Table des matières

1 Programmation **2**
1.1 Choix des téchnologies 2
1.2 Flutter 3
1.2.1 Les widgets 4
1.2.2 Flux des données 4
1.3 La base de données 4
1.3.1 La méthodologie REST 4
1.3.2 REST appliqué à Dart 4
1.4 Notifications 4
1.5 Sécurité 4
1.5.1 Authentification 4
1.5.2 Firebase rules 4

CHAPITRE 1

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 disséminée dans le code ne seront pas énumérées.

1.1 Choix des téchnologies

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

Ces téchnologies, aux noms éxotiques, 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 téchnologie en question.

L'application étant d'une compléxité modérée et ne demandant pas de resource importantes comme tel pourrait être le cas pour une messagerie instantanée à grand échelle ou une application utilisant abondament un domaine spécifique de connaissance comme le machine learning, le traitement d'images, les jeux vidéos, etc. Exclu de facto le choix d'une téchnologie 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 téchnologie.

Ces constats donnent lieu aux critères de selections suivants :

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

Le premier critères est celui qui réduit le plus la liste des possibilités. En effet, les frameworks permetant 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 parmis ces possibilités découle essentiellement de l'arbitraire. Toutefois, Ionic à été exclu car il est nécessaire de maitriser 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 langage.

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

Flutter est un framework crée par Google. Ce dernier offre pour les application 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 écosistème plus facile à apréhender. Ainsi pour la base de données, le choix à été la Real time database. Car cette dernière fournit une API REST.

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

1.2 Flutter

Flutter est un software developpement kit (SDK) développé par Google permettant de concevoir des applications pour plusieur plateformes. Notament, Android et IOS entre autres avec un seul code source.

Le framework a une architecture avec plusieurs éléments dont 3 sont pertienents pour ce projet :

— La plateform Dart
— La bibliothèque Fundation
— Des widgets spécifiques

Dart

Dart est le langage de programmation avec lequel certains éléments du framework on été développé mais il s'agit surtout du langage dans lequel on implemente une application en Flutter.

C'est un langage orienté objet avec un syntaxe de type C à l'image d'autres langages comme Java ou C++. Sa syntaxe est proche du Java.

Il partage certaines fonctionalités propres au langages fonctionnels. Notament, l'inférence de type ou encore certaines fonctionalités comme les map, functors . . .

En Dart, l'allocation de mémoire est gérée automatiquement par un garbage collector.

Voici un exemple minimal d'un hello world en Dart :

```
1 void main() => print('Hello, World!');
```

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