

Romain Manuel, Victor Louvet et Théo Maunier

Application Mobile QuizzR&T



QuizzR&T

Sommaire

1 - Introduction

2 - Présentation de l'application

3 - Partie Front-End

4 - Partie Back-End

5 - Bilan du projet

6 - Conclusion Générale

7- Répartitions des tâches

Introduction

Les objectifs :

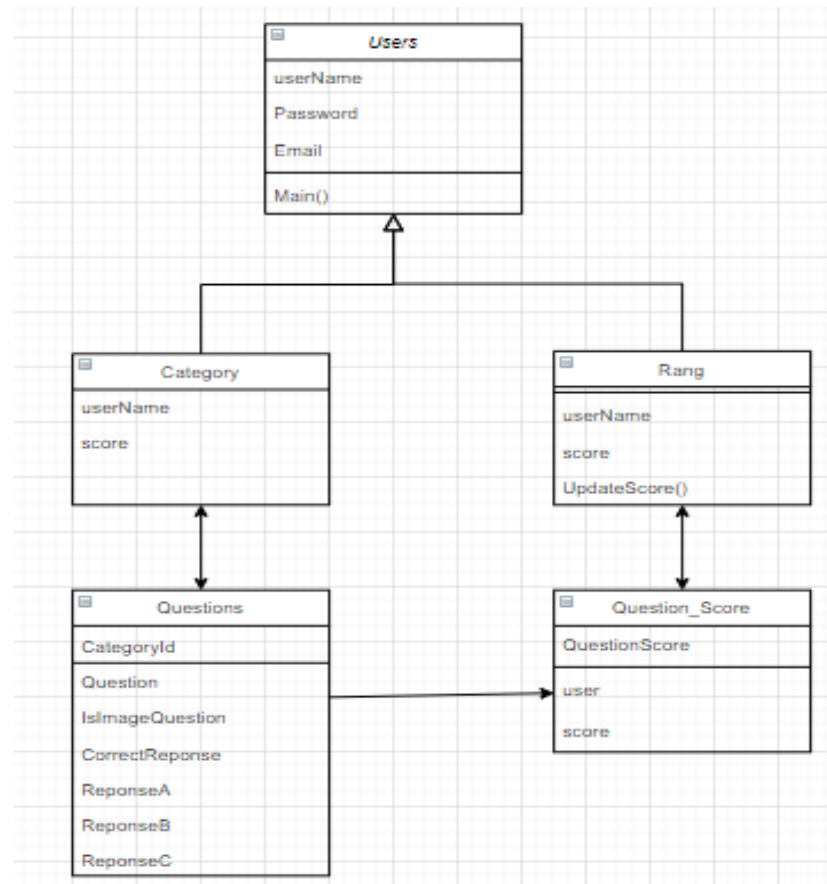
QuizzR&T a pour principal but de tester les connaissances des futurs ou des actuels étudiants grâce à une simple application. Que ce soit pour s'entraîner dans les matières scientifiques et techniques ainsi que d'améliorer ses compétences , cette application est là pour répondre aux différents besoins des Étudiant ou de ceux qui voudrons connaître cette formation .

Cela permettra ainsi à long terme de proposer une application unique non seulement réserver à l'IUT de Nice Sophia-Antipolis pour le département R&T mais pour les aux autre IUT de France. QuizzR&T a également la prétention de vouloir exister ailleurs et d'être disponible sur les différentes plateformes comme Google Play ou encore F-droïde

L'application est donc la première en son genre. Pour l'instant, uniquement disponible sous Android, son but serait également d'être reproduit sous iOS et Windows Phone.

Introduction

Architecture de notre application :



Présentation de l'application

A propos de QuizzR&T :

QuizzR&T est une application Android qui utilise une base de donnée en temps réel qui est Firebase, regroupées sous cinq grandes catégories :

'Category', 'Questions', 'Question_Score', 'Users', et 'Rang'.

Il suffit tous simplement de s'inscrire puis de se connecter pour accéder aux différentes matières proposées.

Ces matières sont classée selon leur niveau qui sont PRE DUT, DUT1 et DUT2 et les questions sont limité par un chrono de 30 sec par question.

En se basant sur vos résultats, l'application vous permet de vous classé par rapport aux autres personnes inscrite sur l'application et ainsi de vous donner un Feedback par rapport a votre avancement .

QuizzR&T a été conçue comme une solution agile et performante qui répond à un besoin tout en alliant ergonomie et fluidité.

Partie Front-End

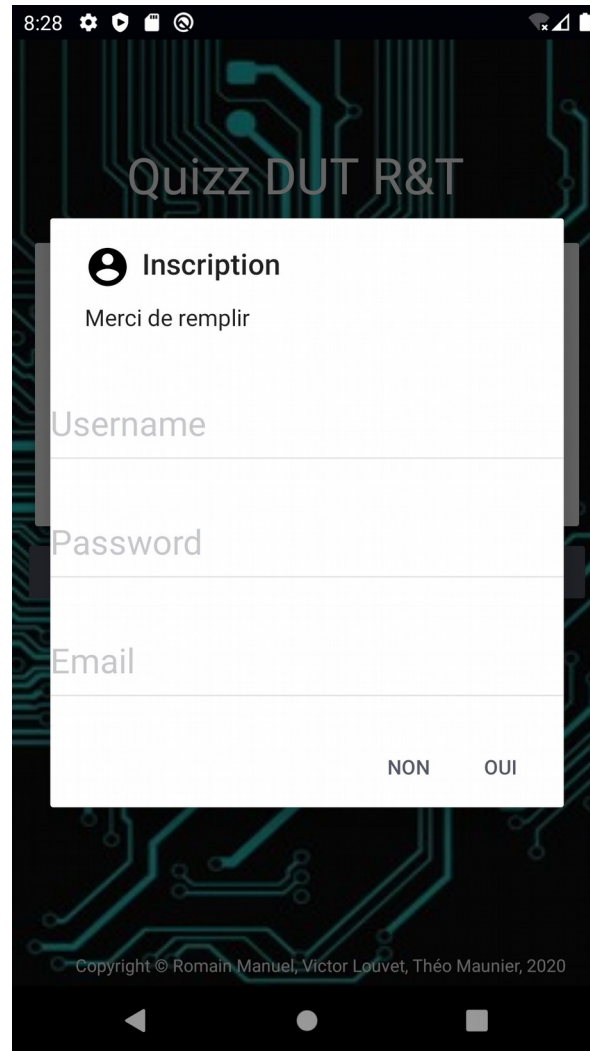
Fonctionnalité :

Cette partie a pour but de décrire les fonctionnalités des différents services proposés par l'application. L'application est composée de cinq grandes catégories ('Catégorie', 'Question_Score', 'Rang', 'Questions' et 'Users').

Les avantages à utiliser notre application :

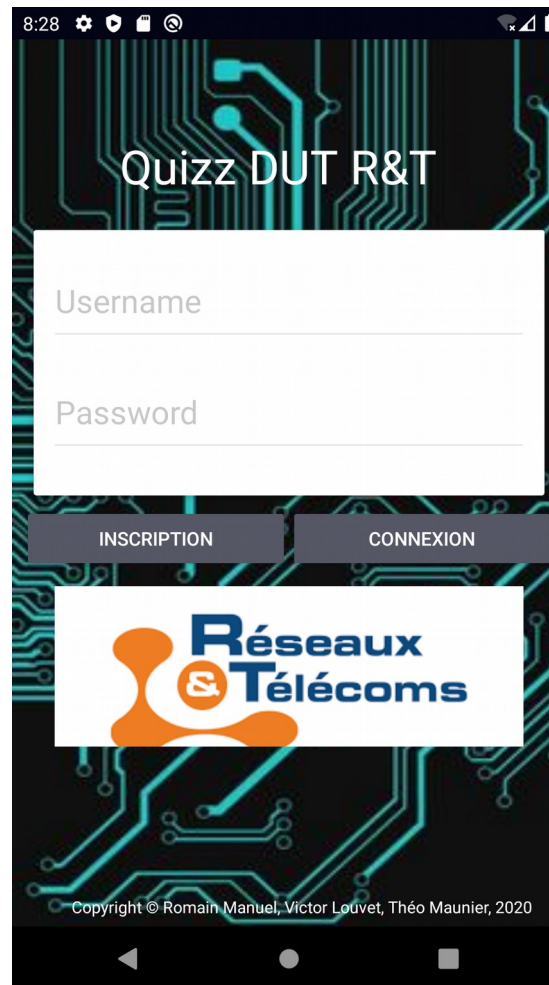
- Très facile d'utilisation**
- Possibilité de voir son rang parmi d'autre utilisateur**
- Application fluide et beau design**
- Contenu des matières importantes**
- On peut s'entraîner à l'infini.**
- Penser pour être ludique avec l'utilisation d'un score pour chaque réponse juste**
- Une proposition des questions en accord avec le niveau attendu de la formation R&T**

Partie Front-End



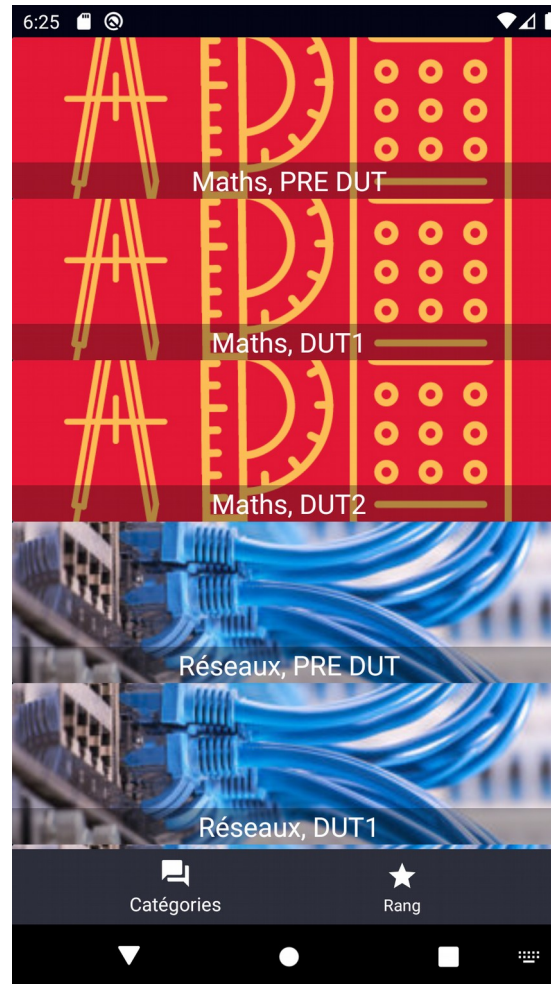
1) Page Inscription

Partie Front-End



2) Page d'entrée

Partie Front-End



3) Page d'accueil

Partie Front-End



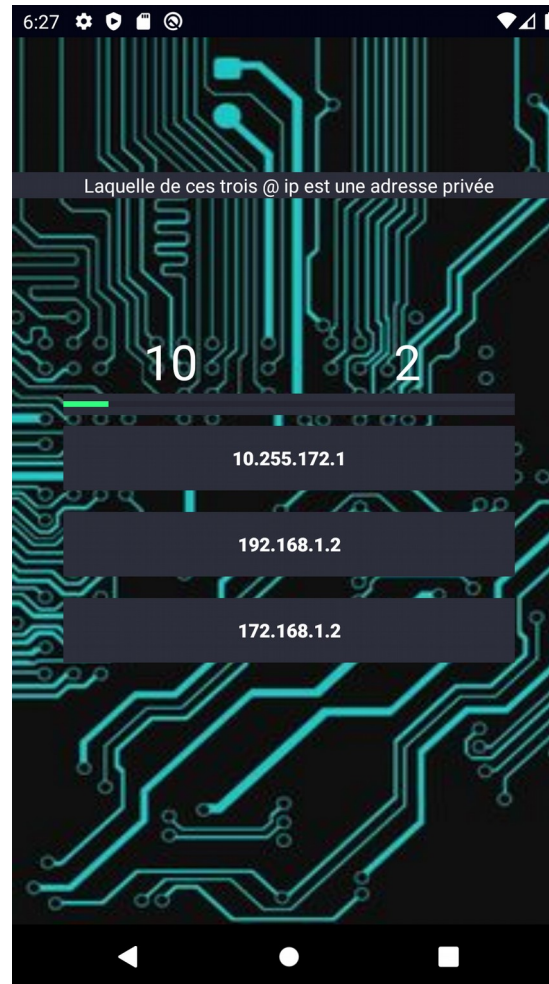
3) Page principale lors d'une sélection d'une matière

Partie Front-End



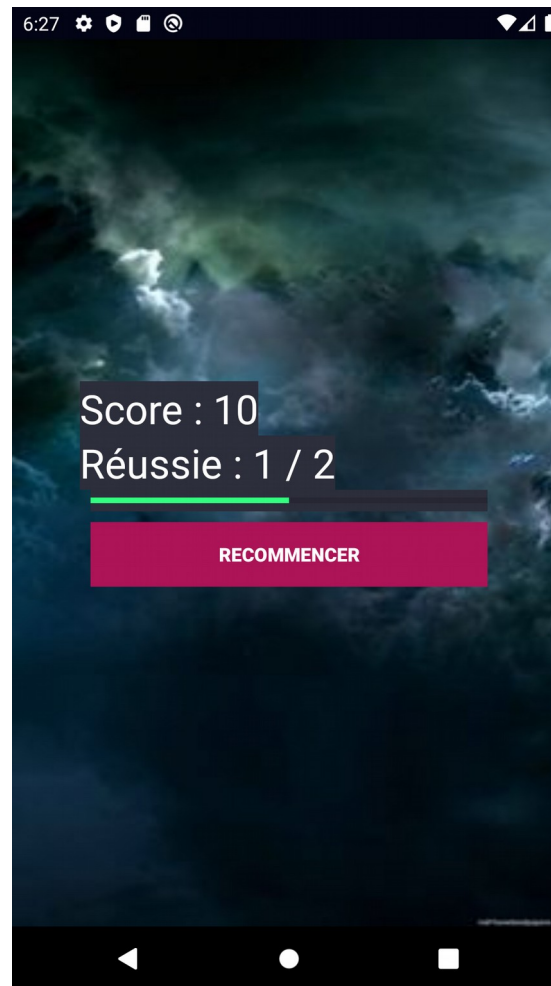
4)Page de la première question

Partie Front-End



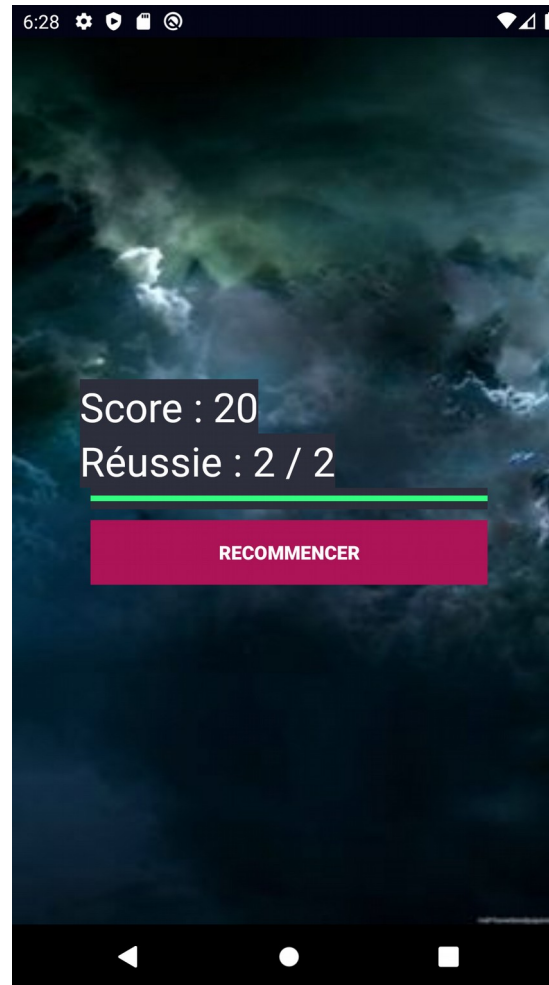
5) Page de la seconde question

Partie Front-End



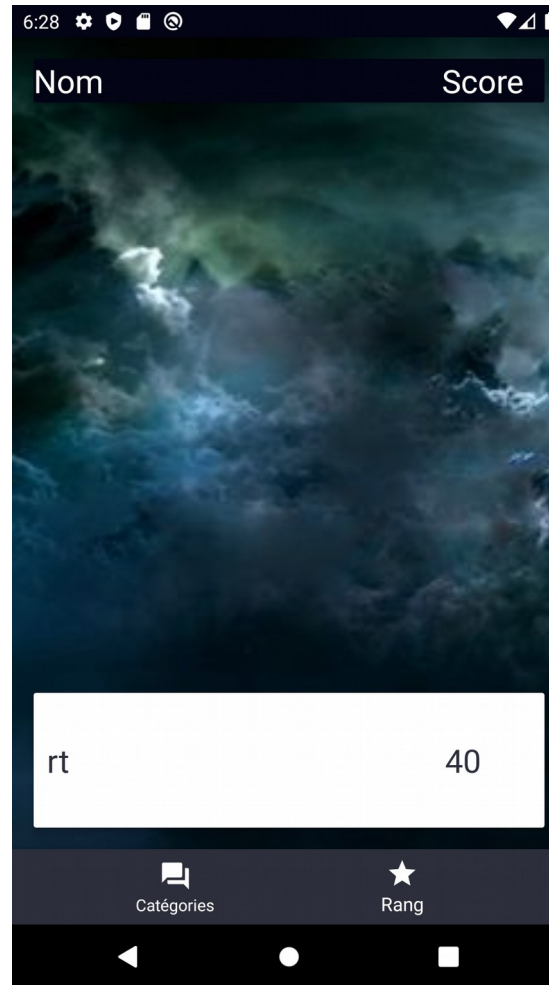
6) Page de fin des questions de la matière choisie

Partie Front-End



7) Page de fin des question de la matière choisie (Second essai)

Partie Front-End



8) Page Rang

Partie Back-End

```
private void signIn(final String user, final String pwd){
    users.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if (dataSnapshot.child(user).exists()) {
                if(!user.isEmpty()) {
                    User login = dataSnapshot.child(user).getValue(User.class);
                    if(login.getPassword().equals(pwd)){
                        Toast.makeText( context: MainActivity.this, text: "login ok", Toast.LENGTH_SHORT).show();
                        Commun.currentUser = login;
                        Intent homeActivity = new Intent( packageContext: MainActivity.this, Home.class);
                        startActivity(homeActivity);
                        finish();
                    }

                    else
                        Toast.makeText( context: MainActivity.this, text: "Mauvais password", Toast.LENGTH_SHORT).show();
                }
            }
            else{
                Toast.makeText( context: MainActivity.this, text: "merci de remplir votre Username", Toast.LENGTH_SHORT).show();
            }
        }
    })
    else
        Toast.makeText( context: MainActivity.this, text: "L'utilisateur n'existe pas", Toast.LENGTH_SHORT).show();
}
```


Partie Back-End

```
private void showQuestion(int index){
    if(index < totalQuestion){
        thisQuestion++;
        textQuestionNum.setText(String.format("%d", thisQuestion, totalQuestion));
        progressBar.setProgress(0);
        progressValue=0;
        if(Commun.questionList.get(index).getIsImageQuestion().equals("true")){
            Picasso.get() //si image
                .load(Commun.questionList.get(index).getQuestion())
                .into(question_image);
            question_image.setVisibility(View.VISIBLE);
            question_text.setVisibility(View.INVISIBLE);
        }
        else{ //si texte
            question_text.setText(Commun.questionList.get(index).getQuestion());
            question_image.setVisibility(View.INVISIBLE);
            question_text.setVisibility(View.VISIBLE);
        }
        btnA.setText(Commun.questionList.get(index).getReponseA());
        btnB.setText(Commun.questionList.get(index).getReponseB());
        btnC.setText(Commun.questionList.get(index).getReponseC());

        mcountDown.start(); //commence le compteur
    }
}
```

Partie Back-End

```
options= new FirebaseRecyclerOptions.Builder<Category>()
    .setQuery(categories, Category.class)
    .build();

adapter= new FirebaseRecyclerAdapter<Category, CategoryViewHolder>(options) {
    @Override
    protected void onBindViewHolder(@NonNull final CategoryViewHolder viewHolder, int position, @NonNull final Category model) {
        viewHolder.t1.setText(model.getName());
        Picasso.get().load(model.getImage()).into(viewHolder.i1, new Callback() {
            @Override
            public void onSuccess() {
                viewHolder.setItemClickListener((view, position, isLongClick) -> {
                    Toast.makeText(getActivity(), String.format("%s|%s", adapter.getRef(position).getKey(), model.getName()), Toa
                    Intent StartGame = new Intent(getActivity(), Start.class);
                    Commun.categoryId = adapter.getRef(position).getKey();
                    startActivity(StartGame);
                });
            }
        });
    }
}
```

Partie Back-End

```
viewHolder.t1.setText(model.getName());
Picasso.get().load(model.getImage()).into(viewHolder.i1, new Callback() {
    @Override
    public void onSuccess() {
        viewHolder.setItemClickListener((view, position, isLongClick) → {
            Toast.makeText(getActivity(), String.format("%s|%s", adapter.getRef(position).getKey(), model.getName()),
            Intent StartGame = new Intent(getActivity(), Start.class);
            Commun.categoryId = adapter.getRef(position).getKey();
            startActivity(StartGame);
        });
    }

    @Override
    public void onError(Exception e) {
    }
});
}

@NonNull
@Override
public CategoryViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.category_layout, parent, attachToRoot: false);
    return new CategoryViewHolder(view);
}
```

Partie Back-End

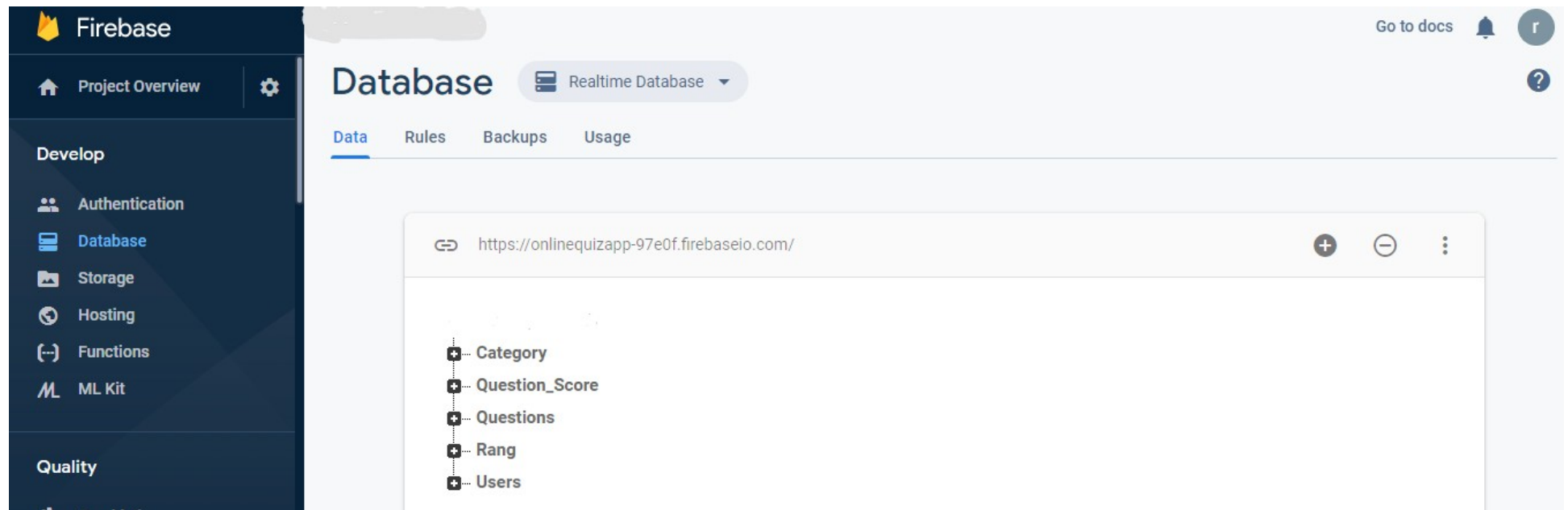
```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".CategoryFragment">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/listCategory"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        >

    </androidx.recyclerview.widget.RecyclerView>

</FrameLayout>
```

Partie Back-End



Bilan du projet

Nous avons trouvé que ce projet a été très enrichissant et avec la découverte de l'API Android Studio nous avons pu développer nos compétences dans la programmation en particulier les langages Java, XML et JSON. C'est aussi la découverte de la base de donnée Firebase qui a été très intéressante à utiliser.

La fonctionnalité qu'a Android de simuler et de tester notre application sur nos téléphones a été ludique.

Conclusion générale

L'accomplissement du projet à été au-delà de nos attente mais non sans grande difficulté. Les objectifs principaux ont été atteint et nous espérons que notre application sera à l'avenir amélioré.

Les grandes difficultés sont :

Une application qui parfois bugger lors de la simulation.

Un temps de travail fournit très important qui a été consacrer uniquement sur ce projet.

Il a fallut résoudre des problèmes concernant les différents modules utilisé.

Et beaucoup de code à fournir voir complexe.

Répartition des Tâches

Le durée total de la programmation de l'application est de 134h

-Victor Louvet c'est concentré sur la partie question, questionScore et le teste des questions : temps de travail 42h

-Théo Maunier sur le design, le choix des questions et l'interface utilisateur : 42h

-Romain Manuel sur la partie Rang, l'intégration des différentes parties pour que l'application dans sa globalité fonctionne et la mise en place de la base de donnée Firebase : 50h