09/01/2024

Guide Développeur

Développement mobile

Alexis COMMEAT

Lilian DAMIEN

IRA-4

Introduction

Bienvenue dans le guide développeur de BeerApp, une application conçue pour les amateurs de bière et les professionnels du monde brassicole. Ce document offre une vue d'ensemble complète de l'architecture logicielle, des composants et des données manipulées de notre application mobile Android.

BeerApp a été développée avec l'objectif principal de fournir une interface simple, réactive et riche en fonctionnalités pour explorer, cataloguer et découvrir une variété de bières du monde entier. L'application offre des fonctionnalités telles que :

- L’authentification
- Le listing des bières
- L’ajout de nouvelles bières
- L’utilisation d’une IA d’OpenAI d'extraction de mots (Keywords)

Ce guide est destiné aux développeurs, ainsi qu'aux parties prenantes souhaitant comprendre le fonctionnement de notre application BeerApp. En parcourant ce guide, vous retrouverez :

L'Architecture Logicielle : Exploration de l'architecture de l'application, y compris notre approche pour gérer les données des bières, l'interaction utilisateur et l'intégration avec des services externes.

Classes et Interfaces : Détails sur les principaux composants de l'application, y compris les activités, les fragments, les adaptateurs et les gestionnaires de données.

Modèle de Données : Présentation de la structure de données, notamment les entités de bière, les préférences utilisateur et les interactions avec la base de données Firebase.

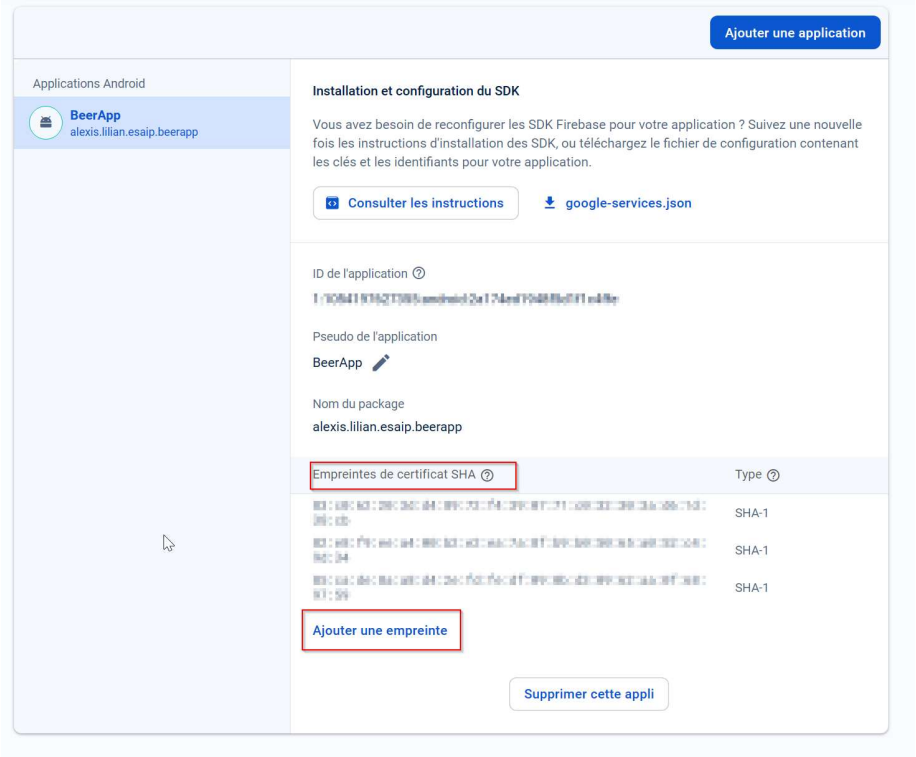
Des diagrammes UML sont inclus pour fournir une représentation visuelle de l'architecture, des classes et des flux de données, facilitant ainsi la compréhension du design et de la structure de l'application.

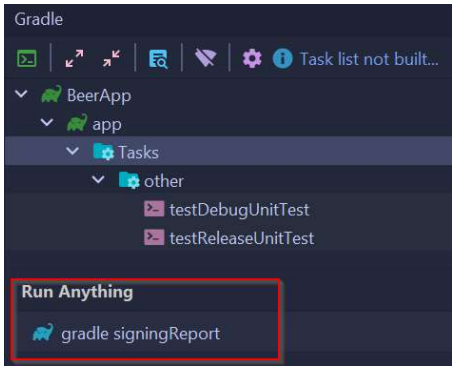
1) Configuration & Dépendances

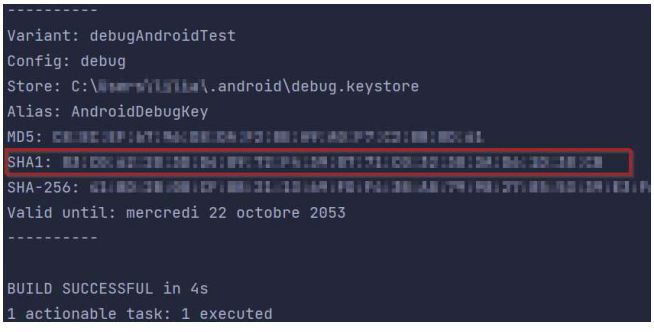
- Android Gradle version 8.2
- Compil SDK version API 34 version_android14_upside_downcake
- Firebase_auth version 22.3.0
- Firebase_bom version 32.7.0
- Firebase_database version 20.3.0
- Volley version 1.2.1
- Picasso version 2.71 828

2) Installation & Utilisation du code

- Extraire ou télécharger le code
- Lire attentivement le README
- Sur android : se rendre dans la tool bar, catégorie build -> make project
- Pour utiliser l'application avec l'émulateur android studio, il faut renseigner la clé sha-1 dans le projet firebase (Nous vous recommandons d'utiliser votre smartphone)





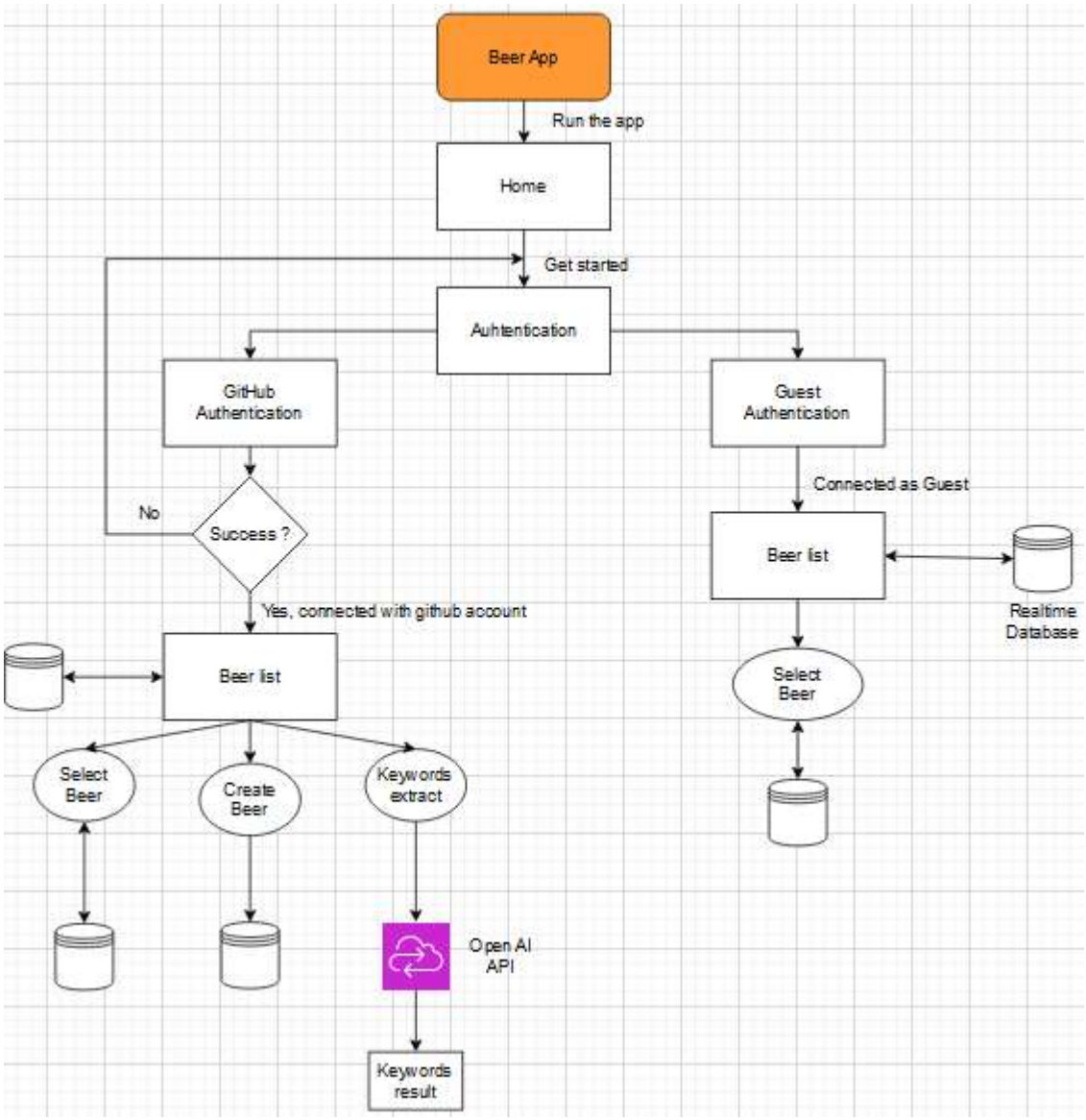


3) Pseudo algorithme : Parcours utilisateur & Echanges

L'objectif de cet algorithme est de comprendre le parcours d'un utilisateur sur l'application. On y retrouve les différentes activités de l'application et les différentes fonctionnalités. Aussi le parcours utilisateur n'est pas le même selon la méthode d'authentification choisi (Github ou Guest).

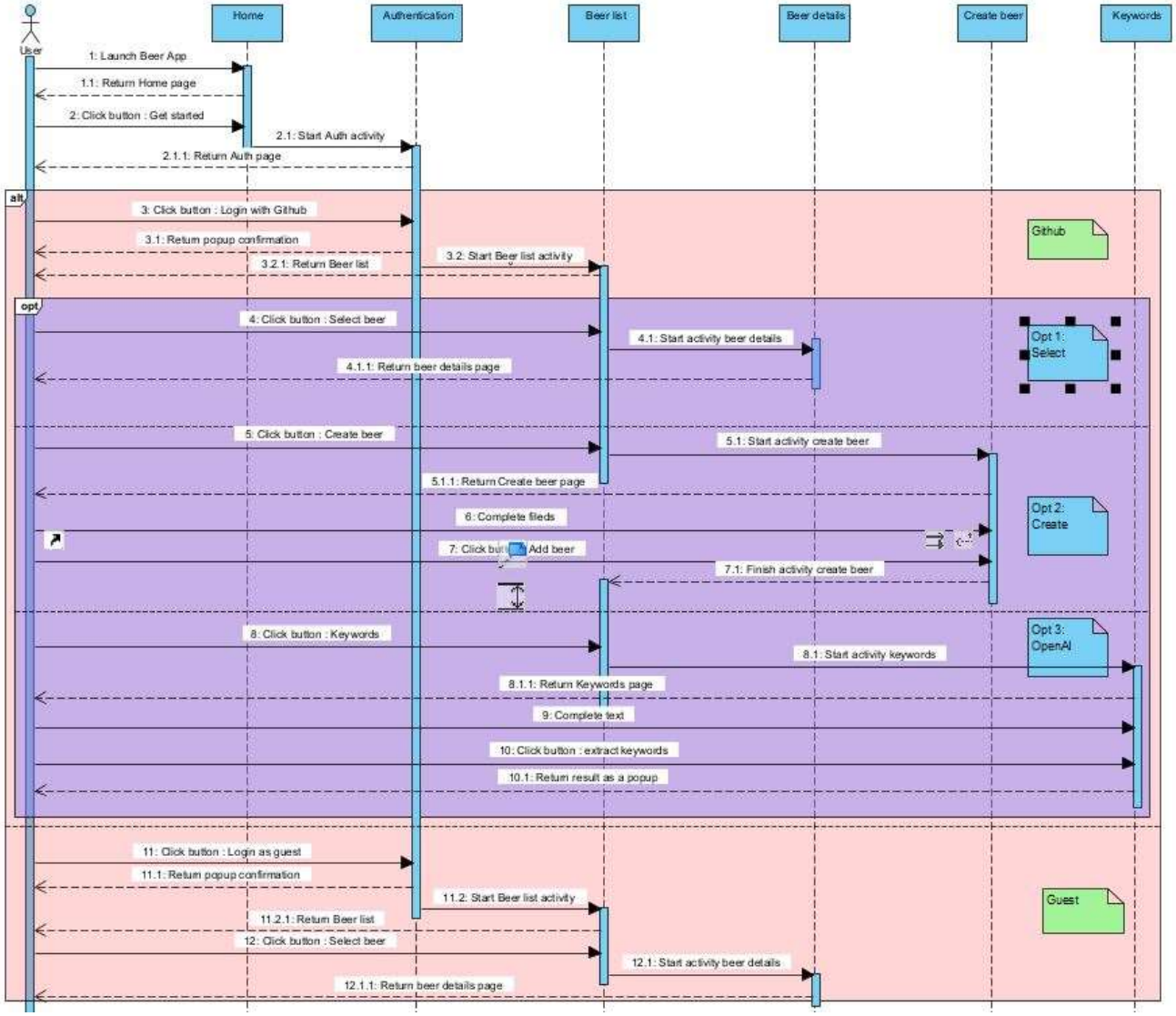
Nous précisons bien « pseudo algorithme » car il reprend en partie la nomenclature algorithmique combiné à un développement logique simplifié. Cet algorithme offre une vision globale du parcours utilisateur, en mettant en évidence les étapes clés et les choix liés à l'authentification. Il s'agit d'une représentation simplifiée et logique du flux de l'application, permettant de comprendre les différentes voies que peut emprunter un utilisateur.

Nous aborderons ultérieurement dans ce guide les échanges de données ainsi que les échanges d'API.



4) Diagramme de séquence

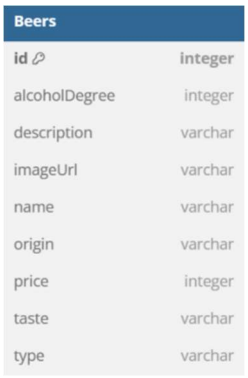
Ce diagramme de séquence est une vue plus détaillée et complète du pseudo algorithme. Il illustre les différents échanges et les différentes options de l'utilisateur. Il permet également d'avoir une première vision du cycle de vie de chaque activité.



5) Modèle de données

La base de données que nous utilisons est Firebase Realtime.

Notre modèle de données se base sur une seule table « Beers ».



Cette base de données interagit avec :

- DatabaseManager -> classe qui permet de gérer une instance de firebase database
- Beer list -> reprends tous les objets de la base (SELECT * FROM beers)
- Beer details -> sélectionne un objet dans la base (SELECT id= '… ' FROM beers)
- Beer create -> ajoute un nouvel objet (nouvelle bière) dans la base (INSERT INTO beers ('…'))

Voici un dictionnaire de données :

Nom	Type	Exemple
Id (clé primaire)	int	-NnZNLNy1G-kMHp7ttNO
alcoholDegree	int	5.8
description	varchar	« the most mexican beer »
imageUrl	varchar	« https://image_beer »
name	varchar	« Leffe »
origin	varchar	« France »
price	int	6.80
taste	varchar	« Arômes fruités »
type	varchar	« lager »

Voici une vue de la base de données Firebase :

6) Les classes

Voici l'explication globale de nos 11 classes :

MainActivity

La MainActivity sert de point d'entrée pour l'application.

AuthenticationActivity

AuthenticationActivity gère le processus d'authentification, y compris l'intégration avec GitHub mais aussi avec le système d'authentification anonyme. Elle fournit des méthodes pour traiter les succès et les échecs d'authentification.

Popup

La classe Popup est utilisée pour afficher des fenêtres contextuelles interactives dans l'application. Elle permet d'adapter le contenu (titre, image, description) de la popup selon les interactions de l'utilisateur.

DatabaseManager

DatabaseManager est une classe utilitaire qui fournit un accès centralisé à la base de données Firebase. Elle contient des méthodes pour obtenir des références à la base de données et pour activer la persistance des données.

BeerAdapter

BeerAdapter est un adaptateur personnalisé pour RecyclerView qui gère l'affichage de la liste des bières. Il contient des méthodes pour lier les données de bière à la vue et pour mettre à jour la liste lorsque les données changent.

BeerViewHolder

BeerViewHolder agit comme un conteneur pour les vues d'éléments de bière dans le RecyclerView. Il est utilisé par BeerAdapter pour remplir chaque élément de la liste avec des données.

BeerListActivity

BeerListActivity est l'activité qui affiche la liste des bières à l'utilisateur. Elle gère la configuration du RecyclerView et l'interaction avec la base de données pour récupérer et afficher les bières.

BeerCreateActivity

BeerCreateActivity offre une interface pour ajouter de nouvelles bières à la base de données. Elle contient des champs de saisie pour les détails de la bière et interagit avec DatabaseManager pour créer des enregistrements dans Firebase.

BeerDetailsActivity

BeerDetailsActivity fournit une vue détaillée pour une bière spécifique. Elle récupère les données de la bière à partir de la base de données et les affiche à l'utilisateur.

KeywordsActivity

KeywordsActivity est une activité qui permet aux utilisateurs d'interagir avec l'API de mots-clés d'OpenAI. Elle gère la saisie de l'utilisateur, envoie des requêtes à l'API et affiche les résultats via une popup.

Beer

La classe Beer est un modèle de données qui représente une bière dans l'application. Elle contient des propriétés telles que le nom, la description, l'origine, le degré d'alcool, l'image, etc., et des méthodes pour obtenir et définir ces propriétés.