Sam Temlock
Section 1

W200 Project 1 Reflection Document

Testing Instructions

1. Libraries to be installed
    - YelpAPI library ("pip install yelp")
2. Imports
    - Categories.json (included in the GitHub project_1 directory)
    - google_maps.py (module with Map class, included in the GitHub project_1 directory)
    - yelp_categories.py (module with the Category and CategoryTree classes, included in the GitHub project_1 directory)
3. Yelp API key
    - Imported from config.py (sent to instructor separately for security purposes)
4. Using the project
    - All interactions are made via the command line (instructions provided in the interface). User input error checking is implemented throughout . Yelp search may not always use provided criteria (e.g., if the input search address is invalid, it may use a different address, or if the appropriate business cannot be found within a certain radius, it may expand the search distance). Additionally, Yelp businesses with invalid addresses may not be found when the Google Maps directions are returned.

Reflection

Given the timeline for the project, I felt that I was able to complete my initial desired design and implementation, which mainly involved being able to create an activity list and search the Yelp database for matching businesses based on a set of criteria. I was also able to add an extra feature to return a Google Maps with directions between the returned businesses, which I originally did not know if I would have enough time for.

If I were to extend my project, I would like to improve the user interface (maybe with improved visualization/formatting, or leveraging a better console than terminal), as well as to provide a few more checks for input. For example, one feature that I noticed could be improved in terms of error checking returns would be that Yelp tends to be very loose on the search radius, and if it cannot find a business in the suggested radius, it may significantly expand the search much beyond what was requested. This may just be the way that Yelp does it's search internally, but I would want to look into improving this accuracy. Another feature that I would try to improve would be to verify that all addresses searched via Google Maps would return appropriate locations. There have been a few instances where because the Yelp database did not store accurate store locations, the Google Maps API was unable find the requested address, and this prevented valid directions from being returned. I would have liked to try to make sure that all addresses returned valid locations on Google Maps before opening the web page with directions.

The biggest challenge I had with my code was trying to create a navigable interface when asking the user to select a matching business category for their activity. The "categories.json" file that contained the metadata on business categories did not come with any kind of structure, and I had to leverage the "parents" metadata that contained information on parent categories in order to create a mapping among all the different categories. I was able to do this by using a recursive function that would leverage this parent data to create a tree like structure for each base category (referred to as a root in the code documentation). By creating a tree traversal function, I was able to find all the different branches and child nodes for each category, and then stored this information in the Category object to be leveraged when a user selected a category to drill down into. This then allowed me to create a nested dictionary structure that would store this mapping like a tree.