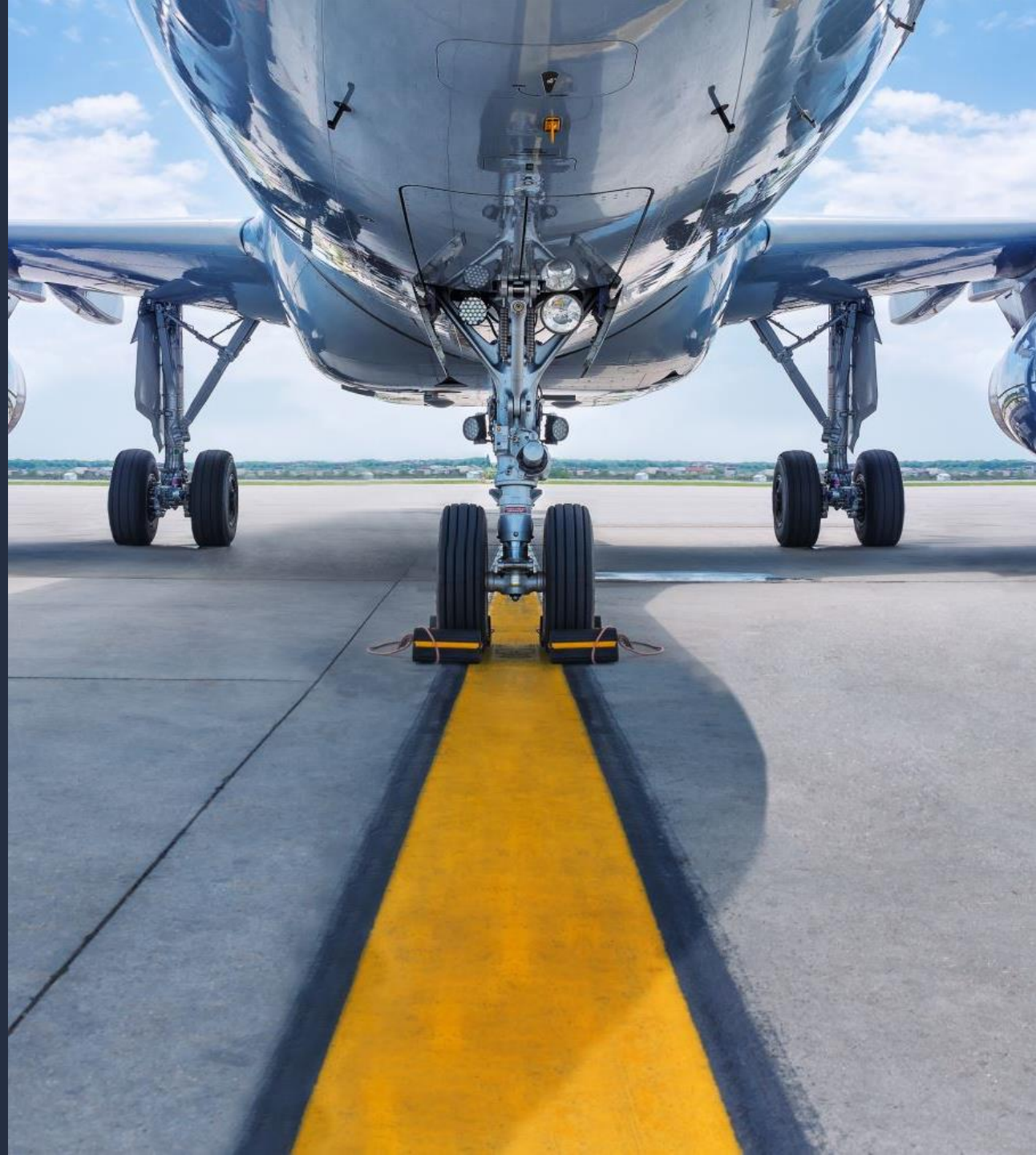# ConTrails

Bahareh Jabarzadnouri,

Aly Westwood,

Kim Alizadeh

# High Level View – ConTrails Application

Flight tracking

File saving

Maintenance tracking

Security

Solution Overview

Menu & Menu Items

# Solution Overview

## View All Service & Printing Report as PDF

# Solution Overview – Tabs

# Solution Overview

## Flights

# Database

# Challenges & Solutions

**SQL Server/Azure**

-Learning curve

-Different datatypes (e.g. used int instead of enum)

-Azure account ran out of credits

-Changing IPs

-Update database structure

## Upgrade now to keep building in Azure

Your free credit has expired, so we've disabled your subscription and services. It's an easy fix—just upgrade to a pay-as-you-go subscription to keep using all your configured services and data in place. You get:

| Rule name | Start IP | End IP | |
|---|---|---|---|
| | | | ... |
| All IPs | 0.0.0.0 | 255.255.255.255 | ... |

# Challenges & Solutions

**ListView Sorting**

| Registration | Model | Next Service Due Date | Last Service Date | Tail # ▲ | Location |
|---|---|---|---|---|---|
| C-FBUY | Embraer175 | 2022-02-18 14:46:03Z | 2022-02-16 14:46:03Z | 72 | C-YUL (Montreal) |
| C-GGUY | Boeing737 | 2022-02-18 17:42:48Z | 2022-02-16 17:42:48Z | 73 | C-YUL (Montreal) |
| C-FBJR | AirbusA220 | 2022-02-18 14:59:34Z | 2022-02-16 14:59:34Z | 74 | C-YUL (Montreal) |
| C-GYUS | AirbusA220 | 2022-02-18 15:01:57Z | 2022-02-16 15:01:57Z | 75 | C-YUL (Montreal) |

```xml
<GridViewColumn DisplayMemberBinding="{Binding MaintPerformedDate, StringFormat='u'}" Width="160">
    <GridViewColumn.Header>
        <GridViewColumnHeader Tag="MaintPerformedDate" Click="lvAircraftColumnHeader_Click">Last Service Date</GridViewColumnHeader>
    </GridViewColumn.Header>
</GridViewColumn>
<GridViewColumn DisplayMemberBinding="{Binding ID}" Width="80">
    <GridViewColumn.Header>
        <GridViewColumnHeader Tag="ID" Click="lvAircraftColumnHeader_Click">Tail #</GridViewColumnHeader>
    </GridViewColumn.Header>
</GridViewColumn>
```

# Challenges & Solutions

## ListView Sorting

```
6 references
private void lvAircraftColumnHeader_Click(object sender, RoutedEventArgs e)
{
    GridViewColumnHeader column = (sender as GridViewColumnHeader);
    string sortBy = column.Tag.ToString();
```

```
3 references
public class SortAdorner : Adorner
{
    private static Geometry ascGeometry =
        Geometry.Parse("M 0 4 L 3.5 0 L 7 4 Z");

    private static Geometry descGeometry =
        Geometry.Parse("M 0 0 L 3.5 4 L 7 0 Z");
```

Tail #  ▲

Tail #  ▼
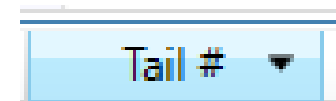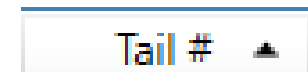
# Challenges & Solutions

## Installer

# Challenges & Solutions

## Status bar

```
9 references
public void ShowStatusMessage(string message)
{
    tbStatus.Text = message;
    var timer = new System.Timers.Timer();
    timer.Interval = 20000; //20 seconds
    timer.Elapsed += delegate (object sender, System.Timers.ElapsedEventArgs e)
    {
        //stop the timer
        timer.Stop();
        //remove the StatusMessage text using a dispatcher, because timer operates in another thread
        this.Dispatcher.BeginInvoke(new Action(() =>
        {
            tbStatus.Text = "";
        }));
    };
    timer.Start();
}
```

```
dbContext.SaveChanges();
mainWindow.ShowStatusMessage("Aircraft Updated");
```

Aircraft Updated

answered Jan 17 2013 at 18:44

Wiley Marques
465 ● 3 ● 15

stack**overflow**
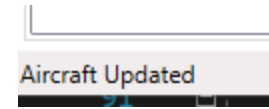
# Challenges & Solutions

Hashing Users Password:

Step 1: Generate Salt:

Step 2: Generate Hash:

```csharp
public static string CreateSalt(int size)
{
    //Generate a cryptographic random number.
    RNGCryptoServiceProvider rng = new RNGCryptoServiceProvider();
    byte[] buff = new byte[size];
    rng.GetBytes(buff);
    return Convert.ToBase64String(buff);
}

public static string GenerateHash(string input, string salt)
{
    byte[] bytes = Encoding.UTF8.GetBytes(input + salt);
    SHA256Managed sHA256ManagedString = new SHA256Managed();
    byte[] hash = sHA256ManagedString.ComputeHash(bytes);
    return Convert.ToBase64String(hash);
}
```

# Challenges & Solutions

**Hashing Users Password:**

Step 3: Generate Hash:

```
public static bool AreEqual(string plainTextInput, string hashedInput, string salt)
{
    string newHashedPin = GenerateHash(plainTextInput, salt);
    return newHashedPin.Equals(hashedInput);
}
```

# Challenges & Solutions

**Flight consistency**

Solution: make a routes table



Create a new flight leg for aircraft# 73; Registration C-GGUY

| C-YUL to C-YYZ | 2022-02-17 | 10 : 0 Z | Add Flight |

| Flight No | Aircraft | Dep. Date | Time | Dep. City | Dep. Code | Arr. Date | Time | Arr. City | Arr. Code | Status |
|---|---|---|---|---|---|---|---|---|
| CT0C02D | C-GGUY | 2022-02-17 10:00:00Z | Montreal | C-YUL | 2022-02-17 11:00:00Z | Toronto | C-YYZ | Upcoming |
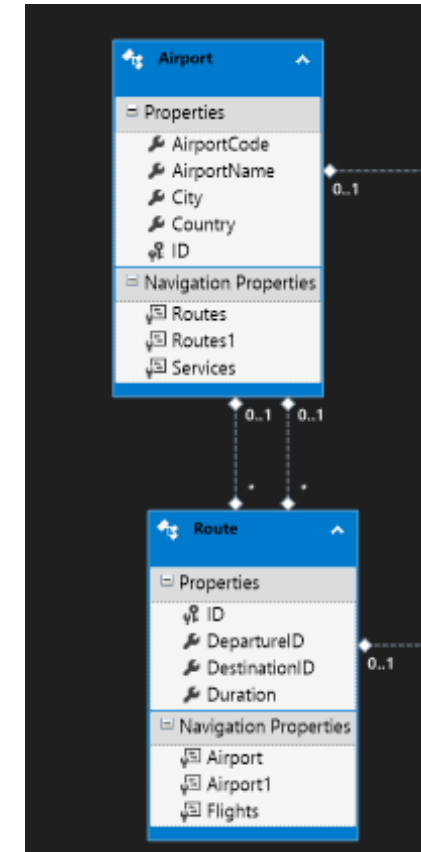
# Challenges & Solutions

✈️ Must land before MaintDueDate

✈️ Must be in the future

✈️ No overlap

✈️ No teleporting - flight path consideration

| Due for Maintenance | Unscheduled Aircraft | All Aircraft | All Flights | All Services |
|---|---|---|---|---|

| Flight No | Aircraft | Dep. Date \| Time | Dep. City | Dep. Code | Arr. Date \| Time |
|---|---|---|---|---|---|
| CT0C039 | C-FDIT | 2022-02-17 09:30:00Z | Montreal | C-YUL | 2022-02-17 10:30:00Z |
| CT0D03A | C-FDIT | 202 | | | |
| CT0E03B | C-FDIT | 202 | | | |
| CT0F03C | C-FDIT | 202 | | | |

**Remove 4 Flights** ✕

❓ This will also remove every flight following the selected flight for this aircraft. Are you sure you want to remove this flight?

---

🖼 Schedule flights for C-FDIT　　　　　　— ☐ ✕

### Create a new flight leg for aircraft# 100; Registration C-FDIT

Departing C-YUL at 2022-02-17 09:30:00Z; Arriving in C-YYZ at 2022-02-17 10:30:00Z
Departing C-YYZ at 2022-02-18 12:30:00Z; Arriving in C-YUL at 2022-02-18 13:30:00Z
Departing C-YUL at 2022-02-18 15:30:00Z; Arriving in US-JFK at 2022-02-18 17:30:00Z
Departing US-JFK at 2022-02-18 19:30:00Z; Arriving in C-YUL at 2022-02-18 21:30:00Z

| C-YUL  to  US-JFK ▾ | 2022-02-18 📅 | 23 ▾ | : | 30 ▾ | Z | **Add Flight** |

**Error** ✕

❌ The flight cannot be scheduled after the maintenance due date.
Maintenance is due 2022-02-19 00:00:00Z.
The flight you selected is 2022-02-18 23:30:00Z to 2022-02-19 01:30:00Z

[ OK ]

```
public static List<Aircraft> FindUnscheduledAc()
{

    return (from a in dbContext.Aircraft
            where a.MaintDueDate >= DateTime.UtcNow
            && a.OutOfService == false
            && !(from f in dbContext.Flights where f.ArrivalTime > DateTime.UtcNow select f.AircraftID).Contains(a.ID)
            select a).ToList().Where(a => a.MaintDueDate > DateTime.UtcNow.AddHours(shortestFlightOut(a) + 1)).ToList();


}


/// <summary>
/// Finds the length of the shortest possible flight out of the airport where the provided aircraft is located
/// </summary>
/// <param name="ac"></param>
/// <returns></returns>
2 references
private static int shortestFlightOut(Aircraft ac)
{
    return (int)Route.FindRoutesOut(LocateAc(ac)).Min(r => r.Duration);
}
```

Sche

**Schedule flights for C-FDIT**

Create a new flight leg for aircraft# 100; Registration C-FDIT

Departing C-YUL at 2022-02-17 10:30:00Z; Arriving in C-YYZ at 2022-02-17 11:30:00Z
Departing C-YYZ at 2022-02-18 14:30:00Z; Arriving in C-YUL at 2022-02-18 15:30:00Z
Departing C-YUL at 2022-02-18 17:30:00Z; Arriving in US-JFK at 2022-02-18 19:30:00Z
Departing US-JFK at 2022-02-18 21:30:00Z; Arriving in C-YUL at 2022-02-18 23:30:00Z

[ ⌄ ]  [ 2022-02-18  📅 ]  [ 21 ⌄ ] : [ 30 ⌄ ] Z   **Add Flight**

**Remove 2 Flights**                                          ✕

❓  This will also remove every flight following the selected flight.
    Are you sure you want to remove this flight?

                                      [ OK ]    [ Cancel ]

                        **Confirm Leg**
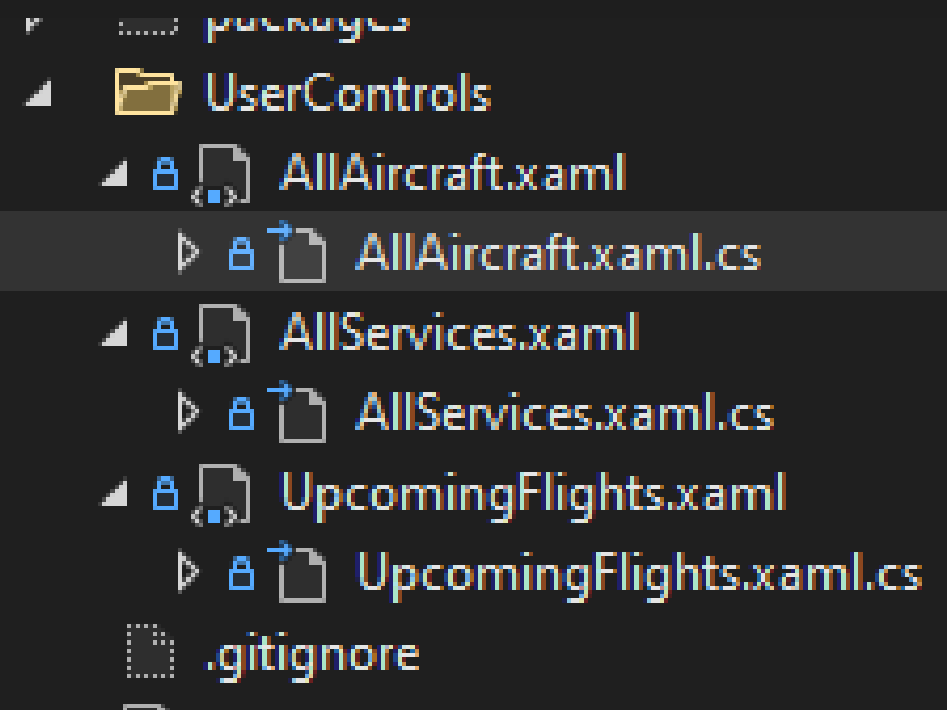
# What We Learned – IValueConverter

```csharp
public class IntModelConverter : IValueConverter
{
    0 references
    public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
    {
        return (Aircraft.ModelEnum)value;
    }

    0 references
    public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
    {
        return (int)value;
    }
}
```

```xml
<UserControl.Resources>
    <local:IntModelConverter x:Key="IntModelConverter" />
    <local:AircraftLocationStringConverter x:Key="AircraftLocationStringConverter" />
</UserControl.Resources>
<Grid>
<GridViewColumn DisplayMemberBinding="{Binding Model, Converter={StaticResource IntModelConverter}}" Width="80">
<GridViewColumn DisplayMemberBinding="{Binding Converter={StaticResource AircraftLocationStringConverter}}" Width="160">
```

# What We Learned – User Controls

```xml
<TabItem Header="Due for Maintenance">
    <local:AllAircraft x:Name="ucDueForMaint" />
</TabItem>
<TabItem Header="Unscheduled Aircraft">
    <local:AllAircraft x:Name="ucUnscheduled" />
</TabItem>
<TabItem Header="All Aircraft">
    <local:AllAircraft x:Name="ucAllAircraft" />
</TabItem>
<TabItem Header="All Flights">
    <local:UpcomingFlights x:Name="ucFlights" />
</TabItem>
<TabItem Header="All Services" >
    <local:AllServices x:Name="ucAllServices"/>
</TabItem>
```

- packages
- UserControls
  - AllAircraft.xaml
    - AllAircraft.xaml.cs
  - AllServices.xaml
    - AllServices.xaml.cs
  - UpcomingFlights.xaml
    - UpcomingFlights.xaml.cs
- .gitignore

# What We Learned

Save Reports as PDF file:

```
PrintDialog printDialog = new PrintDialog();
if (printDialog.ShowDialog() == true)
{
    printDialog.PrintVisual(print, "ReportService");
}
this.DialogResult = true;
```
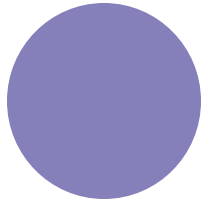
Styling PDF File by using:

**MaterialDesignThemes** by James Willock                    4.3.0
ResourceDictionary instances containing Material Design templates and styles
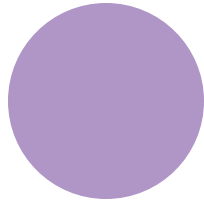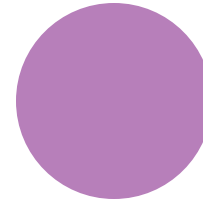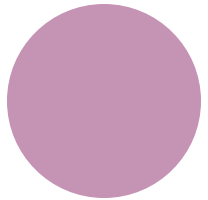for WPF controls in .NET.
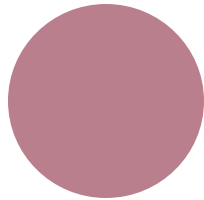
# Future Work

Login
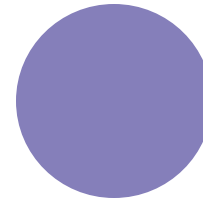
Different types of checks/issue tracking

Having specific bases for maintenance

Incorporating flight hours

Attaching images of snags (broken stuff) to the service reports

API for flight times instead of routes

# Summary

Accomplished many of the goals in our proposal

Created a functional Airplane Maintenance windows application