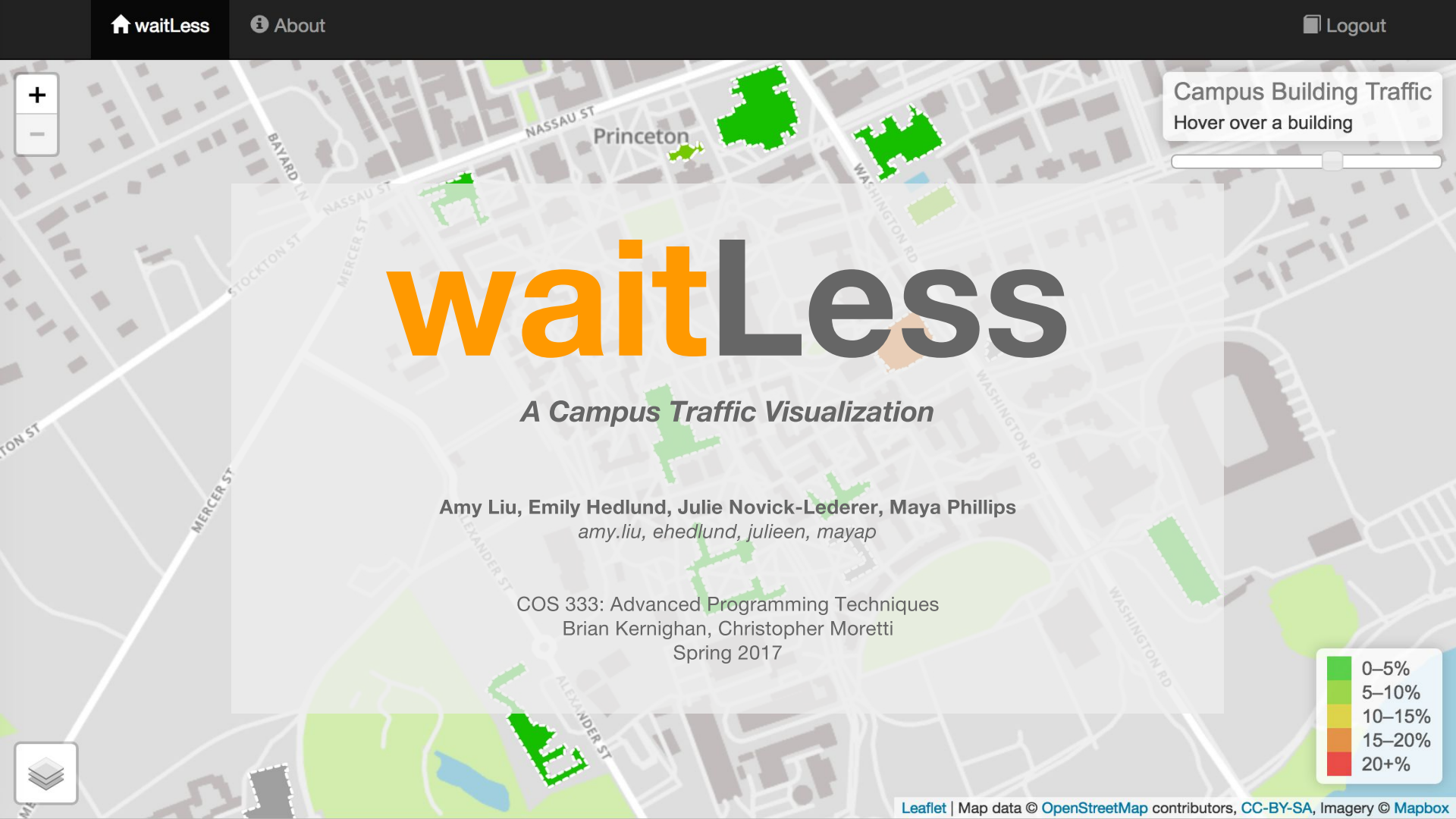


waitLess

A Campus Traffic Visualization

Amy Liu, Emily Hedlund, Julie Novick-Lederer, Maya Phillips
amy.liu, ehedlund, julieen, mayap

COS 333: Advanced Programming Techniques
Brian Kernighan, Christopher Moretti
Spring 2017

Project Motivation

- Busy student body
- Some places on campus are busier than others
- Small, random chunks of time
- Hate waiting in lines
- Need to maximize time
- A visualization of traffic data around campus is an untapped market

Project Goal

Our Vision

waitLess maps the daily flow of traffic through Princeton University's busiest spots on campus: dining halls, cafes, Dillon Gym, and Firestone

Product Market Fit

Is there a need for our product?

Busy, dynamic, always-moving student body

Shared spaces that have limited capacities

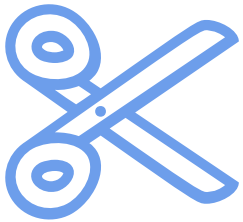
Knowledge of general traffic in popular locations is unknown by students

Who would use our product?

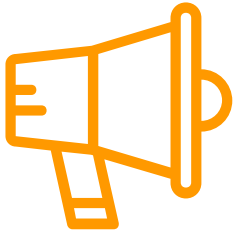
Intended product use for Princeton undergraduate and graduate students

Can expand to any university campus or other spaces/grounds where entry/exit can be tracked

General Design



- Acquired swipe data for each building of interest
- Parse and store data using Django as a framework



- Used Leaflet with GeoJson layers to map and highlight buildings of interest
- Dynamically changed colors of buildings based on densities of buildings at the time of loading the webpage

- Aggregated data into sliders and graphs to dynamically visualize traffic at other times of interest

Key Features

1. **Hover Over Building**: Use our hover function to see the name of the building and the number of people who swiped in from the last half hour.

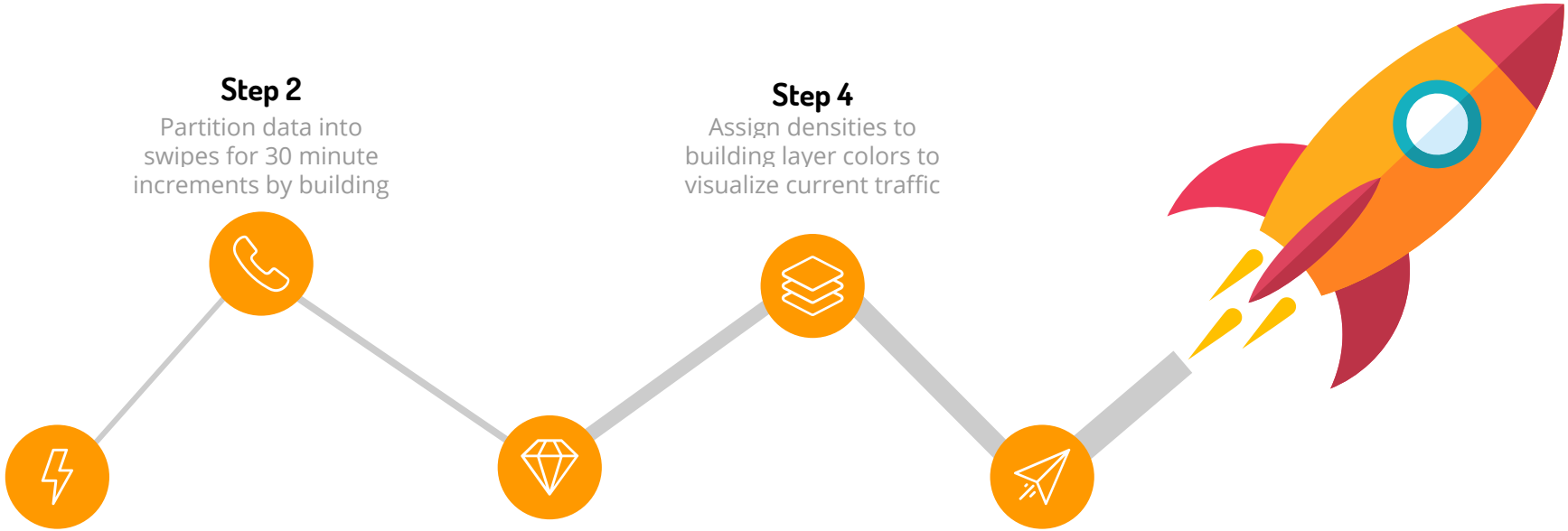
2. **Color Coding**: When a location is closed, its respective building on the map becomes grey. Open locations range in color depending on how busy they are.

3. **Building Filter**: Filter which types of buildings you want to appear on the map. Choose between Firestone, dining halls, cafes, or Dillon gym.

4. **Daily Traffic Graphs**: Click on a building to view a graph of the three-day traffic for that location.

5. **Slider Bar**: The slider bar changes the colors of the buildings to reflect how busy each building is at every time of the day from a high level view.

Process, Planning, and Execution Overview



Step 1
Collect TigerCard swipe data from campus staff

Step 2
Partition data into swipes for 30 minute increments by building

Step 3
Design UI using Leaflet and Mapbox with GeoJson layers

Step 4
Assign densities to building layer colors to visualize current traffic

Step 5
Added slider and filter features for different data visualization

CSV Processing

2	Atrium Cafe	8:32:38 AM	Monday
3	Atrium Cafe	8:32:38 AM	Monday
4	Atrium Cafe	8:38:05 AM	Monday
5	Atrium Cafe	8:38:05 AM	Monday
6	Atrium Cafe	8:46:27 AM	Monday
7	Atrium Cafe	8:52:56 AM	Monday
8	Atrium Cafe	8:54:04 AM	Monday
9	Atrium Cafe	8:54:04 AM	Monday
10	Atrium Cafe	8:54:38 AM	Monday
11	Atrium Cafe	8:54:56 AM	Monday
12	Atrium Cafe	8:55:52 AM	Monday
13	Atrium Cafe	8:58:35 AM	Monday
14	Atrium Cafe	8:59:40 AM	Monday
15	Atrium Cafe	9:36:48 AM	Monday
16	Atrium Cafe	9:47:47 AM	Monday
17	Atrium Cafe	9:51:23 AM	Monday
18	Atrium Cafe	9:54:58 AM	Monday
19	Atrium Cafe	9:55:46 AM	Monday
20	Atrium Cafe	9:56:50 AM	Monday
21	Atrium Cafe	10:09:12 AM	Monday
22	Atrium Cafe	10:09:12 AM	Monday

Figure 1: Example data, converted to CSV form

Use Python to parse data and partition into 30 minute increments per day per building

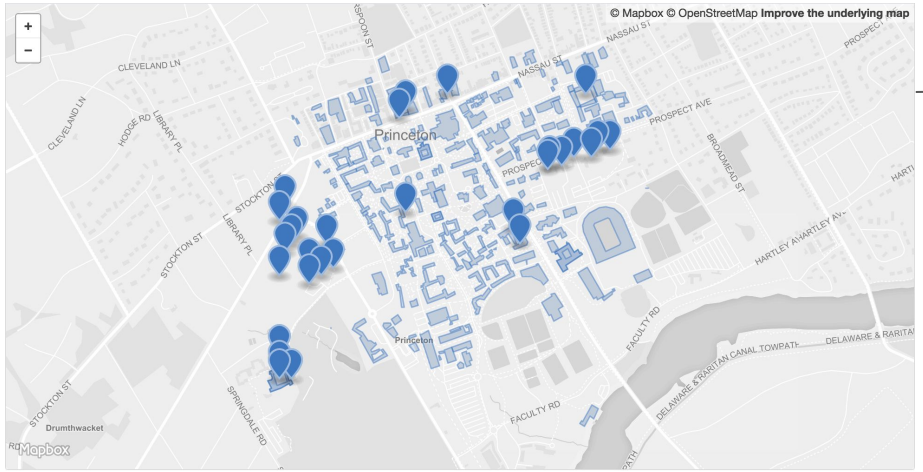
Store in array and convert to Json object

Pass Json variable to frontend HTML template

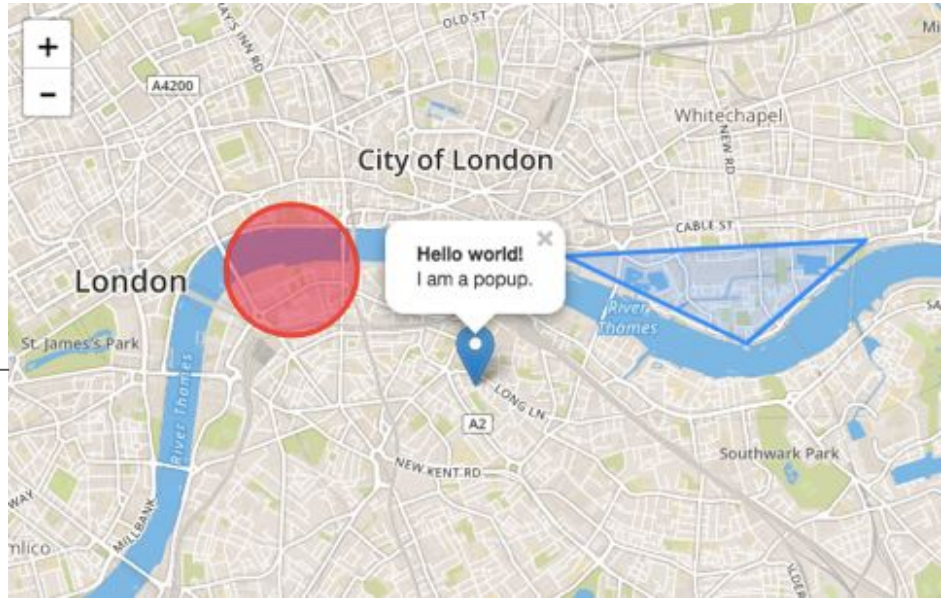
Access array with Javascript for current time and day for each building

Color code buildings based on density percentage of maximum occupancy

Maps



Mapbox



Leaflet

Demo

waitlessprinceton.herokuapp.com

Roadblocks

Making design decisions without knowing the format of our data

Acquiring data

Transitioning from wifi data to prox swipes

Lack of web development experience on our team

Solutions

Flexibility in our frontend design

Communication with a myriad of departments on campus

Reworking our backend code to reflect correct data and changing building locations

Patience, teamwork, and StackOverflow

Moving Forward

- Transition to live prox-swipe data
- Track exit data as well as entrance data for more accurate occupancy counts
- Expand to smaller libraries and other various work spaces on campus
- “Smart” recommendations of where to go when
- Implement waitLess on other college campuses in the U.S.

Acknowledgements