# Introduction

The target user of this application is the average citizen who may not know much the EAB so the data schema needed to informative for decision makers and intuitive for the user.

## Mobile Application

The mobile application is under development for both the iOS and Android platforms. In order to leverage ArcGIS technology in native applications, the ArcGIS Runtime Software Development Kits (SDKs) were used. The ArcGIS Runtime is based on a small, high performance C++ Runtime Core, which has its own geometry engine. The ArcGIS Runtime allows for a faster display via GPU Acceleration with OpenGL or DirectX, depending on which platform is being used. ESRI provides Runtime SDKs to support many platforms in the desktop and mobile environments, and each Application Programming Interface (API) has its own bindings to the Runtime Core (ESRI 2015). The application for this project has been built separately for iOS and Android using the ArcGIS Runtime SDK for iOS and ArcGIS Runtime SDK for Android, respectively.

The Android and iOS platforms were chosen because collectively they make up 93.95% of the mobile market share as of June 2016, with Android taking 70.85% and iOS taking 23.1% (Epstein 2016). Despite the application being developed for two different platforms, the capabilities and functionality will accomplish the same goals: 1) incorporate a map view where new sightings of EAB are reported using the device GPS and stored in an ESRI Geodatabase, 2) allow users to identify/view sightings and Web GIS Services provided by the MDA for traps, biocontrols, and confirmed sightings, 3) provide the user with information on how to identify Ash trees and the EAB, 4) send an email to the the MDA when a new sighting has been logged, and 5) send out push notifications to everyone who has the app when a new sighting has been logged. When the application is finished, it will be distributed in the Apple Store and Google Play Store so it can be freely downloaded.

# Methods

## GIS Data

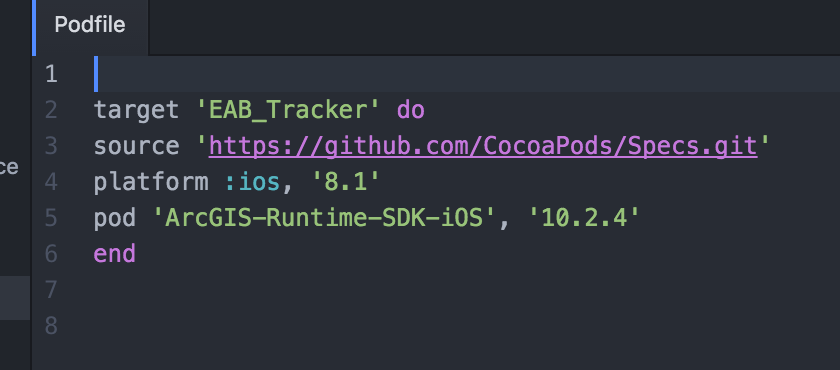
A file geodatabase was set up with a point feature class to track individual sightings and a polygon feature class to track counties with their respected quarantine status. Attribute domains were created with helpful description aliases so that even users unfamiliar with the EAB can record detailed information about the sighting.

The attribute domains allow the user to describe the type of sighting such as adult beetle, visible larvae, D-shaped exit holes, and S-shaped galleries. Another field tracks information about any visible tree stress such as crown dieback, epicormic shoots, and bark splitting. For purposes of checking data integrity and reliability of the user, another field was added to get a sense of the user’s knowledge level of the EAB with options such as that user being a biologist or entomologist, environmental scientist or similar, or not having much knowledge of the beetle. Other fields that user could fill in were site descriptions, comments, and their name and email. Other attributes are automatically populated by logic built into the application to report the latitude and longitude as well as an address provided by reverse geocoding the input point.

The county polygons are much simpler with the only important attribute other than the county name being the current quarantine status. There are three options for the application: currently quarantined, no positive sightings, and needs to be considered for quarantine. The sighting points and county data was published to ArcGIS Online as a hosted Feature Service with editing capabilities and shared as a public service.

## iOS Application

The iOS application was developed using the new Swift programming language created by Apple. Swift offers many benefits over the traditional Apple development language of Objective-C with features such as automatic memory management, dynamic library support, and optional unwrapping. All Swift coding was done within the Xcode Integrated Development Environment (IDE). The first step in this application was to enable the ArcGIS functionality required for the app. Because the ArcGIS Runtime SDK is a third party library provided for ESRI, it needed to be referenced in the iOS app as a dependency. This was added to the EAB Tracker app project by using CocoaPods, which is a dependency manager for Swift and Objective-C projects. A podfile was set up and used to install the ArcGIS API version 10.2.4 locally within the project so it could be compiled with the application’s source code (Figure X).



There are also some application security settings that needed to be managed within the project as well. Because this application will be working with ArcGIS Web Services, the device will need to have appropriate permissions set to make web requests. Exception domains for accessing ArcGIS Online and the Google Maps Geocoding API were added to the application’s information properly list. Also stored in the information property list is the prompt to ask the user if they will allow this application to have access to their location services. The final permission necessary for the application is the use of the device camera, which the prompt text was set in the source code.