

**2011**

Cindy Joseph & Guillaume Leroy

Polytech’Montpellier pour Neorizon représenté par M. Baïla

26/05/2011

Développement d’une application de gestion des titres restaurant



Rapport de

synthèse

Remerciements

Nous tenons à remercier toutes les personnes qui nous ont permis de mener à bien ce projet pendant ces deux mois :

* M. Sébastian Baïla, notre demandeur, pour sa disponibilité et son aide au bon déroulement du projet
* M. Jacques Ruiz, notre responsable pédagogique, qui nous a suivis régulièrement et conseillés tout au long de ce projet
* L’administration de Polytech’ Montpellier pour nous avoir fourni le matériel nécessaire à la bonne réalisation de ce projet.
* M. Marc Dumas, spécialiste de la communication, pour nous avoir aidés à préparer notre soutenance.

Sommaire

[1 Introduction 5](#_Toc294039506)

[2 Présentation de l’environnement 6](#_Toc294039507)

[2.1 L’entreprise 6](#_Toc294039508)

[2.2 Les Titres Restaurant 6](#_Toc294039509)

[2.2.1 Le mécanisme 6](#_Toc294039510)

[2.2.2 Le mode de fonctionnement 7](#_Toc294039511)

[3 Présentation du projet 8](#_Toc294039512)

[3.1 Le problème de gestion 8](#_Toc294039513)

[3.2 Les besoins fonctionnels 8](#_Toc294039514)

[3.3 La mission 8](#_Toc294039515)

[3.4 Contraintes 9](#_Toc294039516)

[3.4.1 Contraintes techniques 9](#_Toc294039517)

[3.4.2 Contraintes temporelles 9](#_Toc294039518)

[4 Déroulement du projet 10](#_Toc294039519)

[4.1 Gestion du projet 10](#_Toc294039520)

[4.2 Démarche 11](#_Toc294039521)

[4.2.1 Méthodes utilisées 11](#_Toc294039522)

[4.2.2 Choix technologiques 11](#_Toc294039523)

[5 Travail réalisé 12](#_Toc294039524)

[5.1 Analyse préalable 12](#_Toc294039525)

[5.1.1 Recueil des données 12](#_Toc294039526)

[5.1.2 Modèle conceptuel de traitements 13](#_Toc294039527)

[5.2 Conception 14](#_Toc294039528)

[5.2.1 Diagramme des cas d’utilisation détaillé 14](#_Toc294039529)

[5.2.2 Diagramme de classes 15](#_Toc294039530)

[5.2.3 Maquettage 16](#_Toc294039531)

[5.2.4 Architecture 16](#_Toc294039532)

[5.3 Développement 17](#_Toc294039533)

[5.3.1 Lecture des codes-barres 17](#_Toc294039534)

[5.3.2 Gestion des établissements 18](#_Toc294039535)

[5.3.3 Gestion des bordereaux 19](#_Toc294039536)

[5.3.4 Statistiques 23](#_Toc294039537)

[5.4 Tests 23](#_Toc294039538)

[5.5 Limites 24](#_Toc294039539)

[5.6 Améliorations possibles 24](#_Toc294039540)

[5.6.1 Impression d’un bordereau complet 24](#_Toc294039541)

[5.6.2 Sauvegarde de la BD 24](#_Toc294039542)

[5.6.3 Design des fenêtres 24](#_Toc294039543)

[6 Difficultés rencontrées 25](#_Toc294039544)

[6.1 Difficultés techniques 25](#_Toc294039545)

[6.2 Gestion du temps 25](#_Toc294039546)

[7 Conclusion 26](#_Toc294039547)

[Annexes 27](#_Toc294039548)

[Résumé 29](#_Toc294039549)

[Summary 29](#_Toc294039550)

Table des illustrations

[Image 2.2.2.1 : Processus d'utilisation des titres restaurant 7](#_Toc294039551)

[Image 5.1.2.1 : Modèle Conceptuel des Traitements 13](#_Toc294039552)

[Image 5.2.1.1 : Diagramme des cas d'utilisation 14](#_Toc294039553)

[Image 5.2.2.1 : Diagramme de classes 15](#_Toc294039554)

[Image 5.3.1.1 : Fenêtre du clavier tactile 17](#_Toc294039555)

[Image 5.3.2.1 : Vue Etablissement 18](#_Toc294039556)

[Image 5.3.3.1 : Vue Bordereau 19](#_Toc294039557)

[Image 5.3.3.2 : Vue Modification d’un bordereau 20](#_Toc294039558)

[Image 5.3.3.3 : Importation des titres de NeoResto 21](#_Toc294039559)

[Image 5.3.4.1 : Vue Statistiques 23](#_Toc294039560)

# Introduction

Dans le cadre de notre formation d’ingénieur en informatique et gestion, nous devons réaliser à la fin de la quatrième année un projet industriel en **collaboration avec une entreprise**. Ce projet a pour but de nous placer dans les conditions du monde professionnel.

Dans cette optique, nous avons choisi de réaliser le projet proposé par l’entreprise Neorizon. Notre choix s’est porté sur ce projet, car il nous permettait de réaliser **un projet dans son intégralité**. L’objectif général était de concevoir une solution permettant de gérer les titres restaurant qui étaient jusque-là traités manuellement.

Ce rapport vise à présenter de manière **synoptique** le travail accompli durant ce projet. Dans un premier temps nous présenterons le contexte général dans lequel nous avons évolué. Puis nous aborderons les objectifs attendus et le travail réalisé pour les atteindre. Nous décrirons ensuite la démarche suivie et les problèmes que nous avons rencontrés. Enfin, nous ferons un bilan sur le travail que nous avons effectué.

# Présentation de l’environnement

## L’entreprise

Neorizon est une société créée en 2006 et spécialisée dans l’équipement informatique des cafés, hôtels, restaurants (CHR) et des points de vente. Elle propose du matériel nécessaire à la gestion des établissements de type CHR mais aussi un logiciel NeoResto.

Le logiciel NeoResto (principal logiciel proposé par l’entreprise) permet de gérer les établissements de restauration. Il permet de prendre les commandes directement aux tables des clients et de les envoyer aux services concernés (bar, cuisine, grillardin, etc.). Ce logiciel propose des statistiques et des mises à jour des stocks. Il permet également une gestion flexible de la caisse, l’utilisateur peut choisir le moyen de paiement, gérer les remises et les partages de note. Le logiciel NeoResto permet de scanner les titres restaurant utilisés par les clients pour payer leurs notes mais seulement à titre d’information. Il ne propose donc pas d’outil approprié pour gérer les titres restaurant.

## Les Titres Restaurant[[1]](#footnote-1)

### Le mécanisme

Le système des titres restaurant a une finalité exclusive : répondre à un besoin social bien identifié, celui de la restauration des salariés et employés pendant la journée de travail. Sur ce plan, le succès rencontré auprès de ses utilisateurs est à mettre au compte des avantages qu’il apporte tant aux entreprises et à leur personnel salarié qu’aux professionnels des métiers de la restauration les acceptant comme moyens de paiement.

### Le mode de fonctionnement

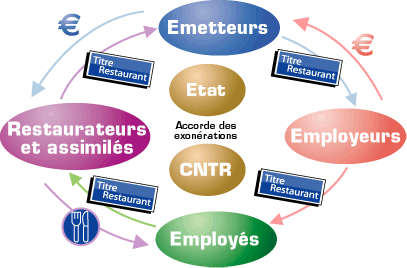


Image . : Processus d'utilisation des titres restaurant

Les émetteurs, sociétés spécialisées dans l’émission des titres restaurant, vendent aux employeurs des titres de différentes valeurs. Ils procèdent au remboursement des titres présentés par les restaurateurs et commerçants assimilés.

 Les employeurscèdent ces titres à leurs salariés en contrepartie d’une somme inférieure à leur montant. La différence correspond à la participation patronale. Cette participation bénéficie d’exonérations fiscales et de cotisations sociales.

 Les employés utilisent les titres-restaurant pour régler un repas ou acheter des préparations alimentaires chaudes ou froides qui doivent répondre à des critères prédéfinis.

 Les restaurateurs et commerçants assimilés peuvent accepter les titres en paiement à raison d’un seul titre par repas ou achat de préparations alimentaires. Il est interdit de rendre la monnaie sur un titre restaurant. Les restaurateurs et commerçants assimilés présentent les titres à la Centrale de Règlement des Titres (CRT).

# Présentation du projet

## Le problème de gestion

Actuellement l’entreprise Neorizon ne propose d’outil approprié pour gérer les différents types de titres restaurant. En effet, pour se faire rembourser les titres restaurant perçus, le restaurateur doit remplir un bordereau qu’il envoie à la CRT. Pour remplir ce bordereau, il doit donc trier, compter et classer les titres perçus manuellement afin de pouvoir être remboursé, ce qui peut être source d’erreurs et nécessite beaucoup de temps. La gestion de ces titres est donc aujourd’hui une réelle **perte de temps** pour la profession.

## Les besoins fonctionnels

L’objectif principal de ce projet était donc de créer une application permettant aux restaurateurs **d'automatiser** et de simplifier cette procédure de **gestion des titres**.

La solution que nous devions réaliser devait atteindre ces objectifs :

* Élaborer les bordereaux pour le remboursement de façon semi-automatique (en utilisant les informations saisies par l’utilisateur) et les imprimer
* Gérer les bordereaux : consultation, modification de leur état, archivage
* Présenter des statistiques simples concernant les données enregistrées à propos des titres

L’application réalisée pourra être utilisée en complément du logiciel NeoResto ou vendue séparément.

## La mission

Notre mission était donc d’analyser, de concevoir et de développer une application devant offrir les fonctionnalités suivantes :

* Saisie des titres restaurant à l’aide d’une douchette ou en saisissant le code à la main
* Elaboration de bordereaux pour le remboursement des titres restaurant
* Consultation et édition des bordereaux
* Gestion de l’état des bordereaux (En cours, imprimé, envoyé ou remboursé)
* Impression des bordereaux selon les standards de la CRT
* Historique des bordereaux et des titres
* Statistiques sur les titres restaurant (chiffre d’affaire, quantité, répartition), les bordereaux (suivi de l’état, écarts de remboursement)
* Paramétrage de l’application (ajout, modification ou suppression des établissements)

## Contraintes

### Contraintes techniques

Pour réaliser ce projet, il était demandé d’utiliser le Framework[[2]](#footnote-2) .NET dans sa version 2.0 avec Visual Studio 2008 et le langage de programmation objet C#, afin de garder une cohérence avec NeoResto.

Le logiciel est destiné à être utilisé sur des écrans tactiles, donc sans utilisation d’une souris. Il faut donc adapter l’interface à ce type d’utilisation, en fournissant des éléments graphiques plus gros, plus accessibles.

### Contraintes temporelles

Nous disposions pour réaliser ce projet d’un délai de 7 semaines à compter du 8 Avril 20111. Le planning prévisionnel a été établi comme suit dans la lettre de mission :

|  |  |
| --- | --- |
| **Début du projet** | 8 Avril 2011 |
| **Validation de la conception et des maquettes** | 21 Avril 2011 |
| **Validation de la première fonctionnalité** | 10 Mai 2011 |
| **Validation du logiciel** | 20 Mai 2011 |
| **Remise des dossiers au demandeur** | 24 Mai 2011 |
| **Remise des dossiers** | 26 Mai 2011 à 12h |
| **Soutenance de projet** | 30 Mai 2011 à 9h30 |

# Déroulement du projet

## Gestion du projet

Le projet a commencé avec le premier entretien que nous avons eu avec le demandeur. À partir des réponses qu’il nous a fournies nous avons élaboré notre lettre de mission qui définit le champ de l’étude, le point d’arrivée et les modalités du projet.

Nous avons ensuite, découpé le projet en plusieurs phases : l’analyse de l’existant, la conception, le développement et les tests. Puis nous avons divisé ces phases en tâches. Nous avons essayé d’évaluer de la façon la plus précise possible la charge de travail nécessaire à la réalisation de chaque tâche de ce projet. Pour chaque tâche nous avons assigné les ressources nécessaires, déterminé le(s) résultat(s) attendu(s) et évalué la charge de travail en jours-homme afin de déterminer les dates de début et de fin de tâches. Nous avons tenu compte dans ces estimations de notre méconnaissance du langage et des logiciels imposés par le demandeur. Nous avons prévu dans le planning une tâche pour l’apprentissage et allongé, notamment dans la phase de développement, le temps nécessaire à la réalisation de la phase.

Nous avons utilisé comme support de planification le logiciel Microsoft Project qui permet de planifier automatiquement les tâches en fonction des contraintes de début et de fin, qui prend en compte les jours fériés et les week-ends. Cet outil permet également de suivre le projet notamment grâce aux outils graphiques (diagramme de Gantt par exemple) qu’il met à disposition des utilisateurs.

Vous trouverez en annexe les deux diagrammes de Gantt sur lesquels nous nous sommes appuyés tout au long de ce projet : le premier présente les tâches telles que nous les avons planifiées au début du projet et le deuxième présente les tâches telles qu’elles se sont déroulées.

## Démarche

### Méthodes utilisées

Nous devions dans un premier temps réaliser l’analyse du système lié au titre restaurant afin de réaliser la base de données relationnelle qui allait contenir toutes les informations nécessaires à l’élaboration des bordereaux. Afin de réaliser cette étude nous avons décidé d’utiliser certains éléments de la **méthode Merise** car elle contient tous les éléments dont nous avions besoin afin d’aboutir à un bon résultat.

Afin de modéliser notre application nous avons utilisé l’Unified Modeling Language (**UML**) car le langage de programmation C#, imposé par le demandeur, est orienté objet.

Pendant la phase de développement nous avons adopté une **démarche itérative** en organisant des entretiens avec le demandeur lorsqu’une fonctionnalité était terminée. Ces rendez-vous réguliers nous ont permis d’être assez réactifs sur les modifications à apporter, notamment au niveau des interfaces utilisateurs qui ont beaucoup évoluées entre l’étape de maquettage et le développement.

### Choix technologiques

Si le langage de programmation et le Framework nous ont été imposés nous avons été chargés par le demandeur de **choisir un système de gestion de base de données** (SGBD). Pour les besoins de l’application à réaliser nous devions choisir un outil offrant les mêmes possibilités qu’un SGBD ordinaire, mais disponible dans une version embarquée. Nous nous sommes intéressés à plusieurs technologies et nous en avons retenues trois : PostGreSQL, Firebird et SQLite. Nous avons choisi Firebird car ce SGBD est disponible sous une version embarquée ce qui n’est pas le cas avec PostgreSQL. SQLite, qui lui est disponible sous cette forme, n’offre que des fonctionnalités minimales : l’intégrité référentielle[[3]](#footnote-3) n’est pas assurée et la modification du schéma de la base n’est pratiquement pas possible.

# Travail réalisé

## Analyse préalable

### Recueil des données

A partir des documents utilisés dans les processus de gestion des titres et de remboursement, nous avons recueilli les données suivantes et mis en évidence leurs interactions.

Chaque établissement est enregistré auprès de la CRT par un code affilié, il s’agit de son identifiant. Chaque établissement (restaurant) peut recevoir des titres restaurant en guise de paiement. Un titre restaurant est identifié par son code-barres. Il contient d’autres informations comme sa valeur faciale et sa date limite. Il existe quatre types de titre : les chèques déjeuner, les chèques de table, les tickets restaurant et les chèques restaurant.

Lorsqu’un établissement demande le remboursement des titres qu’il a perçus, il envoie à la CRT un bordereau accompagné d’une liasse constituée des titres qu’il souhaite se faire rembourser. Le bordereau contient les informations suivantes : le numéro de remise, la date de remise (date à laquelle le bordereau est envoyé), le montant total du bordereau, le nombre de titres pour chaque type et les neufs derniers caractères du code-barres d’un titre tiré au hasard dans la liasse par type de titre.

### Modèle conceptuel de traitements

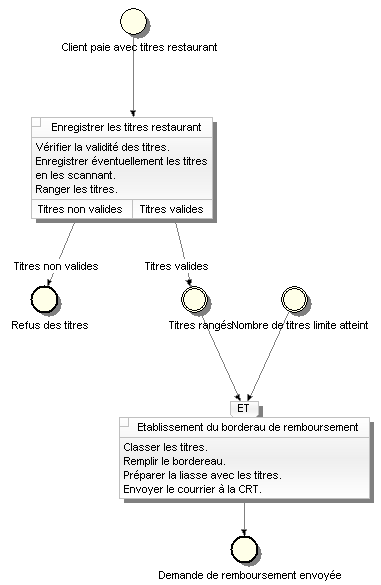


Image . : Modèle Conceptuel des Traitements

Lorsqu’un client paie avec un ou plusieurs titres restaurant, l’employé qui l’encaisse doit vérifier que le titre est valide (par exemple la date limite de validité). S’il a la possibilité de le faire il scanne les titres afin que les données soient enregistrées dans le logiciel de gestion de l’établissement (NeoResto par exemple). Enfin il range les titres.

Lorsqu’un certain nombre de titres est atteint, un employé se charge de remplir le bordereau de remboursement des titres. Il doit classer les titres par type afin de préparer la liasse qui sera envoyée avec le bordereau. Il complète le bordereau pré-rempli. Enfin il envoie la liasse et le bordereau à la CRT.

## Conception

### Diagramme des cas d’utilisation détaillé

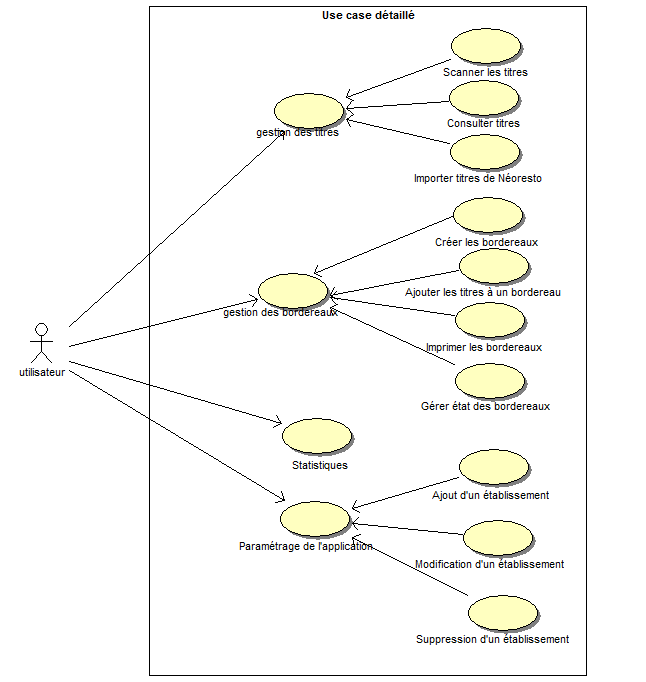


Image . : Diagramme des cas d'utilisation

La gestion des titres restaurant se découpe en quatre fonctionnalités. L’utilisateur pourra, avec cette application, créer des bordereaux, leur ajouter des titres en les scannant ou en les important et imprimer les documents. Il aura la possibilité d’ajouter ou de supprimer des établissements afin de gérer séparément les titres des différents établissements.

### Diagramme de classes

Après notre étude de l’existant, nous avons pu déterminer de quelles informations nous avions réellement besoin pour le bon fonctionnement de notre application.

Par exemple, pour les titres restaurant, toutes les informations nécessaires comme la valeur faciale, le type de titre (l’émetteur) se trouvent codés dans le code-barres. Les autres informations comme le nom du bénéficiaire et son adresse sont superflues.

Afin d’établir le bordereau, nous n’avons besoin que du code affilié de l’établissement. Les autres informations concernant l’établissement et le gérant ne sont présentes qu’à titre informatif.

Concernant le bordereau, nous avons besoin d’avoir des informations supplémentaires qui ne sont pas mentionnées sur le document comme par exemple date de remboursement, le montant remboursé et la raison de l’écart de remboursement s’il y en a un, afin d’assurer la partie statistiques de l’application.

En apportant ces modifications nous sommes arrivés au diagramme de classes suivant.

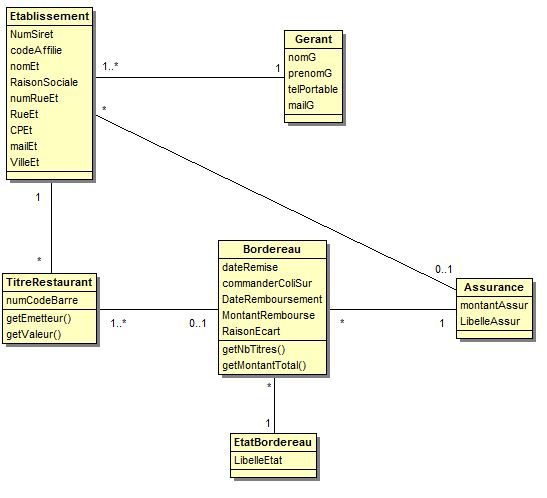


Image . : Diagramme de classes

### Maquettage

Nous avons **modélisé** toutes les interfaces graphiques durant la conception afin de valider l’enchainement des fenêtres et leur disposition. Cependant, celles-ci ont été amenées à évoluer grandement au cours du développement suite aux remarques du client concernant l’ergonomie. Cela nous a obligé à revoir une partie de la conception de nos fenêtres.

(Pour plus de détails sur cette évolution, se référer au rapport technique.)

### Architecture

L’architecture du logiciel est basée sur le design pattern architectural **Model View Controller** (MVC). Elle est représentée par le schéma suivant :

Vues

Classes

Contrôleur

Classes

Data

Base de données Firebird

Classes

Modèle

**Application .NET**

L’architecture peut être divisée en 2 grandes parties :

* La base de données Firebird : elle enregistre toutes les données utilisées par l’application
* L’application : Utilisant le framework .NET, elle interagit avec l’utilisateur et la base de données et effectue les traitements nécessaires

L’application est composée de 4 parties, articulées autour des classes de contrôle.

Côté utilisateur, les vues permettent à l’utilisateur d’interagir avec le logiciel. Plusieurs vues sont proposées, chacune ayant un ou plusieurs contrôleurs associés. Ces contrôleurs assurent l’interface entre les classes métiers (modèle), les classes de données (data) effectuant les requêtes vers la base de données, et les vues.

## Développement

### Lecture des codes-barres

L’une des premières fonctionnalités à avoir été développée est la lecture des codes-barres des titres restaurants à l’aide d’une douchette, et leur enregistrement dans le logiciel.

La douchette se comporte comme une entrée standard, comparable à un clavier, branché sur un port USB. Lorsque l’on scanne un titre, la chaine de caractères est lue et envoyée à l’ordinateur sur l’entrée standard.

Il fallait également prévoir la possibilité qu’un code-barres ne puisse être lu (en cas de défectuosité de celui-ci, par effacement partiel par exemple). Pour se faire, nous avons développé un clavier virtuel muni des 10 chiffres et de boutons d’annulation, de suppression et de validation. L’utilisateur peut alors rentrer la chaine de caractère ainsi.

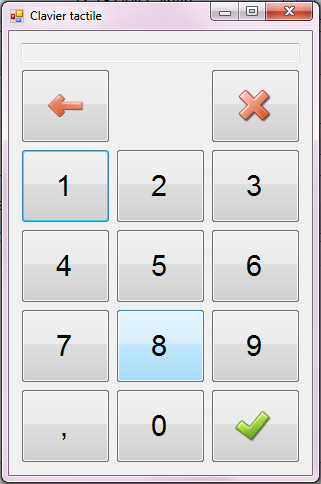


Image . : Fenêtre du clavier tactile

La lecture du code-barres est validée lorsque 26 caractères ont été saisis (24 caractères qui seront interprétés, auxquelles il faut ajouter 2 caractères de début et de fin non interprétés), que ce soit à l’aide de la douchette ou du clavier tactile.

On vérifie ensuite que le titre n’est pas déjà enregistré dans la base de données. Si c’est le cas, on avertit l’utilisateur et le titre est rejeté, sinon, on contrôle sa validité à l’aide de la clé de contrôle contenue dans le code-barres.

Une fois tous ces contrôles effectués et validés, le titre est accepté et ajouté en base de données. L’utilisateur peut alors en saisir un autre.

L’enregistrement de titres par scan peut se produire dans deux cas, définissant deux scénarii distincts :

* Dans le cas d’un scan libre, l’utilisateur scanne des titres pour un établissement sans l’associer à un bordereau. Le titre est donc lié à un établissement mais n’est attribué à aucun bordereau : il reste libre.
* Dans le cas d’un ajout de titres à un bordereau, le titre est ajouté en étant lié à un bordereau.

### Gestion des établissements

L’utilisateur a la possibilité de gérer les titres de plusieurs établissements à la fois. Il peut donc ajouter, supprimer ou modifier des établissements. Quand l’utilisateur enregistre un nouvel établissement on s’assure que le numéro Siret, le code affilié et le nom de l’établissement (utilisé comme identifiant) soient bien renseignés.

Lorsque l’utilisateur clique sur le bouton « Etablissement », l’écran ci-dessous apparait. Par défaut, tous les établissements enregistrés dans la base sont listés.

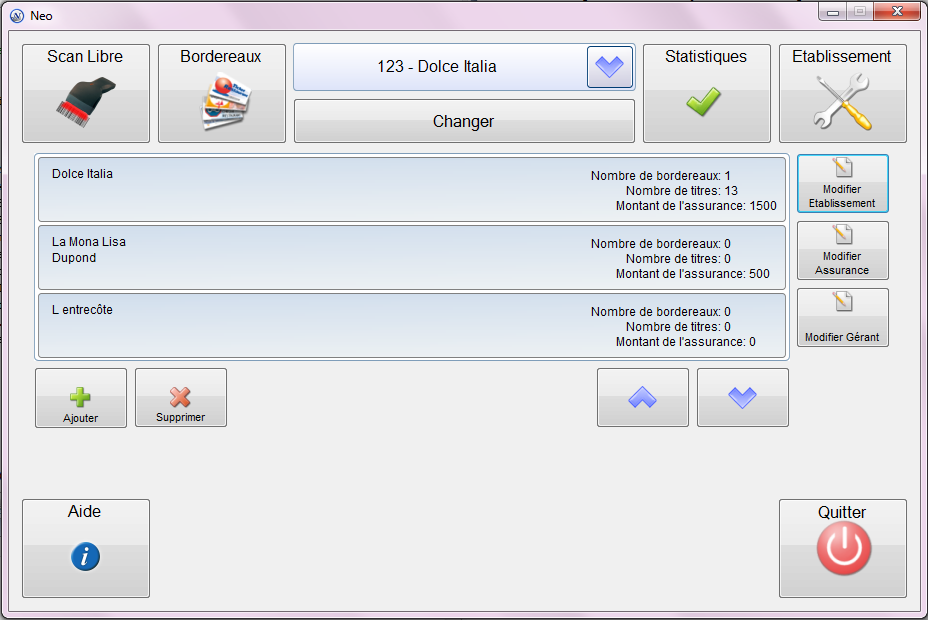
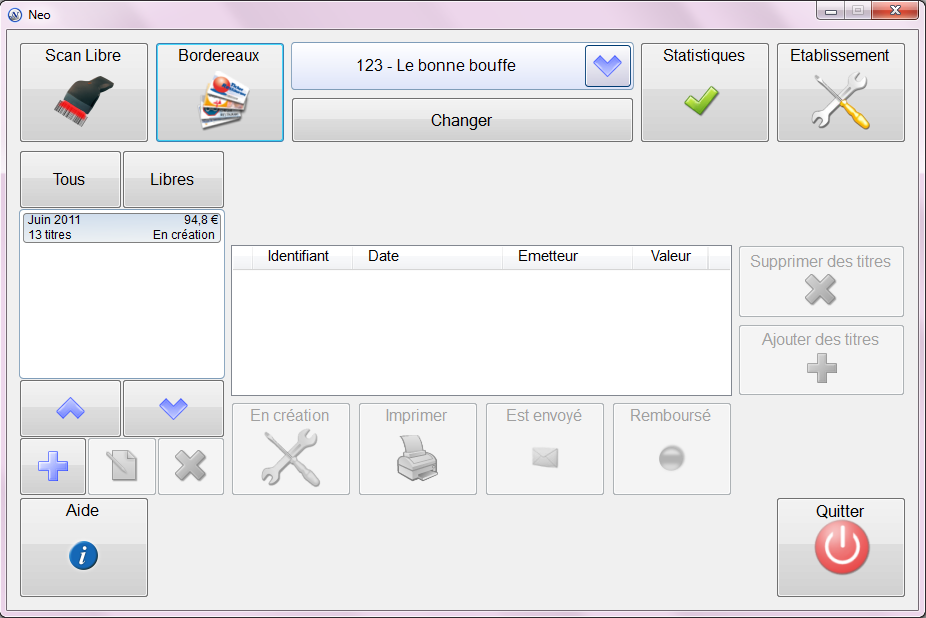


Image . : Vue Etablissement

### Gestion des bordereaux

La gestion des bordereaux est le cœur même du logiciel. En effet, le but de cette application est d’enregistrer rapidement les titres restaurant perçus, de créer les bordereaux et de les imprimer sans perdre le temps de devoir les manipuler manuellement.



**Image 5.3.3.1 : Vue Bordereau**

#### Création d’un bordereau

A partir de la vue « Bordereaux », l’utilisateur peut créer un nouveau bordereau en cliquant sur le bouton « + » sous la liste des bordereaux. La fenêtre ci-dessous s’ouvre pour en saisir le nom. Si celui-ci est inédit, le bordereau est créé dans l’état « En création ».

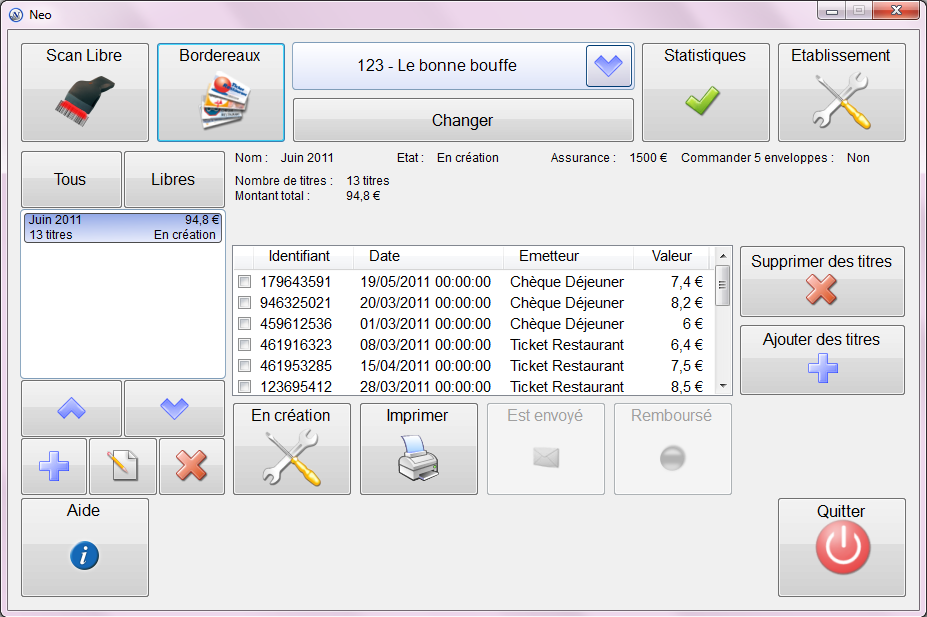
#### Affichage d’un bordereau

Lorsque l’utilisateur sélectionne un bordereau dans la liste, toutes ses informations sont affichées dans la fenêtre. Certaines informations ne sont affichées que si le bordereau est dans un certain état. Par exemple, la date de remboursement n’est affichée que si le bordereau a pour état « Remboursé ».

#### Modification d’un bordereau

Un bordereau n’est modifiable que s’il est en création.

En cliquant sur le bouton d’édition en dessous de la liste des bordereaux, la fenêtre ci-dessous s’ouvre permettant de modifier le nom du bordereau et de choisir ou non de commander 5 enveloppes (cette information est présente sur les bordereaux de la CRT par une case à cocher).



**Image 5.3.3.2 : Vue Modification d’un bordereau**

#### Ajout et suppression de titres à un bordereau

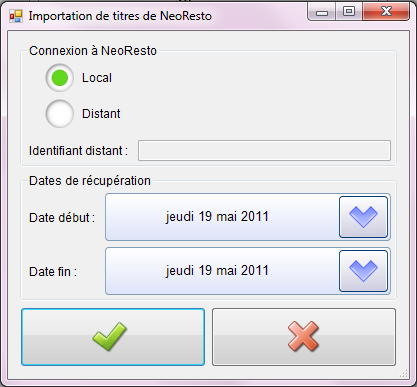
Il n’est possible d’ajouter des titres à un bordereau que s’il est en création.

Pour supprimer des titres d’un bordereau, l’utilisateur les sélectionne en cliquant dans la liste des titres (les titres à supprimer doivent être cochés) puis clique sur le bouton « Supprimer des titres » à droite de cette liste. Un message pop-up demande alors s’il faut conserver les titres à l’état libre (non liés à un bordereau) ou bien les supprimer. Selon le choix de l’utilisateur, les titres sont ensuite désaffectés ou supprimés.

Pour en ajouter, il suffit de cliquer sur le bouton « Ajouter des titres ». Une fenêtre offre 3 possibilités :

* Scanner des titres – Cela ouvre la fenêtre de scan pour enregistrer de nouveaux titres qui seront automatiquement liés au bordereau.
* Choisir des titres libres – Cela ouvre une fenêtre affichant tous les titres sans bordereau. L’utilisateur sélectionne les titres à lier et valide sa sélection.

Importer des titres – Cela ouvre une fenêtre de connexion à NeoResto permettant d’en importer des titres, enregistrés entre deux dates définies par l’utilisateur.



**Image 5.3.3.3 : Importation des titres de NeoResto**

#### Changement d’état d’un bordereau

Un bordereau peut être dans 4 états différents :

* En création : le bordereau a été créé, il est encore possible de changer ses informations et de modifier les titres attachés.
* Imprimé : le titre a été imprimé, plus aucune modification n’est possible à partir de cet état. En cliquant sur le bouton « Imprimer », une fenêtre s’ouvre pour gérer l’impression du bordereau puis l’état est passé à « Imprimé ».
* Envoyé : le bordereau a été envoyé à la CRT, une date d’envoi est définie.
* Remboursé : le remboursement pour ce bordereau a été reçu de la CRT, les informations relatives à ce remboursement (le montant remboursé et la raison d’un éventuel écart) sont indiquées à l’aide de la fenêtre qui s’ouvre en cliquant sur le bouton « Remboursé ».

Le passage d’un état à l’autre se fait séquentiellement dans l’ordre indiqué ci-dessus. Cependant, afin de laisser plus de flexibilité à l’utilisateur, il est possible de revenir à un état précédent à tout moment. Pour modifier un bordereau, il faut absolument que son état soit « en création ».

#### Suppression d’un bordereau

À tout moment, l’utilisateur peut supprimer un bordereau en le sélectionnant dans la liste et en cliquant sur le bouton « X ». Un message demande alors s’il faut conserver libres les éventuels titres associés ou bien les supprimer. Selon le choix, les titres sont désaffectés ou supprimés, puis le bordereau définitivement éliminé.

### Statistiques

L’application présente des statistiques simples calculées à partir des informations enregistrées dans la base de données.

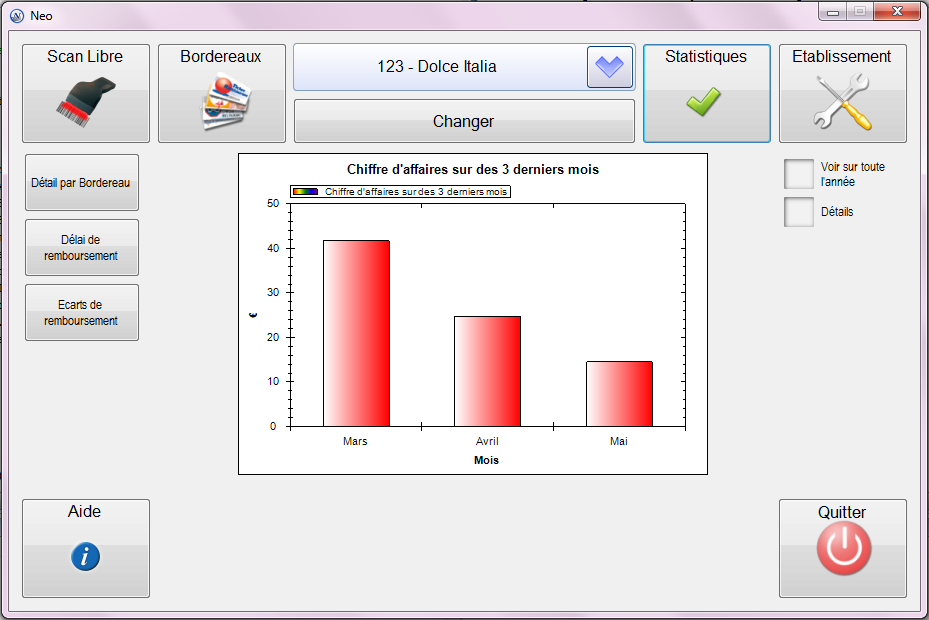


Image . : Vue Statistiques

Par défaut, le chiffre d’affaires des trois derniers mois de l’établissement courant est affiché (dans l’exemple c’est le « Dolce Italia »). Il est possible d’obtenir le chiffre d’affaires sur toute l’année et/ou le chiffre d’affaire ventilé par type de titre restaurant.

D’autres informations plus détaillées sont disponibles : la répartition du chiffre d’affaires par type de titres pour un bordereau choisi par l’utilisateur, le délai de remboursement moyen sur une période donnée et l’écart de remboursement sur chaque mois.

## Tests

Nous avons découpé le développement par fonctionnalité. À chaque fois qu’une fonctionnalité était réalisée elle faisait l’objet de **tests unitaires** afin de vérifier si les résultats obtenus étaient corrects.

Lorsqu’une fonctionnalité était terminée nous l’intégrions au reste de l’application en s’assurant que **l’intégration** ne provoquait pas de **régression** des autres fonctionnalités déjà intégrées.

## Limites

L’application que nous avons réalisée ne fonctionnera que pour des titres du millésime 2011 car les contrôles et les calculs effectués sur le code barre ne sont valables que pour ce millésime et sont susceptibles de changer les années suivantes. Cependant, nous avons commenté le code et tenté de rendre l’application la plus **maintenable** possible. Donc il sera aisé de changer le code à modifier qui ne concernera que deux classes : « ControleurTitre » et « Titre ».

## Améliorations possibles

### Impression d’un bordereau complet

Le logiciel permet actuellement de remplir un bordereau en **imprimant** directement les informations utiles sur un bordereau officiel de la CRT. Une amélioration possible serait d’offrir la possibilité d’imprimer un **bordereau complet** sur une page vierge. Il ne serait ainsi plus utile de se procurer des pré-imprimés.

### Sauvegarde de la BD

La version embarquée de Firebird enregistre toutes les données d’une base de données dans un fichier unique d’extension « .fdb ». Une suppression malencontreuse de ce fichier provoquerait la perte de toutes les données de l’application.

Il faudrait donc donner la possibilité à l’utilisateur d’effectuer des sauvegardes à la demande ou régulièrement de sa base de données afin d’éviter ce problème. Une interface pour restaurer les différentes versions de la base de données devrait également être envisagée.

### Design des fenêtres

Nous nous sommes efforcés tout au long de ce projet de rendre les visuels agréables en utilisant des images fournies par notre demandeur. Cependant, il reste des modifications à apporter au niveau des couleurs et de la présentation de l’interface graphique, afin qu’elle soit plus cohérente avec celle de NeoResto et plus agréable à l’œil.

# Difficultés rencontrées

## Difficultés techniques

Pendant ce projet, nous avons été amenés à manipuler des technologies que nous ne connaissions pas (Visual Studio, Framework .Net, C# et Firebird).

Nous avons dû faire face à un problème assez important au début de la phase de développement : l’intégration de Firebird à l’environnement de Visual Studio. En effet, notre application reposant sur une base de données, nous avions prévu d’utiliser le type DataSet fourni par le langage. Il fallait donc intégrer la base de données Firebird à Visual Studio pour en extraire directement les données (direct binding), ce qui n’a pas été évident à cause du **manque d’informations** et de documentation, et des nombreuses manipulations délicates nécessaires (modification de la base des registres, corrections de codes, édition de fichiers système, etc.).

Nous avons tout de même **réussi** l’intégration, mais nous n’avons finalement pas retenu cette solution : la version embarqué de Firebird permet en effet de s’y connecter à l’aide d’une chaine de connexion et d’y envoyer les requêtes à l’aide de méthodes C#. Nous avons donc créé une classe pour gérer cette connexion, d’autres pour gérer les échanges entre les classes logicielles et les tables de la base de données.

Cependant l’utilisation des technologies citées auparavant ne nous a pas posé d’autres problèmes par la suite.

## Gestion du temps

Des problèmes de temps ont logiquement découlés des soucis techniques que nous avons rencontrés. La recherche d’une solution pour intégrer Firebird à VisualStudio nous a occupés une semaine quasiment à plein temps, au lieu des quelques heures initialement prévues.

De plus, les modifications voulues par le demandeur sur les interfaces durant la phase de développement nous ont obligés à revoir notre conception des fenêtres et à recoder ces dernières, engendrant une charge de travail supplémentaire estimée à 3 jours-homme.

# Conclusion

Ce projet est le premier que nous avons eu à mener pour une entreprise. Durant ces 7 semaines entièrement consacrées à ce projet, nous avons pu mettre en œuvre une **méthodologie complète**, en planifiant nos tâches et en organisant notre temps afin de répondre aux attentes du demandeur.

Cette relation directe avec le client nous a permis de rapidement identifier les problèmes et les modifications à apporter. Nous avons également réalisé qu’il est très difficile de mettre en évidence toutes les attentes de la maîtrise d’ouvrage dans un premier temps, et que cela provoque bien souvent quelques remises en question au cours du projet.

Les échanges fréquents avec M. Baïla ont permis une grande réactivité. Il nous a également transmis quelques connaissances au niveau de l’ergonomie des IHM tactiles et du framework .NET.

Au niveau technique, nous avons dû maitriser l’environnement de travail de Microsoft et ses différents composants que sont Visual Studio et le langage objet C#. Nous avons également appréhendé les spécificités liés au développement d’une application destinée à des interfaces tactiles.

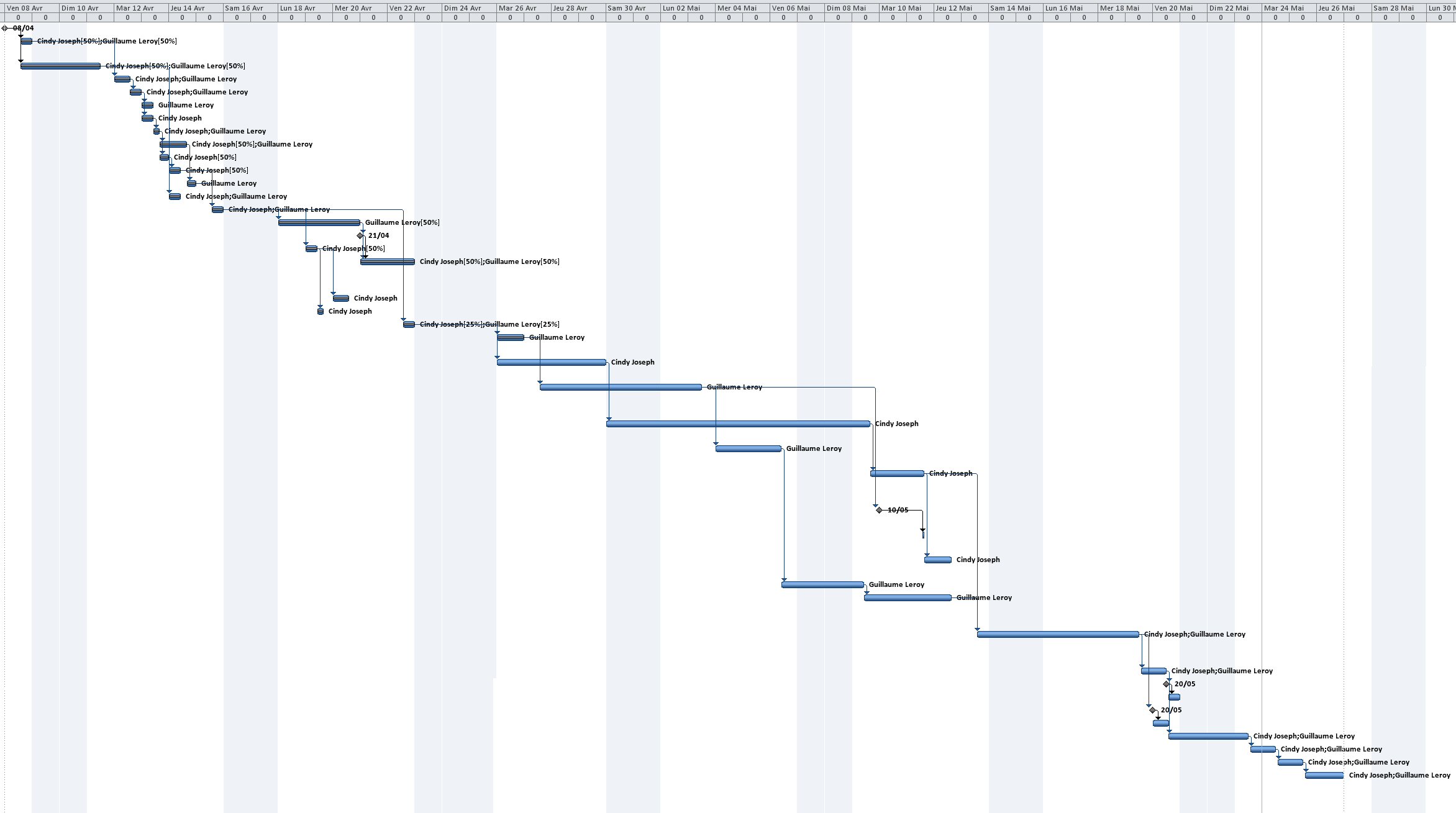
Tout au long de ce projet, nous avons pu être **force de proposition**, notamment sur le choix des statistiques à proposer et sur certaines fonctionnalités du logiciel.

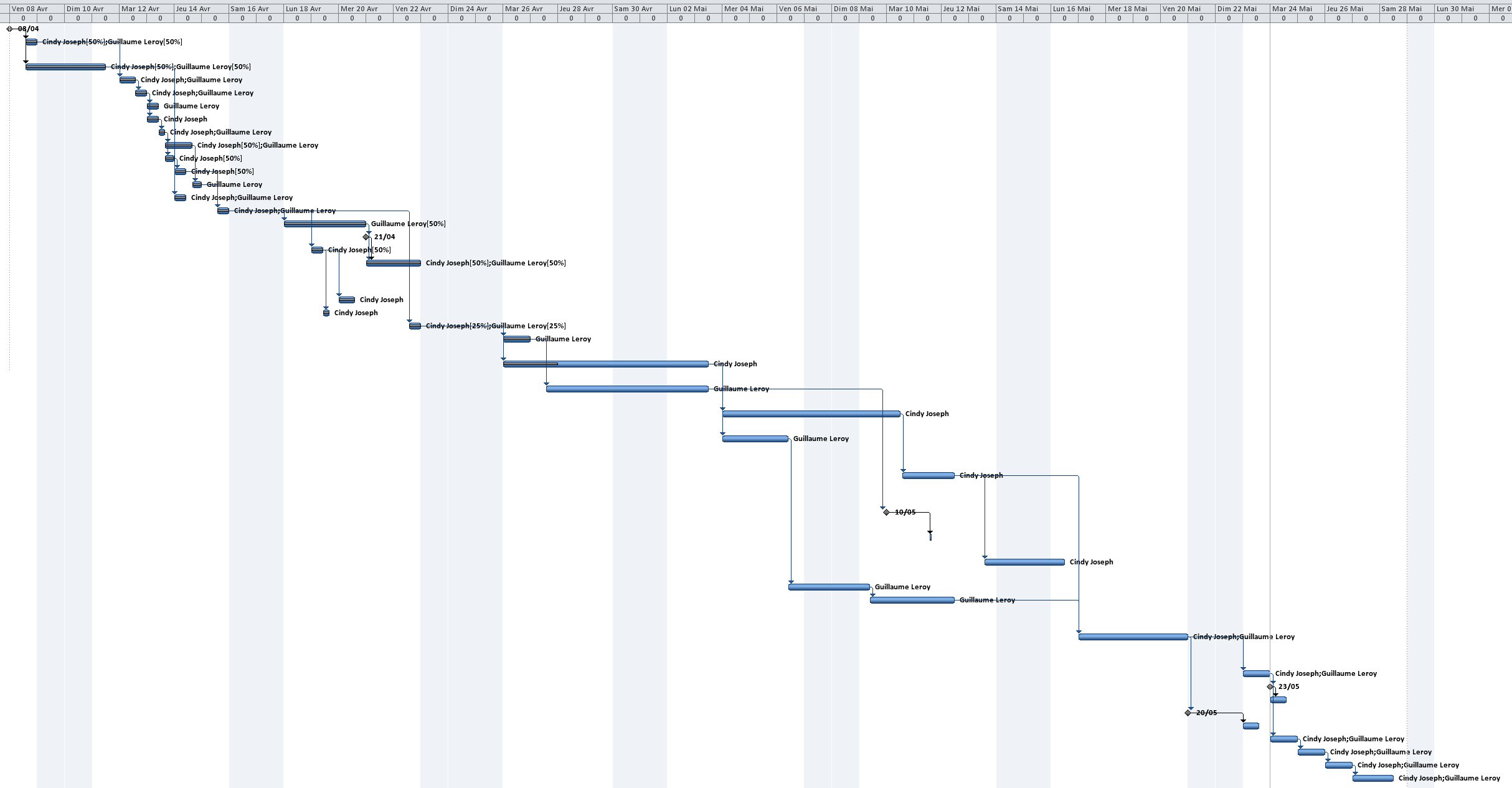
Grâce à ce projet, nous avons pu nous rendre compte qu’il est important de se mettre dans la peau de l’**utilisateur**, d’adopter son **point de vue** afin d’éviter certaines erreurs d’ergonomie. En tant que concepteur, nous avons toujours l’impression que le produit est facile à utiliser, alors qu’une personne non initiée peut se sentir rapidement perdue et submergée d’informations, ou alors incapable de trouver la bonne interface pour faire ce qu’il veut. En essayant de se mettre à sa place, on évite la plupart de ces écueils, et c’est ce que ce projet nous a également apporté.

Nous pouvons dire que nous avons **atteint les objectifs définis** dans la lettre de mission. En outre, nous avons identifié quelques améliorations qu’il serait intéressant de mettre en œuvre afin de perfectionner le logiciel.

Ce travail de groupe de 7 semaines a été très enrichissant tant sur le plan humain que sur le plan professionnel.

# Annexes

 **Annexe 1 – Diagramme de Gantt prévisionnel**

  
Annexe 2 – Diagramme de Gantt réel

# Résumé

L’entreprise Neorizon est une société proposant des solutions afin de gérer des établissements de type café, hôtel ou restaurant. Leur logiciel ne proposait pas d’outils pour la gestion des titres restaurant qui est la cause d’une grande perte de temps pour la profession.

Dans le cadre de notre projet industriel, nous avions pour mission de réaliser une application permettant de pallier ce problème de gestion. L’objectif principal de ce logiciel était de pouvoir automatiser le traitement des titres et l’élaboration des bordereaux de remboursement. Pour commencer, nous avons effectué une analyse de l’existant en utilisant la démarche Merise puis nous avons conçu l’application en la modélisant à l’aide des diagrammes UML. Nous avons adopté une démarche itérative pour le développement afin d’être le plus réactif possible.

Les objectifs que nous nous sommes fixés ont été atteints, toutefois il reste possible d’apporter des améliorations au niveau de la sauvegarde de la base de données, des options d’impression et de la charte graphique.

# Summary

Neorizon is a small company which sells computer equipment and offers the software NeoResto. This software allows to restaurant owners to manage their whole establishment but it doesn’t really handle restaurant ticket. The management of these restaurant tickets is an actual waste of time for this profession.

On the occasion of the industrial project, we had to make software which compensates this management problem. Our objective was to automate the sheet drafting. Firstly we carry out the environment analysis using some elements of the Merise method. Then we design the application with Unified Modeling Language. We adopt an iterative approach for the development part in order to be as responsive as possible.

We reach the goal we have fixed, however the software can be improved in particular on the database maintenance, on the printing options and on the graphical user interface design.

1. Informations tirées du site officiel de la Commission Nationale des Titres Restaurant [http://www.cntr.fr](http://www.cntr.fr/) [↑](#footnote-ref-1)
2. Framework (cadre d’applications en français) : kit de [composants logiciels](http://fr.wikipedia.org/wiki/Composant_logiciel) structurels utilisé pour modeler l’architecture des logiciels applicatifs. [↑](#footnote-ref-2)
3. Intégrité référentielle : Mécanisme participant à assurer la cohérence d’une base de données en vérifiant que toute information d’une table A faisant référence à une information d’une table B existe bien dans la table B. Par exemple, un livre ne pourra être associé à un auteur qui n’existe pas. [↑](#footnote-ref-3)