Team Name: armRESTs
Roster: Robin H. (PM), Aleksandra K., Johnny W.
Project Title: Groovy Movie

Proposal Summary:

This team wishes to create a website that provides the user with information regarding movies. Users would be able to see trending movies, movies sorted under certain categories, search for movies through the use of keywords or their mood at the time, leave comments, and upvote/downvote if so desired.

Roles:

Robin H: As project manager, Robin maintains the devlog, makes sure the team works well together, maintains the design documentation, stays on top of design changes, and handles minor coding issues.

Aleksandra K: Will write API card for Fandango and MovieDB. Will work on the searching component, generating movie pages (and all of the associated features outlined below),

Johnny W: Will write API card for IPStack and ipify. Will work on front-end development, templates, current user location feature.

Timeline:

11/20 Begin work on design doc
11/21 Build basic flask app and and set up devlog
11/22- 11/25 Flesh out basic pages based on sitemap, establish database, complete login/logout/create account functionality.
11/26 Setup homepage with currently trending movies (begin basic API work)
11/27 Write API cards for Fandango API and Movie DB and use them to implement basic search functionality
11/28 Continue work on Search (ie. Mood, Feeling Lucky) and configure templates for movies.
11/29 Begin work on displaying images, reviews, and other on movie pages
11/30 Write API cards for IPStack and ipify APIs and then begin work on nearby movies currently streaming in local theaters
12/1-12/3 Finish work on displaying movies based on current location/ Tune search if Necessary; Complete commenting and voting functionality
12/4 Establish User Profile
12/5 Add similar movies features on movie pages/ recommendations based on user ratings
12/6 Add documentation to code; robustify; fix bugs in need be
12/7 Demo!

API Links:

We will be using several API's to build this site.

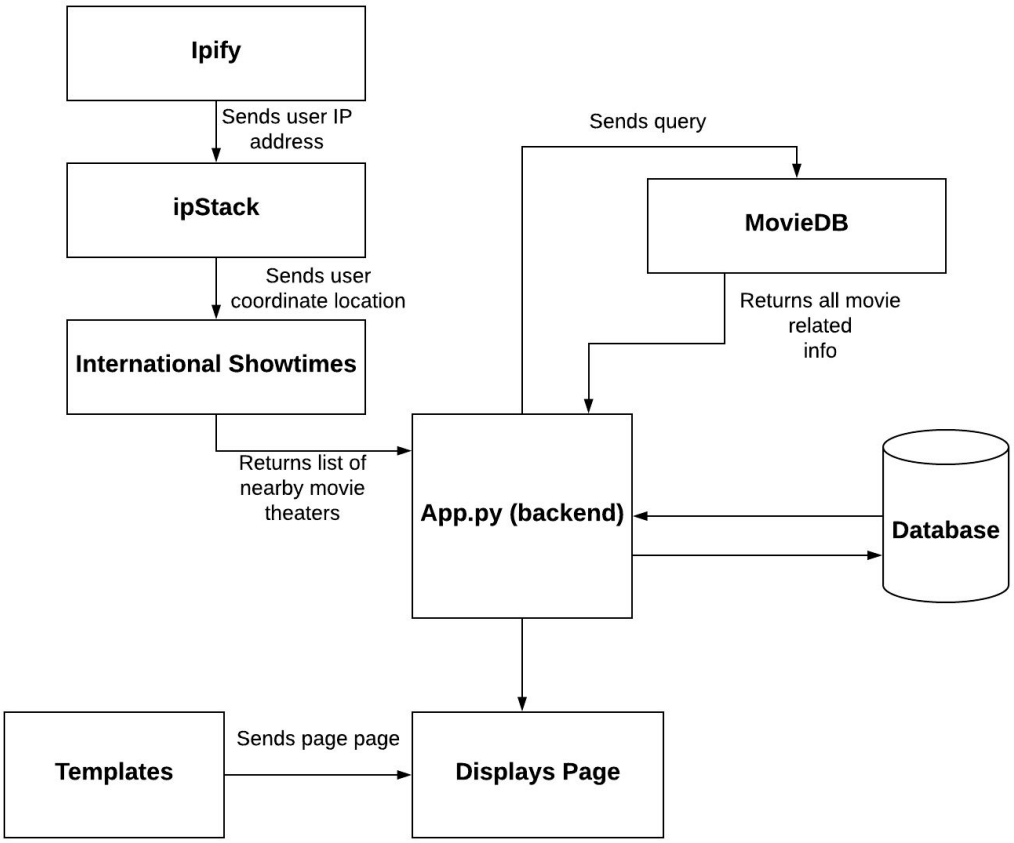
1) Movie DB (https://www.themoviedb.org/documentation/api)
   - We use this API to retrieve top rated and currently trending films.
   - Many features (similar movie recommendations, reviews, trailer, poster, title) visible on Movie pages are sourced from this API.

2) ipstack API (https://ipstack.com/)
   - Used to locate and identify website visitors given an IP address.
   - Necessary for Near Me section of the site since we display movies in the user's locality.

3) ipify API (https://www.ipify.org/)
   - Pulls user's current IP address.
   - Necessary to feed IP into ipstack API.

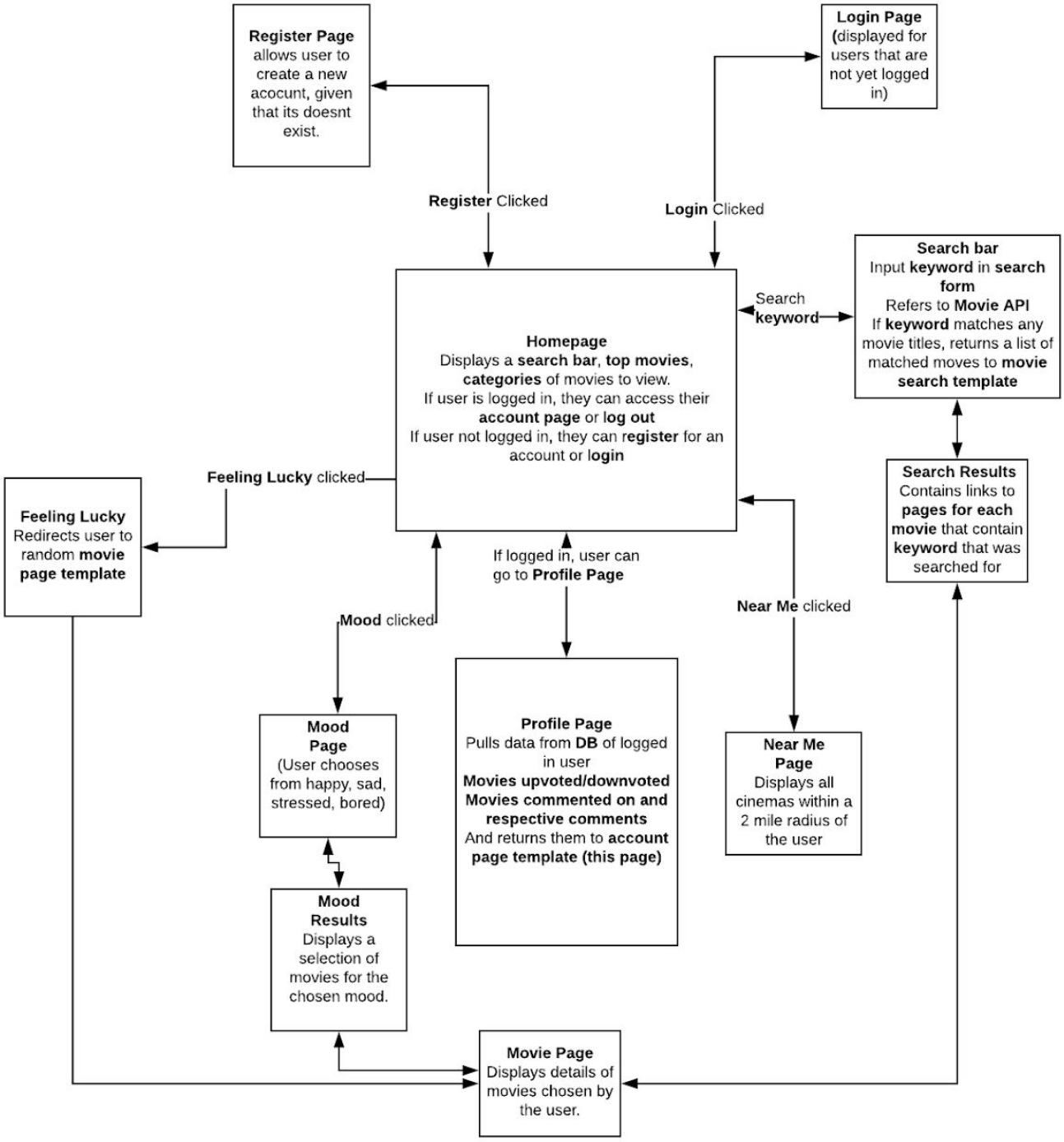
4) International Showtimes API (https://www.internationalshowtimes.com/)
   - User's location is fed into this API to retrieve a list of cinemas near the user.

Component Map:



Site Map

Team Name: armRESTs
Roster: Robin H. (PM), Aleksandra K., Johnny W.
Project Title: Groovy Movie



Note: The user can return to the home page from any page on our website.

Home:

a) If the user is not logged in, they can browse movies based on categories, use the search function (outlined later), login, and register.

b) Else, if the user is logged in, they can do all of the things outlined above with the addition of checking user-specific data, upvoting/downvoting movies, and leaving comments on movie entries. They can also have the option to log out.

c) The search bar is positioned on the upper left of the home page. The user can type in a movie name to search directly. They can also select the "Mood" option which caters to the user's indicated mood (which is a separate input). The last option is an "I'm Feeling Lucky" button that leads to a random chosen movie.

d) In general, the home page displays currently trending movies in the user's area. There is also a sidebar full of categories the user can browse through.

Login:

The Login page check's the user's input credentials against the information stored in the database. Successful login leads back to the home page.

Register:

The Register page allows the user to create a new account. User input is referenced against information stored in db to make sure that the username does not already exist.

Mood:

The "Mood" option below the search bar leads to a separate page that allows the user to select from four moods: happy, sad, bored, and stressed.

Search Results:

A successful search results in a page of recommended movies to watch. If the user searches for a specific movie, or a section of the name, movies with parts of the name show up. If the user searches by mood, corresponding movies show up (eg. top ten comedy for a "Sad" query). The last option, "I'm Feeling Lucky", just returns some random currently trending movies.

Movie:

Clicking on individual movies within Search Results directs the user to a page about that movie. Displayed on each movie's page is the title, a trailer, a short summary, and some reviews. All of this is retrieved from Movie DB. Apart from displaying information about the movie, each movie page allows logged in users to upvote/downvote and to leave comments.

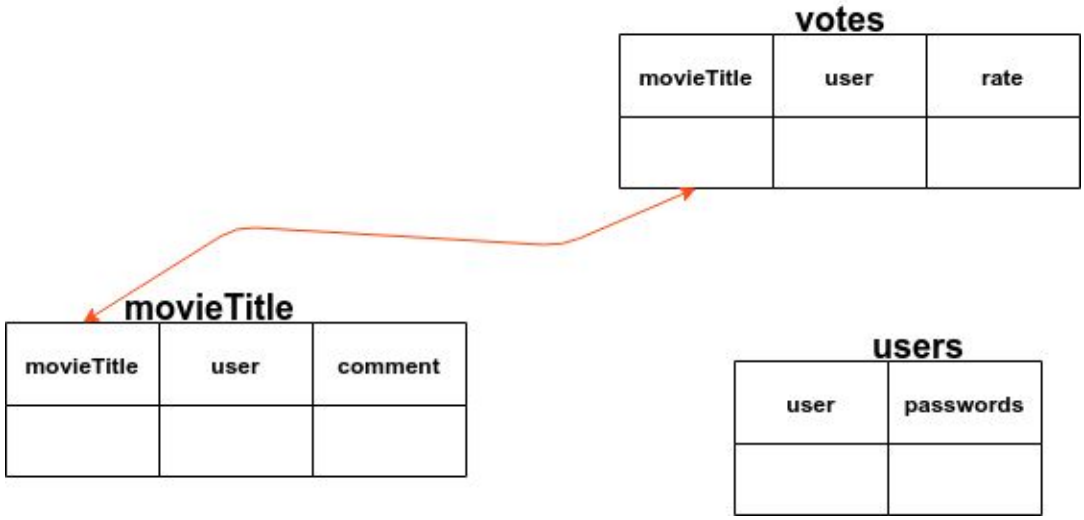
Profile:

A logged in user can reference their profile from the Home page. The User Profile page displays user information as outlined before.

Near Me:

Another page accessible from the home page. Using the user's IP and associated location we show movie theaters near the user.

Database Schema:



movieTitle:

- A a new table is created every time a user accesses a Movie page for the first time. The table is given the exact name of the movie (ie. movieTitle).
- This series of tables is used to access and post comments that a logged in user can leave on a Movie page.
- Whenever a comment is added, the table for that specific movie is updated.
- All the fields for this table are textual.

votes:

- The votes table is updated every time a user upvotes or downvotes a movie.
- The user's rate is stored as an integer (1 for upvote and -1 for downvote).
- Entries are pulled from this table are used to make sure that the user does not vote more than once, to calculate voting stats on a specific film's page, and to allow for voting data to appear on the user's Profile page.

users:

- The votes table is used for authentication purposes.
- Stores username and password upon registration.