Master 1 Info - UE TAC 2022-2023

Evaluation Android

1. Préambule

Cette évaluation est à faire à la fois en binôme et de façon individuelle pour certaines parties (note individuelle). Par ailleurs, même si l'application est simple, un résultat de qualité professionnelle est attendu, que ce soit dans le code, les commentaires, l'IHM, les documents associés à l'application, le GIT, etc. La note dépend du respect des consignes énoncées dans ce document.

Note : dans certains cas, et en accord préalable par mail avec l'enseignant de TD, ce travail peut être réalisé seul. L'évaluation reste cependant exactement la même que pour un travail en binôme.

2. Sujet

Votre application a pour objectif de consulter des informations accessibles sur le web au travers d'une API, et d'en stocker une partie en local. Vous avez le choix de l'API, excepté celles vues en cours. **Pensez à trouver rapidement cette API et à la communiquer à jean-claude.tarby@univ-lille.fr !**

Cette API permet de récupérer une liste d'informations (par exemple une liste de livres avec le détail de chaque livre sélectionné, comme vu en cours,). Vous afficherez ces informations sous forme de liste et de grille (cf. maquettes écran plus loin). Le passage d'un affichage à l'autre est laissé à votre discrétion (rotation de l'écran, bouton, menu...), de même que le look de ces affichages, mais pensez "application professionnelle" pour tout cela. Vous devrez afficher à la fois du texte et des images comme dans les maquettes montrées dans ce document, et afficher pour chaque item de la liste un ou deux boutons qui permettent des actions telles que supprimer, partager, mettre en favori, etc. (à vous de choisir la ou les actions que vous voulez implémenter).

Lorsque l'utilisateur sélectionne un élément de la liste/grille, l'application affiche le détail de cet élément sur un autre écran (vous avez le choix des infos et du look), toujours de façon professionnelle...

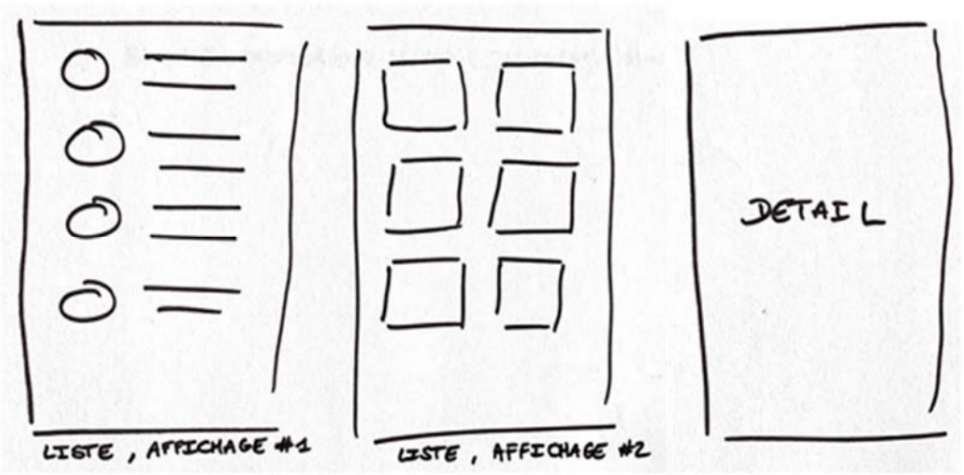
Par ailleurs, vous devrez stocker une partie des informations renvoyées par l'API dans une base de données locale avec Room. Ces informations peuvent être par exemple les détails des éléments consultés, une gestion de favoris, etc.

Vous pouvez aller au-delà de ce cahier des charges si le coeur vous en dit, par exemple :

- en vérifiant si les données demandées sont déjà en local avant de requêter l'API

- en vérifiant les timestamps des données distantes pour mettre à jour éventuellement les données locales,
- etc.

Cependant, cela ne sera pas pris en compte dans la note finale, c'est juste pour votre plaisir.



Maquettes écran simplifiées de l'application demandée

3. Consignes à respecter

3.1. Consignes pour l'application (en résumé)

- récupération des données sur une API autre que celles vues en cours, et dans la mesure du possible différente des autres étudiants
- affichage des items en liste et en grille
- passage de l'affichage de liste à grille, et réciproquement
- en mode liste/grille : affichage de texte(s)+image(s)+bouton(s) d'action(s)
- stockage de données en local
- "look and feel" professionnel et esthétique, qui respecte les standards d'Android.
- programmation professionnelle (pour le code, les commentaires et le GIT)

3.2. Consignes pour le rendu

- **Dates de rendu : avant le 15 janvier 2023**.
- Vous remettrez votre travail sous la forme d'un fichier `NOM_Prénom.zip` (pas d'autre format autorisé !), par exemple `DUPONT_Philippe.zip`, que vous enverrez par l'intermédiaire de https://filesender.renater.fr/ (en vous connectant avec votre identifiant Université de Lille).
    - Attention à ne pas laisser un espace vide après votre identifiant dans filesender sinon ça ne marche pas !
- Suivant vos groupes de TD, les envois se feront à jean-claude.tarby@univ-lille.fr, ou cedric.dumoulin@univ-lille.fr (conservez la preuve d'envoi datée)
- La durée de validité de votre dépôt filesender devra être d'au moins un mois.
    - La date de dépôt de FileSender fera foi en cas de litige. Pensez à anticiper les problèmes de connexion internet ...

- Votre fichier ZIP contiendra 5 éléments :
    1 - Le fichier APK signé que vous placerez dans un dossier séparé 'apk'. Précisez sur quelle version d'Android vous avez développé (8.0,10.1...).
    2 - Un lien vers le repository du projet ACCESSIBLE, pour qu'on puisse le cloner et le tester (pensez à vérifier l'accessibilité de votre projet !)
    3 - Un dossier contenant UNIQUEMENT les fichiers sources de :
        o l'activité principale,
        o un adaptateur de recyclerview,
        o et un fragment
    4 - Un dossier contenant 3 screenshots significatifs de l'application
    5 - La documentation technique en pdf (vérifier qu'elle s'affiche correctement !)

Pour rappel :

- **L'absence d'un des 5 livrables ci-dessus dans le rendu final entraînera une pénalité de 5 points sur la note finale par fichier manquant, en plus des points non attribués si certaines consignes ne sont pas respectées.**
- **Aucun rendu ne sera accepté en retard (d'une minute, une heure ou un jour ...).**

3.3. Consignes pour la documentation technique

La documentation technique doit être, elle aussi, de qualité professionnelle. Cela inclut, entre autres :

- une page de garde avec vos noms et prénoms, un titre, votre (ou vos) Master(s), votre promotion (2022-2023), le logo de l'Université
- une table des matières/un sommaire (avec les n° de pages...)
- des pages numérotées...
- un style français correct (pas de style 'parlé' !)
- pas de faute de grammaire et d'orthographe

Elle doit être rédigée de façon individuelle (cf. plus loin) et contient un diagramme de séquences, un diagramme de classe/package, un contrat d'interface, et une partie explicative.

- Le *diagramme de séquences* doit présenter l'algorithme de récupération et de sauvegarde des données de l'activité principale. Il doit permettre à quelqu'un ne connaissant ni le cahier des charges ni l'application de comprendre le fonctionnement de cette partie de l'application.
    - Il doit donc être lisible, simple à comprendre, et avec des liens éventuels avec le code (noms de méthodes, hyperliens,...)
- Le *diagramme de classes/packages* doit présenter le projet en mettant en valeur d'un côté les objets métier de l'application, et de l'autre les classes liées à Android.
    - Il doit permettre de comprendre le domaine métier et les concepts manipulés par l'application.
    - Il doit être lisible. Ne faites pas simplement un 'reverse engineering' du code !
    - Vous pouvez utiliser des couleurs, ou tout autre artifice graphique lisible, pour montrer cette séparation métier/Android.

- Le *contrat d'interface* doit définir les éléments dont votre application a besoin pour fonctionner en termes d'API.
    - Pour cela, vous devez présenter les différentes requêtes (donnez des explications de ce qui est requêté, les paramètres, le type GET/POST) que l'application effectue ainsi que les retours (formatés) que vous attendez (nom, type, optionnel, etc...). Vous pouvez utiliser Swagger par exemple.
- Une partie de la documentation contiendra des explications sur :
    - les choix qui ont été faits pour le projet (quel que soit le sujet de ces choix),
    - les librairies choisies,
    - les concepts utilisés,
    - le fonctionnement général de l'application,
    - le parc matériel ciblé (API mini et cible),
    - et toute autre information que vous jugerez utile.

Un conseil : ne la commencez pas à la fin du travail. Une documentation bien faite est écrite avec du temps, et cela vous rapportera facilement des points...

3.4. Consignes pour le travail personnel

- Vous devez utiliser un seul dépôt git (que ce soit pour les binômes et les monômes) pour le rendu (le gitlab de l'Université de préférence, mais gitlab.com et github.com sont autorisés).
- Le dépôt git devra être créé dès le début du semestre (pas la veille du rendu ou quelques jours avant !!!) et on doit pouvoir suivre l'évolution du projet, c'est-à-dire constater un réel historique de travail cohérent dans les commits, et ceci de la part de chacun des participants du projet.
    - on doit pouvoir consulter l'arbre des branches et des commits. Un seul commit entraînera la note de 0 (zéro) sans autre discussion. Le nombre de commits, leurs qualités, mais aussi les branches, les merges, etc. entreront en ligne de compte pour la notation. Par exemple, 5 commits sont largement insuffisants pour ce projet...
- Dès le début du projet, vous déposerez sur le git un tableau de planification et de répartition des tâches (tableau excel ou autre, kanban,...) qui permettra de savoir qui fait quoi dans le projet. Si ce tableau est amené à changer durant le projet, on doit le voir et pouvoir consulter l'historique des mises à jour (avec explication de ces changements de planification et de tâches).
    - pour info, vous avez des Kanban disponibles par https://ent.univ-lille.fr/applications ou https://kanban-peda.univ-lille.fr/
- Chaque personne du binôme doit intervenir de façon INDIVIDUELLE et PERSONNELLE sur le projet. Cela signifie que même si vous travaillez en binôme, on doit pouvoir différencier le travail de chacun (les commits, les merges, etc.).
- Vous devez toujours utiliser le même pseudo sur le dépôt git.
    - Les excuses du style "on a travaillé ensemble, mais sur la même machine, alors on a commité a partir du même compte" ne marcheront pas !
- Chaque personne du binôme devra rendre sa propre documentation technique rédigée de façon individuelle et personnelle.
- Chaque chargé de TD se réserve le droit de faire faire des soutenances à certaines personnes après les rendus pour vérifier des points particuliers.

4. Grille de notation

Grille de notation pour l'évaluation Android		
	Barème	
Application		
Respect du cahier des charges et des consignes	30	pts
Absence d'anomalies	40	pts
Qualité de l'architecture	15	pts
Documentation du code	15	pts
Look professionnel et Android feel	20	pts
Documentation technique		
Diagramme de classe/package	10	pts
Diagramme de séquence	10	pts
Contrat d'interface	10	pts
Travail personnel		
GIT, travail en TD...	20	pts
Total	**185**	**pts**

remis sur 20 points pour la note finale avec note calculée comme indiquée à https://www.fil.univ-lille.fr/portail/index.php?dipl=MInfo&sem=ES&ue=TAC&label=%C3%89valuation