

Rapport de Projet

Titre : Développement d'un Site E-commerce et Analyse des Architectures Logicielles

Rapport de Projet.....	1
1. Introduction	2
2. Planification et Gestion du Projet	2
3. Conception UML du Site E-commerce	4
4. Étude des Architectures Logicielles (Activité 3)	9
Analyse Comparative : Tableau Comparatif des Logiciels	13
5. Résultats et Discussion.....	14
6. Conclusion et Perspectives	14
7. Annexes	15

1. Introduction

Objectif du projet :

L'objectif principal est de développer un site e-commerce qui permet à un administrateur de gérer les produits et aux visiteurs de parcourir, commander et payer des articles en ligne. Parallèlement, nous avons étudié les architectures de logiciels appartenant à différentes catégories pour en comprendre les points forts et faibles.

Portée :

- Le site doit être fonctionnel, responsive, et adapté aux utilisateurs finaux.
- L'analyse logicielle inclut des systèmes d'exploitation, des frameworks, des bases de données, et des langages de programmation couramment utilisés dans le développement logiciel.

Structure du rapport :

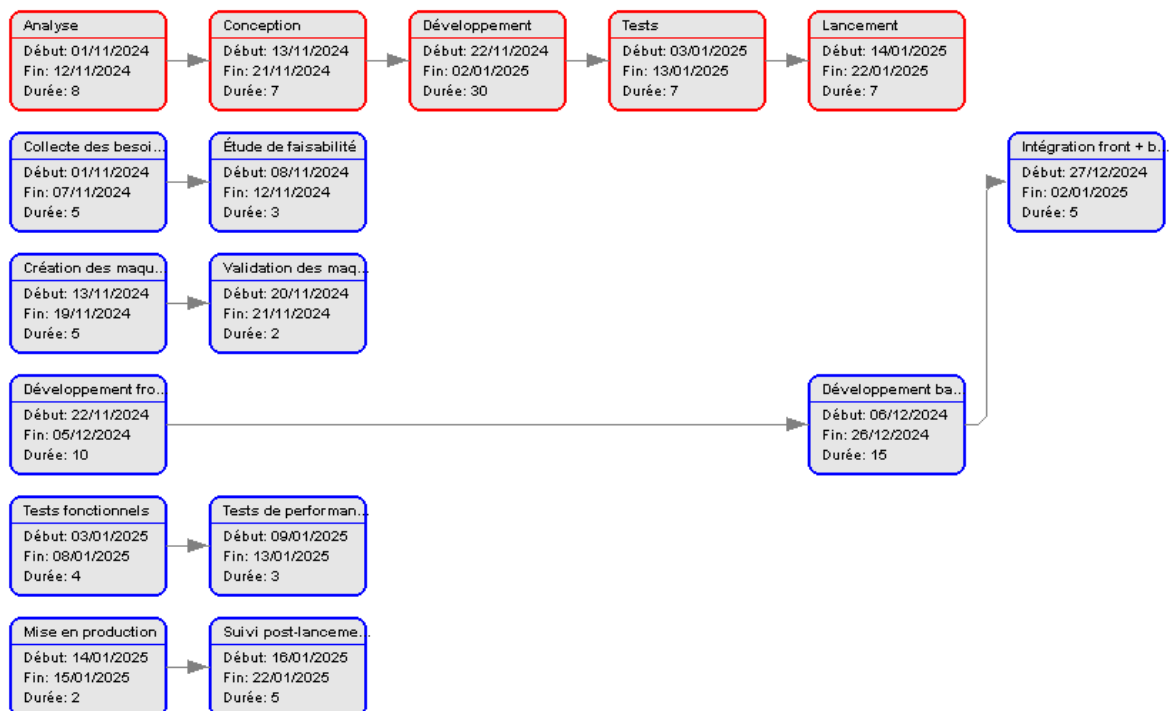
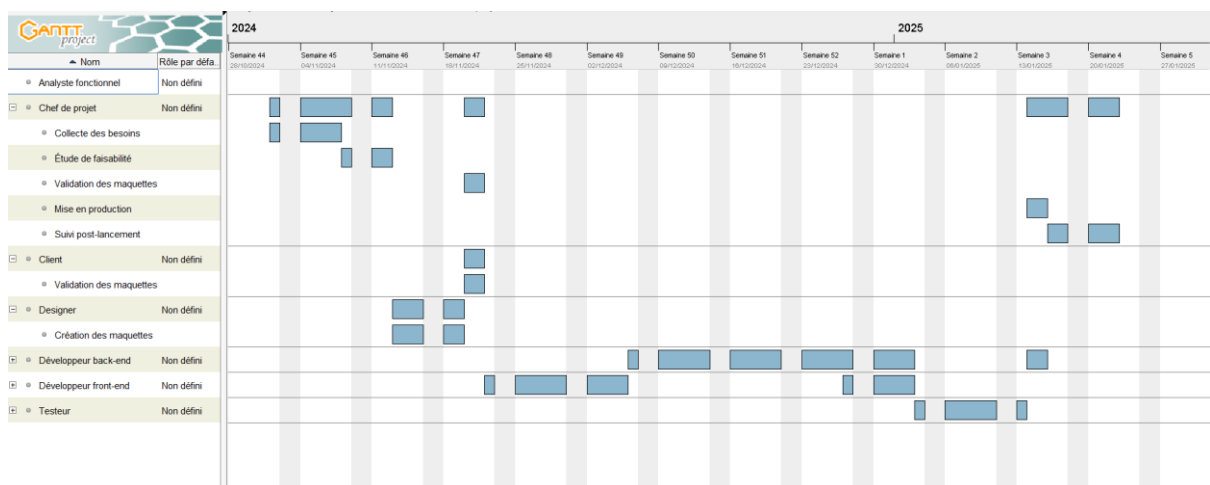
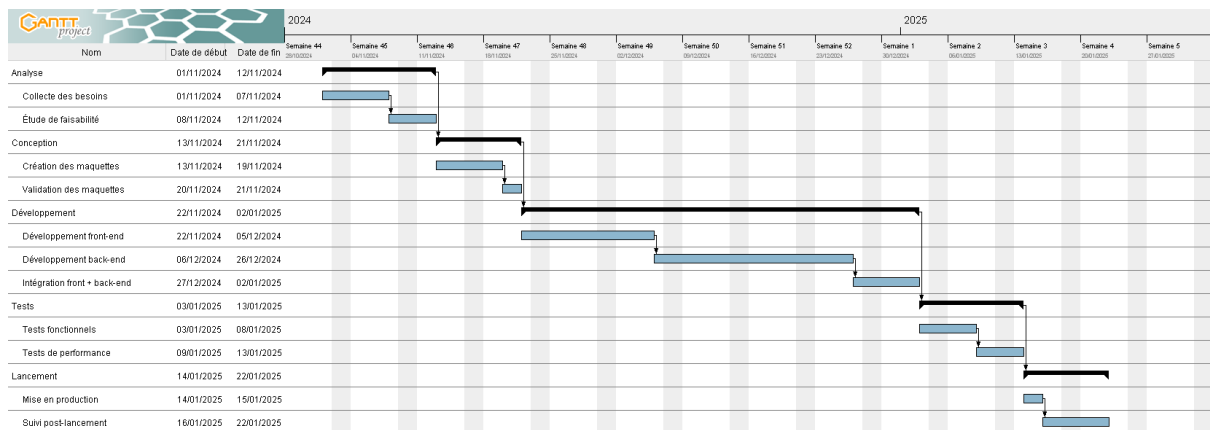
Ce rapport est divisé en trois parties principales : la planification du projet, la conception UML du site e-commerce, et l'analyse des architectures logicielles.

2. Planification et Gestion du Projet

Diagramme de GANTT :

Le projet a été planifié sur 6 semaines avec les étapes suivantes :

1. **Analyse** : Collecte des besoins et faisabilité (5 jours).
2. **Conception** : Création et validation des maquettes (7 jours).
3. **Développement** : Front-end et back-end (25 jours).
4. **Tests** : Tests fonctionnels et de performance (7 jours).
5. **Lancement** : Mise en production et suivi post-lancement (7 jours).



Répartition des tâches :

Rôle	Responsabilité
Chef de projet	Planification et coordination générale.
Analyste fonctionnel	Conception UML et documentation des cas d'utilisation.
Développeur	Traduction UML en base de données, développement back-end.
Architecte logiciel	Analyse des logiciels et création des diagrammes.
Qualité et comparaison	Critères de qualité et comparaison des logiciels.

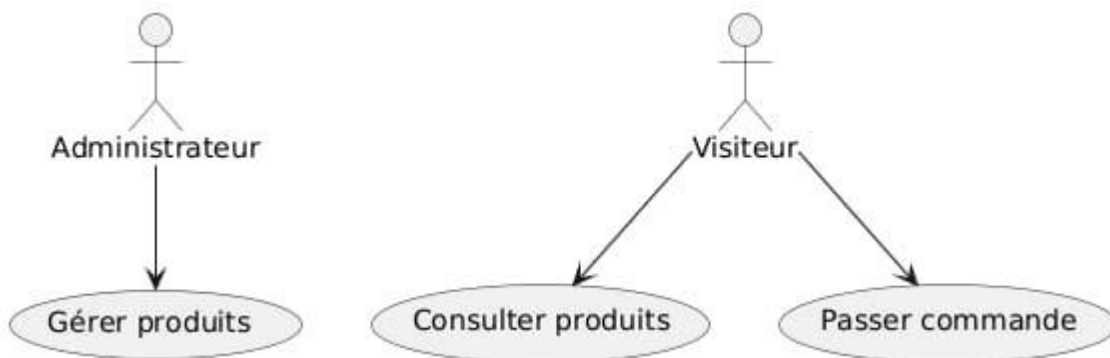
3. Conception UML du Site E-commerce

Résumé des fonctionnalités principales :

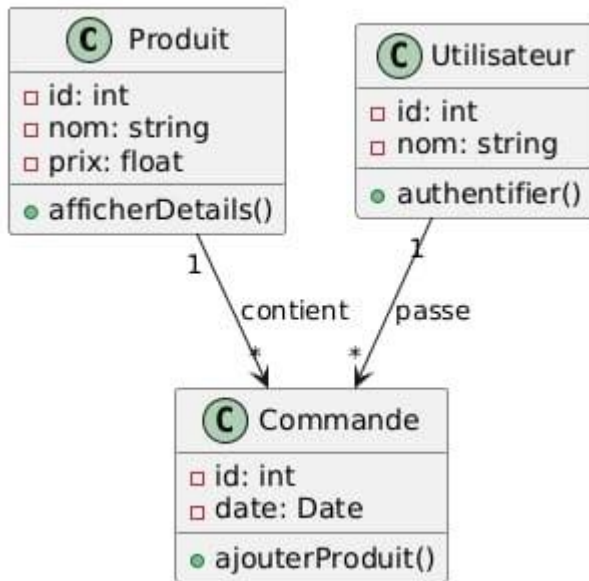
- **Administrateur** : Ajouter, modifier, supprimer des produits et gérer les commandes.
- **Visiteurs** : Consulter les produits, gérer le panier, passer commande, et effectuer le paiement.

Diagrammes UML réalisés :

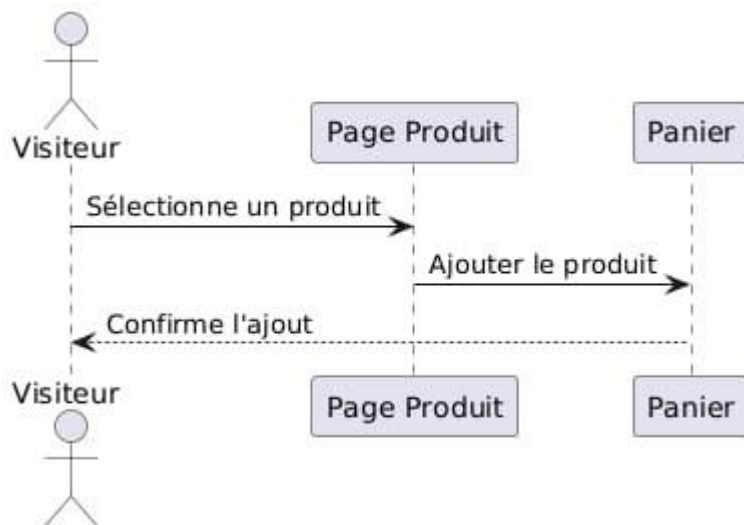
1. Diagramme de cas d'utilisation :



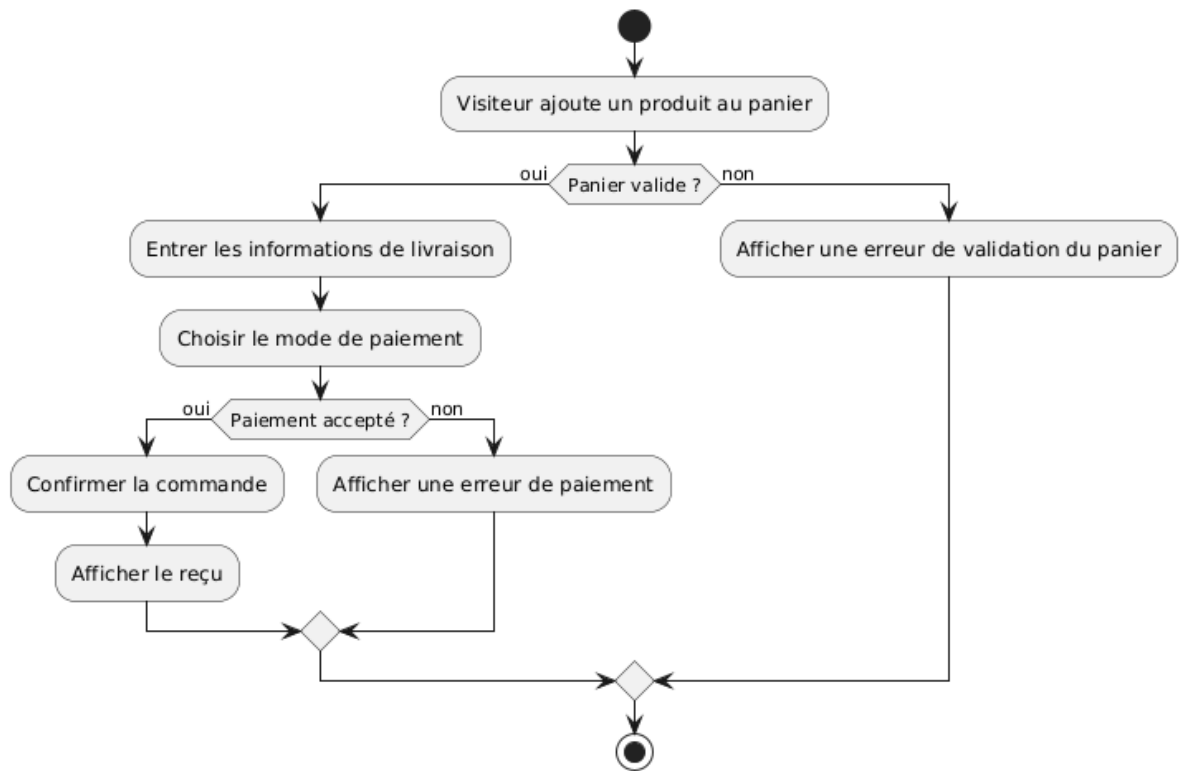
2. Diagramme de classes :



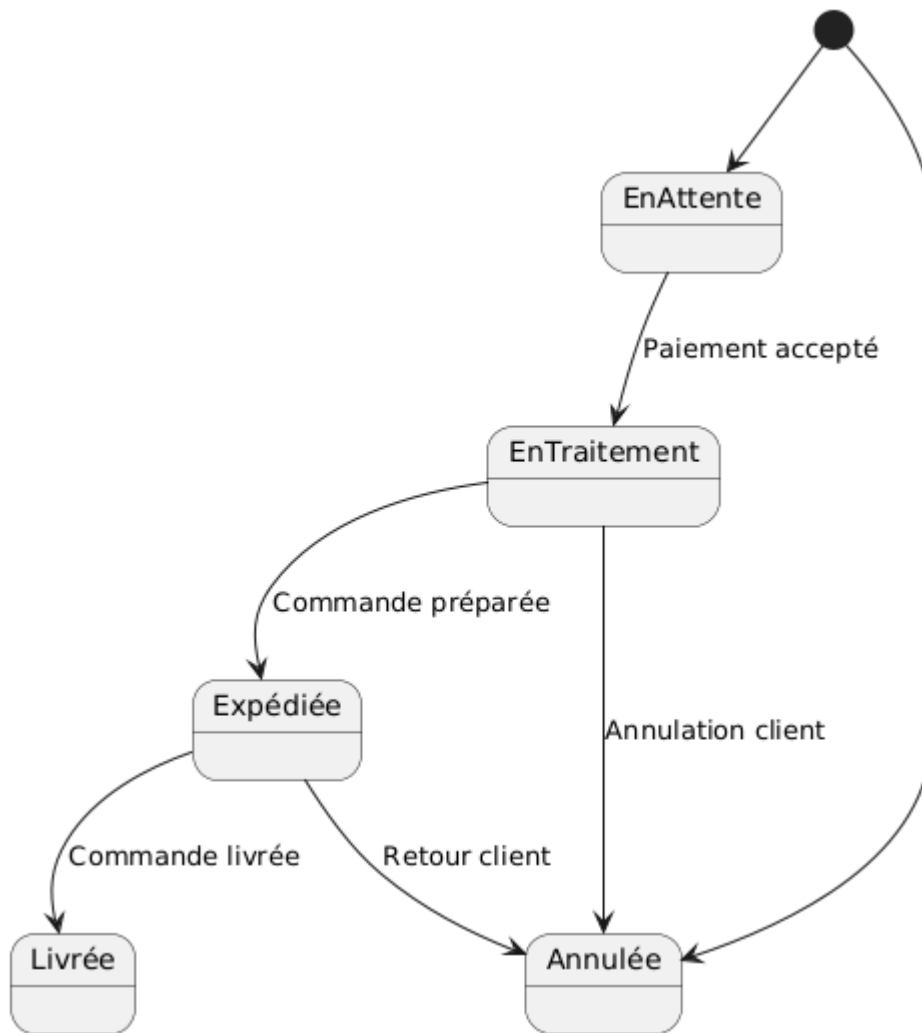
3. Diagramme de séquences :



4. Diagramme d'activité :



5. Diagramme d'état-transition :



6. Schéma relationnel :

```

-- Création des tables
CREATE TABLE Produit (
  id INTEGER PRIMARY KEY, -- SQLite gère automatiquement l'auto-incrémentation
  pour ce type
  nom TEXT NOT NULL,
  prix REAL NOT NULL
);

CREATE TABLE Utilisateur (
  id INTEGER PRIMARY KEY,
  nom TEXT NOT NULL
);

CREATE TABLE Commande (
  id INTEGER PRIMARY KEY,

```

```

    date TEXT NOT NULL, -- Les dates sont stockées en TEXT dans SQLite
    utilisateur_id INTEGER NOT NULL,
    FOREIGN KEY (utilisateur_id) REFERENCES Utilisateur(id)
);

CREATE TABLE Commande_Produit (
    commande_id INTEGER NOT NULL,
    produit_id INTEGER NOT NULL,
    quantite INTEGER NOT NULL,
    PRIMARY KEY (commande_id, produit_id),
    FOREIGN KEY (commande_id) REFERENCES Commande(id),
    FOREIGN KEY (produit_id) REFERENCES Produit(id)
);

-- Ajout de données exemple
INSERT INTO Produit (nom, prix) VALUES ('Produit A', 10.50), ('Produit B', 20.00),
('Produit C', 15.75);
INSERT INTO Utilisateur (nom) VALUES ('Alice'), ('Bob');
INSERT INTO Commande (date, utilisateur_id) VALUES ('2024-12-01', 1), ('2024-12-02',
2);
INSERT INTO Commande_Produit (commande_id, produit_id, quantite) VALUES
(1, 1, 2),
(1, 2, 1),
(2, 3, 5);

-- Visualisation des données
SELECT * FROM Produit;
SELECT * FROM Utilisateur;
SELECT * FROM Commande;
SELECT * FROM Commande_Produit;

-- Jointure pour voir toutes les relations (exemple)
SELECT
    c.id AS commande_id,
    c.date,
    u.nom AS utilisateur,
    p.nom AS produit,
    cp.quantite
FROM Commande c
JOIN Utilisateur u ON c.utilisateur_id = u.id
JOIN Commande_Produit cp ON c.id = cp.commande_id

```


JOIN Produit p ON cp.produit_id = p.id;

id	nom	prix
1	Produit A	10.5
2	Produit B	20
3	Produit C	15.75

id	nom
1	Alice
2	Bob

id	date	utilisateur_id
1	2024-12-01	1
2	2024-12-02	2

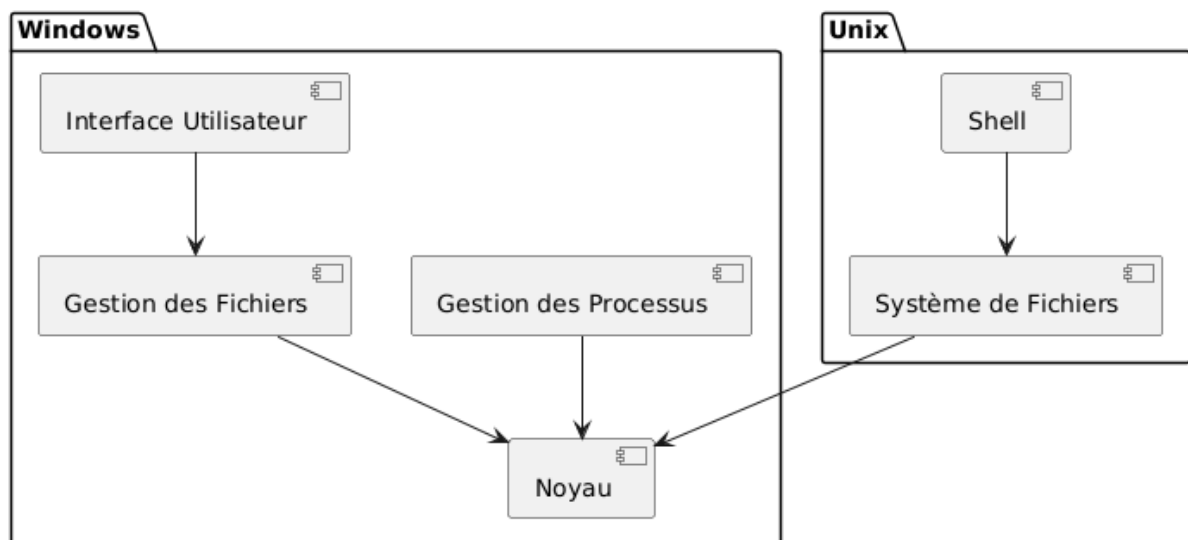
commande_id	produit_id	quantite
1	1	2
1	2	1
2	3	5

commande_id	date	utilisateur	produit	quantite
1	2024-12-01	Alice	Produit A	2
1	2024-12-01	Alice	Produit B	1
2	2024-12-02	Bob	Produit C	5

4. Étude des Architectures Logicielles (Activité 3)

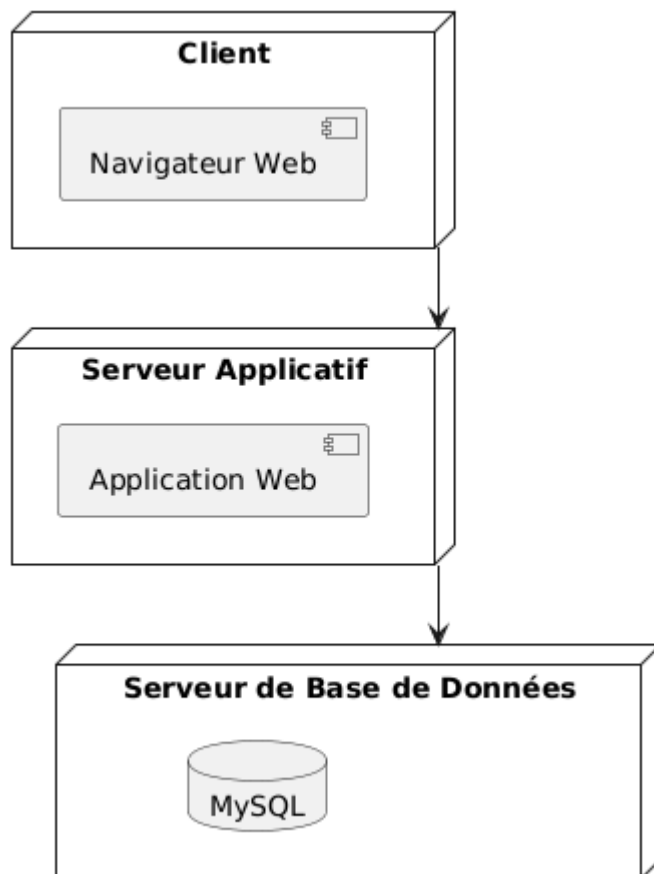
Logiciels étudiés et résultats :

1. **Systèmes d’exploitation :**
(diagrammes et explications de Windows et Unix)



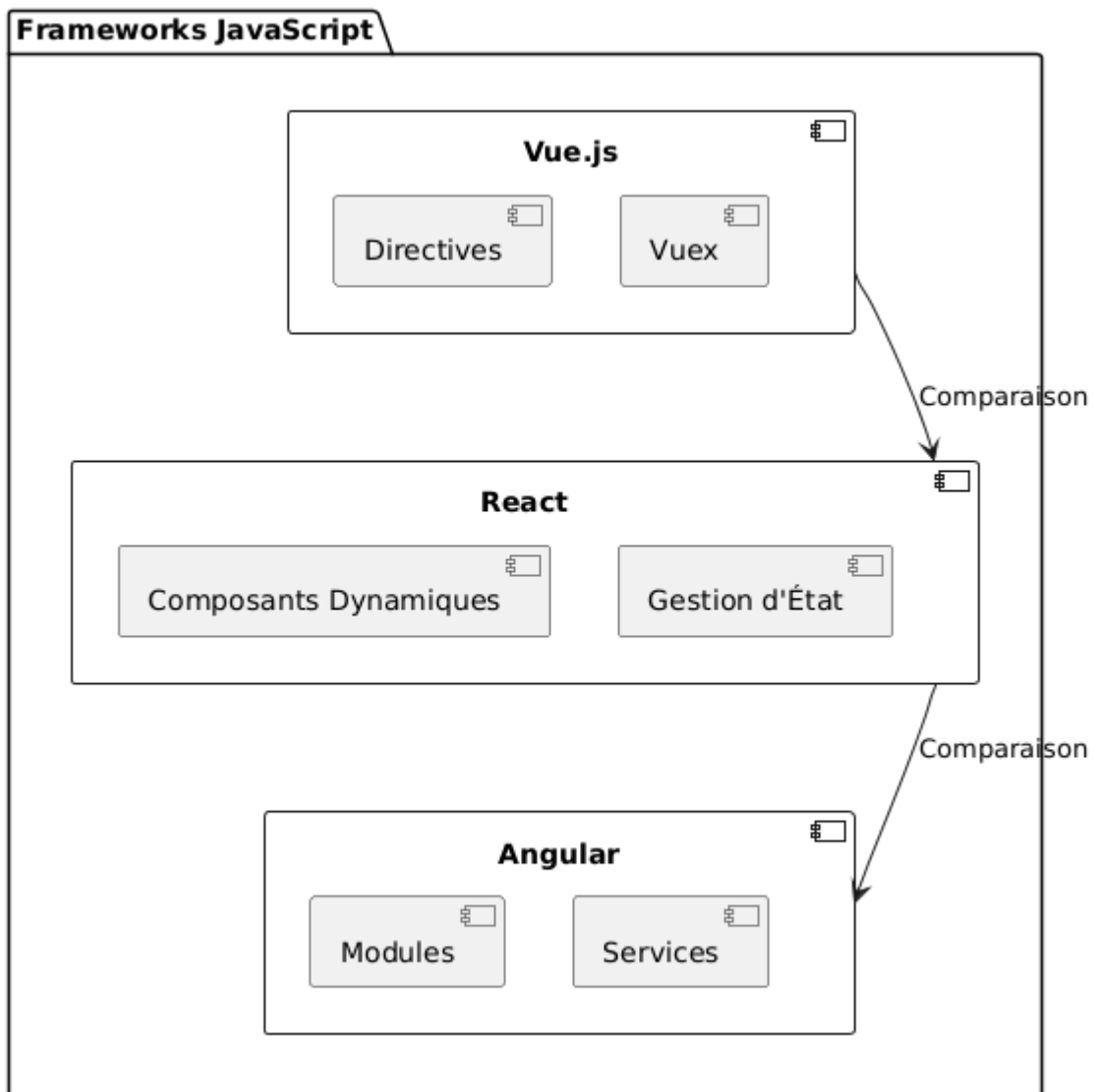
2. SGBD :

(diagrammes et explications de MySQL, SQL Server, MongoDB)



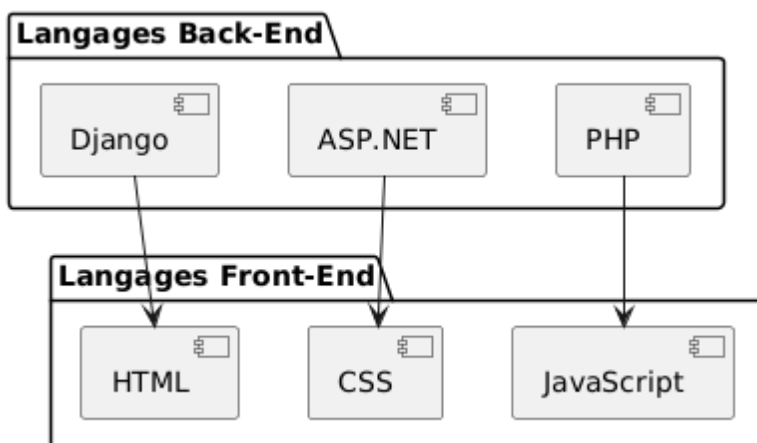
3. Frameworks JavaScript :

(diagrammes et explications de Vue.js, React, Angular)



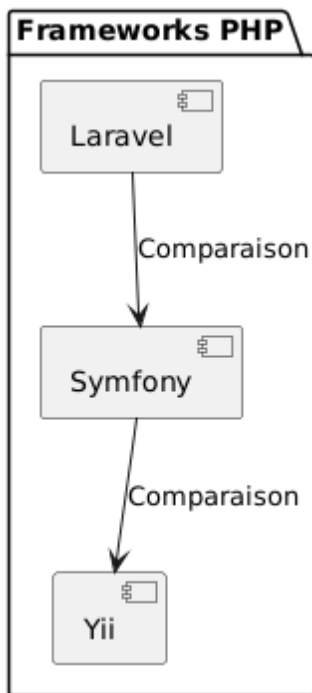
4. Langages Back-end et Front-end :

(diagrammes et explications de Django, ASP.NET, PHP)



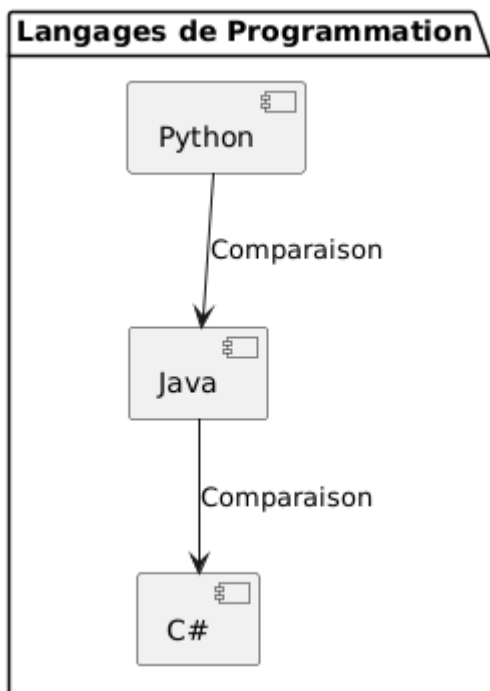
5. Frameworks PHP :

(diagrammes et explications de Laravel, Symfony, Yii)



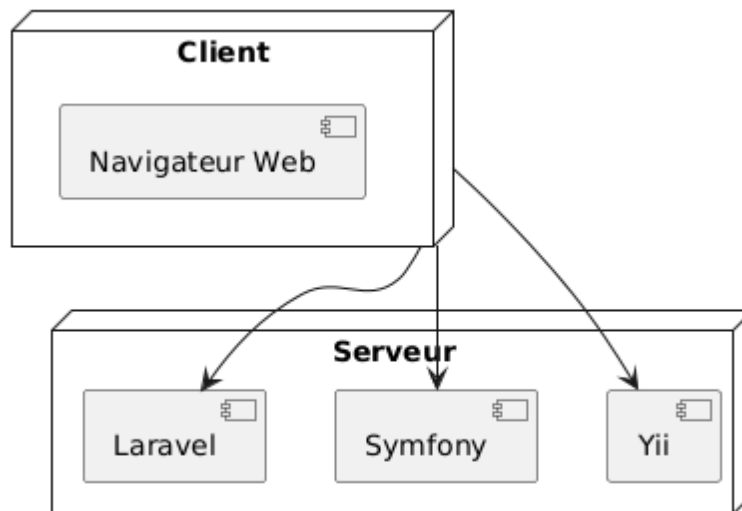
6. Langages de programmation :

(diagrammes et explications de Python, Java, C#)

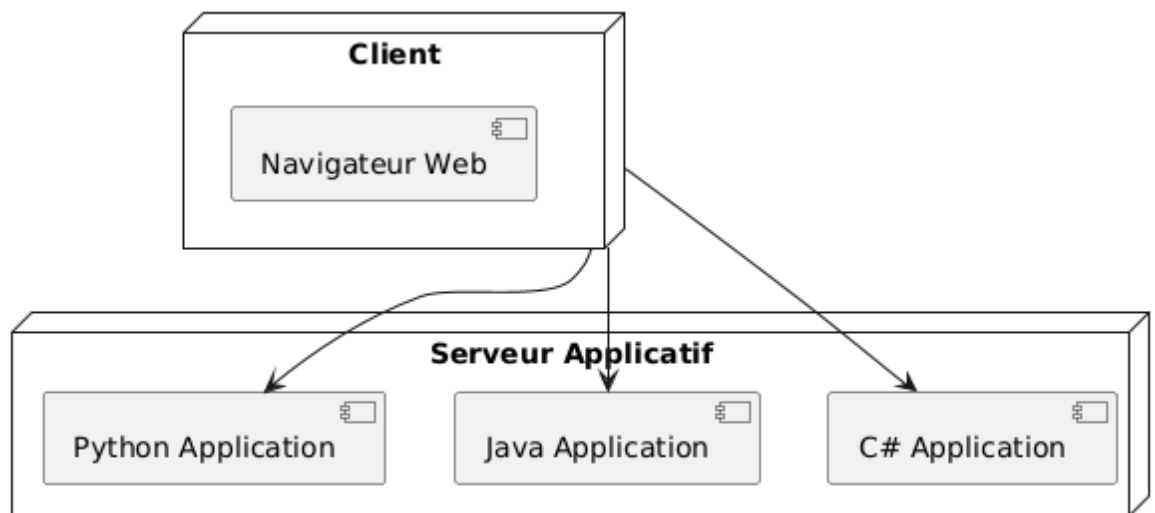


Diagrammes créés :

- **Composants :**



- **Déploiement :**



- **Analyse comparative :**

Analyse Comparative : Tableau Comparatif des Logiciels

Bien qu'il ne s'agisse pas d'un diagramme, voici un exemple de tableau comparatif à créer dans votre rapport :

Catégorie	Logiciel	Performance	Scalabilité	Sécurité	Facilité d'utilisation
-----------	----------	-------------	-------------	----------	------------------------

Systèmes d'exploitation	Windows	Haute	Moyenn	Moyenne	Bonne
	Unix	Moyenne	Haute	Haute	Moyenne
SGBD	MySQL	Haute	Moyenne	Bonne	Facile
	MongoDB	Moyenne	Haute	Moyenne	Moyenne
Frameworks JavaScript	React	Haute	Haute	Moyenne	Moyenne
	Vue.js	Moyenne	Moyenne	Moyenne	Facile
Langages Back-End	Django	Moyenne	Haute	Haute	Moyenne
	PHP	Moyenne	Moyenne	Moyenne	Facile
Frameworks PHP	Laravel	Moyenne	Moyenne	Bonne	Moyenne
Langages de Programmation	Python	Haute	Haute	Moyenne	Facile

5. Résultats et Discussion

Points forts :

- Bonne répartition des tâches et utilisation efficace de GANTTProject.
- Modélisation UML détaillée et adaptée aux besoins fonctionnels du site.

Difficultés :

- Gestion du temps pour combiner les tâches de développement et d'analyse logicielle.

Leçons apprises :

- Importance de la planification et de la collaboration avec des outils comme GitHub.

6. Conclusion et Perspectives

Le projet a permis de développer un site e-commerce fonctionnel et de mener une analyse comparative des logiciels. Les résultats sont conformes aux attentes, et des

perspectives incluent l'ajout de nouvelles fonctionnalités comme des recommandations de produits ou une gestion avancée des utilisateurs.

7. Annexes

7.1 Diagramme de GANTT du projet

7.2 Diagrammes UML du site E-commerce

- 7.2.1 Diagramme de cas d'utilisation
- 7.2.2 Diagramme de classes
- 7.2.3 Diagramme de séquences
- 7.2.4 Diagramme d'activités
- 7.2.5 Diagramme d'états-transitions
- 7.2.6 Schéma relationnel et SQL

7.3 Diagrammes des architectures logicielles analysées

- 7.3.1 Diagrammes de composants (Windows, Unix, SGBD, Frameworks, Langages)
- 7.3.2 Diagrammes de déploiement (application web, frameworks back-end/front-end)

7.4 Analyse comparative des logiciels

(Tableau comparatif : performance, sécurité, scalabilité, facilité d'utilisation, etc.)