



W4 - Mobile

W-WEB-320

my_app

mobile application



my_app

delivery method: app on GitHub

language: free to you



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

INTRODUCTION

Le Père - Noël vient de passer avec une saison de retard, mais il vous offre un nouveau smartphone ! Vous découvrirez que votre appareil est équipé d'un accéléromètre, d'un gyroscope, etc.
Aussi ravis de votre cadeau vous décidez de créer une application mobile qui exploite au mieux les nouvelles fonctionnalités de votre téléphone.

SUJET

Vous devez combiner une API publique (exemple : <https://github.com/toddmotto/public-apis>) avec des fonctionnalités de votre téléphone pour créer une application faisant usage des deux, de façon pertinente.
Bon exemple :

- Partager sa position actuelle avec ses amis sur un chat.

Mauvais exemple :

- Pouvoir allumer sa lampe torche depuis une application de chat.



Attention : l'implémentation des fonctionnalités à la priorité sur la pertinence !

PRESENTATION DU SUJET

Lors de la soutenance les élèves doivent présenter leur projet à l'aide d'un slide. Dans celui-ci, les élèves expliqueront leur choix technologique, qu'ils devront justifier et leur choix d'API. Il illustrera les différentes implémentations de l'API via des exemples, screens, etc. Ils expliqueront également les composants du téléphone qu'ils ont utilisés et dans quel but. Une présentation du projet sur mobile est également requise, afin de pouvoir au mieux tester les fonctionnalités implémentées. Il est aussi possible de faire usage d'un émulateur à la place. (Cf. Liste non exhaustive en fin de document)

IMPLEMENTATION DE L'API

Pour l'usage et l'implémentation de l'API, il vous est demandé d'utiliser au minimum 5 requêtes différentes afin d'obtenir vos informations. Si avec une seule requête vous affichez deux informations différentes, c'est tout de même compté comme étant une seule requête.

Exemple: Si en demandant la liste d'amis vous pouvez en plus savoir lesquels sont connectés ou non, c'est tout de même compté comme étant une seule requête malgré la présence de deux fonctionnalités.

IMPLEMENTATION DES COMPOSANTS DU TELEPHONE

L'application doit interagir avec 3 éléments minimum du téléphone. Par exemple :

- accéléromètre
- gyroscope
- proximité
- orientation
- appareil photo
- contacts
- agenda
- ...

L'utilisation de ses éléments doit permettre la récupération de données spécifiques qui pourront ensuite être envoyées à l'API.



!! Les éléments choisis doivent-êtré impérativement présents sur le téléphone / émulateur. Il faut que cela soit fonctionnel lors de la présentation en soutenance!!

USER EXPERIENCE

La notation prend aussi en compte l'UX, qui note l'aspect simple, intuitif et facile d'emploi de votre application. Elle comprend entre autres :

L'aspect intuitif des paramètres:

- facile d'accès - pas besoin de deviner

L'aspect intuitif de l'application et des fonctionnalités - fonctionnalités claires et faciles à utiliser

L'aspect général :

- choix des couleurs - agencement - effort sur l'apparence

SDK & EMULATEUR UTILISABLE

Liste non exhaustive :

- <https://netbeans.org/community/releases/81/index.html>
- <http://www.javaworld.com/article/2074670/mobile-java/mobile-java-best-tools-for-mobile-application-development.html>
- <https://docs.unrealengine.com/latest/INT/Platforms/Mobile/https://unity3d.com>
- <https://docs.unrealengine.com/latest/INT/Platforms/Mobile/https://unity3d.com>
- <https://developer.android.com/studio/index.html>
- <https://www.microsoft.com/en-us/download/details.aspx?id=35471>



Language you can work on can be Flutter, ReactNative, Ionic, Kotlin, ...