

Rapport de fin de formation

projet: Gestibanque

cursus POEC Java

session Décembre2023/mars2024

sous la supervision de Mohamed Amine
Mezghich

Présenté par la teamcat :

Fatima Benarab

Andrei Zheksimbaev

Dangning Hu

<https://github.com/acdmft/teamcat-angular>

<https://bitbucket.org/kxxdhdn/gestibanque-rest/src/dev/>

Table des matières

I.Introduction

I.A.présentation du projet

I.B.les technologies utilisées

II.Conception

III.développement

IV.le déploiement

V.Conclusion

I.Introduction

I.A.présentation du projet

GestiBank est une banque en ligne offrant aux clients la possibilité d'accéder à leurs comptes et de consulter des informations générales sur les produits bancaires via des appareils connectés tels qu'un ordinateur personnel. Les clients peuvent demander la création de compte et, une fois créé, se connecter à leur espace personnel pour gérer leurs comptes, consulter l'historique des opérations et effectuer des virements.

GestiBank dispose d'espaces variés selon le rôle de la personne connectée (ADMIN, AGENT, CLIENT)

I.B.les technologies utilisées :

Durant la réalisation de ce projet ,nous avons utilisé un large panel de technologies :

Pour la conception :



StarUml

pour le développement backend :



Java

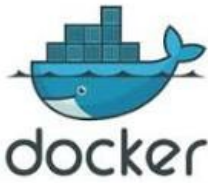


pour le développement front end :



Angular

Pour le déploiement :



Docker



Jenkins

II. Conception

Le système GestiBank est composé d'une entité utilisateurs appelée "User", subdivisée en trois sous-entités, chacune ayant des besoins spécifiques :

Administrateur (Admin):

Il a le pouvoir de créer, gérer (modifier les caractéristiques ou supprimer) des conseillers. L'administrateur peut affecter de nouveaux clients aux conseillers. Il est en charge de recevoir et répondre aux messages des internautes via le formulaire sur la page d'accueil du site web.

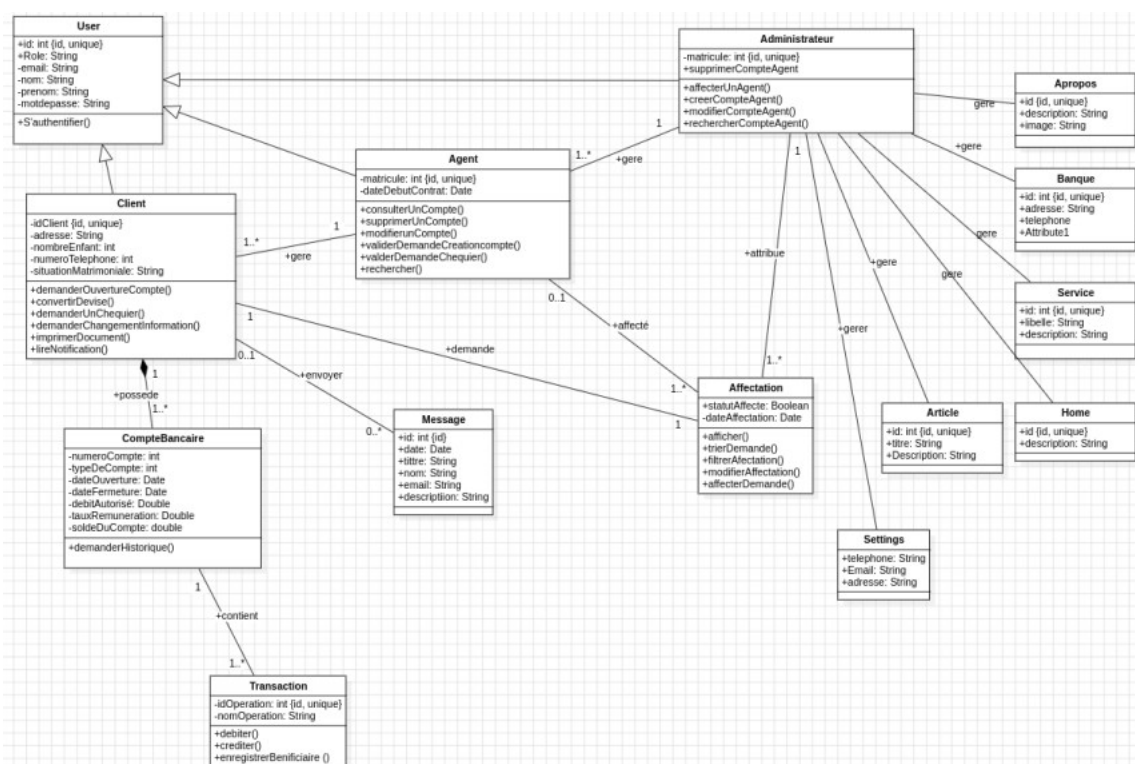
Conseiller (Agent):

Le conseiller peut valider ou refuser les demandes d'ouverture de compte qui lui sont assignées par l'administrateur. Il a le pouvoir de valider les demandes de chèques de ses clients. Le conseiller peut consulter une liste de comptes et de demandes équipée d'une fonction de recherche. Il a la capacité de gérer ces listes en effectuant des modifications de propriétés et des suppressions.

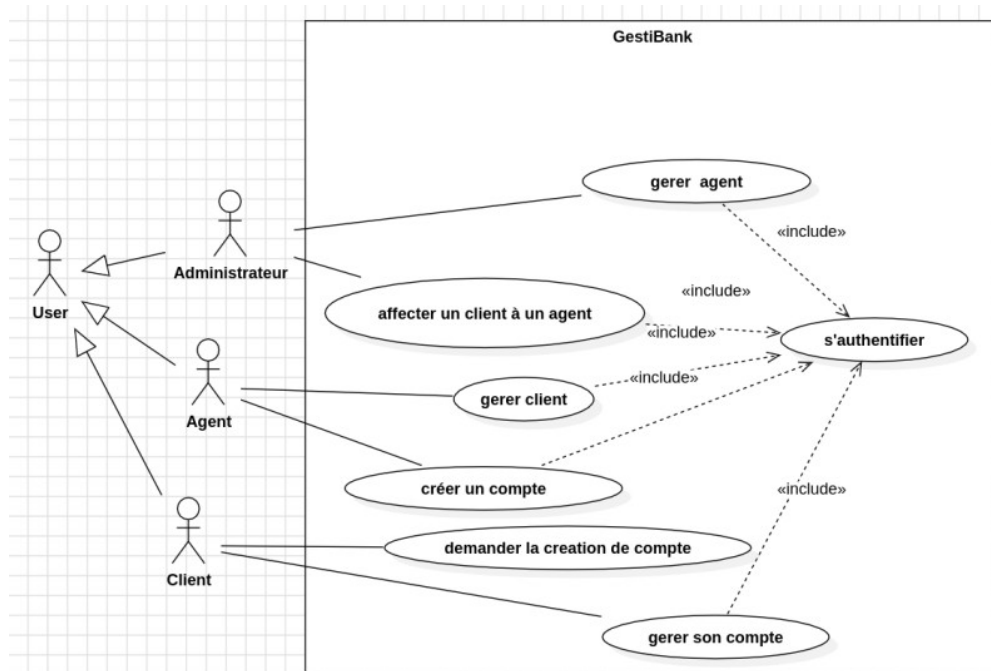
Client :

Le client peut consulter ses comptes en ligne. Il a la possibilité d'effectuer des demandes d'ouverture de nouveaux comptes (épargne, courant avec ou sans découvert). Les clients peuvent effectuer des virements vers des comptes tiers. Ils ont la faculté de faire des demandes de chèques à leur conseiller.

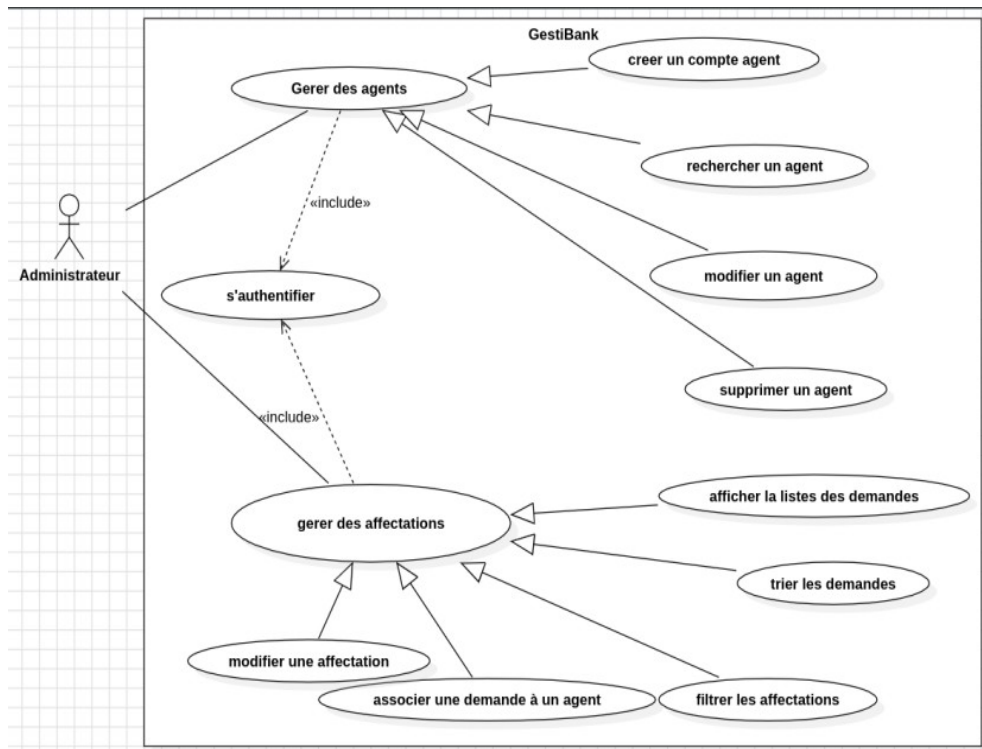
A. Diagramme de classes



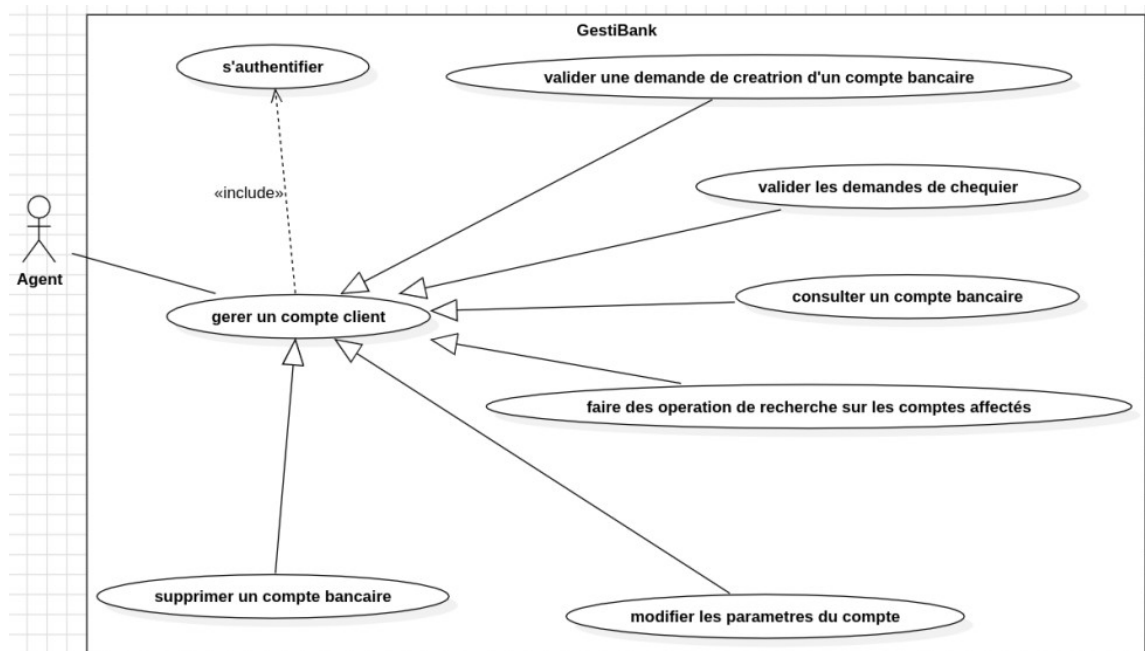
B. Diagramme Use Case Global



B. Diagramme diagramme Use Case admin



C. diagramme use case Agent



III.développement

Notre application Gestibank a connu un développement robuste, avec une attention particulière portée à la structuration du backend utilisant Spring Boot. La méthodologie adoptée pour cette tâche cruciale a grandement contribué à la clarté et à la maintenabilité du code

1. Utilisation de Spring Boot :

Le choix de Spring Boot s'est avéré être une décision judicieuse pour le développement backend de Gestibank. Spring Boot offre une architecture légère et modulaire qui facilite le développement d'applications Java avec une configuration minimale.

2. Division des Tâches :

Une approche fondamentale a été d'organiser le développement en fonction des différentes classes à créer. Cela a permis de diviser efficacement les responsabilités au sein de l'équipe de développement, favorisant ainsi une collaboration fluide.

3. Structuration du Projet

Le projet a été structuré de manière à refléter une organisation logique et ordonnée. L'utilisation de packages a été privilégiée pour classer et regrouper les différentes parties du code source. Cette structure facilite la navigation et la maintenance du projet.

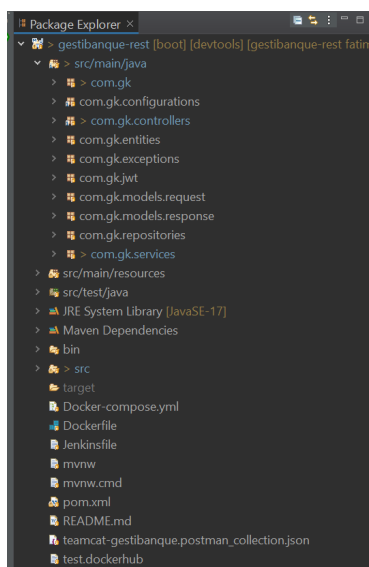
4. Collaboration Efficace :

Cette approche a favorisé une collaboration efficace entre les membres de l'équipe, chaque développeur étant responsable d'une fonctionnalité spécifique, garantissant une cohérence dans l'ensemble du projet.

5. Avantages de la Structuration :

Maintenabilité : La structuration par packages facilite la maintenance et l'extension du code.

Compréhension Facile : La division des tâches et la structure claire rendent le code facile à comprendre, même pour les nouveaux développeurs intégrant le projet.



Nous avons aussi mis en place une couche sécurité avec JWT et Spring security, ainsi nous avons créé des rôles : ADMIN,CLIENT,USER,AGENT et selon son rôle un utilisateur va être redirigé vers un dashboard spécifique.

Nous avons mis en place un service Token intercepteur sur l'application Angular qui va permettre d'intercepter les requêtes envoyées par le navigateur, rajouter un token si il y'en a et puis envoyer cette requête vers notre api en backend.

```
import { Injectable } from '@angular/core';
import { HttpInterceptor, HttpHandler, HttpRequest, HttpResponse } from '@angular/common/http';
import { catchError } from 'rxjs/operators';
import { throwError } from 'rxjs';

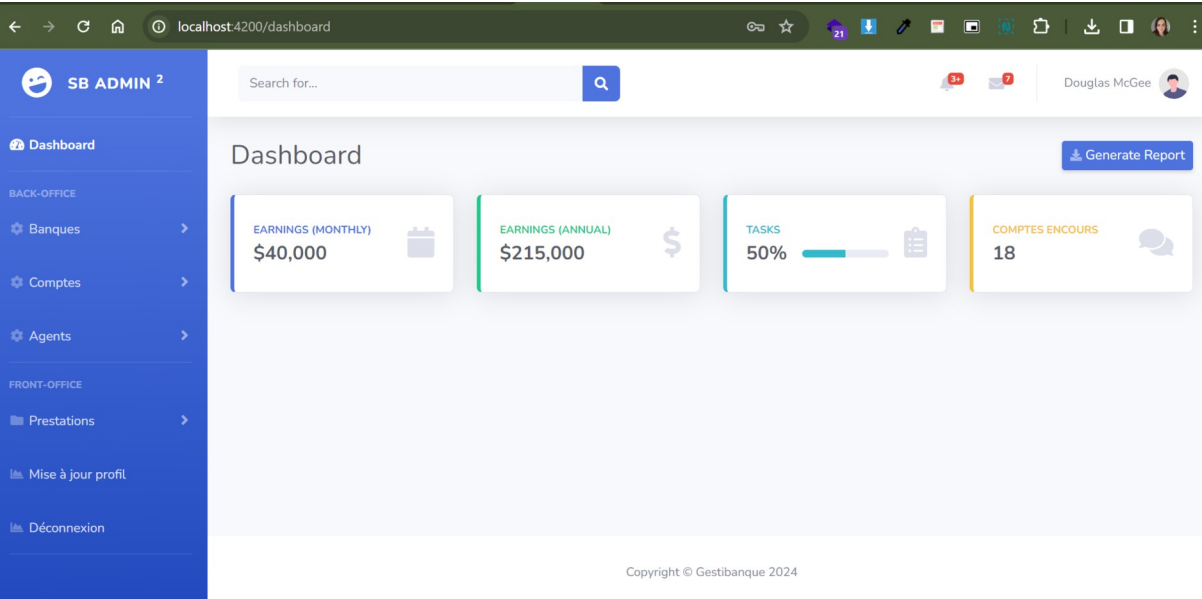
@Injectable({
  providedIn: 'root'
})
export class TokenInterceptorService implements HttpInterceptor {

  constructor() { }

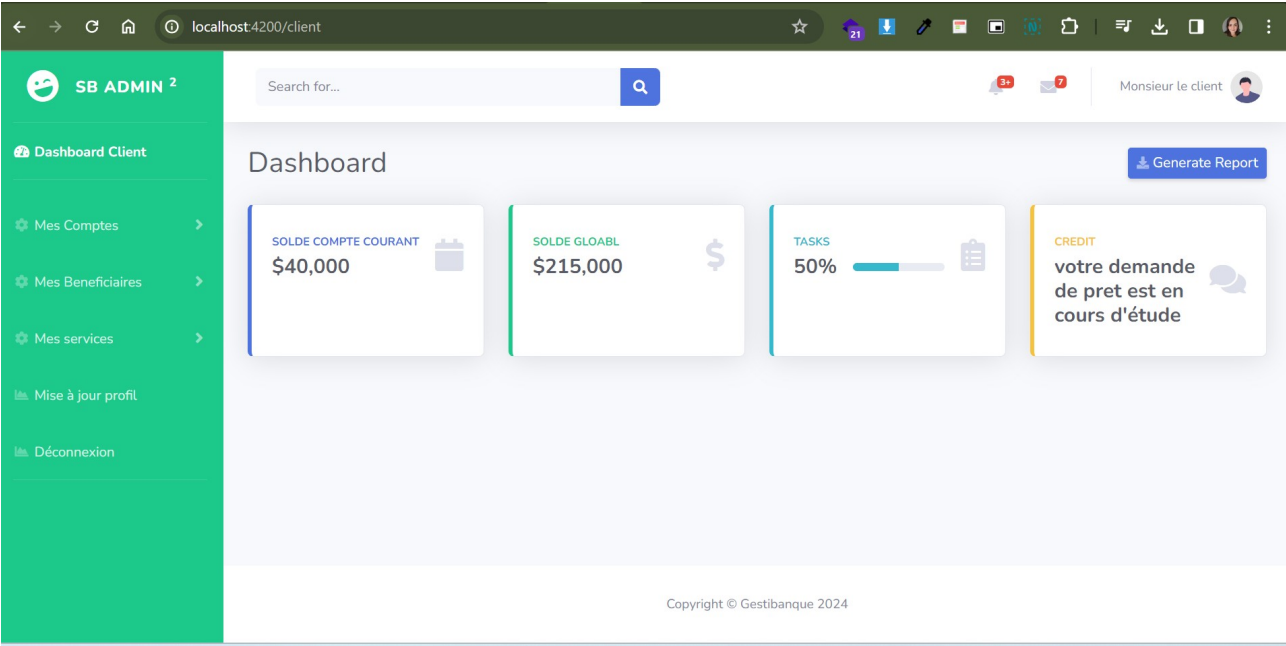
  intercept(req: HttpRequest<any>, next: HttpHandler) {
    const token: any = sessionStorage.getItem('jwtToken');
    req = req.clone({ headers: req.headers.set('Authorization', 'Bearer ' + token) });
    req = req.clone({ headers: req.headers.set('Content-Type', 'application/json') });
    req = req.clone({ headers: req.headers.set('Accept', 'application/json') });
```

```
    return next.handle(req)
      .pipe(
        catchError((error: HttpResponse) => {
          if (error && error.status === 401) {
            console.log("ERROR 401 UNAUTHORIZED")
          }
          const err = error.error.message || error.statusText;
          return throwError(error);
        })
      );
  }
}
```


le dashboard ADMIN



le dashboard client



le dashboard Agent

SB ADMIN 2

Dashboard agent

BACK OFFICE

Banques

Comptes

Clients

FRONT OFFICE

Prestations

Mise à jour profil

Déconnexion

Search for...

Dashboard agent

EARNINGS (MONTHLY)
\$40,000

EARNINGS (ANNUAL)
\$215,000

TASKS
50%

COMPTES ENCOURS
18

Liste des clients

ID	Username	Prenom	Nom	Email	Telephone	Adresse	Matricule	Date de creation	Photo		
112	MarkRotko	Rotko	Mark	rotko@gmail.com	1234567	Paris	abcd1234	12/02/2024	mark-rotko-avatar.png	Modifier	Supprimer
112	AndreDerrain	Derrain	Andre	derrain@gmail.com	1234567	Paris	abcd1234	12/02/2024	derrain-andre-avatar.png	Modifier	Supprimer
112	TashioSaeki	Tashio	Saeki	saeki@gmail.com	1234567	Paris	abcd1234	12/02/2024	tashio-saeki-avatar.png	Modifier	Supprimer
ID	Username	Prenom	Nom	Email	Telephone	Adresse	Matricule	Date de creation	Photo		

Ajouter un client

la page home

GESTIBANK

A propos de nousPrestationsActualitésEquipeContactConnexionCréer compte

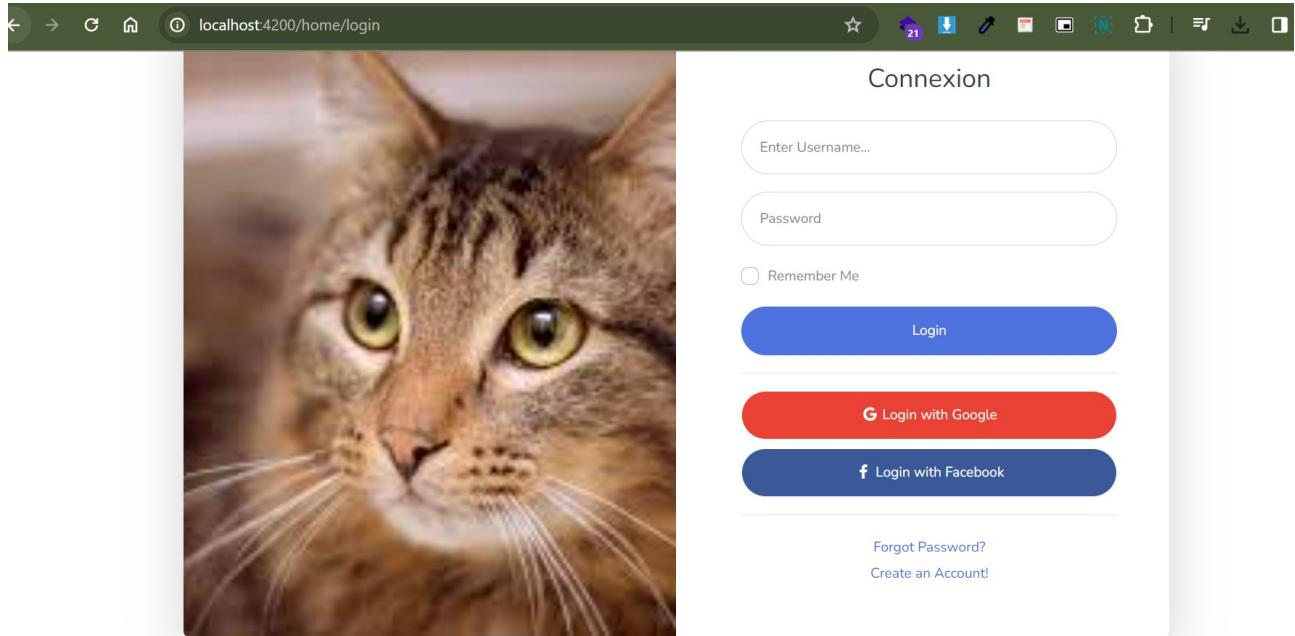
BANKING SOLUTIONS

Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae minima delectus accusamus accusantium deleniti libero excepturi porro illo.

Get In Touch

la page login

En conclusion, le développement de l'application Gestibank avec Spring Boot, basé sur une division efficace



des tâches et une structuration intelligente, a permis la création d'un backend robuste et évolutif.

Concernant la partie front end : nous avons choisis une structure modulaire pour notre projet Angular avec une récupération dynamiques des données grâce aux api rest développées en backend.

Nos avons développer des CRUD pour chaque entité ,par exemples les banques , les agents, les clients...

IV.le déploiement

Pour le déploiement de notre application , on a utilisé docker pour créer une image , puis nous avons initié une pipeline jenkins , qui se charge à chaque commit qu'on fait sur notre repository bitbucket

Stage View



V.Conclusion

La formation que notre équipe a suivie sous la guidance experte d'Amine a été une étape cruciale dans le développement de nos compétences et de notre cohésion en tant qu'équipe de projet. Nous tenons à exprimer notre profonde gratitude pour le travail exceptionnel de coaching fourni par Amine, qui a grandement contribué à la réussite de cette expérience

Grâce à la formation, nous avons acquis des connaissances approfondies et des compétences essentielles qui ont renforcé notre capacité à aborder les défis techniques avec confiance. Amine a su partager son expertise de manière claire et accessible, ce qui a facilité notre compréhension des concepts complexes. Surmontant les difficultés rencontrées tout au long du processus de développement de l'application Gestibank, notre équipe a démontré une résilience impressionnante. Les erreurs ont été perçues comme des opportunités d'apprentissage, et chaque défi a été abordé avec détermination et esprit d'équipe. Cette mentalité positive a été un pilier essentiel de notre progression.

Alors que nous clôturons cette phase de formation, nous sommes enthousiastes à l'idée de relever le prochain défi. Le développement de l'application Gestibank ne s'arrête pas ici - au contraire, le challenge continue et notre engagement à améliorer constamment notre application demeure inébranlable.

Nous sommes convaincus que les compétences acquises et les leçons apprises lors de cette formation continueront à impacter positivement notre travail quotidien. En tant qu'équipe, nous sommes prêts à appliquer ces nouvelles connaissances pour atteindre de nouveaux sommets dans le développement de logiciels et à contribuer de manière significative à la réussite continue de notre projet.

Merci à Amine et à toute l'équipe pour cette expérience enrichissante.