

GSB – DOCUMENTATION TECHNIQUE

Table des matières

I.	Présentation du contexte GSB.....	2
II.	Présentation du projet	2
➤	Expression des besoins.....	2
➤	Proposition de solution	3
III.	Analyse	3
1.	Modélisation des données	4
2.	Les traitements.....	4
1.	Connexion à l'application	5
2.	Ajout médicaments	8
3.	Ajout praticien.....	9
4.	Ajout visiteur	11
5.	Ajout d'un rapport.....	12
6.	Consulter un praticien	14
7.	Consulter un rapport.....	15
8.	Consulter un médicament	17
9.	Valider un rapport	18
10.	Supprimer un médicament.....	20
11.	Supprimer un praticien.....	21
12.	Supprimer un visiteur	22
13.	Supprimer un rapport de visite	24
IV.	Réalisation technique	25
A.	Architecture de l'application	25
B.	Biblio GSB	26
C.	Biblio GSBDAO	27
D.	Projet GSB.....	27
V.	Procédure d'installation de l'application	28

I. Présentation du contexte GSB

Galaxy Swiss Bourdin (GSB) est un laboratoire pharmaceutique issu de la fusion entre le géant américain Galaxy et le conglomérat européen Swiss Bourdin. L'entreprise compte 480 visiteurs médicaux en France métropolitain (Corse comprise), et 60 dans les départements et territoire d'outre-mer. Les territoires sont répartis en 6 secteurs géographiques (Paris-Centre, Sud, Nord, Ouest, Est, DTOM Caraïbes-Amériques, DTOM Asie-Afriques).

Les visiteurs médicaux (ou délégués) démarchent les médecins, pharmaciens, infirmières, et autres métiers de santé susceptibles de prescrire aux patients les produits du laboratoire.

L'objectif d'une visite est d'actualiser et rafraîchir la connaissance des professionnels de santé sur les produits de l'entreprise. Les visiteurs ne font pas de vente, mais leurs interventions ont un impact certain sur la prescription de la pharmacopée du laboratoire.

Pour donner une organisation commune aux délégués médicaux, l'entreprise a adopté l'organisation de la flotte de visiteurs existant chez Galaxy, selon un système hiérarchique par région et, à niveau supérieur, par secteur géographique (Sud, Nord, Paris-Centre, Antilles-Guyane, etc).

II. Présentation du projet

➤ Expression des besoins

Le laboratoire pharmaceutique GSB souhaite disposer d'une application permettant de centraliser les comptes-rendus de visite. Cette base d'information sera utilisée à des fins d'élaboration de la démarche de communication auprès des praticiens et donnera une vision individuelle et synthétique de l'activité de représentation.

L'application sera aussi utilisée pour gérer la traçabilité des échantillons fournis lors des visites. Pour permettre une aide au renseignement des rapports, l'application fournira une description des produits du laboratoire, les coordonnées précises des praticiens et des informations détaillées les concernant. Elle servira aussi à la mise en relation de la hiérarchie de la force commerciale, des visiteurs aux responsables de secteur en passant par les délégués régionaux.

Définition de l'objet

L'application doit permettre d'enregistrer des comptes rendus, de donner des informations sur des médicaments, praticiens.

Forme de l'objet

L'application sera développée en C# avec l'environnement de développement Visual Studio auquel sera associée une base de données SQL SERVER.

Accessibilité/Sécurité

L'environnement doit être accessible aux seuls acteurs de l'entreprise. Une authentification préalable sera nécessaire pour l'accès au contenu selon les missions attribuées.

➤ Proposition de solution

Une application sera développée en C# avec l'environnement de développement Visual Studio auquel sera associée une base de données SQL SERVER. Elle permettra d'enregistrer des comptes rendus, de donner l'ensemble des praticiens et des médicaments. Une authentification sera demandée lors de la connexion à l'application permettant à chaque acteur d'avoir accès au contenu de leurs missions attribuées.

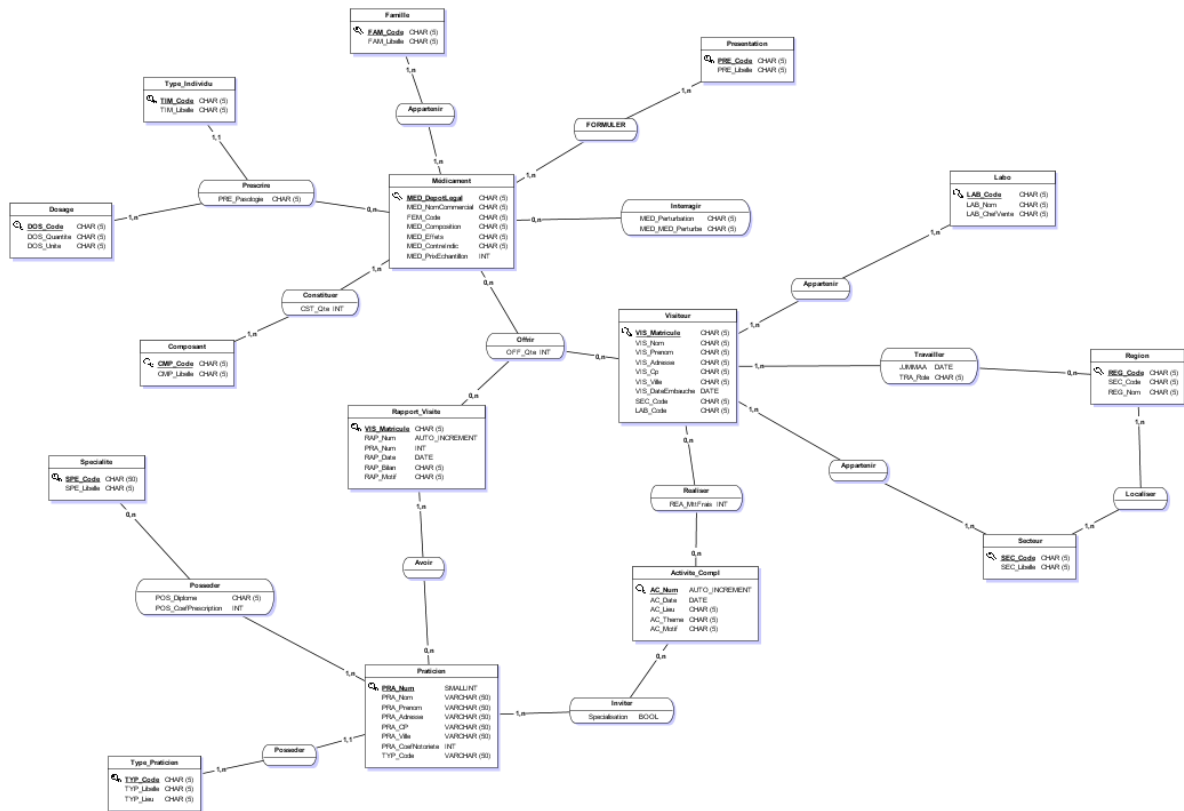
III. Analyse

L'architecture du code produit doit respecter les conventions d'usage : développement basé sur des bibliothèques de classes ou architecture en couches. Les IHM devront harmonieuses permettant aux utilisateurs aisément sur le logiciel. L'application doit également comporter des variables simples à reconnaître pour son faciliter son utilisation, son développement et la maintenance de l'application. Pour finir, l'application devra présenter plusieurs fonctionnalités :

- Enregistrement,
- Consultation,
- Mise à jour,
- Suppression.

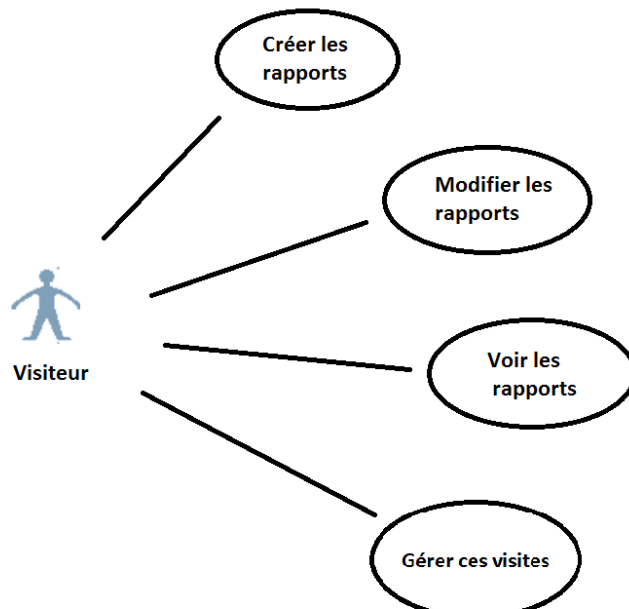
1. Modélisation des données

Schéma des données existant :



2. Les traitements

Le visiteur pourra :



La secrétaire pourra :



1. Connexion à l'application

PROJET : Application GSB	Description cas d'utilisation
---------------------------------	--------------------------------------

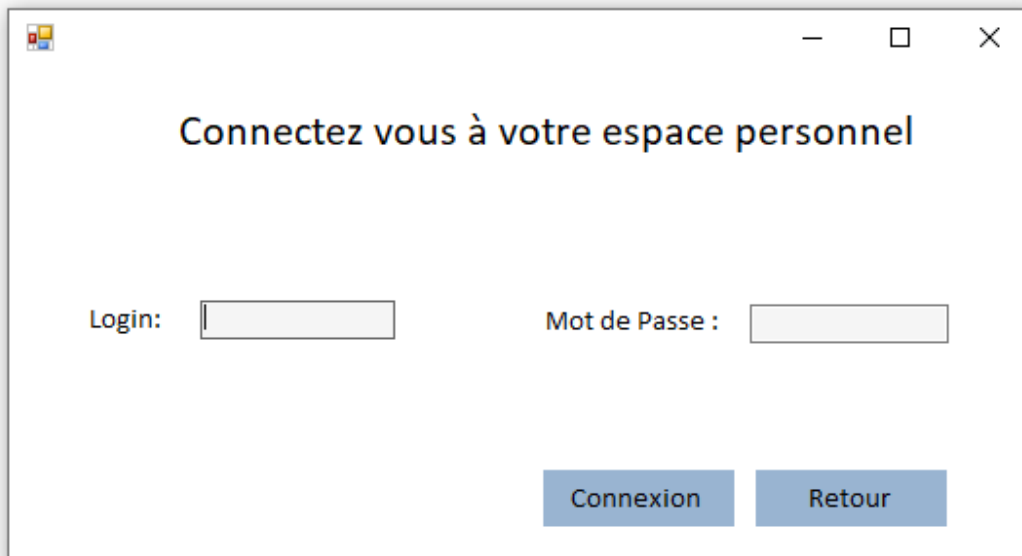
Nom cas d'utilisation : Se connecter
Acteur déclencheur : secrétaire ou visiteur
Pré conditions :
Post condition : La secrétaire ou le visiteur est reconnu(e)

Scénario nominal :

- 1- Le système affiche un formulaire de connexion
- 2- L'utilisateur saisit son login et son mot de passe et valide
- 3- Le système contrôle les informations de connexion et affiche le menu de l'application (menu secrétaire ou visiteur).

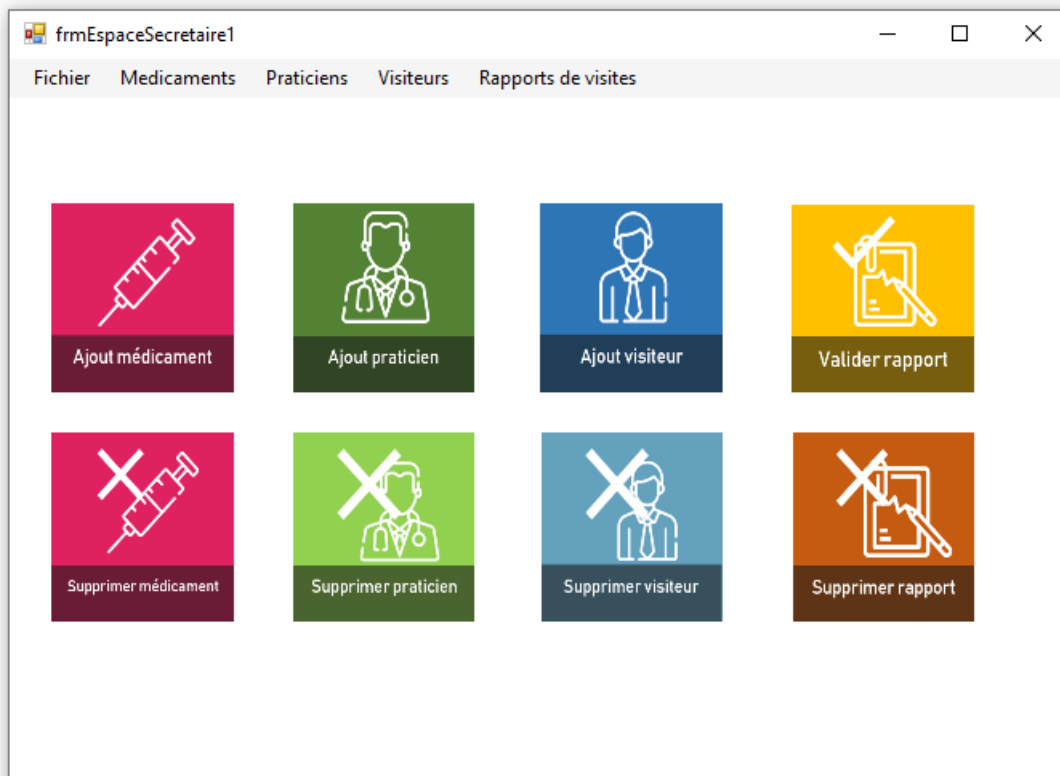
Exceptions :

- 3-a : le nom et/ou le mot de passe n'est pas valide
- 3-a.1 Le système en informe l'utilisateur ; retour à l'étape 1
- 4- L'utilisateur quitte l'application

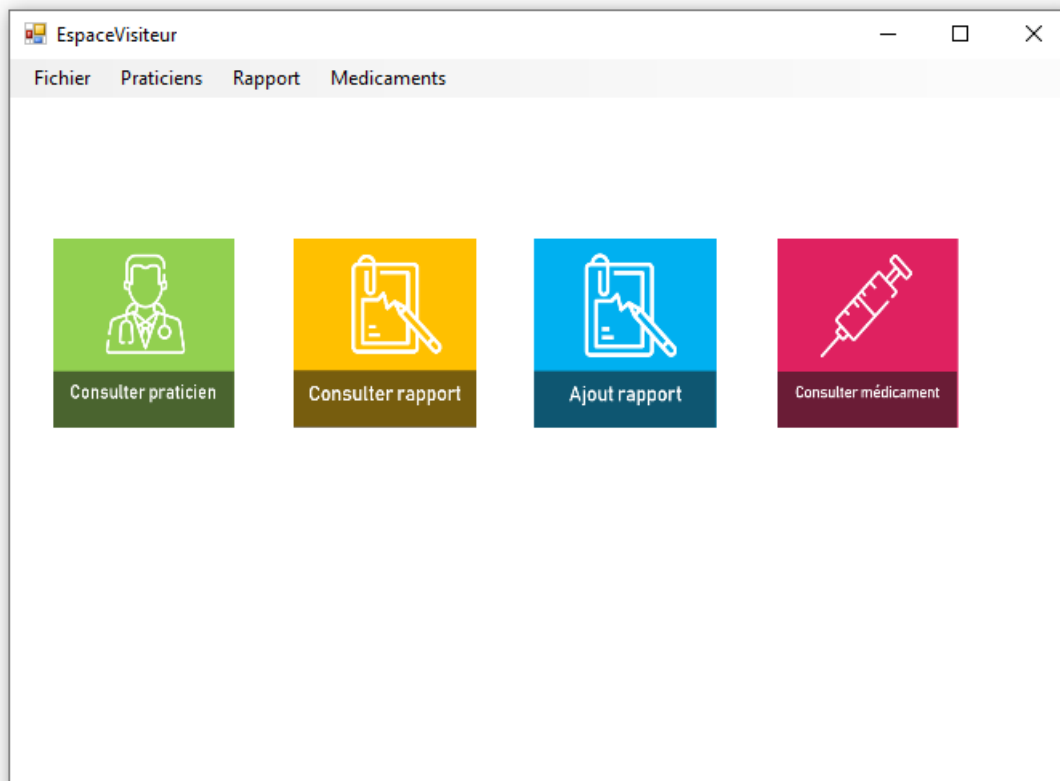
Contraintes :**Questions ouvertes :****Page de connexion :**

The screenshot shows a standard Windows-style window with a title bar containing a small icon on the left and three control buttons (minimize, maximize, close) on the right. The main content area has a title "Connectez vous à votre espace personnel" centered at the top. Below the title, there are two input fields: "Login:" followed by a text box, and "Mot de Passe :" followed by a text box. At the bottom of the form, there are two buttons: "Connexion" and "Retour".

Espace secrétaire :



Espace visiteur :



2. Ajout médicaments

PROJET : Application GSB	Description cas d'utilisation
Nom cas d'utilisation : Ajout médicament	
Acteur déclencheur : Secrétaire	
Pré conditions : Secrétaire authentifiée	
Post condition :	
<p>Scénario nominal :</p> <ol style="list-style-type: none"> 1- L'utilisateur ouvre le formulaire. 2- Le système remplit la liste des familles de médicaments. 3- L'utilisateur saisit le nom de dépôt légal, nom commercial, composition, effet(s) secondaire(s), contre-indications. 4- L'utilisateur choisit la famille. 5- L'utilisateur valide l'ajout du médicament dans la base de données. Continuer à l'étape 3. 6- L'utilisateur ferme le formulaire. 	
<p>Alternatifs :</p> <p>3.a. L'utilisateur annule l'ajout.</p>	
<p>Exceptions :</p> <p>5.a. Le système vérifie si le dépôt légal existe déjà si « oui » message d'erreur.</p> <p>5.b Le système vérifie si le nom commercial existe déjà si « oui » message d'erreur.</p>	
Contraintes :	
Questions ouvertes :	

Page d'ajout d'un médicament

frmAjoutMedicaments

Ajouter un Medicament

Depot Legal:

Nom Commercial:

Famille:

Composition:

Effet secondaire:

Contreindications:

Annuler Valider

3. Ajout praticien

PROJET : Application GSB	Description cas d'utilisation
---------------------------------	--------------------------------------

Nom cas d'utilisation : Ajout d'un praticien
Acteur déclencheur : secrétaire
Pré conditions : secrétaire authentifiée
Post condition :

Scénario nominal :

- 1- L'utilisateur ouvre le formulaire.
- 2- Le système remplit la liste des spécialités.
- 3- L'utilisateur saisit le nom, prénom, adresse, code postal, ville, coefficient de notoriété, coefficient confiance.
- 4- L'utilisateur valide l'ajout du praticien dans la base de données. Continuer à l'étape 3.
- 5- L'utilisateur ferme le formulaire

Alternatifs :

4.a. L'utilisateur annule l'ajout. Retour à l'étape 3.

Exceptions :

4.b. Le système affiche une erreur lors de l'ajout si le praticien existe déjà. Le système test le nom est le prénom. Retour à l'étape 3.

Contraintes :**Questions ouvertes :****Page d'ajout d'un praticien**

The screenshot shows a web browser window titled "frmAjoutPraticiens" with the page title "Ajouter un praticien". The form is displayed on a light blue background and contains the following fields:

- Nom:
- Prénom:
- Adresse:
- Code postal:
- Ville:
- Coeff. notoriété:
- Coeff. confiance:
- Spécialité:

At the bottom right of the form, there are two green buttons: "Annuler" and "Valider".

4. Ajout visiteur

PROJET : Application GSB	Description cas d'utilisation
Nom cas d'utilisation : Ajout visiteur	
Acteur déclencheur : <i>Secrétaire</i>	
Pré conditions : <i>Secrétaire authentifiée</i>	
Post condition :	
<p>Scénario nominal :</p> <ol style="list-style-type: none"> 1- L'utilisateur ouvre le formulaire. 2- Le système remplit la date d'embauche. 3- Le système remplit la liste des régions. 4- L'utilisateur saisi le matricule, prénom, nom, adresse, code postal, ville, login, mot de passe. 5- L'utilisateur valide l'ajout du médicament dans la base de données. Continuer à l'étape 2. 6- L'utilisateur ferme le formulaire. 	
<p>Alternatifs :</p> <p>5.a. L'utilisateur annule l'ajout.</p>	
<p>Exceptions :</p> <p>5.b. Le système vérifie si le matricule existe déjà.</p> <p>5.c Le système vérifie si le login existe déjà.</p>	
Contraintes :	
Questions ouvertes :	

Page d'ajout d'un visiteur

The screenshot shows a web browser window with the title 'Ajouter un visiteur'. The form inside has the following fields:

- Matricule:
- Nom:
- Prenom:
- Adresse:
- Code Postal:
- Ville:
- Region:
- Date Embauche:
- Login:
- Mot de passe:

At the bottom of the form are two buttons: 'Annuler' and 'Valider'.

5. Ajout d'un rapport

PROJET : Application GSB	Description cas d'utilisation
---------------------------------	--------------------------------------

Nom cas d'utilisation : Ajout Rapport
Acteur déclencheur : Visiteur
Pré conditions : Visiteur authentifié
Post condition :

Scénario nominal :

- 1- L'utilisateur ouvre le formulaire.
- 2- Le système remplit la liste des visiteurs.
- 3- Le système remplit la liste de la documentation (OUI ou NON).
- 4- Le système valorise le rapport définitif à NON.
- 5- Le système valorise la date de visite et la date de saisie.
- 6- Le système remplit la liste des motifs.
- 7- Le système remplit la liste des médicaments pour les médicaments présentés et les produits offerts (échantillon).
- 8- L'utilisateur saisi le numéro du rapport, motif autre si nécessaire, le bilan, la quantité en lien avec les échantillons.
- 9- L'utilisateur valide l'ajout du rapport de visite dans la base de données.
Continuer à l'étape 5.
- 10-L'utilisateur ferme le formulaire.

Alternatifs :

9.a. L'utilisateur annule l'ajout. Retour à l'étape 5.

Exceptions :

9.b. Le système affiche une erreur lors de l'ajout car le numéro de rapport existe déjà.
Retour à l'étape 7.

Contraintes :**Questions ouvertes :****Page ajouter rapport de visite**

Ajouter un rapport

Le rapport:

Choisissez un visiteur: Louis Villechalane

Numéro rapport: Rapport définitif: Non Documentation: Non

Praticien: Alain Notini Remplacant:

Date de visite: mardi 19 février Date de saisie: mardi 19 février

Motif: Actualisation annuelle Motif autre:

Bilan rapport:

Produits présentés:

Médicaments: Médicaments:

Produits offerts:

Médicaments:	Qte:	Médicaments:	Qte:
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

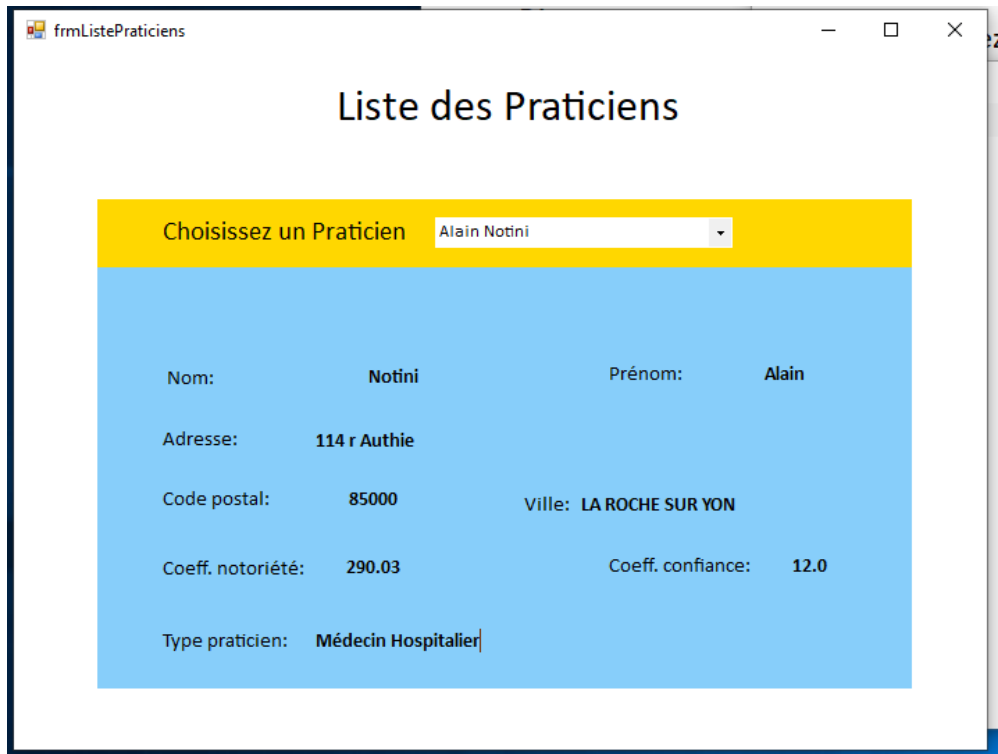
Annuler Valider

6. Consulter un praticien

PROJET : Application GSB	Description cas d'utilisation
---------------------------------	--------------------------------------

Nom cas d'utilisation : Consulter praticien
Acteur déclencheur : Visiteur
Pré conditions : Visiteur authentifié
Post condition :
Scénario nominal : 11- L'utilisateur ouvre le formulaire. 12- Le système remplit la liste des praticiens. 13- L'utilisateur ferme le formulaire.
Alternatifs :
Exceptions :
Contraintes :
Questions ouvertes :

Page consulter praticien



7. Consulter un rapport

PROJET : Application GSB	Description cas d'utilisation
---------------------------------	--------------------------------------

Nom cas d'utilisation : Consulter rapport
Acteur déclencheur : Visiteur
Pré conditions : Visiteur authentifié
Post condition :
Scénario nominal : <ul style="list-style-type: none"> 1- Le système affiche la liste des praticiens. 2- Le système remplit la liste des visiteurs. 3- Le système remplit la liste des rapports rattachés au visiteur.
Alternatifs :

Exceptions :

Contraintes :

Questions ouvertes :

Page consulter un rapport de visite

The screenshot shows a web application window titled "frmConsultRapport" with the main heading "Consulter un rapport". The interface includes several dropdown menus and text fields for selecting a visitor and report, and displaying visit details.

Chisissez un visiteur:

Chisissez un rapport:

Rapport définitif: **Non** Documentation: **Oui**

Praticien : **Elisabeth Flament** Remplacant :

Date de visite: **18/04/2002 00:00:00** Date de saisie: **25/04/2002 00:00:00**

Motif: **Actualisation annuelle** Motif autre:

Bilan rapport: **Médecin curieux, à recontacter en décembre pour réunion**

Produits présentés:

	MED_NOMCOMME
*	

Produits offerts:

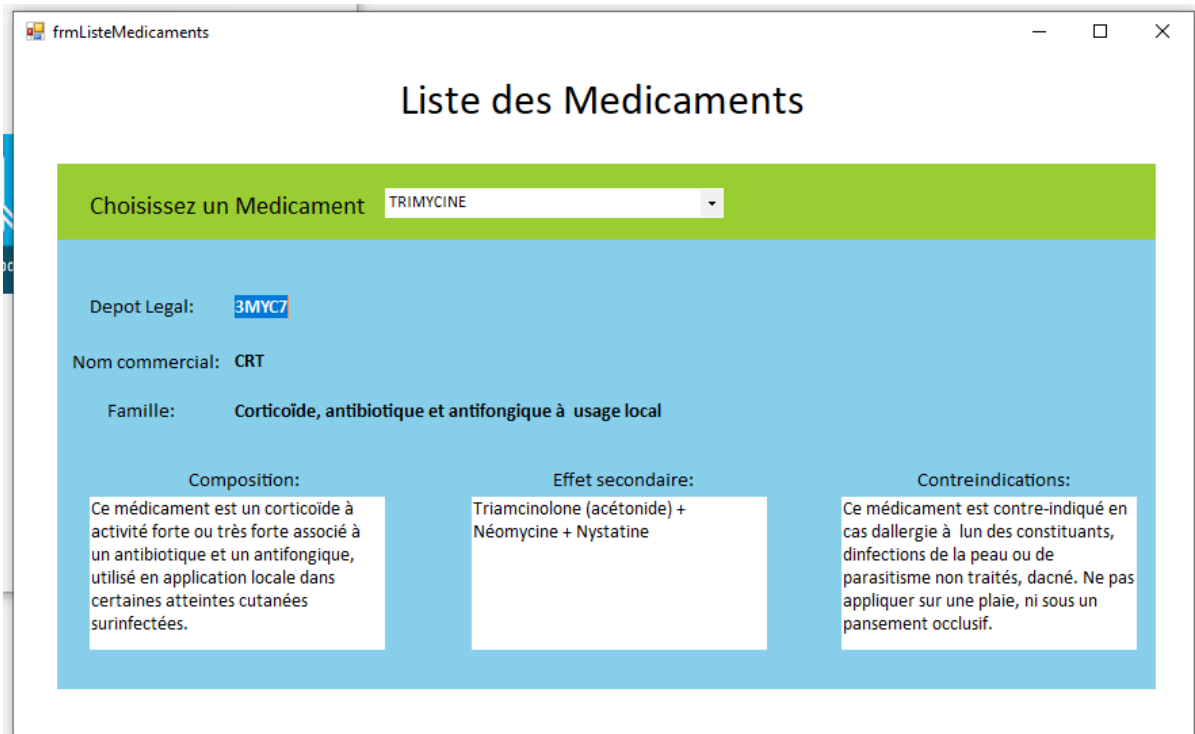
	MED_NOMCOMME	OFF_QTE
▶	ADIMOL	2
*		

8. Consulter un médicament

PROJET : Application GSB	Description cas d'utilisation
---------------------------------	--------------------------------------

Nom cas d'utilisation : Consulter médicament
Acteur déclencheur : Visiteur
Pré conditions : Visiteur authentifié
Post condition :
<ul style="list-style-type: none">1- L'utilisateur ouvre le formulaire.2- Le système remplit la liste des médicaments.3- L'utilisateur ferme le formulaire.
Alternatifs :
Exceptions :
Contraintes :
Questions ouvertes :

Page consulter un médicament



9. Valider un rapport

PROJET : Application GSB	Description cas d'utilisation
---------------------------------	--------------------------------------

Nom cas d'utilisation : Valider rapport
Acteur déclencheur : Secrétaire
Pré conditions : Secrétaire authentifié
Post condition :
<ul style="list-style-type: none"> 1- L'utilisateur ouvre le formulaire. 2- Le système remplit la liste des visiteurs 3- Le système remplit la liste des rapports associés au visiteur. 4- L'utilisateur modifie le rapport de visite. <ul style="list-style-type: none"> 4.a L'utilisateur supprimer les médicaments présentés. 4.b L'utilisateur supprime les produits offerts. 5- L'utilisateur ferme le formulaire.

Alternatifs :

4.a.a Ouverture d'une page de modification.

4.b.a Ouverture d'une page de modification.

Exceptions :

Contraintes :

Questions ouvertes :

Page valider un rapport

The screenshot shows a web application window titled "fmiValdeRapport" with the main heading "Valider un rapport". The interface is divided into two main sections: "Le rapport:" on the left and "Produits présentés:" on the right.

Le rapport:

- Choisissez un visiteur:** A dropdown menu with "Louis Villedelaine" selected.
- Numéro rapport:** A dropdown menu.
- Rapport définitif:** A dropdown menu.
- Documentaion:** A dropdown menu.
- Praticien :** A dropdown menu.
- Remplacant :** A dropdown menu.
- Date de visite:** A text input field.
- Date de saisie:** A text input field.
- Motif:** A dropdown menu.
- Motif autre:** A text input field.
- Bilan rapport:** A large text area.

Produits présentés:

- Médicaments:** A text area with a yellow circular refresh icon on the right.
- Produits offerts:** A section header.
- Médicaments:** A text area with a yellow circular refresh icon on the right.

At the bottom right, there are two yellow buttons: "Annuler" and "Valider".

10. Supprimer un médicament

PROJET : Application GSB	Description cas d'utilisation
---------------------------------	--------------------------------------

Nom cas d'utilisation : Supprimer un médicament
Acteur déclencheur : Secrétaire
Pré conditions : Secrétaire authentifiée
Post condition :
<ul style="list-style-type: none">1- Le système affiche les informations du stage.2- Le système affiche la liste des médicaments.3- Le système affiche la liste des modules déjà affectés au stage.4- L'utilisateur ferme le formulaire.
Alternatifs : 4.a. L'utilisateur annule la suppression. Retour à l'étape 2.
Exceptions :
Contraintes :
Questions ouvertes :

frmSuppMedicaments

Supprimer des médicaments

Médicaments:

TRIMYCINE

Annuler Valider

11. Supprimer un praticien

PROJET : Application GSB	Description cas d'utilisation
---------------------------------	--------------------------------------

Nom cas d'utilisation : Supprimer praticien
Acteur déclencheur : Secrétaire
Pré conditions : Secrétaire authentifiée
Post condition :
<ol style="list-style-type: none"> 1- L'utilisateur ouvre le formulaire. 2- Le système remplit la liste des praticiens. 3- L'utilisateur valide la suppression du praticien dans la base de données. Continuer à l'étape 2. 4- L'utilisateur ferme le formulaire

Alternatifs :

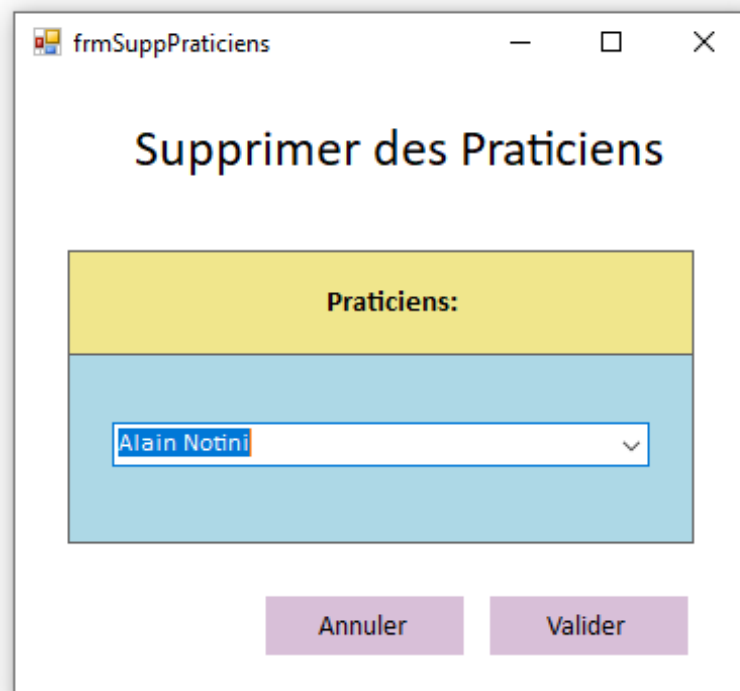
3.a. L'utilisateur annule la suppression. Retour à l'étape 2.

Exceptions :

Contraintes :

Questions ouvertes :

Page de mise à jour des sessions de stage



12. Supprimer un visiteur

PROJET : Application GSB	Description cas d'utilisation
---------------------------------	--------------------------------------

Nom cas d'utilisation : Supprimer visiteur

Acteur déclencheur : Administratif
Pré conditions : Administratif authentifié
Post condition :
<ol style="list-style-type: none">1- L'utilisateur ouvre le formulaire.2- Le système remplit la liste des visiteurs.3- L'utilisateur choisit le visiteur.4- L'utilisateur valide la suppression du visiteur dans la base de données. Continuer à l'étape 3.5- L'utilisateur ferme le formulaire.
Alternatifs : 4.a L'utilisateur annule la suppression. Retour à l'étape 3.
Exceptions :
Contraintes :
Questions ouvertes :

Page supprimer un visiteur



13. Supprimer un rapport de visite

PROJET : Application GSB	Description cas d'utilisation
---------------------------------	--------------------------------------

Nom cas d'utilisation : Supprimer un rapport
Acteur déclencheur : Secrétaire
Pré conditions : Secrétaire authentifiée
Post condition :
<ul style="list-style-type: none">1- L'utilisateur ouvre le formulaire.2- Le système remplit la liste de visiteur.3- Le système remplit la liste des rapports de visite en lien avec le visiteur.4- L'utilisateur valide la suppression du rapport de visite. Continuer à l'étape.5- L'utilisateur ferme le formulaire.
Alternatifs : 4.a. L'utilisateur annule l'ajout. Retour à l'étape 2.
Exceptions :
Contraintes :
Questions ouvertes :

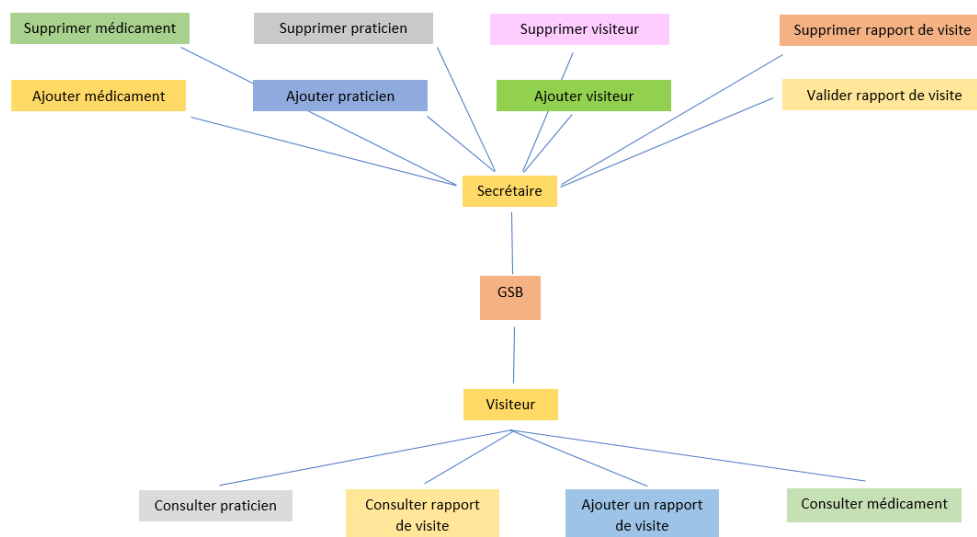
Page suppression d'un rapport de visite



IV. Réalisation technique

A. Architecture de l'application

Arborescence de l'application :



B. Biblio GSB

Principe de la classe Biblio GSB

Biblio ICGO regroupent l'ensemble des données qui seront utilisées dans le programme principal. Les valeurs vont alors transiter entre les différentes biblio et finir par être utilisés dans le projet ICGO afin que l'utilisateur puisse obtenir des résultats ou effectuer une action qui sera interprétée par le programme.

Propriétés Privés : Nous déclarons les variables qui vont être utilisées dans le programme.

Exemple :

```
public partial class visiteur ➡ Nom de la classe
{
    private string matricule, nom, prenom, adresse, cp, ville, dateembauche, login, mdp;
}
```

↙
 Declaration des variables visibles que depuis la classe "visiteur" grâce à l'attribut "private".

*classe : c'est une construction qui vous permet de créer vos propres types personnalisés en regroupant des variables d'autres types, méthodes et événements. Une classe s'apparente à un plan.

Constructeurs : Il recopie les attributs depuis l'objet passé en argument sur l'objet à créer.

```
public partial class visiteur
{
    private string matricule, nom, prenom, adresse, cp, ville, dateembauche, login, mdp;
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
    public visiteur()
    {
        this.realiser = new HashSet<realiser>();
        this.travailler = new HashSet<travailler>();
    }
    Le constructeur porte le même nom que la classe
    Declaration des variables
    public visiteur string UnMatricule, string UnNom, string UnPrenom, string UneAdresse, string UnCp, string UneVille, string UneDateEmbauche, string UnLogin,
    {
        matricule = UnMatricule;
        nom = UnNom;
        prenom = UnPrenom;
        adresse = UneAdresse;
        cp = UnCp;
        ville = UneVille;
        dateembauche = UneDateEmbauche;
        login = UnLogin;
        mdp = UnMdp;
    }
    Les propriétés privées prennent les valeurs du constructeur
```

C. Biblio GSBDAO

Cette biblio fait le lien avec la base de données notamment la connexion avec la base et les requêtes qui vont être utilisées dans l'exécution du programme principal.

Fonction de connexion à la base de données

Nom de la procédure

Informations relative à la base de données: son emplacement, son nom, sa sécurité si nécessaire

La variable 'connexion' reçoit les informations de la base de données puis elle ouvre la connexion

```
public static void ouvrirConnexion()
{
    connexion= new SqlConnection (@"Data Source=localhost;Initial Catalog=GSB_Compte_Rendu_Corrige;Integrated Security=True");
    Connexion.open()
}
```

D. Projet GSB

Principe de programmation d'un IHM

```
public frmAjoutVisiteurs()
{
    InitializeComponent();
}

private void btnValider_Click(object sender, EventArgs e)
{
    string Matricule, Nom, Prenom, Adresse, CodePosal, Ville, Login, Mdp, DateEmbauche;

    Matricule = txtMatricule.Text;
    Nom = txtNom.Text;
    Prenom = txtPrenom.Text;
    Adresse = txtAdresse.Text;
    CodePosal = txtCP.Text;
    Ville = txtVille.Text;
    Login = txtLogin.Text;
    Mdp = txtMdp.Text;
    manager.extraireDateTime(txtDateEmbauche1.Text, out DateEmbauche);

    visiteur UnVisiteur;
    travailler UnTravailler;
    region UneRegion;

    if (Matricule == "")
    {
        MessageBox.Show("Merci de renseigner le matricule du visiteur", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        if (Nom == "")
        {
            MessageBox.Show("Merci de renseigner le nom", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

Il s'agit d'une procédure, elle ne renvoie pas de valeur

Declaration des variables à utiliser et de leur types

Ce test avec l'attribut 'if' permet de verifier que toute les propriété nécessaire au formulaire d'ajout ont bien été renseigné par l'utilisateur lors de la validation Dans le cas d'un formulaire incomplet, un message d'alerte s'affiche à l'écran.

Charger les visiteurs depuis Manager

```
public class manager
{
    #region Visiteur
    public static void chargerlesVisiteurs(ComboBox comboVis)
    {
        List<visiteur> lesVisiteurs = new List<visiteur>();

        lesVisiteurs = visiteur.getLesVisiteurs();

        foreach (visiteur UnVisiteur in lesVisiteurs)
        {
            comboVis.Items.Add(UnVisiteur.Unprenom + " " + UnVisiteur.Unnom);
        }
    }
}
```

Nous passons en paramètres d'entrée, une liste de visiteurs.

Déclaration de la liste des visiteurs

Déclaration d'une variable 'unVisiteur, puis création d'une boucle pour générer la liste des visiteurs avec leur noms et prenom

V. Procédure d'installation de l'application

Le travail a été réalisé avec le logiciel Microsoft Visual Studio. Ce logiciel nous a permis d'utiliser le logiciel Team Foundation Server afin de travailler en réel sur le projet en équipe. Chaque participant au projet était affecté à une tâche précise pour éviter tous risques de modification simultanée sur une même partie du projet. Nous avons utilisé le serveur SRV-TFS afin d'y stocker notre base de données.