

ECOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS

Département Informatique

64 avenue Jean Portalis

37200 Tours, France

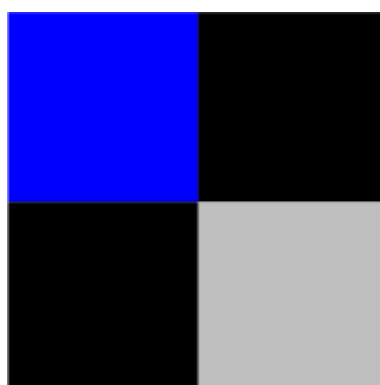
Tél. +33 (0)2 47 36 14 14

[polytech.univ-tours.fr](http://polytech.univ-tours.fr)

## Projet Recherche & Développement

2017-2018

# Amélioration de la plateforme VRP-REP



**Tuteur académique**

Jorge Ernesto MENDOZA

**Étudiant**

Antoine BERNIER (DI5)

2 avril 2018



# Liste des intervenants

Nom	Email	Qualité
Antoine BERNIER	<a href="mailto:antoine.bernier@etu.univ-tours.fr">antoine.bernier@etu.univ-tours.fr</a>	Étudiant DI5
Jorge Ernesto MENDOZA	<a href="mailto:jorge.mendoza@univ-tours.fr">jorge.mendoza@univ-tours.fr</a>	Tuteur académique, Département Informatique



# Avertissement

Ce document a été rédigé par Antoine Bernier susnommé l'auteur.

L'Ecole Polytechnique de l'Université François Rabelais de Tours est représentée par Jorge Ernesto Mendoza susnommé le tuteur académique.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assument l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable du tuteur académique et de l'entreprise.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



## Pour citer ce document

Antoine Bernier, *Amélioration de la plateforme VRP-REP*, Projet Recherche & Développement, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2017-2018.

```
@mastersthesis{
  author={Bernier, Antoine},
  title={Amélioration de la plateforme VRP-REP},
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université François Rabelais de Tours},
  address={Tours, France},
  year={2017-2018}
}
```

# Table des matières

<b>Liste des intervenants</b>	<b>a</b>
<b>Avertissement</b>	<b>b</b>
<b>Pour citer ce document</b>	<b>c</b>
<b>Table des matières</b>	<b>i</b>
<b>Table des figures</b>	<b>ix</b>
<b>Remerciements</b>	<b>1</b>
<b>Introduction</b>	<b>2</b>
1 Contexte de la réalisation.....	3
1.1 Objectifs .....	3
1.2 Base méthodologiques .....	4
<b>1 Description générale</b>	<b>5</b>
1 Terminologies rencontrées sur le site :.....	5
2 Environnement du projet .....	5
3 Fonctionnalités du système .....	5
4 Structure générale du système .....	7
5 Description des interfaces externes du logiciel .....	10
5.1 Interfaces matériel/logiciel .....	10
5.2 Interfaces homme/machine.....	11
5.3 Interfaces logiciel/logiciel .....	11
5.3.1 Système de gestion de la base de données .....	11
5.3.2 Bibliothèques logicielles.....	12

<b>2</b>	<b>État de l'art</b>	<b>13</b>
1	Explication des problèmes « Vehicle Routing Problem » (VRP) .....	13
1.1	Capacited VRP (CVRP).....	13
1.2	VRP with Time Windows (VRPTW) .....	14
1.3	VRP with Backhauls (VRPB) .....	14
1.4	VRP with Pickup and delivery (VRPPD) .....	14
2	Problème multicritère .....	14
<b>3</b>	<b>Veille technologique</b>	<b>17</b>
1	Symfony2 .....	17
1.1	Qu'est-ce qu'un Framework? .....	17
1.2	Architecture d'un projet Symfony2 .....	18
1.3	Explication de l'architecture MVC sous Symfony2.....	18
1.4	Le contrôleur.....	18
1.5	Les vues.....	19
1.6	Modèles .....	19
1.7	Les Bundle .....	20
1.8	Gestion des dépendances.....	21
<b>4</b>	<b>Analyse et conception</b>	<b>24</b>
1	Correction des vendors.....	24
1.1	Analyse de la tâche .....	24
1.2	Conception d'une solution de correction des vendors.....	24
2	Problème d'inscription de bot sur le site .....	25
2.1	Analyse de la tâche .....	25
2.2	Choix de la solution anti-bot.....	25
3	Passage d'une solution monocritère à multicritère.....	26
3.1	Analyse de la tâche .....	26
3.2	Conception de la solution.....	26
3.2.1	Conception du XML .....	26
3.2.2	Conception de la base de données .....	27
3.2.3	Conception des vues.....	27
4	Conception d'un script nettoyant le répertoire tmp .....	28
4.1	Analyse de la tâche .....	28
4.2	Conception de la solution.....	28
5	Page de recherche des données du site.....	29
5.1	Analyse de la tâche .....	29
5.2	Conception de la solution.....	29
6	Mise en évidence des mots-clés lors d'une recherche dans les datatable .....	30
6.1	Analyse de la tâche .....	30

6.2	Conception de la tâche .....	30
7	Affichage de différentes statistiques du site .....	31
7.1	Analyse de la tâche .....	31
7.2	Conception de la solution.....	31
8	Tri des données affichées de manière alphabétique des composants .....	32
8.1	Analyse de la tâche .....	32
8.2	Conception de la tâche .....	32
9	Ajout d'un lien à partir des BKS vers leurs datasets respectifs.....	33
9.1	Analyse de la tâche .....	33
9.2	Conception de la tâche .....	33
10	Ajout d'une fonctionnalité d'envoi de mail .....	33
10.1	Analyse de la tâche .....	33
10.2	Conception de la tâche .....	33
<b>5</b>	<b>Mise en œuvre</b> .....	<b>34</b>
1	Installation du projet.....	34
1.1	Installation d'un environnement sous Linux.....	34
1.2	Installation d'un environnement de développement .....	34
1.3	Installation d'un serveur Apache httpd, Apache2, .....	34
1.4	Installation d'une base de données MariaDB .....	35
1.5	Mise en place de la connexion SSH .....	35
1.6	Récupération du projet Git.....	35
1.7	Configuration des droits sur le projet .....	35
1.8	Récupération des backups du projet .....	36
1.9	Mise en place du Virtual Host .....	36
2	Script de nettoyage du répertoire tmp du serveur.....	36
3	Correction des vendors et du déploiement en bêta .....	37
3.1	Correction des problèmes de déploiement sur la branche develop .....	39
3.1.1	Correction du problème n°1 : fichier de config .....	39
3.1.2	Correction du problème n°2 : migration.....	39
3.2	Correction des vendors.....	42
3.2.1	Correction du problème du Bundle gregwar/captcha.....	42
3.2.2	Correction du problème du Bundle guzzle/guzzle .....	43
3.2.3	Correction du problème des sous-dépendances des vendors.....	43
3.3	Mise en place de la solution Captcha .....	43
4	Ajout d'une page de recherche par mot-clé .....	45
5	Mise en évidence des mots-clés lors d'une recherche dans un datatable.....	47
6	Ajout d'une page de statistique « Statistics » .....	48
7	Correction des données ne s'affichant pas dans l'ordre alphabétique .....	49
8	Ajout d'un lien dans le tableau des BKS vers leur dataset respectif .....	49

<b>6 Bilan</b>	<b>50</b>
<b>Conclusion</b>	<b>51</b>
<b>Annexes</b>	<b>52</b>
<b>A Spécifications fonctionnelles et non fonctionnelles</b>	<b>53</b>
1 Spécifications fonctionnelles.....	53
1.1 Fixer les versions des vendors .....	53
1.1.1 Rôles.....	53
1.1.2 Priorité .....	53
1.1.3 Entrée et sortie.....	53
1.1.4 Composantes.....	53
1.1.5 Traitement.....	54
1.2 Mise en place d'une solution « Captcha » bloquant l'inscription des bots ..	54
1.2.1 Rôles.....	54
1.2.2 Priorité .....	54
1.2.3 Entrée et sortie.....	54
1.2.4 Composantes.....	54
1.2.5 Traitement.....	54
1.3 Ajout d'un menu « Statistics ».....	55
1.3.1 Rôles.....	55
1.3.2 Priorité .....	55
1.3.3 Entrée et sortie.....	55
1.3.4 Composantes.....	55
1.3.5 Traitement.....	55
1.4 Tri des données affichées dans l'ordre alphabétique .....	55
1.4.1 Rôles.....	55
1.4.2 Priorité .....	55
1.4.3 Entrée et sortie.....	55
1.4.4 Composantes.....	56
1.4.5 Traitement.....	56
1.5 Passage d'une solution monocritère à multicritère .....	56
1.5.1 Rôles.....	56
1.5.2 Priorité .....	56
1.5.3 Entrée et sortie.....	56
1.5.4 Composantes.....	56
1.5.5 Traitement.....	56
2 Spécifications non fonctionnelles.....	56



2.1	Contraintes de développement et conception.....	56
2.1.1	Matériel.....	57
2.1.2	Langages .....	57
2.1.3	Bibliothèques .....	57
2.1.4	Environnement .....	57
2.1.5	Protocole de communication.....	57
3	Contraintes de fonctionnement et d'exploitation .....	57
4	Performances .....	58
5	Sécurité .....	58
6	Maintenance et évolution du système.....	58
<b>B</b>	<b>Cahier de développement</b>	<b>60</b>
1	Tâche 1 : Étude des problèmes de tournées de véhicule .....	60
1.1	Description de la tâche .....	60
1.2	Livrables.....	60
1.3	Estimation de charge .....	60
1.4	Temps et date de réalisation .....	60
2	Tâche 2 : Formation Symfony.....	60
2.1	Description de la tâche .....	60
2.2	Estimation de charge .....	61
2.3	Temps et date de réalisation .....	61
3	Tâche 3 : Installation du projet .....	61
3.1	Description de la tâche .....	61
3.2	Estimation de charge .....	61
3.3	Temps et date de réalisation .....	61
4	Tâche 4 : Rédaction du cahier de spécification.....	61
4.1	Description de la tâche .....	61
4.2	Livrables.....	61
4.3	Estimation de charge .....	61
4.4	Temps et date de réalisation .....	61
5	Tâche 5 : Rédaction de l'état de l'art / Veille technologique .....	62
5.1	Description de la tâche .....	62
5.2	Livrables.....	62
5.3	Estimation de charge .....	62
5.4	Temps et date de réalisation .....	62
6	Tâche 6 : Correction des vendors .....	62
6.1	Description de la tâche .....	62
6.2	Estimation de charge .....	62
6.3	Temps et date de réalisation .....	62

7	Tâche 7 : Test des vendors .....	62
7.1	Description de la tâche .....	62
7.2	Estimation de charge .....	63
7.3	Temps et date de réalisation .....	63
8	Tâche 8 : Correction des erreurs Symfony .....	63
8.1	Description de la tâche .....	63
8.2	Estimation de charge .....	63
8.3	Temps et date de réalisation .....	63
9	Tâche 9 : Correction de la barre de recherche des Best Known Solution .....	63
9.1	Description de la tâche .....	63
9.2	Estimation de charge .....	63
9.3	Temps et date de réalisation .....	63
10	Tâche 10 : Tri des données affichées de manière alphabétique des solutions (menu « contribute »), des références de la table d’affichage des instances .....	64
10.1	Description de la tâche .....	64
10.2	Estimation de charge .....	64
10.3	Temps et date de réalisation .....	64
11	Tâche 11 : Ajout d’un Bundle CAPTCHA.....	64
11.1	Description de la tâche .....	64
11.2	Estimation de charge .....	64
11.3	Temps et date de réalisation .....	64
12	Tâche 12 : Passage à des solutions multicritère de minimisation ou maximisation .	64
12.1	Description de la tâche .....	64
12.2	Estimation de charge .....	65
12.3	Temps et date de réalisation .....	65
13	Tâche 13 : Ajout d’un menu statistique « Statistics » .....	65
13.1	Description de la tâche .....	65
13.2	Estimation de charge .....	65
13.3	Temps et date de réalisation .....	65
14	Tâche 14 : Rédaction d’un document d’installation .....	65
14.1	Description de la tâche .....	65
14.2	Estimation de charge .....	65
14.3	Temps et date de réalisation .....	65
15	Tâche 15 : Rédaction du rapport final .....	66
15.1	Description de la tâche .....	66
15.2	Livrables.....	66
15.3	Estimation de charge .....	66
15.4	Temps et date de réalisation .....	66
16	Tâche 16 : Script de nettoyage du répertoire tmp .....	66

16.1	Description de la tâche .....	66
16.2	Estimation de charge .....	66
16.3	Temps et date de réalisation .....	66
17	Tâche 17 : Module de recherche page search.....	66
17.1	Description de la tâche .....	66
17.2	Estimation de charge .....	67
17.3	Temps et date de réalisation .....	67
18	Tâche 18 : Mise en évidence des mots-clés lors d'une recherche dans un datatable	67
18.1	Description de la tâche .....	67
18.2	Estimation de charge .....	67
18.3	Temps et date de réalisation .....	67
19	Tâche 19 : Correction des problèmes de déploiement sur la branche develop.....	67
19.1	Description de la tâche .....	67
19.2	Estimation de charge .....	67
19.3	Temps et date de réalisation .....	67
20	Tâche 20 : Ajout d'un lien à partir des BKS vers leur dataset respectif .....	68
20.1	Description de la tâche .....	68
20.2	Estimation de charge .....	68
20.3	Temps et date de réalisation .....	68
21	Tâche 21 : Ajout d'une fonctionnalité d'envoi de mail.....	68
21.1	Description de la tâche .....	68
21.2	Estimation de charge .....	68
21.3	Temps et date de réalisation .....	68
<b>C</b>	<b>Planning prévisionnels</b>	<b>69</b>
1	Méthodologie .....	69
2	Diagramme de Gantt .....	69
<b>D</b>	<b>Cahier de développeur</b>	<b>72</b>
1	Langage, librairies et convention de nommage .....	72
2	Gestionnaire de version.....	72
3	Conseils de développement.....	72
4	Commentaires.....	73
5	Structure du système .....	73
6	Fonctionnement de Symfony2.....	73
7	Installation.....	73
<b>E</b>	<b>Documentation d'installation du projet</b>	<b>74</b>
1	Avant de commencer .....	74
2	Installation.....	74

2.1	Mise en place d'une connexion au VPS.....	74
2.2	Installation de l'environnement .....	74
2.3	Installation du projet .....	75
2.4	Installation des Backup .....	76
3	Déploiement des modifications sur le serveur .....	77
<b>F</b>	<b>Documentation utilisateur</b>	<b>78</b>
1	Page de recherche .....	78
2	Page Statistics.....	80
<b>G</b>	<b>Documents de tests</b>	<b>82</b>
1	Plan de tests .....	82
2	Grille de test .....	82
	<b>Comptes rendus hebdomadaires</b>	<b>84</b>
	<b>Webographie</b>	<b>86</b>
	<b>Bibliographie</b>	<b>88</b>

# Table des figures

## Introduction

1	Logo du laboratoire d'informatique de Tours.....	2
---	--	---

## 1 Description générale

1	Diagramme de cas d'utilisation des utilisateurs connectés et non-connectés. ....	6
2	Diagramme d'utilisation d'un administrateur.....	7
3	Schéma modélisant l'installation du projet. ....	8
4	Diagramme de classe du projet.[2].....	9
5	Schéma Symfony expliquant le principe simplifié du fonctionnement du site. [15]	10
6	Schéma de fonctionnement de Symfony2. [15].....	11

## 2 État de l'art

1	Front de Pareto. [23].....	15
2	Tableau de problèmes multicritère. [1].....	16

## 3 Veille technologique

1	Architecture d'un projet Symfony. [2] .....	18
2	Exemple de code Twig.....	19

## 4 Analyse et conception

1	Version PHP du serveur.....	25
2	Page des Best Known Solutions (BKS).....	27
3	Fenêtre modale affichant les résultats. ....	28
4	Capture de la barre de recherche .....	29

5	Mock de la page de recherche. ....	30
6	Mock de la page de statistiques Statistics .....	31
7	Capture du problème de tri dans le menu contribute. ....	32
8	Capture du problème de tri dans le tableau des BKS. ....	32
<b>5</b>	<b>Mise en œuvre</b>	
1	Consigne de connexion ssh. [2] .....	35
2	Fichier VirtualHost. ....	36
3	Capture de la table migration de la branche develop. ....	40
4	Capture de la table migration de la branche master. ....	40
5	Capture de la page d'inscription. ....	44
6	Capture de la page contactez nous. ....	45
7	Capture de la page de recherche. ....	47
8	Capture du rendu de datatable.mark.js. ....	48
9	Page statistique « Statistics ». ....	48
<b>A</b>	<b>Spécifications fonctionnelles et non fonctionnelles</b>	
1	Script du fichier post-receive. ....	59
<b>C</b>	<b>Planning prévisionnels</b>	
1	Diagramme de Gantt .....	70
2	Diagramme de Gantt final. ....	71
<b>F</b>	<b>Documentation utilisateur</b>	
1	Guide d'utilisateur : page d'accueil du site. ....	78
2	Guide d'utilisateur : page « Search ». ....	79
3	Guide d'utilisateur : page « Search » dépliées. ....	79
4	Guide d'utilisateur : page d'accueil du site. ....	80
5	Guide d'utilisateur : page « Statistics ». ....	81
<b>G</b>	<b>Documents de tests</b>	
1	Première partie de la grille de test. ....	83
2	Deuxième partie de la grille de test. ....	83



# Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon projet de recherche et développement et qui m'ont aidé lors de la rédaction de ce rapport.

Tout d'abord, j'adresse mes remerciements à mon encadrant, M. Jorge Mendoza, de l'école Polytech Tours pour l'intérêt qu'il a porté à la réussite de ce projet.

Je tiens également à remercier Valentin Maerten pour ses conseils et pour son aide à la correction des problématiques rencontrées...

Enfin je tiens à remercier M. Ameer Soukhal pour l'aide qu'il a apportée à la réalisation de mon cahier de spécifications.

# Introduction

Le Projet de Recherche & Développement (PR&D), réalisé en 5ème année, s'inscrit dans la formation dispensée à l'École Polytechnique de l'Université de Tours et constitue une réelle expérience en termes de conduite de projet. Le PR&D se déroule du 20 septembre 2017 au 30 mars 2018 à raison de deux jours par semaine à temps plein. Il donne lieu à la rédaction d'un rapport avec le cahier de spécifications et à deux soutenances durant lesquelles le travail effectué sera exposé.

Le PR&D est orienté vers une amélioration de la plateforme WEB de résolution de problèmes de tournées de véhicules VRP-REP. Le projet s'inscrit dans un cadre théorique qui fait l'objet d'une étude veille technologique en amont, pour aboutir au développement d'une application. Ce document définit les besoins et l'environnement du projet, ainsi que les objectifs à réaliser. Ce rapport présente les différentes tâches à effectuer et le planning prévisionnel mis en œuvre.

## Les acteurs du projet sont :

- Maitrise d'œuvre (MOE) :
  - Antoine BERNIER, étudiant en 5ème année à l'École d'Ingénieurs Polytech Tours au sein du département Informatique, dans le cadre du PR&D.
  - Les encadrants de l'équipe de recherche ROOT du Laboratoire Informatique de Tours.



**Figure 1** – Logo du laboratoire d'informatique de Tours.

Les recherches menées au sein de cette équipe portent sur l'étude et la résolution des problèmes d'ordonnancement et de planification, que ce soit dans le cadre de problématiques "industrielles" ou théoriques. Il s'agit donc de problèmes d'optimisation, bien souvent combinatoires, pour lesquels des techniques de Recherche Opérationnelle et d'Aide à la Décision sont mises en œuvre ([www.li.univ-tours.fr](http://www.li.univ-tours.fr))



- Maitrise d'ouvrage (MOA) :
  - Jorge E. Mendoza, membre du Laboratoire ROOT d'Informatique.

Ce document a été rédigé par Antoine BERNIER et sera validé par les différents acteurs du projet.

## 1 Contexte de la réalisation

Le projet VRP-REP (Vehicule Routing Problem – Repository) a pour but de fournir à la communauté scientifique une plateforme collaborative de partage de données pour les problèmes de tournées de véhicules. Cette plateforme comporte un nombre important de services facilitant la recherche et la gestion de données, notamment : le téléchargement de jeux de données, le téléchargement de tableaux de résultats pour les jeux de données enregistrés dans la plateforme et la recherche de références bibliographiques par jeux de données ou par type de problèmes.

VRP-REP comporte aussi un nombre d'outils facilitant le traitement des données, parmi lesquels on peut citer : un format universel XML pour la représentation des jeux de données et des bibliothèques de code open source pour la lecture/écriture de données et la vérification de solutions.

Le projet a débuté en 2012. Depuis, plus de 20 personnes de 5 pays différents (étudiants, développeurs, et chercheurs) ont contribué au développement de la plateforme. La version 1.0.0 a été mise en ligne en décembre 2014. Aujourd'hui la plateforme compte 191 utilisateurs actifs venant de 47 pays différents, plus de 35 jeux de données pour une dizaine de types de problèmes et plus de 1 300 solutions pour les instances de ces jeux de données.

URL : <http://www.vrp-rep.org>

### 1.1 Objectifs

La réalisation de cette plateforme a commencé en 2012 dans le cadre du stage d'Hubert Lobit. Suite à celui-ci, différents projet ont permis l'amélioration de cette plateforme, de ce fait l'existant est assez conséquent.

VRP-REP fonctionne sous le langage PHP sous le Framework Symfony 2, avec les bibliothèques TWIG pour le front et Doctrine comme ORM (Mapping objet relationnel) gérant la base de données SQL. Par soucis de budget, un seul serveur OVH héberge les parties production et développement.

L'objectif de ce projet est d'améliorer la plateforme WEB VRP-REP, tout en facilitant sa reprise. Voici une liste non exhaustive des principales tâches à réaliser :

- Ajout d'un module CAPTCHA permettant de bloquer l'inscription des bots ;
- Modification de la fonctionnalité gérant les solutions pour permettre l'ajout de différents critères à améliorer, cette modification affectera le XML et la base de données ;
- Ajout d'un menu « Statistics » pour afficher des statistiques ;
- Réalisation d'une documentation d'installation de projet ;
- Tri des données affichées de manière alphabétique des solutions (menu « contribute »), ainsi qu'au survol des références de la table d'affichage des instances ;
- Faire un lien à partir du tableau des Best Known Solution vers leur datasets respectifs ;
- Réalisation d'un script de nettoyage du dossier tmp du serveur ;
- Etc...

De plus lors des différents projets précédents, certaines erreurs ont été commises et devront donc être corrigées afin de pouvoir procéder à l'amélioration de la plateforme :

- Fixer les versions des vendors ;
- Erreur barre de recherche des dataset par alias => remplacement par un nouveau module de recherche.

## 1.2 Base méthodologiques

Pour la réalisation du projet, la méthodologie agile Scrum a été sélectionnée avec des sprints hebdomadaires se terminant par un rendez-vous avec l'encadrant pour parler de l'avancée du projet. Nous utiliserons un tableau Trello pour voir l'avancement des tâches :

- Ressources, section avec les liens de différentes ressources, PDF ;
- To do, section avec les tâches à réaliser ;
- Doing, section avec les tâches en cours de réalisation ;
- Done, section avec les tâches réalisées ;

Les sprints dureront une semaine et donneront lieu à une réunion avec M. Mendoza chaque jeudi.

Le versionnage du projet se fera avec un dépôt Git hébergé sur le serveur OVH du site (via connexion SSH) sous l'architecture Git flow.

# 1

## Description générale

### 1 Terminologies rencontrées sur le site :

Voici les différentes terminologies rencontrées sur le site :

**VRP** : *Vehicle Routing Problem*, en français problème de tournées de véhicules.

**References** : une référence représente un article dans lequel une méthode de résolution est proposée pour une ou plusieurs variante(s).

**Problem variants** : Une variante de problème correspond à une sous-classe d'un problème de tournées de véhicules.

**Instance** : représente une définition du problème avec les différents nœuds, arcs ainsi que la fonction objective.

**Dataset** : représente un ensemble de solutions. Un dataset est lié à une variante ainsi qu'à une référence.

**Solution** : représente la liste des fonctions objectives pour chaque instance d'un dataset. Une solution est donc liée à un dataset, à une variante, et pour finir à une référence.

**BKS tables** : BKS pour Best Known Solutions est la table des meilleures solutions pour le couple dataset et variante.

### 2 Environnement du projet

La plateforme collaborative VRP-REP (Vehicle Routing Problem – Repository) a pour but de partager les données des recherches sur les problèmes de tournées de véhicules. Un serveur est mis à disposition pour l'hébergement du site et de ses données ainsi que pour la gestion de son versionnage. Nos modifications seront ajoutées à l'existant du site VRP-REP, certaines modifications toucheront l'existant, c'est pourquoi un passage par la branche de développement est nécessaire pour effectuer tous les tests avant la mise en production.

### 3 Fonctionnalités du système

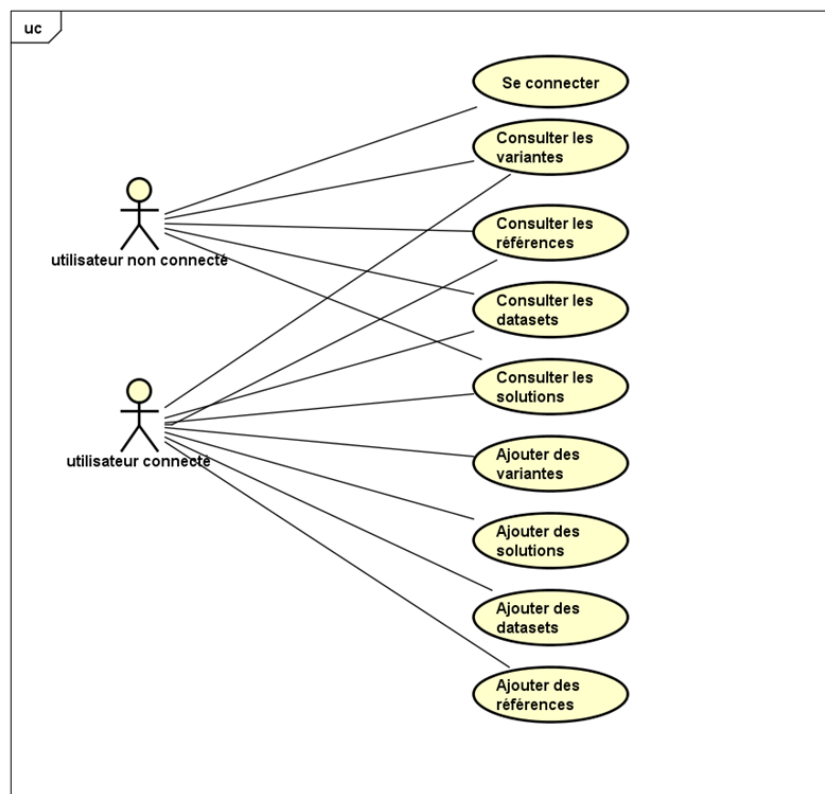
On peut différencier 3 types d'acteurs qui auront chacun des autorisations différentes sur la plateforme web :

- Utilisateur non-connecté : utilisateur ayant accès aux différents datasets, solutions et autres ressources du site ;
- Utilisateur connecté : utilisateur ayant les mêmes accès que l'utilisateur non-connecté mais ayant en plus le droit de contribuer aux données de la plateforme (par exemple proposer un dataset, une solution, ... ) ;
- Administrateur : peut en plus des droits de l'utilisateur connecté, gérer les utilisateurs, groupes, checkers, datasets, datasets utilisés, instances, méthodes, références, solutions, variantes.

Afin de normaliser les datasets et résultats des solutions de VRP-REP dont les utilisateurs connectés vont contribuer, un modèle XSD a été proposé par Monsieur Mendoza et est disponible pour vérifier le jeu de données. Il est disponible à l'URL suivant : [8].

On peut néanmoins déposer un dataset et des solutions sur un format ne respectant pas ce XSD. Les XML non-valides ne sont pas acceptés.

Voici le schéma des cas d'utilisation de l'utilisateur connecté et non-connecté. Il met en évidence les différentes fonctionnalités de l'application VRP-REP :



powered by Astah

**Figure 1** – Diagramme de cas d'utilisation des utilisateurs connectés et non-connectés.

Lorsque l'utilisateur se connecte, il passe selon le rôle de son compte en utilisateur connecté ou en administrateur. Un utilisateur connecté est un utilisateur qui s'est enregistré sur le site. Pour la gestion des utilisateurs le bundle « UserBundle » de Sonata a été utilisé.

L'administrateur, quant à lui, a tous les droits d'un utilisateur connecté en plus de la gestion de différentes parties du site. Le rôle d'administrateur est géré par le bundle Sonata « AdminBundle ». Voici les fonctionnalités du rôle administrateur :

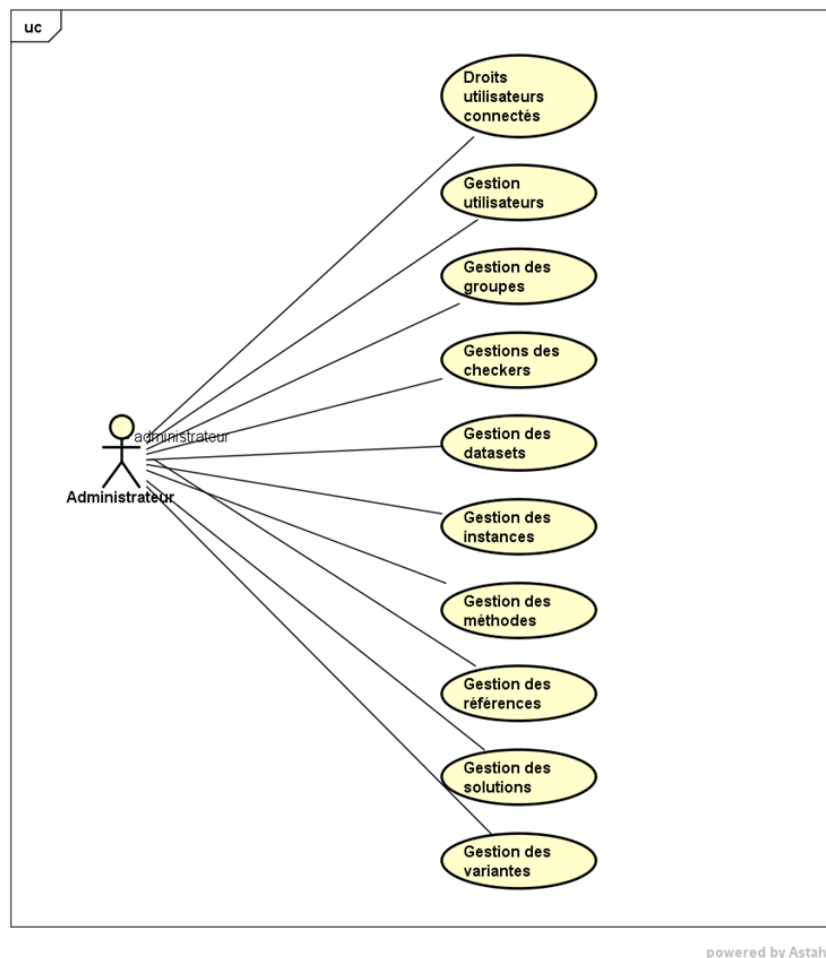


Figure 2 – Diagramme d'utilisation d'un administrateur.

## 4 Structure générale du système

L'existant hébergé sur le serveur OVH est structuré de la manière suivante :

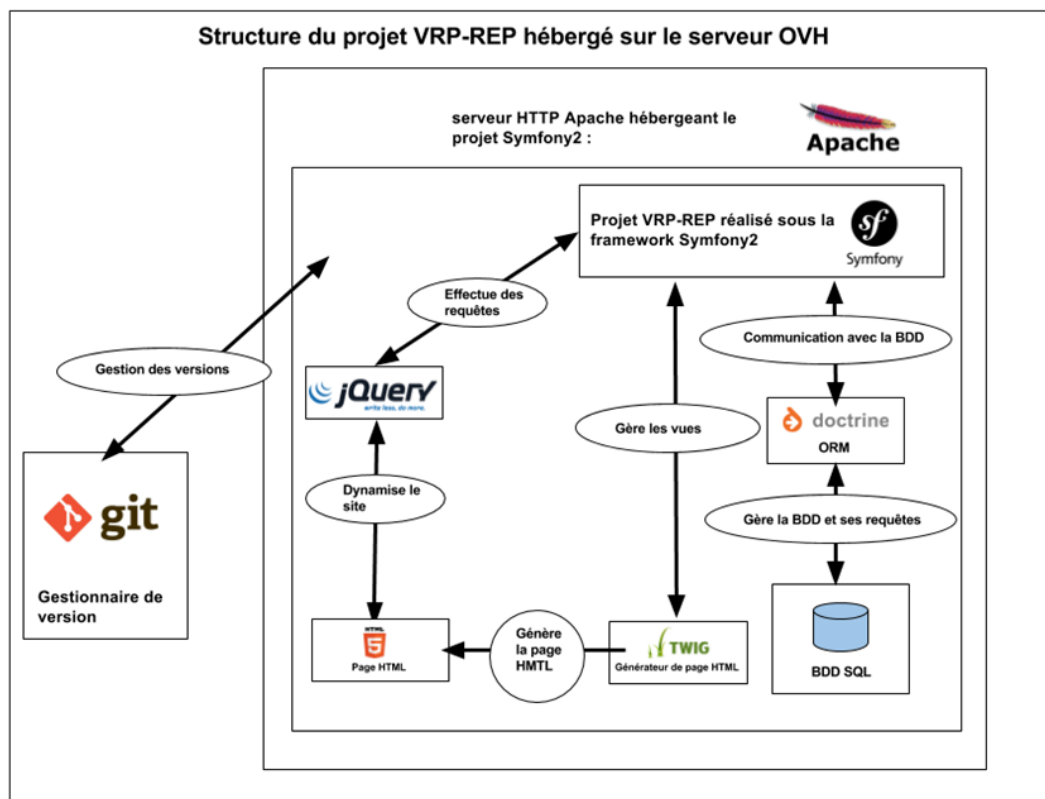
- Hébergement :
  - Serveur VPS OVH hébergeant la partie production et développement (connexion via ssh).
- Langages utilisés :
  - PHP avec le Framework Symfony 2;
  - SQL;
  - HTML;
  - CSS;
  - JavaScript.
- Gestion de la base de données :
  - Doctrine.
- Gestion des vues :
  - TWIG.

Le framework fonctionne de manière MVC :

- Symphony gère la partie contrôleur ;
- Twig gère la partie vue ;
- Doctrine gère la partie Modèle.

Pour plus de détails, se référer à la partie de la veille technologique : **Section 1** (Chapitre 3) - **Symfony2**

Le schéma ci-dessous nous permet d'avoir une vue d'ensemble du fonctionnement du projet :

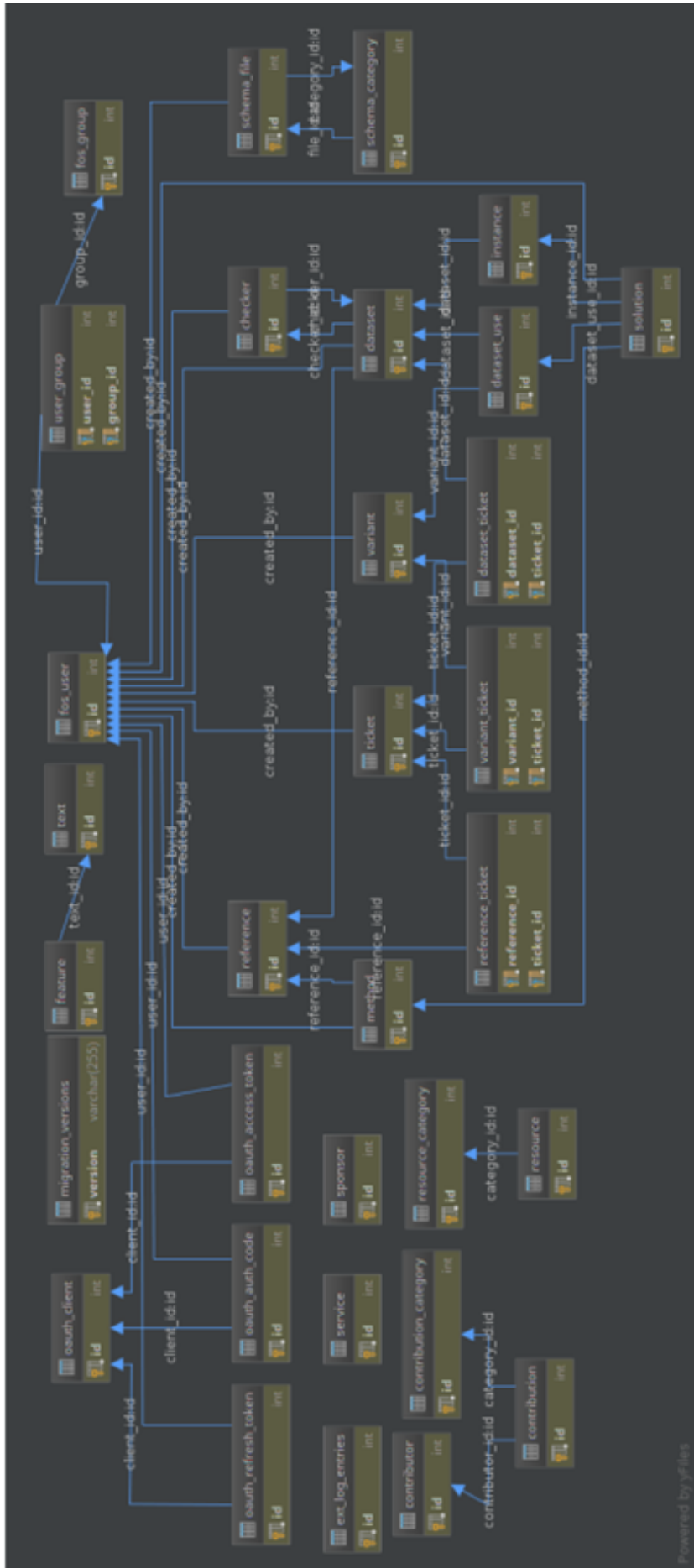


**Figure 3 – Schéma modélisant l'installation du projet.**

**Légende :**

- Les carrés représentent une technologie.
- Les cercles correspondent à une action entre deux composants.

Actuellement, le système est modélisé en diagramme de classe de la façon suivante. Celui-ci évoluera bien évidemment avec les modifications que je vais réaliser.



**Figure 4** – Diagramme de classe du projet.[2]

Voici la représentation des différentes classes :

**Dataset** : représente un ensemble d'instances lié à différentes variantes d'un problème similaire. Il est forcément lié à une méthode.

**Method** : regroupe les différents algorithmes de résolution de problème. Chaque méthode a sa référence, ainsi qu'une liste de solutions trouvées à partir de cette méthode.

**Reference** : représentation d'un article expliquant des algorithmes résolvants des problèmes de tournées de véhicules. À partir d'une référence nous pouvons avoir les *dataset* et *method* associés.

**Instance** : représentation d'un type de problèmes appartenant à un *dataset* et étant composé d'un fichier décrivant son graphe en différents nœuds et arcs du problème.

**Contributor** : représente une personne ayant participé dans le site, ayant publié des *dataset*, des variantes de problèmes, ayant développé le site, etc.

**Contribution** : représente la contribution que le *contributor* a apporté (développement du site, ajout de variantes de problème, etc).

**Ticket** : représente les tickets envoyés par les utilisateurs au support.

**Sponsor** : personne ou organisation soutenant le projet.

**Ressource** : différentes ressources (tutoriels) mises à disposition par le comité pour utiliser le site.

## 5 Description des interfaces externes du logiciel

### 5.1 Interfaces matériel/logiciel

VRP-REP est une plateforme Web, en tant qu'utilisateur une connexion internet ainsi qu'un navigateur sont donc nécessaires pour communiquer avec le serveur web.

**Étape 1** : [15] Le serveur reçoit la requête HTTP de l'utilisateur par le biais du navigateur de l'utilisateur et répond en conséquence :

- **GET** : requête permettant de récupérer des données ;
- **POST** : requête créant une ressource sur le serveur (ex: création d'une solution) ;
- **PUT/PATCH** : requête modifiant une ressource sur le serveur (ex: modification du nom de l'utilisateur) ;
- **DELETE** : requête supprimant une ressource sur le serveur (ex: suppression d'un utilisateur).

**Étape 2** : [15] Quand le serveur a compris la requête et vérifié si l'utilisateur a les droits, il lui retourne une *Response* avec la ressource demandée (par exemple la liste des solutions).

Ce schéma résume bien le fonctionnement de ces deux étapes :

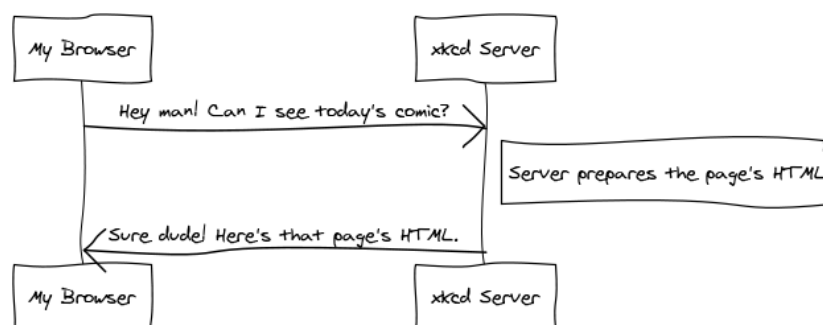


Figure 5 – Schéma Symphony expliquant le principe simplifié du fonctionnement du site. [15]



Voici le schéma expliquant le fonctionnement plus détaillé de Symfony2 :

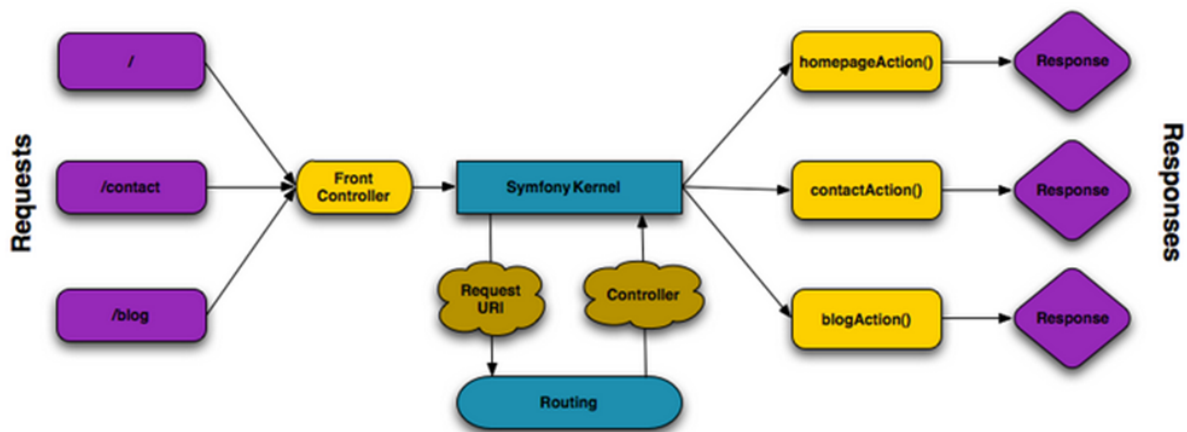


Figure 6 – Schéma de fonctionnement de Symfony2. [15]

Le fonctionnement de la partie côté serveur est de la sorte, l'URL envoyée à partir de notre navigateur est reçue par le « controller frontal », qui définit notre environnement de travail :

- En développement : `app_dev.php` (Le mode développement donne plus de précisions sur les erreurs, un utilisateur lambda n'en n'aura pas besoin);
- En production : `app.php`.

Ensuite le noyau de Symfony va analyser le routing à partir de notre URL (ex : `/menu`) et nous renvoyer vers le Controller qui nous redirige vers la page en fonction de cette requête URL (par exemple l'accès à la page menu). Si l'URL est mauvaise ou que nous n'avons pas les droits, le Controller nous redirige vers la page d'erreur.

## 5.2 Interfaces homme/machine

L'interface homme/machine est affichée via le navigateur internet utilisé par l'utilisateur (Chrome, Mozilla...). Les pages sont générées par le serveur en fonction de l'URL et de nos droits d'utilisateurs (rôle). Les pages sont envoyées à notre navigateur qui nous les affiche.

Les différents profils d'utilisateur sont décrits plus précisément dans la partie [Section 3 - Fonctionnalités du système](#).

## 5.3 Interfaces logiciel/logiciel

### 5.3.1 Système de gestion de la base de données

Le site étant un « repository », il sollicite très régulièrement (quasiment à chaque page), la base de données pour récupérer les données à afficher. L'accès à la base de données située sur le même serveur que le site, se fait via le bundle Doctrine. Cet outil va générer les tables de la base de données à partir des classes Doctrine et de son annotation « Entity ».

Il va également permettre de récupérer à partir d'un *repository*, différentes méthodes génériques permettant d'effectuer des requêtes à la base de données (tel que `find()`), ou bien d'en créer si la requête devient plus compliquée. Pour plus de détails sur son fonctionnement : [Section 1.6](#) (Chapitre 3) - [Modèles](#)

### 5.3.2 Bibliothèques logicielles

En Symfony, les bibliothèques logicielles sont les « Bundle », pour les ajouter, il suffit de renseigner leur nom et la version désirée dans le fichier *package.json*. Pour plus de détails, aller dans la partie de la veille technologique : [Section 1.7](#) (Chapitre 3) - [Les Bundle](#) et [Section 1.8](#) (Chapitre 3) - [Gestion des dépendances](#).

# 2

## État de l'art

### 1 Explication des problèmes « Vehicle Routing Problem » (VRP)

Pour mieux comprendre l'utilité de la plateforme, il est utile de rappeler ce qu'est le *Vehicle Routing Problem*. [3] La résolution des différents problèmes tels que « VRP » par l'informatique est en large expansion depuis les 30 dernières années. De manière générale, on constate que l'utilisation de l'informatique dans le processus de distribution d'une entreprise permet d'économiser généralement 5 à 20% dans les coûts de transport.

Les cas d'application réels du « VRP » sont le ramassage des poubelles, le nettoyage des rues, les trajets des bus scolaires, le transport des personnes handicapées, les trajets du voyageur de commerce et la gestion des unités de maintenance. Ces problèmes de transport de biens sont décrits comme un service effectué dans une période de temps donné, avec un groupe de véhicules et un groupe de conducteurs qui suivent les différentes contraintes du réseau routier.

Ils sont généralement représentés par des graphes où les arcs représentent les routes et où les sommets correspondent aux différents points de jonction entre la position des opérants et les différents points de dépôt. À chaque arc, on associe une direction (certaines routes n'ont qu'un seul sens de direction), ainsi qu'un coût (généralement en temps) qui dépend principalement du véhicule et de la période au cours de laquelle l'arc est traversé. On peut dénombrer différentes variantes de problème de tournées de véhicules qui ont pour chaque cas des critères différents. Voici quelques exemples ...

#### 1.1 Capacited VRP (CVRP)

Pour le CVRP, on définit les critères suivants :

- Chaque client correspond à une livraison. Cette livraison est déterminée à l'avance (on la connaît) et ne peut être divisée.
- Tous les véhicules sont identiques (mêmes critères de déplacement : vitesse, route empruntées...) et sont basés dans un seul et unique entrepôt.
- Les véhicules ont pour contrainte d'avoir une capacité maximale à ne pas dépasser.
- Les coûts de déplacement de chaque arc sont positifs.

L'objectif de résolution de ce problème est de livrer chaque client en minimisant le coût total de transport. Tous les critères ci-dessus doivent être bien évidemment respectés. Il y a donc un

sous-problème dans ce problème qui est le chargement du véhicule. Celui-ci modifiera le trajet de par la contrainte de la capacité.

### 1.2 VRP with Time Windows (VRPTW)

Le VRPTW est une classe de problème qui hérite des critères du CVRP. En plus de cela, nous rajoutons la contrainte « Time Windows » qui pour chaque client définit un intervalle de temps pendant lequel nous pouvons effectuer la livraison, il nous faut aussi prendre en compte le temps de livraison. Pour résumer :

- Chaque circuit passe par le sommet du dépôt ;
- Chaque client doit être livré par un seul et unique circuit ;
- La capacité du véhicule (autonomie) ne doit pas être dépassée par son circuit ;
- Chaque client est disponible pendant un intervalle de temps appelé « Time Windows », et chaque véhicule s'arrête pendant un temps de livraison définit.

### 1.3 VRP with Backhauls (VRPB)

Le VRPB est lui aussi une extension du CVRP pour lequel l'ensemble du problème des clients se compose de deux sous-parties :

- La première partie est d'effectuer pour chaque trajet, toutes les livraisons de client ;
- La seconde partie est de récupérer des quantités de produit des clients lors du retour.

Les conditions sont les suivantes :

- Chaque circuit passe par le dépôt ;
- Chaque client est parcouru par un seul et unique circuit ;
- La capacité totale du véhicule ne doit pas être dépassée lors de la livraison ou de la récupération des produits ;
- Pour chaque circuit la livraison des clients (s'il y en a) précède celle des retours.

### 1.4 VRP with Pickup and delivery (VRPPD)

La version la plus simple du VRPPD associe à chaque client deux quantités : livrable (delivery) et de retrait (pickup). À chaque passage chez un client le transporteur se décharge de la quantité livrable et récupère la quantité de retrait. Le but est donc de diminuer au maximum le coût du trajet. Pour résumer, les critères sont les suivants :

- Chaque circuit passe par le dépôt ;
- Chaque client est visité par un seul et unique circuit ;
- Lors de la tournée, le chargement de chaque véhicule ne peut être négatif ou supérieur à la capacité ;
- On peut définir au préalable un client qui doit être servi **avant** les autres ;
- On peut définir au préalable un client qui doit être servi **après** les autres.

## 2 Problème multicritère

Ayant pour tâche de modifier le système de gestion des solutions du site, il est important de comprendre ce qu'est un problème multicritère. En règle générale, les problèmes VRP étant

complexes, ils sont généralement monocritère, ils essaient de minimiser un critère : temps, coûts, flotte ...

Mais il arrive de rencontrer certains problèmes dit *multicritère*, y compris sur notre plateforme VRP-REP. D'où l'intérêt de la modification que nous allons apporter (voir [Section 1.5](#) (Annexe A) - [Passage d'une solution monocritère à multicritère](#)).

[1] L'objectif de l'approche multicritère est de se rapprocher des contraintes réelles et choisir la solution optimale de plusieurs critères, en appliquant une approche fixant les règles de dominance (ou non). Pour résoudre cette famille de problème, nous utilisons l'« Optimum de Pareto ». Pour mieux comprendre, nous pouvons prendre ce graphique :

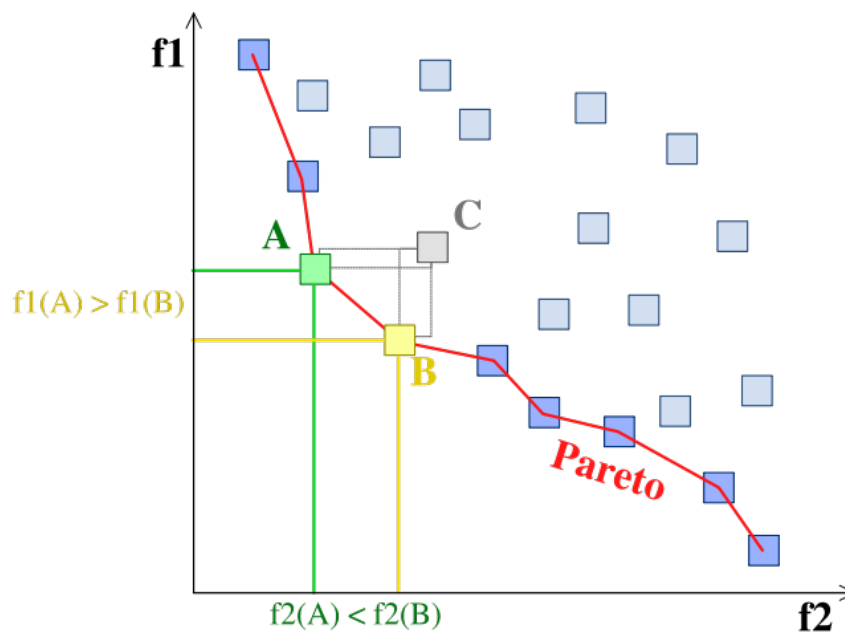


Figure 1 – Front de Pareto. [23]

Comme nous pouvons le voir dans la [Figure 1](#), nous essayons d'obtenir une solution optimale pour  $f1$  (ex: la distance), tout en ayant la meilleure solution possible pour  $f2$  (ex: le nombre de véhicules (flotte)) (dominance de  $f1$  sur  $f2$ ).

Cependant la dominance du potentiel de Pareto n'est pas suffisante pour répondre au problème multicritère. Ces trois approches y répondent :

- A priori : dans l'approche *a priori*, un décideur définit avant la réalisation du processus les dominances entre les critères (lesquels vont être maximisés);
- Interactive : pour cette approche, les choix du décideur sont appliqués lors de la procédure de résolution;
- A posteriori : durant cette approche le décideur utilise un lot de solutions potentiellement non-dominées et sélectionne la solution qui lui convient le mieux.

Voici un tableau regroupant une liste non-exhaustive de problèmes de tournées de véhicules multicritère prolongeant d'un problème monocritère :

Table 1  
Multi-objective routing problems extended from single-objective problems (I)

Authors	Problem	Method	Tour	Nodes/Arcs	Resources
Park and Koelling (1986, 1989) [54,55]	Vehicle routing problem	Goal programming Heuristic	Min. the traveled distance	Max. the realization of urgent queries Max. the station conditional dependence	Min. the merchandise deterioration
Sutcliffe and Board (1990) [63]	Vehicle routing problem	Linear programming	Min. the traveled distance Max. the equalization of the vehicle travel times		Max. the capacity utilization
Current and Schilling. (1994) [12]	Median tour problem Maximal covering tour problem	Heuristics	Min. the total length	Max. access to the tour for the nodes not directly on it	
Lee and Ueng (1998) [45]	Vehicle routing problem	Aggregation Heuristic	Min. the traveled distance Optimize the balance of the load (length)		
Sessomboon et al. (1998) [61]	Vehicle routing problem with time windows	Pareto approach Hybrid genetic algorithm	Min. the traveled distance	Max. the customer satisfaction	Min. the number of vehicles Min. the vehicle waiting times
Hansen (2000) [28]	Multi-objective traveling salesman problem	Scalarizing function Local search	Min. the total lengths		

Figure 2 – Tableau de problèmes multicritère. [1]

# 3

## Veille technologique

Le cahier de spécifications étant très lié à la veille technologique, je vous invite donc à consulter ces deux parties pour plus de détails sur les technologies, ainsi que sur le fonctionnement du site.

### 1 Symfony2

Symfony est un Framework MVC en PHP développé par une équipe web française SensioLabs. SensioLabs : [\[17\]](#), Symfony est un ensemble de composants PHP, un Framework pour application web, une philosophie, ainsi qu'une communauté - le tout fonctionnant en harmonie. Ce puissant Framework permet la réalisation de sites complexes rapidement, de façon structurée avec un code propre modulaire.

#### 1.1 Qu'est-ce qu'un Framework?

En anglais, Framework signifie « cadre de travail ». C'est une boîte à outils qui permet la création de projet sur des fondations et architectures propres (ici MVC). En résumé, il a pour avantage :

- Une communauté active qui utilise le Framework et qui contribue en retour ;
- Une documentation de qualité et régulièrement mise à jour ;
- Un code source maintenu par des développeurs attitrés ;
- Un code qui respecte les standards de programmation ;
- Un support à long terme garanti et des mises à jour qui ne cassent pas la compatibilité.

## 1.2 Architecture d'un projet Symfony2

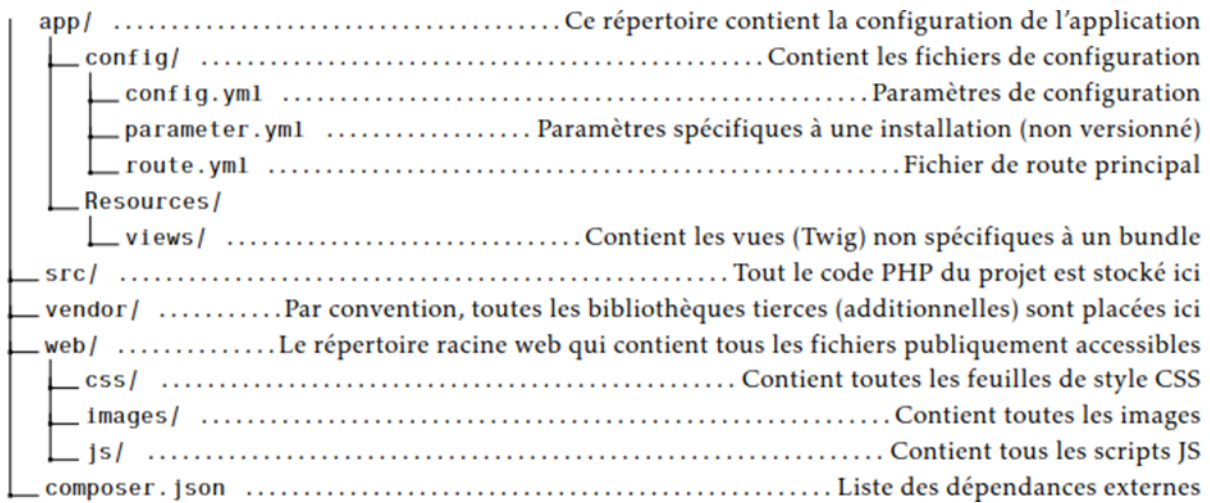


Figure 1 – Architecture d'un projet Symfony. [2]

## 1.3 Explication de l'architecture MVC sous Symfony2

MVC signifie Modèle - Vue - Contrôleur :

- **Le Contrôleur** : son objectif est de générer la réponse à la requête HTTP demandée par l'utilisateur. Il analyse les requêtes puis les traite en conséquence. C'est lui qui va décider quelles vues afficher avec les données qui doivent nous être associées. Il sera ici géré par Symfony.
- **Le Modèle** : son rôle est de modéliser et de gérer les données. Il sera géré par Doctrine.
- **La vue** : son rôle est d'afficher les pages. Le contrôleur lie les données à celui-ci pour qu'il se charge de les afficher. Il sera ici géré par Twig.

## 1.4 Le contrôleur

[12] Le mécanisme de Symfony 2 est expliqué dans la partie [Section 5.1](#) (Chapitre 1) - **Interfaces matériel/logiciel**. Le contrôleur est appelé par une requête HTTP et doit obligatoirement retourner un objet *Response*. Pour envoyer un objet *Response*, il suffit d'utiliser la fonction *render()* qui va générer un template, et l'envoyer dans l'objet *Response*. Voici comment l'utiliser :

```
return $this->render('lucky/number.html.twig', array('name'=>$name))
```

La norme formate le nom du contrôleur et de ses actions sont de la manière suivante [12] :

- XController avec X le nom du controller ;
- YAction avec Y le nom de l'action.

Comme dans l'exemple ci-dessous :

```

1 class LuckyController
2 {
3     /**
4      * @Route("/lucky/number/{max}", name="app_lucky_number")
5      */
  
```



```

6 public function numberAction($max)
7 {
8     $number = mt_rand(0, $max);
9
10    return new Response(
11        '<html><body>Lucky number: '.$number.'</body></html>'
12    );
13 }
14 }

```

## 1.5 Les vues

Les vues de notre projet sont réalisées soit en TWIG soit en PHP. Comme il est indiqué sur la documentation Symfony : [13], le langage TWIG est plus puissant.

Dans notre projet il a été choisi de prendre le langage TWIG (du fait qu'il soit plus performant). En plus des balises HTML qu'il intègre, nous pouvons ajouter les syntaxes suivantes :

- `{{ ... }}` : grâce à ces balises nous pouvons intégrer un variable ou le résultat d'une expression;
- `{% ... %}` : est utilisé pour intégrer du code PHP entre ces balises;
- `{# ... #}` : permet d'ajouter un commentaire.

Il permet aussi d'insérer du code CSS ainsi que du code *JavaScript* à partir de **block** :

```

1  {# app/Resources/views/base.html.twig #}
2  <html>
3      <head>
4          {# ... #}
5
6          {% block stylesheets %}
7              <link href="{{ asset('css/main.css') }}" rel="stylesheet" />
8          {% endblock %}
9      </head>
10     <body>
11         {# ... #}
12
13         {% block javascripts %}
14             <script src="{{ asset('js/main.js') }}"></script>
15         {% endblock %}
16     </body>
17 </html>

```

Figure 2 – Exemple de code Twig.

## 1.6 Modèles

Les modèles de notre application sont gérés via les objets PHP et le bundle Doctrine. [6] Doctrine est un « Object Reational Mapper » pour les projets utilisant PHP 5.4+. En résumé, son utilisation permet de rendre persistant nos objets PHP portant la notation « Entities ». [22] Il utilise le « Data Mapper pattern » qui est une architecture ajoutant une couche supplémentaire

permettant l'accès aux données, en gardant la représentation mémoire (modèle) et le stockage des données persistantes séparées. Pour respecter ce pattern, il doit inclure également les méthodes simples de traitement des données persistantes :

- Create;
- Read;
- Update;
- Delete.

Doctrine peut être inclut directement à la création du projet ou ajouté dans le *composer.json* (voir [Section 1.8 - Gestion des dépendances](#)).

**Création de la base de données :** La base de données génère les tables à partir des classes portant l'annotation « @Entity ». Pour lancer la création de la base de données, il faut exécuter la commande :

```
php app/console doctrine:schema:create
```

ou pour la modification :

```
php app/console doctrine:schema:update --dump-sql
```

**Les entités :** Les entités sont les patrons des tables (leurs modèles), c'est à partir d'elles que lors de la création de la base de données via Doctrine, les tables vont être générées. Elles sont gérées par l'*EntityManager* qui fait le lien entre les entités et la base de données. Voici un exemple de classe PHP portant la notation « @Entity » permettant la sérialisation de ce type d'objet [6] :

```

1 <?php
2 // src/Product.php
3 /**
4  * @Entity @Table(name="products")
5  */
6 class Product
7 {
8     /** @Id @Column(type="integer") @GeneratedValue */
9     protected $id;
10    /** @Column(type="string") */
11    protected $name;
12
13    // .. (other code)
14 }
```

**Enregistrement des données :** Pour enregistrer un nouvel objet, il faut utiliser deux méthodes. La première *persist()* permet d'avertir l'*EntityManager* que nous allons insérer une entité. Voilà un exemple :

```
$entityManager->persist($product);
```

Ensuite après avoir réalisé cette commande nous devons réaliser l'insertion via la commande *flush()* comme ceci :

```
$entityManager->flush();
```

Pour plus de détails, se référer à la documentation de Doctrine : [6].

## 1.7 Les Bundle

Un bundle est similaire à un plugin dans un autre logiciel. La différence majeure est que dans Symfony tout est un bundle : les bases du Framework, ainsi que le code écrit pour notre projet. Pour résumé, un bundle est une structure de fichiers implémentant une fonctionnalité unique.

Chacun de ses répertoires contiennent tout ce qui est relatif à cette fonctionnalité : fichier PHP, templates, fichier de style, fichier JavaScript, tests ainsi que bien d'autres choses...

Son architecture est la suivante :

- Controller/ : Contient les contrôleurs du bundle ;
- DependencyInjection/ : Contient certaines classes d'extension d'injection de dépendance ;
- Resources/config/ : contient les configurations du bundle, y compris celles du fichier routing.yml ;
- Resources/views/ : contient les templates organisés par le nom de leur controller ;
- Ressources/public/ : contient les ressources web « assets » (images, feuilles de style, etc) et son copiées ou liées par un lien symbolique au répertoire du projet /web via la console par la commande « *assets:install* » .

Pour plus de détails sur le principe des bundles, leur installation, voir la documentation de SensioLabs : [16].

## 1.8 Gestion des dépendances

En Symfony, le gestionnaire des dépendances utilisé est celui de PHP et s'appelle composer. Le composer utilise le fichier composer.json pour installer les bundle. Voici notre fichier *composer.json* avec les versions des vendors fixées, il est structuré de la manière suivante :

```

1 {
2     "name": "symfony/framework-standard-edition",
3     "license": "MIT",
4     "type": "project",
5     "description": "The \"Symfony Standard Edition\" distribution",
6     "autoload": {
7         "psr-0": { "": "src/", "SymfonyStandard": "app/" }
8     },
9     "require": {
10         "php": ">=5.3.3",
11         "symfony/symfony": "2.5.7",
12         "symfony/intl": "~2.5",
13         "symfony/swiftmailer-bundle": "2.3.7",
14         "symfony/monolog-bundle": "2.6.1",
15         "symfony/assetic-bundle": "2.5.0",
16         "kriswallsmith/assetic": "1.2.1",
17         "sensio/distribution-bundle": "2.3.5",
18         "sensio/framework-extra-bundle": "3.0.3",
19         "doctrine/orm": "2.4.6",
20         "doctrine/doctrine-bundle": "1.2.0",
21         "doctrine/doctrine-fixtures-bundle": "2.2.0",
22         "doctrine/migrations": "1.0.0",
23         "doctrine/doctrine-migrations-bundle": "1.0.1",
24         "stof/doctrine-extensions-bundle": "1.1.0",
25         "twig/extensions": "1.2.0",
26         "incenteev/composer-parameter-handler": "2.1.0",
27
28         "friendsofsymfony/jsrouting-bundle": "1.5.3",
29         "hwi/oauth-bundle": "0.4.3",
30         "braincrafted/bootstrap-bundle": "2.1.0",
31         "jms/twig-js-bundle": "1.0.0",
32         "willdurand/js-translation-bundle": "v2.1.3",
33
34         "sonata-project/core-bundle": "2.3.1",
35         "sonata-project/admin-bundle": "2.3.10",
36         "sonata-project/doctrine-orm-admin-bundle": "2.3.1",

```

```

37     "sonata-project/user-bundle": "2.2.5",
38     "sonata-project/datagrid-bundle": "2.2.1",
39     "sonata-project/formatter-bundle": "2.3.4",
40
41     "friendsofsymfony/rest-bundle": "1.4.2",
42     "friendsofsymfony/oauth-server-bundle": "1.4.2",
43     "nelmio/api-doc-bundle": "2.7.0",
44     "friendsofsymfony/http-cache-bundle": "1.1.0",
45     "willdurand/hateoas-bundle": "1.0.1",
46     "hautelook/templated-uri-bundle": "2.0.0",
47     "willdurand/rest-extra-bundle": "v1.0.2"
48 },
49 "require-dev": {
50     "sensio/generator-bundle": "v2.4.0",
51     "behat/symfony2-extension": "v2.1.0",
52     "behat/mink-extension": "v2.1.0",
53     "behat/mink-browserkit-driver": "v1.2.0",
54     "behat/mink-zombie-driver": "v1.2.0",
55     "phpunit/phpunit": "4.3.5",
56     "mikey179/vfsStream": "v1.4.0"
57 },
58 "scripts": {
59     "post-root-package-install": [
60         "SymfonyStandard\\Composer::hookRootPackageInstall"
61     ],
62     "post-install-cmd": [
63         "Incenteev\\ParameterHandler\\ScriptHandler::buildParameters",
64         "Sensio\\Bundle\\DistributionBundle\\Composer\\ScriptHandler::buildBootstrap",
65         "Sensio\\Bundle\\DistributionBundle\\Composer\\ScriptHandler::clearCache",
66         "Sensio\\Bundle\\DistributionBundle\\Composer\\ScriptHandler::installAssets",
67         "Sensio\\Bundle\\DistributionBundle\\Composer\\ScriptHandler::installRequirementsFile"
68     ],
69     "post-update-cmd": [
70         "Incenteev\\ParameterHandler\\ScriptHandler::buildParameters",
71         "Sensio\\Bundle\\DistributionBundle\\Composer\\ScriptHandler::buildBootstrap",
72         "Sensio\\Bundle\\DistributionBundle\\Composer\\ScriptHandler::clearCache",
73         "Sensio\\Bundle\\DistributionBundle\\Composer\\ScriptHandler::installAssets",
74         "Sensio\\Bundle\\DistributionBundle\\Composer\\ScriptHandler::installRequirementsFile"
75     ]
76 },
77 "config": {
78     "bin-dir": "bin/"
79 },
80 "extra": {
81     "symfony-app-dir": "app",
82     "symfony-web-dir": "web",
83     "incenteev-parameters": {
84         "file": "app/config/parameters.yml"
85     },
86     "branch-alias": {
87         "dev-master": "2.5-dev"
88     }
89 }
90 }

```

Le *composer.json* fonctionne de deux manières :

- On peut utiliser la commande « *composer install* » qui va lire le *composer.json* et prendre les versions inscrites dans ce fichier.
- On peut également utiliser la méthode « *composer update* » qui va quant à elle prendre tous les bundle contenus dans le fichier *composer.json* et les installer avec leur dernière

version. Cette commande écrasera le fichier

*composer.json* pour en créer un autre. Plus d'informations sur le composer sont disponibles sur le site : [1].

Le projet étant ancien la commande n°2 ne sera pas utilisée, au risque de créer des problèmes de compatibilité. Durant ce projet, nous avons rencontré des erreurs commises dans la gestion des versions des dépendances. En effet certaines consignes sur la gestion des versions ont permis la mise à jour de certains bundle (certaines majeures) et donc de créer des problèmes de compatibilité. Les versions des dépendances sont gérées de la manière suivante : [10]

#### X.Y.Z

**X** : Version majeure, les mises à jour doivent se réaliser manuellement, risques de problème de compatibilité fort entre deux versions majeures.

**Y** : Version mineure, les mises à jour effectuées entre deux versions sont supposées ne pas créer de problème de compatibilité. Par précaution, ces mises à jour s'effectueront manuellement.

**Z** : Version corrective de BUG, les mises à jour ne doivent pas créer de problème de compatibilité.

Ensuite il y a diverses façons de les réaliser :

- En les fixant par exemple 2.3.10 de cette façon lorsque la commande « *composer install* » est réalisée le bundle lié à cette version s'installera dans la version 2.3.10;
- En permettant la mise à jour de certaines versions, par exemple : 2. \* ou 2.1.\*. Ainsi dans le cas 1, il prendra la dernière mineure du bundle et dans le cas 2 la dernière version corrective lors de la commande « *composer install* »;
- Nous pouvons aussi mettre l'annotation '^' qui à l'installation, nous permettra d'obtenir la dernière version supérieure à la version notée. Par exemple supérieure ou égale à la version ^2.3.10;
- Obtenir la dernière version stable avec l'annotation @stable comme par exemple : 2.@stable, ou tout simplement @stable;
- De même avec la @dev qui va donner la dernière version en develop d'une branche (pas stable) tel que @dev-master la dernière version de la branche master (pas stable dans la plupart des cas).

L'ensemble des Bundle est accessible sur la gestion en ligne PHP à partir du site Packagist : [10].

# 4

## Analyse et conception

Dans cette partie, nous allons présenter les différentes analyses effectuées, solutions trouvées ainsi que leurs modélisations pour atteindre les objectifs définis dans la partie **Objectifs**. Cette partie va utiliser nos connaissances acquises et mise en place tout au long du projet. Elle nous aidera lors de la partie mise en œuvre du projet.

### 1 Correction des vendors

#### 1.1 Analyse de la tâche

Le fichier *composer.json* ayant été mal paramétré, il autorise les mises à jour de certains vendor (ou bundle), y compris des mises à jour majeures pouvant créer des problèmes d'incompatibilités entre chaque vendor.

Cela a pour cause d'empêcher la réinstallation en local du projet, cela pourrait être également la cause de certains dysfonctionnements du site actuel dû à différentes mises à jour réalisées par les collaborateurs ayant repris le projet (voir fonctionnement commande *composer install* dans la partie **Section 1.8** (Chapitre 3) - **Gestion des dépendances**). Notamment la version du vendor *sonata-project/admin-bundle* qui requière une version de *sonata-project/core-bundle* supérieure à celle actuelle sur le serveur.

#### 1.2 Conception d'une solution de correction des vendors

Pour cette solution, il nous faut récupérer les bundles autorisant les mises à jour, ce sont ceux ayant les annotations :

- @stable;
- @dev.

Après les avoir récupérés, nous allons récupérer les versions des vendors actuels du site (sur le serveur) qui ont très peu bouger à partir de la commande « *composer show* ».

A partir de cette étape nous avons deux cas de figure :

1. Soit les versions sont disponibles dans la liste des versions récupérées précédemment.

2. Soit nous devons les récupérer à partir du site Packagist [10] avec les informations récupérées via la commande précédente, à partir de l'identifiant du commit de la branche récupérée, ou à partir des informations des versions disponibles (ex : la version 1.2.X quand nous avons 1.2.@dev).

## 2 Problème d'inscription de bot sur le site

### 2.1 Analyse de la tâche

Le site VRP-REP reçoit assez régulièrement l'inscription de bot qui ne valide pas la vérification par email. Cela a pour cause d'obliger les administrateurs de les supprimer à la main. C'est pour cette raison que Monsieur Mendoza m'a donné pour tâche de résoudre ce problème. On peut donc constater que pour éviter ces inscriptions intempestives, il nous faut les empêcher d'atteindre l'étape de validation par email.

### 2.2 Choix de la solution anti-bot

Nous nous sommes donc orienté vers une solution du type CAPTCHA, de part son efficacité contre les bots, ainsi que part son intégration facile dans notre processus d'inscription. Notre solution Captcha doit être compatible avec notre projet :

- Compatible avec la version PHP de notre serveur : 5.6.14 (récupérée avec la commande exécuter « *php -v* » ; voir le résultat : [Figure 1](#));
- Compatible avec la version de notre projet Symfony (voir fichier *composer.json* : [Section 1.8](#) (Chapitre 3) ), il doit également être un bundle ;
- Compatible avec la version de nos vendors actuels.

```
PHP 5.6.14-0+deb8u1 (cli) (built: Oct 4 2015 16:13:10)
Copyright (c) 1997-2015 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2015 Zend Technologies
with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2015, by Zend Technologies
```

Figure 1 – Version PHP du serveur.

A partir du site Packagist [10], nous pouvons voir différents bundle, que nous choisissons en fonction de nos critères et par leur popularité (nous assurant un minimum de qualité):

- **google/recaptcha** : le plus populaire, il ne répond pas à notre demande : ce n'est pas un bundle, et nous ne respectons pas ses prérequis de développement « *phpunit/phpunit: ^4.8* », notre *phpunit/phpunit* étant à la version est 4.3.5;
- **gregwar/captcha-bundle** : [\[captchabundleBiblio\]](#) comme stipulé dans sa documentation, ce bundle en version v1.1 convient pour les Symfony ayant une version < 2.8, c'est pourquoi nous allons le sélectionner.

Nous intégrerons donc le bundle *gregwar/captcha-bundle* en bas de la page d'enregistrement. L'utilisateur ne pourra pas passer à l'étape de validation tant que la vérification CAPTCHA n'est pas validée.

### 3 Passage d'une solution monocritère à multicritère

#### 3.1 Analyse de la tâche

Cette tâche a pour but de prendre en compte les solutions des problèmes VRP à multicritère (voir : [Section 2](#) (Chapitre 2) - [Problème multicritère](#)). Cette modification impliquera de grands changements! En effet, il faudra modifier la structure de différents éléments :

- Fichier XML ;
- Base de données ;
- Vues affichant les solutions sur le site.

#### 3.2 Conception de la solution

##### 3.2.1 Conception du XML

Le XSD actuel (fichier permettant de valider un xml) prend en compte qu'un critère comme vous pouvez le voir dans le XSD récupérable ici : <http://www.vrp-rep.org/resources.html>. Nous devons le modifier pour qu'il accepte plusieurs critères à partir de l'attribut *maxOccurs="unbounded"* permettant d'avoir plusieurs occurrences. Il y aura également d'autres modifications, chaque critère aura un attribut permettant de définir son type (temps, coûts, flotte, ...). Les critères contiendront ces différentes données :

- Value : le résultat du critère ;
- Priority : l'indice de dominance du critère permettant de définir son importance les uns par rapport aux autres.

L'architecture du nouveau XML devra suivre cet exemple :

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <solutions>
3   <solution instance="definir">
4     <criteria type="time" >
5       <value>1000.0</value>
6       <priority>1</priority>
7     </criteria>
8     <criteria type="fleet">
9       <value>10</value>
10      <priority>2</priority>
11    </criteria>
12  </solution>
13  <solution instance="definir2">
14    <criteria type="time" >
15      <value>139.0</value>
16      <priority>2</priority>
17    </criteria>
18    <criteria type="cost">
19      <value>920</value>
20      <priority>1</priority>
21    </criteria>
22  </solution>
23 </solutions>

```



### 3.2.2 Conception de la base de données

Pour la modification de la structure de la base de données, nous allons changer la structure pour qu'une solution ait plusieurs objectifs. La création d'une entité objective sera à réaliser, elle contiendra les attributs suivants :

- Type (ex: time, cost, fleet);
- Value (le résultat);
- Priority (la règle de dominance).

La solution passera d'un objectif à plusieurs objectifs à partir de l'annotation `@OneToMany` [5]. Nous devons ensuite réaliser une migration des anciennes solutions pour ne pas les supprimer ou créer de conflits. Pour cela, doctrine inclut un outil de migration, voir : [14].

### 3.2.3 Conception des vues

Pour la conception des vues, nous réutiliserons les fenêtres déjà présentes sur le site en insérant les données manquantes :

#### Best Known Solutions

This pages allows you to retrieve the best known solutions (BKSs) registered in the VRP-REP database for a given dataset. Please select a problem variant and a dataset, the website will retrieve the BKS for each of the instances of the dataset and display it in the table above. You can sort the table by clicking on the table headers. You can also search the table (for instance for a specific instance) using the search bar in the top right of the table.

* Variant	CVRP   Capacitated Vehicle Routing Problem
* Dataset	Christofides et al. 1979 - CMT

Show	10	entries	Search
------	----	---------	--------

Instance	O.F.	Optimality	References
CMT01	524.61	Proven	9 references
CMT02	835.26	Proven	7 references
CMT03	826.14	Proven	9 references
CMT04	1028.42	Proven	5 references
CMT05	1291.29	Proven	2 references
CMT06	555.43	Unproven	9 references
CMT07	909.68	Unproven	9 references
CMT08	865.94	Unproven	7 references
CMT09	1162.55	Unproven	7 references
CMT10	1395.85	Unproven	4 references

Showing 1 to 10 of 14 entries

Previous 1 2 Next

Figure 2 – Page des Best Known Solutions (BKS).

Reference: Pisinger and Ropke 2007  
Dataset: Christofides et al. 1979 - CMT

Instance	ALNS (best)
CMT01	524.61
CMT02	835.26
CMT03	826.14
CMT04	1029.56
CMT05	1297.12
CMT06	555.43
CMT07	909.68
CMT08	865.94
CMT09	1163.68
CMT10	1405.88
CMT11	1110.10

OK

Figure 3 – Fenêtre modale affichant les résultats.

## 4 Conception d'un script nettoyant le répertoire tmp

### 4.1 Analyse de la tâche

Le site VRP-REP a rencontré plusieurs crashes causés par un manque de mémoire du serveur et des fichiers temporaires d'utilisateurs voulant partager des données, quittant la procédure avant la fin du transfert. Ces fichiers restent donc stockés dans le répertoire tmp du serveur et prennent de la place inutilement. Ce répertoire se réinitialisant qu'au redémarrage, il faut donc faire un script Linux permettant de le nettoyer tous les jours.

### 4.2 Conception de la solution

Pour pouvoir réaliser un script s'exécutant tous les jours, il faut :

- Le fichier de script à exécuter ;
- Un script cron.

Le script cron est un fichier contenant le nom du script que nous voulons répéter dans le temps ainsi que la fréquence de répétition d'exécution du script. Le fichier de script à exécuter sera la commande : `rm -rf /tmp/*`.

## 5 Page de recherche des données du site

### 5.1 Analyse de la tâche

À la place de réparer la barre de recherche du site qui est actuellement hors d'usage comme vous pouvez le voir ci-dessous :

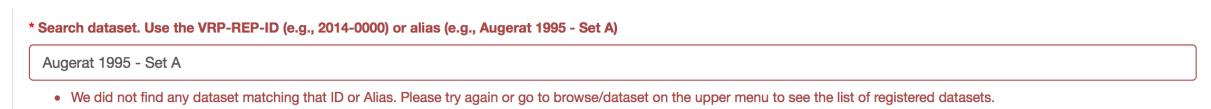


Figure 4 – Capture de la barre de recherche

Nous allons la remplacer par une page de recherche par mots-clés qui va afficher les datasets, références et variantes de problèmes du site. Différents moyens sont possibles pour réaliser cette tâche :

- Réutiliser les tableaux effectuant déjà un filtre ;
- Créer notre tableau et nos filtres.

Le premier choix m'a paru judicieux car cela nous évite de refaire un tableau et un filtre (le filtre du tableau étant déjà très efficace). Et également de garder la même forme de tableau qui va permettre aux utilisateurs du site de prendre la main plus facilement avec ce nouveau modulant en gardant la même forme de mise en page.

### 5.2 Conception de la solution

Pour concevoir cette solution, nous allons devoir réutiliser l'affichage de la barre de recherche et renvoyer vers la page de résultat lorsque nous effectuons une recherche à partir de la page d'accueil. Nous aurons besoin également de modifier le *routing* de ce bouton vers la page de recherche. Voici un mock de la page de recherche voulue :

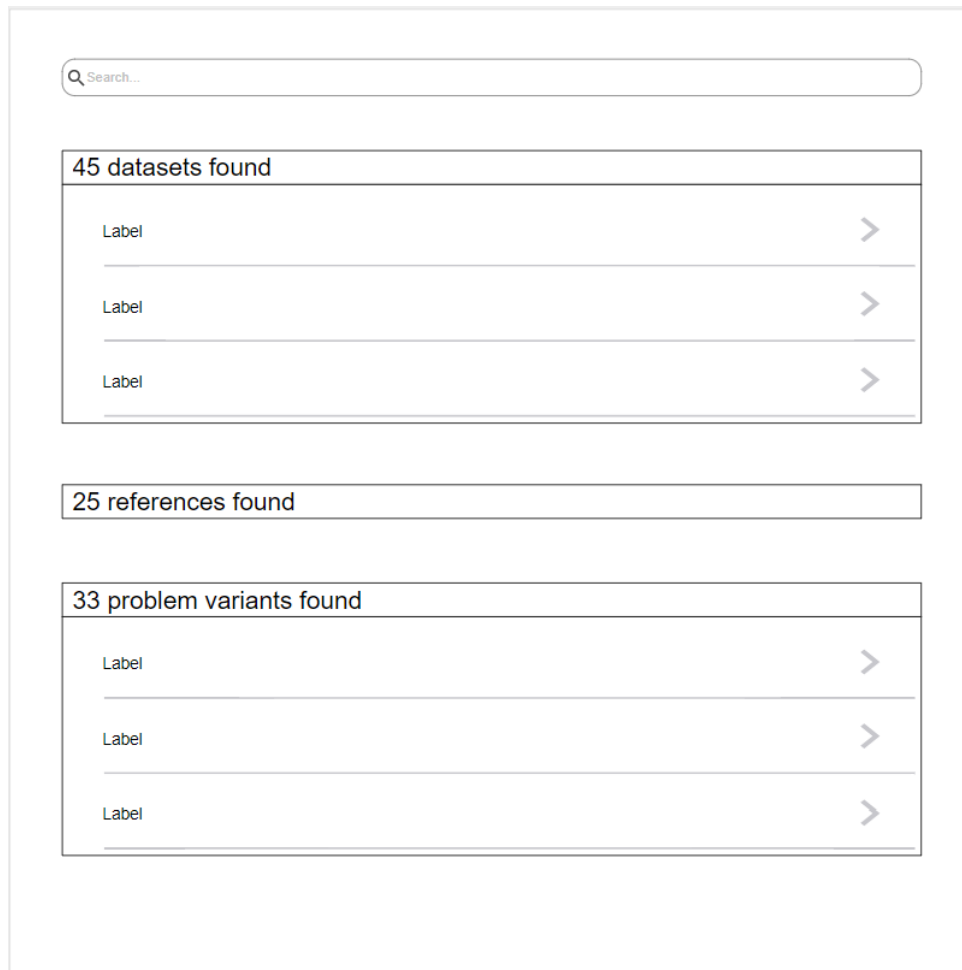


Figure 5 – Mock de la page de recherche.

Comme nous pouvons le voir dans la **Figure 5**, les tableaux sont dépliant pour pouvoir apporter plus de lisibilité. Lorsqu’une nouvelle recherche est effectuée, une nouvelle page est rechargée et tous les tableaux sont refermés (comme pour les références dans le mock). Les nombres affichés sont le nombre de résultat par catégorie.

## 6 Mise en évidence des mots-clés lors d’une recherche dans les data-table

### 6.1 Analyse de la tâche

Monsieur Mendoza souhaite mettre en évidence les mots-clés lors des recherches dans les tableaux.

### 6.2 Conception de la tâche

En consultant la documentation de datatable [20], j’ai trouvé deux librairies permettant de mettre en évidence les recherches :

- highlight.js : <https://datatables.net/blog/2014-10-22>;
- datatable.mark.js : <https://datatables.net/blog/2017-01-19>.

En me documentant un peu plus, il est conseillé d'utiliser la deuxième solution, car elle est encore mise à jour contrairement à la première.

## 7 Affichage de différentes statistiques du site

### 7.1 Analyse de la tâche

M. Mendoza souhaiterait afficher les statistiques suivantes :

- Nombre d'utilisateurs ;
- Nombre de pays différents des utilisateurs ;
- Nombre d'utilisateurs actifs ;
- Nombre de datasets ;
- Nombre de références ;
- Nombre de variantes de problèmes ;
- Nombre de contributeurs.

### 7.2 Conception de la solution

Voici le mock réalisé de cette page :

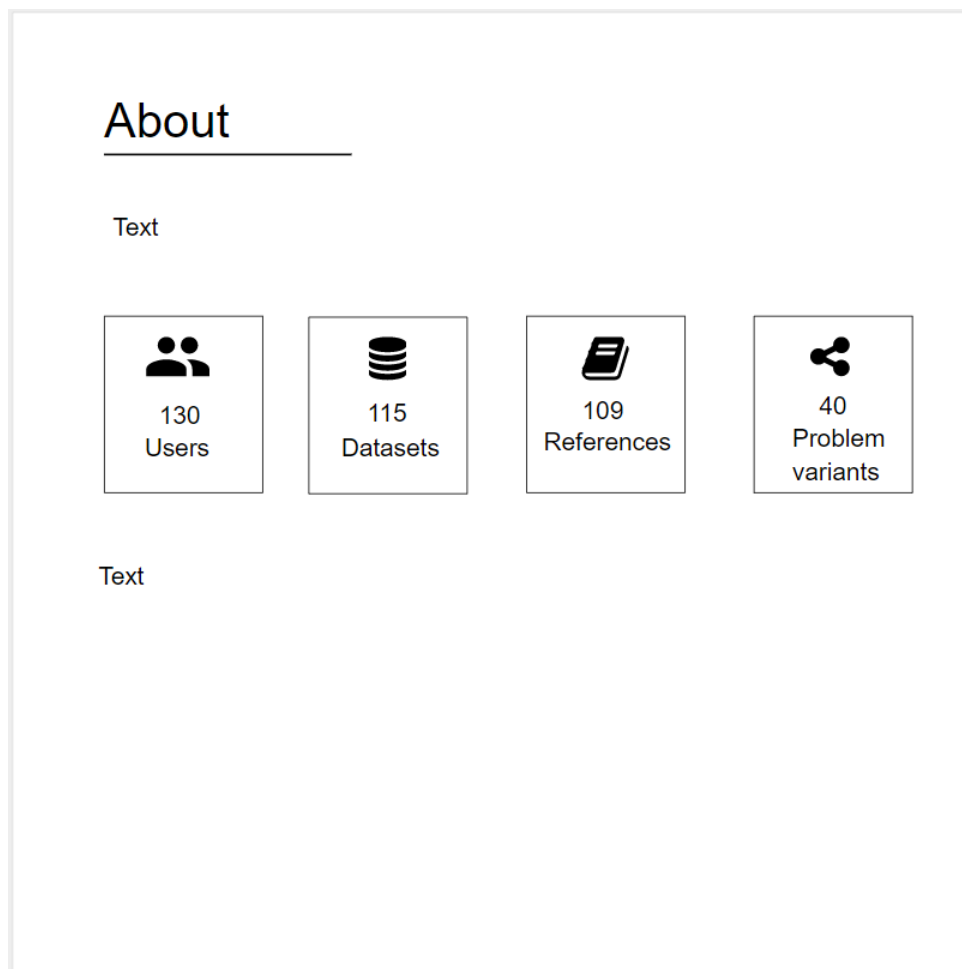


Figure 6 – Mock de la page de statistiques Statistics

## 8 Tri des données affichées de manière alphabétique des composants

### 8.1 Analyse de la tâche

Le menu dépliant des variants dans le menu « *contribute* » et le « *popover* » du tableau des Best Known Solution sont dans le désordre, il faut donc remédier à ce problème et les trier par ordre alphabétique. Voici l’affichage actuel de ces deux pages :

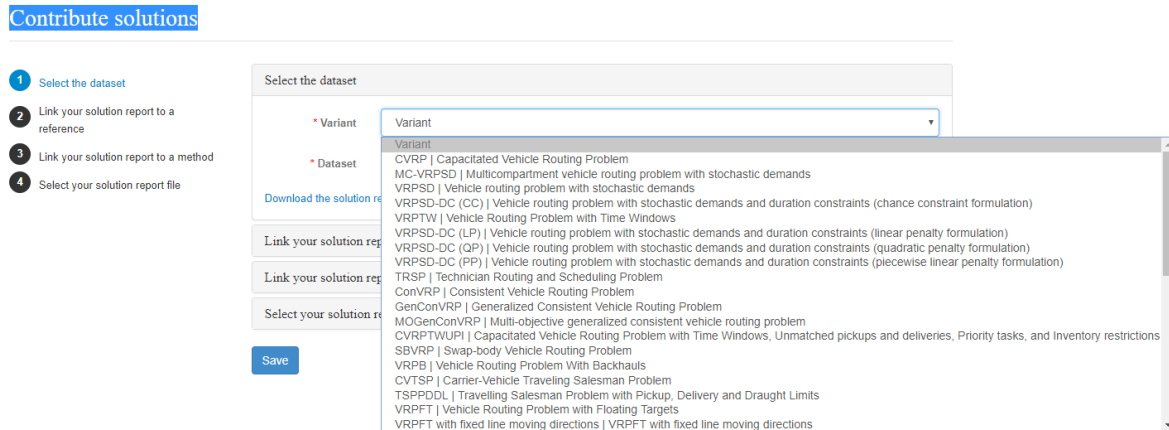


Figure 7 – Capture du problème de tri dans le menu contribute.

Variant

CVRP | Capacitated Vehicle Routing Problem

Dataset

Christofides et al. 1979 - CMT

Show10entries

Instance	O.F.	Optimality	References
CMT01	524.61	Proven	9 references
CMT02	835.26	Proven	7 references
CMT03	826.14	Proven	9 references
CMT04	1028.42	Proven	5 references
CMT05	1291.29	Proven	2 references

References

Pisinger and Ropke 2007

Prins 2009

Cordeau et al. 2001

Uchoa et al. 2017

Prins 2004

Vidal et al. 2012

Mester and Bräysy 2007

Nagata and Bräysy 2009

Rochat and Taillard 1995

Figure 8 – Capture du problème de tri dans le tableau des BKS.

### 8.2 Conception de la tâche

Nous allons dans un premier temps chercher pour quelles raisons les composants ne s’affichent plus dans l’ordre alphabétique. Pour cela, il va falloir trouver quelle requête a été envoyée à la base de données, lors du chargement de ces composants. Et voir s’il s’agit réellement d’un problème ou si les données n’ont tout simplement pas été triées.

## 9 Ajout d'un lien à partir des BKS vers leurs datasets respectifs

### 9.1 Analyse de la tâche

Dans cette partie, il est demandé de faire un lien à partir des solutions contenues dans le tableau des BKS vers leurs datasets respectifs.

### 9.2 Conception de la tâche

Pour réaliser cette tâche, nous allons récupérer l'événement généré lors d'un clic sur une ligne et rediriger vers le dataset qui lui est associé.

## 10 Ajout d'une fonctionnalité d'envoi de mail

### 10.1 Analyse de la tâche

Une fonctionnalité d'envoi de mail général à la communauté pour les administrateurs a été demandé.

### 10.2 Conception de la tâche

Il faut tout d'abord effectuer un formulaire permettant de réaliser le message, ensuite nous allons prendre les utilisateurs ayant accepté les notifications et leur envoyer un mail via un form handler.

# 5

## Mise en œuvre

### 1 Installation du projet

Pour pouvoir travailler sur le projet, il faut le récupérer et l'installer sur notre machine en local. Pour cela, il a fallu passer par différentes étapes d'installation que nous pouvons voir dans la partie [Section 3](#) (Annexe A) - [Contraintes de fonctionnement et d'exploitation](#). Certaines étapes ont donné lieu à des conflits que nous allons voir par la suite.

#### 1.1 Installation d'un environnement sous Linux

Pour reprendre ce projet, il est conseillé d'avoir un environnement sous Linux afin de pouvoir suivre les consignes d'installation déjà expliquées dans [\[2\]](#). Pour des soucis de performance, j'ai préféré réaliser un DualBoot (deux environnements sur la même machine: Windows et Linux), au lieu d'une VM ici faisant tourner les deux environnements en même temps. Les périphériques de mon ordinateur ayant des problèmes de compatibilités avec le noyau Debian, il est conseillé d'utiliser Fedora. J'ai donc choisi Fedora et suivi les consignes d'installation suivantes : [\[3\]](#).

#### 1.2 Installation d'un environnement de développement

Aucun environnement de développement n'est nécessaire pour lancer le projet. Ayant l'habitude de la suite *JetBrains*, j'ai décidé de choisir PHPStorm pour son interface intuitive. Pour voir les consignes d'installations : [\[9\]](#).

#### 1.3 Installation d'un serveur Apache httpd, Apache2, ...

Utilisant la distribution Linux Fedora, le serveur Apache est Apache httpd. pour l'installer, j'ai suivi les instructions suivantes (partie installation) : [\[11\]](#).



## 1.4 Installation d'une base de données MariaDB

Notre projet utilisant une base de données MySQL, nous installons MariaDB (remplaçant MySQL sous Fedora). Nous créons ensuite une base de données pour le projet à partir des commandes suivantes :

```
mysql -u root -p
CREATE DATABASE test_vrprep;
exit
```

## 1.5 Mise en place de la connexion SSH

Pour faire le lien entre le serveur et ma machine, nous passons par une connexion SSH. La connexion se fait via le shell de commande Linux ce qui nécessite l'installation du package SSH. Pour se connecter il faut taper la commande :

```
ssh user@ip
```

avec *user* : le nom de l'utilisateur et *ip* : ip du serveur.

Après l'exécution de cette commande, le mot de passe de l'utilisateur est demandé. Pour éviter cette étape nous pouvons réaliser la manipulation suivante [2] :

Pour plus de sécurité, on va créer un compte qui s'occupera de gérer l'application. Ici je l'ai appelé vrp

```
1 sudo adduser vrp
```

Puis nous allons créer deux paires de clés SSH (une pour root et une pour vrp) afin de faciliter la connexion en SSH.

```
1 ssh-keygen
```

Côté client, on peut ajouter notre clé publique sur le VPS.

```
1 ssh-copy-id -i ~/.ssh/id_rsa.pub titi@toto.host.org
```

**Figure 1** – Consigne de connexion ssh. [2]

Ainsi nous n'aurons plus à rentrer le mot de passe à chaque connexion.

## 1.6 Récupération du projet Git

Pour récupérer le projet sous Git, il faut tout d'abord installer le package Git (s'il n'est pas installé). Ensuite nous pouvons réaliser la commande suivante :

```
git clone vrp@51.254.132.44:var/repos/vrp-rep.git
```

## 1.7 Configuration des droits sur le projet

Le projet Symfony nécessitant certains droits utilisateurs, il nous faut réaliser différentes commandes afin de pouvoir le faire fonctionner. Cette étape a été assez longue, en effet nous

avons rencontré plusieurs problèmes de droit rajoutant d'autres instructions en plus de celles de Sensiolabs.

Les consignes seront disponibles dans le guide d'installation.

## 1.8 Récupération des backups du projet

Cette partie consiste à récupérer les données du site, pour pouvoir ensuite utiliser ces données lors de nos tests. Les consignes seront disponibles dans le guide d'installation.

## 1.9 Mise en place du Virtual Host

Le VirtualHost est un outil serveur permettant de créer un utilisateur virtuel permettant d'accueillir plusieurs noms de domaines. Il nous faut donc le configurer pour que Apache fasse le lien entre un nom de domaine et notre projet. Pour réaliser la configuration sur Fedora, allez voir documentation : [4]

Voici la configuration :

```
<VirtualHost *:80>
    ServerAdmin webmaster@vrp-anto.fr
    ServerName vrp.anto.fr
    DocumentRoot /home/antoine/PhpstormProjects/vrp-rep/web
    ServerAlias vrp.anto.fr
    <Directory /home/antoine/PhpstormProjects/vrp-rep/web>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Figure 2 – Fichier VirtualHost.

Lors de cette étape, nous avons rencontré différents soucis à cause de problèmes de droit sur le projet Symfony. Nous avons pu les résoudre à partir de la commande `chmod 755`, ainsi qu'une autre commande qui sera expliquée dans le guide d'installation.

## 2 Script de nettoyage du répertoire tmp du serveur

Le répertoire tmp surchargeant le serveur (qui a peu de capacité mémoire), nous allons effectuer une mesure palliative, visant à vider ce répertoire. Plus de détails sont disponibles dans la partie **Section 4** (Chapitre 4) - **Conception d'un script nettoyant le répertoire tmp**.

Nous avons donc créé un simple script sh réalisant la commande suivante permettant de vider le répertoire :

```
1 #!/bin/bash
2 DATE=$(date)
3 rm -r /tmp/*
4 echo "$DATE remove tmp success !"
5 exit 0
```

Le script est stocké dans le répertoire : /home/vrp/script/scriptDeleteTmp.sh.

Ensuite, nous avons réalisé le script CRON permettant d'exécuter le script sh à une fréquence voulue (pour nous quotidiennement) , il a la structure suivante [21]:

m H J M U C

**m** : les minutes;

**H** : l'heure;

**J** : le jour;

**M** : le mois;

**U** : l'utilisateur exécutant le script;

**C** : la commande ici notre script.

Voici un exemple : 5 3 \* \* \* root /home/script.sh

Ici le « script.sh » contenu dans le répertoire « home » est réalisé tous les jours à 03h05. Attention, une \* ne peut être mise que si rien n'est défini à droite. Exemple : si on veut réaliser un script se réalisant le premier du moins à 00h05.

**Il ne faut pas faire :**

5 \* 1 \* \* root /home/script.sh

**Il faut faire :**

5 0 1 \* \* root /home/script.sh

Voici notre script cron réalisé pour la plateforme (/etc/cron.d/launchScriptDeleteTmp):

```
1 00 08 * * * vrp /home/vrp/script/scriptDeleteTmp.sh >> /home/vrp/script/logDeleteTmp 2>&1
```

### 3 Correction des vendors et du déploiement en bêta

Nous avons fixé les vendors comme expliqué dans la [Section 1](#) (Chapitre 4) - [Correction des vendors](#). Nous avons ensuite essayé de le déployer sur la branche develop (avec nos modifications du module Captcha) pour l'envoyer sur la bêta du site (voir [Section 6](#) (Annexe A) - [Maintenance et évolution du système](#)) mais une erreur est survenue :

```
1 [antoine@localhost vrp-rep]$ git push origin develop
2 Decompte des objets: 8, fait.
3 Delta compression using up to 8 threads.
4 Compression des objets: 100% (8/8), fait.
5 Ecriture des objets: 100% (8/8), 618 bytes | 618.00 KiB/s, fait.
6 Total 8 (delta 7), reused 0 (delta 0)
7 remote: post-receive started
8 remote: From /home/vrp/var/repos/vrp-rep
9 remote: * branch          develop    -> FETCH_HEAD
10 remote:    8381a6a..a1e1a5e  develop    -> origin/develop
11 remote: error: Your local changes to the following files would be overwritten by merge:
12 remote:    app/config/config.yml
13 remote: Please, commit your changes or stash them before you can merge.
14 remote: Aborting
15 remote: Updating 9022057..a1e1a5e
16 remote: Warning: This development build of composer is over 60 days old. It is ↩
    recommended to update it by running "/usr/local/bin/composer self-update" to get ↩
    the latest version.
17 remote: Loading composer repositories with package information
18 remote: Installing dependencies from lock file
19 remote: Nothing to install or update
20 remote: Generating optimized autoload files
```

```

21 remote: Clearing the cache for the prod environment with debug false
22 remote:
23 remote:           Application Migrations
24 remote:
25 remote:
26 remote: Migrating up to 20160606115005 from 20141211142611
27 remote:
28 remote:   ++ migrating 20160120170836
29 remote:
30 remote:   -> ALTER TABLE dataset ADD vrp_rep_id VARCHAR(255) NOT NULL
31 remote: Migration 20160120170836 failed during Execution. Error An exception occurred ←
    while executing 'ALTER TABLE dataset ADD vrp_rep_id VARCHAR(255) NOT NULL':
32 remote:
33 remote: SQLSTATE[42S21]: Column already exists: 1060 Duplicate column name 'vrp_rep_id'
34 remote:
35 remote:
36 remote:
37 remote: [Doctrine\DBAL\DBALException]
38 remote: An exception occurred while executing 'ALTER TABLE dataset ADD vrp_rep_id ←
    VARCHAR(255) NOT NULL':
39 remote: SQLSTATE[42S21]: Column already exists: 1060 Duplicate column name 'vrp_rep_id'
40 remote:
41 remote:
42 remote:
43 remote:
44 remote:
45 remote:
46 remote: [PDOException]
47 remote: SQLSTATE[42S21]: Column already exists: 1060 Duplicate column name 'vrp_rep_id'
48 remote:
49 remote:
50 remote:
51 remote: doctrine:migrations:migrate [--write-sql] [--dry-run] [--configuration="..."] ←
    [--db-configuration="..."] [--em="..."] [version]
52 remote:
53 remote:
54 remote: Dumping all prod assets.
55 remote: Debug mode is off.
56 remote:
57 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/css/c94925f.css
58 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/fe5efde.js
59 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/55ca44b.js
60 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/58fb25d.js
61 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/2d36c32.js
62 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/dfc76e1.js
63 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/9018a87.js
64 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/618350c.js
65 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/a56e8cb.js
66 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/ad6e4e6.js
67 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/837cf51.js
68 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/960ea3e.js
69 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/da31384.js
70 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/101b9f3.js
71 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/f65c856.js
72 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/c634a24.js
73 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/90c02ae.js
74 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/css/4837d38.css
75 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/css/7c66d68.css
76 remote: 15:27:40 [file+] /home/vrp/var/repos/staging/app/./web/js/28910ec.js
77 remote: Installing assets as hard copies
78 remote: Installing assets for Symfony\Bundle\FrameworkBundle into ←

```

```

    /home/vrp/var/repos/staging/web/bundles/framework
79 remote: Installing assets for FOS\JsRoutingBundle into ↩
    /home/vrp/var/repos/staging/web/bundles/fosjsrouting
80 remote: Installing assets for Sonata\CoreBundle into ↩
    /home/vrp/var/repos/staging/web/bundles/sonatacore
81 remote: Installing assets for Sonata\FormatterBundle into ↩
    /home/vrp/var/repos/staging/web/bundles/sonataformatter
82 remote: Installing assets for Sonata\AdminBundle into ↩
    /home/vrp/var/repos/staging/web/bundles/sonataadmin
83 remote: Installing assets for Sonata\UserBundle into ↩
    /home/vrp/var/repos/staging/web/bundles/sonatauser
84 remote: Installing assets for Ivory\CKEditorBundle into ↩
    /home/vrp/var/repos/staging/web/bundles/ivoryckeditor
85 remote: Installing assets for Nelmio\ApiDocBundle into ↩
    /home/vrp/var/repos/staging/web/bundles/nelmioapidoc
86 remote: Installing assets for Bazinga\Bundle\JsTranslationBundle into ↩
    /home/vrp/var/repos/staging/web/bundles/bazingajstranslation
87 remote: Installing assets for Application\Sonata\UserBundle into ↩
    /home/vrp/var/repos/staging/web/bundles/applicationsonatauser
88 remote: Installing assets for Site\RepositoryBundle into ↩
    /home/vrp/var/repos/staging/web/bundles/siterepository
89 remote: Installing assets for Site\AdminBundle into ↩
    /home/vrp/var/repos/staging/web/bundles/siteadmin
90 remote: Installing assets for Site\TicketBundle into ↩
    /home/vrp/var/repos/staging/web/bundles/siteticket
91 remote: Installing assets for Hubtil\Bundle\FileSetBundle into ↩
    /home/vrp/var/repos/staging/web/bundles/hubtilfileset
92 remote: Installing assets for Sensio\Bundle\DistributionBundle into ↩
    /home/vrp/var/repos/staging/web/bundles/sensiodistribution
93 remote: post-receive finished
94 To 51.254.132.44:var/repos/vrp-rep.git
95 8381a6a..a1e1a5e develop -> develop
96
97 Ajouter un commentaireR

```

### 3.1 Correction des problèmes de déploiement sur la branche develop

#### 3.1.1 Correction du problème n°1 : fichier de config

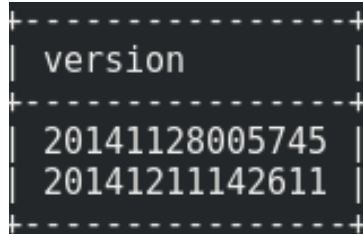
Premièrement, nous avons identifié une première erreur (ligne 11) au niveau du fichier *config.yml*. En effet le *.gitignore* empêchant la mise à jour de ce fichier, les ajouts dans ce fichier ne sont pas pris en compte. Nous avons donc supprimé la branche distante (remote) et réeffectué un push sur la branche develop, pour redéployer le projet avec le fichier de config modifié. Cette correction, n'ayant pas marché nous avons remarqué qu'il y avait toujours un souci avec ce fichier. En effectuant la commande *git status* et *git log*, nous avons vu que le fichier config avait son arborescence git cassée. Cela bloque toutes modifications futures de ce fichier. Ce problème a été résolu en récupérant le fichier de config de la branche master et remplaçant celui de la develop à la main.

#### 3.1.2 Correction du problème n°2 : migration

Ensuite, il nous restait le problème de migration de doctrine. En effet doctrine essaye d'effectuer une mise à jour de la base de données malgré le fait qu'elle le soit déjà. Lorsqu'un changement

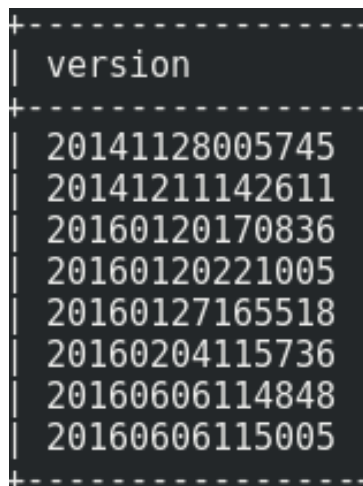
dans la base de données est effectué, Doctrine stocke dans la table **migration** de la base un *timestamp* qui permet d'identifier les mises à jour effectuées. Or nos mises à jour ont bien été effectuées sans l'indiquer à la table et cause l'erreur en essayant de relancer la migration déjà réalisée.

Voici les captures de la table migration de la branche develop et de la master :



version
20141128005745
20141211142611

Figure 3 – Capture de la table migration de la branche develop.



version
20141128005745
20141211142611
20160120170836
20160120221005
20160127165518
20160204115736
20160606114848
20160606115005

Figure 4 – Capture de la table migration de la branche master.

Nous avons donc renseigné manuellement les *timestamp* dans mysql. Suite à cela, le déploiement s'est effectué mais d'autres erreurs sont apparues comme vous pouvez le voir ci-dessous :

```

1 git push origin develop
2 Decompte des objets: 8, fait.
3 Delta compression using up to 8 threads.
4 Compression des objets: 100% (8/8), fait.
5 Ecriture des objets: 100% (8/8), 622 bytes | 622.00 KiB/s, fait.
6 Total 8 (delta 7), reused 0 (delta 0)
7 remote: post-receive started
8 remote: From /home/vrp/var/repos/vrp-rep
9 remote: * branch          develop    -> FETCH_HEAD
10 remote:   5d7a08a..dc990e1 develop  -> origin/develop
11 remote: Updating 9022057..dc990e1
12 remote: Fast-forward
13 remote:  .gitignore                |    2 +
14 remote:  app/AppKernel.php         |    2 +-
15 remote:  app/config/config.yml     |    4 +-
16 remote:  composer.json            |   82 +-
17 remote:  composer.lock            | 2092 ++++++++-----
18 remote:  .../UserBundle/Form/Type/RegistrationFormType.php |    3 +
19 remote:  .../views/Registration/register.html.twig         |    4 +
20 remote:  .../Resources/translations/messages.en.yml       |    2 +-
21 remote:  8 files changed, 1287 insertions(+), 904 deletions(-)
22 remote:  mode change 100644 => 100755 app/AppKernel.php

```

```

23 remote: mode change 100644 => 100755 app/config/config.yml
24 remote: mode change 100644 => 100755 composer.json
25 remote: Warning: This development build of composer is over 60 days old. It is ↩
    recommended to update it by running "/usr/local/bin/composer self-update" to get ↩
    the latest version.
26 remote: Loading composer repositories with package information
27 remote: Installing dependencies from lock file
28 remote: Your requirements could not be resolved to an installable set of packages.
29 remote:
30 remote:   Problem 1
31 remote:     - Installation request for doctrine/annotations v1.6.0 -> satisfiable by ↩
    doctrine/annotations[v1.6.0].
32 remote:     - doctrine/annotations v1.6.0 requires php ^7.1 -> your PHP version ↩
    (5.6.14-0+deb8u1) or "config.platform.php" value does not satisfy that requirement.
33 remote:   Problem 2
34 remote:     - Installation request for doctrine/cache v1.7.1 -> satisfiable by ↩
    doctrine/cache[v1.7.1].
35 remote:     - doctrine/cache v1.7.1 requires php ~7.1 -> your PHP version ↩
    (5.6.14-0+deb8u1) or "config.platform.php" value does not satisfy that requirement.
36 remote:   Problem 3
37 remote:     - Installation request for doctrine/collections v1.5.0 -> satisfiable by ↩
    doctrine/collections[v1.5.0].
38 remote:     - doctrine/collections v1.5.0 requires php ^7.1 -> your PHP version ↩
    (5.6.14-0+deb8u1) or "config.platform.php" value does not satisfy that requirement.
39 remote:   Problem 4
40 remote:     - Installation request for doctrine/common v2.8.1 -> satisfiable by ↩
    doctrine/common[v2.8.1].
41 remote:     - doctrine/common v2.8.1 requires php ~7.1 -> your PHP version ↩
    (5.6.14-0+deb8u1) or "config.platform.php" value does not satisfy that requirement.
42 remote:   Problem 5
43 remote:     - Installation request for doctrine/data-fixtures v1.3.0 -> satisfiable by ↩
    doctrine/data-fixtures[v1.3.0].
44 remote:     - doctrine/data-fixtures v1.3.0 requires php ^7.1 -> your PHP version ↩
    (5.6.14-0+deb8u1) or "config.platform.php" value does not satisfy that requirement.
45 remote:   Problem 6
46 remote:     - Installation request for gregwar/captcha v1.1.3 -> satisfiable by ↩
    gregwar/captcha[v1.1.3].
47 remote:     - gregwar/captcha v1.1.3 requires ext-gd * -> the requested PHP extension gd ↩
    is missing from your system.
48 remote:   Problem 7
49 remote:     - Installation request for guzzle/guzzle v3.9.3 -> satisfiable by ↩
    guzzle/guzzle[v3.9.3].
50 remote:     - guzzle/guzzle v3.9.3 requires ext-curl * -> the requested PHP extension ↩
    curl is missing from your system.
51 remote:   Problem 8
52 remote:     - Installation request for sonata-project/cache 1.1.1 -> satisfiable by ↩
    sonata-project/cache[1.1.1].
53 remote:     - sonata-project/cache 1.1.1 requires php ^7.1 -> your PHP version ↩
    (5.6.14-0+deb8u1) or "config.platform.php" value does not satisfy that requirement.
54 remote:   Problem 9
55 remote:     - doctrine/annotations v1.6.0 requires php ^7.1 -> your PHP version ↩
    (5.6.14-0+deb8u1) or "config.platform.php" value does not satisfy that requirement.
56 remote:     - willdurand/hateoas v2.6.0 requires doctrine/annotations ~1.0 -> satisfiable ↩
    by doctrine/annotations[v1.6.0].
57 remote:     - Installation request for willdurand/hateoas v2.6.0 -> satisfiable by ↩
    willdurand/hateoas[v2.6.0].
58 remote:
59 remote: PHP Fatal error: Class 'Gregwar\CaptchaBundle\GregwarCaptchaBundle' not found in ↩
    /home/vrp/var/repos/staging/app/AppKernel.php on line 56
60 remote: PHP Fatal error: Class 'Gregwar\CaptchaBundle\GregwarCaptchaBundle' not found in ↩
    /home/vrp/var/repos/staging/app/AppKernel.php on line 56

```

```

61 remote: PHP Fatal error: Class 'Gregwar\CaptchaBundle\GregwarCaptchaBundle' not found in ↩
    /home/vrp/var/repos/staging/app/AppKernel.php on line 56
62 remote: PHP Fatal error: Class 'Gregwar\CaptchaBundle\GregwarCaptchaBundle' not found in ↩
    /home/vrp/var/repos/staging/app/AppKernel.php on line 56
63 remote: post-receive finished
64 To 51.254.132.44:var/repos/vrp-rep.git
65 5d7a08a..dc990e1 develop -> develop

```

## 3.2 Correction des vendors

### 3.2.1 Correction du problème du Bundle gregwar/captcha

Comme nous pouvons le voir ligne 47, le module captcha requiert l'extension "ext-gd", il s'agit en fait d'une librairie PHP s'installant sur le serveur à partir de la commande suivante : `sudo ↩ apt-get install php5-gd` Une erreur est alors survenue :

```

1 Err http://ftp.debian.org/debian/ jessie/main libfontconfig1 amd64 2.11.0-6.3
2 404 Not Found [IP: 130.89.148.12 80]
3 Err http://security.debian.org/ jessie/updates/main libpng12-0 amd64 1.2.50-2+deb8u1
4 404 Not Found [IP: 212.211.132.250 80]
5 Err http://ftp.debian.org/debian/ jessie/main libtiff5 amd64 4.0.3-12.3
6 404 Not Found [IP: 130.89.148.12 80]
7 Err http://ftp.debian.org/debian/ jessie/main libx11-6 amd64 2:1.6.2-3
8 404 Not Found [IP: 130.89.148.12 80]
9 Err http://security.debian.org/ jessie/updates/main libfreetype6 amd64 2.5.2-3+deb8u1
10 404 Not Found [IP: 212.211.132.250 80]
11 Err http://ftp.debian.org/debian/ jessie/main libxpm4 amd64 1:3.5.11-1+b1
12 404 Not Found [IP: 130.89.148.12 80]
13 Err http://security.debian.org/ jessie/updates/main php5-gd amd64 5.6.14+dfsg-0+deb8u1
14 404 Not Found [IP: 212.211.132.250 80]
15 Err http://ftp.debian.org/debian/ jessie/main libgd3 amd64 2.1.0-5
16 404 Not Found [IP: 130.89.148.12 80]
17 E: Failed to fetch ↩
    http://security.debian.org/pool/updates/main/lib/libpng/libpng12-0_1.2.50-2+deb8u1_amd64.deb ↩
    404 Not Found [IP: 212.211.132.250 80]
18
19 E: Failed to fetch ↩
    http://security.debian.org/pool/updates/main/f/freetype/libfreetype6_2.5.2-3+deb8u1_amd64.deb ↩
    404 Not Found [IP: 212.211.132.250 80]
20
21 E: Failed to fetch ↩
    http://ftp.debian.org/debian/pool/main/f/fontconfig/libfontconfig1_2.11.0-6.3_amd64.deb ↩
    404 Not Found [IP: 130.89.148.12 80]
22
23 E: Failed to fetch ↩
    http://ftp.debian.org/debian/pool/main/t/tiff/libtiff5_4.0.3-12.3_amd64.deb 404 ↩
    Not Found [IP: 130.89.148.12 80]
24
25 E: Failed to fetch ↩
    http://ftp.debian.org/debian/pool/main/libx/libx11/libx11-6_1.6.2-3_amd64.deb ↩
    404 Not Found [IP: 130.89.148.12 80]
26
27 E: Failed to fetch ↩
    http://ftp.debian.org/debian/pool/main/libx/libxpm/libxpm4_3.5.11-1+b1_amd64.deb ↩
    404 Not Found [IP: 130.89.148.12 80]
28

```



```

29 E: Failed to fetch ↩
    http://ftp.debian.org/debian/pool/main/libg/libgd2/libgd3_2.1.0-5_amd64.deb 404 ↩
    Not Found [IP: 130.89.148.12 80]
30
31 E: Failed to fetch ↩
    http://security.debian.org/pool/updates/main/p/php5/php5-gd_5.6.14+dfsg-0+deb8u1_amd64.deb ↩
    404 Not Found [IP: 212.211.132.250 80]

```

Après avoir fait différentes recherches, le problème est lié au fait que le gestionnaire de paquets *apt-get* ne trouve pas les fichiers alors qu'ils existent. Il faut donc le mettre à jour avec la commande `sudo apt-get update`, relancer l'installation de `php5-gd` et redémarrer le serveur apache.

### 3.2.2 Correction du problème du Bundle guzzle/guzzle

La version v3.9.3 de `guzzle` requiert `ext-curl` de PHP, nous avons donc installé `php5-curl` avec la commande `sudo apt-get install php5-curl`.

### 3.2.3 Correction du problème des sous-dépendances des vendors

Les sous-dépendances sont les dépendances des vendors que nous utilisons. PHP 7 étant le prérequis de ces sous dépendances, je n'ai pas rencontré le problème sur ma machine. Pour palier ce problème, nous les avons donc fixés dans le fichier *composer.json* avec une version antérieure supportant php 5.

Voici la liste des sous-dépendances fixées à partir de Packagist : [10]:

- doctrine/annotations : v1.4.0;
- doctrine/cache : v1.6.2;
- doctrine/collection : v1.4.0;
- doctrine/common v2.7.1;
- doctrine/data-fixtures v1.0.2;
- sonata-project/cache : 1.0.7.

## 3.3 Mise en place de la solution Captcha

Pour la mise en place du module captcha nous avons utilisé le bundle *Gregwar/Captcha-Bundle* (pour plus de détails voir : [Section 2](#) (Chapitre 4) - [Problème d'inscription de bot sur le site](#)). Son intégration est assez simple comme stipulée dans <https://packagist.org/packages/gregwar/captcha-bundle> :

1. Exécuter la commande `composer require gregwar/captcha-bundle:v1.1`;
2. Ajouter `new Gregwar\CaptchaBundle\GregwarCaptchaBundle()` dans le fichier `app/appKernel.php`;
3. Ajouter `gregwar_captcha: ~` dans le fichier `app/config/config.yml`;
4. Et l'importer dans le *formbuilder* de cette façon (différente de la doc) « `$builder->add('captcha', ↩`  
`'captcha');` » en oubliant pas de l'importer : « `use Gregwar\CaptchaBundle\Type\CaptchaType;`  
`»;`
5. et ajouter le `{{form('captcha')}}` dans le fichier Twig voulu.

Voici le résultat :

- La page d'inscription.

## VRP-REP

[HOME](#) [STATISTICS](#) [FAQ](#) [FORUM](#) [RESOURCES](#) [BROWSE ▾](#)

## Registration

Login information

\* Email

\* Password

\* Confirm password

Personal information

\* First Name

\* Family Name

\* Country

Afghanistan ▾

\* Affiliation

\* Notifications

☒ On

☐ Off

Get notifications by email when new datasets or solutions are contributed to VRP-REP.

\* Captcha

 [Renew](#)

Register

Figure 5 – Capture de la page d'inscription.

— La page « contactez nous » .

# VRP-REP

[HOME](#) [STATISTICS](#) [FAQ](#) [FORUM](#) [RESOURCES](#) [BROWSE ▾](#)

## Contact

\* Service

Steering committee ▾

\* Email

\* Name

\* Subject

\* Body

\* Captcha


[Renew](#)

Send

Figure 6 – Capture de la page contactez nous.

## 4 Ajout d'une page de recherche par mot-clé

Dans cette partie, j'ai intégré une page de recherche par mot-clé des datasets, références et variantes de problèmes. Ce module a pour but de remplacer l'ancien module de recherche des datasets qui dysfonctionnaient comme expliqué dans la partie [Section 5](#) (Chapitre 4) - [Page de recherche des données du site](#).

Pour réaliser cette page, j'ai tout d'abord modifié le contrôleur de la page « Home » du bouton de recherche vers notre nouvelle page « Search ». Après avoir effectué le routing, j'ai dû implémenter la vue contenue dans le fichier Twig `search.twig`. Il a fallu rajouter le formulaire de la barre de recherche, intégrer les onglets et les tableaux de données « datatable » de nos données respectives (datasets, références, et variantes). De plus, il nous a fallu modifier les tableaux pour permettre de :

— Donner une recherche en entrée :

Pour cela, j'ai rencontré un faux problème que je trouve important de mettre en évidence. Les fichiers Javascript et CSS en Symfony ne sont pas chargés automatiquement. Symfony 2 utilise les fichiers Javascript et CSS qu'il a mis en cache au préalable dans le répertoire web. C'est donc à **nous** de lui faire charger ces modifications lorsque des ajouts et des modifications ont été faites sur ces fichiers ! Je vous conseille d'effectuer les commandes suivantes lors de la modification d'un de ces deux types de fichiers :

```

1 php app/console cache:clear
2 php app/console assets:install
3 php app/console assetic:dump

```

Ce problème m'ayant fait perdre pas mal de temps, j'utilisais au départ la fonction `fnSearch()` qui effectuait un tri après chargement au lieu du paramètre `OSearch` qui s'effectue dès le chargement du tableau.

— Récupérer le nombre de résultats et l'afficher dans le menu dépliant :

Je désirais obtenir le nombre de résultats que le tableau avait trouvé selon la recherche pour pouvoir l'afficher dans le menu dépliant. Ayant été aussi ralenti par le problème précédent, j'ai trouvé la solution suivante (à faire dans le fichier des tableaux des datasets, variantes, références) :

```
1 "fnServerData": function ( sSource, aoData, fnCallback ) {
2   aoData.push( { "name": "more_data", "value": "my_value" } );
3   $.getJSON( sSource, aoData, function ( json ) {
4     window.location.totalDat = json.iTotalDisplayRecords;
5     fnCallback( json );
6   } );
```

Elle me permet d'instancier une variable globale que je peux récupérer dans ma vue. Un autre souci est apparu, cette requête étant effectuée en asynchrone, la vue était affichée avant que la variable capte le résultat de la variable globale. J'ai donc fait la fonction suivante qui permet de récupérer en asynchrone la valeur puis de l'injecter dans la vue :

```
1 {#function allowing to get async global variable of displayed the number of problems ←
   variants #}
2 function checkVariant() {
3   if ( window.location.totalVar != undefined ) {
4     $('#variantText').prepend(window.location.totalVar + ' ');
5   } else {
6     window.setTimeout("checkVariant();", 100);
7   }
8 }
9
10 {#function allowing to get async global variable of displayed the number of references #}
11 function checkReference() {
12   if ( window.location.totalRef != undefined ) {
13     $('#referenceText').prepend(window.location.totalRef + ' ');
14   } else {
15     window.setTimeout("checkReference();", 100);
16   }
17 }
18
19 {#function allowing to get async global variable of displayed the number of datasets #}
20 function checkDataset() {
21   if ( window.location.totalDat != undefined ) {
22     $('#datasetText').prepend(window.location.totalDat + ' ');
23   } else {
24     window.setTimeout("checkDataset();", 100);
25   }
26 }
```

Voici le résultat :

Search

---

• Search problem variants, references, datasets :

Search

0 problem(s) variant(s) found

5 reference(s) found

Show 10 entries

Identifier	Title	Authors	Year
Mendoza et al. 2011	Constructive heuristics for the multicompartiment vehicle routing problem with stochastic demands	Mendoza, J. AND Castanier, B. AND Guéret, C. AND Medaglia, A. AND Velasco, N.	2011
Mendoza 2016	Test Reference by JEMG	Mendoza, J. E.	2016
Mendoza et al. 2010	A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands	Mendoza, J.E. AND Castanier, B. AND Guéret, C. AND Medaglia, A.L. AND Velasco, N.	2010
Mendoza et al. 2014	A hybrid metaheuristic for the vehicle routing problem with stochastic demand and duration constraints	Mendoza, J.E. AND Rousseau, L.-M. AND Villegas, J.G.	2014
Mendoza and Villegas 2013	A multi-space sampling heuristic for the vehicle routing problem with stochastic demands	Mendoza, J.E. AND Villegas, J.G.	2013

Showing 1 to 5 of 5 entries (filtered from 44 total entries)

Previous 1 Next

9 dataset(s) found

Figure 7 – Capture de la page de recherche.

## 5 Mise en évidence des mots-clés lors d'une recherche dans un data-table

Comme expliqué dans la partie [Section 6](#) (Chapitre 4) - **Mise en évidence des mots-clés lors d'une recherche dans les datatable**, le plugin choisi est `datatable.mark.js`. Nous l'avons installé de la manière suivante à l'aide de la documentation <https://github.com/julmot/datatables.mark.js/> :

1. Installation avec bower la commande « `bower install/datatables.mark.js --save` » ;
2. Installation de `mark.js` : « `bower install mark.js --save` » ;
3. Ajout du paramètre « `"mark" = true` » dans les fichiers js des datatable ;
4. Insertion dans le script.

À l'étape 4, nous avons rencontré un problème car comme stipulé dans la documentation symfony [19], il est nécessaire d'importer les bundles bower dans le dossier `web/assets/vendor/` du projet pour pouvoir les utiliser. Nous avons alors créé un fichier `.bowerrc` contenant ceci :

```

1 {
2   "directory": "web/assets/vendor/"
3 }
```

Ainsi, nous pouvons importer la librairie dans twig comme ceci :

```

1 {#import for mark the search#}
2 <script src="{ asset('assets/vendor/mark.js/dist/jquery.mark.js') }"></script>
3 <script src="{ asset('assets/vendor/datatables.mark.js/dist/datatables.mark.js') }"></script>

```

Voilà le résultat :

Show	10	entries	mendoza
Identifier	Title	Authors	Year
Mendoza et al. 2011	Constructive heuristics for the multicompartiment vehicle routing problem with stochastic demands	Mendoza, J. AND Castanier, B. AND Guéret, C. AND Medaglia, A. AND Velasco, N.	2011

Figure 8 – Capture du rendu de `datatable.mark.js`.

## 6 Ajout d'une page de statistique « Statistics »

Pour le menu Statistics nous avons décidé de faire des cartes avec un compteur faisant une animation comptant le nombre de :

- Utilisateurs;
- Pays des utilisateurs;
- Utilisateurs actifs;
- Datasets;
- Datasets téléchargés;
- Références;
- Variantes de Problèmes;
- Contributeurs.

Voici le résultat :

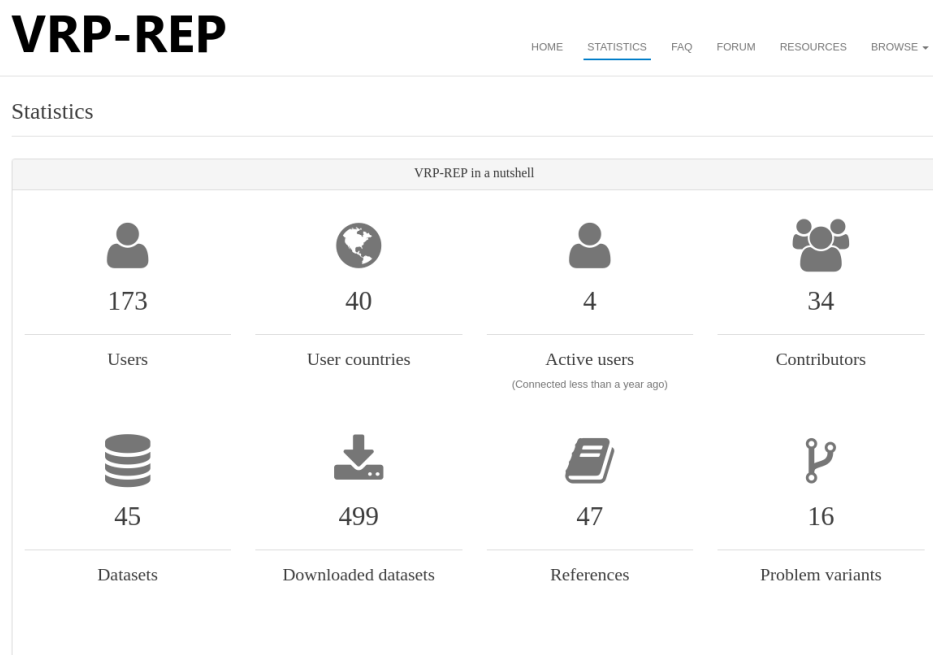


Figure 9 – Page statistique « Statistics ».

## 7 Correction des données ne s'affichant pas dans l'ordre alphabétique

Tout d'abord, nous avons dû trouver d'où venait le problème. Pour cela, il a fallu trouver la requête de la base de données envoyée pour récupérer les données. Après l'avoir trouvée, il a fallu ajouter la consigne "ORDER BY" :

- avec les *acronym* pour les variantes (menu déroulant);
- avec les *identifier* pour les références (popover).

## 8 Ajout d'un lien dans le tableau des BKS vers leur dataset respectif

Pour réaliser le lien, nous avons tout d'abord besoin d'injecter l'id des datasets à chaque solution. Pour cela, nous avons récupéré cette id dans la fonction *table* de la classe *SolutionRepository* et mis dans le tableau de manière invisible et non-recherchable. Après cela, il nous a fallu utiliser la méthode *rowCallback* qui permet de récupérer les événements des lignes comme les clics. Cet événement est ensuite filtré pour ne pas prendre en compte le clic sur les références qui affiche le popover.

Voici le code contenu dans *solution.js*:

```

1 {
2   "data": "dataset_id",
3   "visible": false,
4   "searchable": false
5 },
6
7 "rowCallback": function (row, data, displayIndex) {
8   $(row).click(function (e) {
9     if (e.target.cellIndex !== undefined){
10      window.location.href = Routing.generate(
11        'site_repository_dataset_itembyid',
12        {'id': data.dataset_id}
13      );
14    }
15  });
16 }
```

# 6

## Bilan

Voici le tableau de récapitulatif d'avancement des tâches :

Tâches	État
Installation du projet	Réalisée
Correction des vendors	Réalisée
Correctif du déploiement en beta (branche develop)	Réalisée
Création d'un script de nettoyage quotidien du dossier « tmp » du serveur	Réalisée
Mise en place d'une solution antibot Captcha	Réalisée
Page de recherche des datasets, références et variantes de problèmes	Réalisée
Mise en évidence des mots-clés lors d'une recherche dans un datatable	Réalisée
Correction des données ne s'affichant pas dans l'ordre alphabétique	Réalisée
Ajout d'une page de statistiques « Statistics » dans le menu	Réalisée
Ajout d'un lien dans le tableau des BKS vers leurs datasets respectifs	Réalisée
Création d'une fonctionnalité d'envoi de mail généralisé	En cours
Réalisation d'un XML supportant les solutions multicritères	Annulée
Migration des solutions d'une solution monocritère à multicritère	Annulée

Comme nous pouvons le voir ci-dessus certaines tâches ont été annulées et pour cause :

- Nous avons rencontré différents problèmes que j'ai du régler (non-mentionnés à la base) ;
- Des nouvelles fonctionnalités plus intéressantes demandées.

Malgré cela, nous avons réalisé le maximum de fonctionnalités qui ont pu être déployées et tous cela en respectant la qualité du code.





## Conclusion

En résumé, la partie de recherche m'a permis de mieux comprendre le projet dans son ensemble : son utilité, le domaine où il intervient, les changements à effectuer.

Grâce à cela, j'ai pu m'atteler à la partie développement. J'ai rencontré plusieurs problèmes avec le système actuel qu'il a fallu régler tout au long de ce projet et donc m'a ralenti dans l'exécution de mes tâches. Un nouveau diagramme de Gantt **Figure 2** (Annexe C) a été produit afin de mettre en évidence les changements par rapport au premier Gantt **Figure 1** (Annexe C)

## Annexes

# A

## Spécifications fonctionnelles et non fonctionnelles

### 1 Spécifications fonctionnelles

#### 1.1 Fixer les versions des vendors

##### 1.1.1 Rôles

Les vendors sont l'équivalent des librairies dans d'autres langages. Plus de détails sur les vendors et leur gestion sont disponibles dans la partie [Section 1.8](#) (Chapitre 3) - [Gestion des dépendances](#).

##### 1.1.2 Priorité

Cette tâche est primordiale, elle affecte la reprise du code, le projet en lui-même pouvant le rendre obsolète.

##### 1.1.3 Entrée et sortie

**En entrée**, nous avons le fichier *composer.json* qui contient la liste des dépendances utilisées dans le projet ainsi que leurs versions.

**En sortie**, nous avons le fichier *composer.json* avec les versions fixées par nos soins.

##### 1.1.4 Composantes

Les données utilisées sont celles récupérées à partir de la commande « *composer show* » exécutée sur le serveur en production, cette commande affiche les vendors du site. Bien sûr ils ne sont pas forcément tous bons ou complets, c'est pourquoi nous utilisons le site Packagist : [\[10\]](#).

### 1.1.5 Traitement

Ces vendors ayant bien évolué depuis le commencement du projet, il faut faire attention à ce qu'ils soient bien compatibles entre eux. Le problème est qu'ils ont été mal configurés dans le *composer.json* et autorisent des mises à jour créant ces problèmes d'incompatibilités. Il nous faut donc fixer les vendors sur la bonne version afin de ne plus avoir de soucis de compatibilité.

## 1.2 Mise en place d'une solution « Captcha » bloquant l'inscription des bots

### 1.2.1 Rôles

Le site VRP-REP rencontre un nombre conséquent de bots (robots) s'inscrivant sur le site mais ne validant pas l'étape de confirmation par mail. Pour éviter cela, nous voulons intégrer un bundle CAPTCHA empêchant ces robots de pouvoir accéder à l'étape de vérification des mails et donc empêcher leur enregistrement.

### 1.2.2 Priorité

Cette fonctionnalité est primordiale. Dans l'immédiat, elle n'empêche en rien l'utilisation de la plateforme mais cette mesure préventive est très utile pour éviter qu'un trop grand nombre de connexions se fassent, ainsi que la suppression manuelle de ces bots.

### 1.2.3 Entrée et sortie

**En entrée**, le code de l'image affichée ou bien une sélection d'image correspondant à la demande (ex : choisir les photos avec une voiture, ...).

**En sortie**, la validation par le bundle de la vérification CAPTCHA.

### 1.2.4 Composantes

Cette modification interagit avec la page d'inscription.

### 1.2.5 Traitement

Le Bundle intégrera dans la page d'inscription une vérification CAPTCHA ayant plusieurs variantes :

- Avec un code affiché sur une image, l'utilisateur devra lire l'image et écrire le code ;
- Avec une liste d'images où il faut sélectionner celles qui correspondent aux thèmes (ex : voiture, vélo, ...).

Après validation, il aura accès à la suite de la procédure d'inscription.

### 1.3 Ajout d'un menu « Statistics »

#### 1.3.1 Rôles

Cette fonction donnera lieu à la création d'une page accessible par le menu affichant les données statistiques du site.

#### 1.3.2 Priorité

Cette fonction est secondaire. Dans la mesure où il n'influe pas sur le bon fonctionnement du site mais est utile pour étudier son développement.

#### 1.3.3 Entrée et sortie

**En entrée**, les différentes données récupérables à partir du site ;

**En sortie**, les données transformées pour les afficher en différents graphes, liste de données ...

#### 1.3.4 Composantes

Les composantes intervenant dans cette fonctionnalité sont les données du site.

#### 1.3.5 Traitement

Nous devons sélectionner les données que nous allons extraire et appliquer des transformations choisies pour ensuite afficher les données.

### 1.4 Tri des données affichées dans l'ordre alphabétique

#### 1.4.1 Rôles

Permettre un affichage plus clair des listes de données dans le menu contribute -> solution, ainsi qu'au survol des références de la table d'affichage des instances ;

#### 1.4.2 Priorité

Cette fonction est secondaire. Dans la mesure où il n'influe pas sur le bon fonctionnement du site mais est utile pour étudier son développement.

#### 1.4.3 Entrée et sortie

**En entrée**, les différentes données récupérables à partir du site ;

**En sortie**, les données transformées pour les afficher en différents graphes, liste de données ...

#### 1.4.4 Composantes

Les composantes intervenant dans cette fonctionnalité sont les données du site.

#### 1.4.5 Traitement

Nous devons sélectionner les données que nous allons extraire et appliquer des transformations choisies pour ensuite afficher les données.

### 1.5 Passage d'une solution monocritère à multicritère

#### 1.5.1 Rôles

Cette modification aura pour but de passer d'une solution monocritère à multicritère gérant les solutions pour permettre l'ajout de différents critères à améliorer.

#### 1.5.2 Priorité

Cette fonction est primaire. Elle va permettre d'afficher les résultats des solutions d'une nouvelle famille de problème sous le format XML du site.

#### 1.5.3 Entrée et sortie

**En entrée**, le fichier de solution du problème permettant de prendre en compte plusieurs critères.

**En sortie**, l'affichage de ces données dans les solution du site.

#### 1.5.4 Composantes

Cette importante mise à jour interviendra sur plusieurs endroits du site :

- le fichier XML standard d'ajout d'une solution ;
- la table « Solution » de la base de données ;
- la page d'affichage des solutions.

#### 1.5.5 Traitement

Nous allons revoir la structure du fichier XML et de la base de données pour qu'ils puissent inclure plusieurs critères de minimisation ou de maximisation.

## 2 Spécifications non fonctionnelles

### 2.1 Contraintes de développement et conception

Pour pouvoir se lancer dans le projet, il faut mettre en place les différents outils et environnement nécessaires.

### 2.1.1 Matériel

- Un ordinateur pour effectuer la partie développement ;
- Un serveur pour déployer et tester.

### 2.1.2 Langages

- PHP sous le Framework Symfony 2 ;
- TWIG couplé avec le HTML pour les vues ;
- CSS pour le style des vues ;
- Javascript pour les fonctionnalités dynamiques ;
- SQL pour la base de données.

### 2.1.3 Bibliothèques

Les différentes bibliothèques que nous allons utiliser pour développer nos différentes fonctionnalités se nomment « Bundle ».

### 2.1.4 Environnement

- Linux ;
- Git ;
- Serveur Apache ;
- PHPStorm.

### 2.1.5 Protocole de communication

- SSH pour la communication avec le serveur ;
- HTTP pour la communication en tant qu'utilisateur.

## 3 Contraintes de fonctionnement et d'exploitation

Pour la mise en place du projet existant, il est nécessaire de prendre ces dispositions afin de pouvoir réaliser le projet :

- Installation d'un environnement sous Linux (conseillé) ;
- Installation d'un environnement de développement PHP Storm (conseillé) ;
- Installation d'un serveur Apache httpd, Apache2, etc ;
- Installation d'une base de données MariaDB ;
- Mise en place de la connexion SSH entre ma machine et le serveur ;
- Récupération du projet Git ;
- Configuration des droits sur le projet ;
- Récupération des backup du projet ;
- Mise en place d'un Virtual Host.

## 4 Performances

Comme tout site web, pour favoriser l'utilisation de cette plateforme, la minimisation du temps de réponse entre le serveur et l'utilisateur est souhaitée.

## 5 Sécurité

Actuellement, le site effectue une vérification par email lors de l'inscription. Au cours de ce projet, nous allons rajouter une vérification CAPTCHA durant la phase d'inscription. Seuls les utilisateurs connectés peuvent accéder aux informations suivantes, des utilisateurs ayant contribué sur la plateforme :

- Nom ;
- Pays ;
- Affiliation (ex : Polytech Tours) ;
- Toutes ses contributions.

Les informations personnelles de tous les utilisateurs ne sont accessibles que par l'administrateur.

## 6 Maintenance et évolution du système

Un document d'installation doit être effectué en cas de reprise du projet, dedans il devra être spécifié toute la procédure de mise en place du projet pour le développement. Une correction des vendors doit être effectuée pour faciliter l'installation. Toutes les modifications apportées devront être documentées dans le rapport final.

Le déploiement du travail effectué se fait en deux temps à partir des branches Git : **develop** et **master**. Lorsqu'une modification est envoyée sur une des deux branches Git (hébergée par le serveur), un **hook** est activé, il effectue le script contenu dans le fichier **post-receive** :

- Lors d'un *push* sur la branche develop, le hook déploie sur la beta <http://beta-version.vrp-rep.org/> ;
- Lors d'un *push* sur la branche master, le hook déploie sur le site <http://www.vrp-rep.org/>.



Voici le script du fichier Post-receive :

```
#!/bin/bash
echo "post-receive started"

while read oldrev newrev refname
do
    branch=$(git rev-parse --symbolic --abbrev-ref $refname)
    if [ "develop" == "$branch" ]; then
        export GIT_WORK_TREE="../staging"
        export GIT_DIR="../staging/.git"
        cd $GIT_WORK_TREE
        git pull origin develop
        composer install --no-dev --optimize-autoloader --no-scripts
        bower install
        php app/console cache:clear --env=prod --no-debug
        php app/console --no-interaction doctrine:migrations:migrate --env=prod
        php app/console assetic:dump --env=prod
        php app/console assets:install /home/vrp/var/repos/staging/web
    fi
    if [ "master" == "$branch" ]; then
        export GIT_WORK_TREE="../live"
        export GIT_DIR="../live/.git"
        cd $GIT_WORK_TREE
        git fetch --tags
        latestTag=$(git describe --tags `git rev-list --tags --max-count=1`)
        git checkout -f $latestTag
        composer install --no-dev --optimize-autoloader --no-scripts
        bower install
        php app/console cache:clear --env=prod --no-debug
        php app/console --no-interaction doctrine:migrations:migrate --env=prod
        php app/console assetic:dump --env=prod
        php app/console assets:install /home/vrp/var/repos/live/web
    fi
done
echo "post-receive finished"
```

Figure 1 – Script du fichier post-receive.

# B

## Cahier de développement

### 1 Tâche 1 : Étude des problèmes de tournées de véhicule

#### 1.1 Description de la tâche

Pour mieux comprendre le service auquel la plateforme VRP-REP contribue, il faut mieux comprendre les problèmes VRP. Pour cela, j'ai lu l'œuvre [3]. Cette tâche contribue à une partie importante de l'état de l'art.

#### 1.2 Livrables

Le livrable de l'état de l'art est à rendre le 07/12/2017.

#### 1.3 Estimation de charge

Cette tâche est estimée à 1 jour/homme.

#### 1.4 Temps et date de réalisation

Cette tâche a été réalisée en 1 jour/homme et terminée le 21/09/2017.

### 2 Tâche 2 : Formation Symfony

#### 2.1 Description de la tâche

N'ayant jamais utilisé Symfony 2, une formation était nécessaire afin de mieux comprendre le fonctionnement du projet. Pour cela j'ai suivi les différents tutoriels vidéo expliquant les mécanismes et l'architecture du Framework. Cette tâche est importante pour compléter le cahier de spécification, l'état de l'art ainsi que pour la partie développement.

## 2.2 Estimation de charge

Cette tâche est estimée à 3 jour/homme.

## 2.3 Temps et date de réalisation

Cette tâche a été réalisée en 3 jour/homme et terminée le 28/09/2017.

# 3 Tâche 3 : Installation du projet

## 3.1 Description de la tâche

Cette tâche correspond à l'installation du projet et de ses prérequis. Les étapes d'installation sont listées dans la partie **Section 2.1** (Annexe A) - **Contraintes de développement et conception**.

## 3.2 Estimation de charge

Cette tâche est estimée à 8 jour/homme.

## 3.3 Temps et date de réalisation

Cette tâche a été réalisée en 8 jour/homme et terminée le 16/11/2017.

# 4 Tâche 4 : Rédaction du cahier de spécification

## 4.1 Description de la tâche

La rédaction du cahier de spécifications permet de définir le contexte, les objectifs, le planning, les spécifications fonctionnelles et non fonctionnelles.

## 4.2 Livrables

Le cahier de spécifications est à rendre le 07/12/2017.

## 4.3 Estimation de charge

Cette tâche est estimée à 6 jour/homme.

## 4.4 Temps et date de réalisation

Cette tâche a été réalisée en 6 jour/homme et terminée le 07/12/2017.

## 5 Tâche 5 : Rédaction de l'état de l'art / Veille technologique

### 5.1 Description de la tâche

La rédaction de l'état de l'art consiste à résumer les enjeux des problèmes VRP. Dans la veille technologique, nous faisons l'analyse des outils utilisés pour la plateforme, ou des outils permettant de mettre à bien la tâche.

### 5.2 Livrables

L'état de l'art et la veille technologique sont à rendre le 07/12/2017.

### 5.3 Estimation de charge

Cette tâche est estimée à 6 jour/homme.

### 5.4 Temps et date de réalisation

Cette tâche a été réalisée en 6 jour/homme et terminée le 07/12/2017.

## 6 Tâche 6 : Correction des vendors

### 6.1 Description de la tâche

Cette tâche a pour but de stabiliser les versions des vendors contenus dans le fichier *composer.json*. Le but de cette manipulation est expliqué dans [Section 1](#) (Chapitre 4) - **Correction des vendors**.

### 6.2 Estimation de charge

Cette tâche est estimée à 2 jour/homme.

### 6.3 Temps et date de réalisation

Cette tâche a été réalisée en 2 jour/homme et terminée le 09/11/2017.

## 7 Tâche 7 : Test des vendors

### 7.1 Description de la tâche

Cette tâche a pour but de vérifier si tous les vendors corrigés nous permettent d'obtenir une version stabilisée du site. En cas de conflit, on modifiera la version du bundle jusqu'à stabilité du site.

## 7.2 Estimation de charge

Cette tâche est estimée à 2 jour/homme.

## 7.3 Temps et date de réalisation

Cette tâche a été réalisée en 2 jour/homme et terminée le 14/12/2017.

# 8 Tâche 8 : Correction des erreurs Symfony

## 8.1 Description de la tâche

Des erreurs empêchant le lancement du site de se faire ont été constaté, il est sujet de les corriger afin de passer à la suite. Ces erreurs sont potentiellement en lien avec les vendors.

## 8.2 Estimation de charge

Cette tâche est estimée à 4 jour/homme.

## 8.3 Temps et date de réalisation

Cette tâche a été réalisée en 4 jour/homme et terminée le 11/01/2018.

# 9 Tâche 9 : Correction de la barre de recherche des Best Known Solution

## 9.1 Description de la tâche

La barre de recherche dans menu « home » ne recherche pas les instances, il faut donc la rendre opérationnelle.

## 9.2 Estimation de charge

Cette tâche est estimée à 2 jour/homme.

## 9.3 Temps et date de réalisation

Cette tâche a été réalisée en 1 jour/homme et terminée le 14/03/2018.

## 10 Tâche 10 : Tri des données affichées de manière alphabétique des solutions (menu « contribute »), des références de la table d'affichage des instances

### 10.1 Description de la tâche

Différents composants du site affichent les données dans le désordre. Il faut les afficher de manière alphabétique.

### 10.2 Estimation de charge

Cette tâche est estimée à 2 jour/homme.

### 10.3 Temps et date de réalisation

Cette tâche a été réalisée en 1 jour/homme et terminée le 08/03/2018.

## 11 Tâche 11 : Ajout d'un Bundle CAPTCHA

### 11.1 Description de la tâche

Cette tâche correspond à l'installation d'un bundle captcha dans l'étape d'inscription de la plateforme.

### 11.2 Estimation de charge

Cette tâche est estimée à 4 jour/homme.

### 11.3 Temps et date de réalisation

Cette tâche a été réalisée en 2 jour/homme et terminée le 17/01/2018.

## 12 Tâche 12 : Passage à des solutions multicritère de minimisation ou maximisation

### 12.1 Description de la tâche

Cette étape consiste à permettre l'intégration de différents critères intervenant sur la solution. Cette modification implique des modifications dans :

- XML ;
- Base de données des solutions ;
- Affichage des solutions.

**12.2 Estimation de charge**

Cette tâche est estimée à 5 jour/homme.

**12.3 Temps et date de réalisation**

Cette tâche a été abandonnée au profit d'autres tâches.

**13 Tâche 13 : Ajout d'un menu statistique « Statistics »****13.1 Description de la tâche**

Un onglet « Statistics » doit être intégré, il redigera vers une page affichant les statistiques du site.

**13.2 Estimation de charge**

Cette tâche est estimée à 3 jour/homme.

**13.3 Temps et date de réalisation**

Cette tâche a été réalisée en 2 jour/homme et terminée le 07/03/2018.

**14 Tâche 14 : Rédaction d'un document d'installation****14.1 Description de la tâche**

La rédaction de ce document a pour but de simplifier la reprise du code. Le document d'installation contiendra les consignes d'installation du projet ainsi que les identifiants du serveur.

**14.2 Estimation de charge**

Cette tâche est estimée à 1 jour/homme.

**14.3 Temps et date de réalisation**

Cette tâche a été réalisée en 1 jour/homme et terminée le 21/03/2018.

## 15 Tâche 15 : Rédaction du rapport final

### 15.1 Description de la tâche

La rédaction du rapport final consistera à l'explication globale du PRD.

### 15.2 Livrables

Le rapport final est à rendre début avril.

### 15.3 Estimation de charge

Cette tâche est estimée à 5 jour/homme.

### 15.4 Temps et date de réalisation

Cette tâche a été réalisée en 5 jour/homme et terminée le 04/04/2018.

## 16 Tâche 16 : Script de nettoyage du répertoire tmp

### 16.1 Description de la tâche

Réaliser un script nettoyant quotidiennement le répertoire tmp du serveur.

### 16.2 Estimation de charge

Cette tâche est estimée à 1 jour/homme.

### 16.3 Temps et date de réalisation

Cette tâche a été réalisée en 1 jour/homme et terminée le 17/01/2018.

## 17 Tâche 17 : Module de recherche page search

### 17.1 Description de la tâche

Réaliser une page permettant de rechercher avec un mot-clé : datasets, références, variantes de problèmes.



**17.2 Estimation de charge**

Cette tâche est estimée à 4 jour/homme.

**17.3 Temps et date de réalisation**

Cette tâche a été réalisée en 8 jour/homme et terminée le 14/02/2018.

**18 Tâche 18 : Mise en évidence des mots-clés lors d'une recherche dans un datatable****18.1 Description de la tâche**

Trouver une solution pour mettre en évidence les mots-clés, lors des recherches dans les data-tables.

**18.2 Estimation de charge**

Cette tâche est estimée à 2 jour/homme.

**18.3 Temps et date de réalisation**

Cette tâche a été réalisée en 2 jour/homme et terminée le 21/02/2018.

**19 Tâche 19 : Correction des problèmes de déploiement sur la branche develop****19.1 Description de la tâche**

Corriger les problèmes de déploiement sur le site en bêta.

**19.2 Estimation de charge**

Cette tâche est estimée à 2 jour/homme.

**19.3 Temps et date de réalisation**

Cette tâche a été réalisée en 2 jour/homme et terminée le 28/02/2018.

## 20 Tâche 20 : Ajout d'un lien à partir des BKS vers leur dataset respectif

### 20.1 Description de la tâche

Ajouter un lien cliquable dans le tableau des BKS redirigeant vers la page d'information de leur dataset respectif.

### 20.2 Estimation de charge

Cette tâche est estimée à 1 jour/homme.

### 20.3 Temps et date de réalisation

Cette tâche a été réalisée en 1 jour/homme et terminée le 15/03/2018.

## 21 Tâche 21 : Ajout d'une fonctionnalité d'envoi de mail

### 21.1 Description de la tâche

Ajouter une fonctionnalité permettant l'envoi de mails des administrateurs vers les utilisateurs du site.

### 21.2 Estimation de charge

Cette tâche est estimée à 2 jour/homme.

### 21.3 Temps et date de réalisation

Cette tâche a été réalisée en 2 jour/homme et terminée le 29/03/2018.

# C

## Planning prévisionnels

### 1 Méthodologie

Pour plus d'informations sur la méthodologie allez dans la partie : [Base méthodologiques](#).

### 2 Diagramme de Gantt

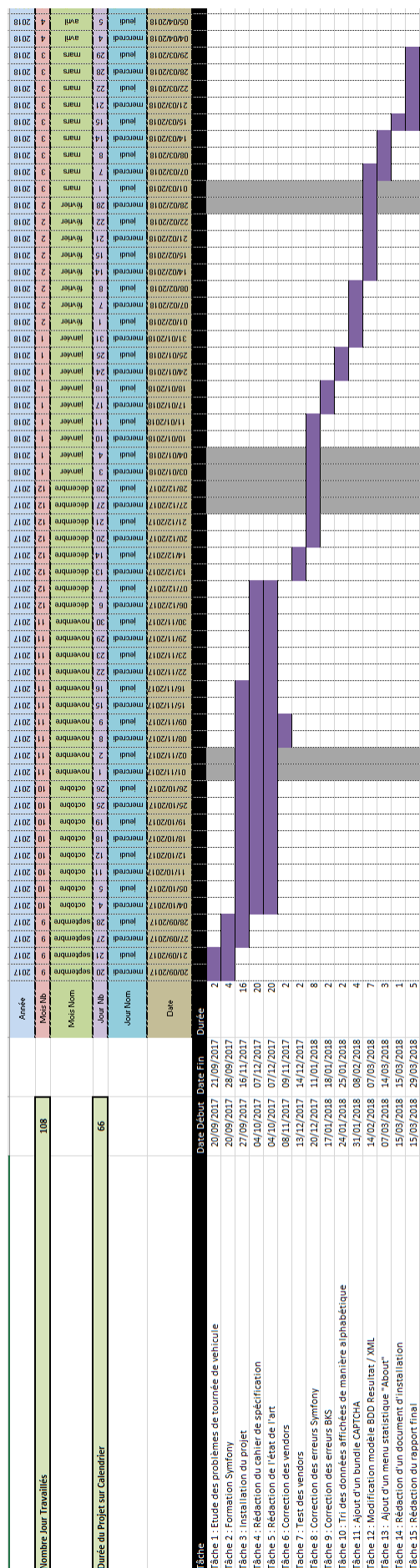


Figure 1 – Diagramme de Gantt

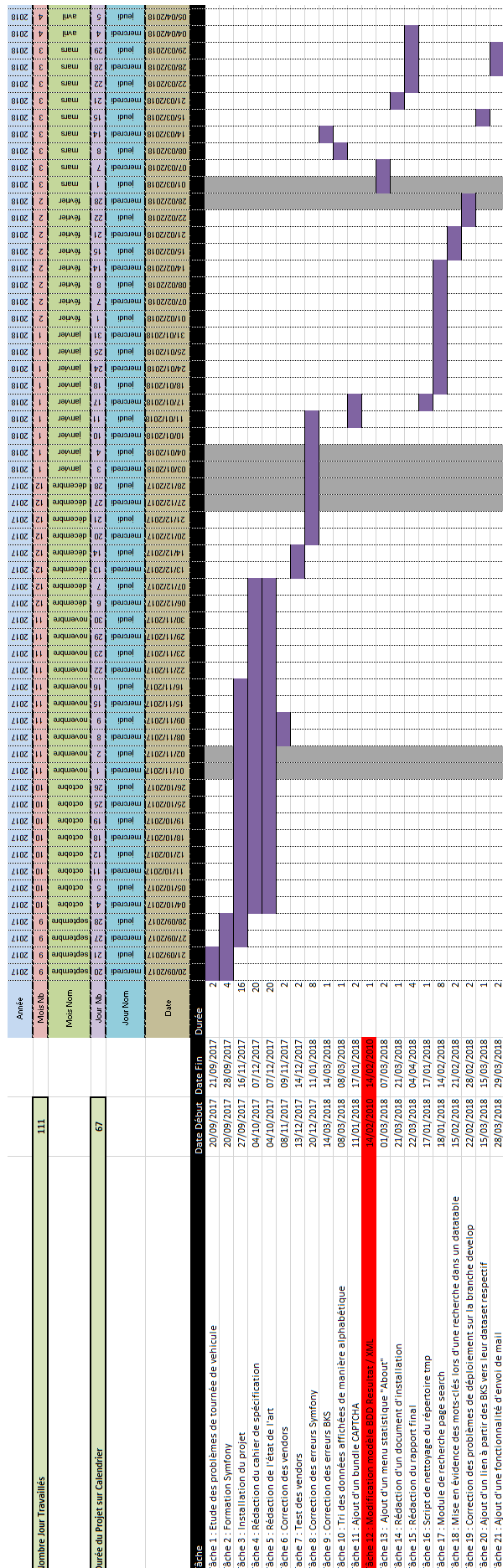


Figure 2 – Diagramme de Gantt final.

# D

# Cahier de développeur

## 1 Langage, librairies et convention de nommage

Le langage utilisé pour réaliser ce Projet de Recherche et Développement est le PHP. Le projet fonctionne sous le framework Symfony 2, qui va encadrer notre structure de projet et nous apporter ses mécanismes. Nous utiliserons l'IDE PhpStorm (gratuit pour les étudiants) qui permet de faciliter le développement grâce à son interface intuitive et ses multiples fonctionnalités.

Pour les librairies, nous utiliserons les systèmes de gestion de dépendances suivantes :

- Composer [1] : il enregistre toutes les dépendances PHP dans le fichier *composer.json*, toutes les librairies sont disponibles sur le site [10];
- NPM : permet d'installer Grunt, Bower, il enregistre toutes les dépendances Javascript dans le *package.json* [7];
- Bower : permet d'installer des librairies Javascript en les enregistrant dans le fichier *bower.json*, nous l'utiliserons au détriment de NPM [2].

Les librairies devront être populaires et être toujours maintenues.

Les conventions respecteront les conventions Symfony 2, elles sont disponibles ici : [18].

## 2 Gestionnaire de version

Le gestionnaire de version utilisé est GIT hébergé sur le serveur VRP-REP, voir le guide d'installation [Annexe E - Documentation d'installation du projet](#).

## 3 Conseils de développement

Un environnement Unix est recommandé, les commandes sont plus simples à réaliser et la documentation d'installation est disponible ici : [Annexe E - Documentation d'installation du projet](#).

Pour le développement, je conseillerais l'IDE PhpStorm pour son interface et ses fonctionnalités qui permettent de développer plus facilement, il est disponible gratuitement lorsque l'on est étudiant : <https://www.jetbrains.com/phpstorm/>.

Il est également fortement conseillé d'utiliser l'architecture GitFlow pour bien respecter l'étape de déploiement en bêta puis en prod, voici une documentation expliquant le fonctionnement : <http://danielkummer.github.io/git-flow-cheatsheet/>

## 4 Commentaires

Pour simplifier la reprise, des commentaires doivent être ajoutés pour expliquer les fonctions ou le code : PHP, Javascript, HTML.

## 5 Structure du système

Le fonctionnement du système est expliqué dans la partie **Section 2** (Chapitre 1) - **Environnement du projet**. La structure du système est modélisée dans le schéma suivant : **Figure 3** (Chapitre 1).

## 6 Fonctionnement de Symfony2

Le fonctionnement de Symfony 2 est expliqué dans la partie **Section 1** (Chapitre 3) - **Symfony2**

## 7 Installation

Pour l'installation du projet, se référer à la partie **Annexe E - Documentation d'installation du projet**.

# E

# Documentation d'installation du projet

## 1 Avant de commencer

**Important :** L'IP du serveur est 51.254.132.44, il vous faudra également demander les mots de passe du serveur, et de la base de données à Monsieur Mendoza.

## 2 Installation

Ce guide d'installation est valable sous distribution Linux, il est donc conseillé d'en installer une pour travailler sur ce projet.

### 2.1 Mise en place d'une connexion au VPS

Pour plus de sécurité, on va créer un utilisateur « vrp » qui s'occupera de gérer l'application :

```
1 sudo adduser vrp
```

Nous allons créer deux paires de clés SSH (une pour root et une pour vrp) afin de faciliter la connexion en SSH :

```
1 ssh-keygen
```

Côté client, on peut ajouter notre clé publique sur le VPS (faire la même manipulation pour l'utilisateur root) :

```
1 ssh-copy-id -i ~/.ssh/id_rsa.pub vrp@51.254.132.44
```

### 2.2 Installation de l'environnement

Tout d'abord nous allons mettre à jour le système pour bénéficier des derniers paquets :

```
1 sudo apt-get update && sudo apt-get upgrade
```



Ensuite nous installons le serveur Apache :

```
1 sudo apt-get install apache2 apache2-doc
```

Mysql :

```
1 sudo apt-get install mysql-server php5-mysql
```

Il existe un script de sécurisation de MySQL qui permet d'avoir MySQL avec un peu plus de sécurité en retirant par exemple le remote login sur root :

```
1 sudo mysql_secure_installation
```

Installation de php5 :


```
1 sudo apt-get install php5-common libapache2-mod-php5 php5-cli
```

Installation des extensions de php5 :

```
1 sudo apt-get install php5-intl php5-gd php5-curl
```

Il ne faudra pas oublier d'ajouter la timezone dans php.ini

Installation de composer par curl :

```
1 //Installation de curl (si nécessaire)
2 sudo apt-get install curl
3 //installation de composer
4 curl -sS https://getcomposer.org/installer | sudo php -- --install-dir=/usr/local/bin -- 
    filename = composer
```

L'application utilise Bower et Grunt qui s'installent grâce à NPM :

```
1 sudo apt-get install nodejs npm
```

Puis on installe Bower et Grunt globalement :

```
1 npm install -g bower
2 npm install -g grunt
```

## 2.3 Installation du projet

Création de la BDD dans Mysql :

```
1 mysql -u root -p
2 //mysql vous demandera votre mot de passe
3 CREATE DATABASE vrp_dev ;
4 exit
```

Importation avec git du projet :

```
1 //si vous n'avez pas git
2 sudo apt-get install git
3
4 git clone vrp@51.254.132.44:var/repos/vrp-rep.git
```

Si on tente d'accéder à l'application, des problèmes d'écriture sur les dossiers app/cache et app/log peuvent arriver. Il faut donc supprimer les fichiers et donner les droits (<http://symfony.com/doc/current/book/installation.html>)

```

1 rm -Rf app/cache && rm -Rf app/logs
2 mkdirmkdir app/cache
3 mkdirmkdir app/logs
4 HTTPDUSER= 'ps axo user,comm | grep -E '[a]pache|[h]ttpd|[_]www|[w]ww-data|[n]ginx' | ↩
    grep -v root | head -1 | cut -d\ -f1'
5 sudo setfacl -R -m u : "$HTTPDUSER":rwX -m u:'whoami':rwX var/cache var/logs
6 sudo setfacl -dR -m u : "$HTTPDUSER":rwX -m u:'whoami':rwX var/cache var/logs

```

Pour accéder à l'application avec une URL, nous allons créer des virtual host.

```

1 //dossier a modifier
2 sudo nano /etc/apache2/sites-enabled/000-default.conf
3
4 //Voici un exemple de deux VirtualHost a adapter
5 <VirtualHost *:80>
6 ServerAdmin webmaster@localhost
7 ServerName vrp.nom_a_modifier.fr
8 <Directory /pathOfProject/web>
9 DocumentRoot /pathOfProject/web
10 Options Indexes FollowSymLinks MultiViews
11 AllowOverride All
12 Require all granted
13 </Directory>
14
15 </VirtualHost>
16
17 <VirtualHost *:80>
18 ServerAdmin webmaster@localhost
19 ServerName beta-vrp.nom_a_modifier.fr
20 DocumentRoot /pathOfProject/web
21 <Directory /pathOfProject/web>
22 Options Indexes FollowSymLinks MultiViews
23 AllowOverride All
24 Require all granted
25 </Directory>
26
27 </VirtualHost>

```

Dans le dossier du projet, il faut installer les dépendances :

```

1 composer install
2 bower install

```

Installation des assets :

```

1 php app/console asset:install web
2 php app/console asset:dump
3 php app/console cache:clear --env=dev

```

Vous pouvez maintenant accéder à l'application grâce à l'url que vous avez définie dans le virtual host.

Si cela ne fonctionne pas, vous pouvez toujours utiliser le serveur PHP avec la commande :

```

1 php app/console server:run

```

## 2.4 Installation des Backup

Pour avoir des données sur notre site en local, nous allons charger une ancienne backup. Il faut tout d'abord modifier le fichier du projet **app/config/config.yml** et remplacer la ligne :

```
1 default_file_path: %kernel.root_dir%/../web/uploads_%kernel.environment%
```

par celle-ci:

```
1 default_file_path: %kernel.root_dir%/../web/uploads_prod
```

Il faut maintenant récupérer les scripts de backup / restauration :

```
1 scp -r vrp@51.254.132.44:backup/ .
```

Le script prend en entrée le nom de l'application à partir de laquelle on souhaite faire la restauration. Pour faire la restauration, il faut créer le dossier **uploads\_prod** dans le dossier **web** où seront stockés les fichiers uploadés. Puis il faut lancer le script grâce aux commandes :

```
1 cd vrp-rep
2 mkdir web/uploads_prod
3 cd ../backup/src
4 bash restore.sh vrp-rep
```

### 3 Déploiement des modifications sur le serveur

Le serveur héberge un dossier `.git` qui s'occupe de versionner le projet. Un hook permet le déploiement automatique :

- Sur la beta lorsqu'un push est réalisé sur la branche « develop » ;
- Sur la prod lorsqu'un push est réalisé sur la branche « master ».

Le hook réalise les actions suivantes :

1. Récupération de la dernière version des sources pour la branche en question ;
2. Installation des nouvelles dépendances si besoin (avec Composer install et bower install) ;
3. Suppression du cache (php app/console cache:clear --env=prod --no-debug) ;
4. Migration Doctrine  
(php app/console --no-interaction doctrine:migrations:migrate --env=prod) ;
5. Installation des assets.

Une grande partie des tests se réalisant sur la bêta, il est donc **très important** de déployer en **beta** avant de déployer en prod.

La bêta est accessible à l'url : <http://beta-version.vrp-rep.org/>.

# F

## Documentation utilisateur

Dans cette partie, je vais parler de l'utilisation des fonctionnalités que j'ai réalisé.

### 1 Page de recherche

Dans la page d'accueil, taper votre recherche dans le champ de texte puis appuyer sur le bouton « Search ». Si aucune recherche n'est mise, toutes les datasets, références et variantes seront chargés.

The screenshot shows the VRP-REP website homepage. At the top, there is a navigation bar with links: HOME, ABOUT, FAQ, FORUM, RESOURCES, and BROWSE. The main content area is divided into two columns. The left column contains a welcome message and a list of features. The right column contains a login form. Below the welcome message, there is a search bar with the text 'mendoza' and a 'Search' button. A green 'Sign Up' button is located at the bottom left. Two arrows point to the search bar and the 'Search' button, with labels (1) and (2) respectively.

**VRP-REP**

HOME ABOUT FAQ FORUM RESOURCES BROWSE

Welcome to VRP-REP: the vehicle routing problem repository

VRP-REP is a collaborative open-data platform for sharing vehicle routing problem benchmark instances and solutions. With VRP-REP you can:

- Upload VRP instances in a pre-defined format (the [VRP-REP instance specification](#)) or your own format
- Share instances with other users or the general public
- Share instances privately with referees and editors during the reviewing process
- Share your solutions with the community
- Download instances shared by other users
- Track the best known solutions for registered datasets
- Subscribe to notifications of newly contributed datasets and solutions
- Access open-source code for [solution checking](#) and instance generation
- Propose extensions and enhancements to the [VRP-REP instance specification](#)
- Browse the VRP-REP database of VRP publications

Search problem variants, references, datasets :

mendoza

Search

Sign Up

Log In

Username:

Password:

☐ Remember me

[Forgot your password ?](#)

Login

(1) Ecrire sa recherche

(2) Appuyer sur le bouton "Search"

Figure 1 – Guide d'utilisateur : page d'accueil du site.

Ensuite, nous sommes redirigés vers la page « Search », nous pouvons voir le nombre de résultats trouvés pour chaque catégorie. Il suffit de cliquer sur les bandeaux pour déployer les tableaux.

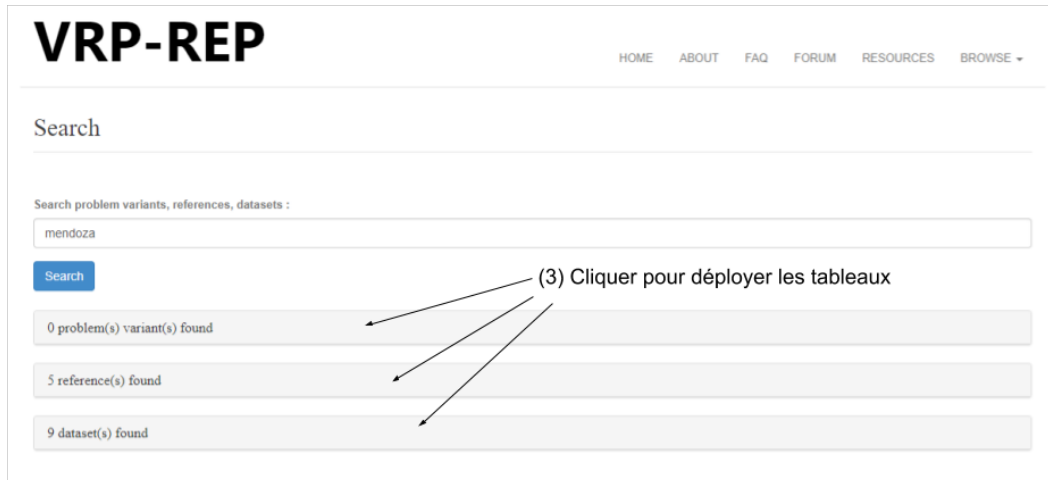


Figure 2 – Guide d'utilisateur : page « Search ».

Nous pouvons donc voir les résultats avec les mots-clés en surbrillance, une nouvelle recherche peut être effectuée.

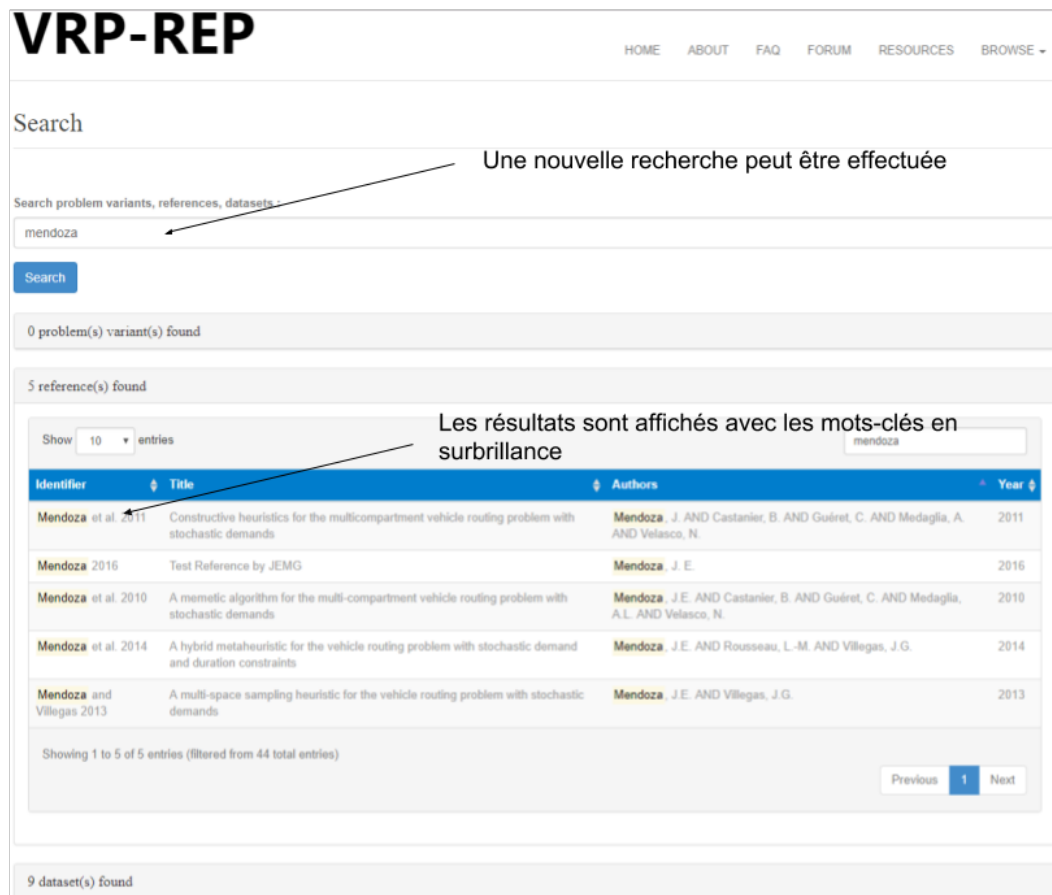
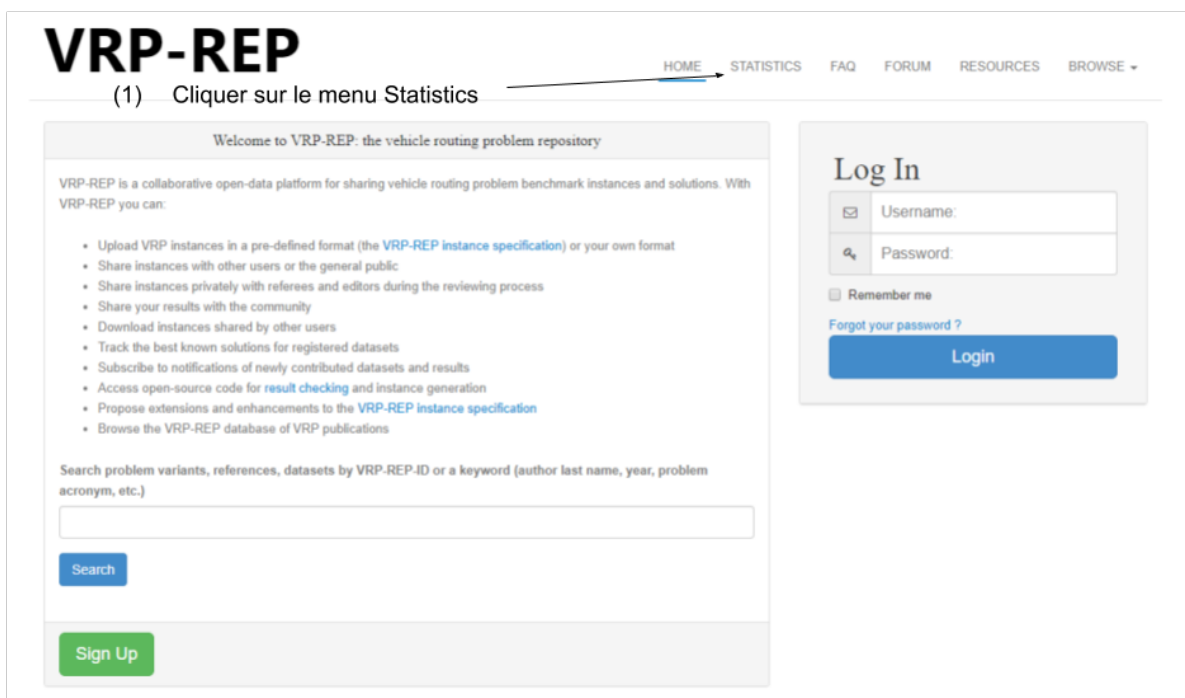


Figure 3 – Guide d'utilisateur : page « Search » dépliées.

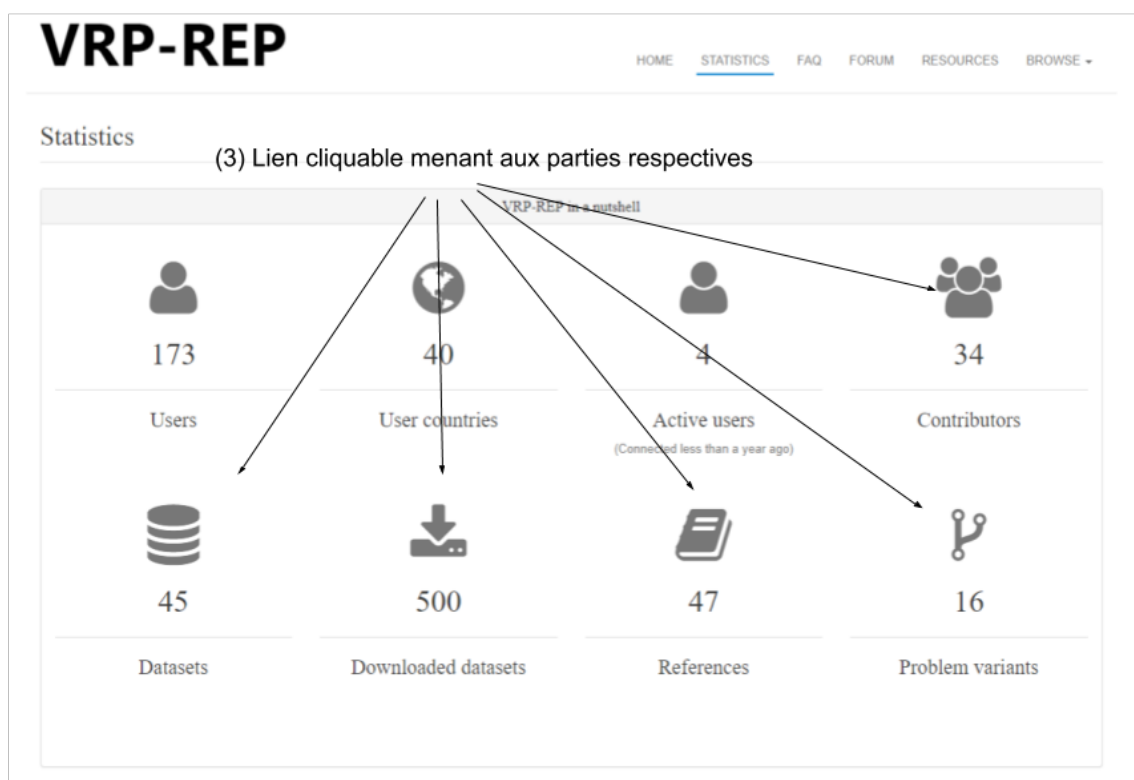
## 2 Page Statistics

Pour accéder à la page Statistics, cliquer dans le menu sur « Statistics ».



**Figure 4** – Guide d'utilisateur : page d'accueil du site.

Cette page affiche les statistiques du site en temps réel. Des liens permettent de rediriger vers leur partie respective pour les datasets, références, variantes et contributeurs.



**Figure 5** – Guide d'utilisateur : page « Statistics ».

# G

## Documents de tests

### 1 Plan de tests

Premièrement, pour notre plan de test, nous allons réaliser plusieurs scénarios fonctionnant avec la méthode all single. Chaque scénario va parcourir chaque cas de notre fonctionnalité et analyser leurs résultats par rapport aux résultats voulus.

Deuxièmement, les modifications étant déployées sur la bêta, un groupe d'utilisateur du site teste l'application et retourne les éventuelles erreurs.

### 2 Grille de test



Numéro Test	Type Test	Fonctionnalités testées	Scénarios de test	Données	Vérifications
T01	Fonctionnel	Correction des vendors	Nous allons vérifier que les pages accessibles en mode non-connecté fonctionnent Nous allons vérifier que les pages en mode connecté ne sont pas accessibles par un utilisateur non-connecté	Mots-clés utilisés dans les tableaux de données : "ac"	Les pages chargent normalement : Ok Les recherches dans les tableaux fonctionnent : Ok
T02	Fonctionnel	Correction des vendors	Nous allons vérifier le fonctionnement des pages accessibles en mode utilisateur connecté sans les pages contribute	Un utilisateur valide et activé	Les pages ne sont pas visibles : Ok Les pages ne sont pas accessibles par l'url : Ok
T03	Fonctionnel	Correction des vendors	Nous allons vérifier que la page contribue dataset	Contribute dataset : dataset au format vrp et non-vrp	Les pages chargent normalement : Ok L'utilisateur arrive à se connecter : Ok La page charge normalement : Ok Ajout d'un dataset au format vrp en public : Ok
T04	Fonctionnel	Correction des vendors	Nous allons vérifier que la page contribue solution accessible en mode utilisateur connecté fonctionne	Contribute solution : une solution	Echec lors d'ajout d'une dataset au format non-vrp (vrp demandée) : Ok Ajout dataset au format non vrp en private : Ok
T05	Fonctionnel	Correction des vendors	Nous allons vérifier que la page contribue solution accessible en mode utilisateur connecté fonctionne	Contribute solution : une solution	Les pages chargent normalement : Ok Echec d'ajout d'une solution au mauvais format : Ok Echec ajout d'une solution à la mauvaise dataset : Ok Ajout d'une solution au format vrp à la bonne dataset: Ok
T06	Fonctionnel	Correction des vendors	Inscription d'un utilisateur respectant et d'un utilisateur ne respectant pas la validation	Des renseignements utilisateurs valides et non-valides	S'inscrire avec un utilisateur : Ok L'utilisateur ne peut pas s'inscrire avec un email déjà existant : Ok L'inscription ne marche pas avec une confirmation par mot de passe non valide : Ok
T07	Fonctionnel	Module Capcha	Envoi de mail aux administrateurs par la page contactez nous avec et sans la validation du capcha	Un formulaire valide	Inscrire un utilisateur avec un code capcha valide : Ok Inscription bloquée d'un utilisateur avec un code capcha non-valide : Ok
T08	Fonctionnel	Module Capcha	Faire une recherche avec un champs vide	Le champ = ""	Envoyer un mail avec un code capcha valide : Ok Envoi de mail bloqué avec un code capcha non-valide : Ok
T09	Fonctionnel	Page de recherche	Faire une recherche avec un mot-clé	Mots-clés utilisés dans les tableaux de données : "ac"	Accès à la page de recherche : Ok Affichage de toutes les données publiques : Ok Non-affichage des datasets privés : Ok
T10	Fonctionnel	Page de recherche	Vérifier les références de toutes les solutions pour une variante et un dataset donné	Variante : CVRP Dataset : Christofides et al. 1979 - CMT	Accès à la page de recherche : Ok Affichage de toutes les données incluant le mot-clé : Ok Non affichage de toutes les données n'incluant pas le mot-clé : Ok
T11	Fonctionnel	Tri des références dans le popover			Toutes les références sont triées par ordre alphabétiques : Ok

Figure 1 – Première partie de la grille de test.

T14	Fonctionnel	Lien des BKS vers les datasets	Vérifier que toutes les solutions pour une variante et un dataset donné renvoient vers le bon	Variante : CVRP Dataset : Christofides et al. 1979 - CMT	Chaque dataset renvoie vers le dataset choisi : Ok
T15	Fonctionnel	Page About	Vérifier que toutes les données affichées sont correctes en comparant avec la BDD		Utilisateurs actifs 3 : Ok Pays d'utilisateurs actifs 3 : Ok Datasets 43 : Ok Références 45 : Ok Variantes 15 : Ok Collaborateurs 34 : Ok
T16	Fonctionnel	Page About	Vérifier qu'ajouter un utilisateur et le connecter incrémente le nombre d'utilisateurs actifs de 1	Un nouvel utilisateur	Utilisateurs actifs 4: Ok
T17	Fonctionnel	Page About	Vérifier qu'ajouter un utilisateur d'un autre pays et de le connecter incrémente le nombre de pays d'utilisateurs actifs de 1	Un nouvel utilisateur d'un pays différent de la liste	Pays d'utilisateurs actifs 4 : Ok
T18	Fonctionnel	Page About	Vérifier qu'ajouter un dataset incrémente le nombre de datasets de 1	Un dataset	Datasets 44 : Ok
T19	Fonctionnel	Page About	Vérifier qu'ajouter une référence incrémente le nombre de références de 1	Une référence	Références 46 : Ok
T20	Fonctionnel	Page About	Vérifier qu'ajouter un collaborateur incrémente le nombre de collaborateurs de 1	Un collaborateur	Collaborateurs 35 : Ok
T21	Fonctionnel	Page About	Vérifier qu'ajouter une variante incrémente le nombre de variantes de 1	Une variante	Variantes 16 : Ok
T22	Fonctionnel	Script répertoire tmp	Vérifier qu'il supprime bien les éléments du bon répertoire en créant au préalable des fichiers à l'intérieur	2 fichiers	Les deux fichiers ont été supprimés : Ok Le dossier n'a pas été supprimé : Ok

Figure 2 – Deuxième partie de la grille de test.



## Comptes rendus hebdomadaires

### **Compte rendu n°1 du 21/09/2017**

Au cours de ses deux premières séances, je me suis attelé à comprendre en quoi ma tâche était liée. À partir du résumé du livre « An Overview of Vehicle Routing Problems », j'ai pu comprendre les différents problèmes VRP.

Ensuite, comme il me l'a été demandé, j'ai visionné plusieurs tutoriels afin de comprendre les différents mécanismes du Framework Symphony et du langage PHP (Bundle, namespace, twig, ...).

En parallèle, j'ai commencé l'installation des différents outils informatique qui vont m'être nécessaires pour la réalisation de mon projet d'étude.

### **Compte rendu n°2 du 28/09/2017**

Lors de cette semaine j'ai fini l'installation des logiciels pour symphony. J'ai récupéré et lu le PRD de Valentin Maerten. Ensuite j'ai bien entamé l'état de l'art.

### **Compte rendu n°3 du 12/10/2017**

Cette semaine j'ai commencé un tutoriel pour m'entraîner sur la technologie Symphony 2. En parallèle, je réalise le cahier de spécification du PRD.

### **Compte rendu n°4 du 19/10/2017**

Cette semaine j'ai quasiment fini l'installation du projet en localhost sur mon ordinateur (on a un petit problème avec les virtual Host). En parallèle, j'ai continué l'avancement de mon cahier de spécifications et état de l'art.

### **Compte rendu n°5 du 26/10/2017**

J'ai rencontré divers soucis d'installation du projet sur un serveur apache en local, plus précisément avec les virtual host que j'ai réglé. J'ai également cherché un membre de l'équipe pédagogique qui pourrait m'aider dans mon problème. Il reste toujours des problèmes avec Apache.

**Compte rendu n°6 du 09/11/2017**

Cette semaine je me suis attelé à avancer le cahier de spécification, état de l'art. J'ai fini de debugger la configuration du serveur Apache en local (sur ma machine). J'ai fixé les versions de composer.json, il me reste donc à fixer les problèmes symfony présent sur le projet.

**Compte rendu n°7 du 23/11/2017**

Cette semaine je me suis concentré sur la finalisation du cahier de spécifications et sur l'état de l'art. J'ai envoyé le rapport contenant une première version du cahier de spécifications à Monsieur Mendoza. La deuxième partie état de l'art n'est pas complètement terminée. J'ai effectué une relecture de mon cahier de spécification avec M. Soukhal.

**Compte rendu n°8 du 30/11/2017**

Migration du rapport sous latex. J'ai effectué une relecture de mon cahier de spécification et de mon état de l'art avec M. Soukhal.

**Compte rendu n°9 du 7/12/2017**

Je finis la rédaction de tous mes documents jusque-là fin de semaine. Je reprendrai le côté technique la semaine prochaine. J'ai effectué une relecture de mon cahier de spécification avec M. Mendoza.

# Webographie

- [1] Nils ADERMANN et Jordi BOGGIANO. *Composer*. URL : <https://getcomposer.org/>.
- [2] BOWER. *A package manager for the web*. Sous la dir. de BOWER. 28 mar. 2018. URL : <https://bower.io/>.
- [3] Pierre-Yves CHIBON, Pascal PAROIS et Charles-Antoine COURET. *Guide d'installation de Fedora en images*. Sous la dir. de FEDORA. URL : [https://doc.fedora-fr.org/wiki/Guide\\_d%27installation\\_de\\_Fedora\\_en\\_images](https://doc.fedora-fr.org/wiki/Guide_d%27installation_de_Fedora_en_images).
- [4] LINUX CONFIG. *Configure Apache VirtualHost on Fedora*. Sous la dir. de LINUX CONFIG. 10 déc. 2017. URL : <https://linuxconfig.org/configure-apache-virtualhost-on-fedora>.
- [5] DOCTRINE. *Annotation references*. Sous la dir. de DOCTRINE. URL : <http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/reference/annotations-reference.html>.
- [6] DOCTRINE. *What is Doctrine?* URL : <http://docs.doctrine-project.org/en/latest/tutorials/getting-started.html#what-is-doctrine>.
- [7] npm ENTERPRISE. *npm*. Sous la dir. de npm ENTERPRISE. 28 mar. 2018. URL : <https://www.npmjs.com/>.
- [8] Jorge Ernesto Mendoza GIMENEZ. *Modèle des instances VRP-REP*. URL : <https://github.com/VRP-REP/model/blob/master/src/main/xsd/instance.xsd>.
- [9] JETBRAINS. *Install and set up PhpStorm*. Sous la dir. de JETBRAINS. 10 déc. 2017. URL : <https://www.jetbrains.com/help/phpstorm/installing-and-launching.html>.
- [10] PACKAGIST. *Packagist*. URL : <https://packagist.org/>.
- [11] Fedora PROJECT. *Apache HTTP Server*. Sous la dir. de Fedora PROJECT. 10 déc. 2017. URL : [http://fedoraproject.org/wiki/Apache\\_HTTP\\_Server](http://fedoraproject.org/wiki/Apache_HTTP_Server).
- [12] SENSIO LABS. *Controller*. URL : <https://symfony.com/doc/current/controller.html>.
- [13] SENSIO LABS. *Creating and Using Templates*. URL : <http://symfony.com/doc/3.3/templating.html>.
- [14] SENSIO LABS. *DoctrineMigrationsBundle*. Sous la dir. de SENSIO LABS. URL : <http://symfony.com/doc/current/bundles/DoctrineMigrationsBundle/index.html>.
- [15] SENSIO LABS. *Symfony and HTTP Fundamentals*. URL : [https://symfony.com/doc/current/introduction/http\\_fundamentals.html](https://symfony.com/doc/current/introduction/http_fundamentals.html).

- [16] SENSIOLABS. *The Bundle System*. URL : <https://symfony.com/doc/3.0/bundles.html>.
- [17] SENSIOLABS. *What is Symfony?* URL : <https://symfony.com/fr/what-is-symfony>.
- [18] SENSIOLABS. *Coding Standards*. Sous la dir. de SENSIOLABS. 28 mar. 2018. URL : <https://symfony.com/doc/current/contributing/code/standards.html>.
- [19] SENSIOLABS. *Using Bower with Symfony*. Sous la dir. de SENSIOLABS. 23 mar. 2018. URL : <https://symfony.com/doc/3.1/frontend/bower.html>.
- [20] SPRYMEDIA. *Datatable*. Sous la dir. de SPRYMEDIA. 23 mar. 2018. URL : <https://datatables.net/>.
- [21] UBUNTU. *Programmer des tâches avec CRON*. Sous la dir. d'UBUNTU. 28 mar. 2018. URL : <https://doc.ubuntu-fr.org/cron>.
- [22] WIKIPEDIA. *Data mapper pattern*. URL : [https://en.wikipedia.org/wiki/Data\\_mapper\\_pattern](https://en.wikipedia.org/wiki/Data_mapper_pattern).
- [23] WIKIPEDIA. *Optimum de Pareto*. URL : [https://fr.wikipedia.org/wiki/Optimum\\_de\\_Pareto](https://fr.wikipedia.org/wiki/Optimum_de_Pareto).



## Bibliographie

- [1] Nicolas JOZEFOWIEZ, Frédéric SEMET et El-Ghazali TALBI, éd. *Multi-objective vehicle routing problems*. 31 mar. 2006.
- [2] Valentin MAERTEN. « VRP-REP ». Mém. de mast. Ecole Polytechnique de l'Université François Rabelais de Tours, 2015-2016.
- [3] Paolo TOTH et Daniele VIGO. *The vehicle routing problem*. 2002.

# Amélioration de la plateforme VRP-REP

Antoine Bernier

Encadrement : Jorge Ernesto Mendoza

## Le projet VRP-REP

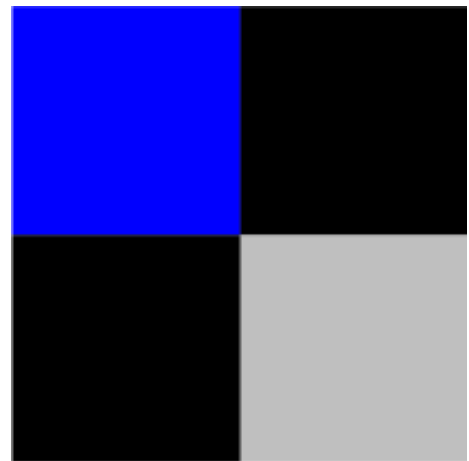
Le projet VRP-REP (Vehicule Routing Problem – Repository) a pour but de fournir à la communauté scientifique une plateforme collaborative de partage de données pour les problèmes de tournées de véhicules. Depuis 2012, plus de 20 personnes de 5 pays différents (étudiants, développeurs, et chercheurs) ont contribué au développement de la plateforme <http://www.vrp-rep.org>.



Image résumant le problème de tournées de véhicules.

## Objectif du PRD

- Ajout d'un module CAPTCHA permettant de bloquer l'inscription des bots ;
- Ajout d'un menu « Statistics » pour afficher des statistiques ;
- Migration d'une solution monocritère - multicritère ;
- Correction de différents BUG.



Logo VRP-REP.

## Technologies utilisées

- Symfony 2 ;
- Doctrine avec BDD MySQL ;
- TWIG ;
- Environnement Linux.



Symfony

Logo Symfony



# Amélioration de la plateforme VRP-REP

Antoine Bernier

Encadrement : Jorge Ernesto Mendoza

## Le projet VRP-REP

Le projet VRP-REP (Vehicule Routing Problem – Repository) a pour but de fournir à la communauté scientifique une plateforme collaborative de partage de données pour les problèmes de tournées de véhicules. Depuis 2012, plus de 20 personnes de 5 pays différents (étudiants, développeurs, et chercheurs) ont contribué au développement de la plateforme <http://www.vrp-rep.org>.

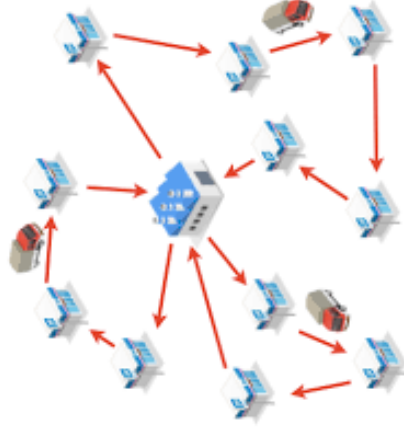


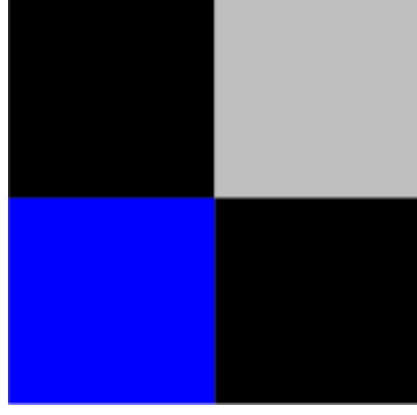
Image résumant le problème de tournées de véhicules.

## Objectif du PRD

- Ajout d'un module CAPTCHA permettant de bloquer l'inscription des bots ;
- Ajout d'un menu « Statistics » pour afficher des statistiques ;
- Migration d'une solution monocritère - multicritère ;
- Correction de différents BUG.

## Technologies utilisées

- Symfony 2 ;
- Doctrine avec BDD MySQL ;
- TWIG ;
- Environnement Linux.



Logo VRP-REP.



Symfony

Logo Symfony



# Amélioration de la plateforme VRP-REP

## Résumé

Le Projet de Recherche & Développement (PR&D), réalisé en 5ème année, s'inscrit dans la formation dispensée à l'École Polytechnique de l'Université de Tours et constitue une réelle expérience en termes de conduite de projet. Le PR&D se déroule du 20 septembre 2017 au 30 mars 2018 à raison de deux jours par semaine à temps plein. Il donne lieu à la rédaction d'un rapport avec le cahier de spécifications et à deux soutenances durant lesquelles le travail effectué sera exposé.

Le PR&D a pour objectif d'améliorer la plateforme VRP-REP (<http://www.vrp-rep.org>), une plateforme collaborative permettant à la communauté scientifique de partager des données pour les problèmes de tournées de véhicules.

Ce projet fonctionne sous le Framework Symfony2 utilisant Doctrine, TWIG, et une base de données SQL.

## Mots-clés

PR&D, VRP-REP, Problème de tournées de véhicules, Projet de recherche et développement, Symfony 2, Doctrine, TWIG

## Abstract

The Research & Development Project (PR&D), realized in 5th year, is part of the training given at the Polytechnic School of the University of Tours and constitutes a real experience in terms of project management. The PR&D begins from 20 September 2017 to 30 March 2018, two days a week on a full-time basis. It gives rise to the drafting of a report with the specifications book and two defences during which the work carried out will be exposed.

The objective of PR&D is to improve the VRP-REP platform (<http://www.vrp-rep.org>), a collaborative platform allowing the scientific community to share data for vehicle routing problems.

This project runs on Symfony2 Framework using Doctrine, TWIG, and an SQL database.

## Keywords

PR&D, VRP-REP, Vehicle Routing Problem, Research and Development Project, Symfony 2, Doctrine, TWIG

**Tuteur académique**

Jorge Ernesto MENDOZA

**Étudiant**

Antoine BERNIER (DI5)