Web application
Architecture Level

Version: 1.0

Version Date: 25-03-2021

Student Number: 3870707

Table of Contents

Introduction

To produce a high-level architecture web application, I have applied some of the SOLID principles.

- **S** - Single-responsibility Principle
- **O** - Open-closed Principle
- **L** - Liskov Substitution Principle
- **I** - Interface Segregation Principle
- **D** - Dependency Inversion Principle

SOLID Applications

Single Responsibility:

Currently I have applied the single responsibility principle by having a class that manages its individual data source (ProfileFakeDataSource, GraduatingYeaFakeDataSource) and having a service class which then manages the FakDataSourceClasses

(ProfileServiceClass, GraduatingYearServiceClass), with the introduction of new classes during the next sprints this principle will carry on to ensure one class perform/manages only one task.

Liskov Substitution principle:

Currently in my source code I have applied the Liskov substitution principle with my use of Inheritance and have also applied polymorphism in managing profiles which I have (Teachers, Yearbookcommittee Student).

Dependency Inversion Principle

I applied dependency inversion with the use of dependency injection and the application of injecting an interface rather than injecting the Fakedatasource directly in the service classes which enables easy switching between the data sources.

Outside principles applied

Apart from the SOLID principles in my application I introduced the use of the three-layered principle by working with a Logic/Service layer, Database layer and Viewing Layer which enables easy navigation between my codes/classes.

Backend framework Choice:

For my application, I am using spring boot to handle all back end operations due it reducing the overall development time and increase efficiency by having a default setup for unit testing via the use of continuous integration.

Also due to spring boot dependency injection it makes my application code easier to test, manage and extend.

Finally, with Java spring boot framework managing my back end, my front end can be properly managed by a JavaScript framework.