

NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

CD Market – CD Rental or Sale Program

**Graduating Project
COM 400**

Student: Metin ULAŞ (20010733)

Supervisor : Mr. Ümit İLHAN

Lefkosa 2006

ACKNOWLEDGEMENTS

“Firstly I’m greatly indebted to my supervisor Mr. Ümit İLHAN, he was so helpful to us, I learned Visual programming language from him, he answered my questions Whenever I had. Also he never made me feel shame to ask him any question, oppositely he became happy when someone asks him a question, and he tries his best to understand him.

We have not to forget Mr. Okan DONANGİL, he was very helpfully during University life, whenever I cross any problem morally or educationally he helped me, I thank him very much with my all for his advices and guiding.

Secondly I would like to give my best regards to my family, especially my Mum, she deserved a lot to stay along from me, also I’m greatly indebted to my Dad, however his financial situation was not good first year but he resisted and forced me to continue my studding, he never made me need anyone. Also I thank my elder sisters, they encouraged me all the time to study and continue my duty, and they send me presents sometimes to make me happy.

*Thirdly I thank my friends Halil KİRİŞÇİ and Muhammed Simko for their help, they get tired with me, and they helped me in Graphic Design and in English writings.
I thank them with my all*

*Finally I would like to thank All my friends here in Cyprus and In Turkey especially Mr. Ömer ÖZDEMİR, he helped me morally and made me resist to continue and to be
An engineer I thank him very much.*

ABSTRACT

CD-MARKET Program is a useful program for CD markets. By using this program they can record and control everything happen in the market, for instance, selling CD's, renting, exchanging, coping...etc.

The program is easy in use, and everything is in detail, I used Visual Basic Programming Language in building it, also Access Database for storing information's. The program records sale operations when done, refunding CD's, exchanging it with another one, saving customer detail, showing top 10 list of CD's that wanted most at that period of time.

For every operation we have special form, for instance when we make a sale to a customer, we open sale page. When we buy new CD's for market, we have special page. For adding item's to stock we have special pages too.

I used many forms in this project, the main form is sale and rent form, when someone make a sale or rent a CD or exchange an old CD with a new one from a specific user in a specific date, the program records everything, we can query all the information's at any time, also when he made that sale and paid part of the money either in cash or by credit card or bank account, the program store paid amount and the remaining loan amount to his/her account.

Using this program enables the control of stock, control's whether the rented CD's comeback or not, queries information of any customer. In the same time users can't trick the manager, because whatever they do is recording by the program, users can't enter everywhere of the program, they are restricted by some pages, only the manager can enter wherever he wants.

TABLE OF CONTENTS

ACKNOWLEDGEMENT.....	i
ABSTRACT.....	ii
TABLE OF CONTENTS.....	iii
LIST OF TABLES.....	v
LIST OF FIGURES.....	vi
INTRODUCTION	vii
CHAPTER ONE: VISUAL BASIC.....	01
1.1.Introducing Visual Basic.....	01
1.2.Visual Basic Environment.....	01
1.3.Visual Basic for application	03
1.4.Step in building a Visual Basic Application.....	05
1.5.Working with controls and writing code.....	06
1.6.Managing data types.....	07
1.6.1.Numeric data types.....	07
1.6.2.Non-Numeric data types.....	07
1.6.3. Declering variables.....	08
1.7.Introduction to Visual Basic Function.....	09
1.7.1.Creating functions.....	09
1.8.Declaring array.....	12
1.9.Database application.....	12
1.9.1.UsingADOcontrol.....	12
CHAPTER TWO: DATABASE STRUCTURE.....	14
2.1.Brief information.....	14
2.2.Table structure in database.....	14
2.3.Relation between table.....	18
2.4.Microsoft Access Database.....	19
2.4.1.Microsoft access database fundamentals.....	19
2.4.2.Creating table.....	20
2.5.Introdusing database.....	20
2.6.Database Keys.....	21
2.7.Working with SQL.....	22
2.7.1.Data manipulation language.....	22
2.7.2.Insert statement.....	23
2.7.3 Select statement.....	23
CHAPTER THREE: STUDENT TRACKING SYSTEM.....	24
3.1.User entrence.....	24
3.2.Main page.....	25
3.2.1.Change pasword.....	26
3.2.2.Add new user or change details.....	26
3.2.3.Customer cards.....	27
3.2.3.1.Find customer by details.....	28

3.2.4.Product cards.....	29
3.2.4.1.Find product by details.....	30
3.2.4.2.Product types.....	31
3.2.4.3.Product category.....	31
3.2.5.Add product or customer picture.....	32
3.2.6. <i>Rent or sale product</i>	33
3.2.7.Show customer infromation details.....	34
3.2.8.Take product.....	35
3.2.9.Product refund.....	36
3.2.10.Cancel last sale.....	37
3.2.11.Customer accounts.....	38
3.2.12.Company Reports.....	39
3.2.12.1.Product refund details.....	40
3.2.12.2.Payment details.....	41
3.2.13.Product reports.....	42
3.2.14.Top 10 lists.....	43
CONCLUSION	44
REFERENCE	45
APPENDICES	46

LIST OF TABLES

Table 1.1: Numeric Data Types.....	07
Table 1.2: Nonnumeric Data Types.....	07
Table 2.1. User Database Table.....	14
Table 2.2. Stock Database Table.....	14
Table 2.3. Customer Database Table.....	15
Table 2.4. Category Database Table.....	15
Table 2.5. Producttype Database Table.....	15
Table 2.6. Sales Database Table.....	16
Table 2.7. Content Database Table.....	16
Table 2.8. Loan Database Table.....	16
Table 2.9. Payment Database Table.....	17
Table 2.10. Refusal Database Table.....	17

LIST OF FIGURES

Figure 1.1 The Visual Basic Start-up Dialog Box.....	02
Figure 1.2: The Visual Basic Enviroment.....	03
Figure1.3. Visual Basic Code Window.....	04
Figure 1.4 : The output of example 1.1.....	05
Figure1.5. Example1.4. output.....	09
Figure1.6. Examples1.5' Output.....	10
Figure 1.7. Showing MsgBox().....	11
Figure1.8. Using Ado Control.....	13
Figure 2.1. Relation Between Tables.....	18
Figure2.2. Sample Table.....	19
Figure 3.1. User Entrance Form.....	24
Figure 3.2.Main Page.....	25
Figure 3.2.1.Change Password.....	26
Figure 3.2.2.Add User or Change Details.....	26
Figure 3.2.3.Customer Cards.....	27
Figure 3.2.3.1.Find Customer by Details.....	28
Figure 3.2.4.Product Card.....	29
Figure 3.2.4.1.Find Product by Details.....	30
Figure 3.2.4.2.Product Types.....	31
Figure 3.2.4.3.Product Category.....	31
Figure 3.2.5.Add Customer or Product Picture.....	32
Figure 3.2.6.Rent or Sale Product.....	33
Figure 3.2.7.Show Customer Information Details.....	34
Figure 3.2.8. Take Product.....	35
Figure 3.2.9. Product Refund.....	36
Figure 3.2.10.Cancel Last Sale.....	37
Figure 3.2.11.Customer Accounts.....	38
Figure 3.2.12.Company Reports.....	39
Figure 3.2.12.1.Product Refund Details.....	40
Figure 3.2.12.2.Payment Informatoin Details.....	41
Figure 3.2.13.Product Reports.....	42
Figure 3.2.14.Top 10 Lists.....	43

INTRODUCTION

Stock Control & Customer Tracking Program for a CD-MARKET is a world widely used program for professional and big CD market's, by the way of this program we can arrange everything of the market, if we use the program in a correct way nothing will be confused during any operation that take place in market for present time and future, also we can investigate past sale operations too.

In the first Chapter Visual basic programming language is described, its properties, components and some examples, I used Visual Basic Program in my project, because I find it easy and I liked its coding system. Visual basic for applications delivers a competitive advantage for ISV seeking to provide full customization and integration capabilities to customer.

In the Second Chapter I described Database system, I used Microsoft Access Database system in my program with Visual basic, Advantages of using access is to provide exactly the same options for the problems we write as it does for the problems we selected from a database. Secondly, the process of writing or selecting problems is almost independent of page layout dictations. Also you can see more details about access and SQL.

Third Chapter is Flow Chart of my program, all the operations that take place in the project, decisions...etc. is described in the form of flow chart.

Finally, the last chapter is the explanation of the program followed by the Appendices. So by developing and moderating of technology our program can be developed and updated. Also new properties could be added in to the program in the future.

CHAPTER ONE: VISUAL BASIC

1.1. Introducing Visual Basic

VISUAL BASIC is a high level programming language evolved from the earlier DOS version called **BASIC**. **BASIC** means **B**eginners' **A**ll-purpose **S**ymbolic **I**nstruction **C**ode. It is a fairly easy programming language to learn. The codes look a bit like English Language. Different software companies produced different version of **BASIC**, such as Microsoft **QBASIC**, **QUICKBASIC**, **GW BASIC**, **IBM BASICA** and so on.

Firstly, I want to say to you about Visual Basic on the screen. When you apply to it, you will see, VB menu and tools, form, toolbox, properties window, project explorer, form layout and source code window. You can see below clearly.

Nowadays, software is very popular. Many programmers do computer programming. If you are a computer programmer, you can consider visual basic as your language of choice for learning how to develop computer programs. Of course there are other programming language is popular. But, Visual Basic is simple as possible. Because, to learning visual basic is so easy, also to do visual interface, has involved into a very powerfull development tools and fun to use. But programming in Visual Basic is a challenging. Thats basic reason is easyness of Visual Basic. Everbody can do program with visual basic easily. All that facilities shows that, someone can learn Visual Basic itself and train it.

Learning Visual Basic is based on only practise. As I said before, it is visual. Directly, you are in an interaction with your graphical user interface. You must not hasitate to try your simple programs, your imagination, try it in Visual Basic programming. As you know, best way to learn Visual Basic is by examples. Also you can find information to learn about visual basic on the internet and can use a best book. Look from there and try what you learn.

Let me remind to you one more things about best programmer. As you estimate, which is to enjoy programming, just be relax and program for fun and treat programming as a hobby. You will be amazed at the progress you make and success.

1.2 The Visual Basic Environment

On start up, Visual Basic 6.0 will display the following dialog box as shown in figure 1.1. You can choose to start a new project, open an existing project or select a list of recently opened programs. A project is a collection of files that make up your application. There are various types of applications we could create, however, we shall concentrate on creating Standard EXE programs (EXE means executable program). Now, click on the Standard EXE icon to go into the actual VB programming environment.

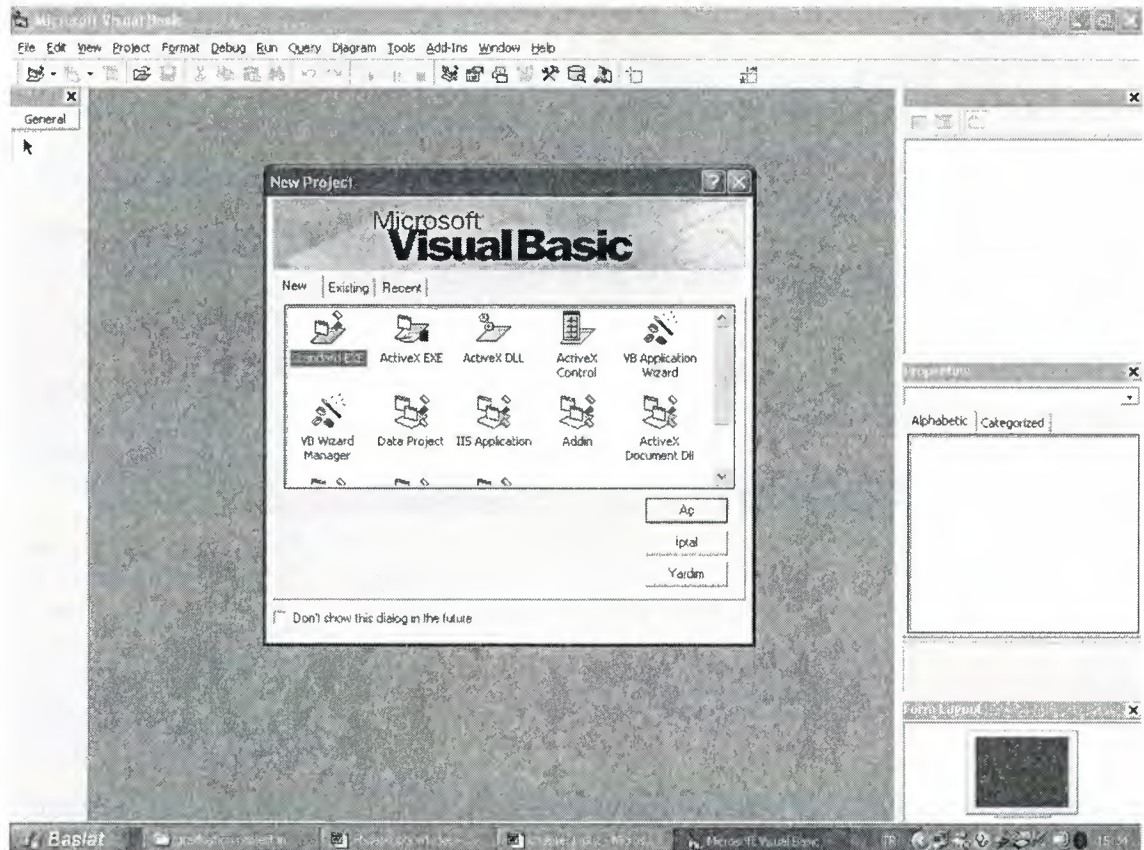


Figure 1.1 The Visual Basic Start-up Dialog Box

In figure 1.2, the Visual Basic Environment consists of the:

- A Blank Form for you to design your application's interface.
- The Project window which displays the files that are created in your application.
- The Properties window which displays the properties of various controls and objects that are created in your applications.

It also includes a Toolbox that consists of all the controls essential for developing a VB Application. Controls are tools such as text box, command button, label, combo box, picture box, image box, timer and other objects that can be dragged and drawn on a form to perform certain tasks according to the events associated with them. Additional objects can be added by clicking on the project item on the menu and click on components on the drop-down list, then select those controls you need to use in your program

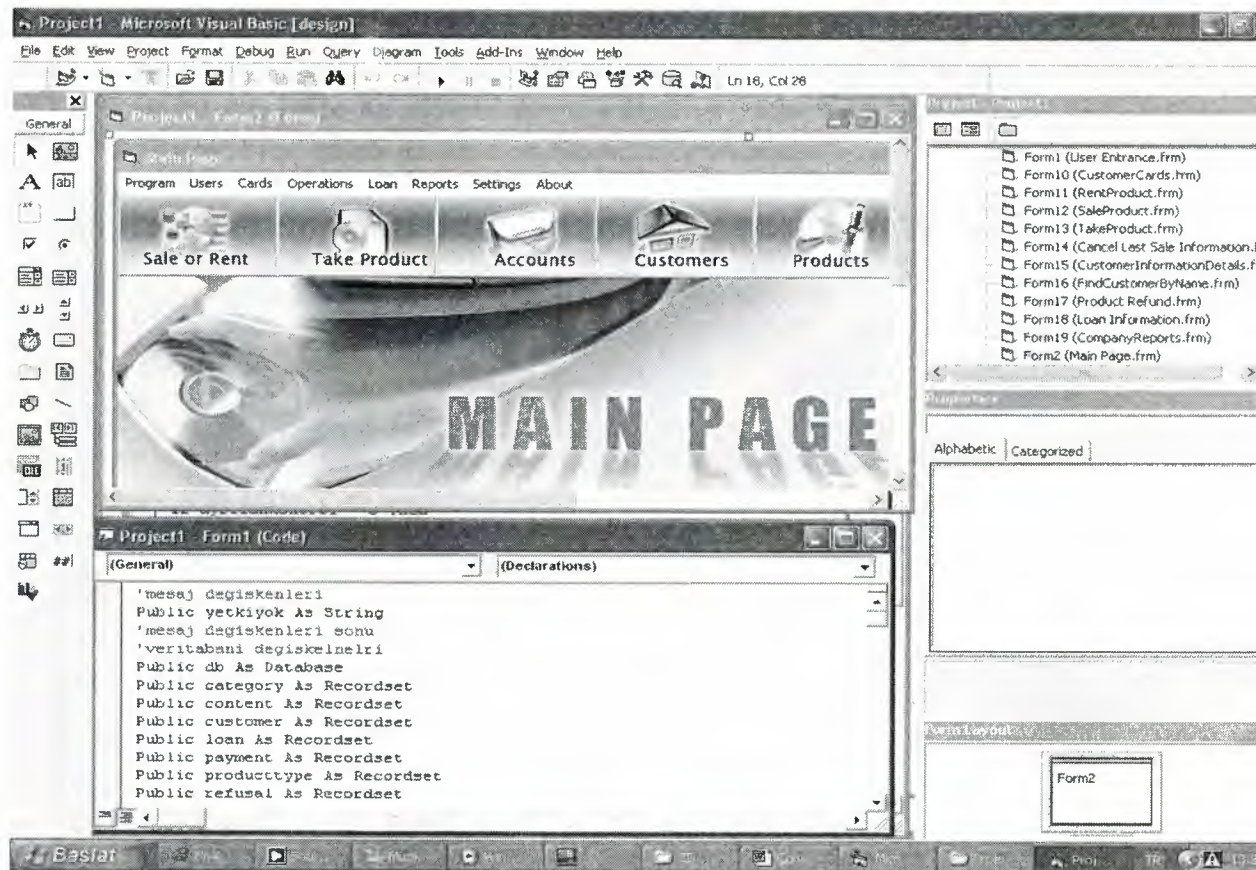


Figure 1.2: The Visual Basic Enviroment

1.3. Visual Basic For Application

Visual Basic for application provides a complete integrated development environment. It is a powerful development technology for rapidly customizing rich client desktop packaged application and integrating them with existing data and system. VBA offers a sophisticated set of programming tools based on the Microsoft Visual Basic Development system, the worlds most popular rapid application development system, which developpers can use to harness the power of packaged applications.

As mentioned before, Microsoft Visual Basic includes many window to facilitate programmers for their application. Which are VB menu and tools, form, toolbox, properties window, project explorer, form layout, debugging tools, and source code window. VBA also include supporting for Microsoft Form, for creating custom dialog boxes, and ActiveX controls for quickly building user interfaces

Software programs, which is include VBA, it is called customizable applications, that can be tailored to fit specific business need. This class of application enables developpers to rapidly built solution that require less and user trainig. For MIS and business managers, customization means that solution can be developped quickly and deployed easily with minimal maintenance. In an industry familiar with two years backlogs for new applications and high end user training cost. This solution provide a tremen dous business benefit in terms of retern on investment and timelines.

In this section, we are not going into the technical aspects of VB programming; just have a feel of it. Now, you can try out the examples. Now you will see application example by writing code. Of course, you can examine by writing code in to Microsoft Visual Basic Code Window. Which is shown below in figure1.4.

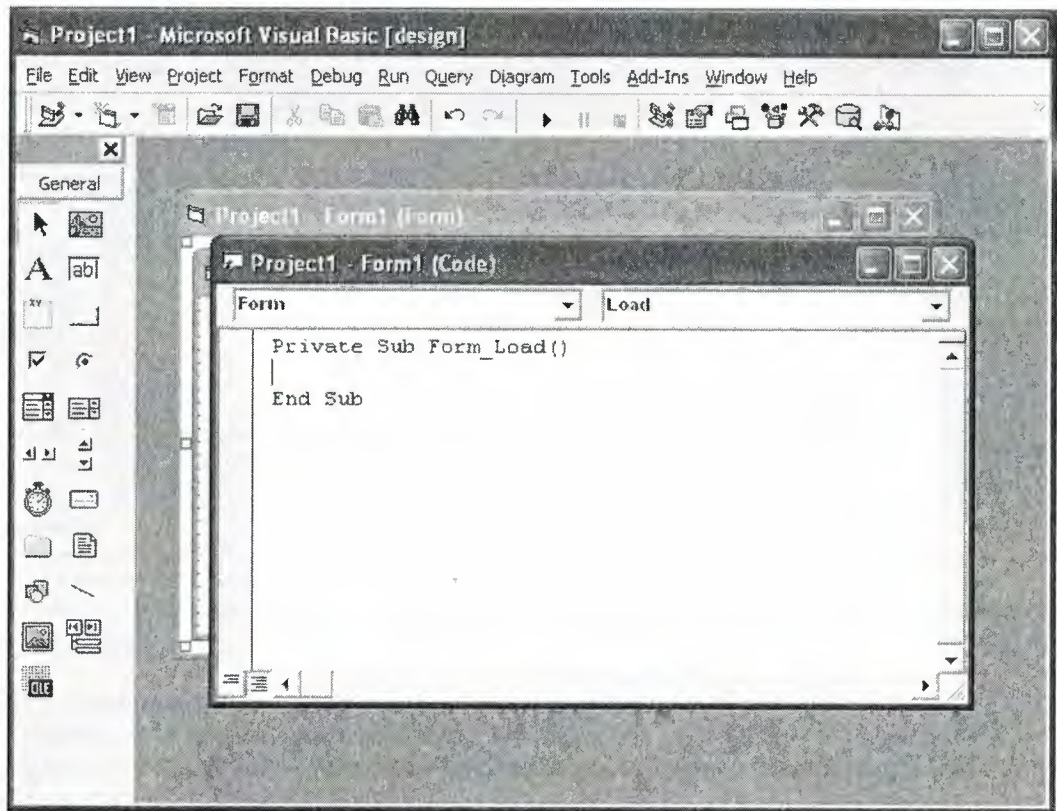


Figure1.3. Visual Basic Code Window

Now you can drop onto form command button, list box, text box, option....etc and select an object and procedure then can write code whatever you want. You can see example below.

Example 1.1.

```
Private Sub Command1_Click()  
Label1.Caption = "Com 400 Graduation Project Metin ULAŞ 20010733"  
End Sub
```

```
Private Sub Form_Load()  
Command1.Caption = "Display"  
Label1.Caption = "You Can Change This Dialog With This Button..!"  
End Sub
```



```
Private Sub Timer1_Timer()
```

```
Label1.Caption = "You Can Change This Dialog With This Button..!"
```

```
End Sub
```

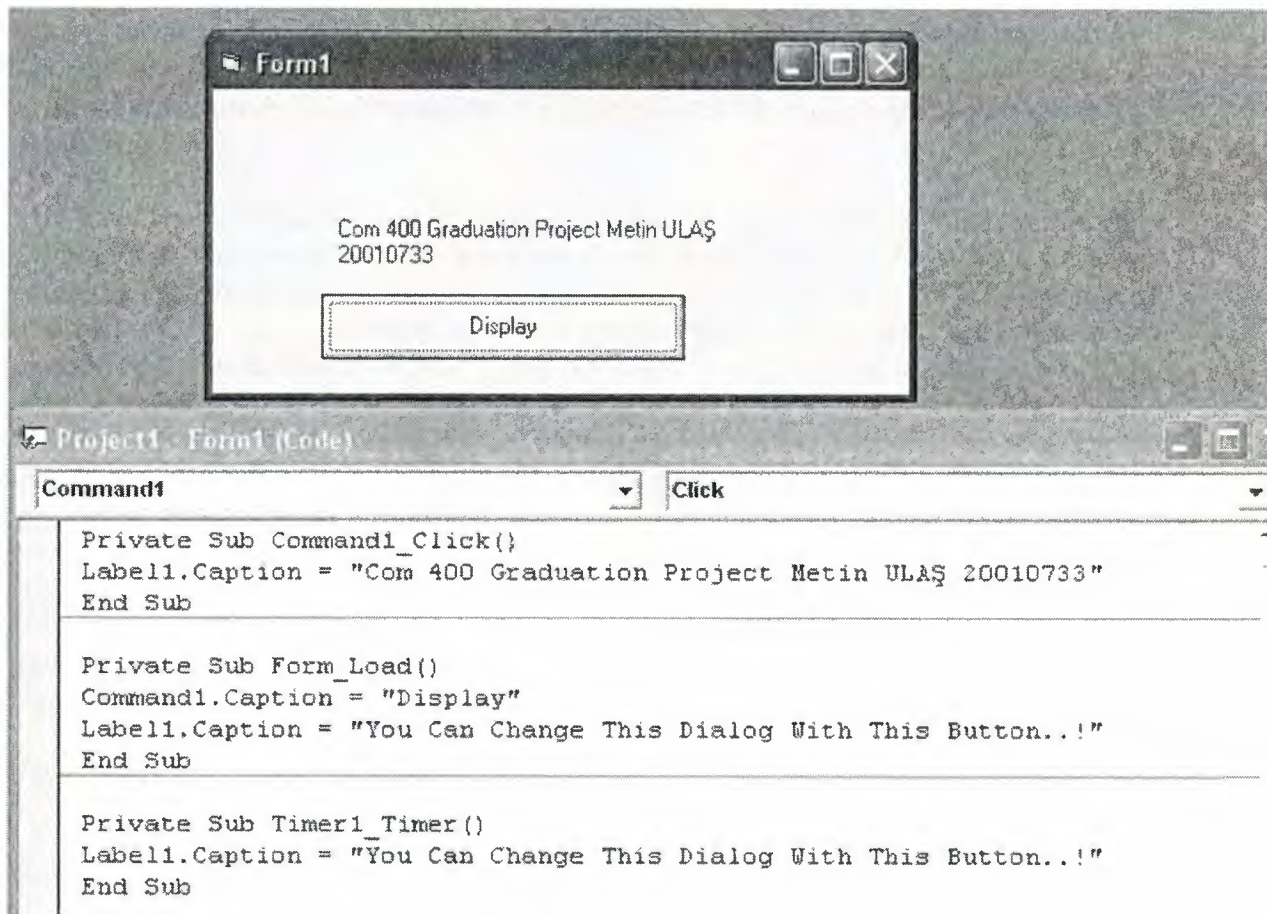


Figure 1.4 : The output of example 1.1.

1.4. Steps in Building a Visual Basic Application

Generally, there are three basic steps in building a VB application. The steps are as follows:

Step 1 : Design the interface

Step 2 : Set Properties of the controls (Objects)

Step 3 : Write the events' procedures

When you will write program. If you can follow that steps whichs are benefit for you. You can improve your project by using other opportunities. Lets talk about other

opportunities. Which are you can design your project without using code. You can change your objects on to your form from properties window, tools, menu editor, format..etc.

1.5. Working With Controls And Writing Code

Before writing an event procedure for a control to response to a user's input, you have to set certain properties for the control to determine its appearance and how it will work with the event procedure. You can set the properties of the controls in the properties windows.

you have learned how to enter the program code and run the sample VB programs but without much understanding about the logics of VB programming. Now, let's get down learning a few basic rules about writing the VB program code. Each control or object in VB can usually run many kinds of events or procedures; these events are listed in the dropdown list in the code window that is displayed when you double-click on an object and click on the procedures' box(refer to Figure 2.3). Among the events are loading a form, clicking of a command button, pressing a key on the keyboard or dragging an object and etc. For each event, you need to write an event procedure so that an action or a series of actions can be performed.

To start writing an event procedure, you need to double-click an object. For example, if you want to write an event procedure when a user clicks a command button, you double-click on the command button and an event procedure will appear like in figure2.4

Example 1.2.

```
Private Sub Command1_Click()  
    Label1.Caption = "Com 400 Graduation Project Metin ULAŞ 20010733"  
End Sub
```

Syntaxes that do not involve setting of properties are also English-like, some of the commands are **Print**, **If...Then....Else....End If**, **For...Next**, **Select Case....End Select**, **End** and **Exit Sub**. For example, **Print " Visual Basic"** is to display the text Visual Basic on screen and **End** is to end the program. Other commands will be explained in details in the coming lessons.

Program codes that involve calculations is very easy to write, you need to write them almost like what you do in mathematics. However, in order to write an event procedure that involves calculations, you need to know the basic arithmetic operators in VB as they are not exactly the same as the normal operators we use, except for + and -. For multiplication, we use *, for division we use /, for raising a number x to the power of n, we use **x ^n** and for square root, we use **Sqr(x)**. More advanced mathematical functions such as **Sin**, **Cos**, **Tan**, **Log** and etc. There are also two important functions that are related to arithmetic operations, i.e. the functions **Val** and **Str\$** where **Val** is to convert text entered into a textbox to numerical value and **Str\$** is to display a numerical value in a textbox as a string (text). While the function **Str\$** is as important as VB can display a numeric values as string implicitly, failure to use **Val** will results in wrong calculation. Let's examine Example1.1.

1.6. Managing Data Types

There are many types of data we come across in our daily life. For example, we need to handle data such as names, addresses, money, date, stock quotes, statistics and etc everyday. Similarly in Visual Basic, we are also going to deal with these kinds of data. However, to be more systematic, VB divides data into different types.

1.6.1 Numeric Data

Numeric data are data that consists of numbers, which can be computed mathematically with various standard operators such as add, minus, multiply, divide and so on. In Visual Basic, the numeric data are divided into 7 types, they are summarized in Table1.1

Type	Storage	Range of Values
Byte	1 byte	0 to 255
Integer	2 bytes	-32,768 to 32,767
Long	4 bytes	-2,147,483,648 to 2,147,483,648
Single	4 bytes	-3.402823E+38 to -1.401298E-45 for negative values 1.401298E-45 to 3.402823E+38 for positive values.
Double	8 bytes	-1.79769313486232e+308 to -4.94065645841247E-324 for negative values 4.94065645841247E-324 to 1.79769313486232e+308 for positive values.

Table 1.1: Numeric Data Types

1.6.2 Non-numeric Data Types

The nonnumeric data types are summarized in Table1.2

Data Type	Storage	Range
String(fixed length)	Length of string	1 to 65,400 characters
String(variable length)	Length + 10 bytes	0 to 2 billion characters
Date	8 bytes	January 1, 100 to December 31, 9999
Boolean	2 bytes	True or False

Table 1.2: Nonnumeric Data Types

1.6.3. Declaring Variables

In Visual Basic, one needs to declare the variables before using them by assigning names and data types. They are normally declared in the general section of the codes' windows using the **Dim** statement.

The format is as follows:

Dim variableName as DataType

Example 1.3

```
Dim password As String
Dim yourName As String
Dim firstnum As Integer
Dim secondnum As Integer
Dim total As Integer
Dim doDate As Date
```

You may also combine them in one line, separating each variable with a comma, as follows: Dim password As String, yourName As String, firstnum As Integer,.....

If....Then....Else Statements with Operators

To effectively control the VB program flow, we shall use If...Then...Else statement together with the conditional operators and logical operators.

The general format for the if...then...else statement is

If conditions **Then**

VB expressions

Else

VB expressions

End If

Select Case

If you have a lot of conditional statements, using If..Then..Else could be very messy. For multiple conditional statements, it is better to use Select Case. The format is :

Select Case expression

Case value1

Block of one or more VB statements

Case value2

Block of one or more VB Statements

End Select

Looping

Visual Basic allows a procedure to be repeated as many times as long as the processor could support. This is generally called looping.

1. Do

Block of one or more VB statements

Loop While condition

2. Do

Block of one or more VB statements

Loop Until condition

3. For....Next Loop

For counter=startNumber to endNumber (Step increment)

One or more VB statements

Next

1.7. Introduction to VB Functions

1.7.1 Creating Functions

The general format of a function is as follows:

Public Function **functionName** (Arg As dataType,.....) As dataType
or

Private Function **functionName** (Arg As dataType,.....) As dataType

*Public indicates that the function is applicable to the whole program and
Private indicates that the function is only applicable to a certain module or procedure.

Example 1.4

In this example, a user can calculate future value of a certain amount of money he has today based on the interest rate and the number of years from now supposing he will invest this amount of money somewhere). The calculation is based on the compound interest rate.

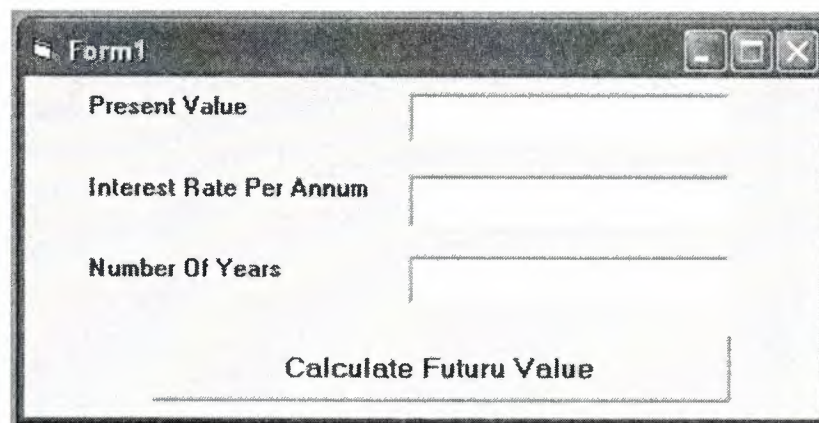
A screenshot of a Visual Basic form titled "Form1". The form has a light gray background and a dark border. It contains three input fields, each with a label to its left: "Present Value", "Interest Rate Per Annum", and "Number Of Years". Below these fields is a button labeled "Calculate Futuru Value". The form is set against a dark background.

Figure1.5. Example1.4. output

```
Public Function FV(PV As Variant, i As Variant, n As Variant) As Variant
'Formula to calculate Future Value(FV)
'PV denotes Present Value
FV = PV * (1 + i / 100) ^ n
End Function
```

```

Private Sub compute_Click()
'This procedure will calculate Future Value
Dim FutureVal As Variant
Dim PresentVal As Variant
Dim interest As Variant
Dim period As Variant
PresentVal = PV.Text
interest = rate.Text
period = years.Text
FutureVal = FV(PresentVal, interest, period)
MsgBox ("The Future Value is " & FutureVal)
End Sub

```

Example1.5.

The following program will automatically compute examination grades based on the marks that a student obtained.

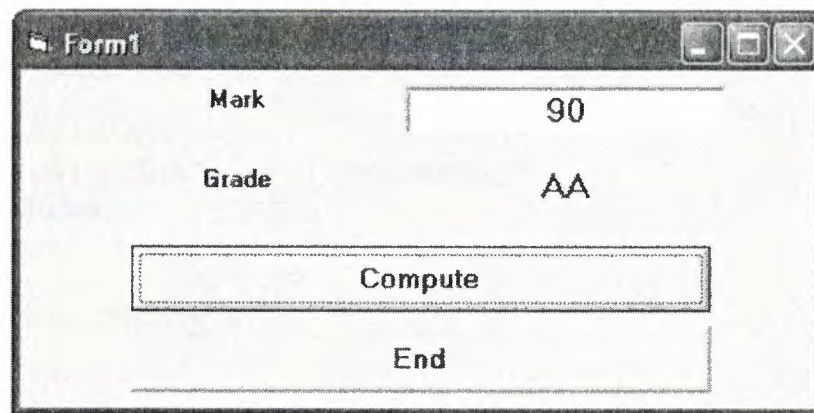


Figure1.6. Examples1.5' output

```

Public Function grade(mark As Variant) As String
Select Case mark
Case Is >= 80
grade = "A"
Case Is >= 70
grade = "B"
Case Is >= 60
grade = "C"
Case Is >= 50
grade = "D"
Case Is >= 40
grade = "E"
Case Else
grade = "F"
End Select
End Function

```



```
Private Sub compute_Click()  
grading.Caption = grade(mark)  
End Sub
```

```
Private Sub End_Click()  
End  
End Sub
```

I want to remind you something about functions in Visual Basic. Also message boxes are functions inside of Visual Basic. When you write **msgbox()** programs automatically call functions from inside of Visual Basic. I will explain little with by example. In this explanation, we are going to learn two very basic but useful internal functions, i.e. the **MsgBox()** and **InputBox()** functions.

The objective of **MsgBox** is to produce a pop-up message box and prompt the user to click on a command button before he/she can continue. This message box format is as follows:

Example 1.6.

```
Private Sub Test_Click()  
  
Dim testmsg As Integer  
testmsg = MsgBox("Click to test", 1, "Test message")  
If testmsg = 1 Then  
Display.Caption = "Testing Successful"  
Else  
Display.Caption = "Testing fail"  
End If  
  
End Sub
```

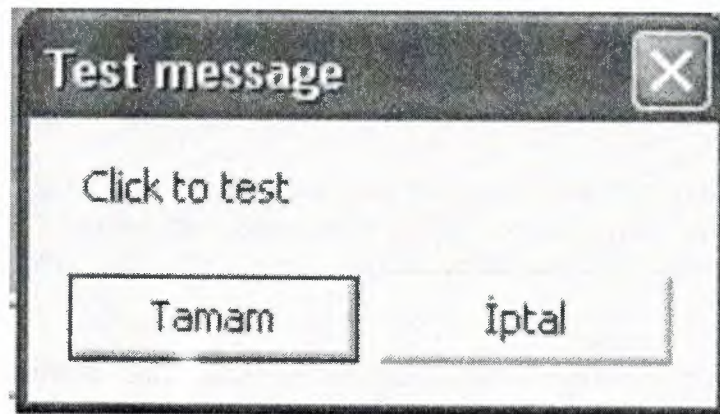


Figure 1.7. Showing **Msgbox()**

1.8.Declaring Array

We could use Public or Dim statement to declare an array just as the way we declare a single variable. The Public statement declares an array that can be used throughout an application while the Dim statement declare an array that could be used only in a local procedure. The general format to declare an array is as follow:

Dim arrayName(subs) as dataType

where subs indicates the last subscript in the array.

Example 13.1

Dim FindName(10) as String

1.9.Database Application

Finally, I want to put end point in that part, with telling about database application. Of course, All topics is not restricted what I emphasize in that part.

Visual basic allows us to manage databases created with different database program such as MS Access, Dbase, Paradox and etc. In this lesson, we are not dealing with how to create database files but we will see how we can access database files in the VB environment

1.9.1. Using ADO Control

We have learned to build VB database applications using data control. However, data control is not a very flexible tool as it could work only with limited kinds of data and must work strictly in the Visual Basic environment. To overcome these limitations, we can use a much more powerful data control in VB known as ADO control. ADO stands for ActiveX data objects. As ADO is ActiveX-based, it could work in different platforms (different computer systems) and different programming languages. Besides, it could access many different kinds of data such as data displayed in the Internet browsers, email text and even graphics other than the usual relational and non relational database information.

To be able to use ADO data control, you need to insert it into the toolbox. To do this, simply press Ctrl+T to open the components dialog box and select Microsoft ActiveX Data Control 6. After this, you can proceed to build your ADO-based VB database applications.

The following example will illustrate how to build a relatively powerful database application using ADO data control. First of all, name the new form as Student and change its caption to Book Tiles- ADO Application. Secondly, insert the ADO data control and name it as Course and change its caption to book. Next, insert the necessary labels, text boxes and command buttons. The runtime interface of this program is shown in the diagram below, it allows adding and deletion as well as updating and browsing of data.

The image shows a screenshot of a Visual Basic form titled "Form1". At the top left, there is an ADO Data Control with a label "Data1" and navigation buttons (back, forward, first, last). Below the data control, there are three text boxes with labels: "Number : 20010733", "Name : Metin", and "Surname : ULAŞ".

Figure1.8. Using Ado Control

Consequently, As mentioned before. All things are not limeted that, 'what I told in that project'. As you gues. Many things can be done using Visual Basic. You can animate, file manage, advance database managing...etc. I summarized here, whichs was used in that project. I hope that, What I told this part. Which will give opinion to you 'how i did that project'.

CHAPTER TWO: DATABASE STRUCTURE

2.1. Brief Information

General structure of database consists of one database and ten (10) tables. Which are User, Stock, Customer, Category, Producttype, Sales, Content, Loan, Payment, Refusal. I constructed program's database using Microsoft Access Database. Aspecially when I was establishing program's database, i paid my attention do not duplicate in my tables. More duplication means more memory and less speed. Also I studied with SQL(Structured Query Language), DAO and ADO commands. It is much usefull to combine program with database.

2.2. Tables Structure In Database

As I mentioned before I used twelve table with a Access Database. All tables are given below.

User database table includes information about student

USER.DB			
Field Name	Type	Size	Key
Username	String	10	*
Password	String	10	
perprocard	Boolean	True/False	
peradduser	Boolean	True/False	
percompreports	Boolean	True/False	
perchangeprice	Boolean	True/False	

Table 2.1. User Database Table

STOCK.DB			
Field Name	Type	Size	Key
stockcode	String	15	*
Type	String	10	*
stockname	String	30	
category	String	20	
quantity	Number	Long	

Table 2.2. Stock Database Table

CUSTOMER.DB			
Field Name	Type	Size	Key
Cuscode	String	10	*
Name	String	20	
Surname	String	20	
Tel	String	10	
Mobile	String	10	
Email	String	30	
Address	String	80	

Table 2.3. Customer Database Table

CATEGORY.DB			
Field Name	Type	Size	Key
Categoryname	String	20	*
Type	String	10	*

Table 2.4. Category Database Table

PRODUCTTYPE.DB			
Field Name	Type	Size	Key
type	String	10	*
rentprice	Number	Double	
saleprice	Number	Double	

Table 2.5. Producttype Database Table

SALES.DB			
Field Name	Type	Size	Key
Salecode	String	10	*
Cuscode	String	10	Foreign Key
Date	Date	Short Date	
Paytype	String	1	
Total	Number	Double	
Discount	Boolean	True/False	
Cancelsale	Boolean	True/False	
Username	String	10	Foreign Key

Table 2.6. Sales Database Table

CONTENT.DB			
Field Name	Type	Size	Key
contcode	String	3	*
salecode	String	10	*
stockcode	String	15	Foreign Key
type	String	10	Foreign Key
contenttype	String	1	
backdate	Date	Short Date	
quantity	Number	Long	
price	Number	Double	

Table 2.7. Content Database Table

LOAN.DB			
Field Name	Type	Size	Key
cuscode	String	10	* Foreign Key
salecode	String	10	* Foreign Key
amount	Number	Double	

Table 2.8. Loan Database Table

PAYMENT.DB			
Field Name	Type	Size	Key
Cuscode	String	10	Foreign Key
Date	Date	Short Date	
Amount	Number	Double	
username	String	10	Foreign Key

Table 2.9. Payment Database Table

REFUSAL.DB			
Field Name	Type	Size	Key
stockcode	String	15	Foreign Key
type	String	10	Foreign Key
quantity	Number	Long	
date	Date	Short Date	
change	Boolean	True / False	
money	Number	Double	
changestockcode	String	15	Foreign Key
changetype	String	10	Foreign Key
username	String	10	Foreign Key

Table 2.10. Refusal Database Table

2.3. Relation Between Tables

All tables was created in Microsoft Access Database. All relations was done from relation menu. There are ten (10) tables. As you know, that tables includes many key to make relation easily. Which keys are primary key, foreign key, komposit key. Some times, to do primary key from one column is not possible. that's why composite key can be selected. All keys are used to make relation between tables. Relations are between that keys. To see relation is little hard, but understand is very easy. Of course you can also see clearly in the figure below.

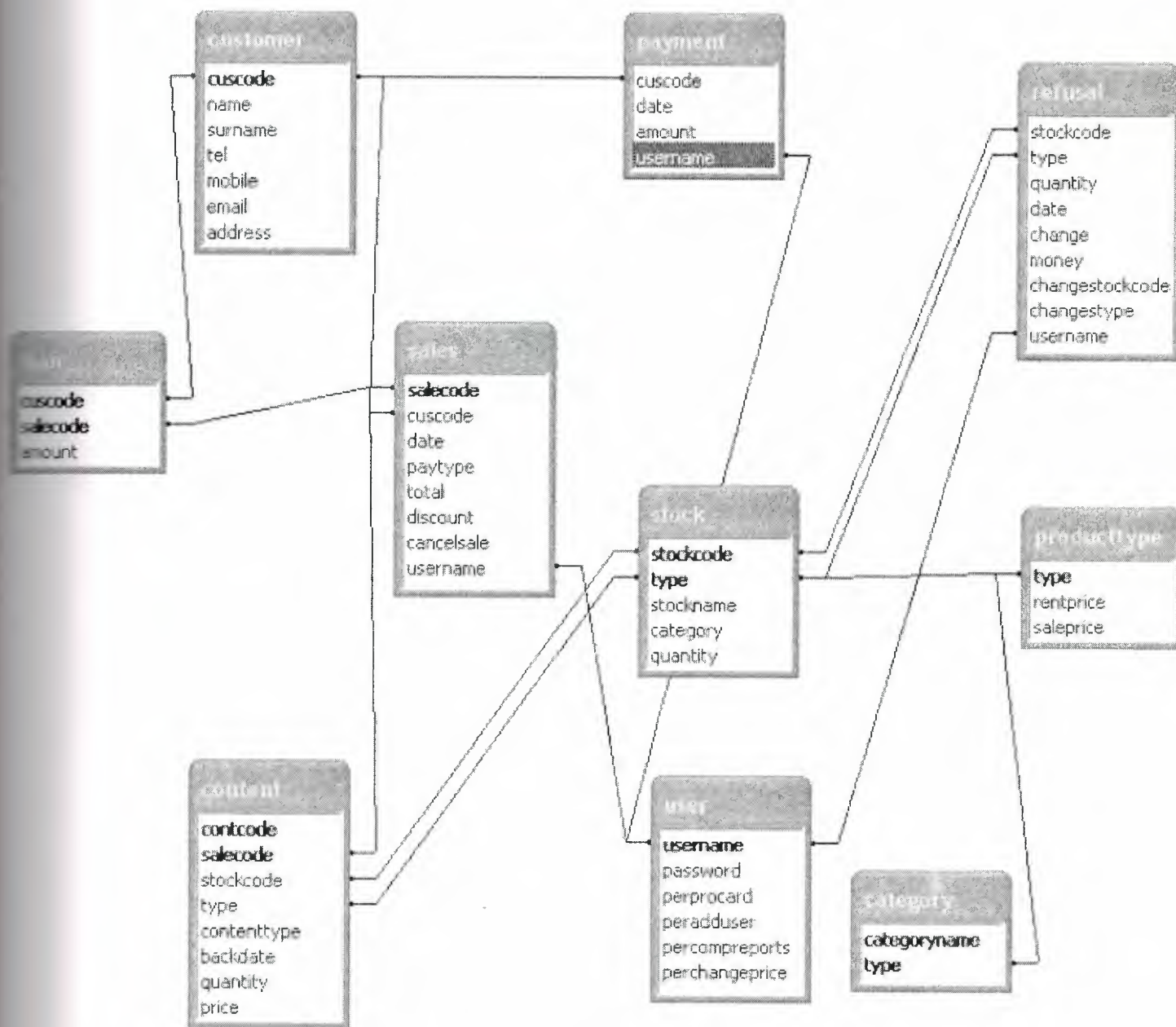


Figure 2.1. Relation Between Tables

As you saw above, all tables was done with using Microsoft Access Database. I will give brief information about Access and SQL(Structured Query Language). Of course, Using SQL in Access Database is possible. Also you will find more information about SQL.

2.4. Microsoft Access Database

2.4.1. Microsoft Access Database Fundamentals

Are you overwhelmed by the large quantities of data that need to be tracked in your organization? Perhaps you're currently using a paper filing system, text documents or a spreadsheet to keep track of your critical information. If you're searching for a more flexible data management system, a database might be just the salvation you're looking for.

What is a database? Quite simply, it's an organized collection of data. A database management system (DBMS) such as Access, FileMaker Pro, Oracle or SQL Server provides you with the software tools you need to organize that data in a flexible manner. It includes facilities to add, modify or delete data from the database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected contents.

Microsoft Access provides users with one of the simplest and most flexible DBMS solutions on the market today. Regular users of Microsoft products will enjoy the familiar Windows "look and feel" as well as the tight integration with other Microsoft Office family products. An abundance of wizards lessen the complexity of administrative tasks and the ever-present Microsoft Office Helper (you know... the paper clip!) is available for those who care to use it. Before purchasing Access, be sure that your system meets Microsoft's minimum system requirements. To further our discussion, let's first examine three of the major components of Access that most database users will encounter – tables, queries, forms. Once we've completed that we'll look at the added benefits of reports, web integration and SQL Server integration.

:

EMPLOYEE.DB			
Field Name	Type	Size	Key
Code	String	10	*
Name	String	20	
Surname	String	20	
DateOfBirth	Date	Short Date	
HireDate	Date	Short Date	
ExitDate	Date	Short Date	
Deptid	String	5	ForeingKey

Figure2.2. Sample Table

The table above contains the employee information for our organization characteristics like name, date of birth and title. Examine the construction of the table and you'll find that each column of the table corresponds to a specific employee characteristic (or **attribute** in database terms). Each row corresponds to one particular employee and contains his or her information. That's all there is to it! If it helps, think of each one of these tables as a spreadsheet-style listing of information.

In the previous section, we learned how tables allow us to create the framework for storing information in a database. Obviously, a database that only stored information would be useless -- we need methods to retrieve information as well. If you simply want to recall the information stored in a table, Microsoft Access allows you to open the table and scroll through the records contained within it. However, the real power of a database lies in its capabilities to answer more complex requests, or **queries**. Access queries provide the capability to combine data from multiple tables and place specific conditions on the data retrieved.

2.4.2. Creating Table

Many techniques allow you to create a database, the fastest of which consists of using one of the provided examples. Microsoft Access 97 shipped with 22 sample databases while Microsoft Access 2000 ships with 10. Furthermore, the 97 version allowed to provide sample data into the database. This is not available with the 2000 release. The databases that ship with Microsoft Access can help you in two main ways: they provide a fast means of creating a database and you can learn from their structure.

To create a database using one of the samples, there is a little detail to follow depending on whether you had launched the program already or not. If Microsoft Access is not running, you can start it. When the first dialog box comes up, you can click the second radio button: Access Database Wizard, Pages

The New dialog box displays two property pages labeled General and Databases. If you want to create a database based on one of the samples, you can click the Databases property page. A list of the sample databases appears. You can then choose one and click OK.

When creating a database using one of the samples, depending on the sample you selected, the Database Wizard will display a few objects and suggest some fields for your database. Some fields are already associated with the objects and some other fields can be added. You can examine them, then add some fields you think are important for your database. You will also have the option of selecting a design layout. Some of the sample databases have been configured to require information about the company you are creating the database for.

2.5. Introducing Database

Databases are designed to offer an organized mechanism for storing, managing and retrieving information. They do so through the use of tables. If you're familiar with spreadsheets like Microsoft Excel, you're probably already accustomed to storing data in tabular form. It's not much of a stretch to make the leap from spreadsheets to databases.

Just like Excel tables, database tables consist of columns and rows. Each column contains a different type of attribute and each row corresponds to a single record. For example, imagine that we were building a database table that contained names and telephone numbers. We'd probably set up columns named "FirstName", "LastName" and "TelephoneNumber." Then we'd simply start adding rows underneath those columns that contained the data we're planning to store.

At this point, you're probably asking yourself an obvious question – if a database is so much like a spreadsheet, why can't I just use a spreadsheet? Databases are actually much more powerful than spreadsheets in the way you're able to manipulate data. Here are just a few of the actions that you can perform on a database that would be difficult if not impossible to perform on a spreadsheet:

- Retrieve all records that match certain criteria
- Update records in bulk
- Cross-reference records in different tables
- Perform complex aggregate calculations

As we walk through this tutorial, you'll learn how you can use databases to achieve each of these objectives. Page 2 of this lesson provides you with an overview of how database keys can be used to uniquely identify records and form relationships between tables. Page 3 describes how the Structured Query Language allows you to interact with your database. On page 4, we examine the different types of databases available on the market today.

2.6. Database Keys

On the previous page of this article, you learned how databases use tables to organize data. As you probably recall, each table consists of a number of rows, each of which corresponds to a single database record. So, how do databases keep all of these records straight? It's through the use of **keys**.

The first type of key we'll discuss is the **primary key**. Every database table should have one or more columns designated as the primary key. The value this key holds should be unique for each record in the database. For example, assume we have a table called Employees that contains personnel information for every employee in our firm. We'd need to select an appropriate primary key that would uniquely identify each employee. Your first thought might be to use the employee's name.

This wouldn't work out very well because it's conceivable that you'd hire two employees with the same name. A better choice might be to use a unique employee ID number that you assign to each employee when they're hired. Some organizations choose to use Social Security Numbers (or similar government identifiers) for this task because each employee already has one and they're guaranteed to be unique. However, the use of Social Security Numbers for this purpose is highly controversial due to privacy concerns.

Once you decide upon a primary key and inform the database of this decision, it will enforce the uniqueness of the key. If you try to insert a record into a table with a primary key that duplicates an existing record, the insert will fail.

Most databases are also capable of generating their own primary keys. Microsoft Access, for example, may be configured to use the AutoNumber data type to assign a unique ID to each record in the table. While effective, this is a bad design practice because it leaves you with a meaningless value in each record in the table. Why not use that space to store something useful?

The other type of key that we'll discuss in this course is the **foreign key**. These keys are used to create relationships between tables. Natural relationships exist between tables in most database structures. Returning to our employees database, let's imagine that we wanted to add a table containing departmental information to the database. This new table might be called Departments and would contain a large amount of information about the department as a whole. We'd also want to include information about the employees in the department, but it would be redundant to have the same information in two tables (Employees and Departments). Instead, we can create a **relationship** between the two tables.

Let's assume that the Departments table uses the Department Name column as the primary key. To create a relationship between the two tables, we add a new column to the Employees table called Department. We then fill in the name of the department to which each employee belongs. We also inform the database that the Department column in the Employees table is a **foreign key** that references the Departments table. The database will then enforce *referential integrity* by ensuring that all of the values in the Departments column of the Employees table have corresponding entries in the Departmentstable.

Note that there is no uniqueness constraint for a foreign key. We may (and most likely do!) have more than one employee belonging to a single department. Similarly, there's no requirement that an entry in the Departments table have *any* corresponding entry in the Employees table. It is possible that we'd have a department with no employees.

2.7. Working with SQL

SQL (pronounced "ess-que-el") stands for structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingress, etc. although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "select", "insert", "delete", "create" and "drop" can be used to accomplish almost everything that one needs to do with a database.

2.7.1. Data Manipulation Language

The Data Manipulation Language (DML) is used to retrieve, insert and modify database information. These commands will be used by all database users during the routine operation of the database. Let's take a brief look at the basic DML commands:

The Data Manipulation Language (DML) is used to retrieve, insert and modify database information. These commands will be used by all database users during the routine operation of the database. Let's take a brief look at the basic DML commands:

2.7.1.1.INSERT

The INSERT command in SQL is used to add records to an existing table. Returning to the `personal_info` example from the previous section, let's imagine that our HR department needs to add a new employee to their database. They could use a command similar to the one shown below:

```
INSERT INTO Stock  
values('1','Titanic', 'DVD', 'Romance',5)
```

Note that there are four values specified for the record.

These correspond to the table attributes in the order they were defined: `first_name`, `last_name`, `employee_id`, and `salary`.

2.7.1.2.SELECT

The SELECT command is the most commonly used command in SQL. It allows database users to retrieve the specific information they desire from an operational database. Let's take a look at a few examples, again using the `personal_info` table from our employees database.

The command shown below retrieves all of the information contained within the `personal_info` table. Note that the asterisk is used as a wildcard in SQL. This literally means "Select everything from the `personal_info` table."

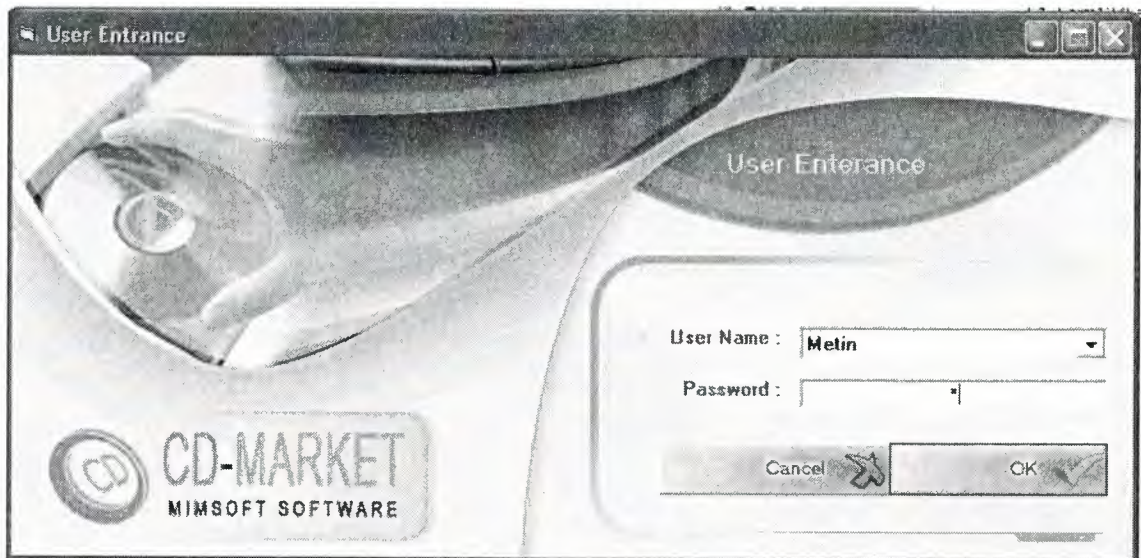
```
SELECT *  
FROM customer;
```

Alternatively, users may want to limit the attributes that are retrieved from the database. For example, the Human Resources department may require a list of the last names of all employees in the company. The following SQL command would retrieve only that information:

```
SELECT name,surname  
FROM customer WHERE cuscode='3';
```


CHAPTER THREE: STUDENT TRACKING SYSTEM

3.1. User Entrance



In this menu we enter User Name and Password, the username field is in combo type, we can select existing user, if we write a wrong password the program alert us and gives a message. After we press on Ok the programs enter to Main Menu.

3.2. Main Page

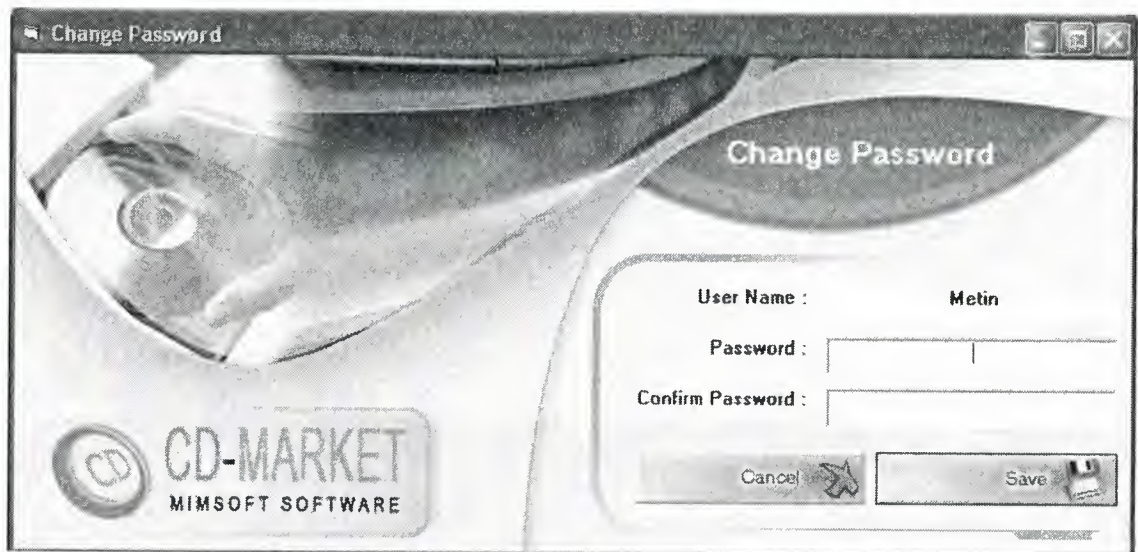


This is the Main Page of our program, through this page we do all the operations necessary.

This page has menu and it's sub menu : Program (Change User, Exit), Users (Change Password, Add User / Change Details), Cards (Product Cards, Customer Cards), Operations (Rent / Sale Product, Take Product, Refund Product, Cancel Last Sale), Accounts (Customer Accounts Information), Reports (Company Reports, Product Reports, Top 10 List) , Settings (Add / Change Product Type, Categories) About (About Program).

Let's firstly describe the buttons of the Main Page;

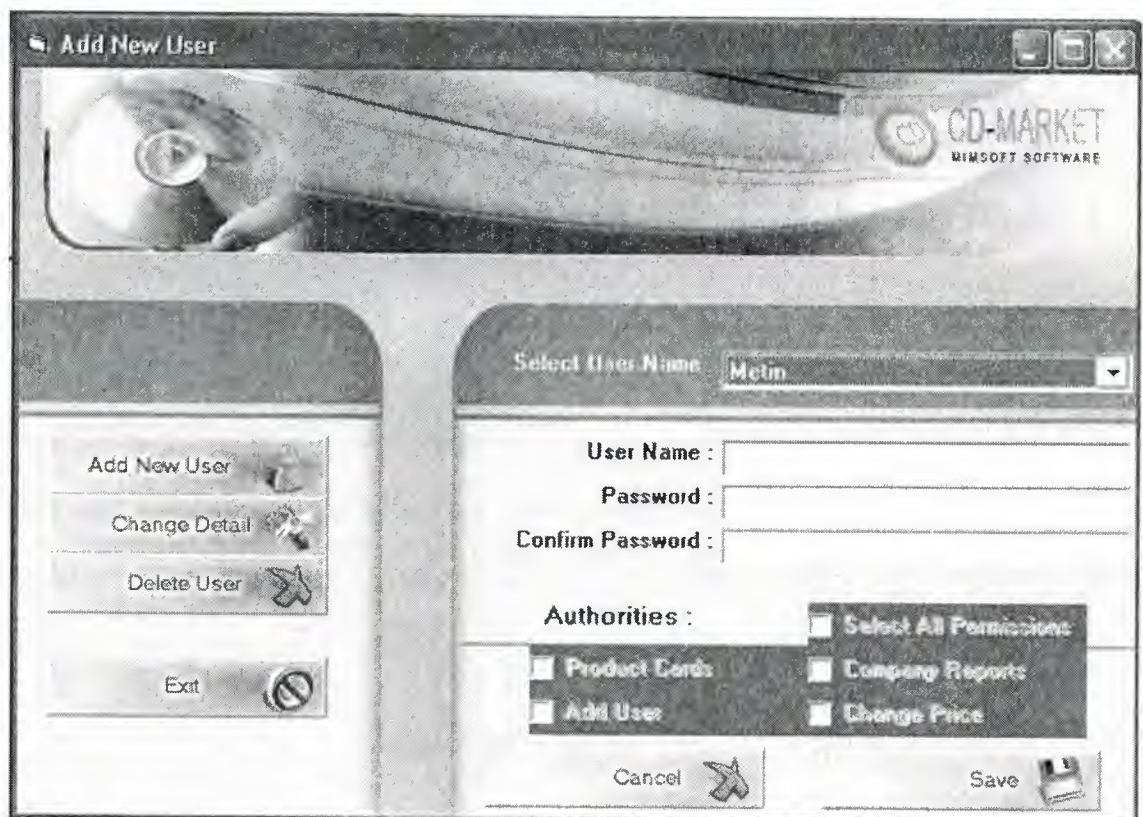
3.2.1. Change Password



The screenshot shows a window titled "Change Password" with a CD-ROM background. On the left is the "CD-MARKET MIMSOFT SOFTWARE" logo. On the right, there are input fields for "User Name" (containing "Metin"), "Password", and "Confirm Password". At the bottom right are "Cancel" and "Save" buttons with icons.

In This page we can change password of the user who entered to the program, we can't change other users password.

3.2.2. Add new user or change details



The screenshot shows a window titled "Add New User" with a CD-ROM background. On the left is the "CD-MARKET MIMSOFT SOFTWARE" logo. On the right, there is a "Select User Name" dropdown menu with "Metin" selected. Below it are input fields for "User Name", "Password", and "Confirm Password". Further down is an "Authorities" section with a grid of checkboxes: "Select All Permissions", "Product Cards", "Add User", "Company Reports", and "Change Price". At the bottom right are "Cancel" and "Save" buttons with icons.

Through this page we can add new user to our program, while adding new user we can give special characteristics to the defined user, for instance the Authorities control could be given if he/she could change price or not, add user or not...etc.

We can Change Details of the existence user, that's by selecting the user from Select User Name and then clicking on Change Detail. Also we can remove any user we want. So after adding, removing or making changes we press on Save to save the operation done or Cancel to Exit without saving.

3.2.3 Customer Card

The screenshot shows the 'Customer Cards' application window. It features a title bar with the text 'Customer Cards'. The main area is divided into three panels. The left panel, titled 'Operations', contains three buttons: 'Add Customer', 'Change Detail', and 'Delete'. The middle panel, titled 'Customer Informations', contains a 'Customer Code' field, a 'Find by Details' button, and a 'Find' button. Below these are input fields for 'Name', 'Surname', 'Phone', 'Mobile Phone', 'E - Mail', and 'Address'. The right panel, titled 'Picture', contains a 'Delete Picture' button, a placeholder image with the text 'No Picture', and an 'Add Picture' button. At the bottom of the window are four buttons: 'Exit', 'Cancel', 'Save', and 'Add Picture'.

Through this page we can add new customer, the program gives automatic code to every new customer, after we add all the details of the desired customer we press Save to store it, also we can add picture to any customer, delete selected customer and change details too.

3.2.3.1. Find Customer By Details

Code	Name	Surname	Phone
	Metin	ULAŞ	01224) 513 65 97

In this page we can find customers by name or by surname, we can write just the first letter of name or surname, the program lists all the customers begins with that letter, after we press on Find it lists below. We can return to Main Menu by pressing on OK

3.2.4. Product Card

The screenshot shows the 'Product Cards' application window. The window is divided into three main sections: 'Operations', 'Product Informations', and 'Picture'. The 'Operations' section on the left contains buttons for 'Add New', 'Change Detail', 'Delete', 'Add Product Type', 'Add Category', and 'Exit'. The 'Product Informations' section in the center contains a 'Product Type' dropdown menu set to 'DivX', a 'Code / Barcode' input field, and 'Find by Details' and 'Find' buttons. Below these are input fields for 'Code / Barcode', 'Name', 'Product Type', 'Category', and 'Quantity', followed by an 'Add Quantity' button. At the bottom are 'Cancel' and 'Save' buttons. The 'Picture' section on the right shows a large 'No Poster' message with a crossed-out circle icon, and buttons for 'Add Picture' and 'Delete Picture'.

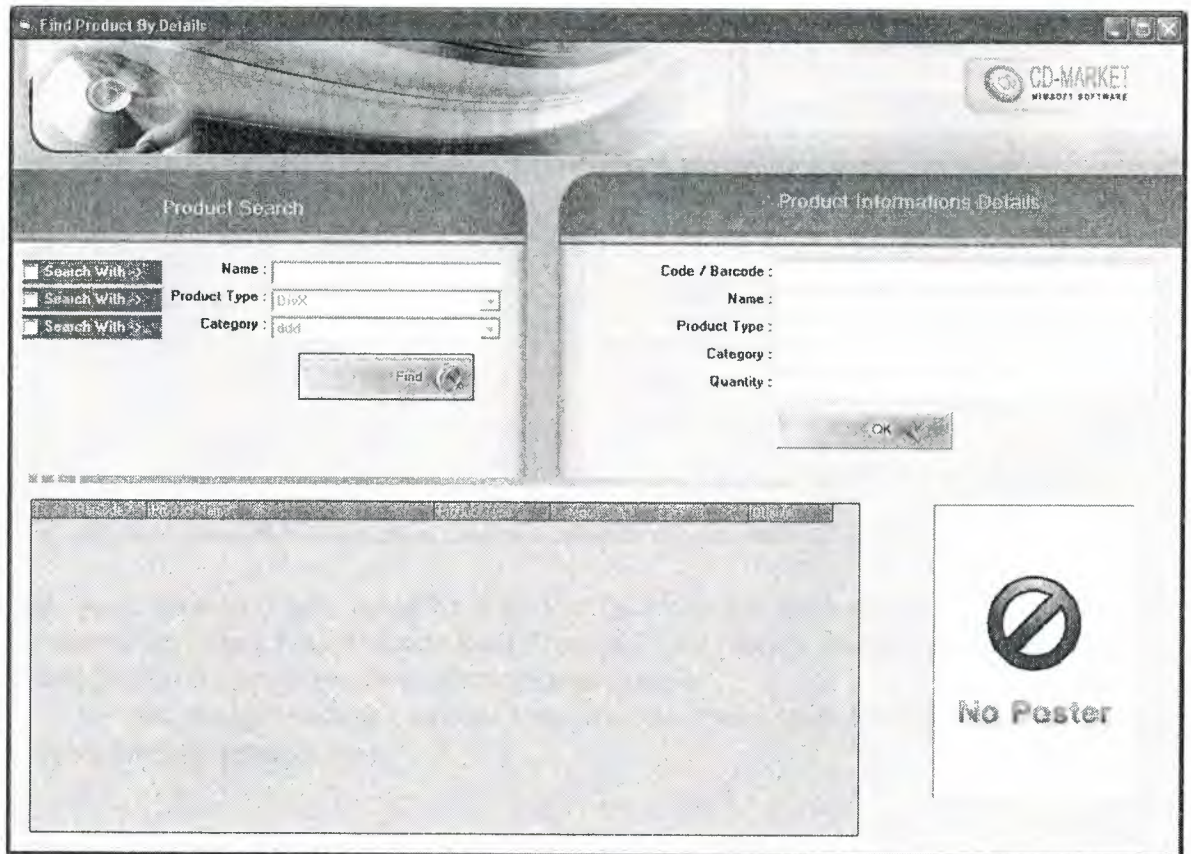
In This page we can Add items to our Stock, in Product Information Field we have a search engine, we choose Product type first, then it's code and we press on find, the program show's us the desired item with it's picture if it is exist otherwise it gives us a message that we don't have this item.

To add an item we can give a code manually otherwise the program gives an automatic code to the new item, after we write code, name and we choose product type and category and we write quantity we press on Save Button to save the changes made, we can add quantity to the existence product by pressing on Add Quantity button.

We can Add or Delete pictures to any product we have, either new or existence, also we can change details of any product.

In Operation Field we have Add Product Type and Add Category, we are going to explain them in other pages in detail.

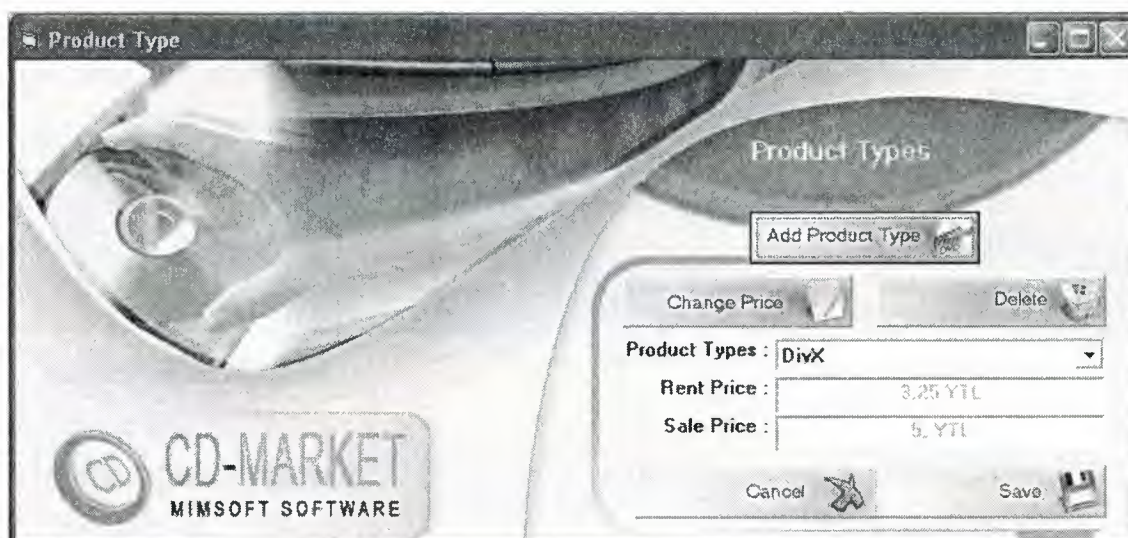
3.2.4.1. Find Product By Details



In the page we have Product Search And Product Details Fields, In Product Search field we can make a search by name, by product type or by category, we can select to options in the same time, for instance we could say search by name the films start with letter M and Product Type DVD, after we press on Find the program lists the found products below.

If we click on any selected item on found list we see that all the details of that items written on Product Information Field, if we press on Ok the desired items goes to the page that we came from.

3.2.4.2. Product Types



In this page product Types could be added to the program, for instance we add DVD type in Product Types filed, Price in Rent Price and Sale Price field, later when we press on Save Button the program saves all the changes made.

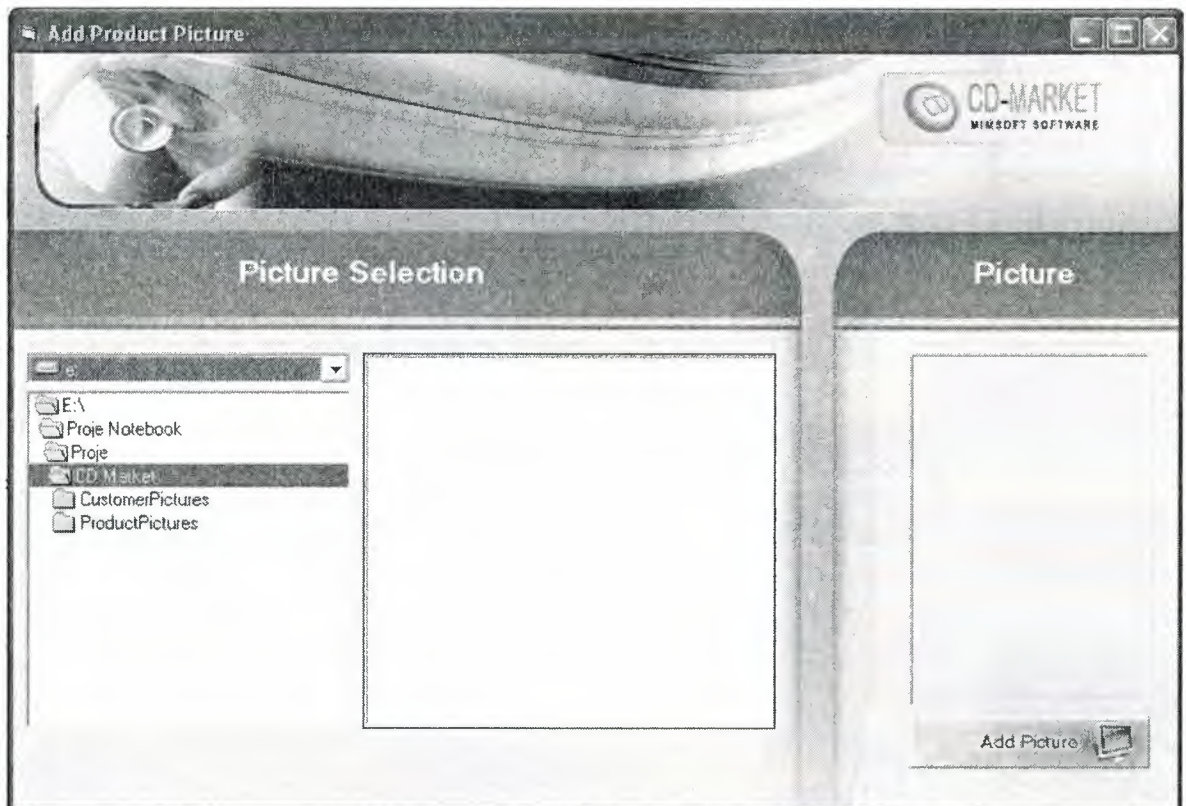
we can change existance product type price by Pressing on Changing Price, also we can delete any product type.

3.2.4.3. Product Category



In this Page we can Add Category and Delete any Desired Category. If we want to add a category we have to select Product Type First, because the Category will be added to that field. We press on Save to save the changes or Cancel to exit from this form and go to Main Menu

3.2.5. Add Product or Customer Picture



In this page we can Add any picture to any product or customer. That's by selecting from the desired drive, then selecting the picture then add picture. But the file list box has only jpeg format pictures (*.jpg). so if we want a picture it is not to be bmp or other picture format. I do this, cause; jpeg format is need smaller disk space than other format.

3.2.6. Rent or Sale Product

Customer Code : Find

Product Type : DVD Product Code : Find

Customer Code : 1 Find by Details

Name : Metin Surname : ULAŞ Show Customer Details

Code / Barcode : Name : Product Type : Category : Stock Quantity : Sale Price :

Sale Quantity : Sale Product Rent Product

Clear Sale List Remove Item Add to List

Code	Name	Type	Status	Quantity	Total
Malix	DVD	Sale	7, YTL	2	14, YTL

Discount Total : 14, YTL

Discount Amount : 1, YTL General Total : 13, YTL

Taken Cash : 3, YTL

Adding to Customer Loan : 10, YTL

Do It

This is sale page, one of the most important pages of the program, in this page we can make a sale operation, firstly we have to select a customer, that's through Customer Code field and pressing on Find, or we can find by pressing on Find By Details button, we can see the selected customer's Details too.

Then we add product to the selected customer, if we rent the desired product we have to select customer, if the customer payes some amount of the payment we have to select customer too, but if the customer payes the payment in cash we don't have to select a customer.

Then we choose product from and we decide if we want to make a sale or rent operation, then we decide again if the customer pay in cash, add to customer loan or payes by credit.

Sometimes we see that after we add some item to the sale list the customer changes his mind and wants to remove that item, at then we select the desired item and press on Remove Item. We can clear sale list too by pressing on Clear Sale List button.

The program show's us the total amount of money made in that sale operation, we can make discount too, that's by pressing on discount button and writting discount Amount then after we press on DO IT, the program save all the changes and return to the main menu.

3.2.7. Show Customer Information Details

Customer Details

CD-MARKET
WINSOFT SOFTWARE

Customer Code: 1

Loans : 14, YTL **Payment :** 0,00 YTL

Total Loan: 14, YTL

Sale Total : 17, YTL

Name : Metin
Surname : ULAŞ
Phone : 0 (224) 513 65 87
Mobile Phone :
E - Mail :
Address :

Rented Products

Product	Format	Genre	Due Date
Kısa	DVD	Action	01.06.2006 (Not Back Yet)

Bought Products

Product	Format	Genre	Quantity	Due Date
Metin	DVD	Action	2	01.06.2006

HACKERS
BIRD UP BY MURPHY
TALAN - TIGLİM

In this page we can see the selected customer information details, that's by finding the customer by customer code first, then we see all the details of that customer shown with his/her picture.

After we choose the customer, we can see his/her Loans, payments, total loans and Total sales that customer make.

Also the Rented details and Bought details that customer made could be seen in this page.

3.2.8. Take Product

The screenshot shows a Windows application window titled 'Form13' with a 'CD-MARKET' logo in the top right corner. The interface is divided into two main sections: 'Searching Rented Products' on the left and 'Rental Information Details and Taking Product' on the right.

Searching Rented Products:

- Customer Code: [Text Box]
- Find by Details [Button] Find [Button]
- Product Type: DivX [Dropdown]
- Product Code: [Text Box]
- Find by Details [Button] Find [Button]

Rental Information Details and Taking Product:

Customer Code :	1	Show Customer Details [Button]
Name :	Metin	
Surname :	ULAŞ	
Code / Barcode :	2	Rented Date :
Name :	Kpax	01.06.2006
Product Type :	DVD	Take Product [Button]
Category :	Action	

Below these sections is a table with columns: Customer Code, Customer Name, Surname, Product Code, Product Name, Price, Rented Date, and Return Date. The table body is currently empty.

In this page we take back the rented product. In searching Rented Product feild we can find the customer that want's to give back the rented product, that's by writting customer code if we know, otherwise we can find by Find by Details too.

Also we can find by product type or product code, it this type of search engine the program show's us the selected product type that is in rent, anyway after we find the desired product and we click on it in the below list the details will bewritten in the above right part of the form, when we press on Take Product, the product will be taken back form the selected customer.

3.2.9. Product Refund

Product Refusal

CD-MARKET
MINDDT SOFTWARE

Customer Searching and Informations

Product Type :

Product Code :

Find by Details

Code / Barcode :

Name :

Product Type :

Category :

Stock Quantity :

Refund Details

☒ Pay Back Price To Customer
☐ Change Product With Difference Product

Product Price :

Refusal Quantity :

Total :

Pay Back Amount :

In this page the customer can refund any product, firstly we search the customer who wants to refund, also product type that he/she want's to refund with it's code, Later in Refund Details field we have to decide if we pay back the amount of money to the selected customer or we are going to change with something else.

Then we write Product price, quantity, the program gives us the total and we decide to pay back the pay back amount, then after we press on Ok the program save the changes and returns to the main menu.

3.2.10. Cancel Last Sale

The screenshot shows a window titled "Sale Information" with a header image of a hand holding a coin. The window is divided into three main sections:

- User and Customer Informations:**
 - User : a
 - Customer Code : 0
 - Customer Name : No Defined Customer
 - Surname : No Defined Customer
- Sale Information Details:**
 - Content Date : 01.06.2006
 - Discount : No
 - Sale Total : 21, YTL
 - Payment Type : Credit Card
 - Taken Cash : 0,00 YTL
 - Loan Amount : 0,00 YTL
- Sale Content Informations:**

Product Code	Product Name	Unit	Quantity	Price	Unit Price	Quantity	Price
Km Bill	DivX	Action	7, YTL Sold	3	21, YTL		

At the bottom right, there are two buttons: "Cancel Sale" and "Exit".

In this page the program show's the last sale made, it show's the user that made the last sale with his/her detail, date, discount if made, sale total, payment type, if taken in cash or not and loan amount.

If we press on cancle sale, the program cancel's all the operation's made for that sale or Exit to exit and return to Main Menu.

3.2.11. Customer Accounts

The screenshot displays the 'Customer Accounts' window of the CD-Market software. The interface is divided into two main sections: 'Searching Customer and Informations' on the left and 'Adding payment and Loan Informations' on the right.

Searching Customer and Informations:

- Customer Code:** A text field containing '1'.
- Name:** A text field containing 'Metin'.
- Surname:** A text field containing 'ULAŞ'.
- Find by Details:** A button with a magnifying glass icon.
- Find:** A button with a magnifying glass icon.
- Options:** Two checkboxes are present: 'Do Not Show Canceled Sales' (unchecked) and 'Show Sales Only With Loan' (checked).
- Table:** A table with one row of data: '01.06.2006 Added To Customer Loan YES 15, YTL NO'.
- Total Of Sales:** A label showing '15, YTL'.

Adding payment and Loan Informations:

- Loans:** A label showing '14, YTL'.
- Total Loan:** A label showing '12, YTL'.
- Add Payment:** A button with a magnifying glass icon.
- Payment Amount:** A text field containing '8, YTL'.
- OK:** A button with a checkmark icon.
- All Payments:** A table with one row of data: '01.06.2006 2, YTL'.
- Total:** A label showing '2, YTL'.

In this page we investigate the customer account's information, firstly we have to tick one of the options of both of them, if we want to show canceled sale for that customer we tick the first option, if we want to see the sales done only with loan we tick the second option, we can tick both of them too, so we see all teh canceled and loadn sales.

Later we select a customer in Searching Customer and Informations Feild, so the program show's the customers in list, now we can add payment by Pressing on Add payment button, then Ok to return to main menu and save the changes.

3.2.12. Company Reports

Description	Status	Date	Amount			
Added To Customer Account	YES	01.06.2006	15, YTL	NO		
Cash	NO	01.06.2006	14, YTL	NO		
Cash	NO	01.06.2006	14, YTL	NO		
Cash	YES	01.06.2006	12, YTL	NO		
Credit Card	NO	01.06.2006	21, YTL	NO		

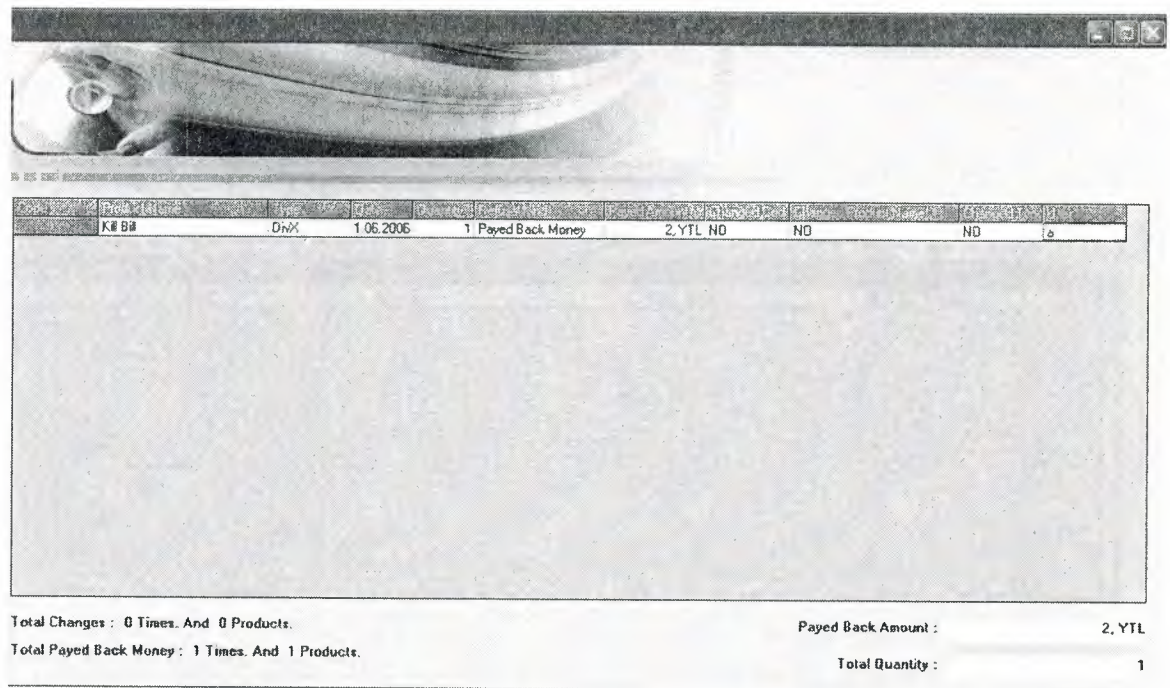
In this page the program queries company information. In selection of Report Details, if we select a user the program show's all the sale's that user made with it's detail in Report Result part.

If we choose show today's repot, the program show's the sales that user made in the present date.

If we choose Show Between Two Dates Report, we have to assign two date and see the sale operation's that user made between the selected dates.

If we choose Show All Report's the program show's All the sale's that customer made .

3.2.12.1. Product Refund Details



Product Code	Product Name	Refund Date	Quantity	Refund Kind	Payed Amount	Changed Product Name	Changed Product Type	Changed Product User	Total
K&B	DnX	1.06.2006	1	Payed Back Money	2, YTL	NO	NO	NO	1

Total Changes : 0 Times. And 0 Products.

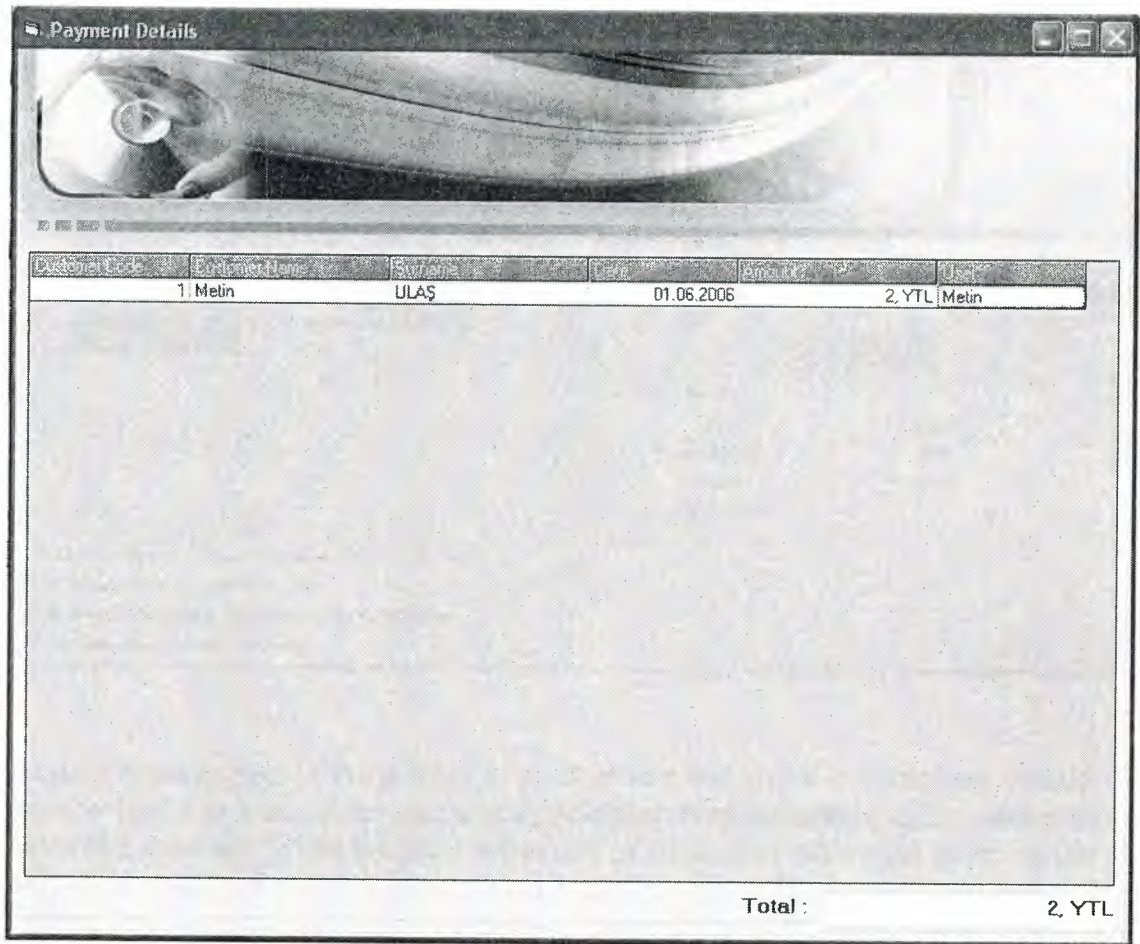
Total Payed Back Money : 1 Times. And 1 Products.

Payed Back Amount : 2, YTL

Total Quantity : 1

In this page we can see all the refunded item's to teh company, the program show's code of refunded item, product name, type, date of refund, quantity, Refund kind, payed amount, if the customer changed the refunded item with something else or not then changed product name, changed product type and the user that made that operation.

3.2.12.2. Payment Details



The image shows a software window titled "Payment Details". At the top, there is a header bar with the title and standard window controls. Below the header, there is a large, empty rectangular area, possibly for a receipt or image. Below this area, there is a table with the following data:

Customer Code	Customer Name	Surname	Date	Amount	User
1	Metin	ULAS	01.06.2006	2, YTL	Metin

At the bottom right of the window, there is a label "Total :" followed by the value "2, YTL".

In this page we can see all the payments made to the company, the program shows the customer code, customer name, surname, date that he/ she made the payment, payment amount and the user that took the payment operation.

3.2.13. Product Reports

Product Reports

CD-MARKET
MUSOFT SOFTWARE

Selection of Report Details

☒ Today
☐ Between Two Dates
☐ Show All

Show

This Product Rented : 0 Times. And Earned : 0, YTL From Rental.
And 0 Products still not return back...
Sale Quantity : 7 Products. And Earned : 49, YTL From Sales.
Total Earns : 49, YTL From This Product.

Searching Product

Product Type : DVD
Product Code :
Find

Find by Details

Code / Barcode : 1
Name : Matrix
Product Type : DVD
Category : Action
Stock Quantity : -1

Product Reports page is the display of product sale and rental informations. Firstly we have to find a product. After, there is three option (radio) buttons today, between two dates and show all. When we must select one of them after must click show button. It will show us to how many product rented in that date (which is we selected with option button) and how much money we earned from this product from rental. Below that string it will show us how many product sold and how muc money earned from this product from sales. And after this string it will show us to totally how much money earned from this product.

3.2.14. Top 10 List

Top 10 List

CD-MARKET
MIMSOFT SOFTWARE

Top 10 List

☒ Show All
☐ Between Two Dates

Show

Top 10 Rental Product List

NO	Product Code	Product Name	Type	Category	Rental Amount
1	2	Kpax	DVD	Action	1

Top 10 Sale Product List

NO	Product Code	Product Name	Type	Category	Sale Amount
1	1	Matrix	DVD	Action	7
2	4	Buzul Cagi	DVD	Animation	2
3	3	lkdsfllkj	DVD	Action	2

This page show's us top 10 item that sale most or rented most in the company.

Firstly we have to make a decision between Show All and Between two dates, if we choose show All, the company list's 10 product in Rent list and sale List that they were in top for rent and sale, if we choose between two dates option, we have to select two dates and see the top 10 products in sale and Rent list.

Conclusion

My Project program is a stock tracking program, it is useful for CD Market's, it's name is CD-MARKET program I used this program, because I'm planning to open a CD-Market in the future, now I have hundred's of films, it's my hobby to collect CD's, I have hundred's of programs, games and music's too. Now I already get benefit from my program, because before I build this I get difficulty to find a CD, but now I can find it easy by using of search engines in the program.

Through building of this project I learned many things, because until now I was just learning programs literary or I was doing small programs, but by the way of this program I investigated a lot in internet, I read many books, I tried many examples in order o improve my knowledge about Visual basic to make my program a useful thing.

There are advantages and disadvantages of Visual basic, the advantage of it is easy in use and gives message when a wrong code is written and some other advantages, the disadvantage of Visual basic is slow and don't have every characteristics, especially we can not change colors of writing on buttons.

The Program can be updated in the future, we can add other properties to it, and we can use it through internet by adding extra codes, while doing the project wonderful imaginations come to my mind but if I applied them I couldn't be able to finish this project in time, but I'll do my imaginations project in the future if God Wishes.

My Project program is a tracking system program, I used Mobile Tracking System, because we are already a wholesaler of Mobile and it's accessories in North of Iraq, we'll get benefit from this program too.

REFERENCES

Book References

- H.M.Deitel, P.J.Deitel and T.R.Nieto, Visual Basic 6: How To Program, Prentice Hall,Inc. Upper Saddle River., New Jersey, 1999
- Ihsan Karagülle and Zeydin Pala, Microsoft Visual Basic 6.0 Pro, Türkmen Printing House, Istanbul ,2001.

Internet References

- <http://www.vbtürk.com/allcodes/capture.htm>
- <http://www.programlama.com>
- <http://www.vbtutor.net/lesson.html>
- www.vbtutorial.com

APPENDICES

FORM - 1

'mesaj degiskenleri
Public yetkiyok As String
'mesaj degiskenleri sonu
'veritabani degiskernelri
Public db As Database
Public category As Recordset
Public content As Recordset
Public customer As Recordset
Public loan As Recordset
Public payment As Recordset
Public producttype As Recordset
Public refusal As Recordset
Public sales As Recordset
Public stock As Recordset
Public user As Recordset
'veritabani degiskenleri sonu
'program directory bilgi degiskeni
Public programyer As String
'program directory bilgi degiskeni sonu

'kullanici degiskenleri
Public username As String
Public password As String
Public izinkulekle As Boolean
Public izinrapor As Boolean
Public izinfiyat As Boolean
Public izinurunkart As Boolean
'kullanici degiskenleri sonu

'FONKSIYONLAR

'Fonksiyon degiskenleri
Public rakamkontrolgelen As String
'fonksiyon degiskenleri sonu

'fonksiyon donen deger degiskenleri
Public rakamkontrolsonucharf As Boolean
'fonksiyon donen deger degiskenleri sonu
Public Sub rakamkontrol()
Dim harf As String
Dim i As Integer
rakamkontrolsonucharf = False
For i = 1 To Len(rakamkontrolgelen)
harf = Mid(rakamkontrolgelen, i, 1)
If Asc(harf) > 57 Or Asc(harf) < 48 Then

```

        rakamkontrolsonucharf = True
    End If
Next
End Sub
Private Sub mesajlar()
yetkiyok = "You have no permission to enter this menu..! Contact with your manager."
End Sub
Public Sub tabloac()
Set db = OpenDatabase(programyer + "\db.mdb",False,False,";pwd=metin20010733")
Set category = db.OpenRecordset("category")
Set content = db.OpenRecordset("content")
Set customer = db.OpenRecordset("customer")
Set loan = db.OpenRecordset("loan")
Set payment = db.OpenRecordset("payment")
Set producttype = db.OpenRecordset("producttype")
Set refusal = db.OpenRecordset("refusal")
Set sales = db.OpenRecordset("sales")
Set stock = db.OpenRecordset("stock")
Set user = db.OpenRecordset("user")
End Sub
Public Sub tablokapat()
category.Close
content.Close
customer.Close
loan.Close
payment.Close
producttype.Close
refusal.Close
sales.Close
stock.Close
user.Close
db.Close
End Sub
Public Sub form1temizle()
Combo1.Clear
Text1 = ""
Command2.Enabled = False
Form1.izinfiyat = False
Form1.izinkulekle = False
Form1.izinrapor = False
Form1.izinurunkart = False
Form1.username = ""
Form1.password = ""
End Sub
Public Sub form1combo1doldur()
Combo1.Clear
If user.RecordCount > 0 Then
    user.MoveFirst
    While Not user.EOF
        Combo1.AddItem user.Fields("username")
    
```



```

        user.MoveNext
    Wend
    Combo1.ListIndex = 0
End If
End Sub
'FONKSIYONLAR SONU
Private Sub Command1_Click()
    Call tablokapat
End
End Sub

Private Sub Command2_Click()
    Form1.user.Index = "primarykey"
    Form1.user.Seek "=", Combo1.Text
    If Form1.user.Fields("password") = Text1 Then
        username = Combo1.Text
        password = Form1.user.Fields("password")
        Form1.izinfiyat = Form1.user.Fields("perchangeprice")
        Form1.izinkulekle = Form1.user.Fields("peradduser")
        Form1.izinrapor = Form1.user.Fields("percompreports")
        Form1.izinurunkart = Form1.user.Fields("perprocard")
        Form2.Show
        Form1.Hide
    Else
        MsgBox ("Wrong Password..!")
        Text1.SetFocus
    End If
End Sub

Private Sub Form_Load()
    programyer = CurDir
    Call form1temizle
    Call tabloac
    Call mesajlar
    If Form1.user.RecordCount > 0 Then
        Call form1combo1doldur
    Else

        Form4.form4gelenform = 1
        Form4.Show
        Form1.Hide
    End If
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Call tablokapat
End Sub

Private Sub Text1_Change()
    If Text1 = "" Then
        Command2.Enabled = False
    Else

```

```
Command2.Enabled = True
End If
End Sub
```

```
Private Sub Text1_GotFocus()
Text1.SelStart = 0
Text1.SelLength = Len(Text1)
End Sub
```

FORM – 2

```
Dim uzerinde As Boolean

Dim ayrilankontrol As Integer
```

```
Private Sub butonresim()

If ayrilankontrol = 1 Then

Command1.Picture = Image3.Picture


End If

If ayrilankontrol = 2 Then

Command2.Picture = Image3.Picture

End If

If ayrilankontrol = 3 Then

Command3.Picture = Image3.Picture

End If

If ayrilankontrol = 4 Then

Command4.Picture = Image3.Picture

End If

If ayrilankontrol = 5 Then

Command5.Picture = Image3.Picture
```

End If

If ayirilankontrol = 6 Then

Command6.Picture = Image3.Picture

End If

If ayirilankontrol = 7 Then

Command7.Picture = Image3.Picture

End If

If ayirilankontrol = 8 Then

Command8.Picture = Image3.Picture

End If

uzerinde = False

ayirilankontrol = 0

End Sub

Private Sub Command1_Click()

Call mnrentsaleproduct_Click

End Sub

Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Call butonresim

If uzerinde = False Then

Image3.Picture = Command1.Picture

uzerinde = True

ayirilankontrol = 1

End If

Command1.Picture = Command1.DownPicture

End Sub

Private Sub Command2_Click()

Call mntakeproduct_Click

End Sub

Private Sub Command2_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Call butonresim

If uzerinde = False Then

Image3.Picture = Command2.Picture

uzerinde = True

ayrilankontrol = 2

End If

Command2.Picture = Command2.DownPicture

End Sub

Private Sub Command3_Click()

Call mnshowloaninformation_Click

End Sub

Private Sub Command3_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Call butonresim

If uzerinde = False Then

Image3.Picture = Command3.Picture

uzerinde = True

ayrilankontrol = 3

End If

Command3.Picture = Command3.DownPicture

End Sub

Private Sub Command4_Click()

Call mncustomercards_Click

End Sub

Private Sub Command4_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Call butonresim

If uzerinde = False Then

Image3.Picture = Command4.Picture

uzerinde = True

ayrilankontrol = 4

End If

Command4.Picture = Command4.DownPicture

End Sub

Private Sub Command5_Click()

Call mnproductcards_Click

End Sub

```
Private Sub Command5_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
Call butonresim
```

```
If uzerinde = False Then
```

```
Image3.Picture = Command5.Picture
```

```
uzerinde = True
```

```
ayrilankontrol = 5
```

```
End If
```

```
Command5.Picture = Command5.DownPicture
```

```
End Sub
```

```
Private Sub Command6_Click()
```

```
Call mnchangepassword_Click
```

```
End Sub
```

```
Private Sub Command6_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
Call butonresim
```

```
If uzerinde = False Then
```

```
Image3.Picture = Command6.Picture
```

```
uzerinde = True
```

```
ayrilankontrol = 6
```

```
End If
```

```
Command6.Picture = Command6.DownPicture
```

```
End Sub
```



```
Private Sub Command7_Click()
```

```
Call mnchangeuser_Click
```

```
End Sub
```

```
Private Sub Command7_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)
```

```
Call butonresim
```

```
If uzerinde = False Then
```

```
Image3.Picture = Command7.Picture
```

```
uzerinde = True
```

```
ayrilankontrol = 7
```

```
End If
```

```
Command7.Picture = Command7.DownPicture
```

```
End Sub
```

```
Private Sub Command8_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub Command8_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)
```

```
Call butonresim
```

```
If uzerinde = False Then
```

```
Image3.Picture = Command8.Picture
```

```
uzerinde = True
```

ayrilankontrol = 8

End If

Command8.Picture = Command8.DownPicture

End Sub

Private Sub Form_Load()

Image3.Visible = False

End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Call butonresim

End Sub

Private Sub Form_Unload(Cancel As Integer)

Form1.tablokapat

End

End Sub

Private Sub Image1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)

Call butonresim

End Sub

Private Sub mnaddcategory_Click()

If Form1.producttype.RecordCount > 0 Then

Form7.Show

Form7.form7gelenform = 2

Form2.Enabled = False

Else

MsgBox ("Firstly you must enter product type to add category")

End If

End Sub

Private Sub mnaddnewuser_Click()

If Form1.izinkulekle = True Then

Form4.Show

Form2.Enabled = False

Form4.form4gelenform = 2

Else

MsgBox (Form1.yetkiyok)

End If

End Sub

Private Sub mncancellassale_Click()

If Form1.sales.RecordCount > 0 Then

Form14.form14gelenform = 2

Form14.Show

Form2.Enabled = False

Else

MsgBox ("No any sale record in database..!")

End If

End Sub

Private Sub mnchangepassword_Click()

Form3.Show

Form2.Enabled = False

End Sub

Private Sub mnchangeprice_Click()

Form6.form6gelenform = 2

Form6.Show

Form2.Enabled = False

End Sub

Private Sub mnchangeuser_Click()

Call Form1.form1temizle

Call Form1.form1combo1doldur

Form1.username = ""

Form1.password = ""

Form1.izinfiyat = False

Form1.izinkulekle = False

Form1.izinrapor = False

Form1.izinurunkart = False

Form1.Show

Form2.Hide

End Sub

Private Sub mncompanyreports_Click()

Form19.Show

Form2.Hide

End Sub

Private Sub mncustomercards_Click()

Form10.Show

Form2.Enabled = False

End Sub

Private Sub mnexit_Click()

Unload Me

End Sub

Private Sub mnproductcards_Click()

Form5.Show

Form2.Enabled = False

End Sub

Private Sub mnrentaproduct_Click()

Form11.Show

Form2.Hide

End Sub

Private Sub mnproductrefusal_Click()

Form17.Show

Form2.Enabled = False

End Sub

Private Sub mnproductreport_Click()

Form20.Show

Form2.Enabled = False

End Sub

Private Sub mnrentsaleproduct_Click()

Form12.Show

Form2.Hide

End Sub

Private Sub mnshowloaninformation_Click()

Form18.Show

Form2.Hide

End Sub

Private Sub mntakeproduct_Click()

Form13.Show

Form2.Hide

End Sub

Private Sub mntop10list_Click()

Form21.Show

Form2.Enabled = False

End Sub

Private Sub Text1_Change()

End Sub

FORM - 3

Private Sub temizle()

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

End Sub

Private Sub Command1_Click()

If Text2 = Text3 And Text2 <> "" Then

Form1.user.Index = "primarykey"

Form1.user.Seek "=", Form1.username

If Form1.user.NoMatch = False Then

Form1.user.Edit

Form1.user.Fields("password") = Text2

Form1.user.Update

```

        MsgBox ("Password has changed..!")

        Unload Me

        Form2.Enabled = True

    End If

Else

    MsgBox ("Passwords are no matches which are you entered..! Check it again...")

    Text2.SetFocus

End If

End Sub


Private Sub Command2_Click()

    Unload Me

End Sub


Private Sub Form_Load()

    Call temizle

    Text1 = Form1.username

End Sub


Private Sub Form_Unload(Cancel As Integer)

    Form2.Enabled = True

End Sub


Private Sub Text2_GotFocus()

    Text2.SelStart = 0

```

```
Text2.SelLength = Len(Text2)
```

```
End Sub
```

```
Private Sub Text3_GotFocus()
```

```
Text3.SelStart = 0
```

```
Text3.SelLength = Len(Text3)
```

```
End Sub
```

FORM – 4

```
Public form4gelenform As Integer
```

```
Private Sub gorunum1()
```

```
Command1.Enabled = True
```

```
Text1.Enabled = False
```

```
Text2.Enabled = False
```

```
Text3.Enabled = False
```

```
Combo1.Enabled = True
```

```
Command5.Enabled = False
```

```
Command4.Enabled = False
```

```
Command3.Enabled = True
```

```
Command2.Enabled = True
```

```
Command6.Enabled = True
```

```
Check1.Enabled = False
```

```
Check3.Enabled = False
```

```
Check4.Enabled = False
```

```
Check5.Enabled = False
```

```
Check6.Enabled = False
```


End Sub

Private Sub gorunum2()

Text1.Enabled = True

Text2.Enabled = True

Text3.Enabled = True

Command1.Enabled = False

Command2.Enabled = False

Command3.Enabled = False

Command4.Enabled = True

Command5.Enabled = False

Command6.Enabled = False

Combo1.Enabled = False

Check1.Enabled = True

Check3.Enabled = True

Check4.Enabled = True

Check5.Enabled = True

Check6.Enabled = True

End Sub

Private Sub gorunum3()

Check1.Enabled = True

Check3.Enabled = True

Check4.Enabled = True

Check5.Enabled = True

Check6.Enabled = True

Combo1.Enabled = False

```

Command1.Enabled = False

Command2.Enabled = False

Command3.Enabled = False

Command4.Enabled = True

Command5.Enabled = True

Command6.Enabled = False

End Sub

Private Sub combo1doldur()

Combo1.Clear

If Form1.user.RecordCount > 0 Then

    Form1.user.MoveFirst

    While Not Form1.user.EOF

        Combo1.AddItem Form1.user.Fields("username")

        Form1.user.MoveNext

    Wend

    Combo1.ListIndex = 0

End If

End Sub

Private Sub temizle()

Text1 = ""

Text2 = ""

Text3 = ""

End Sub

Private Sub vtekle()

If Check1.Value = 1 Then

```

```

        Form1.user.Fields("perprocard") = True
    Else
        Form1.user.Fields("perprocard") = False
    End If

    If Check5.Value = 1 Then
        Form1.user.Fields("percompreports") = True
    Else
        Form1.user.Fields("percompreports") = False
    End If

    If Check4.Value = 1 Then
        Form1.user.Fields("peradduser") = True
    Else
        Form1.user.Fields("peradduser") = False
    End If

    If Check3.Value = 1 Then
        Form1.user.Fields("perchangeprice") = True
    Else
        Form1.user.Fields("perchangeprice") = False
    End If

End Sub

Private Sub Check6_Click()
    If Check6.Value = 1 Then
        Check1.Value = 1
        Check3.Value = 1
    
```


Check4.Value = 1

Check5.Value = 1

Else

Check1.Value = 0

Check3.Value = 0

Check4.Value = 0

Check5.Value = 0

End If

End Sub

Private Sub Command1_Click()

Call temizle

Call gorunum2

Check6.Value = vbChecked

End Sub

Private Sub Command2_Click()

If Combo1.ListIndex >= 0 Then

If (Form4.form4gelenform = 2 And Form1.username <> Combo1.Text) Or
Form4.form4gelenform = 1 Then

Call temizle

Call gorunum3

Check6.Value = 0

Form1.user.Index = "primarykey"

Form1.user.Seek "=", Combo1.Text

If Form1.user.Fields("perprocard") = True Then

```
        Check1.Value = 1
    Else
        Check1.Value = 0
    End If

    If Form1.user.Fields("peradduser") = True Then
        Check4.Value = 1
    Else
        Check4.Value = 0
    End If

    If Form1.user.Fields("percompreports") = True Then
        Check5.Value = 1
    Else
        Check5.Value = 0
    End If

    If Form1.user.Fields("perchangeprice") = True Then
        Check3.Value = 1
    Else
        Check3.Value = 0
    End If

    Text1 = Form1.user.Fields("username")
Else
    MsgBox ("You cannot change details of active user..!")
End If

Else
    MsgBox ("No User record to change details...")
```

End If

End Sub

Private Sub Command3_Click()

Dim msg As Integer

Call temizle

If Combo1.ListIndex >= 0 Then

 If Form4.form4gelenform = 1 Or (Form4.form4gelenform = 2 And Form1.username <> Combo1.Text) Then

 msg = MsgBox("Are you sure to delete this user", 36, "Delete User")

 If msg = 6 Then

 Form1.user.Index = "primarykey"

 Form1.user.Seek "=", Combo1.Text

 Form1.user.Delete

 MsgBox ("The User has been deleted successfully...")

 Call combo1doldur

 End If

 Else

 MsgBox ("You cannot delete active user..!")

 End If

Else

 MsgBox ("You have to select a user to delete")

End If

End Sub

Private Sub Command4_Click()

Call gorunum1

Call temizle

Command5.Enabled = False

End Sub

Private Sub Command5_Click()

Dim mesaj1 As String

mesaj1 = "Saved Successfully"

If Text1.Enabled = True Then

 If Text2 = Text3 Then

 Form1.user.Index = "primarykey"

 Form1.user.Seek "=", Text1

 If Form1.user.NoMatch = True Then

 Form1.user.AddNew

 Form1.user.Fields("username") = Text1

 Form1.user.Fields("password") = Text2

 Call vtekle

 Form1.user.Update

 MsgBox (mesaj1)

 Call combo1doldur

 Call temizle

 Call gorunum1

 Else

 MsgBox ("This username is using by someone else! Select Difference
username.")

```

        Text1.SetFocus

    End If

Else

    MsgBox ("Check password information... it is not same...")

    Text2.SetFocus

    End If

Else

    Form1.user.Index = "primarykey"

    Form1.user.Seek "=", Combo1.Text

    Form1.user.Edit

    Call vtekle

    Form1.user.Update

    MsgBox (mesaj1)

    Call temizle

    Call gorunum1

End If

End Sub

Private Sub Command6_Click()

    Unload Me

End Sub

Private Sub Form_Load()

    Call temizle

```

If Form1.user.RecordCount > 0 Then

Call gorunum1

Call combo1doldur

Else

MsgBox ("Program has no any user record..! Firstly you must add a user to continue...")

Call Command1_Click

End If

End Sub

Private Sub Form_Unload(Cancel As Integer)

Dim msg As Integer

Dim msg2 As Integer

If Form4.form4gelenform = 1 Then

If Form1.user.RecordCount > 0 Then

Call Form1.form1temizle

Form1.Show

Call Form1.form1combo1doldur

Else

msg = MsgBox("Any user defined..! If you still continue program will shut down. Do you want to continue?", 36, "CD Market")

If msg = 6 Then

Form1.tablokapat

End

End If


```

    If msj = 7 Then
        Cancel = True
    End If
End If

End If

If Form4.form4gelenform = 2 Then
    Form2.Enabled = True
End If

End Sub

Private Sub Text1_Change()
    If (Text1 = "" Or Text2 = "" Or Text3 = "") And Text1.Enabled = True Then
        Command5.Enabled = False
    Else
        Command5.Enabled = True
    End If
End Sub

Private Sub Text2_Change()
    If (Text1 = "" Or Text2 = "" Or Text3 = "") And Text1.Enabled = True Then
        Command5.Enabled = False
    Else
        Command5.Enabled = True
    End If
End Sub

```

```
Private Sub Text3_Change()
```

```
If (Text1 = "" Or Text2 = "" Or Text3 = "") And Text1.Enabled = True Then
```

```
    Command5.Enabled = False
```

```
Else
```

```
    Command5.Enabled = True
```

```
End If
```

```
End Sub
```

```
Private Sub Text1_GotFocus()
```

```
Text1.SelStart = 0
```

```
Text1.SelLength = Len(Text1)
```

```
End Sub
```

```
Private Sub Text2_GotFocus()
```

```
Text2.SelStart = 0
```

```
Text2.SelLength = Len(Text1)
```

```
End Sub
```

```
Private Sub Text3_GotFocus()
```

```
Text3.SelStart = 0
```

```
Text3.SelLength = Len(Text1)
```

```
End Sub
```

FORM – 5

```
Public stockkod As String
```

```
Public stocktype As String
```

```
Dim adet As Integer
```

```
Public Sub form5combo1doldur()
```

```
Combo1.Clear
```

```

If Form1.producttype.RecordCount > 0 Then
    Form1.producttype.MoveFirst

    While Not Form1.producttype.EOF

        Combo1.AddItem Form1.producttype.Fields("type")

        Form1.producttype.MoveNext

    Wend

    Combo1.ListIndex = 0

End If

End Sub

Public Sub form5combo2dolur()

    Combo2.Clear

    If Form1.category.RecordCount > 0 Then

        Form1.category.MoveFirst

        While Not Form1.category.EOF

            If Combo1.Text = Form1.category.Fields("type") Then

                Combo2.AddItem Form1.category.Fields("categoryname")

            End If

            Form1.category.MoveNext

        Wend

        If Combo2.ListCount > 0 Then

            Combo2.ListIndex = 0

        End If

    End If

End Sub

Public Sub form5combo3dolur()

```



```

Combo3.Clear

If Form1.producttype.RecordCount > 0 Then
    Form1.producttype.MoveFirst

    While Not Form1.producttype.EOF

        Combo3.AddItem Form1.producttype.Fields("type")

        Form1.producttype.MoveNext

    Wend

    Combo3.ListIndex = 0

End If

End Sub

Private Sub form5temizle()

    Text1 = ""

    Text2 = ""

    Text3 = ""

    Text4 = ""

    Text5 = ""

    Text6 = ""

    Text7 = ""

End Sub

Private Sub stokeykle()

    Form1.stock.AddNew

    Form1.stock.Fields("stockcode") = stockkod

    Form1.stock.Fields("type") = stocktype

    Form1.stock.Fields("stockname") = Text2

    Form1.stock.Fields("category") = Combo2.Text

```

```

Form1.stock.Fields("quantity") = Val(Text5)

Form1.stock.Update

Text1 = stockkod

MsgBox ("Record has been saved..!")

End Sub

Private Sub stokdegistir()

Form1.stock.Edit

Form1.stock.Fields("stockname") = Text2

Form1.stock.Fields("category") = Combo2.Text

Form1.stock.Fields("quantity") = Val(Text5)

Form1.stock.Update

MsgBox ("Changes are saved..!")

Text4 = Combo2.Text

End Sub

Private Sub gorunum1()

Command1.Enabled = True

Command2.Enabled = True

Command3.Enabled = True

Command4.Enabled = True

Command5.Enabled = True

Command6.Enabled = True

Command7.Enabled = True

Command8.Enabled = True

Command9.Enabled = False

Command10.Enabled = False

```

Command11.Enabled = True

Command12.Enabled = True

Check1.Visible = True

Text1.Locked = True

Text2.Locked = True

Text5.Locked = True

Text6.Locked = False

Text3.Visible = True

Text4.Visible = True

Combo1.Visible = False

Combo2.Visible = False

Combo1.Locked = False

End Sub

Private Sub gorunum2()

Command1.Enabled = False

Command2.Enabled = False

Command3.Enabled = False

Command4.Enabled = False

Command5.Enabled = False

Command6.Enabled = False

Command7.Enabled = False

Command8.Enabled = False

Command9.Enabled = True

Command10.Enabled = True

Command11.Enabled = False

Command12.Enabled = False

Check1.Visible = False

Text1.Locked = False

Text2.Locked = False

Text5.Locked = False

Text6.Locked = True

Text3.Visible = False

Text4.Visible = False

Combo1.Visible = True

Combo2.Visible = True

End Sub

Private Sub Check1_Click()

If Check1.Value = 1 Then

 If Text1 <> "" Then

 Text7.Visible = True

```
Label7.Visible = True

Text7 = ""

Call gorunum2

Text1.Locked = True

Text2.Locked = True

Text3.Locked = True

Text4.Locked = True

Text5.Locked = True

Text3.Visible = True

Text4.Visible = True

Combo1.Visible = False

Combo2.Visible = False

Check1.Visible = True

Text7.SetFocus

adet = Text5

Else

Check1.Value = 0

MsgBox ("You must select a product to add quantity")

End If

Else

Text7.Visible = False

Label7.Visible = False

Call gorunum1

End If

End Sub
```

```

Private Sub Combo1_Click()

Call Form5.form5combo2dolur

End Sub

```

```

Private Sub Command1_Click()

```

```

If Combo3.ListCount > 0 Then

```

```

    If Text6 <> "" Then

```

```

        Form1.stock.Index = "primarykey"

```

```

        Form1.stock.Seek "=", Text6, Combo3.Text

```

```

        If Form1.stock.NoMatch = False Then

```

```

            Text1 = Form1.stock.Fields("stockcode")

```

```

            Text2 = Form1.stock.Fields("stockname")

```

```

            Text3 = Form1.stock.Fields("type")

```

```

            Text4 = Form1.stock.Fields("category")

```

```

            Text5 = Form1.stock.Fields("quantity")

```

```

            Image1.Visible = True

```

```

            Image2.Visible = False

```

```

            On Local Error GoTo hata

```

```

            Form5.Image1.Picture = LoadPicture(Form1.programyer + "\ProductPictures\" +
Text3 + Text1 + ".jpg")

```

```

hata:

```

```

    If Err = 53 Then

```

```

        Image1.Visible = False

```

```

        Image2.Visible = True

```

```

    End If

```

```

Text6 = ""

Text6.SetFocus

Else

MsgBox ("Not found record")

Text6.SetFocus

End If

Else

MsgBox ("You must enter the Code..!")

End If

Else

MsgBox ("No any record..!")

End If

End Sub

Private Sub Command10_Click()

If Combo1.ListCount < 1 Or Combo2.ListCount < 1 Then

MsgBox ("Add Stock type and category to add stock")

Exit Sub

End If

If Check1.Value = 1 Then

If Text7 <> "" Then

Form1.stock.Index = "primarykey"

Form1.stock.Seek "=", Text1, Combo1.Text

If Form1.stock.NoMatch = False Then

Form1.stock.Edit

```



```

Form1.stock.Fields("quantity") = Trim(Str(Val(Text5) + Val(Text7)))

Form1.stock.Update

Call gorunum1

Check1.Value = 0

Text5 = Trim(Str(Val(Text5) + Val(Text7)))

End If

Else

MsgBox ("Enter the Adding quantity..!")

End If

Exit Sub

End If

If Text2 = "" Or Text5 = "" Then

MsgBox ("Fill all blank informations..!")

Exit Sub

End If

If Text1.Locked = True Then

Form1.stock.Index = "primarykey"

Form1.stock.Seek "=", Text1, Combo1.Text

If Form1.stock.NoMatch = False Then

Form5.stockkod = Text1

Form5.stocktype = Combo1.Text

Call stokdegistir

Call gorunum1

End If

Else

```

```

If Text1 <> "" Then

Form1.stock.Index = "primarykey"

Form1.stock.Seek "=", Text1, Combo1.Text

    If Form1.stock.NoMatch = False Then

        MsgBox ("This Stock code is using wiht other item..! give a difference code..!")

        Text1.SetFocus

        Exit Sub

    Else

        Form5.stockkod = Text1

        Form5.stocktype = Combo1.Text

        Call stokekle

        Text3 = Combo1.Text

        Text4 = Combo2.Text

        Call gorunum1

        End If

    Else

        If Form1.stock.RecordCount > 0 Then

            stockkod = "0"

            Form1.stock.MoveFirst

            While Not Form1.stock.EOF

                If Val(Form1.stock.Fields("stockcode")) > Form5.stockkod And
Form1.stock.Fields("type") = Combo1.Text And Val(Form1.stock.Fields("stockcode"))
< 10000000 Then

                    stockkod = Form1.stock.Fields("stockcode")

                End If

                Form1.stock.MoveNext

```

```

        Wend

        stockkod = Val(stockkod) + 1

        Else

        stockkod = "1"

        End If

        Form5.stocktype = Combo1.Text

        Call stokekke

        Text3 = Combo1.Text

        Text4 = Combo2.Text

        Call gorunum1

        End If

    End If

End Sub

Private Sub Command11_Click()

    Dim msj As Integer

    If Image1.Visible = True Then

        msj = MsgBox("Are you sure to delete picture?", 36, "CD Market")

        If msj = 6 Then

            Kill Form1.programyer + "\ProductPictures\" + Text3 + Text1 + ".jpg"

            Image1.Visible = False

            Image2.Visible = True

            End If

        Else

            MsgBox ("No Picture to delete..!")

        End If

    End Sub

```

End If

End Sub

Private Sub Command12_Click()

If Text1 <> "" Then

Form9.form9gelenform = 5

Form9.Show

Form5.Enabled = False

Form5.stockkod = Text1

Form5.stocktype = Text3

Else

MsgBox ("You must select a product to add to it a picture..!")

End If

End Sub

Private Sub Command2_Click()

If Form1.stock.RecordCount > 0 Then

Form8.form8gelenform = 5

Form8.Show

Form5.Hide

Form2.Hide

Else

MsgBox ("No any product record in database..!")

End If

End Sub


```
Private Sub Command3_Click()
```

```
Call form5temizle
```

```
Call gorunum2
```

```
Text1.SetFocus
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
Dim i As Long
```

```
If Text1 <> "" Then
```

```
Call gorunum2
```

```
Text1.Locked = True
```

```
For i = 0 To Combo1.ListCount - 1
```

```
    If Combo1.List(i) = Text3 Then
```

```
        Combo1.ListIndex = i
```

```
    Exit For
```

```
End If
```

```
Next
```

```
For i = 0 To Combo2.ListCount - 1
```

```
    If Combo2.List(i) = Text4 Then
```

```
        Combo2.ListIndex = i
```

```
    Exit For
```

```
End If
```

```
Next
```

```
devam:
```

Combo1.Visible = False

Text3.Visible = True

Text2.SetFocus

Else

MsgBox ("Find a product to change it..!")

End If

End Sub

Private Sub Command5_Click()

Dim msj As Integer

silme = False

If Text1 <> "" Then

msj = MsgBox("Are you sure to delete this item..!", 36, "CD Market")

If msj = 6 Then

If Form1.refusal.RecordCount > 0 Then

Form1.refusal.MoveFirst

While Not Form1.refusal.EOF

If Form1.refusal.Fields("stockcode") = Text1 And
Form1.refusal.Fields("type") = Text3 Then

MsgBox ("This Product record using in database for other information..! you
cannot delete it..!")

Exit Sub

End If

Form1.refusal.MoveNext

Wend

End If

```

If Form1.content.RecordCount > 0 Then

    Form1.content.MoveFirst

    While Not Form1.content.EOF

        If Form1.content.Fields("stockcode") = Text1 And
        Form1.content.Fields("type") = Text3 Then

            MsgBox ("This Product record using in database for other information..! you
            cannot delete it..!")

            Exit Sub

            End If

            Form1.content.MoveNext

        Wend

    End If

    Form1.stock.Index = "primarykey"

    Form1.stock.Seek "=", Text1, Text3

    If Form1.stock.NoMatch = False Then

        Form1.stock.Delete

        If Image1.Visible = True Then

            Kill Form1.programyer + "\ProductPictures\" + Text3 + Text1 + ".jpg"

            Image1.Visible = False

            Image2.Visible = True

            End If

            MsgBox ("Deleted..!")

            Call form5temizle

            End If

        End If

    Else

```

MsgBox ("Find a product to delete..!")

End If

End Sub

Private Sub Command6_Click()

Form6.form6gelenform = 5

Form6.Show

Form5.Enabled = False

End Sub

Private Sub Command7_Click()

If Form1.producttype.RecordCount > 0 Then

Form7.form7gelenform = 5

Form7.Show

Form5.Enabled = False

Else

MsgBox ("Firstly you must enter product type to add category")

End If

End Sub

Private Sub Command8_Click()

Unload Me

End Sub


```

Private Sub Command9_Click()

If Text1.Locked = True Then

Form1.stock.Index = "primarykey"

Form1.stock.Seek "=", Text1, Combo1.Text

    If Form1.stock.NoMatch = 0 Then

        Text2 = Form1.stock.Fields("stockname")

        Text5 = Form1.stock.Fields("quantity")

        End If

    Else

        Call form5temizle

    End If

    Call gorunum1

    Check1.Value = 0

End Sub

```

```

Private Sub Form_Load()

Call gorunum1

Call form5temizle

Label7.Visible = False

Text7.Visible = False

Text3.Locked = True

Text4.Locked = True

Call Form5.form5combo1doldur

Call Form5.form5combo2dolur

Call Form5.form5combo3dolur

```

```
Image2.Visible = True
```

```
Image1.Visible = False
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
Form2.Enabled = True
```

```
End Sub
```

```
Private Sub Text1_GotFocus()
```

```
Text1.SelStart = 0
```

```
Text1.SelLength = Len(Text1)
```

```
End Sub
```

```
Private Sub Text1_LostFocus()
```

```
If Text1 <> "" Then
```

```
Form1.rakamkontrolgelen = Text1
```

```
Call Form1.rakamkontrol
```

```
If Form1.rakamkontrolsonucharf = True Then
```

```
MsgBox ("Enter Just digits")
```

```
Text1.SetFocus
```

```
End If
```

```
End If
```

```
End Sub
```

```
Private Sub Text5_LostFocus()
```

```

Form1.rakamkontrolgelen = Text5

Call Form1.rakamkontrol

If Form1.rakamkontrolsonucharf = True Then

MsgBox ("Enter Just digits")

Text5.SetFocus

End If

End Sub

Private Sub Text7_LostFocus()

Form1.rakamkontrolgelen = Text7

Call Form1.rakamkontrol

If Form1.rakamkontrolsonucharf = True Then

MsgBox ("Enter Just digits")

Text7.SetFocus

End If

End Sub

Private Sub Text2_GotFocus()

Text2.SelStart = 0

Text2.SelLength = Len(Text2)

End Sub

Private Sub Text3_GotFocus()

Text3.SelStart = 0

Text3.SelLength = Len(Text3)

End Sub

Private Sub Text4_GotFocus()

```

```

Text4.SelStart = 0

Text4.SelLength = Len(Text4)

End Sub

Private Sub Text5_GotFocus()

Text5.SelStart = 0

Text5.SelLength = Len(Text5)

End Sub

Private Sub Text6_GotFocus()

Text6.SelStart = 0

Text6.SelLength = Len(Text6)

End Sub

```

FORM – 6

```

Public form6gelenform As Integer

Private Sub temizle()

Text1 = ""

Text2 = ""

Text3 = ""

End Sub

Private Sub gorunum1()

Text1.Enabled = False

Text2.Enabled = False

Text3.Visible = False

Command1.Enabled = True

Command2.Enabled = True

Command3.Enabled = True

```



```

Command4.Enabled = False

Command5.Enabled = False

Combo1.Visible = True

End Sub

Private Sub gorunum2()

Text1.Enabled = True

Text2.Enabled = True

Text3.Visible = True

Command1.Enabled = False

Command2.Enabled = False

Command3.Enabled = False

Command4.Enabled = True

Command5.Enabled = True

Combo1.Visible = False

End Sub

Private Sub vtfiyatkaydet()

If Format(Left(Text1, Len(Text1) - 4), "#.##") <> "," Then

    Form1.producttype.Fields("rentprice") = Format(Left(Text1, Len(Text1) - 4), "#.##")

End If

If Format(Left(Text2, Len(Text2) - 4), "#.##") <> "," Then

    Form1.producttype.Fields("saleprice") = Format(Left(Text2, Len(Text2) - 4), "#.##")

End If

End Sub

Private Sub combo1doldur()

Combo1.Clear

```

If Form1.producttype.RecordCount > 0 Then

Form1.producttype.MoveFirst

While Not Form1.producttype.EOF

Combo1.AddItem Form1.producttype.Fields("type")

Form1.producttype.MoveNext

Wend

Combo1.ListIndex = 0

End If

End Sub

Private Sub Combo1_Click()

Form1.producttype.Index = "primarykey"

Form1.producttype.Seek "=", Combo1.Text

Text1 = Format(Form1.producttype.Fields("rentprice"), "#,##0.## YTL")

Text2 = Format(Form1.producttype.Fields("saleprice"), "#,##0.## YTL")

Text3 = Combo1.Text

End Sub

Private Sub Command1_Click()

Call gorunum2

Call temizle

Text3.Enabled = True

Text2 = "0, YTL"

Text1 = "0, YTL"

End Sub

Private Sub Command2_Click()

If Combo1.ListIndex >= 0 Then

Text3 = Combo1.Text

Call gorunum2

Text3.Enabled = False

Text1.SetFocus

Else

MsgBox ("You must select Product Type which one you want to change")

End If

End Sub

Private Sub Command3_Click()

Dim msj As Integer

Dim hata As Boolean

If Combo1.ListCount > 0 Then

msj = MsgBox("When you delete product type you will lose information of this product type's categories...! Do you want to continue?", 36, "Delete Product Type")

If msj = 6 Then

If Form1.stock.RecordCount > 0 Then

Form1.stock.MoveFirst

While Not Form1.stock.EOF

If Form1.stock.Fields("type") = Combo1.Text Then

hata = True

GoTo son

End If

```

        Form1.stock.MoveNext
    Wend
End If
If Form1.category.RecordCount > 0 Then
    Form1.category.MoveFirst
    While Not Form1.category.EOF
        If Form1.category.Fields("type") = Combo1.Text Then
            Form1.category.Delete
        End If
        Form1.category.MoveNext
    Wend
End If
Form1.producttype.Index = "primarykey"
Form1.producttype.Seek "=", Combo1.Text
If Form1.producttype.NoMatch = 0 Then
    Form1.producttype.Delete
    Call combo1doldur
End If
End If
Call temizle
If Combo1.ListCount > 0 Then
    Combo1.ListIndex = 0
End If
Else
    MsgBox ("You must select Product Type which one you want to delete")

```


End If

son:

If hata = True Then

MsgBox ("Some stock is using this product type. So you cannot delete..! Firstly you must change product types of this stock...")

End If

End Sub

Private Sub Command4_Click()

Call temizle

Call gorunum1

If Combo1.ListCount > 0 Then

Call Combo1_Click

End If

End Sub

Private Sub Command5_Click()

If Text1 <> "0, YTL" And Text2 <> "0, YTL" Then

If Text3.Enabled = False Then

Form1.producttype.Index = "primarykey"

Form1.producttype.Seek "=", Text3

Form1.producttype.Edit

Call vtfiyatkaydet

Form1.producttype.Update

Call temizle

Call combo1doldur

```

    Call gorunum1

Else

    If Text3 <> "" Then

        Form1.producttype.Index = "primarykey"

        Form1.producttype.Seek "=", Text3

        If Form1.producttype.NoMatch = True Then

            Form1.producttype.AddNew

            Form1.producttype.Fields("type") = Text3

            Call vtfiyatkaydet

            Form1.producttype.Update

            Call temizle

            Call combo1doldur

            Call gorunum1

        Else

            MsgBox ("The program has this product type..! Try again...")

            Text3.SetFocus

        End If

    Else

        MsgBox ("You must enter product type..!")

        Text3.SetFocus

    End If

End If

Else

    MsgBox ("You must enter a price..!")

    Text1.SetFocus

```

End If

End Sub

Private Sub Form_Load()

Call temizle

Call combo1doldur

Call gorunum1

End Sub

Private Sub Form_Unload(Cancel As Integer)

If Form6.form6gelenform = 2 Then

Form2.Enabled = True

End If

If Form6.form6gelenform = 5 Then

Form5.Enabled = True

Form5.form5combo1doldur

Form5.form5combo3doldur

End If

End Sub

Private Sub Text1_GotFocus()

If Len(Text1) > 4 Then

Text1.Text = Left(Text1, Len(Text1) - 4)

End If

Text1 = Format(Text1, "#.##")

Text1.SelStart = 0

```

Text1.SelLength = Len(Text1)

End Sub

Private Sub Text1_LostFocus()

If Text1 <> "" And Text1 <> "," Then

    If IsNumeric(Text1) = True Then

        Text1 = Format(Text1, "#,##0.## YTL")

    Else

        Text1 = ""

        Text1.SetFocus

    End If

Else

    Text1 = "0, YTL"

End If

End Sub

Private Sub Text2_GotFocus()

If Len(Text2) > 4 Then

    Text2.Text = Left(Text2, Len(Text2) - 4)

End If

Text2 = Format(Text2, "#.##")

Text2.SelStart = 0

Text2.SelLength = Len(Text2)

End Sub

Private Sub Text2_LostFocus()

If Text2 <> "" And Text2 <> "," Then

```



```

If IsNumeric(Text2) = True Then
    Text2 = Format(Text2, "#,##0.## YTL")
Else
    Text2 = ""
    Text2.SetFocus
End If

Else
    Text2 = "0, YTL"
End If

End Sub

```

```

Private Sub Text3_GotFocus()
    Text3.SelStart = 0
    Text3.SelLength = Len(Text3)
End Sub

```

FORM – 7

```

Public form7gelenform As Integer

Private Sub combo2doldur()
    Combo2.Clear

    If Form1.producttype.RecordCount > 0 Then
        Form1.producttype.MoveFirst

        While Not Form1.producttype.EOF
            Combo2.AddItem Form1.producttype.Fields("type")
            Form1.producttype.MoveNext
        Wend
    End If
End Sub

```

```

Combo2.ListIndex = 0

Else

MsgBox ("Firstly you must enter product type to add to it a category..!")

Unload Me

End If

End Sub

Private Sub combo1doldur()

Combo1.Clear

If Form1.category.RecordCount > 0 Then

Form1.category.MoveFirst

While Not Form1.category.EOF

If Form1.category.Fields("type") = Combo2.Text Then

Combo1.AddItem Form1.category.Fields("categoryname")

End If

Form1.category.MoveNext

Wend

If Combo1.ListCount > 0 Then

Combo1.ListIndex = 0

End If

End If

End Sub

Private Sub gorunum1()

Combo2.Enabled = True

Combo1.Visible = True

Text1.Visible = False

```

Command1.Enabled = True

Command2.Enabled = True

Command4.Enabled = False

Command5.Enabled = False

End Sub

Private Sub gorunum2()

Combo2.Enabled = False

Combo1.Visible = False

Text1.Visible = True

Command1.Enabled = False

Command2.Enabled = False

Command4.Enabled = True

Command5.Enabled = True

End Sub

Private Sub Combo2_Click()

Call combo1doldur

End Sub

Private Sub Command1_Click()

Text1 = ""

Call gorunum2

End Sub

Private Sub Command2_Click()

```

Dim msj As Integer

If Combo1.ListIndex >= 0 Then

    msj = MsgBox("Are you sure to delete this category", 36, "Delete Message")

    If msj = 6 Then

        Form1.category.Index = "primarykey"

        Form1.category.Seek "=", Combo1.Text, Combo2.Text

        Form1.category.Delete

        Call combo1doldur

    End If

Else

    MsgBox ("No any record to delete")

End If

End Sub

```

```

Private Sub Command4_Click()

    Text1 = ""

    Call gorunum1

End Sub

```

```

Private Sub Command5_Click()

    If Text1 <> "" Then

        Form1.category.Index = "primarykey"

        Form1.category.Seek "=", Text1, Combo2.Text

        If Form1.category.NoMatch = True Then

            Form1.category.AddNew

```



```

Form1.category.Fields("type") = Combo2.Text

Form1.category.Fields("categoryname") = Text1

Form1.category.Update

Call combo1doldur

Text1 = ""

Call gorunum1

Else

MsgBox (Combo2.Text + " Product Type has this category..! Try another.")

Text1.SetFocus


End If

Else

MsgBox ("Enter the category name..!")

Text1.SetFocus

End If

End Sub


Private Sub Form_Load()

Text1 = ""

Call combo2doldur

Call combo1doldur

Call gorunum1

End Sub


Private Sub Form_Unload(Cancel As Integer)

```

If Form7.form7gelenform = 2 Then

Form2.Enabled = True

End If

If Form7.form7gelenform = 5 Then

Form5.Enabled = True

Call Form5.form5combo2dolur

End If

End Sub

Private Sub Text1_GotFocus()

Text1.SelStart = 0

Text1.SelLength = Len(Text1)

End Sub

OTHER FORMS

Public form9gelenform As String

Dim harddisk As String

Dim dosyalar As String

Private Sub Dir1_Change()

File1.Path = Dir1.Path

End Sub

Private Sub Drive1_Change()

Dir1.Path = Drive1.Drive

End Sub

Private Sub File1_Click()

Image1.Picture = LoadPicture(File1.Path + "\" + File1.FileName)

```

End Sub

Private Sub Form_Load()
    On Error GoTo devam
    Drive1.Drive = harddisk
    Dir1.Path = dosyalar
devam:
    File1.Pattern = "*.jpg"
End Sub

Private Sub Form_Unload(Cancel As Integer)
    harddisk = Drive1.Drive
    dosyalar = Dir1.Path
    If Form9.form9gelenform = 5 Then
        Form5.Enabled = True
    End If
    If Form9.form9gelenform = 10 Then
        Form10.Enabled = True
    End If
End Sub

Private Sub Command1_Click()
    Dim msj As Integer
    If Form9.form9gelenform = 5 Then
        hata1:
        If Err = 76 Then
            MkDir Form1.programyer + "\ProductPictures"
        End If
    End If

```

```

If File1.ListIndex >= 0 Then

'resmi kopyala

On Local Error GoTo hata1

FileCopy File1.Path + "\" + File1.FileName, Form1.programyer + "\ProductPictures\"
+ Form5.stocktype + Form5.stockkod + ".jpg"

'resmi kopyala sonu

Form5.Image1.Visible = True

Form5.Image2.Visible = False

Form5.Image1.Picture = LoadPicture(Form1.programyer + "\ProductPictures\" +
Form5.stocktype + Form5.stockkod + ".jpg")

MsgBox ("Picture has been saved saccessfully...")

Unload Me

Else

MsgBox ("There is no selected picute to add..!")

End If

End If

If Form9.form9gelenform = 10 Then

hata2:

If Err = 76 Then

MkDir Form1.programyer + "\CustomerPictures"

End If

If File1.ListIndex >= 0 Then

'resmi kopyala

On Local Error GoTo hata2

FileCopy File1.Path + "\" + File1.FileName, Form1.programyer +
"\CustomerPictures\" + Form10.customercode + ".jpg"

'resmi kopyala sonu

```



```
Form10.Image1.Visible = True

Form10.Image2.Visible = False

Form10.Image1.Picture = LoadPicture(Form1.programyer + "\CustomerPictures\" +
Form10.customercode + ".jpg")

MsgBox ("Picture has been saved saccessfully...")

Unload Me

Else

MsgBox ("There is no selected picute to add..!")

End If

End If

End Sub
```