**Coursework Summary**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

In the following coursework attempt has been made to create simple classes and interfaces to implement few functions that the university department wants to perform on a student, as mentioned below:

1 – Identify the type of Student

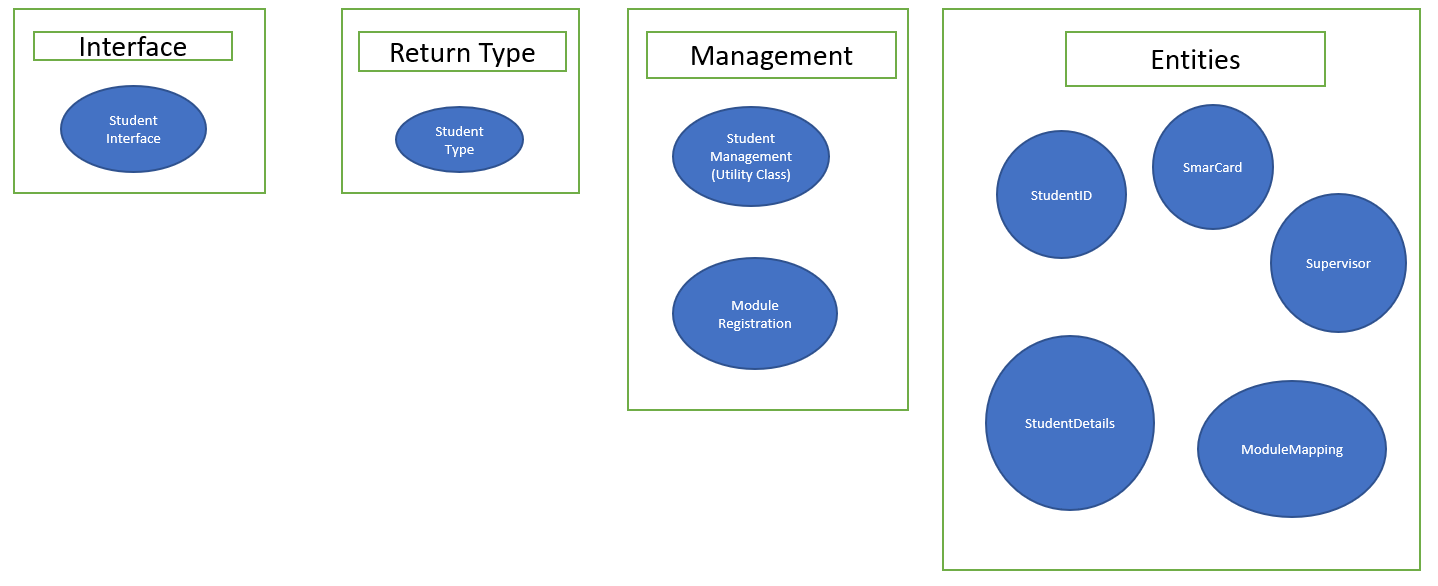
2- Manage the record of the student.

3- Issue smart card to a student

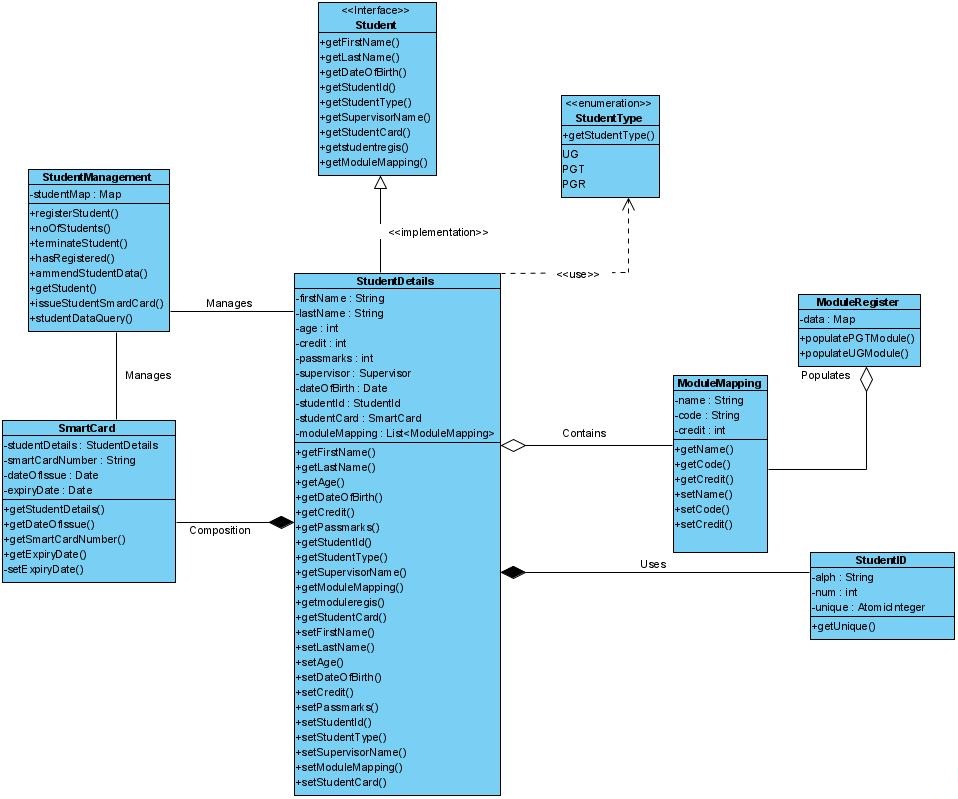
4- Addition of appropriate module to a student record

5- Day to day management activities like registering, terminating, updating, getting the no of students to name a few.

To achieve this the classes and interfaces that are created are grouped as per their roles and they are shown in the below Grouping diagram.



The relationship between these classes and interfaces are shown in the UML below.



Student Interface declares the minimum expected functions from the student class and it is the main Interface of the coursework.

StudentDetails class implements this Student Interface.

StudentDetails class uses the reference type Enum StudentType for getting the type of student like UG (Undergraduate), PGT (Post Graduate Taught and PGR (Post Graduate Research).

This StudentDetails class also generates and stores the unique Student ID using the StudentID class entity during creation of student while registration.

The StudentDetails also provides the details of the modules which a student is registered for and also a Boolean method to return the current registration status as per the credit criteria.

It also returns the supervisor name for the PGR students.

The StudentId class uses the AtomicInteger getUnique() method which in return uses the incrementAndGet() API functionality of AtomicInteger to generate a unique id specific to a particular student. The student id generated from this class has the following pattern a0001 which is a letter followed by a 4 digit number and it ensures that one unique id is created and attached to a particular student.

Similar kind of unique functionality is used to create the Smart Card Number for the student in the SmartCard Class which is again unique and has the pattern of <first initial of first name of student><first initial of last name of student>-(date of issue of card)-<arbitrary 2-digit number). Also, the conditionality for assigning a smart card based on age rule for UG and PG students, non-issue of more than 1 smart card per student as well as setting the expiry date of the smart card as per student type has been implemented in this class. This class is used by Student Management class in conjunction with the Student Details class to issue a SmartCard to a student during the registration process.

3 Data file namely, UGModule.csv, PGTModules.csv and StudentData.csv are created to read module and student data respectively.

Module Register is a class consists of the functionality to read the modules from a csv file UGModule.csv for UG students and PGTModule.csv for PG students and populate it against a student using the Module Mapping class which in turn is used by the StudentDetails class to map the modules to the respective students as per their type.

Student management class is kind of a normal class which acts as a utility, that manages the student. It populates default student data and module which is read from the csv data file called as StudentData.csv. The method that implements this functionality is studentDataQuery().

Using the data gathered from the Student Data file, Student management class is used to register a student into the student repo using the method registerStudent().

Student management class can also amends the student details of a student registered into the student repo by replacing its values from a new studentDetail object. This is achieved by using the ammendStudentDataMethod().

Student management class can also terminate a student registered into the student repo by making use of the terminateStudent() method. The only thing it terminates is the student instance.

Student management class can also assign smart cards to the student while registering using the issueStudentSmartCard() method.

Student management class also returns the count of students registered according to their type using the noOfStudents() method

Using hasRegistered() method it can check if a student is registered in the repository as well etch the details of a student registered into the repo using getStudent() method.

Therefore, from the above description it is visible that StudentManagement Class manages all the details of the student including the Smart Card.

Throughout the coursework it has been attempted to demonstrate the good practice and good design principles as covered in the lectures, namely appropriate overriding object class methods, interface-based hierarchies, Composition, use of appropriate Collection Framework and Object factories.

Finally, a Junit test class called as StudentManagementUnitTest is also created to perform the Junit tests on important methods of the coursework classes.

Apart from the classes and interface mentioned in the above UML diagram, there is also a main critical path class which has been created, to check the working of all the methods of the student management class which are described above.

A sample output from the run of that class is as follows:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Details of the first Student from the data file :

Student Name is : Craig Newton

Student's DOB is : Tue Jul 25 00:00:00 GMT 1995

Student's Age is : 25

Register this Student

Before registering student repo size : 0

After registering student repo size : 1

Get the list of Modules for which this student is eligible for registration :

[CSC8113,Group Project,20, CSC8103,Distributed Algorithm,10, CSC8111,Machine Learning,10, CSC8199,Project & Dissertaion,90, CSC8101,Data Analytics,10, CSC8112,IOT,10, CSC8110,Cloud Computing,10, CSC8106,System Evaluation,10, CSC8404,Advanced Java,10]

Registration Successfull : true

Get a Student Id for this student is : a0001

Issue a SmartCard to the student

Card No CN-2020-08 Issued to Student Id a0001

Issued Smart card details are as follow :

Smart Card Number : CN-2020-08

Student Name :Craig Newton

Student Id : a0001

Student DOB : Tue Jul 25 00:00:00 GMT 1995

Smart Card DOI : Fri Nov 06 16:05:28 GMT 2020

Smart Card DOE : Sun Nov 06 16:05:28 GMT 2022

Details of the second Student from the data file :

Student Name is : Kevin Smith

Student's DOB is : Sat Dec 07 00:00:00 GMT 2002

Student's Age is : 18

Register this Student

Before registering student repo size : 1

After registering student repo size : 2

Registration Successfull : true

Generate a Student Id for this student is : n0002

Issue a SmartCard to the student

Card No KS-2020-09 Issued to Student Id n0002

Issued Smart card details are as follow :

Smart Card Number : KS-2020-09

Student Name :Kevin Smith

Student Id : n0002

Student DOB : Sat Dec 07 00:00:00 GMT 2002

Smart Card DOI : Fri Nov 06 16:05:28 GMT 2020

Smart Card DOE : Wed Nov 06 16:05:28 GMT 2024

Details of the third Student from the data file :

Student Name is : Mark Ham

Student's DOB is : Thu Nov 08 00:00:00 GMT 1990

Student's Age is : 30

Register this Student

Before registering student repo size : 2

After registering student repo size : 3

Name of the supervisor :

John Walker

Registration Successfull : true

Generate a Student Id for this student is : a0003

Issue a SmartCard to the student

Card No MH-2020-10 Issued to Student Id a0003

Issued Smart card details are as follow :

Smart Card Number : MH-2020-10

Student Name :Mark Ham

Student Id : a0003

Student DOB : Thu Nov 08 00:00:00 GMT 1990

Smart Card DOI : Fri Nov 06 16:05:28 GMT 2020

Smart Card DOE : Thu Nov 06 16:05:28 GMT 2025

Details of the fourth Student from the data file :

Student Name is : Anne Ham

Student's DOB is : Sun Apr 14 00:00:00 GMT 1996

Student's Age is : 24

Register this Student

Before registering student repo size : 3

After registering student repo size : 4

Details of the fifth Student from the data file :

Student Name is : Linda Ham

Student's DOB is : Sat Mar 02 00:00:00 GMT 2002

Student's Age is : 18

Register this Student

Before registering student repo size : 4

After registering student repo size : 5

Total no of students registered in UG : 2

Total no of students registered in PGT : 2

Total no of students registered in PGR : 1

Studuent #y0005 Registered into student repo: true

Terminating 5th student

Before removing student repo size : 5

After removing student repo size : 4

Studuent #y0005 Registered into student repo: false

Ammending 3rd student

New Details of the third Student from the data file :

Student Name is : Mark Ham

Student's New Age is : 32

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**The End**