

PLAN

Introduction

I- ANALYSE FONCTIONNELLE

- Analyse des besoins
- Présentation du projet
- Dictionnaire des données
- Règles de gestion

II- CONCEPTION DE LA BASE DE DONNÉE

- Modèle conceptuel des données (MCD)
- Couverture Minimale (CM)
- Schéma Logique Relationnel (SLR)

III- RÉALISATION TECHNIQUE

- Architecture matérielle et logicielle
- Interface utilisateur (UI/UX)
- Scénario de Démo

CONCLUSION

RAPPORT – GESTION AUTOMATISÉE D'UNE PME SPÉCIALISÉE DANS LA PRODUCTION ET LA VENTE DE GADGETS ANIMALIERS

INTRODUCTION

Dans un contexte de transformation numérique, les petites et moyennes entreprises (PME) doivent moderniser leurs méthodes de gestion afin d'améliorer leur productivité et la traçabilité de leurs activités.

Le présent projet s'inscrit dans cette dynamique et vise à concevoir un système informatisé permettant la gestion automatisée d'une PME spécialisée dans la production et la vente de gadgets animaliers (croquettes, jouets, nounours, etc.).

L'objectif principal est de modéliser, concevoir et proposer une solution technique permettant de gérer efficacement les activités suivantes :

- La production des gadgets
- La gestion des stocks
- Les ventes et la facturation
- La gestion du personnel

La méthode MERISE a été utilisée pour la modélisation des données, suivie d'une conception technique basée sur une architecture web utilisant des technologies comme React et NodeJS.

I – ANALYSE FONCTIONNELLE

1. Analyse des besoins

L'entreprise concernée exerce deux activités principales :

1. La production de gadgets animaliers
2. La vente de ces gadgets aux clients

Les besoins identifiés sont les suivants :

- Enregistrer les gadgets produits
- Gérer les quantités produites

- Mettre à jour automatiquement le stock
- Enregistrer les ventes effectuées
- Générer des factures
- Suivre les statistiques de vente
- Gérer les employés selon leurs rôles

Le système doit permettre une gestion fiable, structurée et traçable des opérations.

2. Présentation du projet

Le projet consiste à développer une application web permettant :

- La gestion des produits
- L'enregistrement des productions
- La gestion des ventes
- La génération des factures
- L'affichage de statistiques
- La gestion hiérarchique des employés

L'application devra différencier les rôles suivants :

- Caissier
- Agent de production
- Administratif

Chaque rôle dispose de permissions spécifiques.

3. Dictionnaire des données

GADGET

- id_gadget : Identifiant unique du gadget
- nom_gadget : Nom du produit
- type_gadget : Catégorie du produit
- prix_vente_gadget : Prix de vente
- description_gadget : Description détaillée

CLIENT

- id_client : Identifiant unique

- nom_client : Nom
- prenom_client : Prénom
- email_client : Adresse email
- contact_client : Numéro de téléphone
- adresse_client : Adresse

EMPLOYE

- matricule_employe : Identifiant unique
- nom_employe : Nom
- prenom_employe : Prénom
- email_employe : Email
- adresse_employe : Adresse
- matricule_employe_superviseur : Référence au superviseur

PRODUCTION

- id_production : Identifiant unique
- date_production : Date de fabrication
- quantite_produite : Quantité produite

VENTE

- id_vente : Identifiant unique
- date_vente : Date de la vente

FACTURE

- id_facture : Identifiant unique
 - numero_facture : Numéro de facture
 - date_facture : Date
 - montant_total : Montant total
-

4. Règles de gestion

◆ 1. Gestion des Gadgets

1. RG1 : Chaque gadget est identifié par un identifiant unique.
RG2 : Un gadget possède un nom, un type, un prix de vente et une description.

RG3 : Un gadget peut être produit plusieurs fois ou ne jamais être produit.

RG4 : Un gadget peut apparaître dans plusieurs ventes ou dans aucune.

◆ 2. Gestion de la Production

2. RG5 : Chaque production est identifiée par un identifiant unique.

RG6 : Une production concerne un seul gadget.

RG7 : Une production doit avoir une date et une quantité produite.

RG8 : Une production implique au minimum un agent de production.

RG9 : Un agent de production peut participer à plusieurs productions.

RG10 : La durée de participation d'un agent à une production est enregistrée.

◆ 3. Gestion des Stocks

3. RG11 : Chaque gadget possède un stock.

RG12 : Le stock enregistre la quantité disponible d'un gadget.

RG13 : La quantité disponible augmente après une production.

RG14 : La quantité disponible diminue après une vente.

◆ 4. Gestion des Clients

4. RG15 : Chaque client est identifié par un identifiant unique.

RG16 : Un client peut effectuer plusieurs ventes.

RG17 : Une vente est obligatoirement associée à un seul client.

◆ 5. Gestion des Ventes

5. RG18 : Chaque vente est identifiée par un identifiant unique.

RG19 : Une vente possède une date.

RG20 : Une vente est enregistrée par un seul caissier.

RG21 : Un caissier peut enregistrer plusieurs ventes.

RG22 : Une vente contient au moins un gadget.

RG23 : Pour chaque gadget vendu, on enregistre la quantité vendue et le prix unitaire appliqué.

◆ 6. Gestion des Factures

6. RG24 : Chaque facture est identifiée par un identifiant unique.

RG25 : Chaque facture est associée à une seule vente.

RG26 : Une vente donne lieu à une seule facture.

RG27 : Une facture possède un numéro, une date et un montant total.

◆ 7. Gestion des Employés et Héritage

7. RG28 : Chaque employé possède un matricule unique.

RG29 : Chaque employé appartient obligatoirement à une seule spécialisation :

8. Caissier

9. Agent de production

10.Administratif

11.(Héritage de type XT : partition + totalité)

12.RG30 : Seul un caissier peut enregistrer une vente.

RG31 : Seul un agent de production peut participer à une production.

RG32 : Un administratif peut superviser plusieurs employés.

RG33 : Un employé peut être supervisé par au maximum un administratif.

II – CONCEPTION DE LA BASE DE DONNÉE

1. Modèle Conceptuel des Données (MCD)

Le MCD a été élaboré selon la méthode MERISE afin de représenter les entités du système et leurs relations indépendamment de toute contrainte technique.

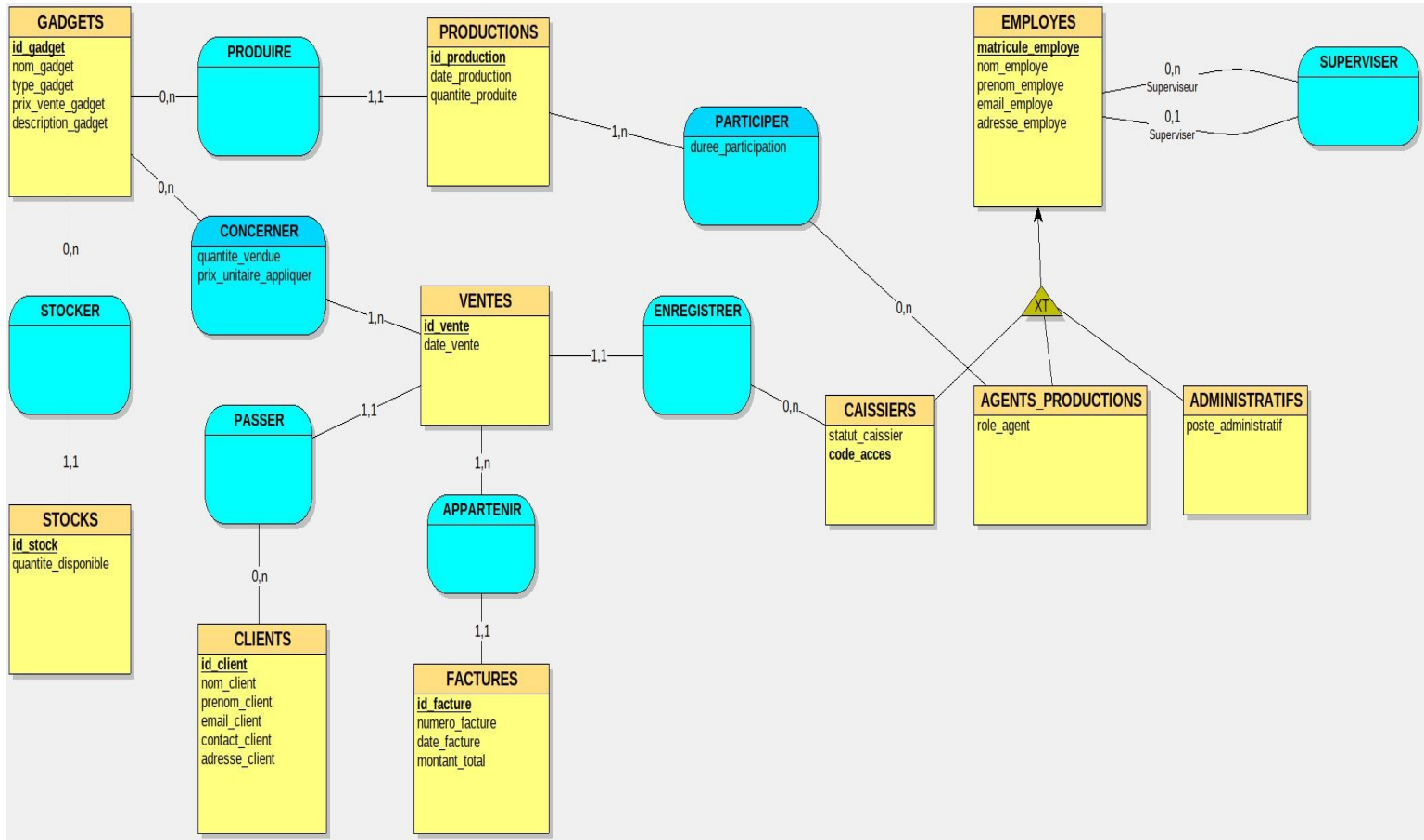
Les entités principales identifiées sont :

- GADGET
- CLIENT
- EMPLOYE
- PRODUCTION
- VENTE
- FACTURE
- STOCK

Les associations permettent de représenter :

- La participation des agents à la production
- L'enregistrement des ventes par les caissiers
- La relation entre vente et gadgets
- La supervision hiérarchique

L'héritage de l'entité EMPLOYE est de type partition et totalité (XT).



2. Couverture Minimale (CM)

La couverture minimale consiste à simplifier l'ensemble des dépendances fonctionnelles en supprimant les redondances et en garantissant que chaque dépendance possède :

- Un seul attribut à droite
- Aucun attribut inutile à gauche

Cette étape permet d'obtenir un schéma relationnel optimisé et cohérent.

3. Schéma Logique Relationnel (SLR)

Le passage du MCD au SLR a permis d'obtenir les tables relationnelles suivantes :

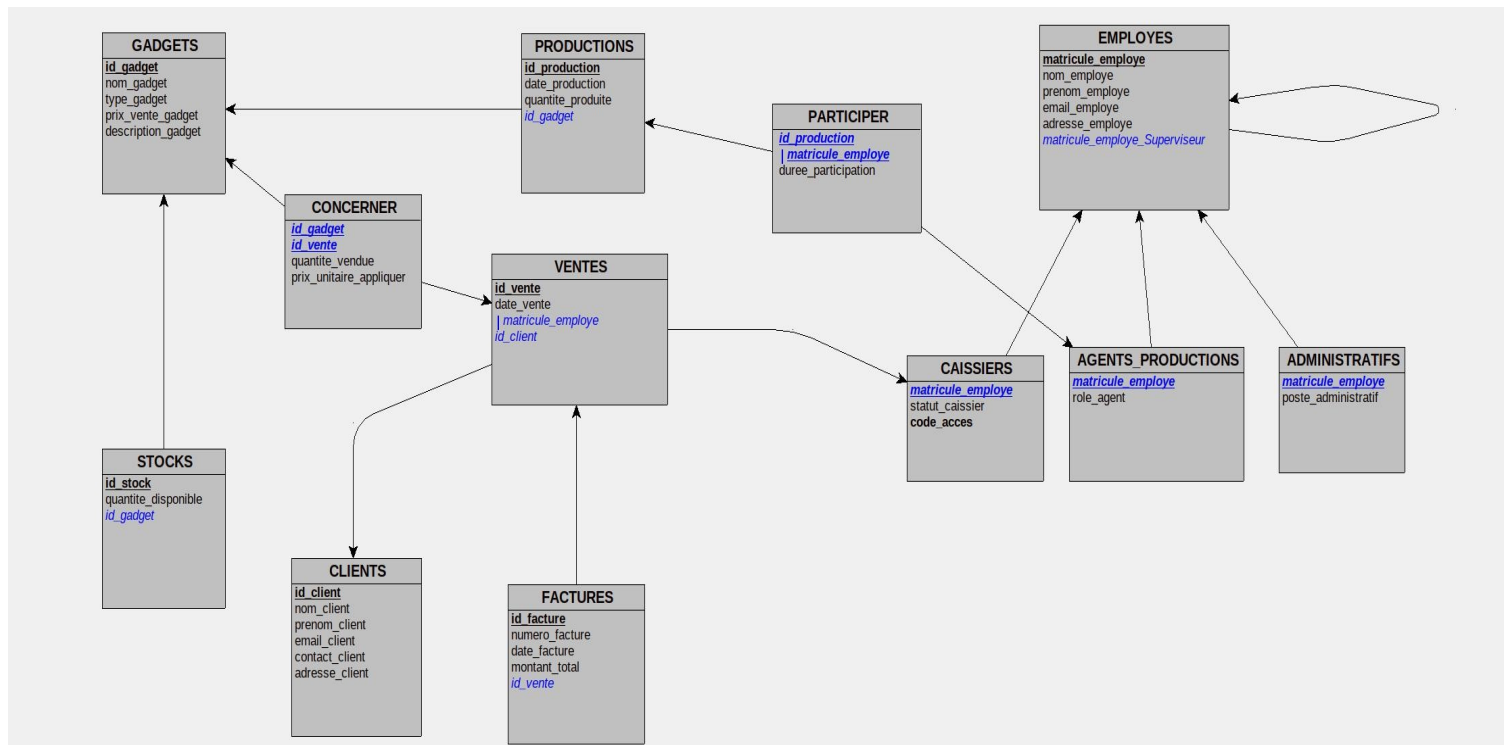
- GADGETS
- CLIENTS
- EMPLOYES

- CAISSIERS
- AGENTS_PRODUCTIONS
- ADMINISTRATIFS
- PRODUCTIONS
- VENTES
- FACTURES
- STOCKS
- PARTICIPER
- CONCERNER

Les clés primaires et étrangères assurent l'intégrité référentielle de la base.

Le SLR (ou Modèle Relationnel) traduit le conceptuel en tables physiques :

- **EMPLOYES** (<u>matricule_employe</u>, nom, prenom, email, #matricule_superviseur)
- **CAISSIERS** (<u>#matricule_employe</u>, statut, code_accès)
- **GADGETS** (<u>id_gadget</u>, nom, type, prix_vente, description)
- **CONCERNER** (<u>#id_gadget, #id_vente</u>, quantite_vendue, prix_unitaire_appliquer)
- **STOCKS** (<u>id_stock</u>, quantite_disponible, #id_gadget)



III – RÉALISATION TECHNIQUE

1. Architecture logicielle

Environnement logiciel :

- Pour garantir une expérience fluide sans rechargement de page (SPA), nous avons choisi :
- **Frontend : React** avec **React-Router** pour une navigation instantanée.
- **Backend : Hono** (Node.js), un framework ultra-rapide pour l'**API REST**.
- **Base de données : MySQL 8.0**.
- **Visualisation : Chart.js** (statistiques de vente) et **D3.js** (organigramme hiérarchique).

Architecture :

Le système adopte une architecture de type MVC simplifiée :

- **Modèle** : gestion base de données
- **Vue** : interface utilisateur
- **Contrôleur** : logique métier

2. Interface utilisateur (UI/UX)

L'interface est organisée autour :

- D'un tableau de bord principal
- D'une barre latérale de navigation
- De pages dédiées à chaque module

Modules disponibles :

- Gestion des gadgets
- Gestion des productions
- Gestion des ventes
- Gestion des factures
- Gestion des employés
- Statistiques

Le tableau de bord affiche :

- Nombre total de ventes
- Quantité produite
- État du stock
- Graphiques des ventes mensuelles

L'interface vise la simplicité, la clarté et l'efficacité.

3. Scénario de Démo

1. Connexion d'un administrateur
2. Ajout d'un gadget
3. Enregistrement d'une production
4. Mise à jour automatique du stock
5. Connexion d'un caissier
6. Enregistrement d'une vente
7. Génération automatique d'une facture
8. Consultation des statistiques

Ce scénario démontre la cohérence et la complétude du système.

Afin de valider le système, des données de test ont été injectées :

- **Catalogue** : Croquettes Premium, Jouets Laser Connectés, Peluches Smart.
 - **Alertes** : Simulation d'un stock critique (ex: 2 unités restantes) pour tester les notifications visuelles.
 - **Ventes** : Historique sur 7 jours injecté pour démontrer la réactivité des graphiques animés.
-

CONCLUSION

Ce projet a permis de mettre en œuvre les concepts fondamentaux du génie logiciel, notamment l'analyse des besoins, la modélisation des données avec MERISE et la conception d'une solution technique adaptée.

Le système proposé assure une gestion structurée de la production, des ventes et du personnel d'une PME spécialisée dans les gadgets animaliers.

La modélisation réalisée constitue une base solide pour un développement complet et évolutif du système, pouvant à terme être enrichi par de nouvelles fonctionnalités telles que la gestion des fournisseurs ou une extension vers une application mobile.
