

CHAMPIAU ANNAËLLE

DOCUMENTATION

Technique

ANDROID  
VisiteurDAO

ENTREPRISE VISIOTECH SOLUTIONS

# SOMMAIRE

**01. Contexte**

**02. Architecture**

**03. Base de données**

**04. Réalisations**

**05 Résultats**

# CONTEXTE

VisioTech Solutions est une startup spécialisée dans le développement d'outils numériques dédiés à la gestion des ressources humaines. Elle conçoit des solutions permettant d'optimiser le suivi du personnel, notamment dans les organisations faisant appel à des intervenants externes ou temporaires.

Dans ce cadre, l'entreprise utilise le terme « visiteurs » pour désigner ses collaborateurs non permanents (consultants, prestataires, employés en CDD, etc.). Ce choix reflète leur statut flexible et la nécessité de disposer d'un système de gestion adapté à leur mobilité et à leur suivi spécifique.

Afin d'améliorer la gestion interne de ces profils, VisioTech Solutions souhaite mettre en place une application mobile Android intitulée « Gestion de visiteurs ». Cette application a pour objectif de centraliser les informations relatives aux visiteurs et de permettre leur gestion de manière simple, rapide et sécurisée.

L'application repose sur une base de données locale SQLite, garantissant un fonctionnement totalement autonome, sans dépendance à une connexion réseau. Elle doit permettre la réalisation des opérations principales de gestion (CRUD), à savoir l'ajout, la consultation, la modification et la suppression des fiches visiteurs.

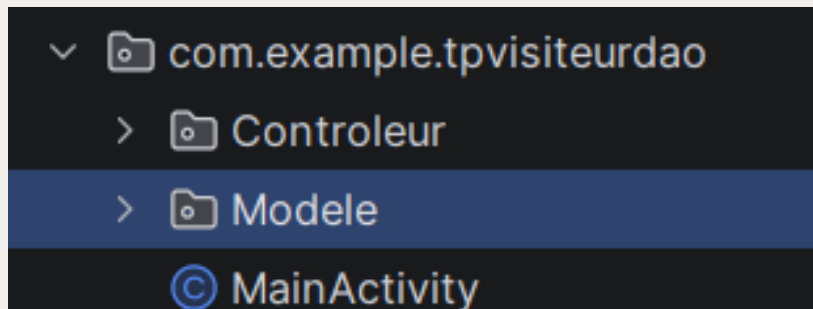
Par ailleurs, un système d'authentification est mis en place afin de sécuriser l'accès à l'application. Chaque utilisateur doit se connecter à l'aide d'un identifiant et d'un mot de passe. Un compte administrateur par défaut (admin/admin) est prévu et prioritaire lors de la vérification des identifiants, avant toute consultation de la base de données.

Dans une seconde phase, l'application est enrichie par un système de gestion des rôles, distinguant les administrateurs des utilisateurs classiques. Cette séparation permet de restreindre certaines fonctionnalités sensibles (ajout, modification, suppression) aux seuls administrateurs, tandis que les autres utilisateurs disposent d'un accès limité en consultation.

Enfin, une gestion de session persistante est implémentée grâce à l'utilisation de SharedPreferences. Cette fonctionnalité permet de conserver les informations de connexion et le rôle de l'utilisateur entre les différents lancements de l'application, améliorant ainsi l'expérience utilisateur tout en maintenant un niveau de sécurité adéquat. Une fonctionnalité de déconnexion sécurisée est également prévue, garantissant la suppression des données de session et empêchant tout accès non autorisé via le retour arrière.

# ARCHITECTURE

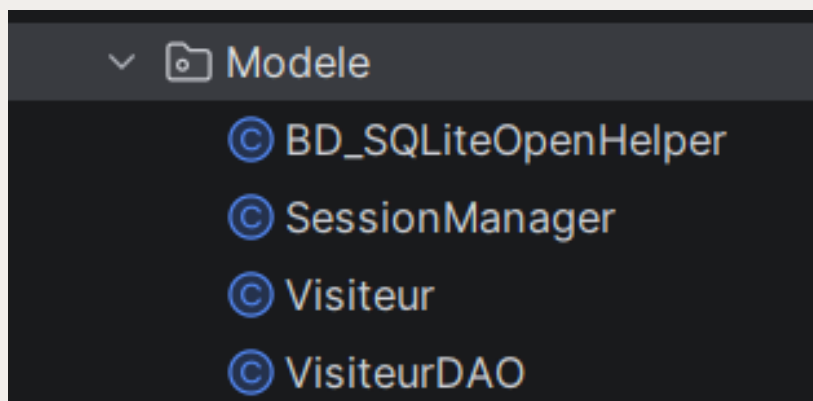
L'Architecture utilisé est une Architecture DAO



**Main Activity** : écran principal de connexion

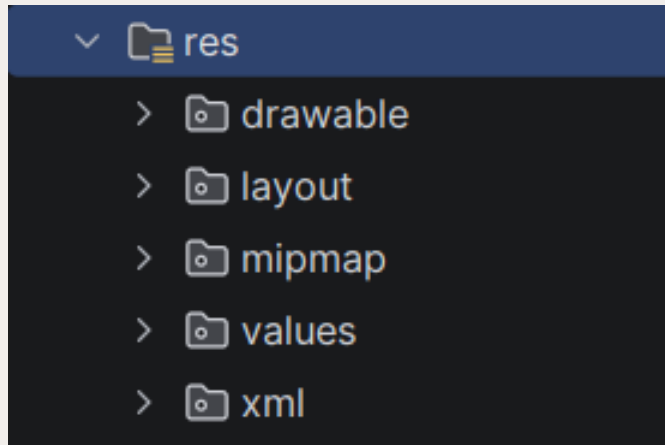


**Contrôleur** : gère les interactions utilisateur et la navigation.

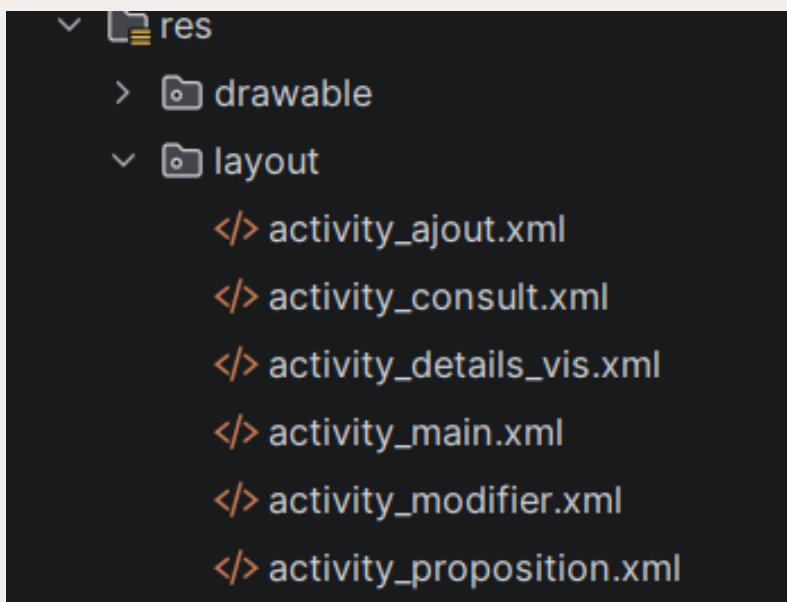


**Modèle DAO** : contient les classes liées aux données et à la base de données.

# ARCHITECTURE DAO



- **res** dossier ressource qui contient l'interfaces, images, textes, styles, ...
- **drawable** → images et formes
- **layout** → interfaces utilisateur
- **mipmap** → icône de l'application
- **values** → textes, couleurs, styles
- **xml** → fichiers de configuration

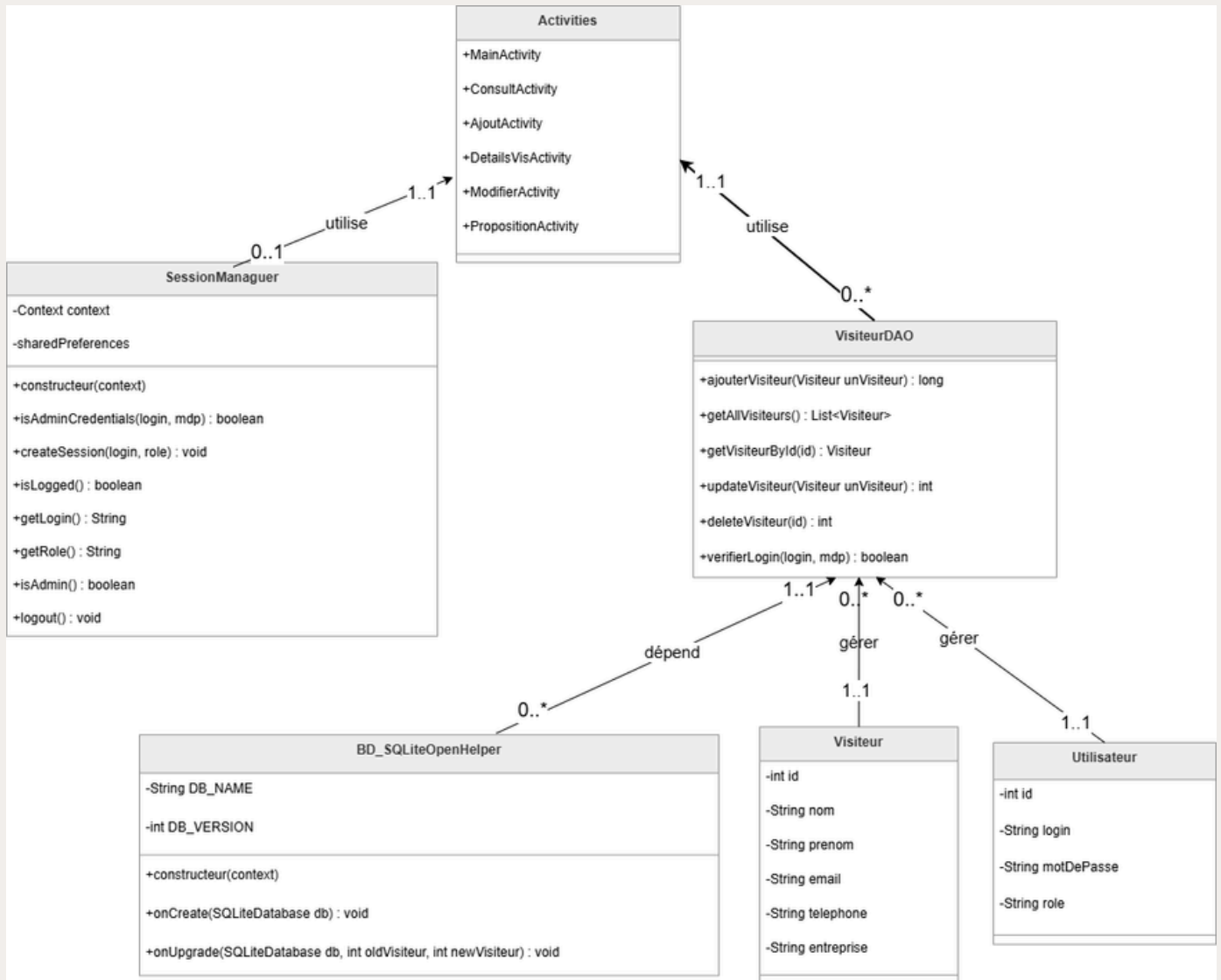


Vue layout Utilisateur / Administrateur

# BASE DE DONNÉES

## UML

### UML\_Android\_VisiteurDAO



# BASE DE DONNÉES

## SQL

```
package com.example.tpvisiteurdao.Modele;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

/**
 * Helper pour la gestion de la base de données SQLite.
 * Gère la création des tables et l'initialisation des données.
 */
public class BD_SQLiteOpenHelper extends SQLiteOpenHelper {

    // Requête SQL de création de la table visiteur
    private String visiteur =
        "CREATE TABLE IF NOT EXISTS visiteur (\n" +
        "  id text PRIMARY KEY NOT NULL,\n" +
        "  nom text DEFAULT NULL,\n" +
        "  prenom text DEFAULT NULL,\n" +
        "  login text DEFAULT NULL,\n" +
        "  mdp text DEFAULT NULL,\n" +
        "  adresse text DEFAULT NULL,\n" +
        "  cp text DEFAULT NULL,\n" +
        "  ville text DEFAULT NULL,\n" +
        "  dateEmbauche text DEFAULT NULL);";

    public BD_SQLiteOpenHelper(Context context, String name, SQLiteDatabase.CursorFactory factory,
int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
        // Méthode appelée lors d'un changement de version de la base
    }

    /**
     * Méthode appelée lors de la première création de la base.
     * Crée la table et insère un jeu de données de test.
     */
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(visiteur);
        // Insertion de données initiales pour le test du projet
        db.execSQL(
            "INSERT INTO visiteur (id, nom, prenom, login, mdp, adresse, cp, ville, dateEmbauche)
VALUES\n" +
            "('a131', 'Villichalane', 'Louis', 'aribiA', 'aaaa', '8 rue des textmes', '46000',
'Cahors', '2005-12-21'),\n" +
            "('a17', 'Andre', 'David', 'dandre', 'oppj5', '1 rue Petit', '46200', 'Lalbenque',
'1998-11-23'),\n" +
            "('a55', 'Bedos', 'Christian', 'cbedos', 'gmhxd', '1 rue Peranud', '46250',
'Montcuq', '1995-01-12'),\n" +
            "('a93', 'Tusseau', 'Louis', 'ltusseau', 'ktp3s', '22 rue des Ternes', '46123',
'Gramat', '2000-05-01'),\n" +
            "('b13', 'Bentot', 'Pascal', 'pbentot', 'doyw1', '11 allée des Cerises', '46512',
'Bessines', '1992-07-09'),\n" +
            "('b16', 'Bioret', 'Luc', 'lbioret', 'hrjfs', '1 Avenue gambetta', '46000', 'Cahors',
'1998-05-11'),\n" +
            "('b19', 'Bunisset', 'Francis', 'fbunisset', '4vbnd', '10 rue des Perles', '93100',
```

# RÉALISATIONS

```
public class AjoutActivity extends AppCompatActivity {
    // Boutons de l'interface
    private Button boutonValid;
    private Button boutonDeconnexion;

    // Managers pour la session et la base de données
    private SessionManager sessionManager;
    private VisiteurDAO unVisiteurDao;

    // Champs de texte du formulaire
    private EditText recupId, recupNom, recupPrenom, recupLogin, recupMdp,
        recupAdresse, recupCP, recupVille, recupDate;

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

        // Liaison des composants avec le fichier XML
        boutonValid=(Button) findViewById(R.id.buttonValiderAjout);
        recupId=(EditText) findViewById(R.id.editTextId);
        recupNom = (EditText) findViewById(R.id.editTextNom);
        recupPrenom = (EditText) findViewById(R.id.editTextPre);
        recupLogin=(EditText) findViewById(R.id.editTextLogin);
        recupMdp=(EditText) findViewById(R.id.editTextMdp);
        recupAdresse = (EditText) findViewById(R.id.editTextAdRue);
        recupCP = (EditText) findViewById(R.id.editTextCP);
        recupVille = (EditText) findViewById(R.id.editTextVille);
        recupDate = (EditText) findViewById(R.id.editTextDate);
        boutonDeconnexion = (Button) findViewById(R.id.buttonDeconnexion);

        // Initialisation des objets métier
        unVisiteurDao = new VisiteurDAO(this);
        sessionManager = new SessionManager(getApplicationContext());

        // Bouton de déconnexion
        boutonDeconnexion.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                deconnecter();
            }
        });

        // Validation et enregistrement du visiteur
        boutonValid.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // Lecture des données
                String resId = recupId.getText().toString();
                String resNom = recupNom.getText().toString();
                String resPrenom = recupPrenom.getText().toString();
                String resLogin = recupLogin.getText().toString();
                String resMdp = recupMdp.getText().toString();
                String resAdresse = recupAdresse.getText().toString();
                String resCP = recupCP.getText().toString();
                String resVille = recupVille.getText().toString();
                String resDate = recupDate.getText().toString();

                // Contrôle de surface (champs obligatoires)
                if(resId.isEmpty()|| resNom.isEmpty()||resPrenom.isEmpty()|| resLogin.isEmpty()||
```

# RÉALISATIONS

```
public class ConsultActivity extends AppCompatActivity {

    // Composants UI
    private ListView lstVisiteur;
    private Visiteur selectedVisiteur;
    private Button retour, boutonDeconnexion;
    private ArrayAdapter monAdapter;
    private SessionManager sessionManager;

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

        // Initialisation des outils
        sessionManager = new SessionManager(getApplicationContext());
        lstVisiteur = (ListView) findViewById(R.id.listView);
        retour = (Button) findViewById(R.id.buttonRetour);
        boutonDeconnexion = (Button) findViewById(R.id.buttonDeconnexion);

        // Récupération des données via le DAO
        VisiteurDAO unVisiteurDAO = new VisiteurDAO(this);
        ArrayList<Visiteur> collecVisiteur = unVisiteurDAO.getLesVisiteurs();

        // Configuration de l'adaptateur pour la liste
        monAdapter = new ArrayAdapter<this, android.R.layout.simple_list_item_1, collecVisiteur>;
        lstVisiteur.setAdapter(monAdapter);

        // Gestion du clic sur un élément de la liste
        lstVisiteur.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long l) {
                // Récupération du visiteur sélectionné
                selectedVisiteur = (Visiteur) parent.getItemAtPosition(position);

                // Passage à l'activité de détails avec transmission de l'ID
                Intent intent = new Intent(ConsultActivity.this, DetailsVisActivity.class);
                intent.putExtra("id", selectedVisiteur.getId());
                // (Optionnel) On peut passer les autres infos ou les re-requêter via l'ID
                intent.putExtra("nom", selectedVisiteur.getNom());
                intent.putExtra("prenom", selectedVisiteur.getPrenom());
                startActivity(intent);
            }
        });

        // Bouton pour revenir au menu
        retour.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                finish(); // Ferme l'activité pour revenir à la précédente
            }
        });

        // Bouton de déconnexion
        boutonDeconnexion.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                deconnecter();
            }
        });
    }
}
```

# RÉALISATIONS

```
public class DetailsVisActivity extends AppCompatActivity {

    // Attributs pour l'identification du visiteur
    private String nom, id;
    private String prenom;

    // Managers
    private SessionManager sessionManager;
    private VisiteurDAO recupVisiteur;
    private Visiteur unVisiteur;

    // Composants UI
    private Button boutonModif;
    private Button boutonSup;
    private Button boutonDeconnexion;
    private Button boutonRetour;
    private EditText txtId, txtNom, txtPrenom, txtLogin, txtMdp, txtAdresse, txtCP, txtVille, txtDateE;

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

        // Liaison des vues
        txtId = (EditText) findViewById(R.id.editTextId);
        txtNom = (EditText) findViewById(R.id.editTextNom);
        txtPrenom = (EditText) findViewById(R.id.editTextPre);
        txtLogin = (EditText) findViewById(R.id.editTextLogin);
        txtMdp = (EditText) findViewById(R.id.editTextMdp);
        txtAdresse = (EditText) findViewById(R.id.editTextAdRue);
        txtCP = (EditText) findViewById(R.id.editTextCP);
        txtVille = (EditText) findViewById(R.id.editTextVille);
        txtDateE = (EditText) findViewById(R.id.editTextDate);

        boutonDeconnexion = (Button) findViewById(R.id.buttonDeconnexion);
        boutonModif = (Button) findViewById(R.id.buttonValider);
        boutonRetour = (Button) findViewById(R.id.buttonRetour);
        boutonSup = (Button) findViewById(R.id.buttonSupprimer);

        sessionManager = new SessionManager(getApplicationContext());
        recupVisiteur = new VisiteurDAO(this);

        // --- GESTION DES DROITS (RÔLES) ---
        if (!sessionManager.isAdmin()) {
            // Un visiteur simple ne peut pas modifier ou supprimer ou ajouter
            boutonModif.setVisibility(View.GONE);
            boutonSup.setVisibility(View.GONE);

            // Désactivation des champs pour empêcher la saisie
            desactiverEdition();

            Toast.makeText(this, "Mode lecture seule (Visiteur)", Toast.LENGTH_SHORT).show();
        }

        // Récupération de l'ID passé par l'activité précédente
        Intent intent = getIntent();
        id = intent.getStringExtra("id");

        // Chargement des données du visiteur depuis la base
        unVisiteur = recupVisiteur.getVisiteur(id);
    }
}
```

## Contrôleur - Détails information un visiteur

# RÉALISATIONS

```
public class ModifierActivity extends AppCompatActivity {
    private Spinner choixV;
    private ArrayList<Visiteur> lesVisiteurs;
    private EditText txtId,txtNom,txtPrenom,txtLogin,txtMdp,txtAdresse,txtCP,txtVille,txtDateE;
    private Visiteur leVisiteur;
    private Button boutonModif;
    private Button boutonDeconnexion;
    private VisiteurDAO leVisteurDao;
    private SessionManager sessionManager;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_modifier);
        lesVisiteurs = new ArrayList<Visiteur>();
        leVisteurDao = new VisiteurDAO(this);
        lesVisiteurs = leVisteurDao.getLesVisiteurs();
        choixV = (Spinner) findViewById(R.id.spinner);
        sessionManager = new SessionManager(getApplicationContext());
        boutonModif=(Button) findViewById(R.id.buttonValiderModif);

        txtNom = (EditText) findViewById(R.id.editTextNom);
        txtPrenom = (EditText) findViewById(R.id.editText3);
        txtLogin = (EditText) findViewById(R.id.editTextLogin);
        txtMdp = (EditText) findViewById(R.id.editTextMdp);
        txtAdresse = (EditText) findViewById(R.id.editTextAdRue);
        txtCP = (EditText) findViewById(R.id.editTextCP);
        txtVille = (EditText) findViewById(R.id.editTextVille);
        txtDateE = (EditText) findViewById(R.id.editTextDate);
        boutonDeconnexion = (Button) findViewById(R.id.buttonDeconnexion);
        boutonDeconnexion.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                deconnecter();
            }
        });

        ArrayAdapter adapter = new ArrayAdapter(this, android.R.layout.simple_spinner_item,
        lesVisiteurs);

        choixV.setAdapter(adapter);
        txtNom.setText(lesVisiteurs.get(0).getNom());
        txtPrenom.setText(lesVisiteurs.get(0).getPrenom());
        txtLogin.setText(lesVisiteurs.get(0).getLogin());
        txtMdp.setText(lesVisiteurs.get(0).getMdp());
        txtAdresse.setText(lesVisiteurs.get(0).getAdresse());
        txtCP.setText(lesVisiteurs.get(0).getCodePostal());
        txtVille.setText(lesVisiteurs.get(0).getVille());
        txtDateE.setText(lesVisiteurs.get(0).getDateEmbauche());

        choixV.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int position, long l) {
                leVisiteur = (Visiteur) adapterView.getItemAtPosition(position);
                Toast.makeText(getApplicationContext(), "ID: " + leVisiteur.getId() + "\nName: " +
                leVisiteur.getPrenom(), Toast.LENGTH_SHORT).show();

                actualiser(position);
            }
        });
    }
}
```

## Contrôle - Modifier un visiteur

# RÉALISATIONS

```
public class PropositionActivity extends AppCompatActivity {

    // Boutons de navigation
    private Button boutonAjout, boutonRetour;
    private Button boutonConsult;
    private Button boutonModif;
    private Button boutonDeconnexion;

    // Gestion de session
    private SessionManager sessionManager;

    // Affichage des infos de session
    private TextView txtRole;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(com.example.tpvisiteurdao.R.layout.activity_proposition);

        // Initialisation du manager de session
        sessionManager = new SessionManager(getApplicationContext());

        // Liaison des composants UI
        boutonAjout = (Button) findViewById(com.example.tpvisiteurdao.R.id.boutonAjout);
        boutonConsult = (Button) findViewById(com.example.tpvisiteurdao.R.id.boutonConsulter);
        boutonModif = (Button) findViewById(com.example.tpvisiteurdao.R.id.boutonModifie);
        boutonRetour = (Button) findViewById(com.example.tpvisiteurdao.R.id.boutonRetour);
        boutonDeconnexion = (Button) findViewById(com.example.tpvisiteurdao.R.id.boutonDeconnexion);
        txtRole = (TextView) findViewById(com.example.tpvisiteurdao.R.id.textViewRole);

        // --- AFFICHAGE DES INFORMATIONS DE CONNEXION ---
        if (txtRole != null) {
            String roleStr = sessionManager.isAdmin() ? "Administrateur" : "Visiteur";
            txtRole.setText("Connecté : " + sessionManager.getLogin() + " (" + roleStr + ")");
        }

        // --- RESTRICTION D'ACCÈS SELON LE RÔLE ---
        // Seuls les administrateurs peuvent voir les boutons d'ajout et de modification globale
        if (!sessionManager.isAdmin()) {
            boutonAjout.setVisibility(View.GONE);
            boutonModif.setVisibility(View.GONE);
        }

        // --- GESTION DES CLICS ---

        // Aller vers le formulaire d'ajout
        boutonAjout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(PropositionActivity.this, AjoutActivity.class));
            }
        });

        // Aller vers la liste de consultation
        boutonConsult.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(PropositionActivity.this, ConsultActivity.class));
            }
        });
    }
}
```

**Contrôleur - Proposition entre les boutons : Modifier - Ajouter - Consulter**

# RÉALISATIONS

```
package com.example.tpvisiteurdao.Modele;

import android.content.Context;
import android.content.SharedPreferences;

/**
 * Classe SessionManager : Gère la persistance de la session utilisateur.
 * Utilise les SharedPreferences pour stocker les informations de connexion.
 */
public class SessionManager {

    // Nom du fichier de préférences
    private static final String PREF_NAME = "VisiteurDAOSession";

    // Clés de stockage
    private static final String KEY_LOGIN = "login";
    private static final String KEY_ROLE = "role";
    private static final String KEY_LOGGED = "isLoggedIn";

    // Définition des rôles constants
    public static final String ROLE_ADMIN = "admin";
    public static final String ROLE_VISITEUR = "visiteur";

    // Identifiants administrateur codés en dur (pour l'exercice)
    private static final String ADMIN_LOGIN = "admin";
    private static final String ADMIN_MDP = "admin";

    private SharedPreferences prefs;
    private SharedPreferences.Editor editor;

    /**
     * Constructeur : initialise les SharedPreferences
     * @param context Contexte de l'activité ou de l'application
     */
    public SessionManager(Context context) {
        prefs = context.getSharedPreferences(PREF_NAME, Context.MODE_PRIVATE);
        editor = prefs.edit();
    }

    /**
     * Vérifie si les identifiants saisis correspondent au compte administrateur
     * @param login identifiant saisi
     * @param mdp mot de passe saisi
     * @return true si admin, false sinon
     */
    public boolean isAdminCredentials(String login, String mdp) {
        return ADMIN_LOGIN.equals(login) && ADMIN_MDP.equals(mdp);
    }

    /**
     * Enregistre les informations de l'utilisateur dans les préférences
     * @param login identifiant utilisateur
     * @param role Rôle ("admin" ou "visiteur")
     */
    public void createSession(String login, String role) {
        editor.putBoolean(KEY_LOGGED, true);
        editor.putString(KEY_LOGIN, login);
        editor.putString(KEY_ROLE, role);
        editor.apply(); // Sauvegarde asynchrone
    }
}
```

**Modèle - Session Manager - Gestion des rôles Utilisateurs**

# RÉALISATIONS

```
package com.example.tpvisiteurdao.Modele;

/**
 * Classe représentant un Visiteur (POJO)
 * Cette classe définit les attributs d'un visiteur médical.
 */
public class Visiteur {
    // Attributs privés correspondant aux colonnes de la table SQL
    private String id;
    private String nom;
    private String prenom;
    private String login;
    private String mdp;
    private String adresse;
    private String codePostal;
    private String ville;
    private String dateEmbauche;

    /**
     * Constructeur complet pour initialiser un objet Visiteur
     */
    public Visiteur(String id, String nom, String prenom, String login, String mdp, String adresse,
String codePostal, String ville, String dateEmbauche) {
        this.id = id;
        this.nom = nom;
        this.prenom = prenom;
        this.login = login;
        this.mdp = mdp;
        this.adresse = adresse;
        this.codePostal = codePostal;
        this.ville = ville;
        this.dateEmbauche = dateEmbauche;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getNom() {
        return nom;
    }

    public void setNom(String nom) {
        this.nom = nom;
    }

    public String getPrenom() {
        return prenom;
    }

    public void setPrenom(String prenom) {
        this.prenom = prenom;
    }

    public String getLogin() {
        return login;
    }
}
```

Modèle - Visiteur - Classe Métier

# RÉALISATIONS

```
package com.example.tpvisiteurdao.Modele;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.util.Log;

import java.util.ArrayList;

/**
 * Classe DAO (Data Access Object) pour l'entité Visiteur.
 * Centralise toutes les opérations CRUD (Create, Read, Update, Delete) vers la base de données.
 */
public class VisiteurDAO {

    // Nom de la base de données et version
    private static String base = "BDpharma";
    private static int version = 1;
    // Helper pour l'accès à la base SQLite
    BD_SQLiteOpenHelper accesBD;

    /**
     * Constructeur : initialise le helper d'accès à la base
     * @param ct Contexte de l'application
     */
    public VisiteurDAO(Context ct){
        accesBD = new BD_SQLiteOpenHelper(ct, base, null, version);
    }

    /**
     * Vérifie les identifiants de connexion d'un visiteur
     * @param unIdentifiant Le login saisi
     * @param unMotDePasse Le mot de passe saisi
     * @return true si la connexion est réussie, false sinon
     */
    public boolean seConnecter(String unIdentifiant, String unMotDePasse) {
        SQLiteDatabase db = accesBD.getReadableDatabase();
        boolean succes = false;

        // Requête paramétrée pour éviter les injections SQL
        Cursor curseur = db.rawQuery(
            "SELECT mdp FROM visiteur WHERE login = ?", new String[]{unIdentifiant});

        if (curseur != null) {
            if (curseur.moveToFirst()) {
                String mdpBD = curseur.getString(0);
                if (mdpBD.equals(unMotDePasse)) {
                    succes = true;
                }
            }
            curseur.close();
        }
        return succes;
    }

    /**
     * Ajoute un nouveau visiteur dans la base de données
     * @param unVisiteur L'objet visiteur à insérer
     * @return l'ID de la ligne insérée, ou -1 en cas d'erreur
     */
}
```

Modèle - Visiteur - Classe DAO

# RÉALISATIONS

```
public class MainActivity extends AppCompatActivity {

    // Composants UI
    private Button bouton;
    private EditText login,mdp;

    // Objets métier
    private SessionManager sessionManager;
    private com.example.tpvisiteurdao.Modele.VisiteurDAO unVisiteurDao;
    private String loginText,mdpText;
    private com.example.tpvisiteurdao.Modele.Visiteur unVisiteur;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(com.example.tpvisiteurdao.R.layout.activity_main);

        // Gestion des Insets pour le design Edge-to-Edge
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(com.example.tpvisiteurdao.R.id.main),
(v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        // Initialisation des managers
        sessionManager = new SessionManager(getApplicationContext());
        unVisiteurDao = new VisiteurDAO(this);

        // Récupération des éléments graphiques
        bouton=(Button)findViewById(com.example.tpvisiteurdao.R.id.buttonV);
        login=(EditText)findViewById(com.example.tpvisiteurdao.R.id.editTextNom);
        mdp=(EditText) findViewById(com.example.tpvisiteurdao.R.id.editTextMDP);
        unVisiteurDao = new VisiteurDAO(this);

        // Événement clic sur le bouton de connexion
        bouton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String loginText = login.getText().toString().trim();
                String mdpText = mdp.getText().toString().trim();

                // Validation des champs vides
                if (loginText.isEmpty() || mdpText.isEmpty()) {
                    Toast.makeText(getApplicationContext(),
                        "Veuillez remplir le login et le mot de passe.",
                        Toast.LENGTH_LONG).show();
                    return;
                }

                // --- LOGIQUE D'AUTHENTIFICATION ---

                // 1. Test du compte ADMIN (via SessionManager)
                if (sessionManager.isAdminCredentials(loginText, mdpText)) {
                    sessionManager.createSession(loginText, SessionManager.ROLE_ADMIN);
                    Toast.makeText(getApplicationContext(), "Connecté en tant qu'administrateur",
Toast.LENGTH_SHORT).show();
                    startActivity(new Intent(MainActivity.this, PropositionActivity.class));
                    return;
                }
            }
        });
    }
}
```

**Main Activity - Gestion de connexion des sessions Utilisateur / Administrateur**

# RÉALISATIONS

```
<?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="236dp"
        android:orientation="vertical"
        tools:ignore="MissingConstraints">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="66dp"
            android:layout_weight="1"
            android:orientation="horizontal">

            <TextView
                android:id="@+id/txtNom"
                android:layout_width="248dp"
                android:layout_height="38dp"
                android:layout_weight="1"
                android:text="Login:"
                android:textSize="16sp"
                tools:layout_editor_absoluteX="48dp"
                tools:layout_editor_absoluteY="75dp" />

            <EditText
                android:id="@+id/editTextNom"
                android:layout_width="match_parent"
                android:layout_height="61dp"
                android:layout_weight="1"
                android:ems="10"
                android:inputType="textPersonName"
                tools:layout_editor_absoluteX="141dp"
                tools:layout_editor_absoluteY="63dp" />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="58dp"
            android:layout_weight="1"
            android:orientation="horizontal">

            <TextView
                android:id="@+id/textViewMdp"
                android:layout_width="wrap_content"
                android:layout_height="37dp"
                android:layout_weight="1"
                android:text="Mot de passe:"
                android:textSize="16sp"
                tools:layout_editor_absoluteX="41dp"
                tools:layout_editor_absoluteY="155dp" />

            <EditText
                android:id="@+id/editTextMDP"

```

Vues - Gestion de connexion des sessions Utilisateur / Administrateur

# RÉALISATIONS

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:transcriptMode="alwaysScroll" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="8dp">

        <Button
            android:id="@+id/buttonRetour"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:layout_marginEnd="4dp"
            android:text="Retour" />

        <Button
            android:id="@+id/buttonDeconnexion"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:layout_marginStart="4dp"
            android:text="Se déconnecter"
            android:textColor="@color/white" />

    </LinearLayout>
</LinearLayout>
```

Vues - Consulter tout les visiteurs

# RÉALISATIONS

```
<?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:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Controleur.DetailsVisActivity">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/textViewId"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Identifiant:" />

        <EditText
            android:id="@+id/editTextId"
            android:layout_width="239dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:inputType="textPersonName"
            android:text="Name" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/textViewNom"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Nom:" />

        <EditText
            android:id="@+id/editTextNom"
            android:layout_width="239dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:inputType="textPersonName"
            android:text="Name" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
```

Vues - Détails des informations d'un visiteur

# RÉALISATIONS

```
<?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:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Controleur.PropositionActivity">
<TextView
    android:id="@+id/textViewRole"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="Connecté en tant que : ..."
    android:textAlignment="center"
    android:textSize="16sp"
    android:textStyle="italic"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="20dp" />

<LinearLayout
    android:id="@+id/groupButtons"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintTop_toBottomOf="@id/textViewRole"
    app:layout_constraintBottom_toTopOf="@id/buttonDeconnexion"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:padding="10dp">

    <Button
        android:id="@+id/buttonAjout"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:layout_marginBottom="10dp"
        android:text="Ajouter un Visiteur" />

    <Button
        android:id="@+id/buttonConsulter"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:layout_marginBottom="10dp"
        android:text="Consulter la liste" />

    <Button
        android:id="@+id/buttonModifieur"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:layout_marginBottom="10dp"
        android:text="Modifier un profil" />

    <Button
        android:id="@+id/buttonRetour"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        style="@style/Widget.MaterialComponents.Button.TextButton"
        android:text="Retour" />
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Vues - Proposition de tout les boutons : Modifier - Ajouter - Consulter

# RÉALISATIONS

```
<?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:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Controleur.AjoutActivity">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/textViewId"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Identifiant:" />

        <EditText
            android:id="@+id/editTextId"
            android:layout_width="239dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:inputType="textPersonName"
            android:text="Name" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/textViewNom"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Nom:" />

        <EditText
            android:id="@+id/editTextNom"
            android:layout_width="239dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:inputType="textPersonName"
            android:text="Name" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```

Vues - Ajout d'un visiteur

# RÉALISATIONS

```
<?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:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Controleur.ModifierActivity">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Choisir un visiteur:" />

    <Spinner
        android:id="@+id/spinner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/textViewNom"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Nom:" />

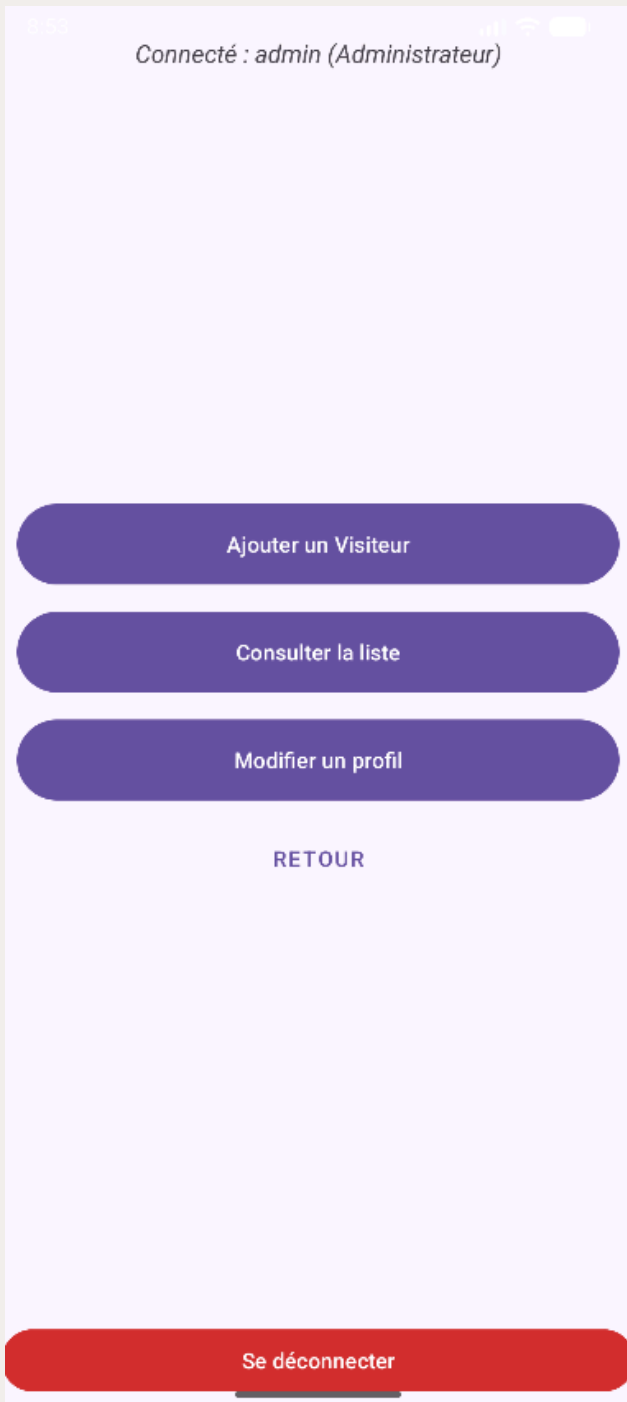
        <EditText
            android:id="@+id/editTextNom"
            android:layout_width="239dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:inputType="textPersonName"
            android:text="Name" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="horizontal">

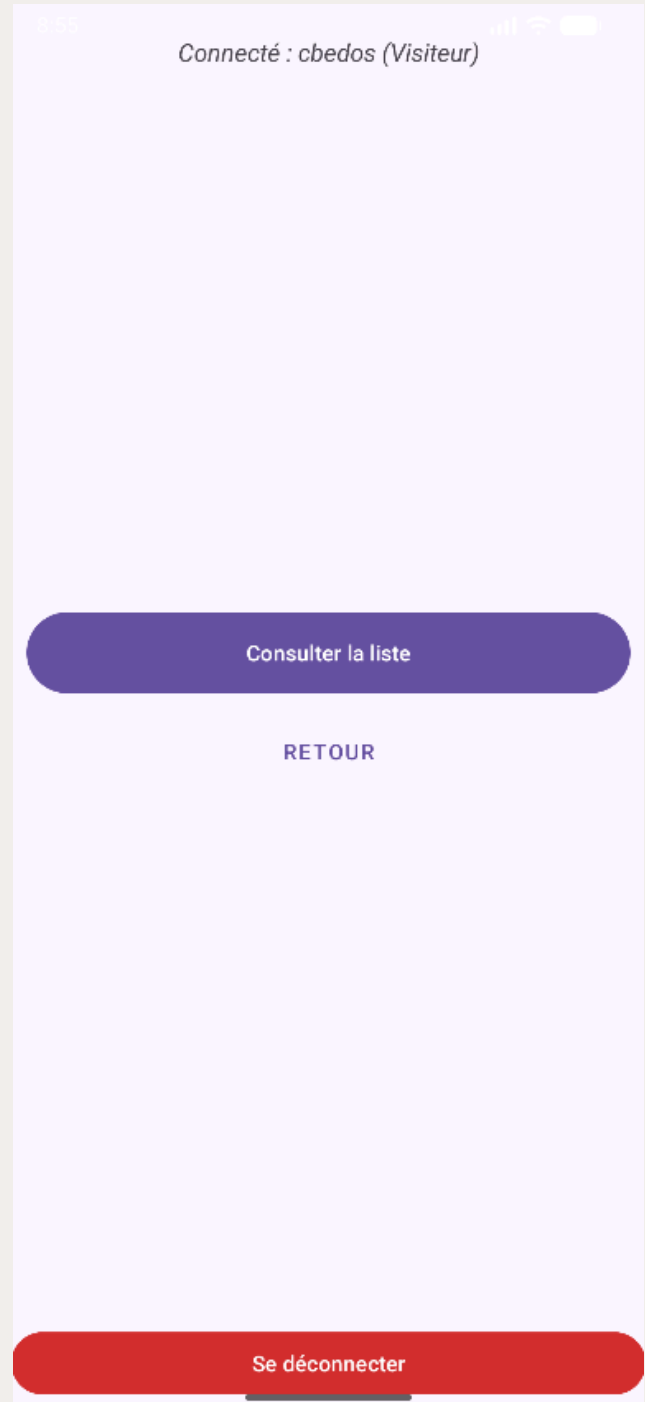
        <TextView
            android:id="@+id/textViewP"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="P:" />
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Vues - Modifier un visiteur

# RÉSULTATS

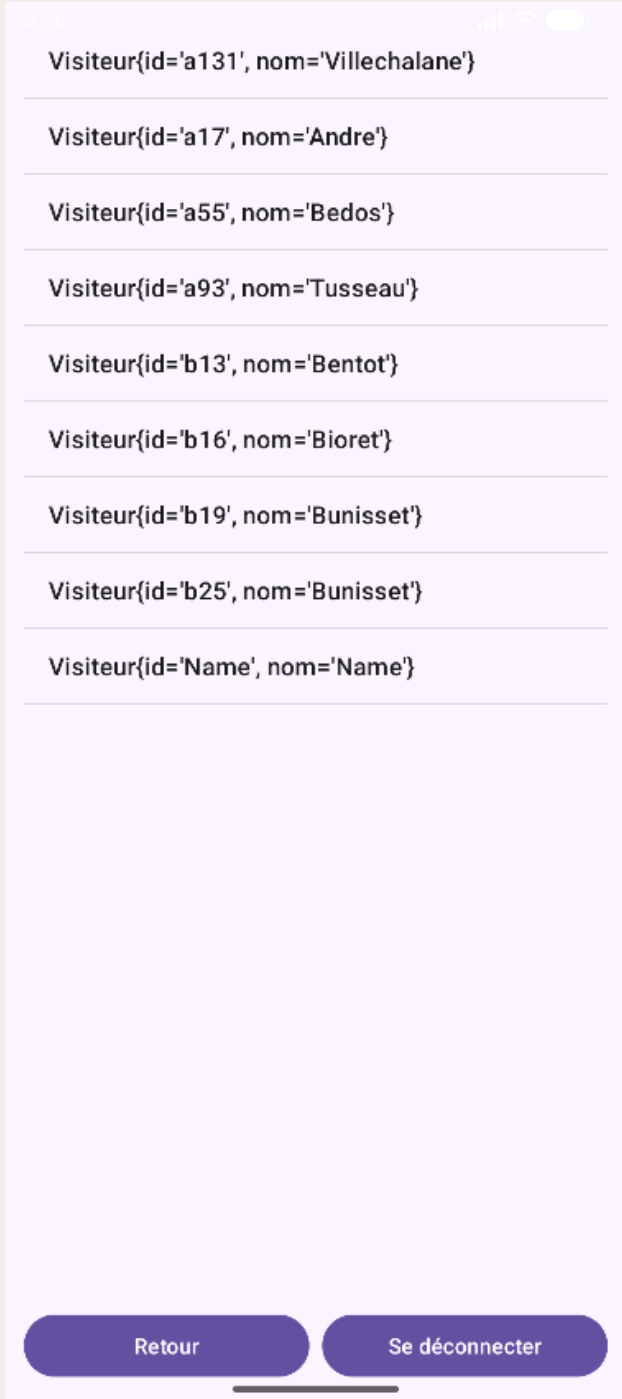


UML\_Android\_VisiteurDAO

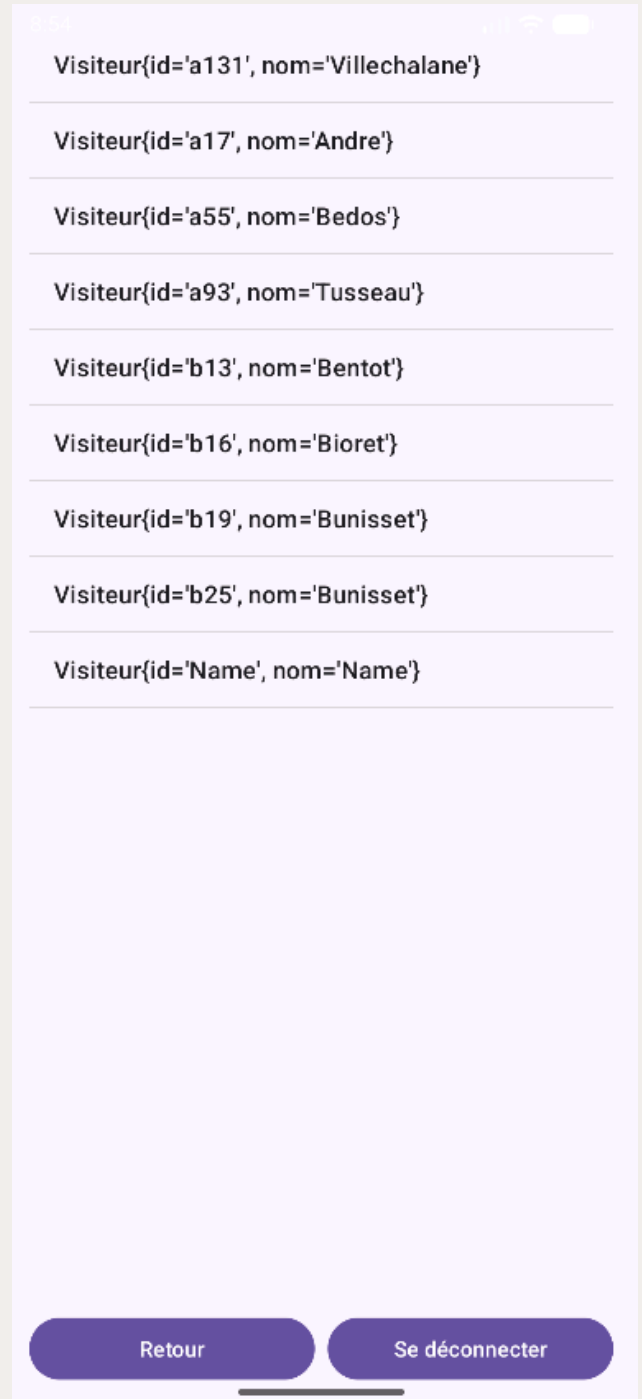


UML\_Android\_VisiteurDAO

# RÉSULTATS



UML\_Android\_VisiteurDAO



UML\_Android\_VisiteurDAO

# RÉSULTATS

Identifiant:	<u>a17</u>
Nom:	<u>Andre</u>
Prenom:	<u>David</u>
Login:	<u>dandre</u>
Mot de passe:	<u>oppg5</u>
Adresse Rue:	<u>1 rue Petit</u>
Code Postal:	<u>46200</u>
Ville:	<u>Lalbenque</u>
Date Embauche:	<u>1998-11-23</u>

[Valider](#) [Supprimer](#)

[RETOUR À LA LISTE](#)

[Se déconnecter](#)

# RÉSULTATS

Identifiant:

Nom:

Prenom:

Login:

Mot de passe:

Adresse Rue:

Code Postal:

Ville:

Date Embauche:

UML\_Android\_VisiteurDAO

# RÉSULTATS

Identifiant:	a131
Nom:	Villechalane
Prenom:	Louis
Login:	aaaa
Mot de passe:	8 rue des textmes
Adresse Rue:	aribiA
Code Postal:	46000
Ville:	Cahors
Date Embauche:	2005-12-21

[RETOUR À LA LISTE](#)

[Se déconnecter](#)