Technical Design

Name: Tony Jiang

Semester: 6

Project: Music trivia game

|  |  |  |
| --- | --- | --- |
| Version | Date | Notes |
| 0.1 | 18 March 24 | Initial document. |
| 0.2 | 12 April 24 | Add C4 UML. |
| 0.3 | 19 April 24 | Add CI pipeline diagram. |
| 0.4 | 05 May 24 | Add yml file. |

Contents

[System Architecture 3](#_Toc165819838)

[Level 1: System Context 3](#_Toc165819839)

[Level 2: Containers 3](#_Toc165819840)

[Level 3: Component 4](#_Toc165819841)

[Level 4: Code 6](#_Toc165819842)

[CI/CD pipeline 6](#_Toc165819843)

[YML file 7](#_Toc165819844)

# Introduction

This is the technical design document for the music trivia web-based game. It includes all the technical details on how the project is implemented and its structure. This document also reflects on the design choices made within the project, helping to plan the configuration and address potential development challenges. Additionally, it aids in conveying the intended design to other developers, ensuring a shared understanding and agreement on the design approach.

# System Architecture

The purpose of System Architecture is to describe the internal system’s overall structure and establish an agreement on the desired design of the system. To make it easy to describe and communicate the system’s architecture, we’ll use a C4 architecture diagram. It’s an architecture design that is easy to understand. The design approach is straightforward and helps us communicate how each part of the system should be set up, even to a non-technical person.

The C4 architecture diagram has 4 level:

• Level 1: System Context (C1)

• Level 2: Containers (C2)

• Level 3: Component (C3)

• Level 4: Code (C4)

We will go to each level and describe what they do when we reach there. The tool that was used to create the C4 model is [Visual Paradigm online](https://online.visual-paradigm.com/diagrams/features/c4-model-tool/). The free version.

## Level 1: System Context

A screenshot of a computer

Description automatically generated

In this C1 model, you can see that the player uses the music trivia system to play the game. The admin can also perform CRUD functions for the songs in the system. This model provides an understanding of which users can interact with the system, what type of system it is, and the actions they can perform.

## Level 2: Containers

A computer screen shot of a diagram

Description automatically generated

t’s somewhat understandable how the system is put together from the C2 model. The system consists of 10 containers, including:

**1 container for frontend.**

* **Description:** This component serves as the user interface, where users interact with the system. It communicates with the backend to process users' actions.
* **Technology:** React.
* It is a popular JavaScript library for building user interfaces
* It has for single-page applications.
* It allows developers to create large web applications that can render efficiently in response to data changes.

**5 containers for backend micro-service.**

* **Description:** These containers are the backend micro-service, each having the respective service/ functions of the system for maintaining and handle the project without any problems in individual setting and in group setting.
* **Technology:** Java and Spring boot

**5 containers for databases for each micro-service.**

## Level 3: Component

A diagram of a company

Description automatically generated

## Level 4: Code

A diagram of a computer

Description automatically generated with medium confidence

# CI/CD pipeline

A diagram of a flowchart

Description automatically generated

## YML file

Yml file on GitHub action.

name: MusicTrivia

on:

push:

branches:

- development

jobs:

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- uses: actions/setup-java@v4

with:

distribution: 'temurin'

java-version: '21'

cache: 'gradle'

- name: Build with Gradle

run: |

cd MusicTrivia

chmod +x gradlew

./gradlew assemble --no-daemon

test:

runs-on: ubuntu-latest

needs: build

steps:

- uses: actions/checkout@v4

- uses: actions/setup-java@v4

with:

distribution: 'temurin'

java-version: '21'

cache: 'gradle'

- name: Test with Gradle

run: |

cd MusicTrivia

chmod +x gradlew

./gradlew test

docker:

runs-on: ubuntu-latest

needs: test

steps:

- uses: actions/checkout@v4

- uses: actions/setup-java@v4

with:

distribution: 'temurin'

java-version: '21'

cache: 'gradle'

- name: Build with Gradle

run: |

cd MusicTrivia

chmod +x gradlew

./gradlew assemble --no-daemon

- name: Set up Docker Buildx

uses: docker/setup-buildx-action@v3

- name: Login to Docker Hub

uses: docker/login-action@v3

with:

username: ${{ secrets.DOCKER\_USERNAME }}

password: ${{ secrets.DOCKER\_PASSWORD }}

- name: Build and push Docker image

run: |

cd MusicTrivia

docker build -t tonyj3/music-trivia-backend:latest .

docker push tonyj3/music-trivia-backend:latest