‘NYUAD Campus Map’ iOS Application
Report

Timeline of the project

1. Familiarizing with the Objective-C language
2. Building the Campus Map using annotations and information windows
3. Developing a more comprehensive application by adding a table view and video controller.

Goals of the project

The project provides a user-friendly interface that allows NYUAD first-time visitors to check the most important places on campus and avoid the confusion of the traditional maps displayed at the building corners. The project was inspired by large amount of visitors that get lost on our small campus, especially those on the lookout for parking lots. The alternative to the lack of such an app is asking security guards or students about directions, which can be confusing and uncomfortable for both parties. The simple download of such an application from Apple Store would make the visitors’ experience a lot simpler and enjoyable.

Learning Outcomes

The learning outcomes of the project are the deep understanding of Objective-C and Swift, as well as debugging and testing of the program. The challenges of writing such a program include coding a number of functions that will provide an user-friendly interface for the user and the connection to the MapKit frameworks.

Description of the functions
Objective-C is a superset of the C programming language and provides object-oriented capabilities and a dynamic runtime. Objective-C inherits the syntax, primitive types, and flow control statements of C and adds syntax for defining classes and methods. In Objective-C, there are three types of files:

- .h files: header files, where the interface and the properties of the class is defined; also, the frameworks and the other classes are imported in the .h file.
- .m files:implementation files, where the rest of the code is implemented

Both .h and .m files are classes.

- .xib files: interface builder files, that show the flow of images, videos and UI view controllers on the screen

List of basic functions

- **AppDelegate (.h, .m):** basic function that sets the setup code and responds to key changes in the code. It determines whether state preservation and restoration should occur and assists in the preservation and restoration process as needed. It responds to events that target the app itself and are not specific to your app's views or view controllers.
- **AnimationViewController (.h, .m, .xib):** class that implements and synthesizes the startup animation whenever the app is launched, by using a UI View Controller. The animation is set for 3 seconds, after which the animated image fades away.
- **Annotation (.h, .m):** class that sets annotations as objects that will be implemented in the MapViewController.m
- **Building (.h, .m):** class that sets the building that will appear on the map as objects that will be implemented in the BuildingStore**.m**
- **BuildingStore (.h, .m):** class that implements the buildings, as objects that have a number of attributes: name, image, description.
- **InformationViewController (.h, .m, .xib):** class that redirects the user to a menu where they could see more information about the building.
- **MapViewController (.h, .m, .xib):** main class that defines the callout windows, sets the buttons at the bottom of the screen and assigns the buildings to the corresponding latitude and longitude on the map
- **TableViewController (.h, .m, .xib):** class that implements the view controller for the buttons at the bottom of the screen: Map, Buildings and Video
- **VideoViewController (.h, .m, .xib):** class that sets the interface for the NYUAD presentation video

List of functions to be implemented

- **Employee (.h, .m):** class that ideally, sets the list of attributes for each employee (objest)
- **EmployeeInformationViewController (.h, .m.,xib+CustomCellViewController.xib):** class that will display a list of employees per building, by presenting their profile (name, education, contact data) and their affiliation with the building
- **ImageStore (.h, .m):** class that will assign a photo to each employee.

More details can be found in the comments in the .xproj file.