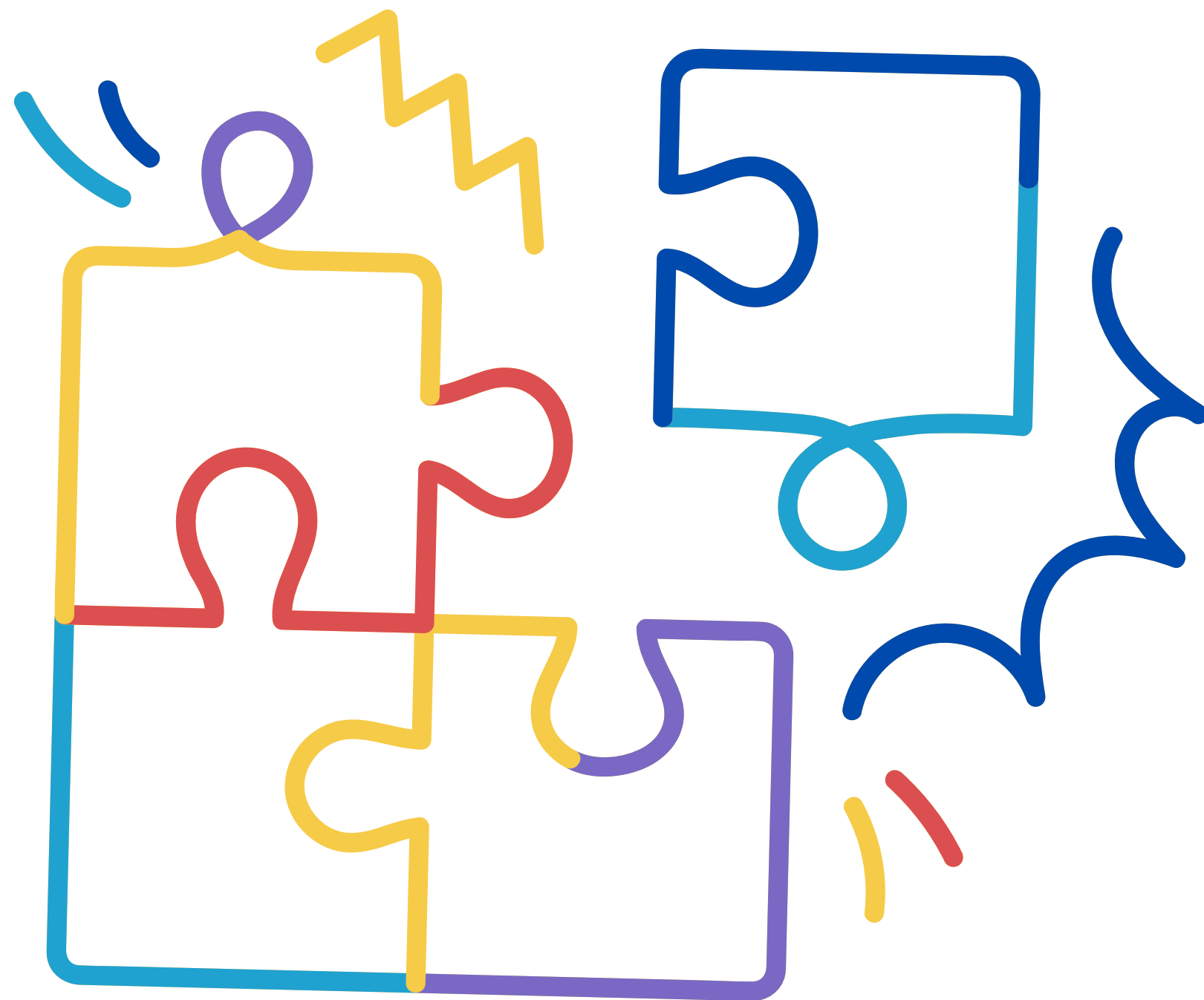
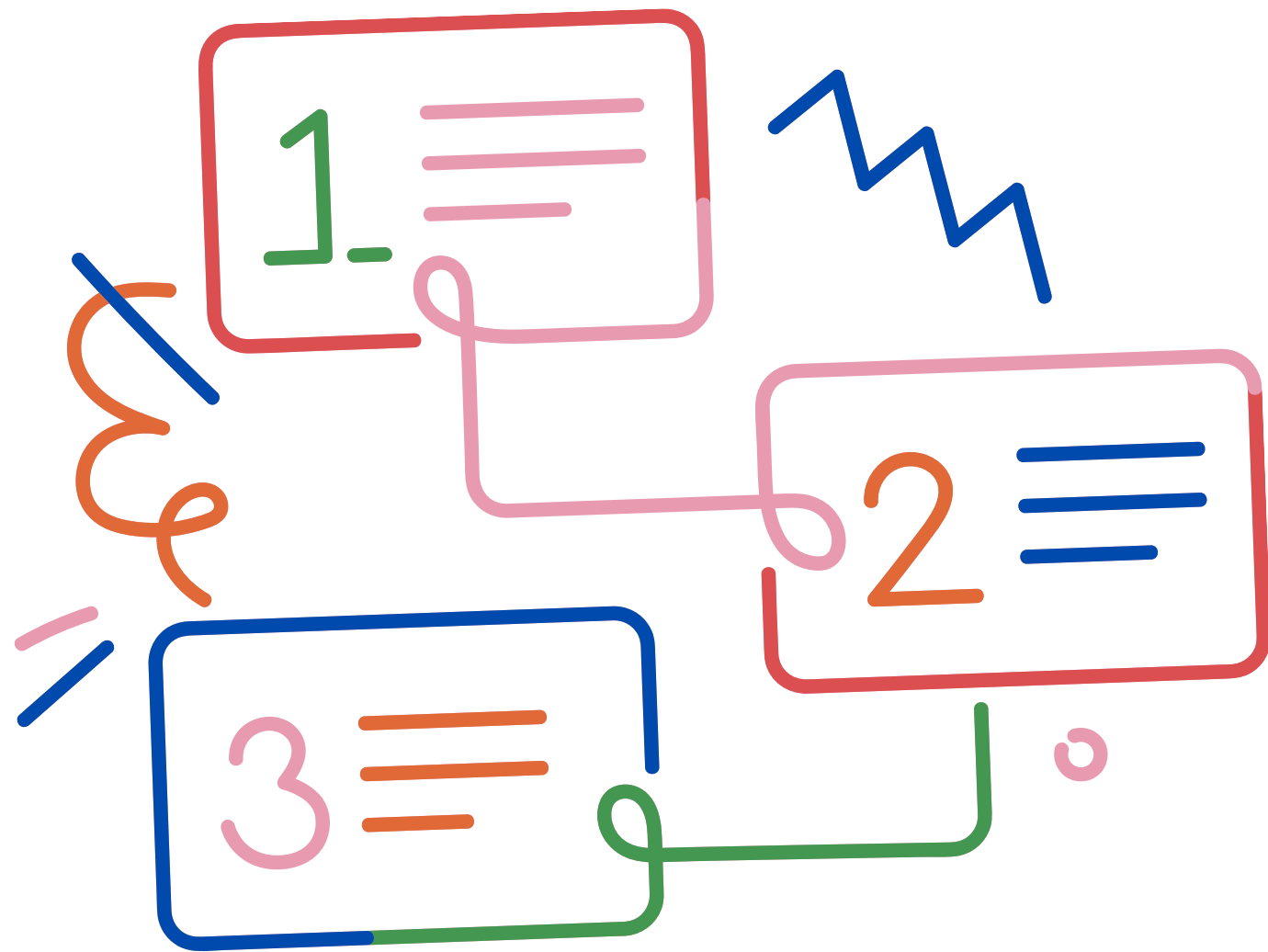


Quiz APP Flutter



Sommaire



- ⊙1. Présentation du projet
- ⊙2. Modèle des collections MongoDB.
- ⊙3. Captures d'écran de l'application.
- ⊙4. Explication de l'architecture (frontend/backend).

Présentation du projet

L'application de quiz permet aux utilisateurs de répondre à des questions et de tester leurs connaissances en langage Java. Elle est développée avec Flutter pour le frontend, utilisant l'architecture **MVVM** (Model-View-ViewModel), et une API Spring Boot pour le backend, qui se connecte à **MongoDB** pour la gestion des données.



Technologies Utilisées



Frontend

Flutter
(Architecture MVVM)



Backend

Spring Boot
(API REST)



Base de données:

MongoDB

Modèle des collections MongoDB.



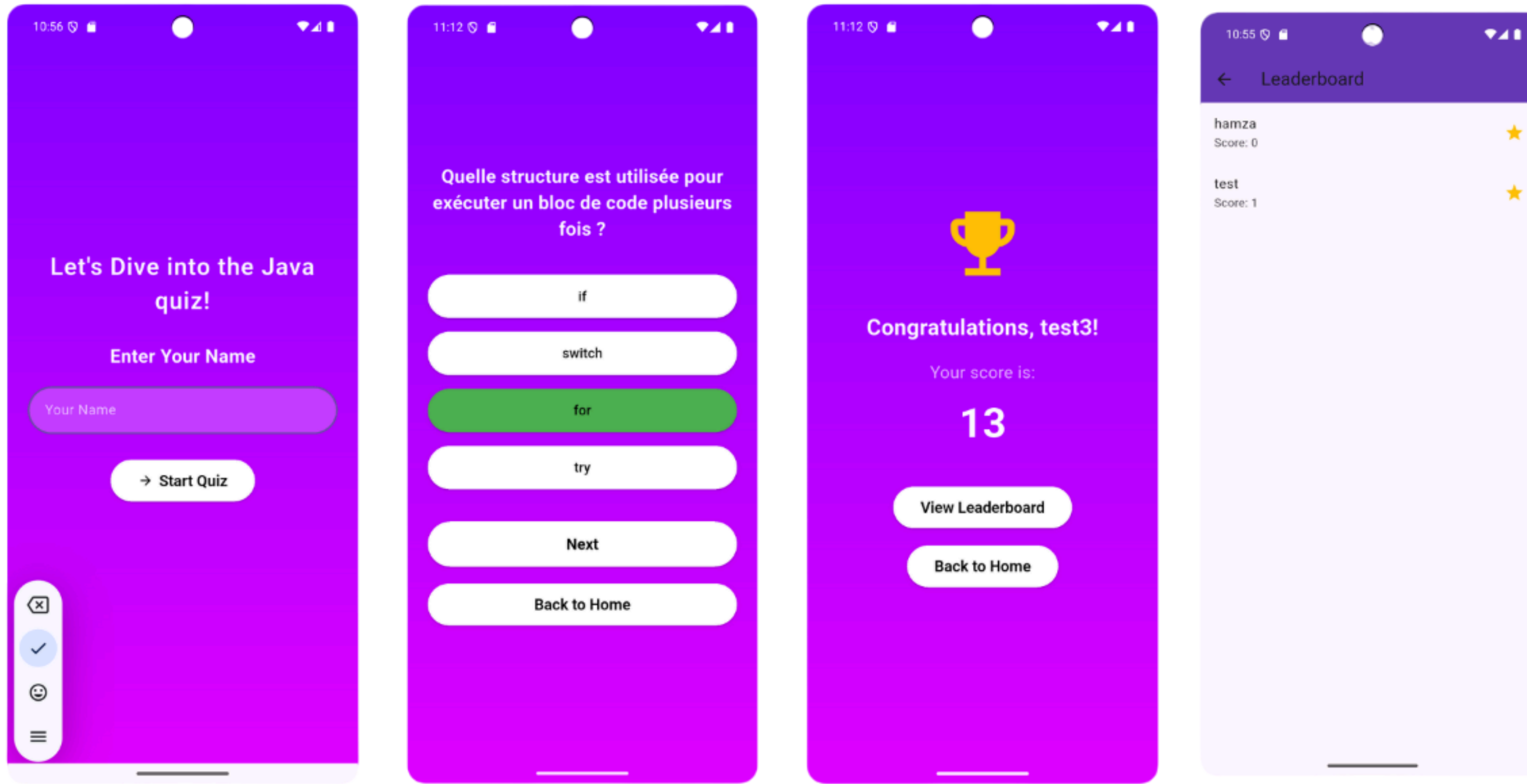
Structure MongoDB Questions

```
{
  "_id": ObjectId("..."),
  "question": "Quel mot-clé est utilisé pour hériter d'une classe en Java ?",
  "options": [
    "implement",
    "inherit",
    "extends",
    "super"
  ],
  "answer": 2
}
```

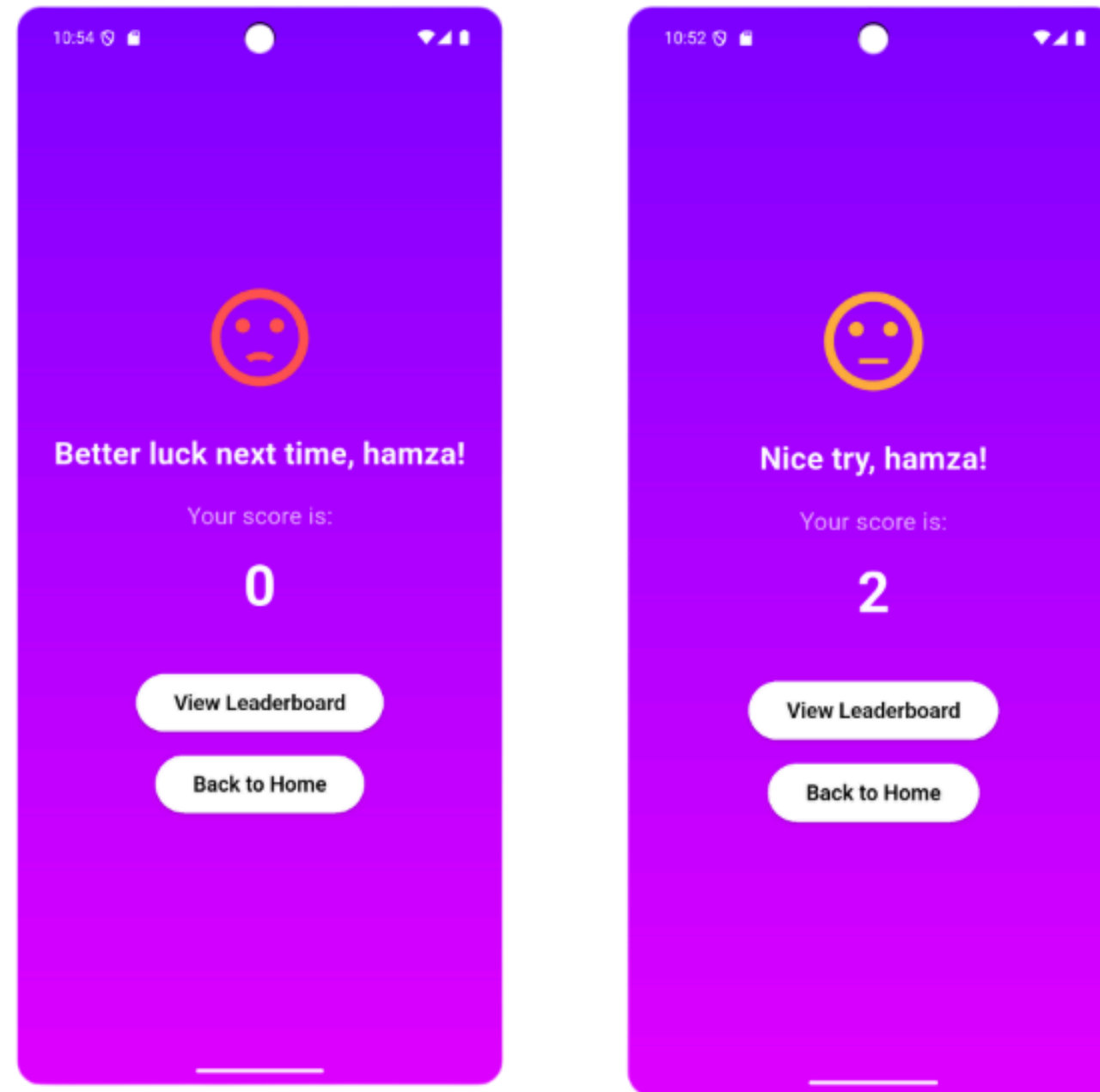
MongoDB Results

```
{
  "_id": ObjectId(".."),
  "joueur": "hamza",
  "score": 10,
  "date": "2025-04-25T10:54:10.035630"
}
```

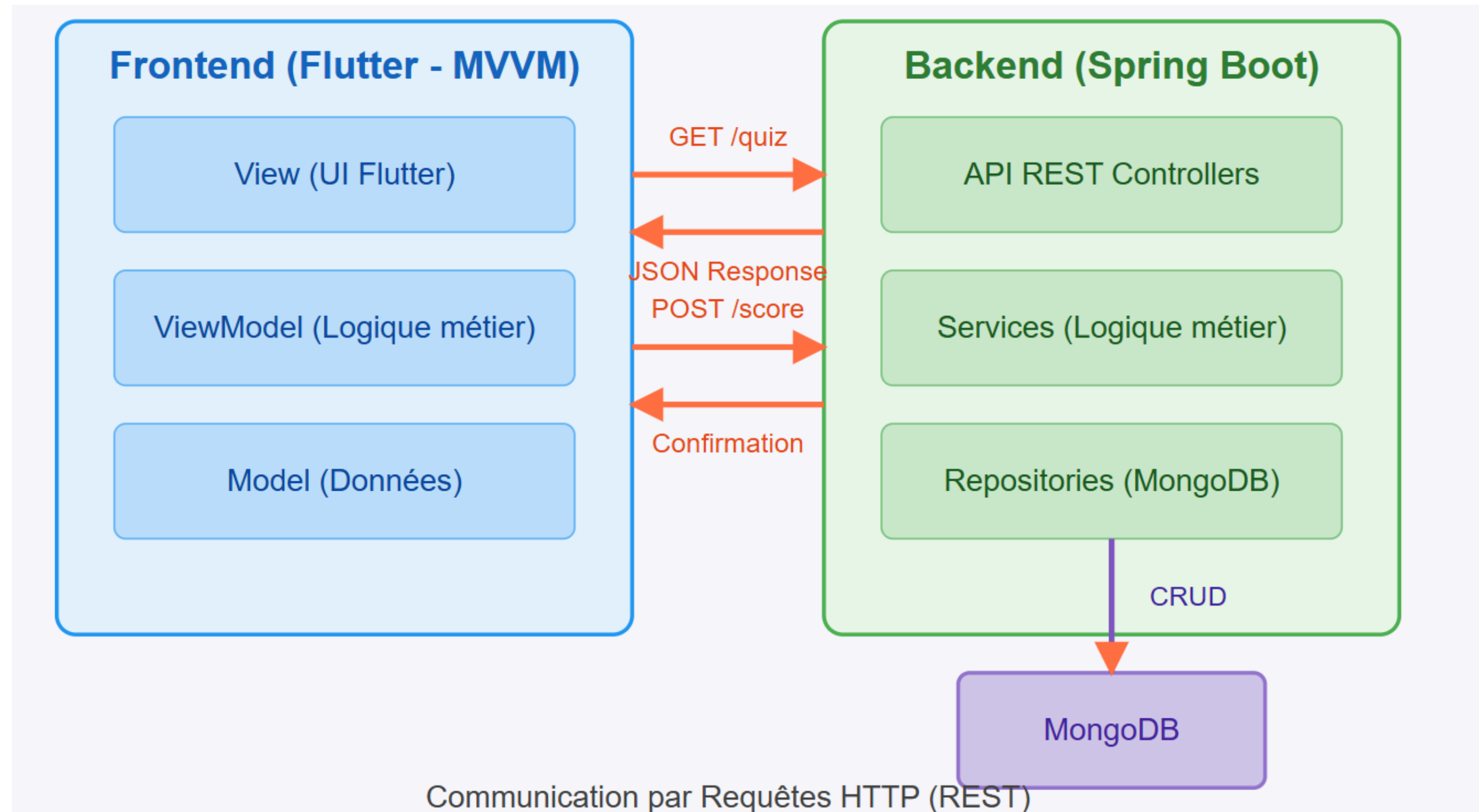
Captures d'Ecran de l'Application



Captures d'Écran de l'Application



Explication de l'architecture (frontend/backend).



Conclusion

Projet d'application de quiz vise à fournir une plateforme interactive et engageante pour les utilisateurs souhaitant tester leurs connaissances en Java. Grâce à l'utilisation de Flutter pour le frontend et de Spring Boot pour le backend, l'application offre une expérience fluide et performante. L'intégration avec MongoDB permet une gestion efficace des données, et l'architecture MVVM assure une séparation claire des responsabilités, favorisant une meilleure maintenabilité et évolutivité du projet.

**Merci pour
votre attention !**



Hamza boubaker