CS411 Team Assignment 1 - Proposal
Group 20 - Manish Patel, Ji Zhang, Chen Feng, Curtis Mason and Takamitsu Shirono

One idea could be creating a scheduling application to work out conflicts with groups (starting with just 2 parties and then possibly expanding later), determining common meeting times, and send reminders as necessary about these meetings. The database used could simply comprise each user's availability/schedule and the application would use each user's data to figure out the meeting times. A few APIs that could be useful here would be the Google Calendar API/Microsoft Teams API and a weather/Holiday API to incorporate into our scheduling decisions. Third-party authentication, through Amazon Cognito for example, would work here since each user would need to log in to move forward with scheduling with others. Although not final, React or React Native would make up the front-end and Python on the backend (i.e. Flask and Javascript).

Our second idea is a gym equipment "guide" app for which users can figure out how to use certain machines or types of equipment at the gym. We can categorize all types of equipment and let the user sort by the specific brands of the machines or even the specific body parts that the machine would be exercising. Some sort(s) of fitness API(s) could be useful in adding functionality for tracking different activities at the gym or nutrition records (i.e. macros), so that the users can have a one-stop application for many of their gym needs. A database of each user's personal information (e.g. height, weight, etc.) along with the actual database of the different equipment and machines would be necessary. Alternatively, if the application appears too complex, we could simplify the application to provide basic goals for each users (e.g. Bulking, Toning, and Burning Calories). For each goal, the application will provide appropriate workout routines, pulled from some Fitness API, as well as nutritional intake goals, pulled from a database/API on nutritional value information.

A third idea, although it may not fulfill the requirements, we could create a chat service that we would develop such that developers can use our developer friendly API to send messages and maintain conversations with their users over our network. This would be more of a service, where users could add chat to a platform. A SQL database could be useful for storing the messages/conversations that each user partakes in.