

Conception et Implémentation d’une solution de control de flotte automobile

Version 1.0

Abréviations

**NH** : Noubru Holding

**ADISCO** : Africa Distribution Company

**ERP**: Enterprise Ressource Planning

**SI**: Système d’Informations

Table de matières

[Introduction Générale 3](#_Toc10650680)

[Contexte 3](#_Toc10650681)

[Problématique 4](#_Toc10650682)

[Motivations et Objectifs 4](#_Toc10650683)

[Méthodologie 4](#_Toc10650684)

[Plan du mémoire 5](#_Toc10650685)

[1. Etat de l’art 6](#_Toc10650686)

[2. Analyse 11](#_Toc10650687)

[2.1. Spécification fonctionnelles 11](#_Toc10650688)

[2.2. Spécification non fonctionnelles 11](#_Toc10650689)

[2.3. Les parties prenantes 12](#_Toc10650690)

[2.4. Les acteurs 12](#_Toc10650691)

[2.5. Diagramme de cas d’utilisation 12](#_Toc10650692)

[2.6. Description textuelle des cas d’utilisation 14](#_Toc10650693)

[i) Gestion de l’inventaire 14](#_Toc10650694)

[ii) Tracking de véhicule 19](#_Toc10650695)

[2.7. Contraintes 22](#_Toc10650696)

[2.8. Architecture globale 22](#_Toc10650697)

[3. Conception 23](#_Toc10650698)

[4. Implémentation 24](#_Toc10650699)

[Conclusion 25](#_Toc10650700)

[Références 26](#_Toc10650701)

Introduction Générale

Un système de Fleet Management (en Français Gestion de Flotte) est un système permettant de gérer l’ensemble des véhicules d’une entreprise. Un tel système pourrait s’avérer très bénéfique pour une entreprise car, il permettrait à cette dernière d’avoir un control sur sa flotte de véhicules et aussi de réduire les coûts engendrés par celle-ci. Il nous est à cet effet demandé d’en concevoir un.

Contexte

Membre du grand groupe d’entreprises NH, l’entreprise ADSICO est une société spécialisée dans la logistique et la livraison de colis [**1**]. Forte de sa flotte automobile d’environ quatre cents camions, fourgonnettes et motos de livraison, elle établit son champ d’action sur l’étendue du territoire nationale et même au-delà, ce qui fait d’elle une référence dans le milieu.

Toutefois, la championne de la logistique du groupe NH fait face à un problème, celui de l’incapacité de contrôler les engins de sa propre flotte automobile lorsqu’ils sont sur la route.

En effet, une fois que les véhicules de l’entreprise sont sur la route pour des livraisons, leurs chauffeurs se permettent de faire des « magouilles » à l’insu des dirigeants. Par « magouilles » nous entendons des opérations telles que :

* Retirer de l’essence du réservoir du véhicule pour revendre ensuite
* Exécuter des livraisons au compte d’entreprises ou de particuliers inconnus
* Signaler de fausses pannes ou de ne pas les signaler quand il y en a
* Se permettre des détours en pleine opération de livraison

ADISCO, au vu de ses objectifs a besoin d’une solution qui lui permettrait d’avoir un control encore plus poussé qu’il ne l’est déjà de tous ces véhicules qui constituent sa grande flotte automobile. Le projet qui nous est confié est donc de mettre sur pied une plateforme qui permettrait non seulement d’intégrer les informations concernant les véhicules d’une flotte automobile dans le système d’information de l’entreprise mais aussi de traquer les déplacements et les dépenses faites sur ceux-ci où qu’ils soient.

Problématique

Il y’a de cela environ trois ans, l’entreprise ADSICO décide de baser l’infrastructure de gestion de son SI sur l’ERP OpenSource nommé Odoo, anciennement appelé OpenERP.

Depuis lors, le système de gestion du SI de l’entreprise n’a pas changé. De ce fait, elle exige que la solution que nous allons mettre en place s’intègre parfaitement à son SI, par conséquent à l’ERP Odoo.

Ceci est une contrainte majeure qui nous est imposée dans la mise en œuvre de notre solution. Car, en plus du problème de conception d’une solution de control de flotte automobile, s’ajoute des exiges de modularité et, de comptabilité.

Nous nous interrogeons donc sur les techniques à mettre en œuvre pour modéliser un système non seulement efficace, mais aussi sécurisé et modulaire et surtout intégrable à n’importe quel Système d’Information en occurrence un qui soit basé sur Odoo.

Motivations et Objectifs

L’objectif principal de ce travail est de mettre sur pieds une solution logicielle intégrable à l’ERP Odoo qui permettrait à l’entreprise ADISCO de contrôler les différentes activités des véhicules de sa flotte.

Pour cela, nous allons dans notre solution donner la possibilité à ADISCO de :

* Traquer en temps réel les véhicules de sa flotte
* Etre alerté en cas de détour d’un véhicule pendant une livraison
* Vérifier la véracité les dépenses faites sur les différents véhicules de sa flotte
* Assurer la confidentialité l’intégrité et la disponibilité des informations sur les véhicules de sa flotte

Méthodologie

/\*\*\*

Parler de la technique de gestion de projet utilisée, du processus de génie logiciel et du langage de modélisation utilisé

\*\*\*/

Plan du mémoire

Etat de l’art

De nombreuses institutions dans le monde et même des particuliers ont déjà fait face à ce problème qu’est de surveiller en temps réel les déplacements des engins qui sont les leurs. Raison pour laquelle sur le marché il existe déjà une pléthore de logiciels de tracking, les unes aussi singulières que les autres.

Toutes fois, la plus part de ses solutions existantes semblent avoir en commun un certain nombre de fonctionnalités que nous présenterons dans la suite.

* 1. Présentation de l’ERP Odoo

**Définition**

Un ERP (Enterprise Ressource Planning) en français Progiciel de Gestion Intégré est un de gérer l'ensemble des processus d'une entreprise en intégrant l'ensemble de ses fonctions, dont la gestion des ressources humaines, la gestion comptable et financière, l'aide à la décision, mais aussi la vente, la distribution, l'approvisionnement et le commerce électronique [**2**].

**Historique**

Anciennement OpenERP[2](https://fr.wikipedia.org/wiki/Odoo#cite_note-:0-2) et Tiny ERP, est initialement un [progiciel](https://fr.wikipedia.org/wiki/Progiciel) open-source de [gestion intégré](https://fr.wikipedia.org/wiki/Progiciel_de_gestion_int%C3%A9gr%C3%A9) comprenant de très nombreux modules permettant de simplifier la gestion d’entreprise dans son ensemble. Le logiciel est utilisé par plus de deux millions d’utilisateurs pour gérer leurs entreprises à travers le monde[3](https://fr.wikipedia.org/wiki/Odoo#cite_note-3). Odoo est le système ERP open-source le plus populaire.

Il existe une version community gratuite sous licence LGPLv3, et une enterprise sous licence propriétaire Odoo Enterprise Edition License v1.0.

À l’origine un [progiciel de gestion intégré](https://fr.wikipedia.org/wiki/Progiciel_de_gestion_int%C3%A9gr%C3%A9) (ERP), le logiciel s’est vu étendre ses fonctionnalités à des applications de [front office](https://fr.wikipedia.org/wiki/Application_de_front_office) (CMS, e-commerce, blogs, forums, news, événements, live chat, job offers…) [**8**].

Il se distingue par son extrême modularité qui non le seulement le rend adaptable à n’importe quel SI mais aussi facilement personnalisable.

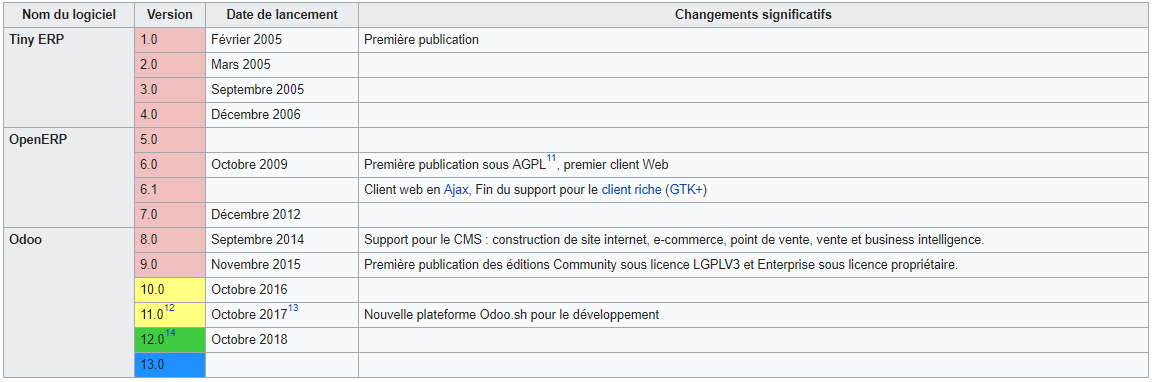


Figure : Historique des versions de Odoo

**Comparaison aux autres ERP**

Selon de multiples articles, Odoo serait le deuxième des ERP les plus utilisés dans le monde. L’image suivante illustre cela.

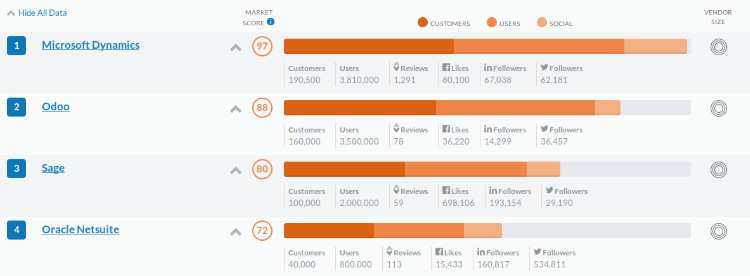


Figure : Les Top ERP de l'année 2018 [**9**]

* 1. Fonctionnalités offertes par les solutions existantes

Parmi les fonctionnalités généralement implémentées par des systèmes de Gestion de Flotte, nous avons relevé les suivantes:

* Tracking de véhicule:

Cette fonctionnalité consiste à tracer sur une carte le chemin parcouru par le véhicule et aussi à avoir en temps réel sa position.

* Gestion de l’inventaire:

Cette fonctionnalité consiste à faire les enregistrements, modifications et suppressions d’informations sur les véhicules de la flotte.

* Planification et control de maintenance de véhicules
* Routage:

Cette fonctionnalité consiste à faire le calcul du chemin le plus court pour le déplacement d’un véhicule.

* Gestion des conducteurs:

Cette fonctionnalité consiste à identifier les conducteurs des véhicules de la flotte et aussi de les contrôler pendant leurs déplacements

* Control des informations des véhicules:

Cette fonctionnalité consiste à avoir en temps réel les informations sur:

* La vitesse du véhicule,
* Le niveau d’essence,
* La batterie,
* La température
* etc…
* **Geofencing**

Cette fonctionnalité consiste à faire une délimitation de l’espace ou un véhicule peut se déplacer

* Emission d’alarme
* Gestion des accidents

Hors mises ces dernières fonctionnalités, les systèmes de Gestion de Flotte les plus poussés intègrent les suivantes:

* Maintenance prédictive:

Cette fonctionnalité consiste à prédire les pannes sur les véhicules de la flotte de manière à faire les maintenances de manière plus stratégique.

* Détection de fatigue du conducteur
  1. Solutions existantes

Parmi les solutions de Fleet Management existantes, nous avons recensé les suivantes :

* **Fleetio**



Une solution de control de flotte conçue en 2012. Elle offre de nombreuses fonctionnalités dont la gestion de l’inventaire, la gestion des conducteurs, le control d’essence la planification des maintenances et le tracking des véhicules. Elle offre une API. Elle est aussi propriétaire [**3**].

* **Onfleet**



Conçue en 2012, cette solution offre des fonctionnalités telles que le tracking de véhicule, la planification de maintenance, le control d’essence, le control de conducteur et le geofencing. Elle offre aussi une API et est propriétaire [**4**].

* **Traccar**



Conçue par l’entrepris russe Traccar Ltd, il s’agit d’une solution offre les fonctionnalités de Tracking de véhicule, geofencing, envoie d’alerte. Elle offre une API et contrairement aux deux précédentes, elle est opensource [**5**].

* **Landmark Tracking**



Solution Sud-Africaine offrant des fonctionnalités telles que le tracking de véhicule, la surveillance de conducteur, le control d’essence, le geofencing. Elle est propriétaire [**6**].

* **Fleet**



Un module Odoo développé par Odoo S.A permettant de toute opération liée à la gestion et à l’inventaire de parc automobile [**7**].

Nous pouvons résumer notre étude comparative de ces solutions existantes de Fleet Management dans le tableau suivant :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Logiciel** | **Tracking de véhicule** | **Control d’essence** | **Surveillance de conducteur** | **Geofence** | **Envoie d’alerte** | **Inventaire** | **OpenSource** |
| Fleetio |  |  |  |  |  |  |  |
| Landmark Tracking |  |  |  |  |  |  |  |
| Onfleet |  |  |  |  |  |  |  |
| Traccar |  |  |  |  |  |  |  |
| Fleet |  |  |  |  |  |  |  |

Tableau : Etude comparative des solutions existantes

Après cette étude comparative, nous avons décidé d’utiliser le système **Traccar** comme noyau de notre solution. Car, elle est opensource par conséquent personnalisable et possède déjà pas mal de fonctionnalités. De plus, le module Fleet de Odoo sera utilisé comme base du module Odoo que nous produirons du fait de ses fonctionnalités d’inventaire déjà existantes.

Analyse

Dans cette partie nous allons énumérer des fonctionnalités qui devront être implémentée dans notre solution. Nous donnerons aussi les contraintes de conception de notre solution, les acteurs du système, le diagramme de cas d’utilisation et en fin la description textuelle des cas d’utilisation.

* 1. Spécification fonctionnelles

Comme fonctionnalités que devra rendre notre solution, nous avons :

* La gestion de l’inventaire

Il s’agit ici des opérations d’ajout, de modification, de suppression de véhicules.

* La gestion et la de maintenance
* Le reporting des dépenses sur les véhicules
* La gestion des contrats des véhicules
* La gestion des conducteurs
* Le tracking temps réel des véhicules
* L’historique des déplacements des véhicules
* La définition de zones de déplacements (geofencing)
* L’envoie des alertes
  1. Spécification non fonctionnelles

Comme spécification non fonctionnelle nous avons les suivantes:

* Le système doit être facile à utiliser
* Le système doit avoir une interface ergonomique
* Le système doit être suffisamment sécurisé pour assurer l’intégrité des données de tracking
* Le système doit être flexible de manière à assurer la maintenabilité
* Le système doit être intégré à Odoo
* Le serveur Traccar doit être utilisé comme noyau pour le serveur de gestion de la flotte
* Le module Fleet Management de Odoo doit être utilisé comme base pour le système
* Les équipements à installer sur les véhicules (traqueurs GPS) devront être compatible avec le serveur Traccar
  1. Les parties prenantes

Il s’agit de toutes les entités affectées par l’existence de notre solution. Nous avons en occurrence :

* L’entreprise ADISCO : il s’agit de ses dirigeants et ses conducteurs de véhicules
* L’entreprise Traccar : car nous utilisons leur système comme noyau pour développer le notre
* L’entreprise Odoo : du fait de l’utilisation de leur ERP
  1. Les acteurs

Il s’agit des différentes personnes ou entités qui interagiront avec notre système. Nous en distinguons deux types, les acteurs principaux et les acteurs secondaires :

* Acteurs principaux :

Nous avons ici :

* Les **Administrateurs**, qui sont ici les dirigeants de l’entreprise ADISCO
* Acteurs secondaires :

Nous avons ici :

* Le **serveur de cartographie** qui nous permettra de visualiser les véhicules sur une carte
* Les **traqueurs GPS** qui seront installés sur les véhicules
  1. Diagramme de cas d’utilisation

Après analyse des fonctionnalités demandez par l’entreprise cliente, nous avons ressorti un ensemble de cas d’utilisation que nous avons séparé en deux diagrammes.

L’un pour ceux concernant l’inventaire du parc, et l’autre pour ceux concernant le tracking des véhicules.

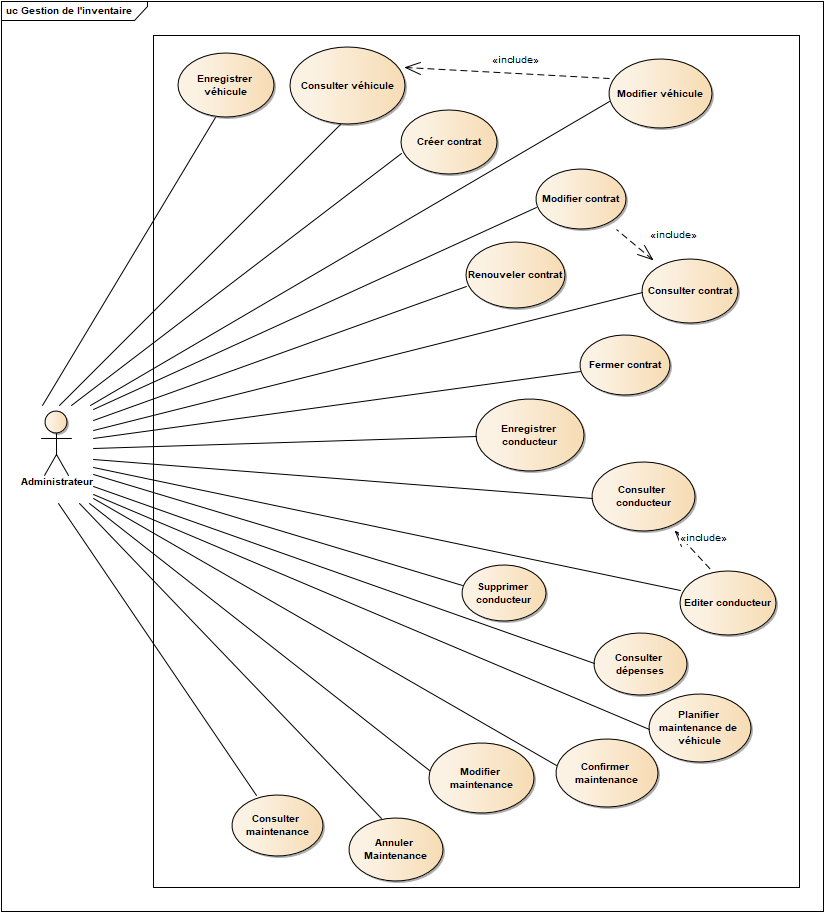


Figure 3:diagramme de cas d’utilisation de gestion de l’inventaire

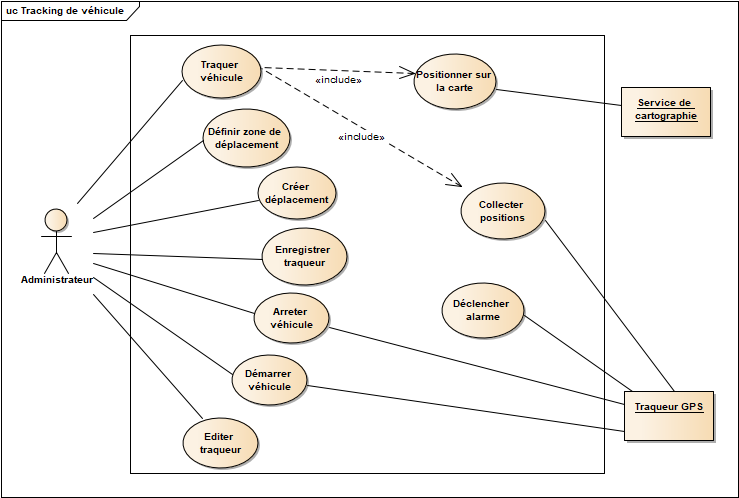


Figure 4: Diagramme de cas d'utilisation de Tracking

* 1. Description textuelle des cas d’utilisation

### Gestion de l’inventaire

* Enregistrer véhicule

ref : 1.1

rôle : Administrateur

postcondition : Un nouveau véhicule est enregistrer sur la plateforme

description :

* L’administrateur ouvre la page de création d’un véhicule
* L’administrateur entre les informations sur le véhicule
* L’administrateur clique sur le bouton de sauvegarde
* Le système enregistre le véhicule dans la base de données
* Le système affiche un message de succès
* Le système affiche les informations du véhicule qui vient d’être enregistré
* Consulter véhicule

ref: 1.2

rôle: Administrateur

description:

* L’administrateur sélectionne le véhicule dont il veut consulter les informations
* Le système collecte les informations sur le véhicule dans la base de données
* Le système affiches les informations sur le véhicule
* Modifier véhicule

ref: 1.3

rôle: Administrateur

postcondition: les informations d’un véhicule sont modifiées

description:

* L’administrateur sélectionne le véhicule dont il veut modifier les informations
* Le système collecte les informations sur le véhicule dans la base de données
* Le système affiches les informations sur le véhicule
* L’administrateur modifie les informations qu’il désire modifier
* L’administrateur clique sur le bouton de sauvegarde
* Le système enregistre les nouvelles informations du véhicule dans la base de données
* Le système affiche un message de succès
* Le système affiche les informations du véhicule qui vient d’être modifié
* Enregistrer conducteur

ref: 1.4

rôle: Administrateur

postcondition: un nouveau conducteur est enregistré sur la plateforme

description:

* L’administrateur ouvre la page de création d’un conducteur
* L’administrateur entre les informations sur le conducteur
* L’administrateur clique sur le bouton de sauvegarde
* Le système enregistre le véhicule dans la base de données
* Le système affiche un message de succès
* Le système affiche les informations du conducteur qui vient d’être enregistré
* Consulter conducteur

ref: 1.5

rôle: Administrateur

description:

* L’administrateur sélectionne le conducteur dont il veut consulter les informations
* Le système collecte les informations sur le conducteur dans la base de données
* Le système affiches les informations sur le conducteur
* Editer conducteur

ref: 1.6

rôle: Administrateur

postcondition: les informations d’un conducteur sont modifiées

description:

* L’administrateur sélectionne le conducteur dont il veut modifier les informations
* Le système collecte les informations sur le conducteur dans la base de données
* Le système affiches les informations sur le conducteur
* L’administrateur modifie les informations qu’il désire modifier
* L’administrateur clique sur le bouton de sauvegarde
* Le système enregistre les nouvelles informations du conducteur dans la base de données
* Le système affiche un message de succès
* Le système affiche les informations du conducteur qui vient d’être modifié
* Supprimer conducteur

ref: 1.7

rôle: Administrateur

postcondition: un conducteur est supprimé de la plateforme

description:

* L’administrateur sélectionne le conducteur qu’il veut supprimer
* L’administrateur clique sur le bouton de suppression
* Le système supprime le conducteur
* Le système affiche un message de succès
* Le système affiche la liste des conducteurs
* Créer contrat

ref: 1.8

rôle: Administrateur

objectif: créer un contrat lié à un véhicule

précondition: le véhicule doit déjà exister

postcondition: un contrat est ajouté à un véhicule

description:

* L’administrateur ouvre la page de création d’un contrat
* L’administrateur sélectionne le véhicule concerné
* L’administrateur entre les informations sur le contrat
* L’administrateur clique sur le bouton de sauvegarde
* Le système enregistre le contrat
* Le système affiche un message de succès
* Le système affiche les informations sur le contrat
* Consulter contrat

ref: 1.9

rôle: Administrateur

description:

* L’administrateur sélectionne le contrat dont il veut consulter les informations
* Le système collecte les informations sur le contrat dans la base de données
* Le système affiche les informations sur le contrat
* Modifier contrat

ref: 1.10

rôle: Administrateur

objectif: modifier les informations d’un contrat

postcondition: les informations d’un contrat sont modifiées dans le système

description:

* L’administrateur sélectionne le contrat dont il veut modifier les informations
* Le système collecte les informations sur le contrat dans la base de données
* Le système affiche les informations sur le contrat
* L’administrateur modifie les informations qu’il désire modifier
* L’administrateur clique sur le bouton de sauvegarde
* Le système enregistre les nouvelles informations du contrat dans la base de données
* Le système affiche un message de succès
* Le système affiche les informations du contrat qui vient d’être modifié
* Renouveler contrat

ref: 1.11

rôle: Administrateur

précondition: le contrat doit exister

postcondition: la période de validité du contrat est mise à jour dans le système

description:

* L’administrateur sélectionne le contrat dont il veut faire le renouvellement
* Le système collecte les informations sur le contrat dans la base de données
* Le système affiche les informations sur le contrat
* L’administrateur clique sur le bouton de renouvellement
* L’administrateur entre les nouvelles informations du contrat
* L’administrateur clique sur le bouton de sauvegarde
* Le système enregistre les nouvelles informations
* Le système affiche les informations du contrat
* Fermer contrat

ref: 1.12

rôle: Administrateur

précondition: le contrat doit exister et ne pas déjà être fermé

postcondition: le contrat n’est plus valide

description:

* L’administrateur sélectionne le contrat qu’il veut fermer
* L’administrateur clique sur le bouton de fermeture
* Le système change l’état du contrat à fermé
* Planifier maintenance véhicule

ref: 1.13

rôle: Administrateur

objectif: Enregistrer à l’avance dans les systèmes des opérations de maintenances qui seront faites sur un véhicule. Une maintenance peut être de tout type (roue, batterie, lavage du véhicule, moteur, etc. …)

postcondition: une nouvelle opération de maintenance est créé dans le système

description:

* L'administrateur clique sur le bouton de planification de maintenance sur la page de consultation d’un véhicule ou alors clique sur le bouton de création de maintenance à l'accueil
* L’administrateur entre les informations sur l’opération de maintenance
* Le système enregistre les la maintenance
* Le système affiche un message de succès
* Le système affiche les informations de la maintenance
* Consulter maintenance

ref: 1.14

rôle: Administrateur

objectif: Consulter les informations d’une opération de maintenance

description:

* L’administrateur sélectionne la maintenance dont il veut consulter les informations
* Le système collecte les informations sur la maintenance dans la base de données
* Le système affiches les informations sur la maintenance
* Confirmer maintenance

ref: 1.15

rôle: Administrateur

objectif: Confirmer qu’une opération de maintenance a été effectuée

précondition: la maintenance ne doit pas être annulée ni confirmée

postcondition: l’état de la maintenance est “confirmé”

description:

* L'administrateur ouvre la page de consultation de la maintenance
* L’administrateur clique sur le bouton de confirmation
* Le système change l’état de la maintenance et la passe à “confirmé”
* Le système affiche un message de succès
* Annuler maintenance

ref: 1.16

rôle: Administrateur

objectif: Confirmer qu’une opération de maintenance a été effectuée

précondition: la maintenance ne doit pas être annulée ni confirmée

postcondition: l’état de la maintenance est “annulé”

description:

* L'administrateur ouvre la page de consultation de la maintenance
* L’administrateur clique sur le bouton d’annulation
* Le système change l’état de la maintenance et la passe à “annulé”
* Le système affiche un message de succès
* Modifier maintenance

ref: 1.17

rôle: Administrateur

postcondition: les informations de la maintenance sont modifiées

description:

* L’administrateur sélectionne la maintenance dont il veut modifier les informations
* Le système collecte les informations sur la maintenance dans la base de données
* Le système affiches les informations sur la maintenance
* L’administrateur modifie les informations qu’il désire modifier
* L’administrateur clique sur le bouton de sauvegarde
* Le système enregistre les nouvelles informations de la maintenance dans la base de données
* Le système affiche un message de succès
* Le système affiche les informations de la maintenance qui vient d’être modifiée

### Tracking de véhicule

* Enregistrer traqueur

ref: 2.1

rôle: Administrateur

objectif: Enregistrer un nouveau traqueur GPS dans le système et le rattacher à un véhicule

description:

* L’administrateur ouvre la page de création d’un traqueur
* L’administrateur sélectionne le véhicule concerné
* L’administrateur entre les informations sur le traqueur
* L’administrateur clique sur le bouton de sauvegarde
* Le système enregistre le traqueur
* Le système affiche un message de succès
* Le système affiche les informations sur le traqueur
* Editer traqueur

ref: 2.2

rôle: Administrateur

objectif: Modifier les informations sur un traqueur

description:

* L’administrateur sélectionne le traqueur dont il veut modifier les informations
* Le système collecte les informations sur le traqueur dans la base de données
* Le système affiches les informations sur le traqueur
* L’administrateur modifie les informations qu’il désire modifier
* L’administrateur clique sur le bouton de sauvegarde
* Le système enregistre les nouvelles informations du traqueur dans la base de données
* Le système affiche un message de succès
* Le système affiche les informations du traqueur qui vient d’être modifié
* Définir zone de déplacement

ref: 2.3

rôle: Administrateur

objectif: Définir une zone géographique dans laquelle un véhicule doit obligatoirement se déplacer

description:

* L’administrateur ouvre la page de consultation d’un véhicule
* L’administrateur dessine la zone de déplacement sur une carte
* Le système enregistre la zone de déplacement
* Créer déplacement

ref: 2.4

rôle: Administrateur

objectif: créer un nouveau déplacement (livraison ou autre) pour un véhicule

description:

* L’administrateur ouvre la page de création d’un déplacement
* L’administrateur sélectionne le véhicule concerné
* L’administrateur peut définir une zone de déplacement
* L’administrateur entre les informations sur le déplacement
* L’administrateur clique sur le bouton de sauvegarde
* Le système enregistre le déplacement
* Le système affiche les informations sur le déplacement et propose de lancer le tracking en temps réel du véhicule
* Traquer véhicule

ref: 2.5

rôle: Administrateur

objectif: Suivre en temps réel les déplacements d’un véhicule

description:

* L’administrateur clique sur le bouton de tracking temps réel qui est soit sur la page de consultation d’un véhicule soit sur la page d’un déplacement
* Le système ouvre une carte permettant de suivre le véhicule concerné
* Arrêter véhicule

ref: 2.6

rôle: Administrateur

objectif: Arrêter le moteur d’un véhicule à distance

précondition: le véhicule doit être en marche

postcondition: l’état du véhicule est “arrêt”

description:

* L’administrateur appuie sur le bouton d’arrêt
* Le système envoie un signal au traqueur GPS du véhicule concerné
* Le traqueur GPS exécute la commande d’arrêt du véhicule
* Le traqueur envoie un signal au système
* le système affiche un message de succès
* Démarrer véhicule

ref: 2.7

rôle: Administrateur

objectif: Démarrer le moteur d’un véhicule à distance

précondition: le véhicule doit être en arrêt

postcondition: l’état du véhicule est “en marche”

description:

* L’administrateur appuie sur le bouton de démarrage
* Le système envoie un signal au traqueur GPS du véhicule concerné
* Le traqueur GPS exécute la commande de démarrage du véhicule
* Le traqueur envoie un signal au système
* le système affiche un message de succès
  1. Contraintes

Un système de Fleet Management tel que le nôtre pourrait être soumis à de contraintes de toutes sortes. Nous avons pu en ressortir celles qui suivent :

* La possibilité qu’un conducteur débranche l’équipement de tracking connecté à son véhicule.
* La possibilité qu’un véhicule traqué entre dans une zone non couverte par le réseau causant une période d’impossibilité de suivre le véhicule en temps réel.
* La possibilité que le réseau de l’opérateur de télécommunication utilisé soit de mauvaise qualité pendant une certaine période.
  1. Architecture globale

Afin de concevoir une solution n’étant pas fortement liée à Odoo, nous avons opté pour une architecture modulaire. En effet, nous avons décidé de séparer le fonctionnement métier de la plateforme en ce sens que les traitements des informations des véhicules de la flotte seront faits dans un sous-système détaché de l’ERP.

Nous en avons ressorti l’architecture suivante

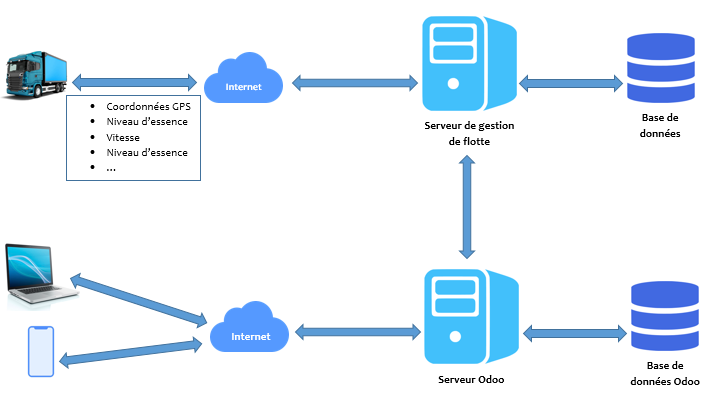


Figure 5: Architecture globale du système

1. Conception
2. Implémentation

Conclusion

Références

* [1] <http://www.adisco-cm.com/>
* [2] <https://en.wikipedia.org/wiki/Enterprise_resource_planning>
* [3] [https://www.fleetio.com](https://www.fleetio.com/)
* [4] <https://onfleet.com/>
* [5] https://www.traccar.org/
* [6] [https://www.landmarktracking.co.za](https://www.landmarktracking.co.za/)
* [7] <https://www.odoo.com/fr_FR/page/fleet>
* [8] <https://fr.wikipedia.org/wiki/Odoo>
* [9] <https://agipme.fr/2018/07/odoo-2eme-erp-le-plus-populaire.html>
* [10]