

Projet développement mobile

Date de rendu : 09/11/2020

Le projet du module Développement Mobile a pour objectif d'approfondir tout, ou en partie, les éléments abordés pendant les séances ainsi que d'explorer des thématiques non abordées. Voici les thèmes vues durant les séances de cours :

1. L'ABC d'Android
2. Kotlin pour Android
3. Personnalisation d'Interface Utilisateur
4. Outils du développeur
5. Interface Utilisateur Native
6. Menus
7. Lliste d'éléments
8. Persistance de Données
9. Communication HTTPS

Date de rendu intermédiaire : 02/11/2020

L'objectif est de créer une application **native Android** avec le langage **Kotlin**. Elle comporterait à minima un menu élaboré (*NavigationDrawer*, *OptionMenu*, etc.) et mettrait en avant un maximum d'élément du *Material Design* (*FAB*, *Cards*, etc.). Le contenu de l'application relève de votre choix.

A. Description du Projet

1] Modalité de rendu

Groupe 2 personnes

Projet Application Mobile Native Android avec Kotlin

README Impressions écrans de l'application

Format Mail macha.dacostabarros@ynov.com avec en Cc les membres du groupe, en sujet [Projet Développement Mobile] et contenant le lien GIT du projet.

SpeechMyApp Présentation de l'application (le 16 / 11 / 2020)

L'application mobile devra être réalisé à l'aide d'un gestionnaire de version GIT (*GitLab* ou *GitHub*).



Certains développements classiques d'Android sont attendus :

- Utilisation d'un *ConstraintLayout*
- Affichage d'une liste d'éléments (cf. *RecyclerView* [2])
- Utilisation de la bibliothèque (cf. *Anko* [1])
- Persistance de données (cf. *SharedPreferences* [5], ou Base de données *textitAnkoSqlite* [3])
- Communication réseau avec un serveur distant (cf. *Retrofit* [4], exemple d'API existante sur <https://www.programmableweb.com/>)

Par ailleurs, voici quelques pistes pour aller plus loin :

- Utilisation des capteurs du support mobile *Android*
- Multiplateforme avec *iOS*

2] Exemple de Projet

<https://github.com/Haeliss/ApplicationProjet>

B. Critères de notation

Ces critères de notation sont classés par ordre de priorité.

1. Élaboration du fichier README.md
2. Respect de la date de rendu
3. Architecture de navigation mis en place (menus, etc)
4. Pertinence du système de navigation mis en place
5. Interface graphique des écrans (éléments natifs)
6. Utilisation des ressources pour externaliser les chaînes de caractères (points négatifs pour les chaînes de caractère codées en dure)
7. Cohérence de nommage des classes et variables
8. Indentation du code
9. Organisation du code dans des packages
10. Attention aux variables ou parties de code inutilisées
11. Utilisation d'éléments non vus en cours (capteurs : son, vibration, bluetooth, etc.)
12. Pertinence des *commits* sur GIT
13. Originalité du projet

Webographie

- [1] ChillCoding Macha DA COSTA. Utiliser la bibliothèque anko. <https://www.chillcoding.com/blog/2017/10/09/utiliser-anko-kotlin-android/>.
- [2] ChillCoding Macha DACOSTA. Afficher une liste d'éléments avec un recyclerview. <https://www.chillcoding.com/blog/2018/10/22/creer-liste-recyclerview-kotlin-android/>.
- [3] ChillCoding Macha DACOSTA. Créer une base de données locale avec anko sqlite. <https://www.chillcoding.com/blog/2018/01/17/creer-bdd-sqlite-kotlin-android/>.
- [4] ChillCoding Macha DACOSTA. Faire une requête http get avec retrofit en kotlin android.
- [5] ChillCoding Macha DACOSTA. Utiliser les fichiers de préférences dans une application android. <https://www.chillcoding.com/blog/2014/10/10/utiliser-fichier-preferences/>.