**Université Abdelmalek Essaadi**

**Faculté des Sciences de Tétouan**

**Département d’Informatique**

**Année universitaire 2024-2025**



N° d’ordre : …………..

**RAPPORT**

**De Projet de Fin d'Études**

**Présenté en vue de l'obtention de la**

**LICENCE FONDAMENTALE**

**Spécialité : Sciences Mathématiques et Informatique**

**Par :**

* **Anass Harrou**
* **Fatima Zahrae Temsamani**
* **Nada El ghmari**

***Titre***

**PANIER INTELLIGENT: VERS UNE EXPERIENCE**

**D’ACHATS SANS PASSAGE EN CAISSE.**

**Soutenu le 23 Juin 2025 devant le jury composé de :**

|  |  |
| --- | --- |
| **Pr.** ………………………….…………………….... | **Président/ Examinateur** |
| **Pr.** ………………………….…………………….... | **Examinateur** |
| **Pr. Abderrahim EL MHOUTI**……………………………. | **Encadrant** |

**Dédicaces**

Du profond du mon cœur je dédie ce travail :

A nos parents :

pour leur présence constante à mes côtés,leur soutien moral et matériel inestimable

et pour les innombrables sacrifices que vous avez faits afin de me permettre de

poursuivre mes rêves ,Ce travail est le fruit de vos efforts autant que des miens.

a mon frère et ma sœur:

Votre étiez toujours présents pour m'aider et m'encourager, je vous souhaite une vie pleine de bonheur et de succès à mes chers amis qui sans leur encouragement ce travail n'aura jamais vu le jour

**Fatima zahrae temsamani**

Je dédie ce modeste travail et ma profonde gratitude :

À mon père et ma mère :

Autant de phrases aussi expressives soient-elles ne sauraient montrer le degré d’amour et d’affection que j’éprouve pour eux. Merci pour tout l’amour et la confiance et pour votre énorme support pendant la réalisation de mon projet. À tous mes amis et mes collègues qui m’ont soutenu. À tous mes professeurs pendant mon parcours de l’école vers l’université.

**Nada elghmari**

A mes chers parents :

Tous les mots du monde ne sauraient exprimer l’immense amour que je vous porte, ni la profonde gratitude que je vous témoigne pour tous les efforts et les sacrifices que vous n’avez jamais cessé de consentir pour mon instruction et mon bien-être.

**Anas harrou**

**Remerciement**

Avant tout, nous remercions Dieu Tout-Puissant, qui nous a donné la force, la patience et la santé nécessaires pour mener ce travail à son terme.

Nous adressons nos remerciements les plus sincères à Monsieur Abderrahim El Mhouti, notre encadrant de Projet de Fin d’Études, pour sa disponibilité,

son accompagnement rigoureux et la richesse de ses conseils tout au long de cette expérience.

Nous tenons également à remercier l’ensemble de l’équipe pédagogique de la Faculté des Sciences de Tétouan – Université Abdelmalek Essaâdi

pour la qualité de l’enseignement dispensé et l’encadrement bienveillant dont nous avons bénéficié tout au long de notre formation.

Enfin, un grand merci aux membres du jury pour le temps qu’ils nous ont accordé et pour l’attention portée à notre travail.

**Résumé**

Ce projet vise à développer un panier intelligent capable de détecter automatiquement les produits ajoutés par le client grâce à un lecteur RFID, des capteurs et une caméra intelligente. L’objectif principal est de faciliter le processus d’achat en affichant directement le montant total sur une tablette embarquée, sans passer par la caisse. Le système comprend également une gestion automatisée du stock et une interface web dédiée aux administrateurs. Ce projet combine matériel embarqué et développement logiciel pour offrir une solution moderne, rapide et efficace à la fois pour les clients et les responsables de magasin.

**Liste des abréviations**

|  |  |
| --- | --- |
| **Abréviation** | **Désignation** |
| UML | Unified Modeling Language |
| XML | HyperText Markup Language |
| MYSQL | My. Structured Query Language |
| PHP | HypertextPreprocessor |
| HTML | HyperText MarkupLanguage |
| CSS | Feuilles de style en cascade |
| JS | Java Script |

*Tableau1:Liste des abréviations*

**Liste des figures**

Figure1: diagramme de gant...........................................................................................................

Figure2:Cycle de vie globale du projet..........................................................................................

Figure3:Cycle de vie d’une seule sprint.........................................................................................

Figure4:UML logo.........................................................................................................................

Figure5:diagramme de cas d’utilisation du client .........................................................................

Figure6:diagramme de cas d’utilisation du gestionnaire du stock.................................................

Figure7:diagramme de cas d’utilisation d’administrateur..............................................................

Figure8:diagramme de cas d’utilisation de client...........................................................................

Figure9:diagramme de classes du système du panier intelligent....................................................

Figure10:diagramme de séquence de cas d’utilisation «demander facture»..................................

Figure11:diagramme de séquence de cas d’utilisation «ajouter produit»......................................

Figure12:diagramme de séquence de cas d’utilisation «ajouter un compte client».......................

Figure13:diagramme de séquence de cas d’utilisation « payer par code QR»...............................

Figure14:Architecture matérielle...................................................................................................

Figure15:Architecture logicielle....................................................................................................

Figure16:MCD du système............................................................................................................

Figure17:MLD du système............................................................................................................

Figure18:php logo..........................................................................................................................

Figure19:Laravel logo...................................................................................................................

Figure20:HTML logo....................................................................................................................

Figure21:CSS logo.........................................................................................................................

Figure22:Boostrap logo.................................................................................................................

Figure23:MYSQL logo..................................................................................................................

Figure24:Visual studio code..........................................................................................................

Figure25:PyCharm logo................................................................................................................

Figure26:Android studio code logo..............................................................................................

Figure27:XAMPP logo.................................................................................................................

Figure28:Python logo...................................................................................................................

Figure29:OpenCV logo................................................................................................................

Figure30:Numpy logo.................................................................................................................

Figure31:Pyzbar logo..................................................................................................................

Figure32:Android IDE logo........................................................................................................

**Liste des tableaux**

Tableau1:Liste des abréviations..............................................................................................................

Tableau2:tableau des tâche.....................................................................................................................

Tableau3:tableau des phases..................................................................................................................

**Sommaire**

[Introduction Générale 11](#_Toc19694)

[Chapitre I : 13](#_Toc14219)

[Présentation et cadrage du projet 13](#_Toc1846)

[1. Présentation du sujet 14](#_Toc27802)

[2. Contexte et définition du projet: 14](#_Toc30323)

[3. Problématique : 14](#_Toc17740)

[4. Objectifs : 16](#_Toc1091)

[5. Etude de l’existant 17](#_Toc12997)

[1. Amazon Dash Cart (États-Unis - 2020) 17](#_Toc23352)

[2. Caper AI Smart Cart (États-Unis - 2019) 17](#_Toc9063)

[4. JD.com & 7Fresh (Chine - 2018) 18](#_Toc11768)

[5.Mishipay & Auchan Scan & Go (Europe - 2022). 18](#_Toc11740)

[6. Produit du projet (Solution Proposée) 18](#_Toc13464)

[7. Fonctionnalités du produit : 18](#_Toc7576)

[8. Besoins fonctionnels 19](#_Toc31149)

[9. Besoins non fonctionnels 20](#_Toc13346)

[Chapitre II : 23](#_Toc7384)

[Déroulement du projet 23](#_Toc19940)

[1. Planification opérationnelle 24](#_Toc7910)

[2. Processus de développement 26](#_Toc20061)

[Chapitre III : 30](#_Toc11191)

[Spécification et modélisation 30](#_Toc5851)

[1. Les acteurs de l’application 31](#_Toc14469)

[2. Langage de modélisation UML 31](#_Toc9270)

[3. Diagramme de cas d’utilisation 32](#_Toc16710)

[4. Diagramme de classes 35](#_Toc4891)

[5. Diagramme de séquences 37](#_Toc1598)

[6. Conception 42](#_Toc2018)

[Chapitre VI : 46](#_Toc29434)

[Implémentation et réalisation 46](#_Toc7701)

[1. Outils de développement 47](#_Toc25164)

[2. Principales interfaces 54](#_Toc25858)

[Conclusions et perspectives 54](#_Toc20224)

[Références 55](#_Toc23060)

# Introduction Générale

Le développement des nouvelles technologies a profondément transformé le secteur du commerce, en particulier à travers l’intégration de solutions intelligentes visant à améliorer l’expérience client et à optimiser la gestion interne des points de vente. Dans ce contexte, le projet de panier intelligent propose une approche innovante pour automatiser le processus d’achat en magasin.

Ce panier est équipé de capteurs de poids, d’un lecteur RFID, d’une caméra intelligente et d’une tablette embarquée. Ces éléments permettent d’identifier automatiquement les produits ajoutés ou retirés du panier, de calculer en temps réel le montant total des achats, et d’éviter le passage en caisse traditionnel. Cela représente un gain de temps important pour le client et une réduction des erreurs de facturation.

En parallèle, ce système permet également une meilleure gestion des stocks, une suivi en temps réel des articles et une interface d’administration pour les gestionnaires. Le panier intelligent s’inscrit donc dans une démarche d’innovation technologique au service d’un commerce plus fluide, automatisé et efficace.

# 

# Chapitre I :

# Présentation et cadrage du projet

**(Cahier de charges)**

1. **Présentation du sujet**

De nos jours, les nouvelles technologies sont largement utilisées dans la vie quotidienne, notamment dans les secteurs du commerce et de la grande distribution. Grâce à **l’internet des objets (Iot)**, il est possible d’améliorer considérablement l’expérience d’achat aux magasins.

Le sujet de notre projet suit cette idée. Il consiste à créer un panier intelligent capable d’automatiser la détection des produits et de calculer le total des achats sans passer par une caisse classique.

Ce projet fait appel à plusieurs domaines comme l’électronique (les capteurs et les lecteurs RFID), l’informatique (développement d’applications web et mobile, gestion de base de données).

1. **Contexte et définition du projet:**

Dans le prolongement de nos études en sciences mathématiques etInformatiques (**SMI),** nous sommes amenés à réaliser un Projet de Fin d'Études (**PFE**), Ce dernier représente une opportunité importante pour mettre en pratiqueles notions théoriques, nos compétences techniques et nos méthodes de travailacquises tout au long de notre formation universitaire.

Le domaine du **commerce intelligent et des technologies intégrées au quotidien (Smart Retail & solutions lot)** a l’intention de développer l’expérience d’achat répondant au besoin des clients de gagner du temps et de simplifier leursInteractions en magasin. Il aide aussi les commerçants à améliorer leursopérations , réduire les coûts et fidéliser leurs clients grâce à des solutionsinnovantes .

Notre travail intitulé “panier intelligent” vise à améliorer l'expérience des clients en leur permettant d'effectuer leurs achats d’une manière rapide , tout en automatisant le calcul des factures par Un panier intelligent constitué de plusieurs capteurs, unecaméra intelligente, lecteur de RFID et une tablette . Ce système élimine ainsi lebesoin de passer par la caisse, ce panier-là représente une solution efficace quioffre des services connectés, personnalisés et durables.

Les supermarchés aujourd'hui cherchent la satisfaction de leurs clients. Grâce au panier intelligent, les gens n’auront plus besoin d’attendre en file pour payer, car cepanier analyse et pèse les produits au fur et à mesure qu’ils sont déposés.

1. **Problématique** :

3.1 Problématique d’administration :

3.1.1 Problème de gestion manuelle des comptes clients :

    L’administration a souvent du mal à gérer les comptes des clients, car les opérations comme la création, la modification ou la suppression se font d’une manière manuelle, ce qui peut entraîner des erreurs, des oublis...

De plus sans interface bien conçue, la recherche ou la mise à jour des données devient lente et compliquée.

3.1.2 Difficulté à suivre l’historique des clients :

   Le manque d’outils pour consulter l’historique des achats empêche l’administrateur d’avoir une vision claire sur l’activité des clients. Cela rend difficile l’analyse des comportements d’achat, la détection des anomalies ou encore la personnalisation des services proposés.

3.2 Problématique de stock.

3.2.1 Gestion manuelle et complexe des produits :

Le système de gestion de stock présente souvent des difficultés lors de l’ajout, la modification ou la suppression de produits. Ces actions peuvent être lentes et provoquer des erreurs, comme des mauvais prix ou des quantités incorrectes, cela rend les informations moins fiables et rend le suivi des produits plus difficile.

3.2.2 Problèmes de rupture de stock :

L’un des problèmes majeurs est le manque d’alerte ou de suivi automatique lorsque certains produits deviennent indisponibles. Les ruptures de stock non détectées à temps peuvent entraîner des pertes de ventes et une insatisfaction chez les clients, qui ne trouvent pas les articles qu’ils cherchent.

3.2.3 Accumulation de produits invendus :

En l’absence de données précises sur les ventes, certains produits restent longtemps en stock sans être achetés. Cela entraîne une accumulation excessive en rayons ou en réserve, ou une augmentation des coûts de stockage.

3.3 Problématiques liées au client et à son expérience d’achat

3.3.1 Attente longue aux caisses

 Parmi les problèmes majeurs qui rencontrent les clients dans les magasins est le temps perdu en fil d’attente, surtout quand il y a beaucoup de monde, comme pendant les week-ends et les promotions. Cela rend l’expérience d’achat moins agréable et peut décourager certains clients à revenir.

3.3.2 Erreurs fréquentes lors de la facturation

 Lors du passage en caisse, des erreurs peuvent survenir, comme des produits mal scannés ou des prix incorrectes, ce genre de problème dérange les clients, car ils ont l’impression qu’on leur fait payer plus que prévu. Cela nuit à la confiance qu’ils ont envers le magasin et peut les pousser à faire des réclamations ou même à ne plus revenir.

3.3.3 Manque de visibilité pendant les achats

   Pendant ses courses, le client ne sait pas toujours combien il a déjà dépensé ni quels produits ont bien été pris en compte dans son panier. Comme il ne voit pas tout en temps réel, il peut avoir de mauvaises surprises au moment de payer, comme un total plus élevé que prévu, ou à une mauvaise gestion de son budget.

3.3.4 Moyens de paiement limités

Aujourd’hui, de nombreux clients aimeraient pouvoir payer de manière plus moderne, l’absence de choix rend l’achat moins pratique et peut poser un problème à ceux qui n’ont pas d’argents ou de carte bancaire.

3.3.5 Difficulté à obtenir une facture électronique

   Pendant l’opération de course, il est essentiel pour les clients de garder une trace claire de leurs achats. Cependant, dans les systèmes traditionnels les factures sont souvent imprimées ou remises manuellement, cette méthode peut compliquer l’accès à un historique numérique ce qui limite la possibilité de consulter les achats passés ou de gérer le budget.

1. **Objectifs** :

4.1 Objectifs liés à l’administration :

4.1.1 Gérer les comptes clients sans complexité.

  Le système permet à l’administration de faire toutes les opérations de gestion   possibles comme créer, modifier, consulter ou supprimer les comptes clients facilement depuis une interface simple et efficace. Cela évite les erreurs manuelles et les pertes de temps.

4.1.2 Assurer le suivi de l’historique des achats.

  Puisque les informations d’historiques sont importantes pour l’administration, il est essentiel d’intégrer un tableau de bord permettant à l’administration de visualiser les historiques d’achats de chaque client afin de faciliter l’analyse de comportements, et améliorer le service personnalisé.

4.2 Objectifs liés à la gestion du stock :

4.2.1 Simplifier la gestion des produits en stock.

  Le système offre au gestionnaire du stock une interface dédiée à l’ajout, la modification et la suppression des produits pour garantir un suivi rapide et plus fiable des informations sur les articles.

4.2.2 Recevoir des alertes en cas de stock faible.

    Le système signale automatiquement quand il ne reste plus de produit en stock. Cela permet de réagir rapidement pour éviter les ruptures et garder les rayons toujours remplis.

4.2.3 Réduire l’accumulation des produits invendus.

L’intégration des graphiques qui montrent les produits qui restent en grande quantité dans le stock permet de repérer rapidement les articles qui se vendent peu, et ainsi éviter l’accumulation inutile, optimiser les commandes, ou lancer des promotions ciblées pour écouler le surplus.

4.3 Objectifs liés à l’expérience d’achat :

4.3.1 Réduire le temps d’attente en caisse.

Grâce à la détection automatique des produits via capteurs et RFID, afficher le montant total sur la tablette du panier, et permettre au client de payer directement via une application mobile ou carte prépayée, sans passer par une caisse classique.

4.3.2 Éviter les erreurs de facturation.

Automatiser le calcul du montant total en temps réel et associer chaque produit à son prix exact, pour minimiser les erreurs humaines et garantir plus de fiabilité.

4.3.3 Offrir une visibilité claire pendant les achats.

Afficher en temps réel sur la tablette le contenu du panier et le total provisoire pour aider le client à suivre ses achats et gérer son budget plus facilement.

4.3.4 Diversifier les moyens de paiement.

  Offrir plusieurs options de paiement comme application mobile, cartes rechargeables ou identifiant client sur la tablette, pour rendre le processus plus accessible, rapide et sécurisé.

4.3.5 Fournir automatiquement une facture électronique.

  Générer une facture numérique à chaque achat et l’envoyer automatiquement dans l’historique du compte client, pour que ce dernier puisse la consulter à tout moment via son smartphone.

1. **Etude de l’existant**

Depuis plusieurs années, plusieurs entreprises à travers le monde ont développé des solutions de paniers intelligents intégrant des capteurs, des caméras et des systèmes de paiement automatisés.

Voici quelques exemples notables :

1. Amazon Dash Cart (États-Unis - 2020)

- Développé par Amazon, le Dash Cart est l’une des références les plus connues. Il utilise

une combinaison de capteurs de poids, caméras, intelligence artificielle et machine learning

pour détecter les articles ajoutés et retirés du panier .

- Le paiement se fait automatiquement via l’application Amazon Go, sans passage en caisse.

- Disponible dans certaines enseignes Amazon Fresh aux États-Unis.

1. Caper AI Smart Cart (États-Unis - 2019)

- Créé par Caper AI, ce panier intelligent est équipé de caméras, d’un lecteur RFID et d’un

écran tactile pour permettre aux clients de scanner eux-mêmes leurs articles.

- Il propose des recommandations en temps réel et une assistance numérique pour

améliorer l’expérience d’achat.- Acquis en 2021 par Instacart, un leader du e-commerce

alimentaire en Amérique du Nord.

1. Carrefour Flash 10/10 (France - 2021)

- Carrefour, en collaboration avec Ai-fi, a lancé un magasin sans caisse à Paris où les clients

peuvent entrer, remplir leur panier et partir sans passer par une

caisse physique.

- Le système repose sur une technologie de caméras intelligentes et d’intelligence artificielle,

Combinée à un panier détectant automatiquement les articles choisis .

4. JD.com & 7Fresh (Chine - 2018)

- En Chine, JD.com a lancé un panier intelligent dans ses magasins Fresh, utilisant des capteurs et un écran numérique pour afficher automatiquement le prix et les informations nutritionnelles des articles.

- Ce système est combiné avec un paiement mobile via WeChat Pay et Alipay.

5.Mishipay & Auchan Scan & Go (Europe - 2022).

- Des solutions comme Mis Hipay et le Scan & Go d’Auchan permettent aux clients d’utiliser leur propre smartphone pour scanner et payer leurs articles sans attendre en caisse.

- Bien que ce ne soit pas un panier intelligent au sens strict, ces solutions visent le même objectif : accélérer le passage en caisse et améliorer l’expérience client.

1. **Produit du projet (Solution Proposée)**

Le projet vise à développer un système de panier intelligent qui modernise l’expérience d’achat en magasin en combinant des technologies comme le RFID, l’IoT et des interfaces numériques (web et mobile). Le système permet de détecter automatiquement les produits, afficher le montant total en temps réel et payer sans passer par la caisse.

\_Composantes principales : Panier équipé d’un lecteur RFID et d’une tablette intégrée :

* **Microcontrôleur** (type Arduino) pour la communication entre les capteurs et le système.
* **Base de données centralisée** pour stocker les informations sur les produits, les clients et les achats.
* **Interfaces web** pour l’administrateur et le gestionnaire de stock.
* **Application mobile** pour le client (paiement, historique, factures).

Interfaces :

**Client** : voit les produits ajoutés, suit ses dépenses, paie depuis la tablette ou une application mobile.

**Administrateur** : gère les comptes clients, consulte l’historique, publie des annonces.

**Gestionnaire de stock** : gère les produits, surveille les niveaux de stock, visualise des statistiques

(ruptures, produits invendus, top ventes).

1. **Fonctionnalités du produit** :

Le panier intelligent dans ce projet intègre plusieurs fonctionnalités visant à améliorer et moderniser l’expérience d’achat et à optimiser la gestion en magasin. Voici les principales fonctionnalités :

* Détection automatique des produits :

Les articles sont automatiquement identifiés lorsqu’ils sont placés dans le panier grâce à un système de capteurs et à un lecteur RFID. Cela limite les oublis et automatise la saisie.

* Affichage en temps réel sur la tablette du panier :

Une tablette intégrée au panier affiche instantanément la liste des produits ajoutés , leur prix et le total.

* Paiement sans passage en caisse :

Le système permet un paiement autonome, sans fil d’attente, via une application mobile, une carte rechargeable ou à un compte personnel.

* Facturation numérique automatique :

Après le paiement, une facture électronique est générée automatiquement dans l’application mobile du client. Elle peut être consultée ou téléchargée.

* Interface d’administration :

L’administrateur dispose d’une interface dédiée qui lui permet de gérer les clients du système. À travers un tableau de bord clair et fonctionnel, il peut créer de nouveaux comptes clients, modifier ou supprimer les informations existantes, consulter l’historique des achats de chaque client ou envoyer des annonces au gestionnaire de stock.

* Interface de gestion de stock :

Une interface de gestion du stock offre les fonctionnalités comme l’ajout, la modification ou la suppression de produits, alertes sur les stocks faibles ou ruptures et visualisation de graphiques.

1. **Besoins fonctionnels**

Dans cette partie nous présentons les fonctionnalités de ce projet en mettant l’accent sur une description précise et claire. Au cœur de ce projet se trouvent quatre fonctions principales :

**8.1 Fonction principale** : Interagir avec la tablette du panier

8.1.1 Sous-fonctions :

- Consulter les articles ajoutés au panier.

- Consulter le solde.

- Consulter le montant total.

- Payer ( carte bancaire, QR code ou carte rechargeable).

**8.2 Fonction principale** : Gérer des comptes clients.

8.2.1 Sous-fonctions :

- Créer un nouveau client.

- Modifier les informations d’un compte client existant.

- Supprimer un compte client.

- Consulter les informations du clients ( leur historiques d’achat , nom ...).

**8.3 Fonction principale** : Gérer le paiement et la facturation.

8.3.1 Sous-fonctions :

- Payer via l’application mobile.

- Scanner un code QR pour payer ou se connecter.

- Générer une facture ( demander facture : pdf , CSV ).

**8.4 Fonction principale** : Gérer le stock.

8.4.1 Sous-fonctions :

- Ajouter un nouveau produit.

- Modifier les informations d’un produit existant.

- Consulter le stock.

- Générer un rapport.

- Enregistrer les réceptions de stock entrant.

1. **Besoins non fonctionnels**

Les besoins fonctionnels décrivent ce que le système doit faire, comme mesurer le poids des produits ou générer une facture. Les besoins non-fonctionnels définissent comment le système doit être, par exemple rapide, sécurisé, facile à utiliser et compatible avec les capteurs et le matériel utilisé.

* **Performance et rapidité d’exécution:**

Le système doit répondre rapidement aux actions des utilisateurs. Par exemple, lorsqu’un produit est détecté par le lecteur RFID ou qu’un panier est validé, le traitement doit être immédiat (temps de réponse inférieur à 1 seconde pour l’affichage).

* **Fiabilité et stabilité:**

Le système doit rester stable même en cas d’interruptions de communication avec les capteurs ou la base de données. Une reconnexion automatique ou une tolérance aux pannes est souhaitable, surtout dans les phases de lecture des produits ou de paiement.

* **Sécurité des données:**

Les mots de passe des utilisateurs doivent être stockés de manière sécurisée (ex : hachage). L’accès à l’administration et à la gestion de stock doit être protégé par des rôles. Les paiements doivent aussi être vérifiés, même si l’aspect sécurité du paiement RFID n’est pas prioritaire.

* **Accessibilité multiplate-forme:**

L’application mobile doit être utilisable sur la majorité des smartphones Android, tandis que l’interface tablette (dans le panier) doit être simple, intuitive, et bien adaptée au tactile.

* **Simplicité d’utilisation (ergonomie):**

Les interfaces (client, admin, gestion stock) doivent être claires, avec une navigation fluide et intuitive, même pour des utilisateurs non expérimentés. L’affichage des produits, du total, ou de l’historique doit être immédiat et lisible.

* **Maintenabilité:**

Le système est développé en utilisant des frameworks modernes comme Laravel et des bonnes pratiques (architecture MVC) pour faciliter les futures mises à jour, corrections ou ajouts de nouvelles fonctionnalités.

* **Évolutivité** **(scalabilité**):

Le système doit être capable d’ajouter facilement de nouveaux produits, nouveaux clients ou même nouveaux paniers sans ralentissement. L’architecture permet également d’ajouter de nouveaux modules (ex : notifications, promotions, statistiques) à l’avenir.

* **Compatibilité avec le matériel**

Le système doit fonctionner correctement avec les capteurs RFID, les caméras, les capteurs de poids et les tablettes embarquées. L’intégration matérielle doit être testée pour éviter les conflits ou les déconnexions.

* **Sauvegarde des données Les informations critiques:**

(Utilisateurs, ventes, paiements) doivent être sauvegardées régulièrement pour éviter toute perte de données.

* **Affichage en temps réel:**

Les données affichées à l’écran (montant total, articles scannés) doivent être mises à jour en temps réel pour offrir une expérience fluide et sans confusion au client.

# 

# Chapitre II :

# Déroulement du projet

1. **Planification opérationnelle** 
   1. Tableau des tâches

Le tableau des tâches permet de présenter les tâches du projet ainsi que le temps prévu pour chaque étape. Ce tableau nous aide à bien organiser le projet, à suivre l’avancement et à respecter les délais fixés.

Cahier de charge :

Le cahier de charge est le document fondateur du projet, définissant les besoins précis du panier intelligent, de l'application mobile et de l'interface d'administration, assurant ainsi une base solide pour le développement.

Conception:

La phase de conception transforme les exigences du cahier des charges en plans concrets, détaillant l'architecture logicielle, l'interface utilisateur et l'intégration des matériels, garantissant ainsi la cohérence et la performance du système.

Installation de l’environnement:

L'installation de l'environnement met en place les outils et infrastructures nécessaires au développement, assurant un cadre de travail stable et performant pour l'équipe.

Réalisation du Front-End:

Le développement front-end crée l'interface utilisateur du panier intelligent et de l'application mobile offrant une expérience intuitive et conviviale aux utilisateurs.

Réalisation du Back-End:

La partie du back-end sera conçue pour être performant , sécurisé et facile à maintenir ainsi ce développement est le cœur du système, assurant la gestion de données , la communication

entre les différents composants.

Intégration du Base de données:

Cette étape permet de connecter le code du projet à la base de données. Grâce à cela, les interfaces peuvent afficher, ajouter, modifier ou supprimer des données (comme les produits ou les utilisateurs) en temps réel.

Intégration du matériels:

L 'intégration des matériels connecte les composants physiques (RFID, caméra, tablette) au système logiciel, garantissant le bon fonctionnement du panier intelligent

|  |  |  |
| --- | --- | --- |
| Étape | Date de début | Date de fin |
| Cahier de charge | 20/01/2025 | 18/02/2025 |
| Installation de l’environnement | 19/02/2025 | 21/02/2025 |
| Conception | 22/02/2025 | 08/03/2025 |
| Développement Front-End | 09/03/2025 | 22/04/2025 |
| Développement Back-End | 23/04/2025 | 22/05/2025 |
| Intégration du bases de données | 23/05/2025 | 28/05/2025 |
| Intégration du matériels | 29/05/2025 | 30/05/2025 |
| Phase de test | 31/05/2025 | 03/06/2025 |
| Rapport | 04/06/2025 | 11/06/2025 |
| Présentation | 12/06/2025 | 16/06/2025 |

*Tableau1:tableau des tâches*

Phase de test:

Les tests ont été faits pour s’assurer que tout fonctionne bien. On a testé chaque partie connexion, affichage des produits, mise à jour du stock, lecture RFID…

Cela a permis de corriger les erreurs avant de terminer le projet.

Rapport:

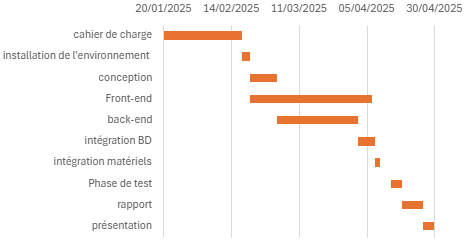
Le rapport rassemble l'avancement , les choix techniques et les défis du projet, permettant un suivi rigoureux et la mémorisation des connaissances.

Présentation:

La présentation du système du panier intelligent.

* 1. Diagramme de GANT

Le diagramme de GANT est un outil de gestion de projet qui permet de visualiser les tâches à réaliser dans le temps.



*Figure1:diagramme de gantt*

"A partir du graphe de PERT, on peut dresser le diagramme de Gant qui permet d’établir un calendrier de travail en considérant les ressources et leurs disponibilités[1]". Il aide à organiser le travaille et avoir une bonne gestion du temps.

1. **Processus de développement** 
   1. Présentation du cycle de vie de développement de l’application

Le cycle de vie de vie de développement logiciel représente l’ensemble des étapes que

traverse un projet depuis le début jusqu’à la fin.

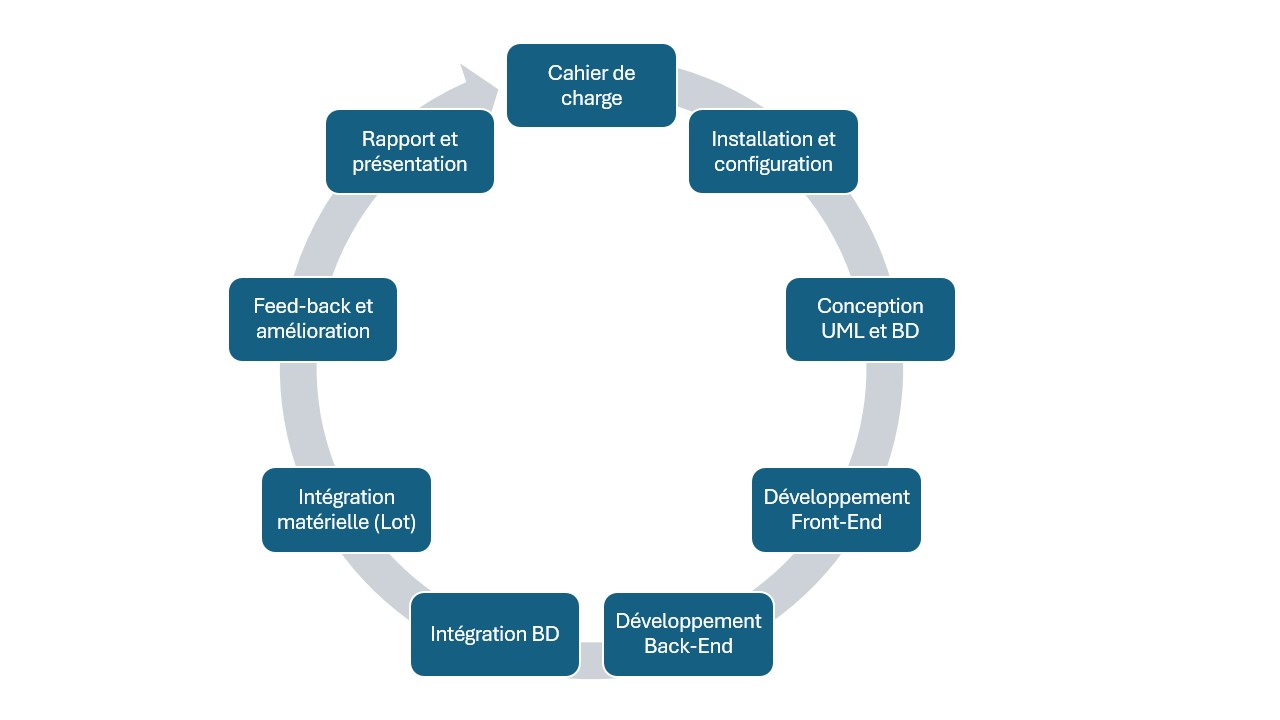
"Les étapes de développement logiciel ne sont pas forcément utilisées de façon linéaire. On parle souvent de cycles de vie, qui ont pour but d’organiser ces étapes de différentes manières en fonction d’un certain nombre de critères relatifs au projet de développement[2]".

Dans notre projet «panier intelligent» , nous avons adapté une méthode de travail Agile basée sur le principe des sprints. Chaque sprint représente une courte période de développement pendant une ou plusieurs fonctionnalités précises sont planifiées, développées, testées, revues puis intégrées .

Nous avons choisis la méthode Agile à cause de la nature de ce projet, car le système de panier intelligent contient plusieurs modules indépendants mais liés ( interface administrateur, interface gestionnaire, application mobile et interface tablette).

Donc la méthode Agile permet de travailler sur chaque module séparément, tout en assurant leur intégration progressive.

Notre projet de panier intelligent a suivi une démarche Agile itérative. Nous avons d'abord défini les besoins et mis en place l'environnement. La conception (UML/BDD) a précédé le développement des interfaces (web/mobile/tablette) et de la logique back-end.

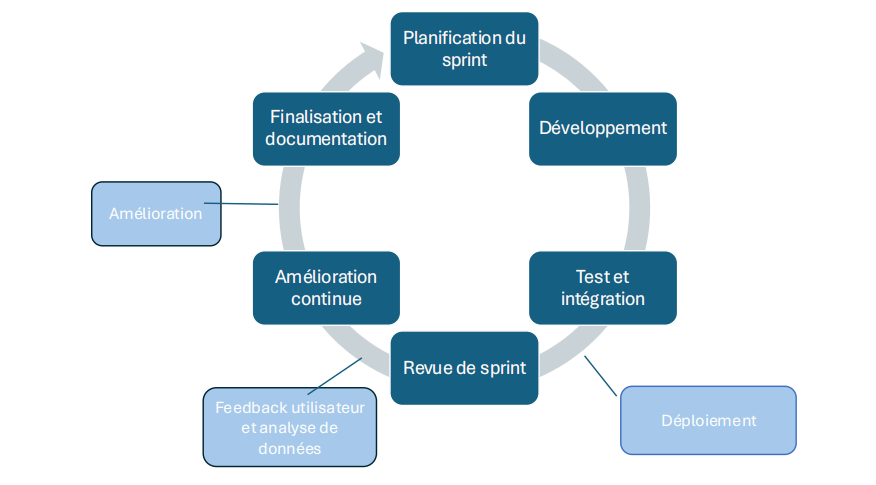


*Figure2:cycle de vie Agile globale du projet*

L'intégration de la base de données et l'intégration matérielle (RFID/Arduino) ont été des étapes clés. Le projet a progressé grâce à des tests, du feedback et une amélioration continue, pour aboutir au rapport final et à la soutenance. Cette approche Agile a permis une progression flexible et adaptée à la complexité de notre solution.

Sprint:

Un sprint est une période de temps souvent 1 à 3 semaines durant laquelle les membres de l’équipe travaillent sur des fonctionnalités à livrer. Chaque sprint se termine par une version fonctionnelle et testable du produit, même si elle n’est pas complète. L’idée c’est d’avancer petit à petit, avec des cycles courtes de développement, test et amélioration .



*Figure3:cycle de vie Agile d’une seule sprint.*

Exemple: « développement du page de gestionnaire de stock durée(3jrs):

1. Planification : Définir les tâches à faire comme créer une page php et ajouter Boostrap...
2. Développement :Créer le fichier dashboard.php et développer le front-end, placer les bottons, les couleurs ...
3. Test et intégration : Vérifier que les bottons fonctionnent , ajouter un message d’erreur si l’utilisateur n’est pas connecté ...
4. Revue de sprint : valider avec l’équipe ou l’encadrant.
5. Amélioration : Ajouter des icônes , modifier les couleurs ...
6. Documentation : Nous notons à la fin du sprint les tâches terminés dans le rapport.
   1. Phases

Ce tableau ci\_dessous présente les différentes phases du projet ainsi que les livrables attendus à la fin de chaque phase. Chaque phase correspond à une étape clé du projet avec des objectifs spécifiques à atteindre. Les livrables décrivent les résultats concrets produits .

|  |  |  |
| --- | --- | --- |
| Phase | Description | Livrables |
| Planification du sprint | définition des tâches ,  choix de technologies et  planification des  fonctionnalités | cahier de charge |
| Développement | implémentation des  fonctionnalités et  développement du  logiciel | application mobile , application web panier,  applications web  d’administrateur et du gestionnaire de stock. |
| Test et intégration | tests unitaires et  intégration sur le logiciel  et le matériel ( lecteur  RFID , affichage sur  écran etc ..) | scénarios et  résultats des tests .  prototype du panier  fonctionnel avec  tous les composants  intégrés |
| Revue du sprint | Présentation des  fonctionnalités  développées . | conception |
| Amélioration continue | Optimisation des  performances ,  corrections , et  ajustements finaux | amélioration du  code et de  l’interface .  - ajustements  matériels. |
| Finalisation/documentation | Rédaction de rapport final  et les livrables finaux | Rapport |

*Tableau2:tableau des phases*

qui valident la réalisation des travaux prévus. Ce tableau facilite la planification, le suivi de l’avancement et l’évaluation de la qualité du projet.

# Chapitre III :

# Spécification et modélisation

1. **Les acteurs de l’application**

• **Client :**

C’est l’utilisateur final du système de panier intelligent. Il effectue ses achats en

utilisant le panier connecté, il peut consulter l’historique de ces achats, payer par

différentes manières (Carte ATM, Carte rechargeable, Code QR).

• **Administrateur :**

C’est le responsable global du système de paniers intelligents, il configure, surveille et

maintient le système, parmi ses actions possibles : créer, modifier et supprimer les

comptes clients et consulter ses informations comme l’historique.

• **Gestionnaire de stock :**

Le gestionnaire de stock, est responsable de la surveillance et de la gestion des produits en stock. Il peut consulter les niveaux de stock, filtrer les produits, visualiser les achats et les commandes, et ajouter de nouveaux produits. Son rôle est d'assurer la disponibilité des produits tout en optimisant les niveaux de stock.

1. **Langage de modélisation UML**

**UML** ( *Unified Modeling Language ),* est un langage de modélisation graphique utilisé en génie logiciel afin de visualiser, spécifier et construire les documents nécessaires au bon développement d’un système logiciel.

"UML a été proposé afin de standardiser les outils de développement: modèles, notations, diagrammes, ...[3]".



*Figure4:UML logo*

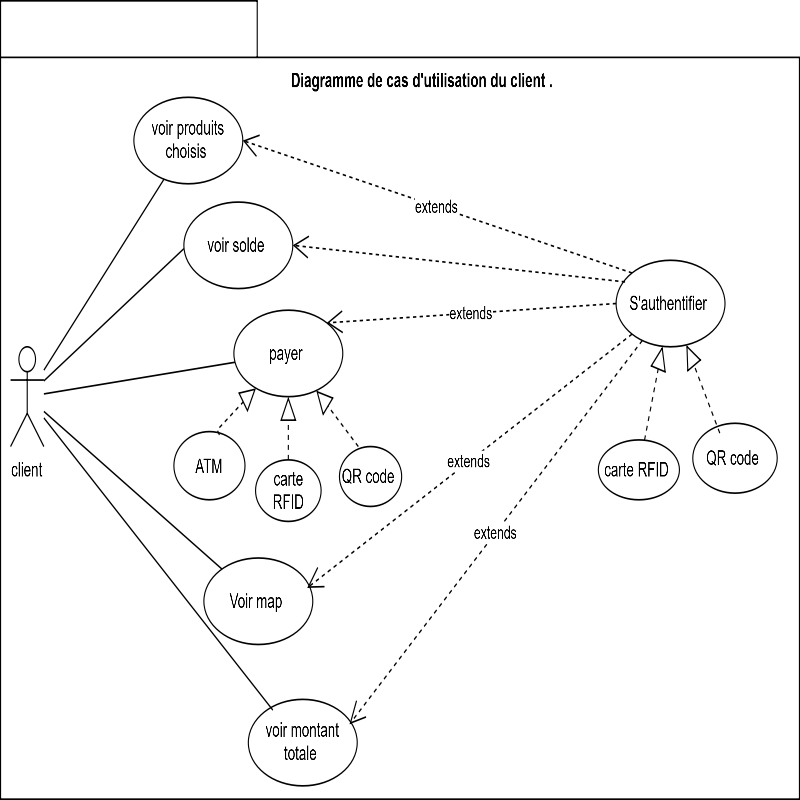
Pour cela nous choisirons le langage de modélisation UML pour mieux présenter l’architecture de nos applications, assurer une communication claire entre les différents membres de l’équipe et pour atteindre toutes les exigences spécifiques de notre projet.

1. **Diagramme de cas d’utilisation**

Le diagramme de cas d'utilisation permet de représenter les différentes fonctionnalités offertes par le système du point de vue de l'utilisateur. Il met en évidence les acteurs (utilisateurs ou systèmes externes) et les cas d'utilisation (fonctions ou services) qu’ils peuvent effectuer. Ce diagramme est essentiel pour définir les besoins fonctionnels du système et faciliter la communication entre les parties prenantes.

3.1 Diagramme de cas d’utilisation du client dans son application web du tablette :

Dans l’application web du tablette, le client peut interagir avec plusieurs fonctionnalités clés. Il peut voir les produits choisis et le montant total de sa sélection. Pour consulter son solde ou payer, le système peut exiger une authentification, réalisable via une carte RFID ou un QR code. Le paiement en lui-même peut s'effectuer par ATM, carte RFID, ou QR code.

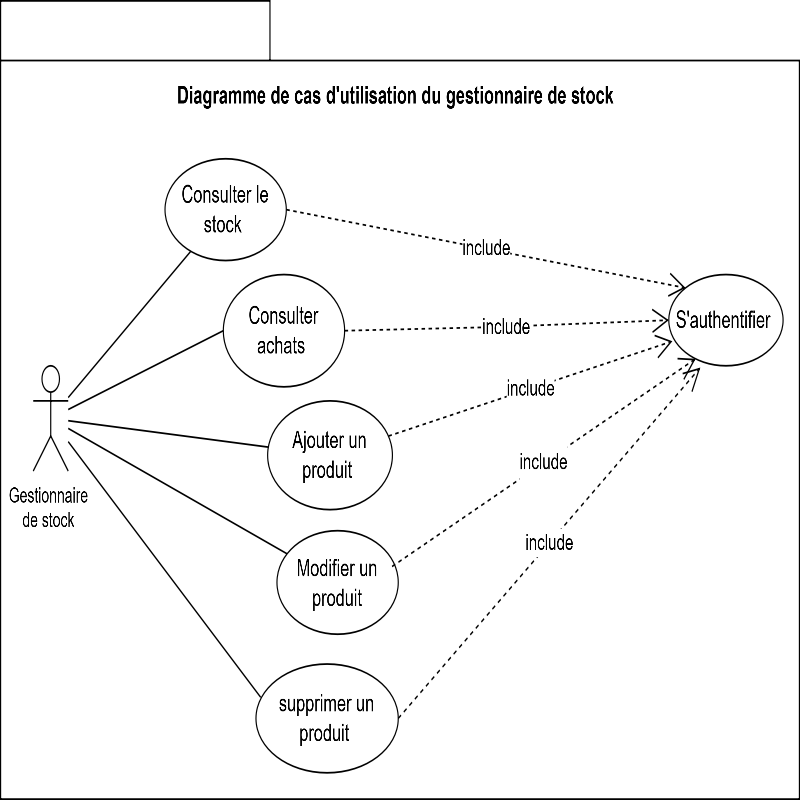


*Figure5:diagramme de cas d’utilisation du client dans sa tablette du panier .*

Enfin, le client a la possibilité de **voir une map**, une fonctionnalité qui pourrait également nécessiter une authentification. L'authentification est donc un processus central qui peut être étendu par plusieurs actions du client au sein de l'application.

3.2 Diagramme de cas d’utilisation du gestionnaire de stock.

Ce diagramme montre les différentes actions qu’un gestionnaire de stock peut réaliser dans le système du panier intelligent. Comme pour l’administrateur, toutes les fonctionnalités nécessitent une authentification préalable.



*Figure6:diagramme de cas d’utilisation du gestionnaire de stock .*

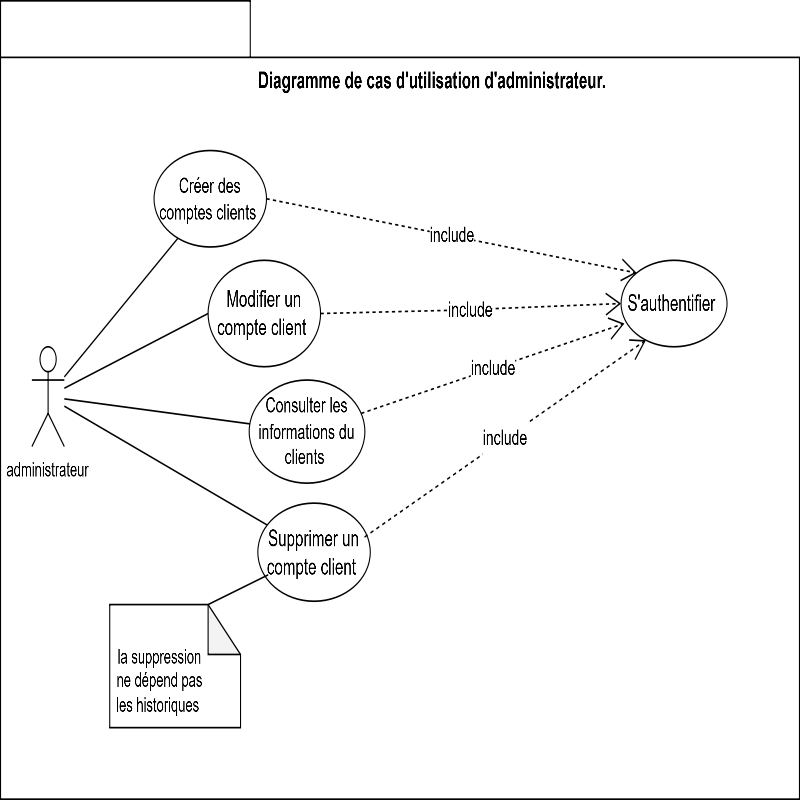
Une fois connecté, le gestionnaire de stock peut effectuer les actions suivantes :

* Consulter le stock : voir les produits disponibles dans l’inventaire.
* Consulter les achats : accéder à l’historique ou au résumé des achats effectués par les clients.
* Ajouter un produit : enregistrer un nouveau produit dans la base de données.
* Modifier un produit : changer les informations d’un produit existant (nom, quantité, prix, etc.).
* Supprimer un produit : retirer un produit du stock lorsqu’il n’est plus disponible ou plus utilisé.

3.3 diagramme de cas d’utilisation d’administrateur.

Ce diagramme représente les différentes actions qu'un administrateur peut effectuer dans le système de gestion du panier intelligent. Il montre les fonctionnalités accessibles à l’administrateur. Une fois authentifié, l’administrateur peut :

* Créer des comptes clients : ajouter de nouveaux utilisateurs dans le système.
* Modifier un compte client : changer les informations d’un client déjà enregistré.
* Consulter les informations des clients : voir les détails des clients, comme leurs coordonnées ou leur historique d’utilisation.
* Supprimer un compte client : effacer un compte utilisateur définitivement.

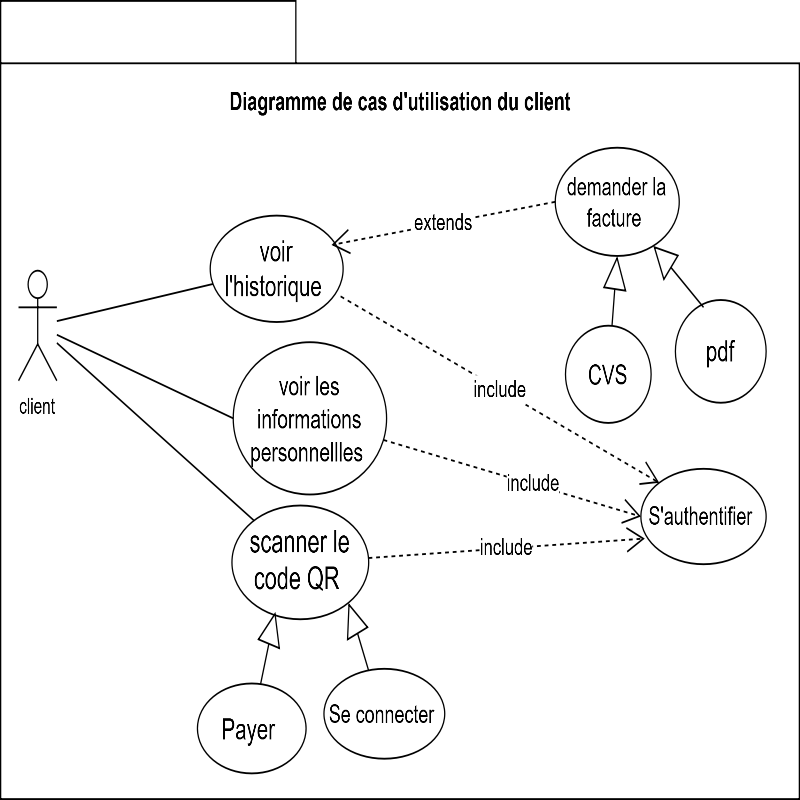


*Figure7:diagramme de cas d’utilisation d’administrateur.*

Il est aussi précisé dans le diagramme que **la suppression d’un compte client ne dépend pas des historiques**, car les historiques sont des données importants pour suivre les activités , analyser les performances et rédiger des rapports administratifs .

3.4 diagramme de cas d’utilisation du client dans son application mobile.

Dans l'application mobile, le client peut voir l'historique de ses actions, avec la possibilité d'étendre cette fonctionnalité pour demander une facture, qui peut être générée aux formats pdf ou csv. Il peut également voir ses informations personnelles, ce qui nécessite de s'authentifier.



*Figure8:diagramme de cas d’utilisation de client*

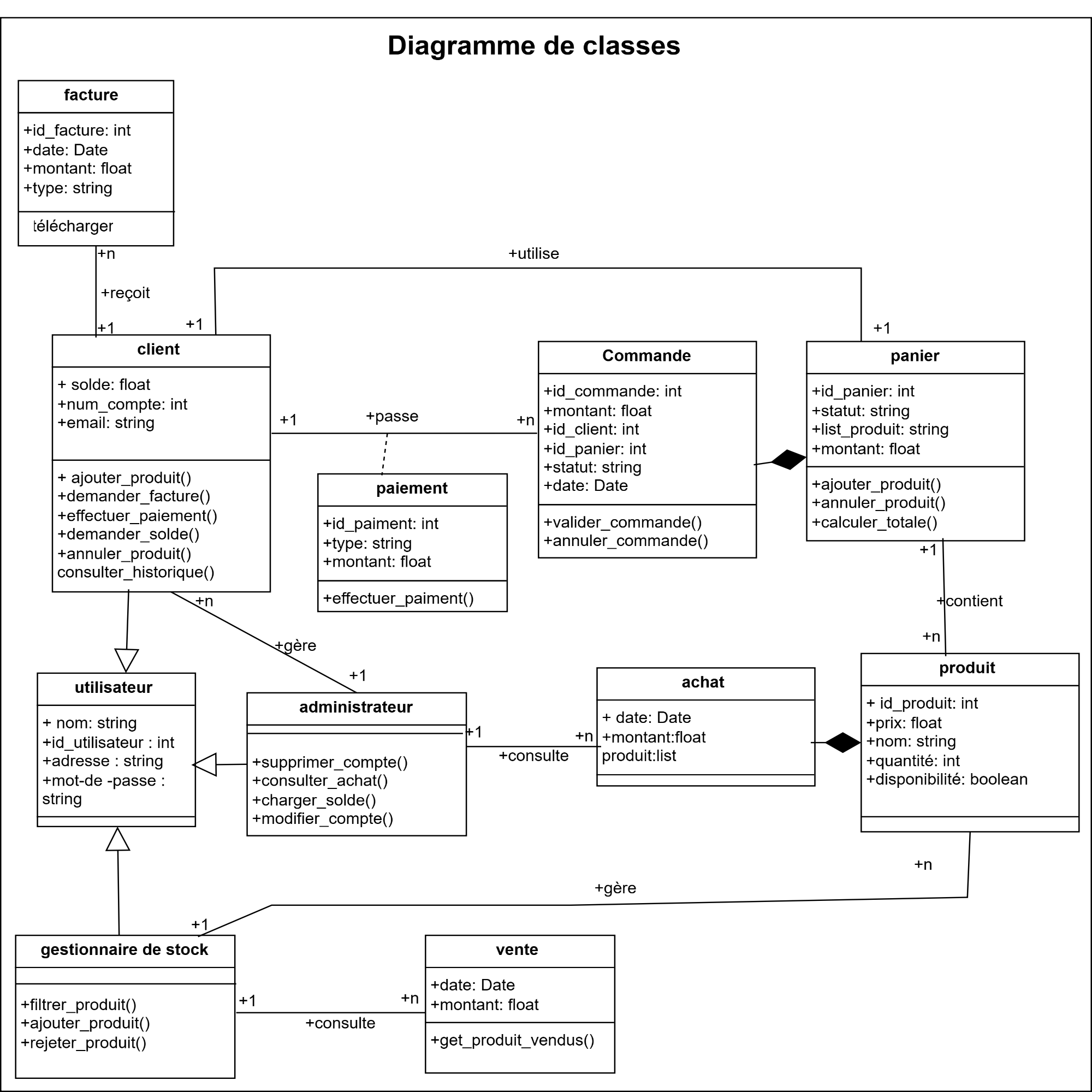
De même, l'action de scanner un QR code, utilisée pour payer ou pour se connecter, requiert également une authentification. Ainsi, l'accès aux informations personnelles et l'utilisation des fonctionnalités de paiement et de connexion via QR code sont sécurisés par un processus d'authentification.

1. **Diagramme de classes**

**Le diagramme de classes** décrit la structure statique du système. Il présente les classes avec leurs attributs et méthodes, ainsi que les relations (association, héritage, composition...) entre elles. Ce diagramme constitue une base essentielle pour le développement orienté objet, car il permet de modéliser les entités principales du système et leurs interactions structurelles.

4.1 Diagramme de classes du système du panier intelligent.

Ce diagramme montre comment fonctionne un système de panier intelligent. Il y a plusieurs types d'utilisateurs. Le client peut ajouter des produits dans son panier, passer une commande, payer, recevoir une facture et consulter son historique d’achats. Le panier contient une liste de produits et calcule le montant total. Chaque produit a un prix, une quantité et une disponibilité. Lorsqu’un client passe une commande, celle-ci est liée à son panier et à lui-même, avec un statut et une date. Une fois la commande passée, il peut faire un paiement qui sera enregistré. Ensuite, une facture est générée.

****

*Figure9:Diagramme de classes du système du panier intelligent.*

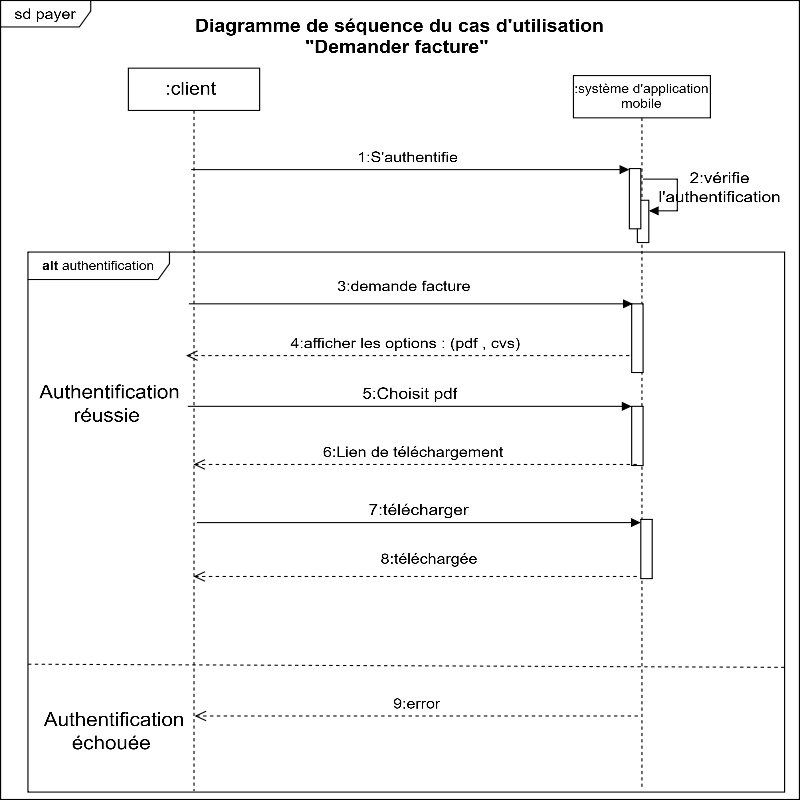
Il existe aussi un administrateur qui gère les comptes clients, peut consulter les achats, recharger le solde ou modifier les informations d’un utilisateur. Le gestionnaire de stock, lui, s’occupe des produits : il peut les ajouter, les supprimer ou les filtrer. Il y a aussi des ventes et des achats qui sont enregistrés avec leurs dates et montants. Tout est bien organisé pour que le système fonctionne de manière fluide et simple.

1. **Diagramme de séquences**

Le diagramme de séquence illustre l'enchaînement des interactions entre les différents objets ou composants du système, selon un scénario particulier. Il montre le déroulement temporel des messages échangés entre les entités pour accomplir une tâche spécifique. Ce type de diagramme est utile pour analyser la logique du traitement et vérifier la cohérence des échanges.

5.1 Diagramme de séquence du cas d’utilisation «demander facture».

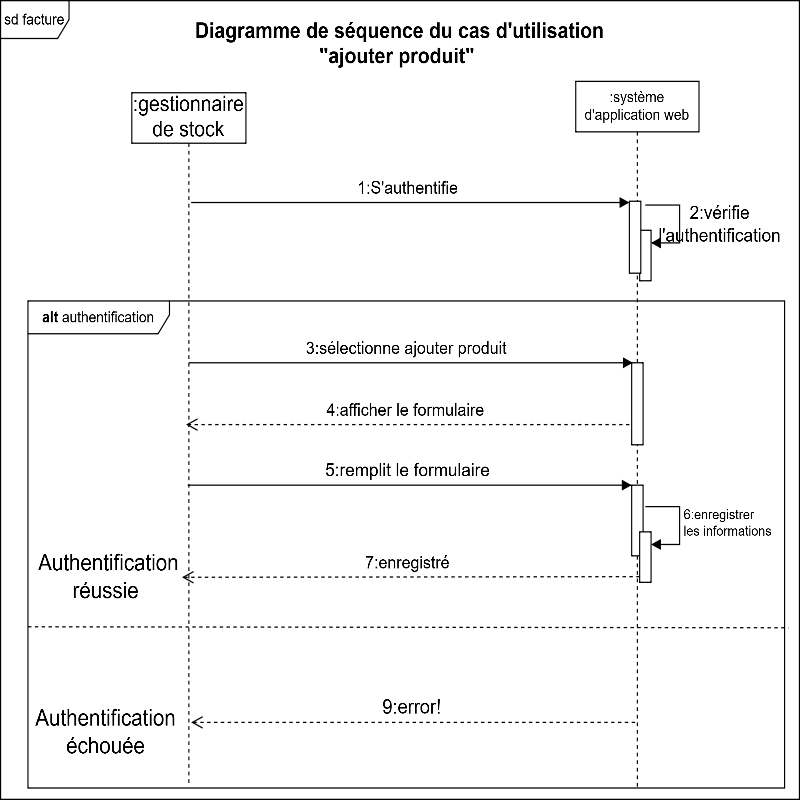
Pour télécharger une facture via l'application mobile, le client doit d'abord s'authentifier auprès du système. Une fois l'authentification réussie, il peut demander la facture, et le système lui propose des options de format, comme PDF ou CSV.



*Figure10:diagramme de séquence du cas d’utilisation «demander facture».*

* 1. Diagramme de séquence du cas d’utilisation «ajouter produit».

Pour ajouter un produit via son application web, le gestionnaire de stock doit d'abord s'authentifier auprès du système. Une fois l'authentification réussie, il sélectionne l'option "ajouter produit",

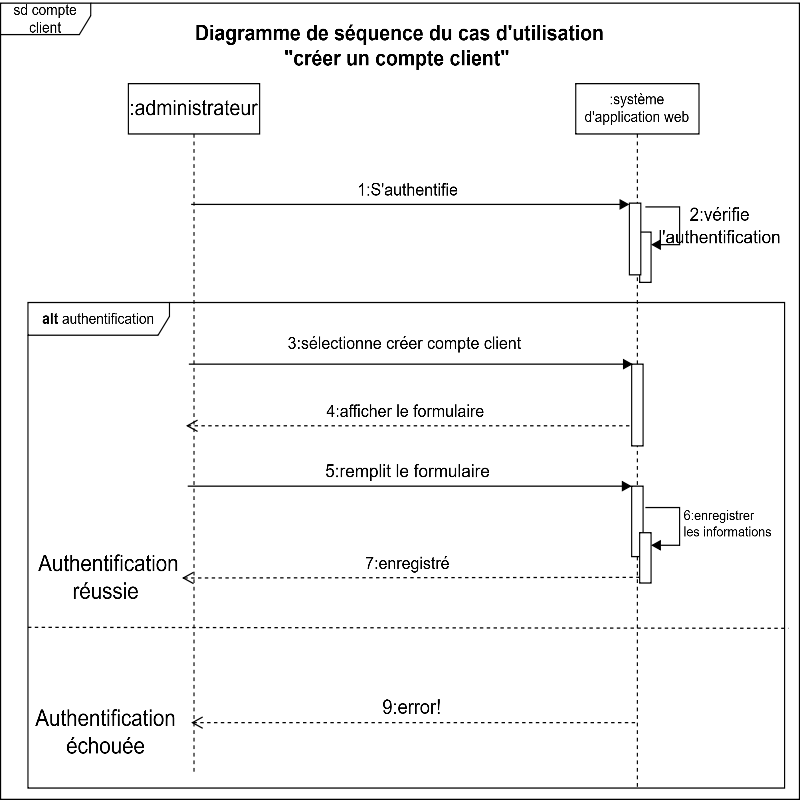


*Figure10:diagramme de séquence du cas d’utilisation «ajouter produit».*

ce qui lui présente un formulaire à remplir. Après avoir renseigné les informations nécessaires, le système enregistre ces données et confirme au gestionnaire que l'ajout a réussi. En revanche, si l'authentification échoue, le système affiche un message d'erreur, empêchant l'ajout du produit.

* 1. Diagramme de séquence du cas d’utilisation «créer un compte client».

Ce diagramme de séquence illustre le processus de création d’un compte client par un administrateur à travers une application web. L’administrateur commence par s’authentifier auprès du système.

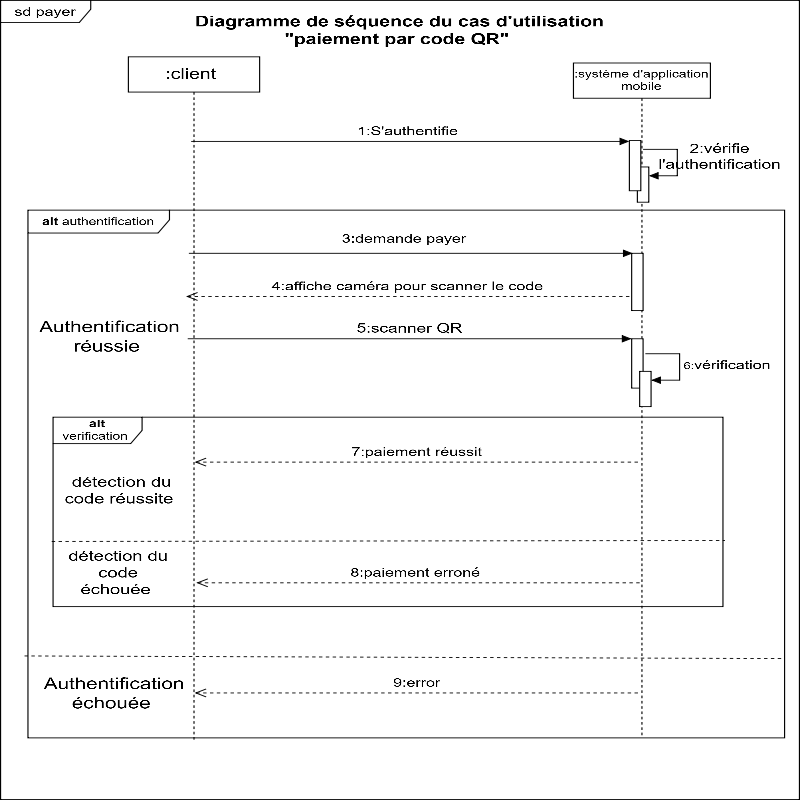


*Figure12:Diagramme de séquence du cas d’utilisation « ajouter un compte client».*

Si l’authentification est valide, il accède à l’option de création de compte client. Le système lui présente alors un formulaire qu’il remplit avec les informations nécessaires. Une fois les données saisies, le système les enregistre et confirme la création du compte. En revanche, si l’authentification échoue, le système affiche un message d’erreur, interrompant le processus.

* 1. Diagramme de séquence de cas d’utilisation «paiement par code QR».

Ce diagramme de séquence représente le scénario de paiement par code QR dans une application mobile. Le client commence par s’authentifier, et le système vérifie son identité. En cas de succès, il peut initier une demande de paiement.

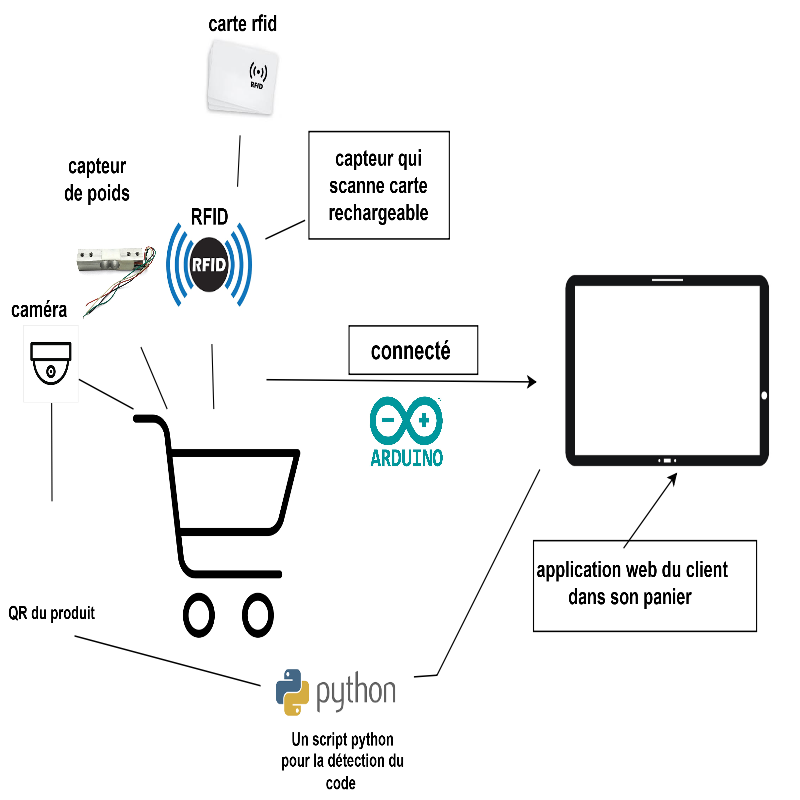


*Figure13:diagramme de séquence du cas d’utilisation « paiement par code QR».*

Le système ouvre alors la caméra pour permettre au client de scanner le code QR. Une fois le code scanné, le système effectue une vérification. Si le code est correct, le paiement est validé. Si le code est erroné, un message d’échec de paiement est affiché. En cas d’échec d’authentification dès le début, le système affiche une erreur.

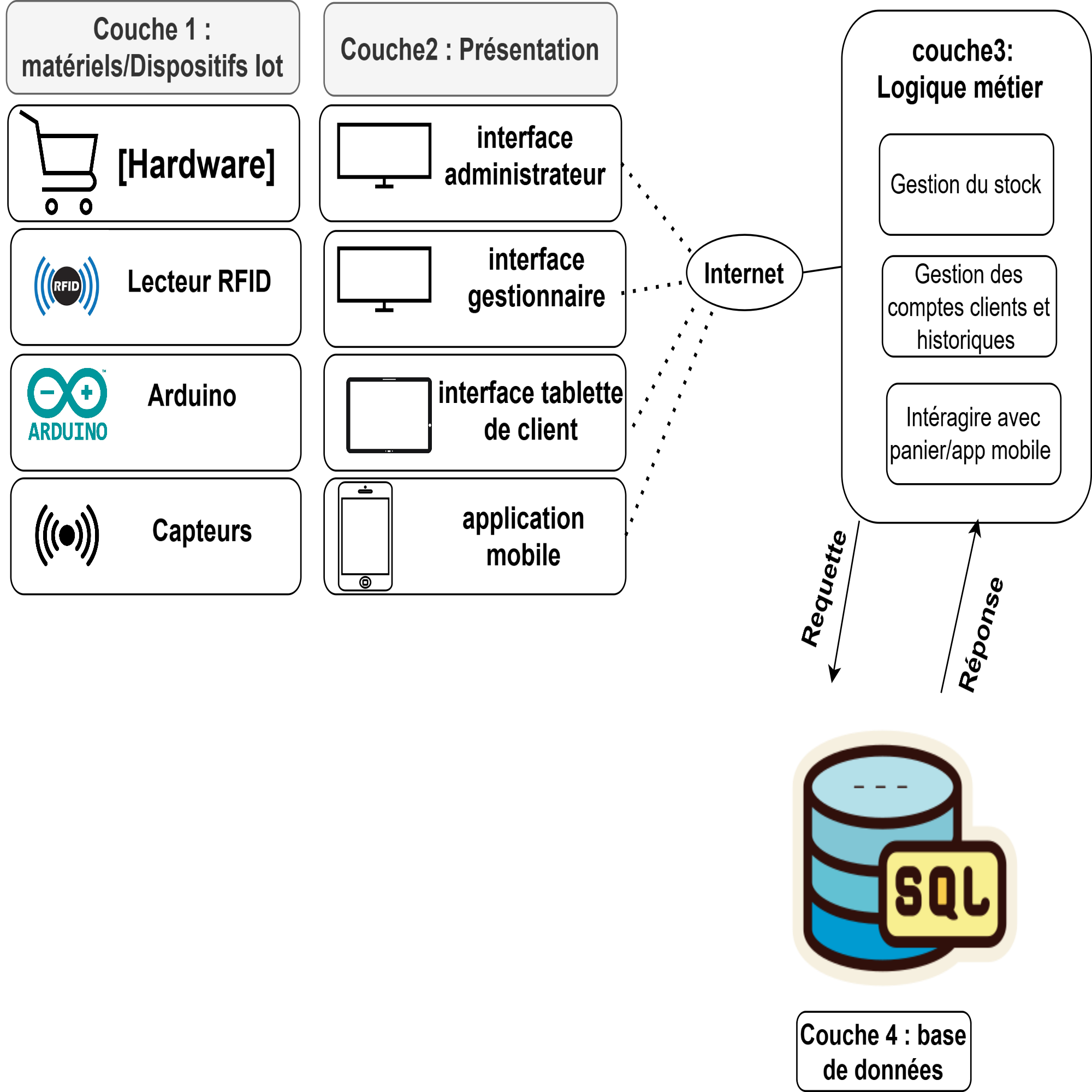
1. **Conception**
   1. Architecture globale de l’application

Une architecture distribuée peut intégrer à la fois des éléments logiciels et **des composants matériels connectés**, comme dans le cas de notre projet . " **L’architecture distribuée est aussi appelée architecture multi-tiers ou architecture N-tier[4]**", car elle est organisée en plusieurs couches indépendantes qui communiquent entre elles.



*Figure14:Architecture matérielle*

Dans notre système de panier intelligent, les capteurs, le lecteur RFID et la carte Arduino représentent la **partie matérielle** de l’architecture. Ces dispositifs interagissent avec la tablette (interface de présentation), envoient des données à la couche applicative (back-end), qui à son tour communique avec une base de données distante.



*Figure15: Architecture logicielle*

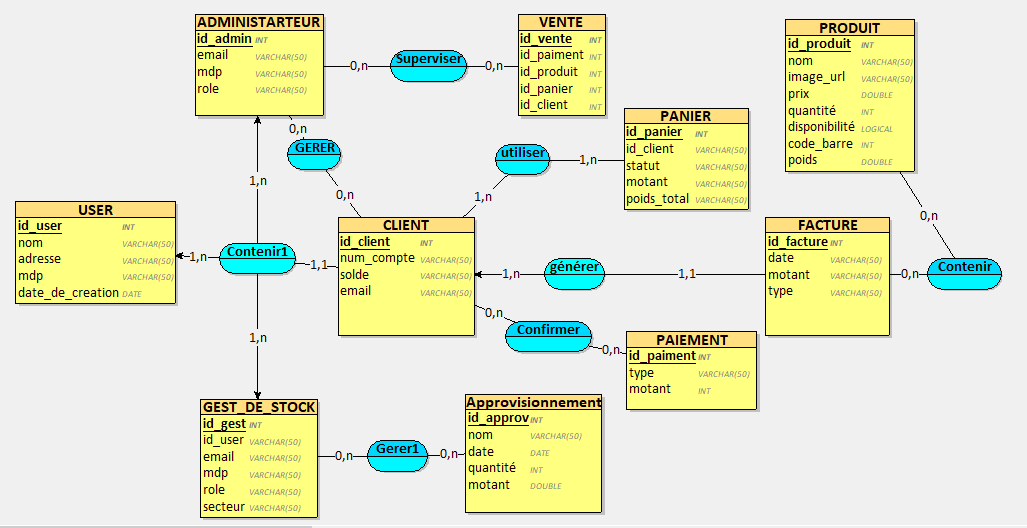
La deuxième couche est la **couche de présentation**, représentée par la tablette du panier et les interfaces web ou mobiles utilisées par les clients et les administrateurs.

Ensuite, la **troisième couche**, appelée **couche applicative**, traite les données reçues, applique les règles de gestion (comme le calcul des prix ou la mise à jour du stock) et fait le lien entre les interfaces et les bases de données. Enfin, la **quatrième couche**, la **couche de données**, est responsable du stockage des informations dans une base de données distante.

* 1. Conception de la base de données
* La conception de notre panier intelligent est conçue à partir d’un modèle conceptuel de données (MCD) qui organise les rôles et les fonctionnalités des utilisateurs. A la base, nous avons une entité principale nommée «utilisateur», qui regroupe les informations essentielles. Chaque utilisateur peut être un administrateur, client ou gestionnaire de stock.

Le client utilise un panier pour faire ses achats, le panier contient plusieurs produits , a un statut (en cours, validé…) et un poids total. Une fois les achats terminés, une vente est enregistrée, une facture est générée automatiquement, et un paiement est effectué.

Les produits ont un nom, un prix, une quantité, un poids, une disponibilité et un code-barres.La facture indique le montant total, la date, et le type de facture. Le paiement garde les informations sur le montant payé et le mode de paiement (par carte, solde...).

**

*Figure16:MCD du système*

D’un autre côté, le gestionnaire de stock s’occupe de l’approvisionnement. À chaque ajout de produits, une fiche de réapprovisionnement est créée avec la date, la quantité ajoutée, et le produit concerné. Cela permet de suivre les mouvements du stock.

Enfin, l’administrateur supervise tout le système, notamment les ventes et les activités des utilisateurs.

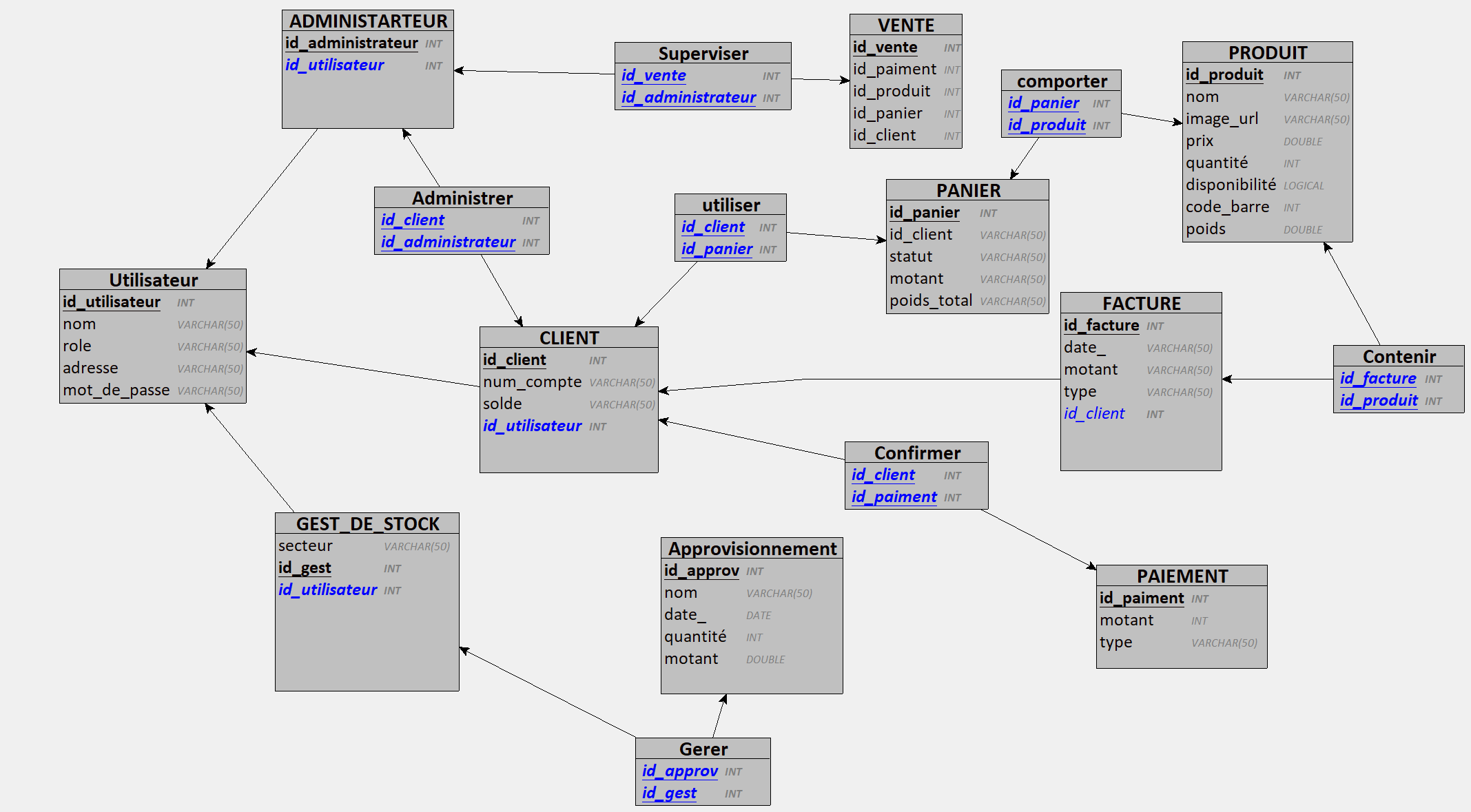
* Le Modèle Logique de Données (MLD) de notre projet transforme les éléments du MCD en tables prêtes à être utilisées dans une base de données.

Chaque entité devient une table avec ses informations (attributs), et chaque lien entre entités est représenté par une clé étrangère pour relier les tables entre elles.

La table principale est Utilisateur, qui contient les données de base comme le nom, le rôle, l’adresse et le mot de passe. Cette table est connectée à trois autres tables : Administrateur, Client, et Gestionnaire de stock, qui utilisent toutes l’Id\_utilisateur comme clé étrangère pour savoir à quel utilisateur elles sont liées.

Le Client peut utiliser un panier, faire un paiement, et avoir une facture. Le Panier contient le statut, le montant total, et le poids total.Les produits dans le panier sont gérés via une table intermédiaire appelée “comporter”, car un panier peut contenir plusieurs produits, et un produit peut être dans plusieurs paniers.Les ventes sont enregistrées dans une table Vente, qui relie les clients, produits, paniers, et paiements.L’administrateur supervise les ventes via une table appelée Superviser.

Pour la gestion du stock :Le gestionnaire de stock utilise la table Approvisionnement pour ajouter des produits.Cette table garde la date, la quantité, et le produit concerné.



*Figure17:MLD du système*

Enfin :La Facture contient les détails de la commande (montant, date, type). Elle est liée à un paiement spécifique, et à travers la table Contenir, elle est reliée aux produits concernés.Ce MLD organise toutes les données du système de façon claire, logique et cohérente, en permettant de suivre chaque étape : de l’achat à la gestion du stock.

# Chapitre VI :

# Implémentation et réalisation

1. **Outils de développement**

(Présentation des outils, choix du Framework, justification des choix)

* 1. Langage de développement
* Définition de PHP

**PHP** ( Hypertext Preprocessor): est un langage de script généraliste et Open Source , spécialement conçue pour le développement des applications web dynamiques.

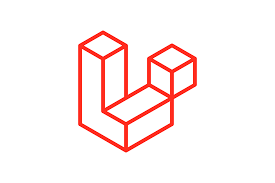


*Figure18:php logo*

Il a été crée par *Rasmus Ledrof en 1993 ,* ce langage est choisi pour le développement du Back-End de notre projet en raison de sa puissance dans la gestion des bases de données telles que MySQL , et de sa capacité de développer des applications web interactives.

* **L’utilisation du framework «Laravel» :**

Laravel est un exellent choix pour notre projet car il facilite la gestion des données sans avoir besoin d’écrire des requêtes SQL compliquées. Il utilise une architecture appelée MVC (Modèle-Vue-Contrôleur), ce qui permet d’organiser le code de manière claire et facile à maintenir.



*Figure19:laravel logo*

De plus, Laravel offre la possibilité de faire des tests automatiques (tests unitaires et fonctionnels), ce qui aide à vérifier que les fonctionnalités marchent bien et restent stables même quand on ajoute de nouvelles options.

* Définition du langage HTML

**HTML** (HyperText Markup Language): est le langage standard utilisé pour structurer le contenue des pages web . Il a été crée par *Tim Bernes-Lee en 1991 ,* Ce n’est pas un langage de programmation, mais un langage de structure. Il définit les éléments de la page (titres, paragraphes, tableaux, images, liens, boutons...) à l’aide de balises.



*Figure20: HTML logo*

HTML est essentiel dans notre projet car il permet de transformer les données en une interface compréhensible par l'utilisateur.

* Définition de CSS

**CSS** ( Cascading Style Sheets ) : est un langage utilisé pour mettre en forme les pages web crées avec HTML ou XML .



*Figure21: CSS logo*

CSS décrit la façon dont les éléments doivent être affichés à l’utilisateur .C’est un langage puissant et polyvalent qui joue un rôle essentiel dans la création d'expériences web modernes et attrayantes.

* L’utilisation du framework Boostrap :

Bootstrap est un outil très pratique pour créer une interface simple, moderne et bien organisée. Il nous aide à rendre notre application jolie et facile à utiliser, même sans être expert en design.



*Figure22: Boostrap logo*

Grâce à ses composants prêts à l’emploi (comme les boutons, les formulaires, les tableaux...), on gagne du temps et notre site s’adapte automatiquement à tous les écrans (ordinateur, tablette, téléphone).

* 1. Base de données :
* Définition de SGBD MySQL

**MySQL** est un SGBD (systeme de gestion de base de données ) Open source largement utilisé , reconnue pour sa fiabilité et sa performance .



*Figure23: MySQL logo*

Il s’intègre facilement avec Python et les Frameworks web , ce qui est un avantage considérable pour notre projet , ainsi MySQL nous permettra de stocker et organiser efficacement les données essentielles au fonctionnement de notre projet.

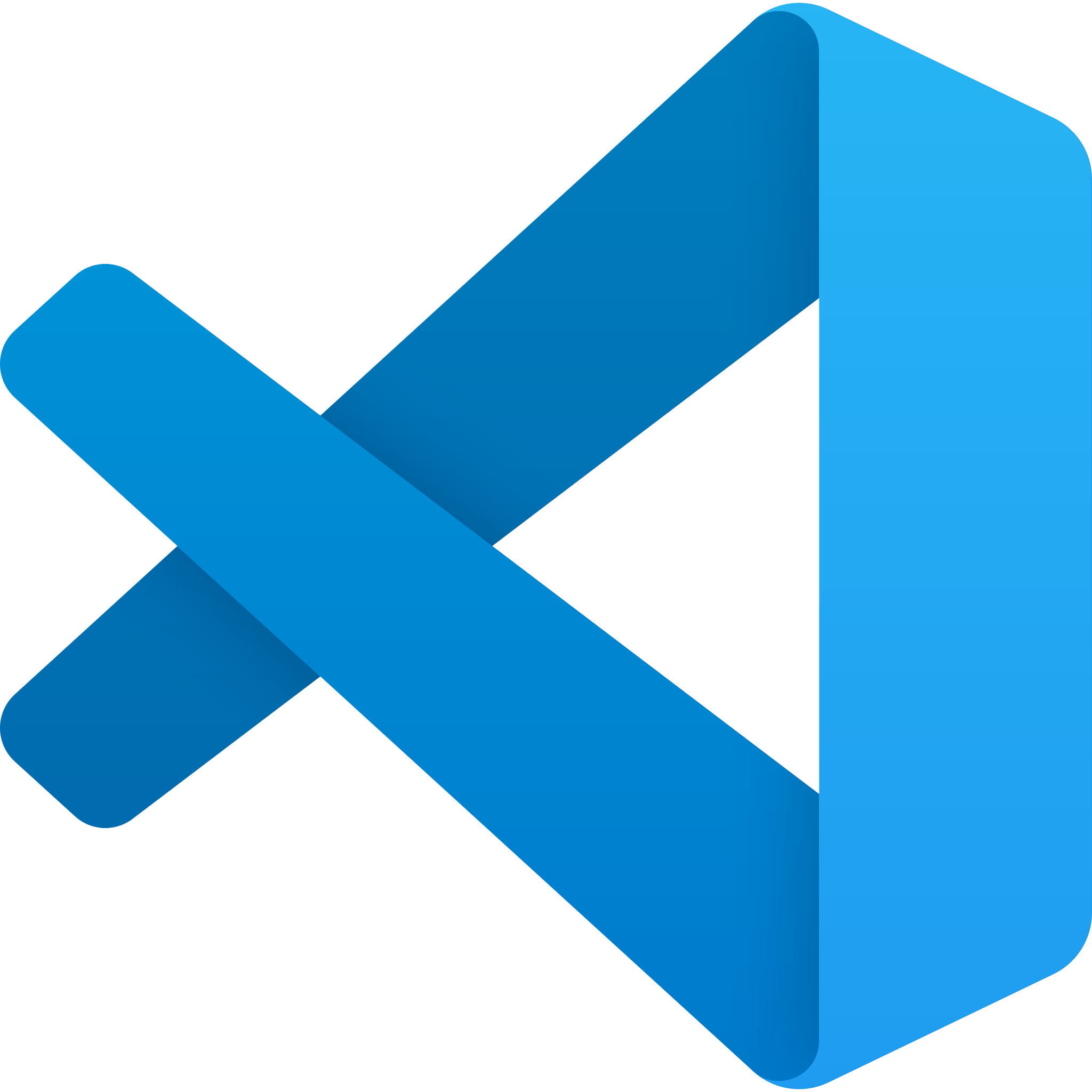
* Définition du langage de requêtes SQL

**SQL**(Structured Query Language ou Langage) est un langage informatique standardisé conçu pour gérer et manipuler des bases de données relationnelles. Le langage de requêtes structuré (SQL) est essentiel pour interagir avec notre base de données MySQL. Ce langage nous permettra de créer, modifier, récupérer et gérer les données de notre panier intelligent. Par exemple, nous utiliserons SQL pour créer des tables, insérer, mettre à jour et supprimer des données, et récupérer des données pour l'affichage dans l'application ou l'interface web. En conclusion, MySQL et SQL constituent une combinaison puissante pour gérer les données de notre projet de panier intelligent.

* Installation des outils nécessaires:

Visual studio code :

Visual Studio Code est un éditeur de code source gratuit développé par Microsoft. Il est léger, rapide et prend en charge de nombreux langages de programmation.

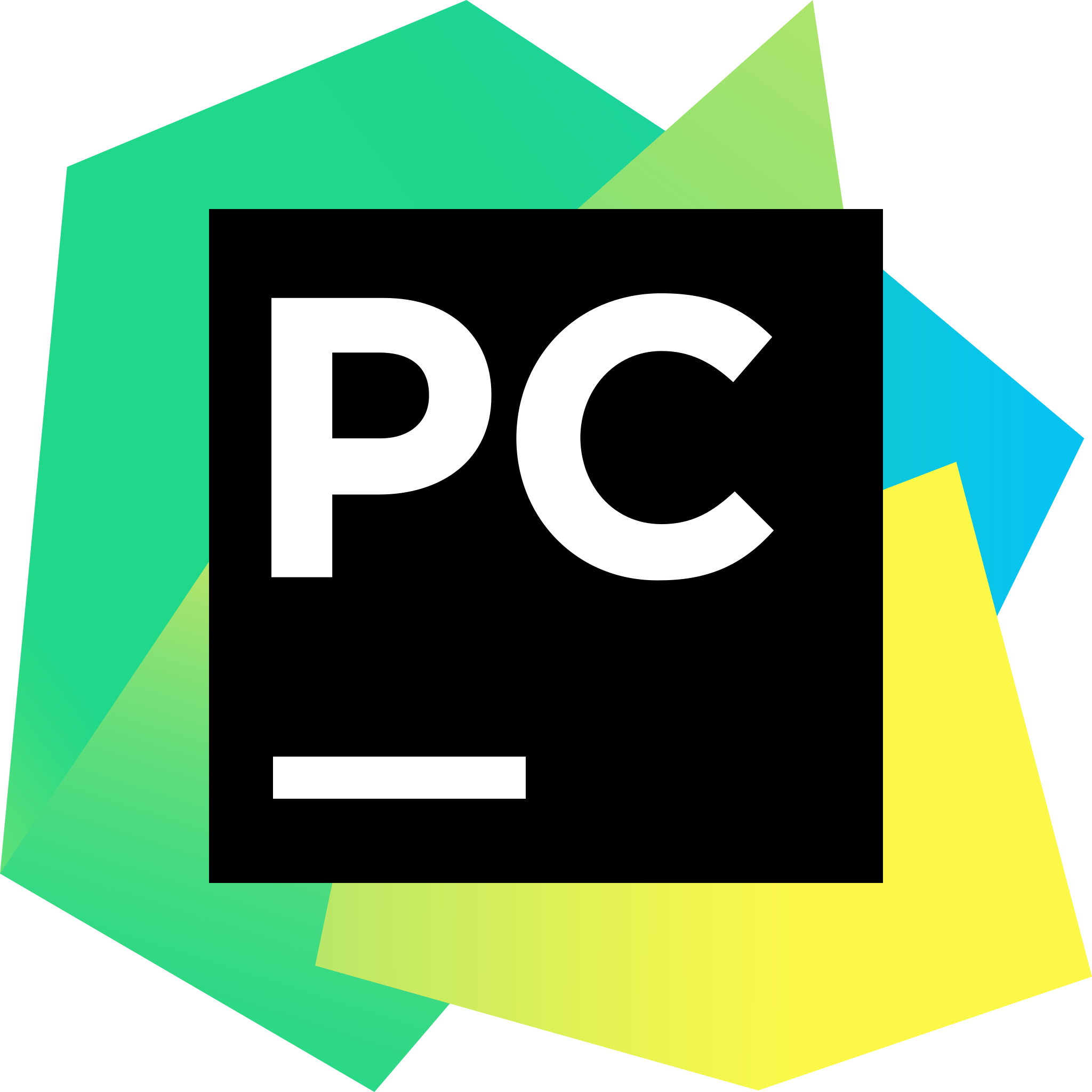


*Figure24:Visaul studio code logo*

Il propose un débogueur intégré, un terminal, des extensions, et une interface personnalisable[5].

PyCharm :

PyCharm est un environnement de développement intégré (IDE) pour Python, créé par JetBrains.



*Figure25:PyCharm logo*

Il offre des outils puissants comme l'autocomplétion du code, le débogueur, les tests et la gestion des projets, ce qui simplifie le développement Python[6].

Android Studio:

Android Studio est l’IDE officiel pour le développement d'applications Android.



*Figure26:Android studio logo*

Développé par Google, il propose des outils pour écrire du code en Kotlin ou Java, concevoir des interfaces, tester les applications, et simuler différents appareils Android.

XAMPP :

XAMPP est un environnement de serveur local gratuit qui regroupe Apache, MySQL, PHP et Perl.

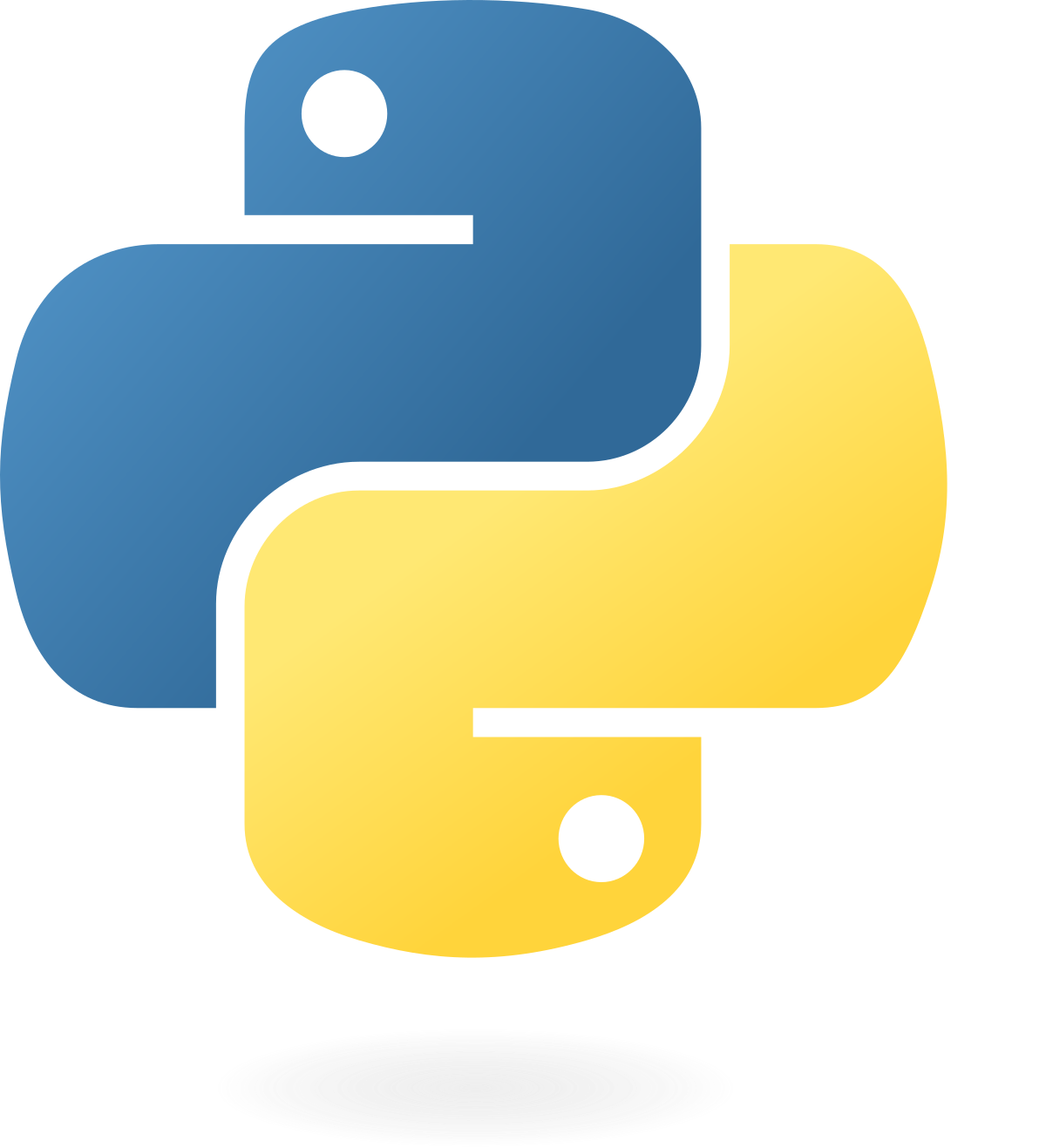


*Figure27:XAMPP logo*

Il permet aux développeurs de créer et tester des applications web localement, sans avoir besoin d’un serveur en ligne.

Python:

Python est un langage de programmation interprété, simple à apprendre et très puissant.



*Figure28:Python logo*

Il est utilisé dans le développement web, l’analyse de données, l’intelligence artificielle, l’automatisation et bien d'autres domaines.

OpenCV:

OpenCV (Open Source Computer Vision Library) est une bibliothèque open-source utilisée pour la vision par ordinateur.



*Figure29:OpenCV logo*

Elle permet d’analyser des images et des vidéos pour effectuer des tâches comme la détection de visages, d’objets ou le suivi de mouvement.

Numpy:

NumPy est une bibliothèque Python utilisée pour le calcul scientifique.

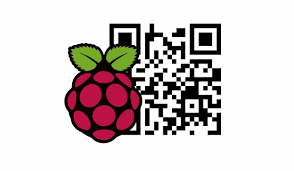


*Figure30:Numpy logo*

Elle permet de manipuler facilement des tableaux (arrays) multidimensionnels et offre de nombreuses fonctions mathématiques pour les statistiques, l’algèbre linéaire et le traitement de données numériques.

Pyzbar:

pyzbar est une bibliothèque Python qui permet de lire les codes-barres et les QR codes à partir d’images ou de vidéos.



*Figure31: Pyzbar logo*

Elle fonctionne avec OpenCV pour détecter et décoder automatiquement les symboles scannés.

Arduino IDE:

L’Arduino IDE est un environnement de développement utilisé pour écrire, compiler et transférer du code vers les cartes Arduino.



*Figure32:ArduinoIDE logo*

Il prend en charge le langage Arduino basé sur C/C++ et permet de programmer facilement des composants électroniques comme des capteurs ou des moteurs.

1. **Principales interfaces**

(Description des principales interfaces)

* Interface d’accueil
* Interface administrateur
* Interface enseignant
* Interface ajout, …

# Conclusions et perspectives

Ce projet de panier intelligent a permis de mettre en place une solution innovante visant à automatiser le processus d’achat en magasin. Grâce à l’intégration de capteurs, d’un lecteur RFID, d’une tablette et de logiciels de gestion, le système offre une expérience client plus fluide, tout en facilitant la gestion des stocks et le suivi des achats pour les administrateurs. Ce projet a également permis de renforcer nos compétences en IoT, développement web, communication série avec Arduino et base de données.

Cependant, certaines améliorations peuvent encore être apportées. À l’avenir, il serait intéressant de :

* Renforcer la sécurité des paiements et des données client.
* Ajouter des fonctionnalités d’intelligence artificielle pour la reconnaissance visuelle des produits.
* Étendre la solution à plusieurs paniers connectés en parallèle, dans un environnement réel de supermarché.

Ce projet constitue ainsi une première étape prometteuse vers des magasins plus intelligents et connectés.

# Références

**Cours : (Filière SMI - Faculté des Sciences Tétouan)**

1. Pr. EL MOHAJIR Mohamed « Gestion De Projet».2021/2022.
2. «CYCLE DE VIE D’UN LOGICIEL»-HASSAN ELBAHI
3. Pr. EL MAHOUTI Abderrahim « Conception Orientée Objets».2021/2022.
4. Pr ABDOUN Othman «Introduction-cours-M36 Architecture Distribuée».

**Sites Internet :**

1. <https://code.visualstudio.com>
2. <https://www.jetbrains.com>