

Table des matières

1	Programmation	2
1.1	Choix des technologies	2
1.2	Flutter	3
1.2.1	Les widgets	3
1.2.2	Flux des données	3
1.3	La base de données	3
1.3.1	La méthodologie REST	3
1.3.2	REST appliqué à Dart	3
1.4	Notifications	3
1.5	Sécurité	3
1.5.1	Authentification	3
1.5.2	Firebase rules	3

Implémentation

Dans ce chapitre sont présentés les aspects liés à l'implémentation informatique de l'application. Seuls les points clefs seront mis en exergue, ainsi une fois présenté un certain aspect, toutes ses apparitions disséminées dans le code ne seront pas énumérées.

1.1 Choix des technologies

Le développement d'applications pour smartphone est depuis un décennie en pleine effervescence et il existe par conséquent une multitude de frameworks, services, langages, méthodologies et paradigmes liés à leur développement.

Ces technologies, aux noms exotiques, aux logos plus brillants les uns que les autres et leurs promotions par diverses conférences, font que leurs différences relèvent plus d'une stratégie marketing, visant les informaticiens, réussie que des attributs intrinsèques de la technologie en question.

L'application étant d'une complexité modérée et ne demandant pas de ressource importantes comme tel pourrait être le cas pour une messagerie instantanée à grand échelle ou une application utilisant abondamment un domaine spécifique de connaissance comme le machine learning, le traitement d'images, les jeux vidéo, etc. Exclu de facto le choix d'une technologie basée exclusivement sur les performances ou sur le développement natif.

N'ayant jamais fait cela auparavant, il n'y a aucune préférence de ma part pour telle ou telle technologie.

Ces constats donnent lieu aux critères de sélection suivants :

- Développement cross-plateforme
- Simplicité
- Apprentissage d'un langage plutôt qu'une multitude
- Vaste documentation et ressources d'apprentissage

Le premier critère est celui qui réduit le plus la liste des possibilités. En effet, les frameworks permettant le développement d'applications pour Android et iOS ne se comptent pas en grand nombre. Il existe :

- Xamarin - Microsoft
- React Native - Facebook
- Flutter - Google
- Adobe PhoneGap - Adobe
- Ionic - MIT

Le choix parmi ces possibilités découle essentiellement de l'arbitraire. Toutefois, Ionic a été exclu car il est nécessaire de maîtriser HTML5 et par conséquent CSS mais encore Angular JS. Ce qui contredit le 3ème critère.

React native a été exclu pour des raisons similaires. I.e. l'apprentissage de divers langages.

Finalement, suite à un cours de Academind d'une durée de 40 heures portant sur les aspects les plus basiques du développement jusqu'au déploiement de l'application en passant par le routage, la gestion de requêtes http, la connexion à tout un écosystème de bases de données, l'utilisation de caméra et géolocalisation, et même sur comment changer le logo de l'application, le choix s'est porté sur Flutter.

Flutter est un framework créé par Google. Ce dernier offre pour les applications le service Firebase qui englobe :

- Cloud Firestore

- Real time database
- Functions
- Machine learning
- Cloud messaging
- ...

Firebase s'intègre, par conception, particulièrement bien et facilement à Flutter. Même s'ils sont indépendants leur utilisation conjointe forme un seul écosystème plus facile à appréhender. Ainsi pour la base de données, le choix a été la Real time database. Car cette dernière fournit une API REST.

De plus, toujours dans cet écosystème, Cloud messaging est utilisé pour l'envoi des notifications et Functions pour effectuer des actions côté serveur lorsque la base de donnée subie des modifications.

1.2 Flutter

1.2.1 Les widgets

1.2.2 Flux des données

Par constructeur

Provider & Consumer

1.3 La base de données

1.3.1 La méthodologie REST

1.3.2 REST appliqué à Dart

1.4 Notifications

1.5 Sécurité

1.5.1 Authentification

1.5.2 Firebase rules