

Lift Track

Project Description:

This application offers a personalized fitness experience by requiring users to create unique accounts and log in to access their dashboard. After the user has logged in, they will be directed to a home page where they will be provided with an introduction to their fitness options and their daily schedule. The user will then be able to navigate to the exercises page, tracking page, my workouts page, and my plan page. The exercises page allows users to look for specific exercises and provides them with basic info on them, such as what muscles they work and how to do them through videos, implemented with a Youtube API. The tracking page holds the user's current goals, personal records, and reps and weights of workouts they completed. The my workouts page has presets for various workout types, along with functionality allowing the user to choose from and create presets for different days of workouts. These presets will include all the different workouts that person will do for that day. The my plan page is where the user can assign different workouts to different days and fully modify their weekly schedule.

Member Details:

Lana Reeves: LanaReeves

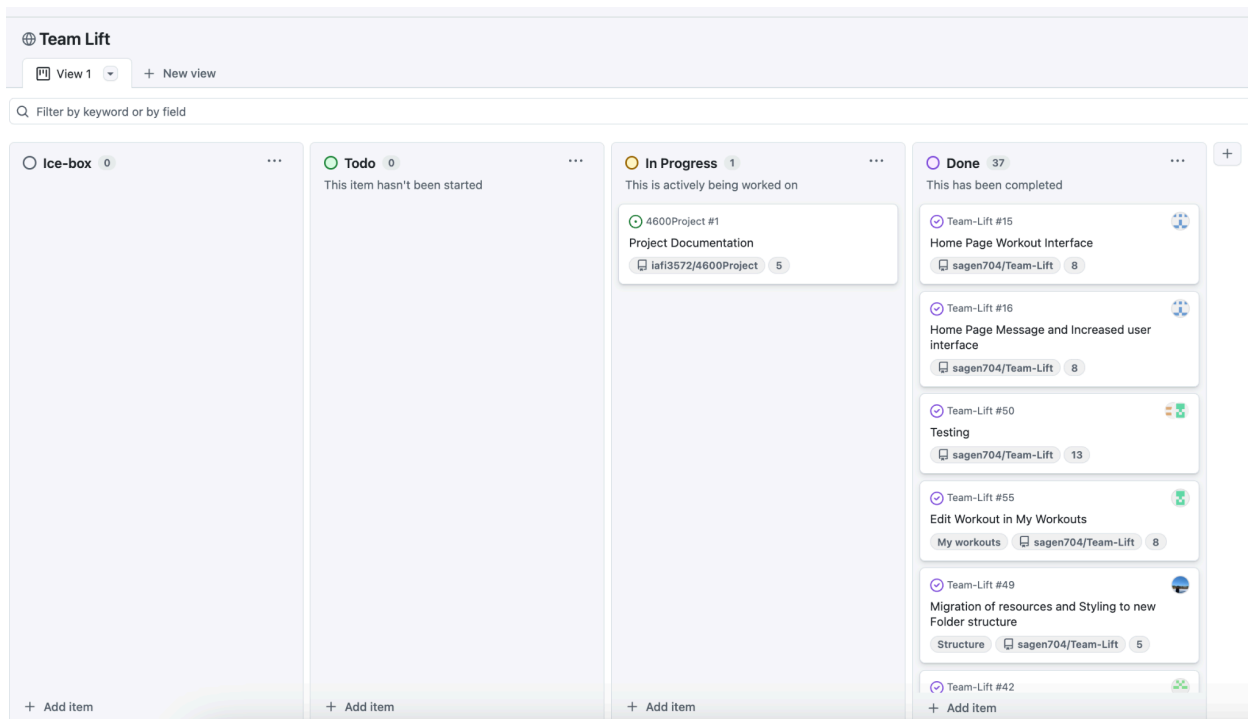
Samuel Hagen: sagen704

Matthew Cooper: matthcopr

Kirin Kawamoto: kika3780

Ian Fischer: iafi3572

Project Tracker: <https://github.com/users/iafi3572/projects/1>



Video: <https://youtu.be/RDwW2vyydJw>

VCS: <https://github.com/sagen704/Team-Lift>

Contributions:

Lana: I worked on the register page, login page, and the my workouts page. For the register and login page, I created the user interface and the users database. For the my workouts page, I created the ability to create a workout or add a default workout, edit the workout, and delete the workout. This consisted of creating the UI, 4 databases, and different API routes. Also, I created some of the automated test cases and helped host the website on Render.

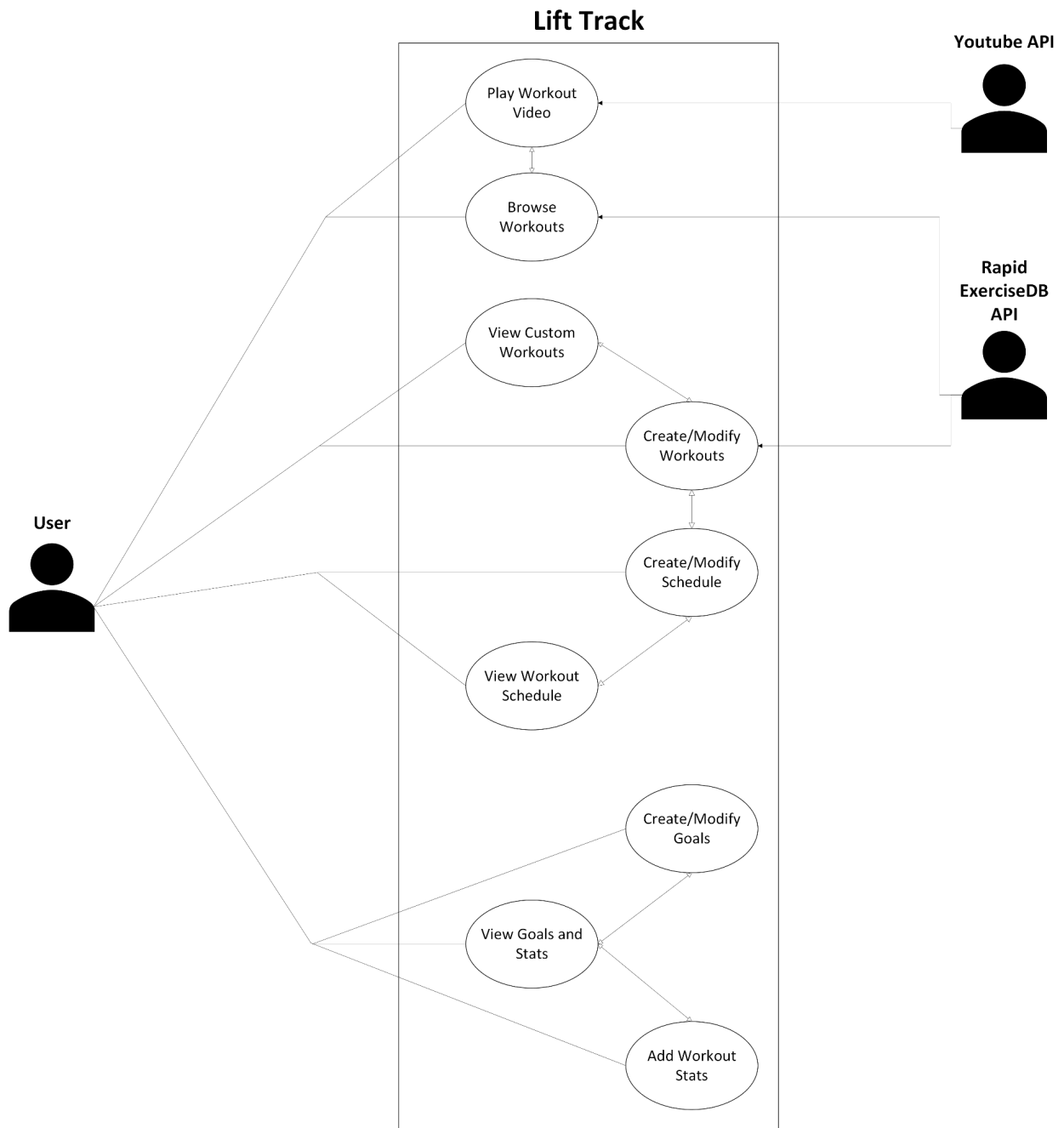
Sam: I worked on the exercise page and the styling. For the exercise page, I needed to find an API that could pull a large amount of exercises, and I make requests to the Youtube API to get a tutorial for each of the exercises. I also styled the website so it maintains a consistent style that reminds people of the gym. I worked on the test cases to ensure route behavior. Lastly, I hosted the website on Render.

Matthew: I created the message, nav, and footer partials guaranteeing their full functionality. I designed the front and back end of the tracking page ensuring that it connected with the other pages and database. This involved connecting the data displayed to the tracking/ analytics page with the exercise defined for the day which was connected to the calendar making sure it only displayed for the right day.

Kirin: I set up the NodeJS server and Express framework to enable HTTP routing and database operations. My main contribution was building the "My Plan" page, which features a weekly calendar for scheduling, editing, and viewing workouts. I developed UI templates for each day, implemented modals for adding, editing, and deleting workouts, and created backend routes and databases to manage the workout schedule.

Ian: I designed and implemented the functionality and look of the home page. This includes displaying the date, today's workout, welcome message, descriptions of the pages of the site, and everything else on the page. The today's workout section interfaces with the my plan and my workouts page, pulling this information in to display the users workout plan for the day. I also created the Github repository, project board and did a significant portion of the various writing parts of this project (Lab-9 write up, this report, slideshow presentation).

Use Case Diagram:



Wireframes:

Home exercise My fitness > logout

Login

username

Password

Login

Home exercise My fitness > logout

Register

username

email

Password

...


Register

Home exercise My fitness > logout

You have been logged out

Home exercise My fitness > logout

My Plan

Sun	Mon	Tues	Wed	Thurs	Fri	Sat
+	+	+		+	+	+

Home exercise My fitness > logout

Tracking/Analytics Page

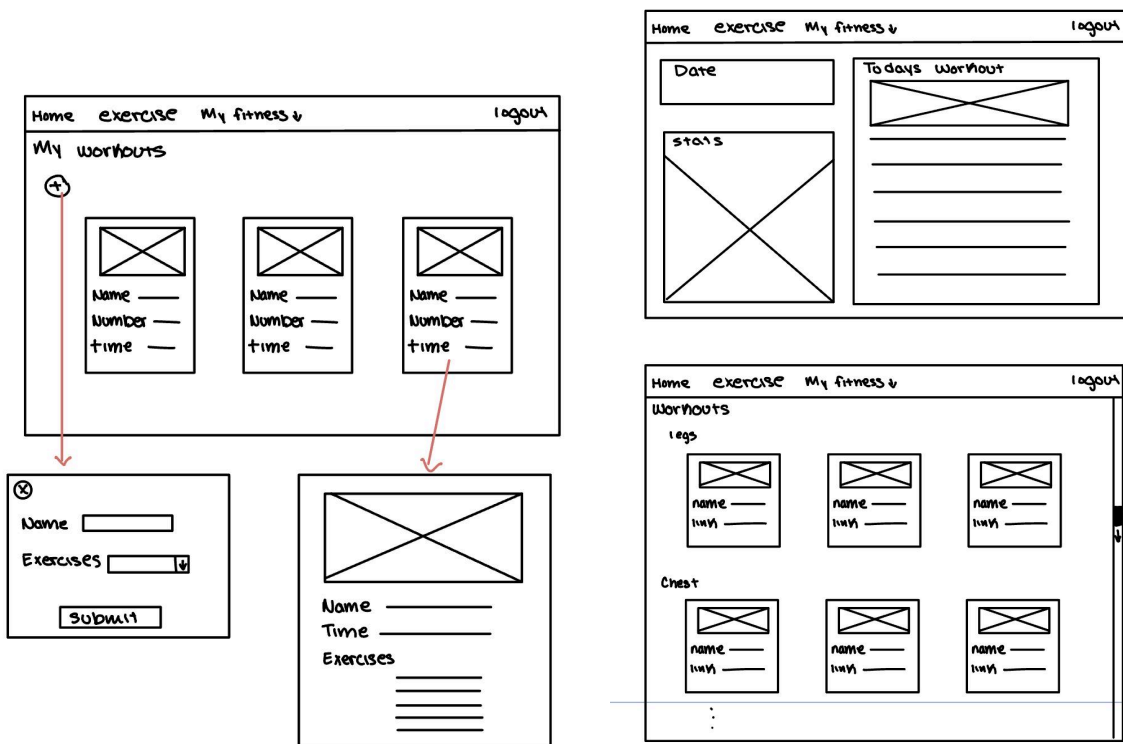
goals 

Average 

Today's Exercise

	Reps	weight
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

Submit



Test results (UAT):

Test Environment: the project was tested in the localhost environment for all test cases

1st Feature:

Acceptance Criteria for Login Page:

- The user should only be able to log in using the username and password they registered with.
- An incorrect username should redirect the user to the register page.
- An incorrect password should not allow login and should create a message stating "incorrect password".
- A successful login should redirect the user to the home page.

Test Case 1: User logs in with correct credentials:

- Description: the user enters the correct username and password
- Data:
 - Username: username that they registered with (ex: test123)
 - Password: password that they registered with (ex: password123)
- User activity (once user has an account):
 - Enter in username
 - Enter in password
 - Click login button

- Test results: User is taken to homepage
- User acceptance: User who workouts
- Actual test results:
 - The user entered their correct credentials and were taken to the home page without any bugs or errors.

Test Case 2: Incorrect username

- Description: the user enters an incorrect username
- Data:
 - Username: Incorrect username (ex: wrong123)
 - Password: Can be correct or wrong password (ex: password123)
- User activity:
 - Enter wrong username
 - Enter password
 - Click login button
- Test results: User is taken to the register page
- User acceptance: User who workouts, but an unregistered user
- Actual test results:
 - The user entered an incorrect username and the user was successfully directed to the register page without bugs.

Test Case 3: Incorrect password

- Description: The user enters an incorrect password
- Data:
 - Username: correct username (ex: test123)
 - Password: wrong password (ex: incorrectpassword123)
- User activity:
 - Enter correct username
 - Enter wrong password
 - Click login button
- Test results: User stays on login page, but message displayed stating “incorrect password”
- User acceptance: User who workouts
- Actual test results:
 - The user entered the correct username, but wrong password, and the user successfully stayed on the login page, and the “incorrect password” message was displayed.

2nd Feature: Acceptance Criteria for Schedule Page:

- User should be able to view their workout schedule for the week
- Users should be able to add a workout to their schedule based on a specific day of the week and specific time.
- Users should be able to delete or edit each workout in their plan.

Test Case 1: Adding workout

- Description: The user adds an workout to their schedule
- Data:

- Day: select The day of the week to add the workout to (ex: Sunday)
 - Start time: Select specific start time (ex: 12:00 pm)
 - Workout: Add specific workout (ex: Legs)
- User activity (need to already have an workout created):
 - Go to the “my plan” page
 - Select the add button based on day wanting to add
 - Select the start time
 - Select the workout
 - Click add to schedule button
- Test results: User stays on my plan page and new workout is added to calendar
- User acceptance: User who workouts
- Actual test results:
 - The user was able to click the add button, specify a time and workout for their plan. The workout was successfully added to the plan without any errors.

Test Case 2: Deleting workout

- Description: The user deletes an workout from their schedule
- Data:
 - No data needs to be entered
- User activity:
 - Go to the “my plan” page
 - Click “view more” button on the workout user wants to delete
 - Then click “delete workout from schedule” button
- Test results: User stays on my plan page and the workout is deleted from calendar
- User acceptance: User who workouts
- Actual test results:
 - The user clicked the “view more” button, then the “delete workout from schedule” to delete the workout. The workout was successfully deleted from the calendar without any errors.

Test Case 3: Editing workout

- Description: The user edits an workout in their schedule
- Data:
 - Day: select the new day of the week (ex: Monday)
 - Start time: select the new specific start time (ex: 1:00 pm)
- User activity:
 - Go to the “my plan” page
 - Click “view more” button on the workout user wants to edit
 - Then click “edit workout day/time” button
 - Update day or time
 - Click “submit edit” button
- Test results: user stays on my plan page and the workout edits are updated
- User acceptance: user who workouts
- Actual test results:

- The user clicked the “view more” button, then the “edit workout day/time” button, and put in data to edit the workout. The workout edits were successfully displayed on the plan without errors.

3rd Feature: Acceptance Criteria for Register Page:

- The user should be able to register with their credentials: email, username, and password, and birthday.
- If the username is already taken, the user should receive an error message and be prompted to enter different credentials.
- Upon successful registration, the user should be redirected to the login page and their data should be stored in the database. Mandatory fields
 - username
 - email
 - birthday
 - password and confirm password

Test Case 1: New email/username:

- Description: The user registers for an account
- Data:
 - Username: user that is not already taken (ex: test1)
 - Email: any valid email
 - Birthday: any birthday
 - Password/confirm password: passwords that match
- User activity:
 - Go to the register page
 - Enter in:
 - username
 - email
 - birthday
 - password
 - Confirm password that matches the original password
 - Click the “Register” button
- Test results: Data is added to the database and the user is redirected to the login page.
- User acceptance: User who workouts
- Actual test results:
 - The user entered valid credentials with a username that was not already taken. The user was successfully taken to the login page.

Test Case 2: already taken username:

- Description: The user registers with an username already taken
- Data:
 - Username: user that is taken
 - Email: any valid email
 - Birthday: any birthday
 - Password/confirm password: passwords that match
- User activity:

- Go to the register page
- Enter in:
 - username that is already taken
 - email
 - birthday
 - password
 - confirm password that matches the original password
- Click the “Register” button
- Test results: User stays on the register page and a message is displayed stating “Username already exists”
- User acceptance: User who workouts
- Actual test results:
 - User entered a username that already existed, and they stayed on the register page and the “Username already exists” message appeared with no bugs or errors

Testing observations: The user's behavior was consistent with all the use cases, and the user's actions followed closely to the user activity for each test case; therefore, no changes were made to the application based on the user testing.

Deployment: <https://team-lift.onrender.com/login>