Project 3: The Scriptures, Mapped

Objective and Background

In this assignment, you will practice using table views, web views, and map views in a split view controller framework. This will be a flexible app that works well on any iOS device. The main task of this project is to write an app that lets users navigate through the LDS standard works and see place names that appear in the scriptures on an Apple map. [*No, we will not be doing Book of Mormon geography (!), and some of the Bible place names are a bit speculative (e.g. Mount Ararat), but we will identify place names with modern features having those names. Church history place-names like "Wight's Ferry" (latitude 39.977222, longitude -93.989722) are the best documented. See josephsmithpapers.org for a great list of geocoded Church history place-names.*]

Required Tasks

1. Write an app that lets the user navigate from volumes to books to chapters to scripture content. (**Volumes:** Old Testament, New Testament, Book of Mormon, Doctrine and Covenants, Pearl of Great Price; **books:** Genesis, Exodus, Leviticus, …; **chapters:** Genesis 1, Genesis 2, …; **content:** "Genesis 1. In the beginning…")
2. I will provide a content database in SQLite format that contains metadata and content data for the scriptures, and I'll give you sample code showing how to access the database. All of the information for Required Task #1 must be loaded from the SQLite database, not hard-coded into the app.
3. Use a split view controller to display all the information for Required Task #1 in a master view within your split view. Make sure this works for all types of iOS devices that support iOS 12. (Note that this also means testing it on the iPad Simulator in multitasking mode.)
4. Make sure your app works properly for all orientations, and that it handles auto-rotation properly. (No need to support upside-down portrait on an iPhone.)
5. In the detail view within your split view, display an embedded map. By default, the map could be centered on the user's current location or whatever else you choose. When the user navigates to a page in the scriptures, if there are geocoded locations in that scripture page, (a) clear any previous pins, (b) drop pins on the map showing each geocoded place in the current page, and (c) zoom the map so that the current pins are all visible.
6. When the user taps a geocoded place link in a scripture page, zoom to that place's pin on the map. Each geocoded place in the database will have a zoomed-in view settings configuration, so you just need to adjust the map to match the heading, elevation, and zoom level indicated in the database.
7. Display an appropriate title above the map indicating the selected place name or describing the collection of place names that are currently shown as pins on the map.
8. For the iPhone, you'll need a way to navigate to the map view from the scripture view (because the view controllers don't fit side-by-side except on the iPhone "Plus" devices—8 Plus, XS Max, XR—in landscape mode). Decide how best to provide this feature.

Show your app to at least one other person and get his/her feedback on the user experience you have created. How would they suggest you improve the user experience? I encourage you to share with that person a meaningful scripture-related experience you've had. In your README file in the root of your project folder, include a report on these items.

How to turn this in: rename your top-level project folder "Project 3 Lastname Firstname" using your own last and first names. Then right-click and compress the folder to "Project 3 Lastname Firstname.zip". Upload that zip file to Learning Suite.

Hints

1. When you start this project, select "universal". Don't forget to create a Git repository when you create the project, and *commit your updates frequently* so you have access to a history of changes you've made along the way. This can be surprisingly helpful if you haven't tried it before.
2. Interacting with the database will be an interesting task. I'll provide sample code to help with certain aspects of this, but I'll also ask you to program some of the queries you'll need, along with code to bring DBMS-formatted external data into Swift objects you can use in your iOS project.
3. There will be lots more hints as we talk about the required tasks in class the next couple of weeks.
4. After creating your zip file, trying extracting it to a new folder and verifying (1) that the extracted folder has the right name—Project 3 Lastname Firstname—and (2) the project runs properly when you open it in Xcode.

Evaluation

In all of the assignments, writing quality code that builds without warnings or errors, and then testing the resulting application and iterating until it functions properly is the goal.

Here are the most common reasons assignments are marked down:

- Project does not build.
- Project does not build without warnings.
- One or more Required Tasks was not completed correctly.
- A fundamental concept was not understood.
- Code is sloppy and hard to read (e.g. indentation is not consistent, etc.).
- Your solution is difficult for someone reading the code to understand due to lack of comments, poor variable/method names, poor solution structure, etc.
- UI is a mess. Things should be lined up and appropriately spaced to "look nice." Xcode gives you those dashed blue guidelines so there should be no excuse for things not being lined up, etc. Get in the habit of building aesthetically balanced UIs from the start of this course.
- Assignment was turned in late.

A common question is, "How much commenting of my code do I need to do?" The answer is that your code must be easily and completely understandable by anyone reading it. You can assume that the reader knows the SDK, but should not assume that they already know the (or a) solution to the problem.

Need More of a Challenge?

There are lots of other things you could do to enhance this app, like create a list of place names that are geocoded, and let the user browse those places by name. You could create an interface to allow the user to store a new set of "zoomed in" settings for a given place name. You could implement various app settings (like default font size for the scriptures, or other scripture layout parameters). I'm sure you can imagine more fine features too.