

Table des matières

1	Généralités	6
1.1	Introduction au sujet	6
1.2	Solutions déjà existants	7
1.2.1	Non technologiques	7
1.2.2	Technologiques	9
1.3	Solution proposé	12
1.4	Plan du rapport	13
2	Les technologies utilisées	14
2.1	Un bref historique du développement d'applications mobiles .	14
2.1.1	Les kits de développement de plate-forme (Platform SDKs)	15
2.1.2	Vues Web (WebViews)	15
2.1.3	Vues réactives (Reactive Views)	16
2.2	Flutter	17
2.2.1	Qu'est-ce que Flutter ?	17
2.2.2	Principes de base	18
2.2.3	Pourquoi utiliser Flutter ?	23
2.3	Firebase	25
2.3.1	Introduction	25
2.3.2	Bref historique	26
2.3.3	Les services de Firebase	26
2.3.4	Pourquoi utiliser Firebase ?	29
2.4	Google APIs	32
2.4.1	Google Maps	32
2.4.2	Google Books	33
3	Conception et implémentation de l'application	34
3.1	Conception	34
3.1.1	Introduction	34
3.1.2	Diagrammes de base	35

TABLE DES MATIÈRES	2
--------------------	---

3.1.3	Maquette fonctionnelle (Wireframe)	36
3.2	Implémentation	36

Table des figures

1.1	Un "échange de livres de rue" à Washington Heights, New York[13]	8
1.2	Logo de BookMooch	9
1.3	Logo de Bookup	10
1.4	Le mecanisme de suggestion et decouverte offert par Book Up ou l'utilisateur parcourt une pile de cartes des livres près de son zone géographique on choisissent "passer" ou "vouloir"	10
1.5	Logo de Bookabikia	11
1.6	Page d'accueil de Bookabikia	11
1.7	Logo de Bindex	12
2.1	Architecture du développement mobile a l'aide des SDK[19]	15
2.2	Architecture du développement mobile a l'aide des WebViews[19]	16
2.3	Architecture du développement mobile a l'aide des Reactive Views[19]	16
2.4	Logo de React Native	17
2.5	Logo de Flutter	17
2.6	Ceci est une application de démonstration nommée Shrine[9]	17
2.7	Architecture du développement mobile a l'aide de Flutter[19]	18
2.8	Exemple d'une hierarchy des widgets[9]	20
2.9	L'architecture de flutter[9]	21
2.10	Modele d'une Widget Stateful[9]	22
2.11	Logo de Dart	24
2.12	Logo de Firebase	25
2.13	Logo de Parse	29
2.14	Échantillonne des entreprises et applications que utilisent Firebase	30
2.15	différents modèles de tarification de Firebase	31
2.16	Exemple de l'intégration de cartes Google Maps pour Android	32
3.1	l'enchaînement des actions pour la recherche d'un livre particulier	35
3.2	l'exploration des livres/utilisateurs à proximité	35
3.3	l'architecture de navigation	36

Glossaire et acronymes

SDK Software Development Kit. 15

AOT Ahead Of Time. 18, 24

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. 18

JIT Just In Time. 24

JSX JavaScript XML. 25

XML Extensible Markup Language. 25

API une API (Application Programming Interface) est un ensemble de règles et spécifications qu'un programme peut suivre pour accéder et faire usage des services et des ressources fournis par un autre programme particulier qui implémente cette API. 26

BETA Une pré-version du logiciel est donnée à un grand groupe d'utilisateurs à essayer dans des conditions réelles. Les versions bêta ont subi des tests alpha en interne et sont généralement assez proches du point de vue de la présentation et de la fonctionnalité du produit final; cependant, des modifications de conception sont souvent apportées.[1]. 26

ML Machine Learning. 26

Webhook Un Webhook en développement Web est une méthode permettant d'augmenter ou de modifier le comportement d'une page Web, ou d'une application Web, avec des rappels personnalisés. Ces rappels peuvent être gérés, modifiés et gérés par des utilisateurs tiers et des développeurs qui ne sont pas nécessairement affiliés au site Web ou à l'application d'origine. Le terme "webhook" a été inventé par Jeff Lindsay en 2007 à partir du terme de programmation pour ordinateur[17]. 27

freemium un modèle commercial, en particulier sur Internet, dans lequel les services de base sont fournis gratuitement, tandis que les fonctionnalités les plus avancées doivent être payées. 31

REST est un style d'architecture logicielle qui fournit une approche pratique et cohérente pour demander et modifier des données. Le terme REST est l'abréviation de "Representational State Transfer". Dans le contexte des API Google, il s'agit d'utiliser des verbes HTTP pour récupérer et modifier les représentations des données stockées par Google. 32

HTTP HyperText Transfer Protocol. 32

JSON JavaScript Object Notation. 32

UML Unified Modeling Language. 34

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 partagent les mêmes intérêts et la même zone géographique, se pose le problème de stockage de ces livres et du fait qu'ils peuvent être lus 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 lisant de nouveaux 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 différents 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[13].

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[13]

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 de livres 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.

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.[3]



FIGURE 1.2 – Logo de BookMooch

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.[4]



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.

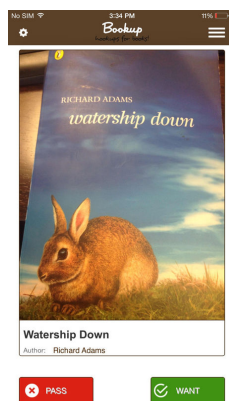


FIGURE 1.4 – Le mécanisme de suggestion et découverte offert par Book Up où l'utilisateur parcourt une pile de cartes des livres près de son zone géographique on choisissent "passer" ou "vouloir"

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 Bookabikia, 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.[2]



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.

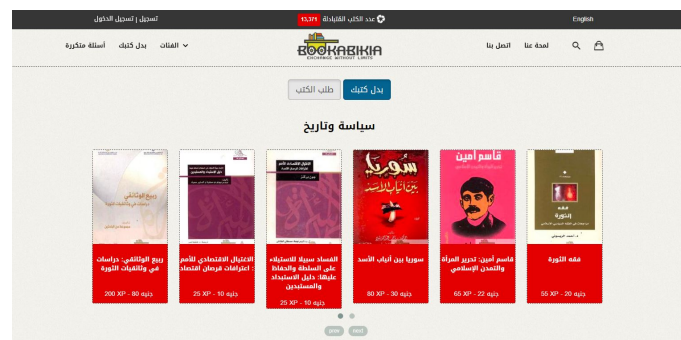


FIGURE 1.6 – Page d'accueil de Bookabikia

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 (Android et iOS) [8]), 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) à 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 et aussi le développement, 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 [16]. Toutefois, au cours de la dernière décennie [19], 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[19].

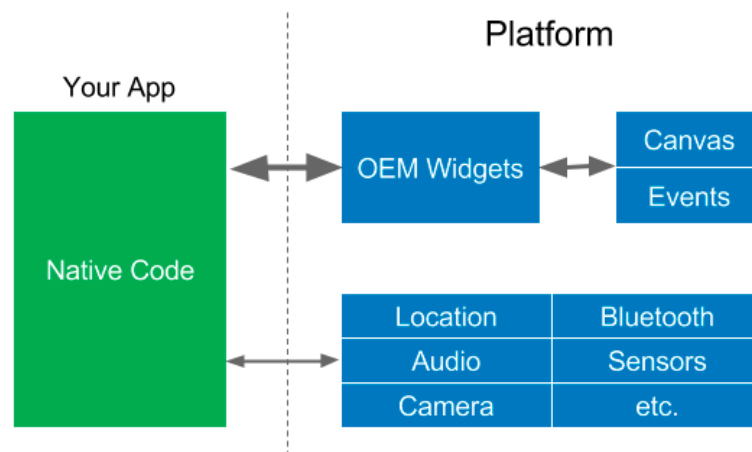


FIGURE 2.1 – Architecture du développement mobile à l'aide des SDK[19]

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

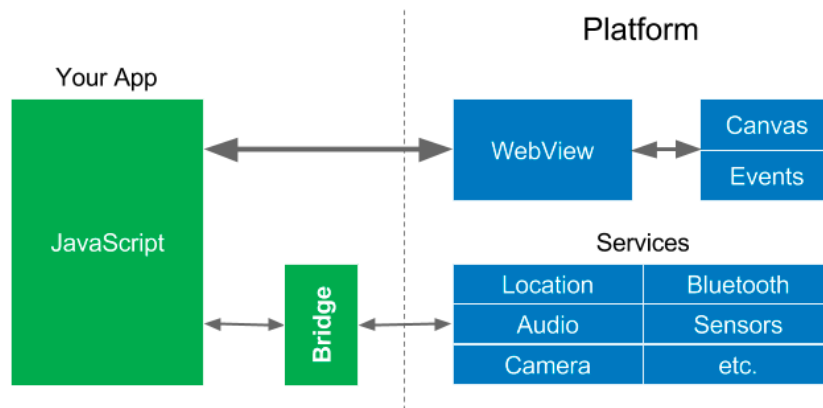


FIGURE 2.2 – Architecture du développement mobile a l'aide des WebViews[19]

généralement pas appelés très souvent, cela n'a pas causé trop de problèmes de performances[19].

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.

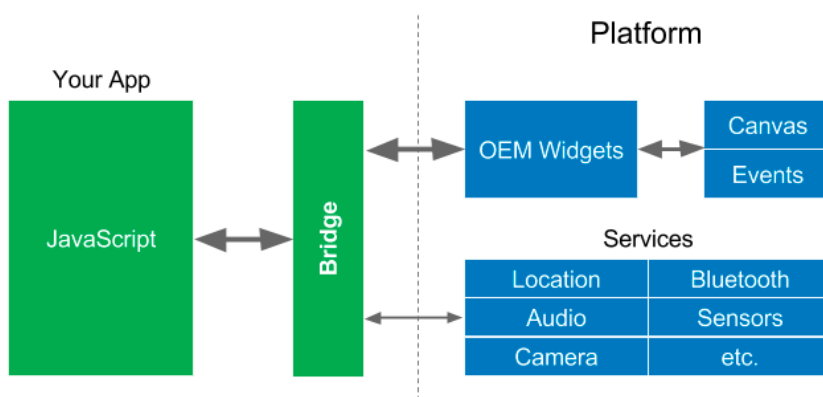


FIGURE 2.3 – Architecture du développement mobile a l'aide des Reactive Views[19]

React Native est très populaire (et mérite de l'être), mais comme le domaine JavaScript accède aux widgets de la plateforme 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.

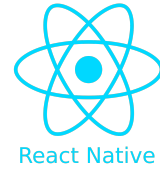


FIGURE 2.4
— Logo de
React Native

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[9].

L'objectif est de permettre aux développeurs de proposer des applications hautes performances qui se sentent naturelles sur différentes plates-formes.



FIGURE 2.5
— Logo de
Flutter

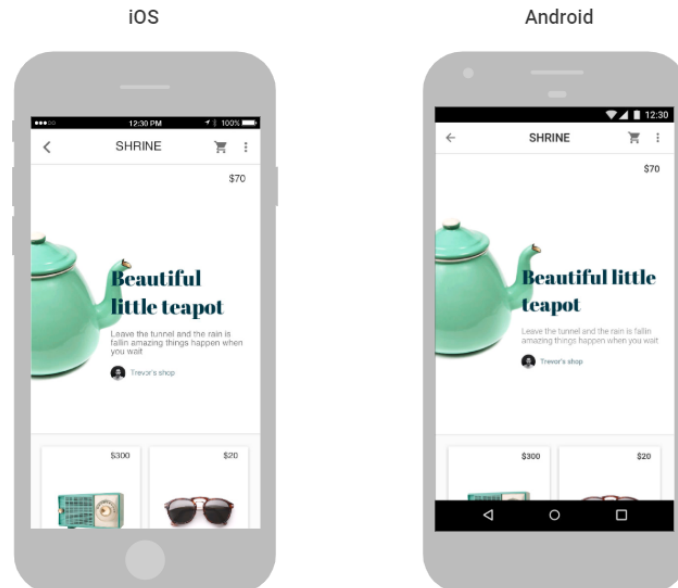


FIGURE 2.6 — Ceci est une application de démonstration nommée Shrine[9]

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.

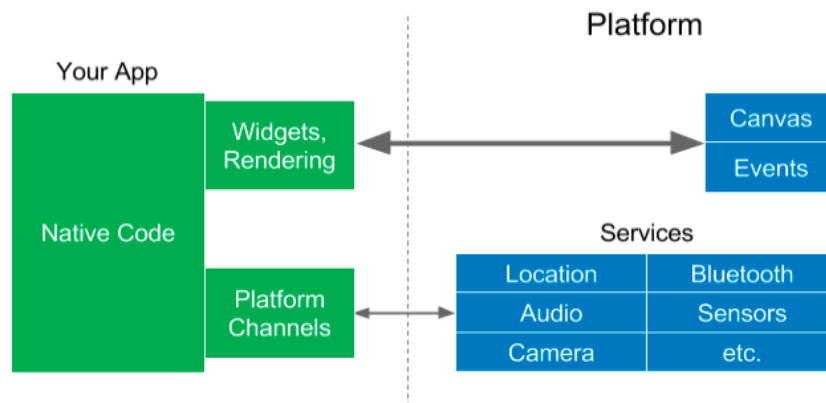


FIGURE 2.7 – Architecture du développement mobile a l'aide de Flutter[19]

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.

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,

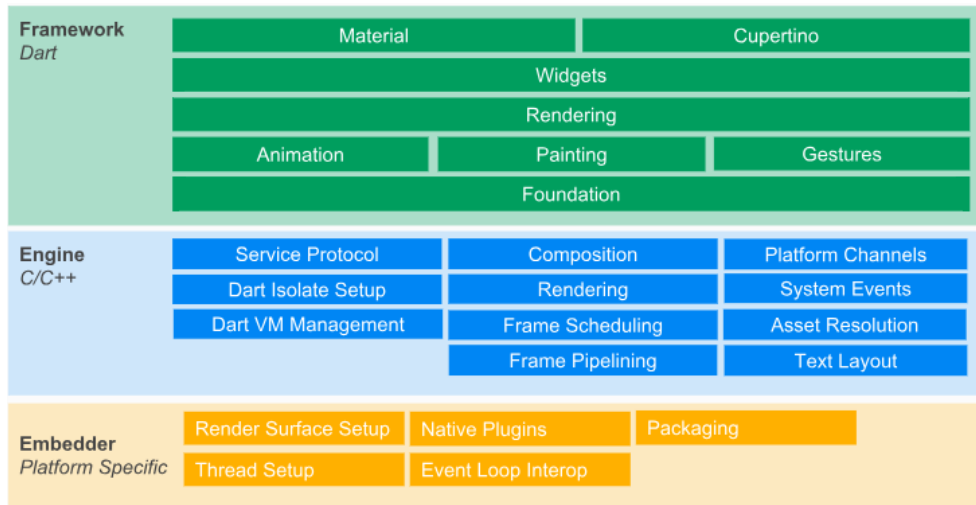


FIGURE 2.9 – L’architecture de flutter[9]

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 applications 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.

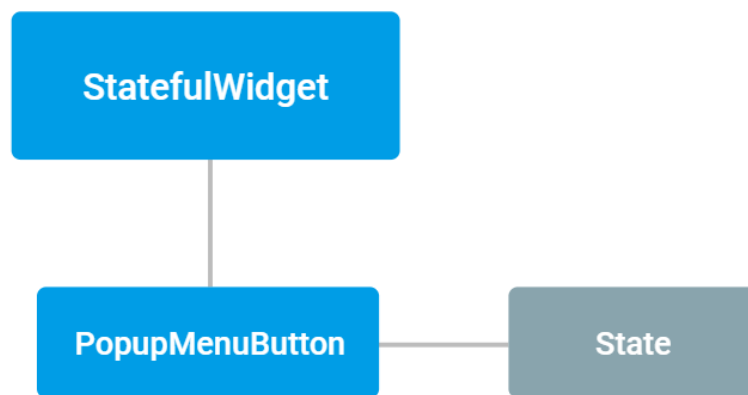


FIGURE 2.10 – Modèle d'une Widget Stateful[9]

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[11]

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'y a 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'y a 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.[14]

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.[20]

2.3 Firebase

2.3.1 Introduction

Firebase est une plate-forme de développement d’applications mobiles et Web qui fournit aux développeurs une multitude d’outils et de services pour les aider à développer des applications de haute qualité, à développer leur base d’utilisateurs et à générer davantage de bénéfices.[18]



FIGURE 2.12 – Logo de Firebase

2.3.2 Bref historique

En 2011, avant que Firebase fût Firebase, c'était une startup appelée Envelope. En tant que Envelope, il fournissait aux développeurs une API qui permettait l'intégration de la fonctionnalité de discussion en ligne dans leur site Web.

Qu'est ce que c'est ? Il est intéressant de noter que c'est utilisé pour envoyer des messages. Les développeurs utilisaient Envelope pour synchroniser les données des applications en tant qu'état de jeu en temps réel sur l'ensemble de leurs utilisateurs.

Ceci a conduit les fondateurs d'Envelope, James Tamplin et Andrew Lee, à séparer le système de chat et l'architecture en temps réel. En avril 2012, Firebase a été créée en tant que société distincte fournissant un système de gestion en temps réel.

Après son acquisition en 2014, Firebase a rapidement évolué pour devenir le monstre multifonctionnel d'une plate-forme mobile et Web qu'il est aujourd'hui.

2.3.3 Les services de Firebase

les services de Firebase peuvent être divisé en trois groupes :

1 - Construction des meilleures applications :



Firestore Cloud Stockez et synchronisez des données entre utilisateurs et appareils - à l'échelle mondiale - à l'aide d'une base de données NoSQL hébergée dans le cloud. Cloud Firestore vous offre une synchronisation en direct et une assistance hors ligne, ainsi que des requêtes de données efficaces. Son intégration avec d'autres produits Firebase vous permet de créer de véritables applications sans serveur.



Kit ML ^{BETA} Intégrez de puissantes fonctionnalités d'apprentissage automatique (Machine Learning) à votre application mobile, que vous soyez débutant ou expérimenté dans ML. Commencez facilement en utilisant les API prêtes à l'emploi pour les cas d'utilisation courants des appareils mobiles ou importez vos propres modèles personnalisés qui peuvent être hébergés et fournis à vos applications par Firebase. Les API du kit ML peuvent s'exécuter sur le périphérique ou dans le cloud, en fonction des fonctionnalités. Certaines vous offrent les deux choix.



Fonctions Cloud Étendez votre application avec du code d'arrière-plan personnalisé sans avoir à gérer et à faire évoluer vos propres serveurs. Les fonctions peuvent être déclenchées par des événements, émis par des produits Firebase, des services Google Cloud ou des tiers, à l'aide de Webhooks.



Authentification Gérez vos utilisateurs de manière simple et sécurisée. Firebase Auth propose plusieurs méthodes d'authentification, notamment un courrier électronique et un mot de passe, des fournisseurs tiers tels que Google ou Facebook et l'utilisation directe de votre système de compte existant. Construisez votre propre interface ou profitez de l'interface utilisateur open source entièrement personnalisable.



Hébergement Simplifiez votre hébergement Web avec des outils spécialement conçus pour les applications Web modernes. Lorsque vous téléchargez vos ressources Web, ils seront automatiquement transféré vers le CDN mondial de la plateforme Firebase et ils aient attribué un certificat SSL gratuit afin que vos utilisateurs bénéficient d'une expérience sécurisée, fiable et à faible latence, où qu'ils se trouvent.



Stockage en ligne Stockez et partagez des contenus générés par les utilisateurs, tels que des images, du son et de la vidéo, avec un stockage d'objets puissant, simple et économique, conçu à l'échelle de Google. Les SDK Firebase pour le stockage en nuage ajoutent la sécurité Google aux téléchargements de fichiers et aux téléchargements pour vos applications Firebase, quelle que soit la qualité du réseau.



Base de données en temps réel Stockez et synchronisez les données entre les utilisateurs et les périphériques en temps réel à l'aide d'une base de données NoSQL hébergée dans le cloud. Les données mises à jour se synchronisent sur les appareils connectés en quelques millisecondes. Les données restent disponibles si votre application est déconnectée, offrant ainsi une expérience utilisateur exceptionnelle, quelle que soit la connectivité réseau.

2 - Amélioration de la qualité de l'application :



Crashlytics Réduisez votre temps de dépannage en transformant une avalanche de collisions en une liste de problèmes gérables. Obtenez des informations claires et exploitables sur les problèmes à résoudre en premier en observant l'impact de l'utilisateur sur le tableau de bord de Crashlytics. Les alertes en temps réel vous aideront à rester au top de votre stabilité, même lors de vos déplacements. Crashlytics est le principal journaliste de crash pour Firebase.



Suivi de la performance Diagnostiquez les problèmes de performances des applications survenant sur les appareils de vos utilisateurs. Utilisez des traces pour surveiller les performances de parties spécifiques de votre application et consultez une vue résumée de la console Firebase. Surveillez les heures de démarrage de votre application et surveillez les requêtes HTTP sans écrire de code.



Laboratoire de test Exécutez des tests automatiques et personnalisés pour votre application sur des périphériques virtuels et physiques hébergés par Google. Utilisez le laboratoire de test Firebase tout au long de votre cycle de développement pour détecter les bogues et les incohérences afin de pouvoir offrir une expérience inoubliable sur une grande variété de périphériques.

3 - Développement d'entreprise :



Messagerie In-App Engagez et entretenez vos utilisateurs actifs avec des messages ciblés et contextuels qui les encouragent à effectuer des actions significatives au sein de votre application. Vous avez le pouvoir de déclencher des messages en fonction du comportement et des intérêts de l'utilisateur. Vous pouvez également personnaliser la conception des messages intégrés à votre marque. La messagerie in-app prend en charge un grand nombre de cas d'utilisation et de formats.



Google Analytics Analysez les attributions et les comportements des utilisateurs dans un seul tableau de bord pour prendre des décisions éclairées concernant votre feuille de route. Obtenez des informations en temps réel à partir des rapports ou exportez vos données d'événements brutes vers Google BigQuery pour une analyse personnalisée.





Prédictions Exploitez toute la puissance de l'apprentissage automatique de Google pour avoir une idée des segments d'utilisateurs susceptibles d'abandonner ou de dépenser (ou de terminer un autre événement de conversion). Utilisez ces segments prédictifs intelligents pour le ciblage dans d'autres produits tels que la configuration à distance, la messagerie en nuage et la messagerie intégrée.





Test A / B ^{BETA} Améliorez votre application en exécutant des expériences sur les produits et le marketing, sans vous soucier de la configuration de l'infrastructure pour l'exécution de tests A / B. Personnalisez les expériences en fonction de vos objectifs. Testez diverses mises à jour de votre application, telles que la copie de messages ou

de nouvelles fonctionnalités. Ensuite, seules les modifications de déploiement permettant de déplacer l'aiguille sur vos indicateurs clés ont été prouvées.

 **Cloud Messaging** Send messages and notifications to users across platforms—Android, iOS, and the web—for free. Messages can be sent to single devices, groups of devices, or specific topics or user segments. Firebase Cloud Messaging (FCM) scales to even the largest apps, delivering hundreds of billions of messages per day.

 **Configuration à distance** Personnalisez le rendu de votre application pour chaque utilisateur. Modifiez l'apparence, déployez les fonctionnalités progressivement, exécutez des tests A / B, fournissez un contenu personnalisé à certains utilisateurs ou effectuez d'autres mises à jour sans déployer une nouvelle version, le tout à partir de la console Firebase. Surveillez l'impact de vos modifications et apportez des modifications en quelques minutes.

 **Liens dynamiques** Utilisez les liens dynamiques pour offrir une expérience utilisateur personnalisée pour iOS, Android et le Web. Vous pouvez les utiliser pour alimenter le Web mobile afin de générer des conversions d'applications natives, le partage d'utilisateur à utilisateur, des campagnes sociales et marketing, etc. Dynamic Links vous fournit les attributions dont vous avez besoin pour mieux comprendre votre croissance mobile.

 **Indexation d'applications** Réengagez les utilisateurs avec leurs applications installées avec cette intégration de Google Search. Si les utilisateurs ont votre application et recherchent du contenu associé, ils peuvent la lancer directement à partir des résultats. Si les utilisateurs ne disposent pas encore de votre application, une carte d'installation s'affiche lorsqu'ils recherchent des applications similaires.

2.3.4 Pourquoi utiliser Firebase ?

Bien que la grande majorité de la communauté des développeurs soit d'accord sur le fait que Firebase est le meilleur BaaS (Backend as a Service) disponible à ce jour, il est toujours vrai qu'il existe d'autres solutions. C'est en plus de la solution traditionnelle de construction et de gestion de votre propre infrastructure backend.

L'une des alternatives les plus remarquables à Firebase est Parse, la plate-forme acquise par Facebook, qui peut être plus abordable que la plate-forme de Google sur le long terme, mais qui offre moins de services, peut être



FIGURE 2.13 —
Logo de Parse

assez compliquée à mettre en place et manque de support structuré. D'autres plates-formes telles que Back4app, Kinvey, Backendless, Kuzzle, Hoodie et autres offrent des services similaires avec quelques différences.

Cependant, comparé à Firebase, ce dernier s'est révélé être la solution la plus appropriée pour notre projet, pour les points suivants.

1. Créer des applications rapidement sans gestion d'infrastructure

L'un des aspects les plus fondamentaux et les plus attrayants de l'utilisation de Firebase est qu'elle permet aux créateurs de se concentrer entièrement sur l'application en cours de développement, outre le fait d'avoir une attention partagée entre le développement de l'application et la création et la gestion de l'infrastructure backend, qui de nos jours, est un projet à lui même, nécessitant une équipe dédiée et une communication solide avec l'équipe que s'occupe de développement de l'application.

Avec Firebase, vous êtes en mesure de faire fonctionner votre application rapidement grâce aux mécanismes de configuration simples fournis par la plateforme.

2. Backend par Google, approuvé par les meilleures applications

Les meilleures applications du marché font confiance aux produits Google. Les produits Google ne manquent généralement pas d'excellents résultats. Dans le cas de Firebase, il est utilisé par 793 entreprises[5], notamment "The New York Times", "Alibaba", "Twitch" et de nombreuses autres.

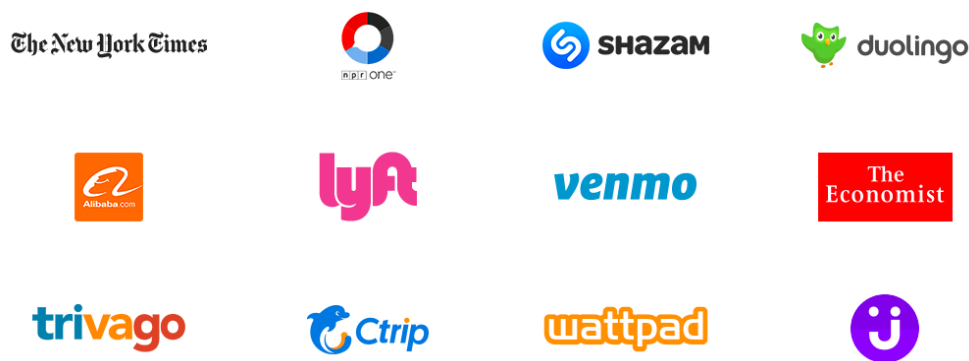


FIGURE 2.14 – Échantillonne des entreprises et applications que utilisent Firebase

3. Facile à intégrer avec Flutter

Flutter a été créé dans le but de le faire fonctionner sans problème pour Firebase. L'ensemble du processus d'intégration peut prendre jusqu'à 10 minutes, sans qu'il soit nécessaire de fournir beaucoup de données et le processus est totalement simple.[10]

4. Tarification flexible

Firebase offre aux développeurs un modèle de tarification freemium flexible et, dans une certaine mesure, généreux, adoptant l'approche consistant à payer à l'échelle de votre application, en commençant par une offre gratuite dans laquelle vous obtenez une quantité limitée de données à stocker et des requêtes de lecture limitées effectuées par pour les applications en croissance, les développeurs peuvent payer un prix fixe pour une utilisation étendue des services de la plate-forme, ainsi que davantage de stockage de données et de requêtes, pour enfin atteindre l'offre illimitée dans laquelle vous êtes facturé pour ce que vous avez utilisé.[6]

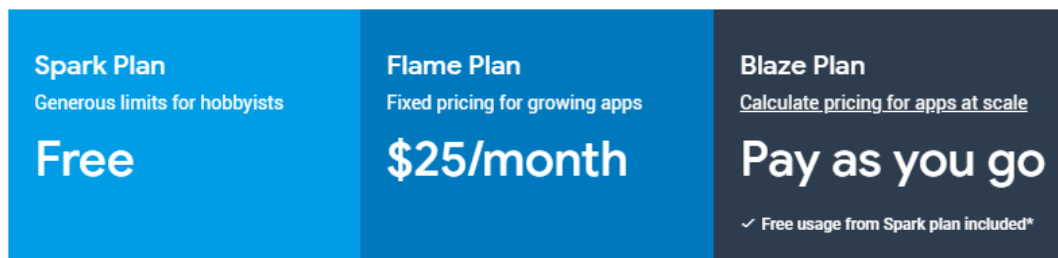


FIGURE 2.15 – différents modèles de tarification de Firebase

2.4 Google APIs

Les API Google sont un ensemble d'interfaces de programmation d'applications (API) développées par Google qui permettent la communication avec Google Services et leur intégration à d'autres services. Parmi ceux-ci figurent Search, Gmail, Translate ou Google Maps. Les applications tierces peuvent utiliser ces API pour exploiter ou étendre les fonctionnalités des services existants.

Les API fournissent des fonctionnalités telles que l'analyse, l'apprentissage automatique en tant que service (l'API de prédiction) ou l'accès aux données utilisateur (lorsque l'autorisation de lire les données est donnée). Un autre exemple important est une carte Google intégrée sur un site Web, qui peut être réalisée à l'aide de l'API Static Maps, API Places ou de Google Earth.[15]

Les deux API (Books, Maps) s'appellent RESTful car elles sont basées sur l'architecture REST qu'elle est basée sur le protocole HTTP permettant de récupérer les données au format JSON.

2.4.1 Google Maps

Google Maps est un service de cartographie Web développé par Google. Il propose des images satellite, des photographies aériennes, des cartes routières, des vues panoramiques à 360 ° sur les rues (Street View), les conditions de circulation en temps réel, ainsi que la planification d'itinéraire pour les déplacements à pied, en voiture, à vélo et en avion (en version bêta), ou les transports en commun.

Avec le kit de développement (SDK) de cartes pour Android, vous pouvez ajouter des cartes basées sur des données Google Maps à votre application. L'API gère automatiquement l'accès aux serveurs Google Maps, le téléchargement de données, l'affichage des cartes et la réponse aux mouvements de la carte. Vous pouvez également utiliser des appels d'API pour ajouter des marqueurs, des polygones et des superpositions à une carte de base, ainsi que pour modifier la vue de l'utilisateur sur une zone de carte particulière. Ces objets fournissent des informations supplémentaires sur les emplacements de carte et permettent une interaction de l'utilisateur avec la carte. L'API vous

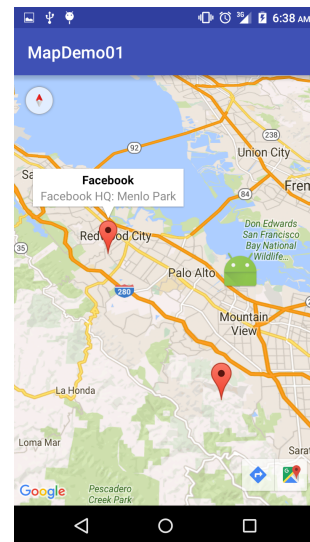


FIGURE 2.16 – Exemple de l'intégration de cartes Google Maps pour Android

permet d'ajouter ces graphiques à une carte :

- Icônes ancrées à des positions spécifiques sur la carte (marqueurs).
- Ensembles de segments de ligne (polylignes).
- Segments fermés (polygones).
- Graphiques bitmap ancrés à des positions spécifiques sur la carte (superpositions au sol).
- Ensembles d'images affichés en haut des mosaïques de la carte de base (superpositions de mosaïque).

2.4.2 Google Books

Google Books (anciennement Google Books Search et Google Print et son nom de code Project Ocean) est un service de Google Inc. qui effectue une recherche dans le texte intégral des livres et des magazines scannés par Google et convertis en texte à l'aide de la reconnaissance optique de caractères. (OCR), et stocké dans sa base de données numérique. Les livres sont fournis par les éditeurs et les auteurs, via le programme Partenaires Google Books, ou par les partenaires bibliothèques de Google, via le projet Library. De plus, Google s'est associé à plusieurs éditeurs de magazines pour numériser leurs archives.[7]

La nouvelle API v1 de Google Books vous permet d'accéder par programme à de nombreuses opérations disponibles sur le site Web de Google Books. Vous pouvez l'utiliser pour créer des applications puissantes offrant une intégration plus poussée avec Google Books. Les principales fonctionnalités de l'API sont les suivantes :

- rechercher et parcourir la liste des livres correspondant à une requête donnée.
- afficher des informations sur un livre, notamment les métadonnées, la disponibilité et le prix, ainsi que des liens vers la page d'aperçu.
- gérer vos propres étagères.

Chapitre 3

Conception et implémentation de l'application

3.1 Conception

3.1.1 Introduction

Tout comme la construction d'une maison nécessite des plans à différents niveaux (vision extérieure, plan des différents étages, plans techniques...), la réalisation d'une application informatique ou d'un ensemble d'applications est basée sur plusieurs diagrammes. Ces diagrammes doivent unifier la vision de l'équipe de développement et l'orienter vers la création d'une solution bien alignée sur les exigences du problème, et permettre également à l'équipe de proposer à ses clients une expérience utilisateur cohérente tout au long de l'application.

Pour atteindre l'objectif de communiquer une vision ou un système difficile à expliquer par des mots, différents langages de modélisation ont été créés, tels que SysML, BON et le langage UML le plus connu et le plus utilisé, ainsi que des méthodes anciennes et récentes, telles que le wireframing.

La pratique des conceptions logicielles n'est pas une pratique rigide dans laquelle plusieurs étapes clés sont toujours présentes dans un ordre particulier, par exemple : Le langage UML ne préconise aucune démarche, ce n'est donc pas une méthode. Chacun est libre d'utiliser les types de diagramme qu'il souhaite, dans l'ordre qu'il veut. Il suffit que les diagrammes réalisés soient cohérents entre eux, avant de passer à la réalisation du logiciel.

3.1.2 Diagrammes de base

Diagrammes de séquence

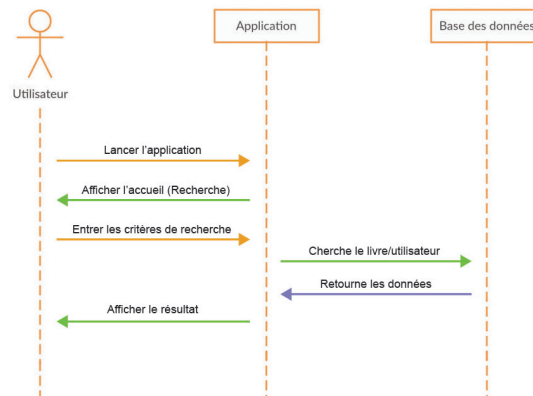


FIGURE 3.1 – l'enchaînement des actions pour la recherche d'un livre particulier

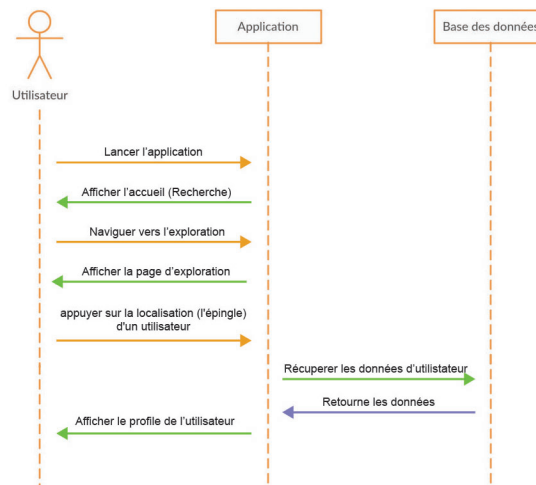


FIGURE 3.2 – l'exploration des livres/utilisateurs à proximité

Diagramme de navigation

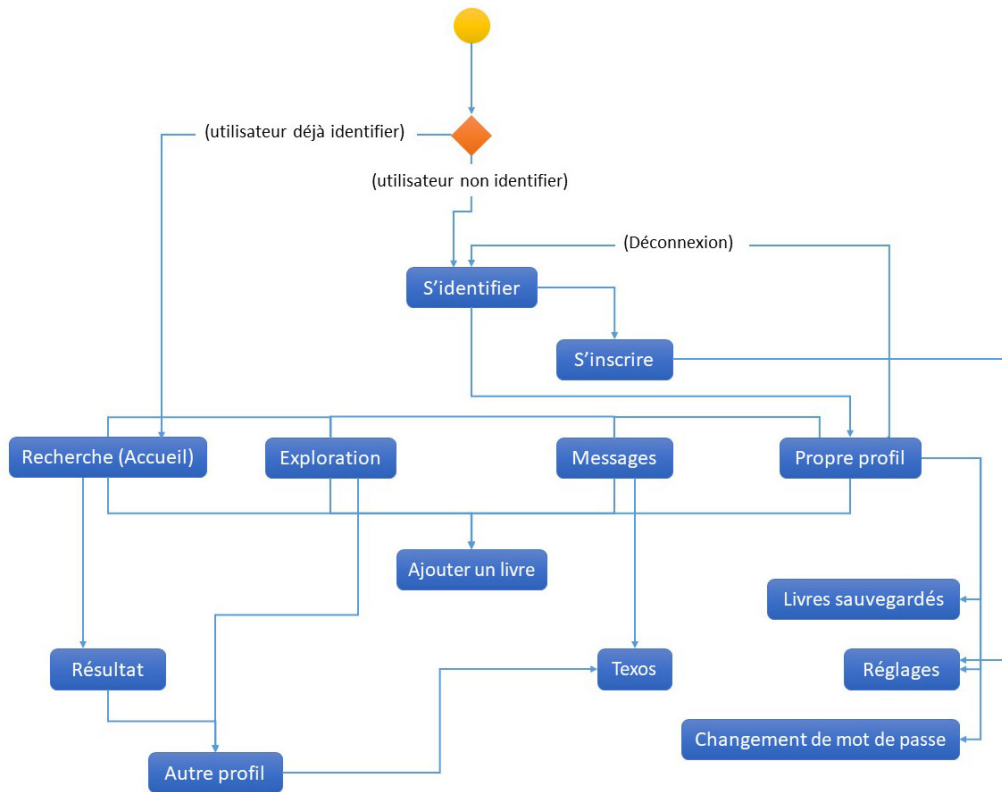


FIGURE 3.3 – l'architecture de navigation

3.1.3 Maquette fonctionnelle (Wireframe)

Le wireframe ou maquette fonctionnelle est un schéma utilisé lors de la conception d'une interface pour définir les zones et composants qu'elle doit contenir. À partir d'un wireframe peut être réalisée l'interface proprement dite par un graphiste. La démarche de recourir à des wireframes s'inscrit dans une recherche d'ergonomie. Elle est surtout utilisée dans le cadre du développement logiciel et des sites et applications Web. Le wireframe consiste concrètement en un croquis, un collage papier ou un schéma numérique[12].

3.2 Implémentation

Bibliographie

- [1] Beta version Definition from PC Magazine Encyclopedia.
<https://www.pcmag.com/encyclopedia/term/38567/beta-version>.
- [2] BookaBikia | About us. <https://bookabikia.com/about>.
- [3] BookMooch Reviews | Online Bookstores Companies | Best Company.
<https://bestcompany.com/bookstores/company/bookmooch>.
- [4] Bookup - hookups for books! Location-based print book swapping app.
<https://www.bookupapp.com/>.
- [5] Companies that use Firebase & Firebase Integrations | StackShare.
<https://stackshare.io/firebase>.
- [6] Firebase. <https://firebase.google.com/pricing>.
- [7] Google Books - Wikipedia. https://en.wikipedia.org/wiki/Google_Books.
- [8] Mobile Operating System Market Share Worldwide.
<http://gs.statcounter.com/os-market-share/mobile/worldwide>.
- [9] Technical Overview. <https://flutter.dev/docs/resources/technical-overview>.
- [10] Using Firestore as a backend to your Flutter app - YouTube.
https://www.youtube.com/watch?v=DqJ_KjFzL9I.
- [11] What is Cross Platform? - Definition from Techopedia.
<https://www.techopedia.com/definition/17056/cross-platform>.
- [12] Wireframe (design) — Wikipédia. [https://fr.wikipedia.org/wiki/Wireframe_\(design\)](https://fr.wikipedia.org/wiki/Wireframe_(design)).
- [13] Book swapping. *Wikipedia*, January 2019. Page Version ID : 879634702.
- [14] Dart (programming language). *Wikipedia*, April 2019. Page Version ID : 894124175.
- [15] Google APIs. *Wikipedia*, January 2019. Page Version ID : 877520298.
- [16] Mobile app development. *Wikipedia*, January 2019. Page Version ID : 878564579.
- [17] Webhook. *Wikipedia*, May 2019. Page Version ID : 895955714.

-
- [18] GeekyAnts. Introduction to Firebase. <https://hackernoon.com/introduction-to-firebase-218a23186cd7>, December 2017.
 - [19] Wm Leler. What's Revolutionary about Flutter. <https://hackernoon.com/whats-revolutionary-about-flutter-946915b09514>, August 2017.
 - [20] Wm Leler. Why Flutter Uses Dart. <https://hackernoon.com/why-flutter-uses-dart-dd635a054ebf>, February 2018.