

Listen Up

Design Document

Mohammad Nazibul Kabir Khan

4263308

Content

How SOLID is guaranteed

- Single responsibility
- Open/Closed principle
- Liskov substitution
- Interface segregation
- Dependency inversion

Important design decision

- Why user spring boot?
- Front end library for building user interface
- Database to use

C4 Architecture

- C1 Architecture
- C2 Architecture
- C3 Architecture
- C4 Architecture/UML Diagram

CI Setup Diagram

How is SOLID guaranteed

Single responsibility

Every class has their own special responsibility so that there is not more than one reason for one class to change.

Open/closed principle

Each management classes are designed such a way it will be closed for modification, but it is open for extension for later phase.

Liskov substitution

Some classes are using reference from their base classes. For example, user of the application is divided by two types: Admin and Customer. Same for song: Single and Album Song.

Interface segregation

At the moment this principle is not used in the application.

Dependency inversion

Business classes is connected with their database classes through abstraction. It means business classes can be depended on different database. The presentation layer is connected to the business layer through another abstraction. Thus, it allows presentation layer to use multiple business layer.

Important Design Decision

Why user spring boot

Spring boot is chosen because it is easy to create stand-alone spring-based application with it. User just have to run it. Dependencies that a project need can also be added to it.

Front end library for building user interfaces

React is used because more can be learned from it. Bigger companies are using react therefore there is no reason to chose otherwise. React also makes coding simpler which will reduce the amount of code developers have to type.

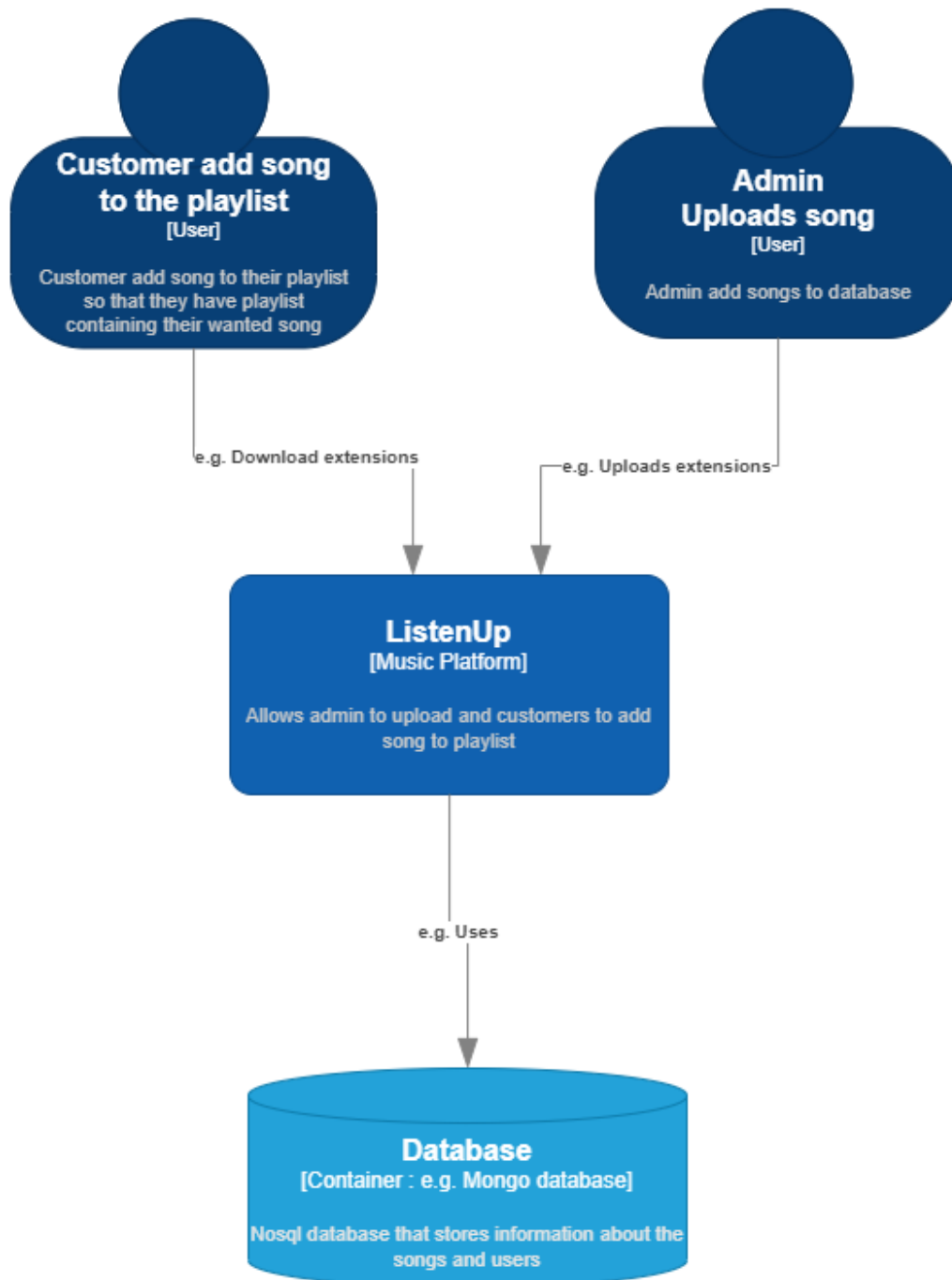
Database to use

MySQL is used because it is simpler to use. Developers have more control over the data needed to be stored.

C4 Architecture

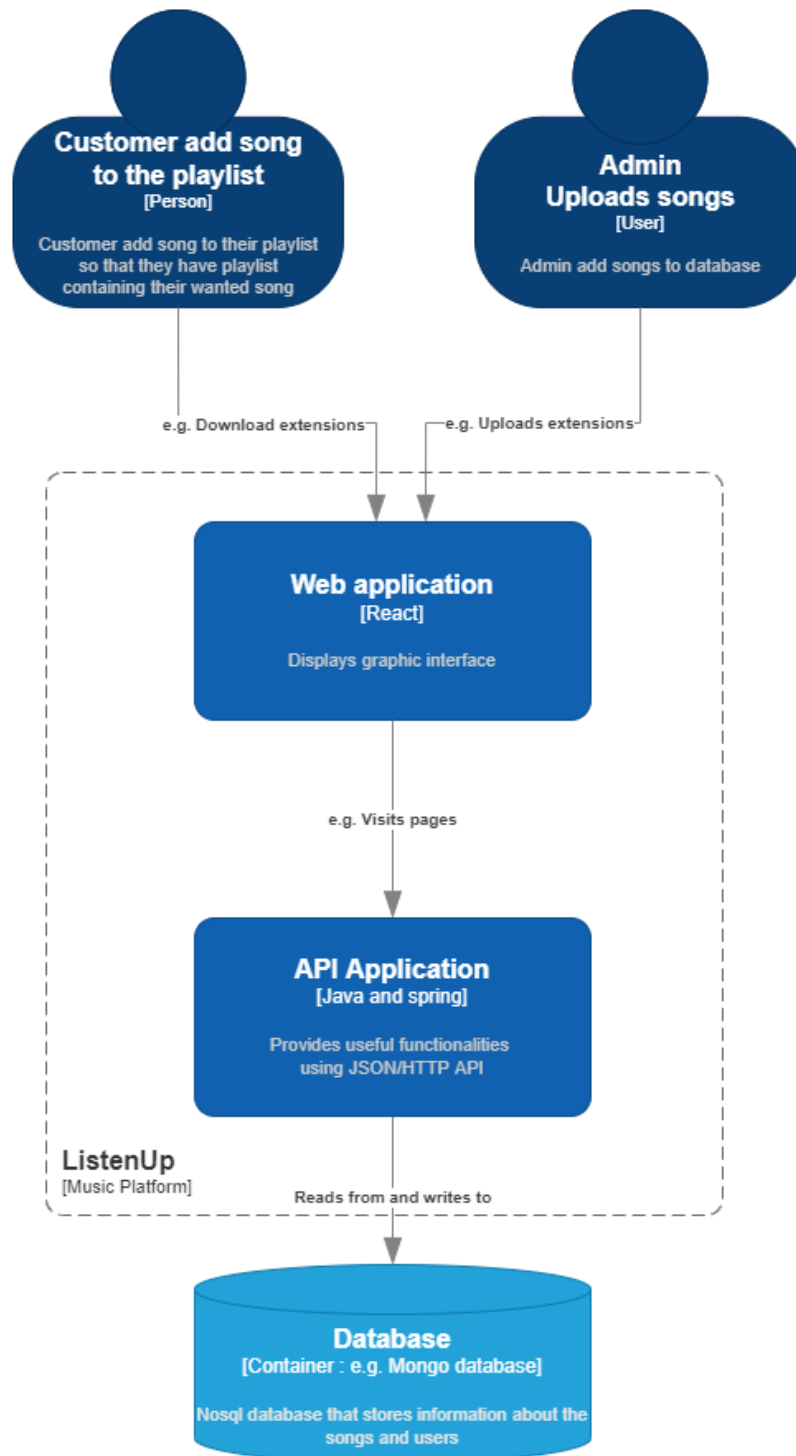
C1

C1 is the first stage of architecture. It shows only the simple structure how two different users: Customer and admin can have two different function using same platform.



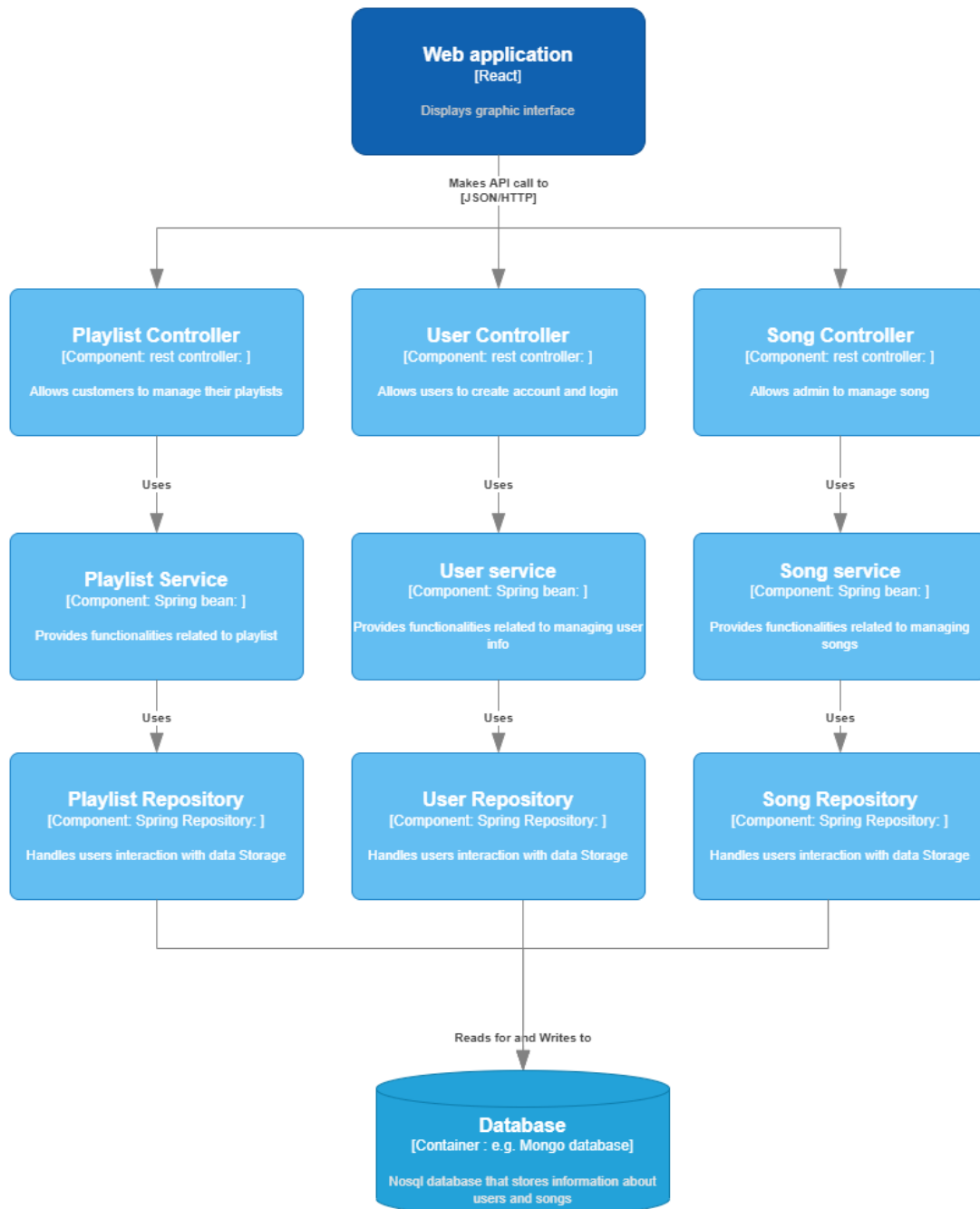
C2

This is the next stage of architect which focus on what goes on in the platform. Here it shows how front-end is connected to back-end. Web application which is a user interface is connected to API which provides all the functionalities.



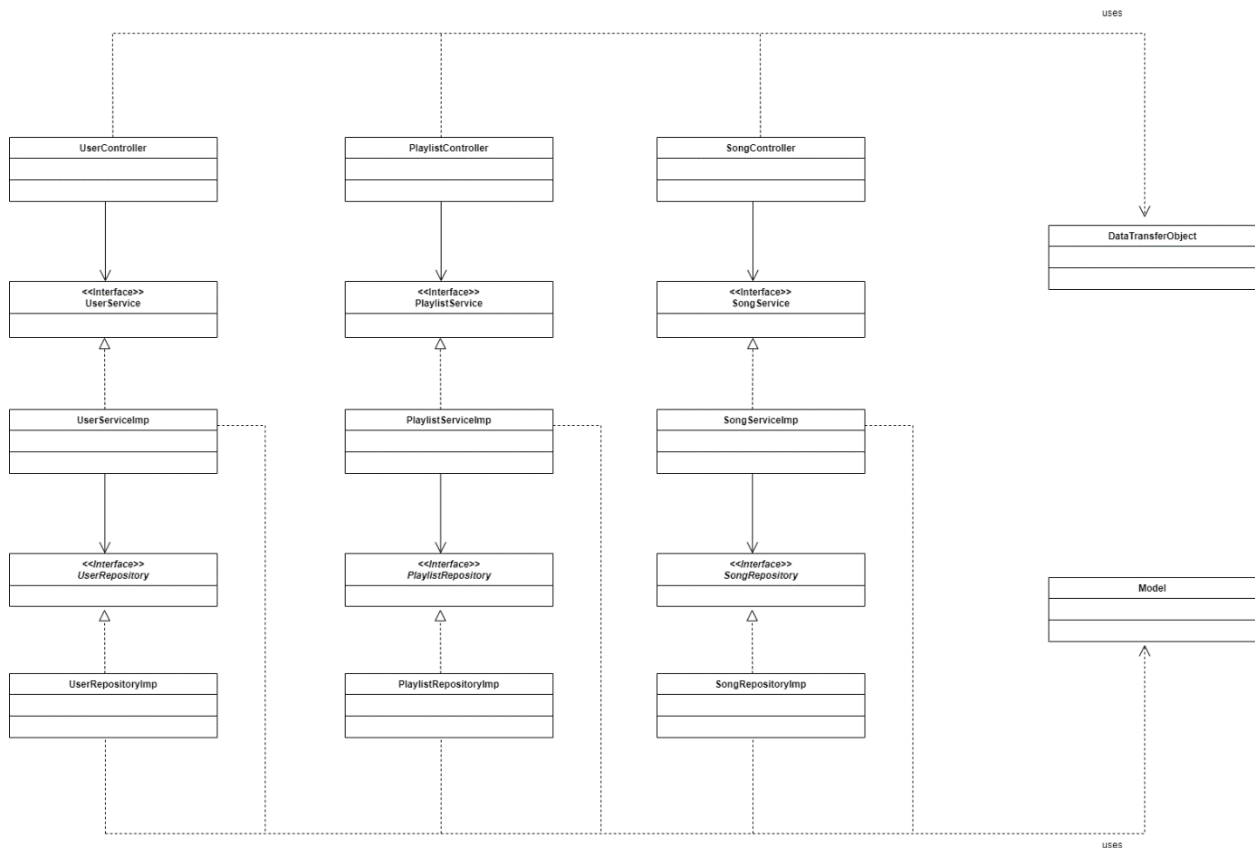
C3

This is the third stage of the architecture where it focuses more on the back-end part. Back-end is divided in three layers: Controller, service, and repository. Only repository layer has connection with the database. It controls the data. Whereas service layer controls the logic part of the software. Controller layer has the REST component which supplies front-end their data.



C4

This is the final stage which zooms in more into the back-end. It shows how each layer as previously mentioned is connected to each other through interface. It ensures if the structure follows the SOLID principle. There is multiple service layer with follows the single responsibility principle. Classes are also designed in such a way that it is open for extension but closed for modification. Thus, following the open/closed principle. Each layer is connected with another layer through interface which follows dependency inversion.



CI Setup

