Correction Devoir Surveillé

Matière : Développement Mobile Avancé

Exercice1 (10 points)

```
private void ajouter() {
        SQLiteSuivi b=new SQLiteSuivi(this, "suivi.db", null, 1);
        SQLiteDatabase db;
        db = b.getWritableDatabase();
        ContentValues v = new ContentValues();
        v.put("matricule", edMatricule.getText().toString());
        v.put("marque", edMarque.getText().toString());
        v.put("couleur", edCouleur.getText().toString());
        v.put("dateDE", "");
        v.put("prixDE", 0);
        db.insert("Voiture", null, v);
        db.close();
        edMatricule.setText("");
        edMarque.setText("");
        edCouleur.setText("");
        edMatricule.requestFocus();
}
private void remplir() {
        SQLiteSuivi b = new SQLiteSuivi(this, "suivi.db", null, 1);
        SQLiteDatabase db = b.getWritableDatabase();
        String sql = "Select * From Voiture;";
        Cursor c = db.rawQuery(sql, null);
        adpVoiture.clear();
        while (c.moveToNext()) {
            int id = c.getInt(0);
            String matricule = c.getString(1);
            String marque = c.getString(2);
            String couleur = c.getString(3);
            String dateDE = c.getString(4);
            int prixDE = c.getInt(5);
            Voiture v = new Voiture(id, matricule, marque, couleur, dateDE, prixDE);
            adpVoiture.add(v);
       }
  }
  protected void actualiser() {
        Voiture v = (Voiture) spVoiture.getSelectedItem();
        if (v != null) {
            edDateDE.setText(v.getDateDE());
            edPrixDE.setText(v.getPrixDE() + "");
        }
 }
 private void modifier() {
        if (spVoiture.getSelectedItemPosition() >= 0) {
            if (!edPrixDE.getText().toString().isEmpty()) {
                Voiture voiture = (Voiture) spVoiture.getSelectedItem();
                ContentValues v = new ContentValues();
                v.put("dateDE", edDateDE.getText().toString());
                v.put("prixDE", Integer.parseInt(edPrixDE.getText().toString()));
                SQLiteSuivi b = new SQLiteSuivi(this, "suivi.db", null, 1);
                SQLiteDatabase db = b.getWritableDatabase();
                db.update("Voiture", v, "id=" + voiture.getId(), null);
                finish();
            }
        }
}
```

Exercice2 (10 points)

```
private void remplir() {

RequestQueue queue = Volley.newRequestQueue(this);
String url = "http://192.168.10.19 :80/ReservationVoiture/ListeVoitures.php";
StringRequest sr = new StringRequest(Request.Method.POST, url,
     new Response.Listener<String>() {
       @Override
       public void onResponse(String response) {
try{
         JSONObject json = new JSONObject(response);
           JSONArray aPr = json.getJSONArray("voitures");
           for (int i = 0; i < aPH.length(); i++) {
             JSONObject o = aPr.getJSONObject(i);
             int id = Integer.parseInt(o.getString("id"));
             String matricule = o.getString("matricule");
             String marque = o.getString("marque");
             String couleur = o.getString("couleur");
             Voiture v = new Voiture(id,matricule,marque,couleur);
             adpVoiture.add(v);
          }
} catch (JSONException error) {
                 Toast t = Toast.makeText(Recherche.this, "Problème d'analyse JSON: " +
error.getMessage(), Toast.LENGTH_LONG);
                 t.show();
               }
       }
     }, new Response.ErrorListener() {
          @Override
          public void onErrorResponse(VolleyError e) {
              Toast t = Toast.makeText(Recherche.this, "Problème d'appel HTTP: " +
e.getMessage(), Toast.LENGTH_LONG);
          t.show();
          }
        })
  };
  queue.add(sr);

   }
```

```
private void reserver() {
RequestQueue queue = Volley.newRequestQueue(this);
String url = "http://192.168.10.19 :80/ReservationVoiture/Reservation.php";
StringRequest sr = new StringRequest(Request.Method.POST, url,
     new Response.Listener<String>() {
       @Override
       public void onResponse(String response) {
       try{
         JSONObject json = new JSONObject(response);
            String reponse = json.getString("ETAT");
            if (reponse.equals("SUCCES"))
               finish();
            else {
           Toast t = Toast.makeToast(Ajout.this,"Problème dans
                           La réservation!",Toast.LENGTH_LONG);
                 t.show() ;
            }
       } catch (JSONException error) {
            Toast t = Toast.makeText(Ajout.this, "Problème d'analyse JSON: " +
                    error.getMessage(), Toast.LENGTH_LONG);
               t.show();
       }
       }
    }, new Response.ErrorListener() {
         @Override
         public void onErrorResponse(VolleyError e) {
             Toast t = Toast.makeText(Ajout.this, "Problème d'appel HTTP: " +
                     e.getMessage(), Toast.LENGTH_LONG);
         t.show();
         }
       }) {
  @Override
  public Map<String, String> getParams() throws AuthFailureError {
    HashMap<String, String> headers = new HashMap<String, String>();
    Voiture v=(Voiture)spVoiture.getSelectedItem() ;
    If(v !=null){
          headers.put("id", v.getId());
          headers.put("cin", edCIN.getText().toString());
          headers.put("dateD", edDateD.getText().toString());
          headers.put("nbJ", edNbJ.getText().toString());
    }
    return headers;
  }
  };
  queue.add(sr);
}
```