



# Présentation de l'application de Gestion

*SALIM ASSOUL  
SOFIANE BENDJELIDA*

**I/ STRUCTURE BD (MCD, MLD)**

**II/SCHÉMA DE CLASSE**

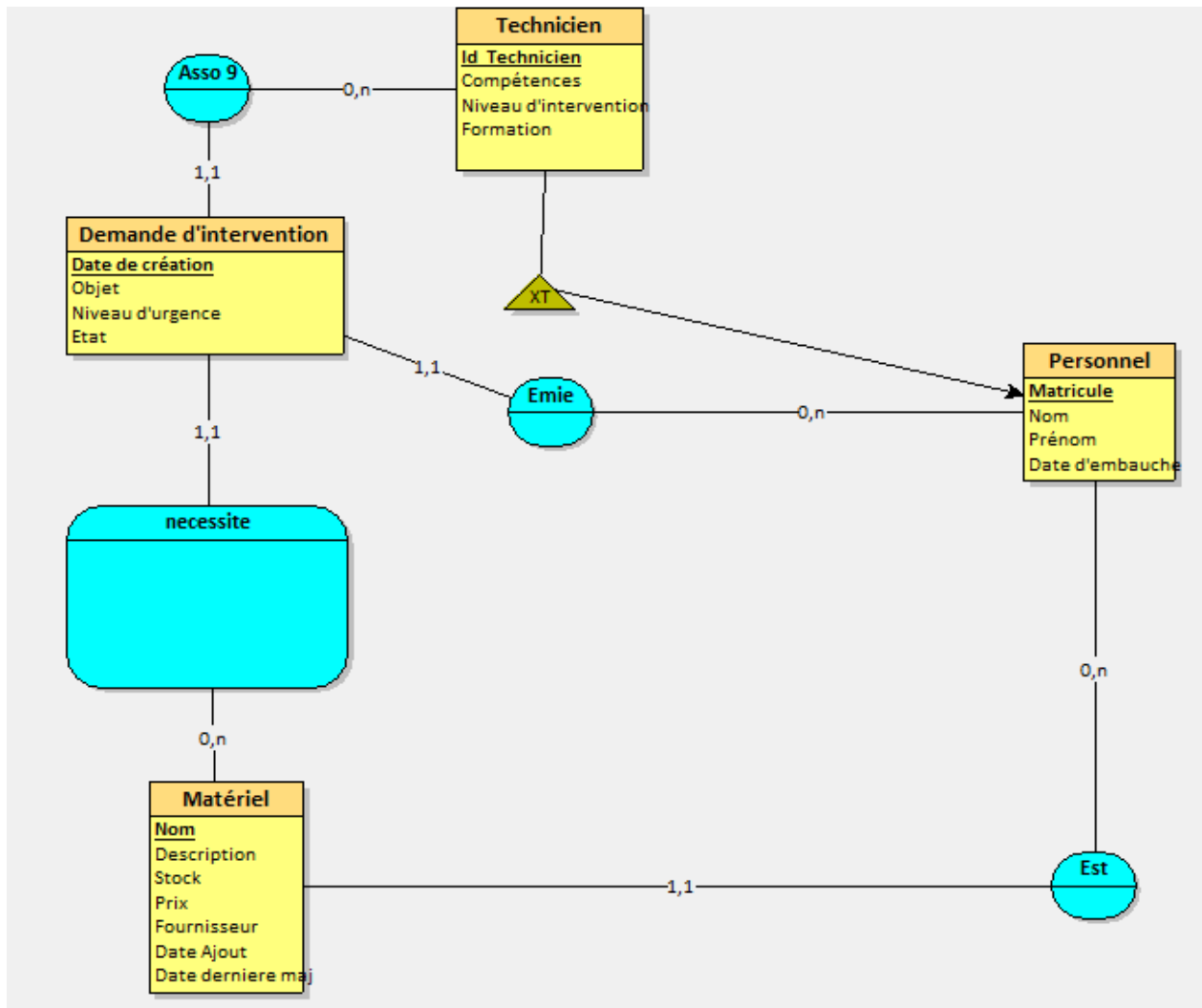
**III/SCREEN INTERFACE**

**IV/DESCRIPTION DE CHAQUE CLIC**

**V/MANUEL UTILISATEUR**

**VI/SCRIPT SAUVEGARDE BD + TUTO GENERATION SET-UP**

# I/ STRUCTURE BD (MCD, MLD)



MLD

```

Personnel = (Matricule VARCHAR(50), Nom VARCHAR(50), Prénom VARCHAR(50), Date_d_embauche VARCHAR(50));
Technicien = (Id_Technicien COUNTER, Compétences VARCHAR(50), Niveau_d_intervention VARCHAR(50), Formation VARCHAR(50), #Matricule);
Matériel = (Nom VARCHAR(50), Description VARCHAR(50), Stock VARCHAR(50), Prix VARCHAR(50), Fournisseur VARCHAR(50), Date_Ajout VARCHAR(50), Date_derniere_maj VARCHAR(50), #Matricule);
Demande_d_intervention = (Date_de_création VARCHAR(50), Objet VARCHAR(50), Niveau_d_urgence VARCHAR(50), Etat VARCHAR(50), #Id_Technicien, #Nom, #Matricule);
  
```

## II/SCHÉMA DE CLASSE

```
CREATE TABLE Personnel(  
    Matricule VARCHAR(50),  
    Nom VARCHAR(50),  
    Prénom VARCHAR(50),  
    Date_d_embauche VARCHAR(50),  
    PRIMARY KEY(Matricule)  
);  
  
CREATE TABLE Technicien(  
    Id_Technicien COUNTER,  
    Compétences VARCHAR(50),  
    Niveau_d_intervention VARCHAR(50),  
    Formation VARCHAR(50),  
    Matricule VARCHAR(50) NOT NULL,  
    PRIMARY KEY(Id_Technicien),  
    UNIQUE(Matricule),  
    FOREIGN KEY(Matricule) REFERENCES Personnel(Matricule)  
);  
  
CREATE TABLE Matériel(  
    Nom VARCHAR(50),  
    Description VARCHAR(50),  
    Stock VARCHAR(50),  
    Prix VARCHAR(50),  
    Fournisseur VARCHAR(50),  
    Date_Ajout VARCHAR(50),  
    Date_derniere_maj VARCHAR(50),  
    Matricule VARCHAR(50) NOT NULL,  
    PRIMARY KEY(Nom),  
    FOREIGN KEY(Matricule) REFERENCES Personnel(Matricule)  
);  
  
CREATE TABLE Demande_d_intervention(  
    Date_de_création VARCHAR(50),  
    Objet VARCHAR(50),  
    Niveau_d_urgence VARCHAR(50),  
    Etat VARCHAR(50),  
    Id_Technicien INT NOT NULL,  
    Nom VARCHAR(50) NOT NULL,  
    Matricule VARCHAR(50) NOT NULL,  
    PRIMARY KEY(Date_de_création),  
    FOREIGN KEY(Id_Technicien) REFERENCES Technicien(Id_Technicien),  
    FOREIGN KEY(Nom) REFERENCES Matériel(Nom),  
    FOREIGN KEY(Matricule) REFERENCES Personnel(Matricule)
```

## III/SCREEN INTERFACE

Accueil Incident Responsable Technicien Donne



**Sélectionne votre statut :**

☐ Utilisateur ☐ Technicien ☐ Responsable

**Login :**

Username:

Password:

**Connexion**

Accueil Incident Responsable Technicien Donne



**Declare l'incident :**

Matricule :  Niveaux d'intervention :

Nom Matériel :

Probleme :

**Enregistrer**

**Suivre des Incidents:**

**Afficher**

Accueil Incident Responsable Technicien Donne

Accueil Incident Responsable Technicien Donne



### Gestion des Comptes :

Nom :   
Prenom :  Choisir

Ajouter

--	--

Afficher

Supprimer

### Modifier un Compte :

Nom :   
Prenom :   
Role : Choisir

Modifier

--



### Gestion du matériel :

Matériels :  Stock :

Fournisseur :  Prix :

Ajouter

--

Afficher

Delete

### Gestion des Incidents :

--

X

Consulter

V

Enregistrer


Accueil

Incident

Responsable

Technicien

Donne



**Visualisation des incidents utilisateurs**

■ Declarer

■ Resolue

■ En Charge

Actualiser

---

**Visualisation des Statistique Technicien**

■ Declarer

■ Resolue

■ En Charge

Actualiser

---

## IV/DESCRIPTION DE CHAQUE CLIC

- **Bouton "Connexion"**: Permet la connexion en fonction du statut sélectionné.
- **Bouton "Enregistrer Incident"**: Enregistre un nouvel incident dans la base de données.
- **Bouton "Ajouter Compte"**: Ajoute un nouveau compte (technicien ou personnel) à la base de données.
- **Bouton "Afficher Compte"**: Affiche les informations des comptes créés.
- **Bouton "Supprimer Compte"**: Supprime un compte sélectionné de la liste des intervenants.
- **Bouton "Modifier Compte"**: Modifie les informations d'un compte sélectionné.
- **Bouton "Ajouter Matériel"**: Ajoute une entrée de matériel dans la base de données.
- **Bouton "Afficher Matériel"**: Affiche les informations du matériel ajouté.
- **Bouton "Supprimer Matériel"**: Supprime une entrée de matériel sélectionnée.
- **Bouton "Consulter Incident"**: Affiche la liste des incidents enregistrés.
- **Bouton "État Incident"**: Modifie l'état d'un incident (Résolu ou Pris en charge).
- **Bouton "Pris en Charge"**: Marque un incident comme "Pris en charge".
- **Bouton "Actualiser Statut Incident"**: Peut être utilisé pour actualiser les informations sur l'état des incidents, bien que sa fonction précise ne soit pas définie dans le code fourni.

---

# MANUEL UTILISATEUR DE L'APPLICATION GSB

Bienvenue dans le manuel utilisateur de l'Application GSB. Ce guide vous aidera à utiliser l'application de manière efficace.

## **Installation:**

Rendez-vous sur le site web officiel de l'Application GSB.

Téléchargez le fichier d'installation correspondant à votre système d'exploitation (Windows, macOS, Linux).

Exécutez le fichier d'installation et suivez les instructions à l'écran pour installer l'application.

## **Connexion:**

Lancez l'application après l'installation.

Saisissez votre nom d'utilisateur et votre mot de passe.

Cliquez sur le bouton "Connexion" pour accéder à votre compte.

## **Fonctionnalités Principales:**

### **1. Incident**

**Signalez les problèmes :** Utilisez cette fonction pour signaler tout incident ou problème technique rencontré lors de l'utilisation de l'application.

**Description détaillée :** Fournissez une brève description de l'incident, en incluant tous les détails pertinents.

**Suivi des incidents :** Une fois signalé, suivez l'évolution de l'incident dans la section "Suivi des Incidents" pour des mises à jour.

**Assistance :** Notre équipe est là pour résoudre les incidents signalés, et nous vous tiendrons informés de leur résolution.



---

## **2. Responsable**

**Ajouter un Compte** : Pour ajouter un compte utilisateur, cliquez sur le bouton "Ajouter Compte". Remplissez les champs avec le nom, le prénom, et le rôle de l'utilisateur, puis appuyez sur "Valider". L'utilisateur sera ajouté à la liste des intervenants.

**Afficher un Compte** : Pour afficher un compte utilisateur, entrez le nom, le prénom et le rôle, puis cliquez sur "Afficher Compte". Les informations du compte seront affichées dans la liste des comptes.

**Supprimer un Compte** : Pour supprimer un compte utilisateur, sélectionnez l'utilisateur dans la liste des intervenants et cliquez sur "Supprimer Compte". Assurez-vous de sélectionner l'utilisateur correctement avant de supprimer.

**Modifier un Compte** : Pour modifier un compte utilisateur, sélectionnez l'utilisateur dans la liste des intervenants. Modifiez les informations nécessaires (nom, prénom, rôle) dans les champs correspondants, puis cliquez sur Modifier Compte. Les informations du compte seront mises à jour.

## **3. Technicien**

**Ajouter du Matériel** : Utilisez le bouton "Ajouter Matériel" pour enregistrer un nouvel équipement. Entrez le nom, le stock, le prix, le fournisseur et le matricule, puis appuyez sur "Valider". Le matériel sera ajouté à la liste.

**Afficher du Matériel** : Pour consulter les informations sur un équipement, entrez les détails dans les champs correspondants et cliquez sur "Afficher Matériel". Les informations seront ajoutées à la liste.

**Supprimer du Matériel** : Pour retirer un équipement, sélectionnez-le dans la liste et cliquez sur "Supprimer Matériel". Assurez-vous de sélectionner l'équipement à supprimer.

**Consulter les Incidents** : En cliquant sur "Consulter Incidents", vous affichez les incidents associés à l'équipement sélectionné.

**Mettre à Jour l'État de l'Incident** : Utilisez les boutons "Résolu" et "Pris en charge" pour mettre à jour l'état de l'incident associé à l'équipement. Les incidents peuvent être marqués comme "Résolus" ou "Pris en charge".

---

## **Support**

Si vous rencontrez des problèmes ou avez des questions, veuillez contacter notre équipe de support technique à l'adresse [supportGSB@outlook.com](mailto:supportGSB@outlook.com).

# VI/SCRIPT SAUVEGARDE BD + TUTO GENERATION SET-UP + GAANT RÉEL

```
static string connStr = "Server=127.0.0.1; Database=gsb; Uid=root;Password=;SslMode=none";
static MySqlConnection conn = new MySqlConnection(connStr);

1 référence
public static void insertTechnicien(Technicien unTechnicien)
{
    conn.Open();
    MySqlCommand cmd = conn.CreateCommand();
    cmd.CommandText = "INSERT INTO tabTechnicien VALUES (@nom, @prenom, @role, @Niveauintervention, @Formation)";
    cmd.Parameters.AddWithValue("@nom", unTechnicien.Nom);
    cmd.Parameters.AddWithValue("@Competences", unTechnicien.Prenom);
    cmd.Parameters.AddWithValue("@role", unTechnicien.Role);
    cmd.Parameters.AddWithValue("@Niveauintervention", unTechnicien.Niveauintervention);
    cmd.Parameters.AddWithValue("@Formation", unTechnicien.Formation);
    cmd.Parameters.AddWithValue("@Matricule", unTechnicien.Matricule);

    cmd.ExecuteNonQuery();
    conn.Close();
}

1 référence
public static void insertPersonnel(Personnel unPersonnel)
{
    conn.Open();
    MySqlCommand cmd = conn.CreateCommand();
    cmd.CommandText = "INSERT INTO tabPersonnel VALUES (@Matricule, @nom, @prenom, @role, @Dateembauche)";
    cmd.Parameters.AddWithValue("@Matricule", unPersonnel.Matricule);
    cmd.Parameters.AddWithValue("@nom", unPersonnel.Nom);
    cmd.Parameters.AddWithValue("@prenom", unPersonnel.Prenom);
    cmd.Parameters.AddWithValue("@role", unPersonnel.Role);
    cmd.Parameters.AddWithValue("@Date_d_embauche", unPersonnel.Dateembauche);

    cmd.ExecuteNonQuery();
    conn.Close();
}

1 référence
public static void insertMateriel(Materiel unMateriel)
{
    conn.Open();
    MySqlCommand cmd = conn.CreateCommand();
    cmd.CommandText = "INSERT INTO tabMateriel VALUES (@nom, @description, @stock, @prix, @fournisseur)";
    cmd.Parameters.AddWithValue("@Nom", unMateriel.Nom);
    cmd.Parameters.AddWithValue("@Description", unMateriel.Description);
    cmd.Parameters.AddWithValue("@Stock", unMateriel.Stock);
    cmd.Parameters.AddWithValue("@Prix", unMateriel.Prix);
    cmd.Parameters.AddWithValue("@Fournisseur", unMateriel.Fournisseur);
    cmd.Parameters.AddWithValue("@Matricule", unMateriel.Matricule);

    cmd.ExecuteNonQuery();
    conn.Close();
}
```

```

1 reference
public static void insertIntervention(Intervention uneIntervention)
{
    conn.Open();
    MySqlCommand cmd = conn.CreateCommand();
    cmd.CommandText = "INSERT INTO Intervention VALUES (@datecreation, @objet, @niveauurgence, @etat, @tetechnicien, @nom, @matricule)";
    cmd.Parameters.AddWithValue("@datecreation", uneIntervention.Datecreation);
    cmd.Parameters.AddWithValue("@objet", uneIntervention.Object);
    cmd.Parameters.AddWithValue("@niveauurgence", uneIntervention.Niveauurgence);
    cmd.Parameters.AddWithValue("@etat", uneIntervention.Etat);
    cmd.Parameters.AddWithValue("@tetechnicien", uneIntervention.Technicien);
    cmd.Parameters.AddWithValue("@nom", uneIntervention.Nom);
    cmd.Parameters.AddWithValue("@matricule", uneIntervention.Matricule);

    cmd.ExecuteNonQuery();
    conn.Close();
}

2 reference
public static List<Material> getlesMaterials()
{
    List<Material> lesMaterials = new List<Material>();
    conn.Open();
    MySqlCommand cmd = conn.CreateCommand();
    cmd.CommandText = "SELECT * FROM material";
    MySqlDataReader reader = cmd.ExecuteReader();
    while (reader.Read())
    {
        lesMaterials.Add(new Material(Convert.ToString(reader["Nom"]), Convert.ToString(reader["Description"]), Convert.ToString(reader["Stock"]), Convert.ToInt32(reader["Prix"]), Convert.ToString(reader["Fournisseur"]), Convert.ToString(reader["Dateajout"]), Convert.ToInt32(reader["Matricule"]));
    }

    return lesMaterials;
}

3 reference
static List<Technician> getlesTechnician()
{
    List<Technician> lesTechnician = new List<Technician>();
    conn.Open();
    MySqlCommand cmd = conn.CreateCommand();
    cmd.CommandText = "SELECT * FROM technician";
    MySqlDataReader reader = cmd.ExecuteReader();
    while (reader.Read())
    {
        lesTechnician.Add(new Technician(Convert.ToString(reader["nom"]), Convert.ToString(reader["prenom"]), Convert.ToString(reader["role"]), Convert.ToString(reader["Niveauintervention"]), Convert.ToString(reader["formation"]), Convert.ToInt32(reader["Matricule"]));
    }

    return lesTechnician;
}

```

```

4 reference
static List<Personnel> getlesPersonnel()
{
    List<Personnel> lesPersonnel = new List<Personnel>();
    conn.Open();
    MySqlCommand cmd = conn.CreateCommand();
    cmd.CommandText = "SELECT * FROM personnel";
    MySqlDataReader reader = cmd.ExecuteReader();
    while (reader.Read())
    {
        lesPersonnel.Add(new Personnel(Convert.ToInt32(reader["Matricule"]), Convert.ToString(reader["Prenom"]), Convert.ToString(reader["role"]), Convert.ToString(reader["dateembauche"]), Convert.ToString(reader["nom"]));
    }

    return lesPersonnel;
}

5 reference
static List<Intervention> getlesIntervention()
{
    List<Intervention> lesIntervention = new List<Intervention>();
    conn.Open();
    MySqlCommand cmd = conn.CreateCommand();
    cmd.CommandText = "SELECT * FROM intervention";
    MySqlDataReader reader = cmd.ExecuteReader();
    while (reader.Read())
    {
        lesIntervention.Add(new Intervention(Convert.ToString(reader["datecreation"]), Convert.ToString(reader["Object"]), Convert.ToString(reader["niveauurgence"]), Convert.ToString(reader["Etat"]), Convert.ToString(reader["tetechnicien"]), Convert.ToInt32(reader["nom"]));
    }

    return lesIntervention;
}

6 reference
public static void SupprimerTechnician(Technician technicien)
{
    using (SqlConnection connection = new SqlConnection(connStr))
    {
        connection.Open();
        using (SqlCommand command = new SqlCommand("DELETE FROM Technicians WHERE Nom = @Nom AND Prenom = @Prenom AND Role = @Role AND Niveauintervention = @Niveauintervention AND Formation = @Formation", connection))
        {
            command.Parameters.AddWithValue("@Nom", technicien.Nom);
            command.Parameters.AddWithValue("@Prenom", technicien.Prenom);
            command.Parameters.AddWithValue("@Role", technicien.Role);
            command.Parameters.AddWithValue("@niveauintervention", technicien.Niveauintervention);
            command.Parameters.AddWithValue("@formation", technicien.Formation);

            command.ExecuteNonQuery();
        }
    }
}

```

```

7 reference
public static void SupprimerMateriel(Material materiel)
{
    using (SqlConnection connection = new SqlConnection(connStr))
    {
        connection.Open();
        using (SqlCommand command = new SqlCommand("DELETE FROM tabMateriel WHERE nom = @nom AND description = @description AND stock = @stock AND prix = @prix AND fournisseur = @fournisseur AND Matricule = @Matricule", connection))
        {
            command.Parameters.AddWithValue("@nom", materiel.Nom);
            command.Parameters.AddWithValue("@description", materiel.Description);
            command.Parameters.AddWithValue("@stock", materiel.Stock);
            command.Parameters.AddWithValue("@prix", materiel.Prix);
            command.Parameters.AddWithValue("@fournisseur", materiel.Fournisseur);
            command.Parameters.AddWithValue("@Matricule", materiel.Matricule);

            command.ExecuteNonQuery();
        }
    }
}

```

## VI/ TUTO GENERATION SET-UP

### **Étape 1 : Création d'un projet de setup**

Ouvrez Visual Studio.

Cliquez sur "Fichier" -> "Nouveau" -> "Projet".

Dans la boîte de dialogue "Nouveau Projet", recherchez "Projet d'Installation" dans les modèles de projet.

Sélectionnez "Projet d'Installation de Windows" sous "Installateur".

Donnez un nom à votre projet et cliquez sur "OK". Un projet de setup sera créé.

### **Étape 2 : Configuration des paramètres de l'installateur**

Dans le projet de setup, vous verrez deux fichiers : "Product.wxs" et "Product.wxi". "Product.wxs" est le fichier principal de configuration de l'installateur.

Double-cliquez sur "Product.wxs" pour l'ouvrir. Vous pouvez maintenant configurer les paramètres de l'installateur, tels que le nom de votre application, les fichiers à installer, etc.

Sous `<Product>` dans le fichier "Product.wxs", vous pouvez modifier le nom de votre application en modifiant la valeur de l'attribut `Name`.

### **Étape 3 : Ajout de fichiers à installer**

Cliquez avec le bouton droit sur le projet de setup dans l'Explorateur de solutions, puis sélectionnez "Ajouter" -> "Projet de sortie de référence".

Sélectionnez le projet de votre application principale. Cela ajoutera automatiquement les fichiers de votre application à l'installateur.

### **Étape 4 : Personnalisation de l'interface utilisateur**

**(facultatif)**

Si vous souhaitez personnaliser l'interface utilisateur de l'installateur, vous pouvez le faire en modifiant le fichier "Product.wxi". Vous pouvez y définir des images, des textes et des paramètres d'apparence.

## Étape 5 : Compilation du projet de setup

Cliquez avec le bouton droit sur le projet de setup dans l'Explorateur de solutions, puis sélectionnez "Compiler".

Une fois la compilation terminée, vous trouverez un fichier d'installation (un fichier .msi) dans le dossier "bin" du projet de setup.

## Étape 6 : Test et distribution

Exécutez le fichier d'installation pour tester l'installation de votre application. Une fois que tout fonctionne correctement, vous pouvez distribuer le fichier d'installation à vos utilisateurs.

## VI/ GAANT REEL

