Presentation géneral de l’algo (IA) :

* Système de point (Aucune certitude pour les sous genres a voir si un titre possede plusieurs attribut de genre):
  + + 4 = un genre liké ( +2 pour les sous genre)
  + + 6 = un genre rajouté en playlist ( + 3 pour les sous genre)
  + - 2 = un genre dislike ( -1 pour le sous genre)
* Système de genre et Sous genre
  + A voir avec l’quipe si un titre peut avoir plusieurs genres et sous genre (Tableau de comparaison à crée)
* Initialisation des preferences de l’utilisateur :
  + Etape 1 : proposer chaque genre (Aucun genre a 0)
  + Etape 2 : proposer chaque Sous genre (Aucun sous genre a 0) -> pas sure
* Système de Random (Si + de 3 genre coté positivement):
  + 30% de chance d’obtenir un genre qui possede une notation négative
  + 70% de chance d’obtenir un genre qui possede une notation possive
* Système de Random (Si 1 et 3 genre coté positivement):
  + 50% de chance d’obtenir un genre qui possede une notation négative
  + 50% de chance d’obtenir un genre qui possede une notation positive
* Système de Random (Si 0 genre coté positivement):
  + Récupere le genre ayant le score le plus haut
* Système de Random  (Si un genre atteint le score de 100)
  + 10% de chance d’obtenir un genre qui possede une notation négative
  + 20% de chance d’obtenir un genre qui possede une notation positive
  + 70% de chance d’obtenir un genre qui possede le score de 100
    - Lorsque le genre atteint le score de 100, on reprend les principes ci-dessus mais uniquement pour les sous genres du genre (Si plus de 3 sous genre)

Les caracteristiques ci-dessus sont prévisionelles pour avoir ne base intéressante, elle peuvent etre amener a évolué notament suite a la gestion de la BDD et aux tests qui seront effectués

Cas particulier a réfléchir : Cas ou tous les genres sont positif ou négatif

Enregistrement des informations en base de données (A travaillé)

Priorité du Back :

* Testé la liason back – front en local (génération de titre aléatoire avec et sans framework)
* Testé le fonctionement du back sur AWS
* Testé la BDD sur AWS
* Testé la liason Back et BDD sur AWS
* Refléchir a la BDD
* Ecrire l’algo