

RAPPORT DU PROJET DEVOO

Développement d'une application en suivant une méthodologie de développement objet



H4103

Décembre 2014

Meryem Benchakroune

Karim Benhmida

Amine El Rhazi

Damien Gallet

Mehdi Kitane

Yassine Moreno

Abdelalim Tribak

Table des matières

I-Introduction	3
II-Capture et analyse des besoins	5
II.1-Planning prévisionnel du projet	5
II.2-Modèle du domaine	6
II.3-Glossaire	7
II.4-Diagramme de cas d'utilisation	9
II.5-Description textuelle abrégée de cas d'utilisation	10
III-Conception	11
III.1-Description textuelle structurée des cas d'utilisation	11
III.2-Diagrammes de packages et de classes	13
III.3-Diagrammes de séquences	14
Calcul Tournée	14
SequenceChargerLivraison(Vue)	15
SequenceChargerLivraisons(Application)	16
SequenceChargerLivraisons(Model)	17
InsertionLivraison	18
IV-Implémentation	19
IV.1-Documentation JavaDoc	19
IV.2-Diagramme de classes rétro-généré à partir du code	21
V-Bilan	22
V.1-Bilan effectif du projet	22
V.2-Bilan humain et technique	24

I-Introduction

L'application est inspirée d'un projet réel, piloté par le Grand Lyon, qui a démarré en avril 2012 et qui vise à optimiser la mobilité durable en ville (voir www.optimodlyon.com). Nous nous focaliserons ici sur la partie concernant le frêt urbain et l'optimisation des tournées de livraisons en ville.

Après 3 semaines de capture des besoins, de conception puis de développement, nous avons réussi à réaliser l'application répondant aux besoins spécifiés. Nous avons choisi d'utiliser la technologie Java et le framework Swing pour la réalisation. Ci-dessous quelques captures d'écran de notre application.

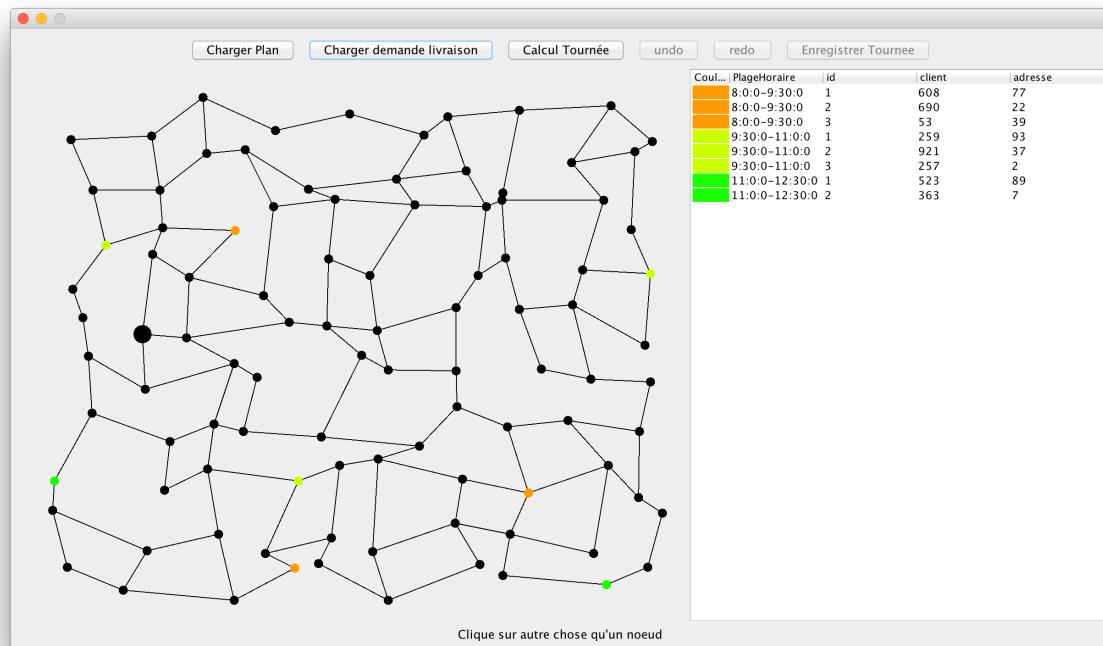


Figure 1 : Capture d'écran après chargement du plan

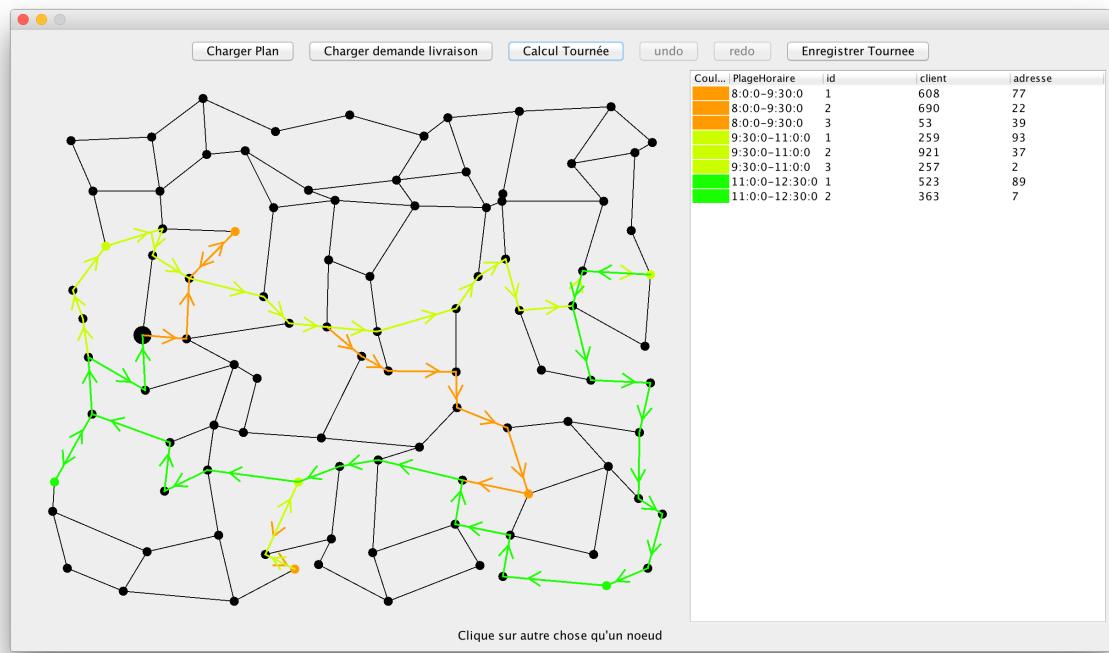


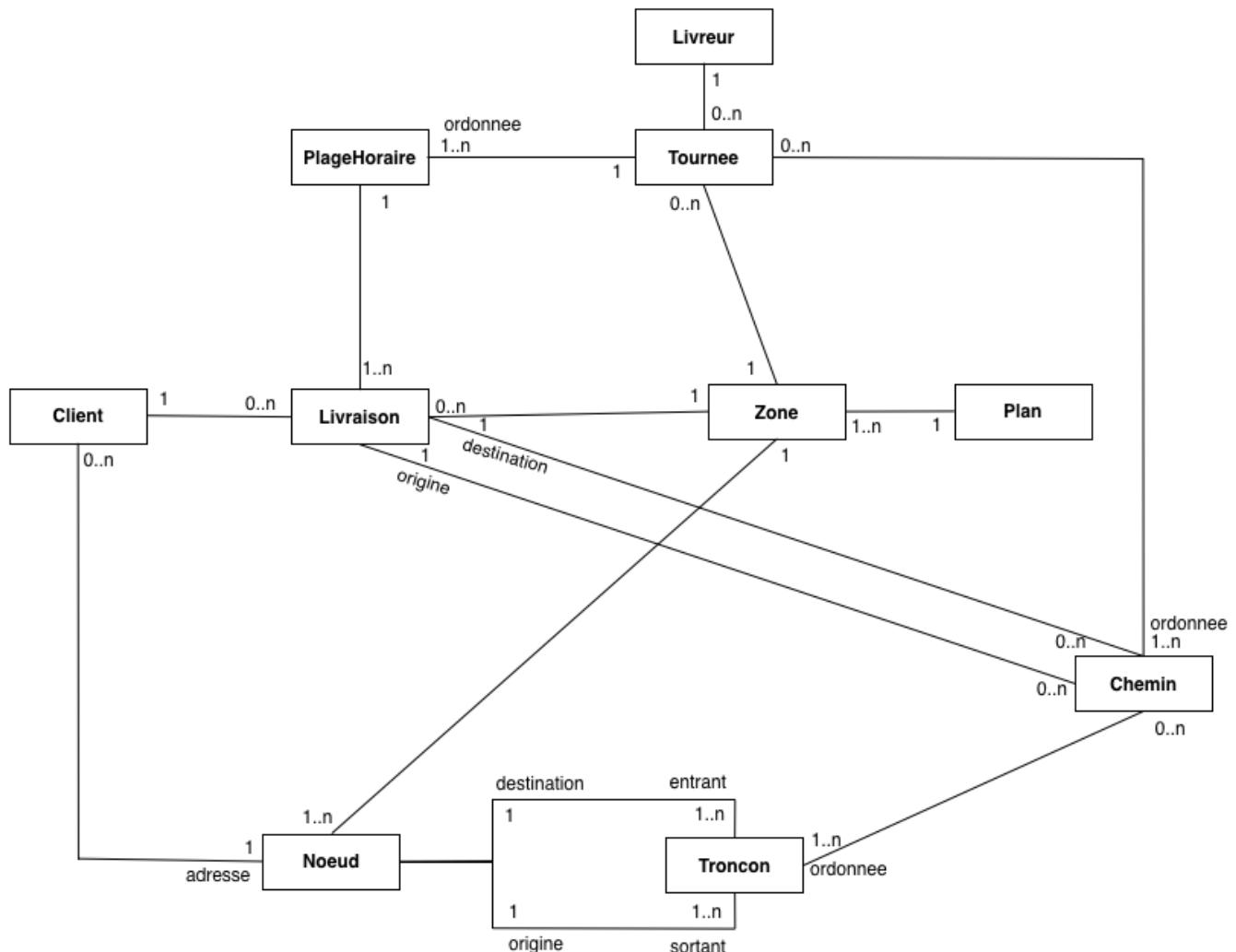
Figure 2 : Capture d'écran après avoir calculé la tournée

II-Capture et analyse des besoins

II.1-Planning prévisionnel du projet

Répartition des tâches			
Tâches	Sous-Taches	Assigner à	Temps prévu par personne
Capture et analyse de besoins	Modèle du domaine	Karim - Damien - Abdelalim	4h
	Diagramme de cas d'utilisation	Meryem - Amine - Mehdi	2h
	Description abrégée des cas d'utilisation	Meryem - Amine	2h
	Glossaire	Karim - Damien - Yassine	3h
Conception détaillé	Description structurée des cas d'utilisation	Meryem - Amine - Yassine	2h
	Diagramme de package et de classes	Karim - Damien - Abdelalim	4h
	Diagrammes de séquence 1	Mehdi - Yassine	3h
	Diagrammes de séquence 2	Damien - Abdelalim - Karim	3h
	Diagrammes de séquence 3	Meryem - Amine	3h
Implémentation / Développement	Implémentation du diagramme du modèle	Meryem - Amine - Mehdi	10h
	Tests unitaires de création	Meryem - Amine	8h
	Vue	Karim - Damien - Mehdi	10h
	Parseur XML	Yassine - Abdelalim	6h
	Implémentation des actions	Mehdi	6h
	Génération fichier texte	Yassine - Abdelalim	4h
	Tests unitaires de commandes-Contrôleur	Mehdi - Yassine	6h
	Calcul tournée (Choco)	Karim - Abdelalim	6h
	Djikstra Plus court chemin	Abdelalim - Karim	3h
	Diagramme de classe rétro-généré	Mehdi	2h
Bilan	Planning effectif du projet	Mehdi	2h
	Bilan humain	Mehdi	2h
	Bilan technique	Karim	2h
	Rapport	Yassine - Damien	3h

II.2-Modèle du domaine



II.3-Glossaire

Client

Entité disposant d'une seule adresse et de coordonnées de contact qui demande la livraison de colis à la société de livraison à un point de livraison.

Point de livraison

Noeud auquel sont livrés des colis.

Colis

Objet devant être livré à un client à un point de livraison.

Noeud

Point repéré par ses coordonnées.

Tronçon

Arc orienté reliant exactement deux noeuds, avec un nom, une vitesse et une longueur, un noeud d'origine et un noeud de destination.

Chemin

Ensemble de tronçons reliant deux livraisons, appelés origine et destination.

Itinéraire

Ensemble de chemins consécutifs formant un circuit (le point de départ est le point d'arrivée).

Dépot

Livraison pouvant étant le point de départ (et d'arrivée) d'un itinéraire, il est inclus dans la plage horaire de la livraison la plus matinale.

Plage de horaire

Intervalle de temps comportant une ou plusieurs livraisons.

Livreur

Personne chargée d'une tournée de livraison par jour, il ne peut livrer qu'une seule zone par jour, il est localisé par le système en temps réel.

Zone géographique

Ensemble de noeuds connectés directement par des tronçons, qui ne peut admettre qu'un nombre limité de livraisons par journée, et qui n'est desservi que par un seul livreur.

Livraison

Evénement défini par un ensemble de colis à livrer à un seul point de livraison dans une plage horaire définie et pour une durée de 10 minutes, la livraison comporte aussi un horaire de livraison, une livraison peut-être effectuée, restant à faire avec retard, ou restant à faire sans retard.

Camion

Assimilé au livreur dans ce TP.

Carte

Ensemble de zones géographiques.

Tronçon entrant

Tronçon dont l'extrémité est le noeud courant.

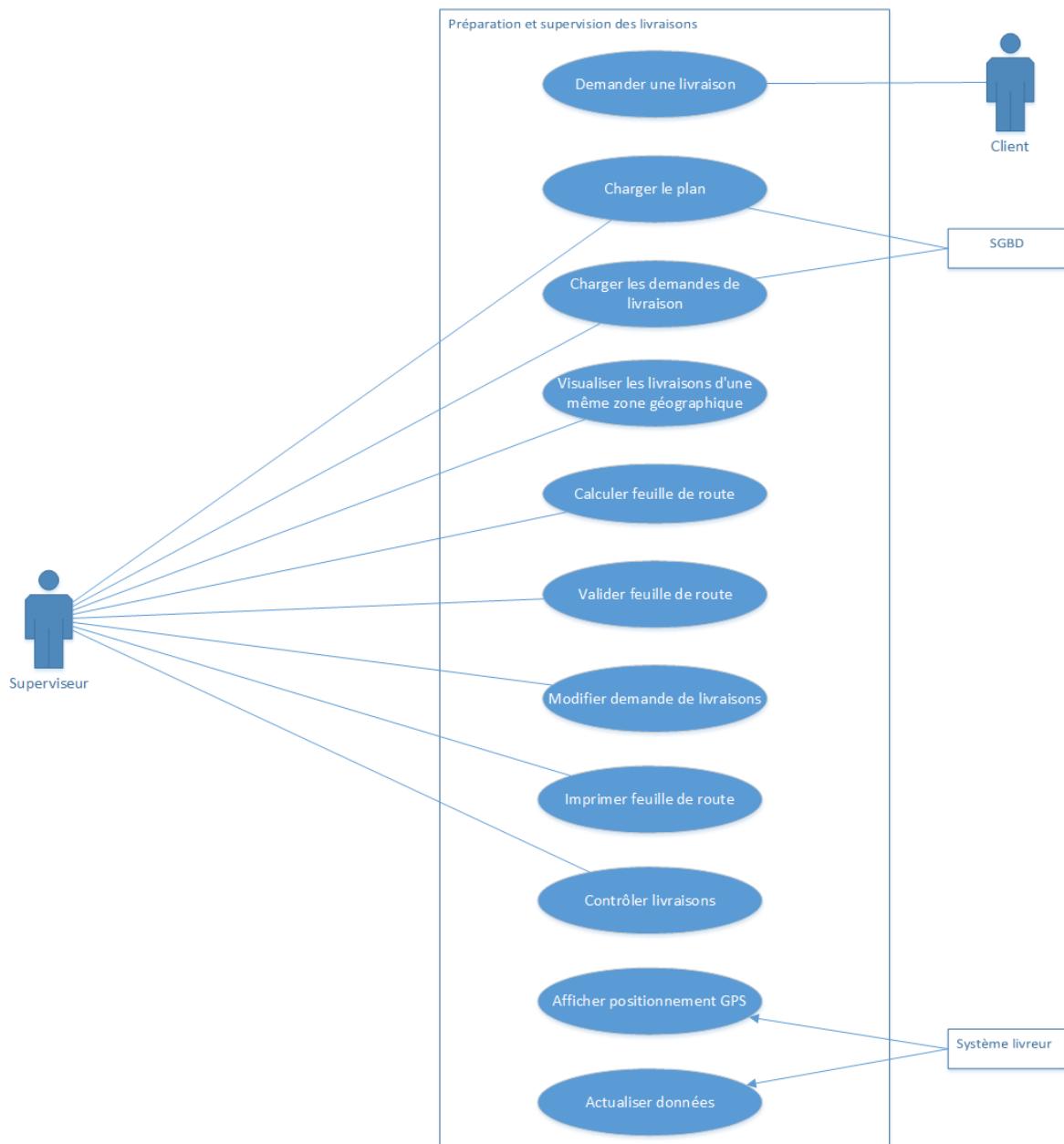
Tronçon sortant

Tronçon dont l'origine est le noeud courant.

Tournée

Ensemble de plages horaires comportant des livraisons associée à une zone qui comporte un dépôt, et, après calcul, une liste ordonnée de chemins.

II.4-Diagramme de cas d'utilisation



II.5-Description textuelle abrégée de cas d'utilisation

Demander une livraison

Le client se connecte sur une application web et demande au système une livraison d'articles souhaités en précisant l'heure et la date de livraison et le moyen de paiement souhaités.

Charger le plan

Le superviseur demande au système de charger le plan de l'agglomération depuis la base de données. Ce plan contient tous les noeuds de l'ensemble des zones géographiques du réseau.

Charger les demandes de livraison

Le superviseur demande au système d'afficher toutes les demandes de livraisons à partir de la base de données.

Visualiser les livraisons d'une même zone géographique

Le superviseur demande au système de visualiser sur le plan de l'agglomération, précédemment chargé, la tournée calculée.

Calculer la feuille de route

Le superviseur demande au système de calculer la feuille de route du lendemain en tenant compte des différentes contraintes.

Valider la feuille de route

Le superviseur valide la feuille de route calculée afin de permettre au système de répercuter les changements sur l'ensemble des livreurs et livraisons concernés.

Modifier la demande de livraison

Le superviseur demande au système de modifier et de mettre à jour la feuille de route déjà calculée en ajoutant, supprimant ou intervertissant les demandes de livraisons. Ces modifications sont interactives, annulables et peuvent survenir à tout moment.

Imprimer la feuille de route

Le superviseur demande au système d'éditer une version papier de la feuille de route incluant toutes les informations nécessaires à la tournée de livraisons.

Contrôler les livraisons

Le superviseur demande au système d'afficher sur le plan de l'agglomération les différentes feuilles de route calculées pour la journée en cours et peut modifier ces feuilles si nécessaire.

Afficher le positionnement GPS

Le système du livreur fournit au système ses coordonnées en temps réel.

Actualiser les données

Le système du livreur fournit au système les nouvelles données (signature client, retard, blocage ...) concernant les livraisons assignées.

III-Conception

III.1-Description textuelle structurée des cas d'utilisation

CU1 : Chargement d'une demande de livraisons à partir d'un fichier XML

Pré-condition

Un plan doit être préalablement chargé.

Scénario de base:

- Le superviseur clique sur le bouton “charger livraison” .
- Le superviseur choisit le fichier XML à charger.
- Le système vérifie la validité du fichier.
- Le système affiche sur le plan chargé les différentes livraisons à effectuer ainsi que le tableau de plage horaire de cette demande de livraisons.
 - ▶ Toute demande antérieure de livraisons est écrasée.

Répétition des étapes 1 et 2 et 3 jusqu'à ce que le superviseur ne veuille plus charger une autre demande de livraison.

Extensions

- Le fichier n'est pas de type XML.
 - ▶ Le système signale l'erreur en invitant l'utilisateur à choisir un fichier XML.
- Le fichier XML est syntaxiquement incorrect.
 - ▶ Le système signale l'erreur ‘fichier XML invalide’.
- Le fichier XML est sémantiquement incorrect.
 - ▶ Le système signale l'erreur ‘fichier XML sémantiquement incorrect’

CU2 : Calcul d'une tournée pour une demande de livraison donnée

Pré-condition

Le superviseur a chargé un plan et une demande de livraisons, le plan est affiché, la liste des livraisons aussi.

Scénario de base:

- Le bouton “Sauvegarder la feuille de route” est grisé.
- Le SV clique sur le bouton “calculer une tournée” .
- Le système affiche une boite de dialogue d'attente qui bloque les autres commandes du logiciel.
- Une fois le calcul terminé, la boite de dialogue s'efface automatiquement.
- Le bouton “Sauvegarder la feuille de route” est actif
 - ▶ Les nœuds et les chemins correspondant aux livraisons calculées dans l'horaire sont affichés dans une certaine couleur selon leur horaire de passage.
 - ▶ Le dépôt et les chemins venant et menant au dépôt sont en violet.

- ▶ Si une des livraisons n'est pas dans la bonne plage horaire, le nœud ainsi que les chemins associés sont affichés en rouge.
- ▶ Tous les tronçons apparaissent avec leur sens de circulation.
- ▶ Les tronçons qui sont empruntés plus d'une fois doivent être explicités d'une certaine façon.
- Si le SV clique sur "Sauvegarder la feuille de route", le système ouvre une boîte de dialogue pour sélectionner la destination du fichier, le SV clique sur Enregistrer pour enregistrer le fichier à cet emplacement (ou aperçu dans une zone dédiée).

Extensions

- Le SV peut cliquer sur le bouton "Annuler" pour annuler le calcul et revenir à l'état précédent.
- Une boîte de dialogue apparaît si la sauvegarde du fichier ne s'est pas déroulée comme prévue.

CU3 : Insertion d'un point de livraison dans une demande de livraisons donnée

Pré-condition

Le superviseur a chargé une demande de livraison et un plan, le plan est affiché, la liste des livraisons aussi.

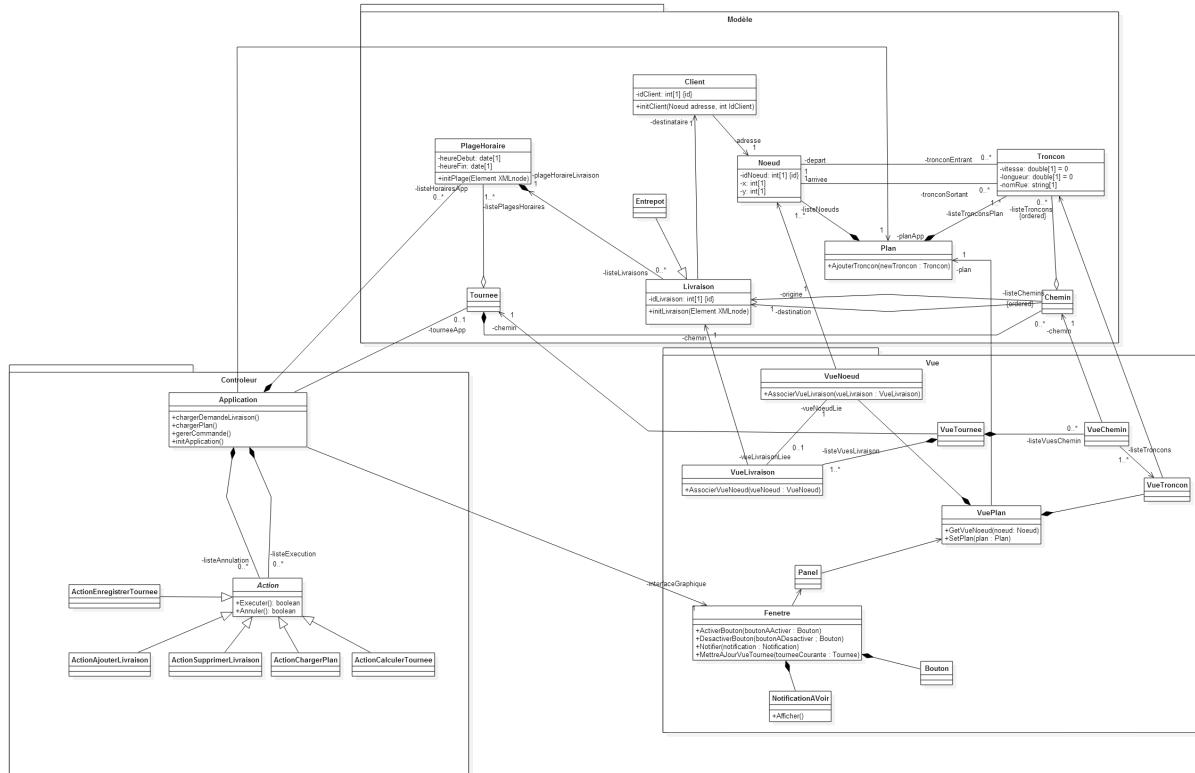
Scénario de base:

- Sélectionner un nœud vierge du canevas.
- Afficher un menu contextuel avec un clic droit.
- Sélectionner point de livraison.
- Remplir les informations du point de livraison.
- Valider la création.

Extensions

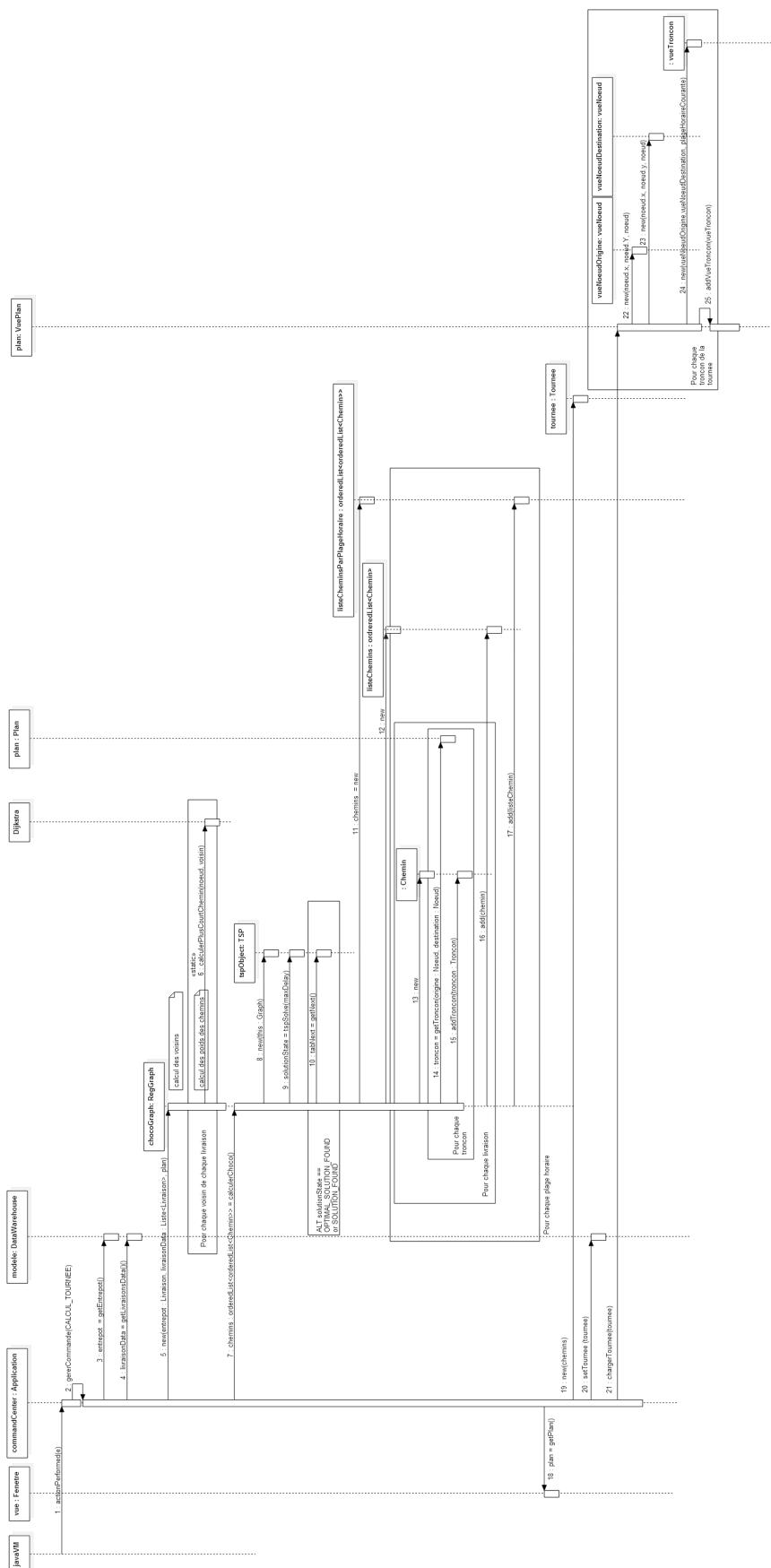
- Le SV peut se rétracter du menu contextuel en cliquant sur une zone vierge du canevas.
- Le SV peut annuler l'action de création avant sa validation.
- Le SV peut cliquer sur le bouton "Undo" pour annuler la création du point de livraison.
- Le SV peut refaire une action de création annulée avec le bouton "Redo".

III.2-Diagrammes de packages et de classes

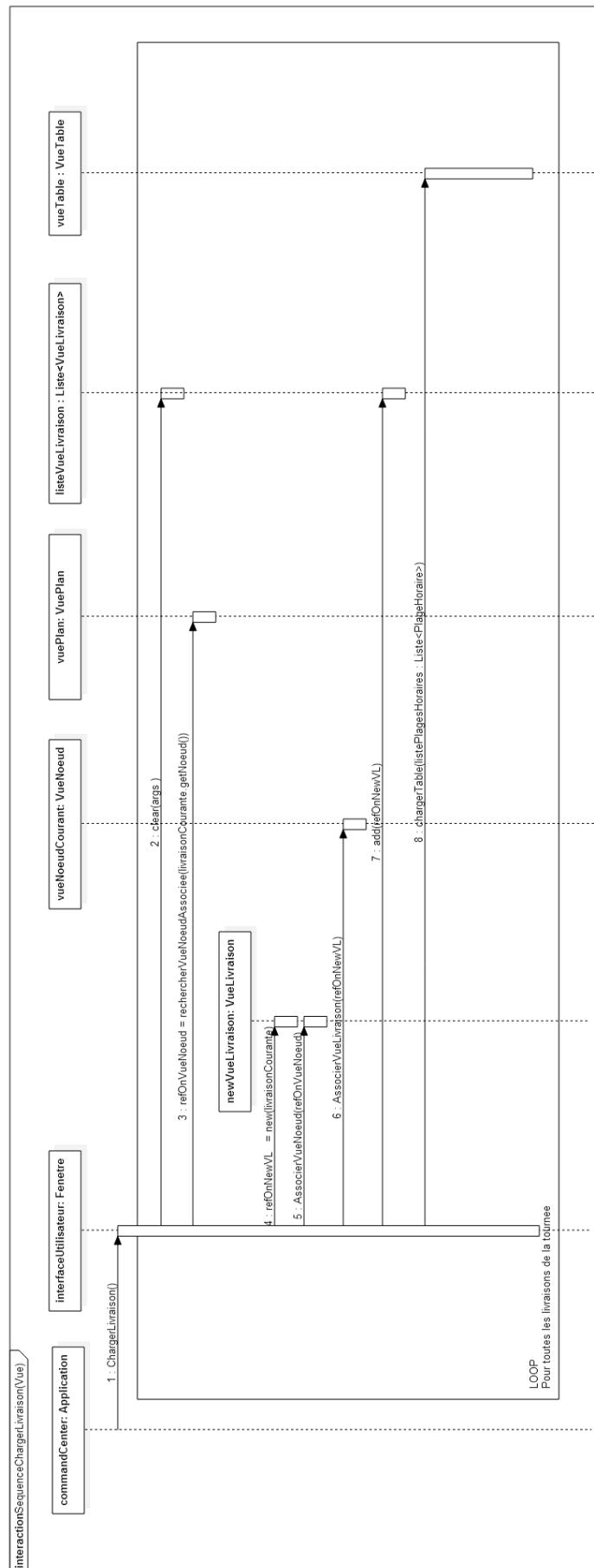


NB: Étant donné que notre application ne gère pas les zones multiples, nous avons simplifié notre modèle en considérant que le plan est équivalent à une zone (ceci revient à remplacer zone par plan et supprimer plan).

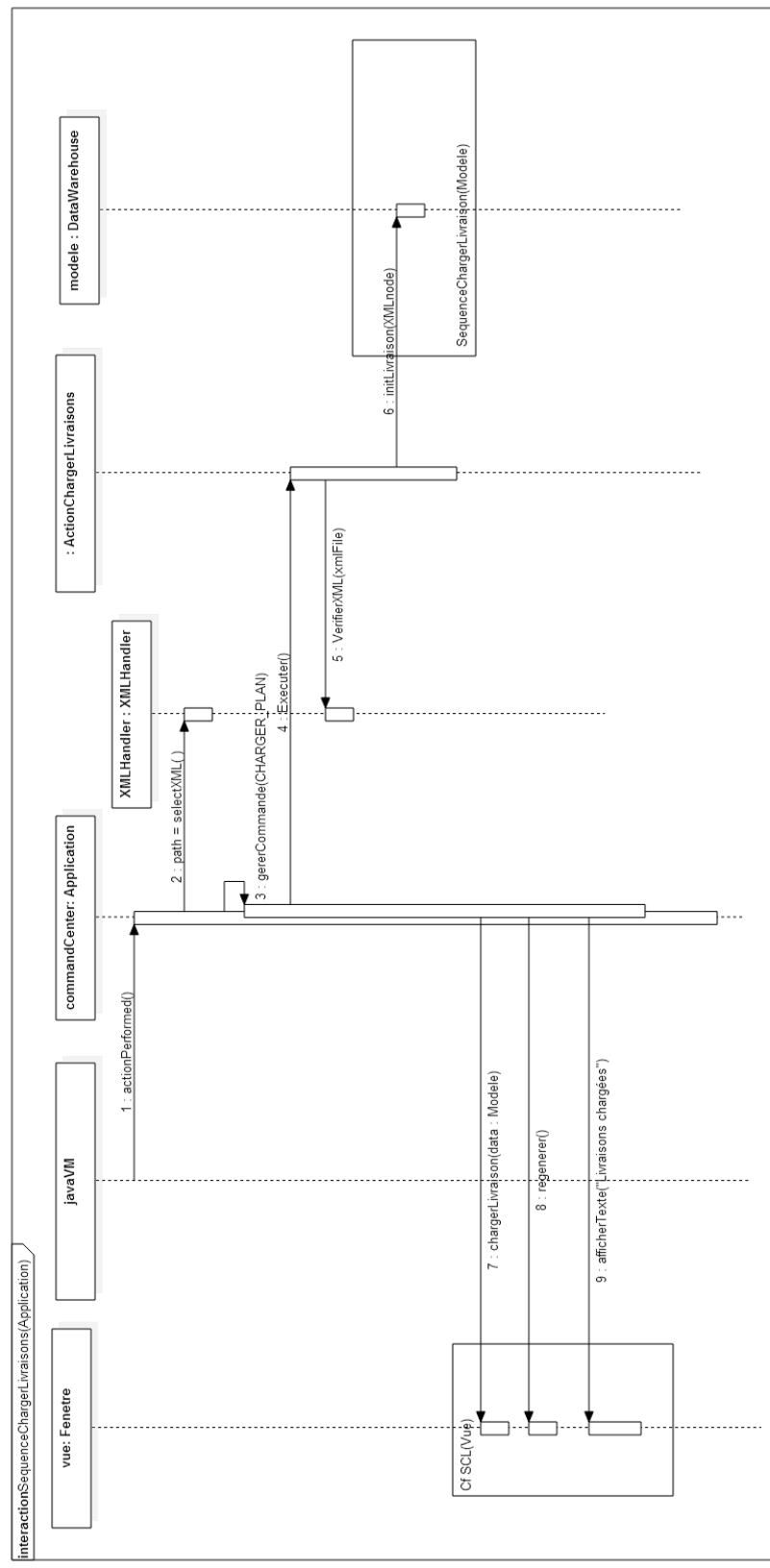
III.3-Diagrammes de séquences

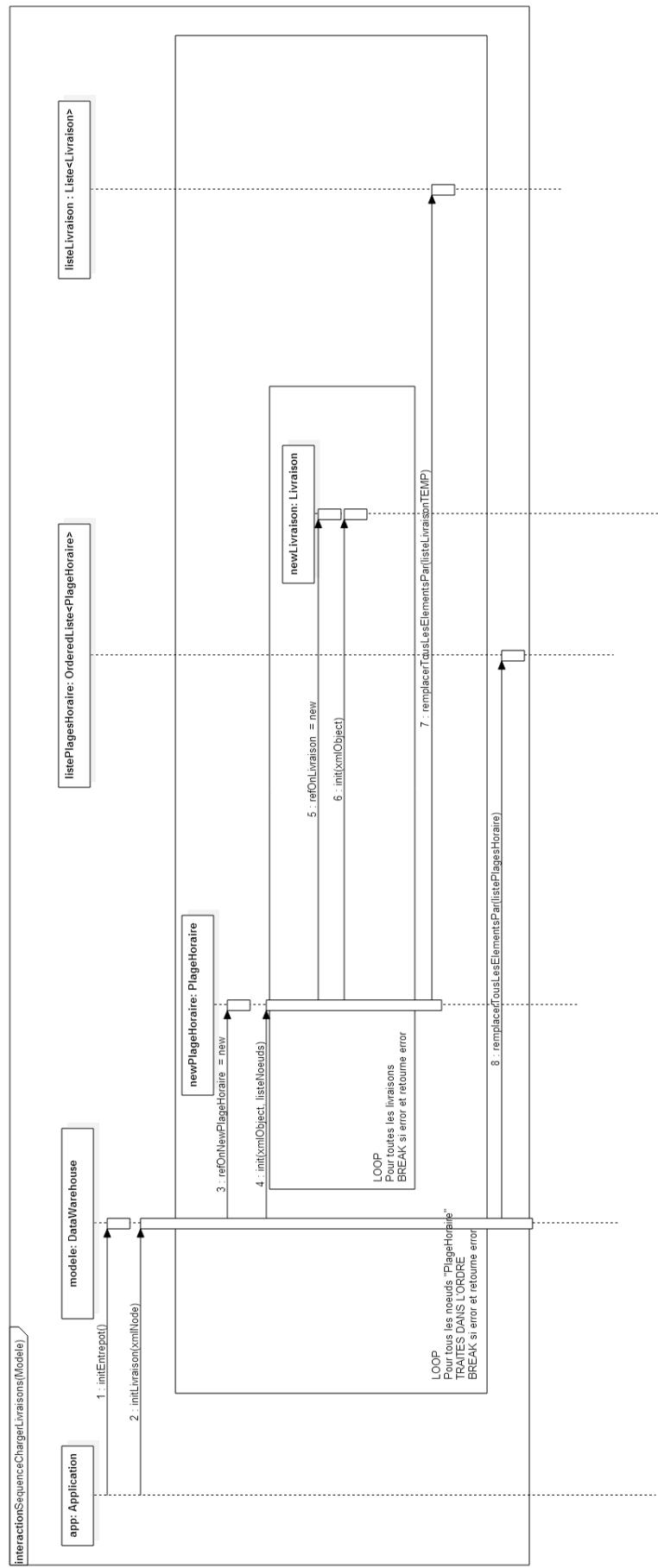


Calcul Tournée

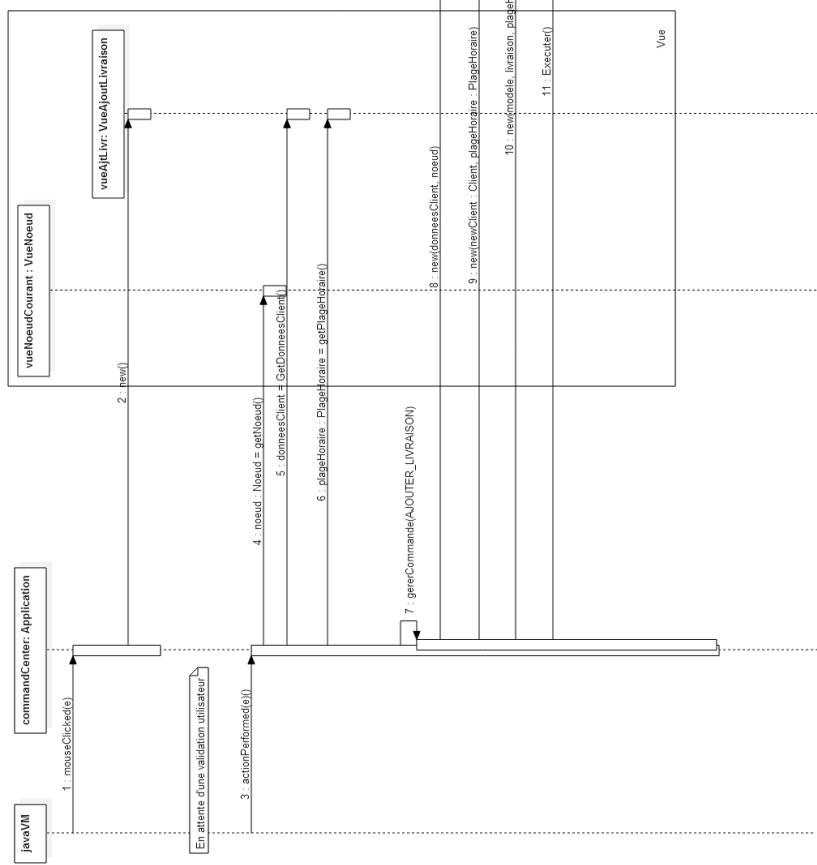
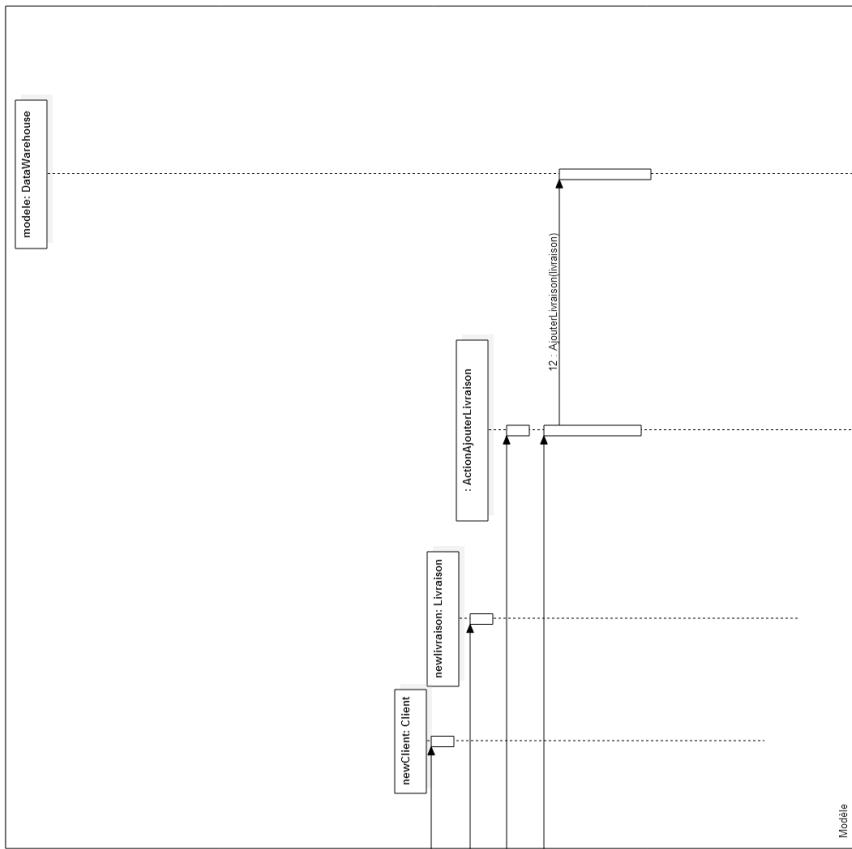


SequenceChargerLivraison(Vue)





SequenceChargerLivraisons(Model)



IV-Implémentation

IV.1-Documentation JavaDoc

La javadoc se trouve dans le dossier Livrables/JavaDoc. Le fichier d'entrée est le fichier Livrables/JavaDoc/index.html. Pour un soucis de lisibilité, voici seulement un extrait.

The screenshot shows a JavaDoc interface with the following details:

- Header:** file:///Users/Mehdi/Desktop/4IF/DevOO/Livrables/JavaDoc/Index.html
- Navigation Bar:** DD, *La vie prof...aFormation.fr, Comment dé...mples de CV, Installer Mag...dre Fillizeau, Serveur idéa..., Eric Kranich, ETIC, IS, INSA, Pers, py, js, DTFY, iOS, nodeJS, css, >>, Application
- Left Sidebar:** All Classes, Packages, Controleur, Modèle, Outils, Tests, Vue, All Classes (selected). Other listed classes include Action, ActionAjouterLivraison, ActionChargerLivraison, ActionChargerPlan, ActionSauvegarder, ActionSupprimerLivraison, AjoutLivraison, Application, Chemin, Client, ControlTest, ControlUndoRedo, DataWareHouse, Dijkstra, DijkstraTest, Fenetre, FileFilterApp, Graph, GraphTest, Livraison, Main, ModeleTable, Nœud, PlageHoraire, Plan, Propriétés.
- Main Content Area:**
 - Class Selection:** CLASS (selected), USE, TREE, DEPRECATED, INDEX, HELP
 - Links:** PREV CLASS, NEXT CLASS, FRAMES, NO FRAMES, SUMMARY: NESTED | FIELD | CONSTR | METHOD, DETAIL: FIELD | CONSTR | METHOD
 - Class Definition:** Controleur
Class Application
java.lang.Object
Controleur.Application
 - Implemented Interfaces:** java.awt.event.ActionListener, java.awt.event.MouseListener, java.util.EventListener
 - Code Snippet:** public class Application extends java.lang.Object implements java.awt.event.MouseListener, java.awt.event.ActionListener
 - Description:** Controleur principal. Reçoit les événements de la vue, effectue les actions correspondantes sur le modèle puis informe la vue des changements.
 - Author:** Yassine Moreno
 - Constructor Summary:**
 - Constructors:** Constructors
 - Constructor and Description:** Application(), Application(Fenetre vue, DataWareHouse modèle)

Extrait 1 : Classe Application

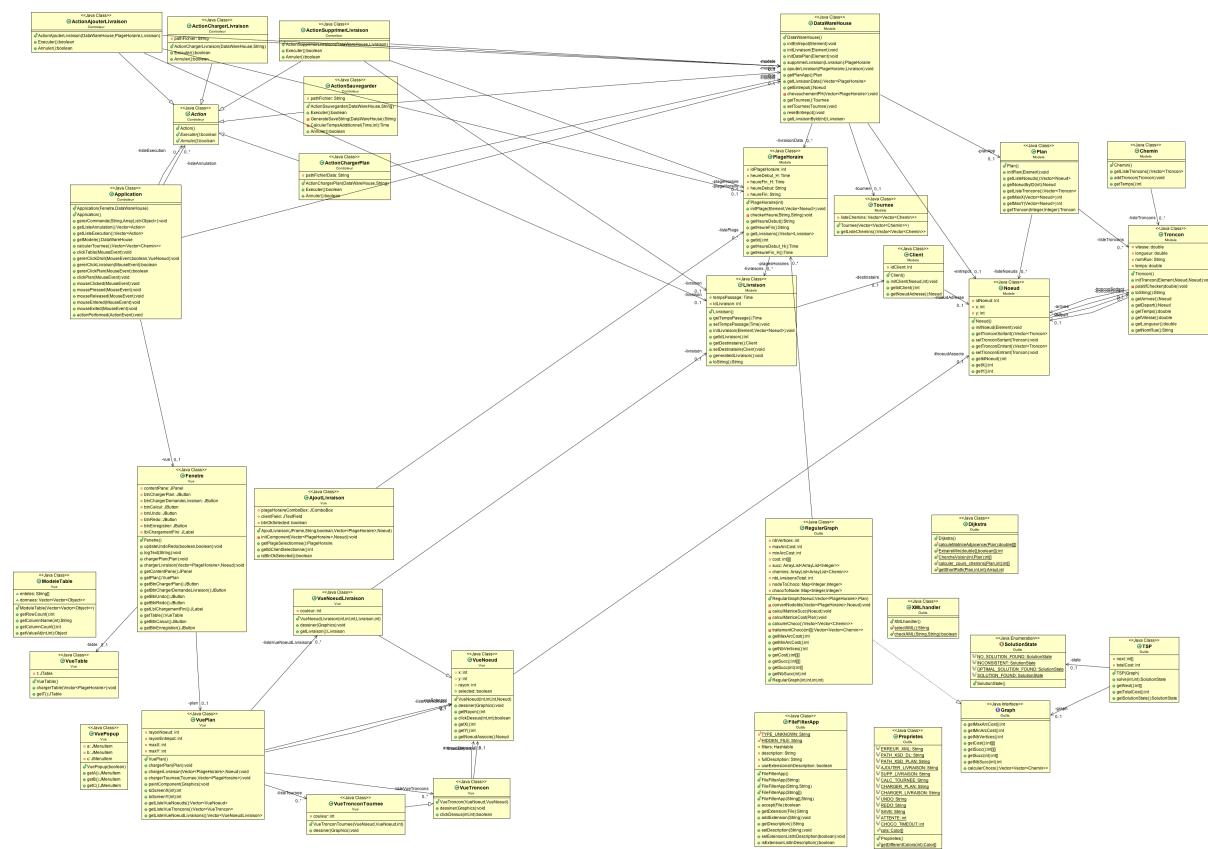
Screenshot of a JavaDoc index page showing the 'Method Summary' for the 'Application' class. The page includes a sidebar with navigation links for various packages and classes.

Method Summary

All Methods	Instance Methods	Concrete Methods
<code>void</code>	<code>actionPerformed(java.awt.event.ActionEvent e)</code>	Methode qui recoit les clics sur les boutons et appelle la methode gererCommande
<code>java.util.Vector<java.util.Vector<Chemin>></code>	<code>calculerTournee()</code>	methode de calcul de la tournée, utilise chocoGraph
<code>void</code>	<code>clickPlan(java.awt.event.MouseEvent e)</code>	Methode qui gère le clic sur un plan
<code>void</code>	<code>clickTable(java.awt.event.MouseEvent e)</code>	Methode qui gère le clic sur la table Selectionne le noeud correspondant sur le plan
<code>void</code>	<code>gererClickDroit(java.awt.event.MouseEvent e, boolean livraison, VueNoeud noeud)</code>	Methode qui gère le click droit sur le plan et affiche le popup avec les bons boutons grisés
<code>boolean</code>	<code>gererClickLivraison(java.awt.event.MouseEvent e)</code>	Methode qui gère le click sur un noeud associé à une livraison
<code>boolean</code>	<code>gererClickPlan(java.awt.event.MouseEvent e)</code>	Methode qui gère le click sur un noeud du plan
<code>void</code>	<code>gererCommande(java.lang.String commande, java.util.ArrayList<java.lang.Object> args)</code>	point d'entrée du contrôleur, dispatch des commandes à exécuter. gestion des undo/redo
<code>java.util.Vector<Action></code>	<code>getListeAnnulation()</code>	
<code>java.util.Vector<Action></code>	<code>getListeExecution()</code>	

Extrait 2 : Classe Application (Suite)

IV.2-Diagramme de classes rétro-généré à partir du code



V-Bilan

V.1-Bilan effectif du projet

Répartition des tâches			
Tâches	Sous-Taches	Assigner à	Temps consacré
Capture et analyse de besoins	Modèle du domaine	Karim - Damien - Abdelalim	5h
	Diagramme de cas d'utilisation	Meryem - Amine - Mehdi	3h
	Description abrégée des cas d'utilisation	Meryem - Amine	3h
	Glossaire	Karim - Damien - Yassine	4h
Conception détaillé	Description structurée des cas d'utilisation	Meryem - Amine -Yassine	3h
	Diagramme de package et de classes	Karim - Damien - Abdelalim	5h
	Diagrammes de séquence 1	Mehdi - Yassine	4h
	Diagrammes de séquence 2	Damien - Abdelalim -Karim	4h
	Diagrammes de séquence 3	Meryem - Amine	4h
Implémentation / Développement	Implémentation du diagramme du modèle	Meryem - Amine - Mehdi	11h
	Tests unitaires de création	Meryem - Amine	9h
	Vue	Karim - Damien - Mehdi	11h
	Parseur XML	Yassine - Abdelalim	8h
	Implémentation des actions	Mehdi	8h
	Génération fichier texte	Yassine - Abdelalim	5h
	Tests unitaires de commandes-Controleur	Mehdi - Yassine	8h
	Calcul tournée (Choco)	Karim - Abdelalim	8h
	Djikstra Plus court chemin	Abdelalim - Karim	4h
	Diagramme de classe rétro-généré	Mehdi	0,5h
Bilan	Planning effectif du projet	Mehdi	3h
	Bilan humain	Mehdi	3h
	Bilan technique	Karim	3h
	Rapport	Yassine - Damien	4h

Tableau 1: Charge de travail

	Temps de travail prévisionnel	Temps de travail réalisé
Meryem Benchakroune	27h	33h
Karim Benhmida	32h	32h
Amine El Rhazi	27h	33h
Damien Gallet	27h	33h
Mehdi Kitane	33h	40,5h
Yassine Moreno	27h	36h
Abdelalim Tribak	30h	35h
Total	203h	242,5h

V.2-Bilan humain et technique

Dès la première séance, nous avons commencé à définir l'ensemble des tâches à faire pour la réalisation du projet : de cette façon, nous avons établi un planning prévisionnel, permettant une meilleure organisation et gestion de l'équipe projet. Cependant, au fur et à mesure de l'avancement du projet, nous avons dû faire face à certaines difficultés et contraintes liées à la réalisation des tâches dans les temps prévus.

Sur un plan humain, la principale difficulté résidait en une répartition équitable des différentes tâches à réaliser. En effet, bien qu'on ait essayé d'équilibrer la charge de travail, certaines tâches ont pris plus de temps que prévu (d'autres moins) et de ce fait, nous avons dû modifier notre planning, en consacrant plus d'heures que prévues à certaines tâches (par exemple, le calcul de la tournée avec Choco et Dijkstra). Cependant, la diversité des profils des membres de l'équipe nous a permis d'approcher le projet sous différents angles et de soulever des remarques intéressantes, nous aidant ainsi à améliorer la performance de notre application (en testant différents cas d'erreurs par exemple). Cette diversité a représenté une force, car chacun des membres de l'équipe a pu obtenir un avis critique de son travail, lui permettant ainsi de l'améliorer.

D'un point de vue technique, la réalisation préalable du modèle du domaine et du diagramme des classes nous a permis d'avoir une vision globale plus précise du projet, facilitant ainsi la phase de développement. Aussi, comme la plupart des membres de l'équipe projet avaient déjà travaillé avec Swing, la réalisation de l'interface n'a pas été une difficulté. Par contre, ce qui nous le plus posé problème est l'implémentation de l'algorithme de Dijkstra plus le calcul du plus court chemin et l'utilisation de la bibliothèque Choco pour le calcul de la tournée. Nous avions assigné chacune de ces deux tâches à deux personnes seulement, et avec du recul, nous pensons qu'il aurait fallu que plusieurs personnes travaillent dessus : il s'agit d'un point clé de l'application, qui implique des besoins mathématiques et techniques complexes.

Finalement, malgré les quelques difficultés rencontrées, à l'exception d'une fonctionnalité que nous n'avons pas eu le temps de compléter (Gestion des livraisons en retard), nous avons réussi à nous organiser pour finir le projet dans les temps.