Project Part B

Objective

Until the end of the course, you will be working on **Project Part B** individually or together in group. Assignment B1 and Assignment B2 has 1 part each. You may decide to discuss in the group but code individually. The project is individual and should be send in as your solution. Pls send in your Visual Studio solutions, compressed into zip format, so I can easily un-compress them and open in Visual Studio 2019 and run the code. Remember to make a Build -> Clean Solution before making the zip. **Pls see deadline in "Kursplaneringen".**

The objective is to create an Application in Xamarin.Forms that presents the data you acquired in Project Part A. By using the Model (and Service) separation of MVVM, you can simply copy your Service code as well as the Model code from Project Part A that communicates with a web api service using Json. For ease to get started, I made a simple Console, where you can copy, with some small modifications, that Main() from your Program.cs into Xamarin Solution Consoles->Program.cs. Here you can make an initial run of you Part A program within the Part B solution. Then you can start working on the UI and move relevant code into your Views.

For **Assignment B1**, **Weather**, I have provided hints and a simple template in code for you to not get stuck on trivial matters. For **Assignment B2, News,** only some basic data types are provided, and only few hints.

Grade criterias for this part of the project:
Godkänt (G) : Successful completion part of "Project Part B: Assignment B1".
Väl godkänt (VG): Successful completion of "Project Part B: Assignment B1 and B2"

The services are Weather webapi service (https://openweathermap.org/) and News webapi service (https://newsapi.org/) are the same for Project Part A.

Note: For the weather web-api on the free plan you have 60 free web-api calls per minute, and for the news web-api you have 100 calls per day. Similar to Project Part A, for the news web-api, I provide in the template embedded sample data and code that you can use to test and debug before calling for live data.

Project Part B: Assignment B1

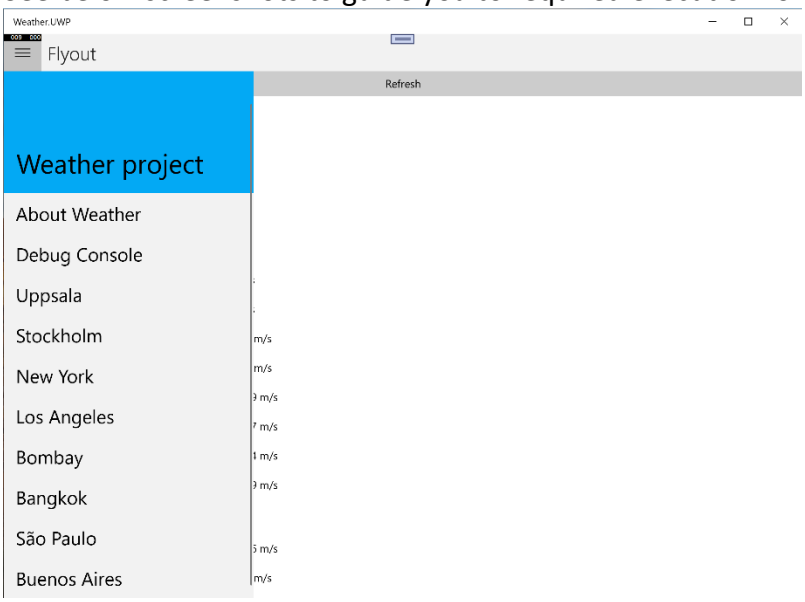
Use the Xamarin Solution Weather template.

Assignment B1

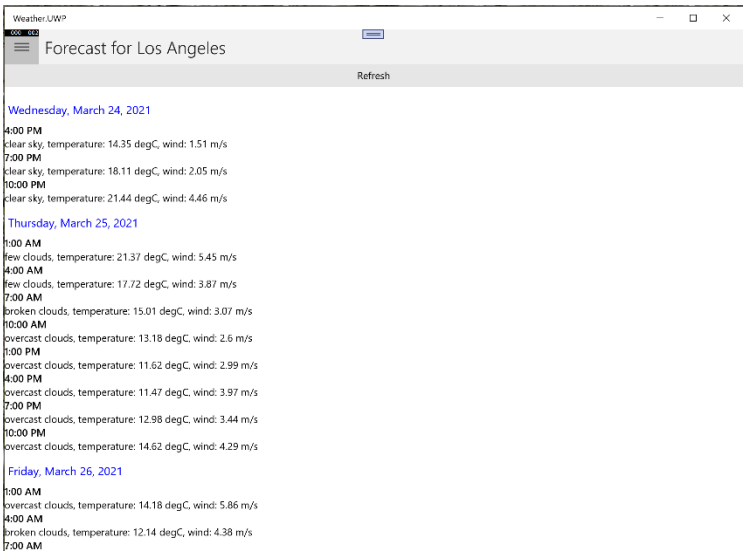
Make a Weather Application using the Flyout navigation pattern that gives a simple UI to the Weather Service by:

- Adding several cities to the Flyout menu
- Presenting the Weather Forecast for the city selected in menu
- Weather Forecast should be presented in the *ForecastPage* in a *ListView*
- The Weather forecast should be shown when the page is visible and refreshed when clicking on a refresh Button
- Pls write you name on the About Page as author

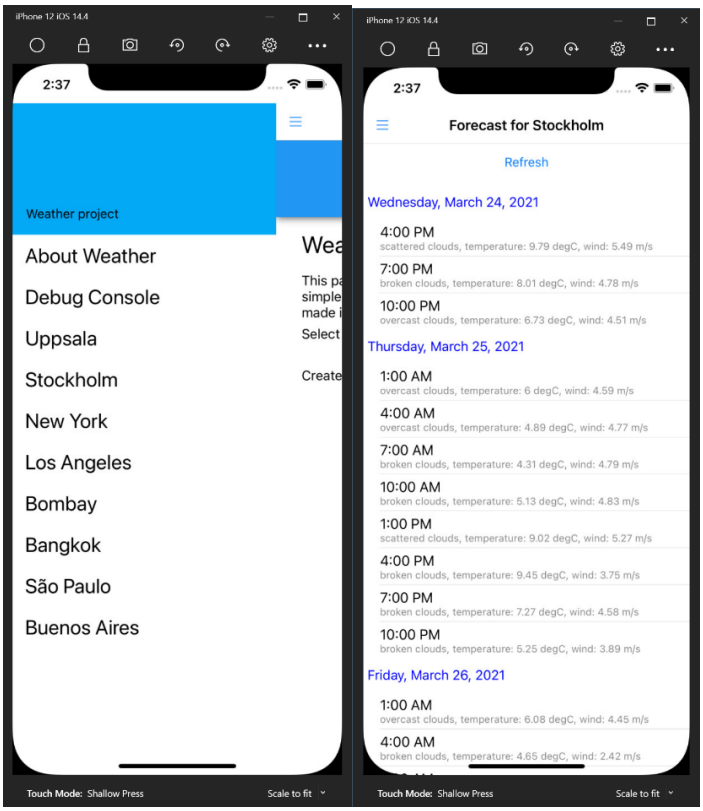
See below screenshots to guide you to required execution for successful accomplishment.



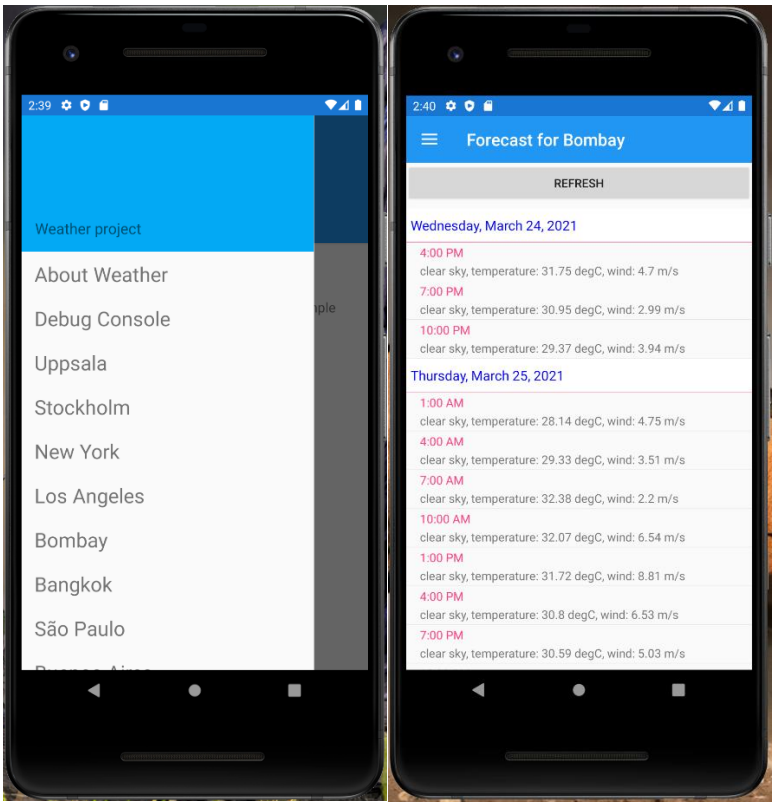
In UWP showing the Cities in the FlyoutMenu



In UWP showing the Forecast for Los Angeles after selecting it in the menu. You also see a refresh Button.



Optional: Screens from iPhone emulator using the same code base



Optional: Screens from Android emulator using the same code base

Project part B: Assignment B2

Use the Xamarin Solution News template.

Note: For the News api you have "only" 100 calls per day. Therefore, I provide *ModelsSampleData* sample data, embedded in the assembly and a Method to read the data asynchronously. With this I am showing a test design pattern that you can use.

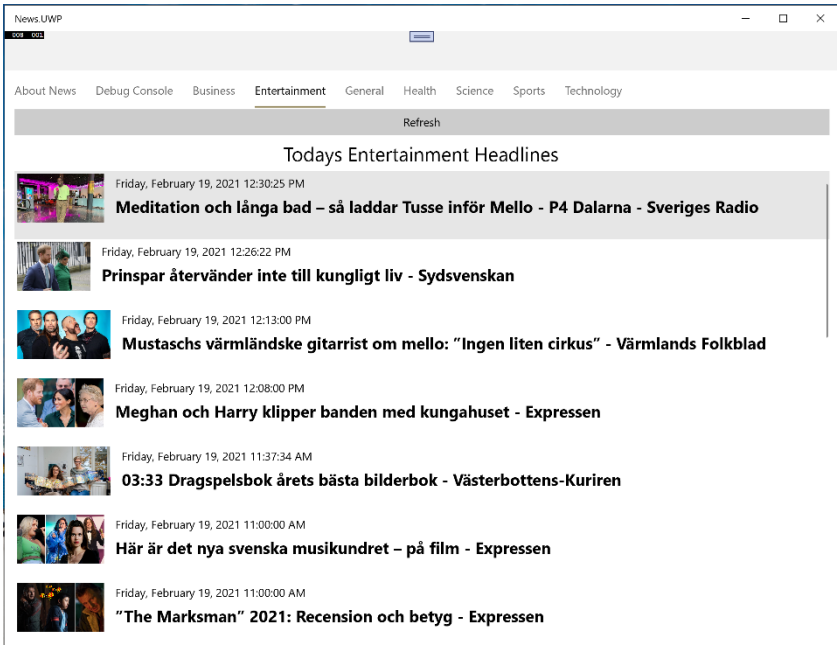
In the Service itself I use the #define structure so you can easily switch to compile for test (one line of code) or live data (your code). You could complete the exercises without reading live data, so pls choose when to go live.

Assignment B2.1

Make a News Application using the Tabbed navigation pattern that gives a simple UI to the News Service by:

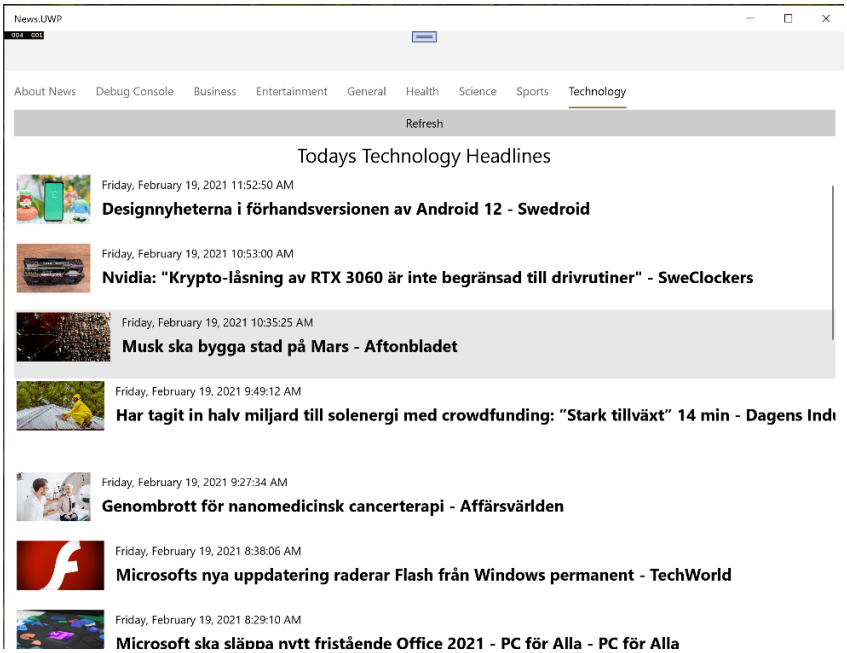
- Adding several News Categories as Tabs
- Presenting the News Headlines for the category selected in the Tab
- News headlines should be presented in the *NewsPage* with a small image, time, headline and description as shown below. Hint. ListView or CollectionView customizations.
- The News headlines should be shown when the page is visible and refreshed when clicking on a refresh Button
- When tapping on a News headline the full news screen should be shown and you should be able to navigate back, using *Navigation.PushAsync*
- Write you name on the About Page as author
- Implement Error handling and show as a modal popup
- Use Activity indicator and simulate a slow network connection

See below screenshots to guide you to required execution for successful accomplishment.



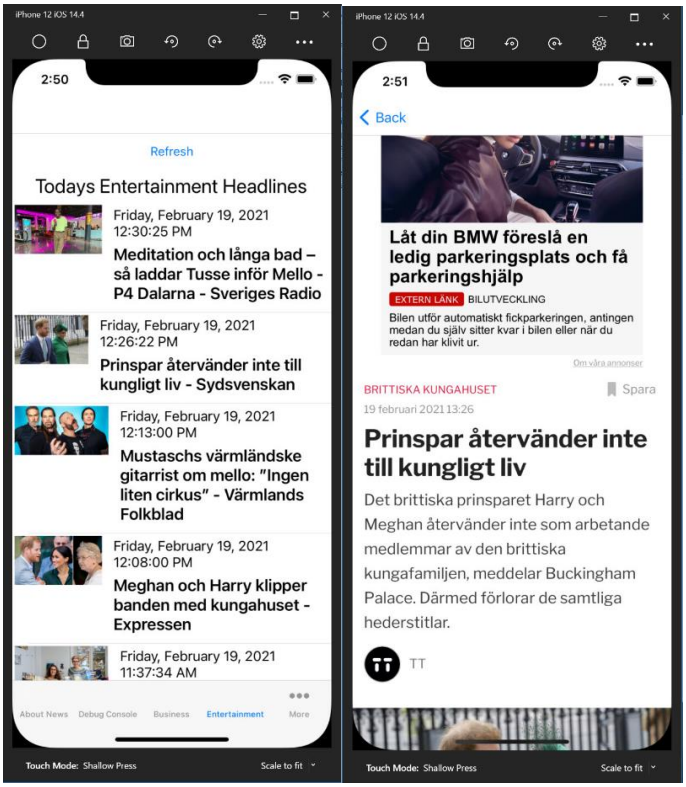
In UWP News headlines in Entertainment category

In UWP after tapping a Headline, showing it in detail

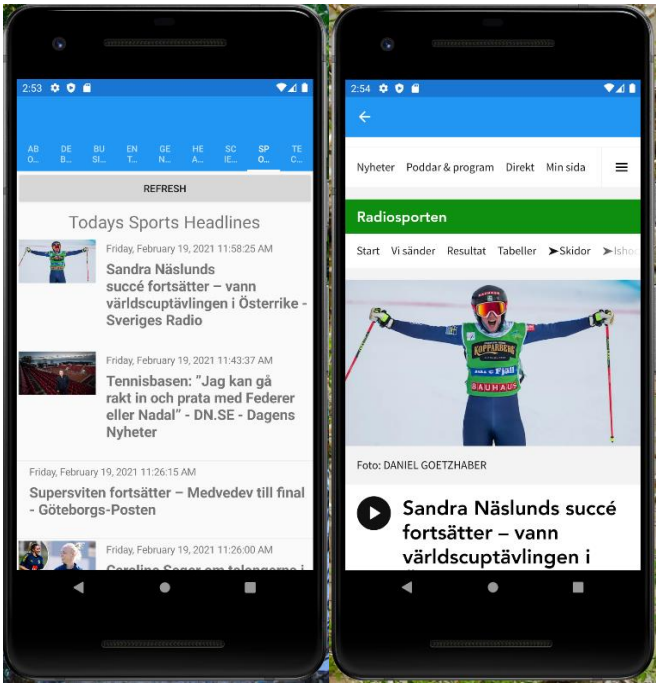


In UWP News headlines in Technology category

In UWP News headlines in Technology category



Optional: Screens from iPhone emulator using the same code base



Optional: Screens from iPhone emulator using the same code base