



HAUTE ÉCOLE DE
NAMUR-LIÈGE-LUXEMBOURG

Catégorie économique – Informatique de gestion

Année académique : 2019 - 2020

Développement mobile

Consignes pour l'examen

Projet Smartcity

Le projet Smartcity couvre l'UE Programmation et nouvelles technologies et doit être réalisé par une équipe de **deux étudiants** d'Informatique de gestion.

Le projet couvre les trois activités d'apprentissage de l'UE. Concernant l'AA Développement Mobile, cela correspond au **prototype fonctionnel** d'une application Android.

Consignes

*Données dans le cloud

L'application Android doit accéder **en lecture et en écriture** à des données stockées dans le cloud, via l'**API** de style architectural **REST** développée dans l'activité d'apprentissage Technologies back-end et Cloud. L'application doit donc sérialiser et désérialiser le contenu échangé en **Json**.

Des Open Data peuvent également être exploitées.

Les accès à toutes REST API doivent impérativement être gérés en Android dans des **tâches asynchrones**.

*Découpe en couches

Une attention particulière sera apportée à l'architecture de l'application (découpe en couches...).

Par conséquent, n'utilisez **pas la librairie Volley** pour les accès à la REST API car il y aurait couplage des couches, la même classe Java gérant à la fois des composants de la vue et l'accès aux données.

*Orientation portrait et paysage

Les deux types d'orientation (paysage/portrait) doivent être gérés pour **au moins deux écrans**. Par conséquent, des **fichiers (layout) différents** doivent être prévus pour l'orientation portrait et l'orientation paysage pour au moins deux écrans. La **disposition** des composants graphiques (views en Android) doit être **différente** dans les deux orientations.

*Clean Code

Le code devra observer les principes d'organisation abordés durant tout le cursus : découpe en couche, clean code, pas de duplication...

Prévoyez des **constantes** là où cela est judicieux.

L'orthographe sera prise en compte aussi bien dans les libellés des interfaces utilisateurs que dans les messages d'erreur.

*Gestion des chaînes de caractères

Ne **pas hard-coder** les chaînes de caractères.

*Validation des formulaires

La validation des différents formulaires est à prendre en charge (données obligatoires, format correct, login déjà existant...).

*Gestion des erreurs

Les messages d'erreur doivent être **compréhensibles et significatifs pour l'utilisateur** (doublons, donnée manquante...), y compris en cas de requêtes http échouant (ex : requête invalide, erreur de connexion à l'API...).

*Gestion de la perte de la connexion internet

Si la connexion internet est perdue, l'application ne doit pas s'arrêter et un **message clair et explicite** doit prévenir l'utilisateur de la perte de la connexion.

Une façon de procéder est de tester l'existence de la connexion avant tout accès aux données dans le cloud.

*Kotlin

Il vous est demandé d'écrire **au moins une classe business en Kotlin**.

Internationalisation

L'application ne doit être prévue qu'en **une seule langue**, anglais ou français. Pas d'internationalisation imposée.

Testing

Il est évidemment plus que conseillé de tester votre application !
Notamment, un système de test unitaire existe dans AndroidStudio, profitez-en.

Bon travail !

⇒ Si un seul critère marqué d'une * n'est pas pris en compte, votre projet sera automatiquement raté !

Evaluation

Remise du travail

Le **code** de l'application sera communiqué aux **deux** professeurs via un lien github au plus tard **une semaine avant le premier jour d'examen** de programmation mobile.

Le code devra être accompagné d'un dossier comprenant :

- Le nom des étudiants constituant le groupe
- Une description du domaine d'application (max ½ p)
- Une liste des fonctionnalités implémentées pour le prototype fonctionnel ainsi que les maquettes d'écran correspondantes
- Une liste des fonctionnalités prévues initialement mais non implémentées ainsi que les maquettes d'écran correspondantes
- Le schéma (ERD) de base de données accompagné d'une brève explication (max ½ p)

Critères de cotation

La cotation est basée d'une part sur un travail de groupe (réalisation de l'application Android : démonstration des fonctionnalités et qualité du code) et d'autre part sur un examen oral. L'évaluation de l'application Android constituera une **cote de base** pour l'examen oral ; cette cote **pourra être sensiblement augmentée ou diminuée** en fonction des réponses données par l'étudiant aux questions posées lors de l'oral. Notez que si un seul critère marqué d'une * n'est pas pris en compte, votre projet sera automatiquement raté !

Lors de l'examen oral, il est attendu de l'étudiant qu'il sache s'approprier le code mis en place par l'équipe et détecter des faiblesses dans un code donné et proposer des solutions pour y remédier. La note de base de l'application pourra être transformée en un échec si l'étudiant est incapable de répondre aux questions posées lors de l'oral, tant au niveau théorique que pratique.