Spaghetti Rat: Lauren Lee, Brianna Tieu, Emerson Gelobter, Nada Hameed
SoftDev
P01 – ArRESTed Development
2022-12-07
time spent: 2 hrs
target ship date: 2022-12-19

Website: Dating App

Database Organization

user.db

Unique Username	Password

profile.db

username	name	birthday	star sign	height	hobby 1	hobby 2	spotify	gender	mbti

preferences.db

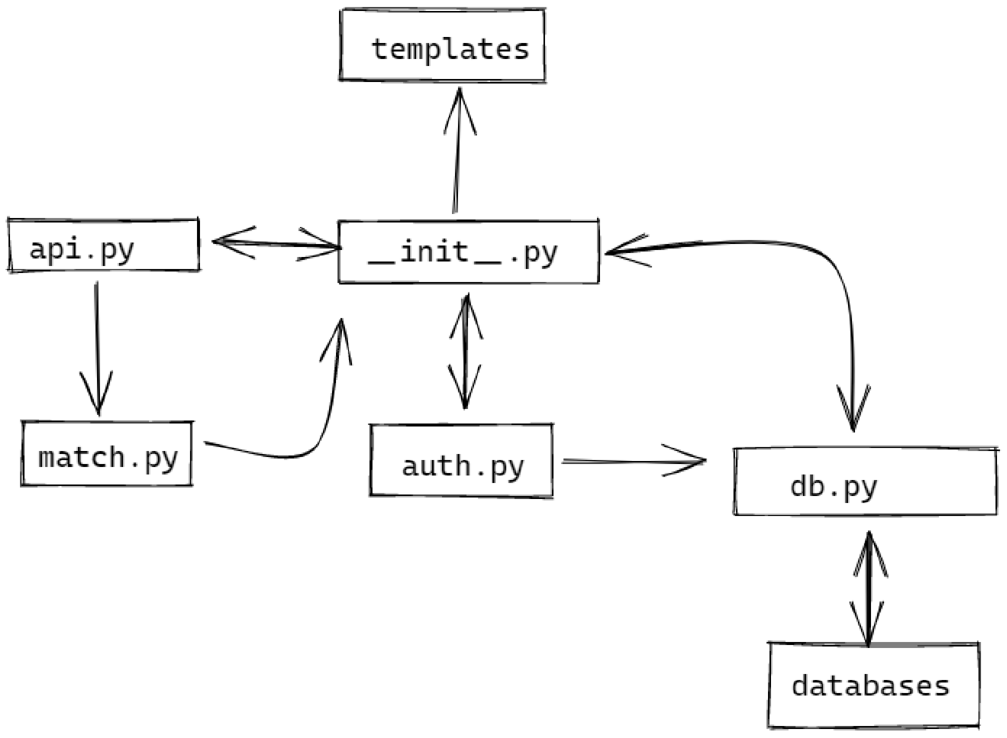
username	star sign	height	hobby 1	hobby 2	gender	bad star sign	bad mbti	mbti

- Users can opt to not consider star sign, mbti, or height
- Star sign is automatically calculated from birthday if opted in

List of Program Components:

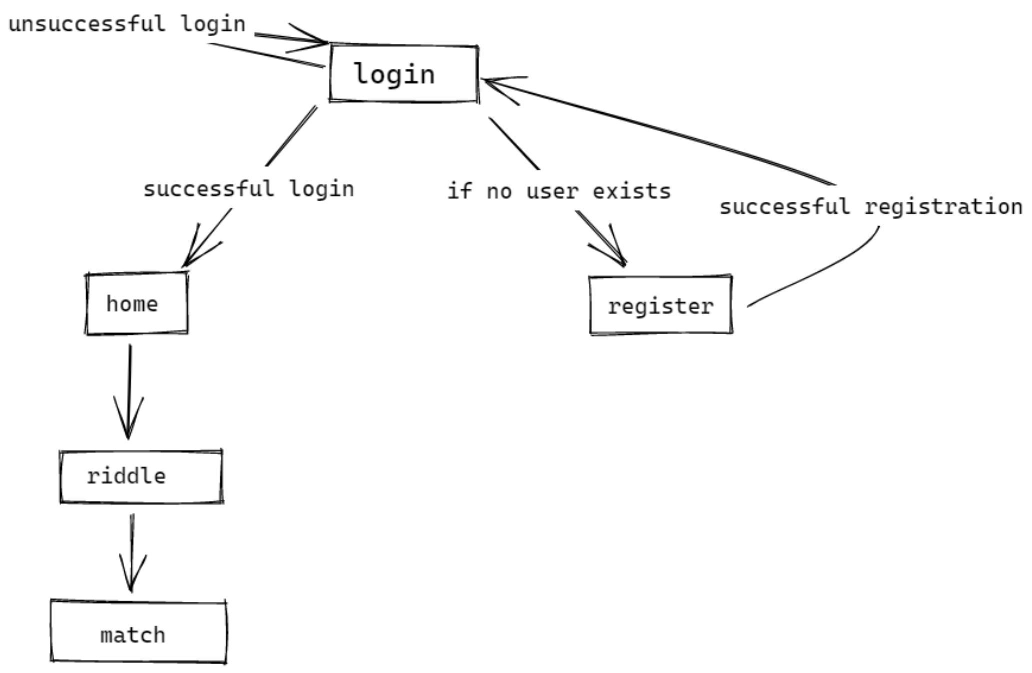
- **__init__.py** – run the flask app → calls functions from match.py, auth.py, db.py, and api.py
- **match.py** - calculates matches based on preference and profile db information
- **auth.py** - login and register
- **api.py** - pulls from apis which will contribute to match data
- **db.py** – creates and manipulates the user, profile, and preference db
- **profile.db** – holds user profile data

- **preference.db** - holds user preference data
- **user.db** - holds user login data
- **Templates**
    - **login.html** - display login page
    - **register.html** - display register page
    - **profile.html** - display profile page which allows users to edit their profile data
    - **home.html** - display home page with match instructions
    - **riddle.html -** display riddle that *must* be solved before viewing matches
    - **match.html** - display page with compatibility match
        - Display all users in the db in decreasing compatibility score



Site map for front end:

- **/login:** this is the starting page for the site, with forms to submit your username and corresponding password. It also has a path to registration
- **/register:** where new users can create a new account. Users must submit a unique username, their name, date of birth and password. Leads back to the login page, where users must log in with their new credentials.
- **/profile:** this is where you put in all your information that allows for others to match with you. You can add personal, editable information, like your star sign, gender, and height. You can also add other facts about yourself, like your Spotify playlist, hobbies, etc. There will also be a heading for your preferences in a partner, like height, hobbies, star sign, and [something else].
- **/home:** This is where all your matches that you have unlocked are displayed with your compatibility score. You can click on the matches to see their information. There is also a button to find a new match. Matches will only occur if you are above a certain compatibility score.
- **/match:** where users can see their matches. However, users must solve a riddle to access the information of their selected match. - each preference has a certain weight in the calculation



A breakdown of the different tasks
Lauren Lee - database manipulation, api integration
Brianna Tieu - html templates, Flask app
Emerson Gelobter - html templates, Flask app
Nada Hameed - database manipulation, api integration

APIs:

- Love Calculator API - would calculate the compatibility between two users by inputting their names into the calculator
- Yes/No API - gives the user a random yes or no answer, will be used to determine whether or not the user receives a match
- Riddles API - user must solve a generated riddle to "unlock" their match and find out more information about them

Which front-end framework you will use and why/how?

We are using Foundation mainly because of its documentation and web editing, which we have found to be easily accessible and convenient.