

**Projet sous thème :**

**Etude, conception et réalisation du module  
"Teleload Planning Tool" dans le cadre du  
projet Xenturion – Atos Wordline**

**Préparé par :** Oumayma OULAHOU

**Sous la direction de :**

M. Youssef BALOUKI (FST SETTAT)

M. Abderrahmane AIT ABDELKADER (ATOS)

*Soutenu publiquement comme exigence partielle en vue de l'obtention du*

**Diplôme Master sciences et techniques**

**Filière Ingénierie de conception et développement d'application**

**Membre de jury :**

M. Youssef Balouki (FST Settat)

M. Abderrahim Marzouk (FST Settat)

M. Abderrahmane Ait Abdelkader (ATOS)



# Remerciements

Au terme de ce travail, je tiens à exprimer mes profonds remerciements à toutes les personnes qui ont contribué au bon déroulement de mon projet de fin d'études, afin de faire de cette expérience un vrai plaisir et non une formalité de fin de parcours, stressante et contraignante.

Ainsi, j'exprime ma profonde gratitude à monsieur YOUSSEF BALOUKI mon encadrant à la faculté sciences et techniques à Settat. Mes remerciements les plus sincères pour vos conseils ainsi votre assistance pour l'amélioration de toutes les phases de ce projet.

Mes remerciements aussi à pour monsieur TAOUFIK GADI et toute l'équipe pédagogique, qui ont assuré le succès du master ingénierie de conception et développement d'application promotion 2018.

Je tiens aussi à exprimer ma profonde gratitude à monsieur AIT ABDELKADER ABDERRAHMANE, le chef de projet Xenturion-Wordline au Maroc, et à KAOUTAR EL GHAILANI mon mentor à Atos, pour leur disponibilité, encouragements, et les conseils précieux qu'ils m'ont prodigués, et qui m'ont été d'un fort appui.

Je présente également mes remerciements à tous les membres de l'équipe Xenturion, notamment KAISSY MOHAMED, ABDELHADI FARAH, LAHROUD ABDELLAH et YOUNESS BENDIDI. Je leur remercie beaucoup pour leur sympathie, leur disponibilité, leurs compétences et leurs conseils qui m'ont permis de surmonter mes difficultés et de progresser dans mon projet de fin d'études.

Je ne saurais oublier ma profonde gratitude à CHAMRANI YOUSSEF, le Delivery Manager, de m'avoir accueilli et assuré toutes les conditions afin que je puisse effectuer ce stage.

Bien sûr ne pas oublier les coordonnateurs de Wordline Belgique PHILIPPE MAILLEU, OLIVIER MATTE et FILIP YSENBAERT pour leurs confiance, suivie et conseils.

Que les membres du jury, trouvent ici l'expression de ma reconnaissance pour avoir accepté de juger mon travail. Que tous celles et ceux qui ont contribué de près ou de loin à l'accomplissement de ce travail, trouvent l'expression de mes remerciements les plus sincères.

# Résumé

Dans le cadre de mon stage de fin d'études, effectué au sein du projet Xenturion chez Atos Maroc-Casablanca, j'ai eu l'occasion de participer à l'étude, la conception, et le développement du module Teleload Planning Tool.

L'objectif du projet est de développer un module de gestion de planification des Teleloads pour un terminal de paiement électronique Wordline. Pour cette fin j'ai proposé un algorithme d'ordonnancement des teleloads dans le temps selon une paramétrisation fourni par l'utilisateur et proposer une représentation graphique des résultats sous forme de digrammes donnant une vue complète sur le statut d'une compagne de teleload.

Ma mission consiste d'abord à étudier le besoin du client Atos-Wordline, le découper en tâches, puis proposer une solution performante et des algorithmes de scheduling optimisés.

Le présent rapport constitue une synthèse du travail réalisé durant la période du stage. Il présente les différentes étapes suivies pour accomplir la mission qui m'a été confiée.

**Mots-clés:** ZK, EJB, SPRING, Algorithme ordonnancement, Algorithme packing, planification, performance, optimisation

# Abstract

As part of my internship, carried out within the Xenturion project at Atos Maroc-Casablanca, I had the opportunity to participate in the study, design, and development of the module Teleload Planning Tool.

The objective of the project is to develop a Teleloads planning management module for a Wordline electronic payment terminal. For this purpose I proposed an algorithm for scheduling teleloads over time according to a parameterization provided by the user and to propose a graphical representation of the results in the form of digrams giving a complete view of the status of a teleload companion.

My mission is first to study the needs of the Atos-Wordline client, then proposing a powerful solution and optimized scheduling algorithms.

This report summarizes the work done during the internship period. It presents the various steps followed to accomplish the mission entrusted to me.

**Keywords:** ZK, EJB, SPRING, Algorithm scheduling, Algorithm packing, planning, performance, optimisation

# Table des matières

Liste des figures.....	8
Liste des abréviations .....	9
Introduction générale .....	10
Chapitre 1: Contexte général du projet.....	13
I.    Présentation de l'organisme d'accueil.....	14
1.    Fondation d'Atos.....	15
2.    Présentation du groupe Atos .....	15
3.    Atos Maroc .....	16
4.    Domaines d'activités du groupe Atos .....	17
5.    Clients d'Atos .....	17
II.    Contexte du Project.....	18
Chapitre 2: Analyse Fonctionnelle et Conception.....	22
I.    Etude de l'existant et présentation de la problématique .....	23
II.    Capture des besoins fonctionnels .....	23
1.    Exigences.....	24
2.    Solution proposée .....	25
3.    Conduite du projet.....	29
4.    Méthode Scrum .....	29
5.    Demarche DevOps .....	31
Chapitre 3: Étude technique .....	34
I.    L'architecture du projet Xenturion.....	35
1.    Architecture Java EE.....	36
2.    L'architecture de la solution .....	36
3.    Conception .....	39
a.    Diagramme de cas d'utilisation .....	39
b.    Diagramme de séquence .....	39
c.    Diagramme de classe.....	48
4.    Améliorer les performances .....	49
a.    Optimisation combinatoire .....	49
b.    Benchmarking.....	49
5.    Optimization.....	51
a.    Théorie des graphes.....	52
b.    Linear approximation algorithm for First-fit decreasing bin-packing algorithm .....	53

<b>Chapitre 4: Mise en œuvre .....</b>	<b>55</b>
<b>I. Technologie Utilisés .....</b>	<b>56</b>
1. Patrons utilisés .....	56
2. Frameworks.....	57
3. Langage de modélisation .....	59
4. Usine de développement .....	59
5. Outils Système de gestion base de données .....	61
<b>II. Présentation graphique.....</b>	<b>62</b>
<b>Conclusion .....</b>	<b>68</b>
<b>Bibliographie .....</b>	<b>69</b>
<b>Webographie .....</b>	<b>70</b>

## Liste des figures

Figure 1 : Carte de distribution des employés du groupe Atos .....	14
Figure 2 : Historique d'Atos .....	15
Figure 3 : L'organigramme d'Atos .....	16
Figure 4 : Clients d'Atos .....	18
Figure 5 : Les Activités de Worldline .....	18
Figure 6 : Les services de Xenturion .....	20
Figure 7 : Chart avec teleload planifié et reserved.....	24
Figure 8 : Chart avec le max désigné.....	25
Figure 9 : Structure de POS_SCHEDULING et POS_SCHEDULING_RESERVED.....	26
Figure 10 : Cycle de vie d'une campagne .....	27
Figure 11 : Structure de la campagne.....	27
Figure 12 : Points of services d'une campagne .....	28
Figure 13 : Durée de teleload.....	28
Figure 14 : Algorithme proposé pour Service.UTC.....	29
Figure 15 : Principe du SCRUM.....	29
Figure 16 : Devops Evolution .....	31
Figure 17 : Cycle de vie DevOps .....	32
Figure 18 : Architecture logicielle .....	37
Figure 19 : Application et modèle MVC .....	38
Figure 20 : Diagramme de cas d'utilisation.....	39
Figure 21 : Génération de la charte.....	41
Figure 22 : Créer point of service .....	43
Figure 23 : Principales topologies typiques de graphes.....	53
Figure 24 : Algorithme bin packing.....	53
Figure 25 : Page authentification .....	62
Figure 26 : Champs pour générer la chart.....	62
Figure 27 : Remplissage de la base de données .....	63
Figure 28 : Charte généré.....	63
Figure 29 : Changer la size de la charte .....	64
Figure 30 : Option de nombres de colonnes .....	64
Figure 31 : Option de zoom .....	64
Figure 32 : Menu pour enregistrer la charte.....	65
Figure 33 : Création d'un nouveau POS.....	65
Figure 34 : Choisir ID pour une POS.....	65
Figure 35 : Choisir la valeur de l'attribut TYPE .....	65
Figure 36 : Affecter le TYPE choisi au POS .....	66
Figure 37 : Choisir la valeur de l'attribut MEDIA .....	66
Figure 38 : Affecter MEDIA au POS.....	66
Figure 39 : Création du POS avec les attributs choisi.....	67
Figure 40 : Choisir campagne et lancer l'algorithme de planification en background.....	67



## Liste des abréviations

POS	Point of service
DevOps	Development and Operations
http	Hypertext Transfer Protocol
Java EE	Java Enterprise Edition
JNDI	Java Naming and Directory Interface
JDBC	Java Database Connectivity
API	Application programming interface
URL	Uniform Resource Locator
HTML	Hypertext Markup Language
JPA	Java Persistence API
SQL	Structured Query Language
JPQL	Java Persistence Query Language
HQL	Hibernate Query Language
NP	Nondeterministic Polynomial
MVC	Model View Controller
DTO	Data Transfert Object
ORM	Object-relational mapping
EJB	Enterprise JavaBeans
UML	Unified Modeling Language
IDE	Integrated development environment
SGBD	Système de gestion de base de données

## Introduction générale

Dans le contexte de scheduling service, le projet Teleload Planning Tool est un nouveau besoin demandé par les clients d'Atos Wordline. En effet, aujourd'hui, il n'y a pas de visibilité sur la campagne de teleload, des synchronisations régulières sont nécessaires avec l'équipe pour éviter les surcharges et les conflits de téléchargement.

Premièrement, il est demandé d'afficher un graphique montrant la charge du serveur prévue en termes de : quantité de teleload simultanés en fonction du temps, afin de donner une présentation significative des données en se basant sur une étude des solutions ayant les performances souhaitées.

Deuxièmement, l'objectif est de proposer un algorithme qui aide à planifier d'une manière optimale les scheduling et minimiser la consommation des ressources. En effet, Cet objectif permettra de diminuer les efforts pour la préparation de la campagne de teleload et de diminuer le risque d'erreur dû aux conflits de teleload.

Le Teleload Planning Tool, doit être opérationnel, évolutif, convivial et offrant les informations nécessaires en temps réel. Pour ceci, le module à réaliser doit satisfaire les exigences utilisateurs suivantes:

- Diffusion du teleload par site / client.
- Diffusion de teleload pendant une période donnée, avec utilisation optimale des créneaux disponibles sur Xenturion (tableau de bord de surveillance de la capacité teleload)

Dans le périmètre du projet Xenturion, l'agilité est l'une des critères indispensables pour l'avancement du projet. En effet, la méthode Scrum et la démarche DevOps sont mises en place pour ce besoin.

L'architecture du projet Xenturion est une architecture multi-couches. En effet, un système fondé sur une architecture en couches offre, sur différent niveaux, les avantages suivants :

- **Le développement** : Du fait que l'application est découpée en couches indépendantes, le temps consacré au développement est diminué, car chaque couche peut être programmée parallèlement et par de nombreux programmeurs
- **Les tests** : Puisque chaque couche est dotée du caractère d'indépendance vis-à-vis des couches adjacentes, chacune d'elles pourrait être testée indépendamment des autres.
- **La maintenance** : Si un problème a été détecté, on peut facilement savoir la couche qui présente une anomalie.
- **Les amendements de l'application** : la modification ou la mise à jour d'une couche n'affecte pas les autres couches.

De même manière notre solution est basée sur une architecture en couches, en utilisant le patron MVC (Model-View-Controller) :

**Son principe** : le MVC est un modèle de conception, à trois couches, qui reposent sur la volonté de séparer les données, de l'affichage et des actions.

Pour structurer notre système, nous avons opté pour un découpage en couches. Ceci est également justifié par l'architecture Java EE adoptée.

Dans une architecture multicouche sans état, il est relativement fréquent de s'échanger des objets en paramètre de méthode. Aujourd'hui, le pattern DTO (Data Transfert Object) est la technique la plus utilisée pour assurer le découplage entre la couche de présentation et les couches objets métier stockés sur le serveur.

Le projet Xenturion est construit sur la plateforme Java Enterprise Edition (Java EE) principalement et d'autre Framework comme ZK, Hibernate, EJB et Spring.

Un atout majeur de la technologie choisie est que le système est techniquement indépendant de l'infrastructure matérielle.

La solution proposée consiste à respecter deux côtés, celui de la performance et celui de l'optimisation.

Les performances sont établies par la mesure lors d'une activité appelée benchmarking grâce à des tests de performance. En effet, le test de performance consiste à définir le processus/système/matériel que l'on veut mesurer, à définir le cas d'utilisation à, et à exécuter le test tout en récoltant les mesures.

Dans notre cas, nous avons réalisé un benchmark qui porte sur quatre solutions :

- Data Transfer Object Design Pattern
- Oracle indexation
- Hibernate search
- JPQL

Pour le coté optimisation, nous avons opté pour la fusion de deux algorithmes, celui de la Théorie des graphes et le Bin packing, enfin construire notre solution propre à Xenturion.

Parmi les outils utilisés pour répondre à notre besoin, on cite :

- **UML** est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter
- **WILDFLY** est un serveur d'application Java certifié par Oracle.
- **Maven** est un outil permettant d'automatiser la gestion de projets Java.
- **IntelliJ IDEA** est un Environnement de développement intégré (IDE) Java développé et édité par JetBrains. IntelliJ est un IDE complet misant sur la productivité avec des systèmes d'autocomplétion intelligente
- **Oracle Database** est un système de gestion de base de données relationnelle qui depuis l'introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet.

Finalement, ce rapport est composé de quatre chapitres :

## **Chapitre 1 : Contexte général du projet**

Ce chapitre a pour objectif de présenter le contexte général du projet. Il commence tout d'abord, par la présentation de l'organisme d'accueil. Ensuite il présente le projet et ses objectifs. Enfin, il décrit le déroulement du projet en présentant le processus de développement adopté ainsi que sa planification.

## **Chapitre 2 : Analyse Fonctionnelle et Conception**

Ce chapitre est consacré à l'étude fonctionnelle du projet. Il commence par l'étude de l'existant et la présentation de la problématique puis la capture des besoins fonctionnels, et la solution proposée. Enfin, il décrit le déroulement du projet en présentant le processus de développement adopté ainsi que sa planification.

## **Chapitre 3 : Étude Technique**

Le choix de l'architecture technique est un facteur déterminant dans la performance de l'application, il dépend de plusieurs contraintes techniques que nous allons étudier en ce qui suit. De plus, les algorithmes et la conception adoptée du choix des diagrammes et les outils utilisés pour mettre en place la solution proposée.

## **Chapitre 4 : Mise en œuvre**

Ce chapitre traite à présent à la phase de réalisation. Il présente les fonctionnalités de l'application qui ont été développées et seront mises en production. Certaines interfaces réalisées y sont présentées et commentées.

# **Chapitre 1:**

## Contexte général du projet

Ce chapitre a pour objectif de présenter le contexte général du projet. Il commence tout d'abord, par la présentation de l'organisme d'accueil. Ensuite il présente le projet et ses objectifs :

- Présentation de l'organisme d'accueil
- Contexte du projet

## I. Présentation de l'organisme d'accueil

AtoS Technology Services Maroc - Casablanca, l'une des principales SSII françaises et le leader en France du paiement sécurisé en ligne pour les entreprises via sa filiale Worldline. Implantée dans 72 pays, l'entreprise a un chiffre d'affaires annuel de 11 milliards d'euros en 2015, et 12 milliards d'euros en 2016 avec environ 100 000 employés [1] dont 1200 au Maroc.

Atos se décrivant comme l'un des principaux acteurs internationaux dans les services informatiques en faisant partie du Top 10 mondial. Elle est parmi les leaders en Europe en service IT, et le leader du paiement sécurisé en ligne en France. Grâce à cette opération, Atos renforce son engagement en matière d'innovation et accélère sa stratégie de croissance dans le domaine des services transactionnels.

Atos délivre les technologies qui accélèrent le développement de ses clients et les aident à réaliser leur vision de l'entreprise du futur. En tant que société responsable et intégrée, Atos s'est engagé à décliner les meilleures pratiques dans les domaines environnementaux, sociaux et éthiques au sein de toute son organisation.

Le Groupe contribue ainsi à promouvoir et à développer un comportement responsable en influençant positivement ses parties prenantes pour qu'elles prennent en compte le développement durable dans leurs processus de décision.

Atos aide ses clients à avancer dans ces domaines, à réduire leur empreinte environnementale et à assurer leur viabilité future en leur fournissant des solutions innovantes et durables.



Figure 1: Carte de distribution des employés du groupe Atos

## 1. Fondation d'Atos

Le 4 juillet 2011, Atos Origin est renommé en AtoS, à la suite de l'acquisition de Siemens IT Solutions and Services.

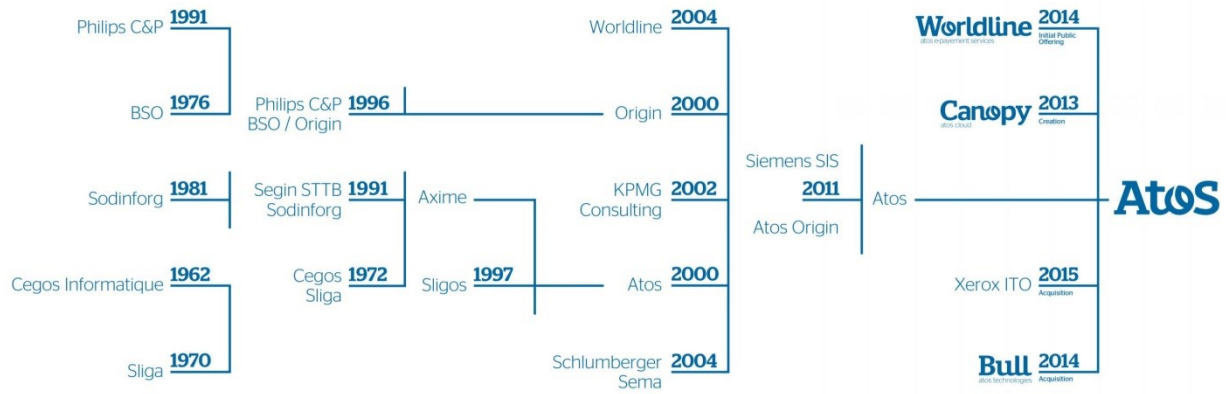


Figure 2: Historique d'Atos

## 2. Présentation du groupe Atos

Le comité est en charge de la coordination mondiale de la direction du Groupe. Il est placé sous l'autorité de Thierry Breton, Président Directeur général. Les membres sont :

- Thierry Breton, Président Directeur Général.
- Charles Dehelly, Directeur Général adjoint, coordination des Opérations Globales Et Programme TOP.
- Gilles Grapinet, Directeur Général adjoint, coordination des fonctions Globales et Directeur Général de Worldline.
- Michel-Alain Proch, Directeur Général Adjoint du Groupe et Directeur-Général des opérations en Amérique du Nord.

La figure ci-dessous représente en détail l'organigramme du groupe AtoS

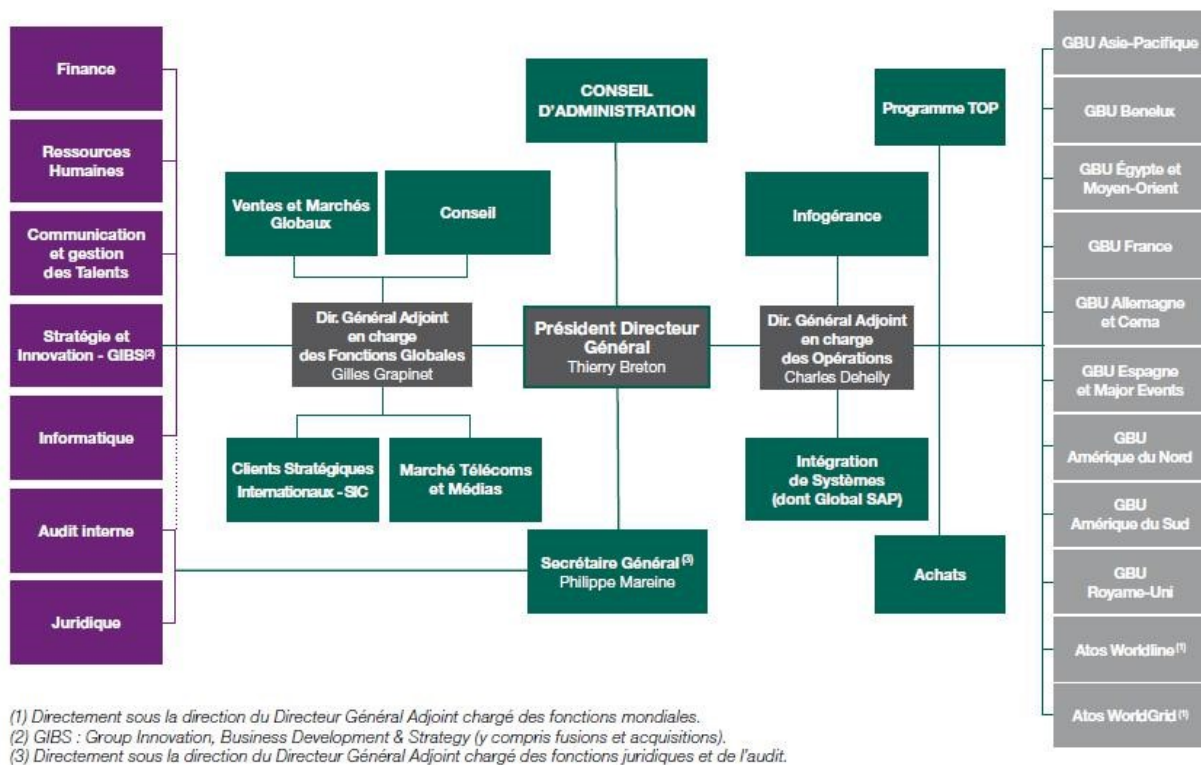


Figure 3: L'organigramme d'Atos

Atos est le partenaire numérique des Jeux Olympiques depuis 1992. De plus, le groupe compte parmi ses clients de grands comptes internationaux dans tous les secteurs d'activités. Il a développé un réseau d'alliances et de partenariats puissant, qui repose sur une solide présence au niveau local dans les pays où il est implanté.

### 3. Atos Maroc

Le groupe Atos est présent depuis 2003 sous la dénomination Atos Origin SARL devenue Atos en 2011, historiquement Atos intervient sur des projets au Maroc depuis 40 ans à travers sa société SEMA.

Aujourd'hui Atos compte plus de 1200 collaborateurs répartis sur Casablanca et Rabat, et réalise des projets d'intégration et d'infogérance pour des clients marocains, francophones et hispanophones du groupe Atos appartenant aux secteurs des télécommunications, financiers, du secteur public, de l'industrie, de l'automobile et de la distribution, via son Centre de Service (CDS) basé à Casablanca.

Atos au Maroc réalise un chiffre d'affaire annuel de 300 millions de dirhams dont le tiers réalisé sur le marché local.



#### **4. Domaines d'activités du groupe Atos**

- Le conseil & services technologiques : (8% de l'activité) : Atos conseille ses clients en leur apportant des solutions dans les télécommunications, l'industrie, les services financiers et le secteur public. Elle accompagne ses clients sur tout le cycle de réalisation du projet: de la définition de la stratégie jusqu'à la mise en œuvre des solutions.
- L'intégration de systèmes (41% de l'activité incluant 14% de TMA – Tierce Maintenance Application) : Atos part du principe « Intégrer pour simplifier ». L'activité Intégration consiste au développement, à la mise en œuvre et à la maintenance de systèmes conçus pour prendre en charge la stratégie d'entreprise des clients dans son ensemble. Atos réalise également des projets Open Source ainsi que des logiciels spécifiques reposant notamment sur des langages et des méthodes de conception différents
- L'infogérance (51% de l'activité) : Une alternative stratégique pour la gestion des risques et des coûts. Par le biais de son activité infogérance, Atos prend en charge la gestion des infrastructures informatiques clés de ses clients : centres de données, assistance microinformatique, parcs de serveurs et réseaux de communication. Atos, qui fournit, à travers son réseau mondial, des services de support accessibles 24h/24 et 7j/7, dispose d'une forte expérience en matière de déploiement de solutions complexes multi-sites.

#### **5. Clients d'Atos**

Les clients d'Atos sont aussi bien de grands groupes ou organisations multinationaux, que des entreprises de taille moyenne ou petite avec lesquelles ils travaillent en partenariat sur le long terme.

Comme je l'ai cité, le groupe est le partenaire informatique mondial du Comité International Olympique et est, à ce titre, en charge de l'informatique des Jeux Olympiques. Il est aussi actif dans d'autres environnements critiques tels que le contrôle aérien, les solutions de paiement, ou les systèmes de contrôle de centrales



Figure 4: Clients d'Atos

## II. Contexte du Project

Worldline est le leader européen et un acteur mondial de référence dans le secteur des paiements et des services transactionnels. En 2015, la filiale a généré un revenu de 1.227 milliards d'euros et elle emploie actuellement plus de 7 300 collaborateurs dans le monde entier.

Les activités de Worldline sont organisées autour de trois axes : Merchant Services & Terminals, Mobility & eTransactional Services, Financial Processing Software Licensing.



Figure 5 : Les Activités de Worldline

Worldline s'appuie sur une longue expérience pour exploiter au mieux le nombre croissant d'opportunités internationales que constitue le secteur des transactions et des paiements.

Elle tire parti de la base de clients Atos tout en s'implantant dans d'autres pays européens, tels que l'Autriche ou la Slovaquie, travaille également pour la quasi-totalité des 10 principales banques de Chine, et possède des clients sur des marchés émergents tels que l'Argentine, le Chili et l'Inde.

//A régler

Xenturion est un système de gestion basé sur le Web pour les terminaux et les applications de point de vente.

Il permet la gestion à distance des applications sur les terminaux dans le domaine sans intervention physique ou humaine supplémentaire. Il fournit des services de gestion pour les différentes familles de terminaux, tels que le XENTA, XENTISSIMO, XENTEO.

Xenturion a été spécialement conçu pour mettre à jour le parc complet du terminal de manière sécurisée et dans un court temps écoulé.

Des interfaces ouvertes, la stabilité et les performances sont ses atouts. Elle contribue à augmenter la productivité et l'efficacité quant au maintien d'un seul terminal, un groupe de terminal ou un parc de terminal complet. Il s'interface en douceur avec les systèmes hôtes existants, tels que les systèmes de gestion de la relation client, de la comptabilité ou des systèmes de facturation.

L'infrastructure matérielle du serveur, cependant, détermine directement le nombre de connexions simultanées des terminaux que le système peut gérer.

Il est exploité par un gestionnaire de terminal pour soutenir un fournisseur de terminal dans son rôle contractuel envers ses clients ou commerçants à installer et à maintenir la fonctionnalité demandée, au moyen d'outils à distance. Les services offerts sont conformes aux lignes directrices des acquéreurs et sont conçus par les fournisseurs d'applications sous la forme d'applications.

Ces applications sont installées sur le matériel terminal qui est conçu par les fabricants de terminaux. Les caractéristiques de Xenturion sont :

- Portabilité (indépendance de la plateforme).
- Performance
- Parc des terminaux : jusqu'à 500.000 terminaux
- 400 connexions simultanées
- Facilité d'utilisation et l'interaction utilisateur Simple interface graphique basée sur le Web avec des profils d'utilisateurs différents.
- Maintenabilité Une solution de Java pur (robuste, portable, flexible, productivité élevée).
- Extensibilité

- Interopérabilité Interface ouverte avec des systèmes externes.
- Pas de frais de licence commercial

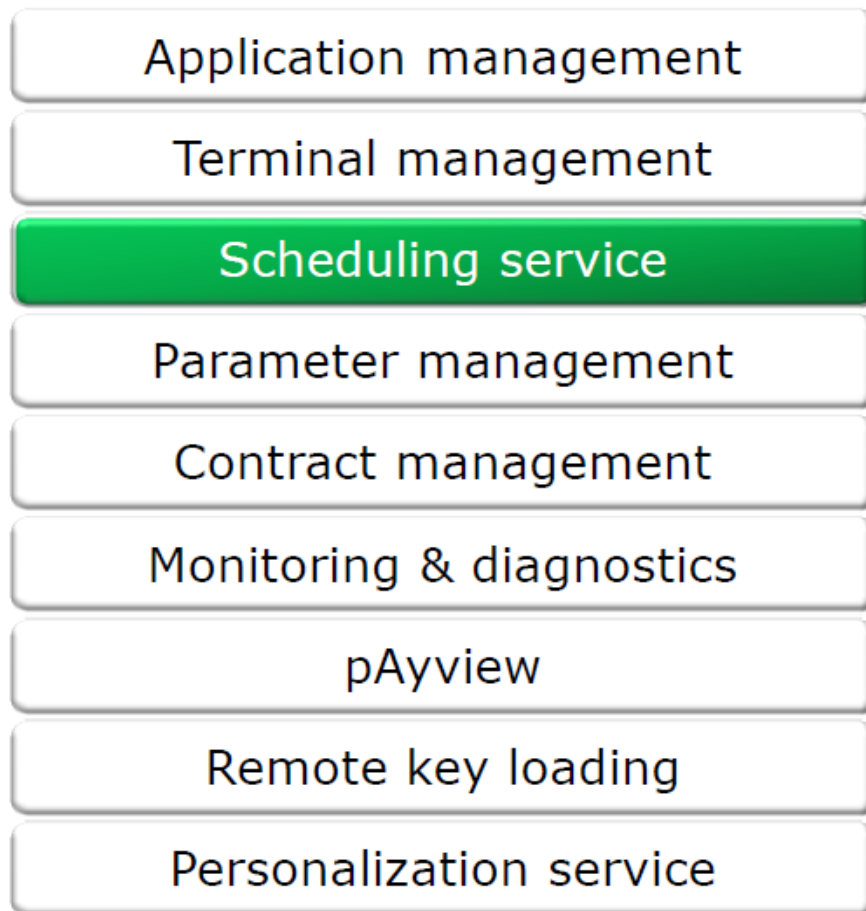


Figure 6 : Les services de Xenturion

Sur Xenturion, le module Scheduling service figure parmi les modules importants, c'est le module où se tourne le thème du Projet de Fin d'Etudes.

La fonction de scheduling ou bien planification est le lecteur pour toutes les autres fonctions de XENTURION. Une planification indique au terminal quand se reconnecter la prochaine fois et quelles actions effectuer.

Les terminaux n'effectueront aucune action sans horaire. À chaque connexion, le terminal télécharge d'abord le programme en cours, calculé par XENTURION (en fonction des entrées de l'utilisateur).

La planification se compose de deux parties : la planification du serveur et la planification du terminal.

- La planification du serveur définit les interfaces avec d'autres systèmes via un processus d'importation / exportation par lots. Tous les autres services XENTURION, par ex. La paramétrisation ou le téléchargement sont planifiés et gérés côté serveur. Le serveur est le maître du service d'ordonnancement.

- La planification du terminal est un outil permettant de définir l'heure et la fréquence auxquelles un terminal se connecte à XENTURION. Un point d'interaction et son terminal associé ont quatre tables de programmation:
  - **Schedule prévu** : un ensemble d'horaires futurs. Ceux-ci peuvent être édités par l'opérateur (ajouter / supprimer). Les horaires dans cette table seront envoyés au terminal lors de la prochaine connexion.
  - **Schedule connu par le terminal** : état actuel lié à la planification du service (lié au terminal). Ce tableau montre l'horaire réel du terminal.
  - **Schedule du terminal calculé** : les horaires sur le serveur sont envoyés au terminal, après que le terminal s'est connecté avec succès à XENTURION.
  - **Historique de la schedule** : journal des connexions au service de planification et leur statut.

Ce chapitre a introduit l'entreprise d'accueil ainsi le contexte du projet. Le chapitre suivant aborde l'étude et la conception de besoin.

# **Chapitre 2:**

## **Analyse Fonctionnelle et Conception**

Ce chapitre est consacré à l'étude fonctionnelle du projet. Il commence par l'étude de l'existant et la présentation de la problématique puis la capture des besoins fonctionnels, et la solution proposée.

Enfin, il décrit le déroulement du projet en présentant le processus de développement adopté ainsi que sa planification

## **I. Etude de l'existant et présentation de la problématique**

En réalité, le scheduling est le driver pour toutes les autres fonctionnalités de XENTURION. Une planification indique au terminal quand se reconnecter la prochaine fois et quelles actions effectuées, dans notre cas l'action consiste à lancer les teleloads c'est-à-dire le téléchargement des fichiers pour assurer le bon fonctionnement du système. Les terminaux n'effectueront aucune action sans horaire. À chaque connexion, le terminal télécharge d'abord le programme en cours, calculé par XENTURION en fonction des entrées de l'utilisateur.

Le scheduling se compose de deux parties : le scheduling du serveur et le scheduling du terminal.

- Le scheduling du serveur définit les interfaces pour d'autres systèmes via un processus d'importation / exportation par lots. Tous les autres services XENTURION, par ex. Le teleload est planifié et géré côté serveur. Le serveur est le maître du service scheduling.
- Le scheduling du terminal est un outil de planification pour définir l'heure et la fréquence auxquelles un terminal se connecte à XENTURION pour faire un teleload.
- Le service de gestion des terminaux est utilisé pour configurer, organiser, surveiller et maintenir les points d'interaction qui est une identification logique qui fait référence à l'étendue des services offerts à un commerçant.

Dans le contexte de scheduling service, le projet Teleload Planning Tool est un nouveau besoin demandé par les clients d'Atos Worldline. En effet, aujourd'hui, il n'y a pas de visibilité sur la campagne de chargement, des synchronisations régulières sont nécessaires avec l'équipe pour éviter les surcharges et les conflits de téléchargement.

## **II. Capture des besoins fonctionnels**

Notre système Teleload Planning Tool, doit être opérationnel, évolutif, convivial et offrant les informations nécessaires en temps réel.

Pour ceci, le module à réaliser doit satisfaire les exigences utilisateurs.

Atos Worldline a besoin d'un outil pour la planification des teleloads, afin de :

- Diffuser les teleloads par site / client.
- Diffuser les teleloads pendant une période donnée, avec utilisation optimale des créneaux disponibles sur Xenturion (tableau de bord de surveillance de la capacité teleload)
- Automatisez la planification manuelle des teleloads effectuée aujourd'hui via Excel / Access. Cet objectif permettra de diminuer les efforts pour la préparation de la campagne de téléchargement et de diminuer le risque d'erreur dû aux conflits de teleload.

Dans le cadre du projet « Teleload Planning Tool », premièrement, il est demandé d'afficher un graphique montrant la charge du serveur prévue en termes de : quantité de teleload simultanés en fonction du temps, afin de donner une présentation significatives des données.

Deuxièmement, l'objectif est de proposer un algorithme qui aide à planifier d'une manière optimale les téléchargements afin de minimiser l'utilisation du matériel.

## 1. Exigences

- Pour la première partie,

Ils y a deux types de teleload :

- Ceux qui sont déjà planifiés (en bleu dans l'exemple ci-dessous)
- Ceux qui sont réservés (en rouge dans l'exemple ci-dessous)

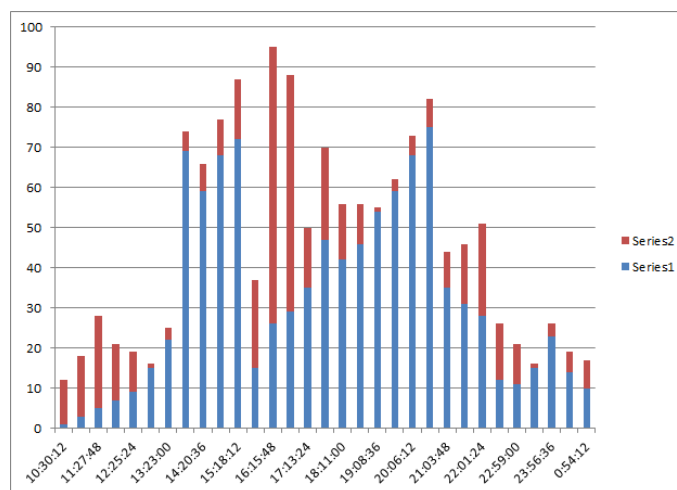


Figure 7 : Chart avec teleload planifié et reserved

L'axe des X représente l'heure et est déterminé par une date de début et une de fin. En effet, la date en totalité peut être affichée seulement quand elle change.

La précision du graphique est représentée par la quantité de "colonnes" affichée.

L'axe des Y représente la somme N de deux montants :



- **N<sub>p</sub>** : quantité de teleload simultanés déjà planifiés
- **N<sub>r</sub>** : quantité de teleload simultanés réservés
- **N = N<sub>p</sub> + N<sub>r</sub>**
- **N<sub>max</sub>** = le maximum théorique de N, exemple **N<sub>max</sub> = 90**, il est conseillé de visualiser une ligne horizontale montrant le **N<sub>max</sub>**

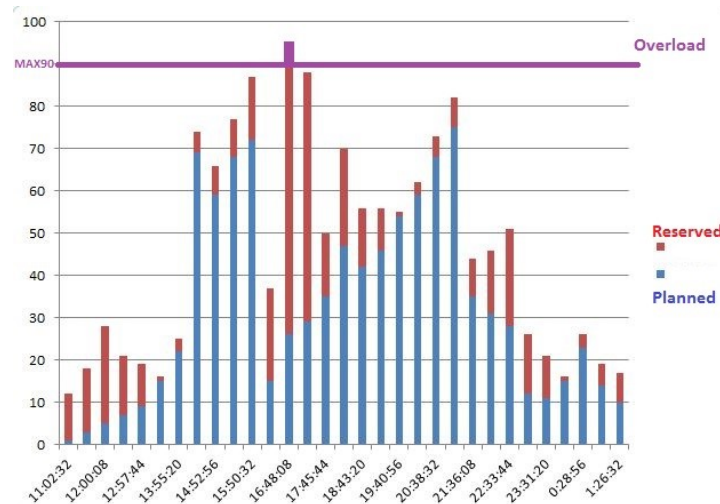


Figure 8 : Chart avec le max désigné

- Pour la deuxième partie, elle consiste à représenter les points of services d'une compagnie par des teleloads qui seront planifiés à base des durées calculé et du Service\_UTC afin de simuler des données existent réellement.

De plus le calcule doit se faire à base d'un algorithme qui permet d'optimiser l'utilisation des serveurs et des terminaux.

## 2. Solution proposée

Afin de répondre au besoin, et remédier les soucis qu'on a cité dans l'étude de l'existant, le travail était divisé en deux phases :

### Priorité 1

- Afficher un graphique basé sur un tableau de données fourni pour le mocking
  - En plus que la table Teleload Plannified, on crée une autre de Teleload Reserved sera utilisé pour la construction de notre chart graphique.
- De ce fait, c'est quoi le **Mocking** ?

En programmation orientée objet, les mocks sont des objets simulés qui reproduisent le comportement d'objets réels de manière contrôlée. Un programmeur crée un mock dans le but de tester le comportement d'autres objets, réels, mais liés à un objet inaccessible ou non implémenté. Ce dernier est alors remplacé par un mock.<sup>1</sup>

Ensuite, nous proposons d'ajouter deux tables en base de données :

- Le **POS\_SCHEDULING** actuel pour les teleloads planifiés
- Un nouveau **POS\_SCHEDULING\_RESERVED** pour les teleloads réservés

Les deux ont exactement la même structure:S

POID	JDOVERSION	POS_POID	SERVICE	REQUESTER	SERVICE_UTC	PERIOD	RETRY_ALGO	ACTION	DURATION	CAMPAIGN POID
									seconds	
944605	1	145	1	6	1347258300000	127	10001000600120000000000000000000	0	3600	15
944605	1	145	1	6	1347258300000	127	10001000600120000000000000000000	0	12000	24
944602	1	16	1	6	1347258180000	127	10001000600120000000000000000000	0	15000	63
944603	0	16	2	6	1347258480000	1440	2400018001800180000000000000000000	0	3000	658
947052	0	5455	1	6	1357587585000	1440	2400001000100060000000000000000000	0	5486	95478
944176	0	5226	1	5	1346263213200	0	10001000600120000000000000000000	0	85000	52
944177	0	5227	1	5	1346263213800	0	10001000600120000000000000000000	0	9000	14
944178	0	5228	1	5	1346263214400	0	10001000600120000000000000000000	0	5412	15
944179	0	5229	1	5	1346263215000	0	10001000600120000000000000000000	0	25690	12

Figure 9 : Structure de POS\_SCHEDULING et POS\_SCHEDULING\_RESERVED

Pour les deux tables les teleloads qui nous intéressent, sont ceux qui contiennent les critères suivants :

**SERVICE = 3** : Signifie un teleload

**PERIOD = 0** : Un teleload non récurrent

**ACTION = 0** : L'action est une énumération sa valeur soit **Add (0)** soit **Delete (1)**, pour notre cas celle avec Add que nous choisissons.

En principe, après avoir sélectionné les teleloads qui répondent aux critères ci-dessus, on donne la main à l'utilisateur pour nous fournir des informations selon son besoin :

**T\_Start** : qui signifie la date de début d'apparition des teleloads planifié et reserved dans la charte.

**T\_Stop** : qui signifie la date de fin d'apparition des teleloads planifié et reserved dans la charte.

**P** : qui désigne la précision c'est-à-dire le nombre de colonnes que l'utilisateur veut visualiser.

Finalement, on se lance dans notre algorithme pour calculer le nombre de teleloads planifié et reserved pour chaque instant dans la chart en assurant le critère de la performance.

<sup>1</sup> [https://fr.wikipedia.org/wiki/Mock\\_\(programmation\\_orient%C3%A9e\\_objet\)](https://fr.wikipedia.org/wiki/Mock_(programmation_orient%C3%A9e_objet))

## Priorité 2

Pour la deuxième partie, nous avons proposé d'alimenter la base de données Xenturion avec une table de plus qui désigne la campagne, cette dernière contient les Point of Services qu'un utilisateur veut planifier ses teleloads. En effet, une campagne aura un cycle de vie comme suite :

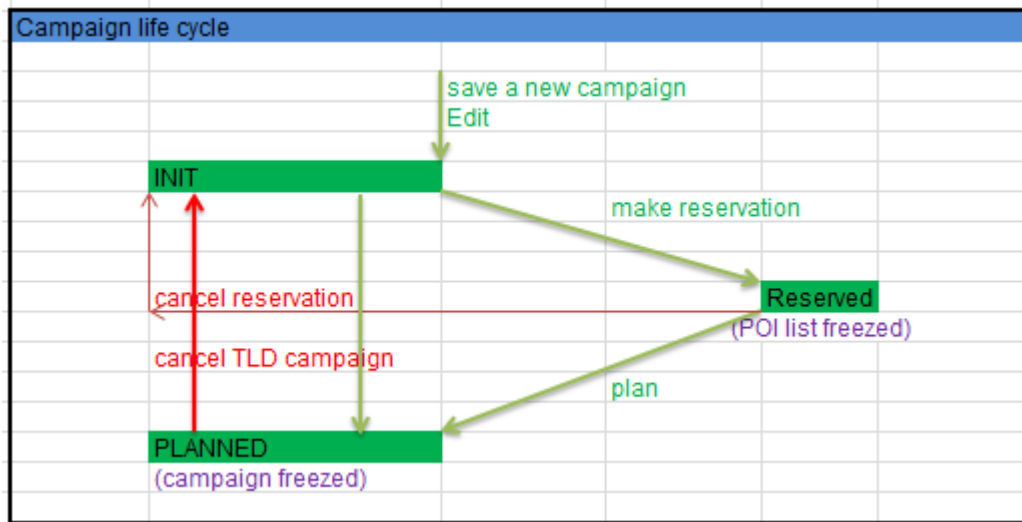


Figure 10 : Cycle de vie d'une campagne

Et une structure qui contient différents champs pour assurer le bon fonctionnement du système, par exemple : il faut choisir une date qui est autorisée lors de la création de la campagne, en plus un state qui décrit l'état de la campagne INIT à la création, RESERVED à la réservation et PLANNIFIED à la planification, aussi garder la trace du dernier teleload lancé...

campaign list			
name	State	last TLD end time	Id
delhaize 10/2016	INIT		245
toto 456	RESERVED	22/06/2017 17:05	123
titi 748	PLANNED	12/01/2017 10:05	4587
toto 821	PLANNED	11/04/2017 17:05	9314

Figure 11 : Structure de la campagne

campaign	
POI list	
IMPORT	
	ADD
DELETE Selected	
CLEAR	
<div> <div>12345681</div> <div>12345684</div> <div>12345687</div> <div>12345690</div> <div>12345693</div> <div>12345696</div> <div>12345699</div> <div>12345702</div> <div>12345705</div> <div>12345708</div> <div>12345711</div> <div>12345714</div> <div>12345717</div> </div>	

Figure 12 : Points of services d'une campagne

Après avoir préparé la campagne et ses points of services, il est temps à proposer un algorithme optimal pour la planification de teleloads de cette dernière.

Premièrement, il est demandé de calculer la durée de chaque teleload à base des attributs assignés au point de service (POS) et son terminal, en effet, nous sommes intéressés aux attributs TYPE pour le POS et MEDIA pour le terminal.

TLD duration				
	TYPE	MEDIA	default = 1h	MEDIA => Terminal
	2	ethernet	0,25	TYPE => POS
	3	GSM	4	
	3	ethernet	0,3	minimal duration = 5min

Figure 13 : Durée de teleload

Pour les POSs sans attributs leur durée par défaut est 1h, et nous prenons en considération que la durée minimale est de 5min.

Finalement, pour le calcul de SERVICE\_UTC qui désigne l'instant de lancement d'un teleload, nous proposons d'utiliser la fusion des algorithmes optimaux.

Puisque nous devons respecter la date début et date fin de la campagne nous choisissons l'algorithme des théories des graphes, ensuite, nos teleloads doivent se présenter d'une manière successive de plus grand au plus petit pour l'équilibrage de charge de serveurs, l'algorithme qui assure l'approximation linéaire de first fit decreasing bin packing est choisi.

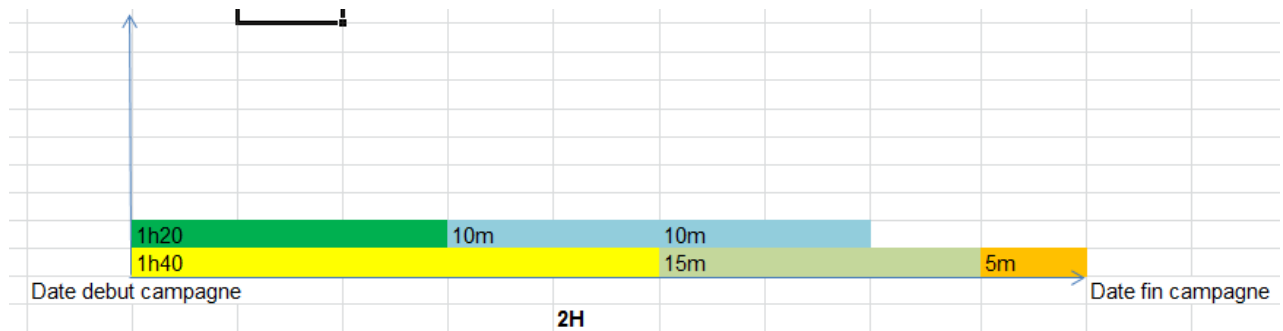


Figure 14: Algorithmme proposé pour Service\_UTC

### 3. Conduite du projet

Avant de passer à la modélisation et la conception du projet, la gestion de projet, est une démarche qui a pour but de structurer, assurer et optimiser le bon déroulement du projet.

Dans le périmètre du projet Xenturion, l'agilité est l'une des critères indispensables pour l'avancement du projet. En effet, la méthode Scrum est dédiée pour ce besoin.

De ce fait, Cette méthode de gestion, ou plutôt ce framework de management de projet, a pour objectif d'améliorer la productivité de son équipe

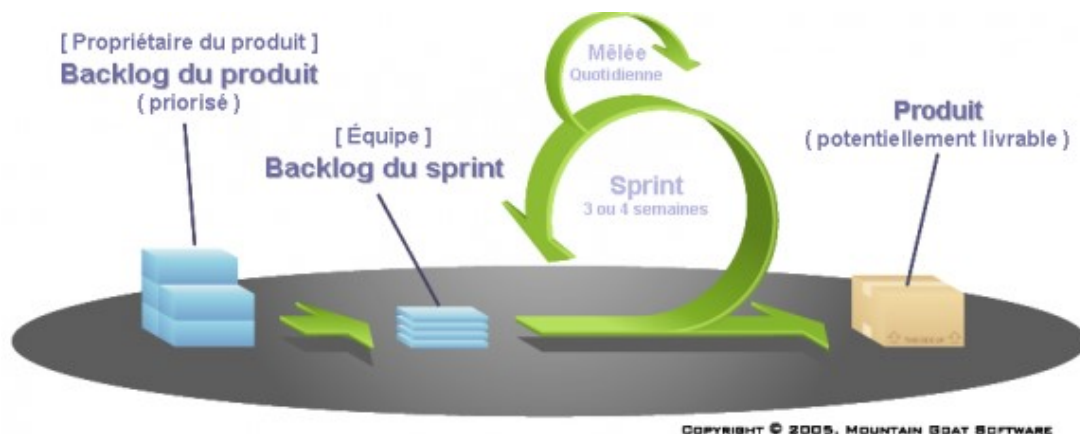


Figure 15 : Principe du SCRUM

2

### 4. Méthode Scrum

#### *Le Scrum Master*

- *S'assure que les principes et les valeurs de Scrum sont respectés*
- *Facilite la communication au sein de l'équipe*
- *Cherche à améliorer la productivité et le savoir-faire de son équipe*

## ***L'équipe***

- *Pas de rôle bien déterminé : architecte, développeur, testeur*
- *Tous les membres de l'équipe apportent leur savoir-faire pour accomplir les tâches*
- *Taille de 6 à 10 personnes en général et pouvant aller jusqu'à 200 personnes*

## ***Le Product Owner***

- *Expert métier, définit les spécifications fonctionnelles*
- *Etablit la priorité des fonctionnalités à développer ou corriger*
- *Valide les fonctionnalités développées*
- *Joue le rôle du client*

## ***Les sprints***

*Le cycle de vie Scrum est rythmé par des itérations de quelques semaines.*

## ***Le product backlog***

*Le référentiel des exigences initiales est dressé et hiérarchisé avec le client. Il ne doit pas nécessairement contenir toutes les fonctionnalités attendues dès*

*Le début du projet, il va évoluer durant le projet en parallèle des besoins du client.*

## ***User Story***

*Les fonctionnalités décrites portent le nom d'User Stories et sont décrites en employant la terminologie utilisée par le client.*

*Une User Story ou Story contient généralement un identifiant unique contient les informations descriptives de la fonctionnalité attendue par le client avec une priorité et une estimation qui changent au fur et à mesure*

## ***Le sprint planning meeting***

*On organise, avant chaque sprint, une réunion de planification, le sprint planning meeting. Ce planning sélectionne dans le product backlog les exigences les plus prioritaires pour le client. Elles seront développées, testées et livrées au client à la fin du sprint. Elles constituent le sprint backlog, un sous ensemble du product backlog.*

## ***La mêlée***

*Au cours du sprint, il est organisé, chaque jour, une réunion d'avancement (environ 15 min) avec tous les membres de l'équipe afin de s'assurer que les objectifs du sprint seront tenus, c'est le Scrum ou mêlée. Chaque jour, après la réunion Scrum, le Scrum Master maintient un graphique appelé sprint burndown chart. Ce graphique donne une très bonne vision de ce qui a été fait et du rythme de travail de l'équipe. Il permet également d'anticiper si toutes les stories du Sprint Backlog seront terminées à la fin de l'itération ou non.*

*La méthode Scrum n'a pas seulement un but informatif, mais aussi de stimuler l'esprit travail en équipe et le niveau d'engagement de chaque membre de l'équipe dans le projet.<sup>3</sup>*

## 5. Demarche DevOps

### Unir le développement et les opérations

L'objectif de DevOps est d'unir le développement et les opérations afin que la société dans son ensemble soit mieux en mesure de fournir des logiciels. Bien que ce soit l'objectif, lorsque vous le réduisez, DevOps est en fait sur l'empathie.

Compte tenu du fait que le logiciel est une source de revenus et qu'il permet de gérer et d'exploiter des entreprises, l'adage moderne veut que chaque entreprise soit une société de logiciels. En transférant une grande partie de ce que nous faisons vers le monde numérique, nous sommes en mesure de gérer le changement et de créer de l'innovation beaucoup plus rapidement que jamais auparavant. Mais la condition préalable à tout cela est de donner à vos développeurs une piste pour construire et lancer du code qui fonctionne. DevOps est cette piste.

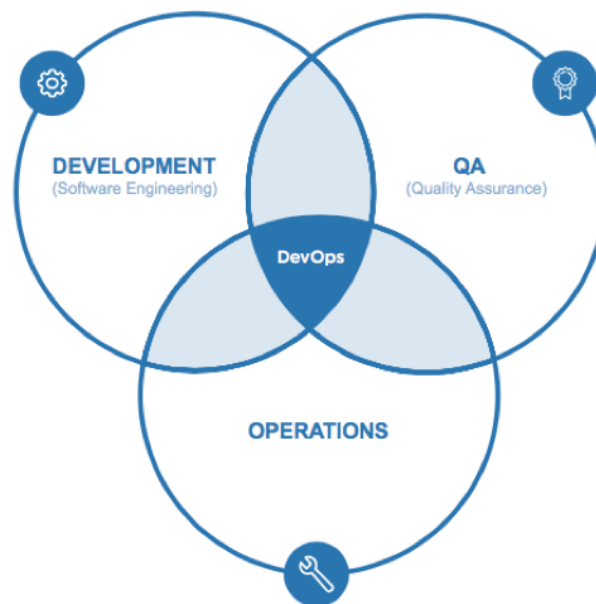


Figure 16 : Devops Evoluion

<sup>3</sup> <https://www.thierry-pigot.fr/scrum-en-moins-de-10-minutes/>

## Montée de tout en continu

Le processus de développement logiciel initial reflétait le processus de développement matériel : le concevoir, le construire, le tester, l'expédier. Malheureusement, ce processus ne se prête pas à la publication rapide de fixes de bogues, de correctifs et d'améliorations générales d'un logiciel. Au fil du temps, un processus plus efficace a commencé à évoluer. La montée en puissance de l'intégration, du déploiement et de la livraison continue a permis le déploiement rapide, le test et la promotion de nouvelles fonctionnalités dans la production.

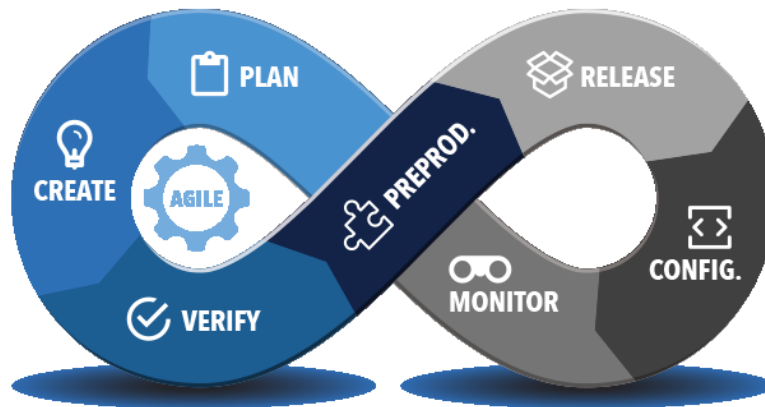


Figure 17 : Cycle de vie DevOps

## Passe vers Microservices

L'approche microservice encourage également les modules couplés de façon lâche, ce qui réduit le risque de briser des fonctionnalités non liées tout en modifiant le code. En matière d'évolutivité et de clustering, les microservices vous permettent de croître plus naturellement sur la courbe de la demande.

## L'automatisation devient standard

Il va sans dire qu'un bon service informatique automatise les tâches répétitives. Ironiquement, de nombreuses équipes se retrouvent tellement occupées à automatiser les tâches métier qu'elles oublient d'automatiser les tâches techniques. C'est pourquoi l'automatisation doit être considérée de manière proactive. Il devrait être déplacé de la liste "nice to have" à la liste des exigences. L'automatisation réduit le risque d'erreur humaine et, par conséquent, les équipes sont moins susceptibles de manquer une étape lors du déploiement de modifications dans un centre de données.

## Au-delà des DevOps : où va la technologie ?

Lorsque vous prenez du recul et observez comment la technologie évolue, vous remarquerez deux choses : Chaque coin de notre vie devient numérique et le taux de changement technologique augmente de façon exponentielle. Cela soulève une question :



## **Comment se fait-il que ce taux de changement puisse être maintenu sans que les roues ne tombent du bus proverbial ?**

Premièrement, le changement dans le monde numérique a beaucoup moins d'obstacles que le monde physique. Il est plus facile de trouver une idée, de la coder et de la tester que de concevoir et de créer un prototype. Mais là où la technologie a vraiment brillé, c'est le mariage d'un état d'esprit de changement et de structure.

L'une des composantes fondamentales de ce mariage est **Agile**, qui formalise l'itération rapide du travail tout en étant sensible aux réactions et aux changements. Viennent ensuite les microservices et les conteneurs qui nous permettent de découper une grande application en petites pièces autonomes et de déployer des logiciels de manière fiable avec un minimum de support, car toutes les dépendances sont regroupées. Et puis il y a DevOps, qui combine outils, culture et méthodologie pour rapprocher deux forces diamétralement opposées du développement et des opérations informatiques pour une plus grande efficacité dans la production et le maintien du changement technologique.

Les initiatives indépendantes d'Agile, des microservices, des conteneurs et des DevOps s'alignent parfaitement pour créer un écosystème qui soutient l'innovation technologique à grande échelle.

Dans ce chapitre, nous avons présenté qu'aucune solution n'est eu lieu auparavant, et procédé directement à l'étude des exigences avant de présenter la solution dans un cadre agile.

# **Chapitre 3:**

## Étude technique

Le choix de l'architecture technique est un facteur déterminant dans la performance de l'application, il dépend de plusieurs contraintes techniques que nous allons étudier en ce qui suit. De plus, les algorithmes et la conception adoptée du choix des diagrammes et les outils utilisés pour mettre en place la solution proposée.

## I. L'architecture du projet Xenturion

L'architecture de l'application est une architecture en couches. En effet, un système fondé sur une architecture en couches offre, sur différents niveaux, les avantages suivants :

- **Le développement** : Du fait que l'application est découpée en couches indépendantes, le temps consacré au développement est diminué, car chaque couche peut être programmée parallèlement et par de nombreux programmeurs
- **Les tests** : Puisque chaque couche est dotée du caractère d'indépendance vis-à-vis des couches adjacentes, chacune d'elles pourrait être testée indépendamment des autres.
- **La maintenance** : Si un problème a été détecté, on peut facilement savoir la couche qui présente une anomalie.
- **Les amendements de l'application** : la modification ou la mise à jour d'une couche n'affecte pas les autres couches.

De plus, Les spécifications techniques, élaborées à ce stade, sont de nature :

- **Ergonomie** : Augmenter l'agilité, la simplicité, le dynamisme et l'interactivité des vues, est un besoin fondateur de l'application, ce qu'on peut concrétiser via les différentes librairies du design.
- **Organisationnel** : Pour avoir une architecture robuste, modulable et évolutive, il faut utiliser le principe de « couche », et donc séparer au maximum les différents types de traitements de l'application dans des packages spécifiques pour chaque traitement.
- **Performance** : L'utilisation des web services, permet de déminer la charge sur le côté client, un grand nombre d'actions effectuées sur le serveur tel que l'envoi de la page, la récupération des données, leur (pré) validation, les données sont envoyées aux clients Android sous forme de JSON, qui se limite de la représentation des données.
- **Sécurité** : Les utilisateurs qui veulent utiliser les différentes fonctionnalités de l'application doivent être authentifiés et d'avoir les différentes permissions, et pour les web services, Xenturion utilise la méthode de Basic Authentication, envoyer le nom de l'utilisateur, et le mot de passe dans la requête HTTP, dans chaque appel d'un web service.

## 1. Architecture Java EE

Le projet Xenturion est construit sur la plateforme Java Enterprise Edition (Java EE) principalement et d'autre Frameworks comme ZK.

Un atout majeur de la technologie choisie est que le système est techniquement indépendant de l'infrastructure matérielle.

### Architecture Java EE :

Java EE est une plateforme fortement orientée serveur et dédiée au développement à l'exécution d'applications distribuées. Elle permet la simplification du processus de développement. En effet, Java EE simplifie le contrôle et la gestion des ressources système en fournissant des méthodes permettant de gérer les transactions et la mise en commun des ressources, ainsi, l'utilisation de Java EE pour développer et exécuter une application qui propose plusieurs avantages :

- Une architecture d'application basée sur les composants qui permet un découpage de l'application et donc une séparation des rôles lors du développement.
- La possibilité de s'interfacer avec le système d'information existant grâce à de nombreuses API : JDBC, JNDI, . . .
- La possibilité de choisir les outils de développement et/ou les serveurs d'applications utilisés qu'ils soient commerciaux ou libres.
- Une solution optimale pour développer d'applications robustes, sécurisées et évolutives.

Dans la mesure où Java EE s'appuie entièrement sur Java, il bénéficie des avantages de ce langage, en particulier une bonne portabilité et une maintenabilité du code.

De plus, l'architecture Java EE repose sur des composants distincts, interchangeables et distribués, ce qui signifie notamment :

- Qu'il est simple d'étendre l'architecture.
- Qu'un système reposant sur Java EE peut posséder des mécanismes de haute disponibilité, afin de garantir une bonne qualité de service.
- La maintenabilité des applications.
- Qu'il est possible de factoriser le code et utiliser des Frameworks ou composants génériques (gain de temps et de performances).

## 2. L'architecture de la solution

Fondamental pour les systèmes logiciels. Elle s'avère indispensable pour une bonne compréhension, la gestion et l'optimisation de tout système.

Pour structurer notre système, nous avons opté pour un découpage en couches. Ceci est également justifié par l'architecture Java EE adoptée.

L'architecture logicielle exprime un schéma d'organisation structurel :

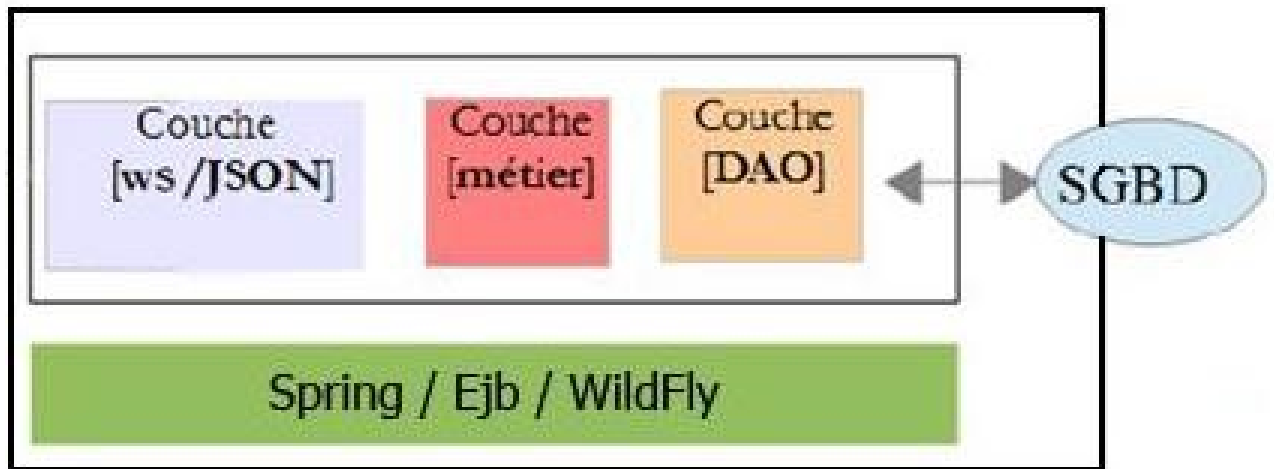


Figure 18:Architecture logicielle

L'architecture ci-dessus, représente la partie serveur qui est construite avec les Frameworks Spring et Ejb principalement et déployée avec WildFly. Cette partie est divisée en plusieurs couches :

- **Couche DAO** : C'est la couche qui regroupe les classes Java capables de fournir les données nécessaires à la couche métier, et par la suite aux services demandés par l'utilisateur, cette couche est modélisée dans notre système par des composants DAO. Un DAO est un design pattern (patron de conception) utilisé dans les architectures logicielles objet visant à isoler la logique de persistance dans des classes d'accès aux données. Pour l'implémentation de cette couche, les Frameworks JPA et Hibernante sont utilisées pour mapper les données.
- **Couche Métier** : Cette couche regroupe les classes Java nécessaires au logique métier, elle comporte donc les services offerts par le système à l'utilisateur. Ces services ne sont autres que l'implémentation des scénarios identifiés lors de la phase de conception. Lors de l'appel d'un service, ce dernier fait appel aux objets métier nécessaires pour répondre à la requête de l'utilisateur. Cette couche est réalisée par EJB.
- **Couche WS/JSON** : La couche Web service est principalement constituée de méthodes qui traitent les web services répondent avec une ligne de texte au format JSON (Javascript Object Notation).

Et finalement, la partie réalisée avec Spring MVC qui représente une implémente du modèle d'architecture **Modèle – Vue – Contrôleur** de la façon suivante :

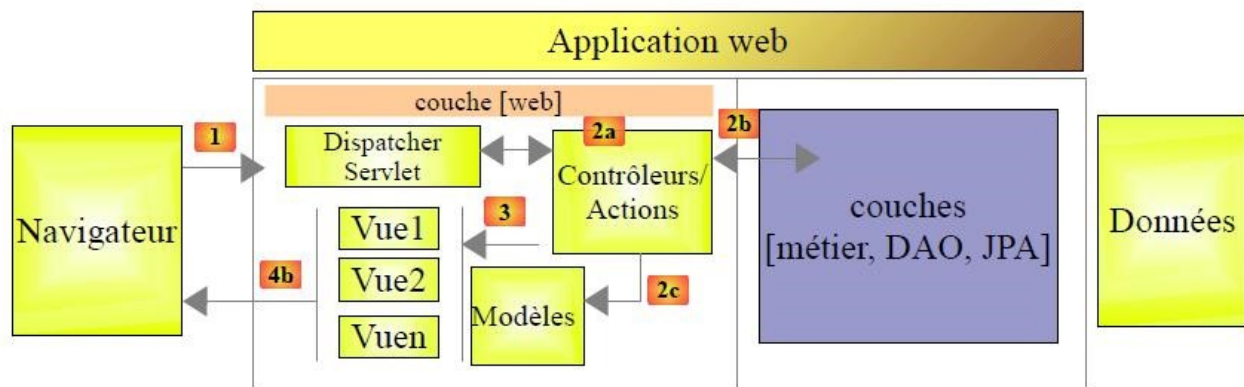


Figure 19 : Application et modèle MVC

Le traitement d'une demande d'un client se déroule de la façon suivante :

— **Demande** : Les URL demandées sont de la forme :

http ://machine :port/contexte/Action/param1/param2/.... ?p1=v1&p2=v2&...

La [Dispatcher Servlet] est la classe de Spring qui traite les URL entrantes. Elle "route" l'URL vers l'action qui doit la traiter. Ces actions sont des méthodes de classes particulières appelées [Contrôleurs]. Le C de MVC est ici la chaîne [Dispatcher Servlet, Contrôleur, Action]. Si aucune action n'a été configurée pour traiter l'URL entrante, la servlet [Dispatcher Servlet] répondra que l'URL demandée n'a pas été trouvée (erreur 404 NOT FOUND [8])

— **Traitement** :

L'action choisie peut exploiter les paramètres parami que la servlet [Dispatcher Servlet] lui a transmis. Ceux-ci peuvent provenir de plusieurs sources :

— du chemin [/param1/param2/...] de l'URL,

— des paramètres [p1=v1&p2=v2] de l'URL,

— de paramètres postés par le navigateur avec sa demande, dans le traitement de la demande de l'utilisateur, l'action peut avoir besoin de la couche [metier] [2b]. Une fois la demande du client traitée, celle-ci peut appeler diverses réponses. Un exemple classique est :

- Une page d'erreur si la demande n'a pu être traitée correctement.
- Une page de confirmation sinon l'action demande à une certaine vue de s'afficher [3]. Cette vue va afficher des données qu'on appelle le modèle de la vue. C'est le M de MVC. L'action va créer ce modèle M [2c] et demander à une vue V de s'afficher [3].

— **Réponse** : la vue V choisie utilise le modèle M construit par l'action pour initialiser les parties dynamiques de la réponse HTML qu'elle doit envoyer au client puis envoie cette réponse.

### 3. Conception

#### a. Diagramme de cas d'utilisation

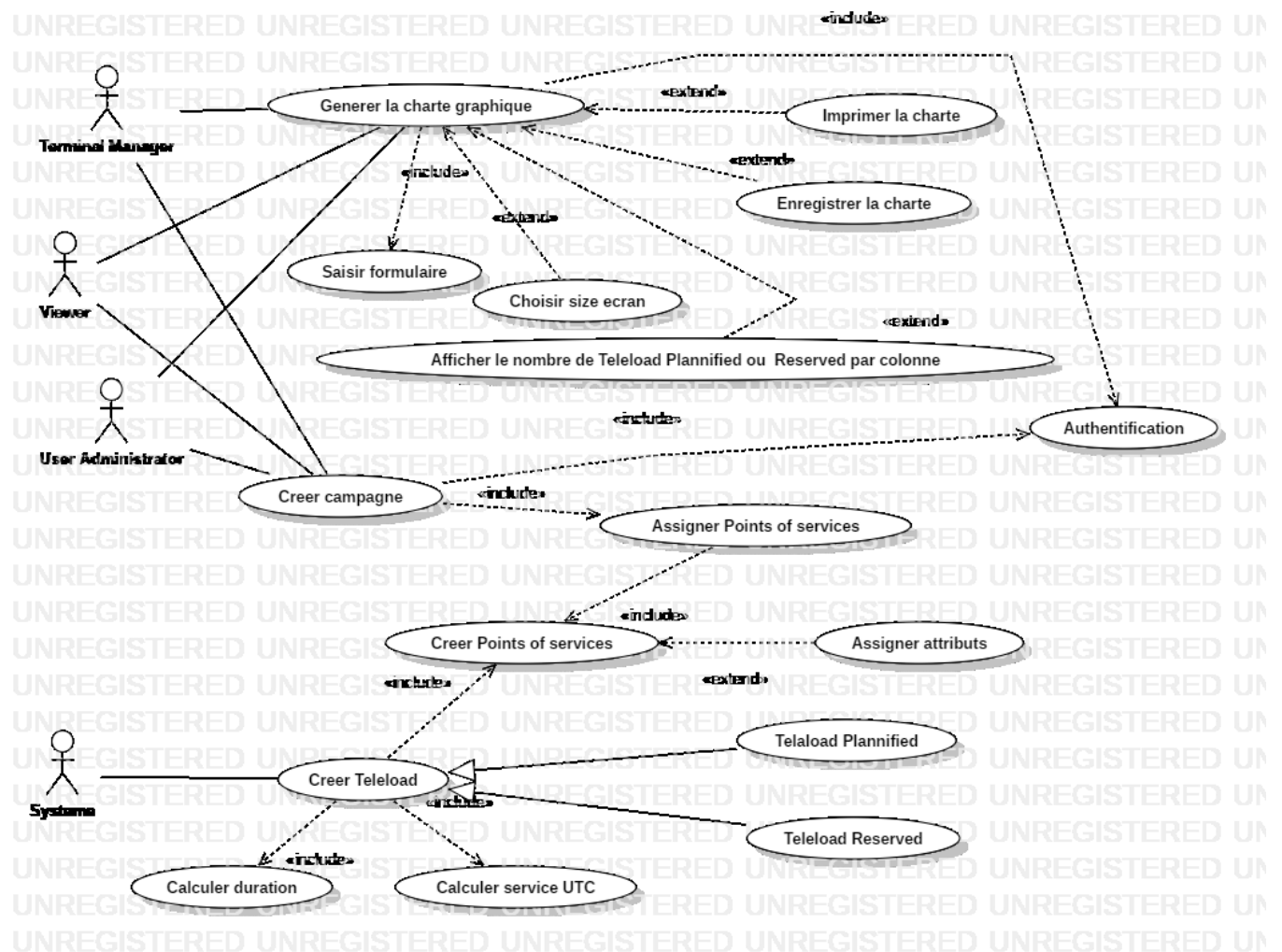


Figure 20 : Diagramme de cas d'utilisation

#### b. Diagramme de séquence

##### Description textuelle de haut niveau :

**Nom :** Générer la charte graphique.

**Résumé :** Pour chaque utilisateur, il suffit de saisir le formulaire qui contient la date de début, la date de fin et la précision pour afficher la planification de teleloads reserved et plannified.

**Acteur principale :** Terminal Manager, Viewer et User Administrator

**Acteur secondaire :** ----

**Déclencheur :** Saisir formulaire

**Terminaison :** Affichage de la charte

### **Description textuelle de bas niveau :**

#### 1) Scénario nominal:

Actions acteurs	Actions système
<u>Action 1 :</u> L'utilisateur saisit le formulaire	
	<u>Action 2 :</u> Le système enregistre les données saisis et fait le traitement
<u>Action 3 :</u> L'utilisateur clique sur le bouton "Generate chart"	
	<u>Action 4 :</u> Le système génère la charte

#### 2) Enchaînement alternatifs et erreurs :

Erreur 1 : S'il manque l'un des champs de formulaire, date de début, date de fin ou précision

- L'enchaînement commence à l'action 1
- Le système affiche un message d'erreur
- L'utilisateur ressaisi les données du formulaire

Alternatif 1 : Si la charte est affichée

- L'enchaînement commence à l'action 4
- Le système propose des fonctionnalités supplémentaires :
  - Afficher le nombre de teleloads par colonnes
  - Choisir size de l'écran
  - Zoomer la charte
  - Imprimer la charte
  - Enregistrer la charte sous format png, jpeg ...
- Le système réagit selon le choix de l'utilisateur

### **Présentation diagramme**



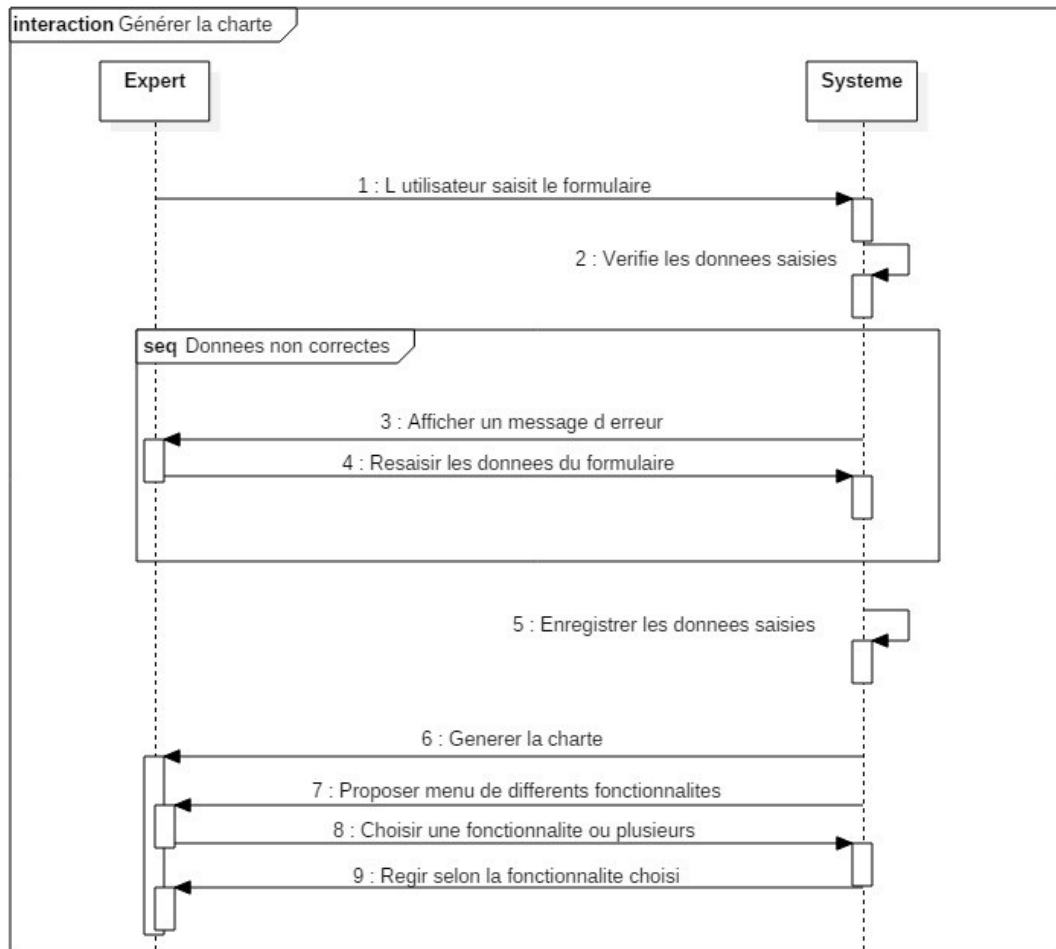


Figure 21 : Génération de la charte

### Description textuelle de haut niveau :

**Nom :** Créer les points of services

**Résumé :** Pour chaque utilisateur, //a regler

**Acteur principale :** Terminal Manager, Viewer et User Administrator

**Acteur secondaire :** ----

**Déclencheur :** créer point of service

**Terminaison :** Valider la création

### Description textuelle de bas niveau :

3) Scénario nominal:

Actions acteurs	Actions système
<u>Action 1 :</u> L'utilisateur clique sur new point of service	

	<u>Action 2 :</u> Le système affiche l'interface de la création des points of services
<u>Action 3 :</u> L'utilisateur saisit les informations nécessaires pour un POS	
<u>Action 4 :</u> L'utilisateur clique sur enregistrer	
	<u>Action 5 :</u> Le système enregistre le POS

#### 4) Enchaînement alternatifs et erreurs :

Erreur 1 : S'il manque l'un des champs nécessaires pour un POS

- L'enchaînement commence à l'action 2
- Le système affiche un message d'erreur
- L'utilisateur ressaisi les données

Alternatif 1 : Si l'utilisateur choisi d'assigner des attributs au POS

- L'enchaînement commence à l'action 3
- Le système propose deux types d'attributs : TYPE et MEDIA chacun avec différentes valeurs
- Le système enregistre le POS

### **Présentation diagramme**

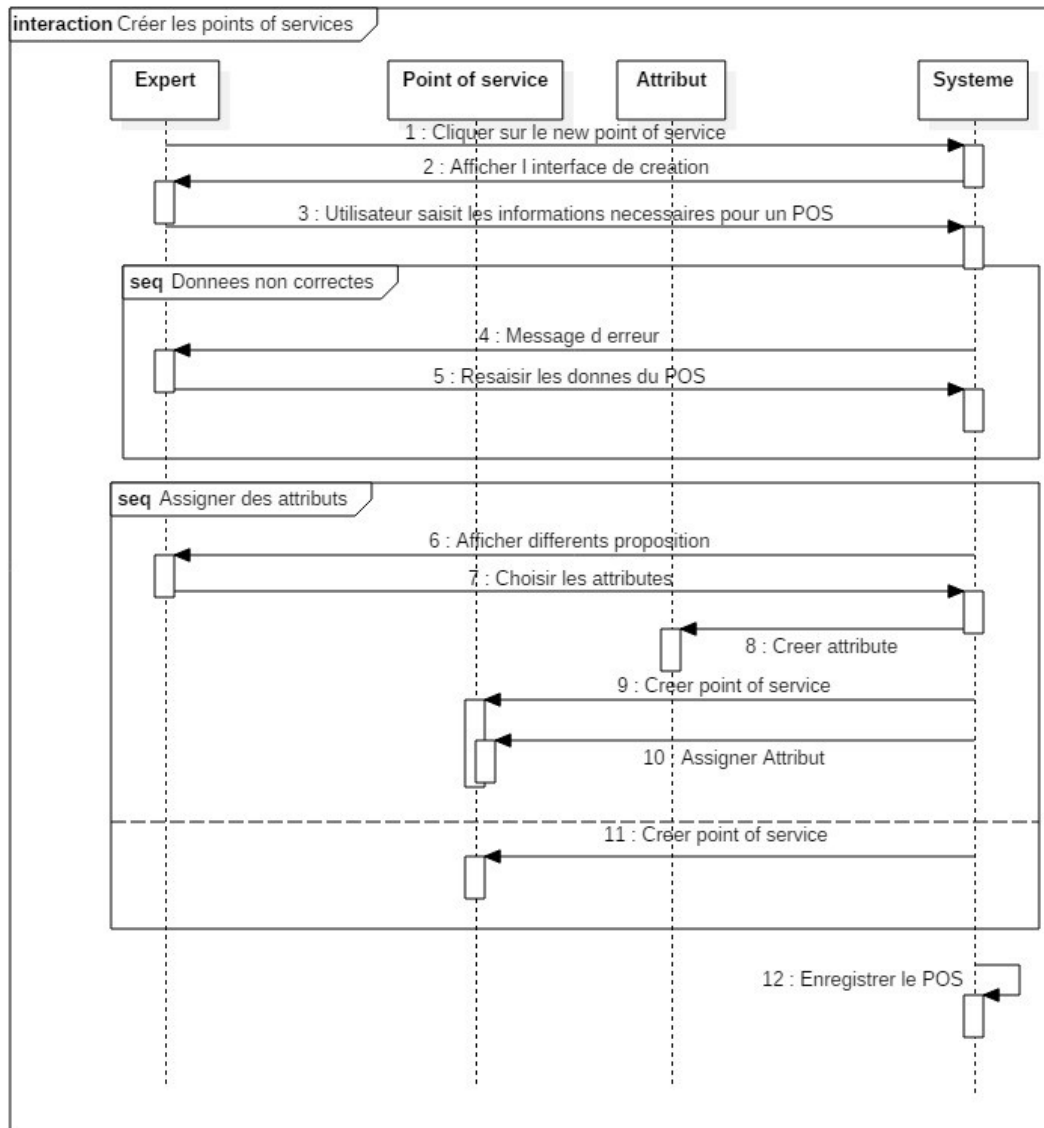


Figure 22 : Créer point of service

### **Description textuelle de haut niveau :**

**Nom :** Créer la campagne

**Résumé :** Après avoir créé l'ensemble de POSs, il est temps de les regrouper dans une campagne afin les planifier.

**Acteur principale :** Terminal Manager, Viewer et User Administrator

**Acteur secondaire :** ----

**Déclencheur :** Créer la campagne

**Terminaison :** Valider la création de campagne

### **Description textuelle de bas niveau :**

5) Scénario nominal:

Actions acteurs	Actions système
<u>Action 1 :</u> L'utilisateur choisit de créer une campagne	
	<u>Action 2 :</u> Le système affiche la page de création de la campagne
<u>Action 3 :</u> L'utilisateur saisit les données pour créer une campagne	
<u>Action 4 :</u> L'utilisateur clique sur enregistrer	
	<u>Action 5 :</u> Le système enregistre la campagne et indique la valeur INIT

6) Enchaînement alternatifs et erreurs :

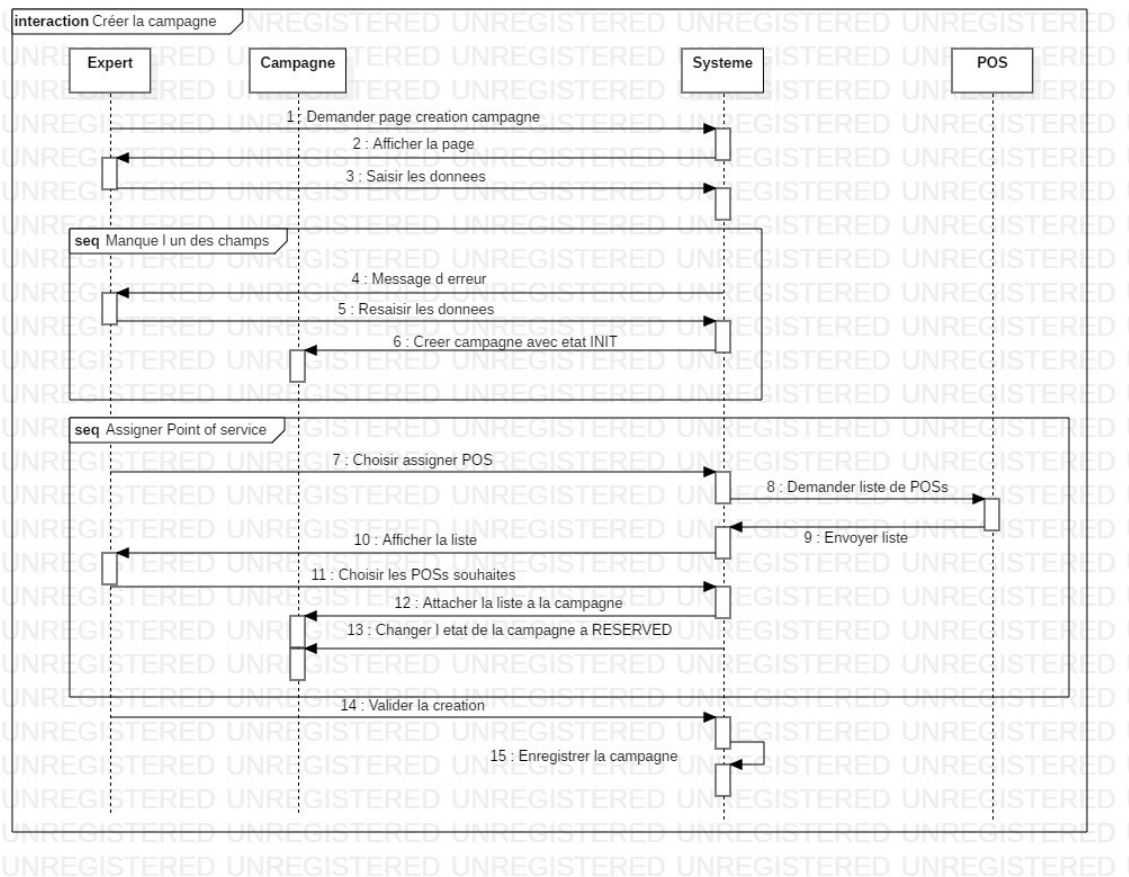
Erreur 1 : S'il manque l'un des champs de la campagne

- L'enchaînement commence à l'action 2
- Le système affiche un message d'erreur
- L'utilisateur ressaisi les données du formulaire

Alternatif 1 : Si l'utilisateur choisit d'assigner des POSs à la création de la campagne

- L'enchaînement commence à l'action 3
- Le système propose les POSs assigner à cet utilisateur pour faire son choix
- Le système change l'etat de la campagne à RESERVED.

**Présentation diagramme**



### **Description textuelle de haut niveau :**

**Nom :** Calculer duration

**Résumé :** Après avoir créé l'ensemble de POSs et les assigner à une campagne, il est temps d'utiliser les attributs TYPE et MEDIA s'il existe pour déterminer la durée.

**Acteur principale :** Système

**Acteur secondaire :** ----

**Déclencheur :** Création teleloads

**Terminaison :** Calcule de la durée à base de l'existence du TYPE et MEDIA

### **Description textuelle de bas niveau :**

#### 7) Scénario nominal:

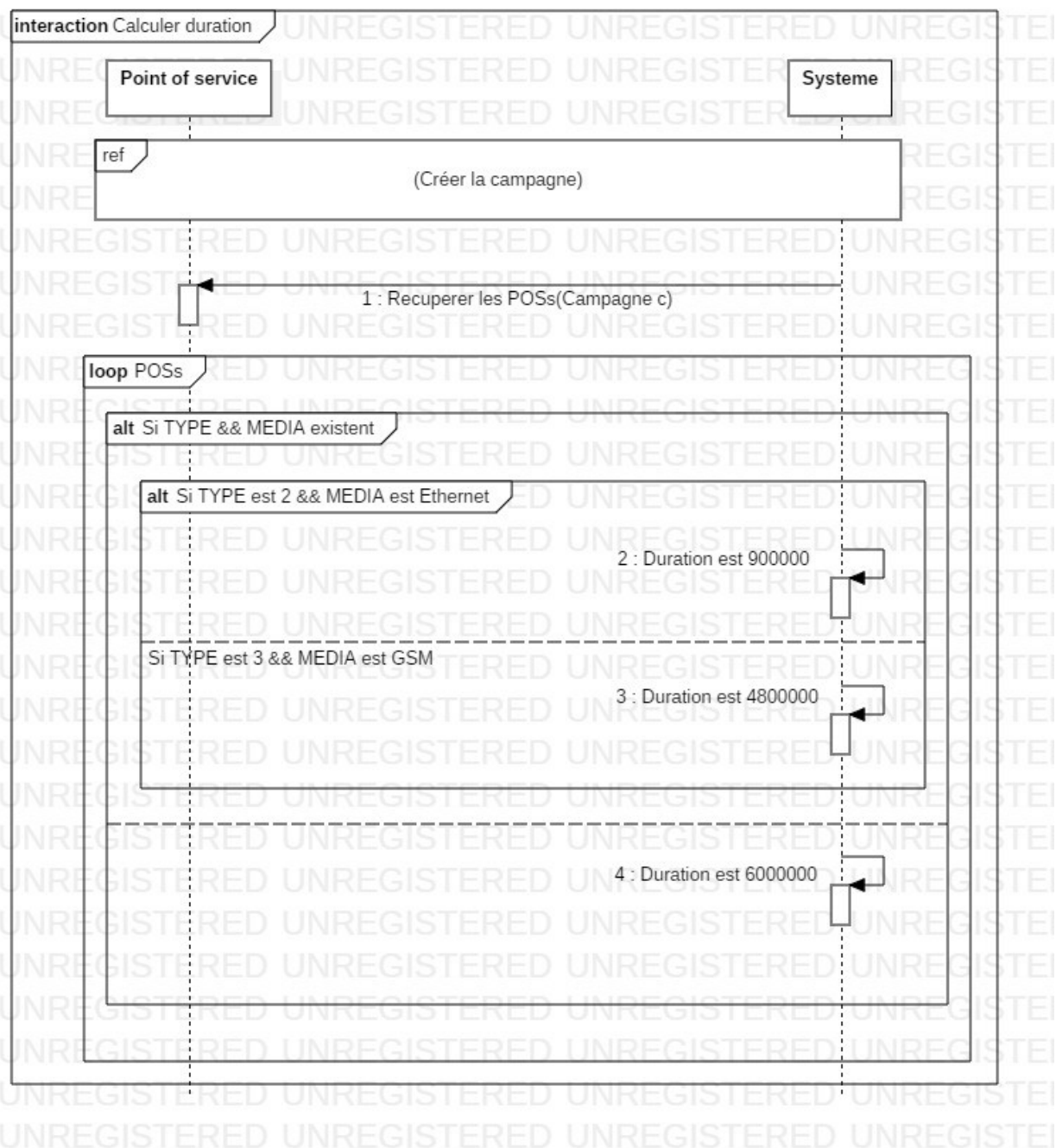
Actions système
<u>Action 1 :</u> Le système récupère la valeur du TYPE et MEDIA
<u>Action 2 :</u> Le système assigne une durée selon la valeur de TYPE et MEDIA

#### 8) Enchaînement alternatifs et erreurs :

Alternatif 1 : Si un POS ne contient ni TYPE ni MEDIA

- L'enchaînement commence à l'action 1
- Le système assigne par défaut 1h comme durée de teleload

### Présentation diagramme



### Description textuelle de haut niveau :

**Nom :** Calculer SERVICE\_UTC

**Résumé :** Après avoir calculé la durée, le système passe à assigner pour chaque teleload un SERVICE\_UTC basé sur cette durée.

**Acteur principale :** Système

**Acteur secondaire :** ----

**Déclencheur :** La création de la durée

**Terminaison :** SERVICE\_UTC calculer.

**Description textuelle de bas niveau :**

9) Scénario nominal:

Actions système
<u>Action 1 :</u> Le système récupère la valeur de la durée
<u>Action 2 :</u> Le système assigne la valeur SERVICE_UTC basé sur la durée calculé

**Description textuelle de haut niveau :**

**Nom :** Créer teleload Reserved

**Résumé :** Après avoir calculer la durée et le SERVICE\_UTC le système insert directement une ligne d'un nouveau teleload dans la base de données

**Acteur principale :** Système

**Acteur secondaire :** ----

**Déclencheur :** La création de la durée

**Terminaison :** SERVICE\_UTC calculer.

**Description textuelle de bas niveau :**

10) Scénario nominal:

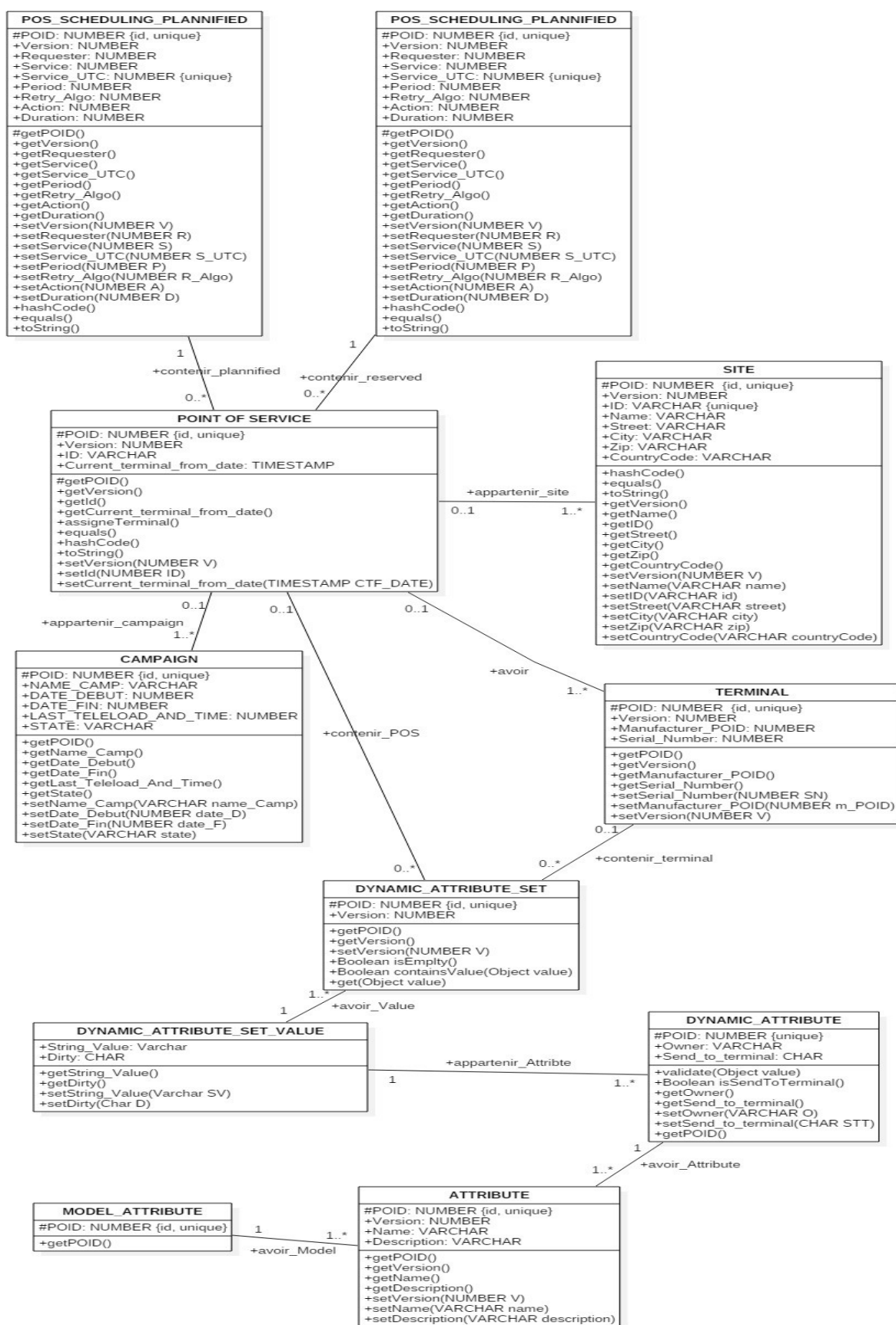
Actions système
<u>Action 1 :</u> Le système récupère la durée calculée plus la valeur SERVICE_UTC du dernier teleload Reserved inséré dans la base de données
<u>Action 2 :</u> Le système calcule le SERVICE_UTC pour le nouveau teleload Reserved

11) Enchaînement alternatifs et erreurs :

Alternatif 1 : Si aucun teleload Reserved est inséré dans la base de données

- L'enchaînement commence à l'action 2
- Le système assigne la valeur date début au SERVICE\_UTC
- Le système crée un nouveau teleload Reserved et change l'état de la campagne d'INIT à RESERVED

### c. Diagramme de classe





## 4. Améliorer les performances

*En informatique, les performances énoncent les indications chiffrées mesurant les possibilités maximales ou optimales d'un matériel, d'un logiciel, d'un système ou d'un procédé technique pour exécuter une tâche donnée.*

*Les performances sont établies par la mesure lors d'une activité appelée benchmarking grâce à des tests de performance. En effet, le test de performance consiste à définir le processus/système/matériel que l'on veut mesurer, à définir le cas d'utilisation à, et à exécuter le test tout en récoltant les mesures.*

*Les benchmarks et tests de performances sont souvent sujets à controverse. Ils sont souvent difficilement reproductibles, car très dépendants de détails non explicités : matériel utilisé, configuration de tous les éléments, charge de la machine <sup>4</sup>*

### a. Optimisation combinatoire

*L'optimisation combinatoire, aussi appelée optimisation discrète, est une branche de l'optimisation en mathématiques appliquées et en informatique, également liée à la recherche opérationnelle, l'algorithmique et la théorie de la complexité.*

*Dans sa forme la plus générale, un problème d'optimisation combinatoire (on dit aussi d'**optimisation discrète**) consiste à trouver dans un ensemble discret un parmi les meilleurs sous-ensembles (**ou solutions**) réalisables, la notion de meilleure solution étant définie par une fonction objective. Formellement, étant donnés :*

*L'ensemble des solutions réalisables ne saurait être décrit par une liste exhaustive car la difficulté réside ici précisément dans le fait que le nombre des solutions réalisables rend son énumération impossible, ou bien qu'il soit **NP-complet** de dire s'il en existe ou non.*

*Trouver une solution optimale dans un ensemble discret et fini est un problème facile en théorie : il suffit d'essayer toutes les solutions, et de comparer leurs qualités pour voir la meilleure. Cependant, en pratique, l'énumération de toutes les solutions peut prendre trop de temps ; or, le temps de recherche de la solution optimale est un facteur très important et c'est à cause de lui que les problèmes d'optimisation combinatoire sont réputés si difficiles.*

*La **théorie de la complexité** donne des outils pour mesurer ce temps de recherche. De plus, comme l'ensemble des solutions réalisables est défini de manière implicite, il est aussi parfois très difficile de trouver ne serait-ce qu'une solution réalisable.*

*Quelques problèmes d'optimisation combinatoire peuvent être résolus (de manière exacte) en temps polynomial par exemple par un **algorithme glouton**, un algorithme de **programmation dynamique** ou en montrant que le problème peut être formulé comme un problème d'**optimisation linéaire** en variables réelles.*

### b. Benchmarking

---

<sup>4</sup> [https://fr.wikipedia.org/wiki/Test\\_de\\_performance](https://fr.wikipedia.org/wiki/Test_de_performance)

Pour la période de 3 mois

La précision est 1000 colonnes, nous avons :

Echantillon 1:

Nombre d'élément dans la table :

POS\_SCHEDULING est 1.942.995  
POS\_SCHEDULING\_RESERVED est 1.946.505

La taille de la liste :

TLD Plannified est 25.556  
TLD Reserved est 25.790

Echantillon 2:

Nombre d'élément dans la table :

POS\_SCHEDULING est 2.542.779  
POS\_SCHEDULING\_RESERVED est 2.547.210

La taille de la liste :

TLD Plannified est 62.727  
TLD Reserved est 63.267

	DTO + JPQL + INDEX		DTO + JPQL		SEARCH + INDEX		SEARCH	
Echantillon	E1	E2	E1	E2	E1	E2	E1	E2
Temps d'exécution à la base de données	3.65 s	4.66s	3.77s	4.64 s	3.74s	5.53 s	4.53s	4.86s
Temps de remplissage de colonnes de la charte	0.16 s	0.99 s	0.36 s	0.99 s	0.93s	2.58 s	1.03s	3.39s
Total	3.81 s	5.65 s	4.14 s	5.64 s	4.67s	8.12 s	5.57s	8.26s

Dans la plupart des cas, le problème est **NP-difficile** et, pour le résoudre, il faut faire appel à des algorithmes de **branch and bound**, à l'**optimisation linéaire en nombres entiers** ou encore à la **programmation par contraintes**.

En pratique, la complexité physiquement acceptable n'est souvent que polynomiale. On se contente alors d'avoir une solution approchée au mieux, obtenue par une **heuristique** ou une **métaheuristique**.<sup>5</sup>

## Data Transfer Object Design Pattern

DTO est un conteneur de données pour déplacer des données entre des couches. Ils sont également appelés objets de transfert. DTO est uniquement utilisé pour transmettre des données et ne contient aucun logique métier. Ils ont seulement des setters simples et des getters.

Pourquoi faire un objet aussi simple que cela ? Parce qu'il fait un excellent conteneur de données, léger et fortement typé ainsi cela aidera à éviter le codage supplémentaire des plaques de la chaudière dans le développement de l'application.<sup>6</sup>

<sup>5</sup> Bellman, Richard (1957), Dynamic Programming, Princeton University Press. Dover paperback edition (2003), (ISBN 0-486-42809-5)

<sup>6</sup> <https://www.theserverside.com/discussions/thread/32389.html>

## **Indexing**

*Oracle inclut de nombreuses structures de données pour améliorer la vitesse des requêtes SQL Oracle. Profitant du faible coût du stockage sur disque, Oracle inclut de nombreux nouveaux algorithmes d'indexation qui augmentent considérablement la vitesse à laquelle les requêtes Oracle sont traitées.*

*Oracle utilise des index pour éviter le recours à des analyses de table complète, et à des tables complètes, qui sont nécessaires lorsque l'optimiseur SQL ne parvient pas à trouver un moyen efficace de traiter la requête SQL.<sup>7</sup>*

## **Hibernate search**

*Cela a un effet sur l'indexation et déclenche des performances de requête améliorées tant qu'Hibernate Search comprend qu'il peut utiliser la nouvelle stratégie de tri plus efficace.<sup>8</sup>*

*Hibernate Search intègre Apache Lucene, une bibliothèque de moteur de recherche en texte intégral extensible et haute performance écrite en Java. Cela combine la puissance de Lucene avec la simplicité d'Hibernate et de JPA.<sup>9</sup>*

## **JPQL**

*JPQL est utilisé pour effectuer des requêtes sur des entités stockées dans une base de données relationnelle. Il est fortement inspiré par SQL, et ses requêtes ressemblent aux requêtes SQL dans la syntaxe, mais opèrent contre les objets entité JPA plutôt que directement avec les tables de base de données.*

*JPQL est basé sur HQL (Hibernate Query Language), un langage de requête non standard antérieur inclus dans la bibliothèque de mappage relationnel objet Hibernate.<sup>10</sup>*

## **5. Optimization**

*En programmation informatique, l'optimisation de code est la pratique consistant à améliorer l'efficacité du code informatique d'un programme ou d'une librairie logicielle. Ces améliorations permettent généralement au programme résultant de s'exécuter plus rapidement, de prendre moins de place en mémoire, de limiter sa consommation de ressources (par exemple les fichiers), ou de consommer moins d'énergie électrique.*

---

<sup>7</sup> [http://www.dba-oracle.com/art\\_9i\\_indexing.htm](http://www.dba-oracle.com/art_9i_indexing.htm)

<sup>8</sup> <http://in.relation.to/2015/11/25/Hibernate-Search-5/>

<sup>9</sup> <http://www.baeldung.com/hibernate-search>

<sup>10</sup> [https://en.wikipedia.org/wiki/Java\\_Persistence\\_Query\\_Language](https://en.wikipedia.org/wiki/Java_Persistence_Query_Language)

## a. Théorie des graphes

*La théorie des graphes est la discipline mathématique et informatique qui étudie les graphes, lesquels sont des modèles abstraits de dessins de réseaux reliant des objets<sup>1</sup>. Ces modèles sont constitués par la donnée de « points », appelés nœuds ou sommets (en référence aux polyèdres), et de « liens » entre ces points ; ces liens sont souvent symétriques (les graphes sont alors dits non orientés) et sont appelés des arêtes.*

*Les algorithmes élaborés pour résoudre des problèmes concernant les objets de cette théorie ont de nombreuses applications dans tous les domaines liés à la notion de réseau (réseau social, réseau informatique, télécommunications, etc...*

*Un graphe est un ensemble de points nommés nœuds (parfois sommets ou cellules) reliés par des traits (segments) ou flèches nommées arêtes (ou liens ou arcs). L'ensemble des arêtes entre nœuds forme une figure similaire à un réseau. Différents types de réseaux sont étudiés suivant leur genre de forme (ou topologie) et propriétés ; les arbres sont une sous-catégorie plus simple de graphes particulièrement importante et qui est très étudiée, notamment en informatique.*

***Dans le cas général, un graphe peut avoir des arêtes multiples, c'est-à-dire que plusieurs arêtes différentes relient la même paire de points.***

*Il existe trois grandes familles de graphes et cinq catégories au total*

- *Structurés : il est alors possible de définir quatre identités topologiques remarquables :*
  - ***Homogènes (1)***: les sommets et les arêtes reproduisent un schéma régulier. Le schéma le plus commun est une architecture de type matriciel aussi appelé "en filet de poisson"
  - ***Hiérarchiques (2)*** : structure typique des graphes où les sommets s'arrangent en couches hiérarchisées et pyramidales
  - ***Cycliques (3)***: on peut identifier des cycles dans le graphe. L'exemple le plus parlant est le graphe circulaire.
  - ***Centralisés/Polaires (4)***: C'est une architecture où tous les sommets sont rattachés à un seul sommet, le pôle.
- *Quelconques (5) : aucune propriété topologique ne semble émerger.*
- *Multipolaires : C'est une architecture mixte entre les graphes centralisés et décentralisés. Les réseaux multipolaires sont très étudiés en raison de leur proximité avec de nombreux cas concrets et notamment Internet ou les réseaux de neurones. Les graphes multipolaires sont caractérisés par deux types d'arêtes, celles qui forment les liens émanant du pôle, les liens forts ; et les liens réunissant deux pôles entre eux : les liens faibles. Les pôles peuvent par ailleurs prendre une architecture structurée (souvent centralisée) ou quelconque.<sup>11</sup>*

---

<sup>11</sup> Olivier Cogis et Claudine Schwartz, *Théorie des graphes*, Cassini, 2018 (ISBN 978-2-84225-189-5)

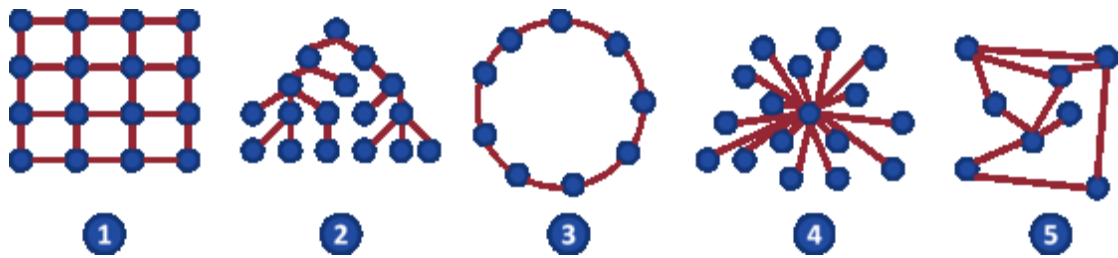


Figure 23 : Principales topologies typiques de graphes.

b. Linear approximation algorithm for First-fit decreasing bin-packing algorithm

Nous présentons un nouvel algorithme d'approximation pour le problème d'emballage. En effet, dans le problème d'emballage des boîtes, ordonnez les boîtes en taille décroissante et appliquez l'algorithme d'emballage de premier montage. Cela rassemble les plus gros éléments en premier, et est plus susceptible de produire une solution optimale que la méthode de premier ajustement simple.

Dans le problème de l'emballage des boîtes, les objets de volumes différents doivent être emballés dans un nombre fini de conteneurs ou de conteneurs de volume  $V$  de manière à minimiser le nombre de conteneurs utilisés. Dans la théorie de la complexité informatique, il s'agit d'un problème NP-difficile combinatoire.

**Principe :**

L'idée de résoudre ce problème est de trier par exemple un tableau des coûts dans l'ordre croissant. Cela arrangera les éléments dans l'ordre croissant du coût.

Passez maintenant au-dessus du tableau des coûts et continuez à calculer la somme des coûts jusqu'à ce que la somme soit inférieure ou égale à nombre  $K$ . Enfin, renvoyez le nombre des éléments utilisés pour calculer la somme qui est juste inférieure ou égale à  $K$ .

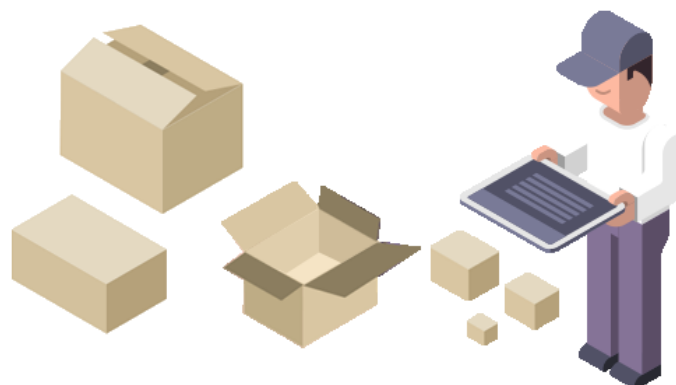


Figure 24 : Algorithme bin packing

*Finalement, à base de ces deux algorithmes celui de la Théorie des graphes et le Bin packing, nous avons réussi à achever notre propre solution à Xenturion.*<sup>12</sup>

Ce chapitre a été consacré au dossier de l'étude technique de l'application. Ainsi, après avoir justifié le choix de l'architecture J2EE, nous avons décrit l'architecture en couches logicielles ainsi que l'architecture applicative du module, ainsi cité les différents outils avec lesquels nous avons implémenté l'application.

L'étude fonctionnelle, avec tous ses éléments réunis, l'étude technique et la conception de l'application permettent de délimiter le projet et d'avoir une idée globale sur sa structure, ce qui peut faciliter amplement l'étape de la réalisation.

---

<sup>12</sup> E.G. CoQmann Jr., M.R. Garay, D.S. Johnson, Approximation algorithms for bin packing: a survey, in: D.S. Hochbaum (Ed.), Approximation Algorithms for NP-Hard Problems, PWS Publishers, 1996, pp. 46–93.

# **Chapitre 4:**

## Mise en œuvre

Après avoir mené à bien les phases de l'étude fonctionnelle, technique et la conception du nouveau module, ce chapitre traite à présent à la phase de réalisation. Il présente les différentes fonctionnalités de l'application.

## I. Technologie Utilisés

Pour l'implémentation de l'architecture logicielle proposée, nous avons utilisé plusieurs technologies.

### 1. Patrons utilisés

On peut diviser les patrons utilisés en deux, ceux qui sont utilisés dans le côté serveur, et ceux qui sont utilisés dans le côté client.

- **Design Pattern MVC:** Le Model-View-Controller (MVC) est un modèle de conception logicielle très répandu, fort utile et indépendant du langage de programmation. Il est aujourd'hui fortement recommandé dans l'univers de JEE.

Son principe : le MVC est un modèle de conception, à trois couches, qui reposent sur la volonté de séparer les données, de l'affichage et des actions. Ainsi l'application se retrouve segmentée en trois composants essentiels :

- **Le modèle** : représente les données et les règles métier. C'est dans ce composant que s'effectuent les traitements liés au cœur du métier. Les données peuvent être liées à une base de données, des services Web, ou même des fichiers XML ou JSON. Il est important de noter que les données sont indépendantes de la présentation et peuvent être exploitées sous forme de classes dans un langage de programmation orientée objet. En d'autres termes, le modèle ne réalise aucune mise en forme. Ainsi, ces données pourront être affichées par plusieurs vies. Du coup, le code sera factorisé : il est écrit une seule fois puis réutilisé par chaque vue.
- **La vue** : dans notre cas, on va utiliser le Framework ZK.
- **Le contrôleur** : il se charge d'intercepter les requêtes de l'utilisateur, d'appeler le modèle correspondant puis de rediriger le résultat vers la vue adéquate. Il ne doit faire aucun traitement de données. Il ne fait que de l'interception et de la redirection.
- **Design Pattern DTO (Data Transfer Object)** : Dans une architecture multicouche sans état, il est relativement fréquent de s'échanger des objets en paramètre de méthode. Aujourd'hui, le pattern DTO (Data Transfert Object) est la technique la plus utilisée pour assurer le découplage entre la couche de présentation et les couches objets métier stockés sur le serveur.

Les DTO sont des structures de données qui véhiculent l'état (les propriétés de cet objet à un instant T) des objets serveur sur le client et permettent également de remonter des informations à persister.

Un DTO n'est rien de plus qu'une classe exposant des propriétés, mais pas de méthodes. Du point de vue design pur, le pattern DTO est une solution quasi parfaite



car il permet d'isoler le modèle du domaine de la présentation, permettant à la fois un couplage lâche et un transfert de données optimisé. Parmi les avantages du modèle de DTO :

- Amélioration des performances. En transmettant plus de données dans un seul appel distant, l'application peut réduire le nombre d'appels distants.
- Transparence des mécanismes internes. Le fait d'émettre et de recevoir davantage de données dans un seul appel masque plus efficacement les mécanismes internes d'une application derrière une interface à granularité grossière.
- Découverte d'objets métier.
- Aptitude au test. L'encapsulation de tous les paramètres dans un objet sérialisable peut favoriser l'aptitude au test.

D'autre part, ce modèle a des inconvénients, citons :

- Risque de multiplication du nombre de classes.
- Calculs supplémentaires.
- Codage supplémentaire.

## 2. Frameworks

- **Spring** : est un Framework libre pour construire et définir l'infrastructure d'une application Java, conçue pour faciliter le développement et les tests. Il est effectivement un conteneur dit « léger », c'est-à-dire une infrastructure similaire à un serveur d'application. Il prend donc en charge la création d'objets et la mise en relation d'objets. Il inclut de divers modules, chacun présentant des services, parmi ses modules utilisés dans l'application, nous citons :
  - L'inversion de contrôle est assurée de deux façons différentes : la recherche des dépendances et l'injection des dépendances.
  - La programmation orientée objet.
  - Une couche d'abstraction permettant d'intégrer d'autres Frameworks et bibliothèques d'une manière flexible.
- **Hibernate**: est une solution open source de type ORM (Object Relational Mapping) qui permet de faciliter le développement de la couche persistance d'une application. Hibernate permet donc de représenter une base de données en objets Java et vice-versa.

Hibernate facilite la persistance et la recherche de données dans une base de données en réalisant lui-même la création des objets et les traitements de remplissage de ceux-ci en accédant à la base de données. La quantité de code ainsi épargnée est très importante d'autant que ce code est généralement fastidieux et redondant.

- **EJB**: La technologie des EJB (pour Enterprise Java Bean) ont été introduite pour offrir aux programmeurs un outil qui facilite la conception et le déploiement d'une couche métier. La version 1.0 très simple est largement revue en 2002 avec la version

2.0. Cette dernière introduit de nombreuses nouveautés avec la possibilité de travailler directement (via des EJB particuliers) sur des données stockées dans une base de données relationnelle.

Les EJB sont des composants et en tant que tels, ils possèdent certaines caractéristiques comme la réutilisabilité, la possibilité de s'assembler pour construire une application, etc... Les EJB et les beans n'ont en commun que d'être des composants. Les JavaBeans sont des composants qui peuvent être utilisés dans toutes les circonstances. Les EJB doivent obligatoirement s'exécuter dans un environnement serveur dédié.

Les EJB sont parfaitement adaptés pour être intégrés dans une architecture trois tiers ou plus. Dans une telle architecture, chaque tiers assure une fonction particulière :

- le client « léger » assure la saisie et l'affichage des données
- sur le serveur, les objets métiers contiennent les traitements. Les EJB sont spécialement conçus pour constituer de telles entités.
- une base de données assure la persistance des informations

Les EJB s'exécutent dans un environnement particulier : le serveur d'EJB. Celui-ci fournit un ensemble de fonctionnalités utilisées par un ou plusieurs conteneurs d'EJB qui constituent le serveur d'EJB. En réalité, c'est dans un conteneur que s'exécute un EJB et il lui est impossible de s'exécuter en dehors.

- **ZK**: est basé sur des technologies open source telles que Java, jQuery et Bootstrap, qui sont utilisées par des millions et des millions de développeurs dans le monde entier. L'implémentation côté client de ZK est basée sur jQuery, mais tout la logique métier est stocké sur le serveur, ce qui signifie une sécurité accrue pour votre application. C'est une préoccupation majeure pour les applications Ajax Enterprise basées sur Java.

ZK vise à combiner la simplicité et la sécurité de son architecture centrée sur le serveur, et la beauté et la dynamique de l'évolution des technologies côté client.

Les composants fantômes peuvent facilement transformer les modèles Bootstrap et les pages HTML standard en pages Web dynamiques avec une liaison de données MVVM en Java. La gestion des données et la liaison des commandes côté client facilitent grandement l'intégration et la réutilisation des widgets ou bibliothèques Javascript tierces de la manière Java la plus appréciée, tout en vous permettant de bénéficier de l'intégration et de la sécurité côté serveur tout aussi importantes.

ZK est livré avec un design moderne et élégant, vous permettant de créer de belles applications en un rien de temps, comme :

- **ZK Chart**: ZK Charts fournit une API complète qui rend la visualisation de vos données exceptionnellement facile. L'API intuitive vous permet de présenter vos données côté serveur sur le client car ZK Charts gère le client-serveur de données pour vous de manière transparente.

L'API complète fournit toutes les fonctionnalités côté client disponibles pour le développeur du côté serveur de Java. Ceci est un avantage distinct pour les développeurs car il leur permet d'atteindre leurs objectifs en utilisant la technologie mature qu'ils aiment.

ZK Charts fournit un nombre incroyable de graphiques attrayants prêts à être insérés dans votre application où vous en avez besoin. C'est excellent pour les développeurs car cela signifie qu'ils peuvent démarrer très rapidement.

### 3. Langage de modélisation



UML, c'est l'acronyme anglais pour « Unified Modeling Language ». On le traduit par « Langage de modélisation unifié ».

La notation UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être exécutées par le logiciel, etc.

UML est né de la fusion des trois méthodes qui ont influencé la modélisation objet au milieu des années 90 : OMT, Booch et OOSE. Il s'agit d'un compromis qui a été trouvé par un Développement d'une application de gestion commerciale tridimensionnelle.

Équipe d'experts : Grady Booch, James Rumbaugh et Ivar Jacobson. UML est à présent un standard défini par l'Object Management Group (OMG). De très nombreuses entreprises de renom ont adopté UML et participent encore aujourd'hui à son développement.

UML est surtout utilisé lorsqu'on prévoit de développer des applications avec une démarche objet (développement en Java, en C++, etc.).

Pourquoi modéliser ? Modéliser, c'est décrire de manière visuelle et graphique les besoins et, les solutions fonctionnelles et techniques de votre projet logiciel.

C'est exactement ce à quoi UML sert dans des projets de réalisation de logiciels.

### 4. Usine de développement



WILDFLY est un serveur d'application Java certifié par Oracle.

WILDFLY est un produit Open Source, sous licence LGPL, édité par la société Red Hat.

L'équivalent propriétaire de WILDFLY est Jboss JEAP, édité par Red Hat.

Contrairement d'autres serveurs d'application Java non certifiés comme Tomcat, WILDFLY fournit l'ensemble des services de niveau entreprise spécifiés dans la spécification Java EE 7.

Les services fournis par WILDFLY sont référencés dans la communauté Java par un document PDF nommé JSR 342 (Java Specification Request 342), ce document est un document de référence pour Java EE 7.

Le nom WILDFLY a été proposé pour éviter toute confusion entre la version open source du serveur d'application Jboss AS (Jboss Application Server) de sa version propriétaire Jboss JEAP (Jboss Entreprise Application Platform).

La dernière version du serveur d'application Jboss AS à utiliser le nom Jboss est Jboss AS 7. Après cette version, c'est WILDFLY qui est utilisé.

On peut dire que WILDFLY 8 = JBOSS AS 8.

Le serveur d'application WILDFLY 8, WILDFLY 9 et WILDFLY 10 sont des implémentations des spécifications Java EE 7 (JSR 342)

Le serveur d'application JBOSS 7 (2012) est une implémentation certifiée des Spécifications Java EE 6 (JSR 316], tout comme 19 autres produits certifiés Java EE 6.

L'architecture de Jboss 7 est très différente de l'architecture des précédentes versions de Jboss (4/5/6). Ce qui déconcerte quelque peu les administrateurs de serveurs d'application Jboss 4/5/6.

L'architecture de Wildfly 8, Wildfly 9 et Wildfly 10 est similaire à l'architecture de Jboss 7. Tout administrateur du serveur Jboss 7 pourra donc s'adapter facilement à WILDFLY 8/9/10.

**Maven™** **Maven** est un outil permettant d'automatiser la gestion de projets Java. Il offre entre autres les fonctionnalités suivantes :

- Compilation et déploiement des applications Java (JAR, WAR)
- Gestion des librairies requises par l'application
- Exécution des tests unitaires
- Génération des documentations du projet (site web, pdf, Latex)
- Intégration dans différents IDE (Eclipse, JBuilder)

Afin de bien gérer les dépendances, Maven s'appuie sur des repositories de librairies (jar). Ces repositories peuvent être locaux à la machine ou accessibles via HTTP. Ainsi, à sa première exécution, Maven télécharge les différents plugins dont il a besoin et les installe dans le répertoire. maven/repository situé dans le répertoire de travail de l'utilisateur. Ainsi, ces mêmes librairies peuvent être réutilisées entre les différents projets.



IntelliJ IDEA (couramment appelé IntelliJ) est un Environnement de développement intégré (IDE) Java développé et édité par JetBrains. IntelliJ est un IDE complet misant sur la productivité avec des systèmes d'autocomplétion intelligente, d'analyse de code en temps réel, de refactoring avancé ; l'intégration d'outils de tests et de debugging ; et une pléthore de raccourcis clavier permettant de réaliser presque n'importe quelle tâche rapidement sans jamais lever les mains du clavier pour utiliser la souris. IntelliJ est disponible dans deux éditions : La Community

Edition est sous licence Apache et propose le support des langages basés sur JVM les plus populaires, tels que Java SE, Groovy et Scala ; permet le développement d'applications Android et intègre JUnit, TestNG, Ant et Maven. La Ultimate Edition est propriétaire et payante et apporte le support de Java EE, de nombreux frameworks Java web (Spring, GWT et Struts pour ne citer qu'eux), des outils de conception avancés et de PHP, Python, Ruby et Node.js.

## 5. Outils Système de gestion base de données



Oracle Database est un système de gestion de base de données relationnelle qui depuis l'introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet.

Oracle se décline en plusieurs versions

- Oracle Server Standard, une version comprenant les outils les plus courants de la solution Oracle. Il ne s'agit pas pour autant d'une version bridée...
- Oracle Server Enterprise Edition

Oracle est un SGBD permettant d'assurer :

- La définition et la manipulation des données
- La cohérence des données
- La confidentialité des données
- L'intégrité des données
- La sauvegarde et la restauration des données
- La gestion des accès concurrents



Oracle SQL Developer est un environnement de développement intégré 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.

Oracle SQL Developer permet le développement de A à Z d'applications en PL/SQL, la mise à disposition de feuilles de travail pour exécuter les requêtes et les scripts, une console pour l'administration de bases de données (DBA), une interface pour la génération de rapports (reporting), une solution complète de conception du modèle de données et une interface de migration permettant de migrer les bases de données d'éditeurs tiers vers Oracle2.

Oracle SQL Developer supporte les produits Oracle ainsi que des plugins qui permettent de se connecter à des bases de données non Oracle. Oracle SQL Developer fonctionne avec IBM DB2, Microsoft Access, Microsoft SQL Server, MySQL, Sybase Adaptive Server, et les bases de données Teradata.

## II. Présentation graphique



Figure 25 : Page authentication

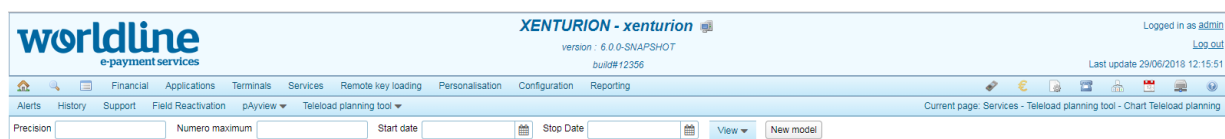


Figure 26 : Champs pour générer la chart

```

"C:\Program ...
**** INIT ****
#####
Type:
- 1 FOR ADDING POS
- 2 FOR ADDING ALERTS
- 3 FOR MAKING SERVICES
- 4 FOR MAKING TRANSACTIONS
- 5 FOR ADDING POS WITHIN A RANGE
- 6 TO ADD TAGS DICTIONARY
- 7 TO ADD POS SCHEDULE PLANNIFIED
- 8 TO ADD POS SCHEDULE RESERVED
- 0 to exit
#####
>> 8
Give the number of the day for Teleload reserved:
>> 170
Current day is 2018-07-02T14:45:50.881
7000 Teleload
358, 1530550516000
7000 Teleload
357, 1530554521000
7000 Teleload
351, 1530578023000
7000 Teleload
357, 1530618133000
7000 Teleload
351, 1530687201000

```

Figure 27 : Remplissage de la base de données

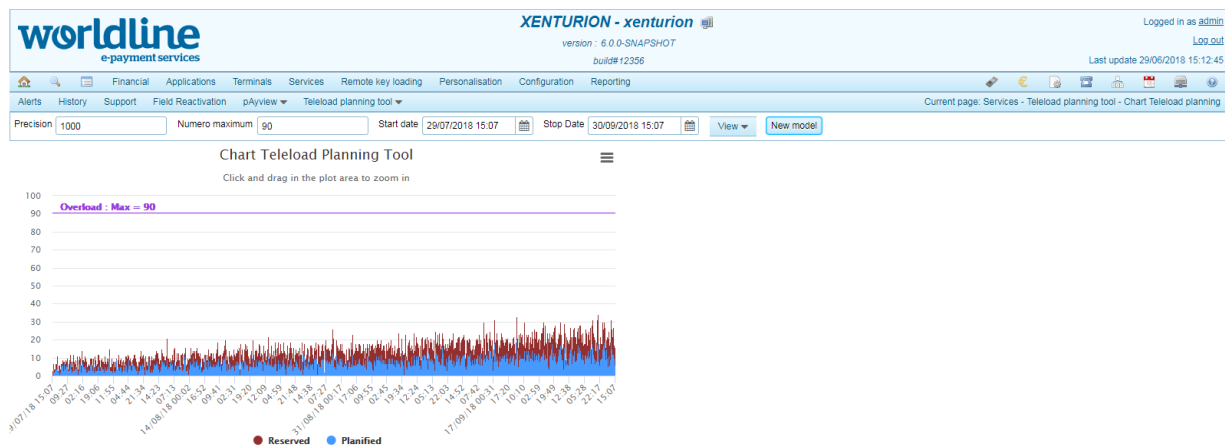


Figure 28 : Charte généré

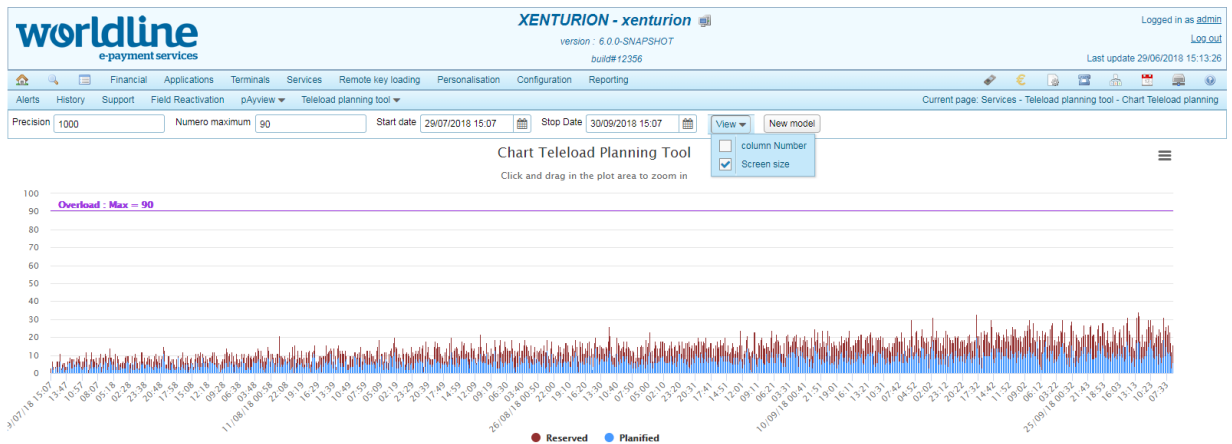


Figure 29 : Changer la size de la charte

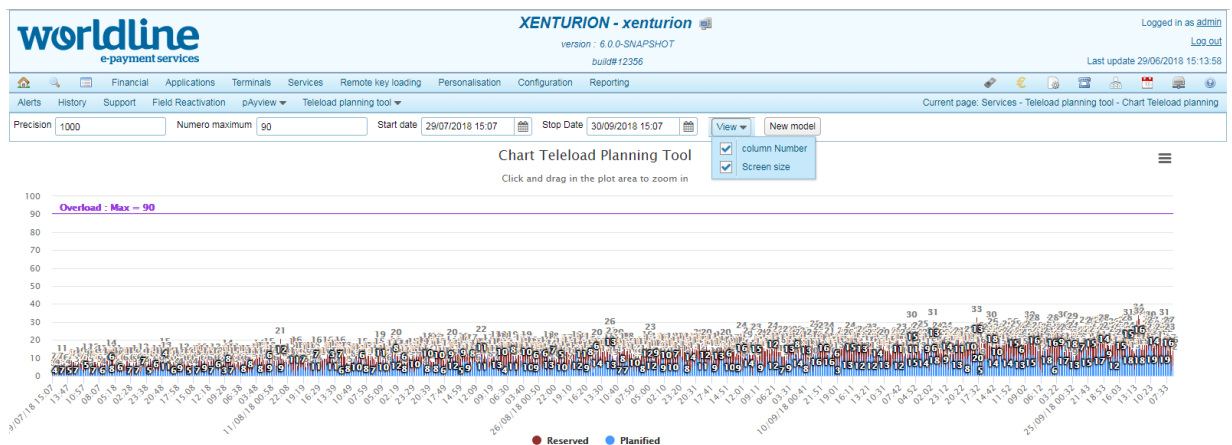


Figure 30 : Option de nombres de colonnes

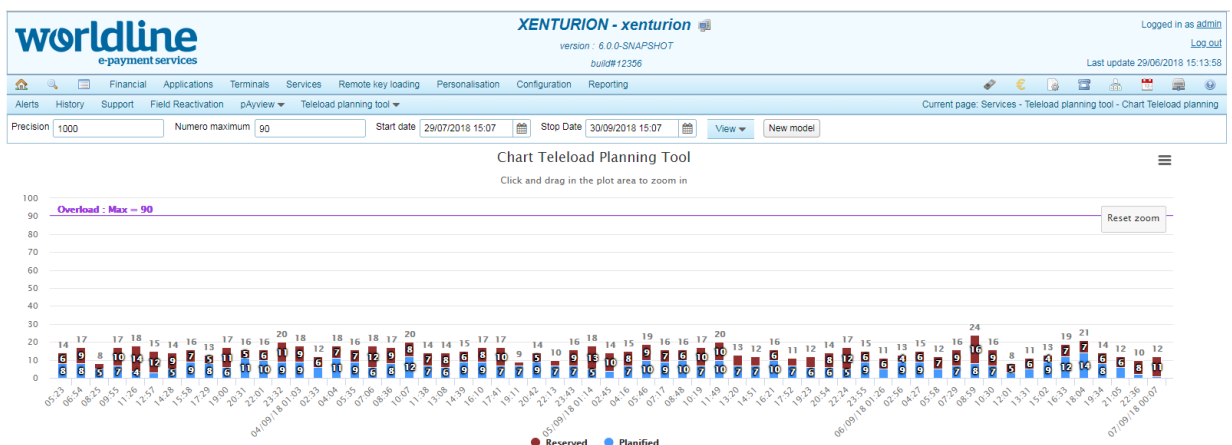


Figure 31 : Option de zoom



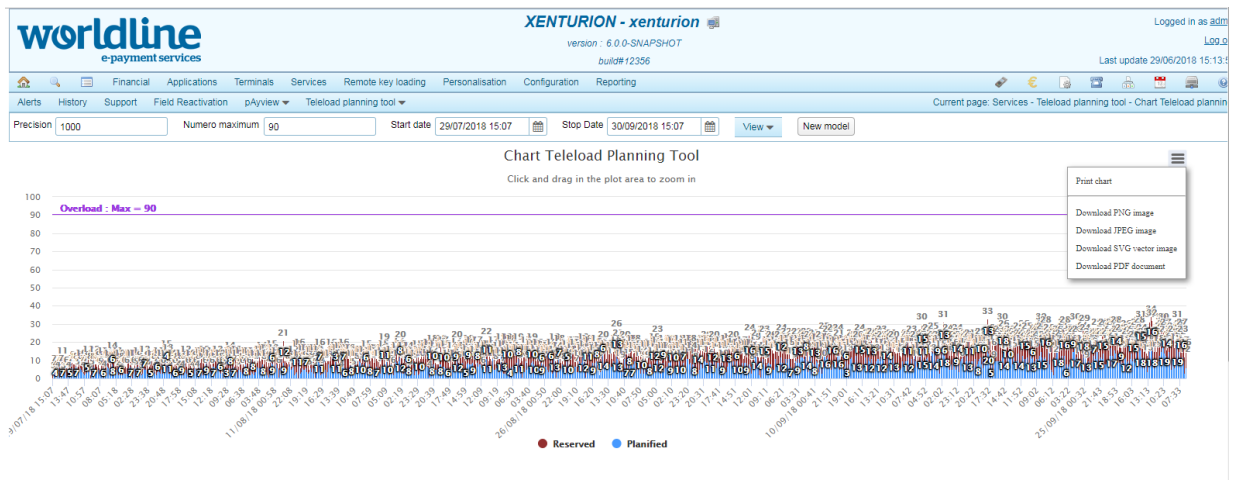


Figure 32 : Menu pour enregistrer la charte



Figure 33 : Création d'un nouveau POS

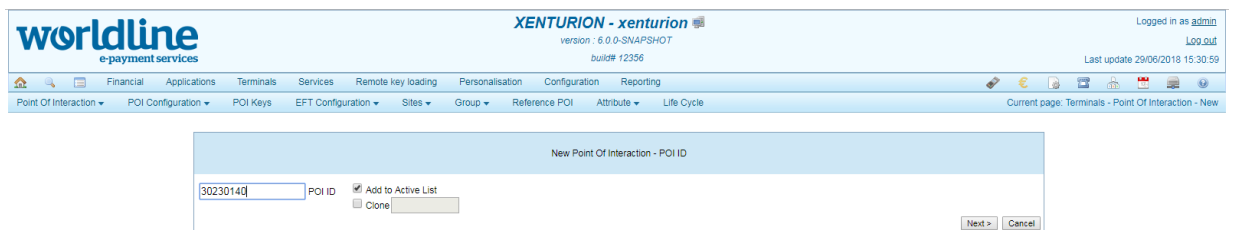


Figure 34 : Choisir ID pour une POS

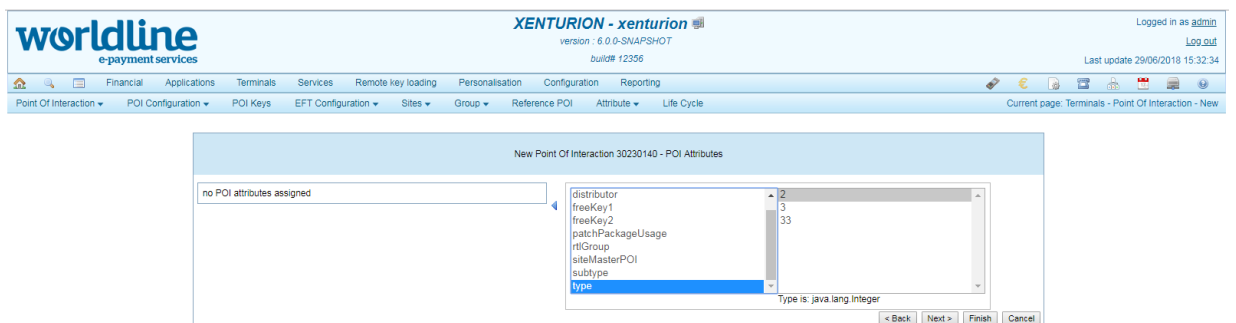


Figure 35 : Choisir la valeur de l'attribut TYPE

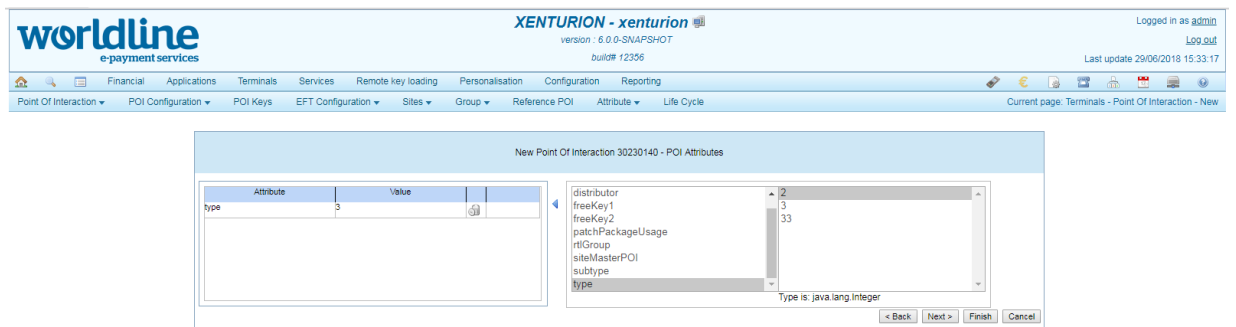


Figure 36 : Affecter le TYPE choisi au POS

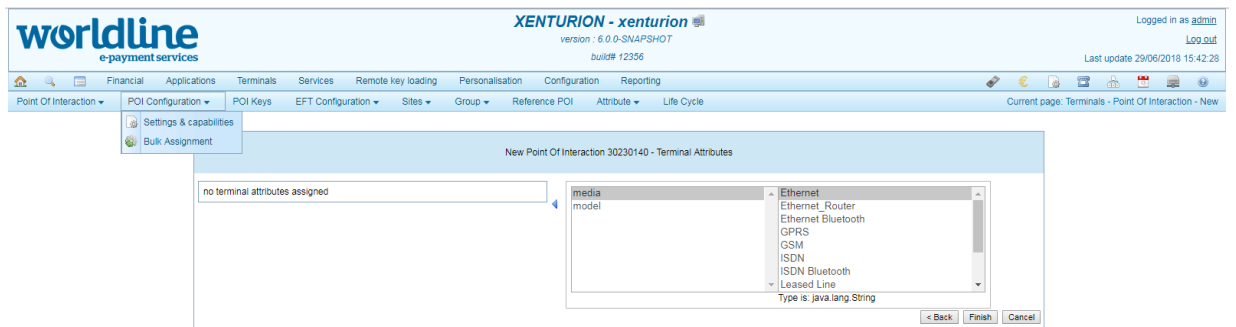


Figure 37 : Choisir la valeur de l'attribut MEDIA

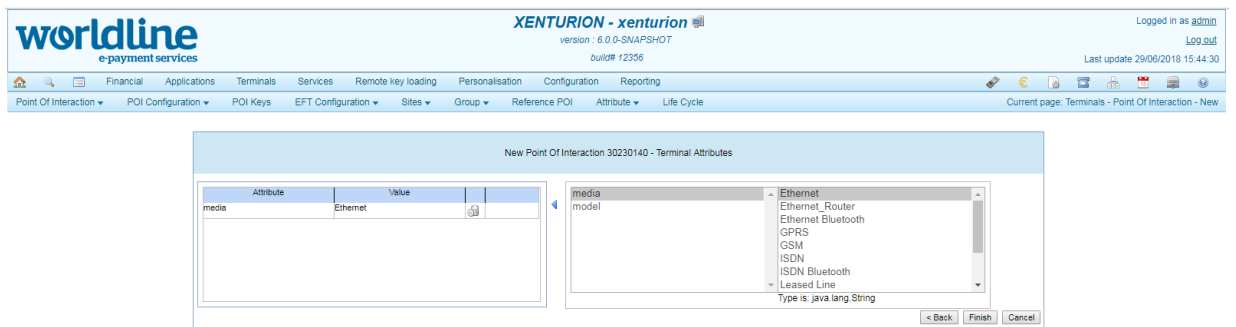


Figure 38 : Affecter MEDIA au POS

**worldline** e-payment services

**XENTURION - xenturion**  
version : 6.0.0-SNAPSHOT  
build# 12356

Logged in as admin  
Last update 29/06/2018 15:45:29

Financial Applications Terminals Services Remote key loading Personalisation Configuration Reporting

Point Of Interaction POI Configuration POI Keys EFT Configuration Sites Group Reference POI Attribute Life Cycle

Active POI list  
New  
Reset  
Delete  
POI ID: 30230140  
Selected terminal (currently assigned): Banksys / Smash family / PHANTOM-0000000400 (29/06/2018)  
Serial number:  
Display all  
# total: 1

Export POI Data for the whole active POI list

POI Terminals Attributes Functionalities Groups Certification Authority Package Version History Teleload Log Teleload Detail

Certification Authority:  
Releases  
Functionalities  
Groups

Reference POI  
Not a reference POI  
Default

Currently assigned terminal  
Banksys / Smash family / PHANTOM-0000000400 (29/06/2018 15:45:25), Status: Not Up To Date

POI Attributes		Terminal Attributes	
Attribute	Value	Attribute	Value
type	3	media	Ethernet

Figure 39 : Création du POS avec les attributs choisis

**worldline** e-payment services

**XENTURION - xenturion**  
version : 6.0.0-SNAPSHOT  
build# 12356

Logged in as admin  
Last update 29/06/2018 16:31:21

Alerts History Support Field Reactivation pAyview Teleload planning tool

Current page: Services - Teleload planning tool - Insert teleload planning

Campaign  
Plan  
TOTO  
TITI  
delhalze 10/2016  
toto 456  
titi 748

Figure 40 : Choisir campagne et lancer l'algorithme de planification en background

Ces interfaces constituent le début de développement du module Teleload Planning Tool, qui sera développé officiellement avec équipe Xenturion la fin du mois août

## Conclusion

Dans le cadre de ce projet de fin d'études, la mission qui m'a été confiée au sein d'AtoS, consistait à développer un outil qui aide en général de planification en se basant sur une performance élevée et des algorithmes de scheduling optimisés.

Pour mener au mieux ce projet, nous avons choisi comme méthodologie du travail la méthode agile SCRUM et la démarche Devps.

Nous avons commencé par une étude des besoins qui ont été identifiés lors des réunions tenues avec l'encadrant dans la société AtoS et apprendre à travailler avec les outils de développement nécessaires à la réalisation du projet. Ensuite, l'analyse de ces besoins nous a permis de délimiter le périmètre du projet. Ces premières phases terminées et validées, nous avons fait approuver la conception de notre système.

Sur le plan technique, Ce stage a été une opportunité pour pratiquer les connaissances, et de découvrir d'autre, en EJB, Spring et ZK, etc ...

Durant tout le projet, nous avons veillé à respecter les exigences qualités afin de fournir une solution qui respecte les besoins des clients.

Tout au long de la période de notre projet de fin d'études, nous avons eu l'opportunité de mettre en application différentes connaissances acquises lors de ma formation à la faculté de sciences et techniques Settat.

Par ailleurs, j'ai tiré un grand bénéfice de ce stage, aussi bien au niveau technique qu'au niveau fonctionnel. En plus, ce stage m'a permis de renforcer le travail d'équipe.

Ce stage a été également une occasion pour découvrir le dynamisme et l'enthousiasme qui caractérisent les employés de la société AtoS en générale et les membres de l'équipe Xenturion en particulier.

# Bibliographie

<sup>5</sup> Bellman, Richard (1957), *Dynamic Programming*, Princeton University Press. Dover paperback edition (2003), (ISBN 0-486-42809-5)

<sup>11</sup> Olivier Cogis et Claudine Schwartz, *Théorie des graphes*, Cassini, 2018 (ISBN 978-2-84225-189-5)

<sup>12</sup> E.G. CoQmann Jr., M.R. Garay, D.S. Johnson, Approximation algorithms for bin packing: a survey, in: D.S. Hochbaum (Ed.), *Approximation Algorithms for NP-Hard Problems*, PWS Publishers, 1996, pp. 46–93.

# Webographie

1 [https://fr.wikipedia.org/wiki/Mock\\_\(programmation\\_orient%C3%A9e\\_objet\)](https://fr.wikipedia.org/wiki/Mock_(programmation_orient%C3%A9e_objet))

2 [http://www.adfkickstart.com/wp-content/uploads/2012/12/scrum\\_framework\\_overview.png](http://www.adfkickstart.com/wp-content/uploads/2012/12/scrum_framework_overview.png)

3 <https://www.thierry-pigot.fr/scrum-en-moins-de-10-minutes/>

4 [https://fr.wikipedia.org/wiki/Test\\_de\\_performance](https://fr.wikipedia.org/wiki/Test_de_performance)

6 <https://www.theserverside.com/discussions/thread/32389.html>

7 [http://www.dba-oracle.com/art\\_9i\\_indexing.htm](http://www.dba-oracle.com/art_9i_indexing.htm)

8 <http://in.relation.to/2015/11/25/Hibernate-Search-5/>

9 <http://www.baeldung.com/hibernate-search>

10 [https://en.wikipedia.org/wiki/Java\\_Persistence\\_Query\\_Language](https://en.wikipedia.org/wiki/Java_Persistence_Query_Language)