CMSC 12200 Project Proposal

Group Name: Transportation Rating

Group Members:
Nick Nigro nicknigro@uchicago.edu
Devan Richter evanrichter@uchicago.edu
Danielle Rubin dlhrubin@uchicago.edu
Teddy Zamborsky tzamborsky@uchicago.edu

Description and Goals

This project will combine a variety of data sources into an easy to use web interface. The primary emphasis is on public transportation. A user should be able to provide a location (ideally with other preferences, such as whether they value transportation to campus/downtown/other locations), and the program will provide a list of transportation options for the user. Other potential functionality is under discussion, such as a rating system for a list of locations based on what the user's preferences are. Goals for this project include combining the variety of databases (CTA, Divvy, UChicago Shuttles, and Google Maps) into a usable form for our program, providing an intuitive interface for end-users to interact with, and ultimately creating a simple, practical tool.

Data Sources

CTA bus tracker API for fetching bus stop locations and bus routes in Hyde Park: http://www.transitchicago.com/developers/bustracker.aspx

Divvy bicycle stations API for fetching divvy bicycle station locations in Hyde Park: https://data.cityofchicago.org/Transportation/Divvy-Bicycle-Stations/bbyy-e7gq

UChicago shuttle data (scraped from TransLoc) for fetching shuttle stop locations and routes in Hyde Park: https://uchicago.transloc.com/t/browse/uchicago

Google maps distance matrix API for fetching walking distances between apartment locations and various locations of interest (campus, grocery stores, etc.): https://google-developers.appspot.com/maps/documentation/distance-matrix/start

Tasks and Timeline for Completing Them

This is the loose ordering of tasks based on dependencies, but they will likely be worked on simultaneously by different group members after discussing specifications

1. Gather Data
    a. Build python wrapper for Divvy and CTA Bus Tracker APIs
    b. Build scrapper for UChicago Shuttle Stops
    c. Add pulled/scraped data to database for reference
2. Build Backend functions
    a. Build functions for backend, such as measuring distance
    b. Design and implement rating scheme
3. Build Django Interface
    a. Design the aesthetics of the interface
    b. Connect backend code with Django Elements
4. Test code and Review