

Trimestre Été, 2014

Mohamed Lokbani

IFT1155 – Examen Final –

Inscrivez tout de suite : votre nom et le code permanent.

Nom : _____ | Prénom(s) : _____ |

Signature : _____ | Login : _____ |

Date : jeudi 10 juillet 2014

Durée : 3 heures (de 17h30 à 20h30)

Local : 3195; Pavillon André-Aisenstadt

Directives :

- Toute documentation est permise.
- Calculatrice **non** permise.
- Répondre directement sur le questionnaire.
- Les réponses **doivent être brèves, précises, claires et nettement présentées.**

1. _____ /20 (1.1 à 1.10)
2. _____ /15 (2.1, 2.2, 2.3, 2.4, 2.5)
3. _____ /15 (3.1, 3.2, 3.3, 3.4, 3.5)
4. _____ /20 (4.1 4.2 4.3)
5. _____ /30 (5.1 à 5.7)

Total : _____ /100

Directives officielles

* Interdiction de toute communication verbale pendant l'examen.

* Interdiction de quitter la salle pendant la première heure.

* L'étudiant qui doit s'absenter après la première heure remettra sa carte d'étudiant au surveillant, l'absence ne devant pas dépasser 5 minutes. Un seul étudiant à la fois peut quitter la salle.

* Toute infraction relative à une fraude, un plagiat ou un copiage est signalée par le surveillant au directeur de département ou au professeur qui suspend l'évaluation.

F.A.S

Exercice 1 (20 points) Répondre par « vrai » ou « faux » en y incluant une très courte explication.

- 1.1** [VRAI | FAUX] Android Studio est un autre environnement de développement pour développer des applications Android.
- 1.2** [VRAI | FAUX] On ne peut utiliser que la machine virtuelle « Dalvik » pour l'exécution simultanée de plusieurs applications Android.
- 1.3** [VRAI | FAUX] Un composant d'une application ne peut pas démarrer le composant d'une autre application.
- 1.4** [VRAI | FAUX] « android:maxSdkVersion » indique la version de l'API pour laquelle le programme a été écrit.
- 1.5** [VRAI | FAUX] Les préférences partagées ne fonctionnent qu'avec les objets de type « boolean », « float », « int », « long » et « String ».
- 1.6** [VRAI | FAUX] On peut savoir si une base de données est en mode lecture seule avec la méthode booléenne « isOpen() ».
- 1.7** [VRAI | FAUX] Si vous installez la base de données sur un périphérique externe, il vous faut demander la permission « READ_EXTERNAL_STORAGE ».
- 1.8** [VRAI | FAUX] Les tâches asynchrones (AsyncTask) sont utilisées pour effectuer des traitements lourds (longues opérations).
- 1.9** [VRAI | FAUX] Il est possible d'afficher une page web avec un WebView.
- 1.10** [VRAI | FAUX] ~~Un service n'a pas de durée définie.~~ Pour avoir accès à la localisation GPS, nous avons besoin de demander les deux permissions :
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />.

Exercice 2 (15 points) Expliquez ce qui suit :

2.1 On peut préserver des fichiers dans la mémoire interne du périphérique. Cependant, on évite de placer de gros fichiers dans cette mémoire interne.

2.2 Il est conseillé de mettre en garde l'utilisateur avant de laisser l'application télécharger quelque chose du réseau, même si ce téléchargement est sans danger pour l'appareil.

2.3 Il est indispensable d'obtenir des informations sur l'état de la connectivité de l'appareil avec la méthode « ConnectivityManager ».

2.4 Il existe plusieurs manières pour détecter la position d'un appareil.

2.5 Il est possible d'afficher des cartes fournies par « Google Maps ». Il est possible aussi d'afficher des informations géolocalisées en fonction de la position de l'utilisateur.

Exercice 3 (15 points) Choisissez la bonne réponse en y incluant une courte explication. Il peut y avoir plusieurs bonnes réponses.

3.1 On veut « mapper » des colonnes d'une base de données SQL vers des vues d'une « ListView » dans une application Android. On utilise pour cela :

- a. SimpleCursor
- b. SimpleCursorAdapter
- c. SimpleAdapter
- d. SQLiteCursor
- e. SQLiteAdapter

3.2 Soit « MaVue » une classe qui dérive de « View » et « uneVue » une instance de « MaVue ». On veut afficher cette instance (« uneVue ») quand l'application démarre :

- a. setCurrentView(uneVue) dans la méthode startApp() de la classe principale.
- b. setContentView(uneVue) dans la méthode startApp() de la classe principale.
- c. setContentView(uneVue) dans la méthode onStart() de la classe principale.
- d. setContentView(uneVue) dans la méthode onCreate() de la classe principale.

3.3 Soit « écran1 » est l'écran principal de l'application « TestAndroid ». Si un autre écran, « écran2 », doit être ouvert à partir de « écran1 » ...

- a. « écran2 » doit faire partie de « TestAndroid ».
- b. « écran2 » peut faire partie de n'importe quelle autre application installée sur l'appareil.
- c. « écran2 » sera toujours lancé de manière asynchrone.
- d. « écran2 » peut-être lancé de manière synchrone.
- e. « écran2 » peut retourner un resultat à « écran1 » si la méthode « startActivity » est utilisée pour démarrer l'activité associée à « écran2 ».
- f. « écran2 » peut retourner un resultat à « écran1 » si la méthode « StartAcivityForResult » est utilisée pour démarrer l'activité associée à « écran2 ».

3.4 Soient les deux actions suivantes « ACTION_CALL » et « ACTION_DIAL » :

- a. Les deux actions sont sollicitées par des « intents » pour effectuer un appel téléphonique.
- b. L'action « ACTION_CALL » est sollicitée pour un appel reçu par l'appareil téléphonique.
- c. L'action « ACTION_CALL » est sollicitée pour composer un numéro téléphonique sans avoir à présenter une interface UI.
- d. L'action « ACTION_DIAL » est sollicitée pour composer un numéro téléphonique sans avoir à présenter une interface UI.
- e. L'action « ACTION_CALL » est sollicitée pour composer un numéro téléphonique sans que l'utilisateur ne soit obligé d'initier l'appel.
- f. L'action « ACTION_DIAL » est sollicitée pour composer un numéro téléphonique sans que l'utilisateur ne soit obligé d'initier l'appel.

3.5 Soit le fragment de code suivant :

```
MediaPlayer mp = new MediaPlayer();  
mp.setDataSource(Chemin_Fichier);  
// Doit-on ajouter ici quelque chose?  
mp.start();
```

- a. mp.prepare();
- b. mp.prepareAsync();
- c. mp.loadMedia();
- d. mp.loadSource();
- e. mp.loadSource(); (puis) mp.prepare();
- f. Il n'est pas nécessaire d'ajouter quoi que ce soit.

Exercice 4 (20 points)

4.1 Écrivez l'instruction qui permet de lire le fichier XML « test.xml » disponible dans le répertoire XML ou dans le répertoire « res ».

4.2 Pour se connecter au site « <http://mail.google.com> » afin de lire ses courriels, il faut fournir les champs suivants :

Champ	Valeur
"Email"	"nom_choisi@gmail.com"
"Passwd"	"le mot de passe"
"signIn"	"Sign in"

À noter que si la valeur de retour est inférieure à 400, la connexion est un succès, dans le cas contraire c'est un échec. Écrivez les quelques lignes nécessaires pour la réalisation d'une telle opération.

```
DefaultHttpClient client = new DefaultHttpClient(); // lere instruction.
```

4.3 Écrivez les quelques instructions permettant de récupérer le mobile de Michel dans la représentation JSon ci-dessous. Le champ récupéré sera sous la forme d'un « String ».

```
{
  "contacts": [
    {
      "id": "p001",
      "nom": "Michel",
      "courriel": "michel@gmail.com",
      "adresse": "Quebec",
      "telephone": {
        "mobile": "514 3430000",
        "maison": "514 3437111",
        "bureau": "514 3436111",
      }
    }
  ]
}
```


Exercice 5 (30 points) Le scénario de l'application est comme suit :

1. L'application démarre, nous sommes en présence de la figure -1-.
 2. L'utilisateur choisi l'heure à laquelle l'alarme doit se déclencher, voir figure -2-.
 3. L'utilisateur clique par la suite le bouton « Démarrer le compteur ». À cet instant, un toast est affiché avec le message « Cette alarme se déclenchera dans x heures et y minutes ». La figure -3- affiche le toast correspondant à 1 minute.
 4. Le chronomètre avance dans le temps, figure -4-. Dès qu'il aura atteint l'heure sélectionnée, nous avons 3 actions, voir figure -5- :
 - a. Un toast est affiché avec le message « Le temps est écoulé ».
 - b. L'appareil vibre pendant 4 secondes.
 - c. Une notification est déclenchée. Le titre est « Alarme » et le message est « Fin de la durée programmée ».
 5. Quand l'utilisateur clique sur la notification, figure -6-, le chronomètre est remis à zéro, voir figure -1-.
- L'application porte le nom de « Exo5FinalE14 ».
 - L'activité associée à l'application porte le nom « AlarmActivity ».
 - Pour choisir l'heure, on utilise « TimePicker » et pour le chronomètre, on utilise « Chronometer ». Il suffit de les définir dans le fichier XML de l'activité pour pouvoir les afficher puis utiliser leurs fonctionnalités.

5.1 Expliquez brièvement l'avertissement ci-dessous, signalé par l'interface de développement « Eclipse » après avoir complété le projet en question et, proposez une correction :

Warning: not targeting the latest versions of Android; compatibility modes apply. Consider testing and updating this version. Consult the android.os.Build.VERSION_CODES javadoc for details.

Resource: AndroidManifest.xml

Path : /Exo5FinalE14

Location : line 10

5.2 L'application a-t-elle besoin de permissions particulières. Si oui, mentionnez-les.

5.3 Expliquez pourquoi l'application a besoin de définir un récepteur d'évènement (BroadcastReceiver).

5.4 Écrivez le contenu du fichier « strings.xml ».

5.5 Écrivez les toasts associés au cas de figure -3- .

5.6 Écrivez le contenu du fichier « activity_main.xml ». Ce fichier contient l'interface de l'application dans le format XML.

5.7 Comme nous l'avons mentionné dans la question (5.3), L'application a besoin de définir un récepteur d'évènement. Nous allons le faire dans le fichier « AlarmReceiver.java ». Écrivez le contenu d'un tel fichier. Il n'est pas nécessaire d'inclure les « import » dans votre programme.

```
public class AlarmReceiver extends BroadcastReceiver {
```

Bon été!

Exercice -5

IFT1155 E2014

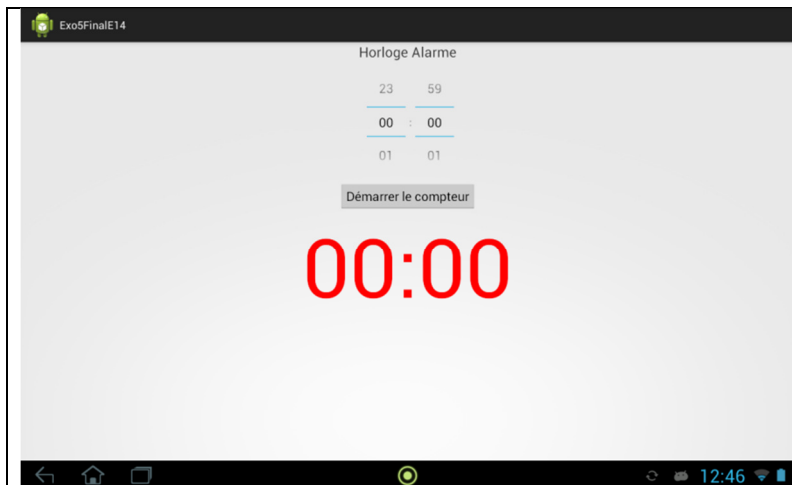


Figure -1-

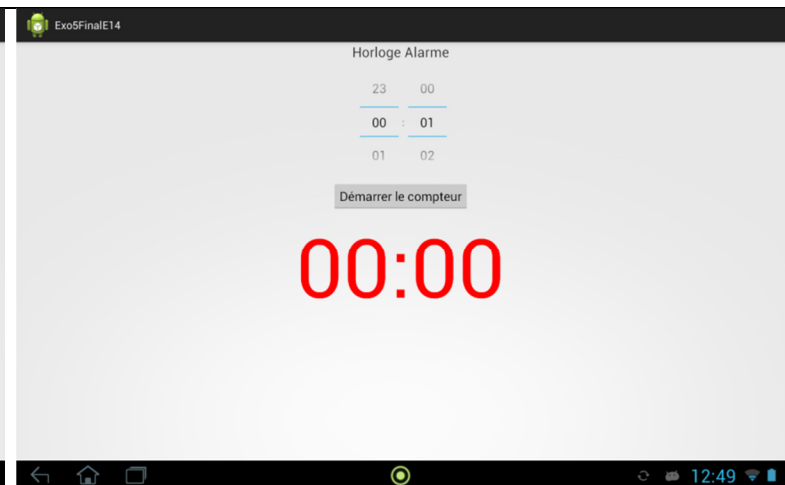


Figure -2-



Figure -3-



Figure -4-



Figure -5-



Figure -6-