



COMPUTER REPAIR SERVICE MANAGEMENT APPLICATION

I. Application Introduction

A. Background

The growing need for computer repairs arises from the increased utilisation of internet-based applications in our daily lives, which in turn causes an increase in issues and problems for instance internal damage (Djeunang Mezafack, Di Mascolo, and Simeu-Abazi, 2022). An inventory, service requests, repair history, and client information must all be efficiently managed and organised. In addition to being inefficient and time-consuming, manual data management procedures also run the risk of creating operational and customer service problems. This application aims to simplify and automate the data management process for computer repair services.

B. Application Objectives

The primary objectives of this application are to:

- Automate the service task and tracking process.
- Improving data processing speed and effectiveness, especially when handling large volumes of repair service data.
- Providing an intuitive user interface so that users can interact with the VBA application and efficiently manage repair service data.

These objectives are achieved through the development of this VBA Application with UserForms and storing information on different Worksheets.

II. Impact of VBA application

A. Business Impacts

This application reduces the need for paperwork in computer repair shops, implementing the VBA application is expected to result in:

- Lower operational costs due to reduced manual efforts.
- Higher efficiency in managing customer information, repair history, and service requests.
- Improved accuracy in monitoring repair statuses and billings.

B. Benefits of application

The VBA application will allow users to manage and track client repair bookings and instantly update complete information on all repair service requests in one program, saving manual labor and accommodating clients' schedules with a collection date. The system also stores all repair history for simple auditing. The application also has a Point-of-Sales system for accurate billing and invoicing and tracking for every repair booking. The program also manages repair material inventory with purchasing orders. The application also allocates repair materials stock and displays low stock. The employer can create staff (technician) profiles to identify application administrators. Unauthorized personnel cannot view customer data through the application. Traceable changes to service bookings can be traced to identifiable staff. Thus, comparing repair service data sorting and organisation to manual operations would verify accuracy.

C. Limitations of application:

- One user can access it at a time.
- Accessible only on PC.
- Performance issues, slower response times, and potential system crashes can occur as the database and the number of repair records grow.
- VBA is primarily designed for Microsoft Office applications and may not be inherently compatible with different operating systems.
- Limited database management causing difficulties in efficiently handling, organising, and querying repair data.
- Application can only be access on Windows due to some functionalities not working on Mac.
- Application should not be open with other VBA applications as the same time.

D. Potential Improvements

- Automated notification updates should be implemented to notify clients, so they are promptly informed of the status of their repairs, reducing the need for manual follow-up, and improving customer satisfaction.

- Users could access the program from home and company systems by storing it on a shared internet database. Removal of the cache would impede the application's functioning.
- The program can provide dynamic scheduling optimisation by considering technician availability, repair complexity, and customer preferences. This can lead to efficient resource allocation and improved service delivery timelines.

Given the repetitive nature of spreadsheets, data analytics, and data organisation, VBA has been particularly compatible with Excel compared to other applications in the Office suite (Kenton, 2023). Moreover, VBA offers comprehensive functionalities for automating operations and customising Excel features, rendering it a versatile and efficient tool for data management applications in computer repair services.

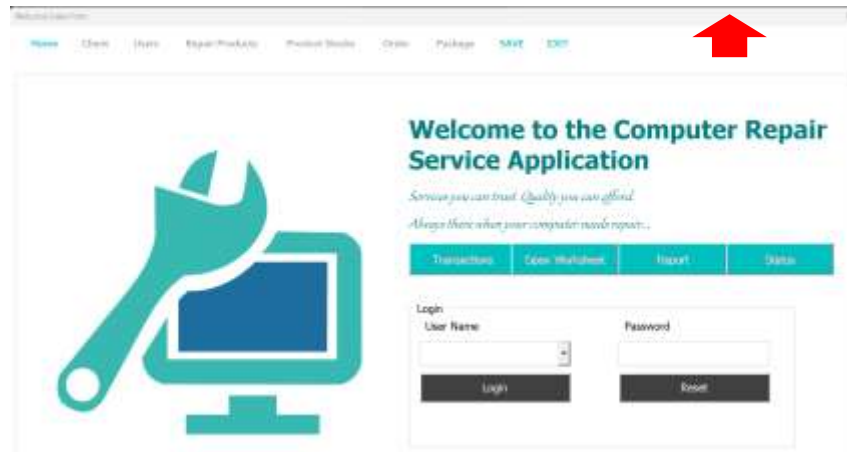
III. User's Operational Manual

1. Login

Open Excel document and you will be greeted with the Welcome Form.

In the 'User Name' dropdown box, SELECT “**Admin**” as your User Name with the password “**admin123**”, click ‘**Login**’ button to activate the application.

*If it does not instantly allow you to enter the password, please give it a little time and click ‘Reset’ button to run.



Before entering user details, all confidential sheets and forms are hidden.

Click and drag the top banner to move the application.

2. Create new transaction

Click ‘Transaction’ button to view the details form (Transaction Form) (1).



Click ‘Open Sheet’ to view Excel Workbook (2).

The second page of the transaction form will appear. Click the ‘Client’ button to open “Customer Data Form”, enter the new client’s details (1).

The screenshot shows the software interface with three main panels. On the left is the 'Customer Details Table' with a list of customers. In the center is the 'Customer Data Form' with fields for Customer Name, Address, Phone Number, Starting Date, and Computer. On the right is the 'Available Package Table' with a list of packages. A red arrow points to the 'Client' button at the top, and a green arrow points to the 'Save' button at the bottom of the 'Customer Data Form'.

No.	Customer Name	Address	Phone
1	Cecil Bridges	45, First Street	172
2	Clara Lambert	82, Driboune St	196
3	Tristan Boyd	12, Maxwell St	123
4	Bobby McQuinn	78, Crescent St	177
5	Kimmy Jones	58, Fordham St	170

Customer Name	
Address	
Phone Number	
Starting Date	
Computer	
Package	
Package	
Number	
Price per	
Total Amount	
Save	Delete
Reset	

Package Name	Price per	Rept.	Package
PC HEALTH C	\$40.00	Virat S	
HARDWARE D	\$15.00	Peripha	
DATA RECDIT	\$25.00	Data R	
INTERNET SET	\$35.00	Setup F	

After entering a new client’s details, click “Save” button to save new customer’s details into ‘Customer Data Table’ (2).

After customer details is inputted, click ‘New’ button to create a new transaction, and click ‘Yes’ (1).

The screenshot shows the software interface with three main panels. On the left is the 'Customer Details Table' with a list of customers. In the center is the 'Transaction Data' form with fields for Transaction ID, Customer Name, Address, Phone Number, Starting Date, Completion Date, Package Data, Number of Computers, Price per Rept, and Total Amount. On the right is the 'Available Package Table' with a list of packages. A red arrow points to the 'New' button at the bottom of the 'Transaction Data' form, and a green arrow points to the 'Yes' button in the 'New Transaction' dialog box.

No.	Customer Name	Address	Phone
1	Cecil Bridges	45, First Street	172
2	Clara Lambert	82, Driboune St	196
3	Tristan Boyd	12, Maxwell St	123
4	Bobby McQuinn	78, Crescent St	177
5	Kimmy Jones	58, Fordham St	170
6	Owen Henshelt	49, Kingsboro	181

Transaction ID	
Customer Name	
Address	
Phone Number	
Starting Date	
Completion Date	
Package Data	
Number of Computers	
Price per Rept (\$)	
Total Amount (\$)	
New	Save
Update	Delete
Print	Reset

Package Name	Price per	Rept.	Package
PC HEALTH C	\$40.00	Virat S	
HARDWARE D	\$15.00	Peripha	
DATA RECDIT	\$25.00	Data R	
INTERNET SET	\$35.00	Setup F	

Double click on a customer on the Customer Details Table and input dates. Press “ENTER” on your keyboard to be directed to the next input field (2).

Input the 'Package Data' by inputting the 'Number of Computers' first (1). Double click on the package in the 'Package Table' that the customer wants (2). After that click the 'Save' button (3).

The screenshot shows the 'Package' section of the software. On the left is the 'Customer Details Table' with columns: No, Customer Name, Address, and Phone. It lists five customers. In the center is the 'Customer Data' form with fields for Transaction ID, Customer Name, Address, Phone Number, Starting Date, and Completion Date. Below these are the 'Package Data' fields: Package (dropdown), Number of Computers (input field with a red arrow pointing to it labeled '1'), Price per Repair (£), and Total Amount (£). At the bottom are buttons: New, Save, Delete, Update, Print, and Reset. On the right is the 'Available Package Table' with columns: Package Name, Price per Repair, and Package. It lists three packages: PC HEALTH CHECK, HARDWARE ID, and DATA RECOVERY. A green arrow points to this table labeled '2'. At the bottom right are buttons: Total Transaction, Type of Payment, Total Amount (£), Paid (£), Return Amount (£), and Staff. A yellow arrow points to the 'Save' button labeled '3'.

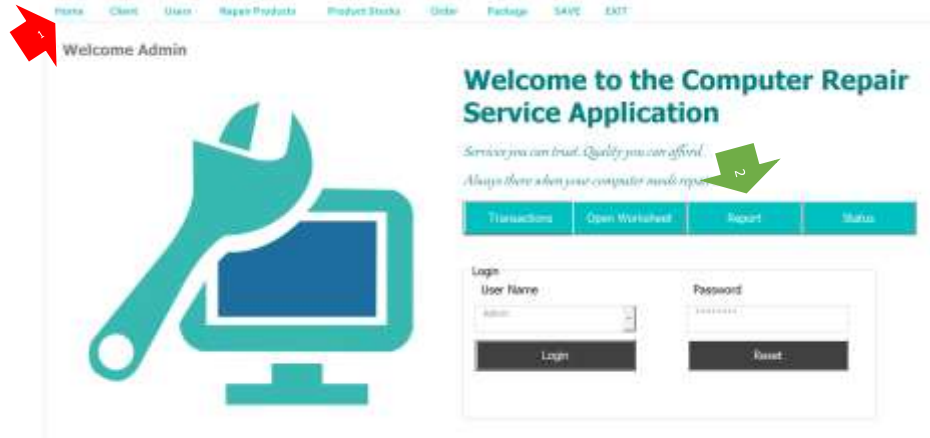
After customer details has been saved, it will appear in the data table (1). Double click on it and input the 'Total Transaction' section (2). Once done, click on the 'Print' button to print out invoice for customer (If needed). Click 'Save' to save the transaction into the transaction report (3) (Worksheet: Transaction).

The screenshot shows the 'Transaction' section of the software. On the left is the 'Customer Details Table' with columns: No, Customer Name, Address, and Phone. It lists five customers. In the center is the 'Customer Data' form with fields for Transaction ID, Customer Name, Address, Phone Number, Starting Date, and Completion Date. Below these are the 'Package Data' fields: Package (dropdown), Number of Computers (input field), Price per Repair (£), and Total Amount (£). At the bottom are buttons: New, Save, Delete, Update, Print, and Reset. On the right is the 'Available Package Table' with columns: Package Name, Price per Repair, and Package. It lists three packages: PC HEALTH CHECK, HARDWARE ID, and DATA RECOVERY. A green arrow points to this table labeled '2'. At the bottom right are buttons: Total Transaction, Type of Payment, Total Amount (£), Paid (£), Return Amount (£), and Staff. A red arrow points to the 'Print' button labeled '5'. A blue arrow points to the 'Search' button labeled '4'. An orange arrow points to the 'Add' button labeled '5'.

Click 'Search' button to find a customer (4) and 'Add' button to add a package (5).

3. View transaction and status details

The transaction history can be viewed in the 'Report' (2) button of 'Home' (1).



The transaction report will display all the transaction history.

Transaction Report Details

No.	Transaction No.	Staff Name	Customer Name	Address	Phone Number	Start Date	Completion Date	Package	Number of compl	Price per Repair	Total Price
1	TR-1001	Jason Jules	Customer 1	Address 1	1837864038	11/11/2022	11/14/2022	Package 1	3	£40	£120.00
2	TR-1002	Jason Jules	Customer 2	Address 2	1723647809	4/12/2023	4/17/2023	Package 2	2	£25	£50.00
3	TR-1003	Holly Plonow	Customer 3	Address 3	1348578545	12/10/2022	12/15/2022	Package 3	5	£15	£75.00
4	TR-1004	Jason Jules	Customer 4	Address 4	1334585673	12/9/2022	12/9/2022	Package 4	1	£25	£25.00
5	TR-1109	Jason Jules	Kimmy Jones	10, Parahart St	1722852847	3/3/2024	3/18/2024	PC HEALTH CHG	5	£30	£150.00
6	TR-1117	Jason Jules	David Jones	10, Parahart St	1722852847	3/3/2024	3/18/2024	PC HEALTH CHG	5	£30	£150.00

Find Data

Starting Date: Completion Date:

All Data Find Reset

Transaction Report

£ 590

You can find specific transactions by inputting valid dates (1).

The repair status details can be found in the ‘Status’ button of ‘Home’ page.1.

Repair Status Details

*To further extend information, please double click on the selected data

No.	Transaction Number	Staff Name	Customer Name	Address	Phone Number	Status	Payment
1	TR-1001	Jason Jules	Customer 1	Address 1	1837386428	Collected	Cash
2	TR-1002	Jason Jules	Customer 2	Address 2	1723647839	Repairing	Cash
3	TR-1003	Molly Monroe	Customer 3	Address 3	1348576545	Repairing	Cash
4	TR-1004	Jason Jules	Customer 4	Address 4	1334555673	Completed	Cash
5	TR-1005	Jason Jules	Customer 5	Address 5	1876543245	Completed	Cash
6	TR-1108	Jason Jules	Kimmy Jones	99, PondHart Street	1722893847	Completed	Cash
7	TR-1137	Aceus	Kimmy Jones	99, PondHart Street	1722893847	Repairing process	Cash

Repair Status Filter

Total Amount: 7Data

Specific status can be found using the ‘Filter’ button (1).

Double-click on the data to update status (1).

Repair Status Details

*To further extend information, please double click on the selected data

No.	Transaction Number	Staff Name	Customer Name	Address	Phone Number	Status	Payment
1	TR-1001	Jason Jules	Customer 1	Address 1	1837386428	Collected	Cash
2	TR-1002	Jason Jules	Customer 2	Address 2	1723647839	Repairing	Cash
3	TR-1003	Molly Monroe	Customer 3	Address 3	1348576545	Repairing	Cash
4	TR-1004	Jason Jules	Customer 4	Address 4	1334555673	Completed	Cash
5	TR-1005	Jason Jules	Customer 5	Address 5	1876543245	Completed	Cash
6	TR-1108	Jason Jules	Kimmy Jones	99, PondHart Street	1722893847	Completed	Cash
7	TR-1137	Aceus	Kimmy Jones	99, PondHart Street	1722893847	Repairing process	Cash

Repair Status Filter

Total Amount: 7Data

Update Status

Transaction ID: TR-1137

Customer Name: Kimmy Jones

Address: 99, PondHart Street

Phone Number: 1722893847

Repair Status: Repairing process

Payment Status: Cash

4. View and add/update customer and staff details

In 'Home' (Welcome), click 'Client' button to add/update new customer details (1).

Customer Details

Customer Name

Address

Phone Number

No.	Customer Name	Address	Phone Number
1	Cecil Bridges	49, Flint Street	1722960589
2	Claran Lambert	82, Ombourne St	1643960415
3	Tristan Boyd	12, Maxwell Str	1233960360
4	Bobby McDowell	70, Crescent Stre	1732665546
5	Kimmy Jones	59, Parthart Str	1722893847

Save Delete Reset

Click 'Users' button to add/update staff's details (1).

Staff Data Form

Full Name Gender Address Phone Number

No.	Full Name	Gender	Address	Phone Number
1	Jason Jules	Male	23, Manor Sling Street	1326550589
2	Sherlyn Simmons	Female	56, Flinder Street	1234345678
3	Molly Monroe	Female	12, Vane Street	1938473283

Save Delete Update Reset

5. View and edit repair stocks/packages

In 'Home' (page.1), click on 'Repair Products', 'Product Stocks', 'Order', and 'Package' respectively to add/edit/delete the respective details.

- Repair Products: Update repair product stocks and usage amounts.
- Product Stocks: Add/Update/Delete repair products.
- Order: Order new repair products
- Package: Add/Update/Delete repair package details.

Repair Products Detail Form

Transaction ID: Product Name: Amount Used:

Product ID: Brand: Stock: Usage Date:

No.	Transaction ID	Product ID	Product Name	Brand	Amount Used	Usage Date
1	TR-1001	PR-101	Multimeter	Acer	5	5/23/2023
2	TR-1002	PR-102	Tweezers	Dell	20	6/5/2023
3	TR-1003	PR-103	Thermal Paste	Apple	30	3/11/2023
4	TR-1004	PR-104	Multimeter	Apple	45	3/11/2023
5	TR-1042	PR-101	Tweezers	Acer	8	12/12/2023

Product Stock Detail Form

Product ID:

Product Name:

Stock: Brand:

Reorder Level: Reorder Status:

No.	Product ID	Product Name	Stocks	Brand
1	PR-101	Tweezers	24	Acer
2	PR-102	Multimeter	42	Apple
3	PR-103	Thermal Paste	31	Lenovo
4	PR-104	Cleaning Kits	15	Dell

Order Detail Form

Transaction ID: Product Name: Total Price:

Product ID: Brands: Total: Purchase Date:

No.	Transaction ID	Product ID	Product Name	Brand	Total	Price	Date of Order
1	TR-1000	PR-101	Tweezers	Acer	10	10	5/5/2023
2	TR-1001	PR-102	Multimeter	Apple	11	55	11/5/2023
3	TR-1002	PR-103	Thermal Paste	Lenovo	12	15	6/5/2023

Repair Package Form

Package Name:

Price per Repair:

Description:

No.	Package Name	Price per Repair	Package Description
1	PC HEALTH CHECK	\$40.00	Virus Scan and Repair, Sys
2	HARDWARE INSTALLATION	\$15.00	Peripheral Installation, Wre
3	DATA RECOVERY PACKAGE	\$25.00	Data Recovery, Scan Hard
4	INTERNET SETUP PACKAGE	\$20.00	Setup Broadband, Setup W

6. Assumptions

- User must remember their password. Wrong input of login details prevents entry into application.
- All dates should be inputted in the format mm/dd/yyyy including forward slashes. Any invalid dates would display an error message and allow user to re-input.
- Client's computer brand is implicitly associated to customers.

7. Limitations

- All the date boxes will not accept date format, such as dd/mm/yyyy or ddmmyyyy.
- Transaction number will not be based on actual previous number.

8. Warning Message

- Incorrect login details: "Please try again!"
- Empty login details: "Please input correctly."
- Not clicking "New" button first in the Transaction Form: "Please fill in the Transaction ID first".
- No input in Transaction Form: "Please fill in completely."
- Incorrect Starting Date/Completion Date: "Please enter a valid date in the format mm/dd/yyyy."
- Invalid date logic: "Completion date cannot be earlier than starting date."
- Incorrect number of computers: "Please enter a valid number."
- Invalid phone number: "Please enter only numbers."
- Invalid name: "Please enter a valid name."
- Incorrect Reorder/Stock/Price number: "Please enter a positive integer for Reorder/Stock/Price."
- No payment method selected: "Transaction details not complete, please recheck."

9. Trouble-shooting section

- Click 'EXIT' button to quit the application.
- If the Transaction ID is not list down accurately, the user can directly update it the worksheet.
- Refer to specific worksheets if details on userform not updated/deleted for accuracy.

IV. Technical Documentation

A. Worksheets

Welcome

This is the main homepage for the program. The login details must be correct to activate the application.

Transaction Form

The worksheet is hidden. It holds the input for customer and repair information. This is also where the billing and invoice matters occur.

Transaction Report Details

The worksheet is hidden. It holds all current and new/updated transaction information. The application will pull information from this when a user creates a transaction and store it when they add a new transaction.

Repair Status Details

This worksheet is hidden. It holds all current and new/updated repair status pull from the transaction details.

B. Userforms

CustomerForm

Provides an interface for user to manage customer details with Customer Name, Address, and Phone number.

OrderForm

Provides an interface for managing repair product stocks.

PackageForm

Provides an interface for managing package details, allowing users to add, edit, and delete packages.

ProductForm

RepairsForm

StaffForm

StatusForm

TransactionForm

Main Program. The multipage allows editing details

CustomerForm

PackageForm

C. Modules

D. Classes are not used in this application.

Reference List

Djeunang Mezafack, R.A., Di Mascolo, M. and Simeu-Abazi, Z. (2022) ‘Systematic literature review of repair shops: focus on sustainability’, *International Journal of Production Research*, 60(23), pp.7093-7112.

Kenton, W. (2023) *Visual Basic for Applications (VBA): Definition, Uses, Examples*. Available at: <https://www.investopedia.com/terms/v/visual-basic-for-applications-vba.asp> (Accessed: 10 January 2024).

V. Appendices (Code)

Userform

CustomerForm

Option Explicit

' Delete customer details based on selected row in DATATABLE

Private Sub CMDDEL_Click()

 If Me.TXTNUMB1.value = "" Then

 ' If no data is selected

 MsgBox "Please double click and select the data on the Data Table.",
vbInformation, "Delete Details"

 Else

 Select Case MsgBox("The selected details will be deleted." & vbCrLf & "Please
confirm.", _

 vbYesNo Or vbQuestion Or vbDefaultButton1, "Delete Details")

 Case vbNo

 Exit Sub

 Case vbYes

 End Select

 ' Clear form data

 Me.DATATABLE.value = ""

 Me.TXTNUMB1.value = ""

 Sheet4.Select

 Selection.EntireRow.Delete

 Call GetData

 Sheet4.Select

End If

End Sub

' Reset input fields in the form

Private Sub CMDRESET_Click()

 Me.TXTNAME.value = ""

 Me.TXTADDRESS.value = ""

 Me.TXTPHONE.value = ""

```

    Me.DATATABLE.value = ""
    Me.TXTNUMB1.value = ""
End Sub

' Save customer details to the Customer sheet
Private Sub CMDSAVE_Click()
    Dim objCust As Object
    Set objCust = Sheet4.Range("A80000").End(xlUp)

    If Not IsValidInput() Then
        Exit Sub
    End If

    ' Save data to the Customer sheet
    objCust.Offset(1, 0).value = "=ROW()-ROW($A$5)"
    objCust.Offset(1, 1).value = Me.TXTNAME.value
    objCust.Offset(1, 2).value = Me.TXTADDRESS.value
    objCust.Offset(1, 3).value = Me.TXTPHONE.value

    MsgBox "The details have been saved.", vbInformation, "Input details"
    Call GetData
    Call GetCustomer

    ' Clear input fields
    Me.TXTNAME.value = ""
    Me.TXTADDRESS.value = ""
    Me.TXTPHONE.value = ""
End Sub

' Validate input data for customer details
Private Function IsValidInput() As Boolean

    ' Check if name, address, and phone number are filled, and in correct format
    If Me.TXTNAME.value = "" Or Me.TXTADDRESS.value = "" Or Me.TXTPHONE.value = "" Then
        MsgBox "Please fill in the details completely.", vbInformation, "Input Data"
        IsValidInput = False
    End If
End Function

```

```

ElseIf Not IsValidName(Me.TXTNAME.value) Then
    MsgBox "Please enter a valid Name.", vbExclamation, "Input Error"
    IsValidInput = False
ElseIf Not IsValidAddress(Me.TXTADDRESS.value) Then
    MsgBox "Please enter a valid address.", vbExclamation, "Input Error"
    IsValidInput = False
ElseIf Not IsValidNumber(Me.TXTPHONE.value) Then
    MsgBox "Please enter valid phone number of 10 numbers.", vbExclamation, "Input
Error"
    IsValidInput = False
Else
    IsValidInput = True
End If
End Function

```

' Validate if the name contains only letters

```
Private Function IsValidName(ByVal value As String) As Boolean
```

```
    Dim i As Integer
```

```
    IsValidName = Len(value) > 0
```

```
    For i = 1 To Len(value)
```

```
        If IsNumeric(Mid(value, i, 1)) Then
```

```
            IsValidName = False
```

```
            Exit Function
```

```
        End If
```

```
    Next i
```

```
End Function
```

' Validate if the address is not empty

```
Private Function IsValidAddress(ByVal value As String) As Boolean
```

```
    ' Allowing both letters and numbers in the address
```

```
    IsValidAddress = Len(value) > 0
```

```
End Function
```

```
Private Function IsValidNumber(ByVal value As String) As Boolean
```

```
    ' Validate if the phone number is 10 digits and numeric
```

```
IsValidNumber = Len(value) = 10 And IsNumeric(value) And Val(value) > 0  
End Function
```

'Elements for the this functionality is inspired from:

'<https://excelmacromastery.com/vba-listbox/>

'Accessed Date: 27 December 2023

' Populate DATATABLE with customer data

```
Private Sub GetData()  
    Dim ICust, irow As Long  
    irow = Sheet4.Range("A" & Rows.Count).End(xlUp).Row  
    ICust = Application.WorksheetFunction.CountA(Sheet4.Range("A6:A10000"))  
    If ICust = 0 Then  
        Me.DATATABLE.RowSource = ""  
    Else  
        Me.DATATABLE.RowSource = "Customer!A6:D" & irow  
    End If  
End Sub
```

'Elements for this functionality is inspired from:

'<https://stackoverflow.com/questions/66270458/vba-class-module-dblclick-event>

' Double-click event for DATATABLE to fill form fields with selected customer data

'Accessed Date: 27 December 2023

```
Private Sub DATATABLE_DblClick(ByVal Cancel As MSForms.ReturnBoolean)  
    On Error GoTo Error  
    Dim DataSource, CellActive As Long  
    Me.TXTNUMB1.value = Me.DATATABLE.value  
    Me.TXTNAME.value = Me.DATATABLE.Column(1)  
    Me.TXTADDRESS.value = Me.DATATABLE.Column(2)  
    Me.TXTPHONE.value = Me.DATATABLE.Column(3)  
    Sheet4.Select  
    DataSource = Sheets("Customer").Cells(Rows.Count, "A").End(xlUp).Row  
    Sheets("Customer").Range("A6:A" & DataSource).Find(What:=Me.TXTNUMB1.value,  
    —  
        LookIn:=xlValues, LookAt:=xlWhole).Activate  
    CellActive = ActiveCell.Row  
    Sheet4.Select
```

```

Exit Sub
Error:
    MsgBox "The selected data will be deleted", vbInformation, "Delete Details"
End Sub

' Initialize the UserForm
Private Sub UserForm_Initialize()
    Call GetData
    Call GetCustomer
End Sub

' Get customer data for CUSTTABLE
Private Sub GetCustomer()
    Dim ICust, irow As Long
    irow = Sheet4.Range("B" & Rows.Count).End(xlUp).Row
    ICust = Application.WorksheetFunction.CountA(Sheet4.Range("B6:B10000"))
    If ICust = 0 Then
        TransactionForm.CUSTTABLE.RowSource = ""
    Else
        TransactionForm.CUSTTABLE.RowSource = "Customer!A6:D" & irow
    End If
End Sub

```

OrderForm

Option Explicit

' Update form fields when product ID in CBID changes

Private Sub CBID_Change()

On Error GoTo Error

Dim ObjProduct As Object

Set ObjProduct = Sheet2.Range("B6:B100000").Find(What:=Me.CBID.value,
LookIn:=xlValues)

Me.TXTNAME.value = ObjProduct.Offset(0, 1).value

Me.TXTBRAND.value = ObjProduct.Offset(0, 3).value

Exit Sub

Error:

Call MsgBox("Product ID not updated", vbInformation, "Product ID")

End Sub

'The following code has been inspired and adapted from a youtube video from the
following website

'https://www.youtube.com/watch?v=TbFSM18LPXs&t=261s&ab_channel=WisdombyData

'Date Accessed: 29 December 2023

' Generate a new transaction ID

Private Sub CMDNEW_Click()

Sheet3.Range("H3").value = Sheet3.Range("H3").value + 1

If Sheet3.Range("H2").value = 1 Then

Me.TXTID.value = "TR-100" & Sheet3.Range("H3").value

End If

If Sheet3.Range("H2").value = 2 Then

Me.TXTID.value = "TR-10" & Sheet3.Range("H3").value

End If

If Sheet3.Range("H2").value = 3 Then

Me.TXTID.value = "TR-1" & Sheet3.Range("H3").value

```

End If
If Sheet3.Range("H2").value = 4 Then
Me.TXTID.value = "TR-0" & Sheet3.Range("H3").value
End If
If Sheet3.Range("H2").value = 5 Then
Me.TXTID.value = "TR-0" & Sheet3.Range("H3").value
End If
End Sub

' Delete selected data
Private Sub CMDDELETE_Click()
Dim UpdateStock As Object
Set UpdateStock = Sheet2.Range("B6:B10000").Find(What:=Me.CBID.value,
LookIn:=xlValues)
If Me.TXTNUMB1.value = "" Then
Call MsgBox("Please select the data on the table.", vbInformation, "Delete Details")
Else
Select Case MsgBox("Are you sure you want to delete the data?" _
& vbCrLf & "Please confirm" _
, vbYesNo Or vbQuestion Or vbDefaultButton1, "Delete Details")
Case vbNo
Exit Sub
Case vbYes
End Select
Me.DATATABLE.value = ""
Me.TXTNUMB1.value = ""
Sheet3.Select
Selection.EntireRow.Delete
Call GetData
UpdateStock.Offset(0, 2).value = Val(UpdateStock.Offset(0, 2).value) -
Val(Me.TXTTOT.value)

'Clear values
Me.TXTNUMB1.value = ""
Me.TXTID.value = ""
Me.CBID.value = ""
Me.TXTNAME.value = ""

```



```

Me.TXTBRAND.value = ""
Me.TXTTOT.value = ""
Me.TXTTOTAL.value = ""
Me.TXTPURDATE.value = ""
Sheet3.Select
End If
End Sub

```

```

' Reset input fields
Private Sub CMDRESET_Click()
Me.TXTID.value = ""
Me.CBID.value = ""
Me.TXTNAME.value = ""
Me.TXTBRAND.value = ""
Me.TXTTOT.value = ""
Me.TXTTOTAL.value = ""
Me.TXTPURDATE.value = ""
Me.DATATABLE.value = ""
Me.TXTNUMB1.value = ""

```

```

End Sub

```

```

' Save transaction details
Private Sub CMDSAVE_Click()
Dim objPurchase As Object
Dim objUpdate As Object
'Display on spreadsheet data
Set objPurchase = Sheet3.Range("A80000").End(xlUp)
Set objUpdate = Sheet2.Range("B6:B10000").Find(What:=Me.CBID.value,
LookIn:=xlValues)

```

```

'Ensure user input details correctly
If Me.TXTID.value = "" _
Or Me.CBID.value = "" _
Or Me.TXTNAME.value = "" _
Or Me.TXTBRAND.value = "" _
Or Me.TXTTOT.value = "" _

```

```

Or Me.TXTTOTAL.value = "" _
Or Me.TXTPURDATE.value = "" Then

'Prompt user to reinput details
Call MsgBox("Please input full details.", vbInformation, "Input Details")
'Save details if complete
Else
objPurchase.Offset(1, 0).value = "=ROW()-ROW($A$5)"
objPurchase.Offset(1, 1).value = Me.TXTID.value
objPurchase.Offset(1, 2).value = Me.CBID.value
objPurchase.Offset(1, 3).value = Me.TXTNAME.value
objPurchase.Offset(1, 4).value = Me.TXTBRAND.value
objPurchase.Offset(1, 5).value = Me.TXTTOT.value
objPurchase.Offset(1, 6).value = Me.TXTTOTAL.value
objPurchase.Offset(1, 7).value = Format(CDate(Me.TXTPURDATE.Text),
"DD/MM/YYYY")

objUpdate.Offset(0, 2).value = Val(objUpdate.Offset(0, 2).value) +
Val(Me.TXTTOT.value)

Call MsgBox("Product details is saved.", vbInformation, "Input Details")
Call GetData
Call StatusOrder
'Clear form after saved
Me.TXTID.value = ""
Me.CBID.value = ""
Me.TXTNAME.value = ""
Me.TXTBRAND.value = ""
Me.TXTTOT.value = ""
Me.TXTTOTAL.value = ""
Me.TXTPURDATE.value = ""

End If

' Validate Total
If Not IsNumeric(Me.TXTTOT.value) Or Me.TXTTOT.value <= 0 Then

```

```
MsgBox "Please enter a positive numeric value for Total.", vbExclamation, "Invalid Input"
```

```
Exit Sub
```

```
End If
```

```
' Validate Total Price
```

```
If Not IsNumeric(Me.TXTTOTAL.value) Or Me.TXTTOTAL.value <= 0 Then
```

```
MsgBox "Please enter a positive numeric value for Total Price.", vbExclamation, "Invalid Input"
```

```
Exit Sub
```

```
End If
```

```
' Validate Purchase Date
```

```
If Not IsDate(Me.TXTPURDATE.value) Then
```

```
MsgBox "Please enter a valid date in the format DD/MM/YYYY for Purchase Date.", vbExclamation, "Invalid Input"
```

```
Exit Sub
```

```
End If
```

```
End Sub
```

```
' Validate user input for transaction details
```

```
Private Function IsValidInput() As Boolean
```

```
If Me.TXTTOT.value = "" Or Me.TXTTOTAL.value = "" Or Me.TXTPURDATE.value = "" Then
```

```
MsgBox "Please fill in the details completely", vbInformation, "Input Details"
```

```
IsValidInput = False
```

```
ElseIf Not IsValidTotal(Me.TXTTOT.value) Then
```

```
MsgBox "Please enter a valid number.", vbExclamation, "Input Error"
```

```
IsValidInput = False
```

```
ElseIf Not IsValidTotalPrice(Me.TXTTOTAL.value) Then
```

```
MsgBox "Please enter a valid number.", vbExclamation, "Input Error"
```

```
IsValidInput = False
```

```
ElseIf Not IsValidPurchaseDate(Me.TXTPURDATE.value) Then
```

```
MsgBox "Please enter a valid date.", vbExclamation, "Input Error"
```

```
IsValidInput = False
```

```
Else
    IsValidInput = True
End If
End Function
```

```
End Function
```

```
' Get data for DATATABLE from the Order sheet
Private Sub GetData()
Dim objPurchase As Long
Dim irow As Long
irow = Sheet3.Range("A" & Rows.Count).End(xlUp).Row
objPurchase = Application.WorksheetFunction.CountA(Sheet3.Range("A6:A10000"))
If objPurchase = 0 Then
Me.DATATABLE.RowSource = ""
Else
Me.DATATABLE.RowSource = "Order!A6:H" & irow
End If
End Sub
```

```
' Update the status of the order based on stock and quantity
Private Sub StatusOrder()
On Error Resume Next
Dim IntStock, IntOrder As Integer
Dim UpdateStock As Object
Set UpdateStock = Sheet2.Range("B6:B10000").Find(What:=Me.CBID.value,
LookIn:=xlValues)
```

```
IntStock = UpdateStock.Offset(0, 2).value
IntOrder = UpdateStock.Offset(0, 4).value
```

```
If IntStock <= IntOrder Then
UpdateStock.Offset(0, 5).value = "Yes"
Else
UpdateStock.Offset(0, 5).value = "No"
End If
```

End Sub

' Update selected transaction details and stock

Private Sub CMDUPDATE_Click()

'Update details

Dim objUpdate As Object

Dim UpdateStock As Object

Set UpdateStock = Sheet2.Range("B6:B10000").Find(What:=Me.CBID.value,
LookIn:=xlValues)

Set objUpdate = Sheet3.Range("A6:A10000").Find(What:=Me.TXTNUMB1.value,
LookIn:=xlValues)

'Notify user data selected will be updated

If Me.TXTNUMB1.value = "" Then

Call MsgBox("Data to be updated not selected", vbInformation, "Update Details")

Else

objUpdate.Offset(0, 1).value = Me.TXTID.value

objUpdate.Offset(0, 2).value = Me.CBID.value

objUpdate.Offset(0, 3).value = Me.TXTNAME.value

objUpdate.Offset(0, 4).value = Me.TXTBRAND.value

objUpdate.Offset(0, 5).value = Me.TXTTOT.value

objUpdate.Offset(0, 6).value = Me.TXTTOTAL.value

objUpdate.Offset(0, 7).value = Format(CDate(Me.TXTPURDATE.value), "DD/MM/YYYY")

UpdateStock.Offset(0, 2).value = Val(UpdateStock.Offset(0, 2).value) -

Val(Me.TXTUPDATE.value) + Val(Me.TXTTOT.value)

'Clear fields

Call MsgBox("Details updated", vbInformation, "Update Details")

Me.TXTNUMB1.value = ""

Me.TXTID.value = ""

Me.CBID.value = ""

Me.TXTNAME.value = ""

Me.TXTBRAND.value = ""

Me.TXTTOT.value = ""

Me.TXTTOTAL.value = ""

Me.TXTPURDATE.value = ""

End If

' Validate Total

If Not IsNumeric(Me.TXTTOT.value) Or Me.TXTTOT.value <= 0 Then

 MsgBox "Please enter a positive numeric value for Total.", vbExclamation, "Invalid Input"

 Exit Sub

End If

' Validate Total Price

If Not IsNumeric(Me.TXTTOTAL.value) Or Me.TXTTOTAL.value <= 0 Then

 MsgBox "Please enter a positive numeric value for Total Price.", vbExclamation, "Invalid Input"

 Exit Sub

End If

' Validate Purchase Date

If Not IsDate(Me.TXTPURDATE.value) Then

 MsgBox "Please enter a valid date in the format DD/MM/YYYY for Purchase Date.", vbExclamation, "Invalid Input"

 Exit Sub

End If

End Sub

' Double-click event for DATATABLE to fill form fields with selected transaction data

Private Sub DATATABLE_DblClick(ByVal Cancel As MSForms.ReturnBoolean)

On Error GoTo Error

Dim ldata, CellActive As Long

'Fill in data to be updated

Me.TXTNUMB1.value = Me.DATATABLE.value

Me.TXTID.value = Me.DATATABLE.Column(1)

Me.CBID.value = Me.DATATABLE.Column(2)

Me.TXTNAME.value = Me.DATATABLE.Column(3)

Me.TXTBRAND.value = Me.DATATABLE.Column(4)

Me.TXTTOT.value = Me.DATATABLE.Column(5)

Me.TXTUPDATE.value = Me.DATATABLE.Column(5)

```

Me.TXTTOTAL.value = Me.DATATABLE.Column(6)
Me.TXTPURDATE.value = Format(CDate(Me.DATATABLE.Column(7)), "DD/MM/YYYY")
'Active columns in sheets
Sheet3.Select
ldata = Sheets("Order").Cells(Rows.Count, "A").End(xlUp).Row
Sheets("Order").Range("A6:A" & ldata).Find(What:=Me.TXTNUMB1.value,
LookIn:=xlValues, LookAt:=xlWhole).Activate
CellActive = ActiveCell.Row
Sheet3.Select
Exit Sub
Error:
Call MsgBox("The selected data will be deleted", vbInformation, "Delete Data")
End Sub

' Initialize the UserForm
Private Sub UserForm_Initialize()
Call GetData
Call GetID
End Sub

' Get product IDs for CBID
Private Sub GetID()
Dim lProduct As Long
Dim irow As Long
irow = Sheet2.Range("B" & Rows.Count).End(xlUp).Row
lProduct = Application.WorksheetFunction.CountA(Sheet2.Range("B6:B10000"))
If lProduct = 0 Then
Me.CBID.RowSource = ""
Else
Me.CBID.RowSource = "RepairStocks!B6:C" & irow
End If
End Sub

```

PackageForm

Option Explicit

```
' Check if the data to be deleted is selected
Private Sub CMDDEL_Click()
If Me.TXTNUMB1.value = "" Then
Call MsgBox("Please double click and select the data on the table.", vbInformation,
"Delete Details")
Else
Select Case MsgBox("Are you sure you want to delete the data?" _
& vbCrLf & "Please confirm." _
, vbYesNo Or vbQuestion Or vbDefaultButton1, "Delete Details")
Case vbNo
Exit Sub
Case vbYes
End Select
' Clear form data
Me.DATATABLE.value = ""
Me.TXTNUMB1.value = ""
Sheet1.Select
Selection.EntireRow.Delete
' Refresh data on the form
Call GetData

Sheet1.Select
End If

End Sub

' Reset input fields
Private Sub CMDRESET_Click()
' Clear input fields
Me.TXTPACKNAME.value = ""
Me.TXTPRICE.value = ""
Me.TXTDES.value = ""
Me.DATATABLE.value = ""
```



```
Me.TXTNUMB1.value = ""
```

```
End Sub
```

```
' Save package details
```

```
Private Sub CMDSAVE_Click()
```

```
    Dim objPack As Object
```

```
    ' Display on worksheet
```

```
    Set objPack = Sheet1.Range("A80000").End(xlUp)
```

```
    ' Prompt user to fill in all details
```

```
    If Me.TXTPACKNAME.value = "" Or Me.TXTPRICE.value = "" Or Me.TXTDES.value = "" Then
```

```
        ' Prompt user to input if empty
```

```
        Call MsgBox("Please fill in completely.", vbInformation, "Input Details")
```

```
        ' Save after complete
```

```
    Else
```

```
        ' Validate Price
```

```
        If Not IsValidPositiveInteger(Me.TXTPRICE.value) Then
```

```
            MsgBox "Please enter a valid positive integer for Price.", vbExclamation, "Input Error"
```

```
        Exit Sub
```

```
    If Not IsValidLetters(Me.TXTPACKNAME.value) Then
```

```
        MsgBox "Please enter a valid name with letters only.", vbExclamation, "Input Error"
```

```
    Exit Sub
```

```
End If
```

```
End If
```

```
' Save details if validation passes
```

```
objPack.Offset(1, 0).value = "=ROW()-ROW($A$5)"
```

```
objPack.Offset(1, 1).value = Me.TXTPACKNAME.value
```

```
objPack.Offset(1, 2).value = Me.TXTPRICE.value
```

```
objPack.Offset(1, 3).value = Me.TXTDES.value
```

```
Call MsgBox("Package details saved", vbInformation, "Input Details")
```

```
' Refresh data on the form
```

```

        Call GetData
        Call GetPackage
        ' Clear after save
        Me.TXTPACKNAME.value = ""
        Me.TXTPRICE.value = ""
        Me.TXTDES.value = ""
    End If
End Sub

' Get data for DATATABLE from the RepairPackage sheet
Private Sub GetData()
    Dim IPack As Long
    Dim irow As Long
    irow = Sheet1.Range("A" & Rows.Count).End(xlUp).Row
    IPack = Application.WorksheetFunction.CountA(Sheet1.Range("A6:A10000"))
    If IPack = 0 Then
        Me.DATATABLE.RowSource = ""
    Else
        Me.DATATABLE.RowSource = "RepairPackage!A6:D" & irow
    End If
End Sub

' Double-click event for DATATABLE to fill form fields with selected package data
Private Sub DATATABLE_Db1Click(ByVal Cancel As MSForms.ReturnBoolean)
    On Error GoTo Error
    Dim IUpdate, CellActive As Long
    ' Fill in the data to be updated
    Me.TXTNUMB1.value = Me.DATATABLE.value

    'Active column sheet
    Sheet1.Select
    IUpdate = Sheets("RepairPackage").Cells(Rows.Count, "A").End(xlUp).Row
    Sheets("RepairPackage").Range("A6:A" & IUpdate).Find(What:=Me.TXTNUMB1.value,
    LookIn:=xlValues, LookAt:=xlWhole).Activate
    CellActive = ActiveCell.Row
    Sheet1.Select
Exit Sub

```

Error:

Call MsgBox("The selected data will be deleted.", vbInformation, "Delete Details")

End Sub

' Initialize the UserForm

Private Sub UserForm_Initialize()

Call GetData

Call GetPackage

End Sub

' Function to get package details for populating a ComboBox

Private Sub GetPackage()

Dim IPack As Long

Dim irow As Long

irow = Sheet1.Range("B" & Rows.Count).End(xlUp).Row

IPack = Application.WorksheetFunction.CountA(Sheet1.Range("B6:B10000"))

If IPack = 0 Then

TransactionForm.CBPACKAGE.RowSource = ""

Else

TransactionForm.CBPACKAGE.RowSource = "RepairPackage!B6:D" & irow

End If

End Sub

' Validate if the input is a positive integer

Private Function IsValidPositiveInteger(ByVal value As Variant) As Boolean

Dim numericValue As Double

' Attempt to convert the value to a numeric type

If IsNumeric(value) Then

numericValue = CDBl(value)

' Check if the numeric value is greater than 0 and is an integer

IsValidPositiveInteger = numericValue > 0 And Int(numericValue) = numericValue

Else

' Return false if the value is not numeric

IsValidPositiveInteger = False

```
End If
End Function

' Validate if the input contains only letters
Private Function IsValidLetters(ByVal value As String) As Boolean
    Dim i As Integer
    IsValidLetters = Len(value) > 0

    ' Check if value if letter
    For i = 1 To Len(value)
        If Not IsLetter(Mid(value, i, 1)) Then
            IsValidLetters = False
            Exit Function
        End If
    Next i
End Function
```

ProductForm

Option Explicit

```
' Delete selected product details
Private Sub CMDDEL_Click()
If Me.TXTNUMB1.value = "" Then
Call MsgBox("Please double click and select data on the table.", vbInformation, "Delete
Details")
Else
Select Case MsgBox("The selected data will be deleted." _
& vbCrLf & "Please confirm." _
, vbYesNo Or vbQuestion Or vbDefaultButton1, "Delete Details")
Case vbNo
Exit Sub
Case vbYes
End Select
Me.DATATABLE.value = ""
Me.TXTNUMB1.value = ""
Sheet2.Select
Selection.EntireRow.Delete
Call GetData
Sheet2.Select
' Clear form fields
Me.TXTPRODID.value = ""
Me.TXTPRODNAME.value = ""
Me.TXTSTOCK.value = ""
Me.CBBRAND.value = ""
Me.TXTREORDER.value = ""
Me.TXTSTATUS.value = ""
End If

End Sub

' Reset input fields
Private Sub CMDRESET_Click()
' Clear form fields
```

```

Me.TXTPRODID.value = ""
Me.TXTPRODNAME.value = ""
Me.TXTSTOCK.value = ""
Me.CBBRAND.value = ""
Me.TXTREORDER.value = ""
Me.TXTSTATUS.value = ""
Me.DATATABLE.value = ""
Me.TXTNUMB1.value = ""

```

```

End Sub

```

```

' Save product details

```

```

Private Sub CMDSAVE_Click()

```

```

    Dim ObjProd As Object

```

```

    ' Display on worksheet

```

```

    Set ObjProd = Sheet2.Range("A80000").End(xlUp)

```

```

    ' Prompt user to fill completely

```

```

    If Me.TXTPRODID.value = "" Or Me.TXTPRODNAME.value = "" Or
Me.TXTSTOCK.value = "" _

```

```

        Or Me.CBBRAND.value = "" Or Me.TXTREORDER.value = "" Or
Me.TXTSTATUS.value = "" Then

```

```

        ' Prompt user to input if empty

```

```

        Call MsgBox("Please fill in the details completely.", vbInformation, "Input Details")

```

```

    ElseIf Not IsValidProductName(Me.TXTPRODNAME.value) Then

```

```

        ' Check if TXTPRODNAME is a valid product name

```

```

        MsgBox "Please enter a valid product name.", vbExclamation, "Input Error"

```

```

    Else

```

```

        ' Save after complete

```

```

        ObjProd.Offset(1, 0).value = "=ROW()-ROW($A$5)"

```

```

        ObjProd.Offset(1, 1).value = Me.TXTPRODID.value

```

```

        ObjProd.Offset(1, 2).value = Me.TXTPRODNAME.value

```

```

        ObjProd.Offset(1, 3).value = Me.TXTSTOCK.value

```

```

        ObjProd.Offset(1, 4).value = Me.CBBRAND.value

```

```

        ObjProd.Offset(1, 5).value = Me.TXTREORDER.value

```

```

        ObjProd.Offset(1, 6).value = Me.TXTSTATUS.value

```

```

    Call MsgBox("Package data is saved.", vbInformation, "Input Details")
    Call GetData
    ' Clear after save
    Me.TXTPRODID.value = ""
    Me.TXTPRODNAME.value = ""
    Me.TXTSTOCK.value = ""
    Me.CBBRAND.value = ""
    Me.TXTREORDER.value = ""
    Me.TXTSTATUS.value = ""
End If
End Sub

' Function to validate if the entered product name is valid
Private Function IsValidProductName(productName As String) As Boolean
    ' Allow letters, numbers, and spaces in the product name
    Dim regex As Object
    Set regex = CreateObject("VBScript.RegExp")

    ' Define the pattern for valid product name
    regex.Pattern = "^[a-zA-Z0-9 ]+$"

    ' Check if the input matches the pattern
    IsValidProductName = regex.Test(productName)
End Function

' Get data for DATATABLE from the RepairStocks sheet
Private Sub GetData()
    Dim IProd As Long
    Dim irow As Long
    irow = Sheet2.Range("A" & Rows.Count).End(xlUp).Row
    IProd = Application.WorksheetFunction.CountA(Sheet2.Range("A6:A10000"))
    If IProd = 0 Then
        Me.DATATABLE.RowSource = ""
    Else
        Me.DATATABLE.RowSource = "RepairStocks!A6:G" & irow
    End If
End Sub

```

End Sub

' Update selected product details

Private Sub CMDUPDATE_Click()

Dim objUpdate As Object

Set objUpdate = Sheet2.Range("A6:A10000").Find(What:=Me.TXTNUMB1.value,
LookIn:=xlValues)

'Notify user selected data will be updated

If Me.TXTNUMB1.value = "" Then

Call MsgBox("Data to be updated not selected.", vbInformation, "Update Details")

Else

objUpdate.Offset(0, 1).value = Me.TXTPRODID.value

objUpdate.Offset(0, 2).value = Me.TXTPRODNAME.value

objUpdate.Offset(0, 3).value = Me.TXTSTOCK.value

objUpdate.Offset(0, 4).value = Me.CBBRAND.value

objUpdate.Offset(0, 5).value = Me.TXTREORDER.value

objUpdate.Offset(0, 6).value = Me.TXTSTATUS.value

Call MsgBox("Details Updated", vbInformation, "Update Details")

' Clear input fields

Me.TXTPRODID.value = ""

Me.TXTPRODNAME.value = ""

Me.TXTSTOCK.value = ""

Me.CBBRAND.value = ""

Me.TXTREORDER.value = ""

Me.TXTSTATUS.value = ""

Me.TXTNUMB1.value = ""

End If

End Sub

' Double-click event for the data table

Private Sub DATATABLE_DbClick(ByVal Cancel As MSForms.ReturnBoolean)

On Error GoTo Error

Dim lUpdate, CellActive As Long

'Input details to be updated

Me.TXTNUMB1.value = Me.DATATABLE.value


```

Me.TXTPRODID.value = Me.DATATABLE.Column(1)
Me.TXTPRODNAME.value = Me.DATATABLE.Column(2)
Me.TXTSTOCK.value = Me.DATATABLE.Column(3)
Me.CBBRAND.value = Me.DATATABLE.Column(4)
Me.TXTREORDER.value = Me.DATATABLE.Column(5)
Me.TXTSTATUS.value = Me.DATATABLE.Column(6)

'Activate column sheet
Sheet2.Select
lUpdate = Sheets("RepairStocks").Cells(Rows.Count, "A").End(xlUp).Row
Sheets("RepairStocks").Range("A6:A" & lUpdate).Find(What:=Me.TXTNUMB1.value,
LookIn:=xlValues, LookAt:=xlWhole).Activate
CellActive = ActiveCell.Row
Sheet2.Select
Exit Sub
Error:
Call MsgBox("The selected data will be deleted.", vbInformation, "Delete Details")

End Sub

' Validate if the input is a positive integer
Private Sub TXTREORDER_Change()
    On Error Resume Next
    Dim IntOrder As Integer

    If Not IsPositiveInteger(Me.TXTREORDER.value) Then
        If Me.TXTREORDER.value <> "" Then ' Check if the value is not empty
            MsgBox "Please enter a valid positive integer for Reorder.", vbExclamation,
                "Input Error"
            Me.TXTREORDER.value = ""
        End If
    End If

    UpdateStatus
End Sub

```

```

' Validate if the input is a positive integer
Private Sub TXTSTOCK_Change()
    On Error Resume Next

    If Not IsPositiveInteger(Me.TXTSTOCK.value) Then
        If Me.TXTSTOCK.value <> "" Then ' Check if the value is not empty
            MsgBox "Please enter a valid positive integer for Stock.", vbExclamation, "Input
Error"
            Me.TXTSTOCK.value = ""
        End If
    End If

    UpdateStatus
End Sub

' Update product status based on stock and reorder values
Private Sub UpdateStatus()
    Dim IntStock, IntOrder As Integer

    IntStock = Me.TXTSTOCK.value
    IntOrder = Me.TXTREORDER.value

    If IntStock <= IntOrder Then
        Me.TXTSTATUS.value = "Yes"
    Else
        Me.TXTSTATUS.value = "No"
    End If
End Sub

' Validate if the input is a positive integer
Private Function IsPositiveInteger(ByVal value As Variant) As Boolean
    Dim numericValue As Double

    If IsNumeric(value) Then
        numericValue = CDBl(value)
        IsPositiveInteger = numericValue > 0 And Int(numericValue) = numericValue
    Else
        IsPositiveInteger = False
    End If
End Function

```

```

    End If
End Function

' Validate if the input contains only letters
Private Function IsValidLetters(ByVal value As Variant) As Boolean
    Dim i As Integer
    IsValidLetters = IsEmpty(value)

    If Not IsValidLetters Then
        For i = 1 To Len(value)
            If Not IsLetter(Mid(value, i, 1)) Then
                IsValidLetters = False
                Exit Function
            End If
        Next i

        IsValidLetters = True
    End If
End Function

'Elements for the this functionality is inspired from:
'https://exceloffthegrid.com/sorting-an-array-alphabetically-with-vba/
'Accessed Date: 29 December 2023
' Check if a character is a letter
Private Function IsLetter(ByVal character As String) As Boolean
    IsLetter = UCase(character) Like "[A-Z]"
End Function

' Initialize the UserForm
Private Sub UserForm_Initialize()
    With CBBRAND
        .AddItem "Acer"
        .AddItem "Apple"
        .AddItem "Dell"
        .AddItem "Lenovo"
        .AddItem "Others"
    End With

```

```
Call GetData  
End Sub
```

RepairsForm

Option Explicit

' Update form fields

Private Sub CBID_Change()

On Error GoTo ErrorHandler

Dim ObjProduct As Object

Set ObjProduct = Sheet2.Range("B6:B100000").Find(What:=Me.CBID.value,
LookIn:=xlValues)

' Find product details based on selected CBID and update form fields

Me.TXTNAME.value = ObjProduct.Offset(0, 1).value

Me.TXTSTOCK.value = ObjProduct.Offset(0, 2).value

Me.TXTBrand.value = ObjProduct.Offset(0, 3).value

Exit Sub

ErrorHandler:

Call MsgBox("Product ID not found.", vbInformation, "Product ID")

End Sub

' Create new ID

Private Sub CMDNEW_Click()

Sheet8.Range("C9").value = Sheet8.Range("C9").value + 1

If Sheet8.Range("C8").value = 1 Then

Me.TXTID.value = "TR-100" & Sheet8.Range("C9").value

End If

If Sheet8.Range("C8").value = 2 Then

Me.TXTID.value = "TR-10" & Sheet8.Range("C9").value

End If

If Sheet8.Range("C8").value = 3 Then

Me.TXTID.value = "TR-1" & Sheet8.Range("C9").value

End If

If Sheet8.Range("C8").value = 4 Then

```

Me.TXTID.value = "TR-" & Sheet8.Range("C9").value
End If
If Sheet8.Range("C8").value = 5 Then
Me.TXTID.value = "TR-" & Sheet8.Range("C9").value
End If

End Sub

```

```

'Delete selected usage details
Private Sub CMDDEL_Click()
Dim objUpdate As Object
Set objUpdate = Sheet2.Range("B6:B10000").Find(What:=Me.CBID.value,
LookIn:=xlValues)
If Me.TXTNUMB1.value = "" Then
Call MsgBox("Please double the data on the table", vbInformation, "Delete Details")
Else
Select Case MsgBox("Are you sure you want to delete the data?" _
& vbCrLf & "Please confirm." _
, vbYesNo Or vbQuestion Or vbDefaultButton1, "Delete Details")
Case vbNo
Exit Sub
Case vbYes
End Select
Me.DATATABLE.value = ""
Me.TXTNUMB1.value = ""
Sheet3.Select
Selection.EntireRow.Delete
Call GetData
objUpdate.Offset(0, 2).value = Val(objUpdate.Offset(0, 2).value) -
Val(Me.TXTTOT.value)
Call StatusOrder
Me.TXTUSED.value = ""
Me.TXTNUMB1.value = ""
Me.TXTID.value = ""
Me.CBID.value = ""
Me.TXTNAME.value = ""

```

```
Me.TXTBrand.value = ""
Me.TXTTOT.value = ""
Me.TXTUSAGEDATE.value = ""
Sheet3.Select
End If
```

```
End Sub
```

```
' Reset input fields
Private Sub CMDRESET_Click()
Me.TXTID.value = ""
Me.CBID.value = ""
Me.TXTNAME.value = ""
Me.TXTBrand.value = ""
Me.TXTTOT.value = ""
Me.TXTUSAGEDATE.value = ""
Me.TXTUSED.value = ""
```

```
End Sub
```

```
' Save usage details to the Usage sheet
Private Sub CMDSAVE_Click()
Dim objPC As Object
Dim UpdateStock As Object
' Display on worksheet
Set objPC = Sheet7.Range("A80000").End(xlUp)
Set UpdateStock = Sheet2.Range("B6:B10000").Find(What:=Me.CBID.value,
LookIn:=xlValues)
```

```
' Prompt user to fill completely
If Me.TXTID.value = "" _
Or Me.CBID.value = "" _
Or Me.TXTNAME.value = "" _
Or Me.TXTBrand.value = "" _
Or Me.TXTTOT.value = "" _
Or Me.TXTUSAGEDATE.value = "" Then
```

```
' Prompt user if not filled
```

```

Call MsgBox("Please fill in the details completely.", vbInformation, "Input Details")
' Save the data to the Usage sheet
Else
objPC.Offset(1, 0).value = "=ROW()-ROW($A$5)"
objPC.Offset(1, 1).value = Me.TXTID.value
objPC.Offset(1, 2).value = Me.CBID.value
objPC.Offset(1, 3).value = Me.TXTNAME.value
objPC.Offset(1, 4).value = Me.TXTBrand.value
objPC.Offset(1, 5).value = Me.TXTTOT.value
objPC.Offset(1, 6).value = Format(CDate(Me.TXTUSAGEDATE.Text), "MM/DD/YYYY")

```

```

UpdateStock.Offset(0, 2).value = Val(UpdateStock.Offset(0, 2).value) -
Val(Me.TXTTOT.value)

```

```

Call MsgBox("Repair Product Details have been saved", vbInformation, "Input Details")
Call GetData
Call StatusOrder
' Clear after save
Me.TXTID.value = ""
Me.CBID.value = ""
Me.TXTNAME.value = ""
Me.TXTBrand.value = ""
Me.TXTTOT.value = ""
Me.TXTUSAGEDATE.value = ""
Me.TXTUSED.value = ""
End If

```

```

End Sub

```

```

'Update selected usage details

```

```

Private Sub CMDUPDATE_Click()
    Dim objUpdate As Object
    Dim UpdateStock As Object

```

```

Set UpdateStock = Sheet2.Range("B6:B10000").Find(What:=Me.CBID.value,
LookIn:=xlValues)

```



```
Set objUpdate = Sheet7.Range("A6:A10000").Find(What:=Me.TXTNUMB1.value,  
LookIn:=xlValues)
```

```
    ' Notify user that selected data will be updated
```

```
If Me.TXTNUMB1.value = "" Then
```

```
Call MsgBox("Data to be updated not selected.", vbInformation, "Update Details")
```

```
Else
```

```
objUpdate.Offset(0, 1).value = Me.TXTID.value
```

```
objUpdate.Offset(0, 2).value = Me.CBID.value
```

```
objUpdate.Offset(0, 3).value = Me.TXTNAME.value
```

```
objUpdate.Offset(0, 4).value = Me.TXTBrand.value
```

```
objUpdate.Offset(0, 5).value = Me.TXTTOT.value
```

```
objUpdate.Offset(0, 6).value = Format(CDate(Me.TXTUSAGEDATE.value),  
"MM/DD/YYYY")
```

```
UpdateStock.Offset(0, 2).value = Val(UpdateStock.Offset(0, 2).value) -
```

```
Val(Me.TXTTOT.value) + Val(Me.TXTUSED.value)
```

```
Call StatusOrder
```

```
Call MsgBox("Details updated.", vbInformation, "Update Details")
```

```
Me.TXTNUMB1.value = ""
```

```
Me.TXTID.value = ""
```

```
Me.CBID.value = ""
```

```
Me.TXTNAME.value = ""
```

```
Me.TXTBrand.value = ""
```

```
Me.TXTTOT.value = ""
```

```
Me.TXTUSAGEDATE.value = ""
```

```
Me.TXTUSED.value = ""
```

```
End If
```

```
End Sub
```

```
' Validate TXTTOT input for a positive integer
```

```
Private Sub TXTTOT_Change()
```

```
    On Error Resume Next
```

```
    ' Check if the input is not numeric or not positive
```

```
    If Not IsNumeric(Me.TXTTOT.value) Or Val(Me.TXTTOT.value) <= 0 Then
```

```

        ' Display error message if the input is not a positive integer and the field is not
empty
        If Me.TXTTOT.value <> "" Then
            MsgBox "Please enter a valid positive integer for the amount used.",
vbExclamation, "Input Error"
        End If
        Me.TXTTOT.value = ""
    End If

    ' Re-enable the default error handling
    On Error GoTo 0
End Sub

```

```

' Validate TXTUSAGEDATE input for a valid date format
Private Sub TXTUSAGEDATE_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    ' Validate the date format
    If Not IsDate(Me.TXTUSAGEDATE.value) Then
        ' Display error message for an invalid date format
        MsgBox "Please enter a valid date format (MM/DD/YYYY) for the usage date.",
vbExclamation, "Input Error"
        Me.TXTUSAGEDATE.value = ""
    End If
End Sub

```

```

' Double-click event for the data table
Private Sub DATATABLE_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
    On Error GoTo ErrorHandler
    Dim lUpdate, CellActive As Long
    'Input details to be updated
    Me.TXTNUMB1.value = Me.DATATABLE.value
    Me.TXTID.value = Me.DATATABLE.Column(1)
    Me.CBID.value = Me.DATATABLE.Column(2)
    Me.TXTTOT.value = Me.DATATABLE.Column(5)
    Me.TXTUSED.value = Me.DATATABLE.Column(5)
    Me.TXTUSAGEDATE.value = Format(CDate(Me.DATATABLE.Column(6)),
"DD/MM/YYYY")

```

```

'Active column sheet
Sheet7.Select
IUpdate = Sheets("Usage").Cells(Rows.Count, "A").End(xlUp).Row
Sheets("Usage").Range("A6:A" & IUpdate).Find(What:=Me.TXTNUMB1.value,
LookIn:=xlValues, LookAt:=xlWhole).Activate
CellActive = ActiveCell.Row
Sheet7.Select
Exit Sub
ErrorHandler:
Call MsgBox("Selected data will be deleted.", vbInformation, "Delete Details")

End Sub

```

```

' Initialize the UserForm
Private Sub UserForm_Initialize()
Call GetID
Call GetData
End Sub
' Populate CBID with data from RepairStocks sheet
Private Sub GetID()
Dim IProd As Long
Dim irow As Long
irow = Sheet2.Range("B" & Rows.Count).End(xlUp).Row
IProd = Application.WorksheetFunction.CountA(Sheet2.Range("B6:B10000"))
If IProd = 0 Then
Me.CBID.RowSource = ""
Else
Me.CBID.RowSource = "RepairStocks!B6:C" & irow
End If
End Sub

```

```

' Populate DATATABLE with data from Usage sheet
Private Sub GetData()
Dim IPur As Long
Dim irow As Long
irow = Sheet7.Range("A" & Rows.Count).End(xlUp).Row
IPur = Application.WorksheetFunction.CountA(Sheet7.Range("A6:A10000"))

```

```

If IPur = 0 Then
Me.DATATABLE.RowSource = ""
Else
Me.DATATABLE.RowSource = "Usage!A6:G" & irow
End If
End Sub

' Update the status of the order based on stock and order quantity
Private Sub StatusOrder()
On Error Resume Next
Dim IntStock, IntOrder As Integer
Dim objUpdate As Object
Set objUpdate = Sheet2.Range("B6:B10000").Find(What:=Me.CBID.value,
LookIn:=xlValues)

IntStock = objUpdate.Offset(0, 2).value
IntOrder = objUpdate.Offset(0, 4).value

If IntStock <= IntOrder Then
objUpdate.Offset(0, 5).value = "Yes"
Else
objUpdate.Offset(0, 5).value = "No"
End If

End Sub

```

StaffForm

Option Explicit

' Delete selected staff details

Private Sub CMDDEL_Click()

If Me.TXTNUMB1.value = "" Then

Call MsgBox("Please double click and select the data on the table.", vbInformation, "Delete Details")

Else

Select Case MsgBox("Are you sure you want to delete the data?" _
& vbCrLf & "Please confirm." _
, vbYesNo Or vbQuestion Or vbDefaultButton1, "Delete Details")

Case vbNo

Exit Sub

Case vbYes

End Select

' Clear fields and delete entire row

Me.DATATABLE.value = ""

Me.TXTNUMB1.value = ""

Sheet6.Select

Selection.EntireRow.Delete

Call GetData

Sheet6.Select

End If

End Sub

' Reset input fields

Private Sub CMDRESET_Click()

' Clear all input fields

Me.TXTNUMB1.value = ""

Me.TXTNAME.value = ""

Me.CBGENDER.value = ""

Me.TXTADDRESS.value = ""

Me.TXTPHONE.value = ""

Me.DATATABLE.value = ""

End Sub

' Save staff details to the Staff sheet

Private Sub CMDSAVE_Click()

Dim objStaff As Object

Set objStaff = Sheet6.Range("A80000").End(xlUp)

' Prompt user to fill required details

If Me.TXTNAME.value = "" _

Or Me.CBGENDER.value = "" _

Or Me.TXTADDRESS.value = "" _

Or Me.TXTPHONE.value = "" Then

' Prompt user to fill required details

Call MsgBox("Please fill in details completely.", vbInformation, "Input Details")

'Save if complete

Else

' Save data to the Staff sheet

objStaff.Offset(1, 0).value = "=ROW()-ROW(\$A\$5)"

objStaff.Offset(1, 1).value = Me.TXTNAME.value

objStaff.Offset(1, 2).value = Me.CBGENDER.value

objStaff.Offset(1, 3).value = Me.TXTADDRESS.value

objStaff.Offset(1, 4).value = Me.TXTPHONE.value

Call MsgBox("Staff Details is saved.", vbInformation, "Input Details")

Call GetData

'Clear after save

Me.TXTNUMB1.value = ""

Me.TXTNAME.value = ""

Me.CBGENDER.value = ""

Me.TXTADDRESS.value = ""

Me.TXTPHONE.value = ""

End If

End Sub

```

' Update selected staff details
Private Sub CMDUPDATE_Click()
Dim objUpdate As Object
Set objUpdate = Sheet6.Range("A6:A10000").Find(What:=Me.TXTNUMB1.value,
LookIn:=xlValues)
'Notify user selected data will be updated
If Me.TXTNUMB1.value = "" Then
Call MsgBox("Data to be updated not selected.", vbInformation, "Update Details")
Else
objUpdate.Offset(0, 1).value = Me.TXTNAME.value
objUpdate.Offset(0, 2).value = Me.CBGENDER.value
objUpdate.Offset(0, 3).value = Me.TXTADDRESS.value
objUpdate.Offset(0, 4).value = Me.TXTPHONE.value
Call MsgBox("Staff details updated.", vbInformation, "Update Details")

' Clear forms
Me.TXTNUMB1.value = ""
Me.TXTNAME.value = ""
Me.CBGENDER.value = ""
Me.TXTADDRESS.value = ""
Me.TXTPHONE.value = ""
End If

End Sub

' Validate TXTPHONE input for a positive numeric value
Private Sub TXTPHONE_Change()
' Disable error handling temporarily
On Error Resume Next

' Check if the input is not numeric or not positive
If Not IsNumeric(Me.TXTPHONE.value) Or Val(Me.TXTPHONE.value) <= 0 Then
' Display error message only if the input is not a positive integer
MsgBox "Please enter a valid phone number.", vbExclamation, "Input Error"
Me.TXTPHONE.value = ""
End If

```

```
' Re-enable the default error handling
On Error GoTo 0
End Sub
```

```
' Validate TXTNAME input for allowing letters and spaces
Private Sub TXTNAME_Change()
    If Not IsValidName(Me.TXTNAME.value) Then
        MsgBox "Please enter a valid name.", vbExclamation, "Input Error"
        Me.TXTNAME.value = ""
    End If
End Sub
```

```
'Elements for the this functionality is inspired from:
'https://answers.microsoft.com/en-us/msoffice/forum/all/single-stepping-a-user-
defined-function-in-vba-for/b5d6ea18-40bf-45d5-ac07-73bfc8571f1f
'Accessed Date: 30 December 2023
' Function to validate
Private Function IsValidName(inputString As String) As Boolean
    Dim i As Integer
    For i = 1 To Len(inputString)
        ' Check if each character is a letter or a space
        If Not (IsCharacterLetter(Mid(inputString, i, 1)) Or Mid(inputString, i, 1) = " ")
Then
            IsValidName = False
            Exit Function
        End If
    Next i
    IsValidName = True
End Function
```

```
'Elements for the this functionality is inspired from:
'https://techniclee.wordpress.com/2010/07/21/isletter-function-for-vba/
'Accessed Date: 30 December 2023
' Function to check if a character is a letter
Private Function IsCharacterLetter(ByVal character As String) As Boolean
    IsCharacterLetter = (character Like "[A-Za-z]")
```


End Function

' Double-click event for the data table

Private Sub DATATABLE_DblClick(ByVal Cancel As MSForms.ReturnBoolean)

On Error GoTo ErrorHandler

Dim IUpdate, CellActive As Long

' Set values based on the selected row in the datatable

Me.TXTNUMB1.value = Me.DATATABLE.value

Me.TXTNAME.value = Me.DATATABLE.Column(1)

Me.CBGENDER.value = Me.DATATABLE.Column(2)

Me.TXTADDRESS.value = Me.DATATABLE.Column(3)

Me.TXTPHONE.value = Me.DATATABLE.Column(4)

' Active worksheet column

Sheet6.Select

IUpdate = Sheets("Customer").Cells(Rows.Count, "A").End(xlUp).Row

Sheets("Customer").Range("A6:A" & IUpdate).Find(What:=Me.TXTNUMB1.value,

LookIn:=xlValues, LookAt:=xlWhole).Activate

CellActive = ActiveCell.Row

' Active worksheet column

Sheet6.Select

Exit Sub

ErrorHandler:

Call MsgBox("Please double click and select data to be deleted.", vbInformation, "Delete Details")

End Sub

' Initialize the UserForm

Private Sub UserForm_Initialize()

With CBGENDER

.AddItem "Male"

.AddItem "Female"

.AddItem "Others"

End With

Call GetData

End Sub

```
' Populate DATATABLE with data from Staff sheet
Private Sub GetData()
' Populate the datatable with existing staff data
Dim IStaff As Long
Dim irow As Long
irow = Sheet6.Range("A" & Rows.Count).End(xlUp).Row
IStaff = Application.WorksheetFunction.CountA(Sheet6.Range("A6:A10000"))
If IStaff = 0 Then
Me.DATATABLE.RowSource = ""
Else
Me.DATATABLE.RowSource = "Staff!A6:E" & irow
End If
End Sub
```

StatusForm

Option Explicit

' Update repair status and payment method for selected item

Private Sub CMDUPDATE_Click()

Dim objUpdate As Object

Set objUpdate = Sheet11.Range("A6:A1000000").Find(What:=Me.TXTNUMB1.value,
LookIn:=xlValues)

If Me.CBSTATUS.value = "" Then

Call MsgBox("Please select repair status first.", vbInformation, "Status")

Else

objUpdate.Offset(0, 6).value = Me.CBSTATUS.value

objUpdate.Offset(0, 7).value = Me.CBPAYMENT.value

End If

Unload Me

End Sub

' Initialize the UserForm

Private Sub UserForm_Initialize()

With CBSTATUS

.AddItem "Repairing process"

.AddItem "Completed"

.AddItem "Keeping"

.AddItem "Collected"

End With

With CBPAYMENT

.AddItem "Cash"

.AddItem "Card"

End With

End Sub

TransactionForm

Option Explicit

' Initialize the user form by calling various subroutines

Private Sub UserForm_Initialize()

Call GetPackage

Call GetInvoice

Call Calculate

Call GetCustomer

Call GetReport

Call GetStatus

' Set initial properties for the MultiPage control

Me.MultiPage1.Style = fmTabStyleNone

Me.MultiPage1.value = 0

' Populate dropdowns with predefined values

With CMUSER

.AddItem "Jason Jules"

.AddItem "Sherlyn Simmons"

.AddItem "Molly Monroe"

.AddItem "Admin"

End With

With CBPAY

.AddItem "Cash"

.AddItem "Card"

End With

With CBSTATUS

.AddItem "Repairing process"

.AddItem "Completed"

.AddItem "Keeping"

.AddItem "Collected"

End With

' Disable various buttons and dropdowns initially

```
' These will be enabled when the user logs in
Me.CMDTRAN.Enabled = False
Me.CMDOPEN.Enabled = False
Me.CMDREPORT.Enabled = False
Me.CMDPRODUCT.Enabled = False
Me.CMDUSER.Enabled = False
Me.CMDCUST.Enabled = False
Me.CMDPACK.Enabled = False
Me.CMDREPAIRPROD.Enabled = False
Me.CMDORDER.Enabled = False
Me.CMDSTATUS.Enabled = False
```

```
End Sub
```

```
' Trigger event for the user dropdown change
Private Sub CMUSER_Change()
' Set default passwords based on the selected user
If Me.CMUSER.value = "Jason Jules" Then
Me.TXTPASSCHECK.value = "jason123"
End If
```

```
If Me.CMUSER.value = "Sherlyn Simmons" Then
Me.TXTPASSCHECK.value = "sherlyn123"
End If
```

```
If Me.CMUSER.value = "Molly Monroe" Then
Me.TXTPASSCHECK.value = "molly123"
End If
If Me.CMUSER.value = "Admin" Then
Me.TXTPASSCHECK.value = "admin123"
End If
End Sub
```

```
' Trigger event for the Login button
Private Sub CMDLOGIN_Click()
' Check for empty user or password fields
If Me.TXTPASS.value = "" Or Me.CMUSER.value = "" Then
```

```

        Call MsgBox("User and Password cannot be empty, please input correctly.",
vbCritical, "Incorrect Input")
        Me.CMUSER.value = ""
        Me.TXTPASS.value = ""
        ElseIf Me.TXTPASS.value <> Me.TXTPASSCHECK.value Then
            Call MsgBox("Incorrect Password, please try again.", vbCritical, "Incorrect
Password")
            Me.CMUSER.value = ""
            Me.TXTPASS.value = ""
        Else

```

```

' Enable controls upon successful login
Me.LBUSER.Caption = "Welcome " & Me.CMUSER.value
Me.TXTSTAFF.value = Me.CMUSER.value
Me.CMUSER.Enabled = False
Me.TXTPASS.Enabled = False
Me.CMDTRAN.Enabled = True
Me.CMDOPEN.Enabled = True
Me.CMDREPORT.Enabled = True
Me.CMDPRODUCT.Enabled = True
Me.CMDUSER.Enabled = True
Me.CMDCUST.Enabled = True
Me.CMDPACK.Enabled = True
Me.CMDREPAIRPROD.Enabled = True
Me.CMDORDER.Enabled = True
Me.CMDSTATUS.Enabled = True
End If

```

```

End Sub

```

```

' Trigger event for the Reset button
Private Sub CMDRESET_Click()
' Disable various buttons and enable login controls upon reset
Me.CMDTRAN.Enabled = False
Me.CMDOPEN.Enabled = False
Me.CMDREPORT.Enabled = False
Me.CMDPRODUCT.Enabled = False

```

```
Me.CMDUSER.Enabled = False
Me.CMDCUST.Enabled = False
Me.CMDPACK.Enabled = False
Me.CMDREPAIRPROD.Enabled = False
Me.CMDORDER.Enabled = False
Me.CMDSTATUS.Enabled = False
Me.CMUSER.Enabled = True
Me.TXTPASS.Enabled = True
```

```
Me.LBUSER.Caption = ""
Me.CMUSER.value = ""
Me.TXTPASS.value = ""
```

```
End Sub
```

```
' Trigger event for the Transaction button
```

```
Private Sub CMDTRAN_Click()
```

```
Me.MultiPage1.value = 1
```

```
End Sub
```

```
' Trigger event for the Open button
```

```
Private Sub CMDOPEN_Click()
```

```
Application.Visible = True
```

```
Unload Me
```

```
Sheet5.Select
```

```
End Sub
```

```
' Trigger event for the Report button
```

```
Private Sub CMDREPORT_Click()
```

```
Me.MultiPage1.value = 2
```

```
End Sub
```

```
' Trigger event for the Status button
```

```
Private Sub CMDSTATUS_Click()
```

```
Me.MultiPage1.value = 3
```

```
Me.LBTOTALDATA.Caption = Me.TABLESTATUS.ListCount & "Data"
```

```
End Sub
```

' Trigger event for the Home button

Private Sub CMDHOME_Click()

Me.MultiPage1.value = 0

End Sub

Private Sub CMDPRODUCT_Click()

ProductForm.Show

End Sub

Private Sub CMDUSER_Click()

StaffForm.Show

End Sub

Private Sub CMDCUST_Click()

CustomerForm.Show

End Sub

Private Sub CMDPACK_Click()

PackageForm.Show

End Sub

Private Sub CMDREPAIRPROD_Click()

RepairsForm.Show

End Sub

Private Sub CMDORDER_Click()

OrderForm.Show

End Sub

Private Sub CMDSAVE_Click()

ThisWorkbook.Save

End Sub

Private Sub CMDEXIT_Click()


```

' Prompt the user to confirm before exiting the application
Select Case MsgBox("Are you sure you want to quit the application?" _
& vbCrLf & "Please confirm." _
, vbYesNo Or vbQuestion Or vbDefaultButton1, "Exit")
Case vbNo
Exit Sub
Case vbYes
End Select
ThisWorkbook.Save
Application.Visible = True
Unload Me
ThisWorkbook.Close

End Sub

' Subroutine to populate the Customer dropdown in the TransactionForm
Private Sub GetCustomer()
Dim ICust, irow As Long
irow = Sheet4.Range("B" & Rows.Count).End(xlUp).Row
ICust = Application.WorksheetFunction.CountA(Sheet4.Range("B6:B10000"))
If ICust = 0 Then
TransactionForm.CUSTTABLE.RowSource = ""
Else
TransactionForm.CUSTTABLE.RowSource = "Customer!A6:D" & irow
End If
End Sub

' Subroutine to populate the Package dropdown in the TransactionForm
Private Sub GetPackage()
Dim IPack As Long
Dim irow As Long
irow = Sheet1.Range("B" & Rows.Count).End(xlUp).Row
IPack = Application.WorksheetFunction.CountA(Sheet1.Range("B6:B10000"))
If IPack = 0 Then
TransactionForm.CBPACKAGE.RowSource = ""
Else
TransactionForm.CBPACKAGE.RowSource = "RepairPackage!B6:D" & irow

```

```
End If
End Sub
```

```
' Event handler for double-clicking on a row in the Customer Data Table in
TransactionForm
Private Sub CUSTTABLE_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
On Error GoTo ErrorHandler
If Me.TXTID.value = "" Then
Call MsgBox("Please fill in the Transaction ID data first.", vbInformation, "Transaction
ID")
Else
```

```
Me.CBCUSTOMER.value = Me.CUSTTABLE.Column(1)
Me.TXTADDRESS.value = Me.CUSTTABLE.Column(2)
Me.TXTPHONE.value = Me.CUSTTABLE.Column(3)
End If
```

```
Exit Sub
ErrorHandler:
Call MsgBox("Please double click on the Customer Data Table.", vbInformation, "Search
Details")
```

```
End Sub
```

```
'Elements for this functionality is inspired from:
'https://excelmacromastery.com/vba-advanced-filter/
'Accessed Date: 27 December 2023
' Trigger event for the Search button in the TransactionForm
Private Sub CMDSEARCH_Click()
'Prompt error handler
On Error GoTo ErrorHandler
Dim SearchData As Object
'Find sheet
Set SearchData = Sheet4
Me.DATATABLE.value = ""
' Set the search criteria
Sheet4.Range("F5").value = "Customer Name"
```

```

Sheet4.Range("F6").value = Me.TXTFIND.value
' Perform advanced filter to search data based on criteria
SearchData.Range("A6").CurrentRegion.AdvancedFilter Action:=xlFilterCopy,
CriteriaRange:= _
Sheet4.Range("F5:F6"), Copytorange:=Sheet4.Range("H5:K5"), Unique:=True
' ' Call the FoundResults subroutine to display the search results
Call FoundResults
'Prmopt error message
Exit Sub
ErrorHandler:
Call MsgBox("Data not found.", vbInformation, "Search Details")

```

```

End Sub

```

```

' Subroutine to display the search results in the CUSTTABLE
Private Sub FoundResults()
Dim ICust, irow As Long
irow = Sheet4.Range("H" & Rows.Count).End(xlUp).Row
ICust = Application.WorksheetFunction.CountA(Sheet4.Range("H6:H90000"))
If ICust = 0 Then
Me.CUSTTABLE.RowSource = ""
Else
Me.CUSTTABLE.RowSource = "Customer!H6:K" & irow
End If
End Sub

```

```

' Reset input fields
Private Sub CMDRESETTABLE_Click()
Me.TXTFIND.value = ""
Me.CUSTTABLE.value = ""
Call GetCustomer
End Sub

```

```

' Trigger event for the New Transaction button in the TransactionForm
Private Sub CMDNEW_Click()
Select Case MsgBox("Do you want to make a new transaction?" _
& vbCrLf & "Please confirm." _

```

```
, vbYesNo Or vbQuestion Or vbDefaultButton1, "New Transaction")
```

```
Case vbNo
```

```
Exit Sub
```

```
Case vbYes
```

```
End Select
```

```
Sheet9.Range("A14:E26").ClearContents
```

```
Sheet9.Range("B7:D10").ClearContents
```

```
Sheet9.Range("B27:D30").ClearContents
```

```
Me.TXTID.value = ""
```

```
Me.CBCUSTOMER.value = ""
```

```
Me.TXTADDRESS.value = ""
```

```
Me.TXTPHONE.value = ""
```

```
Me.TXTIN.value = ""
```

```
Me.TXTOUT.value = ""
```

```
Me.TXTPACKAGE.value = ""
```

```
Me.TXTNUMB.value = ""
```

```
Me.TXTSESSION.value = ""
```

```
Me.TXTAMOUNT.value = ""
```

```
Me.TXTPAY.value = ""
```

```
' Increment the transaction ID
```

```
Sheet8.Range("C5").value = Sheet8.Range("C5").value + 1
```

```
If Sheet8.Range("C4").value = 1 Then
```

```
    Me.TXTID.value = "TR-100" & Sheet8.Range("C5").value
```

```
End If
```

```
If Sheet8.Range("C4").value = 2 Then
```

```
    Me.TXTID.value = "TR-10" & Sheet8.Range("C5").value
```

```
End If
```

```
If Sheet8.Range("C4").value = 3 Then
```

```
    Me.TXTID.value = "TR-1" & Sheet8.Range("C5").value
```

```
End If
```

```
If Sheet8.Range("C4").value = 4 Then
```

```
    Me.TXTID.value = "TR-0" & Sheet8.Range("C5").value
```

```
End If
```

```
If Sheet8.Range("C4").value = 5 Then
```

```
Me.TXTID.value = "TR-" & Sheet8.Range("C5").value
End If
```

```
Sheet8.Range("C12").value = ""
Sheet8.Range("C13").value = ""
End Sub
```

```
' Trigger event for the "Starting date" textbox in the TransactionForm
Private Sub TXTIN_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    CheckAndClearTXT Me.TXTIN
End Sub
```

```
'Elements for this functionality is inspired from:
```

```
`https://learn.microsoft.com/en-us/office/vba/language/reference/user-interface-
help/keydown-keyup-events
```

```
'Accessed Date: 08 January 2024
```

```
' Trigger event for the "Completion date" textbox in the TransactionForm
```

```
Private Sub TXTOUT_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift
As Integer)
```

```
    If KeyCode = vbKeyReturn Then
        CheckAndClearTXT Me.TXTOUT
```

```
        If IsValidDate(Me.TXTIN.value) And IsValidDate(Me.TXTOUT.value) Then
            If CDate(Me.TXTOUT.value) < CDate(Me.TXTIN.value) Then
                MsgBox "Completion date cannot be earlier than the starting date.",
vbExclamation
```

```
                Me.TXTOUT.value = "" ' Clear the wrong input
            End If
        End If
    End If
End Sub
```

```
Private Sub TXTNUMB_AfterUpdate()
    On Error Resume Next
```

```
    Dim A As Currency
    Dim B As Long
```

Static showError As Boolean

' Reset the flag when user starts a new input

showError = False

' Validate input for positive integers

If IsNumeric(Me.TXTNUMB.value) Then

 B = CLng(Me.TXTNUMB.value)

 If B <= 0 Then

 If Not showError Then

 MsgBox "Please enter a number greater than zero for the number of computers.", vbExclamation

 showError = True

 End If

 Me.TXTNUMB.value = ""

 Exit Sub

 ElseIf B > 20 Then

 If Not showError Then

 MsgBox "Maximum allowed number is 20.", vbExclamation

 showError = True

 End If

 Me.TXTNUMB.value = ""

 Exit Sub

 End If

Else

 If Not showError Then

 MsgBox "Please enter a valid number for the number of computers.", vbExclamation

 showError = True

 End If

 Me.TXTNUMB.value = ""

 Exit Sub

End If

A = Me.TXTSESSION.value

Me.TXTSESSION.value = Format(Me.TXTSESSION.value, "#,###")

```

Me.TXTAMOUNT.value = B * A
Me.TXTAMOUNT.value = Format(Me.TXTAMOUNT.value, "#,###")
End Sub

```

```

Private Sub CBPACKAGE_DbClick(ByVal Cancel As MSForms.ReturnBoolean)
On Error GoTo Handler
' Check if details input correctly
If Me.TXTID.value = "" _
Or Me.CBCUSTOMER.value = "" Then
Call MsgBox("Please fill in the ID and Customer information before proceeding.",
vbInformation, "Customer Details")
Else

```

```

Me.TXTPACKAGE.value = Me.CBPACKAGE.value
Me.TXTSESSION.value = Me.CBPACKAGE.Column(1)
End If

```

```

Exit Sub
Handler:
Call MsgBox("Sorry, the item ID is not registered.", vbInformation, "Product ID")
End Sub

```

```

Private Sub CMDSAV_Click()
Dim objTran, objInv As Object

```

```

' Set references to the last used rows in "Transaction" and "Invoice" sheets
Set objTran = Sheet5.Range("A1000000").End(xlUp)
Set objInv = Sheet9.Range("A1000000").End(xlUp)

```

```

' Check if all required fields are filled
If Me.TXTID.value = "" _
Or Me.CBCUSTOMER.value = "" _
Or Me.CBPACKAGE.value = "" _
Or Me.TXTIN.value = "" _
Or Me.TXTOUT.value = "" _
Or Me.TXTNUMB.value = "" Then

```

```

Call MsgBox("Please fill in completely.", vbInformation, "Input Details")
' Display a message if any required field is empty
Else

' Save transaction details to the TRANSACTION sheet
objTran.Offset(1, 0).value = "=ROW()-ROW($A$5)"
objTran.Offset(1, 1).value = Me.TXTID.value
objTran.Offset(1, 2).value = Me.TXTSTAFF.value
objTran.Offset(1, 3).value = Me.CBCCUSTOMER.value
objTran.Offset(1, 4).value = Me.TXTADDRESS.value
objTran.Offset(1, 5).value = Me.TXTPHONE.value
objTran.Offset(1, 6).value = Format(CDate(Me.TXTIN.value), "MM/DD/YYYY")
objTran.Offset(1, 7).value = Format(CDate(Me.TXTOUT.value), "MM/DD/YYYY")
objTran.Offset(1, 7).value = Format(CDate(Me.TXTOUT.value), "MM/DD/YYYY")

objTran.Offset(1, 8).value = Me.TXTPACKAGE.value
objTran.Offset(1, 9).value = Me.TXTNUMB.value
objTran.Offset(1, 10).value = CDec(Me.TXTSESSION.value)
objTran.Offset(1, 11).value = CDec(Me.TXTAMOUNT.value)
objTran.Offset(1, 12).value =
Application.WorksheetFunction.CountA(Sheet5.Range("A6:A1000000"))

' Save invoice details to the Invoice sheet
objInv.Offset(1, 0).value = Me.TXTPACKAGE.value
objInv.Offset(1, 1).value = CDec(Me.TXTSESSION.value)
objInv.Offset(1, 2).value = Me.TXTNUMB.value
objInv.Offset(1, 3).value = CDec(Me.TXTAMOUNT.value)
objInv.Offset(1, 4).value =
Application.WorksheetFunction.CountA(Sheet11.Range("A6:A1000000"))

Call MsgBox("Transaction details has been saved.", vbInformation, "Transaction")
Call GetInvoice
Call Calculate
Call GetReport

' Clear form fields after successful data storage

```



```
Me.TXTPACKAGE.value = ""  
Me.TXTNUMB.value = ""  
Me.TXTSESSION.value = ""  
Me.TXTAMOUNT.value = ""  
Me.TXTPAY.value = ""
```

```
End If
```

```
End Sub
```

```
'Subroutine to get and display invoice data
```

```
Private Sub GetInvoice()
```

```
Dim ITran As Long
```

```
Dim irow As Long
```

```
irow = Sheet9.Range("A" & Rows.Count).End(xlUp).Row
```

```
ITran = Application.WorksheetFunction.CountA(Sheet9.Range("A14:A1000000"))
```

```
If ITran = 0 Then
```

```
Me.DATATABLE.RowSource = ""
```

```
Else
```

```
Me.DATATABLE.RowSource = "Invoice!A14:E" & irow
```

```
End If
```

```
End Sub
```

```
' Subroutine to calculate the total amount
```

```
Private Sub Calculate()
```

```
Dim dblamt As Double
```

```
Dim r As Long
```

```
dblamt = 0
```

```
With DATATABLE
```

```
For r = 0 To .ListCount - 1
```

```
dblamt = dblamt + .List(r, 3)
```

```
Next r
```

```
End With
```

```
Me.TXTTOTPPAID.value = dblamt
```

```
End Sub
```

```
Private Sub TXTAMOUNT_Change()
```

```
On Error Resume Next
Me.TXTAMOUNT.value = Format(Me.TXTAMOUNT.value, "#,###")
```

```
End Sub
```

```
Private Sub TXTSESSION_Change()
```

```
On Error Resume Next
```

```
Dim A As Currency
```

```
Dim B As Long
```

```
A = Me.TXTSESSION.value
```

```
B = Me.TXTNUMB.value
```

```
Me.TXTSESSION.value = Format(Me.TXTSESSION.value, "#,###")
```

```
Me.TXTAMOUNT.value = B * A
```

```
Me.TXTAMOUNT.value = Format(Me.TXTAMOUNT.value, "#,###")
```

```
End Sub
```

```
Private Sub GetReport()
```

```
Dim ICust, irow As Long
```

```
' Get the last row with data in column A on "Transaction" sheet
```

```
irow = Sheet5.Range("A" & Rows.Count).End(xlUp).Row
```

```
' Count the non-empty cells in the range A6:A1000000 on "Transaction" sheet
```

```
ICust = Application.WorksheetFunction.CountA(Sheet5.Range("A6:A1000000"))
```

```
If ICust = 0 Then
```

```
TransactionForm.REPORTTABLE.RowSource = ""
```

```
Else
```

```
TransactionForm.REPORTTABLE.RowSource = "Transaction!A6:M" & irow
```

```
End If
```

```
End Sub
```

```
Private Sub TXTTOTPPAID_Change()
```

```
On Error Resume Next
```

```
Me.TXTTOTPPAID.value = Format(Me.TXTTOTPPAID.value, "#,###")
```

```
End Sub
```

```

Private Sub TXTPAY_Change()
Dim A As Currency
Dim B As Double

' Validate input and allow only positive integers
If IsNumeric(Me.TXTPAY.value) And Me.TXTPAY.value > 0 And InStr(Me.TXTPAY.value,
".") = 0 Then
B = CDBl(Me.TXTPAY.value)
A = Me.TXTTOTPPAID.value

Me.TXTPAY.value = Format(B, "#,###")
Me.TXTRETUR.value = B - A
Me.TXTRETUR.value = Format(Me.TXTRETUR.value, "#,###")
Else
' Clear the input if it's not a valid positive integer
Me.TXTPAY.value = ""
End If
End Sub

'Double-click event for the data table
Private Sub DATATABLE_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
On Error GoTo ErrorHandler
Dim DataSource, CellActive As Long
' Retrieve and display selected data
Me.TXTNUMB1.value = Me.DATATABLE.value
Me.TXTNUMB2.value = Me.DATATABLE.Column(4)
Me.TXTPACKAGE.value = Me.DATATABLE.value
Me.TXTSESSION.value = Me.DATATABLE.Column(1)
Me.TXTNUMB.value = Me.DATATABLE.Column(2)
Me.TXTAMOUNT.value = Me.DATATABLE.Column(3)

Exit Sub
ErrorHandler:
Call MsgBox("Selected data will be deleted", vbInformation, "Delete Details")
End Sub

Private Sub CMDDELETE_Click()

```

Dim DataSource, CellActive As Long

' Check if necessary fields are filled

If Me.TXTNUMB1.value = "" Or Me.TXTNUMB2.value = "" Then

 MsgBox "Please double click and select data from the table.", vbInformation,
 "Delete Details"

 Exit Sub

End If

' Confirm deletion with a message box

Select Case MsgBox("Do you want to delete the data?" & vbCrLf & "Please confirm.",
vbYesNo Or vbQuestion Or vbDefaultButton1, "Delete Details")

Case vbNo

Exit Sub

Case vbYes

' Continue with deletion

End Select

' Delete from "Transaction" sheet

Sheet5.Select

DataSource = Sheets("Transaction").Cells(Rows.Count, "M").End(xlUp).Row

Dim foundCellTransaction As Range

Set foundCellTransaction = Sheets("Transaction").Range("M6:M" &

DataSource).Find(What:=Me.TXTNUMB2.value, LookIn:=xlValues, LookAt:=xlWhole)

If Not foundCellTransaction Is Nothing Then

CellActive = foundCellTransaction.Row

foundCellTransaction.EntireRow.Delete

Else

MsgBox "Value not found in the 'Transaction' sheet.", vbInformation, "Not Found"

End If

' Delete from "Invoice" sheet

Sheet9.Select

DataSource = Sheets("Invoice").Cells(Rows.Count, "A").End(xlUp).Row

Dim foundCellInvoice As Range

```
Set foundCellInvoice = Sheets("Invoice").Range("A14:A" &  
DataSource).Find(What:=Me.TXTNUMB1.value, LookIn:=xlValues, LookAt:=xlWhole)
```

```
If Not foundCellInvoice Is Nothing Then
```

```
CellActive = foundCellInvoice.Row
```

```
foundCellInvoice.EntireRow.Delete
```

```
Else
```

```
MsgBox "Value not found in the 'Invoice' sheet.", vbInformation, "Not Found"
```

```
End If
```

```
' Reset UI and perform other actions
```

```
Call GetInvoice
```

```
Call Calculate
```

```
Sheet5.Select
```

```
End Sub
```

```
Private Sub CMDUPDATE_Click()
```

```
Dim objTran, objInv As Object
```

```
' Set references to the rows containing the specified values in "Transaction" and  
"Invoice" sheets
```

```
Set objTran = Sheet5.Range("A6:A1000000").Find(What:=Me.TXTNUMB2.value,  
LookIn:=xlValues)
```

```
Set objInv = Sheet9.Range("A14:A100").Find(What:=Me.TXTNUMB1.value,  
LookIn:=xlValues)
```

```
' Check if required fields are filled
```

```
If Me.TXTID.value = "" _
```

```
Or Me.CBCUSTOMER.value = "" _
```

```
Or Me.CBPACKAGE.value = "" _
```

```
Or Me.TXTIN.value = "" _
```

```
Or Me.TXTOUT.value = "" _
```

```
Or Me.TXTNUMB.value = "" Then
```

```
' Display a message if any required field is empty
```

Call MsgBox("Please double click and select the transaction.", vbInformation, "Details Selection")

'Save after complete details

Else

' Update data on "Transaction" sheet

objTran.Offset(0, 1).value = Me.TXTID.value

objTran.Offset(0, 2).value = Me.TXTSTAFF.value

objTran.Offset(0, 3).value = Me.CBCUSTOMER.value

objTran.Offset(0, 4).value = Me.TXTADDRESS.value

objTran.Offset(0, 5).value = Me.TXTPHONE.value

objTran.Offset(0, 6).value = Format(CDate(Me.TXTIN.value), "MM/DD/YYYY")

objTran.Offset(0, 7).value = Format(CDate(Me.TXTOUT.value), "MM/DD/YYYY")

objTran.Offset(0, 8).value = Me.TXTPACKAGE.value

objTran.Offset(0, 9).value = Me.TXTNUMB.value

objTran.Offset(0, 10).value = CDec(Me.TXTSESSION.value)

objTran.Offset(0, 11).value = CDec(Me.TXTAMOUNT.value)

objTran.Offset(0, 12).value =

Application.WorksheetFunction.CountA(Sheet5.Range("A6:A1000000"))

objInv.Offset(0, 0).value = Me.TXTPACKAGE.value

objInv.Offset(0, 1).value = CDec(Me.TXTSESSION.value)

objInv.Offset(0, 2).value = Me.TXTNUMB.value

objInv.Offset(0, 3).value = CDec(Me.TXTAMOUNT.value)

objInv.Offset(0, 4).value =

Application.WorksheetFunction.CountA(Sheet5.Range("A6:A1000000"))

Call MsgBox("Transaction details is updated.", vbInformation, "Update Details")

Call GetInvoice

Call Calculate

' Clear form fields after successful data update

Me.TXTPACKAGE.value = ""

Me.TXTNUMB.value = ""

Me.TXTSESSION.value = ""

Me.TXTAMOUNT.value = ""

Me.TXTPAY.value = ""

End If

End Sub

'Elements for this functionality is inspired from:

'https://www.youtube.com/watch?v=xXNu_u_DwfM&ab_channel=Dr.GerardVerschuuren

'Accessed Date: 05 January 2024

Private Sub CMDPRINT_Click()

Dim objPrint As Object

' Set reference to the last used row in "Invoice" sheet

Set objPrint = Sheet9.Range("A1000").End(xlUp)

' Call SaveReport subroutine to save report data

Call SaveReport

' Check if essential fields are filled before printing

If Me.TXTID.value = "" _

Or Me.CBCUSTOMER.value = "" _

Or Me.TXTADDRESS.value = "" _

Or Me.TXTPHONE.value = "" _

Or Me.TXTIN.value = "" _

Or Me.TXTOUT.value = "" _

Or Me.CBPAY.value = "" _

Or Me.TXTPAY.value = "" Then

Call MsgBox("The transaction details is not complete, please recheck.", vbInformation, "Transaction")

Else

Select Case MsgBox("Do you want to print this transaction?" _

& vbCrLf & "Please confirm." _

, vbYesNo Or vbQuestion Or vbDefaultButton1, "Print Details")

Case vbNo

Exit Sub

Case vbYes

End Select

Sheet9.Range("B7").value = Me.CBCUSTOMER.value

```

Sheet9.Range("B8").value = Me.TXTADDRESS.value
Sheet9.Range("B9").value = Val(Me.TXTPHONE.value)
Sheet9.Range("B10").value = Me.TXTID.value
Sheet9.Range("B26").value = "Payment method"
Sheet9.Range("D26").value = Me.CBPAY.value
Sheet9.Range("B27").value = "Total"
Sheet9.Range("D27").value = CDec(Me.TXTTOTPPAID.value)
Sheet9.Range("B28").value = "Pay"
On Error Resume Next
Sheet9.Range("D28").value = CDec(Me.TXTPAY.value)
Sheet9.Range("B29").value = "Return"
On Error Resume Next

```

```

Sheet9.Range("D29").value = CDec(Me.TXTRETUR.value)
Sheet9.Range("B30").value = "Staff"
Sheet9.Range("D30").value = Me.TXTSTAFF.value

```

```

' Print the "Invoice" sheet
Sheet9.PrintOut
Sheet9.Range("A14:E26").ClearContents
Me.TXTID.value = ""
Me.CBCUSTOMER.value = ""
Me.TXTADDRESS.value = ""
Me.TXTPHONE.value = ""
Me.TXTIN.value = ""
Me.TXTOUT.value = ""
Me.TXTPACKAGE.value = ""
Me.TXTNUMB.value = ""
Me.TXTSESSION.value = ""
Me.TXTAMOUNT.value = ""
Me.TXTPAY.value = ""
Me.TXTRETUR.value = ""
Me.TXTTOTPPAID.value = ""
Me.CBPAY.value = ""
End If

```

```

End Sub

```



```

' Subroutine for saving the report data
Private Sub SaveReport()
Dim objTran As Object
Dim objInv As Object

Set objTran = Sheet11.Range("A1000000").End(xlUp)
Set objInv = Sheet9.Range("A1000000").End(xlUp)

' Check if essential fields are filled before saving
If Me.TXTID.value = "" _
Or Me.CBCUSTOMER.value = "" _
Or Me.CBPACKAGE.value = "" _
Or Me.TXTIN.value = "" _
Or Me.TXTOUT.value = "" Then
' Display a message if any required field is empty
Call MsgBox("Please fill in the details correctly.", vbInformation, "Input Details")
Else
objTran.Offset(1, 0).value = "=ROW()-ROW($A$5)"
objTran.Offset(1, 1).value = Me.TXTID.value
objTran.Offset(1, 2).value = Me.TXTSTAFF.value
objTran.Offset(1, 3).value = Me.CBCUSTOMER.value
objTran.Offset(1, 4).value = Me.TXTADDRESS.value
objTran.Offset(1, 5).value = Me.TXTPHONE.value
objTran.Offset(1, 6).value = "Repairing process"
objTran.Offset(1, 7).value = Me.CBPAY.value
objTran.Offset(1, 8).value =
Application.WorksheetFunction.CountA(Sheet11.Range("A6:A1000000"))
' Call the GetStatus subroutine to update the status
Call GetStatus
End If

End Sub

' Subroutine for retrieving and displaying status data
Private Sub GetStatus()
Dim IStatus As Long

```

```

Dim irow As Long
irow = Sheet11.Range("A" & Rows.Count).End(xlUp).Row
IStatus = Application.WorksheetFunction.CountA(Sheet11.Range("A6:A1000000"))
If IStatus = 0 Then
TransactionForm.TABLESTATUS.RowSource = ""
Else
TransactionForm.TABLESTATUS.RowSource = "Status!A6:N" & irow
End If
End Sub

```

```

' Reset button click event
Private Sub CMDRESETT_Click()
Select Case MsgBox("Are you sure you want to reset this transaction?" _
& vbCrLf & "Please confirm." _
, vbYesNo Or vbQuestion Or vbDefaultButton1, "New Transaction")
Case vbNo
Exit Sub
Case vbYes
End Select

```

```

' Clear contents of specific ranges
Sheet9.Range("A14:E26").ClearContents
Sheet9.Range("B7:D10").ClearContents
Sheet9.Range("B27:D30").ClearContents

```

```

' Clear form fields
Me.TXTID.value = ""
Me.CBCUSTOMER.value = ""
Me.TXTADDRESS.value = ""
Me.TXTPHONE.value = ""
Me.TXTIN.value = ""
Me.TXTOUT.value = ""
Me.TXTPACKAGE.value = ""
Me.TXTNUMB.value = ""
Me.TXTSESSION.value = ""
Me.TXTAMOUNT.value = ""
Me.TXTPAY.value = ""

```

```
Me.TXTTOTPPAID.value = ""  
Me.CBPAY.value = ""  
Me.TXTRETUR.value = ""
```

```
End Sub
```

```
' Add button click event  
Private Sub CMDADD_Click()  
PackageForm.Show  
End Sub
```

```
' Trigger event for the "Starting date"  
Private Sub TXTSTART_Exit(ByVal Cancel As MSForms.ReturnBoolean)  
    CheckAndClearTXT Me.TXTSTART  
End Sub
```

```
' Trigger event for the "Completion date"  
Private Sub TXTEND_Exit(ByVal Cancel As MSForms.ReturnBoolean)  
    CheckAndClearTXT Me.TXTEND  
  
    If IsValidDate(Me.TXTSTART.value) And CDate(Me.TXTEND.value) <  
CDate(Me.TXTSTART.value) Then  
        MsgBox "Completion date cannot be earlier than the starting date.",  
vbExclamation  
        Me.TXTEND.value = "" ' Clear the wrong input  
    End If  
End Sub
```

```
' Function to check for invalid characters and show error message  
Sub CheckAndClearTXT(txtBox As MSForms.TextBox)  
    If Not IsValidInput(txtBox.value) Then  
        MsgBox "Please enter a valid date in the format mm/dd/yyyy.", vbExclamation  
        txtBox.value = "" ' Clear the wrong input  
    End If  
End Sub
```

```
' Function to validate if the entered date is in the correct format
```

```

Private Function IsValidDate(dateString As String) As Boolean
    On Error Resume Next
    Dim testDate As Date
    testDate = CDate(dateString)
    IsValidDate = (Err.Number = 0)
    On Error GoTo 0
End Function

```

'The following sub routine was inspired by the following websites:

'<https://stackoverflow.com/questions/7929205/vba-regular-expression-to-match-date>

'Accessed Date: 05 January 2024

' Function to check for invalid characters

```

Private Function IsValidInput(inputString As String) As Boolean
    Dim regex As Object
    Set regex = CreateObject("VBScript.RegExp")

```

' Define the pattern for valid date format

```

regex.Pattern = "^([0-9]|1[0-2])/([0-9]|1[2]|0-9)|3[01])/d{4}$"

```

' Check if the input matches the pattern

```

IsValidInput = regex.Test(inputString)

```

```

End Function

```

```

Private Sub CMDALL_Click()

```

' Error handling

```

On Error GoTo ErrorHandler

```

```

Dim objFind As Object

```

```

Set objFind = Sheet5

```

```

Me.DATATABLE.value = ""

```

```

Sheet10.Range("O6").value = ""

```

```

Sheet10.Range("P6").value = ""

```

' Perform AdvancedFilter to search for unique data based on criteria

```

objFind.Range("A5").CurrentRegion.AdvancedFilter Action:=xlFilterCopy,

```

```

CriteriaRange:= _

```

```

Sheet10.Range("O5:P6"), Copytorange:=Sheet10.Range("A5:M5"), Unique:=True

```

```

' Display the ordered data and report the result
Call OrderedData
Call ReportResult
Me.LBTOTAL.Caption = Sheet10.Range("R6").value
Me.LBTOTAL.Caption = Format(Me.LBTOTAL.Caption, "?#,###")
'Error handling
Exit Sub
ErrorHandler:
Call MsgBox("Data not found.", vbInformation, "Search Details")

End Sub

' Subroutine for reporting the search result
Private Sub ReportResult()
Dim IReport As Long
Dim irow As Long
irow = Sheet10.Range("A" & Rows.Count).End(xlUp).Row
IReport = Application.WorksheetFunction.CountA(Sheet10.Range("A6:A900000"))
If IReport = 0 Then
Me.REPORTTABLE.RowSource = ""
Else
Me.REPORTTABLE.RowSource = "SearchTransaction!A6:M" & irow
End If
End Sub

Private Sub CMDFIND_Click()
On Error GoTo Error
Dim objFind As Object
'Locate sheet object
Set objFind = Sheet5
Me.DATATABLE.value = ""
'Input criteria of finding
Sheet10.Range("O6").value = ">=" & Format(CDate(Me.TXTSTART.value),
"MM/DD/YYYY")
Sheet10.Range("P6").value = "<=" & Format(CDate(Me.TXTEND.value),
"MM/DD/YYYY")
'Prompt findings

```

```

objFind.Range("A5").CurrentRegion.AdvancedFilter Action:=xlFilterCopy,
CriteriaRange:= _
Sheet10.Range("O5:P6"), Copytorange:=Sheet10.Range("A5:M5"), Unique:=True
'Input to find
Call OrderedData
Call ReportResult
Me.LBTOTAL.Caption = Sheet10.Range("R6").value
Me.LBTOTAL.Caption = Format(Me.LBTOTAL.Caption, "?#,###")
'Prompt error
Exit Sub
Error:
Call MsgBox("Data not found.", vbInformation, "Search Details")

```

```

End Sub

```

```

Private Sub CMDRESE_Click()
' Call GetReport to reset search criteria
Call GetReport
Me.TXTSTART.value = ""
Me.TXTEND.value = ""
Me.LBTOTAL.Caption = Sheet5.Range("L3").value
Me.LBTOTAL.Caption = Format(Me.LBTOTAL.Caption, "?#,###")
End Sub

```

```

' Double-click event on the TABLESTATUS to view details

```

```

Private Sub TABLESTATUS_DbClick(ByVal Cancel As MSForms.ReturnBoolean)
On Error GoTo Error

```

```

' Populate StatusForm with details from TABLE
With StatusForm
.TXTID.value = Me.TABLESTATUS.Column(1)
.TXTNAME.value = Me.TABLESTATUS.Column(3)
.TXTADDRESS.value = Me.TABLESTATUS.Column(4)
.TXTPHONE.value = Me.TABLESTATUS.Column(5)
.CBSTATUS.value = Me.TABLESTATUS.Column(6)
.CBPAYMENT.value = Me.TABLESTATUS.Column(7)
.TXTNUMB1.value = Me.TABLESTATUS.Column(8)

```

```

End With
StatusForm.Show
Exit Sub
Error:
Call MsgBox("Please double click on the selected data in the table.", vbInformation,
"Select details")

End Sub

```

```

Private Sub CBSTATUS_Change()
On Error GoTo Error
Dim objFind As Object
'Display worksheet
Set objFind = Sheet11
Me.TABLESTATUS.value = ""
'Input critetia
Sheet12.Range("K6").value = Me.CBSTATUS.value
'Find
objFind.Range("A5").CurrentRegion.AdvancedFilter Action:=xlFilterCopy,
CriteriaRange:= _
Sheet12.Range("K5:K6"), Copytorange:=Sheet12.Range("A5:I5"), Unique:=True
'Input find
Call OrderedData
Call StatusResult
Me.LBTOTALDATA.Caption = Me.TABLESTATUS.ListCount & " Data"
'Error message
Exit Sub
Error:
Call MsgBox("Data not found.", vbInformation, "Search details")

End Sub

```

```

'Display te result of status search
Private Sub StatusResult()
Dim IStatus As Long
Dim irow As Long
irow = Sheet12.Range("A" & Rows.Count).End(xlUp).Row

```

```
IStatus = Application.WorksheetFunction.CountA(Sheet12.Range("A6:A900000"))  
If IStatus = 0 Then  
Me.TABLESTATUS.RowSource = ""  
Else  
Me.TABLESTATUS.RowSource = "SearchStatus!A6:I" & irow  
End If  
End Sub
```

```
Private Sub CMDRESET2_Click()  
Call GetStatus  
Me.CBSTATUS.value = ""  
Me.LBTOTALDATA.Caption = Me.TABLESTATUS.ListCount & " data"  
End Sub
```


Module

OrderedMod

Option Explicit

Sub OrderedData()

' Disable screen

Application.ScreenUpdating = False

' Select the Sheet12 worksheet

Sheet12.Select

' Sort the data in columns A to I

Sheet12.Range("A5:I1000000").Sort KEY1:=Range("I5"), Order1:=xlAscending,

Header:=xlYes

Sheet12.Select

End Sub

' Hide Excel window

Sub OpenForm()

Application.Visible = False

' Show the TransactionForm

TransactionForm.Show

End Sub