

Chapitre 1 : Introduction au développement mobile

Table des matières

1. Chapitre 1 : Introduction au développement mobile	3
1.1. Introduction	3
1.2. Systèmes d'exploitation pour mobile	4
1.3. Fonctions d'un système d'exploitation pour mobile	4
1.3.1. Gestion de la mémoire	4
1.3.2. Gestion du processeur	4
1.3.3. Gestion des dispositifs	4
1.3.4. Gestion des fichiers	4
1.3.5. Sécurité	5
1.4. Exemples d'OS pour Mobile	5
1.5. Types d'application mobile	5
1.6. Outils et frameworks de développement des applications mobiles	8
1.6.1. Flutter	8
1.6.2. Ionic	9
1.6.3. React Native	10
1.7. Conclusion	11
Bibliographie et Webographie	12

1. Chapitre 1 : Introduction au développement mobile

1.1. Introduction

Il y a presque 7 milliards de personnes sur la planète, dont 5,1 milliards possèdent un téléphone portable. Le nombre d'utilisateurs accédant à Internet sur leurs smartphones a plus que doublé au cours des 5 dernières années, tout comme le nombre d'utilisateurs téléchargeant et utilisant des applications mobiles. Parmi ceux qui utilisent Internet ou le courrier électronique sur leur téléphone, plus d'un tiers se connectent principalement via leurs appareils portables.

Une application mobile ou simplement app mobile, est un programme informatique ou une application logicielle conçue pour s'exécuter sur un appareil mobile tel qu'un téléphone, une tablette, téléviseur, une montre, etc. Les applications mobiles contrastent souvent avec les applications de bureau conçues pour s'exécuter sur des ordinateurs de bureau et les applications Web qui s'exécutent dans les navigateurs Web mobiles plutôt que directement sur l'appareil mobile.



Figure 1.1.exemple de périphériques mobiles.

Les applications étaient à l'origine destinées à l'aide à la productivité telles que les bases de données de courrier électronique, de calendrier et de contacts, mais la demande du public pour les applications a entraîné une expansion rapide dans d'autres domaines tels que les jeux mobiles, l'automatisation d'usine, les services GPS et géolocalisés, le suivi des commandes et des achats, de sorte qu'il y a maintenant des millions d'applications disponibles. Les applications sont généralement téléchargées à partir des magasins d'applications, qui sont un type de plateformes de distribution numérique.

1.2. Systèmes d'exploitation pour mobile

Un système d'exploitation mobile permet à l'utilisateur d'exécuter de différents logiciels d'application sur le mobile, les tablettes, les montres intelligentes, etc. De plus, on peut dire qu'il s'agit d'un type de système d'exploitation spécialement conçu pour les mobiles. Ils sont un mélange de système d'exploitation informatique avec quelques fonctionnalités supplémentaires. En outre, ils sont relativement légers et simples.

Un système d'exploitation (OS) est un programme qui agit comme une interface entre le matériel du système et l'utilisateur. De plus, il gère toutes les interactions entre le logiciel et le matériel [1].

1.3. Fonctions d'un système d'exploitation pour mobile

1.3.1. Gestion de la mémoire

Quel que soit le programme exécuté, il doit être présent dans la mémoire principale. Par conséquent, il peut y avoir plus d'un programme présent à la fois ce qui rend nécessaire de gérer la mémoire. Le système d'exploitation alloue et désalloue la mémoire pendant le multitraitement.

1.3.2. Gestion du processeur

Lorsque plus d'un processus s'exécute, le système d'exploitation décide comment et quand un processus utilisera le processeur. Le système d'exploitation Alloue et désalloue le processeur aux processus.

1.3.3. Gestion des dispositifs

Les processus peuvent nécessiter des dispositifs pour leur fonctionnement tels que les capteurs et les périphériques d'entrées sorties. Cette gestion est effectuée par l'OS. Le système d'exploitation alloue et désalloue des dispositifs à différents processus.

1.3.4. Gestion des fichiers

Les fichiers d'un système sont stockés dans différents répertoires. Le système d'exploitation :

- Gère des registres de l'état et de l'emplacement des fichiers.
- Alloue et désalloue des ressources.

1.3.5. Sécurité

Le système d'exploitation assure la sécurité du système et des programmes grâce à l'authentification d'une part, et la gestion des permissions de l'utilisation des ressources d'une autre part.

1.4. Exemples d'OS pour Mobile

1.4.1. Android

Le système d'exploitation Android est le plus utilisé parmi tous les systèmes d'exploitation mobiles qui existent. Il est gratuit et open source basé sur le noyau Linux développé par Google.

1.4.2. Système d'exploitation Blackberry

Le développeur de ce système d'exploitation est Research In Motion (RIM). Il a été spécialement conçu pour les appareils BlackBerry.

1.4.3. Apple iOS

C'est l'un des OS les plus populaires après Android. Il est conçu pour fonctionner sur les appareils Apple tels que les iPhones, les tablettes iPad, etc. De plus, comme dans le cas du système Android, IOS permet le téléchargement de ses applications via son AppStore.

1.4.4. Harmony OS

C'est un OS conçu par la société chinoise Huawei. Il est spécialement conçu pour être utilisé dans les appareils IoT en incluant les téléphones portables et tous ce qui est mobile.

1.4.5. Système d'exploitation Windows Mobile

Le développeur de ce système d'exploitation est Microsoft. Il est essentiellement conçu pour les Pocket PC et les smartphones fabriqués par la société Microsoft.

1.5. Types d'application mobile

Les applications mobiles sont généralement classées en trois types : natives, hybrides, et Web. Les applications natives sont conçues spécifiquement pour un système d'exploitation mobile, généralement iOS ou Android. Les applications Web sont écrites en HTML5 et CSS et

s'exécutent généralement via un navigateur. Les applications hybrides sont conçues à l'aide de technologies Web telles que JavaScript, CSS et HTML 5 et fonctionnent comme des applications Web mais dans un conteneur natif.

Une application mobile est composée principalement d'une partie interface utilisateur et une partie qui définit le comportement des composantes.

A. Interface de l'utilisateur

- Visible : les composantes visibles de l'interface de l'utilisateur sont représentées par les boutons, champs de texte, Images, Liste déroulante, etc. elles sont utilisées comme un moyen d'interaction avec l'utilisateur pour lui permettre d'entrer des informations ou lui afficher un contenu.
- Invisible : sont des composantes qui permettent une interaction avec les utilisateurs sans qu'elles soit visible en permettant d'utiliser les ressources du mobile comme l'envoi et la réception des SMS, la géolocalisation, textToSpeech, accéléromètre, etc.

B. Comportement

- Evènements : ce sont les évènements auxquels peut s'associer une action comme le clique d'un bouton, l'activation de camera, la réception d'un appel, etc.
- Méthodes : ce sont les actions et les comportements qu'on veut associer à des évènements. Elles sont paramétrables, réutilisables et personnalisables.

1.5.1. Les applications natives

Toutes les applications ciblées sur une plate-forme mobile particulière sont appelées applications natives. Par conséquent, une application destinée aux appareils Apple ne s'exécute pas sur les appareils Android. En conséquence, la plupart des entreprises développent des applications pour plusieurs plates-formes.

Lors du développement d'applications natives, on peut intégrer les meilleurs modules d'interface utilisateur. Cela représente de meilleures performances, une cohérence et une bonne expérience utilisateur. Ces derniers bénéficient également d'un accès plus large aux interfaces de programmation d'applications (APIs) et font un usage illimité de toutes les applications de l'appareil en question. De plus, ils passent également d'une application à une autre sans contraintes compliquées.

L'objectif principal de la création de telles applications est d'assurer les meilleures performances pour un système d'exploitation mobile spécifique.

1.5.2. Les applications Web

Les applications Web mobiles s'appuient sur des applications mobiles natives. Elles utilisent des navigateurs pour s'exécuter et, sont généralement écrites en CSS, HTML5 ou JavaScript.

Les applications Web effectuent la même méthode d'organisation que les applications natives, par contre, elles sont accessibles via un navigateur de site Web qui sera installé sur l'appareil mobile. Les applications Web ne sont pas des applications indépendantes qui peuvent être téléchargées et installées, mais elles sont des sites Web accessibles via des navigateurs web et ajustent leur interface utilisateur à l'appareil du client.

Le développement des applications Web mobiles est plus rapide et moins coûteux que l'application mobile native, en particulier lorsque l'objectif est de prendre en charge une large gamme d'appareils. À l'inverse, il peut exister des fonctionnalités natives de l'appareil mobile (comme le capteur de mouvement) qui ne peuvent pas être exploitées par une application web pour mobile.

1.5.3. Les applications hybrides

Entre les applications Web et les applications natives, une application mobile hybride peut être le bon choix en fonction des besoins et des contraintes de ressources. Les applications hybrides, comme les applications natives, s'exécutent sur l'appareil lui-même (par opposition à l'utilisation d'un navigateur par les applications web pour mobile), mais sont écrites avec des technologies Web (HTML5, CSS et JavaScript) et généralement étayées par un cadre d'application hybride. Plus précisément, les applications hybrides s'exécutent dans un conteneur natif et exploitent le moteur de navigateur de l'appareil pour afficher le code HTML et traiter le code JavaScript localement. Une couche d'abstraction Web vers native permet d'accéder aux fonctionnalités de l'appareil qui ne sont pas accessibles dans les applications Web mobiles, telles que l'accéléromètre, la caméra et le stockage local [2]. Les applications développées à l'aide d'Apache Cordova, Xamarin, React Native, Sencha Touch entrent dans cette catégorie d'applications hybride.

1.6. Outils et frameworks de développement des applications mobiles

Beaucoup de frameworks et d'outils ont été développés pour répondre aux besoins spécifiques des constructeurs d'appareils mobiles. Elles offrent des mécanismes puissants, facile à mettre en œuvre et sécurisés pour le développement efficace et rapide d'applications mobiles. Cela est assuré tout en respectant le mieux les standards adoptés par les OS mobile.

La création d'une application mobile est une tâche complexe. En tant que développeur, il faut trouver des moyens créatifs pour combiner toutes les fonctionnalités suivantes dans un seul projet [3] :

- Hautes performances : utile pour garder les utilisateurs connectés et augmenter la base de données.
- Rentabilité : développer une application prend du temps, et le temps coûte cher. Un projet bien fait aura toujours à l'esprit le coût de production et la portée du marché potentiel.
- Évolutivité : le monde des applications mobiles est en constante évolution. De nouvelles mises à jour des systèmes d'exploitation sortent chaque jour ce qui nécessite que les applications à développer doivent être évolutives.

De nouveaux frameworks ont été créés pour répondre à toutes ces considérations. Des frameworks multiplateformes ont connus un succès vu la possibilité de déployer leurs produits sur plusieurs types de mobile comme le cas de Flutter, Ionic, et React native.

1.6.1. Flutter

Flutter est le plus récent des frameworks multiplateformes. Il a été créé par Google en 2013. Son langage de programmation DART a été inventé par Google en 2011. Flutter est présenté comme étant le successeur de JavaScript en mieux. Depuis sa 1ère version en 2018, Flutter connaît une croissance d'utilisation de 279 % [4]. Comme exemple d'applications conçues avec Flutter, on peut citer Alibaba.

A. Avantages de Flutter

- Simplicité : Flutter est facile à apprendre car il est basé sur Dart qui ressemble un peu à Javascript. Il produit des applications compatibles avec les deux plateformes Android et IOS à partir d'une base de code unique.

- Outils et fonctions avancés : Flutter permet de bénéficier de nombreux outils de développement comme la fonction de rechargement à chaud intégrée qui permet de voir instantanément les modifications apportées à l'application sans avoir à la recompiler.
- Utilisable pour autres plateformes non mobiles: en mai 2019, Google a annoncé une nouvelle prise en charge de Flutter pour les applications de bureau, intégrées, mobiles et Web à partir de la même base de code.

B. Inconvénients du flutter

- Des mise à jours fréquentes : cela rend plus difficile le suivi de toutes les mises à jour du framework.
- Dart : ce langage est encore très nouveau. Il peut donc être difficile de trouver un personnel suffisamment qualifiées pour créer une application avec.
- Les applications Flutter sont volumineuses : en raison de leurs widgets intégrés, les applications Flutter sont plus volumineuses que leurs homologues natives. Étant donné que les utilisateurs ont une mémoire limitée sur leurs appareils, il faut toujours en tenir compte de cette contrainte.

1.6.2. Ionic

Ionic est un framework gratuit créé en 2013 qui utilise les technologies web comme HTML, CSS ou JavaScript [5]. Il s'appuie sur AngularJS [6], Cordova [7] et Node JS [8]. IONIC est utilisé pour développer des applications mobiles et aussi des applications web comme le cas de Flutter.

A. Avantages d'Ionic

- Prise en charge de différentes technologies : à partir d'Ionic 4, il est possible d'utiliser n'importe quel framework frontend pour créer des applications mobiles.
- Facile : il se base sur HTML, CSS et JavaScript, donc l'utilisation de Ionic ne nécessite pas beaucoup d'effort pour une personne qui a été déjà formé au développement Web.

B. Inconvénients d'Ionic

- Ne convient pas aux applications volumineuses : Malheureusement, pour les applications volumineuses, l'utilisation de la composante WebView qui est très utile pour Ionic peut entraîner un manque de performances important.
- Peut manquer de prise en charge des plugins natifs : certains plugins natifs ne sont pas stables et peuvent entrer en conflit les uns avec les autres. De plus, ils peuvent être complètement absents dans la liste des plugins d'une distribution.

1.6.3. React Native

React Native [9] a été créé par Facebook. Sa première version date de 2015. Il est basé sur React qui est une librairie JavaScript, langage habituellement utilisé pour le Web. Facebook, Instagram, Skype ou Airbnb [10] ont été réalisés en partie ou entièrement grâce à cette technologie.

A. Avantages de React Native

- Apprentissage facile : comme il est basé sur JavaScript ou React, l'utilisation de ce framework est simple. Il n'y a rien à apprendre de nouveau à part gérer les composants personnalisés offerts par cet outil.
- Idéal pour la productivité : React Native est livré avec des fonctionnalités utiles telles que le rechargement à chaud et les outils de développement Chrome pour le débogage des applications.
- Performances : les performances des applications créées à l'aide de React Native sont proches de celles d'un code Natif.
- Une communauté importante : plus de 1500 contributeurs soutiennent cet outil, et de grandes entreprises technologiques l'utilisent comme Airbnb, Uber et Tesla.

B. Inconvénients de React Native

- Peu de composants tiers : ce projet étant encore nouveau, il n'y a pas autant de composants tiers que parfois, on est obligé de mettre en œuvre une solution personnalisée à partir de zéro.

1.7. Conclusion

Dans ce chapitre, on a présenté des notions de base liées au contexte du développement mobile en général, les OS pour mobile existants, les frameworks les plus populaires afin de pouvoir se positionner vis-à-vis les technologies existantes, les problématiques et les solutions possibles. A la fin de ce chapitre, l'étudiant est capable à comprendre les axes qui influencent le marché des applications mobiles et il sera capable de définir les grandes lignes pour ses futures applications en fonction des clients ciblés et de ce que existent comme technologies sur le marché.

Bibliographie et Webographie

- [1] « Mobile Operating System: Definition, Types, Features and Examples », *Toppr-guides*, 28 janvier 2021. <https://www.toppr.com/guides/computer-science/computer-fundamentals/operating-system/mobile-operating-system/> (consulté le 11 octobre 2021).
- [2] « Developing Mobile Web Applications: When, Why, and How », *Toptal Engineering Blog*. <https://www.toptal.com/android/developing-mobile-web-apps-when-why-and-how> (consulté le 11 octobre 2021).
- [3] « Comparison of Mobile Development Platforms in 2021 ». <https://strapi.io/blog/comparing-mobile-development-platforms-in-2021> (consulté le 11 octobre 2021).
- [4] « Flutter, le prochain leader du développement d'application multiplateforme ? », *Idéematic*, 18 août 2020. <https://www.ideematic.com/actualites/2020/08/flutter-le-prochain-leader-du-developpement-dapplication-multiplateforme/> (consulté le 11 octobre 2021).
- [5] Ionic, « Cross-Platform Mobile App Development », *Ionic Framework*. <https://ionicframework.com/> (consulté le 16 février 2022).
- [6] « AngularJS — Superheroic JavaScript MVW Framework ». <https://angularjs.org/> (consulté le 16 février 2022).
- [7] « Apache Cordova ». <https://cordova.apache.org/> (consulté le 16 février 2022).
- [8] Node.js, « Node.js », *Node.js*. <https://nodejs.org/en/> (consulté le 16 février 2022).
- [9] « React Native · Learn once, write anywhere ». <https://reactnative.dev/> (consulté le 16 février 2022).
- [10] « Airbnb : des logements et des activités incroyables », *Airbnb*. <https://www.airbnb.fr> (consulté le 16 février 2022).