



---

## Document de Conception

« *Projet E-PEMTShop Plateforme E-Commerce Multi-Tenant (SaaS)* »

	Etudiant	Code
<i>Préparé par</i>	Gamanuel Joissaint	JO201741
<i>Cour</i>	Technique de développement de système	
<i>Soumis au Professeur</i>	Jean Andris ADAM, Ingénieur, économiste	
<i>Date</i>	Vendredi 6 Février 2026	

---

*Document de conception*

---

## A. Identification

### A.1 Identification du document

<b>Nom du document</b>	:	Document de Conception
<b>Projet</b>	:	<b>E-PEMTShop Plateforme E-Commerce Multi-Tenant (SaaS)</b>
<b>Version du projet</b>	:	1.0
<b>Version du document</b>	:	1.1
<b>Sécurité du document</b>	:	Interne
<b>Date de création</b>	:	02/02/2026
<b>Par</b>	:	Gamanuel JOISSAINT

### A.2 Historique des changements

Qui ?	Quand ?	Quoi ?
Gamanuel JOISSAINT	25 Janvier 2026	Création initiale du document
Gamanuel JOISSAINT	25 Janvier 2026	Ajout de contenu
Gamanuel JOISSAINT	25 Janvier 2026	Mise en forme du document
Gamanuel JOISSAINT	30 Janvier 2026	Validation du contenu

### A.3 Validation

Version	Nom	Fonction	Date	Signature
V 1.0	Gamanuel Joissant	Développeur	25 Janvier 2026	
V 1.1	Gamanuel Joissant	Dev		

---

*Document de conception*

---

## Table of Contents

<b>A.</b>	<b>Identification .....</b>	<b>2</b>
A.1	Identification du document.....	2
A.2	Historique des changements.....	2
A.3	Validation .....	2
1.	Contexte de la conception.....	4
2.	Objectifs .....	4
2.1.	Objectif général .....	4
2.2.	Objectifs spécifiques .....	4
3.	Modèle et architecture .....	4
3.1.	Modèle architectural .....	4
3.2.	Architecture 3-tiers.....	5
4.	Pattern MVC .....	5
5.	Conception de la base de données.....	5
5.1.	Approche multi-tenant.....	5
5.2.	Principales entités .....	5
6.	Sécurité et gestion des accès .....	6
7.	Choix technologiques .....	6
8.	Conclusion .....	6

---

## *Document de conception*

---

### **1. Contexte de la conception**

Dans le cadre du projet **E-PEMTShop**, une plateforme e-commerce multi-tenant de type SaaS, il est nécessaire de définir une solution technique permettant de répondre aux exigences fonctionnelles et non fonctionnelles précédemment identifiées.

Le système doit permettre à plusieurs boutiques indépendantes (tenants) de partager une même infrastructure logicielle tout en garantissant une séparation logique et sécurisée de leurs données. La conception du système doit également favoriser la maintenabilité, l'évolutivité et la clarté de l'architecture, conformément aux principes du développement des systèmes d'information étudiés dans le cours **Techniques de Développement des Systèmes**.

Ce document de conception présente le **modèle architectural retenu**, les **choix technologiques**, ainsi que l'organisation générale du système.

### **2. Objectifs**

#### **2.1. Objectif général**

L'objectif général de la conception est de définir une architecture logicielle claire, structurée et sécurisée pour la plateforme **E-PEMTShop**, permettant la mise en œuvre efficace d'un système e-commerce multi-tenant conforme aux bonnes pratiques du développement des systèmes d'information.

#### **2.2. Objectifs spécifiques**

Les objectifs spécifiques de la conception sont les suivants :

- ❖ Définir une architecture **3-tiers** adaptée à une plateforme e-commerce multi-tenant.
- ❖ Mettre en place une séparation claire entre le frontend, le backend et la base de données.
- ❖ Concevoir une base de données partagée garantissant l'isolation logique des données entre tenants.
- ❖ Adopter le pattern **MVC** afin d'améliorer l'organisation et la maintenabilité du code.
- ❖ Faciliter la communication entre le frontend et le backend à travers des **API REST**.
- ❖ Utiliser des technologies modernes et adaptées au développement web (React JS, Node.js, MySQL).
- ❖ Prendre en compte les aspects de sécurité et de gestion des accès.

### **3. Modèle et architecture**

#### **3.1. Modèle architectural**

Le modèle retenu pour la conception de **E-PEMTShop** repose sur une architecture **client / serveur**, combinée à une architecture **3-tiers** et au pattern **MVC**.

---

## *Document de conception*

---

Ce modèle permet :

- Une meilleure séparation des responsabilités.
- Une évolution facilitée du système.
- Une maintenance simplifiée du code.

### **3.2. Architecture 3-tiers**

L'architecture du système est structurée en trois couches distinctes :

- **Couche Présentation**

Développée avec **React JS**, elle assure l'interface utilisateur et la gestion des interactions avec les utilisateurs (tenants et clients).

- **Couche Métier (Applicative)**

Implémentée avec **Node.js** et **Express**, elle contient la logique métier du système et expose des API REST.

- **Couche Données**

Bassée sur une base de données relationnelle **MySQL**, elle assure la persistance et la gestion des données.

## **4. Pattern MVC**

Le pattern **MVC (Model – View – Control ler)** est utilisé pour structurer le code applicatif :

- **Model** : entités métier et règles de gestion (Tenant, Utilisateur, Produit, Commande, Paiement).
- **View** : interfaces utilisateur développées avec React JS.
- **Controller** : gestion des requêtes HTTP et coordination entre les modèles et les vues.

## **5. Conception de la base de données**

### **5.1. Approche multi-tenant**

La base de données est partagée entre tous les tenants.

Chaque table métier intègre un champ **tenant\_id** permettant d'assurer l'isolation logique des données.

### **5.2. Principales entités**

Entité	Description
Tenant	Gestion des boutiques

---

## *Document de conception*

---

<b>Utilisateur</b>	Comptes utilisateurs
<b>Produit</b>	Catalogue des produits
<b>Panier</b>	Produits sélectionnés
<b>Commande</b>	Achats validés
<b>Paiement</b>	Informations de paiement

---

## 6. Sécurité et gestion des accès

Les mécanismes de sécurité intégrés à la conception sont :

- Authentification des utilisateurs et des tenants.
- Gestion des rôles et des autorisations.
- Filtrage des données par `tenant_id`.
- Simulation sécurisée du processus de paiement.

## 7. Choix technologiques

Les technologies retenues pour la mise en œuvre du système sont :

- **Frontend** : React JS
- **Backend** : Node.js avec Express
- **Base de données** : MySQL
- **Architecture** : 3-tiers
- **Pattern** : MVC
- **Communication** : API REST (JSON)

## 8. Conclusion

Le présent document de conception définit les choix architecturaux et techniques retenus pour le développement de **E-PEMTShop**. Il constitue une base solide pour la phase d'implémentation du système et garantit la cohérence avec les documents d'exigences et d'analyse précédemment établis.