**République Algérienne Démocratique et Populaire**

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

**Université Benyoucef BENKHEDDA- Alger1**

**Faculté des Sciences**

**Département Mathématiques et Informatique**

****

**Master Ingénierie des Systèmes Informatiques Intelligents (ISII)**

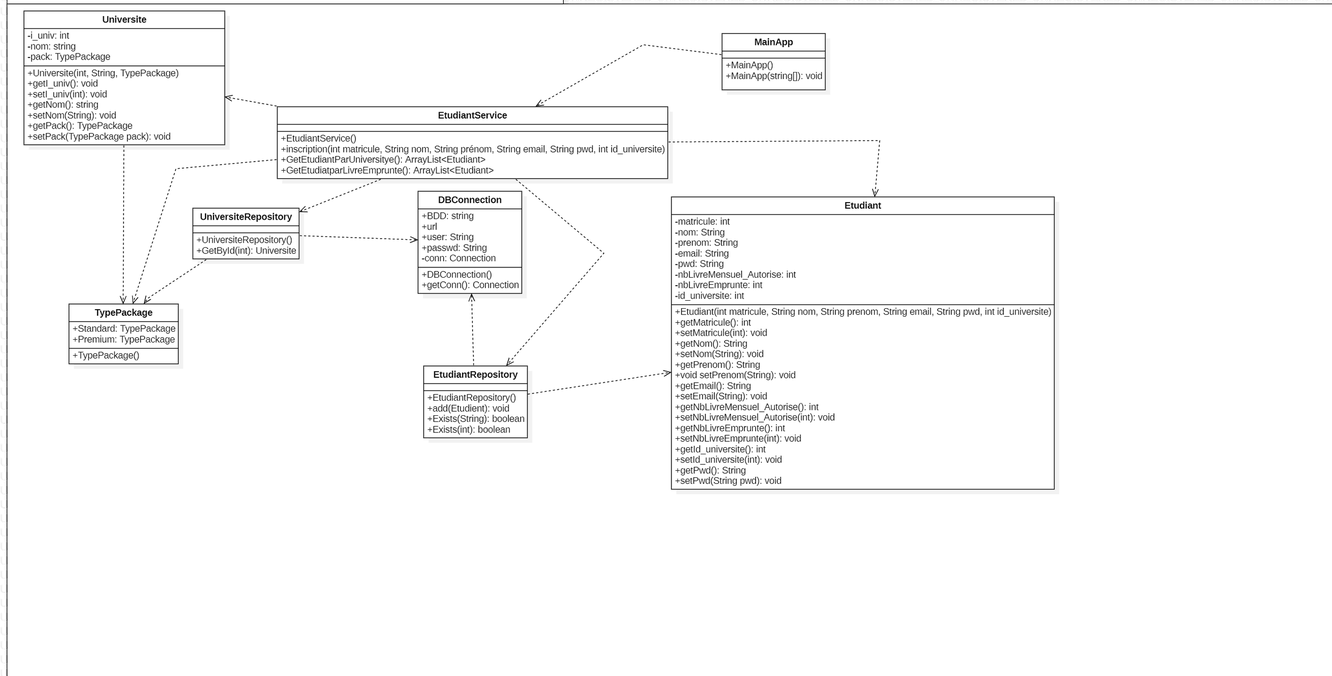
Rapport de TP2 Architecture des logicielles

***Etudiante :* LAZAZI Amina Imene .**

**2021/2022**

**Réponses aux Questions :**

**1. le diagramme de dépendance entre ces classes :**



**4. Expliquez l’objectif de l’inversion de contrôle:**

L'inversion de contrôle est un motif de conception logicielle commun à tous les Framework.

L'idée principale est que les modules de hauts et bas niveaux doivent dépendre d'interfaces, et non les modules dépendre les uns des autres. Les modules dérivent des interfaces. L'inversion de contrôle les rend donc indépendants entre eux.

L’objectif de cette amélioration est d’avoir un vocabulaire formalisé, capitalisation de l’expérience, niveau d’abstraction élevé, complexité réduite, temps de conception de développement réduit.

**10. Analysez chacune de ses responsabilités, puis décidez pour chacune si vous la gardez dans la méthode « inscription » ou l’affectez à une autre classe :**

•Permet d’inscrire un nouveau étudiant avec ses informations et avec Id Univ.

•Vérifier si la matricule et l’email déjà existent dans la base de données.

•Vérifier que l’email et la matricule sont insérés correctement.

**La solution proposée** est de garder les responsabilités dans la méthode « inscription » car ils sont reliés entre eux.

**15. Analysez le code de ces deux fonctionnalités et expliquez le problème qui se trouve dans ce code :**

À force de monter dans les abstractions, on finit par écrire des modules avec de plus en plus de dépendances et le module qui se trouve tout en haut ne connaît même pas sa règle métier même quand celle-ci dépend de lui.

Donc il est vrai que plus un module est abstrait, plus il a de dépendances, et donc beaucoup plus de risque de causer des instabilités

**Solution**

La clef pour résoudre cette énigme, c’est le principe de substitution de Liskov (LSP).

**26. Organisez vos classes en packages :**

On divise le code en 4 packages

**Etudiant** : Etudiant, EtudiantRepository, EtudiantService.

**Université**: Université, UniversiteRepository, ……

**Base de données**

**Interfaces :** Les interfaces.

**27. Donnez le diagramme de dépendance entre les packages :**

