01/04/2022

Deconninck Corentin & Hallet Antoine

Projet Rent A Car

Cahier de charge

Table des matières

[Contexte 1](#_Toc101063622)

[Cadre 1](#_Toc101063623)

[Projet 2](#_Toc101063624)

[Analyse métier 2](#_Toc101063625)

[Description de la solution 2](#_Toc101063626)

[Les intervenants 3](#_Toc101063627)

[Fonction Attendues 3](#_Toc101063628)

[~~Exigences non-fonctionnelles~~ 4](#_Toc101063629)

[Contraintes Business 4](#_Toc101063630)

[Analyse fonctionnelle 4](#_Toc101063631)

[Diagramme des cas d’utilisation 5](#_Toc101063632)

[Description textuelle des cas d’utilisation 5](#_Toc101063633)

[Glossaire 5](#_Toc101063634)

[**Interface Client** 5](#_Toc101063635)

[**Interface Loueur** 8](#_Toc101063636)

[Contraintes fonctionnelles 22](#_Toc101063637)

[Règles de structure 22](#_Toc101063638)

[Règles de validation 24](#_Toc101063639)

[Règles de calcul 26](#_Toc101063640)

[Schéma relationnel de la solution 27](#_Toc101063641)

[Schéma Entité-Association 27](#_Toc101063642)

[Schéma relationnel 27](#_Toc101063643)

[Implémentation des contraintes 28](#_Toc101063644)

[Analyse Technique 28](#_Toc101063645)

[Technologie proposée 28](#_Toc101063646)

[~~Architecture applicative~~ 29](#_Toc101063647)

[~~Algorithmes spécifiques~~ 29](#_Toc101063648)

[Annexes 30](#_Toc101063649)

[Enoncé 30](#_Toc101063650)

Contexte

Cadre

Etudiants 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*** : L’énoncé du projet est présent dans les annexes de ce document.
* ***Postulats sur le projet*** : Afin dé réaliser ce projet en fonction des exigences attendues et des règles business mentionnées, nous avons réalisé différents postulats :
  + Le programme ne prend pas en charge le rapatriement des véhicules entre les différents dépôts.
  + Le programme ne présente pas de fonctionnalités d’enregistrement de comptes utilisateurs et d’authentification, ni pour la partie « Loueur », ni pour la partie « Client ». Nous avons cependant simulé un système fictif et simplifié de comptes utilisateurs pour les clients, ceci afin de garantir la cohérence du projet et de faciliter l’utilisation de l’interface coté « client ».

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.

Du point de vue du gérant, celui-ci pourra :

1. Gérer son parc automobile (Créer une voiture, modifier une voiture, désactiver une voiture)
2. Gérer ses dépôts (Créer un dépôt, désactiver un dépôt, assigner un véhicule à un dépôt)
3. Gérer la notoriété de ses véhicules (créer une notoriété, désactiver une notoriété, assigner une notoriété à un véhicule)
4. Gérer ses forfaits (créer un forfait entre deux dépôts, modifier le prix d’un forfait, désactiver un forfait)
5. Gérer le prix de location au kilomètre (créer un prix, modifier un prix)
6. Gérer les réservations (visualiser des réservations, enregistrer le début effectif d’une location, clôturer une location)
7. Visualiser la facturation pour les réservations clôturées

Du point de vue du client, celui-ci pourra :

1. Créer un compte utilisateur
2. Faire une réservation (Choisir un dépôt de départ et de retour, choisir un véhicule disponible dans ce dépôt, choisir un forfait lié à ce dépôt ou le prix/km)
3. Visualiser l’historique de ses réservations
4. Annuler une réservation (Si elle n’est pas en cours)
5. Visualiser la facturation de ses réservations clôturée

Composition du projet

* Le projet sera décomposé en 4 parties :

1. Cahier de charge
2. Base de données
3. Application
   1. Back-end (API)
   2. Interface utilisateur (MVC)

* Le projet sera remis sous la forme suivante :
  + 1 Solution contenant :
    - 4 projets bibliothèque de classes (DLL)
      * 1 Projet Data Access Layer (DAL)
      * 1 Projet Business Layer (BL)
      * 1 Projet Modèles
      * 1 Projet contrôleurs API
    - 1 Projet MVC, lequel comportant :
      * Un répertoire pour les modèles
      * Un répertoire pour les contrôleurs
      * Un répertoire pour les vues
      * Une Classe dédiée pour la gestion des erreurs
  + 1 Script de base de données en SQL Server

Les intervenants

* Les utilisateurs « loueurs » :

Il s’agit des employés de la société de location de voiture. Ils ont accès à une interface dédiée « loueur » pour la gestion des voitures, des notoriétés, des dépôts, des forfaits et des différents prix/km. L’interface permet en outre la visualisation des réservations et des factures, et permet à l’utilisateur l’enregistrement du départ d’un véhicule pour une location, et l’enregistrement du retour du véhicule afin de clôturer une location.

* Les utilisateurs « clients » :

Il s’agit des clients désirant réserver une voiture pour une location. Ceux-ci ont accès à une interface dédiée permettant d’effectuer une réservation, d’annuler une réservation et de visualiser leur historique de réservation.

Fonction Attendues

Exigences fonctionnelles

Fonctionnalités obligatoires pour le loueur

|  |  |
| --- | --- |
| Code | Description |
| EF-LO-001 | Créer une notoriété |
| EF-LO-002 | Modifier une notoriété |
| EF-LO-003 | Désactiver une notoriété |
| EF-LO-004 | Créer véhicule |
| EF-LO-005 | Modifier un véhicule |
| EF-LO-006 | Désactiver un véhicule |
| EF-LO-007 | Créer un Dépôt |
| EF-LO-008 | Modifier un Dépôt |
| EF-LO-009 | Désactiver un Dépôt |
| EF-LO-010 | Créer un forfait |
| EF-LO-011 | Modifier un forfait |
| EF-LO-012 | Désactiver un forfait |
| EF-LO-013 | Créer une ville |
| EF-LO-014 | Modifier une ville |
| EF-LO-015 | Supprimer une ville |
| EF-LO-016 | Créer un Pays |
| EF-LO-017 | Modifier un pays |
| EF-LO-018 | Supprimer un pays |
| EF-LO-019 | Créer un prix |
| EF-LO-020 | Modifier un prix |
| EF-LO-021 | Démarrer une réservation |
| EF-LO-022 | Clôturer une réservation |
| EF-LO-023 | Visualiser une facturation |

Fonctionnalités obligatoires pour le client

|  |  |
| --- | --- |
| Code | Description |
| EF-CL-001 | S’enregistrer en tant que client |
| EF-CL-002 | S’identifier sur le site en tant que client |
| EF-CL-003 | Effectuer une réservation |
| EF-CL-004 | Annuler une réservation |
| EF-CL-005 | Visualiser une facturation |

### ~~Exigences non-fonctionnelles~~

Contraintes Business

Contraintes techniques

* La base de données (BD) doit être de type SQL serveur.
* L’accès aux tables de la BD ne peut pas être direct.
* L’application doit être découpée en couche : DAL, BL, Modèles, Erreurs, Interface, Web API.

~~Composition du cahier des charges~~

~~Architecture du projet à respecter~~

Analyse fonctionnelle

Diagramme des cas d’utilisation

A faire PlantUml

Description textuelle des cas d’utilisation

### Glossaire

*Sélectionner* : le terme « sélectionner » signifie que l’utilisateur clique sur un bouton ou un choix dans une liste.

*Portail :* le portail fait référence au site web de la société, lequel sert d’interface utilisateur pour l’utilisateur « Client » et l’utilisateur « Loueur ».

### **Interface Client**

#### EF-CL-001 S’enregistrer en tant que client

**Résumé**

Ce cas d’utilisation permet au client de s’inscrire en tant qu’utilisateur.

**Préconditions**

* L’utilisateur (Client) est connecté sur la page client du site de la société.

**Scénario nominal**

1. L’utilisateur sélectionne l’option « S’inscrire » et est redirigé vers une page contenant un formulaire.
2. Il indique son nom, prénom, et adresse courriel dans les champs du formulaire prévu à cet effet.
3. Il peut ensuite sélectionner « confirmer » pour confirmer son inscription.
4. L’utilisateur est redirigé vers une page lui confirmant son inscription.

**Enchainements alternatifs**

*A1 : Nom, Prénom et/ou adresse courriel invalide*

L’enchainement A1 démarre au point 2 du scénario nominal.

3. Un message d’erreur est affiché indiquant à l’utilisateur le champ du formulaire invalide

*A2 : Adresse mail déjà existante dans la base de données.*

L’enchainement A1 démarre au point 2 du scénario nominal.

3. Un message d’erreur est affiché indiquant à l’utilisateur que l’adresse courriel est déjà utilisée.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Un profil client a été créé dans la base de données.

#### ~~EF-CL-002 Modifier son compte client~~

**~~Résumé~~**

~~Ce cas d’utilisation permet au client de modifier les informations de son compte client.~~

**~~Préconditions~~**

* ~~L’utilisateur (Client) est connecté sur la page d’accueil client du portail.~~
* ~~L’utilisateur est connecté avec son compte client.~~

**~~Scénario nominal~~**

1. ~~L’utilisateur sélectionne « Modifier mes informations ». Il est redirigé vers une page contenant un formulaire.~~
2. ~~Il modifie les informations de son choix dans les champs du formulaire prévus à cet effet.~~
3. ~~Il sélectionne « Confirmer » et est redirigé vers la page d’accueil client.~~

**~~Enchainements alternatifs~~**

*~~A1 : Nom, Prénom et/ou adresse courriel invalide~~*

~~L’enchainement A1 démarre au point 2 du scénario nominal.~~

~~3. Un message d’erreur est affiché indiquant à l’utilisateur le champ du formulaire invalide~~

*~~A2 : Adresse courriel déjà existante dans la base de données.~~*

~~L’enchainement A1 démarre au point 2 du scénario nominal.~~

**~~Enchainements d’erreur~~**

~~Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.~~

**~~Postconditions~~**

* ~~Le profil client a été modifié dans la base de données.~~

#### EF-CL-002 S’identifier

**Résumé**

Ce cas d’utilisation permet au client de s’identifier avec son compte client sur le portail.

**Préconditions**

* L’utilisateur (Client) est connecté sur la page client du portail.

**Scénario nominal**

1. L’utilisateur sélectionne l’option « s’identifier ». Il est redirigé vers la page d’identification.
2. Il indique son adresse courriel dans le champ prévu à cet effet.
3. Il sélectionne « Confirmer » et est redirigé vers la page d’accueil client du portail.

**Enchainements alternatifs**

*A1 : Adresse courriel inconnue*

L’enchainement A1 démarre au point 2 du scénario nominal.

3. Aucun compte client ne correspond à l’adresse courriel mentionnée. Un message d’erreur est affiché, invitant l’utilisateur a réessayer de s’identifier.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* L’utilisateur est connecté avec son compte client

#### EF-CL-003 Effectuer une réservation

Ce cas d’utilisation permet à un client d’effectuer une réservation.

**Préconditions**

* L’utilisateur (Client) est connecté sur la page d’accueil client du portail.
* L’utilisateur est connecté avec son compte client.

**Scénario nominal**

1. Il sélectionne l’option « Nouvelle réservation » et est redirigé vers une page contenant un formulaire.
2. Il complète le formulaire
   1. Il sélectionne le pays de départ dans une liste déroulante listant les différents pays possibles.
   2. Il sélectionne la ville de départ dans une liste déroulante listant les différents dépôts de départ possibles.
   3. Il sélectionne le dépôt de retour dans une liste déroulante listant les différents dépôts de retour possible ou le Forfait Prix/Km (pas de dépôt de retour).
   4. Il indique la date de départ dans un champ prévu à cet effet.
   5. Il indique la date de retour dans un champ prévu à cet effet.
   6. Il sélectionne le véhicule dans une liste déroulante listant les véhicules disponibles (à la date de départ).
3. Une fois les différents champs du formulaire rempli, l’utilisateur peut sélectionner « confirmer » afin d’enregistrer sa réservation.
4. Il est redirigé vers la page d’accueil client du portail.

**Enchainements alternatifs**

*A1 : Le véhicule choisi n’est plus disponible au moment de la confirmation (s’il a été réservé entre-temps par un autre client*

L’enchainement A1 démarre au point 3 du scénario nominal.

4. L’utilisateur est renvoyé vers le formulaire avec un message d’erreur lui indiquant que la réservation n’a pas pu être enregistrée.

Le scénario nominal reprend au point 1.

*A2 : Les dates de retour et/ou de départ ne sont pas valides*

L’enchainement A2 démarre au point 3 du scénario nominal.

4. L’utilisateur est renvoyé vers le formulaire avec un message d’erreur lui indiquant que la réservation n’a pas pu être enregistrée.

Le scénario nominal reprend au point 1.

**Enchainements d’erreur**

*E1 : Aucun véhicule disponible pour la période choisie*

L’enchainement E1 démarre au point 2 du scénario nominal.

3. La réservation n’est pas possible. Un message d’erreur est affiché, invitant l’utilisateur à recommencer sa réservation.

**Postconditions**

* Une nouvelle réservation a été créée dans la base de données.

#### EF-CL-004 Annuler une réservation

**Résumé**

Ce cas d’utilisation permet à un client d’annuler une réservation qui n’a pas encore débuté.

**Préconditions**

* L’utilisateur (Client) est connecté sur la page d’accueil client du portail.
* L’utilisateur est connecté avec son compte client.

**Scénario nominal**

1. L’utilisateur (client) sélectionne l’option « Mon historique de réservation ». Il est redirigé vers la liste de ces réservations.
2. Il sélectionne l’option « annuler une réservation » à coté de la réservation choisie. Il est redirigé vers une page de confirmation.
3. Il sélectionne « Confirmer ». La réservation est supprimée dans la base de données.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* La réservation a été supprimée dans la base de données.

### **Interface Loueur**

#### EF-OL-001 Créer une notoriété

**Résumé**

Ce cas d’utilisation permet au loueur de créer une notoriété.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur sélectionne « Gestion des notoriétés ». Il est redirigé vers une page affichant la liste des différentes notoriétés.
2. Il sélectionne « Créer une nouvelle notoriété ». Il est redirigé vers une page affichant un formulaire.
3. Il encode les informations de la notoriété dans les champs du formulaire prévus à cet effet.
4. Il sélectionne « confirmer ». La nouvelle notoriété est créée dans la base de données.
5. Il est redirigé vers la page de visualisation des différentes notoriétés.

**Enchainements alternatifs**

*A1 : Une notoriété du même nom existe déjà*

L’enchainement A1 commence au point 3 du scénario nominal.

3. Un message d’erreur est affiché indiquant à l’utilisateur qu’une notoriété de ce nom existe déjà.

Le scénario nominal reprend au point 3.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Une nouvelle notoriété est créée dans la base de données.

#### EF-OL-002 Modifier une notoriété

**Résumé**

Ce cas d’utilisation permet au loueur de modifier une notoriété existante.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur sélectionne « Gestion des notoriétés ». Il est redirigé vers une page affichant la liste des différentes notoriétés.
2. Il sélectionne l’option « Modifier » pour la notoriété de son choix. Il est redirigé vers une page affichant un formulaire.
3. Il encode les nouvelles informations de la notoriété dans les champs du formulaire prévus à cet effet.
4. Il sélectionne « confirmer ». La notoriété est modifiée dans la base de données.
5. Il est redirigé vers la page de visualisation des différentes notoriétés.

**Enchainements alternatifs**

*A1 : Une notoriété du même nom existe déjà*

L’enchainement A1 commence au point 3 du scénario nominal.

3. Un message d’erreur est affiché indiquant à l’utilisateur qu’une notoriété de ce nom existe déjà.

Le scénario nominal reprend au point 3.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* La notoriété a été modifiée dans la base de données.

#### EF-OL-003 Désactiver une notoriété

**Résumé**

Ce cas d’utilisation permet au loueur de désactiver une notoriété.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur (Loueur) sélectionne « Gestion des notoriétés ». Il est redirigé vers une page affichant la liste des différentes notoriétés.
2. Il sélectionne l’option « Désactiver » pour la notoriété de son choix.
3. Il sélectionne « confirmer ». La notoriété est désactivée dans la base de données.
4. Il est redirigé vers la page de visualisation des différentes notoriétés.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* La notoriété a été désactivée dans la base de données.

#### EF-OL-004 Créer un véhicule

**Résumé**

Ce cas d’utilisation permet au loueur de créer un véhicule.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur sélectionne « Gestion des véhicules ». Il est redirigé vers une page affichant la liste des différents véhicules.
2. Il sélectionne « Créer un nouveau véhicule ». Il est redirigé vers une page affichant un formulaire.
3. Il encode les informations du véhicule dans les champs du formulaire prévus à cet effet.
4. Il sélectionne « confirmer ». Le nouveau véhicule est créé dans la base de données.
5. Il est redirigé vers la page de visualisation des différents véhicules.

**Enchainements alternatifs**

*A1 : Un véhicule avec la même immatriculation existe déjà*

L’enchainement A1 commence au point 3 du scénario nominal.

3. Un message d’erreur est affiché indiquant à l’utilisateur qu’un véhicule avec la même immatriculation existe déjà.

Le scénario nominal reprend au point 3.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Un nouveau véhicule est créé dans la base de données.

#### EF-OL-005 Modifier un véhicule

**Résumé**

Ce cas d’utilisation permet au loueur de modifier un véhicule existant (s’il n’est pas lié à un dépôt).

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur sélectionne « Gestion des véhicules ». Il est redirigé vers une page affichant la liste des différents véhicules.
2. Il sélectionne l’option « Modifier », « Activer » ou « Désactiver » pour le véhicule de son choix. Il est redirigé vers une page affichant un formulaire.
3. Il encode les nouvelles informations du véhicule dans les champs du formulaire prévus à cet effet.
4. Il sélectionne « confirmer ». Le véhicule est modifié dans la base de données.
5. Il est redirigé vers la page de visualisation des différentes notoriétés.

**Enchainements alternatifs**

*A1 : Un véhicule avec la même immatriculation existe déjà*

L’enchainement A1 commence au point 3 du scénario nominal.

3. Un message d’erreur est affiché indiquant à l’utilisateur qu’un véhicule avec la même immatriculation existe déjà.

Le scénario nominal reprend au point 3.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Le véhicule a été modifié dans la base de données.

#### EF-OL-006 Désactiver un véhicule

**Résumé**

Ce cas d’utilisation permet au loueur de désactiver un véhicule.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur (Loueur) sélectionne « Gestion des véhicules ». Il est redirigé vers une page affichant la liste des différents véhicules.
2. Il sélectionne l’option « désactiver» pour le véhicule de son choix.
3. Il sélectionne « confirmer ». Le véhicule est désactivé dans la base de données.
4. Il est redirigé vers la page de visualisation des différents véhicules.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Le véhicule a été supprimé dans la base de données.

#### EF-OL-007 Créer un dépôt

**Résumé**

Ce cas d’utilisation permet au loueur de créer un dépôt.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur sélectionne « Gestion des dépôts ». Il est redirigé vers une page affichant la liste des différents dépôts.
2. Il sélectionne « Créer un nouveau dépôt ». Il est redirigé vers une page affichant un formulaire.
3. Il encode les informations du dépôt dans les champs du formulaire prévus à cet effet.
4. Il sélectionne « confirmer ». Le nouveau dépôt est créé dans la base de données.
5. Il est redirigé vers la page de visualisation des différents dépôts.

**Enchainements alternatifs**

*A1 : Un dépôt existe déjà pour la même ville*

L’enchainement A1 commence au point 3 du scénario nominal.

3. Un message d’erreur est affiché indiquant à l’utilisateur qu’un dépôt existe déjà pour la ville choisie.

Le scénario nominal reprend au point 3.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Un nouveau dépôt est créé dans la base de données.

#### EF-OL-008 Modifier un dépôt

**Résumé**

Ce cas d’utilisation permet au loueur de modifier un dépôt qui n’es pas liée à un forfait

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur sélectionne « Gestion des dépôts ». Il est redirigé vers une page affichant la liste des différents dépôts.
2. Il sélectionne l’option « Modifier », « Activer » ou « Désactiver » pour le dépôt de son choix. Il est redirigé vers une page affichant un formulaire.
3. Il encode les nouvelles informations du dépôt dans les champs du formulaire prévus à cet effet.
4. Il sélectionne « confirmer ». Le dépôt est modifié dans la base de données.
5. Il est redirigé vers la page de visualisation des différents dépôts.

**Enchainements alternatifs**

*A1 : Un dépôt existe déjà pour la même ville*

L’enchainement A1 commence au point 3 du scénario nominal.

3. Un message d’erreur est affiché indiquant à l’utilisateur qu’un dépôt existe déjà pour la ville choisie.

Le scénario nominal reprend au point 3.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Le dépôt a été modifié dans la base de données.

#### EF-OL-009 Désactiver un dépôt

**Résumé**

Ce cas d’utilisation permet au loueur de supprimer un dépôt qui n’est pas liée à un forfait

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur (Loueur) sélectionne « Gestion des dépôts ». Il est redirigé vers une page affichant la liste des différents dépôts.
2. Il sélectionne l’option « Désactiver » pour le dépôt de son choix.
3. Il sélectionne « Confirmer ». Le dépôt est désactivé dans la base de données.
4. Il est redirigé vers la page de visualisation des différents dépôts.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Le dépôt a été désactivé dans la base de données.

#### EF-OL-010 Créer un forfait

**Résumé**

Ce cas d’utilisation permet au loueur de créer un forfait.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur sélectionne « Gestion des forfaits ». Il est redirigé vers une page affichant la liste des différents forfaits.
2. Il sélectionne « Créer un nouveau forfait ». Il est redirigé vers une page affichant un formulaire.
3. Il encode les informations du forfait dans les champs du formulaire prévus à cet effet.
4. Il sélectionne « confirmer ». Le nouveau forfait est créé dans la base de données.
5. Il est redirigé vers la page de visualisation des différents forfaits.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Un nouveau forfait est créé dans la base de données.

#### EF-OL-011 Modifier un forfait

**Résumé**

Ce cas d’utilisation permet au loueur de modifier un forfait qui n’es pas liée à une réservation.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur sélectionne « Gestion des forfaits ». Il est redirigé vers une page affichant la liste des différents forfaits.
2. Il sélectionne l’option « Modifier » pour le forfait de son choix. Il est redirigé vers une page affichant un formulaire.
3. Il encode les nouvelles informations du forfait dans les champs du formulaire prévus à cet effet.
4. Il sélectionne « confirmer ». Le forfait actuel est désactivé dans la base de données et un nouveau forfait correspondant est créé dans la base de données.
5. Il est redirigé vers la page de visualisation des différents forfaits.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Le forfait a été modifié dans la base de données.

#### EF-OL-012 Désactiver un forfait

**Résumé**

Ce cas d’utilisation permet au loueur de désactiver un forfait.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur (Loueur) sélectionne « Gestion des forfaits ». Il est redirigé vers une page affichant la liste des différents forfaits.
2. Il sélectionne l’option « désactiver » pour le forfait de son choix.
3. Il sélectionne « confirmer ». Le forfait est désactivé dans la base de données.
4. Il est redirigé vers la page de visualisation des différents forfaits.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Le forfait a été désactivé dans la base de données.

#### EF-OL-013 Créer une ville

**Résumé**

Ce cas d’utilisation permet au loueur de créer une ville.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur sélectionne « Gestion des villes ». Il est redirigé vers une page affichant la liste des différentes villes.
2. Il sélectionne « Créer une nouvelle ville ». Il est redirigé vers une page affichant un formulaire.
3. Il encode les informations de la ville dans les champs du formulaire prévus à cet effet.
4. Il sélectionne « confirmer ». La nouvelle ville est créée dans la base de données.
5. Il est redirigé vers la page de visualisation des différentes villes.

**Enchainements alternatifs**

*A1 : Une ville du même nom existe déjà*

L’enchainement A1 commence au point 4 du scénario nominal.

5 Il est redirigé vers le formulaire de création. Un message d’erreur est affiché.

Le scénario nominal reprend au point 3.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Une nouvelle ville est créée dans la base de données.

#### EF-OL-014 Modifier une ville

**Résumé**

Ce cas d’utilisation permet au loueur de modifier une ville qui n’es pas liée à un dépôt

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur sélectionne « Gestion des villes ». Il est redirigé vers une page affichant la liste des différentes villes.
2. Il sélectionne l’option « Modifier », « Activer » ou « Désactiver » pour la ville de son choix. Il est redirigé vers une page affichant un formulaire.
3. Il encode les nouvelles informations de la ville dans les champs du formulaire prévus à cet effet.
4. Il sélectionne « confirmer ». La ville est modifiée dans la base de données.
5. Il est redirigé vers la page de visualisation des différentes villes.

**Enchainements alternatifs**

*A1 : Une ville du même nom existe déjà*

L’enchainement A1 commence au point 3 du scénario nominal.

3. Un message d’erreur est affiché indiquant à l’utilisateur qu’une ville de ce nom existe déjà.

Le scénario nominal reprend au point 3.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* La ville a été modifiée dans la base de données.

#### EF-OL-015 Supprimer une ville

**Résumé**

Ce cas d’utilisation permet au loueur de supprimer une ville qui n’est pas liée à un dépôt

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur (Loueur) sélectionne « Gestion des villes ». Il est redirigé vers une page affichant la liste des différentes villes.
2. Il sélectionne l’option « supprimer » pour la ville de son choix.
3. Il sélectionne « confirmer ». La ville est supprimée dans la base de données.
4. Il est redirigé vers la page de visualisation des différentes villes.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* La ville a été supprimée dans la base de données.

#### EF-OL-016 Créer un pays

**Résumé**

Ce cas d’utilisation permet au loueur de créer un pays.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur sélectionne « Gestion des pays ». Il est redirigé vers une page affichant la liste des différents pays.
2. Il sélectionne « Créer un nouveau pays ». Il est redirigé vers une page affichant un formulaire.
3. Il encode les informations du pays dans les champs du formulaire prévus à cet effet.
4. Il sélectionne « confirmer ». Le nouveau pays est créé dans la base de données.
5. Il est redirigé vers la page de visualisation des différents pays.

**Enchainements alternatifs**

*A1 : Un pays du même nom existe déjà*

L’enchainement A1 commence au point 3 du scénario nominal.

3. Un message d’erreur est affiché indiquant à l’utilisateur qu’un pays avec le même nom existe déjà .

Le scénario nominal reprend au point 3.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Un nouveau pays est créé dans la base de données.

#### EF-OL-017 Modifier un pays

**Résumé**

Ce cas d’utilisation permet au loueur de modifier un pays qui ne possède pas de ville.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur sélectionne « Gestion des pays ». Il est redirigé vers une page affichant la liste des différents pays.
2. Il sélectionne l’option « Modifier », « Activer » ou « Désactiver » pour le pays de son choix. Il est redirigé vers une page affichant un formulaire.
3. Il encode les nouvelles informations du pays dans les champs du formulaire prévus à cet effet.
4. Il sélectionne « confirmer ». Le pays est modifié dans la base de données.
5. Il est redirigé vers la page de visualisation des différentes notoriétés.

**Enchainements alternatifs**

*A1 : Un pays avec la même immatriculation existe déjà*

L’enchainement A1 commence au point 3 du scénario nominal.

3. Un message d’erreur est affiché indiquant à l’utilisateur qu’un pays avec la même immatriculation existe déjà.

Le scénario nominal reprend au point 3.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Le pays a été modifié dans la base de données.

#### EF-OL-018 Supprimer un pays

**Résumé**

Ce cas d’utilisation permet au loueur de supprimer un pays qui n’est pas lié à une ville.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur (Loueur) sélectionne « Gestion des pays ». Il est redirigé vers une page affichant la liste des différents pays.
2. Il sélectionne l’option « supprimer » pour le pays de son choix.
3. Il sélectionne « confirmer ». Le pays est supprimé dans la base de données.
4. Il est redirigé vers la page de visualisation des différents pays.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Le pays a été supprimé dans la base de données.

#### EF-OL-019 Créer un prix

**Résumé**

Ce cas d’utilisation permet au loueur de créer un prix.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur sélectionne « Gestion des prix ». Il est redirigé vers une page affichant la liste des différents prix.
2. Il sélectionne « Créer un nouveau prix ». Il est redirigé vers une page affichant un formulaire.
3. Il encode les informations du prix dans les champs du formulaire prévus à cet effet.
4. Il sélectionne « confirmer ». Le nouveau prix est créé dans la base de données.
5. Il est redirigé vers la page de visualisation des différents prix.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Un nouveau prix est créé dans la base de données.

#### EF-OL-020 Modifier un prix

**Résumé**

Ce cas d’utilisation permet au loueur de modifier un prix .

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.

**Scénario nominal**

1. L’utilisateur sélectionne « Gestion des prix ». Il est redirigé vers une page affichant la liste des différents prix.
2. Il sélectionne « Modifier pour le prix choisi ». Il est redirigé vers une page affichant un formulaire.
3. Il encode les informations du prix à modifier dans les champs du formulaire prévus à cet effet.
4. Il sélectionne « confirmer ». Il est redirigé vers la page de visualisation des différents prix.
5. Le prix modifié est clôturé dans la base de données avec la date actuelle comme date de fin. Un nouveau prix correspondant est créé dans la base de données avec la date actuelle comme date de début.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* Un nouveau prix correspondant a été créé dans la base de données.

#### EF-OL-021 Démarrer une location

**Résumé**

Ce cas d’utilisation permet au loueur de démarrer la location pour une réservation.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.
* Une réservation a été enregistrée pour la date actuelle.

**Scénario nominal**

1. Un client se présente dans un dépôt pour retirer un véhicule à la suite d’une réservation effectuée.
2. L’utilisateur (Loueur) sélectionne « Gestion des réservations ». Il est redirigé vers une page affichant la liste des différentes réservations.
3. L’utilisateur sélectionne l’option « Démarrer la location » pour la réservation souhaitée en fonction du client concerné. Il est redirigé vers la page affichant les détails de la location.
4. Il encode le kilométrage actuel du véhicule et sélectionne « Confirmer ». Il est redirigé vers la page affichant la liste des différentes réservations.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* La réservation est en cours de location.

#### EF-OL-022 Clôturer une location

**Résumé**

Ce cas d’utilisation permet au loueur de clôturer une location liée à une réservation.

**Préconditions**

* L’utilisateur (Loueur) est connecté sur la page Loueur du portail.
* Une réservation est en cours de location.

**Scénario nominal**

1. Un client vient remettre un véhicule en cours de location dans un dépôt.
2. L’utilisateur (Loueur) sélectionne « Gestion des réservations ». Il est redirigé vers une page affichant la liste des différentes réservations en cours de location.
3. L’utilisateur sélectionne l’option « Clôturer la location » pour la réservation souhaitée en fonction du client concerné. Il est redirigé vers la page affichant les détails de la location.
4. Il encode le kilométrage actuel du véhicule, le dépôt de retour, et sélectionne « Confirmer ». Il est redirigé vers la liste des réservations.
5. Il remet la facture au client.

**Enchainements alternatifs**

*A1 : Kilométrage de retour encodé inférieur au kilométrage de départ*

L’enchainement A1 commence au point 4 du scénario nominal.

5. Il est redirigé vers le formulaire de clôture de la réservation. Un message d’erreur est affiché.

Le scénario nominal reprend au point 4.

**Enchainements d’erreur**

Une erreur technique se produit. Un message d’erreur est affiché, informant l’utilisateur qu’une erreur s’est produite.

**Postconditions**

* La réservation est clôturée.

Description des entités

|  |  |
| --- | --- |
| Nom | Description |
| Notoriété | Entité regroupant les informations d’une notoriété |
| Voiture | Entité regroupant les informations d’une voiture |
| Dépôt | Entité regroupant les informations d’un dépôt |
| Forfait | Entité regroupant les informations d’un forfait |
| Ville | Entité regroupant les informations d’une ville |
| Pays | Entité regroupant les informations d’un pays |
| Prix | Entité regroupant les informations d’une catégorie de prix/km |
| Clients | Entité regroupant les informations d’un client |
| Réservation | Entité regroupant les informations d’une réservation |

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 champ 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 un nom obligatoire |
| CF-RS-036 | Un pays peut contenir de 0 à plusieurs villes |

Prix

|  |  |
| --- | --- |
| Code | Description |
| CF-RS-037 | Un prix possède un ID unique |
| *DB > Table Prix > Keys > PK\_ Prix : Clé primaire sur IDPrix* |
| CF-RS-038 | Un prix peut se référer à de 0 à 1 pays |
| *DB > Table Prix > Keys > FK\_ Prix\_Pays : Clé étrangère sur IDPays, pointant sur IDPays de la table pays* |
| 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 obligatoire |
| 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 |
| CF-RS-54 | Une réservation possède un champ ‘pénalité’ |

Règles de validation

Effectuer\_Réservation Client

|  |  |
| --- | --- |
| *Code* | *Description* |
| CF-RV-001 | La date de départ doit être postérieur à la date de réservation et ne peut pas se faire dans le passé |
| *VS > contrôleur MVC > Vérification sur les dates encodées : affiche un message d’erreur en cas de dates incorrectes* |
| CF-RV-002 | La date de retour doit être postérieure à la date de départ |
| *VS > contrôleur MVC > Vérification sur les dates encodées : affiche un message d’erreur en cas de dates incorrectes* |
| CF-RV-003 | Choix d’un dépôt de retour uniquement possible si un forfait existe entre le dépôt de départ et le dépôt de retour, sinon choix du prix/km obligatoire |
| *VS > Backend > Requête à la BD pour fournir uniquement les dépôts de retour possible en fonction du dépôt de départ et des forfaits existants*  *VS > Vue MVC > Affichage d’une dropdownlist avec les dépôts de retour possible ou le choix du pris au KM. Affichage d’un message d’erreur si pas de choix indiqué* |
| CF-RV-004 | Choix d’une voiture obligatoire |
| *VS > Vue MVC > Affichage d’un message d’erreur si pas de choix indiqué* |
| CF-RV-005 | Choix d’une voiture qui n’est pas déjà réservée pour la période souhaitée |
| *VS > Backend > Requête à la BD pour fournir uniquement les voitures disponibles non-reprises par d’autres réservations pour la même période.*  *VS > Vue MVC > Affichage d’une dropdownlist avec les voitures disponibles.* |

Annuler\_Réservation Client

|  |  |
| --- | --- |
| *Code* | *Description* |
| CF-RV-00 | Une réservation ne peut être annulée que si celle-ci n’a pas encore débuté effectivement (date de départ) |
| *VS > Vue MVC > Condition sur l’affichage de l’option d’annulation dans l’historique des réservations, en fonction du kilométrage de retour* |

Démarrer\_Location Loueur

|  |  |
| --- | --- |
| *Code* | *Description* |
| CF-RV-00 | La date de départ de la location doit être postérieure à la date de réservation |
| *VS > contrôleur MVC > Encodage automatique de la date de départ* |

Clôturer\_Location Loueur

|  |  |
| --- | --- |
| *Code* | *Description* |
| CF-RV-00 | Le kilométrage au retour du véhicule doit être supérieur au kilométrage au départ |
| *VS > Contrôleur MVC > Vérification sur le kilométrage de retour : affichage d’un message d’erreur si condition non respectée* |
| CF-RV-00 | La date de retour doit être postérieure à la date de départ |
| *VS > Contrôleur MVC > Fixation automatique de la date de retour lors de l’encodage* |

Gestion\_Ville Loueur

|  |  |
| --- | --- |
| *Code* | *Description* |
| CF-RV-0 | Une Ville ne peut être supprimée que si elle ne contient pas de dépôt |
| *VS > Vue MVC > Condition sur l’affichage de l’option de suppression* |

Gestion\_Pays Loueur

|  |  |
| --- | --- |
| *Code* | *Description* |
| CF-RV-0 | Un Pays ne peut être supprimé que si il ne contient pas de ville |
| *VS > Vue MVC > Condition sur l’affichage de l’option de suppression* |

Gestion\_Forfait Loueur

|  |  |
| --- | --- |
| *Code* | *Description* |
| CF-RV-0 | Nécessité de garder un historique des forfaits |
| *VS > backend > Lors de la suppression ou de la modification d’un forfait. Une date de fin est automatiquement ajoutée au forfait actuel et un nouveau forfait est créé avec un nouvel ID, pour les mêmes IDDépôt.* |

Gestion\_Prix Loueur

|  |  |
| --- | --- |
| *Code* | *Description* |
| CF-RV-0 | Nécessité de garder un historique des prix/km |
| *VS > backend > Lors de la suppression ou de la modification d’un prix. Une date de fin est automatiquement ajoutée au Prix actuel et un nouveau prix est créé avec un nouvel ID, pour le même IDPays.* |

Règles de calcul

Réduction pour réservation anticipée

|  |  |
| --- | --- |
| *Code* | *Description* |
| CF-RC-001 | Réduction pour réservation anticipée |
| *VS > Backend > BLCalculPrix > Détermination de la ristourne en fonction de la différence de jours entre la date de réservation et la date de départ.* |

Calcul du kilométrage parcouru

|  |  |
| --- | --- |
| *Code* | *Description* |
| CF-RC-002 | Kilométrage parcourus |
| *VS > Backend > Modèles > Calcul de la différence entre le kilométrage au retour et le kilométrage au départ* |

* Caclul du prix si option Prix/Km

Le calcul du prix si l’option ‘prix au km’ est choisie est le suivant :

*Prix calculé = (prix/km \* nombre de kilomètres parcourus) \* Coefficient multiplicateur.*

De ce total est encore éventuellement déduite la réduction pour réservation anticipée

*Prix Total = Prix calculé – (Prix calculé \* Réduction)*

* Calcul du prix si choix de forfait

*Prix calculé = Prix du forfait \* Coefficient Multiplicateur*

De ce total est encore éventuellement déduite la réduction pour réservation anticipée. On déduit ensuite la ristourne de 5% ou on ajoute une pénalité de 10% en fonction du dépôt de retour.

*Prix total = Prix calculé - (Prix calculé \* Réduction) – (Prix calculé \* ristourne) / + (Prix calculé \* Pénalité)*

Schéma relationnel de la solution

Schéma Entité-Association

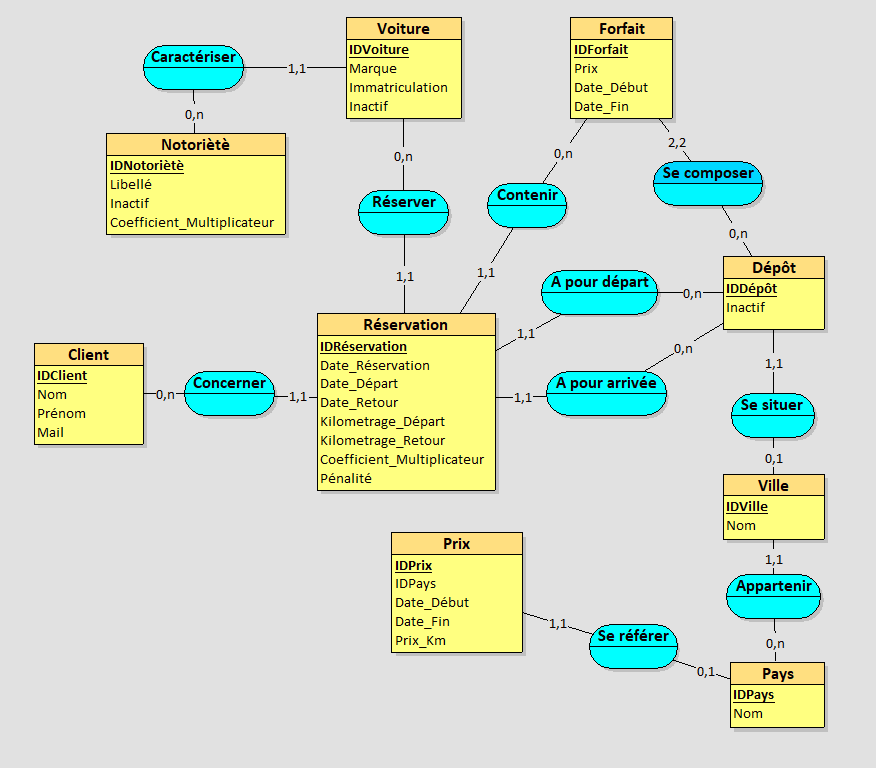
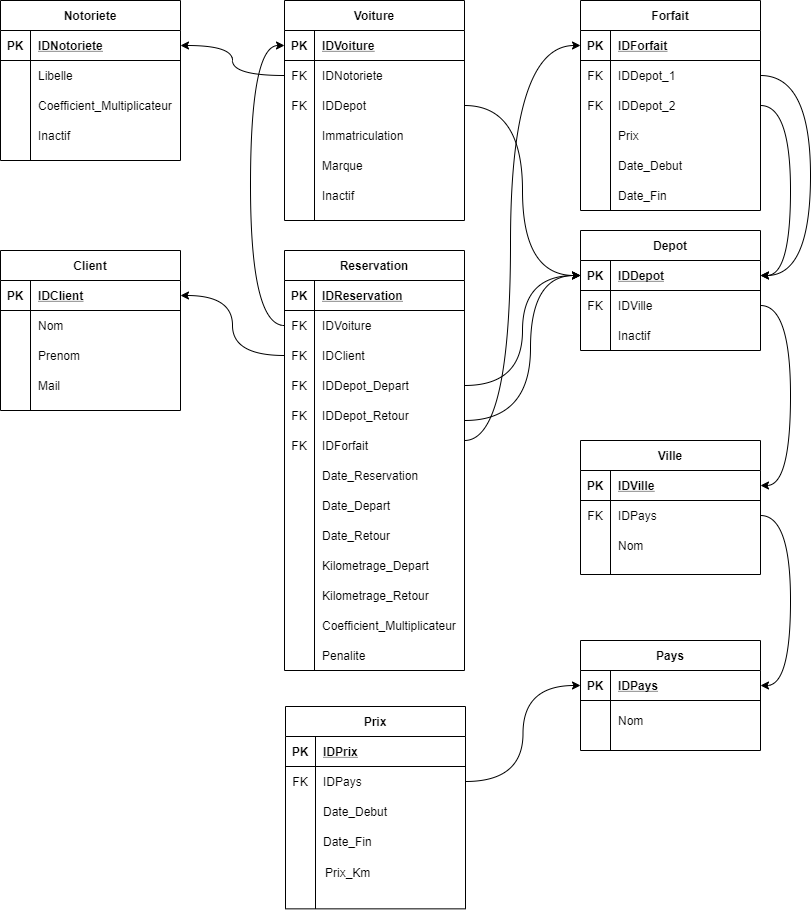


Schéma relationnel



Implémentation des contraintes

L’implémentation des contraintes est directement mentionnée dans la partie ‘Contraintes fonctionnelles’.

Analyse Technique

Technologie proposée

* Nous utiliserons une base de données relationnelles avec SQL Server.
* La partie applicative sera réalisée en .NET.
* Elle contiendra deux parties distinctes
  + La partie back-end, avec une architecture en couches. Le back-end sera chargé de communiquer avec la base de données et de rendre ces données disponibles via une Web API
  + La partie Front-end sera une application Web réalisée avec une architecture MVC. Le front-end sera chargé de l’interface avec les utilisateurs et échangera les données avec le back-end via la web API.
* La partie back-end communiquera avec la base de données via Entity Framework.

~~Architecture applicative~~

~~Algorithmes spécifiques~~

Annexes

Enoncé

*C’est une société qui gère des leasings de voiture…*

*Evidemment, il y plusieurs voitures immatriculées, des type différents et une notoriété adéquate.*

*Cette société opère dans plusieurs pays et villes dans ceux-ci.*

*Chaque voiture est liée à un bureau/dépôt.*

*Un client peut réserver une voiture et définir son voyage. Ce dernier est constitué d’une date de départ et de fin et d’un bureau de départ et du type de calcul de la course.*

*Si le client réserve x semaines avant son départ, il a droit a une réduction de 5 %/semaine jusqu’à un maximum de 20%. La dernière semaine n’a pas de réduction possible.*

*Le prix de la location de la voiture est en fonction de sa notoriété (Luxe, Van,…).*

*Le prix de la course est calculé*

* *Soit en fonction du dépôt de retour de la voiture. Si le retour est défini au moment de la location, une ristourne de 5% est appliquée. Mais si le dépôt de retour n’est pas celui défini, une amende de 10 % luis sera appliquée au forfait le plus onéreux.*
* *Soit en fonction du kilométrage effectué et le prix du kilométrage est lié au pays auquel appartient le dépôt de départ.*

*Il existe donc une liste des forfaits entre les dépôts. On ne peut pas proposer un forfait s’il n’existe pas et donc le prix au KM devra être choisi si nécessaire.*

*Le retour d’une voiture se fait dans le dépôt et ce dernier retrouve le contrat de location y indique soit le dépôt d’arrivée soit le kilométrage adéquat et fournit la facture au client.*

*2 applications sont nécessaires…*

*Une application pour le client qui fait la réservation et l’administration.*

*Une application pour le dépôt…*