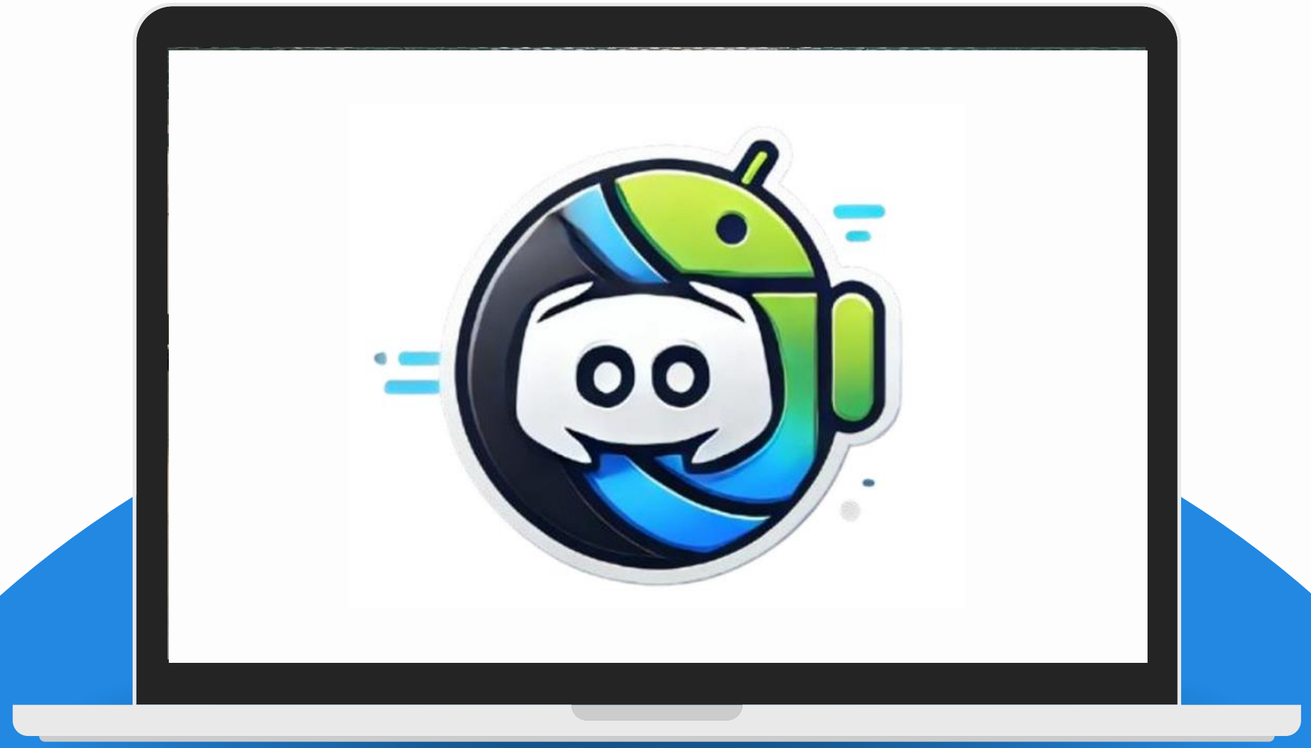
SAE 302 -- Développer des applications communicantes

UNIVERSITÉ CÔTE D'AZUR

TraceCord



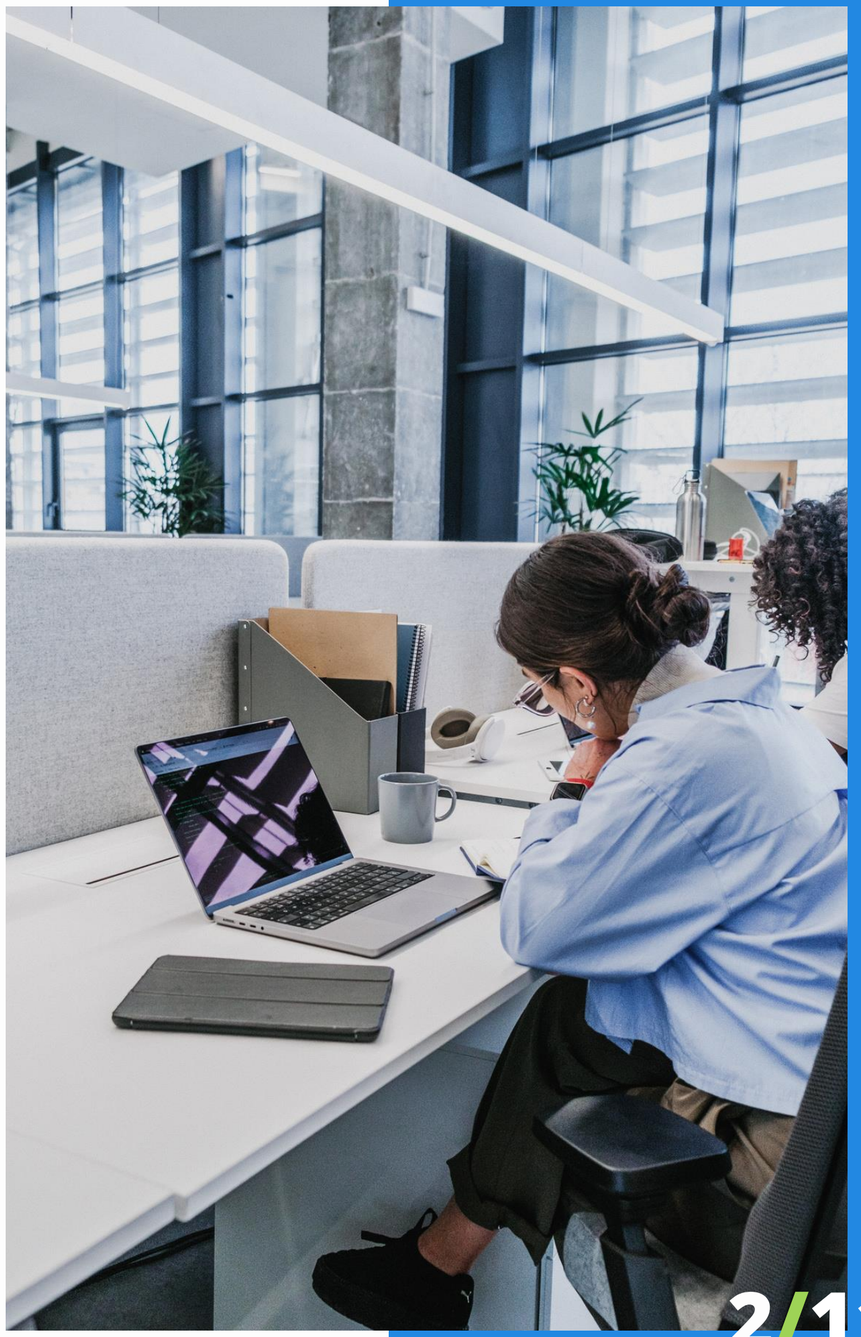
Auteurs :

Bastien Labeste

Robin Kwiatkowski

Quentin Chambelland

Sommaire



Contexte & Problématique

- Discord est une plateforme largement utilisée pour les échanges.
- Les données des messages sont souvent éparpillées et difficiles à analyser.

Comment centraliser et analyser les messages pour mieux comprendre les interactions ?

Objectifs du Projet

- Afficher la liste des utilisateurs inscrits sur un serveur Discord.
- Lister tous leurs messages avec destinataires, dates, ...
- Calculer un score de toxicité des messages.
- Utiliser une base de données MySQL et un backend PHP pour gérer les données.

Technologies Utilisées

TraceCord utilise Java pour l'application mobile, Retrofit pour les requêtes API, et PHP/MySQL pour la gestion des données sur un serveur Apache.

01 Développement Android : Java

02 Communication API : Retrofit

03 Base de données : MySQL

04 Backend : PHP

04 Serveur : Apache + MariaDB

Architecture du Projet

Une application mobile connectée à une API PHP qui communique avec une base de données MySQL.

L'application Android

Affiche la liste des utilisateurs et leurs messages via une interface intuitive.

Le bot Python

Gère les requêtes Discord et envoie les informations à la base de données.

La base de données MySQL

Stocke les utilisateurs et leurs messages de manière structurée.

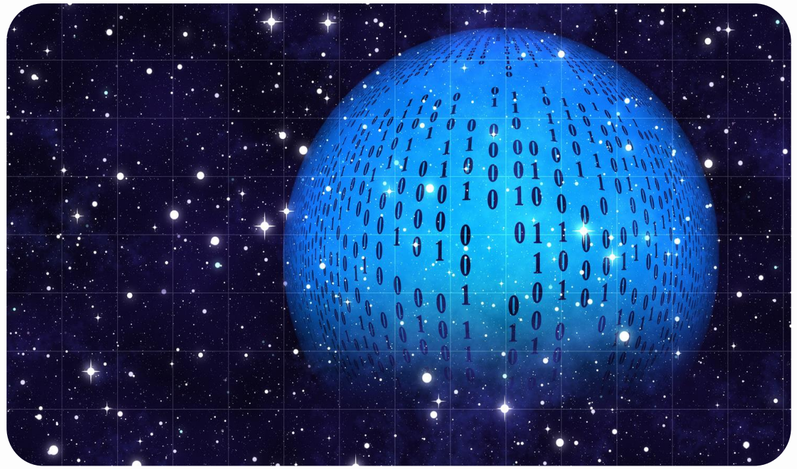
Développement de l’Application

Affichage des utilisateurs

L’application récupère la liste des utilisateurs inscrits sur Discord via une API PHP connectée à une base de données MySQL. Les doublons sont filtrés pour n’afficher chaque utilisateur qu’une seule fois.

Affichage des messages

Lorsqu’un utilisateur est sélectionné, tous ses messages sont affichés avec les détails associés : contenu, destinataire, taille du message et score de toxicité. Ces données sont extraites dynamiquement depuis la base de données via l’API.



Analyse du contenu

Chaque message est analysé pour calculer des indicateurs comme la longueur moyenne des messages, le destinataire le plus courant et le score de toxicité moyen. Cela permet d’identifier les comportements des utilisateurs et d’améliorer la modération.

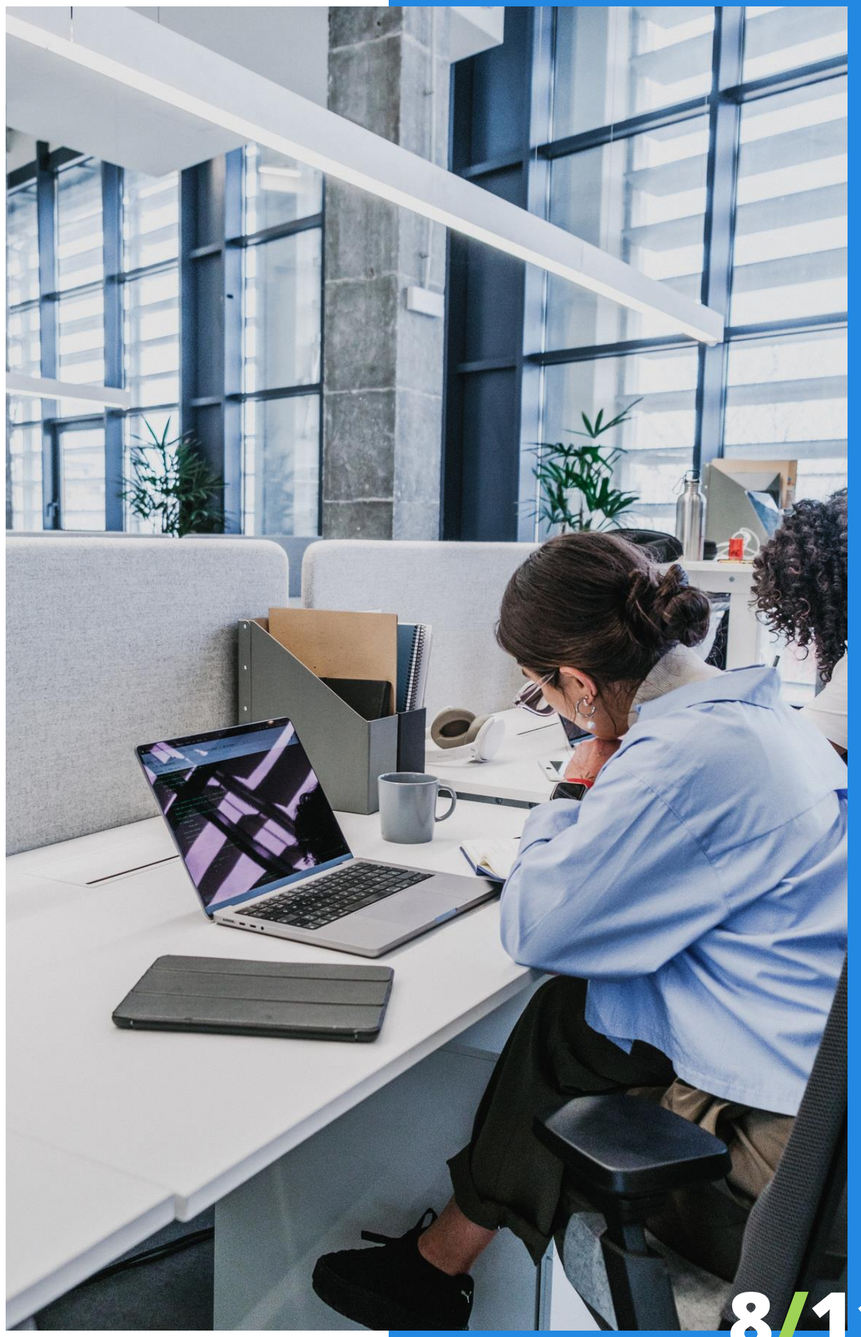
Déploiement & Installation

Serveur Apache + MySQL

Installation de PHP et API

Configuration de la base de données et du bot Python

Connexion Android → API → MySQL



Problèmes rencontrés & Solutions

❌ Problème : Connexion à la base de données impossible

✅ Solution : Modifier bind-address et configurer les accès root.

❌ Problème : Impossible d’accéder aux messages d’un utilisateur

✅ Solution : Correction de la requête SQL ou relance du serveur Apache

RACI

Tâches / Activités	Bastien Labeste	Robin Kwiatkowski	Quentin Chambelland
Définition du projet et objectifs	R	R	R
Architecture de l'application	C	C	R
Développement Backend (MySQL, Python)	R	R	A
Développement Frontend (Android studio)	R	A	R
Connexion entre l'API et l'application (PHP)	R	A	R
Analyse du contenu des messages	A	R	R
Tests et Débogage	C	R	A
Déploiement et Installation	A	R	C
Documentation (README, Installation.docx)	R	A	C
Préparation du PowerPoint	R	A	C

UNIVERSITÉ CÔTE D'AZUR

CONCLUSION

Auteurs :

Bastien Labeste

Robin Kwiatkowski

Quentin Chambelland