

Logan Noel // CNET ID: lnoel

Carlos Alvarado // CNET ID: calvarado

Tom Jarosz // CNET ID: tjarosz

Git: https://mit.cs.uchicago.edu/calvarado/project_cs122_abcd.git

CS122 Project Proposal

Goal: Create a software product that develops a custom one-day itinerary of popular sites/attractions throughout a city based on user's schedule, preferences, and the location of the sites.

Data Sources: GooglePlaces API, GoogleMaps API

Software Flow:

1. User Enters Parameters:
 - City to explore
 - Day of travel and time range he/she wants to visit attractions (i.e. 11AM - 4PM)
 - Category(ies) of attractions he/she would like to visit (museums, parks, etc.), excluding food and/or drink establishments.
 - Mode of transportation
2. System Filters and Selects Locations:
 - Query and filter locations from GooglePlaces API that meet the user's parameters (city and category)
 - Extract and store the name, longitude/latitude, numerical rating (out of five), hours of operation, and other useful attributes
 - Calculate and implement minimum rating threshold for sites that are returned given the mean ratings of the subset
 - Select and return the top 10 attractions (based on ratings) that meet the user's preferences
3. User Enters Parameters:
 - From the returned list, user selects sites he/she is particularly interested in visiting (if any, otherwise all 10 are selected as default)
4. System Plans and Displays Optimal Itinerary:
 - Consider highest-rated attractions from subset returned in step 3 (with the travel algorithm) to create custom itinerary, which minimizes travel time and maximizes the number of locations visited (considering an appropriate time to be spent at each location)
 - Develop an algorithm using GoogleMaps API as well as the user's mode of transportation and hours of operation to determine optimal order of itinerary based on location and travel time

- Deliver ordered itinerary of selected sites that the user should visit given preferences and time constraints

Key Steps and Timeline:

1. Connect to APIs and explore storage through custom data structure to filter and return appropriate data based on user parameters
 - Estimated Completion: Week 6
2. Sort, rank, and return top ten locations/attractions/sites that meet user parameters and search criteria
 - Estimated Completion: Week 7
3. Build response component to allow user to select 1 (minimum) to 10 (maximum) sites he/she is particularly interested in visiting that day (from step 2)
 - Estimated Completion: Week 7
4. Develop algorithm to use latitude and longitude and mode of transportation to determine optimal itinerary, which minimizes transportation time and maximizes the number of locations to visit
 - Estimated Completion: Week 8-9
5. Create return display (and user interface, time-permitting)
 - Estimated Completion: Week 10