



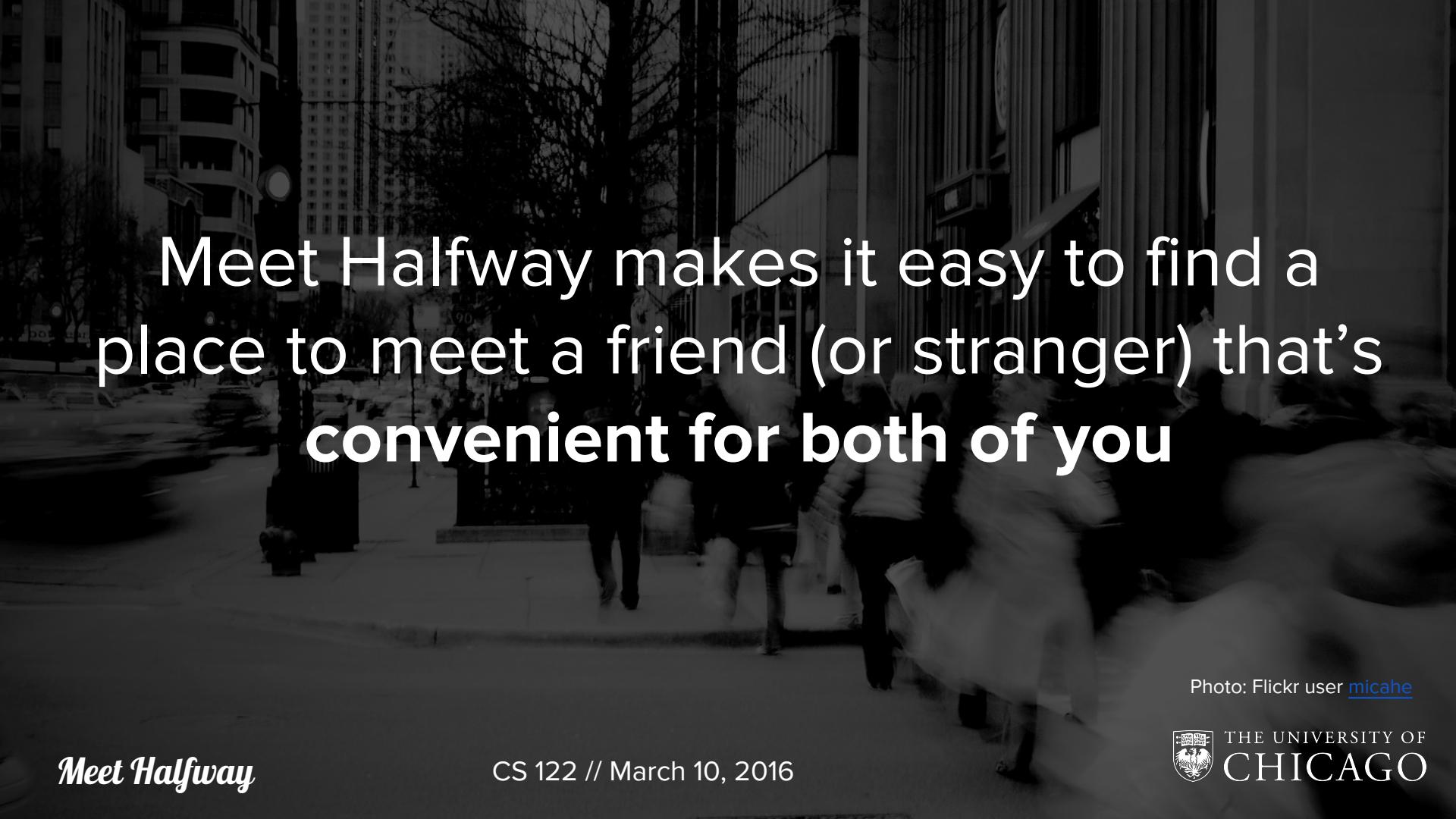
Meet Halfway

Lauren, Christine, Paul, Leith
CS 122, Winter 2016
Final Project Presentation

Find a convenient
place to meet up,
safely and securely



THE UNIVERSITY OF
CHICAGO

A black and white photograph of a city street. In the foreground, several people are walking away from the camera on a sidewalk. The background shows tall buildings and bare trees. The overall atmosphere is urban and slightly overexposed.

Meet Halfway makes it easy to find a
place to meet a friend (or stranger) that's
convenient for both of you

Photo: Flickr user [micahe](#)

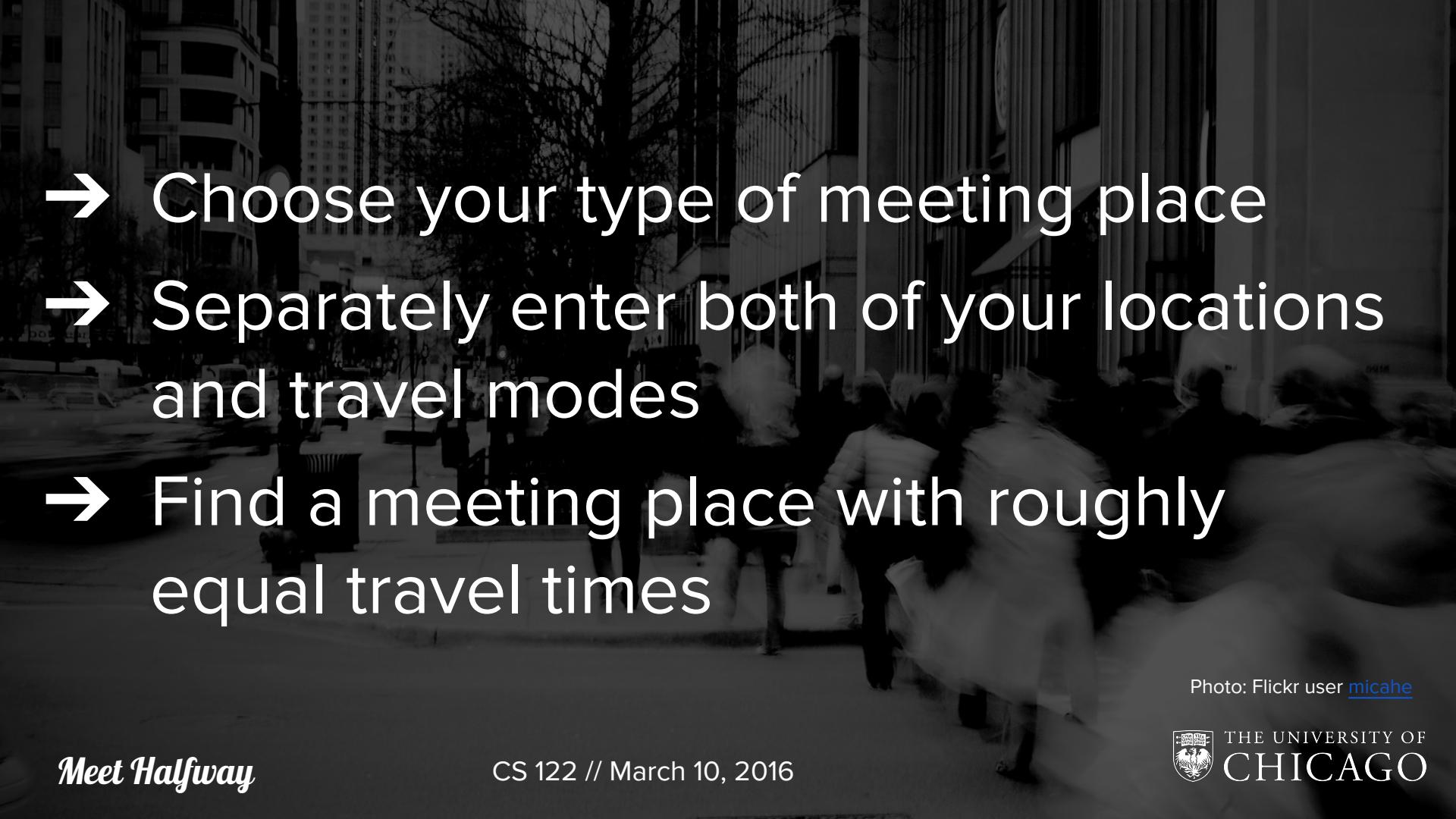
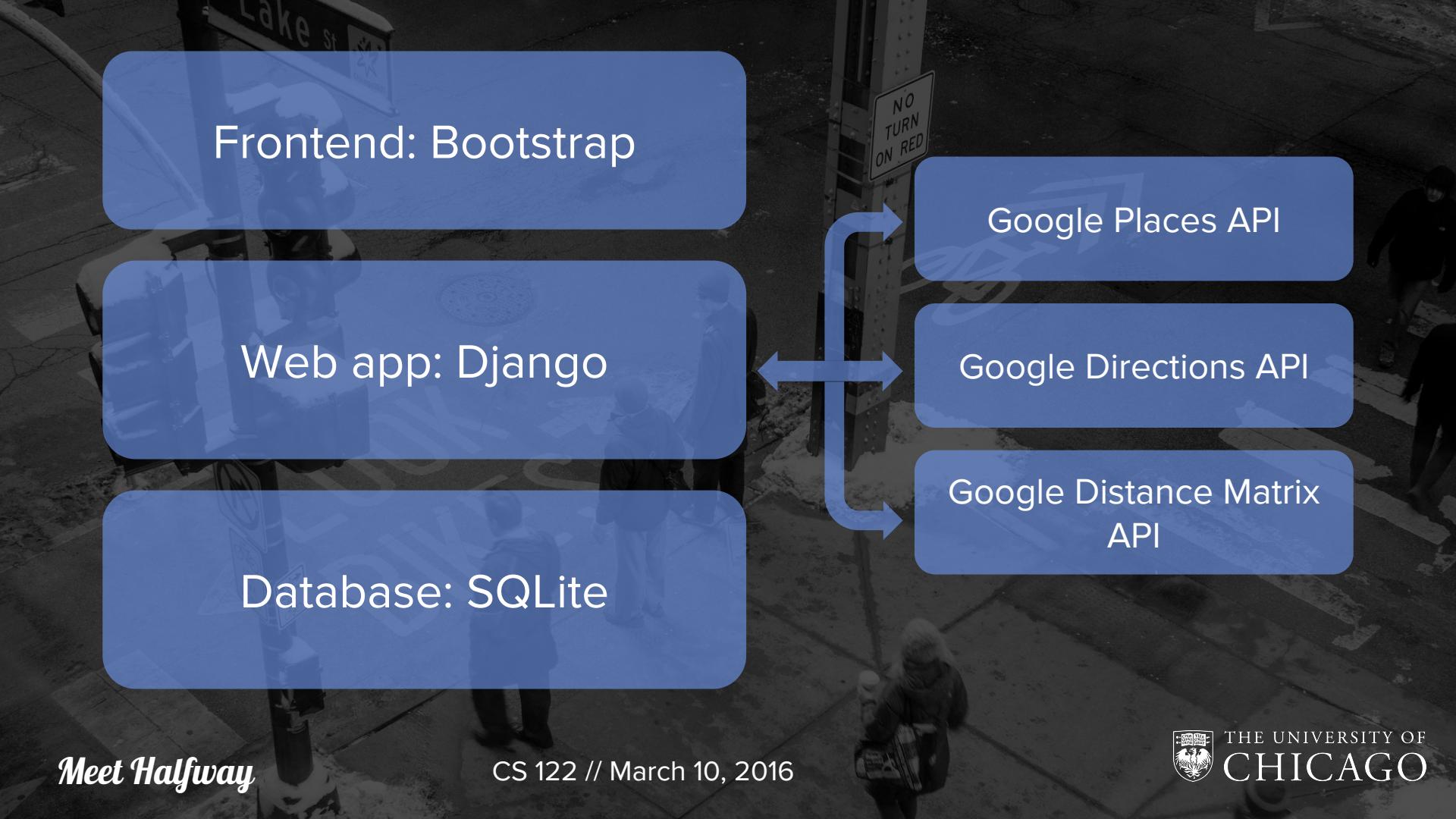
- 
- Choose your type of meeting place
 - Separately enter both of your locations and travel modes
 - Find a meeting place with roughly equal travel times

Photo: Flickr user [micahe](#)

Use Cases

- Arranging to buy something on Craigslist
- Finding somewhere to meet an OKCupid date
- Meeting a friend who lives on the North Side

Photo: Flickr user [micahe](#)



Frontend: Bootstrap

Web app: Django

Database: SQLite

Google Places API

Google Directions API

Google Distance Matrix API

Models

Meeting

Participant 1

Participant 2

Destination(s)

Origin
Address

Origin
Address

Meet Halfway

CS 122 // March 10, 2016

Algorithm Walkthrough

Step 1:

Get **Directions**

Determine **Target Time**

Find all **Potential Meeting Places**



Step 2:

Get **travel times** from each starting location to each meeting place

Calculate **fairness score**



Step 3:

If good results are found, create **destination objects**.

Otherwise, try step 2 again from most fair location.

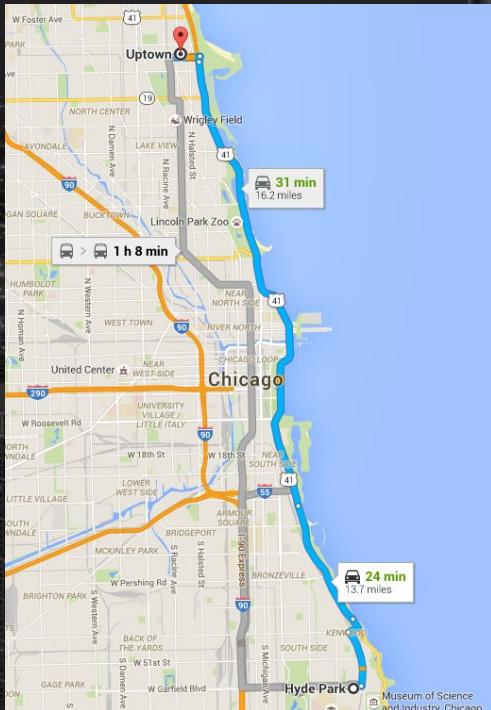
Meet Halfway

CS 122 // March 10, 2016

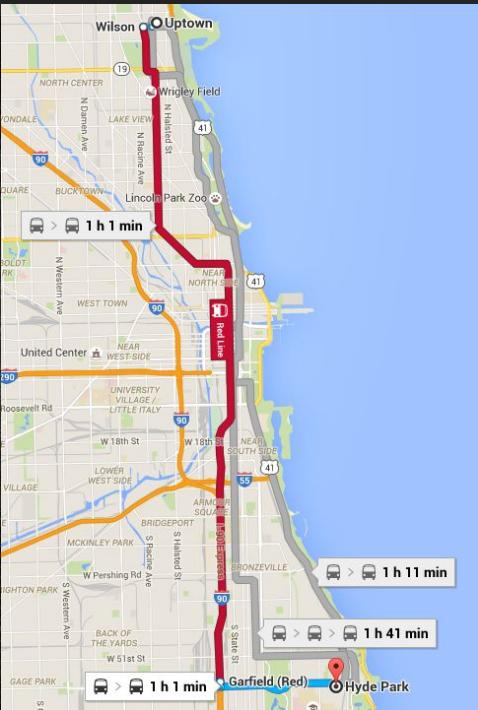


THE UNIVERSITY OF
CHICAGO

Driving from Hyde Park to Uptown:
24 minutes



Public Transit from Uptown to Hyde Park:
61 minutes



Step 1a:
Calculate Target Time

Total Time Traveled
 $24 + 61 = 85$ minutes

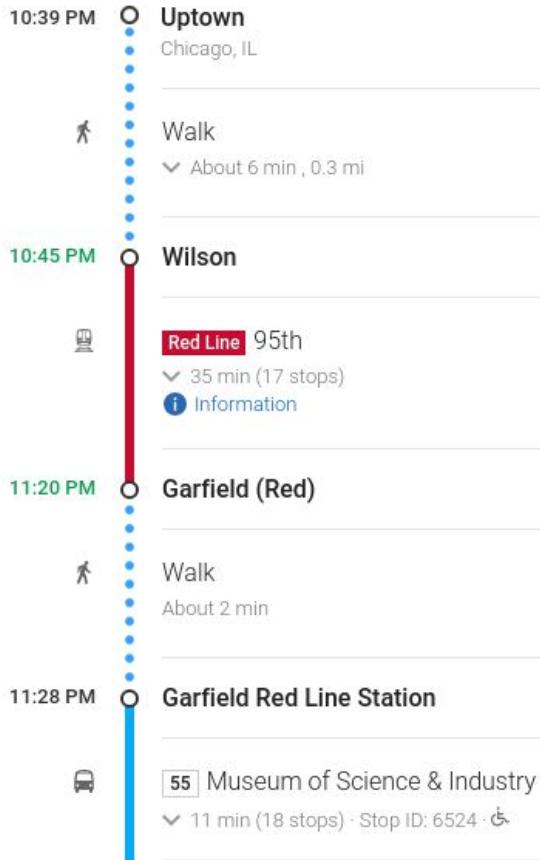
Target Time
 $(24 / 85) * 61 =$
 $(\text{or } (61/85) * 24)$
17 minutes

Meet Halfway

CS 122 // March 10, 2016



THE UNIVERSITY OF
CHICAGO



Step 1b:

Step along directions until we get to target time

Example:

At second step...

11 minutes left

Duration of Step: 35 min

Divide line segment using ratio: (11/35)

Start LatLong

Wilson Station

41.965320, -87.658203

End LatLong

Garfield Station

41.795421, -87.631424

Midpoint

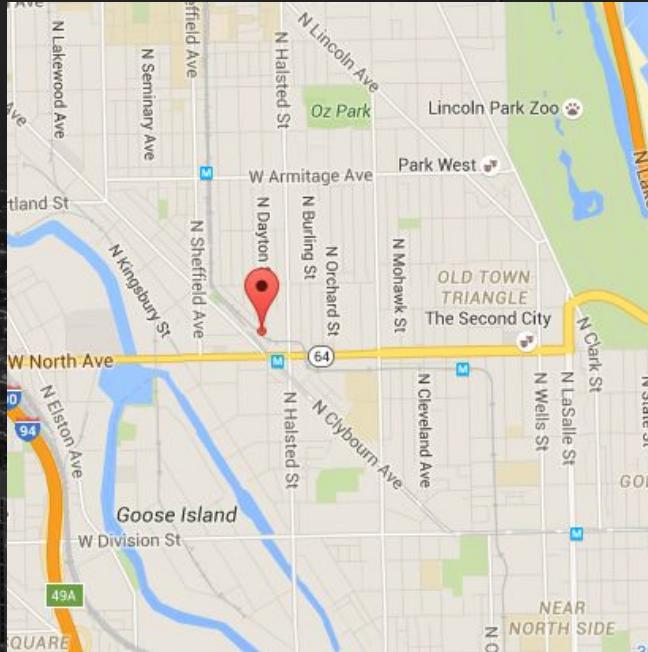
41.911923, -87.649787

Meet Halfway

CS 122 // March 10, 2016



THE UNIVERSITY OF
CHICAGO



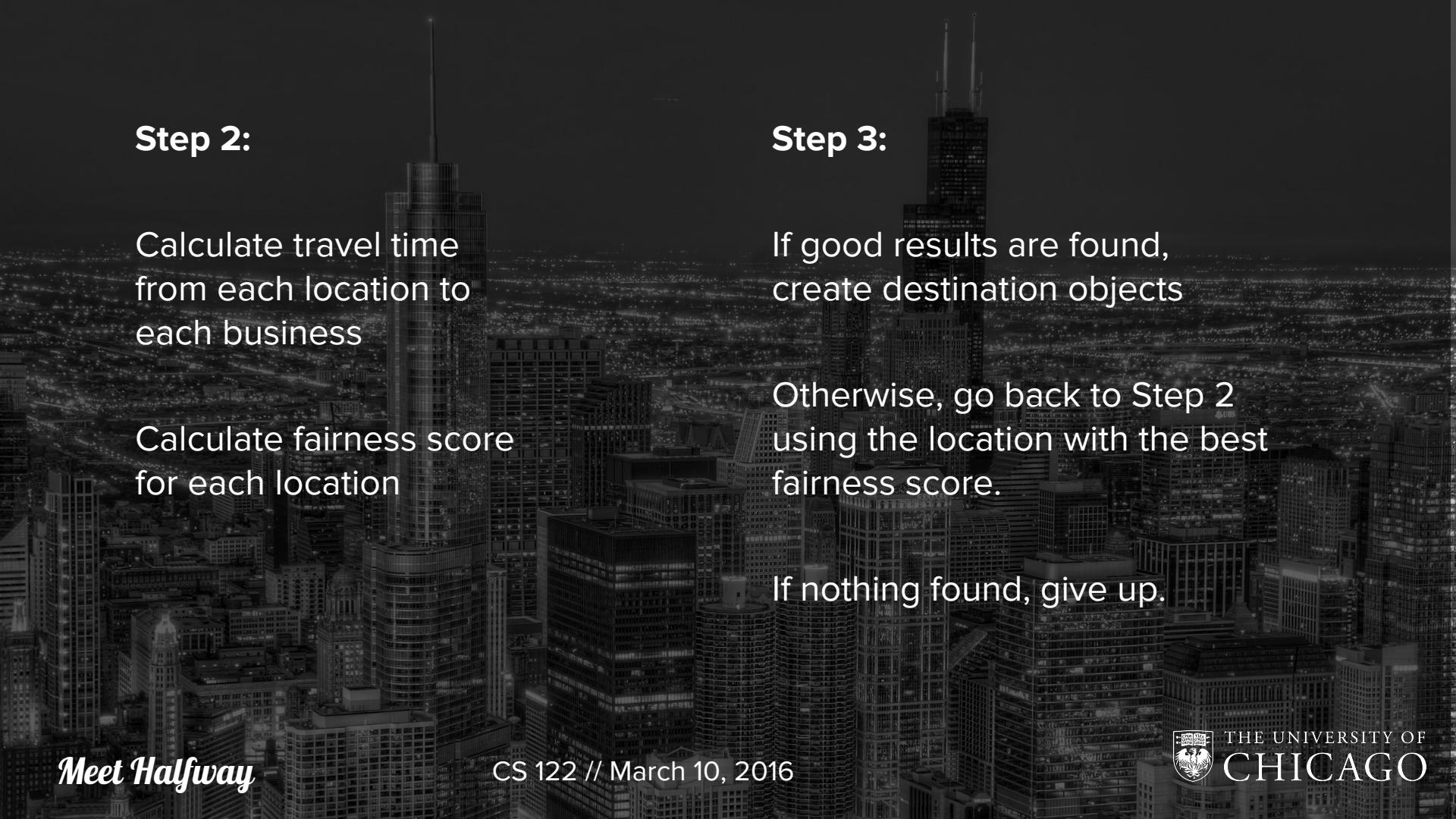
Meet Halfway

CS 122 // March 10, 2016

Step 1c:
Find all businesses
that match criteria
within certain radius of
midpoint



THE UNIVERSITY OF
CHICAGO



Step 2:

Calculate travel time
from each location to
each business

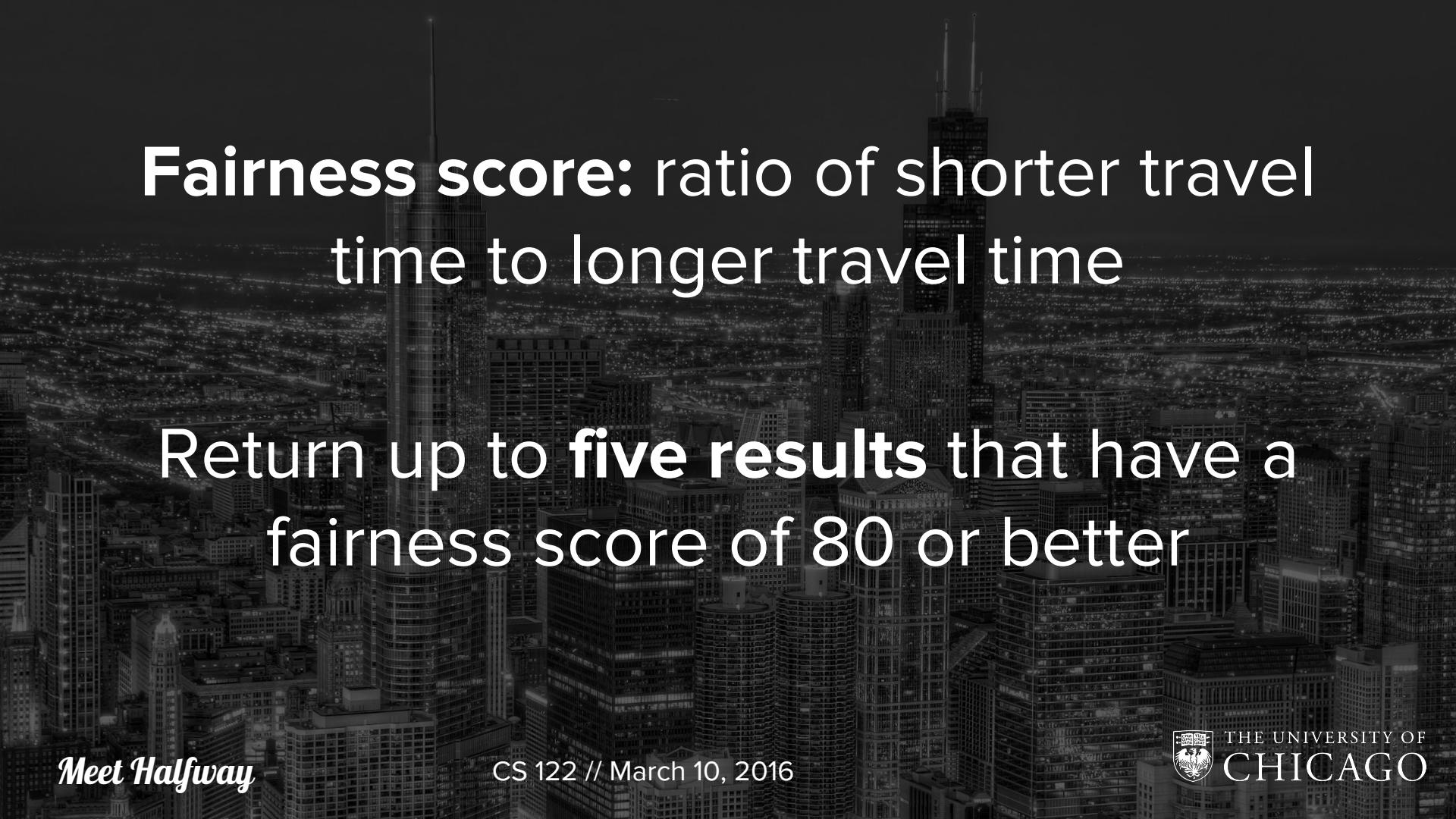
Calculate fairness score
for each location

Step 3:

If good results are found,
create destination objects

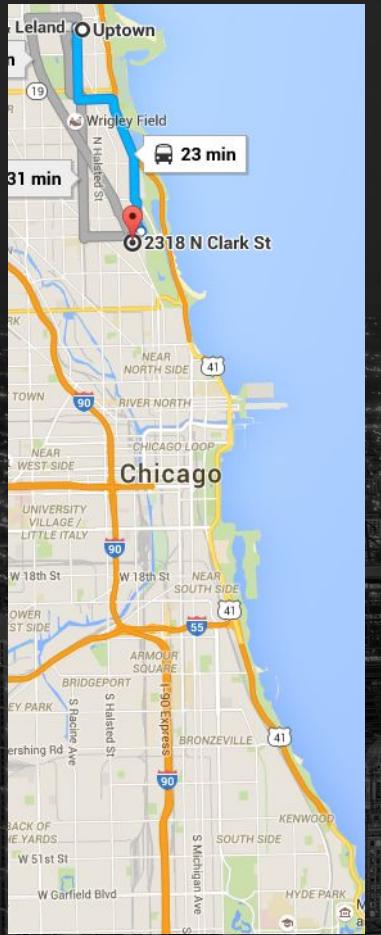
Otherwise, go back to Step 2
using the location with the best
fairness score.

If nothing found, give up.

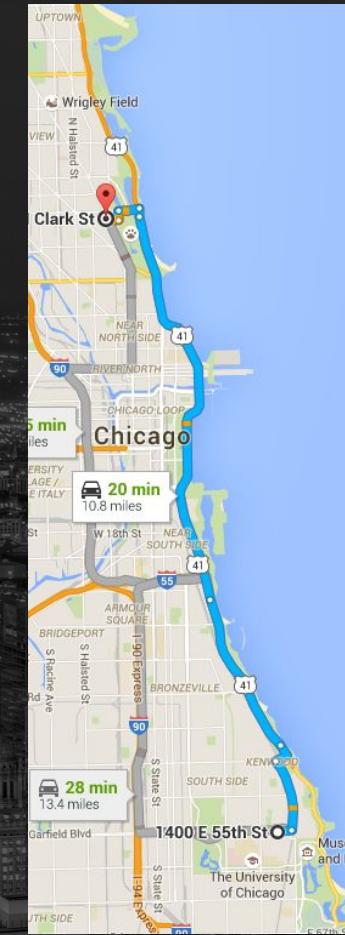


Fairness score: ratio of shorter travel time to longer travel time

Return up to **five results** that have a fairness score of 80 or better



Meet Halfway



CS 122 // March 10, 2016

Results

Meeting Location:

Deli Boutique Wine & Spirits

2318 N Clark St, Chicago, IL 60614, USA

Fairness score: 94

Approximate travel time: 21 mins

[View on map](#)

Travel Time for Leith by Bus: 23 Minutes

Travel Time for Lauren, Christine, and Paul:
20 Minutes



THE UNIVERSITY OF
CHICAGO

Address Verification

Method on Address Model

Error Handling

```
dict([
```

```
    ('address', '6406 IVY LN'),  
    ('city', 'GREENBELT'),  
    ('state', 'MD'),  
    ('zip5', '20770'),  
    ('zip4', '1441'),  
],  
ValueError(' -2147219400: Invalid City. '))
```

```
]
```



<https://github.com/thelinuxkid/pyusps>

Photo: Flickr user [ryanready](#)



Demo!

<http://127.0.0.1:8000/>

Photo: Flickr user [ryanready](#)

Meet Halfway

CS 122 // March 10, 2016



Challenges

- Collaborating with Github
- Resetting midpoint when no results found
- Dependencies
- Time of day variations
- Multiple API calls and speed
- Tradeoffs:
 - Optimal vs. Good Enough
 - Most Efficient vs. Most Equal
 - Exhaustive vs. Know When To Call It Quits
 - Accuracy vs. Flexibility

Photo: Flickr user [ryanready](#)

```
1|recapitulation-screams-galley|1||Coffee
2|men-bites-humps|2|3|Coffee
3|profession-holds-skirt|4||Coffee
4|tricks-visits-ornaments|5|6|Coffee
5|appropriation-exits-crop|7|8|cafe
6|house-sectors-entrapments|9|10|cafe
7|mercury-fillers-bulb|11|13|cafe
8|guns-services-calculation|14|15|cafe
9|lots-motors-escort|16||cafe
10|regrets-shouts-bodies|17|18|cafe
11|serial-dominion-locks|19|21|cafe
12|skins-salutes-solvent|22|25|cafe
13|memorandum-reinforcement-element|26||cafe
14|mates-chit-halls|27|28|cafe
15|throttle-overlay-bead|29|30|cafe
16|cans-dependence-society|31|32|cafe
17|fact-stream-conflicts|33|34|cafe
18|cries-soup-analyzer|35|36|cafe
19|knees-routine-tools|37|38|cafe
20|terrain-calculations-term|39|40|cafe
21|knife-health-acquisition|41|42|bar
22|scene-circulation-board|43|44|bar
23|benefit-duplicate-apportionments|45|46|cafe
```



Questions?

Find our project at github.com/ladyson/pllc