



NEAR EAST UNIVERSITY

I. Table Of Contents

II. Acknowledgements

III. **Faculty of Administrative Sciences**

IV. Introduction

V. System Design

Department Of

VI. Database design

VII. **Computer Information Systems**

1. User Guidelines

2. User Flow

MOBILE SHOP SYSTEM

3. DFD

VIII. Conclusion

Graduation Project

IX. References

CIS - 400

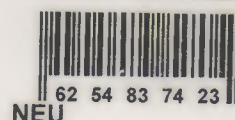
X. Appendix

SUPERVISOR: Asst. Prof. Dr. Nadire Cavus

STUDENT NO: 20011852

ADNAN ABDUL MANNAN

Nicosia – 2010





CONTENTS

I. Table Of Contents	
II. Acknowledgements	i
III. Abstract	ii
IV. Introduction	iii
V. System Design	1
VI. Database design	2
VII. User Manual	6
. User Guidelines	6
. User Flowcharts	19
. DFD	22
VIII. Conclusion	28
IX. References	29
X. Appendix	30

ACKNOWLEDGMENT

ABSTRACT

First of all I would like to thank God for guiding me through my education.

Also I dedicate my graduation and this book and my success to my family, thanks to God I succeeded and made one of their wishes come true.

More over I feel proud to pay my special regards to my project supervisor Dr.Nadire Çavus. She never disappointed me in any affair. She delivered too much information to me and did hers best efforts to make me able to complete my graduation project. She has divine place in my heart. I am really thankful to her.

Also, I want to pay special regards to all of my family who are enduring all of these expenses and supporting me in all events, Could do nothing without their prayers. They always encouraged me in all crises and through all issues. I will never forget their sacrifices for me to complete my education so that I can enjoy my successful life as they expect.

I also want to honor all of these people who gave me all the knowledge that I needed during my education period specially the head of faculty and the chairman of the computer information department Dr.Yalcin Akcali.

I also want to pay my special thanks to my friends who was always supporting me and advised me about facing all the difficulties I had during my life time in Cyprus.

ABSTRACT

This project is an implementation of Visual Basic in the mobiles sales section; here we use a Visual Basic application to manage a mobiles shop with all possible actions and characteristics like customers, stocks, users and receipts.

This project uses Microsoft Visual Basic 6.0, Microsoft Access Database and Structured Query Language (SQL) queries to manipulate the database.

The aim of this project is to help the user to manage the data storage and use it when it is needed.

Customers pass by in mobile shop to search for the best quality of mobiles and prices, some customers pay in cash and some pay by using credit cards. So the selling operation must be managed and controlled daily. On the other hand the company must hire employees with full information about them in which they can help the customers in finding the best cell phones and writing full information about the stock sells with the customer information.

This program helps user to manage the whole information in general and save it in database so they can use it later when it's needed

EXPLANATION

INTRODUCTION

This project describes database system concepts and a simple computer shop program which named computer shop. The application part uses Access and SQL quires. The program is written by Microsoft Visual Basic 6.0 and Microsoft Access to create database and uses SQL queries.

Describes the program execution and basics about purchasing and selling mobiles related accessories and stocks and the use of a mobile shop management system.

EXPLANATION

The minimum requirements of the system are as follows:

Intel Pentium III processor

550 Mhz

128MB of RAM

30 GB harddisk

14" Monitor

Microsoft Windows 2000

LG CD-ROM CRD 8522B

DAVIDOM 9102-BASED PCI Fast Ethernet Adapter

The recommended requirements of the system are as follows:

Intel Pentium IV processor

800Mhz

256MB of RAM

60 GB harddisk

17" Monitor

Microsoft Windows XP Professional

LG CD-ROM CRD 8522B

DAVID OM 9102-BASED PCI Fast Ethernet Adapter

I used Visual Basic programming language for writing this project and Sql And Access database for keeping the data and maintaining it.

MOBILE SHOP SYSTEM

1. PROJECT IDENTIFICATION AND SELECTION

THE AIM OF THE PROJECT:

The Mobile shop system, aims to serve the customers and suppliers who come to shop for mobile and its accessories and the to provide better satisfaction services to the buying customers and the suppliers.

The existing systems handicaps, the easiness and flexibility that the new system wishes to bring to the users are explained in detail under the headlines ``The Existing System Capabilities`` and `` New system Providences``.

THE PROJECT BOUNDARIES:

1. This system has users identifications with information of existing and adding a new user .
2. There is also customers detail in which stored in data.
3. The companies or suppliers information is also stored from which buyer what we bought.
4. There is stocks which means products in which we have our mobiles accessories.
5. This is the Receipts in which we have the sales of products to customers and the products we bought from companies.

RESOURCE LIMITS:

The budget and system is cheap for the users who are working in the shop to help the customers.

2 . PROJECT INITIATION AND PLANNING

TECHNICAL FEASIBILITY:

- Computers for the existing user,
- Computers for the new user,
- Computer and printer for payment receipts.

Besides these hardwares , some hardwares and softwares for the new network system that will be applied and afforded as well.

OPERATIONAL FEASIBILITY:

A test has been made among the users who will be working in the shop so they have the basic knowledge about computers and they should know about the system.

ECONOMICAL FEASIBILITY:

All the expenses will be afforded by the owner of the shop or user.

3. ANALYSIS

THE REQUIREMENT DETERMINATION:

The requirement made by the user are as follows:

- To increase the speed of operation
- To prevent loss of data
- To prevent easy and quick data access
- To restrict data access with the means of system security.

NEW SYSTEMS PROVIDENCES

- The new system will help the users to enter the customer or suppliers data in the database.
- The system will be accessed by only authorized users. The customer and companies info or details can only be changed by the user.
- All data will be kept in and stored in the computer database.
- Users can see through the database the list of customers and companies who bought and have history of all the sales.

4. REPORT

Each of this program is explained in detail below

USER IDENTIFICATION: The form contains the user's name, login name and password and user's authorization, here there is user type which defines the restriction level for each user.

STOCKS: In this form we can add, update or delete the stock items and we can also see the stock items.

COMPANIES: In this form we can add, update or delete the companies and we can also see the list of companies and add or delete or update them.

CUSTOMERS: In this form it allows us to add, update or delete the customers and we can also see the list of customers and add or delete or update them.

RECEIPTS

Product Receipt Receipt: It is given to the customer to which customer the product is sold. After the receipt will be printed and printed with the receipt for security reasons.

Product Receipt Receipt: It is given to the customer to which customer the product is sold. After the receipt will be printed and printed with the receipt for security reasons.

REPORTS: It shows the companies and customers report.

SYSTEM DEFINITION:

Mobile Shop is a business where we buy and sell mobiles like sim cards, headphones, accessories etc such as buying new mobiles from the companies or selling to the local people and then sell them with a bit higher price to make profit.

Solution:

The system defined above works in both fields that is buying and sale as well as keeps the track of its customer's records too. Mainly I divide the system into five (6) main processes.

1. User Identifications

2. Stocks

3. Companies

4. Customers

5. Receipts

6. Reports

Each of this process is explained in detail below.

USERS IDENTIFICATION: The form contains the users and their login names and passwords and their information, here there is user type which defines the restriction level for every user.

STOCKS: In this form we can edit update accessories like sim cards top up cards and mobiles.

COMPANIES: In this form we add companies that produces mobiles and sells it to us so we sell it to other customers and edit or delete or update also.

CUSTOMERS: In this form it shows us the details of a customer who is willing to buy a product from the shop. In this we can add delete and update customers in the database.

RECEIPTS:

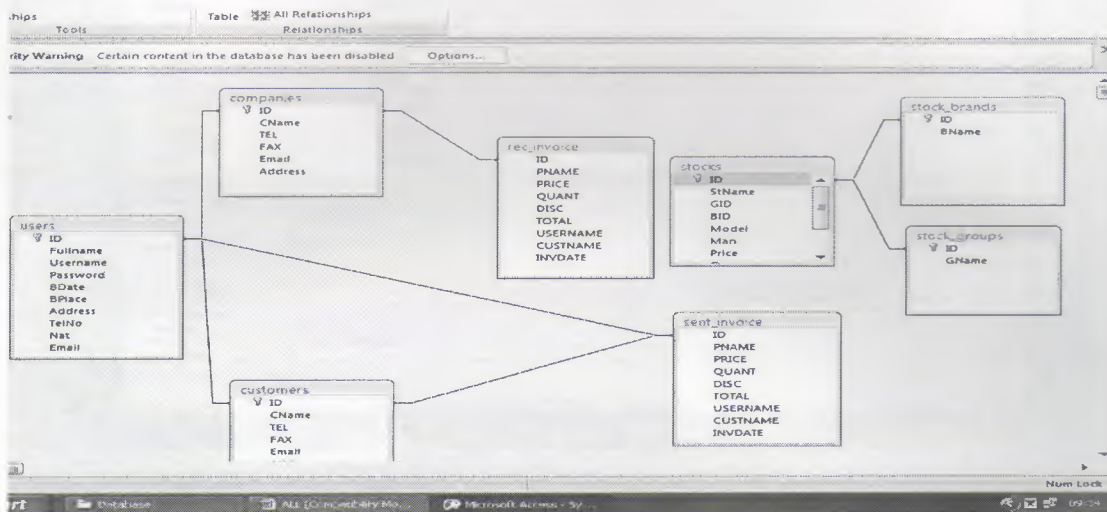
Sent Invoice Receipt: It is given to the customers to which customer the product is sold. Also the username will be saved and printed with the receipt for security reasons.

Receipt Invoice Receipt: It is given to the companies we buy merchandise from and we do not select customer name, we select companies name.

REPORTS: It shows the companies and customers report.

SYSTEM DESIGN

Relationships



Users

The screenshot shows the 'users' table design view in Microsoft Access. The table structure is as follows:

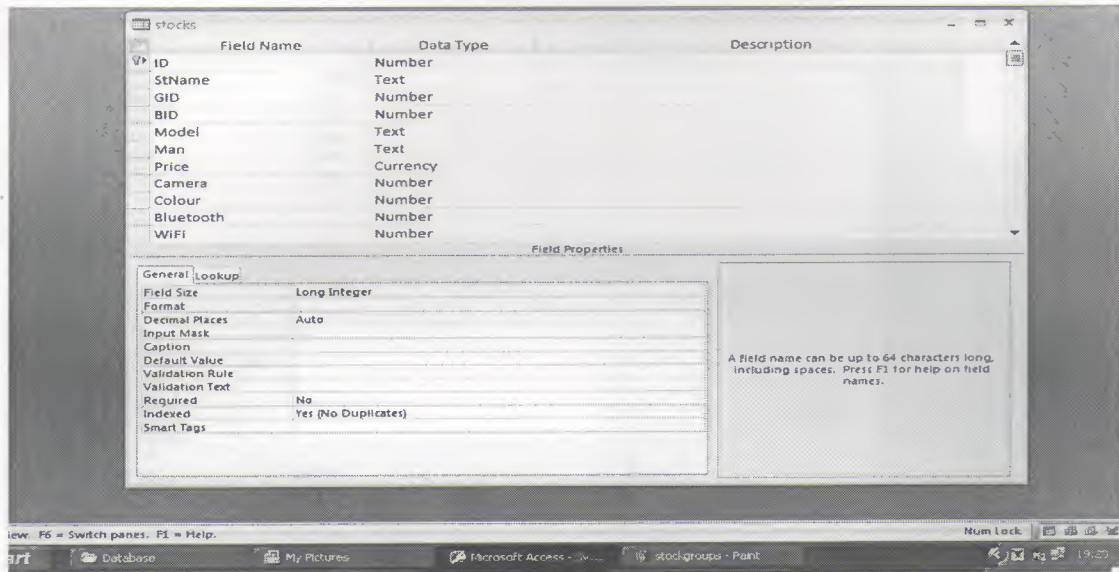
Field Name	Data Type	Description
ID	Number	
Fullname	Text	
Username	Text	
Password	Text	
BDate	Date/Time	
BPlace	Text	
Address	Text	
TelNo	Text	
Nat	Text	
Email	Text	

Below the table structure, the 'Field Properties' section is visible, showing the 'General' tab for the 'ID' field:

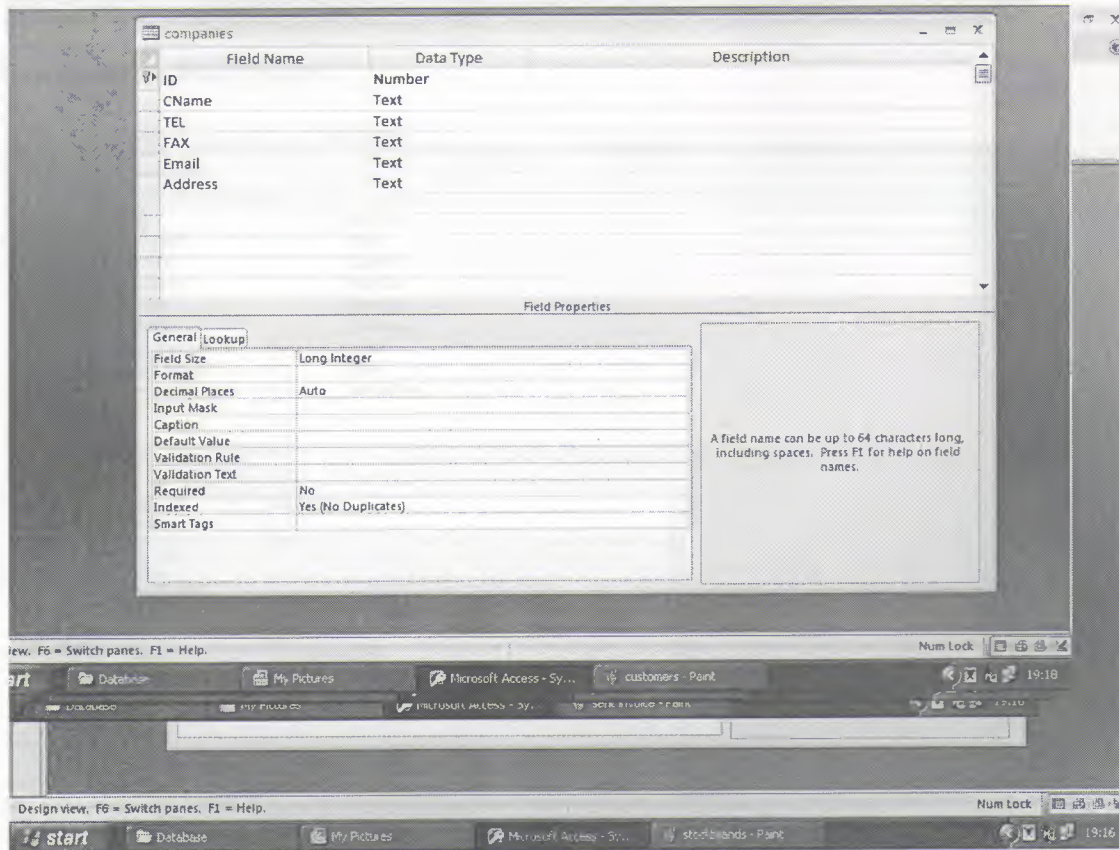
Property	Value
Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Indexed	Yes (No Duplicates)
Smart Tags	

A note on the right side of the Field Properties section states: 'A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.'

Stocks



Companies



Customers

customers

Field Name	Data Type	Description
ID	Number	
CName	Text	
TEL	Text	
FAX	Text	
Email	Text	
Address	Text	

Field Properties

General

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Indexed	Yes (No Duplicates)
Smart Tags	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Design view. F6 = Switch panes. F1 = Help.

Receipts Received Invoice

rec_invoice

Field Name	Data Type	Description
ID	Number	
PNAME	Text	
PRICE	Currency	
QUANT	Number	
DISC	Number	
TOTAL	Currency	
USERNAME	Text	
CUSTNAME	Text	
INVDAT	Date/Time	

Field Properties

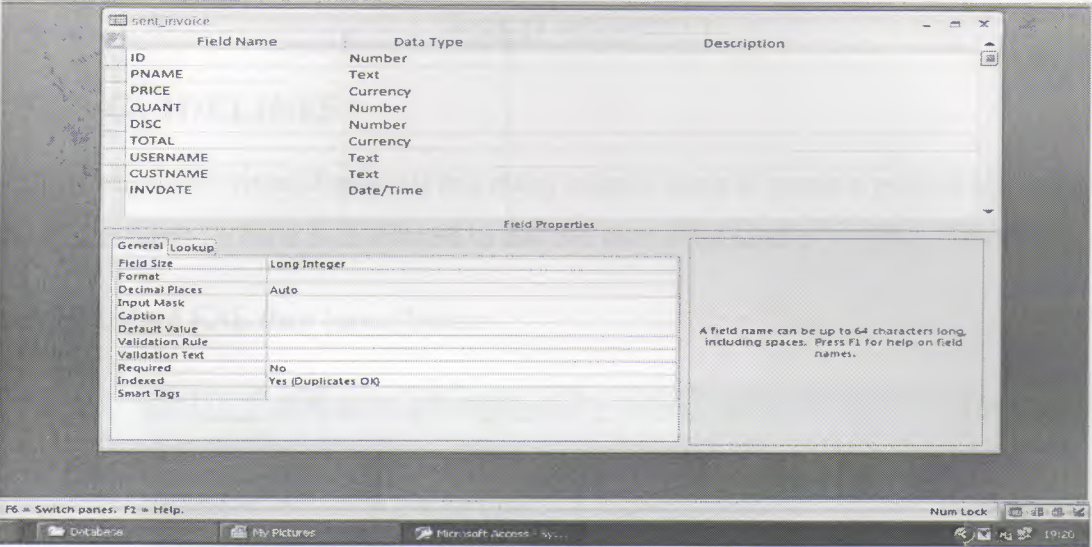
General

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Indexed	Yes (Duplicates OK)
Smart Tags	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Design view. F6 = Switch panes. F1 = Help.

Receipts Sent Invoice



USER MANUAL

USER GUIDELINES

Microsoft visual Basic 6.0 has many special tools to create a project after opening the visual basic , a form is displayed to ask the user what kind of porject is to be selected,

Click standard EXE then (open)button

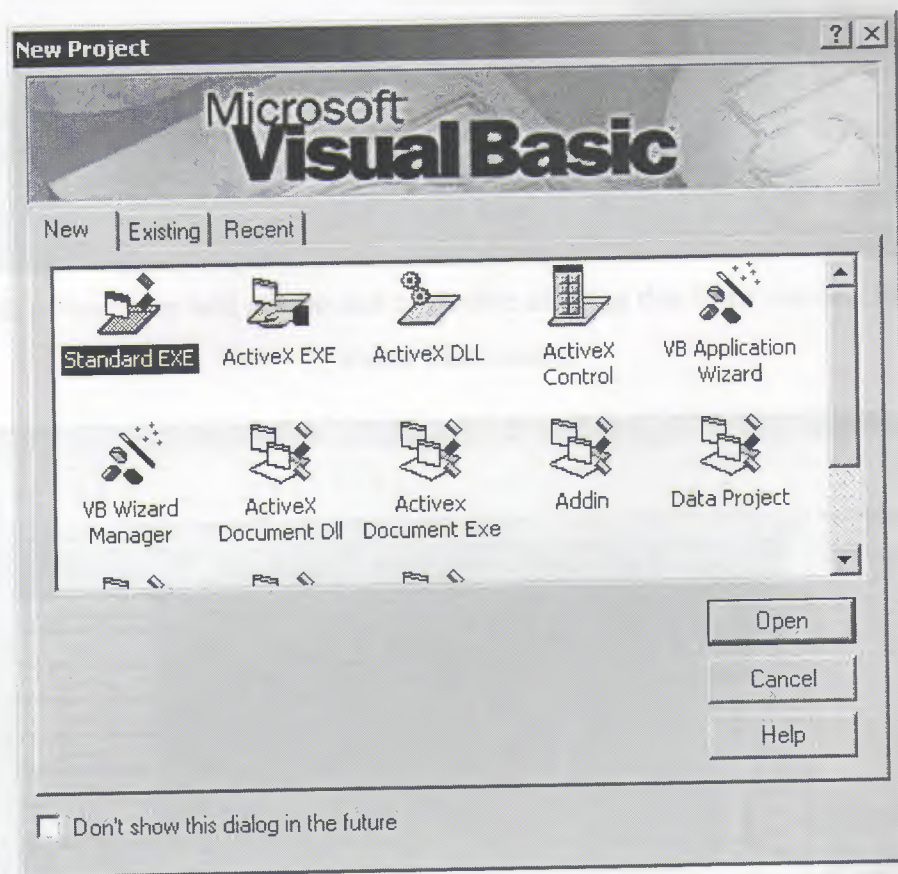


Figure 01 Starting a New Project

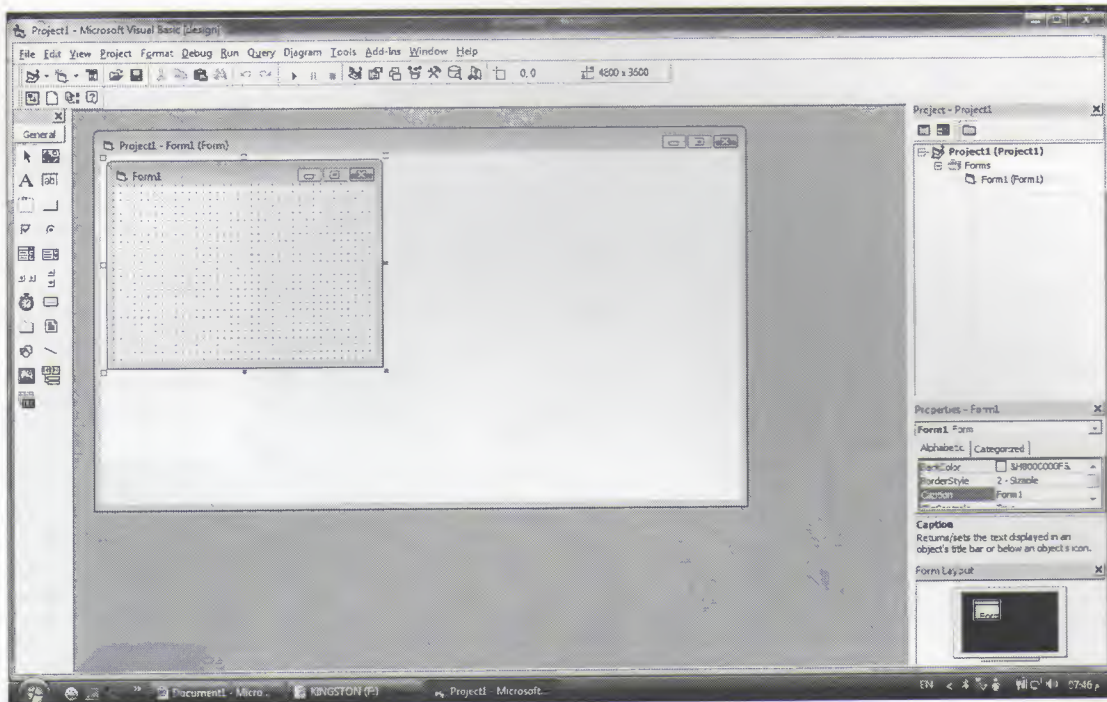


Figure 02 a New form will appear and by double clicking this form you can start writing visual basic codes.



Figure 03 After finishing with writing the codes click the start button from the visual basic menu or easily press F5 button from keyboard. The program will start running unless there are errors on the code.



Figure 04 After testing the project if everything works fine click the visual basic menu file and then save project button, all the changes will be saved.



Figure 05 Saving a project and making a executable file of the project is very easy. by clicking file menu of visual basic then select make EXE selection here.

DESCRIPTION OF FORMS

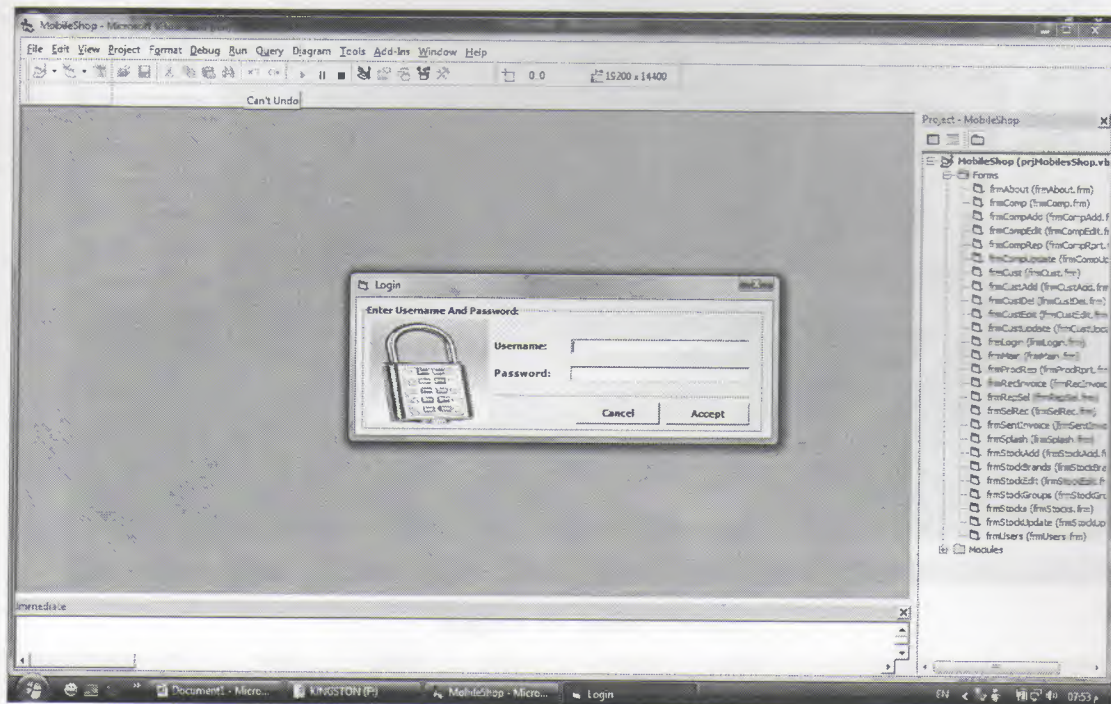


Figure 06 Login Form

In this form users should enter their users name and password to be able to access the program, if the user typed a wrong user name or password a message will be displayed saying invalid user name or password, otherwise the user will enter the program successfully and there are three types of users, admin, users, and guest.

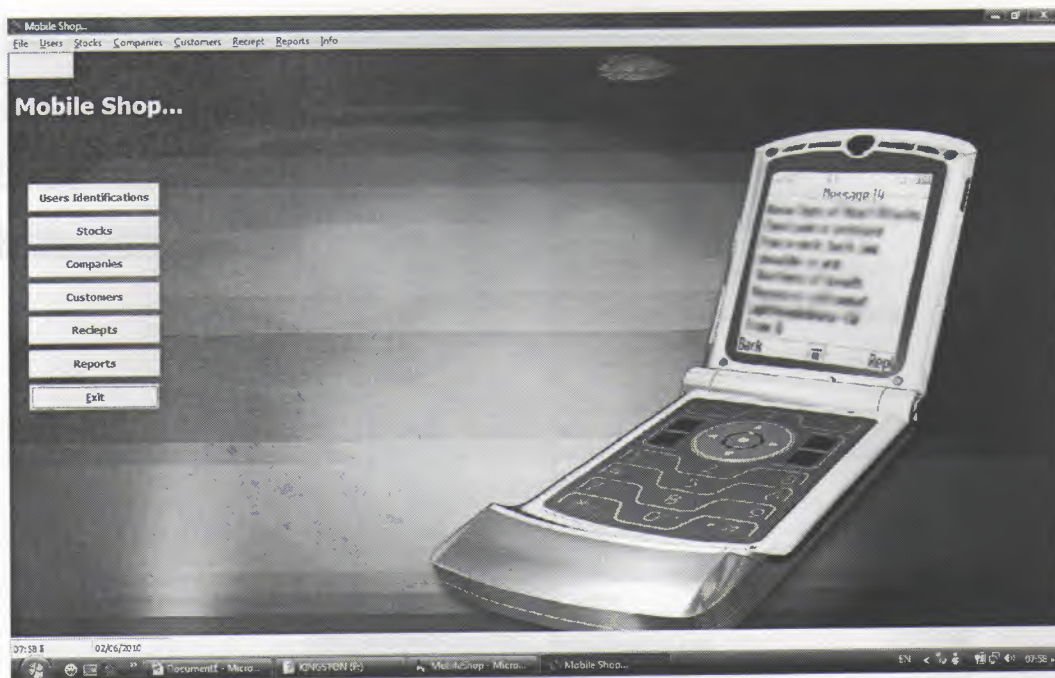


Figure 07 Main Menu.

This is the main menu form, in this form there are 7 menus and each menu access a different section where you can do different operations.

For the file menu there is the exit menu which will exit the program, for the users menu it will lead you to users identifications, for the stocks menu it will lead you to add stock menu, update stock, stock groups and stock brand names, for the companies menu it will lead you to add supplier menu and update supplier menu. For the customers menu it will lead you to add customer menu and update customer menu. The receipt menu it will lead you to add customer menu and update customer menu. There receipt menu contains customer receipt and supplier receipt. Finally the head menu which contain the about menu.

Users Identifications

Search For Username:

emrah

Full Name: emrah

Username: emrah

Password: ****

Birth Date: 5 / 9 /1985

Birth Place: adana

Address: gazipasa bulvari

Telephone No: 1234567

Nationality: turkiye

E-Mail: fgfg@yahoo.com

Add Edit Update Delete

Main Menu

Figure 08 Users Identifications

This form contains the users and their login names and passwords and their information, here there is user type which defines the restriction level for every user.

Add Stock

Add Stock Information

ID: 8

Stock Name: Charger

Group Name: Accessories

Brand Name: Nokia

Model: E61

Manufactured In: Japan

Price: 500.00 TL

Extra Options:

- ☐ Camera
- ☒ Bluetooth
- ☒ Colourful Display
- ☒ WIFI

Close Clear Add

Figure 09 Add Stock

This form adds a new stock to the database using embedded SQL queries.

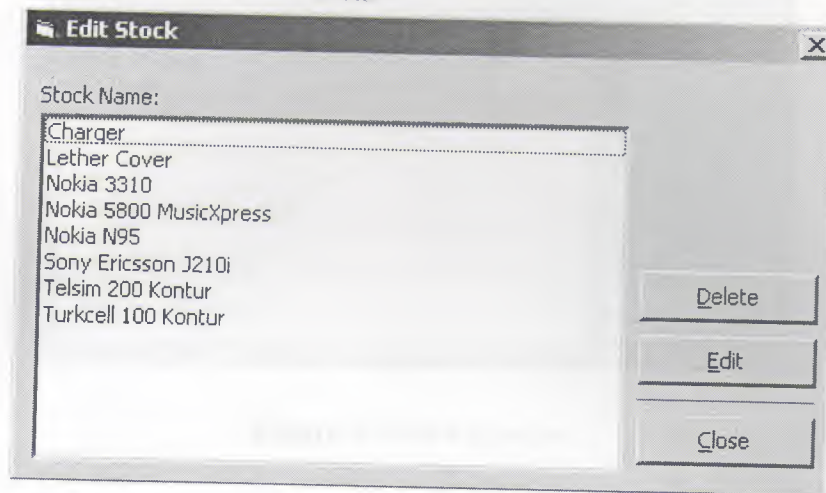


Figure10 Edit Stock

In this form we can select the stock we want to edit then we will edit in the update stock form in figure 5.11

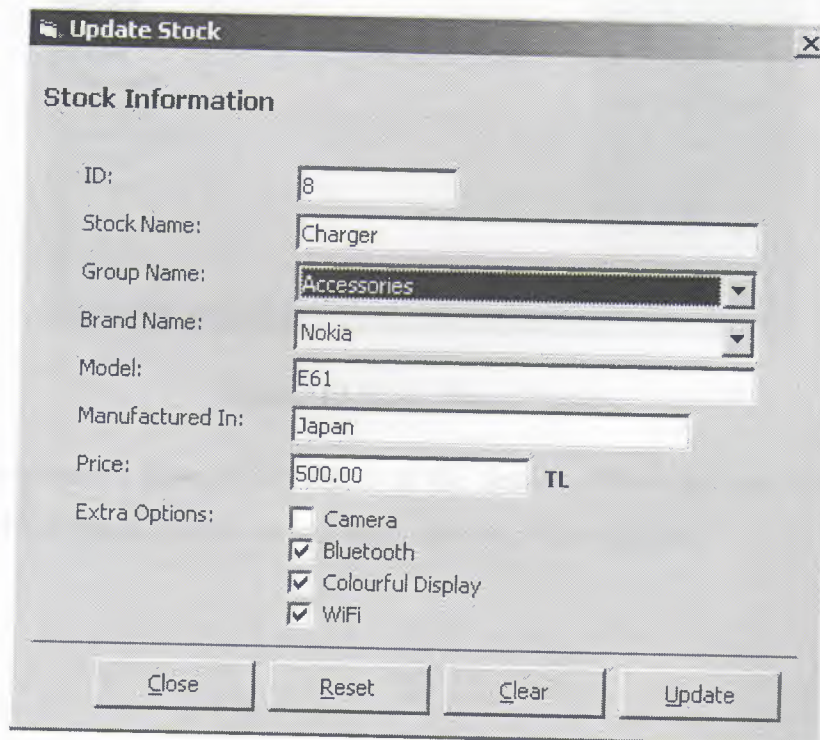


Figure 11 Update Stock

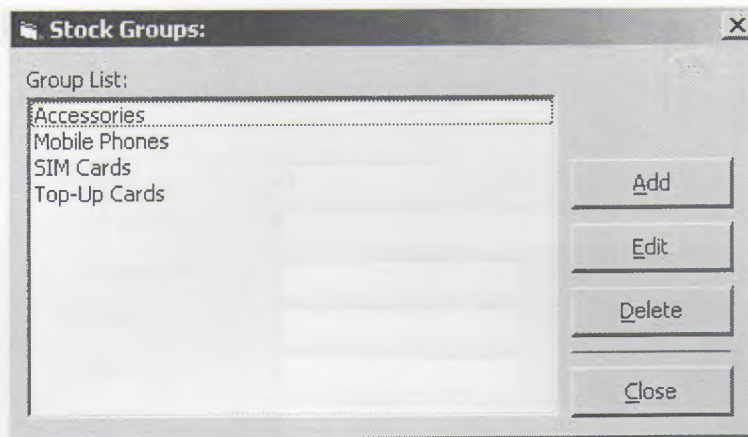


Figure 12 Stock Groups

This is the stock groups form which groups different types of stocks like mobile, mobile accessories.

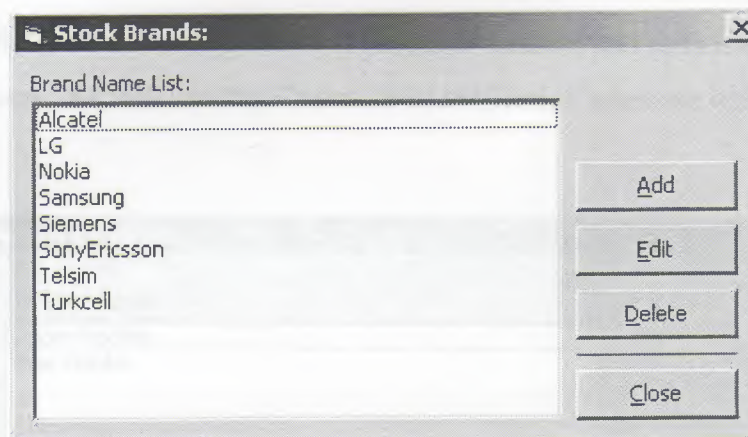


Figure 13 Stock Brand Names

This is a customer form which shows us the details of the customer who is willing to buy a mobile from the shop here we can add customer informations.

Add Company

Add Company Information

ID: 3

Company Name:

Telephone No:

Fax No:

E-mail:

Address:

Close Clear Add

Figure 14 Add Company

In this form we add companies that produces mobiles and sells it to us so we sell it to other customers , we will use this Customer Identification when we write a received invioce.

Edit Company

Company Name:

Golden Mobiles

Ybler Mobiles

Delete

Edit

Close

Figure 15 Edit Company

Here we can see all defined companies as we can see we can select a customer by clicking on it's name and then we can delete it like in the figure 5.16 or we can update it like in the figure 5.17

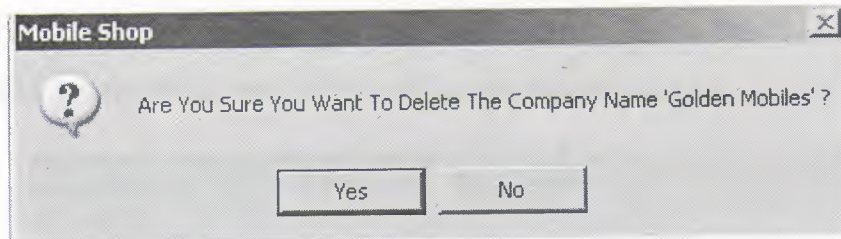


Figure 16 Delete Company

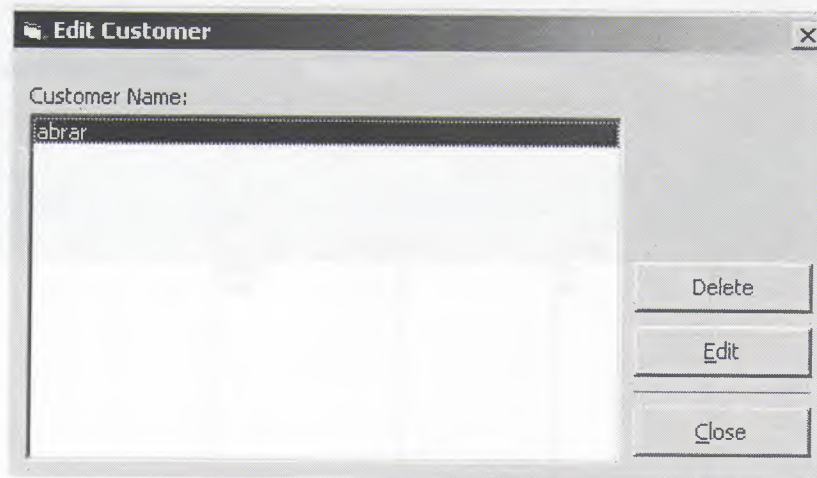
A screenshot of a Windows-style dialog box titled "Update Company". It has a "Company Information" section with the following fields: "ID:" with value "1", "Company Name:" with value "Golden Mobiles", "Telephone No:" with value "0392 228 98 56", "Fax No:" with value "039 2287 987", "E-mail:" with value "golden_mobile@live.com", and "Address:" with value "turkey,istanbul". At the bottom, there are four buttons: "Close", "Reset", "Clear", and "Update".

Figure 17 Update Company

A screenshot of a Windows-style dialog box titled "Add Customer". It has an "Add Customer Information" section with the following fields: "ID:" with value "2", "Customer Name:" (empty), "Telephone No:" (empty), "Fax No:" (empty), "E-mail:" (empty), and "Address:" (empty). At the bottom, there are three buttons: "Close", "Clear", and "Add".

Figure 18 Add Customer

. This is a customer form which shows us the details of a customers who is willing to buy a product from the shop. Here we can add customers information.



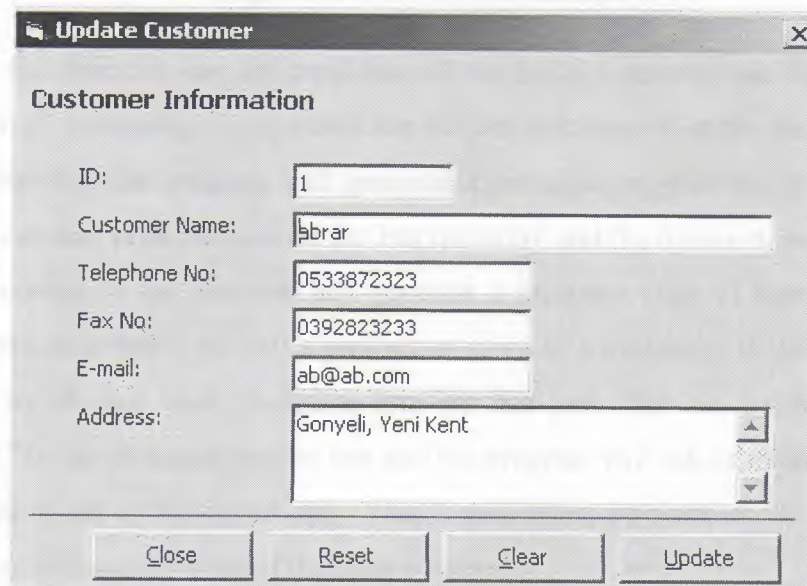
Edit Customer

Customer Name: abrar

Delete Edit Close

Figure 19 Edit Customer

This form shows us all the customers in the database and gives us the ability to edit or delete them.



Update Customer

Customer Information

ID: 1

Customer Name: abrar

Telephone No: 0533872323

Fax No: 0392823233

E-mail: ab@ab.com

Address: Gonyeli, Yeni Kent

Close Reset Clear Update

Figure 20 Update Customer

Sent Invoice

User Name: Customer Name: Date:

Product Name: Price: Quantity: Discount: Total Price:

Control:

Product Name:	Price:	Quantity:	Discount:	Total Price:
Charger	500.00	1	0%	500
Nokia 3310	30.00	2	0%	60

Figure 21 Sent Invoice Receipt

From this form we can add products and set the buy quantity and the discount, and then add them to ashopping list and save the invoice information in the database when we finish, and after that the program will generate a printable copy of the information. From this form we can add products and set the buy quantitiy and the discount, and then save the invoice information in the database and generate a printable copy of them. Note that this invoice is given only when we sell a product or more to a customer, if the cutomer is not defined and we do not want to define him we can just click on the list item "NEW CUSTOMER" in the customer combo box and the program will ask us to enter a temporary customer name to use in this report only .This is used when we want to sell a small amount of merchandise without the need of defining a customer.

Note too that the product prices are also automatically token from the products table in the database.

Also the user name will be saved and printed with the receipt for security reasons, so if a customer wanted to revise the mobiles shope companies we will know who did the sale and will know who had given a customer a big discount.

Received Invoice

User Name: Company Name: Date:

Product Name: Price: Quantity: Discount: Total Price:

Control:

Invoice List

Product Name:	Price:	Quantity:	Discount:	Total Price:
Nokia 5800 MusicXpress	500.00	1	0%	500
Sony Ericsson J210i	100.00	2	0%	200

Figure 22 Received Invoice Receipt

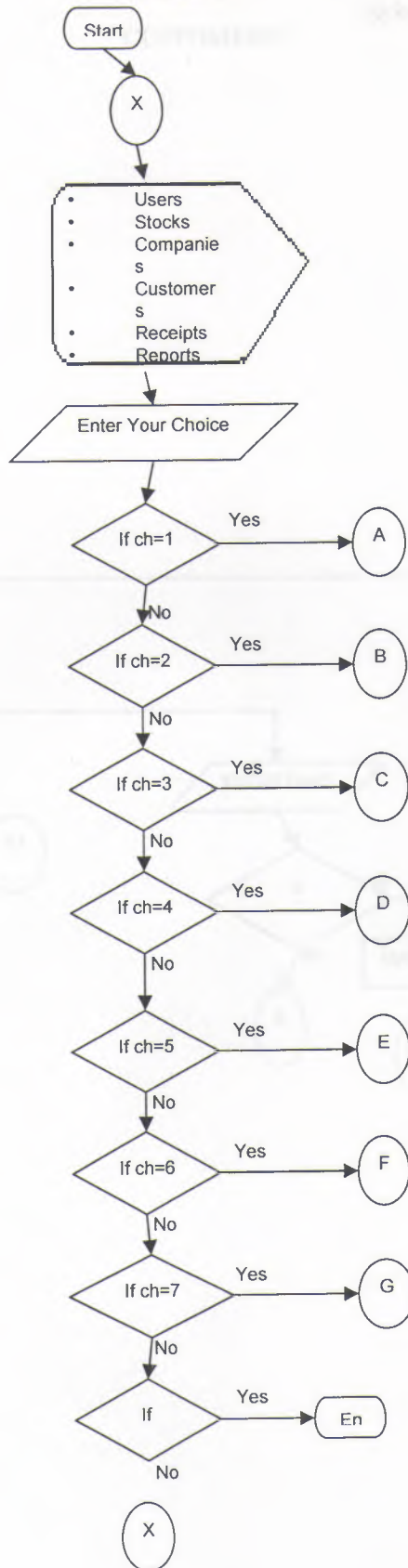
This invoice receipt is almost same as the sent invoice receipt but the only difference here is this receipt is given to the companies we buy merchandise from, and for that we do not select a customer name, we select a company's name.

The reports were made using Crystal Report which is a very unique program that reads the information from the database and shows it to us in a premade report.

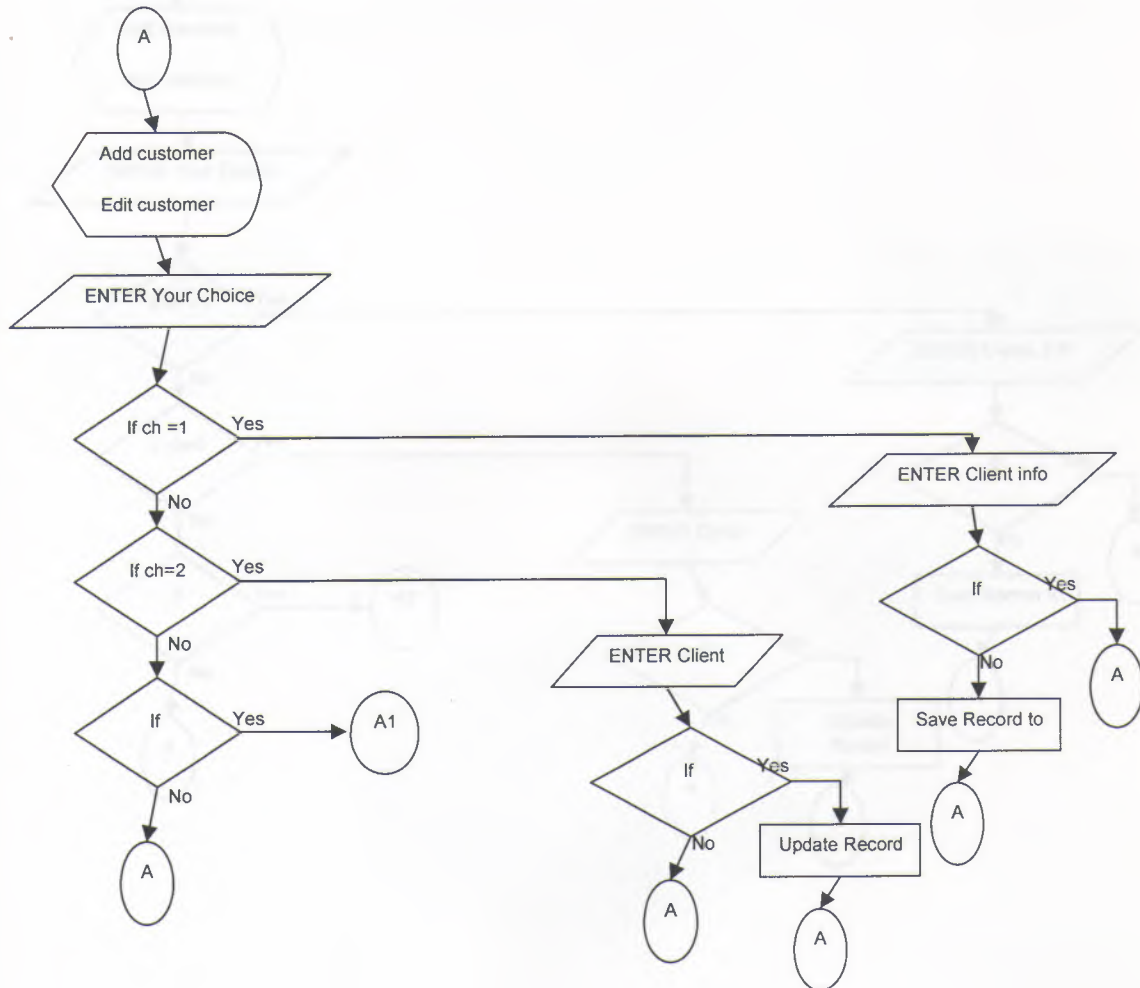
USER FLOWCHARTS



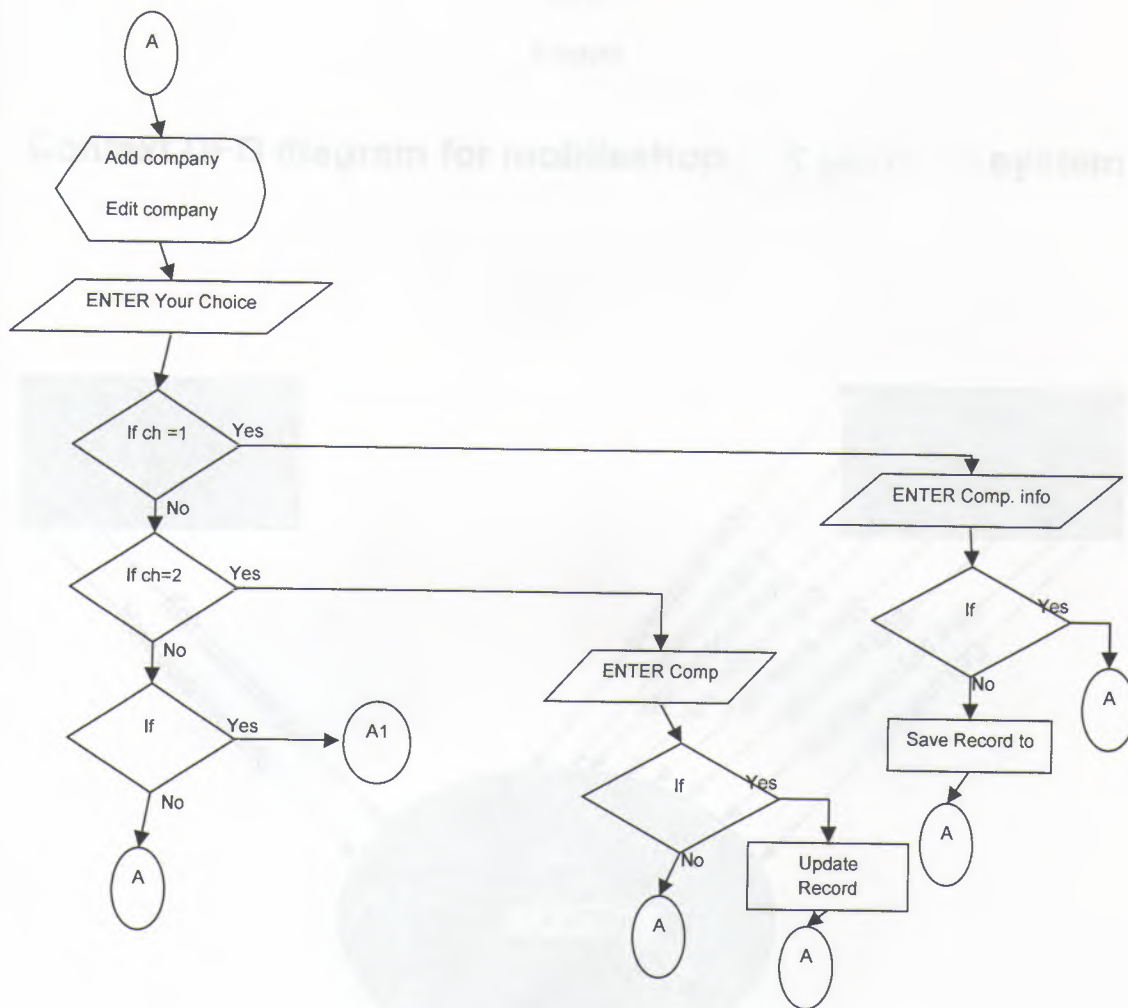
Main Menu



CUSTOMERS



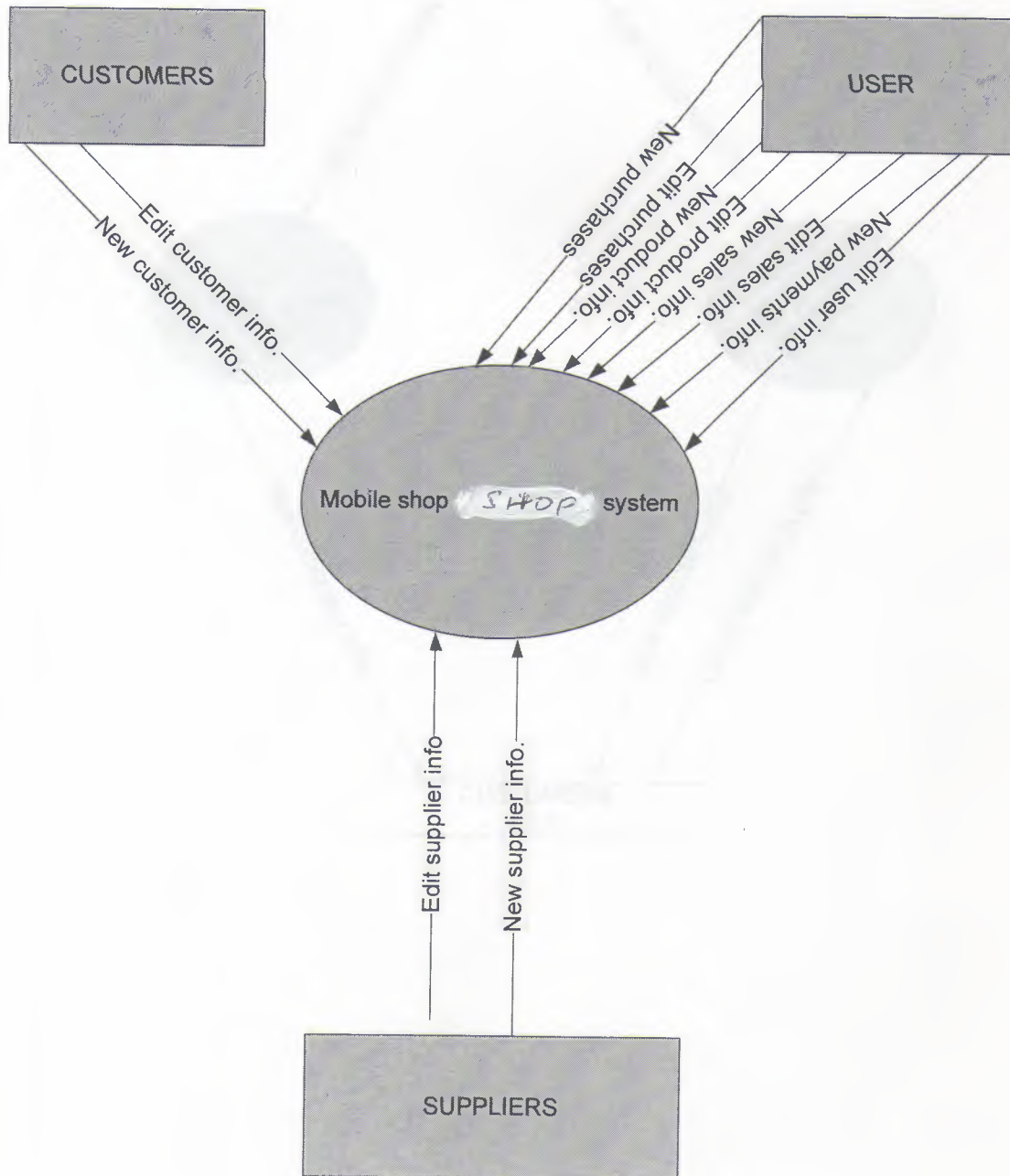
COMPANIES



DFD

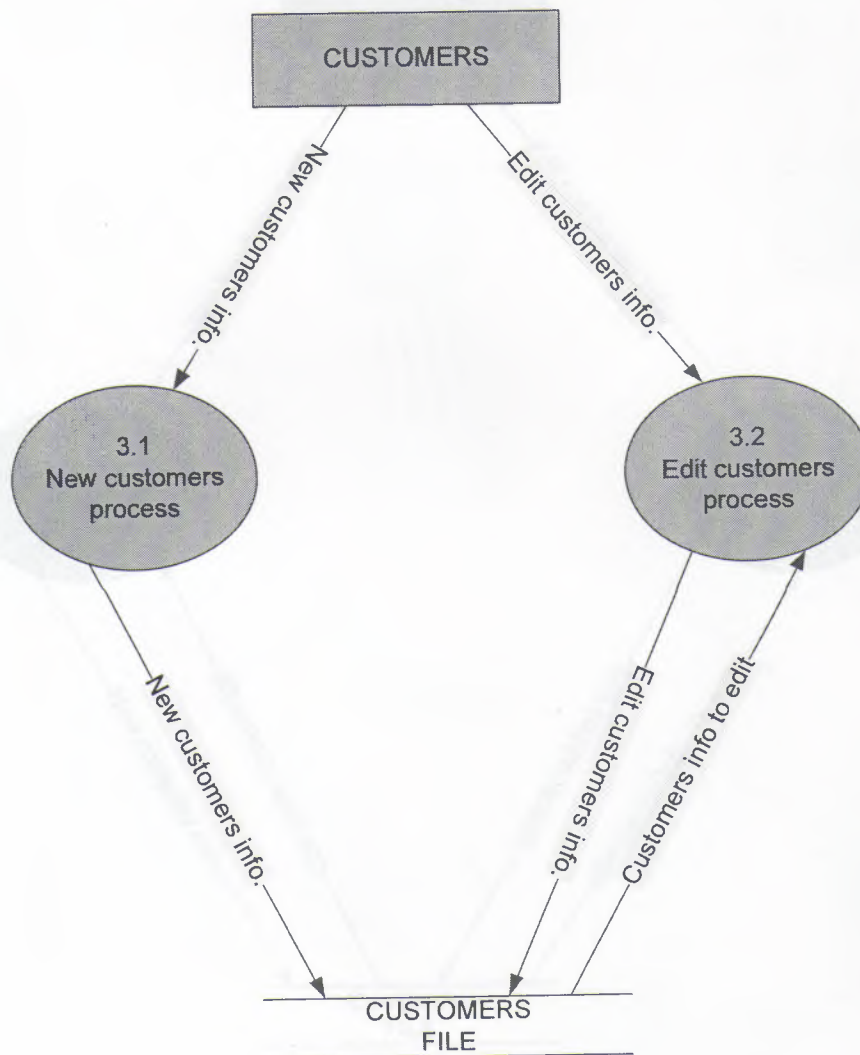
Context

Context DFD diagram for mobileshop *SHOP* system

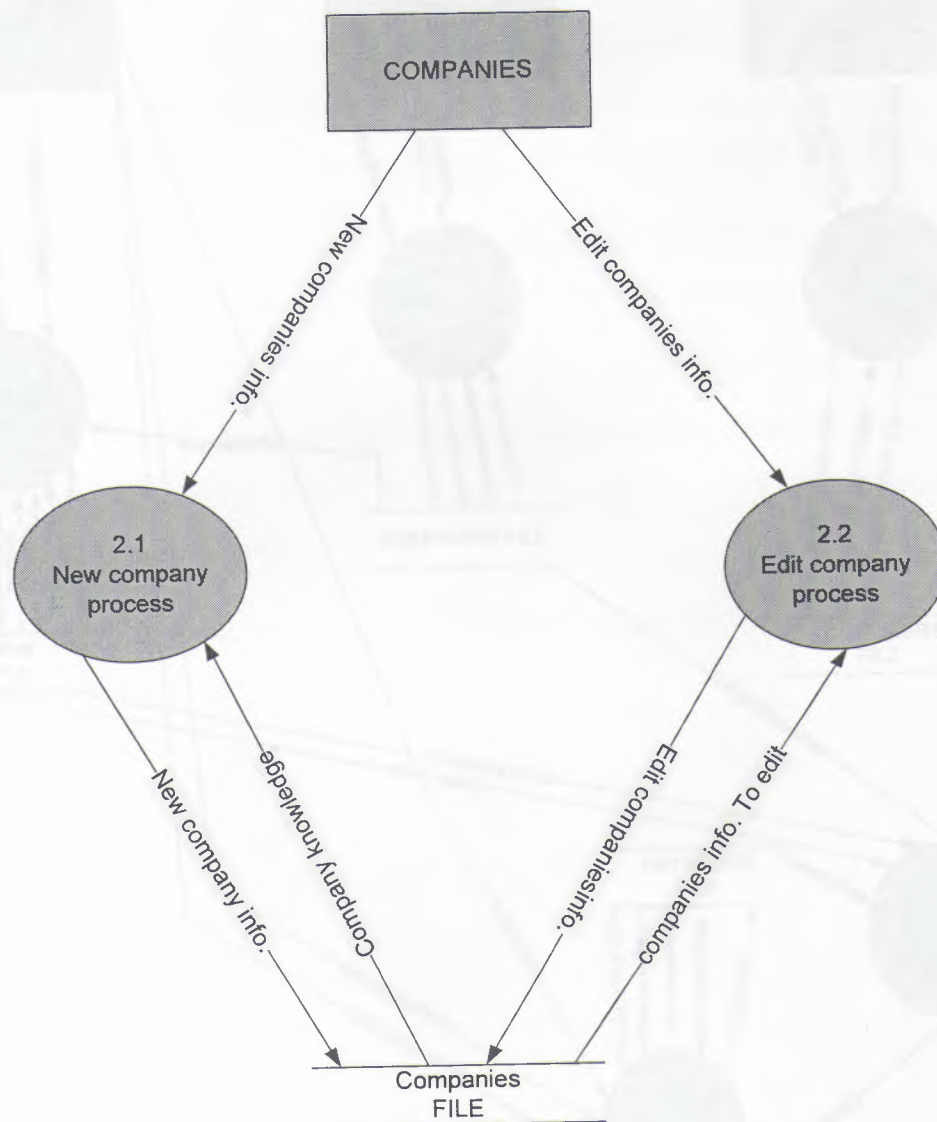


Customers detail

Detail DFD for customers

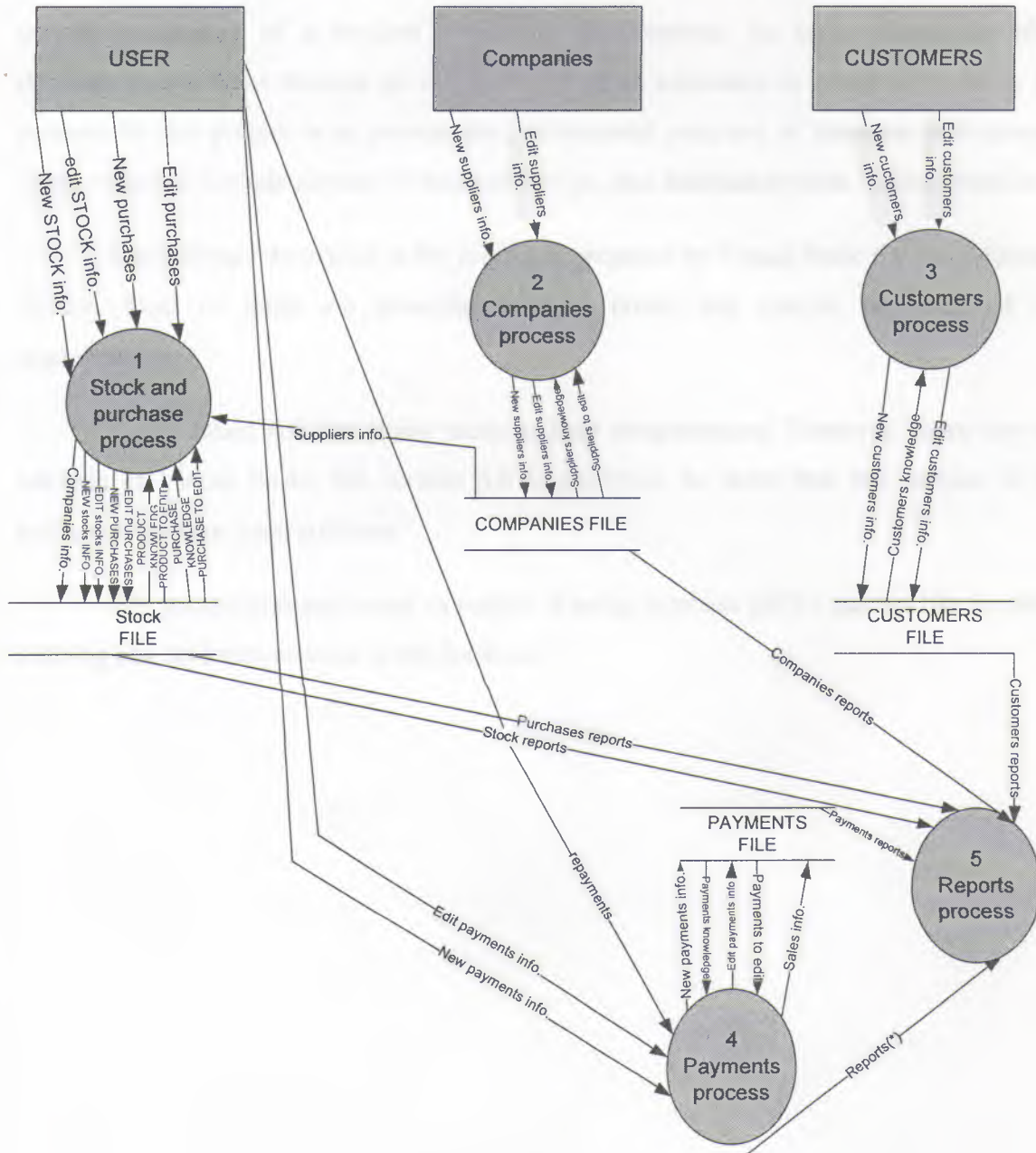


Detail DFD for companies



Top level DFD

Top_level DFD diagram for Mobile SHOP system



(*)
1. customers reports. 2. stock, purchases reports. 3. companies reports.
4. payments reports. 5. Report process.

CONCLUSION

Database management has evolved from a specialized computer application to a central component of a modern computing environment. As such, knowledge about database systems has become an essential part of an education in computer science. Our purpose in this project is to present the fundamental concepts of database management. These concepts include aspects of database design, and database-system implementation.

The software described in the project is prepared by Visual Basic 6.0 and Microsoft Access. Both of them are powerful tools to create and operate sophisticated data organizations.

Visual Basic 6.0 has many tools to help programmers. However, there are new editions of Visual Basic, the version 6.0 is preferred, to show that this version is also sufficient to make good software.

This project contains many examples of using database (SQL) queries like inserting, deleting and updating records in the database.

REFERENCES

BOOKS: Microsoft Visual Basic 2008 Step by Step

Author: Michael Halvorson

Publisher: Microsoft Press USA 2008

BOOKS: The Microsoft SQL Server 2000 Performance Optimization and Tuning Handbook

Author: Ken England

Publisher: Microsoft Certified Trainer 2000

WEBSITES

[1] <http://www.deitel.com/recourseCentres/Programming/VisualBasic/Default.asp>

[2] <http://freevbcode.com>

[3] Visual Basic 6 Black Book

[4] <http://www.vbzone.com>

[5] <http://www.vbpj.com>

APPENDIX

SOURCE CODE LISTINGS

MAIN MODULE

```
Option Explicit

Public DB As ADODB.Connection
Public RS As ADODB.Recordset
Public SQLstr As String
Public LastResp As String
Public LastID As Long
Public Const vbCancelResp = "&116"
Public MyUserName As String
Public Declare Sub InitCommonControls Lib "comctl32" ()

Public Sub Main()
    InitCommonControls
    Set DB = New ADODB.Connection
    DB.CursorLocation = adUseServ
    DB.Open ("provider=Microsoft.jet.oledb.3.51; Data Source=" & App.Path &
"\Database\system.mdb")

    frmSplash.Show
    While frmSplash.Visible = True
        DoEvents
    Wend

    frmLogin.Show
    While frmLogin.Visible = True
        DoEvents
    Wend

    frmMain.Show
End Sub

Public Function GetMaxID(tblName As String) As String
    Set RS = DB.Execute("select max(id) from " & tblName & ";")
    If RS.EOF Then
        GetMaxID = 1
    Else
        If IsNull(RS.Fields(0)) = True Then
            GetMaxID = 1
        End If
    End If
End Function
```



```

Else
    GetMaxID = RS.Fields(0) + 1
End If
End If
End Function

Public Sub SetComboByItemData(TheCombo As ComboBox, TheItemData As Long)
    Dim i As Integer
    For i = 0 To TheCombo.ListCount - 1
        If TheCombo.ItemData(i) = TheItemData Then
            TheCombo.ListIndex = i
        End If
    Next i
End Sub

Public Sub ShowMyReport(DaReportFileName As String, Optional RepSelForm As String = "")
    'On Error Resume Next
    With frmMain.CrystalReport1
        .Reset
        .ReportFileName = App.Path & "\Reports\" & DaReportFileName
        .Connect = App.Path & "\Database\System.mdb" & ";PWD=" & DeCODE(DBPASS)
        .DiscardSavedData = True
        .RetrieveDataFiles
        .ReportSource = 0
        .PrintFileType = crptCrystal
        .WindowMaxButton = True
        .WindowState = crptMaximized
        .ReplaceSelectionFormula (RepSelForm)
        .Action = 1
    End With
End Sub

Public Function MulRep$(X$)
    Dim t1$, t2$
    t1 = Replace$(X, ".", "")
    t2 = Replace$(t1, ",", ".")
    MulRep = t2
End Function

```

THE LOGIN FORM

Option Explicit

Private Sub cmdAccept_Click()

cmdAccept.Enabled = False

SQLstr = "select * from users where username='" & txtUser.Text & "' and password='" &
txtPass.Text & "'"

Set RS = DB.Execute(SQLstr)

If Not RS.EOF Then

If RS![UserName] = txtUser.Text And RS![Password] = txtPass.Text Then

On Error Resume Next

MyUserName = RS![UserName]

Unload Me

Else

MsgBox "Access Denied: Username or Password Error!!!", vbCritical, Me.Caption

End If

Else

MsgBox "Access Denied: Username or Password Error!!!", vbCritical, Me.Caption

End If

cmdAccept.Enabled = True

txtUser.SetFocus

End Sub

Private Sub cmdCancel_Click()

End

End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)

If Not UnloadMode = vbFormCode Then End

End Sub

Private Sub txtPass_GotFocus()

txtPass.SelStart = 0

txtPass.SelLength = Len(txtPass.Text)

End Sub

Private Sub txtUser_GotFocus()

txtUser.SelStart = 0

txtUser.SelLength = Len(txtUser.Text)

End Sub

Private Sub txtUser_KeyDown(KeyCode As Integer, Shift As Integer)

 If KeyCode = vbKeyReturn Then txtPass.SetFocus

End Sub

Private Sub txtPass_KeyDown(KeyCode As Integer, Shift As Integer)

 If KeyCode = vbKeyReturn Then Call cmdAccept_Click

End Sub

MAIN FORM

Dim i As Integer

Private Const My_Name = "Mobile Shop..."

Private Sub cmdCompanies_Click()

 frmComp.Show vbModal

End Sub

Private Sub cmdCustomers_Click()

 frmCust.Show vbModal

End Sub

Private Sub cmdExit_Click()

 Call mnuExit_Click

End Sub

Private Sub cmdRec_Click()

 frmSelRec.Show vbModal

End Sub

Private Sub cmdReports_Click()

 frmRepSel.Show vbModal

End Sub

Private Sub cmdStocks_Click()

 frmStocks.Show vbModal

End Sub

Private Sub cmdUsersID_Click()

```

    Call mnuUsersId_Click
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
    Cancel = True
    Call mnuExit_Click
End Sub

Private Sub Form_Resize()
    With Image1
        .Width = Me.Width
        .Height = Me.Height
    End With
End Sub

Private Sub Image1_Click()
End Sub

Private Sub mnuAbout_Click()
    MsgBox "Made By Omer Al-Aroui" & vbNewLine & "Near East University" & vbNewLine &
vbNewLine & "Nicosia-2008", vbInformation, "Euro-Cell"
End Sub

Private Sub mnuCompAdd_Click()
    frmCompAdd.Show vbModal
End Sub

Private Sub mnuCompRec_Click()
    frmRecInvoice.Show vbModal
End Sub

Private Sub mnuCompUpdate_Click()
    frmCompEdit.Show vbModal
End Sub

Private Sub mnuCustAdd_Click()
    frmCustAdd.Show vbModal
End Sub

```



```
Private Sub mnuCustUpdate_Click()
```

```
    frmCustEdit.Show vbModal
```

```
End Sub
```

```
Private Sub mnuRecCust_Click()
```

```
    frmSentInvoice.Show vbModal
```

```
End Sub
```

```
Private Sub mnuRepComp_Click()
```

```
    frmCompRep.Show vbModal
```

```
End Sub
```

```
Private Sub mnuRepProd_Click()
```

```
    frmProdRep.Show vbModal
```

```
End Sub
```

```
Private Sub mnuStockBrands_Click()
```

```
    frmStockBrands.Show vbModal
```

```
End Sub
```

```
Private Sub mnuStockGroups_Click()
```

```
    frmStockGroups.Show vbModal
```

```
End Sub
```

```
Private Sub mnuStocksAdd_Click()
```

```
    frmStockAdd.Show vbModal
```

```
End Sub
```

```
Private Sub mnuStockUpdate_Click()
```

```
    frmStockEdit.Show vbModal
```

```
End Sub
```

```
Private Sub mnuUsersId_Click()
```

```
    frmUsers.Show vbModal
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
    i = i + 1
```

```
    If i > Len(My_Name) Then
```

```
        Me.Caption = ""
```

```

        lbl.Caption = ""
    i = 0
Else
    Me.Caption = Left(My_Name, i)
    lbl.Caption = Left(My_Name, i)
End If
End Sub

Private Sub mnuExit_Click()
    If MsgBox("Are You Sure You Want To Exit???", vbQuestion + vbYesNo, Me.Caption) = vbYes
    Then End
End Sub

```

USER IDENTIFICATION FORM

```

Option Explicit

Private Sub cmdAdd_Click()
    If cmdAdd.Caption = "Add" Then
        Call FixControlsStat(0)
    Else
        Call FixControlsStat(2)
    End If
End Sub

Private Sub cmdDelete_Click()
    If lstUsers.ListIndex = -1 Or lstUsers.Enabled = False Then
        MsgBox "No User Selected!!!", vbCritical, Me.Caption
    Else
        If MsgBox("Are You Sure You Want To Delete The User " & lstUsers.List(lstUsers.ListIndex) &
        "" ?, vbQuestion + vbYesNo, Me.Caption) = vbYes Then
            DB.Execute "delete * from users where id=" & lstUsers.ItemData(lstUsers.ListIndex) & ";"
            MsgBox "Deleted Successfully!", vbInformation, Me.Caption
            Call RefList
            Call FixControlsStat(2)
        End If
    End If
End Sub

```



```
Private Sub cmdEdit_Click()
```

```
    If cmdEdit.Caption = "Edit" Then
```

```
        If lstUsers.ListIndex = -1 Or lstUsers.Enabled = False Then
```

```
            MsgBox "No User Selected!!!", vbCritical, Me.Caption
```

```
        Else
```

```
            Call FixControlsStat(1)
```

```
        End If
```

```
    Else
```

```
        Call FixControlsStat(2)
```

```
    End If
```

```
End Sub
```

```
Private Sub cmdfrmMain_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub cmdUpdate_Click()
```

```
    txt(4).Text = dtpDate.Value
```

```
    If txt(0).Text = "" Then MsgBox "Full Name Cannot Be Empty!", vbCritical, Me.Caption:  
    txt(0).SetFocus: Exit Sub
```

```
    If txt(1).Text = "" Then MsgBox "User Name Cannot Be Empty!", vbCritical, Me.Caption:  
    txt(1).SetFocus: Exit Sub
```

```
    If txt(2).Text = "" Then MsgBox "Password Cannot Be Empty!", vbCritical, Me.Caption:  
    txt(2).SetFocus: Exit Sub
```

```
    If cmdAdd.Enabled = True Then
```

```
        SQLstr = "insert into users values(" & GetMaxID("users") & ", " & txt(0).Text & ", " &  
        txt(1).Text & ", " & txt(2).Text & ", " & txt(4).Text & ", " & txt(5).Text & ", " & txt(6).Text & ", "  
        & txt(7).Text & ", " & txt(8).Text & ", " & txt(9).Text & ")"
```

```
        DB.Execute (SQLstr)
```

```
    Else
```

```
        SQLstr = "update users set Fullname=" & txt(0).Text & ", Username=" & txt(1).Text & ",  
        Password=" & txt(2).Text & ", BDate=" & txt(4).Text & ", BPlace=" & txt(5).Text & ", Address=" &  
        & txt(6).Text & ", TelNo=" & txt(7).Text & ", Nat=" & txt(8).Text & ", Email=" & txt(9).Text & "  
        where id=" & lstUsers.ItemData(lstUsers.ListIndex) & ";"
```

```
        DB.Execute (SQLstr)
```

```
    End If
```

```
    Call FixControlsStat(2)
```

```

    Call RefList
End Sub

```

```

Private Sub Form_Load()

```

```

    lstUsers.Clear
    Call RefList
    dtpDate.Value = Date

```

```

End Sub

```

```

Private Sub FixControlsStat(iStat As Integer)

```

```

    Dim i As Integer
    If iStat = 0 Then
        For i = 0 To 9
            If i = 3 Then i = 4
            txt(i).Text = ""
            txt(i).Locked = False
        Next i
        cmdAdd.Caption = "Cancel"
        cmdEdit.Enabled = False
        cmdDelete.Enabled = False
        cmdUpdate.Enabled = True
        lstUsers.Enabled = False
    
```

```

    ElseIf iStat = 1 Then

```

```

        For i = 0 To 9
            If i = 3 Then i = 4
            txt(i).Locked = False
        Next i

```

```

        cmdEdit.Caption = "Cancel"
        cmdAdd.Enabled = False
        cmdDelete.Enabled = False
        cmdUpdate.Enabled = True
        lstUsers.Enabled = False
    
```

```

    ElseIf iStat = 2 Then

```

```

        For i = 0 To 9
            If i = 3 Then i = 4
            txt(i).Text = ""
            txt(i).Locked = True
        Next i

```

```

        cmdAdd.Caption = "Add"
        cmdEdit.Caption = "Edit"
    
```



```

cmdAdd.Enabled = True
cmdEdit.Enabled = True
cmdDelete.Enabled = True
cmdUpdate.Enabled = False
lstUsers.Enabled = True
End If
End Sub

Private Sub lstUsers_Click()
    If lstUsers.Enabled = False Or lstUsers.ListCount = 0 Then Exit Sub
    Set RS = DB.Execute("select * from users where id=" & lstUsers.ItemData(lstUsers.ListIndex) &
";")
    txt(0).Text = RS.Fields(1)
    txt(1).Text = RS.Fields(2)
    txt(2).Text = RS.Fields(3)
    txt(4).Text = RS.Fields(4)
    txt(5).Text = RS.Fields(5)
    txt(6).Text = RS.Fields(6)
    txt(7).Text = RS.Fields(7)
    txt(8).Text = RS.Fields(8)
    txt(9).Text = RS.Fields(9)
    dtpDate.Value = txt(4).Text
    cmdEdit.Enabled = True
    cmdDelete.Enabled = True
End Sub

Private Sub RefList()
    SQLstr = "select distinct id, username from users order by username asc;"
    Set RS = DB.Execute(SQLstr)

    DoEvents
    lstUsers.Clear
    If Not RS.EOF Then
        While Not RS.EOF
            lstUsers.AddItem RS.Fields![UserName]
            lstUsers.ItemData(lstUsers.ListCount - 1) = RS![ID]
            RS.MoveNext
            DoEvents
        Wend
    End If
End Sub

```

End Sub

Private Sub txt_Change(Index As Integer)

End Sub

Private Sub txtSrch_Change()

Dim i As Integer

For i = 0 To lstUsers.ListCount - 1

If Left\$(lstUsers.List(i), Len(txtSrch.Text)) = txtSrch.Text Then

lstUsers.ListIndex = i

Exit Sub

End If

Next i

End Sub

STOCK FORM

Option Explicit

Private Sub cmdAdd_Click()

frmStockAdd.Show vbModal

End Sub

Private Sub cmdCancel_Click()

Unload Me

End Sub

Private Sub cmdStockBNames_Click()

frmStockBrands.Show vbModal

End Sub

Private Sub cmdUpdate_Click()

frmStockEdit.Show vbModal

End Sub

Private Sub cmdStockGr_Click()

frmStockGroups.Show vbModal

End Sub

Private Sub Form_Load()

End Sub

STOCKS ADD

Option Explicit

Private Sub cmdAdd_Click()

If comG.ListIndex = -1 Then comG.ListIndex = 0

If comB.ListIndex = -1 Then comB.ListIndex = 0

If txtStName.Text = "" Then MsgBox "Stock Name cannot be empty!", vbInformation, Me.Caption:
txtStName.SetFocus: Exit Sub

DB.Execute "INSERT INTO STOCKS VALUES(" & GetMaxID("STOCKS") & ", " & txtStName.Text & ", " & comG.ItemData(comG.ListIndex) & ", " & comB.ItemData(comB.ListIndex) & ", " & txtModel.Text & ", " & txtMan.Text & ", " & Val(Replace\$(txtPrice.Text, ",", "")) & ", " & chkC.Value & ", " & chkD.Value & ", " & chkB.Value & ", " & chkW.Value & ");"

If MsgBox("Added Successfully.." & vbNewLine & "Add Another?", vbYesNo + vbQuestion, Me.Caption) = vbYes Then

Call cmdClear_Click

Else

Unload Me

End If

End Sub

Private Sub cmdClear_Click()

Dim CT As Control

For Each CT In Me

If TypeOf CT Is TextBox Then

CT.Text = ""

ElseIf TypeOf CT Is ComboBox Then

CT.ListIndex = -1

```

ElseIf TypeOf CT Is CheckBox Then
    CT.Value = vbUnchecked
End If
Next CT
txtID.Text = GetMaxID("STOCKS")
txtStName.SetFocus
End Sub

Private Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    Set RS = DB.Execute("SELECT * FROM STOCK_GROUPS ORDER BY GNAME ASC;")

    comG.AddItem "No Group"
    comG.ItemData(comG.ListCount - 1) = 0
    While Not RS.EOF
        If Not IsNull(RS.Fields("GNAME")) Then
            comG.AddItem RS.Fields("GNAME")
            comG.ItemData(comG.ListCount - 1) = RS.Fields("ID")
        End If
        RS.MoveNext
    Wend

    Set RS = DB.Execute("SELECT * FROM STOCK_BRANDS ORDER BY BNAME ASC;")

    comB.AddItem "No Brand Name"
    comB.ItemData(comB.ListCount - 1) = 0
    While Not RS.EOF
        If Not IsNull(RS.Fields("BNAME")) Then
            comB.AddItem RS.Fields("BNAME")
            comB.ItemData(comB.ListCount - 1) = RS.Fields("ID")
        End If
        RS.MoveNext
    Wend

    txtID.Text = GetMaxID("STOCKS")
End Sub

Private Sub txtPrice_LostFocus()

```

```
txtPrice.Text = Format$(txtPrice.Text, "###,###,##0.00")
```

```
End Sub
```

STOCK BRAND NAMES

Option Explicit

```
Private Sub cmdAct_Click(Index As Integer)
```

```
    Select Case Index
```

```
        Case Is = 0
```

```
            Dim MyAns As String
```

```
            MyAns = InputBox("Enter The New Stock Brand Name Name:", Me.Caption)
```

```
            If MyAns = CStr(vbCancel) Or MyAns = "" Then Exit Sub
```

```
            SQLstr = "insert into stock_brands values(" & GetMaxID("stock_brands") & ", " & MyAns & ")"
```

```
            DB.Execute (SQLstr)
```

```
            MsgBox "Added Successfully!", vbInformation, Me.Caption
```

```
        Case Is = 1
```

```
            If lst.ListIndex = -1 Or lst.Enabled = False Then
```

```
                MsgBox "No Stock Brand Name Selected!!!", vbCritical, Me.Caption
```

```
            Else
```

```
                Dim MyAns2 As String
```

```
                MyAns2 = InputBox("Enter The New Stock Brand Name Name:", Me.Caption,
```

```
lst.List(lst.ListIndex))
```

```
                If MyAns2 = CStr(vbCancel) Or MyAns2 = "" Then Exit Sub
```

```
                SQLstr = "update stock_brands set BName=" & MyAns2 & " where ID=" &  
lst.ItemData(lst.ListIndex) & ";"
```

```
                DB.Execute (SQLstr)
```

```
                MsgBox "Updated Successfully!", vbInformation, Me.Caption
```

```
            End If
```

```
        Case Is = 2
```

```
            If lst.ListIndex = -1 Or lst.Enabled = False Then
```

```
                MsgBox "No Stock Brand Name Selected!!!", vbCritical, Me.Caption
```

```
            Else
```

```
                If MsgBox("Are You Sure You Want To Delete The Stock Brand Name " &  
lst.List(lst.ListIndex) & " ?", vbQuestion + vbYesNo, Me.Caption) = vbYes Then
```



```
Private Sub DB.Execute "delete * from stock_brands where id=" & lst.ItemData(lst.ListIndex) & ";
```

```
MsgBox "Deleted Successfully!", vbInformation, Me.Caption
```

```
Call RefList
```

```
End If
```

```
End If
```

```
Case Is = 3
```

```
Unload Me
```

```
Exit Sub
```

```
End Select
```

```
Call RefList
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
MsgBox "Successfully!", vbInformation, Me.Caption
```

```
lst.Clear
```

```
Call RefList
```

```
End Sub
```

```
End Sub
```

```
Private Sub RefList()
```

```
SQLstr = "select * from stock_brands order by BName asc;"
```

```
Set RS = DB.Execute(SQLstr)
```

```
MsgBox "Successfully!", vbInformation, Me.Caption
```

```
lst.Clear
```

```
While Not RS.EOF
```

```
lst.AddItem RS.Fields("BName")
```

```
lst.ItemData(lst.ListCount - 1) = RS.Fields("ID")
```

```
RS.MoveNext
```

```
DoEvents
```

```
Wend
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Private Sub lst_Click()
```

```
lst.Clear
```

```
End Sub
```

```
End Sub
```

```
End Sub
```

```
End Sub
```

```
STOCK EDIT FORM
```

```
MsgBox "The Stock Add Stock before you edit", vbInformation, Me.Caption
```

```
Option Explicit
```

```
Dim
```

```
Private Sub cmdClose_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub cmdDel_Click()
```

```
    If lst.ListIndex = -1 Then
```

```
        MsgBox "No Stock Name Selected!" & vbNewLine & "Select A Stock First!", vbInformation,  
Me.Caption
```

```
    Else
```

```
        If MsgBox("Are You Sure You Want To Delete The Stock '" & lst.List(lst.ListIndex) & "' ?",  
vbYesNo + vbQuestion) = vbYes Then
```

```
            DB.Execute "DELETE * FROM STOCKS WHERE ID=" & lst.ItemData(lst.ListIndex)
```

```
            MsgBox "Deleted Successfully!", vbInformation, Me.Caption
```

```
        End If
```

```
    End If
```

```
    Call Form_Load
```

```
End Sub
```

```
Private Sub cmdEdit_Click()
```

```
    If lst.ListIndex = -1 Then
```

```
        MsgBox "No Stock Name Selected!" & vbNewLine & "Select A Stock First!", vbInformation,  
Me.Caption
```

```
    Else
```

```
        LastID = lst.ItemData(lst.ListIndex)
```

```
        frmStockUpdate.Show vbModal
```

```
    Call Form_Load
```

```
    End If
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Set RS = DB.Execute("SELECT ID, STNAME FROM STOCKS ORDER BY STNAME ASC;")
```

```
    lst.Clear
```

```
    If RS.EOF Then
```

```
        lst.Enabled = False
```

```
        cmdEdit.Enabled = False
```

```
        cmdDel.Enabled = False
```

```
        lst.AddItem "No Stock Names..."
```

```
        MsgBox "No Stocks, Add Stocks before you edit.", vbInformation, Me.Caption
```

```
        Exit Sub
```

```
    Else
```

```

While Not RS.EOF
    If Not IsNull(RS.Fields("STNAME")) Then
        lst.AddItem RS.Fields("STNAME")
        lst.ItemData(lst.ListCount - 1) = RS.Fields("ID")
    End If
    RS.MoveNext
Wend
End If
End Sub

```

```

Private Sub lst_Click()
    MsgBox "No Stock Group Selected!", vbCritical, Me.Caption
End Sub

```

```

If MsgBox("Are You Sure You Want To Enter The Stock Group " & lst.ItemData(lst.SelectedIndex) &
    "?", vbQuestion + vbYesNo, Me.Caption) = vbYes Then
    Dim SQLstr As String
    SQLstr = "insert into stock_groups values(" & lst.ItemData(lst.SelectedIndex) & ", '" &
        Me.Fields("STNAME").Text & "', vbInformation, Me.Caption
    MsgBox "Added Successfully!", vbInformation, Me.Caption
End If

```

STOCK GROUPS FORM

Option Explicit

```
Private Sub cmdAct_Click(Index As Integer)
```

```
    Select Case Index
```

```
        Case Is = 0
```

```
            Dim MyAns As String
```

```
            MyAns = InputBox("Enter The New Stock Group Name:", Me.Caption)
```

```
            If MyAns = CStr(vbCancel) Or MyAns = "" Then Exit Sub
```

```
            SQLstr = "insert into stock_groups values(" & GetMaxID("stock_groups") & ", '" & MyAns &
```

```
            "')"

```

```
            DB.Execute (SQLstr)
```

```
            MsgBox "Added Successfully!", vbInformation, Me.Caption
```

```
        Case Is = 1
```

```
            If lst.ListIndex = -1 Or lst.Enabled = False Then
```

```
                MsgBox "No Stock Group Selected!!!", vbCritical, Me.Caption
```

```
            Else

```



```

    Dim MyAns2 As String
    MyAns2 = InputBox("Enter The New Stock Group Name:", Me.Caption, lst.List(lst.ListIndex))
    If MyAns2 = CStr(vbCancel) Or MyAns2 = "" Then Exit Sub
    SQLstr = "update stock_groups set GName='" & MyAns2 & "' where ID=" &
    lst.ItemData(lst.ListIndex) & ";"
    DB.Execute (SQLstr)
    MsgBox "Updated Successfully!", vbInformation, Me.Caption
End If

```

```

Case Is = 2
If lst.ListIndex = -1 Or lst.Enabled = False Then
    MsgBox "No Stock Group Selected!!!", vbCritical, Me.Caption
Else
    If MsgBox("Are You Sure You Want To Delete The Stock Group '" & lst.List(lst.ListIndex) &
    "' ?", vbQuestion + vbYesNo, Me.Caption) = vbYes Then
        DB.Execute "delete * from stock_groups where id=" & lst.ItemData(lst.ListIndex) & ";"
        MsgBox "Deleted Successfully!", vbInformation, Me.Caption
        Call RefList
    End If
End If
Case Is = 3
    Unload Me
    Exit Sub
End Select
    Call RefList
End Sub

```

```

Private Sub Form_Load()
    lst.Clear
    Call RefList
End Sub

```

```

Private Sub RefList()
    SQLstr = "select * from stock_groups order by gname asc;"
    Set RS = DB.Execute(SQLstr)

    lst.Clear
    While Not RS.EOF
        lst.AddItem RS.Fields("GNAME")
        lst.ItemData(lst.ListCount - 1) = RS.Fields("ID")
    End While

```

```

        RS.MoveNext
    DoEvents
Wend
End Sub

```

```

Private Sub lst_Click()
End Sub

```

STOCK UPDATE FORM

```
Option Explicit
```

```
Private Sub cmdUpdate_Click()
```

```

    If comG.ListIndex = -1 Then comG.ListIndex = 0
    If comB.ListIndex = -1 Then comB.ListIndex = 0

```

```

    If txtStName.Text = "" Then MsgBox "Stock Name cannot be empty!", vbInformation, Me.Caption:
    txtStName.SetFocus: Exit Sub

```

```

    DB.Execute "UPDATE STOCKS SET STNAME='" & txtStName.Text & "', GID='" &
    comG.ItemData(comG.ListIndex) & "', BID='" & comB.ItemData(comB.ListIndex) & "', Model='" &
    txtModel.Text & "', Man='" & txtMan.Text & "', Price='" & Val(Replace$(txtPrice.Text, ",", "")) & "',
    Colour='" & chkD.Value & "', Bluetooth='" & chkB.Value & "', WiFi='" & chkW.Value & "', Camera='"
    & chkC.Value & " WHERE ID='" & LastID & "',"

```

```
MsgBox "Updated Successfully..", vbInformation, Me.Caption
```

```
Unload Me
```

```
End Sub
```

```
Private Sub cmdClear_Click()
```

```
Dim CT As Control
```

```
For Each CT In Me
```

```
    If TypeOf CT Is TextBox Then
```

```
        CT.Text = ""
```

```
    ElseIf TypeOf CT Is ComboBox Then
```

```
        CT.ListIndex = -1
```

```
    ElseIf TypeOf CT Is CheckBox Then
```

```
        CT.Value = vbUnchecked
```

```

    End If
Next CT
txtID.Text = LastID
txtStName.SetFocus
End Sub

Private Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdReset_Click()
    Set RS = DB.Execute("SELECT * FROM STOCKS WHERE ID=" & LastID & ";")
    txtID.Text = LastID
    txtStName.Text = RS.Fields("STName")
    Call SetComboByItemData(comG, RS.Fields("GID"))
    Call SetComboByItemData(comB, RS.Fields("BID"))
    txtModel.Text = RS.Fields("Model")
    txtMan.Text = RS.Fields("Man")
    txtPrice.Text = Format$(RS.Fields("Price"), "###,###,##0.00")
    chkC.Value = RS.Fields("Camera")
    chkB.Value = RS.Fields("Bluetooth")
    chkD.Value = RS.Fields("Colour")
    chkW.Value = RS.Fields("WiFi")
End Sub

Private Sub Form_Load()
    Set RS = DB.Execute("SELECT * FROM STOCK_GROUPS ORDER BY GNAME ASC;")

    comG.AddItem "No Group"
    comG.ItemData(comG.ListCount - 1) = 0
    While Not RS.EOF
        If Not IsNull(RS.Fields("GNAME")) Then
            comG.AddItem RS.Fields("GNAME")
            comG.ItemData(comG.ListCount - 1) = RS.Fields("ID")
        End If
        RS.MoveNext
    Wend

    Set RS = DB.Execute("SELECT * FROM STOCK_BRANDS ORDER BY BNAME ASC;")

```



```

Private Sub cmdClear_Click()
    Dim CT As Control
    For Each CT In Me
        If TypeOf CT Is TextBox Then
            CT.Text = ""
        ElseIf TypeOf CT Is ComboBox Then
            CT.ListIndex = -1
        End If
    Next CT
    txtID.Text = GetMaxID("COMPANIES")
    txtCName.SetFocus
End Sub

```

```

Private Sub cmdClose_Click()
    Unload Me
End Sub

```

```

Private Sub Form_Load()
    txtID.Text = GetMaxID("COMPANIES")
End Sub

```

COMPANIES EDIT FORM

Option Explicit

```

Private Sub cmdClose_Click()
    Unload Me
End Sub

```

```

Private Sub cmdDel_Click()
    If lst.ListIndex = -1 Then
        MsgBox "No Company Name Selected!" & vbNewLine & "Select A Company Name First!",
vbInformation, Me.Caption
    Else
        If MsgBox("Are You Sure You Want To Delete The Company Name " & lst.List(lst.ListIndex) &
"" ?", vbYesNo + vbQuestion) = vbYes Then
            DB.Execute "DELETE * FROM COMPANIES WHERE ID=" & lst.ItemData(lst.ListIndex)
            MsgBox "Deleted Successfully!", vbInformation, Me.Caption
        End If
    End If
End Sub

```

```

    Call Form_Load
End Sub

Private Sub cmdEdit_Click()
    If lst.ListIndex = -1 Then
        MsgBox "No Company Name Selected!" & vbNewLine & "Select A Company Name First!",
vbInformation, Me.Caption
    Else
        LastID = lst.ItemData(lst.ListIndex)
        frmCompUpdate.Show vbModal
        Call Form_Load
    End If
End Sub

Private Sub Form_Load()
    Set RS = DB.Execute("SELECT ID, CNAME FROM COMPANIES ORDER BY CNAME ASC;")
    lst.Clear
    If RS.EOF Then
        lst.Enabled = False
        cmdEdit.Enabled = False
        cmdDel.Enabled = False
        lst.AddItem "No Company Names..."
        MsgBox "No Company Names, Add Companies before you delete.", vbInformation, Me.Caption
        Exit Sub
    Else
        While Not RS.EOF
            If Not IsNull(RS.Fields("CNAME")) Then
                lst.AddItem RS.Fields("CNAME")
                lst.ItemData(lst.ListCount - 1) = RS.Fields("ID")
            End If
            RS.MoveNext
        Wend
    End If
End Sub

Private Sub lst_Click()
End Sub

```


COMPANIES UPDATE FORM

Option Explicit

Private Sub cmdReset_Click()

Set RS = DB.Execute("SELECT * FROM COMPANIES WHERE ID=" & LastID & ";")

txtID.Text = LastID

txtCName.Text = RS.Fields("CName")

txtTel.Text = RS.Fields("TEL")

txtFax.Text = RS.Fields("FAX")

txtEmail.Text = RS.Fields("EMail")

txtAdd.Text = RS.Fields("ADDRESS")

End Sub

Private Sub cmdUpdate_Click()

If txtCName.Text = "" Then MsgBox "Company Name cannot be empty!", vbInformation,
Me.Caption: txtCName.SetFocus: Exit Sub

DB.Execute "UPDATE COMPANIES SET CNAME=" & txtCName.Text & ", TEL=" &
txtTel.Text & ", FAX=" & txtFax.Text & ", EMAIL=" & txtEmail.Text & ", ADDRESS=" &
txtAdd.Text & " WHERE ID=" & txtID.Text & ";"

MsgBox "Updated Successfully..", vbInformation, Me.Caption

Unload Me

End Sub

Private Sub cmdClear_Click()

Dim CT As Control

For Each CT In Me

If TypeOf CT Is TextBox Then

CT.Text = ""

ElseIf TypeOf CT Is ComboBox Then

CT.ListIndex = -1

End If

Next CT

txtID.Text = LastID

txtCName.SetFocus

End Sub

Private Sub cmdClose_Click()

```

    Unload Me
End Sub

Private Sub Form_Load()
    txtID.Text = LastID
    Call cmdReset_Click
End Sub

Private Sub txtCName_Change()

End Sub

CUSTOMER FORM

Option Explicit

Private Sub cmdAdd_Click()
    frmCustAdd.Show vbModal
End Sub

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdUpdate_Click()
    frmCustEdit.Show vbModal
End Sub

CUSTOMER ADD FORM

Option Explicit

Private Sub cmdAdd_Click()
    If txtCName.Text = "" Then MsgBox "Customer Name cannot be empty!", vbInformation,
Me.Caption: txtCName.SetFocus: Exit Sub

    DB.Execute "INSERT INTO CUSTOMERS VALUES(" & GetMaxID("CUSTOMERS") & ", " &
txtCName.Text & ", " & txtTel.Text & ", " & txtFax.Text & ", " & txtEmail.Text & ", " &

```

```

txtAdd.Text & "");"
    If MsgBox("Added Successfully.." & vbNewLine & "Add Another?", vbYesNo + vbQuestion,
Me.Caption) = vbYes Then
    Call cmdClear_Click()
    Else
        Unload Me
    End If
End Sub

Private Sub cmdClear_Click()
    Dim CT As Control
    For Each CT In Me
        If TypeOf CT Is TextBox Then
            CT.Text = ""
        ElseIf TypeOf CT Is ComboBox Then
            CT.ListIndex = -1
        End If
    Next CT
    txtID.Text = GetMaxID("CUSTOMERS")
    txtCName.SetFocus
End Sub

Private Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    txtID.Text = GetMaxID("CUSTOMERS")
End Sub

Private Sub txtCName_Change()
End Sub

```

CUSTOMER DELETE FORM

Option Explicit


```

Private Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdDel_Click()
    If lst.ListIndex = -1 Then
        MsgBox "No Customer Name Selected!" & vbNewLine & "Select A Customer Name First!",
vbInformation, Me.Caption
    Else
        If MsgBox("Are You Sure You Want To Delete The Customer Name '" & lst.List(lst.ListIndex) &
        "' ?", vbYesNo + vbQuestion) = vbYes Then
            DB.Execute "DELETE * FROM CUSTOMERS WHERE ID=" & lst.ItemData(lst.ListIndex)
            MsgBox "Deleted Successfully!", vbInformation, Me.Caption
        End If
    End If
    Call Form_Load
End Sub

Private Sub Form_Load()
    Set RS = DB.Execute("SELECT ID, CNAME FROM CUSTOMERS ORDER BY CNAME ASC;")
    lst.Clear
    If RS.EOF Then
        lst.Enabled = False
        cmdDel.Enabled = False
        lst.AddItem "No Customer Names..."
        MsgBox "No Customer Names, Add Customers before you edit.", vbInformation, Me.Caption
        Exit Sub
    Else
        While Not RS.EOF
            If Not IsNull(RS.Fields("CNAME")) Then
                lst.AddItem RS.Fields("CNAME")
                lst.ItemData(lst.ListCount - 1) = RS.Fields("ID")
            End If
            RS.MoveNext
        Wend
    End If
End Sub

Private Sub lst_Click()

```

End Sub

CUSTOMER EDIT FORM

Option Explicit

Private Sub cmdClose_Click()

 Unload Me

End Sub

Private Sub cmdEdit_Click()

 If lst.ListIndex = -1 Then

 MsgBox "No Customer Name Selected!" & vbNewLine & "Select A Customer Name First!",
vbInformation, Me.Caption

 Else

 LastID = lst.ItemData(lst.ListIndex)

 frmCustUpdate.Show vbModal

 Call Form_Load

 End If

End Sub

Private Sub cmdDel_Click()

 If lst.ListIndex = -1 Then

 MsgBox "No Customer Name Selected!" & vbNewLine & "Select A Customer Name First!",
vbInformation, Me.Caption

 Else

 If MsgBox("Are You Sure You Want To Delete The Customer Name '" & lst.List(lst.ListIndex) &
" ?", vbYesNo + vbQuestion) = vbYes Then

 DB.Execute "DELETE * FROM CUSTOMERS WHERE ID=" & lst.ItemData(lst.ListIndex)

 MsgBox "Deleted Successfully!", vbInformation, Me.Caption

 End If

 End If

 Call Form_Load

End Sub

Private Sub Form_Load()

 Set RS = DB.Execute("SELECT ID, CNAME FROM CUSTOMERS ORDER BY CNAME ASC;")

 lst.Clear

 If RS.EOF Then

```

    lst.Enabled = False
    cmdEdit.Enabled = False
    cmdDel.Enabled = False
    lst.AddItem "No Customer Names..."
    MsgBox "No Customer Names, Add Customers before you delete.", vbInformation, Me.Caption
    Exit Sub
Else
    While Not RS.EOF
        If Not IsNull(RS.Fields("CNAME")) Then
            lst.AddItem RS.Fields("CNAME")
            lst.ItemData(lst.ListCount - 1) = RS.Fields("ID")
        End If
        RS.MoveNext
    Wend
End If
End Sub

Private Sub lst_Click()
End Sub

```

CUSTOMER UPDATE FORM

Option Explicit

```

Private Sub cmdReset_Click()
    Set RS = DB.Execute("SELECT * FROM CUSTOMERS WHERE ID=" & LastID & ";")
    txtID.Text = LastID
    txtCName.Text = RS.Fields("CName")
    txtTel.Text = RS.Fields("TEL")
    txtFax.Text = RS.Fields("FAX")
    txtEmail.Text = RS.Fields("EMail")
    txtAdd.Text = RS.Fields("ADDRESS")
End Sub

Private Sub cmdUpdate_Click()
    If txtCName.Text = "" Then MsgBox "Customer Name cannot be empty!", vbInformation,
Me.Caption: txtCName.SetFocus: Exit Sub

```



```

frmCompRep.Show vbModal
End Sub

```

```

Private Sub cmdProdRep_Click()
    frmProdRep.Show vbModal
End Sub

```

PRODUCT REPORT FORM

Option Explicit

```

Private Sub cmdCancel_Click()

```

Unload Me

```

End Sub

```

```

Private Sub cmdShow_Click()

```

Screen.MousePointer = vbHourglass

```

Call ShowMyReport("rprtProd.rpt", "{rec_invoice.INVDATE} >= #" & dtpSTR.Value & "# And
{rec_invoice.INVDATE} <= #" & dtpEND.Value & "#")

```

Screen.MousePointer = vbDefault

```

End Sub

```

```

Private Sub Form_Load()

```

```

End Sub

```

COMPANIES REPORT FORM

Option Explicit

```

Private Sub cmdCancel_Click()

```

Unload Me

```

End Sub

```

```

Private Sub cmdShow_Click()

```

Screen.MousePointer = vbHourglass

```

Call ShowMyReport("rprtComp.rpt", "{rec_invoice.INVDATE} >= #" & dtpSTR.Value & "# And
{rec_invoice.INVDATE} <= #" & dtpEND.Value & "#")

```

```
Screen.MousePointer = vbDefault
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
End Sub
```

RECIEVED INVOICE FORM

```
Private Sub txtPrice_Change()
```

```
End Sub
```

```
Private Sub txtQuan_GotFocus()
```

```
txtQuan.SelStart = 0
```

```
txtQuan.SelLength = Len(txtPrice.Text)
```

```
End Sub
```

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

```
If KeyCode = vbKeyReturn Then txtDisc.SetFocus
```

```
End Sub
```

```
Private Sub UpdateTotal()
```

```
On Error Resume Next
```

```
Dim cPrice As Currency, lQuan As Long, iDisc As Double
```

```
cPrice = CCur(txtPrice.Text)
```

```
lQuan = CCur(txtQuan.Text)
```

```
iDisc = Left$(txtDisc.Text, Len(txtDisc.Text) - 1)
```

```
txtTot.Text = Format$(cPrice * lQuan - (cPrice * lQuan) * iDisc / 100)
```

```
End Sub
```

```
Private Sub txtQuan_LostFocus()
```

```
If txtQuan.Text = "" Then
```

```
txtQuan.Text = "0"
```

```
Else
```

```
If IsNumeric(txtQuan.Text) = False Then
```

```
MsgBox "Invalid Data!", vbCritical, Me.Caption
```

```
txtQuan.Text = "0"
```

```
Else
```

```

txtQuan.Text = Val(txtQuan.Text)
End If
End If
Call txtDisc_LostFocus
Call UpdateTotal
End Sub

```

SELECT RECEIPT SETTINGS FORM

Option Explicit

```
Private Sub cmdCancel_Click()
```

```
Unload Me
```

```
End Sub
```

```
Private Sub cmdCompRec_Click()
```

```
frmRecInvoice.Show vbModal
```

```
End Sub
```

```
Private Sub cmdCustRec_Click()
```

```
frmSentInvoice.Show vbModal
```

```
End Sub
```

SENT INVOICE FORM

Option Explicit

```
Private StockPriceArr() As Currency
```

```
Dim MyInvoiceID As Long
```

```
Private Sub cmdAdd_Click()
```

```
lst(0).AddItem combProd.List(combProd.ListIndex)
```

```
lst(1).AddItem txtPrice.Text
```

```
lst(2).AddItem txtQuan.Text
```

```
lst(3).AddItem txtDisc.Text
```

```
lst(4).AddItem txtTot.Text
```

```
Call cmdClear_Click
```

```
End Sub
```

```
Private Sub cmdClear_Click()
```



```

txtPrice.Text = "0"
txtQuan.Text = "0"
txtDisc.Text = "0%"
txtTot.Text = "0"
combProd.ListIndex = -1
combProd.SetFocus
End Sub

Private Sub cmdClose_Click()
    Unload Me
End Sub

Private Sub cmdDel_Click()
    If MsgBox("Are you sure you want to delete the selected line?", vbYesNo + vbQuestion,
Me.Caption) = vbYes Then
        Dim i As Integer, iListIndex As Integer
        iListIndex = lst(0).ListIndex
        For i = 0 To 4
            lst(i).RemoveItem iListIndex
        Next i
    End If
End Sub

Private Sub cmdDelAll_Click()
    If MsgBox("Are you sure you want to delete all lines?", vbYesNo + vbQuestion, Me.Caption) =
vbYes Then
        Dim i As Integer
        For i = 0 To 4
            lst(i).Clear
        Next i
    End If
End Sub

Private Sub cmdSave_Click()
    If MsgBox("Are you sure you want to save?", vbYesNo + vbQuestion, Me.Caption) = vbYes Then
        MyInvoiceID = GetMaxID("SENT_INVOICE")
        Dim i As Integer
        For i = 0 To lst(0).ListCount - 1
            DB.Execute "INSERT INTO SENT_INVOICE VALUES(" & MyInvoiceID & _
                ", " & lst(0).List(i) & _

```

```

        ", " & MulRep$(lst(1).List(i)) & _
        ", " & MulRep$(lst(2).List(i)) & _
        ", " & MulRep$(Left$(lst(3).List(i), Len(lst(3).List(i)) - 1)) & _
        ", " & MulRep$(lst(4).List(i)) & _
        ", " & txtUser.Text & _
        ", " & combCust.List(combCust.ListIndex) & _
        ", " & Format$(Now, "DD/MM/YYYY") & _
        ");"

    DoEvents
Next i
If MsgBox("Successfully Saved!, Show Report?", vbYesNo + vbInformation, Me.Caption) =
vbYes Then
    Call ShowMyReport("rprtRecieptSent.rpt", "{REC_INVOICE.ID}=" & MyInvoiceID)
End If
Unload Me
End If
End Sub

Private Sub combCust_Click()
    If combCust.ListIndex = combCust.ListCount - 1 Then
        Dim X As String
        X = InputBox("Enter The New Customer's Name:", "New Customer")
        If Not X = "" Then
            combCust.RemoveItem (combCust.ListCount - 1)
            combCust.AddItem X
            combCust.AddItem "New Customer"
            combCust.ListIndex = combCust.ListCount - 2
        End If
    End If
End Sub

Private Sub combProd_Click()
    If combProd.ListIndex = -1 Then Exit Sub
    txtPrice.Text = Format$(StockPriceArr(combProd.ListIndex), "###,###,##0.00")
End Sub

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

```

```

Private Sub Form_Load()
    txtUser.Text = MyUserName
    txtDate.Text = Format$(Date, "DD/MM/YYYY")

    Set RS = DB.Execute("SELECT CNAME FROM CUSTOMERS ORDER BY CNAME ASC;")
    combCust.Clear
    If RS.EOF = False Then
        While Not RS.EOF
            If Not IsNull(RS.Fields("CNAME")) Then
                combCust.AddItem RS.Fields("CNAME")
            End If
            RS.MoveNext
        Wend
    End If
    combCust.AddItem "New Customer"

    Dim lStockCount As Long
    Set RS = DB.Execute("SELECT COUNT(STNAME) AS STCOUNT FROM STOCKS;")
    lStockCount = RS.Fields("STCOUNT")

    ReDim StockPriceArr(0 To lStockCount) As Currency
    Dim lCounter As Long
    lCounter = 0

    Set RS = DB.Execute("SELECT STNAME, PRICE FROM STOCKS ORDER BY STNAME
ASC;")
    combProd.Clear
    If RS.EOF = False Then
        While Not RS.EOF
            If Not IsNull(RS.Fields("STNAME")) Then
                combProd.AddItem RS.Fields("STNAME")
                StockPriceArr(lCounter) = RS.Fields("PRICE")
                lCounter = lCounter + 1
            End If
            RS.MoveNext
        Wend
    End If
End Sub

Private Sub txtDisc_GotFocus()

```



```

txtDisc.SelStart = 0
txtDisc.SelLength = Len(txtPrice.Text)
End Sub

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

Private Sub txtDisc_LostFocus()
    If txtDisc.Text = "" Then
        txtDisc.Text = "0%"
    ElseIf InStr(txtDisc.Text, "%") > 0 And Len(txtDisc.Text) > 1 Then
        txtDisc.Text = Left$(txtDisc.Text, Len(txtDisc.Text) - 1) & "%"
    Else
        txtDisc.Text = txtDisc.Text & "%"
    End If
    Call UpdateTotal
End Sub

Private Sub txtQuan_GotFocus()
    txtQuan.SelStart = 0
    txtQuan.SelLength = Len(txtPrice.Text)
End Sub

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

Private Sub UpdateTotal()
    On Error Resume Next
    Dim cPrice As Currency, lQuan As Long, iDisc As Double
    cPrice = CCur(txtPrice.Text)
    lQuan = CCur(txtQuan.Text)
    iDisc = Left$(txtDisc.Text, Len(txtDisc.Text) - 1)
    txtTot.Text = Format$(cPrice * lQuan - (cPrice * lQuan) * iDisc / 100)
End Sub

Private Sub txtQuan_LostFocus()
    If txtQuan.Text = "" Then
        txtQuan.Text = "0"
    End If
End Sub

```

```

Else
    If IsNumeric(txtQuan.Text) = False Then
        MsgBox "Invalid Data!", vbCritical, Me.Caption
        txtQuan.Text = "0"
    Else
        txtQuan.Text = Val(txtQuan.Text)
    End If
End If
Call txtDisc_LostFocus
Call UpdateTotal
End Sub

```

SPLASH FORM

Option Explicit

```

Private Sub Image1_Click()
    Unload Me
End Sub

```

```

Private Sub Timer1_Timer()
    If lbl.Width <= 7215 Then
        lbl.Width = lbl.Width + 40
        DoEvents
    Else
        Unload Me
    End If
End Sub

```

ABOUT FORM

Option Explicit

```

Private Sub cmdClose_Click()
    Unload Me
End Sub Private Sub imgBackground_Click() End S

```