**Auteur :** [Nom et prénom de l’étudiant]  
**Version :** [ex. 1.0.0]  
**Date :** [JJ/MM/AAAA]

**Version** 0.0.1

Documentation et rapport du projet MDD



[**1. Présentation générale du projet 2**](#_dhl3a0hycnr7)

[1.1 Objectifs du projet 2](#_o35fqbk3ygch)

[1.2 Périmètre fonctionnel 2](#_wrjd6nideiva)

[**2. Architecture et conception technique 3**](#_2inqhgqf55pm)

[2.1 Schéma global de l’architecture 3](#_wefoq3j52eak)

[2.2 Choix techniques 3](#_6srmajod90ml)

[2.3 API et schémas de données 4](#_6zw7iaze3qzo)

[**3. Tests, performance et qualité 4**](#_inhasoir9keg)

[3.1 Stratégie de test 4](#_uxy6b4rt5ajd)

[3.2 Rapport de performance et optimisation 4](#_ghh4b7reek8n)

[3.3 Revue technique 5](#_2tf6ciebf0hd)

[**4. Documentation utilisateur et supervision 5**](#_7hmt38lmv2nl)

[4.1 FAQ utilisateur 5](#_4icfmouyvvxl)

[4.2 Supervision et tâches déléguées à l’IA 5](#_xygfxhwyvmiw)

[**5. Annexes 6**](#_btgt7me24jiq)

## **1. Présentation générale du projet**

### **1.1 Objectifs du projet**

*Brièvement, présentez le* ***but*** *du projet, les* ***besoins*** *métiers et les* ***principales fonctionnalités*** *développées.  
Expliquez le* ***contexte*** *de l’entreprise et la* ***valeur ajoutée*** *attendue du produit.*

### **1.2 Périmètre fonctionnel**

*Présentez les* ***fonctionnalités livrées*** *(liste synthétique), en précisant leur état (terminée / en cours / à venir).*

|  |  |  |
| --- | --- | --- |
| **Fonctionnalités** | **Description** | **Statut** |
| *Création d’un compte utilisateur* | *Formulaire et validation d’inscription* | nok |
| *Publication d’un article* | *Gestion CRUD via API* | nok |
| *Commentaires* | *Association article/commentaires* | nok |
| *Authentification* | *Sécurisation JWT* | nok |

## **2. Architecture et conception technique**

### **2.1 Schéma global de l’architecture**

*Intégrez un diagramme d’architecture (UML, C4 ou équivalent) illustrant les liens entre :*

* *le front-end,*
* *l’API,*
* *le back-end et la base de données,*
* *les outils externes ou services tiers.*

Ajoutez une légende explicative et précisez les **choix d’organisation technique** (modules, dossiers, conventions internes).

### **2.2 Choix techniques**

Présentez ici **chaque choix structurant** du projet.  
 Pour chaque élément, complétez le tableau suivant :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Éléments choisis** | **Type** | **Lien documentation** | **Objectif du choix** | **Justification** |
| *Ex : Angular 19* | *Framework front-end* |  | *Structuration de l’application et gestion de la réactivité* | *Respect des standards du parcours et cohérence avec les maquettes Figma* |
| Spring boot 2.7 | Framework backend |  | Ne pas changer l’existant. | Le but de se projet étant de me formé j’ai penser que le framework plus ancien que sur les autre projet avais pour but de m’apprendre d’autre méthode, analyser ma capacité a m’adapter |
| H2 | BDD |  | Pouvoir effectuer les test facilement. | La bdd H2 peut étre reset et remplis avec les donné de test facilement, cela me semble judicieux pour les test. |

### **2.3 API et schémas de données**

*Présentez ici la* ***conception et la structuration de votre API*** *:*

* *endpoints créés,*
* *méthodes HTTP,*
* *exemples de requêtes/réponses JSON,*
* *schémas de données (entités, relations, contraintes).*

|  |  |  |  |
| --- | --- | --- | --- |
| **Endpoint** | **Méthode** | **Description** | **Corps / Réponse** |
| */api/articles* | *GET* | *Récupère la liste des articles* | *JSON – liste d’articles* |
| */api/users/{id}* | *GET* | *Détail d’un utilisateur* | *JSON – profil utilisateur* |
| */api/login* | *POST* | *Authentifie un utilisateur* | *Token JWT* |

*Ajoutez une représentation visuelle des relations (UML / diagramme de classes ou entités).*

## **3. Tests, performance et qualité**

### **3.1 Stratégie de test**

*Décrivez les tests mis en place :*

* ***unitaires****,* ***d’intégration****,* ***end-to-end****,*
* *frameworks utilisés,*
* *taux de couverture.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Type de test** | **Outil / framework** | **Portée** | **Résultats** |
| *Test unitaire* | *JUnit* | *Services du back-end* |  |
| *Test d’intégration* |  |  |  |
| *Test e2e* |  |  |  |

### **3.2 Rapport de performance et optimisation**

*Décrivez les actions menées pour* ***améliorer la performance*** *du code et du rendu :*

* *résultats d’audit (Lighthouse, SonarQube, etc.),*
* *points d’amélioration identifiés,*
* *actions correctives appliquées.*

*Exemple :  
 “Après audit Lighthouse, la performance du front est passée de 65 à 91/100 grâce à l’optimisation des images et du lazy-loading des modules Angular.”*

### **3.3 Revue technique**

*Présentez une* ***synthèse critique du code*** *:*

* *points forts (structure, modularité, lisibilité),*
* *points à améliorer (complexité, dette technique, sécurité),*
* *actions correctives appliquées.*

*Exemple :*

* ***Point fort :*** *modularité des services Spring.*
* ***À améliorer :*** *duplication de logique dans les contrôleurs.*
* ***Action corrective :*** *refactorisation via héritage de service parent.*

## **4. Documentation utilisateur et supervision**

### **4.1 FAQ utilisateur**

*Rédigez une courte section d’aide destinée aux utilisateurs internes ou finaux.  
 Structurez-la en format* ***Question / Réponse****.*

***Q : Comment créer un compte ?*** *R : Cliquez sur “S’inscrire”, remplissez le formulaire et validez. Vous recevrez un email de confirmation.*

***Q : Que faire si l’application ne charge pas ?*** *R : Rafraîchissez la page. Si le problème persiste, contactez le support technique.*

### **4.2 Supervision et tâches déléguées à l’IA**

Décrivez les tâches confiées à l’IA, et comment vous avez **vérifié, validé ou corrigé** son travail.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tâche déléguée** | **Outil / collaborateur** | **Objectif** | **Vérification effectuée** |
| *Ex : Génération de tests unitaires* | *GitHub Copilot* | *Gain de temps sur les tests basiques* | *Revue et correction des assertions* |
|  |  |  |  |

## 

## **5. Annexes**

Intégrez ici toutes les pièces justificatives :

1. **Captures d’écran de l’UI** et vues principales.
2. **Analyse des besoins front-end** (liens avec les spécifications ou maquettes).
3. **Définition des données** (schémas, formats, règles de validation et sécurisation).
4. **Rapports de couverture et de tests** (exports ou impressions d’écran).
5. **Rapport de revue technique** (version complète, datée et signée si applicable).