TreeSpotter

Alex Sun, Gautam Rao, Adam Gradess

App Concept

Using the Google API, have the user find their current location, then prompt the user to log the tree, take a picture, include fields like type of tree, whether the tree looks healthy, whether it's fruit bearing, leaf color, bark color, leaf shape. Trees can be surveyed in this way, and a map of trees in the area and their properties will be available.

Key Functionality

The user can upload a picture of a tree and log its attributes, see what other users have logged, see a map of all trees in the area, filter that map by tree attributes, request to update existing trees if (1) the status changes, i.e. the tree falls down, or (2) the tree is incorrectly labeled, add friends?, notification on new tree in area.

Rough Architecture

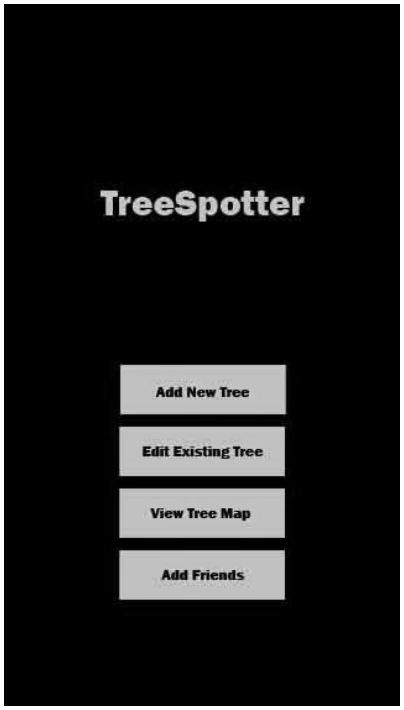
When a user opens the app, the title screen has buttons "Map New Tree", "Edit Existing Tree", "View Existing Trees", possibly "Add Friends". When a user adds a tree or views existing trees, there will be a google maps screen with icons representing existing trees. Once a notification pops up for a new tree added, clicking on the notification results in the map screen, centered on that location, with that tree highlighted. When adding a tree, once a photo has been taken, there will be a form to enter additional information about the tree.

App Components

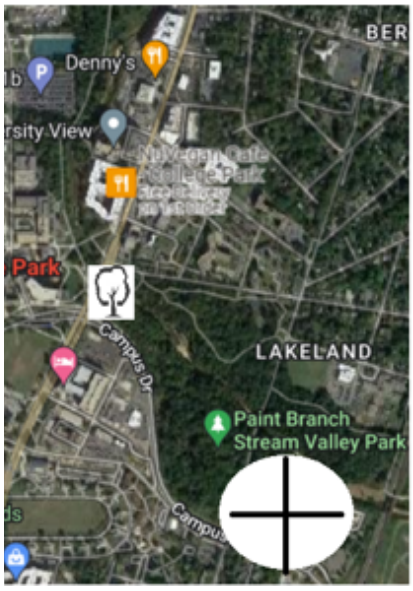
FireBase Realtime database, home screen activity, tree submission form activity, services to handle Google API, notifications added with broadcast receivers.

Work Items for Each Member

Alex - Responsible for setting up the google maps API
Gautam - Responsible for setting up and managing the databases
Adam - Responsible for improving the UI, notifications/friends list



Startup Screen



Map Interface Screen