Final Report

Next Generation Mobile App for eFRI Ground Data Collection (KTTD 6A-2018)

April 19, 2020

ArborData Consulting Ltd.
166 Foxhunt Rd
Waterloo, ON N2K2S2
PH : 226.792.7297
www.arbordata.ca
craig.robinson@arbordata.ca

ArborData consulting ltd.

data... easy

1 Introduction

The Next Generation eFRI Ground Data Collection App is the development of a mobile field application for eFRI Ground Calibration Plots. The application is being developed in Xamarin.Forms as an open source app that will be widely available for anyone to take the core development and expand upon it, or deploy it for an eFRI field data collection program. The app is being deployed in Android and iOS and could also be deployed in UWP if the need arises.

This final report outlines the work completed on the application. Work started slow on the project since the provincial eFRI program evolved slowly into the T2 round of forest inventories. The application being developed here actually looks very little like what was initially envisioned since the eFRI ground plot program is now completely different than what has been done previously in the T1 inventories. This application is designed to allow for the field data collection of fixed area plots that are collected for the calibration of LiDAR data in an enhanced inventory program. This final version focusses on Type A and Type AB and Type AC and ABC plots that collect information on plots, trees, small trees, soils and ecosites, deformities, understorey vegetation and down woody debris.

2 Data Model

In summer 2019, a pilot project in the Romeo Malette forest started the ground data collection of fixed area plots that will be used as calibration in an enhanced Area-Based Lidar inventory. That project finally revealed some of the data types and attributes to be collected, and a data model to suit those plots was assembled in collaboration with Geordie Robere-McGugan that could accommodate the data collected in 2019. All of the development of eLiDAR is based on the plots and data of the 2019 field data collection program.

The database table structure below was created in anticipation of accommodating Type A. AB, AC and ABC plots. In eLidar, this table structure is created in SQLite on the device, to allow for field data collection while disconnected in the field. The same structure is implemented in the cloud in an Azure SQL database. Each app can synchronize its data with the cloud database via an Azure REST API and JSON.

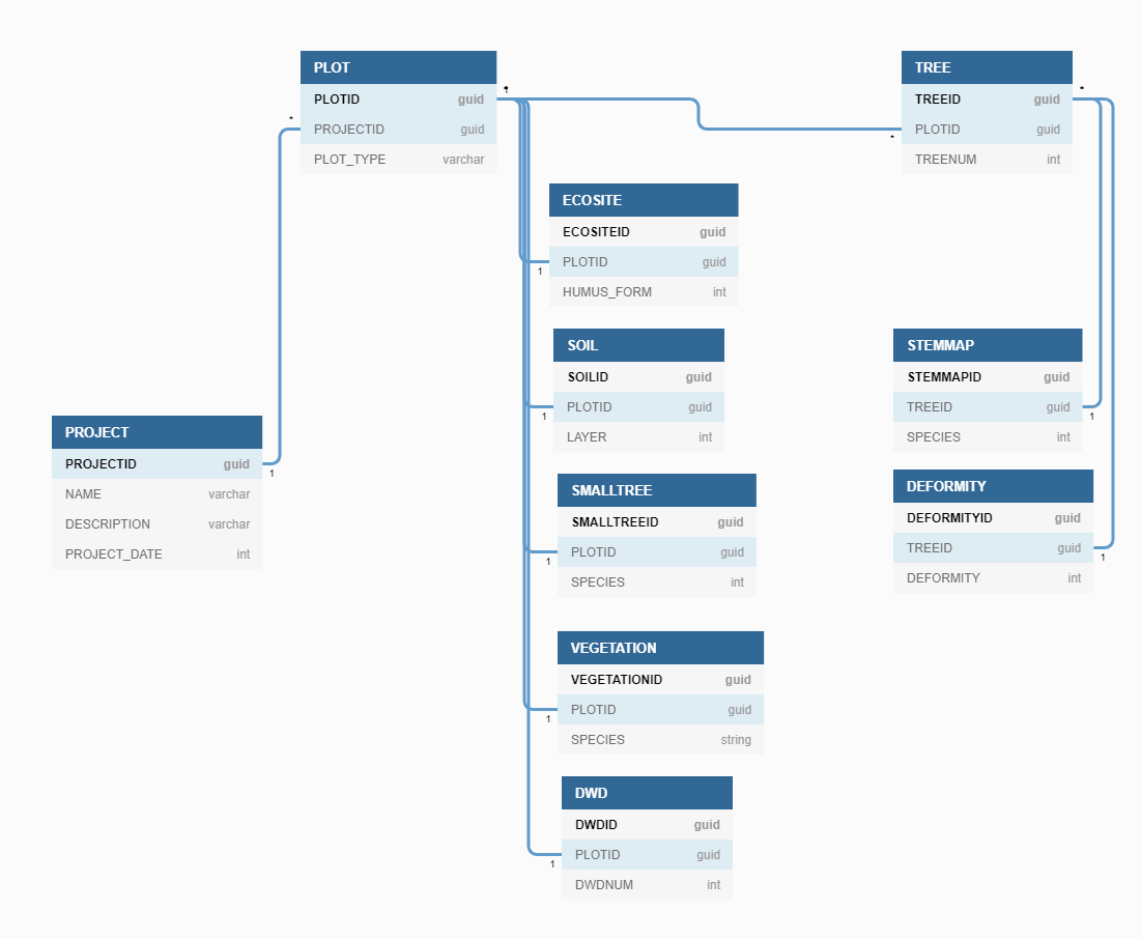

Figure 1 Table structure for eLiDAR, used in SQLite on the device, in iOS and Android. Globally Unique Identifiers (GUIDs) are used as primary keys on all tables to allow for data in all tables to always have unique rows, regardless of where the data originated. All rows are timestamped at create and update to allow data to be synchronized between devices and the cloud.

3 Version 1 (white pine) of eLiDAR

The eLiDAR App in Xamarain.Forms is now functional as a working version and works disconnected in order to collect all of the data and attributes necessary. This working version is ready for download and testing by others (there are links provided in Further Advancements on page 17 to an Android version). Feedback will be used to further shape the function and flow of the application. Below are screenshots from the Android 9.0 version of the application. Descriptions are below each screenshot that describe the function and flow of the app.

3.1 Screen Shots

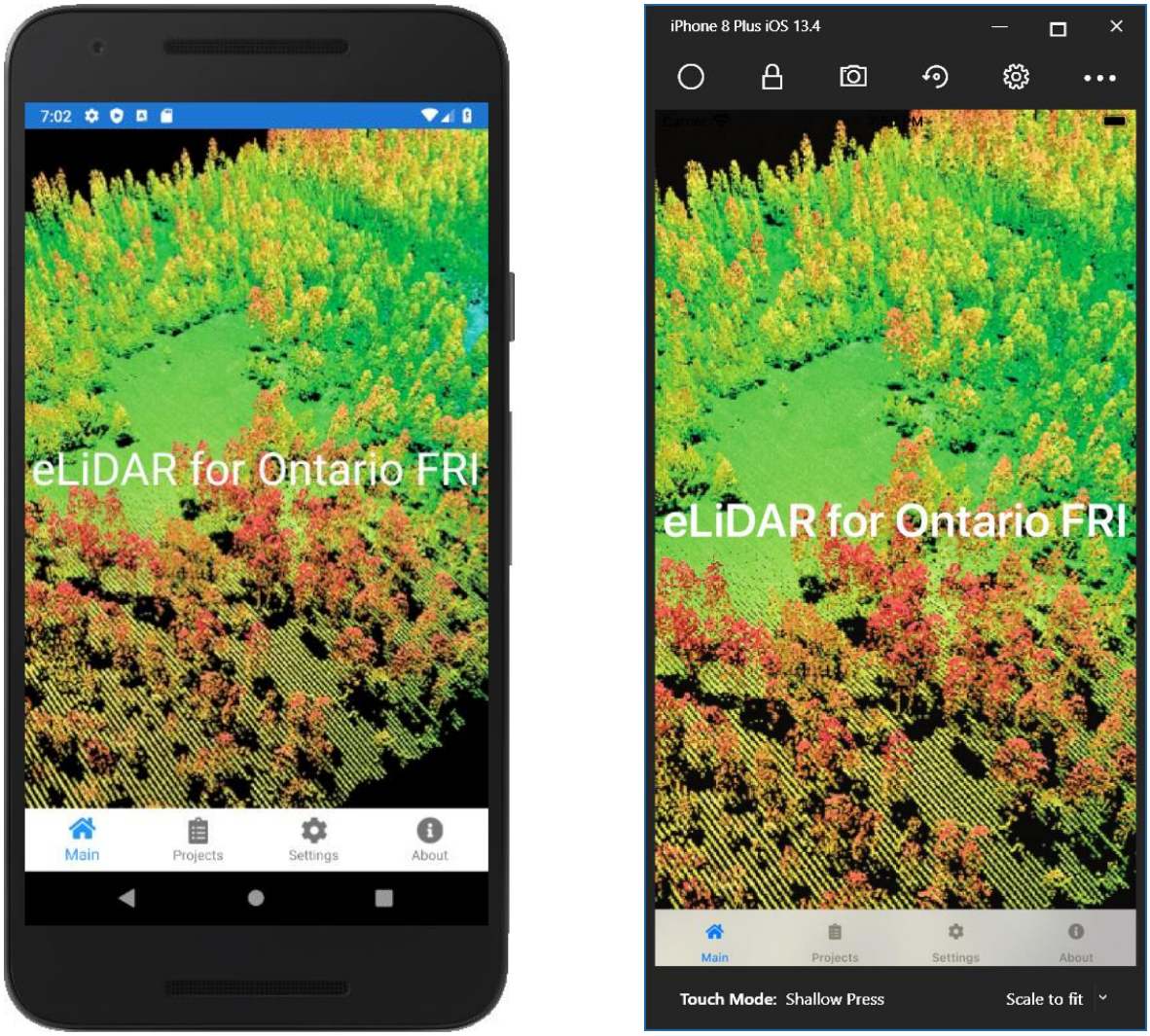

Figure 2 Main screen of eLiDAR. From here a user can navigate to the project or plot screen to begin their data entry sessions. All data entry starts from the Projects tab. On the right is the application on an iOS device. On the left is an Android 9.0 device.

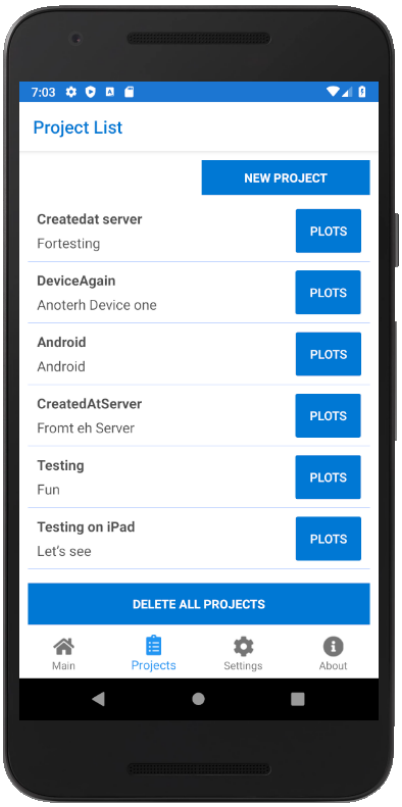

Figure 3 Project Screen. A list of plot projects will appear here. If no project already exists, then a user can Add a New Project. THIS IS THE UPDATED STYLING IN THIS SCREEN SHOT. IT IS CONSISTENT BETWEEN ANDROID AND iOS

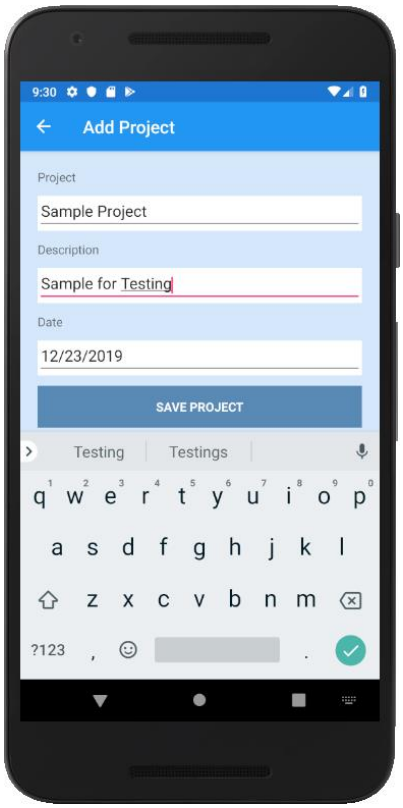

Figure 4 Project Details screen. For entering project attributes. At least one project must exist to add plot data to.

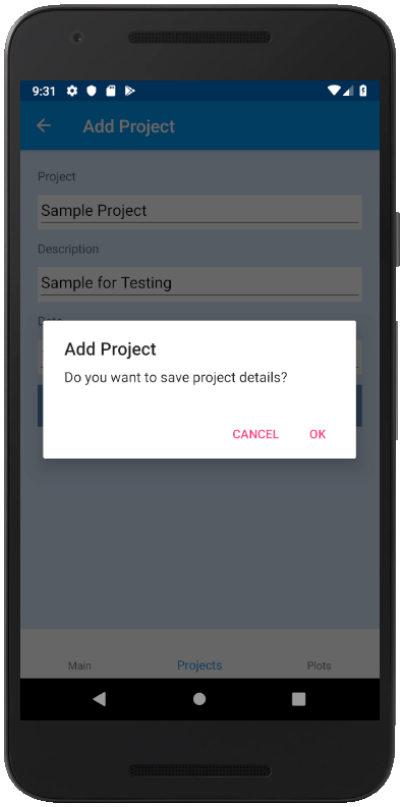

Figure 5 All screens feature a 'Save' utility to save changes to the SQLite database on the device. The device and database always operates disconnected. Data is pushed to the cloud when the user requests it. The data stored on device on SQLite will sync up to an Azure SQL version of the same database when the user has connectivity.

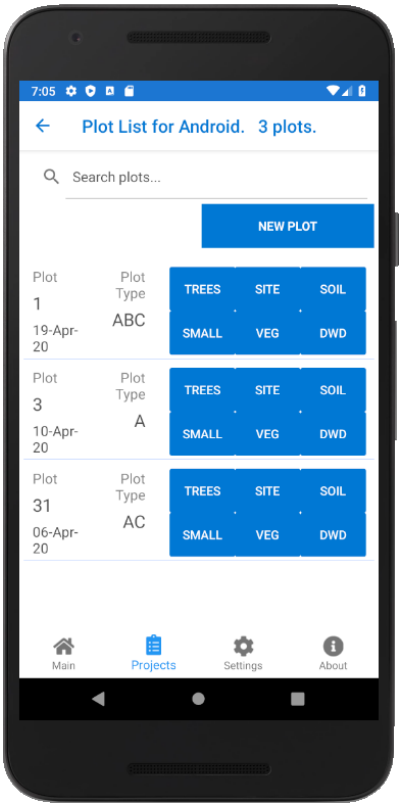

Figure 6 Once a new project is added, the user returns to the previous list to see the results. Tapping the 'Plots' button will take the user to the plot list for that project. Alternatively, you can go directly to the 'Plots' tab and get directly to plots. THIS IS THE UPDATED STYLING IN THIS SCREENSHOT.

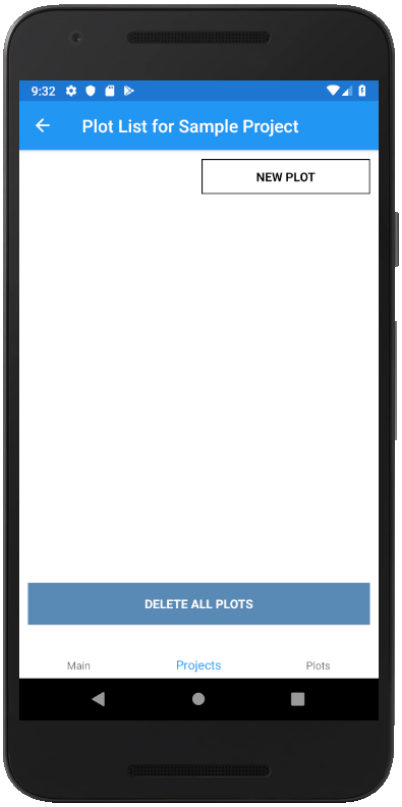

Figure 8 Plots related to the project will appear in a plot list. A new plot can be started from the 'New Plot button.

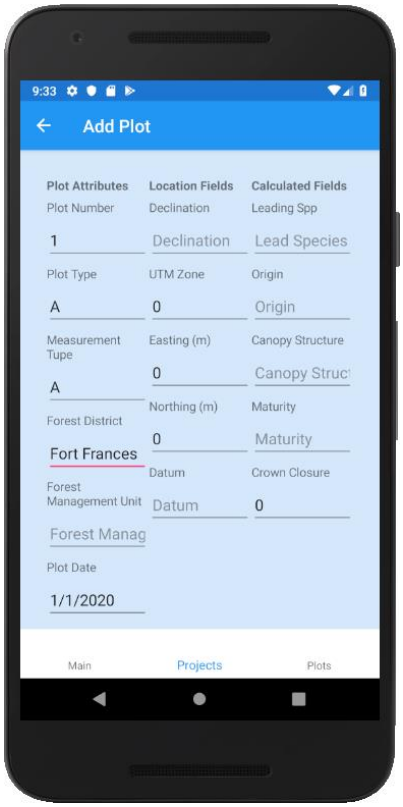

Figure 7 Plot details are entered into a plot screen. All screens are responsive and can work on large screens, small screens, portrait or landscape mode. Fields have been organized into three columns to maximize the use of the screen and reduce scrolling by the user.

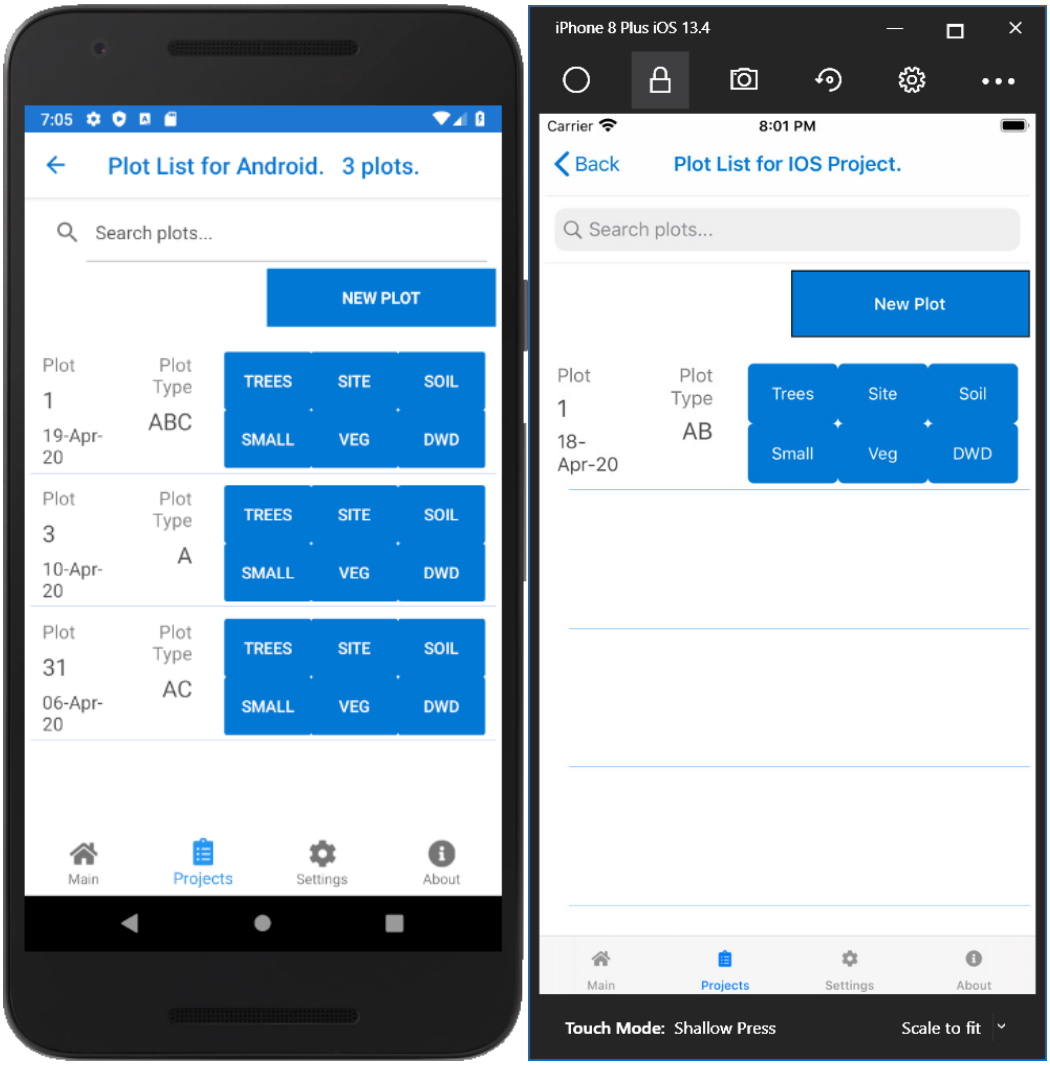

Figure 9 Plot List screen, when plots have been entered (3 plots shown). All of the other plot elements are accessible from here, to get to trees, site, soil or small trees data. Depending on the plot type, the appropriate functions are available (for DWD, Vegetation, Small Trees). On the right is the plot list form on an iOS 13 device. On the left is Android 9.0.

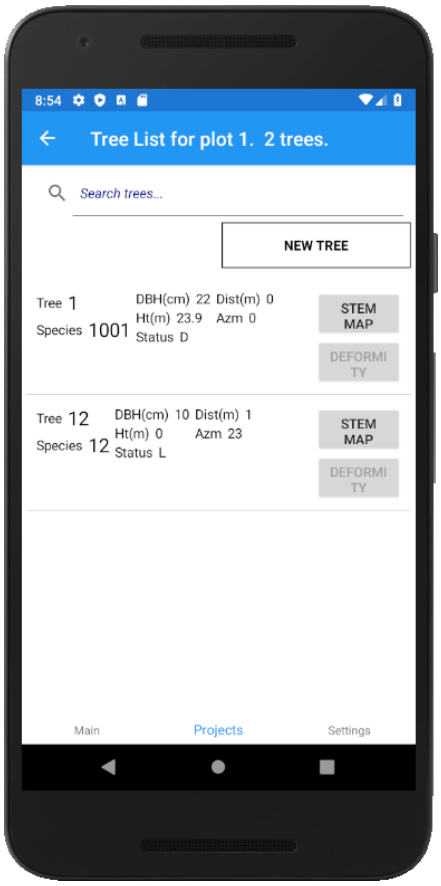

Figure 10 To start adding trees, they can be accessed from the Tree button on the plot list screen. This will bring the user to the tree list, with information on all of the trees added to plot, and the ability to add new trees. Basic tree data is displayed in the tree list. Depending on the plot type, Deformity functions may be available as well.

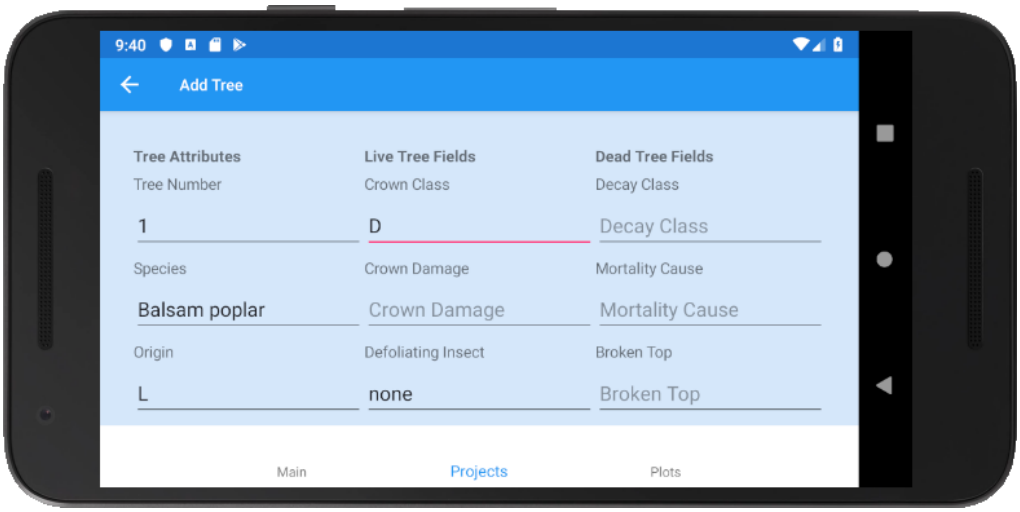

Figure 11 To add or edit a tree, the tree details screen is accessed by tapping on a tree in the tree list.

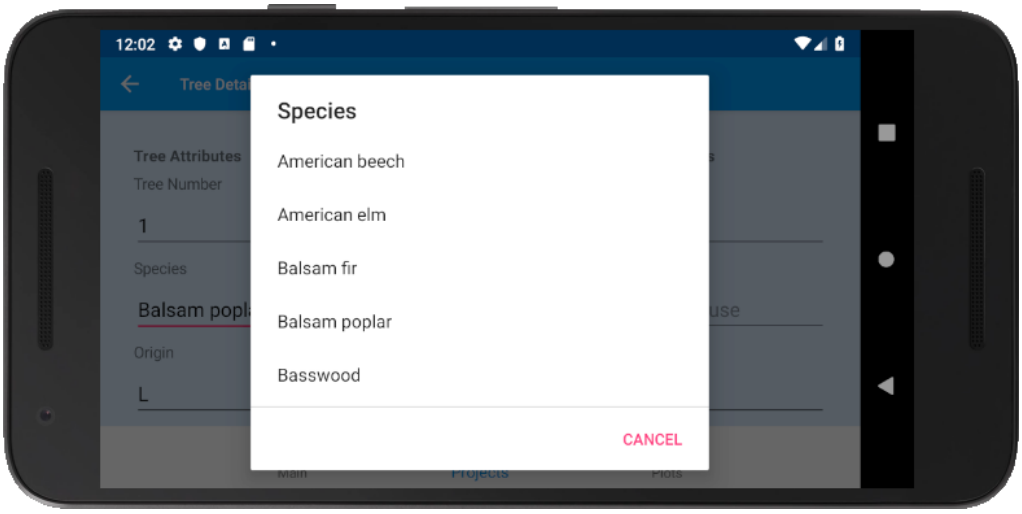

Figure 12 In all screens, and lookup values are available as 'Pickers'. Pickers are a single tap to open a list of valid values, pick the value, and the picker will close and populate the field.

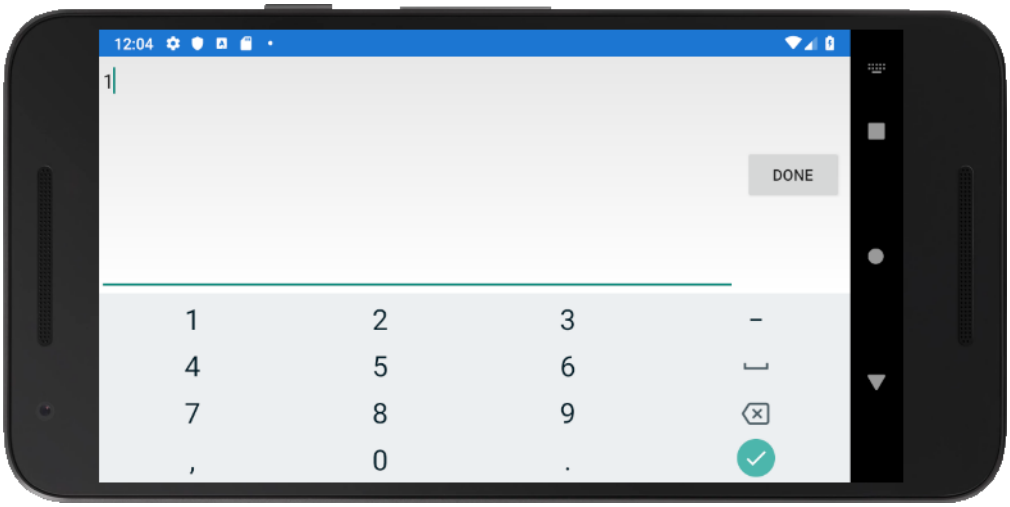

Figure 13 Numeric fields in all screens use a numeric keyboard for quick and efficient numeric data entry.

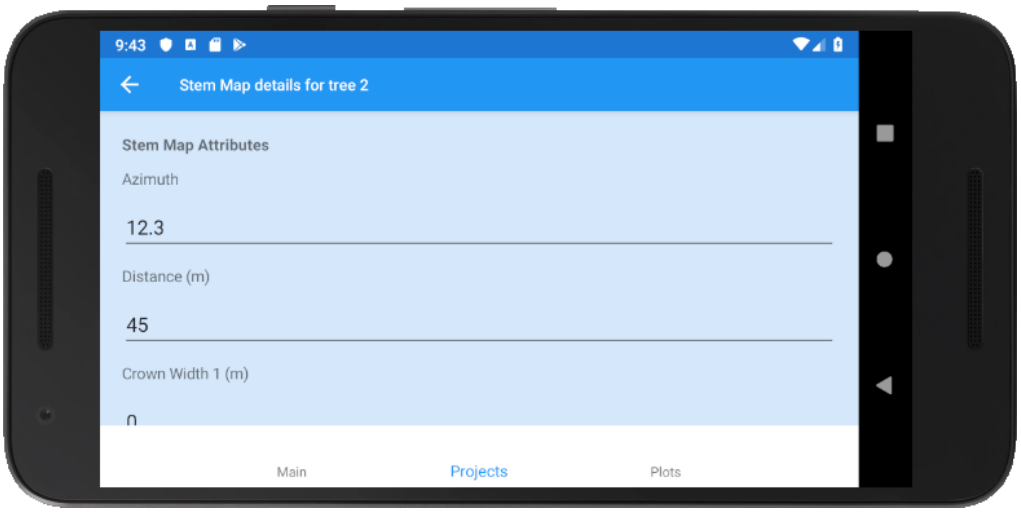

Figure 14 From the tree list, the stem map is accessed separately for each tree from the tree list. The user can see in the tree list which trees have been stem mapped, and quickly tap and edit this data for each tree.

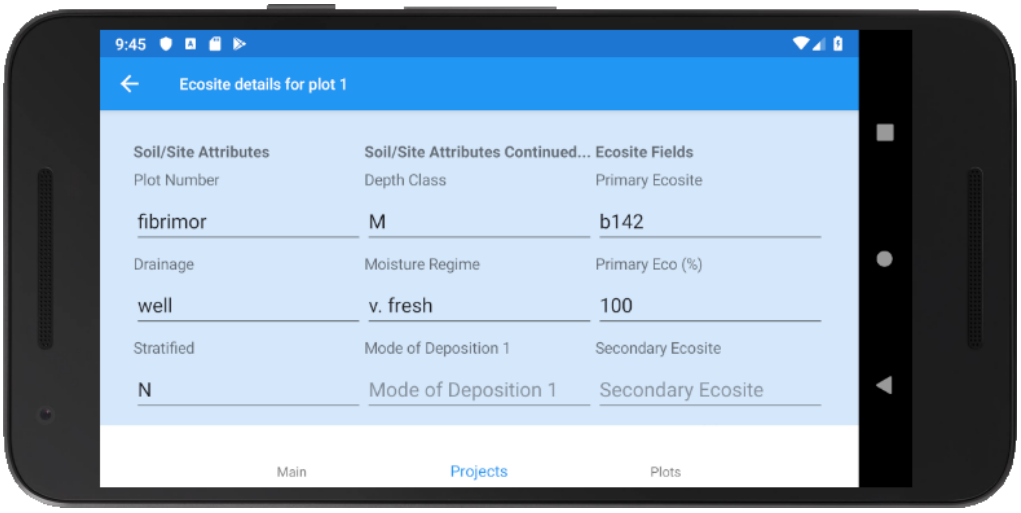

Figure 15 Ecosite/site data is entered from the plot list screen

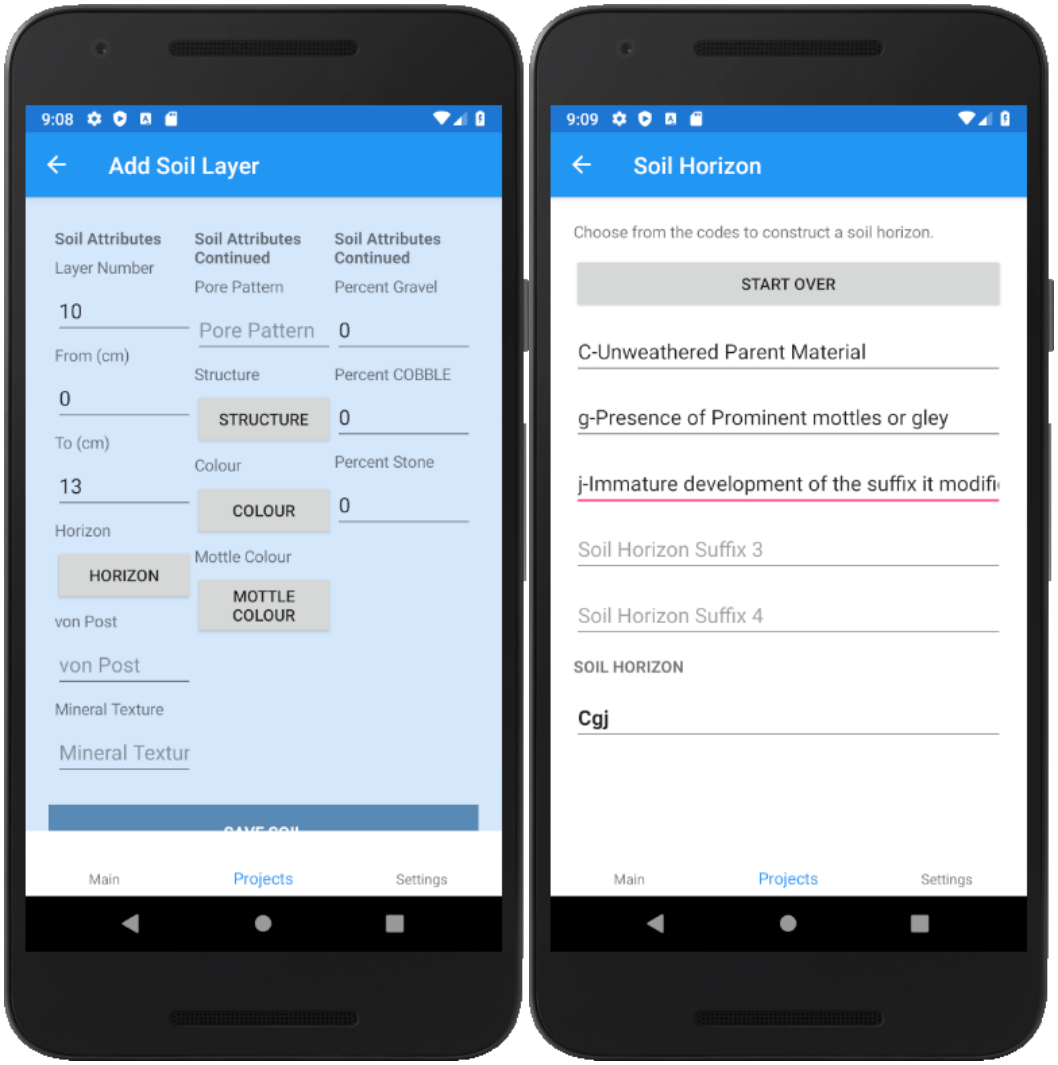

Figure 16 Soil details for each layer are accessed from the soil list. In the final version, more detailed builders were created to allow the user to create complex data types, like Soil Horizon, Soil Colour, Soil Structure and Mottle Colour. These make it easier for the user to create a valid data code.

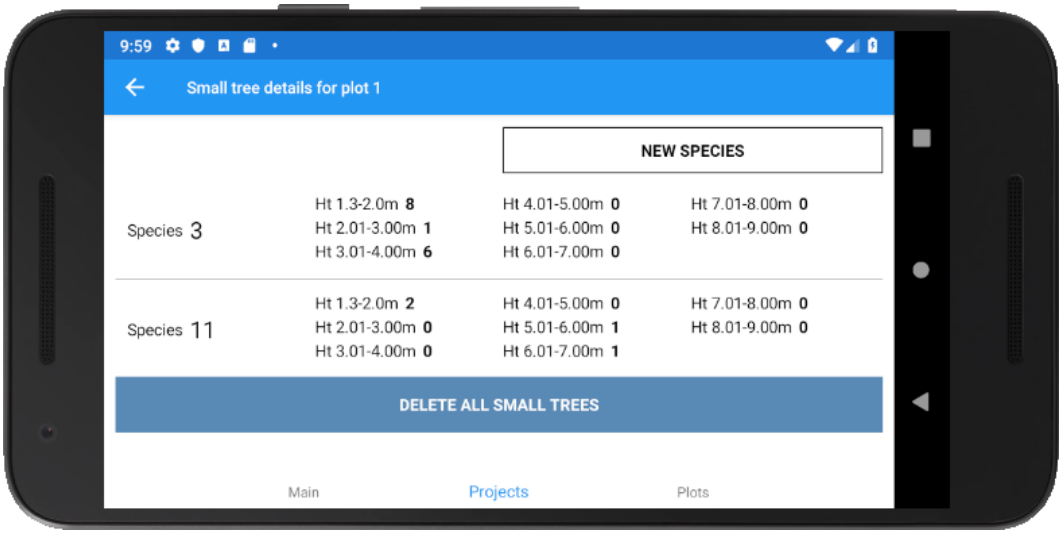

Figure 17 The small tree plot is also accessed from the plot list. Each species in the small tree plot is represented by a row in the list and the key data for each species are visible in the list.

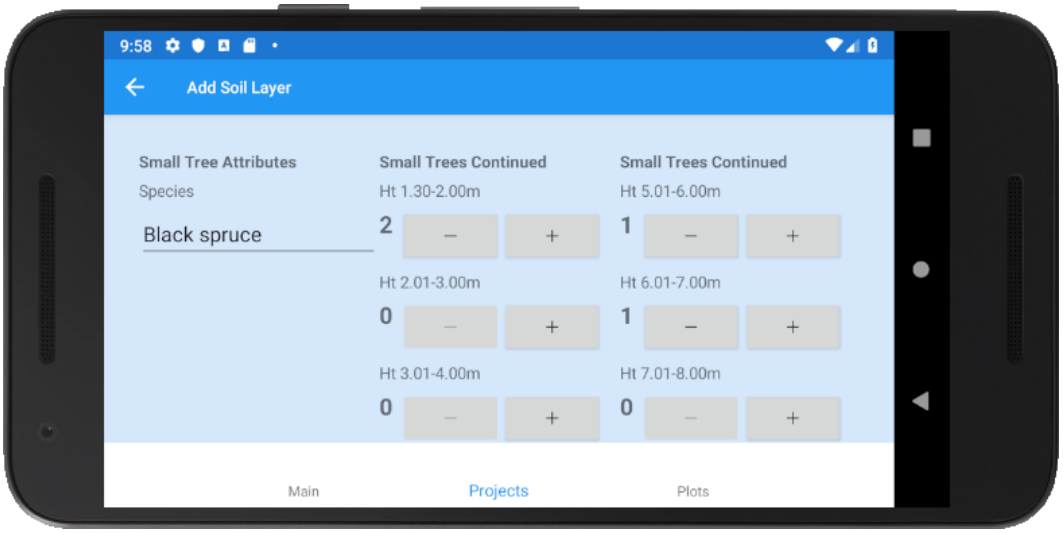

Figure 18 To edit the details for any species in the small tree plot, enter into the details screen for that species. Spinners (plus/minus buttons) let you quickly change the stem count on each height class.

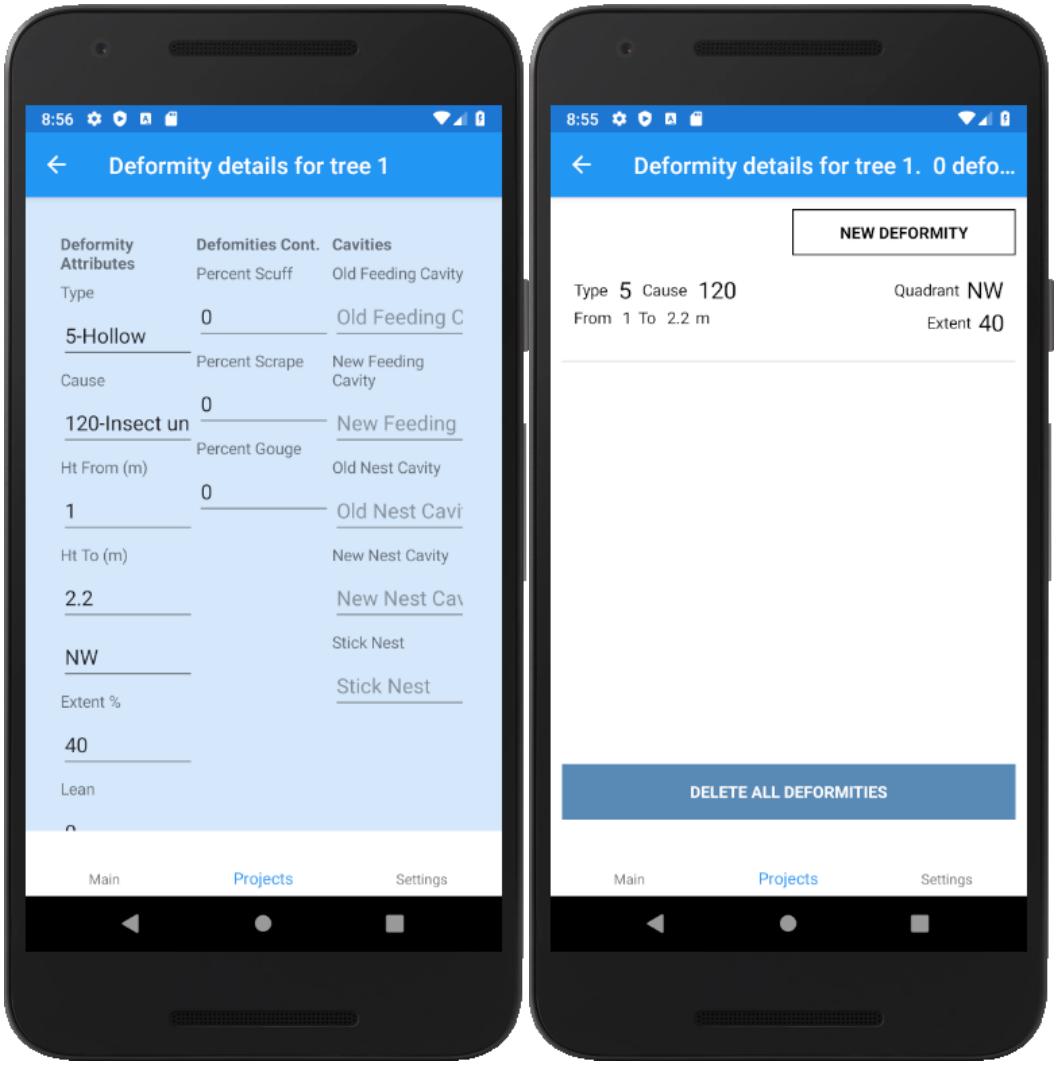

Figure 19 Deformity and cavity data can be accessed from the Tree list to enter multiple defects onto a singe tree.

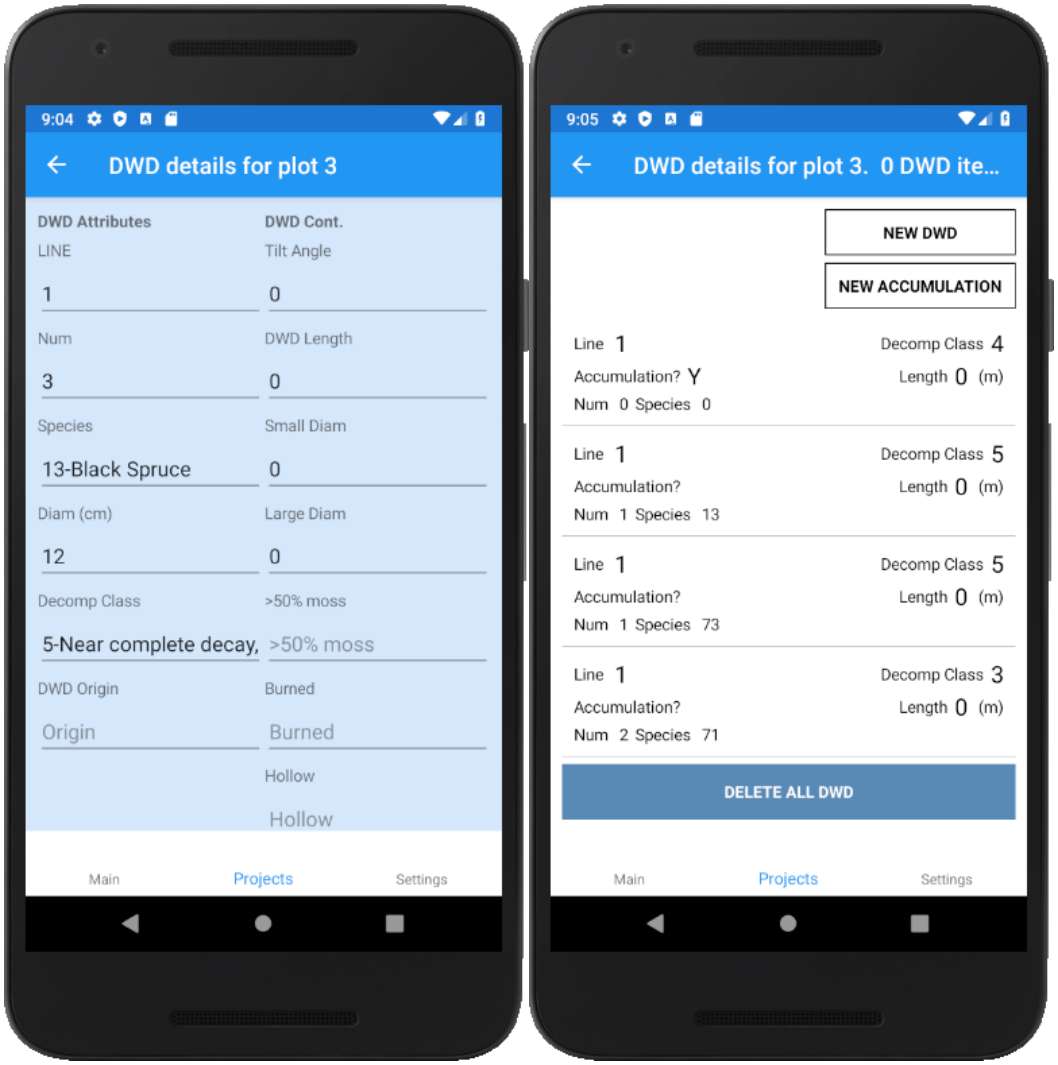

Figure 20 Down woody debris allows for the creation of single pieces of DWD, or pile accumulations, along 2 possible transects.

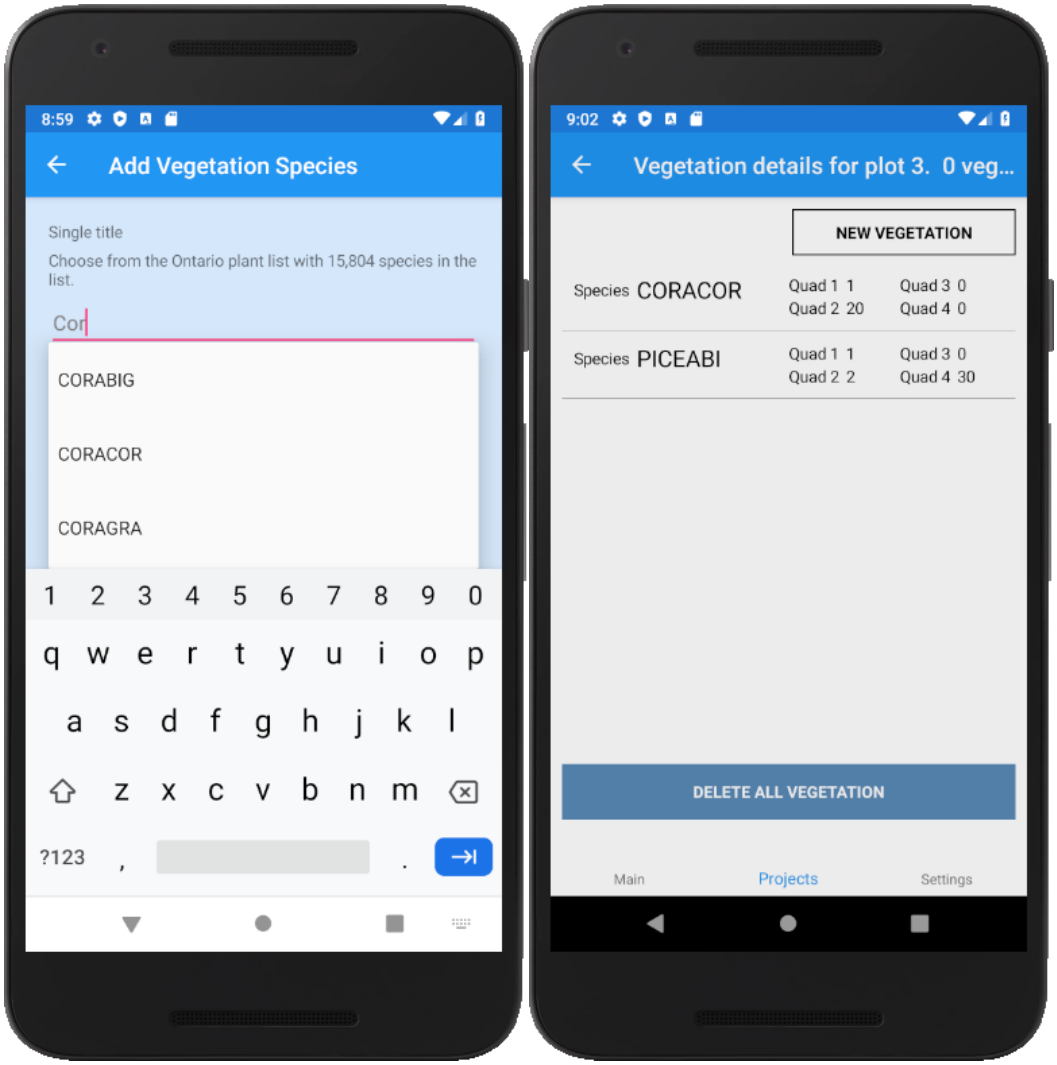

Figure 21 Understory vegetation is collected in an Autocomplete form. There are nearly 16,000 species in the vegetation list, so the user can select a vegetation code by just starting to type, and selections will be matched for the user from the 7-digit code to choose from. The full scientific name accompanies the 7 digit database code.

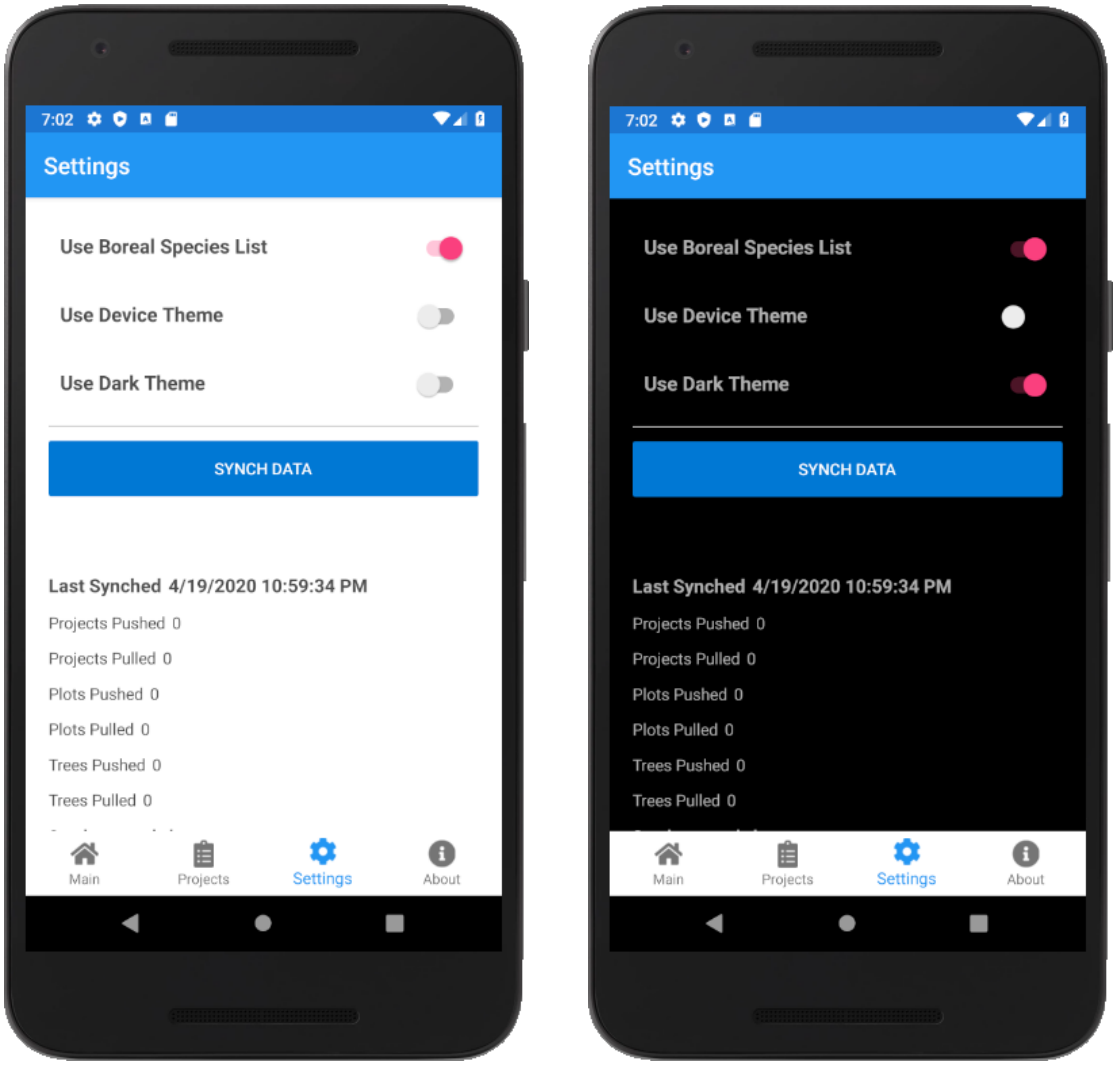

Figure 22 The settings form allows for the easy changing of app settings (like changing to the Boreal Tree List), or to run the synch data operation with the Azure database. You can also change themes on the device, and use a 'Dark Theme' for a different expereince (shown on the right).

4 Location on GitHub

eLiDAR will live in a public repository on Github at https://github.com/csrobins/eFRI_LiDAR_Handheld. From here, anyone can download the full code and related components, or they can choose to modify and continue to develop the application further with their own code and solutions. The current version is available here now. In the repository, you will fin d all the code for:

1. The Xamarin.Forms solution (created in Visual Studio 2019), which contains all of the C# code for the eLiDAR common project, the eLidar.Android project and the eLidar.iOS project.
2. The SQLite database (on device) is included in the db folder.
3. The SQL to recreate the Azure SQL database is in the API folder.
4. The JSON configuration for the 4 Azure Logic Apps to power the REST API are available in the API folder.

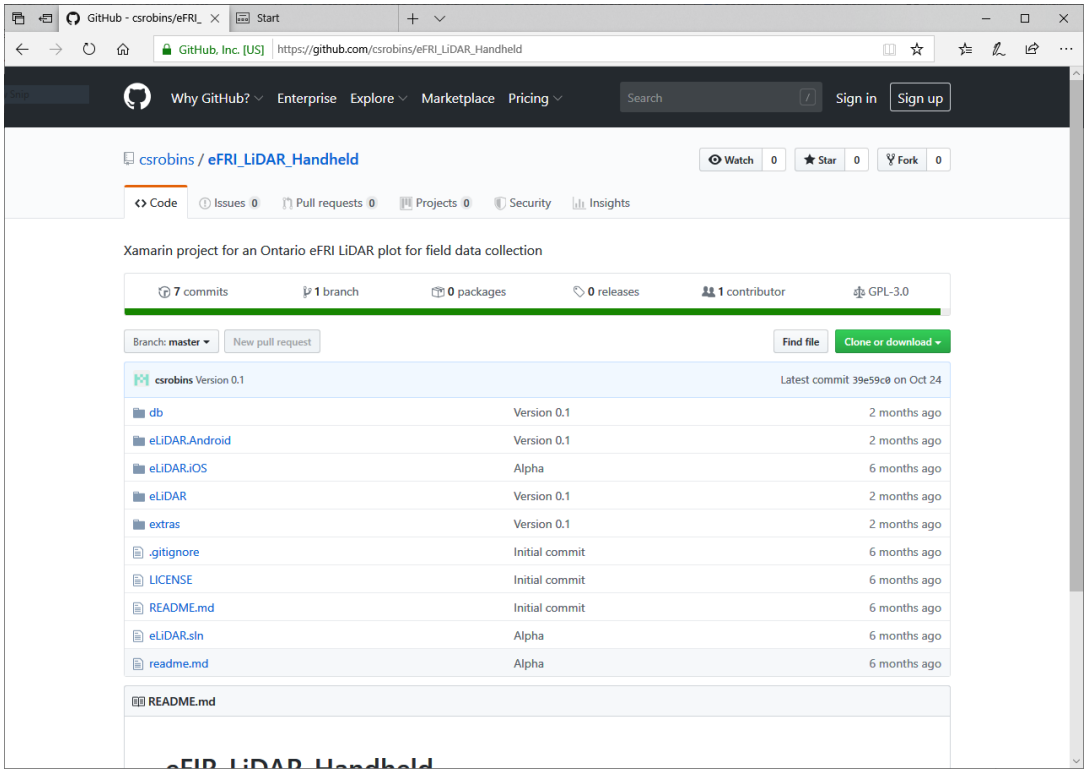

Figure 23 Screen shot of eLiDAR on Github.

5 Further Advancements

- The app will require some production testing on actual eFRI plots when they are available. This will identify further items to be cleaned and/or added. More validations are likely required. The camera page can also be completed to allow for soil pit and plot photos.
- A full Azure API (using Azure API management) should be created and deployed to protect the app and the cloud SQL database. The current database is sitting quite open and unprotected in Azure. When a group of data collectors wishes to deploy or enhance the app, an Azure database and API can be established prior to deployment as a data repository.
- There is an Android 9.0 version of the app currently available here. This is an unsigned version of the app, so to install it, you will need to allow the Android device to install it from an 'unsafe' location. This version is not protected, and the data is not real, so feel free to play with the app and the data. When you go to the link, you will see an APK file to install. Tap to install the file and follow the instructions on the device.
- The iOS version is also available for deployment on iOS 13 and greater. iOS takes more effort to deploy to devices due to the account and device protections created by Apple, so ask Craig Robinson if you would like to try out the iOS version.