**iOS Application – Speedway Motors**

|  |  |
| --- | --- |
| PREPARED BY: | Derek Vogel |
| DATE: | 3-26-19 |

**Introduction**

**Purpose**

The purpose of this document is to provide the reader with ample knowledge on how to successfully run and install the Speedway Motors iOS Application. Prior to installing, the user must get xCode in order to run the application, which is only available on OSX devices. Then the project directory will need to be opened in xCode. One this is complete, the pod library will need to be installed next, or if pods already exist, be deintegrated, cleaned, and reinstalled. Once this is completed, and the project has been cleaned, the application can now be run on either an iPhone, or a simulator within xCode. However some extra steps are required for installing it on a physical iPhone.

**Intended Audience and Reading Suggestion**

The intended audience for this document is the Development Manager at Speedway Motors. The ultimate goal of this hand off and this document is to provide instructions and tutorials on how to make this app usable post handoff.

**References**

|  |  |  |
| --- | --- | --- |
| Reference # | Document Link | Author |
| REF – 1 | Swift Tutorial - [link](https://www.raywenderlich.com/6338-swift-tutorial-part-1-expressions-variables-and-constants) | Ray Wanderlich |
| REF – 2 | Pod Installation Guide - [link](https://guides.cocoapods.org/using/getting-started.html) | Cocoapods |
| REF - 3 | ZenDesk Tutorial – [link](https://support.zendesk.com/hc/en-us/articles/203921213-Introduction#/hc/en-us/articles/203691476) | ZenDesk |

**Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| Term | Definition |
| Pods | Pods, or Cocoapods as they are commonly referred to as, are external APIs that can be installed and integrated into Swift applications. |
| Swift | Swift is the programming language used to make iOS apps. It is fairly similar to C and Python and was derived from Objective C. |
| Info.Plist | This is a file within the xCode project that contains several different flags, such as ask for user location, and privacy settings. |
| Storyboard | The main layer of the app that manages the UI. Typically made up of segues and viewcontrollers. |
| ViewController | This is the basic building block for the UI portion of the application. Each viewcontroller typically has a Swift file associated with it. |

**Assumptions**

|  |  |
| --- | --- |
| Term | Definition |
| Pod Version | It is assumed that the individual pods are the most recent version, and the cocoapods framework is the most updated, which in this case is version 1.6.1 as of the time this document was created |
| iOS | While the impact is minimal, it is assumed that all phones have iOS 11.0 and later |
| ZenDesk | This has already been discussed, but for all of the functionality of the application to work, it needs to have a ZenDesk account attached to the live chat and FAQ URL. |

**Architecture and System Overview**



**Figure 1 Proposed Arhchitecture**

**System Requirements**

In general, the main system requirements are xCode, currently in xCode 10.1, and installation of the cocoapods, currently in version 1.6.1. Once the cocoapods are installed, the project pods are also required which are ZenDesk, GoogleMaps, GooglePlaces, SwiftyJSON and ImageSlideshow, all of which are specified in the podfile in the project space. The licensing for all of the pods outside of ZenDesk doesn’t require licensing outside of API keys that are available from the respective websites, mostly for Google Maps and Google Places. ZenDesk requires two licenses, one of which is currently in use for Speedway systems. The first is the live chat which is currently being used by Speedway Motors, and the keys will need to be altered in the app delegate. The second is configuration and access to the support functionality in ZenDesk, which is currently attached to one of the staff member’s emails.

**Software Installation Instructions**

**Server 1 Middle Tier**

Installation of this product is supported on the following systems and versions:

* MacOS Mojave
* xCode 10.0
* Swift 4.2

**Roles, Features and Packages**

The following software packages much be installed on the operating system prior to installation of the software:

* Cocoapods Version 1.6.1 or 1.6.2
* Pods that include:
  + Google Maps
  + Google Places
  + ZenDesk
  + SwiftyJSON
  + ImageSlideshow

**Configuration**

|  |  |
| --- | --- |
| Config Item | Value |
| ZenDesk Support App ID | 1d9e65ef555180159744b2dc1c6719bfe9c418b0267541b9 |
| ZenDesk Support client ID | mobile\_sdk\_client\_b3e199f185f952237808 |
| ZenDesk Support URL | https://cseseniordesign.zendesk.com |

**Installation Steps**

1. Close xCode project
2. Open the terminal
3. Navigate to the project space
4. Use the command $sudo gem install cocoapods
5. Use the command $pod install
   1. If there already is a pod version on the system, use the command $pod update instead
6. Reopen the project and clean the project using command + shift + k

**Mobile Tier**

**Roles, Features and Packages**

Installation and deployment of the mobile system requires these versions:

* iOS 11.0 or later
* xCode 10.0

**Installation Steps**

1. Once xCode is installed, the cocoapods will need to installed next to run the application and all of the external libraries the application calls
2. First, open up the terminal, and navigate to the project
3. Then run the command $sudo gem install cocoapods
4. Once this is complete the basic cocoapods framework is set up and the individual pods will need to be installed. To do this, use the command $pod install
   1. If a prior version of the pods have already be installed, the command $pod update will then be used to make sure they are all using the most updated version
5. Once this is complete, open the project and clean the project using command + shift + k
6. Run the application. If errors persist, follow the directions for the specified error below.
7. If running on a physical phone, the phone will need to trust the developer of the application within settings. To do this, on the phone, go to settings -> general -> device management -> developer name -> trust. Once this is done, run the application again and it should install correctly.

**Testing the Installation**

1. Select the device that the app will be run on, whether a simulator or a physical device
2. Press “Run” in xCode
3. One the application loads up, use one of the five logins from the login.json file and log in
4. To test the home page, simply tap on one of the links to test that the links are functioning
5. Tab over to live chat, tap start a live chat, and you should be able to either send a message or the error for no agents will occur, depending on if the ZenDesk key has been swapped out or not.
   1. If the key has been changed, send a message through the application and view it on the employee side. The employee should receive a message and a name attached to that chat.
6. To test the FAQ, tab over to the FAQ. Once this is done after a moment it should be connected to the FAQ website where the user can view different articles associated with Speedway.
7. How to test the Curbside Pickup:
   1. If logged in to one of the accounts in the login.json file, that account will have an active order attached to it.
   2. Tap Pick Up Order on the Curbside Menu
   3. On the order detail page, it will display several different data fields containing information about that order. Confirm that the page has viable information rather than just text that says “Label”
   4. Tap on Pick up Order
   5. On the next page it will prompt the user to accept or deny location privileges
      1. If the user accepts, their location should then appear on the map along with a geolocation circle around the Speedway location. If the user is within the circle location wise, they can tap on pick up order and follow the two prompts to be brought to an order confirmation screen.
      2. If the user accepts and is not in the geolocation radius, once the user taps on pick up order, they should be given an error message saying they have to be in the circle and not be allowed to continue.
      3. If the user does not have location services enabled, the flow will be the same as subpoint i.

**Troubleshooting**

**Common Installation Issues**

**Error: Cannot find pod “SwiftyJSON (or other)”**

1. Exit out of xCode
2. Open up Terminal
3. Navigate to project directory
4. Use the following commands:
   1. $pod deintegrate
   2. $pod clean
   3. $pod install
5. Reopen project
6. Use Command + Shift + K to clean the project
7. Run it on the associated device or simulator

**Error: Cannot compile non-modular libraries**

1. Open project navigator

2. Select project target

3. Select Build Settings

4. Press the “+” sign and select “Add user-defined setting”

5. Set setting name to “SWIFT\_ENABLE\_BATCH\_MODE”

6. Set the key to “NO”

7. Clean and run the project

**Error: NSUnknownKeyException**

1. Open up the Swift file where the error occurs
2. Determine if one of the IBOutlet names is outdated or the name has been changed
3. Go to Storyboard
4. Select the component, usually a label that is causing the error
5. Command + Click the component
6. Delete the old link to the Swift file
7. Delete the link in the Swift file to that component
8. Establish a new IBOutlet to the Swift file. Do not change the name of the outlet as that will cause this error again.

**Error: App crashes after attempting Segue**

1. Open up Storyboard
2. Select the Segue that is being called within code that causes the crash
3. Open the Segue’s attributes inspector
4. Confirm and copy the Segue identifier, and if there is none, give it a unique identifier
5. Open the Swift file associated with the previous ViewController, so the file that calls the segue
6. Look for the “PerformSegue” call within the file
7. Replace the identifier with that of the storyboard ID
8. Compile and run the application
9. If the Swift file and Segue do not call eachother, instantiate the PerformSegue function within the code and add an identifier to the Storyboard.

**Known Issues**

|  |  |
| --- | --- |
| Known Issue Title | Description / Recommended Action |
| Live cbat Concurrent Chats | As of right now, the application only allows for one user to chat with the help desk. However, once the application is transferred, the login key for the live chat will need to be changed to Speedway’s proprietary key. |
| Data Calls | While the data is structured the same as it would be with a system API call, it does not currently use any data from the Speedway system and an API call will need to be implemented to fix this. |
|  |  |
|  |  |
|  |  |