

Contents

I.	Présentation	2
II.	Conception de la Base de Données	3
A.	Modèle Conceptuel de Données (MCD)	3
B.	Modèle relationnel	3
III.	Maquettes des Pages de l'application	4
A.	Page écran Menu :	4
B.	Page « Ajouter un Professionnel »	5
C.	Page « Ajouter un Rendez-Vous »	6
D.	Page « Afficher Liste RDV »	7
IV.	Conception de l'interface	8
A.	Page écran Menu	8
B.	Page « Ajouter un Professionnel »	8
C.	Page « Ajouter un Rendez-Vous »	9
D.	Page « Afficher Liste RDVs »	10
V.	Code Commenté	11
A.	Page Menu	11
1.	Interface XML	11
2.	Code Java	11
B.	Page « Ajouter un Professionnel »	12
1.	Interface XML	12
2.	Code Java	17
C.	Page « Ajouter un Rendez-Vous »	22
1.	Interface XML	22
2.	Code Java	24
D.	Page « Afficher Liste RDVs »	28
1.	Interface XML	28
2.	Code Java	31
E.	Classe objet « Professionnel »	34
F.	Classe objet « Rendez-Vous »	37
G.	Classe d'interaction avec la Base de Données	38

I. Présentation

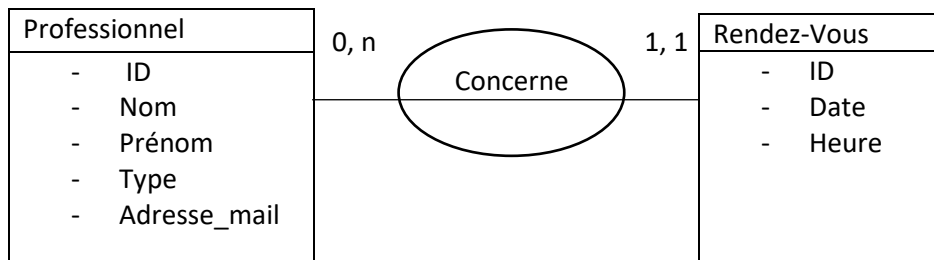
Le projet « Application GSB sous Android » est la création d'une application java, sous Android, permettant la visualisation et la gestion de professionnels et leurs rendez-vous depuis une base de données intégrée.

Pour cela nous allons utiliser différentes pages avec leurs fonction correspondantes définies en détails dans ce rapport.

II. Conception de la Base de Données

A. Modèle Conceptuel de Données (MCD)

Le modèle conceptuel de données est un outil permettant la visualisation globale des relations entre les différentes tables de la base de données.



B. Modèle relationnel

Le Modèle relationnel permet de voir, sous la forme détaillée, le contenu exact des tables de la base de données.

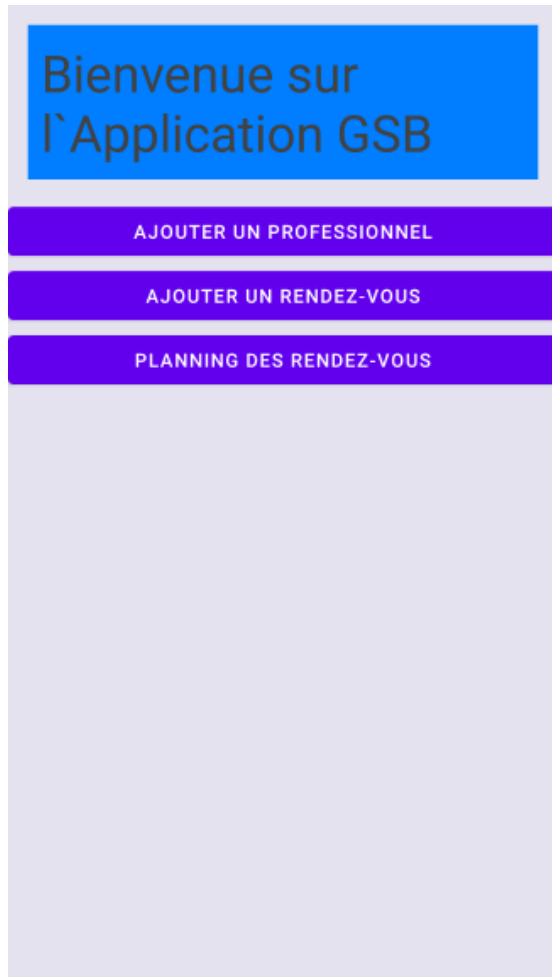
Professionnel (ID, Nom, Prénom, Type, Adresse_mail)

Rendez-Vous (ID, Date, Heure, ID_Pro_Concerne)

III. Maquettes des Pages de l'application

Les maquettes permettent une visualisation simplifiée et rapide de l'interface utilisateur auquel il sera confronté par la suite.

A. Page écran Menu :



B. Page « Ajouter un Professionnel »

➤ MENU ◀

Ajouter un Professionnel

Nom

prenom

Type

Adresse

Mail

ENREGISTRER

AFFICHER LA LISTE DES PROS

Résultat BD :

id :	Nom	prenom	Type
------	-----	--------	------

SUPPRIMER

C. Page « Ajouter un Rendez-Vous »

➤ MENU ➤

Ajouter un Rendez-Vous

< Janvier 1970 >

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

REPLIR LISTE PRO

id :

2 16 AM

3 17 PM

4 18

OBTENIR UN RENDEZ-VOUS

D. Page « Afficher Liste RDV »

→ MENU ←

Liste des Rendez-Vous

AFFICHER LES RDVS

Id Rdv

SUPPRIMER UN RDV

SUPPRIMER TOUT

ID : Date Heure Professionnel concerné

IV. Conception de l'interface

A. Page écran Menu

La page écran Menu, aussi définie comme l'écran d'accueil au démarrage est définie par plusieurs boutons menant aux différentes pages filles de l'application

B. Page « Ajouter un Professionnel »

La page d'ajout d'un professionnel doit permettre à l'utilisateur de renseigner dans des champs spécifiques des informations concernant le professionnel à ajouter, mais aussi d'afficher la liste des professionnels déjà inscrit et enfin vider cette même liste :

➔ MENU ➔

Ajouter un Professionnel

Nom prenom Type

Adresse

Mail

ENREGISTRER

AFFICHER LA LISTE DES PROS

Résultat BD :

id	Nom	prenom	Type
----	-----	--------	------

SUPPRIMER

Les champs d'entrée de texte sont définis avec la caractéristique « hint » permettant l'affichage grisé de l'information à donner dans ce champ.

Le bouton « enregistrer » enregistrera dans la base de données embarquée les informations précédemment renseignées.

Le bouton « afficher la liste des pros » remplira la liste avec les professionnels déjà enregistrés.

Le bouton « supprimer » permettra le vidage de la base de données.

C. Page « Ajouter un Rendez-Vous »

La page d'ajout un rendez-Vous doit permettre à l'utilisateur de renseigner depuis différents widget la date, l'heure et le professionnel concerné par le rendez-vous, mais aussi d'enregistrer ce rendez-vous dans la base de données.

The screenshot shows the 'Ajouter un Rendez-Vous' page. At the top left is a purple button with a right arrow and the text 'MENU'. The title 'Ajouter un Rendez-Vous' is at the top center. Below the title is a calendar widget for 'Janvier 1970'. The calendar shows days of the week (S, M, T, W, T, F, S) and dates. The date '1' is highlighted with a green circle. Below the calendar is a purple button labeled 'REPLIR LISTE PRO'. Below this button is a dropdown menu with a downward arrow. To the right of the dropdown menu is a time picker widget. The time picker shows the hour '2' and the minute '16' with 'AM' selected. Below the time picker is a purple button labeled 'OBTENIR UN RENDEZ-VOUS'.

Le widget Calendrier permet de sélectionner une date.

Le widget Horaire permet de sélectionner une heure.

Le bouton « Remplir liste Pro » permet de remplir la liste déroulante suivante avec les noms des professionnels disponibles.

Le bouton « obtenir un rendez-vous » éponyme de sa fonction permet d'enregistrer le rendez-vous dans la base de données et d'afficher confirmation en bas de la page.

D. Page « Afficher Liste RDVs »

La Page d’affichage de la liste des rendez-Vous doit permettre à l’utilisateur de visualiser et éventuellement supprimer les rendez-vous sélectionnés.



Le bouton « afficher les rdvs » permet l’affichage ou l’actualisation dans la liste depuis la base de données.

Le bouton « supprimer un rdv » permet de supprimer un rendez-vous depuis l’ID renseigné dans le champ de saisie de texte correspondant.

Le bouton « supprimer tout » permet de vider entièrement la liste des rendez-vous, les supprimant de la base de données par la même occasion.

V. Code Commenté

La liste des fichier du projet est disponible sous [GITHUB](#)

A. Page Menu

1. Interface XML

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:orientation="vertical"
    android:background="@color/light_blue"
    tools:layout_editor_absoluteX="29dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/textViewAppName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="@string/menu_page_name"
            android:textColor="@color/cardview_dark_background"
            android:textSize="100px"
            android:padding="10dp"
            android:background="@color/blue"
            android:layout_margin="15dp"/>
        </LinearLayout>

        <Button
            android:id="@+id/buttonGoPage1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"

            android:onClick="clicAfficherPage1"
            android:text="@string/page1_name" />

        <Button
            android:id="@+id/buttonGoPage2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:onClick="clicAfficherPage2"
            android:text="@string/page3_name" />

        <Button
            android:id="@+id/buttonGoPage3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:onClick="clicAfficherPage3"
            android:text="@string/page4_name" />
    </LinearLayout>
```

2. Code Java

```
TextView titre;

/**
 * Foncition dès L'ouverture de La page
 * @param savedInstanceState L'instance actuelle
```

```
*/
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_menu__page);

    titre = findViewById(R.id.textViewAppName);
}

/**
 * Fonction d'affichage de la page une
 * @param view l'écran actuel
 */
public void clicAfficherPage1(View view)
{
    Intent intentAfficher = new Intent(this, MainActivity.class);
    startActivity(intentAfficher);
}

/**
 * Fonction d'affichage de la page deux
 * @param view l'écran actuel
 */
public void clicAfficherPage2(View view)
{
    Intent intentAfficher = new Intent(this, MainActivity2.class);
    startActivity(intentAfficher);
}

/**
 * Fonction d'affichage de la page trois
 * @param view l'écran actuel
 */
public void clicAfficherPage3(View view)
{
    Intent intentAfficher = new Intent(this, MainActivity3.class);
    startActivity(intentAfficher);
}
```

B. Page « Ajouter un Professionnel »

1. Interface XML

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="vertical"
        android:background="@color/light_blue"

        tools:layout_editor_absoluteX="29dp">

        <LinearLayout
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
            android:id="@+id/buttonMenu"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

            android:onClick="clicAfficherMenu"
            android:text="@string/go_to_menu" />
        <TextView
            android:id="@+id/textViewAppName2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

            android:text="@string/page1_name"
            android:textColor="@color/cardview_dark_background"
            android:layout_marginLeft="100dp"/>
    </LinearLayout>

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:id="@+id/editTextNom"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:inputType="textPersonName"
            android:hint="@string/Nom_name" />
        <EditText
            android:id="@+id/editTextPrenom"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:inputType="textPersonName"
            android:hint="@string/Prenom_name" />
        <EditText
            android:id="@+id/editTextType"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:inputType="textPersonName"
            android:hint="@string/Type_name" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <EditText
            android:id="@+id/editTextAdresse"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:inputType="textPersonName"
            android:hint="@string/Adresse_name"/>

    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <EditText
            android:id="@+id/editTextMail"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:inputType="textPersonName"
            android:hint="@string/Mail_name" />

    </LinearLayout>

    <Button
        android:id="@+id/buttonAuRevoir"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="clicEnregistrer"
        android:text="@string/save_name" />

    <Button
        android:id="@+id/buttonAfficher2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:onClick="buttonMajListe"
        android:text="@string/pro_list_name" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Résultat BD :"/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/resultBD"
            android:text="" />
```

```
</LinearLayout>

<HorizontalScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="5dp"
    android:layout_marginRight="5dp"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="5dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
    >
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical"
        >
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/resultBD6"
                android:text="@string/id_name"/>
        </LinearLayout>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text=""
            android:layout_marginLeft="20dp"
            android:layout_marginRight="20dp"/>
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical"
        >
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
```

```
        android:id="@+id/resultBD1"
        android:text="@string/Nom_name"/>
</LinearLayout>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:layout_marginLeft="50dp"
    android:layout_marginRight="50dp"/>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/resultBD2"
        android:text="@string/Prenom_name"/>
</LinearLayout>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:layout_marginLeft="50dp"
    android:layout_marginRight="50dp"/>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/resultBD3"
        android:text="@string/Type_name"/>
</LinearLayout>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:layout_marginLeft="50dp"
    android:layout_marginRight="50dp"/>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/resultBD4"
        android:text="@string/Adresse_name"/>
</LinearLayout>
```



```

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text=""
            android:layout_marginLeft="50dp"
            android:layout_marginRight="50dp"/>
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            >
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/resultBD5"
                android:text="@string/Mail_name"/>
        </LinearLayout>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text=""
            android:layout_marginLeft="50dp"
            android:layout_marginRight="50dp"/>

    </LinearLayout>

</HorizontalScrollView>

<Button
    android:id="@+id/ButtonDelete"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="delAllData"
    android:text="@string/Del_name" />

</LinearLayout>
</ScrollView>

```

2. Code Java

```

Globals g;
public ArrayList<Professionnels> ListePro = new ArrayList<Professionnels>();

SQLiteDataBaseHelper db;

/**
 * Liste des objets graphique liés
 */

TextView resultTextBD;

```

```
EditText editNom;
EditText editPrenom;
EditText editType;
EditText editAdresse;
EditText editMail;

TextView txtNom;
TextView txtPrenom;
TextView txtType;
TextView txtAdresse;
TextView txtMail;
TextView txtID;

/**
 * Fonction dès l'ouverture de la page
 * @param savedInstanceState Les données actuelles de l'application
 */
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //Création de la Base de Données
    db = new SQLiteDataBaseHelper(this);

    /**
     * Liens entre Les objets de l'interface graphique et Les objets virtuels
     */
    editNom = findViewById(R.id.editTextNom);
    editPrenom = findViewById(R.id.editTextPrenom);
    editType = findViewById(R.id.editTextType);
    editAdresse = findViewById(R.id.editTextAdresse);
    editMail = findViewById(R.id.editTextMail);

    resultTextBD = findViewById(R.id.resultBD);
    resultTextBD.setText("texteee");

    txtNom = findViewById(R.id.resultBD1);
    txtPrenom = findViewById(R.id.resultBD2);
    txtType = findViewById(R.id.resultBD3);
    txtAdresse = findViewById(R.id.resultBD4);
    txtMail = findViewById(R.id.resultBD5);
    txtID = findViewById(R.id.resultBD6);

    majListe();
}

/**
 * Bouton affichage de l'écran N°2
 * @param view
 */
public void clicAfficherPage2(View view)
{
    Intent intentAfficher = new Intent(this, MainActivity2.class);
    startActivity(intentAfficher);
}
```

```
}

/**
 * Bouton affichage de l'écran 3
 * @param view
 */
public void clicAfficherMenu(View view)
{
    Intent intentAfficher = new Intent(this, Menu_Page.class);
    startActivity(intentAfficher);
}

/**
 * Fonction d'enregistrement d'un Professionnel
 * @param view
 */
public void clicEnregistrer(View view)
{
    try {
        String Nom, Prenom, Adresse, Type, Mail;

        Nom = editNom.getText().toString();
        Prenom = editPrenom.getText().toString();
        Type = editType.getText().toString();
        Adresse = editAdresse.getText().toString();
        Mail = editMail.getText().toString();

        Professionnels unPro = new Professionnels(Nom, Prenom, Type, Adresse,
Mail);

        db.InsertDataPro(unPro);

        majListe();
    }

    catch (Exception e){
        resultTextBD.setText(e.getMessage());
    }

}

/**
 * Fonction de mise à jour des trois listes
 * @param view
 */
public void buttonMajListe(View view)
{
    majListe();
}

/**
 * Fonction de mise à jour du label avec le contenu de la base de données
```

```
*/  
public void majListe()  
{  
    txtPrenom.setText("Nom \n ");  
    txtNom.setText("Prenom \n ");  
    txtType.setText("Type \n ");  
    txtAdresse.setText("Adresse \n ");  
    txtMail.setText("Mail \n ");  
    txtID.setText("Id \n ");  
    resultTextBD.setText("Professionnel ajouté");  
  
    majListeNom();  
    majListePrenom();  
    majListeType();  
    majListeAdresse();  
    majListeMail();  
    majListeID();  
}  
  
/**  
 * Fonction de mise à jour de la liste des Noms  
 */  
public void majListeNom()  
{  
    try {  
        Cursor dataNom = db.getAllData();  
        while (dataNom.moveToNext())  
        {  
            txtNom.setText(txtNom.getText() + " \n " + (dataNom.getString(1)));  
        }  
    }  
    catch (Exception e)  
    {  
        txtNom.setText(e.getMessage());  
    }  
}  
  
/**  
 * Fonction de mise à jour de la liste des Prénoms  
 */  
public void majListePrenom()  
{  
    try {  
        Cursor dataNom = db.getAllData();  
        while (dataNom.moveToNext())  
        {  
            txtPrenom.setText(txtPrenom.getText() + " \n " +  
(dataNom.getString(0)));  
        }  
    }  
    catch (Exception e)  
    {  
        txtPrenom.setText(e.getMessage());  
    }  
}  
  
/**
```

```
* Fonction de mise à jour de La liste des Types
*/
public void majListeType()
{
    try {
        Cursor dataNom = db.getAllData();
        while (dataNom.moveToNext())
        {
            txtType.setText(txtType.getText() + " \n " + (dataNom.getString(2)));
        }
    }
    catch (Exception e)
    {
        txtType.setText(e.getMessage());
    }
}

/**
* fonction de mise à jour de La liste des Adresses
*/
public void majListeAdresse()
{
    try {
        Cursor dataNom = db.getAllData();
        while (dataNom.moveToNext())
        {
            txtAdresse.setText(txtAdresse.getText() + " \n " +
(dataNom.getString(3)));
        }
    }
    catch (Exception e)
    {
        txtAdresse.setText(e.getMessage());
    }
}

/**
* Fonction de mise à jour de La liste des Mails
*/
public void majListeMail()
{
    try {
        Cursor dataNom = db.getAllData();
        while (dataNom.moveToNext())
        {
            txtMail.setText(txtMail.getText() + " \n " + (dataNom.getString(4)));
        }
    }
    catch (Exception e)
    {
        txtMail.setText(e.getMessage());
    }
}

/**
* Fonction de mise à jour de La liste des IDs
*/
public void majListeID()
{

```

```

    try {
        Cursor dataNom = db.getAllData();
        while (dataNom.moveToNext())
        {
            txtID.setText(txtID.getText() + " \n " + (dataNom.getString(5)));
        }
    }
    catch (Exception e)
    {
        txtID.setText(e.getMessage());
    }
}

/**
 * fonction de suppression de toutes les données de la base de données
 * @param view
 */
public void delAllData(View view)
{
    try {
        db.delAllData();
        resultTextBD.setText("Base de données supprimée");
        majListe();
    }
    catch (Exception e)
    {
        resultTextBD.setText(e.getMessage());
    }
}

```

C. Page « Ajouter un Rendez-Vous »

1. Interface XML

```

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="vertical"

        tools:layout_editor_absoluteX="29dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <Button
                android:id="@+id/buttonBienvenue"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

                android:onClick="clicAfficherMenu"
                android:text="@string/go_to_menu" />

```

```
<TextView
    android:id="@+id/textViewAppName2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:text="@string/page3_name"
    android:textColor="@color/cardview_dark_background"
    android:layout_marginLeft="100dp"/>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="311dp"
    android:orientation="horizontal">

    <CalendarView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/CalendarView"/>

</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/btn_fill_list"
            android:onClick="clieFillList"
            android:text="@string/fill_list_name" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/id_name" />

        <Spinner
            android:id="@+id/SpinnerDesPros"
            android:layout_width="match_parent"
            android:layout_height="44dp" />

    </LinearLayout>
</LinearLayout>
<TimePicker
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/TimePicker"
    android:timePickerMode="spinner"/>
```

```

        </LinearLayout>
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <Button
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/btn_get_select"
                android:onClick="clicGetRDV"
                android:text="@string/get_select_name" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:id="@+id/TextView1"
                android:text="" />

        </LinearLayout>

    </LinearLayout>

</ScrollView>

```

2. Code Java

```

Globals g;
Spinner spin;
TextView text;
String curDate;
String[] langages = {"php", "html", "java", "javascript", "C#", "python"};
TimePicker timerpicker;
CalendarView calendarView;

/**
 * L'heure sélectionnée
 */
int heures;
/**
 * Les minutes selctionnée
 */
int minutes;
/**
 * L'id du pro sélectionnée
 */
int idSelect;

SQLiteDataBaseHelper db;
/**
 * La Liste des Pro (Nom - Prenom - Type)
 */
String[] listeDesPros;

```



```
/**
 * Fonction dès l'ouverture de la page
 * @param savedInstanceState l'instance de l'application
 */
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2);

    g = Globals.getInstance();

    /**
     * Liens entre les objets de l'interface graphique et les objets virtuels
     */
    spin=findViewById(R.id.SpinnerDesPros);
    text = findViewById(R.id.TextView1);
    timerpicker = findViewById(R.id.TimePicker);
    timerpicker.setIs24HourView(true);
    calendarView = findViewById(R.id.CalendarView);

    db = new SQLiteDatabaseHelper(this);

    try {

        listeDesPros = db.getListeIdPro();
        text.setText(db.getNbIdCount() + " " + listeDesPros[2]);

    }
    catch (Exception e)
    {
        text.setText(e.getMessage());
    }

    calendarView.setOnDateChangeListener((new CalendarView.OnDateChangeListener()
{
    /**
     * Fonction de définition de la date du CalendarView et stockage dans une
    variable
     * @param view L'écran actuel
     * @param year L'année
     * @param month le mois
     * @param dayOfMonth le jour du mois
     */
    @Override
    public void onSelectedDayChange(@NonNull CalendarView view, int year, int
month, int dayOfMonth) {

        String leMois = String.valueOf(month+1);
        if((month+1)<10)
        {
            leMois = ("0"+String.valueOf(month+1));
        }
        String leJour = String.valueOf(dayOfMonth);
        if((dayOfMonth)<10)
        {
            leJour = ("0"+String.valueOf(dayOfMonth));
        }
    }
})
```

```
        curDate = String.valueOf(1eJour) + "/" + String.valueOf(1eMois);
    }
}));

timerpicker.setOnTimeChangeListener(new TimePicker.OnTimeChangeListener() {
    /**
     * Fonction de définition de l'heure sélectionnée
     * @param view l'écran actuel
     * @param hourOfDay l'heure
     * @param minute les minutes
     */
    @Override
    public void onTimeChanged(TimePicker view, int hourOfDay, int minute) {
        heures = 0;
        minutes = 0;
        heures = hourOfDay;
        minutes = minute;
    }
});

}

/**
 * clic bouton 1 : affichage de la page 1
 * @param view
 */
public void clicAfficherPage1(View view)
{
    Intent intentAfficher = new Intent(this, MainActivity.class);
    startActivity(intentAfficher);
}

/**
 * clic bouton menu : affichage page menu
 * @param view
 */
public void clicAfficherMenu(View view)
{
    Intent intentAfficher = new Intent(this, Menu_Page.class);
    startActivity(intentAfficher);
}

/**
 * clic bouton 2 : affichage page1
 * @param view
 */
public void clicPage1(View view)
{
    TextView text = findViewById(R.id.textViewAppName);
    g = Globals.getInstance();
    String nom = g.getNom();
    text.setText(nom);
}
```

```
/**
 * fonction de remplissage du spin avec les nom/prénom des professionnels
 enregistré
 * @param view
 */
public void clicFillList(View view)
{
    ArrayAdapter<String> AllPros = new ArrayAdapter<String>(this,
    android.R.layout.simple_spinner_item, listeDesPros);

    AllPros.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spin.setAdapter((AllPros));

    spin.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position,
    long id) {
            idSelect=position;
        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) {

        }
    });
}

/**
 * getteur de l'id de la ligne du spin sélectionnée
 * @param view
 */
public void clicGetSelect(View view)
{
    String ProSelected = listeDesPros[idSelect];
    text.setText(ProSelected);
}

/**
 * Clic bouton obtenir un Rendez-Vous
 * @param view
 */
public void clicGetRDV(View view)
{
    try
    {
        String heure = "00h00";

        String laHeure = String.valueOf(heures);
        String laMinute = String.valueOf(minutes);

        if(heures<10)
            laHeure = ("0"+laHeure);
    }
}
```

```

        if(minutes<10)
            laMinute = ("0"+laMinute);

        if(heures==0)
            laHeure = ("00");
        if(minutes==0)
            laMinute = ("00");

        heure = (laHeure + "h" + laMinute);

        RendezVous unRdv = new RendezVous(curDate, heure, idSelect);
        //db.InsertDataRDV(unRdv);
        text.setText("Rendez-vous ajouté");
        db.InsertDataRDV(curDate, heure, idSelect);

        Cursor listerv = db.getListerRDV();
        while (listerv.moveToNext())
        {
            text.setText(text.getText() + " \n " + " date : " +
(listerv.getString(0)) + "heure : " + (listerv.getString(1)) + "pro concerne : "
+ (listerv.getString(2)));
        }
    }
    catch(Exception e)
    {
        text.setText(e.getMessage());
    }
}

```

D. Page « Afficher Liste RDVs »

1. Interface XML

```

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <Button
                android:id="@+id/buttonBienvenue"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

                android:onClick="clicAfficherMenu"
                android:text="@string/go_to_menu" />
            <TextView
                android:id="@+id/textViewAppName2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

                android:text="@string/listRdv_name"

```

```
        android:textColor="@color/cardview_dark_background"
        android:layout_marginLeft="100dp"/>
    </LinearLayout>

    <Button
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/btn_fill_rdv"
        android:text="@string/fillRDV_name"
        android:onClick="cliclAfficherRDV"/>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <EditText
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:hint="Id Rdv"
            android:id="@+id/editText_idRdv"/>
        <Button
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/btn_del_a_rdv"
            android:text="Supprimer un RDV"
            android:onClick="clicDelARDv"/>

    </LinearLayout>
    <Button
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/btn_del_all_rdv"
        android:text="@string/delallrdv_name"
        android:onClick="clicDelAllRDv"/>

    <HorizontalScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:layout_marginRight="5dp"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="5dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            >
```

```
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/resultID_rdv"
        android:layout_marginRight="20dp"
        android:text="ID :"/>
</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/resultBD2_1"
        android:text="@string/date_name"/>
</LinearLayout>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"/>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/resultBD2_2"
        android:text="@string/heure_name"/>
</LinearLayout>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"/>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```

        android:id="@+id/resultBD2_3"
        android:text="@string/Pro_name"/>
    </LinearLayout>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:layout_marginLeft="50dp"
        android:layout_marginRight="50dp"/>

    </LinearLayout>

</HorizontalScrollView>
</LinearLayout>
</ScrollView>

```

2. Code Java

```

SQLiteDataBaseHelper db;

TextView textDate;
TextView textHeure;
TextView textPro;
TextView txtid;

EditText editTextRdv;

/**
 * Fonction dès l'ouverture de la page
 * @param savedInstanceState l'instance actuelle
 */
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main3);

    db = new SQLiteDataBaseHelper(this);

    textDate = findViewById(R.id.resultBD2_1);
    textHeure = findViewById(R.id.resultBD2_2);
    textPro = findViewById(R.id.resultBD2_3);
    editTextRdv = findViewById(R.id.editText_idRdv);
    txtid = findViewById(R.id.resultID_rdv);

    majListeDate();
    majListePro();
    majListeHeure();
}

/**

```

```
* clic bouton menu : affichage page menu
* @param view
*/
public void clicAfficherMenu(View view)
{
    Intent intentAfficher = new Intent(this, Menu_Page.class);
    startActivity(intentAfficher);
}

/**
* Affichage de La Liste des rendez-vous enregistrés dans La BD
* @param view l'écran actuel
*/
public void cliclAfficherRDV(View view)
{
    textHeure.setText("Date : ");
    textHeure.setText("Heure : ");
    textPro.setText("Professionnel Concerné : ");
    txtid.setText("ID :");

    majListeDate();
    majListePro();
    majListeHeure();
}

/**
* Fonctino mise à jour des IDs
*/
public void majListeID()
{
    try{
        Cursor dataID = db.getListeRDV();
        txtid.setText("ID :");
        while(dataID.moveToNext())
        {
            txtid.setText(txtid.getText() + " \n " + (dataID.getInt(4)));
        }
    }
    catch(Exception e)
    {
        txtid.setText(e.getMessage());
    }
}

/**
* Fonction mise à jour liste des Dates
*/
public void majListeDate()
{
    try {
        Cursor dataDate = db.getListeRDV();

        textDate.setText("Date :");
        while (dataDate.moveToNext())
        {
            textDate.setText(textDate.getText() + " \n " +
(dataDate.getString(1)));
        }
    }
}
```



```
    }
    catch (Exception e)
    {
        textDate.setText(e.getMessage());
    }
}

/**
 * Fonction mise à jour liste des Heures
 */
public void majListeHeure()
{
    try {
        Cursor dataHeure = db.getListeRDV();
        textHeure.setText("Heure :");
        while (dataHeure.moveToNext())
        {
            textHeure.setText(textHeure.getText() + " \n " +
(dataHeure.getString(0)));
        }

    }
    catch (Exception e)
    {
        textHeure.setText(e.getMessage());
    }
}

/**
 * Fonction mise à jour liste des Pros
 */
public void majListePro()
{
    try {
        Cursor dataPro = db.getListeRDV();
        String[] listePro = db.getListeIdPro();
        textPro.setText("PProfessionnel Concerné :");
        while (dataPro.moveToNext())
        {
            textPro.setText(textPro.getText() + " \n " + (dataPro.getString(2)) +
" " + listePro[dataPro.getInt(2)]);

        }

    }
    catch (Exception e)
    {
        textPro.setText(e.getMessage());
    }
}

/**
 * Fonction suppression de la liste entière des rendez-vous
 * @param view l'écran actuel
 */
public void clicDelAllRDv(View view)
{
    try {
```

```

        db.delAllDataRDV();
        textDate.setText("Date : ");
        textHeure.setText("Heure : ");
        textPro.setText("Professionnel Concerné : ");

        majListeDate();
        majListePro();
        majListeHeure();
    }
    catch (Exception e)
    {
        textPro.setText(e.getMessage());
    }
}

/**
 * Fonction de suppression d'un rendez-vous selon son id
 * @param view l'écran actuel
 */
public void clicDelARDv(View view)
{
    try {
        String id = editTextRdv.getText().toString();
        db.delARDv(id);
        editTextRdv.setText(" ");
        majListeHeure();
        majListePro();
        majListeDate();
    }
    catch (Exception e)
    {
        editTextRdv.setText(e.getMessage());
    }
}

```

E. Classe objet « Professionnel »

```

Globals g;
/**
 * Le nom
 */
private String nom;
/**
 * Le prenom
 */
private String prenom;
/**
 * Le type
 */
private String type;
/**
 * L'adresse
 */
private String adresse;
/**
 * Le mail
 */
private String mail;
/**

```

```
* L'id
*/
private int id;

/**
 * Constructeur de la classe Professionnel
 * @param nom son nom
 * @param prenom son prenom
 * @param type son type
 * @param adresse son adresse
 * @param mail son mail
 */
public Professionnels(String nom, String prenom, String type, String adresse,
String mail)
{
    Globals.getInstance();

    this.nom = nom;
    this.prenom = prenom;
    this.type = type;
    this.adresse = adresse;
    this.mail = mail;
}

/**
 * // // // // //LISTE DES SETTERS// // // // //
 */

/**
 * Setter du nom
 * @param nom Le nouveau nom
 */
public void setNom(String nom) {
    this.nom = nom;
}

/**
 * Setter du prénom
 * @param prenom Le nouveau prénom
 */
public void setPrenom(String prenom) {
    this.prenom = prenom;
}

/**
 * Setter du type
 * @param type Le nouveau type
 */
public void setType(String type) {
    this.type = type;
}

/**
 * Setter de l'adresse
 * @param adresse La nouvelle adresse
 */
public void setAdresse(String adresse) {
    this.adresse = adresse;
}
```

```
}

/**
 * Setter du mail
 * @param mail le nouveau mail
 */
public void setMail(String mail) {
    this.mail = mail;
}

/**
 * Setter de l'id
 * @param id le nouvel id
 */
public void setId(int id) {
    this.id = id;
}

/**
 * // // // // //LISTE DES GETTERS// // // // //
 */

/**
 * Getter du nom
 * @return le nom
 */
public String getNom() {
    return nom;
}

/**
 * Getter du prénom
 * @return le prénom
 */
public String getPrenom() {
    return prenom;
}

/**
 * Getter du type
 * @return le type
 */
public String getType() {
    return type;
}

/**
 * Getter de l'adresse
 * @return l'adresse
 */
public String getAdresse() {
    return adresse;
}

/**
 * Getter du mail
 * @return le mail
 */
public String getMail() {
```

```

        return mail;
    }

    /**
     * Getter de l'id
     * @return l'id
     */
    public int getId() {
        return id;
    }

```

F. Classe objet « Rendez-Vous »

```

Globals g;
/**
 * La date
 */
private String date;
/**
 * L'heure
 */
private String heure;
/**
 * Le professionnel concerné (objet)
 */
private Professionnels ProfessionnelConcerne;
/**
 * L'ID du professionnel concerné
 */
private int ProConcerne;

/**
 * Constructeur de la classs RendezVous
 * @param date la date du rendez-vous (format JJ/MM/AAAA)
 * @param heure l'heure du rendez-vous
 */
public RendezVous(String date, String heure, int ProConcerne) {
    this.date = date;
    this.heure = heure;
    this.ProConcerne = ProConcerne;
}

/**
 * // // // // //LISTE DES GETTERS// // // // //
 */

/**
 * getter date
 * @return la date
 */
public String getDate() {
    return date;
}

/**
 * Getter heure
 * @return l'heure
 */

```

```

public String getHeure() {
    return heure;
}

/**
 * // // // // //LISTE DES SETTERS// // // // //
 */

/**
 * Setter date
 * @param date
 */
public void setDate(String date) {
    this.date = date;
}

/**
 * Setter heure
 * @param heure
 */
public void setHeure(String heure) {
    this.heure = heure;
}

/**
 * Setter Professionnel concerné
 * @param id L'id du professionnel concerné
 */
public void setProfessionnelConcerne(int id)
{
    this.ProConcerne = id;
}

/**
 * Getteur du professionnel concerné
 * @return L'id du professionnel concerné
 */
public int getProConcerne()
{
    return this.getProConcerne();
}

```

G. Classe d'interaction avec la Base de Données

```

/**
 * Le nom de la base de donnée
 */
public static final String DATABASE_NAME = "GSB";

/**
 * Le nom de la table Professionnels
 */
public static final String TABLE_NAME_PRO = "Professionnels";

/**
 * Le nom de la table Rendez-Vous
 */
public static final String TABLE_NAME_RDV = "Rendezvous";

/**
 * Liste des colonnes de la Table Professionnel
 */

```

```
/**
 * colonne "nom"
 */
public static final String COL_1_1 = "Nom";
/**
 * colonne "prenom"
 */
public static final String COL_1_2 = "Prenom";
/**
 * colonne "type"
 */
public static final String COL_1_3 = "Type";
/**
 * "colonne adresse"
 */
public static final String COL_1_4 = "Adresse";
/**
 * colonne "mail"
 */
public static final String COL_1_5 = "Mail";
/**
 * colonne "id"
 */
public static final String COL_1_6 = "id";

/**
 * Liste des colonnes de la table Rendez-vousS
 */
/**
 * colonne "date"
 */
public static final String COL_2_1 = "Date";
/**
 * colonne "heure"
 */
public static final String COL_2_2 = "Heure";
/**
 * colonne "idPro"
 */
public static final String COL_2_3 = "idPro";
/**
 * colonne "idRdv"
 */
public static final String COL_2_4 = "idRdv";

Globals g;

/**
 * Constructeur de La base de données
 * @param context prise du contexte actuel des bases de données déjà enregistréesS
 */
public SQLiteDatabaseHelper(Context context)
{
```

```
        super(context, DATABASE_NAME, null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE " + TABLE_NAME_PRO + "(Nom TEXT, Prenom TEXT, Type TEXT, Adresse TEXT, Mail TEXT, id INTEGER PRIMARY KEY AUTOINCREMENT)");
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME_RDV);
        db.execSQL("CREATE TABLE " + TABLE_NAME_RDV + "(Date TEXT, Heure TEXT, idPro INTEGER, idRdv INTEGER PRIMARY KEY AUTOINCREMENT)");
    }

    /**
     * Fonction Override de mise à jour de la base de données
     * @param db la base de données à metre à jour
     * @param oldVersion numéro de l'ancienne version
     * @param newVersion numéro de la nouvelle version
     */
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME_PRO);
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME_RDV);
        onCreate(db);
    }

    /**
     * Insertion d'un pro dans la base de données à partir de tout ses paramètres
     * @param Nom Le nom
     * @param Prenom Le prénom
     * @param Type son type
     * @param Adresse son adresse
     * @param Mail son adresse mail
     */
    public void InsertDataPro(String Nom, String Prenom, String Type, String Adresse, String Mail)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();

        contentValues.put(COL_1_1, Nom);
        contentValues.put(COL_1_2, Prenom);
        contentValues.put(COL_1_3, Type);
        contentValues.put(COL_1_4, Adresse);
        contentValues.put(COL_1_5, Mail);
        db.insert(TABLE_NAME_PRO, null, contentValues);
    }

    /**
     * Insertion d'un pro dans la base de donnée à partir d'un objet pro
     * @param unPro Le professionnel
     */
    public void InsertDataPro(Professionnels unPro)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();

        contentValues.put(COL_1_1, unPro.getNom());
        contentValues.put(COL_1_2, unPro.getPrenom());
    }
}
```



```
        contentValues.put(COL_1_3, unPro.getType());
        contentValues.put(COL_1_4, unPro.getAdresse());
        contentValues.put(COL_1_5, unPro.getMail());
        db.insert(TABLE_NAME_PRO, null, contentValues);
        db.close();
    }

    /**
     * Insertion d'un rendez-vous dans la base de donnée à partir de ses paramètres
     * @param date sa date
     * @param heure son heure
     */
    public void InsertDataRDV(String date, String heure, int idPro)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();

        contentValues.put(COL_2_1, date);
        contentValues.put(COL_2_2, heure);
        contentValues.put(COL_2_3, idPro);

        db.insert(TABLE_NAME_RDV, null, contentValues);
        db.close();
    }

    /**
     * Insertion d'un rendez-vous dans la base de donnée à partir d'un objet rendez-
     vous
     * @param unRDV Le rendez-vous
     */
    public void InsertDataRDV(RendezVous unRDV)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();

        contentValues.put(COL_2_1, unRDV.getDate());
        contentValues.put(COL_2_2, unRDV.getHeure());
        contentValues.put(COL_2_3, unRDV.getProConcerne());

        db.insert(TABLE_NAME_RDV, null, contentValues);
        db.close();
    }

    /**
     * Récupération de l'entièreteé de la table professionnel
     * @return tableau des pro (format Cursor)
     */
    public Cursor getAllData()
    {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor result = db.rawQuery("SELECT * FROM " + TABLE_NAME_PRO, null);

        return result;
    }

    /**
```

```
* Suppression d'une ligne SQL selon son ID
* @param id l'id de la ligne à supprimer
*/
public void delAData(int id)
{
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete(TABLE_NAME_PRO, COL_1_1+"="+id, null);

    db.close();
}

/**
* Suppression de toute les données d'une table PRO
*/
public void delAllData()
{
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete(TABLE_NAME_PRO, null, null);
    db.delete("sqlite_sequence", null, null);

    db.close();
}

/**
* Suppression de toute les données d'une table RDV
*/
public void delAllDataRDV()
{
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete(TABLE_NAME_RDV, null, null);
    db.delete("sqlite_sequence", null, null);

    db.delete("sqlite_sequence", null, null);

    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME_RDV);
    db.execSQL("CREATE TABLE " + TABLE_NAME_RDV + "(Date TEXT, Heure TEXT, idPro
INTEGER, idRdv INTEGER PRIMARY KEY AUTOINCREMENT)");

    db.close();
}

/**
* Getteur de la liste des ID des Professionnels
* @return Le tableau des nom + prenom des pros
*/
public String[] getListeIdPro()
{
    SQLiteDatabase db = this.getReadableDatabase();

    Cursor nbID = db.rawQuery("SELECT * FROM "+ TABLE_NAME_PRO, null);
    int nbIDCount = nbID.getCount();

    String[] test = new String[nbIDCount];
    int i=0;
    while (nbID.moveToNext())
    {
        test[i] = (nbID.getString(0) + " " + nbID.getString(1) + ", type : "+
nbID.getString(2));
        i++;
    }
}
```

```
    }

    return test;
}

/**
 * Getteur du nombre de professionnels enregistré
 * @return Le nombre de professionnels enregistré
 */
public int getNbIdCount()
{
    SQLiteDatabase db = this.getReadableDatabase();

    Cursor nbID = db.rawQuery("SELECT * FROM " + TABLE_NAME_PRO, null);
    int nbIDCount = nbID.getCount();

    return nbIDCount;
}

/**
 * Getteur Liste des rdvs (cursor)
 * @return Le curseur contenant la liste des rdvs
 */
public Cursor getListeRDV()
{
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor list = db.rawQuery("SELECT * FROM " + TABLE_NAME_RDV, null);

    return list;
}

/**
 * Fonction de suppression d'un rendez vous selon son ID
 * @param id L'id du rendez-vous à supprimer
 */
public void delARdv(String id)
{
    SQLiteDatabase db = this.getReadableDatabase();
    db.delete(TABLE_NAME_RDV, COL_2_4 + " = " + id , null);

    db.delete("sqlite_sequence", null, null);
}
```