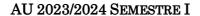
set

ISET SFAX





Devoir Surveillé			
Classe: SEM31	Matière : D éveloppement M obile A vancé		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.

Exercice 1 (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 EditTexts (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 EditTexts 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)

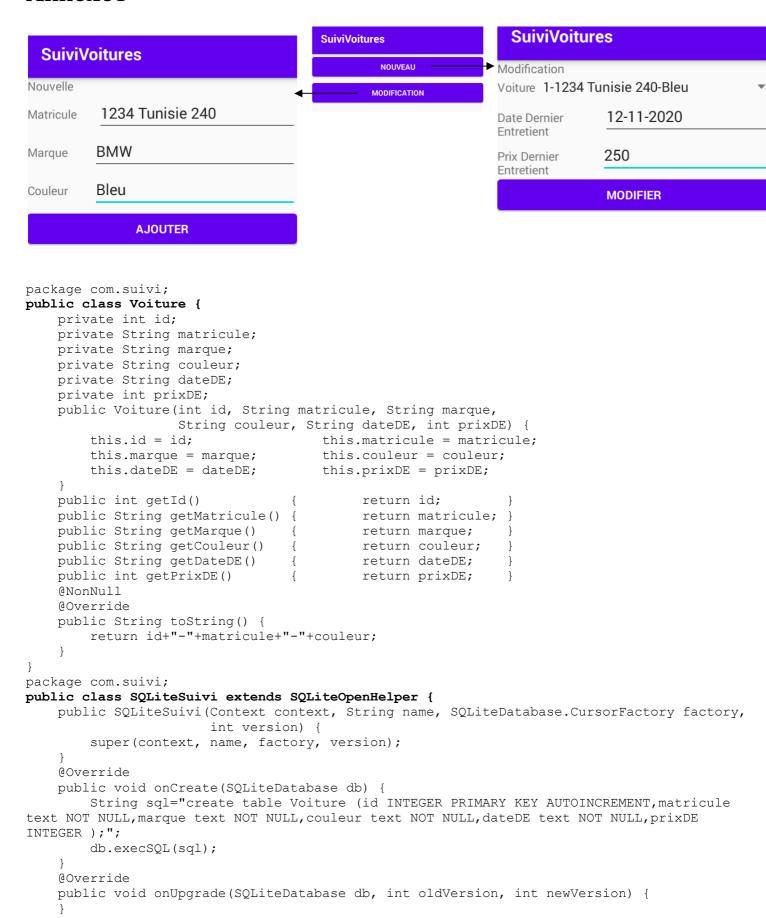
- « **GestionProjets** » est une application Android qui permet à son utilisateur d'ajouter des projets à une base de données distante et d'effectuer des recherches sur les noms des projets ; le résultat d'une recherche est affiché dans un ListView. Un projet est défini par un id (id), un nom (nom),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

- 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



```
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 EdiText
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 entretient et
        //son prix pour la voiture sélectionnée dans le spinner spVoiture
```

Annexe2



DMA

```
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.bud = bud;
                           this.nom = nom;
        this.id = id;
        this.lat = lat;
                               this.lon = lon;
                                return id;
    public int getId() {
    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.
```