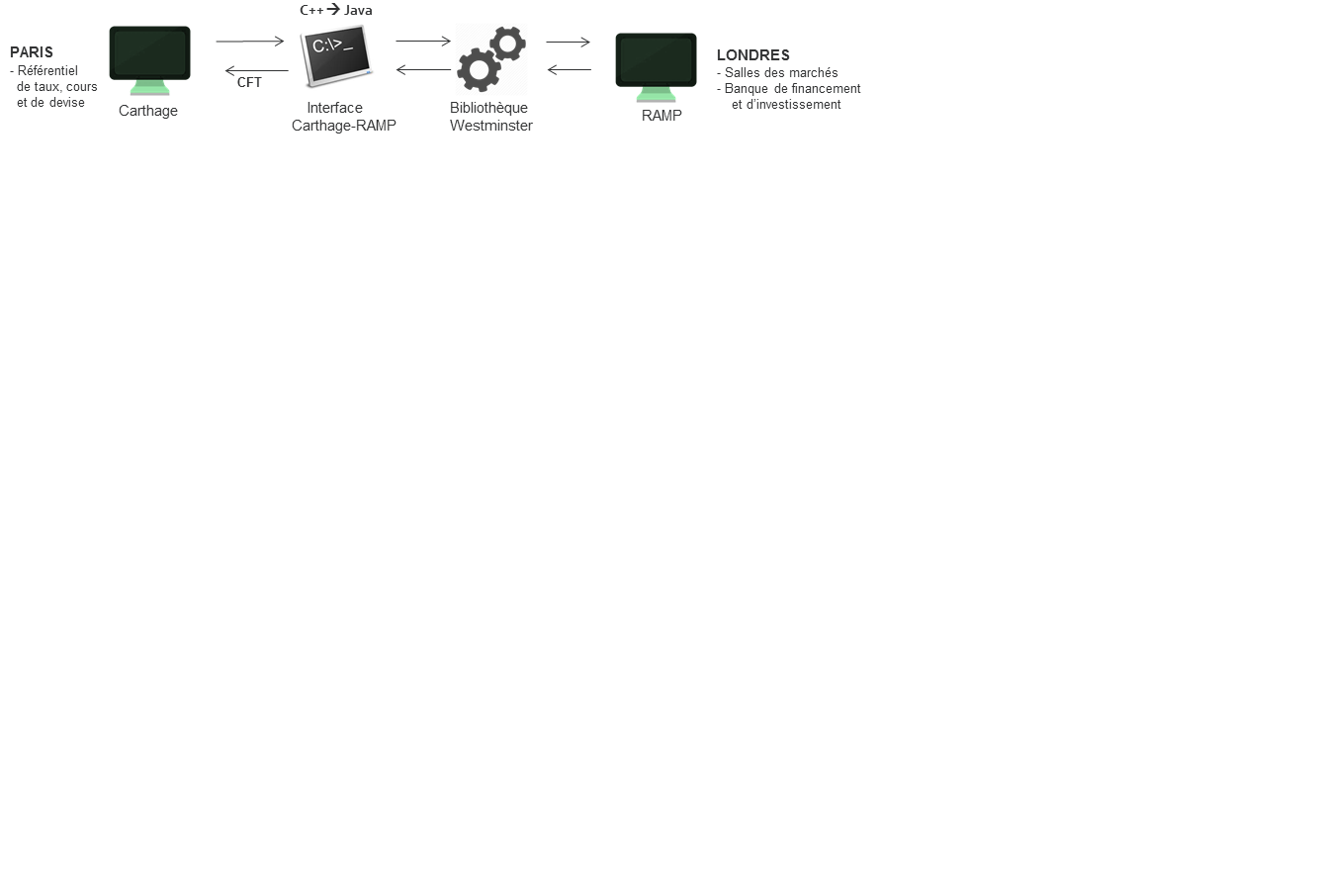
|  |
| --- |
|  |
| Rapport de stage |

2015-2016



Master 2 MIAGE-IF

**Auteur:** ISHOLA Hamed

**Tuteur enseignant :** CAILLOUX Olivier

**Responsable de stage :** BELLAL Teyib

****

Table des matières

[Résumé 2](#_Toc462820324)

[Remerciements 3](#_Toc462820325)

[1. Présentation du cadre de travail de l’entreprise 3](#_Toc462820326)

[1.1. Présentation du département ITG SIT 4](#_Toc462820327)

[1.2. Présentation de l’ALM Trésorerie 6](#_Toc462820328)

[1.3. Présentation du contexte de travail 7](#_Toc462820329)

[1.4. La mission 7](#_Toc462820330)

[2. Présentation du sujet 8](#_Toc462820331)

[2.1. Contexte applicatif (Application Carthage) 8](#_Toc462820332)

[2.1.1. Module Carthage-Ramp 10](#_Toc462820333)

[2.1.2. Système d’information RAMP 10](#_Toc462820334)

[2.1.3. Librairie Westminster 10](#_Toc462820335)

[2.1.4. Taux de cession interne 10](#_Toc462820336)

[2.1.5. Flux CFT 10](#_Toc462820337)

[2.1.6. Les périmètres de taux de l’application Carthage 11](#_Toc462820338)

[2.1.7. Description des instruments financiers utilisés 12](#_Toc462820339)

[Spot exchange rates (Cours de change Spot Forex) 12](#_Toc462820340)

[Taux Swap 12](#_Toc462820341)

[Swap de base 12](#_Toc462820342)

[3. Présentation du travail réalisé 14](#_Toc462820343)

[3.1. Migration application Carthage-Ramp (CarRamp) 14](#_Toc462820344)

[3.1.1. Description de l’architecture logicielle 14](#_Toc462820345)

[3.1.2. Les couches logicielles 17](#_Toc462820346)

[3.1.3. Couches applicatives 17](#_Toc462820347)

[3.1.4. Analyse et Conception de l’application 18](#_Toc462820348)

[3.1.4.1. Diagramme de cas d’utilisation 18](#_Toc462820349)

[4.1.4.3. Diagramme de classes 19](#_Toc462820350)

[4.1.4.4. Diagramme de séquence 22](#_Toc462820351)

[3.1.4.5. Tests unitaires 26](#_Toc462820352)

[Répartition du temps de travail 26](#_Toc462820353)

[Gestion des erreurs 27](#_Toc462820354)

[Typologie des erreurs 27](#_Toc462820355)

[Messages d’erreurs 27](#_Toc462820356)

[Gestion de la traçabilité 27](#_Toc462820357)

[Demandes d’évolutions 27](#_Toc462820358)

[Discounts factors 28](#_Toc462820359)

[Contributions WMI APAC 28](#_Toc462820360)

[Travaux intermédiaires 29](#_Toc462820361)

[Présentation des objectifs et du choix des technologies 29](#_Toc462820362)

[Objectifs 29](#_Toc462820363)

[Choix des technologies 29](#_Toc462820364)

[Choix organisationnels 30](#_Toc462820365)

[Analyse Personnel 31](#_Toc462820366)

[Regard critique (Recul par rapport au travail réalisé) 33](#_Toc462820367)

[Apport du master MIAGE 33](#_Toc462820368)

[Compétences acquises 34](#_Toc462820369)

[Planning du projet 35](#_Toc462820370)

[Glossaire 36](#_Toc462820371)

[Bibliographie 36](#_Toc462820372)

# Résumé

Ce stage a consisté dans un premier temps à la migration d’un module d’une application qui sert de référentiel groupe de la banque BNP Paribas vers une nouvelle technologie qui est le langage Java, et dans un second temps de prendre en charge les demandes d’évolution venant de la Maîtrise d’ouvrage (MOA), sur la nouvelle application.

Dans le but d’externaliser les activités informatiques du groupe, et d’homogénéiser l’ensemble des technologies des applications sous une seule technologie qui est le langage de programmation Java, le projet de migration de l’application Carthage a été lancé.

Cette migration a pour but d’externaliser les développements informatiques vers la Tiers Maintenance Applicative (TMA), qui est spécialisée dans la technologie Java.

Intégré dans l’équipe Carthage, mon travail consistait donc à :

* analyser et comprendre l’application existante.
* rétro-documenter l’application existante,
* Faire la conception globale de l’application en UML,
* Faire valider la conception de l’application par les concepteurs
* Redévelopper entièrement l’application en Java
* Participer à la phase de recette de l’application
* Analyser et implémenter les expressions des besoins pour les demandes d’évolution de l’application.

La dernière partie consiste donc à transférer les compétences de la nouvelle application à la Tiers Maintenance Applicative (TMA), qui elle assurera la maintenance et les éventuelles demandes évolutions de la maitrise d’ouvrage.

# Remerciements

Je remercie Monsieur Ferréol Thouzeau notre manager informatique et Teyib Bellal mon maître de stage de m’avoir accueilli au sein de l’équipe Carthage du département ITG SIT de BNP Paribas, et m’avoir permis de travailler sur des sujets aussi intéressants.

Je remercie très chaleureusement Teyib de m’avoir très bien encadré tout au long de mon passage dans son service. Merci de m’avoir donné des conseils et explications toujours très précis et avisés de ses talents financiers, fonctionnels, scientifiques et informatiques.

Merci à eux de m’avoir fait découvrir ce métier d’analyste et ses multiples missions essentielles.

Je tiens à remercier tout le groupe SIT ITG, auquel j’ai été rattaché depuis Avril 2016 à Octobre 2016, pour leur accueil chaleureux.

# Présentation du cadre de travail de l’entreprise

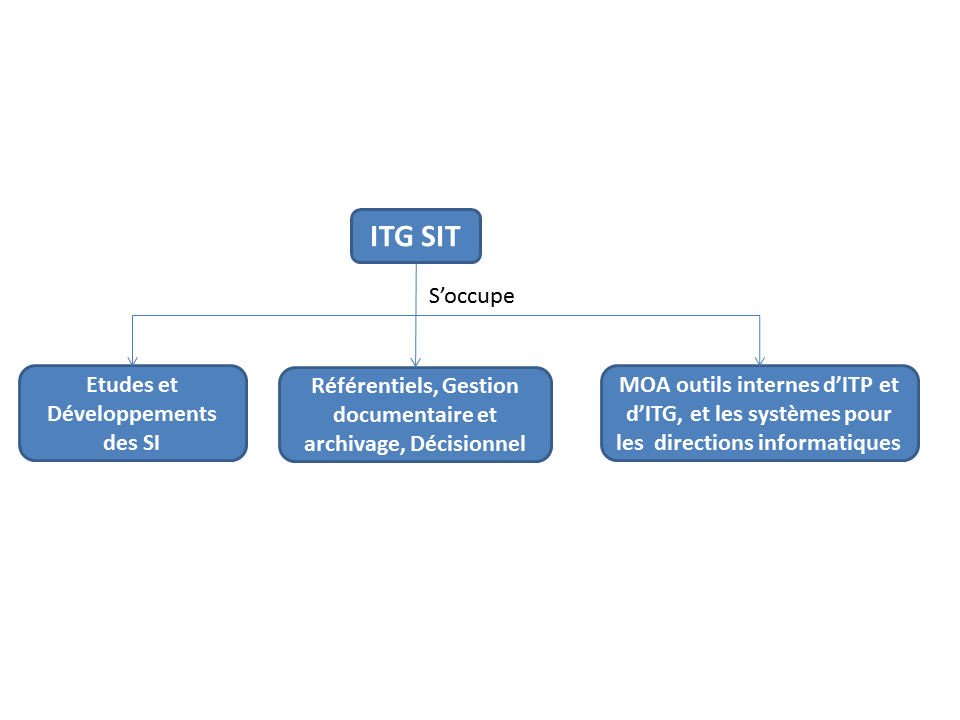
Le but de cette introduction est de rappeler, de façon précise, la mission que l’on m’a confiée pour mon stage de 6 mois au sein du service ALM Trésorerie du département ITG SIT de BNP Paribas, situé à Montreuil. Sous la direction du responsable de l’équipe, mon travail a consisté dans un premier temps à la migration d’un module de l’application référentiel de taux du groupe vers une nouvelle technologie qui est le langage java. Dans un second temps à analyser les demandes d’évolution de la MOA, et assister les utilisateurs pour les tâches de support applicatif.

Notre objectif était donc de suivre le processus de développement d’une application, en partant de la phase d’analyse des besoins, jusqu’à la mise en production de l’application.

Pour cela, nous avons fait une étude de faisabilité de la migration de l’application, et ensuite faire une analyse détaillée en passant par la rédaction d’un dossier de conception, des spécifications fonctionnelles.

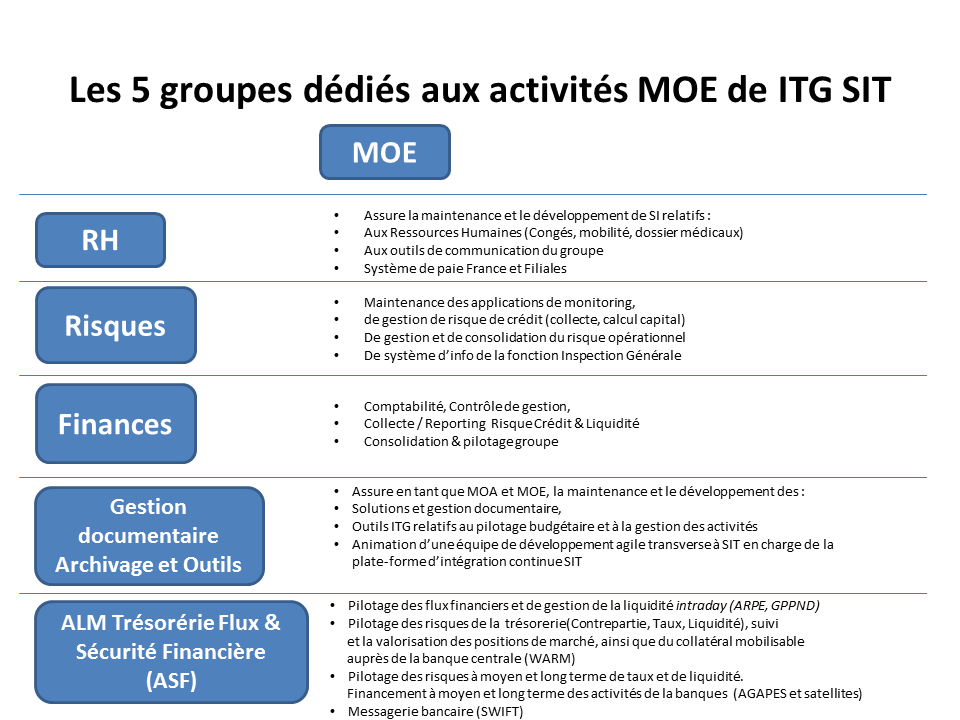
## Présentation du département ITG SIT

Le département ITG SIT de BNP Paribas assure la conception, le développement et la maintenance de projets ou d’applications pour ses clients : les pôles et les fonctions du groupe. Son objectif est de garantir la bonne application des normes et standards et de l’architecture.



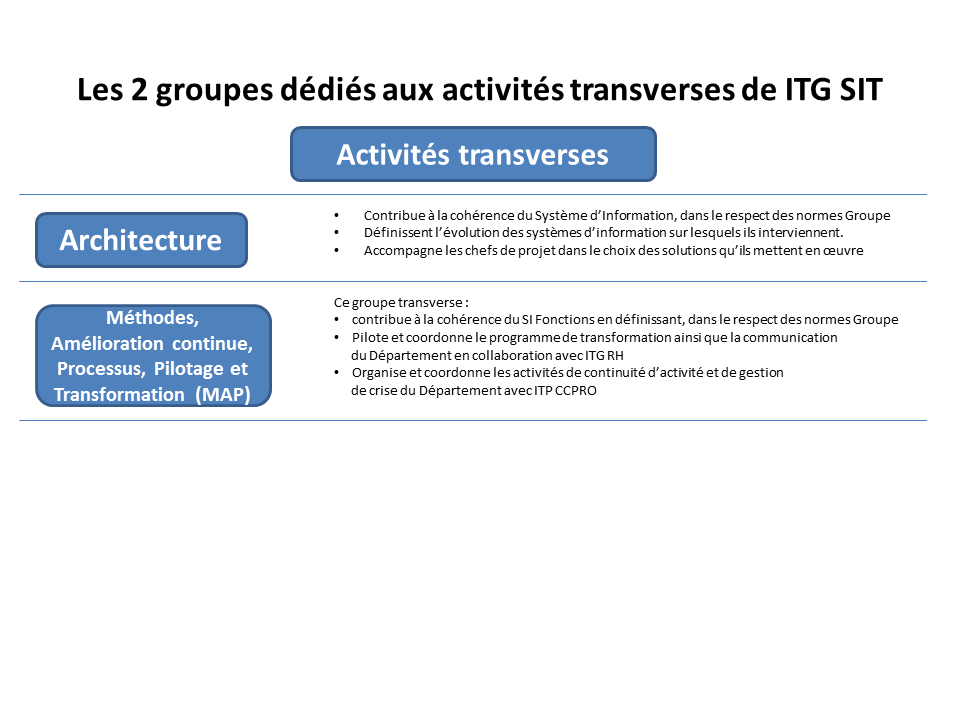
Les grandes missions du département ITG SIT consistent à assurer les activités : d’architecture, d’études et développement pour les systèmes d’informations des fonctions du groupe, ainsi que pour les systèmes transverses et mutualisés du Groupe (messagerie bancaires, ALM trésorerie, etc…).  
Le groupe assure les activités de la maîtrise d’ouvrage d’outils internes d’ITP et d’ITG ainsi que des systèmes pour les directions informatiques.  
il assure également les activités des centres d’expertises et de services partagés tels que les Référentiels, la Gestion documentaire et l’Archivage, le Décisionnel.

L’organisation du groupe ITG SIT s’articule autour de 5 groupes dédiés aux **activités de maitrise d’œuvre** des applications de projets gérés par ITG SIT, et 2 groupes consacrés aux **activités transverses** du département.



Les 2 groupes contribuant aux **activités transverses** sont : l’architecture, Pilotage et amélioration continue des processus.  
Le groupe **Architecture** définit les principes d’architecture logiciels, en définissant dans le respect des normes Groupe, le cadre des architectures dans lequel s’inscriront les projets menés au sein de SIT.

Le groupe **Amélioration Continue** contribue à la cohérence du Système d’information du groupe, en définissant dans le respect des normes Groupe, le cadre méthodologique et les processus dans lesquels s’inscrivent les patrimoines et les projets au sein d’ITG SIT, leurs démarches d’amélioration continue et assure également le pilotage du portefeuille projet et maintenance du département.



## Présentation de l’ALM Trésorerie

ALM signifie Asset Liability Management, ce terme anglais se traduit sous le mot Gestion Actif-Passif.

L’ALM Trésorerie a pour vocation de garantir l’équilibre entre l’actif et le passif de la banque à court terme, elle s’assure que la banque dispose de ressources suffisantes qui lui permettent d’assumer ses charges ou de rembourser les emprunts qu’elle a contractés.

La gestion d’actif-passif a pour objectif d’assurer un meilleur fonctionnement et augmenter les performances de la banque.

La gestion d’actif-passif peut se définir comme un mode de gestion à double objectif : d’une part, de garantir la solvabilité de l’entreprise, et d’autre part, d’améliorer son rendement.

Nous pouvons donc résumer, que le but de l’ALM Trésorerie est de :

* Mesurer et suivre les risques structurels : risque de taux, risque de change, risque de crédit et contrepartie, investissements et participations.
* Anticiper la structure du bilan, les conditions de marché, les ratios prudentiels (appelé gestion prévisionnelle).
* Rechercher de la liquidité sur les marchés financiers.
* Allouer les fonds propres, en refinançant ses différentes filiales.
* Calculer des indicateurs, afin de les envoyer au régulateur.
* Fournir les meilleurs taux pour améliorer les résultats du groupe.

## Présentation du contexte de travail

Le domaine ALM Trésorerie comporte des applications internationales critiques, nécessaires aux dispositifs fondamentaux du groupe BNP Paribas, le refinancement du groupe, la gestion de son bilan, l’identification et la simulation des risques de taux et de liquidité.

J’ai été rattaché aux applications Carthage et AGAPES, ces applications ont pour objectifs d’assurer la rentabilité de la banque à moyen et long terme. Elle gère la collecte et la modélisation de l’ensemble des opérations des métiers de la Banque nécessaires au pilotage des risques par l’ALM Trésorerie.

Dans le cadre de ma fonction, je travaille en tant qu’analyste développeur sur des outils de calcul de taux, de simulation, de reporting ou de supervision. L’amélioration de ses outils est un enjeu stratégique pour la banque, permettant de remplacer et d’optimiser les outils existants par des solutions performantes adaptées à nos besoins.

## La mission

Dans le cadre de mon stage, j’intègre le domaine de la Maîtrise d’œuvre (MOE) de l’ALM Trésorerie et participe à toutes les phases de développement d’un projet informatique : l’analyse, le développement, les tests et support aux utilisateurs.

Les principales activités durant mon stage sont :

* Comprendre l’environnement bancaire et les Marchés Financiers
* Clarifier les besoins en relation avec la Maîtrise d’ouvrage (le métier)
* Rédiger les spécifications techniques et fonctionnelles
* Planifier les tâches et suivre leurs avancements.
* Développer en Java, C++, et en PL/SQL.
* Rédiger les cas de tests et réaliser des recettes
* Définir la stratégie de déploiement en production
* Valider l’adéquation entre la solution déployée et le besoin Métier final.

# Présentation du sujet

Durant mon stage, j’ai intégré l’équipe Carthage, qui s’occupe de la maintenance évolutive de l’application Carthage.

Carthage est une application référentielle du groupe BNP Paribas, gérant les taux du périmètre métier de l’ALM Trésorerie, il s’agit de taux de marchés, de taux de cession interne, et des composantes de leur calcul (coûts de liquidités, coefficient de péréquation, etc…).

Carthage a pour but de gérer les devises, les cours de changes, les taux et les valeurs de ces taux sur un périmètre spécifique.

Dans le cadre de mon stage, j’ai travaillé sur un module de l’application Carthage appelé **Carthage-Ramp**. Ce module sert d’interface entre le référentiel groupe Carthage et un système d’information des salles de marchés basé à Londres.

J’ai été emmené à travailler sur différents sujets :

* La migration de l’application **Carthage-Ramp\*** en Java
* Analyste et implémentation des besoins d’évolution de la maitrise d’ouvrage
* Recette applicative
* Tâches de support applicatif aux utilisateurs de l’application Carthage.

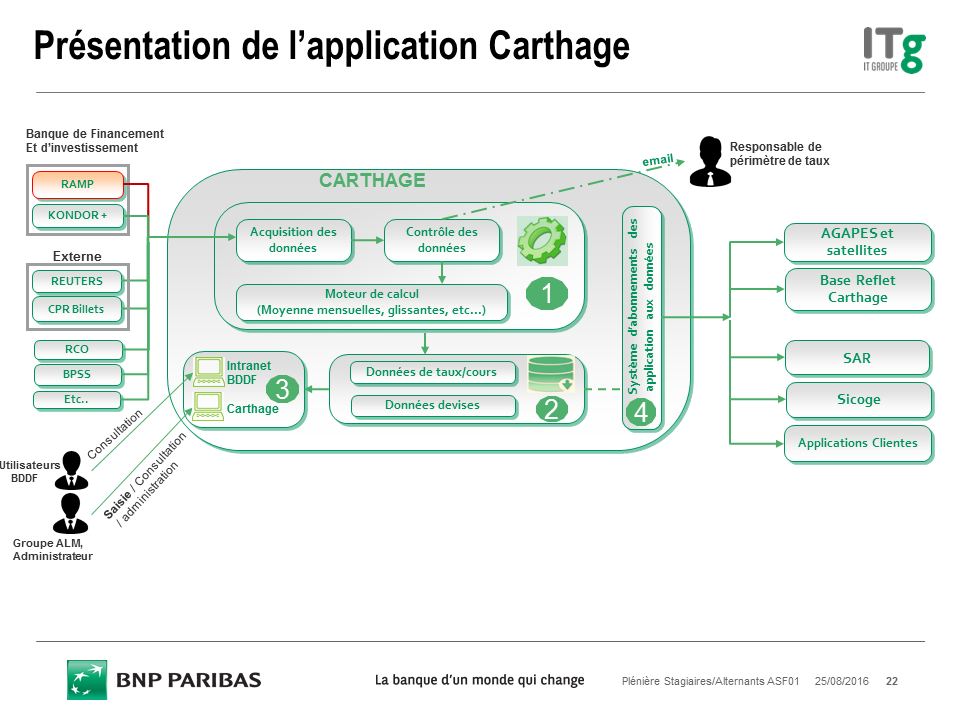
## Contexte applicatif (Application Carthage)

**Carthage** est une base de données gérant les taux du périmètre de l’ALM Trésorerie.

Il s’agit des taux de marchés, de Taux de Cession Interne (TCI) et des composantes de leurs calculs (taux de marché, coûts de liquidités, coefficient de péréquation, etc...).

Cette application a pour objectif de :

* Centraliser tous les besoins de l’ALM Trésorerie en terme de devise, taux et cours (taux de marché, TCI, coûts de liquidité, etc…)
* Garantir ces données en contrôlant leur cohérence et leur validité.
* Diffuser les nomenclatures des devises à l’ensemble du Groupe.
* Diffuser les taux de l’ALM Trésorerie à l’ensemble du groupe par application intranet, grâce à un système de juridiction de portée Groupe.
* D’alimenter les bases reflets des autres filiales (Banque d’investissement, Banque de détail)



Carthage permet de récupérer les données relatives aux taux, cours de change et devises depuis plusieurs sources de données externes et internes (RAMP, Reuters, etc…), afin de les mettre à disposition de l’ensemble des applications de l’ALM Trésorerie.

Ces données sont récupérées par des flux quotidiens, mensuels ou par vacation.

Une fois les données reçus, l’application Carthage contrôle ses valeurs de taux afin de détecter des écarts. Lorsqu’un écart est constaté entre deux valeurs reçues, un email est automatiquement envoyé au responsable du périmètre du taux, qui s’occupe de faire remonter l’information à l’ensemble du Groupe et comprendre l’origine de ces écarts.

Après le contrôle des données, Carthage s’occupe de les traiter, grâce à sa bibliothèque de fonctions internes qui lui sert de moteur de calcul, permettant de calculer des moyens mensuelles, glissantes etc… sur les données importées.

Une fois la phase de traitement, ces données sont ensuite historicisées dans une base de données.  
Ces données sont mises à la disposition du groupe par deux intranets : un intranet pour les banques de détail de France (BDDF), et un autre pour l’ensemble des utilisateurs de l’ALM Trésorerie.

Ces données historicisées sont également mises à la disposition de l’ensemble des applications de la BNP Paribas, grâce à un système d’abonnement.

L’abonnement aux données de taux/cours de Carthage, se fait par flux CFT, et en spécifiant le niveau\* de taux/cours à envoyer à l’application cliente, et le nom du fichier qui contient la liste des taux ou cours à envoyer par CFT.

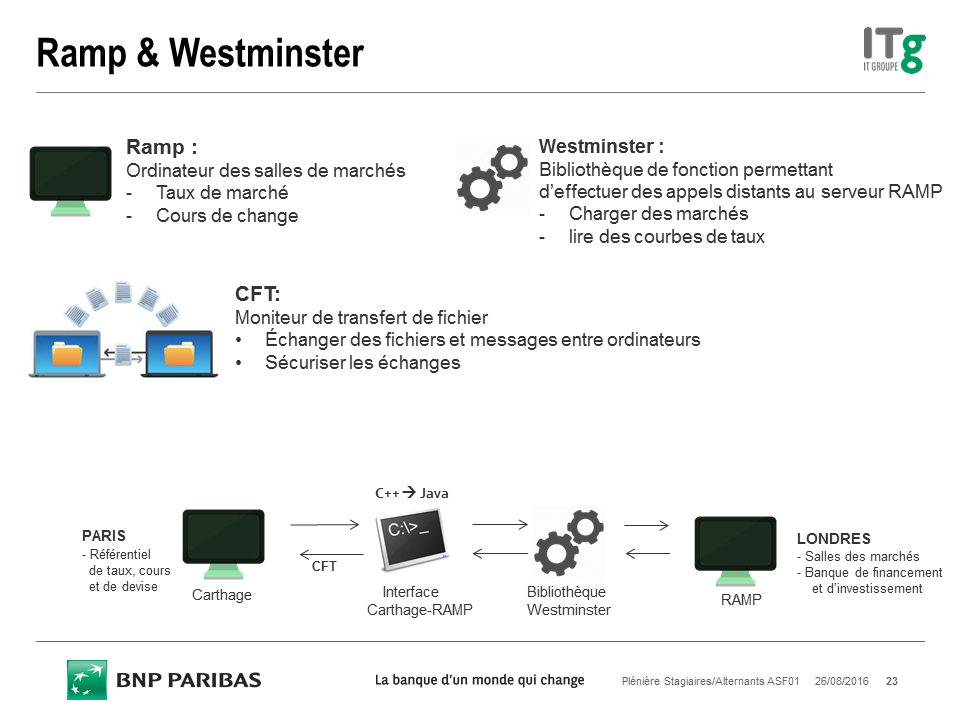
Niveau\* : chaque référence de taux/cours de change se trouve sous une arborescence (niveau) dans l’application Carthage

Dans le cadre de mon stage, j’interviens sur le module Carthage-Ramp qui interroge le l’application RAMP, afin d’importer dans Carthage les cours de change et les taux swaps.

## Module Carthage-Ramp

Carthage-Ramp est une application qui sert d’interface entre le référentiel groupe Carthage et un système d’information des salles de marchés RAMP basé à Londres.

Cette application utilise la librairie **Westminster** pour se connecter au système d’informations RAMP afin de charger des marchés financiers, récupérer des taux de change, et lire les courbes de taux de marchés.

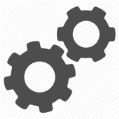


## 

## Système d’information RAMP

**Ramp** un système d’information des salles de marchés basé à Londres.  
C’est une application qui fournit des méthodes d’accès uniforme aux taux et données de courbe de taux, pour différents groupes de taux d’intérêts des systèmes de négociation et de systèmes de valorisation (pricing).

## Librairie Westminster

Westminster est une librairie de BNP Paribas qui contient des fonctions de valorisation d’un actif et de gestion des risques pour les produits dérivés de taux et des obligations.

## Taux de cession interne

 Le Taux de Cession Interne (TCI), c’est le taux auquel l’ALM Trésorerie met

à la disposition des filiales de BNP Paribas, pour pouvoir se refinancer.

## Flux CFT

CFT signifie Cross File Transfert, c’est un moniteur de transfert de fichier de données bancaires. Il assure l’échange de données entre plusieurs types de systèmes d’exploitation, et supporte un grand nombre de type de réseau, et sécurise les échanges en certifiant les partenaires de cet échange.

## Les périmètres de taux de l’application Carthage

L’application Carthage s’occupe de deux principaux groupes de taux : Les cotations Forex (cours de change), et les taux de marchés.

Les cours de changes contiennent :

* **Spot exchange rates :** Les cours de change d’une devise contrevalorisée en Euro ou en dollar.
* **Valinfo exchange rates :** Cours officiel de change pour la comptabilité générale.
* **Fixing BCE :** Fixings de diverses devises contre euro par la BCE.
* **CPR Exchange Billets :** Cours officiels issus de CPR pour achat et vente de billets.

Les taux de marchés contiennent :

* **Futures :** Taux d’un contrat à terme qui est un contrat standardisé négocié sur un marché organisé, permettant de s’assurer ou de s’engager sur un prix pour une quantité déterminée d’un sous-jacent à une date future.
* **Taux nationaux :** taux état en devises, taux état en inflation, taux moyen des emprunts d’état (TME), taux moyen obligataire (TMO).
* **Taux forward :** taux euribor 12M démarrant dans 1 an ou 2 ans
* **Overnight rates :** Taux de marché au jour et ces moyennes associées (T4M, TAG)
* **Taux IBOR :** Fixing quotidien des taux interbancaires sur différents places de cotation.
* **Spread swap / government bond bid rate:** Spread entre le taux interbancaire et le taux état.
* **Taux swap :** Contrat d’échange entre un taux fixe et un taux variable(IRS, CMS, OIS)

## Description des instruments financiers utilisés

### Spot exchange rates (Cours de change Spot Forex)

Le change comptant ou change « spot » consiste à échange 2 devises (gré-à-gré), à un cours négocié, 2 jours ouvrés après la date de négociation. Cette date est appelée « Date spot ».

Les caractéristiques principales d’une opération de change comptant sont :

* La devise principale
* Le sens : achat ou vente
* La devise secondaire, ou « prix » : devise vendue s’il s’agit d’un achat, devise achetée s’il s’agit d’une vente.
* Date de négociation
* La date de valeur ou "spot date": généralement égale à la date de négociation + 2 jours ouvrés. Remarque : un contrat de change au comptant a une durée de vie nulle ; il n'y a pas de « date de fin ».
* Le montant négocié, exprimé dans la devise principale
* Le cours négocié
* Le montant dans la devise secondaire, calculé à partir du montant principal et du cours

### Taux Swap

Contrat gré à gré qui permet d’échanger sur la base d’un montant nominal, des flux d’intérêts calculées sur un taux fixe contre les flux d’intérêts calculés sur un taux variable.

Le montant nominal sert de base au calcul des intérêts. Ce montant est notionnel, mais pas échangé.

Ce type d'opération permet de se couvrir contre le risque de taux.

### Swap de base

Le basis swap (Swap de base) est un type de swap dans lequel deux contreparties échangent des taux d'intérêt variables en fonction des différents marchés monétaires.

Cela se fait habituellement pour limiter le risque de taux d'intérêt que l'entreprise est confrontée à la suite d'un prêt et d’emprunt de différents taux.

Il existe donc 3 types de swap de base :

* Swap de taux d’intérêts
* Swap de devise

Si le swap de base porte sur **une** devise, alors c’est un **swap de taux d’intérêts**

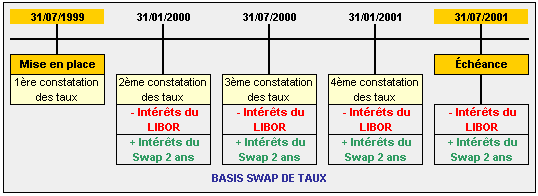
S’il porte sur **deux** devises, alors c’est un **swap de devises.**

#### Swap de taux d’intérêts

* Pas d'échange de capital à la mise en place.
* Flux référencés dans la même devise et payés sur la fréquence d'un des taux.
* Pas d'échange de capital à l'échéance.

**Exemple de swap de taux d’intérêt :**

* Nous sommes receveur du taux de swap 2 ans et payeur du LIBOR 6 mois.
* Période : du 31/07/1999 au 31/07/2001.

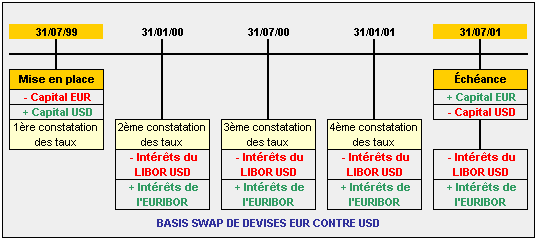


#### Swap de devises

* Échange de capital à la mise en place.
* Flux d'intérêts référencés dans 2 devises différentes et réglés dans leur devise respective.
* Intérêts calculés sur le nominal de la devise correspondante.
* Échange de capital à l'échéance.

**Exemple de swap de base de devises :**

* Nous sommes receveur de l'EURIBOR 6 mois et payeur du LIBOR USD 6 mois.
* Période : du 31/07/1999 au 31/07/2001.



# Présentation du travail réalisé

Pendant mon stage, j’ai pu travailler sur plusieurs sujets.

J’ai travaillé dans un premier temps sur la migration de l’application **CarRamp**.

Ensuite nous avons reçu des demandes d’évolution provenant de la maîtrise d’ouvrage sur la nouvelle application, nous avons analysé ses demandes, rédiger les documentations fonctionnelles associées à ses demandes, et implémenté les solutions.

En parallèle de mon sujet principal de stage, j’ai été emmené à monter en compétence sur les applications interne et faire des tâches de support applicatif pour assister les utilisateurs de l’application Carthage, qui est une application critique qui dispose des données, dont plusieurs applications dépendent.

## Migration application Carthage-Ramp (CarRamp)

Pour réaliser la migration de l’application CarRamp, nous avons rédigé un document de conception afin de définir le choix de conception et des traitements pour la récupération des taux/cours de change de RAMP dans le patrimoine Carthage à l’aide des librairies Westminster.

Pour alimenter la base de données de l’application Carthage, plusieurs requêtes sont effectuées pour récupérer des taux/cours de change depuis RAMP.

Pour dissocier ces requêtes, la notion de Destination a été instaurée pour identifier chaque requête effectuée vers RAMP.

Une destination caractérise donc un ensemble de groupe de taux/cours de change.

Exemple :

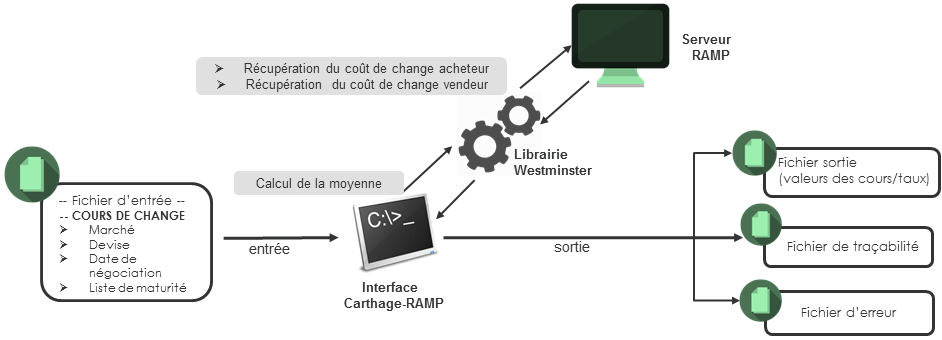
* Destination81 : Cours de change spot Forex
* Destination74 : Swap de taux
* Destination100 : Swap de devises
* Etc…

### Description de l’architecture logicielle

Pour chaque ensemble de taux/cours de change à récupérer, l’application Carthage-RAMP (CarRamp) prend en paramètre le fichier de configuration de cet ensemble, puis interroge le serveur RAMP par l’intermédiaire de la librairie Westminster, pour les valeurs souhaitées.

Voici donc un exemple de la **destination81** **: Récupération des cours de change**.

Cet exemple prend en entrée un fichier de configuration spécifique, interroge le système d’information RAMP grâce à la librairie Westminster, puis génère 3 fichiers en sortie.



Pour récupérer les groupes de taux/cours de change de Ramp, l’application Carthage-Ramp lit en entrée un fichier de configuration qui contient la liste de toutes les données nécessaires pour effectuer des appels sur le système d’information RAMP.

Pour récupérer donc les cours de change (Destination 81), l’application doit connaitre marché à charger, les devises à récupérer, la date de négociation du contrat d’échange, et les différentes maturités des échanges.

L’application prend en entrée ses données, et utilise la libraire Westminster pour récupérer le coût de change acheteur (ASK) et les cours de change vendeur (BID), puis calcule la moyenne des 2 données récupérées.

L’application génère 3 fichiers en sortie.

* Un **fichier de résultats**: qui contient les résultats calculés (ensemble de données)
* Un **fichier de traçabilité**: qui permet de retracer l’application le début de son exécution jusqu’à la fin de son exécution.
* Un **fichier d’erreur** : contenant tous les erreurs rencontrées pendant l’exécution de l’application

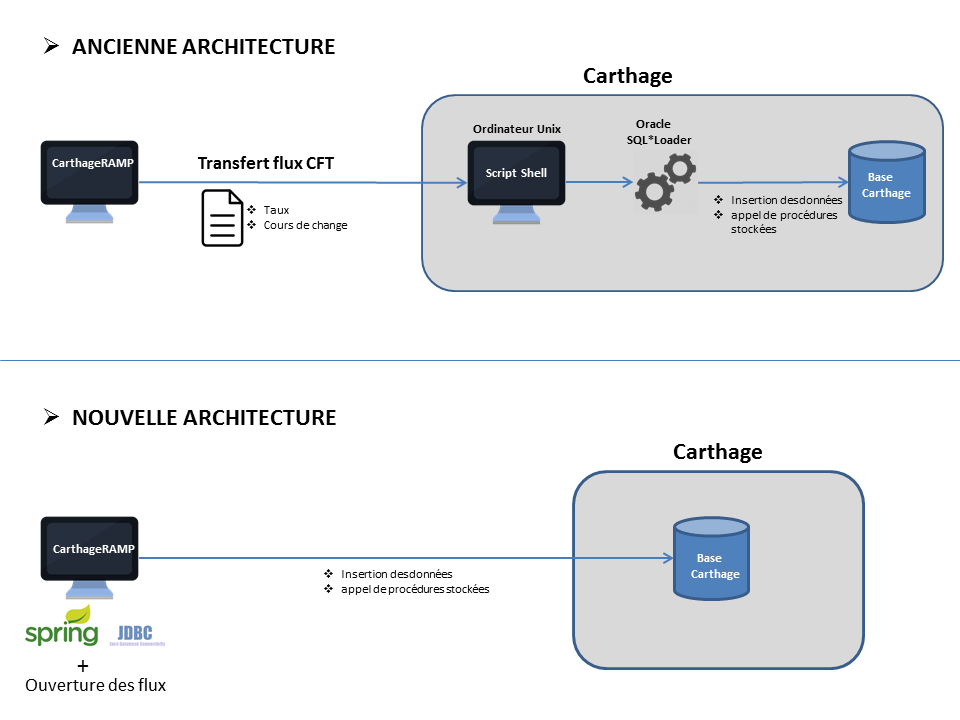
Parmi ces 3 fichiers générés, un seul fichier est envoyé au périmètre Carthage, pour être traité et insérer dans la base de données, c’est le fichier de résultats.

Pendant la migration, nous avons amélioré l’architecture existante de l’application Carthage-Ramp

#### Fonctionnement de l’ancienne architecture de l’application Carthage-RAMP

Dans l’ancienne architecture de Carthage-Ramp, le fichier de résultat généré par l’application Carthage-Ramp était envoyé par flux CFT à Carthage.

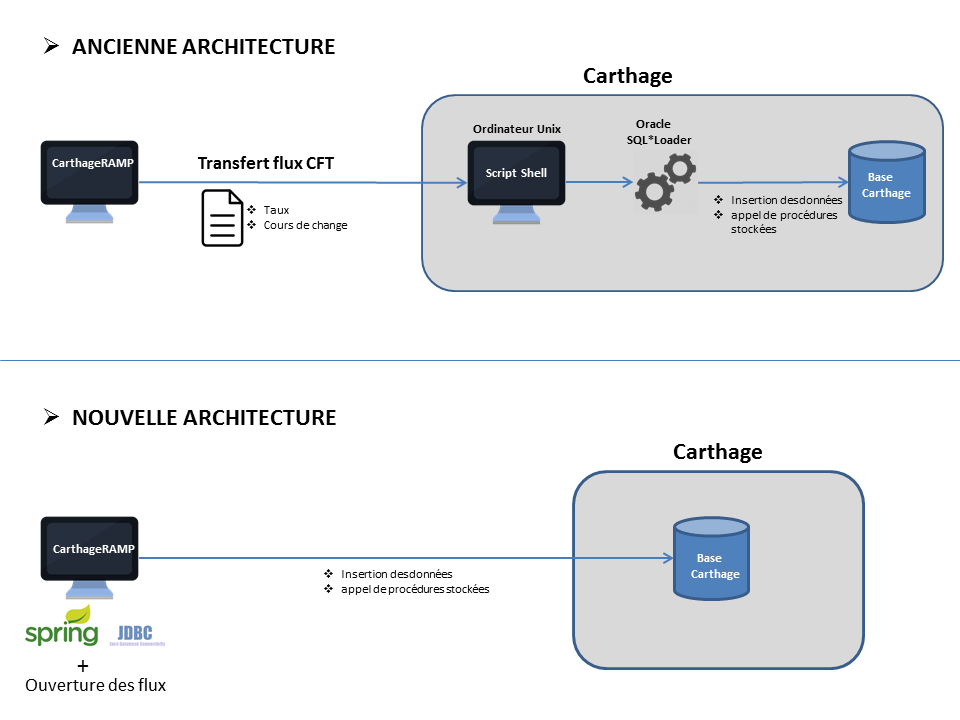
A l’arrivé du fichier du fichier dans Carthage, un script shell **.ksh** est lancé, ce script retrace l’exécution de l’application, utilise SQL\*Loader pour inserer les données en base de données et lance une procédure stockée pour les joindre dans la table principale des taux.



#### Fonctionnement de la nouvelle architecture de l’application CarthageRamp

Dans la nouvelle architecture de Carthage-Ramp, nous avons intégrer le Framework Spring Jdbc et ouvert les flux entre les ordinateurs hébergeant l’application Carthage-RAMP et la base Carthage.

Dans cette nouvelle architecture, nous n’écrivons plus les résultats dans un fichier de sortie, et avons supprimer le transfert de fichier par flux CFT.  
On a également supprimer lancement de programme shell, ce qui implique qu’on n’a plus besoin de SQL\*Loader pour charger le fichier de résultats en base.



Les données de taux ou cours de change sont directement insérées en base de Carthage, l’application Carthage-RAMP se charge elle-même de la traçabilité des insertions en base, et lance une procédure stocké pour les joindre dans la table principale des taux, grâce au Framework Spring

### Les couches logicielles

L’application Carthage-Ramp a 2 couches logicielles.

|  |  |
| --- | --- |
| **Couche logicielle** | **Responsabilités** |
| Westminster | Permet de faire des appels distants RAMP pour le calcul de taux et courbe de taux |
| Spring Jdbc | Librairie open source, permettant d’insérer les données dans la base de données Carthage, et lancer les procédures stockées. |

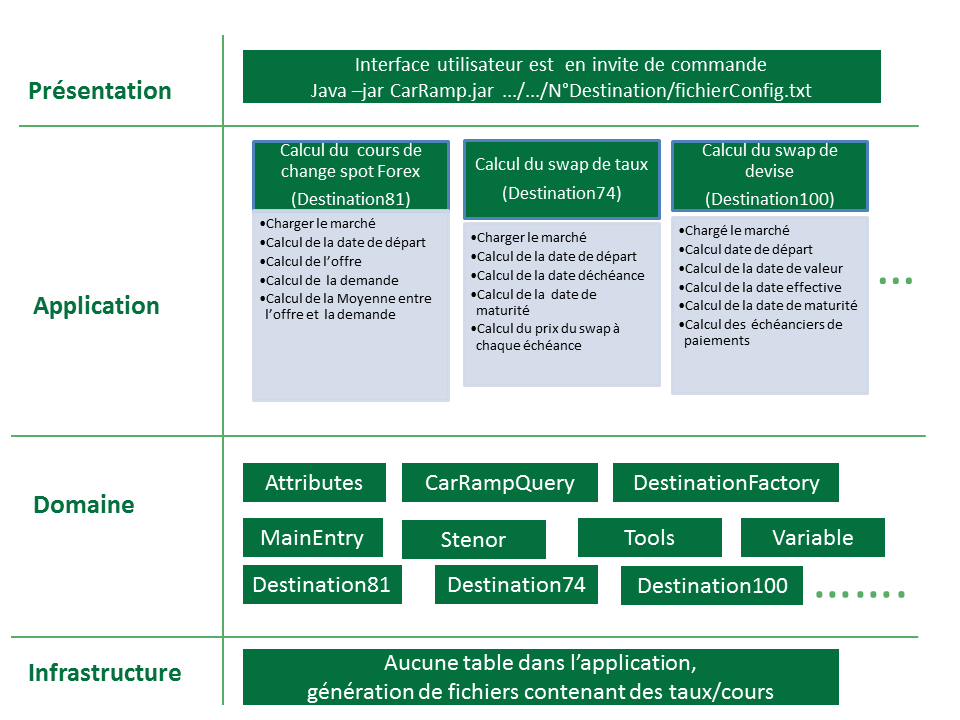
### Couches applicatives

En utilisant le concept de Domain Driven Design (DDD), qui consiste à séparer les couches applicatives de notre application.

La couche **Présentation** permet donc l’exécution de l’application par invite de commande et en faisant appel à une JRE Java.

La couche **Application** décrit les différentes opérations effectuées pour récupérer les ensembles de valeurs de taux / cours de change. Sachant que chaque ensemble de taux/cours récupérés possède ses propres opérations.

La couche **Domaine** décrit contient toutes les classes métiers de l’application.



### Analyse et Conception de l’application

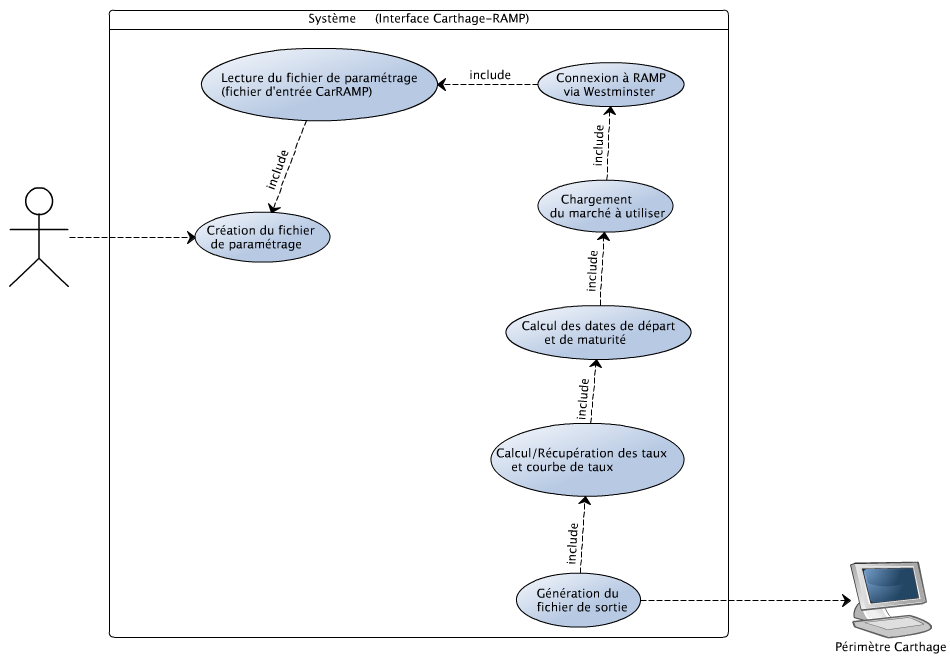
### Diagramme de cas d’utilisation

L’interface Carthage-RAMP s’exécute en arrière-plan.

Pour s’exécuter, le programme devra prendre en entrée un fichier de paramétrage contenant le numéro de la destination, le traitement associé dans l’application Carthage, la date de traitement, etc…

Une fois, ce fichier fournit, l’application Carthage-RAMP (CarRAMP) va lire l’entête de ce fichier, qui permettra grâce à la librairie Westminster de se connecter à RAMP, pour charger les marchés à utiliser, calculer les dates de départ et de maturité des taux ou courbe de taux, et ensuite générer un fichier de sortie qui contiendra les valeurs récupérées ou calculées depuis RAMP.

Ce fichier de sortie sera ensuite envoyé à l’application Carthage afin de lancement le traitement associé et insérer les taux/cours en base de données.

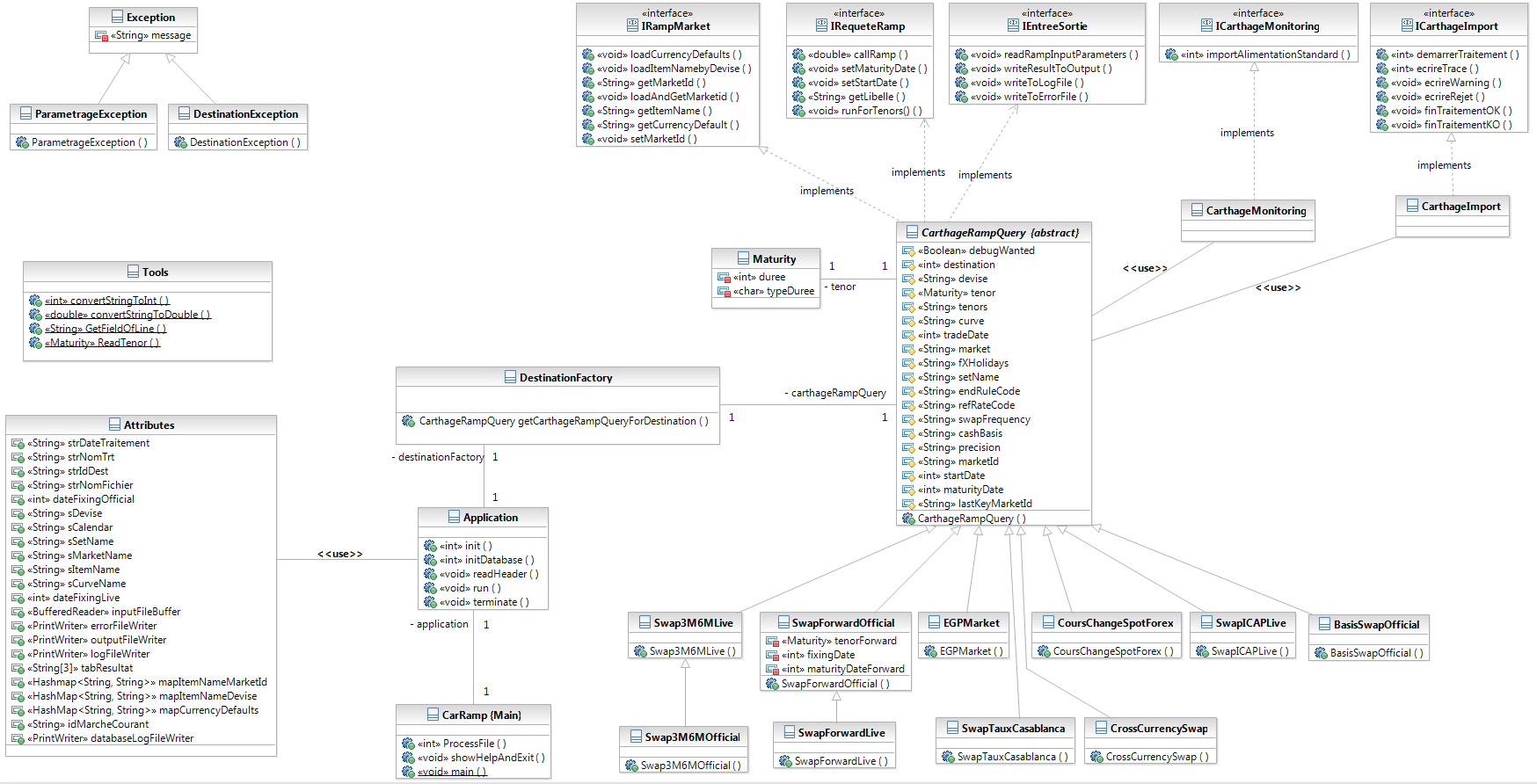


#### Patterns de conception

Le pattern de conception est utilisé pour la conception de l’application est Factory, qui nous permet d’instancier une destination (récupérer un ensemble de taux/cours), en fonction du contexte.

|  |  |
| --- | --- |
| **Nom** | **Description** |
| Factory | Création d’un objet dont le type dépend du contexte (la destination) |

### 4.1.4.3. Diagramme de classes



#### Description des classes

Ce diagramme présente les différentes relations entre les classes.

Il permet décrire les attributs des classes et les différentes fonctions pouvant être appelé dans chacune de ses classes.

* **Classe Maturity :** permet de représenter une date de maturité.
* **Classe Attributes :** permet de représenter les données du fichier de paramétrage, du programme, des fichiers de logs, de d’erreur, et de sortie.
* **Classe Tools :** contient les fonctions utiles pour l’ensemble du programme
* **Classe CarthageRampQuery :** Cette classe contient tous les attributs nécessaires aux groupes de taux à calculer par l’interface Carthage-RAMP.

C’est une classe abstrait (classe mère) dont hérite toutes les classes destinations, elle contient les fonctions abstraites qui doivent être redéfinit dans les classes qui hérite d’elle, et comporte également des fonctions qui peuvent être réutilisée dans l’ensemble du programme.

* **Classe Application :** permet de lire le fichier l’entête du fichier d’entrée, de lancer la destination concernée. Elle contient les fonctions permettant d’initialiser le fichier de sortie, d’erreur ou de log.
* **Classe CarRamp :** elle permet de lancer le programme principal avec en entrée le fichier de paramétrage, et exécute la requête pour l’ensemble du groupe de taux à récupérer.
* **Classe Swap3M6MLive :** Cette classe permet de calculer les taux Swap 3 mois contre 6 mois, (Destination87), elle hérite de la classe *CarthageRampQuery*.
* **Classe SwapForwardOfficial :** Cette classe permet de calculer les taux Swap forward pour une date donnée (Destination94), elle hérite de la classe *CarthageRampQuery*.
* **Classe EGPMarket :** Cette classe permet de calculer les taux de marché, elle hérite de la classe *CarthageRampQuery*.
* **Classe CoursChangeSpotForex :** Cette classe permet de récupérer les cours de change comptant (Destination81), elle hérite de la classe *CarthageRampQuery*
* **etc ….. :** (Toutes les autres destinations décrites dans la partie 4)

#### Description des attributs des classes

Gestion d'une interrogation de Ramp à partir d'un fichier de configuration propre à une destination Carthage. Le séparateur utilisé dans le fichier de configuration est la tabulation.

#### Attributs de la classe CarthageRampQuery

Les paramètres attendus dans le fichier de configuration sont les suivants :

* **Devise** : la devise de la ligne en cours de traitement du fichier de paramétrage
* **tenor** : représente une seule maturité
* **tenors** : liste de maturités séparées par des virgules
* **Curve** : code de la courbe à lire (Vide par défaut)
* **TradeDate** : la date d'interrogation (date du jour par défaut)
* **Market** : le marché à utiliser
* **FXHolidays** : la place utilisée pour les jours fériés
* **SetName** : définit si c’est une récupération de marché en temps réel, ou sur une date donné
* **EndRuleCode** : règle de calculs des dates
* **RefRateCode** : code du taux
* **SwapFrequency:** la fréquence du swap
* **CashBasis**: cashBasis utilisé dans les fonctions Westminster
* **Precision :** la précision à utiliser pour écrire le résultat // Défaut : 10
* **Destination :** identifiant de la destination Carthage en cours de traitement

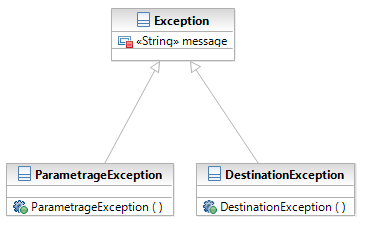
#### Attributs de la classe Attributes

Les attributs de cette classe permettent de sauvegarder en mémoire l’entête du fichier de paramétrage, la référence vers les fichiers d’entrée, de sortie, d’erreur et de traçabilité.

#### Les exceptions

Les exceptions de l’application sont gérées par 2 classes :

* **Classe ParamétrageException :** permet de spécifier les erreurs liées au fichier de paramétrage
* **Classe DestinationException :** permet de spécifier les erreurs liées aux différentes destinations (Exemple : destination introuvable)



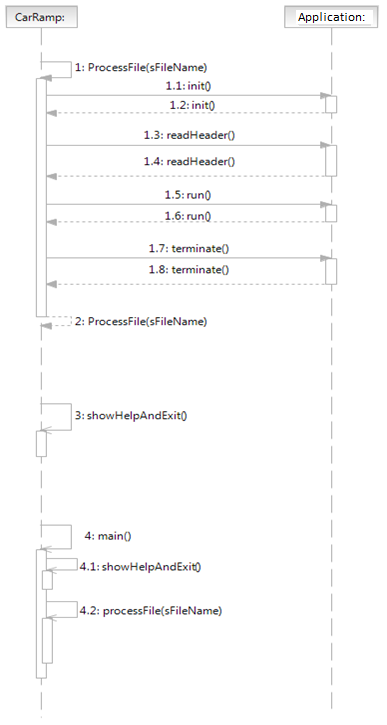
### 4.1.4.4. Diagramme de séquence

Le diagramme de séquence permet représenter les différents messages échangés entre les objets.

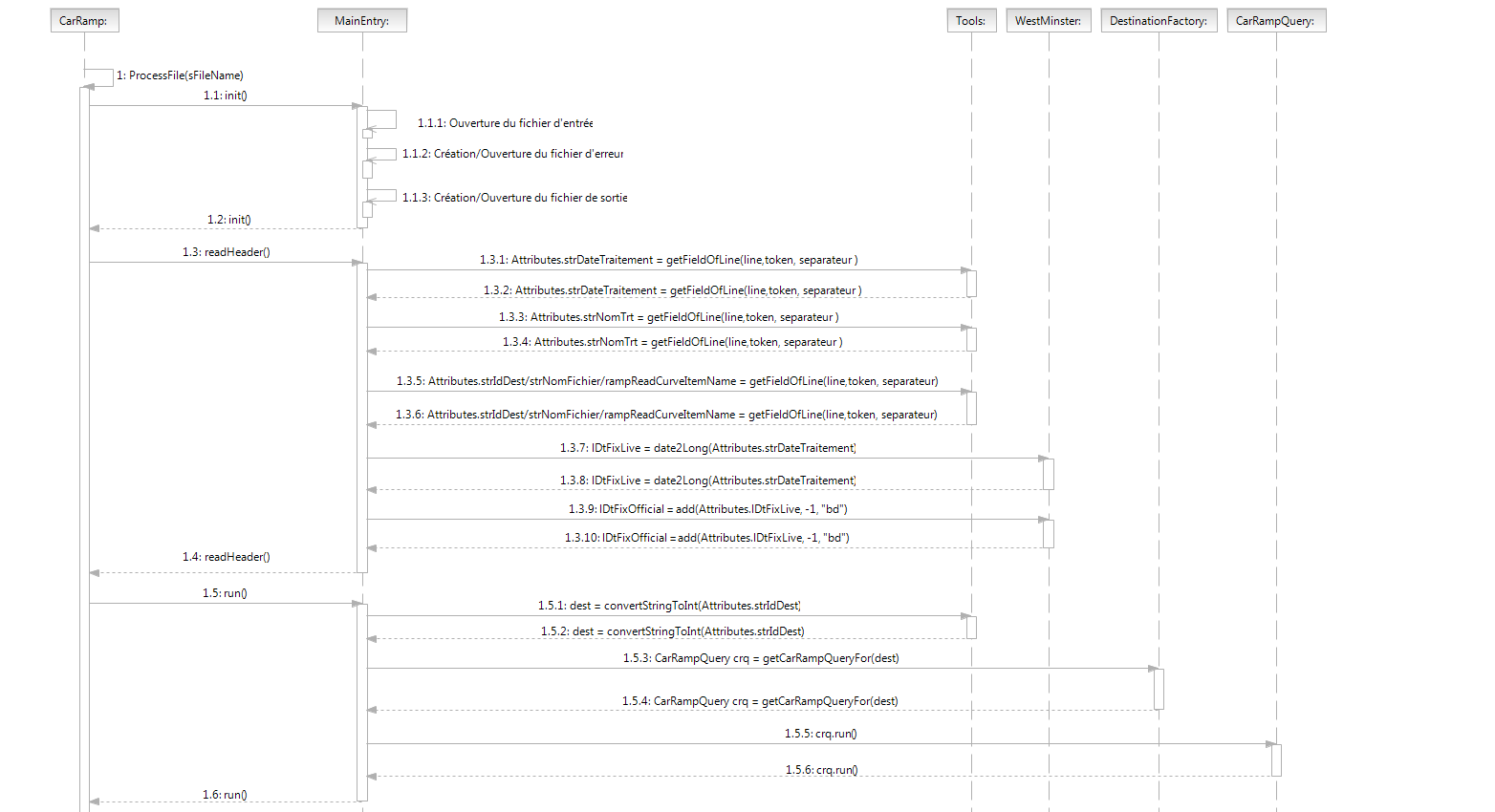
#### Classe CarRamp

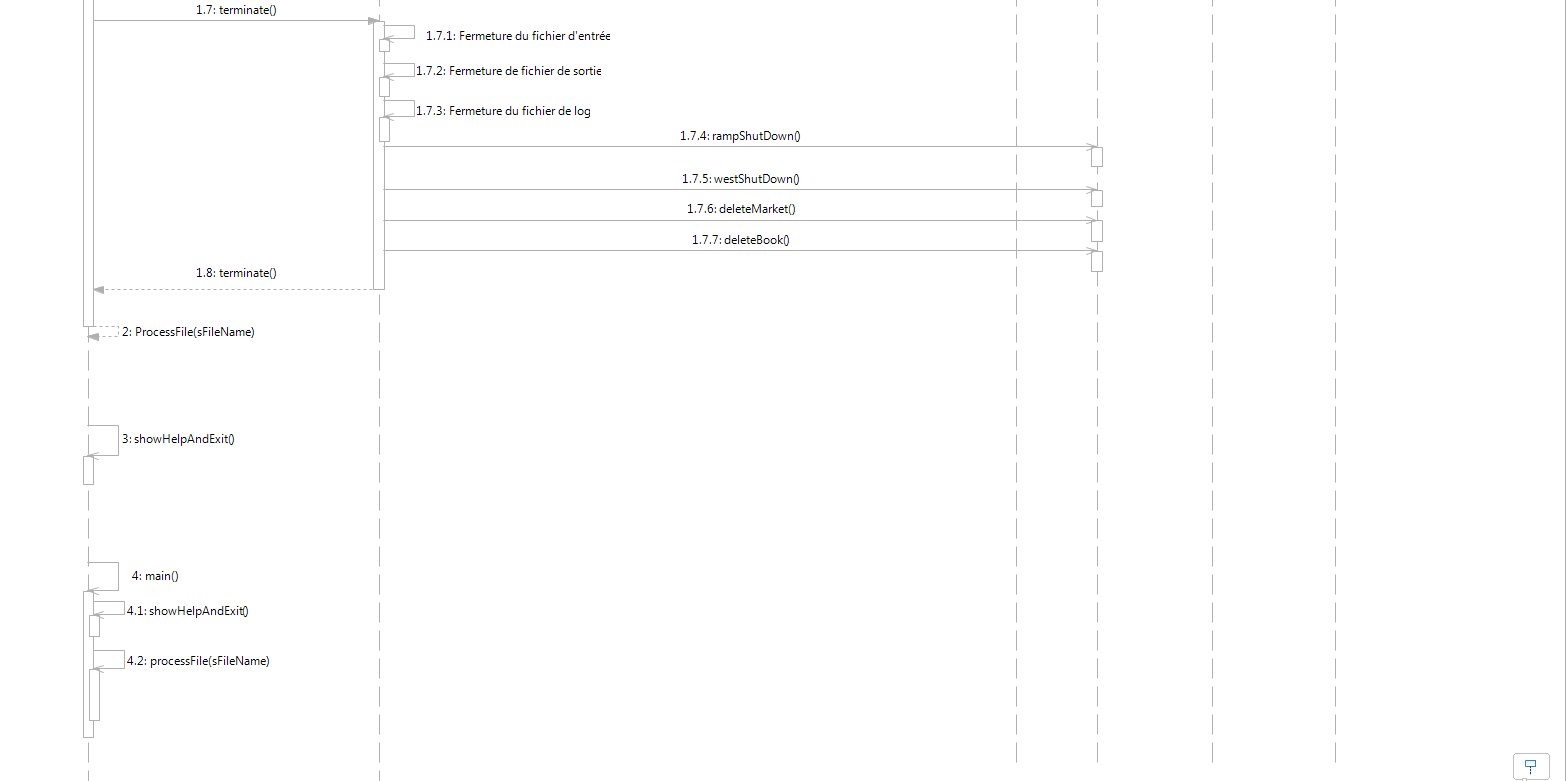
Ce diagramme décrit les appels des fonctions synchrones et asynchrones de la classe CarRamp.

**Exemple** : la fonction ***ProcessFile*** est une fonction asynchrone (fonction void) qui fait appel aux méthodes ***init()***, ***readHeader()***, ***run()*** et ***terminate()*** de la classe ***MainEntry***



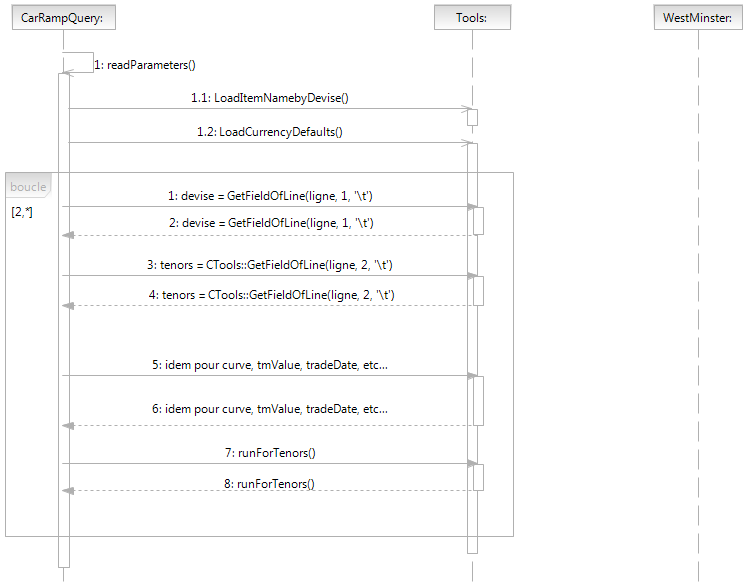
Dans cette partie, nous détaillons en profondeur les différentes fonctions appelées par les méthodes ***init()***, ***readHeader()***, ***run()*** et ***terminate()*** de la classe ***MainEntry.***



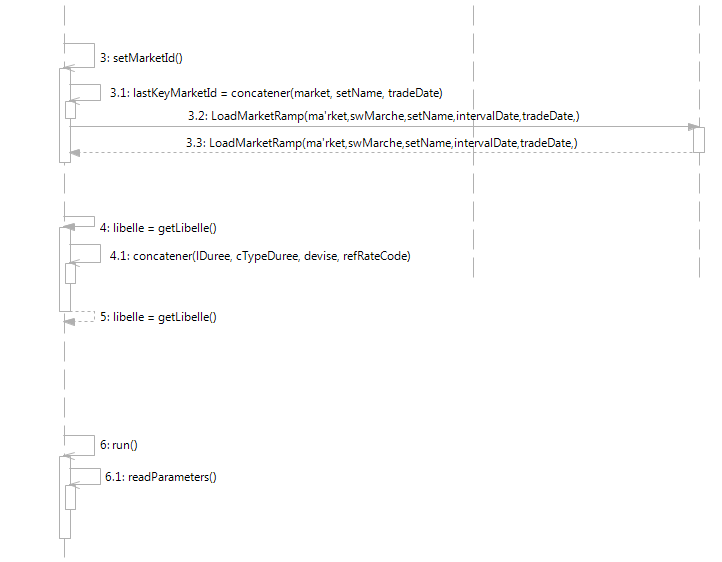


#### Classe CarRampQuery

Ce diagramme décrit les fonctions permettant de charger les marchés, les devises. Il lit les données du fichier d’entrée (Fichier de paramétrage). Pour chaque ténor lu, il calcule la date de maturité, et la date de départ, et récupère le taux/cours, et écrit dans un fichier.

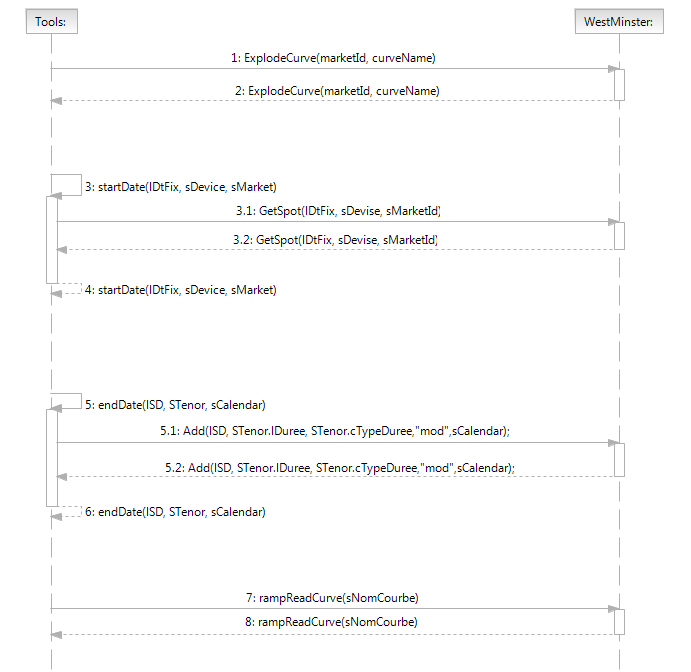


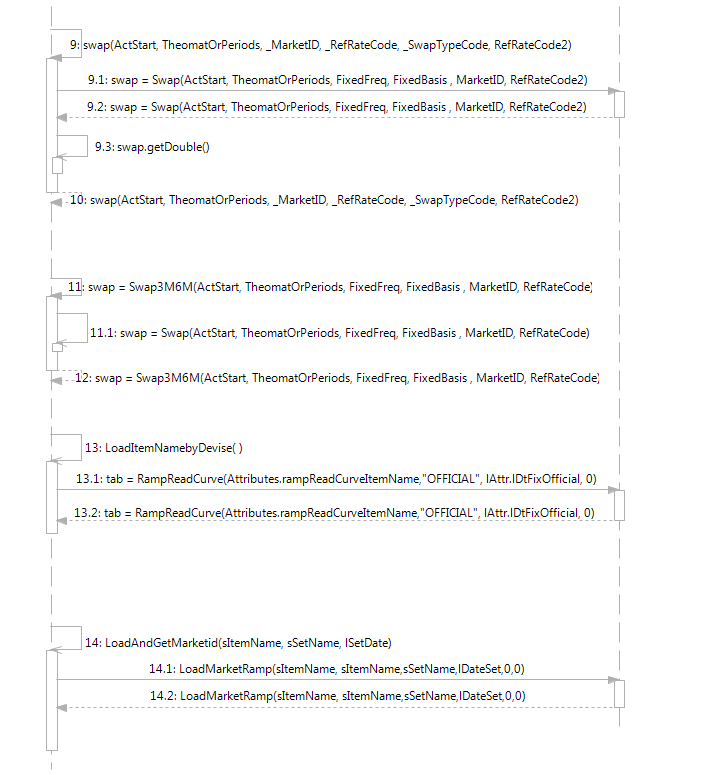




#### Classe Tools

Ce diagramme décrit les fonctions utiles pour l’ensemble du programme





### 3.1.4.5. Tests unitaires

Pendant les phases de développement de l’application, nous avons fait progressivement des tests unitaires, pour tester fonction par fonction les différentes classes implémentées et de tester les différentes fonctions externes de la librairie Westminster, appelés lors de l’exécution de l’application.

La librairie utilisée pour effectuer ses tests est **JUnit.**

**C:\StageBNP\Tmp\JUnit.pngJUnit** est une librairie open source pour le développement et l'exécution de tests unitaires automatisables. Le principal intérêt est de s'assurer que le code répond toujours aux besoins même après d'éventuelles modifications. Plus généralement, ce type de tests est appelé tests unitaires de non-régression.

### Répartition du temps de travail

La phase analyse du projet représente 40% de la réalisation du projet, car elle représente la partie importante de l’application. Elle comprend l’analyse de l’existent, la reprise et la compréhension de code.

La phase de conception représente 15% du projet, elle comprend la conception, la rédaction de spécifications fonctionnelles.

La phase de développement représente 30 %, et les tests unitaires 15%.

## Gestion des erreurs

### Typologie des erreurs

Les différentes d’erreurs rencontrées sont des erreurs liées au paramétrage d’un fichier d’entrée, ou des erreurs liées à la destination, ou erreurs liées aux appels RAMP.

|  |  |
| --- | --- |
| **Type** | **Description** |
| Fichier de paramétrage  (tabulation, etc…) | Aucun incident sur le système, mais les erreurs sont logger dans le fichier d’erreur de la destination (fichier **.err**) |
| Destination introuvable | Aucun incident sur le système, mais les erreurs sont logger dans le fichier d’erreur de la destination (fichier **.err**) |
| Appel RAMP | Aucun incident sur le système, mais les erreurs sont logger dans le fichier d’erreur de la destination (fichier **.err**) |
| Connexion à la base de données | Aucun incident sur le système, mais les erreurs sont logger dans le fichier de log de la base de données de la destination (fichier **database.log**) |

### Messages d’erreurs

Les messages d’erreurs sont affichés en fonction de la destination lancée par l’application CarRamp.

Chaque fonction de l’application permet d’écrire un message d’erreur dans le fichier d’erreur de la destination lancée (fichier **.err** de la destination).

L’application permet donc logger toutes les erreurs rencontrées à tous les niveaux lors de l’exécution d’une destination.

### Gestion de la traçabilité

La traçabilité de l’application est assurée par le fichier de log, et d’erreur,

Le déroulement entier de l’application est enregistré dans le fichier de log.

Lorsqu’une destination est exécutée, l’application enregistre tous les appels de fonction, et les différentes données en entrée dans le fichier de log de la destination, cela permet donc de retracer l’application depuis le début de son exécution.

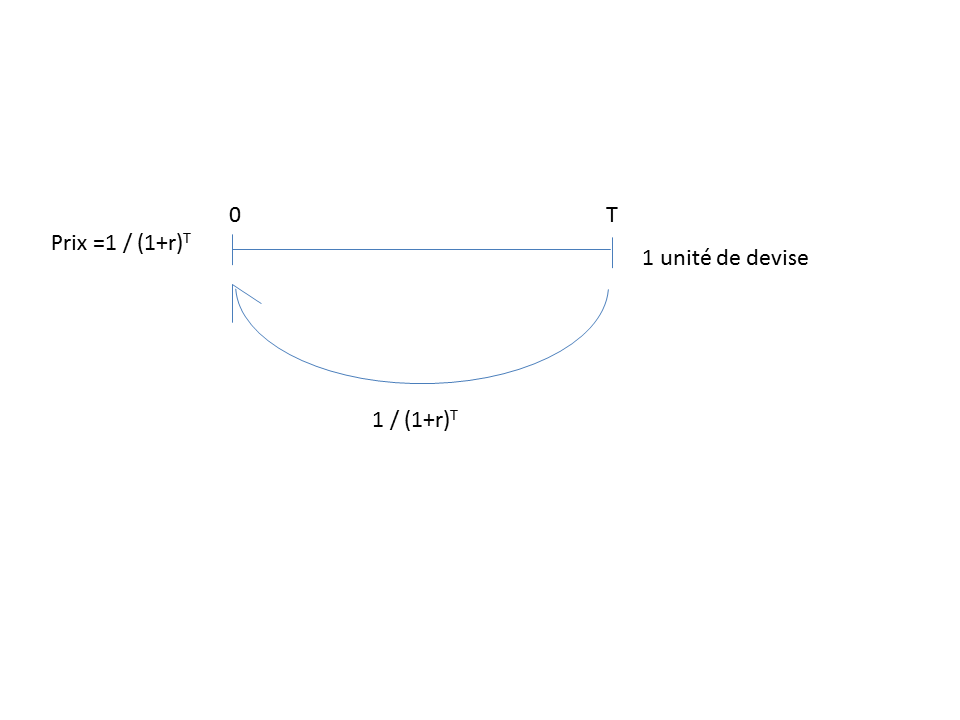
## Demandes d’évolutions

Après le développement de la nouvelle version de l’application, nous avons reçu 2 nouvelles demandes d’évolutions venant de la Maitrise d’ouvrage, pour la récupération de taux d’actualisation des obligations à long terme (Discounts Factors), et la récupération des contributions APAC (ensemble géographie constitué de de l’Extrême-Orient et de l’Océanie).

### Discounts factors

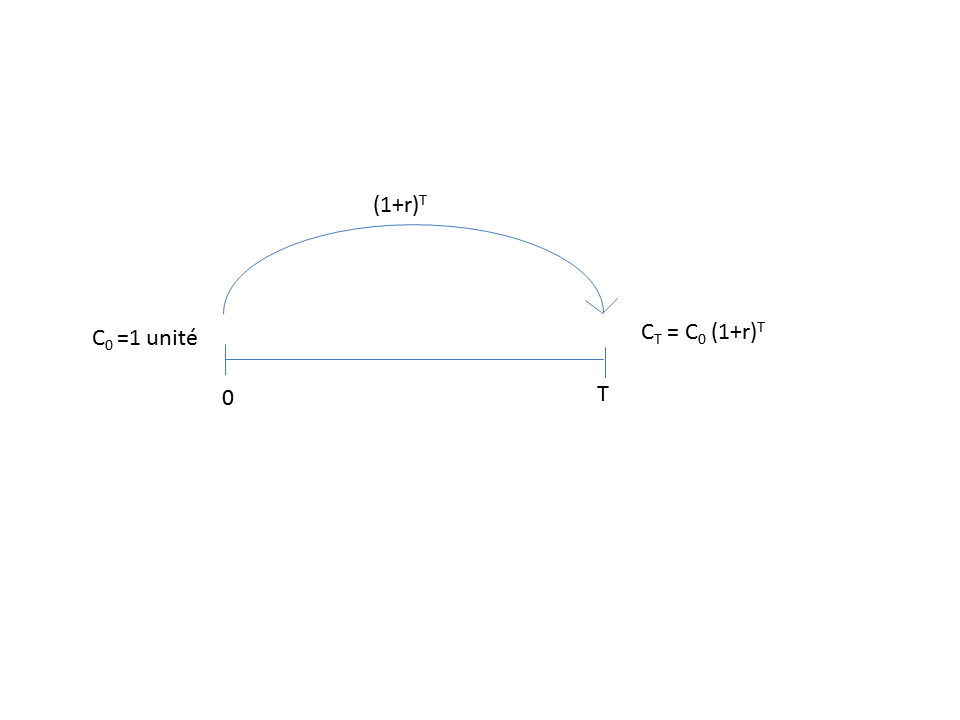
Un discount factor est un facteur d’actualisation associé à une date future, est le prix à payer pour acheter une unité de devise perçu ç cette date future. Il correspond à la quantité d’argent à placer aujourd’hui pour obtenir une unité de devise à l’horizon.

Suite à la demande de l’Autorité Bancaire Européene, il nous a été demandé de produire de nouveaux indicateurs en valeur pour mesurer l’impact de différents chocs de taux sur la valeur de l’intégralité du bilan du Groupe BNP Paribas.



Avec **r :** taux effectif

A l’inverse, on a la capitalisation, qui permet d’évaluer la somme à gagner à l’horizon si on place de l’argent aujourd’hui (t=0).



### Contributions WMI APAC

Pour le besoin de calcul des Taux de Cession Interne (TCI) pour un périmètre du groupe BNP Paribas, il nous été demandé d’intégrer les contributions d’Asie Pacific (APAC) de la trésorerie sur le court terme.

Ces contributions sont des contributions des trésoriers de WMI (Wealth Management International) du groupe BNP Paribas.

APAC est un ensemble géographique constitué de l’Extrême-Orient et de l’Océanie, il intègre les données de la plupart des pays de l’Asie de l’est, l’Asie de l’ouest, de l’Asie de Sud-Est et de l’Océanie.

# Travaux intermédiaires

* **Résolution incidents/anomalies** liées programme C++, qui retournait des valeurs/cours nulles quand elles étaient en réalité sans valeur (valeur vide).
* **REUTERS EIKON :** Appel du support de REUTERS pour la recherche de code RIC, du taux LIBID des devises EUR, USD, AED, AUD, CHF.
* **Rédaction intégrale du dossier de conception :** récupération du Template du dossier de conception de BNP, et rédaction en fonction de ce fichier.
* Création de nouvelles courbes de taux pour le calcul du TCI pour les BDDF, (composante péréquation)
* **Résolution d’incident :** Résolution d’un incident lié aux valeurs de taux vide retourné par l’interface Carthage-Ramp. Quand une valeur de taux récupérée était vide (sans valeur), l’application mettait par défaut la valeur 0. Ce qui n’était pas en parfaite cohérence avec les besoins de la maîtrise d’ouvrage.
* **Travaux de support applicatif :** tâches de support applicatif de niveau 1 aux utilisateurs de l’application Carthage.

## Présentation des objectifs et du choix des technologies

### Objectifs

L’objectif de l’application était donc de remplacer l’application existante écrite en C++, par une nouvelle application écrite en java pour qu’elle soit maintenue par la Tierce Maintenance Applicative (TMA), qui est spécialisée sur la technologie Java.

### Choix des technologies

L’application interface Carthage-Ramp utilise différentes technologies et des librairies externes.

L’implémentation de l’application a été entièrement dévéloppé en **Java**.  
Nous avons utilisé une librairie Open source **Spring Jdbc** pour la connexion à la base de données de Carthage.

Pour effectuer les tests, nous avons utilisé la librairie **JUnit** pour effectuer des tests unitaires à chaque itération de développement de l’application.

La récupération des données depuis le système d’informations distant RAMP, se fait par l’intermédiaire d’une libraire externe Westminster qui est un librairie Windows.

Cette librairie sert d’interface entre l’application Carthage-Ramp et le système d’informations des salles de marché RAMP.

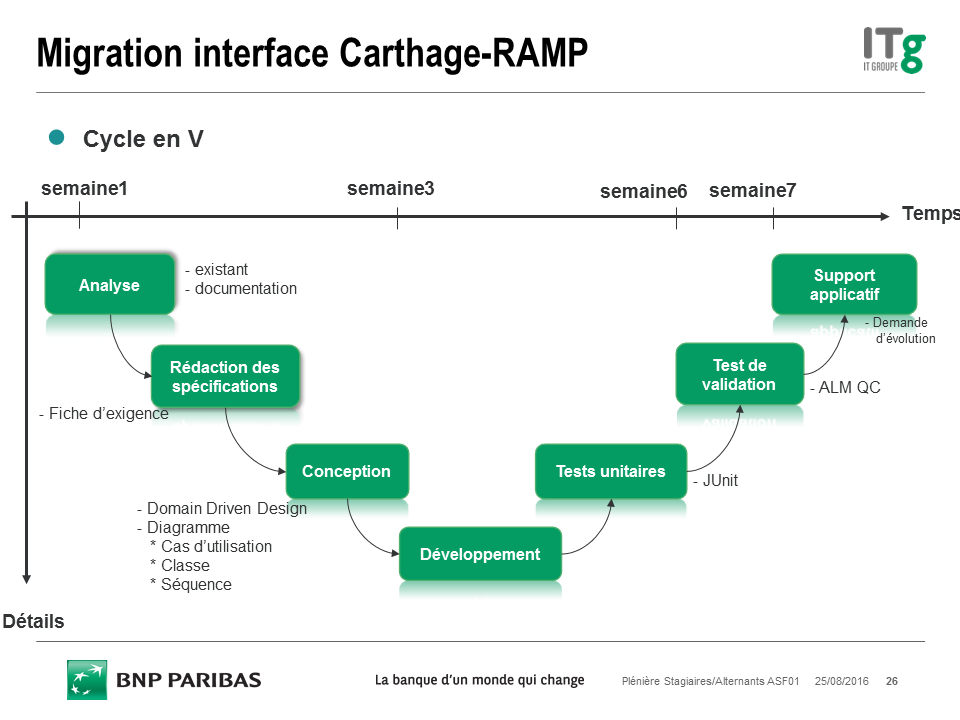
C:\StageBNP\Tmp\JUnit.png

### Choix organisationnels

Pour la réalisation du projet, nous avons fait un mixte de méthode Agile, et de cycle en V pendant chaque itération de la mise en place du projet.

#### Cycle en V

Cycle de développement du projet



#### Mise en production

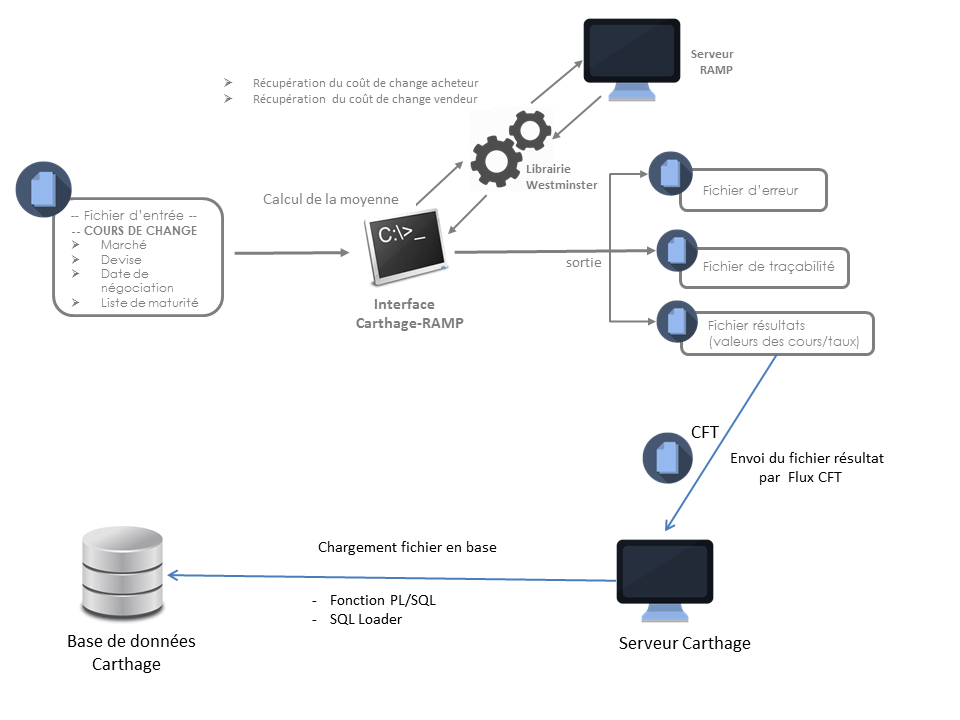
La mise en production de l’application CarRamp nécessite une bonne installation de la librairie Westminster, et l’installation d’une JRE Java 32 bits sur la machine d’exécution, car la bibliothèque externe.

Nous avons installé l’application CarRamp JAVA sur les serveurs de PROD carthage en commençant par la machine non active. Une fois l’installation validée, nous l’avons installé sur le nœud actif.

# Analyse Personnel

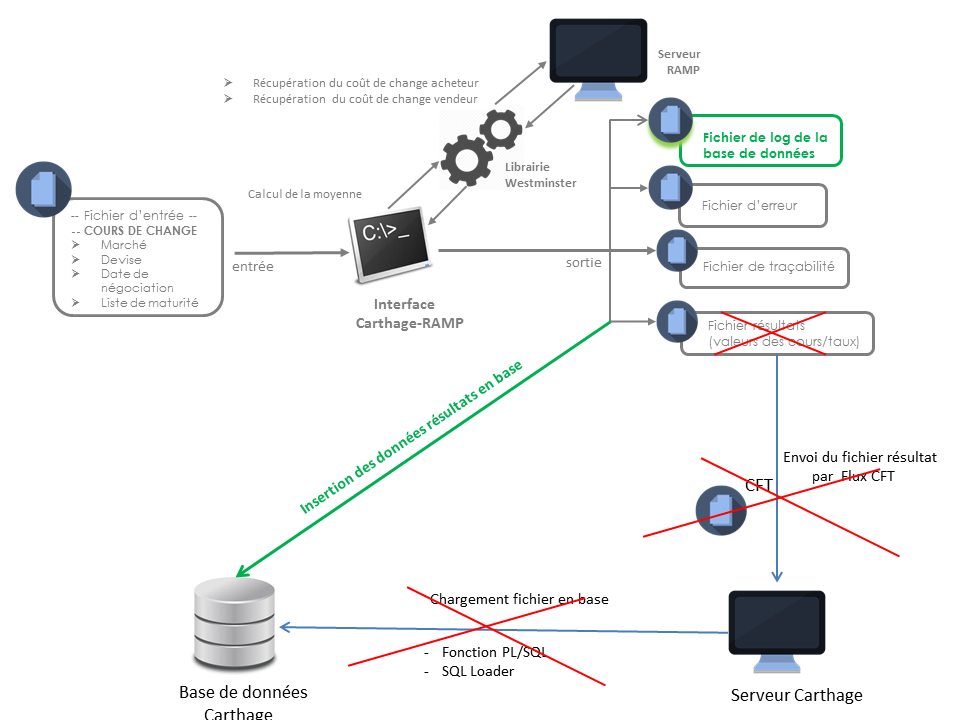
L’ancienne architecture de l’application, l’interface Carthage-Ramp enregistrait les données récupérées depuis Ramp dans un fichier de sortie.

Ce fichier de sortie était ensuite envoyé par transfert CFT au serveur Carthage. Une fois, le fichier reçu dans Carthage, traitement est automatiquement lancé pour charger le fichier reçu dans une table temporaire qui sert de table de traitement.  
Après l’insertion des données dans la table de traitement, l’application fait appel à des procédures stockées, et des fonctions PL/SQL insérer les valeurs de taux et cours de change dans la base de données Carthage.



Sachant que le serveur hébergeant l’application CarRamp est différent du serveur de l’application Carthage. Les résultats générés sur le serveur CarRamp sont donc envoyer par flux CFT sur le serveur principal de Carthage qui s’occupe de charger les fichiers dans la base de données.

Afin d’éliminer le transfert par flux CFT, et ne plus passé par l’intermédiaire du serveur de l’application Carthage, nous avons directement décidé de mettre les résultats obtenus dans la base de données en utilisant une couche logicielle qui nous sert d’abstraction entre l’application CarRamp et le serveur de base de données.



Pour éliminer le transfert par flux CFT du fichier de résultats, on avons supprimé le fichier de sortie, et décidé de créer un fichier de traçabilité de la base de données nous permettant de retracer toutes les données insérées en base de données.

Cela nous permet de gagner en efficacité et en rapidité, et aussi permet de ne plus passé par l’intermédiaire d’un autre serveur (serveur de Carthage), et permet de diminuer la charge de traitement de ce serveur.

# Regard critique (Recul par rapport au travail réalisé)

Face à la solution mise en place, si l’application était à refaire, j’aurais essayé d’améliorer la conception pour pouvoir prendre en compte les différentes évolutions de la librairie DLL externe (Westminster) utilisée par l’interface CarRamp.

Pour prendre en compte la scalabilité de l’application CarRamp, nous avons pensé à définir dans une interface répertoriant les différentes fonctions contenues dans la librairie externe, et utilisé l’interface dans l’application.

Aujourd’hui, nous chargeons en mémoire de la librairie DLL [Westminster], et faisons appel à une classe contenant des fonctions natives, qui sont donc appelées dans l’application CarRamp.

# Apport du master MIAGE

Les cours enseignés durant mes trois années scolaires en MIAGE, m’ont permis d’appréhender les missions confiées pendant mon stage, et de savoir gérer un projet partant d’une expression de besoins client jusqu’à la mise en production de l’application.

La migration de l’application CarRamp n’a pas été simplement une migration vers une nouvelle technologie, mais nous avons également apporté des améliorations sur l’architecture existante.  
Sachant que la plupart des données traitées aujourd’hui dans l’ancien système était envoyés par transfert de flux CFT d’un ordinateur à l’autre pour être ensuite insérer en base de données.  
Nous avons remplacé les transferts de fichiers en ordinateurs par une communication direct avec la base de données.

Les différents cours de bases de données avancées, et de Persistance Objet, et de langages de programmation appris pendant ces années m’ont permis de pouvoir réaliser ce projet de migration de l’application CarRamp en Java. Ce projet qui nécessitait une bonne base en base de données et algorithmique, que j’ai pu acquérir tout au long de ma formation avec des professeurs de qualité tous experts dans leurs domaines.

# Compétences acquises

Durant ce stage, j’ai découvert le secteur bancaire qui m’était assez obscure, et le domaine de l’ALM Trésorerie. J’ai acquis de nouvelles compétences techniques, fonctionnelles et également améliorer mes compétences relationnelles.

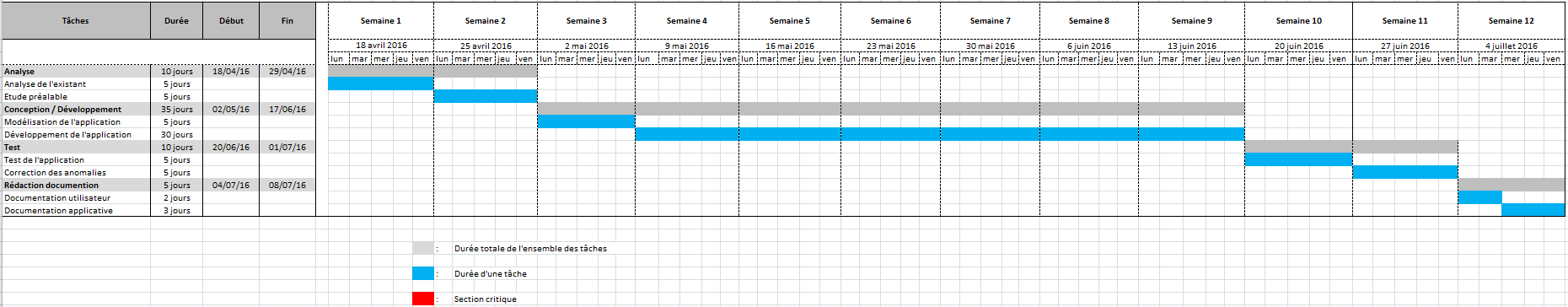
Pour les compétences fonctionnelles, j’ai appris à analyser les besoins client (MOA), à rédiger les spécifications fonctionnelles, à rédiger une documentation de conception complète, à participer à la phase de recette applicative, et enfin à la mise en production de l’application.

J’ai également appris le référentiel de l’architecture applicative de BNP Paribas, qui est le Domaine Driven Design (DDD) qui consiste à séparer en plusieurs couches son application en vue de facilité la maintenance.

En tant que compétences techniques, j’ai appris des nouveaux Framework open source **Spring** et **Spring Jdbc**, qui m’ont servi de couches applicatives pour interagir avec la base de données.

Les différentes compétences acquises me permettent de me positionner en tant que IT Business Analyst au sein de groupe, et peuvent ensuite me diriger vers le métier de maîtrise d’ouvrage, et vers le métier d’architecte de système d’informations après plusieurs années d’expériences.

# Planning du projet



# Glossaire

**ALM:** Asset Liability Management

**CFT:** **C**ross **F**ile **T**ransfert

**APAC:** **A**sia **PAC**ific

**WMI:** **W**ealth **M**anagement **I**nternational

# Bibliographie

CFT: <http://jfgg.free.fr/cmsms/sources/doc-cft.pdf>

ALM Trésorerie : <http://www.finance-banque.com/ALM-Asset-Liability-Management.html>

Cours de change spot Forex :

<http://www.fimarkets.com/pages/operations_change.php>

<http://www.iotafinance.com/Article-Fiche-d-instrument-le-change-au-comptant-FX-Spot.html>

Taux Swap : <http://www.fimarkets.com/pages/swap_taux.php>

Destination74 : <http://www.cambiste.info/sdmpage/prodswap/swapaut10.php>