

PROJET GSB

Android Studio

2020/2021

Sylvain Caron



Sommaire

- I. Présentation du Projet
- II. Structure de la BD et MCD
- III. XML des écrans
- IV. Classe BD
- V. Code Java des écrans

I - Présentation du Projet

L'objectif de ce projet est la réalisation d'une application sous Android Studio qui permettra aux professionnels de chez GSB d'ajouter des professionnels dans une liste, de prendre des rendez-vous avec ceux sélectionné et de voir le planning de chaque jour avec les horaires et les professionnels à aller voir.

Pour cela 4 interfaces principales seront réalisé, l'ajout d'un Pro, l'ajout d'un rendez-vous et le planning plus celle du menu principal.

Voir page 5 pour retrouver les designs des différentes interfaces.

II - Structure de la BD

La structure de cette BD est assez simple, une table « Professionnel » pour contenir les informations du professionnel tel que :

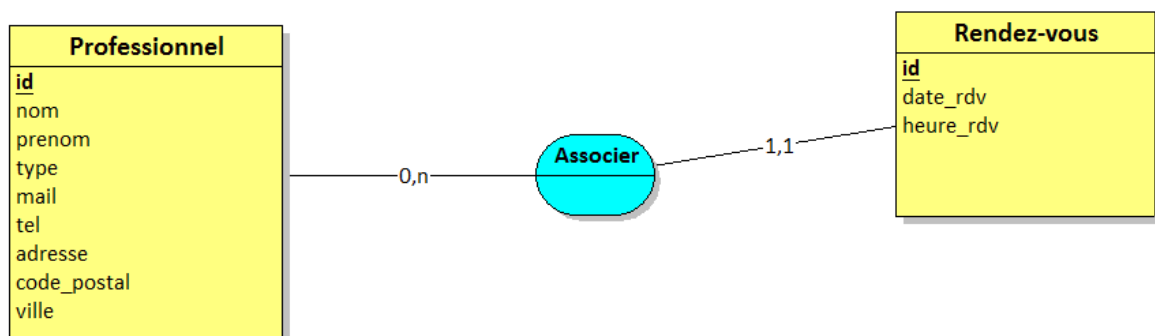
- Un ID
- Le nom
- Le prénom
- Le type
- L'adresse
- La ville
- Le code postal
- Le mail
- Le Téléphone

Et d'autres informations ...

Elle contient aussi une table « Rendez-vous » qui permettra de stocker les informations du rendez-vous tel que :

- Un ID
- La date
- L'heure

Pour cela un MCD (Modèle Conceptuel de Donnée) à été réalisé, en voici le schéma :



III - XML des écrans

Ce projet possède 4 interfaces :

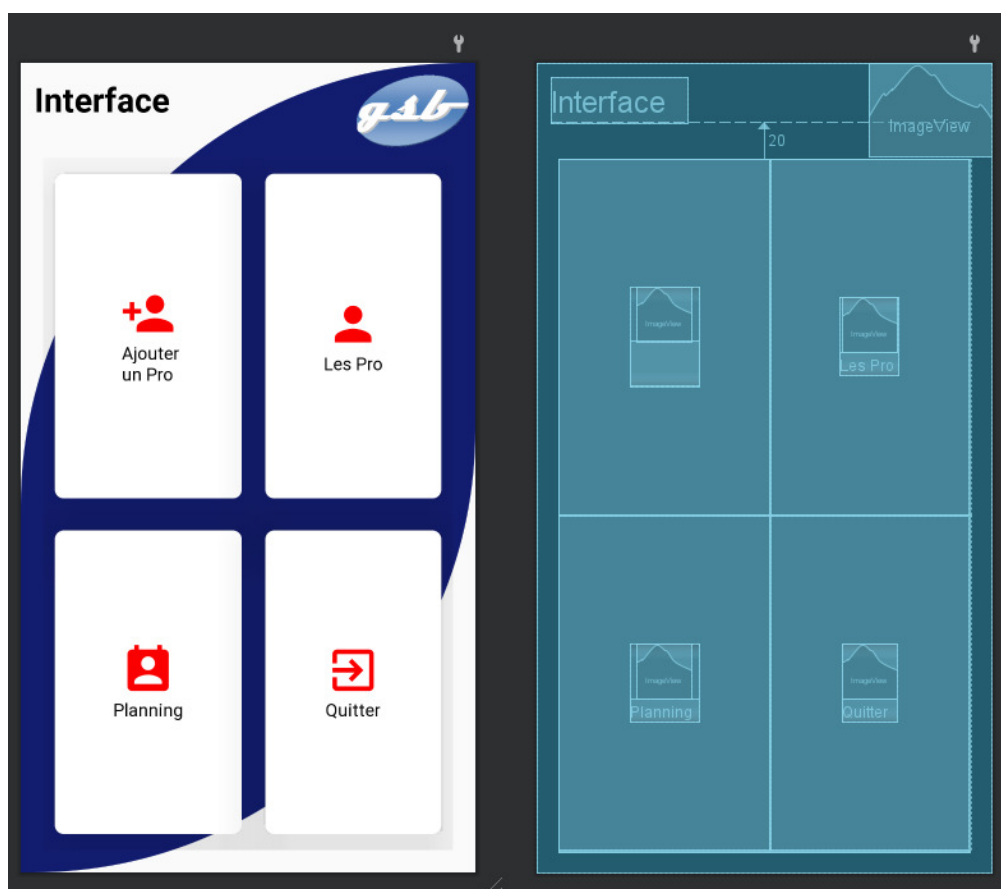
- Menu
- Ajout d'un professionnel
- Ajout d'un Rendez-vous
- Planning

En voici une représentation visuelle de chacun

Le Menu :

Comporte 4 Boutons pour :

- Aller vers l'interface « Ajouter un Pro »
- Aller vers l'interface « Les Professionnels »
- Aller vers l'interface « Planning »
- Quitter l'application



Ajout Professionnel :

Comporte :

- Des EditText pour remplir les coordonnées du Professionnel
- Un bouton « Ajouter un Professionnel »
- Un bouton « Retour vers le Menu »

The image displays two versions of a mobile application form titled "Ajout d'un Professionnel".

Left Version (Clean Design):

- Title:** Ajout d'un Professionnel
- Fields:** Nom, Prénom, Type (dropdown menu with "Généraliste" selected), Adresse, Ville, Code Postal, Mail, Tél.
- Button:** A red button with white text labeled "AJOUT DU PROFESSIONNEL".

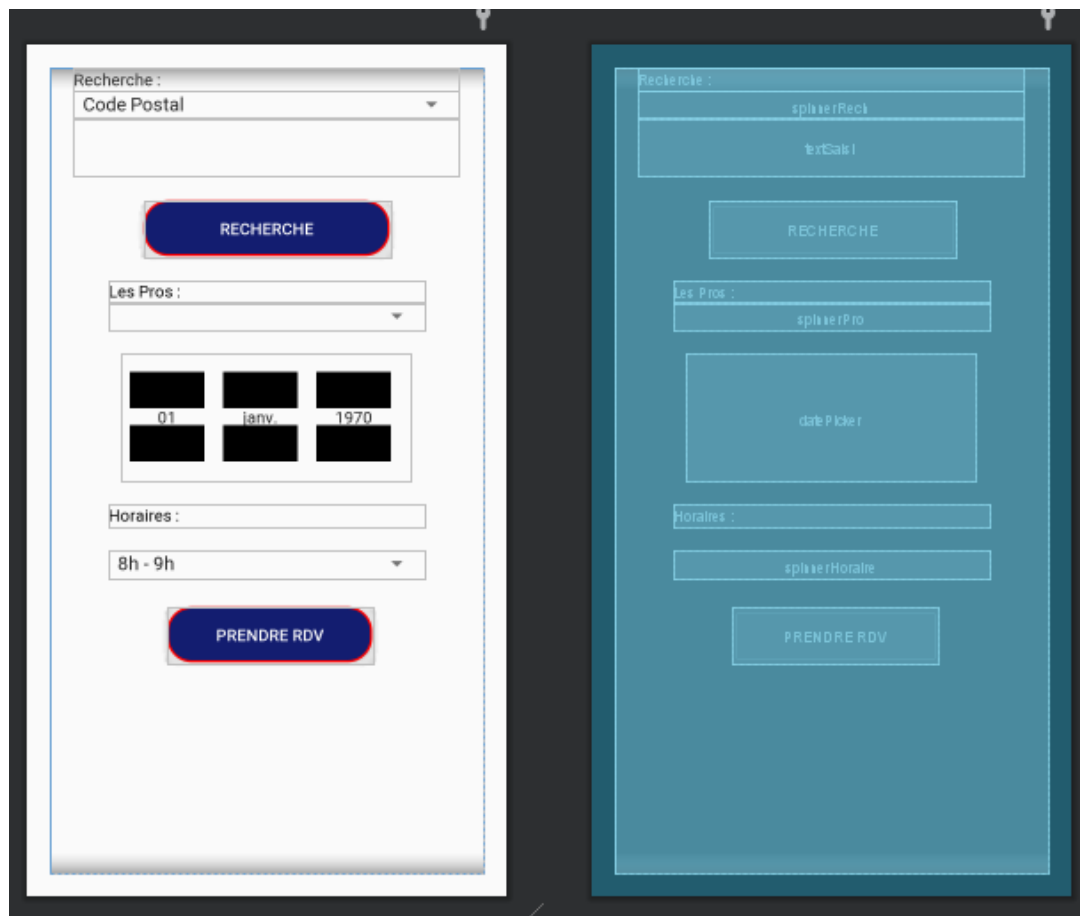
Right Version (Wireframe):

- Title:** Ajout d'un Professionnel
- Fields:** textNom, textPrenom, Type (spinnerType), textAdresse, textVille, textCodePostal, textMail, textTel.
- Button:** A light blue button with dark blue text labeled "AJOUT DU PROFESSIONNEL".

Ajout d'un rendez-vous :

Comporte :

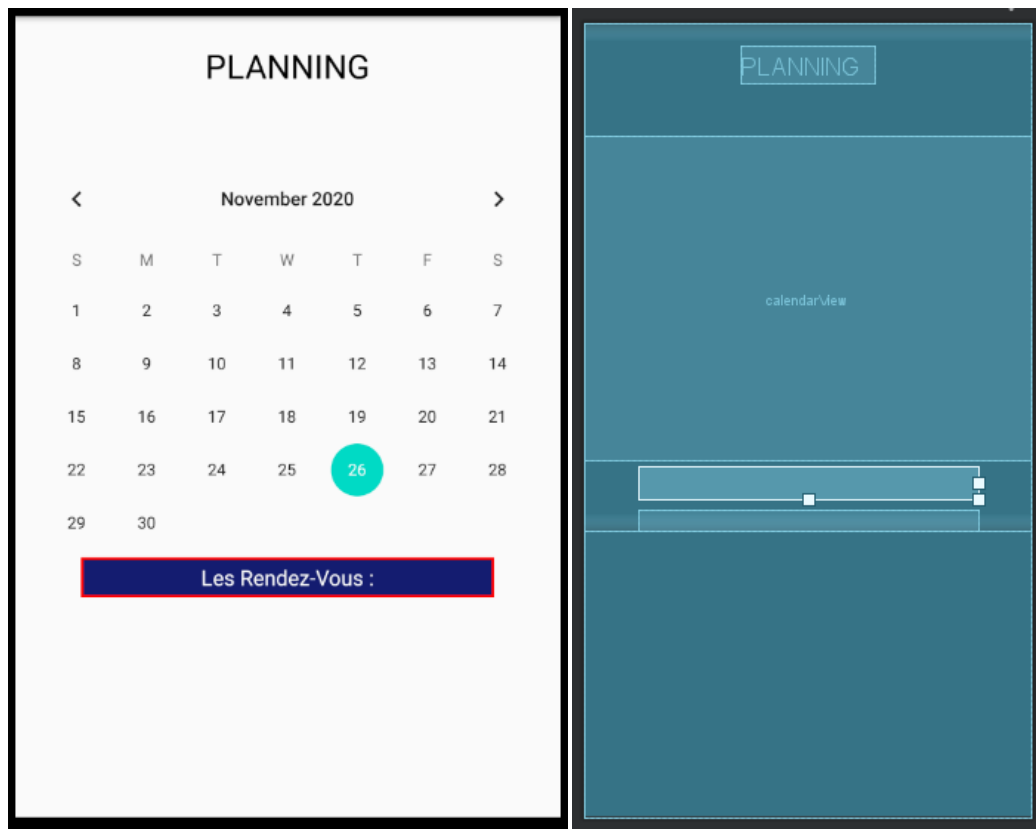
- Une ComboBox (Spinner) pour sélectionner la recherche voulue
- Un EditText pour saisir l'information de la recherche (Ville / Code Postal)
- Un bouton « Recherche » qui changera les Professionnels affichés
- Un bouton « Prendre rendez-vous » qui récupère toutes les informations du Rendez-vous tel que : l'horaire, la date et le professionnel



Le Planning :

Comporte un bouton « Retour vers le menu »

En fonction de la date sélectionner les rendez-vous s'affiche dynamiquement en dessous de « Les Rendez-Vous »



IV – La classe BD

Voici les variables ainsi que les entêtes des fonctions de la classe BD avec leur utilité :

```
public class SQLiteDatabaseGSB extends SQLiteOpenHelper
{
    //Déclaration des variables de la BD
    public static final String DATABASE_NAME = "gsb.db";

    public static final String TABLE_NAME_1 = "pro_table";
    public static final String COL_1_PRO = "ID";
    public static final String COL_2_PRO = "NOM";
    public static final String COL_3_PRO = "PRENOM";
    public static final String COL_4_PRO = "TYPE";
    public static final String COL_5_PRO = "ADRESSE";
    public static final String COL_6_PRO = "VILLE";
    public static final String COL_7_PRO = "CODE_POSTAL";
    public static final String COL_8_PRO = "MAIL";
    public static final String COL_9_PRO = "TEL";

    public static final String TABLE_NAME_2 = "rdv_table";
    public static final String COL_1_RDV = "ID";
    public static final String COL_2_RDV = "HEURE";
    public static final String COL_3_RDV = "DATE";
    public static final String COL_4_RDV = "LE_PRO";

    public SQLiteDatabaseGSB(Context context)
    {
        super(context, DATABASE_NAME, null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db)
    {
        db.execSQL("CREATE table " + TABLE_NAME_1 + "(ID INTEGER PRIMARY KEY AUTOINCREMENT, NOM TEXT, PRENOM TEXT, TYPE TEXT, ADRESSE TEXT, VILLE TEXT, CODE_POSTAL TEXT, MAIL TEXT, TEL TEXT)");
        db.execSQL("CREATE table " + TABLE_NAME_2 + "(ID INTEGER PRIMARY KEY AUTOINCREMENT, HEURE TEXT, DATE TEXT, LE_PRO TEXT)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
    {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME_1);
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME_2);
        onCreate(db);
    }

    public void insertPro(String nom, String prenom, String type, String adresse, String ville, String code_postal, String mail, String tel) // Ajout Pro
```

```
public void insertRDV(String heure, String laDate, String lePro) //Ajout RDV
public Cursor getDataPro() //Obtenir tout les Pro
public Cursor getDataRDV() //Obtenir tout les RDV
public void deletePro(int id) //Supprimer un Pro
public void deleteRDV(int id) //Supprimer un RDV
public Cursor rechCodePostal(String pCodePostal) //Recherche par Code Postal
public Cursor rechVille(String pVille) //Recherche par Ville
public Cursor getRDV(String pDate) //Obtenir les RDV en fonction d'une date
}
```

V – Le code Java des Interfaces

(Des commentaires dans le code expliquent l'utilité des fonctions ou de certaines lignes spécifiques)

Voici le code de l'interface « **Menu** » :

```
public class MainActivity extends AppCompatActivity {

    Globals g;

    SQLiteDatabaseGSB db;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //Création de La BD
        db = new SQLiteDatabaseGSB(this);

        //addData(this);

    }

    //Clic boutons
    public void clicVoirPro(View view)
    {
        //Affichage d'une PopUp + Vue "les Pro"
        Toast.makeText(this, getString(R.string.voirPro),
        Toast.LENGTH_LONG).show();
        Intent intentPro = new Intent(this, viewPro.class); //Lancer l'activité
        DisplayVue
        startActivity(intentPro);
    }
    public void clicAddPro(View view)
    {
        //Affichage d'une PopUp + Vue "Ajoute Pro"
        Toast.makeText(this, getString(R.string.addPro),
        Toast.LENGTH_LONG).show();
        Intent intentAddPro = new Intent(this, addPro.class); //Lancer l'activité
        DisplayVue
        startActivity(intentAddPro);
    }
    public void clicPlanning(View view)
    {
        //Afficher PopUp + Vue "Planning"
        Toast.makeText(this, getString(R.string.planning),
        Toast.LENGTH_LONG).show();
        Intent intentPlanning = new Intent(this, viewPlanning.class); //Lancer
        l'activité DisplayVue
        startActivity(intentPlanning);
    }

    public void clicQuitter(View view)
    {
        finish();
    }
}
```

Voici le code de l'interface « **Ajout Professionnel** » :

```
public class addPro extends AppCompatActivity
{
    SQLiteDatabaseGSB db;

    //Variable de Récupération sur Interface
    EditText nomInput, prenomInput, adresseInput, villeInput, codePostalInput,
    mailInput, telInput;
    EditText idInput;
    EditText newAdresse;
    Spinner typeInput;

    TextView textListe;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_pro);

        db = new SQLiteDatabaseGSB(this);

        //INFO -- Affichage des Pro
        //majListePro();
    }

    public void clicAjoutPro(View view)
    {
        //Récupère toute les données dans des variables
        nomInput = (EditText) findViewById(R.id.textNom);
        prenomInput = (EditText) findViewById(R.id.textPrenom);
        typeInput = (Spinner) findViewById(R.id.spinnerType);
        adresseInput = (EditText) findViewById(R.id.textAdresse);
        villeInput = (EditText) findViewById(R.id.textVille);
        codePostalInput = (EditText) findViewById(R.id.textCodePostal);
        mailInput = (EditText) findViewById(R.id.textMail);
        telInput = (EditText) findViewById(R.id.textTel);

        //Ajoute le Pro
        db.insertPro(nomInput.getText().toString(),
        prenomInput.getText().toString(), typeInput.getSelectedItem().toString(),
        adresseInput.getText().toString(),
        villeInput.getText().toString(),
        codePostalInput.getText().toString(), mailInput.getText().toString(),
        telInput.getText().toString());
        db.close();
        Toast.makeText(this, "Ajout du Professionnel ...",
        Toast.LENGTH_SHORT).show();
        //INFO -- Affichage des Pro
        //majListePro();
    }

    public void majListePro()
    {
        try
        {
            //Toast.makeText(this, "Dans le Try", Toast.LENGTH_LONG).show();
        }
    }
}
```

```

        //Cursor data = db.getRDV("24/10/2020");
        //Cursor data = db.getDataRDV();
        Cursor data = db.getDataPro();
        String texte = "";
        while(data.moveToNext())
        {
            texte = texte + String.valueOf(" ID: " + data.getInt(0) + " Nom : " + data.getString(1) + " Ville : " + data.getString(5));
            //texte = texte + String.valueOf(" ID: " + data.getInt(0) + "
Heure : " + data.getString(1) + " Date : " + data.getString(2) + " LePro : " + data.getString(3));
            //texte = texte + String.valueOf(" Heure" + data.getString(0) + "
Le Pro : " + data.getString(1));
        }
        textListe.setText(texte);
    }
    catch (Exception e)
    {
        textListe.setText(e.getMessage());
    }
}

```

Voici le code de l'interface « **Voir les Pros** » :

```
public class viewPro extends AppCompatActivity {

    Globals g;

    SQLiteDatabaseGSB db;
    DatePicker laDate;
    Spinner spinnerPro, spinnerRech, horaire, lePro;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_pro);

        //Déclaration de la BD
        db = new SQLiteDatabaseGSB(this);

        spinnerPro = (Spinner) findViewById(R.id.spinnerPro);
        spinnerRech = (Spinner) findViewById(R.id.spinnerRech);

        affichePro();
    }

    //Inutile sert d'aide à afficher
    public void clicEnregistrerDonnee(View view)
    {
        EditText text = findViewById(R.id.textSaisi);
        String saisi;
        saisi = text.getText().toString();

        g = Globals.getInstance();
        g.setData(Integer.parseInt(saisi));
    }

    public void clicRecherche(View view)
    {
        EditText recherche = (EditText) findViewById(R.id.textSaisi);

        //Nettoie le spinner
        spinnerPro.setAdapter(null);

        //Afficher les résultats en fonction de la recherche voulu (Code Postal /
Ville)
        if(spinnerRech.getSelectedItem().equals("Code Postal"))
        {
            Cursor resultatCode =
db.rechCodePostal(recherche.getText().toString());
            majPro(resultatCode);
        }
        else if(spinnerRech.getSelectedItem().equals("Ville"))
        {
            Cursor resultatVille = db.rechVille(recherche.getText().toString());
            majPro(resultatVille);
        }
    }

    public void affichePro()
    {

```

```

//Récupération du Resultat
Cursor lesPro = db.getDataPro();

//Création d'une Liste pour l'ajout dans le Spinner
ArrayList<String> spinnerArray = new ArrayList<String>();

//Parcours la Cursor(Liste) de Pro et les ajoutes dans le spinner
while(lesPro.moveToNext())
{
    spinnerArray.add(String.valueOf(lesPro.getString(1) + " " +
lesPro.getString(2) + " -- " + lesPro.getString(3) + " -- " +
lesPro.getString(5)));
    //int i = 0;
    //Log.d("Test :", spinnerArray.get(i));
    //i++;
}

//Création de l'adaptateur pour ajouter la Liste dans le Spinner
ArrayAdapter<String> adapter = new ArrayAdapter<String> (this,
android.R.layout.simple_spinner_dropdown_item, spinnerArray);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinnerPro.setAdapter(adapter);
}
public void majPro(Cursor pLaRecherche)
{
    ArrayList<String> spinnerArray = new ArrayList<String>();

    //Parcours le Cursor(Liste) de Pro et les ajoutes dans le spinner (en
fonctio nde la recherche)
    while(pLaRecherche.moveToNext())
    {
        spinnerArray.add(String.valueOf(pLaRecherche.getString(1) + " " +
pLaRecherche.getString(2) + " -- " + pLaRecherche.getString(3)));
        //int i = 0;
        //Log.i("Test :", spinnerArray.get(i));
        //i++;
    }

    //Création de l'adaptateur pour ajouter la Liste dans le Spinner
    ArrayAdapter<String> adapter = new ArrayAdapter<String> (this,
android.R.layout.simple_spinner_dropdown_item, spinnerArray);

    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spinnerPro.setAdapter(adapter);
}

public void clicRDV(View view)
{
    laDate = (DatePicker) findViewById(R.id.datePicker);
    int day = laDate.getDayOfMonth(); //Jour
    int month = laDate.getMonth() + 1; //Mois
    int year = laDate.getYear(); //Année

    horaire = (Spinner) findViewById(R.id.spinnerHoraire);
    lePro = (Spinner) findViewById(R.id.spinnerPro);

    //Crée la date en fonction des informations récupérer
    String date = String.valueOf(day + "/" + month + "/" + year);

```

```

        //INFO -- Aide Affichage
        /*
        Log.i("horaire","horaire");
        Log.i("horaire",horaire.getSelectedItem().toString());

        Log.i("date","date");
        Log.i("date",date);

        Log.i("lePro","lePro");
        Log.i("lePro",lePro.getSelectedItem().toString());
        */
        db.insertRDV(horaire.getSelectedItem().toString(), date,
lePro.getSelectedItem().toString());
        db.close();
        Toast.makeText(this, "Ajout du Rendez-vous ...",
Toast.LENGTH_SHORT).show();
    }
}

```

Voici le code de l'interface « **Planning** » :

```

public class viewPlanning extends AppCompatActivity {

    //Base de Donnée
    SQLiteDatabaseGSB db;

    //Variable Déclaration
    CalendarView calendar;
    SimpleDateFormat sdf;

    String laDate;
    TextView textRdv;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_planning);

        db = new SQLiteDatabaseGSB(this);

        textRdv = (TextView) findViewById(R.id.textRdv);
        calendar = (CalendarView) findViewById(R.id.calendarView);

        //Format de la Date Jour/Mois/Année
        sdf = new SimpleDateFormat("dd/MM/yyyy");
        laDate = sdf.format(new Date(calendar.getDate()));

        //Récupère la Date sur le Calendrier et la converti en date JJ/MM/AAAA
        calendar.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {
            @Override
            public void onSelectedDayChange(CalendarView view, int year, int
month, int dayOfMonth) {

                month++; //Commence à 0 donc on ajoute 1
            }
        });
    }
}

```



```

        laDate = String.valueOf(dayOfMonth+"/"+month+"/"+year);
        majListeRdv();
    }
});
}

public void afficheDate(View view)
{
    Toast.makeText(this, laDate, Toast.LENGTH_SHORT).show();
}

public void majListeRdv()
{
    try
    {
        //Récupère les RDV en fonction d'une date et les affiche à la date
sélectionnée
        Cursor data = db.getRDV(laDate);
        String texte = "";
        while(data.moveToNext())
        {
            texte = texte + String.valueOf("Horaire : " + data.getString(0) +
"\nProfessionnel : " + data.getString(1) + "\n\n");
        }
        if(texte == "")
        {
            textRdv.setText("Aucun Rendez-Vous aujourd'hui");
        }
        else
        {
            textRdv.setText(texte);
        }
    }
    catch (Exception e)
    {
        textRdv.setText(e.getMessage());
    }
}
}

```