

Devoir à réaliser en binôme ou monôme
Réalisation d'une application de gestion d'une banque (eBank)
avec Spring BOOT et React

Date de présentation: Décembre 2025 (dernière séance)

hamaloussama@research.emi.ac.ma

I- Objectifs :

Il est demandé à réaliser une application web pour la gestion d'une banque en utilisant principalement :

⊕ Pour la couche Back-end :

- ✓ Spring Boot version 3
- ✓ Spring Security
- ✓ JJWT (pour JWT)
- ✓ Spring Data JPA
- ✓ Mysql 8.

⊕ Pour la couche Front-end :

- ✓ React JS.

II- Exigences techniques et Design Pattern :

- La communication entre la couche Front-end et la couche Back-end se fera via Rest ou GraphQL.
- Le développement de l'application doit respecter les bonnes pratiques suivantes :
 - L'IOC
 - L'AOP
 - Le DTO.
- Toutes les fonctionnalités de l'application doivent être sécurisées via JWT et Spring Security.

III- Spécifications fonctionnelles :

Il est demandé d'implémenter les cas d'utilisations suivants :

Use case n°1 : S'authentifier

L'application à mettre en place doit offrir un service d'authentification comme suit :

UC-1 : S'authentifier

- L'application doit gérer les deux profils suivants : CLIENT et AGENT_GUILCHET.
- Pour utiliser les fonctionnalités de l'application, l'utilisateur doit tout d'abord se connecter via son login et son mot de passe selon les règles RG_1, RG_2 et RG_3.
- Une fois connecté, l'application doit afficher seulement les fonctionnalités suivantes :
 - Pour le profile AGENT_GUILCHET :
 - ✓ Ajouter nouveau client
 - ✓ Nouveau compte bancaire
 - Pour le profile CLIENT :
 - ✓ Consulter Tableau de bord.
 - ✓ Nouveau virement.
- L'application offre aussi la possibilité à l'utilisateur de changer son mot de passe via le menu « changer mot de passe ».
- L'application interdit l'accès à une fonctionnalité si l'utilisateur n'est pas authentifié.
- Si l'utilisateur essaye d'accéder à une fonctionnalité dont il n'a pas le droit, l'application affiche le message suivant : « Vous n'avez pas le droit d'accéder à cette fonctionnalité. Veuillez contacter votre administrateur ».

RG_1 : Le mot de passe doit être crypté. Au niveau de la base de données, tous les mots de passe doivent être cryptés.

RG_2 : Si le login n'existe pas ou le mot de passe est erroné, l'application doit afficher « Login ou mot de passe erronés ».

RG_3 : Le délai de validité du Token (JWT) est une heure (01h). Si le Token est échu, l'application affiche le message suivant : « Session invalide, veuillez s'authentifier ».

Use case n°2 : ajouter un nouveau client

UC-2 : Ajouter un nouveau client

- L'administrateur saisit le nom, le prénom, le numéro d'identité, la date anniversaire, l'adresse mail et l'adresse postal du client et ensuite clique sur le bouton créer selon les règles RG_4, RG_5, RG_6 et RG_7.

RG_4 : Le numéro d'identité doit être unique.

RG_5 : Le nom, le prénom, la date anniversaire, l'adresse mail et l'adresse postal sont obligatoires.

RG_6 : L'adresse mail doit être unique.

RG_7 : L'application enverra un mail au client en lui communiquant son login et son mot de passe.

Use case n°3 : Nouveau compte bancaire

UC-3 : Nouveau Compte bancaire

- L'administrateur saisit le RIB et l'identité du client et clique sur le bouton créer selon les règles RG_8, RG_9 et RG_10.

RG_8 : Le numéro d'identité doit exister au niveau de la base de données.

RG_9 : Le RIB doit être un RIB valide.

RG_10 : Le compte bancaire sera créé avec le statut « Ouvert ».

Use case n°4 : Consulter le tableau bord Client

UC-4 : Consulter Tableau de bord

- L'application affiche les informations suivantes :
 - ⊕ Le numéro du RB.
 - ⊕ Le solde du compte.
- ⊕ Les dix dernières opérations bancaires. Pour chaque opération, l'application affiche : l'intitulé de l'opération (par exemple Virement en votre faveur de ...), le type de l'opération (Débit ou Crédit), la date de l'opération et le montant de l'opération.
- ⊕ Si le client dispose de plusieurs compte bancaires, l'application affiche ses comptes dans une liste déroulante et par défaut affiche les dix premières opérations du compte récemment mouvementé. L'application permet également d'afficher les informations des autres comptes en choisissant dans la liste déroulante le compte souhaité.
- ⊕ L'application offre un service de pagination pour consulter les autres opérations.
- L'application affiche la fonctionnalité « Nouveau virement ».

Use case n°5 : Effectuer un nouveau virement

UC-5 : Nouveau virement

- L'application affiche un formulaire pour saisir les données suivantes :
 - ⊕ Le numéro du RIB. Le numéro du RIB est affiché par défaut et est grisé. Au cas où le Client dispose de plusieurs comptes, l'application affiche une liste déroulante.
 - ⊕ Le montant du virement.
 - ⊕ Le RIB destinataire.
 - ⊕ Le motif.
- Le client saisit les données ci-dessus et clique sur le bouton valider selon les règles RG_11, RG_12, RG_13, RG_14 et RG_15.

RG_11 : Le compte bancaire ne doit pas être bloqué ou clôturé.

RG_12 : Le solde de compte doit être supérieur au montant du virement.

RG_13 : Le compte du client sera débité du montant du virement.

RG_14 : Le compte du client destinataire sera crédité du montant du virement.

RG_15 : L'application doit tracer les deux opérations avec leurs dates précises.

NB 1 : Pour le développement de la couche Back-end, vous pouvez vous inspirer de l'atelier réalisé en classe concernant le développement d'un SW multi connecteur dont le code source est disponible sur GITHUB : <https://github.com/abboformations/bank-service-multi-connecteur.git>.

NB 2 : Pour le développement de la couche Front-end, vous pouvez vous inspirer des ateliers réalisés en classe.

NB3 : Par rapport aux interfaces web de votre application, vous pouvez vous inspirer des applications des banques comme par exemple la Banque populaire, etc.

Bonne chance !