COMP228 Assignment 2 w/c 21st November 2022

Developing an “New Brighton Murals” App.

Background:

The town of New Brighton[1], on the Wirral peninsula, has, over the last few years, acquired a number of large murals, painted on the side of buildings, at various places near to the town centre.

Your Task:

You will design and develop an application written in Swift 5.7 using UIkit and Storyboard for iPhone 14. The application is designed to assist visitors to New Brighton who wish to see the murals. Your app includes a map and table and uses the user’s current location. The project must be developed using the Git version control system (i.e. you **must** make regular commits to the repository, stored locally in your project. **Note**: Do not move your project to Windows systems without zipping it first. Doing so can lose the hidden Git repository folder).

In order to get information about the murals, a small database with an API has been created. You will need to retrieve data from a web service.

https://cgi.csc.liv.ac.uk/~phil/Teaching/COMP228/nbm/data2.php?class=newbrighton_murals&lastModified=2022-09-15

https://cgi.csc.liv.ac.uk/~phil/Teaching/COMP228/nbm/data2.php?class=newbrighton_murals

Note that:

- The **lastModifed** parameter (shown in green in the first URL above) is optional and may be used to retrieve only data that has changed since the date specified, i.e. supplying **lastModified=2022-09-15** as the parameter will produce data only for murals whose entries in the database have been updated since 15th September 2022. If you do not provide the parameter, then all the data will be returned.
- Images of the murals, referenced in the JSON data are located at the following base URL:

https://cgi.csc.liv.ac.uk/~phil/Teaching/COMP228/nbm_images/
(Note: use secure URLs, otherwise your app will not load the data or images).

- Note that some of the murals have multiple images. You should deal with this in a sensible fashion.
- Each mural has a single thumbnail image, suitable for displaying in a table cell. The URL to the image is part of the JSON data for each mural.
- The mural should be displayed on the map or in the table only If the value of the *enabled* attribute is “1”. If it has any other value, that mural should be ignored (used e.g. when an old mural has been painted over)

Your application is required to have the following basic features **(worth 50%):**

1. The user is initially presented with a map centred on their current location and at a reasonable level of zoom so that nearby roads etc. can be seen clearly. (A user location based in New Brighton is available for you to use in the simulator. A further gpx location file which simulates the user going on a “walk” around part of New Brighton will be made available shortly.) **(worth 15%)**

2. In portrait view, a table below the map contains a list of murals, ordered by distance from the user’s current location (which can, of course, change as the app runs). A location file featuring a “walk” around New Brighton can be used to test this feature. Each cell should contain a thumbnail image of the mural and some textual details (e.g. title and artist). **(worth 15%)**

3. Tapping on an annotation, or a cell, displays an image (or images) and basic information about a specific mural. **(worth 10%)**

4. The App must feel like a good mobile application - no unnecessary popups or alerts etc. Layout of the interface elements, and navigation within the app must be appropriate and aesthetically correct. **(worth 10%)**

[1] https://en.wikipedia.org/wiki/New_Brighton%2C_Merseyside

The remaining **50%** of the marks may be obtained by implementing useful features such as:

1. The user can, from time to time, mark as “favourite”, a mural that they really like (or unfavourite one). This should be displayed in the relevant table cell by displaying some kind of symbol against that cell. The list of favourites should be saved into persistent storage so that it is retrieved every time the app is run. **(worth 5%)**

2. Caching the mural information (including your choice of favourite murals) in Core Data. This would effectively allow the app to run if there was no network connection. Please note: You do not need to cache the mural images. **(worth 10%)**

3. Synchronising the app on startup, checking to see if new or modified data is available from the web service, using the lastModified parameter. **(worth 10%)**

4. Your code should be reasonably well-written. Functions should be used where it helps improve clarity and reduces repetitive code. Data structures should be appropriate and effective. **(worth 10%)**

5. Ensure that your code is appropriately commented and that meaningful class, variable and constant names are used. **(worth 5%)**

6. Your project should include a Git archive, featuring commits that you have made during the whole period of the development process. **(worth 10%)**

If you use any additional images or other materials, ensure that these are copied into the project – not just referenced somewhere else in your filestore. The zipped folder that you submit should include everything required to compile and run your App. You can test this by copying the folder to another Mac (with the same version of Xcode) and trying to run your project on that.

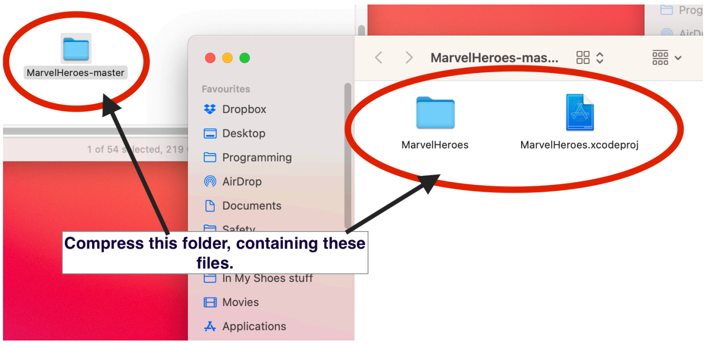
Important - Please note:

Do not use any third-party frameworks in your App (e.g. Alamofire). Use Apple standard frameworks ONLY (i.e. only frameworks provided with Xcode 14). Use of third-party frameworks will involve a **penalty of 25%.**

The App must be written using UIKit and Storyboard. Submissions written using SwiftUI will involve a **penalty of 25%.**

What to Submit

Your completed project (which includes the Git repository, stored automatically in an invisible folder in the project folder) should be zipped up and submitted via the online submission system:
https://sam.csc.liv.ac.uk/COMP/CW_Submissions.pl



Compress this folder, containing these files.

typical example of what to include in your Zip file

(In the Finder, right click the icon for the folder containing the project file and folder and choose “Compress”)
Also submit a short PDF document (maximum of 1-2 sides of A4) which describes how to use your app and any notable features or limitations.

Deadline for submission: Monday December 12th at 12pm (midday)

Reminder: This is the second of two assignments, each of which is worth 15% of the total mark for COMP228. Your portfolio of lab work will be worth another 10%.

Hints:

Explore the API URL on a web browser, copy the JSON data into a pretty printer program - such as https://jsonformatter.org/json-pretty-print

Carefully look at the JSON structure - this will give you information about the Swift structs that you’ll need to create to hold the decoded JSON data. Remember that you’ll need an array of these (similar to the week 9 research papers lab exercise).

You can create a custom table view cell, or use a standard one (your choice). A standard cell *does* contain an image view object, we just haven’t used it so far. One problem with it is that you can’t easily control it’s size etc. If you make a custom cell then you can add an image and set it’s size and other layout options.

To use the gpx fie of a walk around New Brighton, you’ll need to refer to the guidance document “Obtaining the user’s location” which will be published on the canvas site shortly.