



Cahier des charges

Ruben TROTTEIN

[Tapez le résumé du document ici. Il s'agit généralement d'une courte synthèse du document.]

Projet DAN
Hedgehog School
+ DuckDrive API

jeudi 28 novembre 2024

TABLE DES MATIERES

1. Présentation du projet	2
2. Expression des besoins	3
A. Application Hedgehog School :	3
B. API DuckDrive :	3
3. Description fonctionnelle détaillée.....	3
4. Architecture technique.....	5
5. Gestion du projet.....	6
A. Conception et validation des spécifications	6
B. Etapes et échéances.....	6
6. Conclusion:	7
7. Annexes ?	7

1. PRESENTATION DU PROJET

1.1 Contexte

- Présentation rapide du programme de formation CDA.

Ce projet est présenté dans le cadre du titre professionnel de Concepteur Développeur d'Applications (CDA), RNCP de niveau 6 (Bac+3).

Les objectifs de formation couverts par ce travail sont :

- Conception, développement et documentation d'un projet informatique
- Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité
 - Maquetter une application
 - Développer une interface utilisateur de type desktop
 - Développer des composants d'accès aux données
 - Développer la partie front-end d'une interface utilisateur Web
 - Développer la partie back-end d'une interface utilisateur Web
- Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité
 - Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement
 - Concevoir une application
 - Développer des composants métier
 - Construire une application organisée en couches
 - Développer une application mobile
 - Préparer et exécuter les plans de tests d'une application
 - Préparer et exécuter le déploiement d'une application
- Concevoir et développer la persistance des données en intégrant les recommandations de sécurité
 - Concevoir une base de données
 - Mettre en place une base de données
 - Développer des composants dans le langage d'une base de données

1.2 Objectifs

- **Hedgehog School** : Une application Symfony permettant de gérer l'interaction de l'utilisateur avec l'API DuckDrive.
- **DuckDrive API** : Fournir des fonctionnalités backend pour la gestion d'articles via un serveur Node.js.

1.3 Périmètre

- Ce projet inclut le développement frontend (Hedgehog School) et backend (DuckDrive API).
- La gestion des informations et fichiers est au cœur du périmètre fonctionnel.

2. EXPRESSION DES BESOINS

2.1 Besoins fonctionnels

A. APPLICATION HEDGEHOG SCHOOL :

- Gestion des utilisateurs pour l'attribution des fichiers et l'évaluation des articles
- Création d'un formulaire pour soumettre des requêtes à DuckDrive API.
- Validation des données saisies dans les formulaire.
- Retour et affichage des réponses de l'API sous forme lisible pour l'utilisateur.
- Agrégation des fichiers multimédias et du contenu stocké ou géré par l'API

B. API DUCKDRIVE :

- Création d'un endpoint pour gérer les articles (création, récupération, suppression).
- Validation et gestion des erreurs des requêtes entrantes.
- Adressage des fichiers et stockage distant
- Format des échanges en JSON.

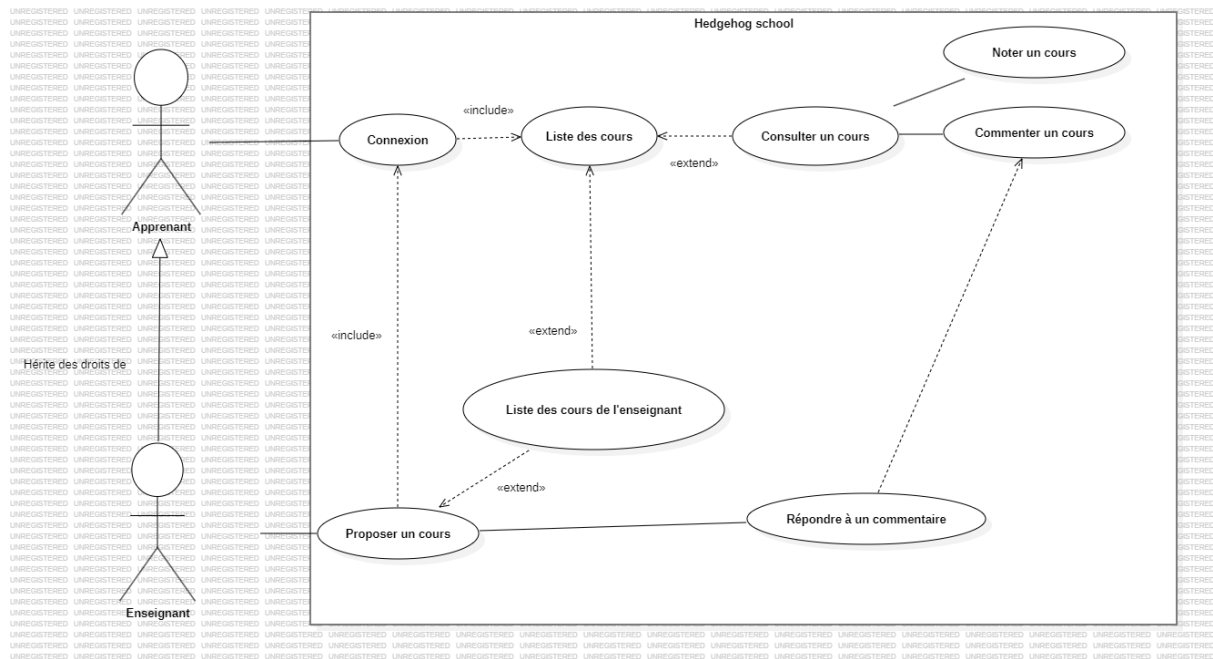
2.2 Besoins non fonctionnels

- Compatibilité avec les navigateurs modernes (pour Hedgehog School).
- Sécurité et authentification des utilisateurs (pour Hedgehog School)
- Sécurité et authentification des appels API.
- Respect des standards REST pour l'API.

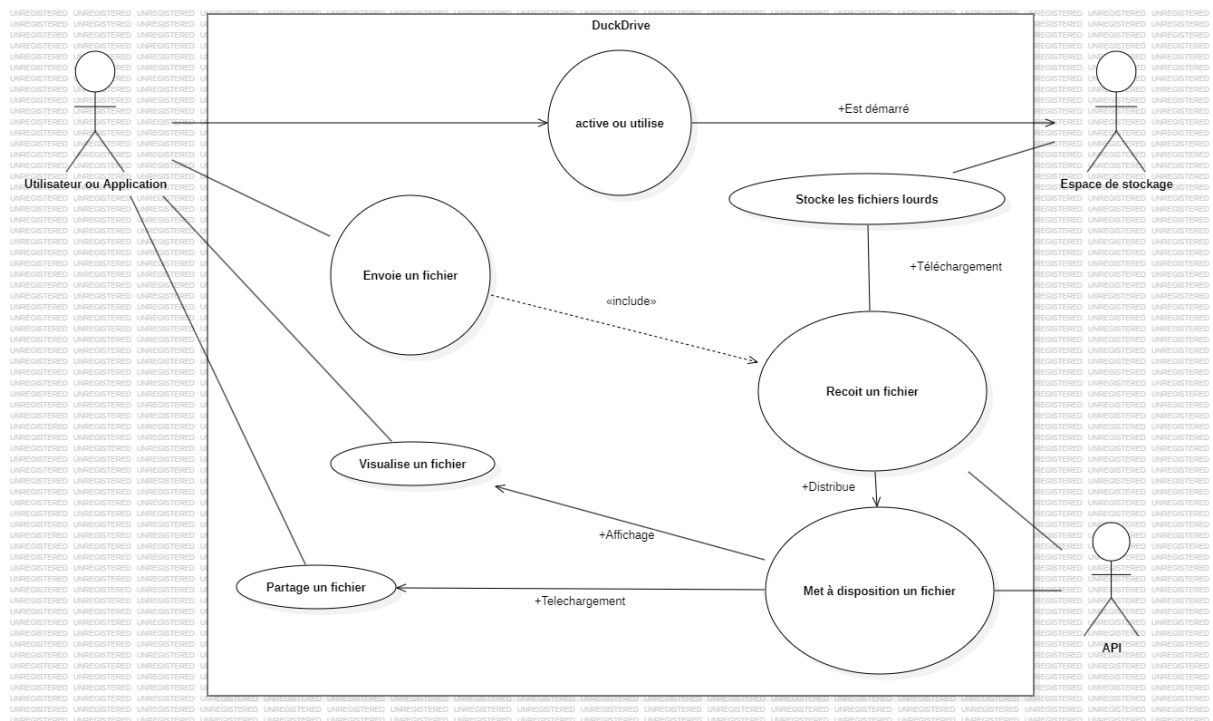
3. DESCRIPTION FONCTIONNELLE DETAILLEE

3.1 Diagramme de cas d'utilisation

Utilisation Type de Hedgehog school : proposition, consultation et review d'articles pédagogiques ou informatifs

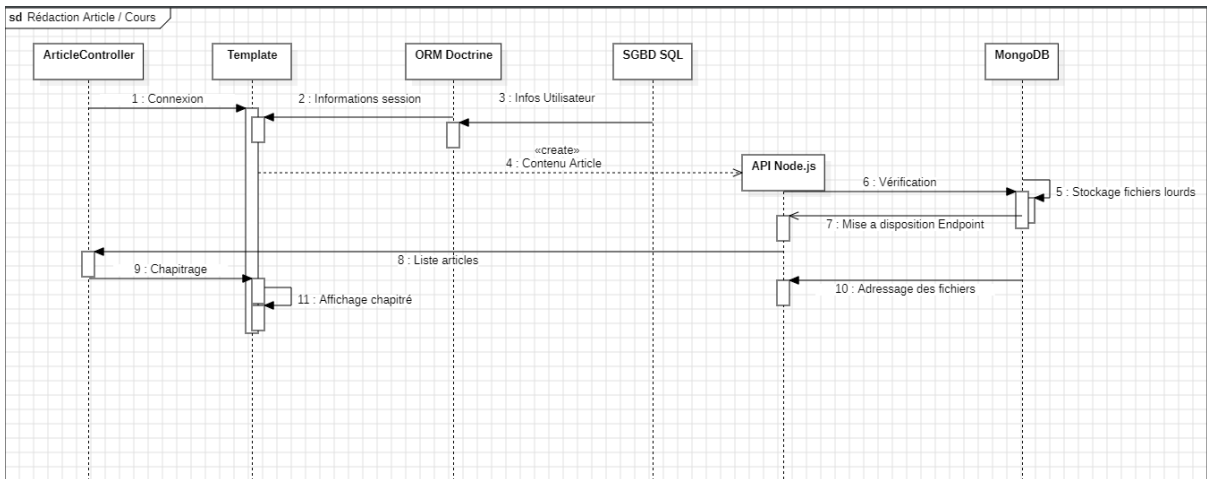


Utilisation Type de DuckDrive API :réception des fichiers, allègement de la charge de Hedgehog School et distribution des données a l'application.



3.2 Scénarios d'utilisation principaux

- Scénario type : L'utilisateur soumet un formulaire via Hedgehog School pour créer un article.



Dans ce diagramme de séquence Hedgehog School reçoit une réponse JSON de DuckDrive API et affiche un message d’erreur si elle rencontre un souci, que l’application traduit en message intelligible pour l’utilisateur.

4. ARCHITECTURE TECHNIQUE

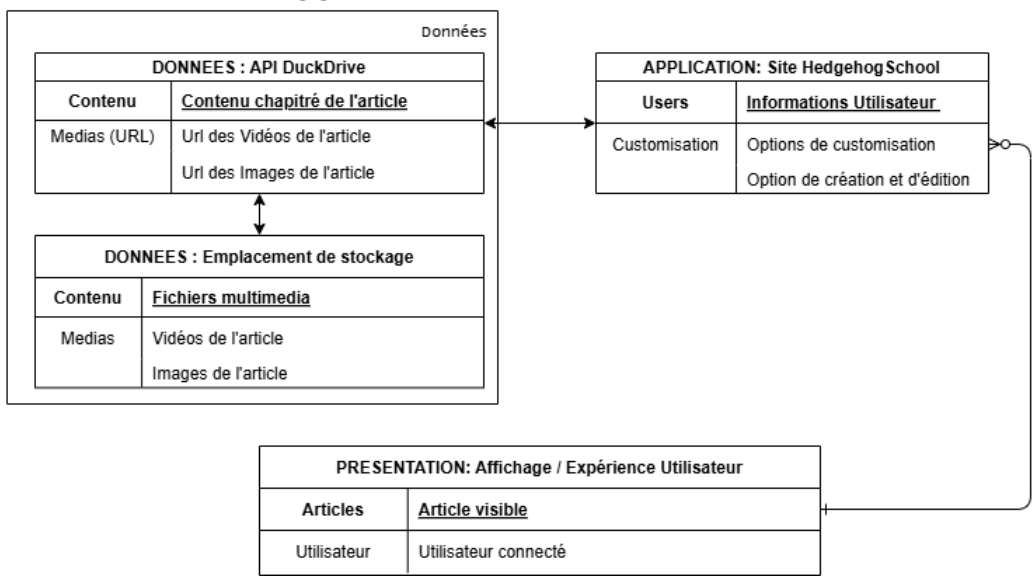
4.1 Technologies utilisées

- Symfony (PHP) pour Hedgehog School. Utilisation de Doctrine ORM avec MySQL
- Node.js avec Express.js pour DuckDrive API.
- Base de données associée : MongoDB

4.2 Diagramme d'architecture

- Diagramme illustrant la communication entre les composants du système :
 - Hedgehog School (Frontend Symfony).
 - DuckDrive API (Backend Node.js).

Architecture de l'application



4.3 Contraintes techniques

- Normes PSR-12 pour le code PHP.
 - Utilisation de Composer pour la gestion des dépendances Symfony.
 - Respect des bonnes pratiques RESTful pour l'API Node.js.
-

5. GESTION DU PROJET

5.1 Organisation et planning

- Description des étapes du projet :
 - Conception et validation des spécifications.
 - Développement de Hedgehog School.
 - Développement de DuckDrive API.
 - Tests unitaires, fonctionnels et d'intégration.

A. CONCEPTION ET VALIDATION DES SPECIFICATIONS

DuckDrive

- Icones, aperçus
- 2 pages (Dashboard avec explorateur et formulaire d'upload + page d'accès rapide aux favoris)
- Volet de visualisation rapide

Hedgehog School

- Page de cours (vidéo + contenu markdown / Wysiwyg)
- Index des cours et recherche
- Dashboard cours proposés / suivis / favoris /
- Système de likes ou notes pour indexer les contenus qualitativement
- Module de connexion et de profil pour décrire le domaine de ses cours et l'expertise de leur(s) auteur(s)

B. ETAPES ET ECHEANCES

Phase	Description	DeadLine
Conceptualisation	Spécifications techniques et Cahier des charges	<i>Fin Mai 2024</i>
Maquettage	Proposition de wireframes et Proof of Concept	<i>Fin Juin 2024</i>
Prototypage	Mise en place du code du projet en environnement dev en déploiement continu	<i>Fin Décembre 2024</i>
Déploiement	Mise en Production des solutions	<i>Janvier 2025</i>

5.2 Rôles et responsabilités

- Rôle du développeur dans l'ensemble du cycle de vie du projet.

Mon rôle a été de développer, écrire la documentation, produire les modélisations et diagrammes, choisir les technologies et l'architecture et mettre en place les différentes solutions pour permettre la conception, l'écriture du code et le déploiement des applications et de ses documents.

- Le Jury CDA comme partie prenante pour valider le produit final.
-

6. CONCLUSION:

Les objectifs pour ce projet ont été remplis. Nous avons pu démontrer nos compétences en développement Symfony et Node.js en concevant un système complet et fonctionnel, dans le cadre d'une conception classique avec une organisation documentée et agile.

Grâce à ce projet nous pouvons illustrer la synergie des diverses compétences acquises et consolidées durant la formation, à commencer par la maîtrise et la combinaison des technologies et techniques nécessaires à l'élaboration d'une solution viable et sécurisée.

7. ANNEXES ?

- Diagrammes UML (cas d'utilisation, architecture) ?
- Description détaillée des endpoints de l'API.
- Exemple de requête/ réponse JSON.