

ICT3611 | ASSIGNMENT 5

Ruan van der Merwe
STUDENT NUMBER: 69723400

Chapter 3 - Programming Project 4 – Airplane Seating Chart

Code

```
Public Class AirplaneSeatingChartForm
```

```
    Dim Seats As New List(Of FlightRow)
```

```
    Dim RowCount As Integer = 15
```

```
    Dim currSeat As Integer = 1
```

```
    Dim SelectedRow As New FlightRow(-1)
```

```
    Dim SelectedSeat As String
```

```
Public Sub New()
```

```
    InitializeComponent()
```

```
    FlightSeatsListView.FullRowSelect = True
```

```
    FlightSeatsListView.MultiSelect = False
```

```
    FlightSeatsListView.Columns.Add("A", 50).TextAlign = HorizontalAlignment.Center
```

```
    FlightSeatsListView.Columns.Add("B", 50).TextAlign = HorizontalAlignment.Center
```

```
    FlightSeatsListView.Columns.Add("C", 50).TextAlign = HorizontalAlignment.Center
```

```
    FlightSeatsListView.Columns.Add("", 50).TextAlign = HorizontalAlignment.Center
```

```
    FlightSeatsListView.Columns.Add("D", 50).TextAlign = HorizontalAlignment.Center
```

```
    FlightSeatsListView.Columns.Add("E", 50).TextAlign = HorizontalAlignment.Center
```

```
    FlightSeatsListView.Columns.Add("F", 50).TextAlign = HorizontalAlignment.Center
```

```
For I = 1 To RowCount
```

```
    Seats.Add(New FlightRow(I))
```

```
Next
```

```
For Each seat As FlightRow In Seats
```

```
    Dim arr As String() = New String(7) {}
```

```
    arr(0) = "."
```

```
    arr(1) = "."
```

```
    arr(2) = "."
```

```
    arr(3) = " "
```

```
    arr(4) = "."
```

```
    arr(5) = "."
```

```
    arr(6) = "."
```

```
    Dim item As New ListViewItem(arr)
```

```
    FlightSeatsListView.Items.Add(item)
```

```
Next
```

```
End Sub
```

```
Private Sub FlightSeatsListView_SelectedIndexChanged(sender As Object, e As EventArgs) Handles FlightSeatsListView.SelectedIndexChanged
```

```
    If FlightSeatsListView.SelectedItems.Count > 0 Then
```

```
        Dim selectedItem = FlightSeatsListView.Items(FlightSeatsListView.SelectedItems(0).Index).SubItems
```

```
        CurrSeatN.Text = FlightSeatsListView.SelectedItems(0).Index + 1
```

```
        CurrSeatA.Text = selectedItem(0).Text
```

```
        CurrSeatB.Text = selectedItem(1).Text
```

```
        CurrSeatC.Text = selectedItem(2).Text
```

```
        CurrSeatD.Text = selectedItem(4).Text
```

```
        CurrSeatE.Text = selectedItem(5).Text
```

```
        CurrSeatF.Text = selectedItem(6).Text
```

```
    End If
```

```
End Sub
```

```
Private Sub CurrSeat_Click(sender As Object, e As EventArgs) Handles CurrSeatA.Click, CurrSeatB.Click, CurrSeatC.Click, CurrSeatD.Click, CurrSeatE.Click, CurrSeatF.Click
```

```
    With DirectCast(sender, TextBox)
```

```
        SelectedSeat = CStr(.Tag)
```

```
    End With
```

```

Dim frm As New SeatAssignForm
frm.callerForm = Me
frm.ShowDialog()
End Sub

Public Sub UpdateSelectedSeat(seatType As String)
If FlightSeatsListView.SelectedItems.Count > 0 Then
    Dim selectedItem = FlightSeatsListView.Items(FlightSeatsListView.SelectedItems(0).Index).SubItems
    Select Case SelectedSeat
        Case "A"
            CurrSeatA.Text = seatType
            selectedItem(0).Text = CurrSeatA.Text
        Case "B"
            CurrSeatB.Text = seatType
            selectedItem(1).Text = CurrSeatB.Text
        Case "C"
            CurrSeatC.Text = seatType
            selectedItem(2).Text = CurrSeatC.Text
        Case "D"
            CurrSeatD.Text = seatType
            selectedItem(4).Text = CurrSeatD.Text
        Case "E"
            CurrSeatE.Text = seatType
            selectedItem(5).Text = CurrSeatE.Text
        Case "F"
            CurrSeatF.Text = seatType
            selectedItem(6).Text = CurrSeatF.Text
        Case Else
            Debug.WriteLine("No Selected Seat")
    End Select
End If
End Sub

Private Sub RefreshStats_Click(sender As Object, e As EventArgs) Handles RefreshStats.Click
Dim seatsFilled = 0
Dim windowsAvailable = 0
Dim RegMeals = 0
Dim LowCalMeals = 0
Dim VeggieMeals = 0
For Each row As ListViewItem In FlightSeatsListView.Items
    Dim currIndex As Integer = 0
    For Each item As ListViewItem.ListViewSubItem In row.SubItems
        If (currIndex = 0 Or currIndex = 6) And item.Text Is "."
            windowsAvailable += 1

        End If
        Select Case item.Text
            Case "R"
                seatsFilled += 1
                RegMeals += 1
            Case "V"
                seatsFilled += 1
                VeggieMeals += 1
            Case "L"
                seatsFilled += 1
                LowCalMeals += 1
        End Select
        currIndex += 1
    Next
Next
SeatsFilledLbl.Text = seatsFilled
WindowsAvalLbl.Text = windowsAvailable
RegMealLbl.Text = RegMeals

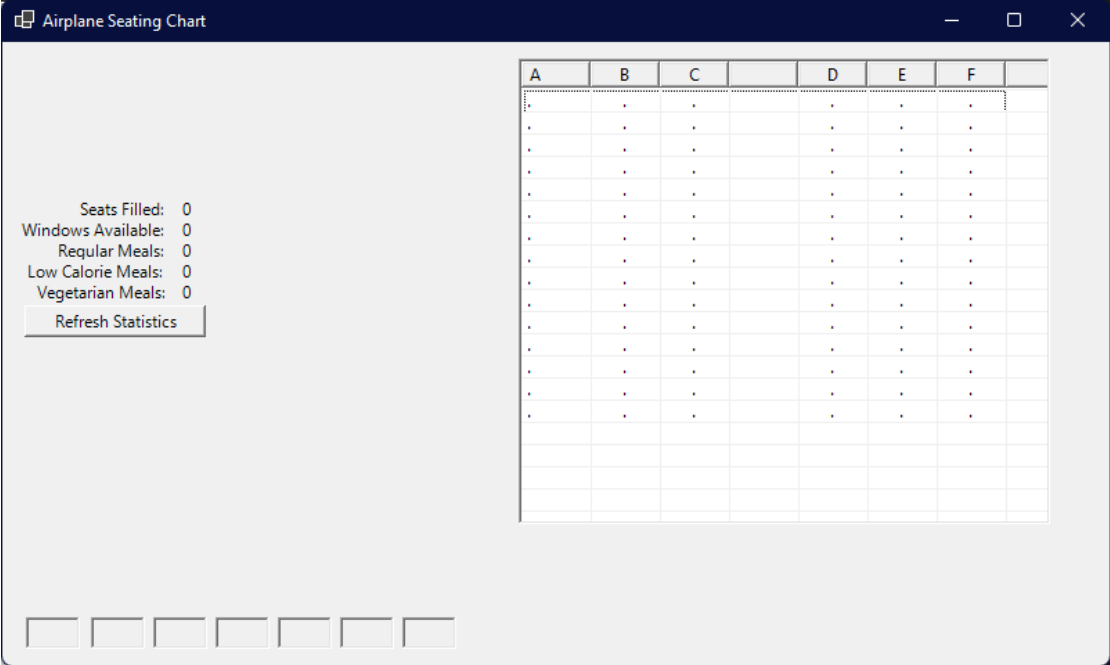
```

```
        LowCalMealLbl.Text = LowCalMeals
        VegMealLbl.Text = VeggieMeals
    End Sub
End Class
```

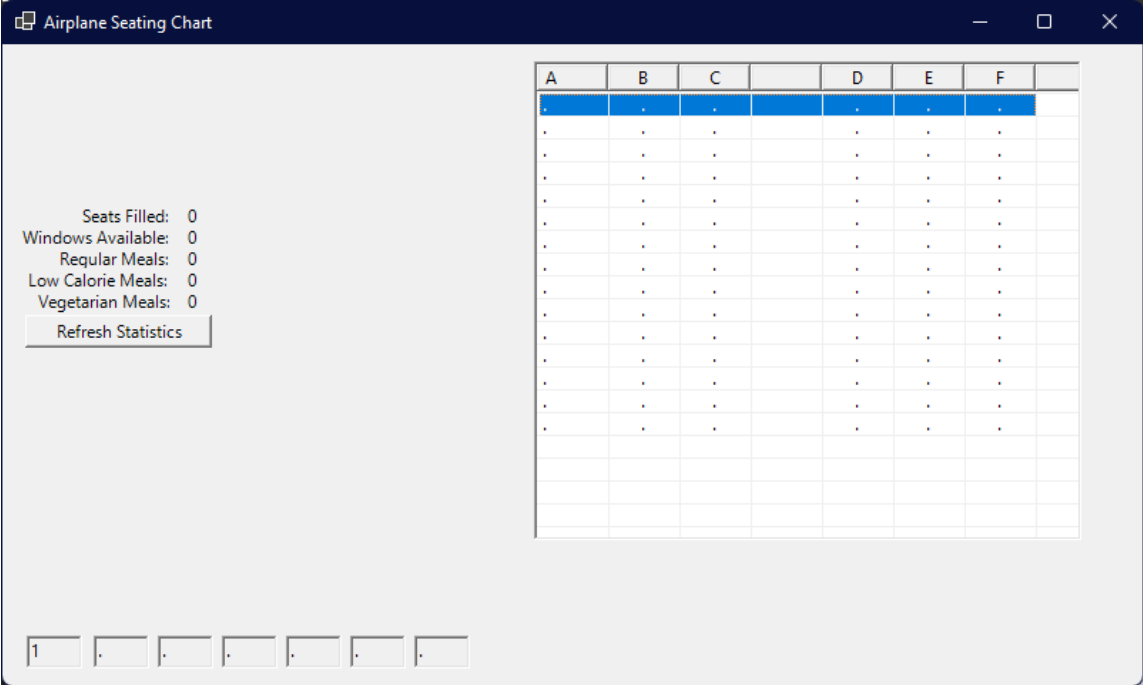
```
Public Class FlightRow
    Public A As String = "."
    Public B As String = "."
    Public C As String = "."
    Public D As String = "."
    Public E As String = "."
    Public F As String = "."
    Public RowNum As Integer
    Public Sub New(rowNum As Integer)
        rowNum = rowNum
    End Sub
End Class
```

Screenshots

On Start Up



Seat Row Click



The screenshot shows the 'Airplane Seating Chart' application. On the left, there are statistics: 'Seats Filled: 0', 'Windows Available: 0', 'Regular Meals: 0', 'Low Calorie Meals: 0', and 'Vegetarian Meals: 0'. Below these is a 'Refresh Statistics' button. In the center, an 'Assign' dialog box is open, featuring four radio buttons: 'Unoccupied', 'Regular', 'Low Calorie', and 'Vegetarian'. The 'Assign' dialog is positioned over a grid representing the airplane's seating chart. The grid has columns labeled A through F. At the bottom of the application, there is a row of seven input fields, with the first one containing the number '1'.

[illegible]

Airplane Seating Chart

Seats Filled: 24
Windows Available: 28
Regular Meals: 7
Low Calorie Meals: 10
Vegetarian Meals: 7

Refresh Statistics

A	B	C		D	E	F	
L	.	R		.	V	.	
V	.	L		.	L	R	
L	R	V		V	L	.	
.	
.	
.	
R	L	V		L	.	R	
.	
L	R	L		.	V	.	
.	
.	
.	
.	
.	
.	R	.		L	V	.	

15

.

R

.

L

V

.

Chapter 4 - Programming Project 2 – Gradebook

Code

Public Class GradebookForm

```
Private Sub Gradebook_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
    Me.StudentsTableAdapter.Fill(Me.GradebookDataSet.Students)
```

```
    Me.GradesTableAdapter.Fill(Me.GradebookDataSet.Grades)
```

```
End Sub
```

```
Private Sub UpdateBtn_Click(sender As Object, e As EventArgs) Handles UpdateBtn.Click
```

```
    GradebookSource.EndEdit()
```

```
    GradesTableAdapter.Update(GradebookDataSet.Grades)
```

```
End Sub
```

```
Private Sub DisplayBtn_Click(sender As Object, e As EventArgs) Handles DisplayBtn.Click
```

```
    Dim query = From grade In GradebookDataSet.Grades
```

```
        Join student In GradebookDataSet.Students On grade.studentID Equals student.studentID
```

```
    Let fullName = student.firstName + " " + student.lastName
```

```
    Let semGrade = (grade.firstExam + grade.secondExam + (2 * grade?.finalExam)) / 4
```

```
    Order By student.firstName
```

```
    Select fullName, semGrade
```

```
    DataGridView.DataSource = query.ToList
```

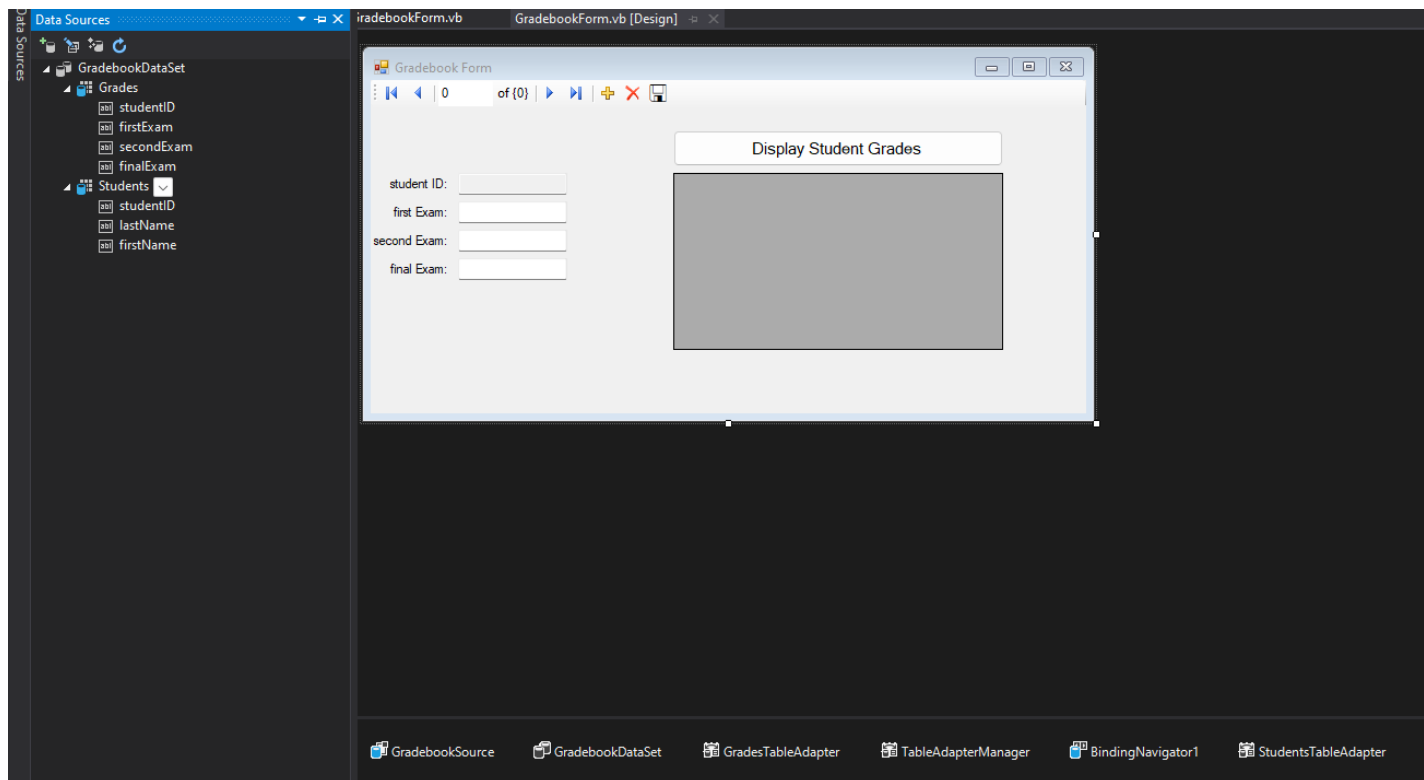
```
    DataGridView.CurrentCell = Nothing
```

```
End Sub
```

End Class

Screenshots

Data Source, Data Sets, Table Adapters and Binding Navigator



On Start Up with data filled

Gradebook Form

1 of 5

Display Student Grades

student ID: AJ-123456

first Exam: 50

second Exam: 60

final Exam: 70

Paged through, populated rest of student’s data, clicked “Display Student Grades” button

Gradebook Form

3 of 5

Display Student Grades

student ID: GG-567876

first Exam: 80

second Exam: 70

final Exam: 50

	fullName	semGrade
	Al Jones	62.5
	Fred Adams	50
	George Jackson	63.5
	Grace Green	62.5
	Lauren Smith	62

Chapter 5 - Programming Project 1 – Bank Account

Code

Public Class Accounts

```
Private checkAccount As New Account("Check")
Private savingsAccount As New Account("Savings")
Private currentAccountType As String = "Check"
Private currentTransferFromAccountType As String = "Check"
```

```
Private Sub Accounts_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    AccountTypeComboBox.SelectedIndex = 0
    TransferAccountTypeComboBox.SelectedIndex = 0
    DateTextBox.Text = DateTime.Now.ToString("yyyy-MM-dd")
    BalanceTextBox.Text = 0.ToString("C2")
    TransactionsListView.Columns.Add("Date", 80, HorizontalAlignment.Left)
    TransactionsListView.Columns.Add("Type of Transaction", 200, HorizontalAlignment.Left)
    TransactionsListView.Columns.Add("Amount", 80, HorizontalAlignment.Right)
    TransactionsListView.Columns.Add("New Balance", 80, HorizontalAlignment.Right)
    TransactionsListView.Columns(TransactionsListView.Columns.Count() - 1).Width = -2
End Sub
```

```
Private Sub TransferAccountTypeComboBox_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
TransferAccountTypeComboBox.SelectedIndexChanged
    currentTransferFromAccountType = TransferAccountTypeComboBox.Text
    If currentTransferFromAccountType = "Check" Then
        TransferLbl.Text = "To Savings:"
    End If
    If currentTransferFromAccountType = "Savings" Then
        TransferLbl.Text = "To Check:"
    End If
End Sub
```

End Sub

```
Private Sub TransferBtn_Click(sender As Object, e As EventArgs) Handles TransferBtn.Click
    If currentTransferFromAccountType = "Check" Then
        checkAccount.AddTransaction(New Transaction("Transfer To: Savings Account", CDec(TransferTextBox.Text), "Debit"))
        savingsAccount.AddTransaction(New Transaction("Transfer From: Check Account", CDec(TransferTextBox.Text), "Credit"))
    End If

    If currentTransferFromAccountType = "Savings" Then
        checkAccount.AddTransaction(New Transaction("Transfer From: Savings Account", CDec(TransferTextBox.Text), "Credit"))
        savingsAccount.AddTransaction(New Transaction("Transfer To: Check Account", CDec(TransferTextBox.Text), "Debit"))
    End If
    UpdateListView()
    DepositTextBox.Text = ""
End Sub
```

```
Private Sub AccountTypeComboBox_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
AccountTypeComboBox.SelectedIndexChanged
    currentAccountType = AccountTypeComboBox.Text
    If AccountTypeComboBox.SelectedIndex = 0 Then
        CheckGroupBox.Show()
    Else
        CheckGroupBox.Hide()
    End If
    UpdateListView()
End Sub
```

```
Private Sub DepositBtn_Click(sender As Object, e As EventArgs) Handles DepositBtn.Click
    If currentAccountType = "Check" Then
        checkAccount.AddTransaction(New Transaction("Deposit", CDec(DepositTextBox.Text), "Credit"))
    End If
    If currentAccountType = "Savings" Then
        savingsAccount.AddTransaction(New Transaction("Deposit", CDec(DepositTextBox.Text), "Credit"))
    End If
End Sub
```

```
End If
UpdateListView()
DepositTextBox.Text = ""
End Sub
```

```
Private Sub WithdrawalBtn_Click(sender As Object, e As EventArgs) Handles WithdrawalBtn.Click
    If currentAccountType = "Check" Then
        checkAccount.AddTransaction(New Transaction("Withdrawal", CDec(WithdrawalTextBox.Text), "Debit"))
    End If
    If currentAccountType = "Savings" Then
        savingsAccount.AddTransaction(New Transaction("Withdrawal", CDec(WithdrawalTextBox.Text), "Debit"))
    End If
    UpdateListView()
    WithdrawalTextBox.Text = ""
End Sub
```

```
Private Sub CheckBtn_Click(sender As Object, e As EventArgs) Handles CheckBtn.Click
    If currentAccountType = "Check" Then
        checkAccount.AddTransaction(New Transaction("Check cashed by: " + CheckRefTextBox.Text, CDec(CheckTextBox.Text), "Debit"))
    End If
    UpdateListView()
    CheckTextBox.Text = ""
    CheckRefTextBox.Text = ""
End Sub
```

```
Private Sub UpdateListView()
    TransactionsListView.Items.Clear()
    If currentAccountType = "Check" Then
        Dim currBalance = 0
        For Each transaction As Transaction In checkAccount.Transactions
            Dim arr As String() = New String(4) {}
            currBalance += If(transaction.TransactionType = "Credit", transaction.Amount, transaction.Amount * -1)
            arr(0) = transaction.TransactionDate.ToString("yyyy-MM-dd")
            arr(1) = transaction.TransactionName
            arr(2) = transaction.Amount.ToString("C2")
            arr(3) = currBalance.ToString("C2")
            Dim item As New ListViewItem(arr)
            TransactionsListView.Items.Add(item)
            My.Computer.FileSystem.WriteAllText($"{My.Computer.FileSystem.CurrentDirectory}\CheckAccount.txt", String.Join(",", arr), True)
        Next
        BalanceTextBox.Text = currBalance.ToString("C2")
    End If

    If currentAccountType = "Savings" Then
        Dim currBalance = 0
        For Each transaction As Transaction In savingsAccount.Transactions
            Dim arr As String() = New String(4) {}
            currBalance += If(transaction.TransactionType = "Credit", transaction.Amount, transaction.Amount * -1)
            arr(0) = transaction.TransactionDate.ToString("yyyy-MM-dd")
            arr(1) = transaction.TransactionName
            arr(2) = transaction.Amount.ToString("C2")
            arr(3) = currBalance.ToString("C2")
            Dim item As New ListViewItem(arr)
            TransactionsListView.Items.Add(item)
            My.Computer.FileSystem.WriteAllText($"{My.Computer.FileSystem.CurrentDirectory}\SavingsAccount.txt", String.Join(",", arr), True)
        Next
        BalanceTextBox.Text = currBalance.ToString("C2")
    End If
End Sub
End Sub
End Class
```

Public Class Transaction

Public TransactionName As String

Public Amount As Decimal

Public TransactionDate As DateTime

Public TransactionType As String

Public Sub New(tName As String, amnt As Decimal, tType As String)

TransactionName = tName

Amount = amnt

TransactionType = tType

TransactionDate = DateTime.Now

End Sub

End Class

Public Class Account

Public Transactions As List(Of Transaction)

Public AccountType As String

Public Function Balance()

Return Transactions.Sum(Function(tr) If(tr.TransactionType = "Credit", tr.Amount, tr.Amount * -1))

End Function

Public Sub New(accountType As String)

accountType = accountType

Transactions = New List(Of Transaction)

End Sub

Public Function AddTransaction(tr As Transaction)

If tr.TransactionType = "Debit" Then

If Balance() - tr.Amount < 0 Then

MsgBox("Insufficient Funds")

Exit Function

End If

End If

Transactions.Add(tr)

End Function

End Class

On Start Up

Changing Transfer From Account Check Box (Updating To {AccountType} label)

[illegible]

[illegible][illegible]

Performing a Check payment

Accounts

Account: Check Balance: \$99,697,500.00 Today's Date: 2022-08-23

Deposit

\$

Deposit

Withdrawal

\$

Withdrawal

Transfer

Transfer From: Savings

To Check: \$

Transfer

Check

Pay This Amount: \$

To The Order Of:

Sign Check

Date	Type of Transaction	Amount	New Balance
2022-08-23	Deposit	\$100,000,000.00	\$100,000,000.00
2022-08-23	Withdrawal	\$2,500.00	\$99,997,500.00
2022-08-23	Check cashed by: WnB Transporters	\$300,000.00	\$99,697,500.00

Transferring Money from Check to Savings

Accounts

Account: Check Balance: \$92,697,500.00 Today's Date: 2022-08-23

Deposit

\$

Deposit

Withdrawal

\$

Withdrawal

Transfer

Transfer From: Check

To Savings: \$

Transfer

Check

Pay This Amount: \$

To The Order Of:

Sign Check

Date	Type of Transaction	Amount	New Balance
2022-08-23	Deposit	\$100,000,000.00	\$100,000,000.00
2022-08-23	Withdrawal	\$2,500.00	\$99,997,500.00
2022-08-23	Check cashed by: WnB Transporters	\$300,000.00	\$99,697,500.00
2022-08-23	Transfer To: Savings Account	\$7,000,000.00	\$92,697,500.00

Performing Insufficient Transaction

[illegible]

Switching to Savings (Hiding check payment box with transfer from Check Account)

[illegible]

[illegible][illegible]

Transfer from Savings to Check Account

Accounts

Account

Savings

Balance

\$6,630,000.00

Today's Date

2022-08-23

Deposit

\$

Deposit

Withdrawal

\$

Withdrawal

Transfer

Transfer From:

Savings

To Check:

\$

70000

Transfer

Date	Type of Transaction	Amount	New Balance
2022-08-23	Transfer From: Check Account	\$7,000,000.00	\$7,000,000.00
2022-08-23	Deposit	\$300,000.00	\$7,300,000.00
2022-08-23	Withdrawal	\$600,000.00	\$6,700,000.00
2022-08-23	Transfer To: Check Account	\$70,000.00	\$6,630,000.00

CheckAccount.txt

CheckAccount.txt

```
1 2022-08-23,Deposit,$100,000,000.00,$100,000,000.00
2 2022-08-23,Withdrawal,$2,500.00,$99,997,500.00
3 2022-08-23,Check cashed by: WnB Transporters,$300,000.00,$99,697,500.00
4 2022-08-23,Transfer To: Savings Account,$7,000,000.00,$92,697,500.00
```

SavingsAccount.txt

CheckAccount.txt SavingsAccount.txt

```
1 2022-08-23,Transfer From: Check Account,$7,000,000.00,$7,000,000.00
2 2022-08-23,Deposit,$300,000.00,$7,300,000.00,2022-08-23
3 Withdrawal,$600,000.00,$6,700,000.00,2022-08-23
4 Transfer To: Check Account,$70,000.00,$6,630,000.00
```