NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

Video Store Management Program

Graduation Project
Com- 400

Student: Mehmet Akif GÜRSOY(20031361)

Supervisor: Assoc.Prof.Dr.Rahib ABIYEV

Nicosia - 2008

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I am glad to complete my project, which I had given with blessing of God.(Thanks to God)

Next I would like to thank Assoc.Prof.Dr.Rahib ABIYEV for his endless and untiring support, help and his persistence, in the course of the preparation of this project.
Under his guidance, I have overcome many diffuculties that I faced during the various stages of the preparation of this project.

I am very thankful for the help and support of my valuable friends especially Ayşen Gökçen, Mustafa Alıcı, Mehmet Erlale, Emre Selim Şahin, Emre Doğan and Kağan Uzun.

Also thanks all teachers who behaved me in patient and understanding during my studying time. Specially to Asist.Prof.Dr.Elbrus Imanov for everything he has done till now to help.

Finally, I would like to thank my family for their financial and psychological support. Their love and guidance saw me through doubtful times. Their never-ending belief in me and their encouragement has been a crucial and a very strong pillar that has held me together.

Mehmet Akif GÜRSOY

ABSTRACT

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

This system has been designed in a way that it would work speedier than the normal record keeping system. It is a specific program only suitable for the video store management system that makes business in keeping several videos and rent/sale them to the public. The system is divided into 6 main processes in order for better user interface and friendly environment. My solution to the problem I guess is quite understandable as I divided the system into main six processes, such as costumers, videos, suppliers, rentals, sales and queries. Each of the section I designed is easy to understand and learn fast.

As there are hundreds of ways to solve any problem so if you find any suggestions about my program please do not hesitate telling me them.

INTRODUCTION

This project is about Video Store Management Program. Borland Delphi and Microsoft Office Access database were used in order to achieve the project. This program can be used in the market places where Video Cds are sold or rented.

I developed the project and designed it to be more flexible for the users. Of course I used the references of some programs which have been designed previously and can be purchased from the web. But all the codes and designing belongs to me.

The goal of this program is to record, rent and sell the video cds in the video stores easly. Which costumers bought or rented which videos, how many video cds are there in the video store, their tarilers and etc. The user of the program can easily reach all of these information.

Why I chose this project and did it? Because of my brother has a video store in my country. He isn't using any program for this. So I decided to make a program for him. You can say that this project is enouhg for this job. I know it is "No". But it is my first time and I am working on this project. If I have enough time next, it will be nice program for really in use.

The Project consist of introduction, five chapters and conclusion.

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

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

Chapter three concerned Video Store Management Program's database tables properties and database relationships screenshots.

Chapter four includes the screenshots and user manual of Video Store Management Program.

Chapter five includes Video Store Management Program' s source codes.

1.INTRODUCTION TO DELPHI

1.1 A Brief History Of Borland's Delphii

Pascal

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

Beginnings

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

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

The 1970's

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

The 1980's

In 1982 ISO Pascal appears. The big event is in November 1983, when Turbo Pascal is released in a blaze of publicity. Turbo Pascal reaches release 4 by 1987. Turbo Pascal excelled on speed of compilation and execution, leaving the competition in its wake.

From Turbo Pascal to Delphi

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

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

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

Delphi for Microsof . Net

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

For about 15 years, starting soon after the release of Windows 3.0, Microsoft has kept promising that its operating system and their API would be based on a real object model instead of functions. According to the speculations, Windows 95 (and later Windows 2000) should have been based on this revolutionary approach. Nothing like this happened, but Microsoft kept pushing COM (Component Object Model), built the Windows 95 shell on top of it, pushed applications integration with COM and derivative technologies and reached the peak by introducing COM+ with Windows 2000.Now, soon after the release of the complete foundation required for high-level COM programming, Microsoft has decided to switch to a new core technology, part of the dotNet initiative. COM wasn't really suited for the integration of fine-grained objects, though it succeeded in providing an architecture for integrating applications or large objects.

Versions and Counting

Some of the original Delphi features that attracted me were its form-based and object-oriented approach, its extremely fast compiler, its great database support, its close integration with Windows programming, and its component technology. But the most important element was the Object Pascal language, which is the foundation of everything else.

Delphi 2 was even better! Among its most important additions were these: the Multi-Record Object and the improved database grid, OLE Automation support and the variant data type, full Windows 95 support and integration, the long string data type, and Visual Form Inheritance.

Delphi 3 added to this the code insight technology, DLL debugging support, component templates, the TeeChart, the Decision Cube, the WebBroker technology, component packages, ActiveForms, and an astonishing integration with COM, thanks to interfaces.

Delphi 4 gave us the AppBrowser editor, new Windows 98 features, improved OLE andCOM support, extended database components, and many additions to the core VCL classes, including support for docking, constraining, and anchoring controls.

Delphi 5 added to the picture many more improvements of the IDE, extended database support (with specific ADO and InterBase datasets), an improved version of MIDAS with Internet support, the TeamSource version-control tool, translation capabilities, the concept of frames, and new components.

Now Delphi adds to all these features support for cross-platform development with the new Component Library for Cross-Platform (CLX), an extended run-time library, the new dbExpress database engine, Web services and exceptional XML support, a powerful Web development framework, more IDE enhancements, and a plethora of new components and classes. Delphi is a great tool, but it is also a complex programming environment that involves many elements.

1.2 What is Delphi?

Delphi is an object oriented, component based, visual, rapid development environment for event driven Windows applications, based on the pascal language. Unlike otjer popular competing Rapid Application Development (RAD) tools, Delphi compiles the code you write and produces really tight, natively executable code for the target platform. In fact the most recent versions of Delphi optimise the compiled code and resulting executables are as efficient as those compiled with any other compiler currently on the market. The term "visual" describes Delphi very well. All of the user interface development is conducted in a What You See Is What You Get environment, which means you can create polished, user friendly interfaces in a very short time, or prototype whole applications in a few hours.

Delphi is, in effect, the latest in a long and distinguished line of Pascal compilers from the company formerly known as Borland, now known as Inprise. In common with the Turbo Pascal compilers that preceded it, Delphi is not just a compiler, but a complete development enviroment. Some of the facilities that are included in the " Integrated development Environment" (IDE) are listed below:

- A syntax sensitive program file editor
- A rapid optimising compiler
- Built in debugging / tracing facilities
- A visual interface developer
- Syntax sensitive help files
- Database creation and editing tools
- Image / icon / cursor / creation / editing tools
- Version control CASE tools

The development environment itself is extensible, and there are a number of add ins available to perform functions such as memory leak detection and profiling. In short, Delphi includes just about everything you need to write applications that will run on an Intel platform under Windows, but if your target platform is a silicon graphics running IRIX, or a Sun Sparc Running SOLARIS, or even a PC running LINUX, then you will need to look elsewhere for your development tool.

This specialisation on one platform and one operating system, makes Delphi a very strong tool. The code it genretas runs very rapidly, and is very stable, once your own bugs have been ironed out.

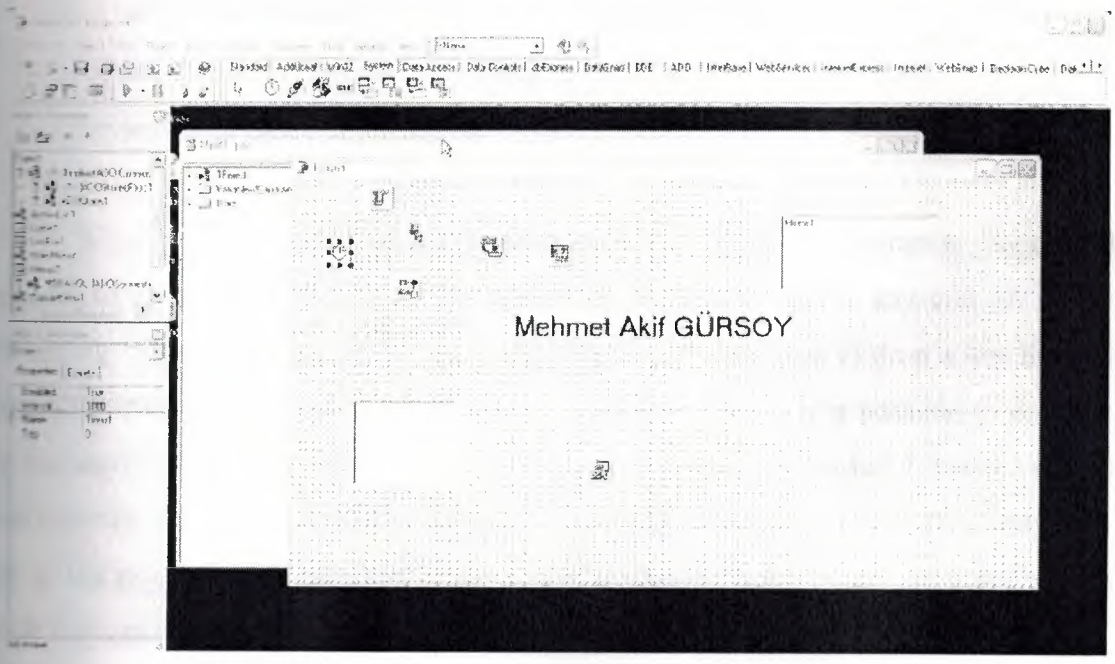

Figure.1.1 Interface of Borland Delphi

1.2.1 What kind of programming can you do with Delphi?

The simple answer is "more or less anything" .Because the code is compiled, it runs quickly, and is therefore suitable for writing more or less any program that you would consider a candidate for the Windows operating system.

You probably won't be using it to write embedded systems for washing machines, toasters or fuel injection systems, but for more or less anything else it can be used.

Some projects to which Delphi is suited:

- Simple single user database applications
- Intermediate multi-user database applications
- Large scale multi-tier, multi-user database applications
- Internet applications

- Graphics applications
- Multimedia applications
- Image processing / image recognition
- Data analysis
- System tools
- Communications tools using the internet, telephone or LAN
- Web based applications

This is not intended to be an exhaustive list, more an indication depth and breadth of Delphi!s applicability. Because it is possible to access any end all of the Windows API, and because if all else fails, Delphi will allow you to drop a few lines of assembler code directly into your ordinary Pascal instructions, it is possible to do more or less anything. Delphi can also be used to write Dynamically Linked Libraries (DLL) and can call out to DLL writen in other programming languages without diffuculty. The job of the programmer has become one of gluing together appropriate components with code that operates them as required.

1.3 The VCL to Applications Developers

Aplications Developers create complete applications by interactions with the Delphi visiual environment. This people use the VCL to create their user-interface and the other elements of their application: database connectivity, data validation, business rules, etc…

Applications Developers should know which properties, eventes, and methods each component makes available. Additionally, by understanding the VCL architecture, Applications Developers will be able to easily identify where they can improve their applications by extending components or creating new ones. The they can maximize the capabilities of these components, and create beter applications.

1.3.1 The VCL to Component Writers

Component Writers expand on the existing VCL, either by developing new components, or by increasing the functionality of existing ones. Many component writers make their components available for Applications Developers to use.

A Component Writer must take their knowledge of the VCL a step further than that of the Application Developer. For example, they must know whether to write a new component or to extend an existing one when the need for a certain characteristic arises. This requires a greater knowledge of the VCL's iner working.

1.3.2 The VCL is made up of components

Components are the building blocks that developers use to design the userinterface and to provide some non-visual capabilities to their applications. To an Application Developer, a component is an object most commonly dragged from the component's properties and add code to the component'svarious events to give the component a specific behavior. To a Component Writer, components are objects in Object Pascal code. Some components encapsulate the behavior of elements provided by the system, such as the Standard Windows 95 controls. Other objects introduce entirely new visual or non-visiual elements, in which case the component's code makes up the entire behaviour of the component.

The comlexity of different components varies widely. Some might be simple while others might encapsulate a elaborate task. There is no limit to what a component can do or be made up of. You can have a very simple component like a TLabel, or a much more complex component which encapsulates the complete functionality of a spreadsheet.

1.4 Component Types

As a component writer, therefore primary types of components that you will work with in Delphi: standart controls, custom controls, graphical controls, and non-visual components. Although these component types are primarily of interest to component writers, it's not a bad idea for applications developers to be familiar with them. They are the foundations on which applications are built.

1.4.1 Standard Components

Some of the components provided by Delphi 2.0 encapsulate the behavior of the standard Windows controls: TButton, TListbox As a component writer, there four primary types of components that you will work with in Delphi: Standard controls, custom controls, graphical controls, and non-visual components. Although these

components types are primarily of interest to component wirites, it's not a bad idea for applications developers to be familiar with them. They are the foundations on which applications are built.

For example. You will find these components on the Standard page of the Component Palette. These components are Windows common controls with Object Pascal wrappers around then.

Each standard component looks and works like the Windows common control which it encapsulates. The VCL wrapper'ssimply makes the control available to you in the form of a Delphi cmponent-it doesn't define the common control's appearance or functionality, but rather, surfaces the ability to modify a control's appearance/functionality in the for of methods and properties.

If you want to use these Standard components unchanged, there is no need to understand how the VCL wraps them. If, however, you want to extend or change one of these components, them you must understand how the Window's common controls is wrapped by the VCL into a Delphi component.

1.4.2 Custom components

Unlike standard components, custom components are controls that don't already have a method for displaying themselves, nor do they have a defined behavior. The Component Writer must provide to code that tells the component how to draw itself and determines how the component behaves when the user interacts with it. Examples of existing custom components are the TPanel and TStringGrid components.

It should be mentioned here that both Standard and custom components are windowed controls. A "windowed control" has a window associated with it and, therefore, has a window handle. Windowed controls have three characterisrics: they can receive the input focus, they use system resources, and they can be parents to other controls. An axample of a component which can be a container is the TPanel component.

1.4.3 Graphical components

Graphical components are visual controls which cannot receie the input focus from the user. They are non-windowed controls. Graphical components allowyou to display something to the user without using up any system resources; they have less

window handle-thus, they have less "overhead" than satandard or custom components. Graphical components don't require a window handle-thus, they cannot can't get focus. Some example of graphical components are the TLabel and TShape components.

Graphical components cannot be containers of other components. This means that they cannot own other components which are paced on top of them.

1.4.4 Structure of a components

All components share a similar structure. Each components consists of common elements that allow developers to manipulate its appearence and function via properties, methods and events. The following sections in this paper will discuss these common elements as well as talk about a few other characteristics of components which don't apply to all components.

1.4.5 Components properties

Properties provide an extension of an object's fields. Unlike fields, properties do not store data: they provide other capabilities. For ezample, properties may use methods to read or write data to end object field to which the user has no access. Thisadds a certain level of protection as to how a given field is assigned data. Properties also cause "side affects" to ocur when the user makes a particular assignment to the property. Thus what appears as a simple field assignment to the component user could trigger a complex operation to occur behind the scenes.

1.5 Types of properties

Properties can be of the Standard data types defined by the Object Pascal rules. Property types also determine how they are edited in Delphi's Object Inspector. The table below shows the different property types as they are defined in Delphi's online help.

1.6 Methods

Since components are realy just objects, they can have methods. We will discuss some of the more commonly used methods later in this paper when we discuss the different levels of the VCL hierarchy.

1.7 Events

Events provide a means for a component to notify the user of some pre-defined occurrence within the component. Such an occurrence might be a buton click or the pressing of a keyboard.

Components contain special properties called events to which the component user assigns code. This code will be executed whenever a certain event occurs. For instance, if you look at the events page of a TEdit component, you'll see such events as OnChange, OnClick and OnDblClick. These events are nothing more than pointers to methods.

When the user of a component assigns code to one of those events, the user's code is referred to as an event handler. For exaple, by double clicking on the events page for a particular event causes Delphi to generate a method and pşaces you in the Code Editor where you can add your code fort hat method.

Note the use of the of object specification. This tells the compiler that the procedure definition is actually a method and performs some additional logic like ensuring that an implicit Self parameter is also passed to this method when called. Self is just a pointer reference to the class to which a method belongs.

1.8 Containership

Some components in the VCL can own other components as well as be parents to other componets. These two concepts have a diffirent meaning as will be discussed in the section to follow.

1.9 Ownership

All components may be owned by other components but not all components can own other components. A component's Owner property contains a reference to the component which owns it.

The basic responsibility of the owner is one of resource management. The owner is respansible for freeing those components which it owns whenever it is destroyed.

Typically, the form owns all components which appear on it, even if those components are placed on another component such as a TPanel. At design-time, the form automatically becomes the owner for components which you place on it.At run-time, when you create a component , you pass the owner as a parameter to the component's constructor. For instance, the below shows how to create a Tbutton component at run-time and passes the form's implicit Self variableto the TButton's Create constructor.TButton .Create will then assign whatever is passed to it.In this case Self or rather the form and assign it to the button's Owner property.

MyButton:=TButton.Create(self):

When the form that now owns this TButton component gets freed, MyButton will also freed.You can create a component without an owner by passing nil to the component's create constructor,however you must ensure that the component is freed when it is no longer needed.

1.10 Parenthood

Parenthood is amuch different concept from ownership. It applies only to windowed components, which can be parents to other components.

Parent components are responsible for the display of other components. They call the appropriate methods internally that cause the children components to draw themselves. The parent property of a component refers to the component which is its parent. Also a component's parent does not have to be it's owner. Although the parent component is mainly responsible for the display of components. It also frees children components when it is destroyed.

1.11 Some Manipulations in DELPHI

Changing the form size with a program

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

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

Changing button dimensions

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

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

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

```
procedure TForm1.Button1Click(Sender: TObject);
begin
Form1.Height := 200;
Form1.Width := 130;
Button1.Height := 30;
Button1.Width := 60;
Button2.Height := 30;
Button2.Width := 60;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
Form1.Height := 300;
Form1.Width := 400;
Button1.Height := 60;
Button1.Width := 120;
Button2.Height := 60;
Button2.Width := 120;
end;
```

Change a button's Caption in a program

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

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

Changing the WindowState property

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

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

2. MICROSOFT ACCESS

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

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

2.1 Uses

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

Some professional application developers use Access for rapid application development, especially for the creation of prototypes and standalone applications that serve as tools for on-the-road salesmen. Access does not scale well if data access is via a network, so applications that are used by more than a handful of people tend to rely on a Client-Server based solution such as Oracle, DB2, Microsoft SQL Server, PostgreSQL, MySQL, MaxDB, or Filemaker. However, an Access "front end" (the forms, reports, queries and VB code) can be used against a host of database backends, including Access itself, SQL Server, Oracle, and any other ODBC-compliant product.

This approach allows the developer to move a matured application's data to a more powerful server without sacrificing the development already in place.

2.2 Features

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

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

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

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

2.3 Development

The programming language available in Access is, as in other products of the Microsoft Office suite, Microsoft Visual Basic for Applications. Two database access

libraries of COM components are provided: the legacy Data Access Objects (DAO), only available with Access, and the new ActiveX Data Objects (ADO).

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

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

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

2.4 Relational Database

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

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

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

2.5 Database Management System (DBMS)

A database is an organized collection of data. Organization means method, it assumes discipline, it also anticipates efficient manner in using that information. Unless you are creating small applications for your personal use, you will usually need to share your data either with other people (users, database developers, etc) or other machines. To make your job easier, Microsoft Access provides in one package the database

information and the tools you need to use your database. To be organized, you will divide your database in different related parts. The method of management you will use makes Microsoft Access a Database Management System (DBMS). When you create a database in Microsoft Access, you create a file that will include different parts of your database. These are referred to as tables queries, forms, reports, etc.

2.6 Microsoft Access as a software product

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

Although Microsoft Office 97 and Microsoft Office 2000 get installed in the C:\Program Files\Microsoft Office folder, they treat the shortcuts that launch them differently. The applications that are part of Microsoft Office 97 designate their shortcuts with full names and these are installed in the Microsoft Office folder. Microsoft Office 2000 (Premium) uses shortcut names to designate its shortcuts and they are installed in the Microsoft Office\Office folder. This means that you could launch an application from Windows Explorer or My Computer. Therefore, in order to launch Microsoft Access, locate its shortcut in Windows Explorer or My Computer and double-click it.If you have a Microsoft Access database such as an E-Mail attachment, a file on a floppy disk, on the network, or in any other means, once you see its icon, you can double-click it. Not only will this action launch Microsoft Access, but also it will open the file.You can also launch Microsoft Access from a shortcut. If you happen to use the software on a regular basis, you can create a shortcut on your desktop or on the Quick Launch area. Many users also take advantage of the Microsoft Office Shortcut Bar. Sometimes, the icon you need will not be there; in that case you should insert it manually.

When Microsoft Access starts, you are presented with a special dialog box that inquires about your intentions. From this dialog, you can do one of four things: create a

database using one of the sample files, create a database from scratch, open an existing database, or open a "raw" program

2.7 Microsoft Access Interface

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

The Title Bar

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

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

The Main Menu

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

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

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

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

2.8 Microsoft Access Database Objects

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

2.8.1 Database Objects: Tables

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

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

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

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

2.8.2 Database Objects: Queries

A table can be large depending on the information it holds. To further organize your data, you should be able to retrieve necessary information for a specific purpose. The solution is to create a query (or queries) in order to limit part of the data in a table for a specific goal, for better management or search. That's the role of a query. Just like tables, queries are managed from the Database Window in their own category. And you would open a query the way you open a table.

2.8.3 Database Objects: Forms

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

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

2.8.4 Database Objects: Reports

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

3. DATABASE TABLES AND RELATIONSHIPS

I used Microsoft Access for my program's database. It's named Video.mdb. I created six tables that are shown below for the process. And than I connected them together for the process.

3.1 Database Tables

A table is a grouping of releated data organized in fields (columns) and records (rows) on datasheet. A database may have one or more tables. Tables are grids that store information in a database.

First we introduce the following terms related to table design.

Field Name: This is the name of the field and should represent the contents of the field such as "Customer Name" , "Address",etc. The name can not exceed 64 characters in length and may include spaces.

Data Type is the type of value that will be entered into the fields. Now we'll talk about Data Type

Text You may type in any alphabetical/numerical data that you desire - up to a maximum of 255 characters. As indicated, this is a text field, so you can't do mathematical calculations. Examples of Text data are: names, addresses, stock numbers, room numbers, zip codes, etc.

Memo This field is for lots of text. You can have up to 32,000 characters.

Number This field is for numbers where you want to add, subtract, multiply, divide, average, and do numerical calculations. This field can be a very large size, so when we get to Field Properties, we'll talk about "sizing" this field so it doesn't take up to much "space" in storage.

Date/Time Dates and Times. You may format these later, as you may desire.

Currency Dollars ($). You may format these later, as you may desire.

AutoNumber This field is an "automatic" counter that assigns a number each time you put data into a new field.

Yes/No This is a "True/False" or "Yes/No" type of field.

OLE Object This means "Object Link Embedding" which indicates you can insert a graphic, picture, sound, etc. Pretty neat to put a photograph in a personnel record or a picture of an inventory item in the stock record (advanced stuff).

Hyperlink A hyperlink will link to an internet or intranet site, or location in the database (URL).

Every record in a table must have a primary key that differentiates it from every other record in the table. A customer id number is an example of a record whose values will only appear once in a database table.

If none of existing fields in the tablewill produce unique values for every record, a seperate field must be added. Access will prompt you to create this type of field at the beginning of the tablet he first time you save the table and a primary key field has not been assigned. The field is named "Customer Id" and the data type is "auto number". Since this extra field serves no purpose to you as the user, the auto number type automatically updates whenever a record is added so there is no extra work on your part.

Table Customers

Customers : Tablo

Alan Adı	Veri Türü	Tanım
CustomerId	Otomatik Sayı	
CustomerName	Metin	
CustomerSurname	Metin	
Telephone	Metin	
Address	Metin	
City	Metin	
Country	Metin	

Alan Özellikleri

Genel | Arama

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

Figure 3.1 Table Of Costumers

Table Suppliers

Suppliers : Tablo

Alan Adı	Veri Türü	Tanım
SupplierId	Otomatik Sayı	
SupplierName	Metin	
Telephone	Metin	
Address	Metin	
City	Metin	
Country	Metin	

Alan Özellikleri

Genel | Arama

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

Figure 3.2 Table of Suppliers

Table Videos

Videos : Tablo

Alan Adı	Veri Türü	Tanım
VideoId	Otomatik Sayı	
SupplierId	Sayı	
VideoName	Metin	
VideoDesc	Not	
VideoDirector	Metin	
VideoType	Metin	
VideoYear	Sayı	
UnitPrice	Para Birimi	
VideoInStock	Sayı	
VideoPicture	OLE Nesnesi	

Alan Özellikleri

Genel | Arama

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

Figure 3.3 Table of Videos

Table Rental

Rental Tablo

Alan Adı	Veri Türü	Tanım
RentalId	Otomatik Sayı	
VideoId	Sayı	
CustomerId	Sayı	
HireDate	Tarih/Saat	
BringingDate	Tarih/Saat	
Quantity	Sayı	
Price	Para Birimi	
Turned	Evet/Hayır	

Alan Özellikleri

Genel | Arama

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

Figure 3.4 Table of Rental

Table Sales

Sale : Tablo

Alan Adı	Veri Türü	Tanım
SaleId	Otomatik Sayı	
VideoId	Sayı	
CustomerId	Sayı	
SaleDate	Tarih/Saat	
Quantity	Sayı	
Price	Para Birimi	

Alan Özellikleri

Genel | Arama

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

Figure 3.5 Table of Sales

Table Lock

Lock : Tablo

Alan Adı	Veri Türü	Tanım
UserName	Metin	
Pass	Metin	

Alan Özellikleri

Genel | Arama

Alan Boyutu	10
Biçim	
Giriş Maskesi	
Resim Yazısı	
Varsayılan Değer	
Geçerlilik Kuralı	
Geçerlilik Metni	
Gerekli	Hayır
Sıfır Uzunluk İzni	Evet
Sıralı	Evet (Yineleme Yok)
Unicode Sıkıştırma	Evet
IME Modu	Denetim Yok
IME Tümce Modu	Hiçbiri
Akıllı Etiketler	

Figure 3.6 Table of Lock

3.2 Relationships between Tables

The database structure of my program is given **Figure 3.7**. There are many relations between tables.

- Between Sale Table and Customers Table the costumerid is index field
- Between Sale Table and Videos Table the indexed field is saleid
- Between Rental Table and Videos Table the rentalid is index field
- Between Videos Table and Suppliers Table the index field is videoid

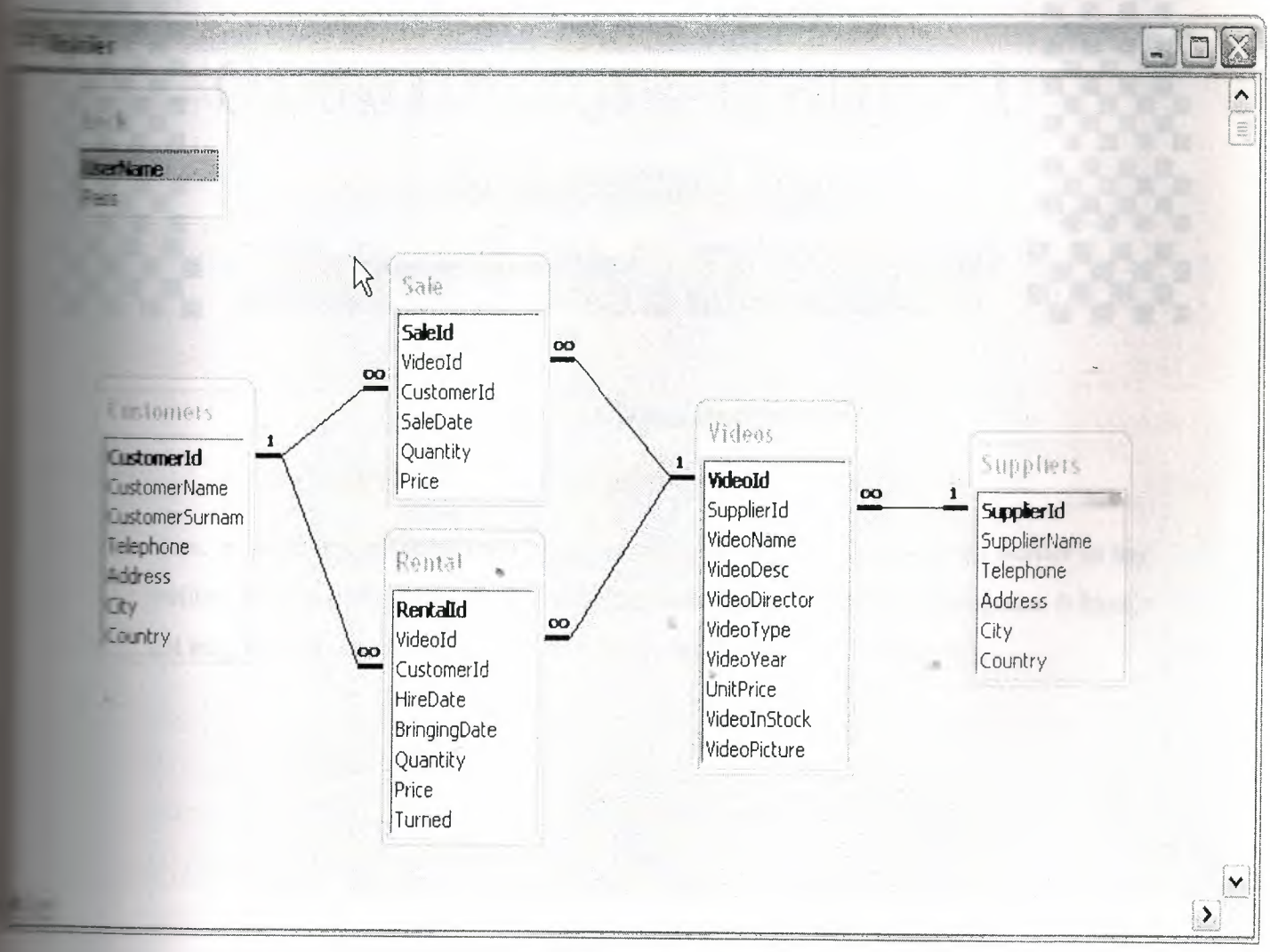

Figure 3.7 Relationships of Tables

4.VIDEO STORE MANAGEMENT PROGRAM

In this chapter the screenshots and the user manual of the program seen explained. I will explain in detail how the system works as the user moves from process to process. The system begins with the wellcome screen that is explained below:

Figure 4.1 Wellcome Screen

Here, in wellcome screen form I used one form and one picture with no border to say wellcome to my program. I designed picture with AAA Logo design program. It hasn't got any important mission. I added this form only for program's visuallity.

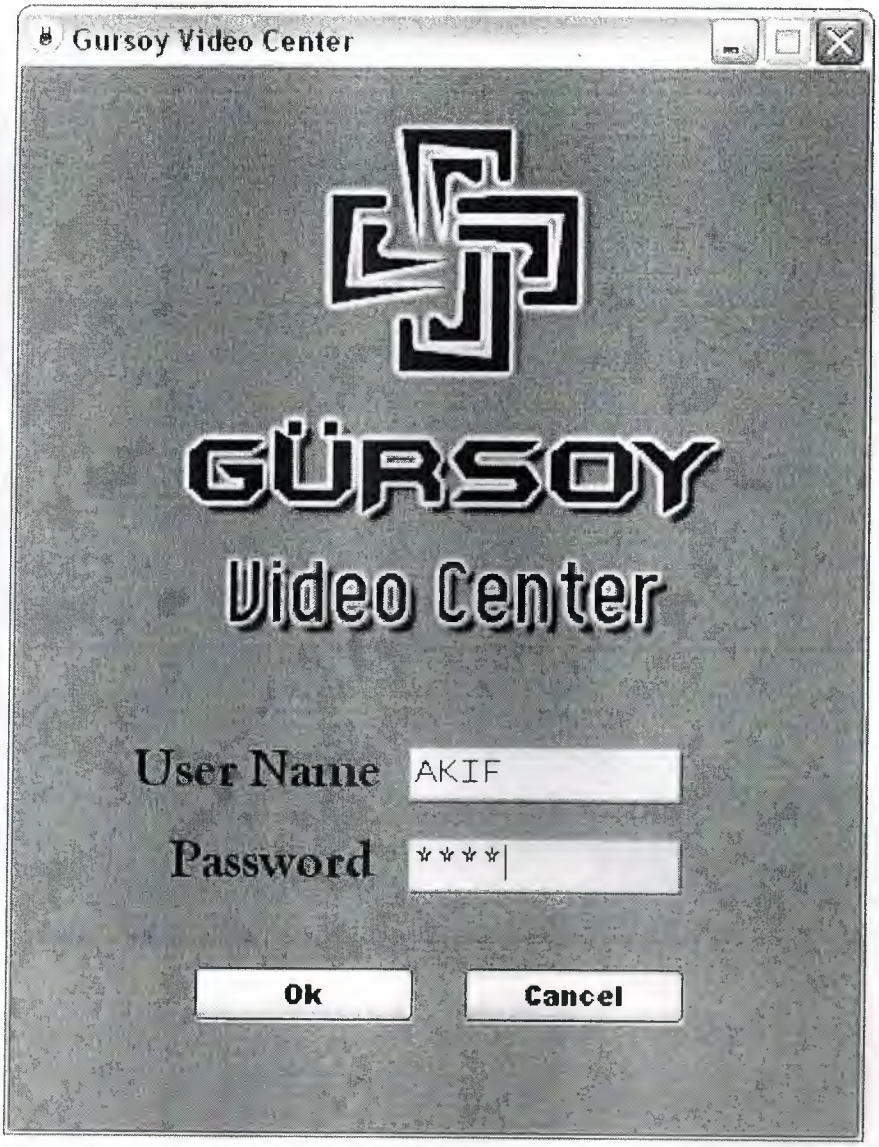

Figure 4.2 Password Screen

Here, I decided to add a password form to my program for security. For enter to the program, you must have an authority. So you must have a User Name and Password for entrance to the program. If you are an unauthority, you can not enter the program. You have got 3 right for the enter. If you enter 3 times wrong password or User Name, the program will be terminated automatically. In my project I add 2 authorities for entrance. User Name : ADMIN Password : ADMIN

User Name: AKIF Password: 1234

You can change, delete or add authority in the program with other form. I will explain next.

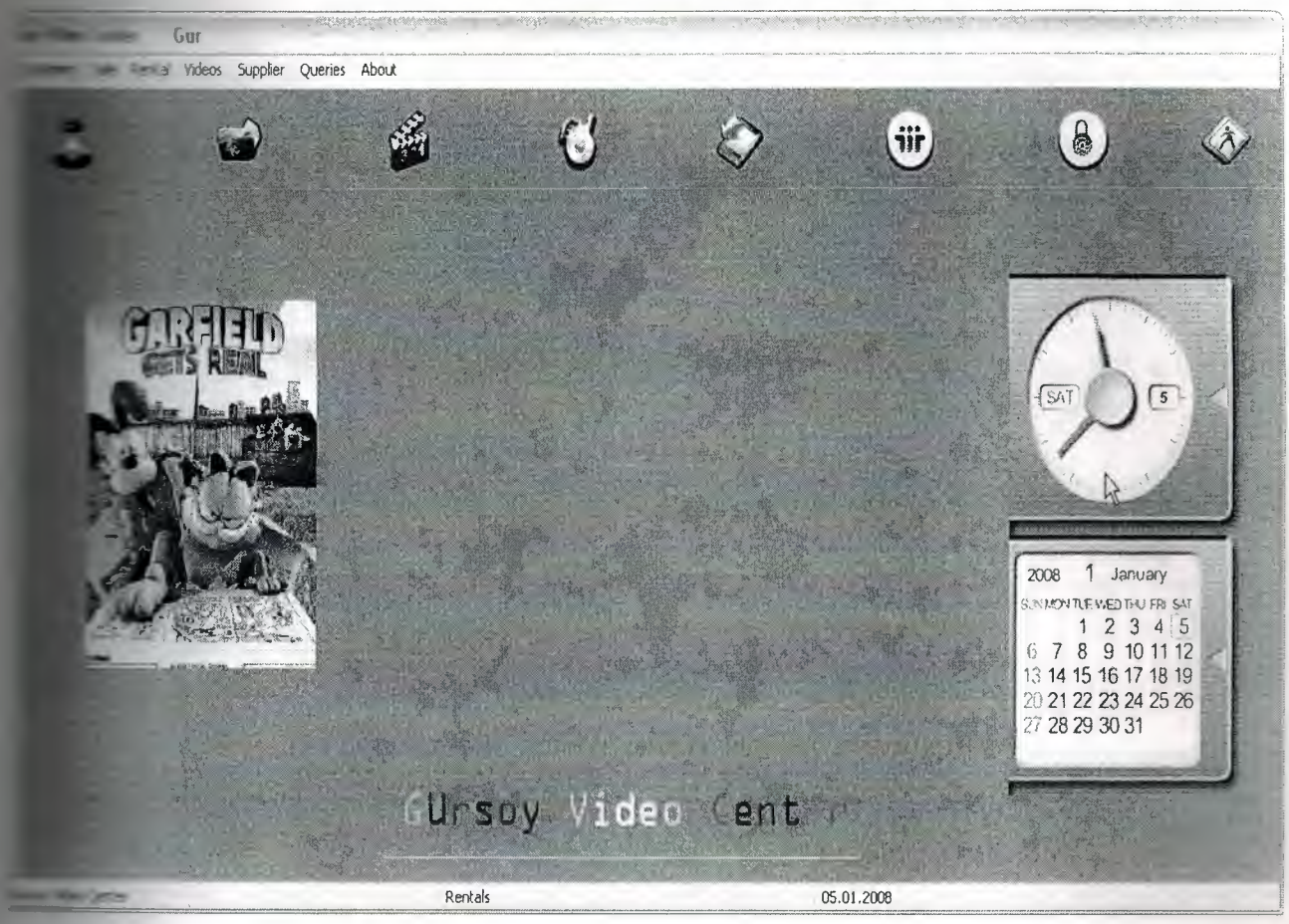

Figure 4.3 Main Form

If you enter the user name and password correctly, you can show the main menu. In this main menu for clock and calendar, Picture slayt show and video store name, I add a new active-x component to Delphi. So the main menu is very animated for visuallity. These are user friendly interface elements. The Calendar shows the current date. The watch is dynamic and shows the computers current time and the slayt shows the films Posters. There is a main menu above, you can reach to all other process with. And then I used visual buttons to reach other process easily. What you want to select you can just click.

Figure 4.4 Customers Form

When you click customers from main menu or under visiual pictures menu, you go to customers menu. In this menu you can make operations related with customers. These operations are;

- All Customer (Shows every registered Customers of the Store)
- Customer Operations (You can add/delete new customers in this section or do such editings to customers)
- Customer Search (You can search registered customers by their name, surname, city, country)

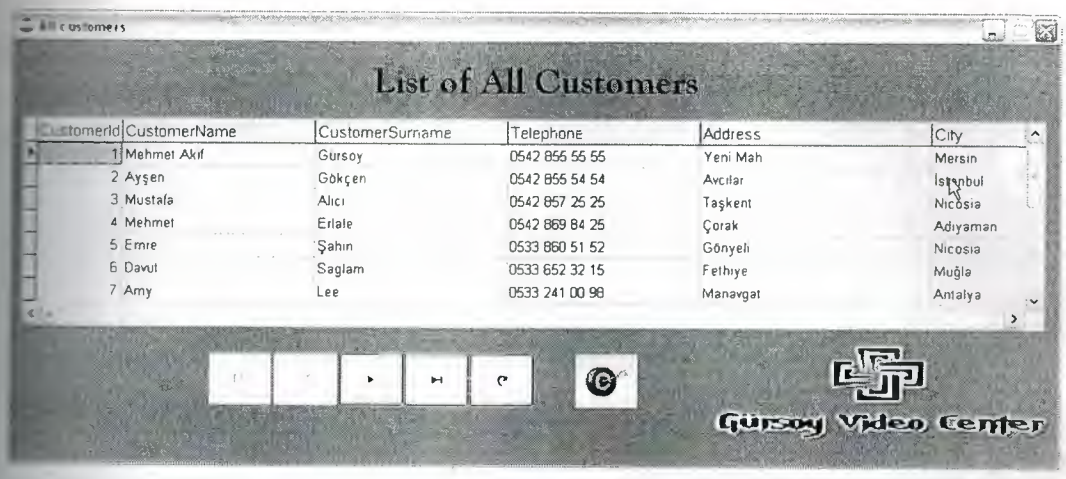

Figure 4.5 List of All Costumers Form

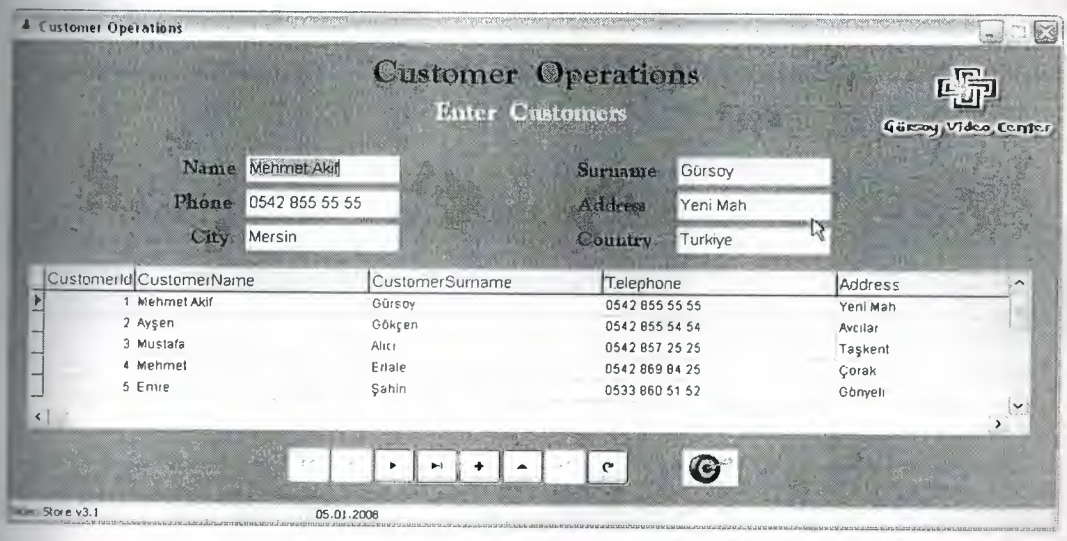

Figure 4.6 Customer Operations Form

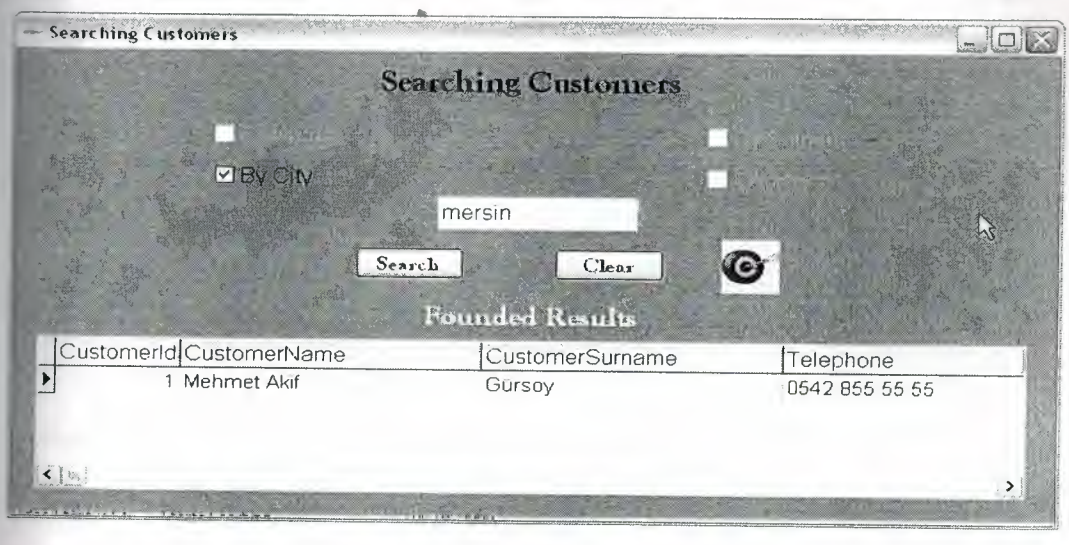

Figure 4.7 Searching Customer Form

Figure 4.8 Products Form

When you click videos from main menu or under visiual pictures menu, you go to products menu. In this menu you can make operations related with videos.

In this menu you can make operations related with videos. These operations are:

- All Videos (Shows every registered Videos of the store)
- Video Operations (This section also separated in two parts these parts are

a)Adding New Video [on this section you can add new videos or do such editings to existing videos excepting editing amount and price]

b)Video Delete (This section lets us to delete selected video but the video in stock has to be " 0 " for delete operation)

c) Purchase An Existing Product [on this section you can edit selected videos amount or price])

- Video Search (You can search registered videos by their name, director, type and year)

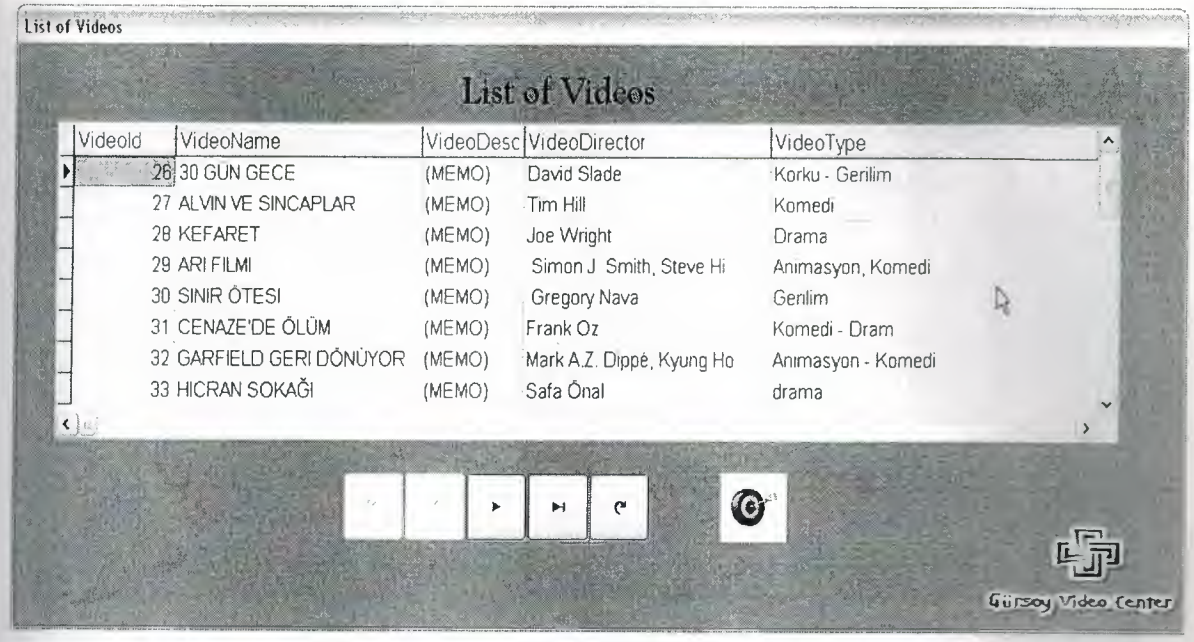

Figure 4.9 List of Videos Form

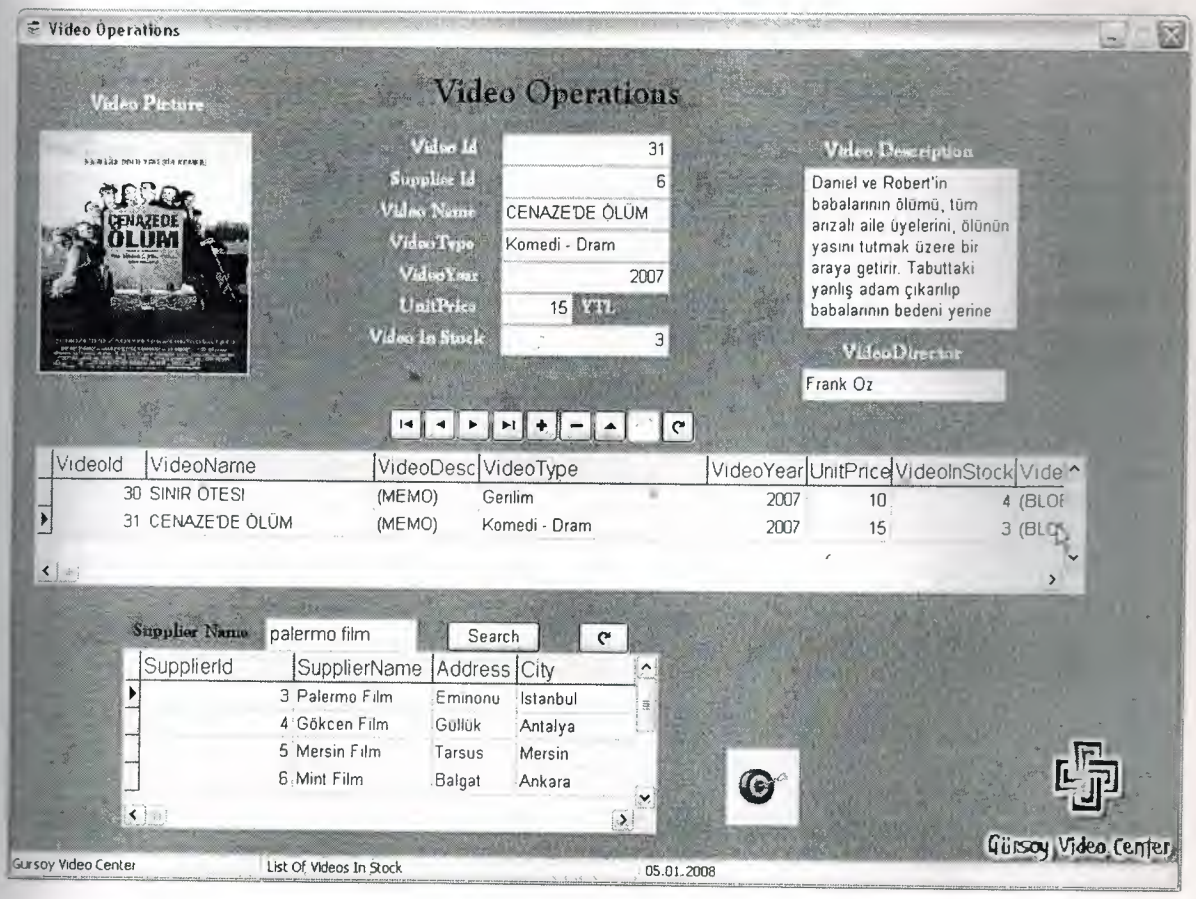

Figure 4.10 Video Operations

Figure 4.11 Purchase Existing Video Form

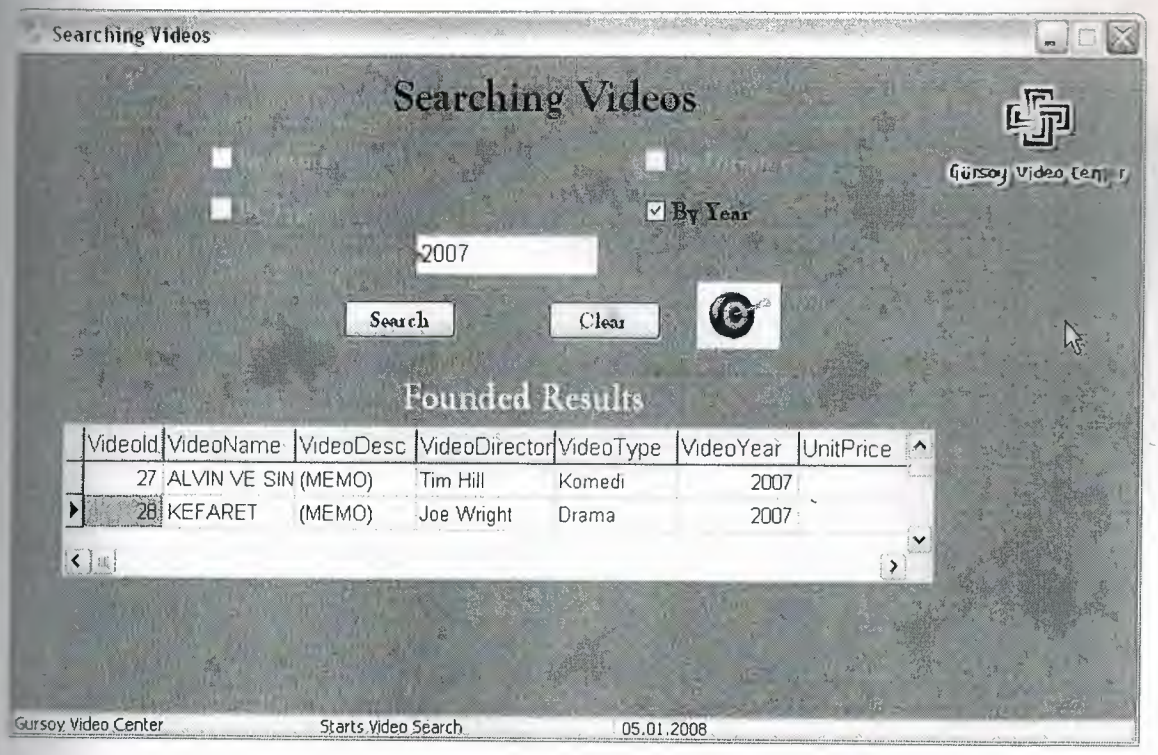

Figure 4.12 Searching Videos

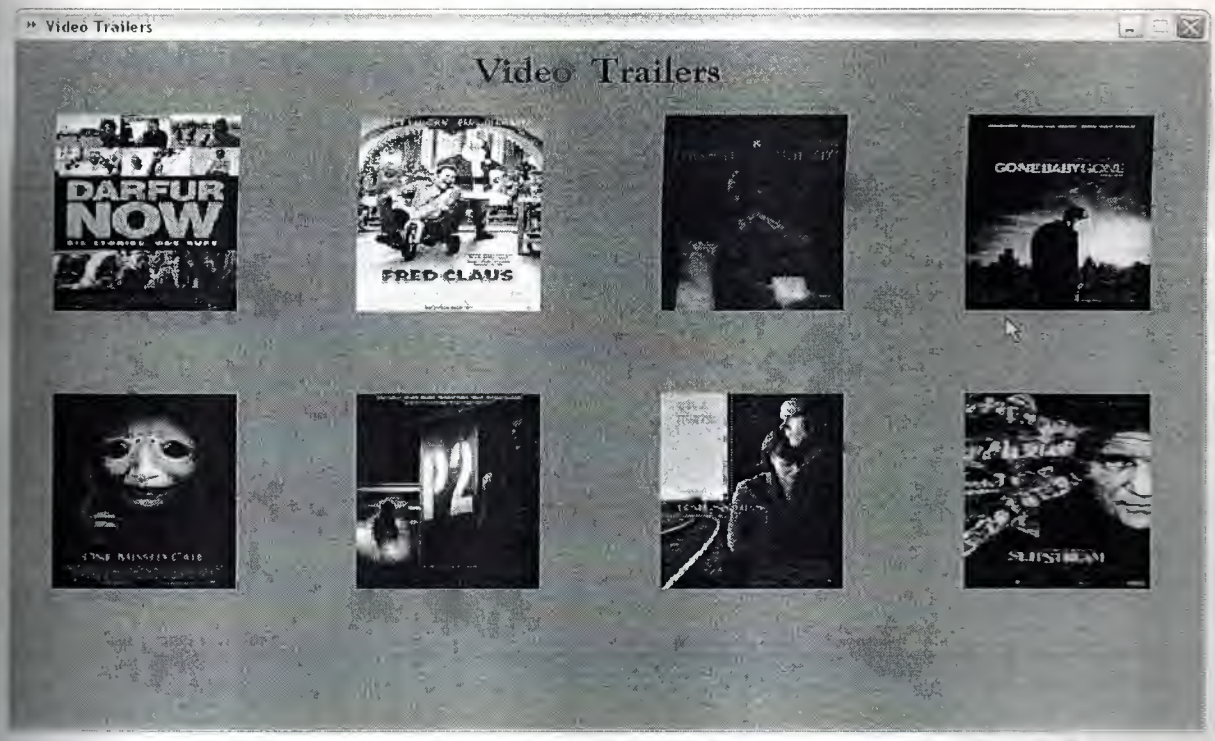

Figure 4.13 Video Trailers Form

In this form, if you want to know anything about video, you can watch video tailers. Only you click the film's pictures, which you want to watch. And than the video trailers will be shown in media player with audio like shown below.

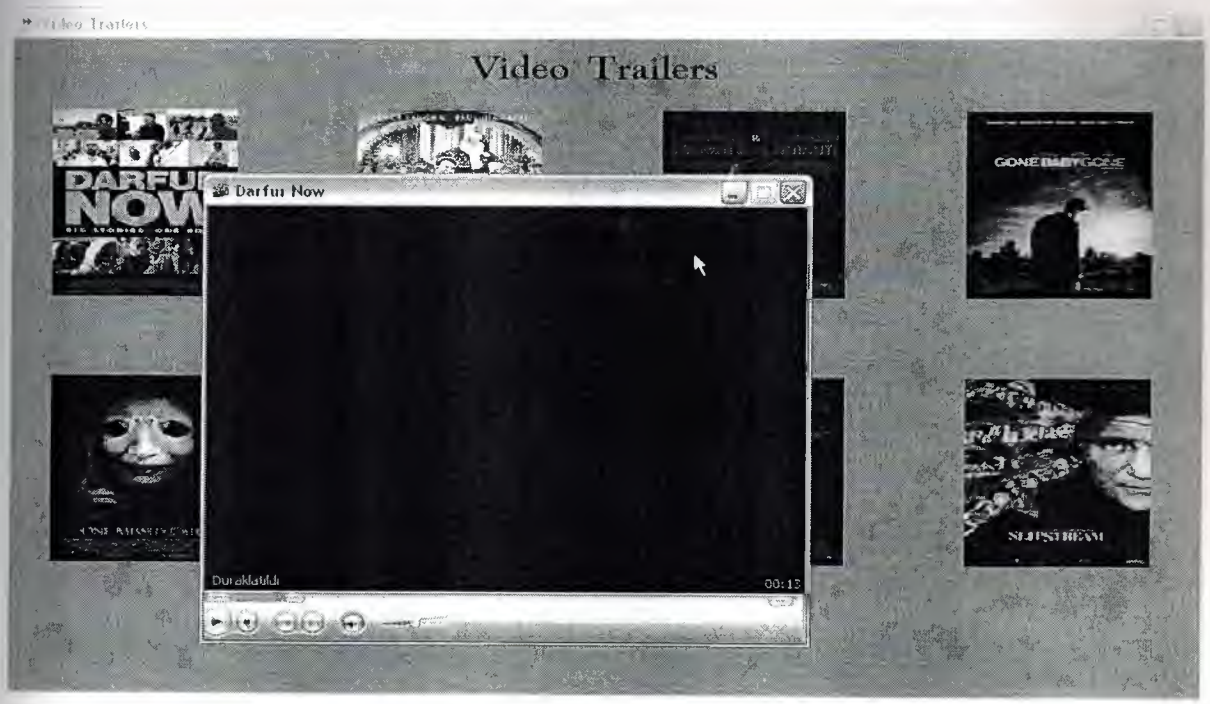

Figure 4.13 Watch Video Trailers with Media Player

Figure 4.14 Suppliers Form

When you click suppliers from main menu or under visiual pictures menu, you go to suppliers menu. In this menu you can make operations related with suppliers. These operations are;

- All Suppliers (Shows every registered Supplier)
- Supplier Operations (On this section you can add or delete suppliers (but supplier's video in stock must be zero.))
- Supplier Search (You can search registered suppliers by their name and city)

Figure 4.15 List Of Suppliers Form

Figure 4.16 Supplier Operations

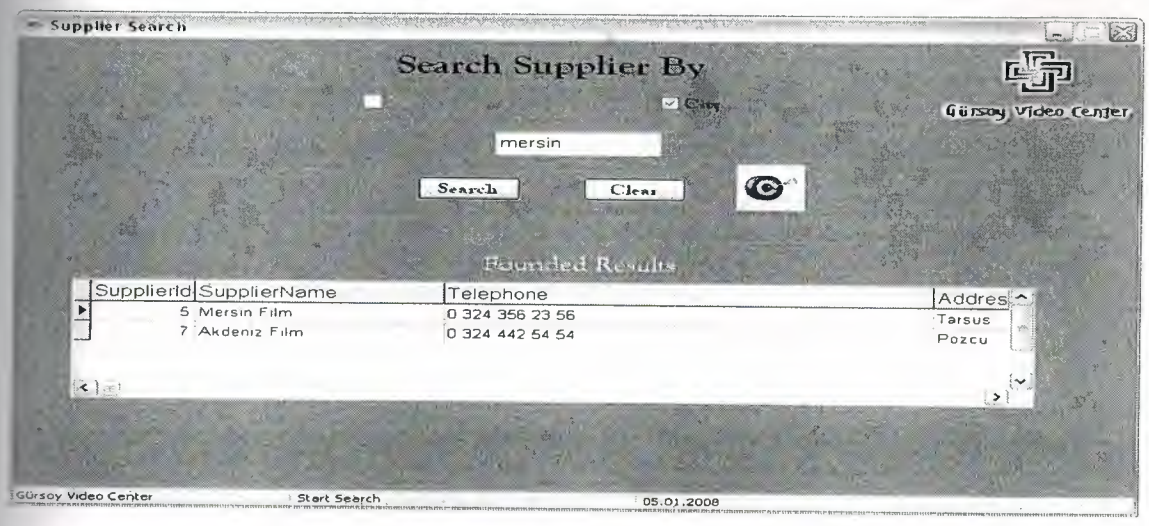

Figure 4.17 Supplier Search Form

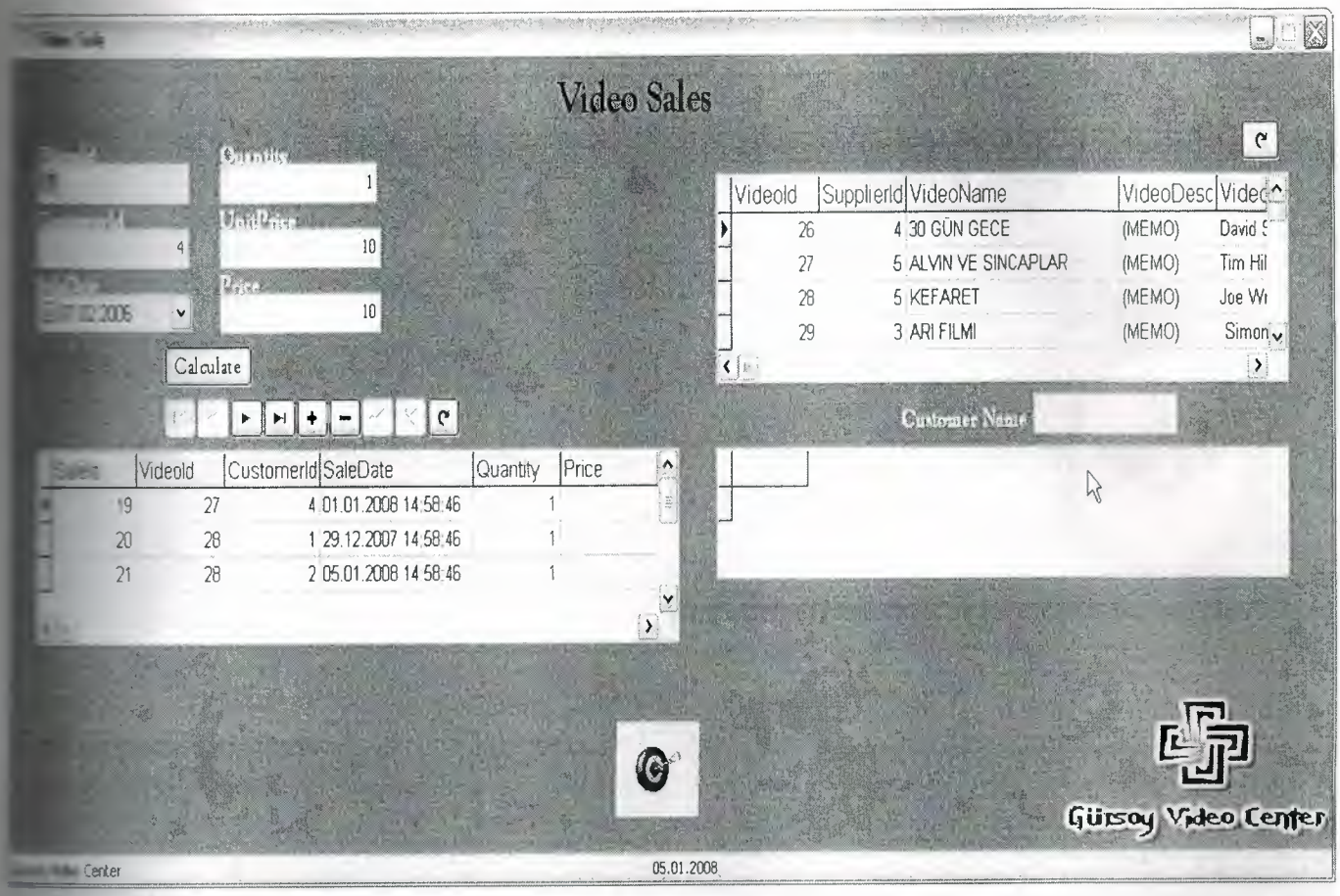

Figure 4.18 Video Sales

When you click sale from main menu or under visiual pictures menu, you go to sale form. In this form you can sale product easily like this:

- You have to select video from right side of the page and than the price will automatically appears on the Unit Price
- After selecting the video than you have to enter videos id.
- Than comes the Customer ID box you can enter the id of customer but if you don't know the id of customer you can make a search on the bottom right of sale page.
- And of course you have to select date part.
- Enter the Video's Quantity which is on sale.
- And As you see the price automatically added while you select the video before.
- Then press calculate button to calculate the Total Price of Sale.
- And last step press '+' button to add your sale on your video store database.

Figure 4.19 Rental Form

When you click rental from main menu or under visiual pictures menu, you go to rental form. On this Section you will see that 2 choices are on the page. These one of them is 'New Rental' and the other one is 'Turned Rentals'. Lets explain what are their process.

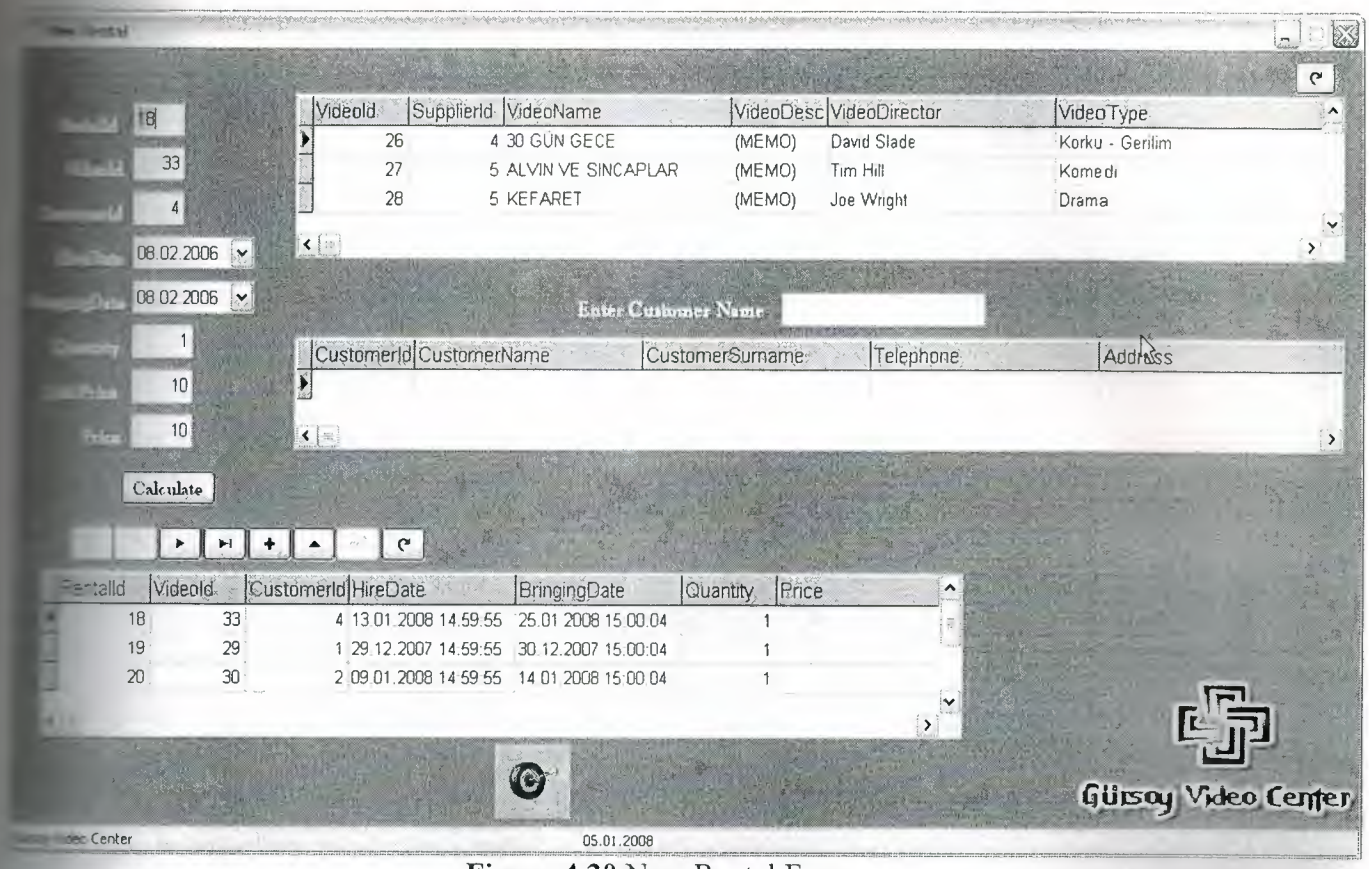

Figure 4.20 New Rental Form

On this part you can make rentals to the customers. These are steps for to make a rental.

- You have to select video from right side of the page and than the price will automatically appears on the Unit Price
- After selecting the video than you have to enter videos id.
- Than comes the Customer ID box you can enter the id of customer but if you don't know the id of customer you can make a search on the bottom right of rental page.
- And of course you have to select Hire Date and Bringing Date
- Enter the Video's Quantity which is on rent.
- And As you see the price automatically added while you select the video before.
- Then press calculate button to calculate the Total Price of Rental.
- And last, press '+' button to add your rental on your video store database.

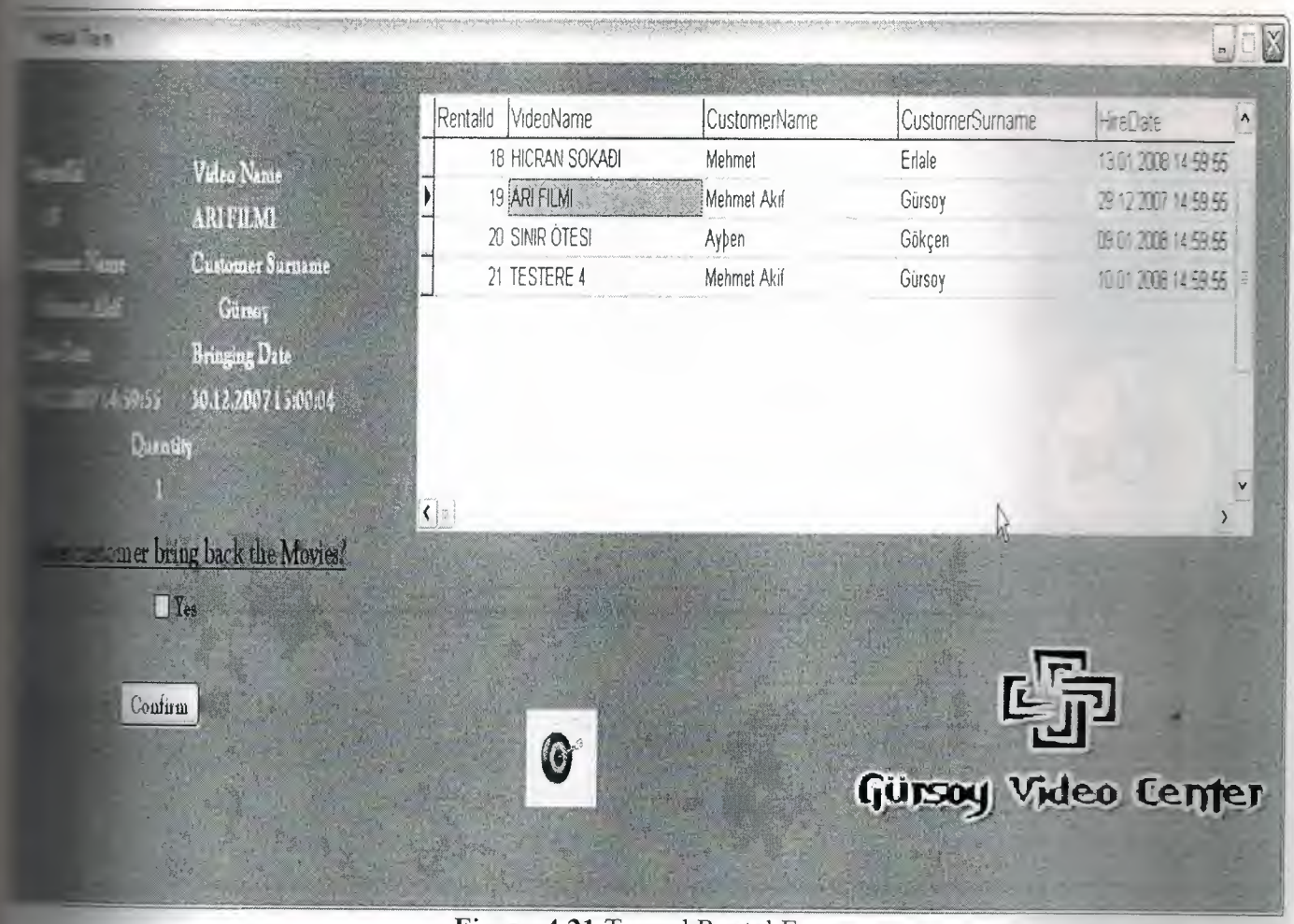

Figure 4.21 Turned Rental Form

In this part, you can confirm that the rental video turned or not turned. First select the video which has returned and than click on the box the and click to confirm button to confirm that the rented video turned this what you do on the rented video turns

Figure 4.22 Queries Form

When you click queries from main menu or under visiual pictures menu, you go to queries form. Here you can interrogate about products, sales, rentals and dates.

Figure 4.23 Search For Bringing Date Form

You just need to enter the beginning bringing date and the ending bringing date as 'dd.mm.yyyy' and after that click on the 'Search' button which will list the founded results. 'Clear' button clears the page

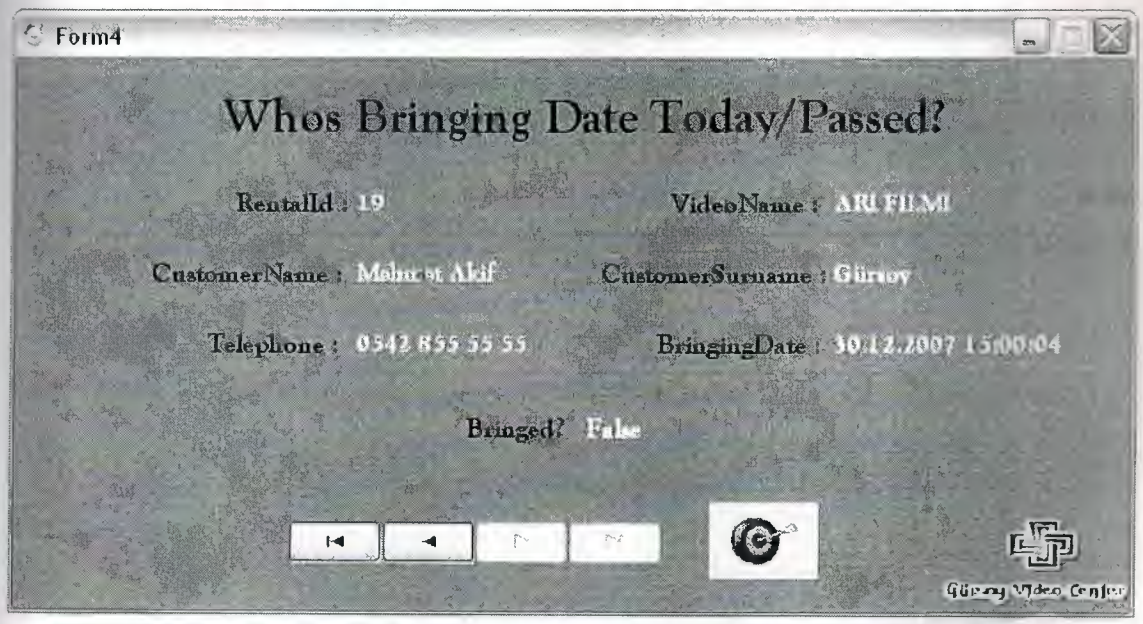

Figure 4.24 Bringing Date Form

This page, we find which customers bringing date passed or today.

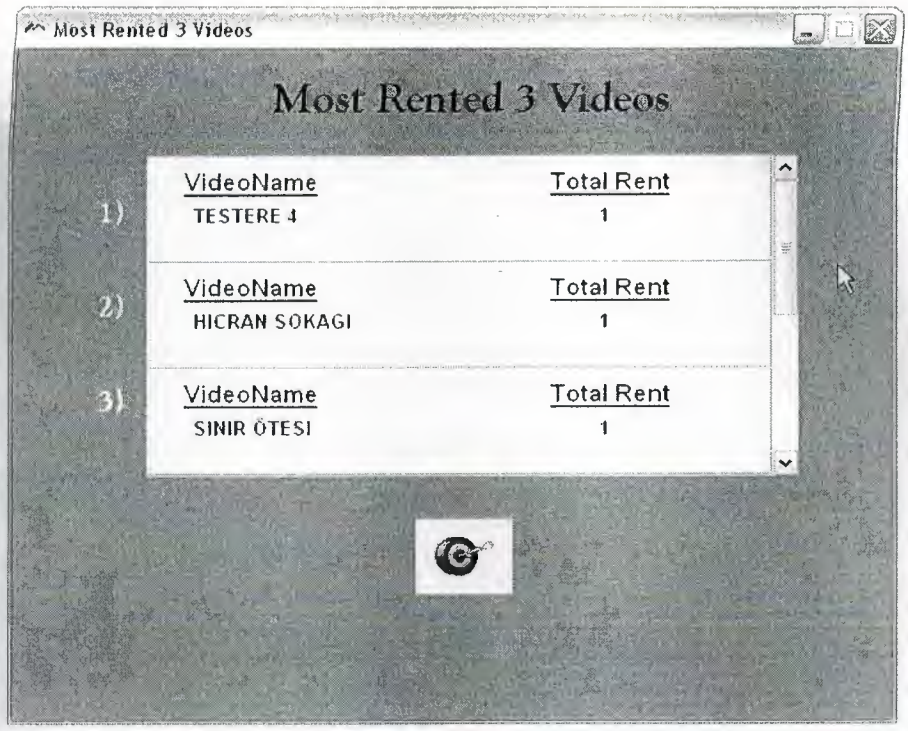

Figure 4.25 Most Rented Products Form

This form show us, which videos are most rented before.

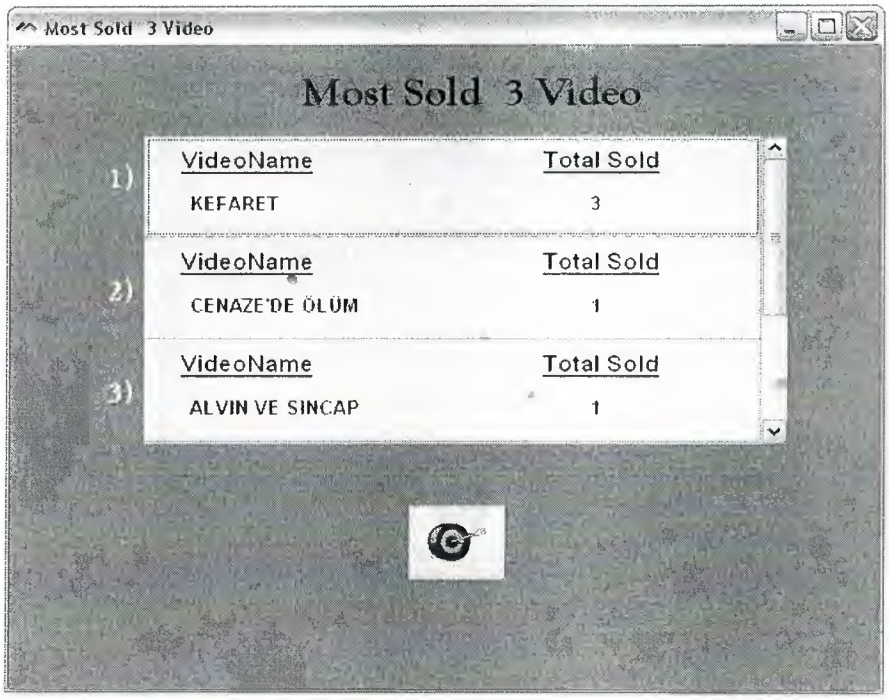

Figure 4.26 Most Sold 3 Video

This form show us, which videos are most sold before.

Figure 4.27 Security Process Form

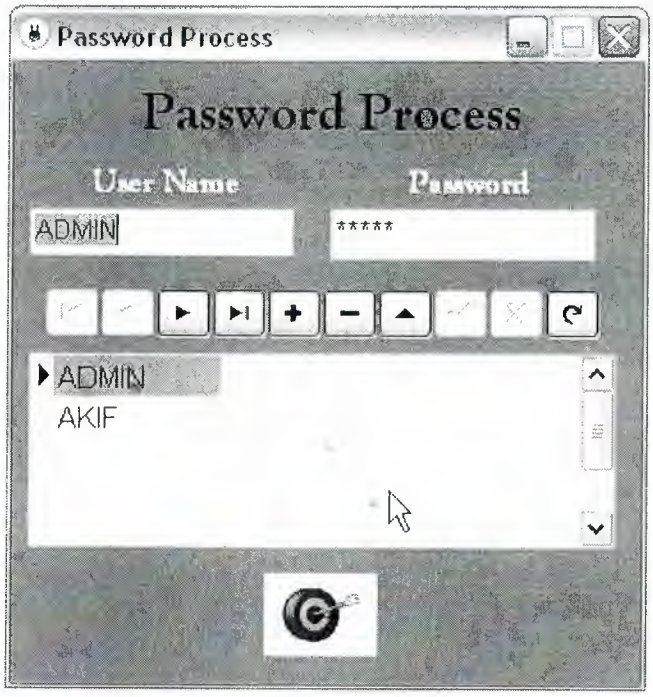

Figure 4.28 Admin Panel Form

In this form only authorized person/s may enter. Authorized person/s may register new program user or may delete existing program user/s from database or update existing user/s.

Figure 4.29 About Me Form

Here, I add a form named "About Me" to show that who is the designer and some explanations about the designer.

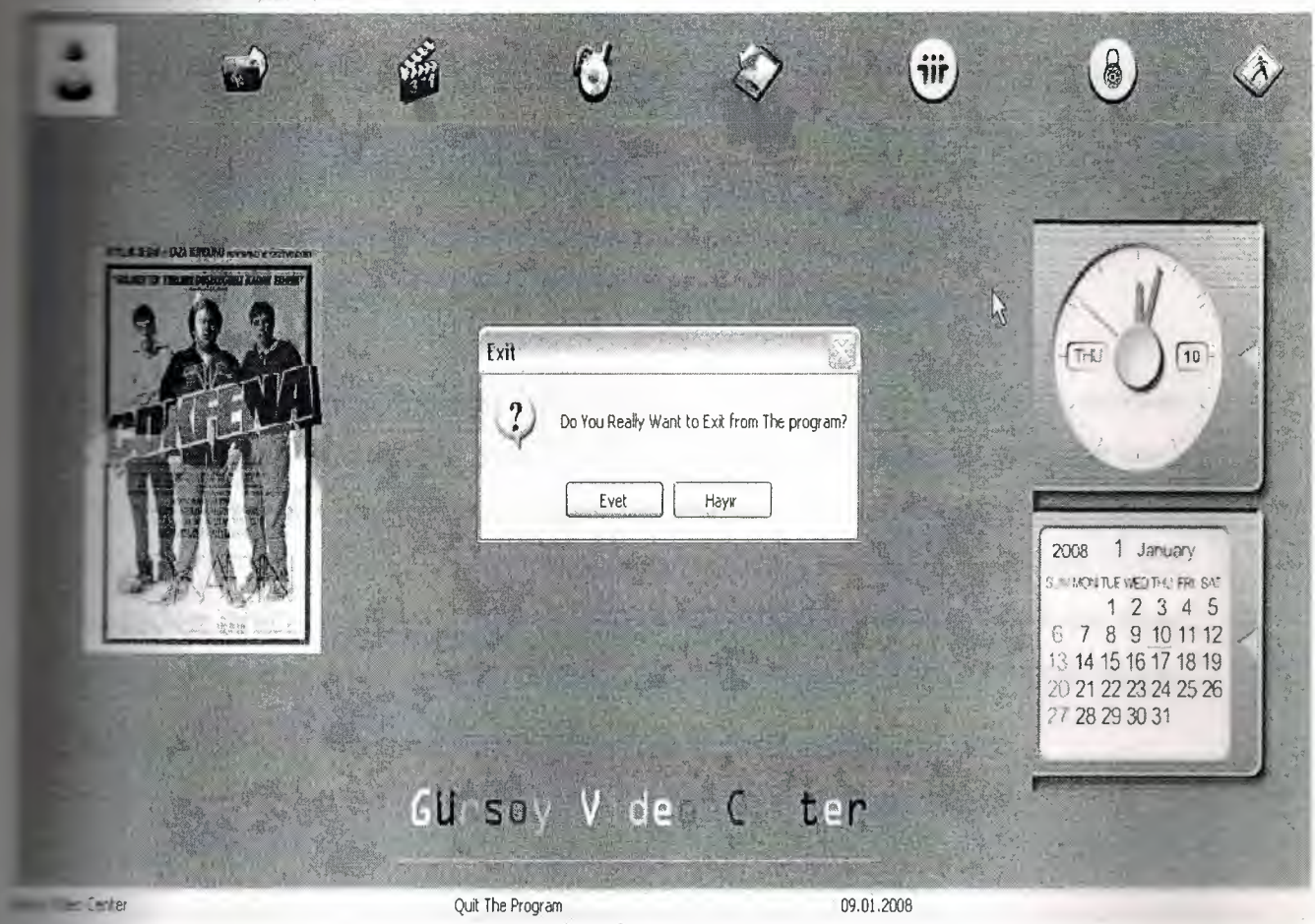

Figure 4.30 Exit Form

Finally we came the last form for this program. In this form when you click the exit button, the program ask you the question "Do you really want to exit?". If you click "Yes" (Evet) button, you exit to program, if you click "No" (Hayır) button, the program will be continue.

CONCLUSION

Borland Delphi is one of the best well known programming languages based on window's environment. Before I did this project, I didn't know anything about windows oriented programmin and it came to me too hard. But when I started to learn Delphi and started to this project I learnt it is easier and flexible than the other programming languages. I learnt many things about visual programming by Delphi and I decided to work on visual programming in future.

Delphi is easy and flexible but I think it is not enough for big project. Anyway Microsoft has bought Delphi. I think it will be Delphi's end. Because Microsoft has a Visual Program too. It is Visual Studio.Net. So I think Microsoft bought delphi only for finish it. That' s so I want to leave working with delphi. I decided to work with Visual Studio.Net for my next project. But I must say this, delphi is more helper for me about window's oriented programming. This project became a good exercise to me for my next programs.

REFERENCES

[1]. www.marcocantu.com

[2]. www.delphi.about.com

[3]. www.access-programmers.com

[4]. www.delphiturk.com

[5]. http://www.microsoft.com/mspress/books/5054.asp

[6]. Jones Dana and Novalis Susan, "Access 2002 VBA Handbook", published by SYBEX, 2002.

[7]. Memik Yanık, "Borland Delphi 8" , published by Seçkin,2007

5. APPENDIX

5.1 The Codes of "Main" Form

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, XPMan, ComCtrls, Menus, Buttons, DBCtrls,
  DB, ADODB, jpeg, OleCtrls, ShockwaveFlashObjects_TLB;

type
  TForm1 = class(TForm)
    StatusBar1: TStatusBar;
    XPManifest1: TXPManifest;
    Button9: TButton;
    GroupBox1: TGroupBox;
    Button10: TButton;
    Button12: TButton;
    GroupBox2: TGroupBox;
    Button15: TButton;
    Button16: TButton;
    Button18: TButton;
    GroupBox3: TGroupBox;
    Button23: TButton;
    Button24: TButton;
    Button26: TButton;
    MainMenu1: TMainMenu;
    Customers1: TMenuItem;
    Sale1: TMenuItem;
    Rental1: TMenuItem;
    Products1: TMenuItem;
    Supplier1: TMenuItem;
    About1: TMenuItem;
    GroupBox4: TGroupBox;
    Button29: TButton;
```

```
Button30: TButton;
Button31: TButton;
Button32: TButton;
Queries1: TMenuItem;
GroupBox5: TGroupBox;
Button11: TButton;
Button13: TButton;
AllCustomers1: TMenuItem;
CustomerOperations1: TMenuItem;
SearchingCustomers1: TMenuItem;
Sales1: TMenuItem;
Rentals1: TMenuItem;
urnedRental1: TMenuItem;
VideoOperations1: TMenuItem;
VideoOperations2: TMenuItem;
SearchingVideos1: TMenuItem;
ListOfSuppliers1: TMenuItem;
SupplierOperations1: TMenuItem;
SearchingSupplier1: TMenuItem;
Between2Dates1: TMenuItem;
BringingDatePassedAndToday1: TMenuItem;
SpeedButton1: TSpeedButton;
SpeedButton3: TSpeedButton;
SpeedButton5: TSpeedButton;
Timer1: TTimer;
Panel1: TPanel;
Timer2: TTimer;
ADODataSet1: TADODataSet;
ADODataSet1Deyim1: TIntegerField;
ADODataSet1VideoName: TWideStringField;
DataSource1: TDataSource;
Timer3: TTimer;
ADODataSet2: TADODataSet;
DataSource2: TDataSource;
N1: TMenuItem;
SearchingVideos2: TMenuItem;
```

```
RentalTurn1: TMenuItem;
N3MostRented1: TMenuItem;
N3MostSold1: TMenuItem;
AboutUS1: TMenuItem;
Image1: TImage;
ShockwaveFlash1: TShockwaveFlash;
SpeedButton4: TSpeedButton;
SpeedButton2: TSpeedButton;
SpeedButton6: TSpeedButton;
SpeedButton8: TSpeedButton;
SpeedButton7: TSpeedButton;
ShockwaveFlash2: TShockwaveFlash;
ShockwaveFlash3: TShockwaveFlash;
Button1: TButton;
VideoTrailer1: TMenuItem;
procedure FormCreate(Sender: TObject);

procedure Button9Click(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure Button12Click(Sender: TObject);
procedure Button15Click(Sender: TObject);
procedure Button16Click(Sender: TObject);
procedure Button18Click(Sender: TObject);
procedure Button23Click(Sender: TObject);
procedure Button24Click(Sender: TObject);
procedure Button26Click(Sender: TObject);
procedure Button29Click(Sender: TObject);
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button7MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button27MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button9MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button10MouseMove(Sender: TObject; Shift: TShiftState; X,
```

```
  Y: Integer);
procedure Button11MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button12MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button15MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button16MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button18MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button23MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button24MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button26MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button30Click(Sender: TObject);
procedure Button11Click(Sender: TObject);
procedure Button13Click(Sender: TObject);
procedure CustomerOperations1Click(Sender: TObject);
procedure Button14Click(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton3Click(Sender: TObject);
procedure SpeedButton3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton4Click(Sender: TObject);
procedure SpeedButton4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);

procedure Button31Click(Sender: TObject);
```

```
    procedure SupplierOperations1Click(Sender: TObject);
    procedure SearchingSupplier1Click(Sender: TObject);
    procedure Between2Dates1Click(Sender: TObject);
    procedure BringingDatePassedAndToday1Click(Sender: TObject);
    procedure N3MostRented1Click(Sender: TObject);
    procedure N3MostSold1Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure VideoTrailer1Click(Sender: TObject);
    procedure DeletingVideo1Click(Sender: TObject);

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

var
  Form1: TForm1;
implementation

uses Unit2, Unit3, Unit5, Unit6, Unit7, Unit8, Unit10,
  Unit14, Unit15, Unit16, Unit17, Unit19,
  Unit20, Unit21, Unit22, Unit23, Unit24, Unit4, Unit11, Unit12, Unit13,
  Unit25, Unit27;

{$R *.dfm}

procedure TForm1.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=350;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=300;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=300;
statusbar1.panels.items[0].Text:='Gursoy Video Center';
```

```
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);

form1.Caption :='  Gursoy Video Center   ';
button1.caption:='Video Trailers';
button9.caption:='All Customers';
button10.caption:='Customer Operations';
button12.caption:='Customer Search';
button15.caption:='All Videos';
button16.caption:='Video Operations';
button18.caption:='Video Search ';
button23.Caption :='All Suppliers';
button24.Caption :='Supplier Operations';
button26.Caption :='Supplier Search';
button29.Caption :='Between 2 Date';
button30.Caption :='Bringing Date';
button11.Caption :='New Rental';
button13.Caption :='Turned Rentals';
Button31.Caption :='Most Rented 3 Video';
Button32.Caption :='Most Sold 3 Video';

groupbox1.caption:='Customers';
groupbox2.caption:='Products';
groupbox3.caption:='Suppliers';
groupbox4.caption:='Queries';
groupbox5.caption:='Rental';

groupbox1.Visible:=False;
groupbox2.Visible:=False;
groupbox3.Visible:=False;
groupbox4.Visible:=False;
groupbox5.Visible:=False;

speedbutton4.Hint:='Customer Menu';
speedbutton1.Hint:='Sales';
speedbutton3.Hint:='Rentals';
speedbutton2.Hint:='Videos Menu';
```

```
speedbutton5.Hint:='Queries Menu';
speedbutton6.Hint:='Supplier Menu';
speedbutton7.Hint:='Exit';
speedbutton8.Hint:='Security Process';

speedbutton1.showhint:=true;
speedbutton2.showhint:=true;
speedbutton3.showhint:=true;
speedbutton4.showhint:=true;
speedbutton5.showhint:=true;
speedbutton6.showhint:=true;
speedbutton7.showhint:=true;
speedbutton8.showhint:=true;

end;

procedure TForm1.Button9Click(Sender: TObject);
begin
form2.show;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
```

```
form22.hide;
form23.hide;
end;

procedure TForm1.Button10Click(Sender: TObject);
begin
form3.show;
form2.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button12Click(Sender: TObject);
begin
form5.show;
form2.hide;
form3.hide;
form4.hide;
form6.hide;
form7.hide;
```

```
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button15Click(Sender: TObject);
begin
form7.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
```

```
form22.hide;
form23.hide;
end;
procedure TForm1.Button16Click(Sender: TObject);
begin
form17.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button18Click(Sender: TObject);
begin
form10.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
```

```
form8.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button23Click(Sender: TObject);
begin
form14.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
```

```
form23.hide;
end;

procedure TForm1.Button24Click(Sender: TObject);
begin
form15.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button26Click(Sender: TObject);
begin
form19.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
```

```
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button29Click(Sender: TObject);
begin
form23.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
```

```
form22.hide;
end;

procedure TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;

procedure TForm1.Button7MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='About Us';
end;

procedure TForm1.Button27MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Rental Turns';
end;

procedure TForm1.Button9MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='All Customers';
end;

procedure TForm1.Button10MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Customer Operations';
end;

procedure TForm1.Button11MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
```

```
statusbar1.Panels.Items[1].Text:='New Rental';
end;

procedure TForm1.Button12MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Customer Search';
end;

procedure TForm1.Button15MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='List of All Videos';
end;

procedure TForm1.Button16MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Video Operations';
end;


procedure TForm1.Button18MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Videos Search';
end;

procedure TForm1.Button23MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='List of All Suppliers';
end;

procedure TForm1.Button24MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
```

```
begin
statusbar1.Panels.Items[1].Text:='Supplier Operations';
end;


procedure TForm1.Button26MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Supplier Search';
end;

procedure TForm1.Button30Click(Sender: TObject);
begin
form4.show;
form2.hide;
form3.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button11Click(Sender: TObject);
```

```
begin
form21.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button13Click(Sender: TObject);
begin
form22.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
```

```
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form23.hide;
end;

procedure TForm1.CustomerOperations1Click(Sender: TObject);
begin
form3.show;

form2.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;end;

procedure TForm1.Button14Click(Sender: TObject);
```

```
begin
form6.Show;
end;

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
form20.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form21.hide;
form22.hide;
form23.hide;
groupbox1.Visible:=False;
groupbox2.Visible:=False;
groupbox3.Visible:=False;
groupbox4.Visible:=False;
groupbox5.Visible:=False;
end;

procedure TForm1.SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState;
 X, Y: Integer);
begin
```

```
statusbar1.Panels.Items[1].Text:='Sales';
end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
groupbox1.Visible:=False;
groupbox2.Visible:=True;
groupbox3.Visible:=False;
groupbox4.Visible:=False;
groupbox5.Visible:=False;
end;

procedure TForm1.SpeedButton2MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Videos Menu';
```

```
end;
procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
groupbox1.Visible:=False;
groupbox2.Visible:=False;
groupbox3.Visible:=False;
groupbox4.Visible:=False;
groupbox5.Visible:=True;
end;

procedure TForm1.SpeedButton3MouseMove(Sender: TObject; Shift: TShiftState;
 X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Rentals';
end;
```

```
procedure TForm1.SpeedButton4Click(Sender: TObject);
begin
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
groupbox1.Visible:=True;
groupbox2.Visible:=False;
groupbox3.Visible:=False;
groupbox4.Visible:=False;
groupbox5.Visible:=False;
end;

procedure TForm1.SpeedButton4MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Customer Menu';
end;

procedure TForm1.Button31Click(Sender: TObject);
```

```
begin
form6.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.SpeedButton5Click(Sender: TObject);
begin
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
```

```
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
groupbox1.Visible:=False;
groupbox2.Visible:=False;
groupbox3.Visible:=False;
groupbox4.Visible:=True;
groupbox5.Visible:=False;
end;

procedure TForm1.SpeedButton5MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Queries About Videos';
end;
procedure TForm1.SpeedButton6Click(Sender: TObject);
begin
groupbox1.Visible:=False;
groupbox2.Visible:=False;
groupbox3.Visible:=True;
groupbox4.Visible:=False;
groupbox5.Visible:=False;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
```

```
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.SpeedButton6MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Suppliers Menu';
end;

procedure TForm1.SpeedButton7Click(Sender: TObject);
var
c:word;
begin
c:=application.MessageBox('Do You Really Want to Exit from The
program?','Exit',mb_yesno+mb_iconquestion);
case c of
idyes :halt;
end;
end;

procedure TForm1.SpeedButton7MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quit The Program';
end;
```

```
procedure TForm1.Timer2Timer(Sender: TObject);
begin
Caption:=copy(caption,2,length(caption)-1)+caption[1];
end;

procedure TForm1.Button32Click(Sender: TObject);
begin
form11.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Button13MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Rental Turn';
end;
procedure TForm1.Button29MouseMove(Sender: TObject; Shift: TShiftState; X,
```

```
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Search For Bringing Date';
end;
procedure TForm1.Button30MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Bringing date passed/today?';
end;

procedure TForm1.Button31MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Most Rented 3 Video';
end;
procedure TForm1.Button32MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Most Sold 3 Video';
end;

procedure TForm1.SpeedButton8Click(Sender: TObject);
begin
form13.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form14.hide;
form15.hide;
```

```
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
groupbox1.Visible:=False;
groupbox2.Visible:=False;
groupbox3.Visible:=False;
groupbox4.Visible:=False;
groupbox5.Visible:=False;
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
form24.Show;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
halt;
end;

procedure TForm1.SpeedButton8MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Security Process';
end;

procedure TForm1.AboutUS1Click(Sender: TObject);
begin
form25.show;
end;

procedure TForm1.AllCustomers1Click(Sender: TObject);
```

```
begin
form2.show;

form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;
procedure TForm1.SearchingCustomers1Click(Sender: TObject);
begin
form5.show;
form2.hide;
form3.hide;
form4.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
```

```
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.Sales1Click(Sender: TObject);
begin
form20.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form21.hide;
form22.hide;
form23.hide;
groupbox1.Visible:=False;
groupbox2.Visible:=False;
groupbox3.Visible:=False;
```

```
groupbox4.Visible:=False;
groupbox5.Visible:=False;
end;

procedure TForm1.Rentals1Click(Sender: TObject);
begin
form21.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.RentalTurn1Click(Sender: TObject);
begin
form22.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
```

```
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form23.hide;
end;

procedure TForm1.VideoOperations1Click(Sender: TObject);
begin
form7.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
```

```
form22.hide;
form23.hide;
end;

procedure TForm1.SearchingVideos1Click(Sender: TObject);
begin
form16.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.SearchingVideos2Click(Sender: TObject);
begin
form10.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
```

```
form7.hide;
form8.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.ListOfSuppliers1Click(Sender: TObject);
begin
form14.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
```

```
form22.hide;
form23.hide;
end;

procedure TForm1.SupplierOperations1Click(Sender: TObject);
begin
form15.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.SearchingSupplier1Click(Sender: TObject);
begin
form19.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
```

```
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;
procedure TForm1.Between2Dates1Click(Sender: TObject);
begin
form23.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
```

```
form22.hide;
end;

procedure TForm1.BringingDatePassedAndToday1Click(Sender: TObject);
begin
form4.show;
form2.hide;
form3.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.N3MostRented1Click(Sender: TObject);
begin
form6.Show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form7.hide;
form8.hide;
```

```
form10.hide;
form11.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
form23.hide;
end;

procedure TForm1.N3MostSold1Click(Sender: TObject);
begin
form11.show;
form2.hide;
form3.hide;
form4.hide;
form5.hide;
form6.hide;
form7.hide;
form8.hide;
form10.hide;
form12.hide;
form13.hide;
form14.hide;
form15.hide;
form16.hide;
form17.hide;
form19.hide;
form20.hide;
form21.hide;
form22.hide;
```

```
form23.hide;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
form27.show;
end;

procedure TForm1.VideoTrailer1Click(Sender: TObject);
begin
Form27.Show;
end;

procedure TForm1.DeletingVideo1Click(Sender: TObject);
begin
form8.Show;
end;
end.
```

5.2 The Codes of "List All Costumer" Form

```
unit Unit2;

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

type
  TForm2 = class(TForm)
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    Label1: TLabel;
```

```
  ADODataSet1: TADODataSet;
  ADODataSet1CustomerId: TAutoIncField;
  ADODataSet1CustomerName: TWideStringField;
  ADODataSet1CustomerSurname: TWideStringField;
  ADODataSet1Telephone: TWideStringField;
  ADODataSet1Address: TWideStringField;
  ADODataSet1City: TWideStringField;
  ADODataSet1Country: TWideStringField;
  DataSource1: TDataSource;
  SpeedButton1: TSpeedButton;
  Image1: TImage;
  procedure FormCreate(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure SpeedButton1Click(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
 Form2: TForm2;

implementation

{$R *.dfm}

procedure TForm2.FormCreate(Sender: TObject);
begin
form2.Caption :='All customers';
end;

procedure TForm2.Button1Click(Sender: TObject);
begin
form2.Hide;
end;
```

```
procedure TForm2.SpeedButton1Click(Sender: TObject);
begin
form2.Hide;
end;
end.
```

5.3 The Codes of "Costumer Operations" Form

```
unit Unit3;

interface

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

type
 TForm3 = class(TForm)
  ADOTable1: TADOTable;
  DataSource1: TDataSource;
  ADOTable1CustomerId: TAutoIncField;
  ADOTable1CustomerName: TWideStringField;
  ADOTable1CustomerSurname: TWideStringField;
  ADOTable1Telephone: TWideStringField;
  ADOTable1Address: TWideStringField;
  ADOTable1City: TWideStringField;
  ADOTable1Country: TWideStringField;
  Label2: TLabel;
  DBEdit2: TDBEdit;
  Label3: TLabel;
  DBEdit3: TDBEdit;
  Label4: TLabel;
  DBEdit4: TDBEdit;
```

```
Label5: TLabel;
DBEdit5: TDBEdit;
Label6: TLabel;
DBEdit6: TDBEdit;
Label7: TLabel;
DBEdit7: TDBEdit;
DBGrid1: TDBGrid;
DBNavigator1: TDBNavigator;
StatusBar1: TStatusBar;
Label1: TLabel;
Label8: TLabel;
SpeedButton1: TSpeedButton;
Image1: TImage;
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit7MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
 private
{ Private declarations }
```

```
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

{$R *.dfm}

procedure TForm3.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=100;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=170;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=250;
statusbar1.panels.items[0].Text:='Video Store v3.1';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);

form3.caption:='Customer Operations';
end;

procedure TForm3.Button1Click(Sender: TObject);
begin
form3.Hide;
end;

procedure TForm3.DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Name';
end;
procedure TForm3.DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
```

```
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Surname';
end;
procedure TForm3.DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Phone Number';
end;
procedure TForm3.DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter City';
end;
procedure TForm3.DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Address';
end;
procedure TForm3.DBEdit7MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Country';
end;
procedure TForm3.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quit';
end;
procedure TForm3.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;
procedure TForm3.SpeedButton1Click(Sender: TObject);
begin
```

```
form3.Hide;
end;
procedure TForm3.SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quit';
end;
end.
```

5.4 The Codes of "Who is bringing?" Form

```
unit Unit4;

interface

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

type
  TForm4 = class(TForm)
    ADODataSet1: TADODataSet;
    DataSource1: TDataSource;
    ADODataSet1RentalId: TAutoIncField;
    ADODataSet1VideoName: TWideStringField;
    ADODataSet1CustomerName: TWideStringField;
    ADODataSet1CustomerSurname: TWideStringField;
    ADODataSet1Telephone: TWideStringField;
    ADODataSet1BringingDate: TDateTimeField;
    ADODataSet1Turned: TBooleanField;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
```

```
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    DBText1: TDBText;
    DBText2: TDBText;
    DBText3: TDBText;
    DBText4: TDBText;
    DBText5: TDBText;
    DBText6: TDBText;
    DBText7: TDBText;
    DBNavigator1: TDBNavigator;
    Label7: TLabel;
    Label8: TLabel;
    SpeedButton1: TSpeedButton;
    Image1: TImage;

    procedure SpeedButton1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);

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

var
  Form4: TForm4;

implementation

{$R *.dfm}

procedure TForm4.SpeedButton1Click(Sender: TObject);
begin
 form4.Hide;
end;
```

```
procedure TForm4.FormCreate(Sender: TObject);
begin
end;
end.
```

5.5 The Codes of " Searching Costumer" Form

```
unit Unit5;

interface

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

type
  TForm5 = class(TForm)
    Edit1: TEdit;
    Button1: TButton;
    DBGrid1: TDBGrid;
    Button2: TButton;
    Label1: TLabel;
    StatusBar1: TStatusBar;
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    CheckBox3: TCheckBox;
    Label2: TLabel;
    CheckBox4: TCheckBox;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
```

```
  procedure FormCreate(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure CheckBox1Click(Sender: TObject);
  procedure CheckBox2Click(Sender: TObject);
  procedure CheckBox3Click(Sender: TObject);
  procedure CheckBox4Click(Sender: TObject);
  procedure CheckBox1MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure CheckBox3MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure CheckBox2MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure CheckBox4MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure SpeedButton1Click(Sender: TObject);
  procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
 Form5: TForm5;

implementation

{$R *.dfm}
procedure TForm5.Button1Click(Sender: TObject);
```

```
begin
if (checkbox1.Checked=false) and (checkbox2.Checked=false) and
(checkbox3.Checked=false)and (checkbox4.Checked=false)then
begin
showmessage('You have to Select one of Them');
end;

if checkbox1.checked then
begin
dbgrid1.visible:= true;
label1.Visible :=true;
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from customers');
adoquery1.SQL.Add('where customername=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
adoquery1.Open;

end;
if checkbox2.Checked then
begin
label1.Visible :=true;
dbgrid1.visible:= true;
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from customers');
adoquery1.SQL.Add('where customersurname=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
adoquery1.Open;
end;
if checkbox3.checked then
begin
label1.Visible :=true;
dbgrid1.visible:= true;
adoquery1.Close;
adoquery1.SQL.Clear ;
```

```
adoquery1.SQL.Add('select * from customers');
adoquery1.SQL.Add('where city=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
adoquery1.Open;
end;
if checkbox4.checked then
begin
label1.Visible :=true;
dbgrid1.visible:= true;
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from customers');
adoquery1.SQL.Add('where country=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
adoquery1.Open;
end;

if edit1.text='' then
begin
showmessage('You Have to Enter Customers info!!');
dbgrid1.Visible :=false;
label1.Visible :=false;
end;
end;

procedure TForm5.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=100;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=170;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=250;
statusbar1.panels.items[0].Text:='Video Store v3.1';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);
edit1.Text:='';
```

```
label1.Caption:='Founded Results';
button1.Caption:='Search';
button2.Caption:='Clear';

form5.Caption:='Searching Customers ';
edit1.Text:='';
edit1.Visible :=False;
dbgrid1.Visible :=False;
label1.Visible :=false;
end;

procedure TForm5.Button2Click(Sender: TObject);
begin
checkbox1.Checked:=False;
checkbox2.Checked:=False;
checkbox3.Checked:=False;
checkbox4.Checked:=False;
checkbox1.Enabled :=True;
checkbox2.Enabled :=True;
checkbox3.Enabled :=True;
checkbox4.Enabled :=True;
label1.Visible :=False;
dbgrid1.Visible :=False;
edit1.Visible :=False;
edit1.text:='';
end;

procedure TForm5.CheckBox1Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox2.Enabled :=FAlse;
checkbox3.Enabled :=False;
checkbox4.Enabled :=False;
end;
```

```
procedure TForm5.CheckBox2Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox1.Enabled :=False;
checkbox3.Enabled :=False;
checkbox4.Enabled :=False;
end;

procedure TForm5.CheckBox3Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox1.Enabled :=False;
checkbox2.Enabled :=False;
checkbox4.Enabled :=False;
end;

procedure TForm5.CheckBox4Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox1.Enabled :=False;
checkbox2.Enabled :=False;
checkbox3.Enabled :=False;
end;

procedure TForm5.CheckBox1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Searching Customer by Name';
end;
procedure TForm5.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
```

```
end;
procedure TForm5.CheckBox3MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Searching Customer by City';
end;
procedure TForm5.CheckBox2MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Searching Customer by Surname';
end;
procedure TForm5.CheckBox4MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Searching Customer by Country';
end;
procedure TForm5.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Starts Searhing';
end;
procedure TForm5.Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Clears Form';
end;
procedure TForm5.SpeedButton1Click(Sender: TObject);
begin
form5.Hide;
checkbox1.Checked:=False;
checkbox2.Checked:=False;
checkbox1.Enabled :=True;
checkbox2.Enabled :=True;
checkbox3.Checked:=False;
checkbox3.Enabled :=True;
checkbox4.Checked:=False;
```

```
 checkbox4.Enabled :=True;
 label1.Visible :=False;
 dbgrid1.Visible :=False;
 edit1.Visible :=False;
 edit1.text:='';
 end;

 procedure TForm5.SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
 begin
 statusbar1.Panels.Items[1].Text:='Quit';
 end;
 end.
```

5.6 The Codes of "Most Rented" Form

```
unit Unit6;
interface

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

type
 TForm6 = class(TForm)
  ADODataSet1: TADODataSet;
  ADODataSet1Deyim1: TIntegerField;
  ADODataSet1VideoName: TWideStringField;
  DataSource1: TDataSource;
  DBCtrlGrid1: TDBCtrlGrid;
  Label2: TLabel;
  DBText1: TDBText;
```

```
  DBText2: TDBText;
  Label4: TLabel;
  Label1: TLabel;
  Label3: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  SpeedButton1: TSpeedButton;
  Image1: TImage;
  procedure SpeedButton1Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
 Form6: TForm6;

implementation

{$R *.dfm}

procedure TForm6.SpeedButton1Click(Sender: TObject);
begin
form6.Hide;
end;

procedure TForm6.FormCreate(Sender: TObject);
begin
 form6.Caption :='Most Rented 3 Videos';
 label6.Caption :='Most Rented 3 Videos'';
end;
end.
```

5.7 The Codes of "List of all Videos" Form

```
unit Unit7;

interface

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

type
  TForm7 = class(TForm)
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    ADOTable1VideoId: TAutoIncField;
    ADOTable1SupplierId: TIntegerField;
    ADOTable1VideoName: TWideStringField;
    ADOTable1VideoDesc: TMemoField;
    ADOTable1VideoDirector: TWideStringField;
    ADOTable1VideoType: TWideStringField;
    ADOTable1VideoYear: TIntegerField;
    ADOTable1UnitPrice: TBCDField;
    ADOTable1VideoInStock: TIntegerField;
    ADOTable1VideoPicture: TBlobField;
    Label1: TLabel;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
```

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

var
  Form7: TForm7;

implementation

{$R *.dfm}

procedure TForm7.Button1Click(Sender: TObject);
begin
form7.hide;
end;

procedure TForm7.FormCreate(Sender: TObject);
begin
form7.Caption :='List of Videos';
end;

procedure TForm7.SpeedButton1Click(Sender: TObject);
begin
form7.Hide;
end;

end.
```

5.8 The Codes of "Video Operations" Form

```
unit Unit8;

interface

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

type
  TForm8 = class(TForm)
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    ADOTable1VideoId: TAutoIncField;
    ADOTable1SupplierId: TIntegerField;
    ADOTable1VideoName: TWideStringField;
    ADOTable1VideoDesc: TMemoField;
    ADOTable1VideoType: TWideStringField;
    ADOTable1VideoYear: TIntegerField;
    ADOTable1UnitPrice: TBCDField;
    ADOTable1VideoInStock: TIntegerField;
    ADOTable1VideoPicture: TBlobField;
    Label1: TLabel;
    DBEdit1: TDBEdit;
    Label2: TLabel;
    DBEdit2: TDBEdit;
    Label3: TLabel;
    DBEdit3: TDBEdit;
    Label4: TLabel;
    DBMemo1: TDBMemo;
    Label5: TLabel;
    DBEdit4: TDBEdit;
    Label6: TLabel;
```

```
DBEdit5: TDBEdit;
Label7: TLabel;
DBEdit6: TDBEdit;
Label8: TLabel;
DBEdit7: TDBEdit;
Label9: TLabel;
DBImage1: TDBImage;
OpenPictureDialog1: TOpenPictureDialog;
DBGrid1: TDBGrid;
DBNavigator1: TDBNavigator;
Edit1: TEdit;
Button2: TButton;
Label10: TLabel;
ADOTable2: TADOTable;
DataSource2: TDataSource;
DBGrid2: TDBGrid;
Bevel1: TBevel;
DBNavigator2: TDBNavigator;
ADOTable1VideoDirector: TWideStringField;
Label11: TLabel;
DBEdit8: TDBEdit;
Label12: TLabel;
Label13: TLabel;
StatusBar1: TStatusBar;
SpeedButton1: TSpeedButton;
Image1: TImage;
procedure DBImage1DblClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
```

```
procedure DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit7MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBImage1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBMemo1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit8MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBGrid1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Edit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton1Click(Sender: TObject);

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

var
 Form8: TForm8;

implementation
```

```
uses Unit7;

{$R *.dfm}

procedure TForm8.DBImage1DblClick(Sender: TObject);
begin
 OpenPictureDialog1.Title:='Pick up an Picture';

 if (OpenPictureDialog1.Execute)then
 begin
 adotable1.Edit ;
 dbimage1.Picture.LoadFromFile(OpenPictureDialog1.FileName);
 adotable1.Post;
 end;
end;

procedure TForm8.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=200;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=300;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=200;
statusbar1.panels.items[0].Text:='Gürsoy Video Center';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);

speedbutton1.Caption:='' ;
button2.Caption:='Search';
Label10.Caption:='Supplier Name';
edit1.Text :='';
dbgrid2.Visible :=False;
form8.Caption :='Video Operations';
end;

procedure TForm8.Button1Click(Sender: TObject);
```

```
begin
form8.Hide;
form7.hide;
end;
procedure TForm8.Button2Click(Sender: TObject);
var
src:boolean;
begin
if edit1.text='' then
begin
showmessage('Please enter a record');
edit1.setfocus
end
else
 src:=ADOtable2.Locate('SupplierName',edit1.Text,[lopartialkey]);
 dbgrid2.canvas.font.Color:=clred;
 dbgrid2.Visible :=true;

if not src=true then
 begin
 showmessage('Record Not Found');
 dbgrid2.Visible :=false;
 edit1.setfocus;
 dbgrid2.Visible :=false;
end;
end;
procedure TForm8.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;
procedure TForm8.DBEdit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Supplier Id';
end;
```

```
procedure TForm8.DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Video Name';
end;
procedure TForm8.DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Video Type';
end;
procedure TForm8.DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
 statusbar1.Panels.Items[1].Text:='Enter Video Year';
 end;
 procedure TForm8.DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
   Y: Integer);
 begin
 statusbar1.Panels.Items[1].Text:='Enter Video Price';
 end;
 procedure TForm8.DBEdit7MouseMove(Sender: TObject; Shift: TShiftState; X,
   Y: Integer);
 begin
 statusbar1.Panels.Items[1].Text:='Enter Amount of Video';
 end;
 procedure TForm8.DBImage1MouseMove(Sender: TObject; Shift: TShiftState; X,
   Y: Integer);
 begin
 statusbar1.Panels.Items[1].Text:='Add a Picture to Video';
 end;
 procedure TForm8.DBMemo1MouseMove(Sender: TObject; Shift: TShiftState; X,
   Y: Integer);
 begin
 statusbar1.Panels.Items[1].Text:='Enter Description For Video';
 end;
```

```
procedure TForm8.DBEdit8MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Video Director';
end;
procedure TForm8.DBGrid1MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='List Of Videos In Stock';
end;
procedure TForm8.Edit1MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Supplier Name For Search';
end;
procedure TForm8.Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Starts Search';
end;
procedure TForm8.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quits the Page';
end;
procedure TForm8.SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState;
 X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quits the Page';
end;
procedure TForm8.SpeedButton1Click(Sender: TObject);
begin
form8.Hide;
end;

end.
```

5.9 The Codes of "Searching Videos" Form

```
unit Unit10;

interface

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

type
  TForm10 = class(TForm)
    DBGrid1: TDBGrid;
    Edit1: TEdit;
    Button1: TButton;
    Label1: TLabel;
    Button2: TButton;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    CheckBox3: TCheckBox;
    CheckBox4: TCheckBox;
    Label2: TLabel;
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    StatusBar1: TStatusBar;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure CheckBox1Click(Sender: TObject);
    procedure CheckBox2Click(Sender: TObject);
    procedure CheckBox3Click(Sender: TObject);
    procedure CheckBox4Click(Sender: TObject);
```

```
procedure CheckBox1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure CheckBox1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure CheckBox2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure CheckBox3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure CheckBox4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);

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

var
 Form10: TForm10;

implementation

{$R *.dfm}

procedure TForm10.Button1Click(Sender: TObject);
begin
```

```
if (checkbox1.Checked=false) and (checkbox2.Checked=false) and
(checkbox3.Checked=false)and (checkbox4.Checked=false)then
begin
showmessage('You have to Select one of Them');
end;

if checkbox1.checked then
begin
dbgrid1.visible:= true;
label2.Visible :=true;
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from videos');
adoquery1.SQL.Add('where videoname=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
adoquery1.Open;

end;
if checkbox2.Checked then
begin
label2.Visible :=true;
dbgrid1.visible:= true;
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from videos');
adoquery1.SQL.Add('where VideoDirector=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
adoquery1.Open;
end;
if checkbox3.checked then
begin
label2.Visible :=true;
dbgrid1.visible:= true;
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from videos');
```

```
adoquery1.SQL.Add('where VideoType=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
adoquery1.Open;
end;
if checkbox4.checked then
begin
label2.Visible :=true;
dbgrid1.visible:= true;
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from videos');
adoquery1.SQL.Add('where Videoyear=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=strtoint(edit1.text);
adoquery1.Open;
end;
end;
procedure TForm10.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=200;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=200;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=200;
statusbar1.panels.items[0].Text:='Gursoy Video Center';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);

button1.caption:='Search';
button2.caption:='Clear';
speedbutton1.caption:='';
Label1.caption:= 'Searching Videos';
label2.Visible :=false;

dbgrid1.Visible :=False;

edit1.Text:='';
```

```
form10.Caption :='Searching Videos';
edit1.Text:='';
edit1.Visible :=False;
end;

procedure TForm10.Button2Click(Sender: TObject);
begin
checkbox1.Checked:=False;
checkbox2.Checked:=False;
checkbox3.Checked:=False;
checkbox4.Checked:=False;

checkbox1.Enabled :=True;
checkbox2.Enabled :=True;
checkbox3.Enabled :=True;
checkbox4.Enabled :=True;

label2.Visible :=False;
dbgrid1.Visible :=False;
edit1.Visible :=False;
edit1.text:='';
end;

procedure TForm10.CheckBox1Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox2.Enabled :=FAlse;
checkbox3.Enabled :=False;
checkbox4.Enabled :=False;
end;

procedure TForm10.CheckBox2Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
```

```
checkbox1.Enabled :=False;
checkbox3.Enabled :=False;
checkbox4.Enabled :=False;
end;

procedure TForm10.CheckBox3Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox1.Enabled :=False;
checkbox2.Enabled :=False;
checkbox4.Enabled :=False;
end;

procedure TForm10.CheckBox4Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox1.Enabled :=False;
checkbox2.Enabled :=False;
checkbox3.Enabled :=False;
end;

procedure TForm10.CheckBox1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Search Videos By Name';
end;
procedure TForm10.CheckBox1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Search Videos By Name';
end;
procedure TForm10.CheckBox2MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
```

```
statusbar1.Panels.Items[1].Text:='Search Videos By Director';
end;
procedure TForm10.CheckBox3MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Search Videos By Type';
end;
procedure TForm10.CheckBox4MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Search Videos By Year';
end;
procedure TForm10.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Starts Video Search ';
end;
procedure TForm10.Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Clears Form';
end;
procedure TForm10.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;

procedure TForm10.SpeedButton1Click(Sender: TObject);
begin
form10.Hide;

checkbox1.Checked:=False;
checkbox2.Checked:=False;
checkbox3.Checked:=False;
checkbox4.Checked:=False;
```

```
checkbox1.Enabled :=True;
checkbox2.Enabled :=True;
checkbox3.Enabled :=True;
checkbox4.Enabled :=True;
label2.Visible :=False;
dbgrid1.Visible :=False;
edit1.Visible :=False;
edit1.text:='';
end;

procedure TForm10.SpeedButton1MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quits Video Searching';
end;
end.
```

5.10 The Codes of "Most Sold" Form

```
unit Unit11;

interface

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

type
  TForm11 = class(TForm)
    ADODataSet1: TADODataSet;
    DataSource1: TDataSource;
    DBCtrlGrid1: TDBCtrlGrid;
    ADODataSet1Deyim1: TIntegerField;
```

```
  ADODataSet1VideoName: TWideStringField;
  Label1: TLabel;
  DBText1: TDBText;
  DBText2: TDBText;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  SpeedButton1: TSpeedButton;
  Image1: TImage;
  procedure SpeedButton1Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
 private
   { Private declarations }
 public
   { Public declarations }
 end;

var
 Form11: TForm11;

implementation

{$R *.dfm}

procedure TForm11.SpeedButton1Click(Sender: TObject);
begin
form11.Hide;
end;
procedure TForm11.FormCreate(Sender: TObject);
begin
form11.Caption:='Most Sold  3 Video';
end;

end.
```

5.11 The Codes of "Password Process" Form

```
unit Unit12;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, DBCtrls, ExtCtrls, DB, Mask, ADODB, Grids, DBGrids,
  Buttons, jpeg;

type
  TForm12 = class(TForm)
    ADOTable1: TADOTable;
    ADOTable1UserName: TWideStringField;
    ADOTable1Pass: TWideStringField;
    Label1: TLabel;
    DBEdit1: TDBEdit;
    DataSource1: TDataSource;
    Label2: TLabel;
    DBEdit2: TDBEdit;
    DBNavigator1: TDBNavigator;
    DBGrid1: TDBGrid;
    Label3: TLabel;
    SpeedButton1: TSpeedButton;
    Image1: TImage;
    procedure SpeedButton1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form12: TForm12;
implementation
```

```
{$R *.dfm}

procedure TForm12.SpeedButton1Click(Sender: TObject);
begin
form12.Hide;
end;

procedure TForm12.FormCreate(Sender: TObject);
begin
form12.Caption:='Password Process';
end;
end.
```

5.12 The Codes of "Password Required" Form

```
unit Unit13;

interface

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

type
  TForm13 = class(TForm)
    Edit1: TEdit;
    Button1: TButton;
    Label1: TLabel;
    Button2: TButton;
    Image1: TImage;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
```

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

var
 Form13: TForm13;
 tur: integer;
implementation

uses Unit12;

{$R *.dfm}

procedure TForm13.FormCreate(Sender: TObject);
begin
form13.Caption:='Admin Panel';
button1.Caption:='Confirm';
button2.Caption:='Cancel';
end;
procedure TForm13.Button1Click(Sender: TObject);
 begin
if edit1.Text ='admin' then
begin
edit1.Text:='';
Application.MessageBox('Acess Granted', '  Congratulation ');
form12.Show;
form13.Hide;
end
else
begin
Application.MessageBox('Wrong Password !'#13'Please Enter Again!','Try Again' );
tur:=tur+1;
edit1.Text:='';
edit1.SetFocus;
```

```
    if tur=3 then
    begin
    edit1.Text:='';
    Application.MessageBox('Sorry You cant Enter this time!','Ooppss!!!');
     form13.Hide;
    end;
end;
end;

procedure TForm13.Button2Click(Sender: TObject);
begin
form13.Hide;
end;
end.
```

5.13 The Codes of "List Of suppliers" Form

```
unit Unit14;

interface

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

type
  TForm14 = class(TForm)
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    DBNavigator1: TDBNavigator;
    Label1: TLabel;
```

```
  SpeedButton1: TSpeedButton;
  Image1: TImage;
  procedure FormCreate(Sender: TObject);
  procedure SpeedButton1Click(Sender: TObject);
 private
   { Private declarations }
 public
   { Public declarations }
 end;

var
 Form14: TForm14;

implementation

{$R *.dfm}

procedure TForm14.FormCreate(Sender: TObject);
begin
speedbutton1.Caption:='';
form14.Caption:='List of All Suppliers';
end;
procedure TForm14.SpeedButton1Click(Sender: TObject);
begin
form14.Hide;
end;
end.
```

5.14 The Codes of "Supplier Operations" Form

```
unit Unit15;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
```

```
Dialogs, StdCtrls, ExtCtrls, DBCtrls, Grids, DBGrids, Mask, DB, ADODB,
ComCtrls, Buttons, jpeg;

type
 TForm15 = class(TForm)
  ADOTable1: TADOTable;
  DataSource1: TDataSource;
  ADOTable1SupplierId: TAutoIncField;
  ADOTable1SupplierName: TWideStringField;
  ADOTable1Telephone: TWideStringField;
  ADOTable1Address: TWideStringField;
  ADOTable1City: TWideStringField;
  ADOTable1Country: TWideStringField;
  Label1: TLabel;
  DBEdit1: TDBEdit;
  Label2: TLabel;
  DBEdit2: TDBEdit;
  Label3: TLabel;
  DBEdit3: TDBEdit;
  Label4: TLabel;
  DBEdit4: TDBEdit;
  Label5: TLabel;
  DBEdit5: TDBEdit;
  Label6: TLabel;
  DBEdit6: TDBEdit;
  DBGrid1: TDBGrid;
  DBNavigator1: TDBNavigator;
  Label7: TLabel;
  StatusBar1: TStatusBar;
  SpeedButton1: TSpeedButton;
  Image1: TImage;
  procedure Button1Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
   Y: Integer);
  procedure DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
```

```
 Y: Integer);
procedure DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
procedure DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
procedure DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
procedure DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
private
 { Private declarations }
public
 { Public declarations }
end;

var
 Form15: TForm15;

implementation

{$R *.dfm}

procedure TForm15.Button1Click(Sender: TObject);
begin
form15.Hide;
end;

procedure TForm15.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=200;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=250;
```

```
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=250;
statusbar1.panels.items[0].Text:='Gürsoy Video Center';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);
form15.caption:='Supplier Operations';
end;

procedure TForm15.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;
procedure TForm15.DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Supplier Name';
end;
procedure TForm15.DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Supplier Phone';
end;
procedure TForm15.DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Supplier Address';
end;
procedure TForm15.DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Supplier City';
end;
procedure TForm15.DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Supplier Country';
```

```
end;
procedure TForm15.SpeedButton1Click(Sender: TObject);
begin
form15.Hide;
end;
procedure TForm15.SpeedButton1MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quit';
end;
end.
```

5.15 The Codes of "Purchase Exiting Video" Form

```
unit Unit16;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Mask, DBCtrls, DB, Grids, DBGrids, ADODB, ExtCtrls,
  ComCtrls, Buttons, jpeg;
type
  TForm16 = class(TForm)
    ADOTable1: TADOTable;
    ADOTable1VideoId: TAutoIncField;
    ADOTable1SupplierId: TIntegerField;
    ADOTable1VideoName: TWideStringField;
    ADOTable1VideoDesc: TMemoField;
    ADOTable1VideoDirector: TWideStringField;
    ADOTable1VideoType: TWideStringField;
    ADOTable1VideoYear: TIntegerField;
    ADOTable1UnitPrice: TBCDField;
```

```
ADOTable1VideoInStock: TIntegerField;
ADOTable1VideoPicture: TBlobField;
DataSource1: TDataSource;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label6: TLabel;
Label9: TLabel;
Label10: TLabel;
DBText1: TDBText;
DBText2: TDBText;
DBText3: TDBText;
DBText5: TDBText;
DBGrid1: TDBGrid;
Label5: TLabel;
DBText4: TDBText;
Label7: TLabel;
DBText6: TDBText;
DBText7: TDBText;
DBText8: TDBText;
Label8: TLabel;
Label11: TLabel;
Button1: TButton;
Edit1: TEdit;
Label12: TLabel;
Edit2: TEdit;
Button2: TButton;
StatusBar1: TStatusBar;
SpeedButton1: TSpeedButton;
Image1: TImage;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Edit2MouseMove(Sender: TObject; Shift: TShiftState; X,
```

```
  Y: Integer);
procedure Edit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
private
  { Private declarations }
public
  { Public declarations }
end;

var
 Form16: TForm16;

implementation

{$R *.dfm}

procedure TForm16.Button1Click(Sender: TObject);
var
x,y,z: integer;
begin
if edit1.Text='' then
showmessage ('Enter Amount')
else
begin
x:= strtoint(edit1.Text);
y:= strtoint(dbtext8.caption);
z:=x+y;
```

```
label12.Caption :=inttostr(z);
adotable1.Edit;
adotable1.FieldValues['videoinstock']:=label12.caption;
adotable1.Refresh ;
adotable1.next;
end;
end;

procedure TForm16.Button2Click(Sender: TObject);
begin
if edit2.Text='' then
showmessage ('Enter New Price')
else
begin
adotable1.Edit;
adotable1.FieldValues['unitprice']:=edit2.text;
adotable1.Refresh ;
adotable1.next;
end;
end;

procedure TForm16.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=200;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=350;
statusbar1.panels.items[1].Alignment:=tacenter;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=200;
statusbar1.panels.items[0].Text:='Gürsoy Video Center';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);

speedbutton1.caption:='';
form16.Caption:='Purchasing Existing Video';
end;
```

```
procedure TForm16.Button3Click(Sender: TObject);
begin
form16.Hide;
end;
procedure TForm16.Edit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter New Price';
end;
procedure TForm16.Edit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter New Amount';
end;
procedure TForm16.Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
 statusbar1.Panels.Items[1].Text:='Apply Changes on New Price';
end;
procedure TForm16.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Apply Changes on New Amount';
end;
procedure TForm16.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;
procedure TForm16.SpeedButton1Click(Sender: TObject);
begin
form16.Hide;
end;
procedure TForm16.SpeedButton1MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
```

```
statusbar1.Panels.Items[1].Text:='Quit';
end;
end.
```

5.16 The Codes of "Make your choose" Form

```
unit Unit17;
interface

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

type
  TForm17 = class(TForm)
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    Image1: TImage;
    Label1: TLabel;
    procedure RadioButton1Click(Sender: TObject);
    procedure RadioButton2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form17: TForm17;

implementation
```

```
uses Unit8, Unit16;

{$R *.dfm}

procedure TForm17.RadioButton1Click(Sender: TObject);
begin
form8.show;
form17.hide;
end;

procedure TForm17.RadioButton2Click(Sender: TObject);
begin
form16.show;
form17.Hide;
end;

procedure TForm17.FormCreate(Sender: TObject);
begin
end;
end.
```

5.16 The Codes of "Search Supplier" Form

```
unit Unit19;
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, DB, ADODB, Grids, DBGrids, ComCtrls, Buttons, jpeg,
  ExtCtrls;
type
  TForm19 = class(TForm)
    Edit1: TEdit;
    Button1: TButton;
```

```
Button2: TButton;
ADOQuery1: TADOQuery;
DataSource1: TDataSource;
DBGrid1: TDBGrid;
CheckBox1: TCheckBox;
CheckBox2: TCheckBox;
Label1: TLabel;
Label2: TLabel;
StatusBar1: TStatusBar;
SpeedButton1: TSpeedButton;
Image1: TImage;
procedure Button1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
procedure CheckBox2Click(Sender: TObject);
procedure CheckBox1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure CheckBox2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
private
  { Private declarations }
public
  { Public declarations }
end;
var
 Form19: TForm19;
```

```
implementation

{$R *.dfm}

procedure TForm19.Button1Click(Sender: TObject);
begin
if (checkbox1.Checked=false) and (checkbox2.Checked=false) then
begin
showmessage('You have to Select one of Them');
end;

 if checkbox1.checked then
 begin
 dbgrid1.visible:= true;
 label2.Visible :=True;
 adoquery1.Close;
 adoquery1.SQL.Clear ;
 adoquery1.SQL.Add('select * from suppliers');
 adoquery1.SQL.Add('where suppliername=:edit1.text');
 adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
 adoquery1.Open;
 end;
 if checkbox2.Checked then
 begin
 dbgrid1.visible:= true;
 label2.Visible :=True;
 adoquery1.Close;
 adoquery1.SQL.Clear ;
 adoquery1.SQL.Add('select * from suppliers');
 adoquery1.SQL.Add('where city=:edit1.text');
 adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
 adoquery1.Open;
 end;
  if edit1.text='' then
 begin
 showmessage('You Have to Enter Suppliers info!');
```

```
label2.Visible :=False;
dbgrid1.Visible :=false;
///edit1.SetFocus;
end;
end;

procedure TForm19.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=200;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=250;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=200;
statusbar1.panels.items[0].Text:='Gürsoy Video Center';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);

label2.Visible :=false;
edit1.Text:='';
edit1.Visible :=False;
dbgrid1.Visible :=False;
button1.Caption:='Search';
button2.caption:='Clear';
end;

procedure TForm19.Button2Click(Sender: TObject);
begin
checkbox1.Checked:=False;
checkbox2.Checked:=False;
checkbox1.Enabled :=True;
checkbox2.Enabled :=True;
label2.Visible :=false;
dbgrid1.Visible :=False;
edit1.Visible :=False;
edit1.text:='';
end;
```

```
procedure TForm19.CheckBox1Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox2.Enabled :=FAlse;
end;

procedure TForm19.CheckBox2Click(Sender: TObject);
begin
edit1.Visible :=True;
dbgrid1.Visible:=False;
checkbox1.Enabled :=False;
end;

procedure TForm19.CheckBox1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Search Supplier By Name';
end;
procedure TForm19.CheckBox2MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Search Supplier By City';
end;
procedure TForm19.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Start Search';
end;
procedure TForm19.Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Clears Page';
end;
procedure TForm19.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
```

```
begin
statusbar1.Panels.Items[1].Text:='';
end;

procedure TForm19.SpeedButton1Click(Sender: TObject);
begin
checkbox1.Checked:=False;
checkbox2.Checked:=False;
checkbox1.Enabled :=True;
checkbox2.Enabled :=True;
label2.Visible :=false;
dbgrid1.Visible :=False;
edit1.Visible :=False;
edit1.text:='';
form19.Hide;
end;

procedure TForm19.SpeedButton1MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quits Page';
end;
end.
```

5.18 The Codes of "Video sales" Form

```
unit Unit20;
interface

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

```
type
  TForm20 = class(TForm)
    ADOTable1: TADOTable;
    ADOTable1SaleId: TAutoIncField;
    ADOTable1VideoId: TIntegerField;
    ADOTable1CustomerId: TIntegerField;
    ADOTable1SaleDate: TDateTimeField;
    ADOTable1Quantity: TIntegerField;
    ADOTable1Price: TBCDField;
    DataSource1: TDataSource;
    Label2: TLabel;
    DBEdit2: TDBEdit;
    Label3: TLabel;
    DBEdit3: TDBEdit;
    Label4: TLabel;
    Label5: TLabel;
    DBEdit5: TDBEdit;
    Label6: TLabel;
    DBEdit6: TDBEdit;
    DBGrid1: TDBGrid;
    DBGrid2: TDBGrid;
    ADOQuery2: TADOQuery;
    DataSource3: TDataSource;
    Edit2: TEdit;
    Label7: TLabel;
    DBGrid3: TDBGrid;
    DateTimePicker1: TDateTimePicker;
    ADOTable2: TADOTable;
    ADOTable2VideoId: TAutoIncField;
    ADOTable2SupplierId: TIntegerField;
    ADOTable2VideoName: TWideStringField;
    ADOTable2VideoDesc: TMemoField;
    ADOTable2VideoDirector: TWideStringField;
    ADOTable2VideoType: TWideStringField;
    ADOTable2VideoYear: TIntegerField;
    ADOTable2UnitPrice: TBCDField;
```

```
ADOTable2VideoInStock: TIntegerField;
ADOTable2VideoPicture: TBlobField;
Label1: TLabel;
DBEdit1: TDBEdit;
DataSource2: TDataSource;
DBEdit4: TDBEdit;
DBNavigator1: TDBNavigator;
Button1: TButton;
Label9: TLabel;
DBNavigator2: TDBNavigator;
Label8: TLabel;
StatusBar1: TStatusBar;
SpeedButton1: TSpeedButton;
Image1: TImage;
procedure FormCreate(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Edit2Change(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Edit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
```

```
  { Private declarations }
 public
  { Public declarations }
 end;
var
 Form20: TForm20;

implementation

{$R *.dfm}

procedure TForm20.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=200;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=350;
statusbar1.panels.items[1].Alignment:=tacenter;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=200;
statusbar1.panels.items[0].Text:='Gürsoy Video Center';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);

form20.caption:= 'Video Sale';
button1.Caption :='Calculate';
end;

procedure TForm20.Button2Click(Sender: TObject);
begin
form20.Hide;
end;
procedure TForm20.Edit2Change(Sender: TObject);
begin
adoquery2.Close;
```

```
adoquery2.SQL.Clear ;
adoquery2.SQL.Add('select * from customers');
adoquery2.SQL.Add('where customername=:edit2.text');
adoquery2.Parameters.ParamByName('edit2.text').Value:=(edit2.text);
adoquery2.Open;
dbgrid3.visible:= true;
end;
procedure TForm20.Button1Click(Sender: TObject);
 var
uni,x,y,z,a: integer ;
begin
uni:=strtoint(dbedit4.Text);
x:= strtoint(dbedit5.Text);
y:= strtoint(dbedit1.Text);
if  x>uni then
begin
showmessage('Less Unit in Stock');
exit;
dbedit5.setfocus
end
else
z:= x*y;
a:=uni-x;
dbedit6.text:=inttostr(z) ;
label9.Caption:=inttostr(a);
datasource1.DataSet.FieldValues['saledate']:=datetimepicker1.Date;
datasource2.Edit;
datasource2.DataSet.FieldValues['videoinstock']:=label9.Caption;
end;
procedure TForm20.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;
procedure TForm20.DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
```

```
begin
statusbar1.Panels.Items[1].Text:='Enter Video ID';
end;
procedure TForm20.DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Customer ID';
end;
procedure TForm20.DBEdit5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Quantity';
end;
procedure TForm20.DBEdit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Unit Price';
end;
procedure TForm20.DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Total Price';
end;
procedure TForm20.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Calculates The TOTAL Price';
end;
procedure TForm20.Edit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Customer Name';
end;
procedure TForm20.SpeedButton1Click(Sender: TObject);
begin
form20.Hide;
```

```
end;
procedure TForm20.SpeedButton1MouseMove(Sender: TObject;
 Shift: TShiftState; X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Close the Page';
end;
end.
```

5.19 The Codes of "New Rental" Form

```
unit Unit21;
interface

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

type
  TForm21 = class(TForm)
    ADOTable1: TADOTable;
    ADOTable1RentalId: TAutoIncField;
    ADOTable1VideoId: TIntegerField;
    ADOTable1CustomerId: TIntegerField;
    ADOTable1HireDate: TDateTimeField;
    ADOTable1BringingDate: TDateTimeField;
    ADOTable1Quantity: TIntegerField;
    ADOTable1Price: TBCDField;
    Label1: TLabel;
    DBEdit1: TDBEdit;
    DataSource1: TDataSource;
    Label2: TLabel;
    DBEdit2: TDBEdit;
    Label3: TLabel;
```

```
DBEdit3: TDBEdit;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
DBEdit6: TDBEdit;
Label7: TLabel;
DBEdit7: TDBEdit;
DateTimePicker1: TDateTimePicker;
DateTimePicker2: TDateTimePicker;
Button1: TButton;
ADOTable2: TADOTable;
ADOTable2VideoId: TAutoIncField;
ADOTable2SupplierId: TIntegerField;
ADOTable2VideoName: TWideStringField;
ADOTable2VideoDesc: TMemoField;
ADOTable2VideoDirector: TWideStringField;
ADOTable2VideoType: TWideStringField;
ADOTable2VideoYear: TIntegerField;
ADOTable2UnitPrice: TBCDField;
ADOTable2VideoInStock: TIntegerField;
ADOTable2VideoPicture: TBlobField;
Label8: TLabel;
DBEdit4: TDBEdit;
DataSource2: TDataSource;
DBEdit5: TDBEdit;
DBGrid1: TDBGrid;
DBGrid2: TDBGrid;
Edit1: TEdit;
ADOQuery1: TADOQuery;
DataSource3: TDataSource;
Label10: TLabel;
DBGrid3: TDBGrid;
DBNavigator1: TDBNavigator;
Label11: TLabel;
DBNavigator2: TDBNavigator;
StatusBar1: TStatusBar;
```

```
SpeedButton1: TSpeedButton;
Image1: TImage;
procedure Button1Click(Sender: TObject);
procedure Edit1Change(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure DBEdit7MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
private
  { Private declarations }
public
  { Public declarations }
end;
var
 Form21: TForm21;
implementation
{$R *.dfm}
procedure TForm21.Button1Click(Sender: TObject);
var
```

```
uni,x,y,z,a: integer ;
begin
uni:=strtoint(dbedit5.Text);
x:= strtoint(dbedit6.Text);
y:= strtoint(dbedit4.Text);
if  x>uni then
begin
showmessage('Less Unit in Stock');
exit;
dbedit5.setfocus
end
else
z:= x*y div 2;
a:=uni-x;
dbedit7.text:=inttostr(z) ;
label10.Caption:=inttostr(a);
datasource1.DataSet.FieldValues['hiredate']:=datetimepicker1.Date;
datasource1.DataSet.FieldValues['bringingdate']:=datetimepicker2.Date;
datasource2.Edit;
datasource2.DataSet.FieldValues['videoinstock']:=label10.Caption;
end;

procedure TForm21.Edit1Change(Sender: TObject);
begin
adoquery1.Close;
adoquery1.SQL.Clear ;
adoquery1.SQL.Add('select * from customers');
adoquery1.SQL.Add('where customername=:edit1.text');
adoquery1.Parameters.ParamByName('edit1.text').Value:=(edit1.text);
adoquery1.Open;
end;

procedure TForm21.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=200;
```

```
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=250;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=200;
statusbar1.panels.items[0].Text:='Gürsoy Video Center';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);
edit1.text:='';
button1.caption:='Calculate';
end;

procedure TForm21.Button2Click(Sender: TObject);
begin
form21.Hide;
end;
procedure TForm21.DBEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Video ID';
end;
procedure TForm21.DBEdit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Rental Id';
end;
procedure TForm21.DBEdit3MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Customer ID';
end;
procedure TForm21.DBEdit6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Quantity';
end;
procedure TForm21.DBEdit4MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
```

```
begin
statusbar1.Panels.Items[1].Text:='Unit Price';
end;
procedure TForm21.DBEdit7MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Total Price';
end;
procedure TForm21.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Calculates The Total Price';
end;
procedure TForm21.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
 Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='';
end;
procedure TForm21.SpeedButton1Click(Sender: TObject);
begin
form21.Hide;
end;
procedure TForm21.SpeedButton1MouseMove(Sender: TObject;
 Shift: TShiftState; X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quits the Page';
end;
end.
```

5.20 The Codes of "Rental Turn" Form

```
unit Unit22;
interface
uses
 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
```

```
 Dialogs, Grids, DBGrids, DB, ADODB, StdCtrls, Mask, DBCtrls, ExtCtrls,
 Buttons, jpeg;
type
 TForm22 = class(TForm)
   ADOConnection1: TADOConnection;
   Label1: TLabel;
   Label2: TLabel;
   Label3: TLabel;
   Label4: TLabel;
   Label5: TLabel;
   Label6: TLabel;
   Label7: TLabel;
   DBCheckBox1: TDBCheckBox;
   DBText1: TDBText;
   DBText2: TDBText;
   DBText3: TDBText;
   DBText4: TDBText;
   DBText5: TDBText;
   DBText6: TDBText;
   DBText7: TDBText;
   Label8: TLabel;
   ADODataSet1: TADODataSet;
   ADODataSet1RentalId: TAutoIncField;
   ADODataSet1VideoName: TWideStringField;
   ADODataSet1CustomerName: TWideStringField;
   ADODataSet1CustomerSurname: TWideStringField;
   ADODataSet1HireDate: TDateTimeField;
   ADODataSet1BringingDate: TDateTimeField;
   ADODataSet1Quantity: TIntegerField;
   ADODataSet1Turned: TBooleanField;
   ADODataSet1VideoInStock: TIntegerField;
   Label10: TLabel;
   DataSource1: TDataSource;
   Button1: TButton;
   ADOTable1: TADOTable;
   ADOTable1VideoId: TAutoIncField;
```

```
ADOTable1SupplierId: TIntegerField;
ADOTable1VideoName: TWideStringField;
ADOTable1VideoDesc: TMemoField;
ADOTable1VideoDirector: TWideStringField;
ADOTable1VideoType: TWideStringField;
ADOTable1VideoYear: TIntegerField;
ADOTable1UnitPrice: TBCDField;
ADOTable1VideoInStock: TIntegerField;
ADOTable1VideoPicture: TBlobField;
DataSource2: TDataSource;
DBGrid1: TDBGrid;
DBEdit2: TDBEdit;
SpeedButton1: TSpeedButton;
Image1: TImage;
procedure Button1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);

private
  { Private declarations }
public
  { Public declarations }
end;
var
 Form22: TForm22;
implementation
{$R *.dfm}

procedure TForm22.Button1Click(Sender: TObject);
var
x,y,z: integer ;
begin
x:= strtoint(dbedit2.Text);
y:= strtoint(dbtext7.caption);
if dbcheckbox1.Checked  then
z:= x+y;
```

```
label10.Caption:=inttostr(z);
datasource1.Edit;
datasource1.DataSet.FieldValues['videoinstock']:=label10.Caption;
end;
procedure TForm22.FormCreate(Sender: TObject);
begin
button1.Caption:='Confirm';
end;
procedure TForm22.SpeedButton1Click(Sender: TObject);
begin
form22.Hide;
end;
end.
```

5.21 The Codes of "Search For Bringing Date" Form

```
unit Unit23;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, ADODB, StdCtrls, ComCtrls, Grids, DBGrids, Mask, Buttons,
  jpeg, ExtCtrls;
type
  TForm23 = class(TForm)
    DBGrid1: TDBGrid;
    Button1: TButton;
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    MaskEdit1: TMaskEdit;
    MaskEdit2: TMaskEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
```

```
Label5: TLabel;
Button2: TButton;
StatusBar1: TStatusBar;
SpeedButton1: TSpeedButton;
Image1: TImage;
procedure Button1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure MaskEdit1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure MaskEdit2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
private
  { Private declarations }
public
  { Public declarations }
end;
var
 Form23: TForm23;
implementation
{$R *.dfm}
procedure TForm23.Button1Click(Sender: TObject);
begin
adoQuery1.Close;
adoQuery1.Sql.Clear;
adoQuery1.Sql.Add('Select * From rental');
```

```
adoQuery1.Sql.Add('Where bringingdate between:br1 AND :br2 and turned=false');
adoquery1.Parameters.ParamByName('br1').Value:=strtodate(maskedit1.text);
adoquery1.Parameters.ParamByName('br2').Value:=strtodate(maskedit2.text);
adoQuery1.open;
end;
procedure TForm23.FormCreate(Sender: TObject);
begin
statusbar1.Panels.Add;
statusbar1.panels.items[0].width:=200;
statusbar1.Panels.Add;
statusbar1.panels.items[1].width:=250;
statusbar1.Panels.Add;
statusbar1.panels.items[2].width:=200;
statusbar1.panels.items[0].Text:='Gursoy Video Center';
statusbar1.panels.items[2].Text:=DATETOSTR(NOW);
maskedit1.text:='';
maskedit2.text:='';
maskedit1.EditMask:='99.99.9999';
maskedit2.EditMask:='99.99.9999';
end;
procedure TForm23.Button2Click(Sender: TObject);
begin
maskedit1.text:='';
maskedit2.text:='';
maskedit1.EditMask:='99.99.9999';
maskedit2.EditMask:='99.99.9999';
end;
procedure TForm23.Button3Click(Sender: TObject);
begin
form23.Hide;
end;
procedure TForm23.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  statusbar1.Panels.Items[1].Text:='';
end;
```

```
procedure TForm23.MaskEdit1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter First Date as dd.mm.yyyy';
end;
procedure TForm23.MaskEdit2MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Enter Second Date as dd.mm.yyyy';
end;
procedure TForm23.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Starts Search';
end;
procedure TForm23.Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Clears The Page';
end;
procedure TForm23.SpeedButton1Click(Sender: TObject);
begin
form23.Hide;
end;
procedure TForm23.SpeedButton1MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
statusbar1.Panels.Items[1].Text:='Quits The Page';
end;
end.
```

5.22 The Codes of "Password Screen" Form

```
unit Unit24;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, DB, ADODB, jpeg;
type
  TForm24 = class(TForm)
    Image1: TImage;
    Edit1: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Edit2: TEdit;
    Button1: TButton;
    Button2: TButton;
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    Timer1: TTimer;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
      procedure Timer1Timer(Sender: TObject);
    procedure Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
```

```
 Form24: TForm24;
 tur: integer;
implementation
uses Unit2, Unit1;
{$R *.dfm}
procedure TForm24.Button1Click(Sender: TObject);
var
GirisPassword: String;
begin
    if tur=0 then
            Begin
            showmessage  ('The program Will Be Terminated');
            application.terminate;
             End;
  If Self.edit1.Text = '' Then Begin
    Application.MessageBox('You must enter user name...','Error!!!');
    edit1.SetFocus;
    Exit;
  End;
  AdoQuery1.SQL.Text := 'SELECT * FROM Lock WHERE
UserName='''+Trim(edit1.Text)+'''';
  AdoQuery1.Open;
  GirisPassword := Trim(AdoQuery1.FieldByName('Pass').AsString);
  If AdoQuery1.RecordCount = 0 Then
    Begin
    Application.MessageBox('There arent any users like that...','Error!');
    tur:=tur-1;
    AdoQuery1.Close;
   if tur=0 then
            Begin
            showmessage  ('The program Will Be Terminated');
            application.terminate;
             End; Exit;
    End
     Else
      Begin
```

```
      If GirisPassword = edit2.Text Then
        Begin
         form1.visible := True;
         form24.Visible := False;
        End
         Else
          Begin
            tur:=tur-1;
           Application.MessageBox('Wrong Password','Error!');
              if tur=0 then
              Begin
              showmessage  ('The program Will Be Terminated');
              application.terminate;
               End;
     end;
  End;
  AdoQuery1.Close;
end;
procedure TForm24.FormCreate(Sender: TObject);
begin
tur:=3;
form24.Caption :='Gursoy Video Center';
button1.Font.Size:=8;
button2.Font.Size:=8;
end;
procedure TForm24.Button2Click(Sender: TObject);
begin
close;
end;
procedure TForm24.Timer1Timer(Sender: TObject);
begin
Caption:=copy(caption,2,length(caption)-1)+caption[1];
end;
procedure TForm24.Button1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
```

```
button1.Font.Size:=11;
end;
procedure TForm24.Button2MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
button2.Font.Size:=11;
end;
procedure TForm24.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
   button1.Font.Size:=9;
button2.Font.Size:=9;
end;
end.
```

5.23 The Codes of "About me" Form

```
unit Unit25;

interface
uses Windows, SysUtils, Classes, Graphics, Forms, Controls, StdCtrls,
  Buttons, ExtCtrls, jpeg, OleCtrls, ShockwaveFlashObjects_TLB;
type
  Tform25 = class(TForm)
    Panel1: TPanel;
    Version: TLabel;
    OKButton: TButton;
    Label1: TLabel;
    Label4: TLabel;
    Label2: TLabel;
    Label6: TLabel;
    Label8: TLabel;
    Bevel1: TBevel;
```

```
  Image1: TImage;
  procedure OKButtonClick(Sender: TObject);
  procedure FormCreate(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;
var
 form25: Tform25;
implementation
{$R *.dfm}
procedure Tform25.OKButtonClick(Sender: TObject);
begin
form25.Hide;
end;
procedure Tform25.FormCreate(Sender: TObject);
begin
form25.Caption:='About Me';
end;
end.
```

5.24 The Codes of " Welcome Screen" Form

```
unit Unit26;

interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, jpeg, ExtCtrls;
type
  TForm26 = class(TForm)
   Image1: TImage;
   procedure FormCreate(Sender: TObject);
```

```
private
  { Private declarations }
public
  { Public declarations }
end;
var
 Form26: TForm26;
implementation
{$R *.dfm}
procedure TForm26.FormCreate(Sender: TObject);
begin
end;
end.
```

5.25 The Codes of "Video trailers" Form

```
unit Unit27;

interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, jpeg, ExtCtrls, StdCtrls;
type
  TForm27 = class(TForm)
    Image1: TImage;
    Image2: TImage;
    Image3: TImage;
    Label1: TLabel;
    Image4: TImage;
    Image5: TImage;
    Image6: TImage;
    Image7: TImage;
    Image8: TImage;
```

```
    Image9: TImage;
    procedure Image2Click(Sender: TObject);
    procedure Image3Click(Sender: TObject);
    procedure Image4Click(Sender: TObject);
    procedure Image5Click(Sender: TObject);
    procedure Image6Click(Sender: TObject);
    procedure Image7Click(Sender: TObject);
    procedure Image8Click(Sender: TObject);
    procedure Image9Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form27: TForm27;
implementation
uses Unit28, Unit29, Unit30, Unit31, Unit32, Unit33, Unit34, Unit35;
{$R *.dfm}
procedure TForm27.Image2Click(Sender: TObject);
begin
form28.show;
end;
procedure TForm27.Image3Click(Sender: TObject);
begin
form29.show;
end;
procedure TForm27.Image4Click(Sender: TObject);
begin
form30.show;
end;
procedure TForm27.Image5Click(Sender: TObject);
begin
Form31.Show;
end;
```

```
procedure TForm27.Image6Click(Sender: TObject);
begin
form32.Show;
end;
procedure TForm27.Image7Click(Sender: TObject);
begin
Form33.Show;
end;
procedure TForm27.Image8Click(Sender: TObject);
begin
form34.Show;
end;
procedure TForm27.Image9Click(Sender: TObject);
begin
form35.Show;
end;
procedure TForm27.FormCreate(Sender: TObject);
begin
end;
end.
```

5.23 The Codes of "One Video Trailers"Form

```
unit Unit28;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, OleCtrls, ShockwaveFlashObjects_TLB, jpeg, ExtCtrls, WMPLib_TLB;
type
  TForm28 = class(TForm)
    WindowsMediaPlayer1: TWindowsMediaPlayer;
    procedure WindowsMediaPlayer1Exit(Sender: TObject);
    procedure FormCreate(Sender: TObject);
```

```
 private
   { Private declarations }
 public
   { Public declarations }
 end;
var
 Form28: TForm28;
implementation
{$R *.dfm}

procedure TForm28.WindowsMediaPlayer1Exit(Sender: TObject);
begin
halt;
end;
procedure TForm28.FormCreate(Sender: TObject);
begin
end;
end.
```