L3/L3Int **Mai 2021**

Projet de Bases de données (TI-603)

Application Rentcar

Instructions générales

Ce projet de gestion de location de véhicules devra être réalisé entièrement en Java et une base de données relationnelle MySQL, Oracle, SqlServer, Access, PostgreSql ou autre.

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)

- Base de données respectant au minimum la 3eme forme normale opérationnelle
- Création de vues nécessaires pour la bonne gestion des droits d'accès
- Gestion des transactions
- Concepts généraux de programmation objet (héritage, polymorphisme, encapsulation, surcharge, méthodes...)
- Constructeurs, interfaces, classes abstraites,
- Collections
- Threads
- Input/Output
- Gestion de la persistance
- IHM en mode console ou graphique

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 selon votre choix en fenêtre graphique ou interface console en ligne de commande.
- Qualité du code (commentaires, algorithmes utilisés)
- Qualité de la base de données
- Bonne gestion des transactions et de la concurrence d'accès
- Respect du design 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 le site suivant : 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

Cette application est destinée aux gestionnaires des agences de location de véhicules pour l'identification des différents clients et la gestion de leurs réservations, locations et retours de véhicules en location.
Il permettra deux types d'actions :

- ✓ La gestion des clients et des véhicules (mise à jour des informations sur les clients : inscription, mise à jour des changements d'informations personnelles, mises à jour des informations sur les véhicules : ajout/suppression de véhicule ainsi que sa disponibilité et/ou indisponibilité).
- ✓ La réservation, la location et le retour de véhicules par les clients.

II – Fonctionnalités techniques

- ✓ N'oubliez pas de créer vos packages spécifiques (ne pas utiliser -par exemple- le package par défaut de Eclipse).
- ✓ 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'interchanger de SGBD avec un minimum de codage et de paramétrage.
- ✓ Pour la gestion de la persistance, l'utilisation de Framework 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 / MariaDB
  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 locations

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

- Les clients ont un nom, un prénom, un email, une adresse (rue, ville, code postal) et un numéro de téléphone. Ayant toutes ces données, on pourra mieux informer les clients de différentes offres promotionnelles. Un client peut souscrire à un programme de fidélité qui a une durée en années, une description, un prix et un taux de réduction applicable sur les locations. On doit connaitre la date de souscription pour vérifier s'il a le droit à une réduction lors de ses locations. Le client peut renouveler sa souscription à un programme de fidélité après son expiration ou choisir un autre.

- Les véhicules ont un matricule, une marque, un modèle, le kilométrage, le type de boite à vitesse manuelle ou automatique, climatisé ou pas, et le type de carburant (Gazole, Essence SP95, GPL, électrique, etc…). Un véhicule appartient à une catégorie économique, confort ou luxe. Chaque catégorie a un tarif par jour une caution. Les clients peuvent effectuer des réservations, des locations ou rendre un véhicule déjà loué.

- Les employés ont un nom, un prénom, un email, une adresse (rue, ville, code postal) et un numéro de téléphone. Ils se connectent à l'application par un login et un mot de passe personnel. Certains employés gèrent les réservations, les locations et les retours de véhicules. D'autres sont des chauffeurs disposant de camions de transports de véhicules et qui ont la tâche de répartir les véhicules au niveau des agences selon le besoin. Ils se connectent à l'application pour obtenir leur circuit et les listes des véhicules à déplacer entre les agences. Chaque agence dispose d'un identifiant, un nom, un téléphone, une adresse et des coordonnées GPS pour sa géolocalisation. On doit savoir à tout moment les véhicules stationnés à son niveau.

Gestion des devis/locations

Toute location est matérialisée sous la forme d'un devis comportant les informations suivantes :

Informations complètes sur le client, sur le véhicule, durée prévue de la location. En fonction de la durée de location et de la catégorie du véhicule, l'application produira un devis en indiquant le montant de la réduction à laquelle il a le droit s'il adhère à un programme de fidélité valide. Il vous appartient de définir les différents tarifs appliqués.

Rentcar propose en option une assurance contre la dégradation et les accidents, à un tarif que vous choisirez.

Tout véhicule peut être loué pour une certaine durée avec un tarif convenu à l'avance selon le devis proposé systématiquement au client. Au retour du véhicule, Rentcar doit éditer une facture tenant compte des points suivants :

- la durée effective de location: si le véhicule est rendu en retard par rapport à la date convenue, une pénalité (que vous définirez) sera appliquée;
- la consommation de carburant : un véhicule est loué avec le plein de carburant. Selon le niveau de carburant restant (vide, ¼ du réservoir, ½ du réservoir, ¾ du réservoir), un supplément sera appliqué ;
- l'état du véhicule : si aucune assurance n'a été souscrite par le client et que le véhicule est endommagé, le client aura à payer des frais de remise en état du véhicule, indiqués sur la facture. Il vous appartient de définir la manière dont seront stockées les informations sur l'état du véhicule.

Les véhicules sont à prendre ou à rendre dans toute agence de la société RentCar. Chaque agence peut prendre un nombre de véhicules maximum. Pour éviter la saturation des emplacements, des agents sont chargé de déposer ou récupérer des véhicules des différentes agences et ainsi maintenir la disponibilité des véhicules en location et des places libres pour le retour des véhicules loués.

Gestion des Ressources

RentCar doit proposer tous les types de recherche et d'affichage suivant pour les ressources :

- Liste des véhicules disponibles par catégorie ;
- Recherche par marque ;
- Recherche de tous les véhicules en cours de location ;

RentCar doit pouvoir insérer, supprimer ou modifier une ressource.

Gestion des clients

RentCar doit proposer tous les types de recherche et d'affichage suivant pour les clients :

- Liste alphabétique ;
- Recherche par nom ;
- Recherche de tous les clients ayant une location en cours ;
- Recherche de tous les clients ayant loués un véhicule donné.

RentCar doit pouvoir insérer, supprimer ou modifier les informations des clients.

Elle doit aussi permettre de visualiser des informations liées à la performance :

- afficher les clients gold qui ont effectué le plus de locations (le montant total de leurs locations dépasse 90% du plus grand montant total généré par client sur l'ensemble de ses locations)
- afficher les client qui n'ont jamais utilisé le service de location
- afficher les emplacements dont le nombre de véhicules disponibles à un instant donné est à plus de 80% de leur capacité.
- Afficher le montant global des recettes de l'entreprise de gestion en euros pour les locations entre une date donnée X et une autre date donnée Y (On considère les recettes générées par les locations effectuées entre les 2 dates).

IV – Fonctionnalités bonus

Ce défi rapportera **+3 pts sur la note finale** aux membres du groupe
Seule condition : **Toutes** les fonctionnalités de base doivent être mises en œuvre.

- ✓ Login avec trois niveaux de privilèges : **administrateur, Agent** et **utilisateur**.
  - L'administrateur aura des droits complets de mise à jour
  - L'Agent peut récupérer le circuit à réaliser pour la récupération et le dépôt des véhicules dans les emplacements prévus.
  - 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 à la base de données (Threads, concurrence d'accès, transactions).

Consignes d'organisation

- S'organiser en groupes de 2 membres minimum et 3 membres au maximum par groupe (liste à finaliser à la fin de la séance du Tp3)
- Validation partielle de votre avancement à la fin de la dernière séance Projet.
- Le rendu sur campus doit se faire avant le lundi 31 mai 2021 à 23h59.

Rendus attendus

1) Au plus tard, **lundi 31/05/2021** à **23h59**, vous devez déposer vos livrables dans un seul fichier zip, sur Moodle (Projet source Java + scripts SQL + Rapport) dans l'espace réservé pour votre groupe.
2) Tous les fichiers source et documents doivent contenir les noms des membres du groupe en en-tête.
3) Les livrables seront importés sur votre projet dans Eclipse ou IntelliJ.