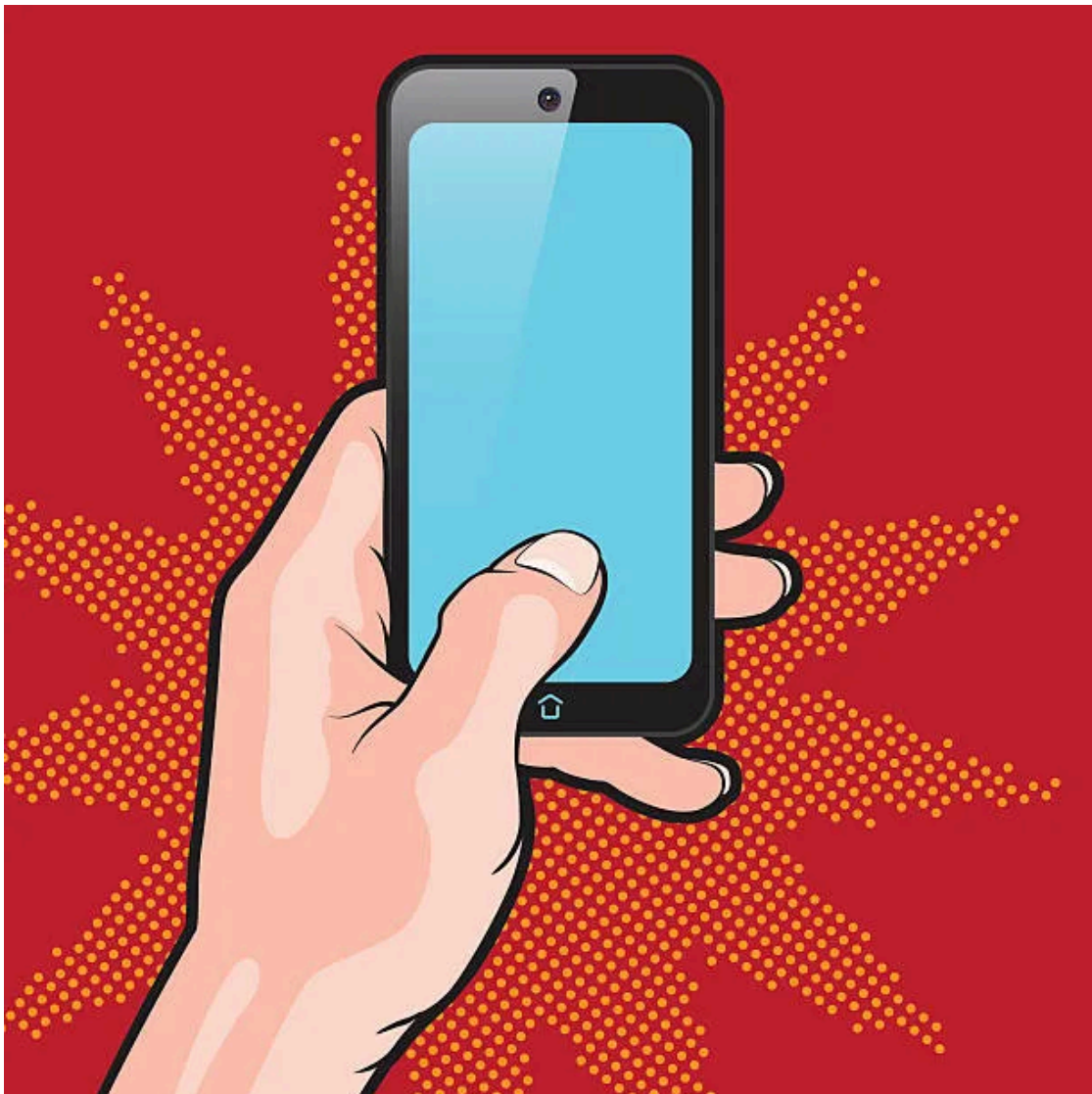



---

Hadrien AH-KANG  
Aaron SALL

# Projet “Mobila”





<b>. I</b>	<b>Identification du projet et ses besoins</b>	<b>p. 3</b>
<b>. II</b>	<b>Les standards du dév. d'appli mobiles</b>	<b>p. 4</b>
<b>. III</b>	<b>Les solutions envisagées</b>	<b>p.</b>
<b>. IV</b>	<b>Notre choix</b>	<b>p.</b>
<b>. V</b>	<b>Mise en place de la solution</b>	<b>p.</b>
<b>. VI</b>	<b>Conclusion</b>	<b>p.</b>

## . I Identification du projet et ses besoins

La société VosRêves souhaite mettre en place une application mobile accessible sur smartphone et tablette.

Pour cela le DSI demande à ses équipes de développeurs de proposer une solution de développement d'application mobile permettant de créer un application hybride (Android+iOS)

L'équipe de développement actuelle est composé de trois personnes : 1 dev html php,

L'équipe en charge devra présenter à la direction un étude exhaustive des différentes alternatives de solutions d'adaptation de code pour les deux systèmes mobiles, avec ces avantages et ces inconvénients et proposer une solution détaillée.

Le DSI souhaiterait avoir une maquette de l'application mobile.

Il s'agit la d'une demande de l'entreprise VosRêves qui nous demande de proposer une solution de développement d'application mobile hybride avec pour cahier des charges :

- étude des solutions
- ses alternatives
- avantages et inconvénients
- présentation de la solution retenue
- application

## . II Les standards du développement d'applications mobiles

### Qu'est-ce qu'une application ?

Une application web désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web.

Contrairement à un logiciel traditionnel, l'utilisateur d'une application web n'a pas besoin de l'installer sur son ordinateur.

Il lui suffit de se connecter à l'application à l'aide de son navigateur favori.

La tendance actuelle est d'offrir une expérience utilisateur et des fonctionnalités équivalentes aux logiciels directement installés sur les ordinateurs.

Les technologies utilisées pour développer les applications web sont les mêmes que celles employées dans la création des sites internet.

### Pourquoi rechercher une solution de développement hybride ?

#### Avantages:

- Moins de problèmes avec l'app store ( pas besoin de soumettre les mise à jours à moins que cela n'affecte le code natif)
- Réduction du coût, du temps et de l'effort ( Le code étant écrit une seule fois il n'est donc pas nécessaire d'en faire d'autre)

#### Limitations:

- Interfaces utilisateurs limités (interface utilisateurs moins fluide qu'une app native, possibilité 3d limités)
- La difficulté ou l'impossibilité d'exploiter toutes les capacités des plateformes ( chaque plateforme ayant des capacités unique il est nécessaires d'utiliser une app native pour pouvoir les utiliser pleinement à contrario une app hybride ne pourra pas utiliser pleinement ce mêmes fonctionnalités)

## Différents langages de programmation :

### Pour Android :

#### Java (natif officiel Android) :

Java est le langage de programmation officiel et le plus populaire pour le développement d'applications Android.

Alors que le code Java lui-même s'exécute généralement sur la machine virtuelle Java, sur la plate-forme Android, le code est compilé par la machine virtuelle Dalvik.

Il s'agit du langage de codage natif utilisé dans Android Studio et d'une plateforme éprouvée pour le développement de logiciels pour smartphones et ordinateurs.

Utiliser Java pour créer toutes sortes d'applications, car il peut être utilisé pour coder l'ensemble du programme.

Il existe de nombreuses ressources d'apprentissage en ligne, et le langage est considéré comme assez simple pour les débutants.

L'un de ses plus grands avantages est que les logiciels créés avec ce langage peuvent être **facilement installés et exécutés sur différents systèmes d'exploitation**, que ce soit Windows, Mac OS, Linux ou Android.

#### Kotlin :

Bien que relativement nouveau dans le monde Android, Kotlin représente une option fantastique pour les développeurs d'applications Android.

Ce langage polyvalent à code source ouvert a été conçu dans un souci de praticité, d'efficacité et d'efficacité.

Kotlin est totalement compatible avec Java et, à ce titre, est rapidement devenu le deuxième langage de programmation officiel d'Android.



Le langage a été conçu dans le but d'être une alternative plus moderne, efficace et pratique à Java.

Ainsi, il présente une mise en page plus logique et une meilleure lisibilité que son concurrent.

Java dispose toujours d'une base d'utilisateurs plus large et est sans doute plus rapide que Kotlin, mais Kotlin offre une saisie plus facile, un code plus court et davantage de fonctions de sécurité.

## C :

Le langage C pose les bases de nombreux langages de programmation.

Java en fait justement partie : c' est le langage utilisé pour le développement Android.

Lorsque l'on a des ressources limitées, le langage C peut être une solution.

Il est utile aux développeurs qui souhaitent concentrer la priorité sur la rapidité de l'app.

Deux autres langages peuvent être employés pour créer une application mobile native Android.

Le langage C est le langage le plus efficace pour les traitements algorithmiques lourds.

Contrairement à Java ou Swift, ce n'est pas un langage orienté objet.

Mais c' est certainement le langage le plus rapide, encore aujourd'hui.

Il permet d'être indépendant des nombreuses bibliothèques actuelles.

Les développeurs l'utilisent pour faire du code efficace en temps et en mémoire.

Les applications en temps réel sont très souvent codées en C.

Elles garantissent le respect des contraintes temporelles autant que fonctionnelles.

Ce sont les applications de pilotage de machines ou de drones, par exemple.

Les applications de loisirs, média ou sociales peuvent être codées en C.

Le langage C convient pareillement au développement mobile du secteur des jeux.

Combiner C avec Unity permet de créer ces applications ludiques facilement.



Le langage C est présent dans la plupart des applications mobiles natives de jeux du Google Play.

Ces apps représentent un gros potentiel financier pour tout porteur de projet.

70 % des revenus des Stores sont générés par les jeux.

Ceux-ci sont proposés via des applications mobiles natives ou hybrides.

### C+ :

C'est un langage informatique standard et normalisé, destiné à interroger ou à manipuler une base de données relationnelle.

Interface de programmation Une interface de programmation permet la communication avec le SGBD, de demander des opérations et de récupérer des données provenant des bases de données.

### C++ (NDK Android) :

Si le C++ n'est pas le premier langage qui vient à l'esprit des débutants, il constitue un excellent choix pour la création d'applications Android plus complexes.

Le C++ est l'un des langages de programmation les plus populaires au monde.

Il est utilisé pour tout, de la conception de jeux vidéo à la réalisation de systèmes d'exploitation, en passant par la production de films en 3D.

Le langage C++ fonctionne en mode natif sur le téléphone et, bien que cela puisse rendre son utilisation plus délicate, il est très polyvalent en termes de fonctionnalités.

Malgré une courbe d'apprentissage assez raide, le C++ est vraiment idéal pour concevoir des jeux compliqués grâce à sa plate-forme native.

## C# :

Combinant les meilleures caractéristiques de C++ et de Java, C# peut être utilisé pour une grande variété de tâches, de la conception de jeux à l'aide d'Unity au développement d'applications Windows complexes.

C# est un membre orienté objet et de niveau supérieur de la famille des langages C, et c'est une option étonnante pour le développement d'Android.

Le langage présente une courbe d'apprentissage raisonnable, une syntaxe plus simple que celle de nombreux autres langages et peut être utilisé à des fins diverses, ce qui en fait un excellent choix pour un langage de programmation Android.

## Lua :

Si vous êtes à la recherche d'un langage facile à utiliser et à code source ouvert pour le développement Android, il est intéressant de considérer Lua.

Il s'agit d'un moteur de jeu entièrement fonctionnel, utile pour créer des jeux simples, mais les jeux plus complexes nécessitent probablement une alternative plus robuste, telle que C++ ou C#.

## BASIC :

Si le nom "BASIC" peut sembler simple, il s'agit en fait d'un acronyme très long. BASIC, ou « Beginners All-Purpose Symbolic Instruction Code », est logique, simple et utile et constitue une excellente option pour les débutants sur Android.


Bien que BASIC ne soit pas un langage de programmation Android universellement pris en charge, vous pourrez néanmoins créer des applications utiles avec ce langage.

C'est un moyen rapide et pratique de commencer, mais nous recommandons tout de même d'utiliser un autre langage pour les programmes plus complexes.

## Outils : Hybride

### React Native :





React Native est un langage multiplateforme, ce qui signifie que vous n'avez pas besoin de créer des applications distinctes pour Android et iOS.

Ce langage de codage robuste a été utilisé pour développer des applications très populaires comme Airbnb et Uber, et est souvent appelé « l'avenir des applications hybrides ».

Avec React Native, vous pouvez écrire des sections de code qui peuvent être utilisées sur différentes plateformes.

Une fois le code déployé, vous pouvez facilement le personnaliser pour les plateformes respectives en détaillant simplement les nuances les plus fines et les éléments de style.


De plus, React Native comprend un certain nombre de composants pré-construits disponibles via une bibliothèque open-source.

Ce langage est idéal pour créer des applications Android plus réactives, plus rapides et plus fluides, et un certain nombre de bibliothèques et de plugins tiers facilitent encore la tâche.

### Flutter (Langage DART) :

Flutter est un cadre de développement d'applications qui utilise le langage de programmation Dart et contient une riche bibliothèque d'outils et de widgets permettant de créer des expériences immersives et très visuelles.

Avec Flutter, vous pouvez créer des applications multiplateformes aussi fluides que les applications natives, et la réutilisation du code signifie que vous pouvez écrire votre code plus rapidement.



En plus d'offrir une vaste bibliothèque de widgets d'interface utilisateur pour vous aider à développer des applications visuellement étonnantes, le framework permet également la personnalisation des widgets, de sorte que vous pouvez obtenir cette riche expérience native que vous recherchez dans un langage de programmation Android.

## Xamarin :

Xamarin est un framework populaire pour le développement multiplateforme qui permet de créer des applications de type natif pour Android et iOS.

Xamarin utilise un seul langage de programmation, le C#, ainsi que le cadre .NET pour créer des applications mobiles répondant à des besoins et des plates-formes variés.

Une pile technologique unique et une base de code partageable au sein de Xamarin permettent un développement plus rapide, car il suffit d'apporter de légères modifications pour diffuser l'application sur différentes plateformes.

Dans le même temps, les API et les boîtes à outils natives répondent efficacement aux performances et à la conception des applications natives, ce qui rend presque impossible la distinction entre une application native et une application Xamarin.

Xamarin présente toutefois un inconvénient notable : il n'est pas très performant pour intégrer des animations et des éléments graphiques riches, ce qui n'en fait pas un choix idéal pour les applications de jeux.

## Ionic :

Ionic est une boîte à outils UI open-source qui permet de développer des applications mobiles hybrides multiplateformes.

Le framework utilise Webview pour le mobile plutôt que d'utiliser des éléments natifs de l'appareil ; il exploite également les bases de code JavaScript courantes, telles que JavaScript, Reach, Angular et Vue.

Ionic est connu pour offrir des performances efficaces avec un minimum de manipulation du DOM et contient de nombreux composants prédéfinis qui offrent une fonctionnalité native.

Si Ionic est un excellent choix pour réduire le temps, les efforts et les ressources nécessaires à la création d'une application multiplateforme, l'utilisation de WebView pour le rendu des applications le rend inadapté à la création d'applications intensives ou complexes.

## Apache Cordova :

Apache Cordova est un framework de développement mobile open source.

Il vous permet d'utiliser des technologies Web standard telles que HTML5, CSS3 et JavaScript pour le développement multiplateforme, en évitant chaque langage de développement natif de plateforme mobile.

Les applications s'exécutent dans des wrappers ciblés sur chaque plateforme, s'appuient sur des liaisons API conformes aux normes pour accéder aux capteurs, aux données et à l'état du réseau de chaque appareil.



## Le SDK

Les outils SDK sont généralement indépendants de la plate-forme et sont nécessaires quelle que soit la plate-forme Android sur laquelle vous travaillez.

Lorsque qu'on installe le SDK Android sur un système, ces outils sont automatiquement installés.

## Le Virtual Device Manager

Il est utilisé pour créer et gérer les AVD (Android Virtual Devices) et pour télécharger les packages du SDK.

L'AVD héberge un émulateur exécutant une version donnée d'Android ; il permet de préciser la version du SDK supportée, la résolution d'écran, l'espace de stockage disponible sur la carte SD et les capacités du matériel disponibles (comme l'écran tactile ou le GPS).

## Pour iOS Apple :

### Swift\* :

Pour certains développeurs, Swift est comme le langage Objective-C sans le C.

Il est très facile à développer et offre en même temps des performances extrêmes à l'application et au processus de développement.

Bien qu'il ne soit pas aussi mature que le langage Objective C, il continue à recevoir des mises à jour, ce qui le rend meilleur avec le temps.

Le meilleur est que de nombreuses nouvelles applications utilisent le langage Objective C pour la programmation, ce qui signifie que les développeurs peuvent coexister et profiter les uns des autres.

### Objective-C :

En ce qui concerne le langage Objective-C, il s'agit d'un autre langage qui mérite d'être essayé si vous vous intéressez au monde du développement iOS.

IT est un langage qui a été utilisé par Apple lorsqu'elle a acquis les produits NeXT dans les années 90.

Le langage est passé par de nombreuses phases de développement.

Il est d'abord apparu comme un sous ensemble orienté objet du langage C.

Par la suite, Apple a intégré ce langage à Mac OS X, ce qui en fait un langage standard pour le développement d'iOS et de macOS.

  
C#

= CROSS PLATFORM

## **Présentation d'Android :**

Android est un système d'exploitation édité par Google pour les appareils embarqués et/ou mobiles, comme les smartphones ou les tablettes.

On le retrouve aussi dans certains GPS, ordinateurs de bord de voitures, dans des télévisions, autoradios, et même des montres.

De plus, de nombreux prototypes d'appareils électroménagers, comme des réfrigérateurs ou des machines à laver, fonctionnant sous Android ont été présentés ces derniers temps, permettant ainsi de lancer une machine à l'aide de son téléphone, ou encore d'être prévenu par son frigo lorsqu'il manque certaines choses.

Le système Android est basé sur un fork du noyau de Linux.

Ce dernier a été modifié pour être plus adapté aux terminaux mobiles ayant peu de puissance de calcul, de mémoire et de batterie.

De fait, certaines librairies standards ne sont pas supportées par le système, et des améliorations ont été apportées sur la gestion de l'énergie.

Les applications sont écrites en Java, et fonctionnent au sein d'une machine virtuelle Dalvik. Cette machine virtuelle a elle aussi été modifiée pour être le plus adapté possible aux appareils de faible puissance.

Ainsi, beaucoup d'efforts ont été fait sur la consommation de mémoire, qui a été largement diminuée par rapport à la machine virtuelle java classique.

## **Mettre son application Android en ligne :**

La création d' applications Android est plus compliquée que celle d'iOS. Ce système d'exploitation est un écosystème open source : il implique de nombreux appareils, composants et logiciels différents.

Pour les développeurs, c'est un environnement de développement assez fragmenté.

De plus, les utilisateurs d'Android ne sont pas tous à jour sur la version la plus récente de la plate-forme d'exploitation.

## **Comment créer une application Android et iOS :**

Android : En premier lieu, il est important de choisir votre cible. Bien connaître vos futurs utilisateurs permet d'orienter les décisions techniques.

Ensuite, vous pouvez définir votre modèle économique.

D'autres étapes de conception sont également à respecter.

Vous devez établir les objectifs de votre application.

Une fois cela décidé, il faudra créer votre prototype ou maquette.

Cela permet de se concentrer sur les parties essentielles de l'interface.

La navigation est un composant clé du succès d'une app Android.

Vous pourrez alors penser aux éléments graphiques, design et styles.

Examinez ce que fait la concurrence directe en matière d'expérience utilisateur.

Créer une app passe par cette étape de prototypage et d'étude de l'écosystème.

L'idée principale est de vous démarquer des apps mobiles cousines.

Le développement proprement dit démarre ensuite.

Une fois votre produit réalisé, une phase de finalisation s'impose.

Avant de l'envoyer sur le PlayStore, elle doit être testée.

Cela fait gagner un temps précieux en évitant les aller-retours pour corriger les bugs.

Et vous avez l'assurance de fournir une appli Android fonctionnelle.

iOS : Si vous avez les compétences nécessaires, programmer des applications natives est possible.

Cela réduit le coût de développement.

Il est cependant indispensable de connaître les langages Swift ou Objective-C.

Cela requiert d'avoir beaucoup de temps, de la patience, et de disposer d'un excellent niveau d'expertise.

Il faut faire preuve d'organisation, suivre pas à pas le processus de création.

Il s'étend du cahier des charges Apple à la promotion de l'app.

Vous devez passer par l'écriture d'un story-board.

Vous aurez également à choisir le design, et à soumettre le projet à la commission de publication de l'App Store.

## . III Les solutions envisagées

A travers des solutions présentées précédemment voici les solutions envisagées pour le développement de notre application hybride.

Liste des frameworks dev app mobile hybride :

Nom	Spécifications
<b>React Native</b> Framework permettant un développement iOS qu' Android utilisé pour des apps natives plutôt que des apps hybrides	<ul style="list-style-type: none"><li>• Compatible avec les plugins tiers</li><li>• Réutilisation du code et rentabilité</li><li>• Facilité d'entretien du code</li></ul>
<b>Flutter</b> Framework qui développe des apps natives rapidement grâce à une gamme de widgets	<ul style="list-style-type: none"><li>• Performance native complète</li><li>• Interface utilisateur flexible</li><li>• Widgets personnalisables</li><li>• Offre de conception de matériaux</li></ul>



	intégrés
<p><b>Xamarin:</b> Le Xamarin, acquis par Microsoft, est un framework open-source basé sur le .NET.</p> <p>Le framework Xamarin permet de développer des applications natives pour Android, iOS et Windows grâce au partage de la base de code commune.</p>	<ul style="list-style-type: none"> <li>• Accès à l'API native et soutien à l'interface utilisateur</li> <li>• Intégration plus facile des API</li> <li>• Développement rapide</li> </ul>
<p><b>Mobile Angular UI</b></p> <p>Framework qui peut transformer une app web en app mobile et qui offre une multitude de composants d'UI comme les barres de navigation ou les zones de défilement tout en fusionnant les frameworks Bootstrap et Angular.</p>	<ul style="list-style-type: none"> <li>• Composants interactifs</li> <li>• Pas de dépendance jQuery</li> </ul>
<p><b>Ionic</b></p> <p>Framework qui peut développer des applications mobiles grâce à des outils comme HTML, CSS ou Javascript.</p>	<ul style="list-style-type: none"> <li>• Développement rapide</li> <li>• Disponibilité des plugins Cordova</li> <li>• Basé sur AngularJS</li> </ul>
<p><b>Adobe PhoneGap</b></p> <p>Framework qui utilise HTML5, CSS3 et Javascript pour développer des applications mobiles.</p>	<ul style="list-style-type: none"> <li>• Bibliothèque de plugins et outils tiers</li> <li>• Multiplateforme</li> </ul>
<p><b>Appcelerator Titanium</b></p> <p>Framework permettant un développement multiplateforme à partir d'une seule base de code Javascript et possède des composants d'interface utilisateur prêts à l'emploi et il est connecté au cloud, ce qui permet d'accéder à son code de n'importe où !</p>	<ul style="list-style-type: none"> <li>• Gestion de l'API</li> <li>• Intégration facile des API</li> <li>• Meilleure réutilisation des codes</li> <li>• Développement rapide</li> </ul>
<p><b>Framework7</b></p> <p>Framework7 est un framework open-source de développement d'applications mobiles, d'apps de bureau et d'apps web.</p> <p>Ce framework permet d'utiliser du HTML, du CSS et du Javascript avec un support</p>	<ul style="list-style-type: none"> <li>• Personnalisation plus facile</li> <li>• Widgets et éléments de l'assurance-chômage</li> <li>• Outils supplémentaires comme Electron &amp; NW.js</li> <li>• Code facile d'entretien</li> </ul>

supplémentaire des composants Vue.js et React et possède un bel écosystème de plugins.	
<b>jQuery Mobile</b> jQuery Mobile est un framework multiplateforme qui permet d'exécuter une seule version de code peu importe l'appareil.  Basé sur HTML5, il est livré avec de multiples plugins il est compatible avec d'autres frameworks de développement d'applications mobiles comme PhoneGap.	<ul style="list-style-type: none"><li>• Prise en charge de plusieurs plateformes et appareils</li><li>• Divers modules complémentaires</li><li>• Framework thématique puissant</li></ul>
<b>NativeScript</b> Native Script est un framework open-source utilisé pour développer des applications mobiles natives en utilisant Angular, Vue.js, TypeScript ou JavaScript.  Les applications développées avec ce framework utilisent les mêmes API que celles de Xcode ou d'Android Studio.	<ul style="list-style-type: none"><li>• Développement multiplateforme</li><li>• Accès aux API Android et iOS</li><li>• Réutilisation du code</li></ul>

<https://mobiskill.fr/blog/conseils-emploi-tech/top-10-des-meilleurs-frameworks-de-developpement-mobile/>

## . IV Notre choix

Notre choix se portera sur **Apccelerator Titanium** pour plusieurs raisons :

Avantage	Inconvénients
<p><b>Développement multiplateforme:</b> Vous pouvez réutiliser 60 à 90% du code sur différentes plates-formes. Cela économise énormément de temps.</p>	<ul style="list-style-type: none"> <li>● <b>Il y a un prix:</b> Bien que ce framework offre une version gratuite, il n'a pas toutes les fonctionnalités et il ne vous donne pas beaucoup de flexibilité. Vous devez dépenser de l'argent sur la version premium pour obtenir toutes les fonctionnalités de Titanium.</li> </ul>
<p><b>Palette glisser-déposer:</b></p> <ul style="list-style-type: none"> <li>● Le concepteur d'applications de Titanium comprend une palette glisser-déposer qui vous permet de concevoir visuellement vos applications sans effort. Il produit un code de haute qualité. Grâce à ce framework, vous pouvez facilement créer des effets visuels tels que des animations dynamiques.</li> </ul>	<ul style="list-style-type: none"> <li>● <b>Difficile de créer des applications complexes:</b> L'un des principaux inconvénients de l'utilisation de Titanium est qu'il est difficile de créer des applications mobiles qui impliquent beaucoup de complexité. [19659023] Problèmes avec le SDK: Vous pouvez rencontrer des problèmes liés au SDK avec Titanium, ce qui vous fera perdre votre temps et vos efforts.</li> </ul>
<ul style="list-style-type: none"> <li>● <b>Service de notification push prédéfini:</b> Vous pouvez engager les utilisateurs de votre application en utilisant le service de notification push de ce framework.</li> </ul>	<ul style="list-style-type: none"> <li>● Comme vous <b>ne pouvez pas ajouter de code HTML et CSS</b>, les éléments DOM et les animations peuvent devenir moins réactives.</li> <li>●</li> </ul>

<ul style="list-style-type: none"><li>● <b>Analyse mobile en temps réel:</b> Titanium vous fournit des données d'analyse mobile en temps réel qui vous aident à réduire les taux de plantage des applications, à améliorer la disponibilité et à résoudre les problèmes. Une fonctionnalité de Titanium appelée Custom Analytics vous permet d'exécuter des requêtes personnalisées sur les données d'analyse, et la fonction Mobile Lifecycle Dashboard vous aide à améliorer la collaboration et à réduire la dépendance aux rapports manuels et aux outils ponctuels.</li></ul>	
<ul style="list-style-type: none"><li>● <b>Excellentes performances:</b> Les applications Titanium sont des applications entièrement natives. Une fonctionnalité appelée Hyperloop vous permet d'accéder directement aux API natives afin que vous puissiez facilement implémenter des fonctionnalités spécifiques à la plate-forme dans votre application. L'accès direct aux API natives vous donne une amélioration des performances. Vous pouvez également exécuter du code natif tel que du code Objective C, Swift ou Java en plus du code Titanium, ce qui augmente étonnamment la vitesse de l'application.</li></ul>	

<ul style="list-style-type: none"> <li>• <b>API Builder:</b> Titanium's API Builder vous permet d'assembler des modèles, des API et des connecteurs pour l'accès aux données, et vous permet de créer des modèles de données par programme et via un assistant visuel. Vous pouvez normaliser vos données et les livrer au client. API Builder vous fournit des connecteurs prédéfinis pour MS SQL, Box, Mongo DB, Salesforce, MySQL, etc.</li> </ul>	
Connecté au cloud donc le code est accessible de n'importe où	
grosse communauté permettant d'avoir de l'aide au besoin	
assez facile à apprendre	
de plus en plus de modules	
Très bon tutoriel en ligne	

Pour argumenter plus ce choix nous pouvons dire qu'en plus de pouvoir travailler dessus depuis n'importe où il est facile à apprendre cela permettra à quelqu'un d'autre de modifier l'application au besoin ainsi que sa grande bibliothèque et communauté permettant de nombreuses choses lors de la création de l'application et enfin ses hautes fonctionnalités et performances ayant pour seuls problèmes la complexité et le prix sachant que nous n'avons pas besoin d'application complexes pour la nôtre et que le succès de notre application et peut-être même celle du futur couvriront les frais du framework.

## V. Mise en place de la solutions

Nous partons du principe que la direction a déjà validé notre solution.

### Etape 1:

Créer un Prototype de l'application puis le présenter à la direction sinon corriger

### Etape 2:

Après validation du prototype élaboration du design ainsi que sa présentation à la direction sinon corriger

### Etape 3:

Après cette validation finalisation de l'application

### Etape 4:

Lancement de la phase de test et correction d'éventuels bugs

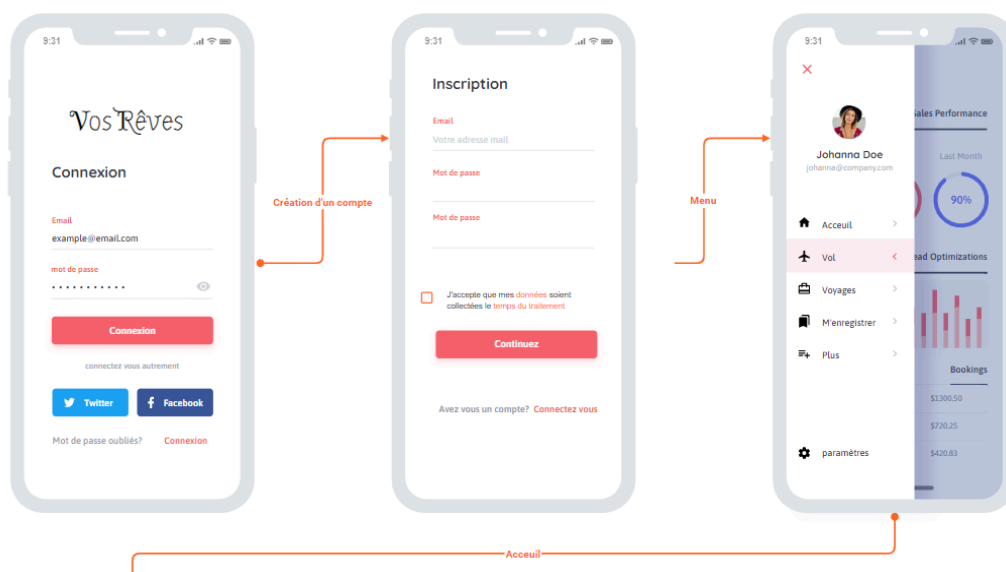
### Etape 5:

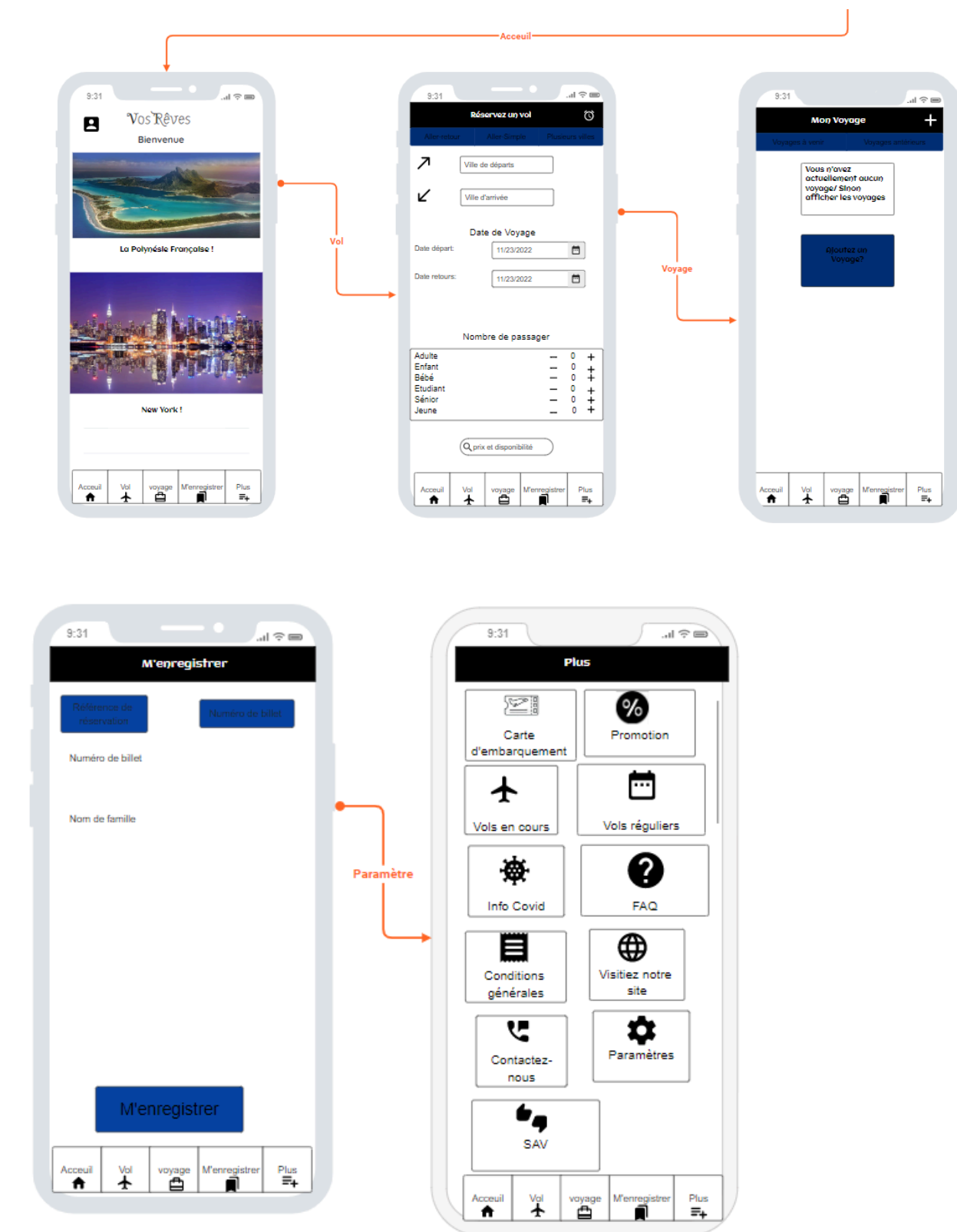
Promouvoir l'application pour son lancement

### Etape 6:

Lancer officiellement l'application

## Schéma de l'application





Plus de détails:

<https://app.moqups.com/gOaasH9wZzKe6fS7XfGRNxASbLwLTCaP/view/page/a21ab8119>

## .VI Conclusion

**Hadrien :** Pour conclure ce projet je dirai que cela m'a beaucoup apporté personnellement pour plusieurs raisons:

-la première et que je m'intéresse de près au développement d'applications mobiles je trouvais le sujet intéressant.

-la seconde est que j'ai pris plaisir à le faire est qu'il s'agit pour moi de quelque chose d'essentiel au travail.

-et enfin troisièmement j'ai appris beaucoup de choses sur l'univers des applications et du développement ainsi que des compétences sur le travail pour un projet et le travail en équipe.

A travers cette conclusion j'espère avoir pu exprimer mon ressenti sur ce projet.

**Aaron:** En conclusion, ce projet m'a apporté beaucoup de satisfactions, la première étant le travail et les recherches en autonomie à savoir quels éléments je devrais aborder dans le projet.

Ensuite, il y a le travail en groupe, établir un plan du projet avec mon binôme et progresser.