

DOSSIER DE VALIDATION

CONCEPTEUR DEVELOPPEUR D'APPLICATIONS NUMÉRIQUES

Nom Prénom	Changey Elodie
Nom(s) Prénom(s) du ou des tuteurs	Labrue Karine
Acronyme de la certification IPI visée	CDAN
Niveau visé	Niveau 6
Date de la soutenance	26 novembre 2019
Lieu de la soutenance	EPSI Bordeaux

Table des matières

Présentation de l'entreprise	1
Les Centres de Services	3
CdS Tenue de Compte	3
CdS Center of Excellence Customer Communication Management.....	4
Présentation des postes occupés	5
Ma mission au sein du CdS Tenue de Compte.....	5
Ma mission au sein du CdS Center of Excellent Customer Communication Management.....	7
Projets présentés	8
Projet zThaler	9
Compétences mises en œuvre	9
Organisation et suivi.....	11
Programme Batch I7B210	15
Exemples complémentaires	29
Synthèse du projet	33
Projet 3D Catalog	34
Compétences mises en œuvre	34
Contexte du projet	35
Analyse du projet	37
Synthèse du projet	45
Conclusion générale	46
Annexes.....	47

Remerciements

Dans un premier temps, je tiens à faire part de mes remerciements et de ma gratitude envers l'ensemble des personnes qui ont participé à mon évolution en entreprise, et à mon accompagnement lors des cours. En particulier David Gayerie, qui a été d'une aide précieuse pour la mise en place de mon projet personnel et la résolution de problèmes bloquants.

Je remercie également l'entreprise CGI d'avoir mis en place l'école U'DEV. Sans cette opportunité, je n'aurais peut-être pas eu la chance de me reconvertir dans le domaine du développement et de la programmation.

Un grand merci à l'ensemble des équipes des Centres de Services de Tenue de Compte et CoE CCM pour leur accueil, leur disponibilité et leur bienveillance. Karine Labrue, ma tutrice, a bien veillé à mon intégration au sein de l'entreprise, toujours disponible et à l'écoute. Merci également à Mériem Massieu-Saad et Romuald Joubert, qui se sont beaucoup investis au début de ma formation pour m'épauler dans ma montée en compétences. Et à Guillaume Dubès, pour son implication dans la récupération de données nécessaires à la réalisation de ce dossier de validation.

Note de confidentialité

Le présent rapport, ainsi que les informations qu'il contient sont confidentiels et destinés exclusivement à l'équipe pédagogique de l'EPSI, aux membres de jury du titre de l'IPI, ainsi qu'à l'entreprise CGI. Toute diffusion, publication, totale ou partielle ou divulgation sous quelque forme que ce soit sont proscrites.

Présentation de l'entreprise

CGI, dont le sigle signifie « Conseillers en gestion et informatique », est une entreprise canadienne d'envergure mondiale. Elle est spécialisée en services-conseils en technologie de l'information, intégration de systèmes, impartition et solutions. Son siège social est établi à Montréal, Québec, Canada. Elle a été fondée en 1976 par Serge Godin et André Imbeau, et ne comptait qu'un seul client à sa création. Ensemble, ils se sont engagés à construire une entreprise fondée sur un rêve commun : « Créer un environnement où nous avons du plaisir à travailler ensemble et où, en tant que propriétaires, nous participons au développement d'une entreprise dont nous sommes fiers ».

Fondée en 1976

43 ans de croissance rentable

11,5 milliards \$CA
de revenus

77 500
consultants

40 pays
400 emplacements

5 000 clients à l'échelle mondiale
font appel à nos services complets

Plus de **175 solutions de propriété**
intellectuelle desservant 30 000 clients



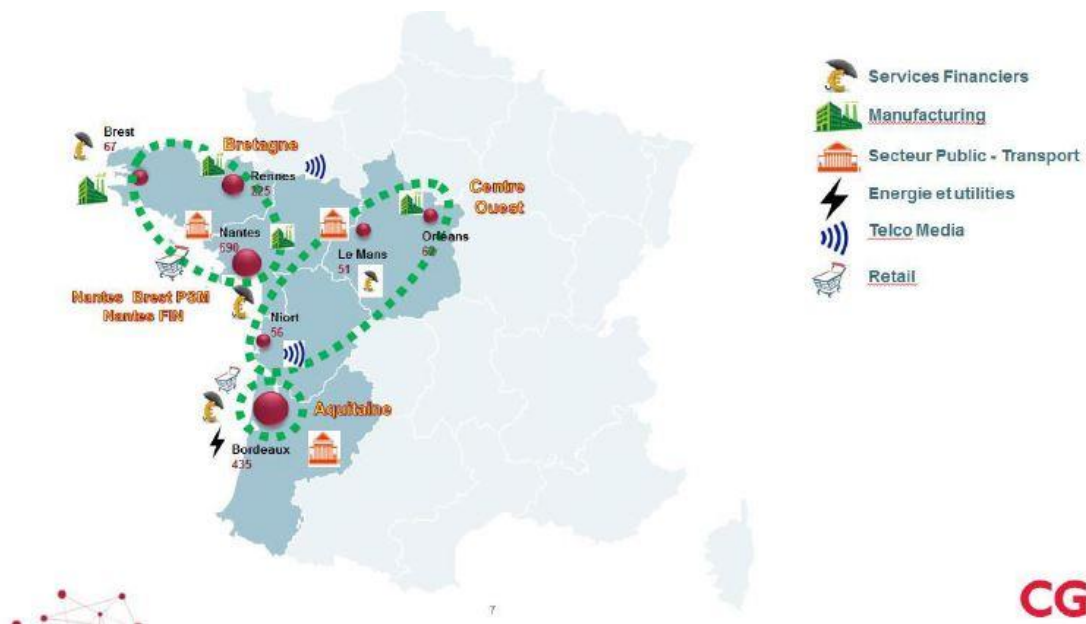
Aujourd'hui CGI compte plus de 5000 clients, avec 77 500 membres travaillant dans 40 pays et répartis dans 400 agences. Le groupe se classe parmi les cinq plus grands acteurs mondiaux du secteur des ESN (Entreprise de Services du Numérique). Sa clientèle est exclusivement orientée vers les grands comptes (banque, assurance, aéronautique, grande distribution, etc).

CGI est implanté dans la France entière avec ses 22 Centres de Services, qui emploient 11 000 membres. L'agence de Bordeaux Le Haillan, qui fait partie des plus importantes en taille en province, compte près de 900 personnes.

En plus d'un découpage par agence, l'entreprise est répartie en Business Units (BU). Ce sont différentes entités, qui adressent des clients avec un profil spécifique.

Ainsi, au sein de l'agence de Bordeaux (Le Haillan), on trouve trois BU distinctes :

- Grand Ouest : adresse des clients issus de tous secteurs d'activités, situés dans les régions de l'Ouest et du Sud-Ouest.
- FGDC : adresse des clients nationaux, tous secteurs d'activités.
- TPS HR : adresse des clients des secteurs Services Publics, Ressources Humaines et Transport.



Pour ma part, je fais partie de la BU Grand Ouest pour deux Centres de Services (CdS) différents. J'ai d'abord intégré le CdS de Tenue de Compte avec un client unique : La Banque Postale, puis le CdS d'Éditique pour Malakoff Médéric.

Les Centres de Services

Un Centre de Services (CdS) est une équipe constituée d'ingénieurs et de techniciens spécialisés sur une technologie ou un secteur d'activité. Ils interviennent à distance sur des projets de développement ou de maintenance applicative, dans les locaux de CGI.

CdS Tenue de Compte

Établissement bancaire à part entière depuis le 1er janvier 2006, La Banque Postale est un acteur majeur de la banque de détail en France. Notre interlocuteur direct est la Direction Informatique des Services Financiers et de l'Enseigne (DISFE). CGI accompagne son client sur près de la moitié du parc de ses applications métiers.

Au sein de l'entité Grand Ouest, plusieurs CdS sont affectés au compte client de La Banque Postale, remplissant un contrat de service historique débuté il y a 18 ans et prolongé actuellement jusqu'en 2021.

Techniquement, le Système d'Information de La Banque Postale a pour caractéristique sa capacité à traiter de très fortes volumétries avec un haut niveau de disponibilité exigé.

Avec pour objectif de gérer l'ensemble des produits d'épargne via le logiciel AM (Account Management), le SI du CdS de Tenue de Compte remplit les fonctionnalités suivantes :

- La gestion des comptes
- La gestion des soldes
- La gestion des mouvements
- Le calcul des intérêts
- Les restitutions réglementaires et restitutions internes

La rénovation du SI de La Banque Postale a démarré depuis 2008, avec une mise en œuvre effectuée sous la forme de lots successifs dont certains sont encore en cours à ce jour. Elle s'accompagne d'une montée de version du logiciel via le projet technique MDV (Montée De Version), effectuée en avril 2019, passant le socle Thaler 2.4 vers AM 4.0. Les nouvelles fonctionnalités sur les produits Tenue de Compte sont implémentées via le projet MC&A (Multi-Comptes et Arrêtés), qui donne lieu à la version 4.1 du logiciel.

J'ai travaillé au sein de ce CdS de décembre 2018 à août 2019, en tant que développeuse sur MDV et MC&A.

CdS Center of Excellence Customer Communication Management

Le cœur de métier du CdS CoE CCM est l'Éditique, en d'autres termes la conception, composition et diffusion automatisée de masse de documents (souvent administratifs), tels que des factures, relevés de compte, prospection, contrats, etc. Nous proposons aux clients des solutions, en collaboration avec nos partenaires, afin de mettre en place chez eux un socle éditique, par la suite alimenté de templates pour générer les documents nécessaires grâce à leurs flux.

Ce CdS compte plusieurs clients comme La Banque Postale ou Régaz. Je travaille pour ma part sur le projet SI2M, pour le groupe de protection sociale Malakoff Médéric. Leurs domaines principaux d'activité sont la gestion de la retraite complémentaire et les assurances de personnes.

La collaboration avec CGI a démarré en 2011, avec un souhait initial de moderniser leur plateforme éditique.

Je travaille dans ce Centre de Services depuis août 2019.

Présentation des postes occupés

Ma mission au sein du CdS Tenue de Compte

En tant que développeuse, je travaille généralement en binôme avec un analyste qui me transmet les éléments d'une demande. Je dois respecter les normes de développement et les processus opérationnels du projet. Je travaille principalement sur des évolutions et des correctifs d'applications, mais aussi pour la création de nouveaux programmes.

Mes missions au sein de ce Centre de Services sont les suivantes :

- Développer les programmes qui me sont confiés, en respectant les spécifications détaillées fournies et les normes qualité
- Réaliser et tracer les Tests Unitaires (faire des fiches de test)
- Prendre connaissance du Plan de Charge chaque jour et suivre les éventuels changements communiqués
- Respecter les dates de livraison demandées
- Respecter les consignes et normes demandées par le client et l'analyste en charge
- Avoir un regard critique sur les spécifications et remonter les éventuelles incompréhensions et incohérences.

Je travaille dans un environnement Mainframe. Il s'agit d'un ordinateur central doté d'une grande puissance de traitement et sur lequel travaillent tous les collaborateurs à partir de terminaux, de façon simultanée. Une large majorité de banques disposent de cette architecture.

Je travaille sur le système d'exploitation z/OS (IBM) via un émulateur de terminal IBM. Cette fenêtre d'émulation est intégrée à l'IDE Eclipse, et fait partie d'un ensemble d'outils nommé UDC (Usine de Développement COBOL). Mon poste de travail est constitué d'un ordinateur fixe pour accéder au réseau sécurisé La Banque Postale, et d'un portable qui accède au réseau CGI.

```

                                IBM
00 00 0          0 00
00 00          0 0000
00 00 0 000 0 0 000 0 00 00 000 000
00 00 0 0 0 0 0 0 000000 0 0 0 0
00 00 0 0 0 0 0000 0 00 00 0 0000
000 0 0 0 0 0 0 0 00 00 0 0 00000 0
0 0 000 0000 0000 0 00 00 000 00 00 0
0 00 00 000 000 0000 000 000 000 0 R
00 00000 0 0 0 0 0 0 0 0
00 0000 0 0 0 0000 0 0000
00 0 0 0 0 0 0 0 0
00 0000 000 0000 0000 000 0000
3.5 V04 5655-F37

Licensed Materials - Property of IBM
(C) Copyright IBM Corp. 1983, 2007.
All Rights Reserved.
* Trademarks of International Business Machines Corp.
----- LICENCE ACCORDEE A MPDGP COS -----
                                08/08/2019

BASE.....: PB12
UTILISATEUR.....: MOT DE PASSE.:
BIBLIOTHEQUE.....: SESSION.....:
NUMERO DE MODIFICATION.:
MODIFICATION MOT PASSE.: CONFIRMATION.:

0: C1 CH:

```

Le langage principal utilisé est le COBOL, technologie historique d’une large majorité de banques, assurances, grandes industries, etc. Je travaille principalement sur Pacbase, un atelier de génie logiciel simplifiant le travail de développement en COBOL. Il comprend un framework et un dictionnaire de données, et génère du code source COBOL.

```

Menu Utilities Compilers Options Status Help
-----
ISPF Menu Primaire z/OS 02.03.00

Option ==>
0 Settings      Terminal and user parameters      User ID  : XGPL504
1 View          Display source data or listings  Time    : 13:58
2 Edit          Create or change source data   Terminal: 3278
3 Utilities     Perform utility functions       Screen   : 1
4 Foreground    Interactive language processing Language: ENGLISH
5 Batch         Submit job for language processing Appl ID : ISP
6 Command       Enter TSO or Workstation commands TSO logon: IKJSTD
7 Dialog Test   Perform dialog testing          TSO prefix: TS0SX11
8 LM Facility   Library administrator functions System ID: SX11
9 IBM Products  IBM program development products MVS acct.: UGXX
10 SCLM         SW Configuration Library Manager Release : ISPF 7.3
11 Workplace    ISPF Object/Action Workplace
12 z/OS System  z/OS system programmer applications
13 z/OS User    z/OS user applications

P PRODUITS     PRODUITS complémentaires
S SDSF         System Display and Search Facility

*LBPàPRI
ME a 04/014

```

TSO (pour Time Sharing Option) est un interpréteur de lignes de commande qui interagit avec z/OS. Sous TSO, je peux utiliser toutes les ressources de la machine pour créer, visualiser, détruire des fichiers, écrire et soumettre des programmes.

Via TSO, il est possible d’accéder à différents outils et applications :

- Platinum : dialogue avec les bases de données DB2
- SDSF : rapports de compilation
- JCL (Job Control Language) : exécution et enchaînement des programmes Batch
- File-Aid : éditeur de fichiers
- Xped : outil de débogage
- Etc.

Ma mission au sein du CdS Center of Excellent Customer Communication Management

Le projet du CdS CoE CCM sur lequel je travaille est une Tierce Maintenance Applicative (TMA) pour Malakoff Médéric, appelée SI2M. Ce projet consiste principalement à l'analyse des entrants (spécifications détaillées), pour maintenir ou créer de nouveaux templates pour les courriers ou outils internes d'Éditique Malakoff Médéric.

La technologie utilisée est le logiciel OpenText Exstream, une des principales solutions d'édition du marché, comprenant un Manager (à gauche, pour la gestion de la base de données des templates : documents, fonctions, variables, etc.) ainsi qu'un Designer (à droite, pour la création desdits documents).

Projets présentés

Les projets auxquels j'ai pu participer en entreprise m'ont permis d'appréhender différentes facettes du métier de la programmation et du développement :

- Analyse
- Mise en place/programmation
- Préparation et réalisation de tests

J'ai choisi de présenter une réalisation que j'ai développée pour La Banque Postale dans le contexte du projet zThaler. Toutefois, mes missions à CGI étaient ciblées et ne couvraient pas la totalité des phases de bout en bout. En complément, je présente également l'application 3D Catalog que j'ai développée pour répondre à un besoin personnel.

Projet zThaler

Compétences mises en œuvre

Voici la liste des compétences mises en œuvre par ce projet. Pour des détails supplémentaires concernant l'acquisition de chaque compétence, merci de vous référer au tableau référentiel de l'IPI en Annexe 4.

Qualité et sécurisation du code réalisé

- Formaliser, identifier les résultats attendus.
- Respecter des contraintes, les recommandations qualité de la norme en vigueur pour l'architecture des logiciels
- Livrer du logiciel déverminé et conforme aux attentes
- Clôturer une mission.

Audit, conception, méthode de projet

- Concevoir des éléments logiciels réutilisables
- Produire du logiciel en équipe
- Remonter les alertes au(x) décideur(s)
- Estimer des délais

Réalisation d'applications logicielles

- Produire du logiciel générique réutilisable et partageable
- Intégrer des éléments logiciels hétérogènes et produire des exécutables livrables
- Modifier un algorithme sans générer de dysfonctionnements
- Contrôler des délais

Communiquer avec les acteurs du projet

- Interagir efficacement dans un environnement de travail collaboratif

Adapter l'environnement d'exécution, échanger des données entre logiciels

- Réaliser des échanges de données informatisés (EDI)
- Automatiser des traitements
- Programmer des scripts systèmes

Organisation et suivi

Organisation

Cette section est dédiée à la présentation de travaux que j'ai effectués au sein du CdS de Tenue de Compte. Lors de ces neuf mois en Centre de Service pour La Banque Postale, j'ai eu l'occasion de travailler sur des développements visant à faire évoluer différentes interfaces existantes.

J'ai choisi de présenter des exemples significatifs par rapport aux compétences demandées pour la validation de ce titre, chacun ayant ses spécificités. Ainsi dans cette section, je présente le développement de différentes solutions :

- Exemple 1 : Création d'un programme Batch avec fichier et point AM en entrée, et fichier en sortie
- Exemple 2 : Évolution de quatre programmes Batch de la chaîne ISI49, et exemple de test de régression

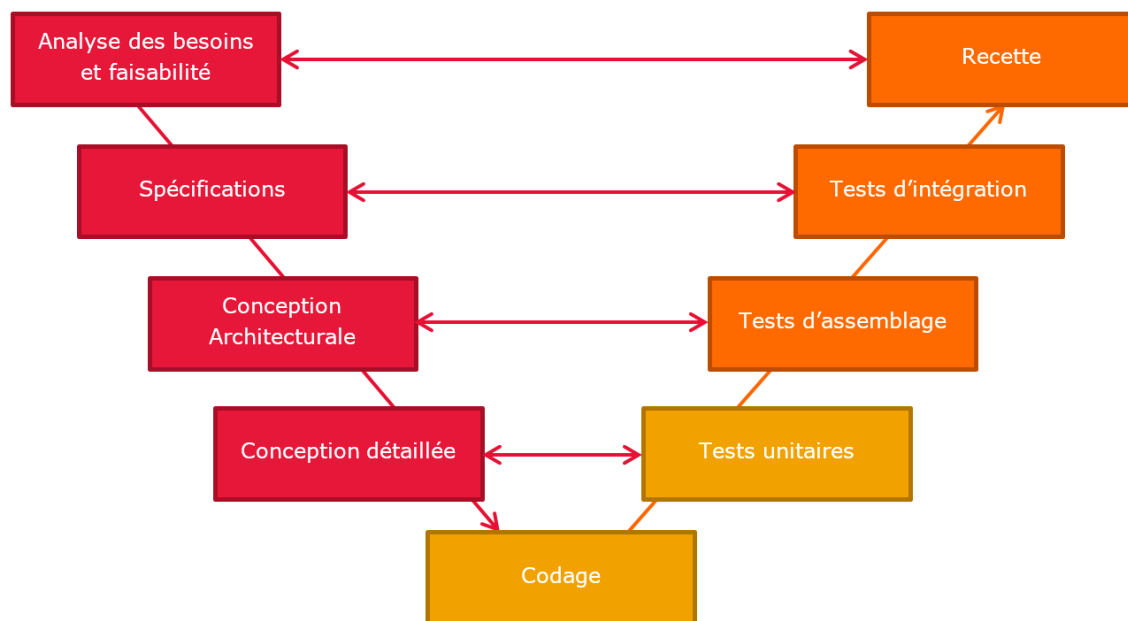
Chaque programme, qu'il soit issu d'une création ou d'une évolution, constitue une tâche à part entière avec son cahier des charges, ses spécifications et sa fiche de tests unitaires. Ils ont tous été développés dans un environnement Mainframe.

La méthodologie du cycle en V

L'équipe de Tenue de Compte travaille selon l'organisation du cycle en V. Ainsi, pour chaque demande du client, un analyste effectue une étude de faisabilité et d'impact, et rédige un devis.

Après acceptation du devis par le client, l'analyste réalise la conception technique (dessins de chaîne) et rédige un Dossier de Conception Général (DCG) à partir des spécifications fonctionnelles issues du Cahier Des Charges (CDC). On y trouve généralement les schémas, les règles de gestion et un algorithme global.

Une fois le DCG validé, l'analyste rédige le Dossier de CONception Détaillé (DCOD) dans lequel chaque module du programme est décomposé pour permettre au(x) développeur(s) de comprendre les attentes et de les traduire dans Pacbase. Pour chaque module, on détaille les entrées/sorties, les segments utilisés et un algorithme détaillant les conditions, boucles, opérations, etc.



En tant que développeur, je travaille sur la phase de réalisation (en jaune sur le graphique). Pour chaque tâche je prends donc connaissance du DCOD, puis je dois rédiger un plan de test du module à développer. Je le valide ensuite avec l'analyste ayant rédigé le DCOD, afin que nous soyons bien d'accord sur les attentes du programme.

Une fois ceci fait, je peux passer à la réalisation puis à l'exécution des Tests Unitaires (TU) pour vérifier que le module répond bien à la demande. Ces TU sont consolidés par le passage de Tests d'Assemblage (TA= passage de tests sur l'enchaînement des différents modules) et réalisés par l'analyste en charge de la demande.

Notre équipe prend en charge les Tests d'Intégration (TI). La recette est effectuée par le client, ainsi que la mise en production.

Suivi du projet

Le suivi de l'avancement est contrôlé par la Chargée de Projet via deux outils distincts et complémentaires : MyPM et le tableau de suivi visuel.

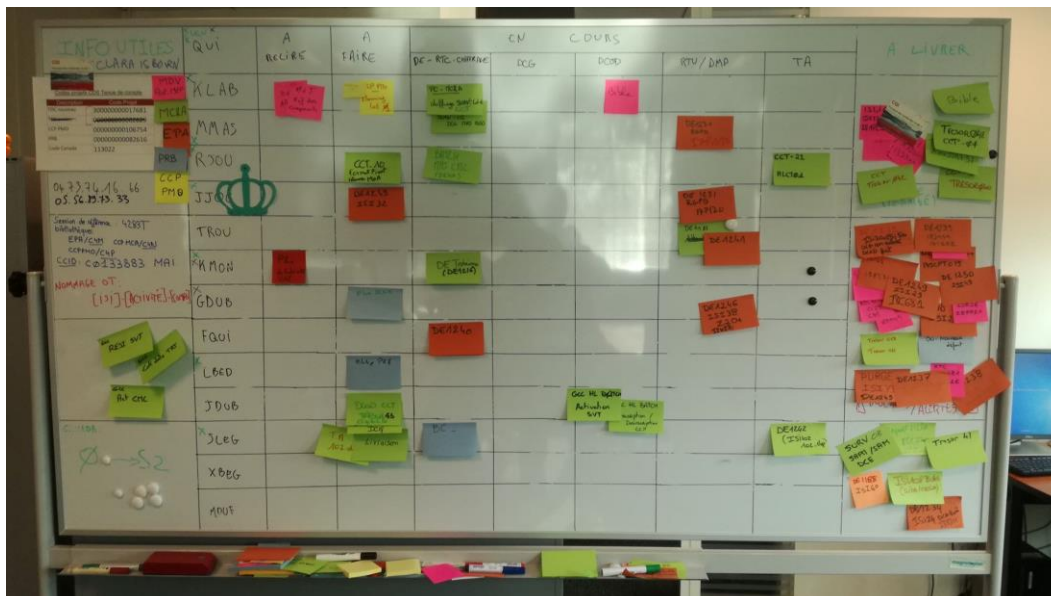
Tout d'abord l'outil MyPM (My Project Management), est un outil d'industrialisation développé en interne par les équipes de CGI.



Il permet à la fois aux membres de l'équipe de saisir leurs imputations, et aux pilotes (Chargés de Projet, Chefs de Projet) de créer les demandes (OT), réaliser le suivi de l'avancement pour chaque tâche, suivre la productivité et la gestion financière du projet.

Pour chaque tâche, je saisis le temps passé à la réalisation ainsi qu'une estimation de mon reste à faire (en nombre de jours). Je dois effectuer ce travail au minimum une fois par jour afin de faciliter le travail du pilote et d'anticiper tout retard, dépassement de charge, etc.

Un Daily Meeting a également lieu chaque jour à 11h45 sur le plateau devant un tableau de suivi visuel indiquant les travaux en cours pour chaque membre de l'équipe. Ce tableau est à disposition permanente de tous les membres de l'équipe, et chacun est tenu de mettre à jour son activité régulièrement. Cet outil, issu des méthodes agiles, est utilisé sur beaucoup de projets La Banque Postale en cycle en V.



Durant ce moment (15 à 20 minutes), chaque membre de l'équipe présente les sujets sur lesquels il travaille ou va travailler. Généralement, un animateur (symbolisé par le post-it "couronne") est nommé chaque jour à tour de rôle pour distribuer la parole et animer les échanges.

Le Daily Meeting est un moment d'échange au sein de l'équipe, pendant lequel tout le monde a la parole à tour de rôle. Il me permet d'évoquer mes difficultés, les problèmes que je rencontre et parfois de trouver des solutions simples et rapides à mettre en place auprès de mes collègues plus expérimentés. J'ai pu constater que pour être efficace, cette réunion ne doit pas se transformer en session de reporting vers les responsables du projet. Elle doit rester un moment d'échange, limité dans le temps, en impliquant tous les membres de l'équipe.

Programme Batch I7B210

Contexte du projet

Expression du besoin

I7B210 (Blocage / déblocage de montant en masse) est le premier programme Batch que j'ai développé pour la Tenue de Compte avec Pacbase dans le cadre de MDV (Montée De Version). Il fait partie d'une nouvelle interface : ISI13 bis, et permet de positionner un blocage de montant sur un Compte Épargne, de modifier un blocage de montant ou de débloquer un montant.

Contraintes

C'est un traitement quotidien, qui se déclenche à réception du fichier FICBLO (fichier séquentiel qui contient la liste des comptes à traiter par le programme). Pour chaque enregistrement lu, un contrôle de validité des données en entrée est effectué. Chaque enregistrement ne validant pas les données requises est ignoré, mais reconduit en sortie avec un code retour spécifique.

À l'initialisation du programme, il effectue les vérifications suivantes :

- Si la date est renseignée et au format correct
- Si le fichier en entrée FICBLO contient ou non des enregistrements.

En sortie de cette étape de contrôle, un fichier de compte-rendu (TRACE) est alimenté. En cas d'erreur concernant la date, le programme s'arrête avec un code retour d'erreur bloquante et les messages d'erreur correspondants dans le fichier TRACE. Si le fichier FICBLO est vide, le traitement s'arrête normalement avec le message dans le fichier TRACE indiquant que le Fichier des comptes est vide.

À la fin d'un traitement normal, le fichier TRACE doit contenir :

- Un entête
- Le nombre d'enregistrements lus
- Le nombre d'enregistrements écrits
- Un pied de page

Livrables attendus

L'analyste en charge de cette demande attend que je produise les éléments suivants :

- Le plan de test du programme I7B210, à valider ensemble avant la phase de développement
- Le programme I7B210 développé avec Pacbase, compilé sur le système et avec une note PQC supérieure à 95/100
- La fiche de tests unitaires dûment complétée
- La mise à jour du tableau de livraison présent sur le réseau sécurisé La Banque Postale

La note PQC (Pacbase Quality Control) est la note attribuée par le compilateur au programme selon des spécifications précises (exemples : toutes les fonctions ont un titre, pas d'utilisation de GO TO hors macro, pas de fonction trop longue, présence suffisante de commentaires, etc.). Cette notation entre dans le contrat passé entre CGI et La Banque Postale. Les programmes livrés ne doivent pas avoir une note inférieure à 90/100 avec un objectif fixé à 95/100.

Analyse du projet

Spécifications fonctionnelles

L'analyste en charge de cette demande a repris la demande du client et produit un document appelé DCG (Dossier de Conception Générale).

Ce document reprend des éléments du cahier des charges et définit les règles de gestion qui devront être respectées par le programme. En voici les quatre premières pour exemple :

Règle de Gestion n°1 : Code action

Le code action doit être valorisé à 'C' pour la création d'un blocage de montant, 'M' pour la modification d'un blocage de montant ou 'S' pour le déblocage d'un montant.

Si ce code action n'est pas renseigné ou est différent de 'C', 'M' ou 'S', l'enregistrement est ignoré. Le code retour est valorisé respectivement à '0102' ou '0103'.

Règle de Gestion n°2 : Numéro de compte

Le numéro de compte doit être renseigné pour tous les types de traitements.

S'il n'est pas renseigné, l'enregistrement est ignoré. Le code retour est valorisé à '0104'.

Règle de Gestion n°3 : Date de blocage du montant

La date de blocage du montant doit être renseignée pour tous les types de traitements.

Si elle n'est pas renseignée ou est erronée, l'enregistrement est ignoré. Le code retour est valorisé à '0107'.

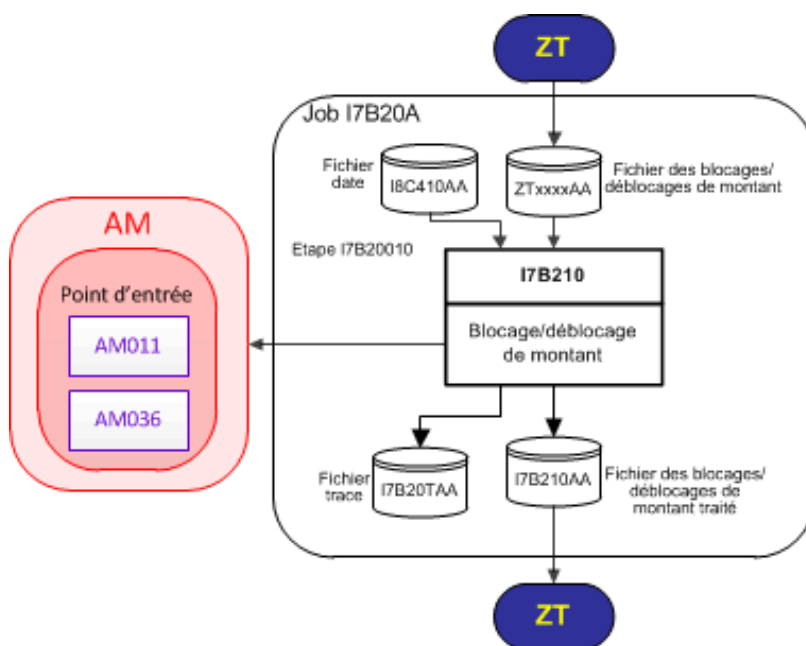
Règle de Gestion n°4 : Montant du blocage

Le montant du blocage doit être renseigné dans le cas d'une création ou d'une modification de blocage de montant.

S'il n'est pas renseigné, l'enregistrement est ignoré. Le code retour est valorisé à '0109'.

Spécifications techniques

Mon travail est la création du programme I7B210, qui est un des modules de l'interface ISI13 bis. Cette interface ne comprend que ce programme pour l'instant.



Chaque application de traitement Batch est divisée en différents jobs (chaînes) qui contrôlent l'exécution automatique de programmes et leur enchaînement le cas échéant, selon une date, un évènement (à réception d'un fichier par exemple), un lancement manuel, etc.

Le module que je dois créer permet à l'application de contrôler chaque enregistrement (ou ligne) d'un fichier entrant en termes de structure et de cohérence. En cas d'incohérence constatée, le traitement ne doit pas être interrompu (le programme doit continuer à traiter les autres enregistrements) mais le compte concerné doit tout de même figurer dans le fichier de sortie, illustré par un code retour spécifique.

Le fichier en entrée peut contenir de nombreux enregistrements, et chacun d'entre eux est segmenté en différentes rubriques (ou colonnes).

Réalisation

Scénario de test

Avant de commencer le développement, je prends connaissance du DCOD et pose des questions à l'analyste sur les éléments que je ne comprends pas ou qui me paraissent manquer de précision.

Dans Pacbase, j'étudie les rubriques des entrants dans le programme grâce au dictionnaire des données inclus, ainsi que les macros qui pourront m'être utiles.

Dans Pacbase, une macro est une sorte de fonction générique qui va pouvoir être utilisée dans de nombreux programmes. Exemples de macro : gestion d'un curseur SELECT pour la base de données, écriture du fichier trace, récupération du timestamp, etc.

Suite à cela, je définis mon plan de test en envisageant tous les cas possibles afin de couvrir l'intégralité des lignes de code (toutes les conditions, y compris les traitements en erreur).

N° séquen- ce cas de tests	ACTION [description des actions ou modes opératoires conduisant à la situation à tester]	RESULTATS ATTENDUS [description du résultat attendu – ex. : copie d'écran, message, valorisation donnée, ...]	RESULTATS OBTENUS [description du résultat obtenu – ex. : copie d'écran, message, valorisation donnée, ...]	STATUT [OK ou KO : selon la conformité du résultat obtenu par rapport à l'attendu]
01	Cas passant : Création d'un motif de blocage Modification d'un motif de blocage Suppression d'un motif de blocage Si code action non renseigné Si code action <> de C, M ou S Si numéro de compte non renseigné Si date blocage non renseignée pour C ou M Si date blocage invalide pour C ou M Si montant blocage non renseigné pour C ou M Si entité blocage non renseignée pour C Si motif blocage non renseigné pour C ou M Si réf blocage non renseignée pour M ou S	<ul style="list-style-type: none">Comptes reconduits en sortie avec ZCRUT0 à 0000 ou alimenté par le code d'erreur correspondant au casFichier TRACE complétéCode retour 00	Conformes aux résultats attendus	OK
02	Cas passant : Numéro de compte non valide Plantage AM	<ul style="list-style-type: none">Compte reconduit en sortie avec ZCRUT0 à 0202Fichier TRACE complétéCode retour 00	Conformes aux résultats attendus	OK
03	Cas non passant : Fichier DATE vide ou date comptable invalide	<ul style="list-style-type: none">Message fichier TRACEPas d'écriture dans fichier FICBLOCode retour 12	Conformes aux résultats attendus	OK
04	Cas passant : Fichier FICBLO vide	<ul style="list-style-type: none">Fichier TRACE complétéCode retour 00	Conformes aux résultats attendus	OK

Dans ce tableau, je renseigne tous les éléments demandés concernant le programme à tester : la description de l'objectif du test, les actions à réaliser pour tester le comportement du programme, ainsi que le résultat attendu (code retour, contenu du fichier trace, contenu du fichier de sortie, etc...).

Pour exemple, voici les cas de test prévus pour ce programme :

- Cas n°1 : Traitement normal / Fichier FICBLO contenant un enregistrement pour chacune des 12 possibilités d'alimentation
- Cas n°2 : Traitement normal / Numéro de compte invalide, ce qui doit provoquer un plantage lors de l'accès au point d'entrée AM
- Cas n°3 : Traitement en erreur / Fichier DATE vide ou format date invalide
- Cas n°4 : Traitement normal / Fichier FICBLO vide

L'intérêt de valider en amont le plan de test est de vérifier que le développeur comprend les attentes du programme. Cela permet également de confronter les idées lors des discussions avec l'analyste afin de partir sur un programme le plus optimal et fiable possible. Une fois ce scénario validé par l'analyste, je commence le développement du programme.

Développement

Tout développement sous Pacbase commence par la création de l'enveloppe du programme, aussi appelée carte de définition. Les programmes sont structurés en cartes (héritage du système historique des cartes perforées). Pour I7B210, elle se présente ainsi :

```
VA Pac 3.5 V04 Tenue de compte - AM (TC) *XGPL504.PB12.C4M.6936
CODE PROGRAMME I7B210
NOM DU PROGRAMME.....: Blocages déblocages en masse
CODE CLASSEMENT DU PROGRAMME.....: I7B210
VARIANTE DU LANGAGE A GENERER.....: X IBM MVS/ESA OS/390
OPTION NUMEROTATION CADRAGE COBOL...:
OPTION CARTES AVANT PROGRAMME.....:
OPTION CARTES APRES PROGRAMME.....:
CODE DU PROGRAMME GENERE.....: I7B210
TYPE DE PROGRAMMATION.....: P
NATURE DU PROGRAMME.....: B
TYPE DE L'ENTITE.....: P PROGRAMME
CONTROLE DE PRESENCE ZONE NUMERIQUE:
GENERATION INDICATEURS SQL AVEC '-':
MOTS CLES ASSOCIES.: THALER ISI13B
MIS A JOUR PAR.....: XRWB123 LE : 03/05/2019 A : 14:05:09 BIB : C4M
NO DE SESSION.....: 6650 BIBLIOTHEQUE : C4M BLOCAGE :
O: C1 CH: ACTION:
```

Puis je renseigne la carte commentaires du programme, qui est généralement une des premières sources de documentation.

```

VA Pac 3.5 V04 Tenue de compte - AM (TC) *XGPL504.PB12.C4M.6936
COMMENTAIRES DU PROGRAMME I7B210
                                Blocages débloages en masse
A NLG : T DESCRIPTION BIBLI
100 : AUTEUR
      : DATE DE CREATION : 10/04/19 AUTEUR
      : NOM DE L'AUTEUR : Elodie Changey AUTEUR
      : SOCIETE : CGI AUTEUR
120 : 6650
140 : *** INTERFACE ISI13B : Blocages et débloages de montants en 6650
160 : masse *** 6650
180 : 6650
200 : DESCRIPTION : Ce programme permet de positionner un blocage 6650
220 : de montant, de modifier un blocage de montant ou de 6650
240 : débloquent un montant pour Compte Epargne. 6650
260 : 6650
280 : ENTREE : Fichier des blocages/débloages de montant 6650
300 : Fichier Date 6650
320 : SORTIE : Fichier des blocages/débloages de montant traité 6650
340 : Fichier compte-rendu du job 6650
      :
      :
O: C1 CH:

```

Cette carte s’enrichira au fur et à mesure des évolutions futures apportées au programme (réalisées potentiellement par d’autres développeurs). Une seule règle : à chaque modification, on renseigne cette carte. On utilise un gabarit pour normaliser les commentaires et faciliter leur lecture.

Je remplis ensuite la carte des entrées / sorties (aussi appelées ressources externes). Les ressources externes sont généralement des fichiers séquentiels ou des tables provenant de la base de données.

```

VA Pac 3.5 V04 Tenue de compte - AM (TC) *XGPL504.PB12.C4M.6936
STRUCTURES DE DONNEES DU PROGRAMME I7B210 Blocages débloages en masse
A PR SU : SD EXTERN OAMOU BLOC T R S U RE SE M UNIT P SELECTION F O D N EM
DA : CR DATE SSFIU 0 R T *07=00 I 1
      : Z.COMPL.: CLE ACC.: IDENT. :
E1 : TH FICBLO SSFIU 0 R C *6A=00 I 1
      : Z.COMPL.: CLE ACC.: IDENT. :
S1 : TH BLOMAJ SSFOU 0 R D *6B=00 I 1
      : Z.COMPL.: CLE ACC.: IDENT. :
TR : NT TRACE SSFOU 0 R D *010203 I 1
      : Z.COMPL.: CLE ACC.: IDENT. :

```

Ici, on prend en entrée deux fichiers (générés en amont de la chaîne) :

- DA00 (DATE) : un fichier contenant la date du jour (ainsi que la date J-1 et J-2)
- E100 (FICBLO) : un fichier contenant les enregistrements à traiter

En sortie, on a besoin de deux nouveaux fichiers (générés par la chaîne lors de l’exécution du programme I7B210) :

- S100 (BLOMAJ) : un fichier contenant les enregistrements et le code retour qui résulte de leur traitement

- TR00 (TRACE) : un fichier de compte-rendu

À partir des informations mises sur cette carte, Pacbase génère du code COBOL qui gère :

- Les lectures des fichiers en entrée
- Les écritures des fichiers en sortie
- Des compteurs pour chaque fichier (nb lectures, nb écritures, etc.)
- Des indicateurs de fin de fichier
- Des indicateurs de fin d'itération pour les fichiers avec rupture (cela permet de savoir si on a lu le dernier enregistrement d'une liste)
- Une zone IK pour stocker le résultat d'un accès direct (0 = OK, 1 = KO)

La carte des macros structures permet d'appeler des fonctions standards permettant d'accélérer le processus de développement sous Pacbase en fournissant du code paramétrable (à la manière d'un framework).

```

VA Pac 3.5 V04 Tenue de compte - AM (TC) *XGPL504.PB12.C4M.6936
MACRO-STRUCTURES DU PROGRAMME I7B210 Blocages déblocages en masse

A MACRO NL S : SIGNIFICATION OU VALEUR DES PARAMETRES D V
ABDB11 : 4/3/1/2/DT/ *CONTROLE DATE
ABDC06 : DA/AX3DAT/ *CALCUL DATE
ABIB01 : *CONSTANTES PROGRAMME
ABIB26 : SX/NK1KZ02B/ *ABEND
ABIC01 : 05/10/04/19/I7B210/ *VERSION
ABSB44 : WS/PE/99/ *APPEL C4API
AEIB04 : TR/DA/ *DEBUT ET FIN TRACE
AEIB06 : TR/FICBLO/NB D ENREG LUS /E1/ *COMPTEUR TR
AEIB06 10 : TR/BLOMAJ/NB D ENREG ECRITS /S1/ *COMPTEUR TR
A3AT11 : AM/ *PARAM AM
A3AT12 : 91/BB/WS/AM/TR/ *ALIM C4HEADER
A3AT14 : 10/ZA/ZB/11/ZC/ZD/91/CB/WS/ *GEST ZG C4
A3AT14 * : AM011/ZGROUA/ZGROUB/AM/TR/
A3AT14 10 : 20/ZE/ZF/36/ZG/ZH/91/CC/WS/ *GEST ZG C4
A3AT14 10 * : AM036/ZGROUC/ZGROUD/AM/TR/

```

Les macros qui peuvent être utiles sont généralement proposées dans le DCOD par l'analyste, reste au développeur à les paramétrer correctement

Par exemple, j'ai utilisé et paramétré la macro structure ABDB11 qui permet de réaliser un contrôle formel de date sur 8 caractères.

Les paramètres de cette macro sont:

- \$1 : Position du jour dans la date
- \$2 : Position du mois
- \$3 : Position du siècle
- \$4 : Position de l'année
- \$5 : Emplacement de la sous-fonction (endroit du programme où je souhaite que la macro s'insère).

Mes paramètres pour cette macro sont donc: 4/3/1/2/DT/ (à partir des infos sur le fichier date que j'ai en entrée).

Pour pouvoir utiliser cette macro dans le programme, je déclare en working (variables utilisées dans le programme) une rubrique EN-PRE (alphanumérique sur 1 caractère), qui servira d'indicateur lors de la vérification du format de la date en entrée.

Une fois toutes ces étapes effectuées, je peux coder les traitements spécifiques du programme.

Pacbase utilise une cinématique commune à chaque programme Batch. Pour comprendre la suite du programme, il est nécessaire de connaître la boucle Pacbase dont un descriptif se trouve en Annexe.

Les premières fonctions sont entièrement gérées par des macros. Elles concernent ici l'affichage des messages de début de programme, l'écriture de l'entête du compte-rendu.

TRAITEMENTS PROGRAMME P I7B210 Blocages déblocages en masse										FONCTION: 04									
A	SS	NLG	OPE	OPERANDE						NVTY	CONDITION								
DA		N		SI FICHIER DATE VIDE						10IT	IDA00L = 0								
DA	100	M		SPACE TR03															
DA	120	M		'FICHIER DATE VIDE '															
DA	140			TR03-ZF080															
DA	160	P		F90TR															
DA	180	M		12	WSX0-ZCSRT7														
DA	200	P		F99SX															
DE		N		CONTROLE DATE						10BL									
DE	100	M		1-DA00-ZCCODA(1) XD11-ZDB0Y8															
DE	120	P		F93DT															
DE	140	*		SI DATE DU JOUR INVALIDE						99IT	EN-PRE = '5'								
DE	160	M		SPACE TR03-ZF080															
DE	180	STR		'DATE INVALIDE : '															
DE	200			1-DA00-ZCCODA(1)															
DE	220			DELIMITED BY SIZE															
DE	240			INTO TR03-ZF080															
DE	260	P		F90TR															
DE	280	M		12	WSX0-ZCSRT7														
DE	300	P		F99SX															
										ocages en masse FONCTION: 93									
										NVTY CONDITION									
										10BL									
										99IT XD11-ZDB0Y8 NOT NUMERIC									
										99IT XD11-ZDB03 > '12'									
										OR XD11-ZDB03 = '00'									
										OR XD11-ZDB04 > '31'									
										OR XD11-ZDB04 = '00'									
										99IT XD11-ZDB04 > '30'									
										AN (XD11-ZDB03 = '04'									
										OR XD11-ZDB03 = '06'									
										OR XD11-ZDB03 = '09'									
										OR XD11-ZDB03 = '11')									
										99IT XD11-ZDB04 > '29'									
										AN XD11-ZDB03 = '02'									
										99IT XD11-ZDB01 NOT = '19'									
										AN XD11-ZDB01 NOT = '20'									
										99IT XD11-ZDB01 = '19'									
										AN XD11-ZDB02 = '00'									
										99EL									
										99IT XD11-ZCD09 NOT = ZERO									
										AN XD11-ZDB04 > '28'									
										AN XD11-ZDB03 = '02'									

* DT 330
* DT 340
* DT 350 M '5' EN-PRE
* DT 360 GT 10
* DT 400 M 'W' EN-PRE
* DT 410
* DT 500 M 01 XD11-ZCD09
* DT 510
* DT 520 C XD11-ZCD09 = XD11-ZDB029 / 4
* DT 526 C XD11-ZCD09 = XD11-ZDB029 -
* DT 530 (XD11-ZCD09 * 4)
* DT 540 M '5' EN-PRE
* DT 550
* DT 560

Je commence à insérer des lignes de code en fonction 04, car l'ouverture du fichier DATE est paramétrée à l'initialisation du programme. Il s'agit des vérifications du contenu de ce fichier. Pour contrôler le format, je fais appel à la fonction située en 93DT (générée par la macro ABDB11) par un PERFORM.

La fonction 93DT valorise la rubrique de travail EN-PRE à '5' en cas de date invalide. J'insère ensuite un test sur cette rubrique afin de renvoyer un message d'erreur à afficher dans le fichier TRACE ainsi qu'un code retour à 12, puis j'appelle la fin du programme.

```

TRAITEMENTS PROGRAMME P I7B210 Blocages déblocages en masse FONCTION: 05
A SS NLG OPE OPERANDE NVTY CONDITION
FB N SI FICHER FICBLO VIDE 10IT 5-E100-CPTENR = 0
FB 100 M SPACE TR03
FB 120 M 'FICBLO' TR03-ZCDDN8
FB 140 M '- FICHER DES COMPTES '
FB 160 TR03-ZLFIC
FB 180 M ': VIDE' TR03-ZX040
FB 200 P F90TR

```

Le contrôle du fichier FICBLO se fait en fonction 05, lors de l'étape d'ouverture des fichiers sans rupture, qui est gérée par Pacbase. Je vérifie le nombre d'enregistrements dans le fichier, grâce au compteur E100-CPTNER généré par Pacbase. Si le fichier est vide ce compteur est égal à zéro, et un message est inséré dans le fichier TRACE.

Dans la boucle Pacbase, les fonctions de 01 à 05 ne sont parcourues qu'une seule fois. C'est donc logiquement dans ces fonctions que l'on place toutes les initialisations de début de programme.

En fonction 50 commence le traitement spécifique, le corps du programme. Le fichier de sortie BLOMAJ étant à l'identique du fichier en entrée FICBLO avec une rubrique supplémentaire pour le code retour, je commence par copier le contenu du segment d'entrée dans le segment de sortie. J'initialise ensuite la rubrique du code retour ZCRUTO à zéro.

A	SS	NLG	OPE	OPERANDE	NVTY	CONDITION
			N	TRAITEMENTS SPECIFIQUES	05BL	
AB			N	INI CODE RETOUR	10BL	
AB	100	M		E100 S100		
AB	120	M		ZERO S100-ZCRUT0		
BB			N	SI CODE ACTION NON RENSEIGNE	10IT	E100-CCATG = SPACE
BB	100	M		'0102' S100-ZCRUT0		OR E100-CCATG = LOW-VALUE
BE			N	SI CODE ACTION <> C, M, S	10IT	E100-CCATG NOT = 'C'
BE	100	M		'0103' S100-ZCRUT0		AN E100-CCATG NOT = 'M'
BE	120					AN E100-CCATG NOT = 'S'
BE	140					AN (E100-CCATG NOT = SPACE
BE	160					AN E100-CCATG NOT = LOW-VALUE)
BE	180					AN S100-ZCRUT0 = ZERO
BH			N	SI NUMERO COMPTE NON RENSEIGNE	10IT	S100-ZCRUT0 = ZERO
BH	100	M		'0104' S100-ZCRUT0		AN (E100-NCNNE = SPACE
BH	120					OR E100-NCNNE = LOW-VALUE)
BK			N	SI DATE BLO NON RENSEIGNEE C/M	10IT	(E100-CCATG = 'C'
BK	100	M		'0107' S100-ZCRUT0		OR E100-CCATG = 'M')
BK	110					AN (E100-DPSBL = SPACE
BK	120					OR E100-DPSBL = LOW-VALUE)
BK	130					AN S100-ZCRUT0 = ZERO
BM			N	VERIF DATE BLOCAGE	10IT	S100-ZCRUT0 = ZERO
BM	100	M		E100-DPSBL XD11-ZDB0Y8		
BM	120	P		F93DT		
BM	140	*		SI DATE BLO INVALIDE C/M	99IT	(E100-CCATG = 'C'
BM	150					OR E100-CCATG = 'M')
BM	160	M		'0108' S100-ZCRUT0		AN EN-PRE = '5'
BQ			N	SI MONTANT BLO NON RENSEIGNE C/M	10IT	(E100-CCATG = 'C'
BQ	100	M		'0109' S100-ZCRUT0		OR E100-CCATG = 'M')
BQ	110					AN E100-MCZTD2 = ZERO
BQ	120					AN S100-ZCRUT0 = ZERO
BT			N	SI ENTITE BLO NON RENSEIGNEE C	10IT	E100-CCATG = 'C'
BT	100	M		'0110' S100-ZCRUT0		AN (E100-CUGGS = SPACE
BT	110					OR E100-CUGGS = LOW-VALUE)
BT	120					AN S100-ZCRUT0 = ZERO
BW			N	SI MOTIF BLO NON RENSEIGNE C/M	10IT	(E100-CCATG = 'C'
BW	80					OR E100-CCATG = 'M')
BW	100	M		'0111' S100-ZCRUT0		AN (E100-CCZMG = SPACE
BW	110					OR E100-CCZMG = LOW-VALUE)
BW	120					AN S100-ZCRUT0 = ZERO
BZ			N	SI REF BLO NON RENSEIGNEE M/S	10IT	(E100-CCATG = 'S'
BZ	80					OR E100-CCATG = 'M')
BZ	100	M		'0112' S100-ZCRUT0		AN (E100-NPSBL = SPACE
BZ	110					OR E100-NPSBL = LOW-VALUE)
BZ	120					AN S100-ZCRUT0 = ZERO
CB			N	AJOUT, MODIF OU SUP BLOCAGE	10IT	S100-ZCRUT0 = ZERO

La partie suivante effectue tour à tour toutes les vérifications d'alimentation des rubriques du fichier FICBLO, en testant si elles sont alimentées à SPACE ou LOW-VALUE. Je refais appel à la fonction de vérification du format d'une date 93DT pour la date de blocage renseignée dans les enregistrements. Chaque condition, si remplie, génère un code retour différent et alimente la rubrique ZCRUT0.

CD	N	ALIM C4HEADER	15BL
CD 100	P	F91BB	
CE	N	SI CODE ACTION = C -> AJOUT BLO	15IT E100-CCATG = 'C'
CE 100	P	F91CB	
CE 120	*	SI RETOUR C4 OK	99IT WS69-ZCRETC = '0'
CE 140	*	SI RETOUR APPEL AM011A OK	AN ZC06-ZCRETC = '0'
CE 160	M	'0000' S100-ZCRUTO	
CH	N	SI CODE ACTION = M -> MODIF BLO	15IT E100-CCATG = 'M'
CH 100	P	F91CC	
CQ	N	SI RETOUR C4 OK	20IT WS69-ZCRETC = '0'
CQ 100	*	SI RETOUR APPEL AM036A OK	AN ZG06-ZCRETC = '0'
CQ 120	P	F91CB	
CV	N	SI RETOUR C4 OK	25IT WS69-ZCRETC = '0'
CV 100	*	SI RETOUR APPEL AM011A OK	AN ZC06-ZCRETC = '0'
CV 120	M	'0000' S100-ZCRUTO	
KK	N	SI CODE ACTION = S -> DEBLOCAGE	15IT E100-CCATG = 'S'
KK 100	P	F91CB	
KK 120	*	SI RETOUR C4 OK	99IT WS69-ZCRETC = '0'
KK 140	*	SI RETOUR APPEL AM011A OK	AN ZC06-ZCRETC = '0'
KK 160	M	'0000' S100-ZCRUTO	
MA	N	ECRITURE SORTIE	10BL
MA 120	P	F90S1	

Les PERFORM appelant les sous-fonctions 91CB et 91CC gèrent respectivement les alimentations nécessaires à l'accès aux points d'entrée AM011 et AM016, et le code retour à afficher dans ZCRUTO si l'appel à un point d'entrée échoue (par exemple si le numéro de compte est invalide).

Une fois l'ensemble des contrôles passés avec succès, l'enregistrement peut être écrit dans le fichier de sortie MAJBLO grâce à la sous-fonction 90S1.

À la compilation, mon programme obtient une note PQC de 100/100.

Réalisation des tests

Une fois le programme compilé, je réalise les cas de test prévus initialement afin de valider son bon fonctionnement.

Afin de passer chaque cas de test je prépare les différents fichiers nécessaires (le programme nécessitant deux fichiers en entrée, chaque cas de test aura ses deux fichiers entrants portant l'extension « .ECHAXX » et ses deux fichiers sortants également).

Pour exécuter un programme COBOL, un fichier JCL (Job Control Language) est nécessaire. Le JCL donne les instructions d'exécution à la machine, et fonctionne de la même façon qu'un script.

```

1 //I7B210 JOB (UGXX,TSO0),DISF,CLASS=C,MSGCLASS=K,REGION=0K,
2 // NOTIFY=&SYSUID
3 /**
4 /*******
5 /** BIBLIOTHEQUES
6 /*******
7 /**
8 //JOB LIB DD DISP=SHR,DSN=RVX.CWXX.2TH40DEV.LOAD
9 // DD DISP=SHR,DSN=RSX.NEXX.NEULMD10
10 // DD DISP=SHR,DSN=RSX.NEXX.NEALMD10
11 // DD DISP=SHR,DSN=RSX.NEXX.NEMLMD90
12 // DD DISP=SHR,DSN=RSX.NEXX.NERLMD2D
13 // DD DISP=SHR,DSN=RSX.NEXX.NERLMD00
14 /**
15 /*******
16 /** DELETE DES FICHIERS DE SORTIE
17 /*******
18 /**
19 //DELETE EXEC PGM=EQQDELD
20 /**
21 /*******
22 /** CONTROLE DE L'EXISTENCE DES BLOCAGES
23 /*******
24 /**
25 //I7B20010 EXEC PGM=IKJEFT01,DYNAMNBR=20
26 /**
27 //SYSOUT DD SYSOUT=*
28 //SYSUDUMP DD SYSOUT=*
29 //SYSABOUT DD SYSOUT=*
30 //SYSDBOU DD SYSOUT=*
31 //SYSOUA DD SYSOUT=*
32 //SYSTSPRT DD SYSOUT=*
33 /**
34 /*******
35 /** FICHIERS EN ENTREE
36 /*******
37 //FICBLO DD DSN=PUTUSR.$PERM.ISI13B.I7B210.D41111AA.ECH01,
38 // DISP=SHR
39 //DATE DD DSN=PUTUSR.$PERM.MDV.I8C410AA.ECH01,
40 // DISP=SHR
41 /*******
42 /** FICHIERS EN SORTIE
43 /*******
44 //BLOMAJ DD DSN=PUTUSR.$PERM.ISI13B.I7B210.I7B210AA.ECH01,

```

Pour exécuter le programme I7B210 sur la machine, il faut soumettre le JCL dans UDC (ou sous TSO), ce qui lui fait appeler le programme. Le JCL est divisé en différentes étapes (steps), à chaque étape correspond une instruction.

Un JCL est un fichier script qui contient des instructions permettant de contrôler l'exécution et l'enchaînement des modules développés avec Pacbase ou COBOL. Il contient également le chemin des fichiers sur le système d'exploitation, les moyens de connexion à la base de données, dispose d'utilitaires de tri, gestion de fichiers, copie, etc.

Un JCL pourra donc exécuter un enchaînement de plusieurs programmes, les uns après les autres.

Je soumetts ce JCL à quatre reprises (une pour chaque cas de test) en prenant soin de renommer les fichiers de sortie pour ne pas les écraser, afin de passer tous les cas initialement prévus au plan de test. Pour chaque cas, je rédige un paragraphe exposant les résultats obtenus, et je les compare à ceux attendus avec pour objectif qu'ils correspondent parfaitement.

01 : Cas passant

FICHIERS EN ENTRÉE :

FICBLO : PUTUSR.\$PERM.ISI13B.I7B210.D41111AA.ECH01

X11externeCGL.hce	SX11externeCGL.hce	I7B210.jcl	PUTUSR.\$PERM.ISI13B.I7B210.D41111AA.ECH01
-----1-----2-----3-----4-----5-----6-----7-----8-----			
000001	C7575070002S20190318000000000000150ZZZZZZZ11234567890ABCDEF00001		
000002	M7575010008F2019031300000000000014ZZZZZZZ1189D09R54VLWGENPA00001		
000003	S7575020008A2019031300000000000007ZZZZZ Z189D09R54VL5GE25M00001		
000004	7575010005C2018041300000000000005ZZZZZ Z11234567890ABCDEF00001		
000005	X0755000001A2018041300000000000005ZZZZZ Z11234567890ABCDEF00001		
000006	C 2019031300000000000005ZZZZZ Z11234567890ABCDEF00001		
000007	C7575210001P 00000000000005ZZZZZ Z11234567890ABCDEF00001		
000008	M0005230002I123120160000000000005ZZZZZZZ11234567890ABCDEF00001		
000009	C7575130003K2019031300000000000001001 Z11234567890ABCDEF00001		
000010	C7575070003T2019031300000000000005 Z11234567890ABCDEF00001		
000011	M0005130002J201903050000000000005ZZZZZ 1234567890ABCDEF00001		
000012	S7575040001G201903050000000000005ZZZZZ U1 00001		

DATE : PUTUSR.\$PERM.MDV.I8C410AA.ECH01

FICHIERS EN SORTIE :

BLOMAJ : PUTUSR.\$PERM.ISI13B.I7B210.I7B210AA.ECH01

X11externeCGL.hce	SX11externeCGL.hce	I7B210.jcl	PUTUSR.\$PERM.ISI13B.I7B210.I7B210AA.ECH01
-----1-----2-----3-----4-----5-----6-----7-----8-----9-----10-----			
000001	C7575070002S20190318000000000000150ZZZZZZZ11234567890ABCDEF00001		0000
000002	M7575010008F2019031300000000000014ZZZZZZZ1189D09R54VLWGENPA00001		0000
000003	S7575020008A2019031300000000000007ZZZZZ Z189D09R54VL5GE25M00001		0000
000004	7575010005C2018041300000000000005ZZZZZ Z11234567890ABCDEF00001		0102
000005	X0755000001A2018041300000000000005ZZZZZ Z11234567890ABCDEF00001		0103
000006	C 2019031300000000000005ZZZZZ Z11234567890ABCDEF00001		0104
000007	C7575210001P 00000000000005ZZZZZ Z11234567890ABCDEF00001		0107
000008	M0005230002I123120160000000000005ZZZZZZZ11234567890ABCDEF00001		0108
000009	C7575130003K2019031300000000000001001 Z11234567890ABCDEF00001		0109
000010	C7575070003T2019031300000000000005 Z11234567890ABCDEF00001		0110
000011	M0005130002J201903050000000000005ZZZZZ 1234567890ABCDEF00001		0111
000012	S7575040001G201903050000000000005ZZZZZ U1 00001		0112

TRACE : PUTUSR.\$PERM.ISI13B.I7B210.I7B20TAA.ECH01

X11externeCGL.hce	SX11externeCGL.hce	I7B210.jcl	PUTUSR.\$PERM.ISI13B.I7B210.I7B20TAA.ECH01
-----1-----2-----3-----4-----5-----6-----7-----8-----			
000001	I7B210260420191440436689 26/04/2019		
000002	FICBLO NB D ENREG LUS	00000012	
000003	BLOMAJ NB D ENREG ECRITS	00000012	
000004	I7B21026042019144043		

Le motif de blocage a bien été créé :

Numéro de compte : 7575070002S EUR -
Livret B - ZZZZZ Default entity

Montants bloqués

Total des montants bloqués : 3.211,00 EUR
Dont sous réserve d'encaissement : 3.211,00 EUR

Date de début	Montant	Origine du blocage	Date de fin	Entité
000000	500,00 EUR	Blocage SRE		Entité par défaut
000000	500,00 EUR	Blocage SRE		Entité par défaut
000000	500,00 EUR	Blocage SRE		Entité par défaut
000000	500,00 EUR	Blocage SRE		Entité par défaut
000000	500,00 EUR	Blocage SRE		Entité par défaut
000000	500,00 EUR	Blocage SRE		Entité par défaut
13-03-2019	5,00 EUR	Blocage SRE		Entité par défaut
14-03-2019	8,00 EUR	Blocage SRE		Entité par défaut
14-03-2019	8,00 EUR	Blocage SRE		Entité par défaut
15-03-2019	8,00 EUR	Blocage SRE		Entité par défaut
15-03-2019	8,00 EUR	Blocage SRE		Entité par défaut
15-03-2019	8,00 EUR	Blocage SRE		Entité par défaut
15-03-2019	8,00 EUR	Blocage SRE		Entité par défaut
15-03-2019	8,00 EUR	Blocage SRE		Entité par défaut
18-03-2019	150,00 EUR	Blocage SRE		Entité par défaut

La modification a bien été prise en compte :

Numéro de compte : 7575010008F EUR -

Pré-Livret A PPH et Personne Morale - ZZZZZ Default entity

Montants bloqués

Total des montants bloqués : 20,00 EUR
Dont sous réserve d'encaissement : 20,00 EUR

Date de début	Montant	Origine du blocage	Date de fin	Entité
13-03-2019	6,00 EUR	Blocage SRE		Entité par défaut
13-03-2019	6,00 EUR	Blocage SRE		Entité par défaut

Numéro de compte : 7575010008F EUR -

Pré-Livret A PPH et Personne Morale - ZZZZZ Default entity

Montants bloqués

Total des montants bloqués : 20,00 EUR
Dont sous réserve d'encaissement : 20,00 EUR

Date de début	Montant	Origine du blocage	Date de fin	Entité
13-03-2019	14,00 EUR	Blocage SRE		Entité par défaut
13-03-2019	6,00 EUR	Blocage SRE		Entité par défaut

La suppression s'est bien effectuée :

Numéro de compte : 7575020008A EUR -

Livret A PPH et Personne Morale - ZZZZZ Default entity

Montants bloqués

Total des montants bloqués : 7,00 EUR
Dont sous réserve d'encaissement : 7,00 EUR

Date de début	Montant	Origine du blocage	Date de fin	Entité
13-03-2019	7,00 EUR	Blocage SRE		Entité par défaut

Numéro de compte : 7575020008A EUR -

Livret A PPH et Personne Morale - ZZZZZ Default entity

Montants bloqués

Total des montants bloqués : 0,00 EUR

Code retour: 00

17B210:JOB24227 [CC 0000]

Les fiches de tests font partie des livrables attendus par le client, disponibles à la consultation sur le réseau sécurisé La Banque Postale. À ce titre, elles doivent être le plus explicite possible. Pour chaque cas je commente les résultats, surligne ou encadre les zones importantes pour davantage de lisibilité.

Exemples complémentaires

Évolution de l'interface ISI49

L'interface ISI49 comporte plusieurs traitements qui déterminent l'envoi de courriers aux détenteurs des comptes sélectionnés. Ces programmes sont liés au service d'Éditique de La Banque Postale, puisqu'ils produisent des flux ensuite transmis à cette plate-forme pour l'émission des documents à envoyer.

Le client a demandé d'écarter de ce flux les comptes dont le propriétaire est décédé, pour ne plus envoyer de courrier dans ce cas-là. La modification doit être appliquée dans plusieurs programmes.

On me confie la modification de quatre programmes de cette chaîne :

- I8EI91 : Courriers de relance PEL
- I8EP71 : Courriers de mise à terme
- I8EP81 : Courriers prorogation impossible
- I8EU51 : Courriers MIRE suite à une opération rejetée

Je développe la modification d'un premier programme dans l'optique de pouvoir le réutiliser dans les trois autres. Je vais présenter ici mes modifications dans I8EI91, et ce sont exactement les mêmes qui ont été appliquées dans les autres programmes.

L'évolution du code se traduit par un ajout d'un accès au puits de données DS505 (table DB2, aussi appelée DRD05) et d'un SELECT pour vérifier si un compte comporte un motif de blocage "décès" actif. S'il y a une correspondance avec un ou plusieurs enregistrements en entrée, ils ne seront pas reconduits dans le fichier de sortie.

```
VA Pac 3.5 V04 Tenue de compte - AM (TC) xxUMY983.PB12.C4M.7146
MACRO-STRUCTURES DU PROGRAMME I8EI91 FLUX RELANCE PEL

A MACRO NL 5 : SIGNIFICATION OU VALEUR DES PARAMETRES D V
ABAB01 : EC/00/E=MMW0-NCNCO/E=NCNCO/
ABAB01 10 : ED/00/E=MMW0-NCNCO/E=NCNCO/
ABAB71 : 10/S1/ MOTBLO_1/DRD05/x/ * SELECT DS505
ABAB9A : CSTYPEVT/EA/FB1U/ * SYNCHRO ARMIDE
ABAB91 : EC/INPUT/EC00/XB/SM/TR/x/E=M-CZA01/ * ERREUR VSAM
ABAB91 10 : ED/INPUT/ED00/XB/SM/TR/x/E=M-CZA01/ * ERREUR VSAM
ABAB99 : XB/XC07/
ABCB02 : PA/00/E=MSH0-ZCSRT7/12/12/12/SM/HM/DT/
ABCB02 * : N/TR/TR03-ZL040/E=TR03-ZX010/ * RECUP PARM ARMIDE
ABCC03 : HM/E=XD41-ZHEXP/ * CONTROLE HEURE
ABDB11 : 4/3/2/1/DT/ * CONTROLE DATE
ABDC06 : DA/AX3DAT/ * CALCUL DATE
ABIB01 : * AFFICHAGE CONSTANTES PROGRAMME
ABIB02 10 : EB/ * DISPLAYS COMPTEURS FICHIERS
ABIB02 20 : EC/ * DISPLAYS COMPTEURS FICHIERS
ABIB02 25 : ED/ * DISPLAYS COMPTEURS FICHIERS
ABIB02 30 : SA/ * DISPLAYS COMPTEURS FICHIERS
ABIB02 40 : SB/ * DISPLAYS COMPTEURS FICHIERS

D: C1 CH:
```

Pour ce faire j'ajoute et je paramètre la macro ABAB71, qui génère le code nécessaire pour effectuer un SELECT dans le puits DS505.

```

VA Pac 3.5 V04 Tenue de compte - AM (TC)                                xxUMY983.PB12.C4M.7146
TRAITEMENTS PROGRAMME P 18E191 FLUX RELANCE PEL                        FONCTION: 95

A SS NLG OPE OPERANDE                                                    NVTY CONDITION
* S1      N LECTURE SIMPLE DE DRD05                                     10BL
* S1 20 M 'DRD05' XC28-ZCTAB
* S1 30 M 'SELECT' XC28-ZLFAD
* S1 40 M 'F95S1' XC28-ZCF5F
* S1 100 EXQ SELECT                                                    99BL
* S1 110      MOTBLO_1,
S1 115      MOTBLO_2, MOTBLO_3,
S1 120      MOTBLO_4, MOTBLO_5,
S1 125      MOTBLO_6, MOTBLO_7,
S1 130      MOTBLO_8, MOTBLO_9,
S1 135      MOTBLO_10, MOTBLO_11,
S1 140      MOTBLO_12, MOTBLO_13,
S1 145      MOTBLO_14, MOTBLO_15,
S1 150      MOTBLO_16, MOTBLO_17,
S1 155      MOTBLO_18, MOTBLO_19,
S1 160      MOTBLO_20, DATBLO_1,
S1 165      DATBLO_2, DATBLO_3,
S1 170      DATBLO_4, DATBLO_5,

O: C1 CH:

```

Après cela je complète la requête générée en sous-fonction 95S1, en ajoutant le nom des colonnes sélectionnées manquantes (qui représentent tous les motifs de blocages possibles, les dates de début et les dates de fin de blocage, ainsi que le nombre d'occurrences de blocage pour un compte), le nom des rubriques du segment Pacbase miroir de la table (PD05), et le critère de sélection de la clause WHERE.

Dans le programme initial, les enregistrements sont écartés à la fin du traitement après avoir passé tous les tests, si l'indicateur INDIC est alimenté à N. Je rajoute un test supplémentaire pour contrôler l'existence du motif de blocage qui m'intéresse. Il commence en sous-fonction 50BO.

```

VA Pac 3.5 V04 Tenue de compte - AM (TC)                                xxUMY983.PB12.C4M.7146
TRAITEMENTS PROGRAMME P 18E191 FLUX RELANCE PEL                        FONCTION: 50

A SS NLG OPE OPERANDE                                                    NVTY CONDITION
BL 100 x ENREGISTREMENT ECARTE
BL 110 M 'N' WMW0-INDIC
-----
BO      N ACCES PUIT DS505                                     15IT WMW0-INDIC NOT = 'N'
BO 100 x APPEL PARTITIONNEMENT
BO 120 P F91DB
BO 140 x APPEL DRD05
BO 160 M WM05-NCNTH4 PD05-NCZTH
BO 180 P F95S1
-----
BP      N SI COMPTE TROUVE                                     20IT SQLCODE = 0
BP 100 INI PM05
BP 120 M PD05 PM05
-----
BQ      N BOUCLE SUR OCCURENCES BLOCAGE                         2500 1 PM05-ZQATC6
BQ 100 AD 65 PM05-DPSB01(J50BQR)
BQ 120      WMW0-DPSBD
BQ 140 AD 65 PM05-DPSD01(J50BQR) 99IT PM05-DPSD01(J50BQR) >

O: C1 CH:

```

```

VA Pac 3.5 V04 Tenue de compte - AM (TC) xxUMY983.PB12.C4M.7146
TRAITEMENTS PROGRAMME P I8EI91 FLUX RELANCE PEL FONCTION: 50

A SS NLG OPE OPERANDE NVTY CONDITION
BQ 160 MMW0-DPSBF SPACE
BQ 180 M SPACES MMW0-DPSBF 99EL
-----
BR N SI MOTIF DECES ACTIF TROUVE 30IT PH05-CCZM01(J50BQR) = 'X7'
BR 100 x ECARTER L'ENREGISTREMENT AN MMW0-DPSBF = SPACE
BR 120 M 'N' MMW0-INDIC
-----
BU N ACCES FICHIER REF SIROCCO 15IT MMW0-INDIC = SPACE
BU 100 M SPACE MMW0-NCNCO
BU 105 M MMW0-NCZCO MMW0-NCNCO
BU 110 P F95EC
BU 120 x SI NON TROUVE SIROCCO 99IT M-EC00-CF = ZERO
BU 130 M 1-EB00 SB00
BU 132 P F905B
BU 135 M SPACE TR03
BU 140 STR MMW0-NCZCO 'NON TROUVE '
BU 150 'DANS LE REFÉRENTIEL SIROCCO'
BU 170 DELIMITED BY SIZE

D: C1 CH:

```

Le programme accède au puits DS505 à l'aide du numéro de compte partitionné. Si le compte est trouvé, le traitement boucle sur les occurrences de blocage contenues dans la table pour ce compte. Si le motif de blocage “décès” X7 est trouvé, INDIC est alimenté à N pour écarter l'enregistrement.

Test de non-régression après modification de I8RP11

L'une des demandes d'évolution d'un programme existant qui m'ont été confiées est celle de I8RP11 (Extraction des données soldes Thaler) de l'interface ISI36. Elle engendre la modification de critères de sélection des arrêtés dans un puits de données.

Un plan de test en cas de modification d'un programme nécessite seulement un cas avant modification, et un cas après modification. Il est recommandé de le compléter par un test de non-régression, pour vérifier que l'ajout de code n'a pas changé le fonctionnement attendu du programme.

N° séquence cas de tests	ACTION [description des actions ou modes opératoires conduisant à la situation à tester]	RESULTATS ATTENDUS [description du résultat attendu – ex. : copie d'écran, message, valorisation donnée, ...]	RESULTATS OBTENUS [description du résultat obtenu – ex. : copie d'écran, message, valorisation donnée, ...]	STATUT [OK ou KO : selon la conformité du résultat obtenu par rapport à l'attendu]
1	- Cas passant avant modification : 0000031153504002M : compte PEL clos 0006297553508629S : compte PEL clos avec critères de sélection incorrects	✓ Code retour 04 ✓ Fichier SOLDE : MCNIA1 compte 1 non alimenté MCNIA1 compte 2 non alimenté	Conformes aux résultats attendus	OK
2	- Cas passant avant modification : 0000031153504002M : compte PEL clos 0006297553508629S : compte PEL avec critères de sélection incorrects	✓ Code retour 04 ✓ Fichier SOLDE : MCNIA1 compte 1 alimenté MCNIA1 compte 2 non alimenté	Conformes aux résultats attendus	OK
3	- Test de non régression	✓ Pas de différence entre les deux fichiers de sortie, excepté sur les zones MCNIA1, MCNIA2 et MCNIA3	Conformes aux résultats attendus	OK

Un test de non-régression en COBOL s'effectue avec un outil comparateur sous TSO, aussi disponible dans UDC. Dans un premier temps, il faut lui soumettre le fichier de sortie avant modification, et le fichier de sortie après modification. Ensuite, on lui indique les colonnes qu'il doit ignorer lors de la comparaison, qui correspondent aux rubriques concernées par la modification du programme.

```

VIEW          TS0SX11.XGPL504.SUPERC.LIST          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000001 1  ISRSUPC -   MVS/PDF FILE/LINE/WORD/BYTE/SFOR COMPARE UTILITY- ISPF
000002 NEW: PUTUSR.$PERM.I8RP11.CSRP1BAB.ECH02          OLD: PUTUS
000003
000004          WORD COMPARE SUMMARY AND STATISTICS
000005
000006
000007
000008          1972 NUMBER OF WORD MATCHES          0 TOTAL CHANGES (PAIRED+
000009          0 NEW FILE WORD INSERTIONS          0 NEW FILE LINES CHANGED
000010          0 OLD FILE WORD DELETIONS          0 OLD FILE LINES CHANGED
000011          1972 NEW FILE WORDS PROCESSED          114 NEW FILE LINES PROCESS
000012          1972 OLD FILE WORDS PROCESSED          114 OLD FILE LINES PROCESS
000013
000014 LISTING-TYPE = DELTA          COMPARE-COLUMNS =          988          LONGEST-LINE
000015 PROCESS OPTIONS USED: NONE
000016
000017 THE FOLLOWING PROCESS STATEMENTS (USING COLUMNS 1:72) WERE PROCESSED:
000018          CMPCOLM 1:135 148:1000

```

Par exemple dans le programme I8RP11, la modification concerne les rubriques MCNIA1, MCNIA2 et MCNIA3, situées entre les caractères 135 et 148 d'un enregistrement du fichier de sortie. Aucune différence n'a été constatée en dehors de la zone exclue, ce qui signifie que la modification apportée au programme n'a pas engendré de dysfonctionnement dans la partie existante. Le test est donc un succès.

Synthèse du projet

Bien que débutante en programmation après trois mois de POEI à l'EPSE, je n'ai pas eu de difficulté à m'adapter à cette technologie que je ne connaissais pas. Plutôt habituée à la programmation Web, cette expérience m'a permis de découvrir l'utilisation de plusieurs nouveaux outils :

- UDC
- Le système d'exploitation z/OS et TSO, en émulation
- Pacbase et quelques notions de COBOL
- Le JCL
- Xped (outil de débogage)

La création de I7B210 a été une étape très importante pour moi, dans la mesure où il s'agissait du premier programme que j'ai dû développer pour zThaler. Il m'a permis de bien comprendre la structure de la boucle Pacbase, l'alimentation des rubriques et l'utilisation de macros qui sont utilisées très régulièrement sur les traitements Batch. J'ai développé deux autres programmes complets, la partie principale de mon activité étant la modification de composants existants comme I8EI91 ou I8RP11.

Après avoir pris mes marques, il m'est apparu que la réalisation d'application en environnement Mainframe aurait pu être l'occasion pour moi d'accéder à un poste d'analyste ou de travailler plus directement sur des problématiques fonctionnelles. Étant avant tout intéressée par le développement d'applications Web et ayant un intérêt pour la technologie, j'ai préféré demander à intégrer un autre CdS afin d'acquérir une expérience dans d'autres domaines.

Parallèlement, et afin de répondre à un besoin personnel, j'ai commencé le développement de l'application 3dCatalog, ce qui a été l'occasion pour moi d'approfondir mes compétences dans la création d'applications Web.

Projet 3D Catalog

Compétences mises en œuvre

Voici la liste des compétences mises en œuvre par ce projet. Pour des détails supplémentaires concernant l'acquisition de chaque compétence, merci de vous référer au tableau référentiel de l'IPI en Annexe 4.

Qualité et sécurisation du code réalisé

- Formaliser, identifier les résultats attendus
- Respecter des contraintes, les recommandations qualité de la norme en vigueur pour l'architecture des logiciels
- Anticiper les évolutions
- Respecter une norme de présentation des écrans et documents de sortie
- Concevoir des programmes avec une orientation objets
- Garantir un accès sécurisé aux données

Audit, conception, méthode de projet

- Formaliser des processus, les règles de gestion et d'organisation des données de l'entreprise
- Concevoir une solution logicielle
- Anticiper des répercussions

Réalisation d'application logicielles

- Encapsuler des solutions logicielles spécifiques dans des services logiciels génériques
- Produire du logiciel générique réutilisable et partageable

Contexte du projet

Depuis plusieurs années, l'un de mes loisirs principaux est le rendu 3D. Il s'agit de la création de scènes en 3D, qui sont ensuite traduites en images où sera restituée la projection 3D, les textures (apparence des surfaces des objets visualisés) mais surtout tous les effets d'éclairage (ombres, réflexions, etc.).

Maintenant que j'ai intégré le domaine du développement informatique, l'application 3D Catalog s'est naturellement imposée pour répondre à mes objectifs personnels et professionnels. Au cours de son évolution, je mets en pratique diverses règles méthodologiques qui m'ont été enseignées, autant à l'école qu'en entreprise. Le développement de cette application mobilise des compétences qui sont représentatives pour l'obtention du titre de Concepteur Développeur d'Applications Numériques de l'IPI.

Expression du besoin

J'utilise le logiciel DAZ Studio, qui se présente comme un studio photo virtuel où l'on place des modèles 3D dans un environnement pour construire une scène. Au fil des années, j'ai accumulé une bibliothèque d'éléments assez conséquente qui contient des personnages, des décors, des jeux de lumière, des collections de textures...

La plupart du contenu que j'utilise est sous licence DAZ Studio. Leurs modèles 3D fonctionnent souvent avec un système de génération (assimilable à une version du logiciel). Par exemple, le modèle Victoria, qui est un personnage, existe dans sept générations différentes. Cela veut dire que chaque génération sera installée dans un répertoire différent sur le disque dur. Avec cette organisation, je ne peux donc pas déterminer facilement combien de modèles, ou lesquels, sont présents sur ma machine.

L'idée d'une application pour faciliter l'inventaire du contenu dont je dispose m'est alors venue. Ainsi, 3D Catalog répond à trois besoins :

- Gérer mon contenu pour DAZ Studio
- Pouvoir partager en ligne des informations sur le contenu de ma bibliothèque
- Apprendre à développer une application à l'aide d'un framework Java, pour approfondir mes connaissances de ce langage
- Présenter mes compétences acquises pour valider le titre de l'IPI

Contraintes

3D Catalog étant réalisé dans un cadre privé, son avancée dépend de ma disponibilité sur mon temps personnel.

Étant donné mon besoin, j'aurais pu opter pour le développement d'une application de bureau (desktop). Cependant, au vu de mes compétences mais aussi de la possibilité de partager plus tard l'application en ligne, je me suis tournée vers le développement d'application Web en Java. Ce type de technologie me semble également nécessaire à acquérir pour la suite de mon parcours professionnel.

Livrables attendus

Mon objectif est de présenter une version de l'application non déployée, qui dispose des fonctionnalités essentielles suivantes :

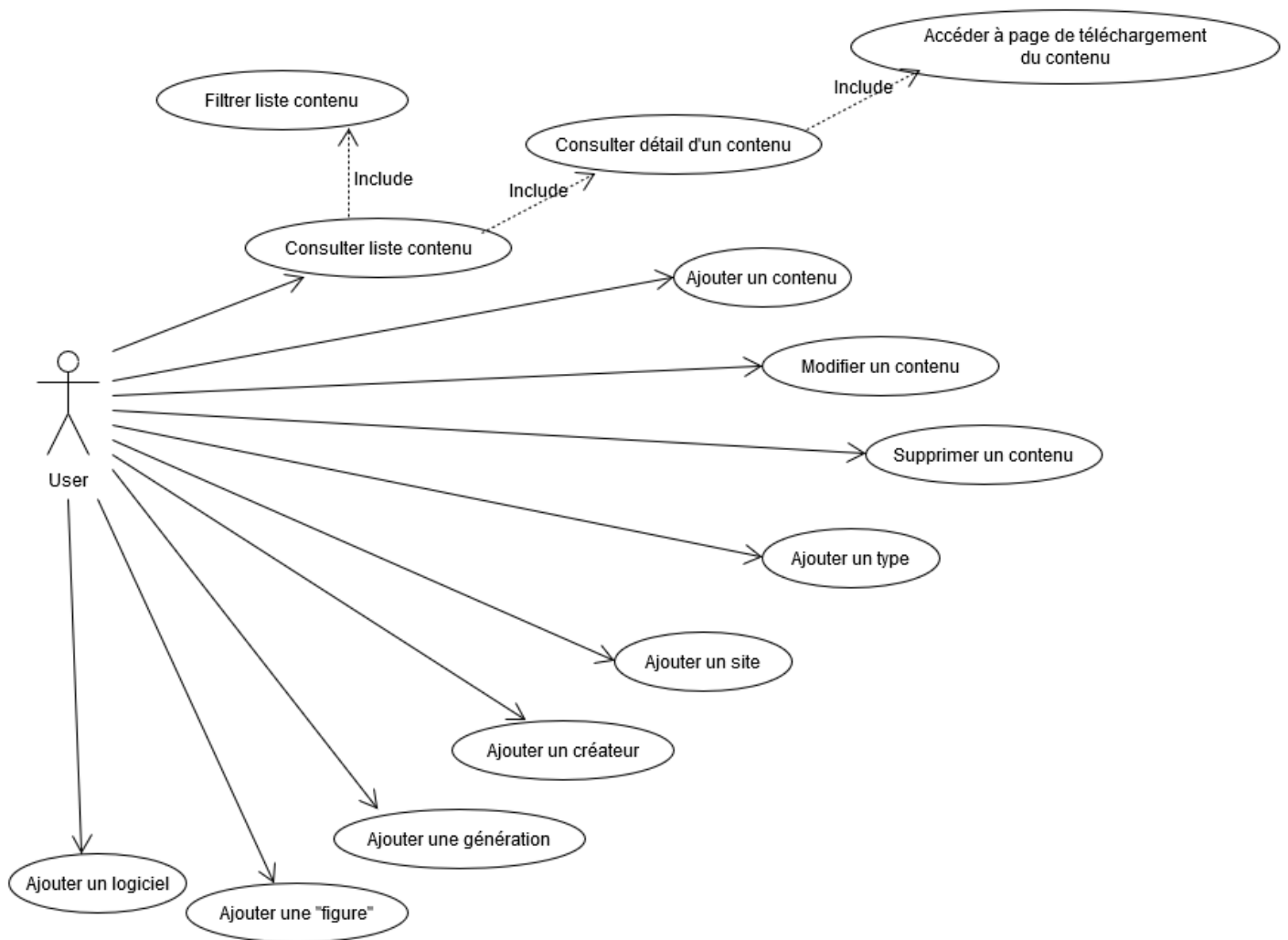
- Afficher la liste de contenus enregistrés en base de données
- Filtrer les contenus par leur type
- Ajouter un contenu

Analyse du projet

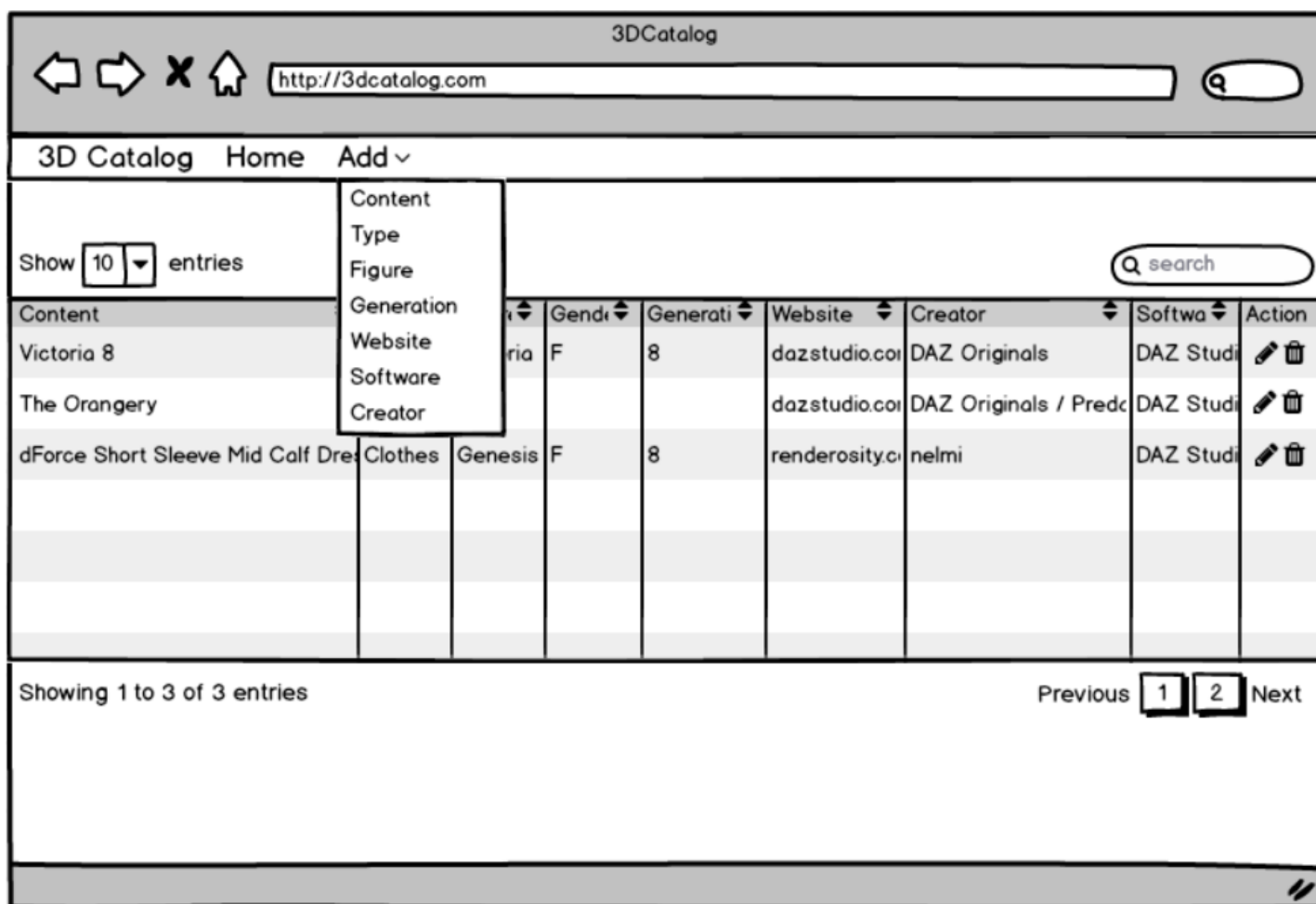
Spécifications fonctionnelles

3D Catalog est une application Web qui permet à l'utilisateur de gérer une liste de contenu pour l'application DAZ Studio.

Le diagramme de cas d'utilisation suivant met en évidence les fonctionnalités de la solution :



Le maquettage est la première étape de conception de l'application. Je conçois les écrans à l'aide de l'outil Balsamiq (<https://balsamiq.com/>). L'écran principal affiche et permet de filtrer la liste des contenus de la bibliothèque. Pour la réalisation, j'envisage d'utiliser DataTables (<https://www.datatables.net/>) qui est un plug-in pour la librairie JavaScript jQuery.



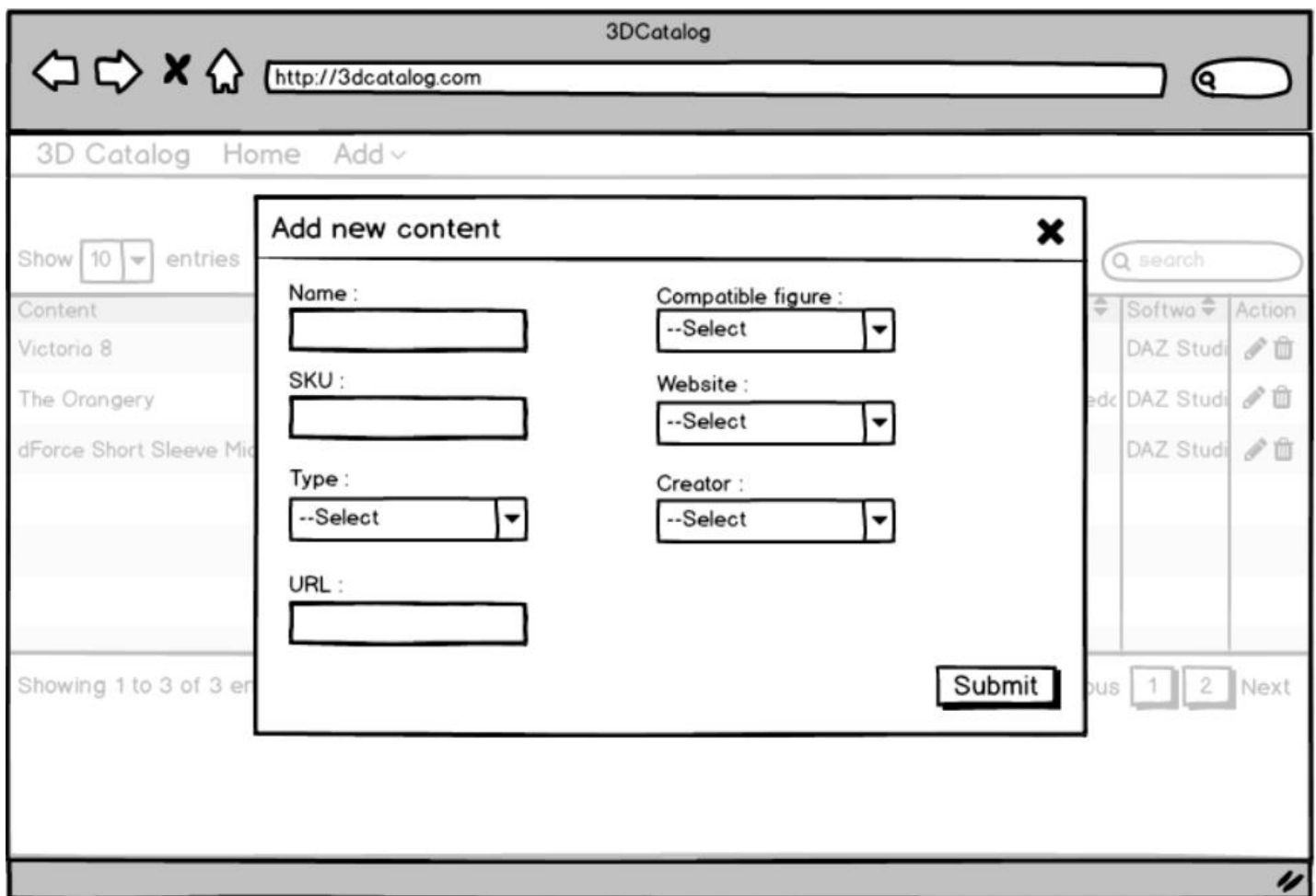
Cet écran est le premier affiché au démarrage de l'application. Il est composé de ces éléments :

- Un menu permettant de revenir à cette page à tout moment (onglet 'Home') et d'afficher une liste déroulante d'ajouts possibles pour l'application (onglet 'Add')
- Un tableau affichant la liste des contenus enregistrés
- Une liste déroulante pour contrôler le nombre de contenus affichés par page
- Une barre de recherche pour filtrer les résultats du tableau
- Un compteur indiquant le nombre de contenus total ainsi que la plage affichée
- La pagination du tableau

À partir de cet écran, l'utilisateur a la possibilité d'accéder au détail d'un contenu en cliquant sur la ligne correspondante dans le tableau, ce qui l'affichera dans une boîte de dialogue modale.

Le bouton 'Modifier' (icône crayon dans la colonne 'Action' du tableau) ouvre une boîte de dialogue modale semblable à celle qui affiche le détail d'un contenu, avec des champs textes et des listes à sélectionner pré-remplis dont l'utilisateur peut modifier le contenu.

Le bouton 'Supprimer' (icône poubelle dans la colonne 'Action' du tableau) ne fait que désactiver un contenu pour qu'il ne s'affiche plus dans le tableau. Il ne l'efface pas de la base de données.



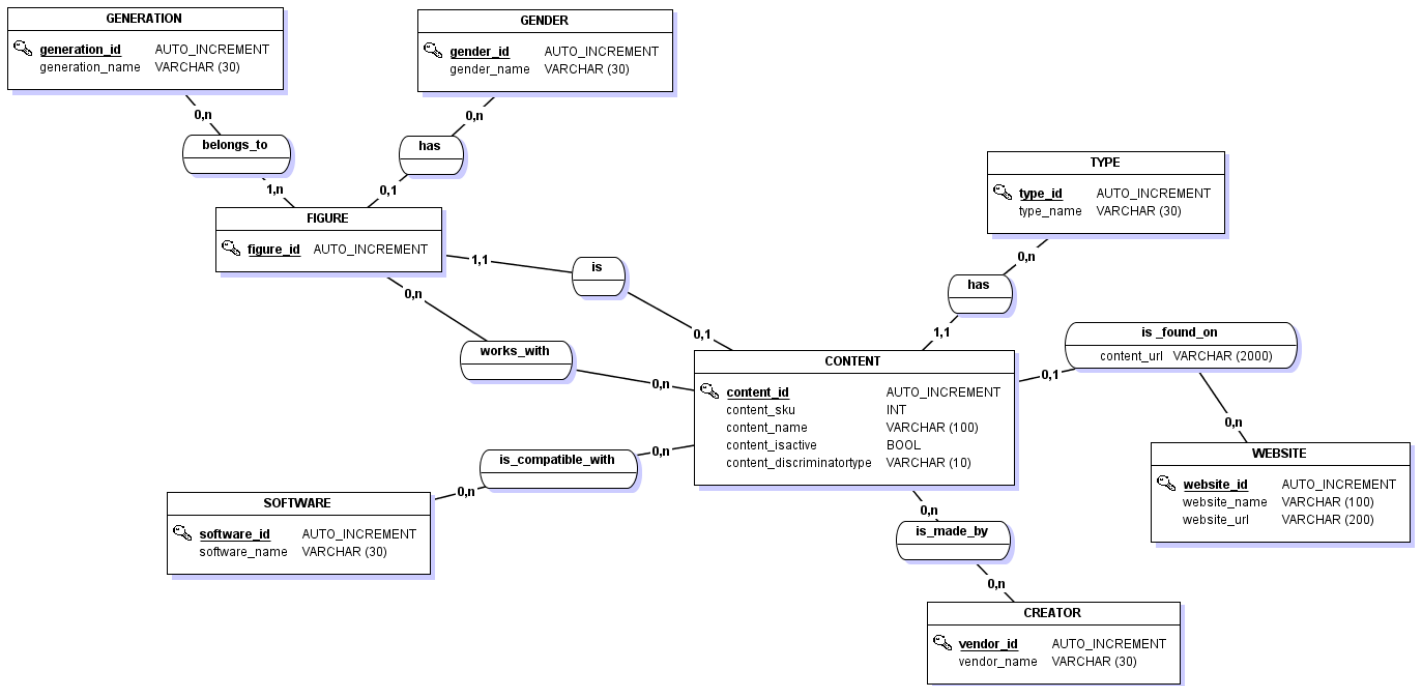
Les boîtes de dialogue modales d'ajout se ressemblant beaucoup, j'ai pensé le design de celle d'ajout de nouveau contenu uniquement. Elle s'affiche lorsque l'utilisateur clique sur 'Add', puis 'Content'. Après avoir rempli les champs nécessaires et cliqué sur 'Submit', un encadré au-dessus du tableau indique dans un cadre vert si le contenu a bien été enregistré, ou affiche un message d'erreur dans un cadre rouge.

Spécifications techniques

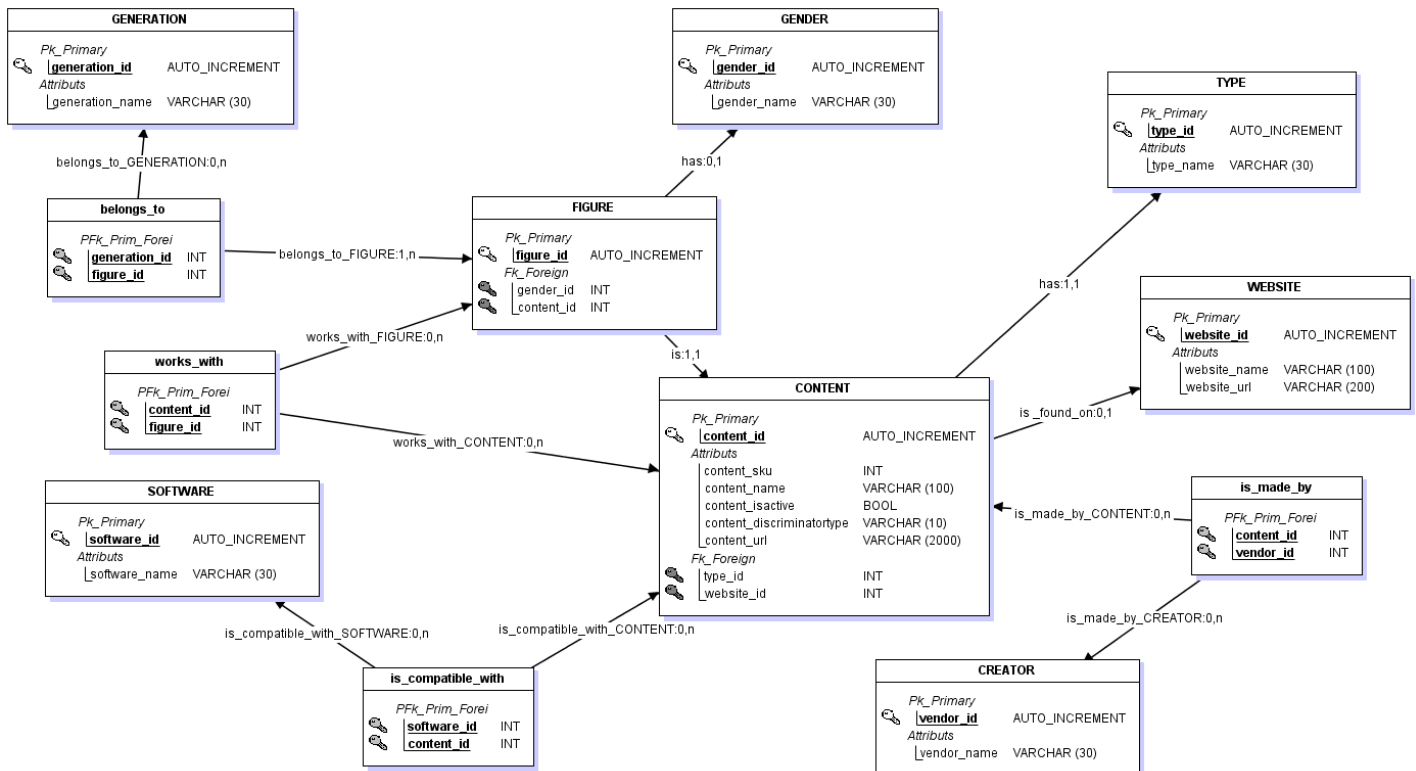
3D Catalog est développé en Java avec le framework Spring et Spring MVC. L'accès aux données est réalisé avec l'ORM Hibernate. L'application est lancée par un serveur Tomcat. La base de données est hébergée par MySQL.

Le modèle de données de la base s'articule autour de la table principale 'CONTENT'. Les autres tables correspondent aux critères pour filtrer les résultats affichés dans l'application.

Le Modèle Conceptuel de Données de la base se présente comme ceci :



Voici le Modèle Logique de Données que j'ai généré, pour une meilleure compréhension du script de création de ma base de données :



Pour l'architecture de l'application, je suis le modèle d'architecture Modèle Vue Contrôleurs (MVC). L'utilisateur effectue des actions à travers les contrôleurs. Les contrôleurs sont chargés d'assurer la synchronisation entre les vues et les modèles. Les modèles sont des objets représentant les données échangées avec l'utilisateur. Les vues sont l'ensemble des composants qui permettent de présenter les données. Une fois les traitements réalisés, les contrôleurs renvoient les informations adéquates à la vue pour les afficher pour l'utilisateur.

3D Catalog suit aussi une architecture d'application multicouche (N-tiers) : une couche de présentation, une couche métier et une couche d'accès aux données.

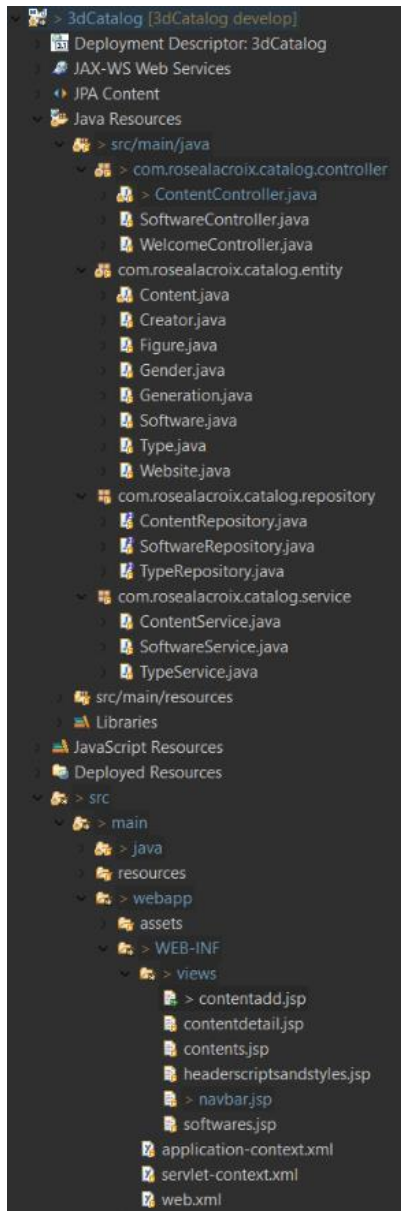
La couche de présentation est la partie Front de l'application, composée de Java Server Pages (JSP) contenant du code HTML et des balises Java Standard Tag Library (JSTL). Elle comporte aussi les contrôleurs Spring MVC, qui réagissent aux actions de l'utilisateur et interrogent la couche métier.

La couche métier est composée de services Spring. C'est dans un service que le traitement applicatif est effectué comme un traitement de vérification pour l'application d'une règle métier. Pour l'application 3D Catalog, les fonctionnalités métiers consistent surtout à réaliser un simple appel à la couche d'accès aux données.

La couche d'accès aux données est un ensemble de repositories qui permettent d'interagir avec la base de données. Ils sont pris en charge Spring Data grâce à une simple déclaration d'interface Java. Ils rendent la création d'une couche d'accès aux données plus rapide à implémenter.

Réalisation

La mise en place de l'architecture de 3D Catalog s'est faite autour du modèle MVC et de l'architecture multicouche. La capture d'écran ci-dessous montre le découpage de l'application.



Quatre packages ont été créés afin de faciliter la lecture et de s'y retrouver plus facilement dans les classes Java de l'application. Un premier package 'controller' contient les contrôleurs de l'application. On y retrouve les classes qui reçoivent les informations lorsque l'utilisateur effectue des opérations sur les vues. Elles présentent l'annotation @Controller pour être utilisées avec le module Spring MVC.

Le second package 'entity' contient des classes qui facilitent la manipulation des tables de la base de données. Il y en a une par table en base. Elles sont reconnues par l'annotation @Entity.

Le troisième package 'repository' fait le lien avec la base de données grâce à Spring Data, avec l'annotation @Repository dans chacune de ses interfaces.

Le dernier package 'service' remplit la fonctionnalité centrale de l'architecture. Les classes qu'il contient renvoient aux classes qui ont la charge de réaliser les fonctionnalités principales. Elles présentent l'annotation @Service.

Les vues sont contenues dans le dossier 'webapp/WEB-INF/views'.

Spring est un framework non intrusif, très souple d'utilisation. Il permet d'écrire beaucoup moins de lignes de code, de par sa capacité à comprendre le rôle des classes dans l'application selon l'annotation utilisée. De ce fait, le code est beaucoup plus facile à maintenir.

La page principale de l'application est générée par la JSP 'contents.jsp'. Elle contient un tableau, dont l'id est pris en paramètre du script jQuery appelant le module DataTables. Ce module ajoute automatiquement le tri par colonnes, le filtrage des résultats affichés, la pagination, etc... Il nécessite un tableau disposant d'un header <thead> et d'un body <tbody> au minimum.

Les imports de feuilles de styles et de scripts sont situés dans un fichier à part, 'headerscriptsandstyles.jsp', que j'inclus dans la partie <head> du code HTML de chaque page JSP. Cela permet de modifier un seul fichier pour ajouter un nouvel import. De la même manière, la barre de navigation de l'application est contenue dans un fichier externe.

Les balises <c:out> de la bibliothèque Core de JSTL permettent d'échapper les caractères spéciaux contenus dans du code HTML, ce qui protège l'application de l'injection de code HTML.

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Content</title>
<%@include file="headerscriptsandstyles.jsp"%>
</head>
<body>
<%@include file="navbar.jsp"%>

<table id="content_result" class="display">
<thead>
<tr>
<th>Name</th>
<th>SKU</th>
<th>Type</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<c:forEach items="${contents}" var="c">
<tr>
<td><c:out value="${c.name}" /></td>
<td><c:out value="${c.sku}" /></td>
<td><c:out value="${c.type.name}" /></td>
<td>
<c:forEach items="${c.softwares}" var="s">
<c:out value="${s.name}" />
</c:forEach>
</td>
</tr>
</c:forEach>
</tbody>
</table>

<script>
$(document).ready(function() {
    $('#content_result').DataTable();
});
</script>
</body>
</html>
```

```
package com.rosealacroix.catalog.controller;

import java.util.List;

@Controller
public class ContentController {

    @Autowired
    private ContentService contentService;
    @Autowired
    private TypeService typeService;

    @GetMapping(path="/content")
    public String getAll(Model model) {
        List<Content> contents = contentService.getAll();
        model.addAttribute("contents", contents);
        return "contents";
    }

    @GetMapping(path="/contentadd")
    public String displayForm(Model model, Content content) {
        List<Type> types = typeService.getAll();
        model.addAttribute("types", types);
        return "contentadd";
    }

    @PostMapping(path="/contentadd")
    public String submitContent(Content content) {
        contentService.save(content);
        return "contentdetail";
    }
}
```

Dans cette première version de l'application, j'affiche pour un contenu son Nom, son SKU (numéro d'identification propre au site d'où il vient), son type, et le logiciel pour lequel il a été créé.

Lorsque la vue a besoin de ces informations, elle appelle le contrôleur ContentController. Celui-ci contient l'annotation @Autowired associée à des attributs de type ContentService et TypeService, ce qui active l'injection automatique de dépendance (référence vers le Service associé).

Pour retourner la liste des contenus enregistrés dans la base de données, le contrôleur fait appel à la méthode getAll(), qui va récupérer une liste de contenus auprès du ContentService.

La méthode displayForm() permet de préparer les données pour l'affichage du formulaire d'ajout de

contenu. Il est nécessaire de disposer de la liste des types de contenus pour créer la liste grâce à l'élément `<select>` du formulaire et permettre d'associer un type au nouveau contenu. Cette méthode demande la liste de types au `TypeService`, et l'ajoute dans le modèle à l'aide de `addAttribute()` pour la rendre disponible à la vue.

La méthode `submitContent()` sert à transmettre le contenu du formulaire au `ContentService`, et retourne les informations à la page `'contentdetail.jsp'` lorsque le contenu est bien enregistré en base.

```
package com.rosealacroix.catalog.service;

import java.util.List;

@Service
public class ContentService {

    @Autowired
    private ContentRepository contentRepository;

    public List<Content> getAll() {
        return contentRepository.findAllWithRelations();
    }

    @Transactional
    public void save(Content content) {
        contentRepository.save(content);
    }
}
```

Le `ContentService` est lui-même lié par `@Autowired` au `ContentRepository`. Il dispose d'une méthode `getAll()` appelant la méthode `findAllWithRelations()` du `ContentRepository`.

Pour la méthode `save()`, l'annotation `@Transactional` est nécessaire pour que l'enregistrement en base de données aboutisse. Elle signale que pour effectuer cette opération, Spring doit démarrer une transaction (BEGIN) avec la base de données et la terminer par un COMMIT lorsque la méthode s'achève.

```
package com.rosealacroix.catalog.repository;

import java.util.List;

public interface ContentRepository extends JpaRepository<Content, Long> {

    @Query("select distinct c from Content c left join fetch c.softwares order by c.name")
    List<Content> findAllWithRelations();
}
```

Le `ContentRepository` contient l'annotation `@Query`, qui précise une requête sur une méthode. La requête est écrite en JPQL (JPA Query Language), un langage de requête objet.

Synthèse du projet

Le projet 3D Catalog me tient vraiment à cœur, car je souhaite continuer à le mettre en place personnellement. Réaliser une application seule, de la conception à la réalisation, représente un défi technique que je me suis lancée. C'est une expérience très formatrice, qui mobilise une grande partie des compétences que j'ai pu acquérir au cours de l'année.

En plus de la volonté de terminer cette application pour pouvoir l'utiliser de manière locale, je prévois de la faire évoluer en la déployant sur Heroku, une plateforme en ligne qui permet d'héberger et d'exécuter des applications Web. Cela me permettra de la tester sur un environnement autre que mon poste, et de me confronter à de nouvelles problématiques.

La découverte du framework Spring m'a convaincue, car il apporte beaucoup de gains de productivité. Il m'a fallu du temps pour comprendre son fonctionnement, à lire la documentation et parcourir le code source, étant donné qu'il prend en charge une bonne partie des problématiques d'architecture.

Je souhaite maintenant approfondir mes connaissances en essayant la conception et le développement d'applications plus complexes.

Conclusion générale

Mes expériences à CGI me font réaliser que l'aspect technique du métier de développeur est ce qui m'intéresse le plus pour l'instant. Je souhaite continuer à approfondir mes connaissances de la programmation Web, avec ou sans framework, et découvrir des technologies que je n'ai pas encore utilisées.

Pour continuer à m'améliorer, je compte terminer le projet 3D Catalog. Dans un premier temps, je travaillerai au déploiement de l'application Web. Une fois cette étape réalisée, je réfléchirai à la meilleure façon de réaliser une version de bureau afin de l'optimiser à mon usage personnel. J'ai pour pistes ElectronJs ou PySide2.

Cette année m'aura permis de me former et d'être formée sur diverses technologies, et de me rendre compte qu'il existe des systèmes très différents les uns des autres mais qui peuvent communiquer ensemble. Je continue à ce jour à travailler au sein du CdS d'Éditique pour Malakoff Médéric, avec la possibilité d'intervenir en parallèle sur un projet qui me permettrait de continuer ma montée en compétences en programmation.

Annexe 1 : Structure type d'un programme Pacbase

F01	Initialisation PAC
F01xx	Ouverture du fichier xx déclaré en –CD
	F02 à F04 Initialisation personnelle
F05	Lecture des fichiers sans ruptures
	F05xx lecture du fichier xx et positionnement des indicateurs de fin itération et fin de fichier xx (xx-FT et xx-FI)
	F06 à F09 Insertions spéciales
F10	Lecture des fichiers avec ruptures
F10xx	Gestions des zones de ruptures du fichier xx Positionne les indicateurs de fin Bascule les zones de lectures en zone de travail (xx00 → 1-xx00)
	F11 à F19 Insertions spéciales
F20	Fin de traitement (si FT= all 1 avec FT = somme des xx-FT)
F20xx	Fermeture du fichier xx
F2099	Stop RUN
F22	Calcul des ruptures
F22xx	Calcul des ruptures du fichier xx
F24	Calcul des configurations (synchronisations)
F24xx	Config du fichier xx (zone xx-CFn)
F26	Calcul des ruptures totales Calcul de RTPn RTDn NRP NRD
	F30 à F79 Insertion du corps du programme ATTENTION ceci est vrai dans un cas général, dans le cadre d'un programme de contrôle et MAJ automatique, les fonctions 30/33/36/39/42/45/73/76 sont utilisées On s'insère en général entre F50 et F70
F8E	Edition de l'état E (si condition mise en –D de l'état) 8EZZ Libellés 8E00 Structures 8E99 Ecritures physiques
F90	Ecriture des fichiers
F90xx	Ecriture du fichier xx
F9099-iter-fn	GOTO F05 (retour lecture suivante)
	F91 à F99 Insertion personnelle à appeler par PERFORM

Annexe 2 : Fiche d'évaluation stagiaire



CERTIFICATION PROFESSIONNELLE EVALUATION ENTREPRISE

STAGIAIRE

Nom : CHANGET
Prénom : Elodie

Certification visée :

ENTREPRISE

Nom de la société : CGI
Tuteur(trice) : Kaume LABRUE
Courriel : kaume.labrue@cgi.com

«tuteur_Civilite».

Afin de parfaire la validation de la certification professionnelle de votre stagiaire, nous vous remercions de remplir exhaustivement cet **original** et nous le retourner dans les **meilleurs délais**. Ce document sera examiné par le jury en vue de l'attribution de la certification professionnelle de votre stagiaire.

- | | |
|---|--|
| L'intégration dans l'équipe de travail est-elle satisfaisante ? | <input checked="" type="radio"/> Oui / <input type="radio"/> Non |
| Le comportement en situation professionnelle est-il adapté à la culture de l'entreprise ? | <input checked="" type="radio"/> Oui / <input type="radio"/> Non |
| Le comportement relationnel est-il adapté au métier visé ? | <input checked="" type="radio"/> Oui / <input type="radio"/> Non |
| La capacité de travail fournie correspond-elle au niveau d'un professionnel du métier ? | <input checked="" type="radio"/> Oui / <input type="radio"/> Non |
| Le niveau de responsabilités attendu est-il satisfait ? | <input checked="" type="radio"/> Oui / <input type="radio"/> Non |
| Des questions sont-elles posées à bon escient ? | <input checked="" type="radio"/> Oui / <input type="radio"/> Non |
| Des difficultés en relation avec le niveau de responsabilités sont-elles surmontées ? | <input checked="" type="radio"/> Oui / <input type="radio"/> Non |
| Le niveau d'obligation de réserve demandé est-il pris en compte ? | <input checked="" type="radio"/> Oui / <input type="radio"/> Non |
| Les activités effectuées concourent-elles à la couverture du champ de compétences ? | <input checked="" type="radio"/> Oui / <input type="radio"/> Non |
| Le métier associé à la certification visée est-il compris ? | <input checked="" type="radio"/> Oui / <input type="radio"/> Non |
| Le stagiaire est-il apte à occuper la fonction visée même comme débutant ? | <input checked="" type="radio"/> Oui / <input type="radio"/> Non |

Appréciation générale suite au stage en entreprise :

Elodie s'est montrée très investie dans l'ensemble des tâches qui lui ont été confiées.
Elle s'est très bien adaptée au contexte parfois difficile du travail en centre de service, avec des jalons et des délais souvent serrés.
Elle a effectué un travail de qualité avec une montée en compétence rapide. Très bonne année de sa part.
Fait à LE HAVRE, le 08/11/2019

Signature du tuteur
et cachet de l'entreprise



Signature du stagiaire



Association ADIP, agréée
par la loi du 1^{er} juillet 1901
Siret : 409 301 677 00017 Code
APE : 85 29 A
N° organisme : 11 75 77 97 775

Centre de formation
44 bis, quai de Jemmapes 75019
Paris
Siège social
13, rue d'Assommoir 75010 Paris

minhnn@gn-pe-igs.fr
www.ipi-ecoles.com



© Institut de la Profession Interprète (2011)

Annexe 3 : Résultats du TOEIC passé à l'EPSI

AMANT	ABS
ANRAS	840
BENQUET	575
BILLARD	750
BONNAT	965
BOUCHET	650
BOUILLET	445
BOUMANS	925
BRISSAC	780
CHANGEY	960
DAGUSE	745
DE LEMOS	ABS
DORVILLE	835
DOUCET	875
DUBARRY	ABS
ECKLY	840
GONZALEZ	ABS
GIRONDIN	ABS
HAMON	795
HILAIRE	ABS
LAERA	910
LAURIN	705
LE MEUR	620
LEPERA	950
MARMIESSE	960
MOREAU	795
PICHERIT	935
REBUZZI	780
REDIEN	615

Annexe 4 : Référentiel IPI

Bloc de compétences : Qualité et sécurisation du code réalisé

Compétences ou capacités qui seront évaluées	Critères d'évaluation	Exemples d'activités et tâches	Activités pratiquées	Origine de l'acquisition	Preuves apportées & ref. annexe
- Formaliser, identifier les résultats attendus.	La liste de contrôle des attendus fonctionnels est paraphée.	Étude de l'existant. Rédaction du cahier des spécifications fonctionnelles.	- Développement basé sur specs - Rédaction de spec non formelle pour 3D Catalog	E (3M) B (1S)	- p18 à 28 - p37 à 41
- Respecter des contraintes, les recommandations qualité de la norme en vigueur pour l'architecture des logiciels. - Anticiper les évolutions. - Qualifier les risques.	Un plan d'assurance qualité est observé. L'application est organisée en couches indépendantes. Les règles métier sont encapsulées dans des services logiciels. L'accès aux données est réalisé par des services logiciels indépendants du mode de stockage. L'exécution de l'application est répartie entre un nombre d'ordinateurs adapté au contexte.	Conception/architecture d'applications logicielles. Conception de services métiers. Conception de services d'accès aux données. Détermination du nombre de tiers de l'application.	- Note compilation Pacbase supérieure à 95/100 - Respect architecture N-tiers - Services séparés faciles à maintenir - N/A	E (1M) B (4M) B (1M) N/A	- p25 - p41 - p42 à 44 - N/A

<p>- Respecter une norme de présentation des écrans et documents de sortie.</p>	<p>Un formulaire d'estimation des risques est rempli.</p> <p>Une norme de présentation des données est respectée. Les interfaces Homme/Machine sont validées.</p>	<p>Réalisation d'une interface homme/machine (IHM)</p> <p>Estimation, qualification des risques sécurité.</p> <p>Réalisation des maquettes de sorties interactives.</p> <p>Réalisation des maquettes de sortie imprimée.</p>	<p>- Maquette Balsamiq</p>	<p>B (1S)</p>	<p>- p37 à 39</p>
<p>- Concevoir des programmes avec une orientation objets.</p> <p>- Garantir un accès sécurisé aux données.</p>	<p>Une programmation orientée objets est utilisée.</p> <p>Le taux de réutilisation du code utile est > 80 %. Des gabarits sont utilisés. Une charte de nommage est utilisée.</p>	<p>Programmation de logiciels.</p>	<p>- Conception 3D Catalog</p> <p>- Base de données hébergée sur serveur différent de l'application, et accès par mot de passe</p>	<p>B (1M)</p> <p>B (1M)</p>	

<p>- Livrer du logiciel déverminé et conforme aux attentes.</p>	<p>Le taux de documentation interne du code est > 8 % et < 15 %.</p> <p>Les anomalies d'accès aux données ne génèrent pas d'interruption de l'exécution et sont répertoriées.</p> <p>Des outils de contrôle automatique du code sont utilisés. Aucun défaut visible ne persiste.</p>	<p>Programmation de l'accès aux données de l'entreprise.</p> <p>Tests unitaires. Préparation des jeux de tests.</p> <p>Contrôles de l'existence d'anomalies.</p> <p>Recettage du logiciel. Validation d'une étape du projet.</p>	<p>- Programme I7B210 et sa fiche de tests</p>	<p>E (3M)</p>	<p>- p18 à 28</p>
<p>- Clôturer une mission.</p>	<p>Les contraintes spécifiques au projet sont respectées. Un manuel d'assurance qualité est respecté. Une méthode de recettage est utilisée. L'étape du projet est validée.</p> <p>Le PV de réception du logiciel est validé.</p>	<p>Mise en exploitation.</p>	<p>- Livraison d'un composant et mise à jour du tableau de livraison</p>	<p>E (1M)</p>	<p>- p16</p>

Bloc de compétences : Audit, conception, méthode de projet

Compétences ou capacités qui seront évaluées	Critères d'évaluation	Exemples d'activités et tâches	Activités pratiquées	Origine de l'acquisition	Preuves apportées & ref. annexe
- Formaliser des processus, les règles de gestion et d'organisation des données de l'entreprise.	<p>La procédure du service utilisateur est formalisée et validée.</p> <p>La procédure du service utilisateur est conforme aux règles du système de management des services de l'entreprise.</p> <p>La circulation du document résultat du traitement prévu est matérialisée dans un diagramme de workflow.</p> <p>La proposition de reconstruction de la procédure est validée.</p> <p>La base de données est modélisée.</p>	<p>Étude de l'existant.</p> <p>Identification des procédures en place.</p> <p>Contrôle de la conformité des procédures utilisées avec la gouvernance de l'entreprise.</p> <p>Recensement des documents utilisés, identification de leur circulation et des acteurs concernés.</p> <p>Reconfiguration de procédure.</p> <p>Conception d'une base de données.</p>	- Conception 3D Catalog	B (1M)	- p37 à 41
- Concevoir des éléments logiciels réutilisables.	Une méthode de conception par objets est utilisée.	Conception de l'architecture applicative.	- Évolution interface ISI49	E (2S)	- p29 à 31

<p>- Produire du logiciel en équipe.</p> <p>- Remonter les alertes au(x) décideur(s).</p>	<p>Une méthode AGILE est utilisée.</p> <p>Absence de signaux d'alertes au point de contrôle du projet.</p>	<p>Programmation en équipe.</p> <p>Écriture de code.</p> <p>Coordination de l'avancement.</p>	<p>- Programmation en binôme avec un analyste</p> <p>- Échanges avec analyste qui remonte les problèmes au client s'il y en a</p>	<p>E (1M)</p> <p>E (1M)</p>	<p>- p18 et 19</p> <p>- p18 et 19</p>
<p>- Estimer des délais.</p>	<p>Les étapes du projet sont planifiées.</p>	<p>Planification des tâches du projet.</p>	<p>- Estimer mon reste à faire</p>	<p>E (5M)</p>	<p>- p12 et 13</p>
<p>- Concevoir une solution logicielle.</p> <p>- Anticiper des répercussions.</p>	<p>Le projet est conforme au schéma directeur de l'entreprise et respecte les principes d'urbanisation du S.I.</p> <p>Les spécifications fonctionnelles produites respectent le cahier des charges fourni.</p> <p>L'impact de modification est acceptable.</p>	<p>Conception de la solution logicielle.</p>	<p>- Conception 3D Catalog</p> <p>- Minimum de code écrit grâce à Spring, simple à maintenir</p>	<p>B (1M)</p> <p>B (1M)</p>	<p>- p37 à 41</p> <p>- P37 à 41</p>

Bloc de compétences : Réalisation d'applications logicielles

Compétences ou capacités qui seront évaluées	Critères d'évaluation	Exemples d'activités et tâches	Activités pratiquées	Origine de l'acquisition	Preuves apportées & ref. annexe
<ul style="list-style-type: none"> - Encapsuler des solutions logicielles spécifiques dans des services logiciels génériques. - Produire du logiciel générique réutilisable et partageable. - Intégrer des éléments logiciels hétérogènes et produire des exécutables livrables. - Modifier un algorithme sans générer de dysfonctionnements. - Contrôler des délais. 	<p>Le service d'accès aux données est opérationnel.</p> <p>Des services logiciels internes sont réutilisables. Des services logiciels sont partageables en local.</p> <p>Des services logiciels sont partageables à distance.</p> <p>Le logiciel est livrable, prêt pour la mise en production.</p> <p>La modification n'entraîne pas de régression fonctionnelle.</p> <p>Le compte-rendu d'activité est renseigné, les écarts sont constatés.</p>	<p>Programmation. Investigations documentaires fonctionnelles ou techniques complémentaires. Transcription des spécifications fonctionnelles en algorithmes. Transcription des algorithmes en code source. Compilation, déverminage du code source. Agglomération des différents éléments logiciels en unités de traitement, réalisation des tests unitaires.</p> <p>Mise à jour du planning de réalisation.</p>	<ul style="list-style-type: none"> - Services et repositories - Évolution interface ISI49 - 3D Catalog déployé sur Heroku - Code Pacbase compilé exécutable par un JCL - Tests de non régression - Estimer mon reste à faire et contrôler mon plan de charge 	<p>B (1M)</p> <p>E (2S)</p> <p>B (1M)</p> <p>E (1M)</p> <p>E (2M)</p> <p>E (5M)</p>	<ul style="list-style-type: none"> - p44 - p29 à 31 - p45 - p25 et 26 - p31 et 32 - p12 et 13

Bloc de compétences : Communiquer avec les acteurs du projet

Compétences ou capacités qui seront évaluées	Critères d'évaluation	Exemples d'activités et tâches	Activités pratiquées	Origine de l'acquisition	Preuves apportées & ref. annexe
<p>- User d'une communication professionnelle tant en français qu'en anglais.</p> <p>- Interagir efficacement dans un environnement de travail collaboratif.</p>	<p>Le compte-rendu de la réunion est validé.</p> <p>Le score du TOEIC est > 749</p>	<p>Élaboration et rédaction de documents techniques, commerciaux ou internes à destination, des utilisateurs, des clients ou des collaborateurs, ...</p>	<p>- Score de 960 au TOEIC</p>	<p>F (1S)</p>	<p>- Annexe 3</p>
	<p>Le document collectant l'expression des besoins des utilisateurs est validé.</p> <p>L'aide du logiciel est rédigée. La documentation du livrable est diffusée.</p> <p>La présentation est appréciée.</p> <p>Les utilisateurs sont opérationnels, le transfert des nouvelles compétences est validé.</p>	<p>Rédaction des spécifications fonctionnelles de la solution informatique.</p> <p>Écriture des interfaces homme/machine.</p> <p>Relations avec les clients. Animation de réunions de travail et interviews d'utilisateurs. Démonstrations, recettage de livrables. Formation des utilisateurs au logiciel.</p>	<p>- Programmation en binôme avec un analyste, Daily Meeting</p>	<p>E (1M)</p>	<p>- p18 et 19, p13</p>

Bloc de compétences : Adapter l'environnement d'exécution, échanger des données entre logiciels

Compétences ou capacités qui seront évaluées	Critères d'évaluation	Exemples d'activités et tâches	Activités pratiquées	Origine de l'acquisition	Preuves apportées & ref. annexe
<ul style="list-style-type: none"> - Réaliser des échanges de données informatisés (EDI). - Automatiser des traitements. - Programmer des scripts systèmes. 	Les données sont consolidées.	Réalisation d'un procédé d'échange de données informatisées.	- Échange données sur réseau sécurisé LBP	E (1M)	- p16 et 28
	La base de données tierce est accédée.	Rétro-documentation de logiciels et de bases de données.	- Programmation de programmes Batch	E (3M)	- p19 à 25
	L'interface d'échange de données est opérationnelle.	Consolidation, agrégation de données. Programmation de l'interface d'échange de données.			
	L'environnement de tests est opérationnel.	Réalisation d'un environnement de tests. Création, configuration de machines virtuelles. Installation, configuration de serveurs d'applications, Web et base de données. Écriture de scripts systèmes pour adapter l'environnement d'exécution.	- Écriture de JCL	E (1M)	- p25 et 26