

# Forest Ecology Android App

## Project Statement

Joseph Landreville, Sergei Poliakov, Moyez Ikhlas

### Abstract

Data collection is an integral part of what our client does. As such, a tool which can expedite that process would be incredibly useful. Our client's goal is to create an app which runs on ruggedized Android tablets and allows for easy recording of tree plots while in the field. Their current system is no longer cutting it and needs to be replaced with something which runs on more modern hardware. Special consideration will need to be made for the usability of the app given the fact that it will be mostly used outdoors on devices with large screens. Feature wise, the app must allow forms which can branch based on inputs, retrieve previous data, and synchronize with a remote server. Thankfully, our client also wants us to utilize the Open Data Kit (ODK) which was designed to handle these very problems. By the end of the work period, our group should have a working app with functional implementations of the features discussed with the client.

### I. PROBLEM DESCRIPTION

Our client wants us to develop an Android app which facilitates data collection while in the field at forestry research plots. Our client is currently using a relatively old electronic data collection system (by today's standards), which they are seeking to replace with something more modernized and less likely to cause problems. The devices they use now have small displays, are clunky to use, and are liable to crash and lose the contained data. This means regular syncing with a laptop on-site is necessary to prevent data loss.

Open source solutions were something our client was interested in, and the Open Data Kit project was the one they decided to have us help them implement. This involves both configuring a server to act as the endpoint which all the Android devices sync to, as well as configuring the suite of Android apps ODK provides in order to effectively replace (and ideally improve upon) our client's current workflow.

In this regard, consideration towards the design of the user interfaces must account for the environment the apps will be used in. Problems such as legibility while outdoors and a potential lack of internet connectivity are going to have a larger impact on design than they might for a normal application. Our client has specified that the app will be deployed on ruggedized tablets for use onsite, meaning we will have to create a UI tailored to a larger screen.

Our client has also listed several features the Android apps must implement. The main one seems to be forms which have the logic within them to be able to branch based on user inputs to particular questions. For example, if a user answers the question "Is the tree dead?" with Yes, then the form should follow up with questions that collect the data needed in such a case rather than the questions asked about living trees. Another feature our client has asked for is the ability to populate forms with previously collected data, so say you revisit a plot, you can open up

a form for that plot and it will automatically have all the fields filled with the data collected the last time that form was filled out for that plot. Finally, they need the Android devices to be able to sync the data they have collected with a server somewhere.

## II. PROPOSED SOLUTION

Most of the use-case problems, as well as the target features, seem to be covered by the capabilities of ODK itself.

ODK-X's documentation specifies that it can create and deploy forms which allow the user to navigate through all fields available in the form, but not into areas which require input validation. Forms are defined in Excel files – all aspects, including fields, structure, and flow – which are then run through the ODK XLSX Converter to be turned into ODK Survey definition files. These files are then copied onto the appropriate devices to be used in the application.

It has built in support for synchronizing with a database on remote servers and can handle updating/submitting changes to previously uploaded data. Additionally, the majority of its user facing screens can be defined and configured through HTML, JS, and CSS. This makes designing the application's workflow and appearance relatively straightforward. We should be able to quickly iterate on the UI in order to make it as usable as it needs to be. We should be able to easily design an interface which is both simple and intuitive to use and also accommodates being used primarily outside.

## III. PERFORMANCE METRICS

In our case, performance will likely best be measured in terms of deliverables. Completion of the various feature requirements set by our client will serve well to demonstrate progress. Given the nature of ODK-X, I feel that meeting all current requirements set by the client in the time we have to work on the project is well within reason. Should we complete work on all currently specified goals we can discuss additional features to work on with the client.

Agile style development will likely work well with this project. Setting certain features as a goal for one or two week sprints will be an easy way to focus development and keep the client in the loop with the progress we are making. Specifically regarding usability, giving our client frequent updates on how the app is shaping up, or even working builds which they can experiment with, will allow us to get feedback on how well we are meeting those requirements. The client can tell us what changes they've found work well for what they want and help guide future development. Having a short development cycle allows both our group and the client make sure that requirements are being met as development continues.

At the end of the work period we should have a functional app which has working implementations of the various feature requirements discussed both prior to and during development. These requirements may change based on how fast or slow progress is made, however, frequent check-ins with the client should insure that both us and them are on the same page about where the project is going to be by the end.