Process Management Module

# MODULE INTERACTION

**Process Management  
Module**

**Database Module**

**IO Module**

# purpose of the module

This module was build in order to be an intermediary layer between IO and Database. The main purpose of this module is to convert data between incompatible modules and to create a way of communication between these without dependencies. In this way, Database Module can be replaced anytime without changing any behavior in the IO Module.

# Module Architecture

The main tree components of this module are:

* Domain Objects
* Interfaces
* Implementations

**Domain Objects**

Objects used in data transfer between modules. Can be defined as a container that stores information of a specific entity.

**Interfaces**In order to not produce dependencies for other modules when are using this module and for testability this module is exposing its functionality through interfaces.

**Implementations**

These are the concrete implementation of the interfaces.

# Entities As Domain Objects

**StudentDO –** this object reflects the student entity

|  |  |
| --- | --- |
| Properties | Type |
| Id | Int |
| Age | String |
| Gender | String |
| FirstName | String |
| LastName | String |
| EmailAddress | String |
| LinkedStatus | String |

**StudentClassDO –** this object reflects the class entity of a student

|  |  |
| --- | --- |
| Properties | Type |
| Id | Int |
| StudentId | Int |
| Name | String |
| Promoted | String |

**StudentStatusDO –** this object reflects the status entity of a student

|  |  |
| --- | --- |
| Properties | Type |
| Id | Int |
| StudentId | Int |
| Credits | Int |
| ECTS | Int |
| Year | Int |
|  |  |

# Entites repositores

**IStudentRepository –** this interface was built in order to offer the main operations that can be made with student entity.

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Method Behavior | Return Type | Parameters |
| GetAll | This method will return all student entities from database. | List<StudentDO> | NA |
| Get | This method will return a specific student entity. | StudentDO | studentId |
| Update | This method will update a specific student entity. | Void | StudentDO |
| Insert | This method will update a specific student entity. | Void | StudentDO |

**IStudentClassRepository –** this interface was built in order to offer the main operations that can be made with class entity.

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Method Behavior | Return Type | Parameters |
| GetAll | This method will return all class entities from database. | List<StudentClassDO> | NA |
| Get | This method will return a specific class entity. | StudentClass | classId |
| GetStudentClasses | This method will return all the classes of a specific student. | List<StudentClassDO> | studentId |
| Update | This method will update a specific class entity. | Void | StudentClassDO |
| Insert | This method will update a specific class entity. | Void | StudentClassDO |

**IStudentStatusRepository –** this interface was built in order to offer the main operations that can be made with status entity.

|  |  |  |  |
| --- | --- | --- | --- |
| Method Name | Method Behavior | Return Type | Parameters |
| GetAll | This method will return all status entities from database. | List<StudentStatusDO> | NA |
| Get | This method will return a specific status entity. | StudentStatusDO | statusId |
| GetStudentStatuses | This method will return all the statuses of a specific student. | List<StudentStatusDO> | studentId |
| Update | This method will update a specific class entity. | Void | StudentStatusDO |
| Insert | This method will update a specific class entity. | Void | StudentStatusDO |

# Entities – Naming Convention

This module is using reflection for mapping the database entities with domain objects.

The **„AutoMapper”** classis usedfor mapping the table information into a domain object based on columns names and properties names. If this class will receive one of the objects as null or empty, will raise an exception. Also, if one the columns from a table is not present in domain object as property an exeption will be raised.

**Entity : Object – Naming Convention**

|  |  |
| --- | --- |
| Columns | Properties |
| Id | Id |
| Name | Name |
| Status | Status |

# Insert/update queries

# Database connection