Assignment 1 – iOS Application Development

Due Date: Friday Week 6 (6th September, 11:55 PM) - Weight: 30%
Late Penalty: 5% Per day including weekends

Purpose:

The purpose of this assignment is to demonstrate understanding of how to develop a complex iOS application using common iOS libraries such as CoreData and CoreLocation.

Completion of this assignment demonstrates understanding of the following learning outcomes:

- Utilise techniques for developing mobile applications
- Develop mobile applications

Task:

For this assignment students are tasked with creating a location-aware application to serve as an aid for tourists interested in Melbourne City's historical sights. Users should be able to view information on the various historical locations throughout Melbourne's CBD that have locations associated with them. Additionally, the application must provide notifications to the user when they enter the range of an attraction.

The application must implement the following functionality, with specific controllers having already been identified. You can create additional controllers to facilitate other interactions as required.

Home Screen

- This will be the initial screen of the application
- It must display a map focused and zoomed in on Melbourne's inner CBD with annotations for each historical location stored in the application
  - You should provide a default set of at least 15 sights located around Melbourne.
  - The sights chosen are up to you but you can use https://www.visitvictoria.com/Regions/Melbourne/Things-to-do/History-and-heritage as a guide
  - Each annotation on the map should include a small icon with the picture of the location, the name and a short description.
  - Clicking on an annotation should take the user to a more detailed screen to provide more information
- The Map must be done via MapKit and iOS Location Services
- The user should have the option to add a new historical location to the application.
- The user should have the ability to delete a location. Making sure that removing a location also removes the associated geo-fence
- The user should have the option of displaying all historical sights in a list view

Sight List Screen

- This screen should display a list of all locations currently within the application
- Each sight should be nearly formatted and provide the following information at the minimum
    - Name of Location
    - Short Description
    - Icon
- The user should have the ability to search the list of locations based on name. This search MUST be case sensitive
- The user should have the ability to sort the list alphabetically (A-Z & Z-A)
- Selecting a location should take the user back to the map screen and focus in on the location

Sight Detail Screen

- This screen should display all information about a given location that has been selected
- At a minimum this screen should provide users with the following information
    - Name of Location
    - Description
    - Location (Think carefully about how to handle this)
    - Photo of the location
- The user should have the option to be taken to another screen to edit the location

Add Sight Screen

- This screen should provide the user the ability to add a new historical location to the application
- At a minimum this screen should provide users with the following information
    - Name of Location
    - Description
    - Location (Think carefully about how to handle this)
    - Photo of the location
    - Map Icon (Think carefully about how to handle this)
- This screen should feature full validation to prevent the user from entering invalid data
- Any invalid data entry should provide the user with a sensible error message to correct it
- This screen should allow the user to capture and store a photo of the location. This may require the use of an additional screen

Marking Criteria:

This assessment is worth 30% of your total marks for this unit. Your application will be assessed on the following criteria (100 Marks):

Application Base Functionality – 15 Marks

- Does the application feature a list of all historical locations within the area? (Include at least 10 default locations?
- Can the user select and view more information about a given sight?
- Can the user add new locations?
- Can the user edit locations?
- Can the user delete locations?
- Does the application feature a map for viewing all historical locations with markers on a map?

Location Services – 15 Marks

- Does the application correctly track the user's current location
- Does the application feature geofencing for each historical sight?
- Does the application provide a notification when a particular geo-fence has been entered?

Mobile User Interface – 15 Marks

- Does the application follow the Apple Design Guidelines?
- Does the application include a consistent design? (Cannot just be default)
- Is the application easy to navigate?

Persistent Data Storage – 15 Marks

- Does the application correctly implement CoreData using a proper controller class?
- Is the database structure designed in a sensible manner?
- Is the data stored complete and meet all requirements?

Code Documentation & Data Validation – 10 Marks

- Does the application include full data validation and exception handling?
- Does the application follow a consistent variable naming convention?
- Does the application include appropriate code documentation?

Image Capture & Storage – 15 Marks

- Does the application include the ability to take photos of each location?
- Does the application include sensible defaults for initial locations?
- Are the photos stored appropriately? (Do not store the images directly in CoreData)

Background Notifications – 15 Marks

- Does the application display notifications of entering locations even when in the background of the application?

Submission Requirements:

Students are required to submit the iOS Application via Moodle as a single archive (.ZIP). Please ensure that you test the project on another machine prior to final submission to ensure all files are included correctly. Students should ensure that the archive is named using the following convention [STUDENTNAME]-iOSApplication.

Any code documentation comments must be written in English. Submissions will be accepted for 7 days after the due date. Any submissions beyond this point will not be accepted without prior consultation.

It is critical that you have cited and provided attribution for any external resource used to build your solution. Any external resources that you have used (such as YouTube video tutorials, StackOverflow solutions, etc.) must be cited within your application code. Failure to appropriate attribution or cite materials used to develop your application will be considered as plagiarism and will result in no marks being awarded for the assessment.

Students seeking extensions will have to follow the Special Consideration guidelines outlined here: http://www.monash.edu.au/exams/special-consideration.html (note: you will need to provide adequate documentation to support your claim for an extension).