



Compte rendu

Projet n°6 : Application GSB-Admin et Intégrité des intranets

Groupe 2

Coindevel Bryan, Jean Loïc Augustin, Emmanuel Lebeau, Treport Elie, Valentin Samuel

Table des matières

I) Rappel du cahier des charges.....	3
II) Partie gestion de projet.....	3
1) Planification (Gantt)	3
2) Organisation du travail.....	5
3) Estimation du projet.....	6
III) Partie technique	6
1) Chiffrement	6
a) Introduction.....	6
b) Chiffrement de VB.net	8
c) Carré de Vigenère	9
2) Base De Données.....	11
a) Création de la Base De Données	11
b) Restriction des privilèges	14
c) Les triggers (déclencheurs	15
d) Procédures stockées	16
3) Application	17
a) Introduction.....	17
c) Fichier de configuration .ini	18
d) Formulaire de connexion	19
Scénario nominal :	19
e) Formulaire de modification du fichier .ini.....	20
f) Formulaire liste des utilisateurs.....	20
Scénario nominal :	22
g) Formulaire de renouvellement de mot de passe	22
Scénario nominal :	23
Formulaire de création d'un utilisateur	23
Scénario nominal :	24
IV) Note évolutive.....	40
1) Application	40
a) Formulaire de renouvellement de mot de passe	40
b) Formulaire de création d'utilisateur	40

I) Rappel du cahier des charges

Après la solution délivrée ultérieurement avec la gestion intranet. Votre entreprise nous a fait nouvelle demande :

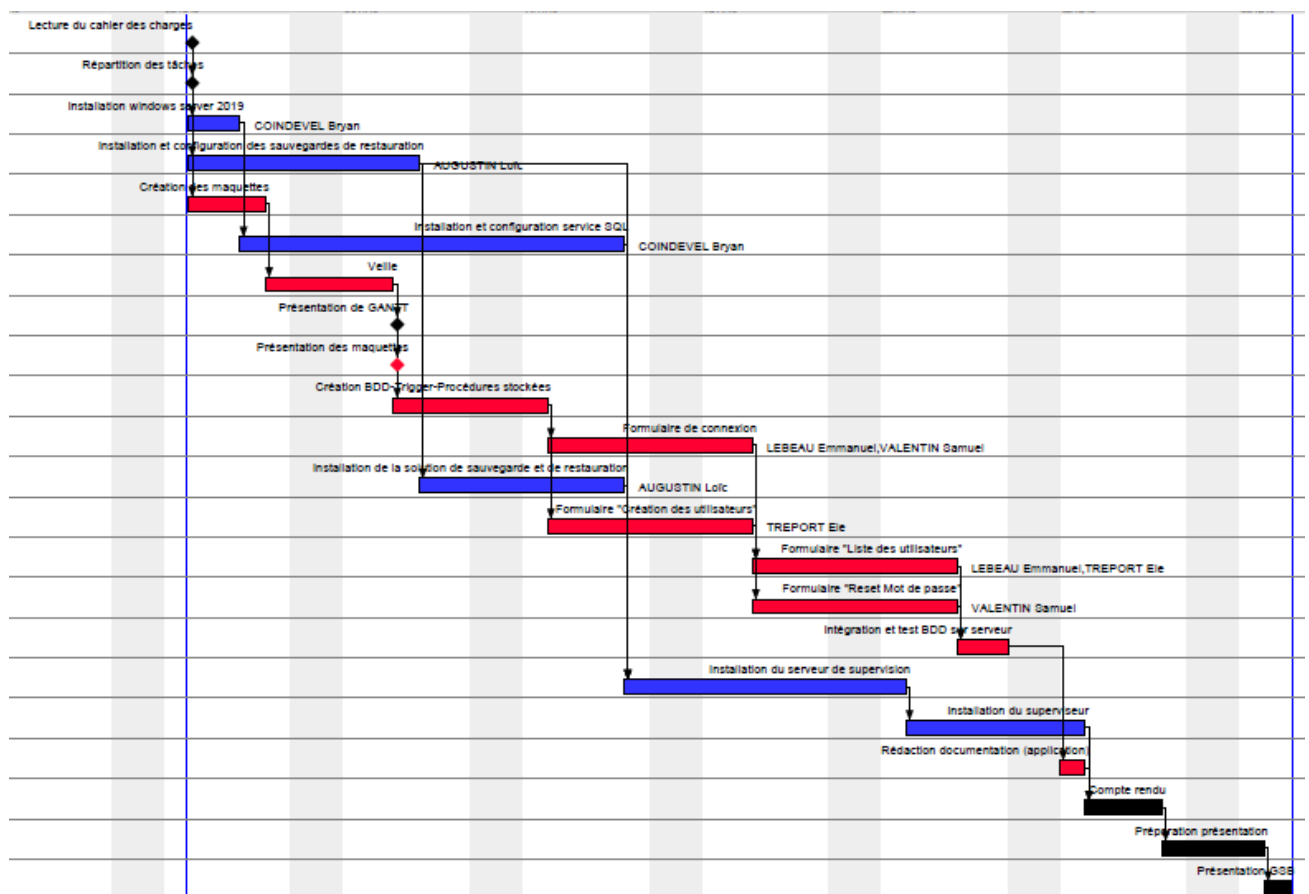
-Premièrement pour les développeurs, l'objectif était de développer un client lourd pour la gestion des utilisateurs de GSB avec pour condition d'en faire une application MDI permettant d'avoir la liste des utilisateurs, renouveler le mot de passe d'un utilisateur et de créer un nouvel utilisateur. Avec cela une certaine sécurité pour un compte super administrateur permettant ainsi toutes ces actions.

-Ensuite pour la partie réseau, le but était de mettre un nouveau serveur de sauvegarde pour stocker les sites à distances, mettre en place un serveur SQL permettant aux développeurs de faire fonctionner leur application sous un Windows Server 2019, une stratégie de sauvegarde et un serveur de supervision.

II) Partie gestion de projet

1) Planification (Gantt)

Gantt lors de la première réunion entre GSB et le groupe 2 :



Ce premier Gantt divisé les tâches en trois grandes parties :

-Celles qui étaient communes aux développeurs et aux administrateurs réseaux (couleur noire).

-Celles qui étaient communes aux développeurs (couleur rouge).

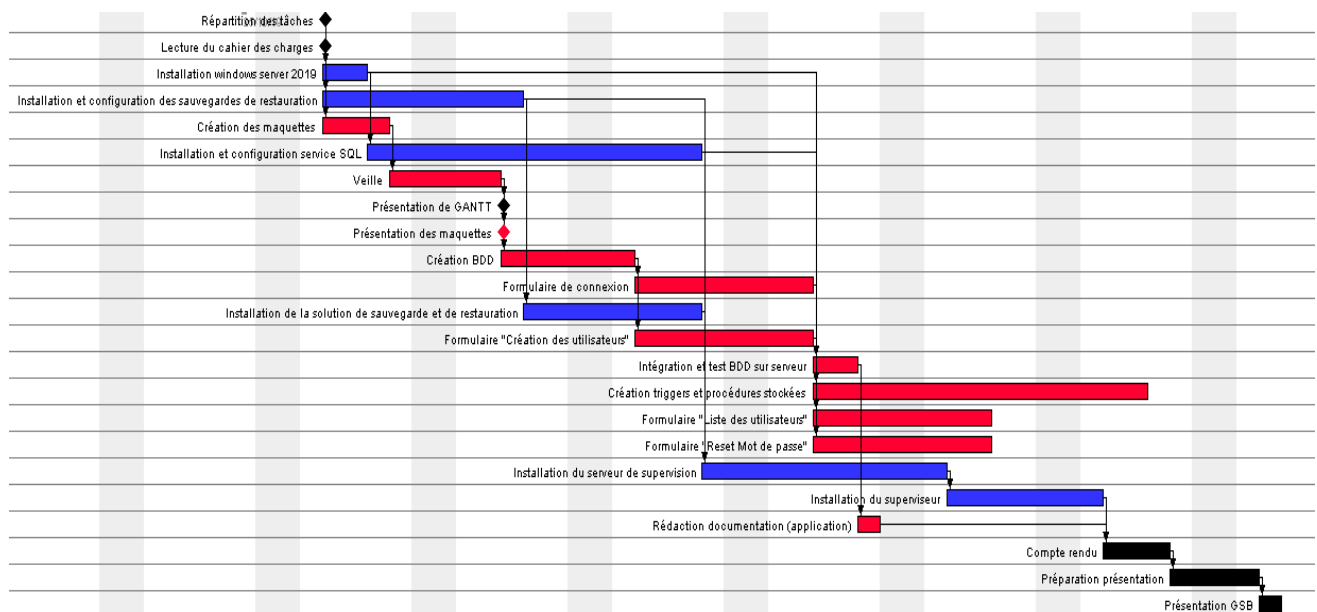
-Celles qui étaient communes aux administrateurs réseaux (couleur bleue).

Notre premier Gantt a dû subir certains changements. Premièrement avec le temps alloué sur certaines tâches comme l'installation et la configuration du service SQL qui devait se faire plus rapidement que prévu car cela aurait empêché les développeurs de tester leur intégration de la base de donnée sur le serveur et ainsi compromettre l'avancement du projet.

Il y a aussi eu des tâches qui étaient bien trop tardive comme l'intégration de la base de données sur le serveur qui s'est déroulé bien plus tôt que les administrateurs puissent continuer leurs tâches sans avoir à se soucier des problèmes qui peut être encourue lors d'une insertion de base données avec des versions différentes (Windows à Linux).

Il y a également une tâche qui a dû être séparé en deux pour la création de Base de données où on voulait créer immédiatement les déclencheurs et les procédures stockées sans attendre le codage des fenêtres de l'application. Ce qui fût être une conjecture assez illogique.

Ainsi à la fin de notre projet, cette planification n'a pas eu de changement majeure, ce qui nous a donné ceci :



2) Organisation du travail

RÉSEAU	
AUGUSTIN Jean-Loïc	Mise en place de la solution de sauvegarde
COINDEVEL Bryan	Mise en place du serveur de base de données « SQL Server » Mise en place de la stratégie de supervision
DÉVELOPPEUR	
LEBEAU Emmanuel	Création des maquettes Chiffrement des informations <ul style="list-style-type: none"> - Chiffrement des informations avec VB.NET - Déchiffrement des informations avec VB.NET Formulaire de création d'un utilisateur <ul style="list-style-type: none"> - Création d'un utilisateur (Objet) - Affichage des voitures Formulaire de la liste des utilisateurs <ul style="list-style-type: none"> - Affichage des utilisateurs - Tri des utilisateurs en fonction de leurs types Connexion à l'application <ul style="list-style-type: none"> - Lecture du fichier .ini - Chiffrement des informations du fichier - Style du formulaire de connexion
TREPORT Elie	Création des classes Chiffrement des informations <ul style="list-style-type: none"> - Création de fonctions de chiffrement et déchiffrement inspiré du chiffrement de Vigenère Formulaire de connexion <ul style="list-style-type: none"> - Récupération des informations du fichier .ini - Création et remplissage des collections avec DataReader Formulaire de la liste des utilisateurs <ul style="list-style-type: none"> - Auto-complétions de la zone de texte de recherche et affichage dans la liste Formulaire de création des utilisateurs <ul style="list-style-type: none"> - Insertion d'un utilisateur (BDD) - Emprunter un véhicule et vérification de sa disponibilité via procédure stockée Bases De Données <ul style="list-style-type: none"> - Création des procédures stockées / Triggers - Création de l'utilisateur AdminGSB et restriction des droits
VALENTIN SAMUEL	Formulaire de réinitialisation de mot de passe <ul style="list-style-type: none"> - Affichage des utilisateurs dans une liste

	<ul style="list-style-type: none"> - Sélection d'un utilisateur - Modification du mot de passe via une procédure stockée <p>Diagramme De Gant</p> <ul style="list-style-type: none"> - Réalisation du diagramme de Gantt <p>Base De Données</p> <ul style="list-style-type: none"> - Création de la Base De Données
--	---

3) Estimation du projet

Début du projet	Mardi 29 Octobre 2019
Fin du projet	Mardi 03 Décembre 2019
Nombre d'effectifs sur le projet	5
Nombres d'heures effectuées sur le projet :	
Coindevel Bryan	24
Jean Loïc Augustin	24
Lebeau Emmanuel	24
Treport Elie	24
Valentin Samuel	24
Heures effectués en totale	120
Charges patronales (5 personnes)	5000€ (projet sur un mois)
Charges salariales Totale (Net)	5143€
Coût total du projet	10 143€

III) Partie technique

1) Chiffrement

a) 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.



Suite aux demandes de Galaxy Swiss Bourdin (GSB) nous avons effectués une veille technologique sur la cryptologie (L'art de l'écriture cachée, ou science du secret), la cryptographie (le chiffrement) et la cryptanalyse (le déchiffrement).

Il existe de nombreux types de chiffrement qui utilisent différents types de clés, il existe plusieurs types d'utilisations 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.

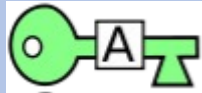
Ainsi sans cette clé, le message a peu de chance d'être déchiffré, il faut donc pour pouvoir le déchiffrer avoir la clé utilisée pour le chiffage du message.



Clé
asymétrique

Une clé asymétrique est un système qui fonctionne avec deux clés différentes :

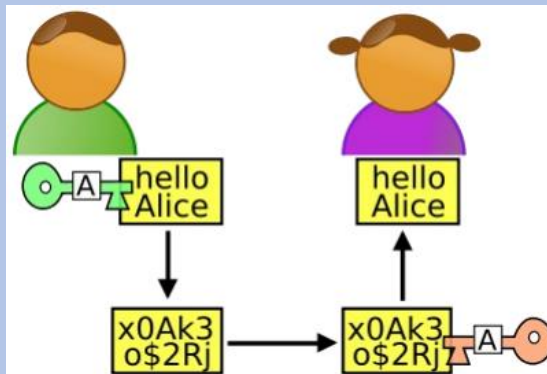
Une clé publique



Une clé privée

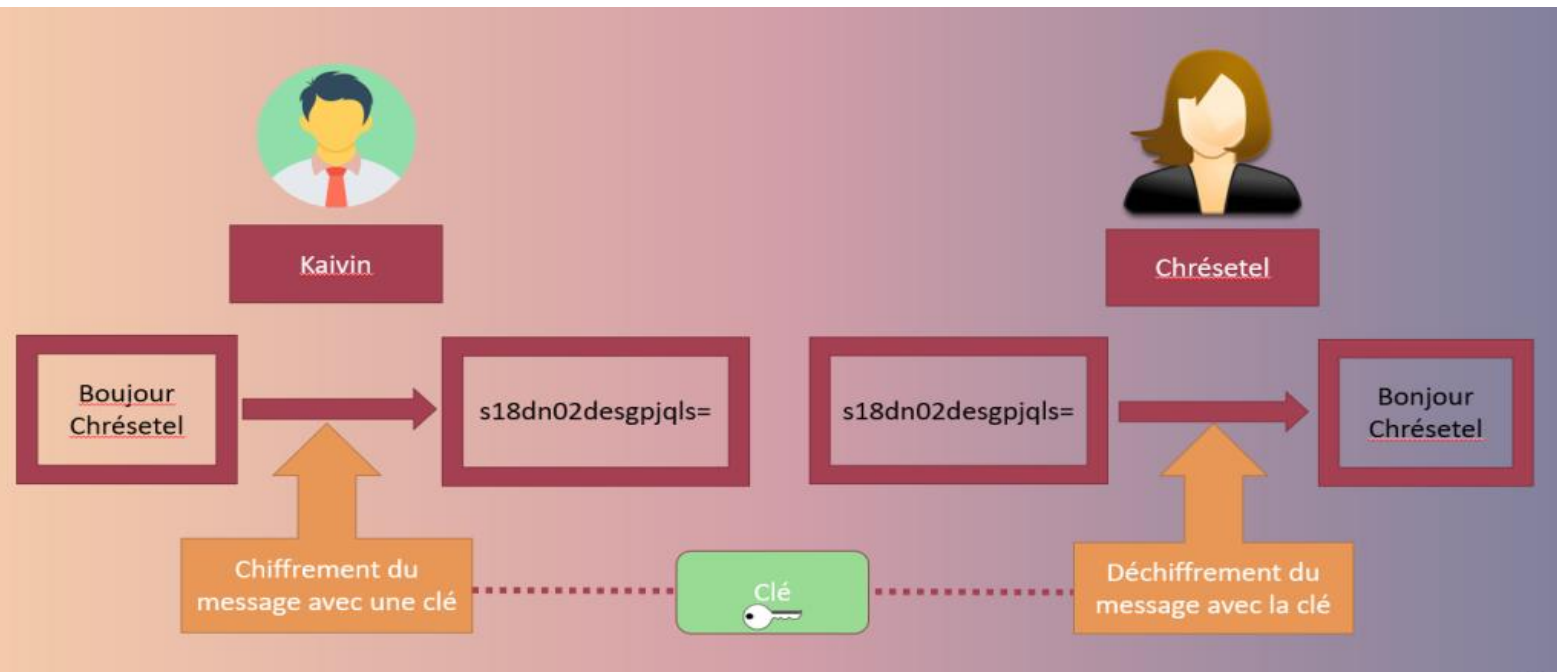


La clé privée comme son nom l'indique n'est pas partagée, elle permet de déchiffrer le message. Elle est donc plus utile dans le cadre d'une communication.



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

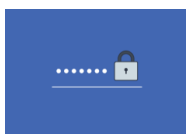
Ainsi le chiffrement à clé symétrique est utilisé dans l'application GSB-Admin, de deux manières différentes. L'utilisation de la clé symétrique utilisée peut être assimilée au schéma ci-dessous :



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 prévient le fait que si un tiers intercepte les données chiffrées, il lui sera donc difficile de les déchiffrer.

b) Chiffrement de VB.net

Le premier chiffrement intégré à VB.Net utilise la classe System.Security.Cryptography. Cette classe fournit plusieurs services de chiffrement qui comprennent le chiffrement et le déchiffrement des chaînes.



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

Pour pouvoir chiffrer notre chaîne de caractères 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 :

```
Dim crypte = cryptValue("12-Soleil&")
```

Cela nous donne ce résultat :

```
crypte "L6IM9llo3fxz8x2RQg7hUg==" {String}
```

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

```
Dim decryptage = deryptValue(crypte)
```

Cela nous retourne ce résultat :

```
decryptage "12-Soleil&" {String}
```

De cette manière on obtient une chaîne de caractères 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.

c) Carré de Vigenère

Le chiffrement de Vigenère a été créé 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 cassé qu'en 1863 par Friedrich Kasiski soit 3 siècles plus tard.



Le chiffrement de Vigenère est une méthode permettant de chiffrer une chaîne de caractères 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 entre 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 d'A à Z comme ci-dessous.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

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 :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S

Ce chiffrement a été utilisé au sein de l'application GSB-Admin mais a été modifié, en insérant dans ce carré les lettres minuscules ([a-z]), majuscules ([A-Z]), les chiffres ([0-9]), ainsi que les caractères spéciaux (², /). Ainsi grâce à cette amélioration et un ordonnancement aléatoire entre chaque caractère. La technique du chiffrement de Vigenère a été améliorée et personnalisée permettant donc un plus grand nombre de possibilités. En effet avec un total de 108 caractères on a donc un total de 108^x 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 contenu 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 appelle notre fonction :

```
Dim cryptage = cryptage_carré_Vigenère("12-Soleil&")
```

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

```
► cryptage "NVéK)NJP=H" {String}
```

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 :

```
Dim decryptage = decryptage_carré_Vigenère(cryptage)
```

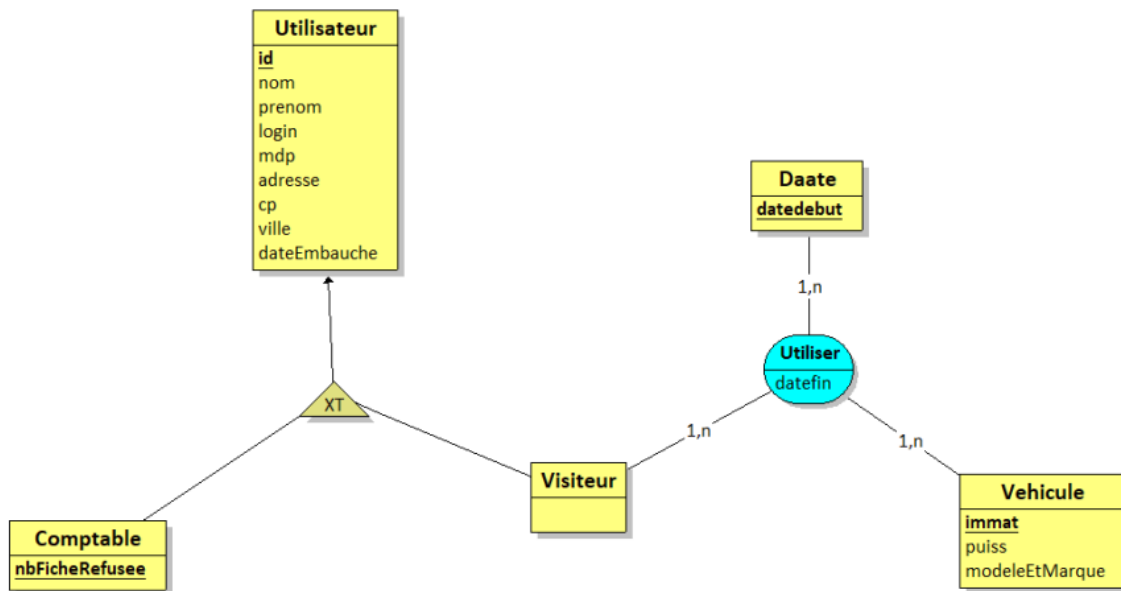
Celle-ci nous retourne la valeur de la chaîne déchiffrée qui est similaire à celle chiffré au début de ce processus :

```
► decryptage "12-Soleil&" {String}
```

2) Base De Données

a) Création de la Base De Données

La base de données proposé dans le cahier des charges n'a pas été 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
    
```

Utilisateur
id
nom
prenom
login
mdp
adresse
cp
ville
dateEmbauche

- 2- 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
```

Visiteur

- 3- 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]
```

Comptable
nbFicheRefusee

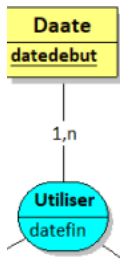
- 4- On a ensuite créé la table « vehicule » contenant les informations en lien avec un véhicule :

```
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
```

Vehicule
immat
puiss
modeleEtMarque

- 5- 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 créée 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 créées il a fallu mettre les contraintes existantes entre 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
```

- 2- 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 la relation d'héritage fût créée.

- 3- 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
```

- 4- 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
```

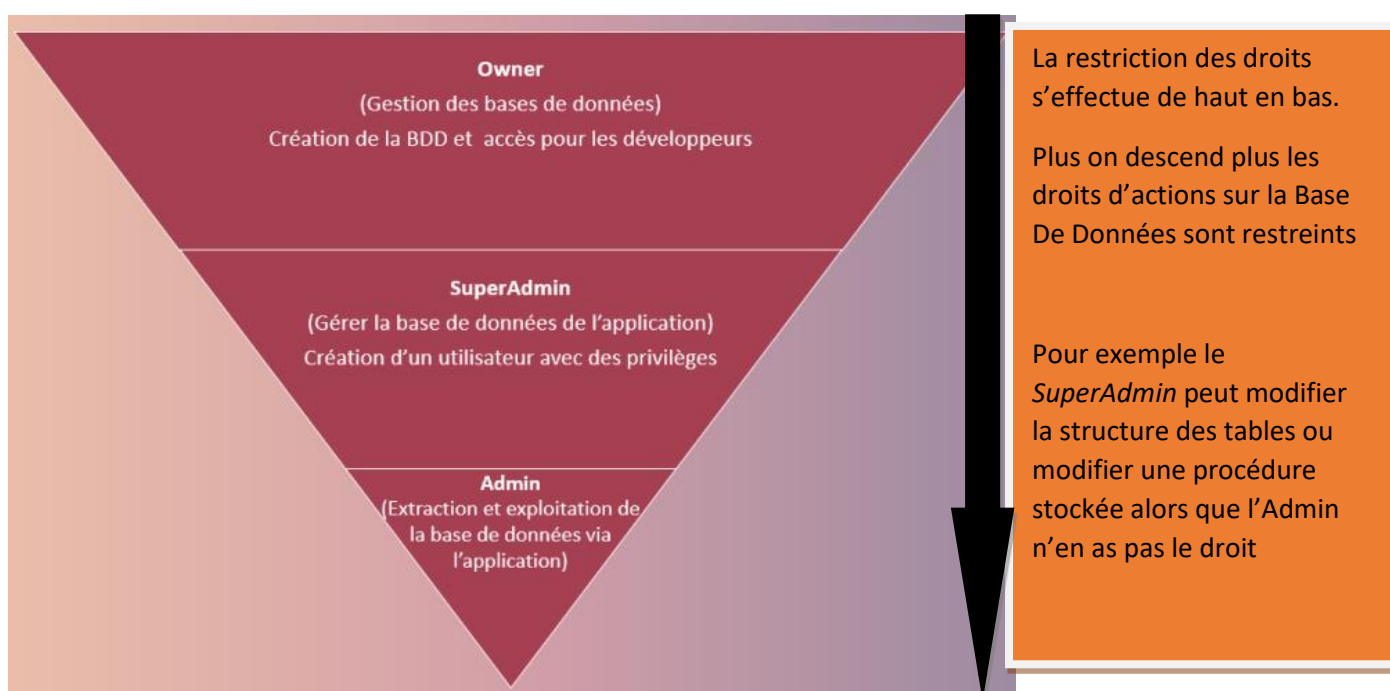
Grâce à cela, on pourra retrouver les emprunts d'un véhicule pour un visiteur, et sur cet emprunt la date de début et la date de fin de l'emprunt.

b) Restriction des privilèges

Après avoir créé la Base De Données grâce aux identifiants de connexion fournis 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.



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 créé, celui-ci n'a pour l'instant aucun droit et peut que se connecter à la BDD sans pouvoir faire d'actions. 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 est destinée à utiliser des procédures stockées présentes sur la BDD GSB_Admin. il a 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;
```

c) Les triggers (déclencheurs)

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 créés sur certaines tables de la BDD :

1- Trigger après suppression sur la table « comptable »

```
-- Permet de supprimer l'utilisateur concerné 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.

2- Trigger après suppression sur la table « visiteur »

```
-- Permet de supprimer l'utilisateur concerné 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 ».

3- 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 par rapport à la cohérence des attributions de voitures à une date donnée.

d) Procédures stockées

Une procédure stockée peut être assimilée 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 appelée avec ou sans valeur en paramètre 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 besoin, et celle-ci se chargera de nous effectuer les opérations et les calculs voulus sur la Base De Données.

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

- 1- La première procédure stockée nommée 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.

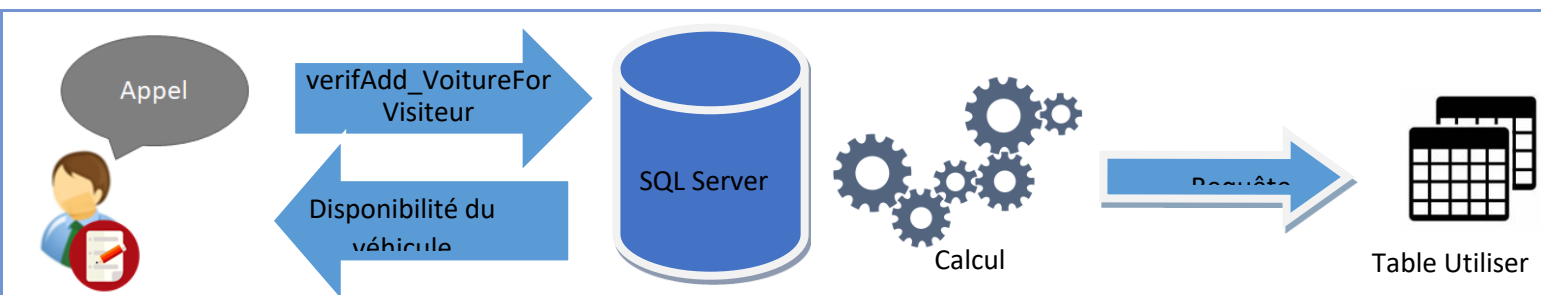
```
CREATE PROCEDURE Insert_Update_User
@param_id INT,
@param_nom nvarchar(50),
@param_prenom nvarchar(50),
...
```

END

- 2- 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
```

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 :

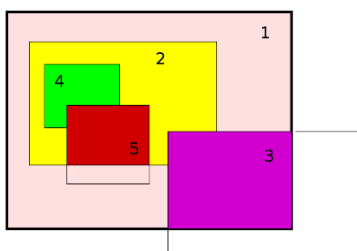


3) Application

a) Introduction

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 dirigeant de GSB seront contenues dans une fenêtre comme nous le montre ce schéma ci-dessous :



- 2- Avoir les identifiants de connexion à l'application ET à la base de données dans un fichier crypté
- 3- Avoir une Base De Données crypté sur les données sensibles

c) Fichier de configuration .ini

Le fichier de configuration permettant de se connecter 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.



```
; 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 nommée « organization ».

Ces 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 a donc été intégré 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=u0MUr6iQvqH8r53GVnh81w==
motdepasse=hGs9UDdVxB2RAWpg5L9IEA==
[Database]
serveur=rHwVLpTnrw6XGJaCue87Ao6/ld00G076
baseDeDonnees=PwHrqFDD6/PtlyD8Cpvukg==
user=3RA/vMorp1s4mf8QYKsEMQ==
mdpUser=hGs9UDdVxB2RAWpg5L9IEA==
```

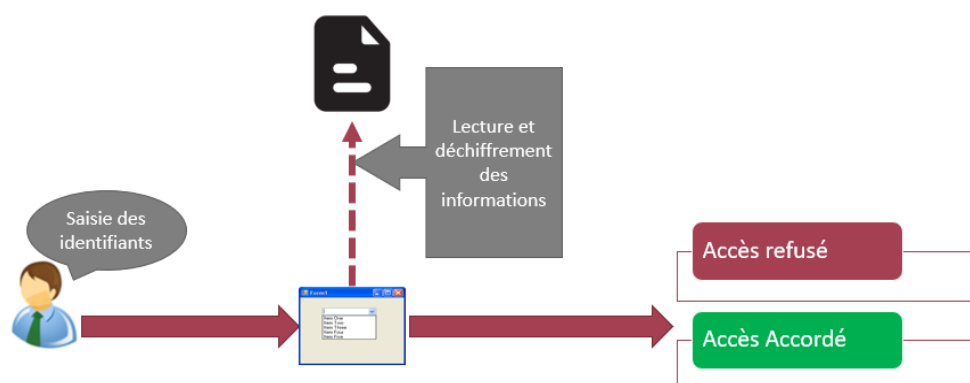
Ce fichier permet ainsi à l'application de récupérer pour la section « Auth » les identifiants de connexion, par ailleurs pour la section « Database » on récupère les informations de connexion à la Base De Données distantes.

d) Formulaire de connexion

Le formulaire de connexion est le premier formulaire qui apparaîtra 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 ».



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



Grille d'écran :

Cas d'utilisation :

Nom cas d'utilisation : Connexion à l'application GSB-Admin
Acteur déclencheur : Admin GSB
Pré conditions : néant
Post conditions : Login et mot de passe valide.
Scénario nominal : 1. L'administrateur entre son login et son mot de passe 2. L'administrateur choisit s'il veut modifier les paramètres de connexion ou se connecter à l'application (il se connecte par défaut à l'application) 3. L'administrateur valide sa sélection 4. Le système authentifie l'administrateur et lance l'application MDI 5. L'administrateur choisit de quitter l'application 6. L'application se ferme.

Exceptions :
3. a- L'administrateur a mal saisi ses informations : Le système lui signale « Login et/ou mot de passe incorrects... » - Retour à 1. 2→3. a- L'administrateur choisit qu'il veut modifier les paramètres de connexion : Le système envoie l'administrateur sur une page auxiliaire à l'application principale permettant de modifier les paramètres de la base de données, le login de l'administrateur et son mot de passe.
Contraintes :

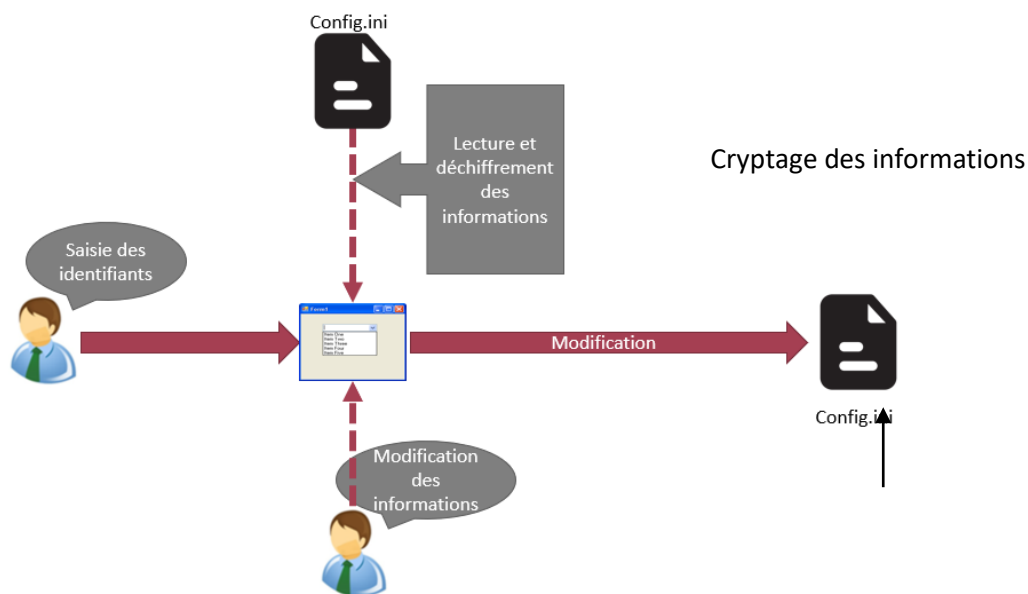
e) Formulaire de modification du fichier .ini

Comme le fichier de configuration permettant de se connecter à l'application et à la Base De Données peut être susceptible 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.



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 :



f) Formulaire liste des utilisateurs

Grille d'écran :

Liste des utilisateurs

Les utilisateurs

☒ Tous
☐ Visiteur
☐ Comptable

Recherche :

N°	Nom	Prénom	Date d'embauche

Réinitialiser le mot de passe

Cas d'utilisation :

Nom cas d'utilisation : Informations sur les utilisateurs
Acteur déclencheur : Admin GSB
Pré conditions : Admin GSB authentifié
Post conditions : néant

Scénario nominal :

1. L'administrateur peut consulter les informations des utilisateurs dans une liste qui s'affiche étant complété automatiquement.
2. L'administrateur double clic sur l'utilisateur pour avoir des informations supplémentaires.
3. L'administrateur choisit sur l'un des boutons radios la catégorie qu'il veut afficher - en fonction de l'option choisit, des utilisateurs différents vont être affichés (Afficher tout le monde, que les visiteurs ou que les comptables).
4. L'administrateur recherche dans la zone de texte un utilisateur : il peut rechercher par rapport au nom, au prénom voire la date d'embauche – par rapport aux informations choisit, la liste des utilisateurs change d'informations pour que cela corresponde aux informations données dans la zone de texte.
5. L'administrateur fait un clic droit sur l'utilisateur choisit : il peut ainsi modifier les informations de l'utilisateur, le système l'envoie sur un autre formulaire - l'administrateur clic sur 'supprimer', le système demandera une confirmation et l'utilisateur sera supprimé.
6. L'administrateur clic sur le bouton 'Réinitialiser le mot de passe' – Le système le renvoie sur le cas de modification de mot de passe ayant comme utilisateur déjà sélectionné l'utilisateur choisit.

Exceptions :

Contraintes :

6. a- L'administrateur doit forcément avoir cliqué sur un utilisateur avant de pouvoir envoyer les informations sur le formulaire de renouvellement de mot de passe

g) Formulaire de renouvellement de mot de passe

Grille d'écran :

Réinitialisation d'un mot de passe

N°	Nom	Prénom	Date d'embauche
1			

3

Mot de passe ☐ Afficher/Masquer

Confirmer le nouveau mot de passe 4

Le mot de passe doit contenir minimum :

- 8 caractères
- 2 minuscules
- 2 majuscules
- 2 chiffres
- 2 caractère spécial

Confirmer 2

Valider 6

Par défaut 5

Cas d'utilisation :

Nom cas d'utilisation : Modification de mot de passe

Acteur déclencheur : Admin GSB

Pré conditions : Admin GSB authentifié

Post conditions : néant

Scénario nominal :

1. **L'administrateur choisit parmi la liste des utilisateurs celui dont il veut changer le mot de passe.**
2. **L'administrateur valide sa sélection.**
3. Le système montre des informations supplémentaires sur la droite. Il permet ainsi la saisie du nouveau mot de passe et retourne en haut le nom de l'utilisateur choisi.
4. **L'administrateur saisie le nouveau mot de passe avec une confirmation.**
5. **L'administrateur choisit d'utiliser le mot de passe par défaut.**
6. **L'administrateur valide la modification**
7. Le système enregistre la modification du mot de passe.

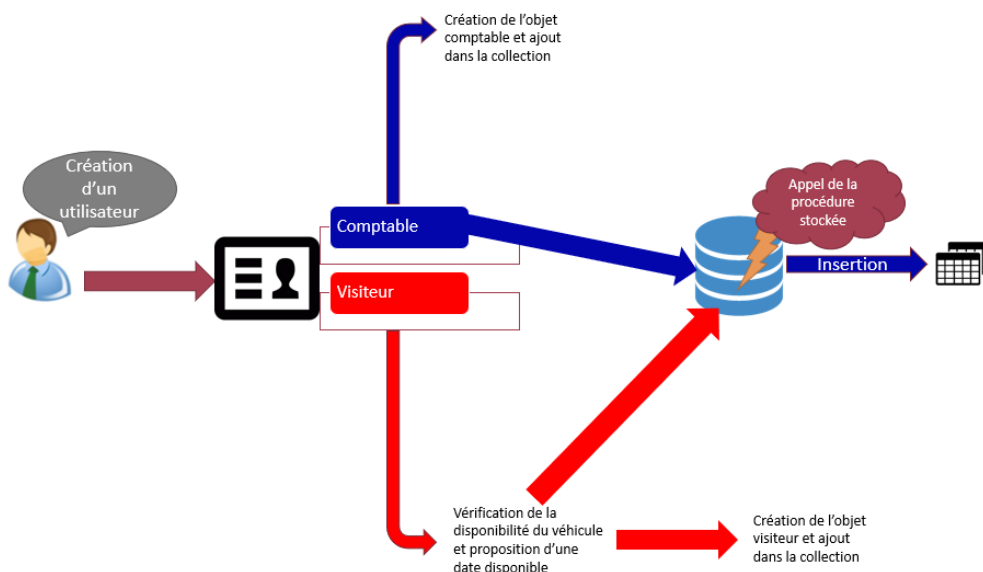
Exceptions :

2. a- **L'administrateur sélectionne un autre utilisateur et le valide : le système demande 'Vous êtes déjà en train de modifier le mot de passe d'un autre utilisateur, êtes-vous sûr de changer votre choix ?' Si oui - retour à 3. Si non – annulation du 2.**
6. a- L'un des champs n'est pas renseigné : le système indique ' les mots de passe ne correspondent pas ! '. Retour à 4.
6. b- Le mot de passe ne correspond pas aux critères de sécurité : le système demande 'Ce mot de passe ne correspond pas aux critères de sécurité, voulez-vous réellement mettre à jour ?' Si oui – la modification continue. Si non – retour à 4.
6. c- Les mot de passe est bon : Le système demande 'Voulez-vous vraiment mettre à jour le mot de passe ?' Si oui – avancer à 7. Si non – retour à 4.

Contraintes :

Formulaire de création d'un 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.



Grille d'affichage :

The screenshot shows a web form for creating a user. It is divided into several sections:

- Informations:** Contains input fields for Nom, Prénom, Adresse, Code Postal, Ville, and Date d'embauche (set to mardi 10 décembre 2019). A callout box labeled '1' points to this section.
- Type d'utilisateur:** Contains radio buttons for 'Visiteur' and 'Comptable'. A callout box labeled '2' points to this section.
- Location:** Contains input fields for Date de début de la location and Date de fin de la location, both set to mardi 10 décembre 2019.
- Authentification:** Contains input fields for Login, Mot de passe, and Confirmation du mot de passe. There is an 'Afficher' checkbox next to the password field. Below these fields, a note states: 'Le mot de passe doit contenir minimum : 8 caractères, 2 minuscules, 2 majuscules, 2 chiffres, 2 caractère spécial'. A callout box labeled '4' points to the 'Valider' button at the bottom of this section.
- Voitures:** A table with columns 'Matricule', 'Puissance', and 'Marque'. A callout box labeled '3' points to the table header. Below the table, there is a checkbox labeled 'Afficher uniquement les voitures disponibles' which is checked.

Cas d'utilisation :

Nom cas d'utilisation : Création des utilisateurs
Acteur déclencheur : Admin GSB
Pré conditions : Admin GSB authentifié
Post conditions : néant
<p>Scénario nominal :</p> <ol style="list-style-type: none"> L'administrateur entre les informations demandées dans le formulaire. L'administrateur choisit si l'utilisateur créé est un visiteur ou un comptable – Si l'utilisateur est un comptable, il n'y aura aucune information supplémentaire. Si l'utilisateur est un visiteur alors le système affichera des informations supplémentaires à droite pour l'attribution de voiture sur une période donnée. L'administrateur entre les informations supplémentaires si l'utilisateur est un visiteur (date de location et attribution du véhicule) L'administrateur valide les informations données : Le système enregistre les données
<p>Exceptions :</p> <ol style="list-style-type: none"> a- L'administrateur n'a pas choisi de voiture pour le visiteur : Le système demande à l'utilisateur s'il est sûr de l'ajout. Si oui l'ajout s'effectue – sinon retour à 3 b – L'administrateur a choisi une date où le véhicule est déjà attribué : Le système va vérifier une date alternative où il peut attribuer la voiture dans l'intervalle de jours donnée, puis il demande si

l'administrateur veut attribuer ce véhicule à cette date proposée : Si oui le véhicule est attribué à l'intervalle donnée par l'application – Si non envoi à 4.

4.a Les informations par rapport au mot de passe ne correspondent pas : Le système signale à l'administrateur que les informations ne sont pas bonnes et l'enregistrement n'est pas fait.

Contraintes :

IV) Mise en place des solutions de sauvegarde

Sous BACULA :

Pour le serveur web :

1 ^{er} Dimanche	2 ^{ème} Dimanche	3 ^{ème} Dimanche
Complète	Différentiel	Incrémentale

Pour la base de données :

Une sauvegarde complète sera faite tous les midis.

Sous SQL server :

Un script se lance tous les jours pour sauvegarder la base de données.

Mise en œuvre :

Bacula :

« Bacula » un logiciel open source de sauvegarde qui permet de vérifier et de restaurer des fichiers en mode client-serveur dans un réseau.

Bacula fonctionne sur la base de 5 fichiers attribuer à un rôle précis :

- Le fichier général de configuration le « Director »
- Le fichier « Storage Daemon » concernant le stockage des données
- Le fichier « File Daemon » permettant l'accès aux données installé sur les clients à sauvegarder
- Le fichier « bconsole » permettant la connexion entre la console et le directeur
- Et le fichier de configuration du moniteur de ressources, pas utilisé dans notre cas

Installation sur le serveur de sauvegarde :

L'installation de bacula serveur se fait sur la machine de sauvegarde que l'on a nommée « HAL9000 ».

HAL9000 contient le fichier de configuration (bacula-dir.conf) ainsi que celui de stockage (bacula-sd.conf).

Le fichier de configuration « bacula-dir.conf » contient différentes parties :

- Tout d'abord nous déclarons le serveur dit « directeur », qui va gérer la sauvegarde.

```
#Serveur de sauvegarde
Director {
    Name = HAL9000-dir
    DIRport = 9101
    QueryFile = "/etc/bacula/scripts/query.sql"

    WorkingDirectory = "/var/lib/bacula"
    PidDirectory = "/var/run/bacula"
    Maximum Concurrent Jobs = 2
    Password = "12-Soleil&"
    Messages = Standard
    DirAddress = 172.18.158.151
}
```

- Les informations clients qui seront sauvegarde.

```
#####
#
# CLIENT WEB
#
#####

Client {
    Name = clientweb
    Address = 172.18.158.152
    FdPort = 9102
    Catalog = MyCatalog
    Password = "12-Soleil&"
    File Retention = 30 days
    Job Retention = 6 months
    AutoPrune = yes
}
```

```
#####
#
# CLIENT SQL SERVER
#
#####

Client {
    Name = clientsql
    Address = 172.18.158.155
    FdPort = 9102
    Catalog = MyCatalog
    Password = "12-Soleil&"
    File Retention = 30 days
    Job Retention = 6 months
    AutoPrune = yes
}
```

- Les chemins des fichiers à sauvegarder et a restauré.

```
#####
#
# FILESET POUR BACKUP /var/www/
#
#####

FileSet {
    Name = SAVEWEB_FSET
    Include {
        Options {
            signature = MD5
        }
        # Tous les fichiers depuis "/var/www/" seront sauvegardés.
        File = "/var/www/"
    }
}
```

```
#####
#
# FILESET POUR BACKUP C:/Backup/BDQ/OSB_Admin/
#
#####

FileSet {
    Name = SAVESQL_FSET
    Include {
        Options {
            signature = MD5
        }
        # Tous les fichiers depuis "C:/Backup/BDQ/OSB_Admin/" seront sauvegardés.
        File = "C:/Backup/BDQ/OSB_Admin/"
    }
}
```

```
#####
#
# FILESET POUR RESTORE /var/www/
#
#####

FileSet {
    Name = "RESTWEB_DO"

    Include {
        Options {
            signature = MD5
        }
        # Tous les fichiers depuis "/home/sauvegarde/web/" seront restaurés.
        File = "/home/sauvegarde/web/"
    }
}
```

```
#####
#
# FILESET POUR RESTORE C:/Backup/BDQ/OSB_Admin/
#
#####

FileSet {
    Name = "RESTSQL_DO"

    Include {
        Options {
            signature = MD5
        }
        # Tous les fichiers depuis "/home/sauvegarde/SQL/" seront restaurés.
        File = "/home/sauvegarde/SQL/"
    }
}
```

- Les « jobs » de sauvegarde et restauration.

```
#####
#
# JOB POUR BACKUP DU CL WEB
#
#####

Job {
    Name = BACKDONES_WEB
    JobDefs = DOMEESWEB
    Client = clientweb
    FileSet = SAVEWEB_FSET
}

JobDefs {
    Name = DOMEESWEB
    Type = Backup
    Level = Incremental
    Schedule = Domeses_sch
    Storage = storage1
    Pool = Pool_Domeses
    Write Bootstrap = /var/lib/bacula/web.bsr
    Priority = 10
    Messages = Standard
}
```

```
#####
#
# JOB POUR BACKUP DU CL SQL
#
#####

Job {
    Name = BACKDONES_SQL
    JobDefs = DOMEESSQL
    Client = clientsql
    FileSet = SAVESQL_FSET
}

JobDefs {
    Name = DOMEESSQL
    Type = Backup
    Level = Full
    Schedule = DomesesSQL_sch
    Storage = storage1
    Pool = Pool_Domeses
    Write Bootstrap = /var/lib/bacula/SQL.bsr
    Priority = 10
    Messages = Standard
}
```

```
#####
#
# JOB POUR RESTORE DU CL WEB
#
#####

Job {
    Name = RESTORE_WEB
    JobDefs = RESTWEB
    client = clientweb
    FileSet = RESTWEB_DO
}

JobDefs {
    Name = RESTWEB
    Type = Restore
    Storage = storage1
    Pool = Pool_Donnees
    Bootstrap = /var/lib/bacula/web.bsr
    Messages = Standard
}
```

```
#####
#
# JOB POUR RESTORE DU CL SQL
#
#####

Job {
    Name = RESTORE_SQL
    JobDefs = RESTSQL
    client = clientsql
    FileSet = RESTSQL_DO
}

JobDefs {
    Name = RESTSQL
    Type = Restore
    Storage = storage1
    Pool = Pool_Donnees
    Bootstrap = /var/lib/bacula/SQL.bsr
    Messages = Standard
}
```

- Les types de sauvegardes et leur différent paramètre.

```
#####
#
# Pool_Donnees
#
#####

Pool {
    Name = Pool_Donnees
    Pool Type = Backup
    Recycle = yes
    AutoPrune = yes
    File Retention = 180 days
    Job Retention = 180 days
    Maximum Volume Jobs = 1
    Label Format = "${Level}_${Job}_${Year}-${Month:p/2/0/r}-${Day:p/2/0/r}-${Hour:p/2/0/r}h${Minute:p/2/0/r}m"
}
```

- Les horaires d'exécutions des sauvegardes automatique.

```
#####
#
# SCHEDULE BACKUP CL WEB
#
#####

Schedule {
    Name = Donnees_sch
    Run = Full 1st sun at 23:05
    Run = Differential 2nd-5th sun at 23:05
    Run = Incremental mon-sat at 21:00
}

#####
#
# SCHEDULE BACKUP CL SQL
#
#####

Schedule {
    Name = DonneesSQL_sch
    Run = Full at 12:05
}
```

- Les informations sur la machine de stockage.

```
#####
#
# ESPACE DE STOCKAGE DES BACKUPS
#
#####

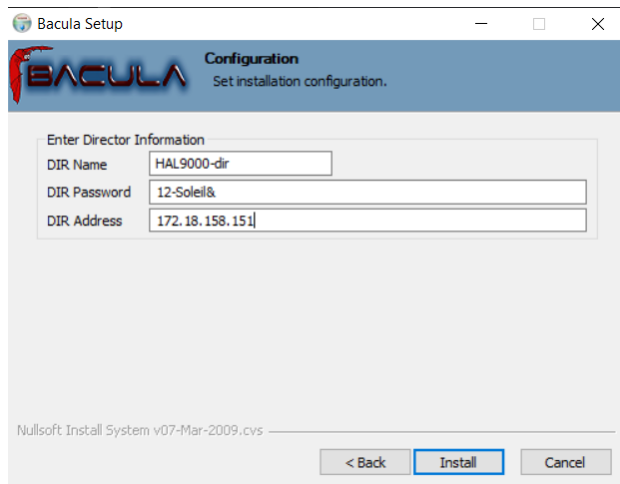
Storage {
    Name = storage1
    Address = 172.18.158.151
    SDPort = 9103
    Password = "12-Soleil&"
    Device = Disk1
    Media Type = File
}
```

Le fichier « bacula-sd.conf » donne les informations du serveur de stockage des sauvegardes a bacula :

Installation sur les machines clients :

Sous la machine linux nous avons installé bacula version client et configurons le fichier « bacula-fd.conf ».

Sous la machine Windows bacula dispose d'un exécutable à installer, lors de l'installation nous précision le nom, mots de passe et adresse IP du serveur directeur « HAL9000 ».



Que ce soit sous linux ou Windows la configuration du fichier « bacula-fd.conf » reste la même :

```
#Déclaration du serveur directeur
Director {
  Name = HAL9000-dir
  Password = "12-Soleil&"
}

#Déclaration du serveur directeur local
Director {
  Name = INTRALAB2-mon
  Password = "12-Soleil&"
  Monitor = yes
}

#Déclaration de la machine client à sauvegarder
FileDaemon {
  Name = INTRALAB2-fd
  FDport = 9102
  WorkingDirectory = /var/lib/bacula
  Pid Directory = /run/bacula
  Maximum Concurrent Jobs = 20
  Plugin Directory = /usr/lib/bacula
  FDAddress = 172.18.154.2
}

Messages {
  Name = Standard
  director = HAL9000-dir = all, !skipped, !restored
}
```

SQL Serveur :

Dans un premier temps nous sommes parties sur l'idée de créer un « plan de maintenance » qui exécutera une sauvegarde complète puis différentiel tout au long de la semaine, mais nous avons eu un problème de droit, alors nous nous sommes rabattues sur la solution du script de sauvegarde qui d'exécutera tous les jours.

Le script batch qui permet exécuter une requêtes SQL qui permettre de faire la sauvegarde complète :

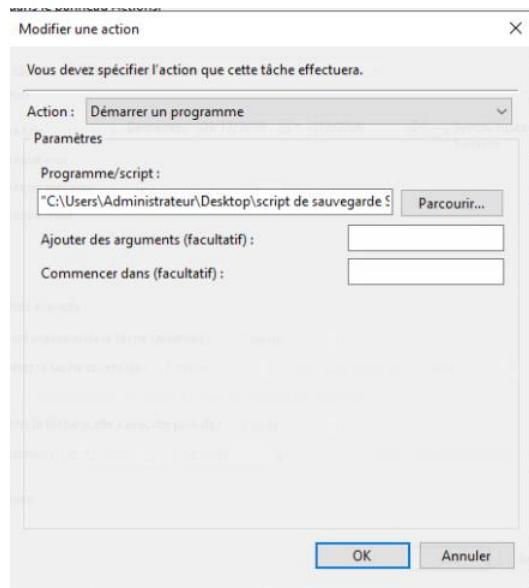
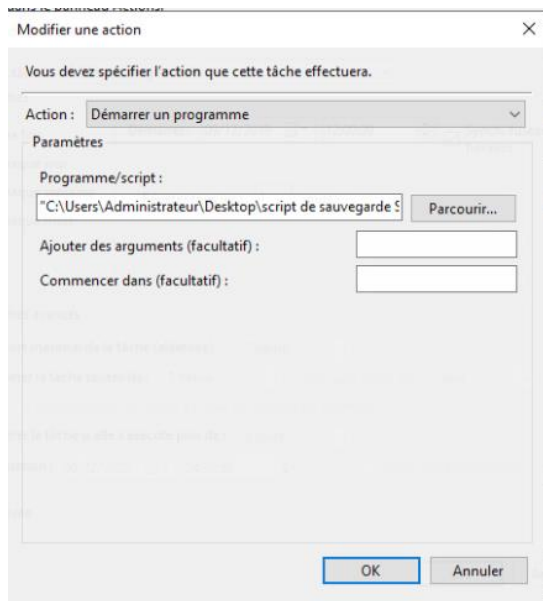
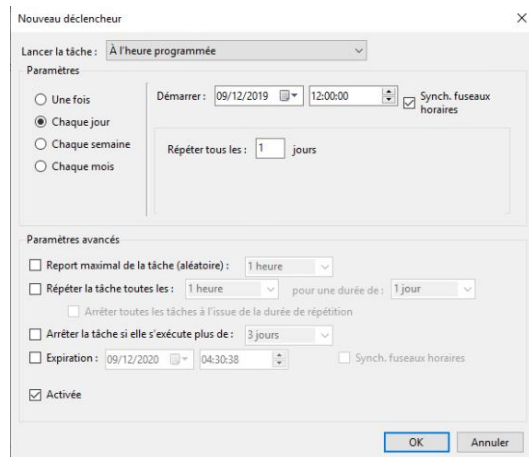
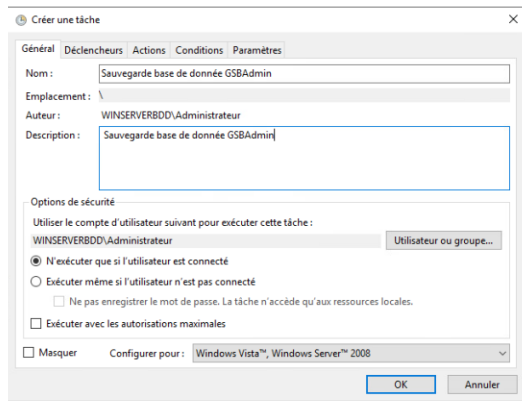
```
sqlcmd -i "backup.sql" -U "user_backup" -P "12-Soleil&" -o log_backup.txt
```

*sqlcmd permet d'exécuter des requêtes SQL grâce a un scripte .bat

La requête SQL de sauvegarde :

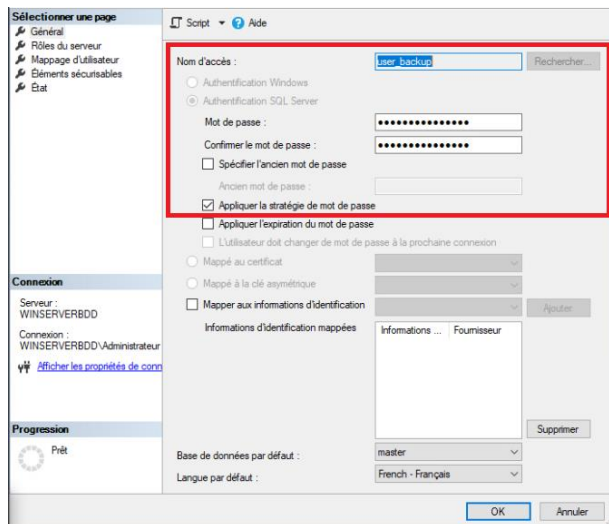
```
BACKUP DATABASE [GSB_Admin] TO DISK = 'C:\backup\GSB_Admin.bak'
WITH NOFORMAT, INIT, NAME = N'sav-base', SKIP, NOREWIND, NOUNLOAD, STATS = 10
GO
```

Pour que ce script s'exécute automatiquement à 12h tous les jours, nous avons créé une tâche planifiée :

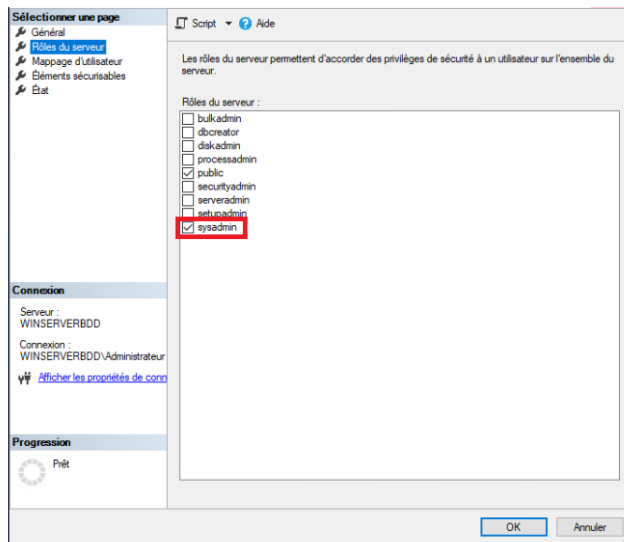


Par la suite nous avons créé un utilisateur qui pourra se connecter à la base pour la sauvegarde, cet utilisateur portera le nom de "user_backup".

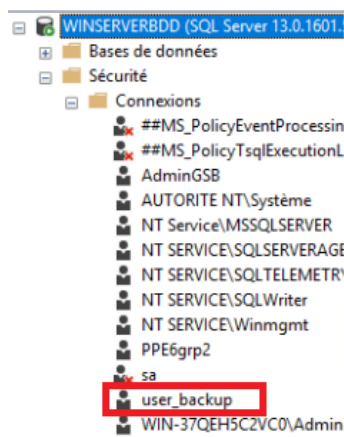
Clic droit sur "connexions" et "nouvelle connexions" :



On lui attribue les rôles pour qu'il puisse sauvegarder et restaurer la base de données "GSBAdmin" :



Résultat l'utilisateur est créé et dispose des droits nécessaires pour l'exécution de la requête SQL :



V) Mise en place du serveur de supervision

Pour avoir un œil sur tous les services et les serveurs, nous avons mis en place un service de supervision qui va surveiller en continuité l'état du système d'information et de détecter des problèmes et alerter en conséquence. Cela va permettre à l'administrateur de prévoir et anticiper le moindre problème.

Nous avons choisi comme outils de supervision « **Nagios** », car c'est des meilleurs outils de surveillance et les autres outils par exemple « **Centreon** » est basé sur Nagios.

D'abord nous installons les prérequis, pour que le service Nagios fonctionne convenablement.

```
*****Installation des prérequis*****|  
apt install build-essential unzip openssl libssl-dev apache2 php libapache2-mod-php php-gd libgd-dev
```

Avant de mettre le service Nagios sur le serveur. Nous avons mis en place un service de mail pour pouvoir recevoir des notifications sur notre boîte de mails en cas de problème d'incident sur les différents services et les serveurs.

Pour ce service de mail nous avons choisi « **Postfix** » comme pour Nagios il faudra installer les prérequis pour que postfix fonctionne.

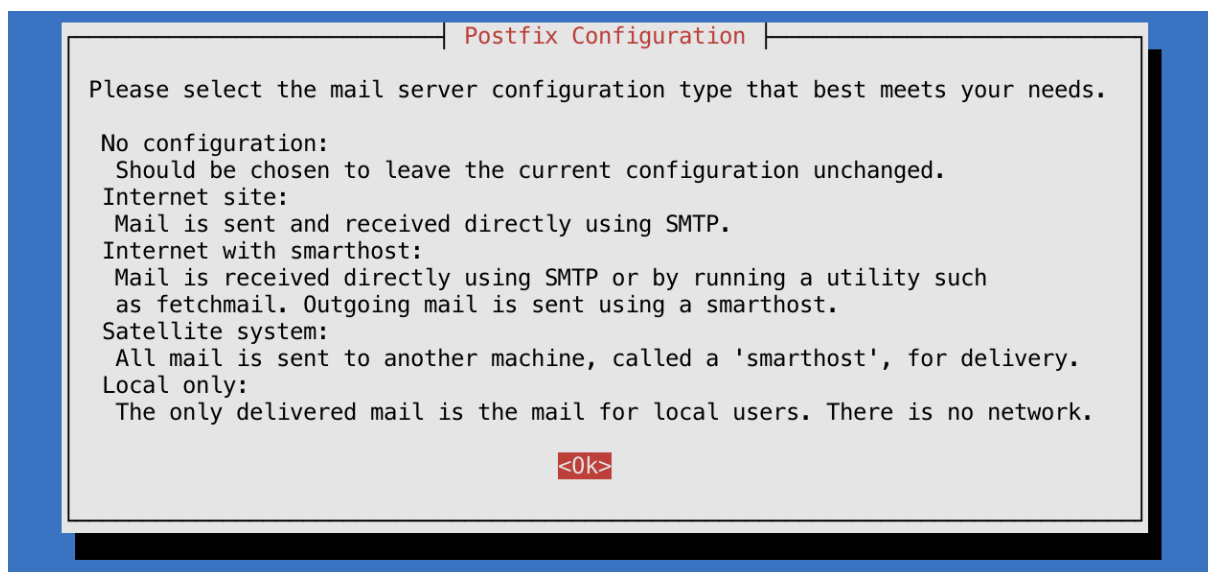
Voici les prérequis :

« ***apt install mailutils*** »

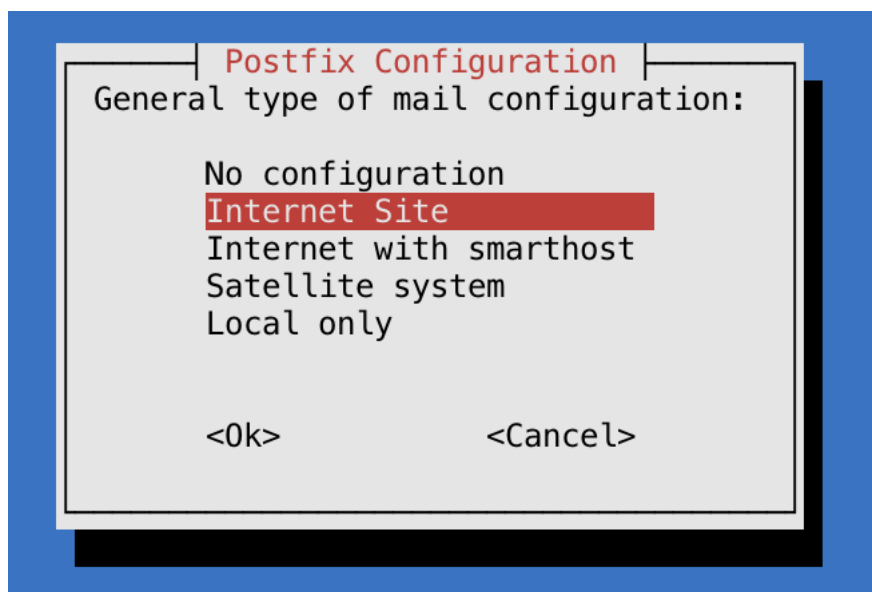
« ***apt install ssmtp*** »

Puis installer « **Postfix** », une fois l'installation de postfix, une interface apparaîtra pour permettre la configuration de postfix.

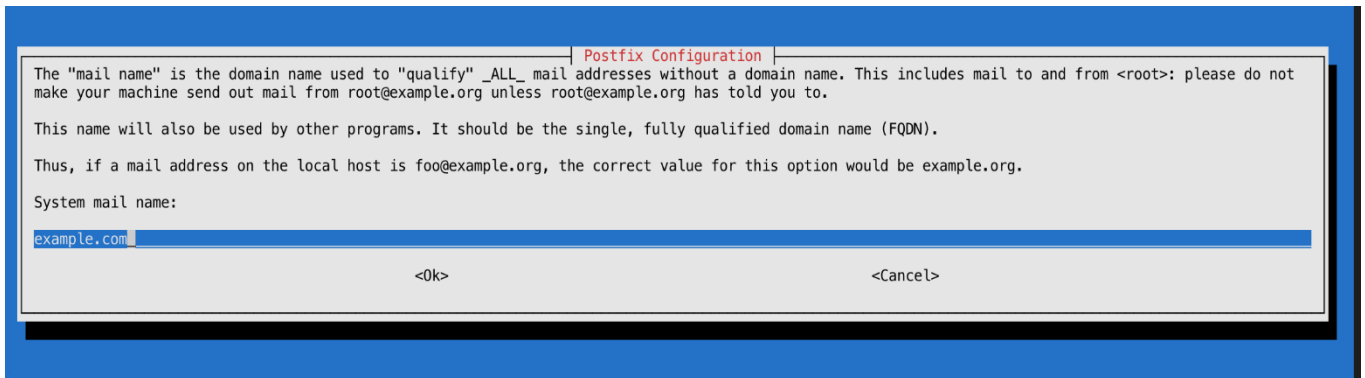
Ensuite faite « **Ok** »



Pour recevoir des mails on choisit « **Internet Site** »



Toujours dans la configuration de Postfix, nous devons d'abord créer un mail pour recevoir des notifications.



Une fois la configuration est terminée, nous allons nous diriger dans le fichier « **nano /etc/postfix/main.cf** » puis modifier et rajouter les lignes suivantes :

- « **relayhost = [smtp.gmail.com]:587** » (Cette ligne est à modifier)
- « **smtp_sasl_auth_enable = yes** »
- « **smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd** »
- « **smtp_sasl_security_options = noanonymous** »
- « **smtp_tls_CAfile = /etc/postfix/cacert.pem** »
- « **smtp_use_tls = yes** »

Vous devez aller dans le fichier « **nano /etc/postfix/sasl_passwd** » puis ajouter une ligne :

« **[smtp.gmail.com]:587 SiSr.nagios@gmail.com:sisr2016** ».

Après avoir ajouté la ligne dans le fichier « **sasl_passwd** » nous attribuons des droits pour pouvoir utiliser le fichier, on exécute la commande « **postmap** », elle permet de créer en quelque sorte de base de donnée, puis on redirige le certificat vers le fichier « **/etc/postfix/cacert.perm** ».

« **chmod 400 /etc/postfix/sasl_passwd** »

« **postmap /etc/postfix/sasl_passwd** »

« **cat /etc/ssl/certs/thawte_Primary_Root_CA.pem | tee -a /etc/postfix/cacert.pem** ».

Pour tester si la configuration a été fait avec succès nous envoyons un mail de test :

« **echo "Test mail from postfix" | mail -s "Test Postfix" SiSr.nagios@gmail.com** »

Nous avons reçu le mail test avec succès on peut dire que la configuration a été accomplis.

Enfin nous passons à l'installation et la configuration de Nagios.

Tout d'abord nous allons créer l'utilisateur pour Nagios :

« **useradd ppe6nagios** »

« **groupeadd ppe6nagcmd** »

« **usemod -a -G ppe6nagcmd ppe6nagios** »

« **usermod -a -G ppe6nagcmd www-data** »

Puis nous téléchargerons le fichier compressé de Nagios.

Voici les commandes suivantes :

- « **cd /tmp /** »
- « **wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.4.5.tar.gz** » pour « wget », il faut une petite configuration dans le fichier « **nano /etc/wgetrc** »
- « **tar -zxvf nagios-4.4.5.tar.gz** »
- « **cd /tmp/nagios-4.4.5 /** ».

Une fois le téléchargement terminé, on va compiler Nagios avec le code ci-dessous :

- « **make all** » (si vous n'avez les paquets « make » vous devez installer avec la commande « **apt -get install make** ».
- « **make install** »
- « **make install-init** »
- « **make install-config** »
- « **make install-commandnode** »
- « **make instal-webconf** ».

Dans le fichier « **nano contacts.cfg** » nous modifions une ligne précise pour recevoir des mails du serveur comme ceci :

```
define contact {  
    contact_name      nagiosadmin          ; Short name of user  
    use               generic-contact      ; Inherit default values from generic-contact  
    alias             Nagios Admin         ; Full name of user  
    email             SiSr.nagios@gmail.com ; <<***** CHANGE THIS TO YOUR EMAIL ADDRESS  
}
```

Ensuite on configure l'interface web de Nagios avec les commandes :

« **htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin** »

« **mot de passe : 12-Soleil&** »

Pour le service APACHE2, on active une méthode d'interaction entre un serveur web et des programmes qui est CGI (Common Gateway Interface) avec la commande :

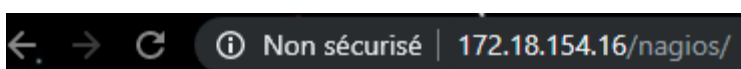
« **a2enmod cgi** ».

Puis relancer le service apache2 :

« **systemctl restart apache2** ».

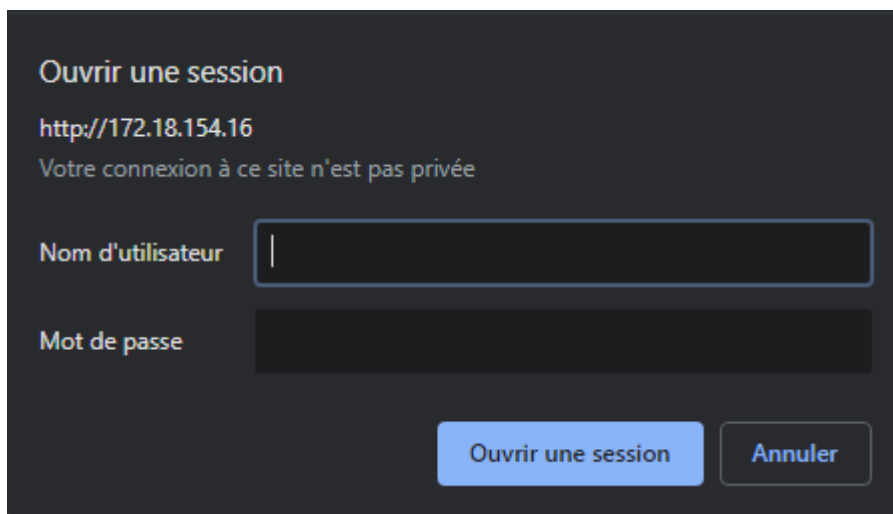
Maintenant passons à l'interface web de Nagios.

D'abord taper votre adresse IP de serveur Nagios :



← → ↻ ⓘ Non sécurisé | 172.18.154.16/nagios/

Après avoir tapé votre adresse IP dans la barre de recherche une pop-up vas apparaître comme ceci :

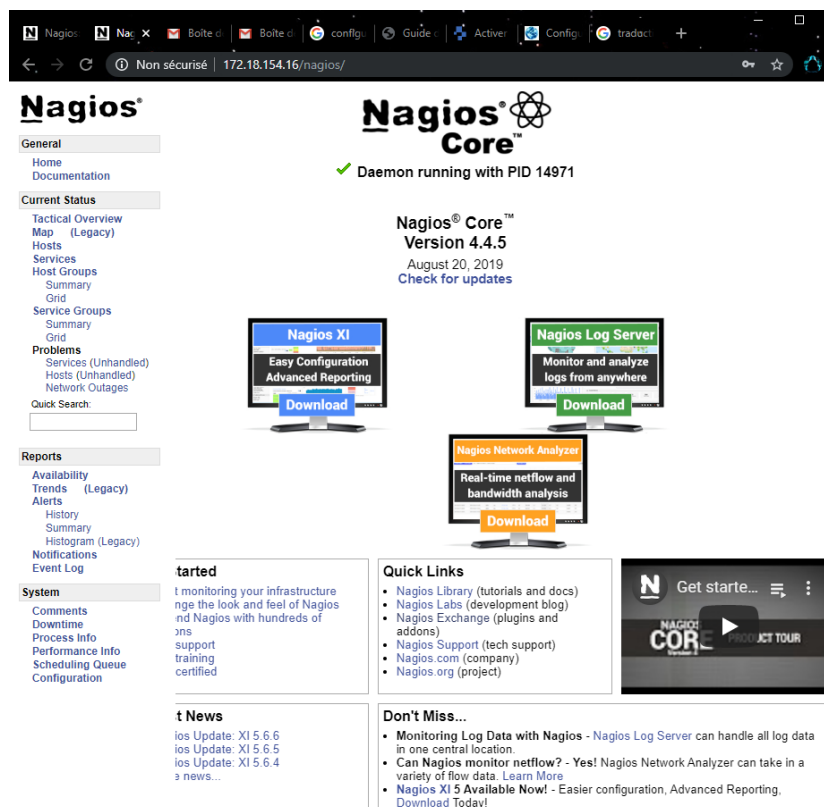
A dark-themed login pop-up window. At the top, it says "Ouvrir une session" in orange. Below that is the URL "http://172.18.154.16" and the text "Votre connexion à ce site n'est pas privée". There are two input fields: "Nom d'utilisateur" and "Mot de passe". At the bottom right, there are two buttons: "Ouvrir une session" (blue) and "Annuler" (grey).

Ensuite veuillez entrer votre « **Nom d'utilisateur** » et le « **Mot de passe** »

« **ID nagios : nagiosadmin** »

« **Mdp : 12-Soleil&** »

Voici le résultat l'interface web s'affichera comme ceci :

A screenshot of the Nagios Core 4.4.5 web interface in a browser. The browser address bar shows "172.18.154.16/nagios/". The interface has a dark theme. On the left is a sidebar with navigation links like "General", "Current Status", "Problems", "Reports", and "System". The main content area shows the "Nagios Core Version 4.4.5" status, a "Daemon running with PID 14971" message, and several product tiles for "Nagios XI", "Nagios Log Server", and "Nagios Network Analyzer". There are also sections for "Quick Links", "Latest News", and "Don't Miss...".

Une fois la vérification faite sur l'interface web de Nagios, nous n'oublions pas d'installer les plugins de Nagios.

On installe avec les commandes suivantes :

- « **cd /tmp** »

- « `wget https://nagios-plugins.org/download/nagios-plugins-2.2.1.tar.gz` »
- « `tar -zxvf /tmp/nagios-plugins-2.2.1.tar.gz` »
- « `cd /tmp/nagios-plugins-2.2.1/` ».

Comme pour l'installation de Nagios, nous allons compiler les plugins puis les installer.

- « `/configure --with-nagios-user=nagios --with-nagios-group=nagios` »
- « `make` »
- « `make install` ».

Pour savoir si nous n'avons pas d'erreurs de syntaxe dans les fichiers de configuration de Nagios.

On applique la commande suivante :

« `/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg` ».

```

172.18.154.16 - PuTTY
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2019-08-20
License: GPL

Website: https://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
  Checked 15 services.
  Checked 3 hosts.
  Checked 3 host groups.
  Checked 0 service groups.
  Checked 1 contacts.
  Checked 1 contact groups.
  Checked 24 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 3 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check

```

Nous procédons aux dernières configurations de Nagios, pour surveiller des machines grâce à plusieurs fichiers. Ainsi pour commencer dans le fichier « `/usr/local/nagios/objects/` » nous avons créé un fichier dont le nom est « **hostgroups** »

```

define hostgroup {
hostgroup_name PPE6grp2
alias bibi
members INTRALAB2,PPEWinServerSQL,localhost
}

```

Enfin, pour terminer nous allons mettre des services qui seront important à surveiller pour pouvoir anticiper les anomalies en cas de panne critique.

Le service que nous surveillerons est le service « **http** »

```
define service {
    use                generic-service
    host_name          INTRALAB2
    service_description HTTP
    check_command       check_http
    notifications_enabled 1
    check_interval      1
    #retry_interval     1
    notification_period 24x7
    notification_options w,u,c,r
}
```

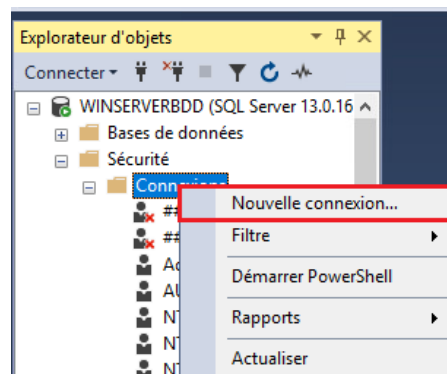
VI) Mise en place du serveur SQL SERVER + Windows server

Pour cette mise en place du Server de base de données (**BDD**), nous utilisons un système d'exploitation **Windows 10 de type Server**. Nous téléchargerons **SQL-Server 2016** et non pas la version 2012 car cela aurait pu être problématique en matière de compatibilité en particulier pour les développeurs utilisant une version supérieure à 2012.

Après avoir installé SQL-Server 2016, passons à la configuration de SQL-Server. Dans le cahier des charges il était demandé de gérer la base de données SQL-Server qui sera importante pour les développeurs.

Cette tâche a été priorisée afin que les développeurs puissent accéder à la Base De Données distante et effectuer leurs tests.

Nous allons créer un utilisateur pour que les développeurs puis avoir l'accès au Serveur de base de données (BDD)



Une fois que nous avons fait « **Nouvelle connexion** », cette page s'apparaîtra :

Nouvelle connexion

Sélectionner une page

- Général
- Rôles du serveur
- Mappage d'utilisateur
- Éléments sécurisables
- État

Connexion

Serveur : WINSERVERBDD

Connexion : WINSERVERBDD\Administrateur

[Afficher les propriétés de connexion](#)

Progression

Prêt

Nom d'accès : PPE6grp2 Rechercher...

☐ Authentification Windows

☒ Authentification SQL Server

Mot de passe :

Confirmer le mot de passe :

☐ Spécifier l'ancien mot de passe

Ancien mot de passe :

☒ Appliquer la stratégie de mot de passe

☐ Appliquer l'expiration du mot de passe

☐ L'utilisateur doit changer de mot de passe à la prochaine connexion

☐ Mappé au certificat

☐ Mappé à la clé asymétrique

☐ Mapper aux informations d'identification

Informations d'identification mappées

Informations ... Fournisseur

Ajouter

Supprimer

Base de données par défaut : master

Langue par défaut : <par défaut>

OK Annuler

Voici les données pour se connecter à la base de données :

Nom d'accès : « **PPE6grp2** »

Mot de passe : « **12-Soleil&** »

Avec une authentification SQL-Server qui va permettre au développeur de se connecter sur SQL pour créer leur base de données.

Nouvelle connexion

Sélectionner une page

- Général
- Rôles du serveur
- Mappage d'utilisateur
- Éléments sécurisables
- État

Paramètres

Autorisation de se connecter au moteur de base de données :

☒ Accorder

☐ Refuser

Connexion :

☒ Activé

☐ Désactivé

État

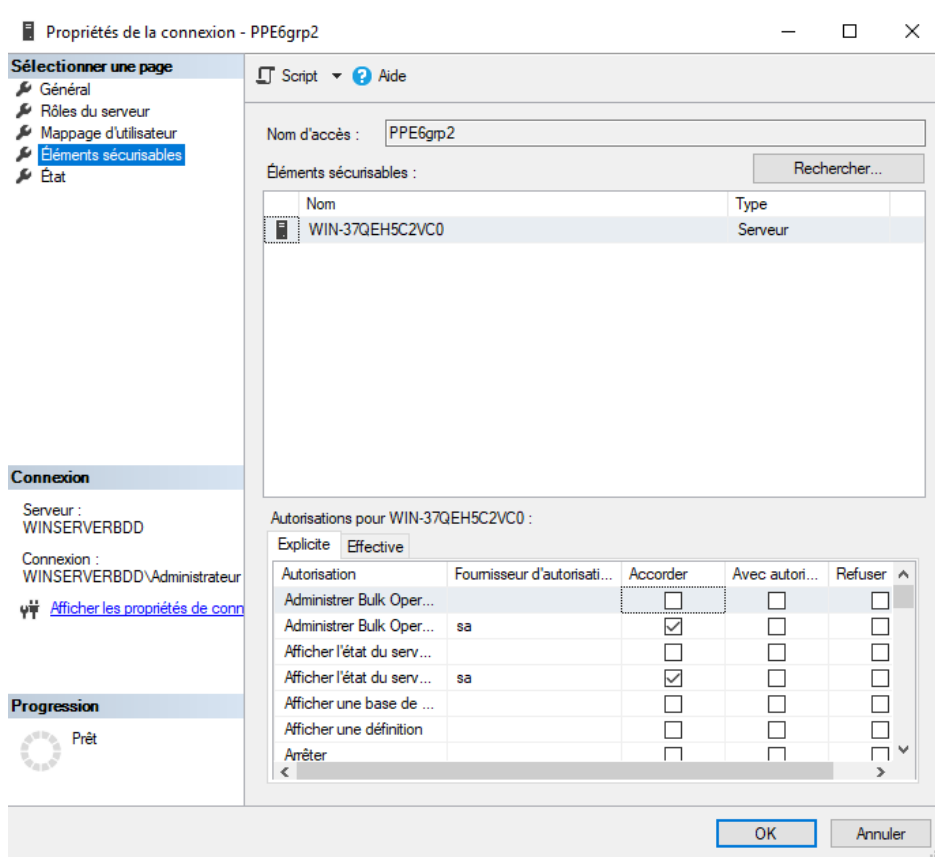
Authentification SQL Server :

Dans ses paramètres nous allons accorder les autorisations pour avoir l'accès au moteur de base de donnée et activer la connexion.

Puis faite « *Ok* »



Maintenant, après avoir créé l'utilisateur nous allons accorder des droits au développeur afin qu'ils puissent « *Créer, Modifier, Supprimer, et de se Connecter* ».



Enfin nous allons passer au test de la connexion de l'utilisateur :

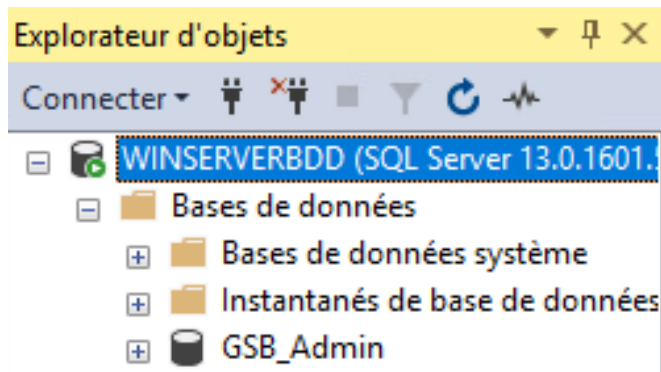
Avec le nom du serveur qui est « **WINSERVERBDD** » avec l'authentification SQL-Server

Il faut tout simplement rentrer :

ID de connexion : PPE6grp2

Mot de passe 12-Soleil&

Ensuite, voici le résultat :



VIII) Note évolutive

1) Application

a) Formulaire de renouvellement de mot de passe

Le formulaire de renouvellement de mot de passe pourrait subir certaines mises à jour, comme par exemple l'ajout d'une auto-complétion s'il y a vraiment trop d'utilisateur dans la base de données

Ensuite, pour rendre plus confortable cette modification de mot de passe, on pourrait ajouter un état aux utilisateurs, où lorsqu'ils auraient plus de mot de passe enverrait un message automatiquement à l'administrateur et où il pourra voir via son application les personnes qui demande une modification de mot de passe (surlignés en rouge) et lorsque cette modification est effectuée. Un mail automatique est envoyée à la personne concernée reçoit son nouveau mot de passe.

Enfin pour le formulaire de réinitialisation, on pourrait modifier l'option « par défaut » qui nous rentre un mot de passe prédéfini, à la place de cela on pourrait générer aléatoirement un mot de passe géré par l'application qui respecterait les critères de sécurité.

b) Formulaire de création d'utilisateur

Pour l'évolution du formulaire de création d'utilisateur, il s'agirait également d'une évolution ergonomique, où lorsqu'on choisirait d'attribuer une voiture à une personne, d'avoir un calendrier nous montrant les jours en rouges lorsque cette voiture est louée au jour concerné.

Ensuite on pourrait faire en sorte lors de la création de l'utilisateur, de permettre l'attribution de plusieurs véhicules sur le même formulaire pour l'utilisateur sur des dates différentes.

V) Conclusion

Pour conclure, en ce qui concerne la partie réseau, des difficultés ont été rencontrées lors de la mise en place de la stratégie de supervision pour certains serveurs.

En ce qui concerne la partie développement il y a eu globalement aucune difficulté sauf pour la mise en place des procédures stockées qui étaient assez longues. Mais il n'y a eu aucun retard à déplorer.

Globalement tout s'est bien passé pour notre groupe, aucun retard n'a été pris.