

# Énoncé du Travail Pratique 3

15 avril 2019

Préparé par  
Benjamin Lemelin

# 1 Résumé

Créez une application au choix utilisant une base de données ainsi qu'une technologie au choix.

## 2 Conditions de réalisation

| Valeur de la note finale | Type      | Durée      | Nombre de remises |
|--------------------------|-----------|------------|-------------------|
| 30 %                     | En équipe | 5 semaines | 1                 |

## 3 Spécifications

### 3.1 Choisir une technologie et une application à créer

Choisissez une, deux ou trois technologies non vues en classe. Votre application doit aussi faire usage d'une base de données locale SQLite. Faites rapidement valider vos choix technologiques ainsi que la nature de votre application : premier arrivé, premier servi.

Voici quelques technologies, avec suggestions d'applications pour chaque agencement. Faites attention à la compatibilité de la technologie avec votre appareil avant de débiter.

- [Fragments](#), [Navigation Drawer](#) et [RecyclerView](#).
  - [Pokedex](#) (recherche par nom, par type et par numéro).
  - Liste de tâches à faire (tâches complétées et tâches à faire).
- [Raccourcis d'application](#) et [RecyclerView](#).
  - Liste de tâches à faire (créer une tâche, compléter une tâche).
  - Liste de notes (créer une note, modifier une note, supprimer une note, colorier une note).
- [Services](#), [Notifications](#) et [RecyclerView](#).
  - Rappels (choix de l'heure, de la date, d'une étiquette ou d'une couleur).
  - Tamagotchi (plusieurs Tamagotchis en même temps, choisir un nom et une personnalité).
- [Navigation Architecture Component](#) et [RecyclerView](#).
  - [Pokedex](#) (naviguer par numéro, par type et par génération).
- [Room](#), [Transitions d'activités](#) et [RecyclerView](#).
  - Liste de notes (créer une note, modifier une note, supprimer une note, colorier une note).
  - Carnet de contacts (créer un contact, modifier un contact, supprimer un contact).
- [Bluetooth](#) et [RecyclerView](#). (Nécessite au moins 2 appareils réels)
  - Messagerie Bluetooth (liste de conversations avec historique).
- [Caméra](#), [Blobs](#) et [RecyclerView](#).
  - Appareil Photo (Stockage sur la BD avec métadonnées).
- [NFC](#), [Room](#) et [RecyclerView](#). (Nécessite au moins 2 appareils réels)
  - [Pokedex](#) (échange de Pokémons par contact).
  - Carnet de contacts (créer un profil, envoyer ce profil par NFC, supprimer un contact).
- [Montre intelligente](#). (Deux applications : 1 pour le téléphone, 1 pour la montre).
  - Liste de tâches à faire (créer une tâche, compléter une tâche).

D'autres options sont possibles, avec l'approbation du professeur.

### 3.2 Créer le projet Android

Créez un nouveau projet Android et placez-le dans un dépôt privé Git ([GitHub](#), [GitLab](#) ou [BitBucket](#)).  
Donnez en accès à votre professeur. Tout le code doit être en Java.

### 3.3 Développer l'application

Développez l'application que vous avez prévu. Pour rappel, vous devez inclure une, deux ou trois nouvelles technologies non vues en classe ainsi qu'une base de données (la complexité importe peu, mais ayez au moins 1 table avec au moins 4 champs).

### 3.4 Présentations en classe

Présentez devant la classe l'application que vous avez développée. Votre présentation devra durer 15 minutes au maximum. Présentez votre travail en incluant les informations suivantes :

- Quelle est l'application que vous avez créée ? Que fait-elle ?
- Quelle est la technologie que vous avez explorée ? Comment cela marche ?
- Quels sont les problèmes que vous avez rencontrés ? Comment les avez-vous résolus ?
- Que feriez-vous différemment si vous aviez à refaire le travail ?

Préparez convenablement votre présentation. Si vous désirez effectuer une démonstration, faites plutôt une vidéo. Ne montrez pas de code directement dans Android Studio : placez cela dans votre support de présentation. Le but est de parler des forces et des faiblesses de la technologie, pas de parler du code.

## !!!! ATTENTION !!!!

La présence aux présentations orales est obligatoire. Toute absence ou retard non justifié entraînera la note de 0 pour l'intégralité du travail.

L'exposé oral est obligatoire pour tous les membres de l'équipe. Tout membre non présent pour une raison non justifiée se fera remettre la note 0 pour l'intégralité du travail.

Tout étudiant ayant un comportement irrespectueux durant une présentation (écoute déficiente, arrogance, intérêt absent, etc.) se fera remettre la note 0 pour l'intégralité du travail.

### 3.5 Autres spécifications

| Description  |
|--|
| L'usage du langage « Java » est obligatoire.   |
| L'état de l'application doit être conservé au travers des changements d'orientation. |
| Les éléments publics de la couche « Modèle » doivent être documentés.                |
| Tout traitement lourd doit être effectué via une tâche asynchrone.                   |
| Une rétroaction visuelle doit être affichée lors de traitements lourds.              |
| L'interface de l'application doit supporter l'orientation portrait et paysage.       |

|   |
|---|
| L'application doit être traduite en français et en anglais.                                     |
| Les erreurs possibles doivent être gérées par l'application, avec des messages d'erreur clairs. |
| Le patron « Repository » est obligatoire pour créer la couche d'accès à la base de données.     |
| L'usage de requêtes préparées (« PreparedStatements ») est obligatoire.                         |
| L'usage de « Parcelables » est obligatoire lorsque possible.                                    |

## 4 Modalités de remise

Remettre sur LÉA un fichier texte incluant :

1. L'adresse vers le dépôt Git
2. Les matricules de chaque membre de l'équipe

Une seule équivalut à une pénalité de 15 %. Au-delà de ce délai, le travail la note « 0 » est attribuée.

## 5 Évaluation

| Éléments   |
|--|
| <b>Fonctionnalités</b> (incluant, mais sans s'y limiter) : <ul style="list-style-type: none"> <li>• L'application fait ce pour quoi elle a été conçue.</li> <li>• L'application ne bogue pas.</li> </ul>   |
| <b>Interface utilisateur</b> (incluant, mais sans s'y limiter) : <ul style="list-style-type: none"> <li>• Fichiers XML d'interface propres et bien structurés.</li> <li>• Visuel de l'interface utilisateur propre et stable.</li> <li>• L'interface doit être disponible en anglais et en français.</li> </ul>  |
| <b>Base de données</b> (incluant, mais sans s'y limiter) : <ul style="list-style-type: none"> <li>• Utilisation du patron de conception « Repository ».</li> <li>• Utilisation de requêtes préparées.</li> <li>• Toutes les erreurs possibles sont gérées correctement, rétroaction incluse.</li> </ul>  |
| <b>Documentation du code</b> (incluant, mais sans s'y limiter) : <ul style="list-style-type: none"> <li>• Les éléments de la couche « Modèle » doivent être documentés intégralement.</li> </ul>   |
| <b>Qualité générale du code</b> (incluant, mais sans s'y limiter) : <ul style="list-style-type: none"> <li>• Logique bien pensée, juste, et « non patché ».</li> <li>• Découpage adéquat du code en classes, méthodes et packages.</li> <li>• Architecture adéquate au type d'application développée.</li> <li>• Séparation raisonnable des responsabilités entre les classes.</li> <li>• Respects des standards du langage de programmation.</li> <li>• Nommage clair et sans ambiguïté des éléments.</li> <li>• Utilisation de constantes, lorsque nécessaires.</li> <li>• Commentaires compensant uniquement le manque d'expressivité du code.</li> <li>• Respect des bonnes pratiques de programmation générales.</li> <li>• Code propre, sans résidus et facilement lisible.</li> <li>• Aucune erreur ni avertissement à la compilation.</li> </ul> |

|  |
|--|
| <p><b>Qualité générale de l'application mobile</b> (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> <li>• Changement d'orientation et d'application supporté, sans perte de données.</li> <li>• Traitements lourds toujours effectués sur une tâche asynchrone.</li> <li>• Présence de rétroaction visuelle lors de traitements lourds.</li> <li>• Utilisation de « Parcelables » lorsque possible.</li> </ul> |
| <p><b>Qualité générale du travail</b> (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> <li>• Configuration Gradle fonctionnelle.</li> <li>• Respect des consignes de remise.</li> <li>• Aucun fichier inutile remis avec le projet.</li> <li>• Travail remis via l'outil gestion de code source Git.</li> </ul>  |
| <p><b>Présentation orale du travail</b> (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> <li>• Qualité des propos.</li> <li>• Qualité du support visuel.</li> </ul>  |

La qualité de la langue française fait partie de l'évaluation. Chaque faute de français retire 0,5 % à la note finale jusqu'à concurrence de 20 %.