

Énoncé du Travail Pratique 3

- Remise 2 -

28 avril 2017

Préparé par
Benjamin Lemelin

1 Résumé du travail

L'application à développer est libre. Cependant, vous devrez respecter certaines contraintes : avoir une base de données d'une bonne envergure et utiliser une technologie (ou technique) non vue en classe.

2 Conditions de réalisation du travail

Valeur sur la note finale	Type	Durée	Nombre de remises
30 %	En équipe	5 semaines	3

3 Travail à effectuer – Remise 2

Cette section vise à vous présenter les différentes étapes du travail à effectuer pour cette première remise.

3.1 Développer l'application

Développez l'application analysée à la remise 1. Documentez (avec de la JavaDoc) uniquement la couche modèle de votre application.

Comme toujours, créez un dépôt Git pour votre travail. Les « User Stories » doivent être inscrites dans la section « Issues » de votre dépôt et mises à jour régulièrement.

4 Modalités de remise

Remettez sur LÉA à l'heure et à la date indiquée par votre professeur un fichier texte comportant l'adresse vers votre dépôt Git. Cette adresse sera utilisée pour effectuer un « git clone ». Ajoutez à ce fichier le nom et le matricule de tous les membres de votre équipe ainsi que tout commentaire pertinent à la correction. Enfin, le nom du fichier texte doit être vos matricules, séparés par « _ », suivis de « .txt ». Une seule journée de retard est tolérée entraînant alors une pénalité de 15 % de la note. Au-delà de ce délai, le travail est refusé et la note « 0 » est attribuée.

5 Évaluation

Ce travail sera évalué avec une correction normale et une correction négative. Les points seront retirés en fonction de la gravité de la faute, de la qualité globale du travail et de l'effort mis par l'étudiant. Le tableau suivant vise à résumer les éléments pouvant réduire la note, jusqu'à atteindre la note de 0 %.

Éléments	Pondération
Conception de l'application : <ul style="list-style-type: none"> Document d'analyse insatisfaisant. (017D-2) Propos impertinents ou inutiles. (017D-2) Aucune nouvelle technologie abordée dans le travail. (017D-2) Récits utilisateurs mal formulés. (0176-2) Récits utilisateurs comportant des détails d'implémentation. (0176-2) Récits utilisateurs mal découpés. (0176-2) Fonctionnalité prévue sans récit utilisateur correspondant. (0176-2) Mauvaise utilisation de l'agrégation. (016T-2) Mauvaise utilisation de la composition. (016T-2) Découpage insuffisant des classes ou interfaces. (016T-2) Non-respect des standards UML. (016T-2) À REMPLIR DANIEL UML. (016T-2) À REMPLIR DANIEL UML. (016T-2) À REMPLIR DANIEL UML. (016T-2) À REMPLIR DANIEL UML. (016T-2) À REMPLIR DANIEL UML. (016T-2) À REMPLIR DANIEL UML. (016T-2) 	20 % Correction négative
Base de données (incluant, mais sans s'y limiter) : <ul style="list-style-type: none"> Non utilisation de requêtes préparées. (017D-5, 017D-6) Connexion à la base de données non fermée. (017D-6) Obtention des données non faite en tâche de fond. (017D-5) Mauvaise gestion des erreurs provenant de la base de données. (017D-5) Non utilisation du pattern « Repository ». (016T-1) Non utilisation de « Mocks » pour les « Repositories » durant les tests. (0177-2, 0177-3) 	10 % Correction négative
Interface utilisateur (incluant, mais sans s'y limiter) : <ul style="list-style-type: none"> Non-respects des standards d'interface des plateformes mobiles. (017D-4) Langue française non supportée. (017D-4) Langue anglaise non supportée. (017D-4) Non-respect de la grille de 8dp. (017D-4) Éléments d'interface mal alignés. (017D-4) Non utilisation d'images vectorielles. (017D-4) Non-utilisation de « Snackbar » pour les messages. (017D-4) Fichier XML de « Layout » Android malpropre. (017D-4) « Layouts » Android instables sous différentes tailles et densités d'écran. (017D-4) Expérience utilisateur désagréable. (017D-4) Perte de données lors d'un changement d'orientation. (017D-3, 017D-4) 	10 % Correction négative

<p>Documentation du code (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> • Classe non documentée ou documentée de manière insuffisante. (017D-7) • Méthode non documentée ou documentée de manière insuffisante. (017D-7) • Paramètres non documenté ou documenté de manière insuffisante. (017D-7) • Valeur de retour non documenté ou documenté de manière insuffisante. (017D-7) • Lien non présent vers une classe référencé dans la documentation. (017D-7) • Documentation malpropre. (017D-7) • Indentation incorrecte de la documentation. (017D-7) • Documentation qui n'est pas en français. (017D-7) • Standard de documentation « JavaDoc » non respecté. (017D-7) 	<p>5 % Correction négative</p>
<p>Tests unitaires (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> • Fonctionnalité d'un contrôleur non testée par un test d'interface. (0177-1, 0177-2, 0177-3) • Fonctionnalité d'une vue non testée par un test d'interface. (0177-1, 0177-2, 0177-3) • Fonctionnalité de la couche modèle non testée unitairement. (0177-1, 0177-2, 0177-3) • Test pas suffisamment robuste. (0177-2) • Test vérifiant plus qu'une fonctionnalité. (0177-1) • Vérification échouant dans un test unitaire. (0177-1, 0177-2, 0177-3) • Vérification manquante dans un test. (0176-4, 0177-1) • Utilisation de la mauvaise vérification « JUnit ». (0177-2) • Utilisation de la mauvaise vérification « Espresso ». (0177-2) <p><i>Sont exclus de la correction :</i></p> <ul style="list-style-type: none"> • Test des appels réseau. • Test des appels à la base de données (les « Repositories »). • Test où le résultat est indéterministe. 	<p>10 % Correction négative</p>
<p>Qualité générale de l'architecture (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> • Contrôleur effectuant de la logique réservée à la couche modèle. (017D-3, 017D-5) • Séparation incorrecte entre la vue et le modèle. (017D-3, 017D-5) • Abstractions non utilisées où cela aurait été nécessaire. (016T-1) • Référence concrète où une référence abstraite aurait été suffisante. (016T-3) • Pas d'usage de l'héritage où cela aurait été nécessaire. (016T-1, 016T-3) • Non-respect du diagramme de classe. (016T-2) • Méthode ayant trop de responsabilités pas découpées convenablement. (016T-3) • Classe ayant trop de responsabilités pas découpées convenablement. (016T-3) • Découpage en « package » insuffisant. (016T-2) • Nom d'un « package » imprécis. (016T-3) • « dot.case » non utilisé pour un nom de « package ». (016T-3) 	<p>15 % Correction négative</p>

Qualité générale du code (incluant, mais sans s'y limiter) : <ul style="list-style-type: none"> Logique applicative « patchée ». (016T-3) Nom d'une variable imprécis. (016T-3) Nom d'une classe imprécis. (016T-3) Nom d'une interface imprécis. (016T-3) Nom d'une constante imprécis. (016T-3) Nom d'une énumération imprécis. (016T-3) Nom d'une méthode imprécis. (016T-3) « camelCase » non utilisé pour un nom de variable ou de méthode. (016T-3) « PascalCase » non utilisé pour un nom d'interface, de classe ou d'énumération. (016T-3) « UPPER_CASE » non utilisé pour un nom de constante. (016T-3) Non-respect des standards d'indentation Java. (016T-3) Manque d'une constante ou cela aurait été nécessaire. (016T-3) Attribut ou méthode sans modificateur de visibilité (ou avec le mauvais). (016T-3) Méthode publique qui devrait être privée. (016T-3) Méthode statique qui ne devrait pas l'être. (016T-3) Méthode protégée qui ne devrait pas l'être. (016T-3) Attribut de classe qui n'est utilisé que dans une seule méthode. (016T-3) Commentaire inutile ne compensant pas le manque d'expressivité du code. (016T-3) Erreur à la compilation. (016T-3, 016T-5) Avertissement à la compilation. (016T-3, 016T-5) 	20 % Correction négative
Qualité générale du travail (incluant, mais sans s'y limiter) : <ul style="list-style-type: none"> Corruption de la configuration Gradle des projets. (016T-5) Non-respect des consignes de remise. (016T-5) Document d'analyse remis malpropre. (016T-5) Fichiers inutiles remis avec les projets. (016T-5) Travail non remis via l'outil gestion de code source Git. (0176-3) 	10 % Correction négative

Notez que la qualité écrite de la langue française fait partie intégrante de l'évaluation de ce travail. Étant donné que l'accès aux ressources de la langue française est permis, chaque faute de français retirera donc 0,5 % à la note finale jusqu'à concurrence de 20 % retirés. Sont corrigées les fautes d'orthographe, les fautes de grammaire, les fautes syntaxiques et les fautes de ponctuation.

L'exposé oral à la fin de ce travail 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.

6 Compétences liées au travail pratique

Voici la liste des compétences évaluées, évaluées partiellement ou non-évaluées par ce travail. Vous trouverez les explications sur ces choix en dessous de cette liste.

- 016T-1 Créer un modèle objet : Évalué.
- 016T-2 Perfectionner le modèle objet : Évalué.
- 016T-3 Procéder à la codification d'une classe : Évalué.
- 016T-5 Générer la version exécutable du programme : Évalué.
- 0176-1 Analyser les caractéristiques de l'application : Évalué.
- 0176-2 Analyser la nature des améliorations à apporter à l'application : Évalué.
- 0176-3 Ajouter et modifier des fonctionnalités à une application : Évalué.
- 0176-4 Valider le fonctionnement de l'application : Évalué.
- 0177-1 Planifier les tests : Évalué.
- 0177-2 Procéder à l'exécution des différents tests : Évalué.
- 0177-3 Valider la qualité de l'application : Évalué.
- 017D-2 Établir le cadre technologique : Évalué.
- 017D-3 Préparer le travail de développement : Évalué.
- 017D-4 Produire le prototype de présentation : Évalué.
- 017D-5 Produire le prototype de communication : Évalué.
- 017D-6 Développer l'application : Évalué.
- 017D-7 Produire la documentation relative à l'application : Évalué.