



**ECE**

PARIS ÉCOLE D'INGÉNIEURS

*Projet Java*

*ING3 2018-2019*

*« Gestion informatique d'une école »*

*Projet Java :*

*Gestion informatique  
d'une école*





# ECE

PARIS ÉCOLE D'INGÉNIEURS

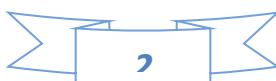
Projet Java

ING3 2018-2019

« Gestion informatique d'un école »

## Sommaire

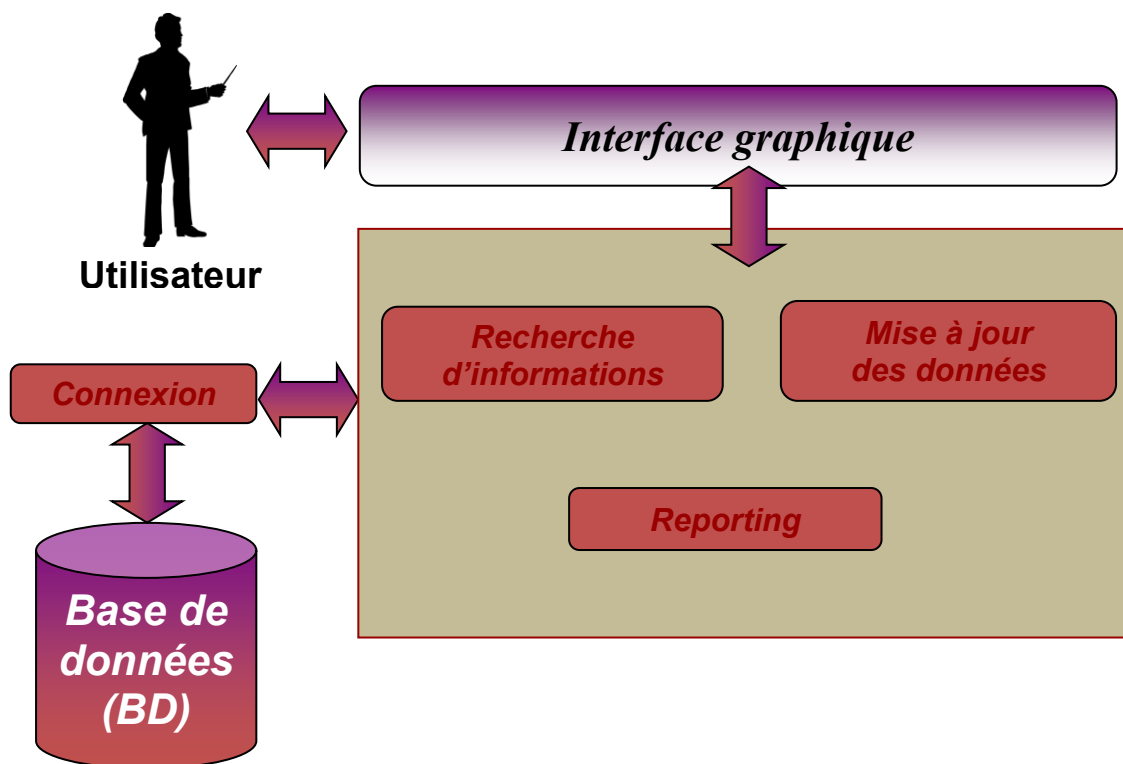
Introduction .....	3
Objectif.....	3
Architecture générale du système .....	3
Description de la base de données .....	4
Création de la base de données.....	6
Mise à jour des données via le système.....	6
Interrogation de la base de données via le système.....	6
Le module de Connexion.....	6
Le module de Reporting.....	6
L'interface graphique .....	6
Ressources sur campus .....	6
Travail à faire et planning .....	6
Livable à déposer sur campus .....	6
Déroulement de la soutenance .....	7
Critères d'évaluation du travail à réaliser .....	8
Recommandations .....	8



## Introduction

### Objectif

Ce projet vise à concevoir un système d'aide à la décision d'une école. L'architecture de ce système est décrite dans le schéma ci-dessous :



### Architecture générale du système

Dans cette section, nous présentons l'Architecture générale du système de gestion informatique d'une école. Ce système compte principalement 5 modules :

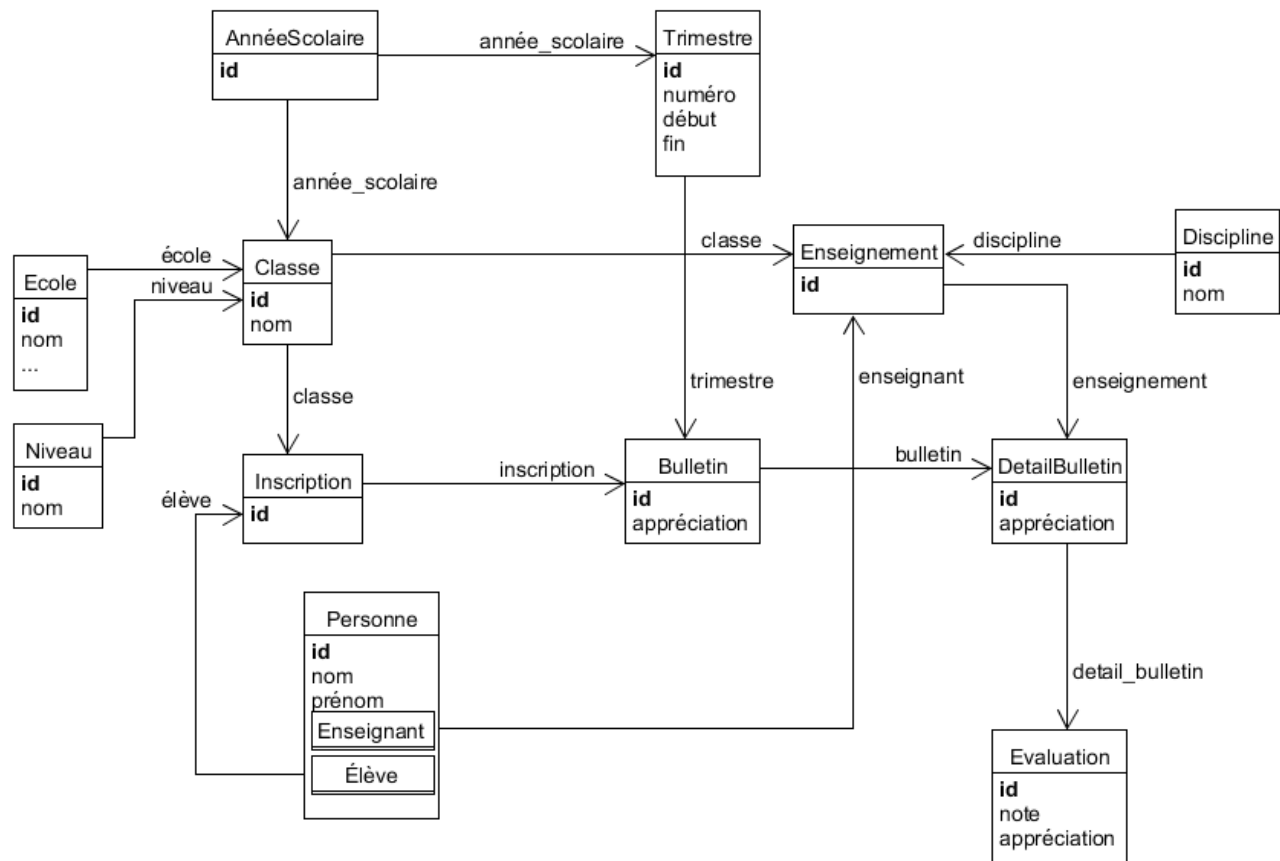
- Le module de **Recherche d'informations**
- Le module de **Mise à jour des données**
- Le module de **Reporting**
- Le module **Connexion** communique entre la BD et les 3 modules précédents
- L'**Interface graphique** communique avec les 3 premiers modules

L'interface graphique permet de mettre à jour (ajout, suppression, modification) toutes les données utilisées pour la gestion informatique de l'école, d'interagir avec les 3 autres modules, et d'afficher au fur et à mesure les résultats de chaque module du processus de gestion. Tous les modules communiquent via la base de données. Les résultats produits sont également stockés dans cette dernière.

L'implémentation du prototype de gestion informatique de l'école va être réalisée en Java (JDK 1.8) et la base de données décrite ci-dessous est stockée sous MySQL.

## Description de la base de données

Dans cette section, nous vous donnons ci-dessous la description de la base de données existante et qui sera exploitée par le système de gestion de l'école. Il s'agit du modèle relationnel d'une base décrivant les données d'une école :



### Remarques :

- Les clés primaires sont en gras
- Les -> indiquent quelle table référence quelle autre table. Par exemple, la table Trimestre référence la table AnnéeScolaire, ce qui signifie que Trimestre a une clé étrangère qui référence la clé primaire de AnnéeScolaire.

En définissant le schéma des tables, les hypothèses suivantes ont été faites, en indiquant les clés primaires en gras et les clés étrangères avec un # :

- Ecole (**id\_ecole**, nom\_ecole, ...) : La seule table facultative.
- AnnéeScolaire(**id**) :

Cette table traduit la notion d'année scolaire. On calcule l'année de début et celle de fin d'après les trimestres qui la constituent. On peut aussi stocker directement la première année et en déduire la seconde. La notion d'année scolaire est importante pour l'historique des classes



- Trimestre(**id**, numéro, début, fin, AnnéeScolaire.id#) :

La notion de trimestre sera importante pour les bulletins et par extension pour les évaluations.

- Niveau (**id**, nom) :

Une liste +/- fixe des différents niveaux scolaire possible : CP, CE2, etc... ; on pourra en ajouter quand le rectorat voudra renommer des niveaux scolaire (ajouter plutôt que modifier, afin de ne pas impacter les années précédentes)

- Classe(**id**, nom, Ecole.id#, Niveau.id#, AnnéeScolaire.id#) :

Représente une classe précise. Le champs nom permet de différencier les classes entre elles quand on a plusieurs classes de même niveau la même année dans la même école.

- Discipline(**id**, nom) :

Liste des disciplines enseignées

- Personne(**id**, nom, prénom, type) :

La notion de personne regroupe 2 notions similaires : enseignant et élève. Si on veut faire la différence, un champs type peut être introduit. Si des informations supplémentaires spécifiques aux enseignants ou aux élèves doivent être introduits, je suggère l'ajout de tables spécifiques.

- Enseignement(**id**, Classe.id#, Discipline.id#, Personne.id#) :

Un enseignant est une personne qui dispense l'enseignement d'une discipline pour une classe

- Inscription(**id**, Classe.id#, Personne.id#) :

Un élève est une personne qui est inscrit à une classe pour l'année de cette classe, laquelle est caractérisée par son niveau.

- Bulletin(**id**, Trimestre.id#, Inscription.id#, appréciation) :

Bulletin global, on calcule la moyenne globale du trimestre d'un élève inscrit, mais cette table fournit en plus l'appréciation globale.

- DetailBulletin(**id**, Bulletin.id#, Enseignement.id#, appréciation) :

Ligne dans le bulletin détaillant un enseignement précis : moyenne trimestrielle pour cet enseignement (calculée) et appréciation

- Evaluation(**id**, DetailBulletin.id#, note, appréciation) :

Une note précise, avec sa valeur et son appréciation. Elle est rattachée à un détail de bulletin, donc à un bulletin, donc à un trimestre.



## Création de la base de données

Dans un premier temps, vous devez créer la base de données, en respect du modèle relationnel présenté plus haut. Cette base sera créée en utilisant wamp avec phpMyAdmin. Une fois cette base créée, vous l'exporterez dans un fichier (script) d'extension .sql. Ce script permettra de créer et d'instancier (instructions d'insertion) les tables en les important dans n'importe quelle base de test.

## Mise à jour des données via le système

Ce module permet de mettre à jour (modifier, insérer, supprimer) les données de la BDD via l'interface graphique de votre système.

## Interrogation de la base de données via le système

Vous trouverez des exemples non exhaustifs de requêtes de recherche d'informations en **ANNEXE** de ce document

## Le module de Connexion

Les fonctionnalités de ce module se trouveront dans une classe. Via un accès JDBC à la base de données, il récupère ou met à jour les données à la demande des modules de Recherche, de Mise à jour et de Génération et Historisation des Activités.

## Le module de Reporting

Ce module permet de générer au fur et à mesure des statistiques (camemberts, histogrammes...) comme par exemple le nombre d'hospitalisés par type de service, les maladies les plus contractées etc.

## L'interface graphique

Votre interface graphique doit contenir un menu avec des items. Chaque item est associé à un module de votre architecture globale. Par exemple, pour mettre à jour les données d'une table particulière, il faut tout d'abord sélectionner votre item de mise à jour qui va ouvrir une fenêtre de mise à jour et mettre à jour la table.

Votre interface graphique constitue la *Vue* dépendante du *Contrôleur* (modules de Recherche, de Mise à jour et de Génération) qui interroge les données (*Modèle*) de la base, via le module de Connexion, pour transmettre ces données à l'interface graphique qui les affiche. Votre interface graphique sera événementielle (touches clavier ou/et clics souris). Une fenêtre d'accueil permettra de saisir le serveur de la base (par exemple, sql-users.ece.fr), le nom de la base, votre login et votre password pour vous connecter à la BD. Votre interface graphique affichera de manière ergonomique, claire et fluide toutes les informations pertinentes que vous fournit la BD.

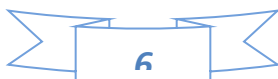
## Ressources sur campus

- Tous les tutoriaux disponibles sur la page campus du Projet Java <https://pedago-ece.campusonline.me/course/view.php?id=1911>: supports de cours, API java etc.

## Travail à faire et planning

### Livrable à déposer sur campus

Votre livrable sera un fichier zip contenant les documents suivants :





## 1) Un powerpoint avec les slides suivants :

- Page de garde avec titre, noms coéquipiers et groupe de TD
- Sommaire
- **Diagramme de classes** avec l'outil [Draw.io](https://draw.io) ou équivalent, présentant les attributs, les méthodes, les cardinalités, sans constructeurs ni getters/setters
- Conception technique présentant la modularité, selon l'approche **Modèle-Vue-Contrôleur**
- **Maquettes de votre interface graphique en commentant les composants graphiques Swing (ou autre librairie Java) utilisés et leur mise en page (layouts).**
- **Versioning GIT** : screenshot et lien avec login et passwd montrant la bonne utilisation et répartition des tâches du code entre coéquipiers.
- **Bilan individuel et collectif** de l'état du travail effectué.
- **Bibliographie de toutes vos sources** (web, livres etc.). **Toute source non citée est considérée de facto comme un plagiat.**

2) **Le code** : tous les dossiers et fichiers du projet développé sur Eclipse ou Netbeans avec les sources, l'exécutable .jar et la documentation **Javadoc** avec commentaires sur les classes et les méthodes

Le livrable est à déposer sur campus le **dimanche 9 juin 2019 à 23h55 dernier délai** à la page <https://pedago-ece.campusonline.me/course/view.php?id=1911>, en respect des consignes spécifiées ci-dessus.

## Déroulement de la soutenance

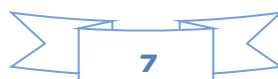
Les **soutenances auront lieu la semaine du 10 juin 2019**. La soutenance sera de **15 minutes maximum par équipe**. Elle permet au client (le prof ;-)) de tester votre programme et de vérifier l'adéquation de la solution que vous proposez avec le cahier des charges qui vous a été remis et **en adéquation avec la version graphique du code déposé sur campus**. Elle devra permettre (en très peu de temps pour le jury, faites attention ! C'est frustrant) de juger votre argumentation, l'esthétique (est-ce beau ?), la clarté (pertinence et précision des informations affichées ?), l'ergonomie (facile d'exécuter une action ?), l'originalité... de votre projet.

**A) Introduction** > Vous démarrez votre soutenance par une présentation sommaire mais précise, en environ 3 minutes, des slides du PowerPoint.

**B) Démonstration** > Vous présentez ensuite votre application en annonçant puis montrant chaque fonctionnalité. La démonstration doit être résolument orientée utilisateur (démonstration des fonctionnalités) : il faut 'vendre' votre application. Pour cela, vous devez dérouler un scénario d'utilisation que vous commentez en évitant le piège d'un discours technique. En l'occurrence, il est pertinent qu'un membre de l'équipe fasse la démo à mesure que le narrateur présente les fonctionnalités. Cette seconde partie doit prendre environ 7 minutes.

**Chaque équipe présentera son projet par vidéoprojecteur devant les autres équipes de son groupe de TD-TP.** L'objectif de cette soutenance est de démontrer que votre système fonctionne dans le respect du cahier des charges. Pour preuve, l'équipe exécutera son archive **jar** en mode graphique.

**Chaque étudiant d'une équipe doit intervenir lors de la présentation orale** et avoir une connaissance globale du projet de l'équipe. Le chargé de TP pourra interroger l'équipe et individuellement certains étudiants sur le **diagramme de classes**. Aussi prévoyez de présenter ces documents afin de pouvoir les argumenter. Un étudiant trop hésitant sur une question le concernant ou n'ayant pas une vision globale du projet pourra se voir pénaliser par rapport à ses coéquipiers. **Tous les étudiants du groupe doivent être présents sous peine d'être pénalisés.**





# ECE

PARIS ÉCOLE D'INGÉNIEURS

Projet Java

ING3 2018-2019

« Gestion informatique d'un école »

Attention, cette soutenance vise à vérifier si votre travail répond aux exigences du client. Ce dernier (le correcteur) **teste votre projet sur la version déposée sur campus dans une machine de l'école** et en vérifie l'adéquation avec ses attentes. S'il « achète » votre projet, vous avez une bonne note, sinon... elle sera mauvaise. Si vous avez développé sur un système particulier, venez avec votre matériel. Un plantage le jour du test client est du plus mauvais effet alors préparez vous un peu à l'avance et non à la dernière minute.

## Critères d'évaluation du travail à réaliser

En respect des consignes spécifiées précédemment, chaque équipe sera évaluée sur les critères suivants :

- La **soutenance** avec l'appui la **Javadoc** de votre code et bien sûr le **jar** de votre code **en adéquation avec celle déposée sur campus**. Elle est **notée sur 20 avec un coefficient de 2 dont 4 points pour le module de Reporting**. Vous serez jugés sur les points suivants : la **fiabilité de votre système**, la **cohérence de la base de données**, l'**interaction de votre programme Java avec la Base de données**, la **qualité de la démonstration** (le soin particulier apporté à sa qualité visuelle, sa fluidité, la pertinence des informations affichées en mode graphique, son originalité particulière, ...), la **qualité de la présentation orale** (respect du temps, clarté dans les explications, cohérence de l'argumentation, cohésion de l'équipe) et la **Javadoc** (montrant la qualité et les commentaires de votre code).
- Le **PowerPoint** sera **noté sur 20 avec un coefficient de 1**.
- Des **pénalités** pourraient être appliquées sur la note de la soutenance dans les cas suivants : retard de dépôt du livrable (**-2 points par jour de retard pour chaque dépôt**), fichiers manquants dans le code déposé, plagiat, absence à une soutenance, prestation individuelle peu ou pas convaincante.
- La **note finale sera égale à (soutenance \* 2 + bilan de conception + rapport final) / 4**.

## Recommandations

Le jour de la soutenance, vous devrez faire tourner le système que vous aurez déposé sur campus. **Aucune amélioration ou nouvelle version développée ultérieurement ne sera acceptée.**

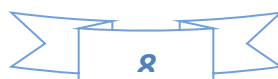
Pensez, après le dépôt sur campus, à télécharger votre dépôt sur une machine de l'école pour le faire tourner dans les conditions de la soutenance et pallier les anomalies.

Ne vous laissez pas abuser par l'illusion d'avoir beaucoup de temps. Respectez l'organisation prévue pour avoir terminé avant les vacances. Commencez suffisamment tôt pour vous organiser au mieux et faire quelque chose de bien. Ne soyez pas trop ambitieux au départ. Préférez faire proprement un projet simple et bien testé avant de partir dans les extensions. Travaillez par version en utilisant au besoin un logiciel de « **versionning** » et ne faites des modifications que sur une nouvelle version en conservant précieusement la dernière qui marche. C'est toujours trop bête de planter le jour de la soutenance.

Ne sous-estimez pas les phases de test qui doivent être régulières et sont presque aussi importantes que les phases de développement.

Pensez à faire appel à vos amis ou votre famille pour tester et critiquer ce que, vous-même, ne pouvez critiquer objectivement.

Vive l'école ☺



JP Segado