

Target ship date: 2022-12-23

Squirtle Squadron, Andrew Piatetsky, Ayman Habib, Weichen Liu, Jeffrey Zou, Raven (Ruiwen) Tang
SoftDev
P01: ArRESTed Development Design Doc
2022-12-06

Feedback From other teams:

Add api cards - in progress

~~Rethink tables~~

Revamp site map

~~Add spotify API~~

~~Explain more about APIS~~

Rethink routes

Add templates

~~Revamp roles~~

Idea: Create a higher-lower game that involves apis that would retrieve that amount of followers an influencer has or of similar variety. We could possibly add some mini-games alongside it.

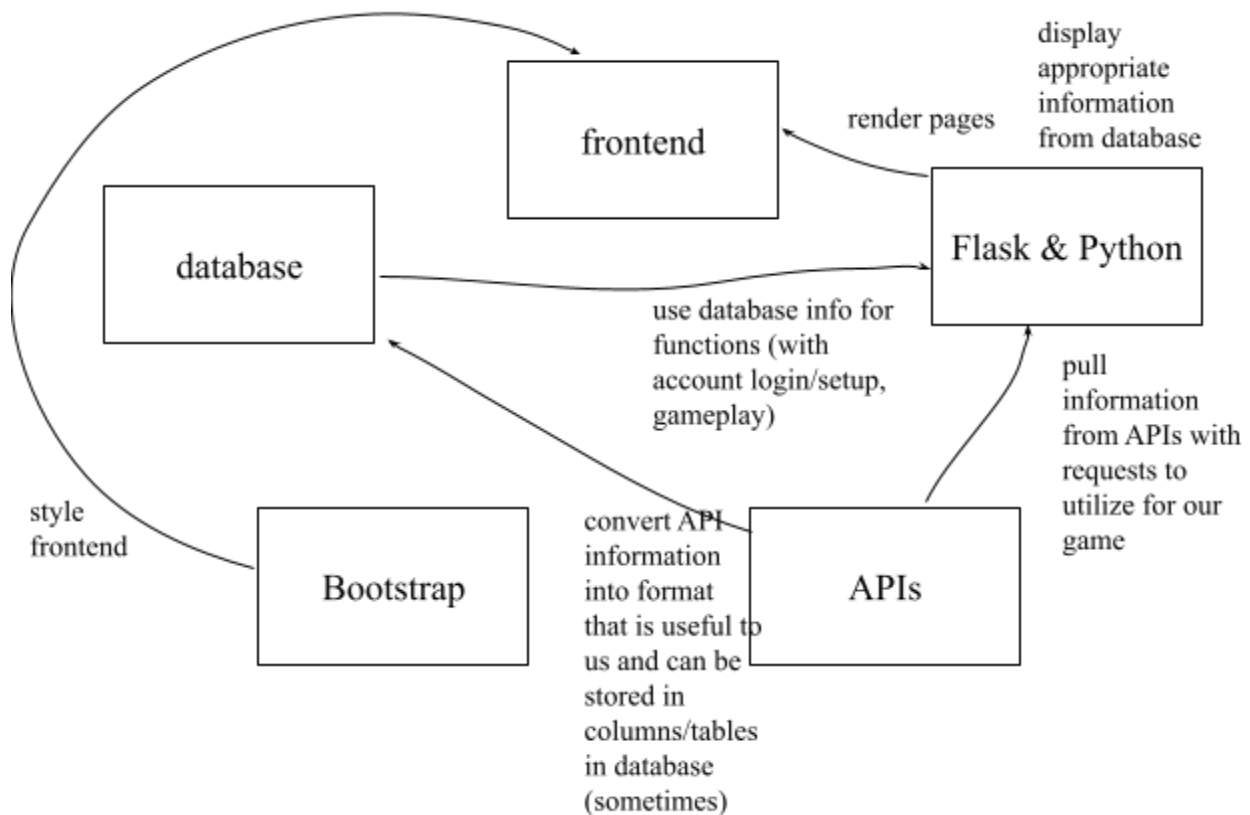
Breakdown of Tasks & Member Assignments

- ☐ Backend database creation, population, and management (Weichen, Jeffrey)
- ☐ Python associated with databases (Weichen, Jeffrey)
- ☐ API usage (Andrew)
- ☐ Connecting API to frontend and backend (Andrew)
- ☐ Python Flask work (Andrew, Jeffery, Weichen)
- ☐ Account signup and login (Ayman, Raven)
- ☐ HTML (Ayman, Raven)
- ☐ Bootstrap framework for styling (Ayman, Raven)
- ☐ Project management (AP)
- ☐ Devlog updating (All)

Components:

1. Database

- a. Store user account information
 - b. Store game information (influencers and amount of followers, artists and amount of listeners, etc.)
2. Frontend
 - a. Display the game to the user
3. Flask & Python
 - a. Render frontend pages
 - b. Game functionality
 - c. Account login/setup functionality
 - d. Database setup
 - e. API requests to populate our database
4. Bootstrap framework
 - a. Make our frontend pretty, intuitive, and user-friendly!
5. APIs
 - a. Provide info for our influencer and artist database tables



SQLite3 Database

We're currently thinking about not using tables overall for the object's we're comparing and use lists that we can get a randomized object from.

- Tables:

1. Users

Username (primary key)	text
Password	text
Points	integer
Profile Picture (potentially)	Link (text)

2. Celebrities/Influencers

Name	text
IDs for all of our supported platforms	text
[...] IDs (will need to know the id of their Twitter, YouTube, and Instagram for easier use of APIs)	

Frontend

- Bootstrap. It is very responsive.
 - Easy to use cards which we can customize
 - Upgraded form controls
 - Cool features such as dropdown menu and progress bar
 - Overall, very customizable and easy to use
- HTML
 - Use templates to connect with flask application

Templates

- Login.html
- Signup/register.html
- Home.html
- Selection.html
- HigherLower.html
- eaderboard.html

Flask Site Map

@app.route('/')

- Login page

@app.route('/auth')

@app.route('/signup')

@app.route('/signup/auth')

@app.route('/home')

- Show high score
- Menu

@app.route('/selection')

- Choose game mode here

@app.route('/higher_or_lower')

- Higher or lower gamemode in here

@app.route('/<gamemode>')

- Possibly other minigames here

@app.route('/leaderboard')

- Possible a leaderboard implementation

APIs

Will create a spam email for this

- Email: sqsquad53@gmail.com
- Password: “ ” (highlight to see)

~~1. Spotify API (how many listeners)~~

- a. Don't offer it through their API, only way to get it is through scraping

2. Instagram API (to get how many followers they have)

- a. <https://developers.facebook.com/docs/instagram-api/guides/business-discovery>
- b. Will get how many followers someone has

~~3. Google Trends (how many people are searching for that thing)~~

- a. Implementations are not restful

4. Twitter followers (and possibly retweets or something of that nature)

- a. <https://developer.twitter.com/en/docs/twitter-api/users/follows/introduction>
- b. Will just fetch the twitter followers of a certain profile (celebrity or other famous person)

5. Youtube subscribers

- a. <https://developers.google.com/youtube/v3/docs/channels>
- b. Will just fetch the number of subscribers that a person has

6. [...] Potentially more to come but these are our base 3

Roadmap:

- Construct SQLITE database along with various helper functions
- Build HTML templates along with flask web application
- Combine API and database into flask application
- Make functions for features
- Bootstrap and CSS to make app look pretty
- Testing and debugging