ECE PARIS
ÉCOLE D'INGÉNIEURS

RAPPORT DE PROJET

Métiers de la création numérique

Application de Running sur Flutter

Auteurs :

Thomas COSTA
Antoine BENARD
Paul ESCHWEGE
Etienne GESLIN

Enseignant :

Jean François BONNET
Emmanuel BELLANGER

Nous attestons que ce travail est original, qu'il est le fruit d'un travail commun au groupe et qu'il a été rédigé de manière autonome.

Paris, le 25/03/2021

PARTIE I. Les prémices et buts de ce projet.

Ce projet a débuté au premier semestre avec un groupe constitué de Thomas, Paul et trois autres élèves. Ils ont décidé de créer une application de running. Pour cela, ils ont développé une app fonctionnelle sur Android Studio sans utiliser de bases de données. Le but du deuxième semestre était donc de continuer le développement de l'application.

Début janvier, l'équipe a été modifiée à la suite du changement de mineure. C'est à ce moment que trois autres étudiants ont changé de mineure et qu'Antoine et Etienne ont rejoint l'équipe. A ce moment, nous avons discuter avec les professeurs afin d'avoir des pistes pour continuer de travailler et d'approfondir notre travail. Pour cela, il nous a été conseillé de tout rependre avec un nouvel outil de développement de programmation : Flutter. Cet outil développé par Google en 2015 par les développeurs de Dart. En effet, la technologie de flutter repose sur le langage informatique Dart et utilisent de nombreuses fonctionnalités avancées. De plus, Dart ne ressemble à aucun langage qu'on ait pu apprendre ou voir pendant nos quatre premières années.

Nous avons donc commencé à apprendre à manipuler ce nouvel outil afin d'obtenir un rendu plus optimal que si on était resté sur Android Studio. De plus, le second but de ce semestre est d'exploiter une base de données afin de rendre notre application utilisable. Pour cela, nous avons commencé à utiliser la BDD de Flutter. Toutefois, ce n'était pas la manière optimale pour nous. Nous avons décidé de basculer sur SQLite, une base de données en ligne qui nous permet d'avoir un accès plus rapide.

Ainsi, les objectifs de ce semestre sont l'utilisation d'un nouvel outil de programmation prometteur Flutter ainsi que la mise en place d'un nouveau type de base de données.

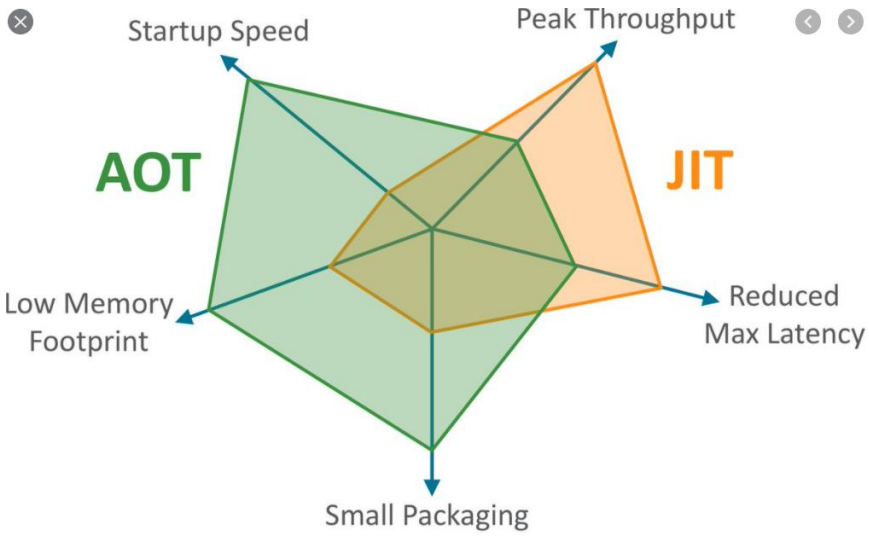
PARTIE II. La partie technique du projet.

A. L'apprentissage de Dart et l'utilisation de flutter.

L'installation de Flutter n'est pas évidente. Malgré les tutoriels existants, certains membres de l'équipe ont eu de mal à installer. Une fois l'installation terminée pour tout le monde, nous avons pu commencer à apprendre à manier cet outil ainsi qu'à apprendre le fonctionnement de Dart. Ce nouveau langage dit orienté objet a été développé par google afin de permettre la création simple et efficace d'applications mobiles ou bureaux.

En effet, compilé sous Javascript, Dart est optimale pour faire des implémentations de hautes performances. Ce nouveau langage compatible avec la majorité des navigateur Web actuels est l'un des plus performant lorsqu'il s'agit de rapidité.

De plus, l'utilisation de Flutter permet une AOT (ahead-of-time compilation), compilation anticipée. En effet, cela permet de diminuer les coûts de traitements des exceptions et permet de gagner de nombreuses performances supplémentaires par rapport à une compilation JIT (just in time) qui s'applique à la demande de l'utilisateur.



L'avantage de Flutter est sa capacité à être développer sur Android comme sur IOS en même temps. Cela permet un gain de temps énorme si nous venions à commercialiser notre application et terme de développement et de maintenance.

Enfin, le dernier plus est la fonction Hot reload qui permet de rafraîchir l'application avec un simple CTRL+S sans devoir relancer l'application comme sur Android Studio.

C'est donc pour ces raisons que nous avons choisit de coder sur Flutter en Dart lors du deuxième semestre.

B. La base de données SQLite.

La base de données utilisée dans notre projet est sqflite qui est l'équivalent en flutter de sqlite, base de données locale que l'on a vue en cours de programmation mobile.

Le principe de sqflite est assez simple : il existe une classe pour chaque table. Chaque classe regroupe plusieurs informations comme les paramètres de la table, des constructeurs pour pouvoir créer une instance de la classe et des getters pour récupérer les éléments de la classe.

```dart
class User {
  //Paramètres de la table User
  int _id;
  String _firstName;
  String _lastName;
  String _username;
  String _email;
  String _password;
  double _distanceTot;
  double _timeTot;
  double _vitesseMoy;

  //Constructeurs
  User(this._firstName,this._lastName,this._email,this._password,this._distanceTot,this._timeTot,this._vitesseMoy);
  User.withId(this._id,this._firstName,this._lastName,this._email,this._password,this._distanceTot,this._timeTot,this._vitesseMoy);

  //Getters
  int get id => _id;
  String get firstName => _firstName;
  String get lastName => _lastName;
  String get email => _email;
  String get username => _username;
  String get password => _password;
  double get distanceTot => _distanceTot;
  double get timeTot => _timeTot;
  double get vitesseMoy => _vitesseMoy;
```

Classe User qui reprend toutes les informations de l'utilisateur

Une fois les différentes classes créées, il faut ensuite faire les méthodes qui vont permettre d'envoyer les informations dans la base de données. Pour cela, il existe une classe DatabaseHelper qui va contenir toutes les méthodes liées à la gestion de la base de données.

```dart
class DatabaseHelper {
  static DatabaseHelper _databaseHelper;
  static Database _database;

  //Récupère les différentes colonnes des tables
  String courseTable = 'courseTable';
  String colIdCourse = 'idCourse';
  String colIdRunner = 'idRunner';
  String colDate = 'date';
  String colTime = 'time';
  String colDistance = 'distance';
  String colVitesse = 'vitesse';

  String userTable = 'userTable';
  String colIdUser = 'idCourse';
  String colFirstName = 'firstName';
  String colLastName = 'lastName';
  String colUsername = 'username';
  String colEmail = 'email';
  String colPassword = 'password';
  String colDistanceTot = 'distanceTot';
  String colTimeTot = 'timeTot';
  String colVitesseMoy = 'vitesseMoy';
```

Déclaration des attributs de la classe DatabaseHelper

Cette classe comporte plusieurs méthodes permettant par exemple d'insérer un nouvel utilisateur dans la base de données, de modifier ses informations, de lire ses informations et de les supprimer :

```
Future<List<Map<String,dynamic>>> getUserList() async {
  Database db = await this.database;

  var result = await db.query(userTable);
  return result;
}

Future<int> insertUser(User user) async {
  Database db = await this.database;

  var result = await db.insert(userTable, user.toMap());
  return result;
}

Future<int> updateUser(User user) async {
  Database db = await this.database;

  var result = await db.update(courseTable, user.toMap(), where: "$colIdUser", whereArgs: [user.id]);
  return result;
}

Future<int> deleteUser(int id) async {
  Database db = await this.database;

  int result = await db.rawDelete('DELETE FROM $userTable WHERE $colIdUser = $id');
  return result;
}
```

Fonctions de gestion de la table User dans DatabaseHelper

Si ce qui est expliqué avant n'est pas totalement clair, l'exemple qui va suivre devrait permettre de comprendre comment la base de données est utilisée.

Dans le fichier gérant le formulaire d'inscription, on crée un objet User vide et un objet DatabaseHelper ainsi que des variables qui vont récupérer les éléments du tableau :

```
class _RegisterScreenState extends State<RegisterScreen> {
  DatabaseHelper databaseHelper = DatabaseHelper();
  User user;

  String email = "";
  String mdp = "";
  String username = "";
```

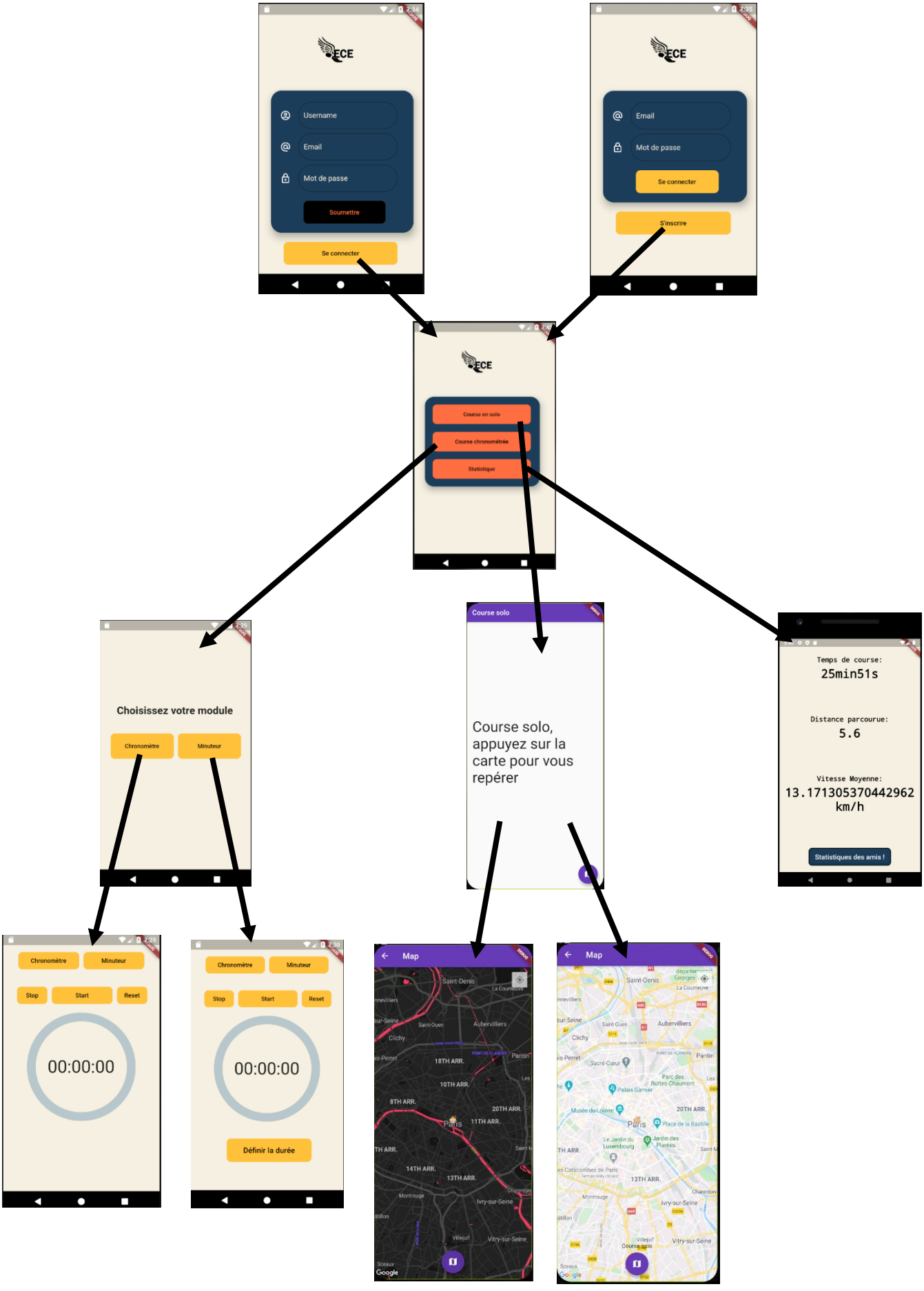
Une fois le formulaire rempli, on donne a notre objet User vide les valeurs récupérées

```
//Les valeurs 0 correspondent a des valeurs de courses comme la vitesse moyenne qui sont nulles à la création du compte
user = User(username,email,mdp,0,0,0);
```

Enfin, on insère le User dans la base de données avec la méthode insertUser de l'objet DatabaseHelper crée plus tôt :

```
databaseHelper.insertUser(user);
```

C. La navigation au sein de notre application.



PARTIE III. La conclusion du semestre.

Pour conclure, ce semestre au sein de la mineure Création numérique nous a permit de découvrir un nouvel environnement de développement Flutter qui fonctionne avec son propre langage : le Dart.

Nous avons repris l'idée du premier semestre pour l'améliorer et la développer. La séparation des taches s'est bien déroulée. En effet, le fait d'avoir un groupe de quatre personnes permet de donner des taches à tout le monde. Chacun a mis la main à la pate pour découvrir cet environnement peu intuitif malgré ses difficultés. L'ambiance générale au sein de ce groupe est restée courtoise et amicale et c'est pourquoi nous avons réussit à produire une application fonctionnelle en seulement deux mois de travail.

Nous voulons également remercier les encadrants pédagogiques de cette mineure : Jean-François Bonnet et Emmanuel Bellanger pour leurs conseils et suivi tout au long du semestre en distanciel.