

# Rapport du **Projet** **Social** en équipe



# Projet!

Résumé du projet en 2 lignes

- Peut être lancé
- Récupère les données
- Ajoute des données
- Le projet consiste à un chat, on choisit le salon qu'on veut rentrer et on peut envoyer des messages et ils s'affichent





1

# Le projet

Contributions individuelles

# Contributions individuelles

- **Nom : Simon**
  - Liste salons + liste messages + page profil (non fonctionnelle), affichage principal des salons et quand on clique sur un salon des messages du salon, + bouton page profil + bouton et case pour écrire et envoyer un message + début refactor lister
- **Nom : Alexandre**
  - Profil : la vue vueProfil.java + le DAO pour Utilisateur + le modèle Utilisateur
- **Nom : Guillaume**
  - Installation de la bdd sur le serveur et création et remplissage des tables.
- **Nom : Esteban**
  - Refactor du DAO + fonction ajouter + fonction lister + test en local avec données puis liaison à la base

À remplir avec les écrans, systèmes ou fonctionnalités travaillés

**Données maître :**

salon,

**Données transactionnelles :**

message,

**Autres données :**

utilisateurs, journal

À remplir avec le nom  
des tables ou autres  
(caches, entité)

# Liste de vérifications

À répondre avec oui  
ou non

## SAUVEGARDE

Avez-vous bien placé chaque import ou export de données dans un DAO ? OUI

Avez-vous bien sauvegardé chaque SQL ou dépôt dans le git du projet ? OUI

## SÉCURITÉ

Avez-vous utilisé les requêtes préparées à chaque fois que c'est possible ? OUI

Avez-vous filtré les inputs du service de données le cas échéant ? NON





# 2

# Les optimisations

Une par participant

**PROGRAMMEUR : Esteban Escudero**

# Optimisation **Redis** pour un cache des messages salons

Création d'un serveur redis en local comportant la base de données messages avec la commande `cache.sadd` pour introduire des données et la commande `cache.smembers` pour prendre les données (non-fonctionnel)

**Lien vers cette partie dans le site github**

**<https://github.com/cegepmatane/projet-social-2020-forum/blob/main/src/donnee/MessageDAO.java>**





## **Optimisation Journal Bdd** **cachant les options**

À l'aide de cron, j'ai réaliser un script bash qui s'exécute tout les soir a 20h. Ce script utilise une procédure stockée qui compte le nombre de connection totale qu'à eu la base de données en une journée.

**Lien vers cette partie dans le site github**



## Optimisation de **taille de la table** message

J'ai mis en place un évènement CRON qui exécute la commande avec requête SQL

```
sudo -u postgres -H -- psql -d  
slouke -c "TRUNCATE TABLE  
messages" ce qui vide la table tous  
les jours à minuit
```

Lien vers cette partie dans le site github :

[https://github.com/cegepmatane/projet-social-2020-20-forum/blob/main/poc/vidertable.sh](https://github.com/cegepmatane/projet-social-2020-forum/blob/main/poc/vidertable.sh)



## **Optimisation DAO en lazy-loading**

Fonction pour initialiser une liste des données pour chaque DAO.  
Cela permet d'effectuer une seule requête lorsque l'on charge une vue.

**Lien vers cette partie dans le site github**

