

Crazy Joe Speaks Frogs:: Kevin Wang (PM), Shreya Roy, Faiyaz Rafee,
Justin Mohabir
SoftDev
P01 -- ArRESTed Development
2022-12-4
time spent: 1 hrs

Target ship date: W **2022-12-21**

Design V.0 for GOOGOL MAPS!

Goal:

Create boot-leg Google Maps. Address inputted from the client is then converted into coordinates for city bike and overpass API. Returns nearest bikes and transit within 5 miles of the client's inputted address.

Roles:

API stuff: Shreya
Frontend: Justin
DB management: Faiyaz
Backend & PM: Kevin

Frontend Framework Chosen: Bootstrap

- Why:
 - Bootstrap is easier to work with & looks cleaner.
 - Bootstrap containers are epic
 - Foundation is bad :D
 - Using only flexbox rather than grid/flex grid/etc
- How:
 - Containers to make sure everything is neatly organized in accordance with screen size
 - Forms to accept user input for a location or address

APIs Utilized:

- https://github.com/stuy-softdev/notes-and-code/blob/main/api_kb/411_on_PositionStack.md - Address -> coordinates, 25000 quota & key
- https://github.com/stuy-softdev/notes-and-code/blob/main/api_kb/411_on_city_bikes.md - coordinates -> nearest citibikes, no quota no key

- https://github.com/stuy-softdev/notes-and-code/blob/main/api_kb/411_on_OpenStreetMap_overpass-api.md - coordinates -> nearest transit, no quota no key

DB:

Authentication

Username	Password
dev01	hehehaha
dev03	gabagoo

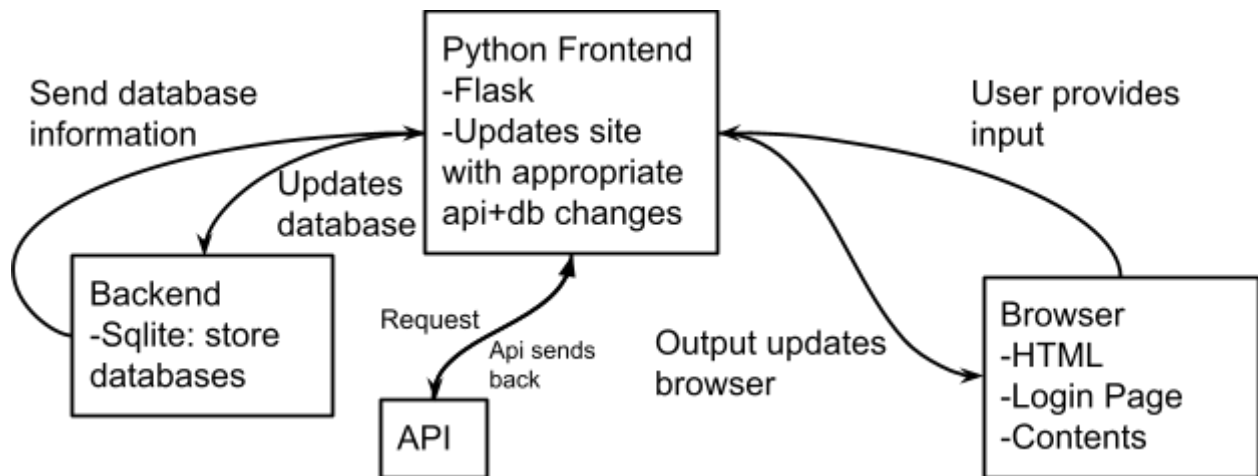
Past Searches

username	Past Searches
ThonnySinclair	345 Chambers Street
ThonnySinclair	1 Empire Avenue

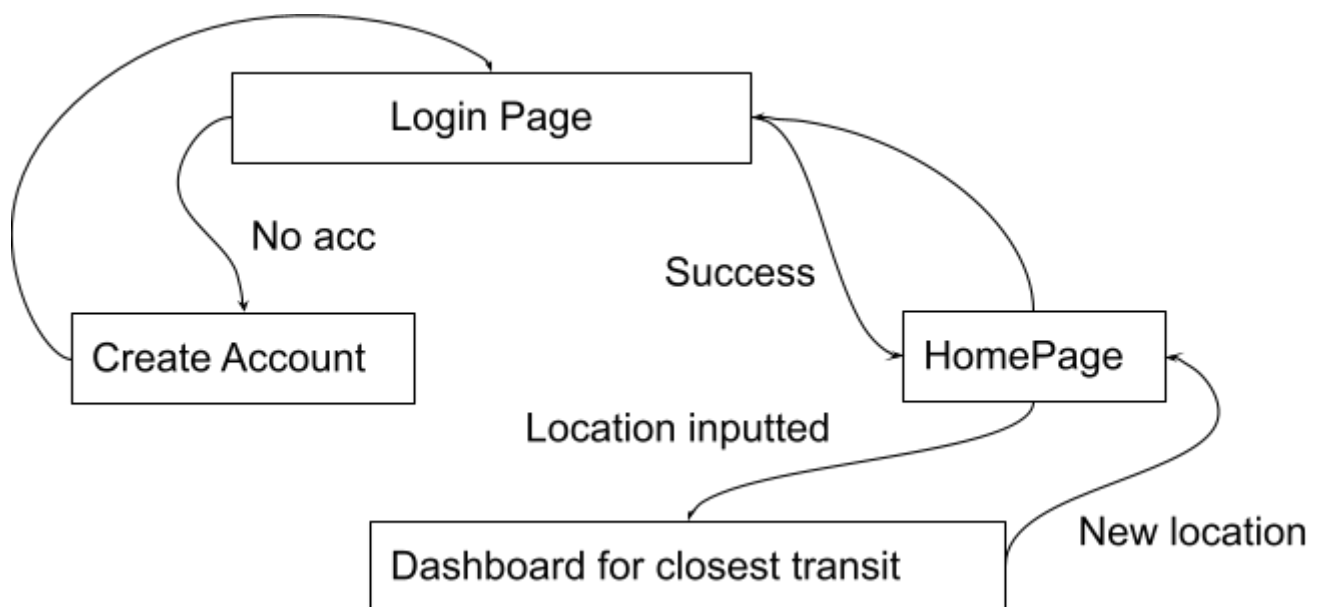
Program Components:

- `__init__.py`
 - Where the flask magic happens
 - Renders templates
 - Uses `hehehaha.py` to interact with the db and handle username/password authentication, as well as past searches
 - Takes in user changes -> sanitizes it from methods in `api.py` -> retrieves data from apis from methods in `api.py` -> renders templates from it
- `gabagoo.db`
 - A table for usernames and passwords
 - A table for storing username and all searches
- `hehehaha.py`
 - Methods for updating, viewing, and creating new entries in the db
- `Api.py`
 - retrieving info from apis
 - Contains methods for cleaning user input into either an address, location name, or latitude and longitude

Component Map:



Site Map:



Templates:

login.html
register.html
home.html
dashboard.html

Flask:

__init__.py

/home:

- Access argument username and all searches
- Display all search results in home.html template

/login:

- Access arguments username, password
- Checks with form inputs
- Logs in/error message if incorrect username/pswd

/register:

- Access arguments username, password and append data to users table

/dashboard:

- Shows closest locations
- Shows distances to city bikes and transit