



DOSSIER FINAL

CQP Développeur Nouvelles Technologies



03 OCTOBRE 2023
OCTO TECHNOLOGY
34, avenue de l'opéra 75001 Paris



LIBRAIRIE
SEPT ARCHE

Fatima KAAOUAN

Consultante Web

SOMMAIRE

TABLE DES MATIERES

INTRODUCTION.....	4
CHAPITRE 1 : RAPPEL DU BESOIN.....	5
1. Équipe.....	6
2. Rappel du cahier des charges.....	6
Modèle de donnée.....	6
Fonctionnalités.....	7
Contraintes	8
CHAPITRE 2 : MÉTHOLOGIE ET OUTILS UTILISÉS.....	9
1. Méthode de conduite de projet.....	9
2. Planification du projet.....	10
3. Outils de versionning.....	11
4. Analyse des risques.....	13
CHAPITRE 3 : MODÉLISATION	14
1. Diagramme cas utilisations complets.....	14
2. Diagramme de classe.....	15
3. Diagramme d'activité	16
4. Modèle de donnée relationnel.....	17
5. Maquettes	19
Page d'accueil	19
Page panier	20
La page authentification	21
Présentation de la charte de nommage	22
CHAPITRE 4 : ARCHITECTURE TECHNIQUE	23
1. Choix de l'architecture.....	23
Côté front.....	23
Côté back.....	23
2. Choix des outils.....	28

3. Tests.....	29
Test unitaire	29
Test API	29
Test d'intégration	29
CHAPITRE 5 : INSTALLATION, MAINTENANCE ET VEILLE.....	30
1. Manuel d'installation et d'exploitation.....	30
Installation.....	30
2. Documentation technique	32
Outil JAVADOC	32
Gestion de processus des incidents selon ITIL.....	32
Clean Code.....	34
3. Maintenabilité.....	34
4. Déploiement.....	34
5. Veille technologique	35
Au niveau des langages	35
CONCLUSION	37
ANNEXE	38

INTRODUCTION

Dans le cadre de l'alternance chez Octo Technology, je vais présenter la synthèse du projet. Je vais détailler les différents cycles de vie du projet, la conduite du projet et les outils utilisés.

Dans le cadre de son développement, la librairie Sept Arche spécialisée dans le domaine du cinéma, elle dispose d'un magasin, souhaite développer un site web dans le but d'être plus visible, ce site sera, dans un premier temps, déployé uniquement en France (DOM-TOM inclus).

La première version concernera la vente de livres uniquement, qui est son cœur de métier.

En tant que développeuse web dans une ESN, je suis chargée de construire un site Web avec les langages JAVA et JavaScript, solution technique adaptée à la problématique.

CHAPITRE 1 : RAPPEL DU BESOIN

À la suite d'une étude marketing, la PME a fait appel à notre ESN pour construire son application d'une manière durable et évolutive.

Elle dispose d'un système d'information minimal, une base de données comprenant l'inventaire des produits à vendre et un logiciel de comptabilité.

À la suite de son besoin exprimé, plusieurs solutions ont été envisagées :

- Un site en CMS a été envisagée car c'est plus rapide à construire et à déployer à très court terme mais peu évolutif.
- Site de e-commerce fournit par des sociétés fournissant des packages clés en main, développés en PHP et avec une Base de données MySQL.
- JavaScript et Java sont des langages parfaits pour l'intégration avec d'autres applications et permettent de construire des sites web et applications mobiles.
- DotNet offre des possibilités similaires.

La solution qui a été retenue :

- Construire un site utilisant les dernières technologies à base de webservices car c'est un choix pérenne pour l'entreprise et adapté en cas de fort trafic avec les langages Java et JavaScript.

Si le client le souhaite, le site pourra évoluer sur plusieurs versions :

V1 : vente de livres uniquement.

V2 : Vente d'autres articles (cd, dvd...).

V3 : création de réservations de places de cinéma, festivals, des jeux/quizz ; création d'un système de notations et d'avis, actualités (sur les sorties de films).

V4 : Si le site fonctionne bien un partenariat pourra s'envisager avec d'autres sites.

1. ÉQUIPE

Pour cette application, une personne est dédiée à ce projet full stack en accord avec le client.

2. RAPPEL DU CAHIER DES CHARGES

MODELE DE DONNEE

Article et Livre :

- Tout article est identifié par une référence, un titre, un résumé, un prix.
- Un livre à un numéro ISBN-13, un titre, un ou plusieurs auteurs, u. éditeur, un genre, une image, un format. Un stock (nombre d'exemplaires en stock, un stock de 0 signifie pas de disponibilité), il n'y a pas de stock dans le format électronique.

Des informations additionnelles comme le nombre de pages ont été ajoutés, l'œuvre n'est pas géré, les livres sont stockés en base de données.

Compte client :

- Un compte client est défini par son nom, prénom, email pour l'identification, mot de passe, une adresse de facturation qui sera celle de la livraison pour la V1.

Panier :

- Un panier contiendra une liste d'articles à commander avec la quantité voulue pour chacun des articles et sera lié à un compte client.
- Pour ajouter des articles au panier, il n'est pas obligatoire d'être connecté, le panier ne sera pas persistant. Sa durée de conservation est liée à la session client, il ne sera pas stocké dans le Storage (dans le navigateur).

Commande :

- Chaque commande possède un numéro de commande, une date, pour chaque article : la quantité voulue, le prix HT et TTC
- Les informations client et l'adresse de livraison, le prix total HT et TTC.
- Les commandes sont sauvegardées dans la base.

Facture :

- Chaque commande validée engendre une facture avec les informations de celle-ci et un numéro de facture et est sauvegardée dans la base.

FONCTIONNALITES

- La page d'accueil présente l'affichage d'une sélection de livres et permettre d'effectuer une recherche de livres, un champ de recherche permettra la recherche d'un livre en fonction de son titre et de son auteur.
- Les caractéristiques suivantes seront affichées :
Titre, auteur, prix, résumé et éditeur.

- Il est possible d'ajouter un ou plusieurs livres dans le panier.
- Depuis la page panier, il est possible de modifier la quantité, de supprimer la ligne du panier.
- La validation entraîne une création de compte client ou la connexion du compte client si l'utilisateur ne l'a pas fait auparavant.
- Une fois connecté, la création de la commande est lancée.
- Dans la V1 :

Si un livre n'est pas disponible en stock, il ne peut pas être commandé.

Si la quantité demandée est supérieure à la quantité en stock, il n'est pas possible de passer la commande, un message d'erreur est affiché.

Le paiement de la commande est simulé, une fois la commande validée, une facture est générée et envoyée par mail au client.

Une page de création de compte client est disponible sur le site.

CONTRAINTES

- L'architecture a été pensée pour les évolutions futures (architectures n-tier), le modèle de données est conçu pour intégrer les futures versions.
- L'application est Single Page Application (REACT) l'interface est conçue principalement sur la page d'accueil. Une barre de menu propose d'accès au panier et au compte utilisateur. Le nombre de livres insérés dans le panier est affiché. L'application est responsive. La page est plus rapide en chargement.

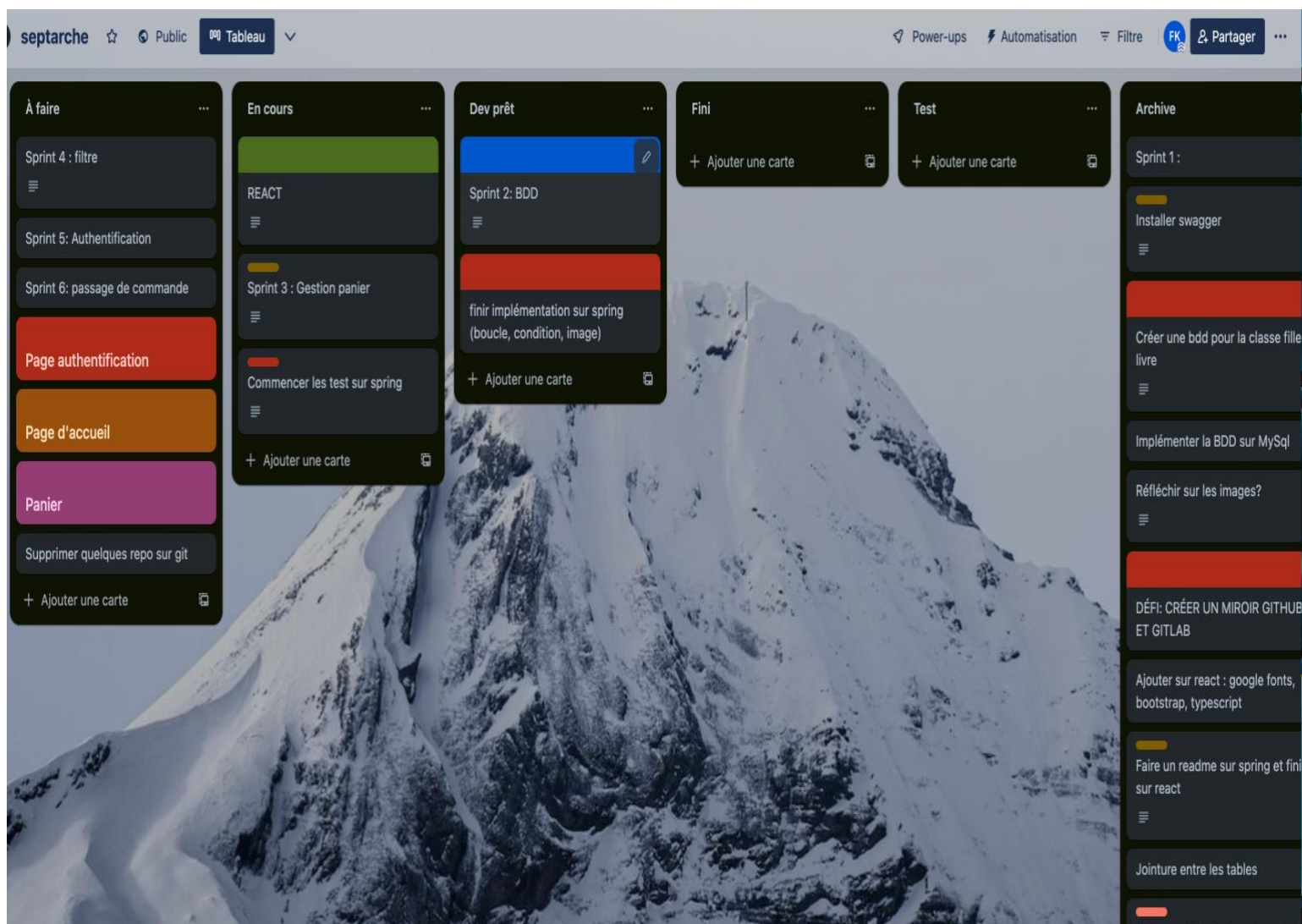
CHAPITRE 2 : MÉTHODOLOGIE ET OUTILS UTILISÉS

1. METHODE DE CONDUITE DE PROJET

Tout au long de ce projet, j'ai appliqué l'agilité car travailler en mode agile consiste à organiser les tâches selon des cycles de livraison courts appelés sprint, afin d'améliorer la qualité du produit tout en réduisant les délais et les imprévus.

Le mode agile permet également de s'adapter aux évolutions du besoin du client sans remettre en cause l'intégralité du projet.

J'ai travaillé avec l'outil Trello, où je faisais des tickets en fonction des priorités liées à l'application.



2. PLANIFICATION DU PROJET

Avec le client, un planning a été mis en place pour mettre en place des deadlines concernant les différentes étapes du projet.

La première étape : étude préalable du besoin à la suite du cahier des charges, ensuite la conception et modélisation ont été mis en place, l'implémentation avec des sprints réguliers, et le choix du déploiement.

Voici le détail du déroulé avec l'outil Excel :

A		B		C D E F G H I J K L M N O P Q R S T U V W X Y Z AA AB AC AD AE AF AG																															
Rétro Planning																																			
Projet : Application SetpArche																																			
Date de début : 02/01/2023																																			
Jour J :																																			
Date de fin : 29/09/2023																																			
Tâche à accomplir	État	02 Janvier - 25 Septembre 2023																																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31			
Analyse du besoin	Terminée																																		
Modélisation de l'application	Terminée																																		
Implémentation																																			
Réalisation de l'applicatiopn	Terminée																																		
Réalisation de la documentation technique	Terminée																																		
Déploiement de l'application	En cours																																		

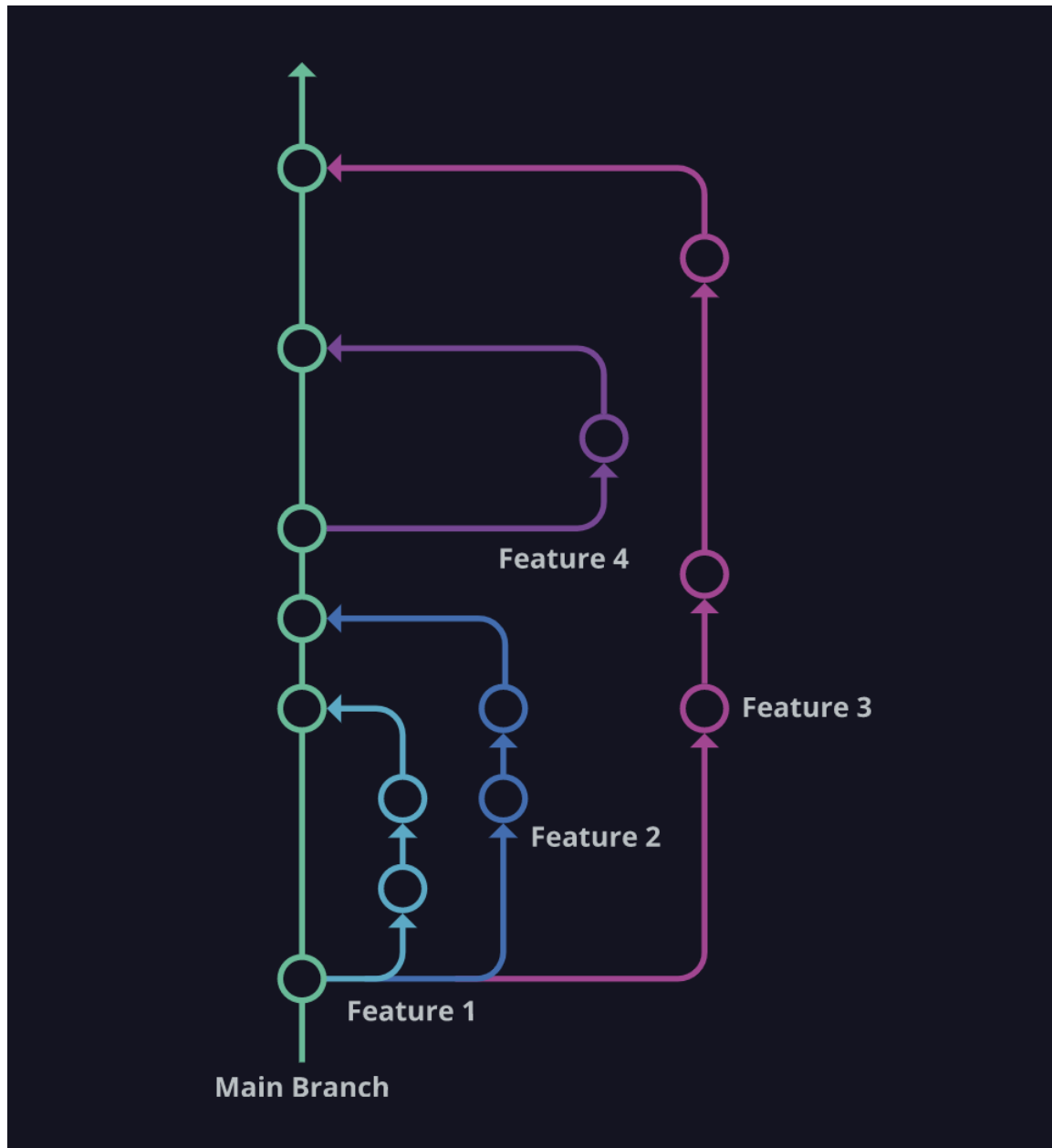
3. OUTILS DE VERSIONNING

Un logiciel de versioning est utilisé afin de garder la trace des différentes versions de l'application et de pouvoir faire des mises à jour sur des branches différentes qui seront ensuite fusionnées. L'outil de versionning utilisé est Git. J'ai utilisé la plateforme GitHub (référentiel remote) pour stocker les sources.

J'ai utilisé les commandes suivantes :

- git init pour créer la branche main, qui est la branche principale,
- git checkout -b : pour créer des features
- git add : pour prendre en compte les fonctionnalités liées à un ticket Trello
- git commit : pour valider et nommer ces fonctionnalités
- git push : pour envoyer le travail sur GitHub
- git merge : une fois la comparaison effectuée, un merge était fait dans la branche main

Voici le work flow qui représente les branches mis en place :



4. ANALYSE DES RISQUES

Tout au long du projet, les risques ont été analysés pour respecter les délais de livraison du projet.

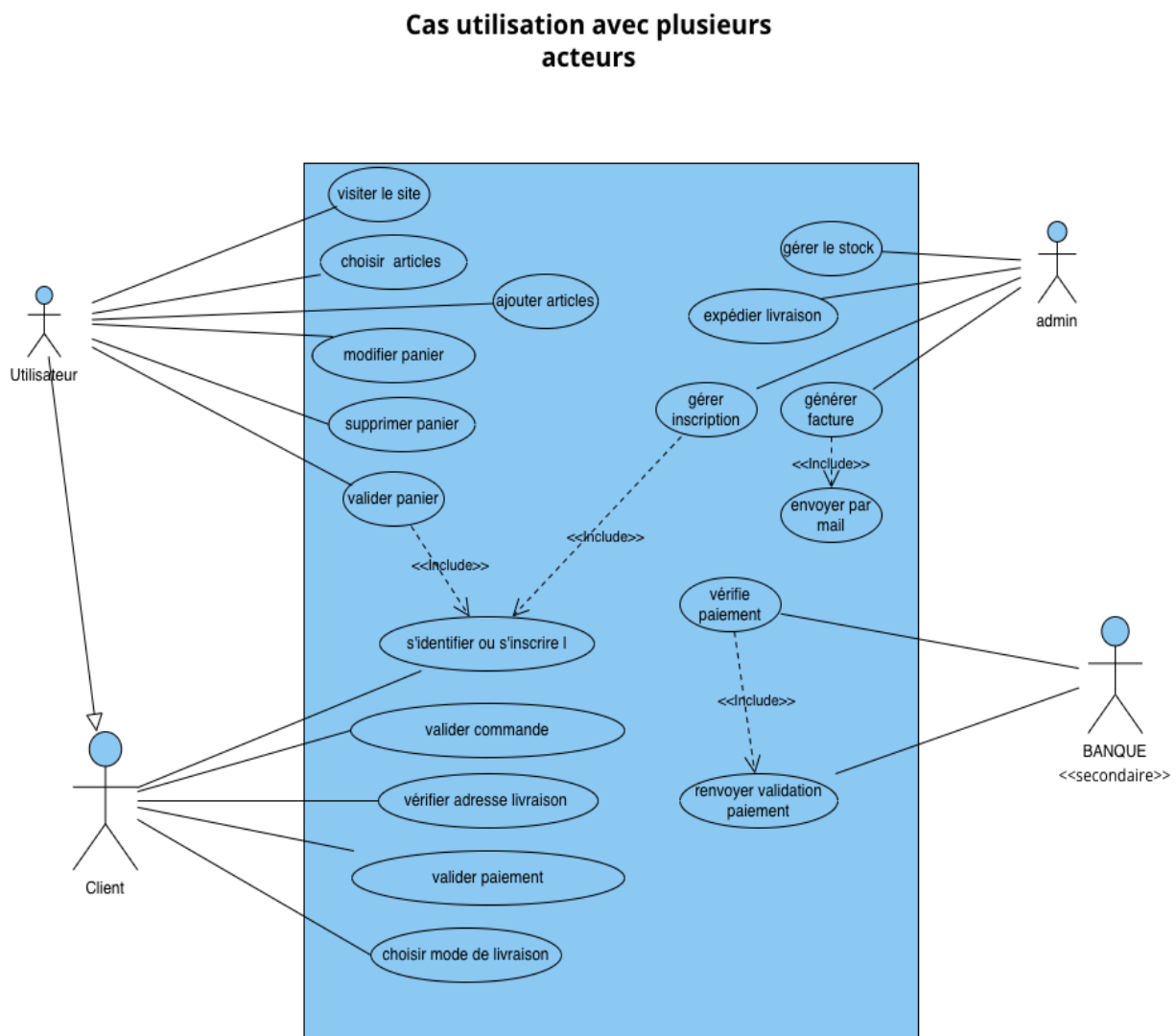
Risques prévus	Imprévus
Projet confié à seul développeur	Congé maladie
Congé payé	Pc en panne
Période de formation entreprise	Changement de pc
Travail sur d'autres missions	

CHAPITRE 3 : MODÉLISATION

J'ai utilisé la méthodologie UML pour la conception du projet.

1. DIAGRAMME CAS UTILISATIONS COMPLETS

Les diagrammes de cas d'utilisation modélisent le comportement d'un système et permettent de capturer les exigences du système. Ces diagrammes identifient les interactions entre le système et ses acteurs. Les cas d'utilisation et les acteurs dans les diagrammes de cas d'utilisation décrivent ce que le système fait et comment les acteurs l'utilisent, mais ne montrent pas comment le système fonctionne en interne.

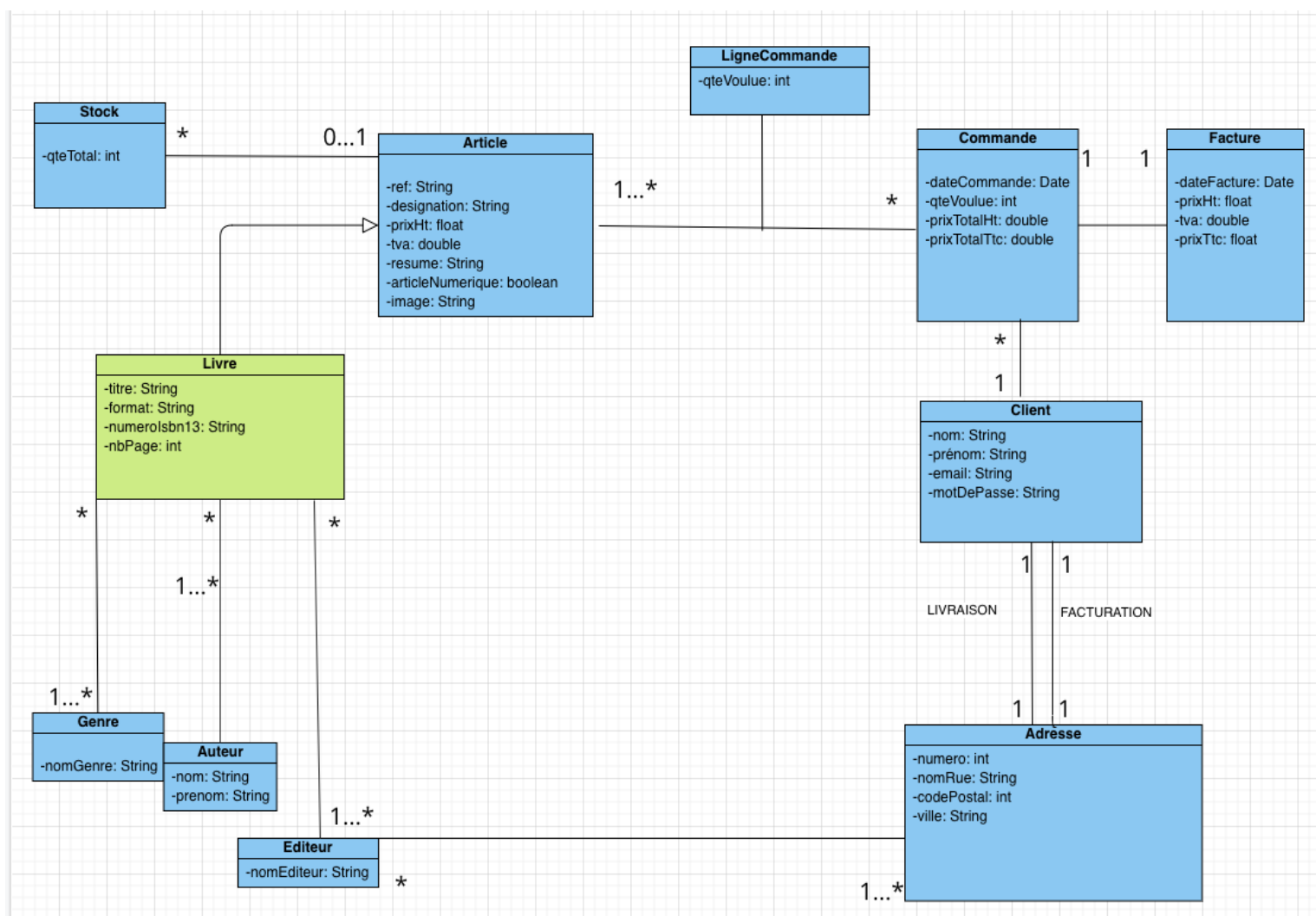


2. DIAGRAMME DE CLASSE

Dans le cadre de la mise en place du projet, j'ai réalisé un diagramme de classe, il décrit la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets. Une classe représente un objet ou un ensemble d'objets qui partagent une structure et un comportement communs.

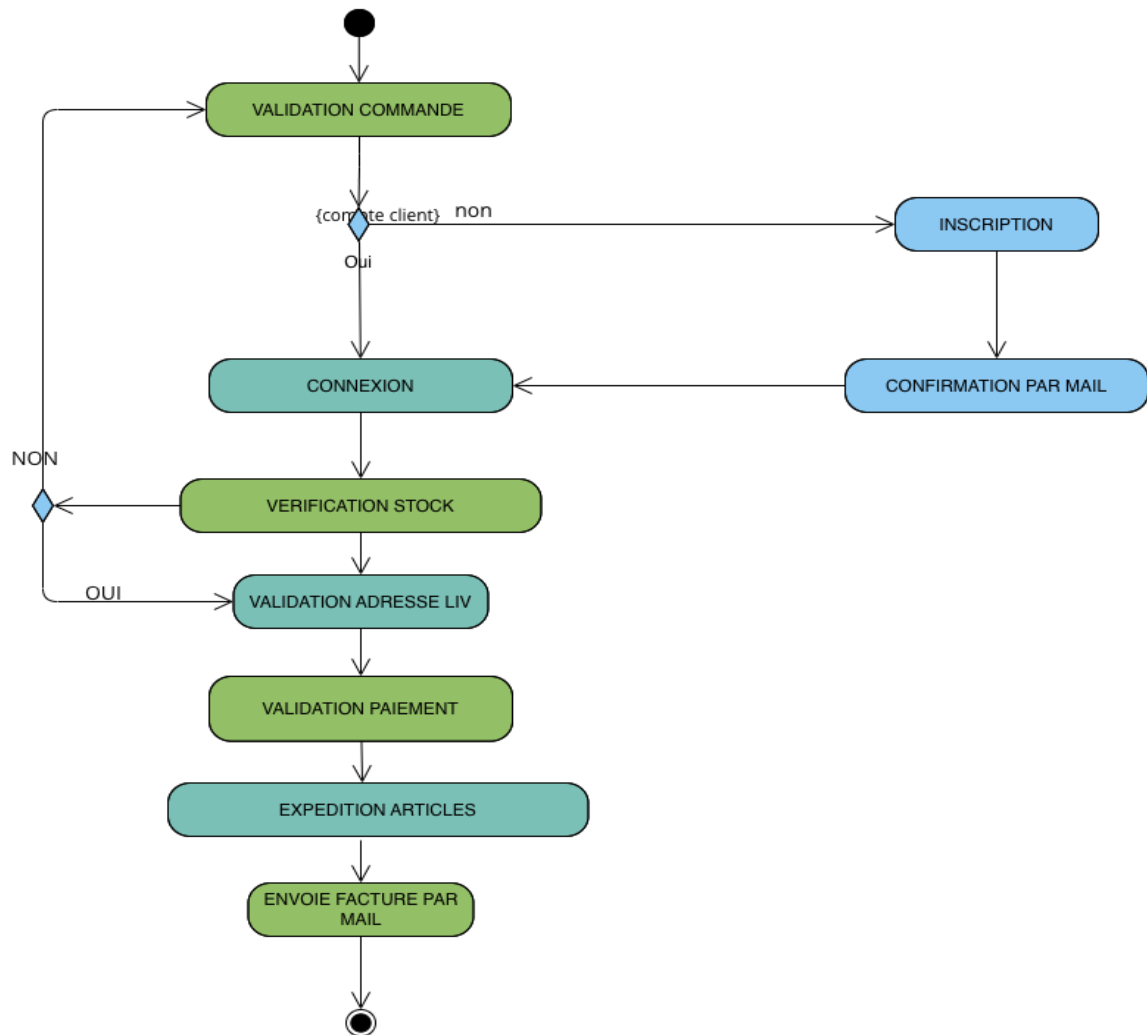
Une classe identifie les attributs, les opérations, les relations que les objets de la classe possèdent.

Voici le schéma représentant le diagramme de classe :



3. DIAGRAMME D'ACTIVITE

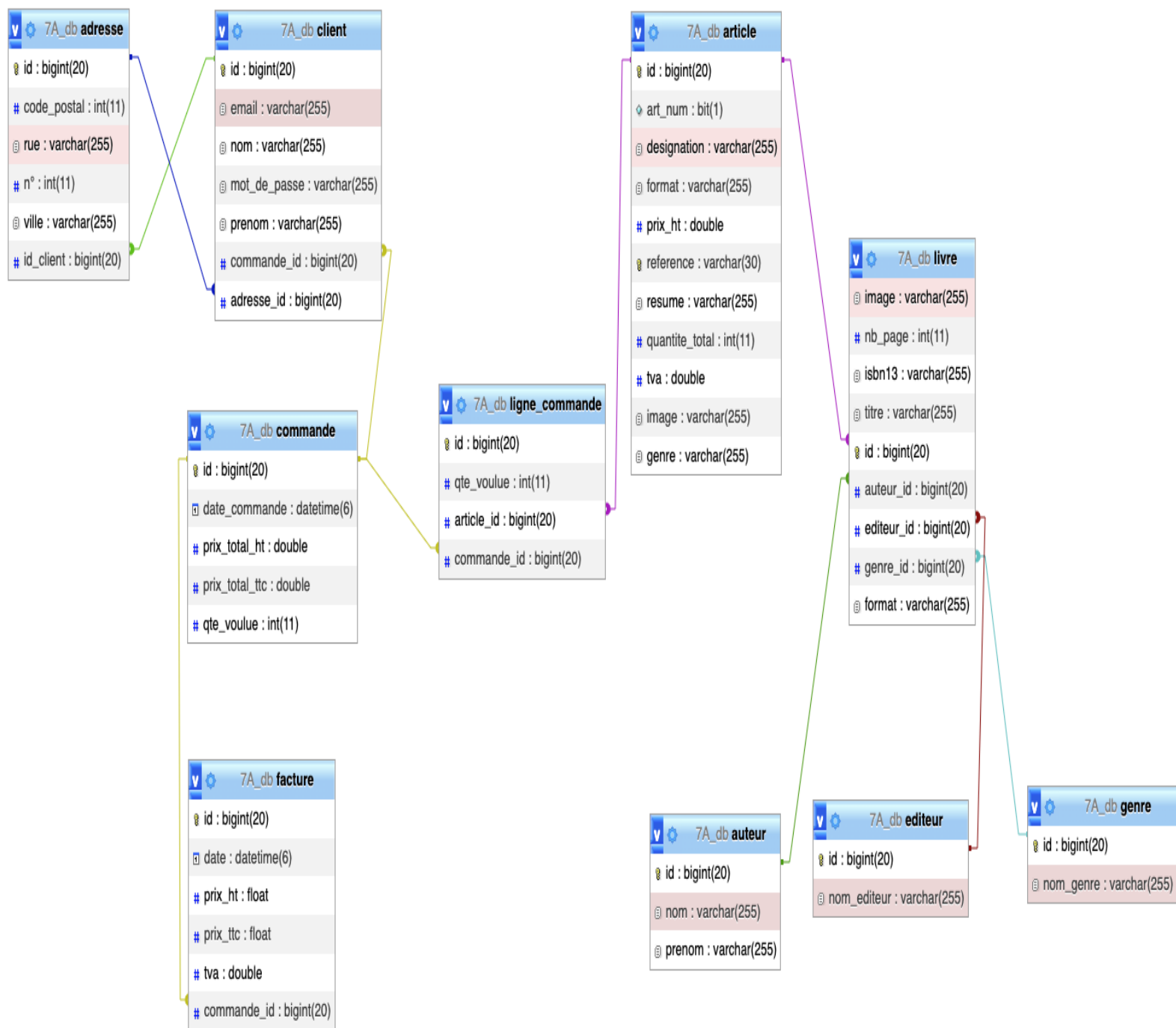
Cela sert à décrire les événements qui ont lieu dans un enchaînement d'activités de processus métier.



4. MODELE DE DONNEE RELATIONNEL

Voici le modèle physique qui montre les informations dans chaque table. Le modèle représente les relations existantes entre les tables, chaque table possède une clé primaire.

MPD



Voici la base de données que j'ai créée avec phpMyAdmin, toutes les tables sont créées pour la version 1 mais aussi pour les prochaines.

Nouvelle base de données

7A_db

Nouvelle table

adresse

article

auteur

client

commande

editeur

facture

genre

ligne_commande

livre

information_schema

mysql

performance_schema

sys

Contenant le mot :

Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> adresse	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8_general_ci	32,0 kio	-
<input type="checkbox"/> article	★ Parcourir Structure Rechercher Insérer Vider Supprimer	9	InnoDB	utf8_general_ci	32,0 kio	-
<input type="checkbox"/> auteur	★ Parcourir Structure Rechercher Insérer Vider Supprimer	24	InnoDB	utf8_general_ci	16,0 kio	-
<input type="checkbox"/> client	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8_general_ci	48,0 kio	-
<input type="checkbox"/> commande	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8_general_ci	16,0 kio	-
<input type="checkbox"/> editeur	★ Parcourir Structure Rechercher Insérer Vider Supprimer	20	InnoDB	utf8_general_ci	16,0 kio	-
<input type="checkbox"/> facture	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8_general_ci	32,0 kio	-
<input type="checkbox"/> genre	★ Parcourir Structure Rechercher Insérer Vider Supprimer	13	InnoDB	utf8_general_ci	16,0 kio	-
<input type="checkbox"/> ligne_commande	★ Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8_general_ci	48,0 kio	-
<input type="checkbox"/> livre	★ Parcourir Structure Rechercher Insérer Vider Supprimer	7	InnoDB	utf8_general_ci	64,0 kio	-
10 tables	Somme	73	InnoDB	utf8_general_ci	320,0 kio	0 o

↑ ☐ Tout cocher Avec la sélection :

5. MAQUETTES

Voici 3 maquettes qui représentent le futur site. Chaque maquette correspond à une page.

PAGE D'ACCUEIL

Il y a aura maximum 10 articles, les dernières nouveautés, les meilleures ventes, l'utilisateur ne sera pas obligé de se connecter pour visiter le site. Il pourra aussi choisir les livres, les ajouter dans le panier où le nombre total s'affichera depuis la page d'accueil.



- QUI SOMMES-NOUS? CGV INFORMATIONS PERSONNELLES COOKIES

PAGE PANIER

Le client pourra mettre des articles sans obligation de s'identifier, il pourra ajouter des livres, les enlever ou supprimer la ligne du panier, le total se calculera automatiquement.

The mockup shows a web page for a cinema bookstore's shopping cart. At the top, there is a navigation bar with links: "Meilleures ventes", "Service client", "Retours/suivi", and "Panier" (with a hamburger menu icon). Below the navigation bar is a dark grey header section with a white oval placeholder for a logo on the left. To the right of the logo, the text "Votre librairie spécialisée dans le domaine du cinéma" is displayed. In the center of this header, the text "CONNEXION/INSCRIPTION" is shown above a white button labeled "Valider". Below the header is a large grey rectangular area containing a form. The form has a white rectangular input field at the top left, followed by a long white rectangular input field. Below these are two rows of four white rectangular input fields each, arranged in a 2x4 grid. At the bottom of the grey area, there is a white rectangular button on the left and a red rectangular button labeled "Valider le panier" on the right. At the very bottom of the page, there is a footer with the text: "• QUI SOMMES-NOUS? CGV INFORMATIONS PERSONNELLES COOKIES".

Meilleures ventes Service client Retours/suivi • Panier

Votre librairie spécialisée dans le domaine du cinéma

CONNEXION/INSCRIPTION

Valider

Valider le panier

• QUI SOMMES-NOUS? CGV INFORMATIONS PERSONNELLES COOKIES

LA PAGE AUTHENTIFICATION

Quand le panier sera validé, il faudra que l'utilisateur s'authentifie, soit il crée un compte client avec obligation de renseigner un formulaire et valider un lien via son email ou il faudra qu'il se connecte s'il ne l'a pas fait auparavant.

Meilleures ventesService clientRetours/suivi• Panier

Votre librairie spécialisée dans le domaine du cinéma

CONNEXION/INSCRIPTION

Valider

Connexion

Cliquez

cliquez

• QUI SOMMES-NOUS? CGV INFORMATIONS PERSONNELLES COOKIES

PRESENTATION DE LA CHARTE DE NOMMAGE

Ce tableau représente la charte de nommage de l'application en général.

ÉLÉMENTS	RÈGLE	EXEMPLE
Package	Nommage minuscule en	controller model repository service
Classes	Nommage en Pascal Case	AdresseController
Attributs	CamelCase/sans accent	articleNumerique
SQL	Snake case	prix_ht
Langue	Nommage en français car le site est déployé en France, cible un public francophone.	

CHAPITRE 4 : ARCHITECTURE TECHNIQUE

1. CHOIX DE L'ARCHITECTURE

COTE FRONT

J'ai choisi comme technologie pour l'interface utilisateur :

REACT, c'est un langage très utilisé par les sites d'e-commerce, c'est un Framework JS, ça permet de créer des sites d'application web single page grâce aux composants.

Chaque application web React est composée de composants réutilisables qui constituent des parties de l'interface utilisateur, on peut avoir un composant distinct pour notre barre de navigation, un pour le pied de page, un autre pour le contenu principal, et ainsi de suite.

Le fait de disposer de ces composants réutilisables facilite le développement car on n'a pas à répéter du code récurrent. Il faut importer le composant dans n'importe quelle partie du code où il est nécessaire.

React est également une application à page unique. Ainsi, au lieu d'envoyer une requête au serveur à chaque fois qu'une nouvelle page doit être rendue, le contenu de la page est chargé directement à partir des composants.

Cela conduit à un rendu plus rapide sans rechargement de la page.

COTE BACK

J'ai choisi SpringBoot car c'est un Framework JAVA qui facilite l'automatisation du code.

Spring Boot est une infrastructure open source d'entreprise couramment

utilisée qui permet de créer des applications autonomes de production qui fonctionnent sur la machine virtuelle Java (JVM).

Java Spring Boot est un outil qui accélère et simplifie le développement d'applications Web et de micro services avec Spring Framework grâce à trois fonctionnalités principales :

Configuration automatique
Approche directive de la configuration
Possibilité de créer des applications autonomes

Ces fonctionnalités fonctionnent ensemble pour fournir un outil qui permet de configurer une application Spring avec une configuration et une installation minimale.

Spring Framework offre une fonction d'injection de dépendances qui permet aux objets de définir leurs propres dépendances que le conteneur Spring leur injecte ensuite. Ainsi, en tant que développeur, on peut créer des applications modulaires constituées de composants faiblement couplés qui sont idéaux pour les micro services et les applications de réseau distribué.

POURQUOI SPRING BOOT ?

Aussi performant et complet que soit Spring Framework, la configuration, l'installation et le déploiement d'applications Spring requièrent toujours beaucoup de temps et de connaissances. Spring Boot réduit cet effort grâce avec trois fonctionnalités importantes.

Configuration automatique

La configuration automatique signifie que les applications sont initialisées avec des dépendances prédéfinies. Java Spring Boot étant doté de fonctionnalités intégrées de configuration automatique, il configure automatiquement les packages Spring Framework et tiers sous-jacents en

fonction de vos paramètres (et des meilleures pratiques) pour éviter les erreurs.

Approche directive

Spring Boot utilise une approche directive pour ajouter et configurer les dépendances de démarrage, en fonction des besoins du projet. Spring Boot choisit lui-même les packages à installer et les valeurs par défaut à utiliser.

Par exemple, la dépendance de démarrage « Spring Web » permet de créer des applications Web Spring avec une configuration minimale en ajoutant au projet toutes les dépendances nécessaires, telles que le serveur Web Apache Tomcat.

Applications autonomes

Avec Spring Boot, on peut créer des applications qui s'exécutent directement. Concrètement, il permet de créer des applications autonomes qui s'exécutent seules, sans dépendre d'un serveur Web externe, en intégrant un serveur Web tel que Tomcat dans l'application au cours du processus d'initialisation.

Il permet de créer des sites web extensibles et évolutifs donc adaptée en cas de forts trafics. Le Framework Spring Boot permet d'écrire des webservices qui pourra supporter les forts trafics.

Le but du Webservices REST est de faire des requêtes et ainsi communiquer avec la BDD. Elles permettent de transmettre des messages et des données pour les architectures client-serveur.

Schéma d'architecture générale

Ce schéma représente l'architecture générale de l'application entre le client et le serveur via la librairie Axios qui permet les requêtes avec le Controller dans Spring Boot.

Spring Boot + React Full Stack Application Architecture

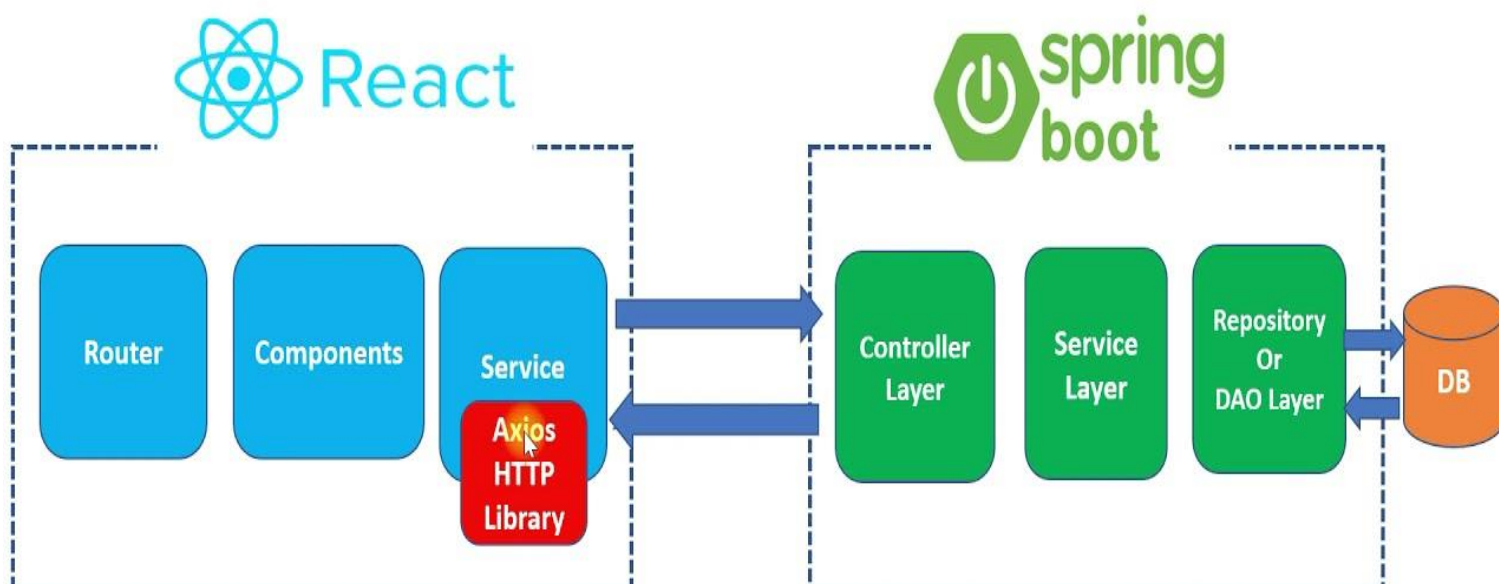
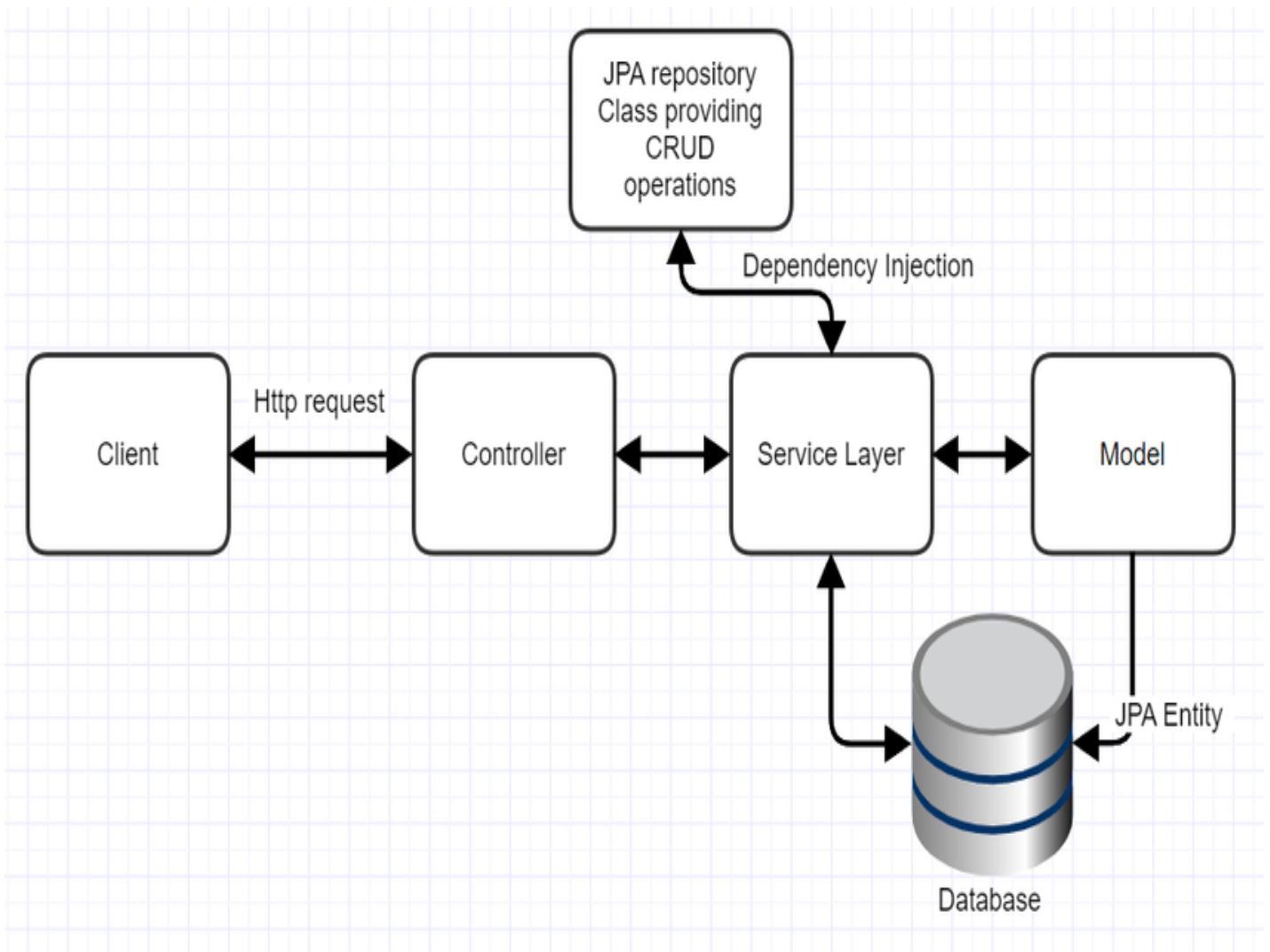


Schéma d'architecture back

Voici le schéma plus détaillé du serveur



2. CHOIX DES OUTILS

J'utilise l'IDE IntelliJ pour la programmation Java du back, car c'est un outil complet, intégrant l'outil de build Maven et Git.

Pour le front, j'ai choisi Visual Studio Code car c'est un outil simple à gérer et il offre la possibilité de pouvoir visionner les pages implémentées.

J'utilise MAMP pour l'utilisation de la base de données MySQL et de sa console d'administration PHPMyAdmin, simple d'utilisation.

3. TESTS

TEST UNITAIRE

J'ai mis en place des tests Unitaires, cela permet de vérifier qu'une partie du code fonctionne correctement et ainsi améliorer le résultat final de l'application.

Le but est d'améliorer la lisibilité du code, ce qui sera plus facilement modifiable par la suite, si un autre développeur reprend l'application pour faire évoluer le site par exemple.

Il permet de tester d différentes parties du code indépendamment, dans le cadre de l'agilité, c'est très pratique.

J'ai réalisé des tests unitaires dans le but de tester le comportement du stock.

La décrémentation et l'incrémentation lors d'une commande.

TEST API

Je teste les requêtes API avec Postman, permettant d'accéder à l'application sans interface utilisateur, ce qui aide à identifier les erreurs plus tôt dans le cycle de développement.

On peut télécharger la documentation via ce lien :

AppSept.postman_collection.json

TEST D'INTÉGRATION

Les tests d'intégrations incluent dans leur champ d'action les composants externes. Ceux-ci peuvent-être une base de données, un service web ou bien un système de fichier. En les intégrant, ils viennent contrôler la communication entre l'application et ces éléments périphériques. J'ai mis en place des tests d'intégration pour vérifier que la communication entre la base de données et le serveur.

CHAPITRE 5 : INSTALLATION, MAINTENANCE ET VEILLE

1. MANUEL D'INSTALLATION ET D'EXPLOITATION

INSTALLATION

Dans tous les cas, créer un compte GitHub pour récupérer le projet avec la clé SSH ou l'url ou générer le projet depuis l'interface :

- Télécharger le projet zip et l'extraire dans le dossier de votre choix.
Importer le projet dans votre IDE.

Pour Spring Boot :

L'application est sur le port : 8080

Ajouter les dépendances suivantes :
Le projet est sur JAR

Le serveur Tomcat est lancé par Spring

TOMCAT permet de lancer l'application sur le port 8080.

MAVEN permet d'automatiser la gestion de projets Java, il offre la compilation et déploiement des applications (JAR), gestion des librairies requises par l'application, Exécution des tests unitaires, génération des documentations du projet.

Spring Web
Spring Data JPA
Spring Dev Tools

Pour lancer l'application dans le terminal :
Run project

Tester l'application :

API

Postman permet de tester les appels API :

<https://www.postman.com/downloads/>

Le lien du fichier généré se trouve en dessous :

[\[AppSept.postman_collection.json\]](#)(*AppSept.postman_collection.json*)

Spring Test : permet de réaliser les tests

La base de données :

- Télécharger Mamp : <https://www.mamp.info/en/downloads/>
- Vérifier qu'apache est bien sélectionné
- Lancer la base de donnée avec le bouton Start
- Connecter vous sur phpMyAdmin :
http://localhost:8888/phpMyAdmin5/index.php?route=/database/structure&db=7A_db

Nom de la base de donnée: 7A_db

port: 8888

React :

Port: 8080

Installer Sass : Framework de CSS

Création du projet : npx create-react-app 7-arche-client

Prérequis pour installer le projet :

Node js 20.5.1

npm 10.0.0

Axios

Pour lancer l'application

npm start

Pour installer une librairie

npm install

2. DOCUMENTATION TECHNIQUE

OUTIL JAVADOC

Le but de cette documentation qui est généré grâce à l'outil JAVADOC, à partir du commentaire spécial mis dans le code, est de permettre à un développeur qui reprendra le projet de comprendre comment le code a été implémenter. J'ai sélectionné quelques classes et packages, qui me paraissent importantes, pour mettre des explications.

Voici le lien :

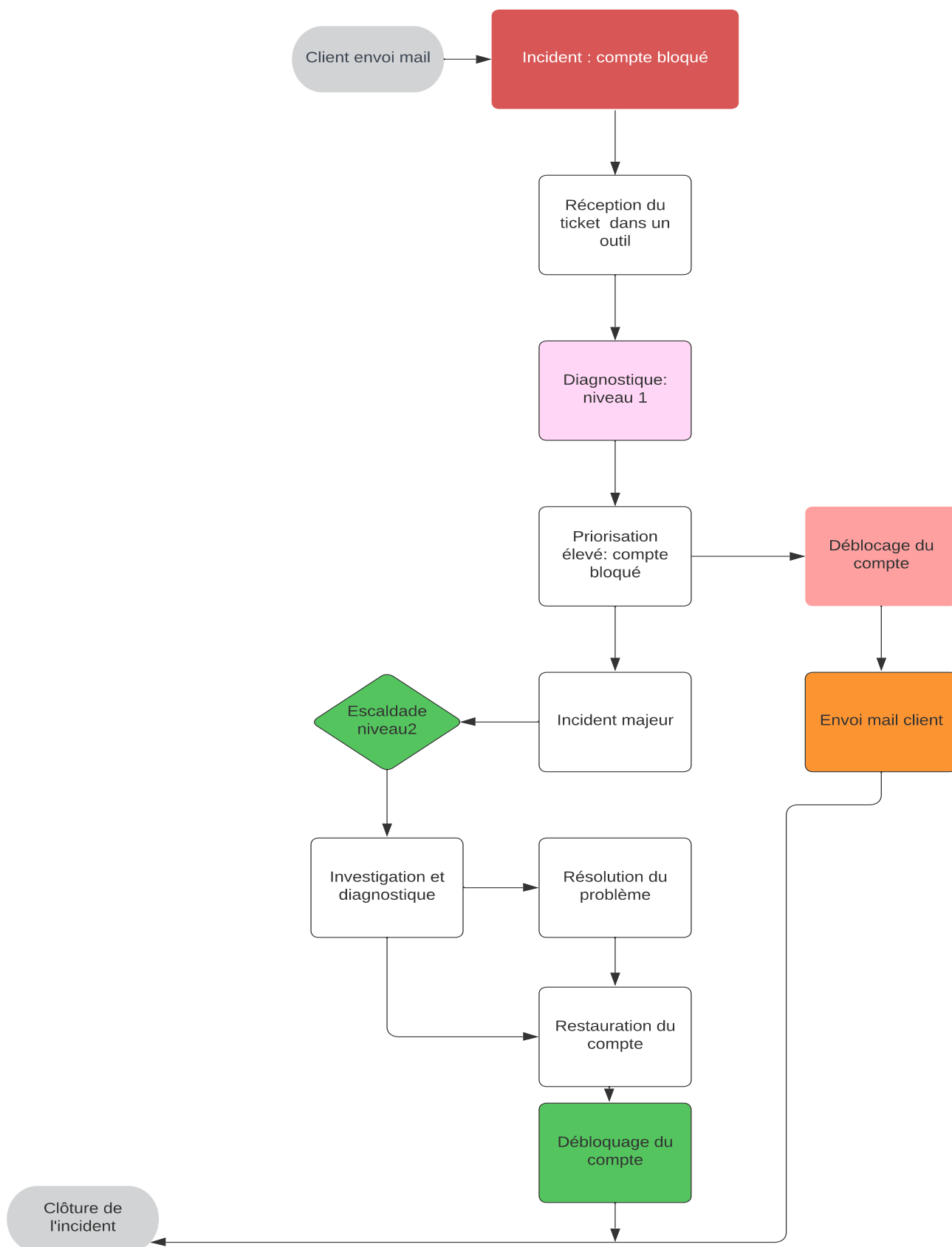
file:///Users/fatima.kaaouan/Documents/Projet_app/javadoc/allclasses-index.html

GESTION DE PROCESSUS DES INCIDENTS SELON ITIL

ITIL signifie Information Technology Infrastructure Library. C'est un guide de bonnes pratiques appliqués dans la gestion IT. Cela a été mis en place pour améliorer la communication entre les services IT et les métiers, des différents problèmes et incidents rencontrés par les utilisateurs.

Le but est de mettre en place des processus pour améliorer la qualité de service et ainsi la productivité.

J'ai mis en place un Algorigramme montrant un incident de compte bloqué à la suite d'une mauvaise saisie du mot de passe. Différents niveaux de priorité ont été mis en place pour que l'incident soit traité le plus efficacement possible et que la demande soit traité sur l'outils GPLI.



CLEAN CODE

J'ai appliqué tout au long du code, de la documentation et de la présentation du Clean Code, qui consiste à rester cohérent, ne pas se répéter, j'ai donné des noms de variables claires et explicite et sont à proximités de leurs usages.

3. MAINTENABILITE

Il est prévu de mettre en place pour que le site continue d'être compétitif, une maintenance automatisée sera mise en place :

Pour que le site soit performant et référencer :

- Suppression des contenus obsolètes (contenu faible en valeur)
- Détection des pages inaccessibles (erreurs 404 : url)
- Mise à jour ou ajout de nouveaux contenus (ajout de nouveaux livres ou articles à la mode)
- Tâches qui seront automatisées dans la mesure du possible :
- Sauvegarde régulière de la base de données
- Nettoyage et Optimisation de la base de données
- Suppression des fonctionnalités inutilisées
- Amélioration de la vitesse du site

4. DEPLOIEMENT

Le serveur est sur le serveur local dans un premier temps. Il est prévu que le site sera hébergé chez Heroku.

5. VEILLE TECHNOLOGIQUE

Pour rester à jour dans le métier, il est nécessaire de maintenir une veille régulièrement, ça permet de rester à jour dans les dernières nouveautés, ce qui permet de s'adapter en permanence et d'anticiper éventuellement.

AU NIVEAU DES LANGAGES

Je vais sur les sites officiels et non officiels des stacks que j'utilise :

- JAVA : <https://docs.oracle.com/en/java/>
- SPRING : <https://spring.academy/courses>
- react : <https://react.dev/>
- MDN : <https://developer.mozilla.org/fr/>
- <https://www.codeur.com/blog/veille-technologique-definition-outils-et-enjeux>
- <https://openclassrooms.com>
- <https://www.jmdoudoux.fr/accueil.html>
- <https://www.ibm.com/>
- <https://numeum.fr/>
- <https://stackoverflow.com/>

Certains de ces sites, ont des forums où l'on peut communiquer entre développeurs en cas de problèmes rencontrés.

Je maintien une veille dans divers sujets liés au code comme la cybersécurité car c'est important d'en prendre compte, la protection des données et le design :

- <https://www.cnil.fr/>
- <https://www.ssi.gouv.fr/en/>
- <https://www.behance.net/onboarding/adobe>

Je fais régulièrement des tutoriels sur YouTube et Openclassrooms car c'est important de s'autoformer pour rester à jour de ses connaissances.

Je suis abonnée aux chaînes suivantes :

- fromScratch,
- professeur Yousri
- code With Murad

J'écoute et je lis régulièrement en anglais car c'est indispensable.

Je lis quelques livres comme Clean Code de Robert C. Martin Series, HTML de Deidre Hayes, quelques documentations et livres mis à dispositions des salariés d'Octo.

CONCLUSION

Tout au long de ce projet, j'ai appliqué la méthodologie agile en mettant en place des tickets sur Trello, en faisant des réunions avec les autres développeurs de la promotion, j'ai fait appel à des développeurs dans mon entreprise, je suis allée sur des forums spécialisés dans l'entraide IT pour demander de l'aide, faire du pair programming avec mes collègues et demander des conseils.

Les risques ont été analysés afin de respecter la deadline.

Acquérir les bases du métier du développeur était mon but tout au long du projet.

ANNEXE

Logos des outils sur lesquels, j'ai travaillé :



Visual Studio Code

