

Program Description

Our program is a messaging application where users will be able to chat with one another along with one another, featuring real-time messaging, group chats, friends, and more. Users will be able to upload their own profile pictures and emojis that they can use in their desired chats. They will be able to connect with their friends by sending friend requests to them along with being suggested new friends based on mutual friends.

Program Components

- `__init__.py`
 - Routes
 - `@app.route('/login')`
 - `@app.route('/registration')`
 - `@app.route('/')`
 - `@app.route('/profile')`
 - `@app.route('/friends')`
 - `@app.route('/settings')`
 - Event Handlers
 - `@socketio.on('connect')`
 - Deals with clients connecting to the server. Will print in console to confirm connection.
 - `@socketio.on('disconnect')`
 - Deals with clients disconnecting to the server. Will print in the console to confirm disconnection.
 - `@socketio.on('new_message')`
 - Handles messages clients send to server, and server responds by sending that message to the intended client.
 - `@socketio.on('join_room')`
 - Handles clients joining a group, adding a new row in the UserAssociation table and allowing the client to access and receive messages from the group along with sending messages to the group. (This also handles groups that are newly formed.)
 - `@socketio.on('upload_image')`
 - Handles client sending a message and stores it in the Cloudinary API then sends the image to the recipient of the message

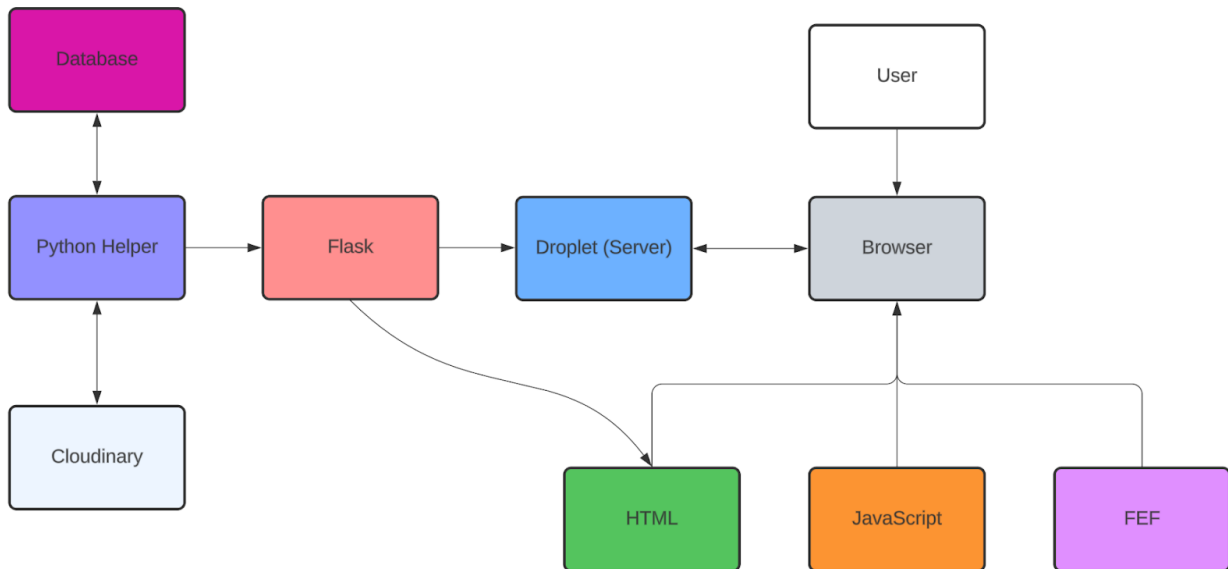
- @socketio.on('upload_emoji')
 - Handles client uploading an emoji to the server where the server stores the emoji in the Cloudinary API which the client can use in future messages
 - @socketio.on('change_profile_picture')
 - Deletes pre existing profile pictures stored in the Cloudinary API and changes to the new one that is uploaded. Server will broadcast the change to all users connected to the server.
- static/
 - css/
 - login.css
 - registration.css
 - home.css
 - profile.css
 - friends.css
 - settings.css
 - js/
 - home.js
 - profile.js
 - friends.js
 - settings.js
- templates/
 - login.html - page with form where users may login into their accounts
 - registration.html - page with form where users may sign up for an account
 - home.html - home page with a friends bar on the left and the chat on the right (similar to discord)
 - profile.html - profile page of a user (displays profile picture, mutual friends, and about me/bio)
 - friends.html - displays list of recommended friends based on mutual friends and be able to search for friends
 - settings.html - settings page where you can change profile pic and add about me
- database.py
 - Setups database tables
 - Contains functions to help access tables
- keys/
 - Key for Cloudinary API

API

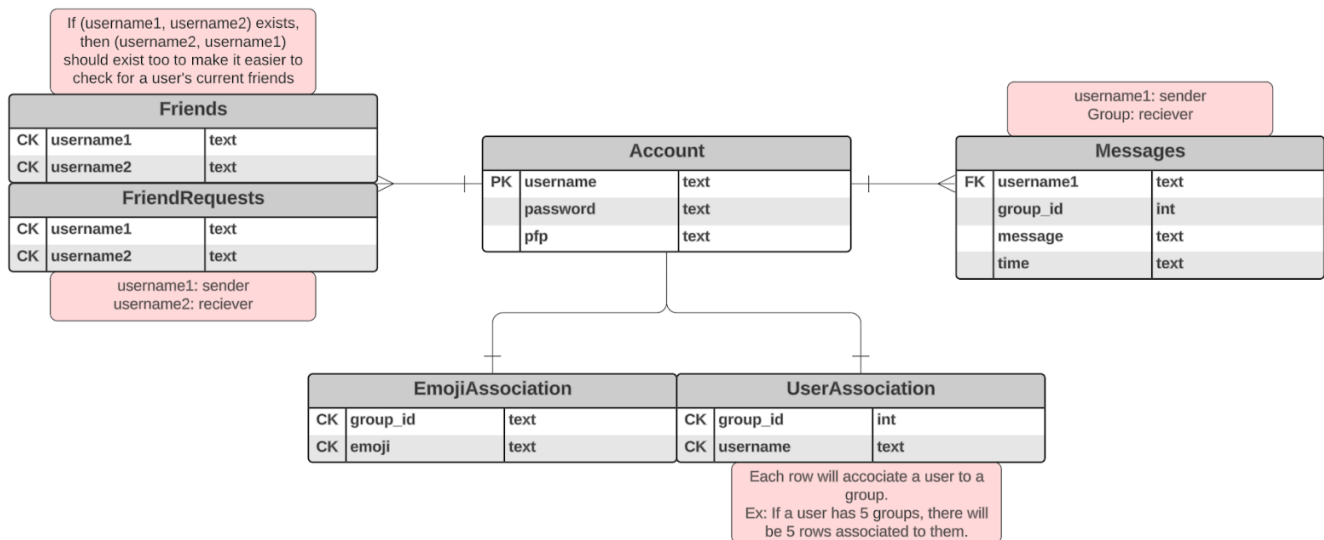
Cloudinary will be used to store images for user uploaded profile pictures and emojis that they would like to use. Images will be transformed depending on what their purpose is, (pfp, emoji, message images) and then stored with the Cloudinary SDK. We can then use the image's public img url in order to display the images.

https://github.com/stuy-softdev/notes-and-code/blob/main/api_kb/411_on_Cloudinary.md

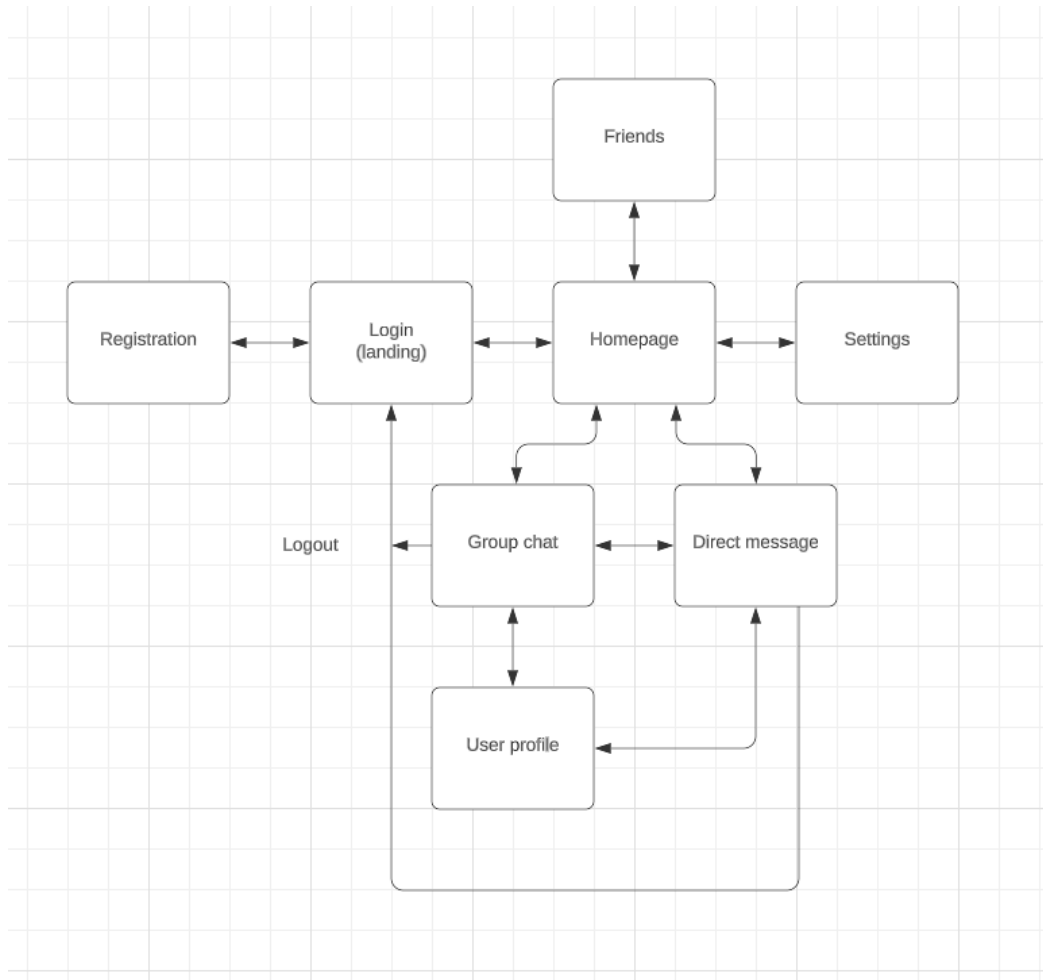
Program Component Interactions/Component Map



Database Organization



Site Map

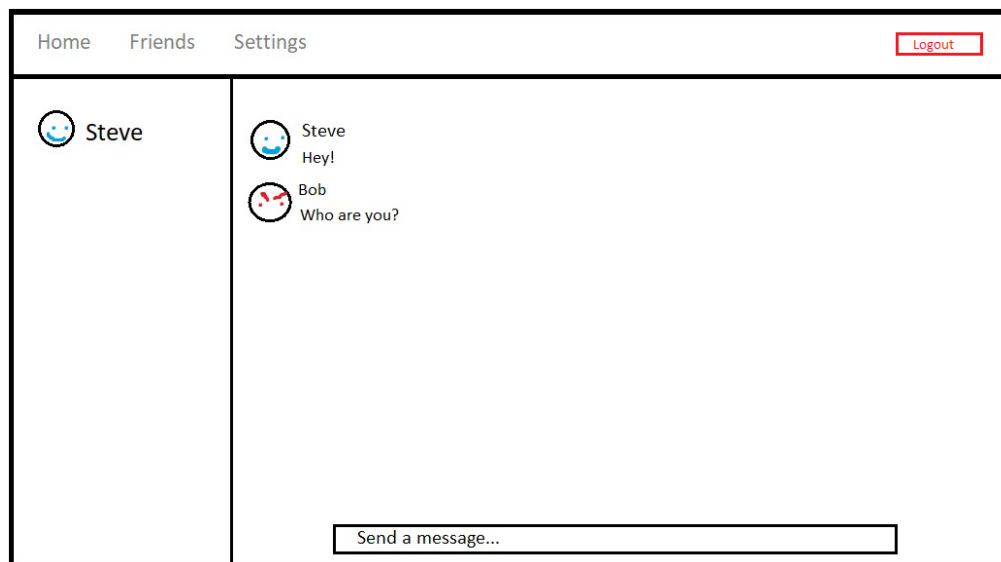


*User profile is the profile of a single user (viewing another user's profile)

Front-end Framework

Bootstrap will be used as our Front-end framework

Home page



Roles

Ivan Yeung - PM

- ❖ Flask
- ❖ Websockets

Jun Hong Wang - DEVO

- ❖ AJAX
- ❖ JS

James Yu - DEVO

- ❖ FEF (Bootstrap)
- ❖ Droplet

Joshua Liu - DEVO

- ❖ Database
- ❖ Cloudinary
- ❖ Websockets

Stretch Goal

- Game Pigeon games where each turns is a message to the other client