



Documentation Technique

MATE

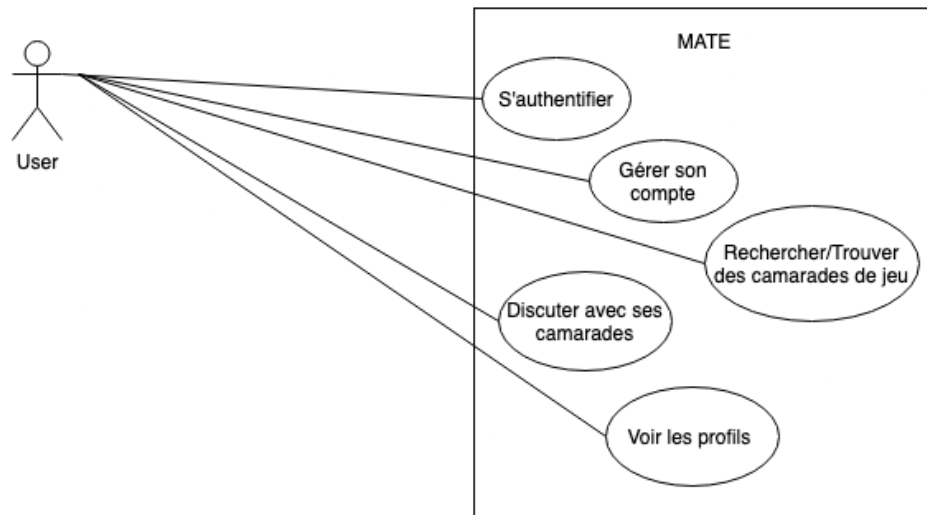
Projet : PPE4 - MATE		
Objectif : Documentation Technique MATE		
Créé par : Golder Lucas, Grande Vincent, Haller Antoine	Vérifié par : Golder Lucas, Grande Vincent, Haller Antoine	
Version : 1.0	Date : 22/01/2021	Pages : 9

1/ Documents

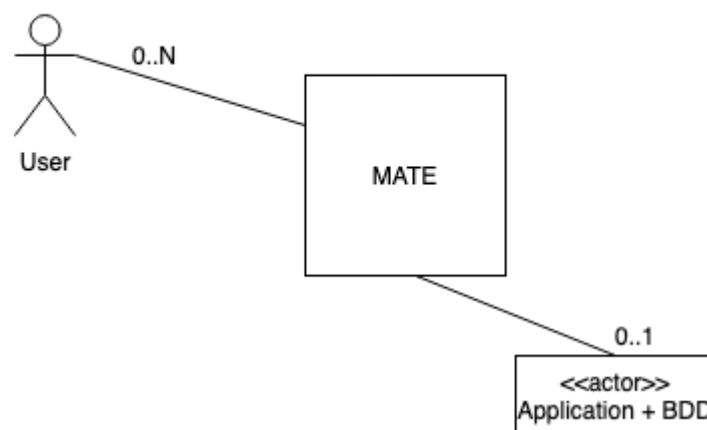
1.1 Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour fournir une vue d'ensemble du comportement fonctionnel d'un système logiciel.

Dans notre cas de diagramme de cas d'utilisation permet de fournir une vue d'ensemble du comportement de l'application MATE.

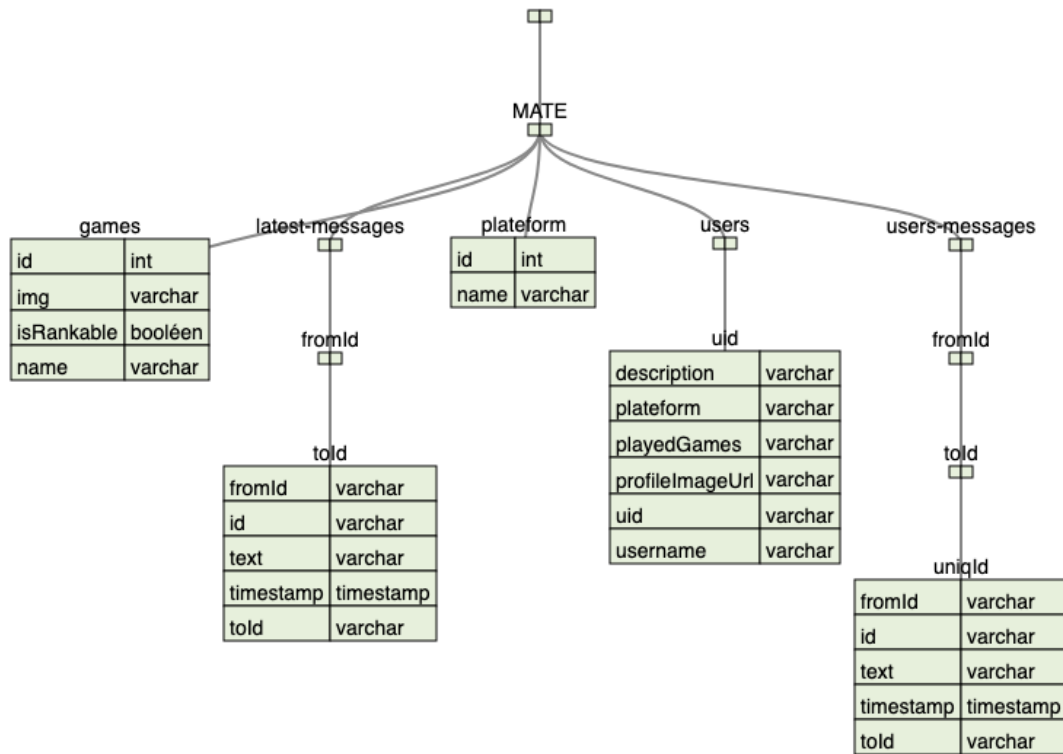


1.2 Diagramme de contexte statique



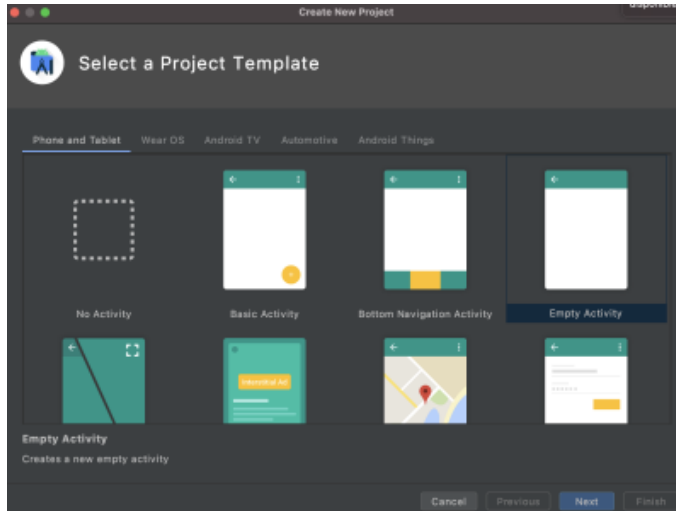
Le diagramme de contexte statique délimite le domaine d'étude en précisant ce qui est à la charge du système et en identifiant l'environnement extérieur au système étudié avec lequel ce dernier communique.

1.3 Base de données (représentation de la base NoSQL)

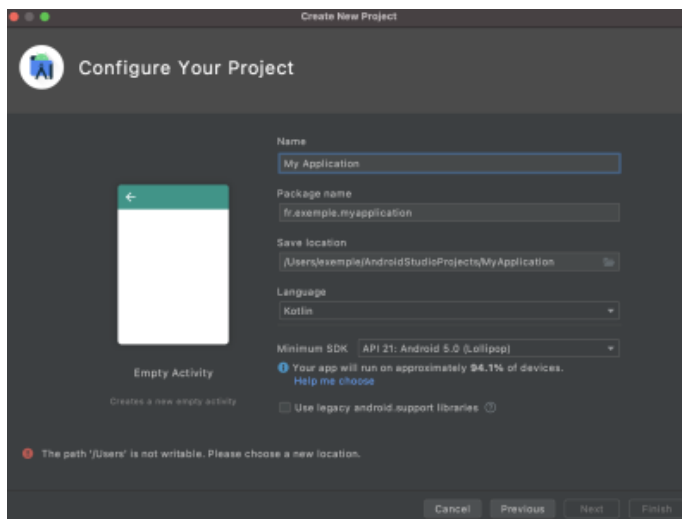


2/ Création du projet Android Studio

Ouvrez Android Studio, Puis dans le menu file, new, new project



Ici, il est possible de choisir un template pour notre activité principale, nous allons choisir “Empty activity” afin d’avoir une activité vide. Cliquez ensuite sur le bouton Next.

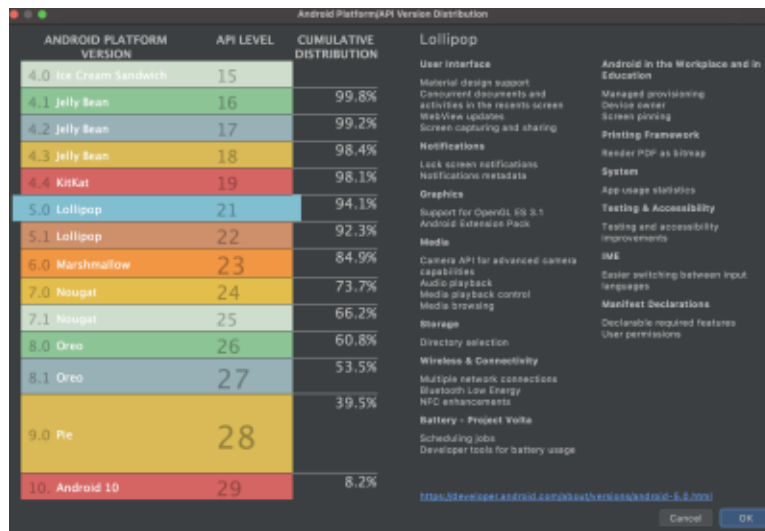


Sur cette page nous allons choisir :

- le nom de notre application
- le nom du package, attention, le nom du package est très important pour la publication de notre application sur le play store, il doit s'agir du nom de domaine à l'envers suivi du nom de l'application. Par exemple, pour une application nommée “azerty” et un nom de domaine nommé “uiop.fr”, le nom du package sera fr.uiop.azerty

- l'emplacement de sauvegarde de notre projet, le langage utilisé (Kotlin ou Java) qui sera Kotlin dans notre cas

Minimum SDK correspond à la version minimum requise pour faire fonctionner notre application, ici aussi attention, il ne faut pas choisir une version d'android trop récente si l'on veut que notre application soit disponible pour le plus grand nombre d'utilisateurs possible. Nous voyons que notre application pourra fonctionner sur environ 94% des appareils si nous choisissons android 5.0, il est possible de cliquer sur "Help me choose" afin d'avoir plus d'informations sur comment choisir la bonne version.



- Pour faire simple, une version plus récente d'android permettra d'utiliser des fonctionnalités plus évoluées mais pourra fonctionner sur moins d'appareils, à vous de voir de quelles fonctionnalités vous aurez besoin pour votre application.

Une fois tous les champs remplis, nous pouvons cliquer sur Finish afin de créer le projet.

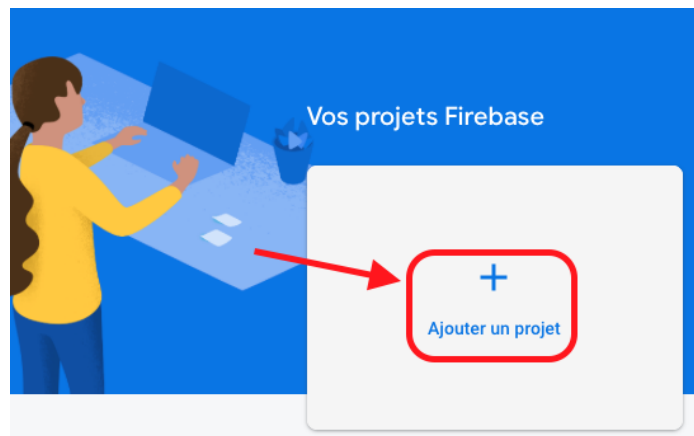
3/ Intégration du service Firebase

3.1 Création du projet

Rendez vous sur la console Firebase en cliquant sur le lien suivant :

<https://console.firebase.google.com/u/0/>

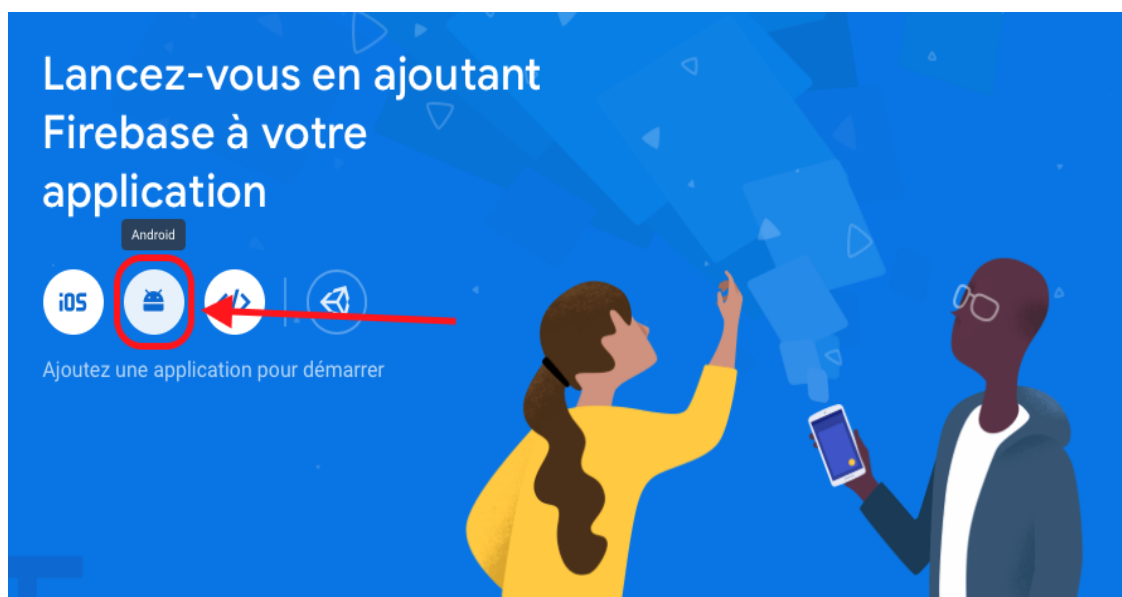
Cliquez sur Ajouter un projet :



Donnez un nom à votre projet, activez ou non Google Analytics pour votre projet puis cliquez sur créer le projet.

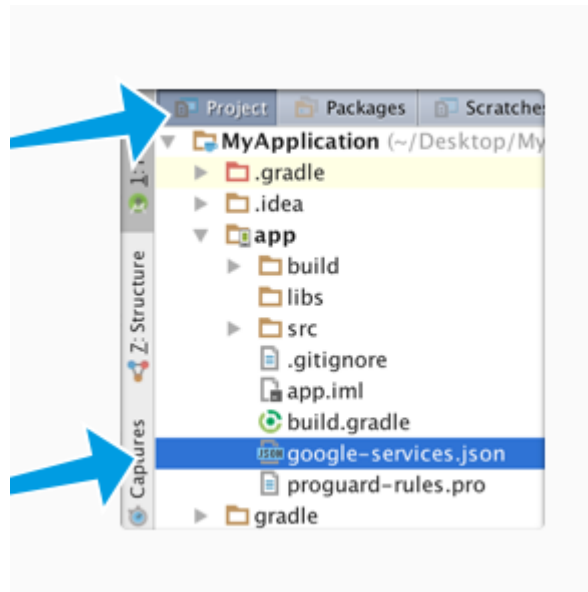
3.2 Ajout des services Firebase sur votre projet Android Studio

Cliquez sur l'icône Android.



Renseignez dans le formulaire, le nom de package que vous avez mis lors de la création du projet Android Studio puis le nom de l'application puis cliquez sur "Enregistrer l'application".

Téléchargez le fichier google-services.json en cliquant sur le bouton “Téléchargez google-services.json” et ajoutez le à la racine de l’application :



Modifiez le fichier build.gradle qui se trouve à la racine du projet en ajoutant les lignes nécessaires pour utiliser le plugin Firebase. Ces lignes sont repérables par le logo copie en bout de ligne comme sur l’exemple ci-dessous:

```
buildscript {
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
    }
    dependencies {
        ...
        // Add this line
        classpath 'com.google.gms:google-services:4.3.5'
    }
}

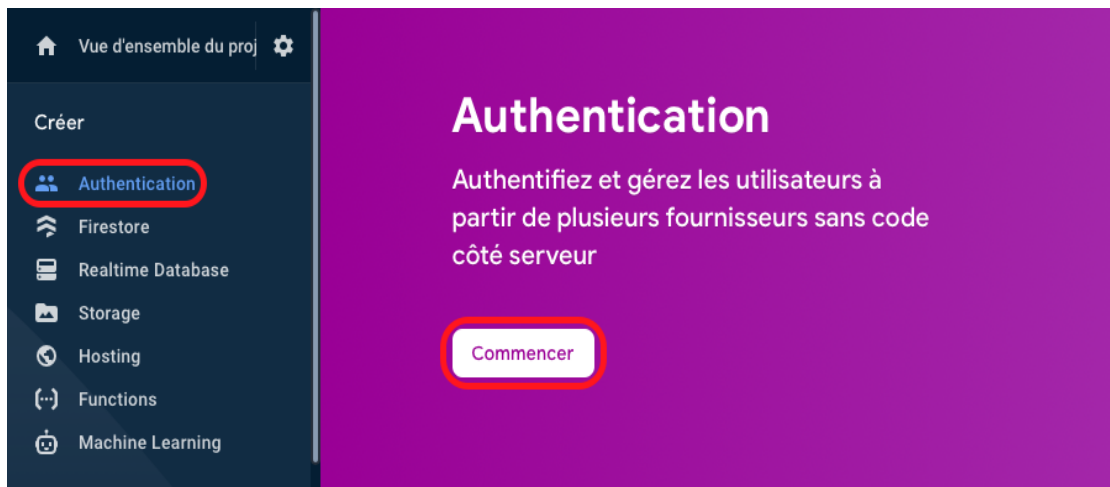
allprojects {
    ...
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
        ...
    }
}
```

Ensuite sélectionnez Kotlin à la place de Java et ajoutez les lignes qui se terminent par le logo copie dans le fichier build.gradle qui se situe dans le dossier racine de votre application. Pour finir cliquez sur le bouton Sync Now qui devrait apparaître en haut de votre page Android Studio.

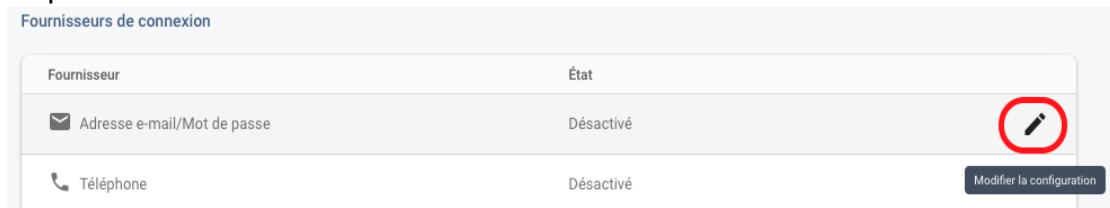


3.4 Configuration de Firebase Authentication

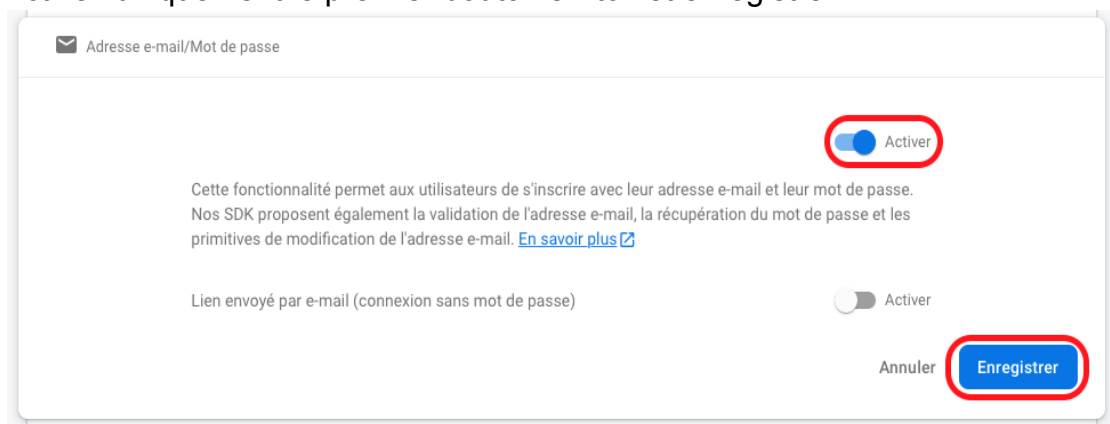
Rendez-vous sur l'onglet Authentication puis cliquez sur Commencer.



Cliquez sur l'icône modifier au bout de la ligne authentication par email/mot de passe.



Activez uniquement le premier bouton switch et enregistrez.



Dans le fichier build.gradle qui se trouve à la racine de votre application, ajoutez la ligne suivante dans les dépendances :



3.5 Inscription / connexion

Pour inscrire un utilisateur il faut utiliser les fonctions que l'on peut retrouver dans la documentation de Firebase. Il suffit de créer une simple interface graphique avec des entrées texte et un bouton de connexion/inscription. Lors du clic sur le bouton, des variables récupéreront les valeurs entrées dans les entrées texte et ces mêmes variables devront être passées en paramètre des fonction d'inscription/connexion récupérées dans la documentation Firebase.

3.6/ Utiliser Firebase Storage

Il faut commencer par ajouter l'implémentation `implementation 'com.google.firebase:firebase-storage-ktx'` dans le fichier build.gradle situé à la racine de votre application. Ensuite pour stocker des fichiers dans Firebase Storage, utilisez une entrée fichier sur votre interface graphique, récupérez ce fichier dans une variable puis passez la en paramètres de la fonction prévue à l'envoi de fichier vers Firebase Storage que vous pouvez retrouver dans la documentation. Pour récupérer un fichier stocké dans Firebase Storage, il faut également se référer aux fonctions de la documentation.

4/ Mise en place de l'interface graphique

Pour mettre en place l'interface graphique de votre application en Kotlin sur Android Studio, deux solutions s'offrent à vous :

- L'utilisation du mode design d'Android Studio qui vous met à disposition les composants graphiques les plus courants avec un bandeau de configuration afin de les adapter à vos besoins.

- La deuxième solution est le mode code, pour ce faire il faut connaître le code XML des composants graphiques dont vous avez besoin.

Il est également totalement possible d'utiliser les deux solutions, par exemple utiliser le mode design pour ajouter votre composant et lui ajouter des contraintes puis le personnaliser au niveau du code.