01/04/2022

Deconninck Corentin & Hallet Antoine

Projet Rent A Car

Cahier de charge

Table des matières

[Contexte 2](#_Toc96095352)

[Cadre 2](#_Toc96095353)

[Projet 2](#_Toc96095354)

[Analyse métier 2](#_Toc96095355)

[Description de la solution 2](#_Toc96095356)

[Composition du projet 3](#_Toc96095357)

[Fonction Attendues 3](#_Toc96095358)

[Exigences fonctionnelles 3](#_Toc96095359)

# Contexte

## Cadre

Etudiant en informatique de gestion en promotion sociale à l’EPHEC, il nous a été demandé, dans le cadre du cours de « *Projet développement SGDB* », de réaliser un projet informatique portant sur la gestion d’une société de location de voiture. Le but du projet est de pouvoir implémenter une base de données ainsi qu’une application qui communique avec celle-ci afin de gérer la société. La cohérence business de l’application n’est pas l’objet de l’exercice.

## Projet

Le projet consiste à créer un programme de location de voiture. Ce programme doit permette à la société de gérer son parc automobile et doit permettre à un client de faire une réservation. Afin de mener à bien ce projet, nous nous somme basés sur les règles business écrites par Mr. Fievez ainsi que sur des postulats que nous avons établis.

* Enoncé du projet :
* Postulats sur le projet :

# Analyse métier

## Description de la solution

La solution permettra au gérant de la société de location de voiture de gérer son parc automobile et de louer ses voitures à des clients.

D’un point de vue gérant, il pourra :

1. Gérer son parc automobile (Créer une voiture, modifier une voiture, supprimer une voiture)
2. Gérer ses dépôts (Créer un dépôt, supprimer un dépôt, assigner un véhicule à un dépôt)
3. Gérer la notoriété de ses véhicules (créer une notoriété, supprimer une notoriété, assigner une notoriété à un véhicule)
4. Gérer ses forfaits (créer un forfait, supprimer un forfait)
5. Gérer ses prix (créer un prix, modifier un prix)
6. Gérer les réservations (visualiser des réservations, modifier le dépôt de retour s’il est vide)

D’un point de vue client, il pourra :

1. Se créer (se créer, se modifier)
2. Faire une réservation (Choisir un dépôt de départ, choisir un véhicule disponible dans ce dépôt, choisir un forfait lié à ce dépôt ou non)
3. Modifier une réservation (Si elle n’est pas en cours)
4. Annuler une réservation (Si elle n’est pas en cours)

### Composition du projet

Le projet sera décomposé en 4 parties :

1. Cahier de charge
2. Base de données
3. Application
4. Interface utilisateur (MVC)

## Les intervenants

## Fonction Attendues

### Exigences fonctionnelles

Fonctionnalités obligatoires pour le loueur

|  |  |
| --- | --- |
| Code | Description |
| EF-LO-001 | Créer un véhicule |
| EF-LO-002 | Modifier un véhicule |
| EF-LO-003 | Supprimer un véhicule |
| EF-LO-004 | Créer un dépôt |
| EF-LO-005 | Supprimer un dépôt |
| EF-LO-006 | Assigner un véhicule à un dépôt |
| EF-LO-007 | Créer une notoriété |
| EF-LO-008 | Supprimer une notoriété |
| EF-LO-009 | Assigner une notoriété à un véhicule |
| EF-LO-010 | Créer un forfait |
| EF-LO-011 | Supprimer un forfait |
| EF-LO-012 | Créer un prix |
| EF-LO-013 | Supprimer un prix |
| EF-LO-014 | Visualiser une réservation |
| EF-LO-015 | Modifier une réservation |
| EF-LO-016 |  |

Fonctionnalités obligatoires pour le client

|  |  |
| --- | --- |
| Code | Description |
| EF-CL-001 | Créer un client |
| EF-CL-002 | Modifier un client |
| EF-CL-003 | Créer une réservation |
| EF-CL-004 | Modifier une réservation |
| EF-CL-003 | Annuler une réservation |
|  |  |

### Exigences non-fonctionnelles

# Contraintes fonctionnelles

## Règles de structure

Client

|  |  |
| --- | --- |
| Code | Description |
| CF-RS-001 | Un client possède un ID unique |
| *DB > Table Client > Keys > PK\_Client : Clé primaire sur IDClient* |
| CF-RS-002 | Un client possède un nom obligatoire |
| CF-RS-003 | Un client possède un prénom obligatoire |
| CF-RS-004 | Un client possède une adresse mail unique obligatoire |
| *DB > Table Client > keys > UK\_Mail : l’adresse mail est unique* |
| CF-RS-005 | Un client peut avoir 0 à plusieurs réservations |

Dépôt

|  |  |
| --- | --- |
| Code | Description |
| CF-RS-006 | Un dépôt possède un ID unique |
| *DB > Table Depot > Keys > PK\_Depot: Clé primaire sur IDDepot* |
| CF-RS-007 | Un dépôt possède un champ statut inactif |
| CF-RS-008 | Un Dépôt fait partie d’une et une seule ville |
| *DB > Table Depot > Keys FK\_Depot\_Ville : Clé étrangère sur IDVille, pointant sur IDVille de la table Ville* |
| CF-RS-010 | Un dépôt peut contenir de 0 à plusieurs voitures |

Forfait

|  |  |
| --- | --- |
| Code | Description |
| CF-RS-011 | Un forfait possède un ID unique |
| *DB > Table Forfait > Keys > PK\_Forfait: Clé primaire sur IDForfait* |
| CF-RS-012 | Un forfait a deux ID dépôt obligatoire |
| *DB > Table Forfait > Keys > FK\_IDDepot1 et FK\_IDDepot2 : Clés étrangères sur IDDepot1 et IDDepot2, pointant sur IDDepot de la table Depot* |
| CF-RS-013 | Un forfait possède un prix obligatoire |
| CF-RS-014 | Un forfait possède une date de début obligatoire |
| CF-RS-015 | Un forfait peut avoir une date de fin |

Notoriété

|  |  |
| --- | --- |
| Code | Description |
| CF-RS-016 | Une notoriété possède un ID unique |
| *DB > Table Notoriete > Key > PK\_Notoriete : Clé primaire sur IDNotoriete* |
| CF-RS-017 | Une notoriété possède un libellé obligatoire |
| CF-RS-018 | Une notoriété possède un coefficient multiplicateur du prix obligatoire |
| CF-RS-019 | Une notoriété possède un champ statut inactif |
| CF-RS-020 | Une notoriété peut être liée à 0 ou plusieurs véhicules |

Voiture

|  |  |
| --- | --- |
| Code | Description |
| CF-RS-021 | Une voiture possède un ID unique |
| *DB > Table Voiture > Keys> PK\_Voiture > Clé primaire sur IDVoiture* |
| CF-RS-022 | Une voiture possède une marque obligatoire |
| CF-RS-023 | Une voiture doit être contenue dans un et un seul dépôt |
| *DB > Table Voiture > Keys > FK\_Voiture\_Depot> Clé étrangère sur IDDepot, pointant sur IDDepot de la table Depot* |
| CF-RS-024 | Une voiture possède une plaque d’immatriculation obligatoire et unique |
| CF-RS-025 | *DB > Table Voiture > Keys > UK\_Immatriculation > Clé unique sur Immatriculation* |
| CF-RS-026 | Une voiture possède une et une seule notoriété |
| CF-RS-027 | *DB > Table Voiture > Keys > FK\_Voiture\_Notoriete > Clé étrangère sur IDNotoriete, pointant sur IDNotoriete de la table Notoriete* |
| CF-RS-028 | Une voiture possède un champs statut inactif |
| CF-RS-029 | Une voiture peut être assignée à de 0 à 1 réservation |

Ville

|  |  |
| --- | --- |
| Code | Description |
| CF-RS-030 | Une ville possède un ID unique |
| *DB > Table Ville> Keys > PK\_ Ville : Clé primaire sur IDVille* |
| CF-RS-031 | Une ville possède un nom obligatoire |
| CF-RS-032 | Une ville appartient à un et un seul pays |
| *DB > Table Ville > Keys > FK\_Ville\_Pays : Clé étrangère sur IDPays, pointant sur IDPays de la table Pays* |
| CF-RS-033 | Une ville peut contenir de 0 à 1 dépôt |

Pays

|  |  |
| --- | --- |
| Code | Description |
| CF-RS-034 | Un pays possède un ID unique |
| *DB > Table Pays > Keys > PK\_ Pays : Clé primaire sur IDPays* |
| CF-RS-035 | Un pays possède une référence de prix obligatoire |
| CF-RS-036 | Un pays possède un nom obligatoire |
| CF-RS-037 | Un pays peut contenir de 0 à plusieurs villes |

Prix

|  |  |
| --- | --- |
| Code | Description |
| CF-RS-038 | Un prix possède une référence de prix obligatoire |
| CF-RS-039 | Un prix possède une valeur (Prix/Km) obligatoire |
| CF-RS-040 | Un prix possède une date de début obligatoire |
| CF-RS-041 | Un prix peut avoir une date de fin |

Réservation

|  |  |
| --- | --- |
| Code | Description |
| CF-RS-042 | Une réservation possède un ID unique |
| *DB > Table Reservation > Keys > PK\_ Reservation : Clé primaire sur IDReservation* |
| CF-RS-043 | Une réservation concerne une et une seule voiture |
| *DB > Table Reservation > Keys > FK\_Reservation\_Voiture : Clé étrangère sur IDVoiture, pointant sur IDVoiture de la table voiture* |
| CF-RS-44 | Une réservation concerne un et un seul client |
| *DB > Table Reservation > Keys > FK\_Reservation\_Client : Clé étrangère sur IDClient, pointant sur IDClient de la table client* |
| CF-RS-45 | Une réservation possède un et un seul dépôt de départ |
| *DB > Table Reservation > Keys > FK\_Reservation\_DepotDepart : Clé étrangère sur IDDepot\_Depart, pointant sur IDDepot de la table Depot* |
| CF-RS-46 | Une réservation peut avoir un dépôt de retour |
| *DB > Table Reservation > Keys > FK\_Reservation\_DepotRetour: Clé étrangère sur IDDepot\_Retour, pointant sur IDDepot de la table Depot* |
| CF-RS-47 | Une réservation peut être liée à un forfait |
| *DB > Table Reservation > Keys > FK\_Reservation\_Forfait : Clé étrangère sur IDForfait, pointant sur IDForfait de la table Forfait* |
| CF-RS-48 | Une réservation possède une date de réservation obligatoire |
| CF-RS-49 | Une réservation possède une date de départ obligatoire |
| CF-RS-50 | Une réservation possède une date de retour |
| CF-RS-51 | Une réservation possède un kilométrage au départ |
| CF-RS-52 | Une réservation possède un kilométrage au retour |
| CF-RS-53 | Une réservation possède un coefficient multiplicateur du prix |