IIM
DIGITAL SCHOOL
DE VINCI PARIS

Cynthia APURA

B3 Coding & Digital Innovation 2024 - 2025

Rattrapage Août 2025

UE : Cahier des charges

PICARD – Cahier des Charges Fonctionnel

Application de gestion de distributeurs réfrigérés sur les campus étudiants

SOMMAIRE

1. Contexte et objectifs du projet

1.1. Contexte

L'entreprise Picard, acteur majeur de la distribution de produits alimentaires surgelés en France, dispose d'un réseau d'environ 1000 points de vente, répartis sur le territoire national. Dans un contexte de diversification des canaux de distributions et face à l'évolution des comportements de consommation, Picard souhaite adresser une nouvelle cible stratégique, les étudiants.

Pour cela, l'entreprise envisage le déploiement de distributeurs automatiques réfrigérés sur les campus universitaires. Ces distributeurs ont pour vocations de proposer une sélection de produits surgelés adaptés aux besoins, aux horaires et aux contraintes des étudiants.

Pour gérer efficacement ces distributeurs sur les campus, Picard souhaite développer une application web pensée pour centraliser toutes les opérations. Celle-ci permet notamment de gérer les stocks, de consulter les produits disponibles, d'assurer un suivi des performances et d'offrir une interface adaptée à différents profils utilisateurs (étudiants, techniciens, gestionnaires, etc..).

1.2. Objectifs

L'objectif principal du projet est la conception et le développement d'une application web centralisé qui permettra à Picard de :

- Gérer à distance les produits (ajout, modification, suppression, mise à jour des stocks) présents dans chaque distributeur
- Suivre l'état des stocks en temps réel, afin de garantir la disponibilité des produits et faciliter la logistique
- Collecter les données d'utilisation (notations, préférences utilisateurs) en vue d'améliorer l'offre
- Offrir une interface utilisateur simple et intuitive à destination des étudiants pour consulter les produits disponibles et interagir avec les distributeurs

L'objectif global est à la fois pratique et stratégique, d'un côté mieux gérer les distributeurs (suivi, réapprovisionnement, supervision) et de l'autre créer un lien direct entre Picard et les étudiants via un canal moderne et autonome.

2. Description des fonctionnalités principales

L'application devra intégrer les fonctionnalités suivantes, réparties selon les profils utilisateurs et les besoins opérationnels de Picard. De plus, l'application devra gérer les différents profils utilisateurs avec des droits d'accès spécifiques. Un système d'alertes et de notifications permettra également de signaler tout incident technique ou besoin de réapprovisionnement, assurant ainsi une maintenance proactive des distributeurs et de leurs stocks.

2.1. Accès gestionnaire

L'application offrira aux gestionnaires une interface leur permettant de superviser l'ensemble du réseau de distributeurs. Les fonctionnalités incluront :

- La gestion du catalogue produit (ajout, modification, suppression)
- La mise à jour des quantités en stocks pour chaque distributeur
- Un tableau de bord de suivi en temps réel, incluant des alertes de seuil critique

Des rapports statistiques et des indicateurs de performance (KPI) permettront d'analyser les ventes et d'adapter l'offre en fonction des comportements d'achat observés.

2.2. Accès utilisateur

L'interface étudiante sera conçue pour une utilisation simple axée sur la rapidité d'accès aux produits. Elle permettra :

- La consultation des produits disponibles dans les distributeurs géolocalisés aux alentours
- L'accès à une fiche produit détaillée (nom, description, prix, note, disponibilité)
- La notation et l'évaluation des produits

En complément, un compte utilisateur permettra d'avoir un système de fidélité, l'historique des achats, et des recommandations personnalisées.

2.3. Accès technicien

Un espace spécifique sera mis à disposition pour les techniciens de maintenance. Celui-ci proposera :

- Une visualisation en temps réel de l'état de chaque distributeur (température, pannes, état de charge)
- Un système de notifications automatiques en cas d'anomalie
- Un module de gestion des interventions, incluant l'historique des opérations techniques effectuées

L'objectif est d'assurer une maintenance proactive, minimisant les interruptions de service et garantissant la qualité de conservation des produits. Cette fonction est essentielle pour maintenir la fiabilité du réseau et la satisfaction des utilisateurs finaux.

3. Besoins techniques et contraintes

L'application web devra répondre à un certain nombre d'exigences techniques pour garantir sa fiabilité, sa maintenabilité et sa compatibilité avec les usages visés. Ces spécifications visent à assurer un socle technique robuste, tout en respectant les contraintes propres à un déploiement sur le terrain.

3.1. Architecture technique

- **Back-end** : API RESTful développée en Symfony 6 avec API Platform, assurant une exposition standardisée et évolutive des ressources métier
- **Front-end** : Interface développée avec Vue.js, responsive, mobile-first conçue pour une utilisation fluide sur bureau ou mobile
- **Base de données** : Utilisation de PostgreSQL, pour répondre aux besoins métiers (produits, utilisateurs, stocks, interventions techniques, etc.)
- **Communication front/back :** Les échanges s'effectuent via des appels HTTP sécurisés, utilisant le format JSON
- **Authentification** : Gestion par tokens JWT, avec un contrôle des rôles utilisateur intégré

3.2. Contraintes techniques

- **Fonctionnement hors-ligne** : L'application doit tolérer les pertes de connectivité temporaires au niveau des distributeurs (mise en cache locale, synchronisation à la reconnexion)
- **Sécurité** :
    - Authentification par rôle avec un contrôle des accès pour chaque utilisateur (gestionnaire, technicien, étudiant)

- Données chiffrées en transit (HTTPS obligatoire)
- RGPD : traitement des données personnelles conforme à la réglementation européenne (stockage, consentement, anonymisation)

- **Supervision :** Intégration d'outils de suivi (logs, alertes, techniques, disponibilité des services)

3.3. Compatibilité

L'application doit être compatible avec les principaux navigateurs web modernes (Chrome, Firefox, Edge, Safari) et adaptée aux écrans mobiles et tablettes pour garantir un accès étendu.

3.4. Hébergement

L'application sera initialement développée et testée dans un environnement local, afin de garder la main sur l'ensemble du processus, de contrôler chaque étape et de valider toutes les fonctionnalités avant de penser au déploiement.

Par la suite, un déploiement sur une plateforme cloud pourra être envisagé. La solution recommandée pour ce déploiement sera Render, reconnue pour sa simplicité d'utilisation et sa compatibilité avec les technologies choisies.

Enfin pour une utilisation plus performante et plus unifiée, nous pourrons dockerisé l'intégralité du projet.

Cette organisation garantit un déroulement efficace et sécurisé du projet, tout en laissant la porte ouverte à une mise en production accessible et évolutive.

3.5. Exigences non fonctionnelles

Au-delà des contraintes techniques, l'application devra répondre à plusieurs exigences non fonctionnelles destinées à garantir la performance, la pérennité et la qualité globale du service :

- **Scalabilité :** L'architecture doit être conçue de manière modulable, afin de permettre l'ajout de futures fonctionnalités (paiement en ligne, gestion avancée des comptes utilisateurs etc…)
- **Accessibilité :** L'interface utilisateur devra être conforme aux normes standard WCAG niveau AA, afin de garantir une utilisation inclusive, y compris pour les personnes en situation de handicap
- **Maintenabilité :** Le code devra être structuré et documenté pour permettre une prise en main rapide par une nouvelle équipe. L'usage de bonnes pratiques est requis. (tests unitaires, documentation API, commentaire clairs)

- **Sécurité passive :** Le système devra limiter les vecteurs d'attaques potentiels (injection, XSS, CSRF) par l'usage de frameworks reconnus et la mise en œuvre de standard de sécurité

4. Utilisateurs et leurs rôles

L'application web cible trois profils utilisateurs principaux, chacun disposant de droits d'accès adaptés à ses besoins et responsabilités dans la gestion des distributeurs Picard.

- **Gestionnaire :** Responsable de la supervision et de la gestion opérationnelle des distributeurs, incluant le pilotage des stocks et de l'analyse des performances commerciales. Ils disposent de droits étendus pour administrer le catalogue produit et configurer les alertes
- **Techniciens :** Chargés de la maintenance et de la surveillance technique des distributeurs. Ils ont accès à des outils spécifiques pour suivre l'état des machines, recevoir des notifications d'incident, et enregistrer leurs interventions
- **Étudiants (utilisateurs finaux) :** Utilisateurs principaux de l'application pour consulter les produits, effectuer des achats, gérer leur compte fidélité, et contribuer à l'amélioration de l'offre via leurs évaluations

5. Parcours utilisateurs

5.1. Étudiants

Contexte : L'étudiant utilisa l'application pour localiser un distributeur, consulter les produits disponibles et éventuellement donner son avis.

Parcours type :

1. Connexion à l'interface utilisateur via navigateur web ou smartphone
2. Géolocalisation automatique ou sélection manuelle du distributeur le plus proche
3. Visualisation des produits disponibles (tri par type, prix, popularité…)
4. Sélection d'un produit → affichage de la fiche détaillée
5. Éventuellement, notation du produit ou ajout à un liste de favoris
6. Consultation de son historique d'achat ou des points de fidélités

5.2. Gestionnaire

Contexte: Le gestionnaire pilote le parc de distributeurs, surveille les ventes et adapte l'offre.

Parcours type:

1. Connexion à l'espace gestionnaire sécurisé
2. Accès au tableau de bord (vue d'ensemble des stocks, alertes, ventes)
3. Consultation ou modification du catalogue produits
4. Analyse des indicateurs (KPI) par distributeur ou segment
5. Prise de décision: ajout d'un nouveau produit, ajustement des quantités ou envoie d'un message à un technicien

5.3. Technicien

Contexte: Le technicien intervient sur les distributeurs (réapprovisionnement ou panne).

Parcours type:

1. Connexion à son espace technicien
2. Accès aux notifications (panne détectée, seuil critique atteint)
3. Consultation des logs techniques du distributeur concerné
4. Saisie d'un rapport d'intervention via formulaire intégré
5. Marquage de la tâche comme résolue

6. Livrables attendus

À l'issue du projet, les livrables suivants devront être produits :

6.1. Livrables techniques

- Application web complète :
    - Front-end Vue.js (interface étudiante, gestionnaire, technicien)
    - API Symfony + API Platform (exposition des ressources métiers)
    - Base de données PostgreSQL fonctionnelle
    - Authentification par JWT

6.2. Documentation

- Cahier des charges fonctionnel *(présent document)*
- Documentation technique :

- Schéma d'architecture
- Instructions d'installation et de déploiement
- Documentation API (routes, méthodes, formats, erreurs)

- Guide de maintenance (recommandations techniques, mises à jour, logs)

6.3. Autres

- Manuel utilisateur *(livrable à part)* : description simple des parcours utilisateur
- Glossaire *(présent à la fin de ce document)*

7. Glossaire

API (Application Programming Interface) - Interface permettant à différentes applications de communiquer entre elles. Dans ce projet, l'API expose les fonctionnalités de gestion des produits, des stocks et des utilisateurs pour les distributeurs Picard.

API Platform - Framework PHP basé sur Symfony, facilitant la création d'API REST conformes aux standards, avec documentation automatique, sécurité, et sérialisation intégrée.

Back-end - Partie serveur de l'application. Il traite de la logique métier, des accès aux données et des communications avec le front-end.

Front-end - Interface graphique visible et manipulée par les utilisateurs.

JWT (JSON Web Token) - Méthode d'authentification sécurisée permettant de gérer les sessions utilisateurs avec un système de jetons cryptés.

Vue.js - Framework JavaScript open source pour le développement d'interfaces utilisateur dynamiques.

KPI (Key Performance Indicators) - Indicateurs clés permettant de mesurer la performance.

Base de données (PostgreSQL) - Système de gestion des données utilisé pour stocker toutes les informations liées aux produits, utilisateurs, stocks et opérations techniques.

Scalabilité - Capacité d'un système informatique à absorber une augmentation de charge (nombre d'utilisateurs, volume de données, nombre de fonctionnalités) sans dégradations significatives des performances.

Accessibilité (WCAG) - Ensemble de normes internationales (Web Content Accessibility Guidelines) définissant les règles pour rendre les contenus web utilisables par tous, y compris les personnes en situation de handicap. Le niveau AA garantit un bon compromis entre accessibilité et faisabilité technique.

Injection - Type d'attaque dans laquelle des requêtes malveillantes sont insérées dans un champ de saisie utilisateur afin de manipuler ou compromettre la base de données.

XSS (Cross-Site Scripting) - Vulnerabilité permettant à un attaquant d'injecter du code JavaScript malveillant dans une page web consultée par d'autres utilisateurs.

CSRF (Cross-Site Request Forgery) - Attaque consistant à tromper un utilisateur authentifié pour qu'il exécute une action involontaire sur une application web, sans son consentement explicite.