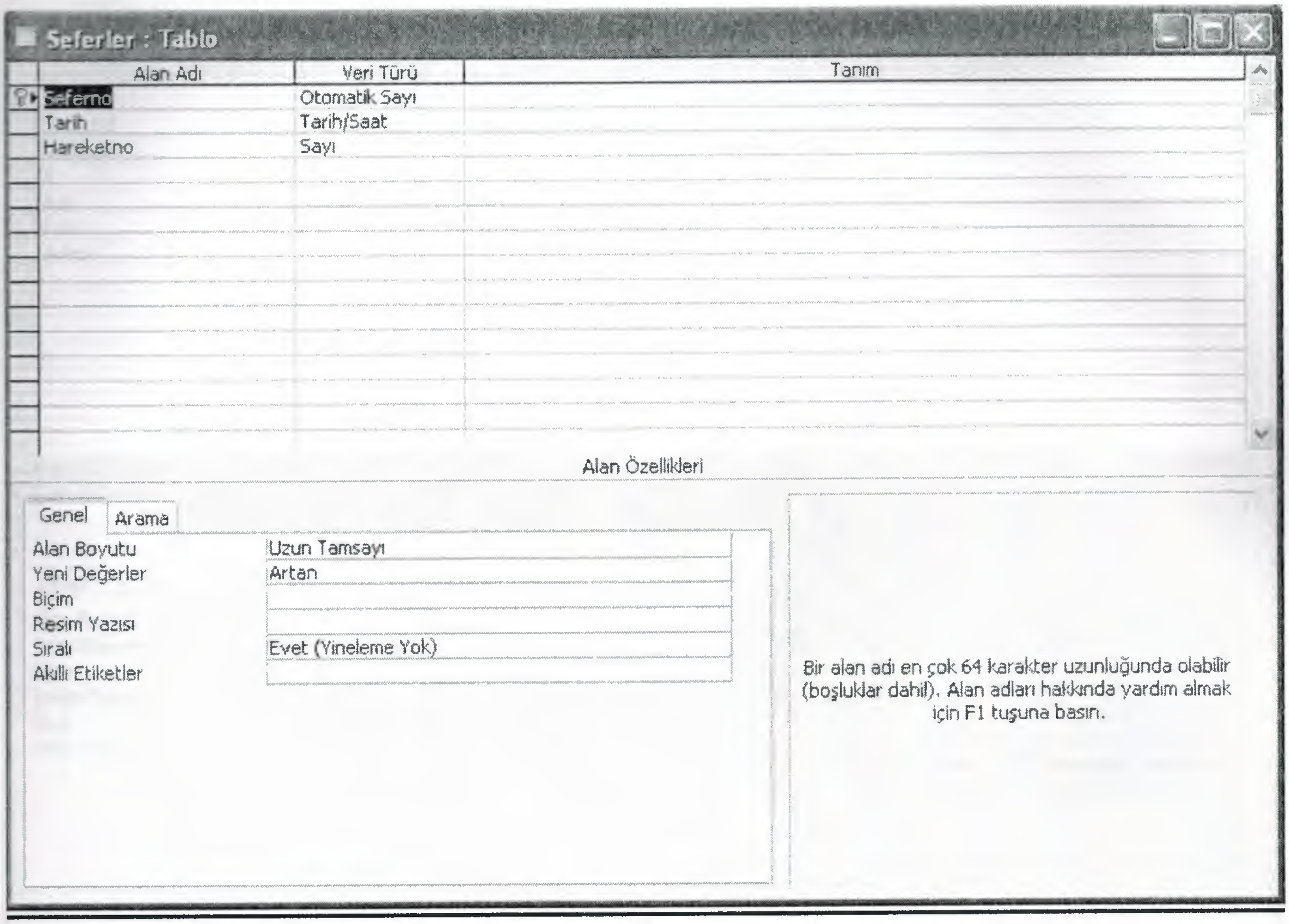
Satislar : Tablo

Alan Adı	Veri Türü	Tanım
satisno	Otomatik Sayı	
Seferno	Sayı	
DurakNo	Sayı	
Koltukno	Sayı	
adı_soyadı	Metin	
Fiyati	Para Birimi	
Aciklama	Metin	
musteriadi	Metin	
tarih	Tarih/Saat	

Alan Özellikleri

Alan Boyutu	Uzun Tamsayı
Yeni Değerler	Artan
Biçim	
Resim Yazısı	
Sıralı	Evet (Yineleme Yok)
Akıllı Etiketler	

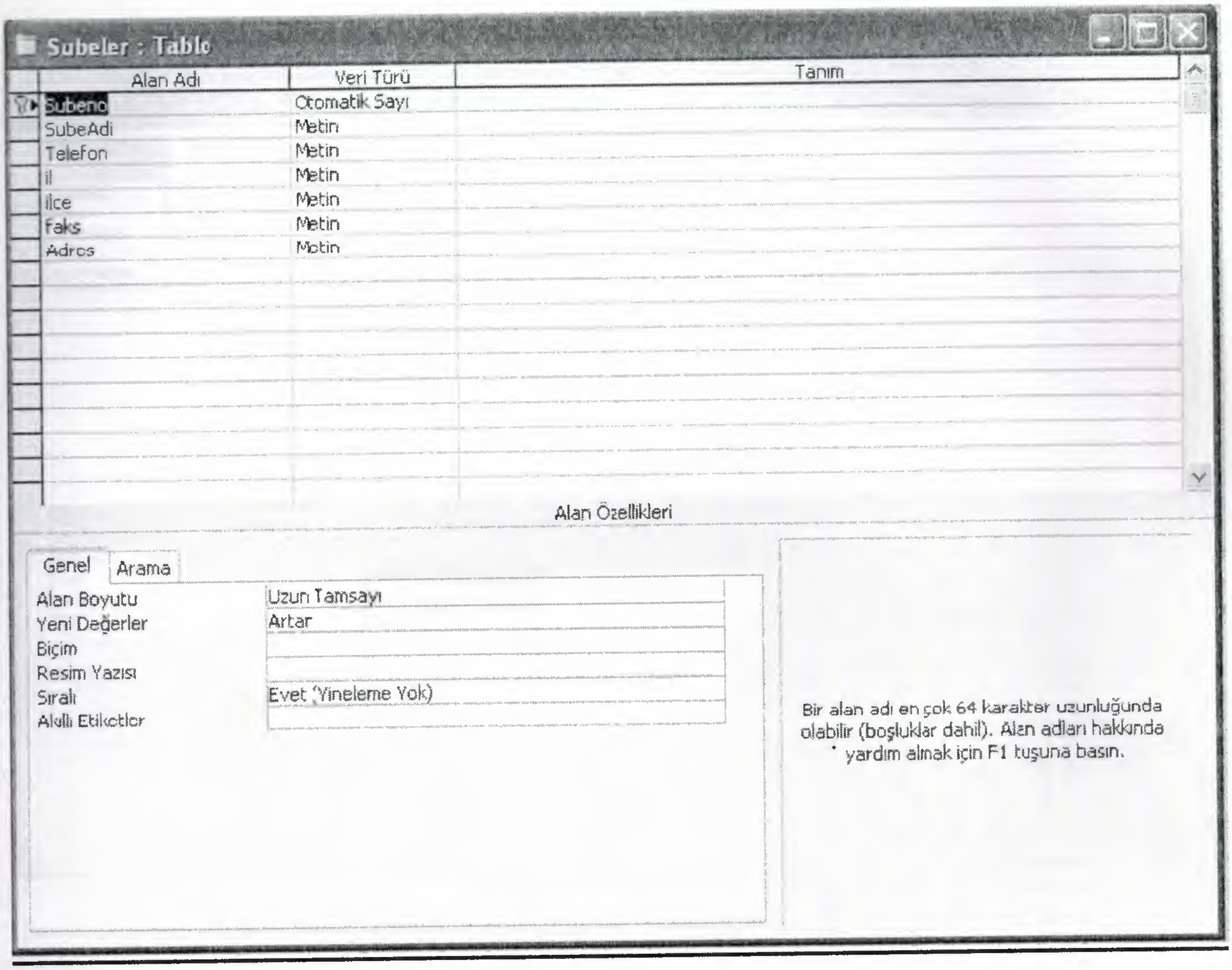
Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1 tuşuna basın.



Alan Adı	Veri Türü	Tanım
Seferno	Otomatik Sayı	
Tarih	Tarih/Saat	
Hareketno	Sayı	

Alan Boyutu	Uzun Tamsayı
Yeni Değerler	Artan
Biçim	
Resim Yazısı	
Sıralı	Evet (Yineleme Yok)
Akıllı Etiketler	

Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1 tuşuna basın.



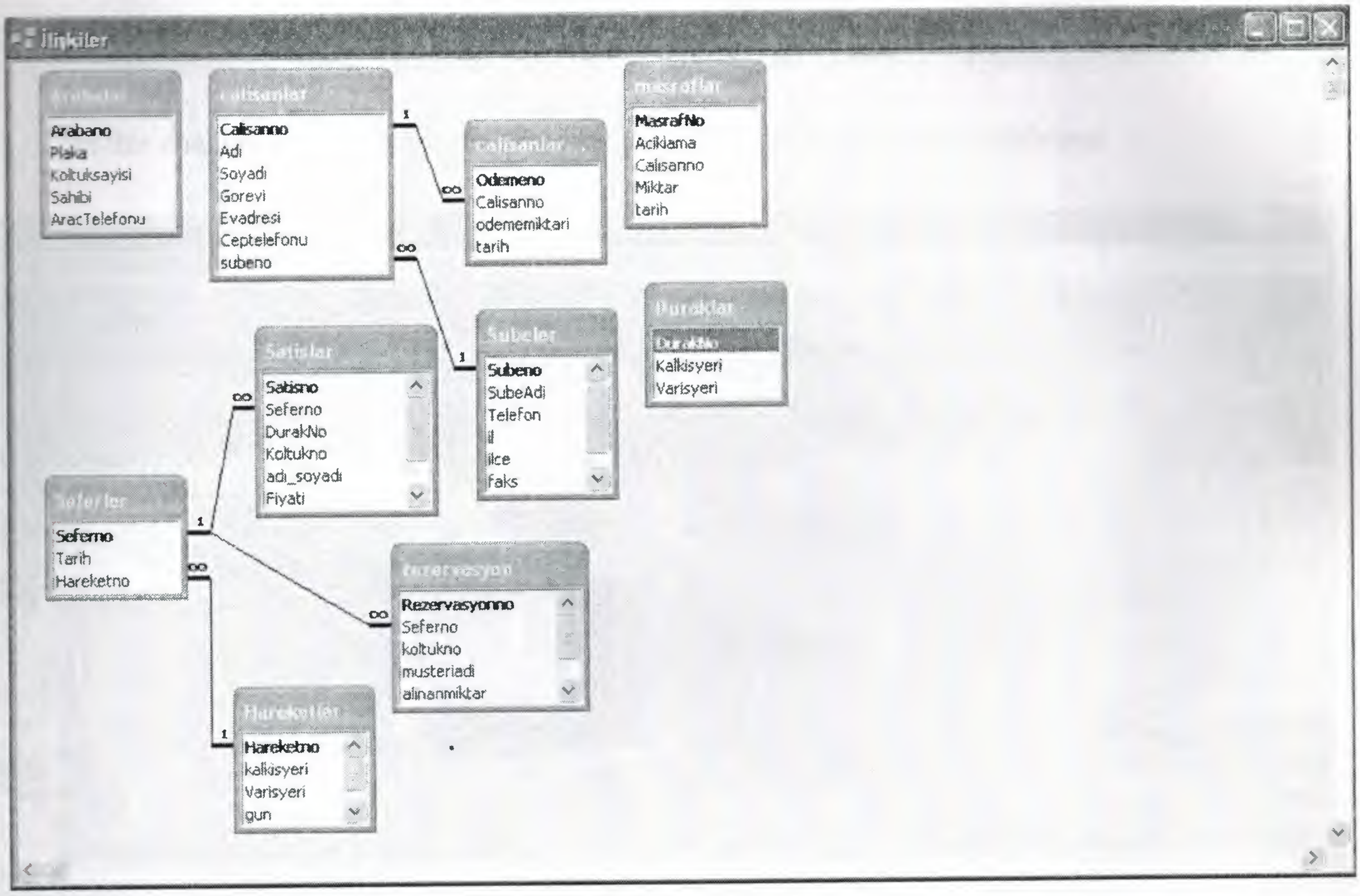
Alan Adı	Veri Türü	Tanım
Subeno	Otomatik Sayı	
SubeAdi	Metin	
Telefon	Metin	
il	Metin	
ilce	Metin	
Faks	Metin	
Adres	Metin	

Alan Özellikleri

Alan Boyutu	Uzun Tamsayı
Yeni Değerler	Artar
Biçim	
Resim Yazısı	
Sıralı	Evet (Yineleme Yok)
Akıllı Etiketler	

Bir alan adı en çok 64 karakter uzunluğunda olabilir (boşluklar dahil). Alan adları hakkında yardım almak için F1 tuşuna basın.

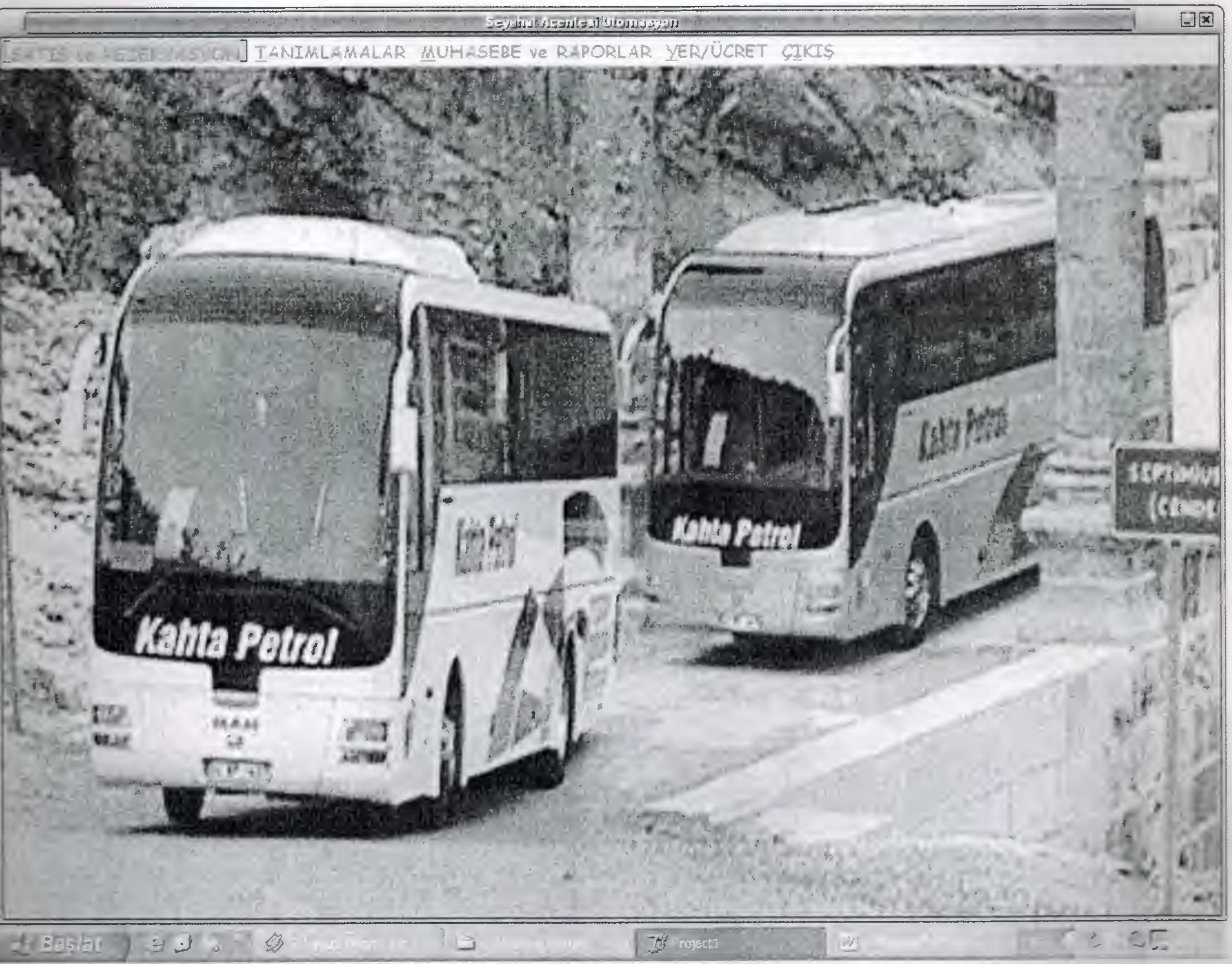
RELATIONSHIP



5. USER MANUEL

TRAVEL AGENCY AUTOMATION SYSTEM

In this chapter the screenshots and the user manual of the program seen explained.



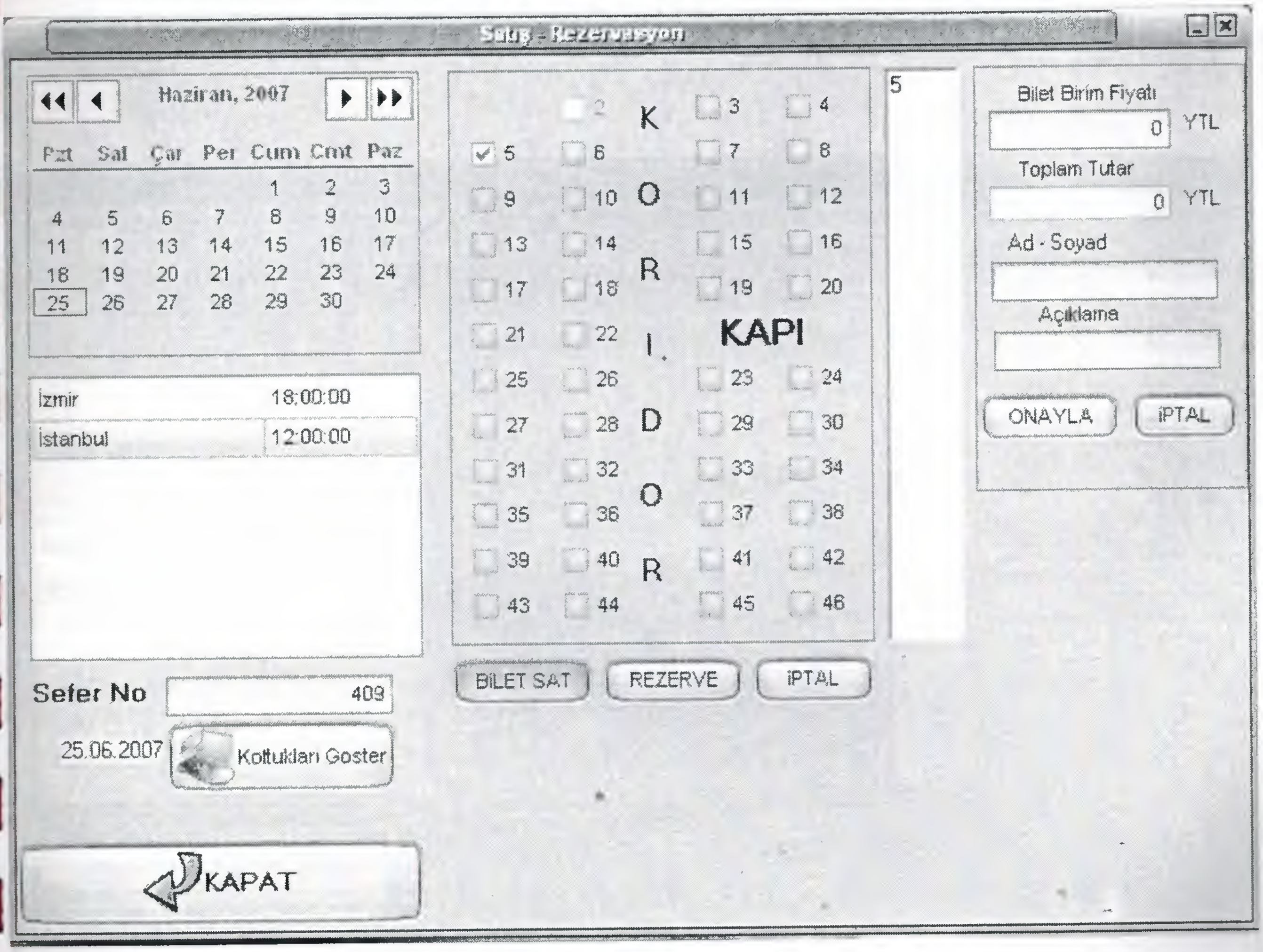
(a) Main Menu View

In this main menu we can see that there is a calendar and an analog watch. These are user friendly interface elements. The Calendar shows the current date. The watch is dynamic and shows the computers current time.

There are several buttons. Each butto shows another window.

The sewen buttons on the bottom of the form are related to accounting and reports.

The program codes of this menu is in Appendix A - 1



Satış - Rezervasyon

Haziran, 2007

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

İzmir	18:00:00
İstanbul	12:00:00

Sefer No 409

25.06.2007

	□ 2	K	□ 3	□ 4
☑ 5	□ 6		□ 7	□ 8
□ 9	□ 10	O	□ 11	□ 12
□ 13	□ 14		□ 15	□ 16
□ 17	□ 18	R	□ 19	□ 20
□ 21	□ 22	I	KAPI	
□ 25	□ 26		□ 23	□ 24
□ 27	□ 28	D	□ 29	□ 30
□ 31	□ 32		□ 33	□ 34
□ 35	□ 36	O	□ 37	□ 38
□ 39	□ 40	R	□ 41	□ 42
□ 43	□ 44		□ 45	□ 46

5

Bilet Birim Fiyatı 0 YTL

Toplam Tutar 0 YTL

Ad - Soyad

Açıklama

(b) Seat Sale

The seat sale form. In this form user double click on the calendar to filter the voyages of the day. After that user select the voyage from the list on the left. And then user click on the "show the seats" button and see the available seats.

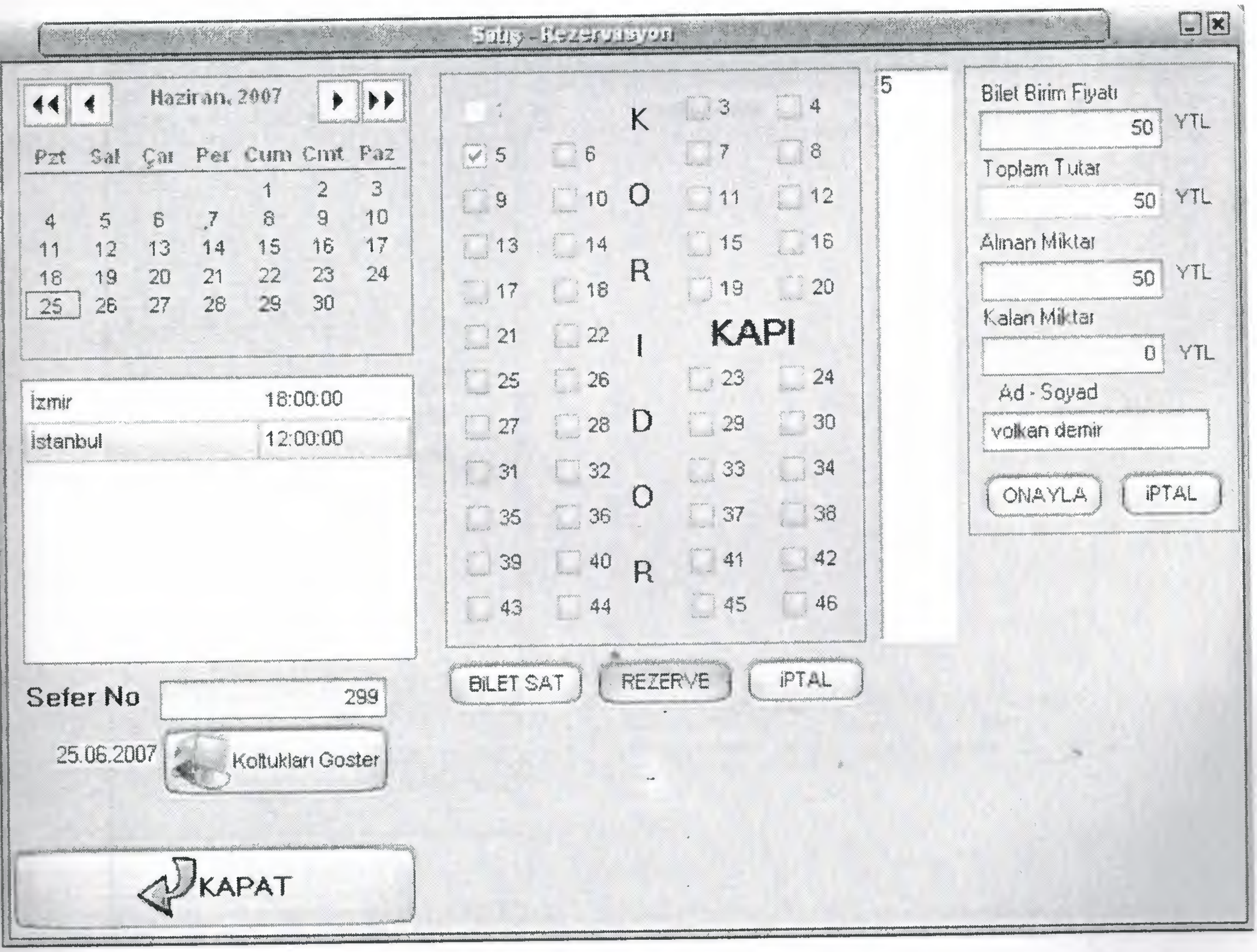
The sold seats will be invisible. And the reserved seats will not be enabled.

After choosing the seats to sell we click on the "sell the tickets" button.

From the list on the right user can see which seats are selected and the enters the unit price of the seat and a escription (this description can be the place where the passenger will get out) .

Then the user clicks on the confirm button and the sale process is been confirmed.

The program codes of this form is in Appendix A - 2



İzmir	18:00:00
İstanbul	12:00:00

Sefer No 299

25.06.2007

□ 1 □ 3 □ 4
☑ 5 □ 6 □ 7 □ 8
□ 9 □ 10 □ 11 □ 12
□ 13 □ 14 □ 15 □ 16
□ 17 □ 18 □ 19 □ 20
□ 21 □ 22
□ 25 □ 26 □ 23 □ 24
□ 27 □ 28 □ 29 □ 30
□ 31 □ 32 □ 33 □ 34
□ 35 □ 36 □ 37 □ 38
□ 39 □ 40 □ 41 □ 42
□ 43 □ 44 □ 45 □ 46

5

Bilet Birim Fiyatı 50 YTL

Toplam Tutar 50 YTL

Alınan Miktar 50 YTL

Kalan Miktar 0 YTL

Ad - Soyad volkan demir

(c) Seat Reservation

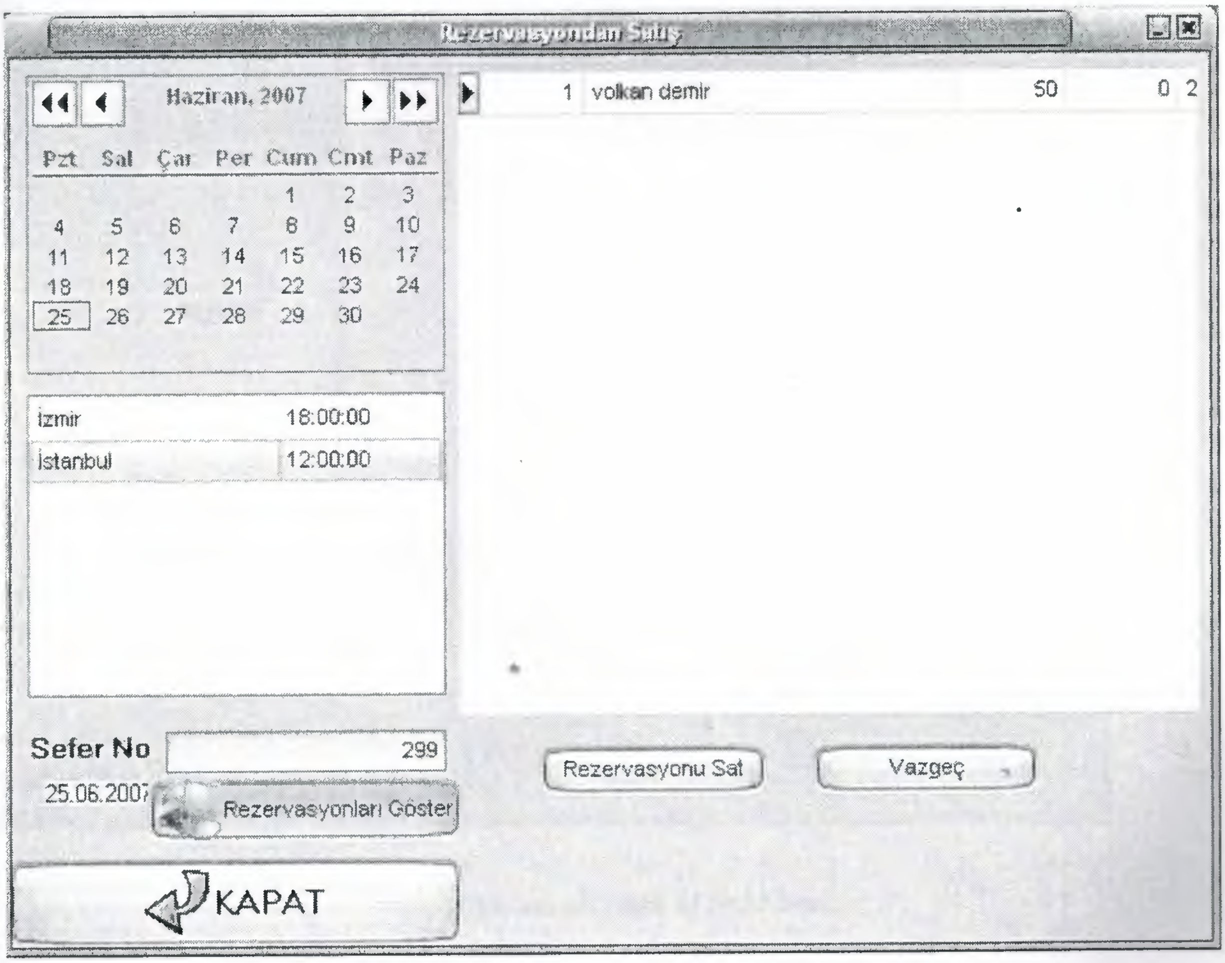
The seat reservation form. In this form user double click on the calendar to filter the voyages of the day. After that user select the voyage from the list on the left. And then user click on the "show the seats" button and see the available seats.

The sold seats will be invisible. And the reserved seats will not be enabled.

After choosing the seats to sell we click on the "reserve the seats" button.

From the list on the right user can see which seats are selected and the enters the unit price of the seat, amount received and name surname of the reserver

The program codes of this form is in Appendix A - 2



Rezervasyondan Satış

Haziran, 2007

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

İzmir	18:00:00
İstanbul	12:00:00

▶	1	volkan demir	50	0	2

Sefer No 299

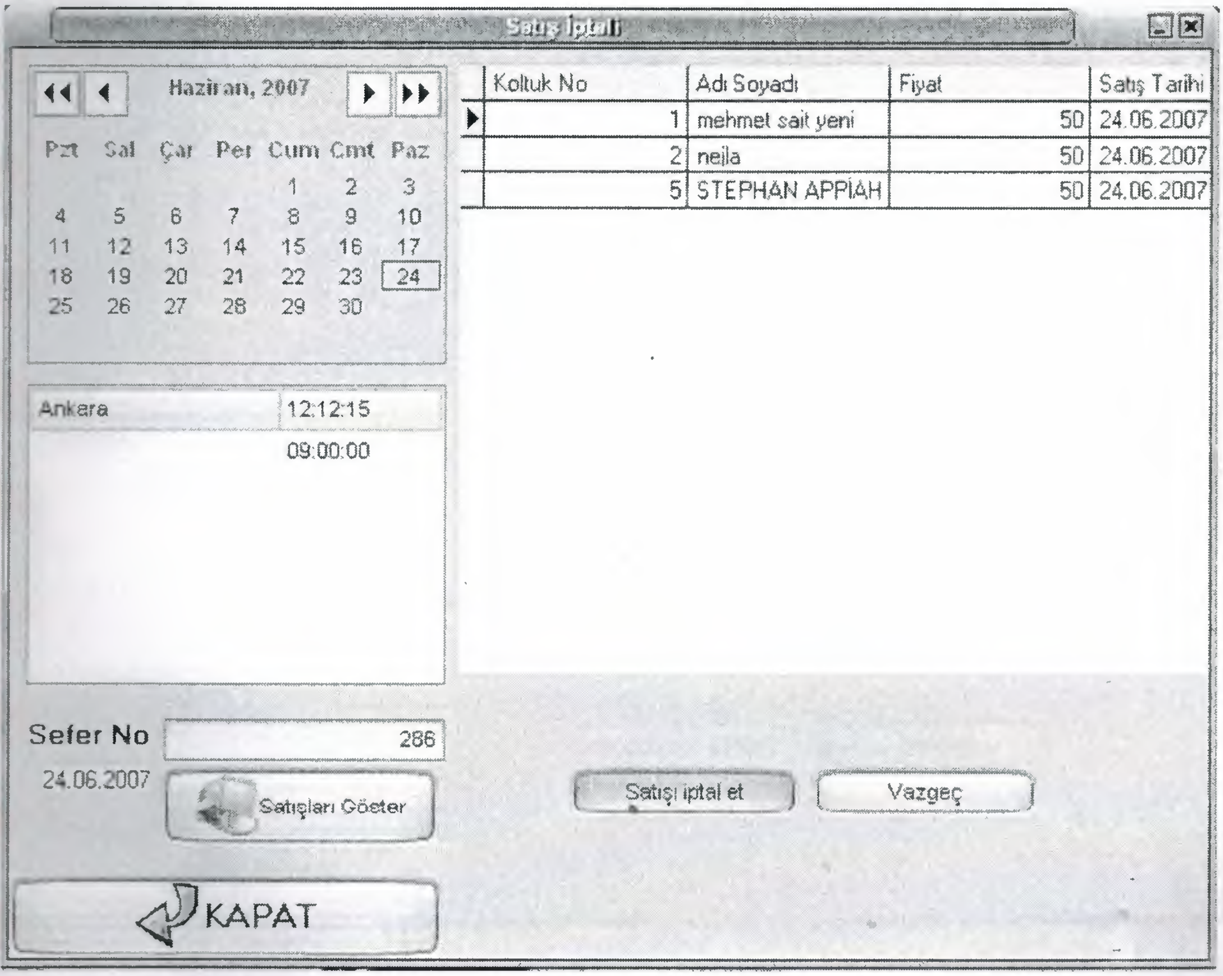
25.06.2007

Sale of Reserved Seat

The sale of reserved seats form. In this form user double click on the calendar to filter the voyages of the day. After that user select the voyage from the list on the left. And then user click on the "show the reservations" button and see the reserved seats.

After that user selects the reserved seat and click on the Sale reserved seat .

The program codes of this form is in Appendix A - 3



Pzt	Sal	Çar	Per	Cum	Cmt	Paz
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Ankara	12:12:15
	09:00:00

Sefer No 286

24.06.2007

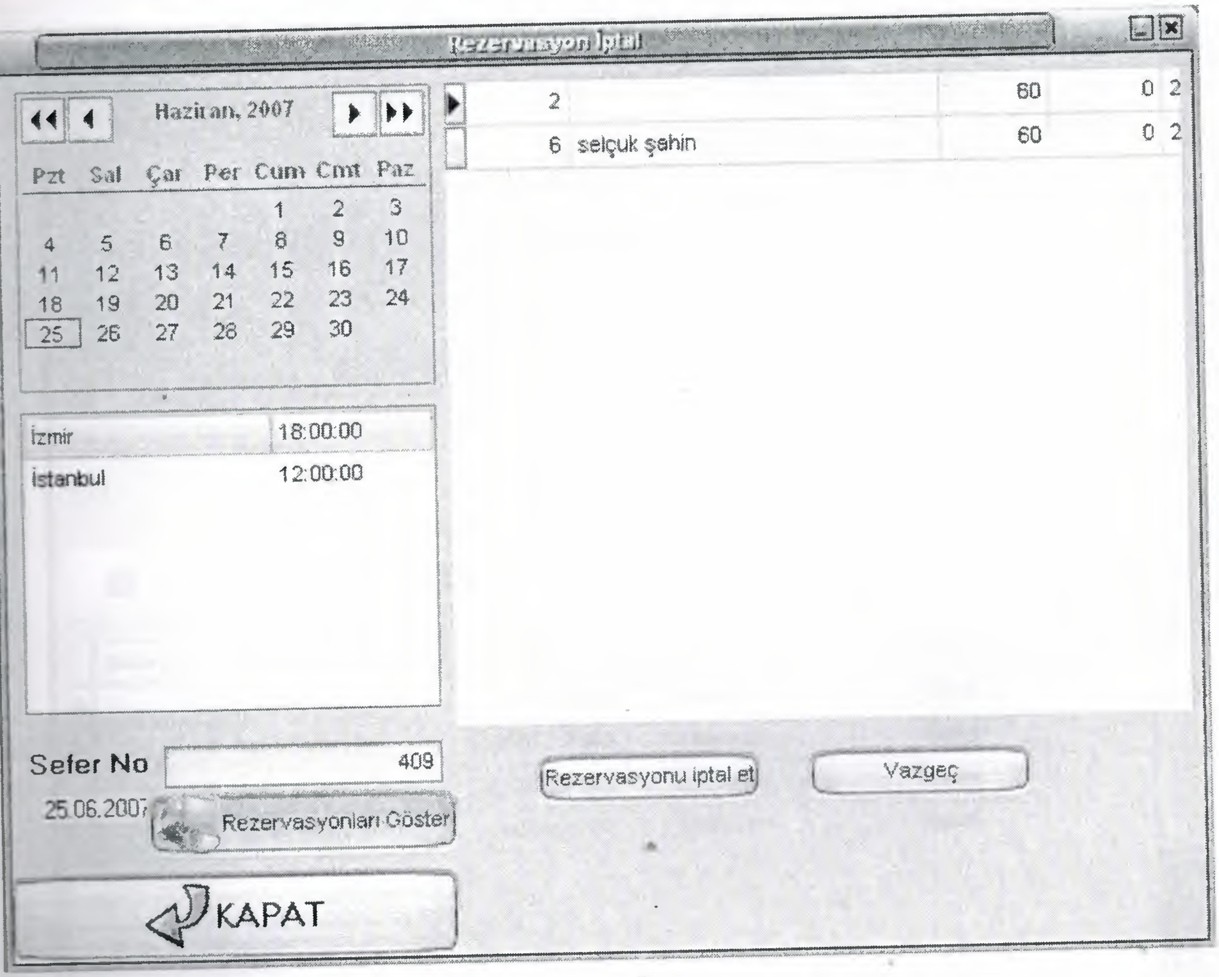
Koltuk No	Adı Soyadı	Fiyat	Satış Tarihi
1	mehmet sait yeni	50	24.06.2007
2	nejla	50	24.06.2007
5	STEPHAN APPİAH	50	24.06.2007

(d) Cancellation of Sold Seat

The cancellation of sold seats form. In this form user double click on the calendar to filter the voyages of the day. After that user select the voyage from the list on the left. And then user click on the "show the sales" button and see the sold seats.

After that user selects the reserved seat and click on the cancel sold seat .

The program codes of this form is in Appendix A - 4



Rezervasyon İptal

Haziran, 2007

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

İzmir	18:00:00
İstanbul	12:00:00

2		60	0	2
6	selçuk şahin	60	0	2

Sefer No 409

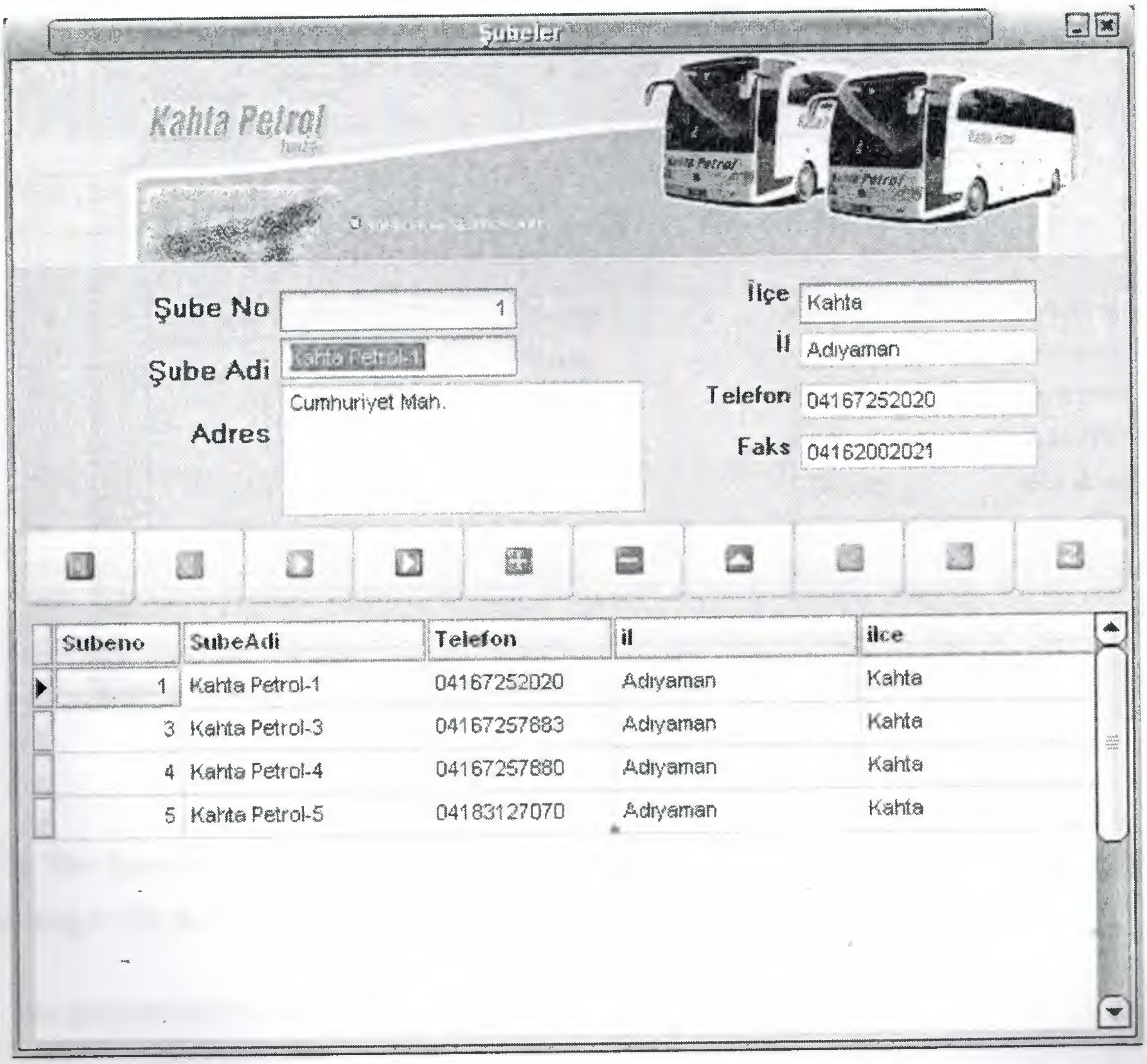
25.06.2007

(e) Cancellation rezervation

The cancellation of rezervation form. In this form user double click on the calendar to filter the voyages of the day. After that user select the voyage from the list on the left. And then user click on the "show the rezervation" button and see the rezervation seats.

After that user selects the reserved seat and click on the cancel rezervation seat .

The program codes of this form is in Appendix A - 5



Şube No: 1

Şube Adi: Kahta Petrol-1

Adres: Cumhuriyet Mah.

İlçe: Kahta

il: Adıyaman

Telefon: 04167252020

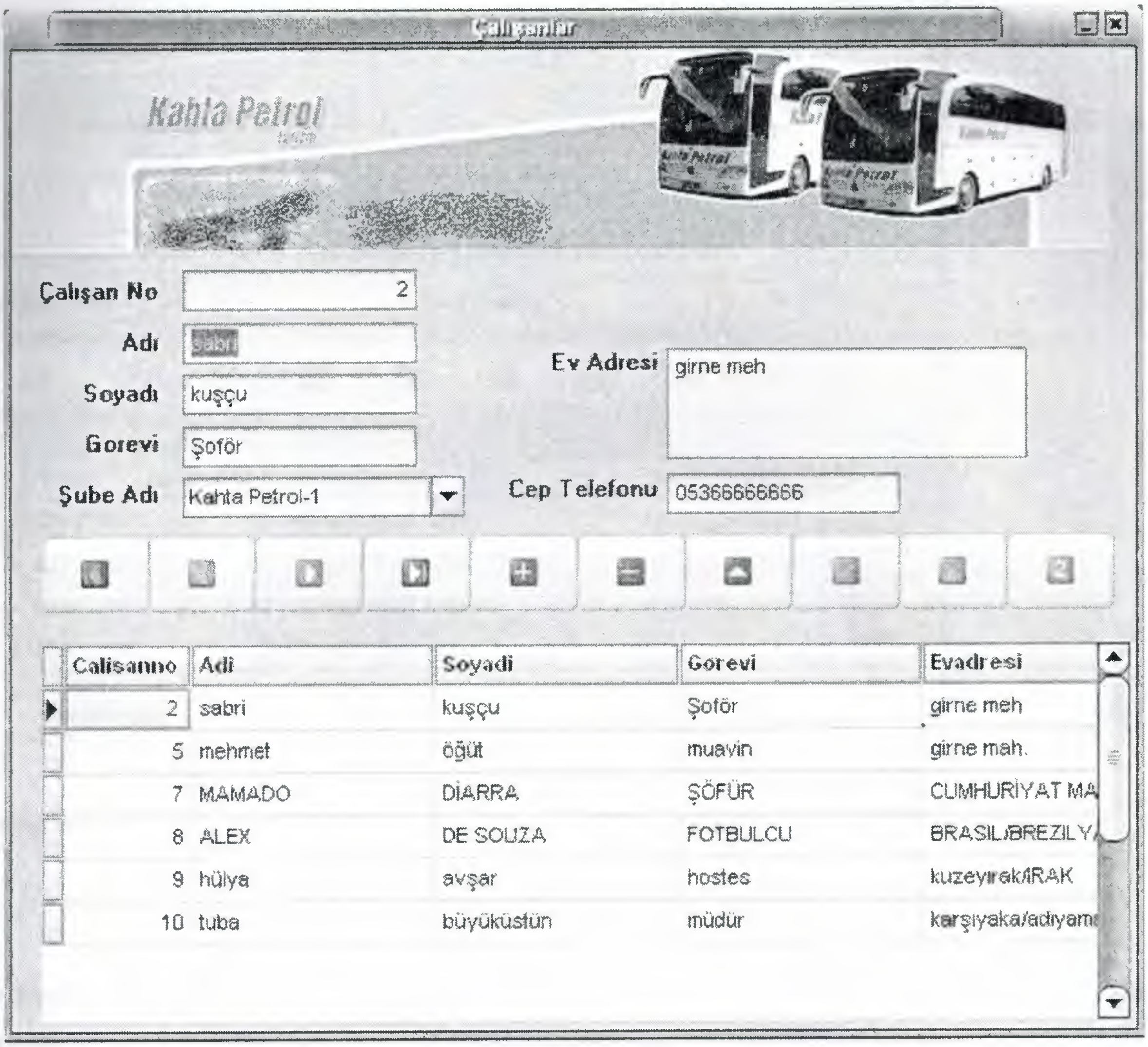
Faks: 04162002021

Subeno	SubeAdi	Telefon	il	ilce
1	Kahta Petrol-1	04167252020	Adıyaman	Kahta
3	Kahta Petrol-3	04167257883	Adıyaman	Kahta
4	Kahta Petrol-4	04167257880	Adıyaman	Kahta
5	Kahta Petrol-5	04183127070	Adıyaman	Kahta

(f) Department

In This form user will enter the related informations and choose the action from the navigator. After recording to file the recorded information can be seen in the grid

The program codes of this form is in Appendix A - 6



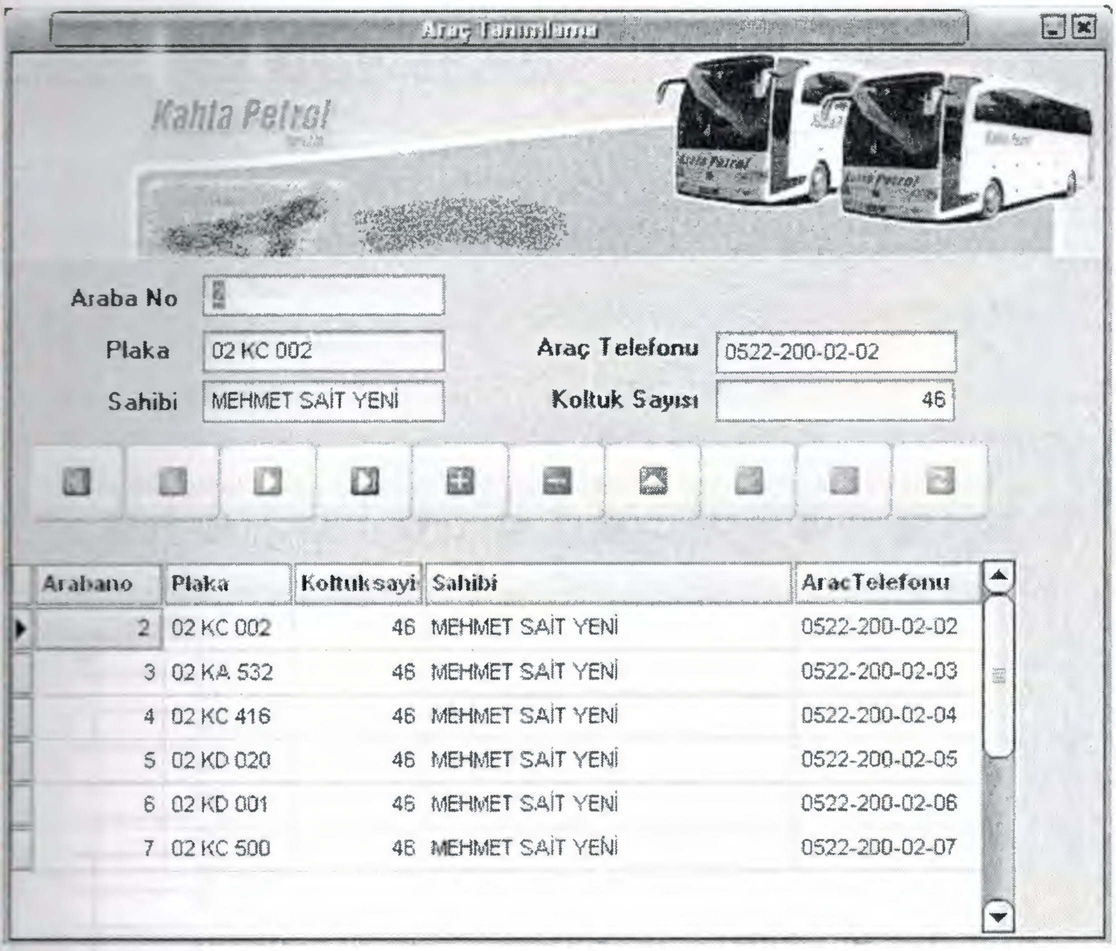
Çalışan No	2
Adı	sabri
Soyadı	kuşçu
Gorevi	Şoför
Şube Adı	Kahta Petrol-1
Ev Adresi	girne meh
Cep Telefonu	05366666666

Calisanno	Adi	Soyadi	Gorevi	Evadresi
2	sabri	kuşçu	Şoför	girne meh
5	mehmet	öğüt	muavin	girne mah.
7	MAMADO	DİARRA	ŞÖFÜR	CUMHURİYAT MA
8	ALEX	DE SOUZA	FOTBULCU	BRASIL/BREZILY
9	hülya	avşar	hostes	kuzeyırak/IRAK
10	tuba	büyüküstün	müdür	karşıyaka/adıyam

(g) Employees

In This form user will enter the related informations and choose the action from the navigator. After recording to file the recorded information can be seen in the grid

The program codes of this form is in Appendix A - 7



Araç Tanımlama

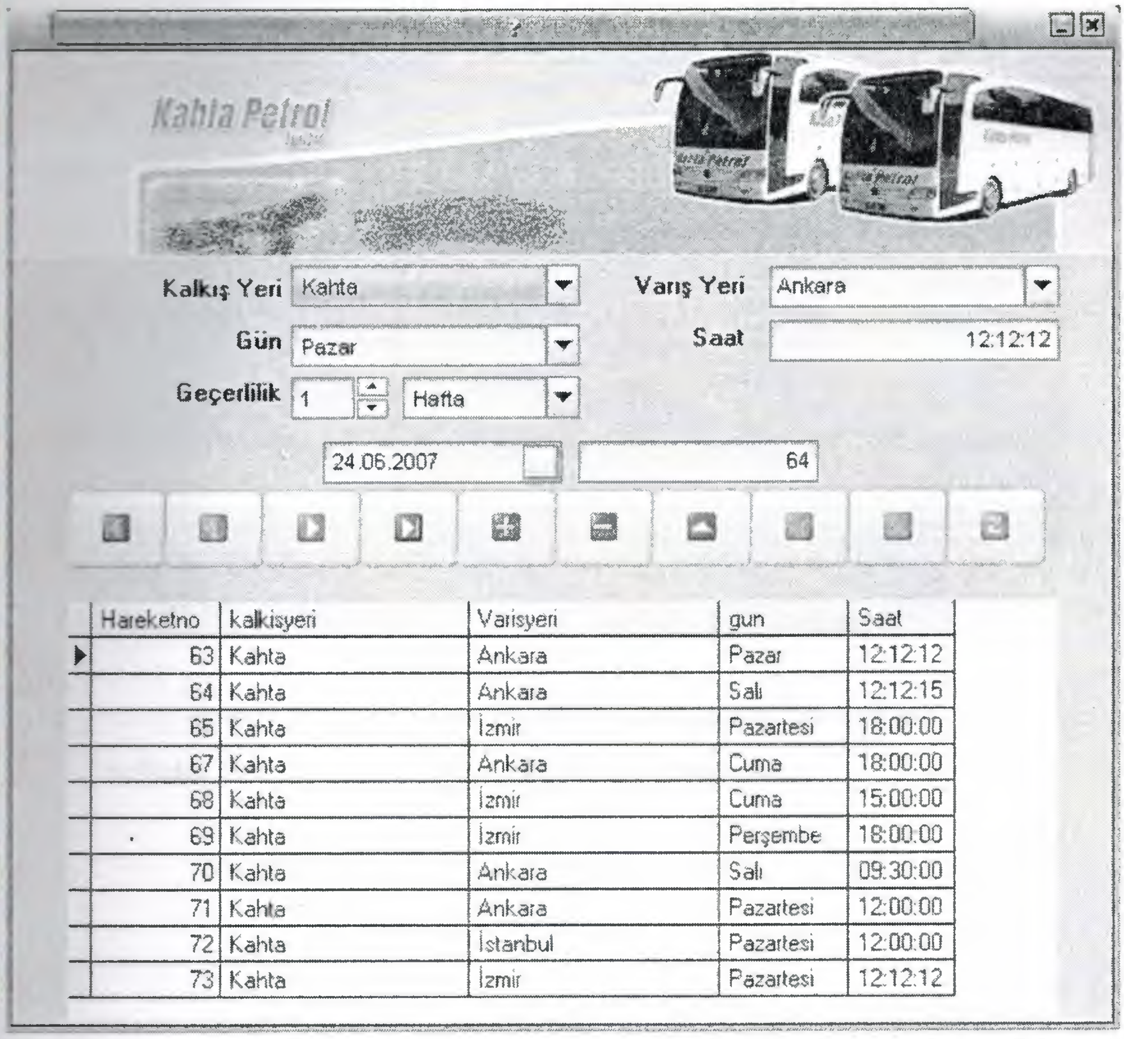
Araba No	2		
Plaka	02 KC 002	Araç Telefonu	0522-200-02-02
Sahibi	MEHMET SAİT YENİ	Koltuk Sayısı	46

Arabano	Plaka	Koltuksayi	Sahibi	AracTelefonu
2	02 KC 002	46	MEHMET SAİT YENİ	0522-200-02-02
3	02 KA 532	46	MEHMET SAİT YENİ	0522-200-02-03
4	02 KC 416	46	MEHMET SAİT YENİ	0522-200-02-04
5	02 KD 020	46	MEHMET SAİT YENİ	0522-200-02-05
6	02 KD 001	46	MEHMET SAİT YENİ	0522-200-02-06
7	02 KC 500	46	MEHMET SAİT YENİ	0522-200-02-07

(h) Vehicles

In This form user will enter the related informations and choose the action from the navigator. After recording to file the recorded information can be seen in the grid

The program codes of this form is in Appendix A - 8



Kalkış Yeri: Kahta

Varış Yeri: Ankara

Gün: Pazar

Saat: 12:12:12

Geçerlilik: 1 Hafta

24.06.2007

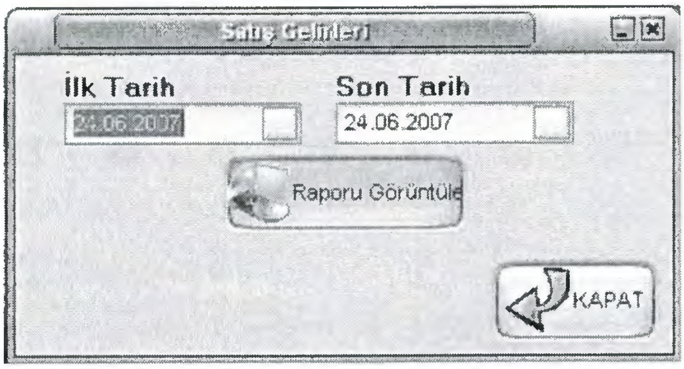
64

	Hareketno	kalkisyeri	Varisyeri	gun	Saat
▶	63	Kahta	Ankara	Pazar	12:12:12
	64	Kahta	Ankara	Salı	12:12:15
	65	Kahta	İzmir	Pazartesi	18:00:00
	67	Kahta	Ankara	Cuma	18:00:00
	68	Kahta	İzmir	Cuma	15:00:00
	69	Kahta	İzmir	Perşembe	18:00:00
	70	Kahta	Ankara	Salı	09:30:00
	71	Kahta	Ankara	Pazartesi	12:00:00
	72	Kahta	İstanbul	Pazartesi	12:00:00
	73	Kahta	İzmir	Pazartesi	12:12:12

(i) Define a Travel

In This form user will enter the related informations, selects the validation criteria and choose the action from the navigator. After recording to file the recorded information can be seen in the grid. The voyages can be stored to file several times as the validation ends.

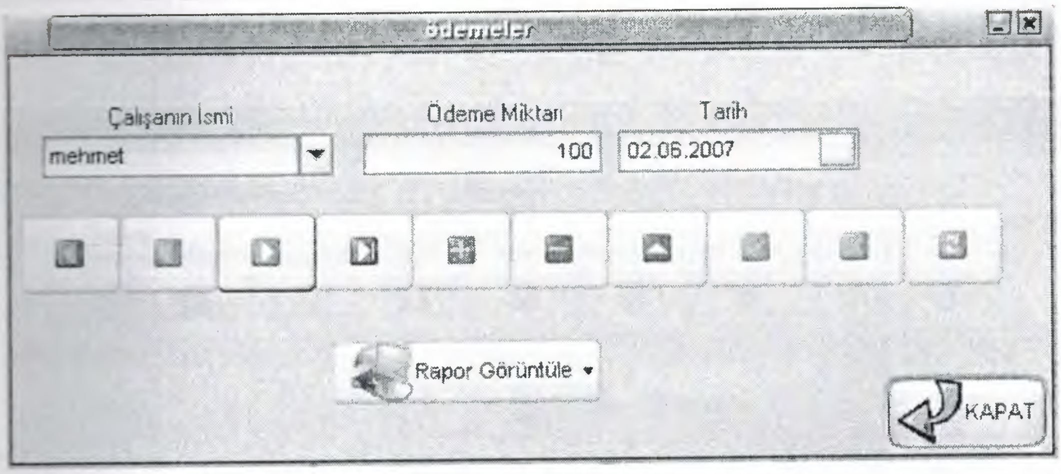
The program codes of this form is in Appendix A - 9



(j) Sale Revenues

In this form user select the first date and the last date and a report automaticaly generated which includes the sales and sale revenues between two dates.

The program codes of this form is in Appendix A - 10



ödemeler

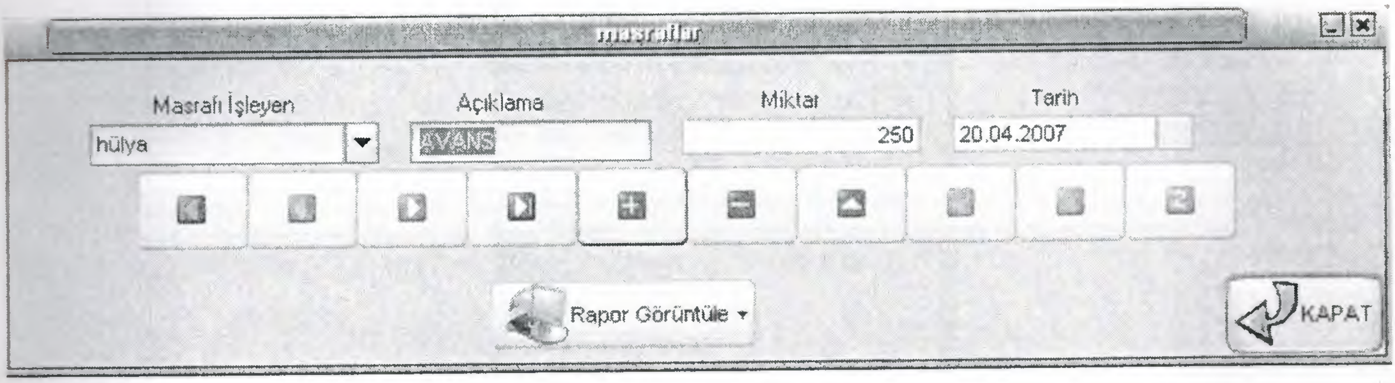
Çalışanın İsmi	Ödeme Miktarı	Tarih
mehmet	100	02.06.2007

(k) Salary Payments

In this form the user selects the employee recorded before enters the salary and then selects the payment date and selects the action from the navigator.

If the user clicks on the show report a report automaticaly generated which includes the salary payments.

The program codes of this form is in Appendix A - 11



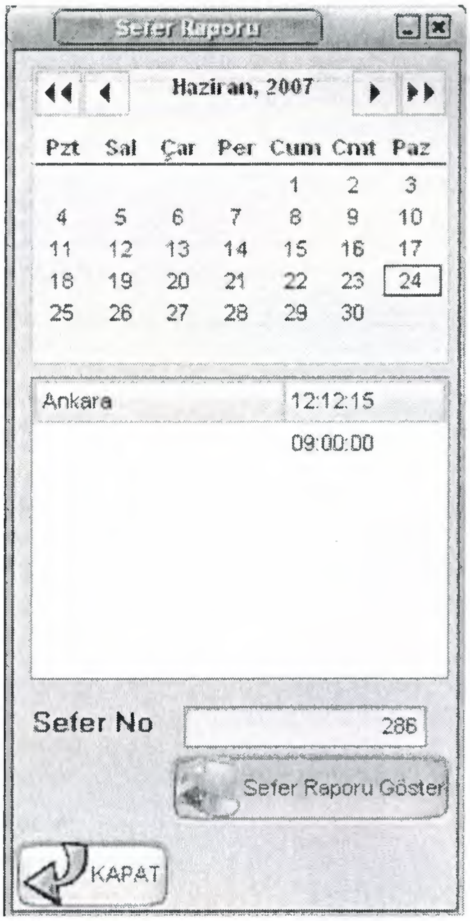
Masrafı İşleyen	Açıklama	Miktar	Tarih
hülya	AVANS	250	20.04.2007

(l) Bill Payments

In this form the user selects the employee recorded before enters the description of the bill, cost and then selects the payment date and selects the action from the navigator.

If the user clicks on the show report a report automaticaly generated which includes the bill payments.

The program codes of this form is in Appendix A - 12



Sefer Raporu

Haziran, 2007

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

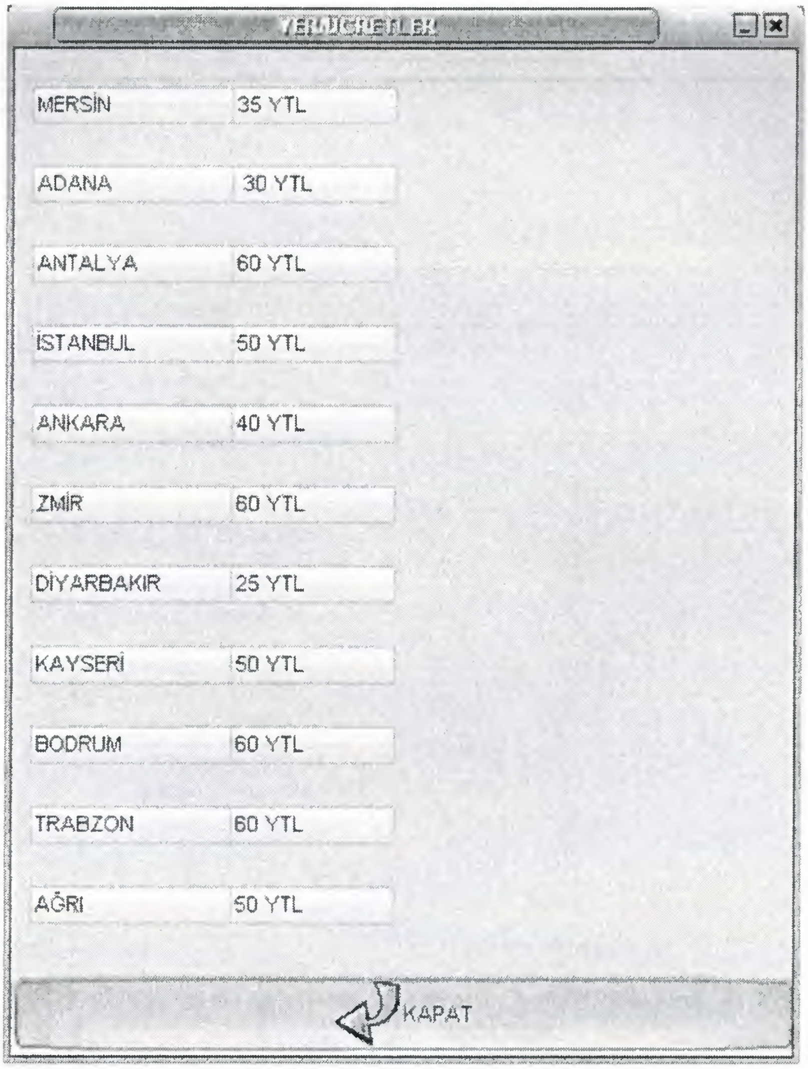
Ankara	12:12:15
	09:00:00

Sefer No: 286

(m)Travel Report

The travel reports form. In this form user double click on the calendar to filter the voyages of the day. After that user select the travel from the list on the left. And then user click on the "show the travel reports" button and see sold seats, and received cargos from the customers.

The program codes of this form is in Appendix A - 13



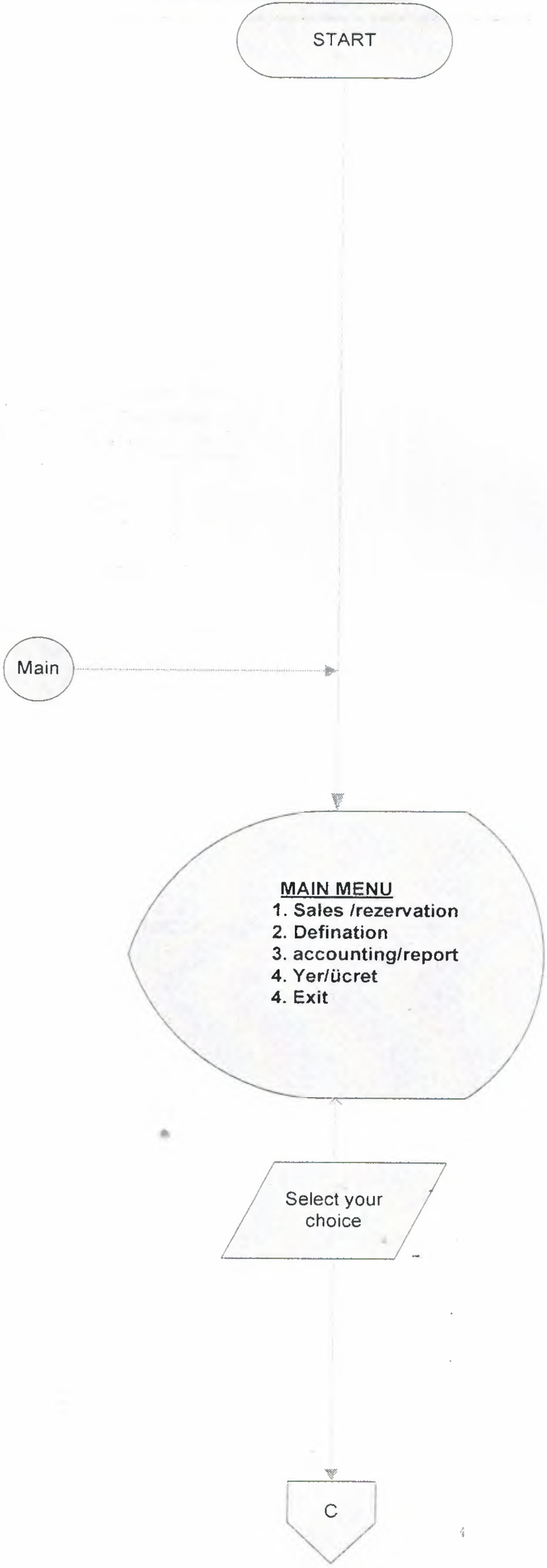
MERSİN	35 YTL
ADANA	30 YTL
ANTALYA	60 YTL
İSTANBUL	50 YTL
ANKARA	40 YTL
ZMİR	60 YTL
DİYARBAKIR	25 YTL
KAYSERİ	50 YTL
BODRUM	60 YTL
TRABZON	60 YTL
AĞRI	50 YTL

(n) help

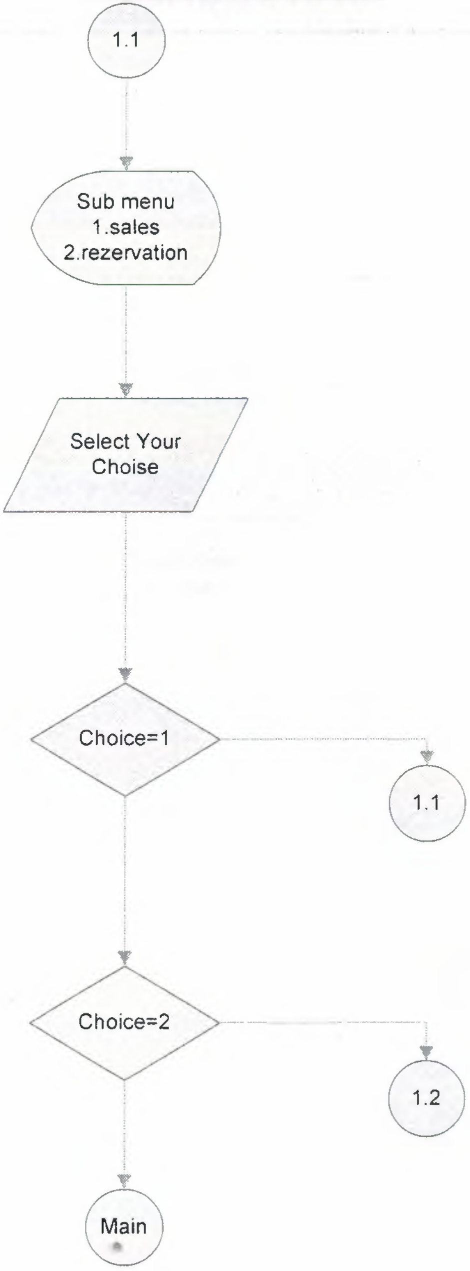
The program codes of this form is in Appendix A - 14

7

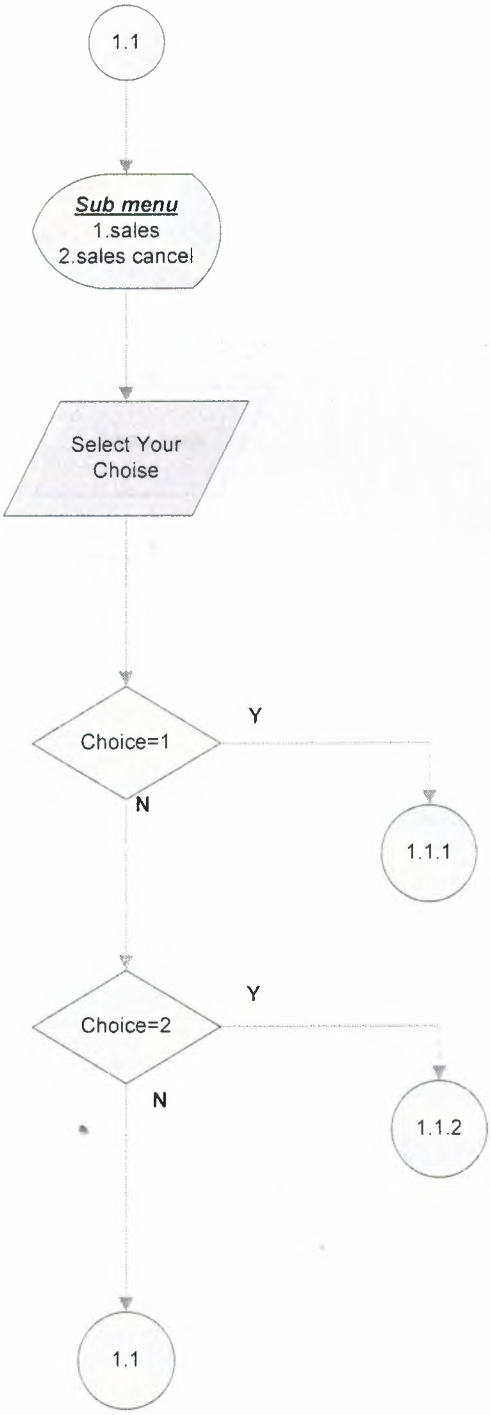
MAIN MENU


START
Main
MAIN MENU
1. Sales /rezervation
2. Defination
3. accounting/report
4. Yer/ücret
4. Exit
Select your choice
C

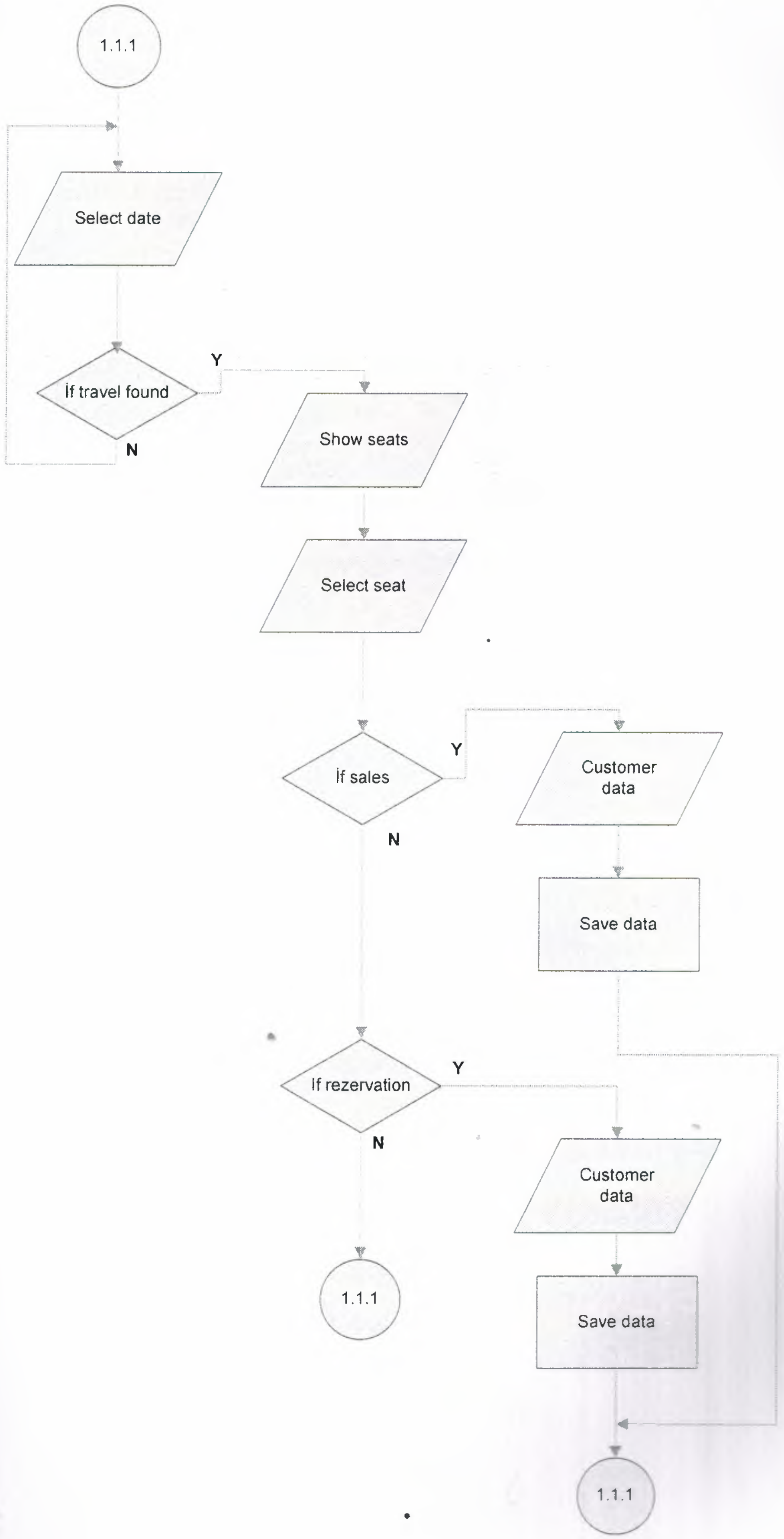
1 .REZERVATION



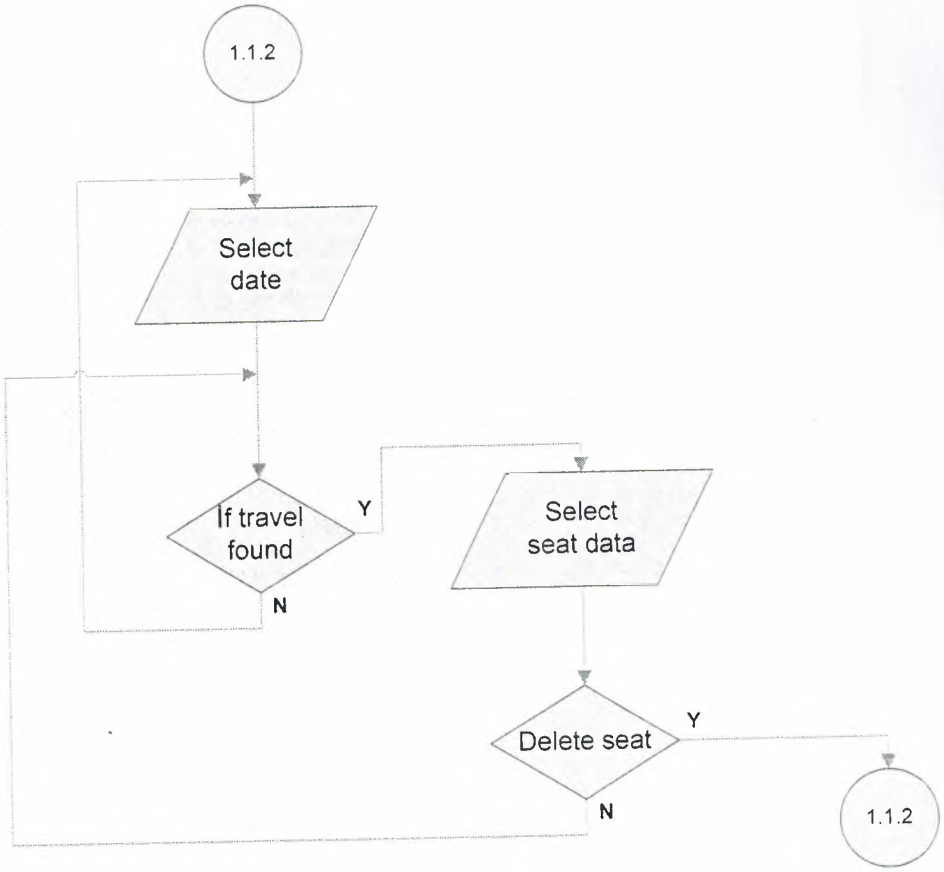
SALES INFORMATION



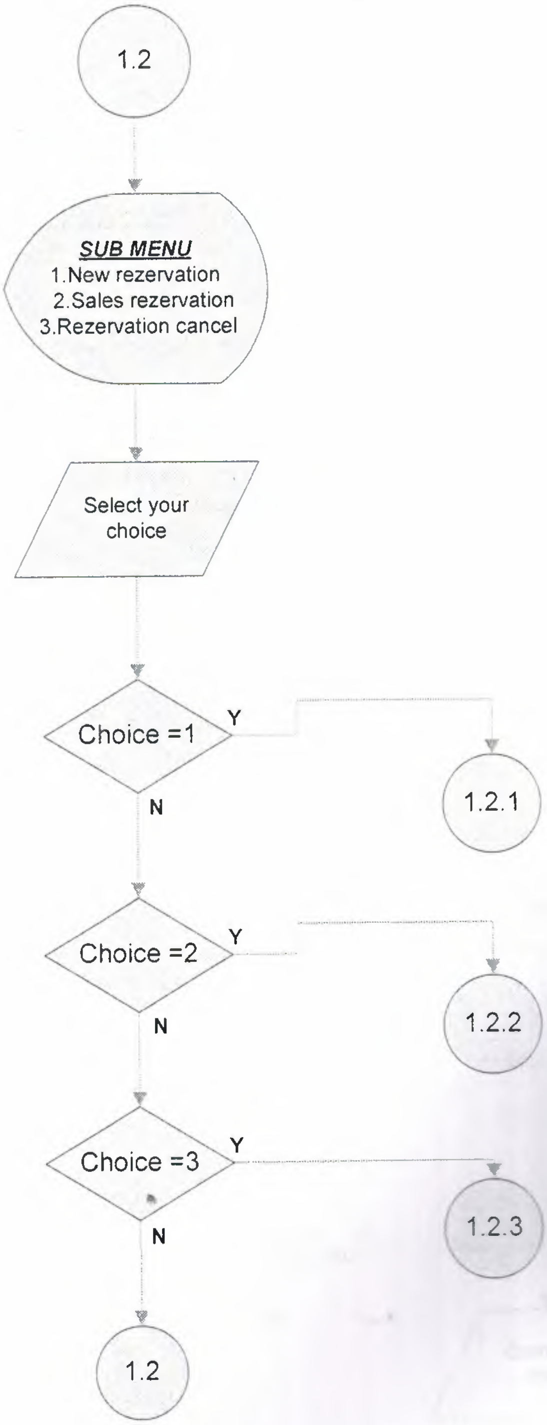
SALES


1.1.1
Select date
İf travel found
Y
N
Show seats
Select seat
İf sales
Y
N
Customer data
Save data
If rezervation
Y
N
Customer data
Save data
1.1.1
1.1.1

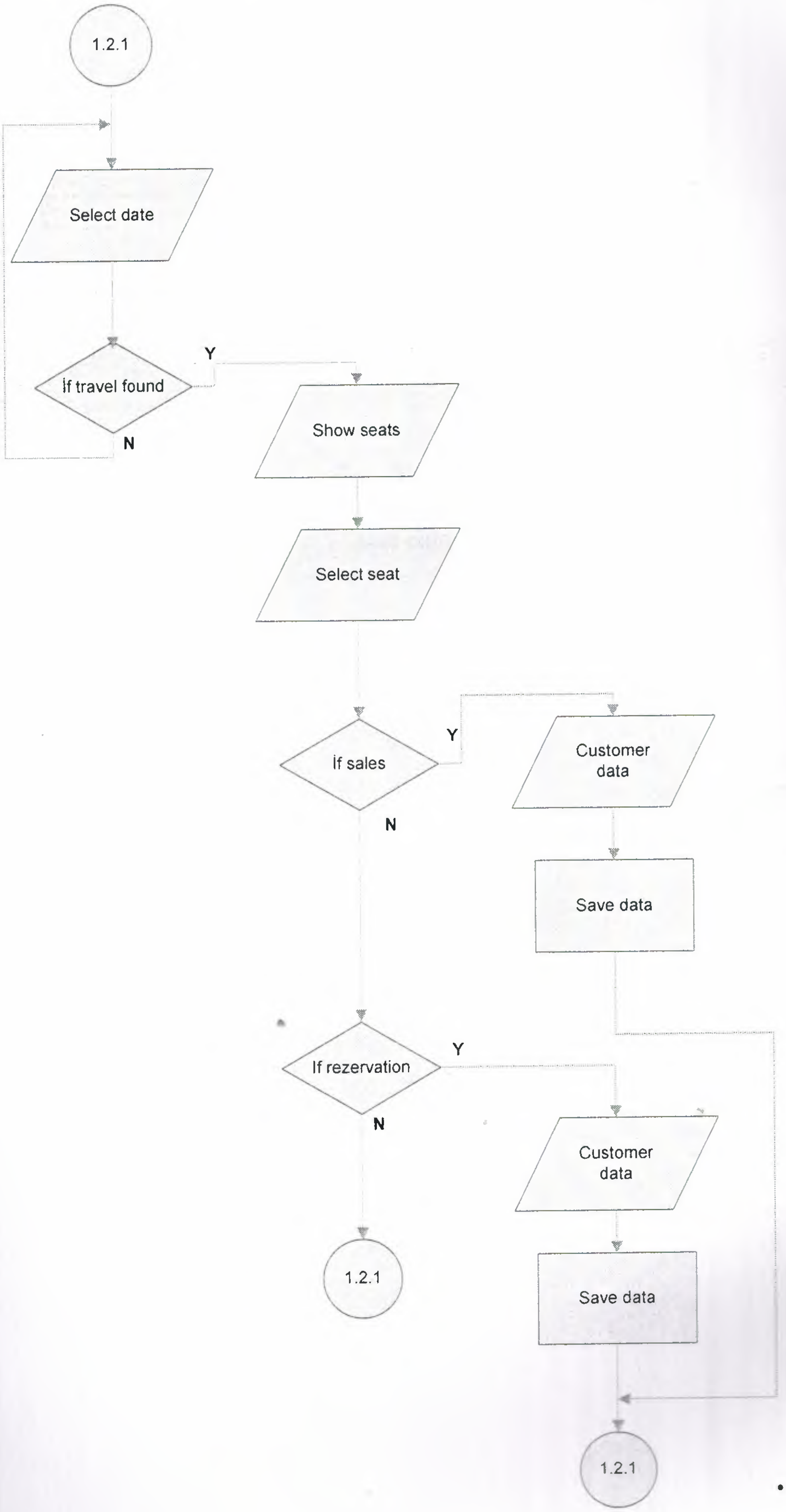
CANCEL SALES



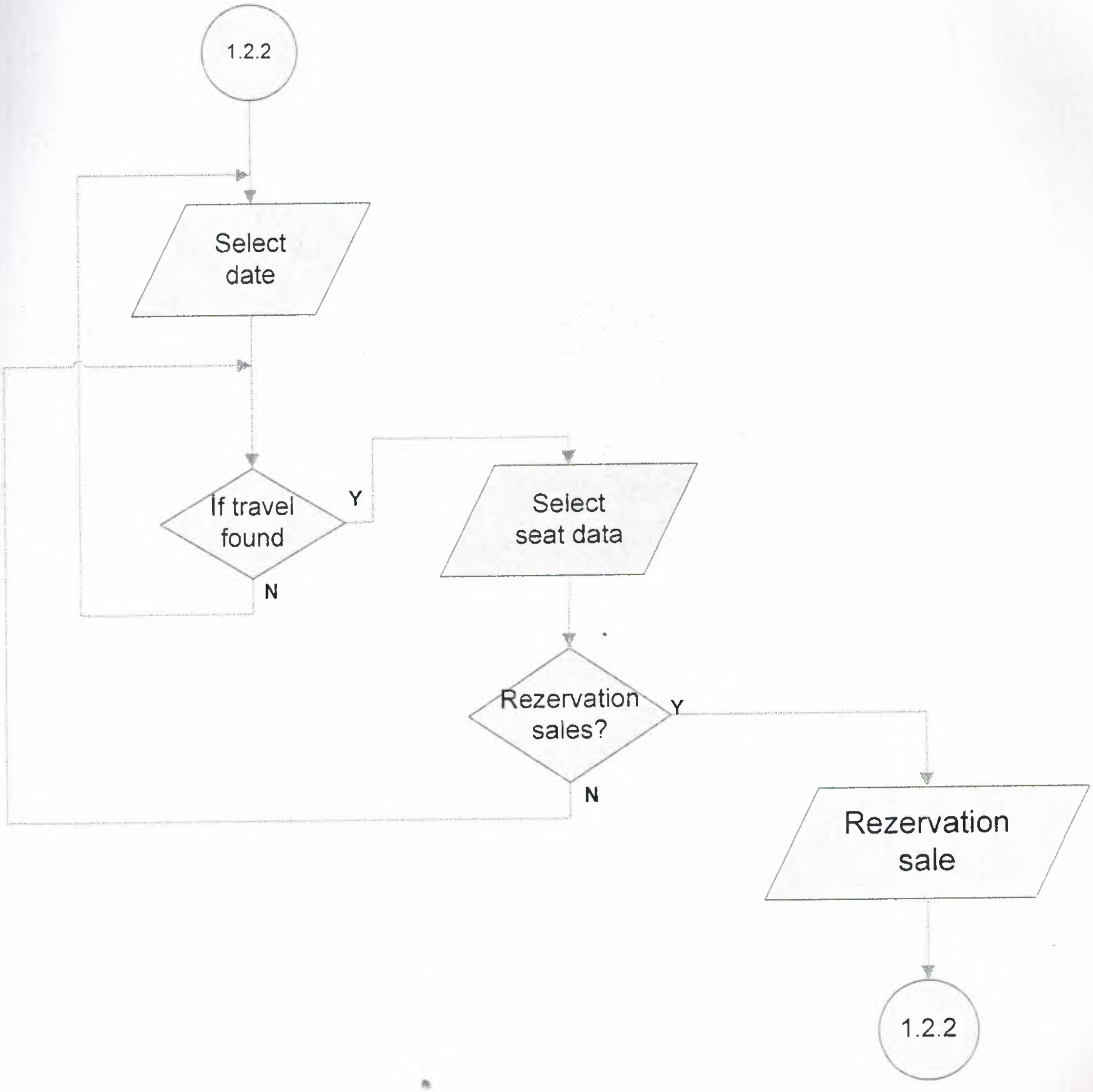
REZERVATİON



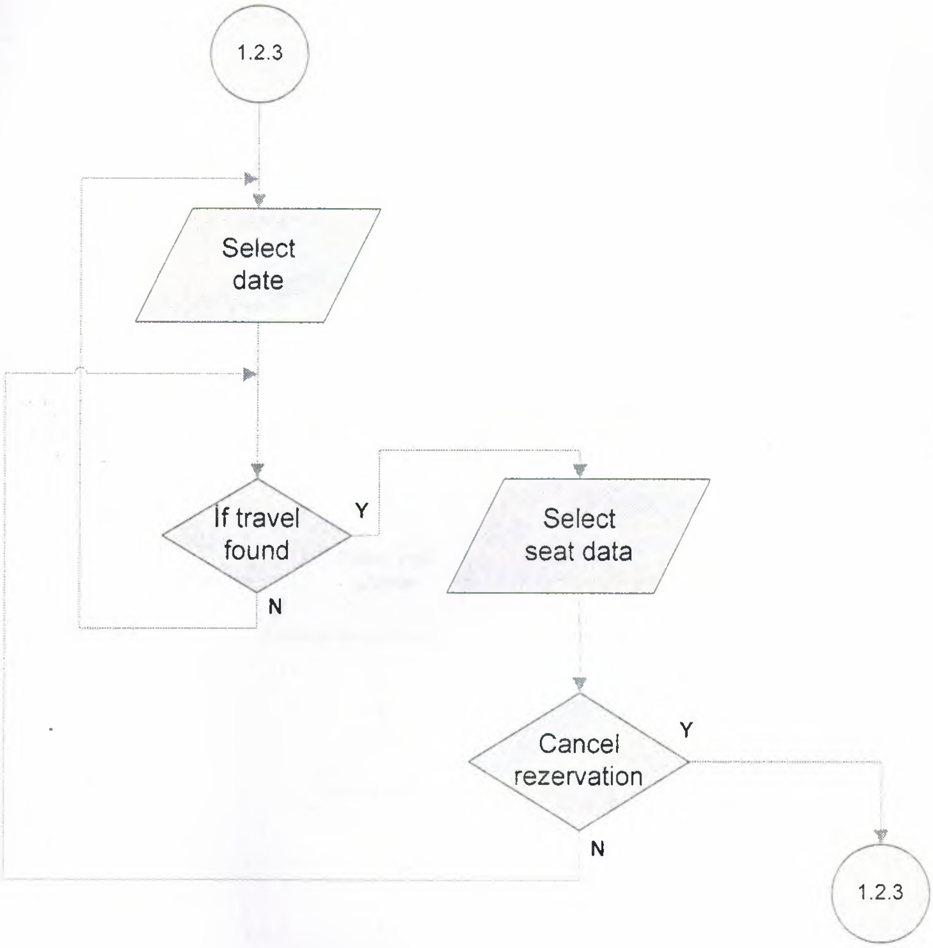
NEW REZERVATION



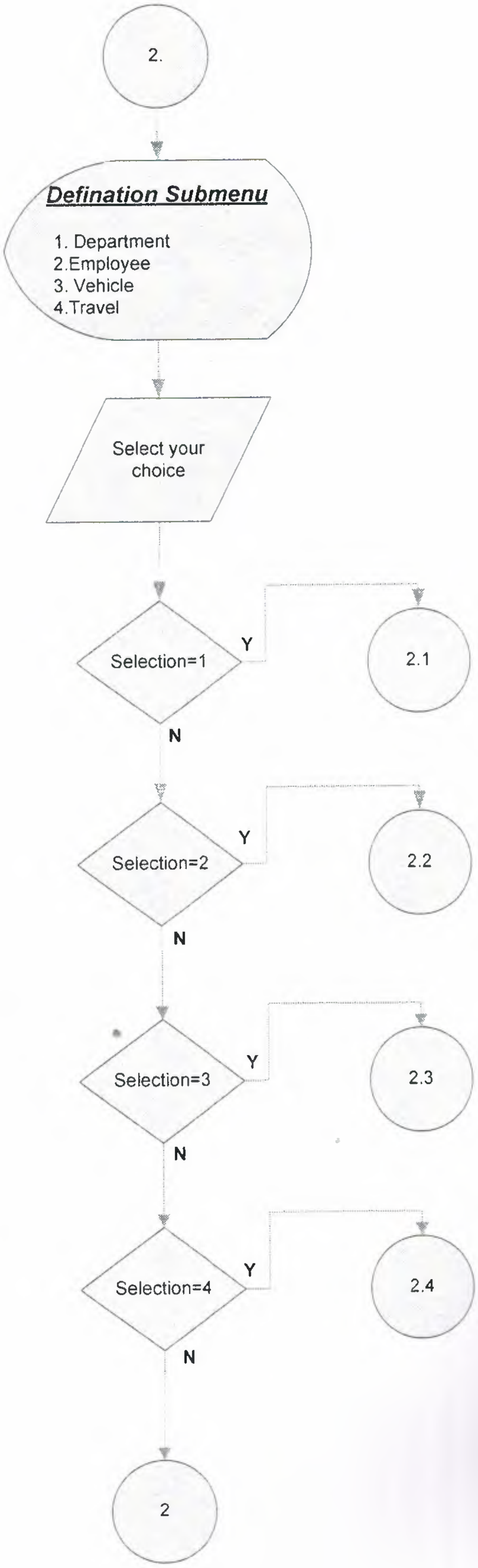
REZERVATİON SALES



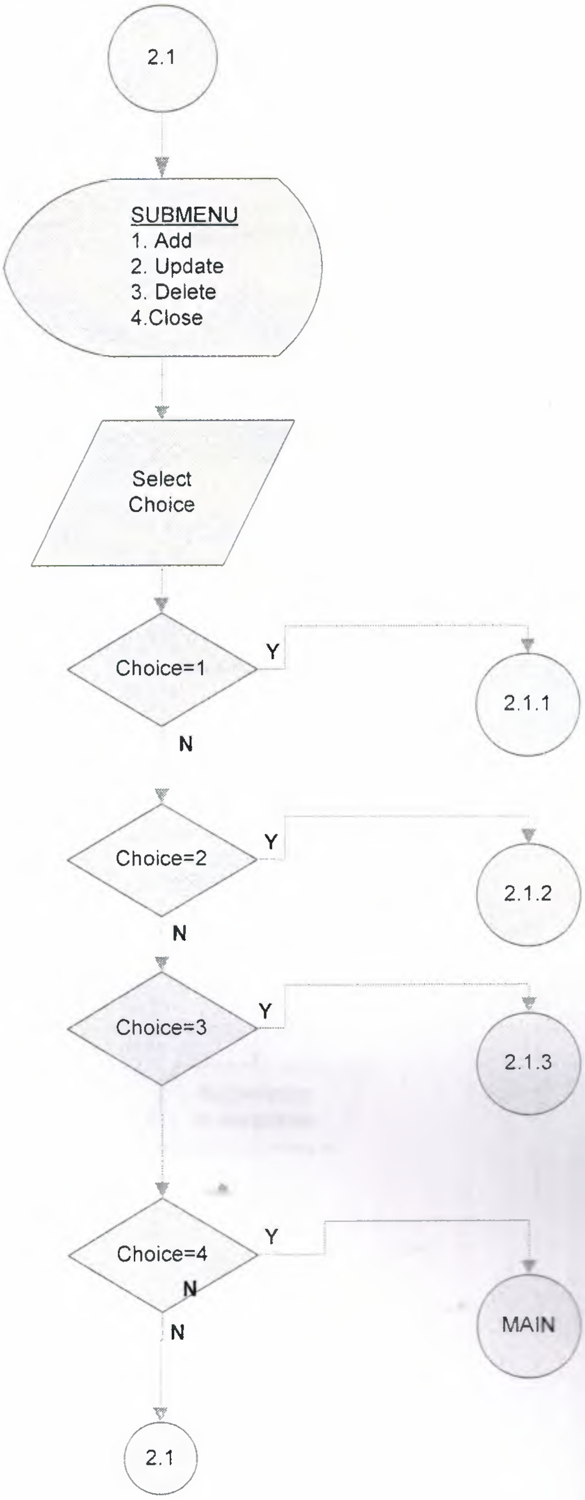
CANCEL REZERVATION



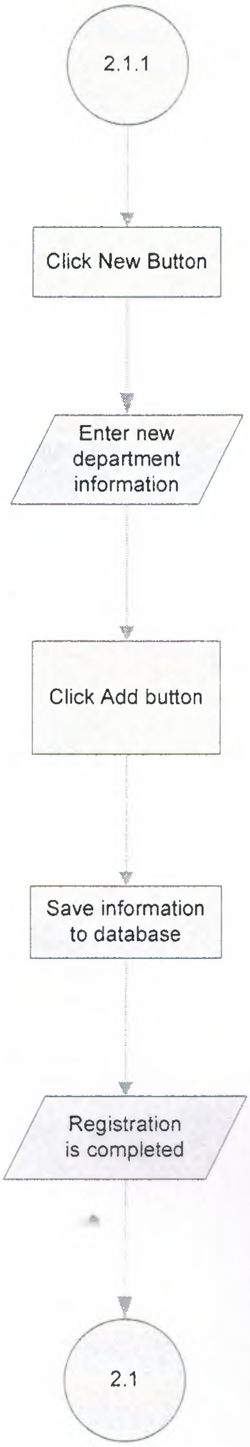
DEFINATION


2.
Defination Submenu
1. Department
2.Employee
3. Vehicle
4.Travel
Select your choice
Selection=1
Y
N
2.1
Selection=2
Y
N
2.2
Selection=3
Y
N
2.3
Selection=4
Y
N
2.4
2

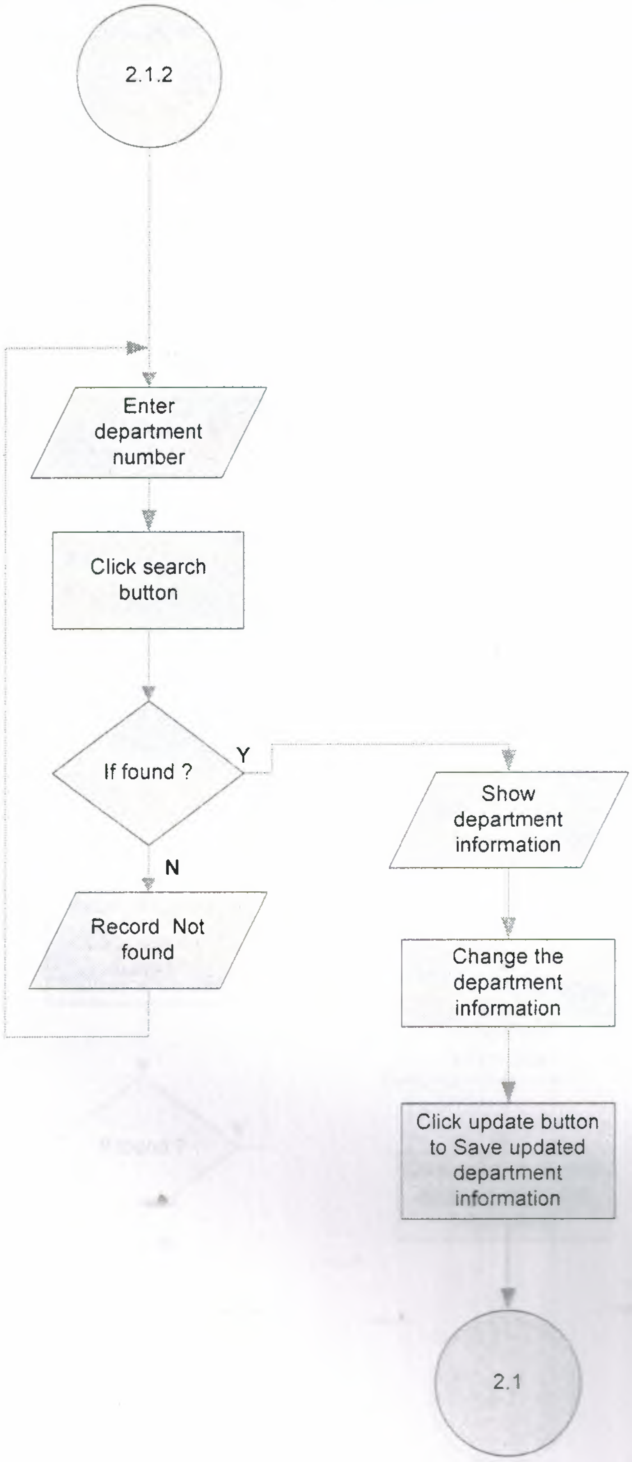
DEPARTMENT


2.1
SUBMENU
1. Add
2. Update
3. Delete
4.Close
Select Choice
Choice=1
Y
2.1.1
N
Choice=2
Y
2.1.2
N
Choice=3
Y
2.1.3
Choice=4
Y
N
N
MAIN
2.1

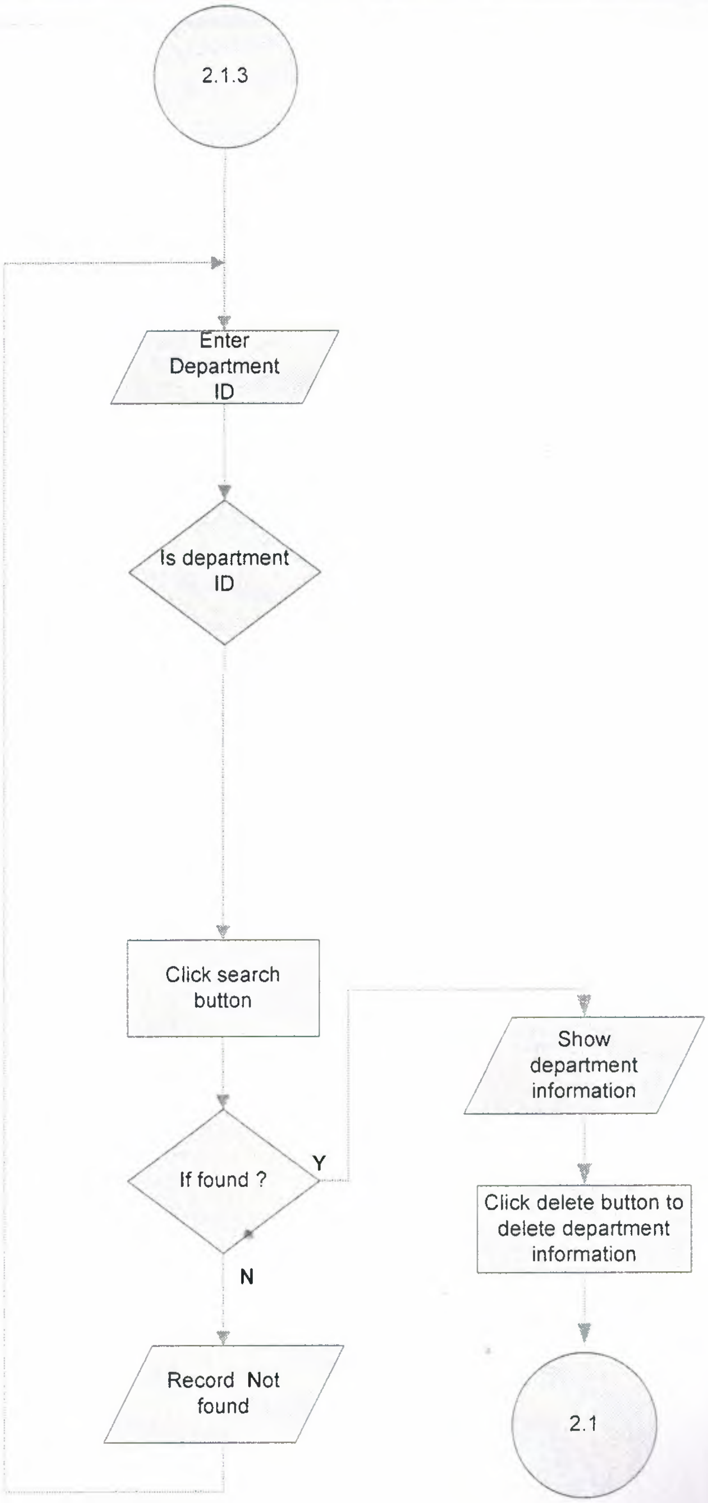
NEW DEPARTMENT INFORMATION


2.1.1
Click New Button
Enter new department information
Click Add button
Save information to database
Registration is completed
2.1

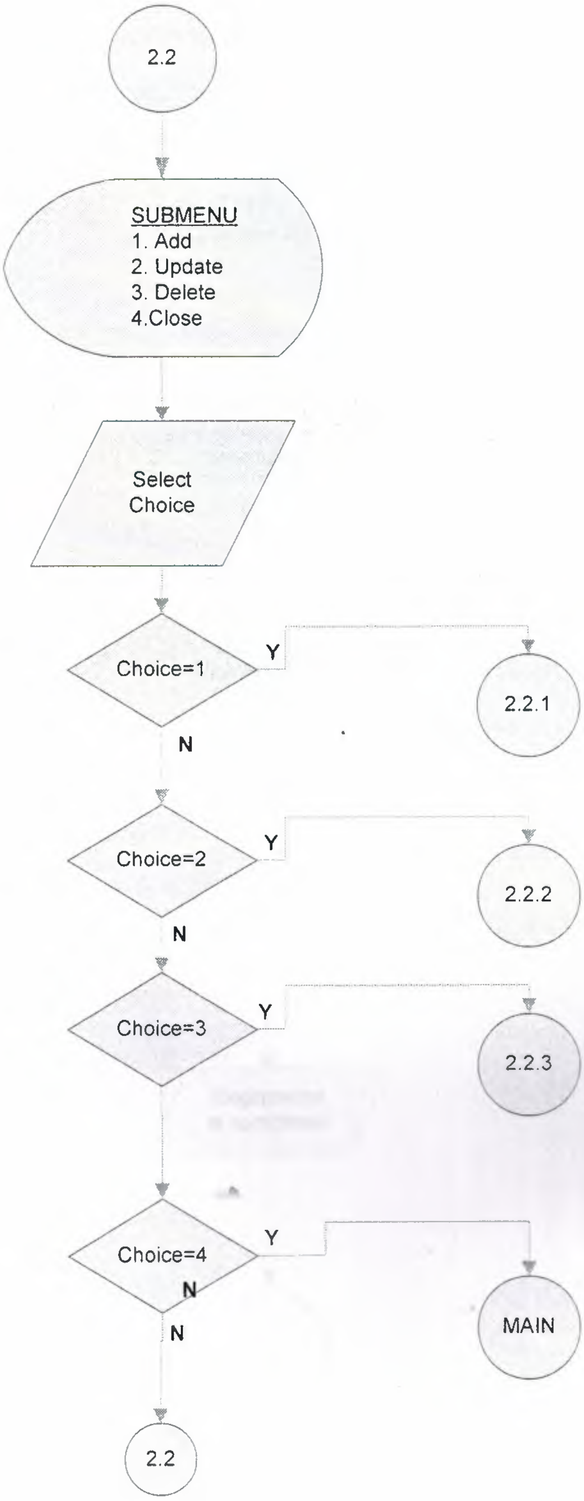
UPDATE DEPARTMENT INFORMATION


2.1.2
Enter department number
Click search button
If found ?
Y
N
Record Not found
Show department information
Change the department information
Click update button to Save updated department information
2.1

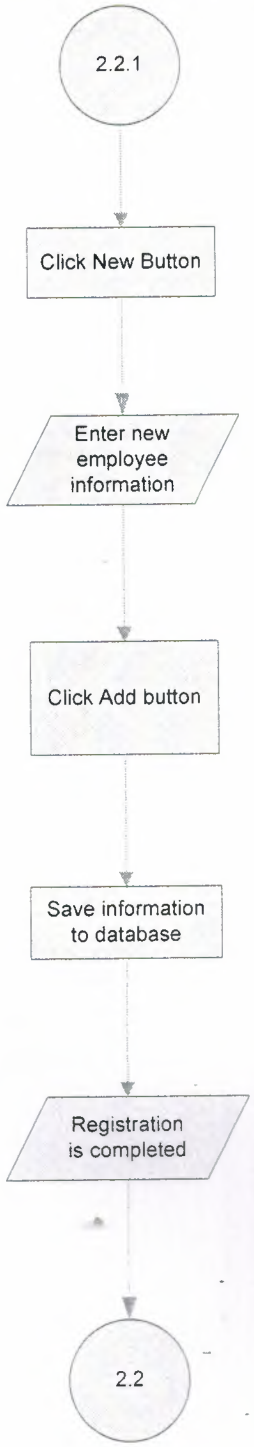
DELETE DEPARTMENT INFORMATION



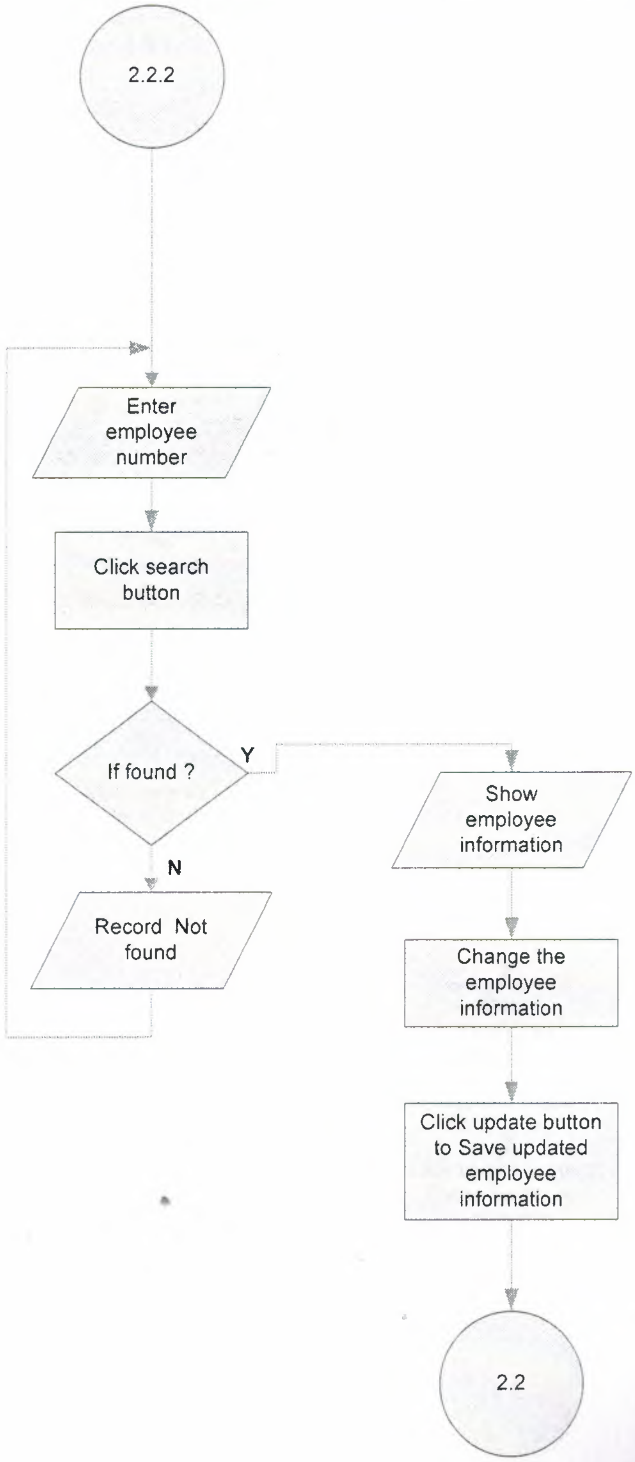
EMPLOYEE


2.2
SUBMENU
1. Add
2. Update
3. Delete
4.Close
Select Choice
Choice=1
Y
2.2.1
N
Choice=2
Y
2.2.2
N
Choice=3
Y
2.2.3
Choice=4
Y
MAIN
N
N
2.2

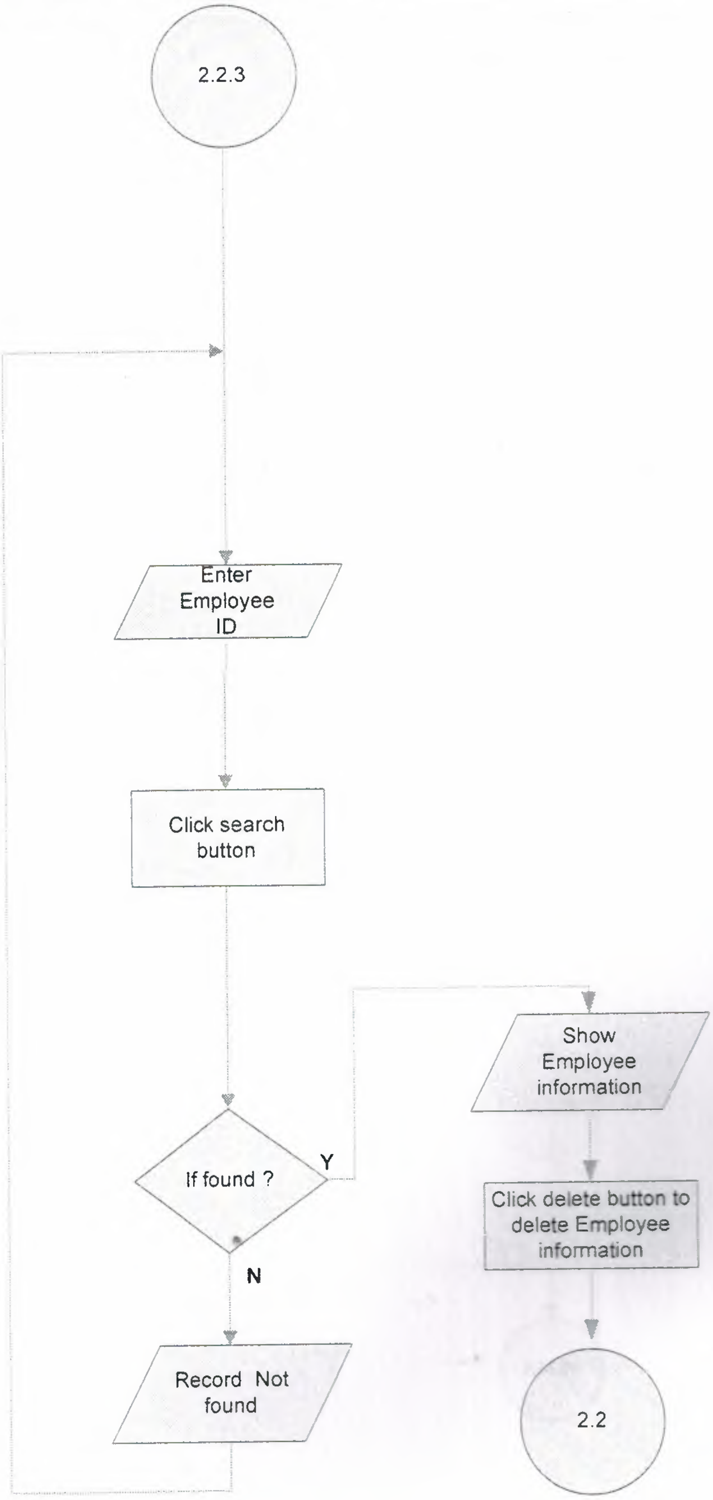
NEW EMPLOYEE INFORMATION


2.2.1
Click New Button
Enter new employee information
Click Add button
Save information to database
Registration is completed
2.2

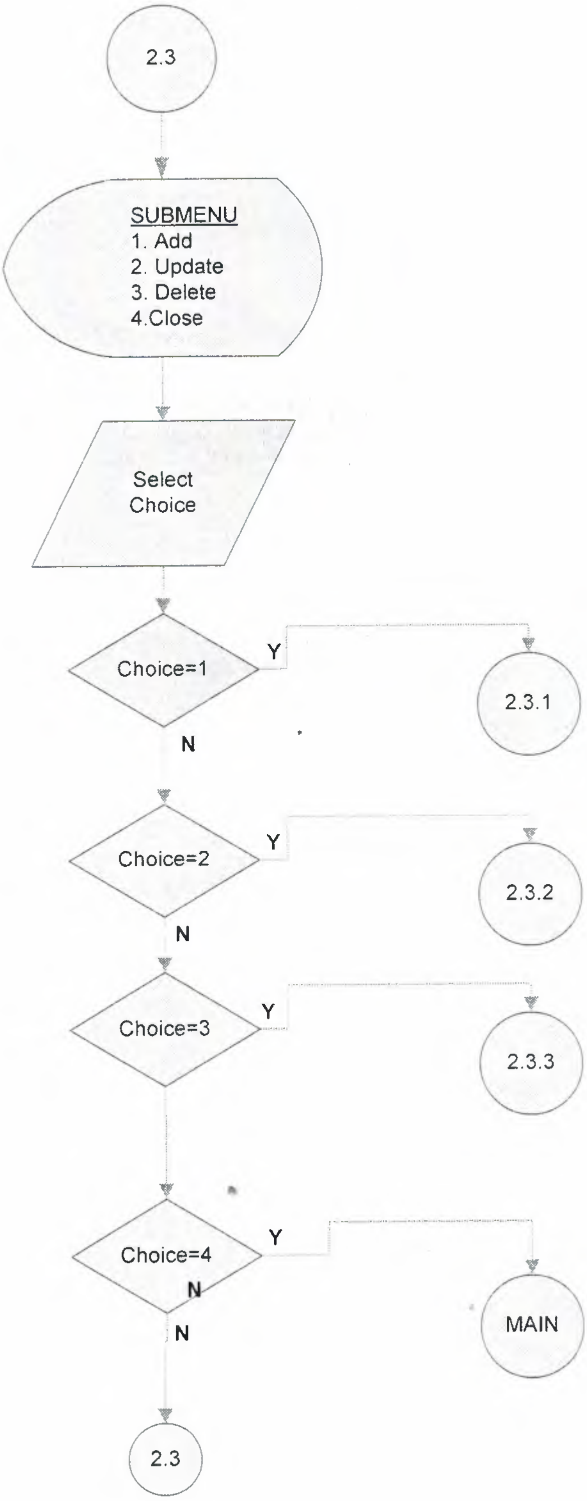
UPDATE EMPLOYEE INFORMATION


2.2.2
Enter employee number
Click search button
If found ?
Y
N
Record Not found
Show employee information
Change the employee information
Click update button to Save updated employee information
2.2

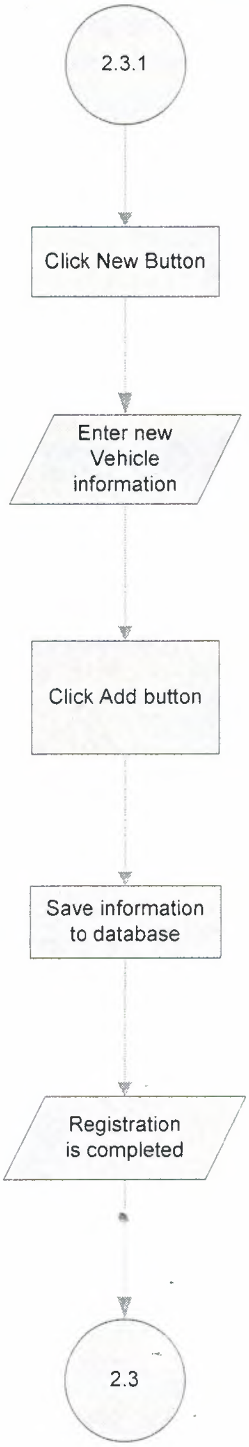
DELETE EMPLOYEE INFORMATION


2.2.3
Enter Employee ID
Click search button
If found ?
Y
N
Show Employee information
Click delete button to delete Employee information
Record Not found
2.2

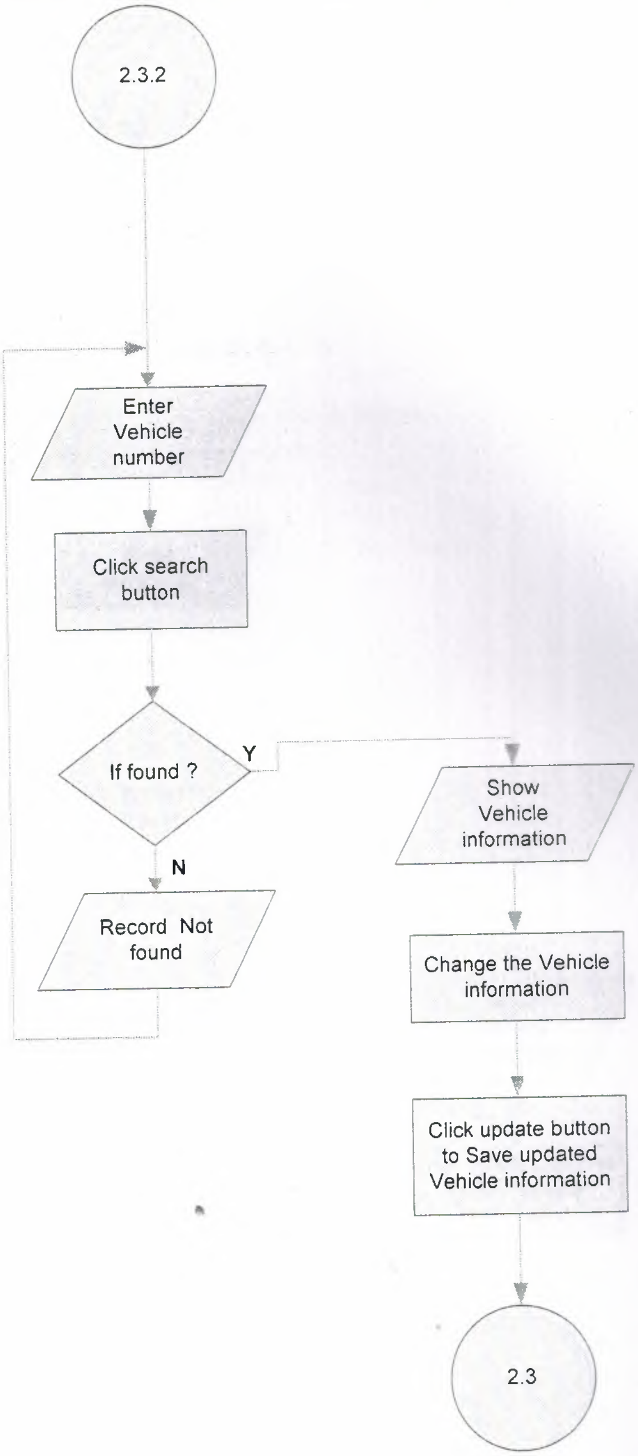
VEHICLE


2.3
SUBMENU
1. Add
2. Update
3. Delete
4.Close
Select Choice
Choice=1
Y
2.3.1
N
Choice=2
Y
2.3.2
N
Choice=3
Y
2.3.3
Choice=4
Y
MAIN
N
N
2.3

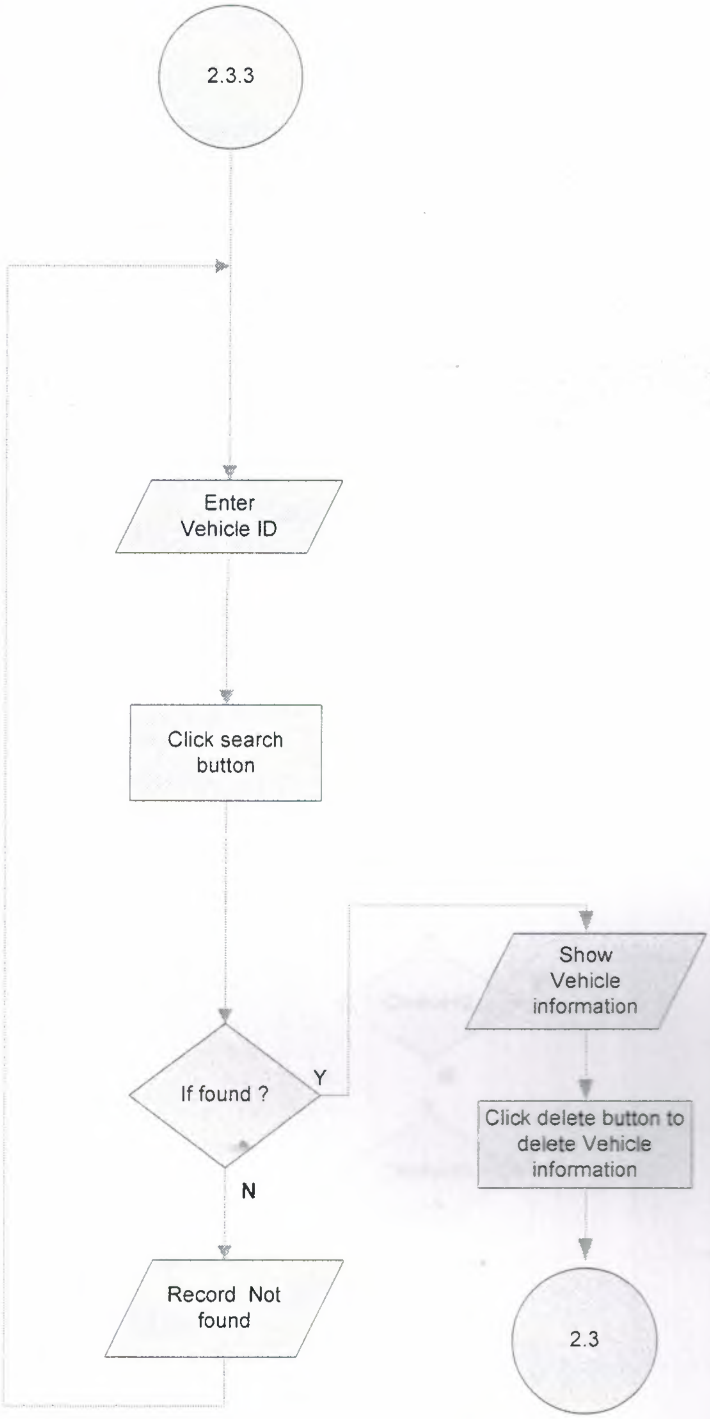
NEW VEHICLE INFORMATION


2.3.1
Click New Button
Enter new Vehicle information
Click Add button
Save information to database
Registration is completed
2.3

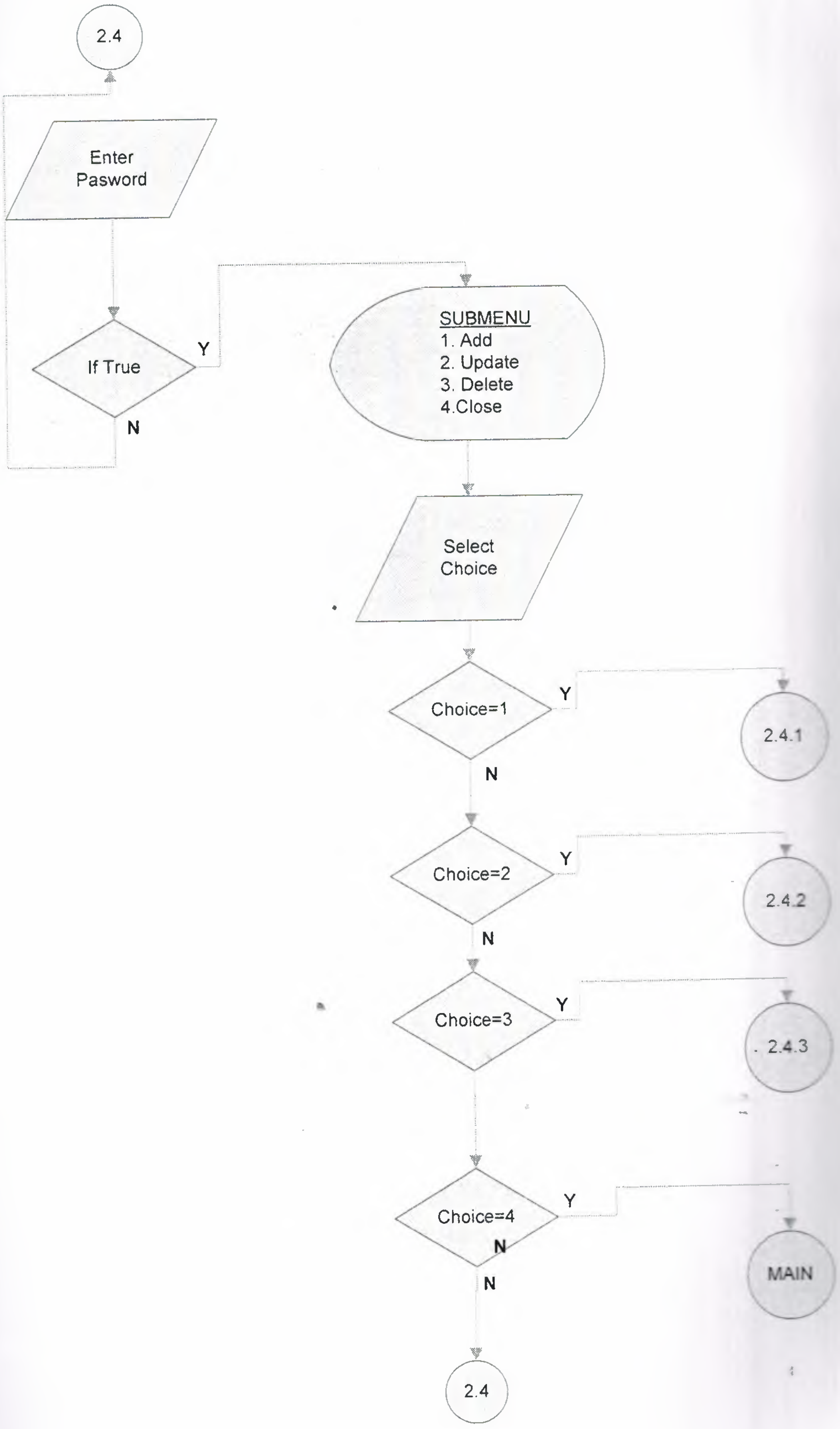
UPDATE VEHICLE INFORMATION



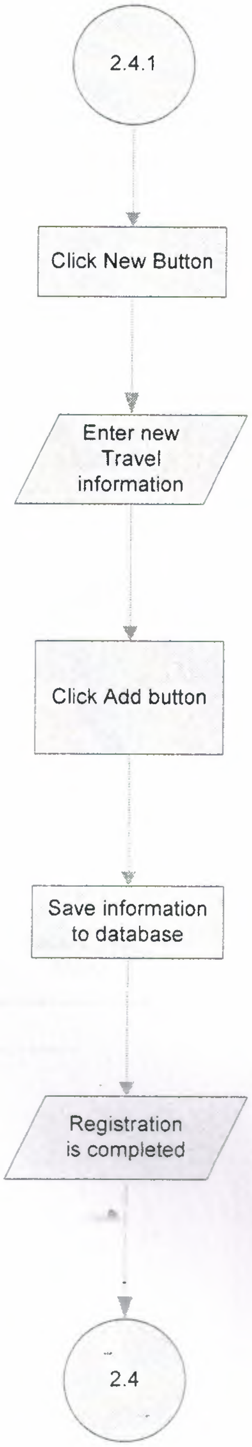
DELETE VEHICLE INFORMATION



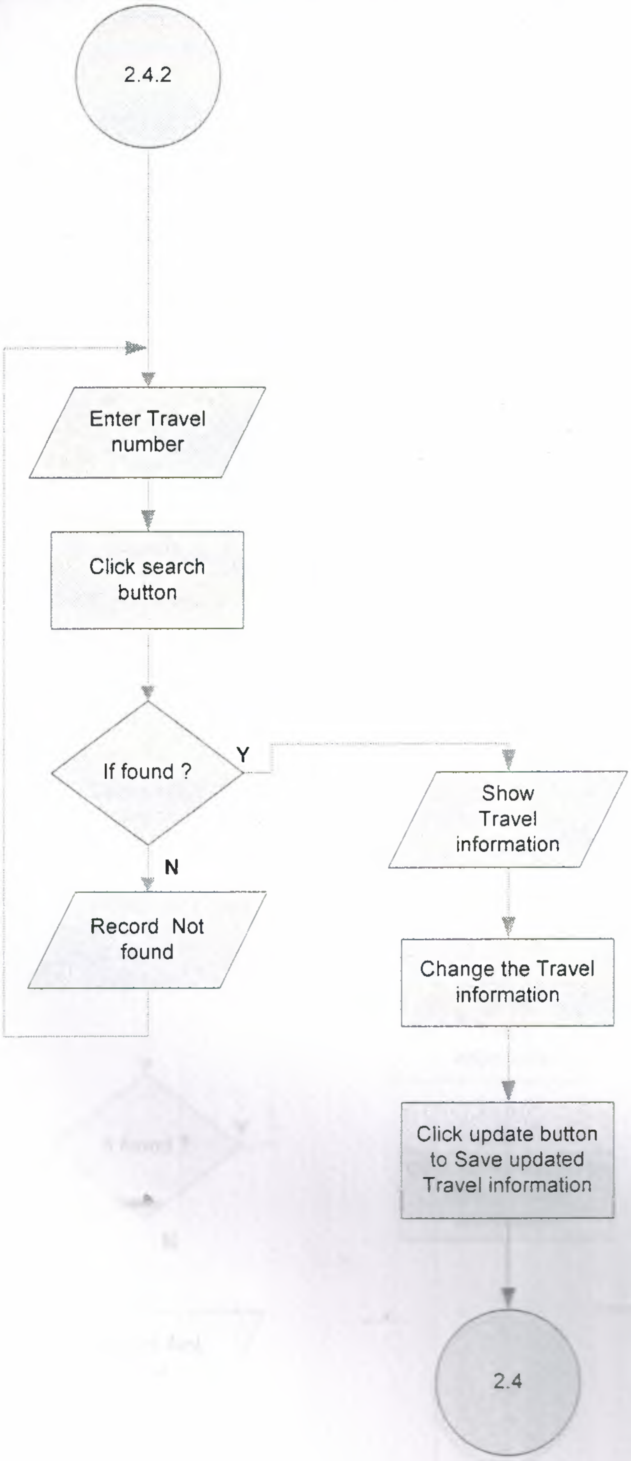
TRAVEL


2.4
Enter
Pasword
If True
Y
N
SUBMENU
1. Add
2. Update
3. Delete
4.Close
Select
Choice
Choice=1
Y
N
2.4.1
Choice=2
Y
N
2.4.2
Choice=3
Y
2.4.3
Choice=4
Y
N
N
MAIN
2.4

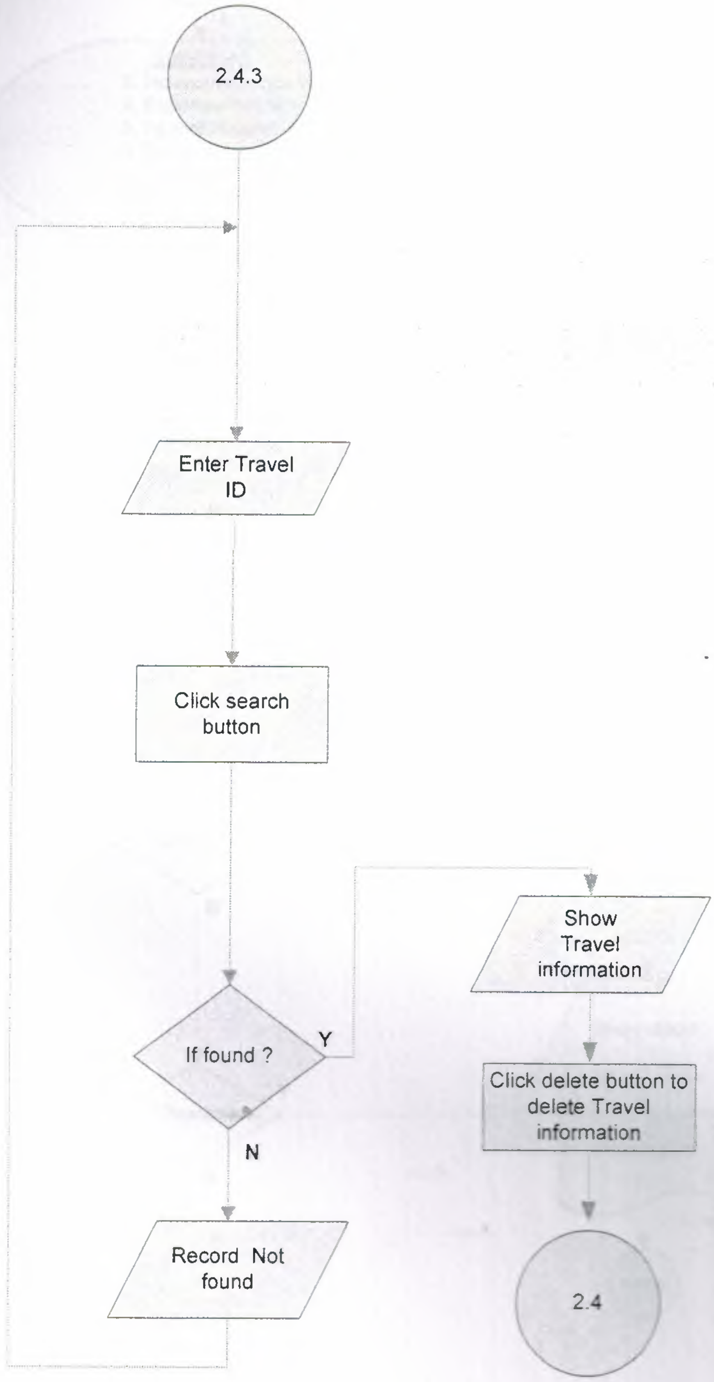
NEW TRAVEL INFORMATION


2.4.1
Click New Button
Enter new Travel information
Click Add button
Save information to database
Registration is completed
2.4

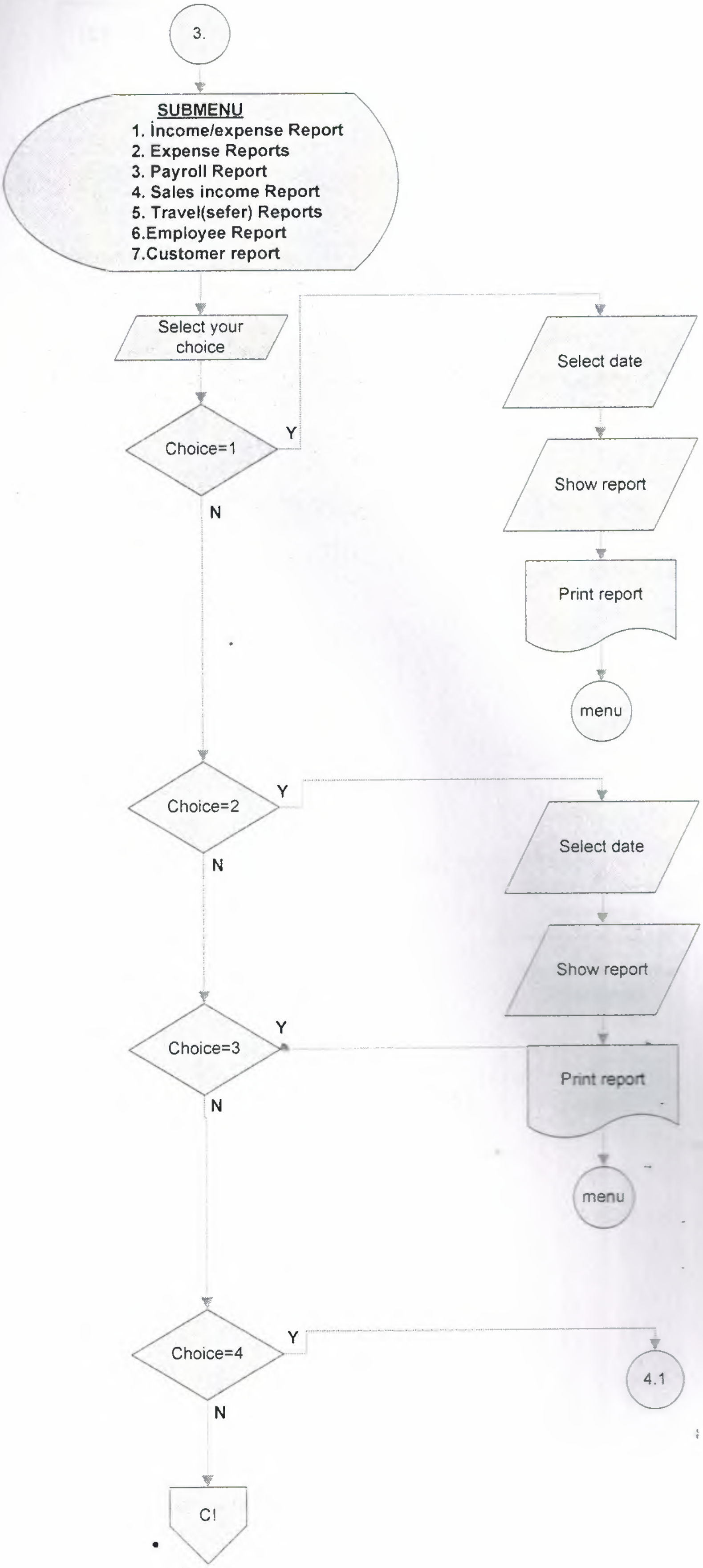
UPDATE TRAVEL INFORMATION

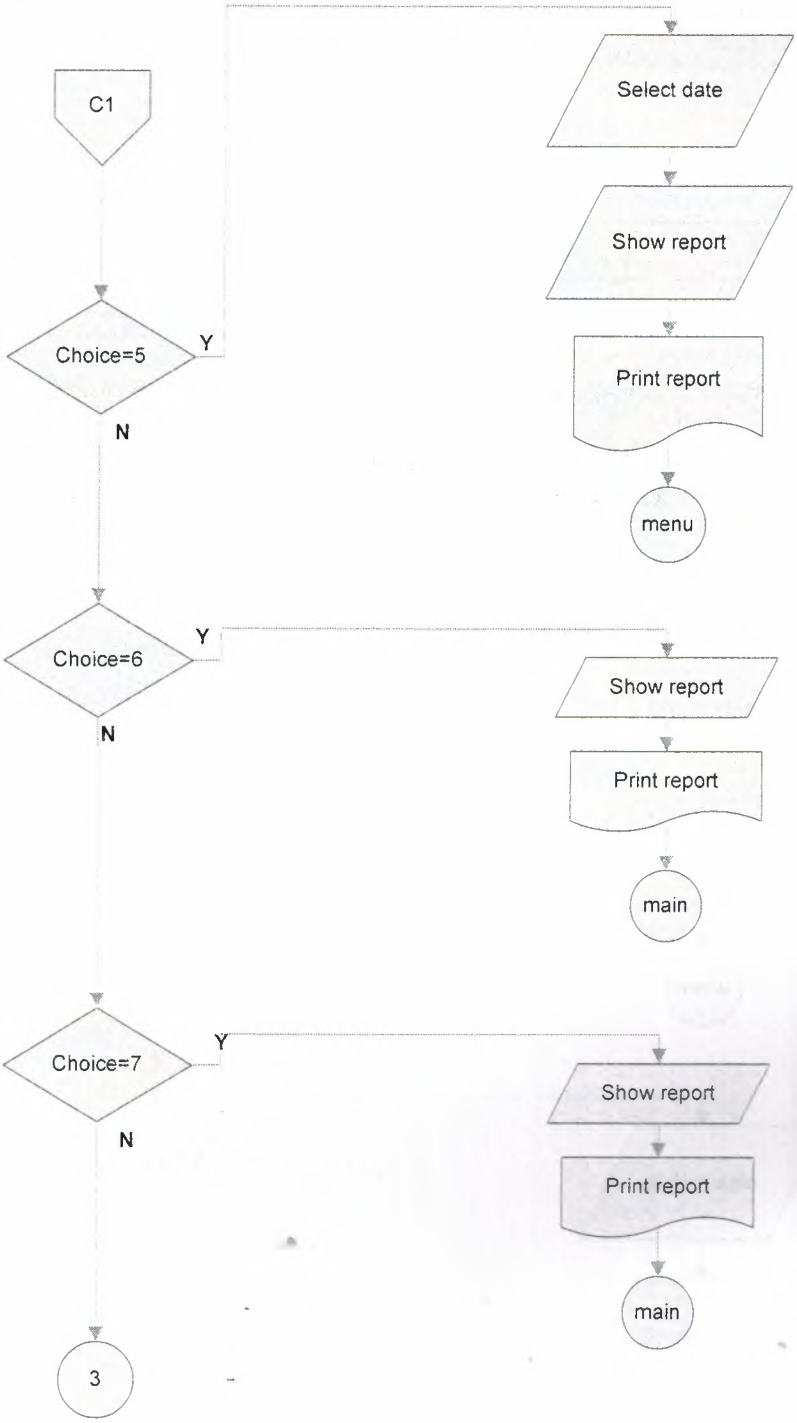


DELETE TRAVEL INFORMATION

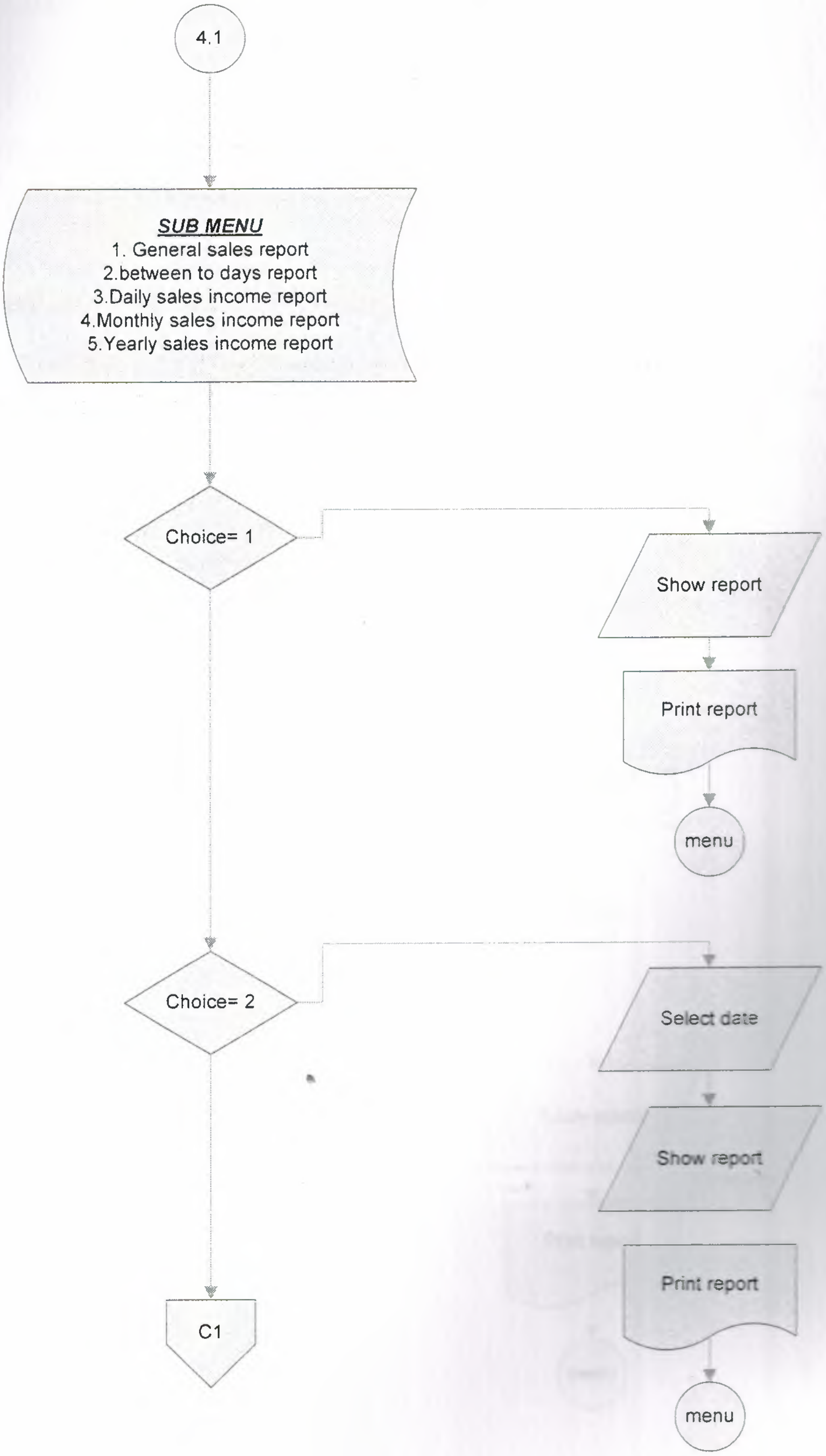


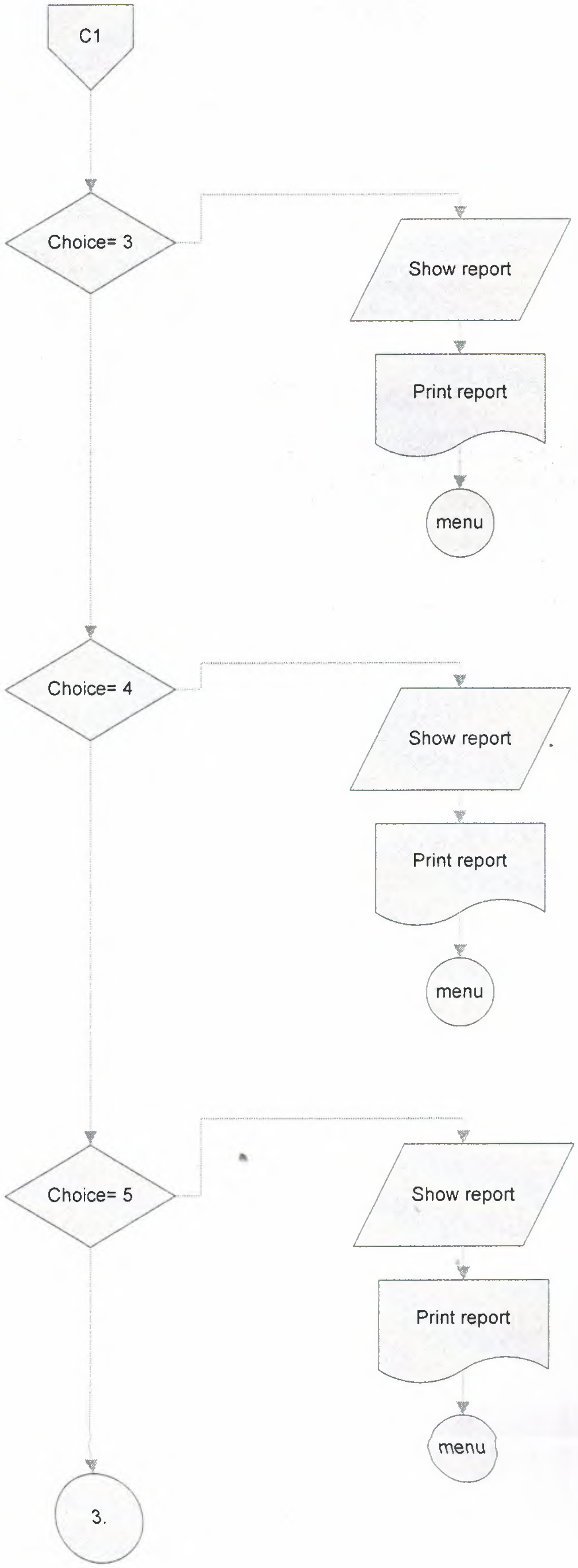
RAPORLAR


3.
SUBMENU
1. İncome/expense Report
2. Expense Reports
3. Payroll Report
4. Sales income Report
5. Travel(sefer) Reports
6.Employee Report
7.Customer report
Select your choice
Choice=1
Y
N
Select date
Show report
Print report
menu
Choice=2
Y
N
Select date
Show report
Print report
menu
Choice=3
Y
N
Choice=4
Y
N
4.1
C!


C1
Choice=5
Y
N
Select date
Show report
Print report
menu
Choice=6
Y
N
Show report
Print report
main
Choice=7
Y
N
Show report
Print report
main
3

SALES İNCOMER REPORTS




C1
Choice= 3
Show report
Print report
menu
Choice= 4
Show report
Print report
menu
Choice= 5
Show report
Print report
menu
3.

6.REFERENCES

Electronic Sources – Online sources from Web :

[1] Mastering Delphi 7 , Retrieved March 15, 2006 from the world wide web "www.marcocantu.com"

[2] Delphiturk CodeBank, Delphi Türkiye, "www.delphiturk.com"

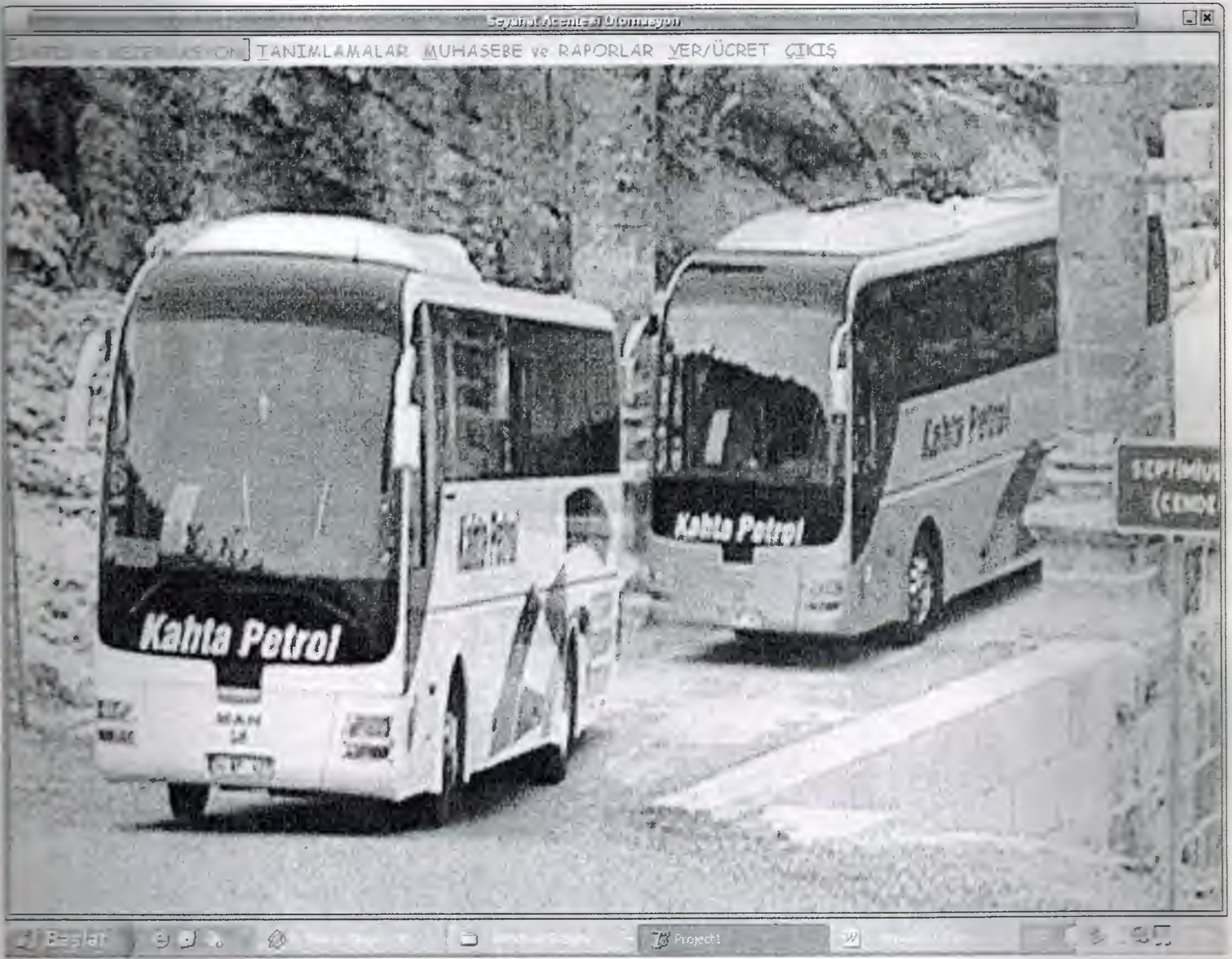
[3] Delphi in examples, Delphi about "www.delphi.about.com"

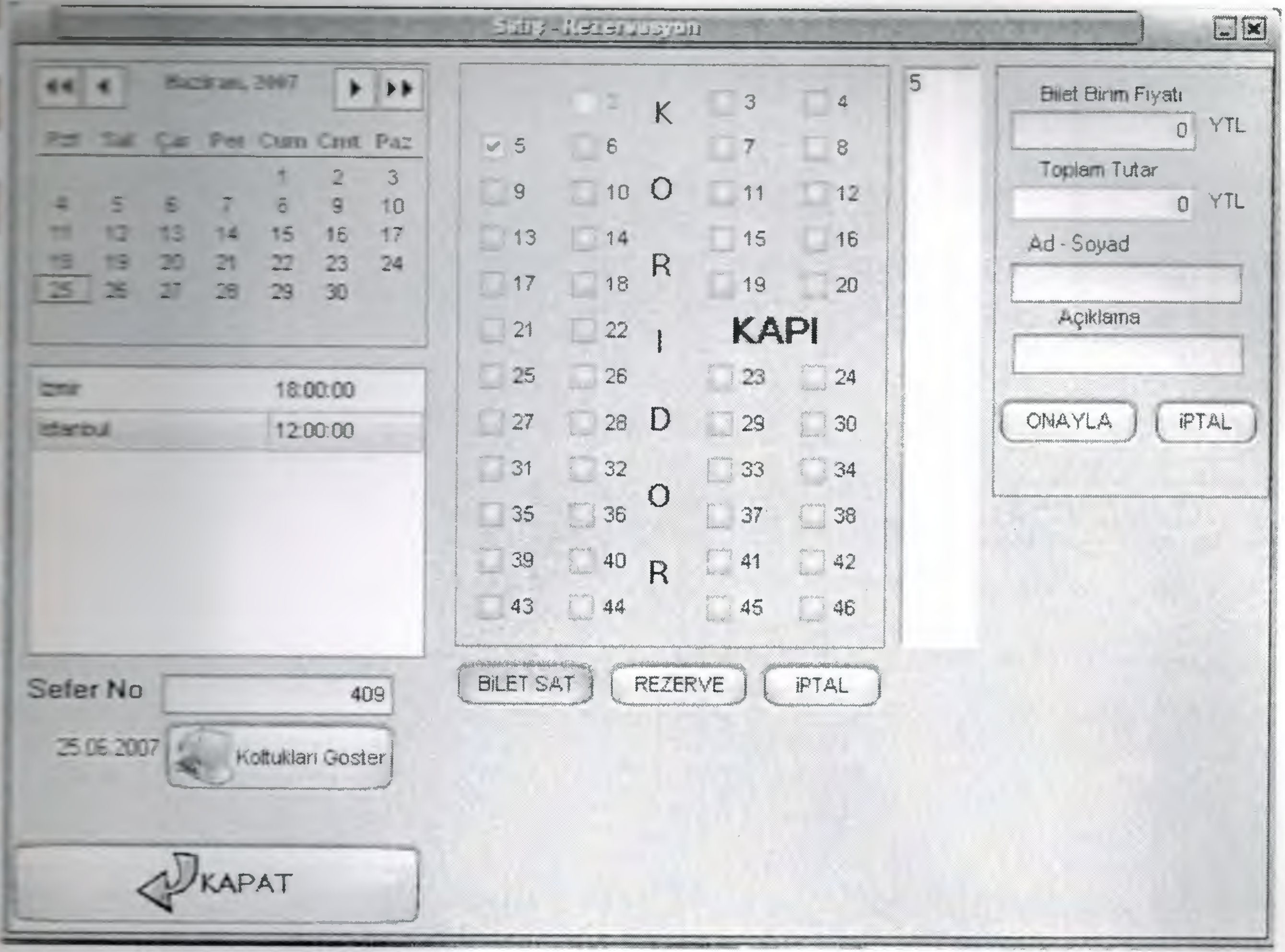
APPENDIX

SCREEN OUTPUTS

TRAVEL AGENCY AUTOMATION SYSTEM

In this chapter the screenshots and the user manual of the program seen explained.





Satış - Rezervasyon

Haziran, 2007

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

İzmir	18:00:00
İstanbul	12:00:00

Sefer No 409

25.06.2007 Koltukları Goster

KAPAT

	□ 2	K	□ 3	□ 4
☑ 5	□ 6		□ 7	□ 8
□ 9	□ 10	O	□ 11	□ 12
□ 13	□ 14		□ 15	□ 16
□ 17	□ 18	R	□ 19	□ 20
□ 21	□ 22	I	KAPI	
□ 25	□ 26		□ 23	□ 24
□ 27	□ 28	D	□ 29	□ 30
□ 31	□ 32		□ 33	□ 34
□ 35	□ 36	O	□ 37	□ 38
□ 39	□ 40	R	□ 41	□ 42
□ 43	□ 44		□ 45	□ 46

5

BİLET SAT REZERVE İPTAL

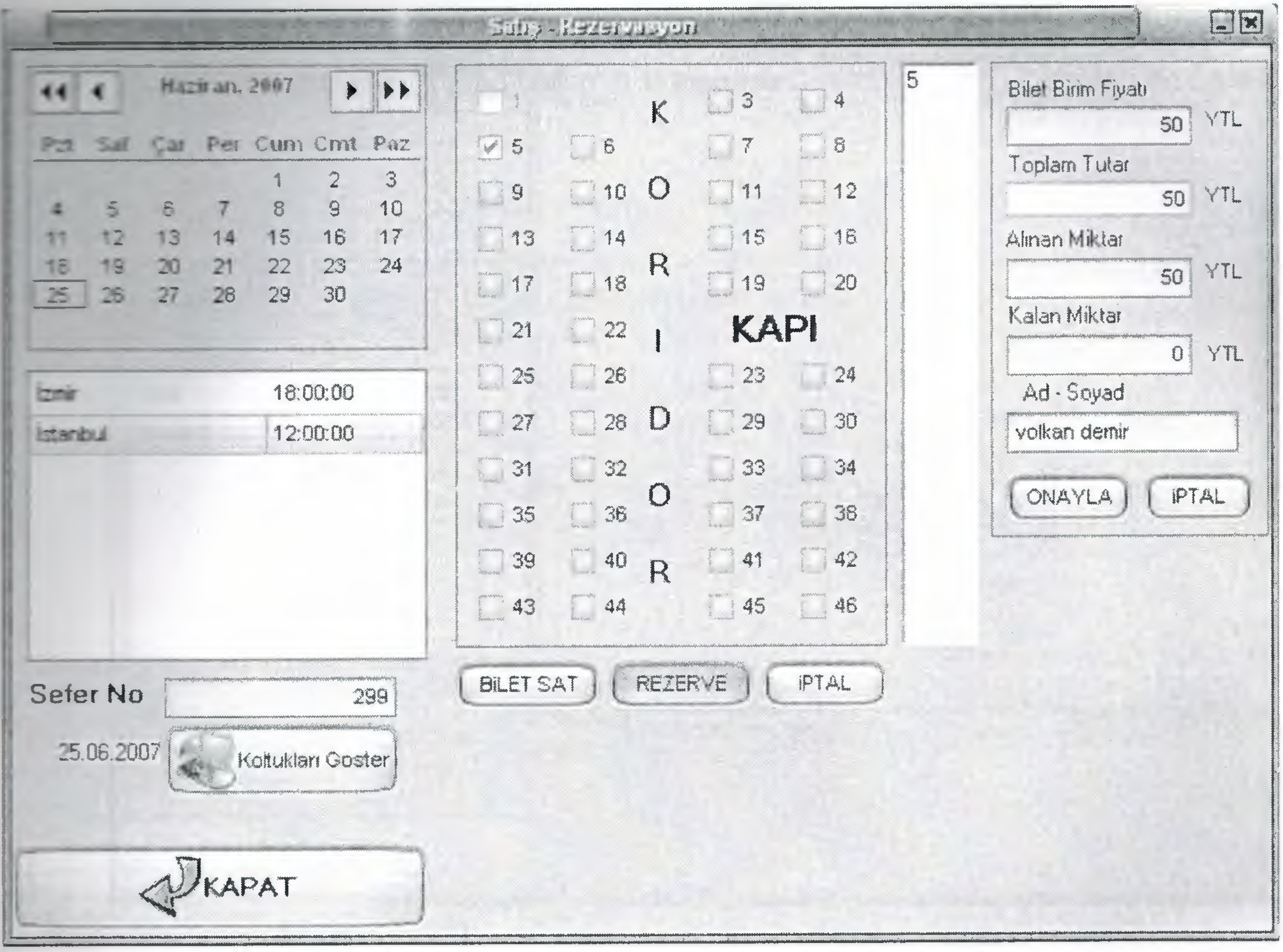
Bilet Birim Fiyatı 0 YTL

Toplam Tutar 0 YTL

Ad - Soyad

Açıklama

ONAYLA İPTAL



Satış - Rezervasyon

Haziran, 2007

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

İzmir	18:00:00
İstanbul	12:00:00

Sefer No: 299

25.06.2007

□ 1		K	□ 3	□ 4
☑ 5	□ 6		□ 7	□ 8
□ 9	□ 10	O	□ 11	□ 12
□ 13	□ 14		□ 15	□ 16
□ 17	□ 18	R	□ 19	□ 20
□ 21	□ 22	I	KAPI	
□ 25	□ 26		□ 23	□ 24
□ 27	□ 28	D	□ 29	□ 30
□ 31	□ 32		□ 33	□ 34
□ 35	□ 36	O	□ 37	□ 38
□ 39	□ 40	R	□ 41	□ 42
□ 43	□ 44		□ 45	□ 46

5

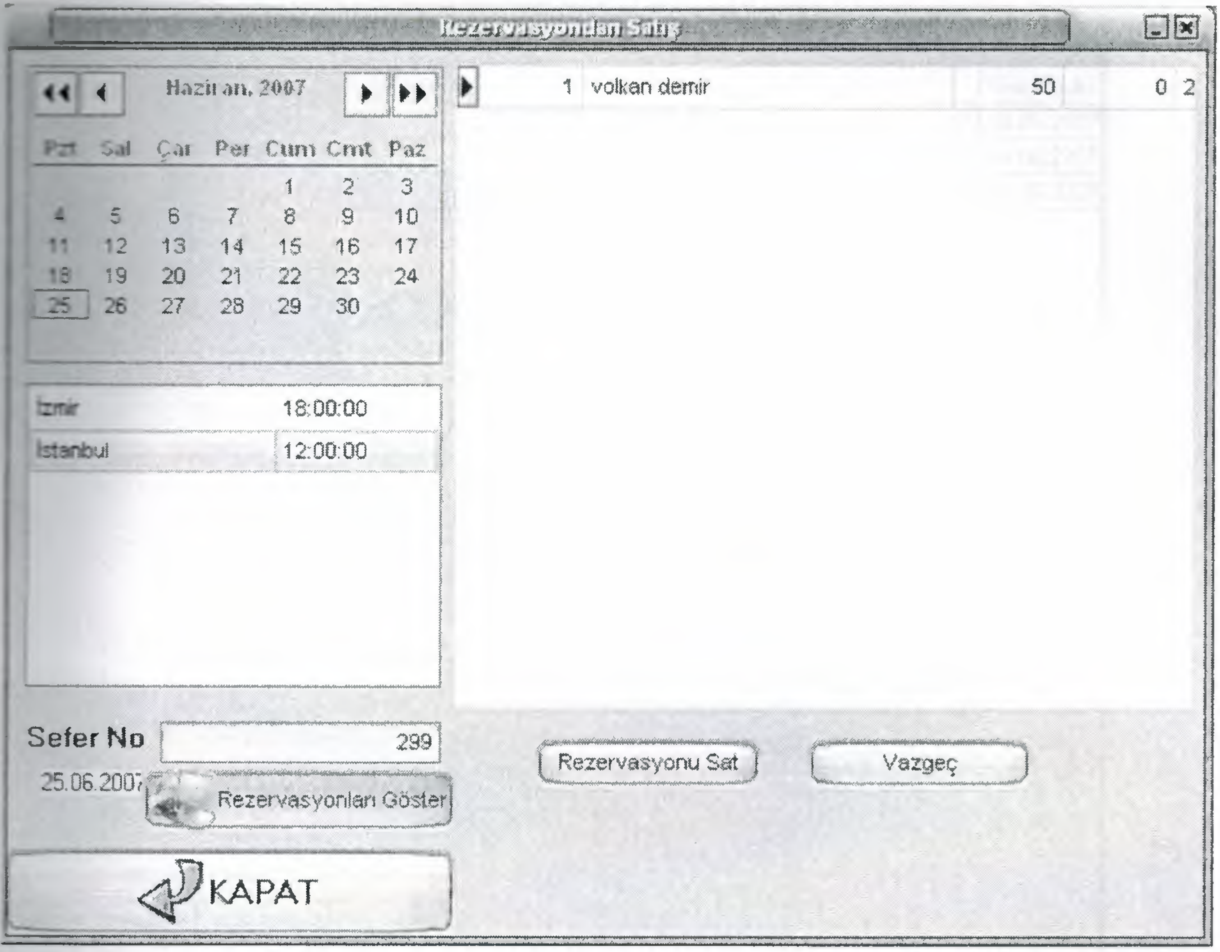
Bilet Birim Fiyatı: 50 YTL

Toplam Tutar: 50 YTL

Alınan Miktar: 50 YTL

Kalan Miktar: 0 YTL

Ad - Soyad: volkan demir



Rezervasyondan Satış

Haziran, 2007

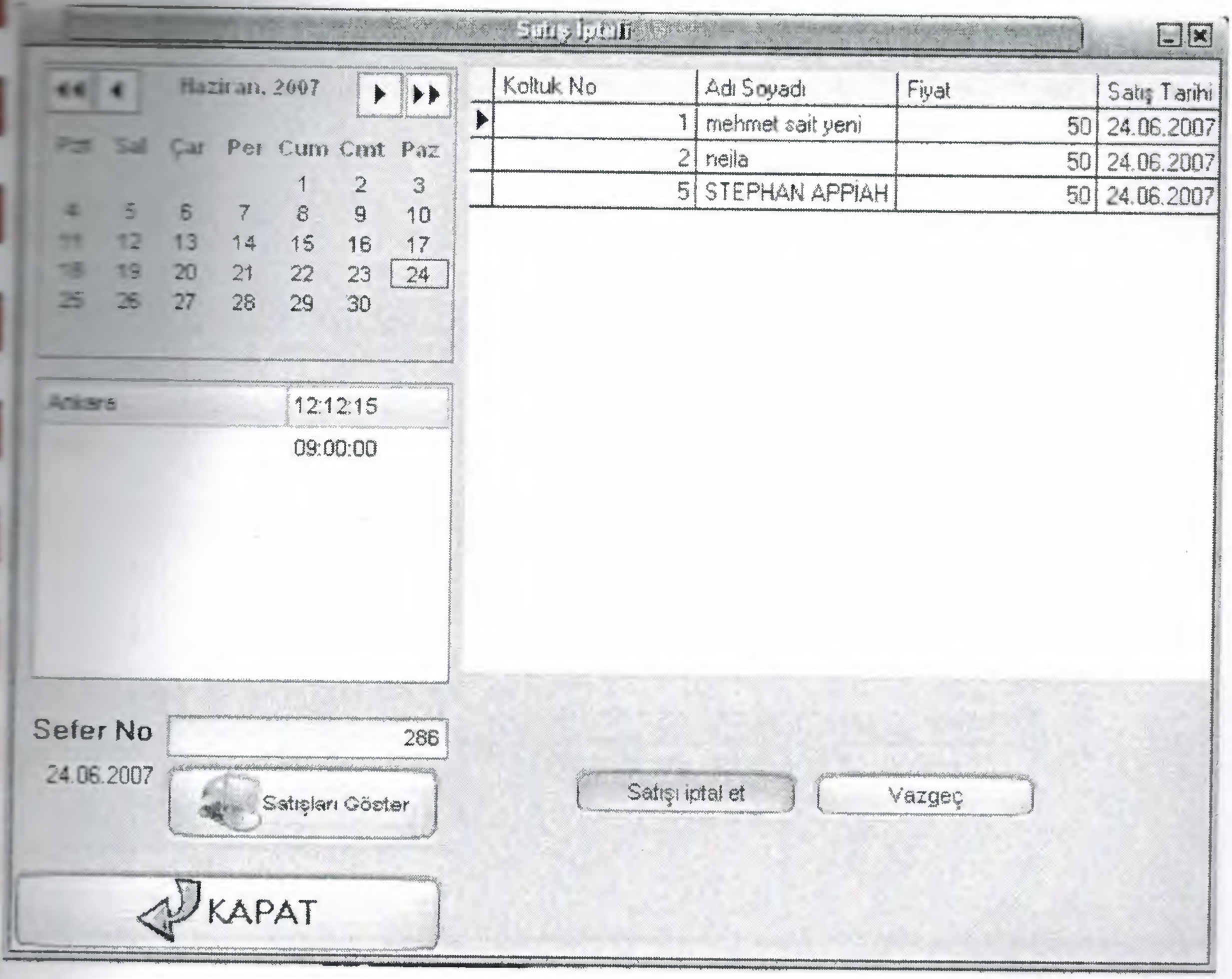
Pzt	Sal	Çar	Per	Cum	Cmt	Paz
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

İzmir	18:00:00
İstanbul	12:00:00

▶	1	volkan demir	50	0	2

Sefer No 299

25.06.2007

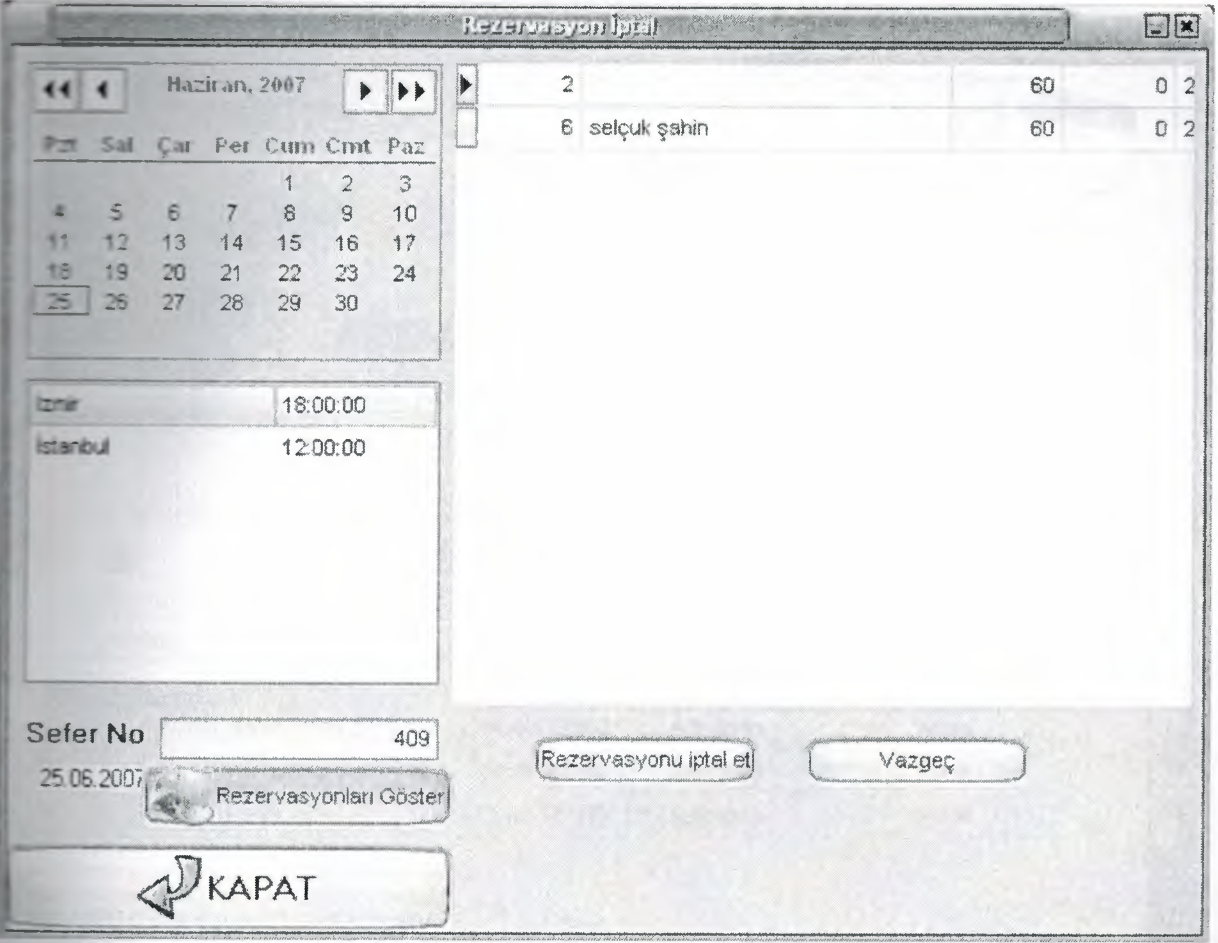


Koltuk No	Adı Soyadı	Fiyat	Satış Tarihi
1	mehmet sait yeni	50	24.06.2007
2	neila	50	24.06.2007
5	STEPHAN APPIAH	50	24.06.2007

Ankara	12:12:15
	09:00:00

Sefer No 286

24.06.2007



Rezervasyon İptal

Haziran, 2007

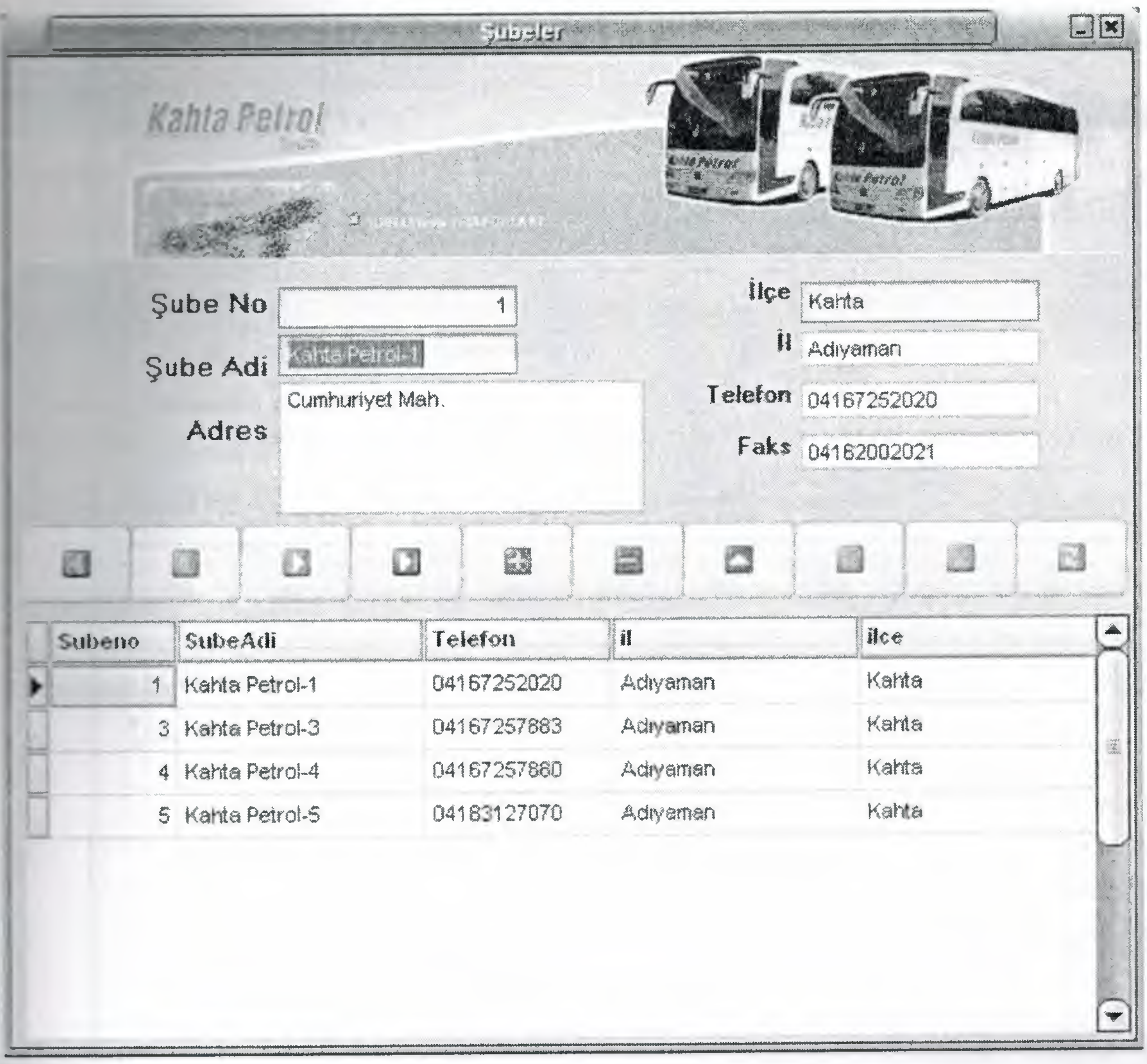
Pzt	Sal	Çar	Per	Cum	Cmt	Paz
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

İzmir	18:00:00
İstanbul	12:00:00

2		60	0	2
6	selçuk şahin	60	0	2

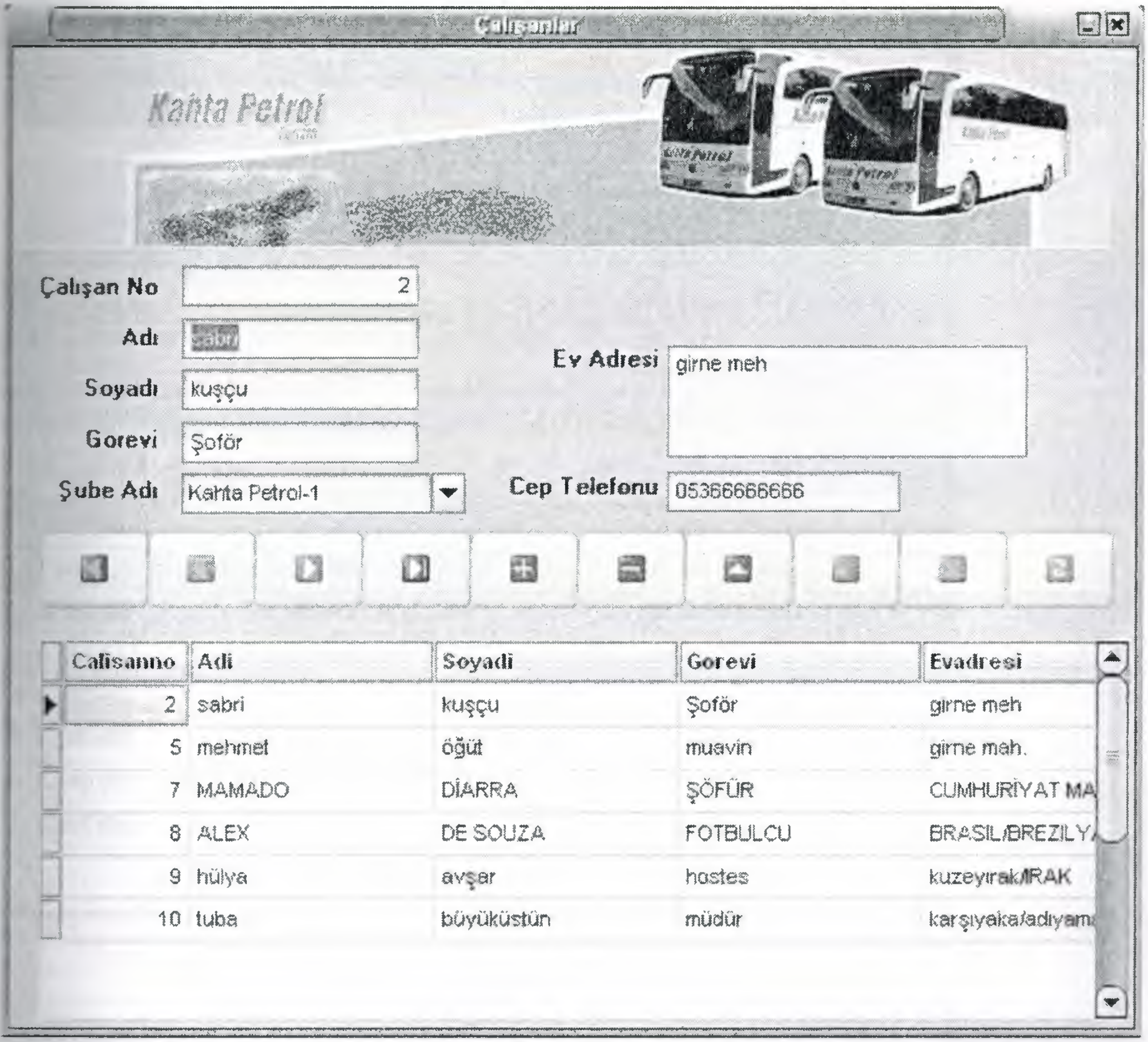
Sefer No 409

25.06.2007



Şube No	1
Şube Adi	Kahta Petrol-1
Adres	Cumhuriyet Mah.
İlçe	Kahta
İl	Adıyaman
Telefon	04167252020
Faks	04162002021

Subeno	SubeAdi	Telefon	il	ilce
1	Kahta Petrol-1	04167252020	Adıyaman	Kahta
3	Kahta Petrol-3	04167257883	Adıyaman	Kahta
4	Kahta Petrol-4	04167257860	Adıyaman	Kahta
5	Kahta Petrol-5	04183127070	Adıyaman	Kahta



Çalışan No: 2

Adı: sabri

Soyadı: kuşçu

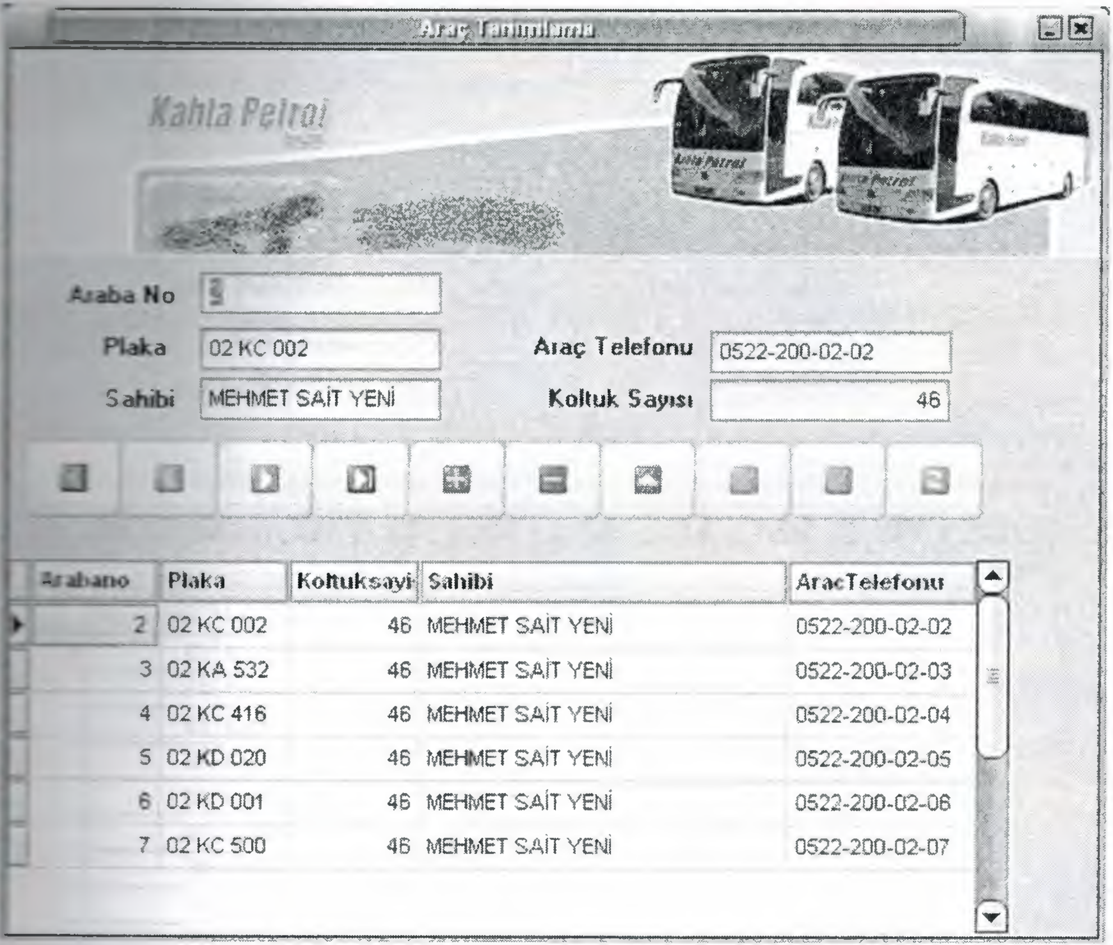
Gorevi: Şoför

Şube Adı: Kahta Petrol-1

Ev Adresi: girne meh

Cep Telefonu: 05366666666

Calisanno	Adi	Soyadi	Gorevi	Evadresi
2	sabri	kuşçu	Şoför	girne meh
5	mehmet	öğüt	muavin	girne mah.
7	MAMADO	DİARRA	ŞÖFÜR	CUMHURİYAT MA
8	ALEX	DE SOUZA	FOTBULCU	BRASIL/BREZILY
9	hülya	avşar	hostes	kuzeyırak/IRAK
10	tuba	büyüküstün	müdür	karşıyaka/adıyam



Araba No: 2

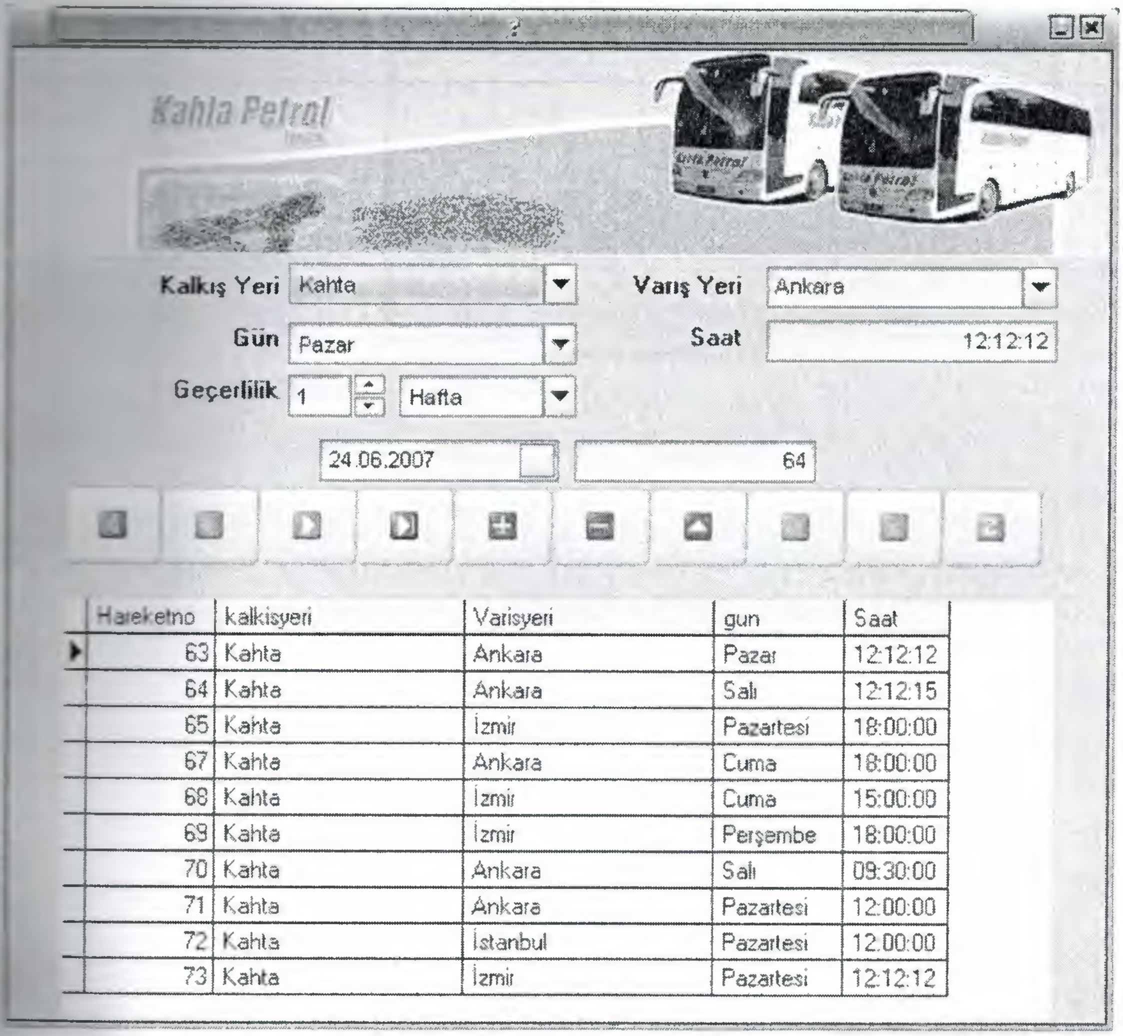
Plaka: 02 KC 002

Araç Telefonu: 0522-200-02-02

Sahibi: MEHMET SAİT YENİ

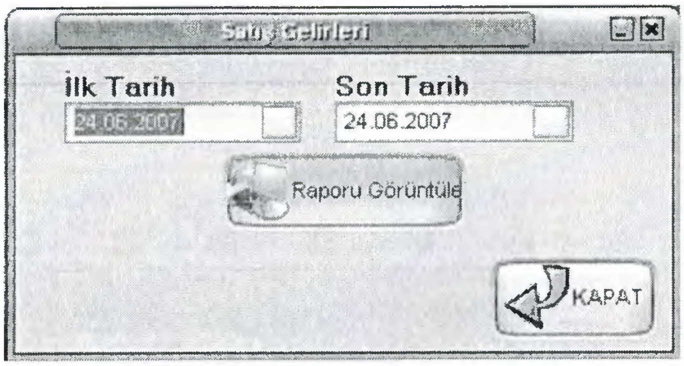
Koltuk Sayısı: 46

Arabano	Plaka	Koltuksayi	Sahibi	AracTelefonu
2	02 KC 002	46	MEHMET SAİT YENİ	0522-200-02-02
3	02 KA 532	46	MEHMET SAİT YENİ	0522-200-02-03
4	02 KC 416	46	MEHMET SAİT YENİ	0522-200-02-04
5	02 KD 020	46	MEHMET SAİT YENİ	0522-200-02-05
6	02 KD 001	46	MEHMET SAİT YENİ	0522-200-02-06
7	02 KC 500	46	MEHMET SAİT YENİ	0522-200-02-07



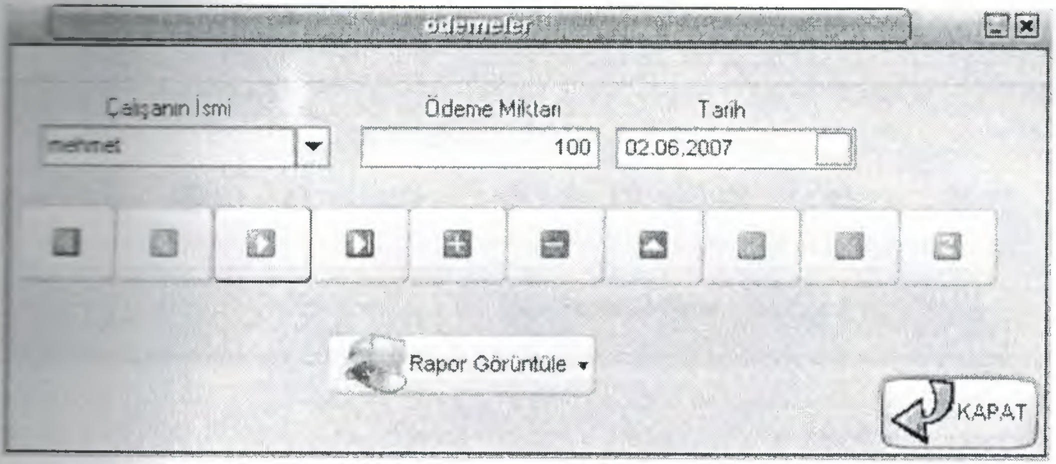
Kalkış Yeri	Kahta	Varış Yeri	Ankara
Gün	Pazar	Saat	12:12:12
Geçerlilik	1 Hafta		
	24.06.2007		64

Hareketno	kalkisyeri	Varisyeri	gun	Saat
63	Kahta	Ankara	Pazar	12:12:12
64	Kahta	Ankara	Salı	12:12:15
65	Kahta	İzmir	Pazartesi	18:00:00
67	Kahta	Ankara	Cuma	18:00:00
68	Kahta	İzmir	Cuma	15:00:00
69	Kahta	İzmir	Perşembe	18:00:00
70	Kahta	Ankara	Salı	09:30:00
71	Kahta	Ankara	Pazartesi	12:00:00
72	Kahta	İstanbul	Pazartesi	12:00:00
73	Kahta	İzmir	Pazartesi	12:12:12



İlk Tarih: 24.06.2007

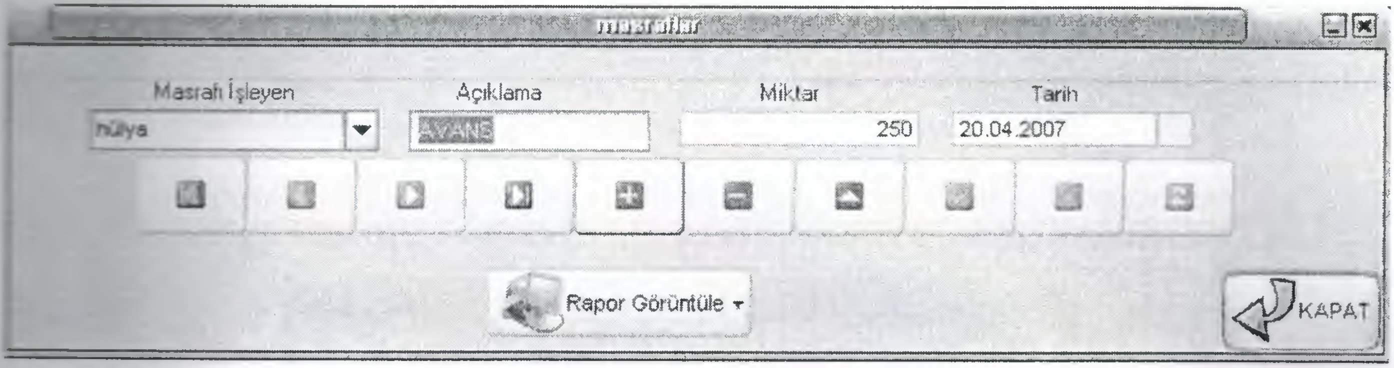
Son Tarih: 24.06.2007



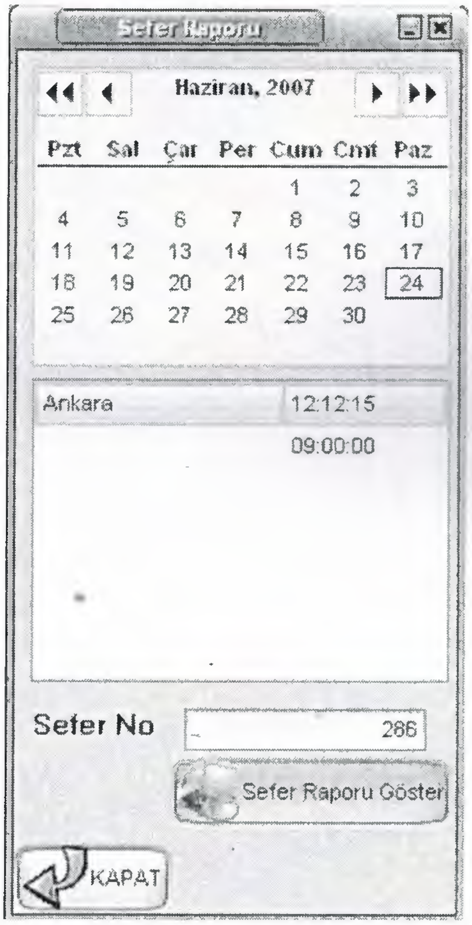
Çalışanın İsmi: mehmet

Ödeme Miktarı: 100

Tarih: 02.06.2007



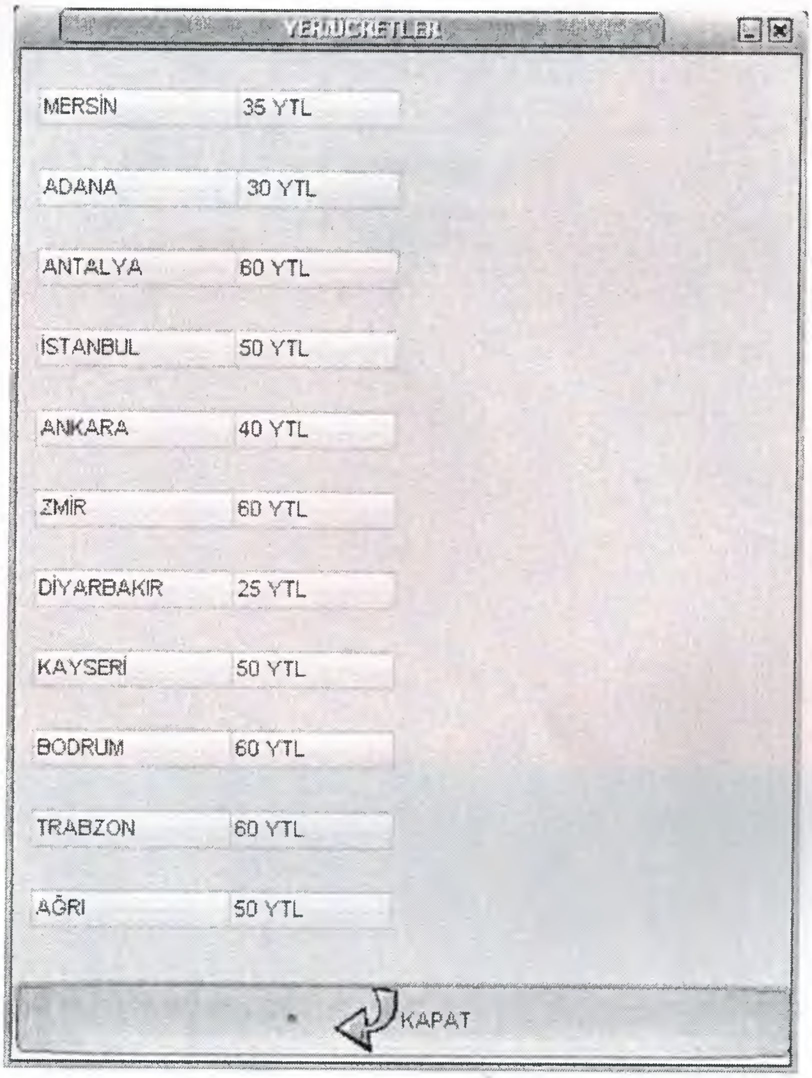
Masrafı İşleyen	Açıklama	Miktar	Tarih
nülya	AVANS	250	20.04.2007



Pzt	Sal	Çar	Per	Cum	Cmt	Paz
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Ankara	12:12:15
	09:00:00

Sefer No 286



MERSİN	35 YTL
ADANA	30 YTL
ANTALYA	60 YTL
İSTANBUL	50 YTL
ANKARA	40 YTL
ZMİR	60 YTL
DİYARBAKIR	25 YTL
KAYSERİ	50 YTL
BODRUM	60 YTL
TRABZON	60 YTL
AĞRI	50 YTL