

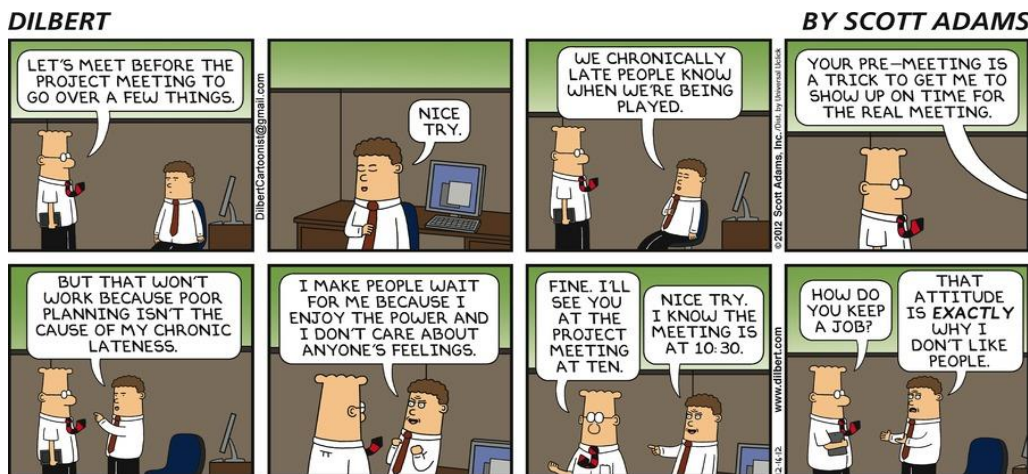
Objectifs du travail

- Utiliser une interface graphique sous Android
- Utiliser les View standards d'Android
- Utiliser les ressources Android pour gérer les diverses configurations
- Utiliser et traiter des événements reliés à l'écran tactile d'un appareil mobile
- Utiliser une base de données SQLite
- Se connecter à un API distant
- Synchroniser les données avec l'API distant
- Utiliser l'API de Google Maps.

Date de remise

Le mercredi 7 décembre 2022 à minuit: Remise finale

- Le projet Android Studio remis sur git avec un tag.
- Une auto-évaluation (une demi-page au format .md) de votre programme (ce qui fonctionne très bien, moins bien, pas du tout et pourquoi). Me dire s'il manque une fonctionnalité pour m'aider lors de la correction. Remis sur git.



Consignes

Voici le moment tant attendu du livrable 2! Pour ce livrable vous devrez :

- Corriger les bugs du livrable 1 (toutes les fonctionnalités du livrable 1 devront être implémentées correctement)
- Vous connecter à un service distant (API REST) :
 - Créer une page de connexion et une page de création de compte
 - Synchroniser les données régulièrement entre votre application et le serveur.
- Selon le type de compte connecté afficher une vue différente
- Convertir les id en uuid dans votre base de données SQLite (la mise à jour doit se faire sans supprimer les données déjà présentes dans la base, prévoir le script SQL adapté) classe : [java.util.UUID](https://docs.oracle.com/javase/8/docs/api/java/util/UUID.html)

Travail Pratique 2

- Vérifier à l'ouverture de l'application si le token de connexion est encore valide (ne pas redemander de s'identifier dans ce cas) le token de connexion ainsi que l'ID de compte doit être sauvegardé dans les préférences de l'application. Si le serveur n'est pas joignable ouvrir l'application en mode hors-ligne.

Spécifications techniques

- Version minimale du SDK supportée : Android avec SDK 28 – 9.0 (Pie)
- Utiliser les ressources Android (layout, strings, drawable etc.)
- Avoir une icône personnalisée (vous pouvez utiliser les sites suivants pour en générer une : <https://crello.com/features/icons/> ou <https://iconsflow.com/>).
- Avoir un nom d'application original
- BD SQLite depuis une classe dérivée de SQLiteOpenHelper

Spécifications graphiques

Vous devez minimalement permettre les actions suivantes:

- L'application devra bien gérer les layouts **portrait et paysage, pour cellulaire seulement**¹. Notez qu'un seul layout est possible si vous jugez l'apparence adéquate pour les différentes configurations d'orientation et d'écrans. N'oubliez pas de les tester !!!
- Consulter la liste de toutes les entreprises.
 - Ajout du filtre des favoris (L'utilisation de fragment est recommandée)
 - Ajout de la recherche (fonctionnelle sur la liste filtrée ou non)
- Pour le compte prof : Consulter la liste des Candidats et leur statistiques (nombres d'entreprises contactées et stage trouvé ou non)

Pour équipes de 2 seulement

1. La liste des étudiants doit permettre d'alterner entre deux tris différents : Tri alphabétique et tri par stage trouvé ou non.
2. Google Maps : Dessin de « pin » personnalisé (ne pas utiliser le dessin par défaut)
3. Ajout des tests unitaires selon la norme Android pour la partie connexion (un exemple vous sera fourni) les tests du livrable 1 doivent être maintenus fonctionnels.

Attention

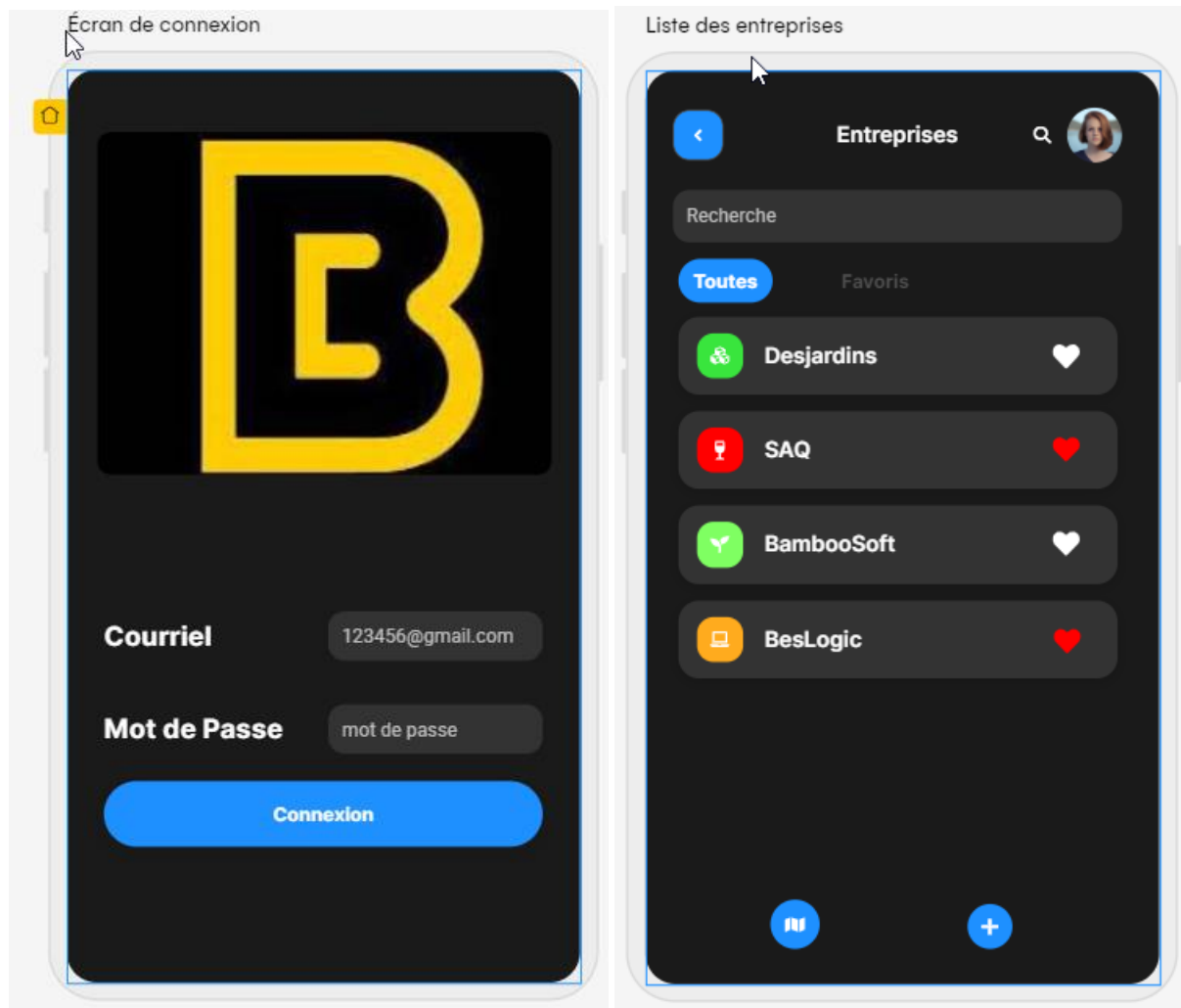
Vous devez porter une attention importante à l'ergonomie et au design de l'application. Référez-vous à des applications mobiles populaires pour reproduire des bonnes façons de faire. Beaucoup de points y sont assignés.

¹ Pour réussir une belle interface sur tablette, nous sommes quasi obligés d'utiliser les `Fragment`, concept qui n'a pas encore été abordé en classe.

Travail Pratique 2

N'hésitez pas à me consulter sur votre interface en cours de développement : même si je n'ai pas toutes les réponses, une bonne discussion permet souvent de trouver des idées intéressantes pour répondre aux problèmes rencontrés.


Voici des éléments de design pour vous aider à commencer :




Travail Pratique 2


Fiche Entreprise


← Annuler

 Desjardins

Contact Jean Dufour

Courriel jdufour@desjardins.com 


Telephone 514-555-5555 

Site web https://desjardins.com/annx 

Adresse 1234, rue des sapins


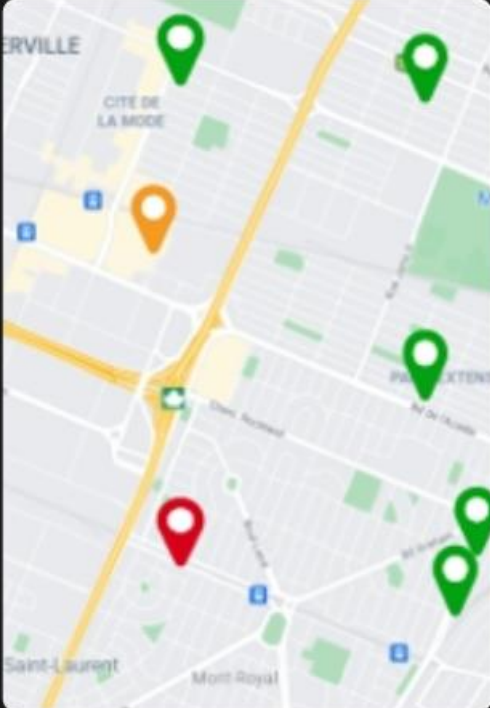
Ville Montréal

Code Postal H0H 0H0

Date de contact 2022/09/24 

OK

Carte



Travail Pratique 2

L'ergonomie de l'application compte pour 25% de la note du travail. Voici comment je noterai celle-ci, sur une échelle subjective :

Note moyenne prévue

1	2	3	4	5	6
Interface aléatoire, où l'interface nuit à l'utilisateur plutôt que de l'aider. Bugs critiques et/ou fonctionnalités absentes.	Interface partiellement complétée, ou présence de bugs critiques qui empêcheraient l'utilisateur de mener correctement toutes les actions désirées. Disposition « très garochée » tm des éléments graphiques.	Interface fonctionnelle qui pourrait contenir des bugs importants (mais non critiques). Interface fonctionnelle au look « garoché » tm .	Interface fonctionnelle qui pourrait contenir des petits bugs non critiques. Effort personnel adéquat mis sur l'interface, même si tout peut sembler un peu placé rapidement dans l'écran.	Interface réfléchie : Pas de bugs, certains trucs drôlement présentés mais on voit clairement un souci de donner à l'utilisateur une interface facile d'utilisation, qui tient compte des spécificités du type d'appareil.	Interface qui va au-delà des attentes, tant au niveau de l'esthétisme, de la facilité d'utilisation, du feedback utilisateur et du respect des standards de GUI sous Android. L'ergonomie a été considérée comme un élément égal en importance que le code derrière l'application.
30%	50%	60%	70%	80%	90%

Critères d'évaluation

Votre programme doit respecter les normes de programmation Android apprises.

Vous devez réaliser ce travail seul ou en équipe de deux. Vous devez informer l'enseignant pour approbation de l'équipe, s'il y a lieu.

Critère	Nombre de points accordés
Auto-évaluation	/5
Ergonomie / qualité de l'interface <ul style="list-style-type: none"> Utilisation de Layouts Adaptabilité, facilité d'utilisation et aspect visuel Respect des Guidelines Android Homogénéité entre fenêtres Sélection judicieuse des composants utilisés 	/25
Programmation <ul style="list-style-type: none"> Clarté, simplicité, adaptabilité Séparation code/vue avec l'aide des ressources Android Respect des normes de programmation orientée objet et Android 	/25
Fonctionnement du programme <ul style="list-style-type: none"> Résultats obtenus Respect des consignes 	/45
Total	/100

