

Annexe 1 : FICHE TECHNIQUE D'UN SCANNER

Scanner INOTEC SCAMAX 510

Le scanner SCAMAX-510 Haut volume INOTEC est une des références actuelles de scanners ultra-rapides. C'est un matériel de très haute productivité capable de numériser jusqu'à 460 images par minutes en N&B et niveaux de gris.

- **Mode d'acquisition** : recto ou recto-verso
- **Résolution optique** : 200 à 400 dpi (dots per inch) ou ppp (nombre de points par pouce)
- **Vitesse de numérisation en 400 dpi A4 paysage** : 230 ppm (pages par minute)
- **Volumétrie maximum** : 15 000 pages par jour
- **Format d'entrée** : A4/A3
- **Alimentation** : automatique, manuelle
- **Connexion** : SCSI III UW
- **Interface** : ISIS et Twain

Options :

- Niveaux de gris
- Endosseur
- Pédale de commande



Annexe 2 (2A et 2B) : DOCUMENTS POUR LE RECOUVREMENT AMIABLE

- *Annexe 2A – Ordre de mission*

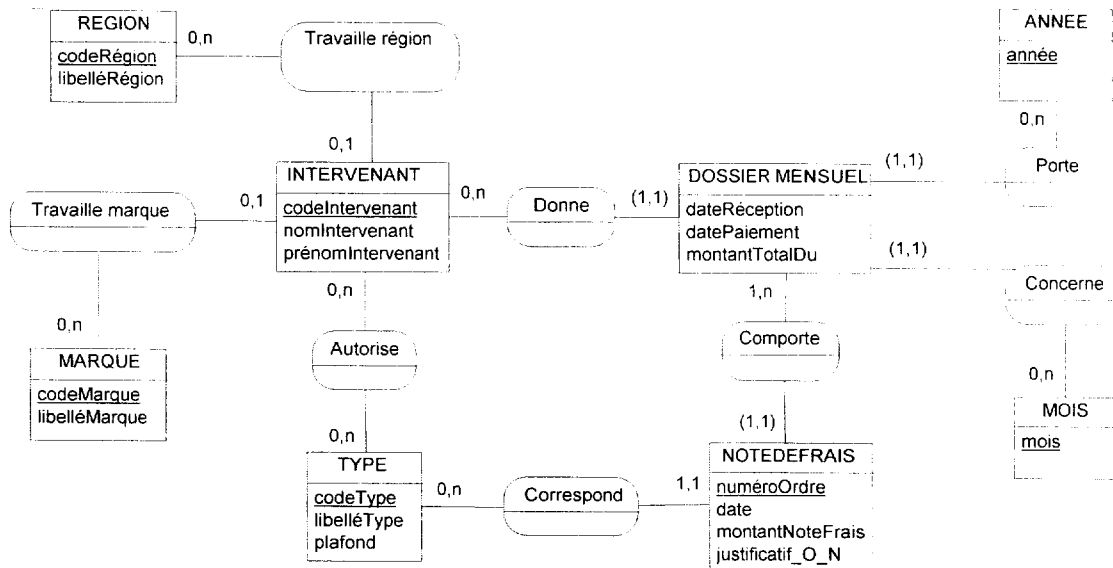
<i>CredAuto</i>	ORDRE DE MISSION N° 7374		
Date ordre de mission : 15 mars 2006	Intervenant : 980045 - Montaz Jean		
Date clôture :	Numéro contrat : 01800020348		
Commentaires : Étudier un réaménagement du prêt en baissant le montant des mensualités. Étaler le remboursement des 5 901,20 € restants sur 38 autres mois. Prendre des personnes cautions.			
Capital non remboursé : 5 901,20 €	N° série véhicule :	VF1PE050759856288	
Éléments du prêt :	Marque véhicule :	Peugeot	
Type de véhicule acheté : <input type="radio"/> Neuf <input checked="" type="radio"/> Occ.	Type véhicule :	406 HDI SV110	
Montant de l'achat : 13 700 €	Immatriculation :	5465 XR 26	
Montant du prêt : 8 000 €	1^{ère} mise en circulation :	16 octobre 2001	
Date de signature : 22 février 2005			
Taux : 5,2000 %			
Contrat initial :	Emprunteur : P15871		
Durée : 36 mois	Nom : DUBUIS		
Jour de l'échéance : 10	Prénom : Pierre		
Première échéance : 10 mars 2005	Adresse : 18 rue Victor Hugo 26000 Valence		
Mensualité : 240,49 €	Profession : Cuisinier		
	N° téléphone : 04 38 99 01 17		
	N° RIB : 30036 17035 00014143657 35		
Dernier avenant :			
Mensualité :			
Durée :			
Echéances impayées			
10/01/06	10/02/06	10/03/06	

• *Annexe 2B – Avenant au contrat de prêt*

<i>CredAuto</i>	AVENANT AU CONTRAT DE PRÊT	
Numéro contrat :	01800020348	Cet avenant ne fait pas novation au contrat.
Numéro avenant :	1	
Date signature avenant :	18 mai 2006	Le taux d'origine ne peut être modifié.
Jour échéance avenant :	05	
Durée avenant :	38 mois	
Date début avenant :	05 juin 2006	
Montant échéance avenant :	168,77 €	
Emprunteur : P15871		
Nom : DUBUIS		
Prénom : Pierre		
Adresse : 18 rue Victor Hugo 26000 Valence		
Profession : Cuisinier		
N° téléphone : 04 38 99 01 17		
N° RIB : 30036 17035 00014143657 35		
Caution rang 1 : P16957		Caution rang 2 : P09524
Nom : BERRARDE		Nom : DUBUIS
Prénom : Daniel		Prénom : Corinne
Adresse : 4 rue des Charmes 26000 Valence		Adresse : 6 Allée Beethoven 26000 Valence
Profession : Représentant		Profession : Informaticien
La caution s'engage à payer en lieu et place de l'emprunteur en cas de défaillance de sa part.		
Date et Signatures		

Annexe 3 (3A et 3B) : GESTION DES FRAIS

- **Annexe 3A – Schéma entité-association**



Remarque : Les cardinalités (1,1) caractérisent un lien identifiant.

NE PAS RENDRE L'ANNEXE 3A AVEC LA COPIE, RECOPIER SUR LA COPIE LES PARTIES UTILES.

- **Annexe 3B – Extrait du schéma relationnel**

Afin de ne pas alourdir le schéma, seules les tables utiles sont présentées :

INTERVENANT(code, nom, prénom, codeMarque, codeRégion)

code : clé primaire

codeMarque : clé étrangère en référence à code de MARQUE (non présentée ici)

codeRégion : clé étrangère en référence à code de REGION (non présentée ici)

DOSSIERMENSUEL(codeIntervenant, année, mois, dateRéception, datePaiement, montantTotalDu)

codeIntervenant, année, mois : clé primaire

codeIntervenant : clé étrangère en référence à code de INTERVENANT

NOTEDEFRAIS(codeIntervenant, année, mois, numéroOrdre, date, montant, justificatif_O_N, codeType)

codeIntervenant, année, mois, numéroOrdre : clé primaire

codeIntervenant, année, mois : clé étrangère en référence à codeIntervenant, année, mois de DOSSIERMENSUEL

codeType : clé étrangère en référence à code de TYPE

TYPE(code, libellé, plafond)

code : clé primaire

Remarques :

- Le champ justificatif_O_N de la table NOTEDEFRAIS prend les valeurs true ou false selon la présence ou l'absence de justificatif pour la note de frais.
- Le champ MontantTotalDu prend pour valeur le cumul de tous les remboursements plafonnés correspondant aux notes de frais déposées par un intervenant pour le mois.

Annexe 4 (4A et 4B) : DESCRIPTIF DES CLASSES

- **Annexe 4A – Classes utilisées pour la partie métier**

➤ Description des classes métier

Classe Région

Privé

code : Chaîne
 libellé : Chaîne
 mesStats : Collection de Stat

Public

Procédure init(unCode : Chaîne, unLibellé : Chaîne)

// initialise les attributs privés de la région

Fonction getCode() : Chaîne *// retourne le code région*

Fonction getNbMaxRecouvrements() : Entier

// retourne le nombre maximum de recouvrements observés parmi

// l'ensemble des modèles ayant fait l'objet de recouvrements dans la région

**Procédure addStat(unLibModèle : Chaîne, unLibMarque : Chaîne,
 unNbPrêts : Entier, unNbRecouv : Entier)**

// ajoute les statistiques concernant un modèle à la région courante

FinClasse

Commentaires du diagramme de classes :

Accessibilité : Symbole – (moins) : membre privé

Symbole + (plus) : membre public

Association entre Région et Stat : Représente l'ensemble des statistiques rattachées à la région (un objet par modèle ayant donné lieu à des recouvrements).

Classe Stat

Privé

libelléModèle : Chaîne
 libelléMarque : Chaîne
 nbPrêts : Entier
 nbRecouvrements : Entier

Public

Procédure init(unLibModèle : Chaîne, unLibMarque : Chaîne, unNbPrêts : Entier,
 unNbRecouv : Entier) *// initialise les attributs privés de l'instance*

Fonction getNbRecouvrements() : Entier *// retourne le nombre de recouvrements*

FinClasse

➤ Description de la classe technique Collection

La classe Collection offre des services pour gérer un ensemble d'objets.

Classe Collection

Public

Fonction cardinal() : Entier *// renvoie le nombre d'éléments de la collection*

Fonction existe(unObjet : Objet) : Entier *// teste si un objet existe dans la collection*

Fonction index(unObjet : Objet) : Entier *// renvoie l'index d'un objet de la collection.*

// Le premier objet de la collection a pour index 1

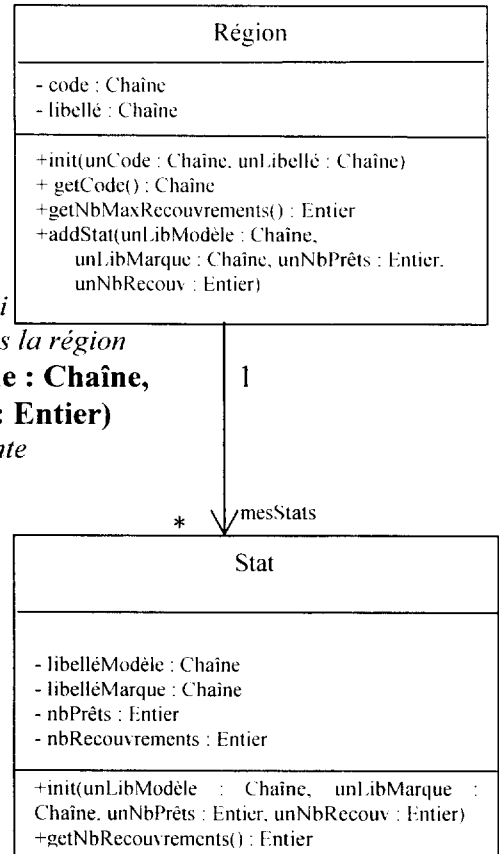
Fonction extraireObjet(unIndex : Entier) : Objet *// retourne l'objet d'index unIndex*

Procédure ajouter(unObjet : Objet) *// ajoute un objet à la collection*

Procédure enlever(unIndex : Entier) *// supprime l'objet d'index unIndex de la collection*

Procédure vider() *// vide le contenu de la collection*

FinClasse



- **Annexe 4B – Classes nécessaires pour l'accès aux données de la base**

La classe Passerelle prend en charge la valorisation des objets Région et Stat en interrogeant la vue vRecouvrement définie dans la base de données relationnelle.

Classe Passerelle

Déclaration des méthodes utiles pour l'étude

// Toutes les méthodes sont à portée classe, dites également **méthodes de classe** ou **static**.

Privé

Fonction à portée classe chargeCaractéristiquesRégion(unCode : Chaîne) : Région

// permet à partir de la vue vRecouvrement de retourner un objet Région dont le

// code est passé en paramètre ; retourne null lorsque la région n'est pas trouvée

Procédure à portée classe chargeLesStats(uneRégion : Région)

// charge à partir de la vue vRecouvrement les statistiques pour une région

Public

Fonction à portée classe donneLaRégion(unCode : Chaîne) : Région

// Cette méthode utilise les deux méthodes privées pour retourner un

// objet Région valorisé avec l'ensemble de ses statistiques associées

Définition de la méthode donneLaRégion

Fonction à portée classe donneLaRégion(unCode : Chaîne) : Région

VAR maRégion : Région

Début

maRégion ← chargeCaractéristiquesRégion(unCode)

si maRégion <> null

alors chargeLesStats.maRégion)

finsi

retourne maRégion

Fin

Un membre souligné désigne un membre à portée classe.

Passerelle
<ul style="list-style-type: none"> - chargeCaractéristiquesRégion(unCode: Chaîne) : Région - chargeLesStats(uneRégion : Région) + donneLaRégion(unCode : Chaîne) : Région

Fin classe

La classe JeuEnregistrements permet de disposer d'un curseur pour manipuler le résultat d'une requête exécutée sur une base de données relationnelle.

Classe JeuEnregistrements

Public

Procédure init(uneRequête : Chaîne) // init réalise la connexion à la base de données

// et charge en mémoire le jeu d'enregistrements résultant de l'exécution de la requête

// SQL passée en paramètre. Le curseur est alors positionné sur le 1er enregistrement s'il existe

Procédure lireSuivant()

// avance le curseur sur l'enregistrement suivant

Fonction fin() : Booléen

// indique si la marque de fin est atteinte

Fonction getValeur(nomChamp : Chaîne) : Variant

// renvoie la valeur du champ nomChamp de l'enregistrement courant.

Procédure fermer()

// ferme le curseur et libère les ressources

FinClasse

Remarques : Variant est un type générique pouvant contenir tout type de valeur.

On peut utiliser l'opérateur « + » pour concaténer des valeurs de type « Chaîne » ou « Variant ».

Exemple :

unJeu : JeuEnregistrements

// déclaration de l'objet

unJeu.init("SELECT * FROM vRecouvrement")

// instantiation de l'objet unJeu qui permet

// d'accéder à tous les enregistrements de la vue vRecouvrement