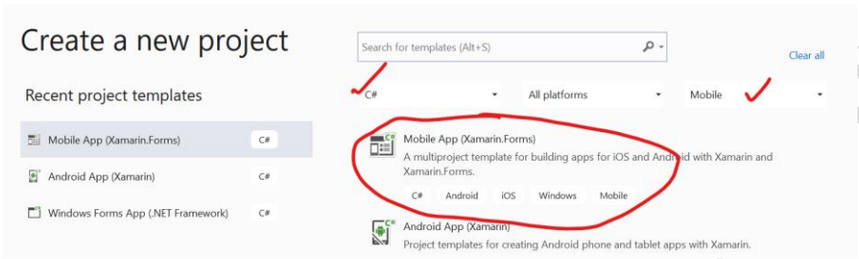
HANDS-ON EXERCISE FOR MID 2

OBJECTIVES:

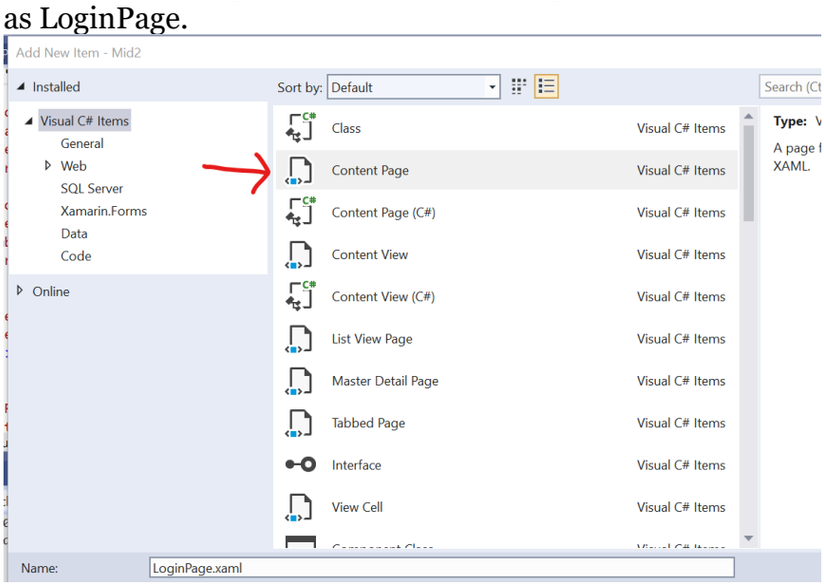
- Learn how to create and navigate between multiple content pages
- Use of popup modal, list view, and web view

DIRECTIONS:

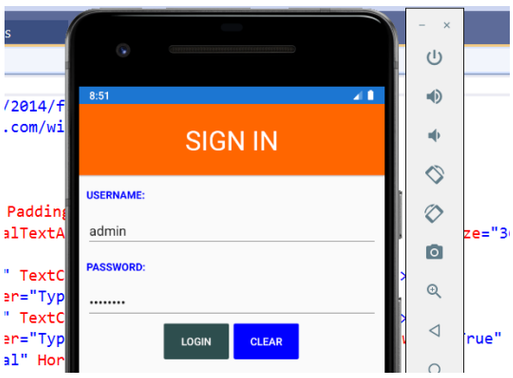
1. Create a new project in Visual Studio 2019 and select Mobile App (Xamarin.Forms) on the template options. **Save it as Mid2_YOURFULLNAME**. Our objective here is to develop a mobile application that can navigate between pages. The application requires to pass through a login authentication first, then it will display the list of the different CICM schools in a list view format. If you click an item on the list view, it will navigate you into a specific page.



2. Select Blank template. Go to Solution Explorer and add a new content page. Name it as LoginPage.



Design this page (similar to the given photo below) to process login authentication from users. Set manually on the page the username as **admin** while password is **Admin123**



Clear button is going to clear the contents of the entries. When the user presses the Login button while the two entries are still empty, you have to display a popup dialog box warning them. A sample is shown below on how to achieve this.

LoginPage.xaml

```
    <Label Text="USERNAME:" Margin="10" TextColor="Blue" FontAttributes="Bold" />
    <Entry x:Name="username" Placeholder="Type your username" Margin="10,0" />
    <Label Text="PASSWORD:" Margin="10" TextColor="Blue" FontAttributes="Bold" />
    <Entry x:Name="password" Placeholder="Type your password" Margin="10,0" IsPassword="True" />
    <StackLayout Orientation="Horizontal" HorizontalOptions="Center">
        <Button Text="LOGIN" Padding="5" BackgroundColor="DarkSlateGray" TextColor="White" Clicked="Login_Clicked"/>
        <Button Text="CLEAR" Padding="5" BackgroundColor="Blue" TextColor="White" Clicked="Clear_Clicked"/>
    </StackLayout>
</StackLayout>
```

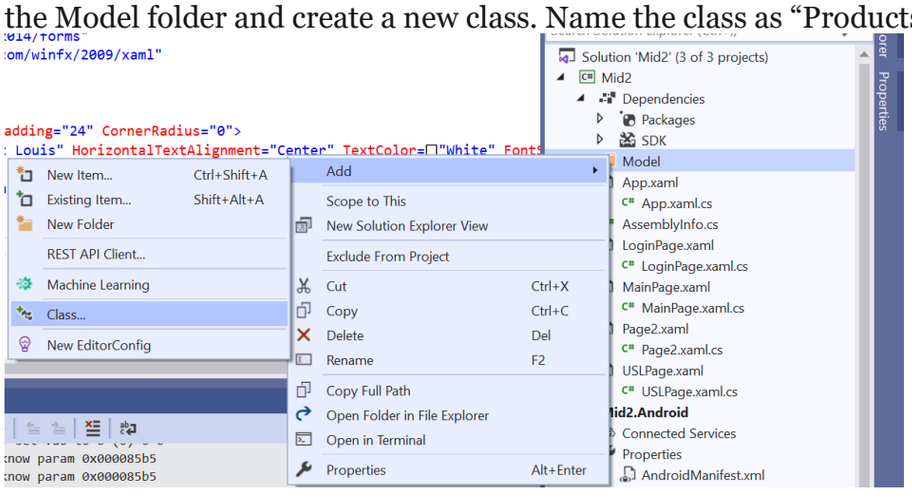
LoginPage.xaml.cs

You must declare the method for login button as “async” so that you can use the popup alert method. Once the username and password are correct, redirect the user to MainPage.xaml by using the PushModalAsync method.

```
async void Login_Clicked(object sender, EventArgs e)
{
    if(username.Text == "" || password.Text == "")
        await DisplayAlert("Alert", "Missing fields required!", "OK"); //for poup
    else if(-- write your condition here --)
        await DisplayAlert("Alert", "Incorrect username or password!", "OK"); //display popup for incorrect username/password
    else
        await Navigation.PushModalAsync(new MainPage()); //navigate to main page when username/password are correct

}
```

3. It’s time to create a list view. Go to Solution Explorer, right click on the project root directory and create a folder called “Model”. Then right click on the Model folder and create a new class. Name the class as “Products.cs”.



4. Open Products.cs and add the following codes and properties of our list view.

```
class Products
{
    5 references
    public string productName { get; set; }
    6 references
    public string productLink { get; set; }
    5 references
    public string productImage { get; set; }
}
```

5. Open MainPage.xaml, delete anything between the <StackLayout> … </StackLayout> and add the following codes for the basic structure of the ListView:

```
<Frame BackgroundColor="#2196F3" Padding="24" CornerRadius="0">
    <Label Text="CICM Schools" HorizontalTextAlignment="Center"
TextColor="White" FontSize="36"/>
</Frame>
<ListView x:Name="lstProducts" HorizontalOptions="StartAndExpand"
VerticalOptions="FillAndExpand"
SeparatorColor="LightGray" SeparatorVisibility="Default" HasUnevenRows="True"
ItemTapped="LstProducts_ItemTapped">
            <ListView.ItemTemplate>
                <DataTemplate>
                    <ViewCell>
                        <StackLayout Orientation="Horizontal" Padding="5"
VerticalOptions="FillAndExpand"
MinimumHeightRequest="100">
                            <Image Source="{Binding productImage}"
WidthRequest="70" HeightRequest="70"/>
                            <StackLayout Orientation="Vertical" Padding="2"
VerticalOptions="Center">
                                <Label Text="{Binding productName}"
FontSize="20" Margin="2" TextColor="Black" FontAttributes="Bold"/>
                                <Label Text="{Binding productLink}"
FontSize="16" Margin="2" TextColor="#eb3443"/>
                            </StackLayout>
                        </StackLayout>
                    </ViewCell>
                </DataTemplate>
            </ListView.ItemTemplate>
</ListView>
```

6. Go to MainPage.xaml.cs and insert the following additional libraries. But make sure that in Line 8, it should not be “using Mid2.Model”. You have to use the actual project name that you have such as “Mid2_YOURFULLNAME.Model”.

```
4   using System.Linq;
5   using System.Text;
6   using System.Threading.Tasks;
7   using Xamarin.Forms;
8   using Mid2.Model;
9   using System.Collections.ObjectModel;
10
11  namespace Mid2
12  {
```

Then add the following lines as highlighted below. Take note that Line 20 is “LSTProducts.ItemSource” but the LST is in lowercase.

```
11  namespace Mid2
12  {
        4 references
13      public partial class MainPage : ContentPage
14      {
15          ObservableCollection<Products> obProducts;
            1 reference
16          public MainPage()
17          {
18              InitializeComponent();
19              AddProducts();
20              lstProducts.ItemsSource = obProducts;
21          }
22
```

Then add the following additional methods:

```
public void AddProducts()
        {
            try
            {
                obProducts = new ObservableCollection<Products>();
                obProducts.Add(new Products
                {
                    productImage = "https://assets2.rappler.com/2021/02/usl-
february-4-2021-1612439834775-546.jpg",
                    productName = "University of Saint Louis (Tuguegarao City)",
                    productLink = "www.usl.edu.ph"
                });

                obProducts.Add(new Products
                {
                    productImage =
"https://64.media.tumblr.com/0a223f6c4a01b3d9d3143723db971ebf/tumblr_mqj1zgbsKP1s
tsh1mo1_500.jpg",
                    productName = "Saint Louis University (Baguio City)",
                    productLink = "www.slu.edu.ph"
                });

                obProducts.Add(new Products
                {
                    productImage = "https://mapio.net/images-p/48243088.jpg",
                    productName = "Saint Mary\'s University (Bayombong, Nueva
Vizcaya)",
                    productLink = "www.smu.edu.ph"
                });

                obProducts.Add(new Products
                {
                    productImage = "https://mapio.net/images-p/25266935.jpg",
                    productName = "Saint Louis College (San Fernando, La Union)",
                    productLink = "www.slc-sflu.edu.ph"
                });

                obProducts.Add(new Products
                {
                    productImage =
"https://puriteens.weebly.com/uploads/5/5/4/3/55438749/1092707.jpg?1435504806",
                    productName = "Saint Louis College (Mandaue City)",
                    productLink = "www.slccebu.edu.ph"
                });

            }
            catch (Exception ex)
            {

            }
}

private async void LstProducts_ItemTapped(object sender, ItemTappedEventArgs e)
```

```
        {
            try
            {
                var selectedProduct = (Products)e.Item;

                switch(selectedProduct.productLink)
                {
                    case "www.usl.edu.ph":
                        await Navigation.PushModalAsync(new USLPage());
                        break;
                    //do the other case for the rest of CICM schools

                }

            }
            catch (Exception ex)
            {
                await DisplayAlert("Error", ex.Message.ToString(), "Ok");
            }
            lstProducts.SelectedItem = null;
}
```

7. Take note that in the last method added in Instruction #6, there is a SWITCH statement. If a user clicked on the list view referring to USL, then we want to navigate into a USLPage.xaml, which you are about to create in Instruction #2. Continue adding the different CICM schools here as mentioned in the ListView. Each item in the list view should have its own page to be redirected with, similar to the one below.

```
private async void LstProducts_ItemTapped(object sender, ItemTappedEventArgs e)
{
    try
    {
        var selectedProduct = (Products)e.Item;
        //await DisplayAlert("You selected", selectedProduct.productName, "Buy", "Cancel");
        switch(selectedProduct.productLink)
        {
            case "www.usl.edu.ph":
                await Navigation.PushModalAsync(new USLPage());
                break;
            //do the other case for the rest of CICM schools

        }

    }
```

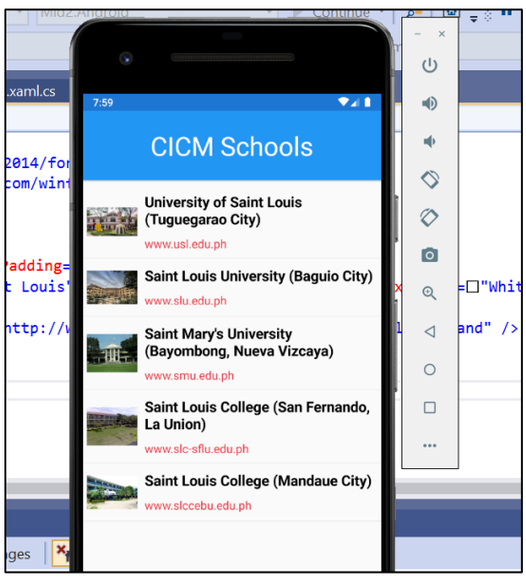
8. Go to Solution Explorer, right click on the project root directory, then add a new content page and name it as USLPage.xaml. Open this page, delete anything between <StackLayout> … </StackLayout> and add the following Frame, Label, and WebView. Do steps in Instructions #7 and #8 for the rest of the CICM schools.

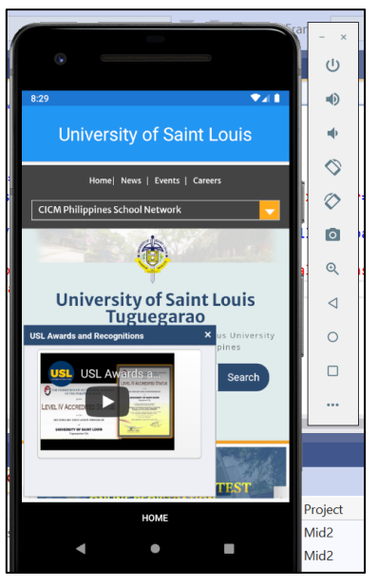
```
<ContentPage.Content>
    <StackLayout>
        <Frame BackgroundColor="#2196F3" Padding="24" CornerRadius="0">
            <Label Text="University of Saint Louis" HorizontalTextAlignment="Center" TextColor="White" FontSize="28"/>
        </Frame>
        <WebView x:Name="webView1" Source="http://www.usl.edu.ph/" VerticalOptions="FillAndExpand" />
        <StackLayout BackgroundColor="Black">
            <Button Clicked ="Home_Clicked" HorizontalOptions="CenterAndExpand" VerticalOptions="EndAndExpand"
                Text="HOME" TextColor="White" BackgroundColor="Black"/>
        </StackLayout>
    </StackLayout>
</ContentPage.Content>
```

Then go to USLPage.xaml.cs, and add the following methods to redirect back to homepage once the user presses the HOME button.

```
public USLPage()
{
    InitializeComponent();
}
0 references
async void Home_Clicked(object sender, EventArgs e)
{
    await Navigation.PushModalAsync(new MainPage());
}
```

9. Once ready, try to run your application. The initial page to appear should be the login page. Once authentication is successful, then it should redirect you to the homepage that shows all CICM schools in a list view format. Try to click on USL list view item, then the screenshot on the right should appear. Try to click HOME at the bottom of USLPage, and it should navigate you back to homepage.





In case the website does not appear, you know that by default mobile apps do not support HTTP protocol. Simply refer to Mid #1 on how to allow this.