# Design document

Content

[Design document 1](#_Toc179474902)

[**1.** **Introduction** 2](#_Toc179474903)

[**2.** **Architectural Overview** 2](#_Toc179474904)

[2.1 System Architecture 2](#_Toc179474905)

[2.2 Technologies Used: 2](#_Toc179474906)

[**3.** **Architecture Constraints and Design Decisions** 2](#_Toc179474907)

[3.1 Choice of Spring Boot (Backend) 2](#_Toc179474908)

[3.2 Choice of React (Frontend) 2](#_Toc179474909)

[3.3 Choice of MySQL (Database) 2](#_Toc179474910)

[**4.** **C4 Model Diagrams** 3](#_Toc179474911)

[4.1 Level 1: System Context Diagram 3](#_Toc179474912)

[4.2 Level 2: Container Diagram 3](#_Toc179474913)

[4.3 Level 3: Component Diagram 4](#_Toc179474914)

[4.4 Level 4: Code Diagram 5](#_Toc179474915)

## **Introduction**

This document provides an overview of the architectural design and technical decisions for the fitness application. The project aims to deliver a personalized fitness and diet planning platform for users and trainers, with features like workout and diet suggestions, progress tracking, and trainer management.

## **Architectural Overview**

### 2.1 System Architecture

The system follows a typical client-server architecture, with the front end developed using React and the backend built with Spring Boot. MySQL is used as a relational database to store user, trainer, and workout data. The system is designed to ensure scalability, security, and ease of maintenance.

### 2.2 Technologies Used:

* Frontend: React (JavaScript)
* Backend: Spring Boot (Java)
* Database: MySQL
* Version Control: Git
* CI/CD: GitLab CI for Continuous Integration
* Containerization: Docker

# **Architecture Constraints and Design Decisions**

### 3.1 Choice of Spring Boot (Backend)

Reason: Spring Boot was chosen for its simplicity in creating production-ready applications with minimal configuration. It provides built-in support for building RESTful APIs, which are critical for communicating between the frontend and backend.

### 3.2 Choice of React (Frontend)

Reason: React’s component-based architecture allows for easier state management and scalable UI development. It works seamlessly with REST APIs, making it an ideal choice for a dynamic web application.

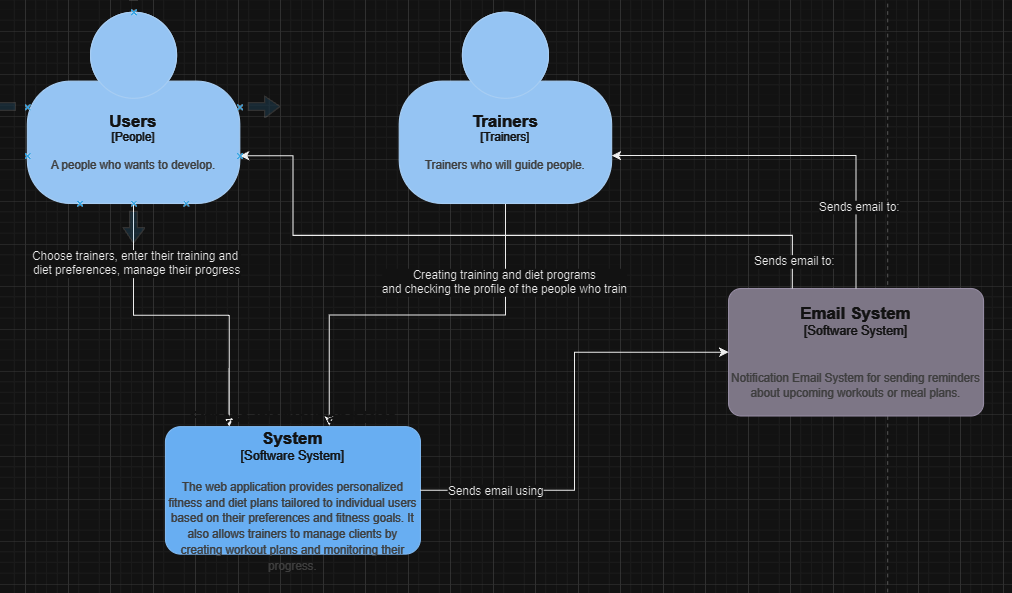
### 3.3 Choice of MySQL (Database)

Reason: MySQL is a widely-used relational database, offering robust ACID properties and easy integration with Spring Boot’s JPA for data management.

## **C4 Model Diagrams**

### 4.1 Level 1: System Context Diagram

This diagram shows how external actors (users and trainers) interact with the system. Users access the application via a web browser, while trainers manage client profiles via the same interface. All communications go through the API gateway hosted in Spring Boot.



### 4.2 Level 2: Container Diagram

This diagram breaks down the system into its main containers:

* Frontend (React): User interface where users and trainers interact with the system.
* Backend (Spring Boot): API layer that handles business logic and communicates with the database.
* Database (MySQL): Stores user, trainer, and workout information.

Картина, която съдържа текст, екранна снимка, диаграма, Шрифт

Описанието е генерирано автоматично

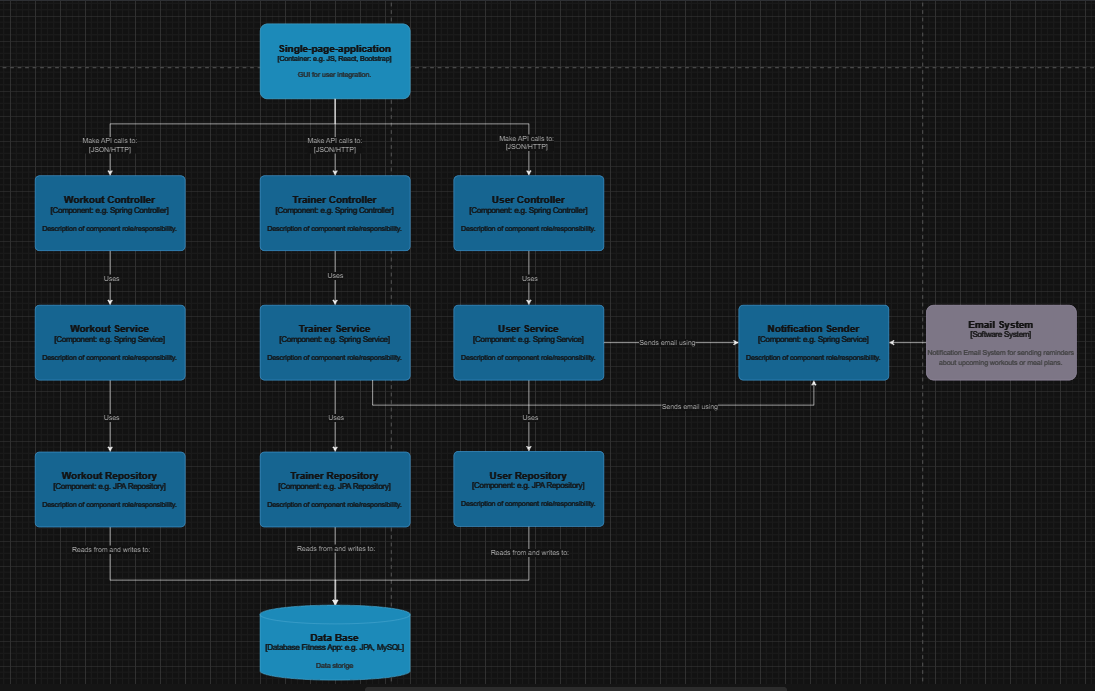
### 4.3 Level 3: Component Diagram

The components within the backend are divided into Controllers, Services, and Repositories:

**Controller Layer:** Handles incoming HTTP requests (e.g., for user registration, and workout plan creation).

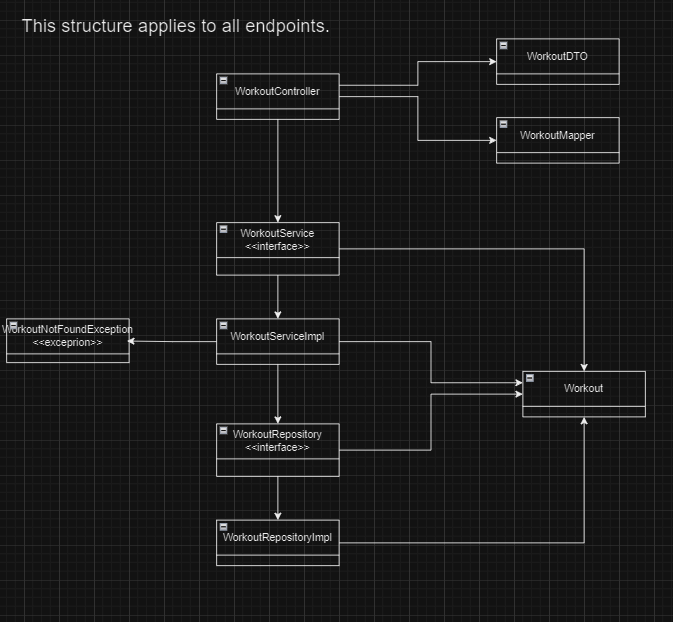
**Service Layer:** Implements the business logic, such as generating personalized workout plans.

**Repository Layer:** Interfaces with the MySQL database for CRUD operations.



### 4.4 Level 4: Code Diagram

The Code Diagram provides a detailed view of the internal structure of individual classes and their methods, showcasing the implementation details of the components described at Level 3. This level focuses on the interactions between classes, methods, and logic within specific layers of the application, offering a precise breakdown of how the system functions at the code level.



This structure was created for training classes but is the same for each one.

## Sonargube

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated