

Rapport TP2 Flutter

TRINQUART Matthieu

1 Introduction

Le projet est constitué de :

- un projet flutter TP2_1 qui contient l'exercice 1 du TP.
- un projet flutter TP2_2 qui contient l'exercice 2 du TP.
- Un dossier qui contient des vidéo de démonstration des deux application
- Asset qui contient les différents shéma UML

2 TP2 exercice 1

Ce projet contient un Package data qui contient les différent model

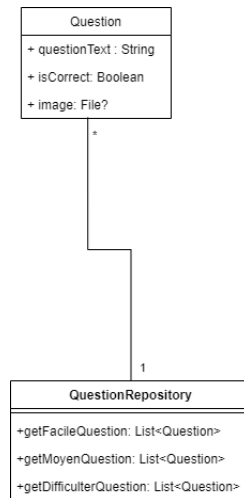


FIGURE 1 – Shéma UML du model tp1

Le package GUI contient toutes les classes qui permet d’afficher les différentes fenêtre de l’application

- Home.dart qui permet d’afficher la page d’accueil avec le choix de difficulté des questions.
- QuizzPage.dart qui affiche la page de quizz avec la question et son image associé.
- finalPage.dart qui affiche le score avec un gif qui change en fonction du résultat

Le projet contient aussi un package Provider qui contient le seul Provider du projet ”QuestionProvider.dart” ce Provider sera partagé par toutes les fenêtre de l’application.

Ce Provider contient les fonctions suivante :

- initdifficulty(QuizzDifficulty quizzDifficulty) qui permet d’indiquer quelle difficulté est choisi par l’utilisateur et instancier la list de question associer. QuizzDifficulty est un enum qui contient les valeur suivantes :facile,moyen,difficile
- Question getQuestion() qui retourne la question courante.
- bool end() qui retourne un boolean verifiant si toutes les questions ont été faite ou pas.
- void nextquestion() permet de passer à la question suivante bool Reponse(bool reponses) retourne si l’tulisateur à repondu la bonne réponse à la question ou pas.

- double `getScore()` retourne le score final de l'utilisateur.
- void `reset()` reinitialise les variable du provider afin de pouvoir recommencer un nouveau Quizz

3 TP2 exercice 2

Le `tp2_2` correspond à une application de météo. Le projet contient un package `model` qui contient toutes les classes coorespondant au Json envoyé par l'API de "openweather-map.org". Ces classes ont été généré par le site https://javiercbk.github.io/json_to_dart/ en passant au site un exemple de Json donnée par l'API.

Le projet contient aussi un package `business_logic` qui contient les diférents bloc et service.

Le projet contient qu'un seul service "Network.dart" qui contient 2 fonction

- `Future<WeatherForecastModel> getWeatherForecast(required String cityname)` qui retourne les informations météo en fonction d'une ville passé en paramètre.
- `Future<WeatherForecastModel> getWeatherForecastLocasisation(required double long , required double lat)` qui retourne les informations météo en fonctions des coordonné passé en paramètre.

Le projet contient un bloc "Weather_bloc" qui permet de charger les informations météo demandé par l'utilisateur. La classe `Weather_event` contient 2 constructeurs diférents un prenant en paramètre des coordonnées pour charger les informations en fonctions d'une longitude et d'une latitude. Un deuxième constructeur prenant en paramètre une ville afin de charger les informations météo de cette ville.

Le projet contient un package GUI contenant toutes les classes permettant d'afficher une fenêtre. Ce package contient qu'une classe "Home.dart" qui affiche la fenêtre principal de l'application.