Galaxy Swiss Bourdin

Compte rendu : Application GSB-Admin et Intégrité des intranets

PPE6 – Groupe 2

Rédiger par :   
- AUGUSTIN Jean-Loïc  
- COINDEVEL Bryan  
- LEBEAU Emmanuel  
- TREPORT Elie  
- VALENTIN Samuel

Table des matières

Chiffrement 4

Introduction 4

Chiffrement de VB.NET 6

Carré de Vigenère 7

Base De Données 9

Création de la Base De Données 9

Restriction des privilèges 13

Triggers 15

Procédures stockées 17

L’application GSB-Admin 18

Introduction 18

Fichier de configuration .ini 19

Formulaire de connexion 20

Formulaire de modification du fichier .ini 21

Formulaire liste des utilisateurs 22

Formulaire de création d’un utilisateur 23

Formulaire de réinitialisation de mot de passe 24

Note évolutive 25

Début avec contexte

Prévoit qui allait faire quoi (ancien Gantt)

Qu’est-ce qui s’est réellement passer (autre Gantt ?? Et tableau)

Explication de pourquoi avoir besoin de ses serveurs 🡪 Transition sur la partie réseau

Explication du pourquoi de l’application 🡪 Transition

|  |  |
| --- | --- |
| RÉSEAU | |
| AUGUSTIN Jean-Loïc |  |
| COINDEVEL Bryan |  |
| DÉVELOPPEUR | |
| LEBEAU Emmanuel | **Création des maquettes**  **Chiffrement des informations**   * Chiffrement des informations avec VB.NET * Déchiffrement des informations avec VB.NET   **Formulaire de création d’un utilisateur**   * Création d’un utilisateur (Objet) * Affichage des voitures   **Formulaire de la liste des utilisateurs**   * Affichage des utilisateurs * Tri des utilisateurs en fonction de leurs types   **Connexion à l’application**   * Lecture du fichier .ini * Chiffrement des informations du fichier * Style du formulaire de connexion |
| TREPORT Elie | **Création des classes**  **Chiffrement des informations**   * Création de fonctions de chiffrement et déchiffrement inspiré du chiffrement de Vigenère   **Formulaire de connexion**   * Récupération des informations du fichier .ini * Création et remplissage des collections avec DataReader   **Formulaire de la liste des utilisateurs**   * Auto-complétions de la zone de texte de recherche et affichage dans la liste   **Formulaire de création des utilisateurs**   * Insertion d’un utilisateur (BDD) * Emprunter un véhicule et vérification de sa disponibilité via procédure stockée   **Bases De Données**   * Création des procédures stockées / Triggers * Création de l’utilisateur AdminGSB et restriction des droits |
| VALENTIN SAMUEL | **Formulaire de réinitialisation de mot de passe**   * Affichage des utilisateurs dans une liste * Sélection d’un utilisateur * Modification du mot de passe via une procédures stockées   **Diagramme De Gant**   * Réalisation du diagramme de Gant   **Base De Données**   * Création de la Base De Données |

## Chiffrement

### Introduction

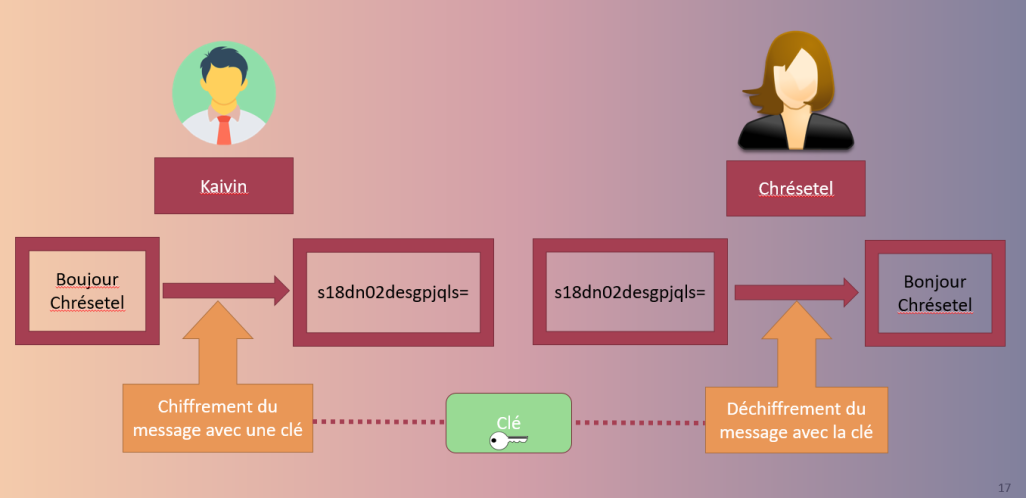
|  |
| --- |
| Le chiffrement est un procédé de cryptographie grâce auquel on souhaite rendre la compréhension d’un document impossible à toute personne qui n’a pas la clé de (dé)chiffrement.  Résultat de recherche d'images pour "chiffrement" Suite aux demandes de Galaxy Swiss Bourdin (GSB) nous avons effectué une veille sur le chiffrement en effectuant des recherches. |

Il existe de nombreux types de chiffrement utilisant différents types de clés, il existe plusieurs types d’utilisation de clé, soit :

|  |  |
| --- | --- |
| Clé symétrique | Une clé symétrique est une clé unique qui permet de chiffrer le message et de le déchiffrer avec la même clé.  Ainsi sans cette clé le message à peu de chance d’être déchiffrer, il faut donc pour pouvoir le déchiffrer avoir la clé utilisée pour le chiffrage du message. |
| Clé asymétrique | Une clé asymétrique est un système qui fonctionne avec deux clés différentes :  Une clé publique Une clé privée    La clé privée comme son nom l’indique n’est pas partager, elle permet de déchiffrer le message. Elle est donc plus utile dans le cadre d’une communication. |

Les algorithmes de chiffrement à clé symétrique ont été choisies car ils sont plus rapides par rapport aux algorithmes à clé publique et peuvent être limité (RSA est limiter mathématiquement à cause des nombres de caractères) et se prêtent bien aux transformations de chiffrement sur les flux de données importants.

Ainsi le chiffrement à clé symétrique est utilisé dans l’application GSB-Admin, de deux manières différentes. L’utilisation de la clé symétrique utilisée peut être assimilée au schéma ci-dessous :



Grâce à cela les communications qui transitent sur le réseau qui sont susceptibles d'être lues voire modifiées par des tiers non autorisés sont protégés grâce à un algorithme de chiffrement cela préviendrais le fait que si un tiers intercepte les données chiffrées, il lui sera donc difficile de les déchiffrer.

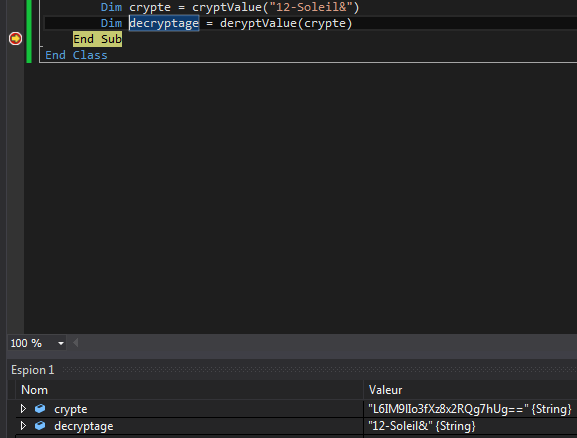
Au sein de l’application GSB-Admin nous avons utilisé deux types de chiffrement totalement différent qui utilisent une clé symétrique différente. Ainsi on as une double sécurité avec un double cryptage de nos informations sensibles et donc deux fois moins de chances de retrouver les données déchiffrer.

### Chiffrement de VB.NET

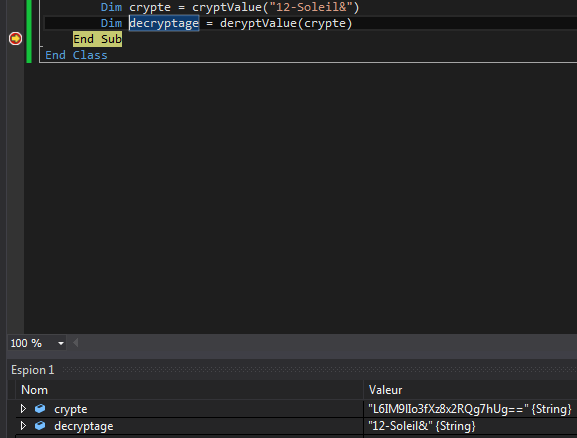
|  |
| --- |
| Résultat de recherche d'images pour "securité password" Le premier chiffrement intégrer à VB.Net utilise la classe [System.Security.Cryptography](https://docs.microsoft.com/fr-fr/dotnet/api/system.security.cryptography?view=netframework-4.8). Cette classe fournit plusieurs services de chiffrement qui comprend le chiffrement et le déchiffrement des chaînes. |

Pour pouvoir chiffre notre chaîne de caractère on utilise l’interface ICryptoTransform Interface qui utilise des algorithmes de hachage.

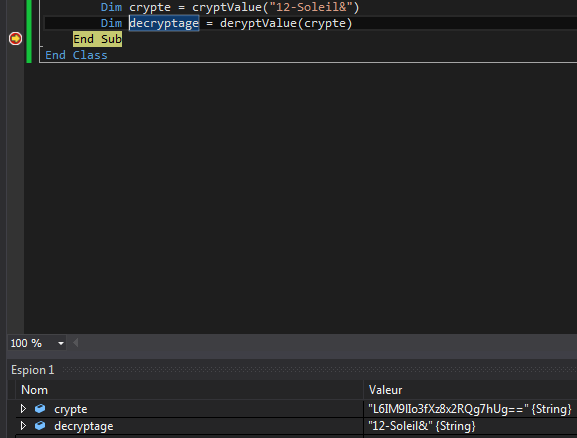
Pour exemple lorsque l’on va chiffrer notre mot de passe « 12-Soleil » avec une clé (« aGioP782 ») en appelant notre méthode :



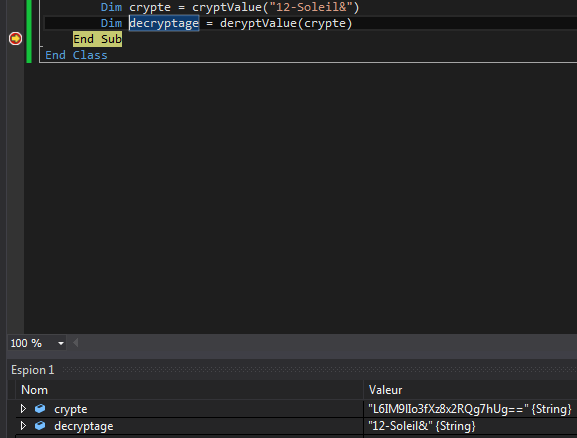
Cela nous donne ce résultat :



Ensuite si l’on veut déchiffrer ce même message, on appelle notre méthode de déchiffrement :



Cela nous retourne ce résultat :



De cette manière on obtient une chaîne de caractère qui n’est pas déchiffrable par l’utilisateur, et les données sensibles peuvent être stockées de manière sécurisée dans un fichier et dans la Base De Données.

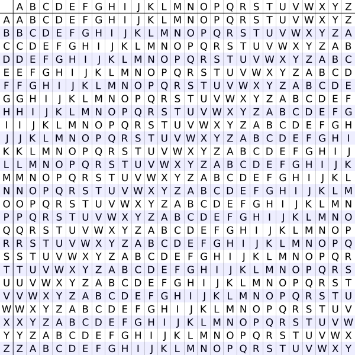
L’intérêt d’utiliser ce chiffrement dans l’application GSB-Admin est d’avoir un chiffrement sur car sans la clé il ne sera pas possible pour une personne tierce de déchiffrer ce message. Ainsi les données sensibles sont sécurisées.

### Carré de Vigenère

|  |
| --- |
| https://upload.wikimedia.org/wikipedia/commons/thumb/1/1a/Vigenere.jpg/220px-Vigenere.jpgLe chiffrement de Vigenère a été créée par un Français nommé Blaise de Vigenère (5 Avril 1523 – 19 Février 1596) qui inventa le chiffrement de Vigenère étant le premier chiffrement difficile à casser. Celui-ci ne fut casser qu’en 1863 par Friedrich Kasiski soit 3 siècles plus tard. |

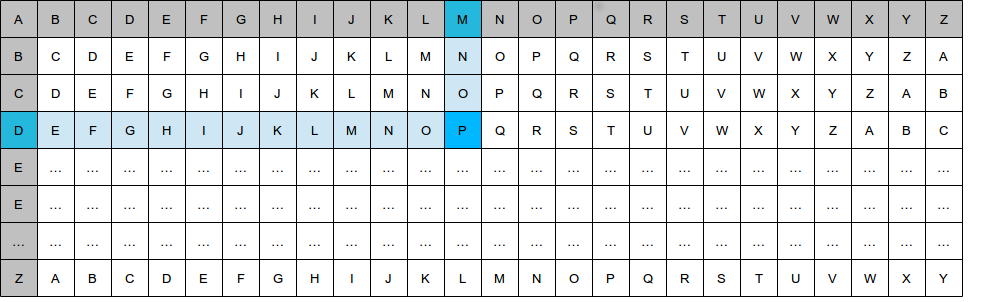
Le chiffrement de Vigenère est une méthode permettant de chiffrer une chaîne de caractère de longueur x grâce à une clé de longueur y.

Plus précisément, ce chiffrement est un système de chiffrement poly alphabétique, qui consiste à permuter les lettres entres elles grâce à la clé de cryptage : C’est le chiffrement par substitution. Ce chiffrement est effectué grâce à un carré composé des lettres de l’alphabet de A à Z comme ci-dessous.



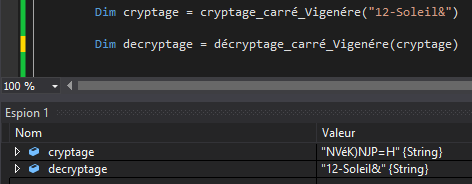
Ce chiffrement permet en croisant la lettre du mot à chiffrer et la ligne du premier caractère de la clé de nous donner un autre caractère ne correspondant pas aux caractères de base. Et on recommence avec le caractère x+1 en reprenant le premier caractère de la clé si celui-ci n’est pas assez grand pour chiffrer la chaîne de caractères.

Comme le montre l’exemple ci-dessous le caractère M chiffrer avec la clé D nous donne P :

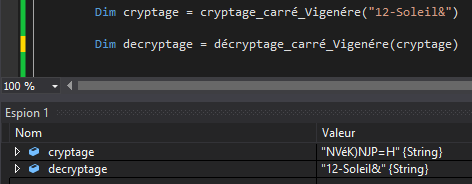


Ce chiffrement a été utilisé au sein de l’application GSB-Admin mais as été modifié, en insérant dans ce carré les lettres minuscules ([a-z]), majuscules ([A-Z]), les chiffres ([0-9]), ainsi que les caractères spécial (², /). Ainsi grâce à cette amélioration et un ordonnancement aléatoire entres chaque caractères la technique du chiffrement de Vigenère as été amélioré et personnifier permettant donc un plus grand nombre de possibilités. En effet avec un total de 108 caractères on a donc un total de 108x possibilités ou x varie en fonction du nombre de caractères de la clé de cryptage.

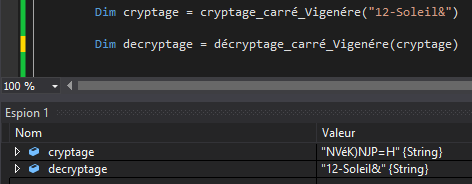
Au sein de l’application GSB-Admin ce chiffrement as été créé de toutes pièces, grâce à l’appel des fonctions qui permettent de récupérer tous les caractères du clavier contenues dans un fichier (all\_touche.txt).  
L’algorithme utilisé est identique au chiffrement utilisé via le carré de Vigenère. Pour exemple lorsque l’on veut crypter la chaîne de caractères « 12-Soleil& » avec la clé « GSB-Admin » on appel notre fonction :



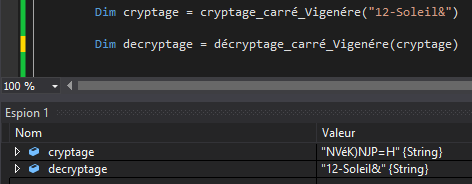
Celle-ci nous donne ce résultat suivant contenue dans la variable nommé « cryptage » :



Par la suite si l’on veut déchiffrer cette chaîne de valeur, on doit utiliser la même clé utilisée pour le chiffrement (« GSB-Admin »). Ainsi on appelle la fonction permettant ce déchiffrement :



Celle-ci nous retourne la valeur de la chaîne déchiffrée qui est similaire à celle chiffré au début de ce processus :

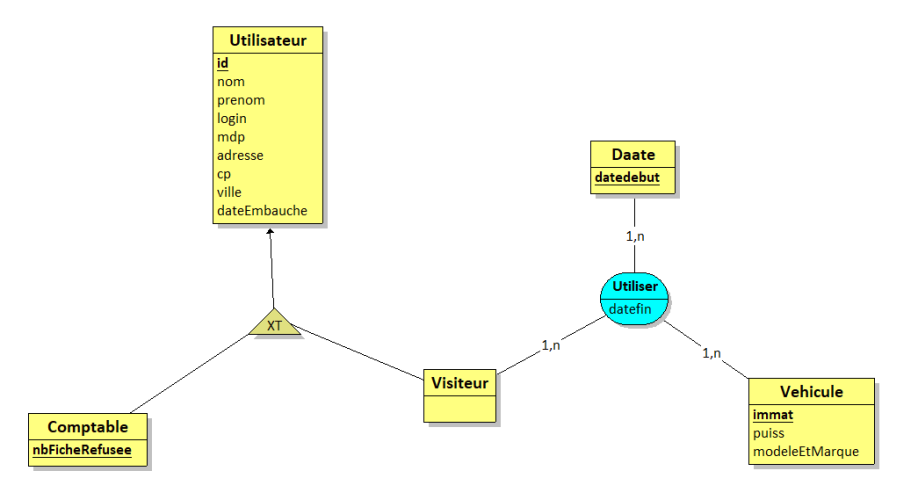


L’intérêt d’utiliser le chiffrement de Vigenère (modifier) dans l’application GSB-Admin est d’avoir un chiffrement personnalisé. Ainsi grâce à la personnalisation du chiffrement de nos chaînes de caractères il est plus compliqué pour une personne tierce de déchiffrer les valeurs chiffrer.

## Base De Données

### Création de la Base De Données

La base de données proposés dans le cahier des charges n’a été aucunement modifié et garde le même aspect. On peut y apercevoir les différents champs contenus dans les tables ainsi que la relation d’héritage avec contrainte qui signifie qu’un utilisateur peut soit être un comptable soit un visiteur et rien d’autre (XT).



Le script de création de la base de données est le suivant :

1. Pour commencer nous avons créé la table « utilisateur » composée des champs spécifiant les données personnelles d’un utilisateur (comptable ou visiteur) :

|  |  |
| --- | --- |
| CREATE TABLE [dbo].[utilisateur](      [id] [int] NOT NULL,      [nom] [varchar](50) NOT NULL,      [prenom] [varchar](50) NOT NULL,      [login] [varchar](50) NOT NULL,      [mdp] [varchar](50) NOT NULL,      [adresse] [varchar](50) NOT NULL,      [cp] [varchar](50) NOT NULL,      [ville] [varchar](50) NOT NULL,      [dateEmbauche] [date] NOT NULL,   CONSTRAINT [PK\_utilisateur\_ID] PRIMARY KEY CLUSTERED  (      [id] ASC  )  ) ON [PRIMARY]  GO |  |

1. Par la suite, on a créé la table « visiteur » qui hérite dans le MCD de la table utilisateur, elle contient donc un champ nommé « id » qui sera en lien avec le champ « id » de la table utilisateur :

|  |  |
| --- | --- |
| CREATE TABLE [dbo].[visiteur](      [id] [int] NOT NULL,   CONSTRAINT [PK\_visiteur\_id] PRIMARY KEY CLUSTERED  (      [id] ASC  )  ) ON [PRIMARY]  GO |  |

1. Ensuite on crée la table comptable, comme la table visiteur elle contient elle aussi un champ nommé « id » permettant de faire le lien avec la table « utilisateur ». Par ailleurs un comptable a une information en plus qui permet de savoir le nombre de fiches qu’il aura refusé, cette information est contenue dans le champ « nbFicheRefusee » :

|  |  |
| --- | --- |
| CREATE TABLE [dbo].[comptable](      [id] [int] NOT NULL,      [nbFicheRefusee] [int] NULL,  PRIMARY KEY CLUSTERED  (      [id] ASC  )  ) ON [PRIMARY] |  |

1. On a ensuite créé la table « vehicule » contenant les informations en lien avec un véhicule :

|  |  |
| --- | --- |
| CREATE TABLE [dbo].[vehicule](      [immat] [varchar](50) NOT NULL,      [puiss] [int] NULL,      [modeleMarque] [varchar](50) NOT NULL,   CONSTRAINT [PK\_vehicule\_immat] PRIMARY KEY CLUSTERED  (      [immat] ASC  )  ) ON [PRIMARY]  GO |  |

1. Enfin nous avons créé la table « utiliser » qui permet de voir les emprunts de véhicules effectués par un visiteur à une date donnée, et quand celui-ci a été rendue ou sera rendue :

|  |  |
| --- | --- |
| CREATE TABLE [dbo].[utiliser](      [immat] [varchar](50) NOT NULL,      [DateDebut] [date] NOT NULL,      [id] [int] NOT NULL,      [dateFin] [date] NOT NULL,   CONSTRAINT [PK\_utiliser\_composer] PRIMARY KEY CLUSTERED  (      [id] ASC,      [DateDebut] ASC,      [immat] ASC  )  ) ON [PRIMARY]  GO |  |

La table utiliser créer ici pour faire le lien entre la table « vehicule » et « visiteur » a accueilli le champ « datedebut » afin de ne pas avoir une contrainte à gérer au niveau des requêtes SQL. Ainsi le champ est directement présent dans la table « utiliser ».

Une fois les tables créer il a fallu mettre les contraintes existantes entres ses tables :

1. Pour commencer il a fallu mettre les contraintes entre la table « comptable » et « utilisateur » :

ALTER TABLE [dbo].[comptable]  WITH CHECK

ADD  CONSTRAINT [FK\_comptable\_id] FOREIGN KEY([id])

REFERENCES [dbo].[utilisateur] ([id])

ON UPDATE CASCADE

ON DELETE CASCADE

1. Ensuite nous avons fait de même avec la table « visiteur » et « utilisateur » :

ALTER TABLE [dbo].[visiteur]  WITH CHECK

ADD  CONSTRAINT [FK\_visiteur\_id] FOREIGN KEY([id])

REFERENCES [dbo].[utilisateur] ([id])

ON UPDATE CASCADE

ON DELETE CASCADE

Grâce à cela nous avons créé la relation d’héritage représenté sur le MCD en liant les clés primaires des tables « visiteur » et « comptable » en les référant à la table utilisateur. Ainsi un identifiant de visiteur ou de comptable doit obligatoirement être présent dans la table utilisateur.

1. Par la suite nous avons créé la relation existante entre la table « utiliser » et la table « vehicule » sur le champ « immat » présent dans la table utiliser qui sera en référence au champ « immat » de la table « vehicule » :

ALTER TABLE [dbo].[visiteur]  WITH CHECK

ADD  CONSTRAINT [FK\_visiteur\_id] FOREIGN KEY([id])

REFERENCES [dbo].[utilisateur] ([id])

ON UPDATE CASCADE

ON DELETE CASCADE

1. Ensuite il a fallu mettre le lien de la table « utiliser » vers la table « visiteur » sur le champ « id » en lien avec le champ « id » de la table « visiteur :

ALTER TABLE [dbo].[visiteur]  WITH CHECK

ADD  CONSTRAINT [FK\_visiteur\_id] FOREIGN KEY([id])

REFERENCES [dbo].[utilisateur] ([id])

ON UPDATE CASCADE

ON DELETE CASCADE

Grâce à cela, on pourra retrouver les emprunts d’un véhicule pour un visiteur, et sur cet emprunt la date de début et la date de fin de l’emprunt.

Pour l’insertion des données de test suite à la création de la Base De Données il a fallu respecter un ordre précis :

1. On a tout d’abord inséré les données dans la table « utilisateur » afin que lorsque l’on insère un comptable / visiteur celui-ci existe déjà dans cette table et donc que la contrainte soit respectée ;
2. Par la suite nous avons insérer les données dans la table visiteur et comptable ;
3. Ensuite nous avons insérer des véhicules dans la table « vehicule » afin de ne pas avoir d’erreur lors de l’insertion dans la table « utiliser » ;
4. Enfin on a pu insérer des données dans la table « utiliser » car il existe déjà des données dans la table « visiteur » et « vehicule », il n’y aura donc pas d’erreur de contraintes si les immatriculations et les identifiants de visiteur existent bel et bien dans les tables référentes.

### Restriction des privilèges

|  |
| --- |
| Afficher l’image source Après avoir créé la Base De Données grâce aux identifiants de connexion fournit par les spécialistes réseaux, la question des restrictions des privilèges s’est posée pour les dirigeants destinés à exploiter cette BDD. |

Les dirigeants n’exploitant la BDD qu’à des fins de lectures, de création, de modification et de suppression (CRUD) nous avons créé un utilisateur ayant des droits restreints afin de ne pas donner aux dirigeants le droit de modifier la structure de celle-ci.

|  |
| --- |
| La restriction des droits s’effectue de haut en bas.  Plus on descend plus les droits d’actions sur la Base De Données sont restreints  Pour exemple le *SuperAdmin* peut modifier la structure des tables ou modifier une procédure stockée alors que l’Admin n’en as pas le droit |

Ainsi pour commencer nous avons créé un login de connexion à la Base SQL Server nommé AdminGSB :

CREATE LOGIN AdminGSB

WITH PASSWORD = ‘……’;

Pour pouvoir restreindre les droits lors de la connexion à la Base SQL Server sur le login AdminGSB nous avons ensuite créé un utilisateur pour ce login de connexion :

CREATE USER AdminGSB FOR LOGIN AdminGSB;

L’utilisateur créer, celui-ci n’a pour l’instant aucun droit et ne put que se connecter à la BDD sans effectuer aucune action. Il a donc fallu lui octroyer des privilèges sur la base De Données GSB\_Admin grâce aux commandes SQL Grant :

GRANT SELECT, UPDATE, DELETE, INSERT

TO AdminGSB;

L’application destiné quant à elle à utiliser des procédures stockées présentes sur la BDD GSB\_Admin il as fallu donner les droits d’exécution sur ces procédures stockées :

GRANT EXECUTE ON OBJECT::Insert\_Update\_User

TO AdminGSB;

GRANT EXECUTE ON OBJECT::verifAdd\_VoitureForVisiteur

TO AdminGSB;

Suite à cela les dirigeants de GSB peuvent utiliser les identifiants de connexion fournis afin de se connecter à la BDD et y effectuer des actions sans y modifier la structure. L’application ayant aussi accès à la BDD grâce aux identifiants de connexion, elle peut effectuer les actions de CRUD et l’appel de procédures stockées.

Ainsi de cette manière les privilèges sont restreints et permettent un contrôle des accès à la BDD comme nous le montre ce schéma ci-dessous.

|  |
| --- |
|  |

### Triggers

|  |
| --- |
| Afficher l’image source Un trigger (déclencheurs en français) est un objet qui est attaché à une table spécifique de la Base De Données. Une fois un Trigger créé il s’exécute lors d’un événement sur la table à laquelle il est lié. |

L’ajout d’un trigger permet de contrôler les différentes actions effectuées par l’utilisateur et peut aussi permettre de garder une cohérence au niveau des données contenues dans les tables grâce au contrôle effectué par les triggers sur les actions CRUD des utilisateurs.

Nous avons donc utilisé les triggers afin de pouvoir contrôler les actions des dirigeants, voici donc les triggers créer sur certaines tables de la BDD :

1. Trigger après suppression sur la table « comptable »

-- Permet de supprimer l'utilisateur concerner par le comptable

CREATE TRIGGER delete\_User\_Comptable ON comptable

AFTER DELETE

AS

BEGIN

DECLARE @id as int;

SELECT @id = id  FROM deleted --On récupère ici le numéro du comptable sur le point d'être supprimé

DELETE FROM utilisateur WHERE id = @id;

END

Ainsi grâce à ce trigger, lorsque l’on va supprimer un comptable au niveau de la table comptable, l’utilisateur le correspondant sera aussi supprimé.

En effet avec la contrainte de clé étrangère qu’à la table comptable vers la table utilisateur, si l’utilisateur x sera supprimé alors le comptable le correspondant sera aussi supprimé mais malheureusement cela n’est pas le cas dans le sens contraire d’où l’intérêt d’avoir créé ce trigger.

1. Trigger après suppression sur la table « visiteur »

-- Permet de supprimer l'utilisateur concerner par le visiteur

CREATE TRIGGER delete\_User\_Visiteur ON visiteur

AFTER DELETE

AS

DECLARE @id as int;

SELECT @id = id  FROM deleted --On récupère ici le numéro du comptable sur le point d'être supprimé

DELETE FROM utilisateur WHERE id = @id;

END

Ici, tout comme l’explication précédente ce trigger fait de même, mais sur la table « visiteur » ayant un lien avec la table « utilisateur ».

1. Trigger avant insertion dans la table « utiliser » nommer « verifAdd\_Voitures »

Grâce à ce trigger, lors de l’insertion d’un emprunt dans la table utilisé le trigger se déclenchera avant l’insertion afin de vérifier que cette insertion peut bien avoir lieu.  
En effet on a constaté que lors d’une insertion certaines informations étaient incohérentes car une voiture empruntée d’une date x à une date y peut aussi être emprunté par une autre personne aux mêmes dates ce qui n’est pas cohérent (une voiture ne peut pas être à deux endroits différents à la fois).

Ainsi nous avons créé un trigger capable de vérifier ses informations. Son but est simple, il permet de vérifier si la voiture est disponible

### Procédures stockées

|  |
| --- |
| Afficher l’image source Une procédure stockée peut être assimilé à une méthode car celle-ci a un nom et est une série d’instruction en SQL. Enregistré dans la Base De Données elle peut être appelé avec ou sans valeurs en paramètres afin d’effectuer cette série d’instruction en SQL et fournir un résultat ou effectuer des actions sur la Base De Données. |

Grâce à l’utilisation d’une procédure on peut effectuer un appel simple en envoyant des valeurs si besoins, et celle-ci se chargera de nous effectuer les opérations et les calculs voulues sur la Base De Données.

Pour l’application GSB-Admin nous avons créé 2 procédures stockées permettant d’effectuer différentes opérations sur les tables qui sont les suivantes :

1. La première procédure stockée nommé permettra à l’utilisateur lors de son appel de pouvoir insérer un utilisateur qui sera comptable ou visiteur en fonction des paramètres envoyés, par ailleurs si cet utilisateur existe déjà il sera mis à jour.

CREATE PROCEDURE Insert\_Update\_User

@param\_id INT,

@param\_nom nvarchar(50),

@param\_prenom nvarchar(50),

…

 END

1. La procédure suivante permet à l’utilisateur d’avoir un retour pour savoir pour une voiture immatriculée y, la disponibilité de celle-ci à une date y :

CREATE PROC [dbo].[verifAdd\_VoitureForVisiteur]

-- Paramètres permettant la vérification

@immat varchar(20),

@dateDebut date,

@dateFin date

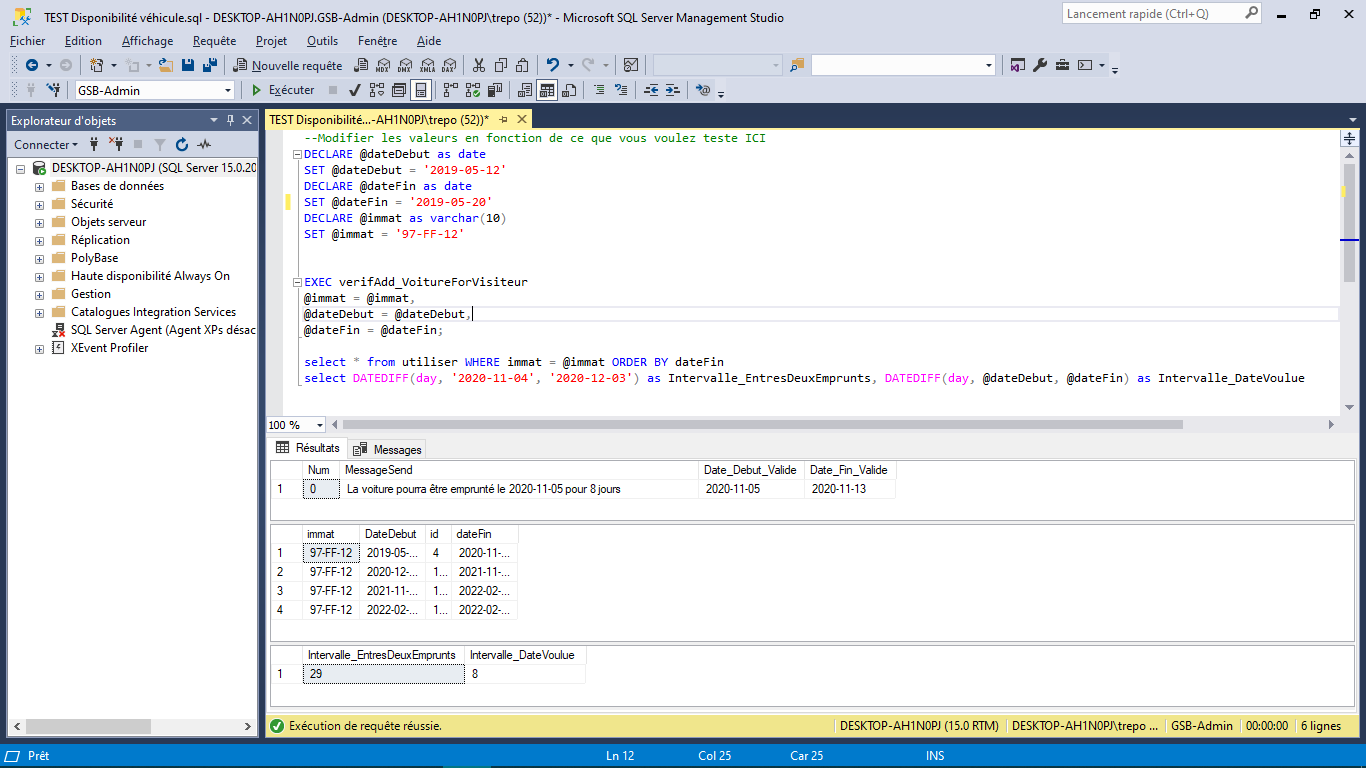
…

END

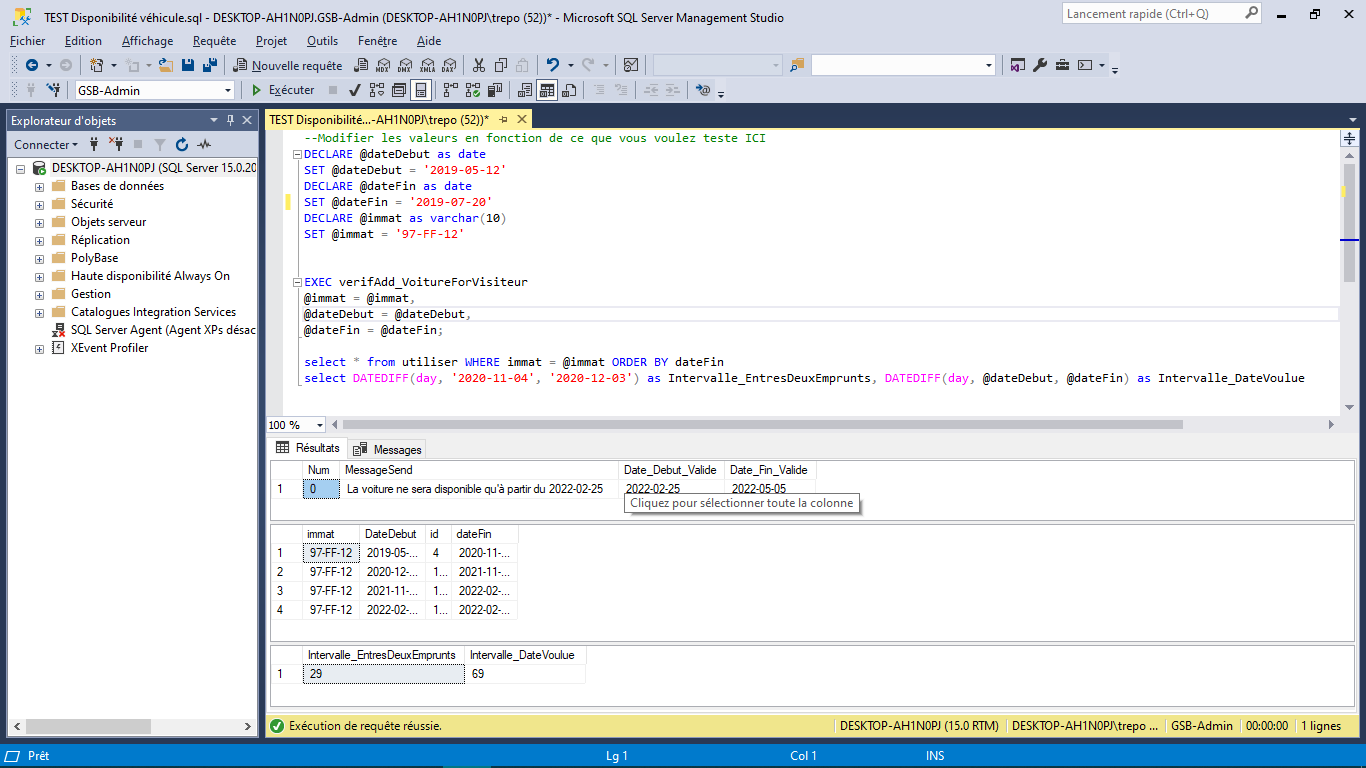
Grâce à cette procédure stocké agissant sur la table « utiliser » on aura un retour après calcul de la disponibilité du véhicule.  
Cette disponibilité utilise exactement le même procédé que le Trigger nommé « verif\_Add\_Voiture » mais cette procédure nous renvoi un résultat afin de savoir quand est-ce que la voiture serait disponible pour une période de x jours, voici donc le schéma de cette procédure :

|  |  |
| --- | --- |
| Afficher l’image source  SQL Server  Afficher l’image source  Requête  Disponibilité du véhicule  Calcul  Table Utiliser |  |

Ainsi, si la voiture est disponible est peut être emprunté la procédure stockée nous retourne un résultat qui sera exploité par l’application GSB-Admin, ce résultat peut varier en fonction de la disponibilité de la voiture en fonction des dates spécifier par l’utilisateur souhaitant emprunter cette voiture.  
Si la voiture est disponible voici ce que nous retourne la procédure :



Contrairement au résultat précédent, la procédure nous renvoi ce résultat si la voiture n’est pas disponible :



Cette procédure stockée est donc très utile pour vérifier la disponibilité d’une voiture pour x jours.

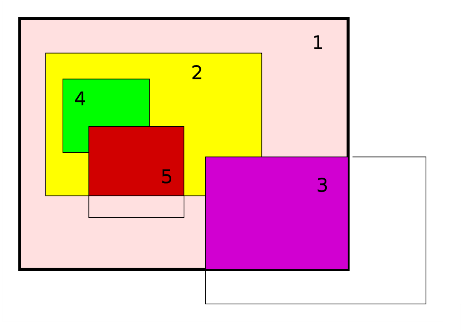
# L’application GSB-Admin

### Introduction

|  |
| --- |
| Résultat de recherche d'images pour "logo VB.Net"" La société Galaxy Swiss Bourdin (GSB) à effectuer la demande d’avoir une application nommée GSB-Admin développer grâce à VB.Net et communiquant avec une Base De Données SQL Server distante. Cette application une fois créé sera destinée à être utilisé par les dirigeants de GSB. |

La société GSB à demander différentes fonctionnalités, en effet voici les différentes demandes de GSB contenues dans le cahier des charges :

1. La première demande était de réaliser cette application dans un formulaire de type MDI pour « Multiple Document Interface », ce type de formulaire permet d’avoir une fenêtre principale qui accueille les fenêtres secondaires à l’intérieur de la fenêtre principale.   
   Ainsi, toutes les fenêtres et les actions du dirigeants de GSB seront contenues dans une fenêtre comme nous le montre ce schéma ci-dessous :



1. Avoir les identifiants de connexion à l’application ET à la base de données dans un fichier crypté
2. Avoir une Base De Données crypté sur les données sensibles

Expliqué la manière de persistance des données avec la BDD (schéma)

### Fichier de configuration .ini

|  |
| --- |
| Le fichier de configuration permettant de ce connecté à l’application a été choisi avec l’extension .ini qui est un format de données introduit par les systèmes d’exploitation de Windows en 1985. Ce type de fichier représentait beaucoup d’avantages au niveau de la gestion du code.  Image associée |

Un fichier portant l’extension .ini est très utiles, en effet ce fichier est composé ainsi :

|  |
| --- |
| *; Last modified 15 July 2010 by Juan Dona*  [owner]  name=juan dona  organization=cablage ideal  [database]  server=192.0.2.42 ; use IP address in case network name resolution is not working  port=143  file = "acme payroll.dat" |

Les commentaires sont précédés d’un « ; ». Les crochets (« [«, «] ») permettent de définir un paragraphe/section.  
Ainsi la section « owner » comporte une information nommée « name » et une autre nommé « organization ».  
Ses informations contenues dans une section peuvent être assimilés à des variables (clé=valeur). Ce qui nous donne pour la clé « owner » la valeur « juan dona ».

Ce fichier portant l’extension .ini as donc été intégrer à l’application GSB-Admin car il permettait de bien organiser les différentes informations qui devait y être stockée. Voici un exemple de notre fichier de configuration utilisé dans l’application (les informations sont chiffrées avec les deux systèmes de chiffrement) :

[Auth]

login=uoMUr6iQvqH8r53GVnh8lw==

motdepasse=hGs9UDdvxB2RAWpg5L9IEA==

[Database]

serveur=rHwVLpTnrw6XGJaCue87Ao6/ldOOG076

baseDeDonnees=PwHrqFDD6/PtlyD8Cpvukg==

user=3RA/vMorp1s4mf8QYKsEMQ==

mdpUser=hGs9UDdvxB2RAWpg5L9IEA==

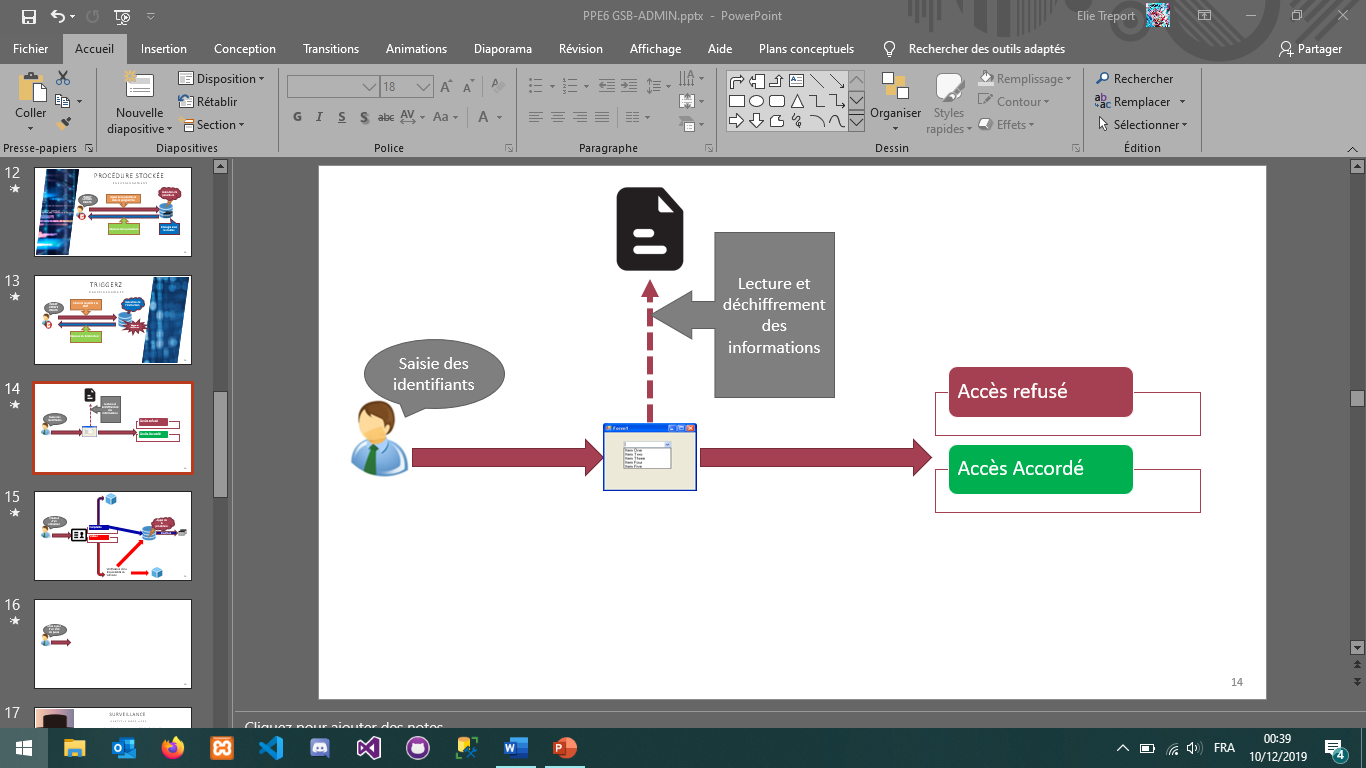
Ce fichier permet ainsi à l’application de récupérer pour la section « Auth » les identifiants de connexion, par ailleurs pour la section « Database » on récupère les informations de connexion à la Base De Données distantes.

### Formulaire de connexion

|  |
| --- |
| Résultat de recherche d'images pour "formulaire connexion" Le formulaire de connexion est le premier formulaire qui apparaitra au lancement de l’application GSB-Admin. Ce formulaire permet tout simplement lors de la connexion de vérifier les identifiants de l’utilisateur en récupérant les valeurs contenues dans le fichier de configuration ayant l’extension «. ini ». |

Lors de la connexion, le formulaire va donc lire le fichier .ini afin de récupérer les identifiants et mot de passe correspondant et les comparer aux identifiants proposés par l’utilisateur. S’ils sont similaires l’un l’autre alors l’application lui donne accès aux différentes fonctionnalités.

Ce schéma explique le procédé lors de la connexion d’un utilisateur :



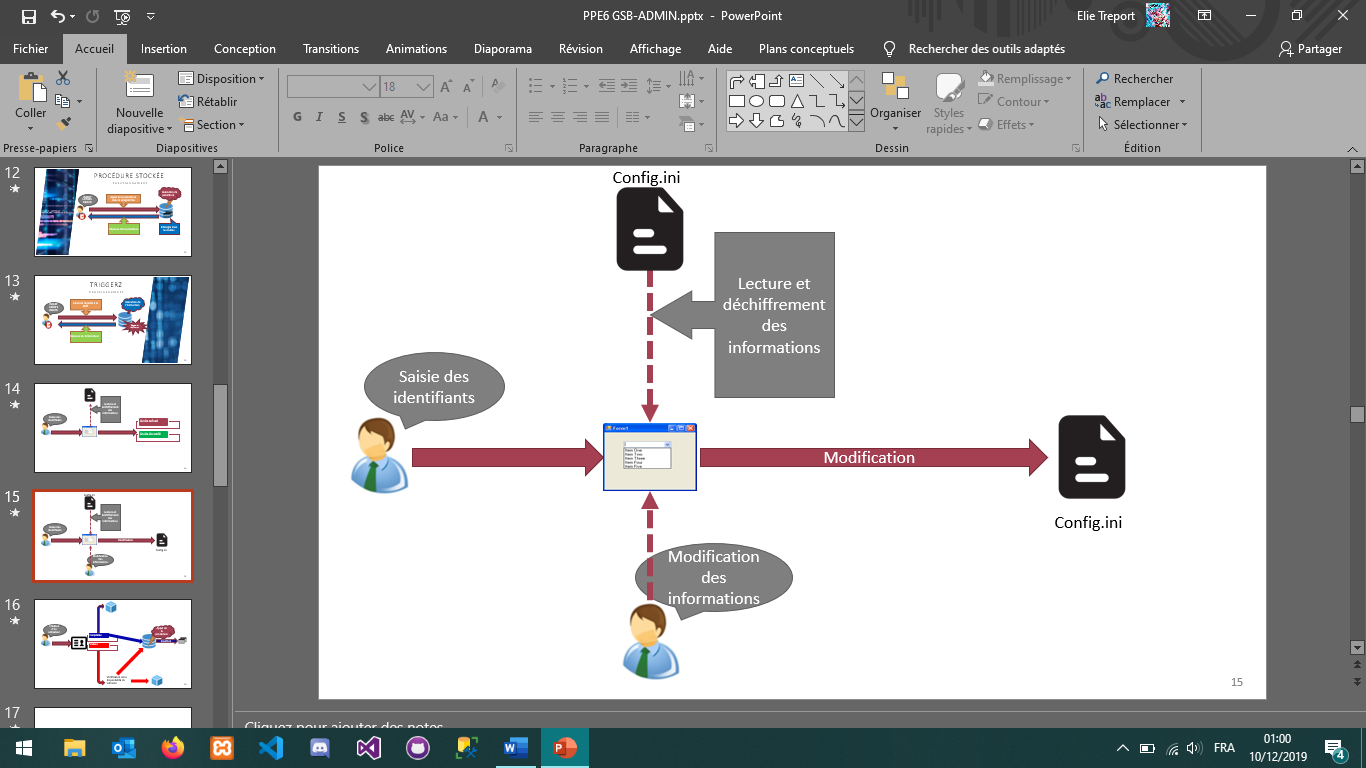
CAS DUSER

### Formulaire de modification du fichier .ini

|  |
| --- |
| Comme le fichier de configuration permettant de se connecter à l’application et à la Base De Données peut être susceptibles d’être modifiés (exemple : changement d’adresse IP, Mise à jour du mot de passe, …) nous avons créé un formulaire permettant de modifier ce fichier.Résultat de recherche d'images pour ".ini" |

Comme le fichier est crypté, cela peut être compliqué pour l’utilisateur de le modifier. Nous avons donc créé un formulaire permettant de modifier les informations dans un formulaire et de recrypter ses informations dans le fichier .ini.

Ci-dessous, un schéma qui permet d’expliquer la modification du fichier .ini :



Cryptage des informations

### Formulaire liste des utilisateurs

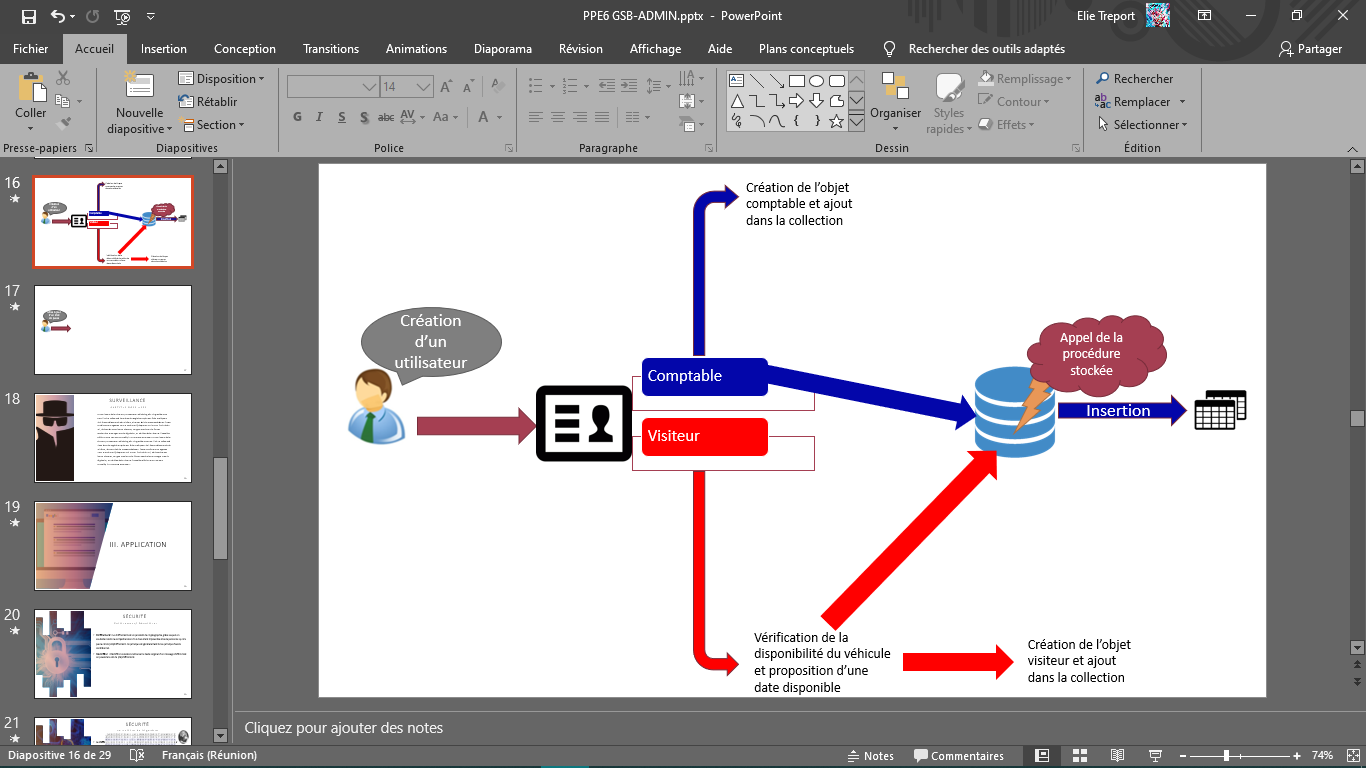
Cas d’utilisation (simple)

Schéma vite fait

### Formulaire de création d’un utilisateur

|  |
| --- |
| Résultat de recherche d'images pour "création utilisateur""  Une des fonctionnalités demandées était la création des utilisateurs et de pouvoir définir le type d’utilisateur (comptable / visiteur) en ajoutant les informations correspondantes à son type. |

L’intérêt de ce formulaire unique est de pouvoir insérer un utilisateur et de choisir son type (visiteur / comptable) et de lui attribuer une voiture si celui-ci est un visiteur. Voici schématiquement ce qu’il se passe à la création d’un visiteur ou d’un comptable :



Ici deux procédures stockées sont utilisées, en effet lors de la vérification de la disponibilité d’un véhicule on fait appel à la procédure stockée nommée « veirfAdd\_Voiture » qui à quel moment la voiture pourra être insérer.  
Par la suite on utilise pour visiteur comme pour compte la procédure stockée « Insert\_Update\_User » qui permet d’insérer un utilisateur en fonction des paramètres envoyés.

### Formulaire de réinitialisation de mot de passe

|  |
| --- |
| Résultat de recherche d'images pour "password"" Une des fonctionnalités demandées était la réinitialisation d’un mot de passe, pour ce faire un formulaire a été créée afin de pouvoir réinitialiser le mot de passe d’un utilisateur sélectionné. |

# Note évolutive

|  |
| --- |
| Résultat de recherche d'images pour "amélioration logo"" L’application est fonctionnelle est peut être utilisé par les dirigeants de la GSB, mais celle-ci peut être améliorer sur certains points qui seront énoncés. |

* Procédures stockée et trigger (utiliser)
* Création d’un utilisateur
* Manipulation des voitures
* Trigger de vérif comptable et visiteur
* PS pour voir tous les emprunts d’un visiteur
* Etc etc
* Note évolutive (calendrier pour choisir un intervalle de dates disponibles (rouge non dispoibles etc) mettre image

Conclusion

|  |  |
| --- | --- |
| Source développement de l’application | |
| *Chiffrement des informations :* | |
| [*Carré de Vigenère*](https://www.apprendre-en-ligne.net/crypto/vigenere/carrevig.html) | *Explication du carré de Vigenère* |
| [*Tableau de Trithème*](https://www.apprendre-en-ligne.net/crypto/tritheme/polyalpha.html) | *Explication du tableau de Trithème* |
| [*Les bases du cryptage*](https://bidou.developpez.com/tutoriels/dotnet/securite/crypto/) | *Explication des méthodes de cryptage* |
| [*Chiffrement sous VB.Net*](https://www.youtube.com/watch?v=NP2tQo68zZM) | *Vidéo YouTube expliquant l’encryptage sous VB.NET* |
| [*Explication chiffrement VB.Net*](https://docs.microsoft.com/fr-fr/dotnet/visual-basic/programming-guide/language-features/strings/walkthrough-encrypting-and-decrypting-strings) | *Documentation Microsoft sur le cryptage* |
| [*Les différents types de cryptages*](http://www.primenumbers.net/Renaud/fr/crypto/Vigenere.htm) | *Explique différents procéder de cryptage / Chiffrement / Encodage / etc.* |
| [*Chiffrement*](https://chiffrer.info/) | *Site expliquant les différences entres les différents termes employés pour le chiffrement* |
| *Utilisation de la commande GRANT en SQL :* | |
| [*La restriction de privilèges*](https://buzut.net/maitrisez-mysql-en-cli/#Gestion-des-utilisateurs) | *Base sur les commandes (Utiles pour les commandes de restriction de privilèges)* |
| [*Gestion des utilisateurs*](https://openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql/1975880-gestion-des-utilisateurs) | *Cours sur la gestion des utilisateurs (OpenClassRooms)* |
| [*Droit d’accès à une BDD*](https://cyberzoide.developpez.com/securite/privileges-base-de-donnees/) | *Explique les différents droits d’accès à une BDD* |
| [*Cours BTS-SIO (Grant)*](http://www.bts-sio.com/cours/SI3/sql/8%20-%20GRANT.php) | *Explication et syntaxe des commandes de restriction et d’autorisations de privilèges sur une Base de données* |
| [*Restriction des privilèges (REVOKE)*](https://www.techonthenet.com/sql_server/grant_revoke.php) | *Explication sur la restriction des privilèges* |
| *Fichier :* | |
| [*Fichier .ini*](https://fr.wikipedia.org/wiki/Fichier_INI) | *Wikipédia expliquant les intérêts et l’utilisation d’un fichier .ini* |
| [*Lecture d’un fichier .txt*](https://docs.microsoft.com/fr-fr/dotnet/visual-basic/developing-apps/programming/drives-directories-files/how-to-read-from-text-files) | *Explication du procédé pour lire un fichier texte avec VB.Net* |
| [*Écriture dans un fichier .txt*](https://docs.microsoft.com/fr-fr/dotnet/visual-basic/developing-apps/programming/drives-directories-files/writing-to-files) | *Explication du procédé pour pouvoir écrire à l’intérieur d’un fichier avec VB.Net* |
| *Application MDI :* | |
| [*Utilisation et création*](http://pise.info/vb-net/17.htm) | *Explique les utilisations d’un formulaire MDI* |
| *T-SQL :* | |
| [*Triggers*](https://openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql/1973090-triggers) | *Cours Open Class Rooms expliquant l’intérêt et l’utilisation des Triggers* |
| [*Procédures stockées*](https://openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql/1971667-procedures-stockees) | *Cours Open Class Rooms expliquant l’intérêt et l’utilisation des Procédures stockées* |
| Lien utiliser pour a mise en place et la configuration des serveur | |
| Windows Server 2019 : | |
| [Installation d’un Windows Server 2019](https://www.toutwindows.com/ws2019-installation/) | Site expliquant pas à pas l’installation d’un Windows Server 2019 |
| SQL Server : | |
| [Installation de SQL Server 2019](https://www.youtube.com/watch?v=KFvNgxa5F8o) | Vidéo tuto YouTube expliquant l’installation de SQL Server |
| Nagios : | |
| [Éléments](https://openclassrooms.com/fr/courses/2035786-mettez-en-place-un-outil-de-supervision-de-production-avec-nagios/6050285-decouvrez-les-elements-constitutifs-de-nagios) | Site de Open Class Rooms expliquant les éléments constitutifs de Nagios |
| [Installation](https://www.itzgeek.com/how-tos/linux/debian/how-to-install-nagios-on-debian-9-stretch.html) | Site expliquant l’installation de Nagios sur une machine Debian |
| [Commande d’installation](https://support.nagios.com/kb/article/nagios-core-installing-nagios-core-from-source-96.html#Debian) | Site répertoriant toutes les commandes nécessaires à l’installation de Nagios |
| [Installation et configuration](https://djibril.developpez.com/tutoriels/linux/nagios-pour-debutant/) | Site développez.com expliquant la configuration et l’installation de Nagios |
| [Installation, configuration et mise en place](https://www.supinfo.com/articles/single/1630-mise-place-une-solution-supervision-avec-nagios-centreon) | Site de Supinfo expliquant en détails les étapes à suivre pour mettre en place une solution de supervision |