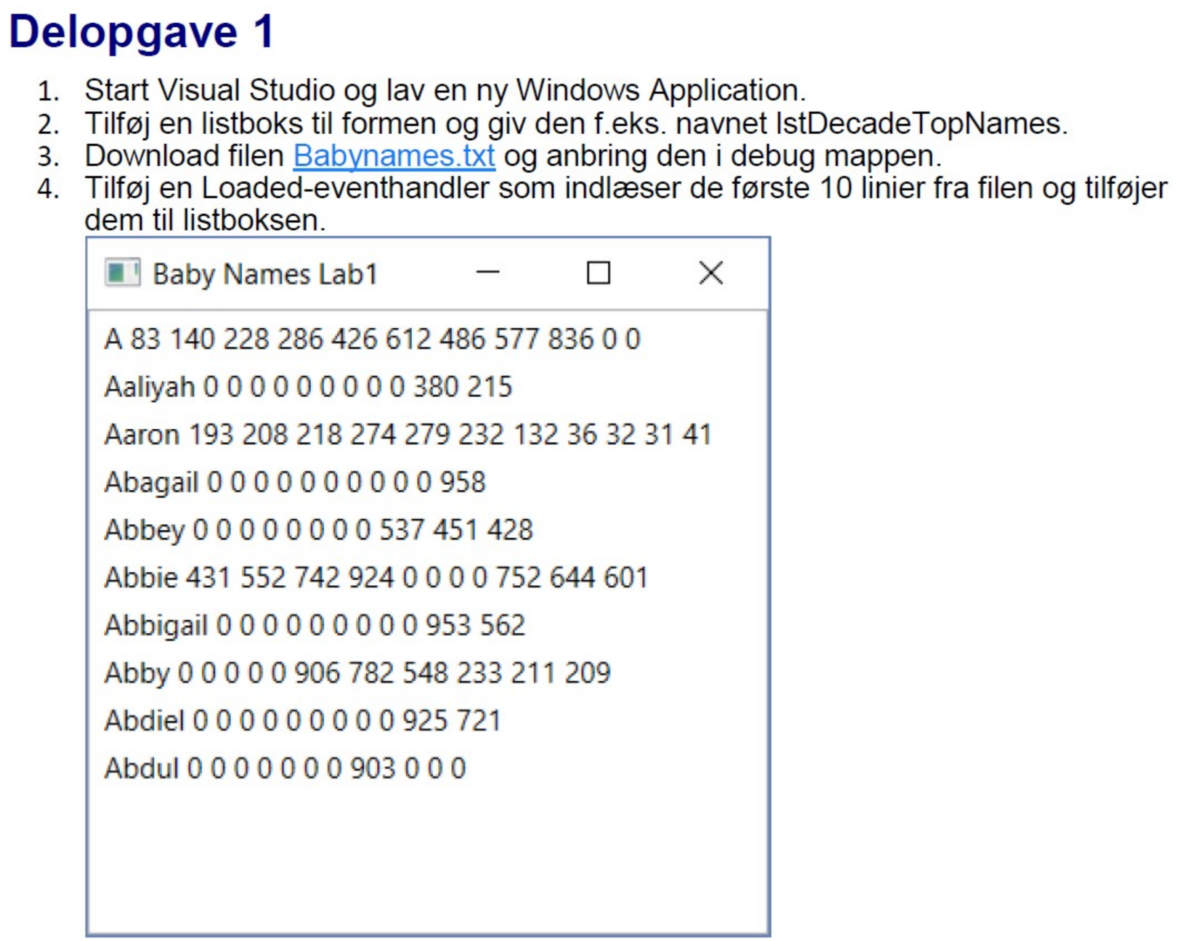
Exercise 2 Baby Names Guidance

AARHUS UNIVERSITY
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Delopgave 1



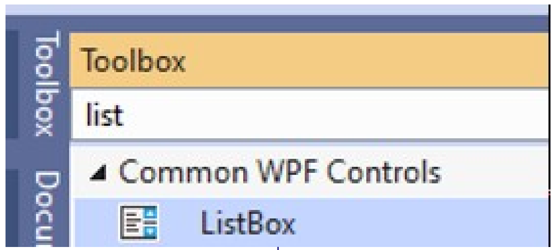
Delopgave 1

1. Start Visual Studio og lav en ny Windows Application.
2. Tilføj en listboks til formen og giv den f.eks. navnet lstDecadeTopNames.
3. Download filen Babynames.txt og anbring den i debug mappen.
4. Tilføj en Loaded-eventhandler som indlæser de første 10 linier fra filen og tilføjer dem til listboksen.

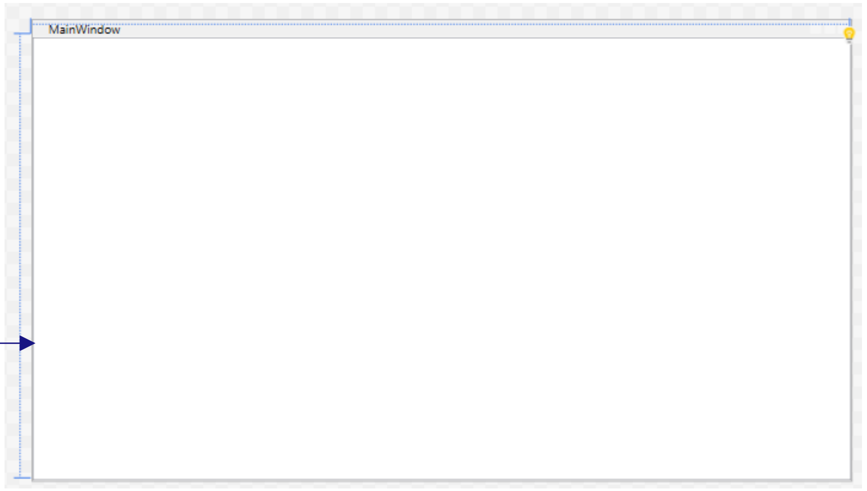
```
A 83 140 228 286 426 612 486 577 836 0 0
Aaliyah 0 0 0 0 0 0 0 0 0 380 215
Aaron 193 208 218 274 279 232 132 36 32 31 41
Abagail 0 0 0 0 0 0 0 0 0 0 958
Abbey 0 0 0 0 0 0 0 0 537 451 428
Abbie 431 552 742 924 0 0 0 0 752 644 601
Abbigail 0 0 0 0 0 0 0 0 0 953 562
Abby 0 0 0 0 0 906 782 548 233 211 209
Abdiel 0 0 0 0 0 0 0 0 0 925 721
Abdul 0 0 0 0 0 0 0 903 0 0 0
```

1. Start Visual Studio og lav en ny Windows Application
2. Tilføj en listboks til formen og giv den f.eks. navnet lstDecadeTopNames.

In XAML



Drag & drop



1. Find ListBox in Toolbox
2. Drag & Drop “Listbox” into main window
3. Give **name** for Listbox in Xaml code

ex.

```
<Grid>
    <ListBox Name="lstDecadeTopNames" />
</Grid>
```

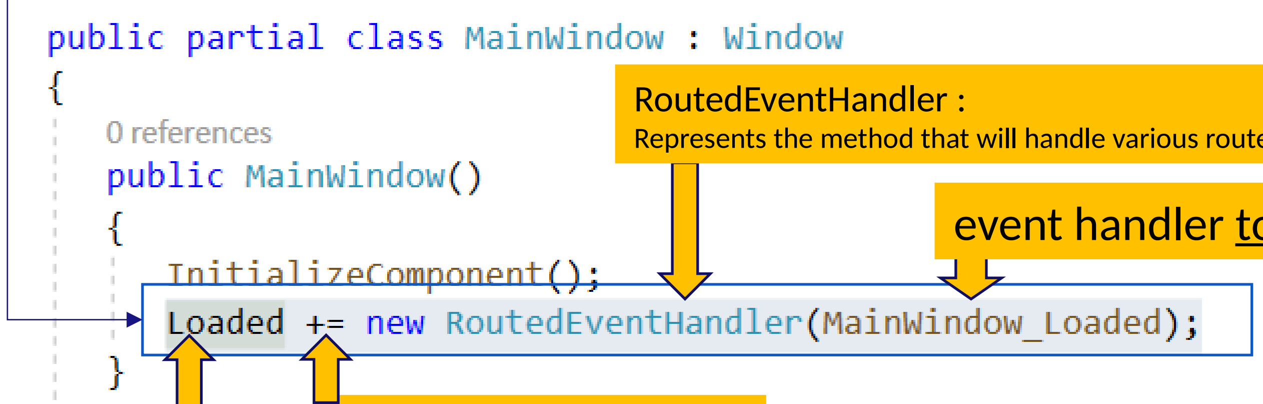
“give element name in the XAML to attach handlers in the code behind”

3. Download filen Babynames.txt og anbring den i debug mappen

- *Brightspace -> 02 Controls and Events -> Exercises-> babynames.txt*

4. Tilføj en Loaded-eventhandler som indlæser de første 10 linier fra filen og tilføjerdem til listboksen.

- Tilføj en Loaded-eventhandler (refer Routed Event slide 13 : attaching event handler in code)



```
public partial class MainWindow : Window
{
    0 references
    public MainWindow()
    {
        InitializeComponent();
        Loaded += new RoutedEventHandler(MainWindow_Loaded);
    }
```

RoutedEventHandler :
Represents the method that will handle various routed events

event handler <u>to implement the logic</u> for ‘delopgave1’. (next slide)

Attaching event handler

<u>Loaded</u> Event :
Occurs when the element is laid out, rendered, and ready for interaction in an element initialization sequence

AARHUS UNIVERSITY
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Implement “MainWindow_Loaded” event handler

- Inside event handler
  - indlæser de første 10 linier fra filen og tilføjerdem til listboksen.

```
void MainWindow_Loaded(object sender, RoutedEventArgs e)
{
    string filename;
    filename = SIO.Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "babynames.txt");
    SIO.StreamReader reader = null;
    try
    {
       reader = new SIO.StreamReader(filename);
       {
          for (int i = 0; i < 10; ++i)
          {
              lstDecadeTopNames.Items.Add(reader.ReadLine());
          }
       }
    }
    finally
    {
       if (reader != null)
          reader.Close();
    }
}
```

Indlæser filen

de første 10 linier tilføjer til listboksen

Controls Slide19 : Items Property to add Items

Listbox name given in xaml

Alterantive Del1

Only To demonstrate that there are also several different ways to implement.

But best way is the event handler to attach in the c# code as like løsningsforslag.

In Xaml

```
<Window
        …
        Title="MainWindow" Height="450" Width="800" Loaded="OnLoad">
    <Grid>
        <ListBox Name="lstTopNames"/>
    </Grid>
</Window>
```

You can also set Loaded Event in window in xaml and attaching event handler in xaml

However, this is just to demonstate the example that attaching event also can be done in xaml. Usually, event handler is better to be attached in code behind (C#)

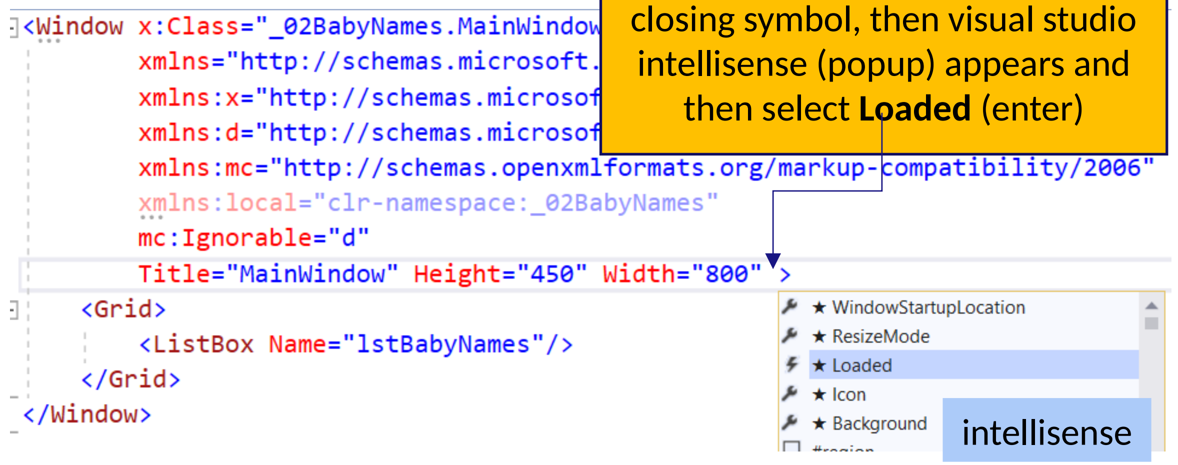
In C#

```
public MainWindow()
{
    InitializeComponent();
}

private void OnLoad(object sender, RoutedEventArgs e)
{
    List<string> lines = System.IO.File.ReadLines(@"C:\FED\F22_lab\babynamesDel1\babynames.txt").ToList();

    if (lines != null)
    {
        for (int i = 0; i <= 10; i++)
        {
            lstTopNames.Items.Add(lines[i]);
        }
    }
}
```

Details in next slide

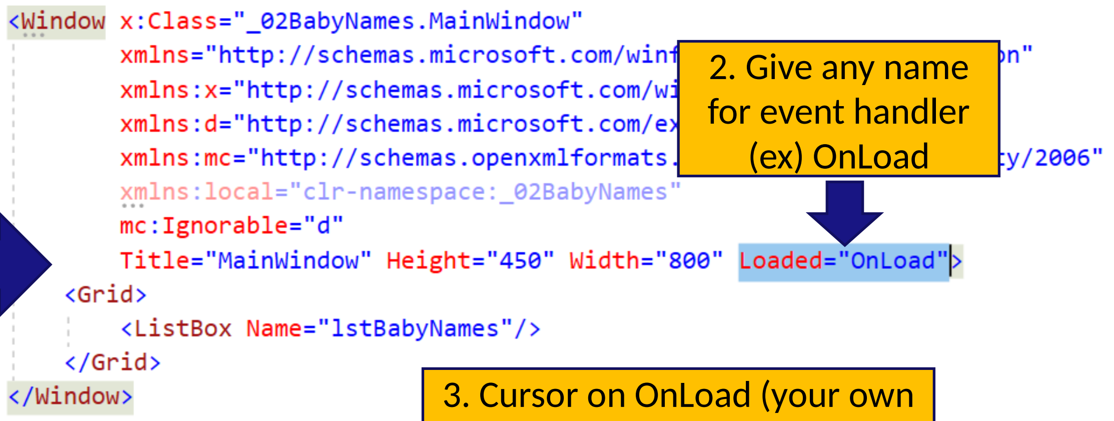
How to use loaded event handler (in xaml)

- There are several ways.
- https://docs.microsoft.com/en-us/dotnet/desktop/wpf/advanced/how-to-handle-a-loaded-event?view=netframeworkdesktop-4.8
- In this example, use “**Loaded**” in window.
  - (Loaded reference if interested)
- Example `Loaded="OnLoad"`
- How to :
  - In xaml main window, follow image.



```
<Window x:Class="_02BabyNames.MainWindow
        xmlns="http://schemas.microsoft.
        xmlns:x="http://schemas.microsof
        xmlns:d="http://schemas.microsof
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:_02BabyNames"
        mc:Ignorable="d"
        Title="MainWindow" Height="450" Width="800" >
    <Grid>
        <ListBox Name="lstBabyNames"/>
    </Grid>
</Window>
```

1. Press the space key before closing symbol, then visual studio intellisense (popup) appears and then select **Loaded** (enter)



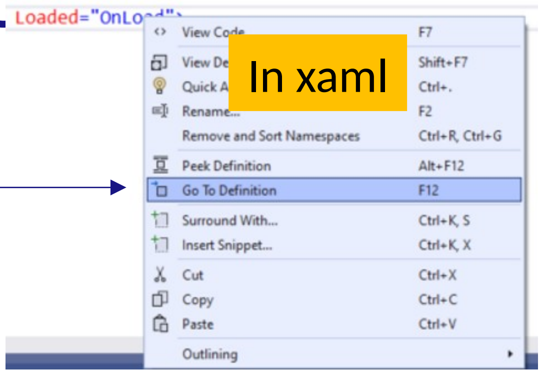
```
<Window x:Class="_02BabyNames.MainWindow"
        xmlns="http://schemas.microsoft.com/winf                              n"
        xmlns:x="http://schemas.microsoft.com/wi
        xmlns:d="http://schemas.microsoft.com/ex
        xmlns:mc="http://schemas.openxmlformats.                            y/2006"
        xmlns:local="clr-namespace:_02BabyNames"
        mc:Ignorable="d"
        Title="MainWindow" Height="450" Width="800" Loaded="OnLoad">
    <Grid>
        <ListBox Name="lstBabyNames"/>
    </Grid>
</Window>
```

2. Give any name for event handler (ex) OnLoad

3. Cursor on OnLoad (your own event handler name) , then Right click on mouse -> Go To Definition (F12) : check next slide

AARHUS UNIVERSITY
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

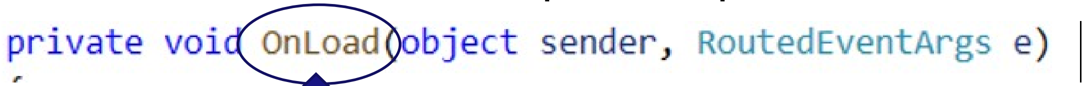
How to implement event handler (C#)



In xaml

- (previous slide, step 3) : Right click on mouse -> "Go to Definition (F12)" will auto generate event handler template in C# code

- Auto generate event handler temaple example in C# code



```
private void OnLoad(object sender, RoutedEventArgs e)
```

Any name you gave in XAML, event handler code is generated with the given name.

- Then implement the logic inside event handler (C#), the logic need to do :
  1. Read the babynames.txt file
     - remember the path of the file , need to give path if it is not in the same folder (if not in the same folder, copy the entire path)
  2. Read lines and then add the lines into listbox

```
List<string> lines = System.IO.File.ReadLines(@"C:\FED\F22_lab\babynamesDel1\
babynames.txt").ToList();

if (lines != null)
{
    for (int i = 0; i <= 10; i++)
    {
        lstBabyNames.Items.Add(lines[i]);
    }
}
```

Read the lines from the file

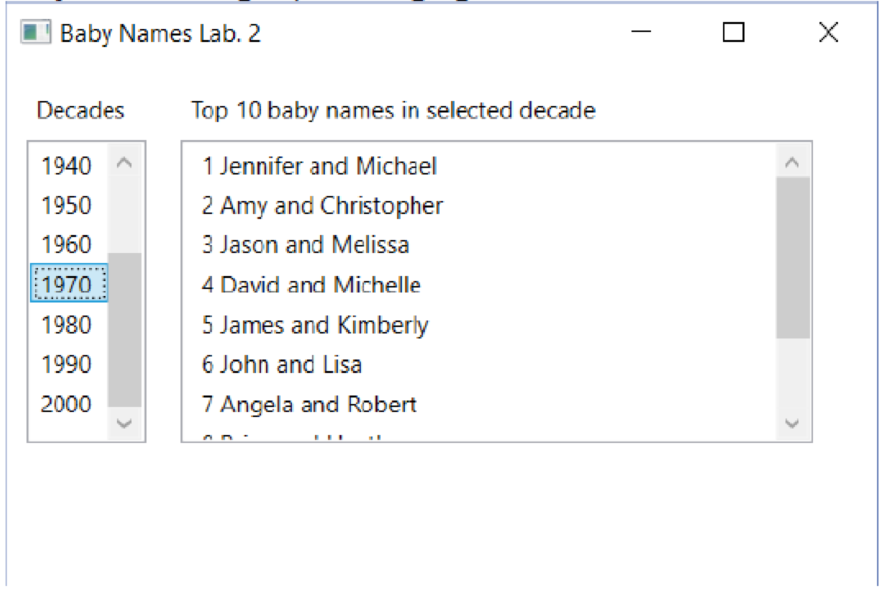
Items Property to add items into Listbox (Controls slide19)

ListBox name given in xaml

Delopgave2

Delopgave 2

1. Udbyg koden således at den laver en instans af BabyName-klassen for hver linie i filen babynames.txt, og tilføj babyName objekterne til en collection klasse - brug f.eks. List<BabyName>.
2. Tilføj en listboks (eller evt. en comboboks) hvor brugeren kan vælge årti (decade).
3. For at få en hurtig respons når brugeren vælger årti, kan man f.eks. lagre information om alle top-10 navnene for hvert årti i et 2-dimensinelt array ~ matrix (dette kan f.eks. ske i forbindelse med indlæsningen), men denne funktionalitet kan også implementeres på anden vis.
4. Tilføj eventhandler til listboksen fra pkt. 2 som opdaterer listboksen med "Top 10 baby names ..." og finpuds brugergrænsefladen.



1940
1950
1960
1970
1980
1990
2000

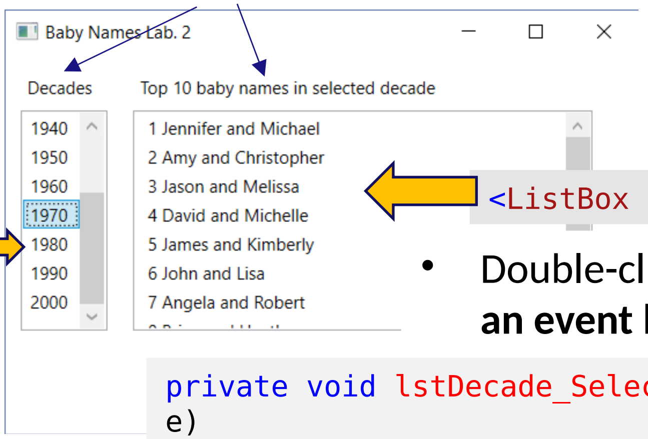
1 Jennifer and Michael
2 Amy and Christopher
3 Jason and Melissa
4 David and Michelle
5 James and Kimberly
6 John and Lisa
7 Angela and Robert

(In Xaml) MainWindow.xaml

ListBox1 name

```
<Grid>
    <ListBox Name="lstDecade" ..>
            <ListBoxItem>1900</ListBoxItem>
            <ListBoxItem>1910</ListBoxItem>
            <ListBoxItem>1920</ListBoxItem>
            <ListBoxItem>1930</ListBoxItem>
            <ListBoxItem>1940</ListBoxItem>
            <ListBoxItem>1950</ListBoxItem>
            <ListBoxItem>1960</ListBoxItem>
            <ListBoxItem>1970</ListBoxItem>
            <ListBoxItem>1980</ListBoxItem>
            <ListBoxItem>1990</ListBoxItem>
            <ListBoxItem>2000</ListBoxItem>
            <ListBoxItem />
      </ListBox>
```

Label



Baby Names Lab. 2

Decades

Top 10 baby names in selected decade

1940
1950
1960
1970
1980
1990
2000

1 Jennifer and Michael
2 Amy and Christopher
3 Jason and Melissa
4 David and Michelle
5 James and Kimberly
6 John and Lisa
7 Angela and Robert

ListBox2 name

```
<ListBox Margin=“.." Name="lstTopBabyNames" />
```

- Double-click inside the empty space in the ListBox1 to **add an event handler method for the SelectionChanged event.**

```
private void lstDecade_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
}
```

(Auto generated event handler In C#)

(In C#) MainWindow.xaml.cs

```
public partial class MainWindow : Window
{
   private List<BabyName> namesCollection;
   private string[,] rankMatrix = new string[11, 10];

   public MainWindow()
   {
      InitializeComponent();
      Loaded += new RoutedEventHandler(MainWindow_Loaded);
      lstDecade.SelectionChanged += new SelectionChangedEventHandler(lstDecade_SelectionChanged);
   }
```

BabyName List

Declaration of 2-dimentional string array ‘rankMatrix’ with size 11, 10
(11: from 1900 to 2000, 11 items, 10 : Top 10 baby names each decade)
(pkt 3. for hvert årti i et 2-dimensinelt array fra opgave beskrivelse)

SelectionChangedEventHandler :
Represents the method that will handle the SelectionChanged routed event.

ListBox1 name from xaml

SelectionChanged Routed Event:
Occurs when the selection of a selector **changes**.

Attatching SelectionChangedEventHandler

New Event Handler to update the Listbox2 when new decade selected from listbox1(next slide) for pkt.4 i Delopgave2

Implement ”lstDecade_SelectionChanged” event handler

MainWindow.xaml.cs

```
void lstDecade_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    ListBoxItem item;

    item = lstDecade.SelectedItem as ListBoxItem; // Because the data entries is done by use of ListBoxItem in XAML
    if (item != null)
    {
        int decade = (Convert.ToInt32(item.Content) - 1900) / 10;
        lstTopBabyNames.Items.Clear();
        for (int i = 1; i < 11; ++i)
        {
            lstTopBabyNames.Items.Add(string.Format("{0} {1}", i, rankMatrix[decade, i - 1]));
        }
    }
}
```

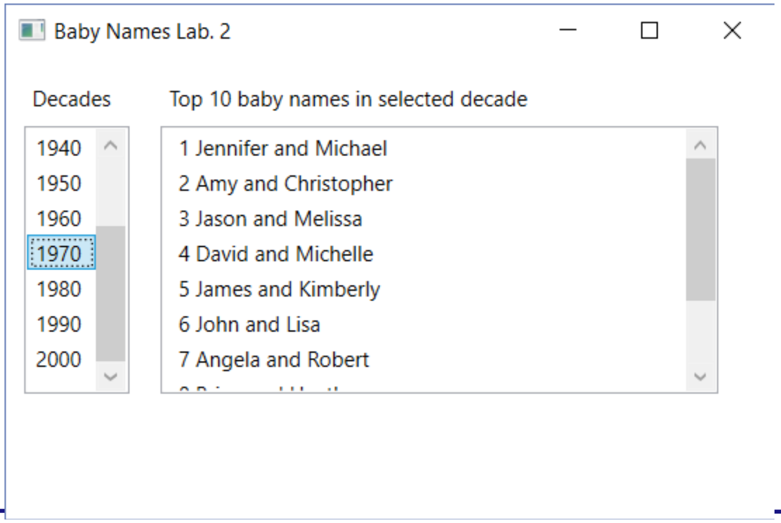
item contains the selected year from Listbox1

Convert to string format

ListBox2 name

Add items into ListBox2

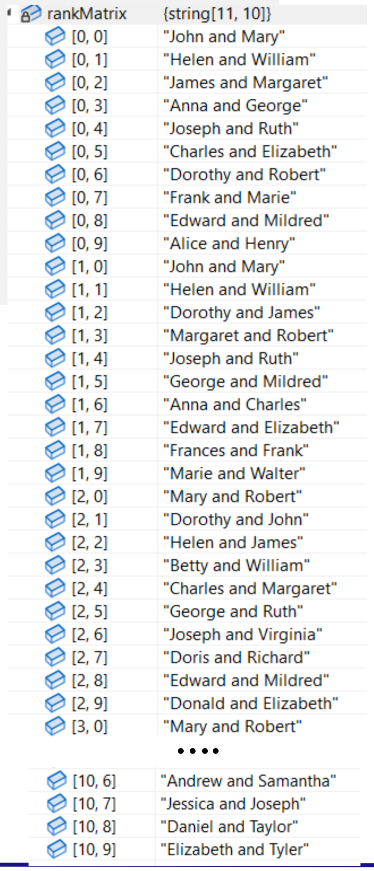
2-dimention array from the previous slide



Decades

Top 10 baby names in selected decade

Decades	Top 10 baby names in selected decade
1940	1 Jennifer and Michael
1950	2 Amy and Christopher
1960	3 Jason and Melissa
1970	4 David and Michelle
1980	5 James and Kimberly
1990	6 John and Lisa
2000	7 Angela and Robert



rankMatrix	{string[11, 10]}
[0, 0]	"John and Mary"
[0, 1]	"Helen and William"
[0, 2]	"James and Margaret"
[0, 3]	"Anna and George"
[0, 4]	"Joseph and Ruth"
[0, 5]	"Charles and Elizabeth"
[0, 6]	"Dorothy and Robert"
[0, 7]	"Frank and Marie"
[0, 8]	"Edward and Mildred"
[0, 9]	"Alice and Henry"
[1, 0]	"John and Mary"
[1, 1]	"Helen and William"
[1, 2]	"Dorothy and James"
[1, 3]	"Margaret and Robert"
[1, 4]	"Joseph and Ruth"
[1, 5]	"George and Mildred"
[1, 6]	"Anna and Charles"
[1, 7]	"Edward and Elizabeth"
[1, 8]	"Frances and Frank"
[1, 9]	"Marie and Walter"
[2, 0]	"Mary and Robert"
[2, 1]	"Dorothy and John"
[2, 2]	"Helen and James"
[2, 3]	"Betty and William"
[2, 4]	"Charles and Margaret"
[2, 5]	"George and Ruth"
[2, 6]	"Joseph and Virginia"
[2, 7]	"Doris and Richard"
[2, 8]	"Edward and Mildred"
[2, 9]	"Donald and Elizabeth"
[3, 0]	"Mary and Robert"
••••	
[10, 6]	"Andrew and Samantha"
[10, 7]	"Jessica and Joseph"
[10, 8]	"Daniel and Taylor"
[10, 9]	"Elizabeth and Tyler"

Udbyg “MainWindow_Loaded” event handler

Udbyg koden således at den laver en instans af BabyName-klassen for hver linie ifilen babynames.txt, og tilføj babyName objekterne til en collection klasse - brugf.eks. List<BabyName>.

foreach is a loop that iterates through a collection of items.

```
void MainWindow_Loaded(object sender, RoutedEventArgs e)
{
   string file = SIO.Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "babynames.txt");
   this.namesCollection = Utility.ReadBabyNameData(file);

   foreach (BabyName name in namesCollection)
   {
      for (int decade = 1900; decade < 2010; decade += 10)
      {
         int rank = name.Rank(decade);
         int decadeIndex = (decade - 1900) / 10;
         if (0 < rank && rank < 11)
            if (rankMatrix[decadeIndex, rank - 1] == null)
               rankMatrix[decadeIndex, rank - 1] = name.Name;
            else
               rankMatrix[decadeIndex, rank - 1] += " and " + name.Name;
      }
   }
}
```

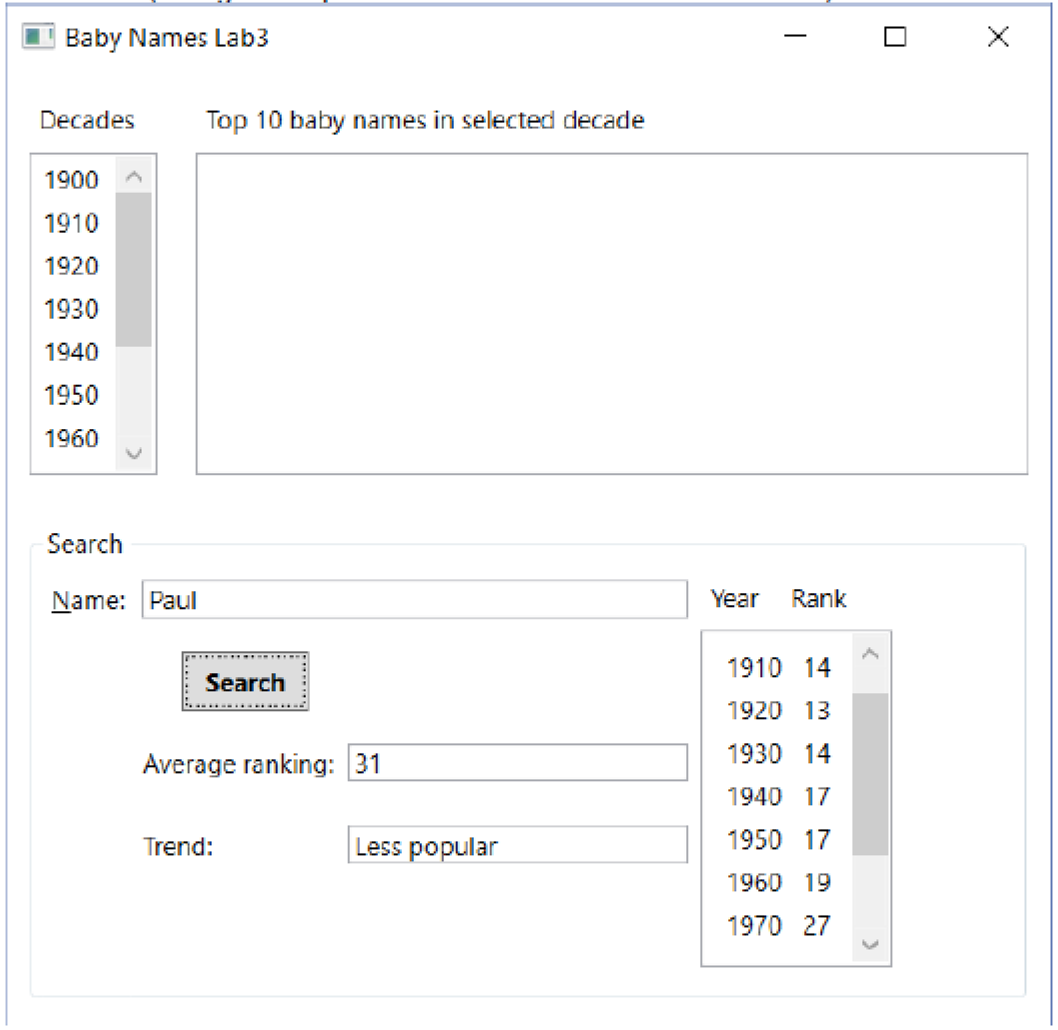
Use as it is

AARHUS UNIVERSITY
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Delopgave3

Delopgave 3

1. Tilføj kontrollerne til brug for søgningen.
2. Tilføj en eventhandler til søge-knappen og implementer søgningen som søger i collection klassen efter det angivne navn, og opdaterer kontrollerne med den fundne information (eller giver en passende besked hvis navnet ikke findes).



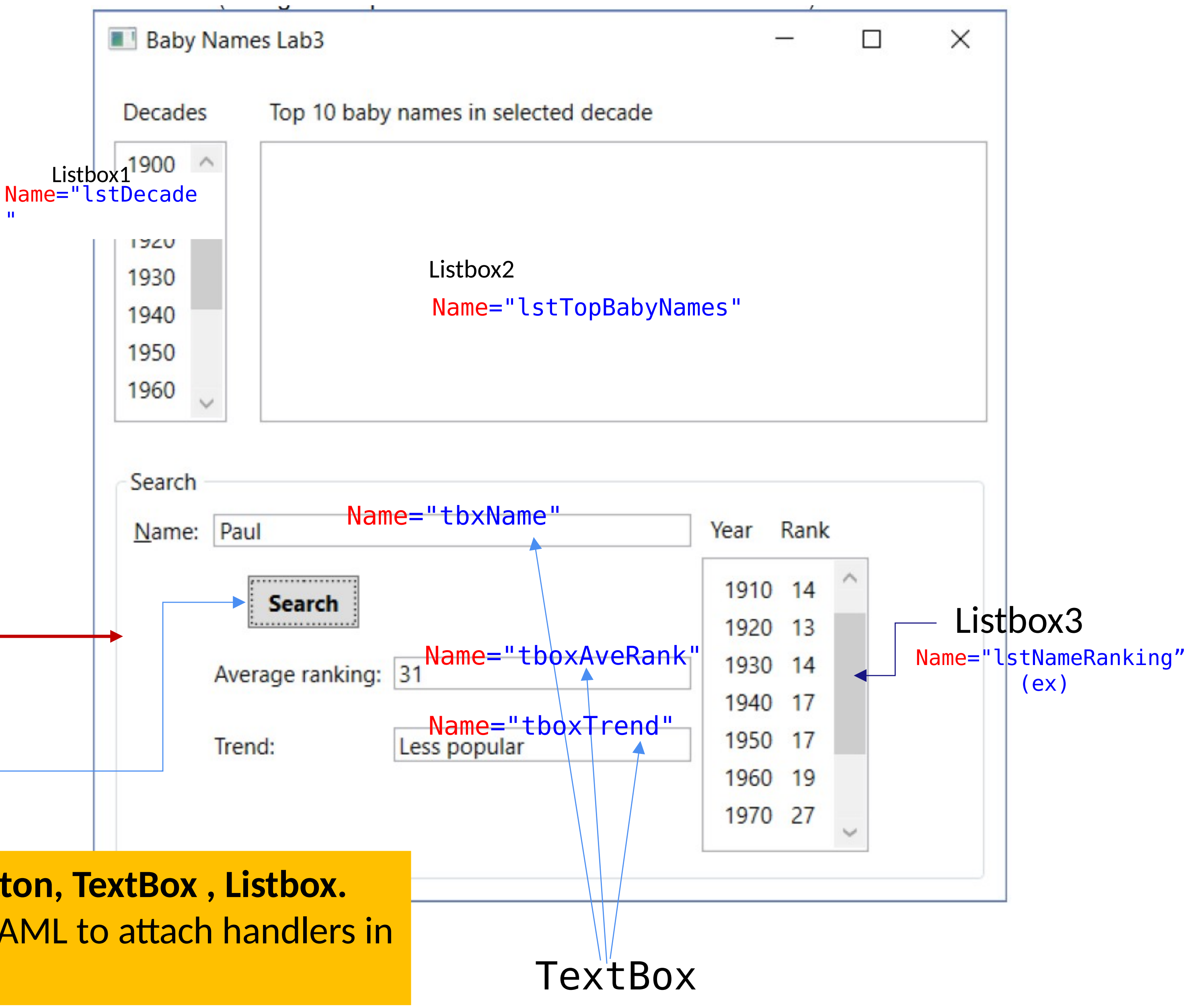
Name: Paul

Average ranking: 31

Trend: Less popular

Year	Rank
1910	14
1920	13
1930	14
1940	17
1950	17
1960	19
1970	27

In Xaml



Baby Names Lab3

Decades   Top 10 baby names in selected decade

Listbox1
Name="lstDecade"

1900
1920
1930
1940
1950
1960

Listbox2
Name="lstTopBabyNames"

Search

Name: Paul   Name="tbxName"

Search

Average ranking: 31   Name="tboxAveRank"

Trend: Less popular   Name="tboxTrend"

Year	Rank
1910	14
1920	13
1930	14
1940	17
1950	17
1960	19
1970	27

Listbox3
Name="lstNameRanking" (ex)

TextBox

Outer : GroupBox
`<GroupBox .. Header="Search" Name="groupBoxSearch">`

Inner : Grid
To add multiple elements

Tilføj en eventhandler til søge-knappen og implementer søgningen
`(ex) Name="btnSearch"`

Remember **give Name for Button, TextBox , Listbox.** (give elements names in the XAML to attach handlers in the code behind)

In C#: MainWindow.xaml.cs

```
private void Search(object sender, RoutedEventArgs e)
{
    // get the name entered by the user:
    string name = tbxInput.Text;
                  TextBox1
    ….
    tboxAveRank.Text = theName.AverageRank().ToString();
    TextBox2

    …
    if (-1 < i && i < namesCollection.Count)
    {
          tblkError.Text = "";
          BabyName theName = namesCollection[i];
          tboxAveRank.Text = theName.AverageRank().ToString();
          if (theName.Trend() > 0)
  TextBox2      tboxTrend.Text = "More popular";
          else if (theName.Trend() == 0)
                tboxTrend.Text = "Inconclusive";
          else
                tboxTrend.Text = "Less popular";


     …
}
```

Example of using textbox names to attach (couple) the necessary logic in C#
Then its value will be updated/presented on textbox in xaml

In C#: MainWindow.xaml.cs

```
public partial class MainWindow3 : Window
{
     private List<BabyName> namesCollection;
     private string[,] rankMatrix = new string[11, 10];

     public MainWindow3()
     {
         InitializeComponent();
         for (int decade = 1900; decade < 2010; decade += 10)
              lstDecade.Items.Add(decade);

         Loaded += new RoutedEventHandler(MainWindow_Loaded);
         lstDecade.SelectionChanged += new SelectionChangedEventHandler(lstDecade_SelectionChanged);
         btnSearch.Click += new RoutedEventHandler(Search);
     }
```

Button name from xaml

Attaching event handler

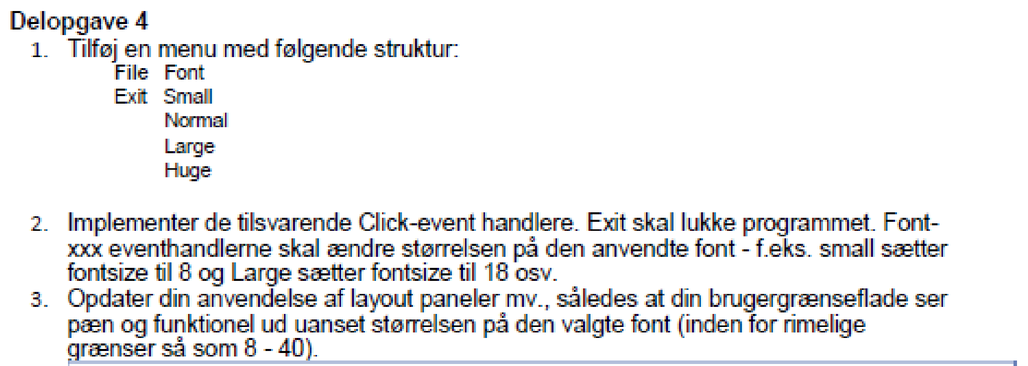
Button click event occurs when the button control is clicked.

Search Event Handler to implement

```
private void Search(object sender, RoutedEventArgs e)
{
    // get the name entered by the user:
    string name = tbxInput.Text;

    ….
    tboxAveRank.Text = theName.AverageRank().ToString();
}
```

AARHUS UNIVERSITY
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Delopgave4

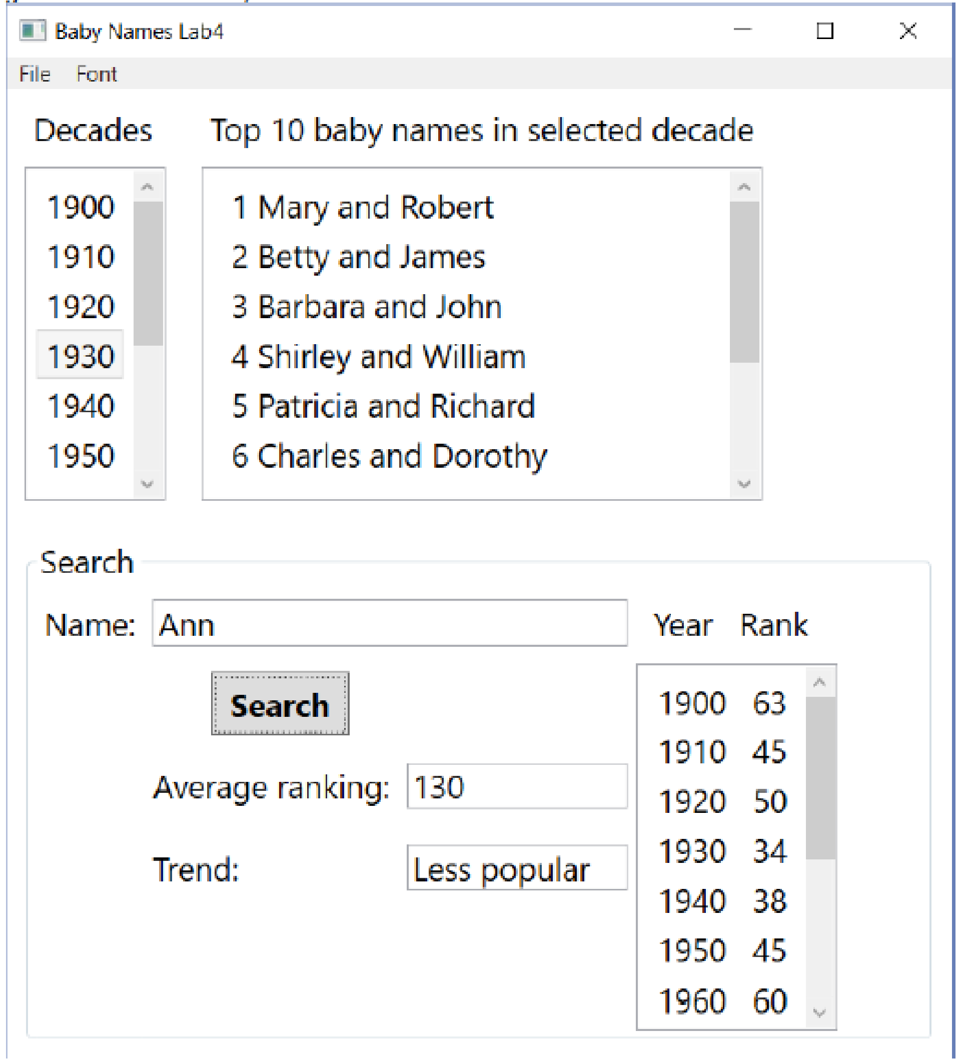


Delopgave 4

1. Tilføj en menu med følgende struktur:

File	Font
Exit	Small
	Normal
	Large
	Huge

2. Implementer de tilsvarende Click-event handlere. Exit skal lukke programmet. Font-xxx eventhandlerne skal ændre størrelsen på den anvendte font - f.eks. small sætter fontsize til 8 og Large sætter fontsize til 18 osv.
3. Opdater din anvendelse af layout paneler mv., således at din brugergrænseflade ser pæn og funktionel ud uanset størrelsen på den valgte font (inden for rimelige grænser så som 8 - 40).



Decades

1900
1910
1920
1930
1940
1950

Top 10 baby names in selected decade

1 Mary and Robert
2 Betty and James
3 Barbara and John
4 Shirley and William
5 Patricia and Richard
6 Charles and Dorothy

Search

Name: Ann

Average ranking: 130

Trend: Less popular

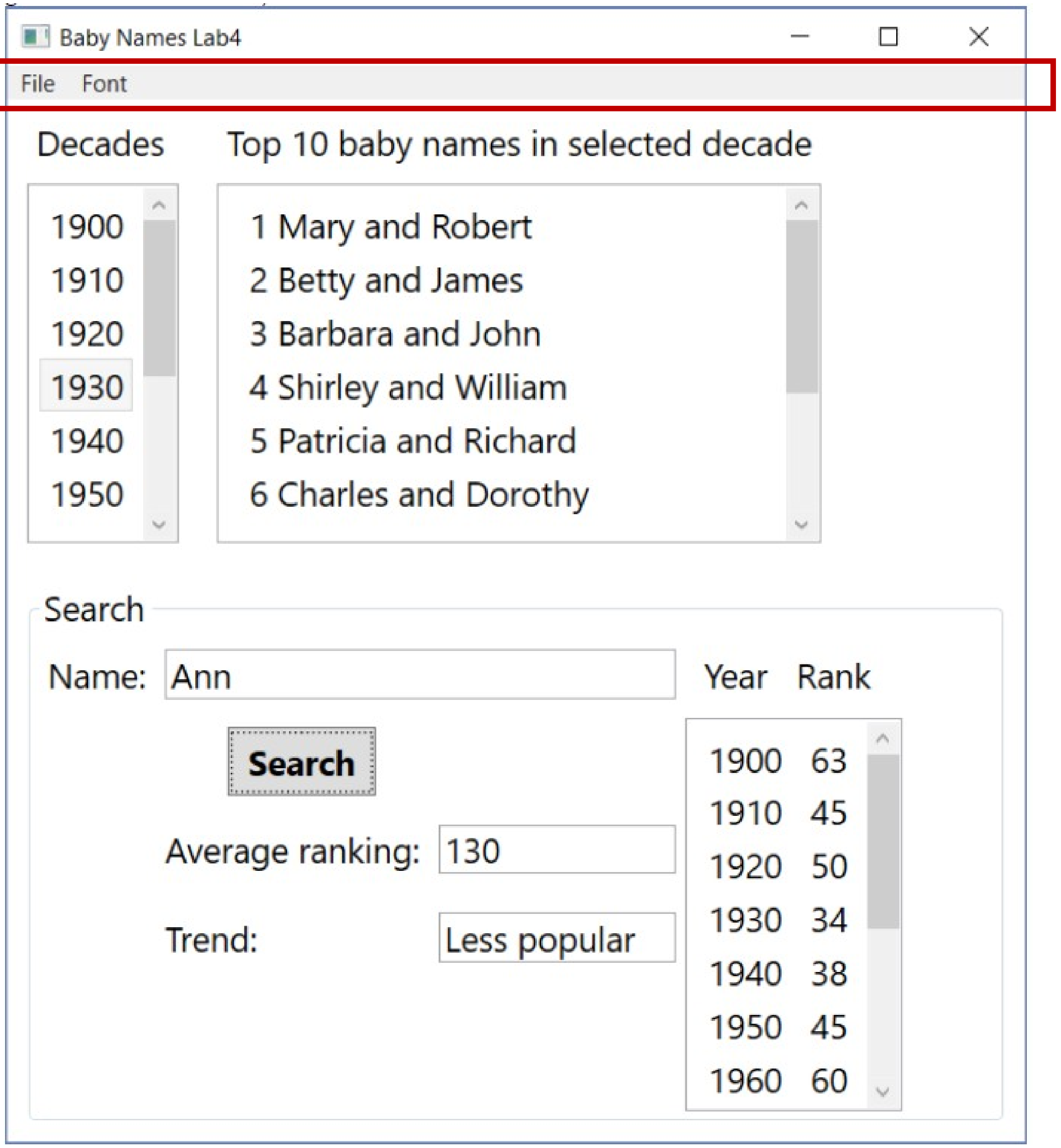
Year	Rank
1900	63
1910	45
1920	50
1930	34
1940	38
1950	45
1960	60

- Use menu in xaml

```
<Menu Margin="0,0,0,407">
   <MenuItem Header="_File">
       <MenuItem Header="E_xit" Click="MI_FileExitClick" />
   </MenuItem>
   <MenuItem Header="F_ont">
       <MenuItem Header="_Small" Click="MI_FontSmall"/>
       <MenuItem Header="_Normal" Click="MI_FontNormal" />
       <MenuItem Header="_Large" Click="MI_FontLarge"/>
       <MenuItem Header="_Huge" Click="MI_FontHuge"/>
   </MenuItem>
</Menu>
```

Clikc Events and event handler names
Put curson on each name and F12 (or Go to definition)
Each event handler will be generated in C#.

```
private void MI_FileExitClick(object sender, RoutedEventArgs
e)
{

}

private void MI_FontSmall(object sender, RoutedEventArgs e)
{

}
…
```



Decades
1900
1910
1920
1930
1940
1950

Top 10 baby names in selected decade:
1 Mary and Robert
2 Betty and James
3 Barbara and John
4 Shirley and William
5 Patricia and Richard
6 Charles and Dorothy

Name: Ann
Average ranking: 130
Trend: Less popular

Year	Rank
1900	63
1910	45
1920	50
1930	34
1940	38
1950	45
1960	60