LTS Solutions

Stand-up Meeting

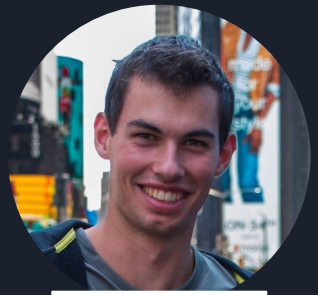
Meet The Team

Christian Layo

Product Manager

Topu Saha

IOS Mobile Engineer



Jules Torfs

Backend Engineer

Stand-up Meeting

01 BUSINESS OVERVIEW

02 UX and UI

03 Backend

04 New Features

05 Budget

Business Stats

Executive Summary:

Workout University is a groundbreaking fitness app designed to revolutionize and optimize the workout experience. Offering a user-friendly platform for accessing and creating workout routines, the app caters to both seasoned gym enthusiasts and newcomers, ensuring a simplified and effective fitness journey.

Purpose:

Empower users to explore various workouts and create personalized routines.

Target Audience:

Fitness enthusiasts, gym-goers, and individuals seeking guidance in their workout routines.

How It Works:

The app's offline functionality ensures accessibility to workouts and routines without the need for an internet connection, specifically catering to users in gym settings. Workout University serves both experienced gym lovers tracking their routines and newcomers exploring diverse exercises to craft personalized workouts. Focused on simplicity, the app stands out in a market often crowded with complex fitness apps, offering clarity in exercise selection and routine creation. Users are likely to integrate the app into their gym sessions for routine guidance and use it during free time to create and plan workout routines.

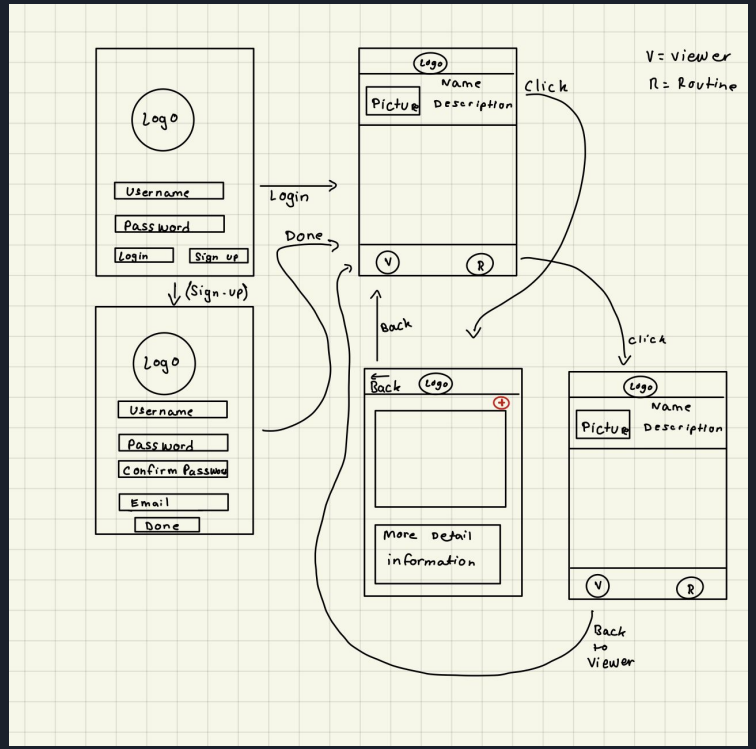
Key Benefits:

Effortless Workout Tracking

Accessible Anytime, Anywhere

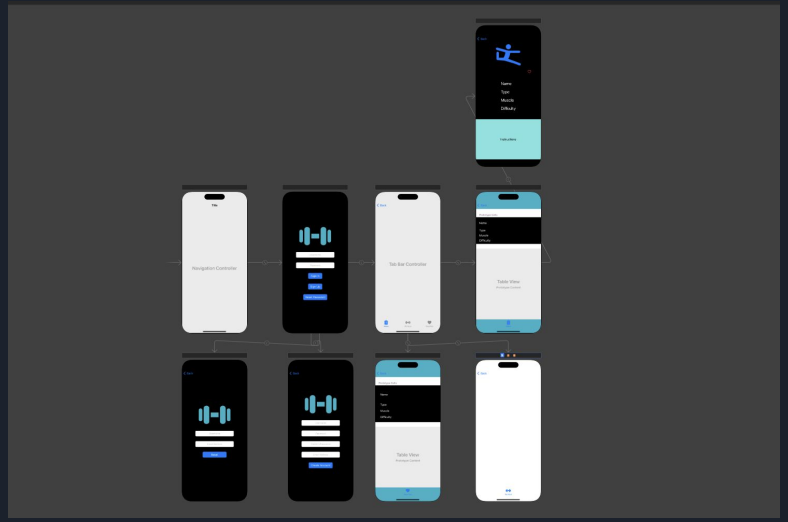
Simplicity and Customization

UX Designer: Wireframes and Control Flows

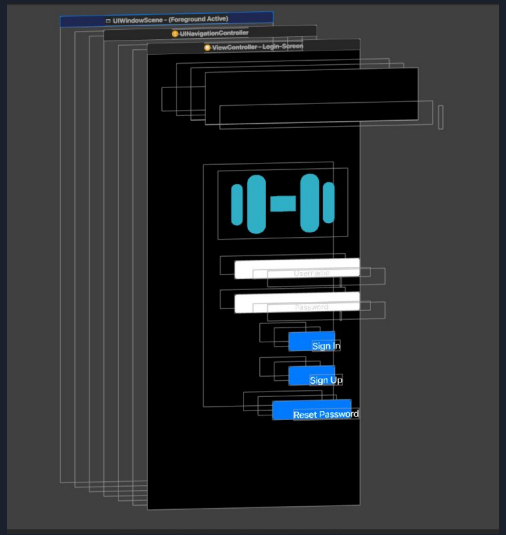


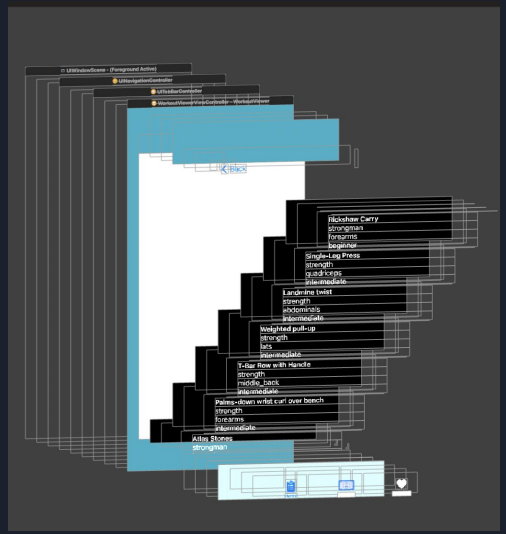
Mobile Engineer: Current Features

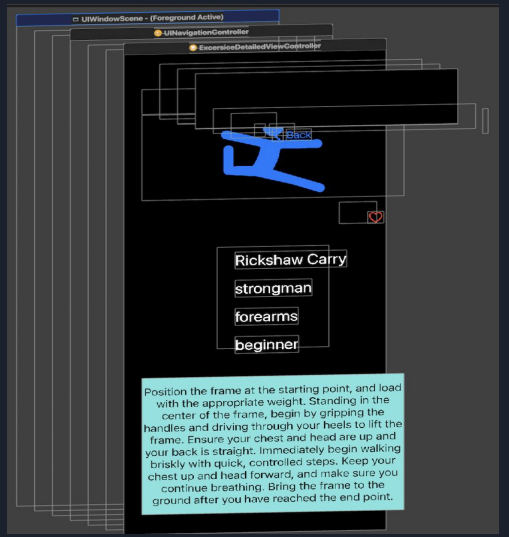
- Navigation Controls
- CPU Affinity
- Local Storage to save workouts



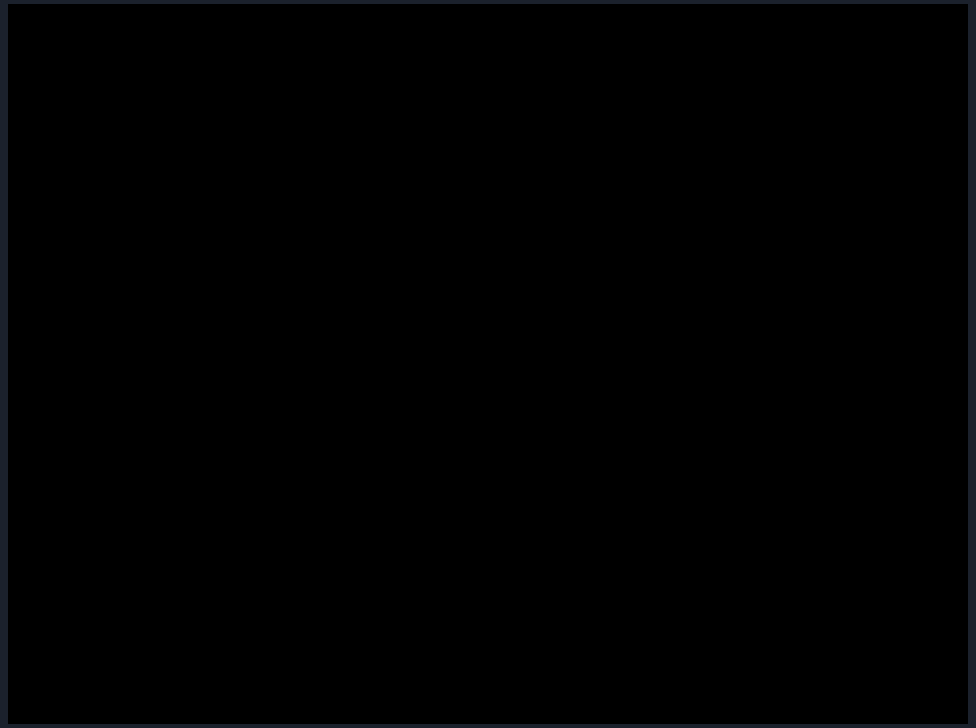
Mobile Engineer: View Hierarchy



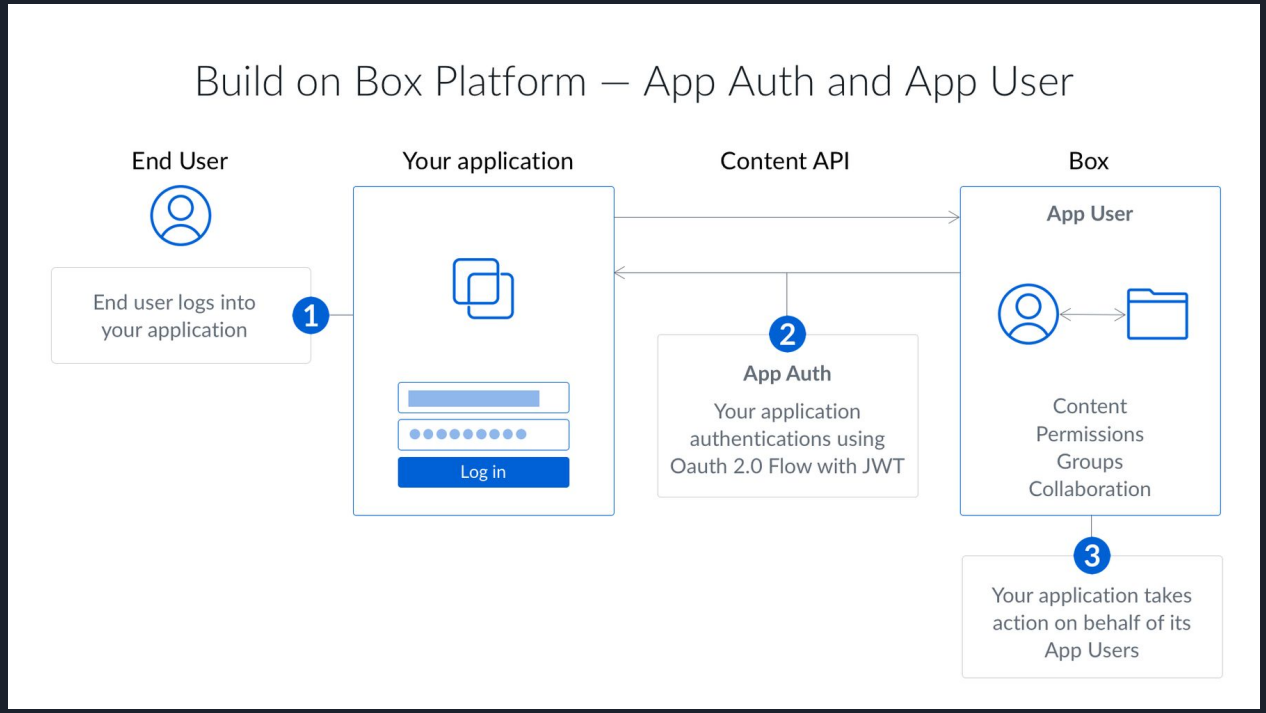




User Experience



Backend Engineer: User Authentication



Docker simulation environment

Development Mode

To run the API in development mode, use the following command:

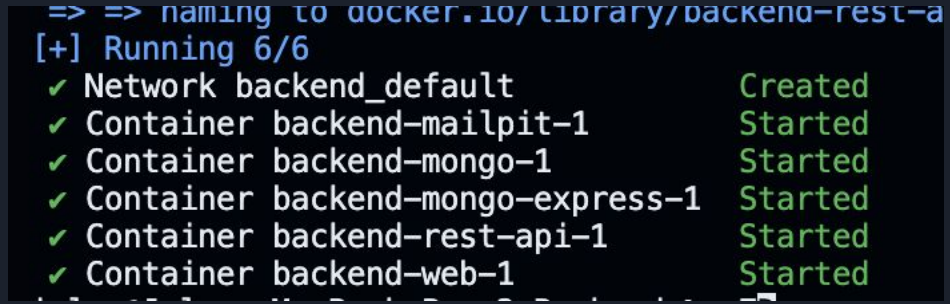
```
make start-dev
```

To stop the API in development mode:

```
make stop-dev
```

To view the API's logs in development mode:

```
make logs-dev
```

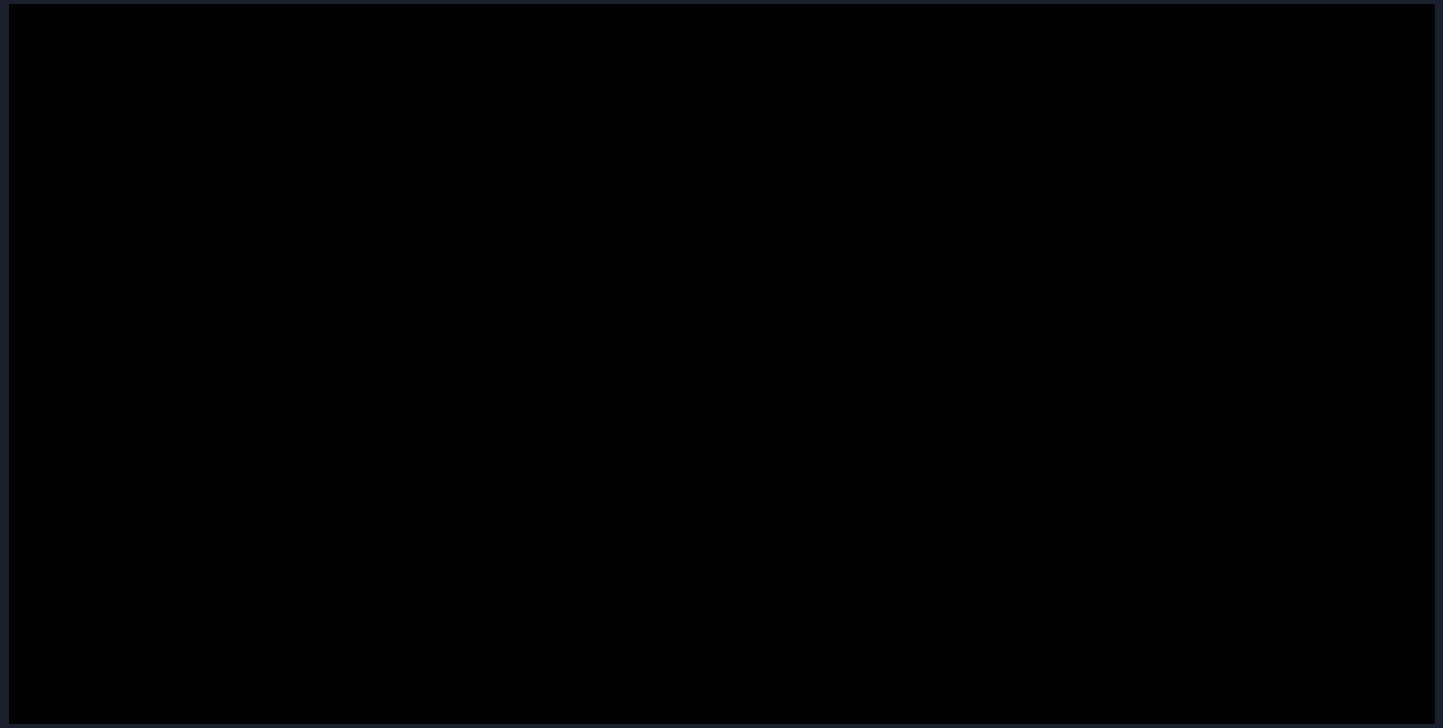


Backend Engineer

- Done
  - Docker simulation environment
  - Implementation email service
  - Added Documentation
  - Fixed bugs with JWT expiring too soon

- To - Do
  - Fix dev Docker-compose file configuration
  - Add more error handling

Email Service



User Authentication Documentation

Get All Users

- **Endpoint:** `/`
- **Method:** GET
- **Description:** Get information about all users
- **Request Body:** N/A
- **Response:** List of all users

Register a New User

- **Endpoint:** `/register`
- **Method:** POST
- **Description:** Register a new user
- **Request Body:**
  - `username`: User's username
  - `password`: User's password
  - `email`: User's email address
- **Response:** JSON Web Token (Token)

Get User Information

- **Endpoint:** `/user`
- **Method:** GET
- **Description:** Get information for a specific user
- **Request Body:**
  - `token`: User's token
- **Response:** User's information

Request Password Reset

- **Endpoint:** `/resetPassword`
- **Method:** POST
- **Description:** Request a password reset for a user, a mail will be sent to the user's email address
- **Request Body:**
  - `email`: User's email address
- **Response:** Success message

QA Testing Procedures

Test Case ID	Scenario	Description
TC001	Log In Button	Log in account to log in to see workouts, able to post pictures on feed for progress, create routines
TC002	Register Button	Creating an account if account does not exist
TC003	Forget Password Button	Reset the password if forgotten
TC004	Workout Viewer	The list of all workouts in the app
TC005	Routine List	The list that has been created when adding from the Workout Viewer
TC006	Adding Workout to Routine	When viewing a workount, you are also able to add it to your routine so you can have a list
TC007	Bottom Screen Buttons	Since there is a Workout Viewer, Routine List, then there should be buttons to be able to navigate
TC008	Exercise API	This API is where it lists all the workouts this app has
TC009	Authorization API	This API is for signing up, signing in, and for resetting the password
TC010	ML Rest API	This API is to populate the Routine List from the Workout Viewer

Preconditions
Must have an account, unique username matching with password
Username must be unique/ new/ not in the dataset, must follow password criteria
To reset it you need to type in the username, and then verify by inputting the email, and if the unique username matches the email in the database then it continues
Account must be existing
Everything before TC005 must be passing
Everything before TC006 must be passing
Everything before TC007 should be passing
Must have API coded and have a list of available workouts to capture and to send to the app
Docker and MongoDB should be created and already available for the API
Workout Viewer must have the workouts filled in, Routine/database shall be ready to be populated

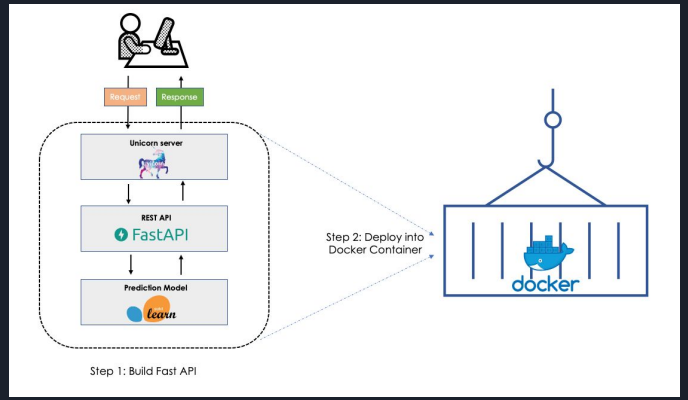
Expected Results
If the unique username is typed, the password matching the username should also be typed and goes into the account, if not then an error of either wrong username and password
If username is not in database, creates account; if username is in database, redirects to login page and say “Username Already Exists”
If username is in database, should have an email linked to the username. So when both are inputted and matches, then the link to reset to password shall be found in the users email
If the account exists, then user is able to view workouts available within the app
If everything before TC005 has passed, then the app should have the routine list should be at the bottom right of the app
If everything before TC006 has passed, then the heart button to add to routine list should work and routine list should be populated
If everything is created and passing, then the buttons on the app should be able to direct them to their respective destinations
If API is fully working, in the Workout Viewer all the workouts should be available to be viewed
If Docker and MongoDB is setup, this API will fill the database with information from the users signing up, with their own personal login tokens which is only accesible in the backend
If preconditions are passing, then the API will catch the workouts that the user has favorited and then add to the Routine List

New Feature: Workout Mode

- Business Case: Users might feel daunted to start using the gym due to lack of knowledge and limitless amounts of resources on the web on how to start working out. What is one way Workout University can reduce this fraction to allow users to feel at ease and start without with less hesitation.

New Features: Machine Learning Approach

- Two Approaches
  - Rule Based Machine Learning
  - Complex Thought Process with Neural Networks ex: LLMs
- System Designs
  - Rule Based: Given Conditions come up with a workout routine
  - Neural Network: API



Budget

- Total Budget: $1,000,000
- Total Employee Salary: $850,000
- Authentication API Hosting: $20,000
  - Annual cost of hosting the authentication API
- Machine Learning REST API Hosting: $30,000
  - Annual cost of hosting the machine learning REST API
- Other Expenses: $25,000
  - Includes costs such as software licenses, hardware, and training
- Total: $925,000
- Amount Left Over for Performance Bonuses: $75,000

Questions?