# Programmation mobile

## Rappels

Un ordinateur est structuré en quatre couches : Hardware, OS, Software, User. Entre le Hardware et OS on a les pilotes (drivers).

Les différentes familles d’ordinateurs sont : Les bureaux, Les portables, Les Smart TV, Les objets connectés, les Smart phone.

Les différents systèmes d’exploitation mobile sont :

* Android
* iOS
* Harmony OS
* Symbian

Application native sont des applications dont le code source ne peut s’exécuter que sur un et un seul système d’exploitation. Ainsi on a un ensemble de langage et de technologie permettant de créer ces types d’application.

|  |  |
| --- | --- |
| Langages | OS |
| Java, Kotlin | Android |
| Swift | iOS |
| Flutter (Dart), Apache Cordova, .Net MAUI, Framework7 | Application multi-plateforme |
| Technologie web | PWA (Processing Web Access) |

Pour créer une application il faut également un SDK (Software Development Kit) qui contient des bibliothèques et des fonctions pour manipuler faciliter la communication avec le système d’exploitation.

De plus, il faut un IDE dont Android Studio pour créer les applications mobiles android.

Un émulateur permet de reproduire le matériel sur lequel on va déployer les applications.

Un simulateur permet

Plus : BlueStack permet de créer des téléphones virtuels.

Pour développer une application mobile il faut chercher à savoir si la technologie à utiliser permet de :

* Répondre aux besoins du clients
* Assurer la sécurité
* Assurer l’évolutivité
* Taille de l’application à demander.

En plus, il faut se poser la question sur sa compétence.

# 02/12/2024 : Cours de développement mobile

## Objectifs du cours

* Connaitre le principe de fonctionnement d’une application mobile
* Connaitre les différentes langages et technologiques de la programmation mobile
* Connaitre les étapes de création

## Plan du cours

* Généralité sur le développement mobile
* Le langage Kotlin
* Environnement Android
* Développement Android
* Hébergement d’une application mobile

## Généralités sur le développement mobile

Material UI permet de faire le design des applications mobiles Android

Recherche : (Sur Linux il n’y a pas question de virus)

Les applications mobiles peuvent être : application hybride, application web, application native.

|  |  |  |  |
| --- | --- | --- | --- |
| Critère | Application Native | Application hybride | Application web |
| Communication | Directement avec des API Calls mises à disposition par le OS. | Avec des API Calls par l’intermédiaire d’un Native Container. |  |

## Les étapes de création d’une application mobile

* Analyse des besoins
  + Etude de l’existent
  + Critique de l’existent
  + Proposition de la solution
* Conception
  + Diagramme des cas d’utilisation
  + Diagramme de description
  + Diagramme de séquence
  + Diagramme de classes (Il est bien de produire un diagramme de données coupler au diagramme de classes afin d’alléger ce dernier).
* Développement et Test
* Maintenance et Mise à jour

Pour plus de détail :

* Identifier le problème
* Faire une étude de marché
* Définir les fonctionnalités (En pensant en long termes, prévoir des fonctionnalités à ajouter au fur et à mesure)
* Penser au design (wireframe, Maquette, Prototype) en utilisant Figma, Adobe XD et Canva.
* Choisir une méthode de développement (Hybride, Native, outils de no code (App Builder)
* Développer l’application
* Déploiement sur le marché
* Recevoir de feedback
* Faire des mises à jour et de la maintenance.

Les outils No code : GoodBarder, como, flutterflow ...

# LE LANGAGE KOTLIN

Kotlin est un langage de POO, fonctionnel, avec un typage statique qui permet de compiler sur la JVM et MVJs.

Grâce au framework Ktor on peut créer des applications serveurs.

## Les avantages de Kotlin

* Interopébilité avec Java
* Développement facile
* Robustesse et Sécurité (Pas de problème de NullPointerException)
* Facile à prendre en main

## Apprendre dans Kotlin

* Création d’un projet
* Structure de base d’un programme
* Les opérateurs
* Les types de données
* Les variables et Structures de données
* Les Structures de contrôles
* Les fonctions et Null Safety
* La programmation orienté objet et fonctionnelle