Team Name: A3POINT14
Members: Chloe Delfau, James Wang, Billy Wong, Haley Zeng
Project Manager: Chloe Delfau

Role Delegation:

- **Front end & User accounts: HTML/CSS (bootstrap), user account functionality**
- **Weather: Gathering forecast data (OpenWeatherMap and/or DarkSky API – https://darksky.net/dev/docs) and displaying it**
- **Transit: Gathering transit data (Subway – MTA API by Herokuapp, http://nycpulse.herokuapp.com/api, Bus – MTA BusTime API) and displaying it**

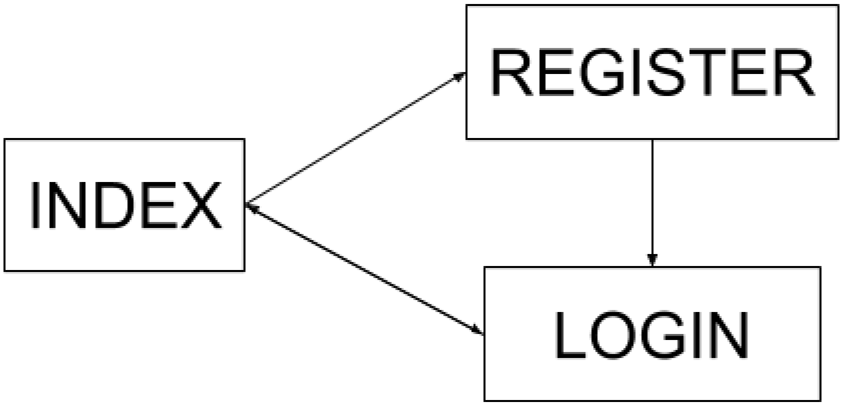
Purpose of the project: Create a website for one-stop information including transit and weather data for those mornings when you are running late because you hit the snooze button too many times or didn't hear the 13 separate alarms you set (i.e. every morning…)

Functionality

- See real-time MTA subway, bus and rail status as well as schedules
- See weather forecast for specified location
- Make an account
- Personalize it with your most frequent locations

Pages and Site Map

- Index.html
  - search bar for users to look up their location for weather
  - options to display subway info, bus info, and/or LIRR info
  - if user is logged in, shows option to save settings
  - if settings are saved, will automatically present information when user visits page (does not need to re-search/re-choose options each time)
- Login.html
  - log into an account
- Register.html
  - register an account



Components & Component Map

app.py

- Holds the routes
- Uses weather.py & transit.py to present information
- Uses accountManager.py to handle user accounts

weather.py

- Uses OpenWeatherMap API & Dark Sky API
- Info shown defaults to Dark Sky API
- If request limit is hit, then info from OpenWeatherMap API is shown
- OpenWeatherMap API gets approximate GPS coordinates for Dark Sky API to use
- Gives weather information based on certain zip code

transit.py

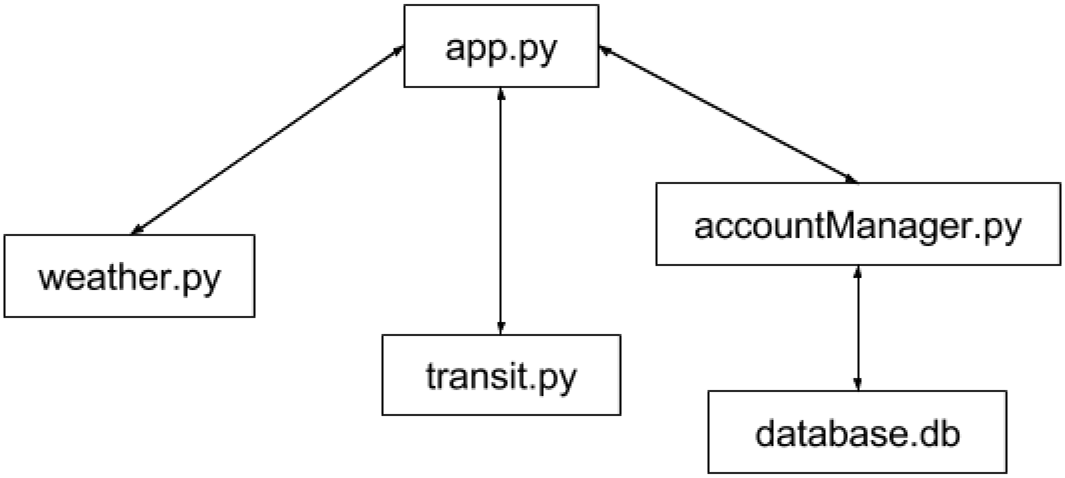
- Uses MTA API
- Used to get information about real-time status of subway, bus, rail as well as schedules

accountManager.py

- Includes authentication functionality for registering/logging in
- Talks to the sqlite database

index.html

- Html for the home page
- Includes jinja templating to allow for different blocks if user is/isn't logged in, has/does not have saved settings



Database Schema: 1 database, 1 table

Username - **TEXT**	Preferred ZIP code - **INTEGER**
Hashed Password - **TEXT**	Preferred transit data choices (Y/N display subway, bus, rail data) - **TEXT**