

# RAPPORT DE SYNTHÈSE

## Projet informatique

**Sujet** : Collecte de déchets valorisables

Avril – Mai 2018

# SOMMAIRE

I. Organisation des tâches .....	3
II. Développement de WasteAdmin1.0 .....	3
<i>a) Base de Données .....</i>	<i>4</i>
<i>b) Programmation des fonctionnalités .....</i>	<i>4</i>
<i>c) Interface graphique .....</i>	<i>5</i>
III. Description des packages .....	5
IV. Exécution du programme .....	6
V. Bilan .....	6
VI. Bibliographie .....	7

ANNEXE : Liste des fonctionnalités de WasteAdmin1.0

Dans le cadre du projet informatique d'Ingénieurs 1<sup>ère</sup> Année de l'ENSG, nous avons réalisé une application nommée WasteAdmin destinée aux gestionnaires d'une entreprise factice de collecte de déchets valorisables. Cette application doit leur permettre de gérer leur entreprise (planification des collectes, gestion du personnel, du matériel, etc ...) mais également d'éditer des statistiques sur les collectes, les équipes et les secteurs de collectes. Les prochaines pages relatent le cheminement qui a été nécessaire pour développer cette application.

## **I. Organisation des tâches**

Lors de la partie analyse, nous avons commencé par faire des recherches sur les collectes des déchets et sur leur organisation. Nos avis n'ont pas toujours convergé, notamment sur la modélisation à adopter. Nous avons ainsi passé de longues heures à discuter, à émettre des hypothèses, pour finalement nous accorder sur la modélisation présentée dans le Rapport d'Analyse.

La partie développement s'est partagée de la manière suivante :

Matthieu :

- Création de la base de données
- Écriture des requêtes SQL (testées au préalable dans SQLite)
- Écriture/Relecture des différents écrits
- Codage de certaines fonctionnalités
- Relecture des Javadocs et du code dans son intégralité
- Test du programme, débogage

Hugo :

- Mise en place de l'architecture du code
- Codage de l'interface graphique (tableau, graphique, boîtes de dialogues)
- Codage des fonctionnalités et écriture des Javadocs
- Écriture de la documentation utilisateur
- Relecture des écrits, débogage

## **II. Développement de WasteAdmin1.0**

Le projet s'est organisé en trois grandes parties : la création de la base de données, la création des fonctionnalités pour l'application WasteAdmin1.0 et enfin, la création d'une interface graphique.

Voici ci-dessous quelques commentaires relatifs à chacune de ces parties :

## **a) Bases de données**

Nous avons utilisé le SGBD SQLite (pour sa flexibilité d'utilisation), pour créer la base de données de l'entreprise, nommée WasteBDD. Dans le but de faciliter les interactions entre l'application et la base de données, nous avons pris la convention de gérer les dates sous forme d'une chaîne de caractères.

Les éboueurs ont tous été regroupés dans une même table quelque soit leur statut (conducteur ou ramasseur). Par conséquent, c'est l'attribut `date_permis` (date d'obtention du permis) qui permet de distinguer les conducteurs des ramasseurs. Lors du rapport d'analyse, le champs `date_permis` des ramasseurs devait être NULL mais cela génèrait de nombreuses exceptions dans le code Java. Les ramasseurs se voient ainsi attribuer la chaîne de caractère « 0 ».

Le nombre de secteurs (16) et de décharges (4) sont fixes car l'entreprise n'a pas stipulé qu'elle souhaitait pouvoir en ajouter et/ou en supprimer. Nous avons laissé un secteur sans équipe et une équipe avec un seul secteur, pour pouvoir utiliser l'application et montrer la fonctionnalité des méthodes.

On précise que les informations contenues dans la base de données servent uniquement à illustrer le fonctionnement de l'application. Dans la plupart du temps, les données sont factices. Ainsi, le poids des déchets récoltés et leurs volumes respectifs peuvent être en inadéquation. Afin de présenter des résultats graphiques, une semaine de collecte a été constituée dans la période du mois d'avril 2018 (04/2018). La totalité des requêtes SQL présentes dans le code Java ont préalablement été testées afin de vérifier leur bon fonctionnement.

## **b) Programmation des fonctionnalités**

Plusieurs méthodes ont été modifiées (paramètres, void ou retour, static ou non) par rapport à ce qui était annoncé dans le rapport d'analyse. De plus, de nouvelles méthodes ont été ajoutées car elles nous ont paru intéressantes et/ou nécessaires dans le fonctionnement de l'application. On citera comme exemple les fonctionnalités *liste()* qui permettent d'afficher à l'utilisateur les données contenues dans la base de données, sans avoir à ouvrir cette dernière, ce qui lui permet de ne pas avoir installé SQLite.

Les statistiques sont calculées annuellement ou mensuellement. Dans le rapport d'analyse, nous voulions les calculer de manière hebdomadaire, mais après réflexion, cette échelle de temps nous a paru trop brève pour dégager quelque analyse que ce soit. Par exemple, les statistiques mensuelles permettent de dégager des périodes de forte consommation (mois de décembre, janvier en période de fêtes) permettant ainsi d'ajuster la capacité des véhicules de collectes sur ces dites périodes.

Nous n'avons pas utilisé toutes les statistiques possibles comme le nombre d'habitants ou la proportion des déchets entre les entreprises et les foyers. De nouvelles fonctions pourront voir le jour lors de la prochaine mise à jour car la base de données contient toutes les données nécessaires.

Les salaires des éboueurs sont calculés à partir du salaire des éboueurs de la ville de Paris [1].

### c) Interface graphique

Pour plus de réalisme, nous avons décidé d'implémenter une interface graphique. Le développement de cette interface a été rendu possible après avoir suivi quelques cours en ligne [2].

Toutefois cette interface n'a pas pu être développée de manière optimale et aboutie car plusieurs points restent à nos yeux « brouillon » :

- Lorsque l'on clique sur la croix ou les boutons « annuler » des boîtes de dialogue, le programme continue et ne se ferme pas. Ainsi, si l'on souhaite quitter le programme en cours, il est nécessaire de faire planter l'application.
- L'aspect des boîtes de dialogue est un peu austère, et il aurait été intéressant, avec plus de temps, de développer davantage leur aspect graphique.
- La mise en place de menus dynamiques aurait pu être réalisée avec davantage de temps.

Les histogrammes ont été réalisés à l'aide des bibliothèques JfreeChart [4] et les boîtes de dialogue avec Swing.

## III. Description des packages

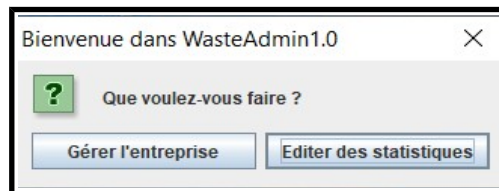
L'application est constituée de 8 packages, comportant chacune une ou plusieurs classes.

Package	Description	Nombre de classes
connexion	Permet d'établir la connexion entre le code et la BDD	2
gestion	Ensemble des fonctionnalités permettant la gestion de l'entreprise	8
interfaceGestion	Affichage graphique du package <b>gestion</b> . Permet à l'utilisateur d'utiliser les fonctionnalités de <b>gestion</b>	3
interfaceGraphique	Affichage graphique de l'application (tableaux, graphiques)	2
interfaceStatistiques	Affichage graphique du package <b>statistiques</b> . Permet à l'utilisateur d'utiliser les fonctionnalités de <b>statistiques</b>	3
interfaceUtilisateur	Interface permettant à l'utilisateur de faire appel à toutes les fonctionnalités des autres packages	3
statistiques	Regroupe les fonctionnalités qui calculent des statistiques depuis des données de la BDD	3
wasteAdmin	Lance l'application	1

## **IV. Exécution du programme**

Pour démarrer l'application, l'utilisateur doit suivre le tutoriel fourni dans le dossier de l'application WasteAdmin1.0. Il est nécessaire d'effectuer les opérations suivantes :

- 1) Ouvrir un terminal (ou invite de commande) et se placer dans le dossier contenant le fichier WasteAdmin.jar.
- 2) Taper `java -jar WasteAdmin.jar` dans le terminal.
- 3) Renseigner le chemin de la base de données WasteBDD et appuyez sur entrée. Si le chemin est correct, la mention « Connexion effectuée avec succès ! » apparaît dans le terminal.
- 4) L'application s'ouvre.



### **Remarques :**

- Si l'on souhaite fermer l'application, on peut cocher sur la croix rouge (celle-ci fonctionne) de la boîte de dialogue d'accueil.
- A la fin de chaque opération, l'application propose à l'utilisateur s'il souhaite continuer. S'il répond « Oui », alors il est renvoyé à la boîte de dialogue d'accueil. Sinon, l'application se ferme.
- Lorsque l'application est fermée, la connexion avec la BDD est automatiquement fermée. La mention « Déconnexion effectuée avec succès ! » apparaît alors dans le terminal.

## **V. Bilan du projet**

Le développement de l'application WasteAdmin a été pour nous une ouverture sur ce que peut être un projet professionnel. La première difficulté que nous avons rencontrée a été les contraintes du cahier des charges. En effet, les différences de compréhension entre le commanditaire et l'exécutant peuvent être extrêmement différentes. Pour pallier à ce problème, nous avons simulé des spécifications plus précises émanant de l'entreprise. Dans un cas réel, nous aurions demandé à nous entretenir avec les commanditaires.

A ceci s'ajoute la modélisation qui s'est révélée très prenante car nous avons fait le choix de nous documenter sur le monde de la collecte des déchets pour en tirer la modélisation la plus fidèle possible. La volonté de fournir un produit de qualité a constamment été une préoccupation, autant sur le code que sur sa documentation ou encore la rédaction des rapports.

Nous avons également pris en compte la nécessité de posséder les mêmes versions d'un programme ou d'une application pour éviter les conflits d'exécution. Dans notre cas, nous avons

travaillé avec deux versions de Java (1.8 et 1.10) ce qui a entraîné des erreurs lors de l'exécution de WasteAdmin1.0.

Ce projet nous a également permis de percevoir la programmation non plus comme un exercice scolaire mais comme un précieux outil pour mettre en œuvre un service pour un tiers. Ainsi, notre attitude face à ce projet nous a parfois étonné, comme le fait de ne pas voir passer les heures devant l'écran, ou bien de toujours avoir dans le coin de notre tête une interrogation de « Pourquoi ça ne marche pas ? », « Comment pourrait-on améliorer ? ». Cette motivation nous a permis de réaliser 28 fonctionnalités utilisables par l'utilisateur. Elles sont présentées en ANNEXE.

## **VI. Bibliographie**

[1] Salaire des éboueurs de la ville de Paris :

<https://www.emploi-collectivites.fr/grille-indiciaire-ville-paris-eboueur-ville-paris/2/6387.htm>

[2] Cours en ligne – Menus et boîtes de dialogues :

<https://openclassrooms.com/courses/apprenez-a-programmer-en-java/les-menus-et-boites-de-dialogue>

[3] Bibliothèque JfreeChart :

<http://www.jfree.org/jfreechart/>

## Annexe : Liste des fonctionnalités de WasteAdmin1.0

### package

Classes	Fonctionnalité	Méthode à utiliser
NomClasse	Ce à quoi sert la méthode	<i>méthode()</i>

-----

### gestion

Classes	Fonctionnalité	Méthode à utiliser
Eboueur	Donne la liste des éboueurs de l'entreprise	<i>listeEboueur()</i>
Equipe	Donne la liste des équipes de l'entreprise	<i>listeEquipe()</i>
Vehicule	Donne la liste des véhicules de l'entreprise	<i>listeVehicule()</i>
Decharge	Donne la liste des décharges partenaires de l'entreprise	<i>listeDecharge()</i>
Secteur	Donne la liste des secteurs de collecte de l'entreprise	<i>listeSecteur()</i>

-----

### interfaceGestion

Catégorie	Fonctionnalité	Méthode à utiliser
GestionCollecte	Planifier une collecte à réaliser	<i>planificationCollecte()</i>
	Avoir accès aux secteurs à collecter du jour	<i>collecteAPlanifier()</i>
	Mettre à jour une collecte réalisée	<i>mettreAJourCollecte()</i>
	Annuler une collecte	<i>annulerUneCollecte()</i>
	Afficher l'ensemble des collectes sur une période annuelle ou mensuelle	<i>inventaireCollecte()</i>
GestionEboueur	Ajouter un éboueur	<i>creerUnEboueur()</i>
	Supprimer un éboueur	<i>supprimerUnEboueur()</i>
	Calculer le salaire de l'éboueur	<i>salaireEboueur()</i>
GestionEquipe	Créer une équipe à partir d'éboueurs déjà existants	<i>creerUneEquipe()</i>
	Supprimer une équipe	<i>supprimerUneEquipe()</i>
	Affecter un secteur à une équipe	<i>affecterSecteur()</i>
GestionVehicule	Ajouter un véhicule	<i>creerUnVehicule()</i>
	Supprimer un véhicule	<i>supprimerUnVehicule()</i>

-----



## interfaceStatistiques

Catégorie	Fonctionnalité	Méthode à utiliser
StatistiquesCollecte	Poids des déchets mensuel ou annuel	<i>poidsCollecte()</i>
	Volume des déchets mensuel ou annuel	<i>volumeCollecte()</i>
	Pourcentage de collectes effectués par mois ou année	<i>ratioCollecteEffectuee()</i>
	Distance parcourue sur les collectes pendant une année	<i>distanceAnnuelle()</i>
StatistiquesEquipe	Nombre de collectes réalisées d'une équipe en une année	<i>nombreCollecteEquipe()</i>
	Temps de travail d'une équipe mensuel ou annuel	<i>tempsTravailEquipe()</i>
StatistiquesSecteur	Poids des déchets par secteur mensuel ou annuel	<i>poidsParSecteur()</i>
	Volume des déchets par secteur mensuel ou annuel	<i>volumeParSecteur()</i>
	Graphique du poids des déchets par secteur	<i>poidsToutSecteur()</i>
	Graphique du nombre de collectes annulées par secteur	<i>collecteAnnuleeSecteur()</i>

-----