**文本

描述已自动生成**

**Computer Systems Engineering**

**COMPSCI5059 Software Engineering M (2021-22)**

**Semester 2**

**Lab title: Assessed Exercise 2**

Date performed:05/03/2022

Date submitted:11/03/2022

**Lab partners:**

**Xian Zhang 2628107Z**

**Youqi Cheng 2698734C**

**Yufeng Li 2592737L**

**Jiwei Ma 2638638M**

**I. Team Introduction**

1. Team name: Group 31
2. Members ‘contribution:

**Jiwei Ma(2638638M):25%**

Responsible for Interface ListOfData; class LoR; class LoS.

**Youqi Cheng(2698734C):25%**

Responsible for class Person; class CourseDate.

**Yufeng Li(2592737L):25%**

Responsible for Interface FileIO; class FileIO\_LoD; class Main.

**Xian Zhang(2628107Z):25%**

Responsible for class Administrator; class Staff.

**II. User Stories**

1. **Produce a list of teaching requirements**

图形用户界面, 应用程序, Teams

描述已自动生成

The sequence diagram of this user story

图形用户界面, 图示, 文本, 应用程序, 聊天或短信

描述已自动生成The contained classes in this user story

图形用户界面, 文本, 应用程序

描述已自动生成

1. **Find staff to meet the teaching requirements**

图形用户界面, 应用程序, Teams

描述已自动生成

The sequence diagram of this user story

图示

描述已自动生成

The contained classes in this user story

文本, 信件

描述已自动生成

1. **Arrange the training for staff**

图形用户界面, 应用程序, Teams

描述已自动生成

The sequence diagram of this user story

图示

描述已自动生成

The contained classes in this user story

文本

描述已自动生成

1. **Add staff to the list**

图形用户界面, 应用程序, Teams

描述已自动生成

The sequence diagram of this user story

图示

描述已自动生成

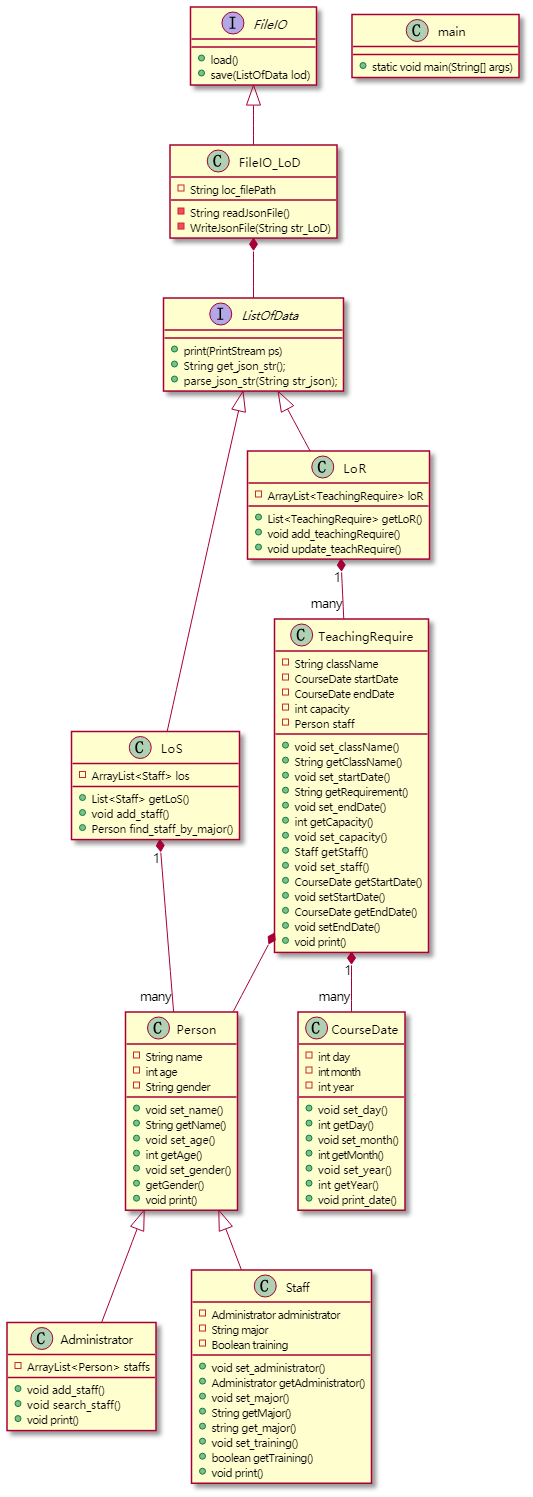
The contained classes in this user story

图形用户界面, 图示, 文本, 应用程序

描述已自动生成

**III. Class Structure Diagram**

1. The class structure diagram of our program will show below.



1. Analysis of class structure diagram
2. **Factory Pattern**

The function of our file read and write part is implemented in the factory pattern. Firstly, we design 2 Interface which was provided to the code user.

* The Interface “FileIO” is provided function” load ()” and “save ()” to code user to save or load the documents as a String;
* The Interface “ListOfData” is provided to code user to read or write the data restricted by JSON structure; Secondly, we design the classes to implement the Interfaces;
* The “FileIO\_LoD” implement the “FileIO” interface to read or store data of type LoD;
* The “LoS” and “LoR” implement the “ListOfData” interface to parse the data from JSON String or convert data to JSON string.

Finally, aggregate the “ListOfData” to “FileIO\_LoD”, and then we use the string variable " loc\_filePath" which is the path of the data stored file to instantiate the corresponding object. That is the factory pattern we implement using code.

The advantage of using factory pattern here is shown as below:

* We decoupled the read-write file data class from the teacher information list class and the teaching requirement class, which means we do not need to modify the file read and write class when we need to add, modify, or delete other types of data.
* We use interface class. In that way, the code users do not need to know the implementation of these functions. That means this code is easy to be used.
* Improves code cohesion. Using classes “FileIO” and “FileIO\_LoD” to handle file read and write functions. And using classes “ListOfData”, “LoS” and “LoR” to complete the data processing function inside the program.

1. **Class Inheritance**

The different personas in the program inherit from the "Person" class. More specifically, the “Staff” class and “Administrator” class are inherited from the “Person” class. The common properties, such as age, gender, and name, are abstracted to the "Person" class. In that way, these personas were decoupled from our code.

1. **Avoid Control Coupling**

Every variable in each class was private, and the “get” and “set” methods are implemented to avoid the control coupling.

**IV. Screenshots of code running**

**V. Retrospective**

1. We thought that when administrators search for staff to meet the teaching requirements, the requirements they input should be the same as the class name. Like inputting “Math” to search for staff to teach Math. But after discussion, it was decided that the requirements are not equal to class names. So we set a new attribute for staff, which is “major”. And then the requirements can match for the staffs’ major.