

Table de matières

| | |
|--|---|
| I. Contexte du projet | 3 |
| II. Objectifs du Projet..... | 3 |
| III. Couverture fonctionnelle | 3 |
| 1. Gestion des Utilisateurs..... | 3 |
| 2. Prise de Rendez-vous :..... | 3 |
| 3. Transfert emis..... | 3 |
| 1. Performance et Scalabilité..... | 4 |
| 2. Sécurité | 4 |
| 3. Accessibilité et UX..... | 4 |
| IV. Architecture et Technologies | 5 |
| 1. Architecture Microservices | 5 |
| 2. Technologies..... | 5 |

I. Contexte du projet

Il s'agit de développer une application qui permet de digitaliser le module Transfert

Le but est de minimiser le délai de réponse de la banque par rapport à la procédure classique.

L'application est composée de deux parties :

- Partie dédiée au client de la banque qui doit initier sa demande. La demande sera envoyée au chargé clientèle de son agence pour traitement.
- Partie dédiée à l'agence : le chargé prend en charge l'étude des demandes reçues. Après la vérification des informations saisies par le client il peut soit :
 - ✓ Accepter la demande ainsi elle sera injectée dans le système .
 - ✓ Rejeter la demande.
 - ✓ Retour au client pour complément d'information.

II. Objectifs du Projet

- Concevoir et développer une application web en ligne.
- Mettre en place une architecture microservices pour assurer modularité, scalabilité et résilience.
- Assurer une expérience utilisateur fluide et interactive via Angular.
- Implémenter des services sécurisés pour la gestion des transactions.

III. Couverture fonctionnelle

1. Gestion des Utilisateurs

- ✓ Inscription / Connexion.
- ✓ Gestion des rôles (Client, Administrateur, Chargé clientèle.).

2. Prise de Rendez-vous :

- ✓ Planification de rendez-vous avec un conseiller financier.
- ✓ Gestion des créneaux disponibles.
- ✓ Notifications et rappels pour les rendez-vous programmés.
- ✓ Affichage des tendances et des analyses graphiques.

3. Transfert emis

Cette opération permet l'ouverture d'un dossier de virement émis par le client.

Le formulaire à renseigner contient les éléments suivants :

- ✓ Informations relatives au client
- ✓ Compte de commission
- ✓ Compte de règlement
- ✓ Type de virement (liste déroulante : Commercial ou Courant)
- ✓ Date d'émission du virement
- ✓ Frais associés (liste déroulante : À la charge du bénéficiaire, À notre charge, Partagés)
- ✓ Devise du virement (liste déroulante)
- ✓ Montant du virement
- ✓ Si le type de virement est Commercial :
- ✓ Numéro et date de la facture
- ✓ Motif du virement
- ✓ Options à cocher en cas d'achat avec : Négociation, Négociation à terme, ou Financement
- ✓ Coordonnées du bénéficiaire : Nom et Pays
- ✓ Banque de destination
- ✓ Si le virement est Commercial, le client doit ensuite joindre les titres concernés, ainsi que les pièces justificatives nécessaires à la demande.

Enfin :Le client enregistre la demande Un **e-mail automatique** est alors envoyé au **validateur concerné** pour traitement du dossier en attente.

Contraintes et Exigences

1. Performance et Scalabilité

- ✓ Le système doit être capable de supporter un grand nombre de transactions simultanées.

2. Sécurité

- ✓ Utilisation des bonnes pratiques en cybersécurité.

3. Accessibilité et UX

- ✓ Interface intuitive et adaptée aux différents écrans.
- ✓ Performances optimisées pour une navigation fluide.

IV. Architecture et Technologies

1. Architecture Microservices

L'architecture microservices sera utilisée afin de garantir :

- ✓ Une séparation claire des responsabilités.
- ✓ Une meilleure maintenabilité et évolutivité.
- ✓ Une résilience accrue en cas de panne.

2. Technologies

Back-end

- ✓ Spring Boot : Framework pour la gestion des microservices.
- ✓ Spring Cloud : Gestion de la communication entre services.
- ✓ Spring Security & JWT : Gestion de l'authentification et de l'autorisation.
- ✓ Base de données : PostgreSQL / Oracle.
- ✓ Kafka / RabbitMQ : Communication entre microservices.

Front-end

- ✓ Angular : Framework JavaScript pour une interface utilisateur dynamique.
- ✓ PrimeNG : Design responsive.

Devops

- ✓ Docker & Kubernetes : Déploiement et orchestration des services.
- ✓ CI/CD (GitHub Actions, Jenkins) : Automatisation des déploiements.