

Énoncé du Travail Pratique 3

3 février 2017

Préparé par
Benjamin Lemelin

1 Résumé du travail

L'application à développer doit permettre de créer une liste de tâches à faire. Vous aurez à effectuer l'analyse des besoins et la conception de l'interface utilisateur.

2 Conditions de réalisation du travail

Valeur sur la note finale	Type	Durée	Nombre de remises
30 %	En équipe	2 semaines	1

3 Travail à effectuer

Cette section vise à vous présenter les différentes étapes du travail à effectuer.

3.1 Analyser les besoins

Contrairement aux autres travaux, vous devrez déterminer vous-même vos fonctionnalités. Votre mandat est de réaliser une application « Tâches à faire » : à vous de voir comment elle fonctionnera et ce qu'elle permettra de faire. Prenez le temps d'analyser sérieusement le fonctionnement de votre application pour créer un produit de qualité professionnelle. Notez que, puisque c'est un travail évalué, il y a une liste de fonctionnalités minimales à respecter. Consultez la section 3.2.

Vous utilisez l'approche par Persona pour identifier des personnages qui vous serviront d'utilisateurs. En temps normal, cette approche demande à ce qu'un sondage soit fait afin d'identifier des profils d'utilisateur et ainsi créer des personnages crédibles. Cependant, étant donné le peu de temps disponible, créez tout simplement des profils crédibles et servez-vous-en pour faire vos personnages.

Récupérez le gabarit fourni avec cet énoncé et suivez les sections suivantes. Chaque section **en jaune** dans le gabarit doit être remplacée par votre propre texte.

3.1.1 Identifier les profils d'utilisateurs (au moins 8)

Identifiez des profils d'utilisateurs de votre application. Un profil est une caractéristique d'un utilisateur. Prenons l'exemple d'une application de traitement d'image. Vous pourriez avoir :

1. « Le programmeur » travaille en collaboration avec des artistes. Il ne connaît rien en retouche d'image en dehors de leur redimensionnement qu'il fait automatiquement en traitement par lot.
2. « Le créatif » a beaucoup d'imagination. Il fait beaucoup de dessins et ses principaux outils sont le crayon, la gomme à effacer et le papier calque.
3. « L'étourdi » fait beaucoup d'erreurs. Il ne se passe pas une minute sans qu'il fasse quelque chose qu'il regrettera par la suite.
4. « Le photographe » est un habitué de la retouche photo. Ses clichés doivent paraître impeccables et plus beaux que nature.

3.1.2 Regrouper les profils d'utilisateurs (au moins 4)

Regroupez les profils d'utilisateurs en fonction de leur objectif final. Demandez-vous ce qu'ils voudraient que l'application fasse pour eux en tout premier lieu. Par exemple :

1. « Le créatif » et « Le photographe » ont tous les deux besoin d'un moyen d'exprimer leur vision des choses. Les outils qu'ils utilisent doivent être complets et performants.
2. « L'étourdi » a besoin d'un moyen d'annuler ses erreurs.
3. « Le programmeur » a besoin d'un moyen simple et rapide de faire quelques ajustements basiques sur une grande quantité de fichiers.

3.1.3 Créer des personnages (au moins 4)

Prenez chaque regroupement de profils et faites-en un personnage. Trouvez-leur un nom et une image pour les représenter. Par exemple :

1. « Alexandre Lachance » est caricaturiste au journal « Le Gros Titre ». Il adore inclure dans ses dessins des images d'actualité qu'il ...
2. « Félix Bouchard » est un père de famille avec 4 enfants. Il adore les prendre en photo, mais il déteste enlever les yeux rouges dessus. Il doit souvent s'y prendre à plusieurs reprises avant de...
3. « William Dumont » est programmeur chez « Infoweb ». Il n'a jamais touché à un logiciel de traitement d'image de sa vie et s'en tient loin le plus possible. La seule chose qu'il fait de temps en temps, c'est redimensionner les images bien trop grosses que lui fournissent les graphistes. Encore là, il possède un script qui ...

3.1.4 Catégoriser les personnages

Identifiez les types de personnages. Vous aurez le personnage principal pour qui l'application sera conçue, le(s) personnage(s) secondaire(s) pour qui certains ajouts seront faits et l'anté-personnage pour qui l'application ne sera pas conçue. Par exemple :

1. « Alexandre Lachance » est le personnage principal, car c'est celui qui a le plus de contraintes : il doit pouvoir dessiner, ajouter des images, les modifier, etc...
2. « Félix Bouchard » est un personnage secondaire. Il a besoin d'un moyen d'annuler ses actions.
3. « William Dumont » est l'anté-personnage. Il n'a besoin que d'un outil pour effectuer du traitement d'images en lot.

3.1.5 Écrire les récits utilisateurs

Écrivez les récits utilisateurs en vous servant du personnage principal et du personnage secondaire. Prenez garde à ce qu'aucun récit utilisateur ne vise votre anté-personnage.

3.2 Développer l'application

3.2.1 Fonctionnalités minimales

No	Description	OK
1	Consulter les tâches.	
2	Ajouter une tâche.	
3	Supprimer une tâche.	
4	Ajouter une étiquette à une tâche.	
5	Regrouper les tâches par étiquette.	

3.2.2 Critères d'acceptabilité de la conception

No	Description	OK
1	L'application doit supporter l'orientation portrait et paysage.	
2	L'application doit respecter les standards de la plateforme au niveau de l'interface et des interactions avec elle.	

3.2.3 Directives techniques

No	Description	OK
1	L'application doit utiliser le style architectural « MVVM ».	
2	Le package de l'application doit être « ca.csf.mobile2.tp3 ».	
3	Conservez les données sur l'appareil en utilisant une base de données « SQLite ».	
4	Utilisez le patron de conception « Repository » pour l'accès à la base de données.	
5	Vous devez effectuer de tests d'interface Espresso pour vos couches d'interface utilisateur ainsi que des tests unitaires pour votre couche modèle.	
6	Vous devez utiliser les bibliothèques « Android Annotations », le « Databinding » Android, « Android Databind List » et « Dagger 2 ».	
7	Vous devez utiliser un dépôt de code sous Git et en donner l'accès à votre professeur.	
8	Vous devez placer les « User Stories » dans la section « Issues » de votre dépôt Git.	

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

Le travail sera évalué selon les critères suivants. Des points seront perdus en fonction de la gravité de la faute, de la qualité globale du travail et de l'effort mis par chaque membre de l'équipe dans ce travail.

Éléments	Pondération
Fonctionnalités et expérience utilisateur (incluant, mais sans s'y limiter) : <ul style="list-style-type: none">Ajout/suppression/étiquetage/regroupement de tâches. (017C-1, 017D-1, 017D-4)Support de l'orientation portrait et paysage. (017C-3, 017D-4)Respect des standards de la plateforme mobile. (017D-4)	5 %
Approche par Persona (incluant, mais sans s'y limiter) : <ul style="list-style-type: none">Profils et personnages identifiés correspondant à l'application. (017D-1)Récits utilisateurs établis bien rédigés et adaptés à l'application. (017D-1)Qualité de l'investissement dans la réalisation de l'approche par Persona. (017D-1)	25 %
Base de données (incluant, mais sans s'y limiter) : <ul style="list-style-type: none">Utilisation d'une base de données adaptée à une plateforme mobile. (017B-2)Schémas de base de données solide et efficace. (017B-2)Utilisation correcte du patron de conception « Repository ». (017C-3, 017C-4)	10 %
Style architectural MVVM (incluant, mais sans s'y limiter) : <ul style="list-style-type: none">Utilisation convenable du style architectural MVVM. (017C-3)Utilisation du « Data Binding » Android. (017C-3, 017C-4, 017D-6)	5 %
Qualité générale du code (incluant, mais sans s'y limiter) : <ul style="list-style-type: none">Logique applicative bien conçue et non « patchée ». (017C-4)Bon nommage des variables, classes, constantes, énumérations et fonctions. (017C-4)Utilisation des constantes, lorsque nécessaires. (017C-4)Classes et fonctions n'ayant qu'une seule responsabilité. (017C-4)Utilisation convenable des « packages ». (017C-4)Utilisation de la composition et de l'héritage, lorsqu'approprié. (017C-4)Commentaires compensant le manque d'expressivité du code. (017D-7)Aucune erreur ni avertissement à la compilation. (017C-4)	20 %
Architecture générale (incluant, mais sans s'y limiter) : <ul style="list-style-type: none">Utilisation de « Android Annotations » pour les activités Android. (017C-3, 017C-4)Utilisation de « Dagger 2 » pour l'injection de dépendances. (017C-3, 017C-4)Séparation adéquate entre l'interface et le domaine d'affaires. (017C-3)	15 %

Tests de l'application (incluant, mais sans s'y limiter) :		
• Utilisation de « Espresso » pour tester toutes les interfaces utilisateur.	(017C-4)	15 %
• Couverture complète de la couche « Modèle » par des tests unitaires.	(017C-4)	
• Propreté des tests unitaires.	(017C-4)	
Qualité générale du travail (incluant, mais sans s'y limiter) :		
• Respect des consignes de remise.	(017B-6)	5 %
• Bonne utilisation des outils de gestion de code.	(017B-6)	
• Suivi régulier des « User Stories » via avec l'outil de gestion du code.	(017B-6)	

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.

6 Compétences reliées 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.

- 017B-2 Établir les besoins technologiques : Évalué.
- 017B-6 Développer les programmes : Évalué.
- 017C-1 Établir le cadre général de l'application : Évalué.
- 017C-3 Modéliser l'application : Évalué.
- 017C-4 Programmer l'application : Évalué.
- 017D-1 Établir les fonctionnalités de l'application : Évalué.
- 017D-4 Produire le prototype de présentation : Évalué.
- 017D-5 Produire le prototype de communication : Non évalué.
- 017D-6 Développer l'application : Non évalué.
- 017D-7 Produire la documentation relative à l'application : Évalué.

Les compétences 017D-5 et 017D-6 ont été évaluées dans les travaux précédents.