



Projet de fin d'études

*Présenté en vue de l'obtention du diplôme en BTS
Informatique de gestion.*

Sujet :

*Conception et réalisation d'une application
web de gestion d'un point de vente de café
restaurant*

Présenté par : Wael Gharbi

Encadré par : Atef Jbebli

Année universitaire : 2022-2023

Dédicaces

Je dédie ce travail à :

Mes chers parents, que nulle dédicace ne peut exprimer mes sincères sentiments, pour leur patience illimitée, leur encouragement contenu, leur aide, en témoignage de mon profond amour et respect pour ses grands sacrifices.

Mon cher frère : pour leur grand amour et leur soutien qu'ils trouvent ici l'expression de ma haute gratitude.

Mes chers amis qui sans leur encouragement ce travail n'aura jamais vu le jour.

Et à toute ma famille scolaire.

Remerciements

Je veux exprimer par ces quelques lignes de remerciements mes gratitude envers tous ceux en qui, par leur présence, leur soutien, leur disponibilité et leurs conseils nous avons trouvé courage afin d'accomplir ce projet.

En fin, je ne peux achever ce projet sans exprimer mes gratitude à tous les enseignants de l'Université Sherbrooke, pour leur dévouement et leur assistance tout au long de cette année.

Table de matières

<i>Dédicaces</i>	2
<i>Remerciements</i>	3
Table de matières	4
Liste des figures	6
Introduction générale :	7
Chapitre N°1 : contexte général du projet	10
Introduction:	11
I. Présentation du contexte de projet	11
1. Sujet proposé.....	11
2. Contrainte	11
3. Objectif	11
II. Etude de l'existant	12
1. Description de l'existant	12
2. Critique de l'existant	12
3. Solution proposée.....	12
Conclusion	13
Chapitre N°2 : Analyse et spécification des besoins	14
Introduction.....	15
I. Spécification des besoins	15
a. Besoins fonctionnels.....	15
b. Besoins non fonctionnels.....	17
II. Analyse des besoins	17
1. Choix d'une méthodologie	17
2. L'approche objet	18

3.	Description du langage UML	18
4.	Demarche utilise	18
5.	Identification et presentation des cas d'utilisation	18
B.	Présentation des cas d'utilisations.....	19
Chapitre N°3 : Conception		28
Introduction.....		29
I.	L'architecture logique MVC.....	29
1.	Ses avantages	29
2.	Ses inconvénients	29
II.	Diagramme de classe	31
Conclusion		33
Chapitre N°4 : Réalisation.....		34
Introduction.....		35
I.	Application web	35
1.	Les avantages des applications web	35
2.	Les inconvénients des applications web	35
II.	Environnement de travail	35
1.	Environnement matériel	35
2.	Choix et technologies utilisées.....	36
3.	Outil de réalisation du rapport	37
III.	Présentation des interfaces	37
Conclusion générale.....		48

Liste des figures

Figure1 : Diagramme de cas d'utilisation global.....	Error! Bookmark not defined.
Figure2 : Diagramme de cas d'utilisation « s'authentifier »	20
Figure3 : Diagramme de cas d'utilisation « gestion des utilisateurs »	21
Figure4 : Diagramme de cas d'utilisation « gestion des catégories ».....	Error! Bookmark not defined.
Figure5 : Diagramme de cas d'utilisation « gestion des produits ».....	26
Figure6 : Diagramme de cas d'utilisation « gestion des ventes »	27
Figure7 : Le paradigme de PHP	31
Figure8 : Diagramme de classe	33
Figure9 : Interface d'authentification	38
Figure10 : Interface d'accueil coté administrateur	39
Figure11 : Interface gestion des catégorie	40
Figure12 : Interface ajout catégorie	40
Figure13 : Interface gestion des produits	41
Figure14 : Interface modification des produits	42
Figure15 : Interface ajout d'un produit	42
Figure16 : Interface gestion utilisateurs.....	43
Figure17 : Interface ajout d'un utilisateur.....	44
Figure18 : Interface historique de vente	44
Figure19 : Interface vente par produit	45
Figure20 : Interface vente par catégorie	46
Figure21 : Interface vente par période (partie 1).....	46
Figure22 : Interface vente par période (partie 2).....	47
Figure23 : Interface caisse.....	47

Introduction générale :

Actuellement, le monde connaît une avance technologique considérable dans tous les secteurs et cela à l'aide de l'informatique, qui joue un rôle important dans le développement de nombreuses entreprises et organisations. Avant l'invention de l'ordinateur, nous enregistrons toutes les informations manuellement sur des supports en papier. Ce qui engendrait beaucoup de problèmes tel que la perte de temps considérable dans la recherche de ces informations ou la dégradation de ces dernières.

La nouvelle logique de l'organisation du travail demande aux établissements de tourisme et divertissement d'utiliser essentiellement l'information comme matière première pour pouvoir être plus efficace. Ils doivent donc intégrer un développement du système d'information dans leurs investissements stratégique, dans la mesure où ils structurent les ventes, le stockage, l'organisation et la communication de l'information.

Aujourd'hui, les cafés restaurants auquel nous rattacherons d'ailleurs notre étude, font partie intégrante des établissements touristiques où l'informatique pourra aider. En effet, la croissance de la population estudiantine nécessite la mise en place d'une gestion rationnelle et rapide, or et jusqu'à ce jour, la manière de gérer manuellement est encore dominante d'où la nécessité d'introduire l'informatique dans ces endroits touristiques.

Problématique :

Sans l'utilisation d'un système de gestion approprié, plusieurs problèmes peuvent survenir dans la gestion d'un salon, notamment :

1. Falsification des rapports de vente : Lorsque les ventes sont enregistrées manuellement, il existe un risque de manipulation des chiffres pour dissimuler des transactions ou augmenter artificiellement les ventes. Cela peut entraîner une perte financière pour l'entreprise et fausser les analyses de performance.
2. Erreurs de saisie et d'enregistrement : La gestion manuelle des données peut entraîner des erreurs de saisie, des omissions ou des doublons lors de l'enregistrement des ventes, des stocks ou

des audits. Ces erreurs peuvent fausser les informations et rendre difficile la prise de décisions éclairées.

3. Perte de temps dans la recherche d'informations : Sans un système de gestion centralisé, retrouver des informations spécifiques sur les ventes, les produits ou les audits peut prendre beaucoup de temps. Cela peut ralentir les opérations quotidiennes et rendre difficile l'accès aux données essentielles pour les analyses ou les rapports.

4. Gestion inadéquate des stocks : Sans un suivi précis des entrées et sorties de produits, il peut être difficile de maintenir des niveaux de stock optimaux. Cela peut entraîner des problèmes de rupture de stock, des pertes financières liées à la péremption des produits ou une mauvaise gestion des commandes.

5. Difficulté à effectuer des audits internes : Sans un système de gestion qui enregistre et suit les audits internes, il peut être difficile de garantir la conformité et la sécurité des opérations. Les audits internes sont essentiels pour prévenir les pertes, détecter les problèmes de qualité et assurer la transparence dans les activités du salon.

En intégrant un système de gestion adapté à votre salon, vous pouvez prévenir ces problèmes en empêchant la falsification des rapports de vente, en assurant une saisie précise des données, en facilitant l'accès aux informations nécessaires, en optimisant la gestion des stocks et en permettant des audits internes efficaces. Cela vous aide à améliorer la transparence, à optimiser les opérations et à prendre des décisions basées sur des données précises et fiables.

L'objectif de notre projet présenté dans ce rapport est la conception et la réalisation d'une application web de gestion d'un lounge ainsi que les enseignants dans un centre de formation. Et pour la réalisation de cette tâche, notre choix s'est porté sur UML (Unified Modeling Language) comme langage de modélisation. Ce présent travail sera structuré en quatre chapitres :

☐ Le premier chapitre, est consacré à l'étude de l'existant, ses anomalies, la problématique et l'objectif du projet.

☐ Le deuxième chapitre montre l'analyse et la spécification des besoins aussi la démarche et la méthodologie adoptée pour la modélisation de notre système et la représentation informatique associée au système à développer.

☐ Dans le troisième chapitre, est consacré à la partie de conception, pour présenter le diagramme de classe, diagramme de séquence et diagramme d'activité

□ Dans Le quatrième chapitre, nous détaillons la réalisation de notre application en présentant l’environnement logiciel et matériel utilisés et les différentes techniques de réalisation. Nous allons aussi donner une vue sur l’application web dans son état final tout en présentant les différentes interfaces de cette dernière.

Chapitre N°1 : contexte général du projet

Introduction:

Dans ce chapitre, on présente l'étude de l'existant, ses anomalies, la problématique et l'objectif du projet.

I. Présentation du contexte de projet :

1. Sujet proposé :

L'application a pour but de gérer les services d'un lounge et réaliser les tâches suivantes :

- Gestion des produits
- Gestion des catégories
- Gestion des utilisateurs
- Gestion des ventes

2. Contrainte :

Voici quelques problèmes potentiels qui peuvent survenir avec votre application :

1. Difficulté à regrouper les demandes des clients : L'application peut rencontrer des difficultés pour ajouter une nouvelle demande d'un client à une commande précédente. Cela peut entraîner une confusion dans le suivi des demandes et rendre la gestion des commandes moins efficace.
2. Problèmes de synchronisation des données en temps réel : cela peut entraîner des incohérences et des erreurs dans les informations partagées.
3. Manque de support technique et de mises à jour régulières : cela peut entraîner des problèmes de stabilité, de sécurité ou de compatibilité avec d'autres systèmes. Les utilisateurs pourraient rencontrer des difficultés techniques non résolues ou être confrontés à des problèmes de compatibilité.

3. Objectif :

L'objectif du projet consiste à concevoir et à développer une application web qui permettra de répondre aux besoins de l'administration de la lounge.

Pour ce faire, dans notre application on va essayer de minimiser les tâches manuelles tel que la gestion des produits, la gestion des catégories et la gestion des ventes et d'implémenter une application qui répond aux requêtes des différents utilisateurs : l'administrateur et le caissier ou bien le serveur.

En effet, le caissier peut seulement réaliser des ventes dans lequel s'enregistre sur la base de donnée.

II. Etude de l'existant :

1. Description de l'existant :

L'étude de l'existant est une phase importante pour bien comprendre le système actuel et définir ses objectifs. Pour chaque module, il sera question d'effectuer une description précise de l'existant en énumérant les principaux acteurs impliqués, les principaux documents manipulés et les moyens de traitements utilisés.

Avant d'aborder cette étape importante, il s'avère nécessaire de circonscrire le périmètre de notre étude. Dans le cadre de ce projet, les principaux modules étudiés sont là :

- Gestion des utilisateurs
- Gestion des produits
- Gestion des catégories
- Gestion des ventes

2. Critique de l'existant :

Les logiciels EXCEL et WORD de Microsoft Corporation, bien qu'ils soient puissants ne satisfont pas à tous les besoins spécifiés dans le cahier des charges relatif à ce projet.

En effet certaines fonctionnalités ne se font pas de manière automatique et nécessite une intervention manuelle qui est la plupart du temps fastidieuse.

L'objectif visé est de satisfaire aux besoins des secrétaires en réduisant au maximum la charge de travail due aux différents traitements de l'information. Dans un souci de concevoir une application avec plus de fonctionnalités possibles et dans le but d'avoir une interface plus conviviale et plus facile à utiliser tout en étant plus efficace, Je vais mettre sur pied une application informatique regroupant tous les modules cités ci-dessus.

3. Solution proposée :

Pour résoudre ces problèmes, j'ai développé une application web pour améliorer la gestion de lounge « LoungeWhiz », cette application permettra une meilleure gestion de ce service ainsi elle permettra de faciliter le travail et de gagner plus de temps. L'application doit garantir :

- Une meilleure gestion des produits
- Une meilleure gestion des catégories
- Une meilleure gestion ventes
- Une meilleure méthode de vente.
- Une meilleure gestion des utilisateurs.

Conclusion

Dans ce chapitre, on a présenté en premier lieu le cadre général et le sujet de notre projet de fin d'étude. En deuxième lieu, analyser l'existant de système et spécifier ses différents besoins, ce qui nous a menés à concevoir une solution plus appropriée et par la suite présenter la méthode de conception à suivre pour l'implémentation de notre projet. Une étude conceptuelle détaillée sera représentée dans le chapitre suivant.

Chapitre N°2 : Analyse et spécification des besoins

Introduction:

La conception d'un système d'information est une étape fondamentale dans le processus de conduite du projet. Cette activité de conception doit suivre une méthode appropriée au contexte du sujet. Dans ce chapitre, on va procéder à présenter la méthode de conception utilisée pour notre application. Puis expliquer les principaux modèles de conception du système tout en précisant les principaux diagrammes.

I. Specification des besoins :

a. Besoins fonctionnels :

Cette phase correspond à une recherche sur le terrain pour bien définir le cadre de notre système.

Le but principal de l'application est de permettre aux utilisateurs la gestion des activités suivantes dont on va détailler chacune d'elles. L'application doit permettre à ses utilisateurs de se connecter à une interface.

1. Gestion des produits :

- ✓ Ajouter un produit
- ✓ Modifier un produit
- ✓ Supprimer un produit
- ✓ Voir la liste des produits

Ajouter un produit:

L'administrateur peut ajouter un nouveau produit dans la liste des produits en enregistrant ses informations et en affectant le nom, la catégorie, une photo de produit, le prix et une description.

Modifier un produit :

L'administrateur peut modifier un produit dans le cas d'un changement

Supprimer un produit :

L'administrateur peut supprimer un produit dans la liste des produits

Voir la liste des produits :

L'administrateur peut consulter facilement la liste des produits.

Chercher un produit :

L'administrateur peut chercher un produit avec son nom.

2. Gestion des Catégories :

- ✓ Ajouter une catégorie

- ✓ Modifier une catégorie
- ✓ Supprimer une catégorie
- ✓ Voir la liste des catégories

Ajouter une catégorie :

L'administrateur peut ajouter une nouvelle catégorie dans la liste des catégories en enregistrant son nom.

Modifier une catégorie :

L'administrateur peut modifier une catégorie dans le cas d'un changement

Supprimer une catégorie :

L'administrateur peut supprimer une catégorie dans la liste des catégories

Voir la liste des catégories :

L'administrateur peut consulter facilement à la liste des catégories.

3. Gestion des utilisateurs :

- ✓ Ajouter un utilisateur
- ✓ Modifier un utilisateur
- ✓ Supprimer un utilisateur
- ✓ Voir la liste des utilisateurs

Ajouter un utilisateur :

L'administrateur peut ajouter un nouvel utilisateur dans la liste des utilisateurs en enregistrant son nom et son mot de passe.

Modifier un utilisateur :

L'administrateur peut modifier un utilisateur dans le cas d'un changement

Supprimer un utilisateur :

L'administrateur peut supprimer un utilisateur dans la liste des utilisateurs

Voir la liste des utilisateurs :

L'administrateur peut consulter facilement à la liste des utilisateurs.

4. Gestion des ventes :

- ✓ Réaliser une vente
- ✓ Modifier une vente
- ✓ Voir la liste des ventes
- ✓ Passer une commande

Caissier :

Le caissier peut réaliser une vente et passer une commande qui sera visible pour lui jusqu'à l'encaissement.

Administrateur :

L'administrateur peut réaliser une vente, passer une commande, voir toutes les commandes non encaissées et voir l'historique des ventes et commandes.

b. Besoins non fonctionnels :

Les besoins non fonctionnels représentent les propriétés du système, les contraintes d'environnement et d'implémentation, les dépendances des plates formes et la fiabilité.

Le site web doit garantir :

La fiabilité :

Il faut s'assurer que le portail réalise d'une manière continue les services attendus.

La validité :

Il faut s'assurer que le site web réalise la fonctionnalité définie par la spécification.

La confidentialité :

- Les utilisateurs doivent s'authentifier par un login et un mot de passe.
- Le système doit assurer la confidentialité et l'intégrité des données.

La sécurité :

Attribuer les droits d'accès précis pour chaque utilisateur.

L'ergonomie :

Le site doit avoir des interfaces conviviales et ergonomiques.

II. Analyse des besoins :

1. Choix d'une méthodologie :

Dans le cadre de notre projet de fin d'étude, on va opter pour une méthode orientée objet basé sur UML. Avant de présenter la démarche choisie, on va commencer par un bref aperçu de l'approche objet et du langage UML.

2. L'approche objet:

Elle a pour but de modéliser des propriétés statiques et dynamiques de l'environnement dans lequel sont définis les besoins appelés domaine du problème. Le couplage devient alors dynamique et les évolutions fonctionnelles ne remettent plus en cause la structure alors dynamique statique du logiciel, ce qui réduit la distorsion entre le réel et le système informatique.

3. Description du langage uml (Unified modeling language) :

UML représente l'état des langages de modélisation objet. Il fournit le fondement pour spécifier, construire, visualiser et décrire les artefacts d'un système informatique.

Alors UML peut substituer sans perte d'information, les autres modélisations orienté objet des notations celle de BOOCH, OMT ou encore OOSE. Il se place comme le successeur naturel des notations définies dans le cadre de ces méthodes ; il s'est d'ailleurs imposé à la fois auprès des utilisateurs et sur le terrain de la normalisation.

UML offre une multitude des diagrammes structurels et comportementaux pour représenter respectivement des vues statiques et dynamiques d'un système. En effet, un diagramme donne à l'utilisateur un moyen de visualiser et de manipuler des éléments de modélisation. On présente dans ce qui suit les diagrammes d'UML.

4. Démarche utilise :

La démarche de conception qu'on va suivie dans notre modélisation avec UML dans le cadre de ce projet est une démarche intuitive et qui s'adapte mieux à l'environnement de notre application et qui se présente comme suit :

Etape1 : Diagramme de cas d'utilisation global.

Etape2 : Diagramme de cas d'utilisation raffiné.

5. Identification et présentation des cas d'utilisation :

A. Identification

a. Définition :

Les diagrammes des cas d'utilisation décrivent sous forme d'action et de réaction, le comportement du système de point de vue utilisateur, ils permettent de définir les limites et les relations entre le système et son environnement et ils recouvrent toutes les phases de développement depuis le cahier des charges jusqu' à l'implémentation.

Un cas d'utilisation représente l'utilisation et le fonctionnement du système.

On va commencer par identifier les acteurs qui interagissent avec le système. Par la suite, nous allons présenter les fonctionnalités (cas d'utilisations) attendues de la part du système.

b. Les acteurs :

- ✓ **L'administrateur** : Étant l'acteur principal, l'administrateur peut effectuer plusieurs Fonctionnalités à savoir : gestion des produits, gestion des catégories, gestion des utilisateurs et gestion des ventes.
- ✓ **Le Caissier** : le caissier est l'acteur secondaire dans notre étude. Il peut passer des commandes pour être encaisser au futur, il peut encaisser ces mêmes commandes et il peut consulter les commandes non encaissées sur son nom.

c. Les cas d'utilisation :

Les cas d'utilisation permettent de représenter le fonctionnement du système vis-à-vis de l'utilisateur. Il s'agit d'une vue du système dans son environnement extérieur. Chaque cas d'utilisation peut être spécifié sous forme de séquences d'informations entrantes et sortantes, indiquant l'aspect dynamique du système.

B. Présentation des cas d'utilisations

1. Diagramme de cas d'utilisation global

Le diagramme de cas d'utilisation global ci-dessous met en relief les cas d'utilisation du système.

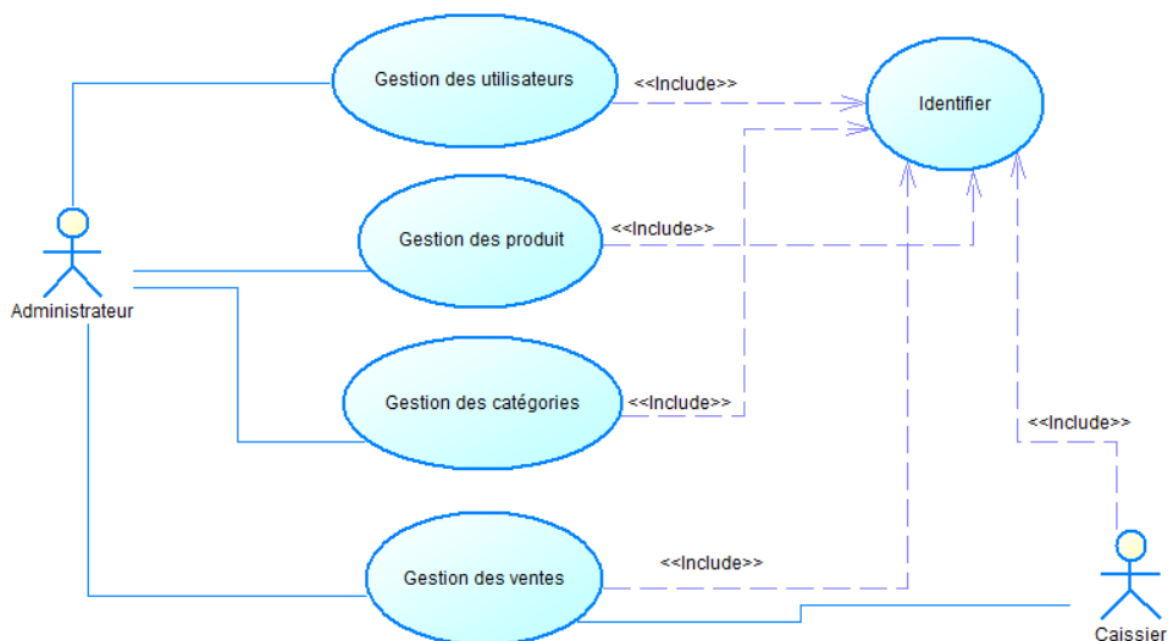


Figure1 : Diagramme de cas d'utilisation global

2. Le cas d'utilisation « s'authentifier »

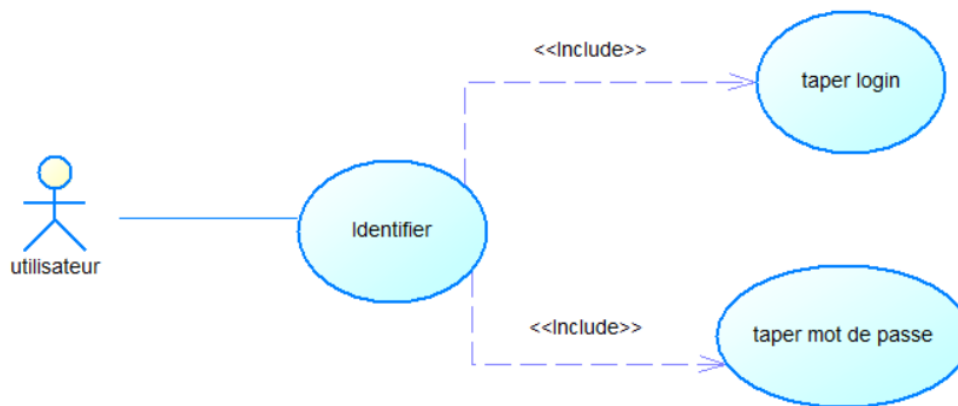


Figure2 : Diagramme de cas d'utilisation « s'authentifier »

- Acteur principal : utilisateurs (Administrateur, Caissier).
- Objectif : S'assurer que l'utilisateur est bien celui qui prétend être.
- **Scenario nominal :**
 1. L'utilisateur saisit son nom d'utilisateur et son mot de passe.
 2. Le système vérifie le nom d'utilisateur et le mot de passe.
 3. Le système affiche l'espace approprié pour chaque utilisateur.
- **Scenario alternatif :**
 1. Login et mot de passe sont incorrects, un retour vers la page d'authentification sera effectué avec un message d'erreur.

3. Le cas d'utilisation « gestion des utilisateurs »

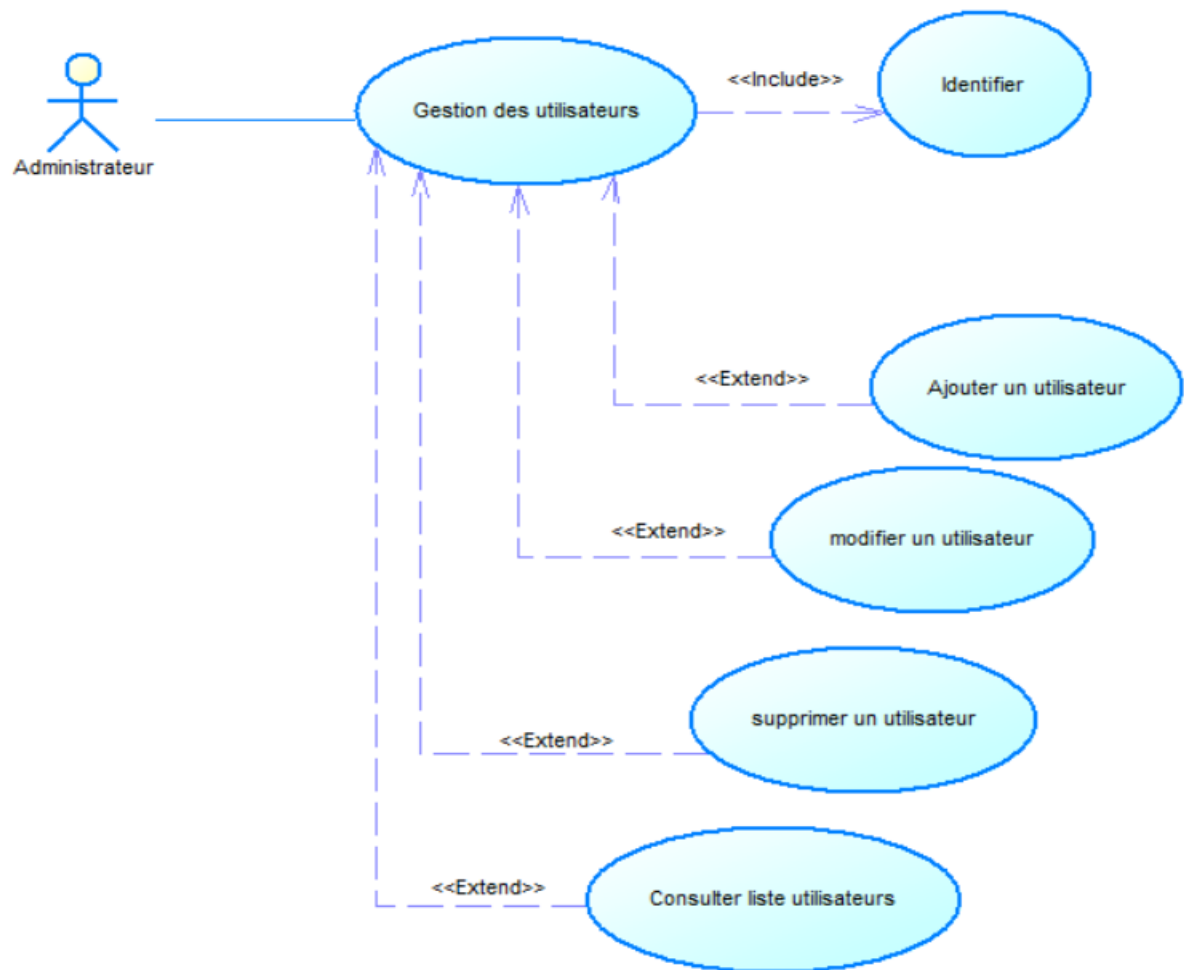


Figure3 : Diagramme de cas d'utilisation « gestion des utilisateurs »

- Acteur principal : Administrateur
- Objectif : Pouvoir ajouter, modifier, supprimer un utilisateur

• Scenario nominal :

Cas 1 : Créer un administrateur

1. L'administrateur choisit d'ajouter un utilisateur
2. Le système affiche le formulaire à remplir.
3. L'administrateur remplit et valide le formulaire.
4. Le système ajoute les informations dans la base.
5. Le système actualise la liste des utilisateurs et l'affiche.

Cas 2 : Modifier un administrateur

1. L'administrateur supérieur choisit l'utilisateur à modifier.
2. Le système affiche le formulaire de modification.
3. Il modifie les champs voulus.
4. Le système met à jour les informations dans la base.
5. Le système actualise la liste des utilisateurs et l'affiche.

Cas 3 : Supprimer un administrateur

1. L'administrateur supérieur choisit l'utilisateur à supprimer.
2. Le système demande une confirmation.
3. L'administrateur confirme ou annule la suppression.
4. Le système supprime l'utilisateur de la base.
5. Le système actualise la liste des utilisateurs et l'affiche.

Scénario alternatif :

Cas 1 : Modification avec des champs vides, champs non conformes aux types : un message d'erreur sera affiché.

4. Le cas d'utilisation « gestion des catégories »

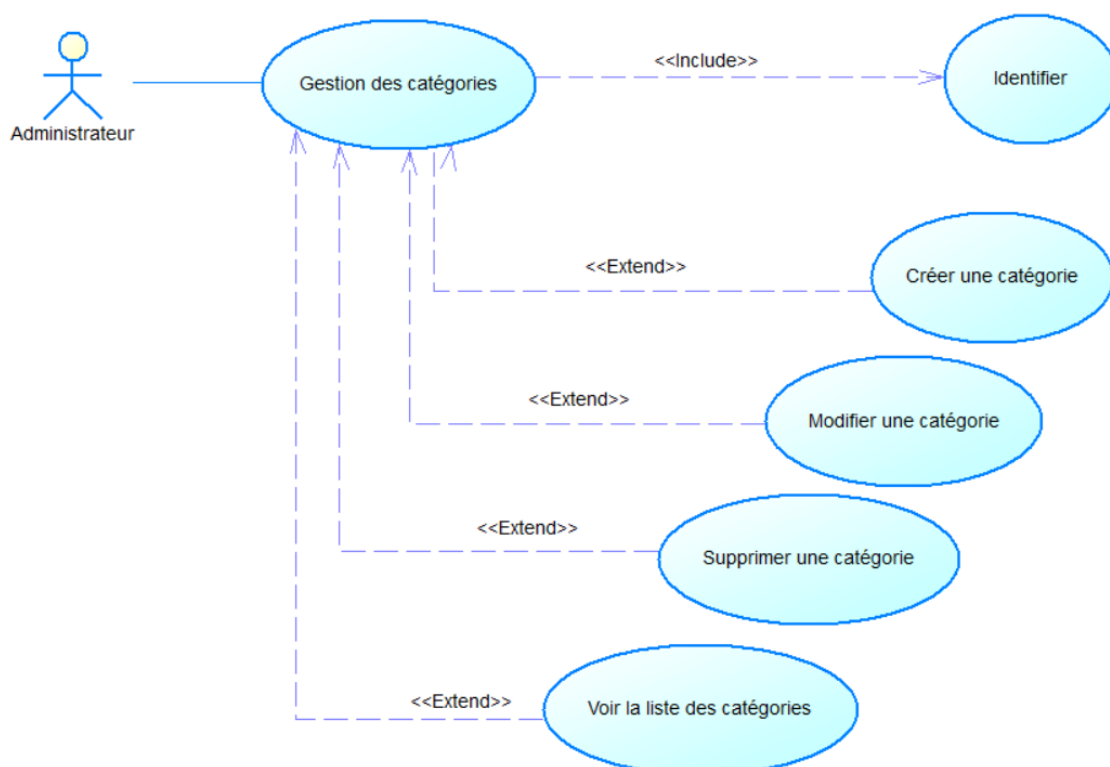


Figure 4 : Diagramme de cas d'utilisation « gestion des catégories »

- Acteur principal : Administrateur
- Objectif : Pouvoir ajouter, modifier, supprimer une catégorie

• **Scenario nominal :**

Cas 1 : Créer une catégorie

1. L'administrateur choisit d'ajouter une catégorie
2. Le système affiche le formulaire à remplir.
3. L'administrateur remplit et valide le formulaire.
4. Le système ajoute les informations dans la base.
5. Le système actualise la liste des catégories et l'affiche.

Cas 2 : Modifier une catégorie

1. L'administrateur supérieur choisit la catégorie à modifier.
2. Le système affiche le formulaire de modification.
3. Il modifie les champs voulus.
4. Le système met à jour les informations dans la base.
5. Le système actualise la liste des catégories et l'affiche.

Cas 3 : Supprimer une catégorie

1. L'administrateur choisit la catégorie à supprimer.
2. Le système demande une confirmation.
3. L'administrateur confirme ou annule la suppression.
4. Le système supprime la catégorie de la base.
5. Le système actualise la liste des catégories et l'affiche.

Scenario alternatif :

Cas 1 : Modification avec des champs vides, champs non conformes aux types : un message d'erreur sera affiché.

Cas 2 : ajout d'une catégorie avec un nom qui existe déjà dans la base : un message d'erreur sera affiché.

5. Le cas d'utilisation « gestion des produits »

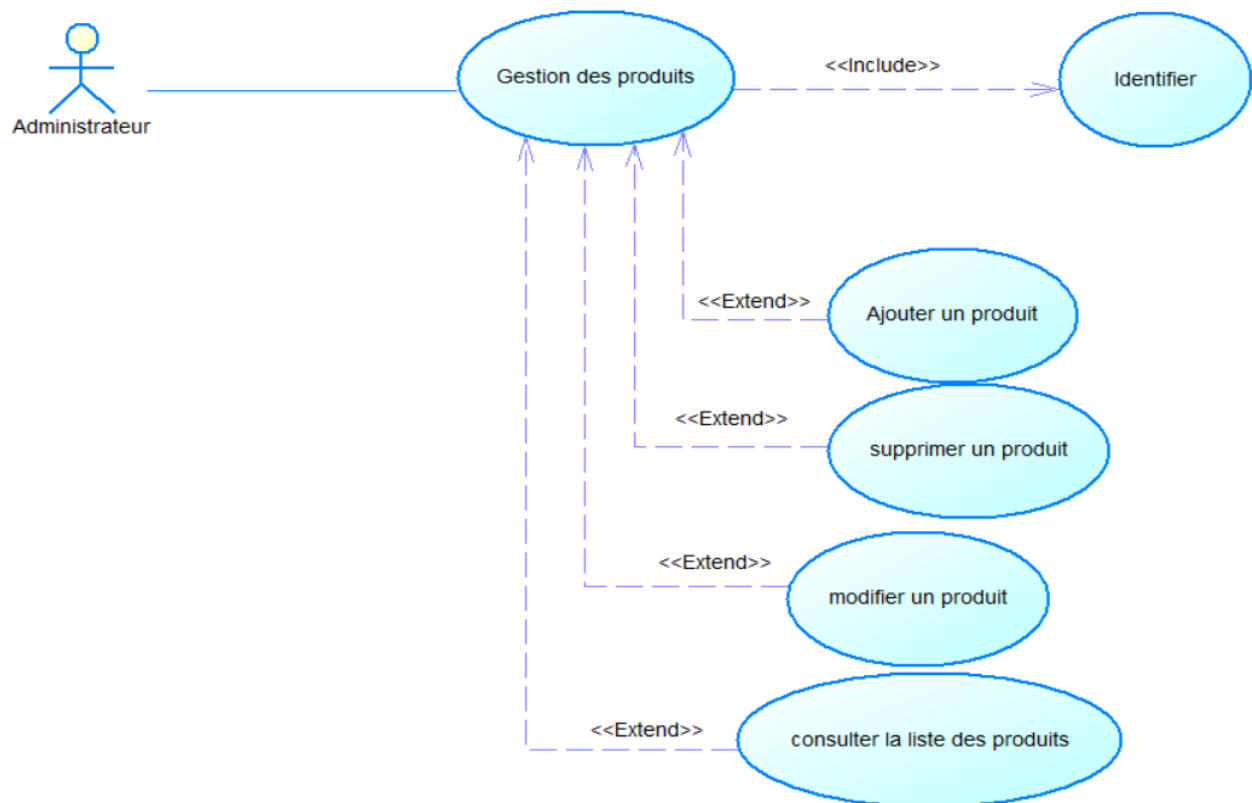


Figure 5 : Diagramme de cas d'utilisation « gestion des produits »

- Acteur principal : Administrateur
- Objectif : Pouvoir ajouter, modifier, supprimer un produit

• Scenario nominal :

Cas 1 : Créer une catégorie

1. L'administrateur choisit d'ajouter un produit
2. Le système affiche le formulaire à remplir.
3. L'administrateur remplit et valide le formulaire.
4. Le système ajoute les informations dans la base.
5. Le système actualise la liste des produits et l'affiche.

Cas 2 : Modifier une catégorie

1. L'administrateur supérieur choisit le produit à modifier.
2. Le système affiche le formulaire de modification.

3. Il modifie les champs voulus.
4. Le système met à jour les informations dans la base.
5. Le système actualise la liste des produits et l'affiche.

Cas 3 : Supprimer une catégorie

1. L'administrateur choisit le produit à supprimer.
2. Le système demande une confirmation.
3. L'administrateur confirme ou annule la suppression.
4. Le système supprime le produit de la base.
5. Le système actualise la liste des produits et l'affiche.

Scénario alternatif :

Cas 1 : Modification avec des champs vides, champs non conformes aux types : un message d'erreur sera affiché.

Cas 2 : Ajout d'une photo qui a une extension autre que jpeg, jpg, gif ou png : un message d'erreur sera affiché

Cas 3 : Ajout d'une photo avec une large taille : un message d'erreur sera affiché

6. Le cas d'utilisation « gestion des ventes »

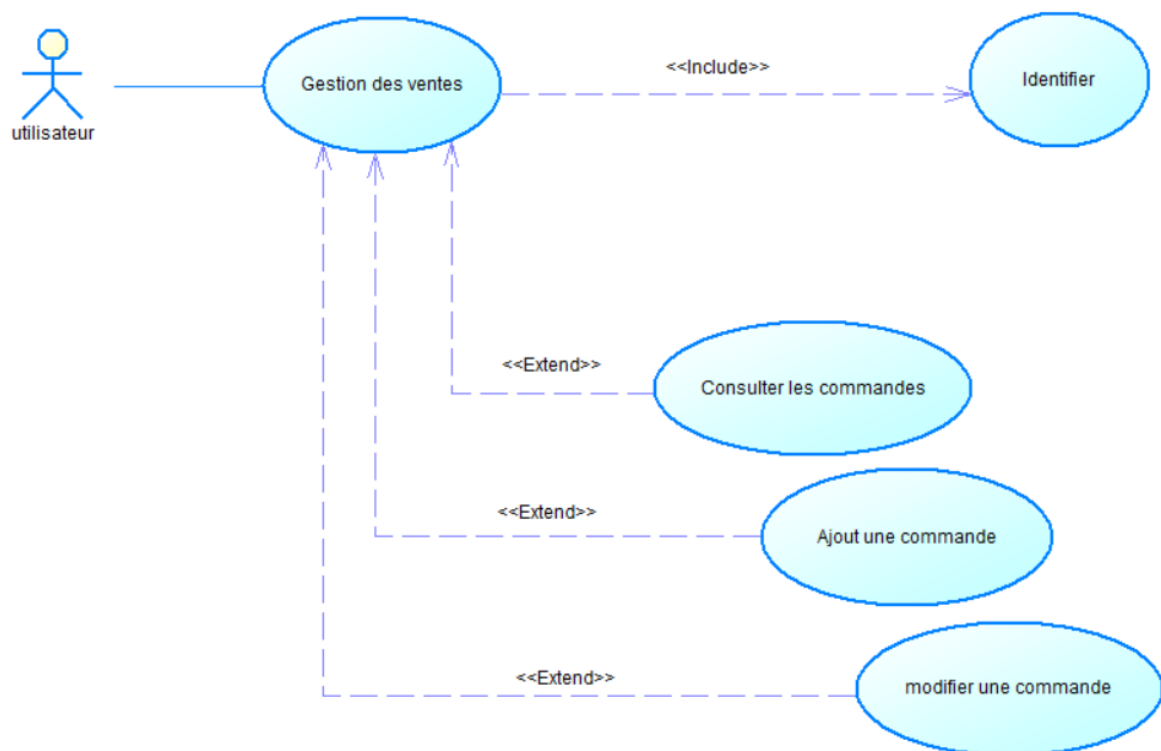


Figure 6 : Diagramme de cas d'utilisation « gestion des ventes »

- Acteur principal : Administrateur, Caissier
- Objectif : Pouvoir consulter, ajouter et modifier une commande

• Scenario nominal :

Cas 1 : Créer une commande

1. L'utilisateur choisit d'ajouter une commande
2. Le système affiche les produits disponibles
3. L'utilisateur valide ses choix et confirme la commande
4. Le système ajoute les informations dans la base.

Cas 2 : Modifier une commande

1. L'utilisateur choisit de modifier une commande
2. Le système affiche les commandes non encaissées
3. L'utilisateur choisit la commande à modifier et change l'état de la commande de non encaissée à encaissée.
4. Le système enregistre les modifications dans la base.

Cas 3 : Consulter les commandes

1. L'utilisateur choisit de consulter les commandes
2. Le système affiche les commandes non encaissées enregistrée avec l'identifiant d'utilisateur dans le cas de caissier et affiche toutes les commandes non encaissées dans le cas de l'admin

Conclusion

Dans la première partie de ce chapitre, nous avons décrit le contexte de notre projet puis nous avons posé la problématique et tracer quelques objectifs. Dans la deuxième partie, nous nous sommes intéressés à analyser les besoins de notre application et à spécifier les différents cas d'utilisations en abordant le diagramme des cas d'utilisations. Dans ce qui suit nous allons entamer la conception en présentant le diagramme de classe.

Chapitre N°3 :

Conception

Introduction

Après l'analyse des besoins fonctionnels et les déterminations des spécifications, on présente dans ce chapitre la conception du projet à travers une multitude de diagrammes UML.

D'abord elle permet de décrire le produit à développer, en détaillant sa structure, les éléments qui le composent ainsi le comportement de chacun de ces éléments. Ce chapitre expose en détail la partie conceptuelle de notre travail à savoir le diagramme de classe.

I. L'architecture logique MVC

Le pattern d'architecture logicielle modèle-vue-contrôleur (en abrégé MVC, en anglais model-view-Controller), tout comme les patterns modèle –vue-présentation ou présentation, abstraction, contrôle, est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective.

Cette méthode présente des avantages et des inconvénients :

1. Ses avantages :

- Séparation des compétences (design, base de données, application).
- Simplicité de mise à jour.
- Vitesse de création de pages.

2. Ses inconvénients

- Pages plus lentes à afficher (hors caches).
- Plus de ressources consommées.
- Développement initial plus long.

Ce paradigme regroupe les fonctions nécessaires en trois catégories :

- Un modèle (modèle de données).
- Une vue (présentation, interface utilisateur).
- Un contrôleur (logique de contrôle, gestion des événements, synchronisation).

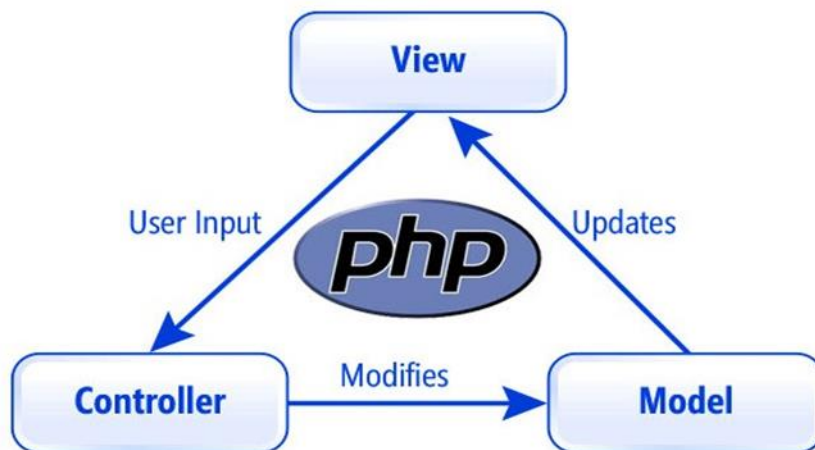


Figure 7 : Le paradigme de PHP

Nous expliquons ces trois parties l'une après l'autre :

1. Le model

Cette partie gère les *données* de votre site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve donc entre autres les requêtes SQL.

2. La Vue

Cette partie se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple une liste de messages.

3. Le contrôleur

Cette partie gère la logique du code qui prend des *décisions*. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le contrôleur contient exclusivement du PHP. C'est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d'accès).

La figure suivante schématise le rôle de chacun de ces éléments.



4. Identification de cycle de vie du système :

Le cycle de vie du logiciel est une représentation conventionnelle de succession des étapes permettant la mise en œuvre d'un produit logiciel.

L'analyse, qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir

Une idée de ce que va réaliser le système en termes de métier. Les résultats de l'analyse ne dépendent d'aucune technologie particulière.

→ La branche architecture technique comporte :

- + La capture des besoins techniques, qui recense toutes les contraintes et les choix dimensionnant la conception du système. Les outils et les matériels sélectionnés ainsi que la prise en compte de contraintes d'intégration avec l'existant conditionnent généralement des prérequis d'architecture technique.
- + La conception générique, qui définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels. Elle a pour objectif d'uniformiser et de réutiliser les mêmes mécanismes pour tout un système. L'architecture technique construit le squelette du système informatique et écarte la plupart des risques de niveau technique. L'importance de sa réussite est telle qu'il est conseillé de réaliser un prototype pour assurer sa validité.

→ La branche de phase de réalisation :

- + La conception préliminaire, qui représente une étape délicate, car elle intègre le modèle d'analyse dans l'architecture technique de manière à tracer la cartographie des composants du système à développer.
- + La conception détaillée, qui étudie ensuite comment réaliser chaque composant l'étape de codage, qui produit ces composants et teste au fur et à mesure les unités de code réalisées.
- + L'étape de recette, qui consiste enfin à valider les fonctions du système développé.

II. Diagramme de classe

En général un diagramme de classe peut contenir les éléments suivants :

- + Les classes : une classe représente la description formelle d'un ensemble d'objets ayant une sémantique et des caractéristiques communes. Elle est représentée en utilisant un rectangle divisé en trois sections. La section supérieure est le nom de la

classe, la section centrale définit les propriétés de la classe alors que la section du bas énumère les méthodes de la classe.

✚ Les associations : une association est une relation entre deux classes (association binaire) ou plus (association aire), qui décrit les connexions structurelles entre leurs instances. Une association indique donc que des liens peuvent exister entre des instances de classes associées.

✚ Les attributs : les attributs représentent les données encapsulées dans les objets des classes. Chacune de ces informations est définie par un nom, un type de données, une visibilité et peut être initialisé. Le nom de l'attribut doit être unique dans la classe

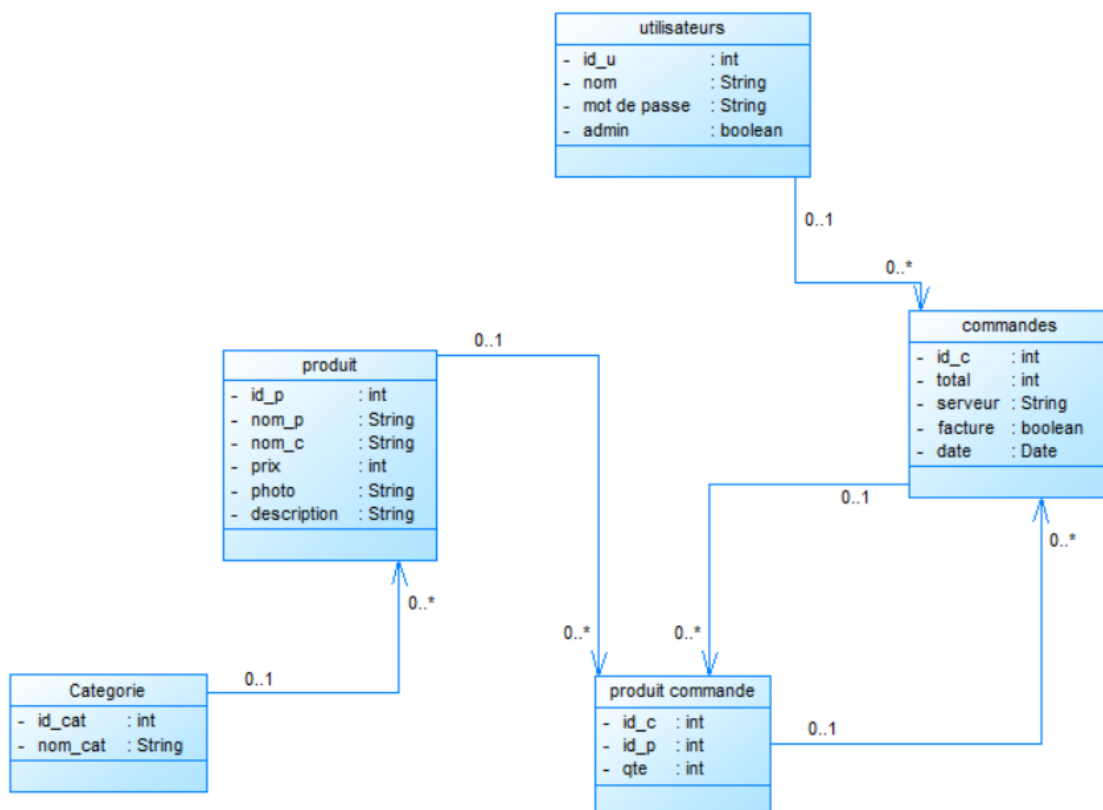


Figure 8 : Diagramme de classe

Conclusion

Durant ce chapitre, nous avons essayé de présenter la partie conceptuelle de notre application, tout en mettant l'accent sur la présentation de classe du système. Nous avons aussi présenté la conception de la base de données. Le chapitre suivant traitera la phase de mise en œuvre de l'application.

Chapitre N°4 :

Réalisation

Introduction

Après l'étude préalable, la spécification des besoins et la conception réalisée dans les chapitres précédents, nous commencerons la partie réalisation de notre projet, nous présenterons l'architecture web, l'environnement matériel et logiciel et aussi que les langages de programmation utilisés.

I. Application web

En informatique, une application web (aussi appelée web application, de l'anglais) est une application manipulable directement en ligne grâce à un navigateur web et qui ne nécessite donc pas d'installation sur les machines clientes, contrairement aux applications mobiles.

1. Les avantages des applications web :

- Aucune installation pour l'utilisateur.
- Mises à jour simplifiées.
- Mises à jour plus fréquentes.
- Une meilleure expérience utilisateur.
- Calcul de données capable de monter en charge.
- Sécurité des contenus.
- Un coût de développement et de maintenance restreint.
- Un accès plus rapide

2. Les inconvénients des applications web :

- L'application est obligée d'être connectée à internet pour pouvoir utiliser le service web.
- En cas de suppression de service web, L'application entraînera une erreur.
- Les services Web souffrent de performances faibles comparées à d'autres approches de l'informatique répartie.

II. Environnement de travail

1. Environnement matériel

Le développement de l'environnement matériel est caractérisé par :

- PC Portable : Lenovo
- Système d'exploitation : Windows 11
- Mémoire : 8.00 Go
- Processeur : 2.40 GHZ

2. Choix et technologies utilisées

Les outils logiciels, les langages et les différentes technologies utilisées pour la réalisation de notre projet sont les suivants :

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données.



Visual Studio Code est un éditeur de code source léger, mais puissant, qui s'exécute sur votre bureau et est disponible pour Windows, MacOS et Linux.



PHP : hypertexte Préprocesseur plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.



MySQL est la base de données open source la plus populaire au monde, ce qui permet la présentation rentable de haute performance fiable et des applications de base de données mobile et intégrés évolutives. Il est facile à utiliser. Vous pouvez bénéficier de fonctionnalités avancées, des outils de gestion et de soutien technique pour développer, déployer et gérer vos applications, MySQL est une open source système de gestion de base de données relationnelle (SGBDR).



JavaScript est un langage de programmation qui permet de créer du contenu mis à jour de façon dynamique, de contrôler le contenu multimédia, d'animer des images, et tout ce à quoi on peut penser. Bon, peut-être pas tout, mais vous pouvez faire bien des choses avec quelques lignes de JavaScript.



3. Outil de réalisation du rapport :

Microsoft Word est un processeur de texte commercial largement utilisé par Microsoft. Microsoft Word est un composant du logiciel Microsoft office suite de productivité, mais peut également être acheté en tant que produit autonome.

Il a été initialement lancé en 1983 et a été révisé à plusieurs reprises. Microsoft Word est disponible pour les systèmes d'exploitation Windows et Macintosh.

Microsoft Word s'appelle simplement Word ou MS Word.



Il représente un logiciel qui nous permet de modéliser les traitements informatiques et leurs bases de données associées et qui gère la plupart des diagrammes spécifiés dans la norme UML2.0, elle est basée sur le langage de modélisation UML (Unified Modeling Language) que nous avons adopté pour toute la suite de notre travail.



III. Présentation des interfaces

1. Interface Identification

A screenshot of a web application's login interface. At the top, the 'LoungeWhiz' logo is displayed. Below it, the heading 'Connectez Vous à Votre Compte' is centered, followed by the instruction 'Entrez Votre Nom et Mot de Passe'. The form contains two main sections: 'Nom d'utilisateur' with a dropdown menu showing 'Mr / Mme :' and a text input field, and 'Mot de passe' with a text input field. A blue 'Login' button is positioned below the password field. At the bottom of the form, it says 'Designed by Wael Gharbi'.

Figure 9 : Interface d'identification

✓ Description textuelle :

Cette interface représente la page d'authentification pour l'administrateur et le caissier ou bien le serveur.

2. Interface Administrateur

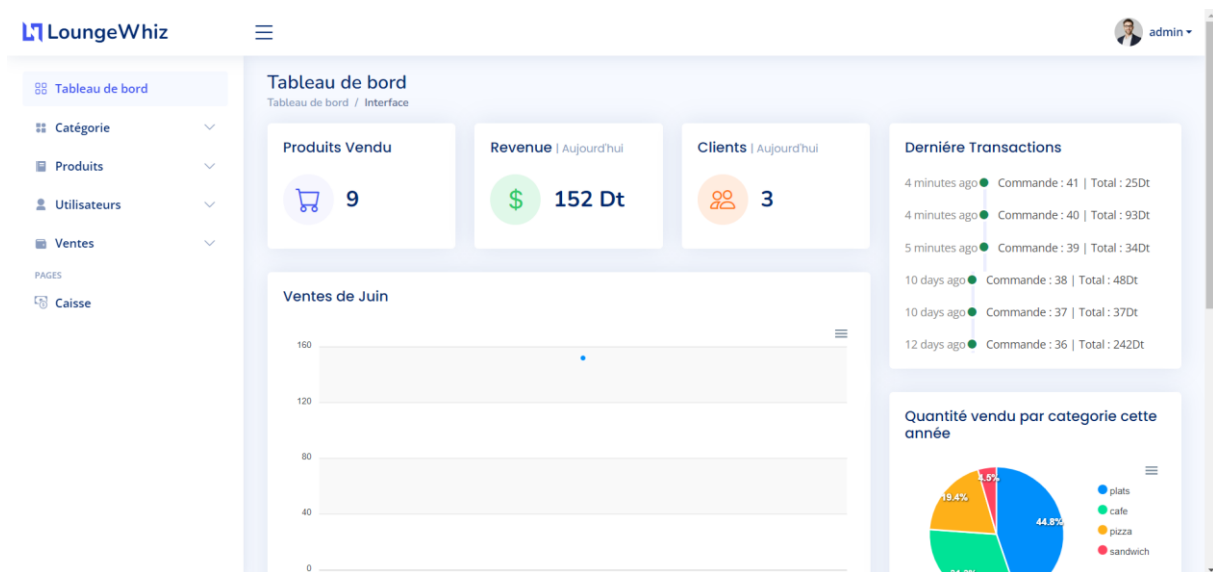


Figure 10 : Interface d'administrateur

✓ Description textuelle :

Cette interface représente la page d'accueil de l'administrateur où il peut trouver plusieurs graphes analytiques concernant les ventes.

3. Interface Gestion des Catégories

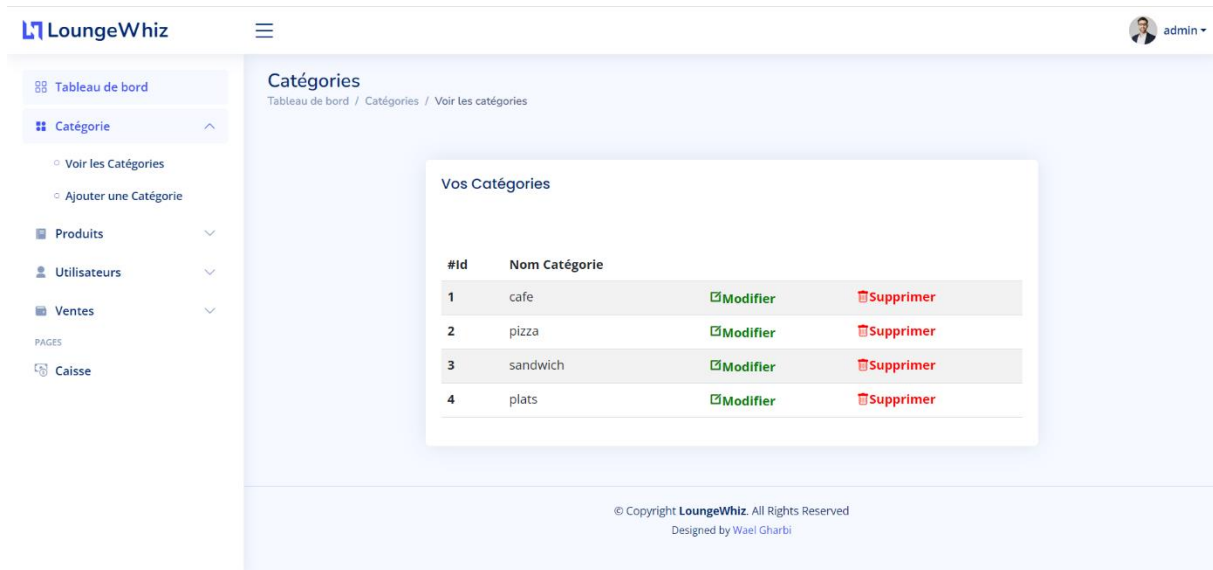


Figure 11 : Interface gestion des catégories

✓ Description textuelle :

Cette interface représente la liste des catégories, on peut modifier le nom d'une catégorie ou bien le supprimer.

4. Interface d'ajout d'une nouvelle catégorie

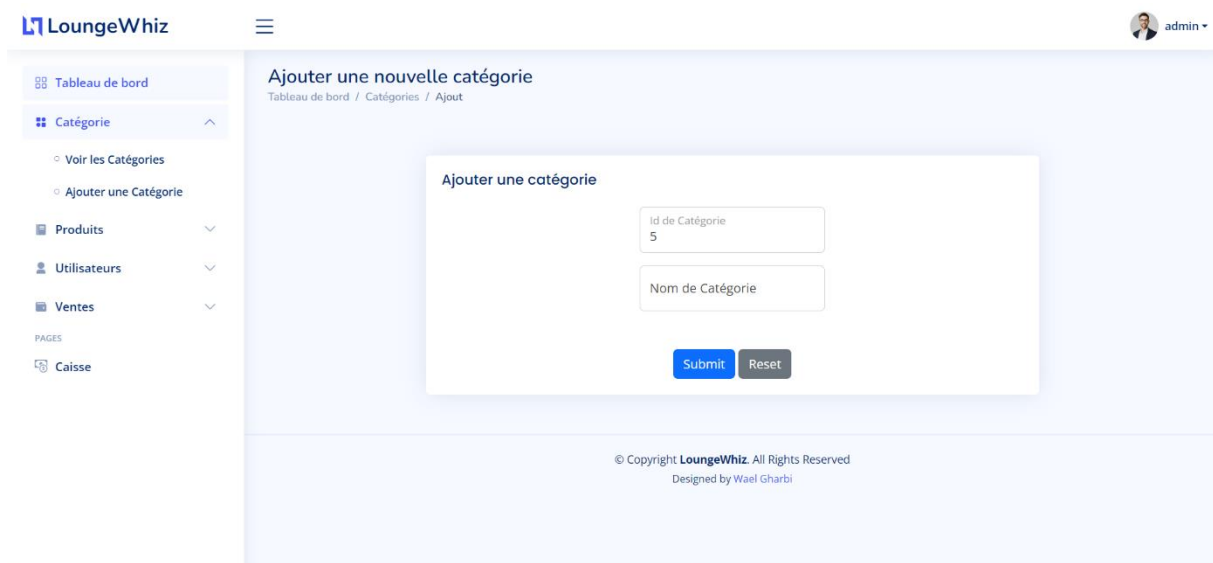
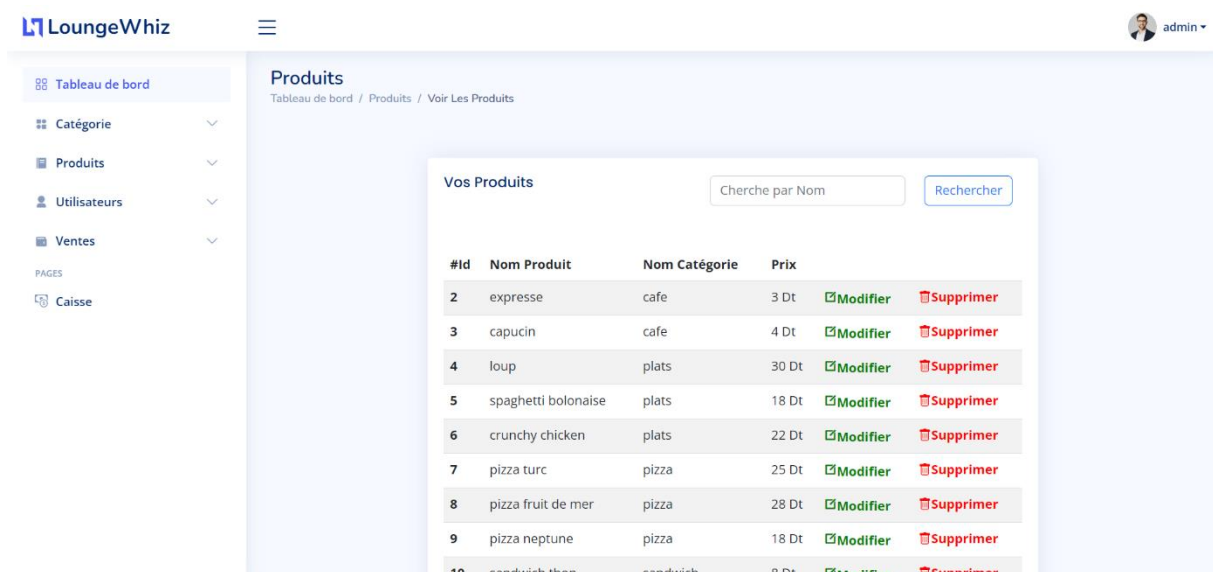


Figure 12 : Interface ajout d'une catégorie

✓ Description textuelle :

Cette interface nous permis d'ajouter une nouvelle catégorie à la base et le system nous donne un numéro d'identifiant automatiquement.

5. Interface gestion produit



Vos Produits

Cherche par Nom

#Id	Nom Produit	Nom Catégorie	Prix		
2	expresse	cafe	3 Dt	<input checked="" type="checkbox"/> Modifier	<input checked="" type="checkbox"/> Supprimer
3	capucin	cafe	4 Dt	<input checked="" type="checkbox"/> Modifier	<input checked="" type="checkbox"/> Supprimer
4	loup	plats	30 Dt	<input checked="" type="checkbox"/> Modifier	<input checked="" type="checkbox"/> Supprimer
5	spaghetti bolonaise	plats	18 Dt	<input checked="" type="checkbox"/> Modifier	<input checked="" type="checkbox"/> Supprimer
6	crunchy chicken	plats	22 Dt	<input checked="" type="checkbox"/> Modifier	<input checked="" type="checkbox"/> Supprimer
7	pizza turc	pizza	25 Dt	<input checked="" type="checkbox"/> Modifier	<input checked="" type="checkbox"/> Supprimer
8	pizza fruit de mer	pizza	28 Dt	<input checked="" type="checkbox"/> Modifier	<input checked="" type="checkbox"/> Supprimer
9	pizza neptune	pizza	18 Dt	<input checked="" type="checkbox"/> Modifier	<input checked="" type="checkbox"/> Supprimer
10	sandwich thon	sandwich	8 Dt	<input checked="" type="checkbox"/> Modifier	<input checked="" type="checkbox"/> Supprimer

Figure 13 : Interface gestion de produit

✓ Description textuelle :

Cette interface représente la liste des produits enregistrée dans la base. On a l'option de modifier un produit ou bien le supprimer.

6. Interface modification produit

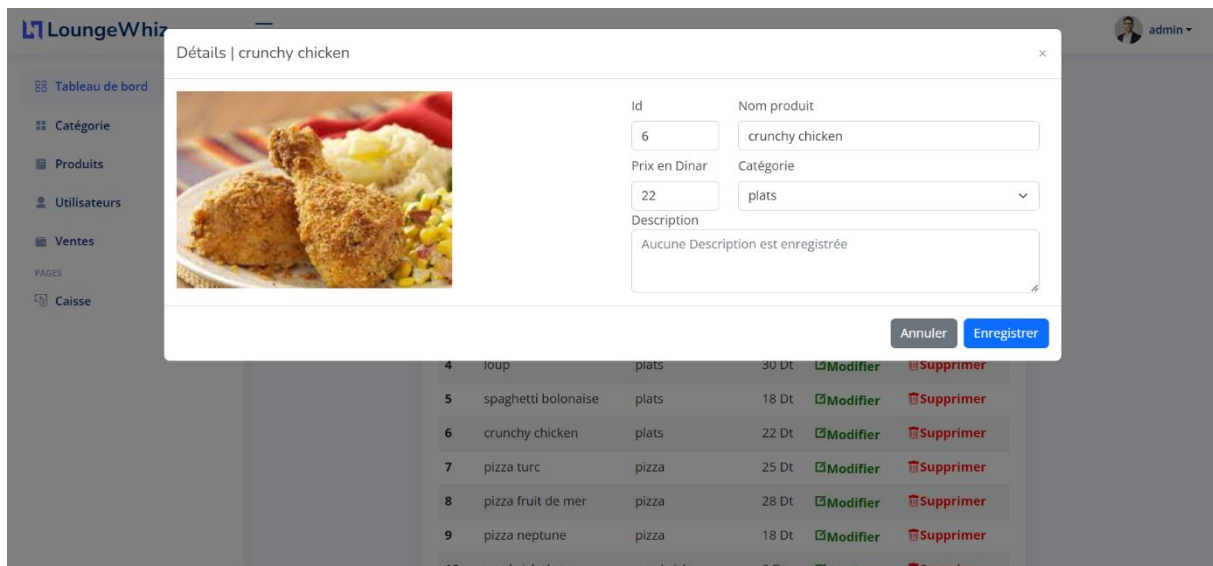


Figure 14 : Interface modification produit

✓ Description textuelle :

Cette interface représente le formulaire qui nous permet de modifier un produit, on peut changer tous sauf l'image et l'identifiant.

7. Interface Ajout Produit

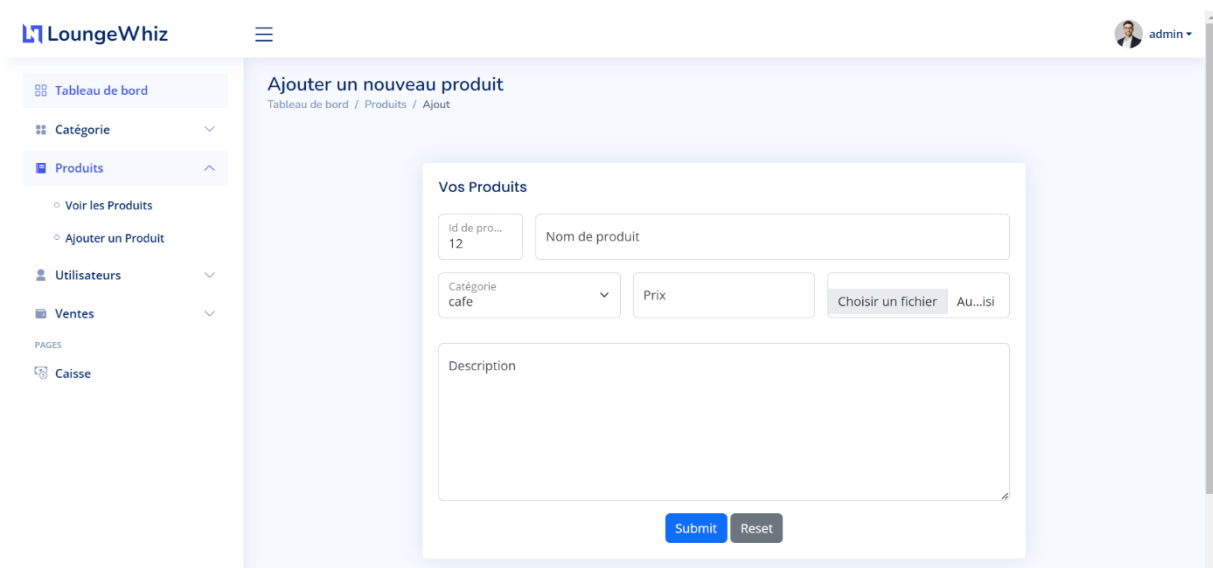


Figure 15 : Interface ajout produit

✓ Description textuelle :

Cette interface nous permis d'ajouter un nouveau produit à la base et le system nous donne un nouvel identifiant automatiquement.

8. Interface gestion utilisateurs

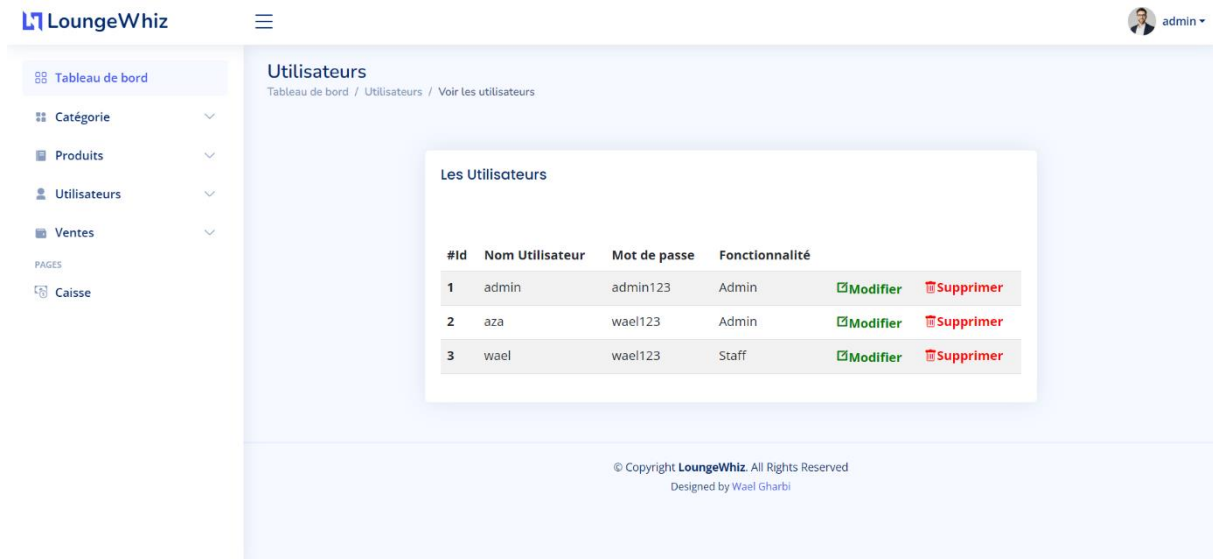


Figure 16 : Interface gestion utilisateurs

✓ Description textuelle :

Cette interface représente la liste des utilisateurs enregistrée dans la base. Il nous donne la possibilité de modifier le nom, le mot de passe et la fonctionnalité d'un utilisateur sélectionnée et il nous donne aussi la possibilité de le supprimer aussi

9. Interface Ajout utilisateurs

Ajouter Un Nouveau Utilisateur

Tableau de bord / Utilisateurs / Ajout

Ajouter Un Utilisateur

Id d'utilisateur: 4

Nom d'utilisateur:

Mot de passe:

Fonctionnalité: Staff

[Enregistrer](#) [Annuler](#)

© Copyright LoungeWhiz. All Rights Reserved
Designed by Wael Gharbi

Figure 17 : Interface Ajout utilisateur

✓ Description textuelle :

Cette interface nous permet d'ajouter un nouvel utilisateur à la base avec un identifiant unique et fonctionnalité.

10. Interface Historique de vente

Ventes

Home / Ventes / Voir tous les ventes

jj/mm/aaaa [Chercher Par Date](#)

Id	Serveur	Total	Payé	Date	action	action
49	admin	618Dt	1	2023-06-15 15:48:55	Voir détail	Annuler
48	admin	46Dt	1	2023-06-15 12:39:17	Voir détail	Annuler
47	admin	176Dt	1	2023-06-14 14:34:15	Voir détail	Annuler
46	admin	37Dt	1	2023-06-14 14:32:12	Voir détail	Annuler
45	admin	30Dt	0	2023-06-12 12:18:03	Voir détail	Annuler
44	admin	66Dt	1	2023-06-12 12:16:33	Voir détail	Annuler
43	admin	33Dt	1	2023-06-12 12:16:05	Voir détail	Annuler

Figure 18 : Interface historique de vente

✓ Description textuelle :

Cette interface représente tout l'historique des ventes avec ses détails et la possibilité d'annuler et supprimer une vente de la base. On peut aussi chercher les ventes d'un jour spécifique.

11. Interface ventes par produit

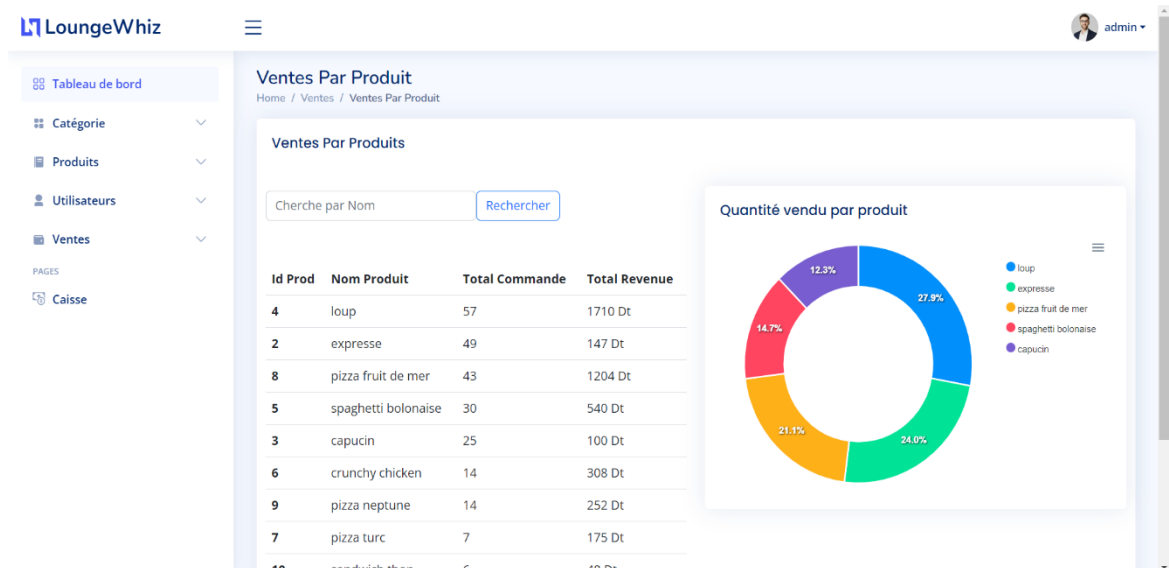


Figure 19 : Interface ventes par produit

✓ Description textuelle :

Cette interface représente les meilleures ventes par produits avec un table qui nous informe des quantités totales et les revenus totales. On a aussi un graphe qui représente la même chose. Il nous permet aussi de chercher les détails de vente d'un produit spécifique.

12. Interface ventes par catégorie

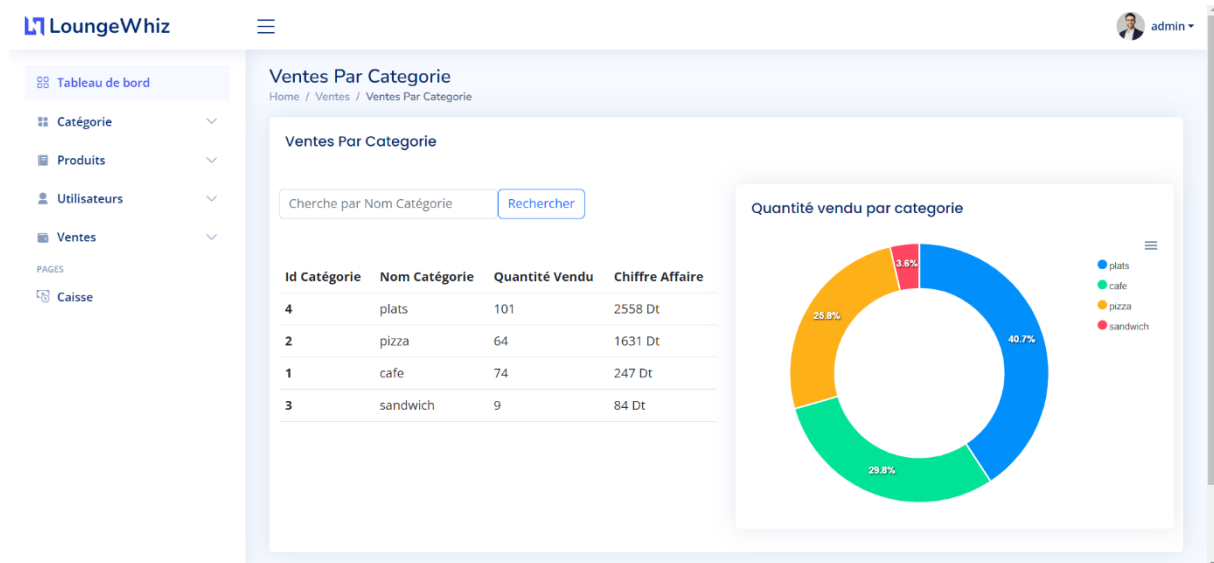


Figure 20 : Interface ventes par catégorie

✓ Description textuelle :

Cette interface représente les meilleures ventes par catégorie avec un table qui nous informe des quantités totales et les revenus totales. On a aussi un graphe qui représente la même chose. Il nous permet aussi de chercher les détails de vente d'une catégorie spécifique.

13. Ventes par période

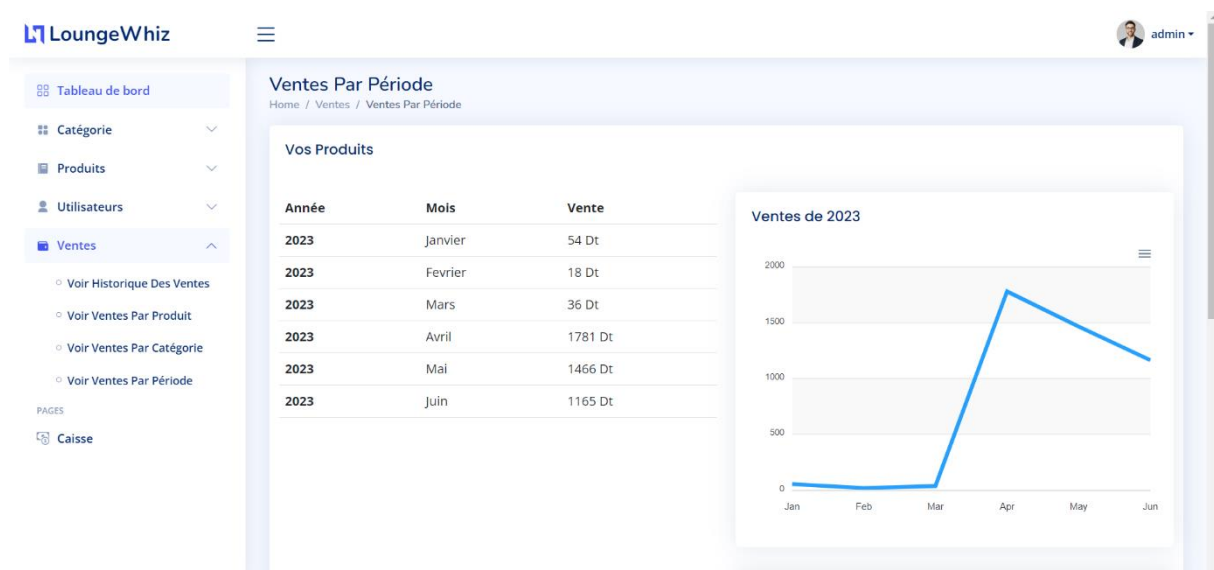


Figure 21 : Interface ventes par période 1

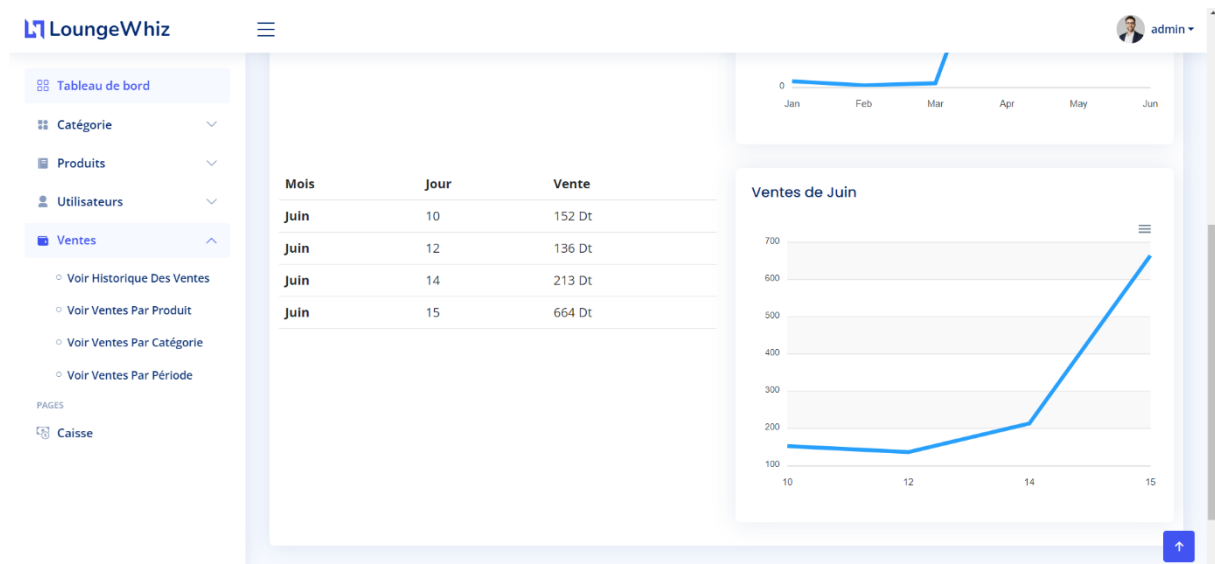


Figure 22 : Interface ventes par période 2

✓ Description textuelle :

Cette interface représente les ventes par année et mois dans le premier tableau et le premier graphe exprime la même chose. Le deuxième tableau représente les ventes du mois courant et même chose pour son graphe.

14. Interface caisse

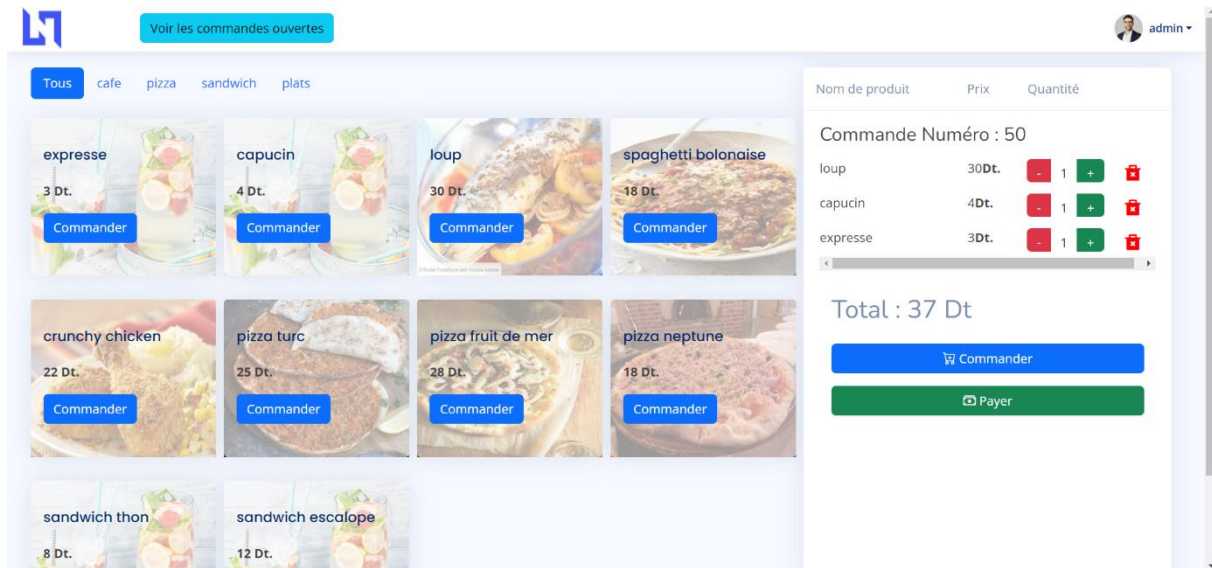


Figure 23 : Interface caisse

✓ Description textuelle :

Cette interface représente la caisse où les opérations de ventes s'enregistrent. Il nous donne l'option de chercher tous les produits ou bien les filtrer selon les catégories. La partie gauche de l'écran nous affiche les produits choisis, leurs prix et quantités et le total de facture avec l'option de facturer directement la commande ou bien le mettre en attente pour encaissement, on peut trouver cette partie quand on clique sur le bouton à le haut de page.

Conclusion générale

Notre projet de fin d'études nous a permis de dégager beaucoup d'expérience et de faire un grand pas du théorique vers le pratique puisqu'il y a une large distance entre l'informatique aux champs de la professionnelle. C'est une bonne occasion pour bien développer et exploiter nos capacités d'observation, d'analyse, de rédaction et de programmation.

Il nous a permis d'améliorer et d'approfondir notre connaissance requise durant nos trois années d'études.

De même, il nous a offert la possibilité de comprendre et d'exploiter la méthodologie de conception « UML »

Ce rapport comporte les différentes étapes de réalisation de notre système qui sont élaborées en quatre chapitres : la cadre générale du projet, analyse et spécification des besoins, étude conceptuelle, réalisation et environnement de travail. Enfin nous clôturons ce rapport en une conclusion générale.

En concluant, il serait important de noter que des améliorations peuvent être introduites au niveau d'application. Entre autres les possibilités d'ajouter un deuxième langage, d'introduire d'autres interfaces publicitaires et aussi le développement d'un système de communication entre les contacts sécurisés.