

Localux

Présentation du contexte

L'organisation cliente : l'entreprise de location de véhicules Localux

Localux est spécialisée dans la location de véhicules haut de gamme et de luxe, avec chauffeur. La société propose des courses sur Paris et sa périphérie avec des trajets prédéfinis (d'un lieu de départ à un lieu de destination) comme, par exemple, Paris-Versailles, Paris-Deauville, Paris-Fontainebleau, Paris-Chantilly.

Dans une démarche qualité, Localux veille à respecter les engagements suivants :

- Offrir une gamme de véhicule vert : électriques ou hybrides
- Proposer une réservation rapide et simple 24/24 et 7j/7, par téléphone ou sur le site
- Offrir la possibilité de réserver un véhicule pour un départ immédiat ou pour une date et une heure précise
- Fournir des chauffeurs professionnels en adéquation avec la qualité du service
- Veiller au respect des règles de sécurité en vigueur
- Protéger les données bancaires et personnelles et ne jamais les divulguer à un tiers

Pour accompagner son activité, Localux dispose :

- du site «www.localuxhubert.com» qui permet la consultation de l'offre et la réservation d'un véhicule avec chauffeur
- de l'application «GestActivité», écrite en Java, qui permet aux salariés de Localux de suivre l'activité de l'entreprise. Elle propose notamment les modules "consultation des caractéristiques des locations", "gestion du parc automobile" et "statistiques sur les locations" ;
- d'une base de données qui héberge toutes les données nécessaires à ces deux applications ;

L'entreprise prestataire de services

DevApp, entreprise de services du numérique (ESN) située en région parisienne, a développé pour Localux, le site et l'application «GestActivité».

DevApp s'occupe, par contrat, de la maintenance évolutive et corrective de cet ensemble.

Le projet

L'offre Localux évolue. L'entreprise a décidé de se lancer dans la location de voiture sans chauffeur ce qui implique des modifications de la base de données et des applications (site «www.localuxhubert.com» et application «GestActivité»). Localux confie ces évolutions à DevApp.

Vous devez intervenir dans l'équipe DevApp, affectée à l'application «GestActivité», du projet confié par Localux.

Présentation de la nouvelle activité "location sans chauffeur"

- **Les formules sans chauffeur** : plusieurs formules de location sans chauffeur seront proposées à la clientèle. Une formule donne droit à une durée de location définie (4 heures, 24 heures, 48 heures) et un nombre de kilomètres inclus forfaitairement. Tout kilomètre supplémentaire effectué par le client sera facturé à un tarif qui dépend du modèle du véhicule loué.
- **Réservation d'une location sans chauffeur** : elle sera réalisée via le site «www.localuxhubert.com». Le formulaire de réservation proposera à la clientèle de choisir la formule et le modèle de véhicule souhaités. Un véhicule, correspondant au modèle choisi par la personne cliente, sera affecté automatiquement. Lorsque la personne cliente aura fourni les renseignements nécessaires au dossier de location, elle devra procéder au règlement.
- **Retrait du véhicule loué** : lors du retrait du véhicule, un dépôt de garantie sera demandé dont le montant dépend du modèle du véhicule et la personne cliente devra signer un document indiquant qu'elle a pris connaissance de l'état du véhicule loué. Elle pourra souscrire une assurance lui offrant des garanties supplémentaires en cas de dommages sur le véhicule.
- **Restitution du véhicule loué** : lorsque la personne cliente restituera le véhicule loué, Localux contrôlera le véhicule pour savoir s'il a ou non subi des dommages. Par ailleurs, le nombre de kilomètres indiqué au compteur sera relevé et comparé au nombre de kilomètres inclus dans la formule : si le forfait kilométrique est dépassé, le montant dû au titre du dépassement devra être réglé.

Pour l'instant, l'équipe DevApp chargée de l'application Web a modifié :

- La base de données afin qu'elle intègre les données indispensables à la consultation et la réservation de véhicule sans chauffeur
- Les deux fonctionnalités de consultation et de réservation, disponibles sur le site «www.localuxhubert.com», qui sont en phase de test

Mission 1 : Retrait et restitution du véhicule loué

Documents à utiliser : 1, 2 et 3

Création d'un déclencheur (trigger)

Dans la base de données, on trouve deux types de formules :

- les formules de type "avec chauffeur" qui ont toutes la même durée de location : 24h ;
- les formules de type "sans chauffeur" qui permettent de louer un véhicule pour une durée exprimée en heures : 4 heures, 24 heures, 48 heures.

Un déclencheur trgDateRetourLAC, fourni dans le dossier documentaire, existe déjà. Il permet de mettre à jour la date et heure de fin de location (colonne dateHreRetourPrevu de la table Location) pour une location avec chauffeur. Il est exécuté lors de l'ajout d'une location avec chauffeur.

Un déclencheur trgDateRetourLSC doit être ajouté pour mettre à jour la colonne date et heure de fin de location de la table Location pour une location sans chauffeur.

Question 1.1

Écrire le déclencheur trgDateRetourLSC. Pour cela, vous utiliserez la partie de la base de données entourée dans le document 1.

Statistiques sur les locations sans chauffeur

Le module "statistique sur les locations" de l'application «GestActivité» doit être modifié pour obtenir des informations sur l'activité location sans chauffeur. De nouvelles statistiques vont permettre d'anticiper le renouvellement de la flotte automobile de **Localux**.

Pour les locations sans chauffeur, **Localux** veut obtenir les modèles les plus demandés.

Votre chef de projet vous demande de réaliser une requête qui permette d'obtenir le nombre de locations sans chauffeur pour chaque nom de modèle, avec les modèles les plus demandés en tête de classement.

Question 1.2

Écrire la requête répondant au besoin exprimé par votre chef de projet.

Évolution de la structure de la base de données

Dans le cadre de la nouvelle activité de location sans chauffeur, un module permettant de gérer les **processus de retrait et de restitution du véhicule** doit être ajouté dans l'application «GestActivité». Vous devez modifier la base de données pour permettre le développement de ce module. La structure de la base de données actuellement opérationnelle vous est fournie dans le dossier documentaire.

Le retrait du véhicule

Avant d'être confié pour une location sans chauffeur, le véhicule subit un contrôle de la part d'un salarié de Localux en présence de la personne cliente : différents éléments sont vérifiés et les dommages constatés sur le véhicule sont saisis sur tablette et consignés dans l'application «GestActivité». Pour chaque élément endommagé (aile avant gauche, calandre, etc.), on conservera le degré de gravité du dommage. Un rapport d'état du véhicule avant la location sera édité par Localux et signé par la personne cliente pour approbation.

Lors du retrait du véhicule, Localux propose à la personne cliente de souscrire une assurance. Cette souscription est facultative. En cas de vol ou de dommages sur le véhicule pendant la location, les garanties prises en charge et les montants de franchise dépendront donc de l'assurance souscrite.

Il y a plusieurs garanties possibles (vol, dommage, etc.). Pour chaque garantie, il existe une franchise de base. Une franchise est la somme maximum qui reste à la charge de la personne cliente, si celle-ci est déclarée responsable au regard de la garantie concernée (vol, dommage, etc.) ou s'il n'existe pas de recours contre un tiers identifié.

Chaque assurance porte un nom, une description et propose différentes garanties. Pour chaque garantie d'une assurance complémentaire, un taux de réduction compris entre 0 et 100% est appliqué sur la franchise de base en cas de dommage.

Voici trois exemples d'assurance :

- l'assurance «essentielle» propose
 - la garantie vol avec la franchise de base.
- L'assurance «confort» propose
 - la garantie vol avec une réduction de 80 % de la franchise de base,
 - la garantie dommage avec une réduction de 50 % de la franchise de base,
 - la garantie bris de glace avec la franchise de base.
- L'assurance «rachat de franchise de base» propose
 - la garantie vol avec une réduction de 100 % de la franchise de base,
 - la garantie dommage avec une réduction de 100 % de la franchise de base.
-

Lors du retrait du véhicule, la personne cliente doit verser un dépôt de garantie qui dépend du modèle de véhicule loué.

La restitution totale ou partielle de ce montant dépendra des dommages constatés lors de sa restitution et de l'assurance souscrite par la personne cliente.

La restitution du véhicule

L'employé de Localux qui réceptionnera le véhicule à la fin de la location sans chauffeur devra vérifier son état. L'application «GestActivité» sera utilisée pour pouvoir enregistrer les différents dommages et leur degré de gravité constatés lors de la restitution du véhicule. Un rapport d'état du véhicule après la location sera édité par Localux et signé par la personne cliente pour approbation.

La différence entre l'état avant la location et après la location sert à déterminer si des dommages sont à imputer à la location. Dans ce cas, l'employé qui réceptionne le véhicule détermine un coût estimatif des réparations en fonction d'une matrice de dommages établie par Localux et ce coût estimé est enregistré au niveau de la location. La gestion de la matrice des dommages sort du cadre de cette mission.

Question 1.3

Selon le formalisme de votre choix, adapter la structure de la base de données existante pour prendre en compte les données du nouveau module.

Important : ne présenter que les éléments nécessaires à la mission.

Mission 2 : Gestion du dépassement du forfait kilométrique

Documents à utiliser : 4, 5, 6, 7 et 8

La nouvelle activité de réservation de véhicule sans chauffeur nécessite une modification de l'application «GestActivité» pour permettre de gérer le dépassement du forfait kilométrique lors de la restitution du véhicule.

Ce nouveau module permettra à l'employé de Localux en charge de la restitution du véhicule de saisir le nombre total de kilomètres apparaissant au compteur afin d'obtenir le surplus à régler par la personne cliente en cas de dépassement du forfait kilométrique.

Un packaging de classes techniques permet d'accéder à la base de données et l'application utilise des classes métiers présentées dans le dossier documentaire.

Pour l'instant les 3 formules ci-dessous sont proposées sur le site de réservation, mais d'autres formules sont à l'étude pour l'année 2023 :

- la formule "Forfait 4 h" donne droit à 4 h pour un forfait (kilomètres inclus) de 150 kilomètres ;
- la formule "Forfait 24 h" donne droit à 24 h pour un forfait de 300 kilomètres ;
- la formule "Forfait 48 h" donne droit à 48 h pour un forfait de 700 kilomètres.

Au-delà du forfait kilométrique, tout kilomètre supplémentaire effectué sera facturé à un tarif dépendant du modèle du véhicule loué. Par exemple, pour la location d'une Renault Espace, tout kilomètre supplémentaire sera facturé 3 euros. En revanche, pour une Peugeot 5800, le kilomètre supplémentaire sera facturé 4 euros.

Si une personne cliente loue une Renault Espace avec la formule 24 h, elle devra régler 240 euros lors de la réservation sur le site «www.localuxhubert.com». Si elle effectue 400 kilomètres, elle devra donc régler 300 euros au titre du dépassement du forfait kilométrique lorsqu'elle restituera le véhicule. En revanche, si elle n'a pas dépassé le forfait kilométrique, elle n'aura rien à régler lors de la restitution.

Question 2.1

Réaliser le code du constructeur de la classe métier LocationSansChauffeur. Le kilométrage au compteur lors de la restitution du véhicule sera initialisé à 0.

Question 2.2

Réaliser le code de la méthode GetMontantDepasForfait() de la classe métier LocationSansChauffeur qui permet d'obtenir le montant à régler par la personne cliente en cas de dépassement du forfait kilométrique.

Des tests unitaires sont nécessaires pour valider chaque méthode. Ceux-ci doivent donc être réalisés pour tester la méthode `GetMontantDepasForfait()`.

Une ébauche d'un test, fourni dans le dossier documentaire, a été réalisée pour tester cette méthode. Il porte sur une location sans chauffeur, réalisée le 15 février 2018 et concerne une Renault Espace immatriculée LA-039-LP, achetée le 28 décembre 2016. La personne cliente a choisi la formule "Forfait 24h" et a réglé 240 euros. Lors du retrait du véhicule, le compteur indique 25000 kilomètres, et lors de sa restitution, il comptabilise 25400 kilomètres.

Question 2.3

Compléter le code du test unitaire de la méthode `GetMontantDepasForfait()`.

Dans le cadre de la mise en place de la nouvelle activité de réservation de véhicule sans chauffeur, monsieur Berthu, directeur du service commercial de Localux, souhaite obtenir le montant total des sommes perçues suite à dépassement du forfait kilométrique pour chaque véhicule. En analysant ces informations, il pourra revoir à la hausse ou à la baisse le prix du kilomètre supplémentaire de certains véhicules.

Votre chef de projet a rédigé un courriel, fourni dans le dossier documentaire, qui présente les modifications à apporter pour répondre au besoin de monsieur Berthu.

Question 2.4

Afin de répondre au besoin de monsieur Berthu :

- a) **apporter les modifications nécessaires au diagramme de classes (seules les classes concernées par les modifications devront être présentées) ;**
- b) **coder les modifications à apporter à la classe `Vehicule`.**

Mission 3 : Location sans chauffeur

Après avoir réalisé les missions une et deux, vous proposerez une implémentation de la fonctionnalité de location sans chauffeur.

Vous prendrez un plus grand soin à fournir un jeu d'essais cohérent (données).

Vous réaliserez l'ensemble des développements indispensables au fonctionnement de l'application.

Lors de la livraison du projet opérationnel et testé, vous fournirez une documentation technique.

Dossier documentaire

Document 1 : Structure de la base de données

Diagramme de classes :

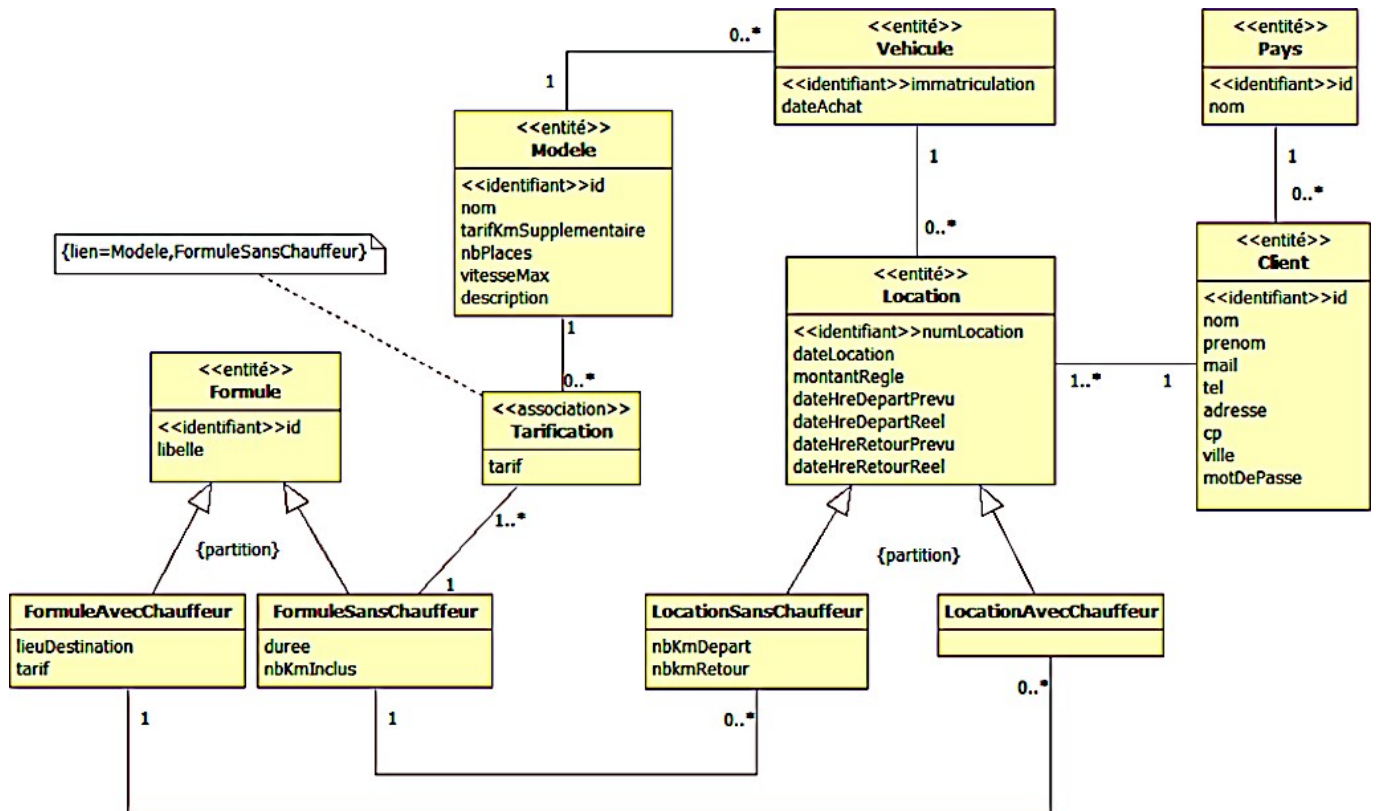


Schéma relationnel

Vehicule (immatriculation, dateAchat, idModele)

Clé primaire : immatriculation

Clé étrangère : idModele en référence à id de Modele

Modele (id, nom, tarifKmSupplementaire, nbPlaces, vitesseMax, description)

Clé primaire : id

EtreProposee (idModele, idFormule, tarif)

Clé primaire : idModele, idFormule

Clés étrangères : idModele en référence à id de Modele
idFormule en référence à id de Formule

Formule (id, libelle)

Clé primaire : id

FormuleAvecChauffeur (id, lieuDestination, tarif)

Clé primaire : id

Clé étrangère : id en référence à id de Formule

FormuleSansChauffeur (id, duree, nbKmInclus)

Clé primaire : id

Clé étrangère : id en référence à id de Formule

Location (numLocation, dateLocation, montantRegle, dateHreDepartPrevu, dateHreDepartReel, dateHreRetourPrevu, dateHreRetourReel, idClient, immatriculation)

Clé primaire : numLocation

Clés étrangères : idClient en référence à id de Client

immatriculation en référence à immatriculation de Vehicule

LocationSansChauffeur (numLocation, nbKmDepart, nbKmRetour, idFormule)

Clé primaire : numLocation

Clés étrangères : numLocation en référence à numLocation de Location

idFormule en référence à id de FormuleSansChauffeur

LocationAvecChauffeur (numLocation, idFormule)

Clé primaire : numLocation

Clés étrangères : numLocation en référence à numLocation de Location

idFormule en référence à id de FormuleAvecChauffeur

Pays (id, nom)

Clé primaire : id

Client (id, nom, prenom, mail, tel, adresse, cp, ville, motDePasse, idPays)

Clé primaire : id

Clé étrangère : idPays en référence à id de Pays

Remarques :

- Toutes les colonnes contenant des dates (dont le nom commence par "date") sont de type horodatage
- La colonne duree dans la table FormuleSansChauffeur est exprimée en heures entières

Document 2 : Déclencheur trgDateRetourLAC

CREATE TRIGGER trgDateRetourLAC BEFORE INSERT ON LocationAvecChauffeur
FOR EACH ROW

BEGIN

DECLARE duree INTEGER ;

DECLARE dateDepart DATETIME;

DECLARE dateRetour DATETIME;

/* la durée est de 24h pour une location avec chauffeur */

SET duree = 24;

/* récupération de la date de départ dans une variable */

SET dateDepart = (SELECT dateHreDepartPrevu FROM Location

WHERE numLocation = NEW. numLocation) ;

SET dateRetour = DATE_ADD(dateDepart, INTERVAL duree HOUR) ;

UPDATE Location set dateHreRetourPrevu = dateRetour

where numLocation = NEW. numLocation ;

END ;

Remarque : la fonction **DATE_ADD**(date, **INTERVAL** nb **unitéDeTemps**) ajoute nb unités de temps à la date passée en paramètre (nb pouvant être une valeur ou le nom d'une variable)

Exemples de valeurs possibles pour **unitéDeTemps** : SECOND, HOUR, DAY, etc.

Document 3 : Rapport établi lors de la restitution d'un véhicule

Numéro de location : 240
Modèle : Lamborghini Huracán LP 610-4 Spyder
Immatriculation : LA – 038 – SP

Numéro de client : 56
Nom et prénom du client : Agathe MORIN

Retrait du véhicule

Contrôle n° 320 réalisé par **Marc BOTUR**

Kilométrage	Date - Heure
12 480	13/04/2018 08:30

Dommages constatés sur le véhicule

	RS	RP	EC		RS	RP	EC
Aile AV G.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Aile AV D.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aile AR G.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Aile AR D.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Calandre	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Phare AV G.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Phare AV D.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Siège cond.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Siège pass.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tableau de b.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Porte AV G.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Porte AV D.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
...				...			

Observations :

L'intérieur de la voiture est poussiéreux.

Restitution du véhicule

Contrôle n°436 réalisé par **Aline LIDBO**

Kilométrage	Date - Heure
12 760	14/04/2018 9:15

Dommages constatés sur le véhicule

	RS	RP	EC		RS	RP	E
Aile AV G.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Aile AV D.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Aile AR G.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Aile AR D.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Calandre	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Phare AV G.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Phare AV D.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Siège cond.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Siège pass.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tableau de b.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Porte AV G.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Porte AV D.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
...				...			

Observations :

À plusieurs reprises, nous avons entendu des bruits anormaux lors du freinage.

RS : Rayure superficielle ; RP : Rayure profonde ; EC : Enfoncement/Choc

Coût estimé des réparations : 2 000 euros

Signature du client :

Document 5 : Classes métier implémentées

Les classes métier sont dans le projet jlocalux dans le package "model".

Document 6 : Méthode de test unitaire à compléter

Les classes de test sont dans le projet jlocalux, "Test Packages".

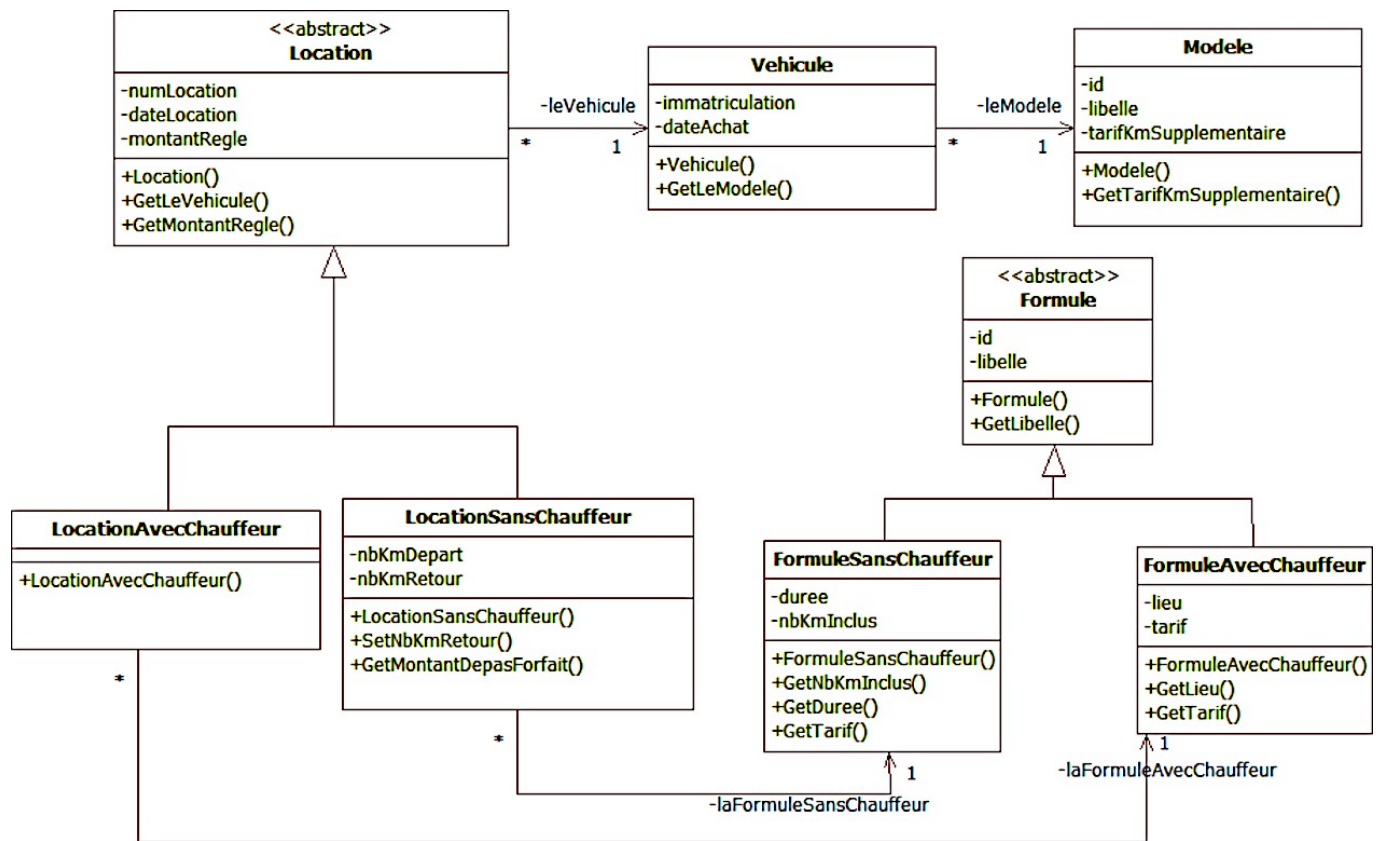
La classe Assert contient différentes méthodes permettant de réaliser les tests unitaires.

Elle contient, notamment, la méthode **assertEquals** dont voici la signature à utiliser ici :

```
public static void assertEquals(double résultatAttendu, double résultatCalculé, double margeDErreur)
```

- *résultatAttendu* : valeur de type double contenant la valeur attendue
- *résultatCalculé* : valeur de type double contenant la valeur obtenue
- *margeDErreur* : écart acceptable en les deux nombres

Document 4 : Classes métier implémentées



Document 7 : Mail de la cheffe de projet

From Sophie.Louvin@devapp.net

To: Alix.Pilou@devapp.net

Subject : modification de GestActivité pour répondre aux besoins de M. Berthu

Bonjour Alix,

Tu dois modifier la classe Vehicule pour y ajouter une méthode nommée GetTotalDepasForfait qui retournera le montant total des sommes versées suite à dépassement du forfait kilométrique pour toutes les locations sans chauffeur réalisées avec le véhicule courant. Tu devras ajouter un nouvel attribut dans cette classe qui permettra d'accéder à toutes les instances LocationSansChauffeur réalisées avec le véhicule et donc modifier le constructeur de la classe. Evidemment, tu ajouteras également dans la classe Vehicule une méthode qui permettra d'ajouter une nouvelle location sans chauffeur au véhicule. Bon courage et n'hésite pas à me contacter si tu as besoin d'un complément d'information.

Sophie Louvin