

Introduction to Cross-Functional Fitness

A Java-based application submitted for fall CSCI E-10b Graduate Credit

Author: Nag Rajendran

Date: Dec 11, 2023

Table of Contents

1. Introduction.....	3
2. Project Purpose and Justification	3
3. Target Audience	3
4. User Interaction and Design	4
4.1 Get Started: Create Your Profile and Embark on Your Fitness Journey	4
4.2 Launch Your Journey: Begin Your 30-Day Workout Program.....	6
4.3 Explore & Engage: Get to Know Your Workout Program Window.....	7
4.4 Track Your Progress: Mark Workouts as Completed and Record Time	8
4.5 Celebrate Your Achievements: Visit the Fitness Summary Screen	10
5. Features and Functionalities	11
5.1 User Onboarding	11
5.2 Daily Workout Plans.....	11
5.3 Detailed Workout Sessions.....	11
5.4 Progress Tracking.....	12
6 Algorithms and Data Structures	12
6.1 Class Descriptions and Interactions	12
6.1.1 CrossFunctionalFitness (Program Class).....	12
6.1.2 MainGUIFrame (Template Class).....	13
6.1.3 NewUserProfile (Template Class)	13
6.1.4 WorkoutPlanGenerator (Template Class)	13
6.1.5 FitnessSummary (Template Class)	14
6.1.6 FitnessMetricsCalculator (Utility Class).....	14
6.1.7 WorkoutData (Template Class).....	14
7 Sample Workout Plan	15
8 Flowcharts and Diagrams	17
9 Conclusion	17

1. Introduction

This proposal outlines the development of "Introduction to Cross-Functional Fitness" a Java-based application that serves as a dynamic and interactive guide to Olympic weightlifting and foundational weight training principles. Drawing from extensive expertise in running, weightlifting, and functional fitness, the application's objective is to make the essential knowledge of functional fitness accessible, engaging, and affordable. It will serve as an educational tool that translates complex fitness concepts into simple, fun learning experiences, fostering a holistic health and wellness approach.

2. Project Purpose and Justification

"Introduction to Cross-Functional Fitness" is designed to raise awareness and appreciation for functional fitness and foundational movements amidst the sedentary trends of contemporary desk jobs and the increasing shift to remote work. This program confronts and seeks to correct the widespread misconceptions of high-intensity functional training being exclusive to those already fit, instead promoting it as a universally accessible, engaging, and viable method for achieving health and vitality. Recognizing the detrimental impacts of sedentary lifestyles on health and dietary habits, the application provides a vital pathway to integrate exercise seamlessly into daily routines. It is a testament to the adaptability of functional training principles, offering a cost-effective solution for those without traditional gym access and a stimulating alternative for individuals looking to break free from the inertia of stationary work environments. By making functional fitness approachable and enjoyable, the program aspires to be more than an exercise platform—it aims to be a holistic catalyst for a healthier, active lifestyle for everyone.

3. Target Audience

The "Introduction to Cross-Functional Fitness" application is meticulously designed to cater to a wide spectrum of users, from complete novices to those with some experience in fitness who seek to enhance their understanding and application of functional training. It specifically benefits:

- **Beginners:** For individuals just starting their fitness journey, the app demystifies functional training with beginner-friendly exercises and progressive learning paths, ensuring a gentle introduction to fitness that builds confidence and foundational strength.
- **Remote Workers:** Employees who have transitioned to working from home and find themselves less active will discover routines that can be seamlessly integrated into their daily schedule, promoting an active lifestyle despite a sedentary work setting.
- **Busy Professionals:** For those who struggle to find time for gym visits, the app provides a flexible and time-efficient way to maintain fitness through short, effective workouts that can be performed in any setting.
- **Cost-Conscious Individuals:** By eliminating the need for expensive gym memberships and equipment, the app serves as an economical option for users who want to engage in a fitness regime without financial strain.

- **Intimidated by Gym Culture:** People who feel overwhelmed or uncomfortable in a gym environment will find a private, pressure-free space to exercise at their own pace, encouraging consistency and long-term commitment to health.

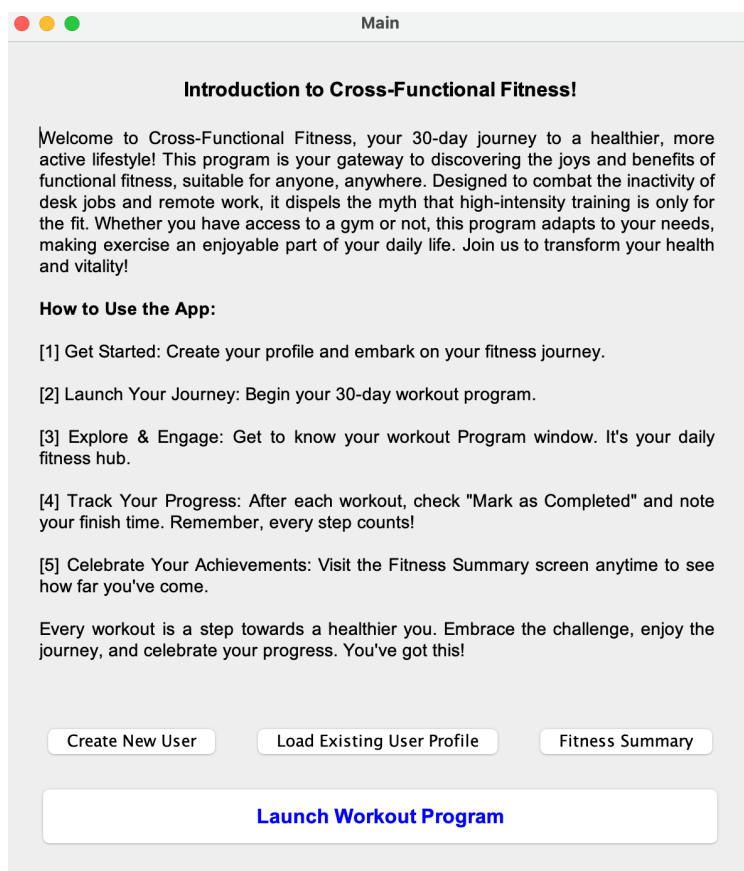
This application welcomes users from all walks of life and offers tailored benefits to each group, ensuring that everyone, regardless of their starting point, can embark on a fulfilling fitness journey.

4. User Interaction and Design

The "Introduction to Cross-Functional Fitness" application is designed with a focus on user engagement and ease of use. The application's workflow is structured around a series of interactive steps that guide users through their fitness journey.

4.1 Get Started: Create Your Profile and Embark on Your Fitness Journey

- **User Action:** Upon launching the application, users are prompted to create a new profile or load an existing one in the `NewUserProfile` class.



- **Design:** The profile creation interface is intuitive, requesting essential details like name, age, weight, and fitness level.

Profile Summary

First Name:

Last Name:

Age:

Weight (lbs):

☐ Beginner ☐ Novice

Fitness Level: ☐ Intermediate ☐ Advanced

☐ Expert

- **Process:** Once the profile is created and saved, users are directed to the main dashboard, symbolizing the start of their fitness journey.

Profile Summary

First Name:

Last Name:

Age:

Weight (lbs):

☒ Beginner ☐ Novice

Fitness Level: ☐ Intermediate ☐ Advanced

☐ Expert

Profile Summary

First Name:

Last Name:


Age:

Weight (lbs):

☐ Intermediate ☐ Advanced

☐ Expert

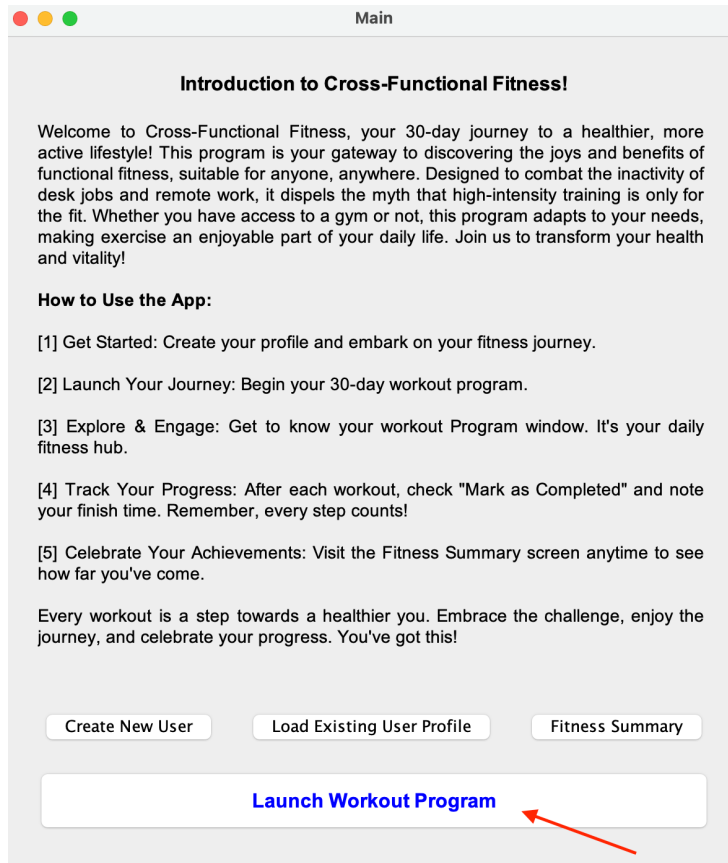
Success



Data saved successfully

4.2 Launch Your Journey: Begin Your 30-Day Workout Program

- **User Action:** From the main dashboard ([MainGUIFrame](#)), users can access the [WorkoutPlanGenerator](#) to view the 30-day workout plan.
- **Design:** The workout plan is presented in a clear, card-like format, with each day's workout accessible at the click of the “**Launch Workout Program**” button.



- **Process:** Selecting a day reveals detailed instructions for that day's workout, including exercises, repetitions, and durations.

30 Day Workout Plan

Day 1 Warm Up: 10 mins

Day 2 Workout of the Day (WOD):(Repeat 2 times)

Day 3 Exercise 1: 200m Jumping Jacks
Exercise 2: 40 air squats
Exercise 3: 200m Jumping Jacks
Exercise 4: 30 sit-ups
Exercise 5: 200m Jumping Jacks
Exercise 6: 20 jump squats
Exercise 7: 200m Jumping Jacks
Exercise 8: 10 burpees

Day 4

Day 5

Day 6

Day 7

Day 8 Post-workout stretching: 10 mins.

Day 9

Day 10

Day 11

Day 12

Day 13

Day 14

Day 15

Day 16

Day 17

Day 18

Day 19

Day 20

☐ Mark as complete Time taken (hours): 0 Minutes: 0 Save

4.3 Explore & Engage: Get to Know Your Workout Program Window

- **User Action:** Users interact daily with the [WorkoutPlanGenerator](#), which serves as the central hub for workout activities.
- **Design:** This window is designed to be informative and user-friendly, displaying workout details and providing navigational cues for moving between different days.

30 Day Workout Plan

Day 1

Day 2

Day 3

Day 4

Day 5

Day 6

Day 7

Day 8

Day 9

Day 10

Day 11

Day 12

Day 13

Day 14

Day 15

Day 16

Day 17

Day 18

Day 19

Day 20

Warm Up: 10 mins

EMOM (Every minute on the minute) 35 mins (Workout for 45 sec, 15 sec rest).

Exercise 1: Air Squat: 20 to 35 reps

Exercise 2: Push Up: 10 to 20 reps

Exercise 3: Reverse lunge: 20 reps (10 reps per side)

Exercise 4: Burpee: 10 to 20 reps

Exercise 5: Russian Twist: 10 to 20 reps

Exercise 6: Jump rope

Exercise 7: V-up: 10 to 20 reps

Stretching: 10 mins.

☐ Mark as complete Time taken (hours): 0 Minutes: 0

- **Process:** The design encourages exploration and engagement, inviting users to familiarize themselves with the variety of workouts and instructions.

4.4 Track Your Progress: Mark Workouts as Completed and Record Time

- **User Action:** Post-workout, users return to the [WorkoutPlanGenerator](#) to mark their workout as completed and record the time taken in the [WorkoutData](#).
- **Design:** Each day's panel includes a checkbox for marking completion and fields for entering the workout duration, fostering a sense of accomplishment and accountability.

30 Day Workout Plan

Day 1

Day 2

Day 3

Day 4

Day 5

Day 6

Day 7

Day 8

Day 9

Day 10

Day 11

Day 12

Day 13

Day 14

Day 15

Day 16

Day 17

Day 18

Day 19

Day 20

Warm Up: 10 mins

Workout of the Day (WOD):(Repeat 2 times)

Exercise 1: 200m Jumping Jacks
Exercise 2: 40 air squats
Exercise 3: 200m Jumping Jacks
Exercise 4: 30 sit-ups
Exercise 5: 200m Jumping Jacks
Exercise 6: 20 jump squats
Exercise 7: 200m Jumping Jacks
Exercise 8: 10 burpees

Post-workout stretching: 10 mins.

☒ Mark as complete Time taken (hours): Minutes:

After clicking save,

30 Day Workout Plan

Day 1

Day 2

Day 3

Day 4

Day 5

Day 6

Day 7

Day 8

Day 9

Day 10

Day 11

Day 12

Day 13

Day 14

Day 15

Day 16

Day 17

Day 18

Day 19

Day 20


Warm Up: 10 mins

Workout of the Day (WOD):(Repeat 2 times)

Exercise 1: 200m Jumping Jacks
Exercise 2: 40 air squats
Exercise 3: 200m Jumping Jacks
Exercise 4: 30 sit-ups
Exercise 5: 200m Jumping Jacks
Exercise 6: 20 jump squats
Exercise 7: 200m Jumping Jacks
Exercise 8: 10 burpees

☒ Mark as complete Time taken (hours): Minutes:

Message

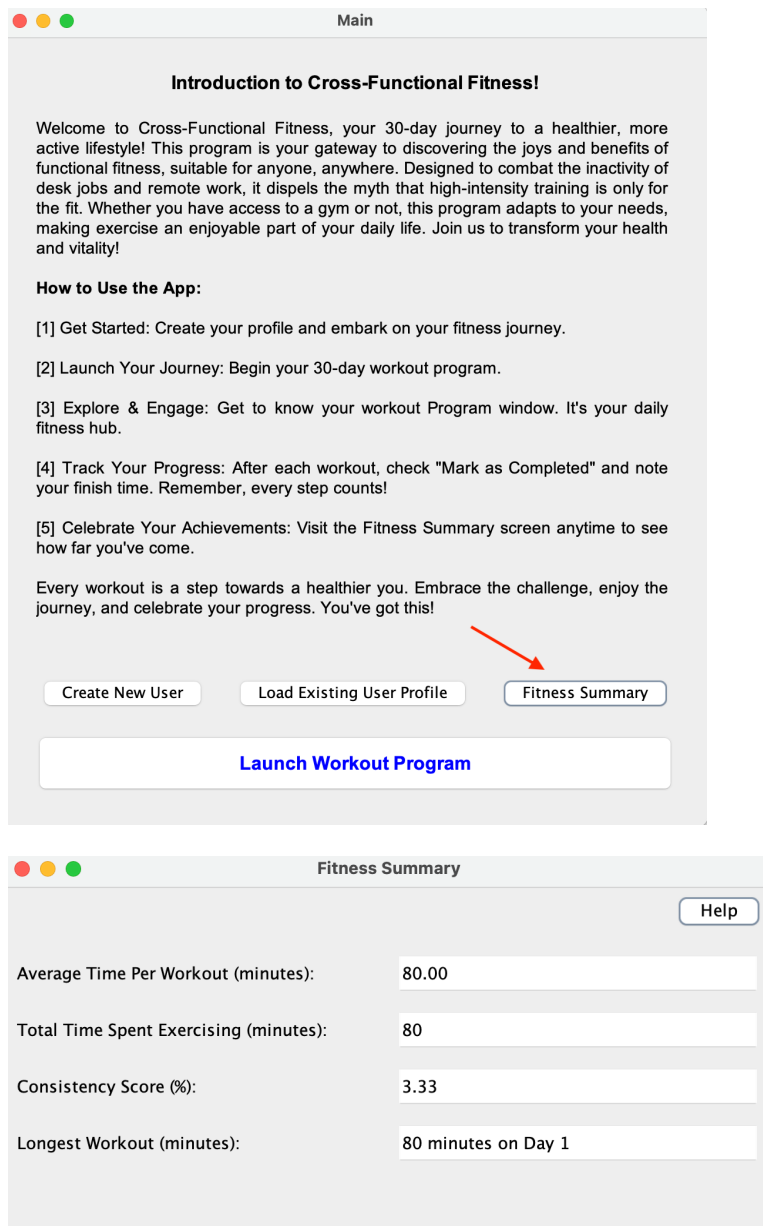


Awesome job! You've completed the workout for Day 1. Keep up the great work!

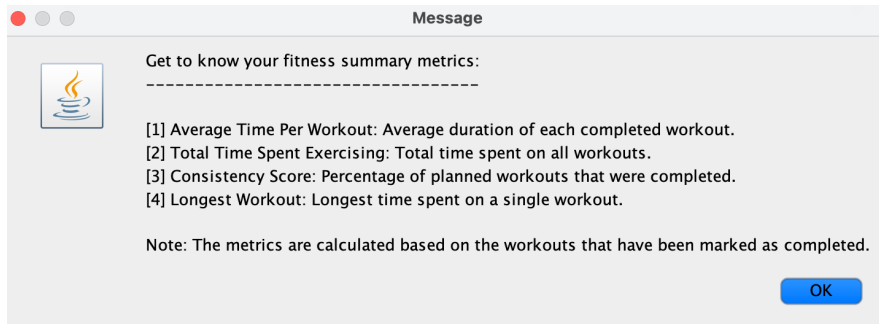
- **Process:** Tracking these details not only motivates users but also feeds into the calculation of fitness metrics, making this step essential for progress monitoring.

4.5 Celebrate Your Achievements: Visit the Fitness Summary Screen

- **User Action:** At any point, users can view their progress in the [FitnessSummary](#).
- **Design:** The summary screen is designed to celebrate achievements, displaying metrics such as average workout time, total time spent, and consistency scores.



The help button will show the brief definition of the metrics presented for clarity.



- **Process:** This reflective aspect of the application underscores the user's journey, highlighting milestones and improvements.

5. Features and Functionalities

5.1 User Onboarding

- **Purpose:** The application begins with an onboarding process where users provide information about their fitness levels and basic wellness details.
- **Implementation:** This is managed through the `NewUserProfile` class, where users input their fitness details.
- **Functionality:** The user inputs their name, age, weight, and fitness level, which are saved to create a personalized profile. This information is not currently used to customize the workout, but it sets the stage for future enhancements to this application.

5.2 Daily Workout Plans

- **Purpose:** The application generates daily workout schedules.
- **Implementation:** The `WorkoutPlanGenerator` class constructs a 30-day workout plan
- **Functionality:** Each day in the workout plan is structured, ensuring a gradual increase in difficulty.

5.3 Detailed Workout Sessions

- **Purpose:** The application enhances user engagement through a detailed description of workout sessions.
- **Implementation:** Within the `WorkoutPlanGenerator`, each workout session is presented with detailed instructions and options for interaction to save the progress.
- **Functionality:** Users receive step-by-step guidance for each day's workout.

- **User Interaction:** The application, though not originally scoped for a live timer or pause/restart functionality, could potentially integrate these features in future iterations for a more interactive experience.

5.4 Progress Tracking

- **Purpose:** Users can track their progress throughout the fitness program, marking workouts as completed and noting the time taken.
- **Implementation:** Utilizing the `WorkoutData` class, the application allows users to record the completion status and duration of each workout.
- **Functionality:** Workout Completion: After finishing a workout, users mark it as complete and record the time taken.
- **Fitness Metrics Visualization:** In the `FitnessSummary` class, users can view their performance metrics such as average workout time, total time spent exercising, and consistency scores. While specific caloric burn tracking isn't a feature, this could be a valuable addition for future development.

6 Algorithms and Data Structures

The "Introduction to Cross-Functional Fitness" application utilizes a structured approach with clearly defined classes, each serving specific roles. This section outlines the purpose of each class, its interaction with other components, the distinction between template and program classes, and a brief explanation of their methods.

6.1 Class Descriptions and Interactions

6.1.1 CrossFunctionalFitness (Program Class)

- Purpose: Serves as the entry point of the application.
- Interaction: Instantiates `MainGUIFrame`.
- Key Methods:
 - o `main(String[] args)`: Initializes the application by creating the `MainGUIFrame`.
 - o `welcomeForm()`: Sets up the initial form users see upon opening the application.
 - o `newUserForm()`: This method is responsible for initializing and displaying the form for creating a new user profile.

- o `loadUserForm()`: This method is designed to handle the loading of an existing user's profile.

6.1.2 MainGUIFrame (Template Class)

- Purpose: Manages the main user interface.
- Interaction: Links to `NewUserProfile`, `FitnessSummary`, `WorkoutPlanGenerator`.
- Key Methods:
 - o `layoutComponents()`: Sets up the UI layout with buttons and panels.
 - o `addListeners()`: Adds action listeners to buttons for navigation.

6.1.3 NewUserProfile (Template Class)

- Purpose: Handles user profile creation and loading.
- Interaction: Saves and retrieves user data and interacts with the file system for persistence.
- Key Methods:
 - o `saveUserProfile()`: Saves user data to a file.
 - o `loadUserData()`: Loads user data from a file.
 - o `validateUserInputs()`: Validates the inputs in the profile form.

6.1.4 WorkoutPlanGenerator (Template Class)

- Purpose: Displays the workout plan and details.
- Interaction: Uses `WorkoutData` to load and save workout details.
- Key Methods:
 - o `addWorkoutDaysAndDetails()`: Dynamically creates workout day buttons and detail panels.
 - o `createWorkoutDetailsPanel(int, String)`: Generates the detail panel for each workout day.
 - o `loadWorkoutDetails(int)`: Loads workout details for a specific day.

6.1.5 FitnessSummary (Template Class)

- Purpose: Displays fitness metrics and summaries.
- Interaction: Retrieves data from [FitnessMetricsCalculator](#).
- Key Methods:
 - [updateMetrics\(String, String, String, String\)](#): Updates the UI with fitness metrics.
 - [addListeners\(\)](#): Adds listeners to UI components, like the help button.

6.1.6 FitnessMetricsCalculator (Utility Class)

- Purpose: Calculates various fitness-related metrics.
- Interaction: Processes data from [WorkoutData](#).
- Key Methods:
 - [calculateAverageTime\(List<WorkoutData>\)](#): Calculates the average workout time for completed workouts.
 - [calculateTotalTime\(List<WorkoutData>\)](#): Computes the total time spent on workouts.
 - [calculateConsistencyScore\(List<WorkoutData>, int\)](#): Determines the consistency score.
 - [calculatePersonalBests\(List<WorkoutData>\)](#): Identifies the longest workout duration.

6.1.7 WorkoutData (Template Class)

- Purpose: Manages individual workout data.
- Interaction: Used by [WorkoutPlanGenerator](#) for data persistence.
- Key Methods:
 - [saveWorkout\(WorkoutData, int\)](#): Saves workout data to a file.
 - [loadWorkouts\(int\)](#): Loads workout data from a file.

Explanation of method usage,

- o `CrossFunctionalFitness.main(String[] args)`: Launches the application by creating an instance of `MainGUIFrame`.
- o `MainGUIFrame.layoutComponents()`: Sets up the GUI layout, including buttons like "Create New User", and adds them to the frame.
- o `NewUserProfile.saveUserProfile()`: Gathers user inputs, validates them and saves them to a file.
- o `WorkoutPlanGenerator.addWorkoutDaysAndDetails()`: Dynamically generates buttons for each day of the workout program and corresponding detail panels.
- o `FitnessSummary.updateMetrics(...)`: Receives calculated metrics and updates the text fields in the `FitnessSummary` UI.
- o `FitnessMetricsCalculator.calculateAverageTime(...)`: Iterates through a list of `WorkoutData`, summing up the time taken for each workout and calculating the average.
- o `WorkoutData.saveWorkout(...)`: Converts workout data into a string format and writes it to a specified file for the corresponding day.

Each class in the application has a specific role, contributing to the overall functionality of the fitness program. The clear distinction between template classes (handling UI and user interaction) and program/utility classes (performing calculations and data handling) ensures a modular and maintainable code structure. The methods within these classes are designed to interact seamlessly, facilitating a smooth user experience and efficient data management.

7 Sample Workout Plan

- **Day 1**

Warm Up: 10 mins

Workout of the Day (WOD):(Repeat 2 times)

- Exercise 1: 200m Jumping Jacks
- Exercise 2: 40 air squats
- Exercise 3: 200m Jumping Jacks
- Exercise 4: 30 sit-ups
- Exercise 5: 200m Jumping Jacks
- Exercise 6: 20 jump squats
- Exercise 7: 200m Jumping Jacks
- Exercise 8: 10 burpees

Post-workout stretching: 10 mins.

- **Day 2**

Warm Up: 10 mins

EMOM (Every minute on the minute) 35 mins (Workout for 45 sec, 15 sec rest).

Exercise 1: Air Squat: 20 to 35 reps

Exercise 2: Push Up: 10 to 20 reps

Exercise 3: Reverse lunge: 20 reps (10 reps per side)

Exercise 4: Burpee: 10 to 20 reps

Exercise 5: Russian Twist: 10 to 20 reps

Exercise 6: Jump rope

Exercise 7: V-up: 10 to 20 reps

Stretching: 10 mins.

.....

- **Day 30**

Warm Up: 10 Min Yoga

Benchmark Test:

1 mile Run

100 Pushups

200 Air Squat

300 Sit-Ups

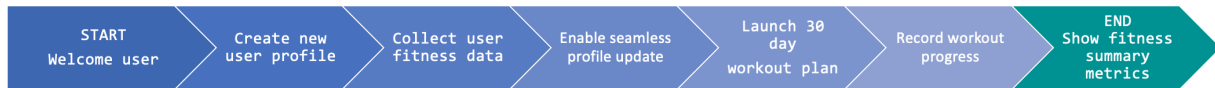
1 mile run

Stretching: 10 mins

8 Flowcharts and Diagrams

Flowcharts detailing the user onboarding process, the daily workout routine, and the data management system are as follows.

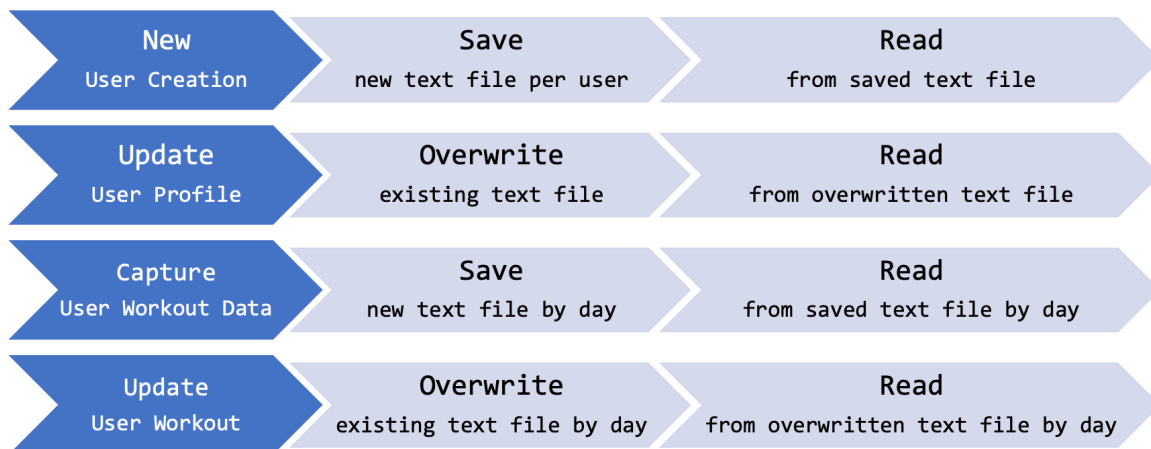
- **User onboarding process**



- **Daily workout routine**



- **Data management**



9 Conclusion

"Introduction to Cross-Functional Fitness" is poised to revolutionize home-based fitness by making high-intensity functional training accessible and adaptable. This application is more than just a tool; it's a catalyst for change designed to foster robust functional fitness and core strength. It aims to create a ripple effect that will improve individual health and well-being and challenge and reshape the landscape of home fitness routines.