

Université de Poitiers

Licence Informatique

Année universitaire 2024-2025

RAPPORT DE STAGE

Développement d'une application de commande de nourriture avec Java, Java Swing et MySQL

Nom de l'étudiant : LE QUANG HUNG

Numéro d'étudiant : 21908455

Entreprise d'accueil : IT&M TECHNOLOGY SOLUTIONS JOINT STOCK COMPANY

Tuteur en entreprise : M. BUI VIEN PHUONG

Enseignant référent : M. SUBRENAT GILLES

Période de stage : du 19 mai 2025 au 19 juin 2025

Remerciements

Je tiens à exprimer ma sincère gratitude à l'ensemble des personnes qui m'ont accompagné et soutenu durant mon stage.

Je remercie tout d'abord l'équipe de **IT&M TECHNOLOGY SOLUTIONS JOINT STOCK COMPANY** de m'avoir accueilli chaleureusement au sein de leur entreprise, ainsi que de m'avoir donné l'opportunité de participer à un projet concret et formateur. Un remerciement tout particulier à **M.BUI VIEN PHUONG**, mon tuteur de stage, pour son encadrement, ses conseils techniques et sa disponibilité tout au long de cette période.

Je remercie également **M.SUBRENAT GILLES**, mon enseignant référent, pour son suivi attentif et ses retours constructifs.

Enfin, je souhaite remercier ma famille et mes proches pour leur soutien moral constant, qui m'a permis d'aborder ce stage dans les meilleures conditions.

Table de matières

Chapitre 1 : Présentation de l'entreprise et du stage.....	5
1.1 Présentation de l'entreprise.....	5
1.2 Présentation du stage	5
1.3 Environnement de travail.....	6
1.4 Gestion de projet.....	6
a. Planification hebdomadaire.....	6
b. Réunions et communication.....	6
Chapitre 2 : Contexte et objectifs du projet.....	7
2.1 Problématique / besoin identifié	7
2.2 Objectifs fonctionnels et techniques	7
2.3 Contraintes et attentes	7
Chapitre 3 : Outils et technologies utilisées.....	8
3.1 Langage de programmation : Java	8
a. Concept	9
b. Diagramme de hiérarchie des classes Java Swing.....	10
c. Les différences entre Java AWT et Java Swing	11
d. Fonctionnalités de Java Swing	12
3.3 Base de données : MySQL	12
3.4 Environnement de développement : NetBeans, XAMPP	13
a. NetBeans :	13
b. XAMPP	14
3.5 Autres outils.....	15
Chapitre 4 : Conception et architecture de l'application.....	15
4.1 Analyse fonctionnelle	15
4.2 Diagrammes UML	16
4.3 Structure technique du projet.....	16
4.4 Schéma de base de données.....	17
Chapitre 5 : Développement et mise en œuvre	18
5.1 Description des interfaces principales	18

5.2 Fonctions réalisées	18
5.3 Difficultés rencontrées et solutions apportées	19
5.4 Tests réalisés et résultats	19
5.5 Résultats expérimentaux.....	20
a. Interface de connexion	20
b. Interface d'accueil	21
c. Interface de gestion des produits :.....	24
d. Interface d'administration	30
e. Interface de chiffre d'affaires	30
Chapitre 6 : Résultats et évaluation	31
6.1 Fonctionnalités implémentées	31
6.2 Fonctionnalités non réalisées.....	31
6.3 Perspectives d'évolution	31
6.4 Retours éventuels de l'entreprise.....	31

Introduction

Dans le cadre de ma formation en **Licence Informatique** à l'Université de Poitiers, il m'a été demandé d'effectuer un **stage de fin d'études** afin de mettre en pratique les compétences acquises tout au long de mon cursus. Cette expérience professionnelle constitue une étape essentielle pour comprendre le fonctionnement concret d'un environnement de développement logiciel en entreprise.

Le monde de la restauration, comme beaucoup d'autres secteurs, est en pleine transformation numérique. Face à l'évolution des habitudes de consommation et à la nécessité de fluidifier les processus internes, les solutions de commande numérique deviennent incontournables. Dans ce contexte, mon stage chez **IT&M TECHNOLOGY SOLUTIONS JOINT STOCK COMPANY**, basé à Hô-Chi-Minh-Ville (Vietnam), m'a permis de participer à un projet concret : **le développement d'une application de commande de nourriture**, sous forme d'application de bureau.

L'objectif principal du projet était de concevoir un outil simple, intuitif et fonctionnel permettant à un utilisateur de passer des commandes, gérer les produits, imprimer des factures et consulter les revenus.

Ce rapport présente dans un premier temps l'entreprise d'accueil et le contexte du stage, avant de détailler le projet, ses objectifs, les outils utilisés, les étapes de développement ainsi que les résultats obtenus. Il se conclut par une évaluation personnelle de l'expérience acquise et des perspectives d'amélioration possibles.

Chapitre 1 : Présentation de l'entreprise et du stage

1.1 Présentation de l'entreprise

Le stage s'est déroulé au sein de l'entreprise **IT&M TECHNOLOGY SOLUTIONS JOINT STOCK COMPANY**, également connue sous le nom abrégé IT&M JSC. Il s'agit d'une société par actions à capital privé, fondée le 14 décembre 2015 et implantée au 197 Trần Văn Kiểu, Phường 10, Quận 6, Hồ-Chi-Minh-Ville, Vietnam.

L'activité principale de l'entreprise est l'édition de logiciels. Bien que de taille modeste avec un nombre limité d'employés, IT&M JSC se spécialise dans le développement de solutions logicielles personnalisées destinées aux petites et moyennes entreprises. Elle intervient dans plusieurs domaines comme les applications de gestion, les outils internes, ou encore les plateformes de services numériques.

Grâce à sa structure légère et agile, l'entreprise offre un environnement de travail dynamique et propice à l'apprentissage pour les jeunes développeurs. Ce cadre m'a permis de découvrir le processus de développement logiciel dans un contexte professionnel réel.

1.2 Présentation du stage

Mon stage de fin d'études en Licence Informatique s'est déroulé du 19 mai au 19 juin 2025 chez IT&M JSC. L'objectif principal était de concevoir et développer une application de bureau permettant la commande de nourriture, destinée à être utilisée dans les restaurants, cafétérias ou services de cantine.

Ce projet a été réalisé en Java avec Java Swing pour l'interface graphique, et MySQL comme système de gestion de base de données. Le développement a été effectué à l'aide de l'IDE NetBeans, et la gestion des données via XAMPP (MySQL + phpMyAdmin).

Les principales fonctionnalités implémentées sont :

- Authentification (admin / employé).
- Navigation par catégories de plats.
- Gestion du panier (ajout, suppression, total).
- Paiement et impression de la facture.
- Gestion des produits (CRUD).
- Suivi des revenus (par jour / par mois).
- Interface d'administration avec gestion des comptes utilisateurs.

Le projet m'a permis de mettre en œuvre mes connaissances théoriques dans un cadre pratique, tout en améliorant mes compétences techniques (Java, Swing, JDBC) et organisationnelles (planification, tests, documentation).

1.3 Environnement de travail

Durant mon stage, j'ai travaillé en présentiel au sein des locaux de l'entreprise trois jours par semaine : les lundis, mercredis et vendredis. L'environnement de travail était calme et propice à la concentration, avec un poste de travail individuel équipé de NetBeans, XAMPP et les outils nécessaires au développement.

J'étais encadré par un développeur senior qui m'a orienté sur les choix techniques et les bonnes pratiques à suivre. Les échanges se faisaient de manière régulière, principalement en face à face, et ponctuellement par messagerie électronique. Bien que je travaillais en autonomie sur mon projet, je pouvais solliciter de l'aide en cas de difficulté.

Les tâches m'étaient assignées en début de semaine sous forme d'objectifs à atteindre, et je rédigeais un rapport hebdomadaire à l'attention de mon enseignant référent. Cette organisation m'a permis de développer mon autonomie tout en bénéficiant d'un accompagnement adapté.

Des réunions informelles avaient lieu chaque lundi matin avec le développeur senior afin de valider les priorités de la semaine et obtenir des retours techniques.

1.4 Gestion de projet

a. Planification hebdomadaire

Le projet a été planifié sur une période de **quatre semaines** avec un rythme de **trois jours de travail par semaine (lundi, mercredi, vendredi)**. Une répartition progressive des tâches a été mise en place en fonction de la montée en compétence sur les technologies utilisées.

Semaine	Tâches principales
1	Installation des outils (NetBeans, XAMPP), prise en main de Java Swing, création des premières interfaces (connexion, menu).
2	Connexion avec la base de données MySQL via JDBC, développement des modules CRUD pour les produits.
3	Génération de facture PDF, ajout des interfaces d'administration et de statistiques.
4	Tests complets, corrections de bugs, documentation et finalisation du rapport.

b. Réunions et communication

Tout au long du stage, la communication avec le tuteur en entreprise s'est faite de manière régulière et efficace. Des **réunions informelles** ont été organisées **chaque lundi matin**, en présentiel ou en ligne, afin de :

- Définir les objectifs de la semaine à venir.
- Faire un retour sur les tâches accomplies.
- Identifier les difficultés rencontrées.
- Réorienter les priorités si nécessaire.

Chaque réunion durait environ **30 à 45 minutes** et permettait d'assurer un bon suivi du projet. En complément, des échanges par e-mail étaient effectués en milieu et fin de semaine pour :

- Envoyer les livrables hebdomadaires (interfaces développées, avancement fonctionnel).
- Poser des questions ponctuelles techniques.
- Obtenir des retours rapides sur certains points spécifiques.

Ce mode de communication a permis un encadrement léger mais efficace, tout en me laissant une certaine autonomie dans l'organisation du travail.

Chapitre 2 : Contexte et objectifs du projet

2.1 Problématique / besoin identifié

Avec l'évolution des habitudes de consommation, les services de restauration cherchent à moderniser leurs processus de commande afin de gagner en efficacité et en rapidité. Dans ce contexte, les applications de commande numérique deviennent de plus en plus indispensables, même dans des structures de taille modeste comme les cafétérias, cantines d'entreprise ou petits restaurants.

Le projet développé pendant mon stage répond à ce besoin concret : créer une solution locale, simple à utiliser, qui ne dépend pas d'une connexion Internet permanente. Le choix d'une application desktop (et non web) a été retenu pour permettre une utilisation rapide sur place.

2.2 Objectifs fonctionnels et techniques

L'objectif principal du projet est de concevoir une application intuitive et fonctionnelle permettant :

- La gestion des produits et des catégories d'articles.
- La prise de commandes avec ajout au panier.
- La validation des commandes et l'impression de factures.
- L'accès à une interface d'administration pour les comptes utilisateurs et le suivi des revenus.

D'un point de vue technique, le projet devait être développé en Java avec la bibliothèque Swing pour l'interface graphique, et utiliser MySQL comme système de gestion de base de données. Le développement s'appuie également sur l'IDE NetBeans et l'environnement XAMPP.

2.3 Contraintes et attentes

Le projet devait respecter certaines contraintes telles que :

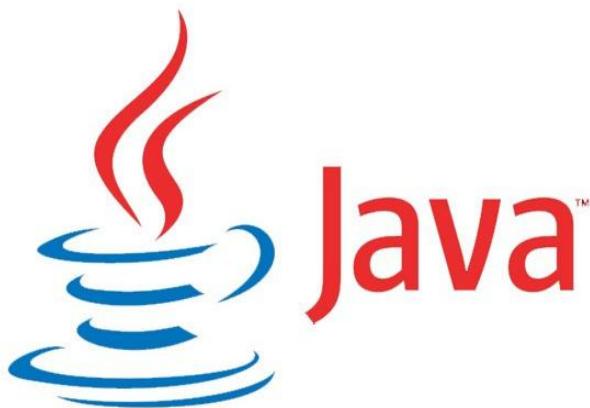
- Un fonctionnement entièrement en local, sans dépendance à Internet.
- Une interface simple et rapide à prendre en main, même pour des utilisateurs non techniques.

- Un système de base de données robuste et facilement administrable via phpMyAdmin.
- Un code source clair, commenté et structuré de manière modulaire.

Les attentes de l'entreprise portaient principalement sur la fiabilité de l'application, la simplicité d'utilisation, et la possibilité d'adapter facilement la solution à d'autres contextes similaires.

Chapitre 3 : Outils et technologies utilisées

3.1 Langage de programmation : Java



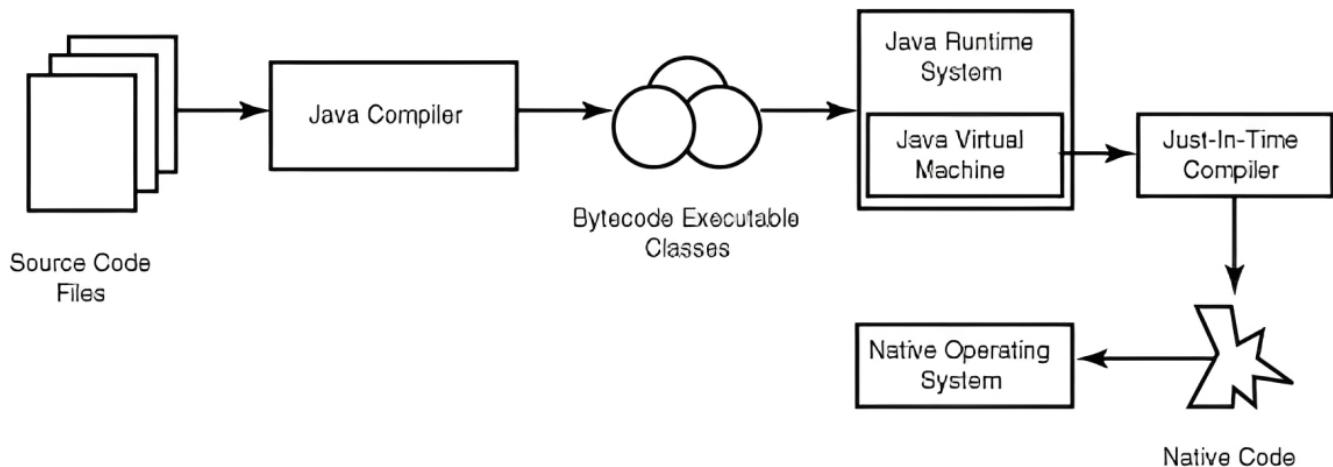
Le langage **Java** a été choisi pour développer l'application en raison de sa portabilité, de sa robustesse et de son large écosystème.

Java est l'un des langages de programmation orientés objet. Il est utilisé dans le développement de logiciels, de sites web, de jeux ou d'applications sur des appareils mobiles.

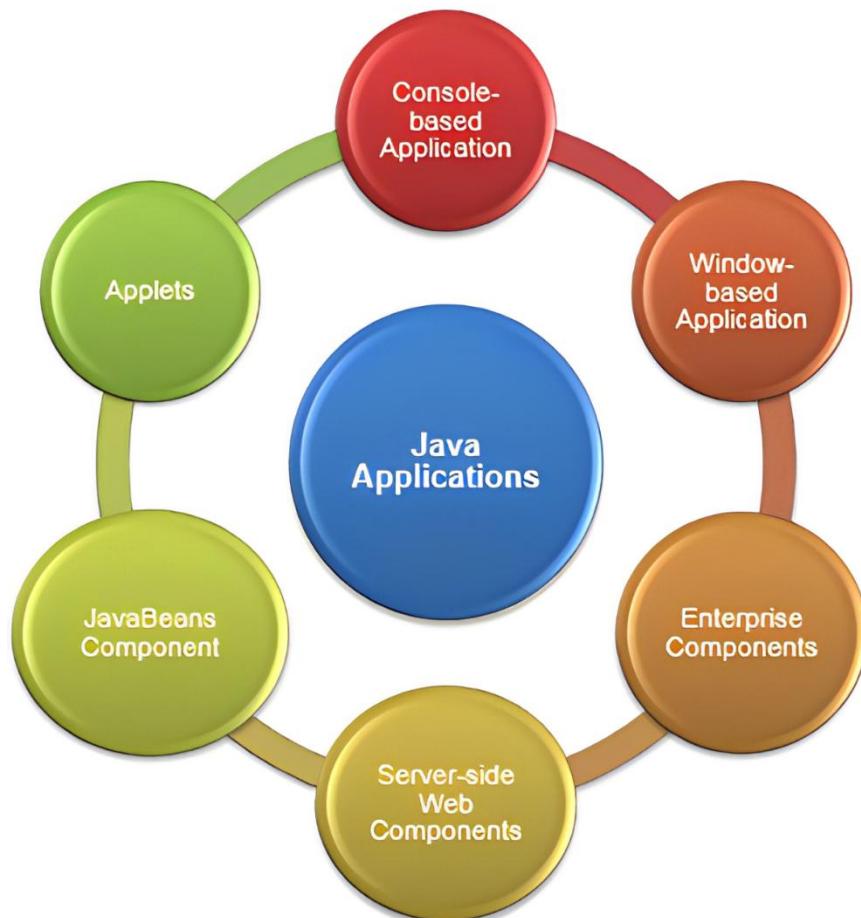
Java a été initié par James Gosling et ses collègues chez Sun Microsystems en 1991. À l'origine, Java a été créé dans le but d'écrire des logiciels pour des produits ménagers et portait le nom d'Oak. Java a été publié en 1994, et en 2010, Oracle a acquis Sun Microsystems.

Java a été conçu avec le principe "Écrire une fois, exécuter partout" (Write Once, Run Anywhere - WORA). Les programmes écrits en Java peuvent s'exécuter sur différentes plateformes grâce à un environnement d'exécution, à condition qu'il existe un environnement d'exécution approprié qui supporte cette plateforme.

Le langage de programmation Java est un langage interprété. Plus précisément, Java est un langage à la fois compilé et interprété. Concrètement, lorsque vous écrivez du code, le système génère un fichier .java. Lors de la compilation, le code source du programme est transformé en bytecode. La machine virtuelle Java (Java Virtual Machine) interprète ce bytecode en code machine (ou code natif) lorsqu'elle reçoit une demande d'exécution du programme.



Les types d'applications développées en utilisant Java :



3.2 Interface graphique : Java Swing

a. Concept

Pour l'interface utilisateur, nous avons utilisé **Java Swing**, une bibliothèque native de Java permettant de créer des interfaces graphiques riches.

Java Swing fait partie des Java Foundation Classes (JFC) conçues pour permettre le développement d'applications Java à grande échelle pour les entreprises. Il est construit sur l'interface de programmation (API) Abstract Windowing Toolkit (AWT) et est écrit en Java. Contrairement à AWT, Java Swing est indépendant de la plateforme et ses composants sont plus légers.

Les JFC sont un ensemble de composants d'interface développés pour les applications Desktop (applications individuelles sur chaque machine).

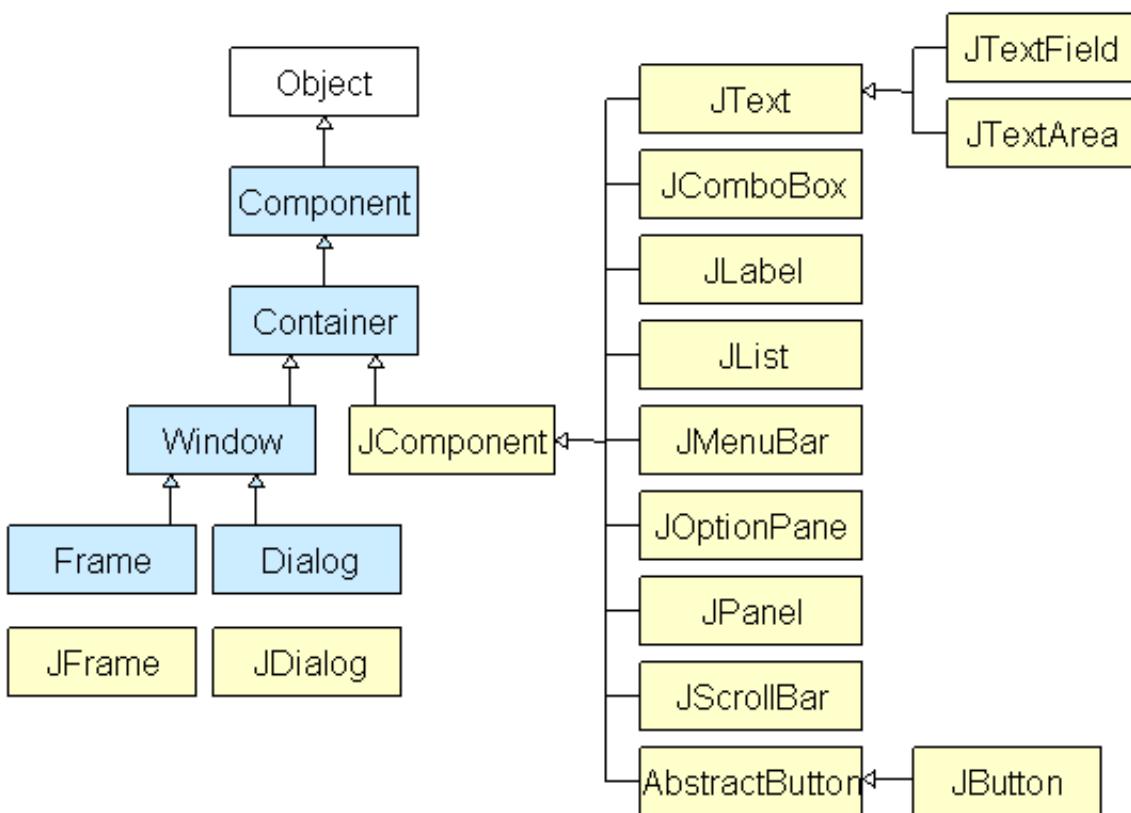
Java Swing est une API fournissant une interface utilisateur graphique (GUI) pour les programmes Java. Java Swing est également appelé la boîte à outils de widgets GUI Java.

Java Swing (ou simplement Swing) a été développé à partir d'API antérieures appelées Abstract Windows Toolkit (AWT). Swing offre des composants GUI plus riches et plus sophistiqués qu'AWT. Ces composants vont d'un simple label à des arbres et des tableaux complexes. Le package javax.swing fournit des classes pour l'API Java Swing, telles que JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser, etc.

Le package javax.swing contient un ensemble de classes et d'interfaces qui étendent et améliorent les composants du package java.awt, permettant de créer des interfaces plus élégantes. Le package javax.swing ne remplace pas java.awt mais est utilisé conjointement avec lui dans la même interface.

Les composants du package javax.swing commencent tous par la lettre J, comme JButton, JFrame, etc. JFrame a une structure multicouche, chaque couche correspondant à un conteneur. Parmi elles, la couche intermédiaire appelée ContentPane est généralement utilisée pour contenir les objets affichés dans l'interface.

b. Diagramme de hiérarchie des classes Java Swing



Les méthodes couramment utilisées de la classe Component :

Méthode	Description
public void add(Component c)	Ajouter un composant à un autre composant.
public void setSize(int width, int height)	Définir la taille d'un composant.
public void setLayout(LayoutManager m)	Définir le gestionnaire de mise en page (layout) pour un composant.
public void setVisible(boolean b)	Définir la visibilité d'un composant. Par défaut, elle est false (cachée).

c. Les différences entre Java AWT et Java Swing

Java AWT	Java Swing
Les composants AWT sont dépendants de la plateforme.	Les composants Java Swing sont indépendants de la plateforme.
Les composants AWT sont lourds.	Les composants Swing sont légers.
AWT ne prend pas en charge le pluggable L&F.	Swing prend en charge le pluggable L&F.
AWT fournit moins de composants que Swing.	Swing fournit des composants plus puissants tels que table, list, scrollpanes, colorchooser, tabbedpane...

AWT ne suit pas le MVC (Modèle-Vue-Contrôleur), où le modèle représente les données, la vue représente la présentation et le contrôleur agit comme une interface entre le modèle et la vue.	Swing suit le MVC.
---	--------------------

d. Fonctionnalités de Java Swing

- **Légereté** - Les composants Swing sont indépendants de l'API du système d'exploitation natif, car les contrôles de l'API Swing sont principalement rendus en utilisant du code Java pur plutôt que des appels système de base.
- **Contrôles riches** - Swing fournit un ensemble de contrôles avancés riches tels que Tree, TabbedPane, des curseurs, un sélecteur de couleur et des contrôles de tableau.
- **Haute personnalisation** - les contrôles Swing peuvent être personnalisés de manière très simple et indépendamment de la représentation interne.
- **Apparence et sensation pluggables** - Swing repose sur l'apparence de l'application GUI et peut être modifié à l'exécution, en fonction des valeurs disponibles.

3.3 Base de données : MySQL

Toutes les données de l'application (comptes utilisateurs, produits, commandes, revenus) sont stockées dans une base **MySQL** locale. C'est un SGBD relationnel open-source largement utilisé et compatible avec Java via l'API JDBC.

La structure de la base de données a été conçue pour être simple mais efficace, avec des relations entre les tables product, bill, user, etc.

Structure | SQL | Search | Query | Export | Import | Operations | Privileges | Routines | Events | Triggers | More

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
bill	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
category	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
detailsbill	Browse Structure Search Insert Empty Drop	11	InnoDB	utf8mb4_general_ci	32.0 KiB	-
product	Browse Structure Search Insert Empty Drop	16	InnoDB	utf8mb4_general_ci	32.0 KiB	-
user	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
5 tables Sum		36	InnoDB	utf8mb4_general_ci	112.0 KiB	0 B

Check all With selected:

[Print](#) [Data dictionary](#)

[Create new table](#)

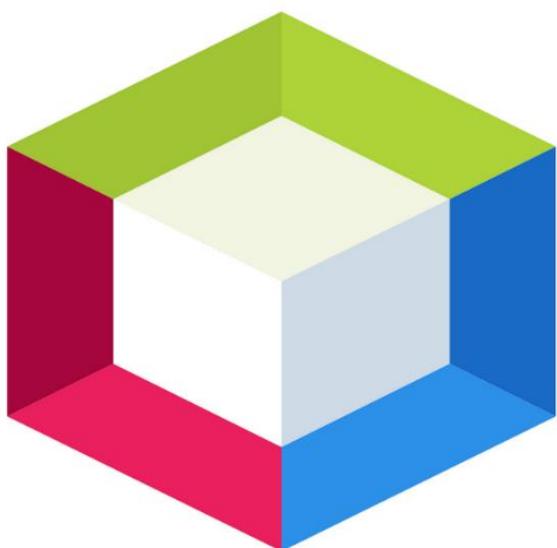
Table name Number of columns

Console

Figure 1 : Structure de la base de données dans phpMyAdmin

3.4 Environnement de développement : NetBeans, XAMPP

a. NetBeans :



Apache NetBeans

NetBeans est un environnement de développement intégré (IDE) open-source, principalement utilisé pour le développement en Java. Il offre une interface graphique conviviale, des outils de débogage performants, et une excellente gestion des projets modulaires.

Dans le cadre de ce projet, NetBeans a été utilisé pour :

- Organiser le projet Java avec une structure claire en packages (`model`, `view`, `controller`, etc.).
- Concevoir les interfaces Swing grâce à son éditeur graphique intégré (GUI Builder), facilitant la mise en page et la gestion des événements.
- Gérer les bibliothèques et dépendances nécessaires au projet (JDBC, etc.).
- Compiler, exécuter et tester rapidement l'application grâce aux raccourcis et au terminal intégré.
- Déboguer les erreurs de manière efficace avec les points d'arrêt et l'inspection de variables en temps réel.

Grâce à ces fonctionnalités, NetBeans a considérablement accéléré le processus de développement et a contribué à la clarté du code source.

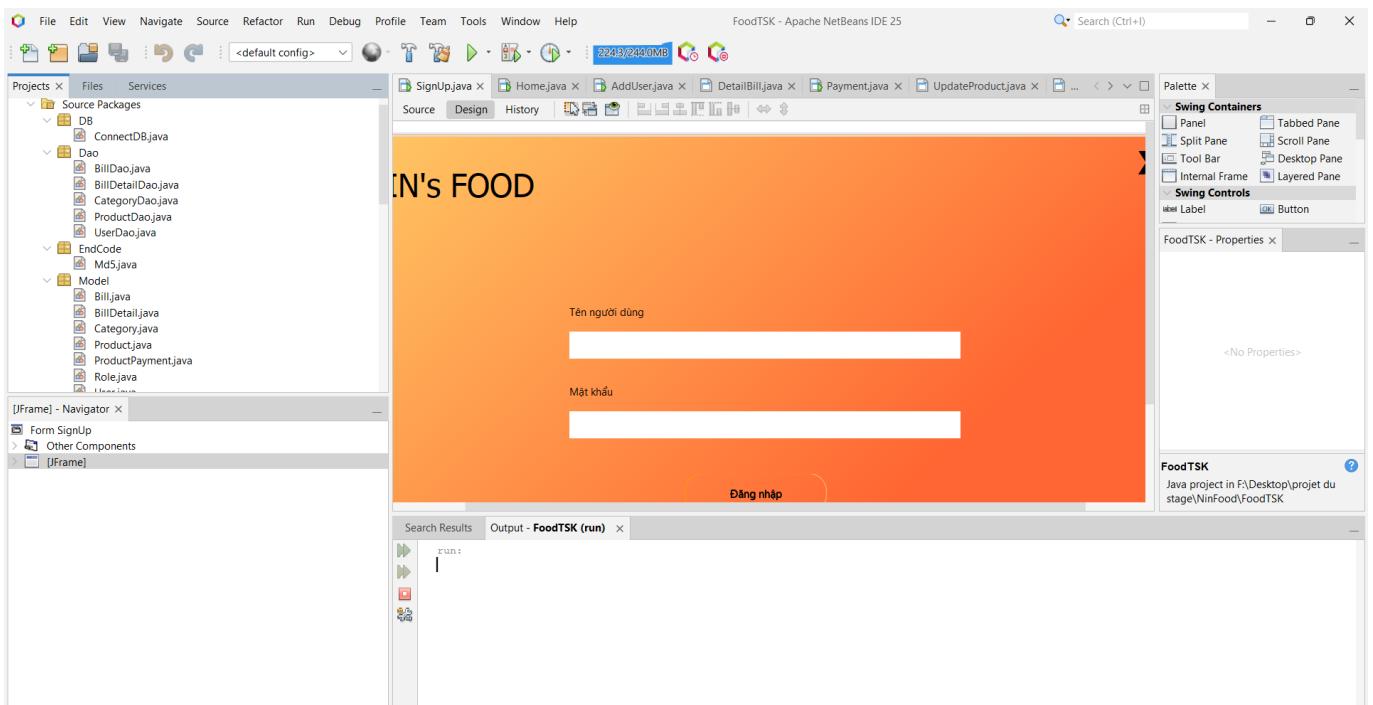


Figure 2 : Vue du projet dans l'IDE NetBeans

b. XAMPP

XAMPP est un environnement de serveur local tout-en-un, regroupant Apache, MySQL, PHP et Perl. Il est largement utilisé pour simuler un serveur web sur une machine locale sans avoir besoin d'une configuration complexe.

Dans le cadre de ce projet, XAMPP a été utilisé principalement pour :

- Héberger le serveur **MySQL** localement, permettant à l'application Java d'y accéder via JDBC.
- Administrer la base de données via **phpMyAdmin**, accessible depuis l'interface web de XAMPP.
- Surveiller et redémarrer rapidement les services (Apache, MySQL) grâce à son **Control Panel** très intuitif.

Son usage a grandement simplifié le développement et les tests en local, tout en assurant un bon niveau de stabilité de la base de données.

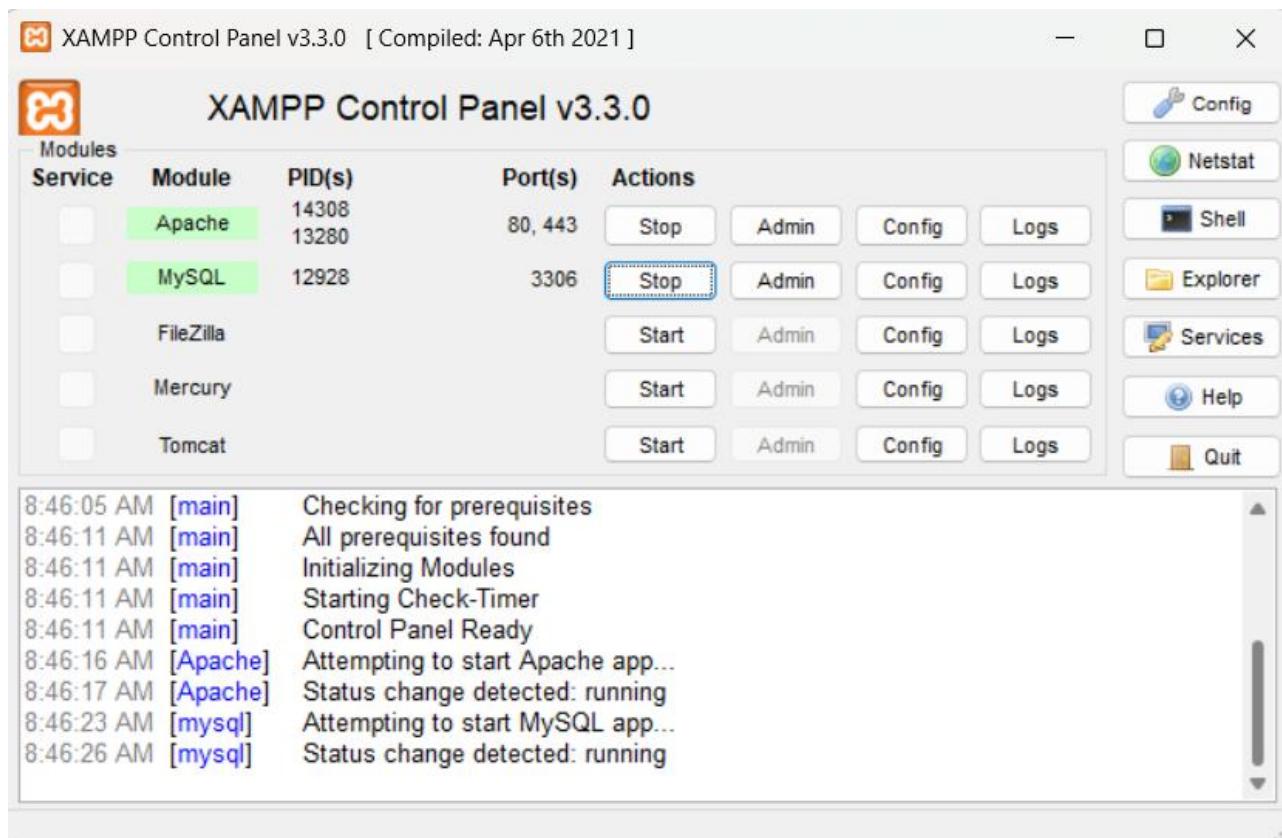


Figure 3 : Tableau de bord XAMPP montrant les services Apache et MySQL en cours d'exécution

3.5 Autres outils

D'autres outils ont été utilisés pour faciliter le travail quotidien :

- **phpMyAdmin** : pour visualiser et gérer la base MySQL via interface web.
- **Google Chrome** : pour tester l'aperçu des impressions (factures au format PDF).
- **Explorateur de fichiers Windows** : gestion des ressources comme les images ou les fichiers de configuration.
- **Outils de capture d'écran** : pour la documentation du rapport.

Chapitre 4 : Conception et architecture de l'application

4.1 Analyse fonctionnelle

L'application développée permet de gérer l'ensemble du processus de commande dans un contexte de restauration rapide. Elle propose différentes fonctionnalités selon le type d'utilisateur (administrateur ou employé) :

- Authentification sécurisée.
- Affichage des produits par catégorie.
- Ajout d'articles au panier, calcul automatique du total.
- Paiement, génération et impression de facture.
- Accès à un tableau de bord administratif : produits, utilisateurs, revenus.
- Gestion CRUD sur les produits.
- Visualisation des statistiques journalières/mensuelles.

4.2 Diagrammes UML

Plusieurs diagrammes UML ont été réalisés afin de modéliser l'architecture du projet :

- Diagramme de cas d'utilisation : identifie les interactions entre les acteurs (admin, employé) et le système.
- Diagramme de classes : montre les différentes classes Java utilisées, leurs attributs et méthodes.
- Diagramme de séquence (optionnel) : pour décrire le déroulement d'une commande.

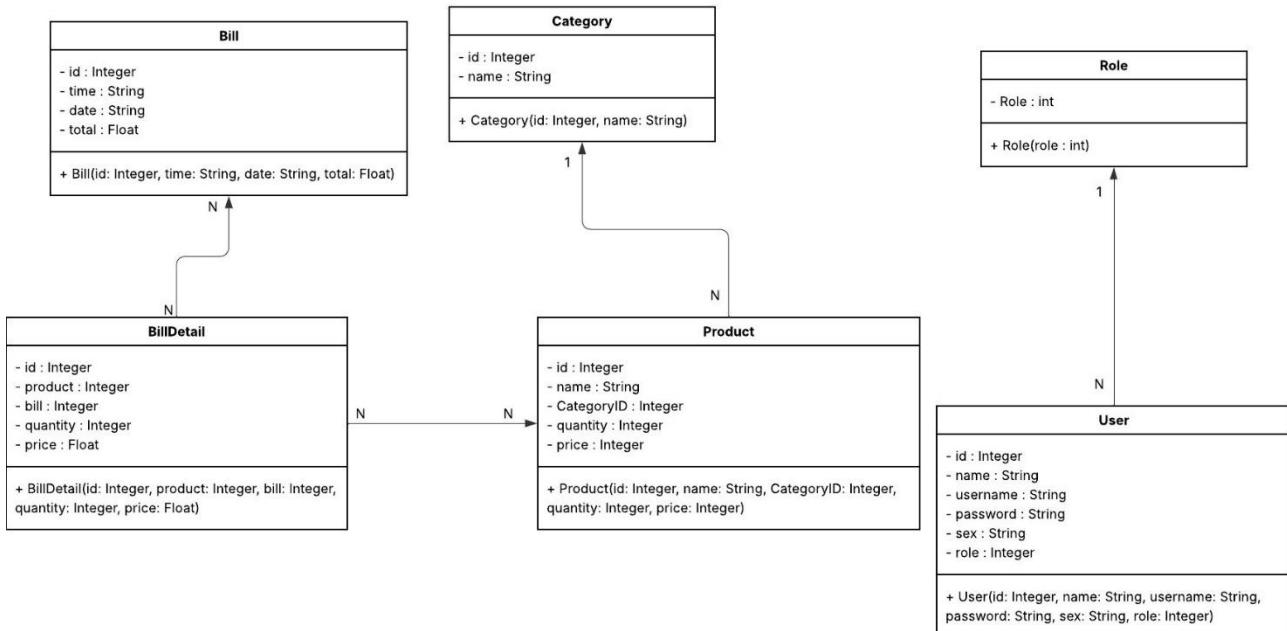


Figure 4 : Diagramme de classes

4.3 Structure technique du projet

Le projet est structuré de manière modulaire en utilisant les packages Java. Chaque package correspond à un rôle spécifique :

- **Dao/** : Contient des classes Data Access Object responsables des opérations de base de données liées à des entités telles que Bill, Product, User, etc.
- **DB/** : Contient des classes de connexion à la base de données et des classes utilitaires.
- **EndCode/** : Contient des classes utilitaires, éventuellement pour l'encodage ou le chiffrement (par exemple, Md5).
- **event/** : Contient des classes de gestion d'événements pour les interactions de l'interface utilisateur.
- **foodnin/** : Contient les classes principales de l'application et les formulaires de l'interface utilisateur (classes Java et fichiers de formulaire).
- **img/** : Contient des ressources d'image utilisées dans l'application.
- **lib/** : Contient des fichiers JAR de bibliothèques externes utilisés par le projet.
- **Model/** : Contient des classes de modèle représentant les entités de données telles que Bill, Product, User, Categorie, etc.

Cette architecture MVC simplifiée permet de séparer clairement la logique métier, la présentation et les données.

4.4 Schéma de base de données

La base de données comporte plusieurs tables principales : `product`, `user`, `bill`, `detailsbill`, `category`. Elles sont liées par des clés primaires et étrangères. La conception vise à éviter les redondances et assurer l'intégrité des données.

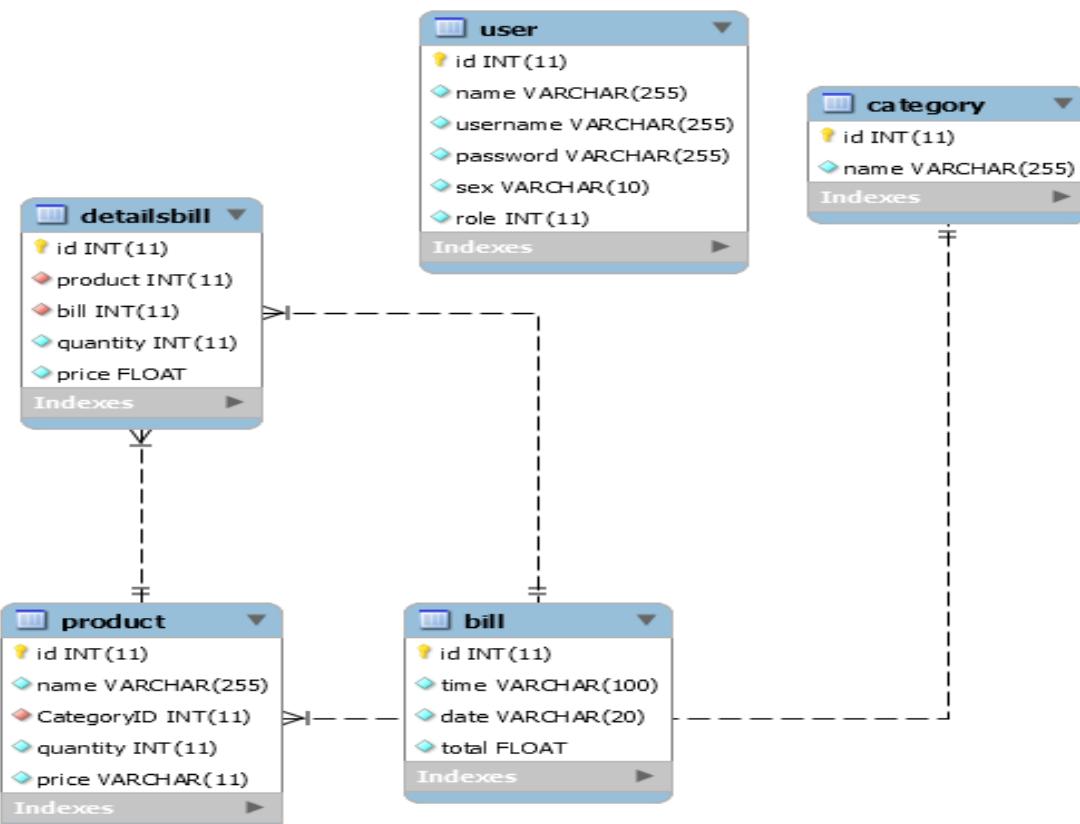


Figure 5 : Schéma de base de données

Chapitre 5 : Développement et mise en œuvre

5.1 Description des interfaces principales

L'application comporte plusieurs interfaces graphiques conçues avec Java Swing, correspondant aux différentes fonctionnalités destinées aux utilisateurs (employé ou administrateur). Voici les interfaces principales :

- ◆ Interface de connexion : C'est l'écran d'accueil de l'application. Il permet à l'utilisateur de s'authentifier à l'aide d'un nom d'utilisateur et d'un mot de passe. En fonction du rôle (admin ou employé), l'accès aux fonctionnalités est différencié. Cette interface comprend des champs de saisie, un bouton "Connexion" et des messages d'erreur en cas d'identifiants incorrects.
- ◆ Interface d'accueil : Après la connexion, l'utilisateur est redirigé vers une interface centrale qui affiche les produits disponibles à la commande, organisés par catégorie. L'utilisateur peut ajouter des articles au panier, ajuster les quantités, voir le total et passer la commande. C'est l'interface principale pour les employés.
- ◆ Interface de gestion des produits : Réservée aux administrateurs, cette interface permet d'effectuer les opérations CRUD (Créer, Lire, Modifier, Supprimer) sur les produits. Elle affiche une table dynamique des produits et fournit des champs pour ajouter de nouveaux éléments ou modifier des existants. Les données sont synchronisées avec la base MySQL.
- ◆ Interface d'administration : Cette interface donne un accès global aux fonctionnalités réservées à l'administrateur, telles que : la gestion des comptes utilisateurs, la consultation de l'historique des commandes, et l'accès aux outils de supervision. Elle centralise toutes les actions liées à l'administration de l'application.
- ◆ Interface de chiffre d'affaires : Accessible uniquement à l'administrateur, cette interface affiche les statistiques de ventes journalières et mensuelles sous forme de tableau ou graphique (le cas échéant). Elle permet de filtrer les données par date et d'exporter les résultats si besoin. Elle offre une vue synthétique de la performance commerciale.

5.2 Fonctions réalisées

Les principales fonctions développées au cours du stage sont les suivantes :

- Authentification des utilisateurs avec vérification des identifiants.
- Affichage dynamique des produits depuis la base de données.
- Ajout, suppression, mise à jour de produits par l'administrateur (CRUD).
- Gestion du panier (calcul automatique des totaux, retrait d'article).
- Paiement et génération automatique d'un reçu imprimable.
- Suivi des ventes avec filtres journaliers et mensuels dans l'interface admin.

5.3 Difficultés rencontrées et solutions apportées

Plusieurs difficultés techniques ont été rencontrées durant le développement :

- Problèmes de connexion à la base de données MySQL : résolus en ajustant les paramètres JDBC et le port XAMPP.
- Encodage des caractères accentués : corrigé en définissant l'encodage UTF-8 dans le code Java et dans phpMyAdmin.
- Gestion des événements dans les interfaces Swing : nécessité de structurer le code avec des classes de contrôle distinctes.
- Impression PDF du reçu : implémentée avec une solution de rendu graphique de JPanel.

5.4 Tests réalisés et résultats

Des tests manuels ont été effectués à chaque étape du développement pour valider les fonctionnalités principales :

- Test de l'authentification avec différents rôles (admin, employé).
- Test de cohérence des données après modification dans l'interface admin.
- Vérification du calcul des totaux et du rendu des factures.
- Test de robustesse en cas d'entrée invalide (champs vides, données incohérentes).

Les résultats ont été satisfaisants, avec un comportement stable de l'application dans l'ensemble des cas testés.

5.5 Résultats expérimentaux

a. Interface de connexion

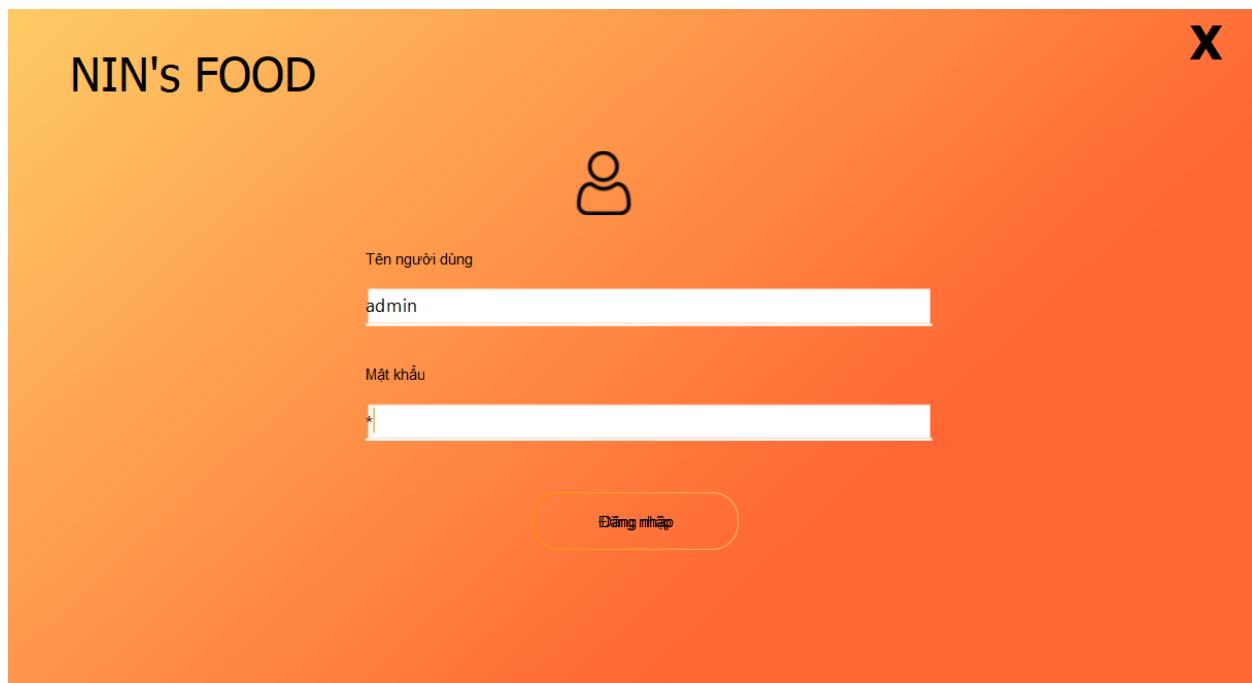


Figure 6 : Interface de connexion

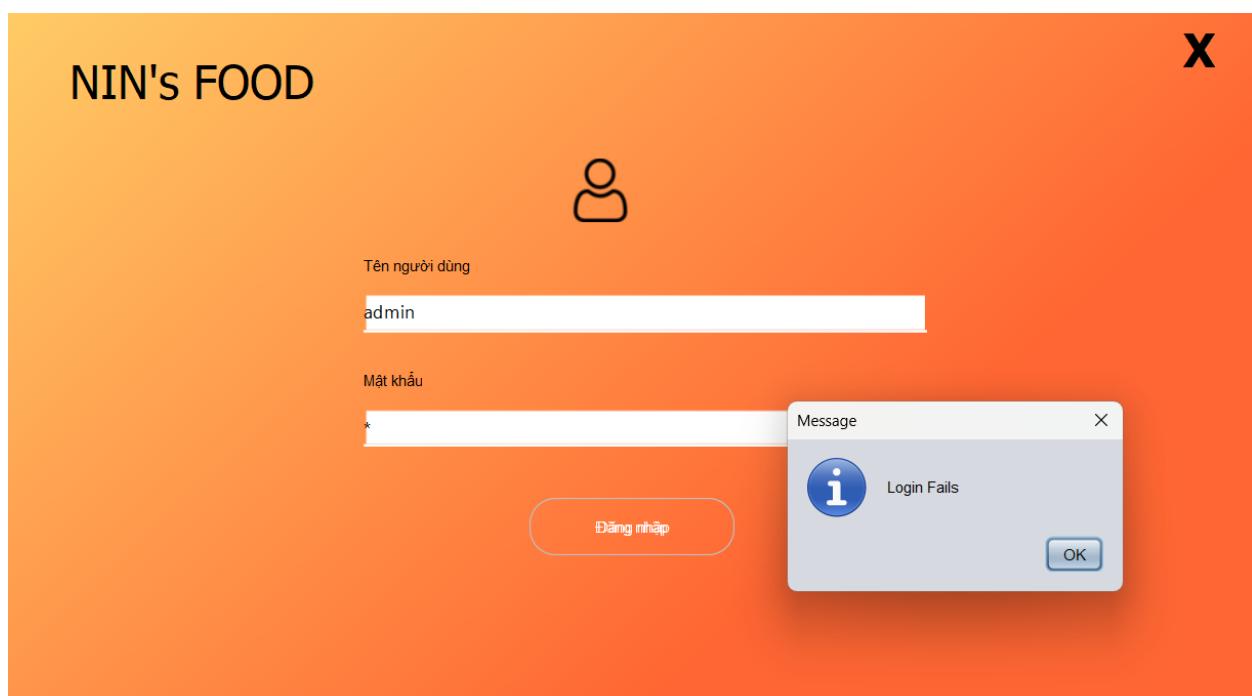


Figure 7 : Interface en cas d'échec de la connexion.

b. Interface d'accueil



Figure 8 : Interface de la page d'accueil



Figure 9 : Barre de sélection des aliments ou des boissons



Figure 10 : Après avoir sélectionné quelques articles

ID	Name	Quantity	Price
1	Gỏi Đu Đủ	1	50000
3	Hột Vịt Lộn	1	20000
4	Bánh Cuộn	1	30000
6	Bún Bò	1	100000
7	Hồ Lô	1	15000

Tổng tiền : 215,000 VND
Đã nhận : Tính toán
Tiền thừa : 0 VND
Quay lại In ra

Figure 11 : Interface de paiement

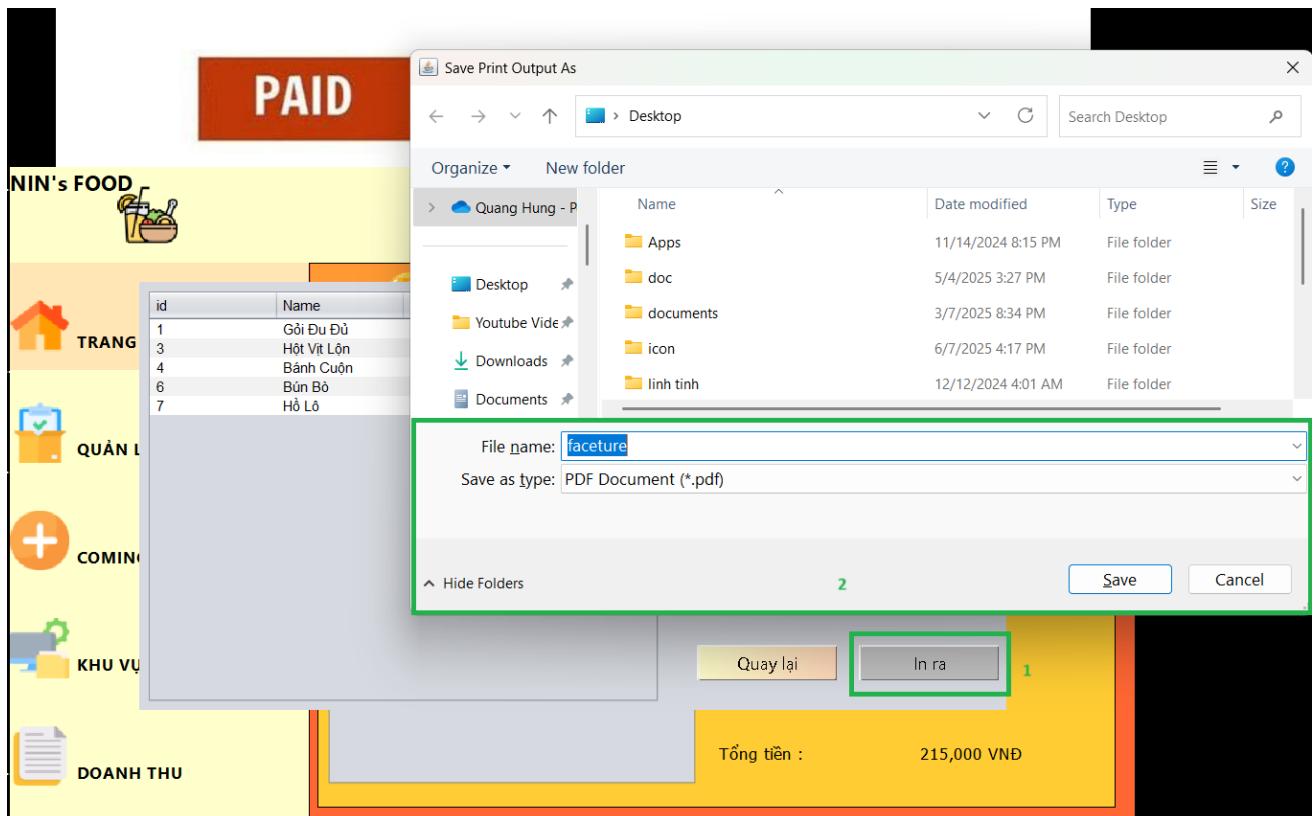


Figure 12 : Interface d'impression

 Nin's Food

 No 00000 Address Line One

 Address Line 02 SRI LANKA

www.facebook.com/NinFood

Item Name	Price
Gỏi Đu Đủ	50000
1 * 50000.0	50000
Hột Việt Lộn	20000
1 * 20000.0	20000
Bánh Cuộn	30000
1 * 30000.0	30000
Bún Bò	100000
1 * 100000.0	100000
Hồ Lô	15000
1 * 15000.0	15000

 Total amount: 215,000 VNĐ

 Recieved : 500,000 VNĐ

 Refund : 285,000 VNĐ

THANK YOU COME AGAIN

SOFTWARE BY: Nin
 CONTACT: lequangh82@gmail.com

Figure 13 : Image d'impression de facture

ID	Name	Quantity	Price
1	Gỏi Đu Đủ	1	50000
3	Hột Vịt Lộn	1	20000
4	Bánh Cuộn	1	30000
6	Bún Bò	1	100000
7	Hồ Lô	1	15000

Tổng tiền : 215,000 VNĐ

ID	Name	Quantity	Price
1	Gỏi Đu Đủ	1	50000
3	Hột Vịt Lộn	1	20000
4	Bánh Cuộn	1	30000
6	Bún Bò	1	100000

Tổng tiền : 200,000 VNĐ

Figure 14 : Après avoir cliqué sur le bouton pour supprimer un article

c. Interface de gestion des produits :

ID	Name	Category	Quantity	Price
1	Gỏi Đu Đủ	Food	10	50000
2	Cháo Gà	Food	32	50000
3	Hột Vịt Lộn	Food	35	20000
4	Bánh Cuộn	Food	69	30000
5	Bánh Tráng Cuốn	Food	50	30000
6	Bún Bò	Food	27	100000
7	Hồ Lô	Food	200	15000
8	Bánh Cuốn	Food	32	20000
9	Trà Sữa	Drink	200	10000
10	Trà Chanh	Drink	50	20000
11	Trà Đào	Drink	40	15000
16	Cháo Vịt	Food	50	69000
20	Nước Dừa	Drink	60	25000
21	Cà Phê Sài Gòn	Drink	24	20000
22	Nước Gạo	Drink	500	30000
23	Sữa Chua	Drink	46	40000

Figure 15 : Interface de gestion des produits

The screenshot shows the main interface of the NIN's FOOD application. On the left, there is a sidebar with icons and labels: TRANG CHỦ, QUẢN LÝ SẢN PHẨM, COMING SOON, KHU VỰC QUẢN TRỊ, and DOANH THU. The main area has a title 'NIN's FOOD' at the top left, a small icon of a shopping cart with hearts at the top right, and a 'Đăng xuất' (Logout) button. Below the title is a navigation bar with buttons for ADD, EDIT, DELETE, REFRESH, and FIND. A dropdown menu for 'Category' is open, showing options: All (selected), Food, and Drink. The main content area is a table listing 23 products:

id	name	Category	quantity	price
1	Gỏi Đu Đủ	Food	32	50000
2	Cháo Gà	Food	35	50000
3	Hột Vịt Lộn	Food	69	20000
4	Bánh Cuộn	Food	50	30000
5	Bánh Tráng Cuốn	Food	27	30000
6	Bún Bò	Food	200	100000
7	Hồ Lô	Food	32	15000
8	Bánh Cuốn	Food	200	20000
9	Trà Sữa	Drink	50	10000
10	Trà Chanh	Drink	40	20000
11	Trà Đào	Drink	60	15000
16	Cháo Vịt	Food	24	69000
20	Nước Dừa	Drink	500	25000
21	Cà Phê Sài Gòn	Drink	500	20000
22	Nước Gạo	Drink	46	30000
23	Sữa Chua	Drink	46	40000

Figure 16 : Interface lorsque **All** est sélectionné

This screenshot shows the same application interface as Figure 16, but with a different selection in the category dropdown. The 'Food' option is now selected, as indicated by a green box around the dropdown menu. The main content area displays a table of products filtered to show only food items:

id	name	Category	quantity	price
1	Gỏi Đu Đủ	Food	10	50000
2	Cháo Gà	Food	32	50000
3	Hột Vịt Lộn	Food	35	20000
4	Bánh Cuộn	Food	69	30000
5	Bánh Tráng Cuốn	Food	50	30000
6	Bún Bò	Food	27	100000
7	Hồ Lô	Food	200	15000
8	Bánh Cuốn	Food	32	20000
16	Cháo Vịt	Food	50	69000

Figure 17 : Interface lorsque **Food** est sélectionné

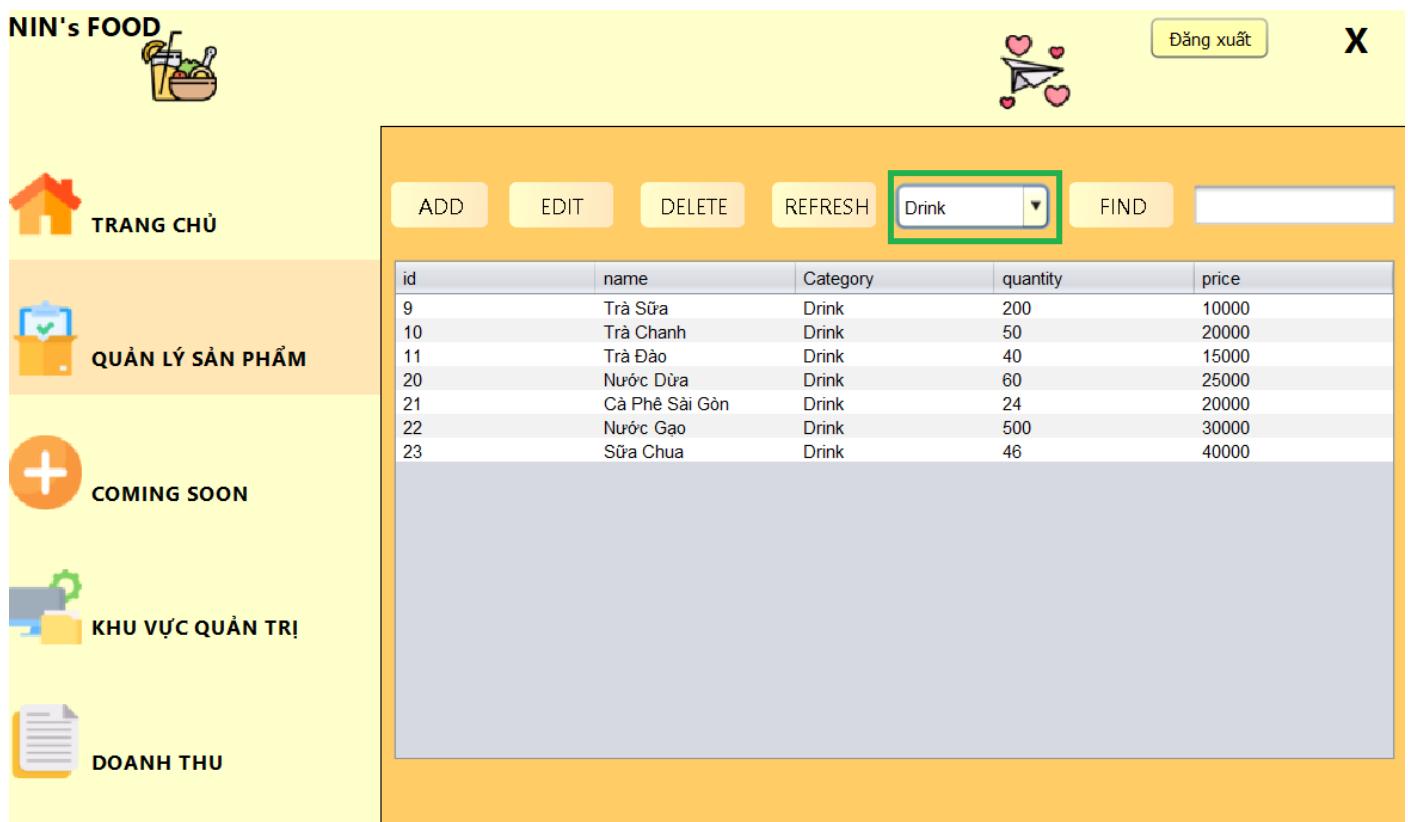


Figure 18 : Interface lorsque **Drink** est sélectionné

THÊM SẢN PHẨM

Tên	Danh mục
<input type="text"/>	<input checked="" type="radio"/> Đồ ăn <input type="radio"/> Đồ uống
Số lượng	Giá
<input type="text"/>	<input type="text"/>
Thêm Quay lại	

Figure 19 : Interface lorsqu'on clique sur le bouton **Add**



Figure 20 : Interface lorsqu'on sélectionne une article et puis clique sur le bouton **Edit**

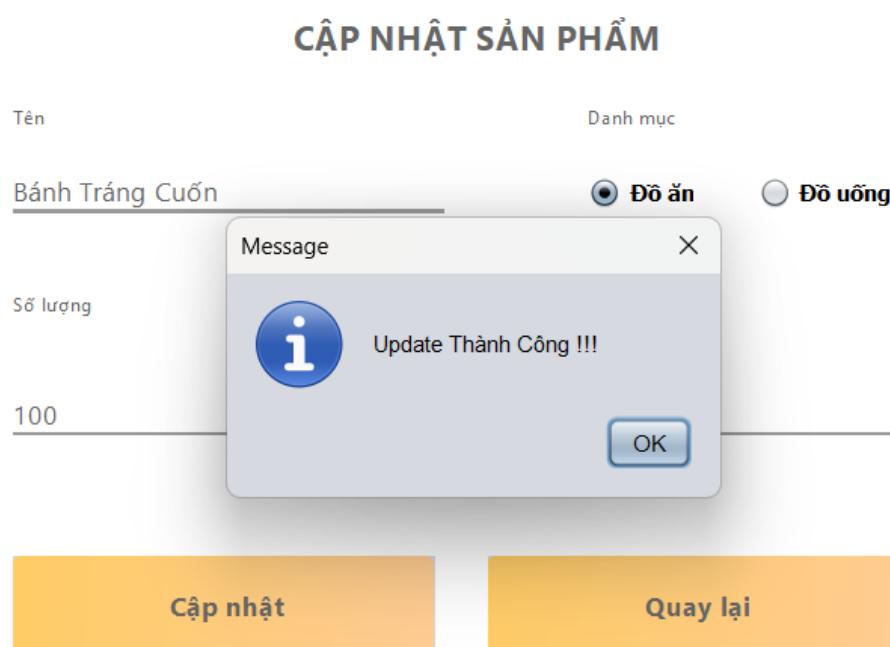


Figure 21 : Interface de modification réussie

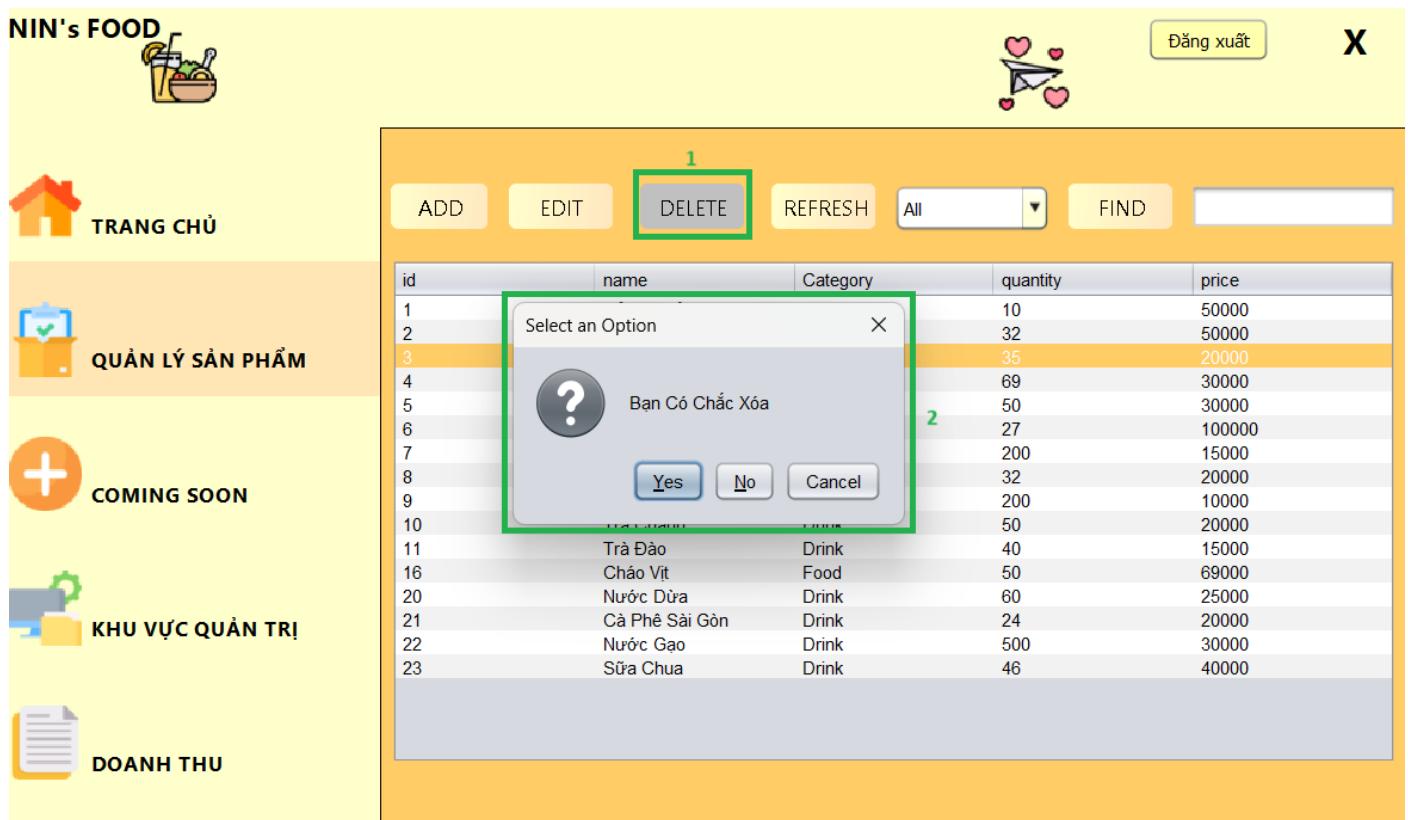


Figure 22 : Interface lorsqu'on clique sur le bouton **Delete**

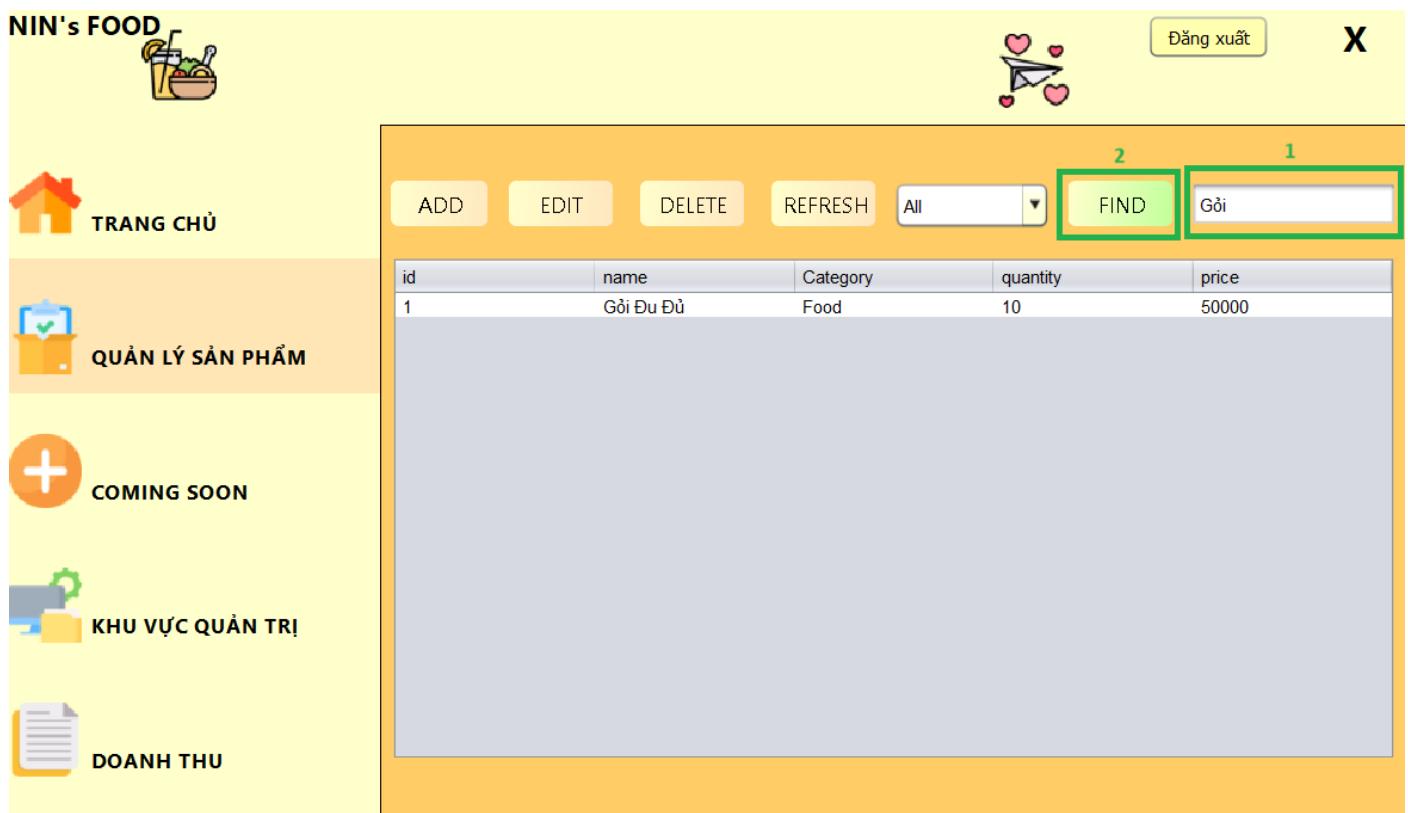


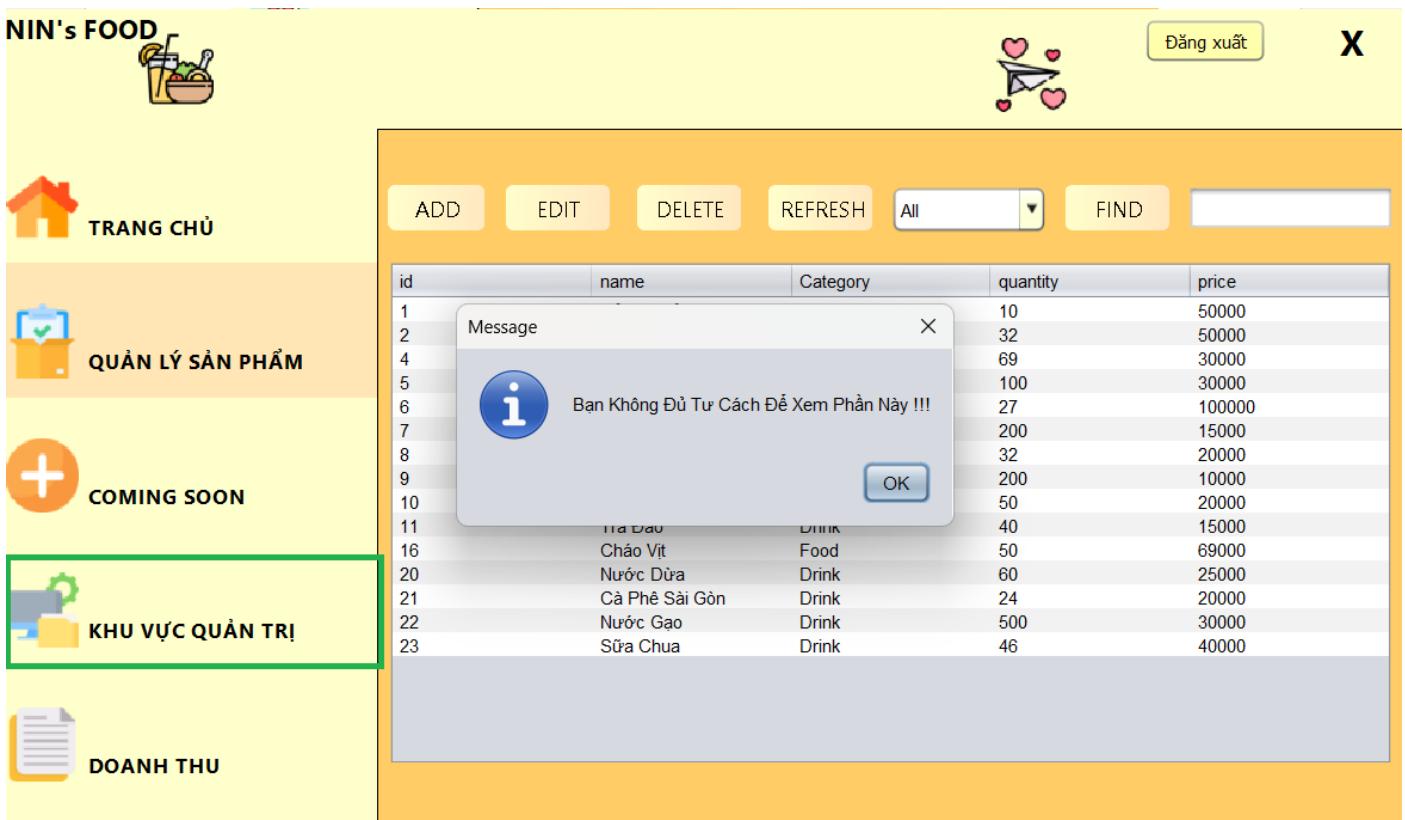
Figure 23 : Interface lorsqu'on clique sur le bouton **Find**



The screenshot shows the NIN's FOOD application interface. On the left sidebar, there are five menu items: TRANG CHỦ (Home), QUẢN LÝ SẢN PHẨM (Product Management), COMING SOON, KHU VỰC QUẢN TRỊ (Administration Area), and DOANH THU (Revenue). The main content area displays a table of products with columns: id, name, Category, quantity, and price. The 'REFRESH' button at the top of the table is highlighted with a green box. A decorative banner with hearts and a shopping cart is visible at the top right.

id	name	Category	quantity	price
1	Gỏi Đu Đủ	Food	10	50000
2	Cháo Gà	Food	32	50000
4	Bánh Cuộn	Food	69	30000
5	Bánh Tráng Cuốn	Food	100	30000
6	Bún Bò	Food	27	100000
7	Hồ Lô	Food	200	15000
8	Bánh Cuốn	Food	32	20000
9	Trà Sữa	Drink	200	10000
10	Trà Chanh	Drink	50	20000
11	Trà Đào	Drink	40	15000
16	Cháo Vịt	Food	50	69000
20	Nước Dừa	Drink	60	25000
21	Cà Phê Sài Gòn	Drink	24	20000
22	Nước Gạo	Drink	500	30000
23	Sữa Chua	Drink	46	40000

Figure 24 : Interface lorsqu'on clique sur le bouton Refresh

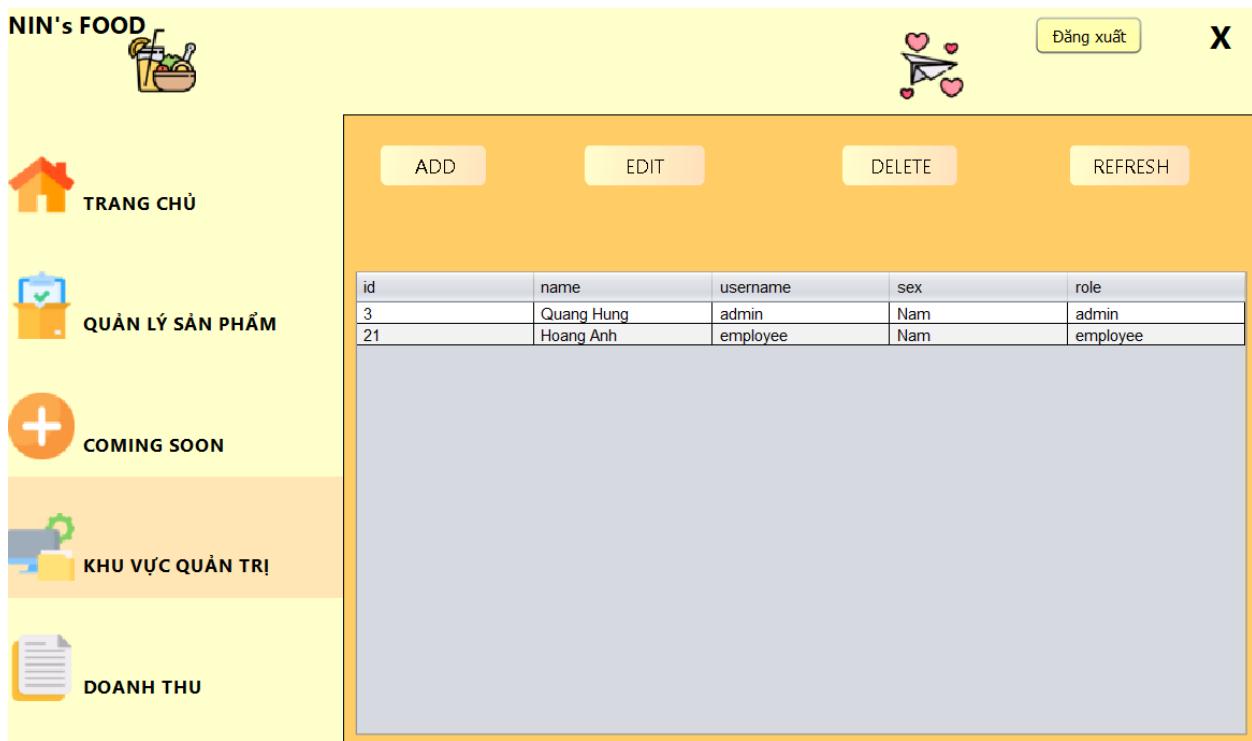


The screenshot shows the NIN's FOOD application interface. The 'KHU VỰC QUẢN TRỊ' (Administration Area) menu item on the left sidebar is highlighted with a green box. The main content area displays a table of products. A modal dialog box is overlaid on the table, containing an information icon, the message 'Bạn Không Đủ Tư Cách Để Xem Phần Này !!!', and an 'OK' button. The rest of the table data is visible below the dialog.

id	name	Category	quantity	price
1	Message		10	50000
2			32	50000
4			69	30000
5			100	30000
6			27	100000
7			200	15000
8			32	20000
9			200	10000
10			50	20000
11	Trà Đào	Drink	40	15000
16	Cháo Vịt	Food	50	69000
20	Nước Dừa	Drink	60	25000
21	Cà Phê Sài Gòn	Drink	24	20000
22	Nước Gạo	Drink	500	30000
23	Sữa Chua	Drink	46	40000

Figure 25 : Interface lorsque l'employé se connecte et tente d'accéder à la section Administration (notification de refus d'accès)

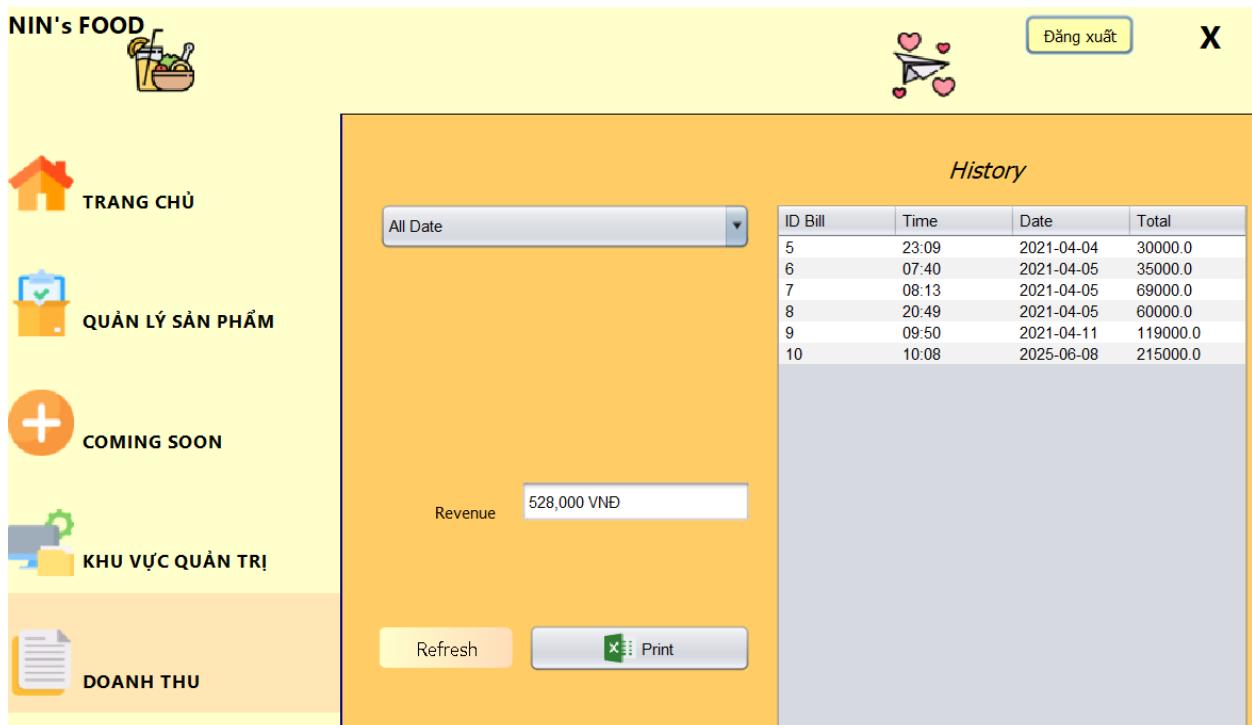
d. Interface d'administration



id	name	username	sex	role
3	Quang Hung	admin	Nam	admin
21	Hoang Anh	employee	Nam	employee

Figure 26 : Interface d'administration de l'Admin

e. Interface de chiffre d'affaires



ID Bill	Time	Date	Total
5	23:09	2021-04-04	30000.0
6	07:40	2021-04-05	35000.0
7	08:13	2021-04-05	69000.0
8	20:49	2021-04-05	60000.0
9	09:50	2021-04-11	119000.0
10	10:08	2025-06-08	215000.0

Figure 27 : Interface de chiffre d'affaires

Chapitre 6 : Résultats et évaluation

6.1 Fonctionnalités implémentées

L'application développée couvre l'ensemble des fonctionnalités prévues au départ :

- Construction réussie avec les fonctionnalités de base nécessaires à un logiciel.
- Gestion des droits de connexion réussie : Admin, Employé.
- L'employé n'a pas accès à l'administration réservée à l'admin.
- Calcul du chiffre d'affaires (Revenue) par jour et par mois réussi.
- Paiement (Payment) des factures réussi, avec émission de facture.

6.2 Fonctionnalités non réalisées

La majorité des fonctionnalités prévues ont été réalisées. Cependant, quelques améliorations auraient pu être envisagées :

- Les fonctionnalités du logiciel sont encore limitées, certaines d'entre elles présentent des erreurs.
- La sécurité n'est pas encore élevée, certains composants nécessitent un temps de traitement long en raison d'algorithmes non optimisés.

6.3 Perspectives d'évolution

Compte tenu du potentiel de l'application et de ses possibilités d'utilisation dans des environnements réels, plusieurs pistes d'amélioration peuvent être envisagées :

- Automatisation de la génération des codes produits, afin d'éviter la saisie manuelle.
- Amélioration des performances pour le traitement de volumes de données plus importants.
- Renforcement de la sécurité des accès et du chiffrement des données.
- Intégration d'un mécanisme de mise à jour automatique du logiciel via Internet.
- Développement d'une version mobile ou multiplateforme.

Ces évolutions contribueraient à rendre l'application plus robuste, plus facile à maintenir et mieux adaptée à un usage professionnel à grande échelle.

6.4 Retours éventuels de l'entreprise

L'entreprise a exprimé une grande satisfaction concernant le travail réalisé. L'application a répondu aux attentes et pourrait servir de base à une version commerciale future. Le tuteur a apprécié l'autonomie, la qualité du code produit ainsi que la régularité dans les livrables hebdomadaires.

Conclusion

Ce stage m'a permis de mettre en pratique les compétences acquises durant ma formation universitaire. J'ai pu participer à un projet concret, gérer toutes les étapes de développement d'une application logicielle, de la conception à la mise en œuvre.

J'ai renforcé mes compétences techniques en Java, Swing, et en base de données, mais aussi mes compétences en communication, en autonomie et en organisation. Je me sens aujourd'hui mieux préparé à intégrer un environnement professionnel.

Dans l'avenir, ce projet pourrait être amélioré en y intégrant une version mobile, une gestion multi-utilisateurs en temps réel, ou encore une synchronisation cloud pour les données.

Glossaire

CRUD : Acronyme de Create, Read, Update, Delete – opérations de base sur les données.

JDBC : Java Database Connectivity – API Java permettant de se connecter à une base de données.

GUI : Graphical User Interface – interface utilisateur graphique.

Swing : Bibliothèque Java utilisée pour créer des interfaces graphiques.

XAMPP : Serveur local regroupant Apache, MySQL, PHP et Perl.

IDE : Integrated Development Environment – logiciel pour écrire et gérer du code source (ex. : NetBeans).

Bibliographie / Webographie

- Documentation officielle Java : <https://docs.oracle.com/javase/>
- Tutoriels Swing : <https://docs.oracle.com/javase/tutorial/uiswing/>
- Documentation MySQL : <https://dev.mysql.com/doc/>
- XAMPP : <https://www.apachefriends.org/>
- phpMyAdmin : <https://www.phpmyadmin.net/>
- Stack Overflow : <https://stackoverflow.com/>

Résumé

Dans le cadre de mon stage de fin d'études en Licence Informatique, réalisé du 19 mai au 19 juin 2025 au sein de l'entreprise IT&M TECHNOLOGY SOLUTIONS JOINT STOCK COMPANY, j'ai développé une application de commande de nourriture en Java, en utilisant la bibliothèque graphique Java Swing et une base de données MySQL.

Ce projet avait pour objectif de créer une application de bureau intuitive permettant aux utilisateurs de passer des commandes, de gérer les produits, d'imprimer les factures et de consulter les revenus. Le développement s'est déroulé avec l'IDE NetBeans et l'environnement XAMPP pour la gestion de la base de données. L'interface graphique a été conçue de manière conviviale, avec une séparation claire des rôles entre administrateur et employé.

Ce stage m'a permis de renforcer mes compétences en programmation Java, en conception d'interfaces utilisateur, ainsi qu'en gestion de bases de données relationnelles. Il m'a également permis de mieux comprendre le cycle de vie d'un projet logiciel, depuis l'analyse jusqu'à la mise en œuvre.

Abstract

As part of my final-year internship for the Bachelor's degree in Computer Science, carried out from May 19 to June 19, 2025 at IT&M TECHNOLOGY SOLUTIONS JOINT STOCK COMPANY, I developed a food ordering desktop application using Java, with Java Swing for the graphical interface and MySQL as the database system.

The goal of this project was to create an intuitive desktop app allowing users to place orders, manage products, print invoices, and view revenue statistics. The application was developed using NetBeans IDE and XAMPP for database management. The interface was designed to be user-friendly, with a clear separation of roles between administrator and employee.

This internship helped me strengthen my skills in Java programming, GUI design, and relational database management. It also gave me a better understanding of the software project lifecycle, from initial analysis to implementation.