**Listen Up**

Design Document

Mohammad Nazibul Kabir Khan

4263308

**How is SOLID guaranteed**

**Single responsibility**

Every class has their own special responsibility so that there in 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**

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

**Front end library for building user interfaces**

I am using react because I would like to learn more about 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 I have to type.

**Database to use**

I have not decided which database I am going to use. I think MongoDB are used by bigger companies. But I am more familiar with MySQL. I am going to decide on later phase.

**C4 Architecture**

C1

Diagram

Description automatically generated

C2

Diagram

Description automatically generated

C3

Diagram

Description automatically generated

C4

Diagram

Description automatically generated

UML Class Diagram

Diagram, schematic

Description automatically generated