



# WalkMe NYC

A Multiplatform Safety-Enhanced  
Pedestrian Routing Application

Team Members Cissy Chen (cissyc@)  
Eric Li (eyli@)  
Julia Wang (juliahw@)

# Problem

- 1) Walking alone late in an unfamiliar city can be unsafe.
- 2) Some areas of a city might be safer than others while walking at night, but you don't have that information.

# **Current Solutions**

- 1) Just take a cab:**
  - a) But is it always worth the money when you're destination is within walking distance?
- 2) Look up 'unsafe' neighborhoods in NYC:**
  - a) Where would you look? Would the macro info be enough?
- 3) Just start walking:**
  - a) But you could feel uneasy about the paths Google Maps gives you

# Our Solution: WalkMe

Recommends the **safest** and shortest path  
from A→ B in NYC

- Based on historical crime data:
  - Coordinate-specific → hyperlocal precision
- Route Safety Estimation:
  - Severity of crimes
  - Number of crimes
  - Worst crime on path
  - Most crime-dense point on path

# Major Features

- 1) *Walking directions from A → B*
- 2) Route options rated with *Safety Index* for NYC
- 3) Compare safety to NYC averages - *gives context*

[Back](#)[About](#)

## What is WalkMe?

WalkMe recommends the safest and shortest walking paths from point A to point B based on historical crime data. Every route option comes with a Safety Index from **1 (very unsafe)** to **10 (very safe)**:



Let WalkMe be your personal escort – walk safe with WalkMe!

## Which cities does WalkMe serve?

Our database currently contains crime data for the New York City area, including Manhattan, Brooklyn, the Bronx, and Queens. The crime data was compiled from [maps.nyc.gov/crime/](http://maps.nyc.gov/crime/).

## Use with Caution:

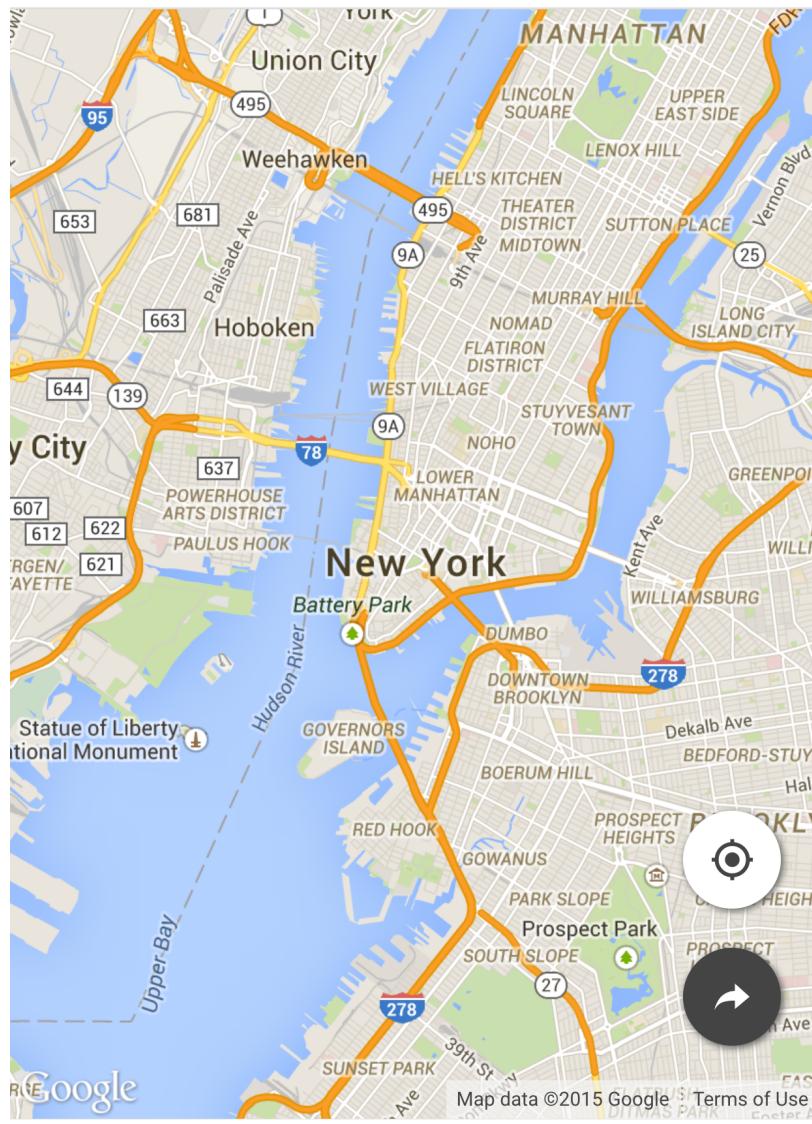
All Safety Indices are calculated purely from past reported and recorded crimes, given by the New York Police Department, but they are still only estimates. We do not take into account any other characteristics of the area other than past crime totals and types. Recorded crimes do not constitute a complete view of safety, so walk with caution – it's always better to be safe!

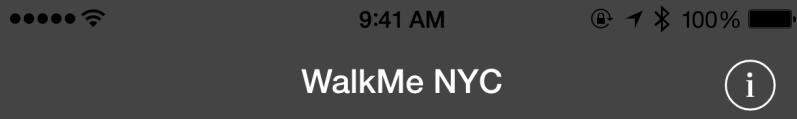
[Disclaimer](#)



Choose starting address...

Choose destination...





Per

X

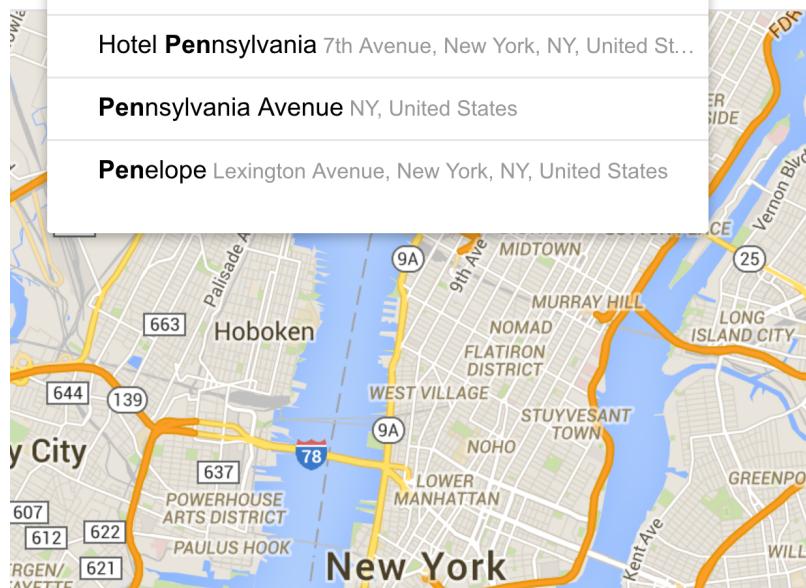
**Penn Station** New York, NY, United States

**Pennsylvania Station** New York, NY, United States

**Hotel Pennsylvania** 7th Avenue, New York, NY, United States

**Pennsylvania Avenue** NY, United States

**Penelope** Lexington Avenue, New York, NY, United States



<

>

Done

Q W E R T Y U I O P

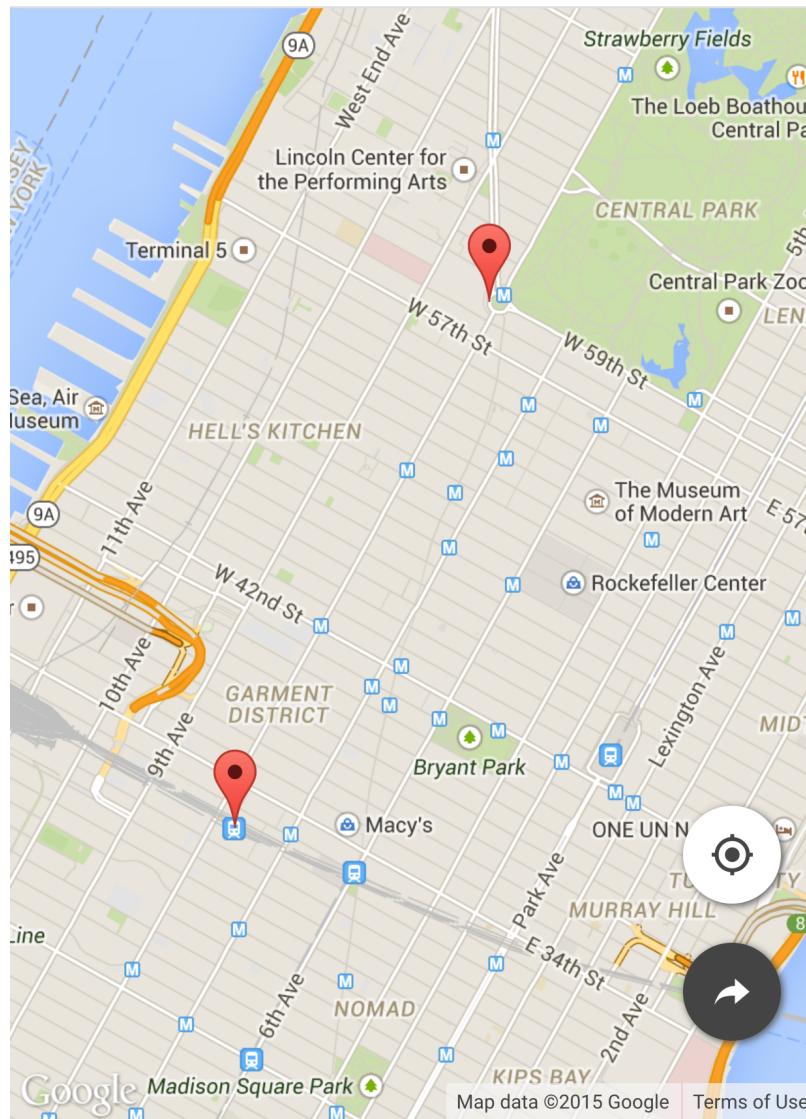
A S D F G H J K L

123 Z X C V B N M



space

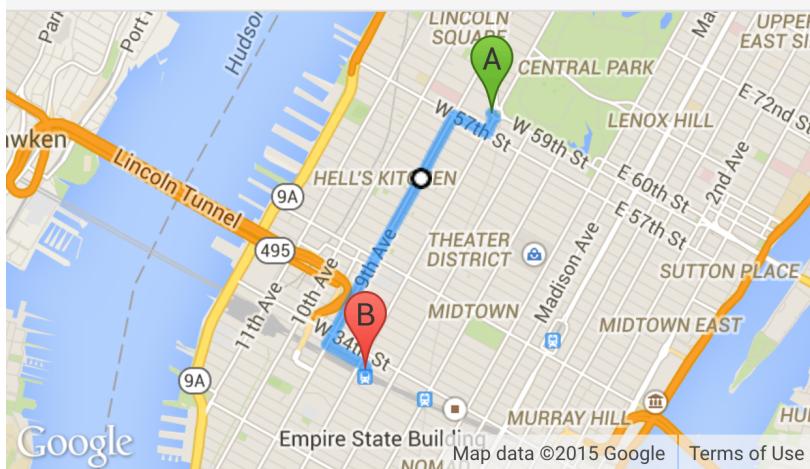
Go

Pennsylvania Station, New York, NY, United States XColumbus Circle, New York, NY, United States X

< Back

## Choose A Route

## Preview



## Options

**5.2** (?)33 mins (1.7 mi)  
via 9th Ave**4.8** (?)29 mins (1.5 mi)  
via Broadway and 7th Ave**4.8** (?)26 mins (1.3 mi)  
via 8th AveRequest an Uber

< Back

## Directions

## Directions

*Walking directions are in beta. Use caution – This route may be missing sidewalks or pedestrian paths.*



**278-280 West 33rd Street, New York, NY 10001, USA**

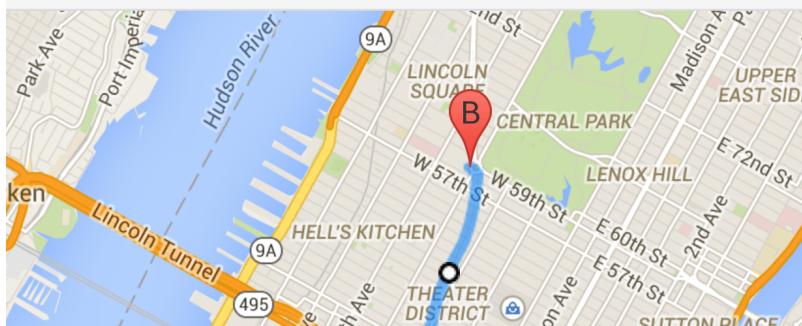
1. Head north toward W 33rd St 75 ft
2. Turn right onto W 33rd St 0.1 mi
3. Turn left at Drop Off onto 7th Ave/Fashion Ave  
Continue to follow 7th Ave 0.6 mi
4. Slight left onto Broadway 0.7 mi
5. Enter the traffic circle 217 ft  
Destination will be on the right



**80 Columbus Circle, New York, NY 10023, USA**

Map data ©2015 Google

## Map



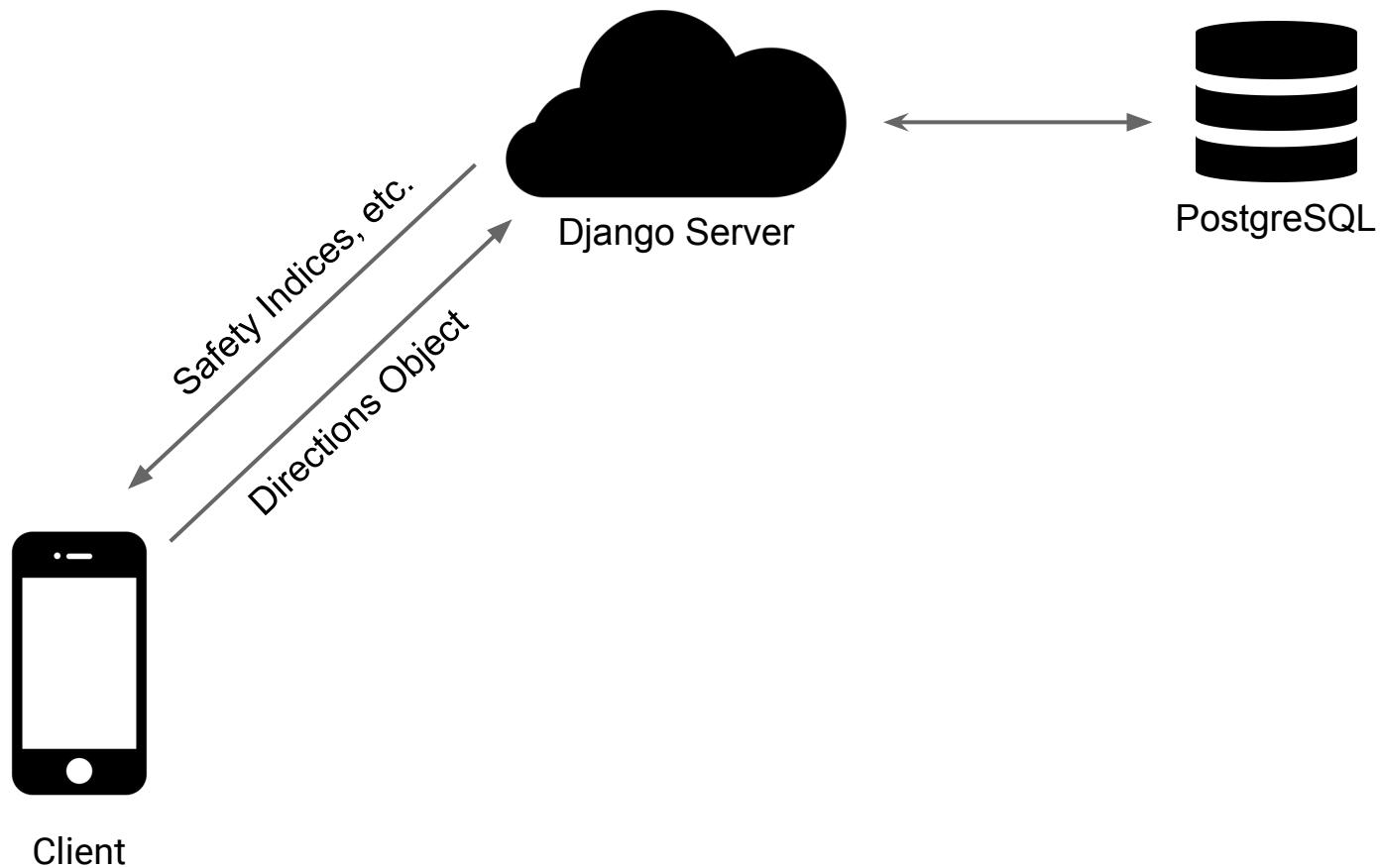
# Demo

[walkme.herokuapp.com](http://walkme.herokuapp.com)

# Implementation

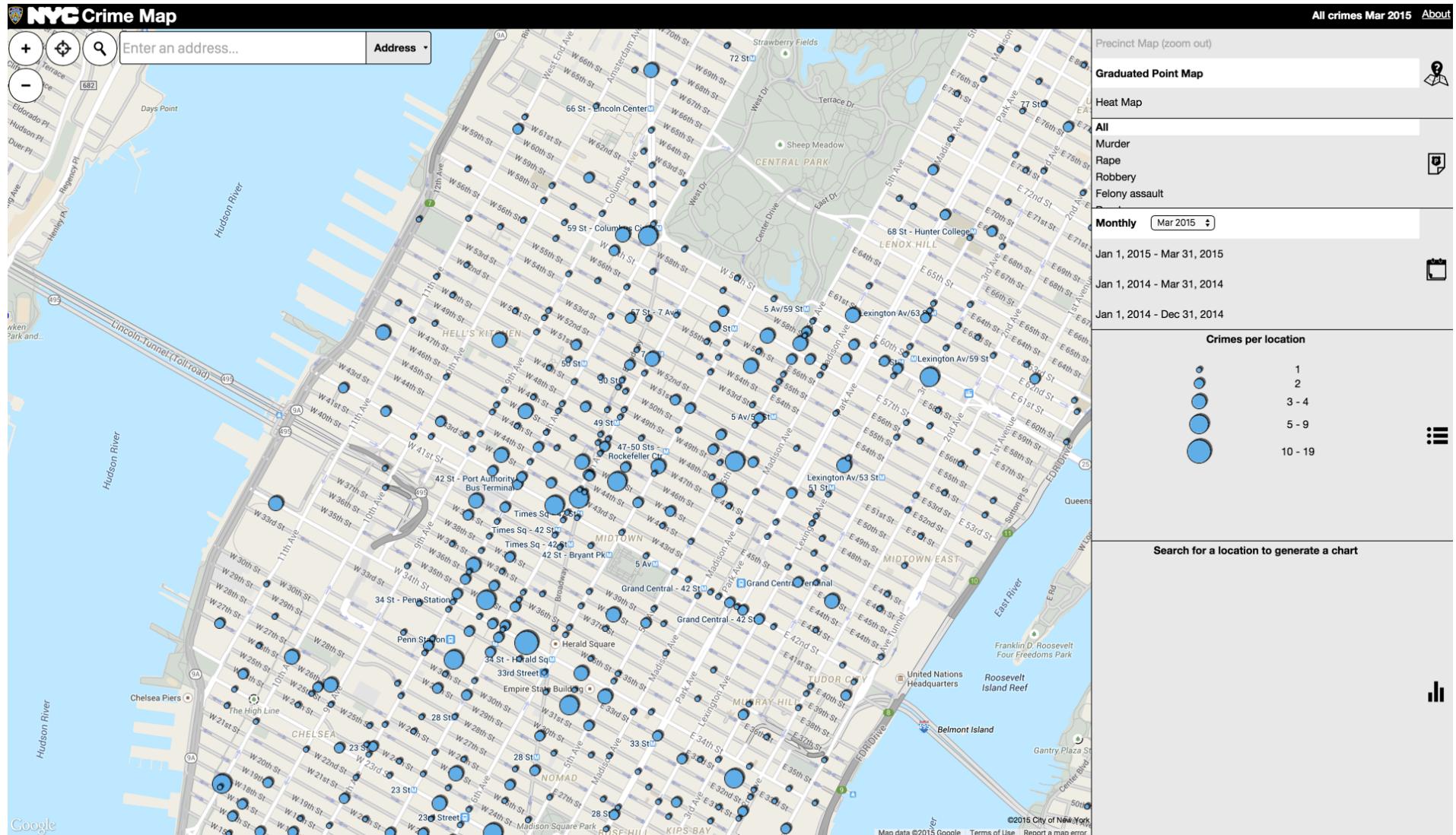
How we did it

# Overview



# Data Collection

- NYC Crime Data from: <http://maps.nyc.gov/crime/>
- Scraping of data: <https://github.com/tlevine/nyc-crime-map>



# Database Structure

## PostgreSQL + PostGIS

```
SELECT * FROM nyc_crime_2014 WHERE ST_DWithin(geom, ST_SetSRID(ST_Point(%s,%s),4326),%s);
```

xxv	longitude double precision	latitude double precision	yr integer	mo integer	x integer	y integer	tot integer	cr character(100)
116	-73.82421031	40.70010084	2014	10	1041023	190777	1	BURGLARY
117	-73.87955856	40.74809287	2014	8	1017622	211850	1	GRAND LARCENY
118	-73.87955856	40.74809287	2014	10	1017622	211850	1	GRAND LARCENY
119	-73.81206064	40.70632054	2014	6	1036357	196664	1	RAPE
120	-73.81206064	40.70632054	2014	8	1036357	196664	1	RAPE
121	-73.85464199	40.68506355	2014	10	1024564	188897	1	GRAND LARCENY
122	-73.81729312	40.7053683	2014	2	1034907	196314	1	RAPE
123	-73.81170375	40.58750625	2014	3	1036549	153377	1	BURGLARY
124	-73.84890098	40.67428839	2014	7	1026163	184974	1	FELONY ASSAULT
125	-73.8365734	40.69563963	2014	4	1029568	192759	1	FELONY ASSAULT
126	-73.88708765	40.87337519	2014	7	1015477	257492	1	GRAND LARCENY OF MOTOR VEHICLE
127	-73.87909114	40.73519204	2014	11	1017758	207150	1	BURGLARY
128	-73.82689162	40.69944894	2014	3	1032250	194152	1	FELONY ASSAULT
129	-73.82421033	40.70010088	2014	12	1032993	194391	1	MURDER
130	-73.82421033	40.70010088	2014	3	1032993	194391	2	BURGLARY

# Backend

## Django Server with RESTful API

```
JSON_ROUTE_OBJECT from Google Maps API  
wm/api/[JSON_ROUTE_OBJECT]
```

Returns in order of routes received

Returns: { "indices": [], "numCrimes": [], "severity": [] }

# Google Maps API

# Javascript Google Maps API v3 on Client

# Implementation: Frontend

Google Maps API

Angular.js

- MVC
- UI routing

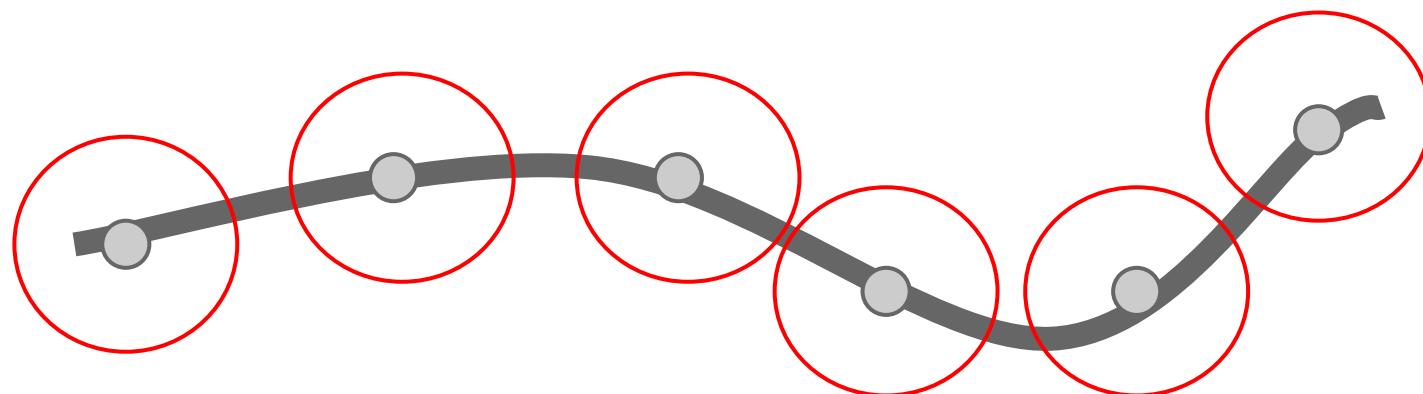
Ionic.js

- cross-platform
- iOS and web compatibility
- in beta for Android

# Heuristic for Route Safety Index

## Four Parameters:

1. avg. crime severity
2. avg. crime rate
3. worst crime severity
4. highest crime rate



# **Reflections**

# What we've learned

- 1) How to use new tools & frameworks
- 2) How to prioritize features
- 3) How to collaboratively build a project
- 4) Difference between designing UI for mobile  
and for web
- 5) Importance of usability tests

# Open Issues

- 1) Heuristic can always be improved
  - a) What's a better estimate of safety?
- 2) Unintended consequences
  - a) Effect on local businesses
  - b) Social controversy
- 3) Still doing usability tests to improve look & feel

# Future Plans

- 1) Expand beyond NYC -- for cities like Chicago, San Francisco, DC
- 2) Take time of day into account
- 3) Introduce our own routing algorithm:
  - a) Optimize for “safety” when finding a path within a certain distance limit
  - b) Allow user to weigh safety + distance



WalkMe NYC