

Module 335

Réaliser une application pour mobile

CIE Informaticiens CFC – Journée 1

Flavio Pacifico

CIE 335 : Sommaire

- Règles de base et comportement
- Qui suis-je ?
- Horaires.
- Présentation des objectifs ICT.
- Présentation du programme du cours.
- Théorie et culture général sur l'environnement mobile
- Théorie sur l'IHM (HMI)
- Exercice en relation

Règles de base et comportement

Salle de cours



- S'asseoir correctement
- Respecter le matériel mis à votre disposition ainsi que celui des autres participants
- Respecter la propreté de la salle; nettoyer votre place après le cours
- Pas d'écouteur pendant le cours



Formation



- Les 5 jours de cours sont obligatoires (écoutez, participer, travailler et valider le module)
- Les cours commencent à 8h30. Les arrivées tardives sont comptabilisées, puis envoyées au maître d'apprentissage en fin de cours
- Passé 10h00 vous êtes marqué/e comme absent/e pour la journée !
- Toute absence nécessite un justificatif à envoyer à cie@ict-fpsso.ch
- Obligation d'être présent/e durant 80% du cours pour se présenter à l'examen et valider le CIE
- Trop d'absence justifiée = Répétition du cours CIE
- Trop d'absence injustifiée = Répétition du cours CIE = facture au maître d'apprentissage

Pause / Extérieur salle de cours



- Avoir un comportement professionnel (pas de chahutage, etc.)
- Respecter l'environnement mis à votre disposition

Cas particuliers



- Certificat médecin, etc. à présenter à l'enseignant au début du cours

Qui suis-je ?

Activités



Formations

- ✓ CIE 294 – Réaliser le front-end d'une application Web interactive - VD
- ✓ CIE 295 – Réaliser le back-end pour des applications – VD
- ✓ CIE 335 - Réaliser une application pour mobile - VD

Développement



- ✓ Langage Web
 - ✓ CRM en mode web (HTML, CSS, Javascript, Framework JS, PHP, nodeJS)
 - ✓ Base de données SQL
- ✓ C# / C++



Support

- ✓ Support sur les applications développées.



CIE 335 : Déroulement



4.5 jours : cours / participation obligatoire



- ✓ Théorie
- ✓ Exercices
- ✓ Pause matin, midi et après-midi

0.5 jour = examen individuel



- ✓ Avec document

CIE 335 : Objectifs

Réaliser une application pour mobile

Compétence

Mettre en œuvre et tester une application pour mobile selon directives.

Objectifs opérationnels

1. Analyser la donnée, développer la fonctionnalité et le scénario.
2. Elaborer le concept de solution pour l'application et vérifier l'implémentation dans des solutions existantes.
3. Programmer une application avec un environnement de développement usuel en tenant compte des possibilités et restrictions des appareils mobiles.
4. Planifier la publication de l'application sur une plateforme usuelle et fixer les étapes nécessaires.
5. Vérifier l'application mobile selon le plan de tests, saisir les résultats des tests et, le cas échéant, entreprendre les corrections nécessaires.

CIE 335 : Programme

Jour	Matière
1	Présentation Théorie et culture général sur l'environnement mobile Théorie sur l'IHM (HMI) Exercice en relation
2	Théorie sur les ressources et la persistance des données Exercice en relation
3	Théorie sur les capteurs Exercice en relation
4	Théorie sur la gestion du temps et les services web Exercice en relation
5	Théorie sur le déploiement Examen 😊😊

CIE 335 : Les systèmes mobiles

Introduction



Il y a plusieurs systèmes mobiles existents que nous allons aborder :

- ✓ Android
- ✓ iOS
- ✓ Windows 10
- ✓ Ancêtres et exotiques

CIE 335 : Les systèmes mobiles

Introduction



Chaque téléphone intelligent, comme tout ordinateur, possède un système.

De même qu'il existe plusieurs systèmes d'exploitation pour ordinateurs personnels, il existe plusieurs systèmes d'exploitation pour téléphones, la plupart étant également utilisés dans les tablettes tactiles, voire sur certains ordinateurs.

CIE 335 : Les systèmes mobiles

Android



Android est un système d'exploitation spécialisé au départ pour les téléphones mobiles. Datant de 2007 et développé par la société Google (à partir d'un projet développé par une start-up rachetée par Google), il est basé sur le noyau Linux.

Android a fait le choix de la plateforme Java pour le développement, mais sans utiliser la machine virtuelle Java standard (propriété de la société Oracle).

Jusqu'à la version 4.4 la machine virtuelle s'appelle Dalvik, puis ART (Android RunTime) à partir de la version 5.0. La bibliothèque standard d'Android ressemble beaucoup à J2SE (Java 2 Standard Edition) utilisé pour le développement « classique » sur ordinateurs, ce qui permet aux développeurs Java d'être familiarisés rapidement avec l'environnement.

D'abord destiné aux téléphones, le système Android est à présent disponible sur des tablettes, des ordinateurs, des montres, des téléviseurs, des box Internet...

C'est le système d'exploitation le plus utilisé dans le monde (en nombre d'appareils).

Les versions majeures d'Android se succèdent au rythme d'environ une par an.

CIE 335 : Les systèmes mobiles

iOS



iOS est le système d'exploitation, créé par Apple, qui équipe les appareils mobiles de la marque à la pomme: iPhone, iPad, iPod.

C'est un système dérivé de macOS (le système d'exploitation des ordinateurs Apple), créé en 2007. Le système évoluant au rythme d'une version par an environ (les mises à jour de sécurité sont beaucoup plus fréquentes, évidemment).

Il est obligatoire d'utiliser un ordinateur Apple sous macOS pour le développement d'applications pour iOS.

CIE 335 : Les systèmes mobiles

Windows 10



Windows est un système d'exploitation initialement destiné aux ordinateurs personnels de type PC, mais différentes branches de ce système ont été réalisées à destination des appareils mobiles (PDA puis téléphones).

Depuis la version 10 de Windows, son éditeur Microsoft a unifié ses différentes branches : C'est la plateforme Windows universelle (UWP: Universal Windows Platform) qui permet d'exécuter avec le même code, des applications sur PC, téléphone mobile, console de jeu Xbox, etc.

La présence de Windows sur des téléphones, bien qu'ancienne, n'a jamais été à la hauteur des deux autres plateformes (Android et iOS) et est en chute libre ces dernières années. Néanmoins, le côté universel de la plateforme la rend intéressante : développer pour téléphone n'est ici plus spécifique!

Il est obligatoire d'utiliser un ordinateur sous Windows 10 pour pouvoir développer pour un appareil Windows 10.

CIE 335 : Les systèmes mobiles

Ancêtres et exotique



Malgré ce qu'on pense souvent, les téléphones intelligents ne sont pas apparus avec l'iPhone !

Avant même les premiers téléphones programmables, il existait des appareils qui, mis à part la fonction de téléphonie, en étaient proches : les PDA (personal digital assistant).

Un des premiers systèmes d'exploitation pour appareils mobiles était Palm OS. Créé en 1995, ce système d'exploitation embarqué était utilisé dans des PDA, des smartphones, des montres, des GPS de voiture, des lecteurs de code barres, des consoles de jeux.

Il utilisait un écran tactile utilisable au doigt ou à l'aide d'un stylet. La programmation se faisait en général en C ou C++, mais il existait une machine virtuelle Java pour Palm OS (donc la possibilité de faire des applications en Java), des compilateurs Pascal et Basic et quelques autres langages plus exotiques.

Palm OS est ensuite devenu Palm WebOs, puis HP Webos et les derniers appareils sous ce système sont sortis en 2011.

CIE 335 : Les systèmes mobiles

Ancêtres et exotique (PDA sous PalmOS)



CIE 335 : Les systèmes mobiles

Ancêtres et exotique



Microsoft, leader dans les systèmes d'exploitation pour ordinateurs, a sorti en 1996 un système d'exploitation spécifique pour appareils mobiles appelé Windows CE. Utilisé dans des PDA, des téléphones, des DAB (Distributeur Automatique de Billets), des terminaux de paiement des lecteurs de code-barres, des consoles de jeu, des GPS de voiture, le système a été très utilisé dans les premières générations d'appareils mobiles.

Le système a ensuite évolué en Windows mobile mais a été abandonné en 2012 pour être remplacé par Windows phone (basé lui sur un autre noyau, donc totalement différent malgré le nom proche), lui-même remplacé par Windows 10 universel.

CIE 335 : Les systèmes mobiles

Ancêtres et exotique (PDA sous Windows CE)



CIE 335 : Les modes de développement

Plusieurs modes



Il existe bien sûr plusieurs modes de développement, nous allons faire un tour de chacun :

- ✓ Développement natif
- ✓ Multiplateforme web
- ✓ Multiplateforme natif

CIE 335 : Les modes de développement

Développement natif



Le développement appelé natif est celui préconisé par la plateforme: il utilise en général un environnement bien précis, définis par l'éditeur de la plateforme (Google pour Android, Apple pour iOS, Microsoft pour Windows).

Il présente l'avantage d'être évidemment particulièrement adapté pour la plateforme ciblée, mais par conséquent il n'est absolument pas utilisable sur les autres plateformes.

Si vous souhaitez développer pour plusieurs téléphones (par exemple, un Android et un iPhone) vous serez obligés d'utiliser plusieurs codes différents pour chaque plateforme : ceux-ci sont en effets natifs à chaque plateforme.

CIE 335 : Les modes de développement

Développement natif



Avantages	Inconvénients
<ul style="list-style-type: none"> ✓ L'app peut utiliser toutes les fonctionnalités du téléphone. ✓ Elle sera rapide à l'exécution. ✓ L'aspect visuel sera cohérent avec les autres applications. ✓ L'environnement de développement est adapté au déploiement direct dans le magasin applicatif. 	<ul style="list-style-type: none"> X Il est obligatoire de développer plusieurs versions en parallèle si l'on veut cibler plusieurs plateformes. X Le langage de programmation est imposé. X L'environnement de développement est imposé. X Le kit de développement peut être complexe à utiliser.

CIE 335 : Les modes de développement

Développement natif



Les environnements de développement natifs sont, à l'heure actuelle, les suivants :

Pour Android : Android Studio. Disponible pour Windows, Linux, Mac OS.
Utilise les langages Java et Kotlin. Gratuit.

Pour Windows 10 mobile : Visual Studio. Disponible pour Windows 10. Une version gratuite existe.
Utilise principalement le langage C#.

Pour iOS : XCode. Disponible pour Mac OS. Gratuit. Utilise le langage Objective C ou le langage Swift.

CIE 335 : Les modes de développement

Multiplateforme web



La première approche pour les développeurs qui souhaitent utiliser le même code sous différentes plateformes est d'utiliser les technologies du web. En effet, tous les systèmes mobiles (ainsi que les systèmes classiques) possèdent un navigateur capable d'exécuter des applications web.

Il faut bien entendu adapter l'affichage à un terminal mobile (écran plus petit, orientation portrait ou paysage) ainsi qu'aux interactions (clavier virtuel, écran tactile...) mais il est assez rapide de passer d'une application web « pour mobiles » à une application « mobile ».

Les langages utilisés sont ceux qui peuvent être interprétés par un navigateur : HTML 5 et CSS 3 pour la présentation et Javascript | ECMAScript pour le code métier.

Il est ainsi possible de faire une « application web » et de l'exécuter dans le navigateur d'un téléphone, mais ce n'est pas une « vraie » application : elle ne peut pas être installée via le magasin applicatif, ne peut pas utiliser certaines parties du téléphone, et présente une lenteur à l'exécution.

Certains outils, ont recours aux langages du web mais ajoutent un framework qui permet d'utiliser certains composants du téléphone inaccessibles au navigateur. De plus, l'outil réalise une vraie application qui peut être installée par le magasin applicatif. La lenteur d'exécution inhérente à l'interprétation Javascript est en revanche toujours présente.

CIE 335 : Les modes de développement

Multiplateforme web



Avantages	Inconvénients
<ul style="list-style-type: none"> ✓ Le même code peut être utilisé sur un grand nombre de plateformes ✓ L'application peut être issue d'un site web existant et adaptée très rapidement. ✓ Les compétences développeur acquises pour le web sont réutilisables directement. 	<ul style="list-style-type: none"> X L'exécution est un peu lente (par rapport à une application native). X Les fonctionnalités spécifiques à une plateforme ne sont pas utilisables. X L'aspect de l'interface peut être différent des applications « classiques ». X Les langages du web peuvent être difficiles à manipuler.

CIE 335 : Les modes de développement

Multiplateforme web



La solution du multiplateforme web est une bonne solution si l'application ne doit pas être très rapide, si le multiplateforme est obligatoire, si du code « web » existe déjà ou si le développeur possède une forte compétence dans ces langages. Si en revanche vous voulez écrire un jeu 3D, ou utiliser un capteur spécifique, ou si vous ne savez pas du tout coder en HTML5/CSS/Javascript, la solution peut être inadaptée et risque de faire perdre beaucoup de temps.

Ce type de développement est souvent privilégié par les éditeurs de logiciels simples dont l'application fait « doublon » avec le site web et doit être présente sur plusieurs plateformes (Android, Windows et iOS) en plus du web. Une très grande partie du code étant commun, le temps de développement et de maintenance est fortement réduit.

Parmi les outils existants, on peut citer :

- Apache Cordova (open source). Il est notamment intégré à l'EDI Visual Studio 2017 (mais peut être utilisé sans, avec d'autres EDI ou en ligne de commande).
- Adobe Phonegap (basé sur Cordova)
- Appcelerator Titanium
- Ionic
- Capacitor

CIE 335 : Les modes de développement

Multiplateforme natif



L'idée est de combiner les avantages de la programmation native et de la programmation multiplateforme, en permettant, avec une grande base de code commune (voire l'intégralité du code), de développer sur plusieurs plateformes différentes.

L'outil traduit dans les API natives le code commun. Certaines limitations sont possibles, mais moins fréquentes qu'avec le multiplateforme « web ».

Il existe beaucoup d'outils de ce type, dans un grand nombre de langages différents. Utilisant des langages « classiques », il est ainsi possible de récupérer du code d'une application « classique » (bureau) ou d'avoir des parties communes avec d'autres applications, qu'elles soient sur ordinateur, sur web ou sur mobile.

CIE 335 : Les modes de développement

Multiplateforme natif



Avantages	Inconvénients
<ul style="list-style-type: none"> ✓ La vitesse d'exécution est proche voire identique aux applications natives (langages compilés). ✓ Le même code peut servir à plusieurs plateformes. ✓ Utiliser un langage déjà maîtrisé gagne du temps dans certains cas, le code peut être commun avec celui d'une application de bureau ou d'une application web. 	<ul style="list-style-type: none"> X L'aspect peut être légèrement différent des contrôles standards de la plateforme X Les outils de déploiement sont parfois moins efficaces que pour l'environnement natif. X Certaines fonctionnalités ne sont pas accessibles.

CIE 335 : Les modes de développement

Multiplateforme natif

Cette solution est souvent la plus intéressante et est de plus en plus utilisée par les éditeurs de logiciels à destination des téléphones.



Parmi les outils multiplateformes natifs :

- Xamarin.Forms. Permet de coder en C#/NET sur un PC/Windows ou un Mac. Cibles possibles : Windows UWP, Android, iOS.
- Qt. Permet de coder en C++ sur un PC/Windows, un PC/Linux ou un Mac. Cibles possibles : Windows, Linux, Mac OS, Android, iOS, Windows UWP.
- Windev mobile. Permet de coder dans un langage spécifique (Wlangage) sur un PC/windows. Cibles possibles : Windows, Windows phone, Windows UWP, Linux, Android, Mac OS, iOS. Produit payant (mais une version gratuite, limitée, existe).
- Delphi. Permet de coder en Pascal sur un PC/Windows. Cibles possibles : Windows, Android, iOS, Mac OS, UWP. Produit payant mais il existe depuis peu une version gratuite.
- C++ Builder. Similaire à Delphi (même éditeur, souvent livrés ensemble) mais avec le langage C++.

Il est important de noter que dans tous les cas, quel que soit l'outil, il est impossible de créer une application pour iOS sans un Mac et il est impossible de créer une application pour Windows UWP sans un PC/Windows 10. Seul Android n'est pas sectaire !

Les éditeurs souhaitant développer pour les trois plateformes les plus importantes (Android, iOS, Windows 10) doivent donc utiliser au moins deux machines différentes :

- Un Apple Mac avec un système OS X
- Un PC Windows 10

CIE 335 : Choisir ses outils

Les outils par plateforme






Pour bien choisir son outil, il convient en premier lieu de connaître sa plateforme de développement, c'est-à-dire la machine sur laquelle on souhaite écrire l'application. Celle-ci n'est évidemment pas la même que la machine qui va exécuter l'application ! On ne code pas sur un téléphone (même si les claviers virtuels ont fait des progrès) !

Le tableau suivant vous permet de savoir quels outils sont utilisables sur votre machine (Windows, macOS ou Linux) :

CIE 335 : Choisir ses outils

Les outils par plateforme



	Visual Studio	Windev	Android Studio	Xcode	Qt	Cordova	Capacitor	Xamarin	Delphi
	Oui	Oui	Oui	Non	Oui	Oui	Oui	Oui	Oui
	Oui	Non	Oui	Oui	Oui	Oui	Oui	Oui	Non
	Non	Non	Oui	Non	Oui	Oui	Oui	Non	Non

Comme on le voit sur ce tableau, aucun système n'exécute tous les outils présentés. Parmi ceux-ci, Android Studio, Qt, Cordova et Capacitor existent sur tous les systèmes présentés

CIE 335 : Choisir ses outils

Les outils par cible






Une autre manière de choisir son outil est de s'intéresser à la cible : sur quel téléphone voulez-vous déployer / exécuter votre application ? Il existe un grand nombre de systèmes, même si actuellement seuls trois (et encore) sont encore en activité : Android, iOS, Windows. Le cas de Windows est un peu particulier car ce n'est pas qu'un système mobile mais également pour tablettes, ordinateurs et consoles de jeu.

Le tableau suivant indique les cibles de chaque outil de développement pour téléphone :

CIE 335 : Choisir ses outils

Les outils par cible



	Visual Studio	Windev mobile	Android Studio	Xcode	Qt	Cordova	Capacitor	Xamarin	Delphi
	Oui	Oui	Oui	Non	Oui	Oui	Oui	Oui	Oui
	Oui	Oui	Non	Oui	Oui	Oui	Oui	Oui	Oui
	Oui	Oui	Non	Non	Oui	Oui	Oui	Oui	Oui

CIE 335 : Choisir ses outils

Les outils par langage



Enfin, vous pouvez choisir votre outil de développement en fonction de votre langage préféré ! En effet, si vous maîtrisez déjà un langage de programmation, il est dommage de devoir en apprendre un autre juste pour s'essayer à la programmation sur téléphones...

La plupart des outils imposant un seul langage. Voici une liste (non exhaustive) d'outils permettant de programmer pour téléphone ou tablette dans votre langage préféré :

- C++ : Qt, C++ Builder, Visual Studio
- C# : Visual Studio
- Pascal : Delphi
- Python : Kivy
- HTML/CSS/Javascript : Cordova, Capacitor
- Java : Android Studio
- Objective-C, Swift : Xcode
- Ruby : XCode

CIE 335 : Outils de développement

Android Studio



Android Studio est l'outil de développement natif pour la plateforme Android. Vous ne pouvez pas écrire d'application pour iOS ou Windows avec cet outil ! A n'utiliser que si vous êtes sûrs de ne faire que de l'Android.

Android Studio est un EDI basé sur IntelliJ Idea (un EDI Java). Il existe pour un grand nombre de systèmes d'exploitation, notamment Windows, Linux et mac OS. Il intègre tous les outils nécessaires au développement Android, notamment le SDK.

Le langage utilisé pour coder l'application est classiquement le Java, mais récemment Google a ajouté le langage Kotlin pour le développement Android. L'EDI est identique sur chacune des plateformes supportées.

CIE 335 : Outils de développement

XCode



L'environnement XCode n'existe que sous macOS. Pour développer pour iPhone, il faut au minimum la version 4 de XCode et 10.6 de macOS. Pour utiliser le langage Swift, il faut au minimum la version 6 de XCode et 10.10 de macOS (sinon il faut se contenter du langage Objective-C).

Pour pouvoir publier sur l'Apple Store, il faut la version la plus récente de XCode pour pouvoir utiliser le SDK d'iOS le plus récent. Il faut donc garder son Mac et XCode à jour !

CIE 335 : Outils de développement

Qt



Le framework Qt est fortement multiplateforme : il existe pour Windows (y compris les versions récentes, WinRT et UWP I), macOS, Linux, Android, iOS... l'environnement de développement intégré (EDI) Qt Creator existe pour Windows, Mac OS, Linux. Le langage utilisé est normalement C++ mais il est possible d'utiliser sous certaines conditions Java, Python ou d'autres langages encore.

Il existe une version open source des outils Qt qui peut donc être librement téléchargée et utilisée pour produire des applications.

CIE 335 : Outils de développement

Visual Studio



L'EDI Visual Studio peut être utilisé, en programmation mobile, avec trois environnements différents :

- Xamarin (intégré) permet de faire du développement Android Windows et iOS en utilisant C# et le framework .NET
- Cordova (intégré) permet de faire du développement Android Windows et iOS en utilisant les langages du web : HTML, CSS, Javascript.
- UWP (intégré) permet de faire du développement pour Windows 10 uniquement, quelle que soit la plateforme (mobile, bureau Xbox, etc...)

Grâce à l'outil Xamarin, il est possible d'utiliser Visual Studio, le langage C# (mais aussi C++, VB, F#...) et le framework .NET pour écrire des applications pour téléphones.

L'outil est disponible pour Windows et macos (pas de Linux disponible pour l'instant).

La version Windows peut faire des applications pour Windows (classique), Windows 10 (universel) et Android.

La version macOS, moins complète, limitée au langage C#, peut réaliser des applications pour macOS, iOS et Android.

CIE 335 : Outils de développement

Capacitor

Nous nous intéresserons à la partie capacitor.



Nous allons utiliser un EDI et Capacitor pour écrire une application « hello world clic me » avec les langages du web, HTML, CSS et Javascript

Commencez par créer une application vide Capacitor (CMD → `npm init @capacitor/app`).

- En premier, nommer votre application
- Deuxièmement, le nom du dossier du projet
- En dernier, on peut laisser par défaut pour le moment.
- Cd «nom dossier»
- Npm install

L'assistant crée un projet contenant divers fichiers :

- Le fichier `src/index.html` qui contient la description de la page principale
- Le fichier `src/css/style.css` qui contient la feuille de style associée à la page principale
- Le fichier `src/js/capacitor-welcome.js` qui contient le code Javascript associé à la page principale

CIE 335 : Capacitor – Exercice 1



Exercice 1 – Clic me

lct-moodle.ch

Jour 1 – Exercice 1