

Microsoft.Premium.70-357.by.VCEplus.51q

Number: 70-357 VCEplus
Passing Score: 800
Time Limit: 120 min
File Version: 1.0



Exam Code: 70-357

Exam Name: Developing Mobile Apps

Certification Provider: Microsoft

Corresponding Certification: MCSA: Universal Windows Platform

Free Exam: <https://vceplus.com/exam-70-357/>

Questions & Answers Exam Engine is rigorously checked before being put up for sale. We make sure there is nothing irrelevant in 70-357 exam products and you get latest questions. We strive to deliver the best 70-357 exam product for top grades in your first attempt.

VCE to PDF Converter : <https://vceplus.com/vce-to-pdf/>

Facebook: <https://www.facebook.com/VCE.For.All.VN/>

Twitter : https://twitter.com/VCE_Plus

Google+ : <https://plus.google.com/+Vcepluscom>

LinkedIn : <https://www.linkedin.com/company/vceplus>

Exam A

QUESTION 1

Case Study

This is a case study. Case studies are not limited separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are developing an application named Timeline that presents information on a timeline. The app allows users to create items and enter details about the item. The app displays item names on a timeline. When users select an item name on the timeline, the app displays additional content about the item.

Business requirements

Timeline section

The timeline element of the app has the following layout requirements:

- The timeline must adapt to the screen size and orientation of the device.
- The timeline size must dynamically change if the window containing the content is resized by the user.
- The user must be able to scroll through the timeline horizontally when the device is in landscape mode.
- The user must be able to scroll through the timeline vertically when the device is in portrait mode.
- The timeline must begin scrolling as soon as a scroll is detected. Scrolling must continue for a short distance after the scroll input has stopped.
- Scroll bars or panning controls must always be visible.

The following image depicts the layout for the timeline section of the app when the device is using landscape orientation:



The following image depicts the layout for the timeline section of the app when the device is using portrait orientation:



Content section

The content element of the app has the following layout requirements:

- When a user selects an item on the timeline, the details for that item must display beneath or to the right of the timeline.

- The content section must display one page of information. The element must be a child of the selected item in the timeline.
- Users must be able to return to a previously selected event by pressing the Back button.

User interface

The user must be able to navigate the application using the interface below:



- The Favorite button marks the current content to be displayed in a Favorites panel.
- The Back and Forward buttons navigate through the app selection history. Both buttons must be available on all devices.
- The Notes button allows the user to manage notes about the current content.
- The app must support touch, mouse, and stylus input.
- The app layout must automatically adapt to the screen size and orientation.

Technical requirements Layout

You identify the following layout requirements:

General

- All user interface (UI) elements must continuously scale when a user resizes the window.
- UI controls must be smaller and spaced closer together if there is a mouse or stylus available.
- UI controls must be larger and spaced farther apart if the device supports touch and there is no mouse or pointer available.

Timeline

- The timeline must be displayed in a horizontal layout when the device is in a landscape orientation or when the horizontal width is greater than the vertical height.
- The timeline must be displayed in a vertical layout when the device is in a portrait orientation or when the vertical height is greater than the horizontal width.
- Each item in the past must be linked to the next item in the future.
- Users must be able to scroll from past events to future events or from future events to past events.
- The app must only allow one level of detail to be linked to each item in the timeline.

Optimization

You must optimize the app using the following guidelines:

- You must minimize the time it takes to display content when an item on the timeline is selected.
- The app must respect memory and resource constraints for all devices.

XAML coding style

All code and markup must conform to the following style guidelines:

- Use resource dictionaries for styles that are used more than once.
- Limit the use of nested panels.
- Use built-in properties of existing panels instead of using separate style objects,
- Use the navigation structure that best models the data without exceeding the requirements of the app.

Application structure MainPage.xaml

Relevant portions of the app files are shown below. {Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.}

```

MP01 <Page
      x: Class = "_70357rm.MainPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:local="using:_70357rm"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:m="http://schemas.openxmlformats.org/markup-compatibility/2006"
      mc: Ignorable="d">
MP02     <Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
MP03
MP04         <RelativePanel BorderBrush="Gray" BorderThickness="10">
MP05             <Rectangle x:Name="A1" Fill="Red" MinHeight="200" MinWidth="400"
                  RelativePanel.AlignLeftWithPanel="True"
                  RelativePanel.AlignTopWithPanel="False" />
MP06             <Rectangle x:Name="B1" Fill="Blue" MinHeight="200" MinWidth="400"
                  RelativePanel.Below="A1"
                  RelativePanel.RightOf="A1"
                  RelativePanel.AlignRightWithPanel="True"
                  RelativePanel.AlignBottomWithPanel="False" />
MP07         </ RelativePanel>
MP08     </ Grid>
MP09 </ Page>
  
```

Settings.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

AS01 <Page
  x: Class ="_70357rm.Settings"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x ="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns: local ="using:_70357rm"
  xmlns: d ="http://schemas.microsoft.com/expression/blend/2008"
  xmlns: m ="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc: Ignorable ="d">
AS02   <Grid Background ="AliceBlue">
AS03     <Border BorderBrush ="DarkBlue" BorderThickness ="3" />
AS04     <Grid Margin ="5 5 5 5">
AS05       <StackPanel HorizontalAlignment ="Center">
AS06         <TextBlock Text ="Date Settings" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center" />
AS07         <StackPanel Orientation ="Horizontal">
AS08           <CheckBox Content ="Center on Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS09           <CheckBox Content ="Set Start Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS10         </StackPanel>
AS11         <TextBlock Text ="Start Date" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center"/>
AS12       <StackPanel Orientation ="Horizontal">
AS13         <TextBlock Text ="Month:" Width ="75" />
AS14         <TextBox Width ="200" />
AS15       </StackPanel>
AS16       <StackPanel Orientation ="Horizontal">
AS17         <TextBlock Text ="Day:" Width ="75" />
AS18         <TextBox Width ="200" />
AS19       </StackPanel>
AS20       <StackPanel Orientation ="Horizontal">
AS21         <TextBlock Text ="Year:" Width ="75" />
AS22         <TextBlock Width ="200" />
AS23       </StackPanel>
AS24       <Ellipse Fill ="Blue" Width ="50" Height="50" Margin ="0 5 0 5"/>
AS25       <TextBlock FontFamily ="Arial" FontSize ="14" Foreground="White" Text ="Save"
Margin ="127 -38 0 0"/>
AS26     </StackPanel>
AS27   </Grid>
AS28 </Grid>
AS29 </Page>
AS30 </Page>

```

ResourceDictionary.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only.)

```
01 <ResourceDictionary
02     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
03     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
04     xmlns:local="using:_70357rm">
05     <Style x:Key="fill">
06         <Setter Property="Foreground" Value="DarkBlue"/>
07     </Style>
08     <Style x:Key="text">
09         <Setter Property="FontFamily" Value="Arial"/>
10     </Style>
11     <Style x:Key="big">
12         <Setter Property="FontSize" Value="20"/>
13     </Style>
14     <Style x:Key="small">
15         <Setter Property="FontSize" Value="14"/>
16     </Style>
17     <Style x:Key="strong">
18         <Setter Property="FontWeight" Value="Bold"/>
19     </Style>
20     <Style x:Key="light">
21         <Setter Property="FontWeight" Value="Normal"/>
22     </Style>
23     <Style x:Key="normal">
```



```

24     <Setter Property="FontStyle" Value="Normal"/>
25 </Style>
26 <Style x:Key="pad">
27     <Setter Property="Margin" Value="0 5 0 5"/>
28 </Style>
29 <Style x:Key="gap">
30     <Setter Property="Margin" Value="20 0 0 0"/>
31 </Style>
32 <Style x:Key="middle">
33     <Setter Property="HorizontalAlignment" Value="Center"/>
34 </Style>
35 <Style TargetType="CheckBox" x:Key="check">
36     <Setter Property="FontFamily" Value="Arial" />
37     <Setter Property="FontSize" Value="14" />
38     <Setter Property="FontStyle" Value="Normal" />
39     <Setter Property="Margin" Value="20 0 0 0" />
40 </Style>
41 <Style TargetType="TextBox" x:Key="heading">
42     <Setter Property="Foreground" Value="DarkBlue" />
43     <Setter Property="FontFamily" Value="Arial" />
44     <Setter Property="FontSize" Value="20" />
45     <Setter Property="FontStyle" Value="Normal" />
46     <Setter Property="FontWeight" Value="Bold" />
47     <Setter Property="Margin" Value="0 5 0 5" />
48     <Setter Property="HorizontalAlignment" Value="Center" />
49 </Style>
50 </ResourceDictionary>

```

MainPage.xaml.es

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

MX01 private void App_BackRequest(object sender, Windows.UI.Core.BackRequestedEventArgs e)
MX02 {
MX03     Frame page = Window.Current.Content as Frame;
MX04     if ( page != null)
MX05     {
MX06         if ( page.CanGoBack )
MX07         {
MX08             page.GoBack();
MX09         }
MX10     }
MX11 }

```

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You need to implement the appropriate XAML layout for the Timeline app.

Solution: You create an instance of a StackPanel class,

Does this meet the goal?

A. Yes

B. No

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

Explanation:

StackPanel is a simple layout panel that arranges its child elements into a single line that can be oriented horizontally or vertically. StackPanel controls are typically used in scenarios where you want to arrange a small subsection of the UI on your page.

The following XAML shows how to create a vertical StackPanel of items.

XAML

```
<StackPanel>
```

```
<Rectangle Fill="Red" Height="44"/>
```

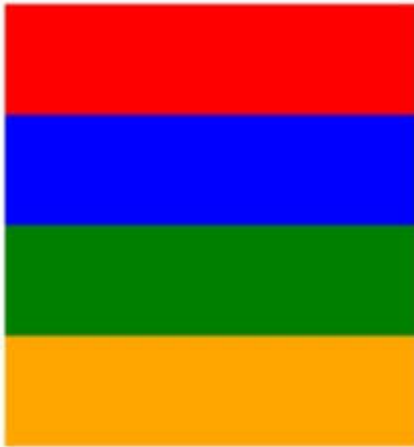
```
<Rectangle Fill="Blue" Height="44"/>
```

```
<Rectangle Fill="Green" Height="44"/>
```

```
<Rectangle Fill="Orange" Height="44"/>
```

```
</StackPanel>
```

The result looks like this.



Reference: <https://docs.microsoft.com/en-us/windows/uwp/layout/layout-panels>

QUESTION 2

Case Study

This is a case study. Case studies are not limited separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are developing an application named Timeline that presents information on a timeline. The app allows users to create items and enter details about the item. The app displays item names on a timeline. When users select an item name on the timeline, the app displays additional content about the item.

Business requirements

Timeline section

The timeline element of the app has the following layout requirements:

- The timeline must adapt to the screen size and orientation of the device.
- The timeline size must dynamically change if the window containing the content is resized by the user.
- The user must be able to scroll through the timeline horizontally when the device is in landscape mode.

- The user must be able to scroll through the timeline vertically when the device is in portrait mode.
- The timeline must begin scrolling as soon as a scroll is detected. Scrolling must continue for a short distance after the scroll input has stopped.
- Scroll bars or panning controls must always be visible.

The following image depicts the layout for the timeline section of the app when the device is using landscape orientation:



The following image depicts the layout for the timeline section of the app when the device is using portrait orientation:



Content section

The content element of the app has the following layout requirements:

- When a user selects an item on the timeline, the details for that item must display beneath or to the right of the timeline.
- The content section must display one page of information. The element must be a child of the selected item in the timeline.
- Users must be able to return to a previously selected event by pressing the Back button.

User interface

The user must be able to navigate the application using the interface below:



- The Favorite button marks the current content to be displayed in a Favorites panel.
- The Back and Forward buttons navigate through the app selection history. Both buttons must be available on all devices.
- The Notes button allows the user to manage notes about the current content.
- The app must support touch, mouse, and stylus input.
- The app layout must automatically adapt to the screen size and orientation.

Technical requirements Layout

You identify the following layout requirements:

General

- All user interface (UI) elements must continuously scale when a user resizes the window.
- UI controls must be smaller and spaced closer together if there is a mouse or stylus available.
- UI controls must be larger and spaced farther apart if the device supports touch and there is no mouse or pointer available.

Timeline

- The timeline must be displayed in a horizontal layout when the device is in a landscape orientation or when the horizontal width is greater than the vertical height.
- The timeline must be displayed in a vertical layout when the device is in a portrait orientation or when the vertical height is greater than the horizontal width.
- Each item in the past must be linked to the next item in the future.
- Users must be able to scroll from past events to future events or from future events to past events.
- The app must only allow one level of detail to be linked to each item in the timeline.

Optimization

You must optimize the app using the following guidelines:

- You must minimize the time it takes to display content when an item on the timeline is selected.
- The app must respect memory and resource constraints for all devices.

XAML coding style

All code and markup must conform to the following style guidelines:

- Use resource dictionaries for styles that are used more than once.
- Limit the use of nested panels.
- Use built-in properties of existing panels instead of using separate style objects,
- Use the navigation structure that best models the data without exceeding the requirements of the app.

Application structure MainPage.xaml

Relevant portions of the app files are shown below. {Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.}

```

MP01 <Page
      x: Class = "_70357rm.MainPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:local="using:_70357rm"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:m="http://schemas.openxmlformats.org/markup-compatibility/2006"
      mc:Ignorable="d">
MP02     <Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
MP03
MP04         <RelativePanel BorderBrush="Gray" BorderThickness="1">
MP05             <Rectangle x:Name="A1" Fill="Red" MinHeight="200" MinWidth="400"
                  RelativePanel.AlignLeftWithPanel="True"
                  RelativePanel.AlignTopWithPanel="False" />
MP06             <Rectangle x:Name="B1" Fill="Blue" MinHeight="200" MinWidth="400"
                  RelativePanel.Below="A1"
                  RelativePanel.RightOf="A1"
                  RelativePanel.AlignRightWithPanel="True"
                  RelativePanel.AlignBottomWithPanel="False" />
MP07         </RelativePanel>
MP08     </Grid>
MP09 </Page>
  
```

Settings.xaml

Relevant portions of the app files are shown below. {Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.}

```

AS01 <Page
  x: Class ="_70357rm.Settings"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x ="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns: local ="using:_70357rm"
  xmlns: d ="http://schemas.microsoft.com/expression/blend/2008"
  xmlns: m ="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc: Ignorable ="d">
AS02   <Grid Background ="AliceBlue">
AS03     <Border BorderBrush ="DarkBlue" BorderThickness ="3" />
AS04     <Grid Margin ="5 5 5 5">
AS05       <StackPanel HorizontalAlignment ="Center">
AS06         <TextBlock Text ="Date Settings" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center" />
AS07         <StackPanel Orientation ="Horizontal">
AS08           <CheckBox Content ="Center on Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS09           <CheckBox Content ="Set Start Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS10         </StackPanel>
AS11         <TextBlock Text ="Start Date" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center"/>
AS12       <StackPanel Orientation ="Horizontal">
AS13         <TextBlock Text ="Month:" Width ="75" />
AS14         <TextBox Width ="200" />
AS15       </StackPanel>
AS16       <StackPanel Orientation ="Horizontal">
AS17         <TextBlock Text ="Day:" Width ="75" />
AS18         <TextBox Width ="200" />
AS19       </StackPanel>
AS20       <StackPanel Orientation ="Horizontal">
AS21         <TextBlock Text ="Year:" Width ="75" />
AS22         <TextBlock Width ="200" />
AS23       </StackPanel>
AS24       <Ellipse Fill ="Blue" Width ="50" Height="50" Margin ="0 5 0 5"/>
AS25       <TextBlock FontFamily ="Arial" FontSize ="14" Foreground="White" Text ="Save"
Margin ="127 -38 0 0"/>
AS26     </StackPanel>
AS27   </Grid>
AS28 </Grid>
AS29 </Page>
AS30

```

ResourceDictionary.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only.)

```
01 <ResourceDictionary
02     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
03     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
04     xmlns:local="using:_70357rm">
05     <Style x:Key="fill">
06         <Setter Property="Foreground" Value="DarkBlue"/>
07     </Style>
08     <Style x:Key="text">
09         <Setter Property="FontFamily" Value="Arial"/>
10     </Style>
11     <Style x:Key="big">
12         <Setter Property="FontSize" Value="20"/>
13     </Style>
14     <Style x:Key="small">
15         <Setter Property="FontSize" Value="14"/>
16     </Style>
17     <Style x:Key="strong">
18         <Setter Property="FontWeight" Value="Bold"/>
19     </Style>
20     <Style x:Key="light">
21         <Setter Property="FontWeight" Value="Normal"/>
22     </Style>
23     <Style x:Key="normal">
```



```

24     <Setter Property="FontStyle" Value="Normal"/>
25 </Style>
26 <Style x:Key="pad">
27     <Setter Property="Margin" Value="0 5 0 5"/>
28 </Style>
29 <Style x:Key="gap">
30     <Setter Property="Margin" Value="20 0 0 0"/>
31 </Style>
32 <Style x:Key="middle">
33     <Setter Property="HorizontalAlignment" Value="Center"/>
34 </Style>
35 <Style TargetType="CheckBox" x:Key="check">
36     <Setter Property="FontFamily" Value="Arial" />
37     <Setter Property="FontSize" Value="14" />
38     <Setter Property="FontStyle" Value="Normal" />
39     <Setter Property="Margin" Value="20 0 0 0" />
40 </Style>
41 <Style TargetType="TextBox" x:Key="heading">
42     <Setter Property="Foreground" Value="DarkBlue" />
43     <Setter Property="FontFamily" Value="Arial" />
44     <Setter Property="FontSize" Value="20" />
45     <Setter Property="FontStyle" Value="Normal" />
46     <Setter Property="FontWeight" Value="Bold" />
47     <Setter Property="Margin" Value="0 5 0 5" />
48     <Setter Property="HorizontalAlignment" Value="Center" />
49 </Style>
50 </ResourceDictionary>

```

MainPage.xaml.es

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

MX01 private void App_BackRequest(object sender, Windows.UI.Core.BackRequestedEventArgs e)
MX02 {
MX03     Frame page = Window.Current.Content as Frame;
MX04     if ( page != null)
MX05     {
MX06         if ( page.CanGoBack )
MX07         {
MX08             page.GoBack();
MX09         }
MX10     }
MX11 }

```


Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You need to implement the appropriate XAML layout for the Timeline app.

Solution: You create an instance of a RelativePanel class.

Does this meet the goal?

A. Yes

B. No

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Explanation:

RelativePanel lets you layout UI elements by specifying where they go in relation to other elements and in relation to the panel. By default, an element is positioned in the upper left corner of the panel.

Reference: <https://docs.microsoft.com/en-us/windows/uwp/layout/layout-panels>

QUESTION 3

Case Study

This is a case study. Case studies are not limited separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are developing an application named Timeline that presents information on a timeline. The app allows users to create items and enter details about the item.

The app displays item names on a timeline. When users select an item name on the timeline, the app displays additional content about the item.

Business requirements

Timeline section

The timeline element of the app has the following layout requirements:

- The timeline must adapt to the screen size and orientation of the device.
- The timeline size must dynamically change if the window containing the content is resized by the user.

- The user must be able to scroll through the timeline horizontally when the device is in landscape mode.
- The user must be able to scroll through the timeline vertically when the device is in portrait mode.
- The timeline must begin scrolling as soon as a scroll is detected. Scrolling must continue for a short distance after the scroll input has stopped.
- Scroll bars or panning controls must always be visible.

The following image depicts the layout for the timeline section of the app when the device is using landscape orientation:



The following image depicts the layout for the timeline section of the app when the device is using portrait orientation:



Content section

The content element of the app has the following layout requirements:

- When a user selects an item on the timeline, the details for that item must display beneath or to the right of the timeline.
- The content section must display one page of information. The element must be a child of the selected item in the timeline.
- Users must be able to return to a previously selected event by pressing the Back button.

User interface

The user must be able to navigate the application using the interface below:



- The Favorite button marks the current content to be displayed in a Favorites panel.
- The Back and Forward buttons navigate through the app selection history. Both buttons must be available on all devices.
- The Notes button allows the user to manage notes about the current content.
- The app must support touch, mouse, and stylus input.
- The app layout must automatically adapt to the screen size and orientation.

Technical requirements Layout

You identify the following layout requirements:

General

- All user interface (UI) elements must continuously scale when a user resizes the window.
- UI controls must be smaller and spaced closer together if there is a mouse or stylus available.
- UI controls must be larger and spaced farther apart if the device supports touch and there is no mouse or pointer available.

Timeline

- The timeline must be displayed in a horizontal layout when the device is in a landscape orientation or when the horizontal width is greater than the vertical height.
- The timeline must be displayed in a vertical layout when the device is in a portrait orientation or when the vertical height is greater than the horizontal width.
- Each item in the past must be linked to the next item in the future.
- Users must be able to scroll from past events to future events or from future events to past events.
- The app must only allow one level of detail to be linked to each item in the timeline.

Optimization

You must optimize the app using the following guidelines:

- You must minimize the time it takes to display content when an item on the timeline is selected.
- The app must respect memory and resource constraints for all devices.

XAML coding style

All code and markup must conform to the following style guidelines:

- Use resource dictionaries for styles that are used more than once.
- Limit the use of nested panels.
- Use built-in properties of existing panels instead of using separate style objects,
- Use the navigation structure that best models the data without exceeding the requirements of the app.

Application structure MainPage.xaml

Relevant portions of the app files are shown below. {Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.}

```

MP01 <Page
      x: Class = "_70357rm.MainPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:local="using:_70357rm"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:m="http://schemas.openxmlformats.org/markup-compatibility/2006"
      mc:Ignorable="d">
MP02     <Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
MP03
MP04         <RelativePanel BorderBrush="Gray" BorderThickness="10">
MP05             <Rectangle x:Name="A1" Fill="Red" MinHeight="200" MinWidth="400"
                  RelativePanel.AlignLeftWithPanel="True"
                  RelativePanel.AlignTopWithPanel="False" />
MP06             <Rectangle x:Name="B1" Fill="Blue" MinHeight="200" MinWidth="400"
                  RelativePanel.Below="A1"
                  RelativePanel.RightOf="A1"
                  RelativePanel.AlignRightWithPanel="True"
                  RelativePanel.AlignBottomWithPanel="False" />
MP07         </RelativePanel>
MP08     </Grid>
MP09 </Page>
  
```

Settings.xaml

Relevant portions of the app files are shown below. {Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.}

```

AS01 <Page
  x: Class ="_70357rm.Settings"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x ="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns: local ="using:_70357rm"
  xmlns: d ="http://schemas.microsoft.com/expression/blend/2008"
  xmlns: m ="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc: Ignorable ="d">
AS02   <Grid Background ="AliceBlue">
AS03     <Border BorderBrush ="DarkBlue" BorderThickness ="3" />
AS04     <Grid Margin ="5 5 5 5">
AS05       <StackPanel HorizontalAlignment ="Center">
AS06         <TextBlock Text ="Date Settings" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center" />
AS07         <StackPanel Orientation ="Horizontal">
AS08           <CheckBox Content ="Center on Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS09           <CheckBox Content ="Set Start Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS10         </StackPanel>
AS11         <TextBlock Text ="Start Date" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center"/>
AS12       <StackPanel Orientation ="Horizontal">
AS13         <TextBlock Text ="Month:" Width ="75" />
AS14         <TextBox Width ="200" />
AS15       </StackPanel>
AS16       <StackPanel Orientation ="Horizontal">
AS17         <TextBlock Text ="Day:" Width ="75" />
AS18         <TextBox Width ="200" />
AS19       </StackPanel>
AS20       <StackPanel Orientation ="Horizontal">
AS21         <TextBlock Text ="Year:" Width ="75" />
AS22         <TextBlock Width ="200" />
AS23       </StackPanel>
AS24       <Ellipse Fill ="Blue" Width ="50" Height="50" Margin ="0 5 0 5"/>
AS25       <TextBlock FontFamily ="Arial" FontSize ="14" Foreground="White" Text ="Save"
Margin ="127 -38 0 0"/>
AS26     </StackPanel>
AS27   </Grid>
AS28 </Grid>
AS29 </Page>
AS30 </Page>

```

ResourceDictionary.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only.)

```
01 <ResourceDictionary
02     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
03     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
04     xmlns:local="using:_70357rm">
05     <Style x:Key="fill">
06         <Setter Property="Foreground" Value="DarkBlue"/>
07     </Style>
08     <Style x:Key="text">
09         <Setter Property="FontFamily" Value="Arial"/>
10     </Style>
11     <Style x:Key="big">
12         <Setter Property="FontSize" Value="20"/>
13     </Style>
14     <Style x:Key="small">
15         <Setter Property="FontSize" Value="14"/>
16     </Style>
17     <Style x:Key="strong">
18         <Setter Property="FontWeight" Value="Bold"/>
19     </Style>
20     <Style x:Key="light">
21         <Setter Property="FontWeight" Value="Normal"/>
22     </Style>
23     <Style x:Key="normal">
```



```

24     <Setter Property="FontStyle" Value="Normal"/>
25 </Style>
26 <Style x:Key="pad">
27     <Setter Property="Margin" Value="0 5 0 5"/>
28 </Style>
29 <Style x:Key="gap">
30     <Setter Property="Margin" Value="20 0 0 0"/>
31 </Style>
32 <Style x:Key="middle">
33     <Setter Property="HorizontalAlignment" Value="Center"/>
34 </Style>
35 <Style TargetType="CheckBox" x:Key="check">
36     <Setter Property="FontFamily" Value="Arial" />
37     <Setter Property="FontSize" Value="14" />
38     <Setter Property="FontStyle" Value="Normal" />
39     <Setter Property="Margin" Value="20 0 0 0" />
40 </Style>
41 <Style TargetType="TextBox" x:Key="heading">
42     <Setter Property="Foreground" Value="DarkBlue" />
43     <Setter Property="FontFamily" Value="Arial" />
44     <Setter Property="FontSize" Value="20" />
45     <Setter Property="FontStyle" Value="Normal" />
46     <Setter Property="FontWeight" Value="Bold" />
47     <Setter Property="Margin" Value="0 5 0 5" />
48     <Setter Property="HorizontalAlignment" Value="Center" />
49 </Style>
50 </ResourceDictionary>

```

MainPage.xaml.es

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

MX01 private void App_BackRequest(object sender, Windows.UI.Core.BackRequestedEventArgs e)
MX02 {
MX03     Frame page = Window.Current.Content as Frame;
MX04     if ( page != null)
MX05     {
MX06         if ( page.CanGoBack )
MX07         {
MX08             page.GoBack();
MX09         }
MX10     }
MX11 }

```


Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You need to implement the appropriate XAML layout for the Timeline app.

Solution: You create an instance of a SplitView control.

Does this meet the goal?

A. Yes

B. No

Correct Answer: B

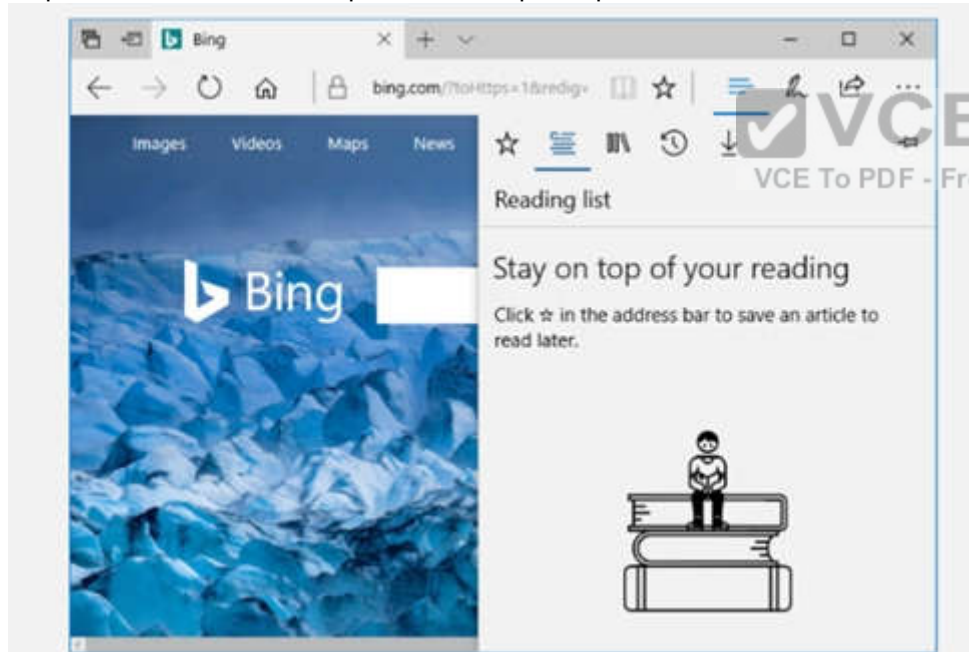
Section: (none)

Explanation

Explanation/Reference:

Explanation:

A split view control has an expandable/collapsible pane and a content area. Here is an example of the Microsoft Edge app using SplitView to show its Hub.



Reference: <https://docs.microsoft.com/en-us/windows/uwp/controls-and-patterns/split-view>

QUESTION 4

DRAG DROP

Case Study

This is a case study. Case studies are not limited separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are developing an application named Timeline that presents information on a timeline. The app allows users to create items and enter details about the item. The app displays item names on a timeline. When users select an item name on the timeline, the app displays additional content about the item.

Business requirements

Timeline section

The timeline element of the app has the following layout requirements:

- The timeline must adapt to the screen size and orientation of the device.
- The timeline size must dynamically change if the window containing the content is resized by the user.
- The user must be able to scroll through the timeline horizontally when the device is in landscape mode.
- The user must be able to scroll through the timeline vertically when the device is in portrait mode.
- The timeline must begin scrolling as soon as a scroll is detected. Scrolling must continue for a short distance after the scroll input has stopped.
- Scroll bars or panning controls must always be visible.

The following image depicts the layout for the timeline section of the app when the device is using landscape orientation:



The following image depicts the layout for the timeline section of the app when the device is using portrait orientation:



Content section

The content element of the app has the following layout requirements:

- When a user selects an item on the timeline, the details for that item must display beneath or to the right of the timeline.
- The content section must display one page of information. The element must be a child of the selected item in the timeline.
- Users must be able to return to a previously selected event by pressing the Back button.

User interface

The user must be able to navigate the application using the interface below:



- The Favorite button marks the current content to be displayed in a Favorites panel.
- The Back and Forward buttons navigate through the app selection history. Both buttons must be available on all devices.
- The Notes button allows the user to manage notes about the current content.
- The app must support touch, mouse, and stylus input.
- The app layout must automatically adapt to the screen size and orientation.

Technical requirements Layout

You identify the following layout requirements:

General

- All user interface (UI) elements must continuously scale when a user resizes the window.
- UI controls must be smaller and spaced closer together if there is a mouse or stylus available.
- UI controls must be larger and spaced farther apart if the device supports touch and there is no mouse or pointer available.

Timeline

- The timeline must be displayed in a horizontal layout when the device is in a landscape orientation or when the horizontal width is greater than the vertical height.
- The timeline must be displayed in a vertical layout when the device is in a portrait orientation or when the vertical height is greater than the horizontal width.
- Each item in the past must be linked to the next item in the future.
- Users must be able to scroll from past events to future events or from future events to past events.
- The app must only allow one level of detail to be linked to each item in the timeline.

Optimization

You must optimize the app using the following guidelines:

- You must minimize the time it takes to display content when an item on the timeline is selected.
- The app must respect memory and resource constraints for all devices.

XAML coding style

All code and markup must conform to the following style guidelines:

- Use resource dictionaries for styles that are used more than once.
- Limit the use of nested panels.
- Use built-in properties of existing panels instead of using separate style objects,
- Use the navigation structure that best models the data without exceeding the requirements of the app.

Application structure MainPage.xaml

Relevant portions of the app files are shown below. {Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.}

```

MP01 <Page
      x: Class = "_70357rm.MainPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:local="using:_70357rm"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:m="http://schemas.openxmlformats.org/markup-compatibility/2006"
      mc: Ignorable="d">
MP02     <Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
MP03
MP04         <RelativePanel BorderBrush="Gray" BorderThickness="10">
MP05             <Rectangle x:Name="A1" Fill="Red" MinHeight="200" MinWidth="400"
                  RelativePanel.AlignLeftWithPanel="True"
                  RelativePanel.AlignTopWithPanel="False" />
MP06             <Rectangle x:Name="B1" Fill="Blue" MinHeight="200" MinWidth="400"
                  RelativePanel.Below="A1"
                  RelativePanel.RightOf="A1"
                  RelativePanel.AlignRightWithPanel="True"
                  RelativePanel.AlignBottomWithPanel="False" />
MP07         </ RelativePanel>
MP08     </ Grid>
MP09 </ Page>

```

Settings.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

AS01 <Page
  x: Class ="_70357rm.Settings"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x ="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns: local ="using:_70357rm"
  xmlns: d ="http://schemas.microsoft.com/expression/blend/2008"
  xmlns: m ="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc: Ignorable ="d">
AS02   <Grid Background ="AliceBlue">
AS03     <Border BorderBrush ="DarkBlue" BorderThickness ="3" />
AS04     <Grid Margin ="5 5 5 5">
AS05       <StackPanel HorizontalAlignment ="Center">
AS06         <TextBlock Text ="Date Settings" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center" />
AS07         <StackPanel Orientation ="Horizontal">
AS08           <CheckBox Content ="Center on Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS09           <CheckBox Content ="Set Start Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS10         </StackPanel>
AS11         <TextBlock Text ="Start Date" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center"/>
AS12       <StackPanel Orientation ="Horizontal">
AS13         <TextBlock Text ="Month:" Width ="75" />
AS14         <TextBox Width ="200" />
AS15       </StackPanel>
AS16       <StackPanel Orientation ="Horizontal">
AS17         <TextBlock Text ="Day:" Width ="75" />
AS18         <TextBox Width ="200" />
AS19       </StackPanel>
AS20       <StackPanel Orientation ="Horizontal">
AS21         <TextBlock Text ="Year:" Width ="75" />
AS22         <TextBox Width ="200" />
AS23       </StackPanel>
AS24       <Ellipse Fill ="Blue" Width ="50" Height="50" Margin ="0 5 0 5"/>
AS25       <TextBlock FontFamily ="Arial" FontSize ="14" Foreground="White" Text ="Save"
Margin ="127 -38 0 0"/>
AS26     </StackPanel>
AS27   </Grid>
AS28 </Grid>
AS29 </Page>
AS30

```

ResourceDictionary.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only.)


```
01 <ResourceDictionary
02     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
03     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
04     xmlns:local="using:_70357rm">
05     <Style x:Key="fill">
06         <Setter Property="Foreground" Value="DarkBlue"/>
07     </Style>
08     <Style x:Key="text">
09         <Setter Property="FontFamily" Value="Arial"/>
10     </Style>
11     <Style x:Key="big">
12         <Setter Property="FontSize" Value="20"/>
13     </Style>
14     <Style x:Key="small">
15         <Setter Property="FontSize" Value="14"/>
16     </Style>
17     <Style x:Key="strong">
18         <Setter Property="FontWeight" Value="Bold"/>
19     </Style>
20     <Style x:Key="light">
21         <Setter Property="FontWeight" Value="Normal"/>
22     </Style>
23     <Style x:Key="normal">
```



```

24     <Setter Property="FontStyle" Value="Normal"/>
25 </Style>
26 <Style x:Key="pad">
27     <Setter Property="Margin" Value="0 5 0 5"/>
28 </Style>
29 <Style x:Key="gap">
30     <Setter Property="Margin" Value="20 0 0 0"/>
31 </Style>
32 <Style x:Key="middle">
33     <Setter Property="HorizontalAlignment" Value="Center"/>
34 </Style>
35 <Style TargetType="CheckBox" x:Key="check">
36     <Setter Property="FontFamily" Value="Arial" />
37     <Setter Property="FontSize" Value="14" />
38     <Setter Property="FontStyle" Value="Normal" />
39     <Setter Property="Margin" Value="20 0 0 0" />
40 </Style>
41 <Style TargetType="TextBox" x:Key="heading">
42     <Setter Property="Foreground" Value="DarkBlue" />
43     <Setter Property="FontFamily" Value="Arial" />
44     <Setter Property="FontSize" Value="20" />
45     <Setter Property="FontStyle" Value="Normal" />
46     <Setter Property="FontWeight" Value="Bold" />
47     <Setter Property="Margin" Value="0 5 0 5" />
48     <Setter Property="HorizontalAlignment" Value="Center" />
49 </Style>
50 </ResourceDictionary>

```

MainPage.xaml.es

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

MX01 private void App_BackRequest(object sender, Windows.UI.Core.BackRequestedEventArgs e)
MX02 {
MX03     Frame page = Window.Current.Content as Frame;
MX04     if ( page != null)
MX05     {
MX06         if ( page.CanGoBack )
MX07         {
MX08             page.GoBack();
MX09         }
MX10     }
MX11 }

```

You need to create the user interface for the timeline.

Which four markup segments should you use to develop the solution? To answer, move the appropriate markup segments from the list of markup segments to the answer area and arrange them in the correct order,

NOTE: More than one order of answer choices is correct. You will receive credit for any of the correct orders you select.

Select and Place:

Markup segments

```
<Button Content ="Favorite"
Click ="UiButton_Click"/>
<TextBlock Text ="|" Margin =
"0 2 0 2"/>
<Button Content ="Back" Click =
"UiButton_Click"/>
<Button Content ="Forward"
Click ="UiButton_Click"/>
```

```
<AppBar>
```

```
</CommandBar>
```

```
<CommandBar>
```

```
<AppBarButton Icon ="Favorite"
Label ="Favorite"
Click="UiButon_Click"/>
<AppBarSeparator />
<AppBarButton Icon ="Back"
Label ="Back"
Click="UiButton_Click" />
<AppBarButton Icon ="Forward"
Label ="Forward"
Click="UiButton_Click"/>
```

```
<AppBar.Tag>
<Button Content ="Comment"
Click="UiButton_Click"/>
</AppBar.Tag>
</AppBar.Content>
<TextBlock Text ="Navigation"
Margin ="12,14"/>
</AppBar.Content>
```

```
<CommandBar.SecondaryCommands>
```

Correct Answer:

Markup segments

```
<Button Content = "Favorite"
Click = "UiButton_Click"/>
<TextBlock Text = "|" Margin =
"0 2 0 2"/>
<Button Content = "Back" Click =
"UiButton_Click"/>
<Button Content = "Forward"
Click = "UiButton_Click"/>
```

```
<AppBar>
```

```
<AppBar.Tag>
<Button Content = "Comment"
Click = "UiButton_Click"/>
</AppBar.Tag>
</AppBar.Content>
<TextBlock Text = "Navigation"
Margin = "12,14"/>
</AppBar.Content>
```

```
<CommandBar>
```

```
<AppBarButton Icon = "Favorite"
Label = "Favorite"
Click = "UiButton_Click"/>
<AppBarSeparator />
<AppBarButton Icon = "Back"
Label = "Back"
Click = "UiButton_Click" />
<AppBarButton Icon = "Forward"
Label = "Forward"
Click = "UiButton_Click"/>
```

```
<CommandBar.SecondaryCommands>
<AppBarButton Icon = "Comment"
Label = "Notes"
Click = "UiButton_Click"/>
</CommandBar.SecondaryCommands>
<CommandBar.Content>
<TextBlock Text = "Navigation"
Margin = "12,14"/>
</CommandBar.Content>
```

```
</CommandBar>
```

Section: (none)

Explanation

Explanation/Reference:

Explanation:

Box 1: <Commandbar>

Command bars (also called "app bars") provide users with easy access to your app's most common tasks, and can be used to show commands or options that are specific to the user's context, such as a photo selection or drawing mode. They can also be used for navigation among app pages or between app sections.

Command bars can be used with any navigation pattern.

XAML provides both the AppBar control and the CommandBar control. You should use the AppBar only when you are upgrading a Universal Windows 8 app that uses the AppBar, and need to minimize changes. For new apps in Windows 10, we recommend using the CommandBar control instead.

Box 2: <AppBarButton .. etc.

The CommandBar control is a general-purpose, flexible, light-weight control that can display both complex content, such as images or text blocks, as well as simple commands such as AppBarButton, AppBarToggleButton, and AppBarSeparator controls.

Box 3: <CommandBar.SecondaryCommands> etc.

The overflow menu is shown only when the command bar is open and the SecondaryCommands property is populated. The new dynamic overflow behavior will automatically move primary commands into the SecondaryCommands area when space is limited.

Box 4: </Commandbar>

Example: Here is a same command bar in its open state. The labels identify the main parts of the control.



This example creates the command bar shown above.

<CommandBar>

<AppBarToggleButton Icon="Shuffle" Label="Shuffle" Click="AppBarButton_Click"/> <AppBarToggleButton Icon="RepeatAH" Label="Repeat"

Click="AppBarButton_Click7"> <AppBarSeparator/>

<AppBarButton Icon="Back" Label="Back" Click="AppBarButton_Click"/> <AppBarButton Icon="Stop" Label="Stop" Click="AppBarButton_Click7"> <AppBarButton Icon="Play" Label="Play" Click="AppBarButton_Click7"> <AppBarButton Icon="Forward" Label="Forward" Click="AppBarButton_Click7">

<CommandBar.SecondaryCommands>

<AppBarButton Icon="Like" Label="Like" Click="AppBarButton_Click7">


```
<AppBarButton Icon="Dislike" Label="Dislike,r Click="AppBarButton_Click7> </CommandBar.SecondaryCommands>
<CommandBar.Content>
<TextBlock Text="Now playing..." Marginal2,147>
</CommandBar.Content>
</CommandBar>
```

Reference: <https://docs.microsoft.com/en-us/windows/uwp/controls-and-patterns/app-bars>

QUESTION 5

HOTSPOT

Case Study

This is a case study. Case studies are not limited separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are developing an application named Timeline that presents information on a timeline. The app allows users to create items and enter details about the item. The app displays item names on a timeline. When users select an item name on the timeline, the app displays additional content about the item.

Business requirements

Timeline section

The timeline element of the app has the following layout requirements:

- The timeline must adapt to the screen size and orientation of the device.
- The timeline size must dynamically change if the window containing the content is resized by the user.
- The user must be able to scroll through the timeline horizontally when the device is in landscape mode.
- The user must be able to scroll through the timeline vertically when the device is in portrait mode.
- The timeline must begin scrolling as soon as a scroll is detected. Scrolling must continue for a short distance after the scroll input has stopped.
- Scroll bars or panning controls must always be visible.

The following image depicts the layout for the timeline section of the app when the device is using landscape orientation:



The following image depicts the layout for the timeline section of the app when the device is using portrait orientation:



Content section

The content element of the app has the following layout requirements:

- When a user selects an item on the timeline, the details for that item must display beneath or to the right of the timeline.
- The content section must display one page of information. The element must be a child of the selected item in the timeline.
- Users must be able to return to a previously selected event by pressing the Back button.

User interface

The user must be able to navigate the application using the interface below:



- The Favorite button marks the current content to be displayed in a Favorites panel.
- The Back and Forward buttons navigate through the app selection history. Both buttons must be available on all devices.
- The Notes button allows the user to manage notes about the current content.
- The app must support touch, mouse, and stylus input.
- The app layout must automatically adapt to the screen size and orientation.

Technical requirements Layout

You identify the following layout requirements:

General

- All user interface (UI) elements must continuously scale when a user resizes the window.
- UI controls must be smaller and spaced closer together if there is a mouse or stylus available.
- UI controls must be larger and spaced farther apart if the device supports touch and there is no mouse or pointer available.

Timeline

- The timeline must be displayed in a horizontal layout when the device is in a landscape orientation or when the horizontal width is greater than the vertical height.
- The timeline must be displayed in a vertical layout when the device is in a portrait orientation or when the vertical height is greater than the horizontal width.
- Each item in the past must be linked to the next item in the future.
- Users must be able to scroll from past events to future events or from future events to past events.
- The app must only allow one level of detail to be linked to each item in the timeline.

Optimization

You must optimize the app using the following guidelines:

- You must minimize the time it takes to display content when an item on the timeline is selected.
- The app must respect memory and resource constraints for all devices.

XAML coding style

All code and markup must conform to the following style guidelines:

- Use resource dictionaries for styles that are used more than once.
- Limit the use of nested panels.
- Use built-in properties of existing panels instead of using separate style objects,
- Use the navigation structure that best models the data without exceeding the requirements of the app.

Application structure MainPage.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

MP01 <Page
      x: Class = "_70357rm.MainPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:local="using:_70357rm"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:m="http://schemas.openxmlformats.org/markup-compatibility/2006"
      mc:Ignorable="d">
MP02     <Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
MP03
MP04         <RelativePanel BorderBrush="Gray" BorderThickness="10">
MP05             <Rectangle x:Name="A1" Fill="Red" MinHeight="200" MinWidth="400"
                  RelativePanel.AlignLeftWithPanel="True"
                  RelativePanel.AlignTopWithPanel="False" />
MP06             <Rectangle x:Name="B1" Fill="Blue" MinHeight="200" MinWidth="400"
                  RelativePanel.Below="A1"
                  RelativePanel.RightOf="A1"
                  RelativePanel.AlignRightWithPanel="True"
                  RelativePanel.AlignBottomWithPanel="False" />
MP07         </ RelativePanel>
MP08     </ Grid>
MP09 </ Page>
  
```

Settings.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

AS01 <Page
  x: Class ="_70357rm.Settings"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x ="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns: local ="using:_70357rm"
  xmlns: d ="http://schemas.microsoft.com/expression/blend/2008"
  xmlns: m ="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc: Ignorable ="d">
AS02   <Grid Background ="AliceBlue">
AS03     <Border BorderBrush ="DarkBlue" BorderThickness ="3" />
AS04     <Grid Margin ="5 5 5 5">
AS05       <StackPanel HorizontalAlignment ="Center">
AS06         <TextBlock Text ="Date Settings" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center" />
AS07         <StackPanel Orientation ="Horizontal">
AS08           <CheckBox Content ="Center on Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS09           <CheckBox Content ="Set Start Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS10         </StackPanel>
AS11         <TextBlock Text ="Start Date" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center"/>
AS12       <StackPanel Orientation ="Horizontal">
AS13         <TextBlock Text ="Month:" Width ="75" />
AS14         <TextBox Width ="200" />
AS15       </StackPanel>
AS16       <StackPanel Orientation ="Horizontal">
AS17         <TextBlock Text ="Day:" Width ="75" />
AS18         <TextBox Width ="200" />
AS19       </StackPanel>
AS20       <StackPanel Orientation ="Horizontal">
AS21         <TextBlock Text ="Year:" Width ="75" />
AS22         <TextBlock Width ="200" />
AS23       </StackPanel>
AS24       <Ellipse Fill ="Blue" Width ="50" Height="50" Margin ="0 5 0 5"/>
AS25       <TextBlock FontFamily ="Arial" FontSize ="14" Foreground="White" Text ="Save"
Margin ="127 -38 0 0"/>
AS26     </StackPanel>
AS27   </Grid>
AS28 </Grid>
AS29 </Page>
AS30

```

ResourceDictionary.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only.)

```
01 <ResourceDictionary
02     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
03     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
04     xmlns:local="using:_70357rm">
05     <Style x:Key="fill">
06         <Setter Property="Foreground" Value="DarkBlue"/>
07     </Style>
08     <Style x:Key="text">
09         <Setter Property="FontFamily" Value="Arial"/>
10     </Style>
11     <Style x:Key="big">
12         <Setter Property="FontSize" Value="20"/>
13     </Style>
14     <Style x:Key="small">
15         <Setter Property="FontSize" Value="14"/>
16     </Style>
17     <Style x:Key="strong">
18         <Setter Property="FontWeight" Value="Bold"/>
19     </Style>
20     <Style x:Key="light">
21         <Setter Property="FontWeight" Value="Normal"/>
22     </Style>
23     <Style x:Key="normal">
```



```

24     <Setter Property="FontStyle" Value="Normal"/>
25 </Style>
26 <Style x:Key="pad">
27     <Setter Property="Margin" Value="0 5 0 5"/>
28 </Style>
29 <Style x:Key="gap">
30     <Setter Property="Margin" Value="20 0 0 0"/>
31 </Style>
32 <Style x:Key="middle">
33     <Setter Property="HorizontalAlignment" Value="Center"/>
34 </Style>
35 <Style TargetType="CheckBox" x:Key="check">
36     <Setter Property="FontFamily" Value="Arial" />
37     <Setter Property="FontSize" Value="14" />
38     <Setter Property="FontStyle" Value="Normal" />
39     <Setter Property="Margin" Value="20 0 0 0" />
40 </Style>
41 <Style TargetType="TextBox" x:Key="heading">
42     <Setter Property="Foreground" Value="DarkBlue" />
43     <Setter Property="FontFamily" Value="Arial" />
44     <Setter Property="FontSize" Value="20" />
45     <Setter Property="FontStyle" Value="Normal" />
46     <Setter Property="FontWeight" Value="Bold" />
47     <Setter Property="Margin" Value="0 5 0 5" />
48     <Setter Property="HorizontalAlignment" Value="Center" />
49 </Style>
50 </ResourceDictionary>

```

MainPage.xaml.es

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

MX01 private void App_BackRequest(object sender, Windows.UI.Core.BackRequestedEventArgs e)
MX02 {
MX03     Frame page = Window.Current.Content as Frame;
MX04     if ( page != null)
MX05     {
MX06         if ( page.CanGoBack )
MX07         {
MX08             page.GoBack();
MX09         }
MX10     }
MX11 }

```


You need to properly handle the size of the user interface objects.

How should you complete the method? To answer, select the appropriate code segment from each list in the answer area NOTE; Each correct selection is worth one point.

Hot Area:

Answer Area

```

public enum UiMode
{
    UI_LARGE,
    UI_SMALL
}

private UiMode GetUiMode()
{
    IReadOnlyList<PointerDevice> pd = PointerDevice.GetPointerDevices();
    MouseCapabilities mc = new Windows.Devices.Input.MouseCapabilities();

    int pointer = pd.Count;
    int pointer = pd.MousePresent;

    if (pointer > 0)
    if (pointer == 1)
    if (pointer == 0)

    {
        return UiMode.UI_SMALL;
    }
    else
    {
        return UiMode.UI_LARGE;
    }
}

```

Correct Answer:

Answer Area

```

public enum UiMode
{
    UI_LARGE,
    UI_SMALL,
}

private UiMode GetUiMode()
{
    IReadOnlyList<PointerDevice> pd = PointerDevice.GetPointerDevices();
    MouseCapabilities pd = new Windows.Devices.Input.MouseCapabilities();

    int pointer = pd.Count;
    int pointer = pd.MousePresent;

    if (pointer > 0)
    if (pointer == 1)
    if (pointer == 0)

    return UiMode.UI_SMALL;
}
else
{
    return UiMode.UI_LARGE;
}
}

```

Section: (none)

Explanation

Explanation/Reference:

From scenario:

UI controls must be smaller and spaced closer together if there is a mouse or stylus available.

UI controls must be larger and spaced farther apart if the device supports touch and there is no mouse or pointer available.

Box 1: MouseCapabilities pd = new Windows.Devices.Input.MouseCapabilitiesf);

The Windows.Devices.Input namespace contains the MouseCapabilities class used to retrieve the properties exposed by one or more connected mice. Just create a new MouseCapabilities object and get the properties you're interested in.

Box 2: int pointer = pd.MousePresent;

Example:

MouseCapabilities mouseCapabilities = new Windows.Devices.Input.MouseCapabilities();

MousePresent.Text = mouseCapabilities.MousePresent != 0 ? "Yes": "No";

Box 3: if {pointer == 1)

This is true if a mouse is present.

Reference: <https://docs.microsoft.com/en-us/windows/uwp/input-and-devices/identify-input-devices>

QUESTION 6

HOTSPOT

Case Study

This is a case study. Case studies are not limited separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are developing an application named Timeline that presents information on a timeline. The app allows users to create items and enter details about the item. The app displays item names on a timeline. When users select an item name on the timeline, the app displays additional content about the item.

Business requirements

Timeline section

The timeline element of the app has the following layout requirements:

- The timeline must adapt to the screen size and orientation of the device.
- The timeline size must dynamically change if the window containing the content is resized by the user.
- The user must be able to scroll through the timeline horizontally when the device is in landscape mode.
- The user must be able to scroll through the timeline vertically when the device is in portrait mode.
- The timeline must begin scrolling as soon as a scroll is detected. Scrolling must continue for a short distance after the scroll input has stopped.
- Scroll bars or panning controls must always be visible.

The following image depicts the layout for the timeline section of the app when the device is using landscape orientation:



The following image depicts the layout for the timeline section of the app when the device is using portrait orientation:



Content section

The content element of the app has the following layout requirements:

- When a user selects an item on the timeline, the details for that item must display beneath or to the right of the timeline.
- The content section must display one page of information. The element must be a child of the selected item in the timeline.
- Users must be able to return to a previously selected event by pressing the Back button.

User interface

The user must be able to navigate the application using the interface below:



- The Favorite button marks the current content to be displayed in a Favorites panel.
- The Back and Forward buttons navigate through the app selection history. Both buttons must be available on all devices.
- The Notes button allows the user to manage notes about the current content.
- The app must support touch, mouse, and stylus input.
- The app layout must automatically adapt to the screen size and orientation.

Technical requirements Layout

You identify the following layout requirements:

General

- All user interface (UI) elements must continuously scale when a user resizes the window.
- UI controls must be smaller and spaced closer together if there is a mouse or stylus available.
- UI controls must be larger and spaced farther apart if the device supports touch and there is no mouse or pointer available.

Timeline

- The timeline must be displayed in a horizontal layout when the device is in a landscape orientation or when the horizontal width is greater than the vertical height.
- The timeline must be displayed in a vertical layout when the device is in a portrait orientation or when the vertical height is greater than the horizontal width.
- Each item in the past must be linked to the next item in the future.
- Users must be able to scroll from past events to future events or from future events to past events.
- The app must only allow one level of detail to be linked to each item in the timeline.

Optimization

You must optimize the app using the following guidelines:

- You must minimize the time it takes to display content when an item on the timeline is selected.
- The app must respect memory and resource constraints for all devices.

XAML coding style

All code and markup must conform to the following style guidelines:

- Use resource dictionaries for styles that are used more than once.
- Limit the use of nested panels.
- Use built-in properties of existing panels instead of using separate style objects.
- Use the navigation structure that best models the data without exceeding the requirements of the app.

Application structure MainPage.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

MP01 <Page
      x: Class = "_70357rm.MainPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:local="using:_70357rm"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:m="http://schemas.openxmlformats.org/markup-compatibility/2006"
      mc: Ignorable="d">
MP02     <Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
MP03
MP04         <RelativePanel BorderBrush="Gray" BorderThickness="10">
MP05             <Rectangle x:Name="A1" Fill="Red" MinHeight="200" MinWidth="400"
                  RelativePanel.AlignLeftWithPanel="True"
                  RelativePanel.AlignTopWithPanel="False" />
MP06             <Rectangle x:Name="B1" Fill="Blue" MinHeight="200" MinWidth="400"
                  RelativePanel.Below="A1"
                  RelativePanel.RightOf="A1"
                  RelativePanel.AlignRightWithPanel="True"
                  RelativePanel.AlignBottomWithPanel="False" />
MP07         </ RelativePanel>
MP08     </ Grid>
MP09 </ Page>
  
```

Settings.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)


```

AS01 <Page
  x: Class ="_70357rm.Settings"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x ="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns: local ="using:_70357rm"
  xmlns: d ="http://schemas.microsoft.com/expression/blend/2008"
  xmlns: m ="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc: Ignorable ="d">
AS02   <Grid Background ="AliceBlue">
AS03     <Border BorderBrush ="DarkBlue" BorderThickness ="3" />
AS04     <Grid Margin ="5 5 5 5">
AS05       <StackPanel HorizontalAlignment ="Center">
AS06         <TextBlock Text ="Date Settings" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center" />
AS07         <StackPanel Orientation ="Horizontal">
AS08           <CheckBox Content ="Center on Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS09           <CheckBox Content ="Set Start Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS10         </StackPanel>
AS11         <TextBlock Text ="Start Date" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center"/>
AS12       <StackPanel Orientation ="Horizontal">
AS13         <TextBlock Text ="Month:" Width ="75" />
AS14         <TextBox Width ="200" />
AS15       </StackPanel>
AS16       <StackPanel Orientation ="Horizontal">
AS17         <TextBlock Text ="Day:" Width ="75" />
AS18         <TextBox Width ="200" />
AS19       </StackPanel>
AS20       <StackPanel Orientation ="Horizontal">
AS21         <TextBlock Text ="Year:" Width ="75" />
AS22         <TextBlock Width ="200" />
AS23       </StackPanel>
AS24       <Ellipse Fill ="Blue" Width ="50" Height="50" Margin ="0 5 0 5"/>
AS25       <TextBlock FontFamily ="Arial" FontSize ="14" Foreground="White" Text ="Save"
Margin ="127 -38 0 0"/>
AS26     </StackPanel>
AS27   </Grid>
AS28 </Grid>
AS29 </Page>
AS30

```

ResourceDictionary.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only.)

```
01 <ResourceDictionary
02     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
03     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
04     xmlns:local="using:_70357rm">
05     <Style x:Key="fill">
06         <Setter Property="Foreground" Value="DarkBlue"/>
07     </Style>
08     <Style x:Key="text">
09         <Setter Property="FontFamily" Value="Arial"/>
10     </Style>
11     <Style x:Key="big">
12         <Setter Property="FontSize" Value="20"/>
13     </Style>
14     <Style x:Key="small">
15         <Setter Property="FontSize" Value="14"/>
16     </Style>
17     <Style x:Key="strong">
18         <Setter Property="FontWeight" Value="Bold"/>
19     </Style>
20     <Style x:Key="light">
21         <Setter Property="FontWeight" Value="Normal"/>
22     </Style>
23     <Style x:Key="normal">
```

```

24     <Setter Property="FontStyle" Value="Normal"/>
25 </Style>
26 <Style x:Key="pad">
27     <Setter Property="Margin" Value="0 5 0 5"/>
28 </Style>
29 <Style x:Key="gap">
30     <Setter Property="Margin" Value="20 0 0 0"/>
31 </Style>
32 <Style x:Key="middle">
33     <Setter Property="HorizontalAlignment" Value="Center"/>
34 </Style>
35 <Style TargetType="CheckBox" x:Key="check">
36     <Setter Property="FontFamily" Value="Arial" />
37     <Setter Property="FontSize" Value="14" />
38     <Setter Property="FontStyle" Value="Normal" />
39     <Setter Property="Margin" Value="20 0 0 0" />
40 </Style>
41 <Style TargetType="TextBox" x:Key="heading">
42     <Setter Property="Foreground" Value="DarkBlue" />
43     <Setter Property="FontFamily" Value="Arial" />
44     <Setter Property="FontSize" Value="20" />
45     <Setter Property="FontStyle" Value="Normal" />
46     <Setter Property="FontWeight" Value="Bold" />
47     <Setter Property="Margin" Value="0 5 0 5" />
48     <Setter Property="HorizontalAlignment" Value="Center" />
49 </Style>
50 </ResourceDictionary>

```

MainPage.xaml.es

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

MX01 private void App_BackRequest(object sender, Windows.UI.Core.BackRequestedEventArgs e)
MX02 {
MX03     Frame page = Window.Current.Content as Frame;
MX04     if ( page != null)
MX05     {
MX06         if ( page.CanGoBack )
MX07         {
MX08             page.GoBack();
MX09         }
MX10     }
MX11 }

```

You are reviewing the App_BackRequested method in the MainPage.xaml.cs file.
For each of the following statements, select Yes if the statement is true. Otherwise, select No NOTE: Each correct selection is worth one point.

Hot Area:

Answer Area

VCE To PDF - Free Practice Exam

Statement	Yes	No
In tablet mode, the app will automatically return to the correct page.	<input type="radio"/>	<input type="radio"/>
In desktop mode, the app will automatically return to the correct page.	<input type="radio"/>	<input type="radio"/>
When a user selects an item on the timeline, the CanGoBack property will always return the value True.	<input type="radio"/>	<input type="radio"/>
The App_BackRequested method meets the navigation requirements.	<input type="radio"/>	<input type="radio"/>

VCEplus
VCE To PDF - Free Practice Exam

Correct Answer:

Answer Area

Statement	Yes	No
In tablet mode, the app will automatically return to the correct page.	<input type="radio"/>	<input checked="" type="radio"/>
In desktop mode, the app will automatically return to the correct page.	<input checked="" type="radio"/>	<input type="radio"/>
When a user selects an item on the timeline, the CanGoBack property will always return the value True .	<input type="radio"/>	<input checked="" type="radio"/>
The App_BackRequested method meets the navigation requirements.	<input type="radio"/>	<input checked="" type="radio"/>

Section: (none)

Explanation

Explanation/Reference:

The App_BackRequested method meets the navigation requirements.

Box 1; No

If we assume that the client is a PC, then the back button is not available in tablet mode.

Box 2: Yes

If we assume that the client is a PC, then the back button is available in desktop mode.

Box 3: No

The user cannot go back from the first item on the timeline.

Box 4; No

From scenario: The Back and Forward buttons navigate through the app selection history. Both buttons must be available on all devices.

Reference: <https://docs.microsoft.com/en-us/windows/uwp/layout/navigation-history-and-backwards-navigation>

QUESTION 7

Case Study

This is a case study. Case studies are not limited separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are developing an application named Timeline that presents information on a timeline. The app allows users to create items and enter details about the item. The app displays item names on a timeline. When users select an item name on the timeline, the app displays additional content about the item.

Business requirements

Timeline section

The timeline element of the app has the following layout requirements:

- The timeline must adapt to the screen size and orientation of the device.
- The timeline size must dynamically change if the window containing the content is resized by the user.
- The user must be able to scroll through the timeline horizontally when the device is in landscape mode.
- The user must be able to scroll through the timeline vertically when the device is in portrait mode.
- The timeline must begin scrolling as soon as a scroll is detected. Scrolling must continue for a short distance after the scroll input has stopped.
- Scroll bars or panning controls must always be visible.

The following image depicts the layout for the timeline section of the app when the device is using landscape orientation:



The following image depicts the layout for the timeline section of the app when the device is using portrait orientation:



Content section

The content element of the app has the following layout requirements:

- When a user selects an item on the timeline, the details for that item must display beneath or to the right of the timeline.
- The content section must display one page of information. The element must be a child of the selected item in the timeline.
- Users must be able to return to a previously selected event by pressing the Back button.

User interface

The user must be able to navigate the application using the interface below:



- The Favorite button marks the current content to be displayed in a Favorites panel.
- The Back and Forward buttons navigate through the app selection history. Both buttons must be available on all devices.
- The Notes button allows the user to manage notes about the current content.
- The app must support touch, mouse, and stylus input.
- The app layout must automatically adapt to the screen size and orientation.

Technical requirements Layout

You identify the following layout requirements:

General

- All user interface (UI) elements must continuously scale when a user resizes the window.
- UI controls must be smaller and spaced closer together if there is a mouse or stylus available.
- UI controls must be larger and spaced farther apart if the device supports touch and there is no mouse or pointer available.

Timeline

- The timeline must be displayed in a horizontal layout when the device is in a landscape orientation or when the horizontal width is greater than the vertical height.
- The timeline must be displayed in a vertical layout when the device is in a portrait orientation or when the vertical height is greater than the horizontal width.
- Each item in the past must be linked to the next item in the future.
- Users must be able to scroll from past events to future events or from future events to past events.
- The app must only allow one level of detail to be linked to each item in the timeline.

Optimization

You must optimize the app using the following guidelines:

- You must minimize the time it takes to display content when an item on the timeline is selected.
- The app must respect memory and resource constraints for all devices.

XAML coding style

All code and markup must conform to the following style guidelines:

- Use resource dictionaries for styles that are used more than once.
- Limit the use of nested panels.
- Use built-in properties of existing panels instead of using separate style objects,
- Use the navigation structure that best models the data without exceeding the requirements of the app.

Application structure MainPage.xaml

Relevant portions of the app files are shown below. {Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.}

```

MP01 <Page
      x: Class = "_70357rm.MainPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:local="using:_70357rm"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:m="http://schemas.openxmlformats.org/markup-compatibility/2006"
      mc: Ignorable="d">
MP02     <Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
MP03
MP04         <RelativePanel BorderBrush="Gray" BorderThickness="10">
MP05             <Rectangle x:Name="A1" Fill="Red" MinHeight="200" MinWidth="400"
                  RelativePanel.AlignLeftWithPanel="True"
                  RelativePanel.AlignTopWithPanel="False" />
MP06             <Rectangle x:Name="B1" Fill="Blue" MinHeight="200" MinWidth="400"
                  RelativePanel.Below="A1"
                  RelativePanel.RightOf="A1"
                  RelativePanel.AlignRightWithPanel="True"
                  RelativePanel.AlignBottomWithPanel="False" />
MP07         </ RelativePanel>
MP08     </ Grid>
MP09 </ Page>
  
```

Settings.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

AS01 <Page
  x: Class ="_70357rm.Settings"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x ="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns: local ="using:_70357rm"
  xmlns: d ="http://schemas.microsoft.com/expression/blend/2008"
  xmlns: m ="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc: Ignorable ="d">
AS02   <Grid Background ="AliceBlue">
AS03     <Border BorderBrush ="DarkBlue" BorderThickness ="3" />
AS04     <Grid Margin ="5 5 5 5">
AS05       <StackPanel HorizontalAlignment ="Center">
AS06         <TextBlock Text ="Date Settings" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center" />
AS07         <StackPanel Orientation ="Horizontal">
AS08           <CheckBox Content ="Center on Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS09           <CheckBox Content ="Set Start Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS10         </StackPanel>
AS11         <TextBlock Text ="Start Date" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center"/>
AS12       <StackPanel Orientation ="Horizontal">
AS13         <TextBlock Text ="Month:" Width ="75" />
AS14         <TextBox Width ="200" />
AS15       </StackPanel>
AS16       <StackPanel Orientation ="Horizontal">
AS17         <TextBlock Text ="Day:" Width ="75" />
AS18         <TextBox Width ="200" />
AS19       </StackPanel>
AS20       <StackPanel Orientation ="Horizontal">
AS21         <TextBlock Text ="Year:" Width ="75" />
AS22         <TextBlock Width ="200" />
AS23       </StackPanel>
AS24       <Ellipse Fill ="Blue" Width ="50" Height="50" Margin ="0 5 0 5"/>
AS25       <TextBlock FontFamily ="Arial" FontSize ="14" Foreground="White" Text ="Save"
Margin ="127 -38 0 0"/>
AS26     </StackPanel>
AS27   </Grid>
AS28 </Grid>
AS29 </Page>
AS30

```

ResourceDictionary.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only.)

```
01 <ResourceDictionary
02     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
03     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
04     xmlns:local="using:_70357rm">
05     <Style x:Key="fill">
06         <Setter Property="Foreground" Value="DarkBlue"/>
07     </Style>
08     <Style x:Key="text">
09         <Setter Property="FontFamily" Value="Arial"/>
10     </Style>
11     <Style x:Key="big">
12         <Setter Property="FontSize" Value="20"/>
13     </Style>
14     <Style x:Key="small">
15         <Setter Property="FontSize" Value="14"/>
16     </Style>
17     <Style x:Key="strong">
18         <Setter Property="FontWeight" Value="Bold"/>
19     </Style>
20     <Style x:Key="light">
21         <Setter Property="FontWeight" Value="Normal"/>
22     </Style>
23     <Style x:Key="normal">
```



```

24     <Setter Property="FontStyle" Value="Normal"/>
25 </Style>
26 <Style x:Key="pad">
27     <Setter Property="Margin" Value="0 5 0 5"/>
28 </Style>
29 <Style x:Key="gap">
30     <Setter Property="Margin" Value="20 0 0 0"/>
31 </Style>
32 <Style x:Key="middle">
33     <Setter Property="HorizontalAlignment" Value="Center"/>
34 </Style>
35 <Style TargetType="CheckBox" x:Key="check">
36     <Setter Property="FontFamily" Value="Arial" />
37     <Setter Property="FontSize" Value="14" />
38     <Setter Property="FontStyle" Value="Normal" />
39     <Setter Property="Margin" Value="20 0 0 0" />
40 </Style>
41 <Style TargetType="TextBox" x:Key="heading">
42     <Setter Property="Foreground" Value="DarkBlue" />
43     <Setter Property="FontFamily" Value="Arial" />
44     <Setter Property="FontSize" Value="20" />
45     <Setter Property="FontStyle" Value="Normal" />
46     <Setter Property="FontWeight" Value="Bold" />
47     <Setter Property="Margin" Value="0 5 0 5" />
48     <Setter Property="HorizontalAlignment" Value="Center" />
49 </Style>
50 </ResourceDictionary>

```

MainPage.xaml.es

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

MX01 private void App_BackRequest(object sender, Windows.UI.Core.BackRequestedEventArgs e)
MX02 {
MX03     Frame page = Window.Current.Content as Frame;
MX04     if ( page != null)
MX05     {
MX06         if ( page.CanGoBack )
MX07         {
MX08             page.GoBack();
MX09         }
MX10     }
MX11 }

```

You need to design the navigation for the timeline.
What navigation should you use?

- A. hierarchy
- B. peer
- C. hub
- D. master/details

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Explanation:

From scenario:

- Each item in the past must be linked to the next item in the future.
- Users must be able to scroll from past events to future events or from future events to past events.
- The app must only allow one level of detail to be linked to each item in the timeline.

Here we can use a hierarchy with each parent node having only one single child node.

Hierarchical structures are good for organizing complex content that spans lots of pages or when pages should be viewed in a particular order. The downside is that hierarchical pages introduce some navigation overhead: the deeper the structure, the more clicks it takes for users to get from page to page.

We recommend a hierarchical structure when:

You expect the user to traverse the pages in a specific order. Arrange the hierarchy to enforce that order.

There is a clear parent-child relationship between one of the pages and the other pages in the group.

There are more than 7 pages in the group.

When there are more than 7 pages in the group, it might be difficult for users to understand how the pages are unique or to understand their current location within the group. If you don't think that's an issue for your app, go ahead and make the pages peers

Reference: <https://docs.microsoft.com/en-us/windows/uwp/layout/navigation-basics>

QUESTION 8

Case Study

This is a case study. Case studies are not limited separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the

questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are developing an application named Timeline that presents information on a timeline. The app allows users to create items and enter details about the item. The app displays item names on a timeline. When users select an item name on the timeline, the app displays additional content about the item.

Business requirements

Timeline section

The timeline element of the app has the following layout requirements:

- The timeline must adapt to the screen size and orientation of the device.
- The timeline size must dynamically change if the window containing the content is resized by the user.
- The user must be able to scroll through the timeline horizontally when the device is in landscape mode.
- The user must be able to scroll through the timeline vertically when the device is in portrait mode.
- The timeline must begin scrolling as soon as a scroll is detected. Scrolling must continue for a short distance after the scroll input has stopped.
- Scroll bars or panning controls must always be visible.

The following image depicts the layout for the timeline section of the app when the device is using landscape orientation:



The following image depicts the layout for the timeline section of the app when the device is using portrait orientation:



Content section

The content element of the app has the following layout requirements:

- When a user selects an item on the timeline, the details for that item must display beneath or to the right of the timeline.
- The content section must display one page of information. The element must be a child of the selected item in the timeline.
- Users must be able to return to a previously selected event by pressing the Back button.

User interface

The user must be able to navigate the application using the interface below:



- The Favorite button marks the current content to be displayed in a Favorites panel.
- The Back and Forward buttons navigate through the app selection history. Both buttons must be available on all devices.
- The Notes button allows the user to manage notes about the current content.
- The app must support touch, mouse, and stylus input.
- The app layout must automatically adapt to the screen size and orientation.

Technical requirements Layout

You identify the following layout requirements:

General

- All user interface (UI) elements must continuously scale when a user resizes the window.
- UI controls must be smaller and spaced closer together if there is a mouse or stylus available.
- UI controls must be larger and spaced farther apart if the device supports touch and there is no mouse or pointer available.

Timeline

- The timeline must be displayed in a horizontal layout when the device is in a landscape orientation or when the horizontal width is greater than the vertical height.
- The timeline must be displayed in a vertical layout when the device is in a portrait orientation or when the vertical height is greater than the horizontal width.
- Each item in the past must be linked to the next item in the future.
- Users must be able to scroll from past events to future events or from future events to past events.
- The app must only allow one level of detail to be linked to each item in the timeline.

Optimization

You must optimize the app using the following guidelines:

- You must minimize the time it takes to display content when an item on the timeline is selected.
- The app must respect memory and resource constraints for all devices.

XAML coding style

All code and markup must conform to the following style guidelines:

- Use resource dictionaries for styles that are used more than once.
- Limit the use of nested panels.
- Use built-in properties of existing panels instead of using separate style objects.
- Use the navigation structure that best models the data without exceeding the requirements of the app.

Application structure MainPage.xaml

Relevant portions of the app files are shown below. {Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

MP01 <Page
      x: Class = "_70357rm.MainPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:local="using:_70357rm"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:m="http://schemas.openxmlformats.org/markup-compatibility/2006"
      mc: Ignorable="d">
MP02     <Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
MP03
MP04         <RelativePanel BorderBrush="Gray" BorderThickness="10">
MP05             <Rectangle x:Name="A1" Fill="Red" MinHeight="200" MinWidth="400"
                  RelativePanel.AlignLeftWithPanel="True"
                  RelativePanel.AlignTopWithPanel="False" />
MP06             <Rectangle x:Name="B1" Fill="Blue" MinHeight="200" MinWidth="400"
                  RelativePanel.Below="A1"
                  RelativePanel.RightOf="A1"
                  RelativePanel.AlignRightWithPanel="True"
                  RelativePanel.AlignBottomWithPanel="False" />
MP07         </ RelativePanel>
MP08     </ Grid>
MP09 </ Page>
  
```

Settings.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)


```

AS01 <Page
  x: Class ="_70357rm.Settings"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x ="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns: local ="using:_70357rm"
  xmlns: d ="http://schemas.microsoft.com/expression/blend/2008"
  xmlns: m ="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc: Ignorable ="d">
AS02   <Grid Background ="AliceBlue">
AS03     <Border BorderBrush ="DarkBlue" BorderThickness ="3" />
AS04     <Grid Margin ="5 5 5 5">
AS05       <StackPanel HorizontalAlignment ="Center">
AS06         <TextBlock Text ="Date Settings" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center" />
AS07         <StackPanel Orientation ="Horizontal">
AS08           <CheckBox Content ="Center on Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS09           <CheckBox Content ="Set Start Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS10         </StackPanel>
AS11         <TextBlock Text ="Start Date" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center"/>
AS12       <StackPanel Orientation ="Horizontal">
AS13         <TextBlock Text ="Month:" Width ="75" />
AS14         <TextBox Width ="200" />
AS15       </StackPanel>
AS16       <StackPanel Orientation ="Horizontal">
AS17         <TextBlock Text ="Day:" Width ="75" />
AS18         <TextBox Width ="200" />
AS19       </StackPanel>
AS20       <StackPanel Orientation ="Horizontal">
AS21         <TextBlock Text ="Year:" Width ="75" />
AS22         <TextBlock Width ="200" />
AS23       </StackPanel>
AS24       <Ellipse Fill ="Blue" Width ="50" Height="50" Margin ="0 5 0 5"/>
AS25       <TextBlock FontFamily ="Arial" FontSize ="14" Foreground="White" Text ="Save"
Margin ="127 -38 0 0"/>
AS26     </StackPanel>
AS27   </Grid>
AS28 </Grid>
AS29 </Page>
AS30 </Page>

```

ResourceDictionary.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only.)

```
01 <ResourceDictionary
02     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
03     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
04     xmlns:local="using:_70357rm">
05     <Style x:Key="fill">
06         <Setter Property="Foreground" Value="DarkBlue"/>
07     </Style>
08     <Style x:Key="text">
09         <Setter Property="FontFamily" Value="Arial"/>
10     </Style>
11     <Style x:Key="big">
12         <Setter Property="FontSize" Value="20"/>
13     </Style>
14     <Style x:Key="small">
15         <Setter Property="FontSize" Value="14"/>
16     </Style>
17     <Style x:Key="strong">
18         <Setter Property="FontWeight" Value="Bold"/>
19     </Style>
20     <Style x:Key="light">
21         <Setter Property="FontWeight" Value="Normal"/>
22     </Style>
23     <Style x:Key="normal">
```

```

24     <Setter Property="FontStyle" Value="Normal"/>
25 </Style>
26 <Style x:Key="pad">
27     <Setter Property="Margin" Value="0 5 0 5"/>
28 </Style>
29 <Style x:Key="gap">
30     <Setter Property="Margin" Value="20 0 0 0"/>
31 </Style>
32 <Style x:Key="middle">
33     <Setter Property="HorizontalAlignment" Value="Center"/>
34 </Style>
35 <Style TargetType="CheckBox" x:Key="check">
36     <Setter Property="FontFamily" Value="Arial" />
37     <Setter Property="FontSize" Value="14" />
38     <Setter Property="FontStyle" Value="Normal" />
39     <Setter Property="Margin" Value="20 0 0 0" />
40 </Style>
41 <Style TargetType="TextBox" x:Key="heading">
42     <Setter Property="Foreground" Value="DarkBlue" />
43     <Setter Property="FontFamily" Value="Arial" />
44     <Setter Property="FontSize" Value="20" />
45     <Setter Property="FontStyle" Value="Normal" />
46     <Setter Property="FontWeight" Value="Bold" />
47     <Setter Property="Margin" Value="0 5 0 5" />
48     <Setter Property="HorizontalAlignment" Value="Center" />
49 </Style>
50 </ResourceDictionary>

```

MainPage.xaml.es

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

MX01 private void App_BackRequest(object sender, Windows.UI.Core.BackRequestedEventArgs e)
MX02 {
MX03     Frame page = Window.Current.Content as Frame;
MX04     if ( page != null)
MX05     {
MX06         if ( page.CanGoBack )
MX07         {
MX08             page.GoBack();
MX09         }
MX10     }
MX11 }

```

You need to ensure that the Timeline app meets the XAML coding requirements.
In Settings.xaml, which markup segment should you select to replace the markup segment at line AS06?

A.

```
<TextBlock Text="Date Settings" Foreground="{StaticResource fill}"  
FontFamily="{StaticResource text}"  
FontSize="{StaticResource big}" FontStyle="{StaticResource normal}"  
FontWeight="{StaticResource strong}"  
Margin="{StaticResource pad}" HorizontalAlignment="{StaticResource middle}"/>
```

B.

```
<TextBlock Text="Date Settings" Foreground="{StaticResource fill}"  
FontFamily="Normal"  
FontSize="{StaticResource big}" FontStyle="Normal"  
FontWeight="{StaticResource strong}"  
Margin="{StaticResource pad}" HorizontalAlignment="{StaticResource middle}"/>
```

C.

```
<TextBlock Text="Date Settings" Style="{ ThemeResource heading }" />
```

D.

```
<TextBlock Text="Date Settings" Style="{ StaticResource heading }" />
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Explanation:

From scenario: All code and markup must conform to the following style guidelines:

- Use resource dictionaries for styles that are used more than once.

- Use built-in properties of existing panels instead of using separate style objects,

XAML resources are objects that are referenced from markup more than once. Resources are defined in a ResourceDictionary, typically in a separate file or at the top of the markup page. In this scenario the ResourceDictionary is defined in the ResourceDictionary.xaml file.

You access members of the resource dictionary like any other dictionary.

Reference: <https://docs.microsoft.com/en-us/windows/uwp/controls-and-patterns/resourcedictionary-and-xaml-resource-references>

QUESTION 9**Case Study**

This is a case study. Case studies are not limited separately. You can use as much exam time as you would like to complete each case. However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other question on this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next sections of the exam. After you begin a new section, you cannot return to this section.

To start the case study

To display the first question on this case study, click the Next button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an All Information tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the Question button to return to the question.

Background

You are developing an application named Timeline that presents information on a timeline. The app allows users to create items and enter details about the item.

The app displays item names on a timeline. When users select an item name on the timeline, the app displays additional content about the item.

Business requirements**Timeline section**

The timeline element of the app has the following layout requirements:

- The timeline must adapt to the screen size and orientation of the device.
- The timeline size must dynamically change if the window containing the content is resized by the user.
- The user must be able to scroll through the timeline horizontally when the device is in landscape mode.
- The user must be able to scroll through the timeline vertically when the device is in portrait mode.
- The timeline must begin scrolling as soon as a scroll is detected. Scrolling must continue for a short distance after the scroll input has stopped.
- Scroll bars or panning controls must always be visible.

The following image depicts the layout for the timeline section of the app when the device is using landscape orientation:



The following image depicts the layout for the timeline section of the app when the device is using portrait orientation:



Content section

The content element of the app has the following layout requirements:

- When a user selects an item on the timeline, the details for that item must display beneath or to the right of the timeline.
- The content section must display one page of information. The element must be a child of the selected item in the timeline.
- Users must be able to return to a previously selected event by pressing the Back button.

User interface

The user must be able to navigate the application using the interface below:



- The Favorite button marks the current content to be displayed in a Favorites panel.
- The Back and Forward buttons navigate through the app selection history. Both buttons must be available on all devices.
- The Notes button allows the user to manage notes about the current content.
- The app must support touch, mouse, and stylus input.
- The app layout must automatically adapt to the screen size and orientation.

Technical requirements Layout

You identify the following layout requirements:

General

- All user interface (UI) elements must continuously scale when a user resizes the window.
- UI controls must be smaller and spaced closer together if there is a mouse or stylus available.
- UI controls must be larger and spaced farther apart if the device supports touch and there is no mouse or pointer available.

Timeline

- The timeline must be displayed in a horizontal layout when the device is in a landscape orientation or when the horizontal width is greater than the vertical height.
- The timeline must be displayed in a vertical layout when the device is in a portrait orientation or when the vertical height is greater than the horizontal width.
- Each item in the past must be linked to the next item in the future.
- Users must be able to scroll from past events to future events or from future events to past events.
- The app must only allow one level of detail to be linked to each item in the timeline.

Optimization

You must optimize the app using the following guidelines:

- You must minimize the time it takes to display content when an item on the timeline is selected.
- The app must respect memory and resource constraints for all devices.

XAML coding style

All code and markup must conform to the following style guidelines:

- Use resource dictionaries for styles that are used more than once.
- Limit the use of nested panels.
- Use built-in properties of existing panels instead of using separate style objects,
- Use the navigation structure that best models the data without exceeding the requirements of the app.

Application structure MainPage.xaml

Relevant portions of the app files are shown below. {Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.}

```

MP01 <Page
      x: Class = "_70357rm.MainPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:local="using:_70357rm"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:m="http://schemas.openxmlformats.org/markup-compatibility/2006"
      mc: Ignorable = "d">
MP02     <Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
MP03
MP04         <RelativePanel BorderBrush="Gray" BorderThickness="10">
MP05             <Rectangle x:Name="A1" Fill="Red" MinHeight="200" MinWidth="400"
                  RelativePanel.AlignLeftWithPanel="True"
                  RelativePanel.AlignTopWithPanel="False" />
MP06             <Rectangle x:Name="B1" Fill="Blue" MinHeight="200" MinWidth="400"
                  RelativePanel.Below="A1"
                  RelativePanel.RightOf="A1"
                  RelativePanel.AlignRightWithPanel="True"
                  RelativePanel.AlignBottomWithPanel="False" />
MP07         </ RelativePanel>
MP08     </ Grid>
MP09 </ Page>
  
```

Settings.xaml

Relevant portions of the app files are shown below. {Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.}

```

AS01 <Page
  x: Class ="_70357rm.Settings"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x ="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns: local ="using:_70357rm"
  xmlns: d ="http://schemas.microsoft.com/expression/blend/2008"
  xmlns: m ="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc: Ignorable ="d">
AS02   <Grid Background ="AliceBlue">
AS03     <Border BorderBrush ="DarkBlue" BorderThickness ="3" />
AS04     <Grid Margin ="5 5 5 5">
AS05       <StackPanel HorizontalAlignment ="Center">
AS06         <TextBlock Text ="Date Settings" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center" />
AS07         <StackPanel Orientation ="Horizontal">
AS08           <CheckBox Content ="Center on Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS09           <CheckBox Content ="Set Start Date" FontFamily ="Arial" FontSize="14"
FontStyle "Normal" Margin ="20,0,0,0"/>
AS10         </StackPanel>
AS11         <TextBlock Text ="Start Date" Foreground ="DarkBlue" FontFamily="Arial"
FontSize ="20" FontStyle ="Normal"
          FontWeight ="Bold" Margin ="0 5 0 5" HorizontalAlignment="Center"/>
AS12       <StackPanel Orientation ="Horizontal">
AS13         <TextBlock Text ="Month:" Width ="75" />
AS14         <TextBox Width ="200" />
AS15       </StackPanel>
AS16       <StackPanel Orientation ="Horizontal">
AS17         <TextBlock Text ="Day:" Width ="75" />
AS18         <TextBox Width ="200" />
AS19       </StackPanel>
AS20       <StackPanel Orientation ="Horizontal">
AS21         <TextBlock Text ="Year:" Width ="75" />
AS22         <TextBox Width ="200" />
AS23       </StackPanel>
AS24       <Ellipse Fill ="Blue" Width ="50" Height="50" Margin ="0 5 0 5"/>
AS25       <TextBlock FontFamily ="Arial" FontSize ="14" Foreground="White" Text ="Save"
Margin ="127 -38 0 0"/>
AS26     </StackPanel>
AS27   </Grid>
AS28 </Grid>
AS29 </Page>
AS30 </Page>

```

ResourceDictionary.xaml

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only.)

```
01 <ResourceDictionary
02     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
03     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
04     xmlns:local="using:_70357rm">
05     <Style x:Key="fill">
06         <Setter Property="Foreground" Value="DarkBlue"/>
07     </Style>
08     <Style x:Key="text">
09         <Setter Property="FontFamily" Value="Arial"/>
10     </Style>
11     <Style x:Key="big">
12         <Setter Property="FontSize" Value="20"/>
13     </Style>
14     <Style x:Key="small">
15         <Setter Property="FontSize" Value="14"/>
16     </Style>
17     <Style x:Key="strong">
18         <Setter Property="FontWeight" Value="Bold"/>
19     </Style>
20     <Style x:Key="light">
21         <Setter Property="FontWeight" Value="Normal"/>
22     </Style>
23     <Style x:Key="normal">
```

```

24     <Setter Property="FontStyle" Value="Normal"/>
25 </Style>
26 <Style x:Key="pad">
27     <Setter Property="Margin" Value="0 5 0 5"/>
28 </Style>
29 <Style x:Key="gap">
30     <Setter Property="Margin" Value="20 0 0 0"/>
31 </Style>
32 <Style x:Key="middle">
33     <Setter Property="HorizontalAlignment" Value="Center"/>
34 </Style>
35 <Style TargetType="CheckBox" x:Key="check">
36     <Setter Property="FontFamily" Value="Arial" />
37     <Setter Property="FontSize" Value="14" />
38     <Setter Property="FontStyle" Value="Normal" />
39     <Setter Property="Margin" Value="20 0 0 0" />
40 </Style>
41 <Style TargetType="TextBox" x:Key="heading">
42     <Setter Property="Foreground" Value="DarkBlue" />
43     <Setter Property="FontFamily" Value="Arial" />
44     <Setter Property="FontSize" Value="20" />
45     <Setter Property="FontStyle" Value="Normal" />
46     <Setter Property="FontWeight" Value="Bold" />
47     <Setter Property="Margin" Value="0 5 0 5" />
48     <Setter Property="HorizontalAlignment" Value="Center" />
49 </Style>
50 </ResourceDictionary>

```

MainPage.xaml.es

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

```

MX01 private void App_BackRequest(object sender, Windows.UI.Core.BackRequestedEventArgs e)
MX02 {
MX03     Frame page = Window.Current.Content as Frame;
MX04     if ( page != null)
MX05     {
MX06         if ( page.CanGoBack )
MX07         {
MX08             page.GoBack();
MX09         }
MX10     }
MX11 }

```


You need to configure the app to meet the load time requirements.
What should you do?

- A. Set the value of the CacheSize to 0.
- B. Set the value of the CacheMode property to BitmapCache.
- C. Set the value of the NavigationCacheMode property to Enabled.
- D. Set the value of the NavigationCacheMode property to Disabled.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

Explanation:

Scenario: You must optimize the app using the following guidelines:

You must minimize the time it takes to display content when an item on the timeline is selected.

The app must respect memory and resource constraints for all devices.

You use the NavigationCacheMode property to specify whether a new instance of the page is created for each visit to the page or if a previously constructed instance of the page that has been saved in the cache is used for each visit.

The default value for the NavigationCacheMode property is Disabled. Set the NavigationCacheMode property to Enabled or Required when a new instance of the page is not essential for each visit. By using a cached instance of the page, you can improve the performance of your application and reduce the load on your server.

Reference: [https://msdn.microsoft.com/en-us/library/system.windows.controls.page.navigationcachemode\(v=vs.95\).aspx](https://msdn.microsoft.com/en-us/library/system.windows.controls.page.navigationcachemode(v=vs.95).aspx)

QUESTION 10

HOTSPOT

You have an app that includes the following method:

```
private void GeoMode()  
{  
    Geolocator geo = new Geolocator();  
    geo.DesiredAccuracy = Windows.Devices.Geolocation.PositionAccuracy.High;  
}
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No

Hot Area:



Answer Area

Statement

The position accuracy will be 50 meters.

The code will configure the device to use the most conservative power settings.

Yes

No

☐
☐
☐
☐


Correct Answer:



Answer Area

Statement

The position accuracy will be 50 meters.

The code will configure the device to use the most conservative power settings.

Yes

No

☐
☒
☐
☒


Section: (none)

Explanation

Explanation/Reference:

The code will configure the device to use the most conservative power settings.

Box 1:No

When neither DesiredAccuracyInMeters nor DesiredAccuracy are set, your app will use an accuracy setting of 500 meters (which corresponds to the

DesiredAccuracy setting of Default). Setting DesiredAccuracy to Default or High indirectly sets DesiredAccuracyInMeters to 500 or 10 meters, respectively.

Box 2: No

The Default value should be used to optimize for power, performance, and other cost considerations.

The High value should be used to deliver the most accurate report possible. This includes using services that might charge money, or consuming higher levels of battery power or connection bandwidth. An accuracy level of High may degrade system performance and should be used only when necessary.

Reference:

<https://docs.microsoft.com/en-us/uwp/api/windows.devices.geolocation.geolocator>

<https://docs.microsoft.com/en-us/uwp/api/windows.devices.geolocation.positionaccuracy>

QUESTION 11

DRAG DROP

You are creating a Universal Windows Platform (UWP) app that takes pictures.

You want to use the camera's built-in interface for taking the pictures.

You need to capture an image from the device's built-in camera.

How should you complete the method? To answer, drag the appropriate code segments to the correct location or locations. Each code segments may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Select and Place:

Code segments

CameraCaptureUI

CameraCaptureUIPhotoFormat

MediaCapture

ImageEncodingProperties

PhotoSettings

VCEplus
VCE To PDF - Free Practice Exam

Answer Area

```
public async void GetImage()  
{  
    Code segment capture = new Code segment ();  
    capture.PhotoSettings.Format = PhotoFormat.Jpeg;  
    StorageFile photo = await capture.CaptureFileAsync(CameraCaptureUIMode.Photo);  
}
```

VCEplus
VCE To PDF - Free Practice Exam

Correct Answer:

Code segments

CameraCaptureUI

CameraCaptureUIPhotoFormat

MediaCapture

ImageEncodingProperties

PhotoSettings

VCEplus
VCE To PDF - Free Practice Exam

Answer Area

```
public async void GetImage()
{
    CameraCaptureUI capture = new CameraCaptureUI();
    capture.PhotoSettings.PhotoFormat = PhotoFormat.Jpeg;
    StorageFile photo = await capture.CaptureFileAsync(CameraCaptureUIMode.Photo);
}
```

VCEplus
VCE To PDF - Free Practice Exam

Section: (none)

Explanation

Explanation/Reference:

Box 1: CameraCaptureUI Box 2: CameraCaptureUI

Example: Using Windows.Media.Capture.CameraCaptureUI API to capture a photo CameraCaptureUI dialog = new CameraCaptureUI();

Size aspectRatio = new Size(16, 9);

dialog.PhotoSettings.CroppedAspectRatio = aspectRatio;

StorageFile file = await dialog.CaptureFileAsync(CameraCaptureUIMode.Photo);

Reference: <https://docs.microsoft.com/en-us/uwp/api/windows.media.capture.cameracaptureui>

QUESTION 12

HOTSPOT

You are developing an app that displays photos.

You need to create a method that displays informational text when a user hovers the pointer over a photo.
How should you complete the method? To answer, select the appropriate code segment from each list in the answer area

Hot Area:

Answer Area

```
private void CreateToolTip(
    [DependencyObject] target,
    [DependencyObject] content)
{
    [ToolTip] tip = new [ToolTipService] ();
    tip.Content = content;
    [ToolTipService].SetToolTip(target, tip);
}
```

The answer area contains four dropdown menus for selecting code segments to complete the method. The segments are as follows:

- Parameter 1 (target): DependencyObject, String
- Parameter 2 (content): DependencyObject, String
- Line 4 (tip = new ... ());: ToolTip, ToolTipService
- Line 5 (.SetToolTip(target, tip);): ToolTip, ToolTipService

Correct Answer:

Answer Area

```

private void CreateToolTip(
    target,
    content)
{
    tip = new ToolTip(
        ToolTipService
    );
    tip.Content = content;
    .SetToolTip(target, tip);
}

```

DependencyObject
String

ToolTip
ToolTipService

ToolTip
ToolTipService

ToolTip
ToolTipService

Section: (none)
Explanation

Explanation/Reference:

Box 1: DependencyObject
Box 2: String
Box 3: ToolTip
Box 4: ToolTip
Box 5: ToolTipService

A ToolTip must be assigned to another UI element that is its owner. In Extensible Application Markup Language (XAML), use the ToolTipService.ToolTip attached property to assign the ToolTip to an owner. In code, use the ToolTipService.SetToolTip method to assign the ToolTip to an owner.

The SetToolTip(DependencyObject, Object) method sets the value of the ToolTipService.ToolTip XAML attached property Reference: https://docs.microsoft.com/en-us/uwp/api/windows.ui.xaml.controls.tooltipservice#Windows_UI_XamlControls_ToolTipService_ToolTipProperty

QUESTION 13

HOTSPOT

You are developing a Universal Windows Platform (UWP) app by using XAML and C#. A team member has written a XAML page that includes a button with an event handler method named ButtonSendNotification_Click() registered to the Click event.

You are reviewing the following code segment written by the team member (line numbers are added for reference only):

```

01 public sealed partial class MainPage : Page
02 {
03     public MainPage()
04     {
05         InitializeComponent();
06         TileUpdateManager.CreateTileUpdaterForApplication().EnableNo-
tificationQueue( true);
07     }
08     private void ButtonSendNotification_Click( object sender, RoutedEventArgs e)
09     {
10         SendTileNotification();
11     }
12     private static string GetNewsTitle()
13     {
14         ...
15     }
16     private void SendTileNotification()
17     {
18         TileNotification tileNotification = GenerateTileNotification();
19         tileNotification.Tag = "newsItem" + GetNewsTitle();
20         TileUpdateManager .CreateTileUpdaterForApplication().Update(tileNotifica-
tion);
21     }
22     private TileNotification GenerateTileNotification()
23     {

```

```

24     string xml = @"
25         <tile version='3'>
26             <visual branding='name'>
27                 <binding template='TileMedium'>
28                     <text hint-wrap='true'>This just in...</text>
29                     <text hint-wrap='true' hint-style='captionSubtle' />
30                 </binding>
31                 <binding template='TileWide'>
32                     <text hint-wrap='true'>This just in...</text>
33                     <text hint-wrap='true' hint-style='captionSubtle' />
34                 </binding>
35                 <binding template='TileLarge'>
36                     <text hint-wrap='true'>This just in...</text>
37                     <text hint-wrap='true' hint-style='captionSubtle' />
38                 </binding>
39             </visual>
40         </tile>";
41     XmlDocument doc = new XmlDocument ();
42     doc.LoadXml(xml);
43     string nowTimeString = DateTime.Now.ToString();
44     foreach (XmlElement textEl in doc.SelectNodes("//text").OfType<XmlElement>())
45     {
46         if (textEl.InnerText.Length == 0)
47             textEl.InnerText = "(" + nowTimeString + ") Top News: " + GetNew-
48             sTitle();
49     }
50 }

```

For each of the following statements, select Yes if the statement is true. Otherwise, select No. NOTE: Each selection is worth one point.

Hot Area:

Answer Area
VCE To PDF - Free Practice Exam

Statement

Yes

No

The code segment will generate a tile notification for all platform tile sizes.

☐
☐

The code segment will generate a tile notification successfully when a user clicks the button on the XAML page.

☐
☐

The app will display only one tile notification, regardless of the number of button clicks.

☐
☐

An exception will be thrown at Line 42 of the code segment.

☐
☐

Correct Answer:

Answer Area
VCE To PDF - Free Practice Exam

Statement

Yes

No

The code segment will generate a tile notification for all platform tile sizes.

☐
☒

The code segment will generate a tile notification successfully when a user clicks the button on the XAML page.

☒
☐

The app will display only one tile notification, regardless of the number of button clicks.

☐
☒

An exception will be thrown at Line 42 of the code segment.

☐
☒

Section: (none)

Explanation

Explanation/Reference:

Box 1: No

There are four tile sizes: small, medium, wide, large. Only three are reference in the TileNotification definition.

Box 2: Yes

Box 3: No

Box 4: No

Line 42 is doc.LoadXml(xml);

The LoadXml method loads an XML document from a string. Returns TRUE on success or FALSE on failure.

If called statically, returns a DOMDocument or FALSE on failure.

If an empty string is passed as the source, a warning will be generated. This warning is not generated by libxml and cannot be handled using libxml's error handling functions.

Reference: <https://docs.microsoft.com/en-us/windows/uwp/controls-and-patterns/tiles-and-notifications-app-assets>

QUESTION 14

HOTSPOT

You are developing a Universal Windows Platform (UWP) app.

You need to implement responsive user design patterns.

Which of the following techniques are supported? To answer, select the appropriate option from each list in the answer area.

Hot Area:

Answer Area

Technique	Response
Reposition	<input type="text"/> Supported Not supported
Resize	<input type="text"/> Supported Not supported
Replace	<input type="text"/> Supported Not supported
Reflow	<input type="text"/> Supported Not supported
Reveal	<input type="text"/> Supported Not supported

Correct Answer:

Answer Area	
Technique	Response
Reposition	<input checked="" type="checkbox"/> Supported <input type="checkbox"/> Not supported
Resize	<input checked="" type="checkbox"/> Supported <input type="checkbox"/> Not supported
Replace	<input checked="" type="checkbox"/> Supported <input type="checkbox"/> Not supported
Reflow	<input checked="" type="checkbox"/> Supported <input type="checkbox"/> Not supported
Reveal	<input checked="" type="checkbox"/> Supported <input type="checkbox"/> Not supported

Section: (none)

Explanation

Explanation/Reference:

Responsive design techniques

When you optimize your app's UI for specific screen widths, we say that you're creating a responsive design. Here are six responsive design techniques you can use to customize your app's UI.

*Reposition

You can alter the location and position of app UI elements to get the most out of each device

*Resize

You can optimize the frame size by adjusting the margins and size of UI elements.

*Reflow

By changing the flow of UI elements based on device and orientation, your app can offer an optimal display of content

* Show/hide
You can show or hide UI elements based on screen real estate, or when the device supports additional functionality, specific situations, or preferred screen orientations.

*Replace

This technique lets you switch the user interface for a specific device size-class or orientation. In this example, the nav pane and its compact, transient UI works well for a smaller device, but on a larger device tabs might be a better choice.

*Re-architect

You can collapse or fork the architecture of your app to better target specific devices.

Reference: <https://docs.microsoft.com/en-us/windows/uwp/layout/design-and-ui-intro>

QUESTION 15

HOTSPOT

You are developing a Universal Windows Platform (UWP) app.

The app does not display content properly on mobile devices.

You need to support smaller window sizes.

How should you complete the relevant XAML markup? To answer, select the appropriate markup segment from each list in the answer area.

Hot Area:

Answer Area

VisualStateGroup

VisualState

VisualState.StateTriggers

MediaQuery

AdaptiveTrigger

AdaptiveBehavior

MinWindowWidth="0"

MinWindowWidth="768"

MaxWindowWidth="256"

MaxWindowWidth="768"

VisualState.StateTriggers

VisualState.Setters

<Setter Target="SplitViewControl.IsPaneOpen" Value ="False" />

<Setter Target="SplitViewControl.DisplayMode" Value ="CompactInline"/>

VisualState

VisualState.StateTriggers

MediaQuery

AdaptiveTrigger

AdaptiveBehavior

MinWindowWidth="0"

MinWindowWidth="768"

MaxWindowWidth="256"

MaxWindowWidth="768"

VisualState.StateTriggers

VisualState.Setters

<Setter Target="SplitViewControl.IsPaneOpen" Value ="True" />

<Setter Target="SplitViewControl.DisplayMode" Value ="Inline"/>

VisualState

VisualStateGroup

VisualStateManager.VisualStateGroups

VCE To PDF - Free Practice Exam

Correct Answer:

Answer Area

```

<SplitView :Name="SplitViewControl" OpenPanelLength="128">
  <VisualStateManager.VisualStateGroups>
    <VisualStateGroup>
      <VisualState>
        <VisualState.StateTriggers>
          <MediaQuery MinWindowWidth="0" />
          <AdaptiveTrigger MinWindowWidth="768" />
          <AdaptiveBehavior MaxWindowWidth="256" />
          <AdaptiveBehavior MaxWindowWidth="768" />
        </VisualState.StateTriggers>
        <VisualState.Setters>
          <Setter Target="SplitViewControl.IsPaneOpen" Value="False" />
          <Setter Target="SplitViewControl.DisplayMode" Value="CompactInline" />
        </VisualState.Setters>
      </VisualState>
      <VisualState>
        <VisualState.StateTriggers>
          <MediaQuery MinWindowWidth="0" />
          <AdaptiveTrigger MinWindowWidth="768" />
          <AdaptiveBehavior MaxWindowWidth="256" />
          <AdaptiveBehavior MaxWindowWidth="768" />
        </VisualState.StateTriggers>
        <VisualState.Setters>
          <Setter Target="SplitViewControl.IsPaneOpen" Value="True" />
          <Setter Target="SplitViewControl.DisplayMode" Value="Inline" />
        </VisualState.Setters>
      </VisualState>
    </VisualStateGroup>
  </VisualStateManager.VisualStateGroups>
</SplitView>
  
```

Section: (none)

Explanation

Explanation/Reference:

One of the tools that Microsoft gives us for building adaptive UIs in UWP apps is state triggers. The version of Windows 10 released at BUILD 2015 features one state trigger: a class named AdaptiveTrigger. AdaptiveTrigger has two important properties: MinWindowWidth and MinWindowHeight. You use AdaptiveTrigger in conjunction with Visual State Manager to adapt the UI to screens and windows of various sizes.

*Inline

The pane is always visible and doesn't overlay the content area. The pane and content areas divide the available screen real estate.

*CompactInline

A narrow portion of the pane is always visible in this mode, which is just wide enough to show icons. The default closed pane width is 48px, which can be modified with CompactPanelLength. If the pane is opened, it will reduce the space available for content, pushing the content out of its way.

Reference: <http://www.wintellect.com/devcenter/jprosize/using-adaptivettrigger-to-build-adaptive-uis-in-windows-10>

QUESTION 16

DRAG DROP

You are developing a Universal Windows Platform (UWP) app.

The app has the following requirements:

- Users must be able to authenticate with the app by using a third-party OAuth provider.
- User must have the option of using Single Sign-On.

You obtain the security identifier (SID) for the app from the Windows Dev Center.

You need to implement authentication for the app.

In which order should you perform the actions? To answer, move all actions from the list of actions to the answer area and arrange them in the correct order.

Select and Place:

Actions

Register the app with the OAuth provider.

Retrieve user information from the OAuth provider.

Call the AuthenticateAsync method of the Web Authentication Broker.

Construct the authentication request.

Answer Area

⏪

⏩

⏴

⏵

VCEplus
VCE To PDF - Free Practice Exam

Correct Answer:

Actions	Answer Area
	Register the app with the OAuth provider.
	Call the AuthenticateAsync method of the Web Authentication Broker.
	Retrieve user information from the OAuth provider.
	Construct the authentication request.

Section: (none)
Explanation

Explanation/Reference:

Step 1: Register your app with your online provider

You must register your app with the online identity provider to which you want to connect. You can find out how to register your app from the identity provider. After registering, the online provider typically gives you an Id or secret key for your app.

Step 2: Build the authentication request URI

The request URI consists of the address where you send the authentication request to your online provider appended with other required information, such as an app ID or secret, a redirect URI where the user is sent after completing authentication, and the expected response type. You can find out from your provider what parameters are required.

Step 3-4: Connect to the online provider

You call the AuthenticateAsync method to connect to the online identity provider and get an access token. The method takes the URI constructed in the previous step as the requestUri parameter, and a URI to which you want the user to be redirected as the callbackUri parameter.

Note: Step 5: Connecting with single sign-on (SSO).

By default, Web authentication broker does not allow cookies to persist. Because of this, even if the app user indicates that they want to stay logged in (for example, by selecting a check box in the provider's login dialog), they will have to login each time they want to access resources for that provider. To login with SSO, your online identity provider must have enabled SSO for Web authentication broker, and your app must call the overload of AuthenticateAsync that does not take a callbackUri parameter. This will allow persisted cookies to be stored by the web authentication broker, so that future authentication calls by the same app will not require repeated sign-in by the user (the user is effectively "logged in" until the access token expires).

Reference: <https://docs.microsoft.com/en-us/windows/uwp/security/web-authentication-broker>

QUESTION 17

HOTSPOT

You are developing a Universal Windows Platform (UWP) app that stores credentials by using the Credential Locker service.

You need to securely retrieve credentials for the current user.

How should you complete the method? To answer, select the appropriate code segment from each list in the answer area

Hot Area:

 **Answer Area**
VCE To PDF - Free Practice Exam

```
private void RetrieveCredentials(string resource, string userName)
{
    var identityStorage = 
    var credentials = 
}
```

new PasswordVault();
KeyCredentialManager.OpenAsync(userName);
CredentialPicker.PickAsync(resource, userName);

identityStorage.GetResults();
identityStorage.Retrieve(resource, userName)

 **VCEplus**
VCE To PDF - Free Practice Exam

Correct Answer:

Answer Area

```
private void RetrieveCredentials(string resource, string userName)
{
    var identityStorage = 
        new PasswordVault();
        KeyCredentialManager.OpenAsync(userName);
        CredentialPicker.PickAsync(resource, userName);

    var credentials = 
        identityStorage.GetResults();
        identityStorage.Retrieve(resource, userName)
}
```

Section: (none)

Explanation

Explanation/Reference:

Box 1:

Example:

```
var vault = new Windows.Security.Credentials.PasswordVault();
```

Box 2:

Example continued:

```
// When there are multiple usernames,
```

```
// retrieve the default username. If one doesn't
```

```
// exist, then display UI to have the user select
```

```
// a default username.
```

```
defaultUserName = GetDefaultUserNameUI();
```

```
credential = vault.Retrieve(resourceName, defaultUserName);
```

Reference: <https://docs.microsoft.com/en-us/windows/uwp/security/credential-locker>

QUESTION 18

DRAG DROP


You are developing a Universal Windows Platform (UWP) app.

A user terminates the app.

You need to ensure that the app initializes when the user starts the app.

How should you complete the method? To answers, drag the appropriate code segment to the correct location or locations. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Select and Place:



VCEplus
VCE To PDF - Free Practice Exam

Code segments

OnLaunched

SearchActivatedEventArgs args

IActivatedEventArgs args

LaunchActivatedEventArgs e

OnActivated

OnSearchActivated

OnShareTargetActivated


ShareTargetActivatedEventArgs args

Answer Area

```
protected async override void
{
    ...
}
```

Code segment

Code segment



VCEplus
VCE To PDF - Free Practice Exam

Correct Answer:

VCEplus Code segments
VCE To PDF - Free Practice Exam

SearchActivatedEventArgs args

IActivatedEventArgs args

OnActivated

OnSearchActivated

OnShareTargetActivated

ShareTargetActivatedEventArgs args

....

VCEplus Answer Area
VCE To PDF - Free Practice Exam

```
protected async override void OnLaunched(  
{  
    ...  
}
```

LaunchActivatedEventArgs e

VCEplus
VCE To PDF - Free Practice Exam

Section: (none)

Explanation

Explanation/Reference:

The OnLaunched method is called when an app is launched. It is passed a LaunchActivatedEventArgs parameter which provides, among other things, the arguments passed to the app, the identifier of the tile that launched the app, and the previous state that the app was in.

Box 1: OnLaunched

Box 2: LaunchActivatedEventArgs e

Reference: <https://docs.microsoft.com/en-us/windows/uwp/launch-resume/app-lifecycle>

QUESTION 19

DRAG DROP

You are developing a Universal Windows Platform (UWP) app. The app runs on multiple device families, including desktop, Windows Phone, and Xbox.

The app must be able to access a user's media playlists if the device supports this feature. If the device does not support this feature, the app must continue to function.