

République Algérienne Démocratique et Populaire

MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE

SCIENTIFIQUE



UNIVERSITE ABOU BEKR BELKAID DE TLEMEN
FACULTE DE TECHNOLOGIE
DEPARTEMENT DE GENIE ELECTRIQUE ET ELECTRONIQUE



MEMOIRE

Présenté pour l'obtention du diplôme

Master en Génie Industriel

Intitulé :

Conception et réalisation d'une application
Pour la gestion d'un restaurant

Réalisé par :

Mr. ALIAT BILLEL

Mr. KHELIFI YAHIA

Soutenu le 17 Juin 2017 devant le Jury :

M ^{elle} HOUBAD Yamina	MAA	Présidente
Mr. BENSMAINE Abderrahmane	MCB	Examineur
Mr. MALIKI Fouad	MAA	Examineur
Mr. BESSENOUCI Hakim Nadhir	MAA	Encadrant

Année Universitaire : 2016- 2017

Remerciement

Nous tenons à remercier en tout premier lieu ALLAH, le tout puissant de nous donner la patience, la force et le courage pour élaborer ce travail.

Au-delà d'un encadrant qui a su révéler le meilleur de chacun de nous, Monsieur Bessenouci Hakim Nadhir, a su nous diriger tout au long de notre parcours, comme il a pu optimiser les différents obstacles que nous avons rencontrés. Notre enseignant, pour que nous devions tout le respect, nous le remercions infiniment.

Nous tenons à exprimer nos remerciements à Melle Houbad Yamina chargé de cours à l'Université de Tlemcen, d'avoir accepté de présider le jury de ce mémoire.

Un grand merci à Mr. Bensmaine Abderrahmane chargé de cours à l'Université de Tlemcen, pour l'honneur d'avoir accepté d'examiner notre travail.

Un grand merci à Mr. Maliki Fouad maitre-assistant à l'université de Tlemcen pour l'honneur d'avoir accepté d'examiner notre travail.

Un vif remerciement à nos enseignants qui ont contribué à notre formation durant tout notre cursus.

Merci

DEDICACES

Je dédie ce modeste travail à:

Mes parents :

Grâce à leurs tendres encouragements et leurs grands sacrifices, ils ont pu

créer le climat affectueux et propice à la poursuite de mes études.

Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes

profonds sentiments envers eux.

Je prie le bon Dieu de les bénir, en espérant qu'ils seront toujours fiers de moi.

A ma très chère sœur, son mari.

Je vous dédie ce travail avec tous mes vœux de bonheur, de santé et de réussite.

A mes très chers frères.

A Ibrahim, Ismaïl, Moustapha ;

Je vous souhaite un avenir plein de joie, de bonheur, de réussite et de sérénité.

A mes oncles : Saïd, Ahmed, Khaïre, Youcef, Omar, Djilali, Mourad, Omar et mes tantes pour leur encouragement

A mes chers ami (s) et mes chers collègues.

Je ne peux trouver les mots pour exprimer mon affection et mes pensées, vous

êtes pour moi des frères, sœurs et des amis

En témoignage de l'amitié qui nous unit et des souvenirs de tous les moments que nous avons passé ensemble, je vous dédie

ce travail et je vous souhaite une vie pleine de santé et de

bonheur.

« YAHIA »

DEDICACES

A mes très chers parents

Aucune dédicace ne serait exprimé assez profondément ce que je ressens envers vous.

Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous avez fait pour moi. Je ferai de mon mieux pour rester un sujet de fierté à

vos yeux avec l'espoir de ne jamais vous décevoir. Que ce modeste travail, soit l'exaucement de vos vœux tant formulés et de vos prières quotidiennes.

Je vous aime.

A mes très chères sœurs et frère

Vous occupez une place particulière dans mon cœur. Je vous dédie ce travail en vous souhaitant un avenir radieux, plein de bonheur et de succès. Que dieu, le tout puissant, vous préserve et vous procure santé et

longue vie.

A mes très chères amies

Je ne peux trouver les mots pour exprimer mon affection et mes pensées, vous

êtes pour moi des frères, sœurs et des amis

En témoignage de l'amitié qui nous unit et des souvenirs de tous les moments que nous avons passé ensemble, je vous dédie ce travail et je vous souhaite une vie pleine de santé et de bonheur

« BILLET »

« ملخص »

لا يزال قطاع المطاعم يواجه التحسن يوما بعد يوم. و هذا يفرض على العديد من المطاعم استغلال أقصى قدراتها وتحسين أدائها على جميع المستويات لتقديم افضل خدمة و بأفضل إدارة. في هذا المشروع قمنا بإنشاء تطبيق لإدارة المطاعم، و لهذا قمنا بانتهاج مختلف مراحل تطوير البرمجيات، ألا وهي متطلبات التحليل والتصميم والتنفيذ. تم تنفيذ التطبيق وفق الشكلية MERISE و طور في نظام التطوير المدمج Delphi. **كلمات البحث:** إدارة المطاعم، MERISE، Delphi، MCD.

« Résumé »

Le secteur de restauration confronte toujours une amélioration jour par jour. Cela oblige nombreux restaurants à exploiter le maximum leurs capacités et améliorer les performances à tous les niveaux afin de fournir un bon service et une bonne gestion.

Dans ce projet nous avons développé une application de gestion de restaurant, pour cela nous avons suivi les différentes phases de développement d'un logiciel, à savoir l'analyse des besoins, la conception et l'implémentation.

L'application a été modélisée selon le formalisme MERISE et développé dans l'environnement de développement intégré Delphi.

Mots clés : gestion de restaurant, MERISE, Delphi.

« Abstract »

The catering sector is always improving day by day. This requires many restaurants to leverage their capabilities and improve performance at all levels to provide good service and management.

In this project we have developed a restaurant management application, for which we have followed the different stages of software development, namely needs analysis, design and implementation.

The application was modeled according to the MERISE formalism and developed in the Delphi integrated development environment.

Key words: restaurant management, MCD, MERISE, Delphi.

Le Table de matière

INTRODUCTION GENERALE.....	12
INTRODUCTION GENERALE.....	13
CHAPITRE 1 : ETUDE ET CRITIQUE DE	14
L'EXISTANT	14
I INTRODUCTION	15
II ÉTUDE DE L'EXISTANT	15
II.1 Définition d'un Restaurant	15
II.2 Organigramme de restaurant	16
II.3 Les différentes formules de restauration	16
II.3.1 La restauration traditionnelle	16
II.3.2 La restauration de collectivité.....	17
II.3.3 La restauration rapide	17
II.4 Les acteurs du système de restauration	17
II.5 Les fonctions	18
II.6 Critique de l'existant	19
II.7 Solution proposée	19
II.8 Spécification des besoins	20
II.8.1 Les besoins fonctionnels	20
II.8.2 Les besoins non fonctionnels.....	20
III CONCLUSION.....	21

CHAPITRE 2: CONCEPTION	22
I INTRODUCTION	23
II CHOIX DE L'OUTIL DE CONCEPTION	23
II.1 Définition de MERISE	23
II.2 Définition d'UML	23
II.3 L'étude comparative entre MERISE et UML	24
II.3.1 Niveaux d'abstraction	24
II.3.2 Approche fonctionnelle.....	24
II.3.3 Dualité des données –traitements	24
II.4 La méthode de notre travail	25
III L'APPLICATION DE LA METHODE MERIS	25
III.1 Le niveau conceptuel	25
III.2 Le niveau logique (organisationnel)	26
III.3 Le niveau physique	26
III.4 Modèle Conceptuel de Données (MCD)	26
III.5 Modèle Conceptuel de Données du restaurant	28
III.5.1 La partie de l'employé :	29
III.5.2 La partie de client :	29
III.5.3 La préparation du plat :	30
III.5.4 Les classifications des plats :	31
III.5.5 Les commandes :	32
III.5.6 La gestion de stock :	32
IV CONCLUSION	35

CHAPITRE 3: REALISATION	36
I INTRODUCTION :.....	37
II MATERIELS ET LANGAGES DE PROGRAMMATION	37
II.1 Matériel	37
II.2 Les langages de programmation	37
II.2.1 Présentation de l'architecture d'un système (client/serveur).....	37
II.2.2 Les différents modèles de client-serveur :	39
II.2.3 Les différentes architectures :	39
II.2.4 Caractéristiques des systèmes client-serveur :.....	41
II.2.5 Avantages de l'architecture client/serveur:.....	42
II.2.6 Inconvénients du modèle client/serveur :	42
III SYSTEME DE GESTION DE BASE DE DONNEES	42
III.1 La base de données	42
III.2 Définition du système de gestion de base de données	42
III.3 Les niveaux de la SGBD	43
III.4 Quelques SGBD connus et utilisés	43
III.5 SQL Server	44
III.5.1 Fonctionnement du SQL Server	44
III.5.2 La création de la base de données	45
Exemple	45
III.5.3 Représentation des requêtes SQL	46
IV DELPHI	47
IV.1 Les Avantages de Delphi	47
V CONNEXION AUX BASES DE DONNEES	48
V.1 Les composants utilisés sur notre projet :	48
V.2 Le code Delphi de connexion	48

VI	LES INTERFACES DE L'APPLICATION	48
VII	CONCLUSION.....	62
	CONCLUSION GENERALE.....	63
	REFERENCES BIBLIOGRAPHIQUE.....	65
	LISTE DES FIGURES.....	10
	LISTE DES TABLEAUX.....	11

Liste des figures

Figure 1-1: Organigramme de restaurant	16
Figure 2-1: La partie de l'employé	29
Figure 2-2: la réservation et la facturation de client	30
Figure 2-3: La préparation du plat	31
Figure 2-4: Les classifications des plats	31
Figure 2-5: La commande du client	32
Figure 2-6: La gestion de stock.....	33
Figure 3-1: l'architecture d'un système client/serveur	38
Figure 3-2: La table de réservation	45
Figure 3-3 : La table de commande de fournisseur.....	43
Figure 3-4: les composants Delphi pour la connexion.....	48
Figure 3-5: l'interface d'inscription.....	49
Figure 3-6: l'interface de connexion.....	50
Figure 3-7: l'interface d'entrée	51
Figure 3-8: l'interface de la modification de mot passe	51
Figure 3-9: l'interface d'accueil.....	52
Figure 3-10: L'interface de gestion de stock	53
Figure 3-11: la liste de livraisons.....	54
Figure 3-12: Interface qui permet d'écrire un ordre d'achat.....	54
Figure 3-13: exemple de rapport de stock.....	55
Figure 3-14: L'interface de gestion des clients.....	55
Figure 3-15: L'interface de gestion de personnelle	56
Figure 3-16: L'interface de gestion des plats.....	57
Figure 3-17: L'interface de gestion de la base de données.....	58
Figure 3-18: L'interface qui permet l'ajout un employé.....	58
Figure 3-19: L'interface qui permet de modifié les informations d'un employé	59
Figure 3-20: ajouter une matière	59
Figure 3-21: ajouter un client.....	60
Figure 3-22: ajouter un poste	60
Figure 3-23: ajouter une catégorie	61
Figure 3-24: ajouter une table	61
Figure 3-25: ajouter un fournisseur.....	62

Liste des tableaux

Tableau 1-1 : Les acteurs du système de restauration.....	17
Tableau 3-1: les caractéristiques	37



Introduction générale

Introduction générale

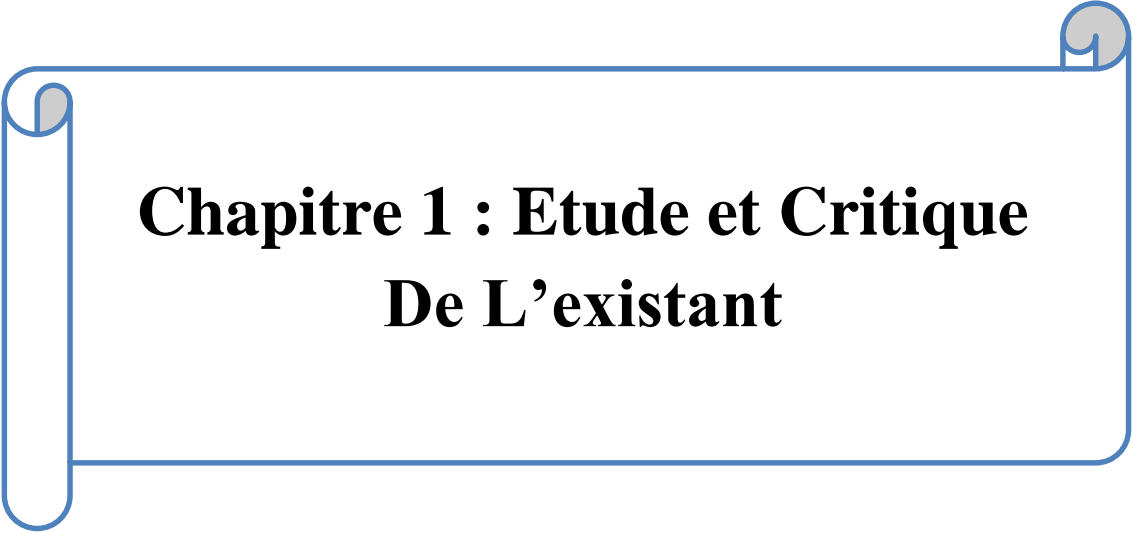
Le monde informatique est très vaste, son apparition correspond aux changements de culture. Jour après jour l'informatique occupe une place intéressante dans les sociétés et les organisations (une entreprise, restaurant) à tous les niveaux en effet l'informatique permet de traiter, mémoriser, et de diffuser l'information dans des délais plus courts et avec des méthodes plus efficaces.

Donc Le système informatique introduit actuellement est capable de gérer et de résoudre tous les problèmes rencontrés dans les entreprises et les organisations, nous nous sommes intéressés sur le secteur de la restauration .

Notre travail est de concevoir et de réaliser une application de gestion de restaurant. Pour ce faire, nous avons fait plusieurs visites dans différents restaurants pour voir le déroulement des différentes tâches et nous permettre de mettre en place une application qui soit bénéfique dans leur gestion, pour éviter au maximum les différents problèmes liés à la gestion traditionnelle (rupture de stock, commande non satisfaite, produits périmés en stock...).

Ce mémoire est constitué de trois chapitres :

- le premier chapitre nous donne une idée de la gestion de restaurant pour lequel nous effectuons notre travail.
- le deuxième chapitre traite les étapes fondamentales pour la modélisation de notre système de gestion de restaurant.
- le troisième chapitre va définir les outils que nous allons utiliser et la réalisation de l'application qui se termine par une série de tests d'exécutions de cette dernière.



**Chapitre 1 : Etude et Critique
De L'existant**

I Introduction

La concurrence entre les restaurants a augmenté d'une façon énorme. Chacun veut offrir à ses clients un service idéal pour garantir leur retour. Afin d'améliorer les services de ce domaine, on a constaté que l'utilisation des systèmes informatisés est nécessaire pour mieux organiser les processus de travail, ainsi que pour faciliter la gestion des restaurants.

Dans ce chapitre nous décrivons et énumérons les différentes fonctions et missions qui existent au sein des restaurants que nous avons visité; ensuite nous présenterons les besoins fonctionnels et non fonctionnels attendus de l'application afin d'atteindre l'objectif de notre projet de fin d'étude.

II Étude de l'existant

L'étude de l'existant permet de déterminer les points faibles et les points forts du système de gestion actuel pour pouvoir apporter des améliorations lors de la conception et la réalisation de l'application. Pour cela notre première tâche a été de visiter différents établissements de restauration et de rencontrer des personnels pour avoir une idée claire et précise sur l'existant quel qu'il soit. Après quoi nous avons réellement débuté le travail en menant différentes recherches. Cette méthodologie de travail nous a permis d'avoir une connaissance large de l'existant.

II.1 Définition d'un Restaurant

Établissement commercial où l'on sert des repas contre paiement Dans

Un restaurant, les repas sont préparés par une équipe de cuisinier.

Il existe plusieurs types de cuisines. Les restaurants sont parfois le dispositif réservé au service des repas au sein d'une plus grande entité (hôtel, université, aéroport...), on parle alors de restaurant collectif par opposition au site de cuisine. Ils peuvent aussi être associés à une activité de traiteur. Le restaurant offre des conditions de confort plus ou moins importantes, et la restauration est dite « rapide » quand le client peut commander et manger en quelques minutes ou dizaines de minutes, éventuellement debout. Un restaurant est un commerce ou un magasin. [1]

II.2 Organigramme de restaurant

La restauration est un secteur qui comporte de nombreux métiers, et il est parfois difficile pour qui n'est pas familier avec ce secteur de les hiérarchiser. Il existe différents organigrammes selon la taille de l'établissement : petit restaurant, grand restaurant ou restaurant gastronomique. Dans notre projet nous nous intéressons aux petits restaurants représentés suivant :

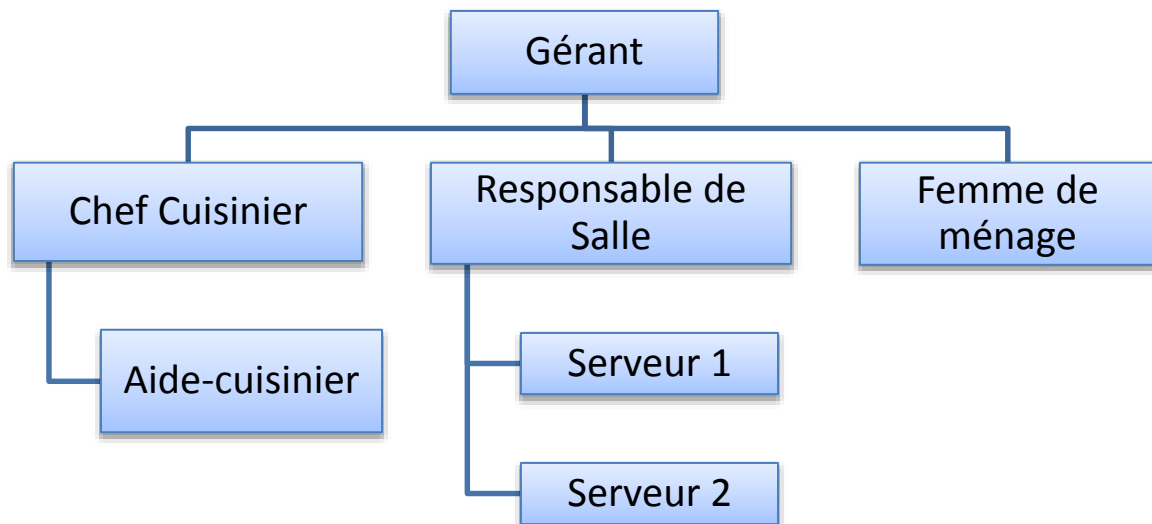


Figure 1-1: Organigramme de restaurant

II.3 Les différentes formules de restauration

II.3.1 La restauration traditionnelle

Cette branche comporte des établissements très variés et se compose essentiellement de petites entreprises individuelles. C'est le type d'établissement où chaque chef d'entreprise peut laisser libre cours à sa créativité et à son imagination. Les restaurants classiques

- Les restaurants d'hôtels
- La restauration à thèmes
- Autour d'un produit
- Autour d'un pays. [1]

II.3.2 La restauration de collectivité

La restauration de collectivité repose sur des contraintes précises : nourrir un grand nombre de gens pour un prix extrêmement bas. Comme la réussite de cette activité repose en grande partie sur le principe des économies d'échelle,

- La restauration d'entreprise
- La restauration scolaire
- La restauration Hospitalière. [1]

II.3.3 La restauration rapide

La restauration rapide est le secteur qui a connu la plus forte croissance ces vingt dernières années. En effet, c'est ce concept qui répond en partie à nos besoins de citadins modernes.

- Le fast-food
- Le restauroute
- La cafétéria. [1]

II.4 Les acteurs du système de restauration

Un acteur est une personne extérieure au système en cours de modélisation. L'acteur peut consulter ou modifier l'état du système.

Dans le tableau ci-dessous, nous identifions les différents acteurs de notre système.

Les acteurs	Les rôles
<u>Gérant</u>	Gérer les clients Consulter les statistiques Gérer les personnels Gérer le stock Gérer les réservations
<u>Serveur</u>	Gérer les commandes Consulter les produits disponibles Gérer les réservations Client
<u>Client</u>	Consulter les produits disponibles Passer une commande Réserver table
<u>Cuisinier</u>	Changer état commande

Tableau II 1-1 : Les acteurs du système de restauration

II.5 Les fonctions

Dans chaque établissement de restauration existent une multitude de tâches pour le bon fonctionnement du restaurant; ci-dessous nous présentons les tâches de gestion principales.

La gestion des commandes : Les étapes de passage d'une commande débutent par notification sur papier des produits commandés par le client. Ensuite, le serveur la saisie dans le terminal du système. Enfin, il en informe oralement le cuisinier. Notre mission consiste à automatiser le passage de la commande de telle sorte qu'on annule le contact direct entre les personnels.

La gestion de stock: Le stock est géré d'une façon manuelle en introduisant la quantité de stock en entrée ainsi que la quantité en sortie par jour. Donc notre mission consiste à informatiser la gestion de stock de tel sorte que chaque mise à jour du stock doit être effectué d'une façon automatique aussi qu'on attribue à chaque article stocké un seuil minimum de tel façon qu'une ligne commande sera ajoutée automatiquement dans la liste des commandes fournisseur pour le produit concerné.

Il existe trois catégories de produits :

- **Produit simple :** produits intégrés au stock et pouvant être vendus immédiatement ou utilisés pour la préparation d'autres recettes.
- **recette :** produit composé de plusieurs autres produits (cocktails, recette de cuisine). On peut ainsi gérer les fiches techniques de manière détaillée. Le stock des produits constituant la recette est automatiquement impacté lors de chaque vente.
- **produit rapide :** produit qui est intégré dans le stock de la même façon que les recettes et pouvant être créé très rapidement (ex : plat du jour).

La réservation des tables : c'est le fait de la prise des réservations des clients et optimiser le taux d'occupation du restaurant grâce à une vue planning. Donc il s'agit du contrôle de chaque réservation afin d'éviter les chevauchements pour un horaire et une table. Actuellement, cette fonction n'est pas informatisée elle est traité d'une façon manuelle donc on risque de perdre des informations.

La gestion du personnel : La gestion des personnels nécessite trois fonctionnalités :

- L'enregistrement des informations personnelles pour chaque embauché.
- Le contrôle des absences et des heures supplémentaires des personnels.
- Calcul de salaire pour chaque personnel en prenant en considération les absences et les heures supplémentaires.

La consultation des statistiques qui concerne :

- La consultation de la recette journalière.
- La recette de chaque serveur.
- La quantité consommée.
- La quantité restante de chaque produit en stock.
- La quantité sortante de chaque produit fini.

II.6 Critique de l'existant

Après analyse de l'existant nous avons pu faire sortir les critiques suivantes:

- la répartition du travail par jour, pour les employés : elle est difficile à gérer pour le manager ;
- la prise en charge et le placement du client dans les différentes zones de l'établissement par les salariés ;
- l'optimisation de la prise de la commande : actuellement le serveur doit prendre commande auprès des clients, puis la transférer au caissier qui édite la note et qui l'envoie en cuisine . la note fait le chemin inverse jusqu'au client.
- la gestion des réservations : pour les grandes occasions (anniversaire, mariage, etc.) les clients ne peuvent actuellement pas faire de réservation.
- l'étude des statistiques de vente, qui est inexistante à l'heure actuelle.

II.7 Solution proposée

Après l'étude que nous avons effectuée, et selon les besoin des restaurateurs nous opterons pour une application avec laquelle nous pourrons éliminer le problème de gestion

Afin d'y remédier à tous ses problèmes, nous avons assigné à notre étude les objectifs suivants :

- Automatiser les taches traitées manuellement.
- Rapidité et gains de temps dans la recherche des informations.
- Gestion et organisation des informations.
- Fiabilité du système.
- Sécurité des informations en les stockant sur des supports informatiques.

II.8 Spécification des besoins

Etant donné la nécessité d'avoir un système informatique de qualité, notre application permettra de répondre aux besoins et exigence des différents acteurs entrent dans le système de restauration que ce soit gérant, cuisinier, serveur ou client.

II.8.1 Les besoins fonctionnels

Notre application "de gestion de restaurant " doit satisfaire les besoins fonctionnels suivants :

- Gestion du Stocks
- Gestion des commandes (par Smartphone)
- Gestion des commandes fournisseurs (pc et Smartphone)
- Gestion des fournisseurs (pc et Smartphones)
- Gestion des produits (menu, carte, traiteur)
- Gestion des réservations clients
- Gestion de la facturation clients
- Gestion des livres de caisse et de recettes
- Gestion du personnel via planning

II.8.2 Les besoins non fonctionnels

Ce sont les besoins qui permettraient d'améliorer les performances de l'application en termes de convivialité, d'ergonomie des interfaces. Parmi ces besoins on cite :

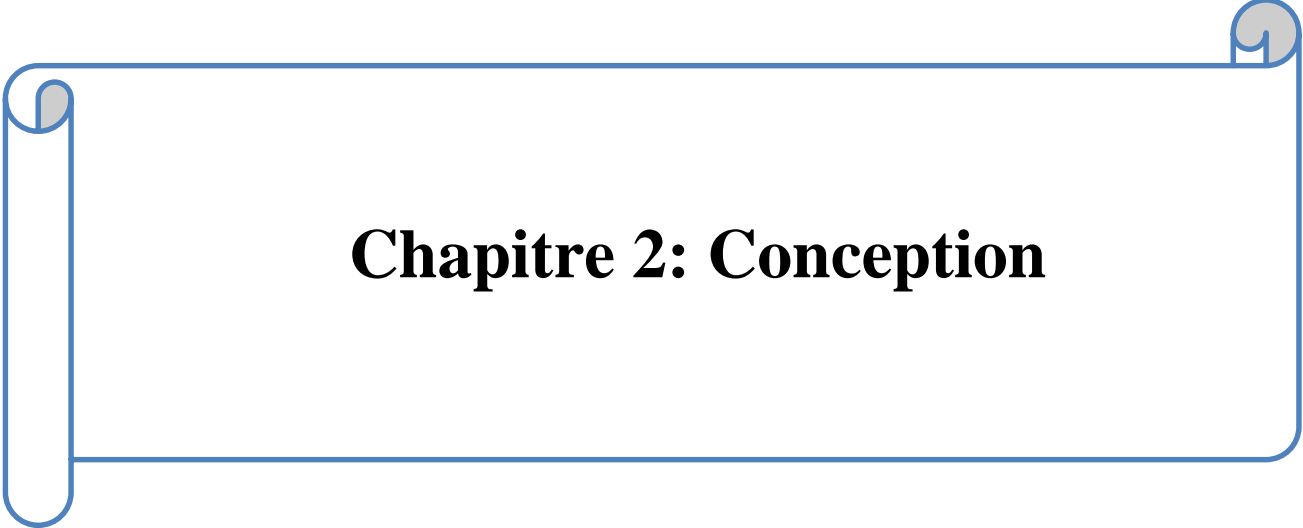
La sécurité : Besoins d'établissement de la connexion, il faut avoir une interface d'authentification qui permet à chaque utilisateur de se connecter pour consulter son profil.

L'ergonomie: l'application doit être facile à utiliser. Il doit présenter un enchaînement logique entre les et un texte compréhensible, visible et lisible.

La disponibilité : Lorsque n'importe quel utilisateur désire consulter l'application elle doit être disponible

III Conclusion

Dans ce chapitre nous avons fait une analyse de l'existant pour dégager les faiblesses du système et les critiques de l'existant, cette analyse nous a permis de récolter le maximum de données que nous allons analyser et traiter pour faire sortir les informations nécessaires à la réalisation de l'étude conceptuelle.



Chapitre 2: Conception

I Introduction

La conception de système d'information est la première étape dans le cycle de vie d'une application et pour la réalisation de cette étape nous avons commencé par une étude comparative entre la méthode merise et UML

II Choix de l'outil de conception

La conception d'un système d'information n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse, ce type de méthode est appelé analyse. Il existe plusieurs méthodes d'analyse parmi les méthodes on cite « **MERISE** » et « **UML** ».

II.1 Définition de MERISE

MERISE (Méthode d'Etude et de Réalisation Informatique des Systèmes d'Entreprise).

Merise est une méthode moderne de développement du système d'information que toute entreprise ou administration de quelle dimension ou taille qu'elle soit ait recours du fait qu'elle caractérise une démarche de construction d'information et même à une étude de données de base et de traitements.

La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et logique et physiques. [2]

II.2 Définition d'UML

UML (Unified Modeling Language) permet de présenter et de manipuler les concepts objet, et de faire une démarche d'analyse qui permet de concevoir une solution de manière itérative grâce aux diagrammes, et d'exprimer visuellement une solution objet.

Il se caractérise comme un langage de modélisation graphique et textuel qui est une étape importante du cycle de développement des systèmes utilisé ainsi pour visualiser, comprendre et définir des besoins, spécifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet , esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. Ces modèles doivent être proches de la réalité. [3]

II.3 L'étude comparative entre MERISE et UML

Cette étude fera l'objet d'une comparaison entre le langage UML et la méthode Merise et en fin de comparaison nous allons choisir la meilleure méthode pour notre travail.

II.3.1 Niveaux d'abstraction

L'approche Merise : Le cycle d'abstraction permet de sérier les niveaux de préoccupations lors de la description ou de l'analyse du système. Les trois niveaux retenus correspondent à des degrés de stabilité et d'invariance de moins en moins élevés.

- Le niveau conceptuel,
- le niveau logique,
- le niveau physique.

L'approche UML : propose différentes notions (cas d'utilisation, paquetage, classe, composant, nœud) et différents diagrammes pour modéliser les systèmes aux différents niveaux d'abstraction.

II.3.2 Approche fonctionnelle

L'approche Merise : propose une approche descendante où le système réel est décomposé en activités, elles-mêmes déclinées en fonctions. Les fonctions sont composées de règles de gestion, elles-mêmes regroupées en opérations. Ces règles de gestion au niveau conceptuel génèrent des modules décomposés en modules plus simples et ainsi de suite jusqu'à obtenir des modules élémentaires. Les limites d'une telle approche résident dans le fait que les modules sont difficilement extensibles et exploitables pour de nouveaux systèmes.

L'approche UML: Les fonctions cèdent la place aux cas d'utilisation qui permettent de situer les besoins de l'utilisateur dans le contexte réel. A chaque scénario correspondent des diagrammes d'interaction entre les objets du système et non pas un diagramme de fonction.

II.3.3 Dualité des données –traitements

L'approche Merise : propose de considérer le système réel selon deux points de vue: un point de vue statique (les données), un point de vue dynamique (les traitements). Il s'agit d'avoir une vision duale du système réel pour bénéficier de l'impression de relief qui en résulte, et donc consolider et valider le système final.

L'approche UML: L'approche objet associe les informations et les traitements. De cette façon, elle assure un certain niveau de cohérence. [4]

En conclusion, Merise et UML sont techniquement complémentaires .Si l'on considère que le SI est modélisable comme deux sous-systèmes inclus l'un dans l'autre: le SIO (système d'information organisationnel) représentant le métier, englobant le SII (système d'information informatisé) représentant l'application informatique associée .Merise est adaptée au SIO, UML au SII. Maintenant, selon le positionnement de chacun (consultant, analyste métier, concepteur de logiciel, développeur), on s'appuiera plus ou moins sur chaque partie de méthode.

II.4 La méthode de notre travaille

Selon l'étude effectuée, la méthode la plus convenable pour la modélisation des de notre système d'information, serait la méthode Merise. Car elle apporte une formalisation éclairant les choix à effectuer. Aussi, elle se caractérise comme étant un langage commun de référence, centré sur le système d'information et non sur l'informatique appliquée. Enfin, grâce à Merise la création d'une base de données devient claire et documentée, tout en atteignant des objectifs de fiabilités des données et d'évolution des applications.

III L'application de la méthode MERIS

La méthode MERISE représente le système d'information sur trois échelles :

- Niveau conceptuel
- Niveau logique (organisationnel)
- Niveau physique

III.1 Le niveau conceptuel

Le niveau conceptuel consiste à concevoir le SI en faisant abstraction de toutes les contraintes techniques ou organisationnelles et cela tant au niveau des données que des traitements, le niveau conceptuel répond à la question quoi ? (Quoi faire, avec quelles données).

Les formalismes Merise employés sont :

- Le modèle conceptuel des données (MCD)
- Le modèle conceptuel des traitements (MCT). [5]

III.2 Le niveau logique (organisationnel)

Le niveau organisationnel a comme mission d'intégrer dans l'analyse les critères liés à l'organisation étudiée, le niveau organisationnel fera préciser les notions de temporalité, de chronologie des opérations, Les questions posées, au niveau des traitements, sont : Qui ? Quand ?

Les formalismes mis en œuvre sont :

- Le modèle organisationnel des données (MOD)
- Le modèle logique des données (MLD)
- Le modèle organisationnel des traitements (MOT)
- Le modèle logique des traitements (MLT). [5]

III.3 Le niveau physique

Le niveau physique permet de définir l'organisation réelle (physique) des données, il apporte les solutions techniques, par exemple sur les méthodes de stockage et d'accès à l'information, c'est la réponse au Comment ?

Les formalismes employés sont :

- Le modèle physique des données (MPD)
- Le modèle opérationnel et physique des traitements (MOPT). [5]

III.4 Modèle Conceptuel de Données (MCD)

Avant la construction du MCD on doit définir les concepts de base sur lesquels elle se base:

- **Objet (individu ou entité) :** En d'autres termes, c'est la représentation dans le système d'information d'un objet matériel ou immatériel du réel perçu.
- **Relation (association) :** Une relation entre entités (objets) est une association perçue dans le réel entre deux ou plusieurs entités. Une relation n'a pas d'existence propre.
- **Propriété (ou l'attribut) :** Une propriété est une donnée (une information) élémentaire qui caractérise un objet ou une relation.
- **Occurrence :**
 - L'occurrence d'un objet(ou entité) : est un élément individualisé appartenant à cet objet (ou entité).
 - L'occurrence d'une relation : est une relation individualisée constituée d'une et d'une seule occurrence de chacun des objets participant à la relation.

- **Identifiant** : Identifiant d'un objet : est une propriété particulière qui caractérise de façon Unique chaque occurrence de cet objet.
- **Cardinalité** : Les cardinalités d'un objet par rapport à une relation s'expriment par deux nombres appelés :
 - Cardinalité minimale : (égale à 0 ou 1) C'est le nombre minimum de fois qu'une occurrence d'un objet participe aux occurrences de la relation.
 - Cardinalité maximale : (égale à 1 ou n) C'est le nombre maximum de fois qu'une occurrence de l'objet participe aux occurrences de la relation.
- **Dimension d'une relation** : La dimension d'une relation désigne le nombre d'objets participant à la relation. Selon le nombre d'objets qui participent à la relation il existe :
 - La relation entre deux objets : dite relation binaire.
 - La relation entre trois objets : dite relation ternaire.
 - La relation entre plusieurs objets (n) : dite relation n-aire.

La construction d'un modèle conceptuel de données permet au concepteur d'approfondir la connaissance du champ d'étude à travers son vocabulaire et ses entités, donc il suit ces étapes pour obtenir un MCD correct et exact :

- Elaboration de la liste des données : dictionnaire de données.
 - Recherche des individus et leurs identifiants avec rattachement des propriétés aux individus.
 - Recensement des relations avec l'étude des cardinalités de chaque couple individu-relation.
 - Schéma conceptuel de données descriptif (BRUT)
 - L'application des règles de vérifications, normalisation et décomposition.
- [6]

III.5 Modèle Conceptuel de Données du restaurant

Voilà notre modélisation du système d'information de restaurant :

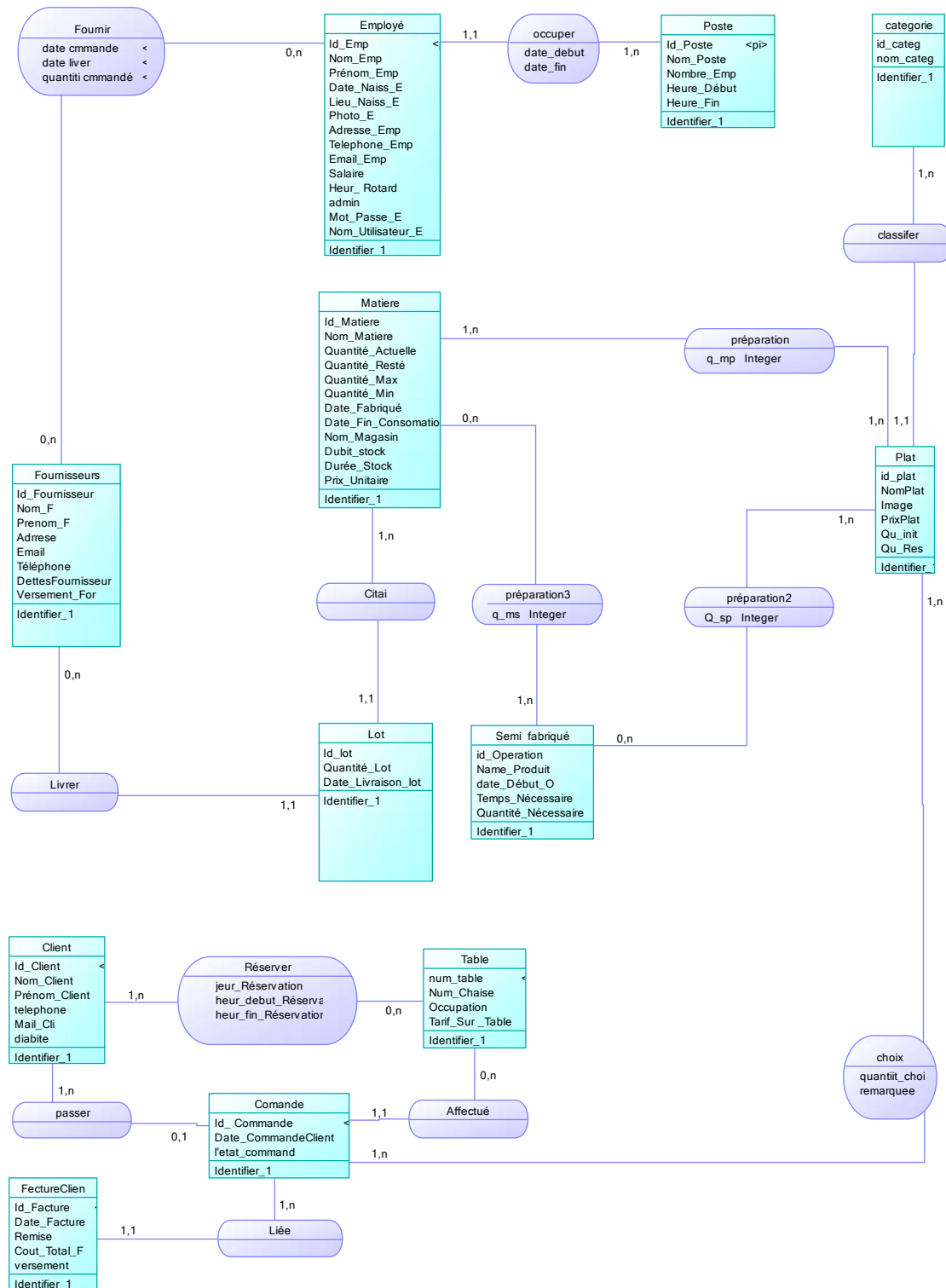


Figure 2-1: MCD

Dans la partie suivante nous allons expliquer les différentes tables et relations du modèle conceptuel de données que nous avons pu réaliser à partir des informations collecté durant l’analyse de l’existant.

III.5.1 La partie de l’employé :

On considère que l’employé dans un restaurant est un élément très important. Chaque employé possède les informations suivante : nom, prénom, date naissance, photo et l’adresse numéro du téléphone, les heures de retard qui vont nous être utile pour le calcul du salaire et en fin un nom utilisateur et un mot passe.

Chaque employé occupe un seul poste mais chaque poste est occupé par au moins un employé (*Figure 2 -1*).

Le restaurant contient plusieurs poste et chaque poste est identifié par : nom du poste ; nombre des employés dans chaque poste, heure début et fin du travail.

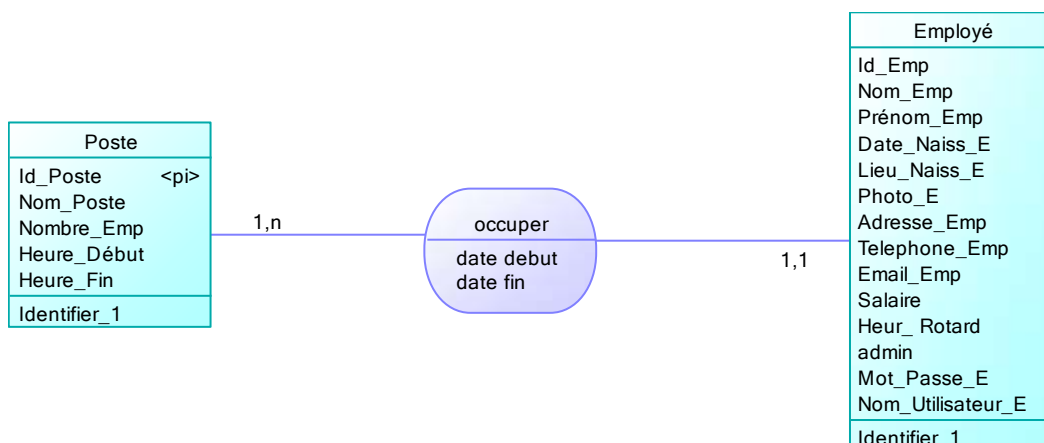


Figure 2-2: La partie de l’employé

III.5.2 La partie de client :

Le client peut réserver une table ou plusieurs si la table n’est pas occupée (*Figure 2-2*). Les attributs de chaque client sont : nom, prénom, numéro téléphone, diabétique client. Chaque client peut passer une ou plusieurs commandes, cette dernière est caractérisée par un identifiant, la date de

réserveation et l'état de la commande qui représente si la commande est satisfaite ou non.

L'entité « facture » regroupe les attributs suivant : identifiant, date, remise, cout total. La facture peut contenir plusieurs commandes d'un seul client.

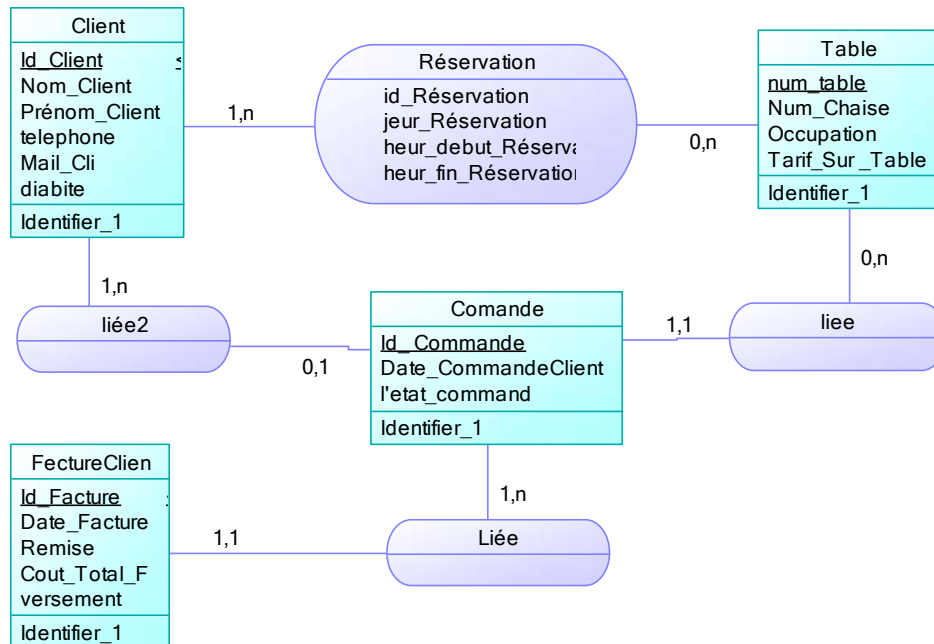


Figure 2-3: la réserveation et la facturation de client

III.5.3 La préparation du plat :

L'entité plat est caractérisée par un (nom et photo de plat, prix et la quantité actuelle), chaque plat peut contenir un ou plusieurs produits. Les produits sont divisés en deux catégories, les produits « semi finis » et les produits « matière première » ce dernier est caractérisé par code et désignation, durée de stock, prix unitaire. Les produits semi finis sont définies par (un code d'opération, date début et date fabrication, la quantité) (*Figure 2-3*).

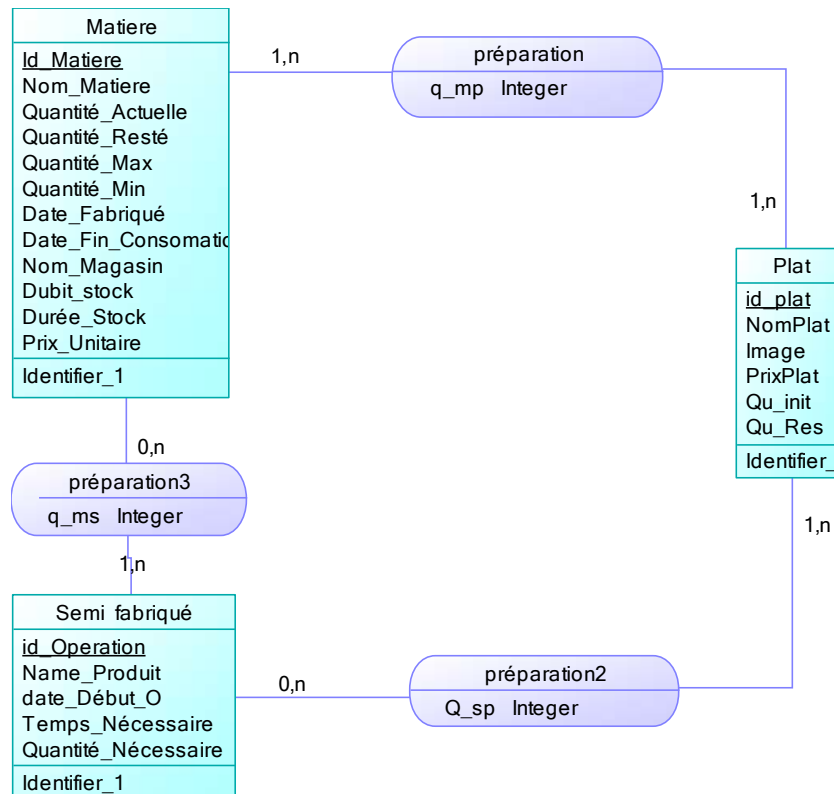


Figure 2-4: La préparation du plat

III.5.4 Les classifications des plats :

Pour utiliser le menu facilement les plats sont classifié par type, chaque type représente une catégorie de plat (exemple : entrée chaude, entrée froide ...) (Figure 2-4).

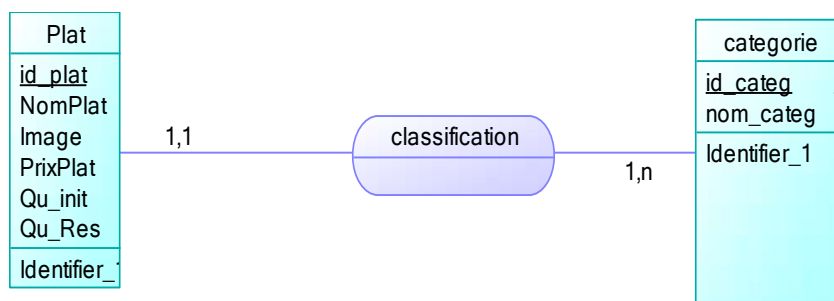


Figure 2-5: Les classifications des plats

III.5.5 Les commandes :

La commande du client contient des choix et chaque choix représente une quantité d'un plat ainsi que les remarques du client (*Figure 2-5*).

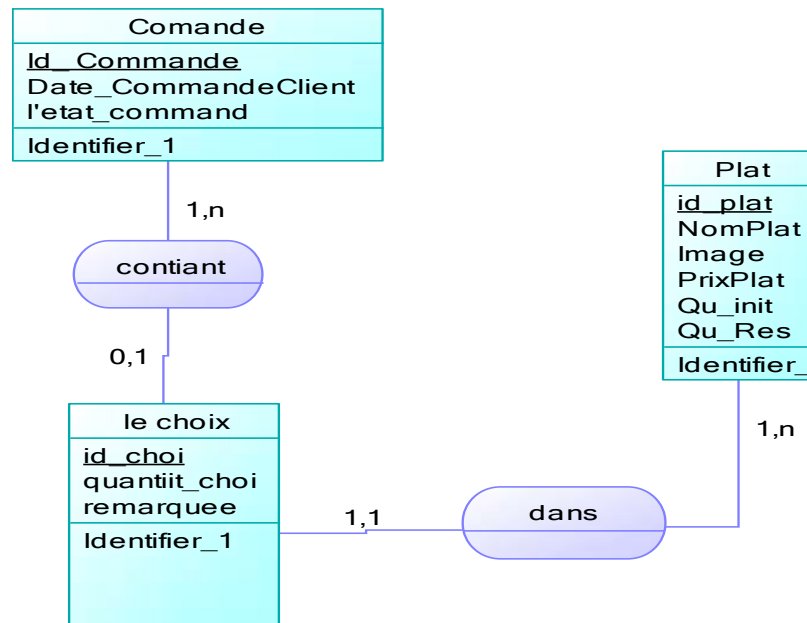


Figure 2-6: La commande du client

III.5.6 La gestion de stock :

La gestion du stock se compose de quatre entités, les quatre entités et leurs attributs sont les suivantes (*Figure 2-6*) :

- Entité fournisseur caractérisé par : un identifiant, un nom, un prénom, une adresse, un ou plusieurs numéro de téléphone.
- Entité lots caractérisé par un identifiant, une quantité et une date de livraison.
- Entité matière caractérisé par un identifiant, désignation matière, quantité en stock, quantité maximal et minimal et date de péremption.
- Entité commande fournisseur caractérisé par un identifiant, une date commande, une date de livraison et des quantités.

Quand la quantité d'une matière est inférieure à la quantité minimal défini par le magasiné, ce dernier choisi un fournisseur et lance la procédure d'une nouvelle commande d'achat.

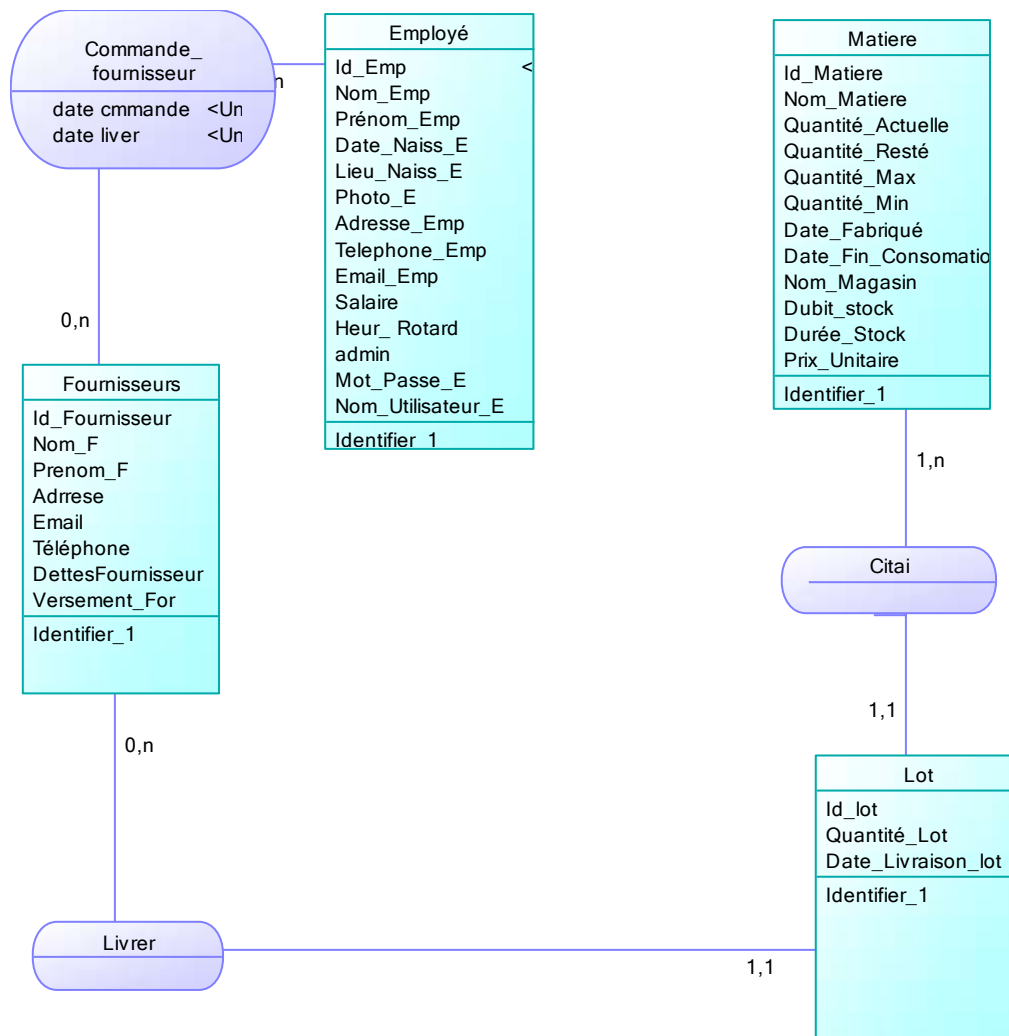


Figure 2-7: La gestion de stock

III.6 Modèle logique de Données (MLD)

La transcription d'un MCD en modèle relationnel s'effectue selon quelques règles simples qui consistent d'abord à transformer toute entité en table, avec l'identifiant comme clé primaire, puis à observer les valeurs prises par les cardinalités maximum de chaque association pour représenter celle-ci soit (**ex:** card. max 1-n ou 0-n) par l'ajout d'une clé étrangère dans une table existante, soit (**ex:** card. max n-n) par la création d'une nouvelle table dont la clé primaire est obtenue par concaténation de clés étrangères correspondant aux entités liées.[6]

Et voilà les tables Après l'application des règles :

- Categorie (**id categ** ,nom_categ)
- Client (Id_Client __,Nom_Client ,Prenom_Client, telephone, Mail_Clidiabite)

- Comande (**Id Commande**, #num_table, Date_Commande ,letat_command , #Id_Client)
- Commandes_Fournisseurs (**Id Cf**, # Id_Fournisseur, #Id_Matiere ,Date_Cf, Date_LivrQuantite_Command)
- Employeur (**Id Emp**, # Id_Poste, Nom_Emp ,Prenom_Emp, Prenom_Emp, Telephone_Emp, Salaire, Heur_Rotard, Admin ,Nom_Utilisateur_E ,Mot_Passe_E)
- FectureClien (**Id Facture2**, #Id_Commande, remise, Cout_Total_F, versement)
- Fournisseurs (**Id Fournisseur**,Nom_F ,Prenom_F ,Adrese, Email ,Telephone, Versement_For, DettesFournisseur)
- le choix (**id choix2** , #Id_Commande, # id_plat ,quantite_choix ,remarque)
- Lot (**Id lot**, #Id_Matiere, #Id_Fournisseur, Quantite_Lot ,Date_Livraison_lot)
- Matiere (**Id Matiere** ,Nom_Matiere ,Prix_Unitaire, Quantite_Actuelle ,Quantite_Max ,Quantite_Min ,Date_Fin_Consomation, Nom_Magasin, Dubit_stockDate_Fabrique)
- Plat (**id plat** , #id_categ ,NomPlat, Image ,PrixPlat ,Qu_init, Qu_Res)
- Poste (**Id Poste**,Nom_Poste, Num_Emp ,Heure_Début ,Heure_Fin)
- Sm(**id Operation**,Nom_Produit,HEUR_Debut_O,T_NécessaireQuantit é_Nécessaire)

Tbl (**Numéro table**,Num_Chaise, Occupation, Tarif_SurTable)

Les nouvelles tables sur notre modélisation sont :

- pr1 (#id_plat , #id_Operation ,q_sp)
- PR2 (#id_plat, #Id_Matiere , #q_mp)
- PR3 (#id_Operation, # Id_Matiere ,Q_ms)

Reservation (#Numéro_ table, #Id_Client, jeux_Reservation ,heur_debut_Reservation ,heur_fin_Reservation)

Les symboles utilisés sont :

<u>ID</u>	Les clés primaires
#	Les clés étrangères

IV CONCLUSION

Dans l'étape de la conception, nous avons réalisé un modèle conceptuel qui a été transformé en un modèle relationnel qui représente la structure de la base de données à mettre en place. Ce modèle a été élaboré loin de toute contrainte afin de répondre aux différents besoins de l'utilisateur.



Chapitre 3: Réalisation

I Introduction

Une fois la partie de la conception achevée, tous les éléments nécessaires au développement de l'application deviennent disponibles. Dans ce chapitre nous allons illustrer le matériel, logiciels de base et les outils de développement qui sont mis en œuvre pour la réalisation de notre application de gestion de restaurant ainsi que le détail de réalisation et l'implémentation des différentes interfaces de notre application.

II Matériels et langages de programmation

II.1 Matériel

La programmation de l'application a été effectuée sur un ordinateur ayant les caractéristiques suivantes :

Composant	Caractéristiques
Marque	HP 620
Processeur	Intel(R) Core (TM) Duo CPU T6670
RAM	3 Go
Disque Dur	300 Go
Ecran	15.3
Système d'exploitation	Microsoft Windows 7

Tableau 3 -1: les caractéristiques

II.2 Les langages de programmation

Avons la définition des langages de programmation on va définir l'architecture de notre application (**client/serveur**).

II.2.1 Présentation de l'architecture d'un système (client/serveur)

II.2.1.1 Notion de client/serveur

De nombreuses applications fonctionnent selon un environnement client/serveur. Cela signifie que des machines clients (des machines faisant partie du réseau) se contactent à un serveur, une machine généralement très puissante en termes de capacités d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, etc.

Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clients. On parle ainsi de client (client FTP, client de messagerie, etc.) lorsque l'on désigne un programme tournant sur une machine cliente, capable de traiter des informations qu'il récupère auprès d'un serveur (dans le cas du client FTP il s'agit de fichiers, tandis que pour le client de messagerie il s'agit de courrier électronique)[7].



Figure 3-1: l'architecture d'un système client/serveur

II.2.1.2 Serveur :

Dans un réseau informatique, un serveur est à la fois un ensemble de logiciels et d'ordinateurs les hébergeant dont le rôle est de répondre de manière automatique à des demandes envoyées par des clients par exemple ordinateur et logiciel et cela via le réseau.

Les serveurs sont d'usage courant dans les centres de traitement de données, les entreprises, les institutions, et les réseaux internet, où ils sont souvent un point central et sont utilisés simultanément par de nombreux utilisateurs pour stocker, partager et échanger des informations. Les différents usagers opérant à partir d'un client. [7]

II.2.1.3 Client:

Dans un réseau informatique un client est l'ordinateur et le logiciel qui envoient des demandes à un serveur. Il peut s'agir d'un logiciel manipulé par une personne. On appelle client aussi bien l'ordinateur depuis lequel les demandes sont envoyées mais aussi le logiciel qui contient les instructions relatives à la formulation des demandes et la personne qui opère les demandes.

L'ordinateur client est généralement un ordinateur personnel ordinaire, équipé de logiciels relatifs aux différents types de demandes qui vont être envoyées. [7]

II.2.2 Les différents modèles de client-serveur :

En fait, les différences sont essentiellement liées aux services qui sont assurés par le serveur. On distingue couramment :

- **Le client-serveur de donnée :**

Dans ce cas, le serveur assure des tâches de gestion, stockage et de traitement de données. C'est le cas le plus connu de client-serveur est qui est utilisé par tous les grands SGBD :

La base de données avec tous ses outils (maintenance, sauvegarde ...) est installée sur un poste serveur.

Sur les clients, un logiciel d'accès est installé permettant d'accéder à la base de données du serveur.

Tous les traitements sur les données sont effectués sur le serveur qui renvoie les informations demandées (souvent à travers une requête SQL) par le client.

- **Client-serveur de présentation :**

Dans ce cas la présentation des pages affichées par le client est intégralement prise en charge par le serveur. Cette organisation présente l'inconvénient de générer un fort trafic réseau.

- **Le client-serveur de traitement :**

Dans ce cas, le serveur effectue des traitements à la demande du client. Il peut s'agir de traitement particulier sur des données, de vérification de formulaires de saisie, de traitements d'alarmes.

Ces traitements peuvent être réalisés par des programmes installés sur des serveurs mais également intégrés dans des bases de données (triggers, procédures stockées), dans ce cas, la partie donnée et traitement sont intégrés. [7]

II.2.3 Les différentes architectures :

Architecture à 1-tiers :

Dans une approche d'application de type 1-tiers, les trois couches sont fortement et intimement liées, et s'exécutent sur la même machine. Dans ce cas, on ne peut pas parler d'architecture client-serveur mais d'informatique centralisée. Dans un contexte simple utilisateur, la question ne se pose pas, mais dans un contexte multiutilisateurs, on peut voir apparaître deux types d'architectures mettant en œuvre des applications 1-tiers : des applications sur site central ; des applications réparties sur des machines indépendantes communiquant par partage de fichiers.

L'architecture 2 tiers :

Dans une architecture deux tiers, encore appelée client-serveur de première génération ou client-serveur de données, le poste client se contente de déléguer la gestion des données à un service spécialisé. Le cas typique de cette architecture est une application de gestion fonctionnant sous Windows ou Linux et exploitant un SGBD centralisé.

Ce type d'application permet de tirer parti de la puissance des ordinateurs déployés en réseau pour fournir à l'utilisateur une interface riche, tout en garantissant la cohérence des données, qui restent gérées de façon centralisée.

La gestion des données est prise en charge par un SGBD centralisé, s'exécutant le plus souvent sur un serveur dédié. Ce dernier est interrogé en utilisant un langage de requête qui, le plus souvent, est SQL. Le dialogue entre client et serveur se résume donc à l'envoi de requêtes et au retour des données correspondant aux requêtes.

L'architecture 3 tiers :

Les limites de l'architecture deux tiers proviennent en grande partie de la nature du client utilisé :

Le frontal est complexe et non standard (même s'il s'agit presque toujours d'un PC sous Windows),

Le middleware entre client et serveur n'est pas standard (dépend de la plateforme, du SGBD ...).

La solution résiderait donc dans l'utilisation d'un poste client simple communiquant avec le serveur par le biais d'un protocole standard.

Dans ce but, l'architecture trois tiers applique les principes suivants :

- Les données sont toujours gérées de façon centralisée,
- La présentation est toujours prise en charge par le poste client,
- La logique applicative est prise en charge par un serveur intermédiaire.

Cette architecture trois tiers, également appelée client-serveur de deuxième génération ou client-serveur distribué sépare l'application en 3 niveaux de services distincts, conformes au principe précédent :

- **premier niveau** : l'affichage et les traitements locaux (contrôles de saisie,

mise en forme de données...) sont pris en charge par le poste client.

- **deuxième niveau** : les traitements applicatifs globaux sont pris en charge par le service applicatif.
- **troisième niveau** : les services de base de données sont pris en charge par un SGBD. [7]

II.2.4 Caractéristiques des systèmes client-serveur :

Les éléments qui caractérisent une architecture client-serveur sont :

- **Service :**

Le modèle client-serveur est une relation entre des processus qui tournent sur des machines séparées. Le serveur est un fournisseur de services. Le client est un consommateur de services.

- **Partage de ressources :**

Un serveur traite plusieurs clients et contrôle leurs accès aux ressources.

- **Protocole asymétrique :**

Conséquence du partage de ressources, le protocole de communication est asymétrique le client déclenche le dialogue ; le serveur attend les requêtes des clients.

- **Transparence de la localisation :**

L'architecture client-serveur doit masquer au client la localisation du serveur (que le service soit sur la même machine ou accessible par le réseau).

Transparence par rapport aux systèmes d'exploitation et aux plates-formes matérielles. Idéalement, le logiciel client serveur doit être indépendant de ces deux éléments.

- **Message :**

Les messages sont les moyens d'échanges entre client et serveur.

- **Encapsulation des services :**

Un client demande un service. Le serveur décide de la façon de le rendre une mise à niveau du logiciel serveur doit être sans conséquence pour le client tant que l'interface message est identique.

- **Evolution :**

Une architecture client-serveur doit pouvoir évoluer horizontalement (évolution du nombre de clients) et verticalement (évolution du nombre et des caractéristiques des serveurs. [7])

II.2.5 Avantages de l'architecture client/serveur:

Le modèle client/serveur est particulièrement recommandé pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux atouts sont :

- **Des ressources centralisées** : étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisées, afin d'éviter les problèmes de redondance et de contradiction.
- **Une meilleure sécurité** : car le nombre de points d'entrée permettant l'accès aux données est moins important.
- **Une administration au niveau serveur** : les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés.
- **Un réseau évolutif** : grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure. [7]

II.2.6 Inconvénients du modèle client/serveur :

L'architecture client/serveur a tout de même quelques lacunes parmi lesquelles:

- Un coût élevé dus à la technicité du serveur
- Un maillon faible : le serveur est le seul maillon faible du réseau client/serveur, Étant donné que tout le réseau est architecturé autour de lui ! Heureusement, le Serveur a une grande tolérance aux pannes (notamment grâce au système RAID Qui permet un haut niveau de disponibilité). [7]

III Système de Gestion de Base de Données

III.1 La base de données

Une base de données est composée de données stockées dans des mémoires de masse sous une forme structurée, et accessibles par des applications différentes et des utilisateurs différents. Une base de données doit pouvoir être utilisée par plusieurs utilisateurs en même temps

III.2 Définition du système de gestion de base de données

Un système de gestion de base de données (SGBD) est un logiciel de haut niveau qui permet de manipuler ses informations, il peut être perçu comme un ensemble de logiciels systèmes permettant aux utilisateur d'insérer, de

modifier et de rechercher efficacement des données spécifiques dans une masse d'informations partagées par de donnée désignée par un nom dans un ensemble d'objet, mais aussi à partir de relations entre objets .

En résumé, un SGBD peut donc apparaître comme un outil informatique permettant la sauvegarde, l'interrogation, la recherche et la mise en forme de données stockées en mémoire pour :

- Permettre l'accès aux données de façon simple.
- Autoriser un accès aux informations à de multiples utilisateurs.
- Manipuler les données présentes dans la base de données (insertion, suppression, modification).
- Le contrôle d'intégrité des données accédées.
- Le maintien de la cohérence des données entre elles. [7]

III.3 Les niveaux de la SGBD

- **Le niveau physique :**

C'est ce que fait le SGBD physiquement, la gestion des données et des index, le partage de ces données et de la concurrence des accès, la distribution des données à travers le réseau.

- **Le niveau logique :**

C'est ce que fait le SGBD logiquement, la définition de la structure des données, la gestion de la confidentialité (sécurité), le maintien de l'intégrité entre les données et les index, la consultation et la mise à jour des données.

- **Le niveau externe :**

C'est la mise en forme et la présentation des données aux programmes D'applications et aux utilisateurs interactifs. Ceux-ci expriment leurs critères de recherche à l'aide de langage basés sur des procédures de recherche progressives en référençant des données dérivées de la base. [7]

III.4 Quelques SGBD connus et utilisés

Il existe de nombreux systèmes de gestion de bases de données, et voici la liste :

- **ACCES** : Plate-forme Windows, mono-poste, licence commerciale
- **SQL SERVER**:Plate-forme Windows, mode client/serveur, licence commerciale

- **ORACLE:**Plate-forme Windows et Linux, mode client/serveur, licence commerciale
- **SYBASE:**Plate-forme Windows et Linux, mode client/serveur, licence commerciale
- **POSTGRESQL:**Plate-forme Windows et Linux, mode client/serveur, licence libre
- **MYSQL:**Plate-forme Windows et Linux, mode client/serveur, licence libre

III.5 SQL Server

SQL (Structured Query Language) en français langage de requête structurée

Pour la création des tables de notre base de données on a utilisés Microsoft SQL Server qui est un langage informatique normalisé servant à effectuer des opérations sur des bases de données relationnelles. La partie langage de manipulation de données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.

En plus du langage de manipulation de données, la partie langage de définition de données permet de créer, et de modifier l'organisation des données dans la base de données, la partie langage de contrôle de transaction permet de commencer et de terminer des transactions, et la partie langage de contrôle de données permet d'autoriser ou d'interdire l'accès à certaines données à certaines personnes. [8]

III.5.1 Fonctionnement du SQL Server

Microsoft SQL Server propose plusieurs fonctionnements :

- Pour les requêtes, SQL Server utilise T-SQL (Transact-SQL), il s'agit d'une implémentation de SQL qui prend en charge les procédures stockées, les fonctions utilisateur ou UDF (User DefinedFunction) et les déclencheurs (trigger).
- Pour les transferts de données, SQL Server utilise le format TDS (Tabular Data Stream) dont les spécifications sont publiques.
- SQL Server étant doté de deux moteurs de bases de données, l'un relationnel et l'autre décisionnel, il est possible de faire des requêtes en langage MDX ou DMX spécifique à l'analyse de données pour les bases décisionnelles.
- SQL Server est un SGBD relationnel. Il est possible de définir des liens entre les tables de façon à garantir fortement l'intégrité des données qui y

sont stockées. Ces liens d'intégrité peuvent être utilisés pour modifier ou supprimer en chaîne des lignes liées.

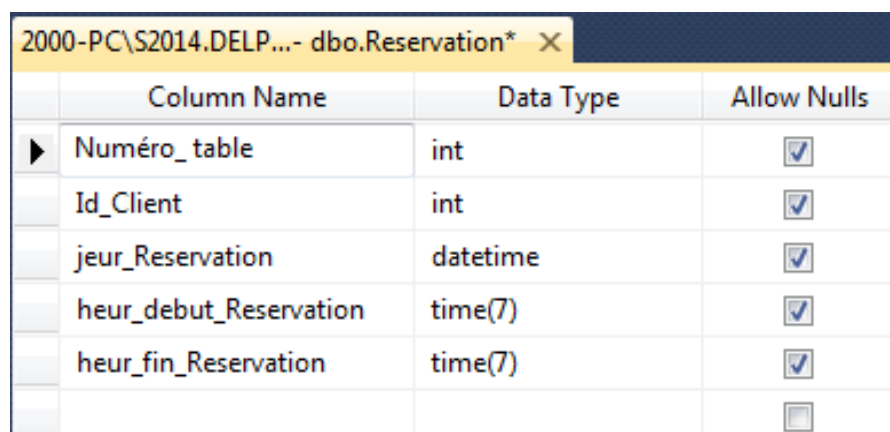
- SQL Server est un SGBD transactionnel. Il est capable de préparer des Modifications sur les données d'une base et de les valider ou de les annuler de Façon atomique, c'est -à-dire en "tout ou rien.
- Du fait de son aspect multi base, SQL Server dispose d'une sécurité à deux niveaux : niveau serveur, par le biais des comptes de connexion et niveau base, par le biais des utilisateurs SQL. [8]

III.5.2 La création de la base de données

Notre base de données est représentée par **18** tables.

Exemple :

- La table de réservation



The screenshot shows a window titled '2000-PC\S2014.DELP...- dbo.Reservation*'. It displays a table with the following columns: 'Column Name', 'Data Type', and 'Allow Nulls'. The table contains five rows of data:

Column Name	Data Type	Allow Nulls
Numéro_table	int	<input checked="" type="checkbox"/>
Id_Client	int	<input checked="" type="checkbox"/>
jour_Reservation	datetime	<input checked="" type="checkbox"/>
heur_debut_Reservation	time(7)	<input checked="" type="checkbox"/>
heur_fin_Reservation	time(7)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Figure 3-2: La table de réservation

- La table de commande de fournisseur

Column Name	Data Type	Allow Nulls
Id_Cf	int	<input type="checkbox"/>
Id_Fournisseur	int	<input type="checkbox"/>
Id_Matiere	int	<input type="checkbox"/>
Date_Cf	datetime	<input checked="" type="checkbox"/>
Date_Livr	datetime	<input checked="" type="checkbox"/>
Quantite_Command	int	<input type="checkbox"/>
		<input type="checkbox"/>

Figure 3-3 : La table de commande de fournisseur

III.5.3 Représentation des requêtes SQL

Voici quelque exemple des requêtes SQL :

- **les commandes et les factures**

```
select Nom_Client ,num_table ,letat_command ,Id_Facture2,Cout_Total_F
,Date_Commande
from client,Comande,FectureClien');
WHERE (Client.Id_Client=comande.Id_Client) and
(comande.Id_Commande=FectureClien.Id_Commande AND
Date_Commande=:q;
```

Cette requête permet de faire la jointure entre les tables client et facture et commande client pour la sélection du nom du client leur commande et facture

- **la réservation**

```
select Nom_Client,id_Reservation,num_table ",heur_debut_Reservation
heur_fin_Reservation from client,Reservation
WHERE (Client.Id_Client=Reservation.Id_Client)
Add(AND jeur_Reservation=:q)
```

Cette requête permet de abécher la liste des réserverions des clients dans un jour définir par l'utilisateur.

- **Les plats**

Select nomplat from plat where id_categ='+DBEdit3.Text

Pour affiché les plats de la différent catégorie

- **Les fournisseurs**

```
select sum(DettesFournisseur) as 'd' , sum(Versement_For) as 'v' from  
Fournisseurs
```

Cette requête permet de calcule la somme des dattes et la somme des versements de toute les fournisseurs.

IV Delphi

Borland Delphi est un environnement de développement de type RAD (Rapide Application Développement) basé sur le langage pascal (plus particulièrement le Pascal objet). Il permet de réaliser rapidement et simplement des applications Microsoft Windows XP, Microsoft Windows 2000 et Microsoft Windows 98, Windows Vista, Windows 7,8.

Nous avons opté pour la version 7 de Delphi car elle fournit un ensemble d'outils nécessaires pour développer, tester et déployer des applications, notamment une importante bibliothèque de composants réutilisables, une suite d'outils de conception, de modèles d'applications, de fiches et d'experts de programmation que les versions précédentes du logiciel ne possédaient pas.[9]

IV.1 Les Avantages de Delphi

Delphi apporte une grande souplesse au développeur. Lorsque Delphi génère un fichier (.exe), il s'agit d'un vrai exécutable. Aucun autre fichier n'est nécessaire pour l'exécution. Vous obtenez donc une application plus propre, et plus facile à distribuer et à maintenir. Vous n'avez à distribuer qu'un seul fichier, sans dépendre de DLL ou d'autres fichiers.

Pour les entreprises soucieuses de l'établissement de normes et de standards, Delphi est également utile. Supposons que vous écriviez des applications Delphi dans une entreprise de 5000 employés. Chaque fois que vous devez déployer une nouvelle application, vous devez envoyer un fichier de 1Mo à tous les utilisateurs. Ceci peut vite encombrer le réseau. Fort heureusement, Delphi vous permet de regrouper les composants standards dans un paquet. Une fois ce paquet installé sur les machines, vous pouvez l'utiliser pour toutes les nouvelles applications que vous déploierez. Dès lors, il vous suffit d'envoyer un fichier exécutable de 200 ko au lieu de 1 Mo. Cette technique est l'une des nouvelles

fonctionnalités de Delphi 3, et permet de minimiser la taille des applicatifs transmis à chaque machine dès lors qu'un paquet standard a été installé.

Delphi vous offre donc un compilateur optimisé qui vous donnera une application rapide sans qu'il soit nécessaire de fournir plus d'efforts pour optimiser le programme qu'il n'en avait fallu pour l'écrire. [9]

V Connexion aux bases de données

Delphi propose en standard des composants pour manipuler des bases de données et des tables relationnelles. Tous ces composants ont une base commune : le Borland DatabaseEngine (BDE), un noyau stable, complet et puissant.

V.1 Les composants utilisés sur notre projet :

Les composants sources permettent d'accéder à des bases de données. Ils sont accessibles dans l'onglet (**DBGO**) la palette des composants

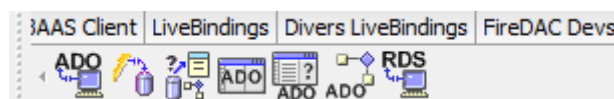


Figure 3-4: les composants Delphi pour la connexion

Ces composants sont (invisibles), non visuels. Ils apparaissent sur une fiche en Mode création mais pas à l'exécution du programme.

V.2 Le code Delphi de connexion

- `DataModule7.ADoconnection1.ConnectionString:=
'provider='+edit1.text+'persist security info=false;user
id='+edit2.text+';password='+edit3.text+';Initial
Catalog='+edit4.text+';Data Source='+edit5.text+'\''+edit6.text ;`
- `DataModule7.adoconnection1.LoginPrompt:=false;`
- `DataModule7.adoconnection1.Connected:=true;`

VI Les interfaces de l'application

Dans cette partie on va présenter quelques interfaces de l'application.

Si vous exécutez l'application la première fois Vous devez créer un mot de passe et un nom d'utilisateur Et remplissez les paramètres de connexion au serveur (l'adresse IP, le nom de la base de données,...) a l'aide des deux interfaces suivante:

The image shows a Windows-style window titled 'Form2' with a blue-themed background. The window contains a registration form titled 'Inscrite'. It has two input fields: 'Le Nom d'utilisateur :' with the text 'admin' and 'Mot de passe :' with six dots. At the bottom, there are two buttons: 'Inscrite' and 'Exite'.

Figure 3 -5: l'interface d'inscription

Et le deuxième :



Figure 3-6: l'interface de connexion

Après avoir terminé les deux étapes précédentes L'application doit redémarrée.

Et Après le redémarrage de l'application l'interface d'Authentification est affichée cette fenêtre s'affiche lors de lancement de l'application, elle nous demandera d'introduire un login et un mot de passe pour accéder à l'application.

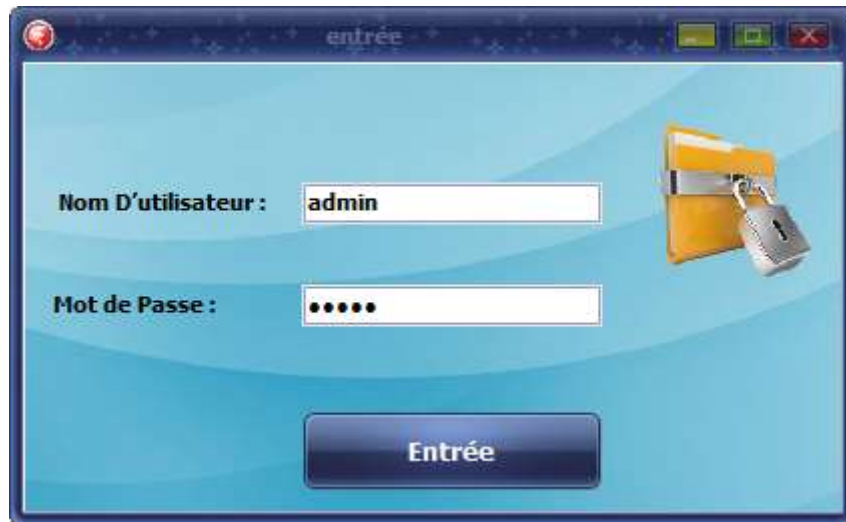


Figure 3-7: l'interface d'entrée

Et la modification de mot passe est fait à l'aide de l'interface suivante :

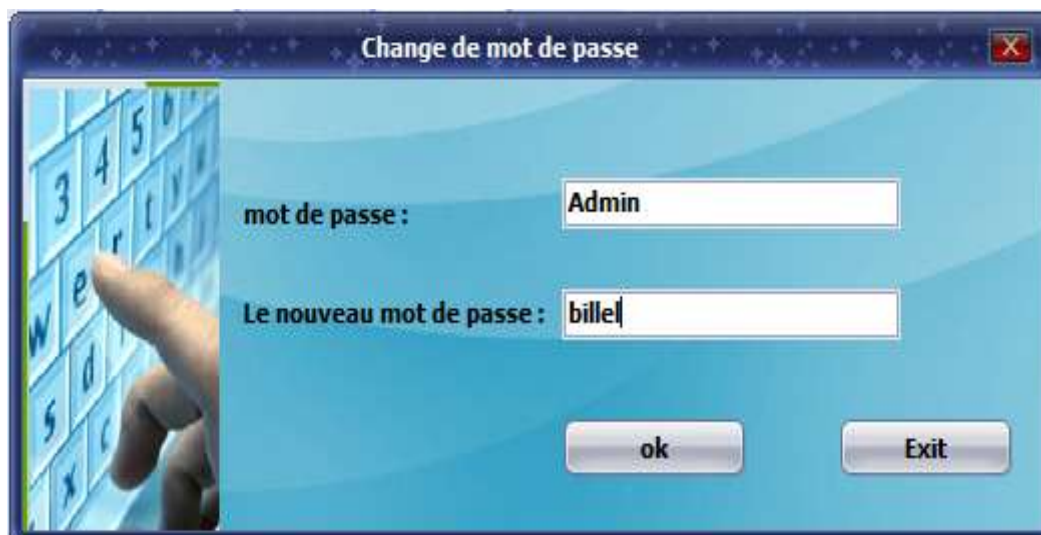


Figure 3-8: l'interface de la modification de mot passe

Et dans le cas de la saisie des données (le nom d'utilisateur et mot de passe) Correctement L'interface principale apparaît.



Figure 3-9: l'interface d'accueil

Dans cette interface, nous trouvons deux diagrammes le premier représente Le nombre de clients dans tous les jours et le et le deuxième représente le nombre de commandes préparer et le nombre des commandes qui est on attente.et Plusieurs des boutons qui permet à l'utilisateur de se déplacer au reste des interfaces de l'application.

L'interface de gestion de stock :

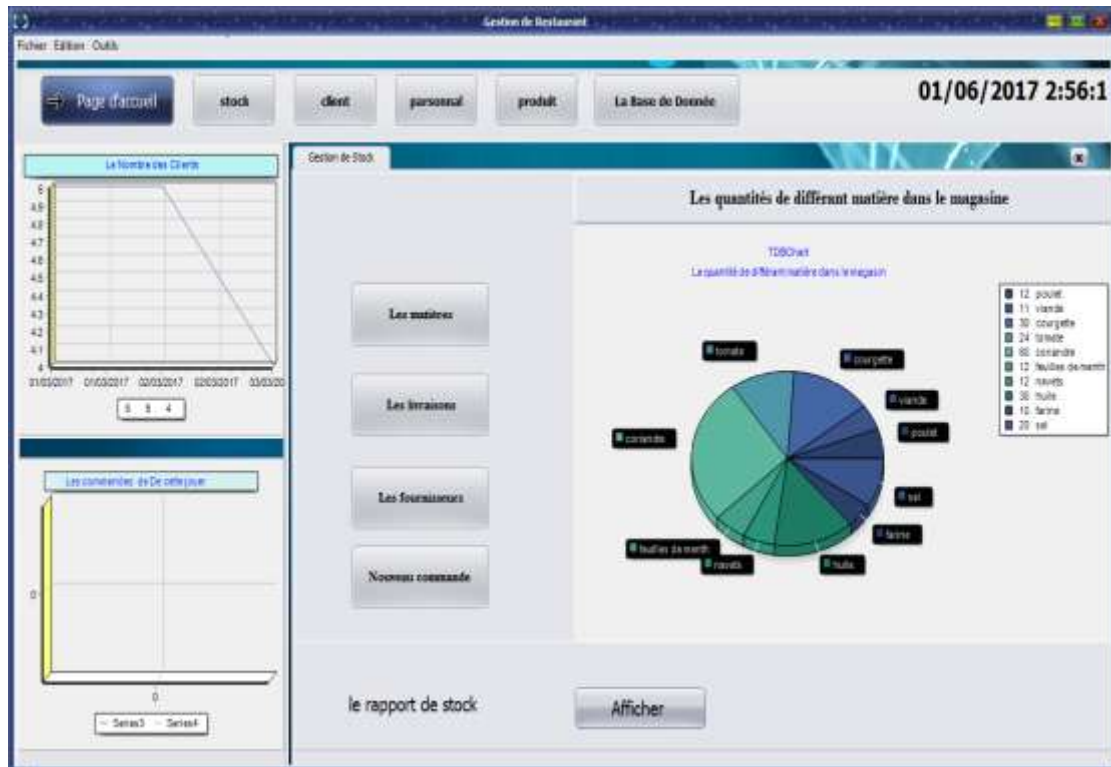


Figure 3-10: L'interface de gestion de stock

Il y a dans cette partie de plusieurs options relatives à la gestion de stocke :

- La liste des fournisseurs de restaurant
- La liste de livraisons
- La liste des matières dans le stock

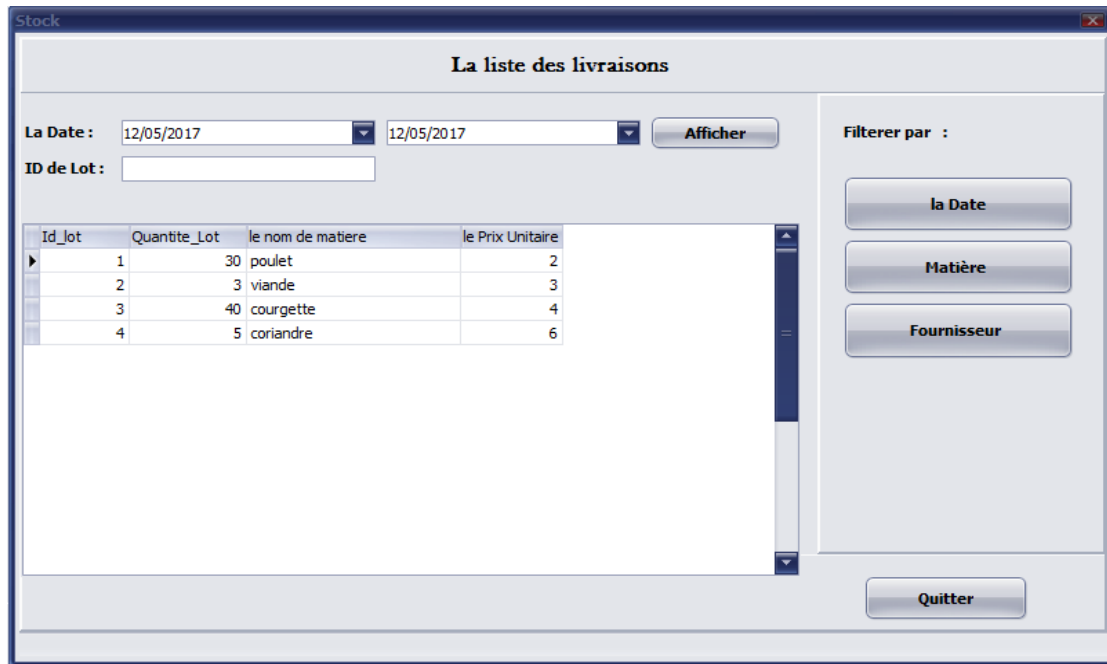


Figure 3-11: la liste de livraisons

Et un bouton pour créer un ordre d'achat et Pour mener à bien ce processus doit Vous devez remplir les données nécessaires de cette commande le nom de matière et le nom de fournisseur et le la date de livraison et quantité commandé.

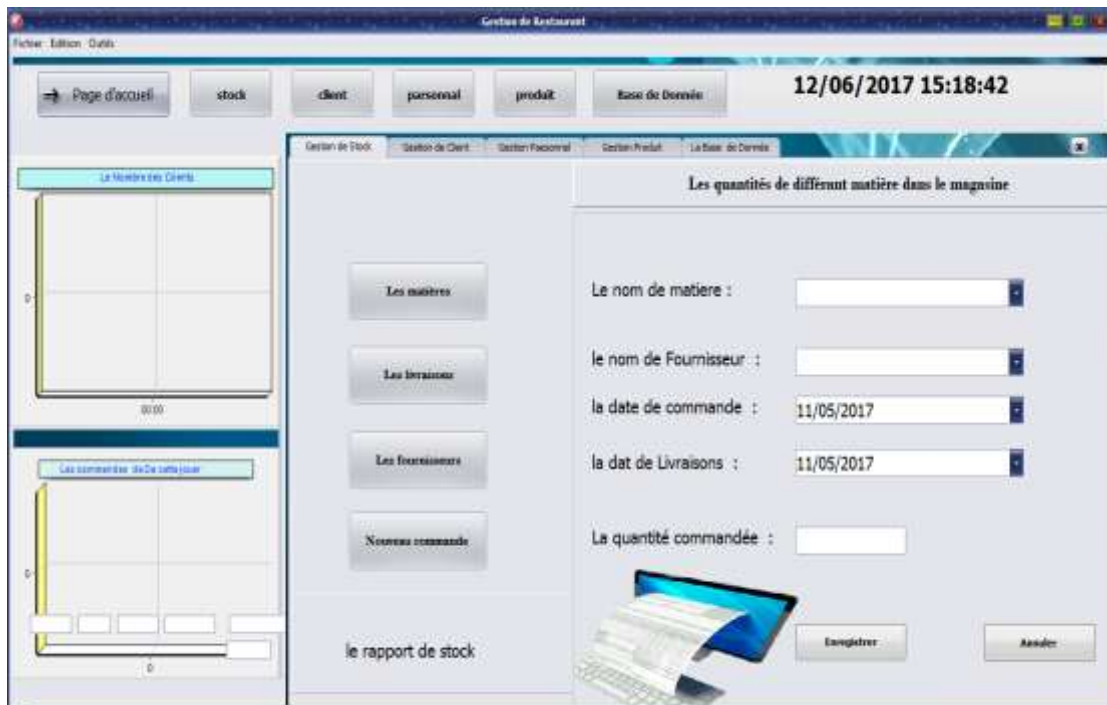


Figure 3-12:Interface qui permet d'écrire un ordre d'achat

Et voilà un exemple de rapport qui présente les Informations pour chaque matière stockée dans le magasin.



Figure 3-13: exemple de rapport de stock

L'interface de gestion des clients :

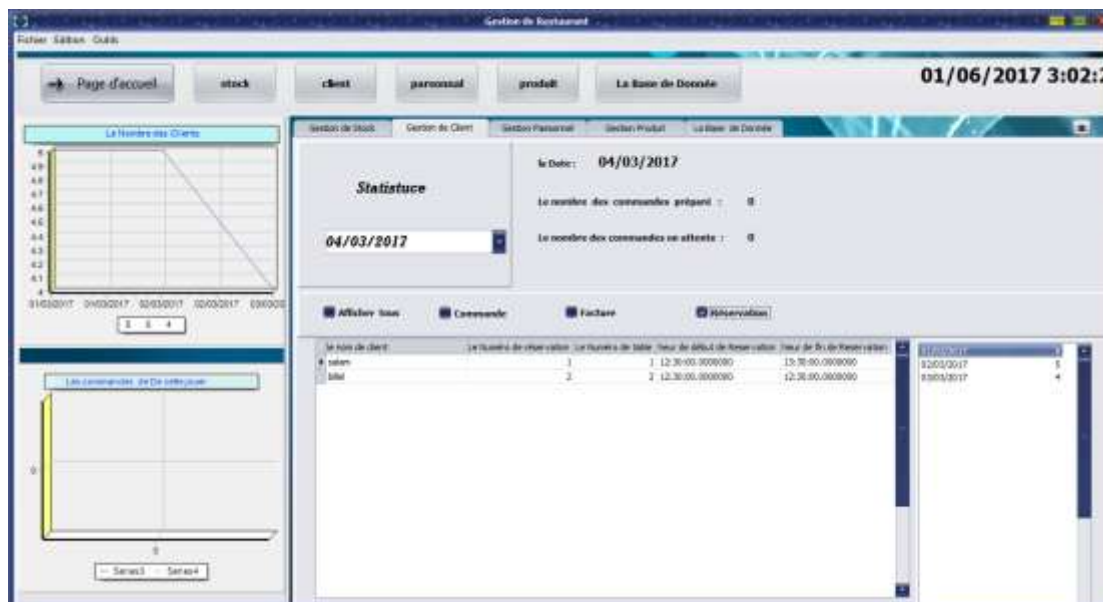


Figure 3-14: L'interface de gestion des clients

Cette interface contient Toutes les listes d'information Liés au client :

- Les réservations des clients
- Les commandes clients
- Les factures

Avec la possibilité d'afficher ces informations dans un jour définir par l'utilisateur.

L'interface de gestion de personelle :

Cette interface permet de voir les informations de Tous les travailleurs de la restauration.

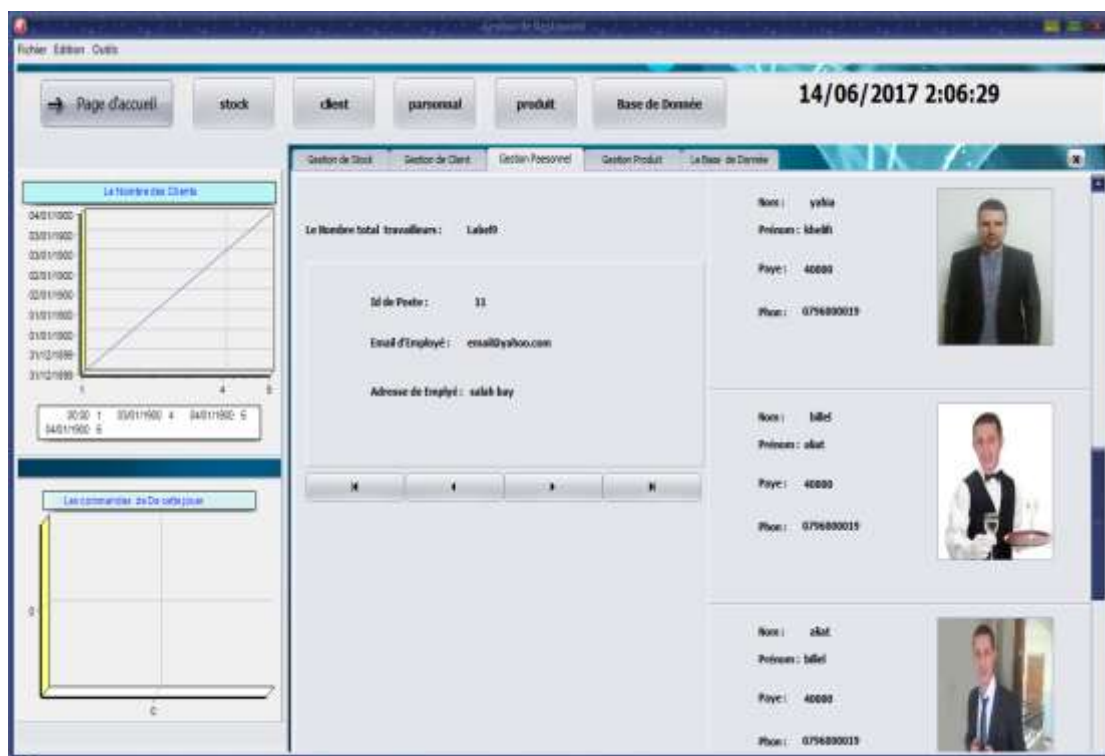


Figure 3-15: L'interface de gestion de personelle

L'interface de gestion des plats :

Sur le côté gauche tous les plats et les autres tables est les tables de les matières et les produits semi fin qui compose les plat en cas ce plat est sélectionner

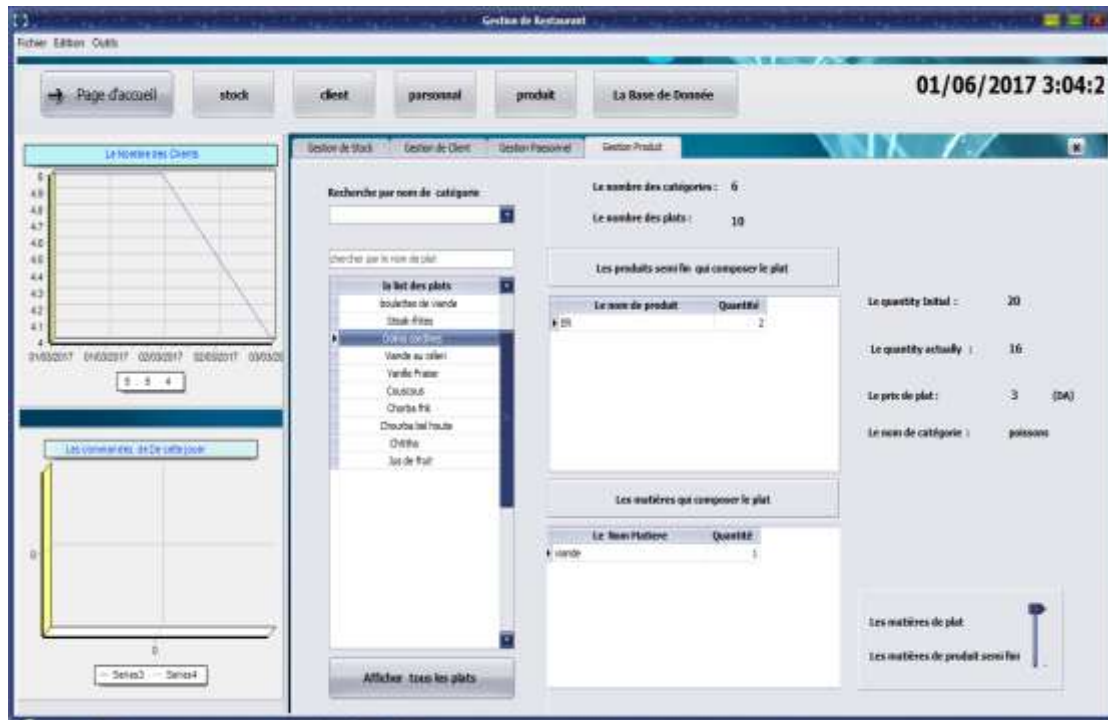


Figure 3-16: L'interface de gestion des plats

L'interface de gestion de la base de données :

Cette interface permet à l'utilisateur de supprimer et de modifier et ajouté à tous les liste

Suivent :

- Client
- Employé
- Fournisseur
- Matière première
- Post
- table



Figure 3-17: L'interface de gestion de la base de données

Exemple :

Cette interface si vous ajoutez un nouvel employé



Figure 3-18: L'interface qui permet l'ajout un employé

Et Cette interface dans le cas de la modification des données des employés



The screenshot shows a window titled "la base de données" with a menu bar containing "Fichier" and "Aide". The main area is a form for editing an employee's details. On the left, there are several input fields: "le nom de Poste :" with a dropdown menu showing "Aide-cuisinier"; "le nom de L'employeur :" with the value "djamal"; "Prenom de L'employeur :" with the value "aliat"; "Le numéro de téléphone :" with the value "0796000019"; "la La date de naissains :" with a date picker showing "04/05/2017"; "Lieu de naissains :" with the value "aine walmame"; "Adresse de l'employeur :" with the value "salah bay"; and "Email :" with the value "email@yahoo.com". On the right, there is a small portrait photo of a man in a chef's hat. Below the photo is an "Admin" section with "Nom d'utilisateur :" and "Mot de Passe :" fields. At the bottom, there are two buttons: "Enregistrer" and "Exit".

Figure 3-19: L'interface qui permet de modifié les informations d'un employé

L'interface qui permettre d'ajouter une matière



The screenshot shows a window titled "la base de donnée" with a menu bar containing "Fichier" and "Aide". The main area features a large, vibrant image of fresh vegetables (tomatoes, bell peppers, onions, mushrooms, and corn) on the left side. On the right, there are four input fields: "Le nom de matière", "Quantité (MAX) :", "Quantité (MIN) :", and "Le nom de magasin :". At the bottom, there are two buttons: "Enregistrer" and "Exit". The text "PixstAhead.com" is visible in the top right corner of the window.

Figure 3-20: ajouter une matière

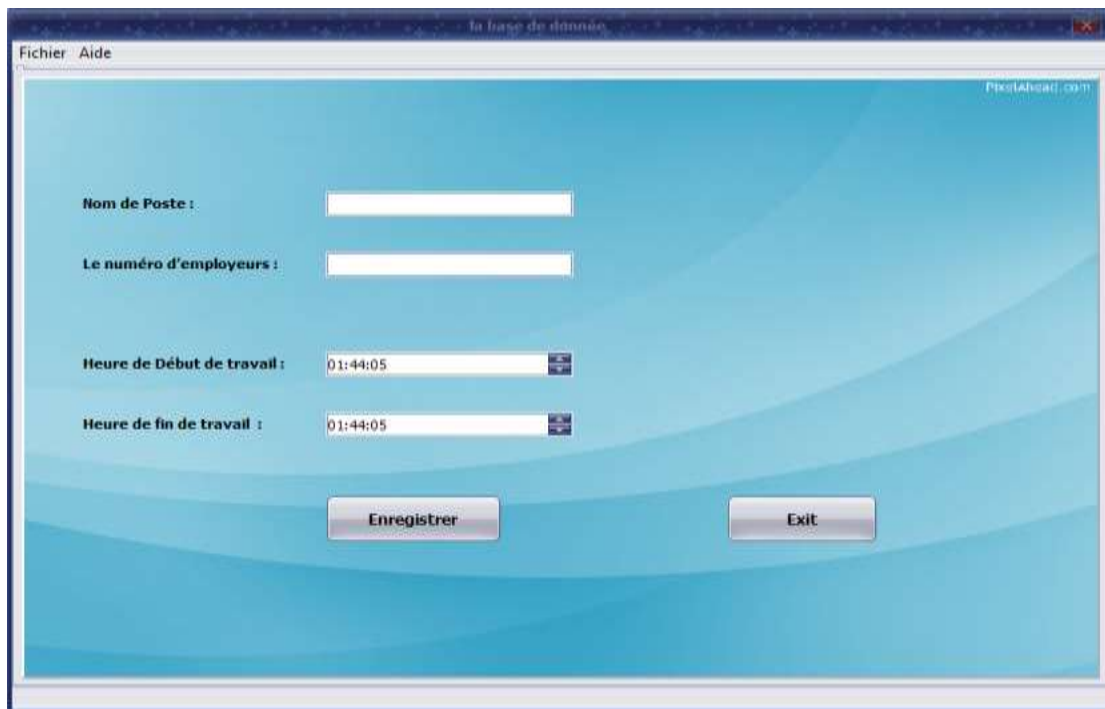
L'interface qui permette d'ajouter un client :



The screenshot shows a window titled "la base de donnée" with a menu bar containing "Fichier" and "Aide". The main area features a background image of a man in a blue shirt sitting at a table with a waiter in a blue uniform. On the right side, there are four input fields labeled "Nom :", "Prenom :", "telephone :", and "Mail :". Below these fields is a checkbox labeled "diabite". At the bottom, there are two buttons: "Enregistrer" and "Exit".

Figure 3-21: ajouter un client

L'interface qui permette d'ajouter un poste :



The screenshot shows a window titled "la base de donnée" with a menu bar containing "Fichier" and "Aide". The main area is a light blue background. On the left side, there are four input fields: "Nom de Poste :", "Le numéro d'employeurs :", "Heure de Début de travail :", and "Heure de fin de travail :". The "Heure de Début de travail" and "Heure de fin de travail" fields contain the value "01:44:05" and have small icons to the right. At the bottom, there are two buttons: "Enregistrer" and "Exit".

Figure 3-22: ajouter un poste

L'interface qui permette d'ajouter une catégorie:



Figure 3-23: ajouter une catégorie

L'interface qui permette d'ajouter une table :

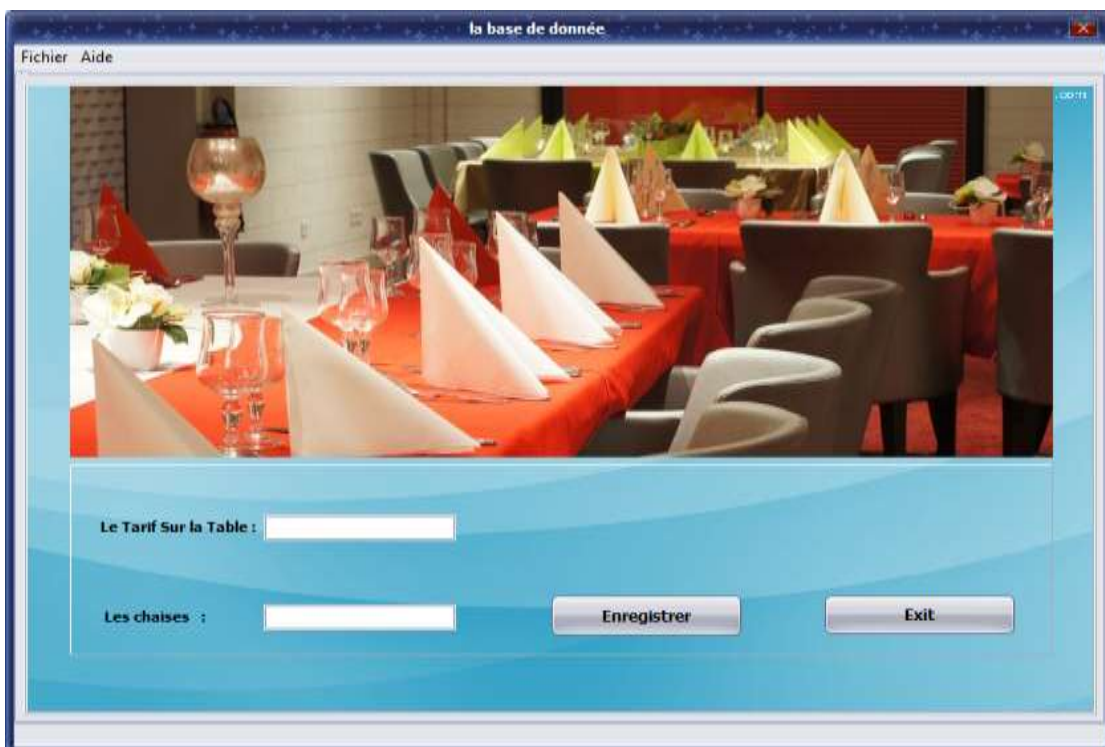
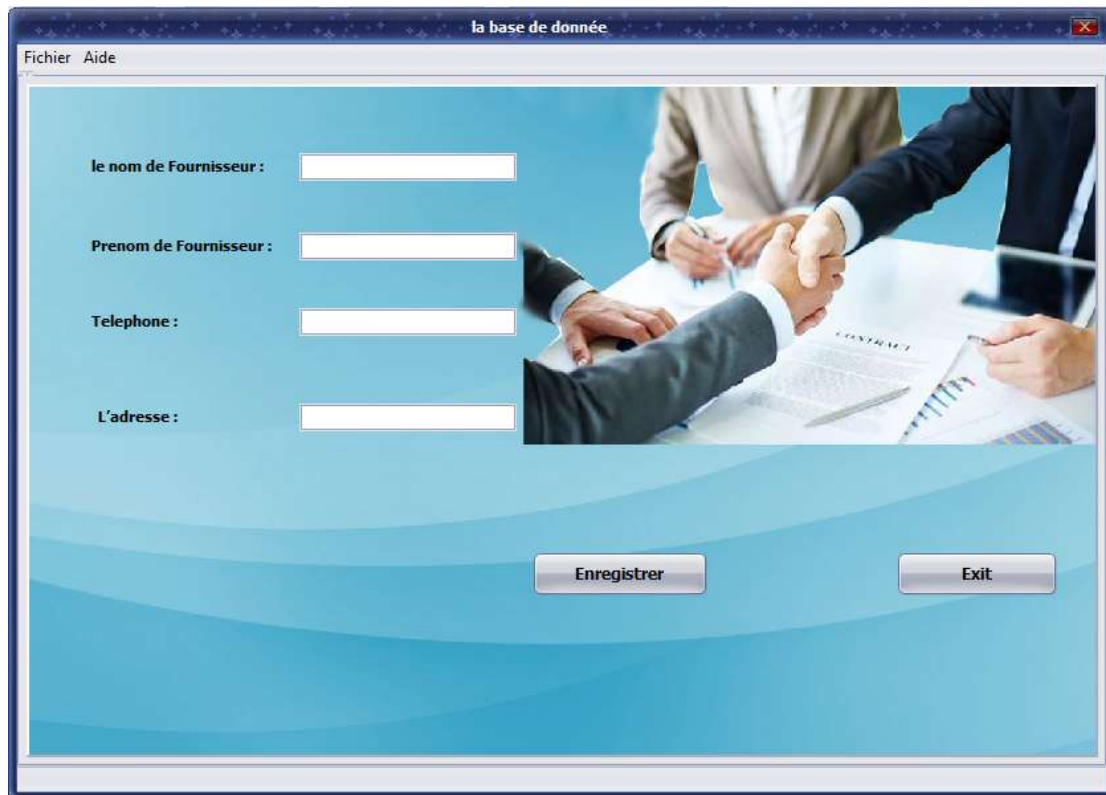


Figure 3-24: ajouter une table

L'interface qui permet de ajouter un fournisseur :



The screenshot shows a software window titled "la base de donnée" with a menu bar containing "Fichier" and "Aide". The main content area has a light blue background with a photograph of business people shaking hands over a contract. On the left side, there are four text labels with corresponding input fields: "le nom de Fournisseur :", "Prenom de Fournisseur :", "Telephone :", and "L'adresse :". At the bottom, there are two buttons: "Enregistrer" and "Exit".

Figure 3-25: ajouter un fournisseur

I Conclusion

Dans ce chapitre, nous avons créé une BDD sous SQL server et Delphi pour l'interface de l'application que nous avons présenté en détail et en expliquons chaque fenêtre et chaque instruction dans l'application.



Conclusion générale

Conclusion générale

L'objectif visé dans ce projet de fin d'études est la conception et réalisation d'une application informatique pour la gestion de restaurant, et Pour atteindre cet objectif, on a abordé notre problème en s'appuyant sur la démarche de Merise, en ce qui concerne la réalisation, on a utilisé le langage Delphi.

Cette application a permis de répondre aux besoins des utilisateurs par la résolution des problèmes liés à la gestion de restaurant, qui se considère comme l'objectif principal de ce travail.

En ce qui nous concerne, ce travail a été pour nous à la fois, un sujet de recherche et d'application qui nous a permis d'améliorer nos connaissances et nos compétences dans le domaine de la programmation, et d'affirmation et préparation à une intégration dans le monde professionnel. En effet cette expérience nous a permis de joindre l'utile à l'agréable en évaluant aussi bien les profondeurs théoriques que pratiques de ce vaste et passionnant domaine qu'est celui des bases de données dans un environnement de travail réel avec des besoins réels.

Le travail réalisé a apporté des solutions aux différents problèmes de gestion rencontrés au niveau des restaurants. Mais évidemment, ce travail étant une œuvre humaine, ce n'est pas un modèle parfait, c'est pourquoi nous restons ouverts à toutes les critiques et sommes prêts à recevoir toutes les suggestions et remarques tendant à améliorer davantage cette étude, étant donné que tout travail informatique a été toujours l'œuvre d'une équipe.

Références Bibliographique

- [1] - 2017. Définition : restaurant.
<http://www.restaurantsquifontamanger.fr/restaurant.htm>. [En ligne] Mai 2017.
- [2] - L. Baptiste, Merise Guide pratique: Modélisation des données et des traitements, manipulations avec le langage SQL, ENI éditions, 2009.
- [3] - Modélisations UML diagrammes structurels, Génie électrique et informatique industrielle, IUT Toulon Var.
- [4] - Méthodologies de développement de logiciels de gestion par P.-A. Sunier.
- [5] - Josef GABAY - Merise et UML pour la modélisation des d'information.
Volume 5. DunodEdition. Mars 2004.
- [6] - Pratique des systèmes d'information : MERISE, par : Mc BELAID et D.BOUYACOUB.
- [7] - Mémoire Online > Informatique et Télécommunications : Conception et réalisation d'une application web pour la gestion des stocks cas d'étude magasin de la faculté des sciences exactes de l'université de Bejaia par Laaziz LAHLOU. Université de Bejaia - Licence Académique en Mathématique et Informatique Option Informatique Générale 2010.
- [8] - Le langage SQL.
- [9] - Mémoire : Conception et réalisation d'une application de suivi de patients dans un établissement hospitalier.Par moghrani arezki et azzoug zoubir Université abederrahmane mira-bejaia - licence en informatique 2009.