

Réf : Ing-GI-2016-25

# Rapport de Projet de Fin d'Études

Pour obtenir le

**Diplôme d'Ingénieur en Génie Informatique**

**Option : Génie Logiciel et Informatique Décisionnelle**

Présenté et soutenu publiquement le 09 Septembre 2016

Par

**Wafa SALLEM**

## Conception, Réalisation et test de l'évolution d'une application Java/JEE de diffusion des contenus multimédia en IP Multicast

### Composition du jury

Madame	Meriem RIAHI	Présidente
Madame	Hajer KRICHENE	Rapporteur
Monsieur	Xavier BATTAS	Encadrant Entreprise
Madame	Ahlem BEN YOUNES	Encadrante ENSIT

Année universitaire : 2015-2016

# Remerciements

Je tiens à remercier tout particulièrement l'ensemble du personnel de la société Quadrille Ingénierie pour m'avoir accueilli et accepté parmi eux.

Je tiens à exprimer toute ma reconnaissance à mes tuteurs de stage :

Monsieur Xavier BATTAS président de la société Quadrille Ingénierie, je le remercie de m'avoir encadré, orienté, aidé et conseillé.

Madame Ahlem BEN YOUNES maître assistante à l'Ecole Nationale Supérieure d'Ingénieurs de Tunis ENSIT , elle m'a guidée dans mon travail et m'a aidée à trouver des solutions pour avancer.

J'adresse mes sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté à me rencontrer et répondre à mes questions durant mes recherches.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

## Dédicace

Je dédie ce travail à :

Mes parents

Ma mère, qui a oeuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi.

Mes frères et soeurs qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité.

Mes professeurs de l'ENSIT qui doivent voir dans ce travail la fierté d'un savoir bien acquis.

A mes adorables amies : Ala, Zahra et Khoubeib

Votre amitié est un honneur et une fierté pour moi. Je vous remercie d'avoir embelli ma vie par de précieux moments de bonheur. J'espère du fond de mon coeur que vous trouverez dans ce travail l'expression de mon amour et de mon estime les plus sincères. Que Dieu, le tout puissant éclaire votre vie de santé, de bonheur et de succès.

# Sommaire

Introduction générale.....	1
Chapitre 1 : Contexte général.....	3
Introduction .....	3
1.1. Présentation de la société .....	3
1.1.1. Solusion QuadriFast.....	4
1.1.2. Clients de la société .....	6
1.2. Présentation de projet .....	8
1.2.1. Contexte de projet.....	8
1.2.1.1. Serveur de téléchargement Unifié .....	8
1.2.1.2. Serveur de téléchargement Multicast .....	9
1.2.1.3. Périmètre fonctionnel .....	9
1.2.2. Etude et critique de l'existant .....	11
1.2.3. Travail demandé.....	12
1.3. Planification des tâches .....	12
Conclusion.....	13
Chapitre 2 : Analyse et spécification des besoins .....	15
Introduction .....	15
2.1. Etude préliminaire .....	15
2.1.1. Recueil des besoins non fonctionnels .....	15
2.1.2. Recueil des besoins fonctionnels .....	15
2.2. Identification des acteurs et des cas d'utilisation .....	17
2.2.1. Identification des acteurs .....	17
2.2.2. Modélisation objet.....	17
2.2.3. Diagramme de cas d'utilisation général .....	18
2.3. Raffinement des cas d'utilisation .....	18
2.3.1. Raffinement de cas d'utilisation « S'authentifier » .....	18
2.3.2. Raffinement de cas d'utilisation « Scrutation et contrôle des sources ».....	20
2.3.3. Raffinement de cas d'utilisation « Actualisation d'un QP».....	21
2.3.4. Raffinement de cas d'utilisation « Gestion de diffusion ».....	23
2.3.5. Raffinement de cas d'utilisation « Administration du module QP » .....	25

Conclusion.....	27
Chapitre 3 : Architecture et conception.....	28
Introduction .....	28
3.1. Méthodologie et formalisme adopté.....	28
3.2. Conception générale .....	29
3.2.1 Modèle MVC .....	29
3.2.2. Architecture n-tiers .....	30
3.3. Conception détaillée .....	31
3.3.1. Aspect statique.....	32
3.3.1.1. Diagramme de paquetage.....	32
3.3.1.1. Diagrammes des classes : Paquetage de diffusion MCAST .....	33
3.3.2. Aspect dynamique .....	35
3.3.2.1. Diagramme de séquence « S’authentifier » .....	35
3.3.2.4. Diagramme de séquence « Scrutation manuelle d’une source ».....	37
3.3.2.5. Diagramme de séquence de « Import d’un QP dans le STU » .....	39
3.4. Structure de la base de données.....	40
3.4.1. Modèle physique de données.....	40
Conclusion.....	41
Chapitre 4 : Réalisation et test.....	42
Introduction .....	42
4.1. Environnement de travail .....	42
4.1.1. Environnement matériel .....	42
4.1.2. Environnement logiciel.....	44
4.1.3. Outils de programmation .....	45
4.2. Description des aperçus de l’application STM .....	45
4.2.1. Interface d’authentification .....	45
4.2.2. Interface principale .....	46
4.2.3. Interface de scrutation d’un QP .....	47
4.2.4. Interface de gestion des actions .....	47
4.2.5. Interface de détail d’une planification .....	48
4.2.6. Interface reporting : suivi du parc .....	49
4.2.7. Interface de purge des paquetages et des planification .....	50
4.3. Tests unitaires.....	51

4.3.1. Test en « boîte noir » .....	51
4.3.2. Test en « boîte blanche » .....	51
Conclusion .....	57
Conclusion générale .....	58

# Liste des figures

Figure 1: Organigramme de la société .....	4
Figure 2: Aperçus du MCarousel .....	5
Figure 4: Logo FDJ .....	7
Figure 5: Logo LotSys.....	7
Figure 6: Réalisation LotSys .....	8
Figure 7: Architecture logicielle et matérielle.....	10
Figure 8: Alerte enlèvement.....	12
Figure 9: Diagramme de Gantt: Chronogramme des tâches .....	13
Figure 10: Diagramme de cas d'utilisation générale .....	18
Figure 11: Diagramme de cas d'utilisation « S'authentifier ».....	19
Figure 12: Diagramme de cas d'utilisation « Scrutation d'une source ».....	20
Figure 13: Diagramme de cas d'utilisation « Actualisation d'un QP » .....	22
Figure 14: Diagramme de cas d'utilisation « Gestion de diffusion » .....	24
Figure 15: Diagramme de cas d'utilisation « Administration du module QP ».....	26
Figure 16: Cycle de vie: Modèle en V .....	29
Figure 17: Modèle MVC .....	30
Figure 18: Architecture n-tiers .....	31
Figure 19: Diagramme de paquetage .....	32
Figure 20: Diagramme des classe.....	34
Figure 21: Diagramme de séquence « S'authentifier » .....	36
Figure 22: Diagramme de séquence « ordre de reboot ».....	37
Figure 23: Diagramme de séquence « Scrutation manuelle d'une source » .....	38
Figure 24: Diagramme de séquence « Import d'un QP dans le STU ».....	39
Figure 25: Modèle physique de données de paquetage de diffusion MCAST.....	41
Figure 26: Description de l'environnement matériel .....	43
Figure 27: Description de l'environnement logiciel .....	44
Figure 28: Interface d'authentification.....	46
Figure 29: Interface principale du STM .....	46
Figure 30: Interface de diffusion et scrutation d'un QP .....	47
Figure 31: Interface de création d'une planifications.....	48

Figure 32: Détails d'une planification .....	49
Figure 33: Suivi d'un parc .....	50
Figure 34: Interface de purge d'une planification et d'un QP.....	50
Figure 35: Diagramme d'activité « Mise à jour d'un QP ».....	xiii
Figure 36: Diagramme d'activité « Lecture et initialisation des sources » .....	xiv
Figure 37 Transfert d'un QP.....	xv
Figure 38: Diagramme d'activité « Archivage d'un QP » .....	xvi



# Liste des tableaux

Tableau 1: Description relatif au cas d'utilisation « S'authentifier » .....	19
Tableau 2: Description relatif au sous cas d'utilisation « Contrôle du certificat ».....	21
Tableau 3: Description relatif au sous cas d'utilisation « Mise à jour d'un QP ».....	23
Tableau 4: Description relatif au sous cas d'utilisation « Diffusion d'un QP ».....	25
Tableau 5: Description relatif au sous cas d'utilisation « Envoie ordre de démarrage d'AE »	27
Tableau 6: Récapitulatif des tests.....	52
Tableau 7: Installation et configuration du serveur SSH .....	52
Tableau 8: Configuration de la plateforme de test .....	53
Tableau 9: Description du test « QP_SCAN_7814_0 » .....	53
Tableau 10: Description du test « QP_SCAN_7814_1 » .....	54
Tableau 11: Description du test « QP_SCAN_7814_2 » .....	54
Tableau 12: Description du test « QP_SCAN_7814_3 » .....	55
Tableau 13: Description du test « QP_SCAN_7814_4 » .....	55
Tableau 14: Description du test « QP_SCAN_7814_5 » .....	56
Tableau 15: Description du test « QP_SCAN_7814_6 » .....	56
Tableau 16: Description du test « QP_SCAN_7814_7 » .....	57

# Abréviation

<b>AE</b>	Alerte d'enlèvement
<b>FDJ</b>	Française Des Jeux
<b>FTP</b>	File Transfert Protocol
<b>PDV</b>	Point de vente
<b>QP</b>	Quick Pack
<b>STB</b>	Set Top Box
<b>STU</b>	Serveur de Téléchargement Unicast
<b>STM</b>	Serveur de Téléchargement Multicast
<b>TCP</b>	Transmission Control Protocol
<b>UDP</b>	User Datagram Protocol
<b>VOD</b>	Video On Demand

## Introduction générale

Les images de télévision diffusées en France correspondent toujours au standard retenu en 1967 lors de l'introduction de la couleur et modifié en 1978. La technologie a accompli d'immenses progrès. Dans ce contexte, l'objectif que s'assigne la télévision à haute définition (TVHD) est d'accroître la résolution des images diffusées pour améliorer la qualité perçue par les téléspectateurs.

Aujourd'hui, la vidéo est devenue un type de données numériques parmi d'autres (données, images, musique, etc.). Ainsi, la transmission des programmes de télévision sera assurée non seulement par les sociétés de distribution classiques (câblo-distributeurs, etc.), mais également par des opérateurs de télécommunications via leurs réseaux fixes ou mobiles.

Le passage de la télévision numérique terrestre (TNT) à la haute définition (HD) traduit une évolution des normes de diffusion de la télévision via les ondes électromagnétiques terrestres, les réseaux des câblo-opérateurs, les satellites, Internet ou encore les réseaux mobiles. Actuellement une seule norme est utilisée pour diffuser les chaînes de la TNT en France : le MPEG-4.

A partir du 5 avril 2016, la norme MPeG-2 a été abandonnée au profit de la norme MPeG-4, plus récente et plus performante, ce qui permettra le passage à la TNT HD.

Le développement des logiciels informatiques dans ce secteur se trouve face à des enjeux où il ne suffit pas tant d'innover que d'assurer la qualité des logiciels produits. Dans le monde compétitif d'aujourd'hui, la satisfaction des clients est la clé de la réussite. Cette satisfaction découle le plus souvent d'une bonne compréhension des besoins du client par l'entreprise pour améliorer la qualité de ses produits et de ses services tout en ajustant les écarts possibles lors de l'identification des besoins.

Afin d'assurer l'alignement aux besoins du client et d'éviter toute éventuelle incertitude lors du développement de ses spécifications, une entreprise dans le domaine de la télévision numérique doit adopter des approches lui permettant de répondre conformément aux exigences de ses clients. C'est dans ce contexte que s'inscrit la société Quadrille Ingénierie spécialisée dans le domaine de TV numérique depuis 18 ans.

Notre projet de fin d'étude vise à faire évoluer le projet « Serveur de Téléchargement Multicast » de la « Française Des Jeux ».

Le travail demandé consiste dans un premier temps à examiner de près le périmètre fonctionnel de l'application pour comprendre et élaborer les scénarios possibles de chaque fonctionnalité et de les implémenter.

Dans un second temps, nous implémentons un cahier des tests et nous évaluons le système dans son milieu d'exploitation.

Le présent rapport synthétise tout le travail que nous avons effectué. Il est organisé en quatre chapitres comme suit :

- Le premier chapitre intitulé « *Contexte général* » présente la société et l'étude de l'existant du projet FDJ sur lequel nous allons travaillé.
- Le second chapitre intitulé « *Analyse et spécification des besoins* » présente les différents besoins fonctionnels et non fonctionnels auxquels doit répondre l'outil.
- Le troisième chapitre intitulé « *Architecture et conception* » décrit l'architecture et la conception détaillée de l'évolution du projet.
- Le quatrième chapitre intitulé « *Réalisation et tests* » décrit les tâches accomplies ainsi que l'évaluation de système par la réalisation d'une série de tests.

# Chapitre 1 : Contexte général

## Introduction

La mise en œuvre de notre projet nécessite une étude approfondie sur certaines notions qui touchent non seulement au cadre générale de projet, mais aussi à la réalisation. Pour bien assimiler ces différentes notions, cette section est consacré à présenter l'organisme d'accueil et la solution existante ainsi que la présentation de travail demandé.

### 1.1. Présentation de la société

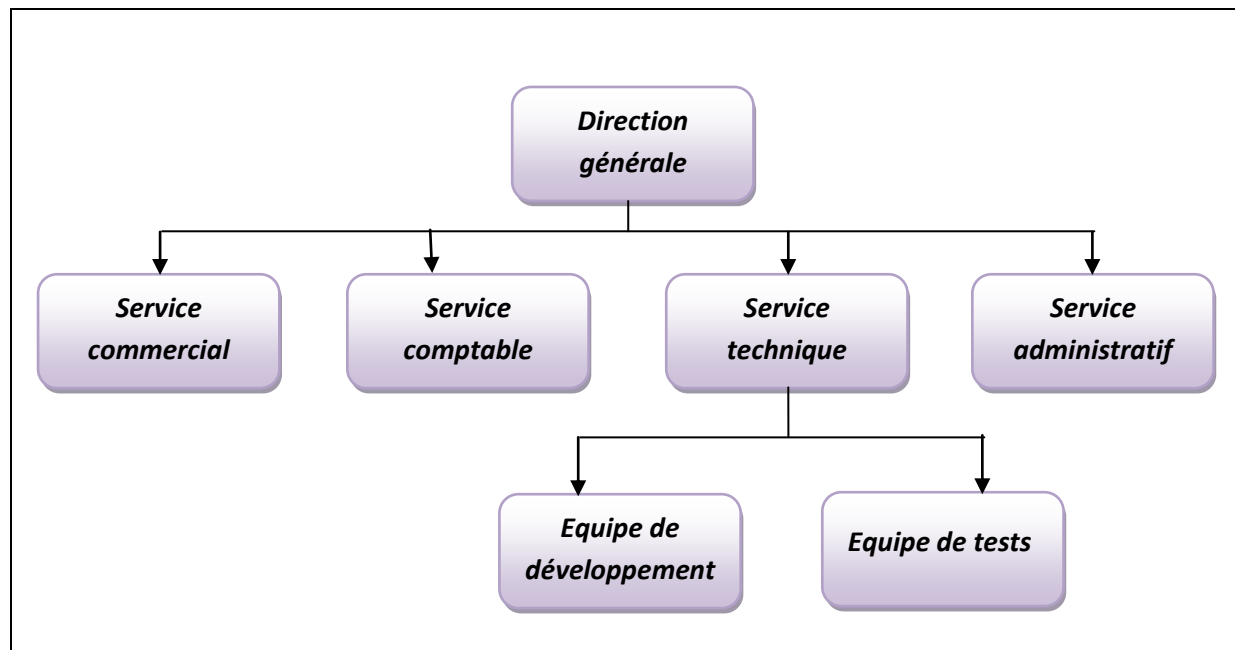
Fondée en 1998, Quadrille est une société de conseils et d'expertise de capital 729 970 euros spécialisée dans les métiers de l'Informatique Multimédia et Télécom , accompagnant les acteurs majeurs des médias et des télécoms à travers son expertise dans le design et le développement d'applications.

Quadrille offre une gamme de logiciels et services dans les métiers du Broadcast et du Broadband et plus spécialement sur les applicatifs dédiés à la télévision numérique, Vidéo à la Demande, et aux systèmes d'encodage et IT.

Quadrille a développé une suite applicative QuadriFast™ qui permet l'envoi de fichiers en mode push sur des réseaux Multicast Satellite, TNT et IP xDSL. Cette plateforme de gestion de contenus sécurisés permet aux opérateurs de proposer à leurs abonnés les dernières innovations de services de télévision (Push VOD, télévision de rattrapage, Canal Virtuel) tout en réduisant leurs coûts de bande passante réseau. La plateforme QuadriFast a déjà été déployée sur de nombreux sites auprès d'acteurs majeurs du secteur. [1]

C'est une société qui intègre 4 types de services : service commercial, service administratif, service comptable et service technique dont ce dernier est composé de deux équipes : une équipe de développement et une équipe des tests.

La figure suivante illustre l'organigramme de différents services de la société.



**Figure 1: Organigramme de la société**

Nous avons travaillé au sein de l'équipe technique. Dans un premier temps, nous avons analysé, conçu et réalisé les fonctionnalités demandées.

Dans un second temps, nous avons élaboré un cahier des tests qui décrit les tests de la non-regression ainsi que les tests des fonctionnalités implémentées et évalué le système.

### **1.1.1. Solution QuadriFast**

QuadriFast© est une solution logicielle multi plateforme de transfert de données fiable en mode push au travers d'un réseau IP. Grâce à QuadriFast©, un fichier est transféré simultanément vers tous les récepteurs qui sont à l'écoute. La technologie Multicast permet une utilisation efficace de la bande passante, et donc une prévention de la congestion du réseau tout en garantissant l'intégrité du contenu grâce à la voie de retour.

Un système QuadriFast™ est une solution de transfert de contenu multimédia. Il permet de délivrer tout type de contenu, sur tout type de réseau, y compris le satellite. Basée sur une technologie robuste de transfert Multicast de fichiers développée par Quadrille, QuadriFast™ permet notamment de :

- Recevoir et gérer un catalogue de programmes à transférer.

- Organiser le transfert de fichiers et des données associées suivant une priorisation et suivant la bande passante disponible.
- Recevoir les contenus, vérifier leur intégrité et les stocker sur des unités distantes.

L'architecture logicielle de QuadriFast comprend trois grandes composantes :

- Les outils **d'administration et d'ordonnancement** qui fonctionnent en interface web et tournent sur serveurs Dell.



**Figure 2: Aperçus du MCarousel**

- Les outils de **diffusion** connectés aux têtes de réseau des diffuseurs. Les contenus sont empaquetés avec un descripteur XML contenant les métadonnées décrivant les vidéos envoyées. Les fichiers sont poussés dans le canal de diffusion broadcast, le système gère la correction d'erreurs à la réception et la diffusion multiple pour obtenir de la redondance.
- Les outils de **réception** installés dans les décodeurs, qui comprennent aussi des logiciels de test.

A noter que le diffuseur peut décider d'envoyer en push-VOD des spots publicitaires qui seront intégrés dans les programmes consommés, le plus souvent en mode "pre-roll" (diffusion avec la vidéo demandée). Cela peut servir notamment à diffuser des bandes annonces de films à consommer ultérieurement.

Le système QuadriFast est constitué des applications tête de réseau et l'agent Push embarqué dans les décodeurs (agent MStore).

L'application tête de réseau est composé de :

- **MIngest** : c'est une application web qui ingère et gère les contenus. Lorsque le début de la période de diffusion delta atteint, MIngest transfère le contenu à Mcarousel.
- **MCarousel** : c'est une application web qui gère la planification des diffusions. Cette application permet également de gérer la bande passante attribué au service PushVOD.
- **Encapsulator** : c'est une application automatique en charge d'adapter les fichiers à diffuser au format TS over IP.
- **MSender** : c'est une application automatique en charge de la diffusion des fichiers sur les réseaux IP.
- **MStore** : stockage des contenus.

### 1.1.2. Clients de la société

Quadrille occupe aujourd'hui une vingtaine de collaborateurs. Leurs clients sont pour l'essentiel français avec notamment **Canal+**, **TDF** et la **Française des Jeux**, etc. Ils ont un client étranger, **Digitalb**, un opérateur de TV payante en Albanie.

Quadrille a mis en place QuadriFast chez la **Française des Jeux** qui équipe ses points de vente avec une Set To Box connectée d'origine Sagemcom en réseaux IP sur lignes spécialisées et qui poussent des contenus toutes les nuits. [2]

Chaque STB située dans les points de vente joue en continu des données multimédia qui sont diffusées et rafraîchies par QuadriFast©.

La Française des jeux est une entreprise publique française créée en 1976 détenue à 72 % par l'État français qui lui a confié le monopole des jeux de loterie et de paris sportifs sur tout son territoire et à Monaco. Aujourd'hui, la Française des Jeux comptent 28,5 millions de joueurs par an sur plus de 38 700 points de vente pour un chiffre d'affaires de près de 9,3 milliards d'euros en 2007. [3]





**Figure 3: Logo FDJ**

LotSys est une société par actions simplifiées unipersonnelle unique et en activité depuis 18 ans. Elle est spécialisée dans le secteur d'activité de la programmation informatique. Son capital est 2 943 250 euros. [4]



**Figure 4: Logo LotSys**

LotSys est la filiale technologique de développement de FDJ. Elle a pour objet le développement, la fabrication, la commercialisation et la fourniture de matériels, de logiciels et de services, en relation avec les jeux de hasard et terminaux de prise de jeux telque le développement du portail de la FDJ pour la navigation sur tablette. [5]

**FDJ** Jeux Résultats Mon compte

EURO MILLIONS LOTO Keno gagnant à vie

Prochain jackpot de 2 M€\*\*\* à gagner au LOTO®

**Mercredi 10 Décembre 2014**

**Tirage :**

27 11 16 3 8 N° chance 6

**JOKER+ 1 556 442**

[Plus de résultats](#)

**Détail LOTO®**

Numéros	CHANCE	Grilles gagnantes	Gains
5	●	1	9 000 000,00 €
5	-	5	49 485,20 €
4	-	1326	401,50 €
3	-	38973	5,90 €
2	-	416689	3,90 €
-	●	481928	2,00 €

\*\*\* Montant à partager entre, les gagnants du 1er rang ou en cas d'absence de gagnants au 1er rang après 34 tirages consécutifs au rang inférieur ayant au moins 1 gagnant. Voir règlement.

[Interdiction volontaire de jeux](#)  
Toute personne souhaitant faire l'objet d'une interdiction volontaire de jeux doit la faire elle-même auprès du ministère de l'intérieur. Cette interdiction est valable dans les casinos, les cercles de jeux et sur les sites de jeux en ligne autorisés en vertu de la loi n° 210-476 du 12 Mai 2010. Elle est prononcée pour une durée de trois ans non renouvelable.

[Accès aux mentions légales et règlements](#)

[Jouez](#) [Disponible sur App Store](#)

[Pour que le jeu reste un plaisir](#) [En savoir plus](#) [Charte Vie privée](#) [Contact](#) [FDJ Store](#)

Figure 5: Réalisation LotSys

## 1.2. Présentation de projet

### 1.2.1. Contexte de projet

#### 1.2.1.1. Serveur de téléchargement Unifié

Le *Serveur de Téléchargement Unifié* est une application développée par LotSys. Il a pour objectif de mettre et maintenir à jour les fichiers nécessaires à l'exploitation des équipements déployés en PDV et prévus pour supporter cette fonctionnalité. Il est dit « unifié » dans le sens où il supporte n'importe quel type de contenu pour plusieurs types de plateformes matérielles différentes.

Il assure le mode de diffusion en unicast, c'est le mode de mise à jour le plus sûr. Ses principaux caractéristiques sont :

- Contrôle synchrone de l'état des versions des paquetages présents sur l'équipement (par exemple au redémarrage).
- Téléchargement unitaire d'un équipement sous le contrôle du serveur, afin de:
  - Restaurer un équipement ayant une version de paquetage corrompue.

- Mettre à jour un équipement ayant une version obsolète, pour lequel le téléchargement par diffusion ne serait pas intéressant.
- Forcer le téléchargement d'une version lors du contrôle des versions d'un équipement.
- Protocole de transfert TCP/IP, donc garantie d'acheminement des données via un chemin réseau sûr.

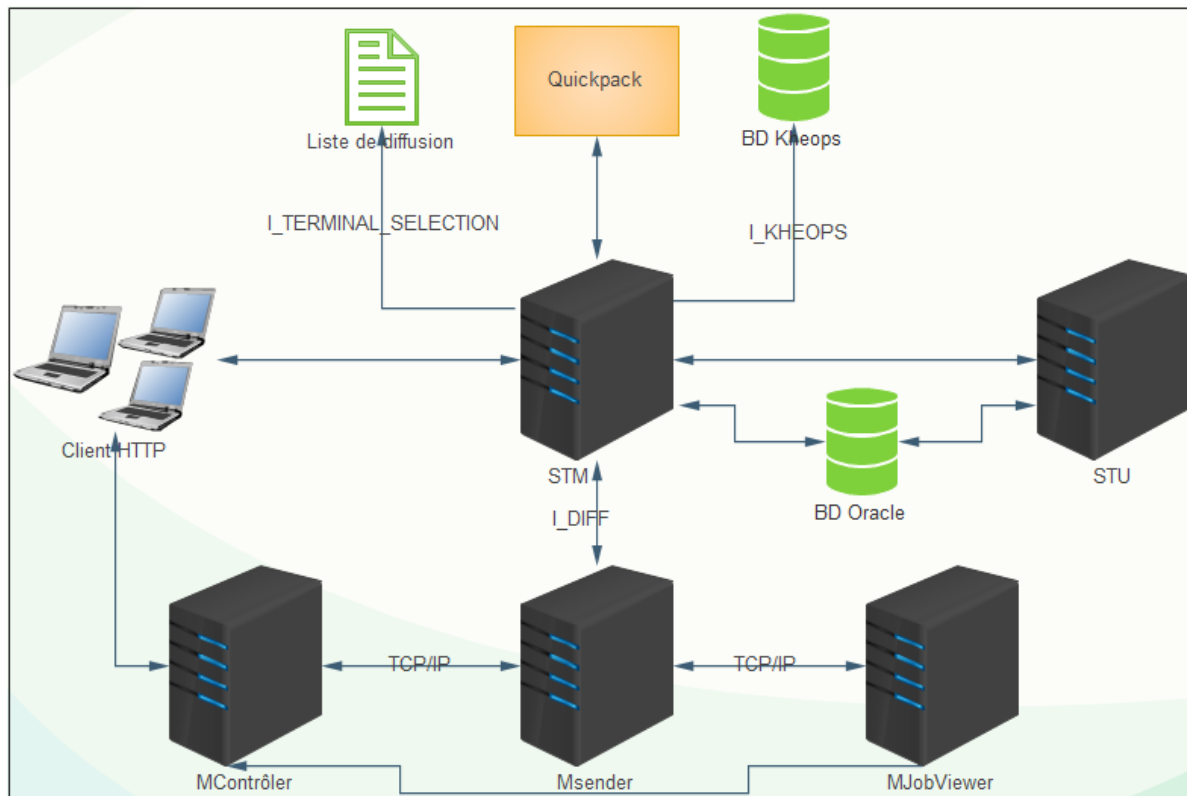
#### **1.2.1.2. Serveur de téléchargement Multicast**

Le *Serveur de Téléchargement Multicast* est une application développée par Quadrille. Il a pour objectif de mettre et maintenir à jour les paquetages ainsi que leurs diffusions vers tous les terminaux simultanément. Ses principales caractéristiques sont :

- Déploiement massif d'une nouvelle version de paquetage sans tenir compte du contexte.
- Possibilité de déterminer une liste d'équipements cible.
- Présence d'un canal de retour pour chaque équipement permettant d'assurer :
  - Une diffusion incrémentale : Le STU établit la liste des fichiers nécessaires à la complétion du paquetage cible préalablement à l'opération.
  - Un bilan de diffusion : Le serveur collecte le statut de chaque équipement afin d'établir un bilan de diffusion.
- Protocole de transfert UDP/IP, donc acheminement des données soumis aux aléas du réseau.

#### **1.2.1.3. Périmètre fonctionnel**

La figure ci-dessous décrit les éléments matériel et logiciel qui correspondent à l'architecture FDJ.



**Figure 6: Architecture logicielle et matérielle**

- **MSender** : c'est une application qui prend en charge la diffusion des paquetages sur des réseaux de type IP Multicast, elle est contrôlée par le STM. MSender est développée en C++ et fonctionne sur RedHat 64 bits.
- **MCôntrolier** : se connecte en TCP au MSender et permet de configurer, superviser les jobs en cours, visualiser l'utilisation de la bande passante et visualiser les logs.
- **MJobViewer** : apparait avec la version 4.2.0. Elle permet de visualiser les jobs d'émission de façon analogue à MContrôler mais par l'intermédiaire d'une interface en ligne de commandes. L'application se connecte en TCP au MSender et permet de visualiser les jobs, les mettre en pause ou les relancer.
- **Interface I\_DIFF** : permet à STM de transmettre des fichiers en Multicast vers des Equipements au moyen de l'application MSender. Elle permet également à l'application MContrôler de visualiser l'état de fonctionnement de MSender et à l'application MJobViewer d'opérer les travaux d'émission ou jobs.
- **Interface I\_KHEOPS** : permet à STM d'accéder aux paramètres des terminaux. Cette interface est de type table Oracle.

- **Interface I\_TERMINALSELECTION** : permet à STM d'accéder à la liste des terminaux concernés par une diffusion. Ces fichiers contiennent une énumération des identifiants des terminaux qui doivent recevoir la diffusion.

### 1.2.2. Etude et critique de l'existant

La FDJ a besoin d'envoyer des informations vers les 35000 équipements dispersés dans ses PDV : afficheurs promotionnels, terminaux de prise de jeu, reçu-checkers, etc.

Dans un premier temps, Quadrille a fourni une solution permettant de transférer vers les afficheurs promotionnels des informations en IP Multicast. Ce système a ensuite été refondu et intégré à la solution de transfert unicast développée par Lotsys.

L'ensemble est nommé Serveur de Téléchargement Unifié, la partie gérée par Quadrille s'appelle Serveur de Téléchargement Multicast.

La solution fournie par Quadrille est composée de :

- Tête de réseau.
- STM.
- MSender.
- Autres applications.

La solution actuelle (STM) consiste à archiver un ensemble de fichiers dite paquetage, puis à transférer cette archive, qui dans le cas actuel se trouve sous le format ZIP, puis désarchiver sur la cible afin de permettre l'exploitation des fichiers.

Il existe trois types de paquetage :

- **Multimédia**: contenus destinés à être joués sur les afficheurs promotionnels.
- **Soft** : mises à jour logicielles des équipements. A noter que le STM n'est plus utilisé pour la mise à jour des terminaux de prise de jeu. C'est le site unicast qui s'occupe de la gestion de ce type de paquetage.
- **Quick Pack**: contenus issus de sources diverses, actualisés et envoyés plusieurs fois par jour, à destination des afficheurs promotionnels. Les cotes des paris sportifs imprimés dans les PDV sont envoyées par ce moyen. Le traitement de ces paquetages est plus rapide que celui des paquetages standard.

La FDJ souhaite diffuser d'autres types d'informations en plus des actualités des jeux en ligne. Les derniers sont diffusées par d'autres sources tels que les réseaux sociaux, les chaînes

de TV, les journaux. Parmi le type de données que la société souhaiterait le propagé est la signalisation des incidents grave comme la perte d'un enfant.

### 1.2.3. Travail demandé

Notre travail consiste en premier partie à ajouter une extension à application STM par la réalisation et l'intégration de la fonctionnalité d'AE au module QP. En deuxième partie notre travail consiste à l'évaluation de système et en particulier le module QP par la réalisation et la mise en place d'une série des tests.

Une AE est un outil mis en place pour signaler la disparition d'un enfant. La figure ci-dessous expose un exemple d'AE qui muni des données de différent type (Texte, image, information multimédia).

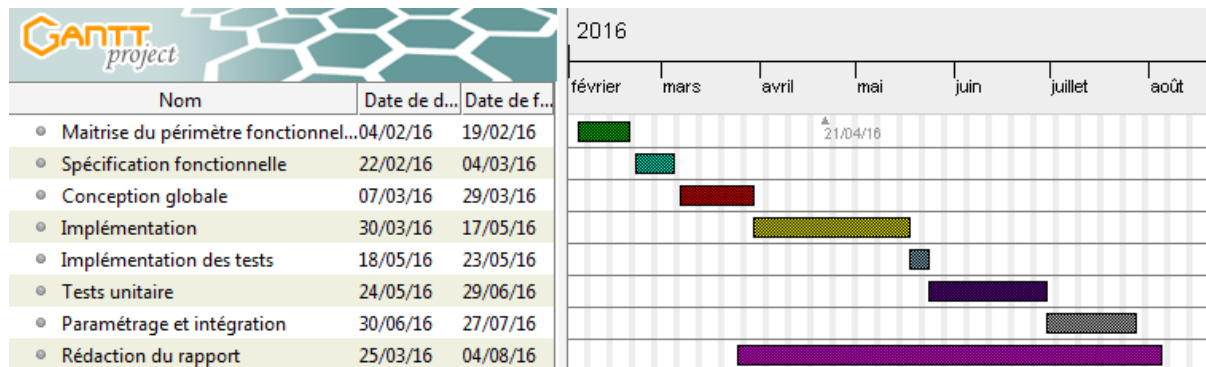


**Figure 7: Alerte enlèvement**

Cette évolution met en évidence le déploiement de la version 4.2.0 de l'application dorsale Multicast STM qui peut maintenant porter de l'information multimédia d'une source locale et être diffusée vers les terminaux au moyen d'un QP.

## 1.3. Planification des tâches

Afin de donner une vue globale de la répartition du temps par rapport au travail demandé, il est indispensable de tracer un diagramme qui décrit la répartition des tâches du projet tout au long de ces quatre mois de stage. Le diagramme de Gantt de la figure ci-dessous illustre le macro-planning de déroulement de notre projet.



**Figure 8: Diagramme de Gantt: Chronogramme des tâches**

Les étapes citées dans le macro-planning :

- *Maitrise de périmètre fonctionnel de la plateforme de diffusion Multicast* : Dans cette étape, nous avons maitrisé le périmètre fonctionnel de notre projet, rassembler et rédiger tous les règles métier relatives au périmètre et ceci à travers des réunions hebdomadaires avec le chef de projet.
- *Spécification fonctionnelle* : Dans cette étape, nous avons exprimé les fonctionnalités de l'évolution de l'application STM.
- *Conception globale* : Dans cette partie, nous avons concevoir les fonctionnalités de STM à l'aide des diagrammes UML ainsi de présenter l'architecture de notre projet.
- *Implémentation* : Pendant cette étape, nous avons traduit ce qui est conçu dans l'étape précédente dans un langage de programmation.
- *Implémentation des tests* : Dans cette étape, nous avons implémenté le cahier des tests en créant des liens de traçabilité entre les exigences du système et les tests. C'est l'étape la plus importante durant le cycle de développement de projet.
- *Tests unitaire* : Pendant cette étape, nous avons vérifier le bon fonctionnement de chaque module indépendamment des autres parties du système.
- *Paramétrage et intégration* : La dernière étape de projet est le paramétrage et l'intégration sur un vrai environnement de déploiement de la FDJ.

## Conclusion

Dans ce chapitre, nous avons mis le projet dans son contexte. Nous avons présenté l'organisme d'accueil Quadrille Ingénierie et ses différentes activités. Nous avons aussi défini les objectifs à atteindre et les principales fonctionnalités.

Ayant maintenant une idée générale à propos de l'application MSender ainsi que l'application STM et son périmètre technique, nous pouvons entamer aisément le prochain chapitre portant la phase d'analyse et spécification des besoins.



## Chapitre 2 : Analyse et spécification des besoins

### Introduction

Dans ce chapitre, nous identifions les acteurs du projet ainsi que les besoins fonctionnels et non fonctionnels à mettre en évidence. Ensuite, nous présentons les diagrammes des cas d'utilisation du projet.

Cette partie est une étape très importante pour la préparation de la phase de conception du produit et a pour but de permettre une meilleure visibilité et lisibilité de la vue globale du projet.

### 2.1. Etude préliminaire

Une exigence est une condition ou une capacité que doit posséder un système afin de satisfaire aux termes d'un contrat, d'une norme ou d'une spécification formellement imposée. Elle regroupe une spécification fonctionnelle et non-fonctionnelle.

#### 2.1.1. Recueil des besoins non fonctionnels

Afin de développer les fonctionnalités liées à la version 4.2.0 du STM, nous devons répondre à certains besoins non fonctionnels. Dans cette section nous couvrons l'essentiel de ces besoins.

- **Fiabilité** : Le système toujours fournit des résultats corrects.
- **Robustesse** : Le système doit réagir correctement même dans un environnement hostile.
- **Documentation** : Une documentation de chaque fonctionnalité accompagne le logiciel afin de faciliter son usage par nos clients FDJ.
- **Traçabilité** : Des liens de traçabilité entre les exigences du système et son code source sont définies afin de suivre la vie d'une exigence et de réduire l'effort de compréhension et de maintenance.

#### 2.1.2. Recueil des besoins fonctionnels

Les besoins fonctionnels sont des exigences qui décrivent les services proposés par le système.

L'application STM doit assurer les fonctionnalités ci-dessous. Elle sont présentées par demandes d'évolutions numérotées issue de la FDJ :

**7814. Scrutation et contrôle des sources.** Cette fonctionnalité consiste à scruter une source locale non signé. Lors de la scrutation d'une source, le module effectue les étapes suivantes :

- Copie intégrale de la source dans le répertoire d'archivage.
- Contrôle du certificat  $C_{part}$  et vérification de la validité du fichier  $F_{sign}$ .
- Vérification de la validité du fichier  $F_{sign}$  à l'aide de la clé publique contenue dans  $C_{part}$ .
- Vérification des hashes inscrits dans la référence.
- Vérification de la taille totale des fichiers de contenu de la source.
- Vérification du nombre de fichiers de contenu de la source.
- Vérification que chaque fichier de contenu respecte les masques autorisés.
- Déplacement des fichiers multimédia et de validité.
- Envoie d'un e-mail d'avertissement.

**7815. Actualisation du QP.** L'actualisation du QP consiste à générer le paquetage à destination du STU, l'importer dans la base de données, et rafraîchir les répertoires de version. Ce cas d'utilisation doit assurer les fonctionnalités ci-dessous :

- Suppression d'un contenu.
- Mise à jour d'un QP.
- Création de la signature « signature.pem ».
- Ajout du certificat « sdadmin\_cert.cer ».
- Destuffing des flux multimédia.
- Création d'un QP dans un archive.
- Import de QP dans le STU.

**7816. Gestion de diffusion.** L'application STM doit permettre de lancer la diffusion d'un QP manuellement vers 1 ou plusieurs terminaux en Multicast.

- Diffusion d'un QP.
- Suivre d'une diffusion.
- Mise à jour de fichier de résultat de fin de diffusion.
- Notification de fin de diffusion.
- Archivage des contenus envoyés et des STB les ayant reçus.

**7817. Administration du module QP.** Il est possible d'envoyer en Multicast un fichier de reboot, d'arrêt ou de démarrage des AE à un équipement depuis la page de gestion des équipements.

- Envoie d'un ordre de reboot.
- Envoie d'un ordre d'arrêt des AE.
- Envoie d'un ordre de démarrage des AE.

## 2.2. Identification des acteurs et des cas d'utilisation

### 2.2.1. Identification des acteurs

**Un acteur:** est une entité externe (personne ou objet) qui agit sur le système d'une façon directe (acteur principal) ou indirecte (acteur secondaire). Il est représenté par un pictogramme sous-titré par le nom de l'acteur.

L'application STM comporte deux acteurs principaux et un acteur secondaire :

- ***L'Administrateur*** : C'est un acteur principal de STM. Il a accès à tous les modules du projet STM, il est le seul qui administre le module de QP.
- ***L'utilisateur FDJ*** : Ce profil donne accès à l'ensemble des fonctionnalités attribuées à l'exploitant FDJ de l'application STM, il a accès au module de scrutation et diffusion de QP.
- ***Le système interne*** : C'est l'application STM qui s'occupe de l'exécution des fonctionnalités internes.

### 2.2.2. Modélisation objet

Pour la modélisation et la conception du travail réalisé, nous avons opté pour le formalisme UML (Unified Modeling Language) qui présente un ensemble de notation orienté objet de modélisation de systèmes. UML n'étant pas une méthode, on peut choisir d'utiliser des diagrammes particuliers selon la nécessité. [6]

Afin de donner une vision globale sur le comportement fonctionnel de notre application, nous avons utilisé le diagramme de cas d'utilisation qui permet de représenter de façon intuitive les différentes interactions entre les acteurs et le système sans se soucier du fonctionnement interne de l'application.

### 2.2.3. Diagramme de cas d'utilisation général

Suite à l'explication des besoins fonctionnels de projet STM et des objectifs des différents membres de l'équipe de développement de LotSys, nous illustrons les fonctionnalités recueillies via un diagramme global de cas d'utilisation.

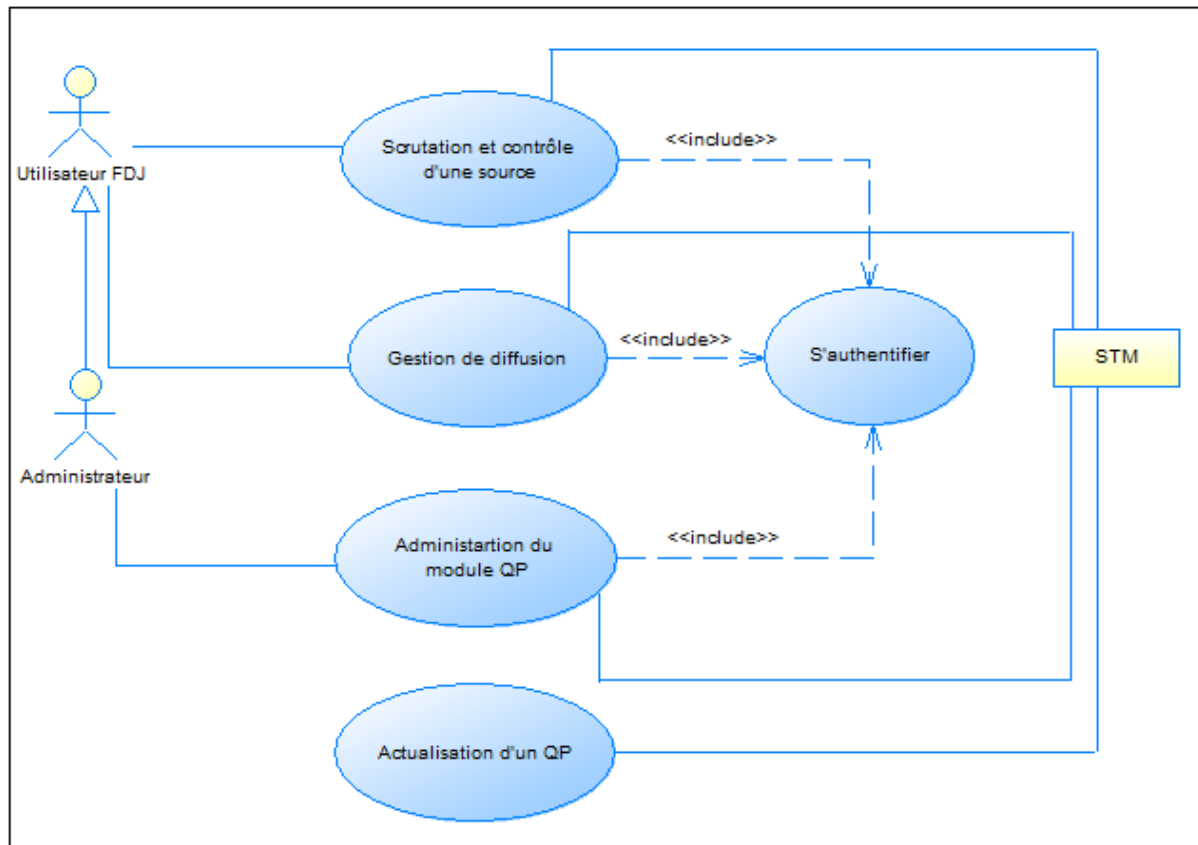


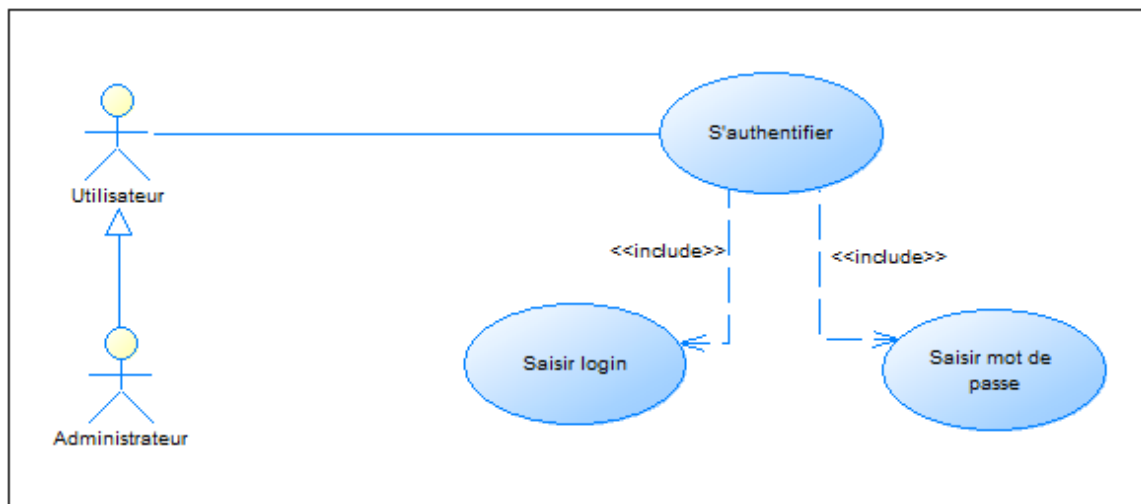
Figure 9: Diagramme de cas d'utilisation générale

## 2.3. Raffinement des cas d'utilisation

Dans la partie suivante, nous avons détaillé les cas d'utilisation générale en introduisant des cas d'utilisations internes.

### 2.3.1. Raffinement de cas d'utilisation « S'authentifier »

Afin que l'utilisateur puisse se connecter au système, il doit saisir correctement son login et son mot de passe. La figure illustrée ci-après présente le cas d'utilisation « S'authentifier ».



**Figure 10: Diagramme de cas d'utilisation « S'authentifier »**

Le tableau ci-dessous décrit le scénario d'authentification par un utilisateur de la FDJ.

Description textuelle relatif au cas d'utilisation « S'authentifier »
<p><b>Titre :</b> « S'authentifier »</p> <p><b>Acteur :</b> Utilisateur FDJ</p> <p><b>But :</b> L'utilisateur doit se connecter pour accéder au site.</p> <p><b>Pré-condition :</b> L'utilisateur doit être créé dans la base de données et connaître ses identifiants.</p> <p><b>Scénario nominal :</b></p> <ol style="list-style-type: none"> <li>1. Le système affiche le formulaire d'identification.</li> <li>2. L'internaute remplit le formulaire avec l'ensemble des informations nécessaires à son identification.</li> <li>3. Le système vérifie les informations saisies par l'utilisateur et renvoie vers la page d'accueil.</li> </ol> <p><b>Scénario d'exception :</b></p> <ul style="list-style-type: none"> <li>- En cas d'échec, le système renvoie un message d'erreur et signale à l'utilisateur de recommencer.</li> </ul>

**Tableau 1: Description relatif au cas d'utilisation « S'authentifier »**

### 2.3.2. Raffinement de cas d'utilisation « Scrutation et contrôle des sources »

La figure suivante détaille les sous cas d'utilisation internes relatif au cas général « Scrutation et contrôle des sources ».

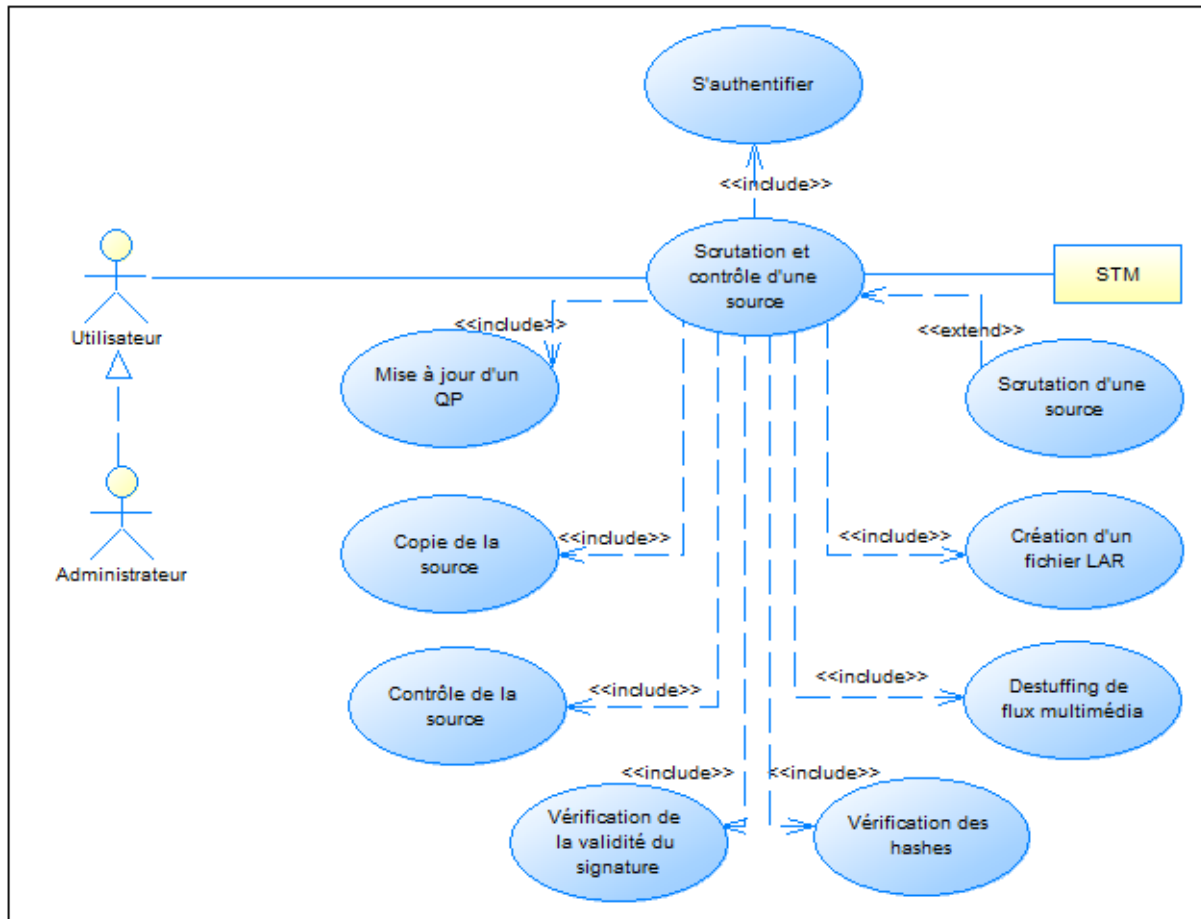


Figure 11: Diagramme de cas d'utilisation « Scrutation d'une source »

Le tableau illustré par la figure suivante décrit le sous cas d'utilisation « Contrôle du certificat ».

Description textuelle relatif au sous cas d'utilisation « Contrôle du certificat »
<p><b>Titre :</b> « Contrôle du certificat »</p> <p><b>Acteur :</b> Utilisateur FDJ</p> <p><b>Objectif :</b> Contrôle des propriétés qui constituent le certificat.</p> <p><b>Pré-condition :</b> Le certificat est chargé et en attente de vérification de ses propriétés.</p> <p><b>Scénario nominal :</b></p> <ol style="list-style-type: none"> <li>1. Chargement du certificat.</li> <li>2. Contrôle de la date de validité.</li> <li>3. Contrôle de la chaine de certification.</li> <li>4. Vérification de la signature <math>F_{\text{sign}}</math> à l'aide de la clé publique contenue dans <math>C_{\text{part}}</math>.</li> </ol> <p><b>Scénario d'exception :</b></p> <ul style="list-style-type: none"> <li>- En cas d'échec, une erreur est tracée et le traitement de cette source s'arrête jusqu'au prochain horaire de scrutation. Un e-mail d'avertissement est envoyé aux adresses définies dans un fichier de configuration.</li> </ul>

**Tableau 2: Description relatif au sous cas d'utilisation « Contrôle du certificat »**

### 2.3.3. Raffinement de cas d'utilisation « Actualisation d'un QP»

Avant de diffuser un QP, le STM doit vérifier le contenu, le mettre à jour et l'empaqueter. La figure suivante illustre toutes les fonctionnalités internes afin d'actualiser un QP.

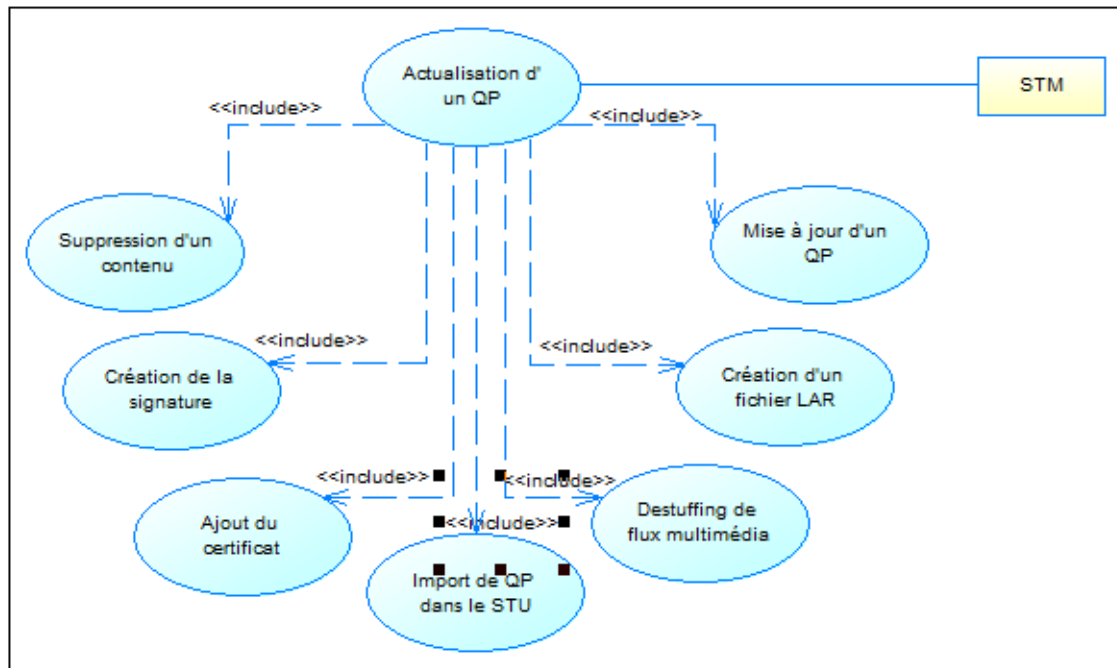


Figure 12: Diagramme de cas d'utilisation « Actualisation d'un QP »



Le tableau 3 décrit le scénario de mise à jour d'un QP réalisé par le STM.

Description textuelle relatif au cas d'utilisation « Mise à jour d'un QP »
<p><b>Titre :</b> « Mise à jour d'un QP »</p> <p><b>Acteur :</b> STM</p> <p><b>Objectif :</b> Vérification de la validité des fichiers de sécurité.</p> <p><b>Pré-condition :</b> Le contenu n'est pas à jour.</p> <p><b>Scénario nominal :</b></p> <ol style="list-style-type: none"> <li>1. Suppression des contenus expirés.</li> <li>2. Mise à jour du fichier de validité.</li> <li>3. Mise à jour de la référence.</li> <li>4. Incrémentation de la version.</li> <li>5. Réécriture de la référence.</li> <li>6. Empaquetage.</li> <li>7. Import de la version dans le STU.</li> <li>8. Purge de la version N-1 de QP.</li> <li>9. Copie de la version N vers N-1.</li> </ol> <p><b>Scénario d'exception :</b></p> <ul style="list-style-type: none"> <li>- En cas d'échec, une erreur est tracée dans les fichiers logs.</li> </ul>

**Tableau 3: Description relatif au sous cas d'utilisation « Mise à jour d'un QP »**

### 2.3.4. Raffinement de cas d'utilisation « Gestion de diffusion »

Ci-dessous nous avons détaillé le sous cas d'utilisation « Diffusion d'un QP » qui présente le processus de diffusion d'un QP.

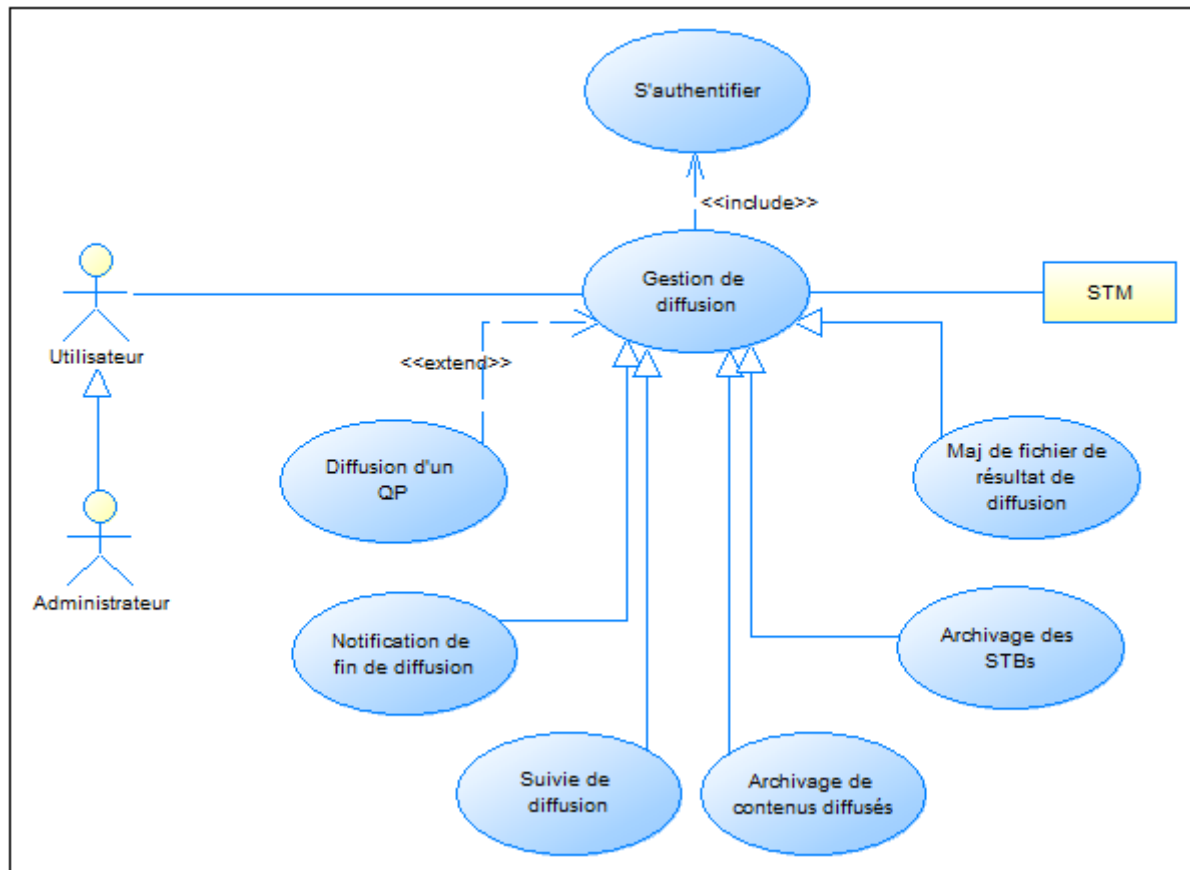


Figure 13: Diagramme de cas d'utilisation « Gestion de diffusion »

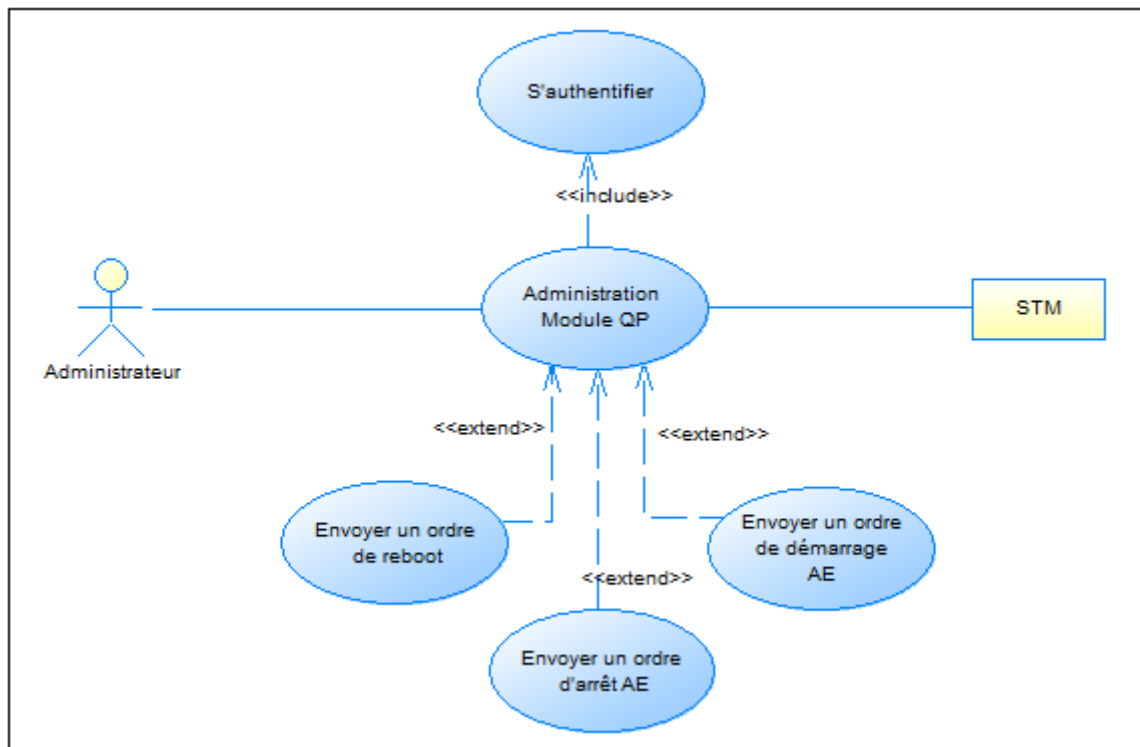
Le tableau 4 décrit le sous cas d'utilisation « Diffusion d'un QP » effectué par l'utilisateur FDJ.

Description textuelle relatif au sous cas d'utilisation « Diffusion d'un QP »
<p><b>Titre :</b> « Diffusion d'un QP »</p> <p><b>Acteur :</b> STM</p> <p><b>But :</b> La version courante de QP doit être diffusée à un ou plusieurs STB.</p> <p><b>Pré-condition :</b> Le QP est mis à jour et en attente de diffusion par le STM.</p> <p><b>Scénario nominal :</b></p> <ol style="list-style-type: none"> <li>1. Création de la version.</li> <li>2. Enregistrement de la version dans la base de données.</li> <li>3. Création du fichier en local.</li> <li>4. Initialisation de la planification.</li> <li>5. Calcul du débit et de l'horaire de diffusion.</li> <li>6. Enregistrement de la diffusion en base de données.</li> <li>7. Diffusion du QP.</li> </ol> <p><b>Scénario d'exception :</b></p> <ul style="list-style-type: none"> <li>- En cas d'échec, une erreur est tracée dans le fichier de résultat de fin de diffusion.</li> </ul>

**Tableau 4: Description relatif au sous cas d'utilisation « Diffusion d'un QP »**

### **2.3.5. Raffinement de cas d'utilisation « Administration du module QP »**

Le diagramme de cas d'utilisation ci-dessous illustre les besoins fonctionnels relatifs au cas d'utilisation « Administration module QP ». Cette fonctionnalité est accessible au profil administrateur de puis le menu Multicast de l'application.



**Figure 14: Diagramme de cas d'utilisation « Administration du module QP »**

Suite à ce diagramme, nous détaillons le sous cas d'utilisation « Envoie ordre de démarrage d'AE ».

Description textuelle relatif au cas d'utilisation « Envoie ordre de démarrage d'AE »
<p><b>Titre :</b> « Envoie ordre de démarrage d'AE »</p> <p><b>Acteur :</b> Administrateur STM</p> <p><b>But :</b> Demande d'émission de fichier de démarrage d'AE.</p> <p><b>Pré-condition :</b> L'administrateur peut accéder au sous menu « création Multicast » afin de</p> <p><b>Scénario nominal :</b></p> <ol style="list-style-type: none"> <li>1. Création des paramètres d'émission.</li> <li>2. Création du job.</li> <li>3. Récupération du job par MSender.</li> <li>4. Emission du job à MSender : un fichier ordre est envoyé dans les terminaux cibles et l'interprétation de ce fichier entraine l'affichage l'AE courante à réception.</li> <li>5. Récupération du job par MContrôler et MJobViewer.</li> <li>6. Sauvegarde du job en base de données.</li> <li>7. Suppression du job.</li> </ol> <p><b>Scénario d'exception :</b></p> <ul style="list-style-type: none"> <li>- En cas d'echec de récupération ou d'émission du job par l'application MSender, une erreur est tracée dans les fichiers de log.</li> </ul>

**Tableau 5: Description relatif au sous cas d'utilisation « Envoie ordre de démarrage d'AE »**

## Conclusion

Dans ce chapitre nous nous sommes intéressés à l'analyse des besoins fonctionnels et non fonctionnels de notre STM et ceci afin d'avoir une vision plus claire du sujet abordé et bien comprendre les différentes tâches à réaliser. Ce chapitre nous permettra d'aborder aisément le chapitre suivant : Architecture et conception des évolutions du projet STM.

## **Chapitre 3 : Architecture et conception**

### **Introduction**

Une phase de conception a pour objectif de permettre et de formaliser les étapes préliminaires du développement d'un système afin de rendre ce dernier plus fidèle aux besoins du client. La phase de conception permet de décrire de manière non ambiguë le fonctionnement désiré du système afin de faciliter la réalisation et la maintenance.

Pour ce faire, nous présenterons dans une première partie une conception globale de notre projet ensuite dans la deuxième partie nous détaillerons la conception. Nous achèverons ceci par des diagrammes d'activités (Voir Annexe A) présentant quelques scénarios de travail demandé.

### **3.1. Méthodologie et formalisme adopté**

Un modèle de cycle de vie d'un logiciel représente les activités ou les phases importantes d'un projet informatique pour aboutir à un produit logiciel prêt à être commercialisé.

Pour achever la réalisation de l'évolution de notre projet, le modèle en V a été notre choix méthodologique pour le processus de développement. C'est un modèle conceptuel de gestion de projet imaginé suite aux problèmes de réactivité du modèle en cascade, il demeure actuellement le cycle de vie le plus connu et certainement le plus utilisé.

La figure qui suit illustre ses différentes phases.

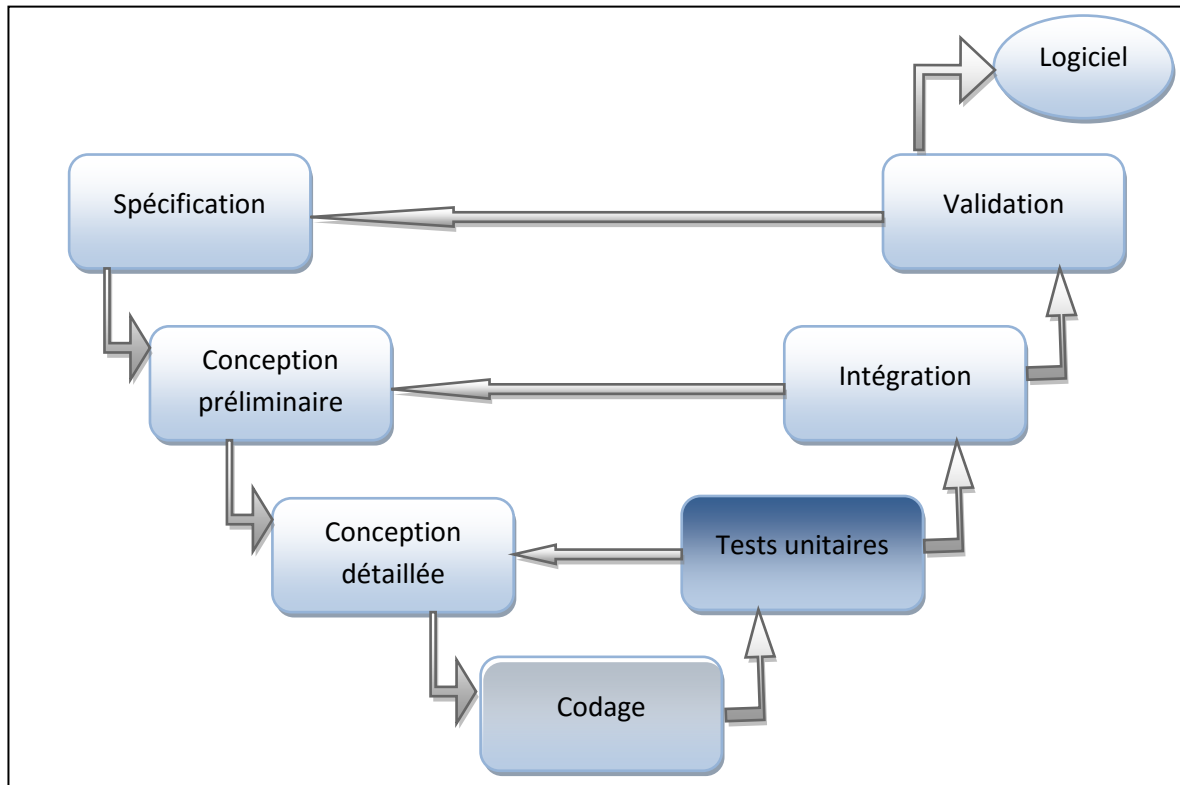


Figure 15: Cycle de vie: Modèle en V

## 3.2. Conception générale

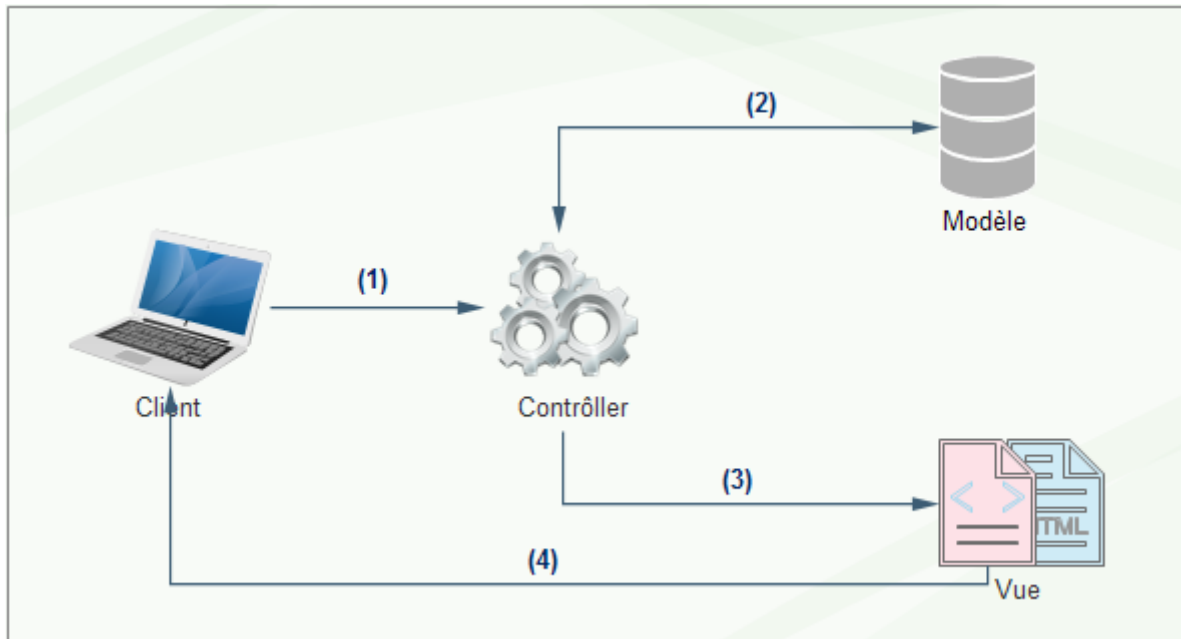
### 3.2.1 Modèle MVC

MVC est un patron de conception très répandu pour réaliser des sites web. Ce patron de conception est une solution éprouvée et reconnue permettant de séparer l’affichage des informations, les actions de l’utilisateur et l’accès aux données.

Le projet STM suit le motif de conception logicielle MVC qui sépare notre application en trois couches principales :

- **Modèle** : cette partie gère les *données* de votre site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur.
- **Contrôleur** : cette partie gère la logique du code qui prend des décisions. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue.
- **Vue** : cette partie se concentre sur l’affichage. Elle se contente de récupérer des variables pour savoir ce qu'elle doit afficher. [7]

Le schéma suivant montre les différentes couches de ce modèle et la façon dont ces dernières interagissent entre elles.



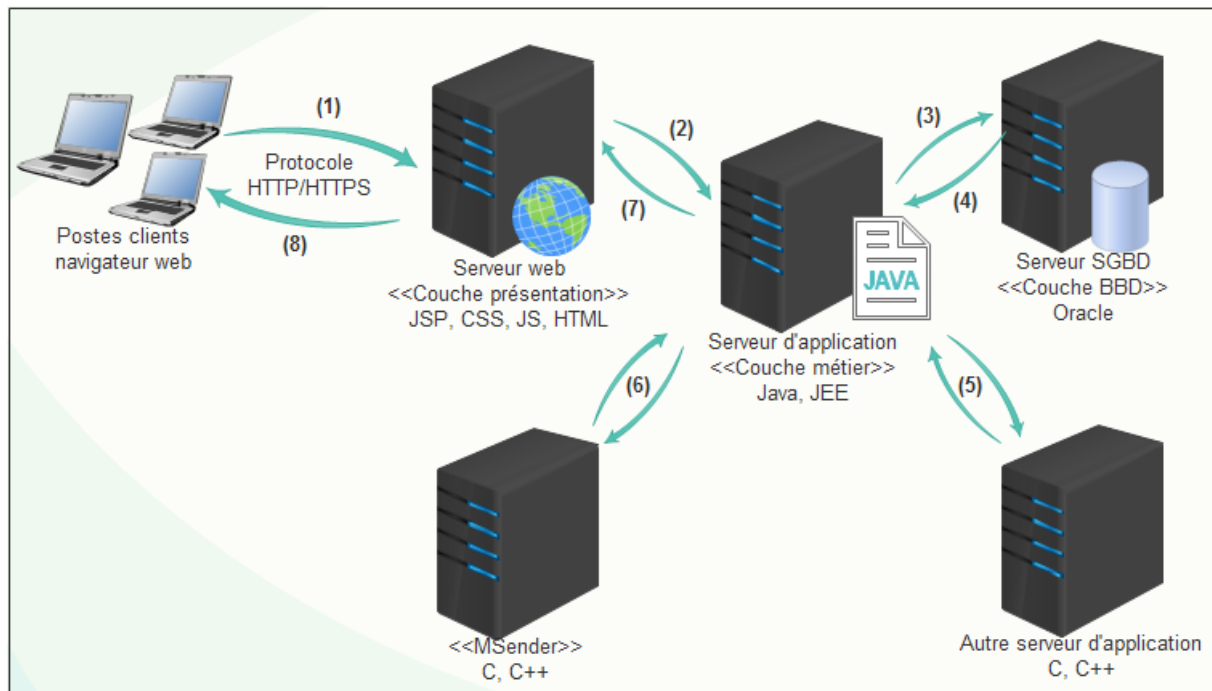
**Figure 16: Modèle MVC**

1. Le client demande une page HTML depuis son navigateur.
2. Le contrôleur consulte et met à jour les données.
3. Le contrôleur décide de la page à afficher.
4. La vue retourne la page HTML demandée.

### 3.2.2. Architecture n-tiers

La réalisation de travail demandé adopte une architecture n-tiers constituée d'une partie client, une partie serveur de base de données, une partie serveur d'applications et une partie serveur web schématisée par la figure citée ci-dessous.





**Figure 17: Architecture n-tiers**

1. Le client envoie une requête HTTP/HTTPS au serveur web.
2. Le serveur web transmet la requête au serveur d'applications.
3. Le serveur d'applications demande les données stockées dans la base de données et traite la requête en manipulant les données stockées dans la base de données.
4. Le serveur d'application traite les requêtes en manipulant les données reçues du serveur de base de données.
5. Le serveur d'application interagit avec d'autres applications externes comme le serveur Mcontrôler, MJobViewer, etc.
6. Le serveur d'application soumet à MSender les planifications qui les diffuse vers l'ensemble des STBs.
7. Le serveur d'applications envoie au serveur Web le résultat de la requête.
8. Le serveur web transmet le résultat au client Web, qui peut enfin l'afficher.

### 3.3. Conception détaillée

Après avoir présenté l'architecture opérationnelle et applicative de l'outil à réaliser, nous élaborons, dans cette partie, la conception détaillée de travail demandé.

### 3.3.1. Aspect statique

La vue statique de système permet d'identifier les propriétés des objets et leurs liaisons avec les autres objets. Nous présentons dans un premier temps le diagramme de paquetage qui illustre les différents entités et les relations entre eux.

Ensuite, nous détaillons le paquetage de planification de diffusion avec un diagramme des classes.

#### 3.3.1.1. Diagramme de paquetage

Un *diagramme de packages* est un diagramme UML qui fournit une représentation graphique de haut niveau de l'organisation de votre application, et nous aide à identifier les liens de généralisation et de dépendance entre les packages. [8]

La figure ci-dessous illustre les paquetages que nous avons utilisés lors de l'élaboration de notre application.

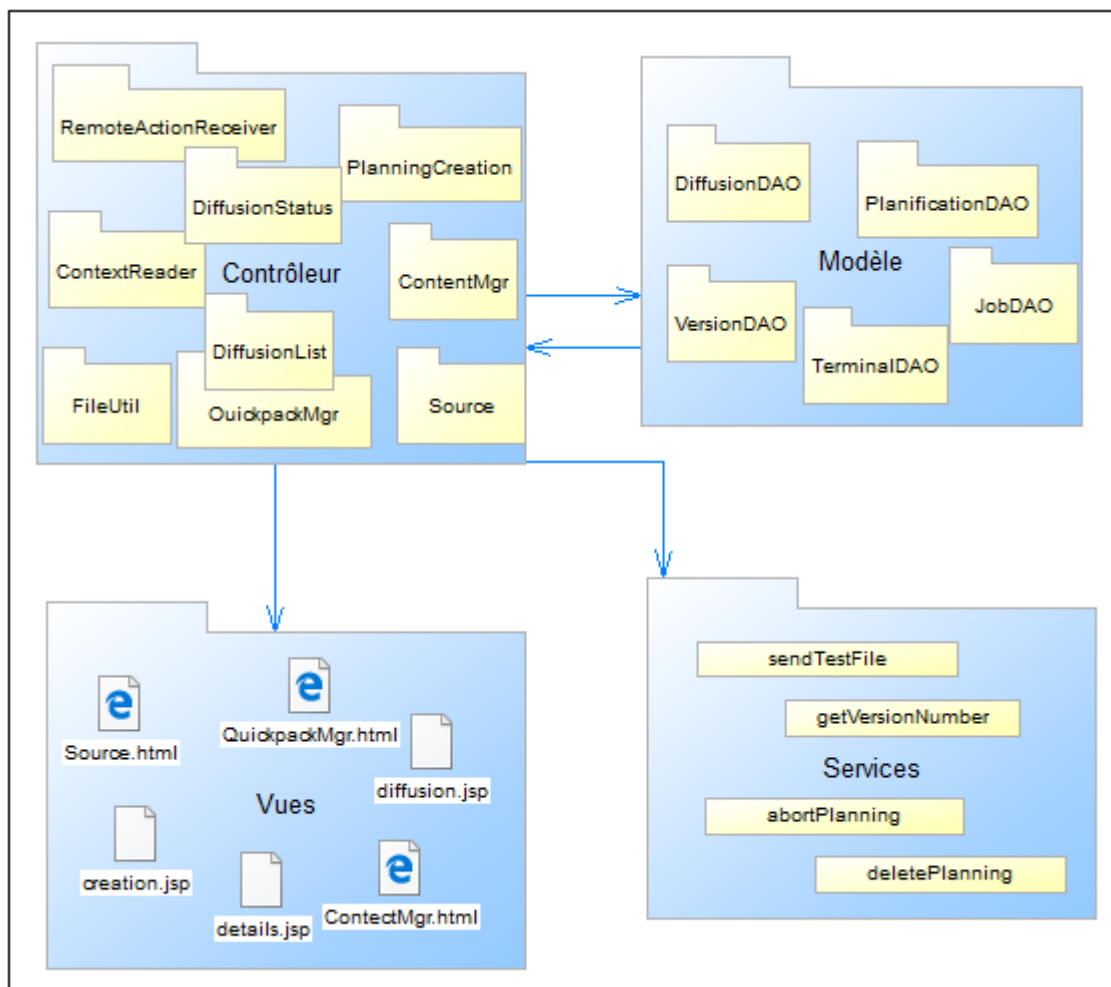


Figure 18: Diagramme de paquetage

Les différents paquetages de notre application sont les suivants :

- Le paquetage des vues contient les fichiers d'interfaces qui englobent les différents composants graphiques.
- Le paquetage des contrôleurs permettant l'acheminement des requêtes client et la communication entre les vues et les modèles.
- Le paquetage des modèles comporte les modèle de données que nous détaillerons par la suite dans le diagramme des classes.
- Le paquetage des services contenant les services web, parmi ces services nous avons présenté :
  - sendTestFile : Demande d'envoi d'un fichier de test ou de reboot.
  - getVersionNumber : Demande d'un indice de version QP.
  - abortPlanning: Demande d'annulation d'une planification Multicast.
  - deletePlanning : Demande de suppression d'une planification Multicast.

### **3.3.1.1. Diagrammes des classes : Paquetage de diffusion MCAST**

Le *diagramme des classes* est considéré comme le plus important de la modélisation orientée objet. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système.

Dans ce qui suit, nous présentons le diagramme de classe illustré par la figure ci-dessous qui rassemble l'ensemble de nos entités.

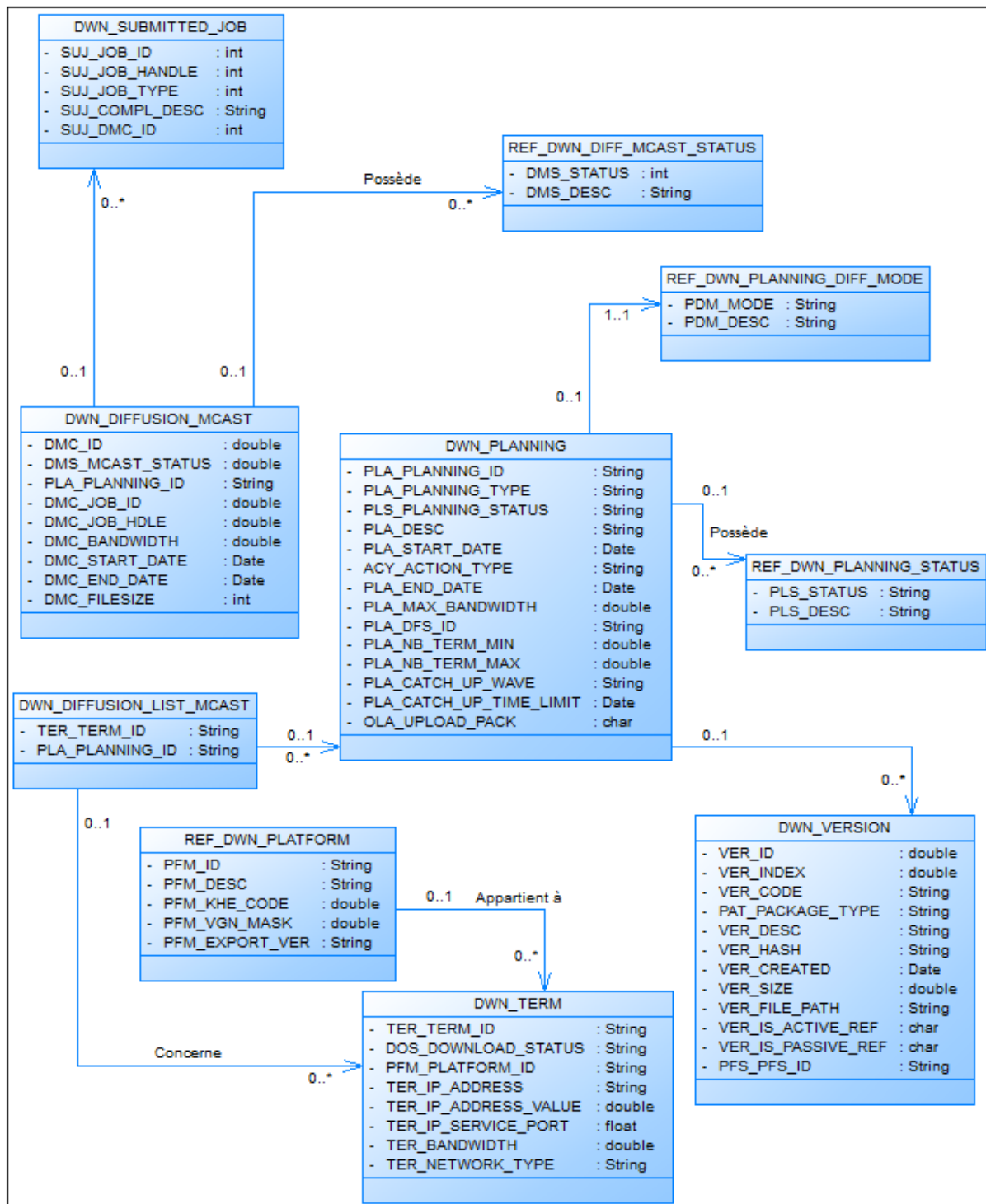


Figure 19: Diagramme des classe

Parmi ces classes nous détaillons :

- La classe DWN\_PLANNING modélise la table STU\_PLA.DWN\_PLANNING qui permet la gestion d'une planification en BDD, chaque enregistrement de la table correspond à une planification de diffusion.
- Chaque enregistrement de la table est associé à une planification.
- Chaque planification créée par l'utilisateur soumet à MSender qui la diffuse vers l'ensembles des STBs.
- Une planification de diffusion d'un QP contient les attributs suivants :
  - Mode : incrémentale ou total.
  - Type : QP
  - Version : le numéro de version de QP.
- La classe DWN\_PLATFORM\_LIST représente une référence vers la liste des plateformes à laquelle appartient les terminaux qui sont concernés par la réception des données.
- La classe REF\_DWN\_PLANNING\_STATUS correspond à la table référence vers les status d'une planification de diffusion. Une planification peut avoir les status suivants : CANCELLED, SCHEDULED, INTERRUPTED, MISSED, STANDBY, IN\_PROGRESS, TERMINATED.

### 3.3.2. Aspect dynamique

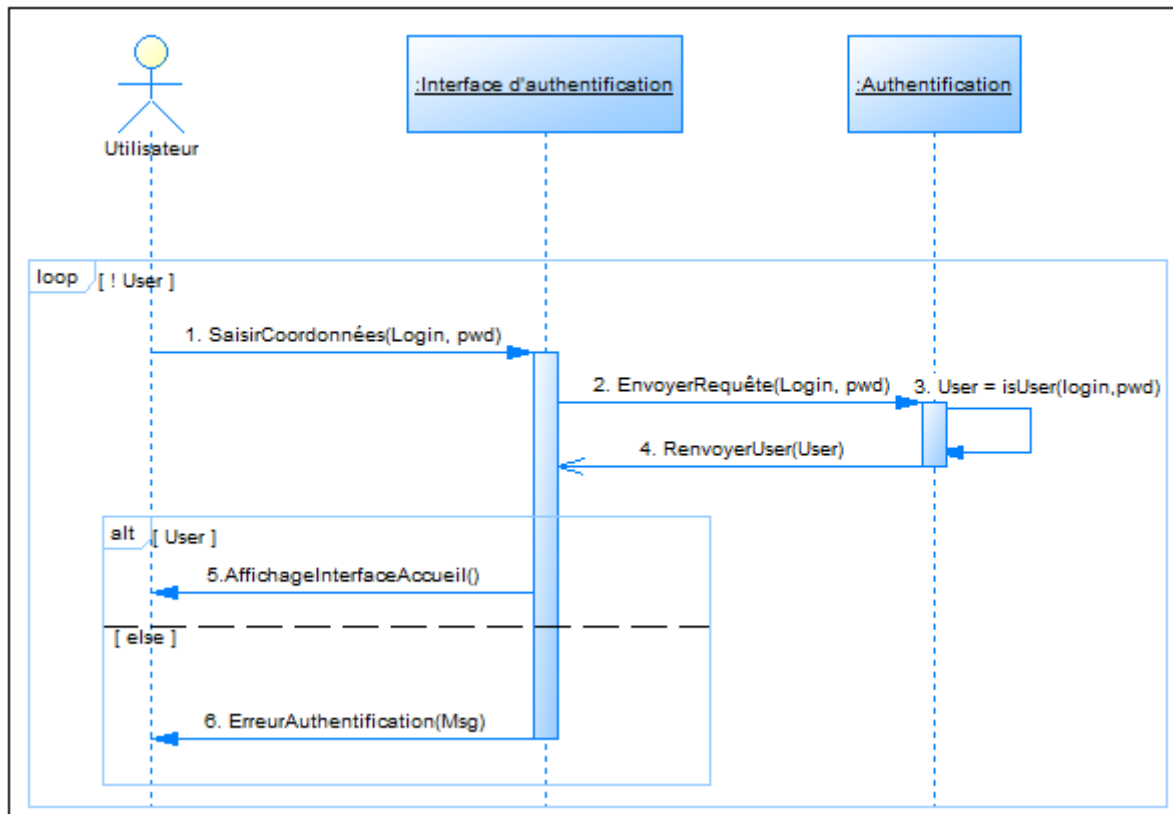
Dans cette section nous présentons quelques scénarios associés à notre système. Nous recourons aux diagrammes de séquences pour montrer l'aspect dynamique de l'outil à réaliser.

Ce type de diagramme permet de montrer les interactions entre l'utilisateur et le système selon un ordre chronologique dans la formulation UML.

#### 3.3.2.1. Diagramme de séquence « S'authentifier »

Lorsque l'utilisateur demande l'accès à l'application, il doit tout d'abord s'identifier par son login et son mot de passe via le serveur d'application qui prend en charge de vérifier et consulter la base de données.

S'il est accepté, donc il y'aura l'accès au système et aux applications du menu correspondant. Sinon, le serveur d'application lui affiche un message d'erreur afin de rectifier ses données. La figure ci-dessous illustre son scénario d'authentification.



**Figure 20: Diagramme de séquence « S'authentifier »**

### 3.3.2.3. Diagramme de séquence « Envoie ordre de reboot »

Cette opération s'effectue hors session de jeu par l'administrateur, synchronisé avec l'opérateur STM. Il est possible d'envoyer en Multicast un fichier de reboot vers les équipements qui appartiennent aux plateformes NSTBU, NSTBM et STB depuis la page de planification d'une création Multicast. L'adresse Multicast et les noms des fichiers sont définis dans le fichier de paramètres du STM.

La figure ci-après montre le déroulement de processus d'envoi d'un fichier de reboot vers une partie des STBs.

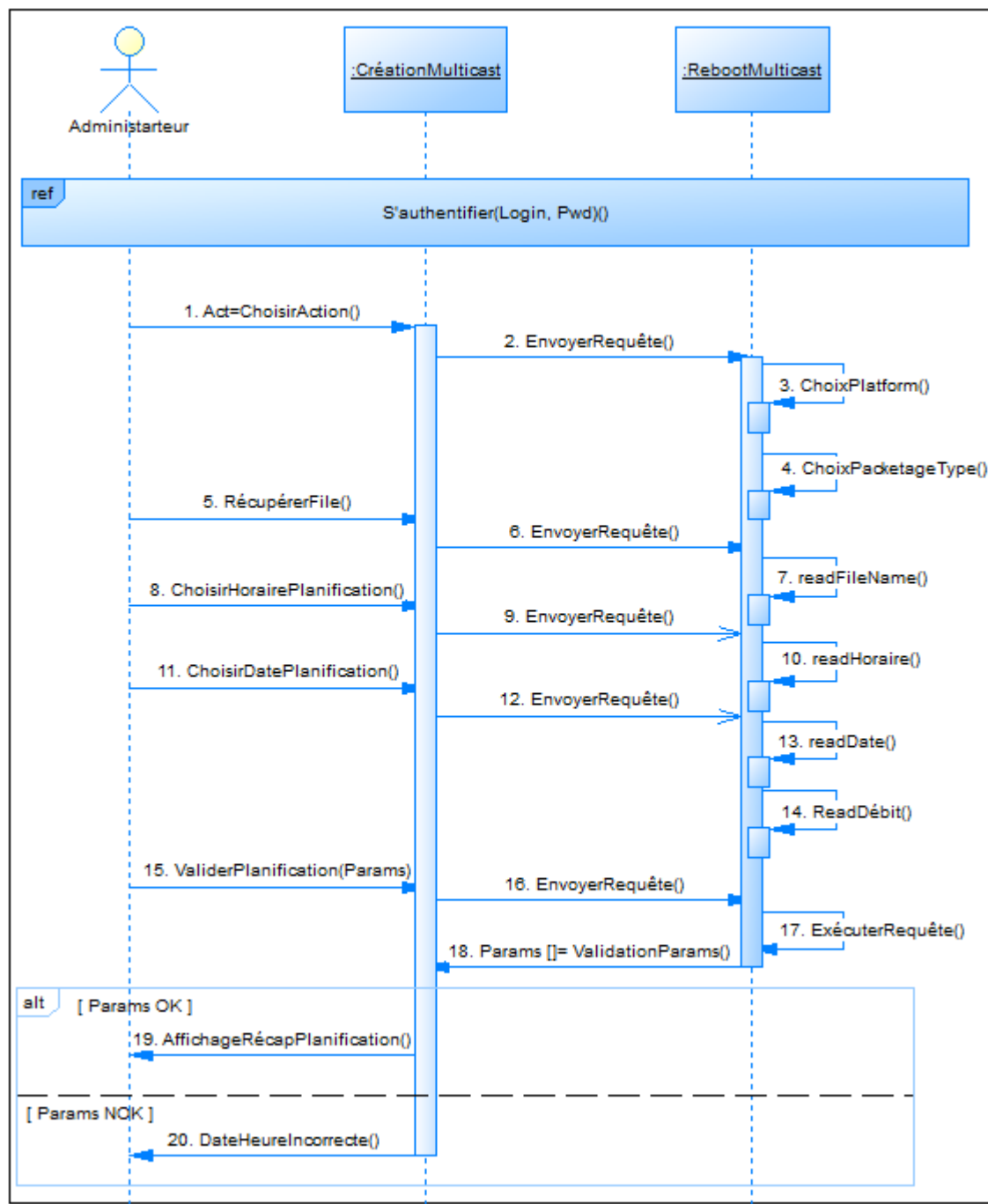


Figure 21: Diagramme de séquence « ordre de reboot »

### 3.3.2.4. Diagramme de séquence « Scrutation manuelle d'une source »

La figure ci-après illustre le déroulement du scan d'une source manuellement. Elle sera effectué s'il n'y a pas une scrutation automatique prévue dans les 10 minutes qui suivent.

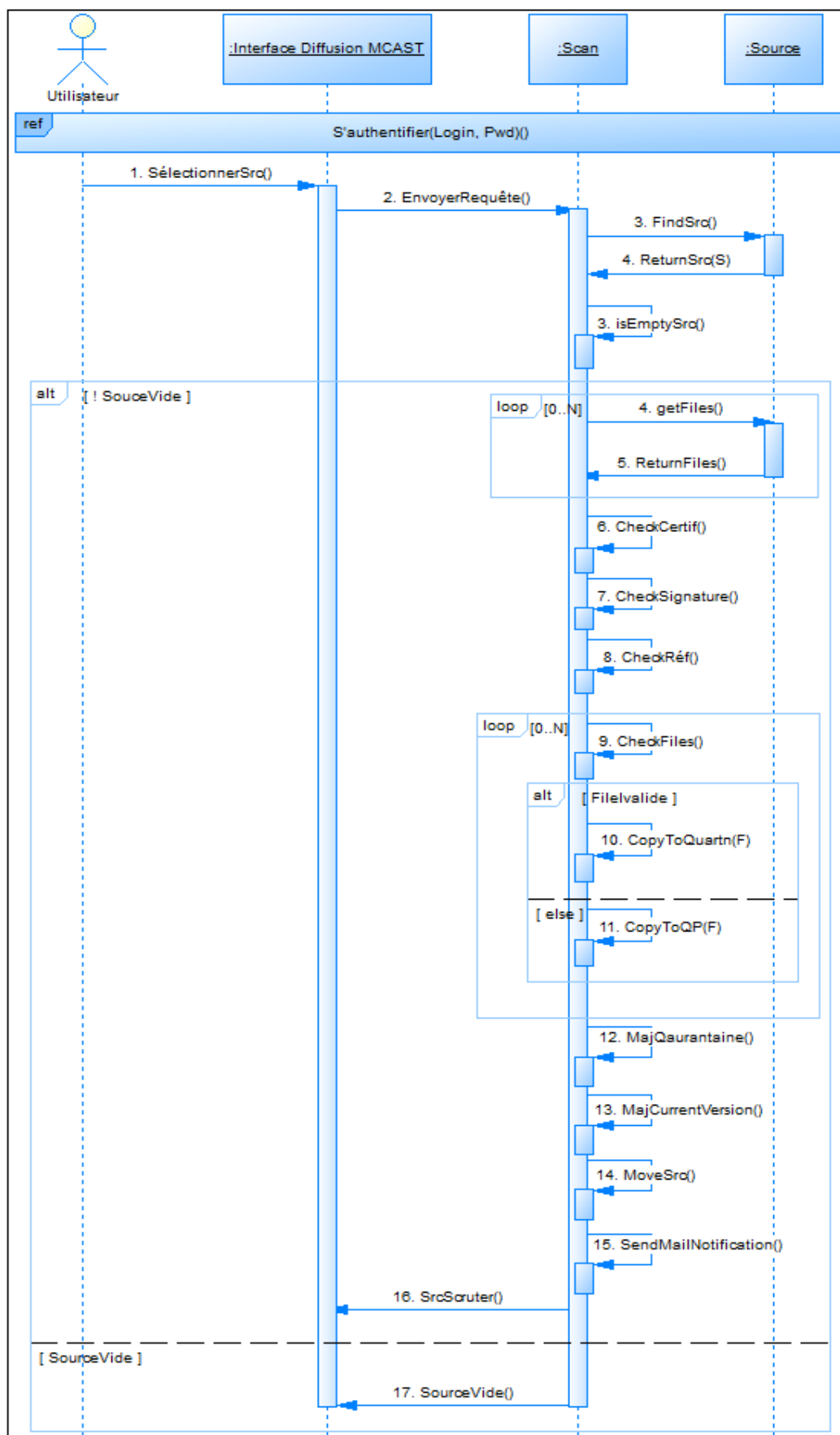


Figure 22: Diagramme de séquence « Scrutation manuelle d'une source »



### 3.3.2.5. Diagramme de séquence de « Import d'un QP dans le STU »

Il est indispensable d'importer le QP dans la base de données de STU afin de le diffuser en unicast en cas de non réception par une partie de l'ensemble des STBs. Le diagramme suivant détaille le processus de cette fonctionnalité.

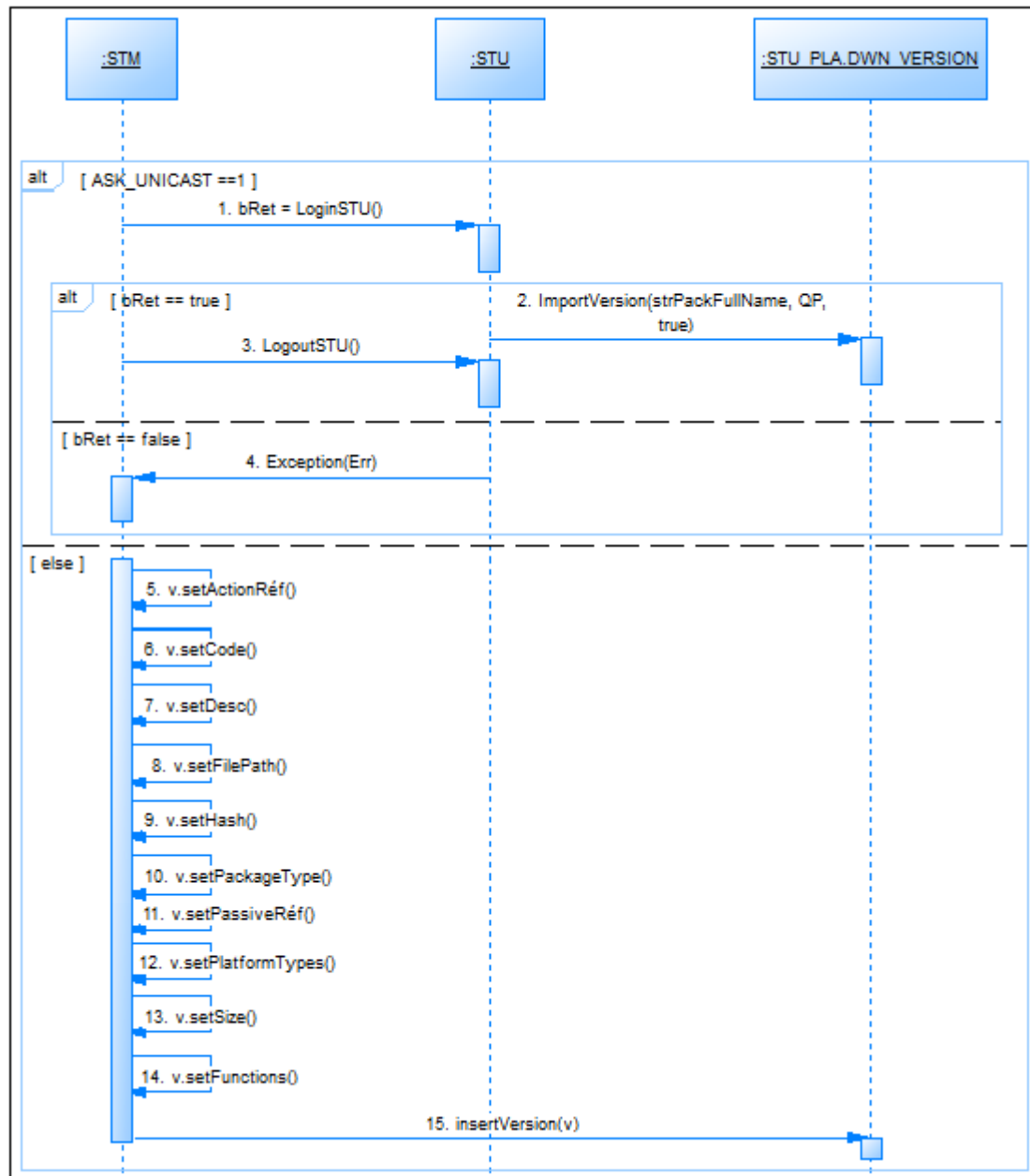


Figure 23: Diagramme de séquence « Import d'un QP dans le STU »

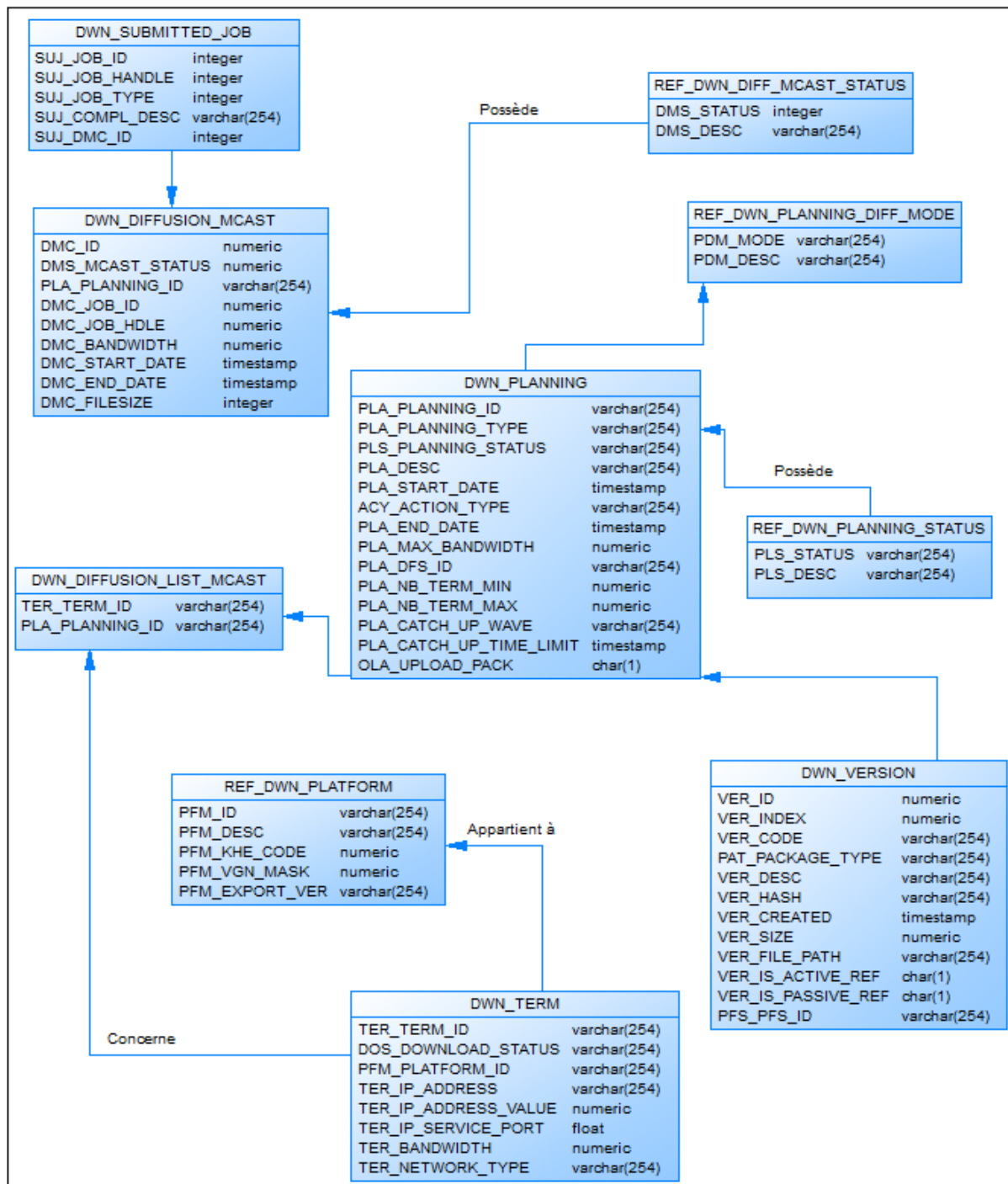
### **3.4. Structure de la base de données**

En s'appuyant sur le diagramme des classes, et pour préciser les tables nécessaires pour nos modules, nous devons implémenter le modèle physique de données qui correspond au packaging planification d'une diffusion MCAST.

#### **3.4.1. Modèle physique de données**

Le modèle physique des données (MPD) d'une base de données permet d'avoir une représentation graphique de la structure d'une base de données et ainsi de mieux comprendre les relations entre les différents tables. Il permet d'avoir un point de vue global sur l'ensemble de la base de données.

Le modèle physique de packaging de diffusion MCAST est illustré dans la figure qui suit.



**Figure 24: Modèle physique de données de paquetage de diffusion MCAST**

## Conclusion

Dans ce chapitre, nous avons mis en évidence l'architecture opérationnelle et applicative de notre système. Par la suite, nous avons traité la partie conceptuelle afin de délimiter les grandes lignes de notre application. Nous abordons dans le chapitre qui suit la réalisation et les tests de l'application STM.

## Chapitre 4 : Réalisation et test

### Introduction

L'interface a pour rôle d'établir la communication entre l'homme et la machine, et donc d'assurer la correspondance des concepts propres à chaque partenaire, en s'appuyant sur un ensemble de représentations qui doivent avoir un double aspect : formel pour la machine et perceptible pour l'homme.

C'est pour cette raison là que l'interface doit être conviviale, et doit également répliquer à plusieurs critères ergonomiques pour les utilisateurs.

Dans ce chapitre, nous allons présenter les outils techniques utilisés pour la réalisation de notre projet. Par la suite nous validons ce qu'on a déjà conçu dans le chapitre précédent.

### 4.1. Environnement de travail

Les environnements de travail informatique englobent désormais des technologies diverses et variées adaptées à nos besoins.

Afin de pouvoir répondre aux objectifs fixés au départ, on va exprimer dans cette partie l'environnement de réalisation qui consiste à présenter le matériel et les logiciels de base utilisés pour le développement de projet.

#### 4.1.1. Environnement matériel









Afin de réaliser ce travail, nous avons utilisé trois machines DELL qui possèdent les caractéristiques présentées par le tableau 21.

	Processeur	Ram	D.D.	S.E.
<b>Machine 1 : Développement</b>	Intel ® Pentium ® D CPU 3 Ghz 2.99 Ghz	4 GO	1 TO	Win7 64 bits
<b>Machine 2 : Base de données</b>	Intel Core i5-4200U	6 GO	1 TO	Win7 64 bits
<b>Machine 3 : Machine de test</b>	Intel Core i5-4200U	4 GO	1 TO	Debian Wheezy 7

**Figure 25: Description de l'environnement matériel**

### 4.1.2. Environnement logiciel

Voici la liste des logiciels employés tout au long le développement de notre projet.

Logo	Présentation
	<b>Eclipse</b> est l'Environnement de Développement Intégré (ou IDE) le plus largement utilisé pour la programmation JEE, très performant, il est de plus gratuit est open source. [9]
	<b>Tomcat</b> est l'un des moteurs de servlets et JSP Java EE les plus utilisés. Il est paramétrable par des fichiers XML et de propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP.  Tomcat a été écrit en langage Java. Il peut donc s'exécuter via la machine virtuelle Java sur n'importe quel système d'exploitation la supportant. [10]
	<b>Cygwin</b> est une collection de logiciels libres à l'origine développés par Cygnus Solutions permettant à différentes versions de Windows de Microsoft d'émuler un système Unix. Il est même possible de lancer une interface graphique comme surcouche de Cygwin, par exemple KDE ou GNOME. [11]
	<b>Oracle SQL Developer</b> est un environnement de développement intégré (EDI) multi-plateforme, fourni gratuitement par Oracle Corporation et utilisant la technologie Java (Java Development Kit).  C'est un outil graphique permettant d'interroger des bases de données Oracle à l'aide du langage SQL. [12]
	<b>VirtualBox</b> est un logiciel libre de virtualisation publié par Oracle. [13]
	<b>GNU/Linux Debian</b> est également un système d'exploitation multi-plateformes. Elle supporte douze architectures et ses milliers de paquets couvrent presque tous les matériels existants et des domaines d'application imaginables. [14]
	<b>PowerAMC</b> est un logiciel de conception qui permet de réaliser tous les types de modèles informatiques et leurs bases de données associées. [15]
	<b>EdrawMax</b> un outil de développement UML. Il présente une interface ergonomique facile à manipuler qui donne la possibilité de générer tous les types de modèles informatiques. [16]

**Figure 26: Description de l'environnement logiciel**

### 4.1.3. Outils de programmation

Dans cette partie, nous avons représenté les différentes technologies utilisées dans la phase de développement.

**Le langage UML : UML (Unified Modeling Language)** est un langage de modélisation graphique. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique. UML est l'accomplissement de la fusion de précédents langages de modélisation objet: Booch, OMT, OOSE. [17]

**Le langage Java : JAVA** est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Ses caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d'être très largement utilisé pour le développement d'applications de types très disparates. Java est notamment largement utilisée pour le développement d'applications d'entreprises et mobiles. [18]

**Le langage XML : (Extensible Markup Language)** est un langage informatique de balisage générique qui dérive du SGML. Cette syntaxe est dite « extensible » car elle permet de définir différents espaces de noms. [19]

**Le langage JSON : (JavaScript Object Notation)** est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée comme le permet XML par exemple. [20]

**JSP : (Java Server Pages)** est un standard permettant de développer des applications Web interactives, dont le contenu est dynamique. [21]

**Le langage SQL : (Structured Query Language)** est un langage informatique normalisé servant à exploiter des bases de données relationnelles. [22]

## 4.2. Description des aperçus de l'application STM

### 4.2.1. Interface d'authentification

Cette première capture présente l'interface d'authentification dans laquelle on doit choisir le nom d'utilisateur et entrer le mot de passe pour commencer à utiliser le STM. Cette étape met en valeur l'aspect sécurité : nous vérifions la disponibilité du compte utilisateur et nous lui attribuons les droits et privilèges nécessaires.



**Site utilisateur**

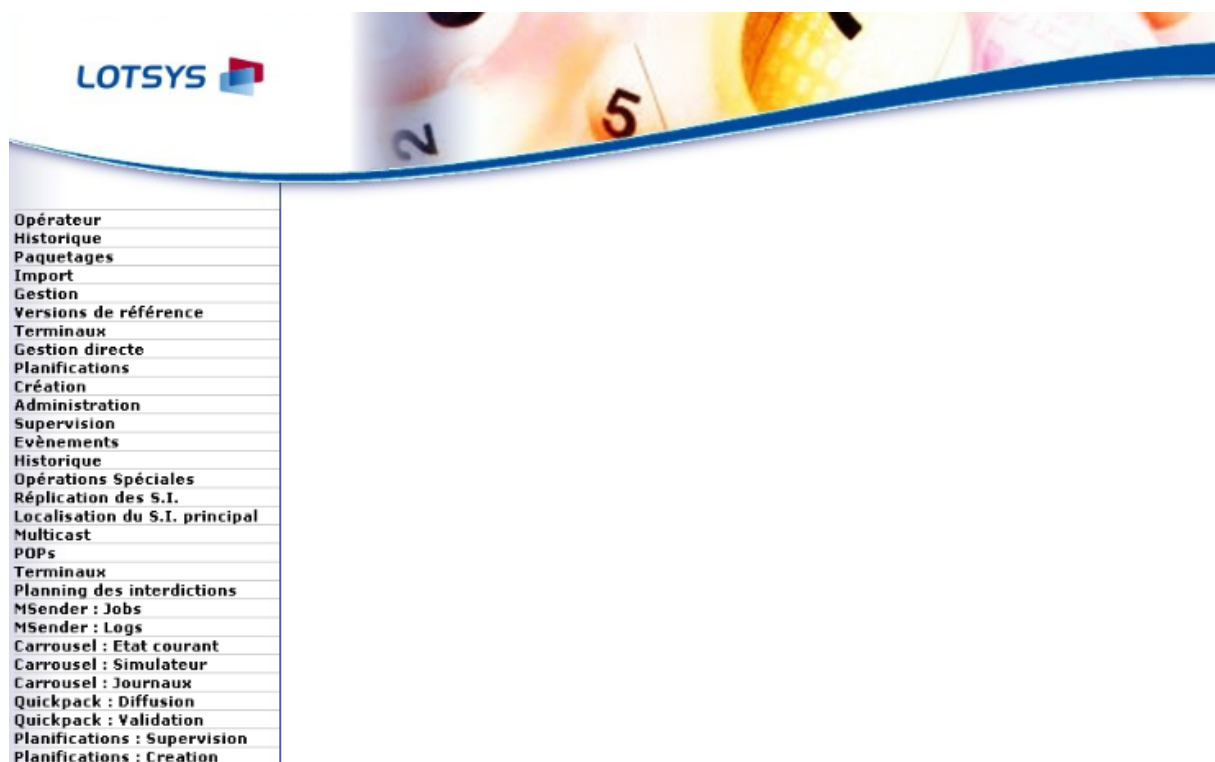
Compte

Mot de passe

**Figure 27: Interface d'authentification**

#### 4.2.2. Interface principale

La figure ci-après décrit le menu principale de l'application logicielle STM. Cette interface comporte le sous menu Administration, elle est utilisable par l'administrateur FDJ.



**Figure 28: Interface principale du STM**

Le menu ci-dessus gère plusieurs éléments tels que les PDV, les paquetages, les terminaux, etc. Nous avons touché aux menus suivants :



- Paquetages.
- Versions de références.
- Administration.
- QP : Diffusion.
- QP : Validation.
- Planifications : Création.

### 4.2.3. Interface de scrutation d'un QP

La figure ci-dessous montre l'interface de scrutation d'un QP ainsi que la diffusion de ce dernier. En pemier lieu, l'utilisateur sélectionne la source ( source locale ou SSH), la scruté et diffuse le contenu si c'est nécessaire selon un mode de diffusion choisie.



**Figure 29: Interface de diffusion et scrutation d'un QP**

### 4.2.4. Interface de gestion des actions

Cette IHM présente l'ensemble des actions qui peuvent être touché les terminaux appartiennent aux différentes plateformes. L'utilisateur remplit les propriétés d'une planification de diffusion et valide cette dernière. Si les propriétés sont corrêcte, les détails de la planification sont affichés dans une nouvelle IHM, sinon un message d'erreur est apparus.

**Définition des conditions**

Action	Plateformes	Type Paquetages	Fonctions
<input type="checkbox"/> Téléchargement dans l'actif	<input type="checkbox"/> Galaté	<input type="checkbox"/> Multimédia	<input checked="" type="checkbox"/> Décodeur Caisse
<input type="checkbox"/> Téléchargement dans le passif	<input type="checkbox"/> Néréide	<input checked="" type="checkbox"/> Quickpack	<input type="checkbox"/> Décodeur Caisse Vertical
<input type="checkbox"/> Téléchargement dans l'actif et copie dans le passif	<input checked="" type="checkbox"/> NSTBM	<input type="checkbox"/> Software	<input checked="" type="checkbox"/> Décodeur Jeux
<input type="checkbox"/> Ordre de redémarrage (sans téléchargement)	<input checked="" type="checkbox"/> NSTBU		<input checked="" type="checkbox"/> Décodeur Vitrine
<input checked="" type="checkbox"/> Démarrage des AE par le MSender	<input type="checkbox"/> QUARTZ		<input type="checkbox"/> Reçu Checker
<input type="checkbox"/> Arrêt des AE par le MSender	<input type="checkbox"/> STB		<input type="checkbox"/> Terminal
	<input type="checkbox"/> S8Q		


Consigne de transfert



Mode de diffusion

Taille du fichier :

**Définition des cibles**

**Horaires**

Date de démarrage

à

Débit en kbps :
( max = 1024 ; préconisé = 128 )

Nombre de répétitions :
Passes partielles :
☒

Date de fin estimée

**Figure 30: Interface de création d'une planification**

#### 4.2.5. Interface de détail d'une planification

La figure suivante indique les différents détails d'une planification. Cette interface est apparus après la validation de la planification.

Planification	
Origine :	mode manuel
Action :	Démarrage des AE par le MSender
Plateformes :	NSTBM+NSTBU
Paquetage :	Quickpack
Fonctions :	Décodeur Caisse+Décodeur Jeux+Décodeur Vitrine
Mode :	total
Consigne :	213
Restreinte :	non
Equip. cibles :	0
Etat de la diffusion : en attente	
Début :	23 juil. 2016 à 12:00:00
Fin :	23 juil. 2016 à 12:00:17
Débit :	128 (kbps)
Passes :	1 totale + 1partielle
Taille du fichier :	5,64Ko
Job ID :	1162

**Figure 31: Détails d'une planification**

#### **4.2.6. Interface reporting : suivi du parc**

Le suivi du parc (Reporting) est une fonction du site unicast, il consiste à suivre le taux de diffusion par plateforme, par fonction et par paquetage réalisé par le STM. La figure suivante le décrit.

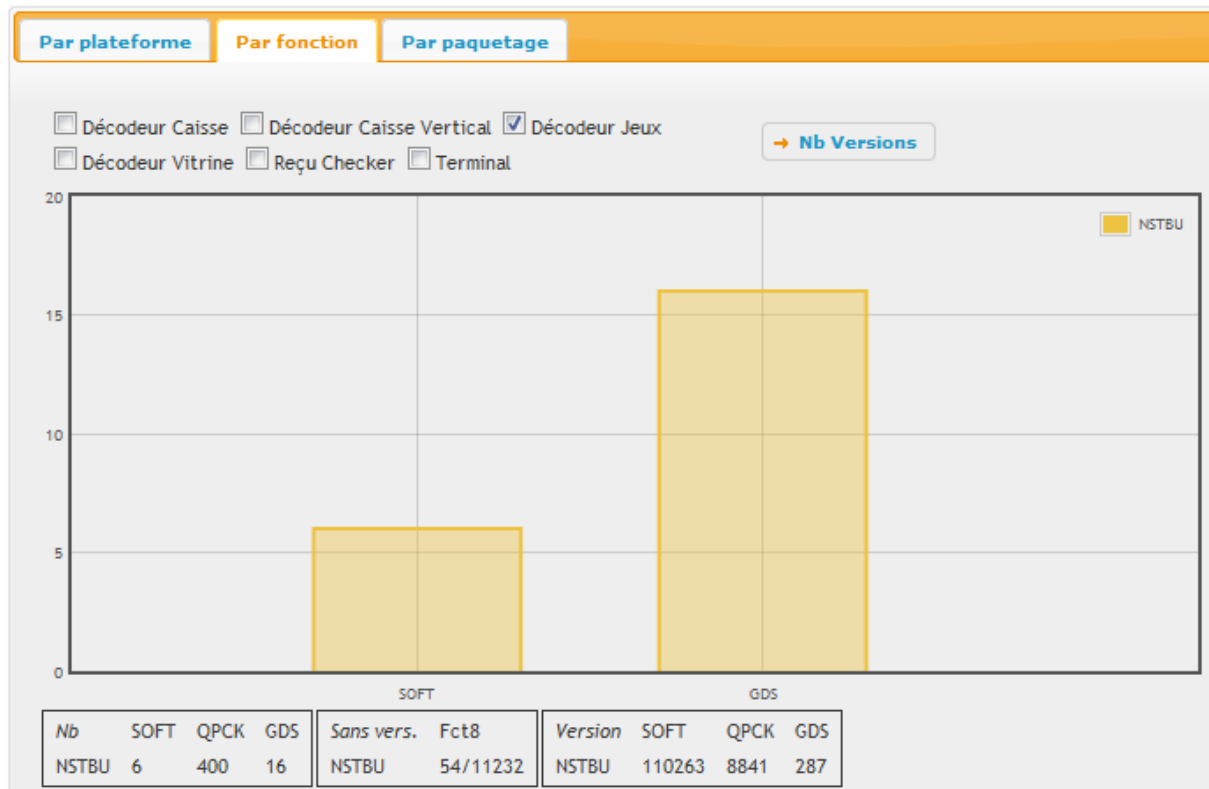


Figure 32: Suivi d'un parc

#### 4.2.7. Interface de purge des paquetages et des planification

La fonction purge d'un QP consiste à choisir une date limite de purge. Après cette dernière le contenu est totalement détruit et la source est vide. Le schéma suivant montre cette action.

##### Purge des paquetages

Toutes les anciennes versions inutilisées seront purgées. Il est conseillé de purger les planifications avant les versions.

Rechercher

##### Purge des planifications

Date limite de purge :  H : 0 M : 0 Visualiser

Requêtes

Etat de l'exécution

Annuler

Figure 33: Interface de purge d'une planification et d'un QP

### 4.3. Tests unitaires

Lorsqu'un logiciel ou une application sont créés, il est vital de réaliser plusieurs types de tests pour s'assurer que le produit est fini, complet, sécurisé et efficace. Pour réaliser ces tests, plusieurs méthodes sont possibles : les tests en « boîte noire » et en « boîte blanche ». Chacune de ces méthodes offre des possibilités différentes, que nous allons exposer dans cette partie.

#### 4.3.1. Test en « boîte noire »

Les tests en « boîte noire » consistent à examiner uniquement les fonctionnalités d'une application, c'est-à-dire si elle fait ce qu'elle est censée faire, peu importe comment elle le fait. Sa structure et son fonctionnement interne ne sont pas étudiés. Le testeur doit donc savoir quel est le rôle du système et de ses fonctionnalités, mais ignore ses mécanismes internes. Il a un profil uniquement « utilisateur ».

#### 4.3.2. Test en « boîte blanche »

Les tests en « boîte blanche » consistent à examiner le fonctionnement d'une application et sa structure interne, ses processus, plutôt que ses fonctionnalités. Sont ici testés l'ensemble des composants internes du logiciel ou de l'application, par l'intermédiaire du code source, principale base de travail du testeur.

Afin d'évaluer le fonctionnement des cas d'utilisation, on a choisi l'approche de type « boîte blanche » en utilisant un paquetage constitué de :

- Des images.
- Des données multimédia.
- Des fichiers textuelles.
- Le fichier `version_descriptor.xml`. Il fournit au serveur toutes les informations sur la version.
- Un fichier `terminal.ref.xml` détaillant les propriétés de la version et de ses fichiers.
- Un fichier `signature.pem` qui contient la signature du `terminal.ref.xml` par une clé privée détenue par l'auteur de l'archive.
- Le certificat de sécurité qui, avec la signature, permet de vérifier l'intégrité du `terminal.ref.xml`.

Ci-dessous un tableau récapitulant les différents tests que nous avons mis en point afin d'évaluer tous les cas d'utilisation.

Nombre de tests	Cas d'utilisation	Résultat attendu
16	Actualisation du QP.	<b>OK</b>
21	Gestion de diffusion.	<b>OK</b>
10	Scrutation et contrôle d'une source	<b>OK</b>
14	Administration du module QP.	<b>11 OK</b>
		<b>3 NOK (1<sup>er</sup> test FDJ)</b>
		<b>3 OK (2ème test FDJ)</b>

**Tableau 6: Récapitulatif des tests**

On a choisie de montrer les tests unitaires relatifs au cas d'utilisation « Scrutation et contrôle d'une source afin de vérifier la non-regression de l'application et les différents fonctionnalités réélisées.

Le tableau 6 présente l'installation et la mise en place de serveur Cygwin afin d'évaluer les cas d'utilisation en exploitant une source distante.

<b>N° Test</b>	
<b>Nom du test</b>	CONF_TEST_1
<b>Description du test</b>	Installation et configuration serveur SSH
<b>Mode opératoire</b>	On fait une configuration selon le manuel développeur.
<b>Résultat attendu</b>	
<b>Statut</b>	<b>OK</b>
<b>Version du test</b>	4.2.0.rc1
<b>Testeur</b>	WSA

**Tableau 7: Installation et configuration du serveur SSH**

Le tableau ci-dessous illustre la configuration des fichiers properties dont le fichier quickpack.properties qui gère les différents type des sources (locale et distante).

<b>N° Test</b>	
<b>Nom du test</b>	CONF_TEST_2
<b>Description du test</b>	Configuration du plateforme
<b>Mode opératoire</b>	On vérifie que les paramètres suivants sont disponibles dans le fichier quickpack.properties de config de l'appli avec les sources locales et les sources SSH: SRC_TYPE_<index>, SRC_SECURITY_<index>
<b>Résultat attendu</b>	
<b>Statut</b>	OK
<b>Version du test</b>	4.2.0.rc1
<b>Testeur</b>	WSA

**Tableau 8: Configuration de la plateforme de test**

Ce tableau montre le test de scrutation manuelle d'une source locale sans heure de scan indiquée dans le fichier properties.

<b>N° Test</b>	7814
<b>Nom du test</b>	QP_SCAN_7814_0
<b>Description du test</b>	Scan manuel source locale sans heure imposée dans quickpack.properties
<b>Mode opératoire</b>	Création d'une source locale. Pas de scrutation indiquée dans le quickpack.properties. Dépôt d'un QP dans la source locale. Scan manuel de la source créée.
<b>Résultat attendu</b>	Le contenu est affiché au niveau de l'interface Quickpack.validation. Le contenu est visible dans le répertoire storage.
<b>Statut</b>	OK
<b>Version du test</b>	4.2.0.rc1
<b>Testeur</b>	WSA

**Tableau 9: Description du test « QP\_SCAN\_7814\_0 »**

Le tableau ci-dessous présente le test de non scrutation d'une source (locale ou distante) si cette dernière ne dispose pas d'une heure de scrutation automatique.

<b>N° Test</b>	7814
<b>Nom du test</b>	QP_SCAN_7814_1
<b>Description du test</b>	Pas de scrutation si cela n'est pas indiqué dans le quickpack.properties
<b>Mode opératoire</b>	Création d'une source locale sans heure de scrutation. Dépôt d'un QP dans la source locale.
<b>Résultat attendu</b>	Au bout de 24 heures le contenu n'est pas traité.
<b>Statut</b>	OK
<b>Version du test</b>	4.2.0.rc1
<b>Testeur</b>	WSA

**Tableau 10: Description du test « QP\_SCAN\_7814\_1 »**

Le tableau 10 présente l'évaluation de la création d'une source locale avec heure de scrutation ainsi que le scan automatique d'un QP existant dans cette source.

<b>N° Test</b>	7814
<b>Nom du test</b>	QP_SCAN_7814_2
<b>Description du test</b>	Scan automatique source locale.
<b>Mode opératoire</b>	Création d'une source locale avec heure de scrutation indiquée dans le quickpack.properties. Dépôt d'un QP dans la source locale.
<b>Résultat attendu</b>	Après l'heure de scrutation le contenu est affiché au niveau de l'interface Quickpack.validation. Le contenu est visible dans le répertoire storage.
<b>Statut</b>	OK
<b>Version du test</b>	4.2.0.rc1
<b>Testeur</b>	WSA

**Tableau 11: Description du test « QP\_SCAN\_7814\_2 »**



Le tableau ci-dessous montre le test de la scrutation manuelle d'une source SSH.

<b>N° Test</b>	7814
<b>Nom du test</b>	QP_SCAN_7814_3
<b>Description du test</b>	Scan manuel d'une source SSH.
<b>Mode opératoire</b>	Création d'une source SSH Dépôt d'un QP dans la source. Scan manuel de la source créée.
<b>Résultat attendu</b>	Avant le scan manuel: le contenu n'est pas affiché. Après le scan manuel le contenu est affiché au niveau de l'interface Quickpack -> validation.
<b>Statut</b>	OK
<b>Version du test</b>	4.2.0.rc1
<b>Testeur</b>	WSA

**Tableau 12: Description du test « QP\_SCAN\_7814\_3 »**

Le tableau 13 expose l'évaluation de la scrutation manuelle d'une source locale avec succès qui dispose d'une heure de scrutation automatique dans le fichier quickpack.properties.

<b>N° Test</b>	7814
<b>Nom du test</b>	QP_SCAN_7814_4
<b>Description du test</b>	Scan manuel d'une source locale automatique avec succès (> 10 min avant).
<b>Mode opératoire</b>	Création d'une source locale. Dépôt d'un QP dans la source locale. Avant l'heure de scrutation automatique: scan manuel de la source créée.
<b>Résultat attendu</b>	Avant le scan manuel: le contenu n'est pas affiché. Après le scan manuel le contenu est affiché au niveau de l'interface Quickpack.validation. Le contenu est visible dans le répertoire storage.
<b>Statut</b>	OK
<b>Version du test</b>	4.2.0.rc1
<b>Testeur</b>	WSA

**Tableau 13: Description du test « QP\_SCAN\_7814\_4 »**

Le tableau 14 expose l'évaluation de la scrutation manuelle d'une source locale avec message d'erreur qui dispose d'une heure de scrutation automatique dans le fichier quickpack.properties.

<b>N° Test</b>	7814
<b>Nom du test</b>	QP_SCAN_7814_5
<b>Description du test</b>	Scan manuel d'une source locale automatique avec message erreur.
<b>Mode opératoire</b>	Création d'une source locale. Dépôt d'un QP dans la source locale. Moins de 10 minutes avant l'heure de scrutation automatique: scan manuel de la source créée.
<b>Résultat attendu</b>	Un message d'erreur est affiché après avoir cliqué sur scan manuel.
<b>Statut</b>	OK
<b>Version du test</b>	4.2.0.rc1
<b>Testeur</b>	WSA

**Tableau 14: Description du test « QP\_SCAN\_7814\_5 »**

Le tableau 15 expose l'évaluation de la scrutation manuelle d'une source SSH avec succès qui dispose d'une heure de scrutation automatique dans le fichier quickpack.properties.

<b>N° Test</b>	7814
<b>Nom du test</b>	QP_SCAN_7814_6
<b>Description du test</b>	Scan manuel d'une source SSH automatique avec succès (> 10 min avant) (non-regression).
<b>Mode opératoire</b>	Création d'une source ssh. Dépôt d'un QP dans la source locale. Avant l'heure de scrutation automatique: scan manuel de la source créée.
<b>Résultat attendu</b>	Avant le scan manuel: le contenu n'est pas affiché. Après le scan manuel le contenu est affiché au niveau de l'interface Quickpack.validation. Le contenu est visible dans le répertoire storage.
<b>Statut</b>	OK
<b>Version du test</b>	4.2.0.rc1
<b>Testeur</b>	WSA

**Tableau 15: Description du test « QP\_SCAN\_7814\_6 »**

Le tableau 16 expose l'évaluation de la scrutation manuelle d'une source SSH avec un message d'erreur qui dispose d'une heure de scrutation automatique dans le fichier quickpack.properties.

<b>N° Test</b>	7814
<b>Nom du test</b>	QP_SCAN_7814_7
<b>Description du test</b>	Scan manuel d'une source SSH automatique avec message erreur (non-regression).
<b>Mode opératoire</b>	Création d'une source locale. Dépôt d'un QP dans la source locale. Moins de 10 minutes avant l'heure de scrutation automatique: scan manuel de la source créée.
<b>Résultat attendu</b>	Un message d'erreur est affiché après avoir cliqué sur scan manuel.
<b>Statut</b>	OK
<b>Version du test</b>	4.2.0.rc1
<b>Testeur</b>	WSA

**Tableau 16: Description du test « QP\_SCAN\_7814\_7 »**

## Conclusion

A travers ce chapitre, nous avons exposé les différents plateformes et outils pour le développement des fonctionnalités de l'application de projet STM ainsi que la réalisation des tests que nous avons implémenté. Et par la suite, nous avons présenté les modules implémentés avec des aperçus de l'application.

A ce stade, nous avons réussi donc à développer la version 4.2.0 de notre application pour arriver à un produit complet et fonctionnel. Notre application Web est donc prête à être exploiter en offrant à l'utilisateur FDJ de gérer un QP portant la fonctionnalité AE.

## **Conclusion générale**

En guise de conclusion, le travail que nous avons réalisé consiste à la réalisation d'une évolution de l'application logicielle « Serveur de Téléchargement Multicast » muni par la « Française Des Jeux » et l'évaluation de projet par la réalisation d'une série de test.

Ce travail nous a permis de mettre en pratique et d'approfondir nos connaissances en développement des applications web, en langage de programmation Java, en technologie JSP, ainsi que toutes les étapes du développement d'un logiciel et principalement d'approfondir la connaissance sur la pratique des tests.

En termes d'objectifs nous avons atteint la totalité des besoins introduites à la phase de spécification et nous avons obtenu un produit fonctionnel et de qualité qui répond aux exigences de notre client FDJ.

Dans le travail réalisé, nous sommes trouvés face à des difficultés nouvelles. D'une part, il a fallu, en effet, comprendre la manière dont le code de la version courante a été pensé alors que nous ne disposions que de très peu de documentation.

D'autre part, la non disposition de quelques détails sur l'environnement de test réel de la FDJ a engendré des anomalies au niveau de la diffusion d'un ordre de reboot chez le client après que nous avons réalisé le premier test.

Cependant, cette expérience au sien de la société Quadrille, nous a offert la bonne occasion pour sortir du cadre théorique et appliquer les connaissances acquises lors des études universitaires dans un environnement réel de travail qui nous a permis de nous initier dans le domaine professionnel et d'apprendre plusieurs attitudes et habitudes sociales telles que le travail en groupe et la collecte d'informations pour extraire les besoins des acteurs du système à mettre en œuvre.

Finalement, les objectifs fixés au départ du présent projet ont été réalisés. Toutefois, il y a lieu de signaler que plusieurs améliorations peuvent être par la suite réalisées à différents niveaux, tels que le module QP qui est prévu devenir ultérieurement une application séparée et la mise à jour d'un contenu qui est souhaité être réalisé par le site unicast après leur récupération de la base de données afin d'alléger la charge sur le STM.

# Nétographie

- [1] : <http://www.directetudiant.com/entreprise/14916/quadrille-ingenierie>, *visité le 24/03/2016*
- [2] : <http://www.oezratty.net/wordpress/2014/francais-tv-connectee-quadrille/>, *visité le 25/03/2016*
- [3] : <http://www.kuzeo.com/jeux-remuneres/9/francaise-des-jeux>, *visité le 26/03/2016*
- [4] : <http://www.societe.com/societe/lotsys-418618088.html>, *visité le 30/03/2016*
- [5] : <http://www.groupefdj-recrute.com/fr/groupe/activite/filiales/>, *visité le 30/03/2016*
- [6] : <http://www.math-info.univ-paris5.fr/~bouzy/Doc/UML-NotesCours.pdf>, *visité le 01/04/2016*
- [7] : <https://openclassrooms.com/courses/apprendre-asp-net-mvc/le-pattern-mvc>, *visité le 01/04/2016*
- [8] : <http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc31018.1650/doc/html/rad1232632577723.html>, *visité le 02/04/2016*, *visité le 02/04/2016*
- [9] : <http://www.jmdoudoux.fr/java/dej/chap-outils.htm>, *visité le 02/04/2016*
- [10] : <http://tomcat.apache.org/tomcat-9.0-doc/changelog.html>, *visité le 04/02/2016*
- [11] : <https://cygwin.com/>, *visité le 03/29/2016*
- [12] : <http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>, *visité le 04/04/2016*
- [13] : <http://www.pack-logiciels-libres.fr/spip.php?logiciel35>, *visité le 11/04/2016*
- [14] : <http://debian-facile.org/doc:debian>, *visité le 11/04/2016*
- [15] : <http://www.techno-science.net/?onglet=glossaire&definition=764>, *visité le 16/04/2016*
- [16] : <https://www.edrawsoft.com/fr/UML-Diagrams.php>, *visité le 25/04/2016*
- [17] : <http://www.additeam.com/SSII/uml/>, *visité le 01/05/2016*
- [18] : <http://www.jmdoudoux.fr/java/dej/chap-presentation.htm>, *visité le 01/05/2016*
- [19] : <http://www.prologue-solutions.tn/technologies/67-xml.html>, *visité le 01/05/2016*
- [20] : [https://ensweb.users.info.unicaen.fr/cours/L2\\_semaine03/](https://ensweb.users.info.unicaen.fr/cours/L2_semaine03/), *visité le 15/05/2016*
- [21] : <http://77.158.28.92:28080/christian.robe/Jsp/cours/jsp2013.pdf>, *visité le 16/05/2016*

[22] : <http://www.lesdeveloppeurs.net/BD-SGBD-definition-de-sql-274>, *visité le 16/05/2016*

# Annexe A: Diagrammes d'activité

## Introduction

Le *diagramme d'activité* est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables (multi-threads ou multi-processus). Le diagramme d'activité est également utilisé pour décrire un flux de travail (workflow).

## Diagramme d'activité « Mise à jour d'un QP »

Le diagramme ci-dessous présente le processus de mise à jour d'un QP avant sa diffusion. Il permet de vérifier la validité de contenu et de mettre à jour son fichier de référence.

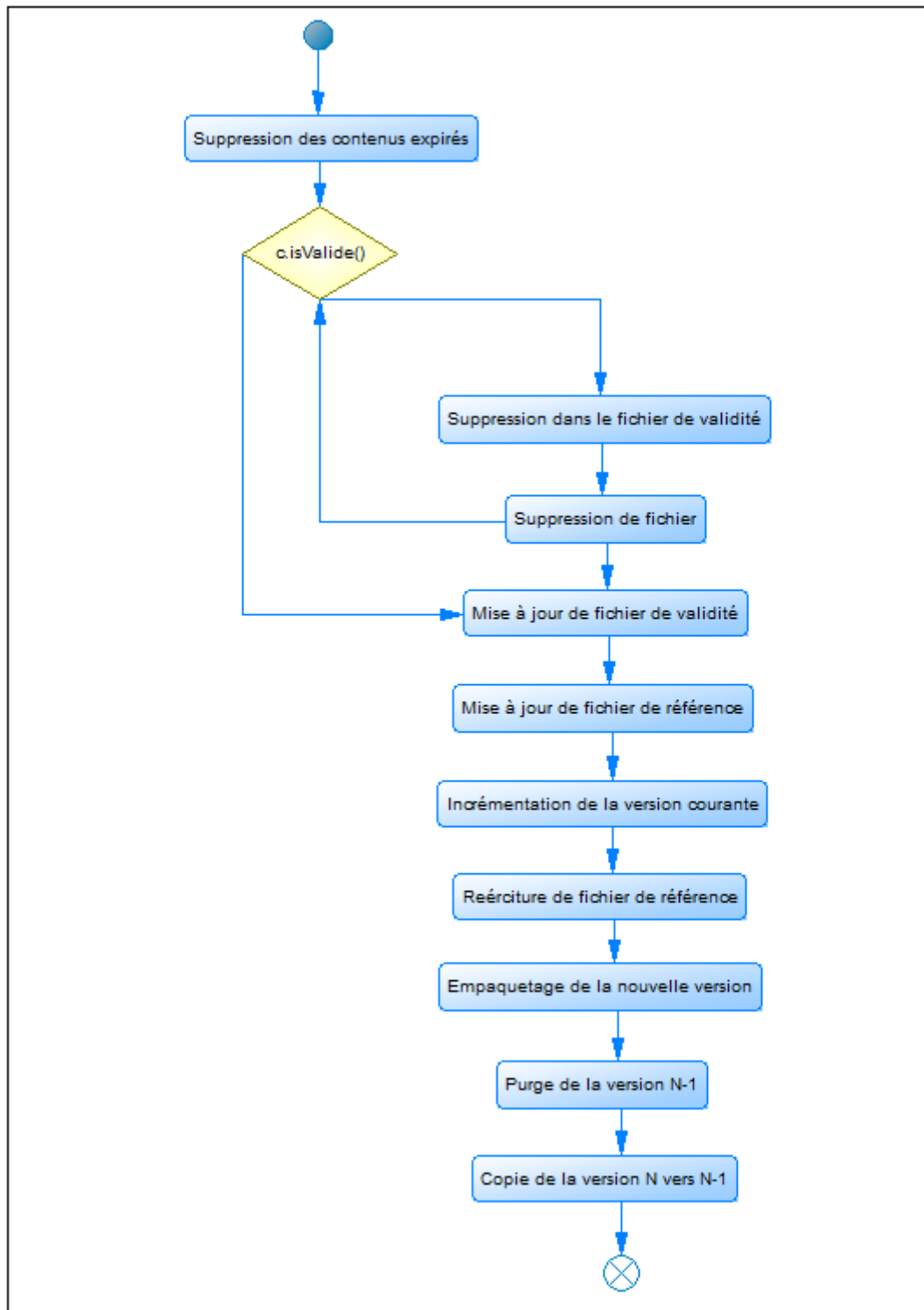


Figure 34: Diagramme d'activité « Mise à jour d'un QP »

## Diagramme d'activité « Lecture et initialisation des Sources »

Avant toute scrutation et diffusion d'un contenu, le STM lit et initialise toutes les sources (distantes et locaux). La figure suivante illustre les différents étapes pour ce processus.



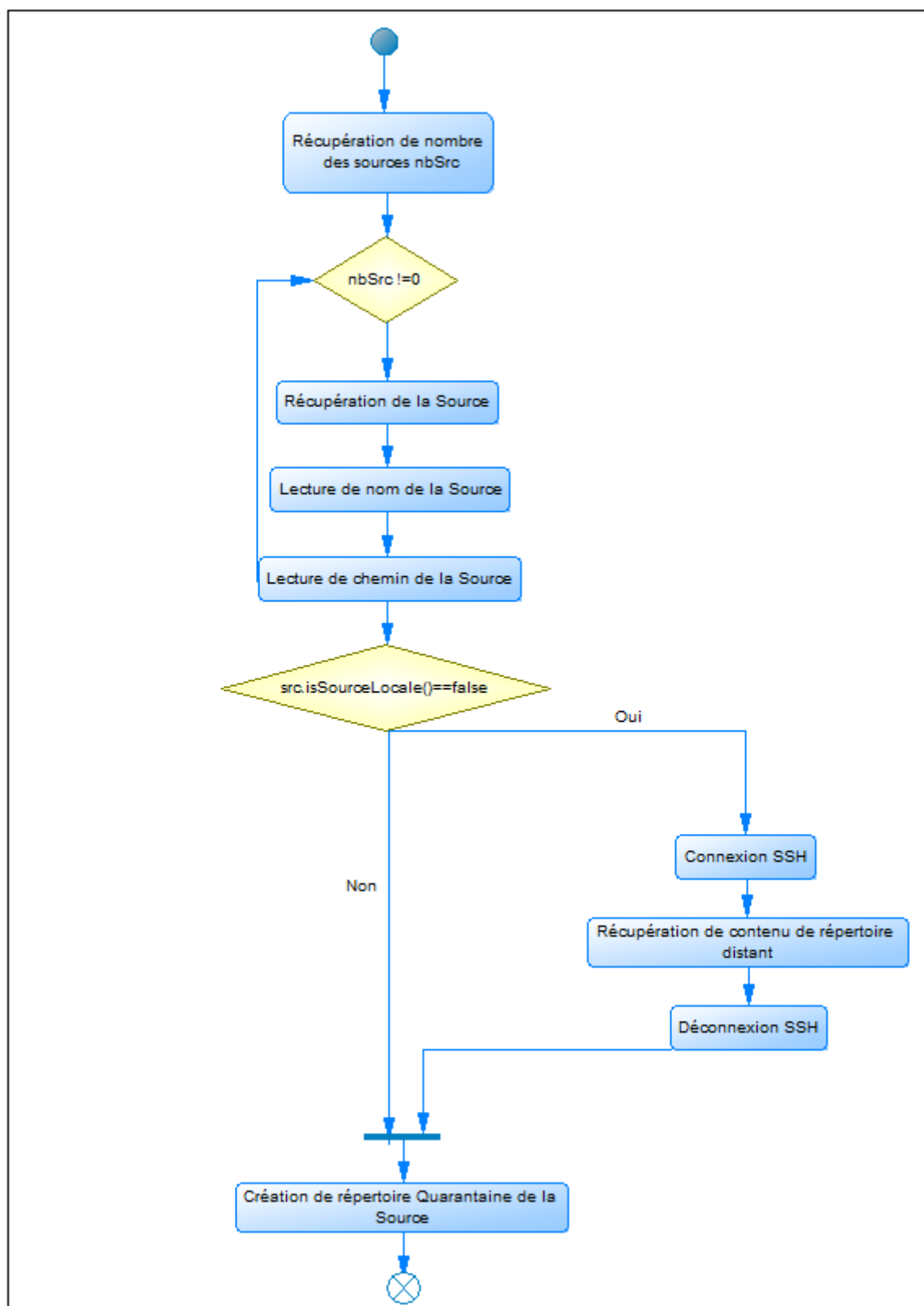


Figure 35: Diagramme d'activité « Lecture et initialisation des sources »

## Diagramme d'activité « Diffusion d'un QP »

Le diagramme ci-dessous présente l'enchaînement de soumission d'un job à MSender. Ce processus consiste à créer la version à diffuser, de joindre les paramètres d'une planification de diffusion ainsi de sauvegarder une copie en locale.

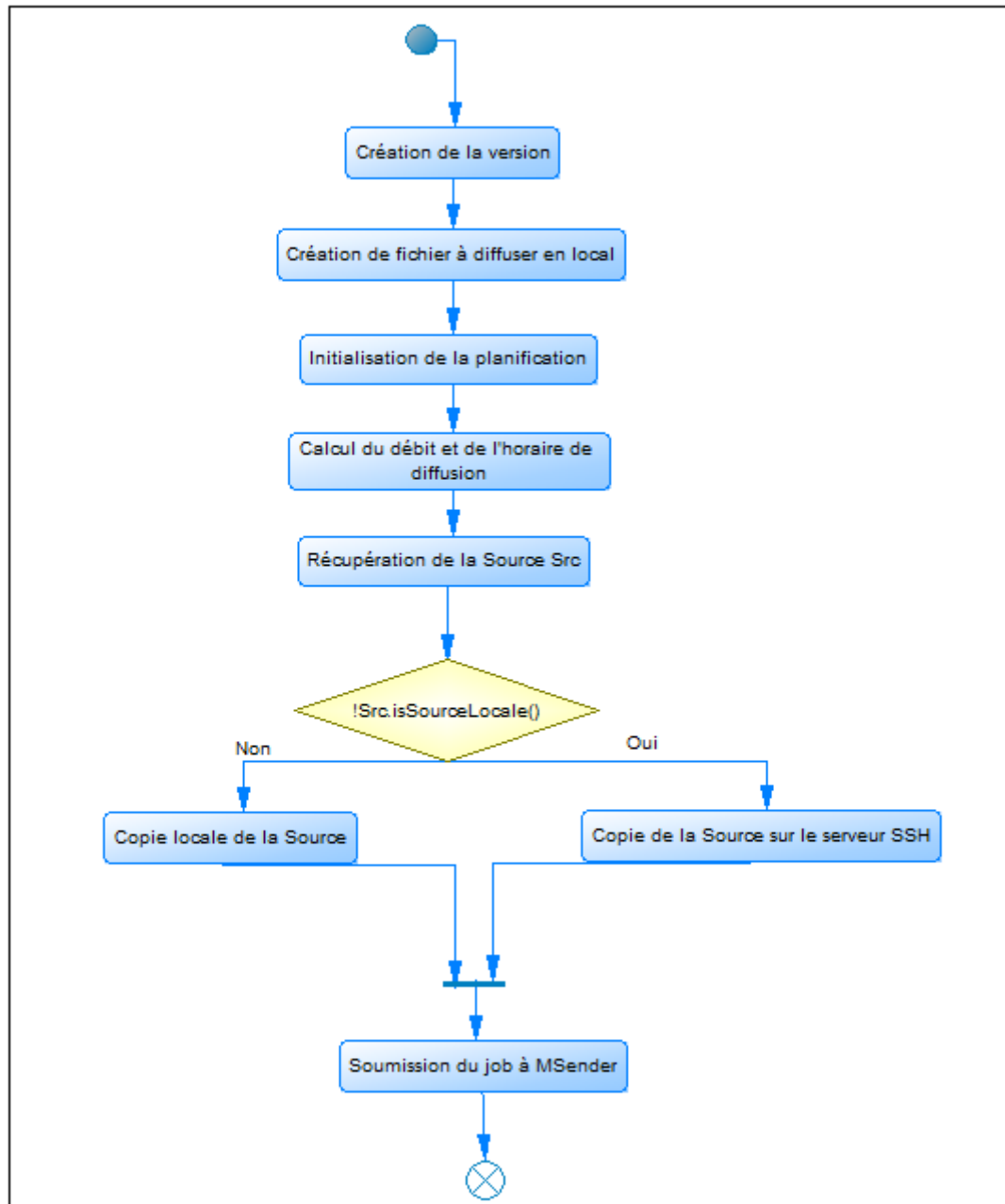


Figure 36 Transfert d'un QP

## Diagramme d'activité « Archivage d'un contenu »

Le diagramme présenté ci-dessous permet d'exposer l'enchaînement de l'archivage et la copie d'un contenu sur une source locale ou distante.

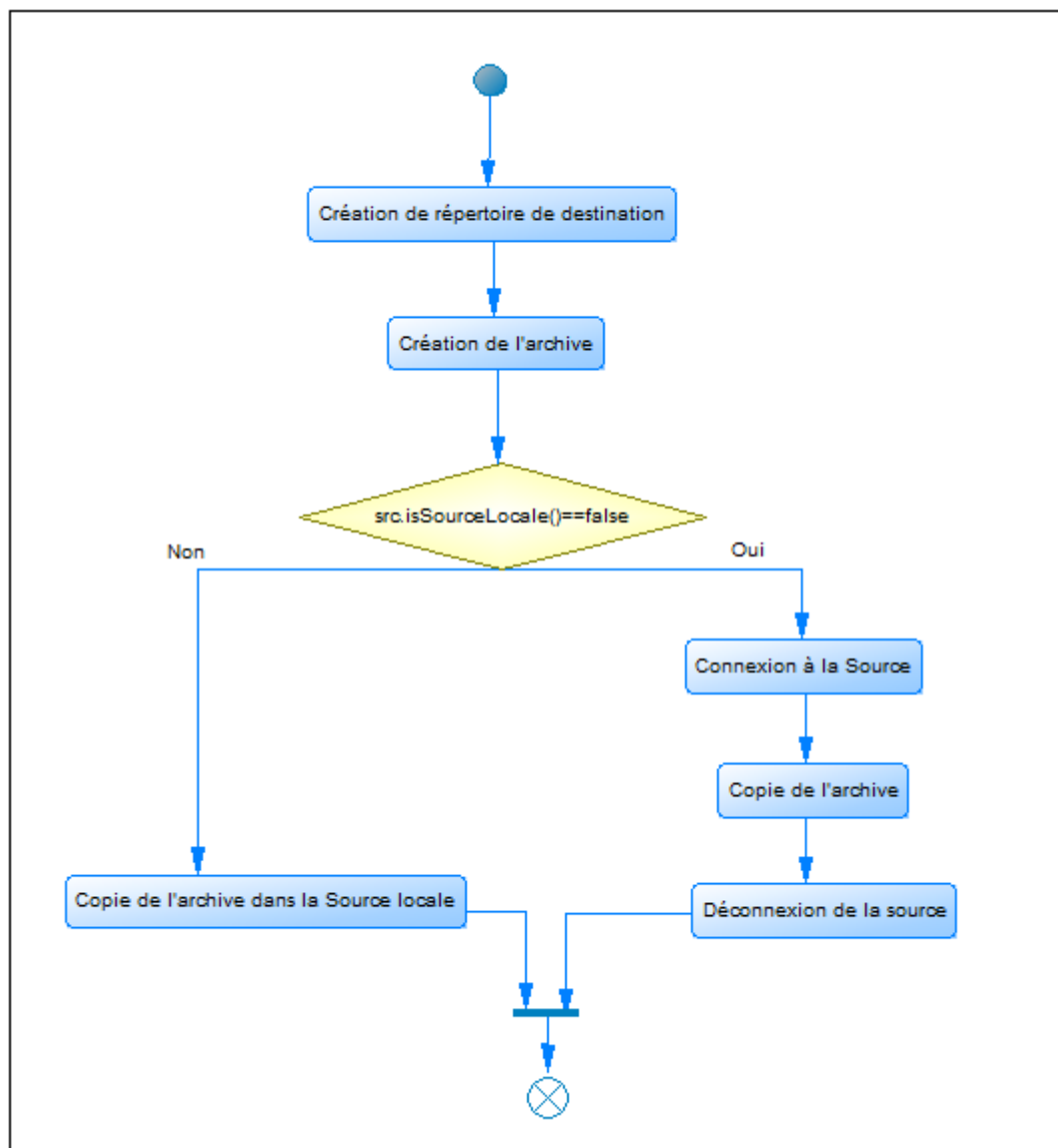


Figure 37: Diagramme d'activité « Archivage d'un QP »