**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)**

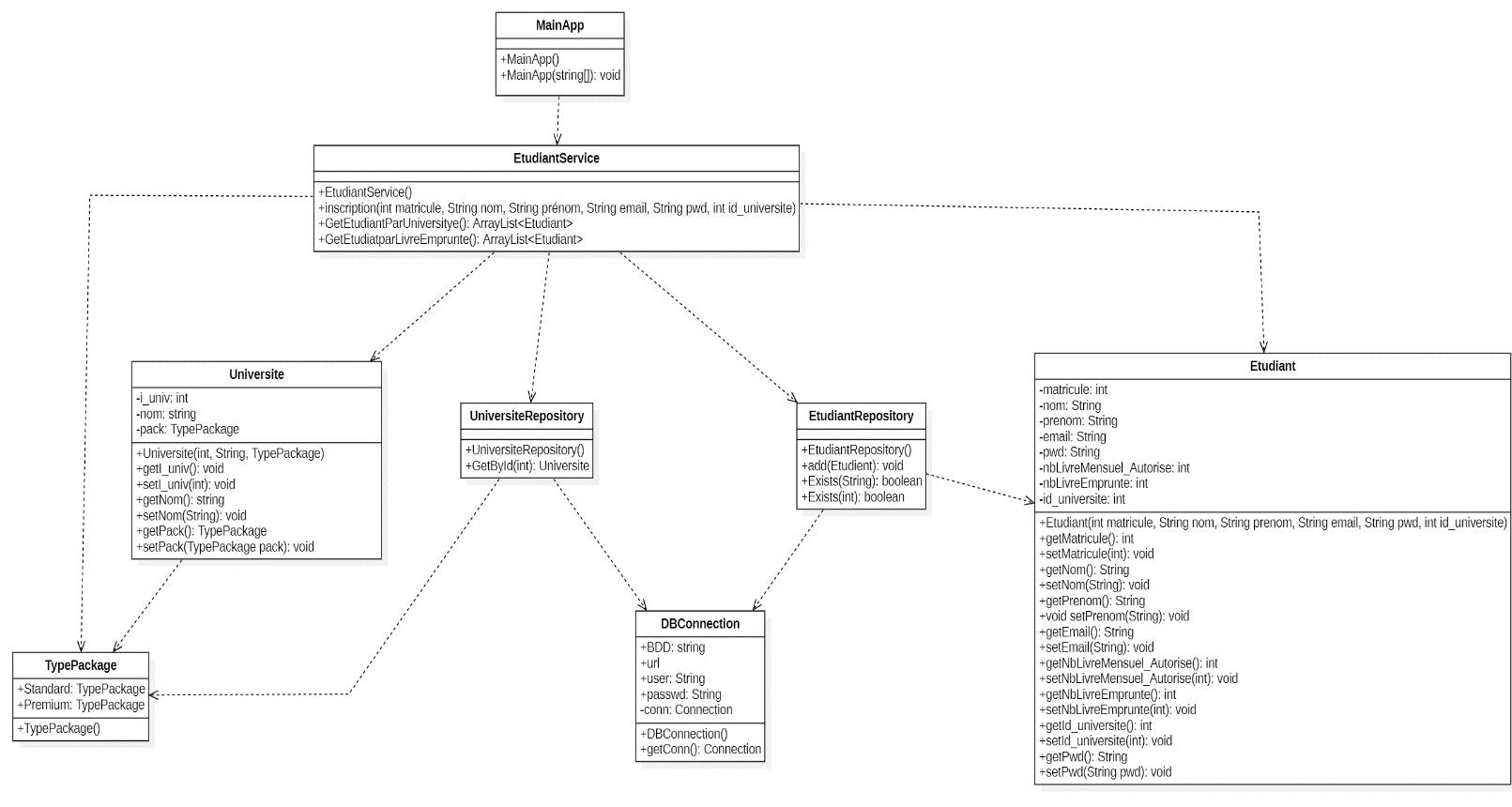
Rapport de TP2 Architecture des logicielles

***Réalisé par* : Zergoug Nabila**

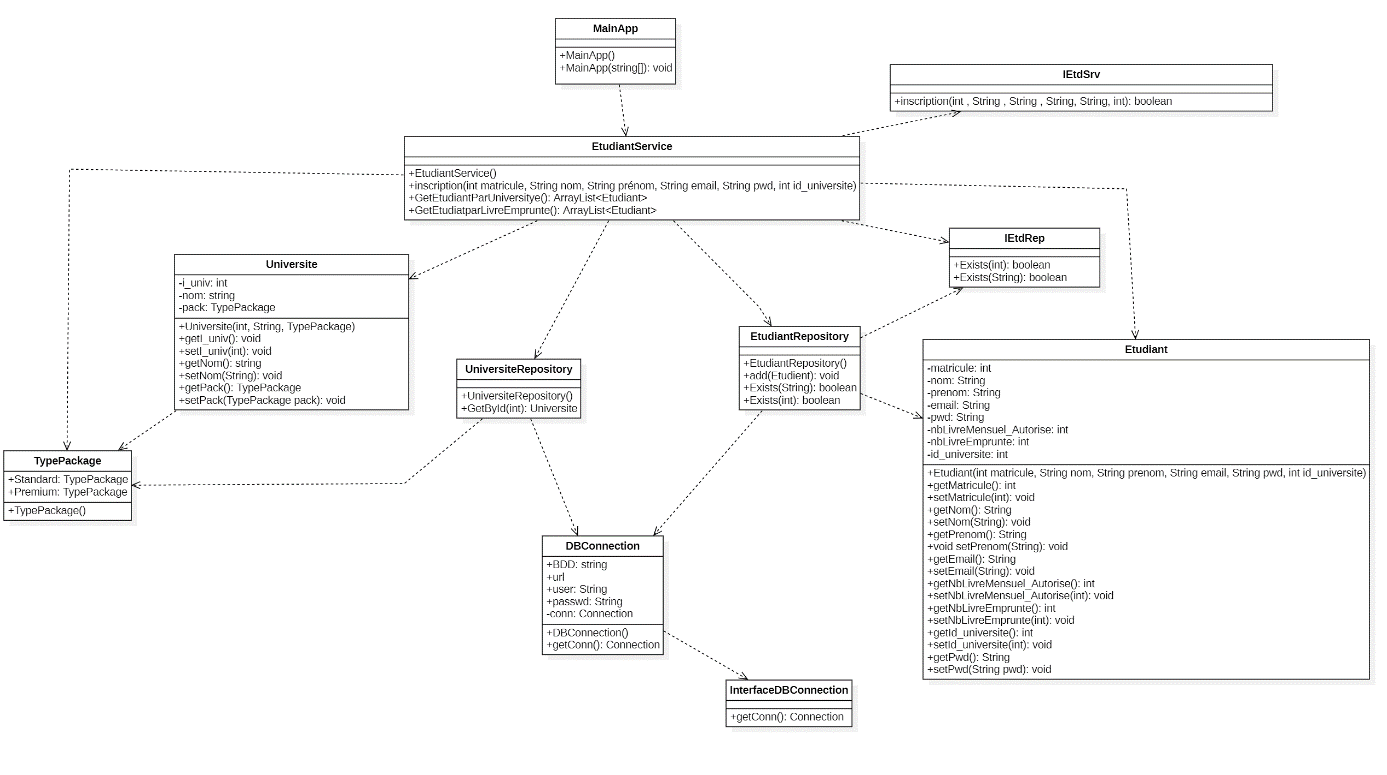
**2021/2022**

Réponses aux Questions :

1. le diagramme de dépendance entre ces classes :



4. Appliquez le principe de l’inversion de contrôle pour améliorer ce code. Expliquez l’objectif de cette amélioration :



L’inversion de contrôle fonctionne selon le principe que le flux d'exécution du logiciel n'est plus directement contrôlé par l'application elle-même, mais par le framework ou la couche logicielle sous-jacente.

L'objectif de cette amélioration est d'acquérir un vocabulaire formel, une capitalisation d'expérience, une abstraction élevée, de réduire la complexité et de réduire le temps de conception du développement.

Et donc j’ai ajouté c’est interface :

InterfaceDBConnection: pour l’interface de DBConnection

IEtdSrv: pour l’interface de EtudiantService

IEtdRep : pour l’interface de EtudiantRepository

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 :

La solution proposée après l’analyse est de garder les responsabilités dans la méthode « inscription » car ils sont liés entre eux (Jusqu'à peut-être des modification aux autres questions). La méthode inscription permet d'inscrire des nouveaux étudiants avec leurs informations et un identifiant de leurs universités. Vérifier si l'email et le matricule sont insérés. Et si ces derniers existaient déjà.

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

Lors de l'analyse des deux fonctionnalités. J'ai remarqué. Qu'il n'existe plus en plus de dépendance entre les classes. Du coup, il y a une grande possibilité de causer des instabilités.

26. Organisez vos classes en packages :

J’ai divisé le code en 4 packages :

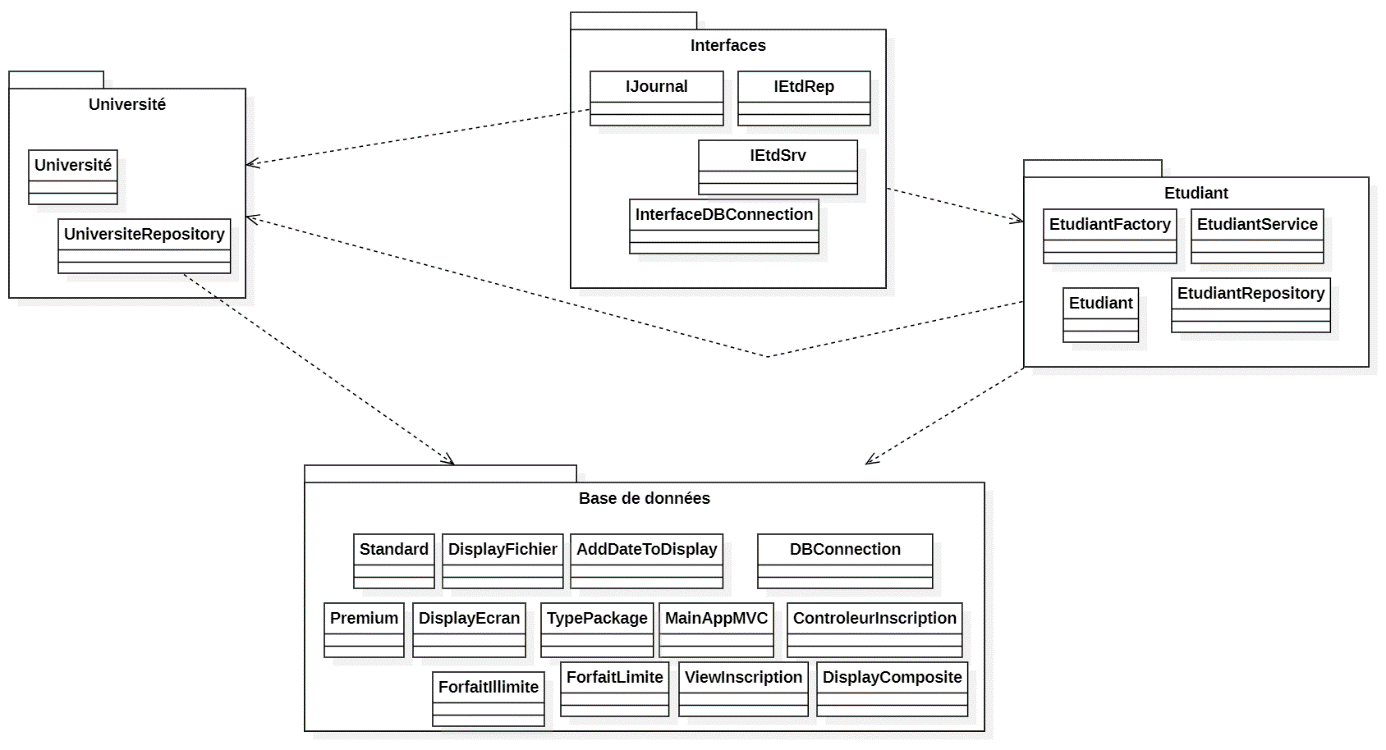
Étudiant : Contient toutes les classes qui ont une relation avec l’étudiant (Etudiant, EtudiantFactory, EtudiantRepository, EtudiantService, ).

Université : Contient toutes les classes qui ont une relation avec l’université (Université, UniversiteRepository).

Interface : Contient toutes les interfaces que j'ai créé (IEtdRep, IEtdSrv, IJournal, InterfaceDBConnection).

Base de données : Contient toutes les informations de la base de données. De ma bibliothèque numérique (AddDateToDisplay, ControleurInscription, DBConnection, DisplayComposite, DisplayEcran, DisplayFichier, ForfaitIllimite, ForfaitLimite , MainAppMVC, Premium, Standard, TypePackage, ViewInscription).

27. Donnez le diagramme de dépendance entre les packages :



**Remarque :**.

J’ai rencontré un problème lors de l'installation d’ObjectAid de Eclipse. Alors je ne pouvais pas faire les diagrammes sur lui. Du coup je les ai dessinées sur StarUML.