



Projet	Back-Office Commercial
Document	Cahier des charges
Objet	Développement d'une application de gestion des clients, commandes et produits
Rédacteur	Gilles RESSOUCHE
Date	02/12/2014
Version	1.0

1. Définition du besoin

GSB souhaite avoir un outil centralisé (utilisable uniquement au siège de la société) permettant gérer les produits, les clients et les commandes. Les fiches de frais ne sont pas concernées par cet outil.

2. Contraintes techniques

2.1 Généralités

Le logiciel de back-office devra être développé en C#. Il sera lié à une base de données SQLServer.

La gestion des configurations sera faite pour chaque équipe par l'utilisation de TeamFoundationServer.

2.2 Programmation

Afin de faciliter la maintenance et les futures évolutions, l'application sera développée en objet. L'architecture sera de type MVC, avec utilisation du modèle DAO.

Les normes de développement seront celles habituellement utilisées par le prestataire.

Au lancement de l'application, les données seront chargées, à partir de la BD, dans le programme sous forme d'objets. Le programme travaillera ensuite uniquement avec les objets.

3. Données

3.1 Les produits

Les produits distribués par le laboratoire sont des médicaments : ils sont identifiés par un numéro de produit (dépôt légal) qui correspond à un nom commercial (ce nom étant utilisé par les visiteurs et les praticiens).

Comme tout médicament, un produit a des effets thérapeutiques et des contre-indications.

On connaît les interactions qu'il peut avoir avec d'autres médicaments (éléments nécessaires à la présentation aux médecins).

La présentation et le dosage sont aussi des données conservées.

Un produit relève d'une famille (antihistaminique, antidépresseur, antibiotique, ...). Ces familles doivent être codifiées dans la base.

Afin de pouvoir être facturé, le prix de vente HT est connu dans la base.

Lors d'une visite auprès d'un médecin, un visiteur présente un ou plusieurs produits pour lesquels il pourra laisser des échantillons. De manière à pouvoir gérer les coûts des visites, le prix de l'échantillon est aussi une donnée à retenir dans la base.

3.2 Les prospects et clients

Les prospects sont les professionnels de la santé qui ont été visités par GSB. On souhaite conserver l'historique des prospects, ainsi que les données des clients, dont les visites :

- Code
- Raison sociale
- Adresse
- Type de professionnel : médecin, dentiste, infirmière etc... ; ces types de professionnels seront codifiés dans la base
- Téléphone
- Contact (nom + adresse mail)
- Date de chaque visite et visiteur médical concerné ; les visites sont aussi conservées dans la base

Un professionnel est d'abord démarché par GSB, c'est alors un prospect. Il devient éventuellement client (mais ce n'est pas toujours le cas) lorsqu'il passe effectivement une commande à GSB.

3.3 Les commandes

Une commande peut concerner un ou plusieurs produits. Les données d'une commande sont les suivantes :

- Numéro de commande
- Date de commande
- Produit(s) commandé(s) avec leur quantité
- Etat de la commande (en cours, livrée, facturée) ; ces états seront codifiés dans la base.

4. Fonctionnalités

Le nouvel outil de back-office devra permettre de gérer toutes les données évoquées précédemment, ainsi évidemment que les données déjà présentes auparavant dans la base.

Les données de base

Ce sont les données qui n'évoluent pas (ou peu) dans le temps, qui concernent les produits, les visiteurs, les prospects/clients. On doit pouvoir les visualiser et les modifier. La suppression éventuelle sera virtuelle : on positionnera une date de suppression dans la base (pas de suppression physique). Les données supprimées seront visibles mais non modifiables dans l'application.

Les codifications de familles de produits, d'états de facture et de type client seront créés directement dans la base : il n'est pas prévu dans l'application de les créer ou modifier.

Les données évolutives

Les commandes

Ce sont les commandes passées par les clients. L'application doit permettre de saisir une commande pour un client. Pour une commande, on sélectionne des produits, et on saisit une quantité pour chaque produit commandé. Lors de la saisie, l'état de la commande sera « en-cours ».

La commande passera ensuite à l'état « expédiée », puis « livrée » puis « facturée » ; l'application permettra de modifier l'état de la commande et de visualiser chaque état avec la date correspondante.

Les visites

Lorsqu'un visiteur médical se déplace chez un praticien, il saisit la date de la visite, ainsi qu'un compte-rendu succinct.

5. Organisation du développement

Les fonctionnalités seront réparties au sein de chaque équipe. Le module d'accès à la base de données (classe *DAOFactory*) sera commun à tous les développeurs de l'équipe. Idem pour la base de données.