



Master Ingénierie des Systèmes Informatiques Intelligents (ISII)

Rapport de TP2 Architecture des logicielles

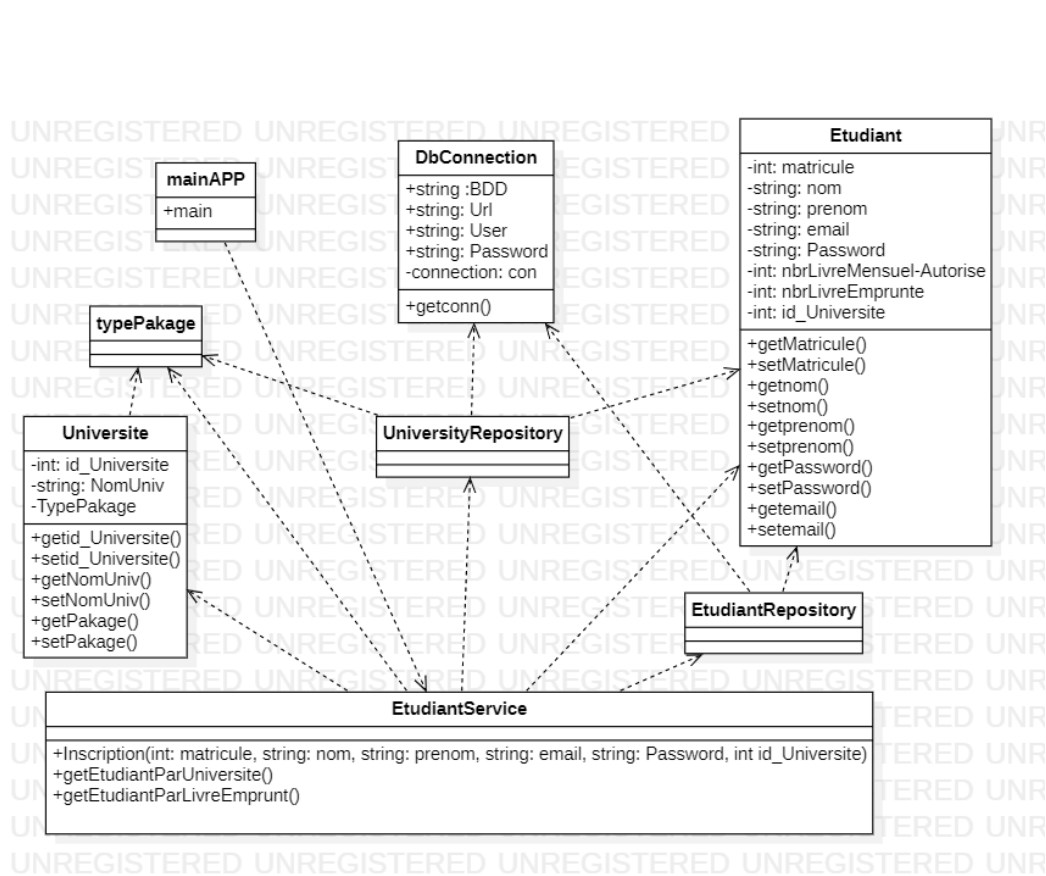
Réalisé par :

- ***ALLAOUI Maroua***

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 est un patron d'architecture commun à tous les Framework. Il fonctionne selon le principe que le flot d'exécution d'un logiciel n'est plus sous le contrôle direct de l'application elle-même mais du Framework ou de la couche logicielle sous-jacente

L'inversion de contrôle permettre au développeur de s'occuper uniquement du code métier (exigence fonctionnelles) et c'est des Framework qui s'occupe du code technique (exigence technique).

D'après ce principe les modules de haut niveau ne doivent pas dépendre des modules de bas niveau.

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 :

Après avoir faire l'analyse ma solution proposée est de garder les responsabilités dans la méthode « inscription » car ils sont reliés entre eux temporellement.

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

En faisant l'analyse de notre code, on remarque bien qu'il y a un problème. L'étudiant reçoit un bonus sur le nombre de livres mensuel mais chaque université a son propre forfait donc le forfait est limité donc toujours on a un nombre maximal d'après l'id de l'université. Dans notre code on n'a pas spécifié le forfait d'après l'université et ça que pose le problème. Et on remarque bien que le principe de LSP n'est pas respecté.

Solution : on doit bien respecter le principe LSP.

23. On souhaite que le contrôleur « ControleInscription » dépend de l'abstraction de la présentation et non pas de son implémentation. Comment peut-on réaliser ça ?

Le contrôleur « ControleInscription » dépend de l'abstraction de la présentation parce que le contrôleur gère la cohérence et dans notre cas il n'existe pas une forte cohérence donc il faut deviser les classes en packages et avoir une association entre les packages pour avoir un contrôleur dépend de l'abstraction d'implémentation.

26. Organisez vos classes en packages :

Pour pouvoir transformer nos classes en packages, j'ai organisé mon code en 4 packages :

Etudiant : Etudiant, EtudiantFactory , Etudiant repository.

Université : Université, universiteRepository.

Interfaces : EtudRepositoryInterf, IJournal, IEtdSrv.

Base de données

27. Donnez le diagramme de dépendance entre les packages :

