

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 vidéos de démonstration des deux applications
- Asset qui contient les différents schémas UML

2 TP2 exercice 1

Ce projet contient un Package data qui contient les différents Model

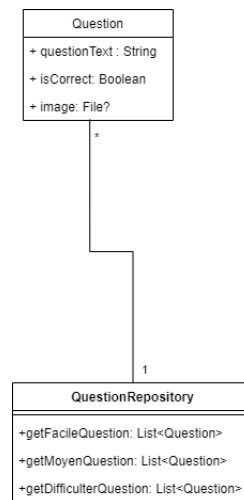


FIGURE 1 – Shéma UML du model tp1

Le package GUI contient toutes les classes qui permettent d’afficher les différentes fenêtres de l’application

- Home.dart qui permet d’afficher la page d’accueil avec le choix de difficultés des questions.
- QuizzPage.dart qui affiche la page de quizz avec la question et son image associée.
- 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 fenêtres de l’application.

Ce Provider contient les fonctions suivantes :

- initdifficulty(QuizzDifficulty quizzDifficulty) qui permet d’indiquer quelle difficulté est choisie par l’utilisateur et instancier la liste de questions associées. QuizzDifficulty est un enum qui contient les valeurs suivantes : facile, moyen, difficile
- Question getQuestion() qui retourne la question courante.
- bool end() qui retourne un boolean vérifiant si toutes les questions ont été faites ou pas.
- void nextquestion() permet de passer à la question suivante bool Reponse(bool reponses) retourne si l’utilisateur a répondu la bonne réponse à la question ou pas.

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

3 TP2 exercice 2

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

Le projet contient aussi un package `business_logic` qui contient les différents blocs et services.

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

- `Future<WeatherForecastModel> getWeatherForecast(required String cityname)` qui retourne les informations météo en fonction d'une ville passée en paramètre.
- `Future<WeatherForecastModel> getWeatherForecastLocalisation(required double long , required double lat)` qui retourne les informations météo en fonction des coordonnées passées en paramètre.

Le projet contient un bloc "Weather_bloc" qui permet de charger les informations météo demandées par l'utilisateur. La classe `Weather_event` contient 2 constructeurs différents un prenant en paramètre des coordonnées pour charger les informations en fonction 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 principale de l'application.