Radioing! In the Rain

Design Document

Team NYCDOE Blocked Website:

Shamaul Dilmohamed, Kate Ly Johnston, Vanna Mavromatis, Michael Woo

SoftDev pd9

I. OVERVIEW

We envision a project that tries to accurately match the connection the weather has on one's emotions through song. To accomplish this task, we plan to use both a weather API as well as a music streaming API, grabbing the location of a user to assist in choosing an appropriate song for the weather of that location. To assist our grand endeavours, we also plan on using the Google Maps API, getting the coordinates of the user through their location services. We plan to use a database in order to hold the location information for each city for maximum efficiency. In addition, we might allow users to create an account to keep track of weather and music combinations that they had in the past. We hope our project is a mindblowing experience.

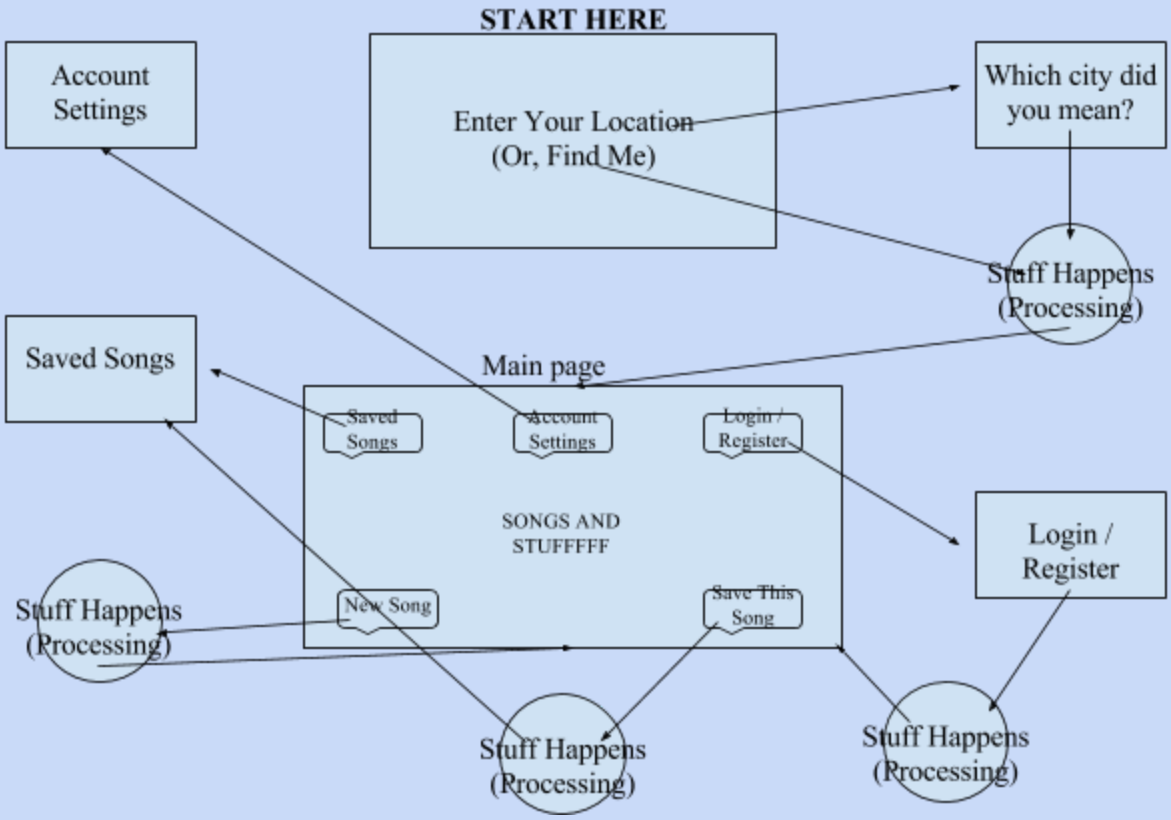
II. DESIGN / ELEMENTS

APIs Used

- Open Weather Map
    - https://openweathermap.org/api
    - Accepts cityid (search, find in our database)
    - Accepts longitude and latitude
- Google Geolocation
    - https://www.googleapis.com/geolocation/v1/geolocate?key=YOUR_API_KEY
    - Returns longitude and latitude

- Shoutcast
    - http://wiki.shoutcast.com/wiki/SHOUTcast_Developer
    - Radio stations separated by genre
    - http://api.shoutcast.com/legacy/genresearch?k=<KEY>&genre=<GENRE>
        - Returns station id
    - http://yp.shoutcast.com<base>?id=<STATION_ID>
        - Base is format required (PLS, M3U, XSPF)

Site Map



Component Map

App.py

Renders templates

Calls proper functions/files/etc. at proper times

Users.py

In charge of logging in, dealing with users database (account preferences, saved songs, etc.)

Processor.py

Takes care of finding location, weather in location, and matching the two

Frontend Files:

main.html

streamingPage.html

login.html

savedSongs.html

accountSettings.html

search.html

style.css

Databases

- Cities
  - name, cityid, longitude, latitude
- Users
  - username, hashed password
- Saved Songs
  - song id, user, cityid
- Weather
  - precipitation amount, precipitation mode, temperature, weather conditions, genre

III. TASKS / TIME FRAMES

Kate

- Project Manager
- Frontend

    - Basic pages without css: 12/6
    - CSS: 12/11
- Google Maps API (longitude and latitude)
    - 12/7

Shamaul

- Flask backend
- Putting website together, making sure everything links together, etc.
- Due along with relevant pieces of website
    - Add to flask as other pieces come together

Vanna

- Weather database
- Playing new, random radio station to match weather
- Initial due: 12/8
    - Algos can change without basic functionality changing

Michael

- Login
    - Due: 12/7
- Account settings
    - Due 12/9
    - Work with frontend, as well
- Saved songs
    - Due 12/9