



TP3

Informatique Mobile et Applications
GIF-3101

Tables des matières

TP3	1
Tables des matières	2
Objectifs	3
Introduction	3
Énoncé	3
Dossier de remise	5
Barème de correction	5
API	7
Aperçu	7
Schéma de navigation l'application	8

Objectifs

À la fin de ce travail pratique, l'étudiant devrait:

- Être en mesure de gérer une liste
- Être en mesure d'effectuer une requête GET sur un serveur
- Être en mesure d'effectuer une requête POST sur un serveur
- Être en mesure d'intégrer l'affichage d'image dans une interface
- Utiliser des libraires externes dans un projet

Introduction

Le but de ce troisième travail pratique dans le cours d'Informatique Mobile et Application (IMA) est connecter votre application à internet. Vous devriez par la suite être en mesure de faire des requêtes à des services web et de consommer des données.

Consignes

- Prenez le temps de lire l'énoncé au **complet** du travail pratique avant de débiter.
- Vous **devez obligatoirement** respecter les éléments de conformité qui sont énumérés sur le portail des cours dans la section ""Contenu et activités" -> Guides -> Conformité".
- Un guide de procédure de remise est disponible sur le portail des cours dans la section "Contenu et activités" -> Guides. Veuillez vous y référer pour vous assurer d'une remise conforme.
- Le travail doit être réalisé individuellement

Énoncé

1. Créez un nouveau projet Android à partir d'un "Empty Activity". Ensuite cliquez bouton de droite dans la section de code de gauche. Effectuez New -> Activity -> Tabbed Activity. Il est possible sinon de débiter un projet de type Tabbed Activity. Cela reviens sensiblement au même.
2. Dans le TabBar vous allez avoir 3 "tabs". Vous pouvez voir une maquette complète de ce que vous devez faire plus bas:
 1. **Offres:** permet de consulter toutes les offres sous forme de liste(Liste Marques->Liste Modèle->Liste Offres->Détail de l'offre)

2. **Vendre:** permet de remplir un formulaire pour ajouter une nouvelle offre
3. **Mes Annonces:** Affiche une liste de vos offres présentement sur le serveur.(Liste Offres->Détail de l'offre)
3. Assurez-vous de retirer tout artifice de navigation qui sont inutile et qu'Android studio a créé pour vous. Par exemple le bouton flottant "courriel".
4. Modifiez les couleurs de base de l'application pour avoir un mélange de rouge et blanc dans l'Action Bar. Référez-vous à la maquette pour les bonnes couleurs..
5. Dans l'onglet Offres, il y aura quatre (4) pages qui permettront de consulter toutes les offres:
 1. Une liste des marques de voitures
 2. Une liste des modèles pour une marque
 3. Une liste d'offres en fonction du modèle sélectionné
 4. Une page de description de l'offre. Vous devez afficher l'image qui est associée à l'offre et sa **fiche complète** (description, vendeur, etc) qui est uniquement disponible sur la méthode de détail dans l'API.
 1. Un bouton contact doit également être présent et ouvrir une application de courriel avec l'adresse courriel du vendeur.
6. Dans l'onglet "Vendre" il y aura une (1) seule page:
 1. Une page de formulaire pour permettre de rajouter des offres.
 2. Pour des fins de correction, chacun de vos champs doit avoir une valeur par défaut, la valeur est à votre choix.
 3. Chaque champ doit être validé dans l'interface avec des valeurs logiques.
7. Le placement de l'information et l'information affichée dans les pages et les cellules doit être conforme à la maquette présentée plus bas.
8. Dans l'onglet "Mes annonces", il y aura deux (2) pages:
 1. Une liste de vos offres qui reprend le même affichage que lors de la consultation des offres "générales".
 2. Une page de description de l'offre qui reprend le même affichage que lors de la consultation des offres "générales".
9. Certaines méthodes sont protégées par un header d'authentification. Pour vous authentifier vous devez appeler la méthode `/account/login/` avec vos informations personnelles de l'Université Laval. Votre courriel est le courriel qui vous a été fourni par l'Université Laval et le numéro d'identification est celui qui est sur votre carte étudiante(composé de 9 chiffres).
 1. Le token qui sera retourné par la méthode login doit être inclus dans le header *Authorization* avec la valeur "*Basic MON_TOKEN*" comme discutée en classe.

2. Le token expire après un certain temps(maximum 24h) vous ne pouvez donc pas le hardcoder dans votre application.
3. Lors d'un "challenge" d'authentification(le serveur vous retourne alors un HTTP 401) vous devrez afficher un AlertDialog avec 2 champs EditText qui contiendront le courriel et le numéro d'identification de même qu'un bouton Connexion. Le bouton Connexion effectuera l'appel au login. Une fois le login effectué votre application doit poursuivre d'elle même l'action que vous alliez effectuer.
 1. Les champs dans l'AlertDialog tout comme le reste de l'application doivent être renseignés par défaut pour fin de correction.
 2. Un aperçu de cet AlertDialog est disponible dans la maquette.
10. Une librairie comme Picasso pourra être d'une grande aide pour le chargement d'image, mais vous pouvez utiliser une autre librairie. Même chose au niveau d'une librairie réseau vous pouvez utiliser OkHttp ou une autre librairie de votre choix.
11. Vous pouvez utiliser des ListView ou des RecyclerView, mais vous devez absolument bien intégrer le concept de **ViewHolder**.
12. Vous n'avez aucune restriction au niveau de l'utilisation des Fragments et/ou Activity, à l'exception du ViewPager dont les onglets sont obligatoirement des Fragments. Vous pouvez également les mélanger. Vous devez par contre vous assurer que le bouton back du téléphone réagisse correctement.
13. Assurez-vous que tout le contenu de votre application soit visible sur tout type de téléphone, petits et gros!
14. Soyez lâche et réutilisez vos classes!

Dossier de remise

La nomenclature de votre dossier de remise doit être de cette forme et uniquement de cette forme:

- Fichier zip à votre nom qui contient
 - L'application en format APK nommé "TP3.apk"
 - Le bundle git nommé "TP3.bundle"

La remise de votre travail doit être effectuée sur le portail des cours. Une boîte de dépôt est disponible dans la section "Évaluations et résultats"

Barème de correction

18% de la note finale est allouée pour ce travail pratique

	Note
Qualité du code source (Nom des objets, respect des normes de programmations Kotlin, réutilisation de code)	/3
Concept de programmation - Liste intégré avec ViewHolder	/2 /2
Réalisation de l'application testé via APK - Présentation des offres(Marques, Modèles, liste d'offres, ViewPager) - Page de description d'offre - Présentation page de formulaire - POST fonctionnel - Login/Authentification - Validation de champ - Page "Mes Annonces"	/22 /6 /4 /4 /3 /3 /2 /2
Respect des consignes (signature de l'APK, format de remise)	/3
Total	/32

API

- Documentation disponible sur cette URL: <https://tp3.infomobile.app/docs/>
- URL de base de l'API : <https://tp3.infomobile.app/api/v1/>

Aperçu

Maquette : <https://adobe.ly/391uFW8>

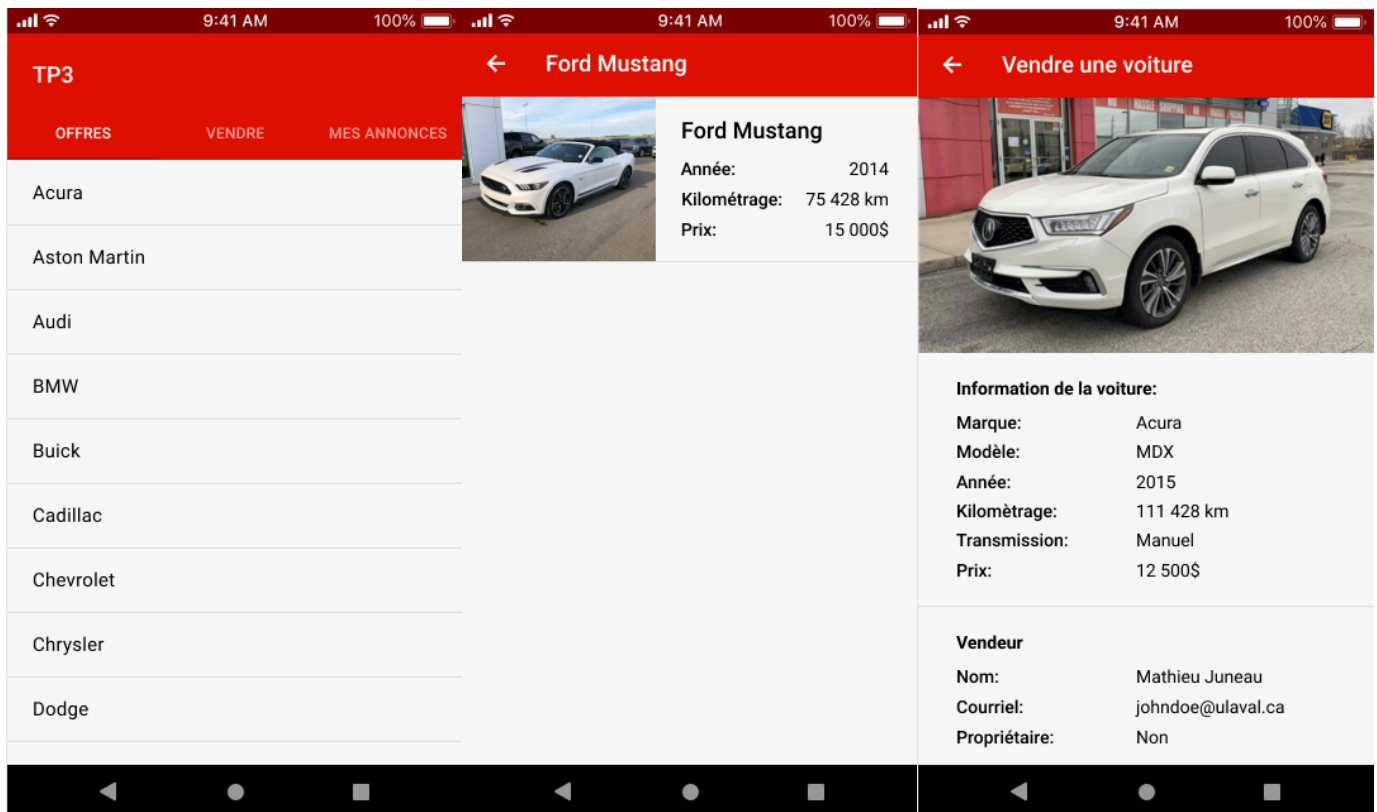
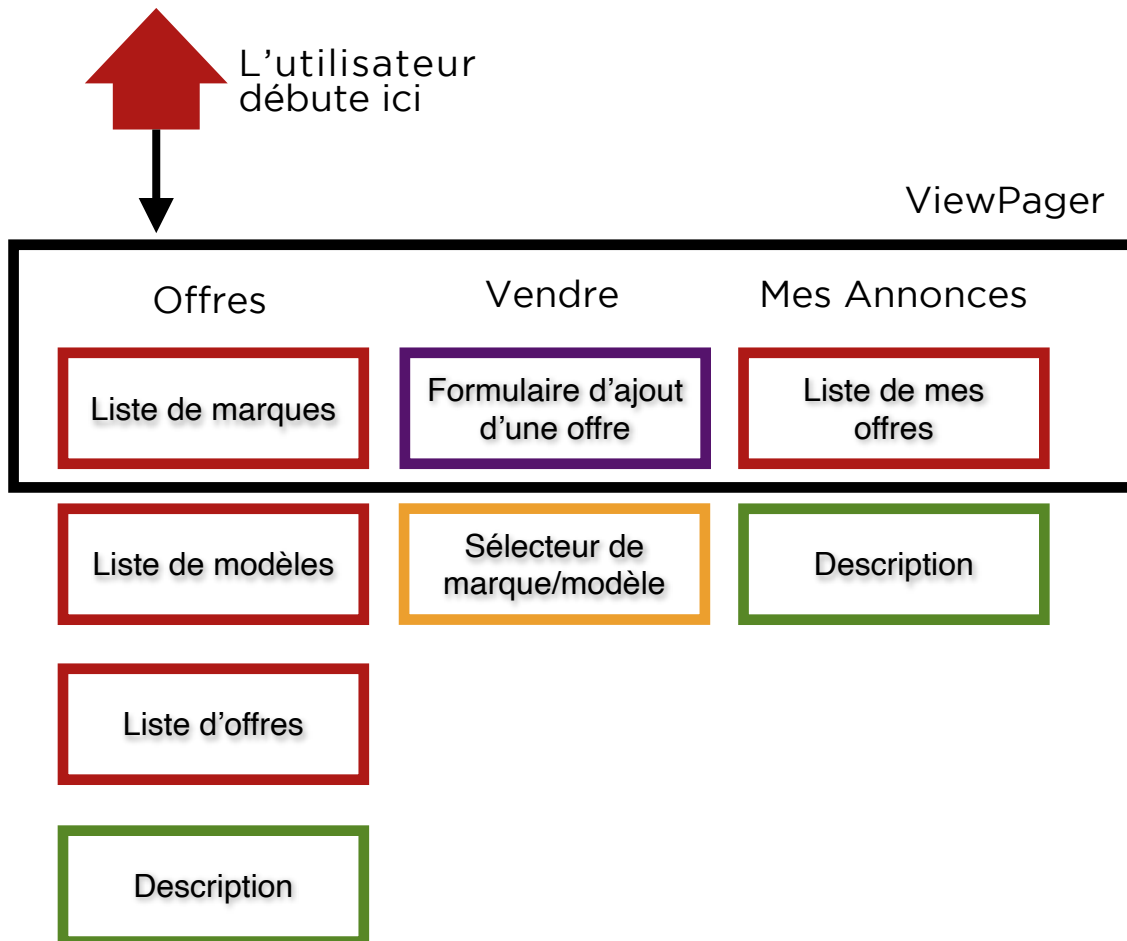


Schéma de navigation l'application



Si on effectue un back ou Up dans le ViewPager on quitte l'application. Dans le reste des pages on retourne au parent.