

Rapport de TP_2
Architecture logicielle

Réaliser par :

- *Ahlem Selmi*
- *Bouchra Belabbas*

Après l'ouverture du projet dans Eclipse et la création d'un repository dans GitHub on a commencé la réalisation de différentes questions :

Question 3 : Diagramme de dépendance entre les classes

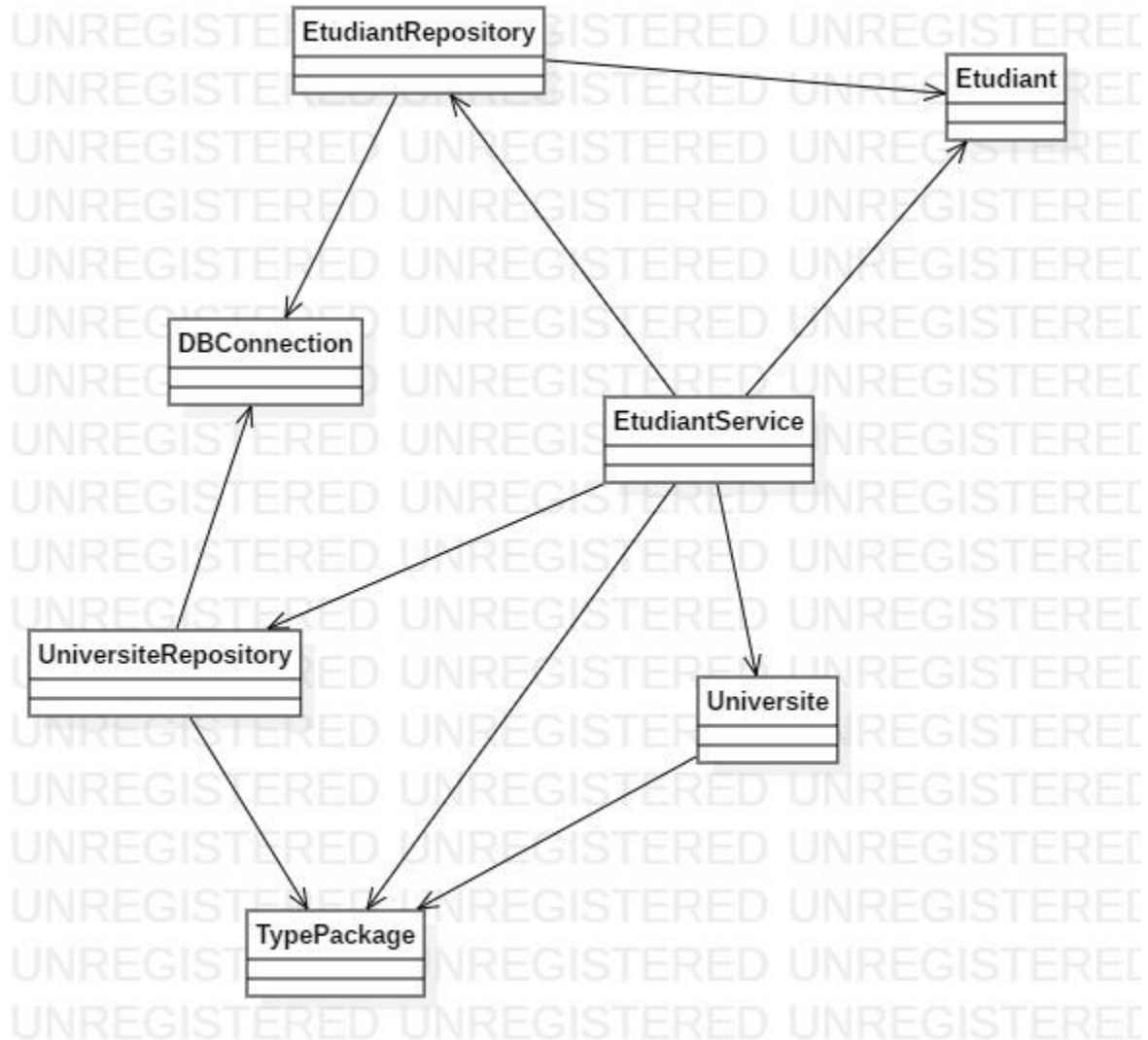


Figure 1: diagramme de dépendance

Question 4 :

Dans cette question on a implémenté le principe singleton pour avoir un seul point d'accès à la base de données, pour cela on a créé une méthode getInstance () dans la classe DBConnexion

```

public static DBConnection getInstance() throws SQLException{
    if(instance==null)
        instance=new DBConnection();
    return instance;
}

```

Figure 2: méthode getInstance

Question 5 :

Dans cette question on a appliqué le commit et le push

Question 6 : on a implémenté le principe de l'inversion de contrôle, pour ce là on a créé 5 interfaces utilisées par la classe EtudiantService

```

public interface EInterface {
    public int getNbLivreMensuel_Autorise();
    public void setNbLivreMensuel_Autorise(int nbLivreMensuel_Autorise);
    public int getMatricule();
    public String getNom();
    public String getPrenom();
    public String getEmail();
    public int getNbLivreEmprunte();
    public int getId_universite();
    public String getPwd();
}

```

Figure 3: interface IEtudiant

```

public class Etudiant implements EInterface{

```

Ces deux figures représentent un des interfaces quand on a créé et sa classe qui l'implémentent

Question 7 : là on a modifié le code principal et on l'exécute

Question 8 : Application du commit et push

Question 9 : Pour l'implémentation du journal on a créé un interface abstract « IJournal » avec une seule méthode abstract outPut_Msg ensuite on a ajouté les 4 classes qui implémentent l'interface IJournal ou chaque classe a un rôle à faire. En plus pour la classe 4 on a implémenté le patron composite.

Le méthode inscription dans la classe EtudiantService est responsable de l'inscription plus elle retourne un type string qui présente les messages du journal, même chose pour la méthode add dans la classe EtudiantRepository.

Question 10 : on a fait des appels aux différentes classes a fin d'afficher les messages du journal avec différents types d'affichage.

Question 11 : on a appliqué le commit et le push

Question 12 : On a affecté la responsabilité de vérifier le format email, existence du l'email et du matricule à la classe EtudiantRepository dans la méthode add, et on a affecté la responsabilité de l'initialisation de NbLivreMensuel à la classe Université.

Question 13 : On a appliqué le commit et le push

Question 14 : Là on a créé une méthode dans la classe étudiant que permet d'augmenter le NbLivreMensuel_Autorise par rapport le type de package de l'université.

Question 15 : Là on a créé une méthode dans la classe EtudiantService que permet d'augmenter le NbLivreMensuel_Autorise pour chaque étudiant par rapport le type de package de l'université.

Question 16 : On a appliqué le commit et le push

Question 17 : le problème c'est que la classe Etudiant ne peux pas définir elle-même ses propres objets à créer.