

Projet : Simulation d'un groupe WhatsApp

1. Technologies et outils à utiliser

- **Communication réseau** : Sockets Java (TCP)
- **Interface graphique** : JavaFX + Scene Builder
- **Base de données** : MySQL ou PostgreSQL
- **Accès aux données** : JPA / Hibernate
- **Utilitaires** : Lombok (pour réduire le code boilerplate)
- **Gestionnaire de dépendances** : Maven ou Gradle

2. Fonctionnalités du projet

Fonctionnalités	Description détaillée	Contraintes
Nombre de membres	Le groupe ne peut pas contenir plus de 7 membres	<ul style="list-style-type: none"> - Au cas ou si le nombre de membres dépasse 7, prévoir un message d'erreur clair côté client et côté serveur. - Possibilité de quitter le groupe volontairement.
Enregistrement du pseudo	Dès qu'un nouveau membre se connecte, il doit choisir un pseudo unique	<ul style="list-style-type: none"> - Ajouter une vérification d'unicité du pseudo dans la base de données
Envoi de messages	Quand un membre envoie un message, tous les autres membres le reçoivent sauf l'expéditeur	<ul style="list-style-type: none"> - Ajouter la date et l'heure du message ainsi que le pseudo de l'expéditeur pour chaque message - son ou pop-up quand un nouveau message arrive. - Historique des messages chargé à la connexion du membre (ex: les 15 derniers messages).
Sauvegarde des messages et membres	Tous les messages et membres sont persistés dans la base de données	<ul style="list-style-type: none"> - Utiliser des entités JPA clairement définies avec des relations (ex : OneToMany membres/messages)
Filtrage des messages injurieux	Si un message contient un mot de la liste noire, l'expéditeur est banni immédiatement.	<ul style="list-style-type: none"> - Liste des mots injurieux (GENOCID, TERRORISM, ATTACK, CHELSEA, JAVA NEKHOUL) - Bannir l'utilisateur côté serveur et notifier tous les membres

3. Exemple de modèles de données (JPA)

```
@Entity
public class Message {
    @Id @GeneratedValue
    private Long id;

    @ManyToOne
    private Membre membre;

    @Column(nullable = false)
    private String contenu;

    @Column(nullable = false)
    private LocalDateTime dateEnvoi;
    // getters/setters
}
```

```
@Entity
public class Membre {
    @Id @GeneratedValue
    private Long id;

    @Column(unique = true, nullable = false)
    private String pseudo;

    private boolean banned = false;

    @OneToMany(mappedBy = "membre", cascade = CascadeType.ALL)
    private List<Message> messages = new ArrayList<>();
    // getters/setters
}
```