



ITravel

By:

Anastassia Titova

Andrey Lyamkin

Jessica Boudreault

An app that lets you document past trips
and plan future ones

The world of Travel Apps

Mostly just mobile friendly

Journal-like layout

Made for scrap-booking your trip or blogging

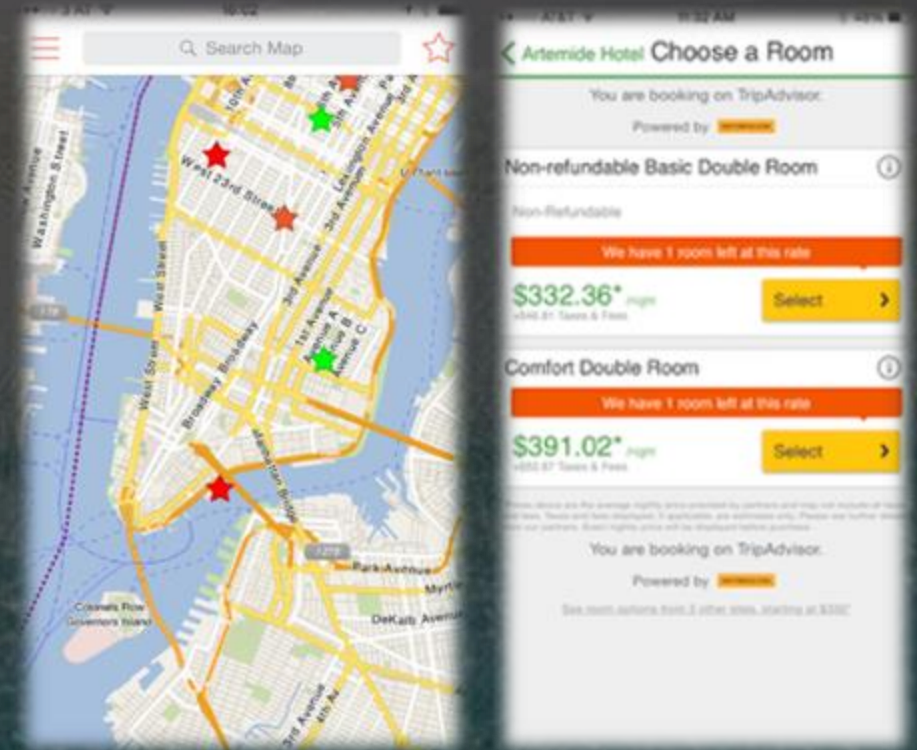


The world of Travel Planners

Made for booking hotels and flights

Map locations and directions

Web-based, therefore un-personalized



Features of our app

Joining both ideas together

- Save trip locations
- Important dates
- Map locations with pictures
- Picture gallery with descriptions
- Stores videos
- Budget creation
- Trip ratings

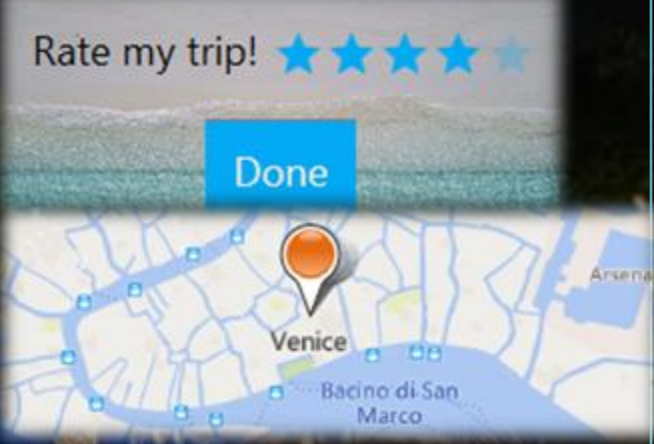
Add video
Delete video
Update video
Video Description



Other Earth



Country	Location
England	London
France	Paris
Italy	Venice
Canada	Montreal



Add/Update Trip

Country: England

Location: London

Start Date: 2021-04-03

End Date: 2021-03-10

Comments: Big Ben was amazing!

End Date warning message

 You are travelling back in Time.
Do not step on butterflies!

OK

Update

We wanted to limit the user as little as possible!

Design Challenges and Solutions

Learning Material Design:

- Manual code for styling

```
Style="{StaticResource btnBlue}"/>
```



```
<ResourceDictionary>  
  <Style TargetType="Button" x:Key="btnBlue">  
    <Setter Property="Background" Value="#FF03A9F4"/>  
    <Setter Property="Foreground" Value="AntiqueWhite"/>  
  </Style>  
</ResourceDictionary>
```

- Material Design code

```
<materialDesign:PackIcon Foreground="AntiqueWhite" />  
<Button.Style>
```

In MainWindow.xaml

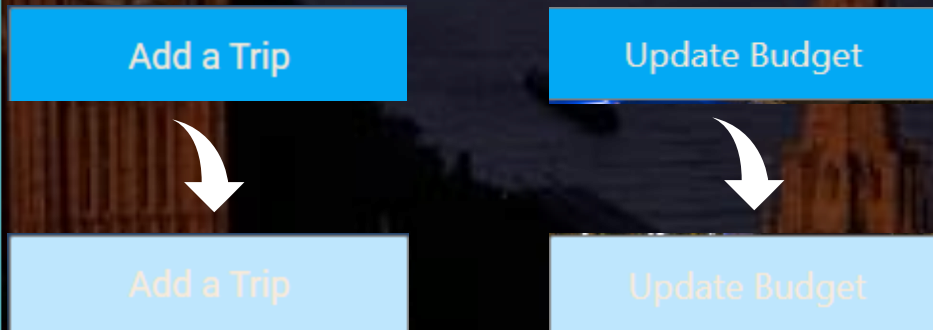
```
.Dark.xaml" />  
.Defaults.xaml" />  
MaterialDesignColor.LightBlue.xaml"
```

In App.xaml

Finding the optimal design layout for the entire app



Making a uniform style



We needed to think, not from the perspective of a programmer, but from the perspective of the user.

#

Welcome to #stop-conflicts-save-yourself!

April 15, 2021



Anastassia 04/15/2021

7.40 main window to add view map



Anastassia 04/15/2021

pushed to git



Jess Bou 04/15/2021

future trip and main windows



Jess Bou 04/15/2021

pushed code



andrew_lyamkin 04/15/2021

Just pushed minor changes in the code. Please check if the project still works correct)



Anastassia 04/15/2021

It works

6.50 MainWindow and AddTrips

Needed to organize
system to avoid conflicts

"It's so... It's so beautiful..."

- Ron Swanson

The picture gallery is a challenging task !!!

How to store Images in the database and organize this data in the gallery for the end user?

Use varbinary(max) data type in the database

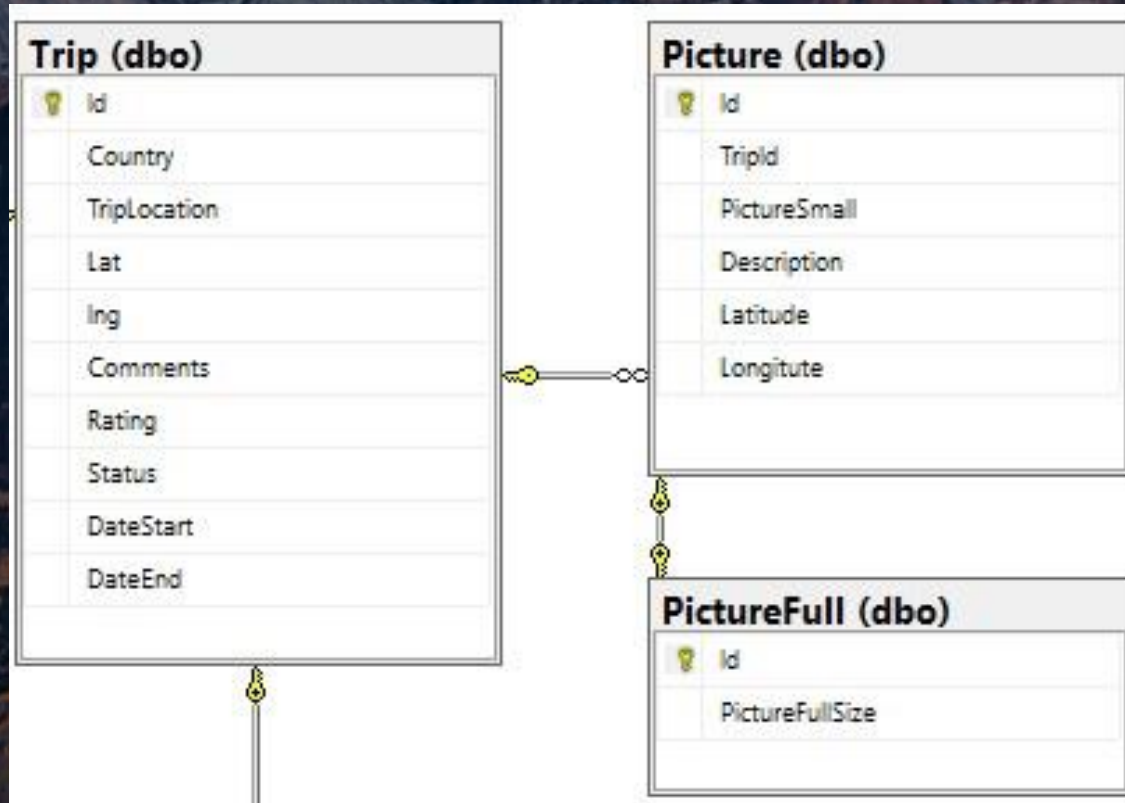
Challenges!

- The gallery may contain 10,100,1000 pictures.
- We don't need to render the original pictures in the UI. Load original only when user needs it.
- Don't lock the app when loading or deleting pictures from DB.

Possible Solutions!

- Create the thumbnail for each picture.
- Resize the original picture.
- Store Original picture separate from thumbnail.
- Use Async/await.

DB to store pictures



Metadata, Description, Lat, Lng ,
Thumbnail, etc..

1

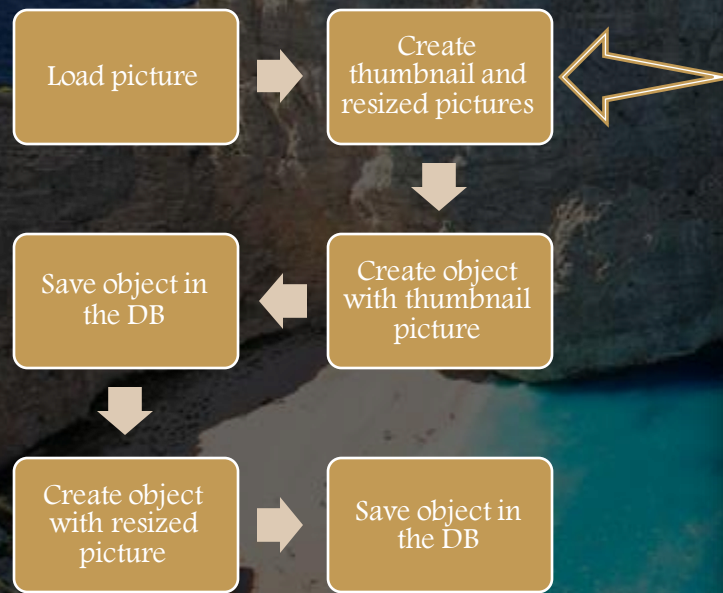
One-to-One relationship

1

Original picture

Thumbnail and resizing

SkiaSharp is a cross-platform 2D graphics API for .NET platforms based on Google's Skia Graphics Library.



```

67  /**** Method to resize picture *****/
68  2 references
69  private byte[] ResizePicture(string inputPath, int size, int quality)
70  {
71      using (var input = File.OpenRead(inputPath))
72      {
73          using (var inputStream = new SKManagedStream(input))
74          {
75              using (var original = SKBitmap.Decode(inputStream))
76              {
77                  int width, height;
78                  if (original.Width > original.Height)
79                  {
80                      width = size;
81                      height = original.Height * size / original.Width;
82                  }
83                  else
84                  {
85                      width = original.Width * size / original.Height;
86                      height = size;
87                  }
88                  using (var resized = original.Resize(new SKImageInfo(width, height), SKBitmapResizeMethod.Lanczos3))
89                  {
90                      if (resized == null) return null;
91                      using (var image = SKImage.FromBitmap(resized))
92                      {
93                          return image.Encode(SKEncodedImageFormat.Jpeg, quality).ToArray();
94                      }
95                  }
96              }
97          }
98      }
99  }

```


Async/await is useful and fun

Step 2: create new task in separate thread

Step 1: makes the function asynchronous

```
/* ***** Method load pictures ***** */
1 reference
private async Task<bool> LoadPictures()
{
    bool done = false;
    await Task.Run(
        () =>
        {
            // 1. Get selected photo(s)
            OpenFileDialog dlg = new OpenFileDialog();
            dlg.Filter = "Image files (*.jpg;*.jpeg;*.gif;*.png)|*.jpg;*.jpeg;*.gif;*.png";
            dlg.Multiselect = true;
        }
    );
}
```

Step 3: use await keyword to wait until task has finished

Step 4: continue main thread

```
/* ***** Listener Add pictures in the dialog ***** */
1 reference
private async void btnDlgAddImages_Click(object sender, RoutedEventArgs e)
{
    if (CurrentTrip == null) return;
    try
    {
        using (TravelDbContext ctx = new TravelDbContext())
        {
            pbStatus.Visibility = Visibility.Visible;
            bool done = await LoadPictures();
            if (done)
            {
                ListBoxDlgGallery.ItemsSource = ctx.Pictures.Where(picture =>
                    picture.TripId == CurrentTrip.Id).ToList(); //ex: SystemException
            }
            pbStatus.Visibility = Visibility.Hidden;
        }
    }
    catch (SystemException ex)
    {
        MessageBox.Show(ex.Message, "Database operation failed",
            MessageBoxButton.OK, MessageBoxImage.Warning);
    }
}
```


Code duplication is evil, but sometimes we had to accept it...Video gallery

The logic is similar to the picture gallery but a little bit more complicated...Just a bit...

Challenges!

- Create the thumbnail from the original video
- Play video when user double click on it (not implemented)
- Don't lock the app when loading or deleting videos from DB.

Possible Solutions!

- Create the thumbnail for each loaded video.
- Store Original video separate from thumbnail.
- Use Async/await.

Thumbnail for video

Media player is the main tool

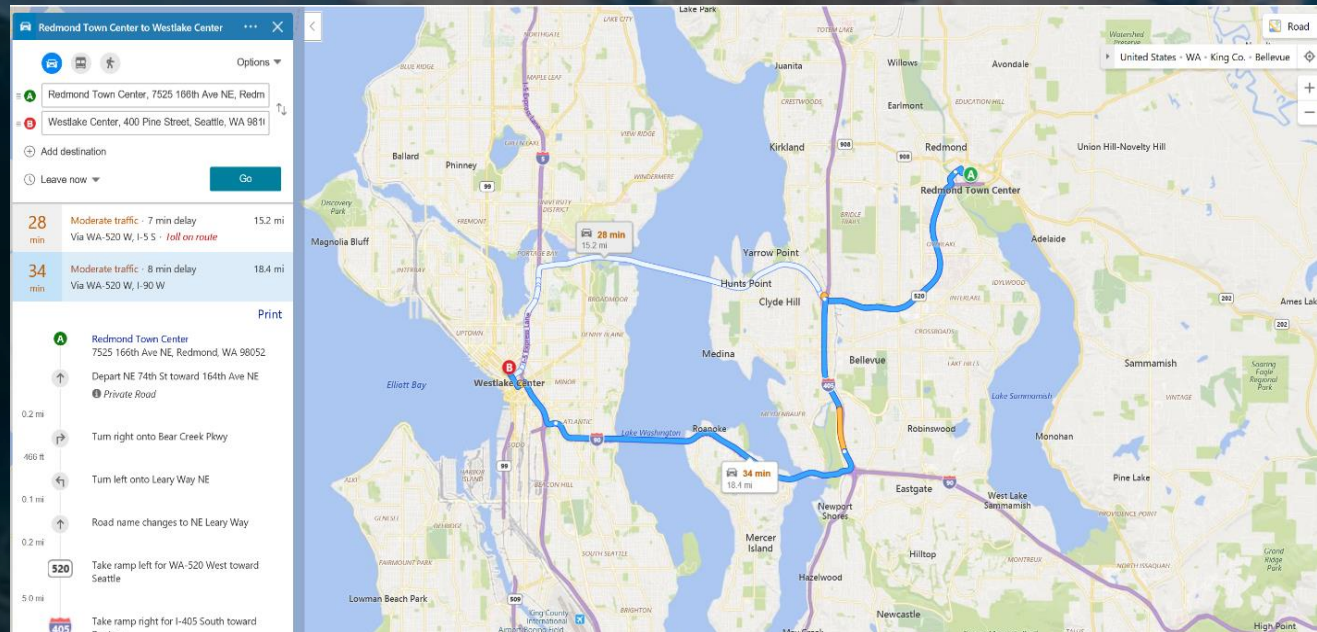
```
Method create Thumbnail Frame *****/
reference
private byte[] CreateThumbNailFrame(string videoPath)
{
    byte[] CurrThumbnailFromVideo;
    MediaPlayer mediaPlayer = new MediaPlayer();
    mediaPlayer.ScrubbingEnabled = true;
    mediaPlayer.Position = TimeSpan.FromSeconds(0);
    mediaPlayer.Open(new Uri(videoPath));
    System.Threading.Thread.Sleep(2000); // TODO:Modify method. Need listener to
        load player
    DrawingVisual drawingVisual = new DrawingVisual();
    DrawingContext drawingContext = drawingVisual.RenderOpen();
    drawingContext.DrawVideo(mediaPlayer, new Rect(0, 0, 160, 100));
    drawingContext.Close();
    double dpiX = 1 / 200;
    double dpiY = 1 / 200;
    RenderTargetBitmap bmp = new RenderTargetBitmap(160, 100, dpiX, dpiY,
        PixelFormats.Pbgra32);
    bmp.Render(drawingVisual);
    //</ draw video_image >
    var encoder = new JpegBitmapEncoder();
    encoder.Frames.Add(BitmapFrame.Create(bmp));
    using (var stream = new MemoryStream())
    {
        encoder.Save(stream);
        CurrThumbnailFromVideo = stream.ToArray();
    }
    mediaPlayer.Close();
    return CurrThumbnailFromVideo;
}
```

Step 1: Create media player in background. Load file from disk.

Step 2: Capture the first frame from video.

Step 3: Save thumbnail to stream and convert to byte array

Bing map – the vision



What did I want to see:

- Pin location on the map
- Place image on the map
- User's search
- Calculate the distance to the search destination from the user's location

The toughest challenge that stalked me for the last 8 days(it never went away).

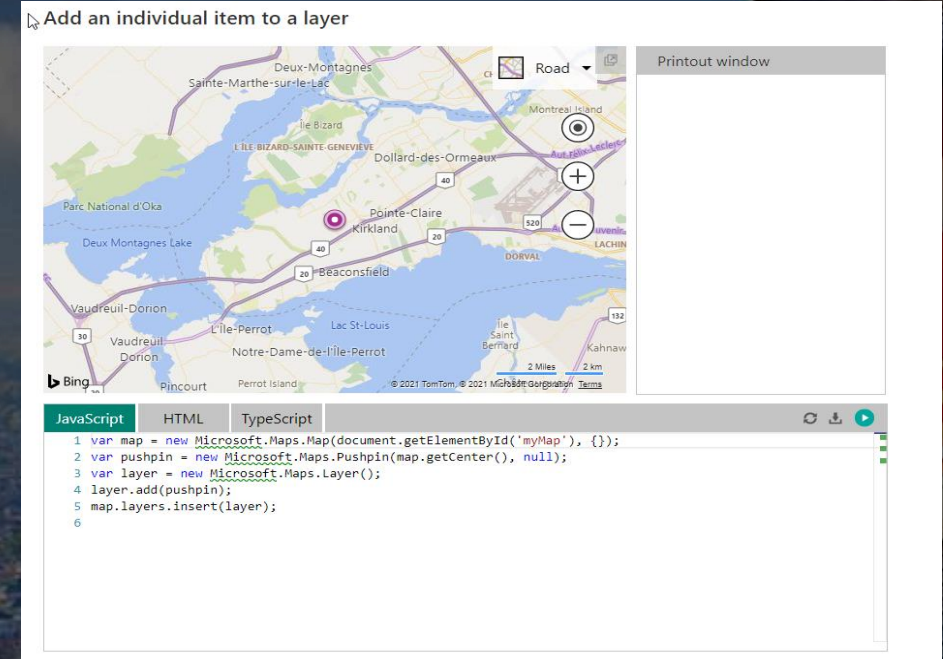
Microsoft WPF Bing Map Tutorial – only XAML, no C# code-behind

Adding Image Media to the Map

To display images on your map, you must add an [Image](#) object (a class in the System.Windows.Controls namespace) to a [MapLayer](#). The [Source](#) property defines the type of image that you want to display ([ImageSource](#)). For example, you can add a [BitmapImage](#) to the map, myMap when you click the “Add Image” button in the following example.

```
XAML
<Window x:Class="WPFTestApplication.ShowImage"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:m="clr-namespace:Microsoft.Maps.MapControl.WPF;assembly=Microsoft.Maps.MapControl.WPF"
        Width="1024" Height="768">

    <Grid x:Name="LayoutRoot" Background="White" >
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
            <RowDefinition Height="40" />
        </Grid.RowDefinitions>
        <m:Map x:Name="myMap"
              Grid.Row="0" Grid.RowSpan="2"
              Center="37.1481402218342,-119.644248783588"
              ZoomLevel="6"
              CredentialsProvider="InsertBingMapsKey" />
        <Button Click="addImageToMap" Opacity="0.8" Width="300" Height="40" Grid.Row="1" >
            <TextBlock Show Image</TextBlock>
        </Button>
    </Grid>
</Window>
```



MicrosoftBing – DevCenter – great tutorial in JS and HTML

FindLocation()

Libraries used : BingMapREST Tool and MicrosoftMaps.MapControlWPF

```
string country = tbCountry.Text; // read user inputs
string place = tbLocation.Text; // read user inputs
// if user input is empty set country to world, if location is empty, set default location
if (country.Length == 0)
{
    country = "world";
    if (place.Length == 0) place = "Montreal";
}
string geocodeRequest = $"http://dev.virtualearth.net/REST/v1/Locations?&countryRegion={country}&locality={place}&maxResults=1&key={Key}"; // a request BingMapAPI
string response;
Microsoft.Maps.MapControl.WPF.Location loc = new Microsoft.Maps.MapControl.WPF.Location();

// get the requested information from virtualearth
try
{
    using (var client = new WebClient())
    {
        response = client.DownloadString(geocodeRequest); // ArgumentNullException, WebException
        Console.WriteLine(response); // to display the info on the console window for the dev
    }
}
```


FindLocation() ~ part 2

Make sure not mix up 2 locations! - BingMapREST.ToolKit and Microsoft.Maps.MapControl.WPF

```
//to serilize the json data string
System.Runtime.Serialization.Json.DataContractJsonSerializer ser = new System.Runtime.Serialization.Json.DataContractJsonSerializer(typeof(Response));
using (var es = new MemoryStream(Encoding.Unicode.GetBytes(response)))
{
    //Response is one of the Bing Maps DataContracts
    var mapResponse = (ser.ReadObject(es) as Response);
    // get the first choice from the response ex. InvalidOperationException
    BingMapsRESTToolkit.Location locn = (BingMapsRESTToolkit.Location)mapResponse.ResourceSets.First().Resources.First();
    // get longitude and latitude of the first location
    double latitude = locn.Point.Coordinates[0];
    double longitude = locn.Point.Coordinates[1];
    //create wpf maps location
    loc = new Microsoft.Maps.MapControl.WPF.Location(latitude, longitude);
}
```

A picture is worth a thousand words

Library: System.Drawing.Common

```
using (Bitmap bitmap = new Bitmap(@"D:\tmp\content_example_ibiza.jpg"))
{
    var longitude = GetCoordinateDouble(bitmap.PropertyItems.Single(p => p.Id == 4));
    var latitude = GetCoordinateDouble(bitmap.PropertyItems.Single(p => p.Id == 2));

    Console.WriteLine($"Longitude: {longitude}");
    Console.WriteLine($"Latitude: {latitude}");
}
```

```
private static double GetCoordinateDouble(PropertyItem propItem)
{
    uint degreesNumerator = BitConverter.ToUInt32(propItem.Value, 0);
    uint degreesDenominator = BitConverter.ToUInt32(propItem.Value, 4);
    double degrees = degreesNumerator / (double)degreesDenominator;

    uint minutesNumerator = BitConverter.ToUInt32(propItem.Value, 8);
    uint minutesDenominator = BitConverter.ToUInt32(propItem.Value, 12);
    double minutes = minutesNumerator / (double)minutesDenominator;

    uint secondsNumerator = BitConverter.ToUInt32(propItem.Value, 16);
    uint secondsDenominator = BitConverter.ToUInt32(propItem.Value, 20);
    double seconds = secondsNumerator / (double)secondsDenominator;

    double coordinate = degrees + (minutes / 60d) + (seconds / 3600d);
    string gpsRef = System.Text.Encoding.ASCII.GetString(new byte[1] { propItem.Value[0] }); //N, S, E, or W

    if (gpsRef == "S" || gpsRef == "W")
    {
        coordinate = coordinate * -1;
    }
    return coordinate;
}
```

Library : MetadataExtractor

```
// read latitude and longitude from exif of image file
var gps = ImageMetadataReader.ReadMetadata(dlg.FileName).OfType<GpsDirectory>()
    .FirstOrDefault();
if (gps != null)
{
    var location = gps.GetGeoLocation();
    lat = (decimal)location.Latitude;
    lng = (decimal)location.Longitude;
}
```

Main challenge to find images
with gps data to test the code

Layers

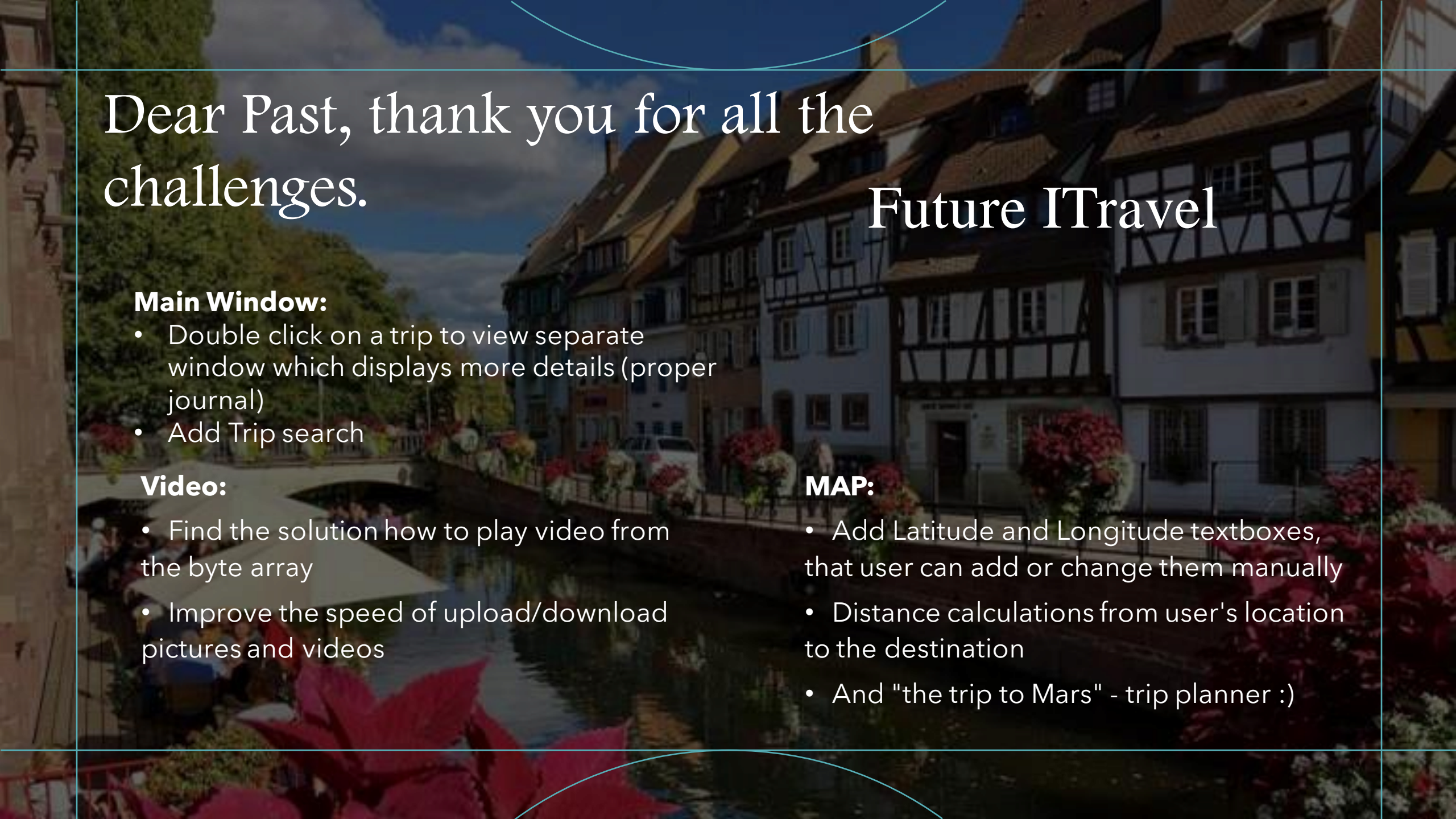
Layers:

Add Image layer to the map layer.

```
MapLayer imageLayer = new MapLayer(); // new map layer
Image image = new Image();
image.Height = 150; // image height

// convert byte array to BitmapImage
byte[] img = selectedImage.PictureSmall;
BitmapImage myBitmapImage = Utils.GenericUtils.ByteArrayToImage(img, 150);
// image setup
image.Source = myBitmapImage;
image.Opacity = 1;
image.Stretch = System.Windows.Media.Stretch.None;

//Center the image around the location specified
PositionOrigin position = PositionOrigin.Center;
//Add the image to the defined map layer
selectedLocation = new Microsoft.Maps.MapControl.WPF.Location((double)selectedImage.Latitude, (double)selectedImage.Longitude);
myMap.SetView(center: selectedLocation, 15);
imageLayer.AddChild(image, selectedLocation, position);
//Add the image layer to the map
myMap.Children.Add(imageLayer);
```

The background image shows a picturesque European town. In the foreground, there's a canal with a small bridge. The buildings are multi-story, featuring half-timbered construction with white plaster and dark wood beams. The roofs are steep and tiled. There are some red flowers in the bottom left corner. The sky is blue with some clouds.

Dear Past, thank you for all the challenges.

Future ITravel

Main Window:

- Double click on a trip to view separate window which displays more details (proper journal)
- Add Trip search

Video:

- Find the solution how to play video from the byte array
- Improve the speed of upload/download pictures and videos

MAP:

- Add Latitude and Longitude textboxes, that user can add or change them manually
- Distance calculations from user's location to the destination
- And "the trip to Mars" - trip planner :)

CONCLUSION



Great things are not done
by one person. They are
done by a team of people.
Steve Jobs

Whatever you do,
always give 100%.
Unless you're
donating blood.
Bill Murray