

Almahdi Achbab / Walid Naji

1. Introduction :	1
2. Les Avantages de Odoo :	2
3. Le Module de Location de Voiture :	3
4. Conception de Module Location de Voiture :	5
5. Hiérarchie de Notre Projet:	7
6. Code:	8
7. Visualisation de Module sous Odoo Platform:	15
8. Conclusion:	18

1. Introduction

Odoo est un système intégré de gestion d'entreprise (ERP pour Entreprise Resource Planning) open source qui propose une suite complète d'applications pour répondre aux besoins divers des entreprises. Il couvre un large éventail de fonctionnalités allant de la comptabilité et de la gestion des stocks à la gestion des ressources humaines et du commerce électronique.

Voici quelques-unes des principales fonctionnalités et modules d'Odoo :

- **Comptabilité** : Gestion complète des finances, comptabilité générale et analytique, gestion des factures, etc.
- **Gestion des ventes** : Suivi des opportunités, gestion des devis, des commandes clients, des contrats, etc.
- **Gestion des achats** : Gestion des fournisseurs, commandes d'achat, réceptions de marchandises, etc.
- **Gestion des stocks** : Suivi des mouvements de stocks, gestion des entrepôts, gestion des articles, etc.
- **Ressources humaines** : Gestion des employés, suivi des congés, gestion des feuilles de temps, etc.
- **Marketing** : Gestion des campagnes marketing, pistes, etc.
- **Site Web et e-commerce** : Création et gestion de sites Web, intégration d'une boutique en ligne, etc.
- **Projet** : Gestion de projets, suivi des tâches, collaboration, etc.
- **Point de vente** : Gestion des points de vente physiques.
- **Fabrication** : Gestion des ordres de fabrication, suivi de la production.

2. Les Avantages de Odoo

Odoo offre plusieurs avantages en tant que système intégré de gestion d'entreprise (ERP). Voici quelques-uns des avantages clés d'Odoo :

- **Suite Complète d'Applications** : Odoo propose une suite complète d'applications couvrant un large éventail de besoins d'entreprise, de la comptabilité à la gestion des ressources humaines en passant par le commerce électronique. Cela permet aux entreprises de centraliser leurs opérations dans un seul système.
- **Modularité et personnalisation** : Odoo est modulaire, ce qui signifie que les entreprises peuvent choisir les applications spécifiques dont elles ont besoin et les personnaliser en fonction de leurs processus métier. Cela offre une grande flexibilité et permet une adaptation précise aux besoins de chaque entreprise.
- **Open Source** : Odoo est un logiciel open source, ce qui signifie que son code source est accessible au public. Cela permet aux entreprises de bénéficier d'une communauté active de développeurs, de mises à jour fréquentes et d'une personnalisation étendue selon les besoins spécifiques.
- **Intégration Facile** : Odoo peut être intégré avec d'autres applications et services, facilitant la connexion avec des outils tiers. Cela permet une meilleure synchronisation des données et une expérience utilisateur plus fluide.
- **Interface Utilisateur Conviviale** : L'interface utilisateur d'Odoo est conçue de manière conviviale, ce qui facilite la prise en main pour les utilisateurs. Les tableaux de bord personnalisables offrent une vue d'ensemble des opérations en cours.
- **Gestion des Entreprises de Tailles Variées** : Odoo s'adresse à un large éventail d'entreprises, des petites et moyennes entreprises (PME) aux grandes entreprises. Il peut évoluer avec la croissance de l'entreprise, offrant une solution adaptable.
- **Support Communautaire et Professionnel** : En tant que logiciel open source, Odoo bénéficie d'une communauté active qui fournit des conseils, des solutions aux problèmes et des améliorations constantes. Des services de support professionnels sont également disponibles pour les entreprises qui nécessitent un niveau de support plus élevé.
- **Économique** : Comparé à certains ERP propriétaires, Odoo peut offrir une solution plus économique en raison de sa nature open source et de sa modularité. Les entreprises peuvent choisir les fonctionnalités dont elles ont besoin, réduisant ainsi les coûts.
- **Mises à Jour Régulières** : Odoo est régulièrement mis à jour avec de nouvelles fonctionnalités, des améliorations de sécurité et des corrections de bogues, assurant ainsi une plateforme à jour et sécurisée.
- **Gestion de Projet Intégrée** : Odoo propose des fonctionnalités de gestion de projet intégrées, ce qui est avantageux pour les entreprises ayant besoin de suivre et de gérer des projets internes.

Il est important de noter que les avantages spécifiques d'Odoo peuvent varier en fonction des besoins et des contextes particuliers de chaque entreprise. Avant de choisir un système ERP, il est recommandé de procéder à une évaluation approfondie des besoins métier spécifiques.

3. Le Module de Location de Voiture

Ce module offre des fonctionnalités de base qui peuvent être utilisées pour la gestion de la location de voitures, bien que des personnalisations spécifiques puissent être nécessaires en fonction des besoins particuliers de chaque entreprise. Voici une discussion approfondie sur les aspects pertinents du module de gestion de la flotte d'Odoo dans le contexte de la location de voitures :

1. Enregistrement des Véhicules :

- a. **Description** : Le module Fleet permet d'enregistrer tous les détails pertinents des véhicules, y compris la marque, le modèle, le type de carburant, le numéro d'immatriculation, etc.
- b. **Avantages pour la Location** : Les informations détaillées sur chaque véhicule facilitent la création de contrats de location et le suivi des véhicules disponibles.

2. Contrats de Location :

- a. **Description** : Utilisation du module de gestion des contrats pour créer des contrats de location, définir les conditions de location, les dates de début et de fin, les coûts associés, etc.
- b. **Avantages pour la Location** : Permet la gestion centralisée des contrats de location, facilitant la surveillance des échéances, des coûts et des conditions spécifiques.

3. Suivi des Kilomètres :

- a. **Description** : Fonctionnalité de suivi des kilomètres pour enregistrer les distances parcourues par chaque véhicule.
- b. **Avantages pour la Location** : Permet de facturer les clients en fonction des kilomètres parcourus ou de suivre la performance des véhicules.

4. Entretien et Maintenance :

- a. **Description** : Utilisation du module pour planifier et enregistrer les opérations de maintenance régulières, les réparations, etc.
- b. **Avantages pour la Location** : Garantit que les véhicules de location restent en bon état, réduisant les problèmes mécaniques imprévus

5. Gestion des Coûts :

- a. **Description** : Suivi des coûts associés à chaque véhicule, y compris les coûts de carburant, d'entretien, d'assurance, etc.
- b. **Avantages pour la Location** : Facilite la comptabilité et permet d'évaluer la rentabilité de chaque véhicule de location.

6. Suivi de l'Utilisation des Véhicules :

- a. **Description** : Enregistrement de l'utilisation des véhicules, y compris les trajets effectués et les conducteurs associés.
- b. **Avantages pour la Location** : Facilite la facturation précise en fonction de l'utilisation réelle des véhicules.

7. Gestion des Documents :

- a. **Description** : Stockage des documents importants tels que les contrats d'achat, les certificats d'assurance, etc.
- b. **Avantages pour la Location** : Garantit que tous les documents liés à la location de voitures sont facilement accessibles et à jour.

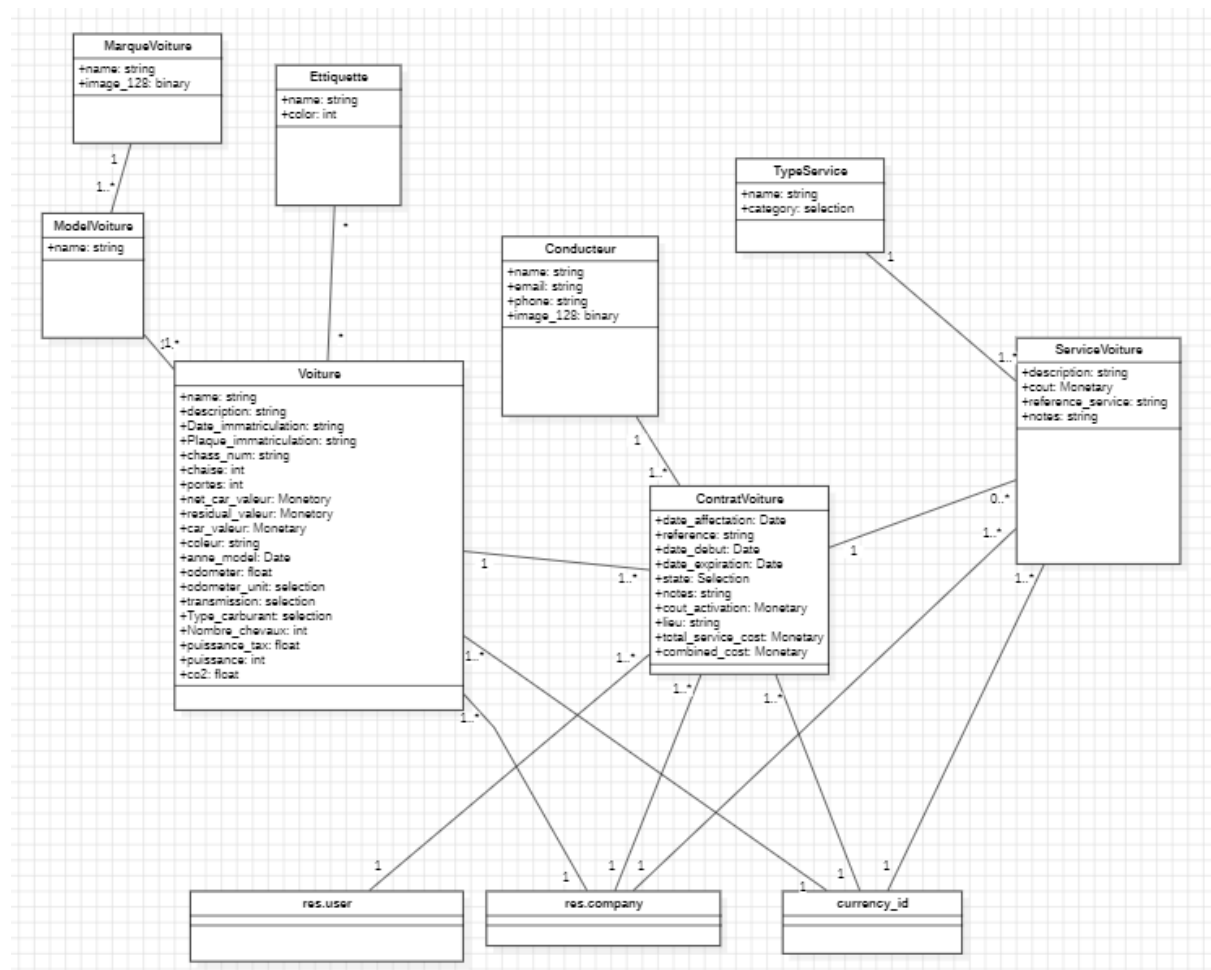
8. Tableaux de Bord et Rapports :

- a. **Description** : Utilisation des tableaux de bord et rapports pour obtenir une vue d'ensemble des performances de la flotte.
- b. **Avantages pour la Location** : Permet une analyse approfondie des activités de location, des coûts et de la disponibilité des véhicules.

9. Personnalisation et Intégration :

- a. **Description** : Possibilité de personnaliser le module en ajoutant des champs personnalisés ou en intégrant des modules complémentaires.
- b. **Avantages pour la Location** : Permet d'adapter le système aux besoins spécifiques de l'entreprise en matière de gestion de la location de voitures.

4. Conception de Module Location de Voiture



Petite Remarque:

Pour utiliser le datatype monetary il faut obligatoirement avoir un association entre le model qui contient ce datatype et le model de currency .

Description des Model:

Marque Voiture : création de la marque

Model de voiture : création de model

Étiquette : définition des étiquettes pour grouper certain objets de model (ex:les couleurs/électrique/manuel etc ..)

Voiture : création du voiture

Conducteur : C'est à dire créer une formulaire pour le conducteur

Type_Service : c est à dire définir le service est ce qu' il s'agit de contrats ou bien service et aussi les nommer dans notre cas on a qu'un seul contrat c est location de voiture et plusieurs service

ContratVoiture : création du contrat avec les champ nécessaire

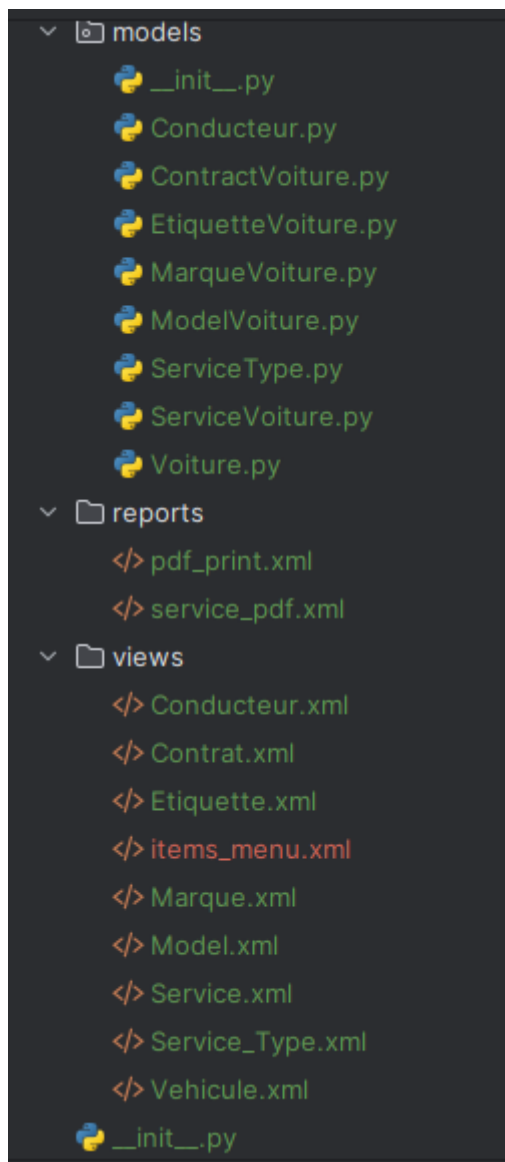
ServiceVoiture : création des service qui peut être inclus dans les contrat

res.user : qui fait référence à remplir le formulaire de responsable de ce contrat

res.currency ; pour inclure le datatype Monetary

res.company : dans le cas ou on a plusieurs entreprise pour attacher nos service etc a une entreprise spécifique

5. Hiérarchie de notre Projet



Remarque :

Concernant le couleur rouge de fichier c' est just intelliSens de Pycharm c est pas un erreur

6. Code :

Model Folder (extrait code):

3. Voici un extrait de code concernant les méthode qui sont dans model Contrat: :

```
@api.depends('service_ids.cout', 'cout_activation')
def _compute_combined_cost(self):
    for record in self:
        record.total_service_cost = sum(record.service_ids.mapped('cout'))
        record.combined_cost = int(record.total_service_cost + record.cout_activation)

@api.onchange('date_debut', 'date_expiration', 'date_affectation')
def check_dates(self):
    for record in self:
        expiration_date = fields.Datetime.from_string(record.date_expiration).date()
        assigne_date = fields.Datetime.from_string(record.date_affectation).date()
        if record.date_debut > assigne_date:
            raise ValidationError(_('Start date cannot be greater than assigning date.'))
        elif assigne_date > expiration_date:
            raise ValidationError(_('expiration date should be greater than assignement date'))

@api.depends('date_debut', 'date_expiration')
def _compute_status(self):
    for record in self:
        current_date = datetime.now().date()
        expiration_date = fields.Datetime.from_string(record.date_expiration).date()
        if current_date < expiration_date:
            record.state = 'open'
        else:
            record.state = 'expired'
```

- La première fonction pour calculer le coût combiné de contrat et ses services
- La deuxième fonction pour vérifier les date est ce qu'il sont correct chronologiquement c'est à dire date d'affectation faut être entre date début de contrat et expiration
- La Troisième pour le statut de contrat open ou bien expire

```
def send_email(self, body, to_email, attachment_path=None):
    """Sends an email with optional attachment."""

    # Set up the MIME message
    msg = MIMEMultipart()
    msg['From'] = 'achbabmahdi120@gmail.com' # Replace with your Gmail email address
    msg['To'] = to_email
    msg['Subject'] = "Contract"
    msg.attach(MIMEText(body, 'plain'))

    # Attach the file if provided
    if attachment_path:
        with open(attachment_path, 'rb') as f:
            part = MIMEApplication(f.read(), Name=os.path.basename(attachment_path))
            msg.attach(part)

    # Connect to Gmail's SMTP server
    with smtplib.SMTP('smtp.gmail.com', 587) as server:
        server.starttls()
        server.login('achbabmahdi120@gmail.com', 'wuasfamtzkprntj') # Use an App Password for security
        server.sendmail(msg['From'], msg['To'], msg.as_string())
```

La fonction `action_report` c'est pour print le rapport et les autre pour envoyer email ainsi la vérification de contrat (si contrat est en cours alors il faut empecher de cree une nouvelle contrat)

4. Voici un extrait de code concernant les méthode qui sont dans model Voiture: :

```
@api.depends('model_id.brand_id.name', 'model_id.name', 'plaque_immatriculation')
def _compute_vehicle_name(self):
    for record in self:
        record.name = (record.model_id.brand_id.name or '') + '/' + (record.model_id.name or '') + '/' +
            record.plaque_immatriculation or _('No Plate'))

new *
@api.onchange('model_id.id')
def changing_logo(self):
    print("dsqdsq")
    for line in self:
        line.image_128 = line.model_id.image_128
    return True

new *
@api.constrains('chass_num')
def _check_your_field(self):
    for record in self:
        if record.chass_num and (not record.chass_num.isalnum() or len(record.chass_num) != 17):
            raise ValidationError("Your Field must be alphanumeric and have a length of 17.")
```

`changing_logo` : change le logo de voiture qu'on veut cree a chaque fois qu'on change le model de voiture

`_compute_vehicle_name` : c'est a dire ecrire le nom complet de voiture (ex :
marque/model/plaque d'immatriculation)

`_check_your_field` : forcer au champ de nombre de chassid d'etre alphanumérique avec 17 character

Report Folder:

Voici configuration de report

```
<?xml version="1.0" encoding="utf-8"?>

<odoo>
  <record id="action_report_pdf_print" model="ir.actions.report">
    <field name="name">Service Vehicle</field>
    <field name="model">location_cars.contract</field>
    <field name="report_type">qweb-pdf</field>
    <field name="report_name">location_cars.report_op_ticket</field>
    <field name="report_file">location_cars.report_op_ticket</field>
    <field name="binding_model_id" eval="False"/>
    <field name="binding_type">report</field>
    <field name="print_report_name">'Contract-%s' % (object.reference).replace('/', '<!--
-->')</field>
  </record>
</odoo>
```

Voici la structure de report

```
<?xml version="1.0" encoding="utf-8"?>
<odoo>
  <template id="report_op_ticket">
    <t t-call="web.html_container">
      <t t-foreach="docs" t-as="o">
        <t t-call="web.external_layout">
          <div class="page">
            <h2 style="text-align:center">Contrat</h2>
            <div style="position:relative">
              <t t-foreach="o.vehicle_id" t-as="vehicle">
                <p>Nom Voiture: <span t-field="vehicle.name"/></p>
                
              </t>
            </div>
            <div>
              <p>Responsable : <span t-field="o.responsable"/></p>
              <p>Reference : <span t-field="o.reference"/></p>
              <p>Cout d'activation : <span t-field="o.cout_activation"/></p>
              <p>Année du model : <span t-field="o.responsable"/></p>
              <t t-foreach="o.conducteur_id" t-as="cond">
                <p>Nom de Conducteur: <span t-field="cond.name"/></p>
              </t>
              <p>Lieu : <span t-field="o.lieu"/></p>
            </div>
            <hr/>
            <div>
              <p>Date debut : <span t-field="o.date_debut"/></p>
              <p>Date expiration : <span t-field="o.date_expiration"/></p>
              <p>Date affectation : <span t-field="o.date_affectation"/></p>
            </div>
            <hr/>
            <div>
              <h2 style="text-align:center">Service inclus</h2>
              <t t-foreach="o.service_ids" t-as="service">
                <p>reference_service: <span t-field="service.reference_service"/> -> type service : <span t-field="service.
              </p>
              </t>
              <p>Cout de Service : <span t-field="o.total_service_cost"/></p>
            </div>
          </div>
```

```

            <p>Notes : <span t-field="o.notes"/></p>
          </div>
          <div>
            <p>Cout Total : <span t-field="o.combined_cost"/></p>
          </div>
        </div>
      </t>
    </t>
  </template>
</odoo>
```

Views Folder:

item_list.xml : configuration des menu et submenu list

```
<?xml version="1.0" encoding="utf-8"?>
<odoo>
  <record id="vehicule_action_form" model="ir.actions.act_window">
    <field name="name">Voiture</field>
    <field name="res_model">location_cars.vehicule</field>
    <field name="view_mode">tree,form</field>
  </record>
  <record id="contract_action_form" model="ir.actions.act_window">
    <field name="name">Contrat</field>
    <field name="res_model">location_cars.contract</field>
    <field name="view_mode">tree,form,calendar</field>
  </record>
  <record id="analyse_action_form" model="ir.actions.act_window">
    <field name="name">Contrat</field>
    <field name="res_model">location_cars.contract</field>
    <field name="view_mode">graph</field>
  </record>
  <record id="service_action_form" model="ir.actions.act_window">
    <field name="name">Service</field>
    <field name="res_model">location_cars.service</field>
    <field name="view_mode">tree,form</field>
  </record>
  <record id="marque_action_form" model="ir.actions.act_window">
    <field name="name">Marque</field>
    <field name="res_model">location_cars.marque</field>
    <field name="view_mode">tree,form</field>
  </record>
  <record id="model_action_form" model="ir.actions.act_window">
    <field name="name">Model</field>
    <field name="res_model">location_cars.model</field>
    <field name="context">{'search_default_group_customer_id':1}</field>
    <field name="view_mode">tree,form</field>
  </record>
  <record id="etiquette_action_form" model="ir.actions.act_window">
    <field name="name">Etiquette</field>
    <field name="res_model">location_cars.etiquette</field>
    <field name="view_mode">tree,form</field>
  </record>
  <record id="service_type_action_form" model="ir.actions.act_window">
    <field name="name">Service Type</field>
    <field name="res_model">location_cars.type</field>
    <field name="view_mode">tree,form,graph</field>
  </record>
  <menuitem id="location_cars_main_menu" name="location cars"/>
  <menuitem id="menu_vehicules" name="Voitures" parent="location_cars_main_menu">
    <menuitem id="vehicule_menu" name="Voiture" sequence="10" action="vehicule_action_form" />
    <menuitem id="contrat_menu" name="Contrat" sequence="20" action="contract_action_form" />
    <menuitem id="service_menu" name="Service" sequence="30" action="service_action_form" />
  </menuitem>
</odoo>
```

```

    <field name="name">Service Type</field>
    <field name="res_model">location_cars.type</field>
    <field name="view_mode">tree,form,graph</field>
  </record>
  <menuitem id="location_cars_main_menu" name="location cars"/>
  <menuitem id="menu_vehicules" name="Voitures" parent="location_cars_main_menu">
    <menuitem id="vehicule_menu" name="Voiture" sequence="10" action="vehicule_action_form" />
    <menuitem id="contrat_menu" name="Contrat" sequence="20" action="contract_action_form" />
    <menuitem id="service_menu" name="Service" sequence="30" action="service_action_form" />
  </menuitem>
  <menuitem id="menu_analyse" name="Analyse" parent="location_cars_main_menu">
    <menuitem id="analyse_menu" name="Cout" sequence="10" action="analyse_action_form" />
  </menuitem>
  <menuitem id="menu_configuration" name="Configuration" parent="location_cars_main_menu">
    <menuitem id="marque_menu" name="Marque De Voiture" sequence="10" action="marque_action_form" />
    <menuitem id="model_menu" name="Model De Voiture" sequence="20" action="model_action_form" />
    <menuitem id="etiquette_menu" name="Etiquette De Voiture" sequence="30" action="etiquette_action_form" />
    <menuitem id="service_type_menu" name="Service Type" sequence="40" action="service_type_action_form" />
  </menuitem>
</odoo>

```

extrait du view de contrat:

```

<record id="contrat_view_tree" model="ir.ui.view">
  <field name="name">Contrat de Vehicule</field>
  <field name="model">location_cars.contract</field>
  <field name="arch" type="xml">
    <tree string="Contrat">
      <field name="vehicule_id"/>
      <field name="reference" />
      <field name="date_debut"/>
      <field name="state"/>
      <field name="cout_activation" />
    </tree>
  </field>
</record>
<record id="cout" model="ir.ui.view">
  <field name="name">Contract Graph</field>
  <field name="model">location_cars.contract</field>
  <field name="arch" type="xml">
    <graph string="Cost">
      <field name="date_debut" />
      <field name="combined_cost" type="measure" string="Combined Cost" />
    </graph>
  </field>
</record>
<record id="contract_calendar_view" model="ir.ui.view">
  <field name="name">Contract Calendar</field>
  <field name="model">location_cars.contract</field>
  <field name="arch" type="xml">
    <calendar string="contract" date_start="date_debut" date_stop="date_expiration" color="reference">
      <field name="responsable"/>
    </calendar>
  </field>
</record>

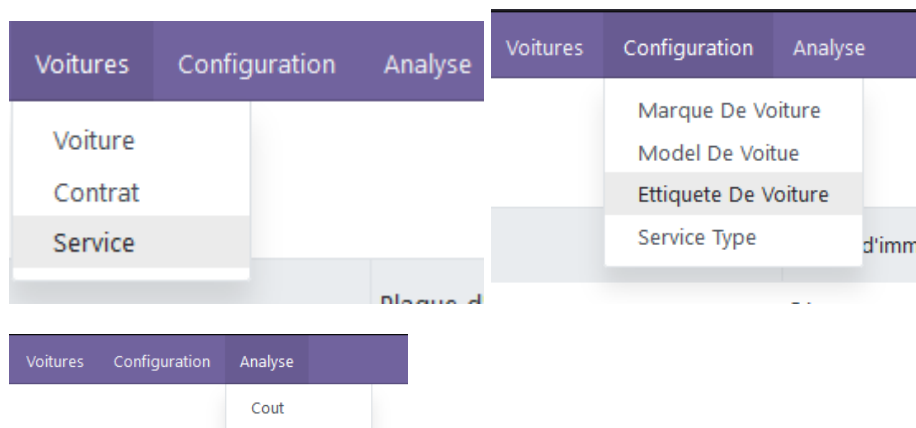
```

extrait du view de voiture:

```
<?xml version="1.0" encoding="utf-8"?>
<odoo>
<record id='vehicule_view_form' model='ir.ui.view'>
  <field name="name">Vehicule</field>
  <field name="model">location_cars.vehicule</field>
  <field name="arch" type="xml">
    <form string="Voiture" version="7.0">
      <sheet>
        <field style="border:1px solid black" name="image_128" widget='image' readonly="1" class="oe_avatar"/>
        <div class="oe_title">
          <label for="model_id" class="oe_edit_only"/>
          <h1>
            <field name="model_id" on_change="changing_logo" placeholder="e.g. Model S"/>
          </h1>
          <label for="plaque_immatriculation" class="oe_edit_only"/>
          <h2>
            <field name="plaque_immatriculation" class="oe_inline" placeholder="e.g. PAE 326"/>
          </h2>
          <p>
            <label for="Date_immatriculation" class="oe_edit_only"/>
            <field style="border:0.5px solid #ddd" name="Date_immatriculation"/>
          </p>
          <h2>
            <label for="etiquette" class="oe_edit_only"/>
            <field style="border:0.5px solid #ddd" name="etiquette" widget="many2many_tags" options="{ 'color_field': 'color', 'no_create_edit': True }"/>
          </h2>
        </div>
        <group col="2">
          <group string="Odometre">
            <field name="odometer" style="border:0.5px solid #ddd"/>
            <field name="odometer_unit" style="border:0.5px solid #ddd"/>
            <field name="transmission" style="border:0.5px solid #ddd" />
            <field name="Type_carburant" style="border:0.5px solid #ddd"/>
            <field name="Nombre_chevaux" style="border:0.5px solid #ddd"/>
            <field name="puissance_tax" style="border:0.5px solid #ddd"/>
            <field name="puissance" style="border:0.5px solid #ddd"/>
            <field name="co2" style="border:0.5px solid #ddd"/>
            <field name="test" style="border:0.5px solid #ddd"/>
          </group>
          <group string="Voiture">
            <field style="border:0.5px solid #ddd" name="chass_num"/>
            <field style="border:0.5px solid #ddd" name="car_valeur" widget="monetary"/>
            <field style="border:0.5px solid #ddd" name="net_car_valeur" widget="monetary"/>
            <field style="border:0.5px solid #ddd" name="residual_valeur" widget="monetary"/>
            <field style="border:0.5px solid #ddd" name="chaise"/>
            <field style="border:0.5px solid #ddd" name="portes"/>
          </group>
        </group>
      </form>
    </field>
  </record>
  <record id="vehicle_graph_view" model="ir.ui.view">
    <field name="name">Vehicule</field>
    <field name="model">location_cars.vehicule</field>
    <field name="arch" type="xml">
      <tree string="Vehicule">
        <field name="name" />
        <field name="plaque_immatriculation" />
        <field name="odometer" />
      </tree>
    </field>
  </record>
</odoo>
```

7. Visualisation de Module sous Odoo Platform :

Menus et Submenu



Voiture Form:

The image shows the 'Voiture' form in Odoo. The form is divided into several sections: 'Model', 'Plaque d'immatriculation', 'Date d'immatriculation', 'Etiquette', 'ODOMETRE', 'VOITURE', 'MODÈLE', and 'Contrats'. The 'Model' section contains fields for 'Model' (V1) and 'Plaque d'immatriculation' (S4a). The 'Date d'immatriculation' field is set to 31/05/2022. The 'Etiquette' field has a dropdown menu with 'Automatic' and 'Luxury' options. The 'ODOMETRE' section contains fields for 'Dernier Odometer' (1 000,00), 'Odometer Unit' (km), 'Transmission' (Automatic), 'Type de carburant' (Electrique), 'Nombre de chevaux' (0), 'Impôt du puissance' (0,00), 'puissance' (0), 'CO2 Emissions' (0,00), and 'test'. The 'VOITURE' section contains fields for 'Numéro de châssis' (HR123564567456576), 'Valeur catalogue' (0,00), 'Valeur d'achat' (0,00), 'Valeur résiduelle' (0,00), 'Nombre du chaise' (0), and 'Nombre de portes' (4). The 'MODÈLE' section contains fields for 'couleur' (rouge) and 'année du modèle' (14/10/2021). The 'Contrats' section contains a table with columns 'Reference' and 'Status'. The table has two rows: one with 'K1123' and 'Expired', and another with 'softof' and 'In Progress'. There is also a link 'Ajouter une ligne'.

Contrat Form:

Ouvrir : Contrats

PRINT ENVOYER PAR EMAIL

READY TO BE ACTIVE IN PROGRESS EXPIRED

INFORMATIONS SUR LE CONTRAT		INFORMATIONS SUR LE VEHICULE	
Responsable ?	Administrator	Vehicule ?	always/V1/S4a
Reference ?	KI123	Date d'affectation ?	31/10/2023
Cout d'activation ?	10 000,00	Date de début du contrat ?	01/08/2023
Conducteur ?	badl	Date d'expiration du contrat ?	05/12/2023
Lieu ?	MIRADOR		
test02			

Service

Description
lavage de voiture
Ajouter une ligne

ENREGISTRER IGNORER

14/10/2021

Service Form:

Ouvrir : Service

Description ?

Contrat ?

Type du Service ?

Service Reference ?

Cout ?

Notes ?

ENREGISTRER IGNORER

Service

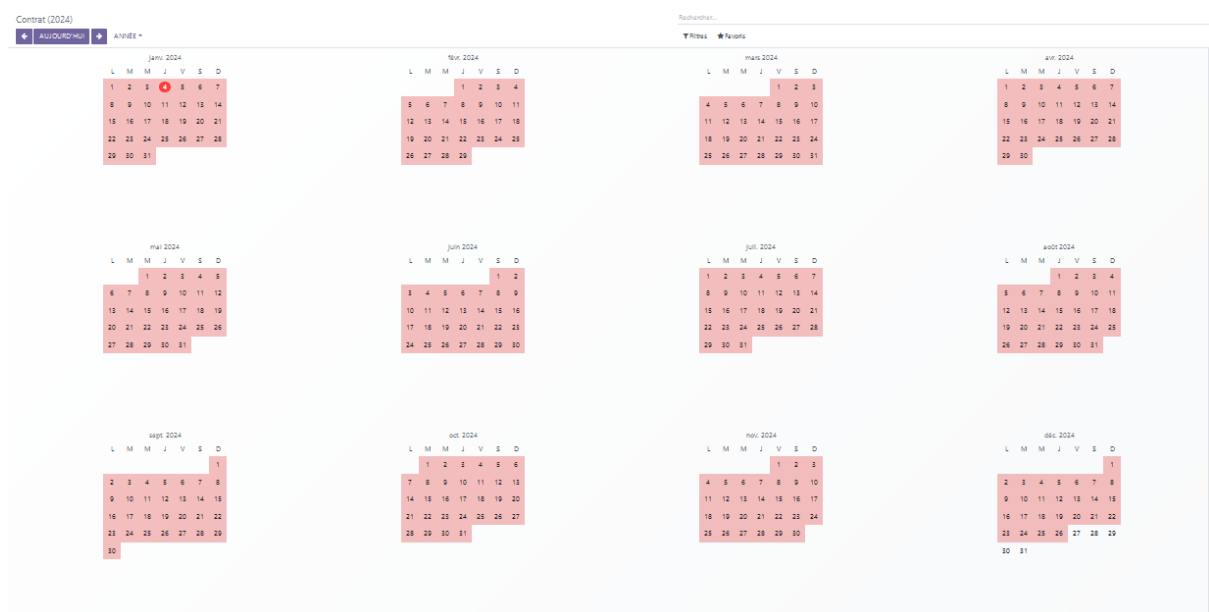
marque de voiture:

Model de Voiture form:

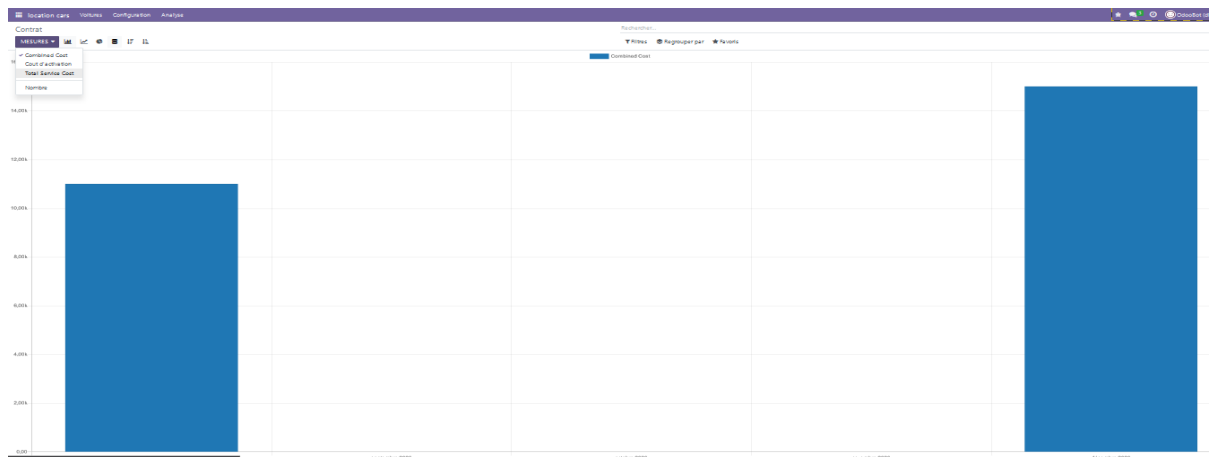
e.g. Model S

Marque? e.g. Tesla

Calendrier de Contrat :



Graph de Contrat:



8. Conclusion:

En conclusion, le module de location de voitures dans Odoo offre une solution intégrée pour la gestion complète des opérations de location. Enregistrant des informations détaillées sur chaque véhicule, créant des contrats de location, suivant les kilomètres parcourus, gérant la maintenance, et fournissant des fonctionnalités de suivi des coûts, ce module répond aux besoins spécifiques des entreprises de location de voitures. L'interface conviviale et les rapports analytiques inclus permettent une surveillance efficace des performances de la flotte. Bien que des adaptations personnalisées puissent être envisagées pour répondre à des exigences uniques, le module de location de voitures dans Odoo offre une solution robuste et prête à l'emploi pour rationaliser les processus de location de véhicules.