



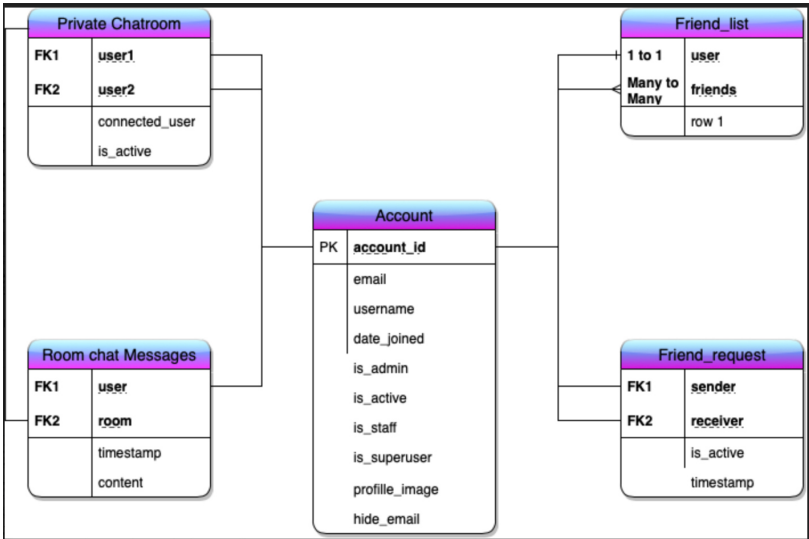
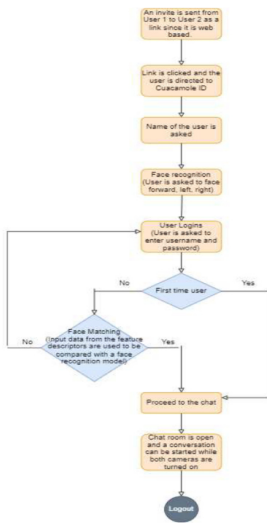
Problem

Let's say someone comes to work, clocks in, and gets paid by the hour. Maybe he/she slacks off and gets paid for not working. Gone are the days where the employee abuses the system. With this amazing app, Guacamole ID is designed to keep track of employee labor, and time accurately. Guacamole ID provides authentication using not only User ID and password, but also Face ID.

Requirements

- Python
- Django
- Visual Studio
- Jupyter Notebook
- pgadmin 4
- Guacamole ID
- Mac OS
- Windows OS

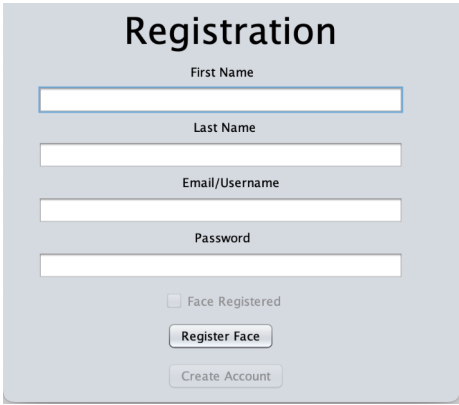
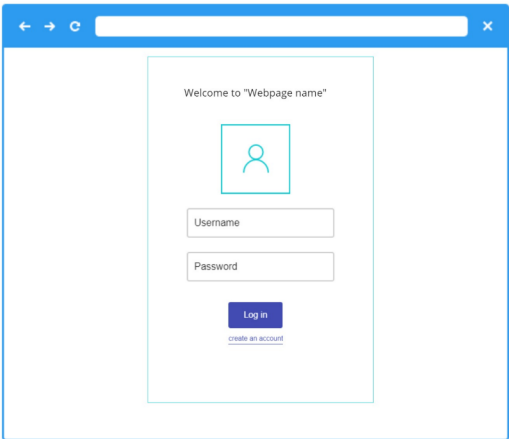
System Design



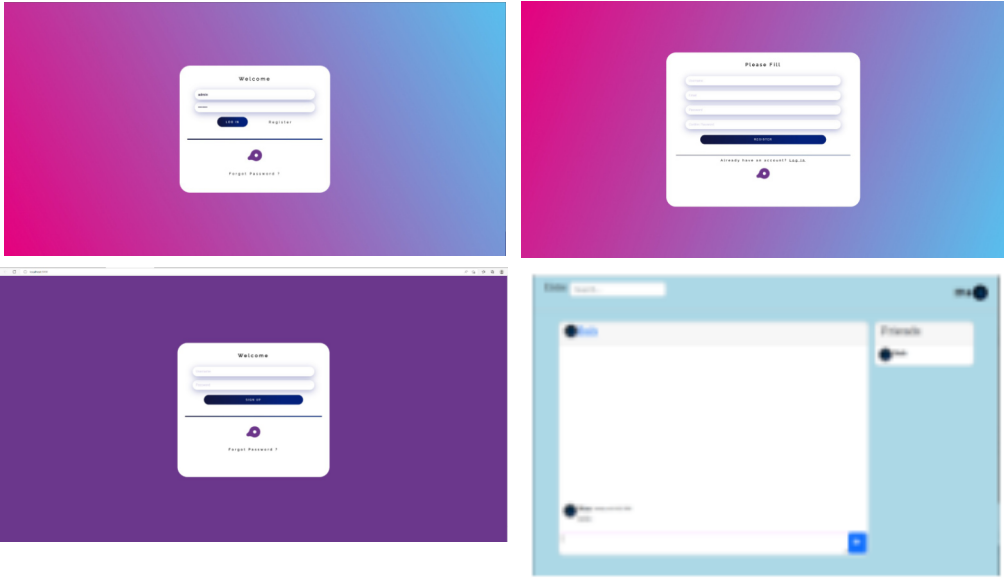
Current System

The current system contains the chatroom and login feature. The website was built from scratch and it initiated this semester. Previously, the application was just an offline based Python GUI application. Our vision was to expand to something more flexible and accessible, thus the web application was born.

Object Design



Screenshots



Implementation

- For the front-end we used React JS
- For the backend we used Visual Studio Code, Django, PostgreSQL, Pgadmin4, Redis, and web-sockets.
- For the AI portion of the application, we used Python, and its libraries including OpenVino, Matplotlib, and Numpy
- We used GitHub to share and edit our work.

Verification

This project was carefully and thoroughly tested and analyzed by our team members, who built front-end, backend, and AI portion of the web application. We had multiple test cases

Req #1
Priority 1
Purpose Login/Register Menu
Input The user can register their username and password and use it as the first measure of security before facial recognition to access the chatroom.
Operations The initial measure to ensure safety of the user's account. This is the first index of security that will be viewed by the user. The user will need to input a username and password. If the user inputs the wrong username/password they will be asked to retry until the correct input is placed.
Output Will then scan the user's face to ensure the person who registered the username and password is the same as the one who is currently logging in. Approximate loading time for inputting the correct username/password should be about 1-5 seconds.

Req #1
Priority 1
Purpose Provide 1 to 1 chat capabilities
Input After going through the login, the user should be able to view the chatroom with the other user.
Operations Provide interactivity between two users; response time should be within a couple seconds to ensure fluidity. Flaky internet connections will most likely cause the response time to be delayed due to poor connection.
Output Will showcase a message on the users chatroom that can be seen by the sender and the receiver. The receiver will also be able to send a message back. Assuming consistent and stable internet connection between both sides, the response time should be within 1-2 seconds.

Req #1
Priority 2(1)
Purpose Logout functionality
Input After logging in, there must be an option to allow the user to sign out to ensure the website does not auto save the credentials after the user closes from the site.
Operations Once logging in the user will need to have a method to be able to sign out to ensure other users are not able to access the website while the account already signed in. The user should not be signed in after they have closed out of the site to ensure no other user can access the site after they have left. The method to logout should be a button to close out of the messaging system, either through the button or closing out of the tab.
Output Will log out the user; they will need to login using the same steps needed for initial login to view the chatroom.

Summary

During Summer 2022, our team were committed in creating this amazing web application from the beginning. Our team worked hard to build the front-end, backend, and AI. We successfully implemented a deliverable product by the end of the semester.

Acknowledgements