**14. Test strategy**

Validation Test

Transaction Form

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Test Data | Data Type | Validation | Expectation |
| TID | T10005 | Normal | Format Check | Should be Accepted |
| T100057 | Abnormal | Show show Error Message |
| Transaction Date | 23/03/2013 | Normal | Format Check |  |
| 23/03/13 | Abnormal |  |

Customer Form

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Test Data | Data Type | Validation | Expectation |
| CID | C1006 | Normal | Format Check |  |
| C10068 | Abnormal |  |
| CustomerName | Zeba Farzana | Normal | Length Check |  |
| Fayez Ahmed Jahangir Masud | Abnormal |  |
| EmailID | akbar.ovi@yahoo.com | Normal | Format Check |  |
| akbarovi71 | Abnormal |  |
| BankName | Exim | Normal | Presence Check |  |
| " " | Abnormal |  |
| BankAcNum | 118799983 | Normal | Type Check |  |
| 11879a | Abnormal |  |

Generator Form

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Test Data | Data Type | Validation | Expectation |
| GID | G104 | Normal | Format Check |  |
| G1045 | Abnormal |  |
| Model | K5100ZA | Normal | Presence Check |  |
| " " | Abnormal |  |
| Stock | 17 | Normal | Range Check |  |
| -1 | Abnormal |  |
| 1, 50 | Extreme |  |

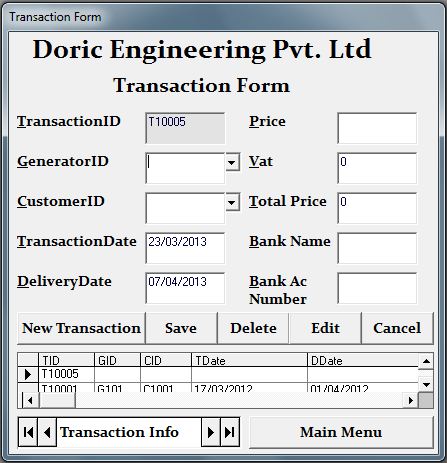


Fig 4.1

Fig 4.2

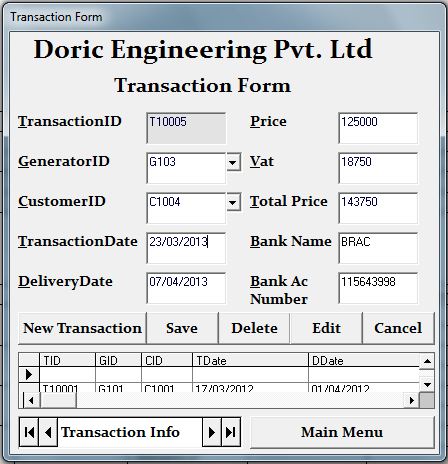
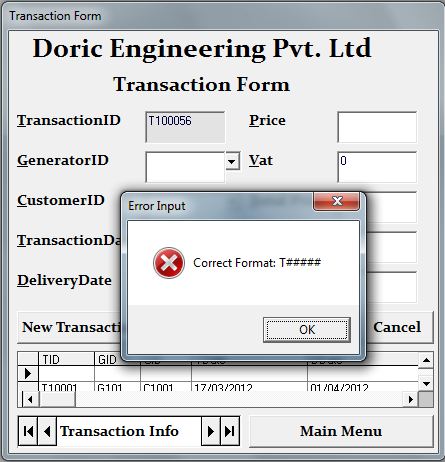


Fig 4.3

Fig 4.4

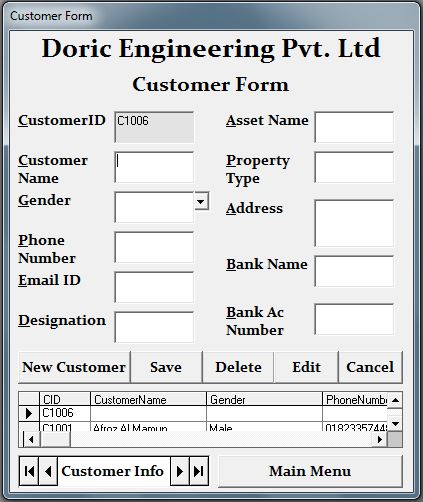
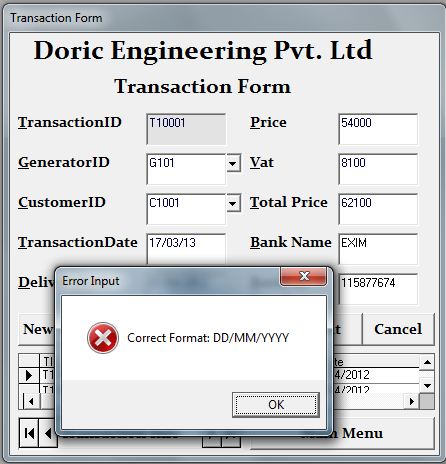


Fig 4.5

Fig 4.6

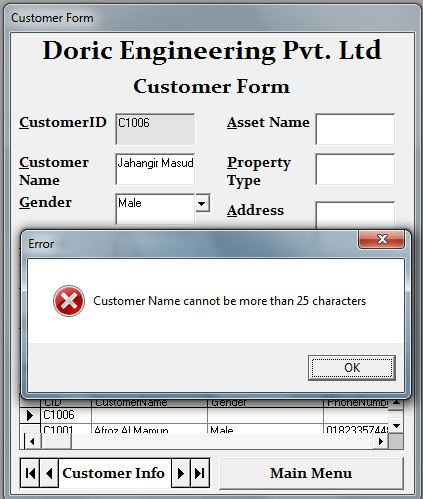
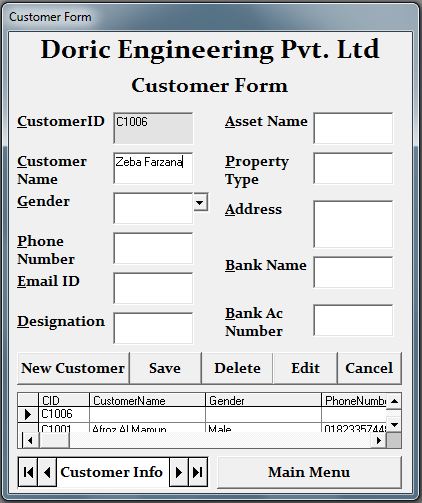
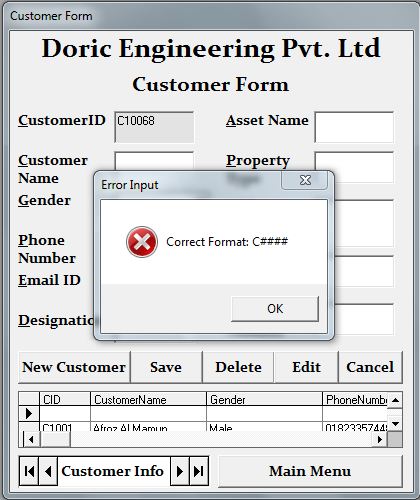


Fig 4.7

Fig 4.8

Fig 4.9

Fig 4.10

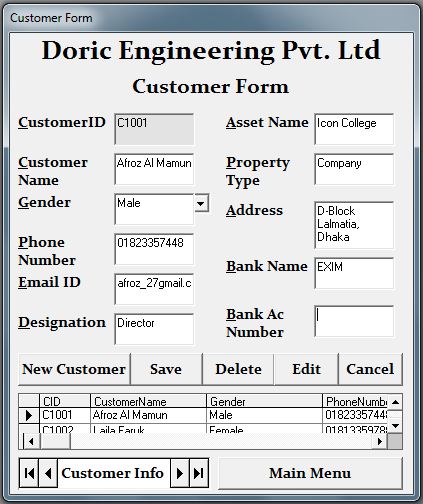
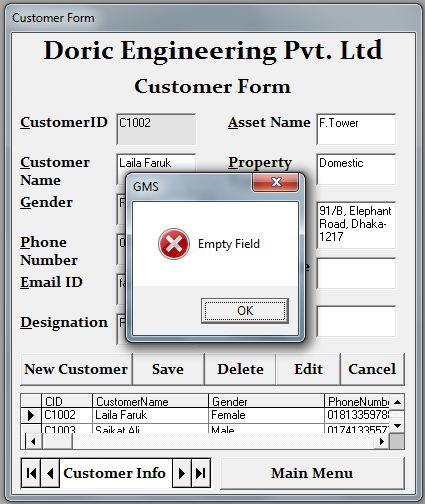
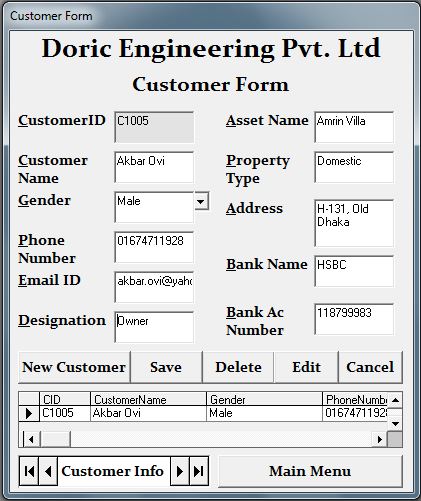
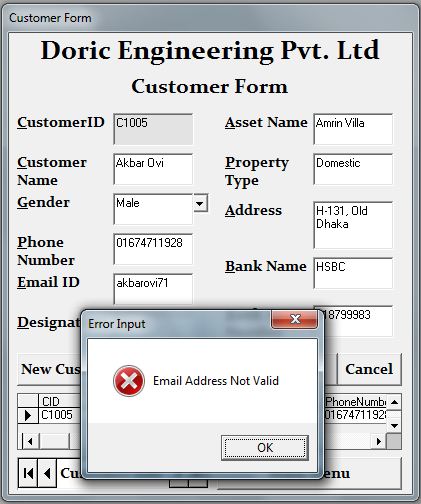


Fig 4.11

Fig 4.12

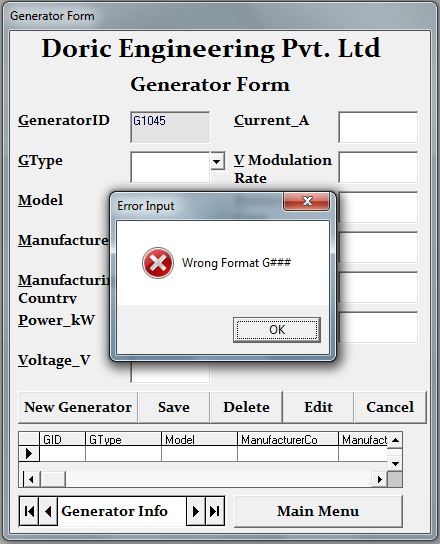
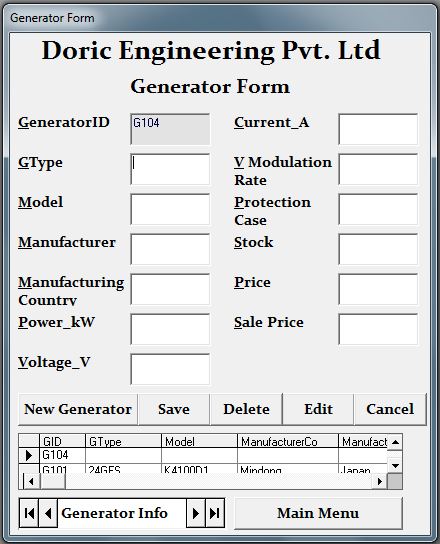


Fig 4.13

Fig 4.14

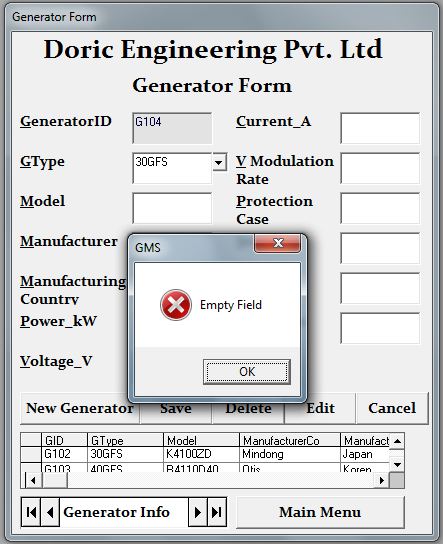
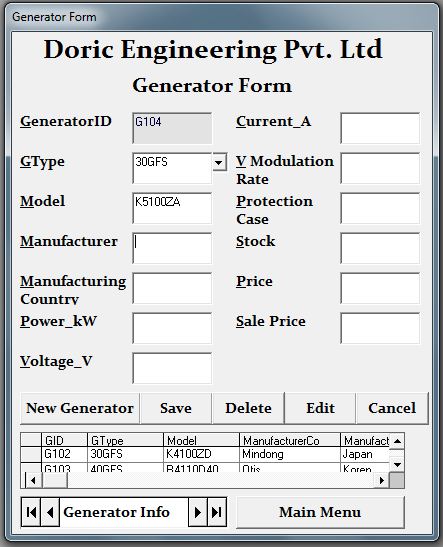


Fig 4.15

Fig 4.16

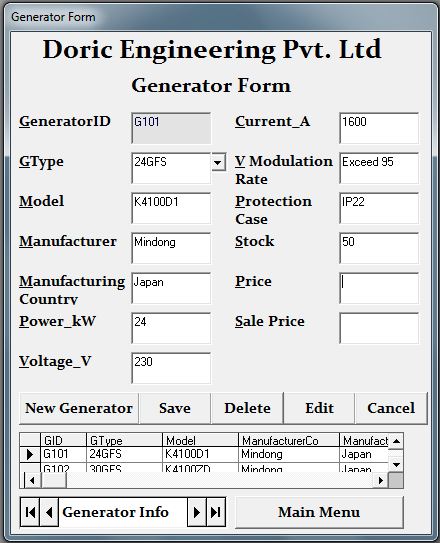
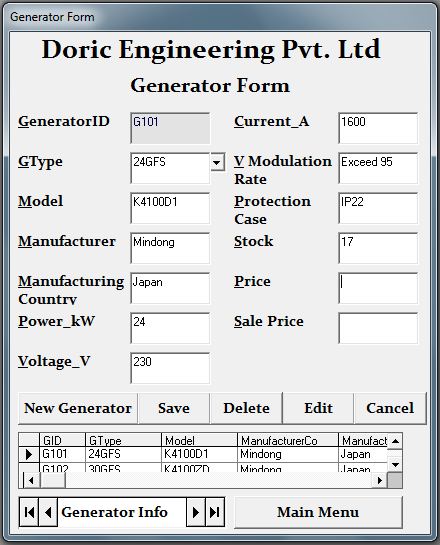
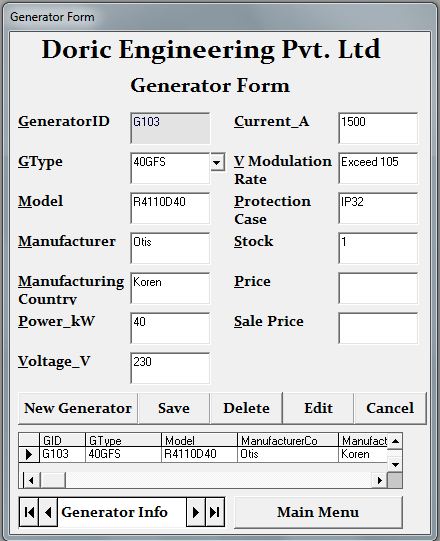
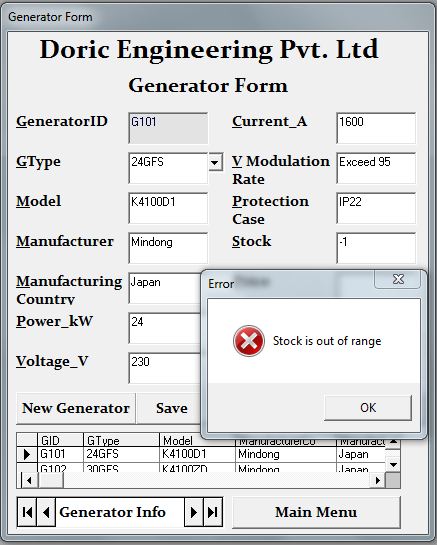


Fig 4.17

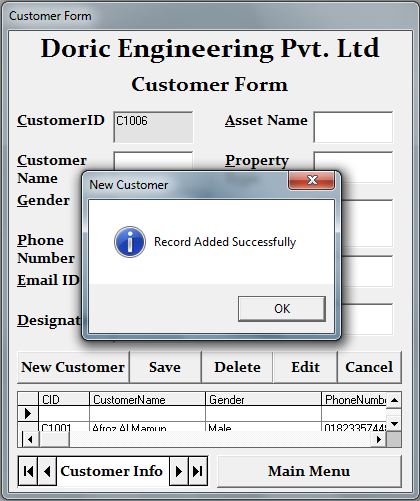
Fig 4.18

Fig 4.19

Fig 4.20

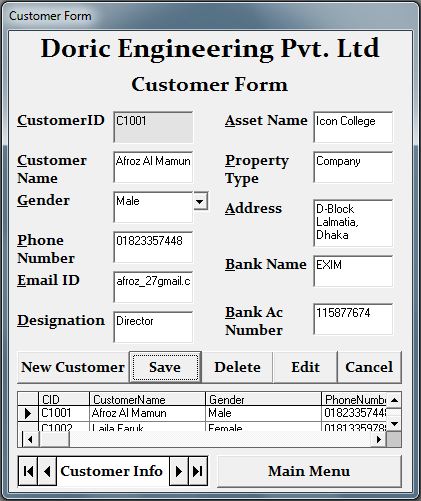
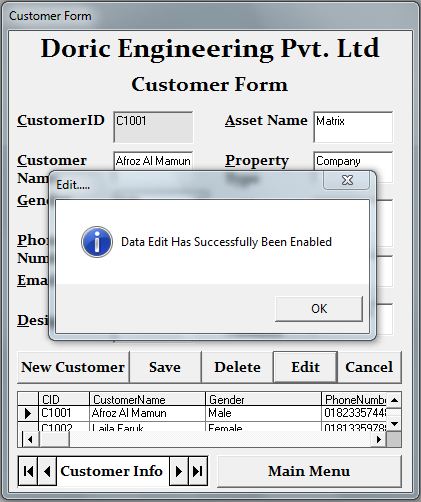


Data Manipulation Testing

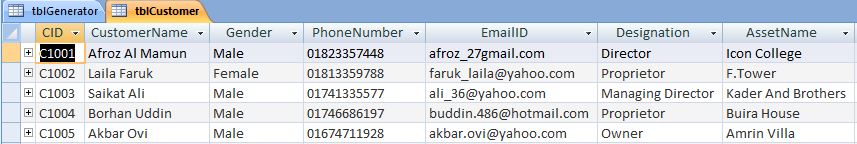


Auto generated ID

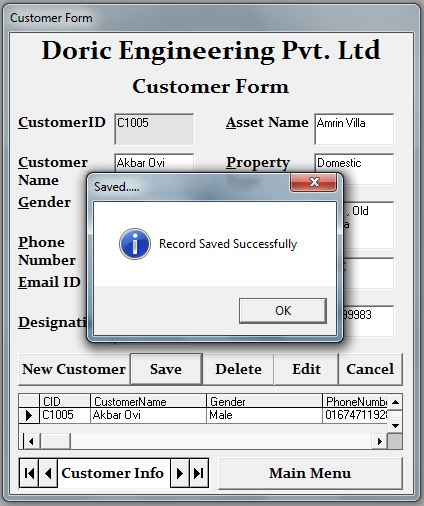
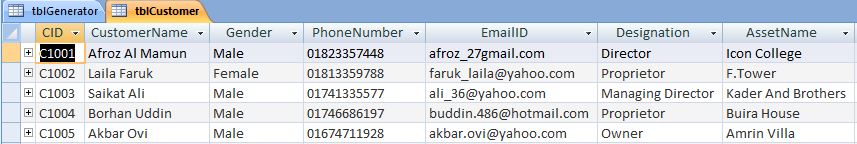
New blank record added



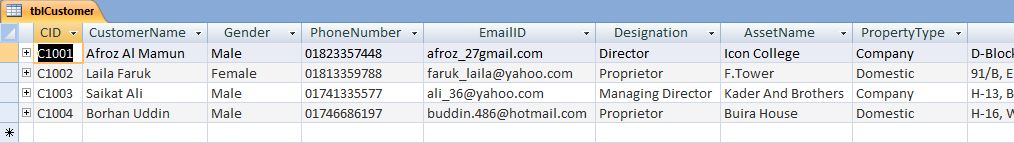
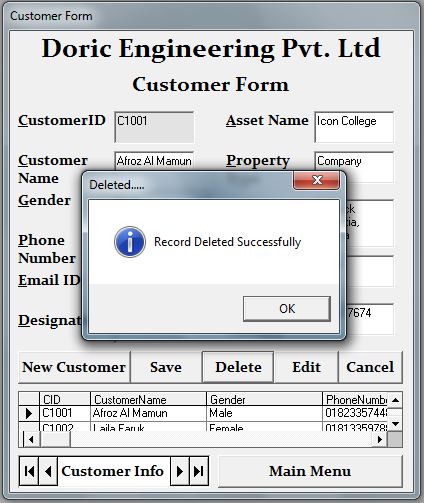
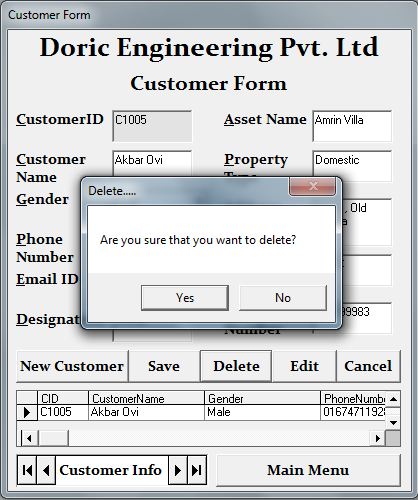
Edited Field



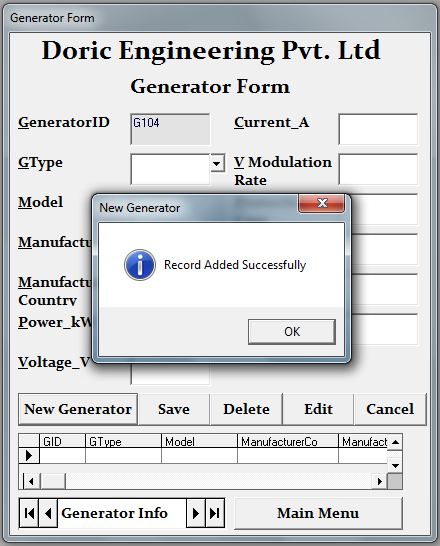
Field Edited in Database



Record Saved into Database

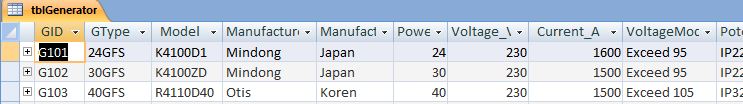
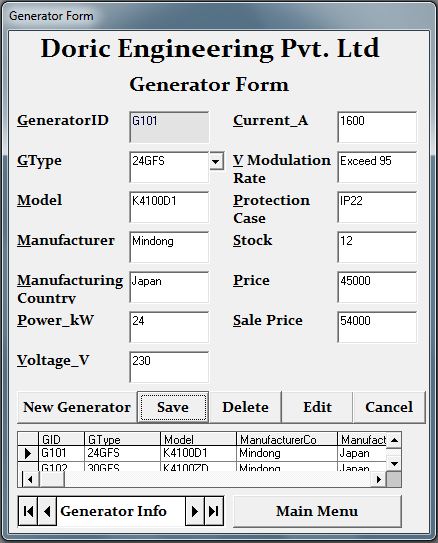
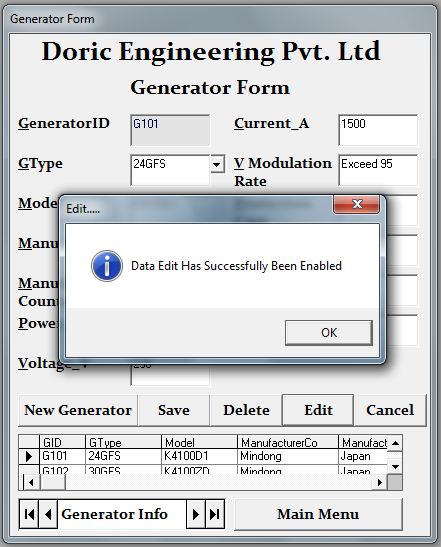


Record Deleted from Database



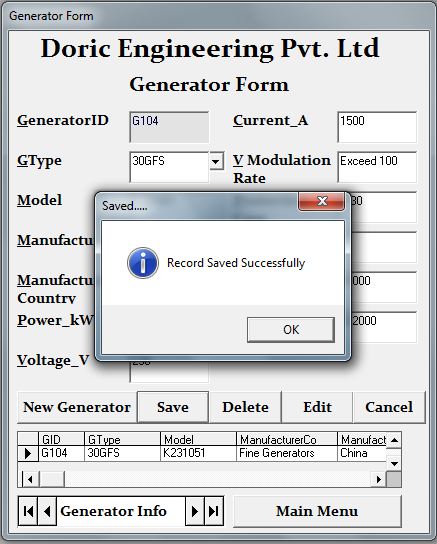
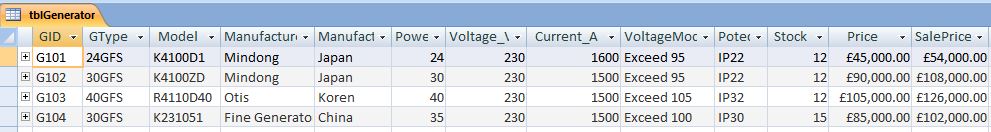
Auto generated ID

New blank record added

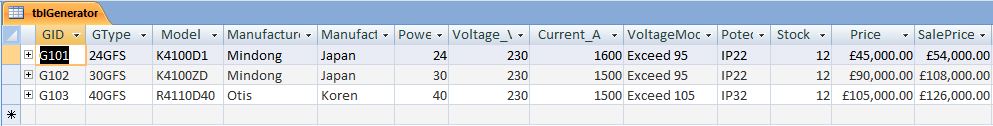
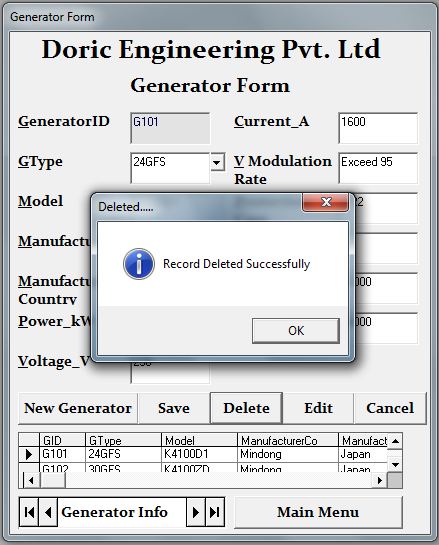
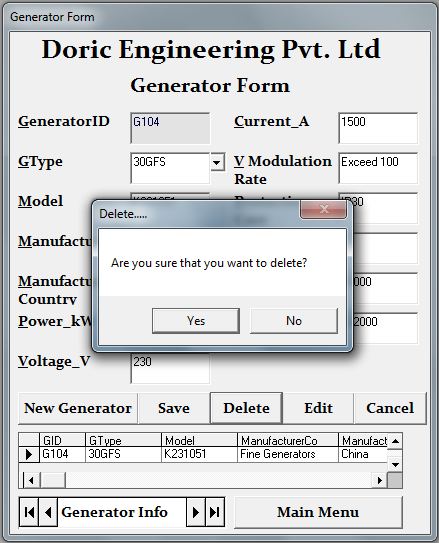


Edited Field

Field Edited in Database

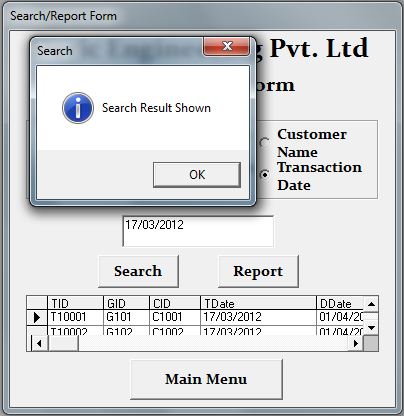
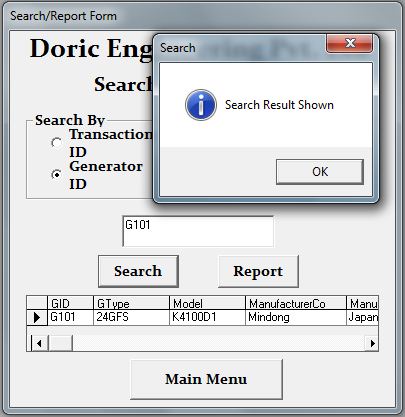
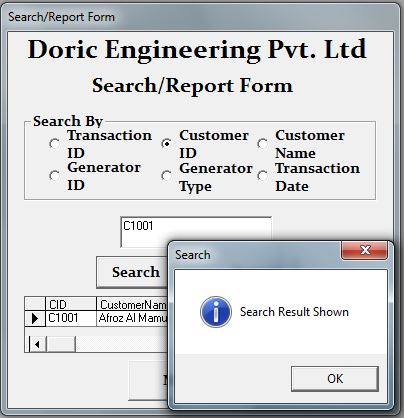


Record Saved into Database



Record Deleted from Database

Data Search



The option button is used to select the search criteria (Customer ID in the Customer table)

The text box is used to enter the search "key words"

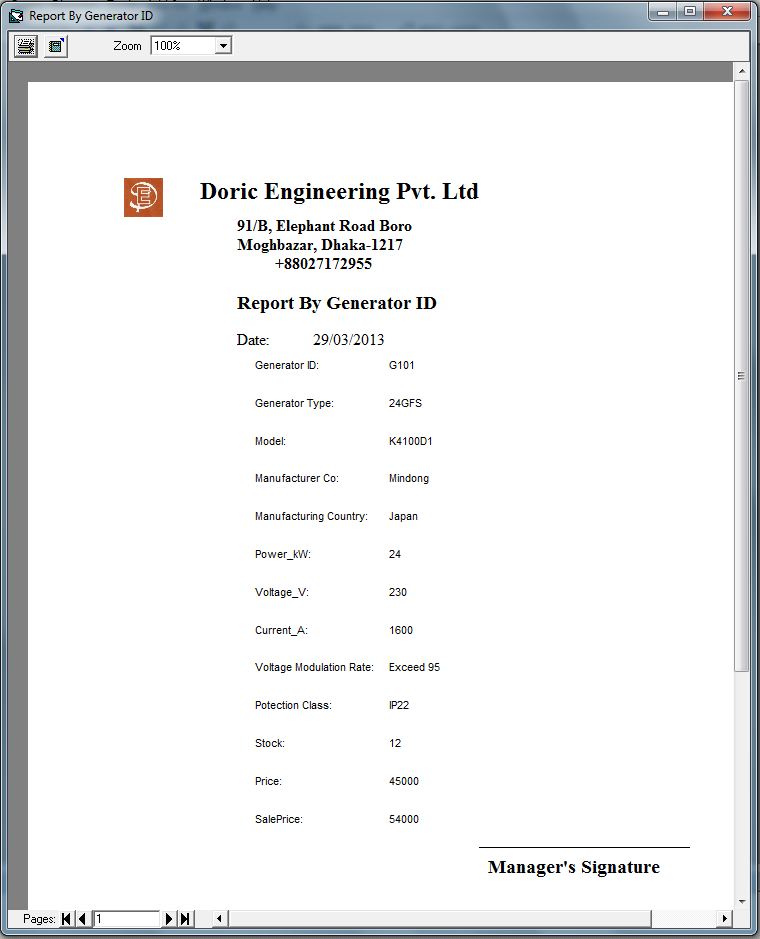
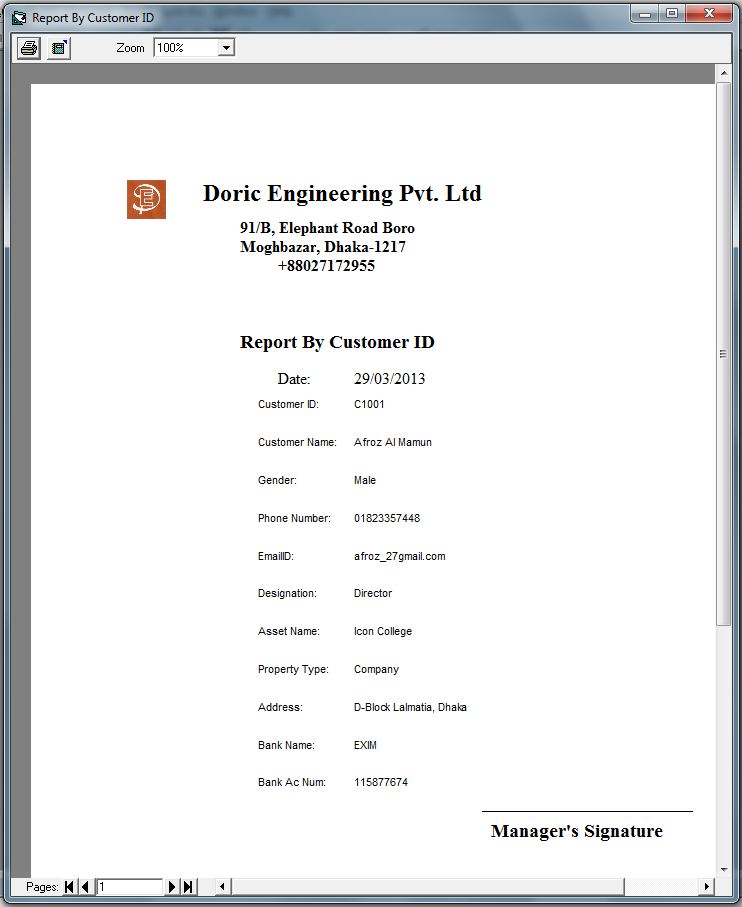
The option button is used to select the search criteria (Generator ID in the Generator table)

The text box is used to enter the search "key words"

The option button is used to select the search criteria (Transaction Date in the Transaction Table)

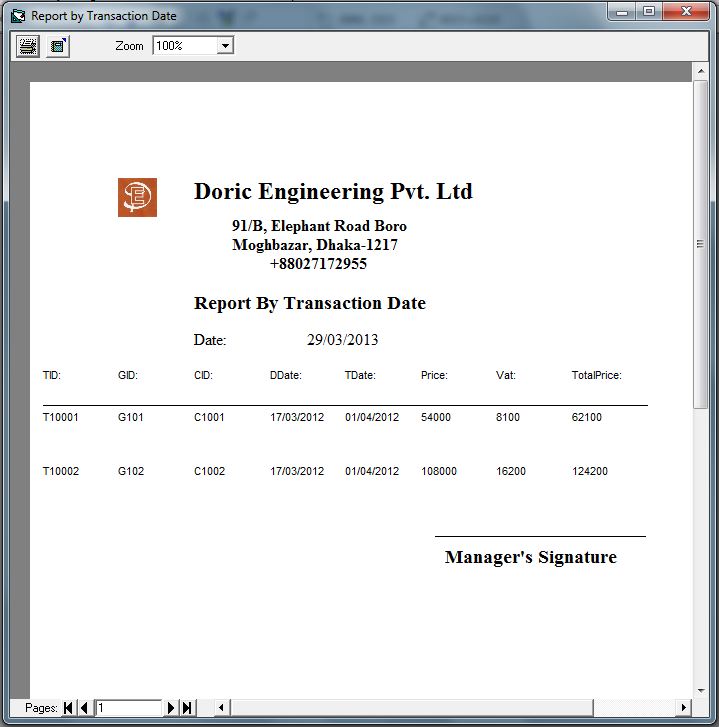
The text box is used to enter the search "key words"

Data Report

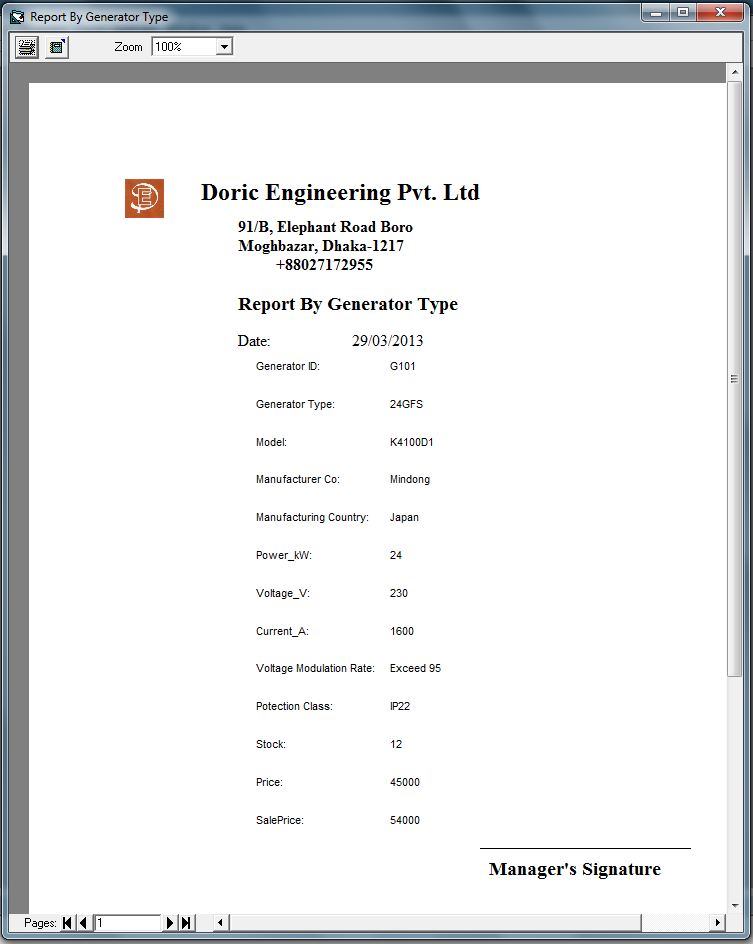


This report is generated by the report criteria - Customer ID in the Customer Table

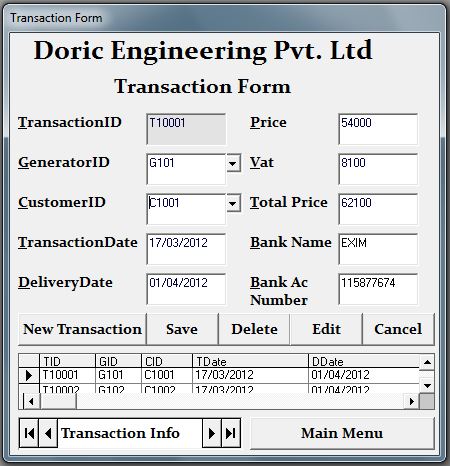
This report is generated by the report criteria - Generator ID in the Generator Table



This report is generated by the report criteria - Transaction ID in the Transaction Table



This report is generated by the report criteria - Generator Type in the Generator Table



Auto generated Transaction ID

Calculated from the data in text box - txtPrice"

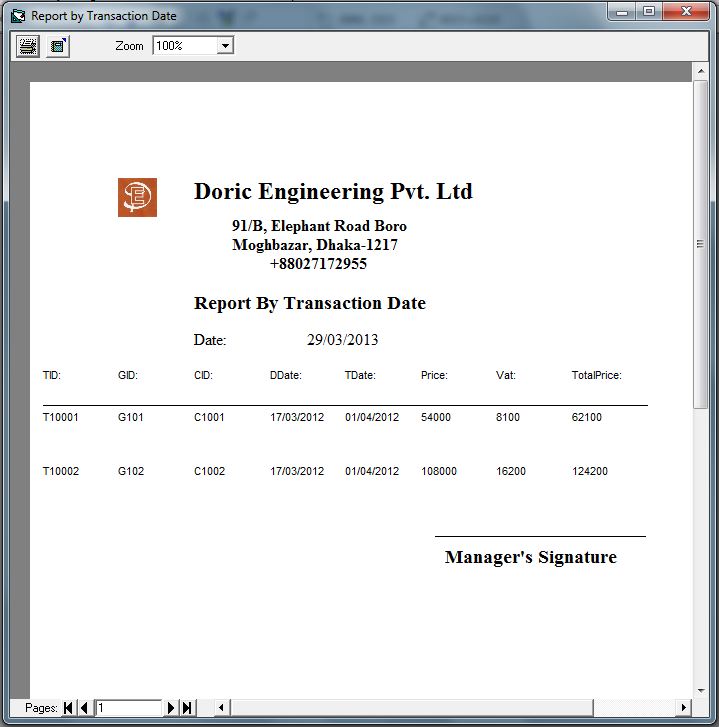
Data is inputted from the Data Combo - "dcmbCustomer" which is connected to the Customer Table by the Adodc - "adcCustomer"

Extracted from the Customer Table, from the record related to the Customer ID

Automatically inputs the current date

Calculated from the data in text box - txtTdate"

Data is inputted from the Data Combo - "dcmbGenerator" which is connected to the Generator Table by the Adodc - "adcGenerator"



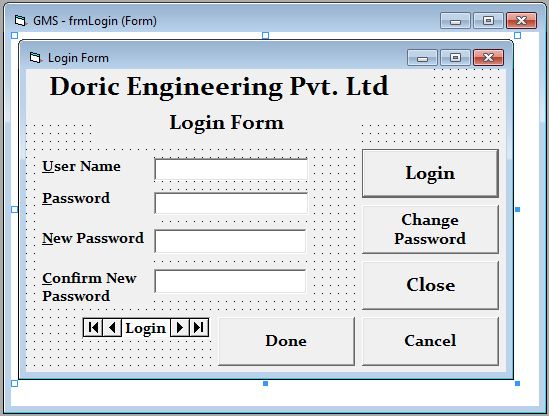
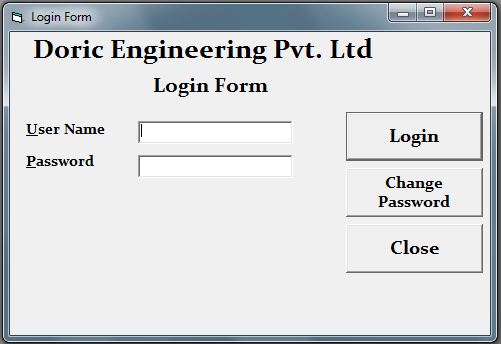
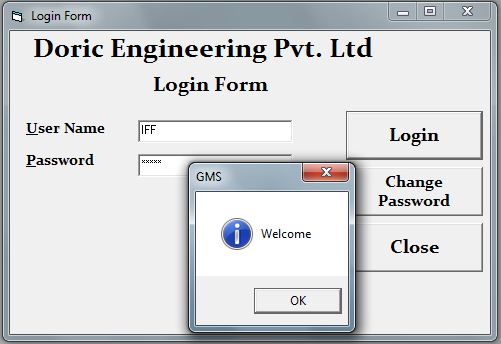


Fig 4.12

Fig 4.12

Fig 4.12

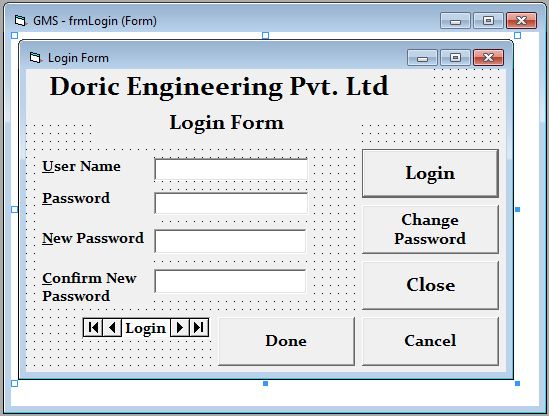
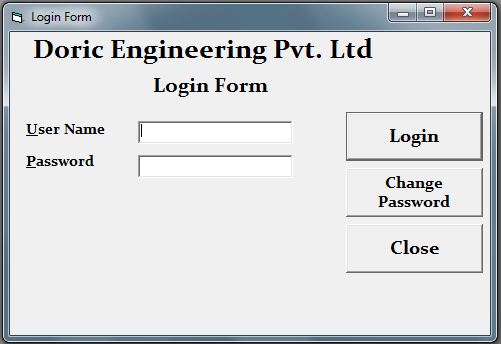
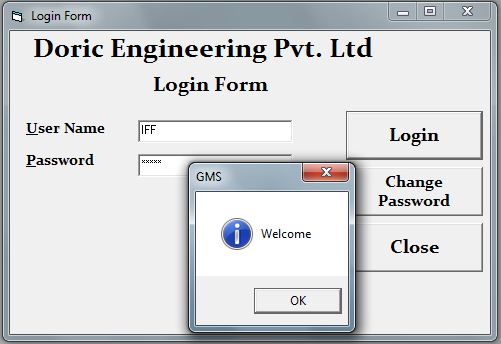


Fig 4.41

Fig 4.42

Fig 4.43

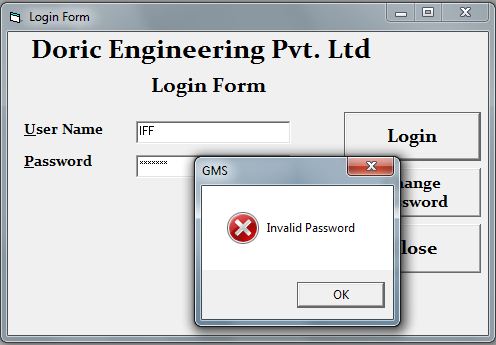
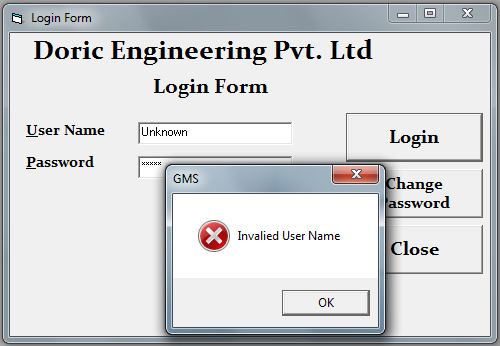


Fig 4.44

Fig 4.45

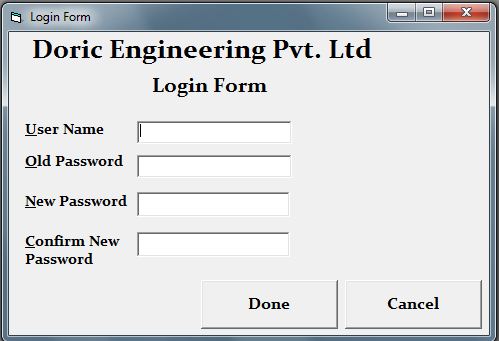


Fig 4.46

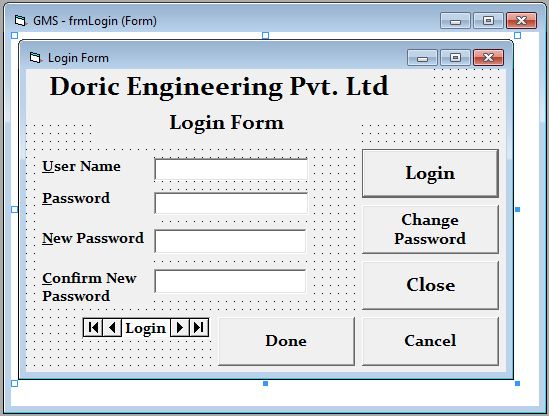
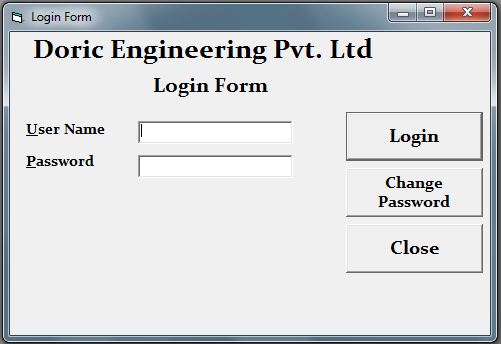
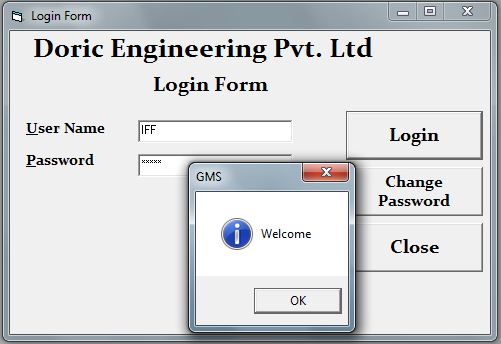


Fig 4.41

Fig 4.42

Fig 4.43

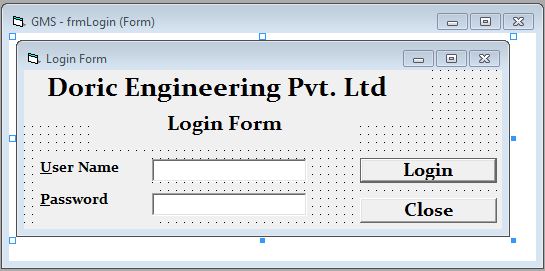


Fig 4.43

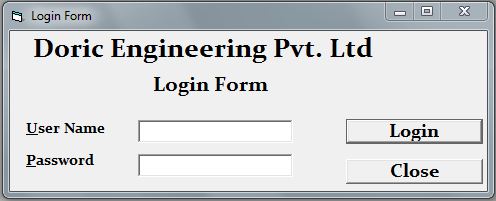


Fig 4.43

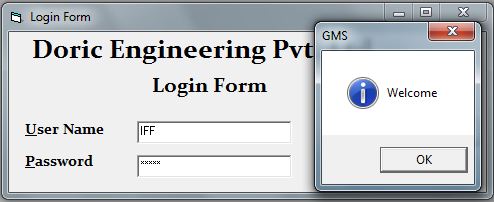


Fig 4.43

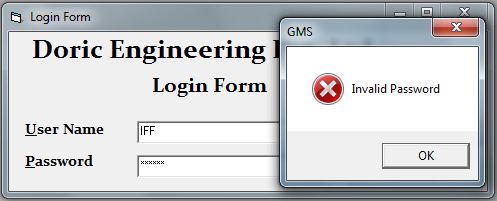
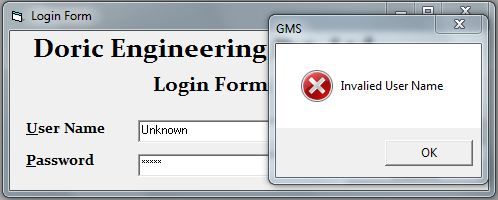


Fig 4.43

Fig 4.43

How the Security System functions

The security system uses an User Name (User ID) and Password system to allow access to the software to only authorized personnel. As the software is not installed in the company's computer and is still being tested, I am using a demo user name and password to see if the security system is functioning properly or not.

The system also allows authorized persons to change their password anytime they wish. They have to enter their user name and old password for verifying their identity and then using that, they can set a new password they like. But of course, I have set up a double entry check for them to be sure that they typed in the right password.

**15. Test Results**

Transaction Form

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Test Data | Data Type | Validation | Expectation | Actual Result | Fig |
| TID | T10005 | Normal | Format Check | Accepted | Accepted | 4.01 |
| T100057 | Abnormal | Error Message | Error Message | 4.02 |
| Transaction Date | 23/03/2013 | Normal | Format Check | Accepted | Accepted | 4.03 |
| 23/03/13 | Abnormal | Error Message | Error Message | 4.04 |

Customer Form

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Test Data | Data Type | Validation | Expectation | Actual Result | Fig |
| CID | C1006 | Normal | Format Check | Accepted | Accepted | 4.05 |
| C10068 | Abnormal | Error Message | Error Message | 4.06 |
| CustomerName | Zeba Farzana | Normal | Length Check | Accepted | Accepted | 4.07 |
| Fayez Ahmed Jahangir Masud | Abnormal | Error Message | Error Message | 4.08 |
| EmailID | akbar.ovi@yahoo.com | Normal | Format Check | Accepted | Accepted | 4.09 |
| akbarovi71 | Abnormal | Error Message | Error Message | 4.10 |
| BankName | Exim | Normal | Presence Check | Accepted | Accepted | 4.11 |
| " " | Abnormal | Error Message | Error Message | 4.12 |
| BankAcNum | 118799983 | Normal | Type Check | Accepted | Accepted | 4.13 |
| 11879a | Abnormal | Error Message | Error Message | 4.14 |

Generator Form

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field Name | Test Data | Data Type | Validation | Expectation | Actual Result | Fig |
| GID | G104 | Normal | Format Check | Accepted | Accepted | 4.15 |
| G1045 | Abnormal | Error Message | Error Message | 4.16 |
| Model | K5100ZA | Normal | Presence Check | Accepted | Accepted | 4.17 |
| " " | Abnormal | Error Message | Error Message | 4.18 |
| Stock | 17 | Normal | Range Check | Accepted | Accepted | 4.19 |
| -1 | Abnormal | Error Message | Error Message | 4.20 |
| 1, 50 | Extreme | Error Message | Accepted | 4.21, 4.22 |

Errors found during testing

While performing the testing of the system I noticed some bugs. The following is about the error which I found and how I solved them.

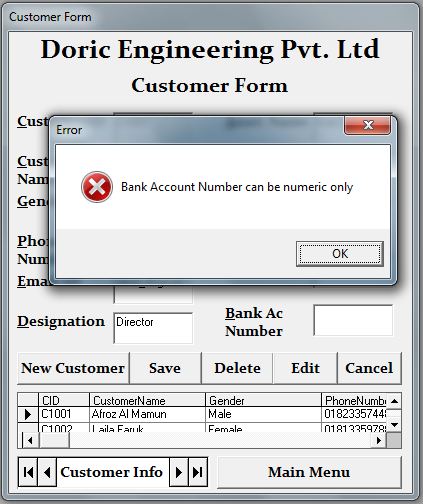


Fig 4.46

The above figures shows screenshot of the validation on the text box *txtBankAcNum*. It shows an errors message even though no data is not present, this happened when I accidentally pressed an extra backspace during the data manipulation of the Customer Form. The code was:

Private Sub txtBAcNum\_KeyPress(KeyAscii As Integer)

If IsNumeric(txtBAcNum.Text) = True Then

Else: MsgBox "Bank Account Number can be numeric only", vbCritical, "Error"

txtBAcNum.Text = ""

txtBAcNum.SetFocus

End If

End Sub

The error with the code was it would show an error message for any sort of key, except numeric, even if those keys were the "enter" or "backspace" keys. It would also show an error once at first if numeric data was entered. Thus I understood out that the code was under the wrong "". So I changed the code from the " txtBAcNum\_KeyPress(KeyAscii As Integer)" to the " txtBAcNum\_LostFocus()"

The following shows the corrected form of the code:

Private Sub txtBAcNum\_LostFocus()

If IsNumeric(txtBAcNum.Text) = True Then

Else: MsgBox "Bank Account Number can be numeric only", vbCritical, "Error"

txtBAcNum.Text = ""

txtBAcNum.SetFocus

End If

End Sub

I also faced another problem. When I set cursor focus on text box *txtBankAcNum* (in the Customer Form) keeping the text box *txtBankName* (in the Customer Form) empty, I found out that no error message was shown. I faced the same problem with text box *txtModel* (in the Generator Form), that it did not show error message when I set focus to other textboxes keeping it bank. The code for the txtBankNamewas:

Private Sub txtBName\_KeyPress(KeyAscii As Integer)

If KeyAscii = 13 And txtBName = "" Then

MsgBox "Empty Field", vbCritical

txtBName.SetFocus

ElseIf KeyAscii = 13 Then

txtBAcNum.SetFocus

End If

End Sub

So identified that the validation would only work if I pressed the "enter" key while my cursor key is focussed on that text box. So I corrected the error and made the validation universal, that it would show "error message", whenever I will change the cursor focus while keeping it blank. I did it by changing the "" from txtBName\_KeyPress(KeyAscii As Integer) to txtBName\_LostFocus() and by editing the code to meet the new correction. The following shows the corrected code:

Private Sub txtBName\_LostFocus()

If txtBName = "" Then

MsgBox "Empty Field", vbCritical

txtBName.SetFocus

End If

End Sub

I did the similar thing with the text box txtModel. Its code was:

Private Sub txtModel\_KeyPress(KeyAscii As Integer)

If KeyAscii = 13 And txtModel = "" Then

MsgBox "Empty Field", vbCritical

txtModel.SetFocus

ElseIf KeyAscii = 13 Then

txtMan.SetFocus

End If

End Sub

I changed the the "" from txtModel\_KeyPress(KeyAscii As Integer) to txtModel\_LostFocus() and by editing the code to meet the new correction. The following shows the corrected code:

Private Sub txtModel\_LostFocus()

If txtModel = "" Then

MsgBox "Empty Field", vbCritical

txtModel.SetFocus

End If

End Sub