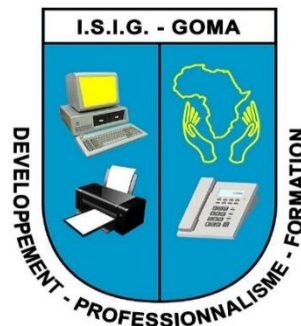


**INSTITUT SUPERIEUR D'INFORMATIQUE ET
DE GESTION**

ISIG-GOMA

Formation, Développement et Professionnalisme

B.P:841-GOMA



EXAMEN DU COURS DE GENIE LOGICIEL

Par : LE GROUPE 1

ANNEE - ACADEMIQUE : 2024-2025

MEMBRES DU GROUPE

LIC 3 / LIAGE

1. AGENORWOTH MUNGANGA (Chef)
2. AHADI KIZA CLÉMENCE
3. ALBERT MULONGA DEBREZIL
4. AMANI BACHA FIDÈLE
5. AMANI BAHAYA DANIEL
6. AMANI BITHA JONATHAN
7. AMISI AMADI CHRISTIAN
8. AMBONE GANYWAMULUME SYLVAIN
9. AMINI MUKWEGE ROSINE
10. ATIABOLI LIJAJA GERMAIN
11. AWA SILIMU ANIFA
12. BACHU ABDUL RAHIM
13. BAGALANE RAMAZANI ANELKA
14. BAHADOOR MULOHWE

LIC 4 / LSI

1. AGISHA MUJUNJU LUC
2. AJIRENGE KARUNGU VIANNEY
3. AKONKWA BUHENDWA ANTHONY
4. AKUZWE GASORE
5. AMANI NTINGINGWA JOSUE
6. AMBOKO EKOMBO ANDRE
7. ANSIMA CHIRIMWAMI BÉNÉDICTE
8. AWEZAYE CHIZUNGU

TRAVAIL DEMANDE

- Créer un repository GitHub de votre choix qui servira à mettre à jour votre code. Vous ajouterez le compte <https://www.github.com/josamuna> comme contributeur (Pour suivre votre travail).
- Tout en vous référant aux slides 65 à 72 des notes de cours ainsi que du code source du repository C# <https://github.com/josamuna/GenieLogicielL12018-2019>, toutes deux relatives à la partie pratique du cours, reproduire le code y correspondant (en évitant tout copier-coller) en respectant ce qui suit:
 - Traduire le diagramme des classes en code source (Y compris la prise en charge des adresses).
 - Créer la Base de Données (SQL Server ou MySQL) avec les tables et les procédures stockées.
 - Créer toutes les interface graphiques (En commençant par celle de login) et faire les différentes manipulations (Ajout, modification et suppression).
 - Créer les trois états de sortie dont : Liste des personnes, liste des personnes avec leurs numéros de téléphone relatifs et liste des personnes avec leurs adresses respectives.
 - Pendant le développement logiciel, une équipe devra travailler en parallèle sur les tests unitaires des différentes fonctions de votre projet.
 - Vous avez le choix entre les langages de programmation Java et C#.
- Faire des commit réguliers jusqu'à finir votre projet.
- Ajouter dans votre repository un fichier qui expliquera tout ce que vous avez fait comme travail.

RESUME

- **Titre du projet** : Application de Gestion des Personnes
- **Objectif** : Tester la capacité des étudiants à développer un projet logiciel complet, en respectant un processus structuré de génie logiciel, comprenant :
 - Traduction d'un diagramme de classes en code source (Java ou C#),
 - Création de la base de données (SQL Server ou MySQL) avec tables et procédures stockées,
 - Développement des interfaces graphiques (login, ajout, modification, suppression),
 - Génération d'états de sortie (listes des personnes, téléphones, adresses),
 - Développement de tests unitaires pour vos fonctionnalités,
 - Utilisation régulière de GitHub pour suivre l'évolution de votre projet,
 - Documentation du projet dans un fichier expliquant tout ce qui a été réalisé,
 - Présentation finale sous forme d'exposé.

En résumé, l'examen vise à évaluer les compétences pratiques en génie logiciel : conception, codage, gestion de version, tests, documentation et présentation.

- **Date de début** : 30 mars 2025
- **Date de fin** : 01 mai 2025
- **Technologies utilisées** :
 - Langage : C#
 - IDE : Visual Studio
 - Base de données : SQL Server
 - Framework : .NET Framework
 - Pour l'accès aux données : ADO.NET
 - Pour l'impression des données : DGVPrinter

ÉTAPES SUIVIES CHAQUE SEMAINE

- Semaine 1 (30 mars – 5 avril)
 - Répartition des tâches.
 - Création du repo.
 - Compréhension du diagramme de classes.
 - Début du codage des classes.
- Semaine 2 (6 – 12 avril)
 - Mise en place de la BD + procédures stockées.
 - Intégration du backend et interfaces de base (login + CRUD).
- Semaine 3 (13 – 19 avril)
 - Implémenter les 3 rapports demandés.
 - Début des tests unitaires.
- Semaine 4 (20 – 27 avril)
 - Finaliser les interfaces + tests.
 - Vérifier le fonctionnement complet (intégration).
 - Ajouter les commits finaux.
- Semaine 5 (28 avril – 01 mai)
 - Finaliser la documentation dans le repo.
 - Préparer la présentation PowerPoint.
 - Répéter l'exposé en groupe.

INTRODUCTION

L'Application de Gestion des Personnes est une solution logicielle développée dans le cadre d'un projet d'examen de Génie Logiciel. Cette application permet la gestion complète des informations personnelles, offrant une interface conviviale pour l'ajout, la consultation, la modification et la suppression des données des personnes.

1. DESCRIPTION GENERALE

Ce projet est une application Windows Forms développée en C# qui permet de gérer efficacement les informations personnelles des individus. L'application offre une interface conviviale pour la gestion des personnes, de leurs adresses, domiciles et numéros de téléphone.

2. STRUCTURE DU PROJET

- Ce projet est organisé selon une architecture modulaire avec les composants suivants :

Examen_GL_Groupe01/

```
|— 1.Classes/           # Classes métier et utilitaires
|— 2. Users Controls/   # Contrôles personnalisés
|— 3. Forms/           # Interfaces utilisateur
|— 4. Connexion/       # Gestion de la connexion à la base de données
|— Properties/        # Configuration du projet
```

- Ce projet est structuré en plusieurs composants principaux :

2.1 Base de Données

Base de données SQL Server nommée *gestion_personne*

Tables principales :

- personne : Stocke les informations de base (nom, postnom, prénom, sexe)
- adresse : Gère les adresses (quartier, commune, ville, pays)
- domicile : Lie les personnes à leurs adresses (avenue, numéro)
- telephone : Stocke les numéros de téléphone des personnes

2.2 Structure du Code

Classes Métier (1.Classes/) :

- Personne.cs : Gestion des personnes
- Adresse.cs : Gestion des adresses
- Domicile.cs : Gestion des domiciles
- Telephone.cs : Gestion des numéros de téléphone
- DGVPrinter.cs : Impression des données

Interface Utilisateur (2. Users Controls/) :

- frmPersonne.cs : Formulaire de gestion des personnes
- frmAdresse.cs : Formulaire de gestion des adresses

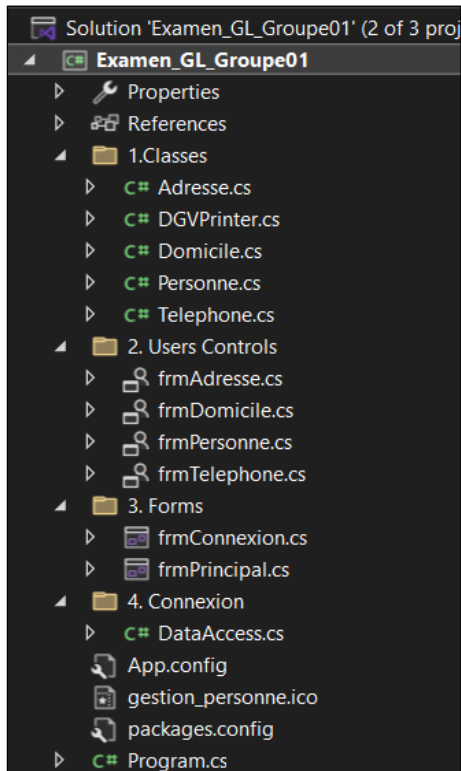
- frmDomicile.cs : Formulaire de gestion des domiciles
- frmTelephone.cs : Formulaire de gestion des téléphones

Formulaires Principaux (3. Forms/) :

- frmPrincipal.cs : Interface principale avec menu de navigation
- frmConnexion.cs : Interface de connexion à la base de données

Accès aux Données (4. Connexion/) :


- DataAccess.cs : Gestion de la connexion à la base de données




3. FONCTIONNALITES PRINCIPALES


3.1 Gestion des Personnes


- Ajout, modification et suppression des personnes
- Stockage des informations : id, nom, postnom, prénom, sexe
- Validation des données et gestion des erreurs
- Recherche et filtrage des personnes




GESTION DES PERSONNES


 Personne

 Adresse







 Domicile

 Téléphone

Déconnection



EXAMEN GENIE LOGICIEL



ID
ahycncytyby



NOM

POSTNOM




PRENOM

SEXE
☒ M ☐ F


Barre de recherche



Id	Nom	Postnom	Prenom	Sexe
dnvnizmhys	AGISHA	MUJUNJU	LUC	M
vuzjrhpls	AMANI	BITHA	JONATHAN	M
ccoaghcdhb	AMINI	MUKWEGE	ROSINE	F
pjvcgazyia	ANSIMA	CHIRIMWAMI	BÉNÉDICTE	F
utukyancse	MULOHWE	MURAD	BAHADOOR	M



- Fonctionnalités d'impression de la liste des personnes



LISTE DES PERSONNES

Liste imprimée : 30/04/2023 19:12


Id	Nom	Postnom	Prenom	Sexe
dnvnizmhys	AGISHA	MUJUNJU	LUC	M
vuzjrhpls	AMANI	BITHA	JONATHAN	M
ccoaghcdhb	AMINI	MUKWEGE	ROSINE	F
pjvcgazyia	ANSIMA	CHIRIMWAMI	BÉNÉDICTE	F
utukyancse	MULOHWE	MURAD	BAHADOOR	M

Examen Génie Logiciel | Groupe 07

Page 1

3.2 Gestion des Adresses

- Création et gestion des adresses complètes
- Organisation hiérarchique : quartier, commune, ville, pays
- Validation des données et gestion des erreurs
- Recherche et filtrage des adresses



GESTION DES PERSONNES

- Personne
- Adresse
- Domicile
- Téléphone

Déconnection

EXAMEN GENIE LOGICIEL

ID

stvsxjipmp

QUARTIER

COMMUNE

VILLE

PAYS

Barre de recherche

	Id	Quartier	Commune	Ville	Pays
▶	ovzutbwzua	Kanyasi	Mbunya	Bunia	RD Congo
	bleckgnueh	Kyeshero	Goma	Goma	RD Congo
	ufykczrff	Lumumba	Mbunya	Bunia	RD Congo
	jdaqqktgev	Mabanga-Sud	Karisimbi	Goma	RD Congo

- Fonctionnalités d'impression de la liste des adresses



LISTE DES ADRESSES

Liste imprimée : 30/04/2025 20:38


Id	Quartier	Commune	Ville	Pays
ovzutbwzua	Kanyasi	Mbunya	Bunia	RD Congo
bleckgnueh	Kyeshero	Goma	Goma	RD Congo
ufykczrff	Lumumba	Mbunya	Bunia	RD Congo
jdaqqktgev	Mabanga-Sud	Karisimbi	Goma	RD Congo

Examen Génie Logiciel | Groupe 01

Page 1

3.3 Gestion des Domiciles

- Association des personnes à leurs adresses
- Gestion des informations détaillées (avenue, numéro)
- Interface intuitive avec sélection via combobox
- Recherche et filtrage des domiciles



GESTION DES PERSONNES

- Personne
- Adresse
- Domicile**
- Téléphone

Déconnexion

EXAMEN GENIE LOGICIEL

ID

ixnowncjd

ID PERSONNE

dnvnlzmhus

PERSONNE

AGISHA MUJUNJU LUC

AVENUE

ID ADRESSE

bleckgnueh




ADRESSE

Kyeshero, Goma, Goma, RD Congo


NUMERO AVENUE

Barre de recherche

	Id	IdPersonne	Noms	IdAdresse	Adresse	Avenue	NumeroAvenue
▶	kxjhxfrcer	vuzjzghips	AMANI BITHA J...	ufykczrfff	RD Congo,Buni...	Boulevard de la ...	14
	eybwtebwjs	pjvcgazlya	ANSIMA CHIRI...	jdaqqktgev	RD Congo,Gom...	Turunga	44
	pjmgxglatj	utukyancse	MULOHWE MU...	bleckgnueh	RD Congo,Gom...	Ruzuba	115






- Fonctionnalités d'impression de la liste des personnes avec leurs adresses respectives



**LISTE DES PERSONNES
AVEC ADRESSES RESPECTIVES**

Liste imprimée : 30/04/2025 20:49




Id	IdPersonne	Noms	IdAdresse	Adresse	Avenue	NumeroAvenue
kxjhxfrcer	vuzjzghips	AMANI BITHA J...	ufykczrfff	RD	Boulevard de la ...	14
eybwtebwjs	pjvcgazlya	ANSIMA CHIRI...	jdaqqktgev	RD	Turunga	44
pjmgxglatj	utukyancse	MULOHWE MU...	bleckgnueh	RD	Ruzuba	115

Examen Génie Logiciel | Groupe 01


Page 1


3.4 Gestion des Téléphones


- Enregistrement des numéros de téléphone
- Association des numéros aux personnes
- Gestion des formats de numéros
- Recherche et filtrage des numéros de téléphone




GESTION DES PERSONNES


 Personne

 Adresse

 Domicile

 Téléphone

Déconnection



EXAMEN GENIE LOGICIEL

ID

nalajrwwlg

ID PERSONNE


PERSONNE


AMINI MUKWEGE ROSINE

INITIAL


NUMERO


Barre de recherche






	Id	IdPersonne	Noms	Initial	Numero
▶	zhbqntgvvd	dvnvlzmhus	AGISHA MUJUNJU LUC	+243	897370705
	ukhoeqohgv	vuzjrglps	AMANI BITHA JONATHAN	+243	826480222
	zxbtpoxlt	ccoaghcdbh	AMINI MUKWEGE ROSINE	+256	787576788
	ekkoykummw	utukyancse	MULOHWE MURAD BAHADOOR	+243	991982155







- Fonctionnalités d'impression de la liste des personnes avec leurs numéros relatifs



**LISTE DES PERSONNES
AVEC NUMEROS RELATIFS**



Liste imprimée : 30/04/2025 21:00

Id	IdPersonne	Noms	Initial	Numero
zhbqntgvvd	dvnvlzmhus	AGISHA MUJUNJU LUC	+243	897370705
ukhoeqohgv	vuzjrglps	AMANI BITHA	+243	826480222
zxbtpoxlt	ccoaghcdbh	AMINI MUKWEGE	+256	787576788
ekkoykummw	utukyancse	MULOHWE MURAD	+243	991982155

Examen Génie Logiciel | Groupe 01

Page 1

4. CARACTERISTIQUES TECHNIQUES

4.1 Sécurité

- Utilisation de procédures stockées pour toutes les opérations CRUD

```
22 public int Enregistrer(Adresse adresse)
23 {
24     int resultat = 0;
25     string strQuery = "sp_AjouterAdresse";
26
27     if (data.OpenConnection())
28     {
29         SqlCommand cmd = new SqlCommand(strQuery, data.connection);
30         cmd.CommandType = CommandType.StoredProcedure;
31         cmd.Parameters.AddWithValue("@id", adresse.Id);
32         cmd.Parameters.AddWithValue("@quartier", adresse.Quartier);
33         cmd.Parameters.AddWithValue("@commune", adresse.Commune);
34         cmd.Parameters.AddWithValue("@ville", adresse.Ville);
35         cmd.Parameters.AddWithValue("@pays", adresse.Pays);
36         resultat = cmd.ExecuteNonQuery();
37         cmd.Dispose();
38         data.CloseConnection();
39     }
40     return resultat;
41 }
42
43 1 reference
44 public int Modifier(Adresse adresse)
45 {
46     int resultat = 0;
47     string strQuery = "sp_ModifierAdresse";
48
49     if (data.OpenConnection())
50     {
51         SqlCommand cmd = new SqlCommand(strQuery, data.connection);
52         cmd.CommandType = CommandType.StoredProcedure;
53         cmd.Parameters.AddWithValue("@id", adresse.Id);
54         cmd.Parameters.AddWithValue("@quartier", adresse.Quartier);
55         cmd.Parameters.AddWithValue("@commune", adresse.Commune);
56         cmd.Parameters.AddWithValue("@ville", adresse.Ville);
57         cmd.Parameters.AddWithValue("@pays", adresse.Pays);
58         resultat = cmd.ExecuteNonQuery();
59         cmd.Dispose();
60         data.CloseConnection();
61     }
62     return resultat;
63 }
64
65 1 reference
66 public int Supprimer(Adresse adresse)
67 {
68     int resultat = 0;
69     string strQuery = "sp_SupprimerAdresse";
70
71     if (data.OpenConnection())
72     {
73         SqlCommand cmd = new SqlCommand(strQuery, data.connection);
74         cmd.CommandType = CommandType.StoredProcedure;
75         cmd.Parameters.AddWithValue("@id", adresse.Id);
76         resultat = cmd.ExecuteNonQuery();
77         cmd.Dispose();
78         data.CloseConnection();
79     }
80     return resultat;
81 }
```

- Validation des données côté serveur et gestion des erreurs et messages d'information

```
215 private void btnEnreg_Click(object sender, EventArgs e)
216 {
217     try
218     {
219         BindingClass();
220         int resultat = domicile.Enregistrer(domicile);
221         if (resultat > 0)
222         {
223             MessageBox.Show("Domicile ajouté avec succès", "Ajout Domicile", MessageBoxButtons.OK, MessageBoxIcon.Information);
224             LoadListDomicile();
225             ClearField();
226         }
227         else
228         {
229             MessageBox.Show("L'ajout du domicile a échoué", "Ajout Domicile", MessageBoxButtons.OK, MessageBoxIcon.Warning);
230         }
231     }
232     catch (Exception ex)
233     {
234         MessageBox.Show("Un problème s'est produit. Contactez l'administrateur svp ! ", "Ajout Domicile", MessageBoxButtons.OK, MessageBoxIcon.Warning);
235     }
236 }
237
238 1 reference
239 private void btnModifier_Click(object sender, EventArgs e)
240 {
241     try
242     {
243         BindingClass();
244         int resultat = domicile.Modifier(domicile);
245         if (resultat > 0)
246         {
247             MessageBox.Show("Domicile modifié avec succès", "Modification Domicile", MessageBoxButtons.OK, MessageBoxIcon.Information);
248             LoadListDomicile();
249             ClearField();
250         }
251         else
252         {
253             MessageBox.Show("La modification a échoué", "Modification Domicile", MessageBoxButtons.OK, MessageBoxIcon.Warning);
254         }
255     }
256     catch (Exception ex)
257     {
258         MessageBox.Show("Un problème s'est produit lors de la modification du domicile. Contactez l'administrateur svp ! ", "Modification Domicile", MessageBoxButtons.OK, MessageBoxIcon.Error);
259     }
260 }
```

4.2 Interface Utilisateur

Cette application propose un design simple et moderne avec :

- Interface commode de connexion à la base de données

The screenshot shows a user interface for a database management application. On the left, a blue sidebar contains a user icon, the title 'GESTION DES PERSONNES', the subtitle 'Examen Génie Logiciel', and the group identifier 'GROUPE 1'. The main content area is white and features a database icon at the top right. Below it, there is a dropdown menu for 'Type base de données' currently set to 'SQLServer'. This is followed by four input fields: 'Serveur' (containing 'BAHADOOR'), 'Base de données', 'Utilisateur', and 'Mot de passe'. A checkbox labeled 'Voir le mot de passe' is located below the password field. At the bottom center, there is a blue 'LOGIN' button.

- Interface principale intuitive avec menu de navigation latéral
- Affichage des données dans des DataGridView
- Fonctionnalités de recherche en temps réel
- Possibilité d'impression des données

4.3 Intégration

- Connexion à la base de données via SQL Server

```
public SqlConnection connection = null;
string connectionString = "Data Source=BAHADOOR;Initial Catalog=gestion_personne;User ID=sa;Password=Espace2027;Encrypt=false";
31 references
public bool OpenConnection()
{
    connection = new SqlConnection(connectionString);
    connection.Open();

    try
    {
        connection = new SqlConnection(connectionString);
        connection.Open();
        return true;
    }
    catch (Exception ex)
    {
        return false;
    }
}

25 references
public void CloseConnection()
{
    if (connection != null)
    {
        connection.Close();
    }
}
```

- Utilisation de procédures stockées pour toutes les opérations
- Gestion des relations entre les différentes entités

5. POINTS FORTS

- Interface utilisateur intuitive et conviviale
- Gestion complète des relations entre les entités
- Validation robuste des données
- Fonctionnalités de recherche et d'impression
- Architecture modulaire et bien organisée

6. TESTS

Les tests ont été réalisés sur :

- Fonctionnalités CRUD
- Ergonomie
- Validation des données
- Gestion des erreurs
- Performance

7. DIFFICULTÉS RENCONTRÉES

- Gestion de la connexion unique à la base de données
- Validation des données
- Optimisation des performances
- Gestion des erreurs

CONCLUSION

Ce projet a permis de développer une application robuste de gestion des personnes, mettant en œuvre les bonnes pratiques de développement et les principes SOLID. L'application offre une solution complète pour la gestion des informations personnelles avec une interface utilisateur intuitive.

Améliorations possibles :

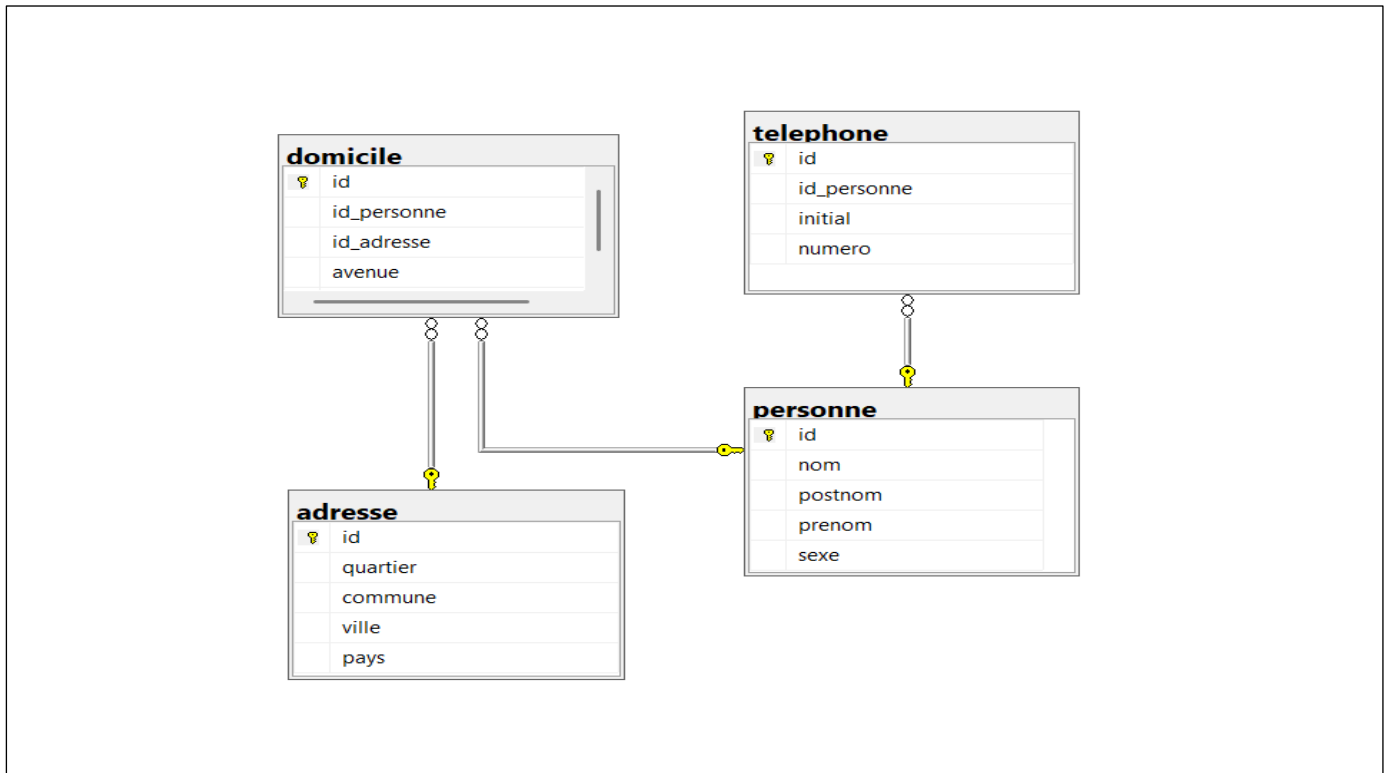
- Ajout de fonctionnalités de recherche avancée
- Implémentation de l'export/import de données
- Ajout de graphiques et statistiques
- Optimisation des performances

Table des matières

MEMBRES DU GROUPE	1
TRAVAIL DEMANDE.....	2
RESUME.....	3
ÉTAPES SUIVIES CHAQUE SEMAINE	4
INTRODUCTION.....	5
1. DESCRIPTION GENERALE	6
2. STRUCTURE DU PROJET	6
2.1 Base de Données.....	6
2.2 Structure du Code	6
3. FONCTIONNALITES PRINCIPALES	8
3.1 Gestion des Personnes	8
3.2 Gestion des Adresses	9
3.3 Gestion des Domiciles	10
3.4 Gestion des Téléphones	11
4. CARACTERISTIQUES TECHNIQUES	12
4.1 Sécurité	12
4.2 Interface Utilisateur	13
4.3 Intégration.....	13
5. POINTS FORTS	14
6. TESTS	14
7. DIFFICULTÉS RENCONTRÉES	14
CONCLUSION.....	15
Table des matières.....	16
Annexes.....	17

Annexes

- Diagrammes de classe



- Requêtes SQL

```

1  --Ajout personne
2  create procedure sp_AjouterPersonne
3      @id varchar(50),@nom varchar(50),@postnom varchar(50),
4      @prenom varchar(50),@sexe varchar(1)
5  as
6  begin
7      if not exists(select * from personne where id=@id)
8          insert into personne(id,nom,postnom,prenom,sexe) values
9              (@id,@nom,@postnom,@prenom,@sexe)
10 end
11 go
12
13 --Modification personne
14 create procedure sp_ModifierPersonne
15     @id varchar(50),@nom varchar(50),@postnom varchar(50),
16     @prenom varchar(50),@sexe varchar(1)
17 as
18 begin
19     update personne set nom=@nom,postnom=@postnom,prenom=@prenom,
20         sexe=@sexe where id=@id
21 end
22 go
23
24 --Suppression personne
25 create procedure sp_SupprimerPersonne
26     @id varchar(50)
27 as
28 begin
29     if exists(select * from personne where id=@id)
30         delete from personne where id=@id
31 end
32 go
33
34 --Affichage personne
35 create procedure sp_AfficherPersonne
36 as
37 begin
38     select * from personne order by nom asc
39 end
40 go

```

- Génération automatique de l'id

```

8 references
public String generateId
{
    get
    {
        Random rnd = new Random();
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < 10; i++)
        {
            char lettre = (char)rnd.Next('a', 'z' + 1);
            sb.Append(lettre);
        }
        string result = sb.ToString();
        return result;
    }
}

```

- Fonctionnalités de recherche en temps réel

```

1 reference
private void txtSearch_KeyPress(object sender, KeyPressEventArgs e)
{
    SqlConnection strcon = new SqlConnection("Data Source=BAHAD00R;Initial Catalog=gestion_personne;User ID=sa;Password=Espace2027;Encrypt=false");

    if (txtSearch.Text != "")
    {
        DgPersonne.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

        SqlDataAdapter adap = new SqlDataAdapter();
        DataSet ds = new DataSet();
        DataView dv = new DataView();

        string query = "select * from personne where id like @search or nom like @search or postnom like @search or prenom like @search or sexe like @search";

        strcon.Open();
        SqlCommand cmd = new SqlCommand(query, strcon);
        cmd.Parameters.AddWithValue("@search", "%" + txtSearch.Text + "%");

        adap = new SqlDataAdapter(cmd);
        adap.Fill(ds);
        dv = new DataView(ds.Tables[0]);
        DgPersonne.DataSource = dv;
        strcon.Close();
    }
    else if (txtSearch.Text == "")
    {
        LoadListPersonne();
    }
}

```