Design Document

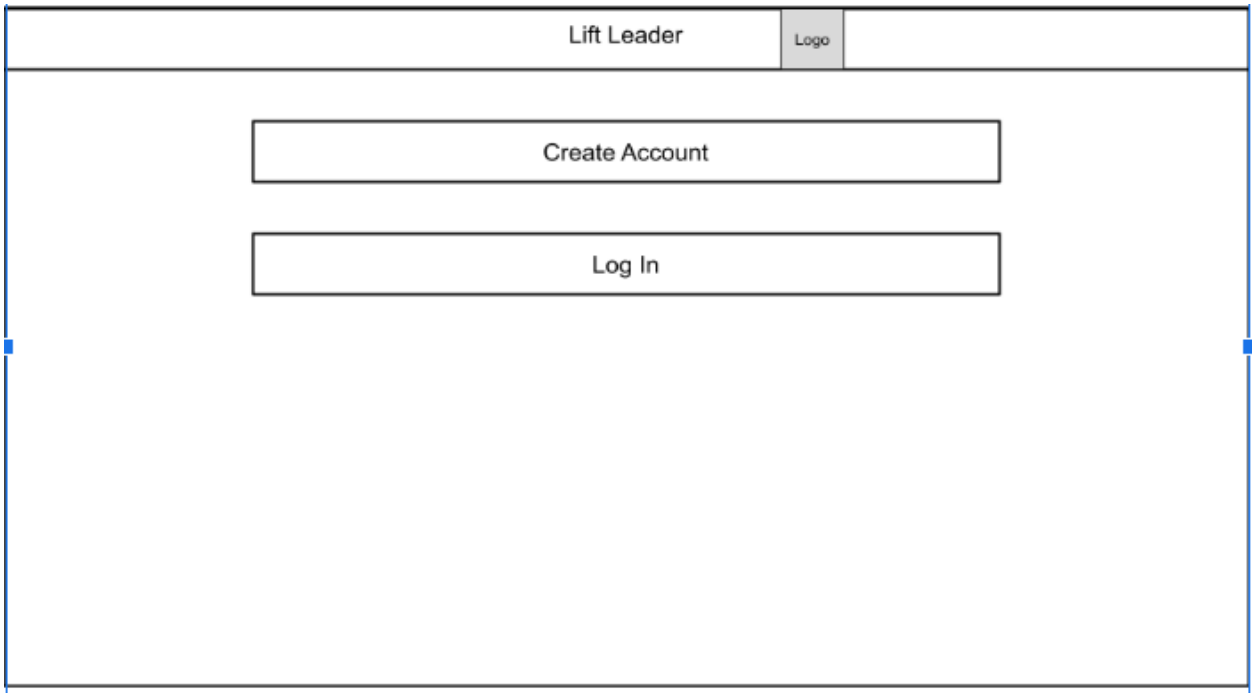
• High level description of the design that includes the system architecture
- Lift Leader is a social media style platform that allows users to record workouts and post to others. There is a prominent focus on a leaderboard system that tracks a majority of common weight lifts and machine movements in a gym.
- The social media platform runs through a web app where users can register, login, and begin browsing and posting.
- Upon initially opening the app, users will be promoted to either register a new account or login to an existing account.
- After login is complete, the user will be placed on their “user” page where they can enter all of their personal information
- A user can then navigate to any of the other display options including “feed”, “gyms”, and “leaderboards”

• Problem solving approaches that were considered during the design phase. You are required to document the two most promising ones that were considered and rationale for selecting the winning approach.

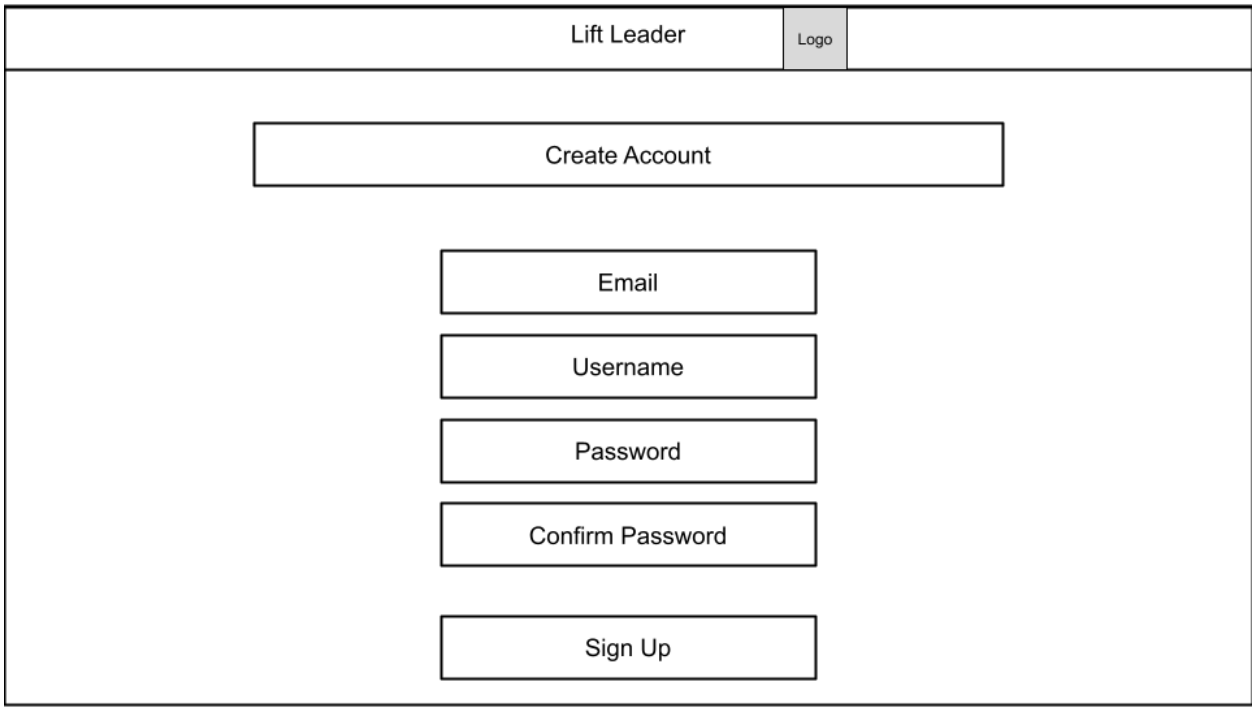
During the design phase we knew we wanted a RESTful API but we had to decide what the best technology to use would be for our team. One promising solution is FastAPI since at least one member of the team is familiar with it. It's also one of the fastest web frameworks and uses python. A second promising solution would be NodeJS which is a more popular framework that uses javascript. At least one member of the team was familiar with this solution too so it made choosing the winning approach difficult.

• Mockups of the different screens in the app.

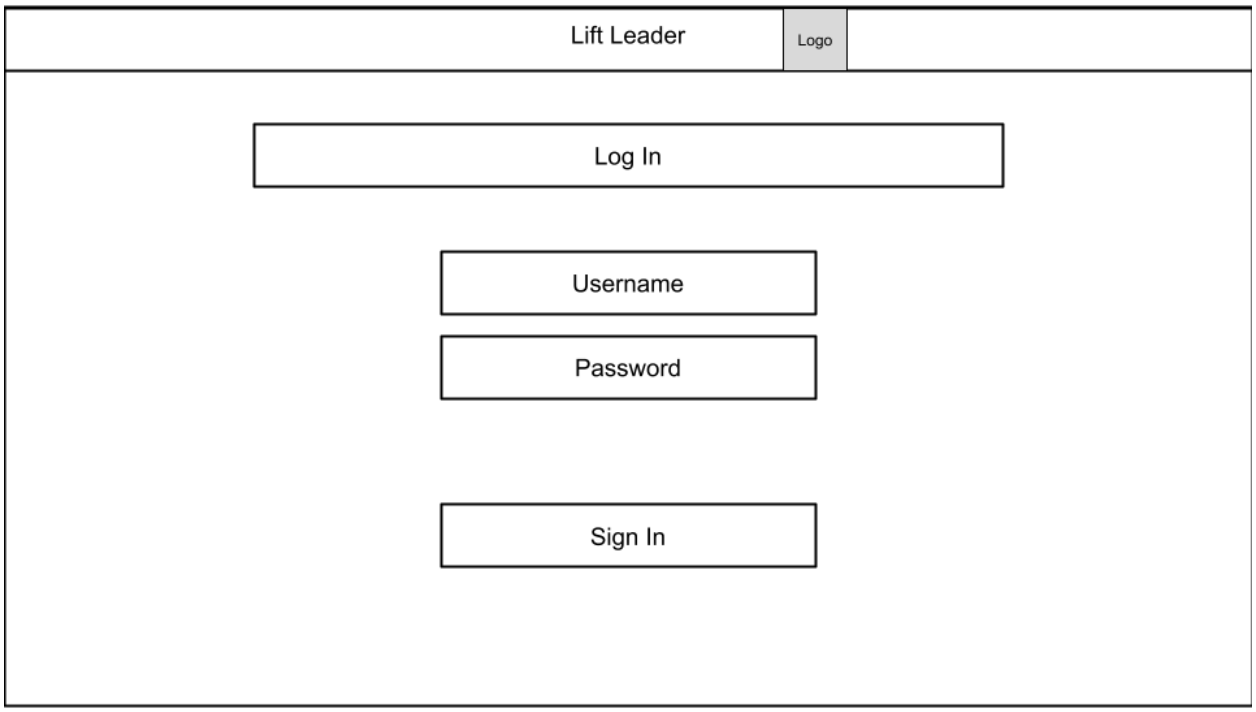
Login/Sign Up page



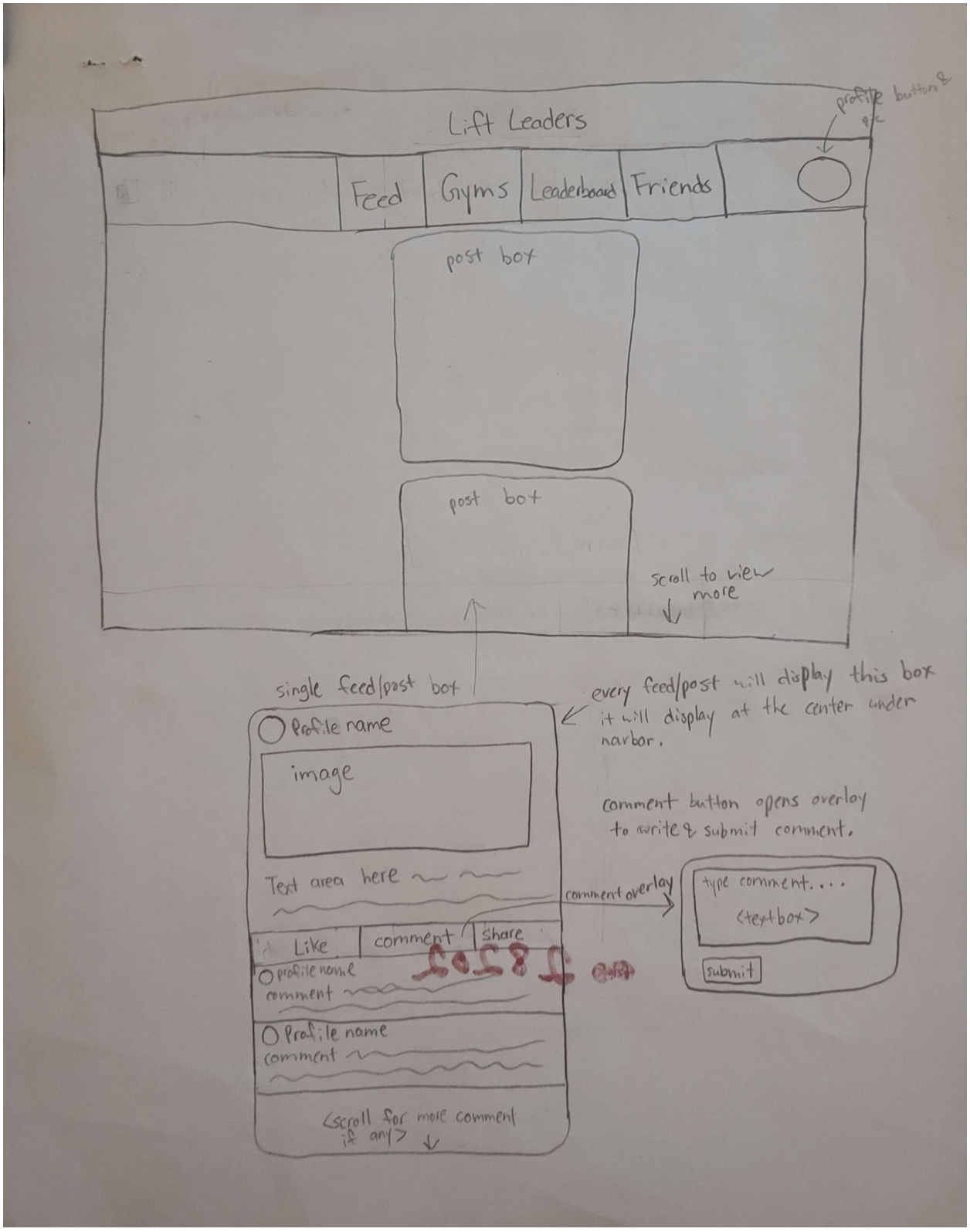
Sign Up/Create Account Page

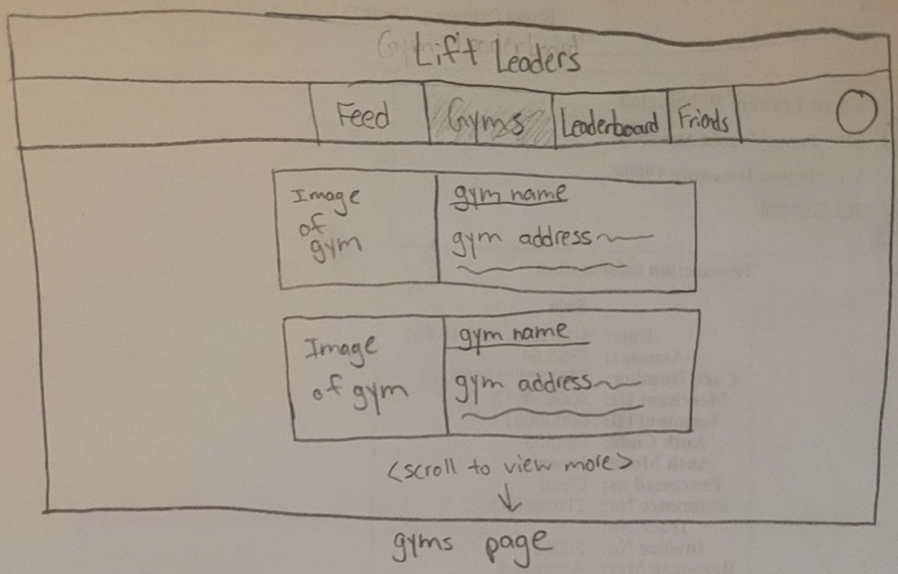


Login Page

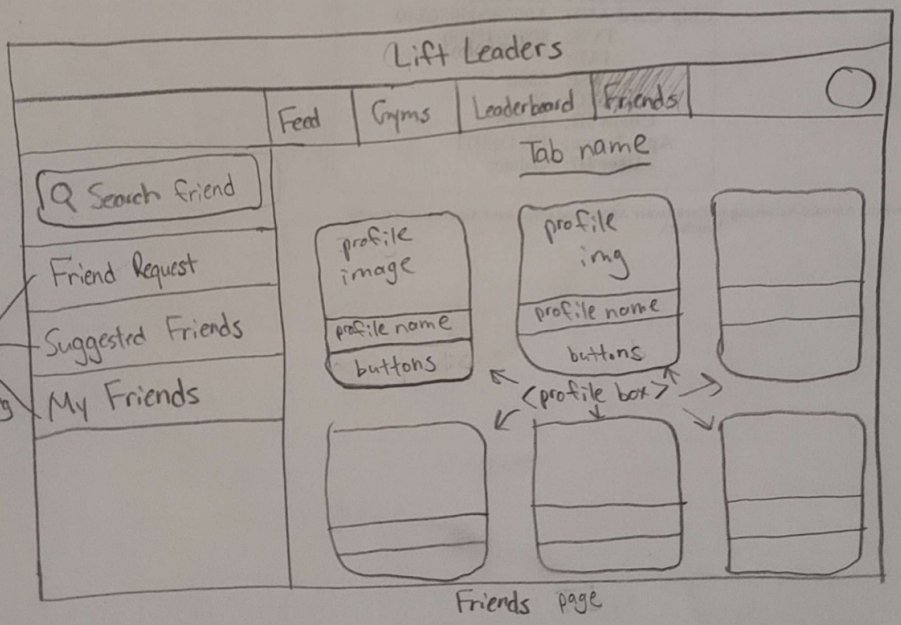


Home/Feed Page



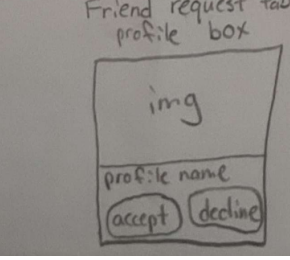


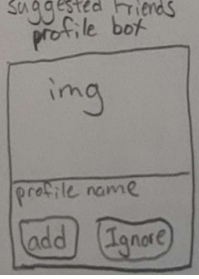
gyms page

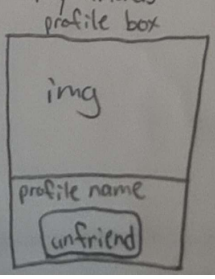


tab filters and displays profile depending on which you press

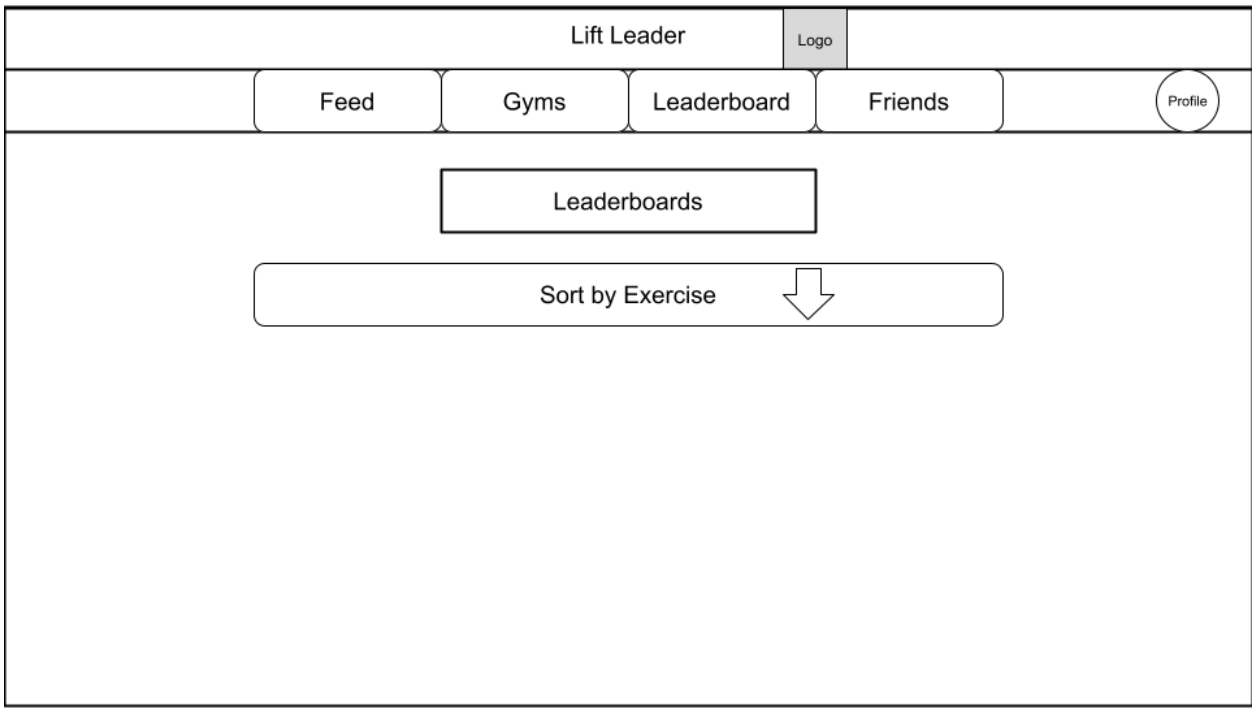
Friends page



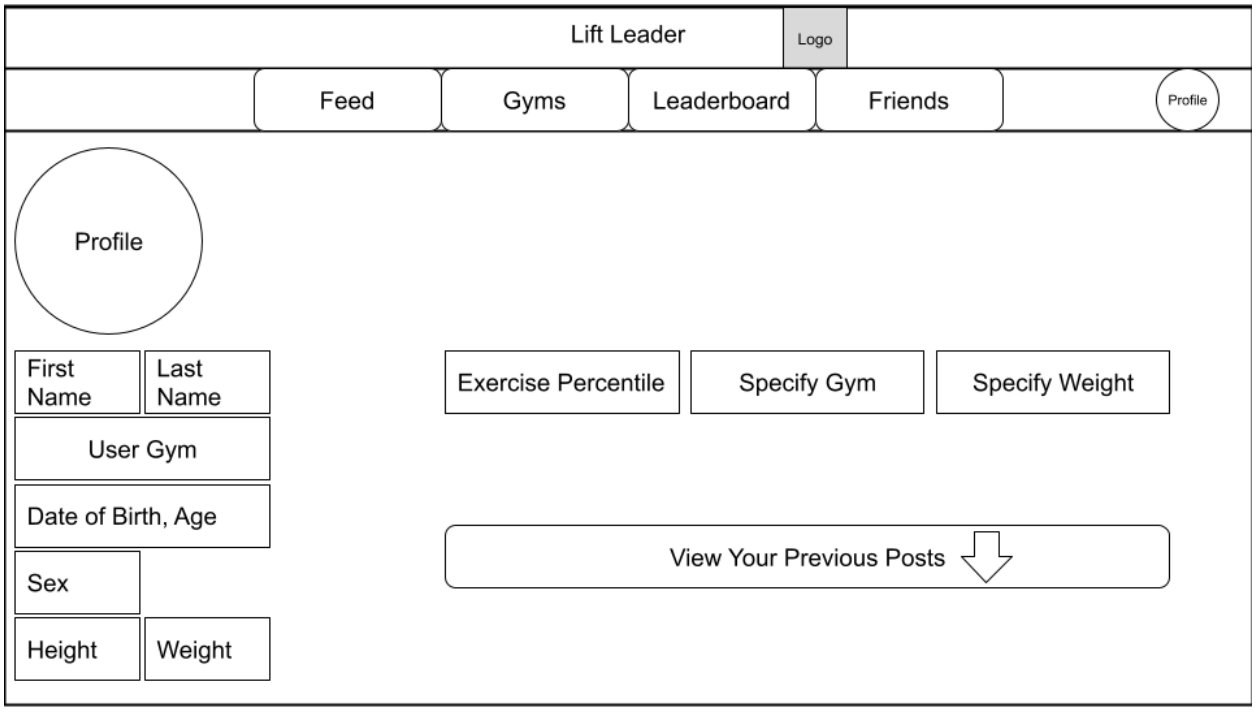




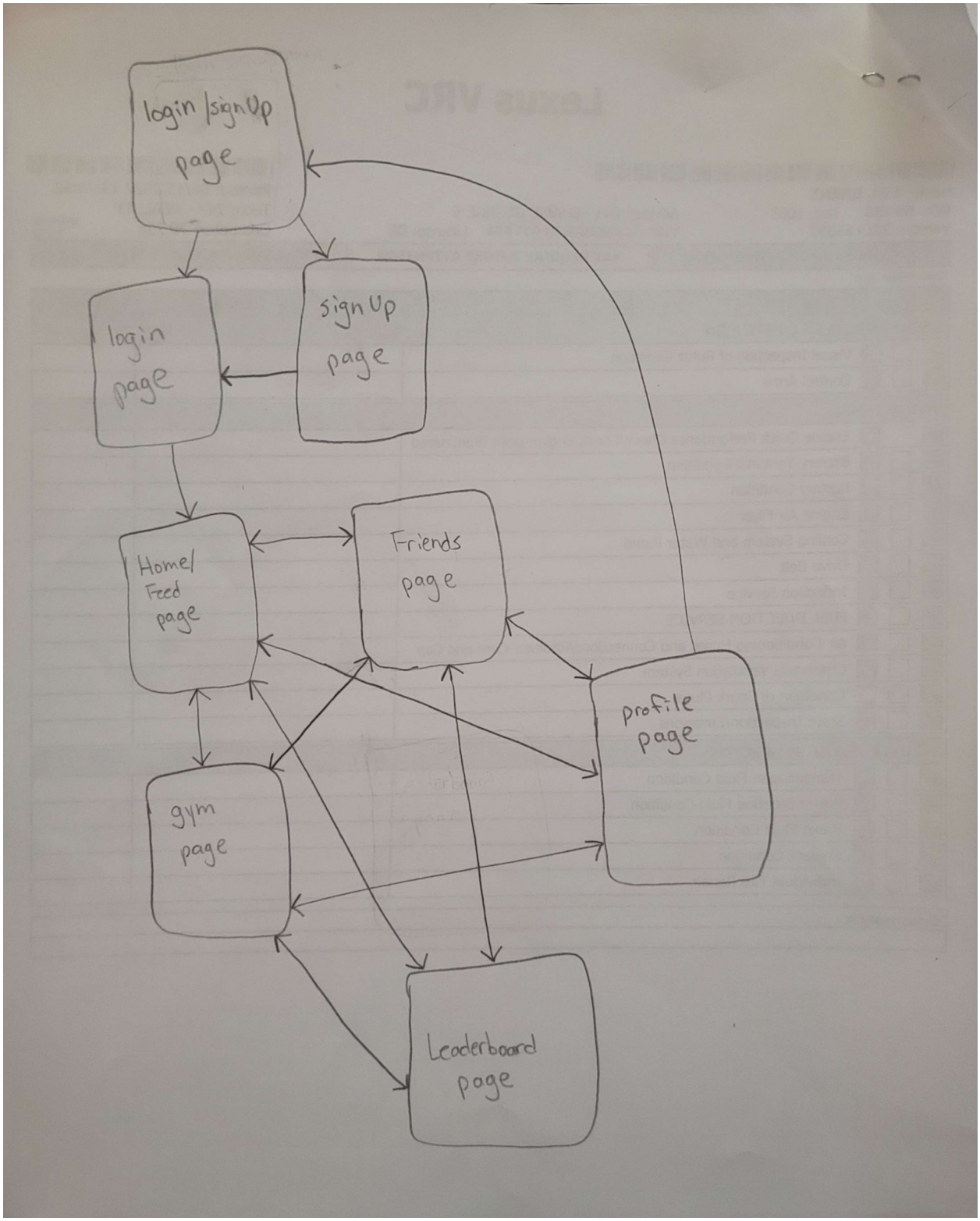
Leaderboard Page



Profile page



• Navigation between app screens



• Backend information

➔ Database schema

➔ RESTful API

Endpoint	Request Method	Description
`/login`	`POST`	Logs a user in
`/users/{user_id}`	`GET`	Get user info by id
`/users/createUser`	`POST`	Create a new user
`/users/updateUser/{user_id}`	`PUT`	Update user info
`/users/deleteUser/{user_id}`	`DELETE`	Delete a specified user
`/users/posts/{user_id}`	`GET`	Get users posts
`/users/following/{user_id}`	`GET`	Gets a user's following list
`/users/followers/{user_id}`	`GET`	Gets a user's followers

		list
/users/unfollow/{user_id}	DELETE	Unfollows a user
/gyms/{gym_id}	GET	Get gym info by id
/gyms/createGym	POST	Create a new gym entry
/gyms/updateGym/{gym_id}	PUT	Update a gym entry
/gyms/deleteGym/{gym_id}	DELETE	Delete a gym entry
/lifts/{user_id}	GET	Get all lifts for a user
/lifts/createLift	POST	Create a new entry for a user's lift
/lifts/deleteLift/{lift_id}	DELETE	Delete a specified lift
/posts/{user_id}	GET	Get (autogenerated) posts for a user

• Functionality provided by each app screen. Explain which RESTful endpoints are accessed by the screen to provide the specified functionality.

• User authentication and data security issues

• Tech stack
  ➔ Platform: Web
  ➔ Tech stack: React.js, Material UI, Python/FastAPI, SQLAlchemy

- team members and list of goals to be accomplished

- Charles Evans
    - evansc24@students.rowan.edu
    - Focus on front end development, assist in backend development
    - Develop homepage
    - Help with REST API

- Emerson Henkel
    - henkel57@students.rowan.edu
    - Develop “User” page
    - Create a standard style scheme to be applied throughout the web app
    - Focus on front end development, assist in backend development

- Brent Liu
    - liubre82@students.rowan.edu
    - Focus on backend development, assist in front end development.
    - Help with REST API
    - Implementation with API
- Sudeepta Sarkar
    - sarkar27@students.rowan.edu
    - Develop and display “Sign Up/Sign In” page, “Sign Up” page, “Sign In” page using React Components

- Nicholas Cullmann
    - cullma57@students.rowan.edu
    - Focus on backend development
    - Create REST API
    - Create database schema and implementation with API