PRDv2

Team Name: Pretty Lil Leetcoders

Project Title: LogMyMotion

Team members: Sung (Sam) Kim, Nikhil Patil, Etgar (Eric) Freilafert, Yuyang (Ethan) Su, Aryaman (Arvan) Das

Introduction

Background

Working out and fitness are essential to both physical and mental health. However, not everyone is familiar with the proper form that should be applied when performing exercises. Lifting weights with poor form can be extremely dangerous, as people suffer injuries that can seriously hinder efforts to maintain fitness and mobility, depending on the severity of the injury. In order to learn proper form, many first-time lifters schedule fitness sessions with physical trainers. Although trainer sessions provide a safe method of learning exercises, many have trouble arranging consistent sessions. It is often difficult to coordinate sessions with 2 people, as both of their schedules need to align properly. In addition, trainer sessions are prohibitively expensive, which provides another barrier. In light of these issues, we have decided to design and implement a system that processes real-time video footage of a subject performing certain exercises in order to provide feedback and guidance on the user's exercise form.

One of the most common weightlifting exercises is the squat. Beyond workout exercises, people constantly perform squat-like movements for tasks as simple as getting up from a chair and getting in and out of a vehicle. Squats are popular because they target multiple muscle groups across the body, including core and leg muscles. Abdominal and oblique muscles are exerted to support a steady, straight back during the exercise. In the legs, large muscles including the glutes, hamstrings, and quadriceps are exerted to support body weight plus any additional weight added by the subject. By having a widespread impact on muscles across the body, the squat is a popular exercise used not only by professional athletes and trainers, but also ordinary people who are looking to improve their physique, strength, and performance.

Problem

Progress is often difficult for beginners who have limited knowledge about safe and efficient workout techniques. As a result, many people end up getting injured or becoming stagnant in their strength growth due to poor form. In order to progressively become stronger, one must ensure that proper form is established early on in the training process. The solution is to provide

an application that can give feedback on form to ensure that workout exercises are done in a safe and effective manner.

Innovation & Core Technologies

poseNet

poseNet is a machine learning model that achieves real-time human pose estimation. This model has the ability to identify either single or multiple human figure poses in photos and video footage. We will innovate by using the model to identify key points across the body to keep track of key cues that will indicate correct form for our exercises. This requires us to develop a system that can analyze the performance and body position of trainees during exercise repetitions.

ml5.js

ml5.js is a collection of libraries that provides a clean, organized interface to solve different problems in machine learning. The collection includes various machine learning models and algorithms that are compatible with a web browser platform, including poseNet and other web models encapsulated in Tensorflow.js. We will use this tool to prepare and configure poseNet to suit our system's processing logic.

p5.js

p5.js a JavaScript library that includes various tools to perform animation and drawing in the browser. We will use this library to identify and visually display the key points and and skeletal frame that poseNet

Proper Form Research

This will be accomplished by examining the squat form of professional trainers and weightlifters using joint detection and creating measurable guidelines that can be utilized to analyze the form of our trainees. By applying these guidelines to a trainee's program, we can ensure that a trainee would never strain their body and injure themselves.

Audio Recognition

In addition to joint detection, we will leverage audio recognition to track breathing patterns to ensure every component of the exercise is done a proper way. Irregular breathing can hinder good form and cause greater stress on muscles during exertion. In our app, we utilize joint detection and audio recognition to craft a holistic evaluation about a trainee's execution of weight-lifting exercises. We understand the importance and safety of performing these exercises correctly.

Assumptions

Our product assumes that the user is actually attempting to perform the specified exercise and not trying to purposefully confuse the analyzer. Additionally, we assume that the user is in a

well-lit environment and is wearing clothing that contrasts with the user's background. For the purpose of audio recognition, we will assume that users will be wearing mic-in headphones.

Team Goals/Objectives

Our goal is to develop a web based platform where users can keep track of their training progression and receive feedback that maximizes the efficiency of their workouts while addressing issues that lead to injury. Our application should help guide users to be able to live fit lives and prevent users from being discouraged by the tricky learning curve of lifting weights.

Stretch Goals

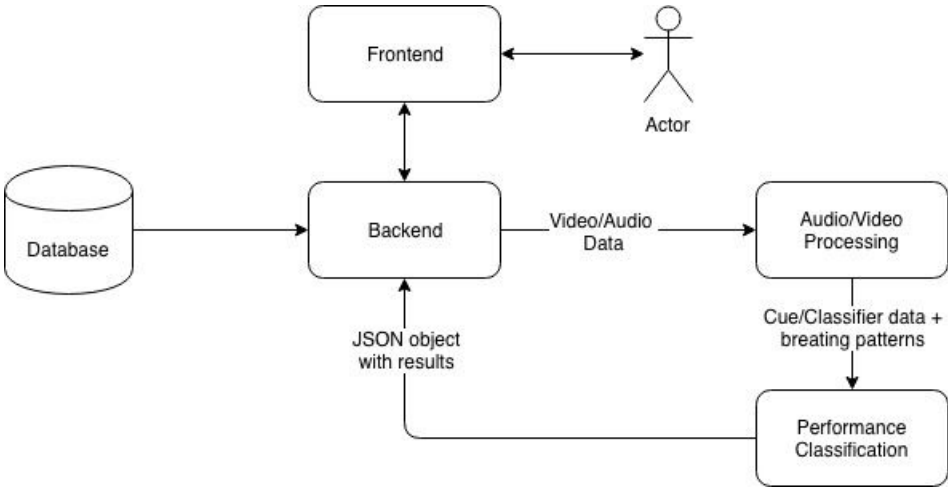
- Analyze a variety of exercises (currently focused on the squat)
- Allow user to specify which exercise they're going to perform by using their voice
- User profiles to keep logs of previous exercises and track progress over time
- Face detection as authentication
- Integration of exercise results with social media

Soft goals

- Web application should be running 24/7 and be powerful enough to not lag
- AI speech should be realistic and smooth

System Architecture Overview

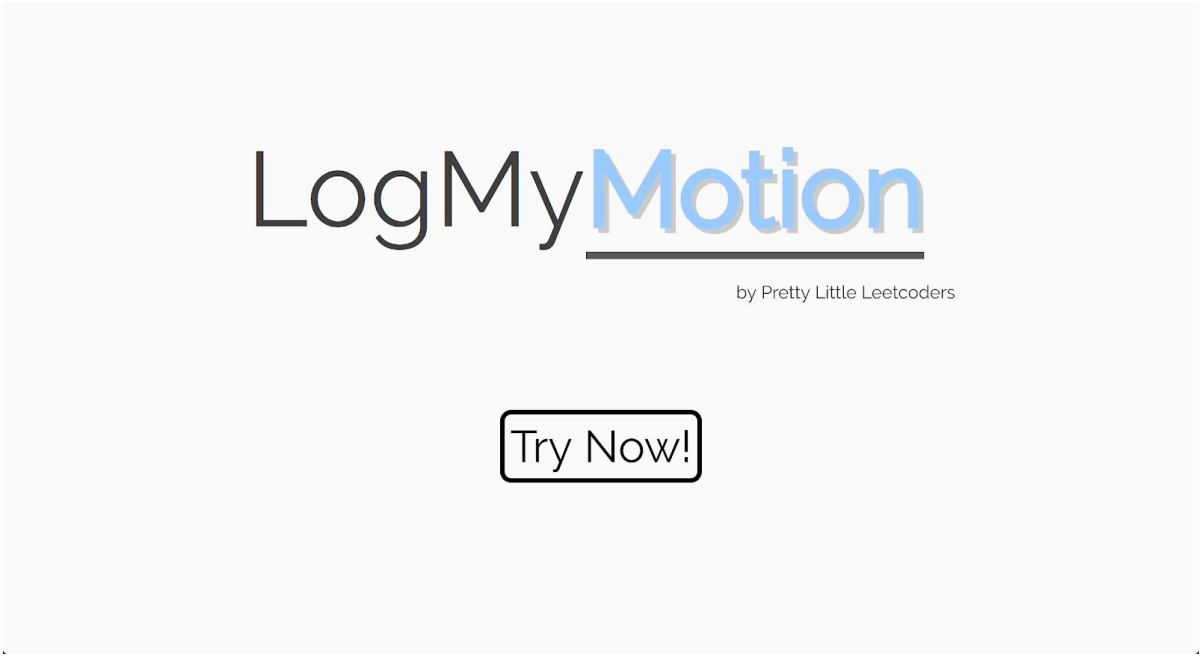
High-level Diagram



User Interaction and Design

Our web application consists of one page, which provides the following:

- Overview of our workout analyzer
- Allows visitors to try out the analyzer in real-time
- Provides both visual and audio feedback to user as they're exercising in real time
- Background information of each team member
- Overview of cues the analyzer checks for and importance of each cue



Squat Depth:

Bad

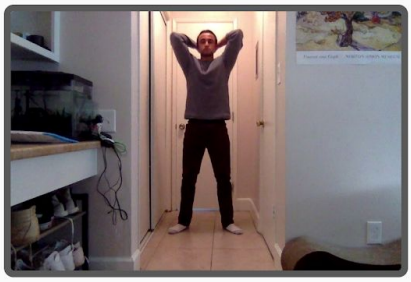
Shoulder Alignment:

Bad

Feet Width:

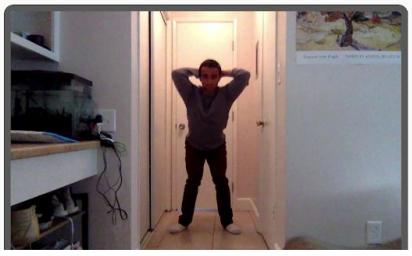
Bad

Cue #1: Squat Depth



It's important to have proper depth when performing squats. Doing so not only ensures that you're properly training the entire lower body musculature, but also indicates that you have excellent overall movement. By performing deep squats, you'll be properly training all the muscle groups (glutes, hamstrings, and quads), improving your posture, making your knees stronger, and training your muscles to be more explosive. This is greatly beneficial to both experienced athletes and individuals looking to stay in shape.

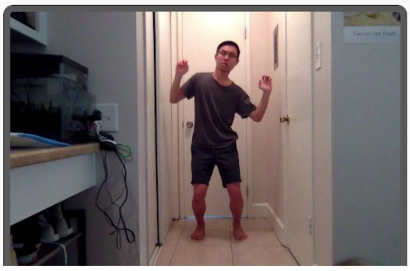
If you're not going low enough when squatting, you're most likely not getting the most out of the exercise. In addition, if you're unable to perform deep squats, it might indicate that you have mobility, stability, or technique issues. This is fine; not everyone starts out being able to perform exercises properly. If this is the case for you, try stretching your lower body before squatting. Start with no weights and just focus on your form and your depth. After some time, you'll gradually notice an improvement in your overall form.



Cue #2: Shoulder Alignment

When performing a squat, make sure that your shoulders stay aligned and that you're not leaning too much to one side. This is especially important if you're squatting with a barbell, as it ensures that your weights don't fall off.

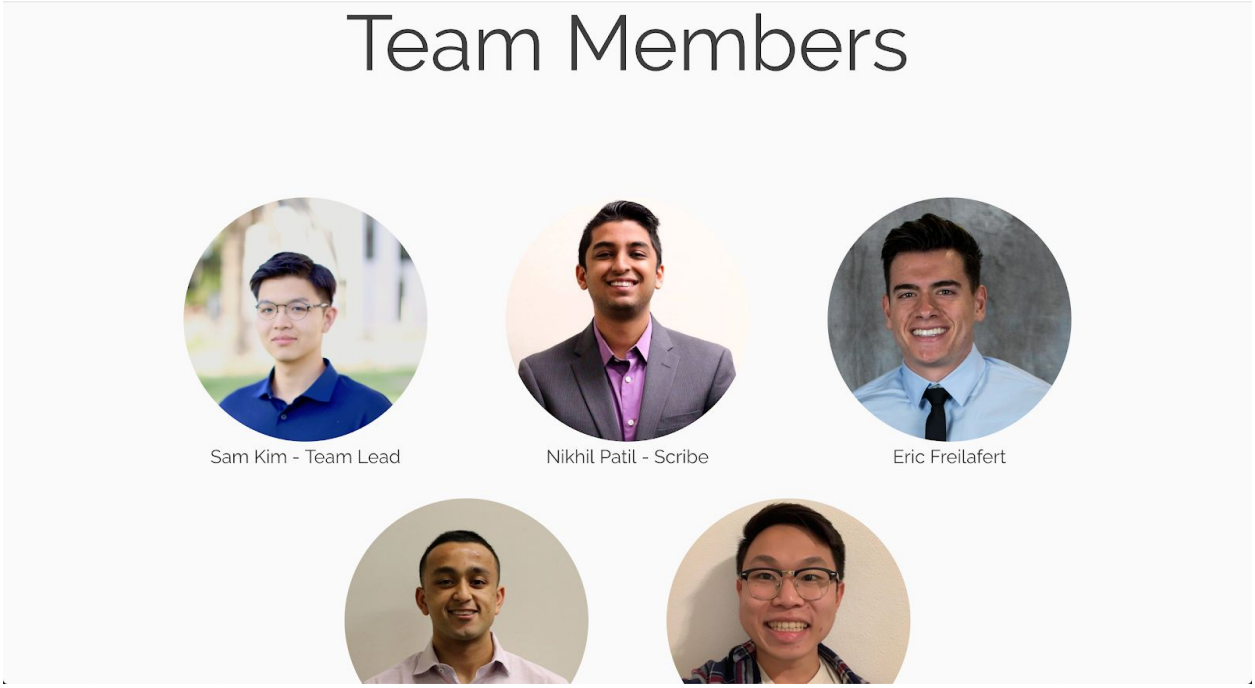
Failure to keep shoulders aligned while squatting can result in serious injury if squatting with a barbell. Leaning even a tiny bit towards one side can cause you to lose balance and lose control of both yourself and the barbell. To avoid this, make sure that you have the same amount of weight on each side of the bar, and that you're squatting evenly.



Cue #3: Feet Width

Before starting your set, make sure that your feet are about shoulder-width apart. By having proper feet width, you'll be able to squat lower and have better control of your motion. This is because the weight is more spread out not just concentrated in one area.

Placing your feet too close together or too far apart can also result in serious injury and will definitely result in improper form. If your feet are too close together, you won't be able to squat deep enough or have your knees properly angled. Placing your feet too far apart will result in you having less control of your movement and will cause you to not optimally work all the targeted muscles.



Requirements

Use case: Login
Actors: User, User Account Database.
System Precondition: User must be in the login page.
Flow of Events: Basic Path: 1. User types in username/password. 2. System checks the database to confirm if user is valid. If the user credentials are valid, then the user information will be loaded.
Alternative Paths: If the user credentials are not valid, an error will be thrown and the user will be asked to enter a valid username/password.
Postcondition: User should now be in the user's home page of our web application.
Github commit link: Users should be able to login

Use case: Upload video
Actors: User, User Account Database
System Precondition: User has logged in.

Flow of Events: Basic Path: 1. User clicks upload video. 2. User chooses video from computer (or cloud drives). 3. Upon confirming, video will now be stored in system database to be analyzed.
Alternative Paths: If the file size is too big than our system accounts for, an error message will be thrown and the user will be asked to upload a file within the size limit.
If the upload fails, an error will be thrown and the user will be asked to try again.
Postcondition: User should be able to see the video he/she uploaded in her uploaded videos either in an analyzed or to be analyzed state.
Github commit link: Figured out real time webcam feed

Use case: Analyze squat depth
Actors: System Workout Analyzer
System Precondition: There is either an uploaded video in the "to be analyzed" state in the system's database or a real-time video feed streaming from the user's webcam
Flow of Events:
1. System either retrieves video to be analyzed from the database or real-time video stream from webcam
2. System applies our algorithm to determine the grade of the user's squat depth.
3. If video was retrieved from database, the system updates the state of the video in the database and uploads analysis of the workout to the front-end; if video was real-time stream, then system provides analysis in real-time to the front-end
Alternative Paths: If the algorithm fails for some reason, an error will be thrown and the user will be informed that the analysis was not performed.
If the video is not found in the database, an error will be thrown that the video was not found in the database.
If application can't get access to real-time video stream, an error will be thrown that application can't access webcam
Postcondition: An analysis of the user's squat depth will be shown in the respective user's page.
Github commit link: Final squat depth cue for demo

Use case: Analyze shoulder alignment
Actors: System Workout Analyzer
System Precondition: There is either an uploaded video in the "to be analyzed" state in the system's database or a real-time video feed streaming from the user's webcam
Flow of Events:
1. System either retrieves video to be analyzed from the database or real-time video stream from webcam
2. System applies our algorithm to determine the grade of the user's shoulder alignment.
3. If video was retrieved from database, the system updates the state of the video in the database and uploads analysis of the workout to the front-end; if video was real-time stream, then system provides analysis in real-time to the front-end

Alternative Paths: If the algorithm fails for some reason, an error will be thrown and the user will be informed that the analysis was not performed.
If the video is not found in the database, an error will be thrown that the video was not found in the database.
If application can't get access to real-time video stream, an error will be thrown that application can't access webcam
Postcondition: An analysis of the user's shoulder alignment will be shown in the respective user's page.
Github commit link: Final shoulder align cue for demo

Use case: Analyze feet width
Actors: System Workout Analyzer
System Precondition: There is either an uploaded video in the "to be analyzed" state in the system's database or a real-time video feed streaming from the user's webcam
Flow of Events:
1. System either retrieves video to be analyzed from the database or real-time video stream from webcam
2. System applies our algorithm to determine the grade of the user's feet width.
3. If video was retrieved from database, the system updates the state of the video in the database and uploads analysis of the workout to the front-end; if video was real-time stream, then system provides analysis in real-time to the front-end
Alternative Paths: If the algorithm fails for some reason, an error will be thrown and the user will be informed that the analysis was not performed.
If the video is not found in the database, an error will be thrown that the video was not found in the database.
If application can't get access to real-time video stream, an error will be thrown that application can't access webcam
Postcondition: An analysis of the user's feet width will be shown in the respective user's page.
Github commit link: Final feet width cue for demo

Use case: Analyze knee angle
Actors: System Workout Analyzer
System Precondition: There is either an uploaded video in the "to be analyzed" state in the system's database or a real-time video feed streaming from the user's webcam
Flow of Events:
1. System either retrieves video to be analyzed from the database or real-time video stream from webcam
2. System applies our algorithm to determine the grade of the user's knee angle.
3. If video was retrieved from database, the system updates the state of the video in the database and uploads analysis of the workout to the front-end; if video was real-time stream, then system provides analysis in real-time to the front-end
Alternative Paths: If the algorithm fails for some reason, an error will be thrown and the user will be informed that the analysis was not performed.

If the video is not found in the database, an error will be thrown that the video was not found in the database.
If application can't get access to real-time video stream, an error will be thrown that application can't access webcam
Postcondition: An analysis of the user's knee angle will be shown in the respective user's page.
Github Commit Link: Issue for Knee Angle Analysis

Use case: Visual Feedback
Actors: System Workout Analyzer
System Precondition: Uploaded video or real-time video stream has been processed and analyzed
Flow of Events:
1. System retrieves numeric results of video/stream
2. System applies our grading criteria to each cue and separates into "good", "okay", or "bad" categories
3. System displays different colors for each cue on the front-end based on category, where green is "good", yellow is "okay", and red is "bad".
Alternative Paths: If user has just started the exercise, or video/stream footage can't be retrieved, cues on the front-end won't have any colors assigned to them yet
Postcondition: Cue button on the front-end will have a color assigned to it based on the grading criteria of numerical information provided by workout analyzer
Github commit link: Good is green, bad is red (commit to visually display cue grades)

Use case: Audio Feedback
Actors: System Workout Analyzer
System Precondition: Uploaded video or real-time video stream has been processed and analyzed
Flow of Events:
1. System retrieves numeric results of video/stream
2. System applies our grading criteria to each cue and separates into "good", "okay", or "bad" categories
3. System provides real-time speech feedback for each cue if they're in the "okay" or "bad" categories
Alternative Paths: If user has just started the exercise, or video/stream footage can't be retrieved, there'll be no audio feedback to provide
If Amazon Polly service is down, no audio feedback will be provided
Postcondition: Once a user completes a repetition, they'll receive real-time audio feedback telling them what to improve on for the next rep

Use case: Calibrate user's position in relation to webcam
Actors: User, System
System Precondition: User has just opened the application in real-time mode

Flow of Events:
1. User stands in front of webcam while application is in real-time analysis mode.
2. System determines distance of user from the webcam and angle of webcam.
3. If user is too close to webcam, system notifies user that they should increase distance from the webcam. If the user is too far away from the webcam, system notifies the user that they should move closer. If the webcam is too low, system notifies user that they should move the webcam higher up. If the webcam is too high, system notifies user that they should move the webcam lower.
Alternative Paths: If the system calibration fails, system moves on to next part of analysis.
Postcondition: User is in a good position for squat analysis to be performed.
Github Commit Link: Issue for Calibrating User's Position

Use case: Draw keypoints over detected joints
Actors: System
System Precondition: User is using the application in real-time mode
Flow of Events:
1. User stands in front of webcam while application is in real-time analysis mode.
2. System identifies coordinates of each joint.
3. System draws ellipses over each detected joint and creates skeleton of the user.
Alternative Paths: If the joints are not detected with a high enough confidence level, the joints will not be drawn over.
Postcondition: User's joints have ellipses drawn over them to indicate that they are being detected by our algorithm.
Github commit link: Posenet is now running...

Use case: Count repetitions
Actors: System
System Precondition: Video to be analyzed must exist.
Flow of Events:
1. System retrieves a video to be analyzed in the database.
2. System applies our algorithm to count the amount of repetitions of the specified workout in the video.
3. System will update the repetition count to be displayed.
Alternative Paths: If the algorithm fails for some reason or the user performs the workout erroneously, an error will be thrown and the user will be informed that the repetition wasn't counted.
Postcondition: A repetition will be incremented and displayed on webpage near the webcam feed fragment.
Github commit link: Allow users to see the amount of repetitions that they have performed

Use case: Display cue information
Actors: System
System Precondition: User is in the home page

Flow of Events:
1. System loads a good and bad example video of every cue
2. System loads a description of cue and why it is important to follow
Alternative Paths: N/A
Postcondition: The cue grades are displayed for the user
Github commit link: Finish information for all cues

Use case: Delete video
Actors: User, User account database.
System Precondition: A user has uploaded a video before and is currently showing in the uploaded videos page.
Flow of Events: Basic Path: 1. User selects video from uploaded videos. 2. Upon confirmation of video chosen to be deleted, system will remove the video from the database.
Alternative Paths: Video is not found in database
Postcondition: Selected video will no longer appear in user's list of uploaded videos (as well as the database).

Use case: Logout
Actors: User, User account database.
System Precondition: A user has to be logged into their account.
Flow of Events: Basic Path: 1. User clicks the logout button. 2. The user is brought back onto the home page.
Alternative Paths: If the user does not log out, a cookie would be used to set a timer for how long the user could be long on for.
Postcondition: The cookie should be deleted and the user should be logged out and be on the homepage.

Use case: Check workout results history
Actors: User, User account database
System Precondition: User has uploaded sufficient number of videos to be analyzed
Flow of Events: User opens progress page. System analyzes data from user videos in relation to each other. If system determines that user has had multiple videos in a row with good grades in cues, system recommends that the user increase weight. This will only happen if the user has received good grades for each of the cues multiple times in a row.
Alternative Paths: If the user has not uploaded sufficient number of videos for progress to be checked, the user will be notified that they need to upload more videos.
Postcondition: User is redirected to homepage. New progress bar is set.

Use case: Create an Account
Actors: User, User account database
System Precondition: User does not have an account already for a specific email.
Flow of Events: User clicks on button to create an account. User enters information. Account is created and added to backend storage system.

Alternative Paths: If the email already has an account associated with it, the user is informed that an account already exists.
If the user's password is invalid, the user is informed to create a password within the required constraints.
If the system fails to create an account for the user, the user is informed that account creation failed.
Postcondition: User is brought into the application signed in with their account.
Github commit link: Users should be able to create an account

Use case: Social media integration
Actors: User
System Precondition: User's video or real-time video stream has been analyzed
Flow of Events:

1. User clicks on "Share results" button
2. Popup appears allowing user to choose which social media to share on
3. User picks social media to share on, and pre-drafted post on chosen social media appears
4. User modifies post and clicks "post"

Alternative Paths: User exits out of popup instead of posting; chosen social media platform could be down, not allowing the user to post
Postcondition: User has shared their analyzed workout video/stream results to their chosen social media platform

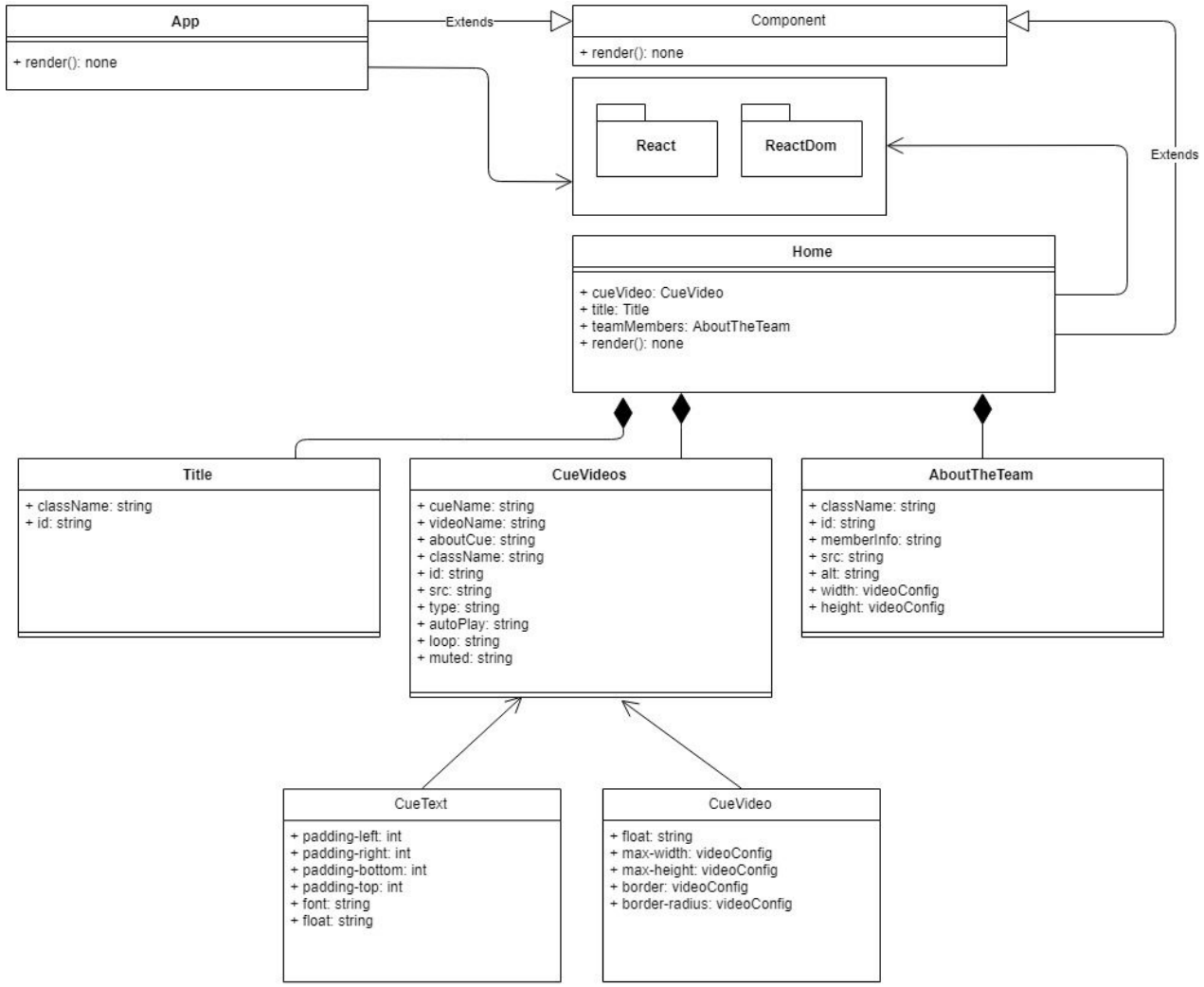
Use case: Create/Edit Profile
Actors: User, User account database.
System Precondition: User just created their account and has not set up the profile. User wants to make changes to their profile.
Flow of Events: User goes to their profile. User changes their info on their profile. User saves the profile.
Alternative Paths: User's profile fails to update, so user is notified. User cancels out from editing a profile.
Postcondition: User has updated their profile information

Use case: Analyze Audio
Actors: System
System Precondition: The video has been uploaded to the server
Flow of Events: 1. System retrieves audio file to be analyzed in the database.
2. System applies our algorithm to determine the grade of the breathing.
3. System will upload analysis of the workout to the respective user's database.
Alternative Paths: Audio file is not found, so error is thrown and user is notified that their audio was not found.
Postcondition: User is able to view analysis of their breathing on their homepage.
Use case: Suggest improvements

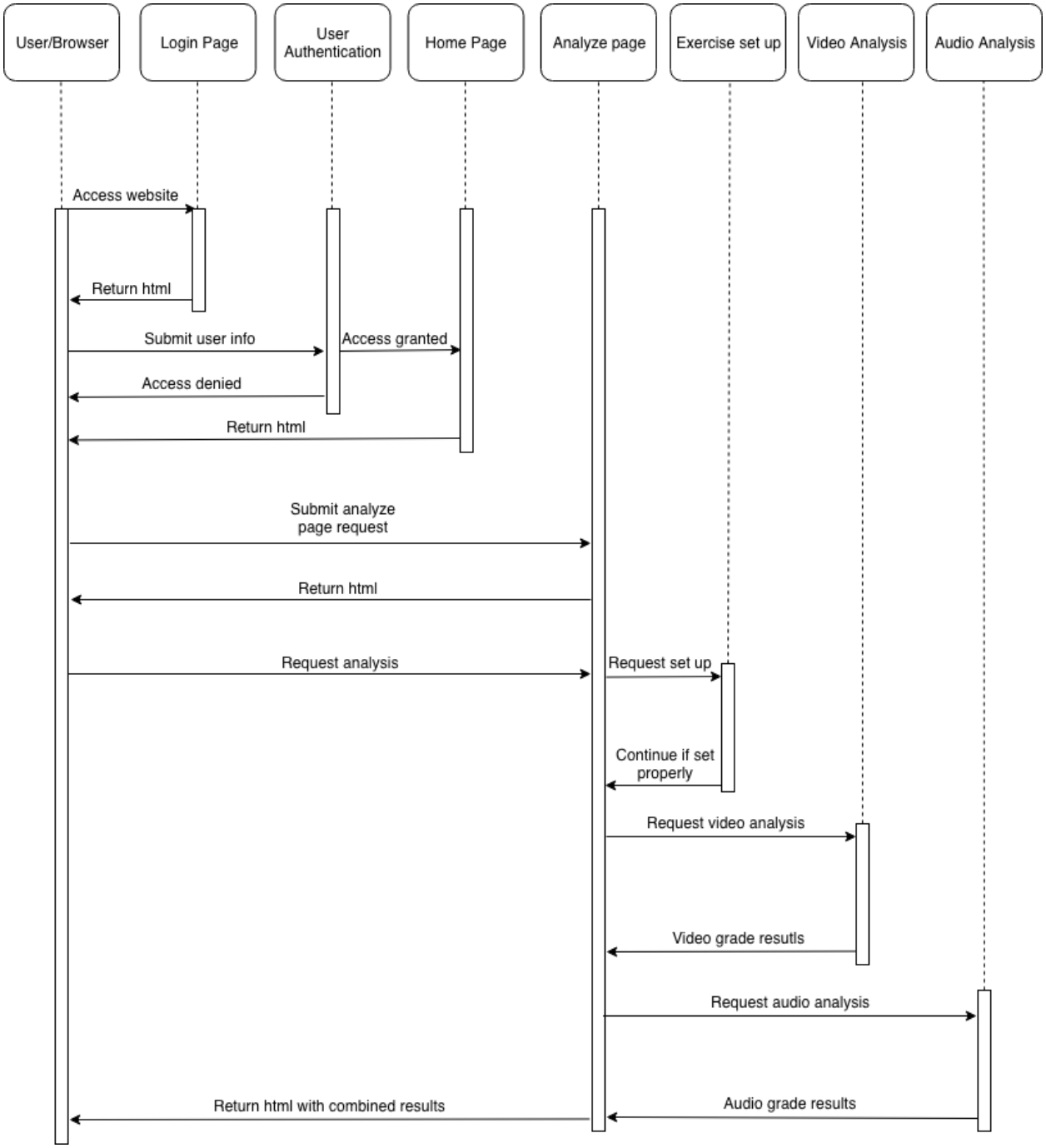
Actors: User
System Precondition: Website has loaded into the user's browser
Flow of Events:
1. User fills out form with their email and their suggestions
2. User clicks "send"
3. System forwards user's suggestions to our email
Alternative Paths: If user entered an invalid email, will tell user to use properly-formatted email
If any of the required boxes were left blank, will tell user to fill out the required fields.
Postcondition: User has sent email to our inbox with their suggestions to improve LogMyMotion

System Models

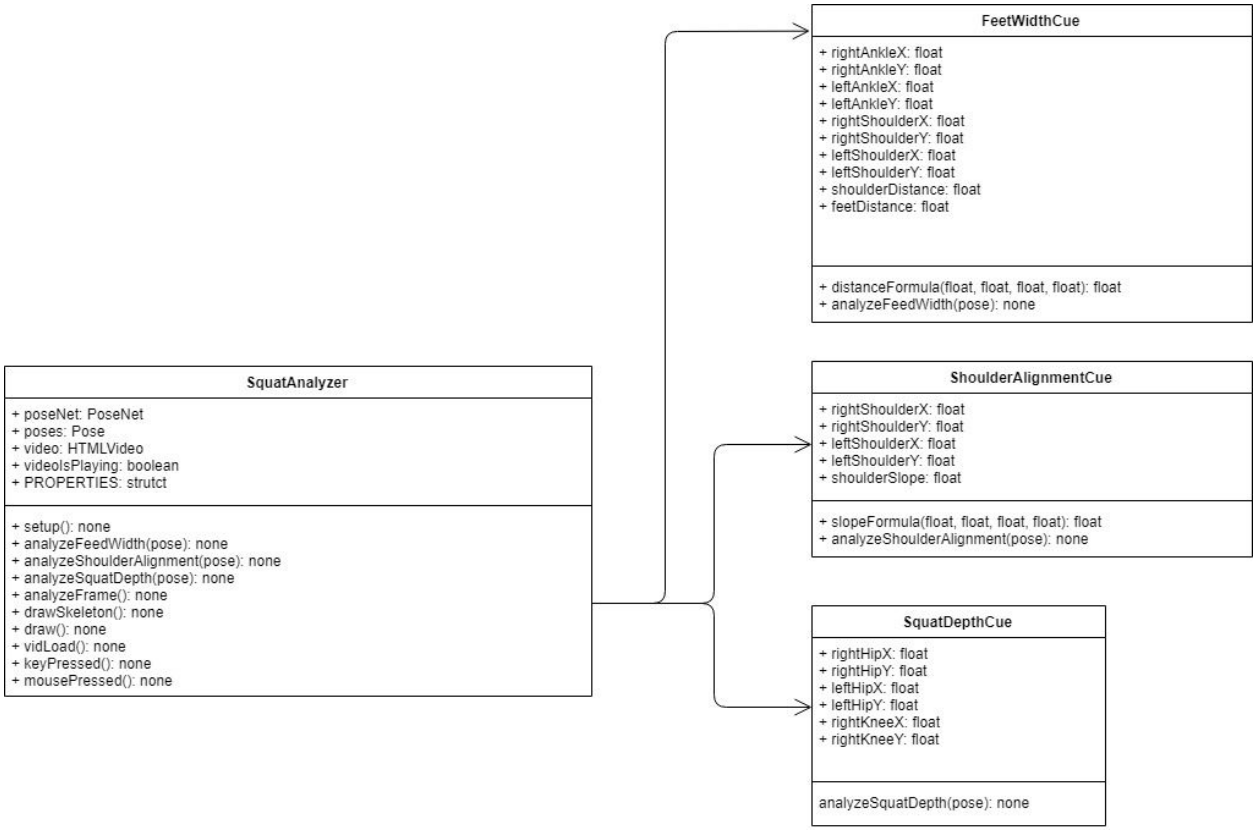
Web Services (Class diagram)



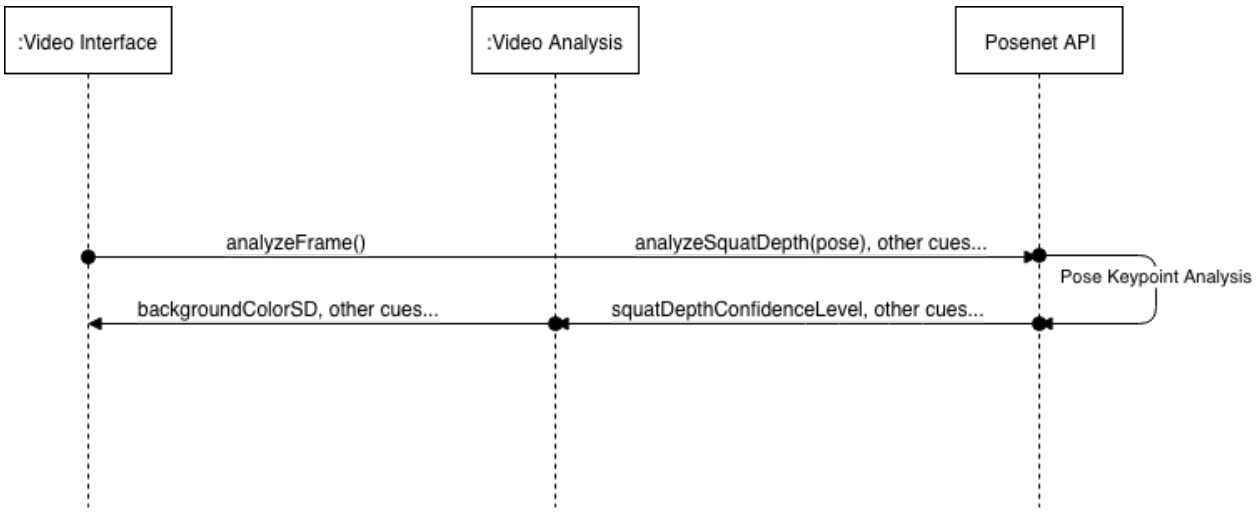
Web Services (Sequences diagram)



Video Analysis (Class diagram)



Video Analysis (Sequences diagram)



Appendices

Technologies Employed:

- Front end
    - React.js
    - HTML
    - CSS
    - Javascript
- Back end
    - Google Firebase
- Video Processing
    - Tensorflow.js
    - PoseNet
    - FFmpeg
- Text Processing
    - Amazon Polly