

Projet Bases de Données et Conception Objet

1 – Introduction

Ce projet a pour but de mettre en œuvre vos compétences en systèmes de gestion de bases de données relationnelles et en conception objet. Ce projet vient en application du cours de « Principes des systèmes de gestion de bases de données » (PSGBD) et du cours d'« analyse et conception objet de logiciels embarqués » (ACLE). Vous devez réaliser l'analyse, la conception et la programmation complète d'une application de gestion d'une application d'autopartage (type *BlaBlaCar*) utilisant une base de données. Vous devez notamment concevoir et implanter un schéma relationnel. Le programme doit être écrit en *Java*, et les accès à la base de données effectués grâce l'API *JDBC*.

Le projet est à faire en équipes de cinq et donnera lieu à la remise d'un rapport, ainsi qu'à une soutenance comprenant, entre autres, une démonstration. L'évaluation concerne l'analyse, la conception et l'implémentation, à la fois sur les aspects bases de données et sur la partie orientée objets.

2 – Description de l'application

La société *VerbiageVoiture* (ou *VV*) souhaite que vous l'aidiez à la conception et à l'implémentation de son application d'autopartage utilisant une base de données relationnelle et offrant une interface utilisateur graphique. Dans l'autopartage, certains utilisateurs proposent des places dans leur voiture sur les trajets qu'ils vont effectuer, ils sont donc conducteurs sur ces trajets. D'autres utilisateurs cherchent des places pour leurs déplacements et pourront donc être passagers. Le but de *VerbiageVoiture* est justement de mettre ces utilisateurs en relation et de leur faciliter les démarches et le paiement des déplacements. Tout utilisateur peut être conducteur sur certains trajets et passager sur d'autres.

Les utilisateurs sont identifiés par un email avec lequel ils se sont inscrits à *VV*, et sont décrits par leur nom, leur prénom, leur ville de résidence et un mot de passe. Certains utilisateurs peuvent conduire un ou plusieurs véhicules. Chaque véhicule est identifié par son immatriculation et on en connaît la marque, le modèle, l'énergie utilisée (essence, diesel ou électrique), la puissance fiscale et surtout le nombre de places qu'il offre. Il est possible que plusieurs utilisateurs conduisent le même véhicule.

Un trajet, identifié de façon unique et proposé par un utilisateur, part à une date et une heure donnée et est réalisé en utilisant un véhicule dont l'utilisateur est conducteur. On précise également le nombre de places disponibles dans le véhicule au départ. Un trajet est divisé en un ou plusieurs tronçons. Chaque tronçon comporte les villes et les coordonnées GPS (longitude et latitude) des points de départ et d'arrivée, la distance parcourue et le temps de parcours estimé (en minutes). Une durée maximum d'attente des voyageurs au départ peut également être précisée. Un trajet peut donc être composé de plusieurs tronçons,

et des passagers peuvent monter dans le véhicule à chaque étape. Cela permet également aux passagers de construire leurs parcours en combinant des tronçons de différents trajets (et donc dans différents véhicules). VV limite la possibilité de tels montages à un changement de véhicule maximum, et encore si l'attente prévue à l'étape n'excède pas une heure, et que la distance (euclidienne) entre les points GPS d'arrivée et de départ de l'étape ne dépasse pas 0,01 (environ 1 km).

Pour faciliter les paiements, VV associe à chaque utilisateur un porte-monnaie. Celui-ci pourrait être crédité par une carte bancaire par exemple, mais pour cette application cet aspect devra être simulé simplement. Le porte-monnaie est utilisé pour payer les tronçons parcourus par les passagers aux conducteurs et ne doit jamais avoir un solde négatif. Le tarif kilométrique pour un passager dépend de la puissance fiscale et de l'énergie utilisée par le véhicule (*Coût kilométrique* = *Puissance fiscale* × α × 0,10 € ; α valant 1 pour l'essence, 1,5 pour le diesel et 0,5 pour l'électrique).

Prototype à réaliser

Votre prototype doit couvrir les trois principales activités de *VerbiageVoiture* : (i) la proposition de trajets/tronçons par les chauffeurs, (ii) la composition des parcours par les passagers, et (iii) le suivi des parcours et le paiement. Nous allons maintenant décrire les fonctionnalités de ces trois activités (certaines fonctionnalités peuvent être communes).

Proposition de trajets

- 1) La création d'un compte sur VV.
- 2) L'ajout de véhicule(s) pour le compte.
- 3) La proposition d'un trajet et du ou des tronçons le composant, en précisant la durée éventuelle des arrêts entre deux tronçons.

Composition de parcours

- 1) La création d'un compte sur VV.
- 2) La recherche d'un parcours à partir d'un point de départ (ville et coordonnées GPS). Les solutions proposées devront en priorité ne comprendre qu'un seul trajet (un ou plusieurs tronçons appartenant à un même trajet proposé). Des solutions combinant des tronçons de deux trajets différents seront ensuite proposées. Bien sûr les propositions ne pourront comprendre que des tronçons pour lesquels il reste de la place dans le véhicule. Les critères de recherche peuvent être :
 - a) Quelles sont les destinations possibles à partir du point de départ (ville et coordonnées GPS) à une date donnée ?
 - b) Comment aller du point de départ à un point d'arrivée donné ?

Les solutions proposées indiqueront les distances (euclidiennes) entre les points de départ et d'arrivée souhaités, l'heure d'arrivée estimée, la liste éventuelle des étapes intermédiaires, ainsi que le ou les véhicules utilisés (marque, modèle, immatriculation). Une estimation du prix du parcours sera également donnée pour chaque proposition.

Suivi des parcours et paiement

- 1) Pour les conducteurs :
 - a) Confirmation du départ et de l'arrivée des conducteurs sur les tronçons qu'ils ont proposés.
 - b) Annulation de tronçons en cas de problème.

- 2) Pour les passagers :
 - a) Rechargement du porte-monnaie (simulé).
 - b) Confirmation des montées et descentes des véhicules au fur et à mesure de leurs parcours, une fois les conducteurs arrivés aux étapes bien sûr.
 - c) Paiement des tronçons effectués en fin de parcours.

3 – Travail à réaliser (12h encadrées + travail personnel)

3.1 – Analyse

Cette étape consiste à analyser le problème posé.

- a) **Réaliser l'analyse dynamique**, en identifiant les différents cas d'utilisation. Faire un diagramme de cas d'utilisation. Documenter le fonctionnement global du système par un ou plusieurs diagrammes états/transitions pertinents. Documenter les cas d'utilisation par des explications ; utiliser un ou plusieurs diagrammes de séquence pour illustrer les points ambigus.
- b) **Réaliser l'analyse statique** : inventorier les données, identifier les dépendances fonctionnelles et les contraintes (invariants) entre les données persistantes.

3.2 – Conception de bases de données

- a) **Élaboration du schéma conceptuel**. Construire, à partir de l'analyse statique un schéma Entités/Associations (ce schéma correspond au diagramme de classes d'analyse). Expliquer ce schéma (notamment les points difficiles de sa construction). Préciser les contraintes d'intégrité non représentées dans le schéma Entités/ Associations.
- b) **Conception de la base de données**. Traduire le schéma Entités/Associations en relationnel ; écrire les tables et les contraintes. Préciser et justifier la forme normale des relations, les contraintes d'intégrité implantées, ainsi que les contraintes ne pouvant être implantées en SQL (et qui sont donc à vérifier dans le code applicatif Java).
- c) **Analyse des accès à la base de données**. Écrire les requêtes nécessaires aux différentes fonctionnalités à implanter. Délimiter les transactions permettant d'implanter les fonctionnalités tout en préservant la cohérence globale des données.

3.3 – Conception de l'application

- a) **Architecture du système**.

Proposer une architecture pour le prototype.

Préciser en particulier l'interface utilisateur ainsi que les liens avec la base de données. Pour cela, vous étudierez deux scénarios typiques :

 - i. une action de l'utilisateur sur l'interface.
 - ii. le déclenchement et l'exécution d'une requête depuis l'application.

Vous pourrez illustrer ces scénarios, à l'aide d'un diagramme de collaboration ou de séquences.
- b) **Conception objet**. Préciser les patrons de conception que vous avez utilisés en donnant un diagramme de classe (local au patron) et en précisant son positionnement et son utilisation dans l'architecture globale de votre système.

3.4 – Implémentation

- a) **Instancier la base de données.** Elle doit être créée sur le serveur Oracle disponible sur le serveur *oracle1*.
- b) **Programmer l'application en Java.** L'accès à la base de données se fera avec l'API JDBC.

3.5 – Bilan

Rédiger un bilan de ce projet (répartition des tâches, difficultés rencontrées, leçons retenues, *etc.*). Pour vous aider dans cette tâche, il vous est demandé de maintenir **un journal de projet** dans lequel vous préciserez (en quelques lignes) ce qui a été réalisé au cours de chaque séance encadrée de projet (ou en dehors de ces séances). Ce journal devra être déposé sur l'application *Teide* **à la fin de chaque séance** encadrée (déposé, **mais non validé**) et pourra être consulté à tout moment par vos encadrants tout au long du projet.

La **documentation finale** à rendre reprend le travail à réaliser, de 3.1 à 3.5.

4 – Soutenance

Votre projet donne lieu à une soutenance de 20 à 25 minutes (inscription sur les créneaux horaires via l'application *Teide*), suivie de 5 à 10 minutes de questions. La soutenance devra reprendre le travail réalisé (3.1 à 3.5) en choisissant les points les plus pertinents pour mettre en valeur le projet. En particulier, il est attendu :

- Une présentation de votre modélisation entités-associations et de son implantation en relationnel (principalement les points difficiles),
- Une présentation de votre architecture et de votre conception (en insistant sur l'articulation entre les différents éléments de l'architecture)
- Une démonstration des fonctionnalités ; prévoir un scénario montrant le bon fonctionnement des transactions,
- Un bilan du déroulement du projet.

L'objectif est de vendre **votre produit** (en montrant qu'il fait bien ce qu'on attend) **et votre équipe** de projet (en montrant que le produit a été bien construit et que le projet a été bien géré).

Note : Il est très fortement recommandé de consulter régulièrement les pages de Chamilo consacrées à ce projet, en particulier le forum.