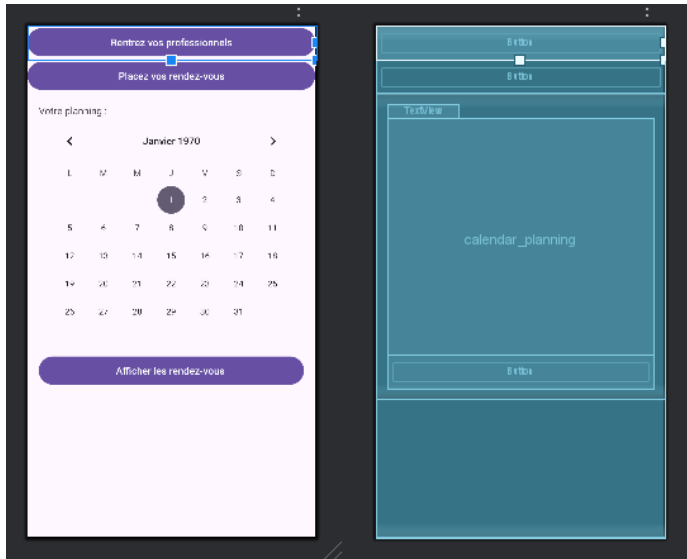


XML de la page d'accueil : page principal



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <Button
        android:id="@+id/nouveau_professionnel"
        android:layout_width="match_parent"
        android:onClick="professionnel"
        android:layout_height="wrap_content"
        android:text="Rentrez vos professionnels"/>

    <Button
        android:id="@+id/placement_rdv"
        android:layout_width="match_parent"
        android:onClick="rdv"
        android:layout_height="wrap_content"
        android:text="Placez vos rendez-vous"/>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
```

```

        android:orientation="vertical"
        android:padding="16dp">
        <TextView
            android:id="@+id/texte_cal"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Votre planning : "/>
        <CalendarView
            android:id="@+id/calendar_planning"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
        <Button
            android:id="@+id/planning"
            android:layout_width="match_parent"
            android:onClick="planning"
            android:layout_height="wrap_content"
            android:text="Afficher les rendez-vous "/>
    </LinearLayout>
</ScrollView>
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

```

Code java de la page d'accueil : page principal

```

package com.example.projet_final_android;

import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.CalendarView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
/**
 * Écran principal de l'application.
 * Affiche le calendrier de planning et permet de consulter les rendez-vous enregistrés.
 */
public class MainActivity extends AppCompatActivity {
    CalendarView calPlanning;
    String datePlanning;
    Base_donnee db;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {

```

```

        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });
    db = new Base_donnee(this);
    calPlanning = findViewById(R.id.calendar_planning);
    // Initialisation par défaut avec la date du jour
    java.util.Calendar c = java.util.Calendar.getInstance();
    datePlanning = c.get(java.util.Calendar.DAY_OF_MONTH) + "/" +
(c.get(java.util.Calendar.MONTH) + 1) + "/" + c.get(java.util.Calendar.YEAR);

    // Ecouteur du calendrier
    calPlanning.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {
        @Override
        public void onSelectedDayChange(CalendarView view, int year, int month, int
dayOfMonth) {
            datePlanning = dayOfMonth + "/" + (month + 1) + "/" + year;
        }
    });
}
/**
 * Lance l'activité de création d'un nouveau professionnel.
 * Cette méthode est appelée lors du clic sur le bouton correspondant dans l'interface.
 *
 * @param view Le bouton "Ajouter un professionnel" sur lequel l'utilisateur a cliqué.
 */
public void professionnel(View view) {
    Intent intentprofessionnel = new Intent (this, creation_professionnel.class);
    startActivity(intentprofessionnel);
}

/**
 * Lance l'activité de prise de rendez-vous.
 * Permet à l'utilisateur d'accéder à l'écran de recherche et de sélection de date/heure.
 *
 * @param view Le bouton "Prendre RDV" sur lequel l'utilisateur a cliqué.
 */
public void rdv(View view) {
    Intent intentrdv = new Intent (this, rdv.class);
    startActivity(intentrdv);
}

// La méthode appelée par le bouton "Afficher les rendez-vous"
/**
 * Interroge la base de données pour la date sélectionnée sur le calendrier.
 * Affiche les détails des RDV dans une boîte de dialogue (AlertDialog).
 * @param view Bouton Afficher Planning.
 */
public void planning(View view) {
    Cursor cursor = db.getRdvParDate(datePlanning);

```

```

if (cursor.getCount() == 0) {
    StringBuilder message_non_rdv = new StringBuilder("RDV du " + datePlanning + " :\n\n");
    message_non_rdv.append("Aucun RDV prévu pour le " + datePlanning + "\n");
    new android.app.AlertDialog.Builder(this)
        .setTitle("Planning du jour")
        .setMessage(message_non_rdv.toString())
        .setPositiveButton("OK", null)
        .show();
    // Cas où il n'y a pas de rendez-vous
    Toast.makeText(this, "Aucun RDV prévu pour le " + datePlanning,
Toast.LENGTH_LONG).show();
} else {
    // Message pour tous les rendez-vous trouvés
    StringBuilder message = new StringBuilder("RDV du " + datePlanning + " :\n\n");

    while (cursor.moveToNext()) {
        String praticien = cursor.getString(1);
        String heure = cursor.getString(3);

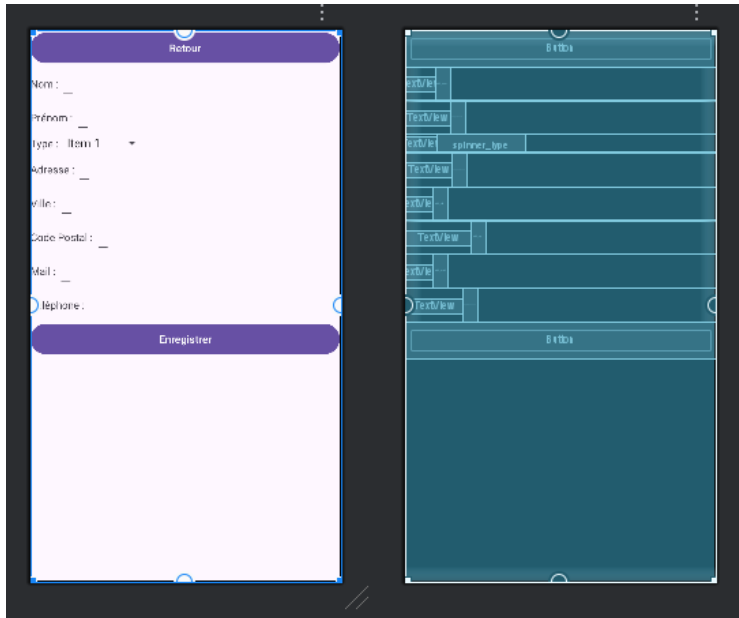
        String adresse = cursor.getString(8);
        String ville = cursor.getString(9);
        String cp = cursor.getString(10);
        String mail = cursor.getString(11);
        String tel = cursor.getString(12);

        message.append("🕒 ").append(heure).append(" - ").append(praticien).append("\n");
        message.append("📍 ").append(adresse).append(", ").append(ville).append("
").append(cp).append("\n");
        message.append("✉ ").append(mail).append("\n");
        message.append("☎ 0").append(tel).append("\n");
        message.append("-----\n\n");
    }

    // Affichage dans une boîte de dialogue
    new android.app.AlertDialog.Builder(this)
        .setTitle("Planning du jour")
        .setMessage(message.toString())
        .setPositiveButton("OK", null)
        .show();
    }
    cursor.close();
}
}

```

XML de la page de création d'un professionnel



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".creation_professionnel">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <Button
            android:id="@+id/retour"
            android:layout_width="match_parent"
            android:onClick="retour"
            android:layout_height="wrap_content"
            android:text="Retour"/>

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

            <TextView
                android:id="@+id/nom"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Nom : _" />

            <TextView
                android:id="@+id/prenom"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Prénom : _" />

            <TextView
                android:id="@+id/type"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Type : Item 1" />

            <TextView
                android:id="@+id/adresse"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Adresse : _" />

            <TextView
                android:id="@+id/email"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Email : _" />

            <TextView
                android:id="@+id/telephone"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Téléphone : _" />

            <Button
                android:id="@+id/enregistrer"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Enregistrer" />

        </LinearLayout>

    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

```

        android:text="Nom : "/>
<EditText
    android:id="@+id/edit_nom"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>

</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/prenom"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Prénom : "/>
    <EditText
        android:id="@+id/edit_prenom"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/type"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Type : "/>
    <Spinner
        android:id="@+id/spinner_type"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/adresse"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Adresse : "/>
    <EditText
        android:id="@+id/edit_adresse"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
</LinearLayout>
<LinearLayout

```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/ville"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Ville : "/>
    <EditText
        android:id="@+id/edit_ville"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/cp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Code Postal : "/>
    <EditText
        android:id="@+id/edit_cp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/mail"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Mail : "/>
    <EditText
        android:id="@+id/edit_mail"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/telephone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Téléphone : "/>
```

```

        <EditText
            android:id="@+id/edit_telephone"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:inputType="phone"/>
    </LinearLayout>
    <Button
        android:id="@+id/enregistrer"
        android:layout_width="match_parent"
        android:onClick="enregistrer"
        android:layout_height="wrap_content"
        android:text="Enregistrer"/>
</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

Code java de la page de création d'un professionnel

```

package com.example.projet_final_android;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
/**
 * Activité gérant l'interface de création d'un nouveau profil professionnel.
 * Permet la saisie des informations et leur enregistrement dans la base de données.
 *
 * @author VotreNom
 * @version 1.0
 */
public class creation_professionnel extends AppCompatActivity {
    /** Instance de gestion de la base de données SQLite */
    Base_donnee db;

    /** Champs de saisie pour les informations du professionnel */
    EditText nomInput, prenomInput, adresseInput, villeInput, cplInput, mailInput, telephoneInput;

    /** Menu déroulant pour la sélection du métier */
    Spinner type_select;

```

```

/** Liste des métiers disponibles dans le Spinner */
String [] type = {"Pharmacien", "Médecin Généraliste", "Ophtalmologue", "Gynécologue",
"Dentiste"};

/** Index du métier sélectionné dans le Spinner */
int idSelect;

/**
 * Initialise l'interface, lie les composants XML au code Java et configure le Spinner.
 * @param savedInstanceState État précédent de l'activité.
 */
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_creation_professionnel);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });

    db = new Base_donnee(this);
    db.getWritableDatabase();

    //lien interface / base de donnée
    nomInput=(EditText) findViewById(R.id.edit_nom);
    prenomInput=(EditText) findViewById(R.id.edit_prenom);
    type_select=(Spinner) findViewById(R.id.spinner_type);
    adresseInput=(EditText) findViewById(R.id.edit_adresse);
    villeInput=(EditText) findViewById(R.id.edit_ville);
    cpInput=(EditText) findViewById(R.id.edit_cp);
    mailInput=(EditText) findViewById(R.id.edit_mail);
    telephoneInput=(EditText) findViewById(R.id.edit_telephone);

    ArrayAdapter<String> Type = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, type);
    Type.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    type_select.setAdapter(Type);

    type_select.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
            idSelect = position;
        }
    })

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

```

```

    }
    });
}

/**
 * Récupère les données saisies par l'utilisateur et les insère dans la table Professionnel.
 * Affiche un Toast de confirmation et ferme l'activité en cas de succès.
 * @param view Le bouton sur lequel l'utilisateur a cliqué.
 */
public void enregistrer(View view) {
    try {
        db.insertDataProfessionnel(
            nomInput.getText().toString(),
            prenomInput.getText().toString(),
            type[idSelect],
            adresseInput.getText().toString(),
            villeInput.getText().toString(),
            Integer.parseInt(cplInput.getText().toString()),
            mailInput.getText().toString(),
            Integer.parseInt(telephoneInput.getText().toString())
        );

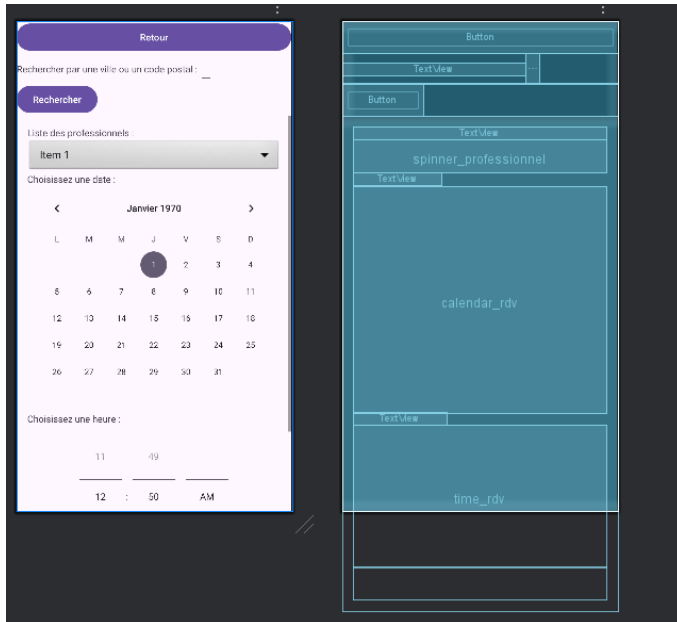
        // Affiche un message de succès
        Toast.makeText(this, "Enregistré avec succès !", Toast.LENGTH_SHORT).show();

    } catch (Exception e) {
        // Affiche l'erreur si ça plante (ex: champ vide)
        Toast.makeText(this, "Erreur : " + e.getMessage(), Toast.LENGTH_LONG).show();
        e.printStackTrace();
    }
    this.finish();
}

/**
 * Ferme l'activité actuelle sans enregistrer pour retourner à l'écran précédent.
 * @param view Le bouton de retour.
 */
public void retour(View view) {
    this.finish();
}
}
}

```

XML de la page des prises de rdv



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".rdv">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <Button
            android:id="@+id/retour"
            android:layout_width="match_parent"
            android:onClick="retour"
            android:layout_height="wrap_content"
            android:text="Retour"/>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <TextView
                android:id="@+id/recherche_ville_cp"
                android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:text="Rechercher par une ville ou un code postal : "/>
<EditText
    android:id="@+id/edit_ville_cp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <Button
        android:id="@+id/rechercher"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="rechercher"
        android:text="Rechercher"/>
</LinearLayout>
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="16dp">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Liste des professionnels :"/>
        <Spinner
            android:id="@+id/spinner_professionnel"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:minHeight="50dp"
            android:background="@android:drawable/btn_dropdown" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Choisissez une date : " />

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

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



```

CalendarView calendrier;
String curDate; // Simple déclaration de la chaîne de caractères curDate;
TimePicker heure_choisi; // <--- Ajoutez cette ligne
Base_donnee db;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_rdv);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });

    java.util.Calendar c = java.util.Calendar.getInstance();
    curDate = c.get(java.util.Calendar.DAY_OF_MONTH) + "/" + (c.get(java.util.Calendar.MONTH)
+ 1) + "/" + c.get(java.util.Calendar.YEAR);
    calendrier= findViewById(R.id.calendar_rdv);

    calendrier.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {
        @Override
        public void onSelectedDayChange(CalendarView view, int year, int month, int
dayOfMonth) {
            java.util.Calendar c = java.util.Calendar.getInstance(); //permet de mettre la date du jour
actuelle
            curDate = dayOfMonth + "/" + (month + 1) + "/" + year;

            //Afficher un message pour confirmer
            Toast.makeText(rdv.this, "Date choisie : " + curDate, Toast.LENGTH_SHORT).show();
        }
    });
    // Relier le TimePicker de l'interface vers le code
    heure_choisi = findViewById(R.id.time_rdv);
    heure_choisi.setIs24HourView(true); //format 24h

    editRecherche = findViewById(R.id.edit_ville_cp);
    spinnerPro = findViewById(R.id.spinner_professionnel);
    db = new Base_donnee(this);
}

/**
 * Effectue une recherche dans la base de données et met à jour le Spinner
 * avec les résultats trouvés (pour le nom et la profession des praticiens .
 * @param view Bouton rechercher.
 */
// La méthode appelée par le bouton "Rechercher"
public void rechercher(View view) {
    String saisie = editRecherche.getText().toString();

```

```

Cursor cursor = db.chercherProfessionnel(saisie);

if (cursor != null && cursor.getCount() > 0) {
    ArrayList<String> resultats = new ArrayList<>();

    // On parcourt le curseur pour récupérer les noms et prénoms
    while (cursor.moveToNext()) {
        String ligne = cursor.getString(1) + " " + cursor.getString(2) + " (" + cursor.getString(3) + ")";
        resultats.add(ligne);
    }

    // On crée l'adaptateur et on l'applique au spinner
    ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
        android.R.layout.simple_spinner_item, resultats);
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spinnerPro.setAdapter(adapter);

} else {
    // Si le curseur est vide, on vide le spinner et on prévient l'utilisateur
    spinnerPro.setAdapter(null);
    spinnerPro.setVisibility(View.VISIBLE);
    Toast.makeText(this, "Aucun professionnel trouvé", Toast.LENGTH_SHORT).show();
}
}
/**
 * Enregistre le rendez-vous finalisé dans la table rdv.
 * Récupère le choix du professionnel, la date du calendrier et l'heure du TimePicker.
 * @param view Bouton valider le RDV.
 */
public void enregistrerRDV(View view) {
    //Récupérer le nom du praticien dans le Spinner
    if (spinnerPro.getSelectedItem() == null) {
        Toast.makeText(this, "Veuillez d'abord choisir un professionnel",
            Toast.LENGTH_SHORT).show();
        return;
    }
    String praticienSelectionne = spinnerPro.getSelectedItem().toString();

    //Récupérer l'heure
    int heureSelect = heure_choisi.getHour();
    int minuteSelect = heure_choisi.getMinute();
    String heureFormat = String.format(Locale.getDefault(), "%02d:%02d", heureSelect,
        minuteSelect);

    //Envoyer à la base de données
    // On utilise la variable 'curDate' qui a été mise à jour par le calendrier
    db.insertDatardv(praticienSelectionne, curDate, heureFormat);

    //Confirmation du rendez-vous et retour à la page d'accueil

```

```
        Toast.makeText(this, "Rendez-vous enregistré pour le " + curDate + " à " + heureFormat,
Toast.LENGTH_LONG).show();
        this.finish(); // Retour à l'accueil
    }

    /**
     * Ferme l'activité actuelle sans enregistrer pour retourner à l'écran précédent.
     * @param view Le bouton de retour.
     */
    public void retour(View view) {
        this.finish();
    }
}
```

Code java de la page de la base de données

```
package com.example.projet_final_android;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.widget.AdapterView;

/**
 * Classe de gestion de la base de données SQLite (SQLiteOpenHelper).
 * Gère la création des tables, les mises à jour de structure et les opérations CRUD.
 */
public class Base_donnee extends android.database.sqlite.SQLiteOpenHelper{
    //Création de la base de données

    /** Nom de la base de données */
    public static final String DATABASE_NAME = "Bd_medical.db";
    /**
     * Construction des tables Professionnel et rdv avec le nom des colonne pour chaque tables.
     */

    //Création des colonnes de la table professionnel:
    public static final String TABLE_professionnel = "Professionnel";
    public static final String COL_1 = "id";
    public static final String COL_2 = "nom";
    public static final String COL_3 = "prenom";
    public static final String COL_4 = "type";
    public static final String COL_5 = "adresse";
    public static final String COL_6 = "ville";
    public static final String COL_7 = "cp";
    public static final String COL_8 = "mail";
    public static final String COL_9 = "telephone";

    // Création de la table rdv
    public static final String TABLE_rdv = "rdv";
    public static final String COLrdv_1 = "id";
    public static final String COLrdv_2 = "praticien";
    public static final String COLrdv_3 = "date";
    public static final String COLrdv_4 = "heure";
    public static final int DATABASE_VERSION = 3;

    //le Constructeur
    public Base_donnee(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    //Le onCreate permet de créer les colonnes de la table professionnel et rdv
    /**
     * Crée les tables Professionnel et rdv lors du premier lancement de l'application.
     */
}
```

```

    * @param db L'instance de la base de données.
    */
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table " + TABLE_professionnel + "(id integer primary key autoincrement,
nom text, prenom text, type text, adresse text, ville text, cp integer, mail text, telephone
integer)");
        db.execSQL("create table " + TABLE_rdv + "(id integer primary key autoincrement, praticien
text, date text, heure text)");

    }

    /**
     * Met à jour la structure de la base de données lors d'une mise à jour de la version.
     * recherche si la table existe et la supprime
     * @param db L'instance de la base de données.
     * recrée la table professionnel et rdv avec les nouvelles colonnesx
     */
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        //permet de mettre à jour la table en supprimant les anciennes données et en mettant les
nouvelles
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_professionnel);
        //Mettre ici le 2ème drop (de la table RDV)
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_rdv);
        onCreate(db);
    }

    /**
     * Insère un nouveau professionnel de santé dans la table correspondante.
     * @param nom Nom du praticien.
     * @param prenom Prénom du praticien.
     * @param type Métier sélectionné.
     * @param adresse Adresse postale.
     * @param ville Ville.
     * @param cp Code Postal.
     * @param mail Adresse mail.
     * @param telephone Numéro de téléphone.
     */
    //insert les données dans les colonnes de la table professionnel
    public void insertDataProfessionnel(String nom, String prenom, String type, String adresse,
String ville, int cp,String mail, int telephone) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_2, nom);
        contentValues.put(COL_3, prenom);
        contentValues.put(COL_4, type);
        contentValues.put(COL_5, adresse);
        contentValues.put(COL_6, ville);
        contentValues.put(COL_7, cp);
    }

```

```

        contentValues.put(COL_8, mail);
        contentValues.put(COL_9, telephone);
        db.insert(TABLE_professionnel, null, contentValues);
        db.close();
    }
    //insère les données dans les colonnes de la table RDV
    /**
     * Insère un nouveau professionnel de santé dans la table correspondante.
     * @param praticien Nom du praticien.
     * @param date Date du RDV.
     * @param heure Heure du RDV.
     */
    public void insertDatardv(String praticien, String date, String heure) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COLrdv_2, praticien);
        contentValues.put(COLrdv_3, date);
        contentValues.put(COLrdv_4, heure);
        db.insert(TABLE_rdv, null, contentValues);
        db.close();
    }

    //requête SQL pour trouver les professionnelles qui on le code postale ou la ville
    correspondant à la recherche
    /**
     * Recherche les professionnels en fonction d'un Code Postal ou d'une Ville.
     * @param recherche La chaîne saisie par l'utilisateur.
     * @return Un Cursor contenant les colonnes id, nom, prenom et type du / des professionnel
    trouver avec le Code Postale.
     */
    public Cursor chercherProfessionnel(String recherche) {
        SQLiteDatabase db = this.getReadableDatabase();
        // On cherche si le CP est égal OU si la VILLE ressemble à la recherche
        String query = "SELECT id as _id, nom, prenom, type FROM " + TABLE_professionnel + "
WHERE cp = ? OR ville LIKE ?";
        return db.rawQuery(query, new String[]{recherche, "%" + recherche + "%"});
    }

    // Récupère tous les professionnels de la table
    /**
     * Récupère l'intégralité des professionnels enregistrés dans la table.
     * Cette méthode est créée pour les teste unitaires
     * @return Un Cursor contenant toutes les lignes et colonnes de la table Professionnel.
     */
    public Cursor getProfessionnel() {
        SQLiteDatabase db = this.getWritableDatabase();
        return db.rawQuery("SELECT * FROM " + TABLE_professionnel, null);
    }
}

```

```

/**
 * Récupère l'intégralité des rendez-vous enregistrés dans la table rdv.
 * Cette méthode est créée pour les teste unitaires
 * @return Un Cursor contenant toutes les lignes et colonnes de la table rdv.
 */
public Cursor getRDV() {
    SQLiteDatabase db = this.getWritableDatabase();
    return db.rawQuery("SELECT * FROM " + TABLE_rdv, null);
}

/**
 * Récupère les rendez-vous d'une date précise en effectuant une jointure (JOIN)
 * avec la table professionnel pour obtenir les coordonnées complètes.
 * @param dateChoisie Date au format JJ/MM/AAAA.
 * @return Cursor contenant les détails du RDV et du professionnel.
 */
public Cursor getRdvParDate(String dateChoisie) {
    SQLiteDatabase db = this.getReadableDatabase();
    String querySimplifree = "SELECT * FROM " + TABLE_rdv +
        " INNER JOIN " + TABLE_professionnel +
        " ON " + TABLE_rdv + "." + COLrdv_2 + " LIKE '%" + TABLE_professionnel + "." + COL_2 + "
|| '%" +
        " WHERE " + TABLE_rdv + "." + COLrdv_3 + " = ?";

    return db.rawQuery(querySimplifree, new String[]{dateChoisie});
}
}

```