

Projet réalisé dans le cadre de la soutenance  
devant le Jury Final Pour l'obtention du titre

## **Concepteur Développeur d'Applications**



## SOMMAIRE

### Table des matières

Résumé / Abstract.....	5
1 Introduction.....	5
2 Contexte du projet.....	5
3 Cahier des charges.....	6
3.1 Benchmark.....	6
3.2 Persona.....	7
3.3 Expression & analyse des besoins.....	8
3.4 Exigences technico-fonctionnelles.....	9
3.5 MVP (V. 1.0).....	10
3.6 Version 2.0.....	10
3.7 Version 3.0.....	10
4 Méthodologie de travail.....	10
4.1 Cycle en Spirale.....	10
4.2 Cycle en V.....	11
4.3 Méthodologie Agile.....	12
4.3.1 Méthodologie Agile/Scrum.....	12
4.3.2 Méthodologie Agile/Kanban.....	13
4.3.3 Méthodologie Agile/Scrumban.....	15
4.3.4 Méthodologie Agile/TDD.....	15
4.3.5 Méthodologie Agile/FDD.....	16
5 Spécifications fonctionnelles.....	17
5.1 Fonctionnalité "S'authentifier".....	17
5.2 Fonctionnalité "S'inscrire".....	17
5.3 Fonctionnalité "Ajouter un manga".....	17
5.4 Fonctionnalité "Ajouter article au panier".....	17
6 Spécifications techniques.....	17
6.1 L'architecture logicielle multicouches MVC.....	17
6.2 L'architecture n-Tiers → (3-Tiers).....	18
6.3 Technologies choisies.....	18

6.3.1 Technologies côté Front-end.....	18
6.3.2 Technologies côté Back-end.....	20
6.3.3 Technologies côté Persistance des données.....	21
6.3.4 Technologies transverses.....	21
7 Conception de l'interface graphique.....	22
7.1 Expérience UX/UI.....	22
7.2 Charte graphique.....	22
7.3 Responsive.....	23
7.4 Zoning.....	23
7.5 Wireframes.....	27
7.6 Maquettes.....	30
7.7 Arborescence du site.....	33
8 Conception de l'application.....	33
8.1 Le langage de modélisation UML.....	33
8.2 Diagramme de cas d'utilisation.....	34
8.3 Diagramme de séquences.....	34
8.4 Diagramme de classes.....	35
8.5 Définition des Routes.....	35
8.5.1 Routes des pages.....	35
8.5.2 Routes de l'API REST.....	36
9 Conception de la Base de données.....	36
9.1 Conception de la Base de données relationnelle MySQL.....	36
9.1.1 La méthode MERISE.....	36
9.1.2 Le Modèle Conceptuel de Données (MCD).....	36
9.1.3 Le Modèle Logique de Données (MLD).....	37
9.1.4 Le Modèle Physique de Données (MPD).....	38
9.1.5 Le Dictionnaire de Données (DD).....	38
9.2 Conception de la Base de données non relationnelle NoSQL.....	39
9.2.1 Paramètre de Connexion.....	39
9.2.2 Création de la Base de Données.....	39
10 Développement, Intégration et Réalisations.....	40
10.1 Structure du projet.....	40
10.2 Fonctionnalité.....	41
10.2.1 Fonctionnalité « Ajouter un manga ».....	41

10.2.2 Fonctionnalité « Supprimer un manga ».....	42
.....	42
10.3 Requêtes SQL.....	42
10.3.1 Requête SQL de lecture.....	42
10.3.2 Requêtes SQL d'écriture.....	43



## Résumé / Abstract

My name is Jester CESAR, I'm 25 years old, I'm became from a computer science degree, in this degree I learned to use java, c++, python, html, css and others languages. So I was looking for a study contract but I didn't find an employer and I'm find this training program. I'm here today to present my project in order to have the certification of Concepteur Développeur d'Applications

I really enjoyed this formation all along, I learned some new languages and frameworks like JavaScript, React and Node js.

I'm really passionate to creates websites, it's something indescribable when you're coding as it might be not correct at the moment but you will try to find a solution on your own and when it's work it's something special.

For the final project, I choose to do a streaming website for anime only. The website will have a streaming service, a e-commerce service and a reading online section. Therefore the customers will need to be connected to benefit all the services except for the e-commerce.

As the administrator, I will have access to all the function and I will add some new manga or some new products in the store. For the future, I want to add a blog section with integrated messaging like a social media for the fans.

## 1 Introduction

Je m'appelle Jester CESAR, j'ai 24 ans et je suis actuellement étudiant pour obtenir le titre professionnel Concepteur Développeur d'Application chez Doranco. J'étais en licence en Informatique à l'université François-Rabelais de Tours à l'antenne de Blois. J'ai validé mes deux premières années puis je voulais pour ma troisième année faire une alternance mais malheureusement je n'ai pas trouvé d'employeur. Puis j'ai trouvé cette formation pour devenir Concepteur Développeur d'Application que je voulais faire et qui a été très intéressante, j'ai appris beaucoup de nouveaux langages et viennent compléter mes compétences déjà acquise. Après cette formation je voudrais continuer mes études car c'est grâce à Doranco et l'opportunité qui m'ont donnée qui m'a redonné envie de continuer mes études. Pour mon projet de fin d'études j'ai choisi de faire un site de vidéo à la demande qui est spécialisé dans les animes.

## 2 Contexte du projet

3 mars 2024, les dirigeants des 4 plus grandes maisons d'éditions de manga du Japon se réunissent afin de parler de la soirée de la veille avec comme sujet la 8ème édition des Crunchyroll Anime Awards. Les 4 maisons d'éditions sont :

- ◆ Shueisha : c'est la maison d'édition japonaise la plus connue dans le monde avec l'un des magazines de prépublication de manga le plus populaire; le Weekly Shōnen Jump qui a eu dans son histoire des séries telles que Dragon Ball, Naruto ou One Piece.
- ◆ Kadokawa Future Publishing : c'est la branche d'édition de Kadokawa Corporation qui publie des mangas, romans, light novel et des magazines.

- ◆ Kōdansha : c'est la maison d'édition la plus grosse du Japon, oeuvrant pour la littérature et les mangas avec des magazines spécialiser en tant qu'éditeur de manga et en tant qu'éditeur conventionnel.
- ◆ Square Enix : c'est une société japonaise qui développe des jeux vidéo et publie des mangas avec leurs différents magazines.

Leurs soucis c'est Crunchyroll, un service de vidéo à la demande américain qui appartient à Sony Corporation, est l'un des seules service de streaming qui est spécialisée sur les animes et qui à le monopole sur cette catégorie. Il y a bien sur d'autres services de streaming comme Netflix, Amazon Prime ou Disney + qui propose des animes mais aucun qui puissent concurrencer Crunchyroll. Leur solution retirer tous les droits aux services de streaming précédemment citer et créer leur propre site de vidéo à la demande avec aucune collaboration avec les américains et concurrencer Crunchyroll à l'international.

Plusieurs semaines plus tard un tournoi de Budokai Tenkaichi est lancé pour savoir qui sera le concepteur de ce projet. Après avoir remporter le tournoi, j'ai donc été choisi pour concevoir ce projet avec l'appui des dirigeants.

### 3 Cahier des charges

#### 3.1 Benchmark

Définition : Benchmark aussi appelé Benchmarking, est un terme anglais qui peut être traduit par les mots « référence » ou « repère ». Le benchmark est une technique marketing basé sur l'analyse comparative. Elle vise à étudier d'autres entreprises pour optimiser ses propres techniques de gestion et modes d'organisation.

Voici 3 exemples de Benchmark :



FORCES	FAIBLESSES
<ul style="list-style-type: none"> <li>- Réputations de marque.</li> <li>- Qualité des contenus</li> <li>- Large clientèle + Présence mondiale</li> <li>- Communauté active et engagée</li> </ul>	<ul style="list-style-type: none"> <li>- Qualité vidéo selon les régions</li> <li>- Expérience utilisateur</li> <li>- Concurrence croissante sur le marché</li> <li>- Prix MegaFan Trop chère</li> </ul>
OPPORTUNITÉS	MENACES
<ul style="list-style-type: none"> <li>- Expansion vers nouveaux marchés</li> <li>- Amélioration qualité vidéo et expérience Utilisateur</li> <li>- Contenu originaux et collaborations</li> </ul>	<ul style="list-style-type: none"> <li>- Concurrence nouveaux acteurs sur le marché</li> <li>- Risque de perte contenu exclusif</li> <li>- Pression sur les prix</li> </ul>



FORCES	FAIBLESSES
<ul style="list-style-type: none"> <li>- Catalogue large (animes populaires et exclusifs)</li> <li>- Interface utilisateur conviviale</li> <li>- Abonnement flexibles avec options d'achat</li> <li>- Partenariat studios japonais pour diffusion simulcast</li> </ul>	<ul style="list-style-type: none"> <li>- Disponibilité limités</li> <li>- Absence fonctionnalités sociales et interactives</li> <li>- Moins de contenus originaux ou exclusifs</li> </ul>
OPPORTUNITÉS	MENACES
<ul style="list-style-type: none"> <li>- Expansion géographique</li> <li>- Renforcer les partenariats</li> <li>- Intégration fonctionnalités sociales</li> </ul>	<ul style="list-style-type: none"> <li>- Concurrence nouveaux acteurs sur le marché</li> <li>- Risque de perte contenu exclusif</li> <li>- Rester compétitif face aux meilleures offres</li> </ul>

## Les sites de streaming illégales

FORCES	FAIBLESSES
<ul style="list-style-type: none"> <li>- Accès gratuit à un large contenus</li> <li>- Disponibilité certains titres pas encore disponibles légalement</li> <li>- Popularité auprès des utilisateurs qui recherchent des options gratuites</li> </ul>	<ul style="list-style-type: none"> <li>- Qualité vidéo inférieurs avec une traduction de mauvaise qualité</li> <li>- Manque de fiabilité et sécurité</li> <li>- Aucun soutien aux créateurs</li> </ul>
OPPORTUNITÉS	MENACES
<ul style="list-style-type: none"> <li>- Exploitation lacunes certains contenus non disponibles</li> <li>- Monétisation par les pubs</li> </ul>	<ul style="list-style-type: none"> <li>- Risques poursuites judiciaires</li> <li>- Concurrence croissante des streaming légaux</li> <li>- Pression sur la lutte contre le piratage</li> </ul>

### 3.2 Persona

Mrs. Tiffany Kutch

<b>Intérêts</b> Animés de tous genres, surtout les œuvres visuellement captivantes, culture japonaise, cosplay, conventions d'anime	<b>Comportement en ligne</b> Regarde des animés pendant son temps libre, surtout le soir. Elle aime commenter et noter les séries sur des forums spécialisés. Elle suit aussi les influenceurs qui parlent d'animes sur YouTube et Twitter.
<b>Objectifs</b> Découvrir des animés moins connus, échanger avec une communauté de passionnés, suivre les nouveautés en streaming de manière légale et sans publicité.	<b>Facteurs d'influence des décisions d'achat</b> Une plateforme qui propose des abonnements étudiants à prix réduit, des recommandations basées sur ses préférences visuelles, et une interface moderne. Elle souhaite aussi une section communautaire où elle peut échanger avec d'autres passionnés.
<b>Frustrations</b> Certaines séries ne sont pas toujours disponibles en streaming en France ou à l'international, coût élevé des abonnements sur certaines plateformes, manque de diversité dans les recommandations personnalisées.	

**Données démographiques**

Âge  
19

Lieu  
Paris

Niveau d'instruction  
Etudiante en 1ère année d'université,

## Projet Final – Concepteur Développeur d'Applications



Léo Chang Yung

**Données démographiques**

Âge: 12  
Lieu: Strasbourg  
Niveau d'instruction: Collégien en classe de 5ème

**Intérêts**  
Animés populaires, jeux vidéo, culture geek, manga

**Objectifs**  
Suivre les animés les plus en vogue parmi ses amis, découvrir de nouveaux épisodes dès leur sortie, et discuter des épisodes à l'école.

**Comportement en ligne**  
Regarde les animés après l'école, principalement sur son téléphone ou sa tablette. Il partage ses découvertes avec ses amis via les réseaux sociaux ou par messages. Il utilise les plateformes gratuites ou les essais gratuits pour accéder au contenu.

**Facteurs d'influence des décisions d'achat**  
Un site facile à utiliser, compatible avec les appareils mobiles, avec des recommandations d'animés populaires et un système de notifications pour les nouveaux épisodes. Le site doit proposer des options gratuites ou abordables.

**Frustrations**  
Difficulté à trouver des plateformes gratuites ou abordables avec tous les épisodes disponibles, publicités intrusives, connexion internet parfois lente chez lui.



Thomas Nicolas

**Données démographiques**

Âge: 28  
Lieu: Lille  
Niveau d'instruction: Photographe dans une startup

**Intérêts**  
Animés de science-fiction et cyberpunk, lecture de light novels, jeux vidéo, culture pop japonaise, musique japonaise (J-Pop)

**Objectifs**  
Regarder des animés de qualité en haute définition, se tenir au courant des dernières sorties, découvrir des séries plus matures et complexes, suivre les tendances tout en ayant accès à un catalogue varié.

**Comportement en ligne**  
Regarde des animés principalement le soir ou pendant ses pauses. Il préfère les versions en streaming premium sans publicité et en haute résolution. Il utilise des forums et des plateformes comme Reddit pour échanger des conseils et des avis.

**Facteurs d'influence des décisions d'achat**  
Une plateforme proposant une expérience premium avec des options de haute qualité, des recommandations personnalisées en fonction de son historique de visionnage, et une gestion facile de ses séries en cours.

**Frustrations**  
Manque de temps pour explorer de nouvelles séries, qualité vidéo et audio médiocre sur certaines plateformes, doublages de mauvaise qualité, trop de services de streaming à la fois.

## 3.3 Expression &amp; analyse des besoins

Fonctionnalités	Descriptions
Catalogue riche et varié	L'utilisateur s'attend à un vaste catalogue d'animés incluant des titres populaires, classiques et nouveaux par thème (shônen, shôjo, seinen, isekai).
Accessibilité des épisodes récents	L'utilisateur veut pouvoir visionner les derniers épisodes en simulcast avec le Japon.
Option linguistiques	L'utilisateur veut pouvoir choisir une option de langue pour les sous-titres et les doublages. (VOSTFR, VF, VOSTA, etc)
Navigation simple	L'utilisateur veut avoir accès à une interface intuitive et ergonomique simple et fluide. Cela inclut un système de tri et de filtrage et de recommandations.
Gestion liste de visionnage	L'utilisateur doit pouvoir créer et gérer simplement une liste de séries à

	suivre.
Qualité vidéo	L'utilisateur veut pouvoir regarder son films ou son épisodes avec une bonne qualité (minimum HD)
Disponibilité multi-plateformes	L'utilisateur souhaite accéder au site sur d'autres appareils avec un optimisation sur toutes les appareils.
Expérience immersive	L'utilisateur souhaite une interface sans distraction, sans publicité intrusive, et des options de personnalisation (avatars).
Communauté et appartenance	L'utilisateur veut pouvoir communiquer avec d'autres utilisateurs passionnés sur un forum.
Abonnement flexible et abordable	L'utilisateur veut un modèle économique adapté pour les différentes catégories d'utilisateur.
Sécurité des données	L'utilisateur attend une protection adéquate de leurs données personnelles.
Support client réactif	L'utilisateur veut pouvoir contacter un support technique disponible pour résoudre rapidement les problèmes de connexion, de paiement ou de visionnage.
Exclusivités	L'utilisateur souhaite des contenus exclusifs, avoir accès à des épisodes qui sortent sur la plateforme avant la diffusion sur les chaînes japonaises.
Documentaires et contenus additionnels	L'utilisateur peut également apprécier des contenus supplémentaires comme des documentaires sur la création des animés, des interviews du mangaka, de son représentant de sa maison d'édition, ainsi que des making-of

### 3.4 Exigences technico-fonctionnelles

Définition : c'est une déclaration sur la façon dont un système doit se comporter. Il définit ce que le système doit faire pour répondre aux besoins ou aux attentes de l'utilisateur et comment le système doit fonctionner en interne.

Les Exigences techniques et fonctionnelles de mon projet sont :

- ◆ Gestion des données des animes, des utilisateurs avec une base de données relationnelle et une non relationnelle
- ◆ Compatibilité avec les navigateurs
- ◆ Sécurité des données avec le respect de la loi RGPD, le respect des recommandations de la CNIL, les mentions légales, CGU, les cookies et le droit de rétraction
- ◆ Une fonction de recherche par titre et une filtration par genres
- ◆ Se prémunir et se protéger des différentes attaques de site comme DDoS, sécurisation des données utilisateurs (chiffrement, SSL/TLS)
- ◆ Interface utilisateur ergonomique et attrayante

### 3.5 MVP (V. 1.0)

**Définition :** Le minimum viable product (produit minimum viable en français) est la version d'un produit qui permet d'obtenir un maximum de retours clients avec un minimum d'effort. Il désigne la stratégie utilisée pour fabriquer, tester et mettre sur le marché ce produit.

Dans la 1ère version, il y aura :

- ✓ L'accueil
- ✓ L'authentification
- ✓ Profil
- ✓ Catalogue riche et varié
- ✓ La Boutique
- ✓ Sécurité des données

### 3.6 Version 2.0

Dans la 2ème version, il y aura :

- ✗ La page du Feeds disponible
- ✗ Les pages Mentions Légales
- ✗ Des Mangas d'autres maisons d'éditions qui rentrent dans le projet
- ✗ Options Linguistiques
- ✗ Gestion de liste de visionnage
- ✗ Support client

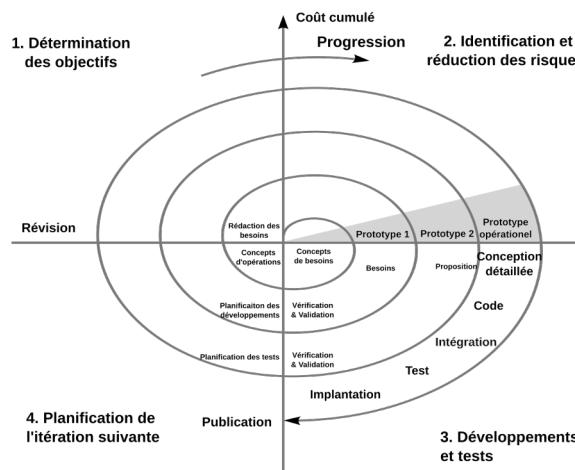
### 3.7 Version 3.0

Dans la 3ème version, il y aura :

- ✗ Nos propres pages des Personnages pour chaque manga
- ✗ Amélioration des pages Séries et Films
- ✗ Documentaires et contenus supplémentaires
- ✗ Page pour chaque studio d'animation

## 4 Méthodologie de travail

### 4.1 Cycle en Spirale



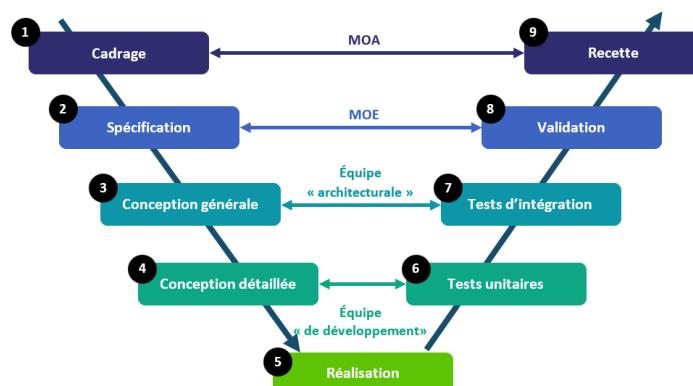
**Définition :** c'est un modèle de cycle de développement logiciel qui par plusieurs versions successives recommence en proposant un produit de plus en plus complet et dur. C'est l'un des modèles qui met le plus l'accent sur la gestion des risques.

On distingue quatre phases dans son déroulement :

- détermination des objectifs, des alternatives et des contraintes
- analyse des risques, évaluation des alternatives
- développement et vérification de la solution retenue
- revue des résultats et vérification du cycle suivant

Son plus gros inconvénient c'est qu'il faut que tous les éléments du diagramme soit faits dans l'ordre indiqué sans possibilité de retour en arrière. Les éléments du projet suivent une seule séquence en spirale.

## 4.2 Cycle en V



**Définition :** c'est un modèle d'organisation des activités d'un projet qui se caractérise par un flux d'activité descendant qui détaille le produit jusqu'à sa réalisation, et un flux ascendant qui assemble le produit en vérifiant sa qualité. Il est issu du modèle en cascade dont il reprend l'approche séquentielle et linéaire de phases.

Les étapes du modèle sont alors :

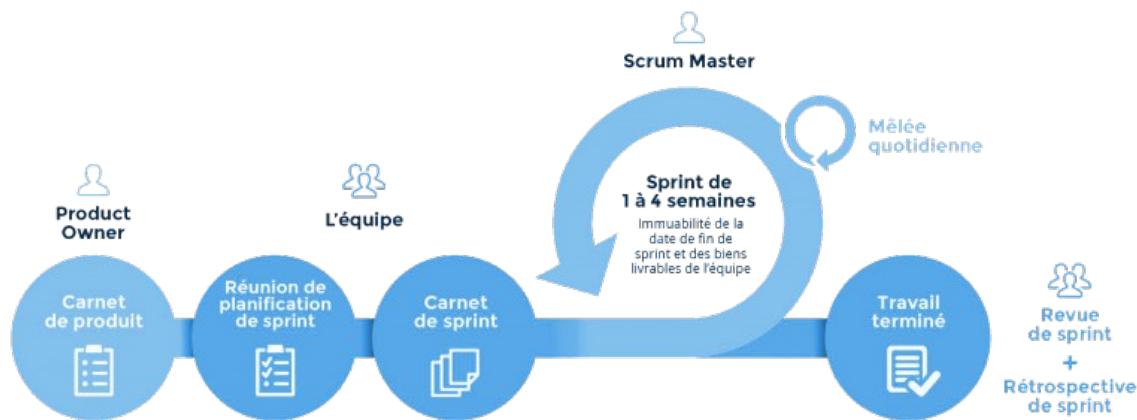
- ◆ Cadrage (ou exigences) → le client vous décrit ce qu'il souhaite et la faisabilité, le coût, le temps

- ◆ Spécifications → à partir des exigences du client on créer un cahier des charges fonctionnels
- ◆ Conception générale → il s'agit de concevoir le système qui doit répondre aux exigences et de définir son architecture, et en particulier les différents composants nécessaires
- ◆ Conception détaillée → il s'agit de concevoir chaque composant,
- ◆ Réalisation → elle permettra de créer et d'assembler les éléments pour finaliser le produit
- ◆ Tests unitaires → sert à vérifier le bon fonctionnement et la conformité de chaque composant à sa conception détaillée
- ◆ Tests d'intégration → sert à assembler le système à partir de tous ses composants, et de vérifier que le système fonctionne conformément à sa conception générale
- ◆ Validation → une fois l'application créée, des utilisateurs testent sa fonctionnalité globale
- ◆ Recette → le client vérifie que le produit fini correspond bien à ses attentes avant de valider

## 4.3 Méthodologie Agile

Définition: c'est une méthode de gestion de projet qui mettent en avant la collaboration entre des équipes auto-organisées et pluridisciplinaires et leurs clients. Elles s'appuient sur l'utilisation d'un cadre méthodologique léger mais suffisamment centré sur l'humain et la communication.

### 4.3.1 Méthodologie Agile/Scrum



Définition: l'approche Scrum est une méthode agile consacrée à la gestion de projet. Son objectif est d'améliorer la productivité des équipes, tout en permettant une optimisation du produit et une mise à jour régulière des priorités avec la réduction des bugs. Il y a 3 rôles différents dans la méthode Scrum :

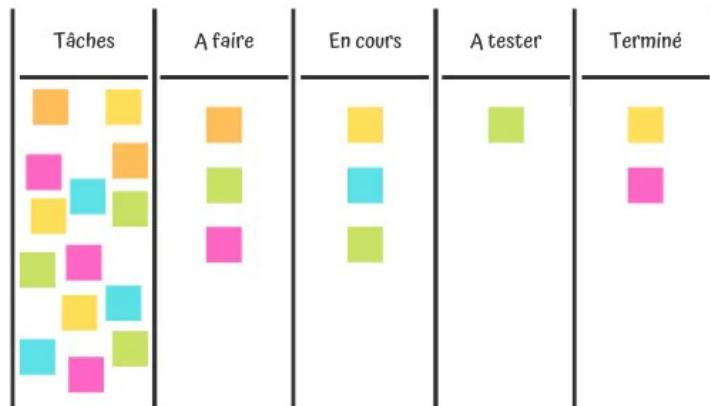
- ◆ le Scrum Master (c'est le coordinateur des équipes, c'est celui qui s'assure que les principes et les valeurs du Scrum sont respectés. Il vérifie que il y a de la communication entre les équipes et améliore la productivité)

- ◆ le Product Owner (c'est celui qui collabore avec le client, il définit avec le client les spécificités fonctionnelles du produit puis les priorise ensuite avec l'équipe et valide les fonctionnalités développées. C'est lui qui joue le rôle du client auprès de l'équipe)
- ◆ l'Equipe de développeurs (aucune hiérarchie entre eux et une équipe de 6 à 10 personnes pour être efficace)

Les étapes de la méthode Scrum :

- ◆ Le Product Backlog : Le Product Owner rencontre le client et analyse son besoin, il identifie toutes les fonctionnalités dont le produit devra être composé (c'est les user stories). Le Product Backlog n'est pas fixé, il peut évoluer en fonction des besoins du client et de l'avancement du projet.
- ◆ Le Sprint : c'est le tri des fonctionnalités et des tâches d'un projet en forme de cycle qui peut varier de quelques heures en un mois maximum. Pendant un cycle, l'équipe s'occupera uniquement d'une fonctionnalité.
- ◆ Sprint Review : A chaque fois à la fin d'un sprint, on teste les bénéfices de la fonctionnalité avec le Product Owner puis le client confirme si la fonctionnalité marche comme il le souhaitait.

#### 4.3.2 Méthodologie Agile/Kanban



Définition : c'est une méthode de gestion des connaissances relatives au travail, qui met l'accent sur une organisation de type juste-à-temps en fournissant l'information ponctuellement aux membres de l'équipe. Elle emploie un système de tirage de tâches en cours comme mécanisme central afin de déterminer les processus du système et de stimuler la collaboration dans le but d'une amélioration continue du système.

J'ai utilisé les outils Trello et Jira pour gérer et planifier mes tâches :

## Projet Final - Concepteur Développeur d'Applications

C  
E  
S  
A  
R  
  
J  
e  
s  
t  
e  
r

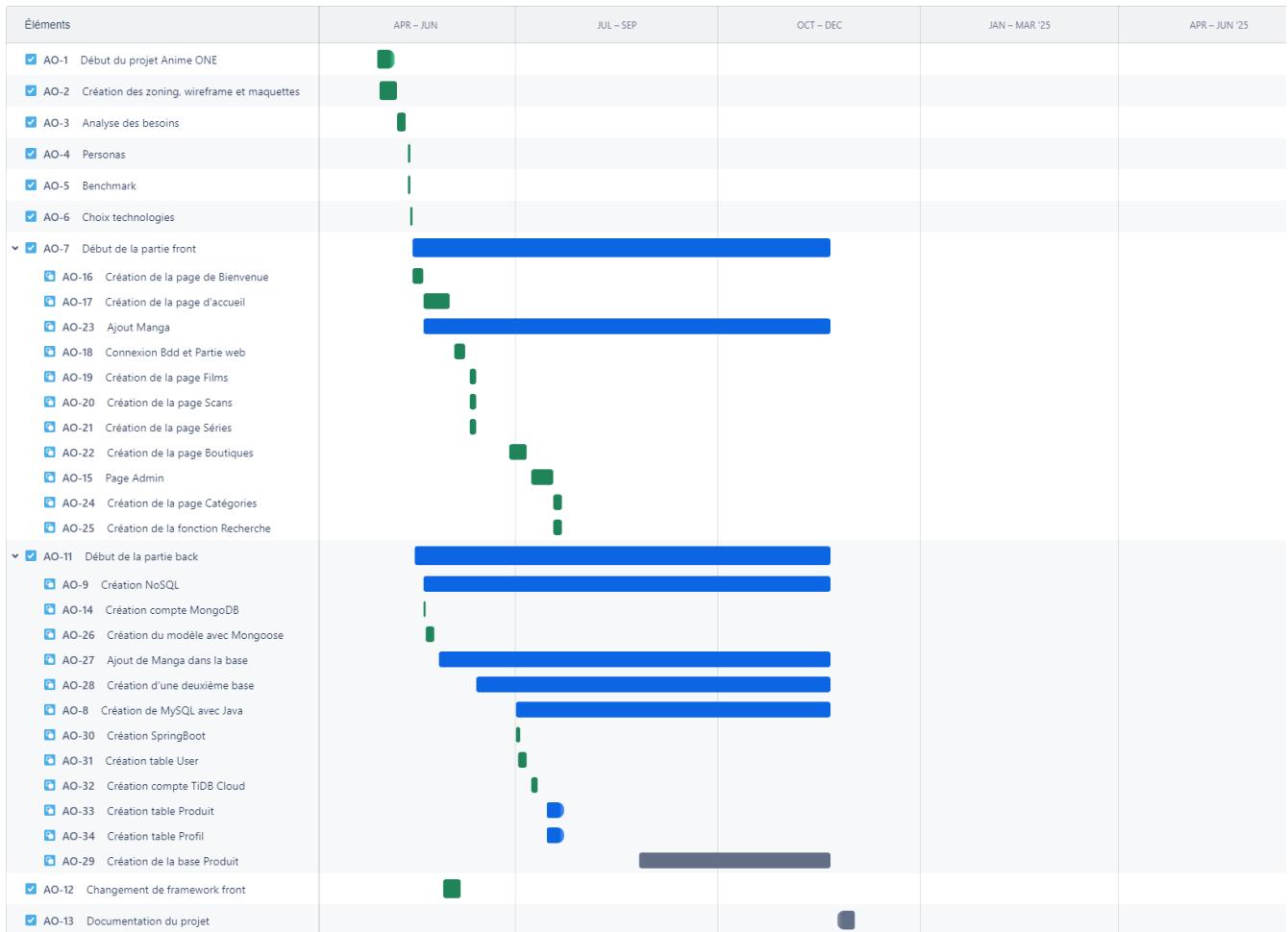
P  
R  
O  
J  
E  
T

F  
I  
N  
A  
L

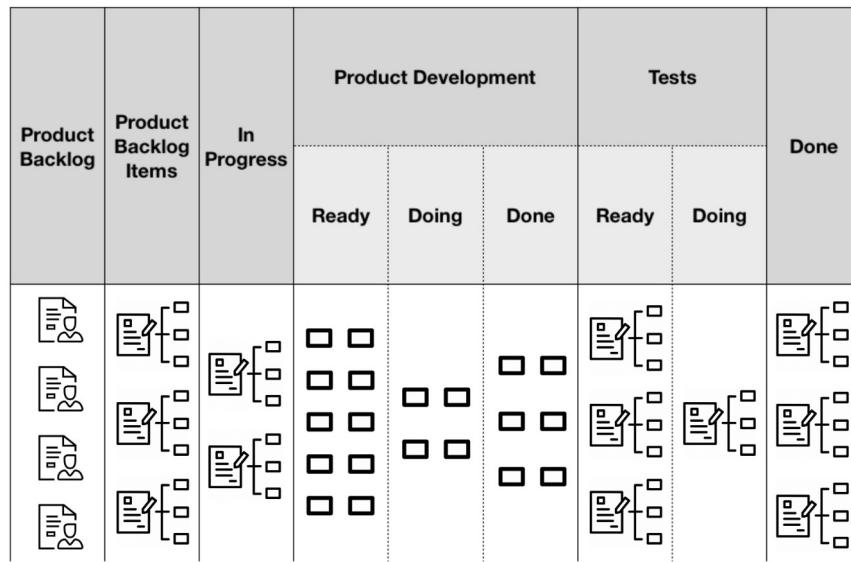
C  
D  
A

The Kanban board displays the following tasks:

- Conception:**
  - Terminer: Création zoning, wireframe, maquette (Due 30 avr. - 7 mai)
  - Terminer: Analyse des besoins (Due 8 mai - 11 mai)
  - Terminer: Personas (Due 13 mai)
  - Terminer: Benchmark (Due 13 mai)
  - Terminer: Choix technologies (Due 14 mai)
  - Terminer: Mise en place (Due 27 mai - 31 mai)
- Front-end:**
  - A Faire: Gestion liste de Visionnage (Due 15 mai - 20 nov.)
  - En cours: Ajout Manga (Due 20 mai - 20 nov.)
  - En cours: Création page Boutique (Due 28 juin - 20 nov.)
  - En cours: Connexion Bdd MySQL et Partie Web (Due 31 juil. - 20 nov.)
  - Terminer: Création page de Bienvenue (Due 15 mai - 19 mai)
  - Terminer: Création page d'accueil (Due 20 mai - 31 mai)
  - Terminer: Connexion Bdd NoSQL et Partie web (Due 21 mai - 24 mai)
- Back-end:**
  - A Faire: Création table Produit
  - En cours: Base NoSQL (Due 15 mai - 20 nov.)
  - En cours: Ajout manga à la base (Due 27 mai - 20 nov.)
  - En cours: Base MySQL (Due 1 juil. - 20 nov.)
  - En cours: Création table Profile
  - Terminer: Création MongoDB (Due 20 mai)
  - Terminer: Création modèles Mongoose (Due 21 mai - 24 mai)
- Documentation:**
  - Terminer: Déposer sur GitHub
  - Terminer: Mise à jour: Déploiement sur Internet (Due 1 déc.)



#### 4.3.3 Méthodologie Agile/Scrumban

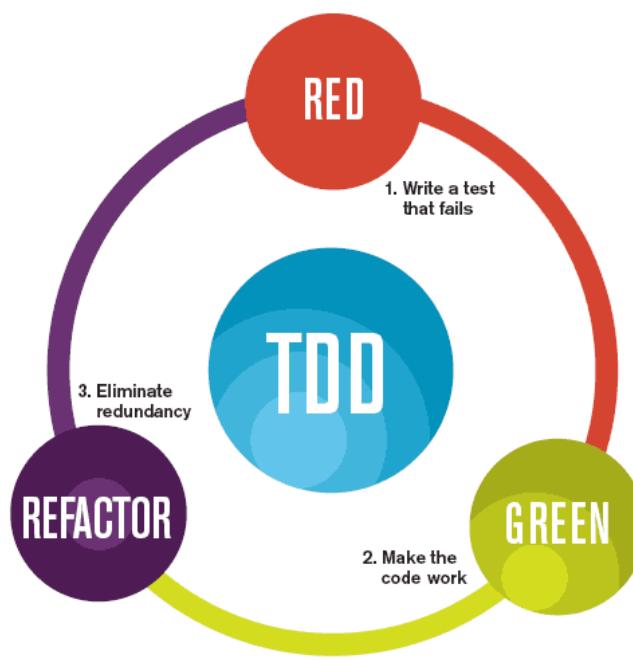


### SCRUMBAN

© G. MOISIO

Définition: c'est une méthode agile hybride qui combine les méthodes Scrum et Kanban. Elle a été développée pour aider les équipes à effectuer la transition de Scrum à Kanban et vice versa mais cette solution est devenue un système qui permet de mener des projets complexes et à long terme. Elle utilise la structure stable de Scrum composée de sprints et ajoute le flux de travail visuel de Kanban ainsi que les limitations du travail en cours.

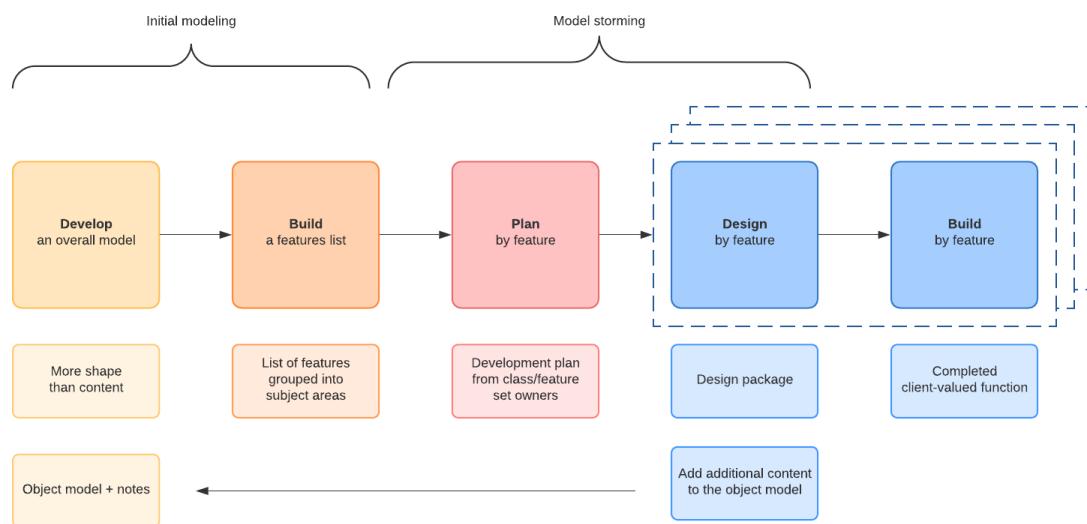
#### 4.3.4 Méthodologie Agile/TDD



The mantra of Test-Driven Development (TDD) is "red, green, refactor."

Définition : la méthode Test-driven Development est une méthodologie de développement logiciel piloté par les tests. Le TDD pousse les développeurs à corriger les bugs en temps réel, au fur et à mesure de la programmation. Cette méthode est basée sur 3 cycle : red, green et refactor

#### 4.3.5 Méthodologie Agile/FDD



Définition : le développement basé sur les fonctionnalités (FDD) est un processus de développement logiciel itératif et incrémental. C'est l'une des méthodes légères en agile pour le développement de logiciels. FDD regroupe un certain nombre de meilleures pratiques reconnues par l'industrie en un tout cohérent. Ces pratiques sont toutes issues d'une perspective de fonctionnalité valorisée par le client. Son objectif principal est de fournir des logiciels de travail tangibles et répétés en temps opportun.

## 5 Spécifications fonctionnelles

Définition : c'est la description des fonctions d'un logiciel en vue de sa réalisation. Elle décrit dans le détail la façon dont les exigences seront prises en compte

### 5.1 Fonctionnalité "S'authentifier"

Description : la fonctionnalité de connexion permet aux utilisateurs inscrits d'accéder à leur compte en fournissant un identifiant et un mot de passe.

### 5.2 Fonctionnalité "S'inscrire"

Description : la fonctionnalité d'inscription permet aux utilisateurs de créer un compte en fournissant des informations personnelles et de connexion.

### 5.3 Fonctionnalité "Ajouter un manga"

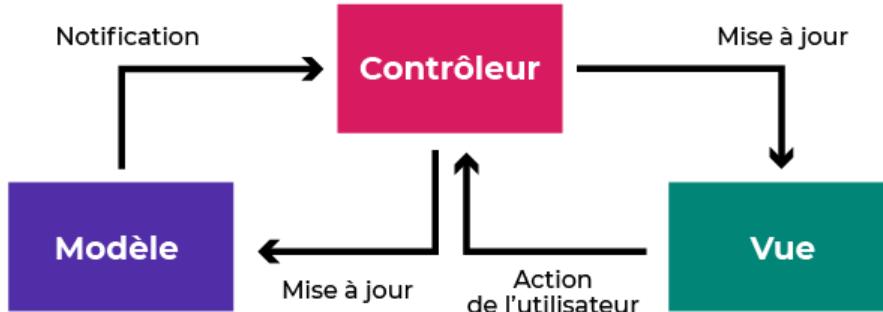
Description : la fonctionnalité d'ajouter un manga permet aux admins d'ajouter des mangas, leurs contenus et toutes les infos dans la base de données pour permettre à l'utilisateur d'avoir du contenu à voir.

### 5.4 Fonctionnalité "Ajouter article au panier"

Description : la fonctionnalité d'ajout d'article au panier permet aux utilisateurs de sélectionner et ajouter des articles à leur panier.

## 6 Spécifications techniques

### 6.1 L'architecture logicielle multicouches MVC



**Définition :** l'architecture Modèle-vue-contrôleur est un motif d'architecture logicielle pour les interfaces graphiques. Il est composé de trois types de modules :

- ◆ un Modèle → il contient les données à afficher
- ◆ une Vue → qui contient la présentation de l'interface graphique
- ◆ un Contrôleur → il contient la logique concernant les actions effectuées par l'utilisateur

## 6.2 L'architecture n-Tiers → (3-Tiers)

**Définition :** l'architecture en couches est un style architectural d'application informatique. C'est apparu quand les progrès technologiques ont permis de séparer logiquement et physiquement les programmes et les bases de données. C'est alors qu'apparaît l'architecture dite 2-tiers avec le client qui gère la présentation et la logique applicative et le serveur qui stocke les données. Plus généralement une architecture multi-tiers va encore plus loin dans le découpage logique et physique du système informatique en répartissant les données sur différents serveurs mais aussi la logique applicative.

J'ai choisi pour mon projet l'architecture 3-tiers qui est séparé en 3 couches :

- ◆ l'interface homme-machine dite IHM → c'est l'interface graphique qui s'exécute sur un navigateur web
- ◆ la couche métier → c'est la partie fonctionnelle de l'application, elle collecte les infos qui proviennent de l'IHM puis traitées et peut soit ajouter, supprimer ou modifier les données envoyées à la couche données
- ◆ la couche données → c'est la base de données où sont stockées et gérées les informations traitées par l'application. Il peut s'agir d'un système de gestion de base de données relationnelle (MySQL) ou non relationnelle (NoSQL)

## 6.3 Technologies choisies

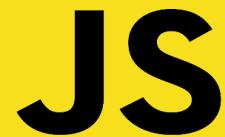
### 6.3.1 Technologies côté Front-end

- Langages et Framework :



Next Js : c'est un framework gratuit et open source s'appuyant sur la bibliothèque JavaScript React et sur la technologie Node Js. J'ai choisi ce framework car il prend en charge le côté serveur (Server Side Rendering), le côté statique d'une page web (Static Site Generation) et l'application web est dynamique.

JavaScript : c'est un langage de programmation de scripts employé dans les pages web. Avec HTML et CSS, JavaScript fait partie des langages les plus utilisés par les développeurs. J'ai choisi ce langage pour ajouter de l'interactivité et des fonctionnalités de l'expérience utilisateur



## HTML



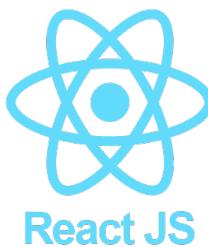
HTML : c'est un langage de balisage conçu pour représenter les pages web. Ce langage permet d'écrire de l'hypertexte, de créer des formulaires, et des programmes informatiques. Il est souvent utilisé conjointement avec JavaScript et CSS. J'ai choisi ce langage car c'est le langage le plus connu pour créer une page web.

## CSS



CSS : c'est un langage informatique de type feuilles de style en cascade qui permet de mettre en forme des documents de types HTML et XML. J'ai choisi ce langage car il est toujours utilisé avec HTML.

- Outils :



React Js : c'est une librairie JavaScript pour créer des interfaces utilisateurs qui peut être utilisée avec un framework MVC comme Next Js. J'ai choisi cette bibliothèque car c'est plus simple à écrire et permet d'utiliser des composants qu'on peut réutiliser.

Tailwind CSS : c'est une bibliothèque CSS open source qui facilite le développeur. J'ai choisi cette librairie car contrairement à Bootstrap, il n'y a pas de classes prédéfinies comme pour les boutons ou les tables. C'est au développeur de choisir ces propres classes et les modifier à sa manière comme sur les feuilles de styles de CSS.



### 6.3.2 Technologies côté Back-end

- Langages et Framework :



Next Js : c'est un framework gratuit et open source s'appuyant sur la bibliothèque JavaScript React et sur la technologie Node Js. J'ai choisi ce framework car il prend en charge le côté serveur (Server Side Rendering), le côté statique d'une page web (Static Site Generation) et l'application web est dynamique.

Node Js : c'est une plateforme logicielle libre en JavaScript qui permet le développement de serveur HTTP. Donc qui est axée sur le côté serveur. J'ai choisi cette plateforme car comme j'utilise la librairie de ReactJS, Node permet de créer des applications web évolutives qui peuvent gérer de grandes quantités de trafic et de connexions simultanées.



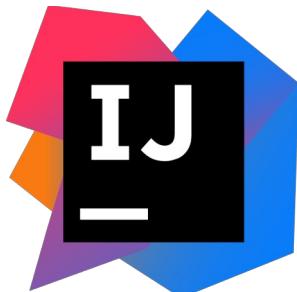
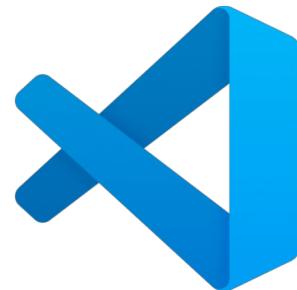
- Outils :



**Prisma**

Prisma : c'est un ORM (Object-Relational Mapping), une bibliothèque permettant de gérer une base de données de manière grandement simplifiée en représentant les tables sous forme d'entités.

Visual Studio Code : c'est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré. J'ai choisi Vs Code car c'est l'un des éditeurs pour coder le plus complet



IntelliJ : c'est un environnement de développement destiné au développement de logiciels informatiques reposant sur la technologie Java. Il est développé par JetBrains et disponibles en deux versions, une version open source et une autre payante. J'ai choisi cet éditeur de code car il est prédéfini pour une application Java donc facilite la création du code avec Springboot.

### 6.3.3 Technologies côté Persistance des données

- Outils :

MySQL Workbench : c'est un logiciel de gestion et d'administration de base de données MySQL. Via une interface graphique intuitive, il permet, entre autres, de créer, modifier ou supprimer des tables, des comptes utilisateurs, et d'effectuer toutes les opérations inhérentes à la gestion d'une base de données. Pour ce faire, il doit être connecté à un serveur MySQL.



MongoDB : c'est un système de gestion de base de données orienté documents en NoSQL. Avec ces applications Atlas et Compass on peut gérer les données depuis n'importe où avec le CRUD (Create Read Update Delete).



### 6.3.4 Technologies transverses

- Outils :

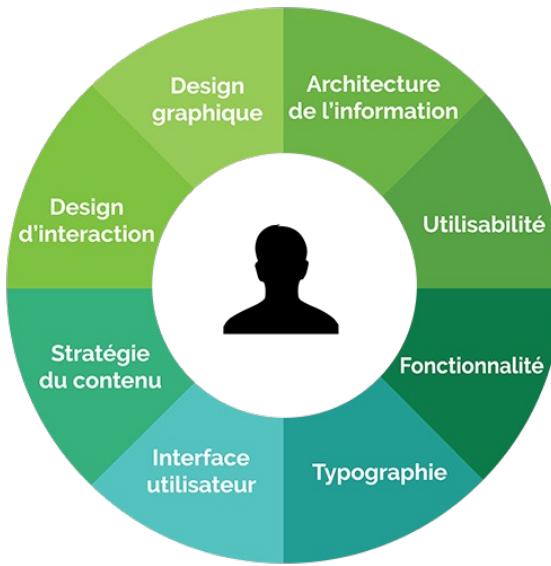
Canva : c'est un outil de conception graphique en ligne. J'ai choisi cette outil pour créer un logo et pour redimensionner mes images afin de les mettre à la même taille. C'est un outil très facile à utiliser fait pour tout type de personne.



Figma : c'est un éditeur de graphiques vectoriels et un outil de prototypage. Ses fonctionnalités sont axées sur l'utilisation dans la conception de l'interface utilisateur et de l'expérience utilisateur, en mettant l'accent sur la collaboration en temps réel. J'ai choisi cet outil pour créer mes différentes maquettes et l'utilisation est plutôt simple.

## 7 Conception de l'interface graphique

### 7.1 Expérience UX/UI



Définition : c'est la qualité du vécu de l'utilisateur dans des environnements numériques. C'est une notion qui prend en compte l'apparence, l'esthétique, le plaisir, l'émotion, etc.

### 7.2 Charte graphique

Mes Différents Logos :



Bannière :

Les couleurs changent dynamiquement en fonction de la page, chaque page à sa propre couleur pour la couleur de fond et pour la barre de navigation

Exemple →

Cyan foncé (#164E63) :



Teal foncé (#134E4A) :



Rouge (#DC2626) :



Jaune foncé (#FBBF24) :

Styles de texte :

J'ai choisi comme police d'écriture Roboto, c'est une police de caractère qui appartient à Google et est de type sans-serif. J'ai choisi cette police car elle est très simple à lire, et dans un site de streaming l'utilisateur vient pour se divertir et non pour se compliquer la tâche en essayant de déchiffrer des mots.

### 7.3 Responsive

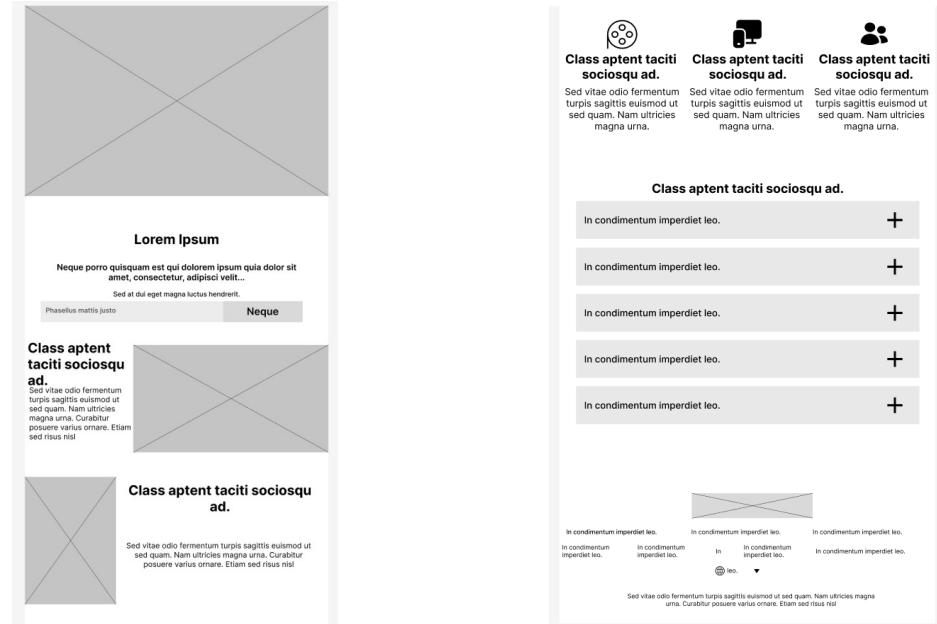
Définition : le mot responsive provient avant tout de « responsive web design ». Cela consiste à rendre accessible et adapter la conception du site web et le rendre confortable sur des écrans de taille différentes. L'utilisateur peut ainsi consulter le même site web à travers une large gamme d'appareils (ordinateurs, smartphone, tablette, ...) tout en gardant le même confort visuel sur les différents appareils.

### 7.4 Zoning

version ordinateur :

page de bienvenue

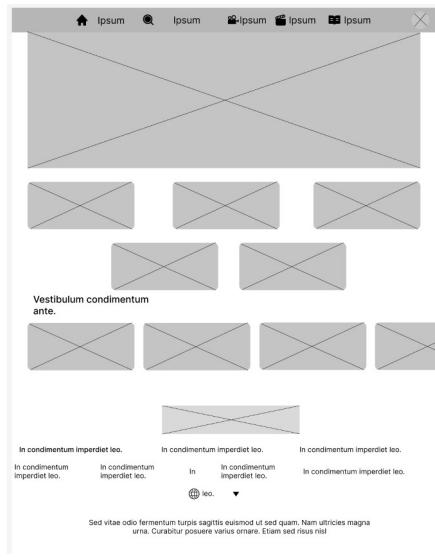
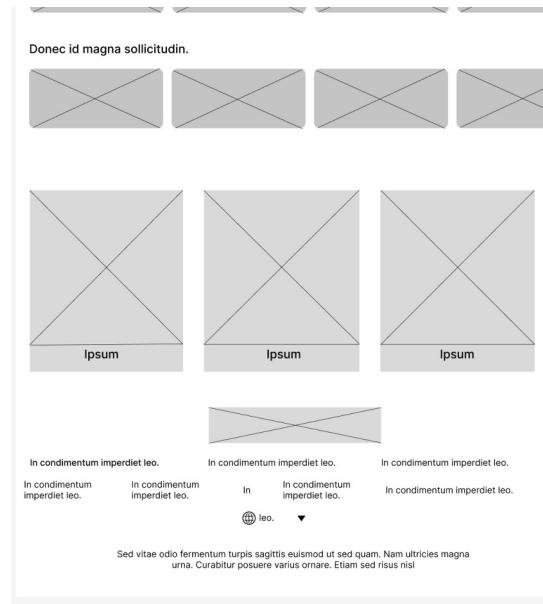
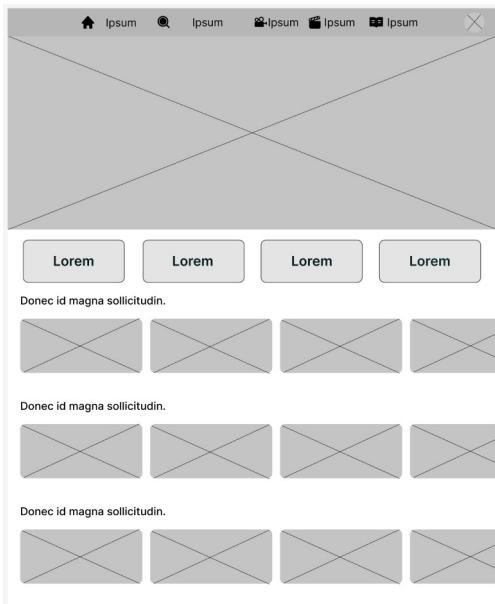
## Projet Final - Concepteur Développeur d'Applications



page d'authentification



page d'accueil

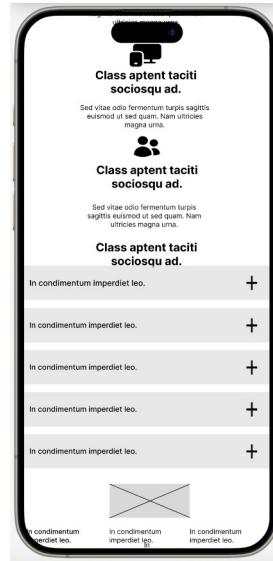


page de manga

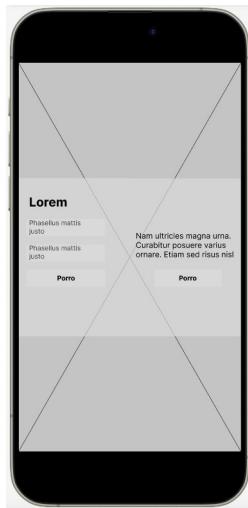
version mobile :

page de bienvenue

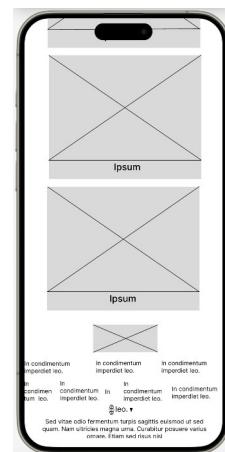
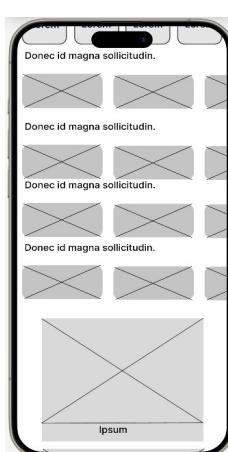
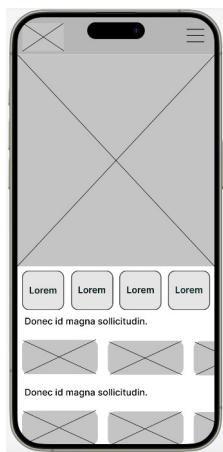
## Projet Final - Concepteur Développeur d'Applications



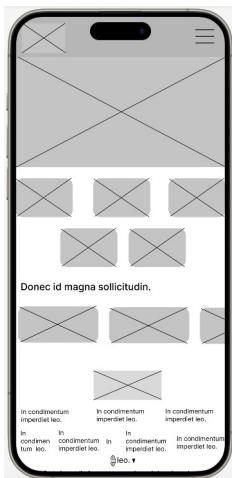
page d'authentification



page d'accueil



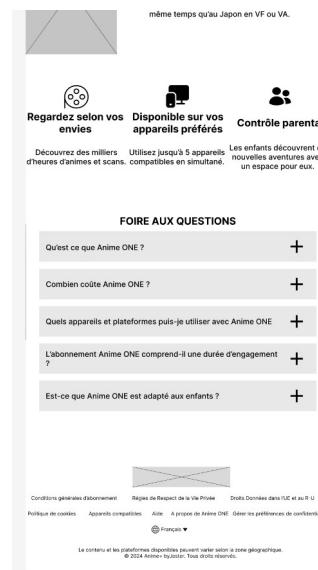
page manga



## 7.5 Wireframes

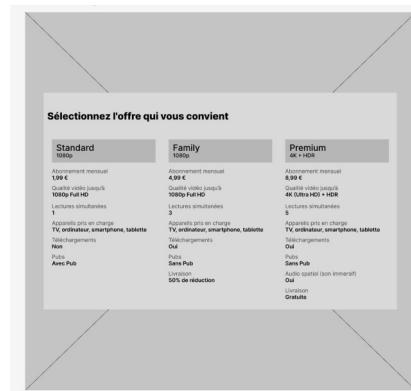
version ordinateur :

page de bienvenue

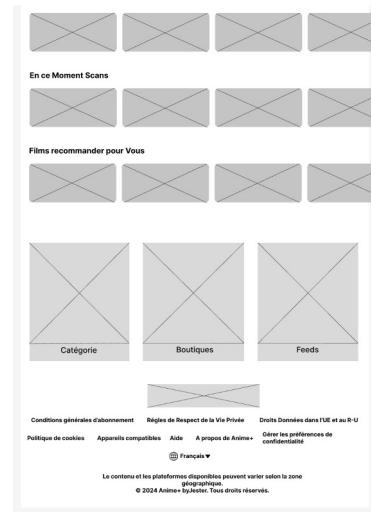
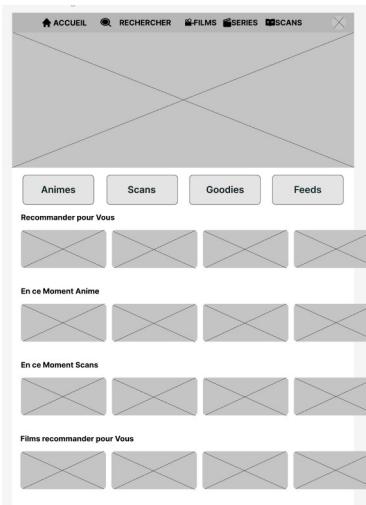


page d'authentification

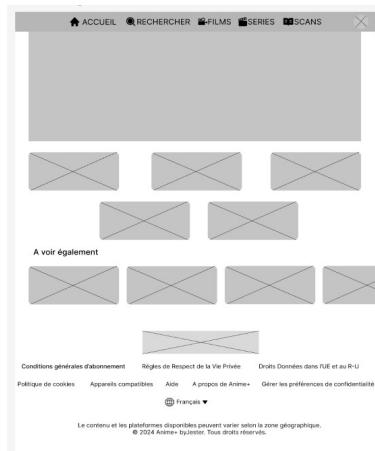
page choix offres



page d'accueil



page manga

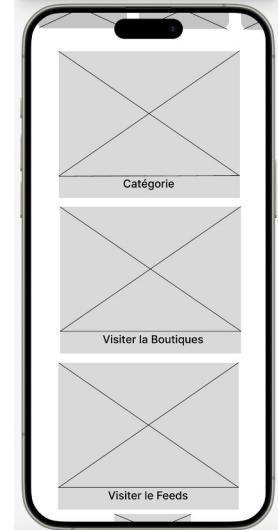
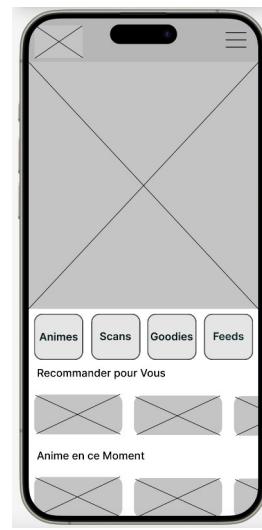


version mobile :

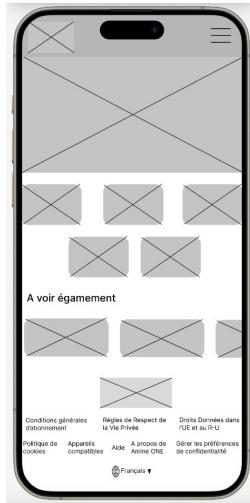
page de bienvenue



page d'authentification



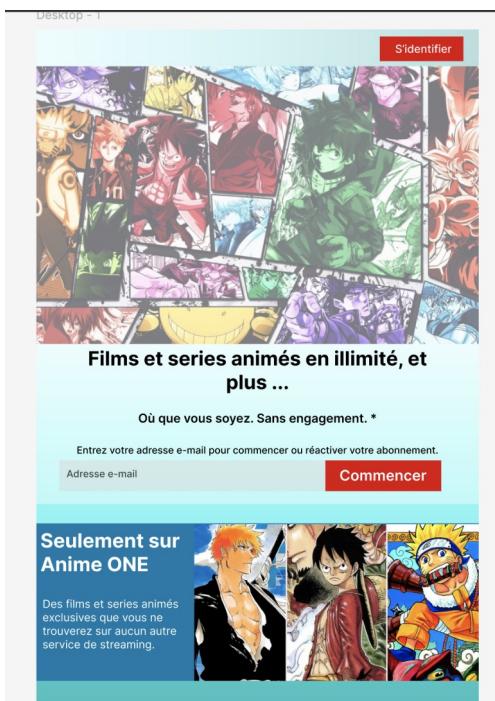
page manga



## 7.6 Maquettes

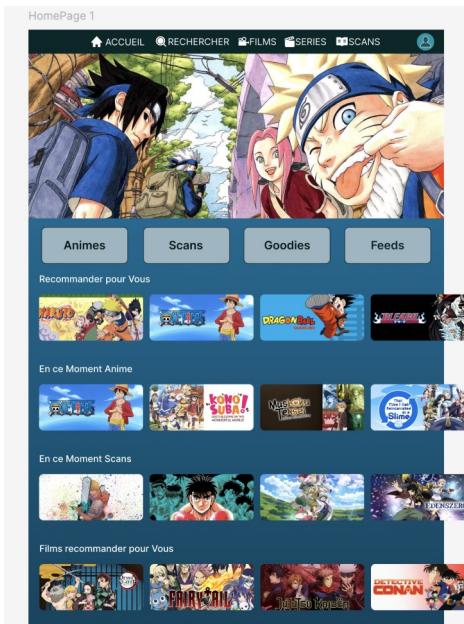
vérions ordinateur :

page de bienvenue

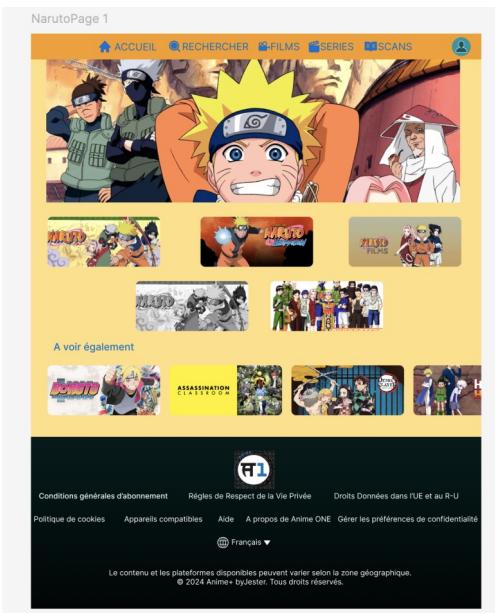


La maquette de la page d'accueil d'Anime ONE est divisée en plusieurs sections. En haut, une bannière bleue avec l'illustration d'un personnage et le texte "Regardez vos scan où que vous soyez". Ensuite, une section "Regardez selon vos envies" avec un icône de regard, une section "Disponible sur vos appareils préférés" avec un icône de smartphone et une section "Contrôle parental" avec un icône de personnes. En bas, une "FOIRE AUX QUESTIONS" avec des questions telles que "Qu'est ce que Anime ONE ?" et "Combien coûte Anime ONE ?". La page contient également des liens utiles comme "Conditions générales d'abonnement", "Politique de cookies", "Français", "Le contenu et les plateformes disponibles peuvent varier selon la zone géographique.", et "© 2024 Anime+ byJester. Tous droits réservés."

page d'accueil

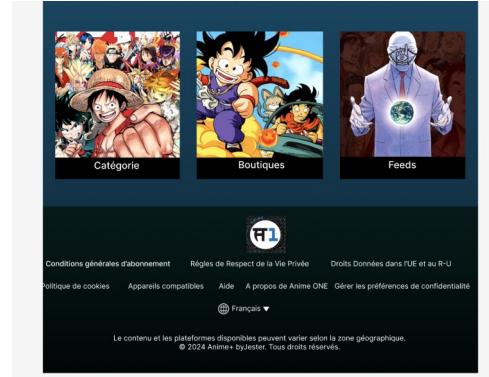


page Naruto

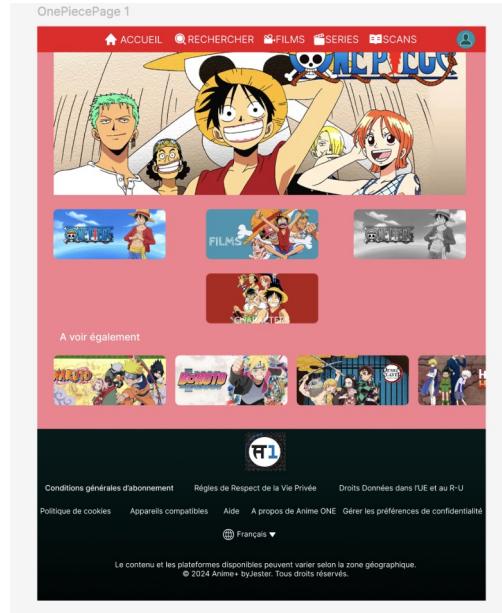


version mobile :

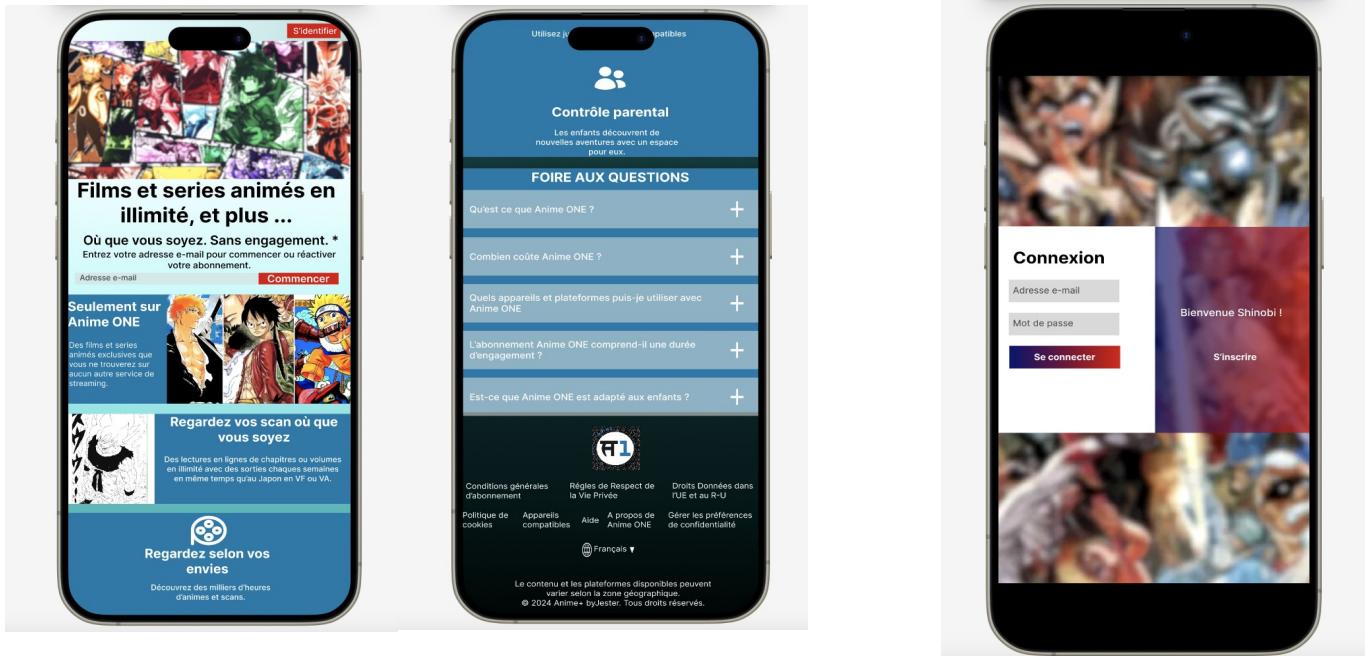
page de bienvenue



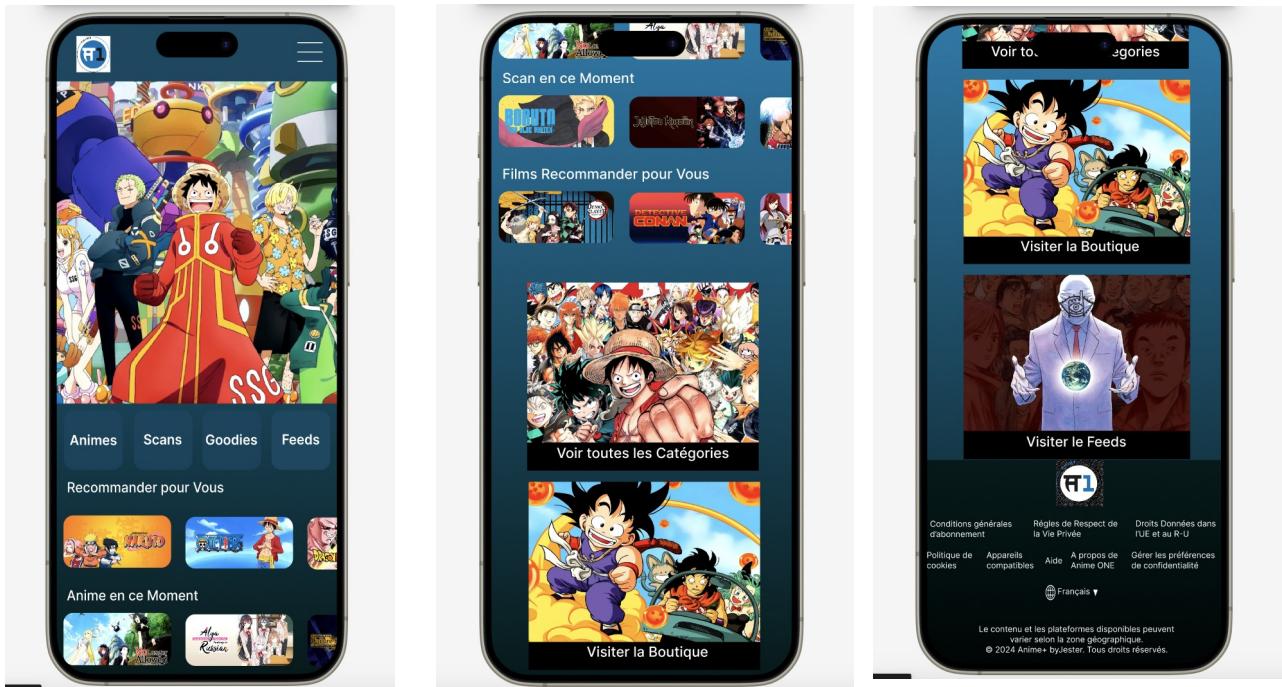
page One Piece



page d'authentification



page d'accueil

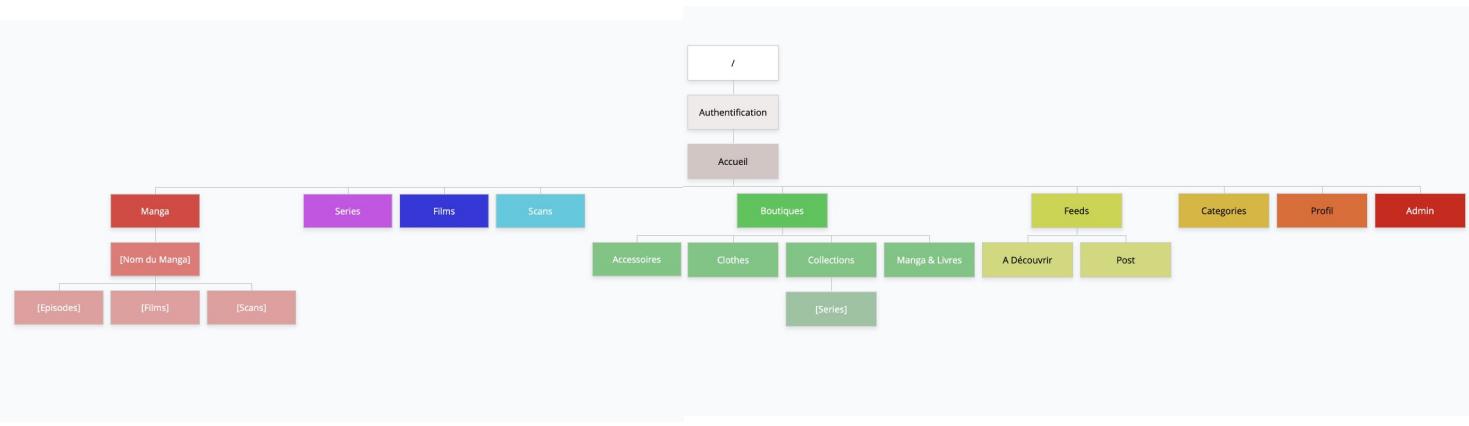


page bleach

page dragon ball



## 7.7 Arborescence du site

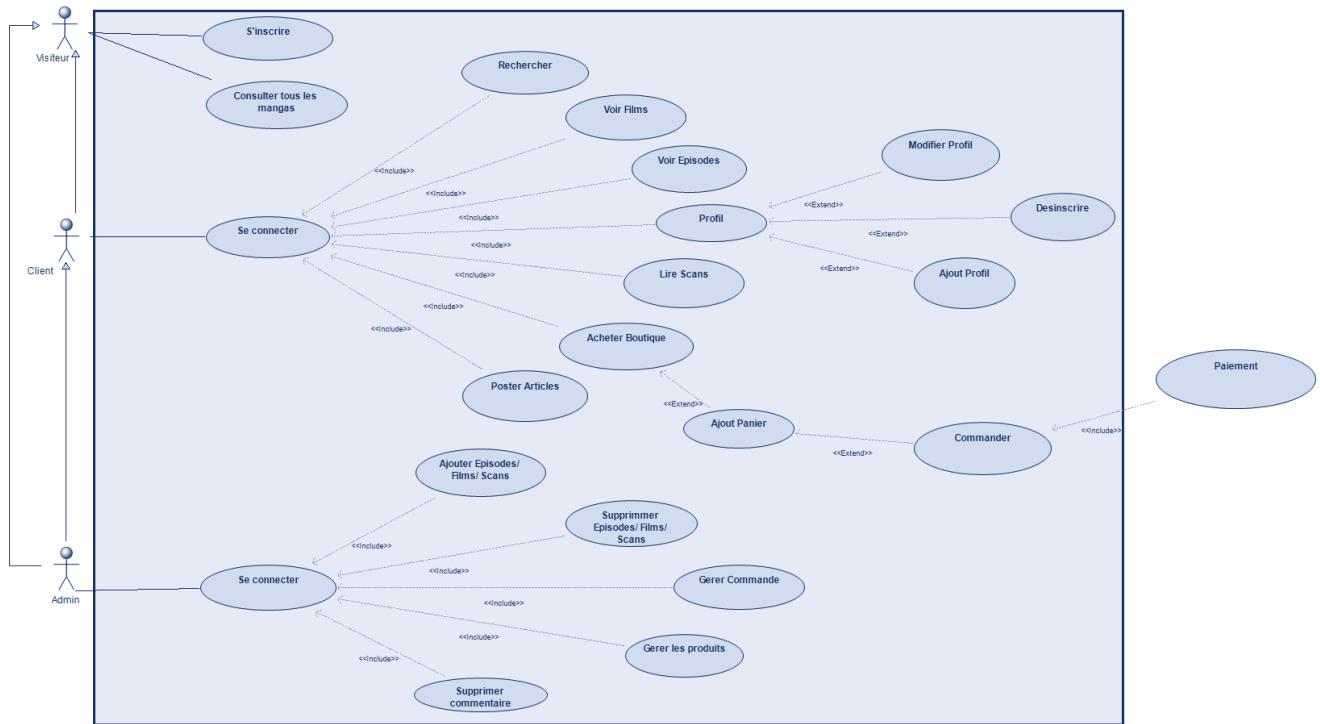


## 8 Conception de l'application

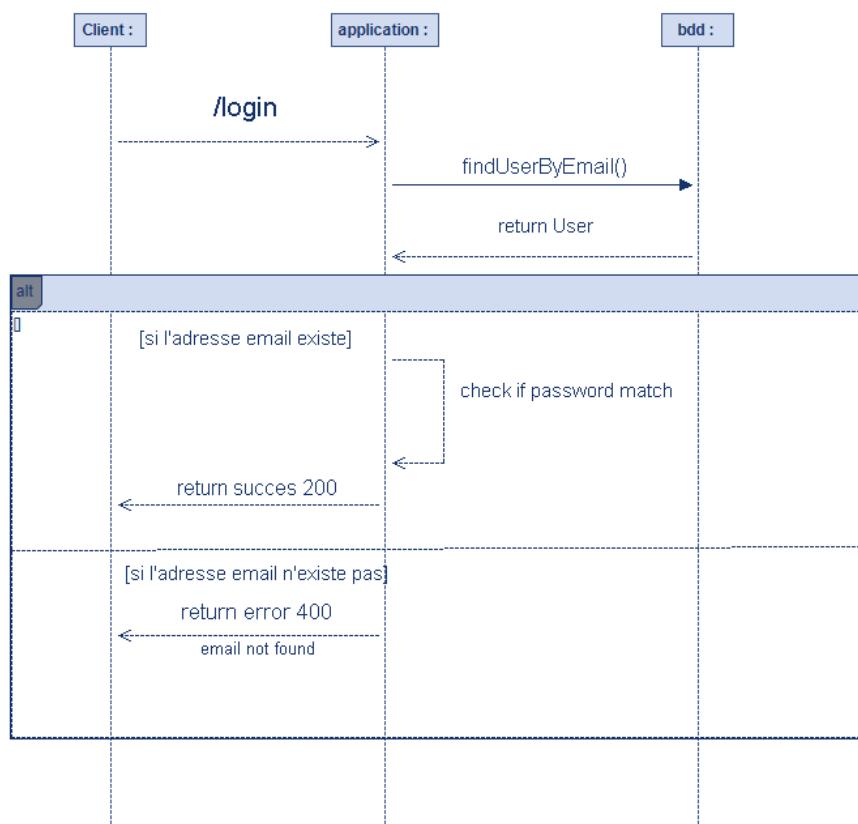
### 8.1 Le langage de modélisation UML

Définition : c'est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation en conception orienté objet.

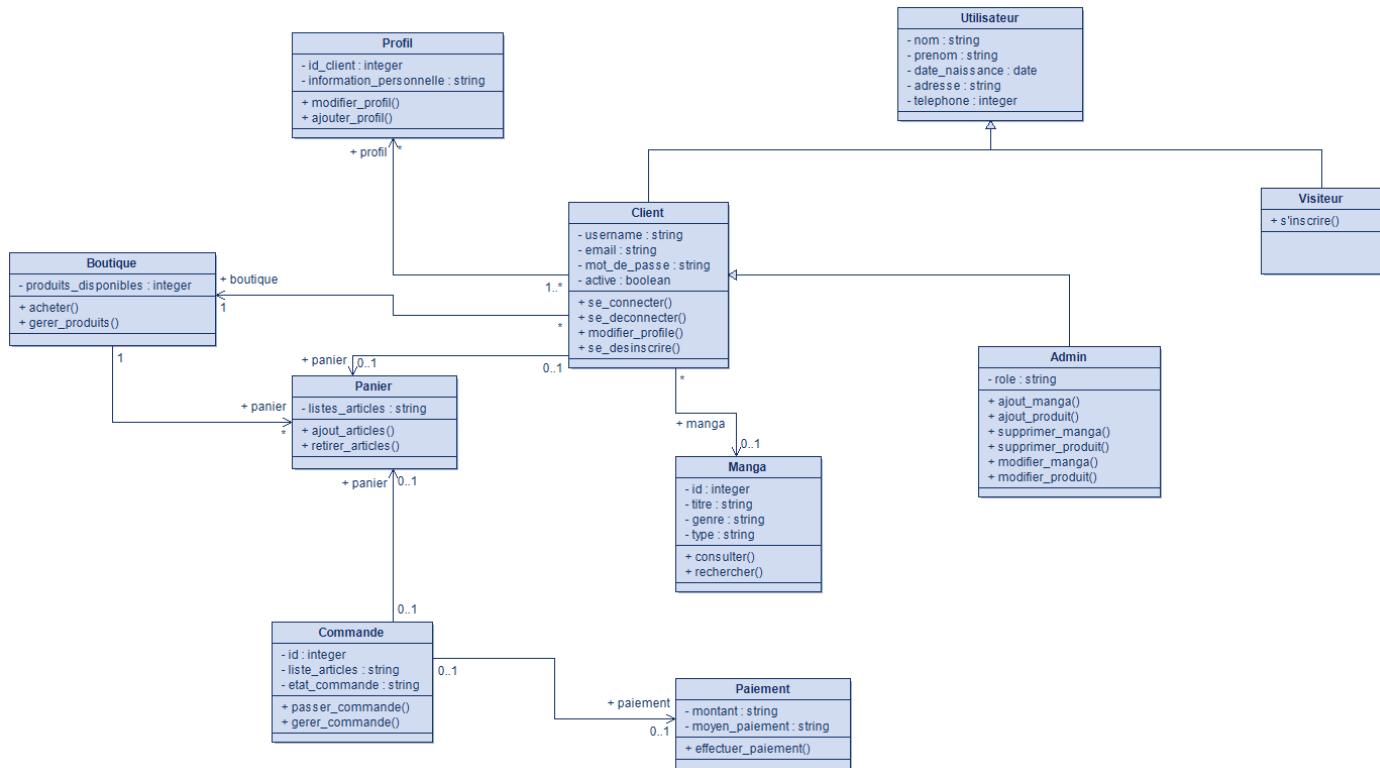
## 8.2 Diagramme de cas d'utilisation



## 8.3 Diagramme de séquences



## 8.4 Diagramme de classes



## 8.5 Définition des Routes

### 8.5.1 Routes des pages

Page	URL
Page de garde	<a href="https://anime-one-project.vercel.app/">https://anime-one-project.vercel.app/</a>
authentification	<a href="https://anime-one-project.vercel.app/authentification">https://anime-one-project.vercel.app/authentification</a>
accueil	<a href="https://anime-one-project.vercel.app/accueil">https://anime-one-project.vercel.app/accueil</a>
manga	<a href="https://anime-one-project.vercel.app/manga">https://anime-one-project.vercel.app/manga</a>
[id]	<a href="https://anime-one-project.vercel.app/manga/[id]">https://anime-one-project.vercel.app/manga/[id]</a>
[animation]	<a href="https://anime-one-project.vercel.app/manga/[id]/[animation]">https://anime-one-project.vercel.app/manga/[id]/[animation]</a>
séries	<a href="https://anime-one-project.vercel.app/series">https://anime-one-project.vercel.app/series</a>
films	<a href="https://anime-one-project.vercel.app/films">https://anime-one-project.vercel.app/films</a>
scans	<a href="https://anime-one-project.vercel.app/scans">https://anime-one-project.vercel.app/scans</a>
catégories	<a href="https://anime-one-project.vercel.app/categories">https://anime-one-project.vercel.app/categories</a>

boutiques	<a href="https://anime-one-project.vercel.app/stores">https://anime-one-project.vercel.app/stores</a>
feeds	<a href="https://anime-one-project.vercel.app/feeds">https://anime-one-project.vercel.app/feeds</a>
profile	<a href="https://anime-one-project.vercel.app/profile">https://anime-one-project.vercel.app/profile</a>
admin	<a href="https://anime-one-project.vercel.app/admin">https://anime-one-project.vercel.app/admin</a>

### 8.5.2 Routes de l'API REST

Page	URL
mangas	/api/mangas
[slug]	/api/mangas/[slug]
[animation]	/api/mangas/[slug]/[animation]
liste	/api/liste
boutiques	/api/boutiques
users	/api/users

## 9 Conception de la Base de données

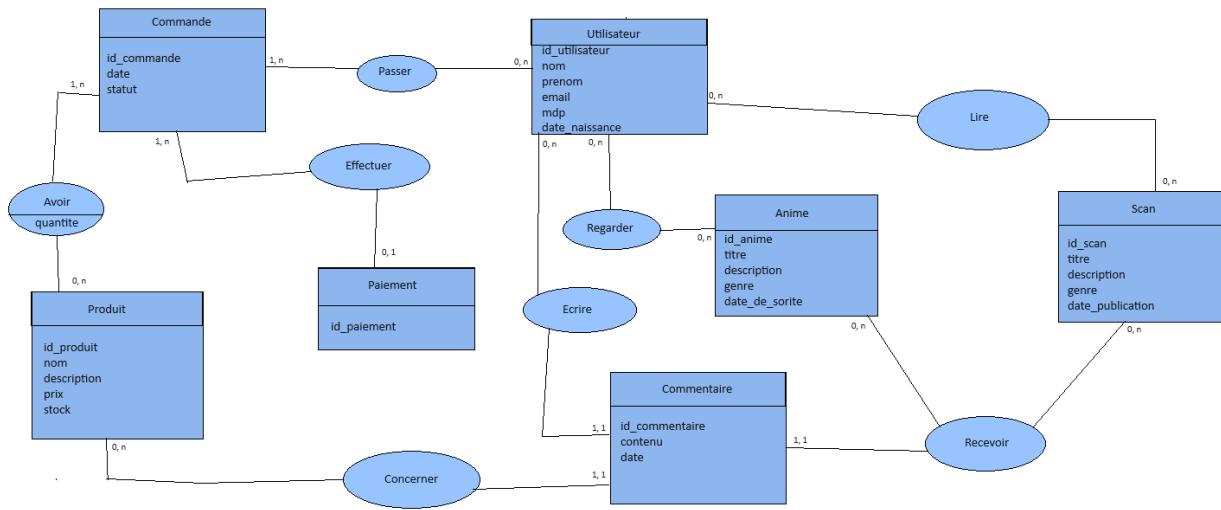
### 9.1 Conception de la Base de données relationnelle MySQL

#### 9.1.1 La méthode MERISE

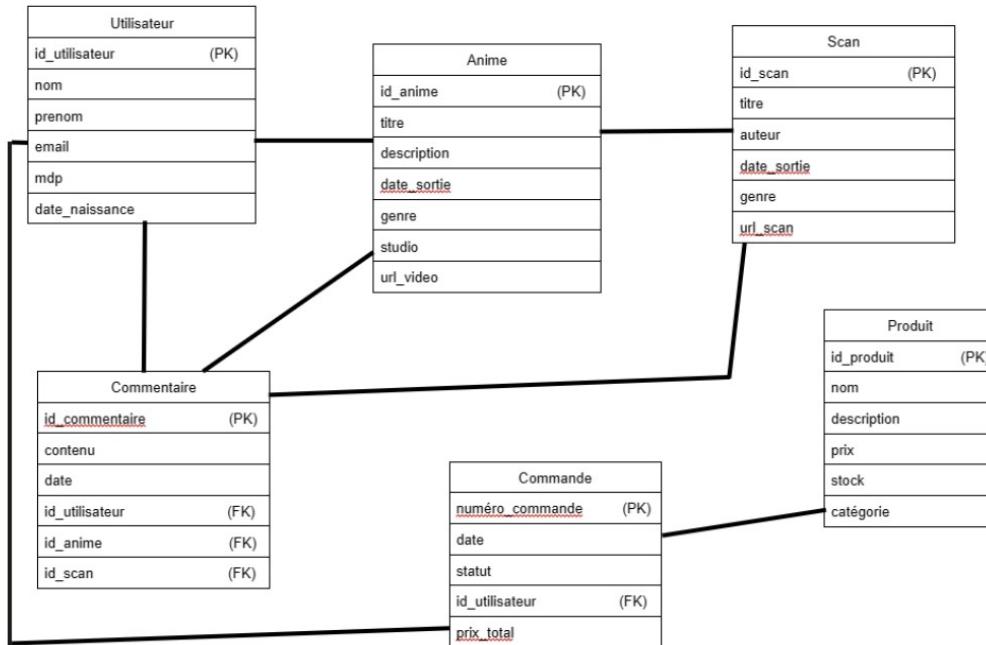
Définition : c'est une méthode d'analyse, de conception et de gestion de projet informatique. C'est une méthode française où on consacre beaucoup de temps à concevoir avant de coder. Voici les différentes étapes de la méthodes MERISE :

- ◆ le Modèle Conceptuel de Données (MCD)
- ◆ le Modèle Logique de Données (MLD)
- ◆ le Modèle Physique de Données (MPD)
- ◆ le Dictionnaire de Données (DD)

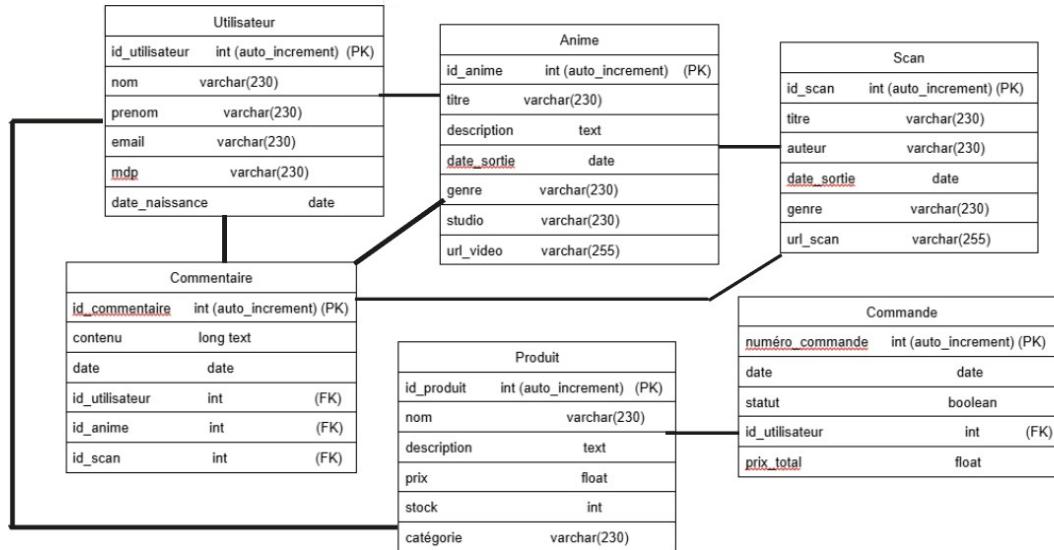
#### 9.1.2 Le Modèle Conceptuel de Données (MCD)



### 9.1.3 Le Modèle Logique de Données (MLD)



## 9.1.4 Le Modèle Physique de Données (MPD)



## 9.1.5 Le Dictionnaire de Données (DD)

### Entité User:

Champ	Type	Taille	Contraintes	Min	Max	Format	Description
id	INT	11	PRIMARY KEY, NOT NULL, AUTO-INCREMENT	1	9999999999	Numérique	L'identifiant de la table
username	VARCHAR	255	NOT NULL	2	255	Alphabétique	L'username du client
prenom	VARCHAR	255	NOT NULL	2	255	Alphabétique	Le prénom du client
nom	VARCHAR	255	NOT NULL	2	255	Alphabétique	Le nom du client
adresse	VARCHAR	512	NULL	2	512	Alphanumérique	L'adresse du client
email	VARCHAR	255	NOT NULL, UNIQUE	2	255	Alphanumérique, Caractère spéciaux	L'email du client
password	VARCHAR	32	NOT NULL	12	32	Alphanumérique, Caractère spéciaux	Le mot de passe du client
phone	VARCHAR	12	NULL	10	12	Numérique	Le numéro de téléphone du client
carte_bancaire	VARCHAR	16	NOT NULL	16	16	Alphanumérique	Le numéro de carte bancaire du client
date_naissance	DATE	8	NOT NULL	8	8	Numérique	La date de naissance du client

role	VARCHAR	32	NOT NULL	2	32	Alphabétique, par défaut client	Le rôle du client
active	BOOLEAN	1	NOT NULL	0	1	1 pour dire active 0 pour dire non active	Si le compte du client est active ou non

## 9.2 Conception de la Base de données non relationnelle NoSQL

### 9.2.1 Paramètre de Connexion

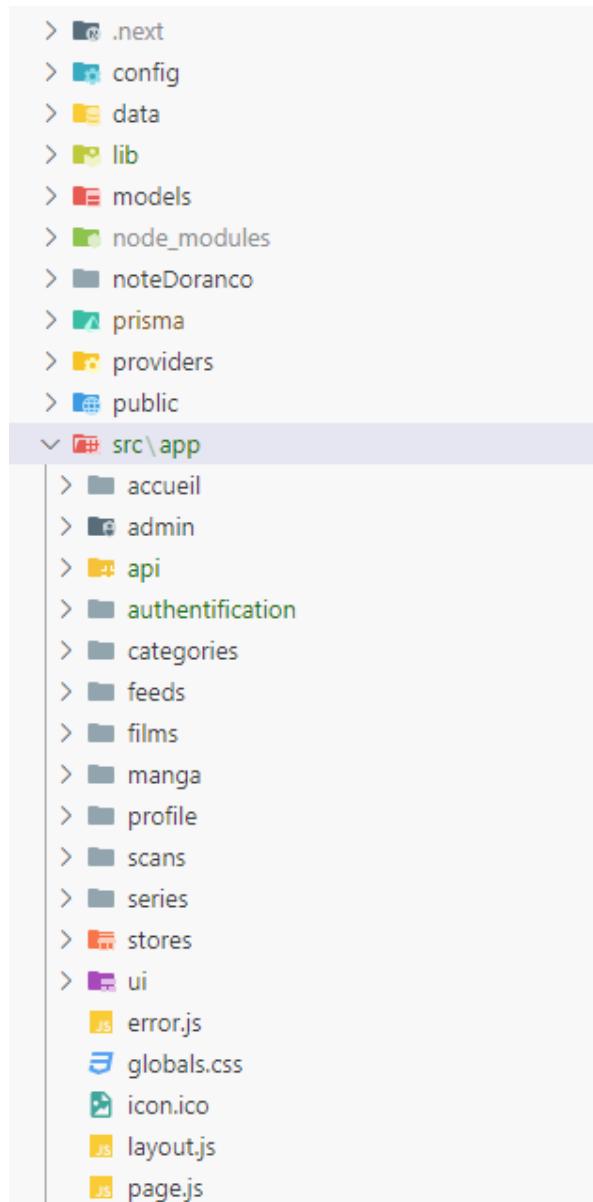
```
config > database.js > connectDB
1 import mongoose from "mongoose";
2 import mysql from "mysql2/promise";
3 import fs from "fs";
4
5 const connectDB = async () => {
6   if (mongoose.connections[0].readyState) {
7     return true;
8   }
9
10 try {
11   await mongoose.connect(process.env.MONGODB_URI, {
12     useNewUrlParser: true,
13     useUnifiedTopology: true,
14     serverSelectionTimeoutMS: 10000,
15   });
16   console.log("MongoDB connecter");
17   return true;
18 } catch (error) {
19   console.log(error);
20 }
21 };
22 }
```

### 9.2.2 Crédit de la Base de Données

```
models > js mangaModel.js > ...
1  'use server'
2  import mongoose, {Schema, models} from "mongoose";
3
4  let MangaModel;
5
6  if (mongoose.models.mangaName) {
7    MangaModel = models.mangaName;
8  } else {
9    const mangaSchema = new Schema({
10      slug: {
11        type: String,
12        unique: true,
13      },
14      imageTop: String,
15      navClass: String,
16      back: String,
17      search: String,
18      listeSearch: String,
19      imageShow: [
20        {
21          img: String,
22          url: String,
23        },
24      ],
25      titre: String,
26      imageCarousel: [
27        {
28          image: String,
29          name: String,
30          url: String,
31        },
32      ],
33      animation: [
34        {
35          animeId: String,
36          anime_imageTop: String,
37          classNav: String,
38          backNav: String,
39          searchNav: String,
40          listeSearchNav: String,
41          imageSee: [{ images: String, href: String }],
42          anime_titre: String,
43          anime_imageCarousel: [
44            { anime_image: String, anime_name: String, anime_href: String },
45          ],
46        },
47      ],
48    }, { timestamps: true });
49    MangaModel = mongoose.model("mangaName", mangaSchema);
50  }
51
52  export default MangaModel;
53
```

## 10 Développement, Intégration et Réalisations

### 10.1 Structure du projet



## 10.2 Fonctionnalité

### 10.2.1 Fonctionnalité « Ajouter un manga »

## Projet Final – Concepteur Développeur d'Applications

```

src > app > api > mangas > route.js > POST
  1 "use server";
  2 import { NextResponse } from "next/server";
  3 import connectDB from "../../../../../config/database";
  4 import MangaModel from "../../../../../models/mangaModel";
  5
  6 export async function POST(request) {
  7   await connectDB();
  8
  9   const data = await request.json();
 10   console.log("Données reçus:", data);
 11
 12   const nouveauManga = new MangaModel(data);
 13
 14   try {
 15     const savedManga = await nouveauManga.save();
 16     console.log("Nouveau Manga Crée:", savedManga);
 17     return NextResponse.json(
 18       { message: "Nouveau Manga Crée" },
 19       { status: 201 }
 20     );
 21   } catch (error) {
 22     console.error("Erreur lors de l'ajout du document:", error);
 23     return NextResponse.json(
 24       { message: "Erreur lors de l'ajout du document" },
 25       { status: 500 }
 26     );
 27   }
 28 }
 29

```

## 10.2.2 Fonctionnalité « Supprimer un manga »

```

src > app > api > mangas > [slug] > route.js > DELETE
  43
  44   export async function DELETE(request, { params }) {
  45     const { slug } = params;
  46     await connectDB();
  47     const deletedManga = await MangaModel.findOneAndDelete({ slug });
  48     if (deletedManga) {
  49       return NextResponse.json({ message: "Manga supprimé" }, { status: 200 });
  50     } else {
  51       return NextResponse.json({ message: "Manga non trouvé" }, { status: 404 });
  52     }
  53   }
  54

```

## 10.3 Requêtes SQL

### 10.3.1 Requête SQL de lecture

SELECT \* from user WHERE prenom = 'Test'

<b>id</b>	<b>username</b>	<b>prenom</b>	<b>nom</b>	<b>adresse</b>	<b>password</b>	<b>email</b>	<b>phone</b>	<b>carte</b>	<b>date_birth</b>	<b>rôle</b>	<b>active</b>
3	Testing	Test	Ing	10 rue de Test	T3st_1N12g@	test@gmail.com	0600123456	1234 5678 9123 4567	01/01/2000	client	1

### 10.3.2 Requêtes SQL d'écriture

```
INSERT INTO 'profile' ('userId', 'nom', 'avatar') VALUES ('3', 'Test2', '/assets/avatar/img.webp')
```