

# **Résumé du projet:**

Prérequis:

- avoir une database nommé : bankdb vide
- un serveur mysql lancer
- un navigateur avec une page sur l'adresse : localhost:8080
- VS CODE avec l'extension mobioos forge

## **Fonctionnalités effectuer dans le projet:**

- Status admin
- Les utilisateurs sont bien reliés à des comptes
- Les comptes et utilisateurs sont bien rajouter a la base de donnés
- La connexion via google
- La connexion via un compte local (gestion des adresse mail sans doublon y compris avec ceux des compte de google)
- Déconnexion
- Dépôt sur son compte
- Retrait avec découvert lorsque l'on est admin dans la limite de 2000€ sinon pas de découvert
- Possibilité de générer des variantes de l'application avec un large champ de features possible.
- Créations de pages html
- Créations de projet firebase
- Coordination entre le code back-end et front-end
- Création de variants de notre application Web
- Ajout d'argent dans les comptes concernés affiché sur la page html ainsi que dans la base de données.

## **Description:**

### **Partie Back-end :**

Dans un premier temps nous avons développé une application simple de services bancaires qui consiste à pouvoir créer des comptes utilisateurs et effectuer des opérations sur son compte:

- Technologie utilisé : Maven, Spring boot, Java, API Rest, Javascript, JQuery, HTML, CSS, Bootstrap, Ajax
- Génération d'un projet type web application avec spring initializr

Edition du fichier pom.xml afin d'ajouter les différentes dépendances

Pour cela dans la partie back-end nous avons codé quatre controller

- un qui gère les différentes vue donc les changement de pages
- un qui gère les différentes interaction comme le dépôt ou le retrait
- un qui gère les différentes erreurs possible.
- un qui nous permet d'arriver à la page d'accueil lors du lancement de l'application

Nous avons utilisés une base de donnée comportant 3 tables :

- Users : qui recense les différents utilisateur
- Account : qui lie chaque utilisateur avec son solde
- Connected : qui contient l'utilisateur qui est connecter

Pour chacune de ces 3 tables nous avons une classe java qui comporte les différents attributs (colonne) ainsi que leur condition (non null ou bien valeur minimum)

Nous avons également une classe UserService nous permettant d'effectuer les différentes actions demandés par les requêtes qui ont été au préalable traitées par les controller.

-Status admin : un compte possédant le statut administrateur aura plus de fonctionnalité comme une vue sur l'ensemble des comptes que recense la banque, la modification et suppression de ceux-ci ainsi que la possibilité de créer un nouveau compte. Il a également un autre avantage, qui est un découvert possible de 2000€.

-Possibilité de se connecter via son compte google ainsi qu'un compte local à l'application que l'on aura créé au préalable.

-Possibilité de déposer de l'agent ainsi que d'en retirer.

## Partie Front-end :

développement des différentes pages en html/css : Login, Création de compte, Connexion via compte local, index (menu après connexion) et users (gestion des utilisateurs pour les admin)

Ajout du javascript pour charger les données json du type de banque choisit, ainsi que pour l'intégration de requêtes ajax pour les différentes interactions

## Mobioos Forge :

Ajout de l'extension mobioos forge sur l'IDE VSCode

Création d'un features model

Ajout des différentes dépendances ainsi que des caractéristiques nécessaires pour chaque features.

Ajout de marker en associant les différentes features avec leur lignes de codes respectivement.

test de génération d'application en sélectionnant différentes combinaisons de features.

ICON: Nous avons réussi via les features a changer les icônes correspondant à la banque qui peut être maintenant différentes de la banque postale.

DÉCOUVERT: Nous avons également pu ajouter la possibilité d'avoir un découvert sans limite pour chaque utilisateur.

ADMIN: l'option n'est plus présente ce qui enlève tout avantages au différents utilisateurs. champ admin n'est plus visible lors de la création de compte.

CONVERSION: Possibilité de retirer le convertisseur de devise

CONSORTIUM: Permet de choisir quel carte bancaire sont acceptés

LOGIN: Permet de choisir le type de connexion possible, une condition est ici appliquée par le fait que l'on doit en choisir au minimum une.

THÈME: Permet de choisir le style, les couleurs des textes et boutons changent en fonction de la banque choisie.

DATA: Permet de choisir le texte et les informations concernant la banque choisi

## Problème rencontré :

Nous avons eu quelques soucis notamment avec la déconnexion via un compte google, nous étions bien renvoyer vers la page de connexion et la table connected était bien vidée cependant nous étions aussitôt reconnecter avec notre compte google.

Nous avons amélioré notre application web en mettant une optimisation quant à la déconnexion lorsque nous nous identifions avec l'application google qui permet de se déconnecter sans rencontrer de bug.

Nous avons également ajouté la fonctionnalités permettant d'avoir le droit à des découvert plus ou moins élevé quant à la situation du client (admin). Cela nous permet de retirer de l'argent même si les fonds sont à 0.

Pour finir, nous avons tenté d'intégrer le service firebase à notre projet (afin de pouvoir nous authentifier avec github ainsi que microsoft) mais nous n'avons pas réussi. Ce dernier provoquait des erreurs et cette fonctionnalité n'était malheureusement pas en adéquation avec le reste de notre code. Nous avons donc fourni un projet supplémentaire que vous trouverez dans notre dossier github mais ne faisant que la fonctionnalité de connexion avec github (les autres fonctionnalités ne sont pas disponibles dans cette autres application web)