

# Documentation Technique AREA

*Nom* : AREA

*Début* : 19/09/2022

*Fin* : 13/11/2022

*Objectif* : L'AREA est une application web et mobile pouvant permettre à son utilisateur de choisir des réactions faites par une plateforme en fonction d'une action qui a été faite avec une autre plateforme (ex : recevoir un mail avec le nombre de like d'une vidéo dès que celle-ci est likée, action réaction entre YouTube et Gmail.). Dès qu'il y a une action venant d'une plateforme, celle-ci va engendrer la réaction d'une autre plateforme qui à son tour fera une action.

*L'utilisateur devra pouvoir* :

- ✓ 1. Se créer un compte.
- ✓ 2. Se connecter via son compte Google, Facebook ou Twitter.
- ✓ 3. Pouvoir se connecter à différentes plateformes (Discord, YouTube, Twitch, Google, Twitter, Spotify)
- ✓ 4. Pouvoir choisir des actions réactions en fonctions des plateformes connectées.
- ✓ 5. Pouvoir désactiver des actions réactions.
- ✓ 6. Pouvoir se déconnecter d'une plateforme.
- ✓ 7. Pouvoir se déconnecter de l'AREA.
- ✓ 8. Avoir accès à son compte sur l'appli web et mobile avec les mêmes paramètres enregistrés.

*Langages utilisés* :

Application web : Java Script

Application mobile : Progressive web app React PWA

*Applications utilisées pour le projet* :



Docker : lancer les applications dans les conteneurs logiciels.



Notion : Suivi du projet, des réunions de suivi ainsi que des tâches tout au long de son déroulement.



GitHub : Hébergement du projet.

ssh : [git@github.com:EpitechPromo2025/B-DEV-500-LIL-5-1-area-eliott.ayetand.git](ssh:git@github.com:EpitechPromo2025/B-DEV-500-LIL-5-1-area-eliott.ayetand.git)

https : <https://github.com/EpitechPromo2025/B-DEV-500-LIL-5-1-area-eliott.ayetand.git>



Git flow : Gestion des branches Git.

*Organisation du dépôt* « [B-DEV-500-LIL-5-1-area-eliott.ayetand](#) » :

Nous utilisons Git flow pour organiser nos différentes branches. Cela permet un développement en équipe tout en limitant les conflits. Chacun a la possibilité de travailler dans son coin pour développer des features et de les merger à n'importe quel moment dans le projet. On peut également vérifier le bon fonctionnement de notre projet étape par étape à fin d'avoir sur notre branche master, qui est notre branche principale de notation, un projet fonctionnel.

La documentation de Git flow : <https://git-flow.readthedocs.io/fr/latest/presentation.html>

## **ARCHITECTURE**

Server :

- actions/ : Toutes les fonctions de nos actions / réactions, une pour chaque
- config/ : Là ou sont gérés les connections à passport
- controllers/ : Les fonctions qui gèrent le système CRUD
- middlewares/ : Contient les fonctions middlewares
- models/ : Contrôle les modèles (représentation de la base de donnée) selon le système

CRUD

- routes/ : Toutes les routes de l'API

Client :

- actions/ ; reducers/ ; services/ : Implémente le système Redux de React
- components/ : Des composants react qui peuvent être utilisé partout dans l'application
- config/ : Gère les requêtes API
- views/ : Chacune des pages de l'application est dans ce dossier
- assets/ : Contient tous nos assets