Université d'El-Oued

Faculté des Sciences Exactes
Département d'Informatique
3eme année Licence
Module : **Applications mobiles**

Date : 26/05/2022
Durée : 1h

Corrigé type de Contrôle

Questions de cours : (10,5 points)

1) (a) ×
(b)
© ×
(d) ×
(e) ×
(f)

2)

conception	développement	déploiement
sticky notes, prototype, UX, ergonomie, MVC,	google usb driver, crossplatformes JAVA, activity, DAO, AVD, service, Androidmanifest, Linearlayout	téléchargement, APK, boutique en ligne, Google Play

3)

l'emplacement et le nom du fichier	correction	explication
App/manifests/Androidmanifest.xml	<uses-permission android:name android.permission.ACCESS_FINE_LOCATION />	Demander l'autorisation à l'accès, par l'app, au récepteur GPS (localisation)
App/java/package_name/mainactivity.java	setContentView(R.layout.activity_main);	Méthode qui visualise durant l'exécution l'interface graphique incorporée dans le fichier ressource layout désigné
App/java/package_name/ CommentDbHelper.java	public CommentDbHelper extends SQLiteOpenHelper	La signature de déclaration d'une classe d'aide pour une DB SQLite
App/java/package_name/ CommentDbHelper.java	private static final String TABLE_CREATE = "CREATE TABLE Comments" + "(" + COLUMN_ID + " INTEGER PRIMARY KEY autoincrement, comment text not null);";	La déclaration d'une variable décrivant une requête SQL de création d'une table Comments()
App/java/package_name/mainactivity.java	startActivity(new Intent(this, DisplayMessageActivity.class));	Méthode qui fairt appel à une activité définit par l'objet « intent »

chargé du module : K.Mohib Eddine

4) import android.app.Activity;
import android.os.Bundle;
public class **miniSkeletonActivity** extends Activity {
/* Méthode appelée à la création de l'activité */
public void onCreate (Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(**R.layout. miniSkeletonLayout**);

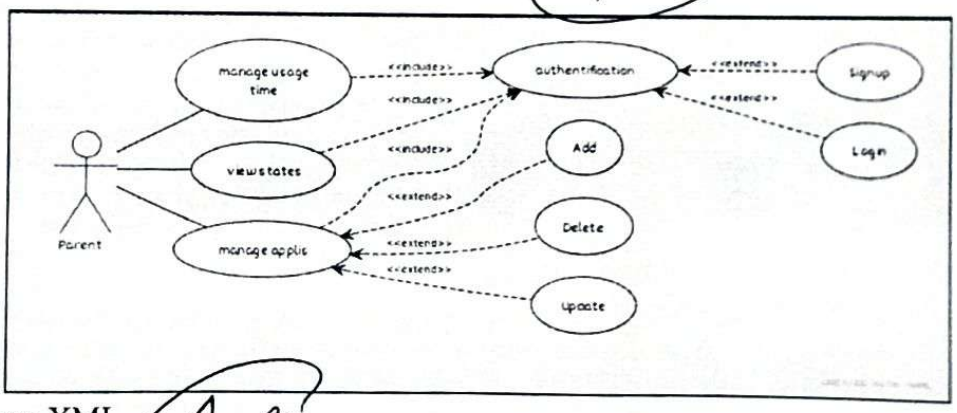
}}

Problème : (9,5 points)

1)

Type d'appli	environnement de développement	environnement de test	environnement de déploiement
Appli mob native	Android studio, Android SDK API 29 ou NDK,	Appareil physique, AVD ou Genymotion with Android	Smartphone ou Tablette with Android Q.

2)
Le diagramme de cas d'utilisation (use case) générale



3)
a) Fichier strings.XML

```
<resources>
    <!-- Default screen margins, per the Android Design guidelines. -->
   <String name="login_ID">user_ID</String>
   <String name="login_pwd">user_pwd</String>
   <String name="login_btn">Login</String>
   <String name="signup_btn">Signup</String>   </resources>
```

b) le contenu de ressource layout:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.act_srv_tp.MainActivity" >


<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/ login_ID" />
<EditText
    android:id="@+id/user_ID"
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName" >

    <requestFocus />
</EditText>

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/ login_pwd"  />
<EditText
    android:id="@+id/user_pwd"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName" >

    <requestFocus />
</EditText>

<Button android:id="@+id/action"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="@string/ login_btn"  />
<Button android:id="@+id/action"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
        android:text="@string/ signup_btn"  />
</LinearLayout>
```

4) 3,5

```
public class DBHelper extends SQLiteOpenHelper {
    public static final String DATABASE_NAME = " partner_control.db";
     public static final String Parent_TABLE_NAME = " parent";
     public static final String Parent_COLUMN_ID = " userID ";
     public static final String Parent_COLUMN_UN = " username ";
     public static final String Parent_COLUMN_PWD = "pwd";
     String CREATE_Parent_TABLE = "CREATE TABLE " + Parent_TABLE_NAME + "("
              + Parent_COLUMN_ID + "INTEGER PRIMARY KEY autoincrement," + Parent_COLUMN_Nome +
               " TEXT NOT NULL," + Parent_COLUMN_PWD + " TEXT NOT NULL );";
    public DBHelper(Context context) {
         super(context, DATABASE_NAME , null, 1);
    }
    public void onCreate(SQLiteDatabase db) {
           db.execSQL(CREATE_Parent_TABLE);
           db.execSQL("CREATE TABLE usetimes(safetimeID INTEGER PRIMARY KEY autoincrement,
           logoapp TEXT NOT NULL,  timestatrt date, timeend date); ");
           db.execSQL("CREATE TABLE stats(logID INTEGER PRIMARY KEY autoincrement,
           safetimeNAME TEXT NOT NULL,  timeacces INTEGER, timestamp date); ");
           db.execSQL("CREATE TABLE appactivity(appCODE INTEGER PRIMARY KEY autoincrement,
           logo TEXT NOT NULL,  safetime TEXT NOT NULL); ");
    }
        public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

                db.execSQL("DROP TABLE IF EXISTS parent");
                onCreate(db);
            }}
```

la classe qui représente le modèle de données

```
public class parent {
        int _userID ;
      String _username;
       String _pwd;

   // constructor
   public parent (int userID, String username nom, String pwd){
         this._ userID = userID;
         this._ username = username;
         this._pwd = pwd;
   }
   public int getuserID(){
   return this._userID;
   }
   public void setuserID(int userID){
         this._userID = userID;
   }
   public String getusername(){
         return this._username;
   }
   public void setusername(String username){
         this._username = username;
   }


}
```