



**NEAR EAST UNIVERSITY**

**Faculty of Engineering**

**Department of Computer Engineering**

**GRADUATION PROJECT  
COM – 400**

**SUBSCRIPTION TRACKING PROGRAM**

**Student : Muhammed TEKİN**

**Supervisor : ASSOC. PROF. DR. RAHİB  
ABİYEV**

**Nicosia - 2004**

## **ACKNOWLEDGMENTS**

First, I want to thank my wonderful teacher ASSOC. PROF. DR. RAHİB ABİYEV for his infinite helpful attitudes. I cannot finish the project without his helps.

And I want to thank my family for the things they had to put up with until they bought me to the age that I am. Especially during the time of my life in the faculty.

Also I have to thank all the instructors for their helpful attitudes.

And lastly thank my friends who helped me to for the project.

## ABSTRACT

As the information age has affected every aspect of our life, the need for computerizing many information systems has raised.

Once of the important branches that are affected by information revolution is the computer programming languages.

This project is written using Visual Basic 6.0 programming language and Microsoft Access 2002 database language is used for database. Visual Basic is one of the best and easy programming languages.

Aim of this project is to track the subscriptions for ALTINOLUK magazine. This program must do these operations:

- Adding a new subscription to the database
- Updating any data for an existing record
- Keeping dept and payment records for subscriptions and group managers
- Canceling an existing record
- Listing last operations for records
- Listing distribution lists for group managers

Before coming to this point, this project has gone through some important steps;

- First one was that I had to have some knowledge about how subscription records to make and learn subscription record working systems for the requirement definitions. So, I examined some subscription tracking programs and met people I know who working in that magazine company.
- Second step was to design and to put in order information about the program.

The later steps were steps of the implementation of the designed information on computer by using Visual Basic Language.

## **TABLE OF CONTENTS**

1) INTRODUCTION .....	1
2) SUBSCRIPTION TRACKING PROGRAM PROBLEMS .....	3
3) DATABASE STRUCTURE .....	5
4) DEVELOPMENT OF SUBSCRIPTION TRACKING PROGRAM .....	11
4.1) LOG ON SCREEN .....	11
4.2) MAIN FORM SCREEN .....	12
4.3) MAKE PAYMENT SCREEN .....	13
4.4) SEARCH SCREEN .....	14
4.5) LISTS SCREEN .....	15
4.6) DISTRIBUTION LISTS SCREEN .....	16
4.7) PARAMETERS SCREEN .....	17
4.8) USER ACCOUNTS SCREEN .....	18
5) CONCLUSION .....	20
6) REFERENCES .....	21
7) APPENDIX .....	22
7.1) frmAnaPencere SOURCE CODES .....	22
7.2) frmAra SOURCE CODES .....	36
7.3) frmDağıtımListeleri SOURCE CODES .....	41
7.4) frmKullanıcıHesapları SOURCE CODES .....	42
7.5) frmListeler SOURCE CODES .....	44
7.6) frmLogOn SOURCE CODES .....	46
7.7) frmÖdemeYap SOURCE CODES .....	48
7.8) frmParametreGirişi SOURCE CODES .....	50
7.9) frmYeniKullanıcıEkle SOURCE CODES .....	56

7.10) AnaPencereModülü SOURCE CODES .....	59
7.11) AraçlarModülü SOURCE CODES .....	81
7.12) HataVeUyarıModülü .....	89
7.13) VeriTabanıModülü .....	94



## **1. INTRODUCTION**

Visual Basic is a Microsoft Windows programming Language. Visual Basic programs are created in an Integrated Development Environment (IDE). The IDE allows the programmer to create, run and debug Visual Basic programs conveniently. IDEs allow a programmer to create working programs in a fraction of the time that it would normally take to code programs without using IDEs. The process of rapidly creating an application is typically referred to as Rapid Application Development (RAD). Visual Basic is the world's most widely used RAD language.

Visual Basic is derived from the BASIC programming language. Visual Basic is a distinctly different language providing powerful features such as graphical user interfaces, even handling, access to the Win32 API, object-oriented features, error handling, structured programming, and much more.

The Visual Basic IDE allows Windows programs to be created without the need for the programmer to be a Windows programming expert.

Microsoft provides several version of Visual Basic, namely the Learning Edition, the Professional Edition and the Enterprise Edition. The Learning Edition provides fundamental programming capabilities than the Learning Edition and is the choice of many programmers to write Visual Basic applications. The Enterprise Edition is used for developing large-scale computing systems that meet the needs of substantial organizations.

Visual Basic is an interpreted language. However, the professional and Enterprise Edition allows Visual Basic code to be compiled to native code.

Visual Basic evolved from BASIC (Beginner's all purpose Symbolic Instruction Code). Basic was developed in the mid 1960's by Professors John Kemeny and Thomas Kurtz of Dartmouth College as a language for writing simple programs. BASIC's primary purpose was to help people learn how to program.

The widespread use of BASIC with various types of computers (sometimes called hardware platforms) led to many enhancements to the language. With the development of the Microsoft windows graphical user interface (GUI) in the late 1980s and the early 1990s, the natural evolution of BASIC was Visual Basic, which was created by Microsoft Corporation in 1991.

Until Visual Basic appeared, developing Microsoft Windows-based applications was a difficult and cumbersome process. Visual Basic greatly simplifies Windows application development. Since 1991 six versions have been released, with the latest- Visual Basic 6-appearing in September 1998.

After a brief explanation about the Visual Basic 6.0 and the developing layers, I hope that you will find the necessary information that you need all about the Visual Basic even if you are a text based programmer.

## **2. SUBSCRIPTION TRACKING PROGRAM PROBLEMS**

Aim of this project is to track the subscriptions for ALTINOLUK magazine.

This program must do these operations:

- Adding a new subscription to the database
  - a. If he/she subscriptions himself/herself, then open an account for this record
  - b. If not, record him/her to the specified group and add the subscription fee to that group.
- Updating any data for an existing record
  - a. If he/she is not in a group and has some dept, then do not allow to update the beginning issue
  - b. If he/she is a group manager or he/she does not have any dept, then update the informations
  - c. Always check what is updated and save it to last changes tables
- Keeping dept and payment records for subscriptions and group managers
- Canceling an existing record
  - a. If he/she is a group member, then ask the money that will be decreased from the dept of that group
- Listing last operations for records
- Listing distribution lists for group managers
- Search the records that match the given informations
- Show the records that have dept
- Show the records that are canceled
- Keep the program in safe by using a logon screen
- Rapidly change the user that is logged on
- Keep the user accounts
- Add, delete, update the parameters



Before coming to this point, this project has gone through some important steps;

- First one was that I had to have some knowledge about how subscription records to make and learn subscription record working systems for the requirement definitions. So, I examined some subscription tracking programs and met people I know who working in that magazine company.
- Second step was to design and to put in order information about the program.

### 3. DATABASE STRUCTURE

I have used 16 tables in the database. My database name is ALTINOLUK.MDB. It is createn by using Microsoft Access 2002. There are a lot of fields in those tables. Let me explain all the tables and fields:

- ABONELER table:

<u>NAME</u>	<u>TYPE</u>	<u>MAX</u>
SıraNo	Automatic	
AboneNo	Text	10
Ad	Text	50
Soyad	Text	50
Adres	Memo	
EvTel	Text	20
İşTel	Text	20
CepTel	Text	20
Fax	Text	20
ÜlkeKodu	Long	
İlKodu	Long	
İlçeKodu	Long	
BaşSay	Long	
AboneYapanNo	Long	
DağıtımıcıNo	Long	
HediyeKitapVerildiMi	Boolean	
HediyeKitapKodu	Long	
Açıklama	Memo	
İlkKayıt	Long	
EnSonYenileme	Long	
EnSonAdresDeğişikliği	Long	
EnSonİsimDeğişikliği	Long	
İptalEdildiMi	Boolean	
İptalTarihi	Long	
AbonelikÜcreti	Long	
ParaBirimiKodu	Long	

## DESCRIPTION:

This is the main table in the database. All subscription records are here. When updating name/surname, adres, and beginning issue we use the appropriate fields to record the date of change. Other informations are recorded appropriate fields, that's name goes to Ad field, address goes to Adres field etc.

- ENSONADRESİDEĞİŞTİRENLER table:

<u>NAME</u>	<u>TYPE</u>
SıraNo	Automatic

- ENSONADSOYADDEĞİŞTİRENLER table:

<u>NAME</u>	<u>TYPE</u>
SıraNo	Automatic

- ENSONİPTALEDİLENLERLER table:

<u>NAME</u>	<u>TYPE</u>
SıraNo	Automatic

- ENSONYENİKAYITLAR table:

<u>NAME</u>	<u>TYPE</u>
SıraNo	Automatic

- ENSONYENİLENENLER table:

<u>NAME</u>	<u>TYPE</u>
SıraNo	Automatic

### **DESCRIPTION:**

These five tables are used to report the last changes. We only need the number of the subscription. Then we can get all informations from that record by using Aboneler table.

- GRUPHESAPLARI table:

<u>NAME</u>	<u>TYPE</u>
GrupNo	Automatic
ToplamBorç	Long
KalanBorç	Long
ParaBirimiKodu	Long

### **DESCRIPTION:**

This table is used to keep the accounts of the groups. When recording a subscription to Aboneler table, if the record is a group member than we have to add its fee to this table.

- GRUPLAR table:

<u>NAME</u>	<u>TYPE</u>
GrupNo	Automatic
Ad	Text
Başkan	Long

### **DESCRIPTION:**

This table keeps the groups. If a group does not have a member, then Başkan field is filled with 0, else the SıraNo field of Aboneler table is token and recorded to here.

- HEDİYEKİTAPLAR table:

<b><u>NAME</u></b>	<b><u>TYPE</u></b>
Kod	Long
Ad	Text

### **DESCRIPTION:**

This table is only responsible to keep the gift book names. When updating beginning issue for each year, we give a gift book.

- İLÇELER table:

<b><u>NAME</u></b>	<b><u>TYPE</u></b>
Kod	Automatic
İlKodu	Long
Ad	Text

- İLLER table:

<b><u>NAME</u></b>	<b><u>TYPE</u></b>
Kod	Automatic
İlKodu	Long
Ad	Text



- PARABİRİMLERİ table:

<u>NAME</u>	<u>TYPE</u>
Kod	Automatic
Ad	Text

- ÜLKELER table:

<u>NAME</u>	<u>TYPE</u>
Kod	Automatic
Ad	Text

### **DESCRIPTION:**

This four tables are for parameters. Every of them has its unique number and name. But for city and district, we must specify a key, for city the number of the country and for district the number of the city.

- KİŞİSELHESAPLAR table:

<u>NAME</u>	<u>TYPE</u>
AboneSıraNo	Long
KalanBorç	Long

### **DESCRIPTION:**

This table is used to see the dept of a subscription that he/she was recorded as himself/herself. No group members are recorded here. They are recorded in GrupHesapları table.

- KULLANICIHESAPLARI table:

<u>NAME</u>	<u>TYPE</u>
Ad	Text
Parola	Text
YöneticiMi	Boolean

### **DESCRIPTION:**

This table is used to keep the user of the program. It records a user name and its password. Also we have to specify either it is an administrator or guest user.

- ÖDEMELER table:

<u>NAME</u>	<u>TYPE</u>
SıraNo	Long
Miktar	Long
ParaBirimiKodu	Long
Tarih	Long
Açıklama	Memo

### **DESCRIPTION:**

This table is used for all subscriptions. When we pay some money we record it here. By summing all Miktar fields, we can calculate the amount of payment.

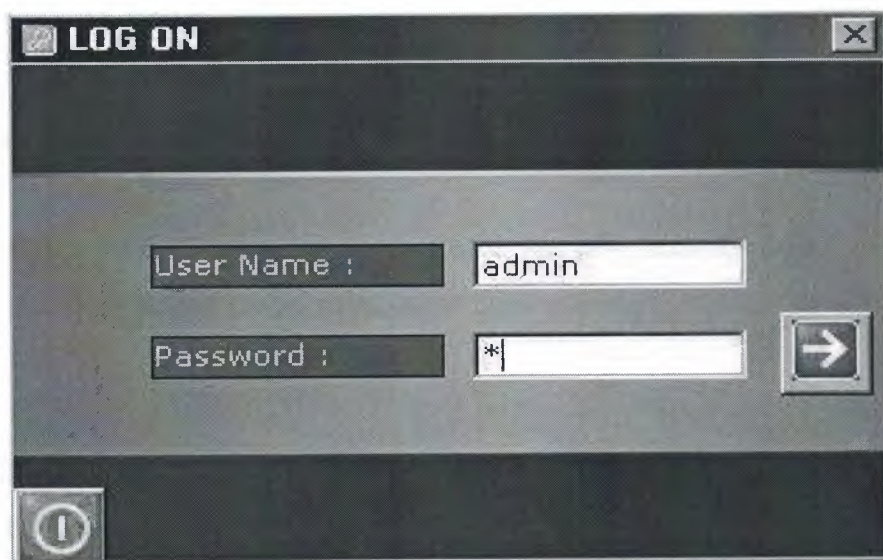
## **4. DEVELOPMENT OF SUBSCRIPTION TRACKING PROGRAM**

This section is about the program and the windows. There 8 windows in this program. All windows include some operations. Now it is time to explain the operations and windows.

### **4.1 LOG ON SCREEN**

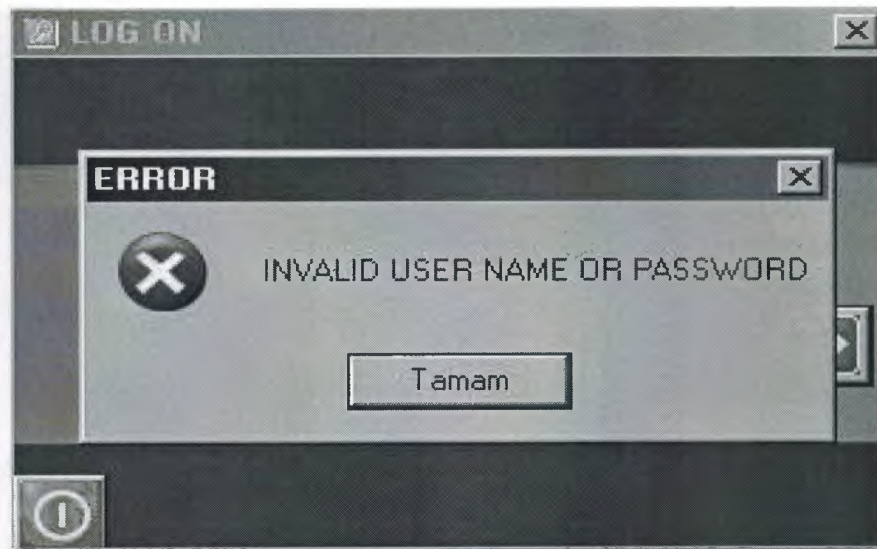
This is the first window of the program. In this window, the user must enter a valid user name and password. After, he/she must click the button that has an arrow top of it. After that, the program is going to check the informations entered and if there is such a user, than it will show the main window, but if not, then it is going to show the a warning message. That means, unless you enter a valid user name and password, you are not going to be able to log on program.

Also the user can directly exit the program by clicking the button on the left hand side.



Enter a valid user name and password





The warning message shows that you entered a wrong user name or password

## 4.2 MAIN FORM SCREEN

This form is main form which is appearing during program running and includes sub menus as adding new record, save/update, search, lists etc. on its top. On the main menu, we reach sub forms by using these sub menus.

When program is started, mission of main menu is to fill in the record names to listbox if there are any records. And select the first record and show the informations of this record. If there is no record, then it is going to close the information tab.

The operation buttons on the left side are new record, update, cancel, post sticker, clear, search, the records that have been canceled, and the records that have dept.

The operation buttons on the right side are lists, distribution lists, parameters, user accounts, about, help, log off, and exit.

Main window of the program

### 4.3 MAKE PAYMENT SCREEN

This form is used to pay the dept of an existing record. If the record is a group member and if he is not the manager of that group, then the make payment button in the main window will not be accessible. This window only works for subscriptions who are a manager of a group or not in a group.

For the groups, we have to choose the money that we are paying. Because, inside a group, there could be different subscribers that have different currency. But for subscribers that have subscribed themselves, there is only one money unit.

Also when we paid all of the dept, the make payment button will not be accesable.

These payment transactions are shown in the information tab three.



Enter a valid dept and choose the currency to pay the dept

#### 4.4 SEARCH SCREEN

This form is used to search any record that matches given data here. If all cells are empty then it will show all records in subscription name lists in the main window but if at least one of them has some data in it then it will search the records that have that data.

The connection between the informations is made upon with AND Boolean operation. That is to say, if we enter name and surname, it is going to check the records that have same name and surname.

If we want to consider first subscription date or last update date, then we have to open the date cells, otherwise close them.

**SEARCH**

NAME:  SURNAME:  NO:  BEGINING ISSUE:

FIRST SUBSCRIPTION:  LAST UPDATE:

PRICE & PRICE UNIT:

OPEN / CLOSE OPEN / CLOSE

ADDRESS:

COUNTRY:  CITY:  DISTRICT:

SUBSCRIBER & DISTRIBUTOR: ☐ HIMSELF / HERSELF ☐ GROUP MEMBER  ☐ NO MATTER DISTRIBUTOR NO:

DESCRIPTION:

IS GIFT BOOK GIVEN ? ☐ YES ☐ NO ☒ NO MATTER

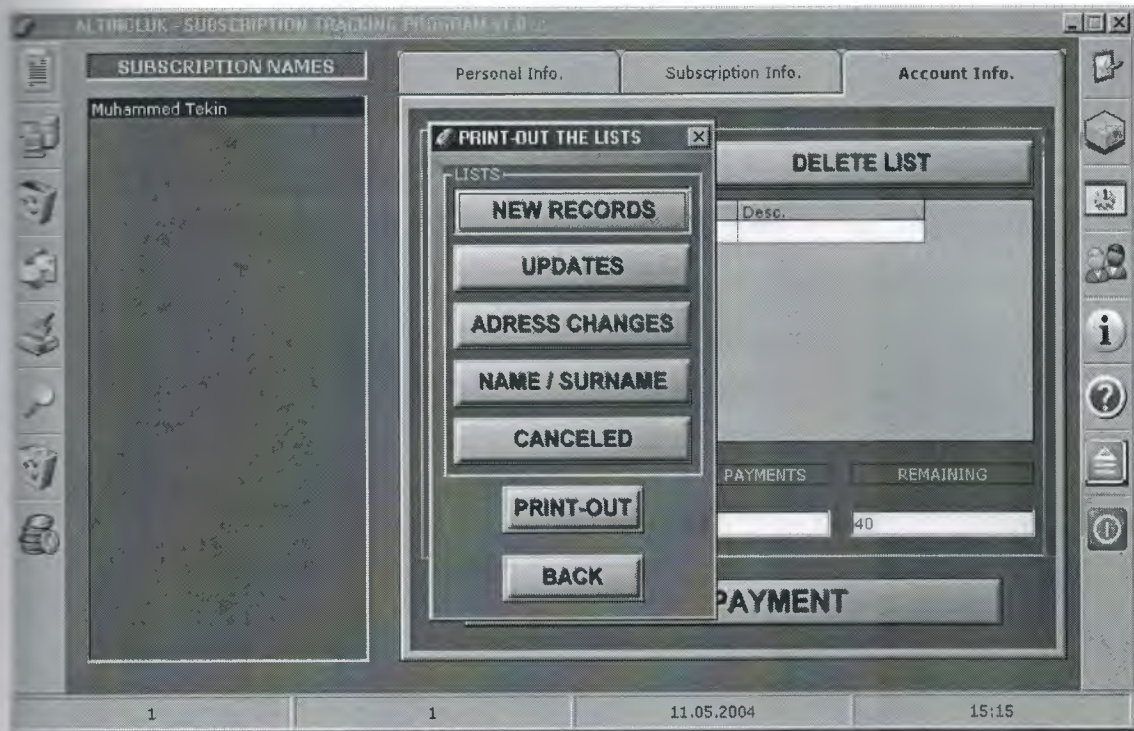
**SEARCH** **BACK**

Enter the informations that are need to search operation

## 4.5 LISTS SCREEN

This form is used to print-out of the last new record list, last updates list, last address changes list, last name/surname changes list, last canceled list. When one of the list is chosen the user will click to print-out button and it will show print-out preview. Also it will ask to delete the last corresponding list (for example, last new records list)





Choose the list that you want to print-out

#### 4.6 DISTRIBUTION LISTS SCREEN

This form is used to print-out of the distribution lists. First, the user must enter an issue (for example this month the issue of the magazine is 200) then he must enter the distribution number for which he wanted to print-out.

After that, it will print-out the records that have the same distribution number and their beginning issue is less than or equal to issue number and ending issue is greater than or equal to issue number.

This print-out is needed for each month to the distributors. Because, some of the magazines are not send to address. The distributors will bring them to the subscriber.

You can directly return to main screen by clicking the button back.

Enter the issue and distributor number

#### 4.7 PARAMETERS SCREEN

This form is used to adjust the parameters used in the program. Under the parameters text there is a combobox which has the parameter types used in the program.

For example when we choose country, values are loaded to next combobox. If we want to add a new country then we can write it down in the combobox but if we want to update it then we have to use the textbox under this combobox. Also if that parameter is not used then we can delete it here. To add or modify a city, at first we have to choose a country from the right combobox. Also same thing is valid for district, plus we have to choose a city.

You can delete any unused parameter. When you want to delete a parameter, it is going to check whether it is used or not. If it is not used, then you will be able to delete it.



ALTINILUK - SUBSCRIPTION TRACKING PROGRAM #1.0

**SUBSCRIPTION NAMES**

Muhammed Tekin

Personal Info. Subscription Info. Account Info.

FORMER PAYMENTS

PRICE UNIT: \$

**DELETE LIST**

Amount	Unit	Date	Desc.

**PARAMETERS**

PARAMETERS VALUES CITIES COUNTRIES

COUNTRIES Türkiye

SAVE / UPDATE DELETE

TOTAL DEPT TOTAL PAYMENTS REMAINING

40 0 40

**MAKE PAYMENT**

1 1 11.05.2004 15:16

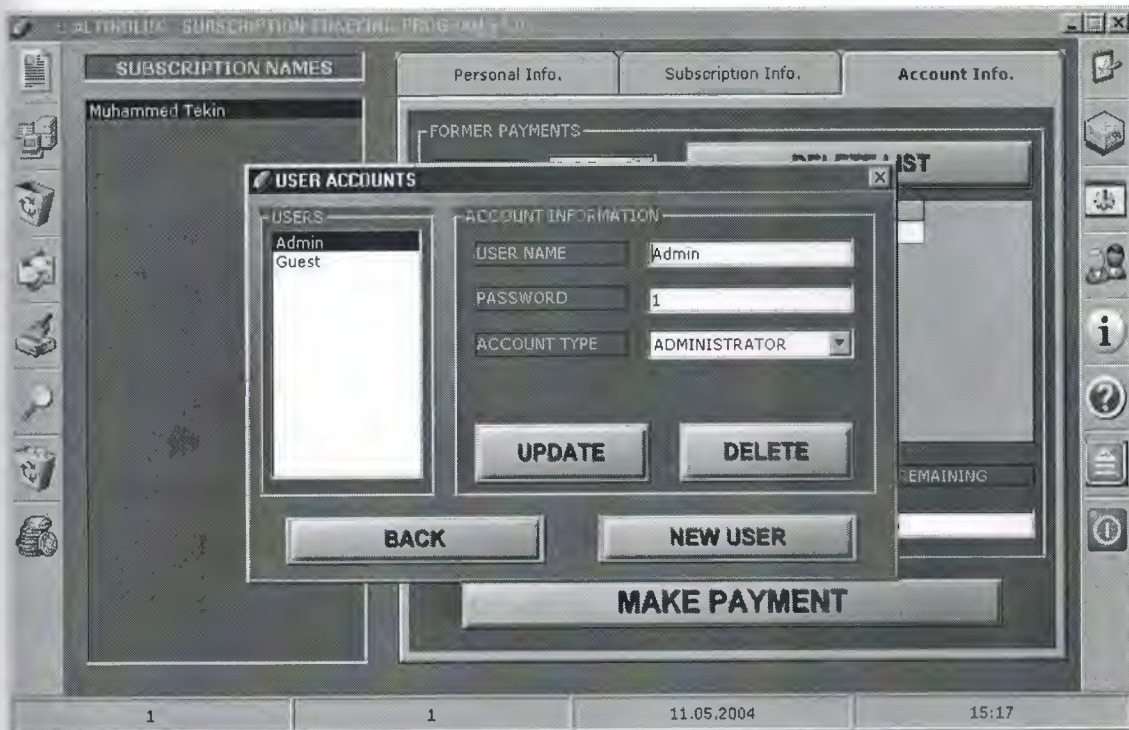
Choose a parameter to add or modify

## 4.8 USER ACCOUNTS SCREEN

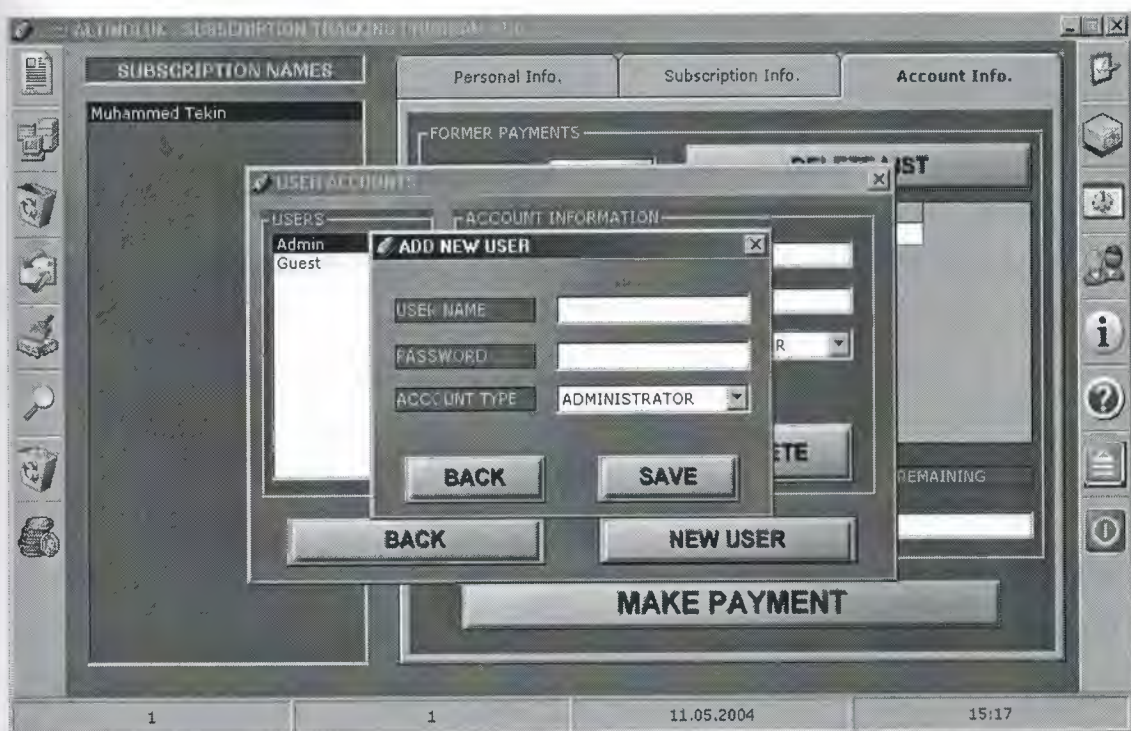
This form is used to change the password of an existing account or to add a new account or delete an existing account.

To add a new one, the user must click new user button, and a new form will be displayed. Here he must enter a unique user name that's it should not been recorded, also he must enter a password and he must specify the type of this account. It can be an administrator account or guest account. If an account is guest in main window some of the buttons will be disabled. Also to delete an account there must be minimum 2 account recorded.





Update, delete or add a new user account here



Enter a uniq user name and password and choose the account type

## **5. CONCLUSION**

I have demonstrated a small program which deals with data and processes them. This demonstration has been valuable and helpful for the future so that we can write programs much more accurate than the first attempt.

I have discovered that once a program is needed to be written firstly we must discover the needs and wants of the user. Once the flow chart of the system is in picture half of the program is counted as done. We have discovered that to do something well you must understand and apply to your work. With Visual Basic we can construct programs which can help a lot of users.

Nowadays, windows oriented programs became more popular and flexible. Visual Basic 6.0 is one of the best well-known programming languages based on window's environment. That's why I prefer this project. Now I can understand why these programming languages are very popular. Even I do not have experience with Visual Basic; this project did not become difficult to me. Visual Basic 6.0 has lots of help than other programming languages.

In my project, I have used important components of Visual Basic 6.0. Therefore I learned these components very well. Now I can use these components of Visual Basic 6.0 in an efficient manner. Also I have learned how to use new data access logic, which is ActiveX Data Objects (ADO). Additionally, I have used a database in my project. So I have gained many practices, experiences and knowledge of database. As known, database is very important topic for software programmers.

Finally, most important thing is for me that I have learned how to prepare an individual software project by using Visual Basic 6.0 to real life problems. After I have started my projects, I saw that you could face with unexpected real life problems. These real life problems are very different from the courses problem. This project became a good exercise to me for the real life and I used the things in my project that I learned from courses as theoretically.



## 6. REFERENCES

### 1. Books

- Visual Basic 6.0 / Zaydin Pala / Türkmen Kitabevi
- Visual Basic 6.0 / Mithat Uysal / Beta Yayınları
- Visual Basic 6.0 / Memik Yanık / Beta Yayınları
- Visual Basic 6.0 How to Program / H. M. Deitel
- SQL / James R. Groff / Alfa Yayınları
- Database System / Peter Roob / Wads Worth Publishing

### 2. Web Sites

- [www.microsoft.com](http://www.microsoft.com)
- [www.altinoluk.com](http://www.altinoluk.com)
- [www.alfa.com.tr](http://www.alfa.com.tr)
- [www.turkmenkitapevi.com.tr](http://www.turkmenkitapevi.com.tr)
- [www.beta.com.tr](http://www.beta.com.tr)

## 7. APPENDIX

Here is, the source codes of the program. For each window it is written name of the form. Also for modules module name is written firstly.

### 7.1 frmAnaPencere SOURCE CODES

Option Explicit

Private Sub lstAboneİsimleri\_Click()

    AboneBilgileriniGöster

End Sub

Private Sub lstAboneİsimleri\_KeyDown(KeyCode As Integer, Shift As Integer)

    Dim lTMP As Long

    If tbrSolAraçÇubuğu.Buttons.Item(5).Enabled And KeyCode = vbKeyDelete

Then 'silme açıksa sil

    If lstAboneİsimleri.ListIndex > -1 Then

        If Not grKayıt.Collect("İptalEdildiMi") Then

            If vUyarı(AboneİptalEtmeOnayı, vbYesNo + vbDefaultButton2 +  
vbQuestion) = vbYes Then

                grKayıt("İptalEdildiMi") = True

                grKayıt("İptalTarihi") = CLng(Date)

                grKayıt.Update

                If grKayıt.Collect("AboneYapanNo") = KENDİSİABONENO Then

                    If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT \* FROM  
KişiselHesaplar WHERE AboneSıraNo=" & grKayıt.Collect("SıraNo")) Then

                        If grYedekKayıt.RecordCount = 1 Then

                            grYedekKayıt.Delete

                            grYedekKayıt.Update

                        End If

                    End If

```

        If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT * FROM
Ödemeler WHERE SıraNo=" & grKayıt.Collect("SıraNo")) Then
            If grYedekKayıt.RecordCount = 1 Then
                grYedekKayıt.Delete
                grYedekKayıt.Update
            End If
        End If
    Else
        ITMP = Val(InputBox("Amount : ", "THE DEPT OF THIS
RECORD", 0))
        If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT * FROM
GrupHesapları WHERE GrupNo=" & grKayıt.Collect("AboneYapanNo") & " AND
ParaBirimiKodu=" & grKayıt.Collect("ParaBirimiKodu")) Then
            If grYedekKayıt.RecordCount = 1 Then
                grYedekKayıt("ToplamBorç") =
grYedekKayıt.Collect("ToplamBorç") - grKayıt.Collect("AbonelikÜcreti")
                grYedekKayıt("KalanBorç") = grYedekKayıt("KalanBorç") -
ITMP
                grYedekKayıt.Update
            End If
        End If
        If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT * FROM
Gruplar WHERE Başkan=" & grKayıt.Collect("SıraNo")) Then
            If grYedekKayıt.RecordCount = 1 Then
                grYedekKayıt("Başkan") = 0
                grYedekKayıt.Update
            End If
        End If
        If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT * FROM
EnSonİptalEdilenler") Then
            grYedekKayıt.AddNew
            grYedekKayıt("SıraNo") = grKayıt.Collect("SıraNo")
            grYedekKayıt.Update

```



```

End If
If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT * FROM
HaberGrubu WHERE SıraNo=" & grKayıt.Collect("SıraNo")) Then
    If grYedekKayıt.RecordCount = 1 Then
        grYedekKayıt.Delete
        grYedekKayıt.Update
    End If
End If
sstBilgiler.Tab = 0
HücreİçerikleriniTemizle
If fListBoxElemanSil(lstAboneİsimleri, lstAboneİsimleri.ListIndex)
Then
    End If
    DurumÇubuğunuGüncelle
    lstAboneİsimleri.SetFocus
End If
End If
End If
End Sub

```

```

Private Sub tbrSolAraçÇubuğu_ButtonClick(ByVal Button As
MSComctlLib.Button)
    Select Case Button.Key
        Case "YeniKayıt"
            Form_KeyDown vbKeyF2, 0
        Case "KaydetGüncelle"
            Form_KeyDown vbKeyF3, 0
        Case "İptalEt"
            lstAboneİsimleri_KeyDown vbKeyDelete, 0
        Case "PostaEtiketiYazdır"
            Form_KeyDown vbKeyF6, 0
        Case "Temizle"
            Form_KeyDown vbKeyF5, 0
    End Select
End Sub

```

```

Case "Ara"
    Form_KeyDown vbKeyF4, 0
Case "İptalEdilenler"
    KayıtDüğmeleriniKapat
    Form_KeyDown vbKeyF7, 0
Case "BorcuOlanlar"
    KayıtDüğmeleriniKapat
    Form_KeyDown vbKeyF8, 0
End Select
End Sub

Private Sub tbrSağAraçÇubuğu_ButtonClick(ByVal Button As
MSComctlLib.Button)
    Select Case Button.Key
        Case "Listeler"
            Form_KeyDown vbKeyF9, 0
        Case "DağıtımçıListeleri"
            Form_KeyDown vbKeyF10, 0
        Case "Parametreler"
            frmAnaPencere.Enabled = False
            frmParametreGirişi.Show
        Case "KullanıcıHesapları"
            frmAnaPencere.Enabled = False
            frmKullanıcıHesapları.Show
        Case "Hakkında"
            MsgBox "NEAR EAST UNIVERSITY - COMPUTER ENGINEERING" &
vbCr & vbCr & "GRADUATION PROJECT" & vbCr & vbCr & "MUHAMMED
TEKİN - 990485", vbOKOnly, "ALTINOLUK - SUBSCRIPTION TRACKING
PROGRAM v1.0"
        Case "Yardım"
            Form_KeyDown vbKeyF1, 0
        Case "LogOff"
            gvCevap = vUyarı(Çıkış, vbQuestion + vbYesNo, "LOG OFF")
            If gvCevap = vbYes Then

```

```

        If fVeriTabanıKapat(gnBağlantı, grKayıt) Then
            frmLogOn.Show
        Else
            Hata VeriTabanıKapatılamadı
        End If
        gfSormadanÇık = True
        Unload Me
    End If
    Case "Çıkış"
        Form_KeyDown vbKeyEscape, 0
    End Select
End Sub

Private Sub txtAd_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then
        txtSoyad.SetFocus
    End If
End Sub

Private Sub txtSoyad_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then
        rtfAdres.SetFocus
    End If
End Sub

Private Sub cboÜlke_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then
        cboİl.SetFocus
    End If
End Sub

Private Sub cboÜlke_Click()
    cboİl.Clear
    cboİlçe.Clear

```

```

If gfNesneClickleriÇalışabilirMi Then
    ComboBoxGüncelle cboİl, gnBağlantı, grYedekKayıt, "SELECT * FROM İller
WHERE ÜlkeKodu=(SELECT Kod FROM Ülkeler WHERE Ad="" &
cboÜlke.Text & " ')" & " ORDER BY Ad", "Ad", True
End If
End Sub

```

```

Private Sub cboÜlke_Change()
    cboÜlke_Click
End Sub

```

```

Private Sub cboİl_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then
        cboİlçe.SetFocus
    End If
End Sub

```

```

Private Sub cboİl_Click()
    cboİlçe.Clear
    If gfNesneClickleriÇalışabilirMi Then
        ComboBoxGüncelle cboİlçe, gnBağlantı, grYedekKayıt, "SELECT * FROM
İlçeler WHERE İlKodu=(SELECT Kod FROM İller WHERE Ad="" & cboİl.Text &
"" AND ÜlkeKodu=(SELECT Kod FROM Ülkeler WHERE Ad="" & cboÜlke.Text
& " ')" & " ORDER BY Ad", "Ad", True
    End If
End Sub

```

```

Private Sub cboİl_Change()
    cboİl_Click
End Sub

```

```

Private Sub cboİlçe_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then
        txtEvTel.SetFocus
    End If
End Sub

```



End If

End Sub

Private Sub txtEvTel\_KeyDown(KeyCode As Integer, Shift As Integer)

    If KeyCode = vbKeyReturn Then

        txtİşTel.SetFocus

    End If

End Sub

Private Sub txtİşTel\_KeyDown(KeyCode As Integer, Shift As Integer)

    If KeyCode = vbKeyReturn Then

        txtCepTel.SetFocus

    End If

End Sub

Private Sub txtCepTel\_KeyDown(KeyCode As Integer, Shift As Integer)

    If KeyCode = vbKeyReturn Then

        txtFax.SetFocus

    End If

End Sub

Private Sub txtFax\_KeyDown(KeyCode As Integer, Shift As Integer)

    If KeyCode = vbKeyReturn Then

        rtfAçıklama.SetFocus

    End If

End Sub

Private Sub txtAboneNo\_KeyDown(KeyCode As Integer, Shift As Integer)

    If KeyCode = vbKeyReturn Then

        txtBaşSay.SetFocus

    End If

End Sub

Private Sub txtBaşSay\_Change()

```

OnBirSayıSonrasınıHesapla txtBaşSay, txtBitSay
'başsay değiştiğine göre yenileme yapılacak o zaman yeni abonelik ücretini
'girmeyi unutmasın diye orayı silelim
txtÜcret.Text = ""
ComboBoxTemizle cboParaBirimi1
End Sub

Private Sub txtBaşSay_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then
        txtBitSay.SetFocus
    End If
End Sub

Private Sub txtBitSay_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then
        optKendisi.SetFocus
    End If
End Sub

Private Sub optKendisi_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then
        txtDağıtımıcıNo.SetFocus
    End If
End Sub

Private Sub optKendisi_Click()
    cboGruplar.Enabled = False
    ComboBoxTemizle cboGruplar
    chkGrupBaşkanı.Enabled = False
    chkGrupBaşkanı.Value = Unchecked
End Sub

Private Sub optGrup_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then

```

```

        cboGruplar.SetFocus
    End If
End Sub

Private Sub optGrup_Click()
    cboGruplar.Enabled = True
    chkGrupBaşkanı.Enabled = True
End Sub

Private Sub cboGruplar_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then
        txtDağıtımçıNo.SetFocus
    End If
End Sub

Private Sub txtDağıtımçıNo_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then
        txtÜcret.SetFocus
    End If
End Sub

Private Sub optEvet_Click()
    cboHediyeKitap.Enabled = True
End Sub

Private Sub optHayır_Click()
    cboHediyeKitap.Enabled = False
    ComboBoxTemizle cboHediyeKitap
End Sub

Private Sub cboHediyeKitap_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then
        txtÜcret.SetFocus
    End If

```



End Sub

Private Sub txtÜcret\_Change()

    TextBoxFormatlıYazım txtÜcret, gtParaFormatıYaz

End Sub

Private Sub txtÜcret\_KeyDown(KeyCode As Integer, Shift As Integer)

    If KeyCode = vbKeyReturn Then

        cboParaBirimi1.SetFocus

    End If

End Sub

Private Sub cboParaBirimi2\_KeyDown(KeyCode As Integer, Shift As Integer)

    If KeyCode = vbKeyReturn Then

        sstBilgiler.Tab = 0

        txtAd.SetFocus

    End If

End Sub

Private Sub cboParaBirimi2\_Click()

    If gfNesneClickleriÇalışabilirMi And cmdÖncekiÖdemeleriSil.Enabled Then

        ÖncekiÖdemeler

    End If

End Sub

Private Sub cmdÖncekiÖdemeleriSil\_Click()

    ÖncekiÖdemeleriSil lParametreNoBul(gnBağlantı, cboParaBirimi2.Text,  
    "ParaBirimleri", "Kod", "Ad")

    ÖncekiÖdemeler

End Sub

Private Sub cmdÖdemeYap\_Click()

    Me.Enabled = False

    frmÖdemeYap.Show

```
ComboBoxKopyala cboParaBirimi2, frmÖdemeYap.cboParaBirimi,  
cboParaBirimi2.Text  
End Sub
```

```
*****  
*****  
*****  
*****  
*****  
*****
```

```
Private Sub Form_Load()  
    ToolTipTextleriAta  
    HücreBüyüklikleriniAyarla  
    ComboBoxlarıGüncelle  
    ListBoxaYatayKaydırmaÇubuğuEkle lstAboneİsimleri  
    sstBilgiler.Tab = 0  
    gtParaFormatıYaz = "###,###,###"  
    gtParaFormatıOku = "#"  
    gfNesneClickleriÇalışabilirMi = True  
    tAçılışSQL = "SELECT * FROM Aboneler WHERE İptalEdildiMi=False  
ORDER BY Ad"  
End Sub
```

```
Private Sub Form_Activate()  
    If gfNesneClickleriÇalışabilirMi Then  
        If gfYöneticiHesabıAktifMi Then  
            KayıtDüğmeleriniAç  
        Else  
            KayıtDüğmeleriniKapat  
        End If  
        ListBoxGüncelle lstAboneİsimleri, grYedekKayıt, tAçılışSQL, "Ad", "Soyad",  
"SıraNo"  
        If lstAboneİsimleri.ListCount = 0 Then
```

```

        'bunu yapmazsak txtad ta setfocus kalıyor
        lstAboneİsimleri.SetFocus
    End If
    DurumÇubuğunuGüncelle
    gfSormadanÇık = False
End If
gfNesneClickleriÇalışabilirMi = True
'program içinde diğer formları gösterip sonra tekrar ana forma dönünce onun
'activate olayı oluyo onu engelledik artık nesne clickleri çalışabilir
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
    If Not gfSormadanÇık Then
        gvCevap = vUyarı(Çıkış, vbQuestion + vbYesNo, "EXIT TO WINDOWS")
        If gvCevap = vbNo Then
            Cancel = -1
        Else
            ProgramdanÇık
        End If
    End If
End Sub

```

```

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    On Error Resume Next
    Dim lTMP1 As Long, lTMP2 As Long
    Select Case KeyCode
        Case vbKeyF1
            YardımıGöster
        Case vbKeyF2
            If tbrSolAraçÇubuğu.Buttons.Item(1).Enabled Then
                KayıtModunaGeç
            End If
        Case vbKeyF3
            If tbrSolAraçÇubuğu.Buttons.Item(3).Enabled Then

```



```

        KaydetGüncelle
    End If
Case vbKeyF4
    Me.Enabled = False
    frmAra.Show
Case vbKeyF5
    HücreİçerikleriniTemizle
Case vbKeyF6
    DEV.rsPostaEtiketi.Close
    DEV.rsPostaEtiketi.Open "SELECT A.AboneNo, A.Ad, A.Soyad,
A.Adres,İlçeler.Ad,İller.Ad,Ülkeler.Ad,A.BaşSay,A.BaşSay+11 FROM Aboneler
A, İlçeler,İller,Ülkeler WHERE A.SıraNo=" &
IstAboneİsimleri.ItemData(IstAboneİsimleri.ListIndex) & " AND
İller.Kod=A.İlKodu AND İlçeler.Kod=A.İlçeKodu AND
Ülkeler.Kod=A.ÜlkeKodu"
    DEV.rsPostaEtiketi.Requery
    PostaEtiketi.Show
Case vbKeyF7
    If fSQLÇalıştır(gnBağlantı, grKayıt, "SELECT * FROM Aboneler WHERE
İptalEdildiMi=True") Then
        IstAboneİsimleri.Clear
        If grKayıt.RecordCount > 0 Then
            ListBoxGüncelle IstAboneİsimleri, grKayıt, "SELECT * FROM
Aboneler WHERE İptalEdildiMi=True", "Ad", "Soyad", "SıraNo"
            DurumÇubuğunuGüncelle
        Else
            HücreİçerikleriniTemizle
            sstBilgiler.Tab = 0
            sstBilgiler.Enabled = False
            vUyarı KayıtBulunamadı, vbOKOnly + vbExclamation
            IstAboneİsimleri.SetFocus
        End If
    End If
Case vbKeyF8

```

```

    If fSQLÇalıştır(gnBağlantı, grKayıt, "SELECT * FROM Aboneler WHERE
SıraNo IN (SELECT AboneSıraNo FROM KişiselHesaplar WHERE
KalanBorç>0)") Then
        lstAboneİsimleri.Clear
        If grKayıt.RecordCount > 0 Then
            KendisiAboneOlanlarıYaz
        Else
            HücreİçerikleriniTemizle
            sstBilgiler.Tab = 0
            sstBilgiler.Enabled = False
            vUyarı KayıtBulunamadı, vbOKOnly + vbExclamation
            lstAboneİsimleri.SetFocus
        End If
    End If
Case vbKeyF9
    If tbrSağAraçÇubuğu.Buttons.Item(1).Enabled Then
        frmListeler.Show
        Me.Enabled = False
    End If
Case vbKeyF10
    If tbrSağAraçÇubuğu.Buttons.Item(3).Enabled Then
        Me.Enabled = False
        frmDağıtımListeleri.Show
    End If
Case vbKeyEscape
    Unload Me
End Select
End Sub

```

## 7.2 frmAra SOURCE CODES

Option Explicit

```
Public Sub VeriyiSQLeEkle(ptVeri As String, ByRef ptSQL As String,  
ptVeriAlanıAdı As String, ptKoşul As String, Optional ptKoşulSonlandırma As  
String = "")
```

```
    If ptVeri <> "" Then
```

```
        ptSQL = ptSQL & " AND " & ptVeriAlanıAdı & ptKoşul & ptVeri
```

```
        If ptKoşulSonlandırma <> "" Then
```

```
            ptSQL = ptSQL & ptKoşulSonlandırma
```

```
        End If
```

```
    End If
```

```
End Sub
```

```
Public Sub Ara()
```

```
    Dim tKökSQL As String, tSQL As String
```

```
    Dim lTMP1 As Long, lTMP2 As Long, lTMP3 As Long
```

```
    tKökSQL = "SELECT * FROM Aboneler WHERE İptalEdildiMi=False"
```

```
    tSQL = tKökSQL
```

```
    VeriyiSQLeEkle txtAd.Text, tSQL, "Ad", " LIKE '%', '%'"
```

```
    VeriyiSQLeEkle txtSoyad.Text, tSQL, "Soyad", " LIKE '%', '%'"
```

```
    VeriyiSQLeEkle rtfAdres.Text, tSQL, "Adres", " LIKE '%', '%'"
```

```
    lTMP1 = lParametreNoBul(gnBağlantı, cboÜlke.Text, "Ülkeler", "Kod", "Ad")
```

```
    lTMP2 = lParametreNoBul(gnBağlantı, cboİl.Text, "İller", "Kod", "Ad",
```

```
"ÜlkeKodu", lTMP1)
```

```
    lTMP3 = lParametreNoBul(gnBağlantı, cboİlçe.Text, "İlçeler", "Kod", "Ad",
```

```
"İlKodu", lTMP2)
```

```
    If lTMP1 > -1 Then
```

```
        VeriyiSQLeEkle Str(lTMP1), tSQL, "ÜlkeKodu", "="
```

```
    End If
```

```
    If lTMP2 > -1 Then
```

```
        VeriyiSQLeEkle Str(lTMP2), tSQL, "İlKodu", "="
```



```

End If
If ITMP3 > -1 Then
    VeriyiSQLeEkle Str(ITMP1), tSQL, "İlçeKodu", "="
End If
VeriyiSQLeEkle rtfAçıklama.Text, tSQL, "Açıklama", " LIKE '%', '%'"
VeriyiSQLeEkle txtAboneNo.Text, tSQL, "AboneNo", "=", ""
VeriyiSQLeEkle txtBaşSay.Text, tSQL, "BaşSay", "="
If dtpİlkAbonelikTarihi.Enabled Then
    VeriyiSQLeEkle Str(CLng(dtpİlkAbonelikTarihi.Value)), tSQL, "İlkKayıt",
"="
End If
If dtpEnSonYenilemeTarihi.Enabled Then
    VeriyiSQLeEkle Str(CLng(dtpEnSonYenilemeTarihi.Value)), tSQL,
"EnSonYenileme", "="
End If
If optKendisi.Value Then
    VeriyiSQLeEkle Str(KENDİSİABONENO), tSQL, "AboneYapanNo", "="
ElseIf optGrup.Value Then
    ITMP1 = IParametreNoBul(gnBağlantı, cboGruplar.Text, "Gruplar",
"GrupNo", "Ad")
    If ITMP1 > -1 Then
        VeriyiSQLeEkle Str(ITMP1), tSQL, "AboneYapanNo", "="
    End If
End If
VeriyiSQLeEkle txtDağıtımçıNo.Text, tSQL, "DağıtımçıNo", "="
ITMP1 = Val(Format(txtÜcret.Text, gtParaFormatıOku))
If ITMP1 > 0 Then
    VeriyiSQLeEkle Str(ITMP1), tSQL, "AbonelikÜcreti", "="
End If
ITMP1 = IParametreNoBul(gnBağlantı, cboParaBirimi.Text, "ParaBirimleri",
"Kod", "Ad")
If ITMP1 > -1 Then
    VeriyiSQLeEkle Str(ITMP1), tSQL, "ParaBirimiKodu", "="
End If

```

```

If optEvet.Value Then
    VeriyiSQLeEkle "True", tSQL, "HediyeKitapVerildiMi", "="
    ITMP1 = IParametreNoBul(gnBağlantı, cboHediyeKitap.Text,
"HediyeKitaplar", "Kod", "Ad")
    If ITMP1 > -1 Then
        VeriyiSQLeEkle Str(ITMP1), tSQL, "HediyeKitapKodu", "="
    End If
ElseIf optHayır.Value Then
    VeriyiSQLeEkle "False", tSQL, "HediyeKitapVerildiMi", "="
End If
tSQL = tSQL & " ORDER BY Ad,Soyad"
If fSQLÇalıştır(gnBağlantı, grYedekKayıt, tSQL) Then
    If grYedekKayıt.RecordCount = 0 Then
        vUyarı KayıtBulunamadı, vbOKOnly + vbExclamation
    Else
        tAçılışSQL = tSQL
        Unload Me
    End If
End If
End Sub

```

```

Private Sub optEvet_Click()
    cboHediyeKitap.Enabled = True
End Sub

```

```

Private Sub optHayır_Click()
    ComboBoxTemizle cboHediyeKitap
    cboHediyeKitap.Enabled = False
End Sub

```

```

Private Sub optFarkEtmez2_Click()
    optHayır_Click
End Sub

```

```
Private Sub optKendisi_Click()  
    ComboBoxTemizle.cboGruplar  
        cboGruplar.Enabled = False  
End Sub
```

```
Private Sub optGrup_Click()  
    cboGruplar.Enabled = True  
End Sub
```

```
Private Sub optFarkEtmez1_Click()  
    optKendisi_Click  
End Sub
```

```
Private Sub txtÜcret_Change()  
    TextBoxFormatlıYazım.txtÜcret, gtParaFormatıYaz  
End Sub
```

```
Private Sub cmdİATAçKapat_Click()  
    If dtpİlkAbonelikTarihi.Enabled Then  
        dtpİlkAbonelikTarihi.Enabled = False  
    Else  
        dtpİlkAbonelikTarihi.Enabled = True  
    End If  
End Sub
```

```
Private Sub cmdESYTAçKapat_Click()  
    If dtpEnSonYenilemeTarihi.Enabled Then  
        dtpEnSonYenilemeTarihi.Enabled = False  
    Else  
        dtpEnSonYenilemeTarihi.Enabled = True  
    End If  
End Sub
```

```
Private Sub cmdAra_Click()
```



```
Ara  
End Sub
```

```
Private Sub cmdGeri_Click()  
    Unload Me  
End Sub
```

```
Private Sub Form_Load()  
    dtpİlkAbonelikTarihi.Enabled = False  
    dtpEnSonYenilemeTarihi.Enabled = False  
    dtpİlkAbonelikTarihi.Value = Date  
    dtpEnSonYenilemeTarihi.Value = Date  
    optFarkEtmez1.Value = True  
    optFarkEtmez2.Value = True  
    ComboBoxKopyala frmAnaPencere.cboParaBirimi1, cboParaBirimi  
    ComboBoxKopyala frmAnaPencere.cboGruplar, cboGruplar  
    ComboBoxKopyala frmAnaPencere.cboHediyeKitap, cboHediyeKitap  
    ComboBoxTemizle cboGruplar  
    ComboBoxTemizle cboParaBirimi  
    ComboBoxTemizle cboHediyeKitap  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
    frmAnaPencere.Enabled = True  
End Sub
```

### 7.3 frmDağıtımListeleri SOURCE CODES

Option Explicit

Private Sub cmdGeri\_Click()

    Unload Me

End Sub

Private Sub cmdYazdır\_Click()

    On Error Resume Next

    If Val(txtBaşSay.Text) > 0 And Val(txtDağıtımçıNo) > 0 Then

        DEV.rsdevDağıtım.Close

        DEV.rsdevDağıtım.Open "SELECT AboneNo,Ad,Soyad,BaşSay,BaşSay+11  
FROM Aboneler WHERE BaşSay+11>=" & txtBaşSay.Text & " AND BaşSay<=" &  
txtBaşSay.Text & " AND İptalEdildiMi=False AND DağıtımçıNo=" &  
txtDağıtımçıNo.Text

        DEV.rsdevDağıtım.Requery

        DRPDağıtım.Show

    End If

End Sub

Private Sub Form\_Unload(Cancel As Integer)

    frmAnaPencere.Enabled = True

End Sub

## 7.4 frmKullanıcıHesapları SOURCE CODES

Option Explicit

Private Sub lstKullanıcılar\_Click()

If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT \* FROM KullanıcıHesapları  
WHERE Ad='" & lstKullanıcılar.Text & "'") Then

txtAd.Text = grYedekKayıt.Collect("Ad")

txtParola.Text = grYedekKayıt.Collect("Parola")

If grYedekKayıt.Collect("YöneticiMi") Then

cboHesapTipi.ListIndex = 0

Else

cboHesapTipi.ListIndex = 1

End If

End If

End Sub

Private Sub cmdGüncelle\_Click()

If txtParola.Text = "" Then

Hata GeçersizKullanıcıAdıYadaŞifre

Else

grYedekKayıt("Parola") = txtParola.Text

grYedekKayıt.Update

End If

End Sub

Private Sub cmdSil\_Click()

If lstKullanıcılar.ListCount > 1 Then

grYedekKayıt.Delete

ListBoxGüncelle lstKullanıcılar, grYedekKayıt, "SELECT \* FROM  
KullanıcıHesapları ORDER BY Ad", "Ad"

Else

Hata EnAzBirKullanıcıKayıtlıOlmalı



```
End If  
End Sub
```

```
Private Sub cmdYeniKullanıcı_Click()  
    Me.Enabled = False  
    frmYeniKullanıcıEkle.Show  
End Sub
```

```
Private Sub cmdGeri_Click()  
    Unload Me  
End Sub
```

```
Private Sub Form_Load()  
    ListBoxGüncelleIstKullanıcılar, grYedekKayıt, "SELECT * FROM  
    KullanıcıHesapları ORDER BY Ad", "Ad"  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
    frmAnaPencere.Enabled = True  
End Sub
```

## 7.5 frmListeler SOURCE CODES

Option Explicit

Private Sub cmdGeri\_Click()

    Unload Me

End Sub

Private Sub cmdYazdır\_Click()

    On Error Resume Next

    DEV.rsEnSonYeniKayıtlar.Requery

    DEV.rsEnSonYenilenenler.Requery

    DEV.rsEnSonAdSoyadDeğiştirenler.Requery

    DEV.rsEnSonİptalEdilenler.Requery

    DEV.rsESAD.Requery

    If optEnSonYeniKayıtlar.Value Then

        ESYK.Show

        If vUyarı(ListeSilinsinMi, vbYesNo + vbDefaultButton1) = vbYes Then

            If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "DELETE FROM  
EnSonYeniKayıtlar", False) Then

                End If

        End If

    ElseIf optEnSonYenilenenler.Value Then

        ESKY.Show

        If vUyarı(ListeSilinsinMi, vbYesNo + vbDefaultButton1) = vbYes Then

            If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "DELETE FROM  
EnSonYenilenenler", False) Then

                End If

        End If

    ElseIf optEnSonAdresDeğiştirenler.Value Then

        ESAD.Show

        If vUyarı(ListeSilinsinMi, vbYesNo + vbDefaultButton1) = vbYes Then

```

        If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "DELETE FROM
EnSonAdresiDeğiştirilenler", False) Then
            End If
        End If
        ElseIf optEnSonAdSoyadDeğiştirenler.Value Then
            ESSD.Show
            If vUyarı(ListeSilinsinMi, vbYesNo + vbDefaultButton1) = vbYes Then
                If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "DELETE FROM
EnSonAdSoyadDeğiştirilenler", False) Then
                    End If
                End If
            ElseIf optEnSonİptaller.Value Then
                ESİE.Show
                If vUyarı(ListeSilinsinMi, vbYesNo + vbDefaultButton1) = vbYes Then
                    If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "DELETE FROM
EnSonİptalEdilenler", False) Then
                        End If
                    End If
                End If
            End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
    frmAnaPencere.Enabled = True
End Sub

```



## 7.6 frmLogOn SOURCE CODES

Option Explicit

```
Private Sub HücreBüyüklikleriniAyarla()  
    txtKullanıcıAdı.MaxLength = 10  
    txtParola.MaxLength = 10  
End Sub
```

```
Private Sub ToolTipTextleriAta()  
    frmLogOn.txtKullanıcıAdı.ToolTipText = "Enter a user name."  
    frmLogOn.txtParola.ToolTipText = "Enter the password."  
    frmLogOn.cmdLogOn.ToolTipText = "Click here to log on."  
    frmLogOn.cmdÇıkış.ToolTipText = "Click here to exit."  
End Sub
```

```
Private Sub txtKullanıcıAdı_KeyDown(KeyCode As Integer, Shift As Integer)  
    If KeyCode = vbKeyReturn Then  
        If txtParola.Text <> "" Then  
            cmdLogOn_Click  
        Else  
            txtParola.SetFocus  
        End If  
    End If  
End Sub
```

```
Private Sub txtParola_KeyDown(KeyCode As Integer, Shift As Integer)  
    If KeyCode = vbKeyReturn Then  
        cmdLogOn_Click  
    End If  
End Sub
```

```
Private Sub cmdLogOn_Click()  
    If txtKullanıcıAdı.Text = "" Or txtParola.Text = "" Then  
        Hata GeçersizKullanıcıAdıYadaŞifre  
        txtKullanıcıAdı.SetFocus  
    ElseIf fSQLÇalıştır(gnBağlantı, grKayıt, "SELECT * FROM KullanıcıHesapları  
WHERE Ad='" & txtKullanıcıAdı.Text & "' AND Parola='" & txtParola.Text & "'")  
Then  
        If grKayıt.RecordCount > 0 Then  
            If grKayıt.Collect("YöneticiMi") = True Then  
                gfYöneticiHesabıAktifMi = True  
            Else  
                gfYöneticiHesabıAktifMi = False  
            End If  
            frmAnaPencere.Show  
            gfSormadanÇık = True  
            Unload Me  
        Else
```

```

        Hata GeçersizKullanıcıAdıYadaŞifre
        txtKullanıcıAdı.SetFocus
    End If
End If
End Sub

Private Sub cmdÇıkış_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    gtDosyaAdı = "Altinoluk.mdb"
    gtYedekDosyaAdı = "Altinoluk.bak"
    gtSql = "SELECT * FROM KullanıcıHesapları"
    tYardımDosyası = "Readme.txt"
    gfSormadanÇık = False
    If fBaşkaÇalışanVarMı Then
        Hata BirdenFazlaProgramÇalıştırma
        gfSormadanÇık = True
        Unload Me
    End If
    If Not fVeriTabanıAç(gtDosyaAdı, gnBağlantı, grKayıt, gtSql) Then
        Hata VeriTabanıAçılamadı
        gfSormadanÇık = True
        Unload Me
    End If
End Sub

Private Sub Form_Activate()
    ToolTipTextleriAta
    HücreBüyüklikleriniAyarla
    txtKullanıcıAdı.SetFocus
End Sub

Private Sub Form_Unload(Cancel As Integer)
    If Not gfSormadanÇık Then
        If Not fVeriTabanıKapat(gnBağlantı, grKayıt) Then
            Hata VeriTabanıKapatılamadı
        End If
    End If
End Sub

```

## 7.7 frmÖdemeYap SOURCE CODES

Option Explicit

```
Private Sub txtMiktar_Change()
```

```
    TextBoxFormatlıYazım txtMiktar, gtParaFormatıYaz
```

```
End Sub
```

```
Private Sub txtMiktar_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
    If KeyCode = vbKeyReturn Then
```

```
        cboParabirimi.SetFocus
```

```
    End If
```

```
End Sub
```

```
Private Sub cboParabirimi_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
    If KeyCode = vbKeyReturn Then
```

```
        dtpTarih.SetFocus
```

```
    End If
```

```
End Sub
```

```
Private Sub dtpTarih_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
    If KeyCode = vbKeyReturn Then
```

```
        rtfAçıklama.SetFocus
```

```
    End If
```

```
End Sub
```

```
Private Sub cmdİptal_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub cmdTamam_Click()
```

```
    Dim tTMP As String, tSQL1 As String
```

```
    Dim rTMP As Recordset
```

```
    tTMP = Format(txtMiktar.Text, gtParaFormatıOku)
```

```
    If Val(tTMP) > 0 Then
```

```
        If grKayıt.Collect("AboneYapanNo") = KENDİSİABONENO Then
```

```
            tSQL1 = "SELECT * FROM KişiselHesaplar WHERE AboneSıraNo=" &
```

```
grKayıt.Collect("SıraNo")
```

```
        Else
```

```
            tSQL1 = "SELECT * FROM GrupHesapları WHERE GrupNo=" &
```

```
grKayıt.Collect("AboneYapanNo") & " AND ParaBirimiKodu=(SELECT Kod FROM  
ParaBirimleri WHERE Ad=" & cboParabirimi.Text & " ")"
```

```
        End If
```

```
        If fSQLÇalıştır(gnBağlantı, grYedekKayıt, tSQL1) Then
```

```
            If Val(tTMP) > grYedekKayıt("KalanBorç") Then
```

```
                Hata LütfenGeçerliBirMiktarGirin
```

```
                txtMiktar.SetFocus
```

```
            Else
```

```
                If fSQLÇalıştır(gnBağlantı, rTMP, "SELECT * FROM Ödemeler") Then
```

```

rTMP.AddNew
rTMP("SıraNo") = grKayıt.Collect("SıraNo")
rTMP("Miktar") = Val(Format(txtMiktar.Text, gtParaFormatıOku))
rTMP("Tarih") = CLng(dtpTarih.Value)
rTMP("Açıklama") = rtfAçıklama.Text
grYedekKayıt("KalanBorç") = grYedekKayıt("KalanBorç") - Val(rTMP)
If grKayıt.Collect("AboneYapanNo") = KENDİSİABONENO Then
    rTMP("ParaBirimiKodu") = grKayıt.Collect("ParaBirimiKodu")
Else
    rTMP("ParaBirimiKodu") = grYedekKayıt.Collect("ParaBirimiKodu")
End If
rTMP.Update
grYedekKayıt.Update
ÖncekiÖdemeler
Unload Me
Else
    Hata ÖdemeKayıtEdilemedi
End If
End If
Else
    Hata ÖdemeKayıtEdilemedi
End If
Else
    Hata LütfenGeçerliBirMiktarGirin
End If
End Sub

Private Sub Form_Load()
    dtpTarih.Value = Date
End Sub

Private Sub Form_Unload(Cancel As Integer)
    gfNesneClickleriÇalışabilirMi = False 'ana formun activate i çalışmasın diye
    frmAnaPencere.Enabled = True
End Sub

```



## 7.8 frmParametreGirişi SOURCE CODES

```
Private tTablo As String
Private lİl As Long
Private lÜlke As Long
```

```
Private Sub cboParametreler_Click()
    txtDüzeltilme.Text = ""
    cboDeğerler.Clear
    cboİller.Clear
    cboÜlkeler.Clear
    cboİller.Enabled = False
    cboÜlkeler.Enabled = False
    Select Case cboParametreler.ListIndex
        Case 0
            tTablo = "Ülkeler"
            ComboBoxGüncelle cboDeğerler, gnBağlantı, grYedekKayıt, "SELECT *
FROM Ülkeler ORDER BY Ad", "Ad", False
        Case 1
            tTablo = "İller"
            cboÜlkeler.Enabled = True
            ComboBoxGüncelle cboÜlkeler, gnBağlantı, grYedekKayıt, "SELECT * FROM
Ülkeler ORDER BY Ad", "Ad", True
        Case 2
            tTablo = "İlçeler"
            cboÜlkeler.Enabled = True
            cboİller.Enabled = True
            ComboBoxGüncelle cboÜlkeler, gnBağlantı, grYedekKayıt, "SELECT * FROM
Ülkeler ORDER BY Ad", "Ad", True
        Case 3
            tTablo = "Gruplar"
            ComboBoxGüncelle cboDeğerler, gnBağlantı, grYedekKayıt, "SELECT *
FROM Gruplar ORDER BY Ad", "Ad", False
        Case 4
            tTablo = "HediyeKitaplar"
            ComboBoxGüncelle cboDeğerler, gnBağlantı, grYedekKayıt, "SELECT *
FROM HediyeKitaplar ORDER BY Ad", "Ad", False
        Case 5
            tTablo = "ParaBirimleri"
            ComboBoxGüncelle cboDeğerler, gnBağlantı, grYedekKayıt, "SELECT *
FROM ParaBirimleri ORDER BY Ad", "Ad", False
    End Select
End Sub
```

```
Private Sub cboÜlkeler_Click()
    lÜlke = lParametreNoBul(gnBağlantı, cboÜlkeler.Text, "Ülkeler", "Kod", "Ad")
    If cboİller.Enabled Then
        ComboBoxGüncelle cboİller, gnBağlantı, grYedekKayıt, "SELECT * FROM İller
WHERE ÜlkeKodu=" & lÜlke & " ORDER BY Ad", "Ad", True
    End If
End Sub
```

```

Else
    ComboBoxGüncelle cboDeğerler, gnBağlantı, grYedekKayıt, "SELECT * FROM
    İller WHERE ÜlkeKodu=" & lÜlke & " ORDER BY Ad", "Ad", False
End If
End Sub

Private Sub cboİller_Click()
    İl = lParametreNoBul(gnBağlantı, cboİller.Text, "İller", "Kod", "Ad")
    ComboBoxGüncelle cboDeğerler, gnBağlantı, grYedekKayıt, "SELECT * FROM
    İlçeler WHERE İlKodu=" & İl & " ORDER BY Ad", "Ad", False
End Sub

Private Sub cmdSakla_Click()
    If cboDeğerler.Text = "" Then
        Hata BirParametreDeğeriGirilmeli
        Exit Sub
    End If
    Select Case tTablo
        Case "Ülkeler"
            If cboDeğerler.Text = "" Then
                Hata BirParametreDeğeriGirilmeli
                Exit Sub
            End If
            If fDahaÖncedenKayıtlıMı(gnBağlantı, grYedekKayıt, "SELECT * FROM
            Ülkeler WHERE Ad=" & cboDeğerler.Text & "") Then
                If txtDüzeltilme.Text <> "" Then
                    grYedekKayıt("Ad") = txtDüzeltilme.Text
                    grYedekKayıt.Update
                    cboParametreler_Click
                Else
                    Hata BirParametreDeğeriGirilmeli
                End If
            Else
                grYedekKayıt.AddNew
                grYedekKayıt("Ad") = cboDeğerler.Text
                grYedekKayıt.Update
                cboParametreler_Click
            End If
        Case "İller"
            If cboÜlkeler.Text = "" Then
                Hata BirParametreDeğeriGirilmeli
                Exit Sub
            End If
            If fDahaÖncedenKayıtlıMı(gnBağlantı, grYedekKayıt, "SELECT * FROM İller
            WHERE ÜlkeKodu=" & lÜlke & " AND Ad=" & cboDeğerler.Text & "") Then
                If txtDüzeltilme.Text <> "" Then
                    grYedekKayıt("Ad") = txtDüzeltilme.Text
                    grYedekKayıt.Update
                    cboParametreler_Click
                Else

```

```

        Hata BirParametreDeğeriGirilmeli
    End If
Else
    grYedekKayıt.AddNew
    grYedekKayıt("Ad") = cboDeğerler.Text
    grYedekKayıt("ÜlkeKodu") = lÜlke
    grYedekKayıt.Update
    cboParametreler_Click
End If
Case "İlçeler"
    If cboÜlkeler.Text = "" Or cboİller.Text = "" Then
        Hata BirParametreDeğeriGirilmeli
        Exit Sub
    End If
    If fDahaÖncedenKayıtlıMı(gnBağlantı, grYedekKayıt, "SELECT * FROM
İlçeler WHERE İlKodu=" & lİl & " AND Ad=" & cboDeğerler.Text & "") Then
        grYedekKayıt("Ad") = txtDüzeltilme.Text
        grYedekKayıt.Update
        cboParametreler_Click
    Else
        grYedekKayıt.AddNew
        grYedekKayıt("Ad") = cboDeğerler.Text
        grYedekKayıt("İlKodu") = lİl
        grYedekKayıt.Update
        cboParametreler_Click
    End If
Case "Gruplar"
    If cboDeğerler.Text = "" Then
        Hata BirParametreDeğeriGirilmeli
        Exit Sub
    End If
    If fDahaÖncedenKayıtlıMı(gnBağlantı, grYedekKayıt, "SELECT * FROM
Gruplar WHERE Ad=" & cboDeğerler.Text & "") Then
        If txtDüzeltilme.Text <> "" Then
            grYedekKayıt("Ad") = txtDüzeltilme.Text
            grYedekKayıt.Update
            cboParametreler_Click
        Else
            Hata BirParametreDeğeriGirilmeli
        End If
    Else
        grYedekKayıt.AddNew
        grYedekKayıt("Ad") = cboDeğerler.Text
        grYedekKayıt("Başkan") = 0
        grYedekKayıt.Update
        cboParametreler_Click
    End If
Case "HediyeKitaplar"
    If cboDeğerler.Text = "" Then
        Hata BirParametreDeğeriGirilmeli

```

```

Exit Sub
End If
If fDahaÖncedenKayıtlıMı(gnBağlantı, grYedekKayıt, "SELECT * FROM
HediyeKitaplar WHERE Ad="" & cboDeğerler.Text & """) Then
    If txtDüzeltilme.Text <> "" Then
        grYedekKayıt("Ad") = txtDüzeltilme.Text
        grYedekKayıt.Update
        cboParametreler_Click
    Else
        Hata BirParametreDeğeriGirilmeli
    End If
Else
    grYedekKayıt.AddNew
    grYedekKayıt("Ad") = cboDeğerler.Text
    grYedekKayıt.Update
    cboParametreler_Click
End If
Case "ParaBirimleri"
    If cboDeğerler.Text = "" Then
        Hata BirParametreDeğeriGirilmeli
        Exit Sub
    End If
    If fDahaÖncedenKayıtlıMı(gnBağlantı, grYedekKayıt, "SELECT * FROM
ParaBirimleri WHERE Ad="" & cboDeğerler.Text & """) Then
        If txtDüzeltilme.Text <> "" Then
            grYedekKayıt("Ad") = txtDüzeltilme.Text
            grYedekKayıt.Update
            cboParametreler_Click
        Else
            Hata BirParametreDeğeriGirilmeli
        End If
    Else
        grYedekKayıt.AddNew
        grYedekKayıt("Ad") = cboDeğerler.Text
        grYedekKayıt.Update
        cboParametreler_Click
    End If
End Select
End Sub

Private Sub cmdSil_Click()
    Dim rTMP As Recordset
    Select Case tTablo
        Case "Ülkeler"
            If cboDeğerler.Text = "" Then
                Hata BirParametreDeğeriGirilmeli
                Exit Sub
            End If
            If fDahaÖncedenKayıtlıMı(gnBağlantı, grYedekKayıt, "SELECT * FROM
Ülkeler WHERE Ad="" & cboDeğerler.Text & """) Then

```



```

        If fDahaÖncedenKayıtlıMı(gnBağlantı, rTMP, "SELECT * FROM Aboneler
WHERE ÜlkeKodu=" & grYedekKayıt.Collect("Kod")) Then
            Hata BuParametreyiKullananKayıtlarVar
        Else
            grYedekKayıt.Delete
            cboParametreler_Click
        End If
    End If
    Case "İller"
        If cboÜlkeler.Text = "" Then
            Hata BirParametreDeğeriGirilmeli
            Exit Sub
        End If
        If fDahaÖncedenKayıtlıMı(gnBağlantı, grYedekKayıt, "SELECT * FROM İller
WHERE ÜlkeKodu=" & IÜlke & " AND Ad=" & cboDeğerler.Text & "") Then
            If fDahaÖncedenKayıtlıMı(gnBağlantı, rTMP, "SELECT * FROM Aboneler
WHERE İlKodu=" & grYedekKayıt.Collect("Kod")) Then
                Hata BuParametreyiKullananKayıtlarVar
            Else
                grYedekKayıt.Delete
                cboParametreler_Click
            End If
        End If
        Case "İlçeler"
            If cboÜlkeler.Text = "" Or cboİller.Text = "" Then
                Hata BirParametreDeğeriGirilmeli
                Exit Sub
            End If
            If fDahaÖncedenKayıtlıMı(gnBağlantı, grYedekKayıt, "SELECT * FROM
İlçeler WHERE İlKodu=" & Iİl & " AND Ad=" & cboDeğerler.Text & "") Then
                If fDahaÖncedenKayıtlıMı(gnBağlantı, rTMP, "SELECT * FROM Aboneler
WHERE İlçeKodu=" & grYedekKayıt.Collect("Kod")) Then
                    Hata BuParametreyiKullananKayıtlarVar
                Else
                    grYedekKayıt.Delete
                    cboParametreler_Click
                End If
            End If
            Case "Gruplar"
                If cboDeğerler.Text = "" Then
                    Hata BirParametreDeğeriGirilmeli
                    Exit Sub
                End If
                If fDahaÖncedenKayıtlıMı(gnBağlantı, grYedekKayıt, "SELECT * FROM
Gruplar WHERE Ad=" & cboDeğerler.Text & "") Then
                    If fDahaÖncedenKayıtlıMı(gnBağlantı, rTMP, "SELECT * FROM Aboneler
WHERE AboneYapanNo=" & grYedekKayıt.Collect("GrupNo")) Then
                        Hata BuParametreyiKullananKayıtlarVar
                    Else
                        grYedekKayıt.Delete

```



```
cboParametreler_Click
End If
End If
Case "HediyeKitaplar"
If cboDeğerler.Text = "" Then
Hata BirParametreDeğeriGirilmeli
Exit Sub
End If
If fDahaÖncedenKayıtlıMı(gnBağlantı, grYedekKayıt, "SELECT * FROM
HediyeKitaplar WHERE Ad=" & cboDeğerler.Text & """) Then
If fDahaÖncedenKayıtlıMı(gnBağlantı, rTMP, "SELECT * FROM Aboneler
WHERE HediyeKitapKodu=" & grYedekKayıt.Collect("Kod")) Then
Hata BuParametreyiKullananKayıtlarVar
Else
grYedekKayıt.Delete
cboParametreler_Click
End If
End If
Case "ParaBirimleri"
If cboDeğerler.Text = "" Then
Hata BirParametreDeğeriGirilmeli
Exit Sub
End If
If fDahaÖncedenKayıtlıMı(gnBağlantı, grYedekKayıt, "SELECT * FROM
ParaBirimleri WHERE Ad=" & cboDeğerler.Text & """) Then
If fDahaÖncedenKayıtlıMı(gnBağlantı, rTMP, "SELECT * FROM Aboneler
WHERE ParaBirimiKodu=" & grYedekKayıt.Collect("Kod")) Then
Hata BuParametreyiKullananKayıtlarVar
Else
grYedekKayıt.Delete
cboParametreler_Click
End If
End If
End Select
End Sub

Private Sub Form_Load()
cboParametreler.ListIndex = 0
End Sub

Private Sub Form_Unload(Cancel As Integer)
frmAnaPencere.Enabled = True
ComboBoxlarıGüncelle
AboneBilgileriniGöster
End Sub
```

## 7.9 frmYeniKullanıcıEkle SOURCE CODES

Option Explicit

```
Private Sub txtAd_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then
        txtParola.SetFocus
    End If
End Sub
```

```
Private Sub txtParola_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyReturn Then
        cmdKaydet.SetFocus
    End If
End Sub
```

```
Private Sub cmdGeri_Click()
    Unload Me
End Sub
```

```
Private Sub cmdKaydet_Click()
    Dim rTMP As Recordset
    If txtAd.Text = "" Or txtParola.Text = "" Or cboHesapTipi.Text = "" Then
        Hata GeçersizKullanıcıAdıYadaŞifre
    Else
        If fdahaÖncedenKayıtlıMı(gnBağlantı, rTMP, "SELECT * FROM
        KullanıcıHesapları WHERE Ad='" & txtAd.Text & "'") Then
            Hata BulsimdeBirKullanıcıZatenVar
        ElseIf fSQLÇalıştır(gnBağlantı, rTMP, "SELECT * FROM KullanıcıHesapları")
        Then
            rTMP.AddNew
            rTMP("Ad") = txtAd.Text
            rTMP("Parola") = txtParola.Text
            If cboHesapTipi.ListIndex = 0 Then
                rTMP("YöneticiMi") = True
            Else
                rTMP("YöneticiMi") = False
            End If
            rTMP.Update
            ListBoxGüncelle frmKullanıcıHesapları.lstKullanıcılar, grYedekKayıt,
            "SELECT * FROM KullanıcıHesapları ORDER BY Ad", "Ad"
            Unload Me
        End If
    End If
End Sub
```

```
Private Sub Form_Load()
    cboHesapTipi.ListIndex = 0
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    frmKullanıcıHesapları.Enabled = True
End Sub
```



Also there are 4 modules in this program. This modules have some functions and variables that are used in the program.

1. **AnaPencereModülü** = This module has the variables and functions that are used in frmAnaPencere form.
2. **AraçlarModülü** = This module has a lot of functions and variables that are used in all over the program.
3. **HataVeUyarıModülü** = This module is only used for error and warning messages in the program.
4. **VeriTabanıModülü** = This module has a lot of functions and variables related with database communications.

Source codes of these modules are as shown following pages.

## 7.10 AnaPencereModülü SOURCE CODES

Option Explicit

Enum Modlar

KayıtModu

DüzeltilmeModu

End Enum

Public geGeçerliMod As Modlar

Public Const KENDİSİABONENO = -1

Public tAçılışSQL As String

Public Sub KayıtDüğmeleriniKapat()

With frmAnaPencere

.tbrSolAraçÇubuğu.Buttons.Item(1).Enabled = False

.tbrSolAraçÇubuğu.Buttons.Item(3).Enabled = False

.tbrSolAraçÇubuğu.Buttons.Item(5).Enabled = False

.tbrSağAraçÇubuğu.Buttons.Item(1).Enabled = False

.tbrSağAraçÇubuğu.Buttons.Item(3).Enabled = False

.tbrSağAraçÇubuğu.Buttons.Item(5).Enabled = False

.tbrSağAraçÇubuğu.Buttons.Item(7).Enabled = False

.cmdÖncekiÖdemeleriSil.Enabled = False

.cmdÖdemeYap.Enabled = False

End With

End Sub

Public Sub KayıtDüğmeleriniAç()

With frmAnaPencere

.tbrSolAraçÇubuğu.Buttons.Item(1).Enabled = True

.tbrSolAraçÇubuğu.Buttons.Item(3).Enabled = True

.tbrSolAraçÇubuğu.Buttons.Item(5).Enabled = True

.tbrSağAraçÇubuğu.Buttons.Item(1).Enabled = True

```

        .tbrSağAraçÇubuğu.Buttons.Item(3).Enabled = True
        .tbrSağAraçÇubuğu.Buttons.Item(5).Enabled = True
        .tbrSağAraçÇubuğu.Buttons.Item(7).Enabled = True
        .cmdÖncekiÖdemeleriSil.Enabled = True
        .cmdÖdemeYap.Enabled = True
    End With
End Sub

```

```

Public Sub OnBirSayıSonrasınıHesapla(ptBaşSay As TextBox, ByRef ptBitSay As
TextBox)
    Dim lTMP As Long
    ptBitSay.Text = ""
    If IsNumeric(ptBaşSay.Text) Then
        lTMP = Val(ptBaşSay.Text)
        If lTMP > 0 Then
            ptBitSay.Text = lTMP + 11
        End If
    End If
End Sub

```

```

Public Sub HücreBüyüklikleriniAyarla()
    With frmAnaPencere
        .txtAd.MaxLength = 50
        .txtSoyad.MaxLength = 50
        .txtEvTel.MaxLength = 20
        .txtİşTel.MaxLength = 20
        .txtCepTel.MaxLength = 20
        .txtFax.MaxLength = 20
    End With
End Sub

```

```

Public Sub YeniKayıtİçinNesneleriKapat()
    With frmAnaPencere
        .dtpEnSonYenilemeTarihi.Enabled = False
    End With
End Sub

```

.sstBilgiler.TabEnabled(2) = False

End With

End Sub

Public Sub ToolTipTextleriAta()

With frmAnaPencere

.lstAboneİsimleri.ToolTipText = "Names of the found subscriptions."

.dtpEnSonYenilemeTarihi.ToolTipText = "Open if this rec. had been updated"

.cboGruplar.ToolTipText = "Group names."

.cboParaBirim1.ToolTipText = "Money units."

.sbrDurumÇubuğu.Panels(1).ToolTipText = "# of total records."

.sbrDurumÇubuğu.Panels(2).ToolTipText = "# of found records."

.sbrDurumÇubuğu.Panels(3).ToolTipText = "Date."

.sbrDurumÇubuğu.Panels(4).ToolTipText = "Time."

.tbrSolAraçÇubuğu.Buttons.Item(1).ToolTipText = "Enter a new record. [F2]"

.tbrSolAraçÇubuğu.Buttons.Item(3).ToolTipText = "Save a record or update an  
existing one. [F3]"

.tbrSolAraçÇubuğu.Buttons.Item(5).ToolTipText = "Cancel the selected record.  
[DELETE]"

.tbrSolAraçÇubuğu.Buttons.Item(7).ToolTipText = "Print-out a post sticker. [F6]"

.tbrSolAraçÇubuğu.Buttons.Item(9).ToolTipText = "Clear the cells. [F5]"

.tbrSolAraçÇubuğu.Buttons.Item(11).ToolTipText = "Search a record. [F4]"

.tbrSolAraçÇubuğu.Buttons.Item(13).ToolTipText = "List the canceled records.  
[F7]"

.tbrSolAraçÇubuğu.Buttons.Item(15).ToolTipText = "List the records that have  
dept. [F8]"

.tbrSağAraçÇubuğu.Buttons.Item(1).ToolTipText = "Show the lists window. [F9]"

.tbrSağAraçÇubuğu.Buttons.Item(3).ToolTipText = "Show the distributor lists  
window. [F10]"

.tbrSağAraçÇubuğu.Buttons.Item(5).ToolTipText = "Show the parameters  
window."

.tbrSağAraçÇubuğu.Buttons.Item(7).ToolTipText = "Show the user accounts  
window."

.tbrSağAraçÇubuğu.Buttons.Item(9).ToolTipText = "Show the about window."



```

        .tbrSağAraçÇubuğu.Buttons.Item(11).ToolTipText = "Open the help file."
        .tbrSağAraçÇubuğu.Buttons.Item(13).ToolTipText = "Log off."
        .tbrSağAraçÇubuğu.Buttons.Item(15).ToolTipText = "Exit."
    End With
End Sub

```

```

Public Sub HücresİçerikleriniTemizle()

```

```

    With frmAnaPencere

```

```

        .txtAd.Text = ""
        .txtSoyad.Text = ""
        .rtfAdres.Text = ""
        ComboBoxTemizle .cboÜlke
        ComboBoxTemizle .cboİl
        ComboBoxTemizle .cboİlçe
        .txtEvTel.Text = ""
        .txtİşTel.Text = ""
        .txtCepTel.Text = ""
        .txtFax.Text = ""
        .rtfAçıklama.Text = ""
        .txtAboneNo.Text = ""
        .txtBaşSay.Text = ""
        .txtBitSay.Text = ""
        .dtpİlkAbonelikTarihi.Value = Date
        .dtpEnSonYenilemeTarihi.Value = Date
        .dtpEnSonYenilemeTarihi.Enabled = False
        .optKendisi.Value = True
        ComboBoxTemizle .cboGruplar
        .chkGrupBaşkanı.Value = Unchecked
        .txtDağıtımıcıNo.Text = ""
        .optHayır.Value = True
        ComboBoxTemizle .cboHediyeKitap
        .txtÜcret.Text = ""
        ComboBoxTemizle .cboParaBirimi1

```

```

    End With

```

End Sub

Public Sub KayıtModunaGeç()

geGeçerliMod = KayıtModu

gfNesneClickleriÇalışabilirMi = False 'alttaki yordam da combo ların listindex i  
değişecek o yüzden

HücreİçerikleriniTemizle

gfNesneClickleriÇalışabilirMi = True

YeniKayıtİçinNesneleriKapat

frmAnaPencere.sstBilgiler.Tab = 0

frmAnaPencere.sstBilgiler.Enabled = True

frmAnaPencere.txtAd.SetFocus

End Sub

Public Sub ComboBoxlarıGüncelle()

ComboBoxGüncelle frmAnaPencere.cboÜlke, gnBağlantı, grYedekKayıt, "SELECT  
\* FROM Ülkeler ORDER BY Ad", "Ad", True

ComboBoxGüncelle frmAnaPencere.cboGruplar, gnBağlantı, grYedekKayıt,  
"SELECT \* FROM Gruplar ORDER BY Ad", "Ad", False

ComboBoxGüncelle frmAnaPencere.cboHediyeKitap, gnBağlantı, grYedekKayıt,  
"SELECT \* FROM HediyeKitaplar ORDER BY Ad", "Ad", False

ComboBoxGüncelle frmAnaPencere.cboParaBirimi1, gnBağlantı, grYedekKayıt,  
"SELECT \* FROM ParaBirimleri ORDER BY Ad", "Ad", False

End Sub

Public Sub DurumÇubuğunuGüncelle()

If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT \* FROM Aboneler WHERE  
İptalEdildiMi=False") Then

frmAnaPencere.sbrDurumÇubuğu.Panels(1).Text = grYedekKayıt.RecordCount

frmAnaPencere.sbrDurumÇubuğu.Panels(2).Text =

frmAnaPencere.lstAboneİsimleri.ListCount

If frmAnaPencere.lstAboneİsimleri.ListCount = 0 Then

frmAnaPencere.sstBilgiler.Enabled = False

Else

```

        frmAnaPencere.sstBilgiler.Enabled = True
    End If
End If
End Sub

Public Function fGerekliBilgilerGirildiMi() As Boolean
    With frmAnaPencere
        fGerekliBilgilerGirildiMi = .txtAd.Text <> "" And .txtSoyad.Text <> "" And
        .rtfAdres.Text <> "" And .cboÜlke.Text <> "" And .cboİl.Text <> "" And .cboİlçe.Text
        <> "" And Val(.txtBaşSay.Text) > 0 And (.optKendisi.Value = True Or (.optGrup.Value
        = True And .cboGruplar.Text <> "")) And Val(.txtDağıtımıcıNo.Text) > 0 And
        (.optHayır.Value = True Or (.optEvet.Value = True And .cboHediyeKitap.Text <> ""))
        And Val(Format(.txtÜcret.Text, gtParaFormatıOku)) > 0 And .cboParaBirim1.Text <>
        ""
    End With
End Function

```

```

Private Sub BilgileriYazOrtak()
    With frmAnaPencere
        grKayıt("EvTel") = .txtEvTel.Text
        grKayıt("İşTel") = .txtİşTel.Text
        grKayıt("CepTel") = .txtCepTel.Text
        grKayıt("Fax") = .txtFax.Text
        grKayıt("Açıklama") = .rtfAçıklama.Text
        grKayıt("AboneNo") = .txtAboneNo.Text
        If .optEvet.Value = True Then
            grKayıt("HediyeKitapVerildiMi") = True
            grKayıt("HediyeKitapKodu") = lParametreKaydetGüncelle(gnBağlantı,
            grYedekKayıt, "HediyeKitaplar", "Kod", "", "Ad", .cboHediyeKitap.Text, False)
        ElseIf .optHayır.Value = True Then
            grKayıt("HediyeKitapVerildiMi") = False
        End If
        grKayıt("DağıtımıcıNo") = .txtDağıtımıcıNo.Text
    End With

```

End Sub

Private Sub BilgileriYazAdSoyad()

With frmAnaPencere

grKayıt("Ad") = .txtAd.Text

grKayıt("Soyad") = .txtSoyad.Text

End With

End Sub

Private Sub EnSonAdSoyadDeğiştirilenlereKayıtYap()

If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT \* FROM  
EnSonAdSoyadDeğiştirilenler WHERE SıraNo=" & grKayıt.Collect("SıraNo")) Then

If grYedekKayıt.RecordCount = 0 Then

grYedekKayıt.AddNew

grYedekKayıt("SıraNo") = grKayıt.Collect("SıraNo")

grYedekKayıt.Update

End If

Else

Hata EnSonAdSoyadDeğiştirilenlereKayıtYapılamadı

End If

End Sub

Public Sub AdSoyadDeğişikliğiniKontrolEt()

With frmAnaPencere

If (grKayıt.Collect("Ad") <> .txtAd.Text) Or (grKayıt.Collect("Soyad") <>  
.txtSoyad.Text) Then

BilgileriYazAdSoyad

EnSonAdSoyadDeğiştirilenlereKayıtYap

.lstAboneİsimleri.List(.lstAboneİsimleri.ListIndex) = .txtAd.Text & " " &  
.txtSoyad.Text

grKayıt("EnSonİsimDeğişikliği") = CLng(Date)

grKayıt.Update

End If

End With



End Sub

Private Sub BilgileriYazAdres()

With frmAnaPencere

grKayıt("Adres") = .rtfAdres.Text

grKayıt("ÜlkeKodu") = IParametreKaydetGüncelle(gnBağlantı, grYedekKayıt, "Ülkeler", "Kod", "", "Ad", .cboÜlke.Text, False)

grKayıt("İlKodu") = IParametreKaydetGüncelle(gnBağlantı, grYedekKayıt, "İller", "Kod", "ÜlkeKodu", "Ad", .cboİl.Text, False, grKayıt.Collect("ÜlkeKodu"))

grKayıt("İlçeKodu") = IParametreKaydetGüncelle(gnBağlantı, grYedekKayıt, "İlçeler", "Kod", "İlKodu", "Ad", .cboİlçe.Text, False, grKayıt.Collect("İlKodu"))

End With

End Sub

Private Sub EnSonAdresDeğiştirenlereKayıtYap()

If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT \* FROM EnSonAdresiDeğiştirilenler WHERE SıraNo=" & grKayıt.Collect("SıraNo")) Then

If grYedekKayıt.RecordCount = 0 Then

grYedekKayıt.AddNew

grYedekKayıt("SıraNo") = grKayıt.Collect("SıraNo")

grYedekKayıt.Update

End If

Else

Hata EnSonAdresDeğiştirenlereKayıtYapılamadı

End If

End Sub

Public Sub AdresDeğişikliğiniKontrolEt()

Dim ITMP1 As Long, ITMP2 As Long, ITMP3 As Long

With frmAnaPencere

ITMP1 = IParametreNoBul(gnBağlantı, .cboÜlke.Text, "Ülkeler", "Kod", "Ad")

ITMP2 = IParametreNoBul(gnBağlantı, .cboİl.Text, "İller", "Kod", "Ad", "ÜlkeKodu", ITMP1)

```

ITMP3 = IParametreNoBul(gnBağlantı, .cboİlçe.Text, "İlçeler", "Kod", "Ad",
"İlKodu", ITMP2)
If (grKayıt.Collect("Adres") <> .rtfAdres.Text) Or (grKayıt.Collect("ÜlkeKodu")
<> ITMP1) Or (grKayıt.Collect("İlKodu") <> ITMP2) Or (grKayıt.Collect("İlçeKodu")
<> ITMP3) Then
    BilgileriYazAdres
    EnSonAdresDeğiştirenlereKayıtYap
    grKayıt("EnSonAdresDeğişikliği") = CLng(Date)
    grKayıt.Update
End If
End With
End Sub

```

```

Private Sub BilgileriYazYenileme()

```

```

    With frmAnaPencere
        grKayıt("BaşSay") = .txtBaşSay.Text
        grKayıt("AbonelikÜcreti") = Format(.txtÜcret.Text, gtParaFormatıOku)
        grKayıt("ParaBirimiKodu") = IParametreKaydetGüncelle(gnBağlantı,
grYedekKayıt, "ParaBirimleri", "Kod", "", "Ad", .cboParaBirimi1.Text, False)
    End With
End Sub

```

```

Private Sub EnSonYenilenenlereKayıtYap()

```

```

    If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT * FROM EnSonYenilenenler
WHERE SıraNo=" & grKayıt.Collect("SıraNo")) Then
        If grYedekKayıt.RecordCount = 0 Then
            grYedekKayıt.AddNew
            grYedekKayıt("SıraNo") = grKayıt.Collect("SıraNo")
            grYedekKayıt.Update
        End If
    Else
        Hata EnSonYenilenenlereKayıtYapılamadı
    End If
End Sub

```

```

Public Sub AbonelikYenilemeyiKontrolEt()
    With frmAnaPencere
        If grKayıt.Collect("BaşSay") <> .txtBaşSay.Text Then
            If vUyarı(AbonelikYenilemeOnayıAl, vbYesNo + vbDefaultButton2 +
vbQuestion) = vbYes Then
                If grKayıt.Collect("AboneYapanNo") = KENDİSİABONENO Then
                    If Not fEskiHesapSil("SELECT * FROM KişiselHesaplar WHERE
AboneSıraNo=" & grKayıt.Collect("SıraNo"), grKayıt.Collect("ParaBirimiKodu"))
Then
                        Hata AboneninBorcuVarYenilemeYapamaz
                        Exit Sub
                    End If
                    BilgileriYazYenileme
                    KendisiAboneYadaGrupÜyesiOlarakKaydet
                    HesapKayıt
                    EnSonYenilenenlereKayıtYap
                    GrupBaşkanıKaydetSil
                Else
                    If .optKendisi.Value Or (grKayıt.Collect("AboneYapanNo") <>
lParametreNoBul(gnBağlantı, .cboGruplar.Text, "Gruplar", "GrupNo", "Ad")) Then
                        KendisiAboneYadaGrupÜyesiOlarakKaydet
                        BilgileriYazYenileme
                        HesapKayıt
                        EnSonYenilenenlereKayıtYap
                        GrupBaşkanıKaydetSil
                    Else
                        BilgileriYazYenileme
                        HesapKayıt
                        EnSonYenilenenlereKayıtYap
                        GrupBaşkanıKaydetSil
                    End If
                End If
            End If
            grKayıt("EnSonYenileme") = CLng(.dtpEnSonYenilemeTarihi.Value)
        End With
    End Sub

```

```

        grKayıt.Update
    End If
End If
End With
End Sub

Private Sub GrupBaşkanıKaydetSil()
    Dim fTMP1 As Boolean, fTMP2 As Boolean
    'bu kişi başka bir grubun başkanı mı
    fTMP1 = fDahaÖncedenKayıtlıMı(gnBağlantı, grYedekKayıt, "SELECT * FROM
    Gruplar WHERE Başkan=" & grKayıt.Collect("SıraNo") & " AND GrupNo<>" &
    grKayıt.Collect("AboneYapanNo"))
    'bu grupta hiç başkan kayıtlı değil mi yada var olan başkanı zaten bu abone mi
    fTMP2 = fDahaÖncedenKayıtlıMı(gnBağlantı, grYedekKayıt, "SELECT * FROM
    Gruplar WHERE GrupNo=" & grKayıt.Collect("AboneYapanNo") & " AND
    (Başkan=0 OR Başkan=" & grKayıt.Collect("SıraNo") & ")")
    With frmAnaPencere
        If .chkGrupBaşkanı.Value = Checked Then
            If Not fTMP1 Then
                If fTMP2 Then
                    grYedekKayıt("Başkan") = grKayıt.Collect("SıraNo")
                    grYedekKayıt.Update
                Else
                    Hata BuGrubunBaşkanıZatenVar
                End If
            Else
                Hata BuAboneBaşkaBirGrubunBaşkanı
            End If
        ElseIf fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT * FROM Gruplar
        WHERE Başkan=" & grKayıt.Collect("SıraNo")) Then
            If grYedekKayıt.RecordCount = 1 Then
                grYedekKayıt("Başkan") = 0
                grYedekKayıt.Update
            End If
        End If
    End With
End Sub

```



```

End If
End With
End Sub

Private Sub HesapKayıt()
    If frmAnaPencere.optKendisi.Value = True Then
        If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT * FROM KişiselHesaplar")
Then
            grYedekKayıt.AddNew
            grYedekKayıt("AboneSıraNo") = grKayıt.Collect("Sırano")
            grYedekKayıt("KalanBorç") = Format(frmAnaPencere.txtÜcret,
gtParaFormatıOku)
            grYedekKayıt.Update
        End If
    Else
        If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT * FROM GrupHesapları
WHERE GrupNo=" & grKayıt.Collect("AboneYapanNo") & " AND
ParaBirimiKodu=" & grKayıt.Collect("ParaBirimiKodu")) Then
            If grYedekKayıt.RecordCount = 0 Then
                grYedekKayıt.AddNew
                grYedekKayıt("GrupNo") = grKayıt.Collect("AboneYapanNo")
                grYedekKayıt("ToplamBorç") = grKayıt.Collect("AbonelikÜcreti")
                grYedekKayıt("KalanBorç") = Format(frmAnaPencere.txtÜcret,
gtParaFormatıOku)
                grYedekKayıt("ParaBirimiKodu") = grKayıt.Collect("ParaBirimiKodu")
                grYedekKayıt.Update
            Else
                If grYedekKayıt("KalanBorç") = 0 Then
                    grYedekKayıt("ToplamBorç") = 0
                End If
                grYedekKayıt("KalanBorç") = grYedekKayıt.Collect("KalanBorç") +
Val(Format(frmAnaPencere.txtÜcret, gtParaFormatıOku))
                grYedekKayıt("ToplamBorç") = grYedekKayıt.Collect("ToplamBorç") +
Val(Format(frmAnaPencere.txtÜcret, gtParaFormatıOku))
            End If
        End If
    End If
End Sub

```

```

        grYedekKayıt.Update
    End If
End If
End If
End Sub

```

```

Public Sub KendisiAboneYadaGrupÜyesiOlarakKaydet()
    If frmAnaPencere.optKendisi.Value = True Then
        grKayıt("AboneYapanNo") = KENDİSİABONENO
    Else
        grKayıt("AboneYapanNo") = lParametreKaydetGüncelle(gnBağlantı,
        grYedekKayıt, "Gruplar", "GrupNo", "", "Ad", frmAnaPencere.cboGruplar.Text, False)
    End If
End Sub

```

```

Public Sub KaydetGüncelle()
    On Error GoTo Hata
    If frmAnaPencere.sstBilgiler.Enabled = False Then
        Hata ÖnceBirKayıtSeçin
        Exit Sub
    End If
    If fGerekliBilgilerGirildiMi Then
        If geGeçerliMod = KayıtModu Then
            If fSQLÇalıştır(gnBağlantı, grKayıt, "SELECT * FROM Aboneler") Then
                grKayıt.AddNew
            End If
            BilgileriYazOrtak
            BilgileriYazAdres
            BilgileriYazAdSoyad
            BilgileriYazYenileme
            KendisiAboneYadaGrupÜyesiOlarakKaydet
            grKayıt("İlkKayıt") = CLng(frmAnaPencere.dtpİlkAbonelikTarihi.Value)
            grKayıt.Update

```

```

If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT * FROM
EnSonYeniKayıtlar WHERE SıraNo=" & grKayıt.Collect("SıraNo")) Then
    grYedekKayıt.AddNew
    grYedekKayıt("SıraNo") = grKayıt.Collect("SıraNo")
    grYedekKayıt.Update
Else
    Hata EnSonYeniKayıtlaraKayıtYapılamadı
End If
HesapKayıt
GrupBaşkanıKaydetSil
ComboBoxlarıGüncelle
ListBoxaYeniElemanEkle frmAnaPencere.lstAboneİsimleri,
grKayıt.Collect("Ad") & " " & grKayıt.Collect("Soyad"), grKayıt.Collect("SıraNo"),
True
    frmAnaPencere.sstBilgiler.Tab = 0
    grKayıt.Update
    AboneBilgileriniGöster
    DurumÇubuğunuGüncelle
Else
    If vUyarı(BilgileriGüncelleştirmeOnayıAl, vbYesNo + vbDefaultButton2 +
vbQuestion) = vbYes Then
        BilgileriYazOrtak
        AdSoyadDeğişikliğiniKontrolEt
        AdresDeğişikliğiniKontrolEt
        AbonelikYenilemeyiKontrolEt
        grKayıt.Update
        ComboBoxlarıGüncelle
        AboneBilgileriniGöster
    End If
End If
Else
    vUyarı BütünBilgilerGirilmemiş, vbCritical + vbOKOnly
End If
Exit Sub

```

Hata:

Hata BilgilerKayıtEdilemedi

End Sub

Public Sub ÖncekiÖdemeler()

Dim tÖncekiÖdemeSQL As String, tÖncekiÖdemeSıraBiçimi As String

With frmAnaPencere

.txtToplamBorç.Text = ""

.txtToplamÖdenen.Text = ""

.txtKalan.Text = ""

tÖncekiÖdemeSQL = "SELECT X.Miktar,Y.Ad,X.Tarih,X.Açıklama FROM  
Ödemeler X,Parabirimleri Y WHERE X.SıraNo=" & grKayıt.Collect("SıraNo") & "  
AND X.ParaBirimiKodu=(SELECT Kod FROM ParaBirimleri WHERE Ad=" & "  
.cboParaBirimi2.Text & ") AND X.ParaBirimiKodu=Y.Kod"

tÖncekiÖdemeSıraBiçimi = " ORDER BY Tarih"

If fSQLÇalıştır(gnBağlantı, grYedekKayıt, tÖncekiÖdemeSQL & "  
tÖncekiÖdemeSıraBiçimi) Then

Set .dbgÖncekiÖdemeler.DataSource = grYedekKayıt

.dbgÖncekiÖdemeler.Columns(0).Caption = "Amount"

.dbgÖncekiÖdemeler.Columns(1).Caption = "Unit"

.dbgÖncekiÖdemeler.Columns(2).Caption = "Date"

.dbgÖncekiÖdemeler.Columns(3).Caption = "Desc."

.dbgÖncekiÖdemeler.Columns(0).NumberFormat = gtParaFormatıYaz

.dbgÖncekiÖdemeler.Columns(2).NumberFormat = "dd/mm/yyyy"

If grKayıt.Collect("AboneYapanNo") = KENDİSİABONENO Then

.txtToplamBorç.Text = Format(grKayıt.Collect("AbonelikÜcreti"),

gtParaFormatıYaz)

If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT \* FROM  
KişiselHesaplar WHERE AboneSıraNo=" & grKayıt.Collect("SıraNo")) Then

.txtKalan.Text = Format(grYedekKayıt.Collect("KalanBorç"),

gtParaFormatıYaz) '0 kaldıysa yazamaz

End If

Else



```

        If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "SELECT * FROM
GrupHesapları WHERE GrupNo=" & grKayıt.Collect("AboneYapanNo") & " AND
ParaBirimiKodu=(SELECT Kod FROM Parabirimleri WHERE Ad="" &
.cboParaBirimi2.Text & "")) Then
            .txtToplamBorç.Text = Format(grYedekKayıt.Collect("ToplamBorç"),
gtParaFormatıYaz)
            .txtKalan.Text = Format(grYedekKayıt.Collect("KalanBorç"),
gtParaFormatıYaz)
        End If
    End If
Else
    Hata ÖncekiÖdemelerOkunamadı
    Exit Sub
End If
If .txtKalan.Text = "" Then ' o yüzden biz yazacaz
    .txtKalan.Text = "0"
End If
.txtToplamÖdenen.Text = Format(Val(Format(.txtToplamBorç.Text,
gtParaFormatıOku)) - Val(Format(.txtKalan.Text, gtParaFormatıOku)),
gtParaFormatıYaz)
If .txtToplamÖdenen.Text = "" Then ' aynı üstteki durum
    .txtToplamÖdenen.Text = "0"
End If
If .txtToplamBorç.Text = "" Then
    .txtToplamBorç.Text = "0"
End If
If Val(.txtKalan.Text) = 0 Then
    .cmdÖdemeYap.Enabled = False
Else
    .cmdÖdemeYap.Enabled = True
End If
End With
End Sub

```

Public Function fEskiHesapSil(ptSQL As String, plParaBirimiKodu As Long) As Boolean

fEskiHesapSil = True

If fSQLÇalıştır(gnBağlantı, grYedekKayıt, ptSQL) Then

If grYedekKayıt.Collect("KalanBorç") = 0 Then

grYedekKayıt.Delete

Else

fEskiHesapSil = False

End If

Else

fEskiHesapSil = False

Hata AboneHesabınaUlaşılamadı

End If

End Function

Public Sub ÖncekiÖdemeleriGöster()

With frmAnaPencere

.cboParaBirimi2.Clear

If grKayıt.Collect("AboneYapanNo") = KENDİSİABONENO Then

.cboParaBirimi2.AddItem .cboParaBirimi1.Text

gfNesneClickleriÇalışabilirMi = True

.cboParaBirimi2.ListIndex = 0

ElseIf .chkGrupBaşkanı.Value = Checked Then

ComboBoxGüncelle .cboParaBirimi2, gnBağlantı, grYedekKayıt, "SELECT X.Ad FROM ParaBirimleri X,GrupHesapları Y WHERE X.Kod=Y.ParaBirimiKodu AND Y.GrupNo=" & grKayıt("AboneYapanNo") & " ORDER BY X.Ad", "Ad", True

If .cboParaBirimi2.ListCount = 0 Then

Hata BuGruptakiParaBirimleriOkunamadı

End If

Else

If .sstBilgiler.Tab = 2 Then 'bu grup üyesi ama başkan değil eğer en son hesaplar tab ı seçil ise bu kişi için o tab dan ayrılmalıyız

.sstBilgiler.Tab = 1

End If

End If  
End With  
End Sub

Public Sub ÖncekiÖdemeleriSil(plParaBirimiKodu As Long)  
    'print-out istiyosa ver ama önceki ödemeleri mutlaka sil  
    If fSQLÇalıştır(gnBağlantı, grYedekKayıt, "DELETE FROM Ödemeler WHERE  
SıraNo=" & grKayıt.Collect("SıraNo") & " AND ParaBirimiKodu=" &  
plParaBirimiKodu, False) Then  
        End If  
End Sub

Public Sub AboneBilgileriniGöster()  
    With frmAnaPencere  
        If .lstAboneİsimleri.ListIndex >= 0 Then  
            If fSQLÇalıştır(gnBağlantı, grKayıt, "SELECT \* FROM Aboneler WHERE  
SıraNo=" & .lstAboneİsimleri.ItemData(.lstAboneİsimleri.ListIndex)) Then  
                .txtAd.Text = grKayıt.Collect("Ad")  
                .txtSoyad.Text = grKayıt.Collect("Soyad")  
                .rtfAdres.Text = grKayıt.Collect("Adres")  
                ComboBoxElemanSeç.cboÜlke, tParametreDeğeriBul(gnBağlantı,  
grKayıt.Collect("ÜlkeKodu"), "Ülkeler", "Kod", "Ad")  
                ComboBoxElemanSeç.cboİl, tParametreDeğeriBul(gnBağlantı,  
grKayıt.Collect("İlKodu"), "İller", "Kod", "Ad", "ÜlkeKodu",  
grKayıt.Collect("ÜlkeKodu"))  
                ComboBoxElemanSeç.cboİlçe, tParametreDeğeriBul(gnBağlantı,  
grKayıt.Collect("İlçeKodu"), "İlçeler", "Kod", "Ad", "İlKodu",  
grKayıt.Collect("İlKodu"))  
                .txtEvTel.Text = grKayıt.Collect("EvTel")  
                .txtİşTel.Text = grKayıt.Collect("İşTel")  
                .txtCepTel.Text = grKayıt.Collect("CepTel")  
                .txtFax.Text = grKayıt.Collect("Fax")  
                .rtfAçıklama.Text = grKayıt.Collect("Açıklama")  
                .txtAboneNo.Text = grKayıt.Collect("AboneNo")

```

.txtBaşSay.Text = grKayıt.Collect("BaşSay")
.dtpİlkAbonelikTarihi.Value = grKayıt.Collect("İlkKayıt")
If grKayıt.Collect("EnSonYenileme") > 0 Then
    .dtpEnSonYenilemeTarihi.Enabled = True
    .dtpEnSonYenilemeTarihi.Value = grKayıt.Collect("EnSonYenileme")
Else
    .dtpEnSonYenilemeTarihi.Value = Date
    .dtpEnSonYenilemeTarihi.Enabled = False
End If
If grKayıt.Collect("AboneYapanNo") = KENDİSİABONENO Then
    .optKendisi.Value = True
    'eğer önceden bu seçiliyse click çalışmayacak o yüzden elle
    'gruplar combosunu temizlicez
    ComboBoxTemizle .cboGruplar
    .sstBilgiler.TabEnabled(2) = True
Else
    .optGrup.Value = True
    ComboBoxElemanSeç .cboGruplar, tParametreDeğeriBul(gnBağlantı,
grKayıt.Collect("AboneYapanNo"), "Gruplar", "GrupNo", "Ad")
    If fdahaÖncedenKayıtlıMı(gnBağlantı, grYedekKayıt, "SELECT * FROM
Gruplar WHERE Başkan=" & grKayıt.Collect("SıraNo")) Then
        .chkGrupBaşkanı.Value = Checked
        .sstBilgiler.TabEnabled(2) = True
    Else
        .chkGrupBaşkanı.Value = Unchecked
        .sstBilgiler.TabEnabled(2) = False
    End If
End If
.txtDağıtımıcıNo.Text = grKayıt.Collect("DağıtımıcıNo")
If grKayıt.Collect("HediyeKitapVerildiMi") Then
    .optEvet.Value = True
    ComboBoxElemanSeç .cboHediyeKitap, tParametreDeğeriBul(gnBağlantı,
grKayıt.Collect("HediyeKitapKodu"), "HediyeKitaplar", "Kod", "Ad")
Else

```



```

.optHayır.Value = True
'eğer önceden bu seçiliyse click çalışmayacak o yüzden elle
'hediyekitapların combosunu temizlicez
ComboBoxTemizle .cboHediyeKitap
End If
.txtÜcret.Text = grKayıt.Collect("AbonelikÜcreti")
ComboBoxElemanSeç .cboParaBirimi1, tParametreDeğeriBul(gnBağlantı,
grKayıt.Collect("ParaBirimiKodu"), "ParaBirimleri", "Kod", "Ad")
ÖncekiÖdemeleriGöster
geGeçerliMod = DüzeltmeModu
End If
End If
End With
End Sub

```

```

Public Sub KendisiAboneOlanlarıYaz()
frmAnaPencere.cmdÖncekiÖdemeleriSil.Enabled = True
frmAnaPencere.cmdÖdemeYap.Enabled = True
ListBoxGüncelle frmAnaPencere.lstAboneİsimleri, grKayıt, "SELECT * FROM
Aboneler WHERE SıraNo IN (SELECT AboneSıraNo FROM KişiselHesaplar WHERE
KalanBorç>0)", "Ad", "Soyad", "SıraNo"
DurumÇubuğunuGüncelle
End Sub

```

```

Public Sub GrupBaşkanlarınıEkle()
Dim lTMP1 As Long, lTMP2 As Long
If fSQLÇalıştır(gnBağlantı, grKayıt, "SELECT * FROM Aboneler WHERE SıraNo
IN (SELECT Başkan FROM Gruplar WHERE GrupNo IN (SELECT DISTINCT
GrupNo FROM GrupHesapları WHERE KalanBorç>0)) ORDER BY Ad") Then
If grKayıt.RecordCount > 0 Then
frmAnaPencere.cmdÖncekiÖdemeleriSil.Enabled = True
frmAnaPencere.cmdÖdemeYap.Enabled = True
lTMP1 = grKayıt.RecordCount
grKayıt.MoveFirst

```

```

For ITMP2 = 1 To ITMP1
    ListBoxaYeniElemanEkle frmAnaPencere.lstAboneİsimleri,
    grKayıt.Collect("Ad") & " " & grKayıt.Collect("Soyad"), grKayıt.Collect("SıraNo"),
    False
    grKayıt.MoveNext
Next
DurumÇubuğunuGüncelle
AboneBilgileriniGöster
End If
End If
End Sub

```

```

Public Sub GrupBaşkanlarınıYaz()
    If gfYöneticiHesabıAktifMi Then
        frmAnaPencere.tbrSolAraçÇubuğu.Buttons.Item(5).Enabled = True
    Else
        frmAnaPencere.tbrSolAraçÇubuğu.Buttons.Item(5).Enabled = False
    End If
    frmAnaPencere.cmdÖncekiÖdemeleriSil.Enabled = True
    frmAnaPencere.cmdÖdemeYap.Enabled = True
    ListBoxGüncelle frmAnaPencere.lstAboneİsimleri, grKayıt, "SELECT * FROM
    Aboneler WHERE SıraNo IN (SELECT Başkan FROM Gruplar WHERE GrupNo IN
    (SELECT DISTINCT GrupNo FROM GrupHesapları WHERE KalanBorç>0)) ORDER
    BY Ad", "Ad", "Soyad", "SıraNo"
    DurumÇubuğunuGüncelle
    AboneBilgileriniGöster
End Sub

```

```

Public Sub YardımıGöster()
    Dim ITMP As Long
    If tDosyaBul(tYardımDosyası) = "" Then
        Hata YardımDosyasıYok
    Else
        ITMP = Shell("notepad.exe " & tYardımDosyası, vbNormalFocus)
    End If
End Sub

```

```
End If  
End Sub
```

```
Public Sub ProgramdanÇık()
```

```
    If Not fVeriTabanıKapat(gnBağlantı, grKayıt) Then
```

```
        Hata VeriTabanıKapatılamadı
```

```
    End If
```

```
End Sub
```

## 7.11 AraçlarModülü SOURCE CODES

Option Explicit

```
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal  
hWnd As Long, ByVal wParam As Long, ByVal lParam As Any) As  
Long
```

```
Public gfSormadanÇık As Boolean
```

```
Public gfNesneClickleriÇalışabilirMi As Boolean
```

```
Public gfYöneticiHesabıAktifMi As Boolean
```

```
Public tYardımDosyası As String
```

```
Public gvCevap As VbMsgBoxResult
```

```
Public gtParaFormatıYaz As String
```

```
Public gtParaFormatıOku As String
```

```
Public Function fBaşkaÇalışanVarMı() As Boolean
```

```
    If App.PrevInstance Then
```

```
        fBaşkaÇalışanVarMı = True
```

```
    Else
```

```
        fBaşkaÇalışanVarMı = False
```

```
    End If
```

```
End Function
```

```
Public Function tDosyaBul(ptDosya As String) As String
```

```
    tDosyaBul = Dir(ptDosya)
```

```
End Function
```

```
Public Function fDosyaKopyala(ptKaynak As String, ptHedef As String) As Boolean
```

```
    On Error GoTo Hata
```

```
    FileCopy ptKaynak, ptHedef
```

```
    fDosyaKopyala = True
```

```
Exit Function
```



Hata:

fDosyaKopyala = False

End Function

Public Sub TextBoxFormatlıYazım(ByRef ptxtTextBox As TextBox, ptYazımBiçimi As String)

ptxtTextBox.Text = Format(ptxtTextBox.Text, ptYazımBiçimi)

ptxtTextBox.SelStart = Len(ptxtTextBox.Text)

End Sub

Public Sub ListBoxGüncelle(ByRef plstListBox As ListBox, ByRef prKayıt As Recordset, ptSQL As String, ptVeriAlanı1 As String, Optional ptVeriAlanı2 As String = "", Optional ptAnahtarAlanı As String = "", Optional plSeçilecekElemanNo As Long = -1)

Dim lTMP As Long

Dim tVeri As String

If fSQLÇalıştır(gnBağlantı, prKayıt, ptSQL) Then

If prKayıt.RecordCount > 0 Then

plstListBox.Clear

For lTMP = 1 To prKayıt.RecordCount

tVeri = prKayıt.Collect(ptVeriAlanı1)

If ptVeriAlanı2 <> "" Then

tVeri = tVeri & " " & prKayıt.Collect(ptVeriAlanı2)

End If

plstListBox.AddItem tVeri

If ptAnahtarAlanı <> "" Then

plstListBox.ItemData(plstListBox.NewIndex) = prKayıt.Collect(ptAnahtarAlanı)

End If

prKayıt.MoveNext

Next

If plSeçilecekElemanNo >= 0 Then

plstListBox.ListIndex = plSeçilecekElemanNo

Else

```

        plstListBox.ListIndex = 0
    End If
End If
End If
End Sub

```

```

Public Sub ComboBoxKopyala(pcboKaynakcomboBox As ComboBox, ByRef
pcboHedefComboBox As ComboBox, Optional ptSeçilecekEleman As String = "")
    Dim lTMP As Long
    pcboHedefComboBox.Clear
    For lTMP = 1 To pcboKaynakcomboBox.ListCount
        pcboHedefComboBox.List(lTMP - 1) = pcboKaynakcomboBox.List(lTMP - 1)
    Next
    ComboBoxElemanSeç pcboHedefComboBox, ptSeçilecekEleman
End Sub

```

```

Public Sub ComboBoxTemizle(ByRef pcboComboBox As ComboBox) 'combo içeriği
durur sadece metni siler ve listindex i 0 yapar
    pcboComboBox.Text = ""
    If pcboComboBox.ListCount > 0 Then
        pcboComboBox.ListIndex = -1
    End If
End Sub

```

```

Public Sub ComboBoxElemanSeç(ByRef pcboComboBox As ComboBox, Optional
ptSeçilecekEleman As String = "")
    Dim lSayaç As Long
    If pcboComboBox.ListCount > 0 Then
        If ptSeçilecekEleman = "" Then
            pcboComboBox.ListIndex = 0
        Else
            For lSayaç = 0 To pcboComboBox.ListCount - 1
                pcboComboBox.ListIndex = lSayaç
                If pcboComboBox.Text = ptSeçilecekEleman Then

```

```

Exit Sub
End If
Next
pcboComboBox.ListIndex = 0
End If
End If
End Sub

```

```

Public Sub ComboBoxGüncelle(ByRef pcboComboBox As ComboBox, pnBağlantı As
Connection, ByRef prKayıt As Recordset, ptSQL As String, ptVeriAlanı As String,
pfNesneClickleriÇalışabilirMi As Boolean, Optional ptSeçilecekEleman As String = "")

```

```

Dim lKayıtAdedi As Long, lSayaç As Long
gfNesneClickleriÇalışabilirMi = pfNesneClickleriÇalışabilirMi
pcboComboBox.Clear
If fSQLÇalıştır(pnBağlantı, prKayıt, ptSQL) Then
    lKayıtAdedi = prKayıt.RecordCount
    If lKayıtAdedi > 0 Then
        For lSayaç = 1 To lKayıtAdedi
            pcboComboBox.AddItem prKayıt.Collect(ptVeriAlanı)
            prKayıt.MoveNext
        Next
        If ptSeçilecekEleman = "" Then
            pcboComboBox.ListIndex = 0
        Else
            ComboBoxElemanSeç pcboComboBox, ptSeçilecekEleman
        End If
    End If
End If
gfNesneClickleriÇalışabilirMi = True
End Sub

```

```

Public Function fDahaÖncedenKayıtlıMı(pnBağlantı As Connection, ByRef prKayıt As
Recordset, ptSQL As String) As Boolean

```

```

If fSQLÇalıştır(pnBağlantı, prKayıt, ptSQL) Then

```

```

If prKayıt.RecordCount = 0 Then
    fDahaÖncedenKayıtlıM1 = False
Else
    fDahaÖncedenKayıtlıM1 = True
End If
End If
End Function

```

```

Public Function tParametreDeğeriBul(pnBağlantı As Connection, plParametreNo As
Long, ptTablo As String, ptParametreNoAlanı As String, ptParametreDeğerAlanı As
String, Optional ptAnahtarAlanı As String = "", Optional plAnahtarNo As Long = -1)
As String

```

```

    Dim rKayıt As Recordset
    Dim tSQL As String
    tSQL = "SELECT * FROM " & ptTablo & " WHERE " & ptParametreNoAlanı &
"=" & plParametreNo
    If ptAnahtarAlanı <> "" Then
        tSQL = tSQL & " AND " & ptAnahtarAlanı & "=" & plAnahtarNo
    End If
    If fSQLÇalıştır(pnBağlantı, rKayıt, tSQL) Then
        If rKayıt.EOF = False Then
            tParametreDeğeriBul = rKayıt.Collect(ptParametreDeğerAlanı)
        Else
            tParametreDeğeriBul = ""
        End If
    Else
        tParametreDeğeriBul = ""
    End If
End Function

```

```

Public Function lParametreNoBul(pnBağlantı As Connection, ptParametreDeğeri As
String, ptTablo As String, ptParametreNoAlanı As String, ptParametreDeğerAlanı As
String, Optional ptAnahtarAlanı As String = "", Optional plAnahtarNo As Long = -1)
As Long

```



```

Dim rKayıt As Recordset
Dim tSQL As String
tSQL = "SELECT * FROM " & ptTablo & " WHERE " & ptParametreDeğerAlanı &
"=" & ptParametreDeğeri & ""
If ptAnahtarAlanı <> "" Then
    tSQL = tSQL & " AND " & ptAnahtarAlanı & "=" & plAnahtarNo
End If
If fSQLÇalıştır(pnBağlantı, rKayıt, tSQL) Then
    If rKayıt.EOF = False Then
        lParametreNoBul = rKayıt.Collect(ptParametreNoAlanı)
    Else
        lParametreNoBul = -1
    End If
Else
    lParametreNoBul = -1
End If
End Function

```

```

Public Function lParametreKaydetGüncelle(pnBağlantı As Connection, prKayıt As
Recordset, ptParametreTablosu As String, ptPrimaryKey As String, ptForeignKey As
String, ptDeğerAlanı As String, ptDeğer As String, pfGüncellemeYapılsınMı As
Boolean, Optional plFKNo As Long, Optional ptYeniDeğer As String, Optional
plYeniFKNo As Long) As Long
    Dim tSQL As String
    tSQL = "SELECT * FROM " & ptParametreTablosu & " WHERE " & ptDeğerAlanı
& "=" & ptDeğer & ""
    If ptForeignKey <> "" Then
        tSQL = tSQL & " AND " & ptForeignKey & "=" & plFKNo
    End If
    If Not fDahaÖncedenKayıtlıMı(pnBağlantı, prKayıt, tSQL) Then
        prKayıt.AddNew
        prKayıt(ptDeğerAlanı) = ptDeğer
        If ptForeignKey <> "" Then
            prKayıt(ptForeignKey) = plFKNo

```

```

End If
prKayıt.Update
ElseIf pfGüncellemeYapılsınMı Then
    tSQL = "SELECT * FROM " & ptParametreTablosu & " WHERE " &
ptDeğerAlanı & "=" & ptYeniDeğer & ""
    If ptForeignKey <> "" Then
        tSQL = tSQL & " AND " & ptForeignKey & "=" & plYeniFKNo
    End If
    If fdahaÖncedenKayıtlıMı(pnBağlantı, prKayıt, tSQL) Then
        lParametreKaydetGüncelle = -1
        Exit Function
    Else
        prKayıt(ptDeğerAlanı) = ptYeniDeğer
        If ptForeignKey <> "" Then
            prKayıt(ptForeignKey) = plYeniFKNo
        End If
        prKayıt.Update
    End If
End If
lParametreKaydetGüncelle = prKayıt(ptPrimaryKey)
End Function

```

'bu yordam listbox ın içeriğine dokunmadan bir elemanı onun sonuna ekler

```

Public Sub ListBoxaYeniElemanEkle(ByRef plstListBox As ListBox, ptEleman As
String, plAnahtar As Long, pfSeçilsinMi As Boolean)
    Dim lTMP As Long
    lTMP = plstListBox.ListCount
    plstListBox.AddItem ptEleman, lTMP
    'plstListBox.ItemData(lTMP) = plAnahtar
    plstListBox.ItemData(plstListBox.NewIndex) = plAnahtar
    If pfSeçilsinMi Then
        plstListBox.ListIndex = lTMP
    End If
End Sub

```

```
Public Function fListBoxElemanSil(ByRef plstListBox As ListBox, plEleman As Long)  
As Boolean
```

```
    fListBoxElemanSil = False
```

```
    If plstListBox.ListIndex >= 0 Then
```

```
        plstListBox.RemoveItem (plEleman)
```

```
        fListBoxElemanSil = True
```

```
    If plstListBox.ListCount > 0 Then
```

```
        If plEleman <= (plstListBox.ListCount - 1) Then
```

```
            plstListBox.ListIndex = plEleman
```

```
        Else
```

```
            plstListBox.ListIndex = plEleman - 1
```

```
        End If
```

```
    End If
```

```
End If
```

```
End Function
```

```
Public Sub ListBoxaYatayKaydırmaÇubuğuEkle(ByRef plstListe As ListBox)
```

```
    Dim fTMP As Boolean
```

```
    fTMP = SendMessage(plstListe.hwnd, &H415, 2 * plstListe.Width /
```

```
Screen.TwipsPerPixelX, 0)
```

```
End Sub
```

## 7.12 HataVeUyarıModülü SOURCE CODES

Option Explicit

Enum HataKodları

BirdenFazlaProgramÇalıştırma  
VeriTabanıAçılamadı  
VeriTabanıKapatılamadı  
GeçersizKullanıcıAdıYadaŞifre  
YardımDosyasıYok  
EnSonYeniKayıtlaraKayıtYapılamadı  
EnSonYenilenenlereKayıtYapılamadı  
EnSonAdSoyadDeğiştirilenlereKayıtYapılamadı  
EnSonAdresDeğiştirenlereKayıtYapılamadı  
EnSonİptalEdilenlereKayıtYapılamadı  
BilgilerKayıtEdilemedi  
ÖnceBirKayıtSeçin  
BuGrubunBaşkanıZatenVar  
BuAboneBaşkaBirGrubunBaşkanı  
ÖncekiÖdemelerOkunamadı  
BuGruptakiParaBirimleriOkunamadı  
LütfenGeçerliBirMiktarGirin  
ÖdemeKayıtEdilemedi  
AboneHesabınaUlaşılamadı  
AboneninBorcuVarYenilemeYapamaz  
BuAboneGrupBaşkanıGrupDeğiştiremez  
BirParametreDeğeriGirilmeli  
BuParametreyiKullananKayıtlarVar  
EnAzBirKullanıcıKayıtlıOlmalı  
BuİsimdeBirKullanıcıZatenVar

End Enum

Enum UyarıKodları



```

BütünBilgilerGirilmemiş
Çıkış
BilgileriGüncelleştirmeOnayıAl
AbonelikYenilemeOnayıAl
AboneİptalEtmeOnayı
AboneSilmeOnayı
KayıtBulunamadı
ListeSilinsinMi
End Enum

Public Sub Hata(peHataKodu As HataKodları, Optional vTuşVeSimgeler As
VbMsgBoxStyle = vbCritical, Optional tBaşlık As String = "ERROR")
    Dim tHataMetni As String
    Select Case peHataKodu
        Case BirdenFazlaProgramÇalıştırma
            tHataMetni = "PROGRAM IS WORKING"
        Case VeriTabanıAçılamadı
            tHataMetni = "DATABASE COULDN'T BEEN OPENED"
        Case VeriTabanıKapatılamadı
            tHataMetni = "DATABASE COULDN'T BEEN CLOSED"
        Case GeçersizKullanıcıAdıYadaŞifre
            tHataMetni = "INVALID USER NAME OR PASSWORD"
        Case YardımDosyasıYok
            tHataMetni = tYardımDosyası & " IS NOT FOUND"
        Case EnSonYeniKayıtlaraKayıtYapılamadı
            tHataMetni = "IT IS NOT RECORDED TO THE LIST OF LAST NEW
RECORDS"
        Case EnSonYenilenenlereKayıtYapılamadı
            tHataMetni = "IT IS NOT RECORDED TO THE LIST OF LAST UPDATES"
        Case EnSonAdSoyadDeğiştirilenlereKayıtYapılamadı
            tHataMetni = "IT IS NOT RECORDED TO THE LIST OF LAST
NAME/SURNAME"
        Case EnSonAdresDeğiştirenlereKayıtYapılamadı

```

tHataMetni = "IT IS NOT RECORDED TO THE LIST OF LAST ADRES  
CHANGES"

Case EnSonİptalEdilenlereKayıtYapılamadı

tHataMetni = "IT IS NOT RECORDED TO THE LIST OF LAST CANCELED  
RECORDS"

Case BilgilerKayıtEdilemedi

tHataMetni = "THE INFORMATIONS ARE NOT RECORDED"

Case ÖnceBirKayıtSeçin

tHataMetni = "TO DO THE OPERATION CHOOSE A RECORD"

Case BuGrubunBaşkanıZatenVar

tHataMetni = "THIS GROUP HAS A MANAGER"

Case BuAboneBaşkaBirGrubunBaşkanı

tHataMetni = "THIS MAN IS THE MANAGER OF ANOTHER GROUP"

Case ÖncekiÖdemelerOkunamadı

tHataMetni = "FORMER PAYMENTS COULDN'T BEEN READ"

Case BuGruptakiParaBirimleriOkunamadı

tHataMetni = "THE MONEY UNITS OF THIS GROUP COULDN'T BEEN  
READ"

Case LütfenGeçerliBirMiktarGirin

tHataMetni = "PLEASE ENTER A VALID AMOUNT OF MONEY"

Case ÖdemeKayıtEdilemedi

tHataMetni = "THE PAYMENT COULDN'T BEEN RECORDED"

Case AboneHesabınaUlaşılamadı

tHataMetni = "THE ACCOUNT OF THIS RECORD COULDN'T BEEN  
FOUND"

Case AboneninBorcuVarYenilemeYapamaz

tHataMetni = "UPDATE IS IMPOSSIBLE BECAUSE THIS RECORD HAS  
DEPT"

Case BuAboneGrupBaşkanıGrupDeğiştiremez

tHataMetni = "THIS RECORD IS THE MANAGER OF A GROUP."

Case BirParametreDeğeriGirilmeli

tHataMetni = "PLEASE ENTER A PARAMETER VALUE"

Case BuParametreyiKullananKayıtlarVar

```
tHataMetni = "THERE ARE SOME RECORDS THAT USE THIS  
PARAMETER"
```

```
Case EnAzBirKullanıcıKayıtlıOlmalı
```

```
tHataMetni = "THERE MUST BE AT LEAST ONE USER"
```

```
Case BuİsimdeBirKullanıcıZatenVar
```

```
tHataMetni = "THERE IS A USER THAT HAS SAME NAME"
```

```
Case Else
```

```
tHataMetni = "UNKNOWN ERROR"
```

```
End Select
```

```
MsgBox tHataMetni, vTuşVeSimgeler, tBaşlık
```

```
End Sub
```

```
Public Function vUyarı(peUyarıKodu As UyarıKodları, vTuşVeSimgeler As  
VbMsgBoxStyle, Optional tBaşlık As String = "WARNING") As VbMsgBoxResult
```

```
Dim tUyarıMetni As String
```

```
Select Case peUyarıKodu
```

```
Case BütünBilgilerGirilmemiş
```

```
tUyarıMetni = "YOU MUST ENTER ALL THE FIELDS"
```

```
Case Çıkış
```

```
tUyarıMetni = "ARE YOU SURE ?"
```

```
Case BilgileriGüncelleştirmeOnayıAl
```

```
tUyarıMetni = "ARE YOU SURE YOU WANT TO UPDATE ?"
```

```
Case AbonelikYenilemeOnayıAl
```

```
tUyarıMetni = "ARE YOU SURE YOU WANT TO UPDATE THE BEGINING  
ISSUE"
```

```
Case AboneİptalEtmeOnayı
```

```
tUyarıMetni = "ARE YOU SURE YOU WANT TO CANCEL THIS  
SUBSCRIPTION"
```

```
Case AboneSilmeOnayı
```

```
tUyarıMetni = "ARE YOU SURE YOU WANT TO DELETE THIS  
SUBSCRIPTION"
```

```
Case KayıtBulunamadı
```

```
tUyarıMetni = "NO RECORDS FOUND"
```

```
Case ListeSilinsinMi
```

```
tUyarıMetni = "DELETE LIST"
Case Else
    tUyarıMetni = "UNKNOWN WARNING"
End Select
vUyarı = MsgBox(tUyarıMetni, vTuşVeSimgeler, tBaşlık)
End Function
```



### 7.13 VeriTabanıModülü SOURCE CODES

Option Explicit

Public gtDosyaAdı As String

Public gtYedekDosyaAdı As String

Public gtSql As String

Public gnBağlantı As Connection

Public grKayıt As Recordset

Public grYedekKayıt As Recordset

Public Function fVeriTabanıAç(ptDosyaAdı As String, ByRef pnBağlantı As  
Connection, ByRef prKayıt As Recordset, ptSQL As String) As Boolean

On Error GoTo Hata

Set pnBağlantı = New Connection

pnBağlantı.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & ptDosyaAdı  
& ";Persist Security Info=False"

If fSQLÇalıştır(pnBağlantı, prKayıt, ptSQL) = True Then

fVeriTabanıAç = True

Else

fVeriTabanıAç = False

End If

Exit Function

Hata:

fVeriTabanıAç = False

End Function

Public Function fVeriTabanıKapat(ByRef pnBağlantı As Connection, ByRef prKayıt  
As Recordset) As Boolean

On Error GoTo Hata

prKayıt.Close

pnBağlantı.Close

Set pnBağlantı = Nothing

Set prKayıt = Nothing

fVeriTabanıKapat = True

fVeriTabanıYedekle gtDosyaAdı, gtYedekDosyaAdı

Exit Function

Hata:

fVeriTabanıKapat = False

End Function

Public Function fVeriTabanıYedekle(ptDosyaAdı As String, ptYedekDosyaAdı As String) As Boolean

If fDosyaKopyala(ptDosyaAdı, ptYedekDosyaAdı) Then

fVeriTabanıYedekle = True

Else

fVeriTabanıYedekle = False

End If

End Function

Public Function fSQLÇalıştır(pnBağlantı As Connection, ByRef prKayıt As Recordset, ptSQL As String, Optional pfElemanSayısıGerekliMi As Boolean = True) As Boolean

On Error GoTo Hata

Set prKayıt = New Recordset

prKayıt.CursorLocation = adUseClient

prKayıt.Open ptSQL, pnBağlantı, adOpenKeyset, adLockOptimistic

If pfElemanSayısıGerekliMi Then

IKayıtElemanSayısı prKayıt

End If

fSQLÇalıştır = True

Exit Function

Hata:

fSQLÇalıştır = False

End Function

Public Sub IKayıtElemanSayısı(ByRef prKayıt As Recordset)

```
If Not prKayıt.EOF Then  
    prKayıt.MoveLast  
    prKayıt.MoveFirst  
End If  
End Sub
```