

**Devoir Surveillé**

Classe : SEM31	Matière : Développement Mobile Avancé	Nb pages : 6+4
Documents Non Autorisés	Enseignant : Souissi Hafedh	Durée : 1 heure
Barème : 20 = 10 + 10		

N.B. : - Ne donner que le code des méthodes demandées.

Exercice1 (10 points)

« **SuiviVoitures** » est une application Android qui permet à une agence de location de voitures de faire le suivi des entretiens de ses voitures. « **SuiviVoitures** » permet d'ajouter un ensemble de voitures à une base de données SQLite "suivi.db" et d'enregistrer la date du dernier entretien ainsi que son prix. Une voiture est définie par un id (id), un matricule (matricule), une marque (marque) et une couleur (couleur), une date du dernier entretien (dateDE) et un prix du dernier entretien (prixDE).

« **SuiviVoitures** » contient trois activités « **MainActivity** », « **Nouveau** » et « **Modification** ». Le code de ces classes et leurs interfaces sont donnés dans « **Annexe1** ».

- 1- Donner le code des méthodes **ajouter()** de la classe « **Nouveau** » qui permet d'ajouter la voiture saisie et d'effacer les EditText (la valeur par défaut de dateDE= "" et celle de prixDE est 0).
- 2- Donner le code de la méthode **remplir ()** de la classe « **Modification** » qui permet de remplir spVoiture par les voitures de la table "Voiture" en utilisant la requête « **Select * From Voiture** ».
- 3- Donner le code de la méthode **actualiser()** de la classe « **Modification** » qui permet d'afficher la dateDE (date Dernier Entretien) et le prixDE (prix Dernier Entretien) de la voiture sélectionné dans spVoiture dans les EditText correspondants.
- 4- Donner le code de la méthode **modifier()** de la classe « **Modification** » qui permet de modifier dans la table "Voiture" la **dateDE** (date Dernier Entretien) et la **prixDE** (prix Dernier Entretien) de la voiture sélectionné dans spVoiture.

Exercice2 (10 points)

« **GestionProjets** » est une application Android qui permet à son utilisateur d'ajouter des projets à une base de données distante et d'effectuer des recherches sur les noms des projets ; le résultat d'une recherche est affiché dans un ListView. Un projet est défini par un id (id), un nom (nom), un budget (bud), une latitude (lat) et une longitude (lon).

« **GestionProjets** » contient trois activités : « MainActivity », « Ajout » et « MapsActivity ». Le code de ces classes et leurs interfaces sont donnés dans « Annexe2 ».

« **GestionProjets** » appelle une application Web hébergées dans l'adresse « http://192.168.10.14:80/GestionProjets / » et elle contient deux pages « Ajout.php » et « Recherche.php » qui utilisent la méthode « POST ».

« Ajout.php » prend **quatre paramètres** qui sont le nom (nom), le budget (bud), la latitude (lat) et la longitude (lon) du projet, elle ajoute le projet à la table « projet » de la base MySQL et retourne :

- { " **ETAT**" : "SUCCES" } en cas de succès
- { " **ETAT**" : "ECHEC" } en cas d'échec.

« Recherche.php » prend **un seul paramètre** qui est le nom (nom), elle effectue une recherche dans la table « projet » de la base MySQL et retourne un objet JSON de la forme suivante :

```
{  "projets":[
    { "id": "1", "nom": "Aménagement1", "bud": "12000",
      "lat": "34.7805448", "lon" : " 10.7927901" },
    { "id": "2", "nom": "Aménagement2", "bud": "14000",
      "lat": "34.7980853", "lon" : "10.7805919" },
    { "id": "3", "nom": "Aménagement3", "bud": "16000",
      "lat": "36.4403357", "lon" : "10.7295528" }
  ]
}
```

- 1- Donner le code de la méthode **ajouter()** de la classe « **Ajout** » qui permet d'ajouter le projet saisie, si la réponse est { " **ETAT**" : "SUCCES" } elle ferme l'activité, sinon elle affiche un message d'erreur.
- 2- Donner le code de la méthode **rechercher()** de la classe « **Recherche** » qui permet de :
 - a. rechercher les projets qui correspondent au nom saisie,
 - b. vider la ListView,
 - c. analyser la réponse JSON,
 - d. ajouter les projets trouvés au ListView.

Annexe1

SuiviVoitures

Nouvelle

Matricule

1234 Tunisie 240

Marque

BMW

Couleur

Bleu

AJOUTER

SuiviVoitures

NOUVEAU

MODIFICATION

SuiviVoitures

Modification

Voiture

1-1234 Tunisie 240-Bleu

Date Dernier Entretien

12-11-2020

Prix Dernier Entretien

250

MODIFIER

```
package com.suivi;
public class Voiture {
    private int id;
    private String matricule;
    private String marque;
    private String couleur;
    private String dateDE;
    private int prixDE;
    public Voiture(int id, String matricule, String marque,
        String couleur, String dateDE, int prixDE) {
        this.id = id;
        this.matricule = matricule;
        this.marque = marque;
        this.couleur = couleur;
        this.dateDE = dateDE;
        this.prixDE = prixDE;
    }
    public int getId() { return id; }
    public String getMatricule() { return matricule; }
    public String getMarque() { return marque; }
    public String getCouleur() { return couleur; }
    public String getDateDE() { return dateDE; }
    public int getPrixDE() { return prixDE; }
    @NonNull
    @Override
    public String toString() {
        return id+"-"+matricule+"-"+couleur;
    }
}
package com.suivi;
public class SQLiteSuivi extends SQLiteOpenHelper {
    public SQLiteSuivi(Context context, String name, SQLiteDatabase.CursorFactory factory,
        int version) {
        super(context, name, factory, version);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        String sql="create table Voiture (id INTEGER PRIMARY KEY AUTOINCREMENT,matricule
text NOT NULL,marque text NOT NULL,couleur text NOT NULL,dateDE text NOT NULL,prixDE
INTEGER );";
        db.execSQL(sql);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    }
}
```

```
package com.suivi;  
public class MainActivity extends AppCompatActivity{
```

```
    ...  
}  
package com.suivi;  
public class Nouveau extends AppCompatActivity {  
    private EditText edMatricule;  
    private EditText edMarque;  
    private EditText edCouleur;  
    private Button btnAjouter;
```

```
    ...  
  
    private void ajouter() {  
        // Permet d'ajouter la voiture saisie à la base SQLite,  
        // efface les trois EditText  
  
    }
```

```
}  
package com.suivi;  
public class Modification extends AppCompatActivity {  
    private Spinner spVoiture;  
    private EditText edDateDE;  
    private EditText edPrixDE;  
    private Button btnModifier;  
    private ArrayAdapter<Voiture> adpVoiture;  
    ...
```

```
    private void remplir() {  
        //Permet de remplir le Spinner spVoiture par les voitures de la table.  
  
    }
```

```
    protected void actualiser() {  
        //Permet d'afficher dans edDateDE et edPrixDE les valeurs relatives à  
        //la voiture sélectionnée dans le spinner spVoiture.  
    }
```

```
    private void modifier() {  
        //Si le Spinner spVoiture contient une sélection et si edPrixDE ne contient pas  
        //une chaîne vide, alors elle modifie la date du dernier entretien et  
        //son prix pour la voiture sélectionnée dans le spinner spVoiture  
  
    }
```

```
}
```

Annexe2

GestionProjets

Ajout

Nom

Aménagement route Tunis

Budget

10000 DT

Latitude

34.802899

Longitude

10.761270

AJOUTER

ANNULER

GestionProjets

AJOUTER

RECHERCHER

Gestion Projets

Recherche

Nom

Aménagement

Liste des projets

1 - Aménagement1 -12000.0DT

2 - Aménagement2 -14000.0DT

3 - Aménagement2 -16000.0DT

JSONObject

```
JSONObject json = new JSONObject("{\"c1\":\"v1\",\"c2\":\"12\",\"c3\":\"12.4\"}");
String s = json.getString("c1");
int s = json.getInt("c2");
float s = Float.parseFloat(json.getString ("c3"));
```

JSONArray

```
try {
    JSONArray a = json.getJSONArray("liste");
    for (inti = 0; i<a.length(); i++) {
        JSONObject o = a.getJSONObject(i);
    }
} catch (JSONException e) {
    e.printStackTrace();
}
```

```

package projet.com.gestionprojet;
public class Projet {
    private int id;
    private String nom;
    private float bud;
    private double lat;
    private double lon;
    public Projet(int id, String nom, float bud, double lat, double lon) {
        this.id = id;          this.nom = nom;          this.bud = bud;
        this.lat = lat;        this.lon = lon;
    }
    public int getId() {      return id;      }
    public String getNom() {  return nom;    }
    public float getBud() {   return bud;     }
    public double getLat() {  return lat;     }
    public double getLon() {  return lon;     }
    @Override
    public String toString() {
        return id + " - " + nom + " - " + bud + "DT" ;
    }
}

package projet.com.gestionprojet;
public class MainActivity extends AppCompatActivity {
    private Button btnAjout;
    private Button btnRecherche;
    ...
}

package projet.com.gestionprojet;
//imports
public class Ajout extends AppCompatActivity {
    private EditText edNom;
    private EditText edBud;
    private EditText edLat;
    private EditText edLon;
    private Button btnAjouter;
    private Button btnAnnuler;
    ...

    protected void ajouter() {
        // permet de verifier les données, d'ajouter le projet saisie et d'analyser la réponse
    }

}

package projet.com.gestionprojet;
public class Recherche extends AppCompatActivity {
    private EditText edNom;
    private Button btnRechercher;
    private Button btnQuitter;
    private ListView lstP;
    private ArrayAdapter<Projet> adpP;
    ...
    private void init() {
        ...
        adpP=new ArrayAdapter<Projet>(this,android.R.layout.simple_list_item_1);
        lstP.setAdapter(adpP);
        ajouterEcouteur();
    }

    protected void rechercher() {
        //permet de :
        // - rechercher les projets qui correspondent au nom saisie,
        // - vider la ListView
        // - analyser la réponse JSON,
        // - ajouter les projets trouvés au ListView.
    }

}

```