



SIVY
Société d'informatique de Vichy

- Logiciel de Facturation -

Présentation de la société cliente

A la création de VICHY PAPETERIE en 1983, Pierre DUVAL avait un souhait, celui d'apporter à ses clients le meilleur matériel et surtout, « la solution » qui convenait le mieux à leur entreprise. Elle est spécialisée dans :

- Les fournitures de bureau.
- Les consommables informatiques.
- La papeterie.
- Les fournitures scolaires.

Son territoire est essentiellement l'Allier (03) et surtout l'axe Vichy Moulins et le nord du Puy de Dôme (63). Avec la logistique « Calipage », VICHY PAPETERIE est également capable de livrer toute la France sous 24 heures.



Fournitures de bureau



Consommables



Papeterie



Dessin / Ecole Loisirs Créatifs



Au cours des années précédentes, VICHY PAPETERIE a confié l'informatisation de sa gestion à une EURL (Entreprise unipersonnelle à responsabilité limitée) de Cusset. Cette entreprise a développé, entre autres :

- Son site Internet (=site vitrine).
- Un environnement Intranet (HTML, CSS, PHP et MySQL) lui permettant de gérer ses articles, fournisseurs, clients, puis d'établir des devis et des factures.

L'EURL a depuis cessé son activité.

L'année dernière, un étudiant du BTS SIO, avait développé une application autour d'une IHM Windows en C# et MySQL, permettant de gérer le retour d'articles défectueux. Ceci a donné l'idée au gérant de VICHY PAPETERIE de profiter du besoin d'effectuer la maintenance corrective et évolutive de l'application Intranet existante, pour migrer vers une interface Windows et inclure ainsi l'application développée par le stagiaire à la future application.

VICHY PAPETERIE s'est adressé donc à la SSII SIVY dans le but de demander le développement d'une fonctionnalité de la future application. Ceci doit lui permettre de pouvoir comparer les deux technologies (web et Windows), puis de prendre la décision définitive de migrer l'intégralité de l'application Intranet dans une plateforme Windows / C#.

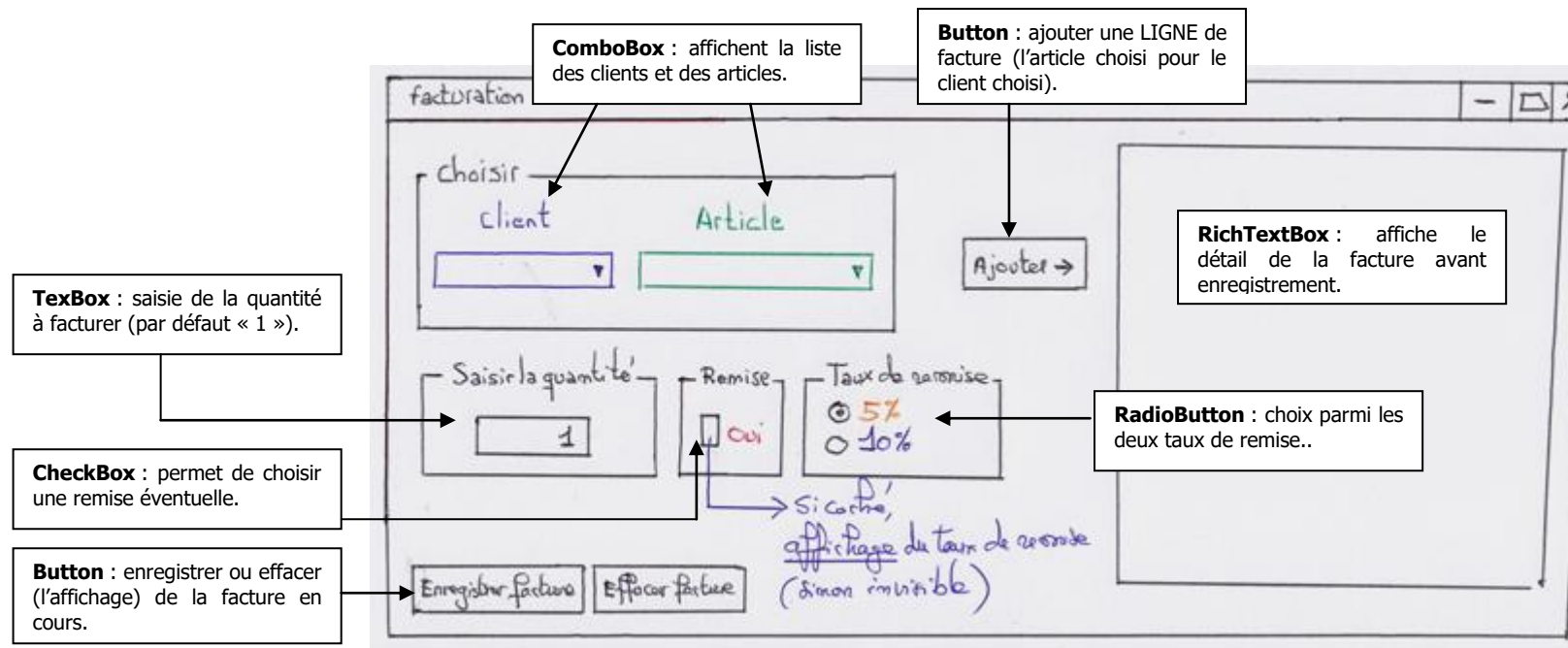


La fonctionnalité « pilote » choisie a été la FACTURATION.

CAHIER DE CHARGES

Un cahier de charges a été établi entre le gérant de VICHY PAPETERIE et le responsable du pôle développement de SIVY. Il présente à la fois les besoins utilisateurs (IHM, règles de gestion et fonctionnement attendu) et techniques.

I) INTERFACE – PRINCIPE DE FONCTIONNEMENT



Exemple de fonctionnement : choix d'un premier produit pour le client « Dupond et Fils ». Un clic sur « Ajouter » produit un premier affichage de la facture (une seule ligne de facture). Le fait de choisir un deuxième article, un clic sur « Ajouter » ajoute une nouvelle ligne à la facture et met à jour le total.

1 Dupond et Fils ▾ 2 Trombones 77mm - boîte*100 5,29€ ▾

+ Quantité: 10
+ Remise: oui → Taux 5% } Ajouter →

1 Agrafeuse grande puissance 179€ ▾

+ Quantité: 1
+ Remise: Non } Ajouter →

FACTURE : 74			
CLIENT : 1		Dupond et Fils	
2	Trombones 77 mm - boîte*100	10	Total ligne : 55 €
TOTAL A PAYER : 55 €			

FACTURE : 74			
CLIENT : 1		Dupond et Fils	
2	Trombones 77 mm - boîte*100	10	Total ligne : 55 €
1	Agrafeuse grande puissance	1	Total ligne : 179 €
TOTAL A PAYER : 234 €			

II) SOURCES DE DONNEES.

Il vous est demandé de travailler sur les DEUX sources de données suivantes (l'utilisateur aura la possibilité de les choisir) :

- Une BD MySQL.
- Un SERVICE WEB.

On vous fournit (suivant le cas) les données de TEST concernant les ARTICLES et les CLIENTS :

a) BD MySQL

```
DROP TABLE IF EXISTS Client;
CREATE TABLE Client
(
  code_client INT(3),
  raison_sociale VARCHAR(50) NOT NULL,
  CONSTRAINT pk_client PRIMARY KEY (code_client)
) ENGINE=INNODB;

DROP TABLE IF EXISTS Article;
CREATE TABLE Article
(
  code_article INT(3),
  designation VARCHAR(50) NOT NULL,
  prix_unitaire FLOAT DEFAULT 0,
  CONSTRAINT pk_article PRIMARY KEY (code_article)
) ENGINE=INNODB;
```

```
mysql> select * from article;
+-----+-----+-----+
| code_article | designation                | prix_unitaire |
+-----+-----+-----+
| 1 | Agrafeuse grande puissance | 185 |
| 2 | Trombones 77 mm boite*100 | 6.5 |
| 3 | Correcteur a sec roller 4,2 | 6.9 |
| 4 | Attaches geantes 47x60*50 | 4.99 |
| 5 | Lames pour cutter boite*10 | 2.15 |
| 6 | Pate adhesive Uhu Patafix | 3.99 |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from client;
+-----+-----+
| code_client | raison_sociale |
+-----+-----+
| 1 | Dupond et Fils |
| 2 | Etablissements Durand |
| 3 | Bob le Magnifique |
| 4 | L'Alcazar |
| 5 | Le Poney Fringant |
| 6 | Pub The Bell |
| 7 | Riton la bricole |
| 8 | Zaza aux mains d'argent |
+-----+-----+
8 rows in set (0.00 sec)
```



LA BD DEVRA ETRE COMPLETEE AVEC LA(LES) TABLES NECESSAIRES A L'ENREGISTREMENT D'UNE FACTURE.

b) SERVICE WEB

Opérations à <http://localhost:1737/WCFService.svc/rest>

Cette page décrit les opérations du service au point de terminaison.

Uri	Méthode	Description
	GET	Service à http://localhost:1737/WCFService.svc/rest/
{max}	GET	Service à http://localhost:1737/WCFService.svc/rest/{MAX}

Référence pour http://localhost:1737/WCFService.svc/rest/

Url : http://localhost:1737/WCFService.svc/rest/

Méthode HTTP : GET

Direction du message	Format	Corps
Demande	N/A	Le corps de Demande est vide.
Réponse	Xml	Exemple , Schéma
Réponse	Json	Exemple

Ceci est un exemple de corps Xml de réponse :

```
<ArrayOfArticle xmlns="http://schemas.datacontract.org/2004/07/WCFService">
  <Article>
    <id>Contenu de chaîne</id>
    <nom>Contenu de chaîne</nom>
    <pu>Contenu de chaîne</pu>
  </Article>
  <Article>
    <id>Contenu de chaîne</id>
    <nom>Contenu de chaîne</nom>
    <pu>Contenu de chaîne</pu>
  </Article>
</ArrayOfArticle>
```

Ceci est un exemple de corps de réponse Json :

```
[{
  "id":"Contenu de chaîne",
  "nom":"Contenu de chaîne",
  "pu":"Contenu de chaîne"
}]
```

Référence pour http://localhost:1737/WCFService.svc/rest/{MAX}

Url : http://localhost:1737/WCFService.svc/rest/{MAX}

Méthode HTTP : GET

Direction du message	Format	Corps
Demande	N/A	Le corps de Demande est vide.
Réponse	Xml	Exemple , Schéma
Réponse	Json	Exemple

Ceci est un exemple de corps Xml de réponse :

```
<ArrayOfClient xmlns="http://schemas.datacontract.org/2004/07/WCFService">
  <Client>
    <id>Contenu de chaîne</id>
    <raisonSociale>Contenu de chaîne</raisonSociale>
  </Client>
  <Client>
    <id>Contenu de chaîne</id>
    <raisonSociale>Contenu de chaîne</raisonSociale>
  </Client>
</ArrayOfClient>
```

Ceci est un exemple de corps de réponse Json :

```
[{
  "id":"Contenu de chaîne",
  "raisonSociale":"Contenu de chaîne"
}]
```

Ceci est le schéma Xml de réponse :

```
// Classes METIER sérialisables
[Serializable]
[DataContract]
4 références
public class Article
{
    [DataMember]
    public string id;
    [DataMember]
    public string nom;
    [DataMember]
    public string pu;
}
```

```
[Serializable]
[DataContract]
6 références
public class Client
{
    [DataMember]
    public string id;
    [DataMember]
    public string raisonSociale;
}
```

```
// Interface de "contrat" du WS
[ServiceContract]
1 référence
public interface IService
{
    [OperationContract]
    [WebGet(UriTemplate = "", ResponseFormat = WebMessageFormat.Json)]
    1 référence
    List<Article> GetLesArticles();

    [OperationContract]
    [WebGet(UriTemplate = "{max}", ResponseFormat = WebMessageFormat.Json)]
    1 référence
    List<Client> GetLesClients(string max);
}
```

```
// Classe du WS : implémente l'INTERFACE
0 références
public class WCFServiceDemo : IService
{
    1 référence
    public List<Article> GetLesArticles()...

    1 référence
    public List<Client> GetLesClients(string max)...

    1 référence
    private string ChargeLesArticles()...

    1 référence
    private string ChargeLesClients()...
}
```

```
public List<Article> GetLesArticles()
{
    XDocument document = XDocument.Parse(this.ChargeLesArticles());
    var resultat =
        from p in document.Root.Elements("produit")
        select new Article
        {
            id = p.Element("id").Value,
            nom = p.Element("nom").Value,
            pu = p.Element("pu").Value
        };

    return resultat.ToList<Article>();
}
```

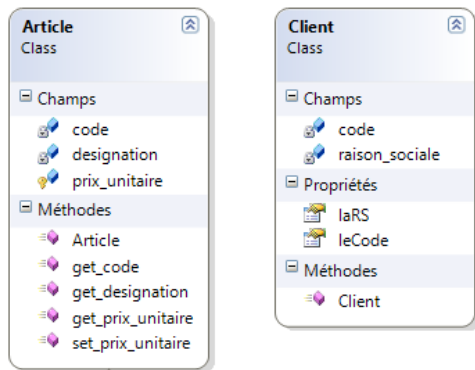
III) CONTRAINTES TECHNIQUES



Une approche 100% OBJET avec les techniques « ad hoc » mises à disposition par le langage C#, devra être mise en œuvre. Il faudra également adopter une ARCHITECTURE APPLICATIVE LA PLUS OPTIMALE ET EFFICACE possible afin d'améliorer et donc d'assurer la maintenance future de l'application (énumération non exhaustive) :

- En organisant et regroupant les classes en fonction de leur rôle.
- En encapsulant les données fournies par les classes techniques dans des collections d'objets métier afin d'en faciliter leur exploitation.
- En permettant l'ajout aisé de futures classes techniques d'accès aux données, avec un impact réduit sur le code existant.
- En prévoyant d'effectuer des requêtes sur les données (articles, clients et factures) à des fins de statistiques.
- En évitant toute redondance du code produit, mais également une dépendance trop forte lors de l'utilisation des classes techniques d'accès aux données.
- En ayant le souci de commenter le code produit, de manière efficace et pertinente.
- Etc.

Classes METIER (ARTICLE et CLIENT)



Ces classes métiers devront être COMPLETES afin de permettre la prise en compte les REMISES sur les ARTICLES et surtout le traitement des FACTURES.

De(s) classe(s) TECHNIQUES : une classe d'accès à une BD MySQL et une classe de consommation du Service Web ont été conçues mais devront être complétées.

- Quel que soit la source de données choisie, leur rôle est d'abord d'« alimenter » l'interface graphique avec une liste de CLIENTS et d'ARTICLES.
- Par la suite, une fois les données d'une FACTURE constituées, les DEUX classes techniques devront effectuer le MEME rôle : enregistrer la FACTURE et son DETAIL dans la BD MySQL.

IV) REMARQUES :

- Le bouton « Ajouter » doit ajouter une LIGNE supplémentaire à l'objet FACTURE en cours, puis formater un objet « string » représentant le RECAPITULATIF de la facture en cours, et qui sera affiché dans le contrôle graphique « RichTextbox ».
- La FACTURE à enregistrer dans la base de données, doit naturellement correspondre à l'AFFICHAGE présenté dans le contrôle « RichTextbox ».
- Dans un souci de simplification, le numéro de FACTURE pourra être calculé au démarrage de manière ALEATOIRE.
- Une fonctionnalité « Afficher factures » devra être prévue. Elle doit permettre d'afficher dans le contrôle « RichTextbox » l'ENSEMBLE des FACTURES enregistrée dans la BD. Par exemple :

FACTURE : 1			
CLIENT : 5		Le Poney Fringant	
<hr/>			
3	Correcteur a sec roller 4,2	1	Total ligne : 6,19 €
2	Trombones 77 mm boite*100	10	Total ligne : 57,9 €
<hr/>			
			TOTAL A PAYER : 64,09 €
<hr/>			
FACTURE : 10			
CLIENT : 1		Dupond et Fils	
<hr/>			
3	Correcteur a sec roller 4,2	100	Total ligne : 619 €
<hr/>			
			TOTAL A PAYER : 619 €

Etc...

V) EVOLUTION

VICHY PAPETERIE envisage dans le futur de mettre en œuvre une application mobile. Pour cela, M. Pierre DUVAL vous demande de lui présenter une MAQUETTE, autour de la gestion de la FACTURATION. Votre chef de projet, en accord avec M. DUVAL, vous transmet les consignes suivantes :

- 1) En attendant la formation sur des techniques mobiles (Xamarin.Forms et Xamarin.Android), la maquette sera réalisée autour de la technologie WPF (« Windows Presentation Foundation ») en prenant soin d'offrir une interface graphique similaire à celle d'un appareil mobile.
- 2) Vous être LIBRE de choisir le type de traitement à présenter. Celui-ci ne peut être qu'un EXTRAIT du traitement complet de facturation que vous avez à développer dans la première mission du projet, par exemple :
 - Choix des produits, puis affichage de la facture.
 - Choix d'un client, puis affichage des factures déjà établies.
 - Etc...
- 3) Pour montrer les ELEMENTS COMMUNS que des applications .NET, quel que soit la technologie choisie, vous devez reprendre UNE ou PLUSIEURS des classes METIER que vous aurez mis en œuvre dans la première mission.
- 4) Vous n'avez pas à utiliser les SOURCES de DONNEES imposées dans la première mission. Vous êtes libre, soit de mettre en œuvre des données de test « en DUR », soit par exemple, à partir d'un fichier XML ou texte.