**Draft**

**Controller**

**Model**

**View**

Group Class

Session Class

Course Class

Student Class

Admin Class

Enumerator Class

FileIO Class

Notifier Class

Formatter Class

Menu Class

Logger Class

MySTAR Class

The MySTAR Class serve as the main interface for the user to interact with the system, with it having the all the methods to receive the user input and invoke the services of the other classes to serve their requests. This encapsulate the whole STAR system and hides the inner workings of it from the user. Therefore any changes made to any of the classes, with the exception of the MY STAR class, are transparent to the user, as the MySTAR class serve as their only interface to the system.

**The system follows the Model, View, Controller (MVC) framework and the rest of the classes that make up the system are categorised as one of the three components based on shared domain of interest that they work on.**

**The classes under View deal with formatting of any data that is to be displayed to the user and do not process the data in any other manner. The classes grouped as Controller provide services for the processing, loading and storage of data about the user and their requests to be registered to a course. The classes in Model represent various entities that are part of the process such as the Student Class holding data about students and the Course Class holding details about the Course offered.**

**By dividing the system into these three categories it is possible for us to achieve the following principals. Single Responsibility Principal (SRP), Dependency Injection Principal (DIP) and Open Closed Principal (OCP).**

**SRP states that a class should only have one responsibility so that the likelihood of changes made to it are minimized. The MVC framework helps in this regard by segregating the responsibility of your system into three parts, from there the classes that fall under any of the categories can further divide up the tasks that are needed to be done and provide specific services to handle them. This ensures that each class only perform one duty at most and minimise the chance of them having to be changed.**

**DIP refers to modules not being dependant on lower modules to function.**