Assignment 2   w/c 20th November 2023

Developing a visitor App for Ness Gardens.

Background:

Ness Botanical Gardens is located on the Wirral Peninsula (at 53.27566, -3.04122) and is owned by the University of Liverpool. It features a wide variety of plants gathered from all over the world. Visitors may walk around the gardens exploring the plants and scenery.

The Computer Science department worked with Ness in the 90's to develop a mySQL database to hold it's thousands of plant records. This database does not record the GPS location of individual plants. Instead, plants are allocated to "beds" and it is the geographic locations of these that are recorded in the database table "**beds**".
Plant records in the database are held in a simplified version of the itf2 format (International Transfer Format 2 - an international standard for transferring plant data) in the table named "**plants**",

Your Task:

You will design and develop a portrait orientation application written in Swift 5.7 or later, using UIkit and Storyboard for iPhone 15. The application is designed to assist visitors to Ness Gardens. 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 plants, an API has been created. You will need to retrieve data from a web service (in the following URL's the value for the record number, recnum, is just an example):

https://cgi.csc.liv.ac.uk/~phil/Teaching/COMP228/ness/data.php?class=beds	all bed data
https://cgi.csc.liv.ac.uk/~phil/Teaching/COMP228/ness/data.php?class=plants	all plant data
https://cgi.csc.liv.ac.uk/~phil/Teaching/COMP228/ness/data.php?class=plants&recnum=4041	data on a specific plant
https://cgi.csc.liv.ac.uk/~phil/Teaching/COMP228/ness/data.php?class=images	all image data
https://cgi.csc.liv.ac.uk/~phil/Teaching/COMP228/ness/data.php?class=images&recnum=1129	image data on a specific plant

Note that:

- As can be seen in the example above, some plants have associated images referenced in an "**images**" table. The actual jpeg images referred to in the table are located at the following base URL:

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

- Thumbnail (i.e. small) versions of the images will be available shortly from the following base URL:
  https://cgi.csc.liv.ac.uk/~phil/Teaching/COMP228/ness_thumbnails/

- Some of the plant records have multiple images. You should deal with displaying these in a sensible fashion.
- The plant record is only valid if the accsta field has the value 'C'. If it has any other value, that plant should be ignored (such plants may have originally been present, but subsequently died, or been removed)
- A single plant record in the database, might reference a plant in multiple beds (some plants may be split or propagated and grown as several different instances in different locations). You will need to take the bed attribute and split it into an array of beds (split it based on whitespace - see hint at the end).

Your application is required to have the following basic features **(worth 55%):**

1. The user is initially presented with a map displaying a satellite view, centred on their current location and at a reasonable level of zoom so that features can be seen clearly. (A user location based in Ness Gardens 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 the gardens is planned to be made available shortly.) **(worth 15%)**

2. A table below the map contains plant data, grouped by bed (each of which should be implemented as a different table section) and ordered by distance from the user's current location (which can, of course, change as the app runs). The location file featuring a "walk" around the gardens can be used to test this feature. Each cell should contain a thumbnail image of the plant (if available) and some textual details (e.g.genus, species, infraspecific epithet, vernacular name and cultivar name). **(worth 15%)**

3. Tapping on a cell, displays an image (or images) where these exist in the database and other relevant information about a specific plant (you might wish to use a TextView for this purpose). Plant records that include a valid latitude and longitude of the original plant origin, should include a map showing this. Don't show the map for a plant that is missing this information. **(worth 15%)**

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

The remaining **45%** of the marks may be obtained by implementing useful features such as:

1. The user can, from time to time, mark as "favourite", a plant 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 plant and other information (including your choice of favourite plants) 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 plant images. **(worth 10%)**

3. 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%)**

4. Ensure that your code is appropriately commented and that meaningful class, variable and constant names are used. **(worth 10%)**

5. 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:

This is real plant data, typed into the database by volunteers and so may contain errors (e.g. the bed field might contain references to non-existent beds). You should handle such data gracefully (e.g. if a bed referred to in a plant record, does not exist, then the plant shouldn't be displayed in the table, since that is organised by valid bed names - each of which have a recorded location).

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 15). 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:

[redacted]

(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.

[redacted]

Reminder: This is the second of two assignments, each of which is worth 15% of the total mark for COMP228. Your second part of the portfolio of lab work will be worth another 6%.

Hints:

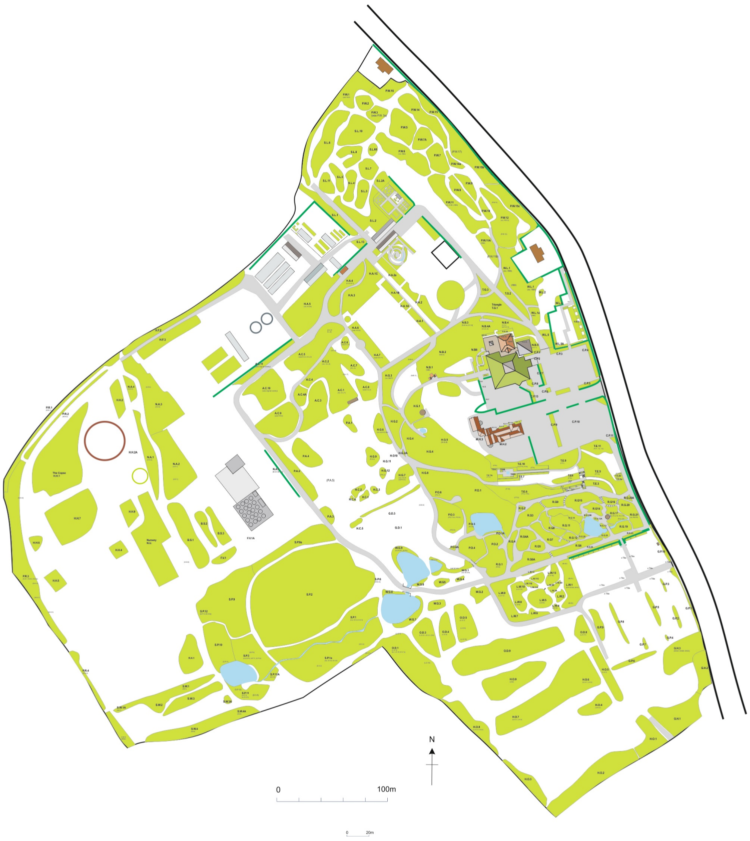
Explore the API URLs on a web browser, copy the JSON data into a pretty printer program - such as https://jsonformatter.org/json-pretty-print. This will show you the structure of the JSON data that you will need to convert into Swift Codable classes (see

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).

To use the gpx fie of a walk around Ness Gardens, you'll need to refer to the guidance document "Obtaining the user's location" which will be published on the canvas site shortly.

Since each plant record's bed field may contain a reference to multiple beds, you'll need to take that bed string and split it into an array of items. Here's an example:

```swift
//This fixed string would actually be data from the plant record
let beds = "PW1 PW2 PW3 PW4"
let bedArray = beds.components(separatedBy: .whitespaces)
//bedArray is now an array of individual bed Strings
```



Ness Gardens map, showing the location of plant beds

Important Tables:

plants	Individual plant records
images	References an individual jpeg image of a plant
beds	Short and full-name and location of a plant bed

The plants table contains the following fields:

recnum	record number (**String, not Integer as originally stated**)
acid	accession ID - plant instance String
accsta	accession status - “C” means a plant that is still alive
family	family name
genus	genus name
species	species name
infraspecific_epithet	optional part of the species name of a plant
vernacular_name	common name for the plant
cultivar_name	name for a plant variation produced by selective breeding
donor	name of the donor who supplied the plant
latitude	latitude of origin
longitude	longitude of origin
country	full name of country of origin
iso	ISO 2-letter code for country of origin
sgu	specific geographical unit in country of origin
loc	locality in country of origin
alt	altitude of plant origin location
cnam	Primary collector’s name
cid	Collector’s identifier
cdat	Collection date
bed	Bed(s) in Ness gardens where plant is located
memoriam	Plant is sponsored in memory of a deceased person
redlist	Plant is endangered
last_modified	Date and time of last change to this plant record