Project 3
Google Maps and GPS

Ryan Rose (rtr29) and Walter Huang (zxh108)

October 26th, 2015

Code Detail

MapActivity.java:

The first bit of code we added was to pass an Intent to take a picture on click of the provided button. This was done by implementing the method given in the assignment and calling it in the On-Click method of the button listener. Next we established location services and Google Play Services. This was done almost directly from code on the Android Developer page, which included building the Google API Client so that we could use the Google Maps API, then making a Location Request with High Accuracy, a Location Listener that registered location changes, and starting/stopping location update methods.

A large component of our app is the CSV file that saves the time and location of saved pictures. When our app runs onCreate(), it initializes a file called “markers.csv” if it does not already exist. If it does not exist it writes the first line of headers for the CSV: Timestamp, Latitude, and Longitude. If it does exist, our app reads the contents of the CSV into a list of marker entries. When the csv file is read, this list inside is then iterated through and markers are created at the given latitude and longitudes (further discussion of our code will explain how the thumbnail images are associated). All markers will be recreated based on previously saved information.

When the user clicks the take photo button and proceeds to take and confirm a picture, the following happens in onActivityResult(): the image is timestamped, then read into a Bitmap. The Bitmap is compressed and saved as a PNG file named by its timestamp. The app then captures the location given by the LocationListener and creates a marker with the given position and with an icon provided by the Bitmap. The marker’s timestamp and coordinates are saved into the CSV file. This saving is important, as this is how the markers are saved for following app sessions if the app is closed--the CSV is read line by line to recreate the markers, and the timestamp is used to pull the PNG thumbnail to server as the marker icon again. All changes are immediately saved to the CSV.

The next big feature of our app is the ability to change the title and snippet of markers on the map. This is done first by using the behavior of OnMarkerClick to display the InfoWindow, which holds the title and snippet. We then used a OnInfoWindowClick to edit the information. This new title and snippet can be added via two EditTexts that appear in a Dialog when the user clicks the InfoWindow. The user can then hit cancel to keep the old title/snippet, or add their own and then click save to save the information. The new title and snippet appear when clicking on the marker again. Further description of the Dialog can be found below. Once edits are confirmed, all edits are reflected in the markers.csv file.

Another helpful feature we added is the ability to remove a marker from the map. This is done by user marking “delete” when they click on info window to popup the title-snippet Dialog. By

using a checkbox, users could double-check before they delete a marker and its associated csv information.

TitleSnippetDialogFragment.java:

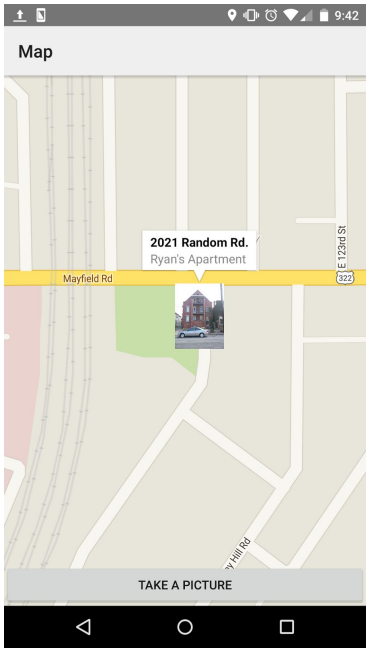
This Dialog is the solution to the fact that the InfoWindows themselves are not directly editable, and in fact are rendered on the map as an image. This class uses an AlertDialog Fragment with a custom View to allow for user input. The Dialog has two EditText Views, one for title one for snippet, and two buttons: one positive (save) and one negative (cancel). The negative button calls the Dialog's cancel method and the user is returned to the map with no change in title or snippet. The positive button retrieves the information from both EditTexts and uses them to set the title and snippet of the marker. We used a separate method to pass the marker into our fragment, as the fragment must have an empty constructor. We were then able to directly change the attributes of the InfoWindow through the marker. This Dialog also writes the title and snippet to the CSV file, which allows the info to be saved throughout app sessions.

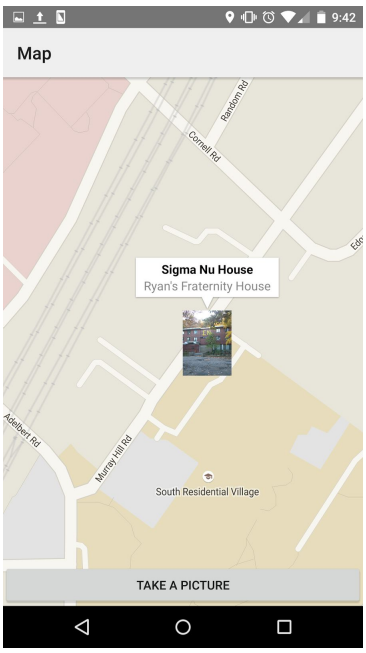
Experimental Results

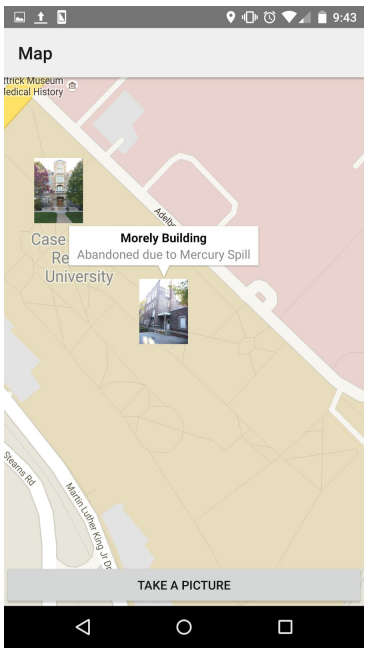
We used our app to capture photo-locations and titles/snippets for ten buildings on campus. Here is the complete map with all ten locations (on the following page):

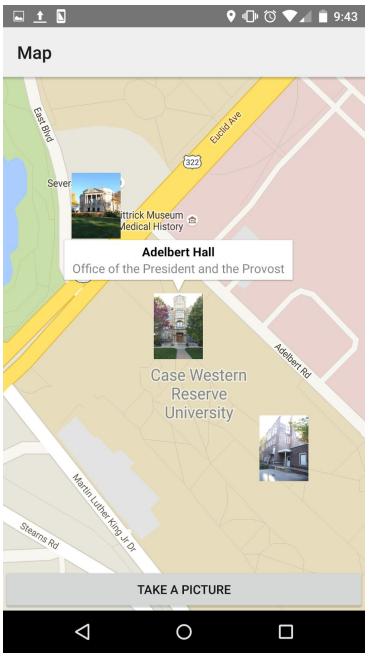

9:24
Map
East Cleveland
E 118th St
East Blvd
Euclid-E.120
M
20
Cleveland
Botanical Garden
Ford Dr
Museum of
Contemporary Art...
UNIVERSITY
CIRCLE
Mayfield Rd
Euclid Ave
Cornell Rd
Circle Dr
Random Rd
Health
Sciences
Campus
TAKE A PICTURE

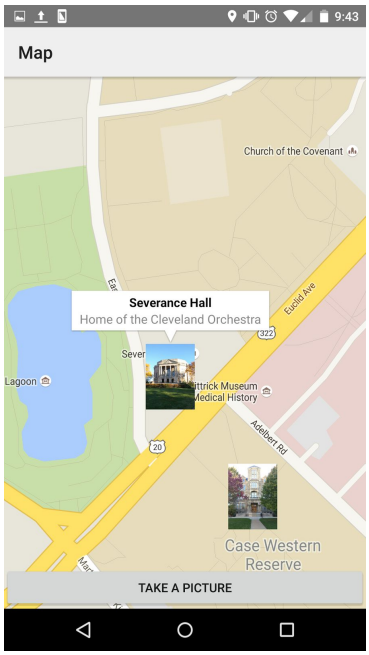
The buildings we did were as follows:

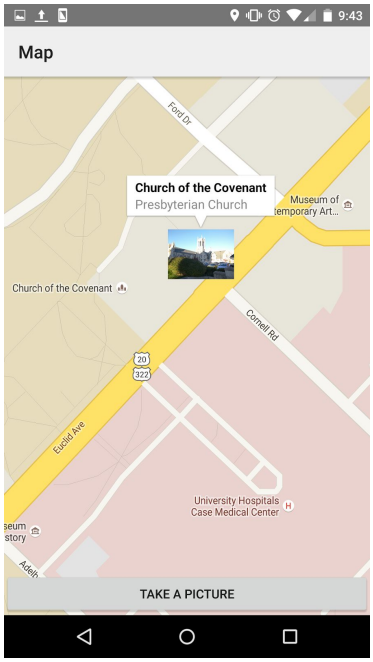


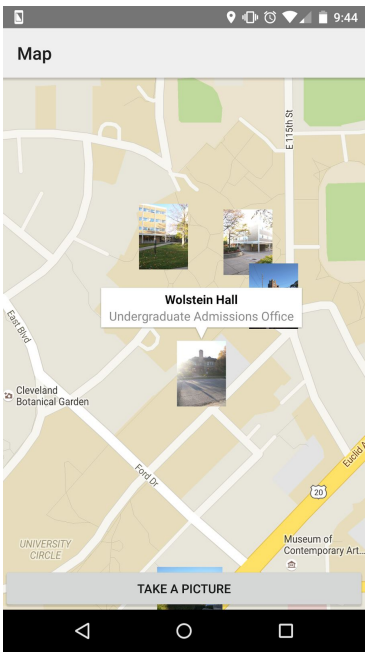


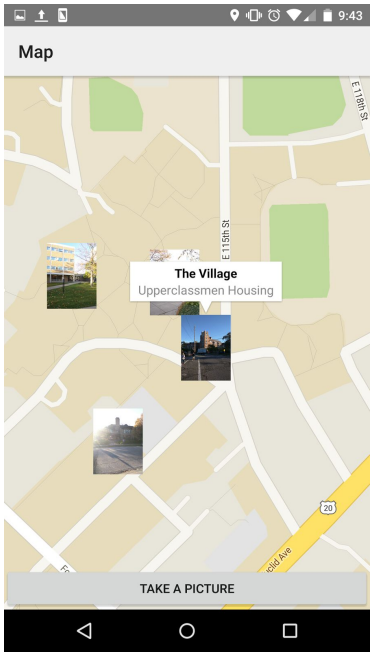


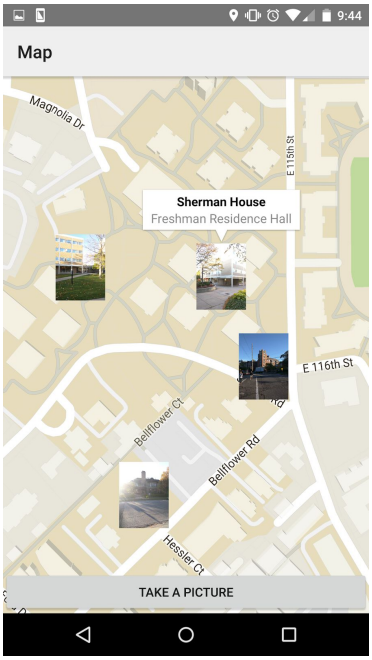


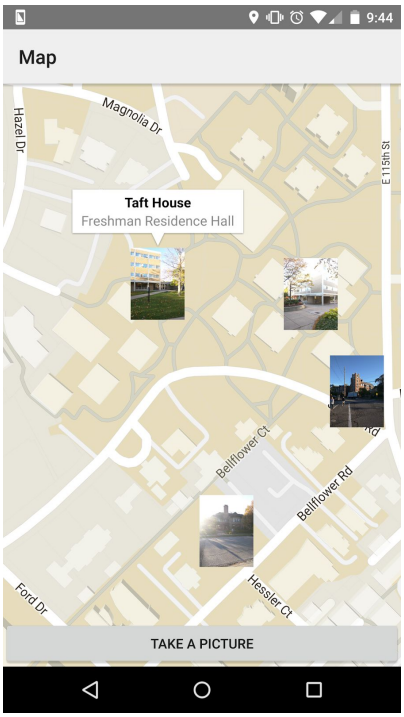




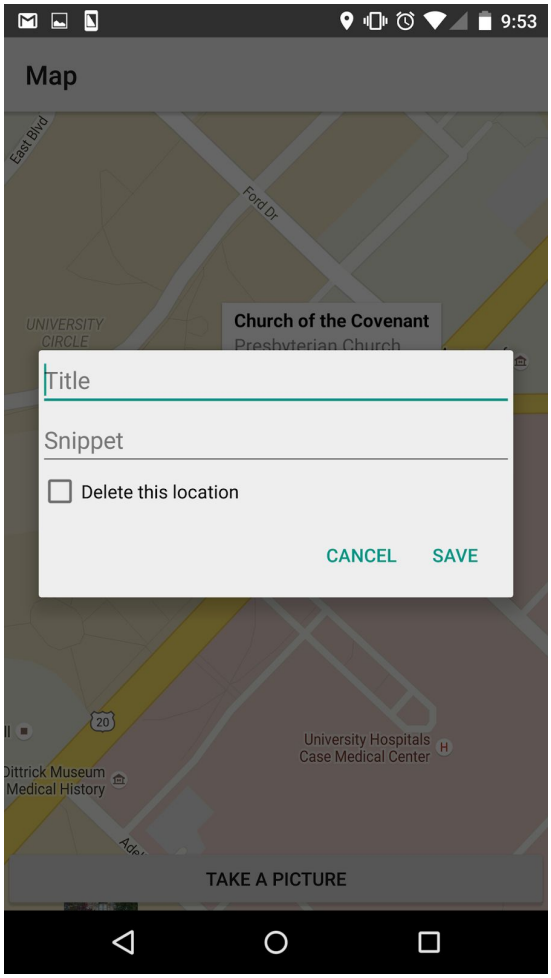

9:44
Map
E 115th St
Wolstein Hall
Undergraduate Admissions Office
East Blvd
Cleveland Botanical Garden
Ford Dr
UNIVERSITY CIRCLE
Museum of Contemporary Art...
TAKE A PICTURE


9:43
Map
E 118th St
E 115th St
The Village
Upperclassmen Housing
TAKE A PICTURE


9:44
Map
Magnolia Dr
E 115th St
Sherman House
Freshman Residence Hall
E 116th St
Bellflower Ct
Bellflower Rd
Hessler Ct
TAKE A PICTURE


9:44
Map
Hazel Dr
Magnolia Dr
E 115th St
Taft House
Freshman Residence Hall
Bellflower Ct
Bellflower Rd
Ford Dr
Hessler Ct
TAKE A PICTURE

Here is a screenshot of the editable title and snippet, as well as the delete option:



Here is a capture from our CSV file. We added the columns (Title) and (Snippet) in order to retain user's changes of the title and snippet:

No.	TimeStamp	Latitude	Longitude	(Title)	(Snippet)
1	1445784382832	41.5079812	-81.5991251	2021 Random Rd.	Ryan's Apartment
2	1445862588864	41.5026845	-81.6014615	Sigma Nu House	Ryan's Fraternity House
3	1445864465667	41.5127376	-81.6054112	Sherman House	Freshman Residence Hall
4	1445864571582	41.5128115	-81.6069066	Taft House	Freshman Residence Hall
5	1445864731973	41.5120252	-81.6049593	The Village	Upperclassmen Housing
6	1445864834143	41.5110044	-81.606233	Wolstein Hall	Undergraduate Admissions Office
7	1445865122633	41.5082368	-81.6064422	Church of the Covenant	Presbyterian Church
8	1445865366976	41.5057644	-81.6096639	Severance Hall	Home of the Cleveland Orchestra
9	1445865598667	41.5046553	-81.6086445	Adelbert Hall	Office of the President and the Provost
10	1445865712588	41.5035298	-81.6073357	Morely Building	Abandoned due to Mercury Spill

Contributions

The project was split somewhat evenly. Ryan Rose got started and worked on setting up the environment, repository and finished the initial steps of the project, where markers are created after confirming each photo. Walter Huang followed up afterward and added all the bookkeeping mechanism, as well as displaying info to the info windows. After these two parts are done sequentially, we met up and integrated the title-snippet edit and marker delete features, in a paired programming manner, and all the bookkeeping mechanism are updated to accommodate the changes. We each completed a part of the report and Ryan collected the data with the finished app.