

CAHIER DE CHARGE : application de notifications push

1 objectif principal

Développer une application desktop java permettant à un administrateur d'envoyer des notifications par sms ou email, planifiées ou immédiates, a des groupes d'utilisateurs via une interface claire sécurisée et performante.

2 fonctionnalités principales

➤ Gestions des utilisateurs

- Création /modification /suppression/ d'utilisateurs (administrateurs).
- Import/export des utilisateurs (csv ou base SQLite)
- Classification par groupes, rôles, tags.
- Activation / désactivation d'un compte.

➤ GESTION DES MESSAGES

- Rédaction de messages type (modèles)
- Choix du canal d'envoi : SMS ou Email
- Champs dynamiques personnalisables (nom, dates , etc.)
- Aperçu avant envoi

➤ Planification des campagnes

- Envoi immédiat ou diffère
- Planification unique ou récurrente via un Schedule
- Fuseau horaire configurable

➤ Envoi segmente

- Filtrage des destinataires selon les groupes, rôles , régions et statut actif

➤ Journalisation et historique

- Enregistrement de chaque envoi : date, statut (envoyé, échoue) , message d'erreur
- Consultation par utilisateur ou par message

➤ Statistiques

- Nombres de messages envoyés (total, réussis, échoués)

- Agrégation par période ou utilisateurs
- Gestion des pièces jointes (Email uniquement) (optionnelle)
 - Ajout de fichiers ; PDF, images.

3 interfaces graphiques (UI/UX)

- Technologies recommandées : Java Fx (sceneBuilder conseille) , ou Swing
- Design : sobre, professionnel, barre latéral ou ruban
- Fenêtre : création de message : champs texte, canal(SMS/email), planification

4 technologies utilisées

- Email : jakarta Mail , Apache Commons Email
- SMS : SMSLib, Orange SMS
- Interface : JavaFX , swing
- http : Apache HttpClient
- base de données : SQLite (via JDBC)
- connexions API :API Email via SMTP sécurisé (SSL/TLS) , API SMS via appel REST POST (orange)

5 base de données (SQLite)

➤ Tables

- Users : id, nom, prénom, email, téléphone , rôle, actif
- Messages : id , contenu, type, sujet, date_creation , auteur
- Campaigns :id, message id, date programme ,récence ,statut

6 architectures, performance, sécurité

➤ Asynchrone et performance

- Utilisation de threads pour ne pas bloquer l'UI
- Découpage par paquets (100 contacts), délais programmes
- Reprise automatique en cas d'échec (retry)

➤ Scalabilite

- Externalisation via RabbitMQ ou microservices si besoin
- Passage possible a une base MySQL/PostgreSQL

➤ Sécurité

- API en HTTPS
- Stockage sécurisé (SQLCipher pour SQLite, fichiers chiffrés)
- Validation des entrées (email, numéro)
- Authentification forte à l'accès

➤ Répartition des rôles

- Chef de projet : pilotage, livrables, validations
- Développeur front-end : UI avec JavaFX/SWING
- Développeur back-end : base de données, API
- Responsable sécurité : chiffrement, validation
- Intégrateur API : configuration des envois SMS/email