

# FINAL PROJECTS

## AirPlane and MongoDB Applications

Mansoura University

Faculty Of Engineering

### 3<sup>rd</sup> Year CSED

Abdulrahman Issam AbouOuf

Data Structure CSE 3312

# Table of Contents

<b>TABLE OF CONTENTS.....</b>	<b>0</b>
<b>2 PROJECT ONE AIRPLANE .....</b>	<b>2</b>
2.1 RANDOM FILE .....	3
2.2 FORM .....	3
2.3 CODE EXPLANATION .....	4
2.3.1 FLIGHT STRUCTURE .....	4
2.3.2 FORM LOAD .....	4
2.3.3 ADD A FLIGHT .....	5
2.3.4 BOOKING A SEAT .....	6
2.3.5 BOOKING CONFIRMATION.....	7
2.3.6 CANCELING A BOOKING .....	8
2.3.7 VIEWS AND BUTTON HANDLINGS .....	9
2.3.7.1 View() .....	9
2.3.7.2 Button Mapping Foreach Seat .....	9
<b>3 PROJECT TWO MONGODB STORE.....</b>	<b>10</b>
3.1 WHAT IS MONGO DATABASE? .....	11
3.2 FORM .....	11
3.2.1 CRUD .....	11
3.3 CODE EXPLANATION .....	12
3.3.1 FORM LOAD .....	12
3.3.2 CRUD OPERATIONS.....	12
3.3.2.1 Add a model .....	12
3.3.2.2 Edit a model .....	13
3.3.2.3 Show data (read).....	13
3.3.2.4 Delete data.....	14

## **2 Project One Airplane**

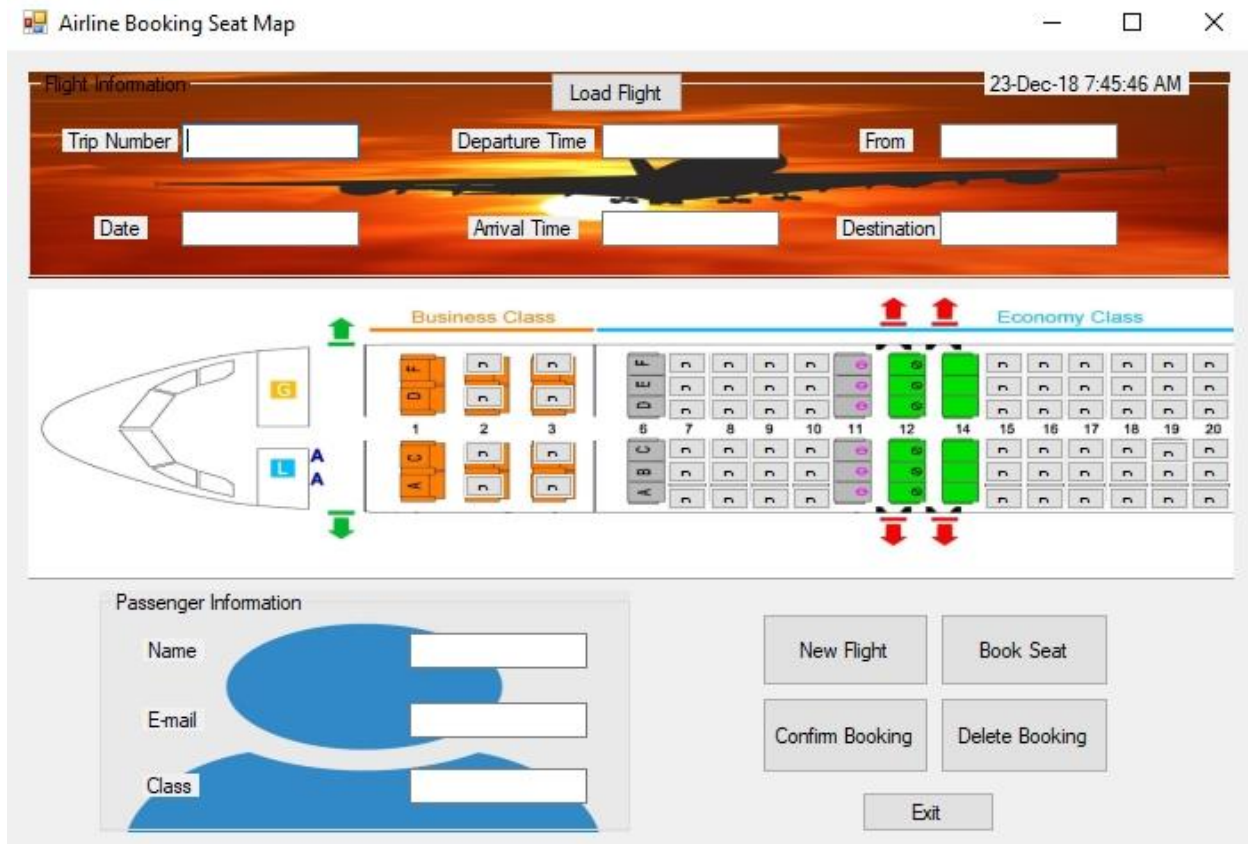
## 2.1 Random File

Random access files permit nonsequential, or random, access to a file's contents. To access a file randomly, you open the file, seek a particular location, and read from or write to that file.

## 2.2 Form

The form takes first a new trip then adds it to the random file, opening seats for the flight, and each passenger information.

The seats are free for booking (green), Needs Confirmation (orange) or Confirmed (Red).



The screenshot shows a Windows application titled "Airline Booking Seat Map". The interface is divided into several sections:

- Flight Information:** A header section with a "Load Flight" button and a timestamp "23-Dec-18 7:45:46 AM". Below this are input fields for "Trip Number", "Departure Time", "From", "Date", "Arrival Time", and "Destination".
- Seat Map:** A central area showing a top-down view of an airplane cabin. It is divided into "Business Class" (left) and "Economy Class" (right). The Business Class section has 3 rows (1-3) and 3 columns (A-C). The Economy Class section has 4 rows (4-7) and 17 columns (D-S). Seats are represented by buttons with different colors: green for "free for booking", orange for "Needs Confirmation", and red for "Confirmed".
- Passenger Information:** A section with a blue silhouette of a person and input fields for "Name", "E-mail", and "Class".
- Buttons:** A set of buttons at the bottom right: "New Flight", "Book Seat", "Confirm Booking", "Delete Booking", and "Exit".

Figure 2-1 – Airplane Project Form

With each seat represented by a button that changes its interface depending on the seat state. You can book seats easily.

## 2.3 Code Explanation

### 2.3.1 Flight Structure

This Structure defines the information about the flight(Arrival, Destination, Date, etc..) and each passenger of the flight in the vector.

So You can create flights using this structure.

```
Public Structure Booking
    Dim TripNo As Integer
    Dim TripDate As String
    Dim DepTime As String
    Dim ArriveTime As String
    Dim From As String
    Dim Desti As String

    <VBFixedArray(68)> Dim Seat() As Integer
    <VBFixedArray(68)> Dim name() As String
    <VBFixedArray(68)> Dim mail() As String
    <VBFixedArray(68)> Dim flightclass() As String
End Structure
```

### 2.3.2 Form Load

Loads the flight and initializes the flight information

```
Private Sub Form1_Load(sender As Object, e As EventArgs)
    Handles MyBase.Load
        Label10.Text = DateTime.Now.ToString()
        FileOpen(1, "flightdb.txt", OpenMode.Random,
OpenAccess.ReadWrite, OpenShare.Shared, Len(FlightBook))
        FlightBook.Seat = New Integer(68) {}
        FlightBook.name = New String(68) {}
        FlightBook.mail = New String(68) {}
        FlightBook.flightclass = New String(68) {}

        For i As Integer = 1 To 68
            Buttons(i - 1) = Me.Controls("Button" & i)
            AddHandler Buttons(i - 1).Click, AddressOf SeatNo
        Next
    End Sub
```

### 2.3.3 Add a Flight

This takes the info from the textboxes and save it in the random file. It initializes the seat states too.

```
Private Sub btn_newflight_Click(sender As Object, e As EventArgs) Handles
btn_newflight.Click
    If txt_no.Text = "" Or txt_arriv.Text = "" Or txt_date.Text = "" Or
txt_dest.Text = "" Or txt_from.Text = "" Or txt_depa.Text = "" Then
        MsgBox("Missing data about the Flight Info")
        Exit Sub
    End If
    position = Loc(1) 'Specifies the current pos in a file
    Seek(1, Val(txt_no.Text)) 'take the file number and the position

    FlightBook.TripNo = Val(txt_no.Text)
    FlightBook.TripDate = txt_date.Text
    FlightBook.ArriveTime = txt_arriv.Text
    FlightBook.DepTime = txt_depa.Text
    FlightBook.From = txt_from.Text
    FlightBook.Desti = txt_dest.Text

    For i As Integer = 0 To 67 '68 seats in the flightbook array from 0 to
67 of the passengers
        FlightBook.Seat(i) = 0
        FlightBook.name(i) = "0"
        FlightBook.mail(i) = "0"
        FlightBook.flightclass(i) = "0"
    Next

    Dim cur As Integer = Val(txt_no.Text)
    FilePut(1, FlightBook, cur)
    ClearFlightInfo()
End Sub
```

### 2.3.4 Booking a Seat

Checks if the data is filled, then asks for which seat do you want and fill the passenger info. Then adding all this information to the file

```
Private Sub btn_book_Click(sender As Object, e As EventArgs) Handles btn_book.Click
    If txt_name.Text = "" Or txt_mail.Text = "" Or txt_class.Text = "" Then
        MsgBox("Please Fill your data and choose a seat")
        Exit Sub
    End If
    View()
    Dim result As DialogResult
    result = MessageBox.Show("Do you want to book this seat?", "Seat Booking",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Question)

    If result = DialogResult.OK Then
        Buttons(CurrSeat).BackColor = Color.Orange
        SeatFlag = 1
        FileGet(1, FlightBook, Val(txt_no.Text))
        FlightBook.Seat(CurrSeat) = SeatFlag
        FlightBook.name(CurrSeat) = txt_name.Text
        FlightBook.mail(CurrSeat) = txt_mail.Text
        FlightBook.flightclass(CurrSeat) = txt_class.Text

        Dim cur As Integer = Val(txt_no.Text)
        FilePut(1, FlightBook, cur) 'add new passenger and update
    ElseIf result = DialogResult.Cancel Then
        Buttons(CurrSeat).BackColor = Color.Lime
        Exit Sub
    End If
    ClearPassengerInfo()
End Sub
```

### 2.3.5 Booking Confirmation

This makes sure if you want to confirm your seat based on your information. Asks if you want to Confirm or to Cancel. Then updates the states to the random file

```
Private Sub btn_confirm_Click(sender As Object, e As EventArgs) Handles
btn_confirm.Click
    View()

    If SeatFlag = 0 Then
        MessageBox.Show("This seat is not booked", "Booking Confirm",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
    ElseIf SeatFlag = 1 Then
        Dim result As DialogResult
        result = MessageBox.Show("Confirm booking this seat?", "Booking
        Confirm", MessageBoxButtons.OKCancel, MessageBoxIcon.Question)

        If result = DialogResult.OK Then
            Buttons(CurrSeat).BackColor = Color.Red
            SeatFlag = 2
            FileGet(1, FlightBook, Val(txt_no.Text))

            FlightBook.Seat(CurrSeat) = SeatFlag
            Dim cur As Integer = Val(txt_no.Text)
            FilePut(1, FlightBook, cur) 'update after confirming
        ElseIf result = DialogResult.Cancel Then
            Buttons(CurrSeat).BackColor = Color.Orange
            Exit Sub
        End If
    End If
    ClearPassengerInfo()
End Sub
```



### 2.3.6 Canceling a Booking

Cancel the booking if you didn't confirm. If you confirmed, you can't cancel and then updates the random file

```
Private Sub btn_delete_Click(sender As Object, e As EventArgs) Handles
btn_delete.Click
    View()
    If SeatFlag = 1 Then
        Dim result As DialogResult
        result = MessageBox.Show("Booking is not confirmed yet, Do you want to
cancel?", "Cancel Booking", MessageBoxButtons.OKCancel, MessageBoxIcon.Question)

        If result = DialogResult.OK Then
            Buttons(CurrSeat).BackColor = Color.Lime
            SeatFlag = 0

            FlightBook.Seat(CurrSeat) = 0
            FlightBook.name(CurrSeat) = "0"
            FlightBook.mail(CurrSeat) = "0"
            FlightBook.flightclass(CurrSeat) = "0"
        ElseIf result = DialogResult.Cancel Then
            Buttons(CurrSeat).BackColor = Color.Orange
            Exit Sub
        End If
    ElseIf SeatFlag = 2 Then
        MessageBox.Show("You already Confirmed booking this seat, You can't
cancel your booking", "Error Canceling Booking", MessageBoxButtons.OK,
MessageBoxIcon.Error)
    End If
    ClearPassengerInfo()
End Sub
```

## 2.3.7 Views and button handlings

### 2.3.7.1 View()

This method is used to show the states of each flight, taking the flight number and it will show you the Airplane seat map and the passenger information for each seat.

```
Public Sub View()  
    FileGet(1, FlightBook, Val(txt_no.Text))  
  
    If FlightBook.TripNo <> Val(txt_no.Text) Then  
        MessageBox.Show("We don't have this Flight", "Error", MessageBoxButtons.OK,  
        MessageBoxIcon.Error)  
        FileClose(1)  
        Exit Sub  
    End If  
    txt_date.Text = FlightBook.TripDate  
    txt_from.Text = FlightBook.From  
    txt_dest.Text = FlightBook.Desti  
    txt_arriv.Text = FlightBook.ArriveTime  
    txt_depa.Text = FlightBook.DepTime  
  
    For i As Integer = 0 To 67  
        If FlightBook.Seat(i) = 1 Then  
            Buttons(i).BackColor = Color.Orange  
        ElseIf FlightBook.Seat(i) = 2 Then  
            Buttons(i).BackColor = Color.Red  
        Else  
            Buttons(i).BackColor = Color.Lime  
        End If  
    Next  
End Sub
```

### 2.3.7.2 Button Mapping Foreach Seat

To know which seat is booked for new seat or confirming it.

```
Public Sub SeatNo(sender As Object, e As EventArgs)  
    FileGet(1, FlightBook, Val(txt_no.Text))  
    For i As Integer = 0 To Buttons.Length - 1  
        If Buttons(i) Is sender Then  
            CurrSeat = i  
            Exit For  
        End If  
    Next  
  
    SeatFlag = FlightBook.Seat(CurrSeat)  
    If SeatFlag <> 0 Then  
        txt_name.Text = FlightBook.name(CurrSeat)  
        txt_mail.Text = FlightBook.mail(CurrSeat)  
        txt_class.Text = FlightBook.flightclass(CurrSeat)  
    End If  
End Sub
```

## **3 Project Two MongoDB Store**

### 3.1 What is Mongo DataBase?



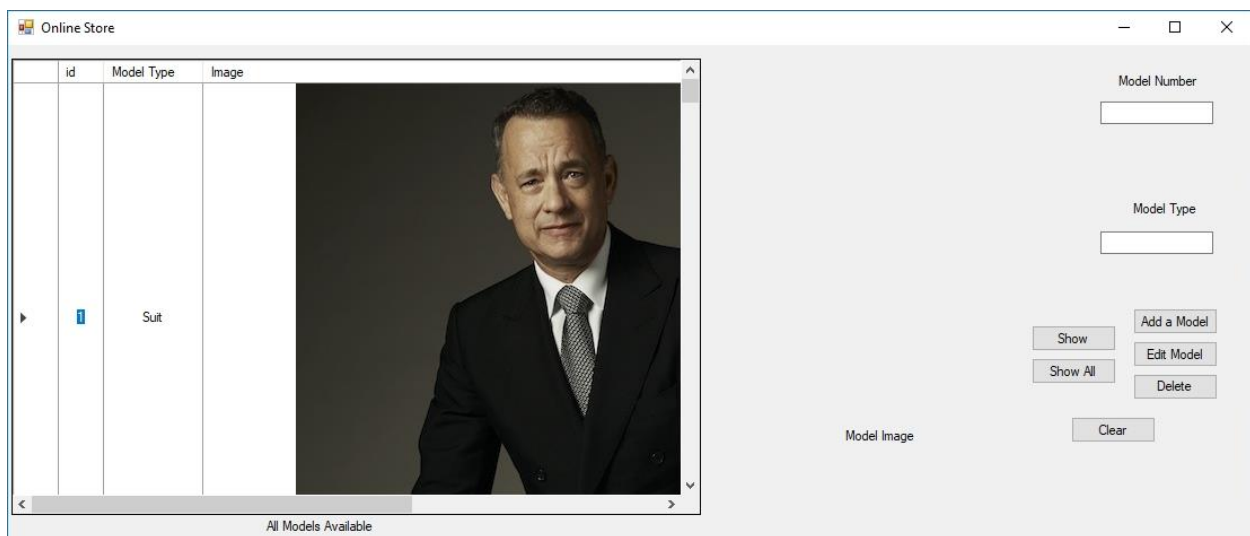
MongoDB is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemata.


### 3.2 Form

The form represents an Online Clothes Store with variety of clothes for both genders. Also, adding new models, Editing and deleting. Using **CRUD**.

#### 3.2.1 CRUD

In computer programming, create, read, update, and delete are the four basic functions of persistent storage.



id	Model Type	Image
1	Suit	

Model Number:

Model Type:

Buttons: Show, Show All, Add a Model, Edit Model, Delete, Clear

Figure 3-1 - MongoDB Store Project Form

## 3.3 Code Explanation

### 3.3.1 Form Load

This Loads the database, Sets the connection and structuring the datatable.

```
Dim client As MongoClient
Dim db As IMongoDatabase
Dim collection As IMongoCollection(Of BsonDocument)
Dim dt As New DataTable
Dim datasourceflag As Boolean

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    client = New MongoClient("mongodb://localhost:27017/")
    db = client.GetDatabase("db")
    collection = db.GetCollection(Of BsonDocument)("Models")
    dt.Columns.Add("id", Type.GetType("System.Int32"))
    dt.Columns.Add("Model Type", Type.GetType("System.String"))
    dt.Columns.Add("Image", Type.GetType("System.Byte[]"))
End Sub
```

### 3.3.2 CRUD Operations

#### 3.3.2.1 Add a model

Adds a model to the database. And specifying the image types

```
Private Sub btn_add_Click(sender As Object, e As EventArgs) Handles btn_add.Click
    OpenFileDialog1.Filter = "Image Files |*.bmp;*.gif;*.jpg;*.png;*.tif|Allfiles|*.*"
    If OpenFileDialog1.ShowDialog() = DialogResult.OK Then
        PictureBox1.BackgroundImage = Image.FromFile(OpenFileDialog1.FileName, True)
    End If

    Dim f = Builders(Of BsonDocument).Filter.Empty
    For Each cmp As BsonDocument In collection.Find(f).ToList
        Dim id As BsonElement = cmp.GetElement("id")
        If txt_mno.Text = id.Value Then
            MsgBox("This id is already taken")
            Exit Sub
        End If
    Next
    Dim memStr As MemoryStream = New MemoryStream()
    PictureBox1.BackgroundImage.Save(memStr, ImageFormat.Png)
    Dim byteArray As Byte() = memStr.ToArray()
    Dim doc As BsonDocument = New BsonDocument()
    With doc
        .Add("id", txt_mno.Text)
        .Add("model", txt_mtype.Text)
        .Add("image", byteArray)
    End With
    collection.InsertOne(doc)
```

### 3.3.2.2 Edit a model

```
Private Sub btn_edit_Click(sender As Object, e As EventArgs) Handles btn_edit.Click
    OpenFileDialog1.Filter = "Image Files
|*.bmp;*.gif;*.jpg;*.png;*.tif|Allfiles|*.*"
    If OpenFileDialog1.ShowDialog() = DialogResult.OK Then
        PictureBox1.BackgroundImage = Image.FromFile(OpenFileDialog1.FileName, True)
    End If

    Dim f = Builders(Of BsonDocument).Filter.Empty
    For Each cmp As BsonDocument In collection.Find(f).ToList
        Dim id As BsonElement = cmp.GetElement("id")
        If txt_mno.Text = id.Value Then
            Dim memStr As MemoryStream = New MemoryStream()
            PictureBox1.BackgroundImage.Save(memStr, ImageFormat.Png)
            Dim byteArray As Byte() = memStr.ToArray()
            'Dim doc As BsonDocument = New BsonDocument()
            Dim doc As BsonDocument = New BsonDocument()
            With doc
                .Set("id", txt_mno.Text)
                .Set("model", txt_mtype.Text)
                .Set("image", byteArray)
            End With
            collection.InsertOne(doc)
        End If
    Next

End Sub
```

### 3.3.2.3 Show data (read)

Using two buttons, Show: to show a specific model, and Show All to show the models in a datagridview based on the database.

#### 3.3.2.3.1 Show Button

```
Private Sub btn_show_Click(sender As Object, e As EventArgs) Handles
btn_show.Click
    Dim f = Builders(Of BsonDocument).Filter.Empty
    For Each cmp As BsonDocument In collection.Find(f).ToList
        Dim id As BsonElement = cmp.GetElement("id")
        Dim model As BsonElement = cmp.GetElement("model")
        Dim image As BsonElement = cmp.GetElement("image")
        Dim img As Byte() = image.Value
        Dim memStr As MemoryStream = New MemoryStream(img)
        If txt_mno.Text = id.Value Then
            txt_mtype.Text = model.Value
            PictureBox1.BackgroundImage = FromStream(memStr)
            Exit Sub
        End If
    Next
    MsgBox("No models found")
    txt_mno.Clear()
    txt_mtype.Clear()
    PictureBox1.BackgroundImage = Nothing
End Sub
```

### 3.3.2.3.2 Show All Button

```
Private Sub btn_showall_Click(sender As Object, e As EventArgs) Handles btn_showall.Click
    If datasourceflag = True Then DataGridView1.DataSource.clear()
    Dim f = Builders(Of BsonDocument).Filter.Empty
    For Each cmp As BsonDocument In collection.Find(f).ToList
        Dim id As BsonElement = cmp.GetElement("id")
        Dim model As BsonElement = cmp.GetElement("model")
        Dim image As BsonElement = cmp.GetElement("image")
        Dim img As Byte() = image.Value
        Dim memStr As MemoryStream = New MemoryStream(img)
        dt.Rows.Add(id.Value, model.Value, memStr.ToArray())
    Next
    DataGridView1.DataSource = dt
    datasourceflag = True
End Sub
```

### 3.3.2.4 Delete data

```
Private Sub btn_delete_Click(sender As Object, e As EventArgs) Handles btn_delete.Click
    Dim f = Builders(Of BsonDocument).Filter.Empty
    For idx = 0 To dt.Rows.Count - 1
        Dim cmp As BsonDocument = (collection.Find(f).ToList)(idx)
        Dim id As BsonElement = cmp.GetElement("id")
        Dim model As BsonElement = cmp.GetElement("model")
        Dim image As BsonElement = cmp.GetElement("image")
        Dim img As Byte() = image.Value
        Dim memStr As MemoryStream = New MemoryStream(img)
        If txt_mno.Text = id.Value Then
            Dim val = Builders(Of BsonDocument).Filter.Eq(Of String)("id", txt_mno.Text)
            collection.DeleteOne(val)
            Exit For
        End If
    Next
    txt_mno.Clear()
    txt_mtype.Clear()
    PictureBox1.BackgroundImage = Nothing
    DataGridView1.DataSource = dt
    datasourceflag = True

End Sub
```