

FITNESS TRACKING SYSTEM

Software Design Document

Purpose

The purpose of this design document is to outline the design, architecture, and interactions between components of the Fitness Tracking System. This document serves as a blueprint for the development team, ensuring a clear understanding of the system's structure and operation. It is intended for use by developers, architects, and stakeholders involved in the development and maintenance of the system.

Scope

The Fitness Tracking System is designed to allow users to log their workout details, which are processed and stored in two databases: one for workouts and one for reports. The system utilizes Kafka for message processing and includes the following subsystems:

Workout Service: Handles user inputs and stores workout details.

Report Service: Processes messages from Kafka and stores reports.

System Architecture

The system architecture includes the following major components:

Workout Service: A Play Framework application that provides RESTful APIs for logging and retrieving workout data.

Report Service: An Akka-based service that consumes messages from Kafka and processes them to generate reports.

Kafka: Serves as the message broker between Workout Service and Report Service.

MySQL Databases: Two separate databases for storing workout and report data.

Technological Stack

Scala: Primary programming language

SBT: Build tool

Play Framework: Web framework for the Workout Service

Akka: Framework for the Report Service

Docker: Containerization platform

Kafka: Message broker

MySQL: Database management system

GCP: Hosting platform

3. Design Considerations

Assumptions and Dependencies

The system assumes a reliable network connection between services.

Dependencies include Kafka, MySQL, Docker, and GCP services.

The system assumes the availability of GCP infrastructure for deployment.

Goals and Guidelines

FITNESS TRACKING SYSTEM

Performance: Ensure low latency in message processing and data retrieval.

Scalability: Design the system to handle increasing loads by adding more instances.

Compatibility: Ensure compatibility with standard RESTful API practices and data formats like JSON.

User Experience: Provide a simple and intuitive UI for logging and viewing workouts.

Component Descriptions

Workout Service

Purpose: To log workout details and provide APIs for retrieving logged data.

Interface: RESTful APIs for logging (logWorkout) and retrieving (listWorkouts) workouts.

Functionalities:

Accept workout details from users.

Send workout data to Kafka.

Retrieve and display logged workouts.

Performance Characteristics: Should handle concurrent requests efficiently with low latency.

Report Service

Purpose: To process workout data from Kafka and generate reports.

Interface: Consumes messages from Kafka and updates the report database.

Functionalities:

Consume messages from Kafka.

Process and store report data.

Performance Characteristics: Should process messages in near real-time.

Data Architecture

The data storage solutions include two MySQL databases: one for storing workout data and one for storing report data.

Database Design

Workout Database Schema

workouts table: Stores details of each workout.

Columns: id, user_id, type, duration, calories_burned, timestamp

Report Database Schema

reports table: Stores aggregated report data.

Columns: id, user_id, total_workouts, total_duration, total_calories, timestamp

FITNESS TRACKING SYSTEM

Data Dictionary

workouts table:

id (INT): Primary key

user_id (INT): ID of the user

type (VARCHAR): Type of workout

duration (INT): Duration in minutes

calories_burned (INT): Calories burned during the workout

timestamp (TIMESTAMP): Time of the workout

reports table:

id (INT): Primary key

user_id (INT): ID of the user

total_workouts (INT): Total number of workouts

total_duration (INT): Total duration of workouts in minutes

total_calories (INT): Total calories burned

timestamp (TIMESTAMP): Time of the report

User Interfaces

Log Workout Page:

Allows users to input and submit workout details.

View Workouts Page: Displays a list of logged workouts.

User Workflows

Log Workout:

User navigates to the Log Workout page.

User enters workout details and submits the form.

Workout details are sent to the Workout Service API.

View Workouts:

User navigates to the View Workouts page.

The page retrieves and displays the user's logged workouts from the Workout Service.

Accessibility Features

Ensure the UI is accessible to users with disabilities by following WCAG guidelines.

Implement keyboard navigation and screen reader support.

FITNESS TRACKING SYSTEM

Security Requirements

Protect user data during transmission and storage.

Implement authentication and authorization mechanisms.

Encryption & Data Protection

Use HTTPS for secure data transmission.

Encrypt sensitive data in the database using MySQL's encryption functions.

Deployment Architecture

Use Docker containers for all services.

Deploy containers on GCP

This software design document provides a comprehensive overview of the Fitness Tracking System's architecture and design. It ensures that all team members and stakeholders have a clear understanding of the system's structure and operation.