**ASSIGNMENT 1-REPORT**

**Submitted by: Dikchha Rijal, Student number: 040806753**

This project consists of several classes and interfaces, that work together to process, save, and write the data of employees to files with different formats. To adhere to the SOLID principles, five new classes and two new interfaces were created. The purpose and the alignment with the SOLID principles of the newly created classes and interfaces are outlined below:

**TextFormatter.java:** Purpose- Using a string builder, this class implements the ‘Formatter’ interface and saves the ‘Employee’ object as key-value pairs. This class was created to format employee data in text format.

**PersistenceService.java:** Purpose- This class provides functionality for saving ‘Employee’ data to a file. This class allows employee data to be formatted using different strategies and then written to a specified file.

**PermanentEmployeeImpl.java:** Purpose- This class represents only permanent employees with additional attributes like bonus and pension. It has respective methods for bonus and pension calculations.

**ContractEmployeeImpl.java**: Purpose- This class represents only contract employees with an attribute for the renewal date. It has a method for date renewal.

**EmployeeValidator.java**: Purpose- This class provides validation functionality for ‘Employee’ objects. This class contains methods to ensure that the attributes of the employee meet specific criteria ensuring the correctness of the data.

**ParttimeEmployeeService.java:** Purpose-This interface extends the EmployeeService interface and has one abstract method that is only implemented by contract employees thus following the Interface Segregation Principle.

**PermanentEmployeeService.java:** Purpose-This interface extends the EmployeeService interface and has two abstract methods that are implemented only by permanent employees, thus following the Interface Segregation Principle.

**Application of SOLID principles:**

1. Two new classes were introduced from EmployeeController.java, PersistenceService.java and EmployeeValidator.java to adhere to the principle of Single Responsibility Principle as EmployeeController.java initially had more than one responsibility with many methods.
2. Both the TextFormatter.java and JSONFormatter.java implement the Formatter interface, thus extending the functionality of the Formatter interface without modifying the existing code adhering to the Open/closed Principle.
3. The EmployeeService interface was split into specific interfaces for permanent and contract employees, ensuring that the classes only implement methods relevant to them but not any other methods, adhering to the Integration Segregation Principle.
4. The class PersistenceService depends on the ‘Formatter’ interface rather than concrete classes and their implementations (‘JSONFormatter’ and ‘TextFormatter’), thus adhering to the Dependency Inversion Principle.

The UML diagram for this project:

A diagram of a company

Description automatically generated