Université El Arbi Ben M'Hidi Oum El Bouaghi
Faculté de sciences exactes et Sciences de la nature et de la vie
Département de Mathématique et informatique

Niveau : 3éme années licence SI
Module : Les applications mobiles

Dimanche 12 mai 2024
Durée 1h 30 m

Examen Final

Exercices 01 (4 points) : recopier les bonnes réponses

1. Dans une application mobile, une permission est nécessaire :
   A. Pour accéder à la localisation du mobile.
   B. Pour accéder aux ressources graphiques de l'application.
   C. Pour accéder aux caméras du mobile.
   D. Pour accéder à la carte SD.
2. Toutes les composant fondamentales sont déclarées dans le fichier :
   A. R.java
   B. strings.xml
   C. AndroidManifest.xml
   D. values.xml

1pt * 4

3. Lequel des énoncés suivants est faux à propos de runtime Dalvik ?
   A. Il a remplacé le moteur d'exécution Android (ART) dans les versions ultérieures d'Android.
   B. Il utilise une compilation Just-In-Time (JIT) pour exécuter le code.
   C. Il compile le code source Java directement en code machine.
4. Sous Android Studio, les intents permettent :
   A. De lancer une nouvelle activité.
   B. La communication entre applications.
   C. De gérer les autorisations et les restrictions d'accès aux ressources de l'appareil
   D. De lire des données à partir d'une base de données

Exercice 02 : (5 points)

Vous êtes un développeur chargé de créer une application de réseaux sociaux pour une start-up. L'application vise à fournir aux utilisateurs des fonctionnalités telles que l'authentification de l'utilisateur, le partage de photos, la connexion avec des amis et la réception de notifications en temps réel. En outre, l'application doit conserver l'historique des conversations que les utilisateurs ont avec leurs amis. L'entreprise souhaite accéder au plus grand nombre de clients possible dès le premier lancement sans devoir investir beaucoup d'argent.

1. Déterminez le type d'application mobile le mieux adapté à ce scénario et expliquez votre choix.
   Réponse : Le type d'application mobile le mieux adapté à ce scénario serait une application hybride. Cette décision s'explique par la volonté stratégique de la startup d'accéder au plus grand nombre de clients possible dès le premier lancement, sans dépenser beaucoup d'argent. De plus, les applications hybrides offrent un équilibre entre rentabilité et fonctionnalité, en donnant accès à des notifications en temps réel tout en tirant parti des technologies web pour une compatibilité multiplateforme.

0.5

1

2. Comment vous assurer que l'application conserve l'historique des conversations que les utilisateurs ont avec leurs amis ? et pourquoi ?
   Réponse : Pour que l'application conserve l'historique des conversations des utilisateurs avec leurs amis, vous pouvez utiliser une combinaison de bases de données locales et de stockage à distance. Vous pouvez utiliser une base de données locale sur l'appareil de l'utilisateur pour stocker les conversations récentes afin qu'elles soient accessibles même hors ligne. Parallèlement, vous pouvez synchroniser ces données avec un serveur distant pour que les utilisateurs puissent accéder à leurs conversations à partir de différents appareils et pour assurer la sauvegarde des données en cas de perte ou de changement d'appareil.

1

1

3. Supposez que le type d'application choisi est natif. Indiquez la meilleure approche à adopter pour développer l'application : utiliser uniquement des activités ou un mélange d'activités et de fragments ? et justifiez votre décision ?

   Réponse : Si l'application est de type natif, la meilleure approche pour développer l'application serait d'utiliser un mélange d'activités et de fragments. En utilisant des fragments, vous pouvez créer une interface utilisateur flexible et modulaire qui s'adapte bien à différentes tailles d'écran et offre une meilleure expérience utilisateur. Cela permet également de réutiliser efficacement le code et de faciliter la maintenance de l'application à long terme.

0.5

1

Exercice 03 : (11 points)

Au cours du développement de l'application précédente, vous avez réalisé qu'il était nécessaire de permettre aux utilisateurs de créer des comptes (fonctionnalité de sign-in). Le code XML illustré dans la figure 1 représente l'interface de cette fonctionnalité.

1. Pouvez-vous décrire la structure (layout) principale de cette interface ? LinearLayout. **1**
2. Si l'activité s'appelle Sign_activity, quels sont les noms des fichiers Java et XML correspondants ?
   Fichier java : Sign_activity.
   Fichier XML : activity_2. **1**
3. Le code XML contient des erreurs et des attributs manquants. Pouvez-vous les corriger ? (Notez que tous les EditText doivent avoir la même taille, et veuillez utiliser le numéro de ligne pour ajouter ou corriger un attribut).

0.25 erreur * 12

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:id="@+id/main"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:layout_margin="20dp"
  android:orientation="vertical"
  tools:context=".Activity2">
  <EditText
    android:id="@+id/mat"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:hint="matricule"
    android:textSize="24sp" />
  <EditText
    android:id="@+id/nom"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:hint="Nom"
    android:textSize="24sp" />
  <EditText
    android:id="@+id/prenom"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:hint="prenom“
    android:textColor="#000000"
    android:textSize="24sp" />
  <EditText
    android:id="@+id/adress"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:hint="Adress"
    android:inputType="text"
    android:textSize="24sp" />
  <EditText
    android:id="@+id/email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:hint="Email"
    android:inputType="text"
    android:textSize="24sp" />
  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:orientation="horizontal">
    <Button
      android:id="@+id/cancel"
      android:layout_width="0dp"
      android:layout_height="wrap_content"
      android:layout_weight="1"
      android:text="cancel" />
    <Button
      android:id="@+id/ajouter"
      android:layout_width="0dp"
      android:layout_height="wrap_content"
      android:layout_weight="1"
      android:text="ajouter" />
  </LinearLayout>
</LinearLayout>
```

4. La figure 2 montre la partie Java de l'activité. Complétez le code de cette classe et fournissez le code de la fonction insert (ligne 20)

```
public class Activity2 extends AppCompatActivity {
   EditText mat,nom,prenom,adress,email;
   Button cancel,ajouter;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_2);
      Database base = new Database(this);
      mat = findViewById(R.id.mat);
      nom = findViewById(R.id.nom);
      prenom = findViewById(R.id.prenom);
      adress = findViewById(R.id.adress);
      email = findViewById(R.id.email);
      cancel = findViewById(R.id.cancel);
      ajouter = findViewById(R.id.ajouter);

      ajouter.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            String matricule = mat.getText().toString();
            String Nom = nom.getText().toString();
            String Prenom = prenom.getText().toString();
            String Adress = adress.getText().toString();
            String Email = email.getText().toString();

            if(matricule.isEmpty() || Nom.isEmpty() || Prenom.isEmpty() || Adress.isEmpty() || Email.isEmpty()){
               Toast.makeText(Activity2.this, "Veuillez remplir tous les champs", Toast.LENGTH_SHORT).show();
            }else{

               int succes = base.insert(matricule,Nom,Prenom,Adress,Email);
               if(succes == 1 ){
                  Toast.makeText(Activity2.this, "insertion effectuée avec succès", Toast.LENGTH_SHORT).show();
               }else{
                  Toast.makeText(Activity2.this, "l'insertion a échoué", Toast.LENGTH_SHORT).show();
               }}}});}}


public long insertData(String Matricule,String Nom,String
                       Prenom,String Adress,String Email) {

      SQLiteDatabase db = this.getWritableDatabase();
      ContentValues values = new ContentValues();
      values.put("matricule", Matricule);
      values.put("nom", Nom );
      values.put("prenom", Prenom);
      values.put("address", Address);
      values.put("email", Email);
      long id = db.insert("my_table", null, values);
      db.close();
      return id;
}
```

1.75

1.25

0.75

0.5

0.75

0.5

0.5