Table des matières

Table des figures

Glossaire et acronymes

AOT Ahead Of Time. 16, 23

framework Un framework est une sorte d'infrastructure de développement, il désigne un ensemble cohérent de composants logiciels structurels qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel ou application. 17

JIT Just In Time. 23

JSX JavaScript XML. 23

XML Extensible Markup Language. 23

Chapitre 1

Généralités

1.1 Introduction au sujet

Ces dernières années, la communauté des lecteurs algériens a connu une croissance rapide et le taux de participation aux salons du livre est prometteur. Avec cette quantité de livres achetés, de nouveaux problèmes voient le jour, en plus des problèmes déjà existants tels que : les prix relativement élevés de certains livres, la rareté de certains autres et la tâche difficile que les lecteurs doivent endurer pour s'identifier avec d'autres membres qui partage les mêmes intérêts et la même zone géographique, se pose le problème de stockage de ces livres et du fait qu'ils peuvent être lu par un nouveau lecteur que de prendre de la place sur une étagère quelque part.

Il existe des clubs de lecture et des réunions sociales où les gens échangent, vendent, donnent des livres et même rencontrent de nouvelles personnes, ce qui laisse penser qu'il existe un groupe de personnes qui :

- Ont des livres.
- Sont disposés à abandonner les livres déjà lus.
- Veulent économiser de l'argent tout en lisent de nouveau livres

Comme nous venons de le dire, certaines solutions existent pour cette communauté et nous en explorerons certaines dans les sections suivantes tout en soulignant leurs lacunes et en expliquant comment une fenêtre d'opportunités est encore ouverte pour résoudre le problème dans ces différentes aspects.

1.2 Solutions déjà existants

1.2.1 Non technologiques

L'analyse de cette question donnera lieu à deux concepts intéressants qui existent déjà et qui sont, dans une certaine mesure, fonctionnels :

1. Événements d'échange de livres

Certains clubs de lecture et cafés organisent des événements où les participants sont invités à apporter les livres qu'ils souhaitent transmettre aux autres lecteurs et à en obtenir de nouveaux dans une atmosphère conviviale.

Avantages

- Les participants peuvent vivre toute l'expérience humaine d'échanger un livre.
- Les participants découvrent de nouveaux livres par des personnes autres que des simple avis en ligne.
- Les participants établissent des liens significatifs avec leurs collègues lecteurs de livres.

Inconvénients

- De tels événements durent au mieux 2 jours, ce qui fait que beaucoup de gens ratent l'occasion.
- En raison de la limitation géographique, de tels événements ne peuvent être accessibles que par quelques résidents proches de la région.
- coûteux en terme de temps investi.

2. Échanges informels de livres

Certains échanges de livres sont informels - une étagère ou une boîte est fournie où les livres peuvent être laissés ou ramassés. L'échange repose sur les utilisateurs qui sortent et prennent des livres et n'est généralement pas supervisé.

C'est une pratique courante dans les auberges de jeunesse où les voyageurs peuvent laisser un livre et emporter un livre différent avec eux. Certaines gares ferroviaires en Grande-Bretagne ont des échanges de livres informels et une a également été installée dans une cabine téléphonique à Kington Magna[7].

Avantages

- Le processus est très rapide et pratiquement pas de temps perdu.
- Absence de limite de temps, ils peuvent être échangés à tout moment.

Inconvénients

- Le besoin de donateurs au début du projet.
- L'absence d'un système de contrôle de qualité pour les livres mis contre les prises.

FIGURE 1.1 – Un "échange de livres de rue" à Washington Heights, New York[7]

1.2.2 Technologiques

Il existe plusieurs sites Web et très peu d'applications populaires offrant le type de bonne expérience présente dans la manière traditionnelle, nous allons explorer certaines des plus populaires solutions existant aujourd'hui.

1. BookMooch :

Portant bien son nom, avec le fameux slogan *"Give books away, Get books you want"*,BookMooch est une société de publication delivres en ligne par laquelle ses membres peuvent échanger des livres entre eux. Son fondateur, John Buckman, a choisi ce nom pour l'entreprise en référence à l'acte de donner un livre sans attendre de le récupérer. Les utilisateurs, alors, sont considérés comme des BookMoochers et bénéficient à bien des égards de ce système commercial unique.

FIGURE 1.2 – Logo de BookMooch

En bref, les utilisateurs «achètent» les livres des autres membres en utilisant uniquement des points. Chaque membre peut accumuler des points en soumettant les titres des livres qu'il veut donner et en envoyant ses livres à d'autres utilisateurs. Avec suffisamment de points, ils peuvent alors commencer à recevoir les titres d'autres personnes, et le processus continue à partir de là. Le seul coût pour les membres est celui de l'envoi des livres qu'ils envoient à d'autres.[2]

Avantages

- L'adhésion à cette entreprise est gratuite.
- Vous pouvez choisir parmi une grande variété de livres.
- Une liste de souhaits connectée à amazon où vous recevrez des notifications lorsque des livres sont disponibles.

Inconvénients

- Pour recevoir, il faut donner.
- Dans une certaine mesure, la plate-forme peut être un terrain de jeu pour les fraudeurs.
- La plate-forme n'exploite pas l'emplacement des membres.

2. Bookup :

L'application Bookup est destinée aux lecteurs de livres imprimés qui souhaitent échanger leurs livres contre d'autres livres de leur région, gratuitement, et éventuellement se faire un nouvel ami partageant le même intérêt pour la lecture. Bookup fournit une fonctionnalité d'échange de livre en fonction de l'emplacement.[3]

FIGURE 1.3 – Logo de Bookup

Avantages

- La plate-forme exploite l'emplacement des membres.
- Fonctionnalité de bibliothèque personnelle que vous pouvez personnaliser à votre guise.
- Liberté absolue pour les utilisateurs de discuter en temps réel et de trouver un moyen d'exécuter l'échange.
- Un mécanisme simple d'exploration aléatoire de livres proches.

Inconvénients

- L'application est disponible uniquement sur les appareils marchent avec iOS.
- Une expérience utilisateur moyenne.
- Un manque évident d'informations sur les livres et les utilisateurs.

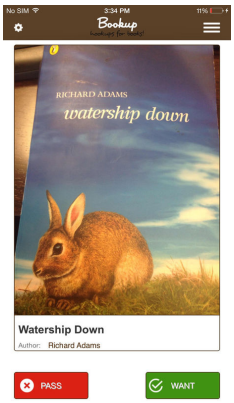

FIGURE 1.4 – Le mecanisme de suggestion et decouverte offert par Book Up ou l'utilisateur parcours une pile de cartes des livres près de son zone géographique on choisissent "passer" ou "vouloir"

3. Bookabikia

Bookabikia est une plate-forme et un service fabriqués en Égypte qui permettent aux gens d'échanger des livres. L'équipe derrière le projet a fait un excellent travail en résolvant non seulement les problèmes évidents auxquels la communauté des lecteurs est confrontée, mais également en l'exploitant avec brio. Sur le site Bookabkia, vous ferez une demande de don pour le site avec vos livres. L'équipe de livraison viendra à vous pour prendre les livres dans les jours à venir et 24 heures après avoir reçu les livres, vous obtiendrez un crédit qui se présente sous forme de points, avec ce crédit, vous pourrez acheter plusieurs livres en même temps sans avoir besoin d'intérêt personnel pour vos livres.[1]

FIGURE 1.5 – Logo de Bookabikia

Avantages

- Les utilisateurs ne sont pas obligés de passer des offres d'échange avec d'autres utilisateurs.
- La fonction de crédit donne beaucoup de liberté de choix aux gens.
- Tous les livres sont expédiés au domicile de l'utilisateur.

Inconvénients

- La plate-forme ne permet pas les interactions peer-to-peer entre les membres de la communauté.
- Le service n'est pas totalement gratuit.
- L'expérience utilisateur n'est pas personnalisable.

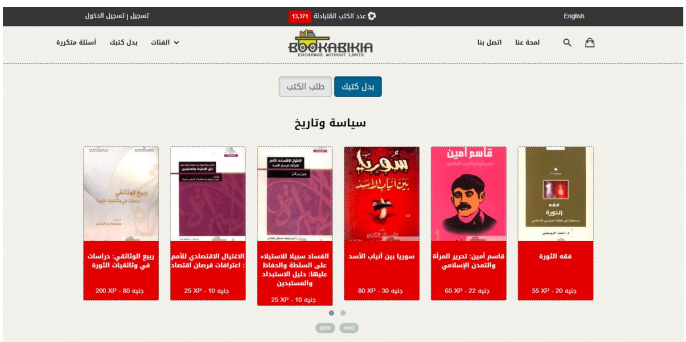

FIGURE 1.6 – Page d'acceuil de Bookabikia

1.3 Solution proposé

Aussi merveilleux que les solutions existantes sont, nous pensons pouvoir encore offrir une meilleure solution et une meilleure expérience, en exploitant deux aspects de la pratique de l'échange de livres :

- La sociabilité humaine et la capacité à s'identifier aux autres et à créer de petites communautés et des relations durables et significatives.
- Une interface utilisateur magnifique et une expérience utilisateur personnalisable qui permettent une totale liberté d'interaction avec d'autres utilisateurs.

La solution consiste en une application mobile fonctionnant à la fois sur Android et sur iOS (97,43% des appareils du monde [4]), où les utilisateurs enregistrés peuvent :

- Créez un profil et répertoriez tous les livres qu'ils possèdent et sont disposés à échanger, prêter, vendre ou même faire un don à un autre utilisateur de leur choix. Le tout dans une interface utilisateur magnifiquement structuré.
- Recherchez les livres qu'ils souhaitent acquérir en utilisant des mécanismes de recherche avancés qui permettent d'obtenir les résultats les plus homogènes, les plus fonctionnels et les plus pratiques.
- Explorez les utilisateurs à proximité (propriétaires de livres) a l'aide de la map, parcourez leurs bibliothèques d'un simple balayage et d'un clic.
- Envoyez et recevez des messages en temps réel avec le service de messagerie, sans aucune limitation.

Enfin, l'application portera le nom «Bindex», qui est une forme abrégée d'index de livre qui résume le travail que l'application accomplit en indexant toutes les bibliothèques personnelles et en les mettant au bout du doigt d'un utilisateur donné.

FIGURE 1.7 – Logo de Bindex

1.4 Plan du rapport

Après avoir passé en revue les généralités et l'introduction, nous plongerons dans la manière dont ce projet prospère va prendre vie :

Dans le deuxième chapitre, nous aborderons les technologies et les solutions les plus appropriées pour la réalisation de l'application mobile, en mentionnant également certains services de base utilisés, et dans le chapitre suivant, nous aborderons les processus qui ont été pris en compte pour la création. de l'application, de la conception initiale et brouillon au prototypage, tout en soulignant les outils utilisés, ainsi que les mécanismes de co-travail et les outils qui ont permis de faire de l'ensemble du projet une expérience fluide.

Après cela, nous plongerons encore plus profondément dans une démonstration où nous présenterons toutes les fonctionnalités qui ont été développées et comment nous prévoyons de nous développer à l'avenir.

Chapitre 2

Les technologies utilisées

2.1 Un bref historique du développement d'applications mobiles

Le développement d'applications mobiles est l'acte ou le processus par lequel une application est développée pour les appareils mobiles. Ces applications peuvent être préinstallées sur les téléphones au cours de la fabrication ou accessibles via un navigateur Web [9]. Toutefois, au cours de la dernière décennie [10], des développeurs tiers ont été capables de créer des applications mobiles. Cependant, en raison de la concurrence intense dans les logiciels mobiles et des modifications apportées à chacune des plateformes, ces développeurs doivent prendre en compte un large éventail de tailles d'écran, de spécifications matérielles et de configurations.

Le développement d'applications mobiles est un domaine d'activité relativement récent. il n'est donc pas surprenant que les outils évoluent encore.

2.1.1 Les kits de développement de plate-forme (Platform SDKs)

Le SDK Apple iOS est sorti en 2008 et le SDK Google Android en 2009. Ces deux SDK étaient basés sur des langages différents : Objective-C et Java, respectivement. Votre application communique avec la plateforme pour créer des widgets ou accéder à des services tels que la caméra. Les widgets sont rendus dans un canevas d'écran et les événements sont renvoyés aux widgets. C'est une architecture simple, mais vous devez créer des applications séparées pour chaque plate-forme car les widgets sont différents, sans parler des langues natives[10].

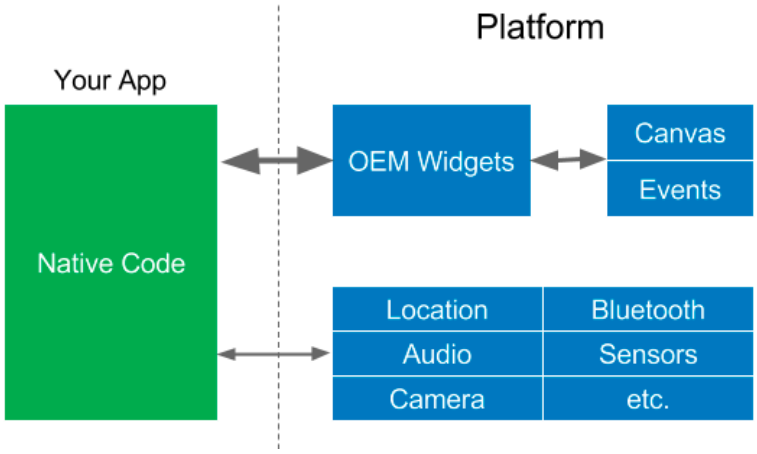

FIGURE 2.1 – Architecture du développement mobile a l'aide des SDK[10]

2.1.2 Vues Web (WebViews)

Les premiers frameworks multi-plateformes étaient basés sur JavaScript et WebViews. Les exemples incluent une famille de frameworks liés : PhoneGap, Apache Cordova, Ionic, etc. Avant de publier leur SDK iOS, Apple avait encouragé les développeurs tiers à créer des applications Web pour iPhone. Il était donc évident de créer des applications multiplates-formes à l'aide de technologies Web.

Votre application crée du HTML et l'affiche dans une vue Web sur la plateforme. Notez qu'il est difficile pour des langages tels que JavaScript de parler directement au code natif (comme les services), de sorte qu'ils passent par un «pont» qui fait que le contexte bascule entre le domaine JavaScript et le domaine natif. Comme les services de plate-forme ne sont

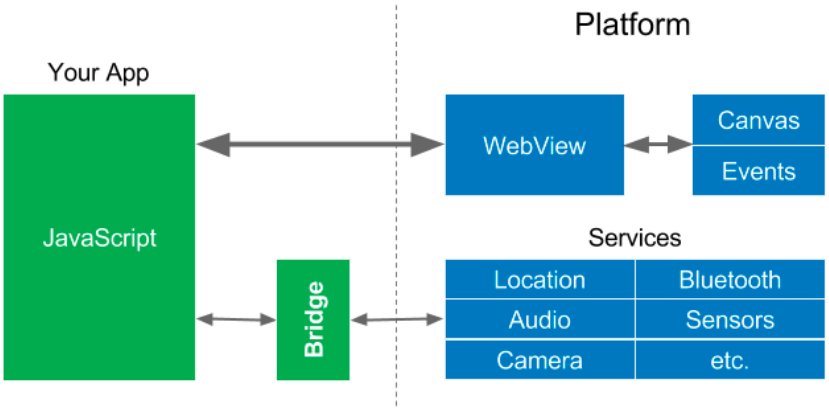

FIGURE 2.2 – Architecture du développement mobile a l'aide des WebViews[10]

généralement pas appelés très souvent, cela n'a pas causé trop de problèmes de performances[10].

2.1.3 Vues réactives (Reactive Views)

Les infrastructures Web réactives telles que ReactJS (et d'autres) sont devenues populaires, principalement parce qu'elles simplifient la création de vues Web grâce à l'utilisation de modèles de programmation empruntés à la programmation réactive. En 2015, React Native a été créé pour apporter les nombreux avantages des vues réactives aux applications mobiles.

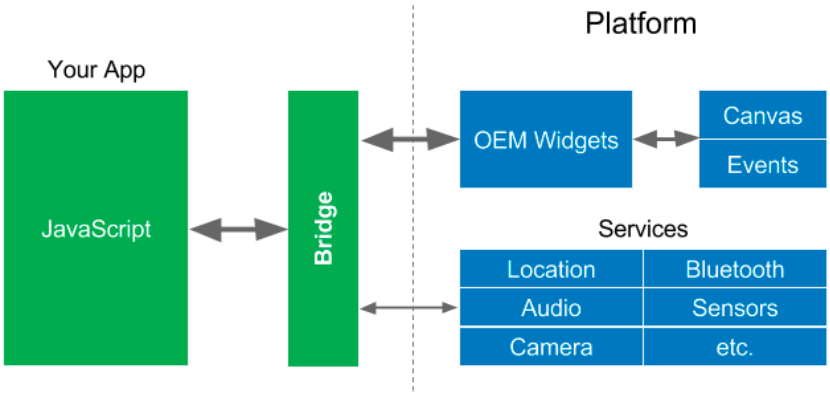

FIGURE 2.3 – Architecture du développement mobile a l'aide des Reactive Views[10]

React Native est très populaire (et mérite de l'être), mais comme le domaine JavaScript accède aux widgets de la plate-forme dans le domaine natif, il doit également passer par le pont. Les widgets sont généralement utilisés assez fréquemment (jusqu'à 60 fois par seconde lors d'animations, de transitions ou lorsque l'utilisateur balaie quelque chose sur l'écran avec son doigt), ce qui peut entraîner des problèmes de performances.

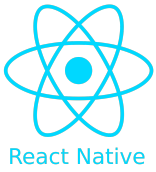

FIGURE 2.4 – Logo de React Native

2.2 Flutter

2.2.1 Qu'est-ce que Flutter ?

Flutter est un SDK pour applications mobiles permettant de créer des applications hautes performances et haute fidélité pour iOS et Android à partir d'une seule base de code[5].

L'objectif est de permettre aux développeurs de proposer des applications hautes performances qui se sentent naturelles sur différentes plates-formes. Nous adoptons des différences dans les comportements de défilement, la typographie, les icônes, etc.

FIGURE 2.5 – Logo de Flutter

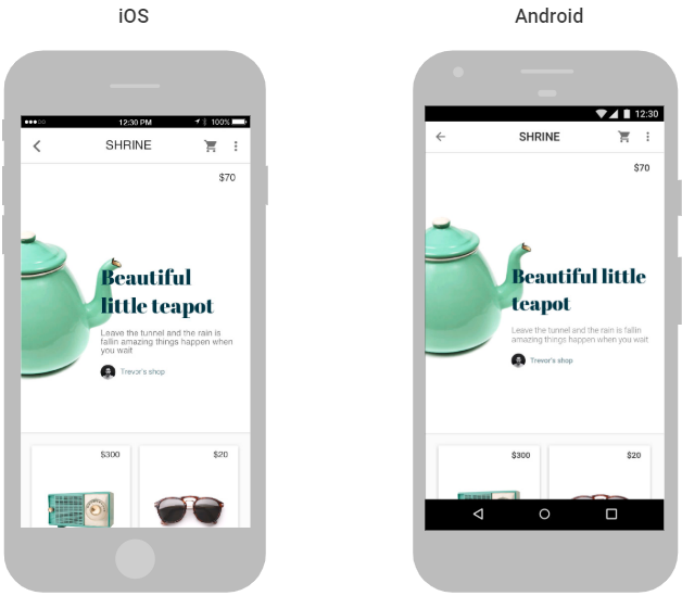

FIGURE 2.6 – Ceci est une application de démonstration nommée Shrine[5]

Comme React Native, Flutter fournit également des vues de style réactif. Flutter adopte une approche différente pour éviter les problèmes de performances causés par la nécessité d'un pont JavaScript en utilisant un langage de programmation compilé, à savoir Dart. Dart est compilé «à l'avance» AOT en code natif pour plusieurs plates-formes. Cela permet à Flutter de communiquer avec la plate-forme sans passer par un pont JavaScript qui effectue un changement de contexte. La compilation en code natif améliore également les temps de démarrage des applications.

Flutter a une nouvelle architecture qui inclut des widgets qui ont l'air agréable, qui sont rapides, personnalisables et extensibles. il n'utilise pas les widgets de plate-forme (ou DOM WebViews), il fournit ses propres widgets.

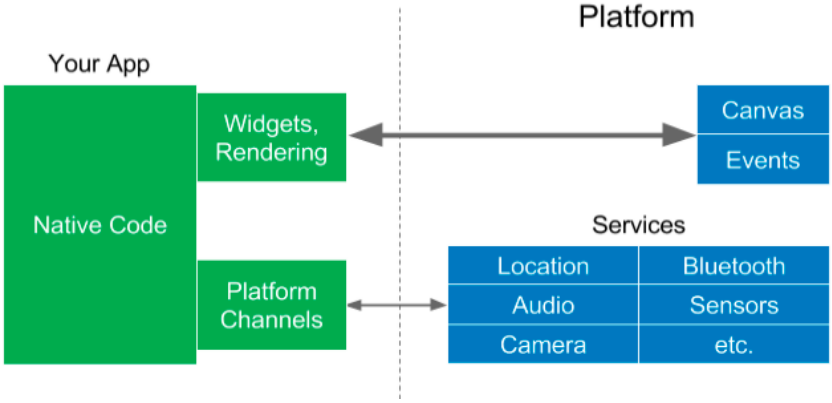

FIGURE 2.7 – Architecture du développement mobile a l'aide de Flutter[10]

Flutter soulève les widgets et le moteur de rendu de la plateforme dans l'application, ce qui leur permet d'être personnalisables et extensibles. Tout ce que Flutter a besoin de la plate-forme est un canevas dans lequel les widgets doivent être rendus afin qu'ils puissent apparaître sur l'écran du périphérique, ainsi que l'accès aux événements (touches, minuteries, etc.) et aux services (localisation, caméra, etc.).

Il existe toujours une interface entre le programme Dart (en vert) et le code de la plate-forme native (en bleu, pour iOS ou Android) qui effectue l'encodage et le décodage des données, mais cela peut être beaucoup plus rapide qu'un pont JavaScript.

2.2.2 Principes de base

Flutter comprend un framework de style réact moderne, un moteur de rendu 2D, des widgets prêts à l'emploi et des outils de développement. Ces composants fonctionnent ensemble pour vous aider à concevoir, créer, tester et déboguer des applications. Tout est organisé autour de quelques principes fondamentaux.

1 - Tout est un widget

Les widgets sont les éléments de base de l'interface utilisateur d'une application Flutter. Chaque widget est une déclaration immuable d'une partie de l'interface utilisateur. Contrairement aux autres frameworks qui séparent les vues, les contrôleurs de vue, les présentations et d'autres propriétés, Flutter possède un modèle objet cohérent et unifié : le widget. Un widget peut définir :

- un élément structurel (comme un bouton ou un menu).
- un élément stylistique (comme une police ou un jeu de couleurs).
- un aspect de la mise en page (comme le rembourrage).
- etc...

Les widgets forment une hiérarchie basée sur la composition. Chaque widget niche à l'intérieur et hérite des propriétés de son parent. Il n'y a pas d'objet «application» séparé. Au lieu de cela, le widget racine joue ce rôle. Vous pouvez répondre à des événements, comme une interaction utilisateur, en indiquant au cadre de remplacer un widget de la hiérarchie par un autre. La structure compare ensuite les nouveaux et les anciens widgets et met efficacement à jour l'interface utilisateur.

Composition> héritage

Les widgets sont souvent eux-mêmes composés de nombreux petits widgets à usage unique qui se combinent pour produire des effets puissants. Par exemple, Container, un widget couramment utilisé, est composé de plusieurs widgets responsables de la mise en page, de la peinture, du positionnement et du dimensionnement. Plus précisément, Container est composé des widgets LimitedBox, ConstrainedBox, Align, Padding, DecoratedBox et Transform. Plutôt que de sous-classer Container pour produire un effet personnalisé, vous pouvez composer ces widgets simples, ainsi que d'autres, de manière innovante.

La hiérarchie des classes est peu profonde et large afin de maximiser le nombre possible de combinaisons.

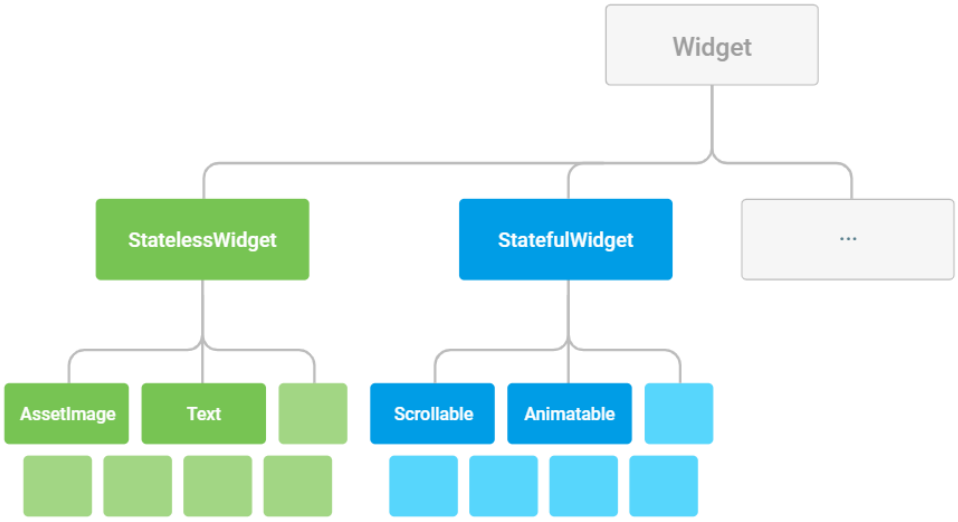

FIGURE 2.8 – Exemple d'une hierarchy des widgets[5]

Vous pouvez également contrôler la mise en page d'un widget en le composant avec d'autres widgets. Par exemple, pour centrer un widget, vous l'enroulez dans un widget Centre. Il existe des widgets pour le remplissage, l'alignement, les lignes, les colonnes et les grilles. Ces widgets de présentation ne possèdent pas de représentation visuelle. Au lieu de cela, leur seul objectif est de contrôler un aspect de la présentation d'un autre widget. Pour comprendre pourquoi un widget est rendu d'une certaine manière, il est souvent utile d'inspecter les widgets voisins.

Couches de framework Flutter

Le framework Flutter est organisé en une série de couches, chaque couche s'appuyant sur la couche précédente.

Les couches supérieures du cadre sont utilisées plus fréquemment que les couches inférieures.

Le but de cette conception est de vous aider à faire plus avec moins de code. Par exemple, le calque Matérial est construit en composant les widgets de base à partir du calque Widgets, et le calque Widgets lui-même est construit en orchestrant des objets de niveau inférieur à partir du calque de rendu. Les couches offrent de nombreuses options pour créer des applications.

Choisissez une approche personnalisée pour libérer toute la puissance d'expression du framework, ou utilisez des blocs de construction de la couche

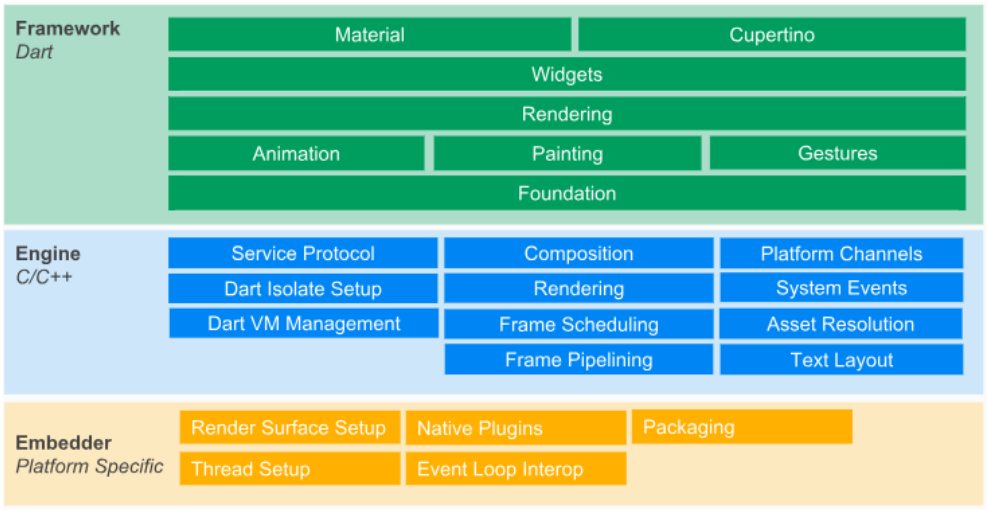

FIGURE 2.9 – L'architecture de flutter[5]

de widgets, ou combinez-les. Vous pouvez composer les widgets prêts à l'emploi fournis par Flutter ou créer vos propres widgets personnalisés à l'aide des mêmes outils et techniques que ceux utilisés par l'équipe Flutter pour créer le framework.

2 - La construction de widgets

Vous définissez les caractéristiques uniques d'un widget en implémentant une fonction de génération qui renvoie une arborescence (ou une hiérarchie) de widgets. Cet arbre représente la partie de l'interface utilisateur du widget de manière plus concrète. Par exemple, un widget de barre d'outils peut avoir une fonction de génération qui renvoie une disposition horizontale de texte et de divers boutons. Le framework demande ensuite de manière récursive à chacun de ces widgets de construire jusqu'à ce que le processus se transforme en widgets entièrement concrets, que le framework assemble ensuite en un arbre.

La fonction de génération d'un widget doit être exempte d'effets secondaires. À chaque fois qu'il est invité à construire, le widget doit renvoyer une nouvelle arborescence de widgets, quel que soit ce que le widget a précédemment renvoyé. Le cadre fait beaucoup pour comparer la version précédente à la version actuelle et pour déterminer les modifications à apporter à l'interface utilisateur.

Cette comparaison automatisée est assez efficace, permettant des appli-

cations interactives hautes performances. Et la conception de la fonction de construction simplifie votre code en vous concentrant sur la déclaration d'un widget, plutôt que sur la complexité de la mise à jour de l'interface utilisateur d'un état à un autre.

3 - Gestion de l'interaction utilisateur

Si les caractéristiques uniques d'un widget doivent être modifiées en fonction de l'interaction de l'utilisateur ou d'autres facteurs, ce widget est avec état. Par exemple, si un widget a un compteur qui s'incrémente chaque fois que l'utilisateur appuie sur un bouton, la valeur du compteur correspond à l'état de ce widget.

Lorsque cette valeur change, le widget doit être reconstruit pour mettre à jour l'interface utilisateur.

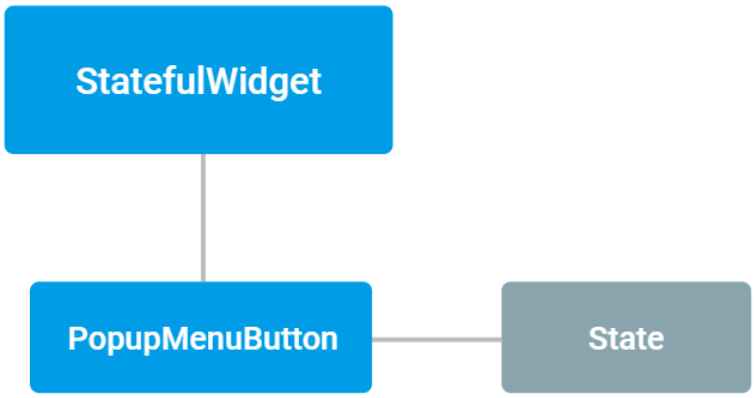

FIGURE 2.10 – Modele d'une Widget Stateful[5]

Ces widgets sous-classent StatefulWidget (plutôt que StatelessWidget) et stockent leur état mutable dans une sous-classe d'État.

Chaque fois que vous modifiez un objet State (par exemple, incrémentez le compteur), vous devez appeler setState() pour indiquer au framework de mettre à jour l'interface utilisateur en appelant à nouveau la méthode de compilation de State.

Avoir des objets state et widget séparés permet à d'autres widgets de traiter les widgets sans état et avec état de la même manière, sans craindre de perdre un état. Plutôt que de devoir conserver un enfant pour préserver son état, le parent est libre de créer une nouvelle instance de l'enfant sans

perdre son état persistant. La structure fait tout le travail de recherche et de réutilisation des objets d'état existants, le cas échéant.

2.2.3 Pourquoi utiliser Flutter ?

Comme nous l'avons mentionné précédemment, il existe de nombreuses options pour développer une application mobile, qu'il s'agisse de SDK natifs ou de frameworks multiplates-formes (Xamarin, ReactNative, Ionic, PhoneGap). Pour ce projet particulier, le framework Flutter a été choisi pour plusieurs points, dont les principaux sont :

1. Multiplate-forme (cross-platform)

Un produit ou système informatique multiplate-forme est un produit ou système pouvant fonctionner sur plusieurs types de plates-formes ou d'environnements d'exploitation.

Différents types de systèmes multiplates-formes incluent des systèmes matériels et logiciels, ainsi que des systèmes qui impliquent des versions séparées pour chaque plate-forme, ainsi que d'autres systèmes plus vastes conçus pour fonctionner de la même manière sur plusieurs plates-formes. La plateforme croisée est également connue en tant que plate-forme indépendante[6]

Flutter offre la possibilité de créer une application compilée pour fonctionner à la fois sur Android et iOS à partir de la même base de code. Tout en maintenant l'expérience utilisateur appropriée sur chaque plate-forme.

2. Productivité

Flutter offre une augmentation de la productivité que provient du «rechargement à chaud» de Flutter («Stateful Hot Reload» et «Hot Restart». Ensemble, ils permettent aux développeurs de voir les modifications apportées à l'état d'une application en moins d'une seconde ; structure de l'application en moins de dix.

Il n'est pas nécessaire d'exécuter une autre version de Gradle. Vous voyez vos modifications dès que vous enregistrez. Pour les développeurs, cela est souvent très facile à maîtriser - il n'ya que peu ou pas de courbe d'apprentissage liée à l'utilisation du «rechargement à chaud» car, par défaut, cela se produit chaque fois que vous enregistrez. Cependant, les avantages sont essentiels. Le temps de développement est souvent réduit de 30% à 40%, car les délais de reconstruction de Gradle qui ralentissent le développement des développeurs Android prennent généralement plus de temps à chaque modification appliquée.

3. Intégration directe avec Firebase

Firebase fournit un support prêt à l'emploi pour un ensemble de services tels que le stockage en nuage, les fonctions de nuage, les bases de données en temps réel, l'hébergement, l'authentification et bien plus encore. Votre infrastructure est instantanément sans serveur, redondante et évolutive. Cela signifie que vous n'avez pas à passer beaucoup de temps et de ressources à construire le backend. Il est également simple de le combiner avec un outil permettant d'automatiser votre processus de développement et de publication, tel que Fastlane ; faciliter la livraison continue. Par conséquent, vous n'avez pas besoin de support DevOps dédié dans votre équipe.

4. Performance

L'application est compilée à l'avance en code ARM natif, pas au moment de l'exécution, comme dans React Native. Cela donne de meilleures performances car il n'ya pas de pont JS au milieu pour analyser et exécuter le code.

5. L'utilisation de Dart

Dart est un langage de programmation polyvalent développé à l'origine par Google et ensuite approuvé en tant que norme par Ecma (ECMA-408). Il est utilisé pour créer des applications Web, serveur, bureau et mobiles.

FIGURE 2.11 – Logo de Dart

Dart est un langage basé sur les objets, orienté objet, défini par la classe et utilisant une syntaxe de style C qui transcompile éventuellement en JavaScript. Il prend en charge les interfaces, mixins, classes abstraites, génériques réifiés, typage statique et un système de type sonore.[8]

Voici une liste rapide des fonctionnalités de Dart qui, ensemble, le rendent indispensable pour Flutter :

- Dart est AOT compilé en un code natif rapide et prévisible, qui permet à presque tout de Flutter d'être écrit en Dart. Cela rend non seulement Flutter rapide, mais pratiquement tout (y compris tous les widgets) peut être personnalisé.
- Dart peut également être compilé JIT (Just In Time) pour des cycles de développement exceptionnellement rapides et un flux de travail révolutionnaire (y compris le populaire rechargement à chaud avec état sous la seconde de Flutter).

- Dart facilite la création d'animations lisses et de transitions exécutées à 60 images par seconde. Dart peut faire l'allocation d'objets et la collecte de place sans verrous. Et comme JavaScript, Dart évite la planification préemptive et la mémoire partagée (et donc les verrous). Les applications Flutter étant compilées en code natif, elles ne nécessitent pas de pont lent entre les domaines (par exemple, JavaScript vers natif). Ils démarrent aussi beaucoup plus vite.
- Dart permet à Flutter d'éviter le recours à un langage de présentation déclaratif distinct, tel que JSX ou XML, ou à des générateurs d'interface visuelle distincts, car la présentation déclarative et programmatique de Dart est facile à lire et à visualiser. Et avec toute la mise en page dans une langue et à un endroit, il est facile pour Flutter de fournir des outils avancés qui facilitent la mise en page.
- Les développeurs ont découvert que Dart est particulièrement facile à apprendre car il comporte des fonctionnalités bien connues des utilisateurs de langages statiques et dynamiques.

Toutes ces fonctionnalités ne sont pas propres à Dart, mais leur combinaison constitue un atout qui confère à Dart une puissance unique pour la mise en œuvre de Flutter. Tant pis, il est difficile d'imaginer que Flutter soit aussi puissant qu'il est sans Dart.[11]

2.3 Firebase

2.4 Google Maps

2.5 Google Books

Bibliographie

[1] BookaBikia | About us. https ://bookabikia.com/about.

[2] BookMooch Reviews | Online Bookstores Companies | Best Company. https ://bestcompany.com/bookstores/company/bookmooch.

[3] Bookup - hookups for books ! Location-based print book swapping app. https ://www.bookupapp.com/.

[4] Mobile Operating System Market Share Worldwide. http ://gs.statcounter.com/os-market-share/mobile/worldwide.

[5] Technical Overview. https ://flutter.dev/docs/resources/technical-overview.

[6] What is Cross Platform ? - Definition from Techopedia. https ://www.techopedia.com/definition/17056/cross-platform.

[7] Book swapping. *Wikipedia*, January 2019. Page Version ID : 879634702.

[8] Dart (programming language). *Wikipedia*, April 2019. Page Version ID : 894124175.

[9] Mobile app development. *Wikipedia*, January 2019. Page Version ID : 878564579.

[10] Wm Leler. What's Revolutionary about Flutter. https ://hackernoon.com/whats-revolutionary-about-flutter-946915b09514, August 2017.

[11] Wm Leler. Why Flutter Uses Dart. https ://hackernoon.com/why-flutter-uses-dart-dd635a054ebf, February 2018.