Speedway Motors Senior Design Transition Checklist and Guide

**Accessing the Repository**

To access the repository and to run the application within xCode, open up the repository on github, and make sure the master branch is selected. Once there, click on “clone or download” and select either “Open in xCode” or “Download Zip”. Choose the directory that you would like to hold the application files, just make sure it is easily accessible via the terminal, as that will be required to install. Once this step is complete, open up the terminal and navigate to the main project directory. For example if you save the files as “speedwayMaster” navigate to that directory. Once there, run the commands:

$pod deintegrate

$pod install

This will remove any remaining pod frameworks in the project and install the versions specified in the PodFile. This is very important because the application will not run if this step is not taken. Next, navigate to the directory in finder and open the just created “speedway.xcworkspace” file. There will be 2 of these files within the directory. This will open the project in xCode and at this point you should be able to run the project on the simulator or on a phone. If errors persist when running the application, consult the Installation & Deployment Guide, as that file contains several fixes for problems we run into when trying to install. This will need to be completed each time a new version of the application is downloaded, as the pods need to be installed separately. But once this is complete, the project files can be pushed directly to the new repository.

**ZenHub**

Attached to our repository we have a board which tracks issues and tasks for the project. The current issues can be viewed within the Github, but to alter them, simply go to ZenHub.com and log in using Github credentials, and the board is attached to cse senior design. Each of the tasks have a description attached to them, as well as several other key pieces of information. The backlog is currently populated with tasks that we weren’t able to get to or have identified as post-project, like the API implementation.

**Testing**

To test the application, there are two options for testing. The first is to use a simulator within xCode with any device, or to attach it to a physical phone. To use a simulator, xCode already contains the functionality to allow that to happen, so you select the device from the menu in the top left corner next to the target. Any iPhone device will work. To download it onto a phone, the phone will need to be connected to the laptop via the lightning cable, and will need to be trusted by the computer. Once this happens, try to “run” the application on the device, selected from the drop down menu. If it fails, on the phone go to settings -> General -> Profiles -> and trust the developer, and run the application again.

**User Training Requirements**

There is no training needed from the user's perspective in order to be able to use application. With the current layout of the application, the only knowledge the user will need is the login information, but this will subside once the API to the Speedway Servers has been implemented.

**ZenDesk Notes**

The training that is required for the person who will take over the code is to be familiar with Xcode environment and understanding how Zendesk SDK works. Within the context of the services provided, the keys for the ZenDesk Live Chat and FAQ will need to be changed. To do this, all of the information is in the AppDelegate.Swift file, and four pieces of information are needed:

1. The live chat key is within the first ZDCChatInitialize call in the file. This will be the same key that is currently in use for the website via Zopim.
2. The FAQ implementation has three keys, an AppID, ClientID, and URL, which is housed in the second ZDCChatInitialize call. All of these are displayed on the website that holds the FAQ information, though this can be changed by creating a new URL.

**PodFile Specification**

For all of the cocoapod instances installed to the project, are all specified within the PodFile under the “Pods” project. Here, we have specified specific versions for GoogleMaps, SwiftyJSON, and ZenDeskSDK. This was done because had they been the most updated versions of each, we’d have to change the code to be Swift 5 compliant, and we didn’t have enough time to complete this. Other than that, the 7 pods are all specified to the main project target, and all are required to run the project, but these are integrated during the pod installation progress. In the future, this change will need to be implemented, and once it is, simply drop the “, ‘version’” in the PodFile specification and run pod deintegrate and pod install again. In addition, the current base OS for the application is iOS 12.0, which can be updated at any point to reflect the current version of iOS.

**Google Maps And Places**

Similar to the ZenDesk portion, an API key will need to be provided within the AppDelegate.Swift file. This is just above the call for ZDCInitialize and is housed in the GMSPlacesClient and GMSServices calls. These keys are available through the Google Maps API online, but we have provided our own for the purpose of this course. A proprietary key may be required, but in its current state that isn’t needed.

**SwiftyJSON**

This is a cocoapod that is installed during the pod installation process that allows the application to handle JSON requests and files. No major changes are going to be needed on the end of the developer, but if installing and running a new project and an error comes up saying that SwiftyJSON is not valid to run in Swift 5, go to the pods project, select SwiftyJSON, select Build Settings, and scroll down to Swift version. Here it should say “Swift 5 (unsupported)”. Select it and change it to Swift 4.2, clean using command + shift + k and run the project again.

**Miscellaneous Notes:**

In general, once the project files are correctly installed on xCode, the application should not run into any problems when trying to run, given the device fulfills the requirements specified. In general, it is good practice to run the application via the simulator before attempting to run it on a phone, just because the deployment process is much more convoluted for running it on a physical device. If an unknown error appears when running the application, consult the Installation & Deployment Guide as towards the end of that document it contains several common errors and fixes that we’ve run into over the course of the development cycle.

**Supporting Artifacts**

* Installation & Deployment Guide
* Transition Guide