

Trimestre Hiver, 2018

Mohamed Lokbani

## IFT1155 – Examen Final –

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

Nom : \_\_\_\_\_ | Prénom(s) : \_\_\_\_\_

Signature : \_\_\_\_\_ | Login : \_\_\_\_\_

Date : Vendredi 27 avril 2018

Durée : 3 heures (de 16h30 à 19h30)

Local : Z-310, Pavillon Claire McNicoll

### 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

2. \_\_\_\_\_ /24

3. \_\_\_\_\_ /20

4. \_\_\_\_\_ /36

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)** Répondez par « vrai » ou « faux » en y incluant une très courte explication.

**1.1** [VRAI | FAUX] Android 7.0 (Nougat) introduit un nouveau modèle de gestion des autorisations, qui permet de rationaliser le processus pour les utilisateurs lors de l'installation et de la mise à niveau des applications.

**1.2** [VRAI | FAUX] Il faudra vérifier toujours si l'autorisation a été accordée.

**1.3** [VRAI | FAUX] Pour une meilleure expérience utilisateur, il est important de demander l'autorisation dans le contexte.

**1.4** [VRAI | FAUX] Si votre application a besoin de géolocaliser l'utilisateur, vous devez lui demander uniquement la permission d'accéder à l'internet.

**1.5** [VRAI | FAUX] Il existe au moins deux moyens de connaître les permissions demandées par une application.

**1.6** [VRAI | FAUX] Par défaut, la clé « API Google Maps » que vous avez générée sur le site des développeurs, son utilisation est restreinte qu'à votre application Android.

**1.7** [VRAI | FAUX] Il est possible d'associer aux marqueurs sur une carte, des fenêtres d'informations.

**1.8** [VRAI | FAUX] Les notifications peuvent posséder des boutons d'action.

**1.9** [VRAI | FAUX] Afficher de la publicité via une notification est une approche qu'on conseille.

**1.10** [VRAI | FAUX] Les notifications peuvent être étendues ou regroupées.

## Exercice 2 (24 points)

Fichier Json test99.php	Affichage obtenu sur l'émulateur
<pre>{   "voiture": null,   "naissance": "1975-04-27",   "marie": false,   "age": 76,   "nom": "Tom",   "nourriture_favorite": [     "gateau",     "poisson",     "fritte"   ],   "passeport": {     "id": "100001",     "nationalite": "Canadienne"   } }</pre>	

L'application se contente d'ouvrir à distance un fichier Json, formate le contenu afin de l'afficher à l'écran. L'affichage se fait à travers la variable « textView » en se servant de la variable « affichage ».

Compléter le bloc « try » ci-dessous.

```
public class TestFinal extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        String pagephp = "http://localhost/test99.php";
        StringBuilder affichage = new StringBuilder("");
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_test_final);
        StrictMode.ThreadPolicy policy = new
            StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
        InputStream in = openHttpConnection(pagephp);
        TextView textView = findViewById(R.id.textView1);
        try {
            String jsonResult = inputStreamToString(in);

            // à compléter

            textView.setText(affichage);
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}
```

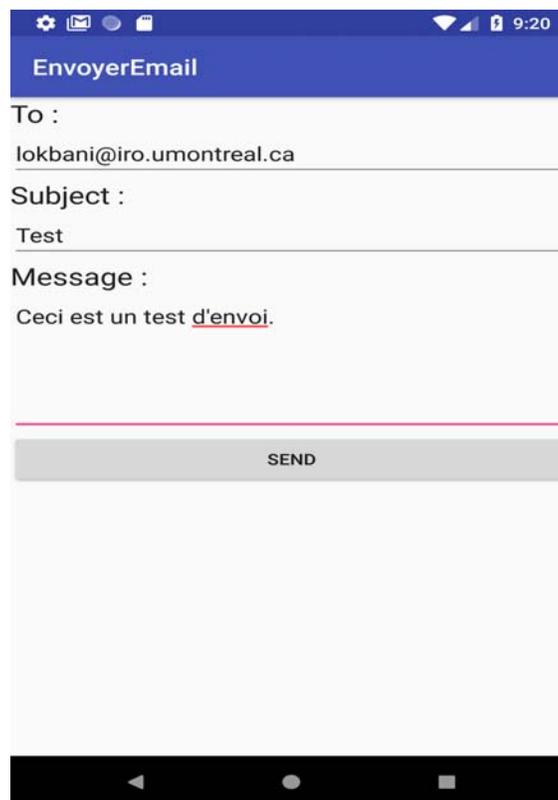


**Exercice 3 (20 points)** vous allez réaliser une application Android qui va permettre d'envoyer des courriels. Elle va se servir indirectement des applications définies sur votre appareil pour traiter ce genre d'opération (par exemple l'application Gmail). En utilisant les intentions, compléter la méthode «onClick» ci-dessous. À noter que le type associé au courriel est « message/rfc822 ».

```
public class EnvoiEmail extends AppCompatActivity {
    Button boutonSend;
    EditText textTo;
    EditText textSubject;
    EditText textMessage;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_envoi_email);
        boutonSend = findViewById(R.id.boutonSend);
        textTo = findViewById(R.id.editTextTo);
        textSubject = findViewById(R.id.editTextSubject);
        textMessage = findViewById(R.id.editTextMessage);
        boutonSend.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                // à compléter

            }
        });
    }
}
```



## Intent

### Standard Extra Data

These are the current standard fields that can be used as extra data

- [EXTRA\\_ALARM\\_COUNT](#)
- [EXTRA\\_BCC](#)
- [EXTRA\\_CC](#)
- [EXTRA\\_CHANGED\\_COMPONENT\\_NAME](#)
- [EXTRA\\_DATA\\_REMOVED](#)
- [EXTRA\\_DOCK\\_STATE](#)
- [EXTRA\\_DOCK\\_STATE\\_HE\\_DESK](#)
- [EXTRA\\_DOCK\\_STATE\\_LE\\_DESK](#)
- [EXTRA\\_DOCK\\_STATE\\_CAR](#)
- [EXTRA\\_DOCK\\_STATE\\_DESK](#)
- [EXTRA\\_DOCK\\_STATE\\_UNDOCKED](#)
- [EXTRA\\_DONT\\_KILL\\_APP](#)
- [EXTRA\\_EMAIL](#)
- [EXTRA\\_INITIAL\\_INTENTS](#)
- [EXTRA\\_INTENT](#)
- [EXTRA\\_KEY\\_EVENT](#)
- [EXTRA\\_ORIGINATING\\_URI](#)
- [EXTRA\\_PHONE\\_NUMBER](#)
- [EXTRA\\_REFERRER](#)
- [EXTRA\\_REMOTE\\_INTENT\\_TOKEN](#)
- [EXTRA\\_REPLACING](#)
- [EXTRA\\_SHORTCUT\\_ICON](#)
- [EXTRA\\_SHORTCUT\\_ICON\\_RESOURCE](#)
- [EXTRA\\_SHORTCUT\\_INTENT](#)
- [EXTRA\\_STREAM](#)
- [EXTRA\\_SHORTCUT\\_NAME](#)
- [EXTRA\\_SUBJECT](#)
- [EXTRA\\_TEMPLATE](#)
- [EXTRA\\_TEXT](#)
- [EXTRA\\_TITLE](#)
- [EXTRA\\_UID](#)



**Exercice 4 (36 points)** ci-dessous des captures d'écran de l'application « SQLite Image Demo ». L'image 1 est obtenue au démarrage de l'application. En cliquant sur le bouton « Fetch ... », l'application va télécharger une image de l'emplacement externe pour la stocker dans une base de données SQLite et afficher un toast « image 2 ». Le bouton « View ... » va extraire cette image de la base de données pour l'afficher à l'écran et afficher par la même occasion un toast « image 3 ».



Vous allez compléter les 2 méthodes « FetchImage » et « ViewImage ».

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    int STORAGE_PERMISSION_CODE = 23;
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ActivityCompat.requestPermissions(this, new String[]
        {Manifest.permission.READ_EXTERNAL_STORAGE}, STORAGE_PERMISSION_CODE);
    imageView = findViewById(R.id.imageView);
    db = this.openOrCreateDatabase("test.db", Context.MODE_PRIVATE, null);
    db.execSQL("create table if not exists imageTb ( image blob )");
}

```

- 4.1 (14 pts)** Écrire la méthode « ViewImage » qui permet de :
- récupérer l'image de la base de données
  - afficher l'image à l'écran
  - afficher le toast « Done »

```
public void viewImage(View view)
```

**4.2 (22 pts)** Écrire la méthode `fetchImage` qui permet de:

- récupérer l'image « `Android-logo.jpg` » de l'emplacement externe des images.
- insérer l'image dans la base de données.
- afficher le toast « `Image Fetched` ».

```
public void fetchImage(View view)
```

Bon été!