FitGenius System Design Document

Project Title: FitGenius - Personalized Fitness and Workout App

Date: Sun Nov 17, 2024

Prepared by: FitGenius Team

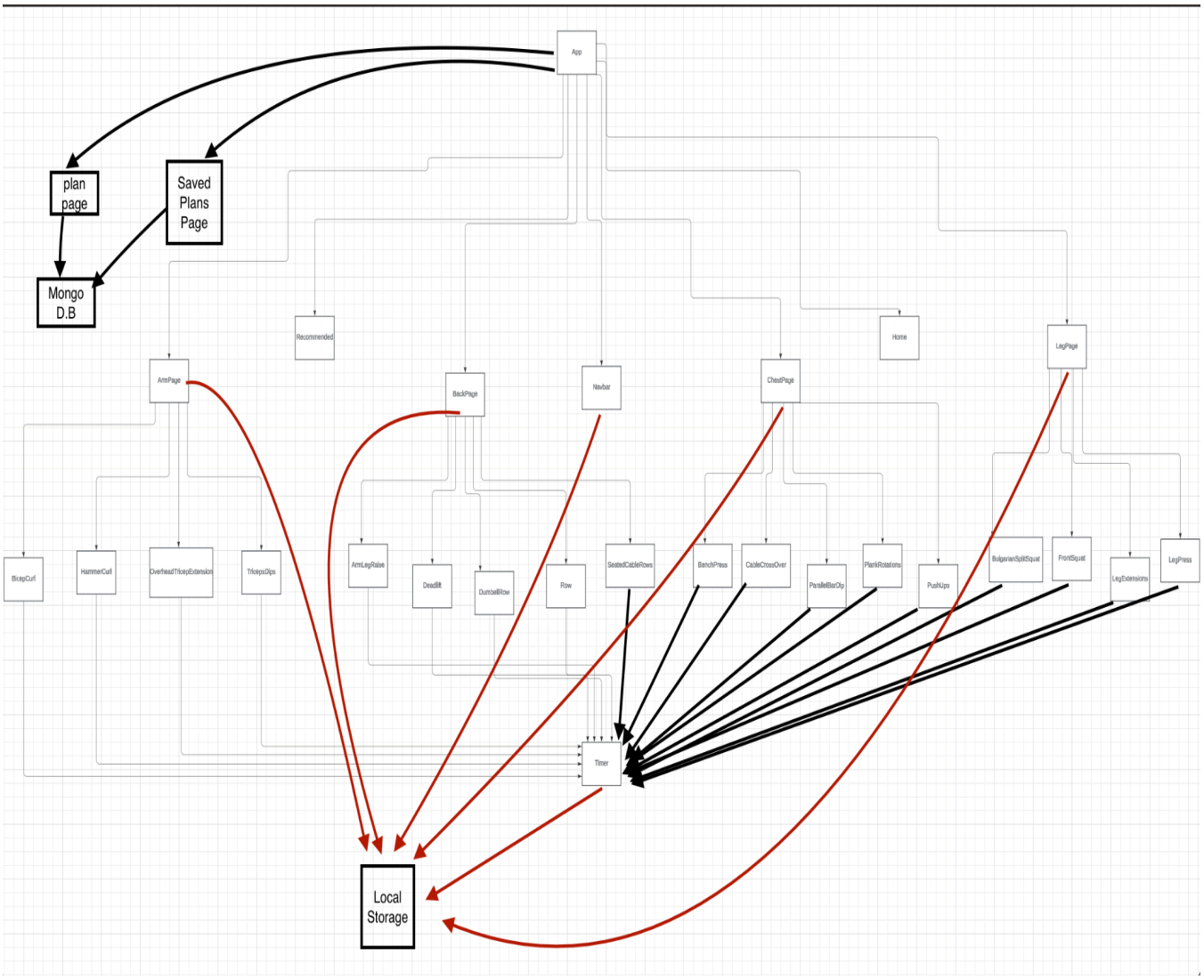
Table of Contents

High-Level Architecture Overview

(1.1)

FitGenius is made up of a primary application component that controls the application's routing. The App includes the Navbar component, which lets users move between pages such as Home, Recommended, Plan Page, and Saved Plans Page. Recommended is where you will find each of the four primary muscle group pages (Arm, Back, Chest, and Leg) that users can access from the app. Additionally, every muscle group page has an intensity feature available where users can choose an intensity (low, med, high) that will correspondingly change the duration of the workout in every exercise page (such as BicepCurl and HammerCurl).

(1.2)



(2.1)

Class Name: App	**Responsibilities:**	**Collaborators:**
Parent Class: None **Subclasses:** Home, Recommended, ArmPage, ChestPage, LegPage, BackPage	• Set up routing for the application • Render Navbar and main page components based on the route • Serve as the root component	• Navbar (for navigation links) • Home, Recommended, LegPage, ChestPage, BackPage, ArmPage, and all specific exercise components (like BenchPress, CableCrossover, etc.) • React Router (Route, Routes) for navigation

(2.2)

Class Name: Home	**Responsibilities:**	**Collaborators:**
Parent Class: App **Subclasses:** None	• Display the main title of the application ("Fit Genius") • Serve as the landing page for the application	• None (it's a standalone component without direct dependencies on other components)

(2.3)

Class Name: Navbar	**Responsibilities:**	**Collaborators:**
Parent Class: App **Subclasses:** None	• Display navigation links for key pages • Provide links to Home and Recommended pages • Provide Plan Page, and Saved Plans Page. Plan Page allows users to plan their custom workout and has a save option which saves the workout to Saved plans Page..	• Link from react-router-dom to enable navigation without page reloads • Home and Recommended components/pages as linked destinations

(2.4)

Class Name: Plan Page	**Responsibilities:**	**Collaborators:**
Parent Class: App **Subclasses:** None	• Display a selection box for each muscle group, allowing the user to choose exercises for their plan • Display a selected exercise at the bottom of the page. When clicked, it should navigate to a page that displays a timer. • Provides an option for user to select intensity level (low, med, high) through radio buttons • Stores intensity level in a variable, that is stored in local storage	• UseNavigate from React Router for navigation • Use Select from React Select to choose one or multiple items from a list of exercise options

(2.5)

Class Name: Saved Plans Page	**Responsibilities:**	**Collaborators:**
Parent Class: App **Subclasses:** None	• Displays a custom saved workout that was created using the Plane Page •	• TimerWidget: Used to track and display the time associated with the exercise. An animated GIF image imported to visually demonstrate the exercise form

(2.6)

Class Name: Recommended	**Responsibilities:**	**Collaborators:**
Parent Class: App **Subclasses:** None	• Display recommended muscle groups for workouts • Provide links to muscle group pages (Leg, Chest, Arm, Back) with corresponding images	• Link from React Router for navigation • Imported images (Leg, Chest, Arm, Back) for visual representation of each muscle group

(2.7.1)

Class Name: ArmPage	**Responsibilities:**	**Collaborators:**
Parent Class: App **Subclasses:** None	• Display a list of arm exercises • Provide navigation buttons to specific arm exercise pages (Hammer Curl, Bicep Curl, Tricep Dips, Overhead Tricep Extension) • Provides an option for user to select intensity level (low, med, high) through radio buttons • Stores intensity level in a variable, that is stored in local storage	• UseNavigate from React Router for navigation • Individual exercise components/pages (/hammer-curl, /bicep-curl, /tricep-dips, /overhead-tricep-extension)

(2.7.2)

Class Name: BackPage	**Responsibilities:**	**Collaborators:**
Parent Class: App **Subclasses:** None	• Display a list of back exercises • Provide navigation buttons to specific back exercise pages (Row, Deadlift, Seated Cable Rows, Dumbbell Row, Arm Leg Raise) • Provides an option for user to select intensity level (low, med, high) through radio buttons • Stores intensity level in a variable, that is stored in local storage	• useNavigate from React Router for navigation • - Individual exercise components/pages (/row, /deadlift, /seated-cable-rows, /dumbbell-row, /arm-leg-raise)

(2.7.3)

Class Name: ChestPage	**Responsibilities:**	**Collaborators:**
Parent Class: App **Subclasses:** None	• Display a list of chest exercises • Provide navigation buttons to specific chest exercise pages (Bench Press, Cable Crossover, Parallel Bar Dip, Push Ups, Plank Rotations) • Provides an option for user to select intensity level (low, med, high) through radio buttons • Stores intensity level in a variable, that is stored in local storage	• - useNavigate from React Router for navigation • - Individual exercise components/pages (/bench-press, /cable-crossover, /parallel-bar-dip, /push-ups, /plank-rotations)

(2.7.4)

Class Name: LegPage	**Responsibilities:**	**Collaborators:**
Parent Class: App **Subclasses:** None	• Display a list of leg exercises • Provide navigation buttons to specific leg exercise pages (Leg Press, Bulgarian Split Squat, Front Squat, Leg Extensions) • Provides an option for user to select intensity level (low, med, high) through radio buttons • Stores intensity level in a variable, that is stored in local storage	• useNavigate from React Router for navigation • Individual exercise components/pages (/leg-press, /bulgarian-split-squat, /front-squat, /leg-extensions)

(2.8)

Class Name: Leg Press, Bulgarian Split Squat, Front Squat, Leg Extension, Bench Press, CableCrossover, Parallel Bar Dip, Push Ups, Plank Rotations, Hammer Curl, Bicep Curl, Tricep Dips, Overhead Tricep Extension, Row, Deadlift, Seated Cable Rows, Dumbbell Row, Arm Leg Raise	**Responsibilities:** Display Exercise Details: Show the name of the exercise prominently at the top of the page. Render an animated GIF of the exercise to guide the user. Exercise Timer: Integrate a timer widget (TimerWidget) to track the duration or rest time for the exercise.	**Collaborators:** TimerWidget: Used to track and display the time associated with the exercise. An animated GIF image imported to visually demonstrate the exercise form.
Parent Class: App **Subclasses:** None		

(2.9)

Class Name: Timer	**Responsibilities:**	**Collaborators:**
Parent Class: None **Subclasses:** None	• Display a circular progress bar that represents elapsed time • Allow users to play/pause the timer • Track and update the elapsed time based on settings • Handle settings for duration and switching modes • Uses intensity variable that is stored in local storage to generate the timer based on its intensity level (low: 1min, med: 2mins, high: 3 mins)	• CircularProgressbar from react-circular-progressbar to display a visual progress indicator • SettingsContext to retrieve timer settings • PlayButton and PauseButton components for controlling the timer • SettingButton for accessing additional settings

i)The description of system interaction with the environment: FitGenius is designed to run in any modern browser (Chrome, Firefox, Safari, Edge) with JavaScript enabled. It is built with React and relies on React Router for client-side routing and Vite for local development.

ii) Architecture of the system: The system architecture of FitGenius consists of a root component, App, which manages routing for the entire application. App is connected to Navbar for consistent navigation and to the main pages (Home, Recommended, LegPage, ArmPage, etc.), each of which serves as a hub for specific muscle groups. Each muscle group page then connects to individual exercise components, which provide content specific to each exercise.

iii) System Decomposition:
Component Roles: Each component in FitGenius plays a specific role within the hierarchy. App manages routing and acts as the central hub. Navbar provides consistent navigation links. Each muscle group page (e.g., LegPage, ArmPage) serves as a collection point for exercises related to that muscle group, with individual exercise components displaying specific content.

Error Handling: For navigation errors, a 404 page is set up to handle any undefined routes. If future features depend on external data sources, error handling will include user-friendly messages for network issues. In cases of invalid input (e.g., for future user interactions), basic validation will be implemented to guide users.