

# Projet de Bases de données

## Application de gestion d'autoroutes

---

### **Instructions générales**

Ce programme de gestion d'autoroutes devra être réalisé entièrement en Java ou PHP ou C# ou tout autre langage en accord avec votre chargé de TP.

#### **Vous pouvez vous aider :**

- ✓ Du support de cours (incontournable)
- ✓ Des ressources disponibles sur Internet, notamment sur le site d'Oracle
- ✓ De l'encadrement en TP/Projet.

#### **Modules, API et concepts concernés par ce programme (à titre d'information)**

- Concepts généraux de Java (héritage, polymorphisme, encapsulation, surcharge, méthodes...)
- Constructeurs, interfaces, classes abstraites,
- *Garbage collection*
- Collections
- Threads
- Input/Output
- Gestion de la persistance
- Swing

#### **Critères d'évaluation**

- Ergonomie (vous avez carte blanche sur le choix du '*look and feel*' de l'IHM). Mais votre interface doit être visuellement agréable, "*user friendly*" et intuitive en fenêtre graphique ou interface en ligne de commande.
- Qualité du code (commentaires, algorithmes utilisés)
- Respect du pattern MVC souhaité
- Gestion des exceptions
- Gestion optimale, flexible et simple de la persistance
- Richesse des concepts mis en œuvre et des API utilisées.
- Respect du cahier des charges

#### **Etapes de mise en œuvre**

- 1) Avant tout, imprégnez-vous des exigences du cahier des charges
- 2) Générez des maquettes de l'IHM. Pour ce, vous pouvez par exemple utiliser au choix l'un des sites suivants :  
<http://lumzy.com/app/>  
<https://gomockingbird.com/mockingbird/>
- 3) Rédigez un mini document de conception technique avec le dictionnaire de données, le MCD, MLD et/ou MPD, un diagramme de classes souhaité pour les classes Java voir aussi un diagramme de cas d'utilisation et/ou séquences élémentaire. Le but est d'illustrer le choix de vos entités amenées à être implémentées sous forme de classes et les interactions entre ces entités.

# **Cahier des charges**

## ***I – Introduction***

Ce programme est destiné aux gestionnaires des autoroutes pour l'identification des différents tronçons d'autoroutes, leurs interconnexions ainsi que les villes qu'ils desservent.

Il permettra deux types d'actions :

- ✓ La gestion des autoroutes (mise à jour des différentes autoroutes, tronçons, interconnexions, ...) selon l'évolution du réseau autoroutier ou de sa maintenance.
- ✓ La recherche des meilleurs parcours entre un point de départ et un point d'arrivée

## ***II – Fonctionnalités techniques***

- ✓ N'oubliez pas de créer vos packages spécifiques (ne pas utiliser -par exemple- le package par défaut de NetBeans).
- ✓ Lors de la fermeture du programme, un message de confirmation doit être proposé à l'utilisateur.
- ✓ Un couplage faible avec le SGBD est impératif. Ceci permettra entre autres d'inter-changer de SGBD avec un minimum de codage et de paramétrage.
- ✓ Pour la gestion de la persistance, l'utilisation des frameworks Hibernate ou JPA est autorisée. Toutefois nous attirons votre attention sur le point concernant le couplage faible.
- ✓ Libre à vous de choisir l'un des SGBD suivants pour implémenter votre base. Il faudra juste préciser la version que vous utilisez dans votre document technique. Pour tout autre SGBD, bien vouloir nous prévenir à l'avance.
  - a. Oracle / MySQL
  - b. PostgreSQL
  - c. Apache Derby (alias Cloudscape ou Java DB).
  - d. HSQLDB
  - e. Access / SqlServer
- ✓ Toutes les ressources ouvertes doivent être libérées au pire à la fermeture du programme.

## ***III – Gestion des autoroutes***

Le but de ce projet est de créer une base de données relationnelle "ROUTES" qui satisfait le descriptif suivant :

- cette base « ROUTES » doit contenir des tronçons d'autoroutes qui correspondent au découpage de l'autoroute en plusieurs morceaux afin de faciliter sa gestion. Chaque tronçon est identifié par un code (CodT), appartient à une autoroute donnée de nom CodA (ex: A6) et est délimité par un Kilomètre initial (DuKm) et un Kilomètre final (AuKm).
- les autoroutes possèdent des sorties vers des villes. Ces Sorties portent un nom de sortie (Libelle) et un numéro de sortie (Numero) de valeur 1, 2, ... pour l'autoroute en question. Les villes joignables par autoroute sont identifiées par un code postal (CodP) et décrites par un Nom. Les autoroutes concédées appartiennent à l'État qui en a confié, pour une durée déterminée, le financement, la construction, l'entretien et l'exploitation à des sociétés concessionnaires d'autoroutes en contrepartie de la perception d'un péage. Pour cette raison, certains tronçons d'autoroutes possèdent un système de péage sur une partie de tronçon délimité par un début (PGDuKm) et fin (PGAuKM) et correspond à un tarif spécifique. Un Système de péage est géré par une entreprise (SCA) identifiée par un code et possède un nom, un chiffre d'affaire et la durée de son contrat. Tous les tronçons qui sont fermés pour travaux ou accident sont enregistrés dans un registre de fermeture avec une date de début et une date de fin de fermeture et un descriptif de la cause. Le registre est identifié par numéro d'enregistrement et un descriptif.

A partir d'une interface graphique, on souhaiterait interroger la base de données "ROUTES" pour :

- Mise à jour des listes d'autoroutes, tronçons, interconnexions et villes desservies.
- Sélectionner et afficher les codes d'autoroutes, et les libellés de la sortie joignant une ville donnée

#### ***IV – Recherche de parcours et des sociétés de gestion d'autoroutes***

- Afficher les tronçons à parcourir des autoroutes pour aller d'une ville à une autre en tenant compte des fermetures
- Proposer l'affichage des tronçons qui sont gérés par une SCA donnée et en affichant le temps restant de leur contrat et leur gain sur (100 véhicules qui prennent leur péages)

#### ***V – Fonctionnalités bonus***

Ce défi rapportera **+3 pts sur la note finale** aux membres du groupe

Seule condition : **Toutes** les fonctionnalités doivent être mises en œuvre.

- ✓ Login avec deux niveaux de privilèges : **administrateur** et **utilisateur**.
  - L'administrateur aura des droits complets de mise à jour
  - L'utilisateur ne pourra utiliser l'application qu'en mode consultation. Toutefois il n'aura aucune restriction pour imprimer.
- ✓ Possibilité à deux clients "administrateurs" de se connecter simultanément à la base ce qui implique de votre part la mise en œuvre d'une gestion transparente et optimale de l'accès concurrentiel au SGBD (Threads, concurrence d'accès, transactions).

### **Consignes d'organisation**

- S'organiser en groupes de 2 membres maximum par groupe (liste à remettre à la fin du 1<sup>er</sup> Tp/Projet)
- Séance de validation du travail fini à la dernière séance de Tp/Projet
- Le rendu sur campus doit se faire avant le vendredi 26 mai 2017 à 23h50.

### **Rendus attendus**

- 1) Au plus tard, le **26/05/2017 à 23h50**, vous devez déposer vos livrables dans un fichier zip, sur Campus (Projet source Java + scripts SQL + Rapport).
- 2) Toutes les classes java et documents doivent contenir les noms des membres du groupe en en-tête.
- 3) Les livrables seront importés sur votre projet dans Eclipse. Objectif "zéro erreur" à ce niveau.
- 4) La documentation comptera pour 20% de la note du projet et sera évaluée minutieusement avant même le lancement de votre programme.