

Projet d'examen

Développement d'applications mobiles - TP

Daniel Schreurs <daniel.schreurs@hepl.be>

2022 – 2023

1 Introduction

Ce travail est demandé dans le cadre du cours de "Développement d'applications mobiles".

L'objectif de ce projet est de vous donner un cadre pour développer une application mobile en Flutter afin d'asseoir les concepts vus en cours. Ce projet est aussi une occasion de réfléchir à l'expérience utilisateur sur les petits et moyens écrans.

1.1 Procédure

La plateforme GitHub permet de publier le code source de vos projets. Elle doit être utilisée dès le début et régulièrement durant toute la réalisation de celui-ci. Nous aimerions ainsi vous habituer à l'outil de gestion de versions git. Nous voulons vous faire ressentir la facilité qu'il amène. Vous devez comme pour les autres projets accepter la mission dans la section *projets* de la page officielle du cours.

2 Objectifs

Vous devez dans le cadre de ce projet réfléchir à un besoin pour un utilisateur final afin de répondre à celui-ci au travers d'une application mobile. Vous êtes libre de choisir ce besoin. En revanche, celui-ci devra faire l'objet d'une présentation orale en classe ainsi qu'une validation par le professeur. Nous vous offrons une certaine liberté pour mettre à profit votre créativité et les connaissances que vous avez acquises durant ce bachelier. Ce projet met aussi en avant l'approche agile. C'est-à-dire qu'il est attendu que vous fournissiez un travail régulier pendant l'année afin que je puisse vous orienter vers des pistes d'améliorations.

2.1 Documentation

Votre dépôt doit obligatoirement contenir un fichier `readme.md`. Celui-ci contiendra au moins :

- Une présentation des principaux dossiers de votre dépôt. Quelles sont les différentes ressources qu'il contient? Par exemple, les maquettes, le code, etc.;
- Une présentation de votre application. Ce dernier répond à un besoin, présentez-le. Ne faites aucune hypothèse sur le niveau de connaissances de votre lecteur. Vous vous adressez ici à un internaute quelconque qui découvre votre dépôt. Évitez un jargon technique dans cette partie de votre présentation;
- Une brève étude de l'existant. Est-ce que d'autres ont déjà couvert ce besoin et si oui en quoi allez-vous vous distinguer d'eux ? N'hésitez pas à renseigner ici les points forts et faibles que vous observez;
- Une présentation des différentes fonctionnalités de votre application au travers de récits utilisateurs (user story). Soit une description courte et simple d'un besoin ou d'une attente exprimée par un

utilisateur. Chacun de ces récits suivra la syntaxe "En tant que <qui>, je veux <quoi> afin de <pourquoi>" :

- Le *qui* indique l'utilisateur du point de vue duquel on se place. Il s'agit souvent d'un rôle. Par exemple "élève" ou "enseignant" dans le cas d'une application d'apprentissage. Pour mieux illustrer la diversité des besoins, on peut également utiliser le concept de persona, c'est-à-dire une personne fictive et représentative à laquelle on peut s'identifier pour mieux comprendre ses attentes. L'identification et la description des personas se fait alors avant de commencer l'écriture des récits utilisateurs. Par exemple, "Odile est une enseignante qui utilise pour la première fois le système";
- Le *quoi* décrit succinctement la fonctionnalité ou le comportement attendu. Le but du récit n'est pas d'en fournir une explication exhaustive;
- Le *pourquoi* permet d'identifier l'intérêt de la fonctionnalité et d'en justifier le développement. Il permet également de mieux évaluer la priorité des fonctionnalités.

Pour chacune de ces fonctionnalités, présentées par un récit utilisateur, vous présenterez les maquettes qui s'y rapportent;

- Un état d'avancement pour chaque fonctionnalité de votre application. Ceci doit évidemment être mis à jour régulièrement. Dès lors que vous aurez terminé de programmer une fonctionnalité, ajoutez dans le document readme un gif qui l'illustre. Vous pouvez vous servir de [GIF Brewery](#)¹ ou de [Gyazo](#)².

¹Si vous êtes sous macOS.

²Si vous êtes sous Windows.

2.2 Maquette

Afin de pouvoir se concentrer pleinement sur l'expérience utilisateur et l'identité graphique de votre application, il est nécessaire de produire des maquettes avant l'implémentation. Pensez à

- Renseigner de manière claire vos [Moodboard](#);
- Documenter à travers des commentaires ainsi que les user stories (cf. section [2.1](#)) les parties qui nécessitent des explications supplémentaires.
- Dans vos choix de conception, placez la personne au premier plan, afin de prendre en compte *ses besoins* avec *ses incapacités* permanentes, temporaires, contextuelles, ou changeantes - c'est à dire nous tous;³

2.3 Application

La partie qui vous demandera le plus de temps sera la conception de l'application en Flutter. Pour ce faire, il est nécessaire d'être à l'aise avec les projets réalisés en cours qui contiennent la plupart des parties dont vous avez besoin. Vous veillerez :

- À gérer l'identification ainsi que l'authentification de votre utilisateur;
- Un écran d'accueil qui *accueille* votre utilisateur et présente un aperçu des différentes ressources. Pensez à la première connexion. Comment faire pour que vos utilisateurs *comprennent* et *adoptent* votre application ? Soignez l'[onboarding](#) de ces derniers.
- Au moins un écran de détail qui permet d'afficher plus d'information sur la ressource.
- Mettre à jour des informations, par exemple son profil;
- Gérer la persistance des données avec Firebase ou une API existante;
- Exploiter les possibilités du temps réel que vous offre l'environnement des périphériques mobiles. Par exemple, les notifications.

3 Échéance et évaluation

La *remise* de votre projet se fera *automatiquement* une semaine avant la date de l'examen via GitHub. Concrètement, le dépôt que vous aurez généré ne serait plus accessible en écriture sur le serveur.⁴

La présentation de votre application se fera durant l'examen oral prévu à l'horaire. Durant cette présentation, vous aurez un premier temps de parole de maximum 15 minutes pour présenter votre application et un deuxième temps de 20 minutes pour répondre aux questions. Un horaire de passage vous sera communiqué sur la page Moodle. Vous devrez être présent au moins 1 heure afin d'anticiper des désistements.

³Il faut donc relire les [principes de conceptions inclusives](#).

⁴Ceci implique que vous ne pourrez plus faire de push vers le serveur Github. Ceci n'a aucun impact sur votre version locale.