Ivan Yeung, Vivian Graeber, Jeff Chen, Brian Chen (Team soup noodles)

Soft Dev

P01

Target ship date: 2022-12-16

To touch grass or to not touch grass?

Program Description

A site to determine if you should go outside today based on user preferences.

Program Components

A. Python Files
  a. database.py
    i. get_uid(username): Retrieves user id from username
    ii. get_password(username): Retrieves password from username
    iii. get_username(user_id): Retrieves username from a user id
    iv. get_weather(user_id): Retrieves the if user cares about weather or not
    v. get_league_pref(user_id): Returns how much the user likes league
    vi. get_curfew_pref(user_id): Returns how much user cares about curfew if they have one
    vii. get_anime_pref(user_id): Returns how much the user likes anime
    viii. get_curfew(user_id): Returns a user's curfew
    ix. get_anime(user_id): Returns user's favorite anime
  b. api_info.py
    i. get_sunrise(user_location):
    ii. get_sunset(user_location):
    iii. get_weather(user_location): Returns weather of current location (city)
    iv. get_lol_clash(): Returns List of info for next or current clash tournament(Name and schedule)
    v. get_anime_date(anime): Returns anime date
  c. app.py
    i. Flask
      1. @app.route("/"):

    a. redirect to /login
2. @app.route("/login"):
    a. renders login.html
3. @app.route("/login/auth"):
    a. login form: username & password
        i. check for existence of username and validity of password
4. @app.route("/register"):
    a. renders register.html
5. @app.route("/register/auth"):
    a. register form: username & password
        i. check for availability of username
        ii. if account is successfully created, information is stored in database
6. app.route("/home"):
    a. directs to a page that allows the user to go to the page where they can access other pages
    b. display content that is potentially interesting to the user(maybe?)
7. app.route("/pref "):
    a. directs to a page that allows user to customize their preferences
8. app.route("/grass"):
    a. runs the algorithm that determines if the user should go out on the particular day
        i. Each factor has points associated with it.
        ii. Factors that would keep the user in add points to the total while factors causing the user to go out will deduct points from the total.
        iii. The amount of points that a factor gets is based on if there's anything occurring that might interest the user.

iv. It is also affected by how much the user is interested in the activity (1-10), changing the weight of the points
v. If total points exceed a certain threshold, the user should stay inside, otherwise the user should go out.

b. Returns page with results + activities

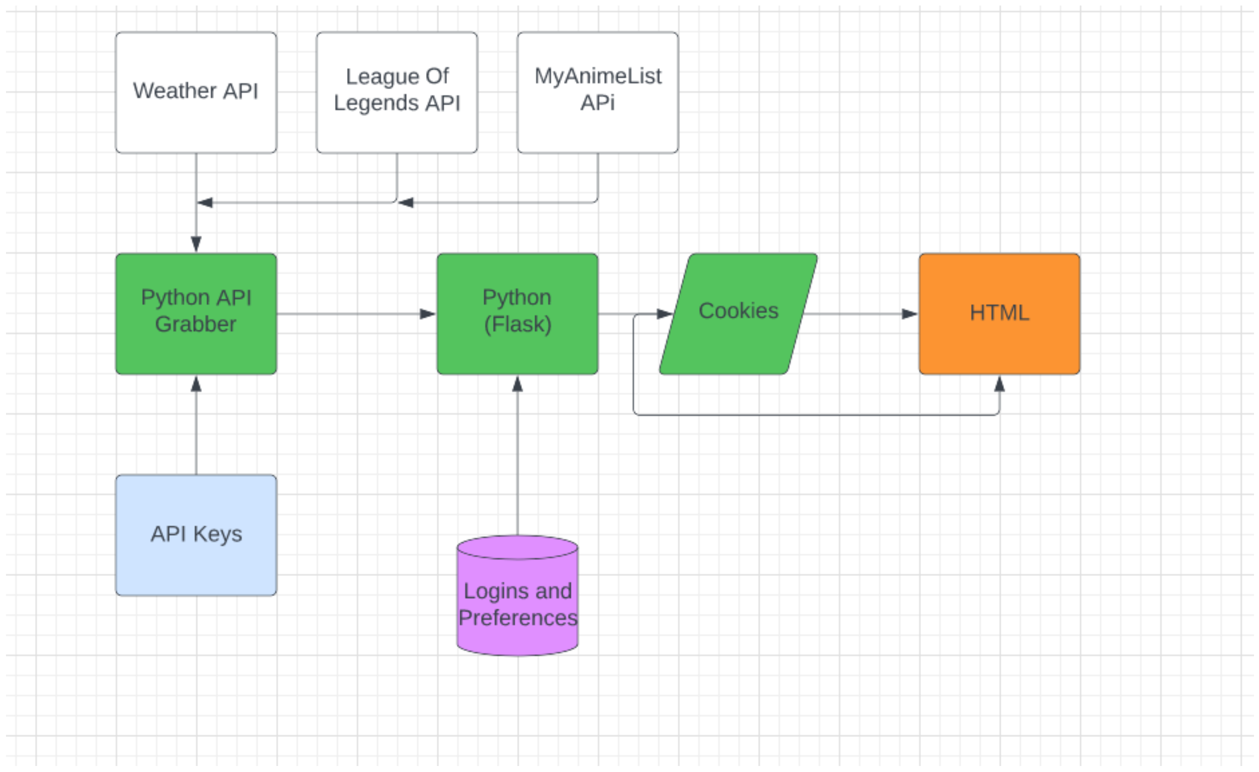
9. app.route("/info"):
   a. Serves the pages with relevant information of the topics we are working with

ii. Sessions
   1. Session["user id"]: Stores the username of the user that is logged in

B. Html Files (Bootstrap)
   a. login.html
      i. form for username and password
      ii. Info about our site
   b. register.html
      i. form for username and password
   c. preferences.html
      i. Check boxes for different topics that user can show interest in
      ii. Form to enter curfew
      iii. Sliders to show amount of interest for each supported topic
      iv. Form to enter city/region that user lives in
   d. grass.html
      i. Information about individual topics

C. Misc.
   a. key_weather.txt
   b. key_LOL.txt
   c. key_MAL.txt

Component Interactions/Component Map



Database Organization

Logins

Username	UserID	Password

Preferences

UserID	League (Clash Tournament)	Curfew (Maybe)	Weather (Cut for now)	Anime
	0-10	0-10	0-10	0-10

User Info

UserID	Location	Desired Curfew	Favorite Anime

APIs

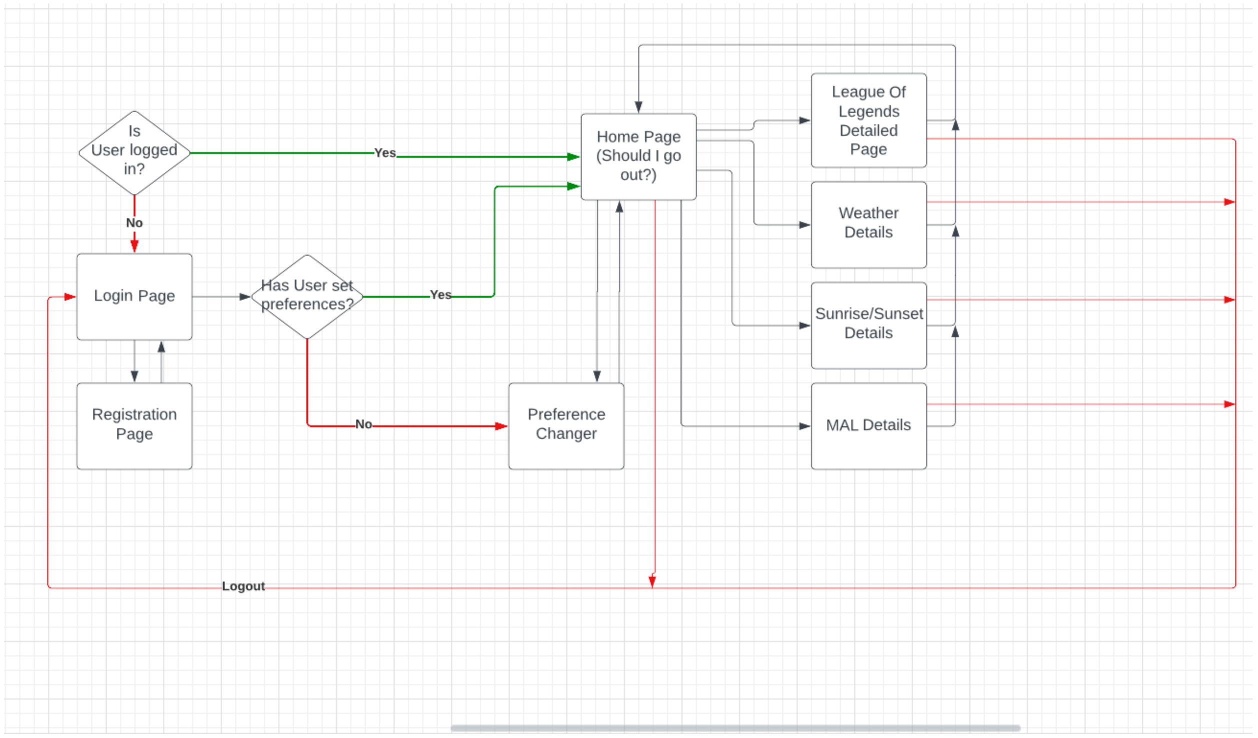
- Weather API
- Myanimelist api
- Riot API
- NBA schedule API(maybe)

Bootstrap

We are using bootstrap because the style appeared more modern and clean.

- Navbar at the top of each page with links
  - Dropdowns for individual preferences on navbar
- Bootstrap forms to provide information
- General styling and information placement
- Checkboxes

Site Map



Task Breakdown (Strikethrough as we complete)

- ~~Create design doc~~

- Revise design doc
- Write Python to pull API data (*Jeff*)
    - Confirm all APIs work
        - Test by having all data from API put on a throwaway HTML file
    - Functions to retrieve information from APIs
- Write Python to serve the HTML (*Vivian*)
    - Cookies to store user login status
    - Login + registration
    - Some sort of algorithm to determine whether user should touch grass or not
- Create database (*Ivan*)
    - Login storage
    - Preferences storage
    - Functions to retrieve data from database
- Create HTML (*Brian*)
    - Login Page
    - Registration Page
    - Preference Changer
    - Pages that show relevant information about certain topics(based on the APIs we are using)
    - Should I go out? page
        - Have API update (constantly or set interval)
    - CSS! (Bootstrap)
    - Individual API pages
    - Create API cards for APIs not already in database
- TEST throughout the process!!!