Extensibility Exercise

We believe modifying GeoPost to support serialized input via web URLs would be a straightforward task for a competent developer. The major components are a new input menu that accepts a URL and downloads JSON (or other format) data, and the parser that takes this data and converts it into a collection of pins. Once a collection of pins exists, the main logic in GeoPost requires a few simple modifications to add these pins to the map in a new color, and in a persistent fashion. Persistence is achieved easily, we simply never add these local pins to the Parse database, and they'll be cleared upon exiting the application. There are some straightforward changes necessary to support things like the new filtering option, and the new pin color. Below are listed the files, and their particular pieces, that would require modification to support this new use case.

File Changes

Main Activity:

-add new temporary option to sorting enum

-change addPin to check for a temporary pin and change its color to green

-change onOptionsItemSelected to handle clicks on the menu item and open a fragment

-change onItemSelected to handle choosing the temporary sorting option

-add a case to getVisible to check for the temporary option

-change drawMarkers to not delete temporary pins

-add a listener on the new fragment that takes in a collection of pins and stores them in the local pin storage map

main.xml (in res/menu):

-add new item to menu bar

New Fragment off of settings menu:

-allow users to enter a url and hit an "ok" button

-parse the JSON in the url and return a collection of pins to the listener

stringarrays.xml (in res/values):

-add a new item to the sorting array for the new sorting option

dialog_help.xml:

-add a new item to the key with a green pin

DBStore:

-unlockPin() changes to support temporary pins by not querying database if temp pin

Pin class:

-Temporary field and all the boilerplate that comes with it