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 7](#__RefHeading___Toc1813_1098725337)

[1 Introduction 7](#__RefHeading___Toc1815_1098725337)

[2 Contexte du projet 7](#__RefHeading___Toc1817_1098725337)

[3 Cahier des charges 7](#__RefHeading___Toc1819_1098725337)

[3.1 Benchmark 7](#__RefHeading___Toc1821_1098725337)

[3.2 Persona 8](#__RefHeading___Toc1823_1098725337)

[3.3 Expression & analyse des besoins 8](#__RefHeading___Toc1825_1098725337)

[3.4 Exigences technico-fonctionnelles 8](#__RefHeading___Toc1827_1098725337)

[3.5 MVP (V. 1.0) 9](#__RefHeading___Toc1829_1098725337)

[3.6 Version 2.0 9](#__RefHeading___Toc1831_1098725337)

[3.7 Version 3.0 9](#__RefHeading___Toc1833_1098725337)

[4 Méthodologie de travail 9](#__RefHeading___Toc1835_1098725337)

[4.1 Cycle en Spirale 9](#__RefHeading___Toc1837_1098725337)

[4.2 Cycle en V 9](#__RefHeading___Toc1839_1098725337)

[4.3 Méthodologie Agile/Scrum 10](#__RefHeading___Toc1841_1098725337)

[4.4 Méthodologie Agile/Kanban 10](#__RefHeading___Toc1843_1098725337)

[4.5 Méthodologie Agile/Scrumban 10](#__RefHeading___Toc1845_1098725337)

[4.6 Méthodologie Agile/TDD 10](#__RefHeading___Toc1847_1098725337)

[4.7 Méthodologie Agile/FDD 10](#__RefHeading___Toc1849_1098725337)

[5 Spécifications fonctionnelles 10](#__RefHeading___Toc1851_1098725337)

[5.1 Fonctionnalité "S'authentifier" 11](#__RefHeading___Toc1853_1098725337)

[5.2 Fonctionnalité "S'inscrire" 11](#__RefHeading___Toc1855_1098725337)

[5.3 Fonctionnalité "Payer par CB" 11](#__RefHeading___Toc1857_1098725337)

[5.4 Fonctionnalité "Ajouter article au panier" 11](#__RefHeading___Toc1859_1098725337)

[6 Spécifications techniques 11](#__RefHeading___Toc1861_1098725337)

[6.1 L'architecture logicielle multicouches MVC 11](#__RefHeading___Toc1863_1098725337)

[6.2 L'architecture n-Tiers  (3-Tiers) 11](#__RefHeading___Toc1865_1098725337)

[6.3 Technologies choisies 11](#__RefHeading___Toc1867_1098725337)

[6.3.1 Technologies côté Front-end 11](#__RefHeading___Toc1869_1098725337)

[6.3.2 Technologies côté Back-end 13](#__RefHeading___Toc1871_1098725337)

[6.3.3 Technologies côté Persistance des données 14](#__RefHeading___Toc1873_1098725337)

[6.3.4 Technologies transverses 14](#__RefHeading___Toc1875_1098725337)

[7 Conception de l'interface graphique 15](#__RefHeading___Toc1877_1098725337)

[7.1 Expérience UX/UI 15](#__RefHeading___Toc1883_1098725337)

[7.2 Charte graphique 15](#__RefHeading___Toc1885_1098725337)

[7.3 Responsive 16](#__RefHeading___Toc1887_1098725337)

[7.4 Zoning 16](#__RefHeading___Toc1889_1098725337)

[7.5 Wireframes 16](#__RefHeading___Toc1891_1098725337)

[7.6 Maquettes 16](#__RefHeading___Toc1893_1098725337)

[7.7 Arborescence du site 16](#__RefHeading___Toc1895_1098725337)

[8 Conception de l'application 17](#__RefHeading___Toc1897_1098725337)

[8.1 Le langage de modélisation UML 17](#__RefHeading___Toc1899_1098725337)

[8.2 Diagramme de cas d'utilisation 17](#__RefHeading___Toc1901_1098725337)

[8.3 Diagramme de séquences 17](#__RefHeading___Toc1903_1098725337)

[8.4 Diagramme d’activité 17](#__RefHeading___Toc2212_3243764837)

[8.5 Diagramme de packages 17](#__RefHeading___Toc2214_3243764837)

[8.6 Diagramme de classes 17](#__RefHeading___Toc1907_1098725337)

[8.7 Diagramme des entités 17](#__RefHeading___Toc2216_3243764837)

[8.8 Définition des Routes 17](#__RefHeading___Toc2218_3243764837)

[8.8.1 Routes des pages 17](#__RefHeading___Toc2220_3243764837)

[8.8.2 Routes de l’API REST 18](#__RefHeading___Toc2222_3243764837)

[9 Conception de la Base de données 18](#__RefHeading___Toc1911_1098725337)

[9.1 Conception de la Base de données relationnelle MySQL 18](#__RefHeading___Toc2224_3243764837)

[9.1.1 La méthode MERISE 18](#__RefHeading___Toc2226_3243764837)

[9.1.2 Le Modèle Conceptuel de Données (MCD) 18](#__RefHeading___Toc2228_3243764837)

[9.1.3 Le Modèle Logique de Données (MLD) 18](#__RefHeading___Toc2230_3243764837)

[9.1.4 Le Modèle Physique de Données (MPD) 18](#__RefHeading___Toc2232_3243764837)

[9.1.5 Le Dictionnaire de Données (DD) 18](#__RefHeading___Toc2234_3243764837)

[9.2 Conception de la Base de données non relationnelle NoSQL 18](#__RefHeading___Toc2236_3243764837)

[9.2.1 Paramètre de Connexion 18](#__RefHeading___Toc2238_3243764837)

[9.2.2 Création de la Base de Données 18](#__RefHeading___Toc2240_3243764837)

[9.2.3 Le Dictionnaire de Données (DD) 18](#__RefHeading___Toc2242_3243764837)

[10 Développement, Intégration et Réalisations 18](#__RefHeading___Toc1923_1098725337)

[10.1 Structure du projet 18](#__RefHeading___Toc1925_1098725337)

[10.2 Fonctionnalité 19](#__RefHeading___Toc1927_1098725337)

[10.2.1 Fonctionnalité « se connecter » 19](#__RefHeading___Toc2244_3243764837)

[10.2.2 Fonctionnalité « s'inscrire » 19](#__RefHeading___Toc1929_1098725337)

[10.2.3 Fonctionnalité « ajouter un manga » Fonctionnalité «  19](#__RefHeading___Toc1933_1098725337)

[10.3 Création de la base de données SQL 19](#__RefHeading___Toc2246_3243764837)

[10.4 Création de la base de données NoSQL 19](#__RefHeading___Toc2248_3243764837)

[10.5 Requêtes SQL et NoSQL 19](#__RefHeading___Toc2250_3243764837)

[10.5.1 Requête SQL de lecture 19](#__RefHeading___Toc2252_3243764837)

[10.5.2 Requêtes SQL d’écriture 19](#__RefHeading___Toc2254_3243764837)

[10.5.3 Requêtes SQL de jointures 19](#__RefHeading___Toc2256_3243764837)

[10.5.3.1 Jointure interne (JOIN ou INNER JOIN) 19](#__RefHeading___Toc2258_3243764837)

[10.5.3.2 Jointure à gauche (LEFT JOIN) 19](#__RefHeading___Toc2260_3243764837)

[10.5.3.3 Jointure à droite (RIGHT JOIN) 19](#__RefHeading___Toc2262_3243764837)

[11 Sécurité de l'application et des données 19](#__RefHeading___Toc1937_1098725337)

[11.1 Les différentes attaques de site 19](#__RefHeading___Toc1939_1098725337)

[11.1.1 Les attaques XSS 19](#__RefHeading___Toc1941_1098725337)

[11.1.2 Les attaques CSRF 19](#__RefHeading___Toc1943_1098725337)

[11.1.3 Les attaques CROS 20](#__RefHeading___Toc1945_1098725337)

[11.1.4 Les attaques Force Brute 20](#__RefHeading___Toc1947_1098725337)

[11.1.5 Les attaques de type injection SQL 20](#__RefHeading___Toc1949_1098725337)

[11.1.6 Authentification/Autorisation avec JWT 20](#__RefHeading___Toc1951_1098725337)

[11.1.7 Authentification Duo/Trio/Quad 20](#__RefHeading___Toc1953_1098725337)

[11.2 HTTPS 20](#__RefHeading___Toc1955_1098725337)

[11.3 La loi RGPD 20](#__RefHeading___Toc1957_1098725337)

[11.3.1 Confidentialité des données 20](#__RefHeading___Toc1959_1098725337)

[11.4 Mentions Légales 20](#__RefHeading___Toc1961_1098725337)

[11.4.1 Mentions Légales 20](#__RefHeading___Toc1963_1098725337)

[11.4.2 Conditions Générales d'utilisation du site web 20](#__RefHeading___Toc1965_1098725337)

[11.4.3 Conditions Générales de vente (sites de commerce ou de service) 20](#__RefHeading___Toc1967_1098725337)

[11.4.4 Politique de confidentialité 20](#__RefHeading___Toc1969_1098725337)

[11.4.5 Cookies 20](#__RefHeading___Toc1971_1098725337)

[11.4.6 Droit de rétractation 20](#__RefHeading___Toc1973_1098725337)

[12 Sécurité de la base de données 20](#__RefHeading___Toc1975_1098725337)

[12.1 Choix des noms de user et des password 20](#__RefHeading___Toc1977_1098725337)

[12.2 Limiter les droits au user 20](#__RefHeading___Toc1979_1098725337)

[12.3 Limiter l'accès à la base de données 21](#__RefHeading___Toc1981_1098725337)

[13 Sécurité et qualité du code 21](#__RefHeading___Toc1983_1098725337)

[13.1 Qualité du code 21](#__RefHeading___Toc1985_1098725337)

[13.1.1 Convention de nommage 21](#__RefHeading___Toc1987_1098725337)

[13.1.2 Conventions de codage 21](#__RefHeading___Toc1989_1098725337)

[13.1.3 Journalisation (LOG) 21](#__RefHeading___Toc1991_1098725337)

[13.1.4 Commentaires sur les lignes de code 21](#__RefHeading___Toc1993_1098725337)

[13.2 Sécurité du code 21](#__RefHeading___Toc1995_1098725337)

[13.2.1 Tests unitaires et couverture de code 21](#__RefHeading___Toc1997_1098725337)

[13.2.2 Gestion des erreurs et des exceptions 21](#__RefHeading___Toc1999_1098725337)

[13.2.3 Fermeture correcte des flux de données 21](#__RefHeading___Toc2001_1098725337)

[14 DevOps 21](#__RefHeading___Toc2003_1098725337)

[15 Déploiement et hébergement 21](#__RefHeading___Toc4431_1290156777)

[16 Référencement SEO 21](#__RefHeading___Toc2005_1098725337)

[17 Veilles technologiques 22](#__RefHeading___Toc2007_1098725337)

[17.1 Veille concernant le développement 22](#__RefHeading___Toc2011_1098725337)

[17.2 Évolutions des technologies 22](#__RefHeading___Toc2013_1098725337)

[17.3 Veille sur la sécurité des applications 22](#__RefHeading___Toc4433_1290156777)

[17.3.1 Les différentes failles de sécurité les plus dangereuses 22](#__RefHeading___Toc2015_1098725337)

[17.3.2 Les recommandations de l'ANSSI (https://cyber.gouv.fr/) 22](#__RefHeading___Toc2017_1098725337)

[17.3.3 La CNIL et ses recommandations 22](#__RefHeading___Toc2019_1098725337)

[17.4 Veille sur la sécurité des bases de données 22](#__RefHeading___Toc2021_1098725337)

[18 Perspectives 23](#__RefHeading___Toc2023_1098725337)

[19 Conclusion 23](#__RefHeading___Toc2025_1098725337)

[Annexe 24](#__RefHeading___Toc2027_1098725337)

# **Résumé / Abstract**

# **Introduction**

Je m’appel Jester CESAR, j’ai 24 ans. 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 que je voulais faire qui a été très intéressante, j’ai beaucoup appris et elle m’a redonnée envie de continuer mes études.

# **Contexte du projet**

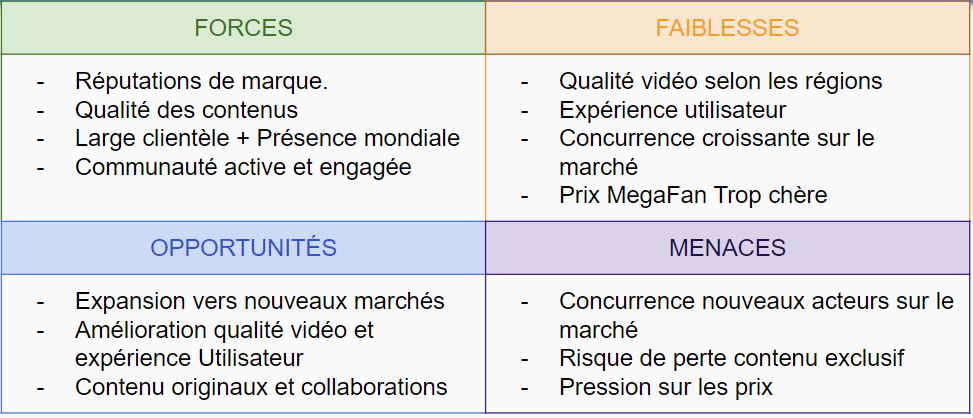
(problématique, constat ...)

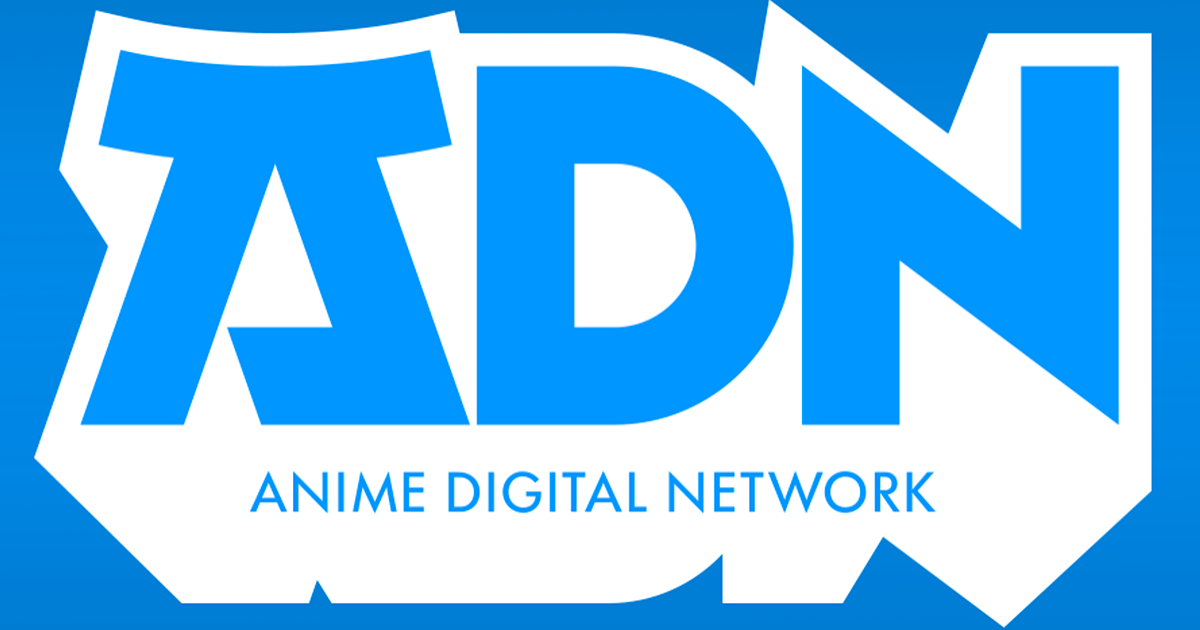
# **Cahier des charges**

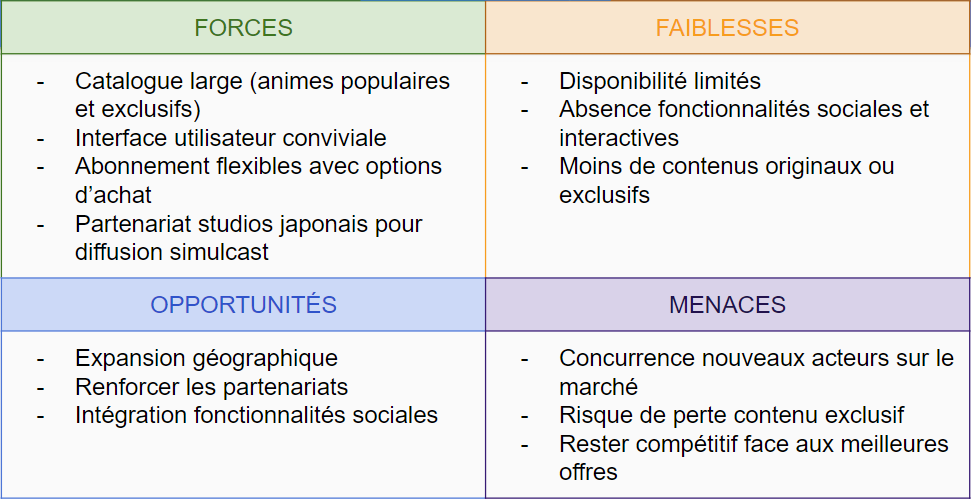
## Benchmark

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 :







## Persona

## Expression & analyse des besoins

Liste ou tableau des fonctionnalités par profil

## Exigences technico-fonctionnelles

Navigateurs, Sécurité des données (Respect de la loi RGPD, Respect des recommandations de la CNIL, Mentions légales, CGU, Cookies, Droit de rétractation), se prémunir et se protéger des différentes attaques de site.

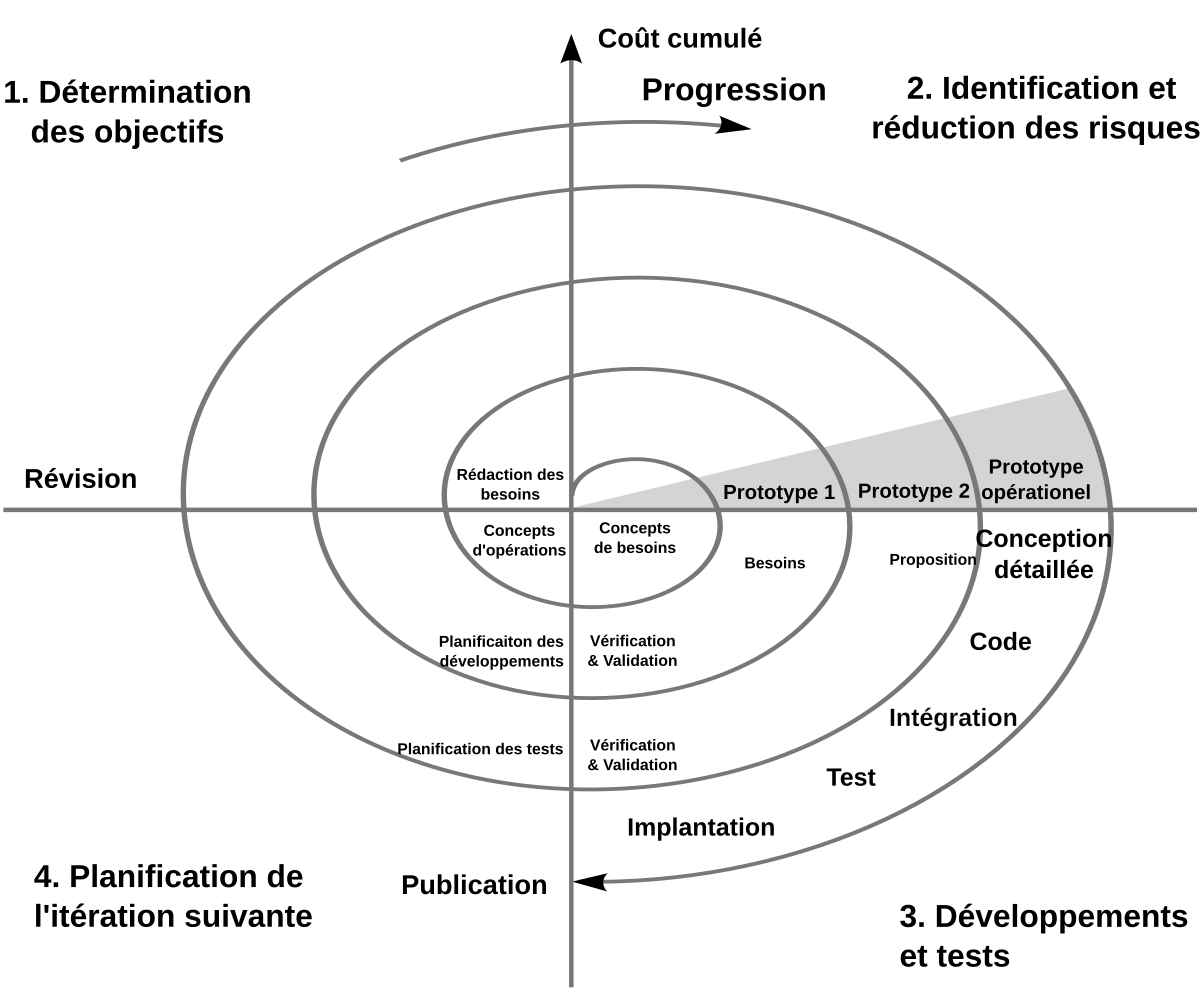
## MVP (V. 1.0)

## Version 2.0

## Version 3.0

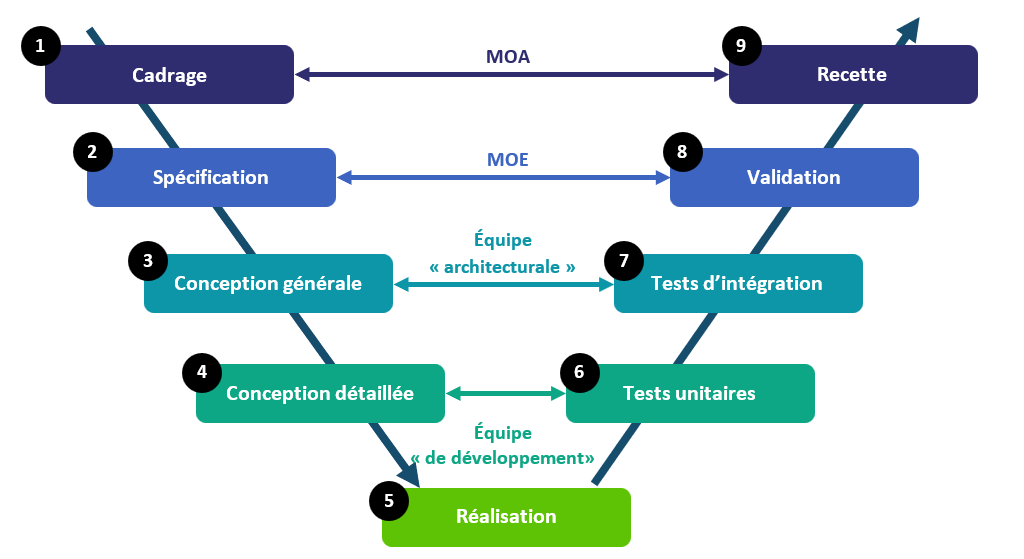
# **Méthodologie de travail**

## Cycle en Spirale



Définition :

## 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 → dd
* Spécifications → dd
* Conception générale → dd
* Conception détaillée → ded
* Réalisation → ded
* Tests unitaires → dd
* Tests d’intégration → ddd
* Validation → ddd
* Recette → dde

(parler du diagramme de Gantt, et des différentes phases d'un projet en cycle en V)

## Méthodologie Agile/Scrum

(Définir les sprints, ...)

## Méthodologie Agile/Kanban

(Pour la gestion et planification des tâches => pour cela, utilisez l'outil TRELLO)

## Méthodologie Agile/Scrumban

## Méthodologie Agile/TDD

(Pour la mise en place de tests unitaires avant le développement d’une méthode)

## Méthodologie Agile/FDD

(Pour la réalisation d’une fonctionnalité de bout en bout càd de l’interface graphique jusqu’à la base de données)

# **Spécifications fonctionnelles**

(expliquer les fonctionnalités une à une et par profil, en montrant pour trois ou quatre la **description textuelle**)

## Fonctionnalité "S'authentifier"

## Fonctionnalité "S'inscrire"

## Fonctionnalité "Ajouter un manga"

## Fonctionnalité "Ajouter article au panier"

# **Spécifications techniques**

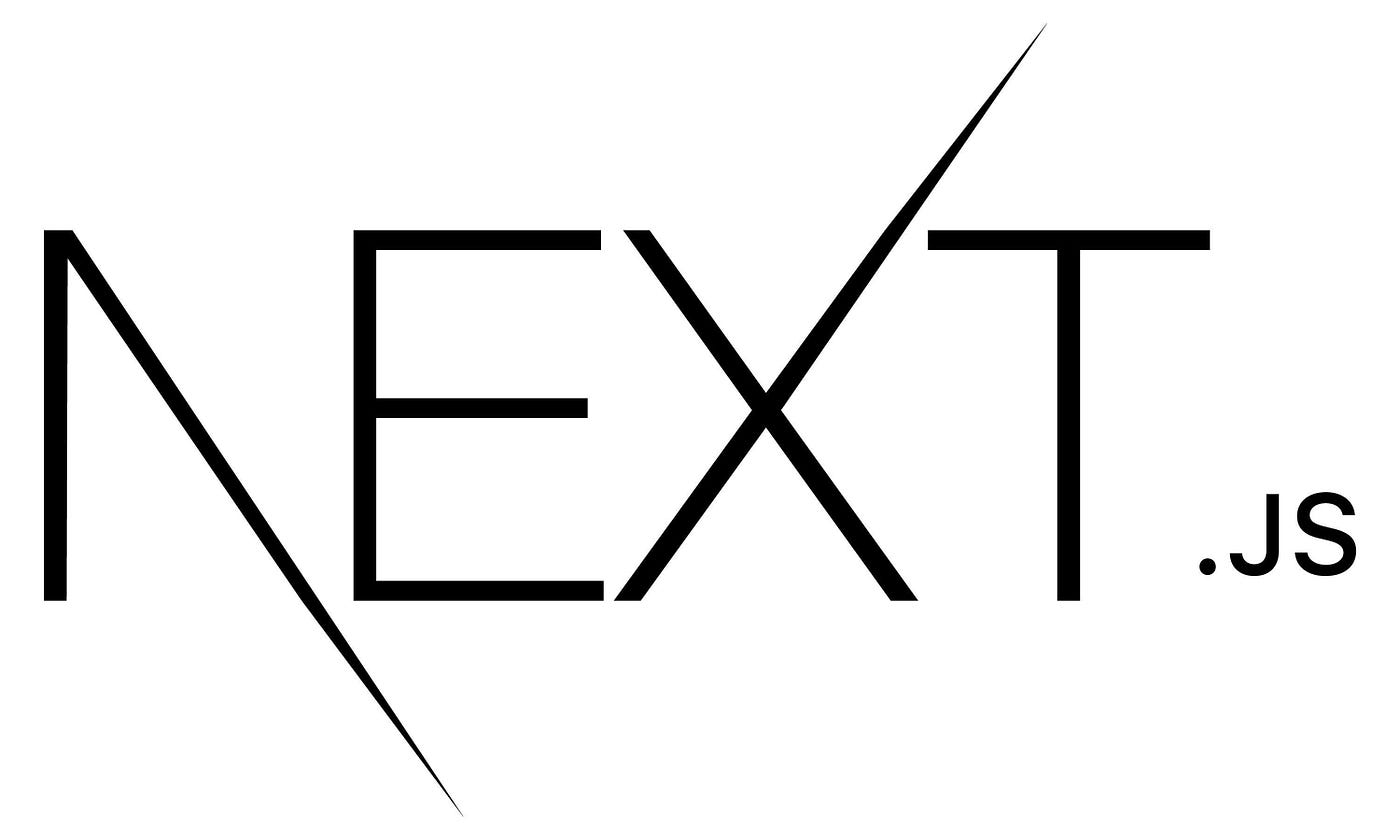
## L'architecture logicielle multicouches MVC

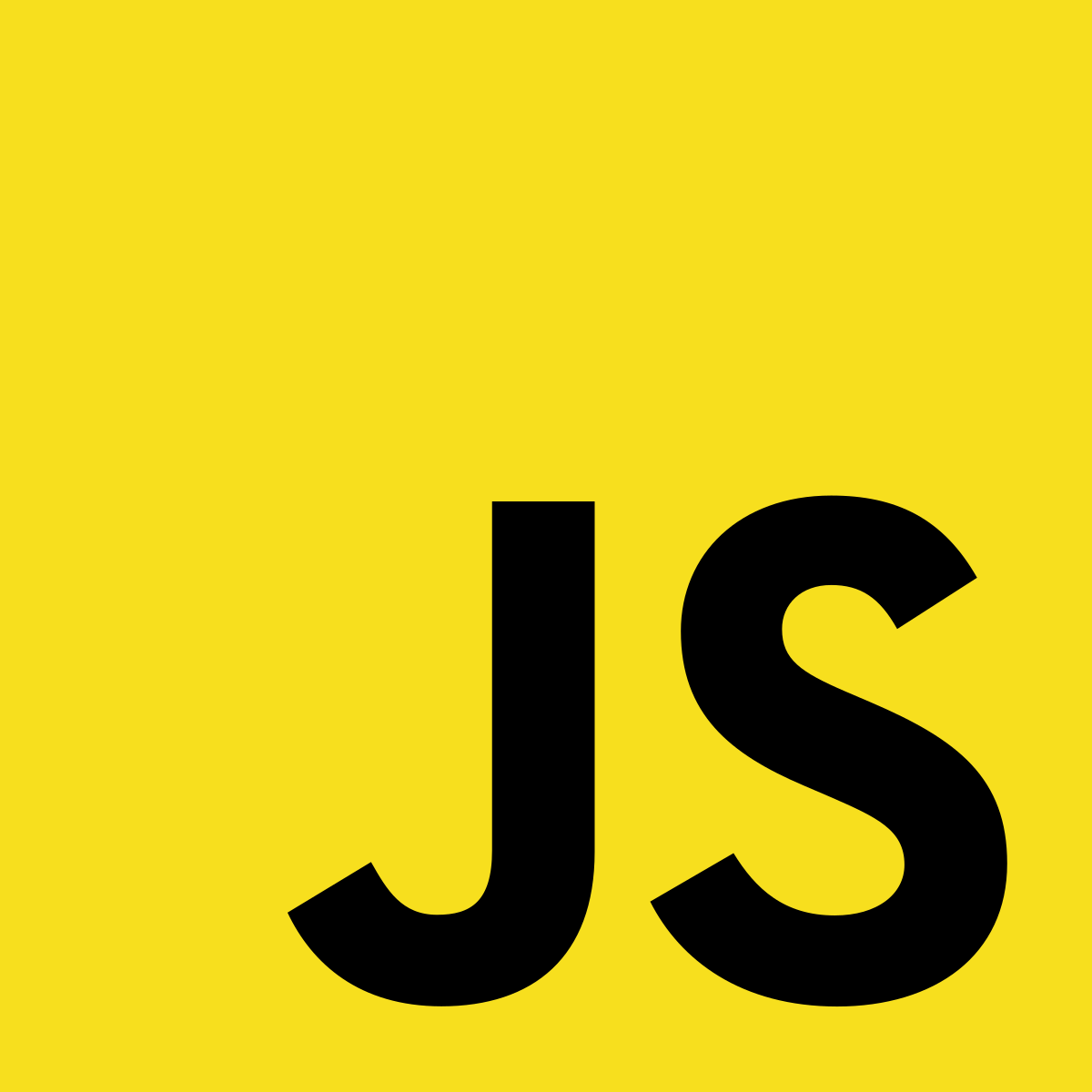
## L'architecture n-Tiers (3-Tiers)

## Technologies choisies

### 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 choisit 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 choisit ce langage pour ajouter de l’interactivité et des fonctionnalités de l’expérience utilisateur

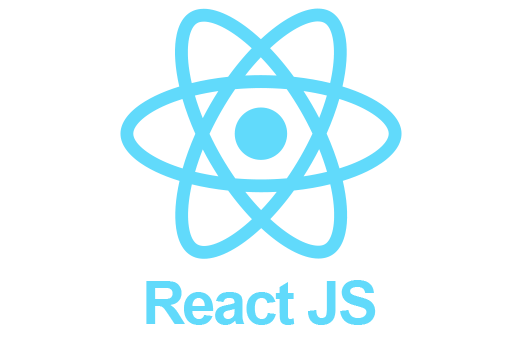


*HTML :* c’est un langage de balisage conçu pour représenter les pages web. Ce langage permet d’écrire de l’hypertexte, de créer des formulaires, et des programmes informatiques. Il est souvent utilisé conjointement avec JavaScript et CSS. J’ai choisit ce langage car c’est le langage le plus connu pour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



*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 choisit ce langage car il est toujours utilisé avec HTML.

* **Outils :**

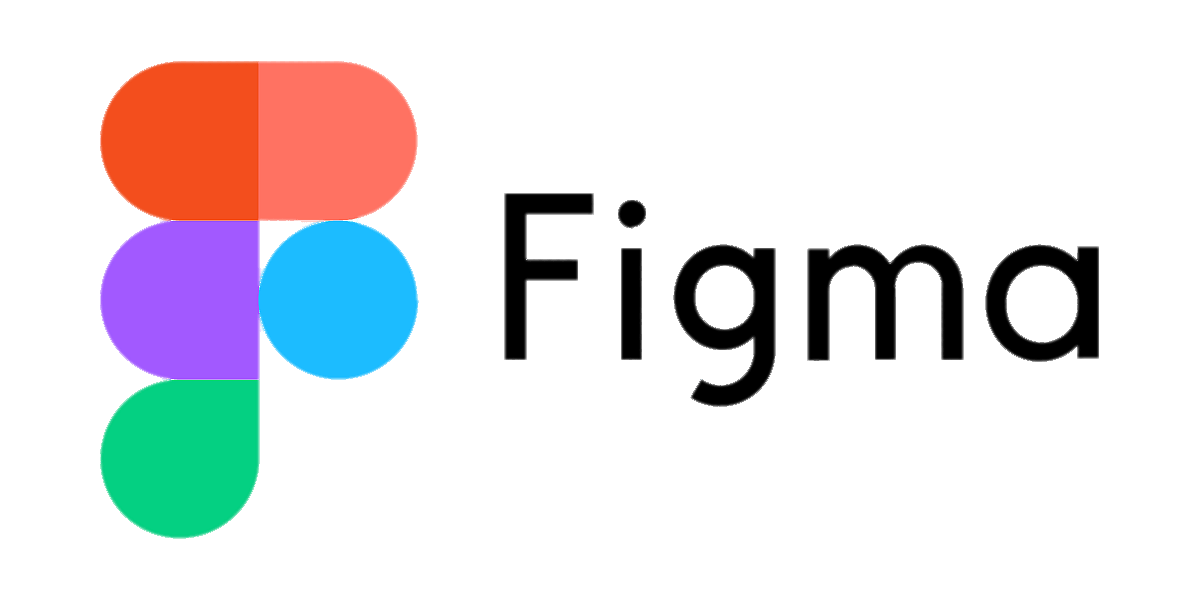
****

*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 choisit 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 choisit cette librairie car contrairement a Bootstrap, il n’y a pas de classes prédéfinis comme pour les boutons ou les tables. C’est au développeur de choisir ces propres classes et les modifies a sa manière comme sur les feuilles de styles de CSS.

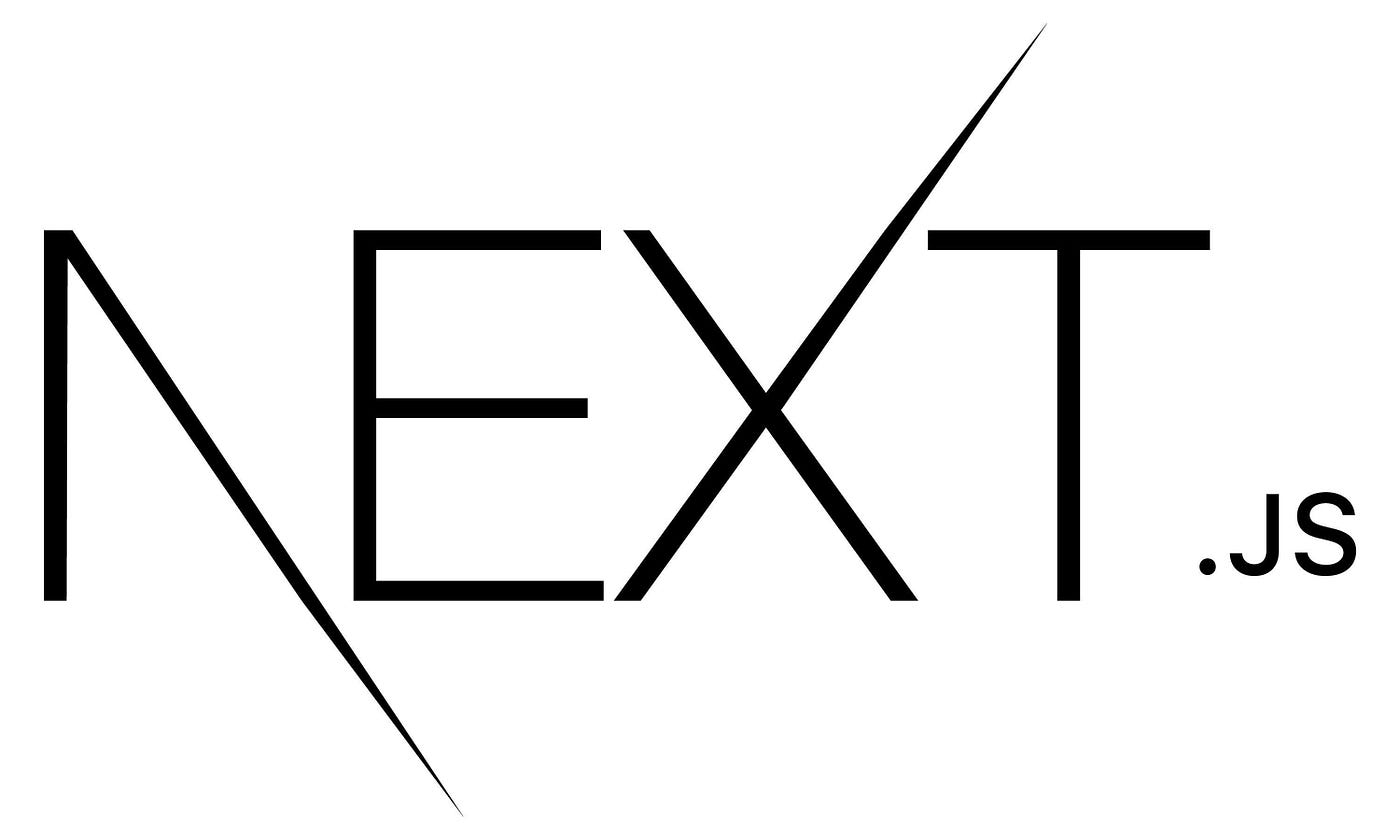


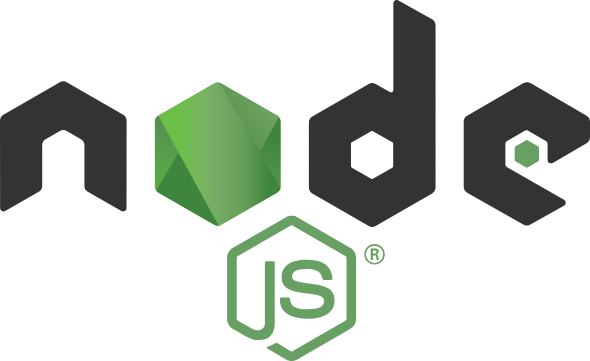
*Canva :* c’est un outil de conception graphique en ligne. J’ai choisit cette outil pour créer un logo et pour redimensionner mes images afin de les mettre a la même taille. C’est un outil très facile a utiliser fait pour tout type de personne.

*Figma :* c’est un éditeur de graphiques vectoriels et un outil de prototypage. Ses fonctionnalités sont axés 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 choisit cet outil pour créer mes différentes maquettes et l’utilisation est plutôt simple.

### 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 choisit 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 choisit 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.

 *Spring Boot :* c’est un framework Java open source utilisé pour programmer des applications web et de microservices avec Spring Framework grâce a la configuration automatique. J’ai choisit ce framework car il est orienté objet et il est très facile d’utilisation

* **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 ou avec le CRUD (Create Read Update Delete).

TiDB Cloud : c’est une base de données de service qui est open source qui permet de créer ou exporter une base de données MySQL. Comme MongoDB, TiDB utilise un service de Cloud. J’utilise TiDB pour le déploiement de l’application en exportant la base de donnée qui est sur Workbench.

### Technologies côté Persistance des données

* **Langages et Framework :**
* **Outils :**

### Technologies transverses

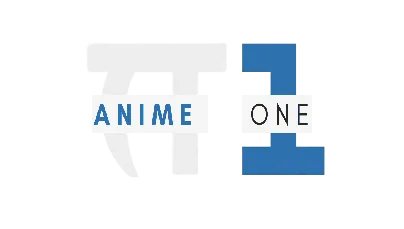
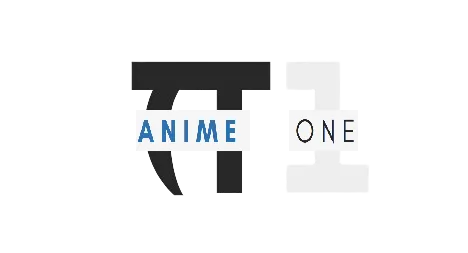
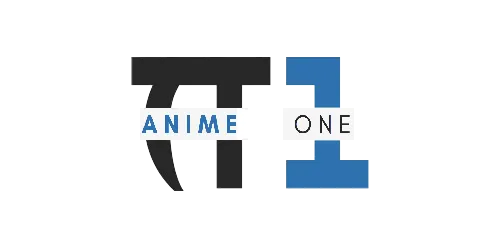
* **Langages et Framework :**
* **Outils :**

# **Conception de l'interface graphique**

## Expérience UX/UI

## Charte graphique

Mes Différents Logos :





Bannière :

Les couleurs changent dynamiquement en fonction de la page (ex : accueil, films, anime sélectionner)→

Cyan foncé (#164E63) :

Teal foncé (#134E4A) :

Rouge (#DC2626) :



Jaune foncé (#FBBF24) :



Styles de texte :

J’ai choisit comme police d’écriture Roboto, c’est une police de caractère qui appartient à Google et est de type sans-serif. J’ai choisit 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.

## 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.

## Zoning

(versions Desktop, mobile et si possible tablette)

## Wireframes

(versions Desktop, mobile et si possible Tablette)

## Maquettes

(versions Desktop, mobile et si possible tablette)

## Arborescence du site

Il faut mettre une arborescence indiquant l’enchainement des page web

(versions Desktop, mobile)

# **Conception de l'application**

## 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.

## Diagramme de cas d'utilisation

## Diagramme de séquences

## Diagramme d’activité

## Diagramme de packages

## Diagramme de classes

## Diagramme des entités

## Définition des Routes

## Routes des pages

|  |  |
| --- | --- |
| **Page** | **URL** |
| Page de garde | https://anime-one-project.vercel.app/ |
| authentification | https://anime-one-project.vercel.app/authentification |
| accueil | https://anime-one-project.vercel.app/accueil |
| manga | https://anime-one-project.vercel.app/manga |
| [id] | https://anime-one-project.vercel.app/manga/[id] |
| [animation] | https://anime-one-project.vercel.app/manga/[id]/[animation] |
| séries | https://anime-one-project.vercel.app/series |
| films | https://anime-one-project.vercel.app/films |
| scans | https://anime-one-project.vercel.app/scans |
| catégories | https://anime-one-project.vercel.app/categories |
| boutiques | https://anime-one-project.vercel.app/stores |
| feeds | https://anime-one-project.vercel.app/feeds |
| profile | https://anime-one-project.vercel.app/profile |
| admin | https://anime-one-project.vercel.app/admin |

## 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 |

# **Conception de la Base de données**

## Conception de la Base de données relationnelle MySQL

## La méthode MERISE

Définition : c’est une méthode d’analyse, de conception et de gestion de projet informatique. C’est une méthode française ou 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)

## Le Modèle Conceptuel de Données (MCD)

## Le Modèle Logique de Données (MLD)

## Le Modèle Physique de Données (MPD)

## Le Dictionnaire de Données (DD)

## Conception de la Base de données non relationnelle NoSQL

## Paramètre de Connexion

## Création de la Base de Données

## Le Dictionnaire de Données (DD)

# **Développement, Intégration et Réalisations**

## Structure du projet

(copie d'écran de l'arborescence de votre projet sur VsCode et/ou Eclipse/Inteliij)

Pour toutes les fonctionnalités ci-après, aites des copies d'écran de vos codes sources partie front (HTML/CSS/JS), back-end (Java/PHP/NextJS /...) et persistance (Java/PHP/NextJS/...)

## Fonctionnalité

## Fonctionnalité « se connecter »

## Fonctionnalité « s'inscrire »

## Fonctionnalité « ajouter un manga »

## Fonctionnalité « supprimer un manga »

## Création de la base de données SQL

## Création de la base de données NoSQL

## Requêtes SQL et NoSQL

## Requête SQL de lecture

## Requêtes SQL d’écriture

## Requêtes SQL de jointures

## Jointure interne (JOIN ou INNER JOIN)

## Jointure à gauche (LEFT JOIN)

## Jointure à droite (RIGHT JOIN)

# **Sécurité de l'application et des données**

## Les différentes attaques de site

Pour ce qui suit, il faut montre une copie d'écran du code front-end et back-end et persistance qui permet de contrer toutes ses attaques

### Les attaques XSS

### Les attaques CSRF

### Les attaques CROS

### Les attaques Force Brute

### Les attaques de type injection SQL

### Authentification/Autorisation avec JWT

### Authentification Duo/Trio/Quad

Duo : password + case à cocher (code Captcha)

Trio : Duo + question supplémentaire

Quad : Trio + confirmation par email + saisie du code reçu par email

## HTTPS

## La loi RGPD

### Confidentialité des données

(Cryptage des données (du mot de passe) (du numéro de carte CB))

Suivi des recommandations de sécurité de la CNIL : Parler surtout du mot de passe (12 caractères minimum depuis Janvier 2023 et doit être composé au minimum de : un caractère minuscule, un caractère majuscule, un chiffre, un caractère spécial)

## Mentions Légales

### Mentions Légales

### Conditions Générales d'utilisation du site web

### Conditions Générales de vente (sites de commerce ou de service)

### Politique de confidentialité

### Cookies

### Droit de rétractation

# **Sécurité de la base de données**

## Choix des noms de user et des password

Création d'un user avec un nom d'utilisateur et mot de passe complexes (C'est ce user qu'il faut utiliser pour que votre application accède à votre base de données)

## Limiter les droits au user

On créé un user spécifique pour la base de données concernée

## Limiter l'accès à la base de données

uniquement à la machine qui contient votre application serveur.

# **Sécurité et qualité du code**

## Qualité du code

### Convention de nommage

(des méthodes, des classes, des variables, des constantes, des interfaces, ...)

### Conventions de codage

### Journalisation (LOG)

### Commentaires sur les lignes de code

Documentation du code

JavaDoc ou PhpDoc ou autre

## Sécurité du code

### Tests unitaires et couverture de code

(mettre des copies d'écran de vos codes tests unitaires et couverture de code)

### Gestion des erreurs et des exceptions

### Fermeture correcte des flux de données

(Connexions, fichiers, flux, ...)

# **DevOps**

(de quoi s’agit-il ?, intérêt, démarche DevOps, intégration continue avec JENKINS, contrôle de qualité avec SonarQube, conteneurisation avec Docker)

# **Déploiement et hébergement**

* + Montrer le déploiement du site (en local et site internet)

# **Référencement SEO**

* + C’est quoi ? pourquoi ? comment le faire ?
  + <https://blog.hubspot.fr/marketing/guide-seo-referencement>
  + <https://www.seo.com/fr/basics/>
  + Si votre application est un intranet (destinée uniquement aux salariés de l’entreprise) alors le référencement n’est pas nécessaire

# **Veilles technologiques**

## Veille concernant le développement

-- Montrer les copies d'écran des erreurs -- Montrer les démarches faites pour résoudre le bug (recherche internet, solutions, copie écran d’une solution, copie écran de la console après la solution appliqué)

## Évolutions des technologies

(donner les versions des outils utiliser ex : react : 1.9)

## Veille sur la sécurité des applications

## Les différentes failles de sécurité les plus dangereuses

<https://owasp.org/>

(TOP 10 des failles de sécurité)

Le numéro 1 des failles de sécurités les plus dangereuses est : l’injection SQL

## Les recommandations de l'ANSSI (<https://cyber.gouv.fr/>)

Agence nationale de la sécurité des systèmes d'information

(suivre les nouvelles recommandations)

<https://cyber.gouv.fr/>

L'ANSSI est l'autorité nationale en matière de sécurité et de défense des systèmes d'information.

### La CNIL et ses recommandations

CNIL (suivre les nouvelles recommandations) (<https://www.cnil.fr/fr>)

## Veille sur la sécurité des bases de données

# **Perspectives**

Dire qu'on va continuer à entretenir et améliorer le projet

Futurs améliorations/fonctionnalités (réaliser la Version 2)

...

# **Conclusion**

# **Annexe**

Annexe A : ….

Annexe B : …

Annexe C : …