

# Rapport TP3 Flutter

TRINQUART Matthieu

## 1 Introduction

Le projet est constitué de :

- un projet flutter Administrateur, qui permet de générer des Data sur le Firebase et de reset la BDD Firebase.
- un projet flutter tp1 qui correspond à l'exercice du TP
- Une vidéo de démonstration qui montre les fonctionnalités de l'application
- Asset qui contient les différents schémas UML

## 2 Administrateur

Ce projet contient un package Model qui contient une classe Question qui va représenter une question et QuestionRepository qui va permettre de générer des questions avec différentes catégories (Question difficile, moyenne, facile).

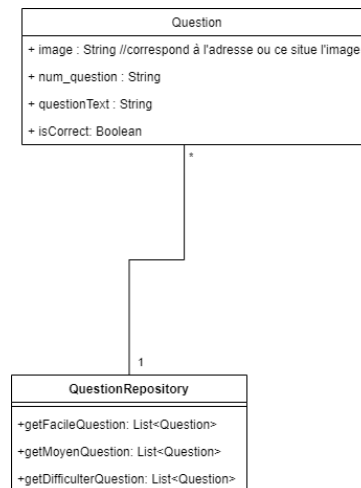


FIGURE 1 – Shéma UML du model de Administrateur

Administrateur contient un service Database.dart qui a des fonctions qui permettent de communiquer avec la Base de Donnée FireBase.

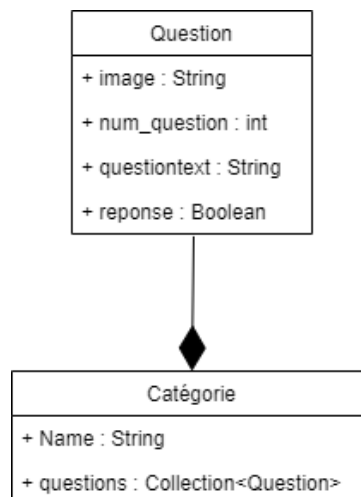


FIGURE 2 – Shéma UML de la Base de donnée FireBase

DataBase contient les fonctions suivantes :

- `addquestion(Question question,String iddoc)` qui permet d'ajouter une question à une catégorie donnée

- CreateQuestionData() qui permet de créer les différentes catégories et Question de QuestionRepository
- resetBDD() permet de réinitialiser la BDD FireBase.

### 3 tp1

Ce projet contient un Package data qui contient les différents Model

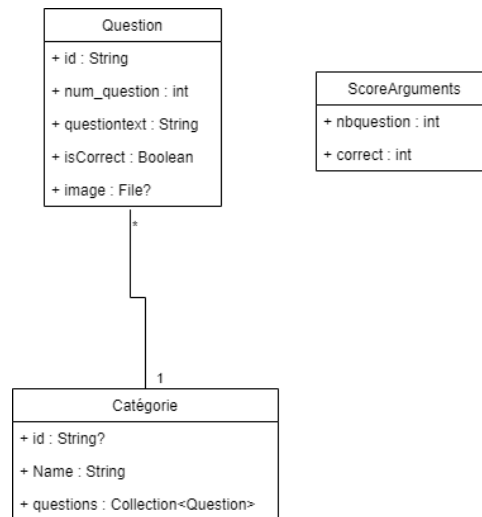


FIGURE 3 – Shéma UML du model tp1

data contient un Package service qui contient la classe QuizService qui permet de communiquer avec la Base de données FireBase. Cette fonction contient les fonctions suivantes :

- Future<List<Catégorie > > getCategory() qui retourne toutes les catégories de la base de données (sans les questions pour gagner du temps de chargement)
- Future<Catégorie> getCategorywithquestions(String idcategorie) retourne une categorie en fonction de son identifiant cette fois ci la catégorie est retournée avec les questions associées
- Future<void> addCatégorie(Catégorie categorie) permet d'ajouter une catégorie avec ces questions dans la base de données FireBase
- Future<void> EditCatégorie(Catégorie categorie) permet de modifier une catégorie de la Base de données
- EditQuestion(Question question,String iddoc) permet de modifier une question d'une catégorie
- addquestion(Question question,String iddoc) permet d'ajouter une question dans une catégorie donnée

- `GetQuestionsById(String id_categorie)` permet de retourner toutes les questions d'une catégorie.

Le projet contient un package `business_logic` qui contient les différents bloc.

- le bloc `categories` qui permet de charger les différents catégories pour afficher le choix des différentes catégories à l'utilisateur
- le bloc `EditQuizz` qui charge une catégories avec ces questions afin de les modifier
- Le bloc `question` qui permet de charger toutes les questions d'une catégories sélectionnées par l'utilisateur

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 les différents choix de catégories
- `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
- `CreateQuizzPage.dart` qui est la page qui permet de créer une catégorie
- `EditQuizzPage.dart` qui est la page qui permet de modifier une catégorie (ressemble à l'interface de `CreateQuizzPage`)